UCLouvain

LOUVAIN
School of Management

Louvain School of Management

Machine Learning versus the Comparables Method: Predicting Transaction Multiples in European Mid-Market M&A

Author: Valentine Van Herstraeten
Supervisor: Nathanaël Betti
Academic year 2025-2026
Dissertation for the master of Business Engineering & Management
Machine Learning in Investment Banking
Daytime schedule

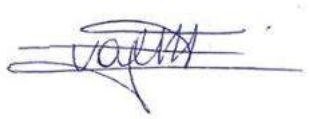
Statements on responsible use of generative AI

The content of my master's thesis is my original output, even if I used generative AI to help prepare it.	☒Yes ☐No
I master the theories, tools and methods that I use for the thesis.	☒Yes ☐No
I verified that the output presented in the thesis is accurate and valid, and that it contains no hallucination, no false reference, etc.	☒Yes ☐No
I acknowledge that I master the final output and that I am able to explain it and answer questions about it.	☒Yes ☐No
I made sure not to provide generative AI tools with access to confidential data or information	☒Yes ☐No

Signature

Van Herstraeten Valentine

11th August 2026



Acknowledgements

I would first like to express my sincere gratitude to my supervisor, Nathanaël Betti, for his valuable guidance and advice throughout this thesis. I am particularly grateful for his encouragement to approach my work in a structured way, to carefully assess potential risks while considering alternative approaches, and to clearly emphasise the contributions of my work to the existing literature. His guidance has been highly valuable throughout this research process.

I would also like to thank my parents, my brother, and my sister for their constant support and encouragement throughout my studies. I am especially grateful to my father, Olivier, for his careful proofreading and for his valuable advice, which greatly contributed to the completion of this thesis.

Finally, I would like to thank my professors and friends who have accompanied me throughout my studies. Their support, discussions, and different perspectives have made my academic journey both more enriching and more enjoyable.

Table of Content

Acknowledgements ..... ***II***
List of Tables ..... ***V***
List of Figures ..... ***V***
Introduction ***1***
Literature Review ***2***
Valuation in M&A: Intrinsic and Relative Methods 2
Relative Valuation, Peer Selection, and the Private Company Discount 4
Machine Learning in Finance and Private Company Valuation 6
Financial Fundamentals as Predictors of Private M&A Multiples 9
Methodology ***12***
Research Design 12
Sample Construction 13
Comparable Company Set Construction 14
Variable Selection and Data Preprocessing 15
Train/Test Construction 18
Analyst Method Construction and Calibration 19
Linear Regression 20
Machine Learning Models 22
Results and Analysis ***24***
The Analyst Method 24
Linear Regression 26
Machine Learning 28
Comparison and Analysis of All Methods 32
Limitations and Future Research ***36***
Sample Size 36
Historical Data and Imputation 37
Analyst Benchmark Design 38
Scope 38
Conclusion ***39***
Bibliography ***41***
Appendices ***45***
Methodology 45
The Analyst Method 47
Linear Regression 49
Machine Learning 52
Comparison and Analysis of All Methods 56
Glossary 58
R Markdown 59
Preprocessing 59
Setup 59
Candidate variables 60
Equity ratio 61

Correlation matrix 62
Regression feature selection 63
Cross-validation folds 67
Regression diagnostics 68
Ridge 68
EV/Revenue regression 70
Augmented regressions 71
ML feature sets 73
Grid search 74
Grid search, augmented 77
Scoring functions 79
Opening the test set 79
Regression on the test set 80
Buckets for the regression 82
Final ML models, pure 83
Final ML models, augmented 86
Buckets and extremes 89
Feature importance 91
Analyst benchmark 93

List of Tables

Table 1. Machine learning applied to valuation/private settings
Table 2. Hyperparameter grids
Table 3. Sample construction
Table 4. Sector composition and train/test allocation
Table 5. Retained explanatory variables
Table 6. Feature sets used by each model
Table 7. Variable transformations applied
Table 8. Train/test balance diagnostics
Table 9. Discount calibration under two error criteria
Table 10. Discount sensitivity on the test set (EV/EBITDA)
Table 11. Analyst benchmark test-set performance
Table 12. Analyst benchmark by transaction bucket (EV/EBITDA)
Table 13. Analyst benchmark by sector
Table 14. Feature selection sequence and multicollinearity
Table 15. OLS coefficients: EV/EBITDA, pure specification
Table 16. OLS coefficients: EV/Revenue, pure specification, HC3 standard errors
Table 17. Regression specifications compared
Table 18. OLS and Ridge coefficients compared (EV/EBITDA)
Table 19. Regression test-set performance
Table 20. Cross-validation results: all sixteen specifications
Table 21. ML test-set performance
Table 22. ML test-set performance: additional metrics
Table 23. Feature importance: EV/EBITDA, pure specifications
Table 24. Feature importance: EV/EBITDA, augmented specifications
Table 25. Feature importance: EV/Revenue compared with EV/EBITDA
Table 26. Master comparison: all methods and both multiples
Table 27. Bucket decomposition: all methods (EV/EBITDA)
Table 28. Bucket decomposition (EV/Revenue)
Table 29. Glossary of abbreviations

List of Figures

Figure 1. Discount calibration curves
Figure 2. Distribution of EV/EBITDA transaction multiples
Figure 3. MedAE across all methods
Figure 4. Feature importance: EV/EBITDA, pure specifications
Figure 5. Feature importance: EV/EBITDA, augmented specifications
Figure 6. The asymmetry between the two multiples
Figure 7. Performance by transaction bucket

Introduction

Private companies have no price. To value a company, investment bankers follow two logics. Intrinsic valuation, of which the discounted cash flow (DCF) model is the standard form, bases the valuation on the projected cashflows of the firm. Relative valuation instead prices the firm against its peers, on the assumption that similar assets should trade at similar ratios to earnings or revenue. Applied to a private target, intrinsic valuation compounds three estimation problems: cash flow projections come from a management team with an interest in optimism, the terminal value (TV) that usually dominates the result is highly sensitive to the assumed growth and to the cost of capital, and the discount rate must itself be proxied from listed peers, which reintroduces the comparison problem the exercise was meant to avoid. Practice therefore mainly leans on the second, usually by averaging both results (Guo et al., 2023).

Comparable company analysis (CCA) proceeds in two steps. A set of listed firms operating in the same sector is assembled, the multiple at which they trade is computed, and its median is applied to the target's own earnings or revenue. Because a private company cannot be sold on demand, that figure is then reduced by a private company discount (PCD), conventionally set around twenty percent (Koeplin et al., 2000). The result is close to arithmetic, and the discount at its centre is a convention rather than an estimate. The same reduction is applied to a fast-growing consultancy and to a declining printing business.

Machine learning (ML) has meanwhile displaced linear methods across empirical finance. Gu et al. (2020) roughly doubled the predictive accuracy of equity return models using tree ensembles, and Geertsema and Lu (2023) and Staňková (2024) show the same tools beating peer-group benchmarks when pricing listed firms. Both of those studies, however, price companies that already have an observable market price. The private mid-market, where the comparables rule actually determines what changes hands, has never been tested against a data-driven alternative. Indeed, this thesis addresses that gap and contributes on three fronts. It moves the comparison to a market where no price exists until the transaction closes and where the incumbent is a calibrated professional convention rather than a statistical construction. It holds the information set constant by supplying the ML models with the same comp-set median that the benchmark uses as its own anchor, which allows the study to explain why one method wins rather than only which one does. And by locating where along the distribution of transactions the practitioner method fails, and by bounding how much of that

failure the discount parameter can account for, it yields a diagnosis an advisory team can act on rather than a verdict on model performance alone.

This thesis runs that test on 513 European Business Services transactions closed between 2015 and 2026, combining deal data with target financials. Four methods predict the same held-out transactions. The analyst benchmark is reproduced mechanically, as a sector-year comp-set median scaled by a calibrated discount. Random Forest (RF) and Extreme Gradient Boosting (XGB) are each run in a pure specification using only company fundamentals, and in an augmented one that adds the comp-set median, the Stoxx Europe 600 and the ten-year rate at closing. An ordinary least squares (OLS) regression is also included, not as a competitor but as a control, isolating whether any ML advantage comes from non-linearity rather than from access to more variables. Both enterprise value (EV) over earnings before interest, taxes, depreciation and amortisation (EBITDA) and EV over revenue multiples are predicted.

The central difficulty is that private companies disclose little. Listed firms report quarterly under standardised rules, whereas private European companies file with national registries whose requirements differ by country and by company size, and smaller firms in lighter-disclosure jurisdictions often file abbreviated accounts with no profit-and-loss detail (Bourveau et al., 2023).

The question this raises is whether a model trained on such data can outperform the rule it would replace. Answering it requires four sub-questions. Do financial fundamentals carry any measurable relationship with private transaction multiples in the first place? If ML improves on the linear baseline, is that gain attributable to non-linear structure rather than to specification? Does supplying the models with market information, including the comp-set median that the benchmark itself uses, narrow whatever gap remains? And does the answer hold uniformly, or does it depend on which multiple is being predicted and on where the transaction sits in the distribution?

Literature Review

Valuation in M&A: Intrinsic and Relative Methods

Valuation in M&A is in between finance theory and practice, and the gap is even more visible when it comes to choosing between a DCF intrinsic valuation and a multiples-based relative valuation. As Koeplin et al. (2000) state, the DCF method is the theoretically correct way to

proceed: it values a company as the present value of its expected future free cash flows, discounted at a rate that reflects the systematic risk (Damodaran, 2006). It is computed with this general formula:

$$EV = \sum_{t=1}^{n} \frac{FCFF_t}{(1+WACC)^t} + \frac{TV}{(1+WACC)^n}$$

where EV is the Enterprise Value, TV is the Terminal Value, FCFF represents Free Cash Flow to the Firm, and WACC is the Weighted Average Cost of Capital. In practice, applying DCF to private targets implies three layers of estimation that can each introduce bias. Firstly, CF projections for private companies highly depends on the forecasted business plan of the management, that is very often optimistic when it comes to selling the company. Secondly, the TV can also be over- or under-valued since it is based on the assumed long-term growth and is highly sensitive to the cost of capital. Since it is a dominant component of the EV when computing DCF, a small change on either the growth or the cost of capital has a crucial impact on the EV. Thirdly, estimating a discount rate for private companies also introduce a lot of potential bias since it requires proxying systematic risk using public comparables, which reintroduces the peer selection issue. As suggested by Koeplin et al. (2000), other valuation methods are useful in addition to the DCF to serve as reality check on the implied EV, and CCA is often the chosen one. That is even more accurate for the valuation of privately held companies, since they do not come with an observable stock price that can be used for assessing market value.

These estimation challenges explain why relative valuation, including multiples-based approaches, has become an essential practical method in mergers and acquisitions (M&A) advisory that is used by investment banks in all M&A transactions (Eaton et al., 2022). Indeed, Gompers, Kaplan, and Mukharlyamov (2016) explain that private equity (PE) investors prefer CCA over DCF. Shaffer (2024) also states that accounting multiples are key in M&A deals, documenting a trend toward using multiples over the past two decades. The CCA method is especially useful because it bypasses explicit cash flow projections. In fact, it anchors valuation to what comparable firms are trading at in the market, embedding the market's own assessment of growth, risk, and profitability in a single ratio. Between different multiples, EV multiples have become dominant over equity multiples, which is a logical choice in M&A transactions. Indeed, EV, that represents market capitalisation plus net financial debt, measures the value of the firm's operating assets independently of how they are financed, which matters most because acquirers typically refinance a target's debt at closing. As

explained by Koeplin et al. (2000), metrics such as earnings before interest and taxes (EBIT) and EBITDA are independent of the capital structure of the company; by contrast, earnings figure also account for the mix of debt and equity of the company. Shaffer (2024) also expresses a preference for using EBITDA as a value driver and CF proxy instead of EBIT or net income. EV/EBITDA and EV/Revenue, both EV multiples used for predictions in this thesis, measure economically distinct quantities and are driven by different fundamentals. Starting from the Gordon constant-growth model, the equation for the revenue multiple is the following:

$$\frac{EV}{Revenue} = \frac{pm \cdot (1 + g)}{r - g}$$

where pm is the profit margin, g is the expected growth rate, and r is the cost of capital. This equation implies that the revenue multiple is jointly proportional to profitability and to the growth factor. In other words, a firm earns a high EV/Revenue only when margin and growth are simultaneously high (Bhojraj & Lee, 2002; Damodaran, 2006). EV/EBITDA occupies a different analytical position. Indeed, EBITDA is the standard base used for sizing leveraged buyout (LBO) debt. This implies that the warranted EV/EBITDA reflects not only the target's earnings but the actual financing environment, including the cost of debt, the leverage market's risk aversion, and the PE buyer's required return. These theoretical priors predict a structural asymmetry between the two multiples: market-context variables such as interest rates and sector sentiment should improve EV/EBITDA prediction, while growth fundamentals should mainly dominate EV/Revenue.

Relative Valuation, Peer Selection, and the Private Company Discount

The theoretical foundation of relative valuation is the Law of One Price. In an efficient market, assets with identical cash flows and risk must trade at identical prices, so firms with equivalent fundamentals should trade at equivalent multiples (Staňková, 2024). The practical challenge, amplified for private targets with unobservable comparables, is that no two firms are identical. This law implies an important question: along which dimensions must firms be comparable for their multiples to converge?

After testing the price-earnings method under alternative peer-selection rules based on industry, risk, and earnings growth, Alford (1992) finds industry membership to be the dominant criterion. This result became the standard justification for industry-based comparable sets, and it justifies the sector-matched comp-set construction of the analyst benchmark in this thesis. Bhojraj and Lee (2002) then challenge this rule by deriving a

warranted multiple from fundamentals. According to them, sharp improvements are observed when the selection of comparable firms is based on the fundamental determinants of valuation multiples rather than solely on industry classification. This finding correlates with the ML methods applied in this thesis: both estimate the cross-sectional drivers of multiple variation and price off fundamentals rather than assuming SIC co-membership implies comparability. Liu, Nissim, and Thomas (2002) studied a systematic ranking of value drivers by predictive accuracy. Forward earnings information comes first, historical earnings measures come second, followed by CF and book value of equity, with sales showing the worst performance. The bad performance of sales implies that EV/Revenue will be the harder multiple to predict from fundamentals in this analysis.

For these reasons, peer selection in practice is not a neutral exercise but is subject to systematic strategic distortion. De Franco et al. (2015) find in the first large-scale study of analyst peer choice that analysts on average tend to be influenced in the peer selection toward selecting companies with high valuations. This bias is often intensified with investment-banking incentives to negotiate higher takeover prices (Eaton et al., 2022). The core adjustment that distinguishes private-company valuation from listed companies' valuation is the application of a discount to the multiples of listed comparables. Because of illiquidity reasons since private shares cannot be freely traded, the multiples of listed peers overstate the price a private target should command, and a downward correction is required. The relationship applied in practice and reproduced by the analyst benchmark in this thesis takes the form:

$$V_{private} = V_{comparable} \times (1 - PCD)$$

where $V_{comparable}$ is the value implied by applying listed peers' multiples and PCD is the Private Company Discount. The theoretical basis for a strictly positive PCD is the liquidity premium in asset pricing. Amihud and Mendelson (1986) show that less liquid assets must offer higher expected returns to compensate investors for illiquidity. For a private firm, the discount is due to the absence of a public exit for existing owners as well as to the high due-diligence cost and information asymmetry an acquirer faces when buying a non-reporting entity. Koeplin et al. (2000) developed a theory on the PCD applied relative to the used multiple. When using earnings multiples to value, the PCD is on average between 20% and 30% and it is a bit lower when the valuation is based on book value multiples. Concerning companies traded using the revenue multiple, there is no significant difference between public and private companies, implying a null PCD. They also highlight that the discount does not only

capture marketability discount because of the difference in operating between the private firms and their listed comps. Silber (1991) also adds that the PCD for larger and more creditworthy companies are way smaller (around 14%) than the one applied on smaller companies reporting negative retained earnings (around 50%), implying that the discount varies by size, profitability, and deal type. Another article written by Rodríguez-Valencia et al. (2023) included a comparison between listed companies and unlisted Spanish SMEs, where no discount was observable. In fact, some ratios even benefitted from a premium over public firms, which confirms that the discount depends on the context and is not an universal constant. This heterogeneity is the structural weakness of any fixed-discount approach. Indeed, a single discount calibrated to minimise error on the bulk of the distribution must fail on sub-populations whose true discount differs.

Machine Learning in Finance and Private Company Valuation

In the past decade, recognising that many economically important relationships in empirical finance are non-linear lead to a methodological shift. Indeed, many of them involve interactions among numerous predictors, and are unstable over time, which are features that linear econometrics cannot handle properly. To face this, ML methods prioritise out-of-sample predictive accuracy over in-sample interpretability, use regularisation to control overfitting, and accommodate complex functional forms without pre-specifying them (Gu et al., 2020). This shift is particularly interesting for private companies' valuation because the link between fundamentals and the price is plausibly non-linear and interactive in ways a fixed-discount comparable method cannot represent.

The two ML tools employed in this thesis, RF and XGB exploit non-linearity through complementary mechanisms. Breiman (2001, p. 6) defines a RF as a *'classifier consisting of a collection of tree-structured classifiers {h(x, ϑk), k = 1, ...} where the {ϑk} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x'*. He proves that the generalisation error of the forest is bounded by a function of two quantities:

$$PE^* \leq \frac{\bar{\rho}\,(1 - s^2)}{s^2}$$

where s is the strength of individual trees and $\bar{\rho}$ is the correlation between them. As observable in the equation, accuracy improves when trees are individually strong (high s) and mutually decorrelated (low ρ̄), which is exactly what bootstrapping added with random feature selection achieves. Since the error converges to a limit (Breiman, 2001), ensemble size is not

itself a source of overfitting and can be fixed rather than tuned. What makes the algorithm suited to a sample of roughly 370 training transactions is separate: the variance reduction obtained by averaging decorrelated trees is largest precisely where individual trees are unstable, which is the small-sample case. XGB instead builds trees sequentially, each correcting the residuals of its predecessors, and controls complexity through an explicit regularised objective (Chen & Guestrin, 2016):

$$\mathcal{L}(\varphi) = \Sigma_i \, \ell(\hat{y}_i, y_i) + \Sigma_k \, \Omega(f_k)$$

where ℓ is the training loss between prediction $\hat{y}_i$ and target y_i and Ω penalises the complexity of each tree f_k. Chen and Guestrin (2016) explain that each additional term serves as regularisation and helps the learning of the previous one to avoid over-fitting. The differences between the two methods have an impact for small private-deal samples: RF reduces variance through parallel averaging, whereas XGB reduces bias through sequential correction, which raises its overfitting risk when the sample is small and noisy.

The landmark comparative study is from Gu et al. (2020), who applied the full ML methodology, including penalised linear models, regression trees, Random Forests, gradient-boosted trees, and neural networks to predict individual stock returns from 94 firm characteristics. As major result, they demonstrated the economic gains of using ML forecasts that sometimes even doubled the predictive performance compared to classical linear methods, finding trees and neural networks as the most efficient methods. Gu et al. (2020) show that a generalised linear model with non-linear transformations of individual predictors but without interactions with predictors will not beat the performance of a linear method. It is therefore crucial to enable interactions between predictors for the ML to add value to a basic linear analysis. To apply this study result on this thesis in other words: if acquisition multiples are additive-linear in fundamentals, OLS and ML will tie; if they are interactive, ML will win. This is exactly why the thesis retains OLS, not as a competitor, but as a diagnostic. Any ML gain reveals non-linearity between drivers. Amini et al. (2021) provide the closest methodological precedent, applying the same tree ensembles to corporate leverage and lifting the out-of-sample R^2 from 36% to 56% over LASSO and OLS. They further show that adding polynomial and interaction terms to the linear model does not close that gap, which is precisely the inference the OLS control is designed to license here.

In addition, two main studies apply this ML relative-valuation framework directly to multiples. Geertsema and Lu (2023), using gradient boosting on US public firms, report that ML can outperform traditional models when it comes to valuation accuracy. In fact, it cuts median

absolute errors (MedAE) to 0.29-0.32 against 0.40-0.59 for the best peer-group benchmarks and lifting out-of-sample R^2 from 0.38-0.48 to 0.76-0.78. They insist on the factor enabling this higher performance, which is the ability to model non-linear interactions between the variables. Concerning the variables that mainly drive the valuation, the importance feature analysis leads to metrics relative to profitability, growth, financial soundness, and efficiency being more impactful than industry classification to predict valuation. Staňková (2024) extends the approach to the exact multiples of this thesis, comparing gradient-boosted decision trees against GICS industry mean benchmarks on US listed firms from 1980 to 2021. The ML model outperforms on every metric and every multiple tested, with the median absolute percentage error (MedAPE) falling by an average of 24 percentage points across the four multiples examined. For EV/EBITDA specifically, MedAPE drops from 25% under the industry method to 6% under GBDT, an improvement of 19 percentage points, while the coefficient of determination rises from 11% to 68%.

The private-company setting this thesis is contributing to is an additional challenge since it makes it harder to find qualitative data inputs. Bitetto et al. (2025), who applied ML to evaluate the credit risk of unlisted firms, summarised the issues related to working with SMEs: the information asymmetries as well as the potential unreliability of the required disclosures, which is due to noisy accounting data that can enable accurate default prediction. Their solution is methodologically close to the one applied in this thesis: they add market-based signals from listed peers to compensate for the noise in private accounting data. This is an effective solution in their study since injecting peers' market base information demonstrated a significant increase in accuracy. This is the same intuition that motivates the augmented ML specification of this thesis, which supplies the models with the listed comp-set median and market-level variables, aiming to fight the limits of private-firm fundamentals.

Taken together, these studies establish that machine learning improves on linear and peer-group benchmarks across a range of financial prediction problems, yet none of them addresses the setting of this thesis. Table 1 compares them along the three dimensions that matter for the present exercise: whether the asset being priced already carries an observable market price, what is being predicted, and what the model is measured against. Two boundaries emerge. The valuation studies price listed firms, so their target has a market price of its own and their benchmarks are statistical or naive peer constructions rather than the rule practitioners apply; the one study conducted in the private setting predicts default rather than value. No existing work applies machine learning to predict transaction multiples for private mid-market targets, benchmarked against the illiquidity-adjusted comparables method that

actually determines these prices, and none supplies the model with the same comp-set information the benchmark itself uses. Closing that gap, on European Business Services transactions closed between 2015 and 2026, is the contribution of this thesis.

Table 1. Machine learning applied to valuation/private settings

Study	Setting	Target	Benchmark	Headline result	What it leaves open
Gu et al. (2020)	US listed equities	Monthly returns	OLS and linear factor models	Monthly out-of-sample R^2 0.33 - 0.40% for trees and neural nets	Returns, not valuation; no practitioner benchmark
Amini et al. (2021)	128,417 US listed firms	Target leverage	OLS and LASSO	Out-of-sample R^2 36% to 56% with random forests; interactions added to OLS do not close the gap	Leverage, not value; listed firms
Geertsema and Lu (2023)	US listed firms	EV/Sales, market-to-book, EV/assets	Academic peer models	MedAE 0.40 - 0.59 to 0.29 - 0.32; out-of-sample R^2 0.38 - 0.48 to 0.76 - 0.78	Listed firms; academic peer models, not the practitioner rule
Staňková (2024)	US listed firms (1980 - 2021)	EV/EBITDA, EV/EBIT, P/E, EV/Sales	GICS industry mean	EV/EBITDA MedAPE 25% to 6%, R^2 11% to 68%	Listed firms; naive benchmark, no illiquidity adjustment
Bitetto et al. (2025)	10,136 Italian unlisted MSMEs	Probability of default	ML on accounting data only	Peers' Merton PD lifts RF F1 from 88.4% to 95%	Private, but default rather than value
This thesis	**513 European private mid-market deals, 2015-2026**	**Transaction EV/EBITDA and EV/Revenue**	**Practitioner CCA: comp-set median with calibrated PCD, given to the models as a feature**		

The four listed-firm studies price assets for which a market price already exists, and benchmark against statistical or naive peer constructions rather than the method practitioners apply. Bitetto et al. (2025) work in the private setting but predict default rather than value. The present study occupies the intersection: it predicts transaction multiples in a market where no price exists until the deal closes, and benchmarks against the illiquidity-adjusted comparables rule itself, supplied to the models as a feature so that the comparison is information-matched.

Financial Fundamentals as Predictors of Private M&A Multiples

Corporate finance theory gives a clear prediction for what should drive cross-sectional variation in acquisition multiples. Rearranging the DCF identity for a going concern, the EV/EBITDA multiple can be expressed as a function of the expected growth rate g, the return on invested capital (ROIC), the cost of capital and the depreciation intensity of the business. Starting from the stable-growth firm value identity (Damodaran, 2006) and substituting the reinvestment relationship $g = I_r \times ROIC$ (Lukanima, 2023), the multiple can be written as:

$$\frac{EV}{EBITDA} = \frac{\left(1 - \frac{D\&A}{EBITDA}\right)(1-t)\left(1 - \frac{g}{ROIC}\right)}{WACC - g}$$

where t is the effective tax rate and $\frac{D\&A}{EBITDA}$ is the share of depreciation and amortisation in EBITDA, which captures the asset intensity of the business. The multiple falls as depreciation intensity and the cost of capital rise and increases with ROIC. Its response to growth is conditional: differentiating with respect to g shows that the multiple rises with growth only where ROIC exceeds WACC, since growth funded at a return below the cost of capital destroys value. Bhojraj and Lee (2002) formalise a similar logic for EV/Sales, arguing peer selection should be based on the expected profitability, growth, and the WACC. The variable set of this thesis, containing metrics on size, revenue and EBITDA growth, EBITDA and EBIT margins, return on assets (ROA), EBITDA volatility, and equity ratio, matches these theoretical drivers. Liu et al. (2002) confirm empirically that earnings-based drivers dominate sales in explaining listed multiples. Accordingly, the same ranking implies that EV/Revenue is the noisier target.

These theoretical learnings are accurate for listed companies but far from obvious when it comes to predicting private companies' valuation. This is due to three structural features of private M&A pricing. Firstly, the input data are noisier for private firms than for the listed firms on which the theoretical evidence was built. As Bitetto et al. (2025) emphasise, private disclosures are not always reliable and contain noisy data, so even a genuine underlying relationship between fundamentals and value is influenced by measurement error in the regressors themselves. Secondly, M&A prices are negotiated outcomes in which the acquirer's synergies, strategic rationale, and competitive position impact willingness to pay in ways that are not directly observable on the target's balance sheet. Greene (2017) studied private-firm acquisitions and demonstrated that the financial condition of the target influences its valuation multiple. Indeed, financially constrained targets struggle more in negotiations. This result highlights that many other factors, that are not standard fundamentals, have an impact on the realised multiple, going far beyond reported financials. Thirdly, private-deal samples are heterogeneous even within a sector. It exists a various number of conditions in which a deal takes place: family successions, distressed disposals, PE secondary transfers, or even strategic add-ons, that price on structurally different logics. It explains why a single linear model treating them as exchangeable will detect little stable signal and therefore struggle to be statistically significant. For EV/Revenue the same reasoning predicts weaker attenuation. Because the revenue multiple is by construction proportional to the profit margin times the growth factor, cross-sectional variation in growth should explain a real if small share of

EV/Revenue variation even in a small and noisy private sample, whereas the EBITDA multiple, which already absorbs cost structure in its denominator, has no comparably direct link to any single observable variable. Whether either prediction holds is an empirical question and is addressed in the results chapter.

Another well-documented variable with an explanatory power in the multiple prediction is the PE acquisition premium. Bargeron et al. (2008) show that acquirer type systematically shapes premiums with private and PE acquirers paying lower premiums than public acquirers, and Gorbenko and Malenko (2014) model takeover auctions with mixed bidder pools, showing that competition among financial bidders who rely on leverage and financial engineering can push equilibrium prices above any single bidder's standalone valuation. In many sectors, including Business Services which is an asset-light, recurring-revenue, and heavily represented in PE portfolios sector, targets are frequently run through auction processes to maximise competitive tension.

The combined theory points to a coherent conclusion that frames the entire thesis. ML models can extract the non-linear structure of the observable signal, which is stronger on the EV/Revenue once market context is added, but the ceiling on any fundamentals-based prediction is set by the information available, not by the sophistication of the model.

This literature review establishes a coherent theoretical basis for the valuation of private mid-market targets. On one side, the absence of an observable price incentivises the analyst to complete intrinsic DCF with an added relative valuation. However, the CCA rests on a PCD that theory and evidence show to be highly dependent on the context and therefore heterogeneous across industry, size, profitability, and deal type, giving any fixed-discount implementation a structural ceiling. On the other side, the relationship between observable private-firm fundamentals and M&A transaction multiples is weak in the linear domain and dominated by unobservable, deal-specific factors that are amplified in the private setting. ML offers a way to capture the non-linearities and interactions that linear comparable and regression methods cannot and has been used to sharpen relative valuation. It is important to notice that it mainly has been used for listed firms with observable market multiples. Two conclusions follow for the design of this thesis. First, there is a clear and unaddressed gap: no prior study applies ML to predict transaction multiples for private mid-market M&A targets, benchmarked against the illiquidity-adjusted comparable method that practitioners actually use. As developed in Table 1, existing ML valuation work (Geertsema & Lu, 2023; Staňková, 2024) prices listed firms against observable market multiples, while the private-firm ML

literature (Bitetto et al., 2025) addresses default rather than valuation. Second, the literature does not conclude on which approach should prevail. On one hand, the comparable method is anchored to contemporaneous market prices, a real-time signal that no historical fundamental can reconstruct, and theory suggests this anchoring should be most valuable for the typical deal that trades respecting the norm of its sector. On the other hand, the flexibility of ML should matter precisely where a fixed discount is weakest, including on atypical, premium-positioned, or tail transactions that a single sector median cannot represent, while its capacity to exploit market-context variables may partly compensate the information disadvantage inherent in private data. Whether that flexibility is sufficient to rival a market-anchored benchmark built on comparable information, and where along the distribution of deals each method holds its edge, cannot be resolved from theory alone. It is an empirical question, and it is the one the analysis that follows is designed to answer.

Methodology

Research Design

This study examines whether ML models can predict mid-market M&A transaction multiples at a level of accuracy comparable to the comparables-based approach used by practitioners. The research question is empirical and requires a strict comparison between the two approaches on an identical transactions database. This section details the design choices, including the number of methods compared, the choice of benchmarks, the split strategy, and the evaluation metrics. The target variables are EV/EBITDA and EV/Revenue, that are multiples mostly used in mid-market M&A advisory documentation (Shaffer, 2024). The EV/EBIT multiple was excluded due to weak data availability. It could be computed for 344 of the 513 retained transactions, against 493 for EV/EBITDA and 494 for EV/Revenue. This lower EBIT availability is explainable since it requires depreciation and amortisation to be disclosed separately, and it is precisely the smaller firms filing abbreviated accounts that omit that line.

To answer this research question, the analysis is based on the comparison of four methods. The first one is the analyst method benchmark, which reproduces the standard M&A practice by applying a calibrated illiquidity discount to the median multiple of a set of listed comparable companies in the same sector. The second one is a linear regression, which serves not as a primary competitor but as a statistical control. Indeed, if ML models outperform linear regression by a meaningful margin, this constitutes evidence that non-linear relationships exist

in the data and that the ML models are capturing them, but if they do not, the claim of non-linearity cannot be supported regardless of how the ML models compare to the analyst benchmark. The third and fourth methods are RF and XGB models, each tested in two variants: one using only company-level financial fundamentals (called the “pure ML”) and one also incorporating market-context variables including the sector comp-set median, the Stoxx Europe 600 index level, and the 10-year government bond yield at the transaction closing date (called the "augmented" ML). The augmented versus pure distinction is methodologically important. Indeed, if augmented models substantially outperform pure models, this tells us that market information adds signal beyond what fundamentals alone contain. If they do not, it suggests that the relationship between fundamentals and multiples is the primary constraint on predictive accuracy, not market information availability.

The evaluation framework prioritises the Median Absolute Error (MedAE) as the primary metric, with Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as metrics reported for completeness. This decision is due to a structural feature of the dataset: the EV/EBITDA multiple distribution in mid-market M&A contains a subset of transactions where the EBITDA was near zero at the time of closing, mechanically inducing extreme ratios with no direct relationship to the economic variables of the transaction. These transactions generate prediction errors of above 100x regardless of the method applied, and they make any mean-based metric a poor summary of typical prediction quality. The MedAE, being insensitive to distributional extremes by construction, provides a reliable measure of performance on the broad population of typical mid-market deals. All three metrics are reported in the appendices, and a sensitivity analysis explicitly quantifying the impact of the extreme transactions on each metric is conducted for every method, ensuring that the choice of primary metric does not suppress information about the tails of the error distribution.

Sample Construction

The sample is constructed from two databases, containing transaction data and target company-level financials. All transaction data are sourced from Capital IQ, including deal identifiers, closing dates, transaction EV/EBITDA and EV/Revenue multiples, buyer type, and target country for transactions across the Business Services sector in Europe since begin of 2015. Company-level financial data that include income statement and balance sheet variables are sourced from Bureau van Dijk Orbis, a database that aggregates financial statements from national registries across Europe. Those financials were then used to compute several derived ratios for the years preceding each transaction closing date. The combination of these two

sources produced an initial dataset of 542 transactions for which both a transaction multiple and company financial data were available.

A filter concerning sectors was then applied, reducing this initial set to the 513 transactions used throughout the study (see Table 3). This sector-based filter is essential to ensure comparability between methods. In fact, the comparables-based analyst method requires a set of listed comparable companies operating in the same sector whose multiples serve as the pricing anchor, and this set must be sufficient to produce a statistically robust median. To ensure this sufficient set of listed comparables, all sectors in the initial dataset were tested against a minimum threshold of three valid comparables per sector per year. After this selection, seven Business Services sub-sectors met the requirement consistently across the 2015-2026 period covered by the sample: Diversified Support Services, Research and Consulting Services, Environmental and Facilities Services, Human Resource and Employment Services, Office Services and Supplies, Security and Alarm Services, and Commercial Printing (see Table 4).

Comparable Company Set Construction

The analyst method benchmark requires a set of listed comparable companies for each transaction, matched by sector and closing year (CY), to ensure that it reflects the market pricing environment at the time of the private transaction, not an average across periods with different risk aversion and financing conditions. To avoid extreme values, listed company multiples were cleaned using the interquartile range (IQR) method within each sector per year cell before computing summary statistics. The bounds applied use a k equal to 1.5, a standard non-parametric outlier detection rule that is robust to non-normal distributions and was applied uniformly across all 84 sector per year cells. Removing these from the comp-set prevents a small number of distressed listed companies from over influencing the median used as the pricing anchor. A minimum of three valid comparables per cell after outlier removal was required to use a cell as a pricing anchor, excluding below standard sectors. The median was preferred here for its robustness to the extremes, though it should be noted that Liu et al. (2002) find valuation performance improves when multiples are aggregated using the harmonic mean rather than the simple mean or median, a refinement that a larger comp-set would have permitted. The augmented ML and regression specifications use the same sector per year comp-set median as a model feature, ensuring that any performance difference between the analyst benchmark and the augmented models reflects the models' capacity to exploit market information, not a difference in the quality of the comp-set information itself.

Variable Selection and Data Preprocessing

The Orbis database provides a large number of financial variables per company across multiple years. Rather than entering all available variables into the model and allowing algorithmic feature selection to manage the resulting noise, a selection was conducted before any model fitting. The variable selection is grounded in corporate finance theory, keeping variables that drive acquisition multiples and removing mechanically redundant variables and variables that are less relevant in this sector context.

Ten variables were retained across four categories, along with a deal type variable (see Table 5). Firstly, the size block includes log-transformed total assets (CY-1) and log-transformed employee count (CY-1). Log transformation, that produces a more stable and interpretable regression coefficient, is applied because the underlying distributions are heavily right skewed. Indeed, a company ten times larger is not ten times more valuable in a linear sense. The two size measures are retained alongside each other because they capture complementary dimensions: total assets reflect financial scale, while employee count reflects operational scale. The employee count, and thereby operational scale, is particularly informative in this sector of Business Services which is asset-light. Secondly, the growth block, which is particularly important since, as explained through the Gordon-Shapiro present value framework, the multiple is proportional to $g/(r-g)$ making growth the primary theoretical driver of EV/Revenue multiples and a secondary driver of EV/EBITDA multiples. This block includes the Compound Annual Growth Rate (CAGR) of operating revenue, the CAGR of EBITDA, and a revenue growth acceleration measure computed as the difference between the one-year revenue growth rate in CY-2 and the preceding revenue growth rate. This last acceleration metric term captures changes in growth trajectory that a single CAGR figure would miss. The third block concerns profitability margins, reflecting current earnings quality, and includes EBITDA margin and EBIT margin for CY-1. These two measures are deliberately retained alongside each other since their difference, depreciation and amortisation as a share of revenue, carries information about asset intensity. The fourth and last block represents risk and capital structure including log-transformed EBITDA volatility, ROA, and equity ratio. On one hand, EBITDA volatility is a measure of earnings instability that captures execution risk and earnings predictability, which are primary components of the discount rate applied to an acquisition. On the other hand, ROA captures asset efficiency and the equity ratio reflects the target's current financial leverage, which affects both the acquirer's financing capacity and the implied valuation premium. A non-financial binary indicator, specifying if the deal was PE-backed or not, is also included in the feature set. This variable is a structural deal characteristic

since PE-backed acquisitions versus strategic acquisitions often price differently, driven by the auction dynamics, financial engineering capacity, potential synergies and portfolio management logic specific to PE buyers.

The CAGR variables required a specific construction decision because the preferred three-year horizon (CY-3 to CY-1) is available for only approximately half the sample, due to the limited financial history of smaller mid-market companies in Orbis. A fallback hierarchy was therefore applied: for each transaction, the three-year CAGR is used when available, otherwise a two-year rate (CY-2 to CY-1) is computed, and so on. A separate indicator variable encodes which horizon was used for each observation (coded 3, 2, 1, or 0 for imputed), providing the ML models with information about the reliability of each growth estimate without discarding any observation. Critically, this horizon indicator was excluded from the regression specification because it was near-perfectly collinear with the missingness indicators for other variables (a variance inflation factor (VIF), above 100 in the initial specification), but it is available to the ML models operating with no such constraint (ML-B). Those were also run using the exact same set as for the linear regression, meaning excluding indicator variables, to assess the efficiency impact of those variables (ML-A).

The distinction between ML-A and ML-B is methodologically interesting (see Table 6). ML-A uses the same feature set as the final regression specification, enabling a clean comparison between OLS and ML on identical inputs: any performance difference must reflect modelling capacity (non-linearity capture) rather than information asymmetry. ML-B adds the variables that cannot enter the regression because of their structural collinearity, which are the two CAGR horizon indicators and nine of the eleven missingness dummies. Two dummies are left out because they carry no independent information: the equity ratio missing flag and the sponsor missing flag. Tree-based models have no multicollinearity constraint, since a RF or XGB tree can split on a missingness flag independently of other variables without any mathematical issue. Including these variables in ML-B tests whether the pattern of missing data, which tends to be correlated with company size, age, and reporting quality, contains predictive information about transaction multiples beyond the imputed values of the underlying financial variables.

Concerning data preprocessing, four variables required transformation before any model was trained, based on distributional inspection of the training set (see Table 7). The necessity of each transformation was established from the training set only, for then applying the same unchanged parameters to all test sets and augmented datasets. This sequencing is essential since computing transformation parameters from the test set would allow test-set

distributional information to influence the preprocessing pipeline, inducing data leakage that would inflate performance.

EBITDA volatility was log-transformed. In its raw form, the variable demonstrated extreme right skew on the training set, with a mean-to-median ratio of approximately 6.5 (mean 4,566; median 699; maximum 188,911). In addition, the economic interpretation also favours the log scale since a 1% increase in EBITDA volatility is more meaningful than a one-unit absolute increase on a variable whose range spans five orders of magnitude. In the case of EBITDA volatility, the log-transformation was sufficient to produce a well-behaved distribution (training set range: 2.06 to 12.15; mean 6.77; median 6.55) and no further winsorisation was therefore required.

The revenue growth acceleration and CAGR EBITDA were winsorised at the 1st and 99th percentiles of the EBITDA training set. Both variables contained values that are economically impossible and arise from near-zero denominators in the growth rate calculation: the revenue growth acceleration reached a minimum of -572, and CAGR EBITDA reached a maximum of 47.8. Log-transformation was not applicable to these variables because both contain negative values that cannot be handled by the logarithm. Winsorisation at the 1st/99th percentiles produced caps of [-6.63, 1.92] for revenue growth acceleration and [-17.72, 3.02] for CAGR EBITDA, removing the most extreme artifacts while retaining the full range of economically plausible values.

The equity ratio was also winsorised at the 1st and 99th percentiles following discovery of a single extreme observation. Nine training-set transactions had negative equity ratios, which is economically possible for distressed or highly leveraged companies, and one of these reached −5.02, three times more extreme than the next most negative case. The rule produced caps of [−0.527, 0.981]. At the lower end it clipped four observations, the −5.02 case together with three ratios between −1.18 and −1.66, while the five least extreme negative-equity transactions were left untouched. At the upper end it brought down four firms reporting a ratio of exactly 1.00, corresponding to companies with no recorded liabilities.

Concerning the missing values, they were imputed to avoid discarding partially observed transactions. With ten numerical variables each missing between 23% and 50% of observations, a complete-case analysis dropping any row with a missing value on any variable would eliminate most of the training set. The imputation method used is sector-median imputation with IQR-trimming. In fact, for each variable, the imputed value is the median computed from all non-missing observations in the same sector, after removing outliers from

that sector's distribution using the standard IQR rule (k=1.5). The sector dimension is included because financial fundamentals differ structurally across sectors. Indeed, imputing the EBITDA margin for a missing Research and Consulting transaction using a global median that includes Commercial Printing companies would introduce a systematic bias, since the two sectors substantially have different typical margins. The IQR-trimming step was added after inspection of the smallest sectors revealed that single extreme observations within a sector could substantially shift the sector median. For instance, in Security and Alarm Services (19 training observations), EBITDA volatility had a raw sector median of 1,062 but a trimmed median of 582 after removing two outlier observations at 43,777 and 135,070. The raw median would have been an unreliable benchmark driven by two atypical companies, and the trimmed median is more representative of the sector's actual volatility profile. A missing-value indicator dummy was created for each imputed variable, providing the ML models with the information that a given observation was imputed and allowing them to adjust their predictions accordingly. These dummies were included in the ML-B feature set but excluded from the regression specification and from ML-A.

Train/Test Construction

A held-out test set was constructed before any model training, or variables selection (see Table 8). The purpose of holding out a fixed test set is to simulate genuine out-of-sample prediction since the test set transactions are unknown to every model during all training and parameter selection steps, and each method is evaluated on them exactly once. Violating this principle invalidates the entire comparative exercise, since the reported performance would then be biased and reflect the models' ability to fit known data rather than to reflect its application on unknown transactions. The split between train and test was applied identically to all three method databases (analyst, regression, ML). Within each multiple, every method is therefore evaluated on exactly the same held-out transactions, so that no part of the difference between methods can be attributed to a difference in the sample on which they were scored. The EV/EBITDA and EV/Revenue test sets are not identical to one another, since availability of the two denominators differs across transactions (see Table 4).

The split was performed at the entity level rather than at the transaction row level since 31 of the 460 unique entities in the EV/EBITDA sample have more than one transaction. A row-level split would place some of the same company's transactions in the training set and others in the test set. Because the ML models learn patterns at the company level, a model that has seen one transaction from a given company in training might implicitly recognise that company

when predicting its second transaction in the test set, inflating the models' accuracy. Grouping the split by entity ensures that all transactions of a given company land on the same side of the split, eliminating this bias risk.

Sector stratification was applied during the split to ensure that the seven sectors are proportionally represented in both training and test sets. This is particularly important given the wide variation in sector sample sizes, going from 22 transactions in Commercial Printing to 140 in Research and Consulting, which means that a purely random entity-level split could concentrate a small sector almost entirely in training or test, making the per-sector error analysis meaningless. Stratification ensures representativeness without imposing an exact proportionality constraint that would conflict with the entity-grouping requirement.

Analyst Method Construction and Calibration

The analyst method benchmark is defined as:

$$Multiple = Median * (1 - d)$$

where $Median$ is the IQR-trimmed median multiple of listed comparable companies in the same sector and CY as the target transaction, and d is a discount parameter estimated on the training set. This formula reflects the logic of standard M&A comparables valuation divided in two steps. The first step defines the valuation to what listed peers trade at in the same market environment. The second step consists of applying a downward adjustment to reflect the structural price difference between public and private markets, primarily driven by the illiquidity premium expected by investors in private companies.

The discount parameter d was calibrated on the training set by exhaustive search over 601 tested values from 0% to 60% in steps of 0.1%, minimising the prediction error on the transactions in the training set. Two error criteria were evaluated: MAE and MedAE. The MAE criterion produced near-zero optimal discounts (0.9% for EV/EBITDA, 0.0% for EV/Revenue). This finding is an artifact of the same distributional issue that motivates using MedAE as the primary evaluation metric throughout this study: transactions with near-zero EBITDA at closing produce prediction errors of above 100x regardless of the discount applied, and their overwhelming contribution to the mean pulls the optimal discount towards the value that best accommodates these outliers rather than the typical transaction. The MedAE criterion is by construction insensitive to these extremes and leads to discounts of 21.5% for EV/EBITDA and 3.5% for EV/Revenue. The EBITDA value closely matches the approximately typical 20% PCD documented in the M&A literature and applied by practitioners, providing independent

validation of the calibration approach. Indeed, it is coherent with the study of Koeplin et al. (2000) predicting a discount between 20% and 30% for earnings-based multiples and a near-zero discount when the valuation is based on the revenue multiple. These MedAE-calibrated discounts are used as the official analyst method parameters throughout the study.

Linear Regression

The linear regression is not a primary valuation method in this study but a statistical control. Its role is to establish the linear prediction baseline against which ML models are compared. It demonstrates that if RF and XGB outperform OLS by a meaningful margin on the same feature set and the same test transactions, this constitutes evidence that non-linear interactions exist in the data, captured by the ML models. Without this control, it would be impossible to attribute any ML advantage over the analyst benchmark to the models' capacity for non-linearity rather than to other factors such as variable selection or regularisation properties. The initial feature set for the regression was identical to ML-A, supplemented by the missingness dummies and horizon indicators for ML-B. This broader starting point was subjected to four sequential diagnostics, each identifying and resolving a specific statistical problem. The first two reduced the feature set from twenty-four variables to eleven (see Table 14), and the third and fourth validated the resulting specification.

The first issue was a perfect singularity: the equity ratio missing dummy was found to correlate at exactly 1.0 with the total asset missing dummy. The cause is structural since both variables originate from the same financial statements and that any company lacking balance sheet data lacks both total assets and equity ratio simultaneously, making one column a perfect linear copy of the other. This variable was therefore removed from the feature set.

The second issue was severe multicollinearity in the horizon and missingness variables. The two CAGR horizon indicators (CAGR revenue and CAGR EBITDA) produced VIF values of 145 and 27 respectively, which are far above the threshold of 10 that indicates severe collinearity. The VIF for a variable j is defined as:

$$\frac{1}{(1 - R^2 j)}$$

where $R^2 j$ is the coefficient of determination from regressing j on all other predictors. A VIF of 145 means that 99.3% of the variance of the horizon variable of CAGR revenue is explained by the other predictors, making its coefficient estimate unstable and meaningless. The source of this extreme VIF is structural: the horizon indicator reflects approximatively the same

information as the combination of missingness flags for CAGR and related variables, since a horizon of 0 (imputed) and a missingness flag of 1 are the same signal expressed in different ways. Removing both horizon indicators resolved the extreme VIF but left the missingness dummies with VIFs running from 1.3 to 33.7, five of them above the conventional threshold of 10. This arises from the same structural reason: companies that are missing EBITDA margin data tend simultaneously to miss EBIT margin, ROA, and CAGR data, because the source is the same for all financial statements filing. Therefore, collinearity between the missingness dummies is not collinearity between distinct signals but redundancy within a set of variables encoding the same underlying event, which is company not filing complete financial statements. Dropping all ten missingness dummies reduced all remaining VIFs below 4, yielding a clean specification. The final regression feature set is therefore identical to ML-A: eleven variables with no missingness dummies and no horizon indicators.

The third diagnostic tested for heteroscedasticity using the Breusch-Pagan test. Heteroscedasticity, representing non-constant residual variance across observations, does not bias OLS coefficient estimates but invalidates the standard errors (SE), making t-statistics and p-values unreliable. The EV/EBITDA model returned a p-value of 0.902, providing no evidence of heteroscedasticity and confirming that OLS SE are valid for inference in that specification. The EV/Revenue model returned a borderline p-value of 0.051. For further confirmation, heteroscedasticity-consistent (HC3) robust SE were computed for the revenue model, which correct the SE without changing coefficient estimates or prediction accuracy.

The fourth diagnostic identified influential observations using Cook's distance with a threshold of 4/n, where n is the number of training observations. Cook's distance measures the aggregate influence of each observation on all coefficient estimates simultaneously. The observations above the threshold exert disproportionate leverage on the model. Thirteen influential observations were identified in the EBITDA training set. These were not removed since those influential observations are not errors. Removing them would bias the model toward performing well on average-multiple transactions while degrading its performance on precisely the high-leverage cases that generate the most valuation uncertainty in practice. Their identification is reported for transparency and to confirm that the same structural feature (near-zero EBITDA at closing) that dominates the test-set error metrics also affects the training-side coefficient estimates.

The augmented regression specification adds three market-context variables to the eleven-variable base: the sector comp-set median multiple, the Stoxx Europe 600 closing level at the

transaction date, and the 10-year euro area government bond yield at the transaction month (see Table 6). All VIFs in the augmented specification remain below 4, confirming that adding market-context variables does not reintroduce the collinearity problems that the feature selection process resolved.

Ridge regression was also run as a robustness check on the pure specification for both targets. Ridge adds a penalty on the sum of squared coefficients to the OLS objective, which shrinks all coefficients toward zero and can improve prediction in the presence of moderate multicollinearity. The regularisation parameter λ was selected by five-fold cross-validation (CV) on the training set, using the glmnet package with α=0. A comparison of OLS and Ridge coefficient estimates was conducted to assess whether the OLS specification is stable: if Ridge substantially changes the sign or magnitude of any coefficient, this indicates that OLS was relying on collinear structure that Ridge resolves by shrinking. Sign changes between OLS and Ridge are only informative for this purpose when they occur on variables with statistically distinct coefficients in OLS; sign changes on variables whose OLS coefficients are already indistinguishable from zero represent noise shrinkage around zero rather than genuine instability. The Ridge results serve as a methodological verification of the OLS specification rather than as an alternative primary model.

Machine Learning Models

Two tree-based algorithms were selected: RF and XGB. RF builds each tree independently on a bootstrap sample of training data, using a random subset of features at each split, and averages predictions across all trees. This parallel construction provides strong implicit regularisation and makes the algorithm relatively insensitive to hyperparameter choices, which is an advantage when training on approximately 370 observations. XGB builds trees sequentially, each correcting the prediction errors of all previous trees. This sequential boosting is more powerful on large datasets where it can learn subtle patterns through many correction steps. However, it is also more prone to overfitting on smaller samples because early trees can set the model on a path that later trees cannot fully correct. Including both algorithms allows the results to distinguish between findings that are robust across the class of ensemble tree models and findings that are specific to one algorithm's bias.

Hyperparameter grids were defined before any model was fitted, so that the search space could not be adjusted in response to results. Each grid brackets the default settings of its algorithm, with the upper bounds deliberately restrained: on a training set of roughly 370 observations the binding risk is overfitting rather than underfitting, so deep trees and long

boosting sequences were excluded from the outset. Table 2 reports the parameters varied and the values tested, together with those held fixed.

Table 2. Hyperparameter grids

Parameter	Values tested	What it controls
Random Forest — 3 parameters tuned, 18 combinations		
mtry	4, 6, 8	Number of features drawn at random as candidates at each split. Spans the ranger default of √p, which runs from 3.3 to 5.0 across the four feature sets (Breiman, 2001)
min.node.size	5, 10, 20	Minimum number of observations a terminal node must contain. Larger values force shallower trees and constitute the main regularisation lever of the algorithm
max.depth	NULL, 4	Maximum number of splits between root and leaf. NULL leaves growth unconstrained, so that it stops only once min.node.size binds
num.trees	500 (fixed)	Size of the ensemble. Set high enough for predictions to stabilise and therefore conventionally fixed rather than tuned
XGBoost — 4 parameters tuned, 16 combinations		
nrounds	100, 200	Number of boosting iterations, each adding one tree to the sequence
max_depth	2, 3	Maximum tree depth. Boosting compounds depth across iterations, so the useful range sits well below that of RF
eta	0.05, 0.10	Learning rate, scaling the contribution of each new tree. Lower values correct the residual more cautiously and require more iterations to converge
min_child_weight	5, 10	Minimum sum of instance weights required in a child node. Under the squared-error objective used here every observation carries a weight of one, so the parameter reduces to a minimum leaf size, the direct counterpart of min.node.size in RF
subsample	0.8 (fixed)	Fraction of training rows drawn for each boosting iteration
colsample_bytree	0.8 (fixed)	Fraction of features made available to each tree

Combinations are the product of the tested values: 3 × 3 × 2 = 18 for RF and 2 × 2 × 2 × 2 = 16 for XGB. Each combination was evaluated by five-fold entity-grouped cross-validation on the training set, with mean absolute error as the selection criterion. Parameters marked as fixed were not varied.

The fold construction reused the same entity-level assignment logic as the train/test split, with fold numbers assigned to entities using a fixed random seed (42). The CV criterion was MAE across the five held-out folds. MAE rather than RMSE was used as the CV criterion because RMSE's sensitivity to large errors would steer hyperparameter selection toward parameters that minimise the impact of extreme-multiple transactions, rather than those that best predict the typical deal. MedAE was not used either, since with roughly 74 observations per held-out fold the median is determined by a single transaction, making the criterion jump erratically between neighbouring parameter values rather than providing the stable surface that selection requires.

For each of the eight pure specifications (two algorithms × two feature sets × two targets) and the eight augmented equivalents, giving sixteen models in total, the parameter combination with the lowest mean CV MAE was identified and the final model was trained on the complete training set using those parameters. Feature importance was extracted after final model training using two methods: permutation importance for RF (the increase in MAE when each variable's values are randomly permuted, averaged across all trees, with negative values indicating that a variable adds noise rather than signal) and gain-based importance for XGB (the fraction of total information gain from all splits attributable to each variable across all trees and all rounds). These two importance measures are complementary: permutation importance measures the impact on prediction accuracy when a variable is made uninformative, while gain-based importance measures the contribution to the model's learning process. When the two measures rank variables similarly, the importance finding is robust. However, when they diverge, this reveals that the two algorithms are exploiting different aspects of the data, and the divergence is then read as evidence about how securely each variable's contribution is established rather than as a structural property of the multiple being predicted.

The software implementation uses the *ranger* package for RF and the *xgboost* package for XGB, both in R. All random seeds are set to 42 throughout, covering fold construction, model training, and CV analysis. The complete R code is provided in the appendices as an annotated R Markdown document.

Results and Analysis

The Analyst Method

The calibration step produces a methodological finding that sets the interpretive frame for this analysis. Minimising the MAE on the training set converges to a near-zero discount, which would suggest no illiquidity adjustment is useful for private companies. This result illustrates the distortion mechanism due to extreme observations in the methodology. The MedAE criterion, insensitive to this distortion by construction, leads to a calibrated discount of 21.5% for EV/EBITDA and 3.5% for EV/Revenue (see Table 9 and Figure 1). Those results, mainly for the discount applied on the EBITDA multiple, match with the approximately 20% PCD most frequently used. These discounts were applied unchanged to the test set (see Table 10).

On the 124 test transactions, the analyst method reaches a MAE of 12.37x and a MedAE of 4.07x for EV/EBITDA (see Table 11). This gap is not a statistical anomaly, but is mainly due to the seven extreme transactions (referring to transactions with EV/EBITDA above 50x) since removing them reduces the MAE from 12.37x to 5.70x. It means that removing only 7 of the 124 data points induces a 54% reduction of the MAE (see Figure 2). The RMSE makes the asymmetry even more visible because squaring the prediction errors amplifies the influence of the extreme observations disproportionately (36.87x on the full set and 9.00x excluding the extremes).

Considering those metrics without this decomposition fundamentally misrepresents the performance of the analyst method. To have a better understanding of the analyst method's performance, the test set of 124 transactions was divided by EV/EBITDA multiple ranges: typical (0-15x), elevated (15-50x) and extreme (>50x). For the typical section counting 97 transactions, this method achieves a MAE of 3.33x with an actual multiple of 7.62x. Concerning the 20 elevated observations, the observed performance was worse with a MAE of 17.19x. Lastly, the 7 extreme observations show the worst performance with a MAE of 123.96x (see Table 12). Those results are explainable since the analyst method treats every deal identically, without considering premium above the sector median when needed. Therefore, this method works well for the majority of deals that trade near the sector median, but fails for the minority that do not. In general, this method predicts the typical deal to within approximately 4x, and it fails completely on a structurally identifiable subset that no discount calibration could address because of the extreme economic parameters. This confirms the PCD is not a constant and highly depends on the context. Indeed, the analyst method, with its uniform 21.5% EV/EBITDA discount, achieves a MAE of 3.33x on typical transactions (EV/EBITDA 0-15x) but degrades to 17.19x on elevated transactions (EV/EBITDA 15-50x), where premium-positioning dynamics that no fixed discount can encode take over (see Table 12). These known limitations, including mechanical failure on the tails, inability to capture deal-specific premiums, and sensitivity to peer selection, motivate the ML comparison at the heart of this thesis. For EV/Revenue, the same structural problem does not exist because the revenue denominator does not reach near-zero values in this sample.

To deepen the analysis, sector-level distinction was applied to assess performance differences between sub-sectors (see Table 13). On one hand, the best performing sector according to EV/EBITDA accuracy is the Security and Alarm Services (4.16x) and Research and Consulting Services (5.00x), though the former rests on only five test transactions and should be read as indicative. Research and Consulting, with thirty-five test observations, is the more reliable

result and is consistent with this sector having both the deepest comp-set and the most homogeneous business models among the seven retained. On the other hand, Office Services and Supplies records the highest error at 45.54x, a figure driven by two extreme transactions closing at 255x and 66x. Excluding them, the remaining six transactions produce a MAE of 9.33x, in line with the rest of the sample. Setting that artifact aside, Human Resource and Employment Services is the weakest performer at 20.41x, and it is also the worst sector on EV/Revenue at 4.60x, the only one to rank poorly on both multiples. This is plausibly a heterogeneity problem rather than a comp-set depth problem, since the sector groups staffing agencies, recruitment consultancies and HR technology providers whose margin structures and growth profiles differ enough that a single sector median anchors any individual transaction poorly. The rankings are otherwise not stable across the two multiples, which is itself informative. Environmental and Facilities Services produces the lowest revenue error of any sector at 0.96x while sitting only third on EV/EBITDA, and Diversified Support Services is the second weakest on EBITDA once Office Services is set aside, at 15.04x, yet second best on revenue at 1.60x. A sector in which the comp-set anchors revenue well is not necessarily one in which it anchors earnings well, which is a further expression of the two-multiple asymmetry developed in the literature review.

To conclude, the key takeaways of the analyst method firstly concern the median-based calibration that recovers a 21.5% discount, closely matching the convention documented in the literature, where mean-based calibration collapses to near zero under the weight of seven extreme transactions. Secondly, the benchmark predicts the typical transaction to within 3.33x, with its aggregate MAE of 12.37x which is a poor summary since 6% of the test set generates 57% of total absolute error. Most importantly, accuracy is not uniform: the method is strongest near the sector norm and systematically underestimates the 15-50x range.

Linear Regression

As previously explained, the role of the linear regression is to determine the linear signals existing in the financial fundamentals before applying the ML models. The training set for EV/EBITDA was used to define the model's statistical performance, leading to an adjusted R^2 of 0.018 and a p-value of 0.095, which represent a model that is not significant at conventional levels. Another regression was run removing the 17 extreme transactions of the train set to verify their influence on the model's statistical accuracy. This new model without extremes remains flat with an adjusted R^2 of 0.014 and a p-value of 0.150. These models containing over 300 transactions for eleven variables have a comfortable observations-variables ratio,

meaning that the absence of signal reflects the genuine structure of the data. Indeed, the eleven financial fundamentals of private firms as a group carry no detectable linear relationship with EV/EBITDA multiples in this sample.

However, two variables are statistically significant (see Table 15). Firstly, with a p-value of 0.002 and a coefficient of 8.7: the PE-backed dummy, a finding that is consistent with the literature on the PE acquisition premium. Indeed, the +8.7x coefficient captures a premium that is structural, arising from deal process and buyer type, rather than target's fundamental financials. Secondly, log of total assets carries a coefficient of −2.57 with a p-value of 0.007, indicating that targets with a larger asset base trade at lower EBITDA multiples. Treating the asset base as a proxy for target size, the negative coefficient is notable: it runs against both the intuition that scale commands a premium and the practitioner evidence, since the Vlerick M&A Monitor records European EV/EBITDA multiples rising with deal size, from 5.0 for deals below €5 million to 10.5 for deals above €100 million (Luypaert et al., 2025). The two are not strictly comparable, since the Monitor reports an unconditional relationship whereas the coefficient here is estimated holding employees, margins, growth and deal type constant, but the divergence in sign is worth noting. A plausible reading is therefore that in an asset-light sector, total assets proxies less for scale than for asset intensity. Since margins and ROA are already controlled for, a larger balance sheet indicates a firm requiring more capital to produce the same returns, which within this asset-light sector family is associated with lower multiples. What the model fails to detect is any signal from the growth, profitability, or risk variables. Of the eleven regressors, the two that survive are a deal-process indicator and a scale control, neither of which describes the target's operating performance. The important implication to keep in mind in this section is that whatever the ML models achieve on EV/EBITDA, is achieved in a context with almost no linear relationship between the fundamentals and the multiple, except for two variables that do not represent operating performance.

In opposition with the EBITDA model, the revenue model leads to different statistical results (see Table 16). With an adjusted R^2 of 0.028 and a p-value of 0.031, the fundamentals as a group contain some linear relationships with the EV/Revenue multiples. However, the Breusch-Pagan test returns a p-value of 0.051, which is borderline enough to warrant HC3 robust SE, and no individual variable is significant at 5% under those corrected SE. Statistically, the fundamentals are jointly associated with the EV/Revenue multiple at conventional levels, but the association is not strong enough for any individual variable to be identified once inference is adjusted for the borderline heteroscedasticity.

Concerning the addition of market context variables, it has a very different impact on the EBITDA model in comparison with the revenue model. For the augmented EBITDA model, it reaches an adjusted R^2 of 0.029 with a p-value of 0.040, making this model statistically significant, in opposition with the pure EBITDA model (see Table 17). The variable explaining this shift in statistical significance is the Stoxx Europe 600 at the transaction closing date with a p-value of 0.042. This impact makes sense and is economically coherent, since multiples are exposed to credit conditions and equity market appetite. Indeed, the Vlerick M&A Monitor records European mid-market EV/EBITDA multiples moving from 6.2 in the mid-2010s to a peak of 8.5 before settling at 6.5 in 2024, confirming that private transaction pricing tracks the wider market cycle rather than firm fundamentals alone (Luypaert et al., 2025). By contrast, the augmented revenue model is slightly worse than the pure one with an adjusted R^2 falling from 0.028 to 0.022 and the F-test losing significance. This augmented versus pure distinction highlights the difference of the two multiples (EV/EBITDA and EV/Revenue) in their sensitivity to market context.

Once run on the test set, the regression confirms the predicted results in the training set. Indeed, OLS pure achieves a MedAE of 8.00x on EV/EBITDA and 1.68x on EV/Revenue. OLS augmented improves modestly to 7.24x on EBITDA but worsens to 1.94x on revenue. Neither of these models reaches the performance of the analyst method benchmark on either metric (see Table 19). However, these regression results are mostly used in this analysis to assess the non-linear relationships that ML models could achieve.

The analysis also contains a Ridge robustness check that confirmed the OLS specification is stable. Indeed, the two variables that carry real signal, log of total assets and the PE-backed dummy, keep their signs and relative ranking (see Table 18).

To conclude the linear regression analysis, financial fundamentals carry almost no measurable linear relationship with EV/EBITDA, with an adjusted R^2 of 0.018 and only a scale control and a buyer-type indicator surviving. The relationship is weak for EV/Revenue and does not survive robust standard errors. The last key takeaway is that augmentation raises explanatory power for EV/EBITDA and reduces it for EV/Revenue, the first appearance of the asymmetry between the two multiples.

Machine Learning

For the EBITDA model, the CV MAE across all eight specifications runs from 13.31x to 14.04x (see Table 20), a spread of 0.73x, or roughly 5.5% of the best value. Restricting attention to the

four pure specifications narrows the range further, to 13.31x-13.79x. Either way, the difference in output produced by the best and worst parameter choices is small, meaning that the data contains little learnable structure for algorithmic sophistication to differentiate outcomes. For the revenue model, the eight specifications run from 3.31x to 3.53x, and XGB outperforms RF in every one. In absolute terms, this spread of 0.22x is smaller than the EBITDA spread, but the two multiples sit on different scales, and relative to the best model the revenue spread is the wider of the two at roughly 6.6% against 5.5% for EBITDA. This finding is coherent with the linear regression assessing both multiples individually. On the test set, the best pure ML result for EV/EBITDA is RF ML-B at 7.62x MedAE (see Table 21, Table 22 and Figure 3). The improvement in comparison with OLS pure (8.00x) is 0.38x. All four pure specifications sit below OLS pure on this metric, by between 0.06x and 0.38x. Two qualifications are required before that is read as evidence of non-linearity. First, only the two ML-A specifications share the regression's feature set; the ML-B specifications add eleven variables, so any difference there confounds functional form with information. The clean comparisons are RF ML-A at 7.70x and XGB ML-A at 7.94x against OLS pure at 8.00x, gains of 0.30x and 0.06x respectively. Second, the tree ensembles differ from the regression in regularisation as well as in functional form: with the retained RF parameters at minimum node size twenty and depth four, and XGB at depth two with a 0.05 learning rate, both are heavily shrunk, whereas the regression is unpenalized, so part of even that margin reflects variance reduction rather than captured non-linearity. The result is consistent with the presence of thin non-linear structure, however, it does not close the gap to the analyst benchmark.

Firstly, on the ML-A versus ML-B comparison, adding the eleven horizon and missingness variables changes almost nothing for either algorithm on EV/EBITDA. XGB ML-A and XGB ML-B differs by 0.01x of MedAE (7.94x versus 7.93x), while RF ML-B improves marginally on RF ML-A (7.62x versus 7.70x). Concerning RF models, adding these eleven additional variables very lightly improves the models' performance. In fact, RF ML-B slightly outperforms RF ML-A on EV/EBITDA (7.62x versus 7.70x). Due to very small improvement, the pattern of missing data therefore carries almost no detectable predictive information about transaction multiples beyond the imputed values.

Secondly, the comparison between OLS and the pure ML models on EV/Revenue depends on the metric, and the divergence is itself informative. On MedAE across the full test set, all four pure ML specifications outperform OLS pure, which ranks last of the five at 1.68x against 1.50x to 1.64x. Within the typical bucket the ranking reverses on MAE, where OLS pure is the most accurate statistical model at 2.36x against 2.72x to 3.14x for the tree ensembles. The tree

models therefore produce more small errors but a longer right tail of larger ones inside the typical range, while OLS is more uniformly middling. Neither ordering supports a claim of ML superiority on this multiple, and what both share is that the distance between linear and non-linear specifications is small: roughly 0.22x of MedAE separates the best specification from the OLS pure baseline on revenue, against 1.66x on EV/EBITDA. This is consistent with a revenue relationship that is close to additive in a small number of growth and scale variables, which the augmented revenue regression supports through its two significant growth coefficients. However, the feature importance evidence, examined next, does not corroborate it.

The feature importance analysis also gives interesting results, in systematic opposition between algorithms, for both multiples. On one hand, RF identifies EBIT margin (40.2 permutation importance) and EBITDA margin (30.0) as the dominant variables, that together account for the majority of total permutation importance, with size variables behind (see Table 23 and Figure 4). This implies that RF is learning that current profitability is the primary factor influencing the EV/EBITDA multiple. On the other hand XGB spreads its gain more evenly, with log of total assets (0.262) and log of employees (0.253) effectively tied at the top and the two margin variables placed mid-table and low respectively. Therefore, XGB most highly considers proxies for business models and market positions when predicting multiples, compared to profitability margins. Growth variables (CAGR revenue, CAGR EBITDA, growth acceleration) carry negative permutation importance in RF and only 15% of XGB's total gain for EBITDA, against 63% for EV/Revenue. Indeed, the revenue models do not resolve this disagreement, they reproduce it. XGB concentrates 95% of its gain on three variables, log employees (0.332), CAGR EBITDA (0.326) and CAGR revenue (0.296), while RF again places the two margin variables first, EBIT margin at 24.4 and EBITDA margin at 22.9, and assigns negative permutation importance to both CAGR variables, at −3.58 and −6.99. Each algorithm therefore returns the same ordering for both multiples, profitability first for RF and scale or growth first for XGB, so the conflict is a constant of the exercise rather than a property of EV/EBITDA. This is the case the methodology anticipated: gain records how heavily a variable was used to grow the trees, permutation records whether making that variable uninformative damages prediction, and a variable carrying a large share of gain while returning negative permutation importance is one the model split on repeatedly without buying accuracy. The growth reading of EV/Revenue therefore rests on XGB's gain and on the two significant growth coefficients of the augmented revenue regression, and RF does not corroborate it.

The augmented ML models also bring interesting results. For EV/EBITDA, XGB Aug-B achieves a MedAE of 6.34x, which is 17% better than the best pure ML result (RF ML-B at 7.62x) and

12% better than the best OLS (augmented at 7.24x). This stays true after removing extreme transactions, at 5.89x against 7.26x. These results are explained in the feature importance analysis (see Table 24 and Figure 5). Indeed, in RF Aug-A, the comp-set median records the highest permutation importance of any variable at 33.9, ahead of the 10-year rate at 20.6 and EBIT margin at 14.9. In XGB Aug-A, the three market variables (comp-set median, Stoxx 600, 10-year rate) represent together approximately 32% of the total distributed gain. These two models can be compared to the analyst method because they draw on the same information: what the market says comparable companies are worth, macro conditions, and company characteristics. However, the way they use it differs. The analyst method proceeds in two steps, anchoring on the comp-set median and then applying a fixed adjustment, whereas the augmented models estimate a single function in which the median enters as one feature among fourteen, with its weight determined by the fit rather than imposed. The importance rankings show the consequence: the median is the single most important variable for RF but only sixth for XGB, where it is effectively tied with the Stoxx 600 and sits behind the ten-year rate.

To compare these augmented models with the linear regression, they also study the non-linear interactions between market context and company attributes that the OLS cannot account for in a single additive coefficient. For instance, the Stoxx 600 level, which was the key significant variable in the augmented OLS, is here one component of a richer and non-linear function. This dominance of the peer-group anchor is not specific to valuation: applying the same family of models to capital structure, Amini et al. (2021) found that RF selects industry median leverage among the small set of reliably informative predictors, alongside size and profitability. However, augmented models do not improve efficiency for EV/Revenue. No market context data adds signal that growth fundamentals did not already reflect. This confirms an asymmetry in the added value of market information on both multiples (see Table 25 and Figure 6), and this asymmetry reflects the economic definition and structure of the two multiples. On one side, EBITDA is the basis for leverage, making EV/EBITDA multiple sensitive to credit conditions and risk aversion. On the other side, revenue is a pre-leverage and pre-cost metric, which is not sensitive to the financing environment.

In conclusion of the ML models analysis, the best ML specification improves modestly on the linear baseline (XGB Aug-B at 6.34x against OLS pure at 8.00x, a reduction of roughly 21%), confirming that non-linear structure exists, and is richer in augmented models than in pure ones. The two algorithms disagree on which fundamentals matter, and they disagree in the same direction for both multiples: RF ranks profitability first, through EBIT margin and EBITDA

margin, while XGB ranks total assets and employees first for EV/EBITDA and employees with the two growth rates for EV/Revenue. The disagreement is therefore a property of the two importance measures rather than a diagnostic of either multiple. It is sharpest on EV/Revenue, where RF returns negative permutation importance for the two CAGR variables that carry 62% of XGB's gain, indicating a growth signal the boosting model uses heavily without measurably improving accuracy. In addition, market context adds a further 17% for EV/EBITDA and nothing for EV/Revenue, and in the augmented models the comp-set median becomes the single most important variable for RF.

Comparison and Analysis of All Methods

After analysing the MedAE for EV/EBITDA, the analyst benchmark beats all other methods with a MedAE of 4.07x (see Table 26). The broad ordering places OLS pure at the bottom (8.00x) and the augmented XGB specifications at the top (6.34x and 6.66x). RF Aug-B (7.41x) is outperformed by OLS augmented (7.24x), and every augmented specification outperforms its pure counterpart on EV/EBITDA MedAE, without exception across both algorithms and the linear control, so the ranking supports that market context systematically helps. Firstly, this improvement over the linear baseline points to a non-linear relationship between the explanatory variables and the EV/EBITDA multiple, which the tree ensembles can capture and the regression cannot. This is consistent with the finding of Gu et al. (2020) that tree ensembles extract predictive structure from firm characteristics that linear models cannot access by construction. Secondly, these results also reflect the importance of market context and macro conditions variables, with each augmented specification outperforming its pure counterpart on EV/EBITDA.

After analysing the EV/Revenue column, it shows results reinforcing the two-multiple asymmetry developed in the literature review. The analyst benchmark again dominates, at 0.58x, but its lead over the field is proportionally larger even though the absolute errors are an order of magnitude smaller than for EV/EBITDA. It is also interesting to notice that the two gradients that organised the EBITDA results were both essentially dissolved. Augmentation delivers only a marginal benefit for revenue, with three of the four tree pairings improved by under 5% and augmented OLS at 1.94x is worse than pure OLS at 1.68x, against gains of 16% and 20% for the two XGB pairings on EV/EBITDA. The outperformance of tree models versus linear analysis, while still present on MedAE, narrows to a small margin that no longer survives a change of metric: within the typical bucket, OLS pure is in fact the most accurate statistical model on MAE (see Table 28). This is the predictable consequence of a signal that is already

close to linear. As shown in the regression analysis and anticipated by the Gordon-Shapiro decomposition of the revenue multiple in the literature review, EV/Revenue is driven by an identifiable and largely additive set of growth and scale variables, so the flexible machinery of tree ensembles adds variance rather than signal. The same economics explain why market-context variables help EBITDA but not revenue. Indeed, revenue is a pre-leverage, pre-cost metric that carries none of the exposure to credit conditions and equity-market appetite that the augmented feature set encodes. By contrast, EBITDA, being the base on which acquisition leverage is sized, is directly sensitive to them (Damodaran, 2006).

The most consequential result of the comparison is that no model, including the strongest augmented XGB, closes the gap to the analyst benchmark on the typical transactions which account for 78% of the EV/EBITDA test set. On EV/EBITDA, the residual gap of 2.27x MedAE (4.07x versus 6.34x) is still observable after the removal of extreme transactions (3.81x versus 5.89x), and is proportionally even larger for EV/Revenue. The gap exists even though the augmented models were deliberately supplied with the sector comp-set median that serves as the analyst's own pricing anchor. In real cases, the method's edge could be attributed to the private, forward-looking knowledge a real advisor brings to a deal such as management projections, deal-structure terms, competitive tension, and buyer motivation that a historical database cannot record, and which Liu et al. (2002) show to dominate predictive power in valuation accuracy. However, that explanation does not apply here because the benchmark in this study is a mechanical reproduction of the method rather than a practitioner. In fact, it is built on the sector-by-year comp-set median scaled by a single calibrated discount, with no access to projections, deal terms, or qualitative judgement. A real advisor would, in addition, curate the peer set and layer in that judgement (Eaton et al., 2022), all of which the reproduction deliberately omits, making it a conservative and information-poor version of the method. In this analysis, the benchmark operates on strictly less information than the augmented models, which received that same comp-set median plus the eleven fundamentals and the market-context variables, together with a tool capable of learning non-linear relationships. The fact that the information-poor anchor still wins therefore isolates the explanation to the structure of the signal itself rather than to any data availability advantage.

This comparison highlights the fact that the comp-set median is not one variable among many but a near-sufficient statistic for market-relative value. It is understandable that the traded prices of listed peers already aggregate the market's forward-incorporating assessment of what firms in the sector are worth at that moment, and their median is an accurate estimate of it (Yan, 2024; Bourveau et al., 2023). By contrast, the fundamentals used in the other models

are trailing accounting figures that carry almost no cross-sectional signal for the EBITDA multiple, and conditioning on them adds variance without reducing bias. At roughly 370 training observations the simple market anchor is the better-regularised estimator, and the flexible learner cannot improve on it. At root this is a bias-variance outcome, and it is the empirical counterpart of the argument made in the literature review: a market-relative multiple encodes a pricing that backward-looking company data cannot reconstruct regardless of modelling sophistication. It also sharpens the contrast with Geertsema and Lu (2023) and Staňková (2024), who report ML outperforming peer-group methods when pricing listed firms against their own observable market multiples. In the private setting studied here, the target has no such price, yet the peer median substitutes for it more efficiently than any combination of fundamentals the models could exploit.

However, the reconstructed analyst method's advantage is not unconditional, and the point at which it breaks down is the clearest opening for a data-driven approach. Segmenting the test set by multiple range shows that the benchmark's dominance is limited to the typical group of transactions. On the 16% of transactions in the elevated range (15-50x), every statistical specification tested outperforms the analyst method, from OLS pure at 10.81x through to the best of them, RF Aug-B at 8.98x, against 17.19x for the benchmark (see Table 27 and Figure 7). The reversal is therefore not an artefact of one algorithm or of the augmented feature set: it holds for all ten specifications, including the unpenalised linear control operating on eleven fundamentals alone, and it recurs in the same direction on EV/Revenue (Table 28). The mechanism is exactly the one anticipated previously. The analyst method applies a single calibrated discount (21.5% for EV/EBITDA) uniformly to every deal, yet the PCD is not a constant. Koeplin et al. (2000) document that it ranges from roughly 14% for large, creditworthy targets to 50% for small, distressed ones, and Klein and Scheibel (2012) show that each transaction also embeds a control premium of its own. A fixed discount applied to the sector median therefore carries a structural limitation. As observable in Table 10, the impact of it is limited. Indeed, removing the discount entirely changes test-set MedAE by less than half a turn, so the discount can account for only a small part of the 17.19x recorded on the elevated bucket. The larger part is attributable to the anchor rather than to the adjustment applied to it, since a single sector median cannot represent a premium-positioned target at any discount level. The augmented RF, by contrast, learns a non-linear adjustment around the comp-set anchor and can raise its prediction when the configuration of fundamentals and market conditions warrants it. This is the most constructive finding of the study, though it comes with a cost that must be stated alongside it: the augmented RF wins the elevated bucket

while losing the typical bucket by a wide margin (RF Aug-B at 7.54x against 3.33x), so it does not repair the benchmark's blind spot so much as trade its principal strength for it. No specification tested here dominates both sub-populations. What the result establishes is that the failure is addressable in principle, on the high-premium transactions where the auction dynamics modelled by Gorbenko and Malenko (2014) and the PE premium documented by Bargeron et al. (2008) push prices above what a fixed discount can reach.

In addition, four major conclusions can be drawn beyond a simple comparative ranking. First, the adjusted R^2 of the pure EBITDA regression (0.018) is a substantive statement about the limits of information available in financial statements. When multiples are set in competitive auctions based on forward expectations, and when the underlying accounting figures are the noisier disclosures of private firms (Bitetto et al., 2025), historical fundamentals as a group do not determine the price, and the two coefficients that survive describe scale and buyer type rather than operating performance. Second, the improvement of the pure tree ensembles over OLS on the identical feature set averages 0.20x of MedAE, or 2.5%, confirming that non-linear structure exists in these data but is thin rather than rich. The larger 1.66x distance between OLS pure and the best augmented specification is not only a measure of non-linearity, since it also reflects three additional variables. Third, and more promising, the 20% gain of augmented over pure XGB is the clearest evidence that market information carries signal the model can exploit non-linearly in ways the additive regression cannot. The inclusion of the STOXX Europe 600 index, represented in the augmented OLS model by a single significant coefficient, allowed the tree ensemble to exploit more complex interactions with company-specific attributes. Lastly, another main conclusion of this study is the persistent asymmetry between the two multiples. The ML feature importance results are less decisive on this point: growth dominates XGB's gain for EV/Revenue and stays marginal for EV/EBITDA, but RF ranks profitability margins first for both multiples and returns negative permutation importance for the growth variables in each, so only XGB supports the asymmetry. That the two multiples reward different fundamentals is not a modelling artifact but a direct consequence of their different construction, established theoretically in the literature review. EV/Revenue multiples are more predictable and more stable across sectors, and the growth and firm-size reading of what drives them comes from XGB's gain and from the augmented revenue regression rather than from agreement between the two algorithms. EV/EBITDA multiples are less predictable and more sensitive to deal type and macro conditions. The conflicting feature importance rankings across RF and XGB are not a property of this multiple: they recur unchanged on EV/Revenue, and measure the distance between permutation importance and gain. This traces directly to

the economic definitions established in the literature review. Because EBITDA is the basis on which acquisition leverage is sized, its multiple is exposed to credit conditions and risk aversion, which is not the case concerning the pre-leverage revenue multiple. The two multiples represent distinct prediction problems and treating them as interchangeable within a single valuation model would therefore be inappropriate.

On the central research question, the answer is that ML does not match the analyst benchmark on typical transactions, for either multiple, not even with the strongest augmented XGB specification. The practitioner method is roughly twice as accurate across the deals that trade near their sector norm (78% of the EV/EBITDA test set). But this advantage reverses on the elevated deals where the fixed discount breaks down, and the correct reading of the overall results is not that ML fails in M&A valuation, but that the comparables approach, where it works, works because its functional form enforces a disciplined anchoring to the sector median that the ML models, despite having access to the same comp-set information as a feature, do not replicate. The analyst method's advantage on typical transactions is structural: its prediction is the comp-set median minus a fixed discount, leaving no room to drift from the market anchor. Accordingly, the augmented ML's advantage on elevated transactions is also structural: its flexibility to depart from that anchor, penalised on the majority of deals, becomes an asset on the minority where the fixed discount is systematically too conservative. The question this opens for future research is whether a hybrid architecture, enforcing the comp-set anchor as a structural constraint while allowing a data-driven and non-linear adjustment around it, could preserve the analyst method's accuracy on typical deals while repairing its identifiable failure on premium-positioned ones, and this opening will be further developed in the following sections.

Limitations and Future Research

Sample Size

The sample is quite restricted with roughly 370 training and 124 test observations, for models using up to 25 features. Grouped CV and conservative grids limit overfitting without removing it, and two readings in the results depend on this. The absence of a difference in performance between ML-A and ML-B cannot be established on 124 observations, since sampling noise could have produced the same pattern. Two sectors are also too thin to interpret, Commercial Printing and Security and Alarm Services having six and five test transactions. The scale of the

constraint is visible by comparison: Amini et al. (2021) obtain an increase in out-of-sample R^2 from 36% to 56% over OLS and LASSO using the same class of tree ensembles, but do so on 128,417 firm-year observations, roughly three hundred times the training sample available here. These modest results are also consistent with the conditions under which ML is expected to be most effective. Indeed, Hoang and Wiegratz (2023) argue that ML outperforms traditional methods primarily when the data are high-dimensional, that is, when the number of variables is large relative to the number of observations. With eleven to twenty-five features against roughly 370 training transactions, the present setting is not high-dimensional, and the expected advantage of flexible learners is correspondingly limited. Accounting for the limits of this sample, three extensions of the study follow. Firstly, repeating the entity-grouped partition many times and recording the distribution of each method's error would attach confidence intervals to the narrower gaps reported here as point estimates. Secondly, replicating on a larger sample of transactions, while tracking performance as the training set grows, would establish at what scale the ranking reverses. Lastly, the low dimensionality of the setting is itself worth testing. Hoang and Wiegratz (2023) locate ML's advantage where the variable space is wide relative to the number of observations, a condition this study does not meet. Combining a larger set of transactions with a broader range of financial fundamentals would move the setting toward that regime and would establish whether the modest gains observed here reflect the limits of the method or the limits of the data available to it.

Historical Data and Imputation

Firstly, all methods use data looking backwards, and Liu et al. (2002) show that forward earnings predict better than historical ones. This affects every method equally, so the comparison remains valid, but the reproduction of the analyst method in this thesis understates what advisors can achieve when using management projections. Secondly, the imputation also represents a limitation due to data availability. In this sample, individual variables are missing for between a fifth and a half of observations, and a three-year growth rate can be computed for only around half. Discarding incomplete records would have removed most of the sample, so missing values were imputed from sector medians and shorter growth horizons were used where longer ones were unavailable. Replacing missing values with a sector median removes cross-sectional variation, and a variable with less variation produces a coefficient pulled toward zero. The adjusted R^2 explaining under 2% of variance is therefore a lower bound rather than an unbiased estimate: it shows what a fundamentals-based model can achieve on the data available for private companies, not that fundamentals are irrelevant (Bitetto et al., 2025). Of the eleven regressors only two survive, a deal-process indicator and a

size control, so the growth, profitability and risk blocks contribute nothing detectable. Whether unobserved deal characteristics such as auction intensity or bidder composition would have supplied that missing signal cannot be established here, since no such variable is available in either database (Greene, 2017; Gorbenko & Malenko, 2014). Concerning the future research this imputation-related limitation opens, comparing model performance across sub-samples with differing disclosure completeness would separate the effect of the data environment from a genuine absence of relationship, and would place a lower bound on what a fundamentals-based model could achieve on complete accounts.

Analyst Benchmark Design

The benchmark applies one discount to all sectors, sizes and buyer types, while the literature shows it varies with all three (Koeplin et al., 2000; Klein & Scheibel, 2012). Conditioning it was not possible, since the smallest sectors leave fifteen to nineteen training observations. This is a deliberate simplification, and its cost is the failure on the elevated bucket, so the advantage of the augmented RF there is measured against a benchmark not profiting from the flexibility the literature recommends. In addition, it should be acknowledged that classification-based matching is an imperfect proxy for genuine comparability: Eaton et al. (2022) find that product market space is among the most important factors in how investment banks actually select peers, and that SIC codes are not the best way to categorise. This points to the most direct extension of the study: building a discount conditioned on size, profitability and buyer type, then re-running the comparison, would establish whether the augmented models keep their advantage on the elevated bucket once the benchmark is granted that flexibility.

Scope

The seven sub-sectors are asset-light, with value in client relationships rather than in recorded assets, which may partly explain why fundamentals carry so little signal for EV/EBITDA. The result should not be expected to hold in capital-intensive sectors. The sample is also European, drawn from registries with different disclosure rules. The split is also grouped by entity rather than by date, so temporal ordering is not preserved, whereas the machine-learning literature in finance conventionally splits chronologically (Gu et al., 2020; Amini et al., 2021). Each boundary suggests potential future research. Extending to capital-intensive sectors would test whether fundamentals regain explanatory power where fixed assets carry pricing information, and a chronological split would establish whether the augmented models' reliance on the rate and the index survives a change of regime, which is not a hypothetical concern: Staňková (2024) finds gradient-boosted predictions becoming more accurate over time while the

industry benchmark stays flat. The hybrid remains the most promising avenue: fixing the comp-set anchor as a constraint and estimating only the deviation would concentrate flexibility on the transactions where the benchmark fails.

Conclusion

This thesis asked whether a ML model trained on financial fundamentals can price private mid-market acquisitions more accurately than the comparables rule that investment bankers apply. Across a sample of 513 European Business Services transactions, of which 493 carry an EV/EBITDA multiple and 494 an EV/Revenue multiple, and eleven competing specifications, the answer is no. The analyst method wins on the transactions that make up the bulk of the market, and it does while operating on strictly less information than the models measured against it. The reason behind it is more interesting than the ranking itself.

This thesis brings several results that give a clear answer to the four sub-questions posed in the introduction. Firstly, financial fundamentals carry almost no measurable linear relationship with the EV/EBITDA multiple. The pure regression returns an adjusted R^2 of 0.018, and of the two coefficients that survive, one is a scale control and the other a buyer-type indicator. The relationship is real but weak for EV/Revenue. Secondly, ML does improve on the linear baseline, and because the comparison holds the feature set fixed, that improvement is attributable to non-linearity rather than to specification. The gain is nonetheless small: averaged across the four pure specifications it amounts to 0.20x of median error, or 2.5%. Non-linear structure exists in these data, but it is thin rather than rich, and it is not what separates the winning method from the losing ones. Thirdly, market information contributes to the accuracy of the models, and it does where theory predicts it should. Adding the comp-set median, the equity index, and the ten-year rate improves the best EBITDA specification by 17% while doing nothing for revenue. It does not, however, reach the analyst method prediction performance. Lastly, the answer is emphatically not uniform. It differs by multiple, since revenue is more predictable and indifferent to market context, and it differs by transaction, since the benchmark's dominance is confined to deals trading near their sector norm and reverses on those trading well above it.

What follows from this is that the comp-set median should not be treated as one predictor among several. The augmented models were handed exactly that number and still drifted away from it, because a learner with fourteen features and 370 observations has every

incentive to spread weight across them. The benchmark cannot drift, and on these data that constraint is worth more than the flexibility it gives up. The comparison is therefore not really a flexible model against a rigid rule, but a disciplined use of a strong signal against an undisciplined use of the same one. To conclude, the rule's persistence is not professional inertia. It is a well-regularised estimator that practice discovered before econometrics could derive it. That claim is bounded by the setting that produced it, and the boundaries are as much a finding as the result they contain.

Two sets of implications follow, developed at greater length in the preceding section: one for practice, one for research. For practice, the study locates the analyst method's weakest component precisely: the uniform discount, which conditioning on size, profitability and buyer type would improve without disturbing what already works. It also identifies where ML earns its place. Since the augmented models are more accurate on transactions trading above their sector norm, running both approaches and comparing their outputs gives the analyst a usable signal: where the model's prediction sits materially above the comparables estimate, the target's profile resembles that of deals the fixed discount historically under-prices. The role is diagnostic rather than substitutive, and whether that divergence is a reliable ex-ante indicator would require a dedicated analysis. For research, the bias-variance reading developed here supplies a design principle for a hybrid model. Rather than supplying the comp-set median as one feature among 14, as the augmented specifications did, it would be fixed as the base of the prediction, with the model estimating only the deviation from it. The output remains a predicted multiple, but the anchor is imposed rather than learned, so the model can only correct the comparables estimate instead of replacing it, and that correction matters most on the minority of deals where the anchor is wrong. Beyond that, the reliance of the augmented models on the interest rate and the equity index makes regime stability a natural next question. A chronological split, training on earlier transactions and testing on later ones, would establish whether the gain transfers to conditions the model has not seen.

The most useful finding of this thesis is therefore not that the rule wins, but that its defeat now has a map. The comparables method fails on a bounded, identifiable and economically interpretable minority of extreme transactions, and it is on that minority alone that the case for a data-driven method appears to be more accurate.

Bibliography

Alford, A. W. (1992). The effect of the set of comparable firms on the accuracy of the price-earnings valuation method. *Journal of Accounting Research, 30*(1), 94-108. https://doi.org/10.2307/2491093

Amihud, Y., & Mendelson, H. (1986). Asset pricing and the bid-ask spread. *Journal of Financial Economics, 17*(2), 223-249. https://doi.org/10.1016/0304-405X(86)90065-6

Amini, S., Elmore, R., Öztekin, Ö., & Strauss, J. (2021). Can machines learn capital structure dynamics? *Journal of Corporate Finance, 70*, 102073. https://doi.org/10.1016/j.jcorpfin.2021.102073

Bargeron, L. L., Schlingemann, F. P., Stulz, R. M., & Zutter, C. J. (2008). Why do private acquirers pay so little compared to public acquirers? *Journal of Financial Economics, 89*(3), 375-390. https://doi.org/10.1016/j.jfineco.2007.11.005

Bhojraj, S., & Lee, C. M. C. (2002). Who is my peer? A valuation-based approach to the selection of comparable firms. *Journal of Accounting Research, 40*(2), 407-439. https://doi.org/10.1111/1475-679X.00054

Bitetto, A., Filomeni, S., & Modina, M. (2025). Machine learning for the unlisted: Enhancing MSME default prediction with public peer signals. *Journal of Corporate Finance, 94*, 102830. https://doi.org/10.1016/j.jcorpfin.2025.102830

Bourveau, T., Chen, J. V., Elfers, F., & Pierk, J. (2023). Public peers, accounting comparability, and value relevance of private firms' financial reporting. *Review of Accounting Studies, 28*(4), 2642-2676. https://doi.org/10.1007/s11142-022-09707-y

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

Damodaran, A. (2006). *Valuation approaches and metrics: A survey of the theory and evidence* [Working paper]. Stern School of Business, New York University.

De Franco, G., Hope, O.-K., & Larocque, S. (2015). Analysts' choice of peer companies. *Review of Accounting Studies, 20*(1), 82-109. https://doi.org/10.1007/s11142-014-9294-7

Eaton, G. W., Guo, F., Liu, T., & Officer, M. S. (2022). Peer selection and valuation in mergers and acquisitions. *Journal of Financial Economics, 146*(1), 230-255. https://doi.org/10.1016/j.jfineco.2021.09.006

Geertsema, P., & Lu, H. (2023). Relative valuation with machine learning. *Journal of Accounting Research, 61*(1), 329-376. https://doi.org/10.1111/1475-679X.12464

Gompers, P., Kaplan, S. N., & Mukharlyamov, V. (2016). What do private equity firms say they do? *Journal of Financial Economics, 121*(3), 449-476. https://doi.org/10.1016/j.jfineco.2016.06.003

Gorbenko, A. S., & Malenko, A. (2014). Strategic and financial bidders in takeover auctions. *The Journal of Finance, 69*(6), 2513-2555. https://doi.org/10.1111/jofi.12194

Greene, D. (2017). Valuations in corporate takeovers and financial constraints on private targets. *Journal of Financial and Quantitative Analysis, 52*(4), 1343-1373. https://doi.org/10.1017/S0022109017000527

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies, 33*(5), 2223-2273. https://doi.org/10.1093/rfs/hhaa009

Guo, F., Liu, T., & Tu, D. (2023). Neglected peers in merger valuations. *The Review of Financial Studies, 36*(8), 3257-3298. https://doi.org/10.1093/rfs/hhad013

Hoang, D., & Wiegratz, K. (2023). Machine learning methods in finance: Recent applications and prospects. *European Financial Management, 29*(5), 1657-1701. https://doi.org/10.1111/eufm.12408

Klein, C., & Scheibel, M. (2012). The private company discount from a European perspective: An analysis based on the acquisition approach for comparable transactions of European target companies. *The Journal of Private Equity, 16*(1), 74-82. https://doi.org/10.3905/jpe.2012.16.1.074

Koeplin, J., Sarin, A., & Shapiro, A. C. (2000). The private company discount. *Journal of Applied Corporate Finance, 12*(4), 94-101. https://doi.org/10.1111/j.1745-6622.2000.tb00022.x

Liu, J., Nissim, D., & Thomas, J. (2002). Equity valuation using multiples. *Journal of Accounting Research, 40*(1), 135-172. https://doi.org/10.1111/1475-679X.00042

Lukanima, B. K. (2023). *Corporate valuation: A practical approach with case studies*. Springer. https://doi.org/10.1007/978-3-031-28267-6

Luypaert, M., Muller, S., & Floru, T. (2025). *2025 M&A monitor: Shedding light on M&A in Belgium*. Vlerick Business School, Centre for Mergers, Acquisitions and Buyouts.

Rodríguez-Valencia, L., Lamothe-Fernández, P., & Alaminos, D. (2023). The market value of SMEs: A comparative study between private and listed firms in alternative stock markets. *Annals of Finance, 19*(1), 95-117. https://doi.org/10.1007/s10436-022-00420-z

Shaffer, M. (2024). Which multiples matter in M&A? An overview. *Review of Accounting Studies, 29*(3), 2724-2752. https://doi.org/10.1007/s11142-023-09768-7

Silber, W. L. (1991). Discounts on restricted stock: The impact of illiquidity on stock prices. *Financial Analysts Journal, 47*(4), 60-64. https://doi.org/10.2469/faj.v47.n4.60

Staňková, V. (2024). (Out)smart the peer group in market comparison: Building business valuation multiples by machine learning. *European Journal of Business Science and Technology, 10*(2), 156-172. https://doi.org/10.11118/ejobsat.2024.011

Yan, D. (2024). Do private firms (mis)learn from the stock market? *Review of Finance, 28*(5), 1483-1511. https://doi.org/10.1093/rof/rfae020

Appendices

Methodology

Table 3. Sample construction

Step	Transactions	Removed	Rationale
Initial Capital IQ × Orbis intersection	542	—	Deal multiple and target financials both available
Sector filter applied	513	29	Retain only sectors with ≥3 listed comparables per sector-year cell
of which EV/EBITDA available	**493**	20	
of which EV/Revenue available	**494**	19	

Table 4. Sector composition and train/test allocation

Sector	EBITDA train	EBITDA test	Revenue train	Revenue test
Research and Consulting Services	105	35	107	35
Diversified Support Services	96	34	99	32
Environmental and Facilities Services	68	23	66	22
Human Resource and Employment Services	43	13	44	15
Office Services and Supplies	22	8	22	8
Security and Alarm Services	19	5	17	6
Commercial Printing	16	6	15	6
Total	**369**	**124**	**370**	**124**

Totals differ between the two multiples because availability of EBITDA and revenue is not identical across transactions.

Table 5. Retained explanatory variables

Block	Variable	Economic rationale
Size	Log total assets	Financial scale; log-transformed to correct right skew
	Log employees	Operational scale, informative for asset-light services firms
Growth	CAGR revenue	Primary theoretical driver of the revenue multiple
	CAGR EBITDA	Quality of growth beyond top-line expansion
	Growth acceleration revenue	Change in growth trajectory that a single CAGR conceals
Profitability	EBITDA margin	Current operating profitability
	EBIT margin	Retained alongside EBITDA margin; the gap proxies asset intensity
Risk and structure	Volatility EBITDA	Earnings instability, proxy for execution risk
	ROA	Efficiency of asset deployment
	Equity ratio	Capital structure and financial leverage
Deal type	Sponsor backed dummy	1 if PE-backed, 0 if strategic acquirer

Table 6. Feature sets used by each model

Specification	N	Composition	Used by
ML-A	11	The eleven variables of Table 5	OLS pure, RF ML-A, XGB ML-A
ML-B	22	ML-A + 2 CAGR horizon indicators + 9 missingness dummies	RF ML-B, XGB ML-B
Augmented ML-A	14	ML-A + comp-set median + Stoxx 600 + 10Y rate	OLS augmented, RF Aug-A, XGB Aug-A
Augmented ML-B	25	ML-B + comp-set median + Stoxx 600 + 10Y rate	RF Aug-B, XGB Aug-B

ML-B variables were excluded from the regression because of collinearity (tree ensembles carry no such constraint).

Table 7. Variable transformations applied

Variable	Treatment	Raw training range	Post-treatment range
Volatility EBITDA	Log transformation	7.86 – 188,911	2.06 – 12.15
Growth acceleration revenue	Winsorisation at 1st / 99th pct	-572.28 – 6.76	-6.63 – 1.92
CAGR EBITDA	Winsorisation at 1st / 99th pct	-44.44 – 47.76	-17.72 – 3.02
Equity ratio	Winsorisation at 1st / 99th pct	-5.02 – 1.00	-0.527 – 0.981

All thresholds derived from the EBITDA training set and applied unchanged to every other partition. Winsorisation clipped four observations at each tail of the three affected variables. Volatility EBITDA was log-transformed only, the transformation being sufficient to remove the skew without further treatment.

Table 8. Train/test balance diagnostics

Criterion	Train	Test	Assessment
Observations (EV/EBITDA)	369	124	25.2% held out
Observations (EV/Revenue)	370	124	25.1% held out
Mean EV/EBITDA	16.18x	17.32x	Comparable
Median EV/EBITDA	8.97x	9.35x	Comparable
Mean closing year	2019	2019	Comparable
Sector proportions	—	—	Match within 0.5 percentage points
Entities present in both sets	0	0	No entity-level leakage

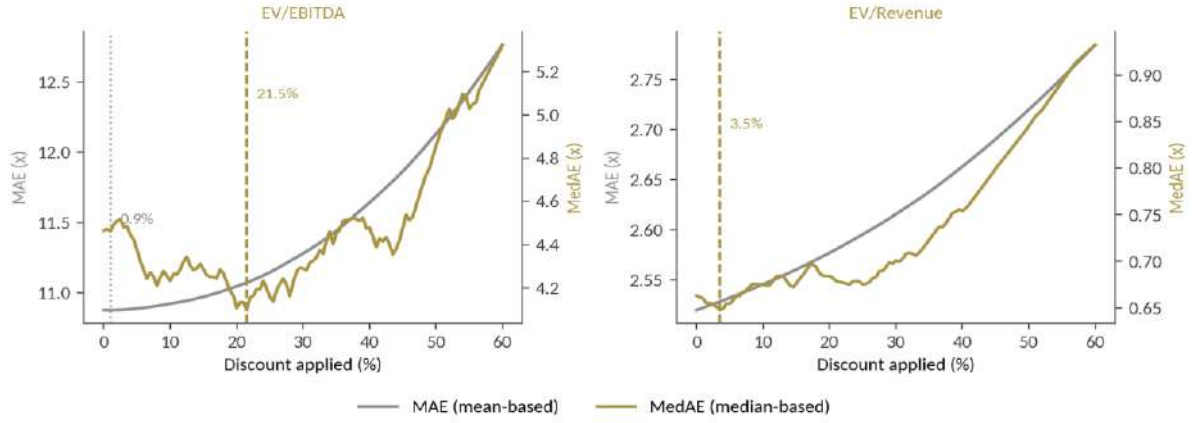
The Analyst Method

Table 9. Discount calibration under two error criteria

Multiple	Criterion	Calibrated discount	Training error	Adopted
EV/EBITDA	MAE (mean-based)	0.9%	MAE 10.876x	No
EV/EBITDA	**MedAE (median-based)**	**21.5%**	**MedAE 4.096x**	**Yes**
EV/Revenue	MAE (mean-based)	0.0%	MAE 2.520x	No
EV/Revenue	**MedAE (median-based)**	**3.5%**	**MedAE 0.647x**	**Yes**

Grid search over 601 candidate values from 0% to 60% in steps of 0.1%, computed on the training set only.

Figure 1. Discount calibration curves



Grey: MAE. Gold: MedAE. The MAE curve is minimised near zero because extreme-multiple transactions dominate the mean. The MedAE curve recovers a discount consistent with the PCD documented in the literature.

Table 10. Discount sensitivity on the test set (EV/EBITDA)

Discount applied	MAE	RMSE	MedAE
0% (no discount)	12.299	36.391	4.533
10%	12.267	36.598	3.909
21.5% (calibrated)	**12.373**	**36.867**	**4.072**
30%	12.556	37.085	3.802

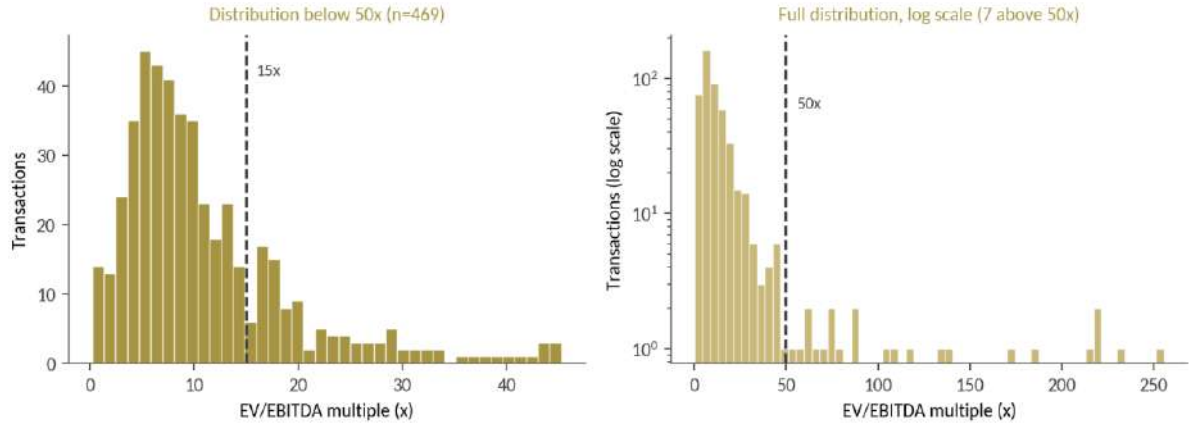
Aggregate accuracy is almost invariant to the discount, since the seven extreme transactions generate errors independent of it.

Table 11. Analyst benchmark test-set performance

Metric	EV/EBITDA (n=124)	EV/EBITDA excl. extremes (n=117)	EV/Revenue (n=124)
MAE	12.373	5.697	2.251
RMSE	36.867	9.002	6.167
MedAE	**4.072**	**3.808**	**0.576**

EV/Revenue contains no transaction above 50x, so full and excl.-extremes figures coincide.

Figure 2. Distribution of EV/EBITDA transaction multiples



Left: transactions below 50x. Right: full distribution on a logarithmic scale, showing the seven observations above 50x that dominate every mean-based error measure.

Table 12. Analyst benchmark by transaction bucket (EV/EBITDA)

Bucket	N	Share	Mean actual multiple	MAE
Typical (0-15x)	**97**	**78.2%**	**7.62x**	**3.33x**
Elevated (15-50x)	20	16.1%	24.4x	17.19x
Extreme (>50x)	7	5.6%	131.3x	123.96x

Table 13. Analyst benchmark by sector

Sector	N (EBITDA)	EBITDA MAE	N (Revenue)	Revenue MAE
Security and Alarm Services	5	**4.161**	6	3.037
Research and Consulting Services	35	4.995	35	2.514
Environmental and Facilities Services	23	6.498	22	**0.964**
Commercial Printing	6	8.040	6	1.700
Diversified Support Services	34	15.037	32	1.602
Human Resource and Employment Services	13	20.412	15	4.603
Office Services and Supplies	8	45.539	8	2.652

Office Services and Supplies falls to 9.33x on EV/EBITDA once the two transactions above 50x are excluded. Sectors with five or six test observations should be read as indicative.

Linear Regression

Table 14. Feature selection sequence and multicollinearity

Step	Action	Variables	Maximum VIF	Diagnosis
1	Initial specification	24	singular	Equity_ratio_CY1_missing perfectly collinear with Log_total_assets_CY1_missing
2	Drop the singular variable	23	145	CAGR horizon indicators near-perfectly explained by the missingness dummies
3	Drop both horizon indicators	21	34	Missingness dummies mutually redundant
4	**Drop all missingness dummies**	**11**	**3.78**	**Specification clean, all VIF below 4**

A VIF of 145 implies that 99.3% of the variance of that regressor is explained by the others, rendering its coefficient uninterpretable.

Table 15. OLS coefficients: EV/EBITDA, pure specification

Variable	Estimate	Std. error	t	p-value
Intercept	23.771	7.912	3.005	0.003 **
Log total assets	**−2.567**	**0.941**	**−2.729**	**0.007 ****
Log employees	1.754	1.040	1.686	0.093 *
CAGR revenue	0.815	2.451	0.333	0.740
CAGR EBITDA	0.050	0.664	0.075	0.940
Growth acceleration revenue	1.148	2.368	0.485	0.628
EBITDA margin	0.071	0.188	0.380	0.704
EBIT margin	−0.098	0.180	−0.547	0.585
Volatility EBITDA (log)	0.688	1.195	0.576	0.565
ROA	−4.529	5.193	−0.872	0.384
Equity ratio	3.817	6.514	0.586	0.558
Sponsor backed dummy	**8.726**	**2.839**	**3.073**	**0.002 ****

n = 369. Adjusted R^2 = 0.018. F-test p = 0.095. Breusch-Pagan p = 0.902, so conventional SE are valid. ** p < 0.01, * p < 0.10.

Table 16. OLS coefficients: EV/Revenue, pure specification, HC3 standard errors

Variable	Estimate	HC3 std. error	t	p-value
Intercept	1.961	1.993	0.984	0.326
Log total assets	0.059	0.216	0.272	0.786
Log employees	−0.927	1.030	−0.900	0.369
CAGR revenue	3.639	10.939	0.333	0.740
CAGR EBITDA	0.248	0.443	0.561	0.575
Growth acceleration revenue	2.878	3.752	0.767	0.443
EBITDA margin	−0.039	0.099	−0.396	0.693
EBIT margin	0.086	0.093	0.923	0.357
Volatility EBITDA (log)	0.545	0.774	0.704	0.482
ROA	−1.764	2.920	−0.604	0.546
Equity ratio	−0.659	1.639	−0.402	0.688
Sponsor backed dummy	**3.095**	**1.844**	**1.679**	**0.094**

n = 370. Adjusted R^2 = 0.028. F-test p = 0.031. Breusch-Pagan p = 0.051, borderline, so HC3 robust SE are reported. Under conventional SE, CAGR revenue, Growth acceleration revenue and the sponsor backed dummy are significant at 5% but none survives the robust correction.

Table 17. Regression specifications compared

Specification	Adj. R^2	F p-value	Significant at 5%	Breusch-Pagan p
OLS pure EV/EBITDA	0.018	0.095	Total assets, PE dummy	0.902
OLS augmented EV/EBITDA	0.029	0.040	Total assets, PE dummy, Stoxx 600	0.788
OLS pure EV/Revenue	0.028	0.031	None under HC3	0.051
OLS augmented EV/Revenue	0.022	0.076	CAGR revenue, Growth acceleration revenue, PE dummy	0.139

Augmentation raises explanatory power for EV/EBITDA and reduces it for EV/Revenue, the first appearance of the asymmetry between the two multiples.

Table 18. OLS and Ridge coefficients compared (EV/EBITDA)

Variable	OLS	Ridge	Sign change	OLS significance
Log total assets	**-2.567**	**-0.363**	**No**	**p = 0.007**
Log employees	1.754	0.187	No	p = 0.093
CAGR revenue	0.815	0.009	No	p = 0.740
CAGR EBITDA	0.050	-0.014	Yes	p = 0.940
Growth acceleration revenue	1.148	0.303	No	p = 0.628
EBITDA margin	0.071	-0.010	Yes	p = 0.704
EBIT margin	-0.098	-0.015	No	p = 0.585
Volatility EBITDA (log)	0.688	0.006	No	p = 0.565
ROA	-4.529	-1.046	No	p = 0.384
Equity ratio	3.817	0.595	No	p = 0.558
Sponsor backed dummy	**8.726**	**2.862**	**No**	**p = 0.002**

The two sign changes all occur on variables whose OLS coefficients are statistically indistinguishable from zero. Ridge shrinks noise through zero rather than correcting instability; the two significant variables retain their signs.

Table 19. Regression test-set performance

Method	EV/EBITDA MAE	RMSE	MedAE	EV/Revenue MAE	RMSE	MedAE
Analyst benchmark	**12.373**	**36.867**	**4.072**	**2.251**	**6.167**	**0.576**
OLS pure	14.213	35.299	7.996	3.096	5.743	1.682
OLS augmented	14.243	35.497	7.239	3.217	5.739	1.937
Ridge pure	14.340	35.433	7.805	2.917	5.765	2.089

Machine Learning

Table 20. Cross-validation results: all sixteen specifications

Multiple	Feature set	Model	CV MAE	Retained parameters
EV/EBITDA	**ML-A**	**RF**	**13.310**	mtry 4, min.node 20, depth 4
EV/EBITDA	ML-A	XGB	13.730	100 rounds, depth 2, eta 0.05, mcw 10
EV/EBITDA	ML-B	RF	13.455	mtry 8, min.node 20, depth 4
EV/EBITDA	ML-B	XGB	13.793	100 rounds, depth 2, eta 0.05, mcw 10
EV/EBITDA	Aug-A	RF	13.498	mtry 4, min.node 20, depth 4
EV/EBITDA	Aug-A	XGB	13.948	100 rounds, depth 2, eta 0.05, mcw 5
EV/EBITDA	Aug-B	RF	13.557	mtry 4, min.node 20, depth 4
EV/EBITDA	Aug-B	XGB	14.044	100 rounds, depth 2, eta 0.05, mcw 5
EV/Revenue	ML-A	RF	3.532	mtry 4, min.node 20, depth 4
EV/Revenue	ML-A	XGB	3.365	100 rounds, depth 2, eta 0.05, mcw 10
EV/Revenue	ML-B	RF	3.340	mtry 4, min.node 20, depth 4
EV/Revenue	ML-B	XGB	3.336	100 rounds, depth 2, eta 0.05, mcw 10
EV/Revenue	Aug-A	RF	3.485	mtry 4, min.node 20, depth 4
EV/Revenue	Aug-A	XGB	3.342	100 rounds, depth 2, eta 0.05, mcw 10
EV/Revenue	Aug-B	RF	3.340	mtry 4, min.node 20, depth 4
EV/Revenue	**Aug-B**	**XGB**	**3.313**	100 rounds, depth 2, eta 0.05, mcw 10

MAE across five entity-grouped folds on the training set. The eight EV/EBITDA specifications span only 0.73x, indicating that little learnable structure separates parameter choices. mcw = min_child_weight.

Table 21. ML test-set performance

Model	EV/EBITDA MedAE	EV/EBITDA MedAE excl. extremes	EV/Revenue MedAE
Analyst benchmark	**4.072**	**3.808**	**0.576**
RF ML-A	7.698	7.587	1.503
RF ML-B	7.617	7.261	1.636
XGB ML-A	7.935	7.363	1.526
XGB ML-B	7.926	7.642	1.504
RF Aug-A	7.166	6.923	1.510
RF Aug-B	7.410	7.162	1.587
XGB Aug-A	6.657	6.021	**1.462**
XGB Aug-B	**6.338**	**5.891**	1.476
OLS pure	7.996	7.603	1.682
OLS augmented	7.239	6.911	1.937

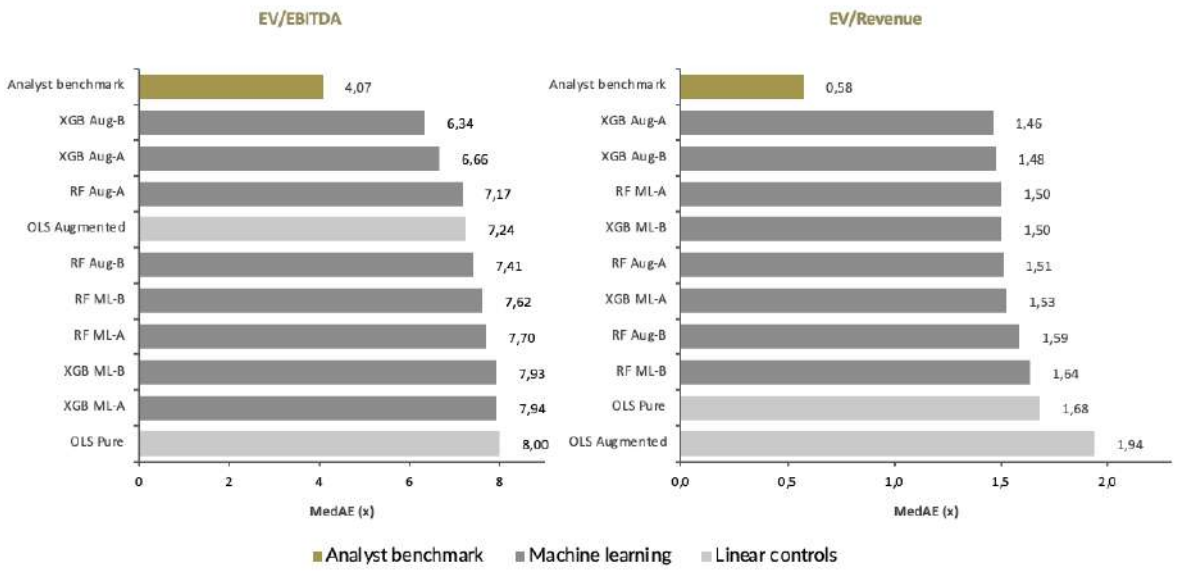
n = 124 for both multiples; excl. extremes n = 117 for EV/EBITDA.

Table 22. ML test-set performance: additional metrics

Model	EV/EBITDA MAE	EV/EBITDA RMSE	EV/Revenue MAE	EV/Revenue RMSE
Analyst benchmark	12.373	36.867	2.251	6.167
RF ML-A	13.964	34.305	3.650	8.734
RF ML-B	13.784	34.109	3.530	8.445
XGB ML-A	14.753	35.850	3.816	9.083
XGB ML-B	14.705	35.478	3.927	9.408
RF Aug-A	13.532	34.245	3.627	8.712
RF Aug-B	13.579	34.603	3.491	8.006
XGB Aug-A	13.599	34.036	3.915	9.296
XGB Aug-B	13.356	33.751	3.927	9.410
OLS pure	14.213	35.299	3.096	5.743
OLS augmented	14.243	35.497	3.217	5.739

n = 124 for both multiples

Figure 3. MedAE across all methods



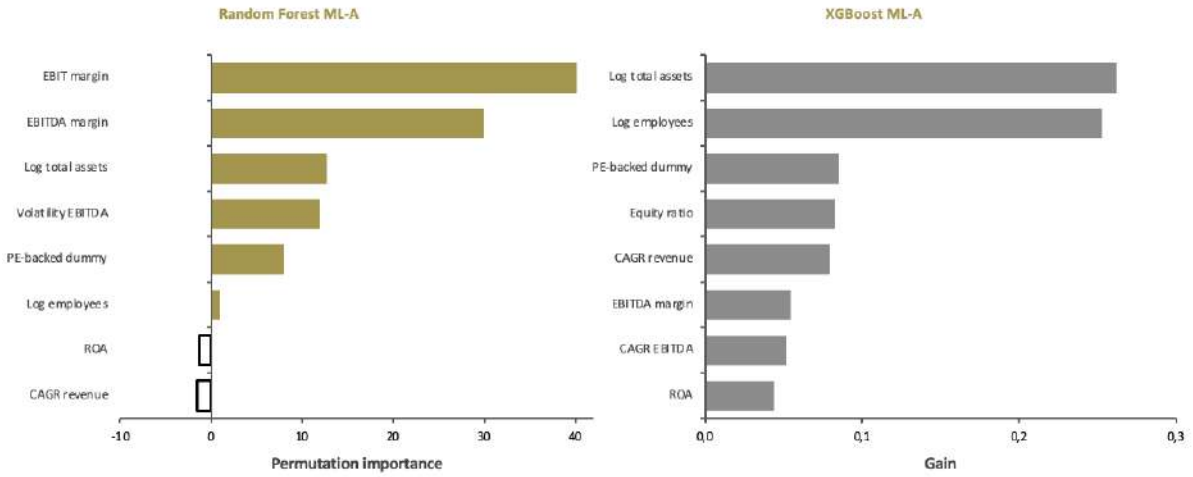
The benchmark leads on both multiples, and its margin is proportionally larger for EV/Revenue.

Table 23. Feature importance: EV/EBITDA, pure specifications

Rank	Random Forest ML-A	Permutation	XGBoost ML-A	Gain
1	EBIT margin	40.16	Log total assets	0.262
2	EBITDA margin	29.98	Log employees	0.253
3	Log total assets	12.66	Sponsor backed dummy	0.085
4	Volatility EBITDA	11.90	Equity ratio	0.082
5	Sponsor backed dummy	7.97	CAGR revenue	0.079
6	Log employees	1.01	EBITDA margin	0.054
7	ROA	−1.36	CAGR EBITDA	0.052
8	CAGR revenue	−1.57	ROA	0.044

The two algorithms disagree on the primary driver: RF ranks profitability margins first, XGB ranks size first. The same disagreement recurs on EV/Revenue (Table 25), so it reflects the difference between permutation and gain rather than anything specific to EV/EBITDA. Negative permutation importance indicates a variable whose removal would improve the model.

Figure 4. Feature importance: EV/EBITDA, pure specifications



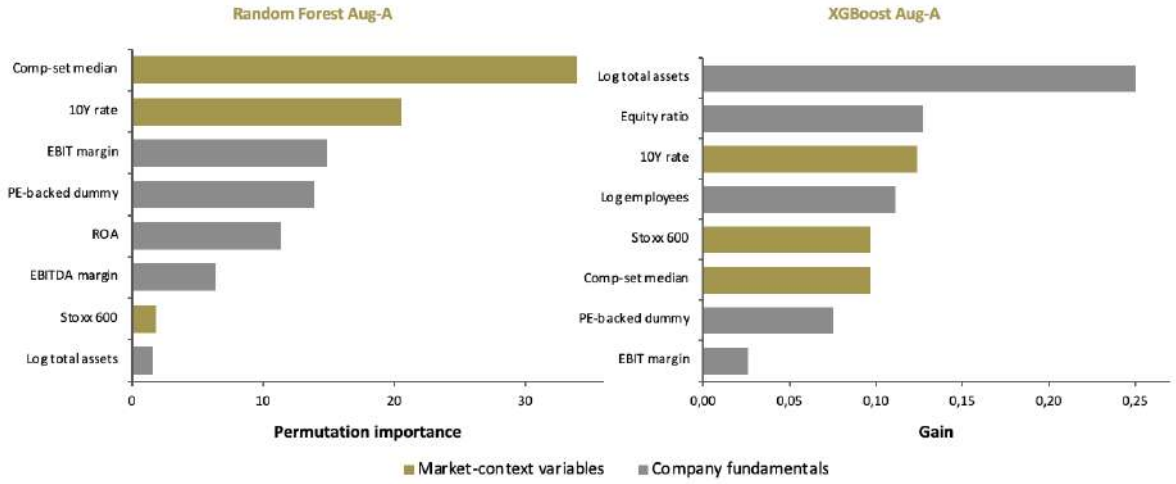
The divergence between the two rankings reflects the difference between permutation importance and gain, not only a property of EV/EBITDA: the same divergence recurs on EV/Revenue (Figure 6).

Table 24. Feature importance: EV/EBITDA, augmented specifications

Rank	Random Forest Aug-A	Permutation	XGBoost Aug-A	Gain
1	**Comps Median EV/EBITDA**	**33.93**	Log total assets	0.250
2	**Rate 10Y at closing**	**20.55**	Equity ratio	0.127
3	EBIT margin	14.90	**Rate 10Y at closing**	**0.124**
4	Sponsor backed dummy	13.95	Log employees	0.111
5	ROA	11.35	**STOXX600 at closing**	**0.097**
6	EBITDA margin	6.42	**Comps Median EV/EBITDA**	**0.097**
7	**STOXX600 at closing**	**1.88**	Sponsor backed dummy	0.075
8	Log total assets	1.57	EBIT margin	0.026

Market-context variables are shaded. In RF the comp-set median becomes the single most important variable; in XGB the three market variables together account for roughly a third of total gain.

Figure 5. Feature importance: EV/EBITDA, augmented specifications



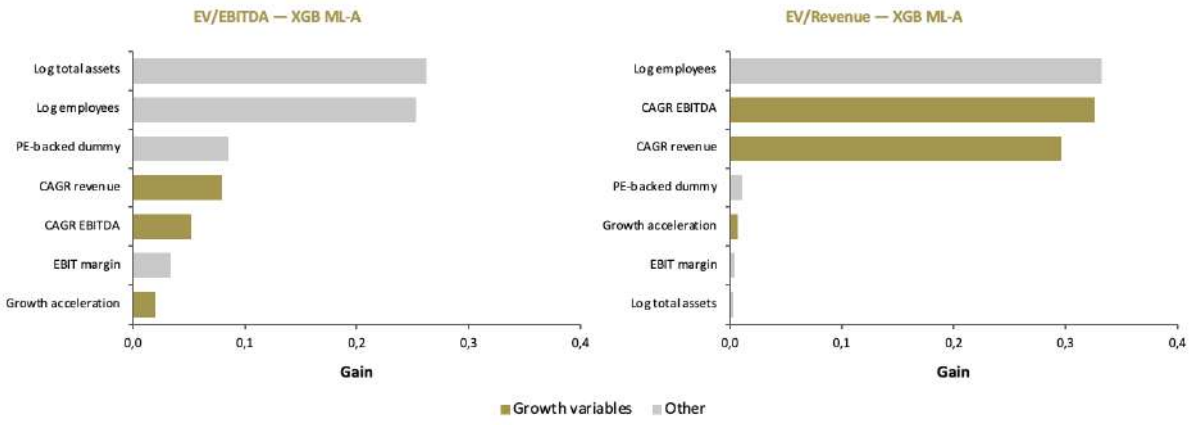
Gold bars denote the three market-context variables. Both algorithms allocate substantial weight to them, which is the mechanism behind the 20% improvement of augmented over pure XGB.

Table 25. Feature importance: EV/Revenue compared with EV/EBITDA

Variable	XGB EV/Revenue gain	XGB EV/EBITDA gain	Direction
Log employees	**0.332**	**0.253**	Important for both
CAGR EBITDA	**0.326**	0.052	Revenue only
CAGR revenue	**0.296**	0.079	Revenue only
Log total assets	0.003	**0.262**	EBITDA only
Sponsor backed dummy	0.011	**0.085**	EBITDA only
Growth acceleration revenue	0.007	0.020	Marginal for both
EBIT margin	0.004	0.034	Marginal for both

Pure ML-A specifications, XGB gain only. Growth and scale account for 95% of gain for EV/Revenue; size and deal type dominate for EV/EBITDA. RF permutation importance does not reproduce this split: it ranks the margin variables first for both multiples and returns negative importance for both CAGR variables on EV/Revenue.

Figure 6. The asymmetry between the two multiples



The three growth variables carry 63% of XGB's gain for EV/Revenue and 15% of it for EV/EBITDA. RF permutation importance does not reproduce the pattern: it ranks margins first for both multiples.

Comparison and Analysis of All Methods

Table 26. Master comparison: all methods and both multiples

Method	EBITDA MedAE	EBITDA MedAE excl. extremes	Revenue MedAE
Analyst benchmark	***4.072***	***3.808***	***0.576***
OLS pure	7.996	7.603	1.682
OLS augmented	7.239	6.911	1.937
RF ML-A	7.698	7.587	1.503
RF ML-B	7.617	7.261	1.636
XGB ML-A	7.935	7.363	1.526
XGB ML-B	7.926	7.642	1.504
RF Aug-A	7.166	6.923	1.510
RF Aug-B	7.410	7.162	1.587
XGB Aug-A	6.657	6.021	**1.462**
XGB Aug-B	**6.338**	**5.891**	1.476

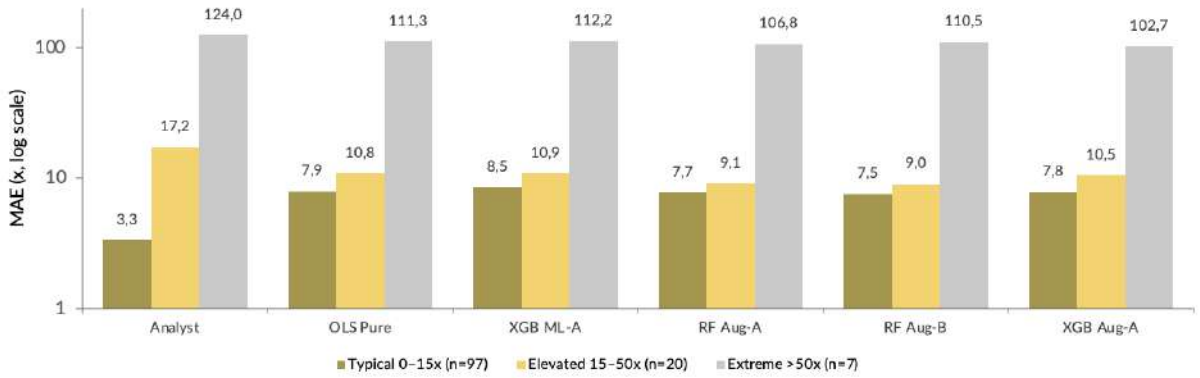
The residual gap on EV/EBITDA is approximately 2.3x, and it survives the removal of extreme transactions.

Table 27. Bucket decomposition: all methods (EV/EBITDA)

Method	Typical 0-15x (n=97)	Elevated 15-50x (n=20)	Extreme >50x (n=7)
Analyst benchmark	***3.33***	*17.19*	*123.96*
OLS pure	7.91	10.81	111.26
OLS augmented	7.91	11.36	110.17
RF ML-A	7.89	10.96	106.78
RF ML-B	7.58	11.31	106.77
XGB ML-A	8.52	10.88	112.21
XGB ML-B	8.39	11.90	110.26
RF Aug-A	7.72	9.11	106.77
RF Aug-B	7.54	**8.98**	110.46
XGB Aug-A	7.80	10.53	**102.68**
XGB Aug-B	7.41	10.76	103.20

MAE within each bucket. The benchmark leads on typical transactions and trails on elevated transactions by roughly 47%. No specification leads in both.

Figure 7. Performance by transaction bucket



Logarithmic vertical scale. The benchmark leads on typical transactions and trails on elevated ones; on extreme transactions all methods fail at a comparable level.

Table 28. Bucket decomposition (EV/Revenue)

Method	Typical 0-15x (n=120)	Elevated 15-50x (n=4)
Analyst benchmark	***1.36***	*29.06*
OLS pure	2.36	25.14
OLS augmented	2.51	**24.47**
RF ML-A	2.88	26.78
RF ML-B	2.76	26.78
XGB ML-A	2.99	28.44
XGB ML-B	3.13	27.77
RF Aug-A	2.87	26.42
RF Aug-B	2.72	26.67
XGB Aug-A	3.11	28.08
XGB Aug-B	3.14	27.40

No EV/Revenue transaction exceeds 50x, so no extreme bucket exists. Within the typical bucket OLS pure is the most accurate statistical model on MAE, reversing the ranking observed on MedAE.

Glossary

Table 29. Glossary of abbreviations

Abbreviation	Term
Aug-A / Aug-B	Augmented feature sets
CAGR	Compound annual growth rate
CCA	Comparable company analysis
CV	Cross-validation
CY	Closing year
DCF	Discounted cash flow
EBIT	Earnings before interest and taxes
EBITDA	Earnings before interest, taxes, depreciation and amortisation
EV	Enterprise value
FCFF	Free cash flow to the firm
GICS	Global Industry Classification Standard
HC3	Heteroscedasticity-consistent estimator, type 3
IQR	Interquartile range
LBO	Leveraged buyout
M&A	Mergers and acquisitions
MAE	Mean absolute error
MedAE	Median absolute error
MedAPE	Median absolute percentage error
ML	Machine learning
ML-A / ML-B	Pure feature sets
OLS	Ordinary least squares
PCD	Private company discount
PE	Private equity
RF	Random Forest
RMSE	Root mean squared error
ROA	Return on assets
ROE	Return on equity
ROIC	Return on invested capital
SE	Standard errors
SIC	Standard Industrial Classification
SME	Small and medium-sized enterprise
TV	Terminal value
VIF	Variance inflation factor
WACC	Weighted average cost of capital
XGB	Extreme gradient boosting (XGBoost)

R Markdown

Preprocessing

Four of the ten quantitative variables were transformed. The screen was whether an extreme value reflected the economics the variable is meant to capture, or arose from the arithmetic of a ratio whose denominator approached zero. Growth acceleration reaches -572, a swing of 57,200 percentage points between two growth rates, and EBITDA CAGR runs from -44.4 to +47.8: in both cases the inputs are accurate but a base-year figure near zero inflates the ratio far beyond the range the variable is meant to measure. Volatility EBITDA reaches 188,911 against a median of 699, the right skew a dispersion measure produces by construction. The equity ratio falls to -5.02, six times more liabilities than assets, where the next worst case is -1.66. The six left untouched carry extremes that are rare but interpretable. Revenue CAGR is the telling comparison: it belongs to the same family as EBITDA CAGR but revenue cannot be negative, so it is floored at -1 by construction, and its minimum of -0.96 sits at that floor.

Treatment then follows from the sign. Volatility EBITDA is strictly positive, so the log applies, and it came before any capping since winsorising first would set thresholds on a scale the log discards. The other three contain economically real negative values, so the log is undefined and the 1st and 99th percentiles of the training set were used.

The work was done in Excel. “n.s.” entries cleared with Find and Replace; CAGR fallback through nested IF with a parallel horizon column; LN for volatility; PERCENTILE.INC with MIN and MAX for the caps; QUARTILE.INC, MEDIAN and IF with ISBLANK for the IQR-trimmed sector medians and the missingness dummies. Caps and medians carried unchanged to the test and augmented sheets. Table 7 reports the ranges, raw and post-treatment, the four clipped observations per tail counted by COUNTIF against each threshold, and the Security and Alarm imputation example quoted in the methodology read off the same sheet. The equity ratio was the exception, reaching R untransformed.

Setup

```
library(readxl)
library(car)
library(lmtest)
library(sandwich)
library(glmnet)
library(ranger)
library(xgboost)
library(corrplot)

DATA <- "~/Documents/MASTER/MEMOIRE/Data Fully reconciled/"
RES_ML <- "~/Documents/MASTER/MEMOIRE/Results/ML/"
RES_IMP <- "~/Documents/MASTER/MEMOIRE/Results/ML/Importance feature/"
RES_LR <- "~/Documents/MASTER/MEMOIRE/Results/Linear regressions/"

# training sets only, the test files stay shut until everything is decided
ebitda_train <- read_excel(
  paste0(DATA, "Reconciliation_ML_Train_v2.xlsx"),
  sheet = "EBITDA_Train")
revenue_train <- read_excel(
  paste0(DATA, "Reconciliation_ML_Train_v2.xlsx"),
  sheet = "Revenue_Train")

cat("EBITDA train:", nrow(ebitda_train), "rows,",
    ncol(ebitda_train), "cols\n")
EBITDA train: 369 rows, 32 cols
cat("Revenue train:", nrow(revenue_train), "rows,",
    ncol(revenue_train), "cols\n")
Revenue train: 370 rows, 32 cols
```

Candidate variables

```
exclude_cols <- c("Transaction_ID", "BvD_ID", "Target_Name", "Sector",
                  "Closing_Date", "Closing_Year",
                  "Transaction_EV_EBITDA", "Transaction_EV_Revenue")

feature_cols <- setdiff(names(ebitda_train), exclude_cols)
cat("Features:", length(feature_cols), "\n")
Features: 24
print(feature_cols)
 [1] "Log_total_assets_CY1"          "Log_total_assets_CY1_missing"
 [3] "Log_employees_CY1"             "Log_employees_CY1_missing"
 [5] "CAGR_revenue_fallback"         "CAGR_revenue_horizon_years"
 [7] "CAGR_revenue_fallback_missing" "CAGR_EBITDA_fallback"
 [9] "CAGR_EBITDA_horizon_years"     "CAGR_EBITDA_fallback_missing"
[11] "Growth_accel_revenue"          "Growth_accel_revenue_missing"
[13] "EBITDA_margin_CY1"             "EBITDA_margin_CY1_missing"
[15] "EBIT_margin_CY1"               "EBIT_margin_CY1_missing"
[17] "Volatility_EBITDA"             "Volatility_EBITDA_missing"
[19] "ROA_CY1"                       "ROA_CY1_missing"
[21] "Equity_ratio_CY1"              "Equity_ratio_CY1_missing"
[23] "Sponsor_backed_dummy"          "Sponsor_backed_missing"

# check the treated variables came through with the ranges they should
cat("Volatility_EBITDA - min:",
    round(min(ebitda_train$Volatility_EBITDA), 3),
    "max:", round(max(ebitda_train$Volatility_EBITDA), 3), "\n")
Volatility_EBITDA - min: 2.062 max: 12.149
cat("Growth_accel_revenue - min:",
    round(min(ebitda_train$Growth_accel_revenue), 3),
    "max:", round(max(ebitda_train$Growth_accel_revenue), 3), "\n")
Growth_accel_revenue - min: -6.632 max: 1.918
cat("CAGR_EBITDA_fallback - min:",
    round(min(ebitda_train$CAGR_EBITDA_fallback), 3),
    "max:", round(max(ebitda_train$CAGR_EBITDA_fallback), 3), "\n")
CAGR_EBITDA_fallback - min: -17.716 max: 3.022
cat("Equity_ratio_CY1 - min:",
    round(min(ebitda_train$Equity_ratio_CY1), 3),
    "max:", round(max(ebitda_train$Equity_ratio_CY1), 3), "\n")
Equity_ratio_CY1 - min: -0.527 max: 0.981
summary(ebitda_train[, feature_cols])
```

```
 Log_total_assets_CY1 Log_total_assets_CY1_missing Log_employees_CY1
 Min.   :-6.140       Min.   :0.0000               Min.   : 0.000
 1st Qu.: 8.573       1st Qu.:0.0000               1st Qu.: 3.638
 Median : 9.624       Median :0.0000               Median : 4.625
 Mean   : 9.824       Mean   :0.2276               Mean   : 4.754
 3rd Qu.:11.065       3rd Qu.:0.0000               3rd Qu.: 5.690
 Max.   :16.164       Max.   :1.0000               Max.   :11.922
 Log_employees_CY1_missing CAGR_revenue_fallback CAGR_revenue_horizon_years
 Min.   :0.0000            Min.   :-0.96152      Min.   :0.000
 1st Qu.:0.0000            1st Qu.: 0.02221      1st Qu.:0.000
 Median :0.0000            Median : 0.08058      Median :3.000
 Mean   :0.3035            Mean   : 0.15902      Mean   :1.721
 3rd Qu.:1.0000            3rd Qu.: 0.11023      3rd Qu.:3.000
 Max.   :1.0000            Max.   :12.85712      Max.   :3.000
 CAGR_revenue_fallback_missing CAGR_EBITDA_fallback CAGR_EBITDA_horizon_years
 Min.   :0.0000                Min.   :-17.71577    Min.   :0.000
 1st Qu.:0.0000                1st Qu.: -0.01927    1st Qu.:0.000
 Median :0.0000                Median :  0.08027    Median :2.000
 Mean   :0.3604                Mean   : -0.13202    Mean   :1.604
 3rd Qu.:1.0000                3rd Qu.:  0.19999    3rd Qu.:3.000
 Max.   :1.0000                Max.   :  3.02228    Max.   :3.000
 CAGR_EBITDA_fallback_missing Growth_accel_revenue Growth_accel_revenue_missing
 Min.   :0.0000               Min.   :-6.631853    Min.   :0.0000
 1st Qu.:0.0000               1st Qu.:-0.081457    1st Qu.:0.0000
 Median :0.0000               Median :-0.046815    Median :0.0000
 Mean   :0.3794               Mean   :-0.104758    Mean   :0.4959
 3rd Qu.:1.0000               3rd Qu.: 0.006575    3rd Qu.:1.0000
 Max.   :1.0000               Max.   : 1.918132    Max.   :1.0000
 EBITDA_margin_CY1 EBITDA_margin_CY1_missing EBIT_margin_CY1   EBIT_margin_CY1_missing
 Min.   :-64.228   Min.   :0.0000            Min.   :-81.148   Min.   :0.0000
 1st Qu.:  5.968   1st Qu.:0.0000            1st Qu.:  3.822   1st Qu.:0.0000
 Median : 11.472   Median :0.0000            Median :  6.945   Median :0.0000
 Mean   : 12.352   Mean   :0.3902            Mean   :  8.164   Mean   :0.3821
 3rd Qu.: 13.513   3rd Qu.:1.0000            3rd Qu.:  9.148   3rd Qu.:1.0000
 Max.   : 95.445   Max.   :1.0000            Max.   :100.000   Max.   :1.0000
 Volatility_EBITDA Volatility_EBITDA_missing    ROA_CY1         ROA_CY1_missing
 Min.   : 2.062    Min.   :0.0000            Min.   :-1.00980   Min.   :0.0000
 1st Qu.: 6.008    1st Qu.:0.0000            1st Qu.: 0.04366   1st Qu.:0.0000
 Median : 6.549    Median :0.0000            Median : 0.06905   Median :0.0000
 Mean   : 6.768    Mean   :0.4201            Mean   : 0.10409   Mean   :0.3388
 3rd Qu.: 7.354    3rd Qu.:1.0000            3rd Qu.: 0.10457   3rd Qu.:1.0000
 Max.   :12.149    Max.   :1.0000            Max.   : 3.14211   Max.   :1.0000
 Equity_ratio_CY1  Equity_ratio_CY1_missing Sponsor_backed_dummy Sponsor_backed_missing
 Min.   :-0.5270   Min.   :0.0000           Min.   :0.0000       Min.   :0.00000
 1st Qu.: 0.2822   1st Qu.:0.0000           1st Qu.:0.0000       1st Qu.:0.00000
 Median : 0.3965   Median :0.0000           Median :0.0000       Median :0.00000
 Mean   : 0.4097   Mean   :0.2276           Mean   :0.3957       Mean   :0.00271
 3rd Qu.: 0.5249   3rd Qu.:0.0000           3rd Qu.:1.0000       3rd Qu.:0.00000
 Max.   : 0.9815   Max.   :1.0000           Max.   :1.0000       Max.   :1.00000
```

Equity ratio

The equity ratio was the last variable treated. Its raw minimum of -5.02, six times more liabilities than assets against a next worst case of -1.66, was capped in Excel at the training-set 1st and 99th percentiles alongside the two growth variables. This block confirms the range on import and lists the negative-equity observations that motivated it. No capping is applied here; the file already arrives at [-0.527, 0.981].

```
cat("Equity_ratio_CY1 range:",
    round(range(ebitda_train$Equity_ratio_CY1), 3), "\n")
Equity_ratio_CY1 range: -0.527 0.981
cat("negative equity ratios:",
    sum(ebitda_train$Equity_ratio_CY1 < 0), "\n")
negative equity ratios: 9

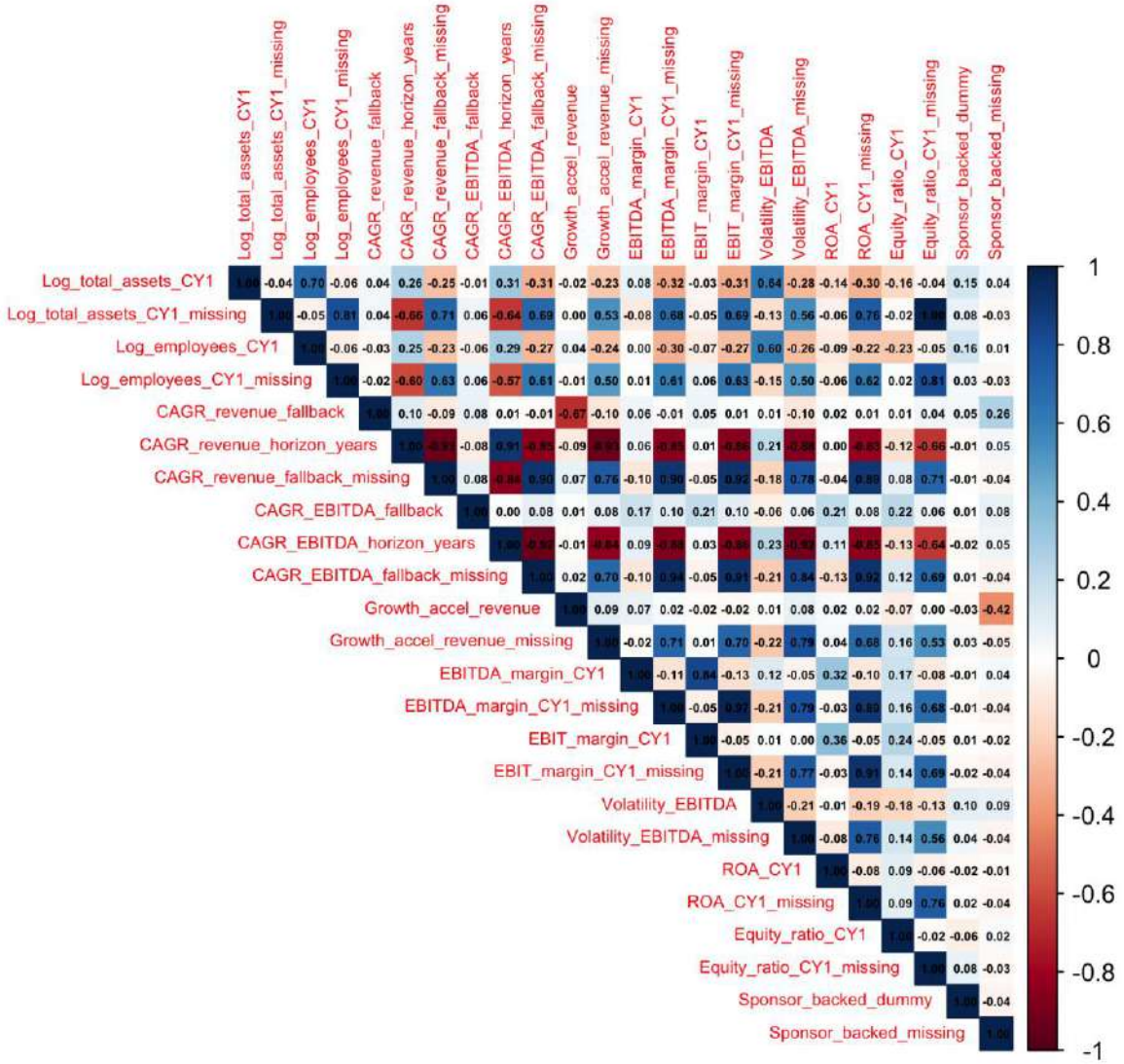
# the nine cases behind the treatment, four of them now sitting on the
# lower cap
ebitda_train[ebitda_train$Equity_ratio_CY1 < 0,
             c("Transaction_ID", "Target_Name", "Equity_ratio_CY1")]
```

```
# A tibble: 9 × 3
  Transaction_ID Target_Name                                    Equity_ratio_CY1
  <chr>          <chr>                                                     <dbl>
1 SPTRD352419    1st Credit Limited                                      -0.117
2 SPTRD820510    Asons Solicitors Limited                                -0.219
3 SPTRD830104    Centrum Wspierania Projektów Europejskich S.A.          -0.00658
4 SPTRD2107496   Full Circle Group Holding B.V.                          -0.160
5 SPTRD828079    ICT Education AB                                        -0.527
6 SPTRD369873    Synarbor Limited                                        -0.527
7 SPTRD2112154   Trilogy Consultants International Limited               -0.527
8 SPTRD2041884   Trios Group Limited                                     -0.527
9 SPTRD2107497   Visegurity Express Sl                                   -0.0470
```

Correlation matrix

```
X_train <- ebitda_train[, feature_cols]
cor_matrix <- cor(X_train, use = "complete.obs")

corrplot(cor_matrix, method = "color", type = "upper", tl.cex = 0.55,
         addCoef.col = "black", number.cex = 0.4, mar = c(0, 0, 2, 0))
```



```
high_cor <- which(abs(cor_matrix) > 0.80 & abs(cor_matrix) < 1,
                  arr.ind = TRUE)
high_cor_df <- data.frame(
  Var1 = rownames(cor_matrix)[high_cor[, 1]],
  Var2 = colnames(cor_matrix)[high_cor[, 2]],
  Correlation = round(cor_matrix[high_cor], 3))
high_cor_df <- high_cor_df[high_cor_df$Var1 < high_cor_df$Var2, ]

print(high_cor_df)
```

	Var1	Var2	Correlation
1	Log_employees_CY1_missing	Log_total_assets_CY1_missing	0.808
3	Equity_ratio_CY1_missing	Log_employees_CY1_missing	0.808
4	CAGR_revenue_fallback_missing	CAGR_revenue_horizon_years	-0.930
5	CAGR_EBITDA_horizon_years	CAGR_revenue_horizon_years	0.905
6	CAGR_EBITDA_fallback_missing	CAGR_revenue_horizon_years	-0.848
13	CAGR_EBITDA_horizon_years	CAGR_revenue_fallback_missing	-0.841
14	CAGR_EBITDA_fallback_missing	CAGR_revenue_fallback_missing	0.902
20	CAGR_EBITDA_fallback_missing	CAGR_EBITDA_horizon_years	-0.924
33	CAGR_revenue_horizon_years	Growth_accel_revenue_missing	-0.928
34	CAGR_EBITDA_horizon_years	Growth_accel_revenue_missing	-0.837
35	EBIT_margin_CY1	EBITDA_margin_CY1	0.837
36	CAGR_revenue_horizon_years	EBITDA_margin_CY1_missing	-0.851
37	CAGR_revenue_fallback_missing	EBITDA_margin_CY1_missing	0.904
38	CAGR_EBITDA_horizon_years	EBITDA_margin_CY1_missing	-0.880
39	CAGR_EBITDA_fallback_missing	EBITDA_margin_CY1_missing	0.943
40	EBIT_margin_CY1_missing	EBITDA_margin_CY1_missing	0.972
43	CAGR_revenue_horizon_years	EBIT_margin_CY1_missing	-0.858
44	CAGR_revenue_fallback_missing	EBIT_margin_CY1_missing	0.920
45	CAGR_EBITDA_horizon_years	EBIT_margin_CY1_missing	-0.855
46	CAGR_EBITDA_fallback_missing	EBIT_margin_CY1_missing	0.914
49	CAGR_revenue_horizon_years	Volatility_EBITDA_missing	-0.876
50	CAGR_EBITDA_horizon_years	Volatility_EBITDA_missing	-0.917
51	CAGR_EBITDA_fallback_missing	Volatility_EBITDA_missing	0.840
52	CAGR_revenue_horizon_years	ROA_CY1_missing	-0.829
53	CAGR_revenue_fallback_missing	ROA_CY1_missing	0.894
54	CAGR_EBITDA_horizon_years	ROA_CY1_missing	-0.846
55	CAGR_EBITDA_fallback_missing	ROA_CY1_missing	0.915
56	EBITDA_margin_CY1_missing	ROA_CY1_missing	0.895
57	EBIT_margin_CY1_missing	ROA_CY1_missing	0.910

Regression feature selection

Step 1: the singular variable

```
formula_obj <- as.formula(paste("Transaction_EV_EBITDA ~",
                                paste(feature_cols, collapse = " + ")))
model_ols <- lm(formula_obj, data = ebitda_train)
summary(model_ols)
```

```
Call:
lm(formula = formula_obj, data = ebitda_train)

Residuals:
    Min      1Q  Median      3Q     Max
-26.637 -10.966  -5.734   2.086 207.721

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   8.295096  35.875252   0.231  0.81728
Log_total_assets_CY1         -1.972746   1.049090  -1.880  0.06089 .
Log_total_assets_CY1_missing -3.995652   7.426934  -0.538  0.59093
Log_employees_CY1             1.659129   1.083837   1.531  0.12674
Log_employees_CY1_missing     1.322877   5.588697   0.237  0.81303
CAGR_revenue_fallback         0.990014   2.615180   0.379  0.70524
CAGR_revenue_horizon_years    4.029466  11.947507   0.337  0.73612
CAGR_revenue_fallback_missing 1.534018  22.857738   0.067  0.94653
CAGR_EBITDA_fallback         -0.003193   0.727887  -0.004  0.99650
CAGR_EBITDA_horizon_years    -0.422685   5.294199  -0.080  0.93641
CAGR_EBITDA_fallback_missing -10.406967  17.119617  -0.608  0.54366
Growth_accel_revenue          1.519013   2.666918   0.570  0.56934
Growth_accel_revenue_missing  2.650245  14.760877   0.180  0.85762
EBITDA_margin_CY1             0.059061   0.196189   0.301  0.76357
EBITDA_margin_CY1_missing    -0.918606  16.433036  -0.056  0.95545
EBIT_margin_CY1              -0.093179   0.187053  -0.498  0.61870
EBIT_margin_CY1_missing       3.354288  15.984503   0.210  0.83391
Volatility_EBITDA             0.437664   1.270870   0.344  0.73077
Volatility_EBITDA_missing     9.571846   9.857423   0.971  0.33221
ROA_CY1                      -3.167996   5.784382  -0.548  0.58427
ROA_CY1_missing              11.261620   9.762377   1.154  0.24947
Equity_ratio_CY1              2.552801   6.794874   0.376  0.70737
Equity_ratio_CY1_missing            NA         NA      NA       NA
Sponsor_backed_dummy          8.291992   2.918271   2.841  0.00476 **
Sponsor_backed_missing        8.561498  30.531950   0.280  0.77933
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.49 on 345 degrees of freedom
Multiple R-squared:  0.0588,    Adjusted R-squared:  -0.003952
F-statistic: 0.937 on 23 and 345 DF,  p-value: 0.5485

# one coefficient comes back NA, so check what it is collinear with
table(ebitda_train$Equity_ratio_CY1_missing)

  0   1
285  84
cor(ebitda_train$Equity_ratio_CY1_missing,
    ebitda_train[, feature_cols[feature_cols !=
                                "Equity_ratio_CY1_missing"]])
     Log_total_assets_CY1 Log_total_assets_CY1_missing Log_employees_CY1
[1,]          -0.03549011                           1       -0.05010432
     Log_employees_CY1_missing CAGR_revenue_fallback CAGR_revenue_horizon_years
[1,]                 0.8083273            0.04215349                 -0.6584968
     CAGR_revenue_fallback_missing CAGR_EBITDA_fallback CAGR_EBITDA_horizon_years
[1,]                     0.7097205           0.0567287                -0.6415003
     CAGR_EBITDA_fallback_missing Growth_accel_revenue Growth_accel_revenue_missing
[1,]                     0.694338         -0.002704262                    0.5344021
     EBITDA_margin_CY1 EBITDA_margin_CY1_missing EBIT_margin_CY1 EBIT_margin_CY1_missing
[1,]       -0.08334811                 0.6786209     -0.04504653               0.6903591
     Volatility_EBITDA Volatility_EBITDA_missing     ROA_CY1 ROA_CY1_missing
[1,]        -0.1313639                 0.5593408 -0.05974216       0.7585027
     Equity_ratio_CY1 Sponsor_backed_dummy Sponsor_backed_missing
[1,]      -0.01515372           0.07618624            -0.02830045

# correlation of exactly 1 with Log_total_assets_CY1_missing, so drop it
feature_cols <- feature_cols[feature_cols != "Equity_ratio_CY1_missing"]

formula_obj <- as.formula(paste("Transaction_EV_EBITDA ~",
                                paste(feature_cols, collapse = " + ")))
model_ols <- lm(formula_obj, data = ebitda_train)
cat("Adj R2:", round(summary(model_ols)$adj.r.squared, 4), "\n")
Adj R2: -0.004
print(summary(model_ols)$coefficients)
```

```
                                Estimate Std. Error      t value    Pr(>|t|)
(Intercept)                  8.295095763 35.8752519  0.231220558 0.817280527
Log_total_assets_CY1        -1.972745778  1.0490895 -1.880436054 0.060890909
Log_total_assets_CY1_missing -3.995651990  7.4269336 -0.537994844 0.590927494
Log_employees_CY1            1.659129192  1.0838368  1.530792408 0.126737096
Log_employees_CY1_missing    1.322877053  5.5886973  0.236705799 0.813025516
CAGR_revenue_fallback        0.990013871  2.6151803  0.378564291 0.705244289
CAGR_revenue_horizon_years   4.029466331 11.9475070  0.337264196 0.736122838
CAGR_revenue_fallback_missing 1.534017857 22.8577375  0.067111535 0.946531786
CAGR_EBITDA_fallback        -0.003193469  0.7278867 -0.004387315 0.996501976
CAGR_EBITDA_horizon_years   -0.422685499  5.2941990 -0.079839367 0.936411305
CAGR_EBITDA_fallback_missing -10.406967079 17.1196170 -0.607897191 0.543655612
Growth_accel_revenue         1.519012556  2.6669176  0.569576122 0.569335894
Growth_accel_revenue_missing 2.650245180 14.7608769  0.179545240 0.857615001
EBITDA_margin_CY1            0.059060891  0.1961893  0.301040394 0.763565041
EBITDA_margin_CY1_missing   -0.918605709 16.4330363 -0.055899938 0.955453879
EBIT_margin_CY1             -0.093178781  0.1870528 -0.498141701 0.618701565
EBIT_margin_CY1_missing      3.354287809 15.9845033  0.209846233 0.833911553
Volatility_EBITDA            0.437663659  1.2708704  0.344381025 0.730769577
Volatility_EBITDA_missing    9.571846189  9.8574227  0.971029295 0.332213970
ROA_CY1                     -3.167996436  5.7843824 -0.547681015 0.584265093
ROA_CY1_missing             11.261619826  9.7623771  1.153573537 0.249473593
Equity_ratio_CY1             2.552801337  6.7948744  0.375695147 0.707374440
Sponsor_backed_dummy         8.291991933  2.9182706  2.841406081 0.004758433
Sponsor_backed_missing       8.561498380 30.5319500  0.280411123 0.779330173
```

Step 2: the horizon indicators

```
vif_values <- vif(model_ols)
print(sort(vif_values, decreasing = TRUE))
```

```
   CAGR_revenue_horizon_years CAGR_revenue_fallback_missing  CAGR_EBITDA_fallback_missing
                      144.927275                     63.353010                     36.298505
    EBITDA_margin_CY1_missing       EBIT_margin_CY1_missing Growth_accel_revenue_missing
                       33.800083                     31.731411                     28.650056
    CAGR_EBITDA_horizon_years     Volatility_EBITDA_missing                ROA_CY1_missing
                       27.178267                     12.451147                     11.229182
 Log_total_assets_CY1_missing             EBITDA_margin_CY1              EBIT_margin_CY1
                        5.101294                      3.991775                      3.990711
    Log_employees_CY1_missing          Log_total_assets_CY1          Growth_accel_revenue
                        3.473037                      2.935439                      2.455427
            Log_employees_CY1         CAGR_revenue_fallback             Volatility_EBITDA
                        2.332599                      2.153164                      2.106273
                      ROA_CY1          CAGR_EBITDA_fallback        Sponsor_backed_missing
                        1.489187                      1.333023                      1.325238
             Equity_ratio_CY1          Sponsor_backed_dummy
                        1.252654                      1.071139
```

```
# the two horizon variables come out at 145 and 27, far above the usual
# threshold of 10, so both go
feature_cols_clean <- feature_cols[!feature_cols %in%
                                   c("CAGR_revenue_horizon_years",
                                     "CAGR_EBITDA_horizon_years")]

formula_clean <- as.formula(paste("Transaction_EV_EBITDA ~",
                                  paste(feature_cols_clean,
                                        collapse = " + ")))
model_ols2 <- lm(formula_clean, data = ebitda_train)
cat("Adj R2:", round(summary(model_ols2)$adj.r.squared, 4), "\n")
```

```
Adj R2: 0.0015
```

Step 3: the missingness dummies

```
vif2 <- vif(model_ols2)
print(sort(vif2, decreasing = TRUE))
```

```
    EBITDA_margin_CY1_missing       EBIT_margin_CY1_missing  CAGR_EBITDA_fallback_missing
                     33.718189                     31.344626                     19.429881
            ROA_CY1_missing CAGR_revenue_fallback_missing     Volatility_EBITDA_missing
                     11.192178                     10.650538                      5.067298
 Log_total_assets_CY1_missing             EBITDA_margin_CY1               EBIT_margin_CY1
                      5.034935                      3.988157                      3.979473
 Growth_accel_revenue_missing     Log_employees_CY1_missing          Log_total_assets_CY1
                      3.511495                      3.449324                      2.929927
        Growth_accel_revenue             Log_employees_CY1         CAGR_revenue_fallback
                      2.439369                      2.322110                      2.113502
           Volatility_EBITDA                       ROA_CY1        Sponsor_backed_missing
                      2.103787                      1.483081                      1.318149
            Equity_ratio_CY1          CAGR_EBITDA_fallback          Sponsor_backed_dummy
                      1.248560                      1.174902                      1.067536
```

```
# the five above 10 flag one event, a company not filing full accounts,
# so the block goes together; Sponsor_backed_missing flags 1 row of 369

missing_dummies <- c("Log_total_assets_CY1_missing",
                     "Log_employees_CY1_missing",
                     "CAGR_revenue_fallback_missing",
                     "CAGR_EBITDA_fallback_missing",
                     "Growth_accel_revenue_missing",
                     "EBITDA_margin_CY1_missing",
                     "EBIT_margin_CY1_missing",
                     "Volatility_EBITDA_missing",
                     "ROA_CY1_missing",
                     "Sponsor_backed_missing")

feature_cols_final <- feature_cols_clean[!feature_cols_clean %in%
                                         missing_dummies]
print(feature_cols_final)
 [1] "Log_total_assets_CY1"  "Log_employees_CY1"     "CAGR_revenue_fallback"
 [4] "CAGR_EBITDA_fallback"  "Growth_accel_revenue"  "EBITDA_margin_CY1"
 [7] "EBIT_margin_CY1"       "Volatility_EBITDA"     "ROA_CY1"
[10] "Equity_ratio_CY1"      "Sponsor_backed_dummy"
```

Step 4: final specification

```
formula_final <- as.formula(paste("Transaction_EV_EBITDA ~",
                                  paste(feature_cols_final,
                                        collapse = " + ")))
model_ols3 <- lm(formula_final, data = ebitda_train)

cat("Adj R2:", round(summary(model_ols3)$adj.r.squared, 4), "\n")
Adj R2: 0.0178
cat("F p-value:", round(pf(summary(model_ols3)$fstatistic[1],
                           summary(model_ols3)$fstatistic[2],
                           summary(model_ols3)$fstatistic[3],
                           lower.tail = FALSE), 4), "\n")
F p-value: 0.0951
print(summary(model_ols3)$coefficients)
                         Estimate Std. Error     t value    Pr(>|t|)
(Intercept)          23.77124448  7.9117082  3.00456538 0.002847474
Log_total_assets_CY1 -2.56680071  0.9405607 -2.72901126 0.006666908
Log_employees_CY1     1.75371136  1.0404019  1.68560948 0.092744823
CAGR_revenue_fallback 0.81494343  2.4506869  0.33253673 0.739679182
CAGR_EBITDA_fallback  0.04975225  0.6641410  0.07491219 0.940326539
Growth_accel_revenue  1.14785306  2.3679082  0.48475404 0.628148116
EBITDA_margin_CY1     0.07137913  0.1879124  0.37985316 0.704280292
EBIT_margin_CY1      -0.09840644  0.1800378 -0.54658771 0.585003466
Volatility_EBITDA     0.68846744  1.1952126  0.57602089 0.564964005
ROA_CY1              -4.52856967  5.1925562 -0.87212723 0.383725133
Equity_ratio_CY1      3.81734226  6.5135744  0.58605952 0.558205851
Sponsor_backed_dummy  8.72605425  2.8391358  3.07348952 0.002278440
# everything under 4 now
vif3 <- vif(model_ols3)
print(sort(vif3, decreasing = TRUE))
```

```
      EBIT_margin_CY1     EBITDA_margin_CY1  Log_total_assets_CY1     Log_employees_CY1
             3.778816              3.743114             2.411726              2.196954
Growth_accel_revenue CAGR_revenue_fallback     Volatility_EBITDA               ROA_CY1
             1.978535              1.932662             1.904183              1.226603
    Equity_ratio_CY1  CAGR_EBITDA_fallback  Sponsor_backed_dummy
             1.176558              1.134323             1.036271
```

Cross-validation folds

The folds are built here rather than alongside the machine learning models because the ridge penalty below is selected on the same partition. One grouped assignment therefore governs every cross-validated choice in the study: the ridge lambda, and the sixteen hyperparameter searches.

```
# folds go to companies and not to rows, so no entity can end up on both
# sides of a split

set.seed(42)
unique_entities <- unique(ebitda_train$BvD_ID)
entity_folds <- sample(rep(1:5, length.out = length(unique_entities)))
names(entity_folds) <- unique_entities

cat("unique entities in the EBITDA training set:",
    length(unique_entities), "\n")
unique entities in the EBITDA training set: 345
cat("of which with more than one transaction:",
    sum(table(ebitda_train$BvD_ID) > 1), "\n")
of which with more than one transaction: 23
ebitda_train$fold <- entity_folds[ebitda_train$BvD_ID]
revenue_train$fold <- entity_folds[revenue_train$BvD_ID]

print(table(ebitda_train$fold))

 1  2  3  4  5
71 77 73 74 74
cat("rows with no fold assigned:", sum(is.na(ebitda_train$fold)), "\n")
rows with no fold assigned: 0

# four companies are in the revenue file but not the EBITDA one so they
# come back with no fold, without this ranger stops on "Missing data in
# dependent variable"
missing_entities <- unique(revenue_train$BvD_ID[

  is.na(revenue_train$fold)])
cat("Entities with missing fold:", length(missing_entities), "\n")
Entities with missing fold: 4
set.seed(42)
extra_folds <- sample(1:5, length(missing_entities), replace = TRUE)
names(extra_folds) <- missing_entities
for (e in missing_entities) {
  revenue_train$fold[revenue_train$BvD_ID == e] <- extra_folds[e]
}

revenue_train_clean <- revenue_train[
  !is.na(revenue_train$Transaction_EV_Revenue), ]
cat("Revenue train clean:", nrow(revenue_train_clean), "rows\n")
Revenue train clean: 370 rows
cat("rows with no fold assigned:",
    sum(is.na(revenue_train_clean$fold)), "\n")
rows with no fold assigned: 0
print(table(revenue_train_clean$fold))

 1  2  3  4  5
73 78 69 74 76
```

Regression diagnostics

```
bp <- bptest(model_ols3, studentize = TRUE)
print(bp)

    studentized Breusch-Pagan test

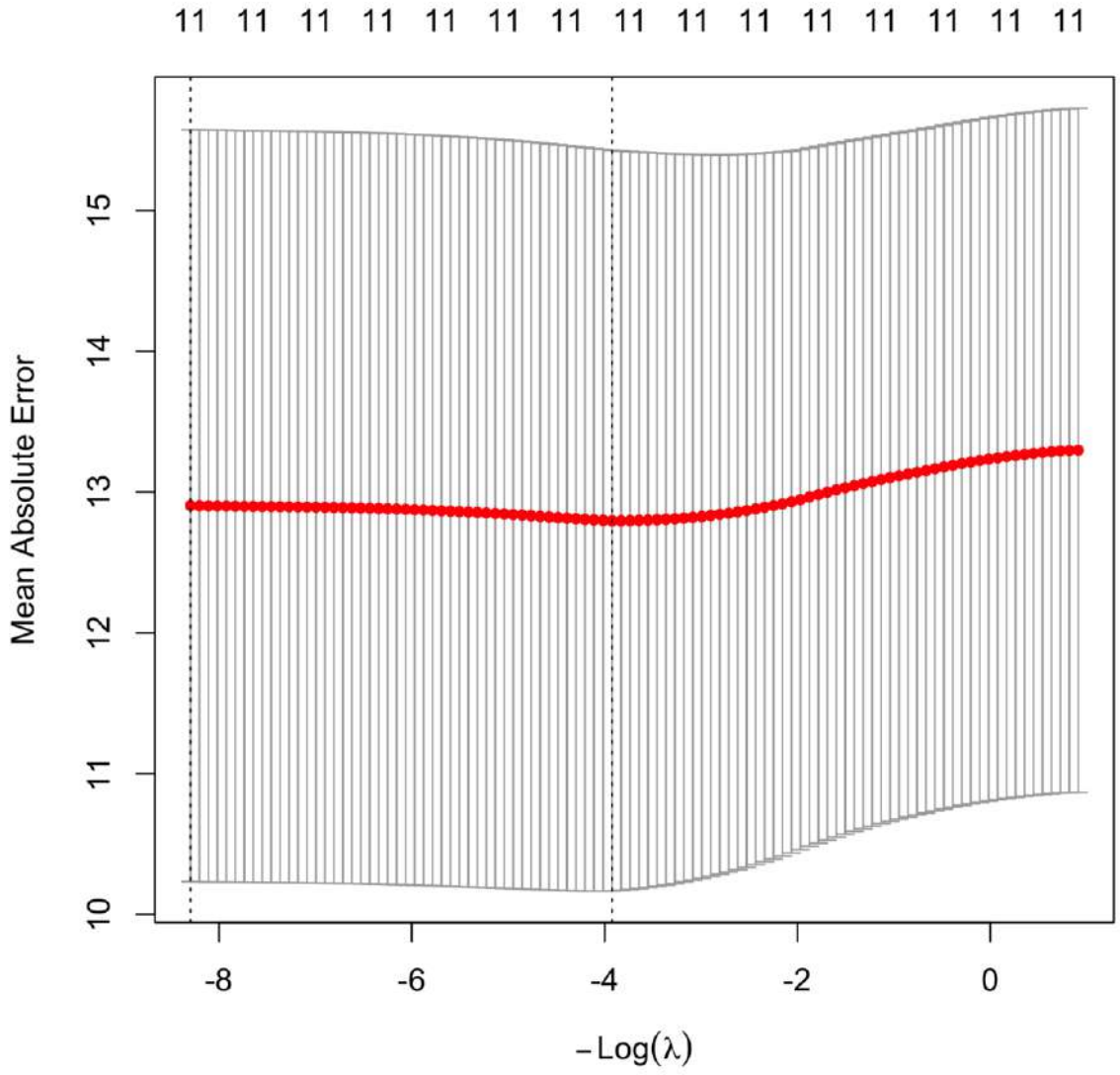
data:  model_ols3
BP = 5.5523, df = 11, p-value = 0.9015
cooksd <- cooks.distance(model_ols3)
influential_idx <- which(cooksd > 4 / nrow(ebitda_train))
cat("Influential observations:", length(influential_idx), "\n")
Influential observations: 13
print(ebitda_train$Transaction_EV_EBITDA[influential_idx])
 [1]  79.25734  10.00000  16.79102 115.39730 134.09314 108.54541 232.73162  60.00000
 [9] 138.30427 183.12819 172.43033 220.16288  41.59709

# is the flat fit coming from the tail or does it hold throughout
extreme_mask <- ebitda_train$Transaction_EV_EBITDA > 50 |
                ebitda_train$Transaction_EV_EBITDA < 0
cat("Extreme transactions in train:", sum(extreme_mask), "\n")
Extreme transactions in train: 17
ebitda_train_clean <- ebitda_train[!extreme_mask, ]
model_ols_clean <- lm(formula_final, data = ebitda_train_clean)
cat("Adj R2 without extremes:",
    round(summary(model_ols_clean)$adj.r.squared, 4), "\n")
Adj R2 without extremes: 0.0139
cat("F p-value:", round(pf(summary(model_ols_clean)$fstatistic[1],
                           summary(model_ols_clean)$fstatistic[2],
                           summary(model_ols_clean)$fstatistic[3],
                           lower.tail = FALSE), 4), "\n")
F p-value: 0.1498
print(summary(model_ols_clean)$coefficients)
                         Estimate Std. Error     t value     Pr(>|t|)
(Intercept)          10.530050692 2.66992030  3.94395693 9.736058e-05
Log_total_assets_CY1  0.023863701 0.37678174  0.06333561 9.495365e-01
Log_employees_CY1    -0.188635885 0.36192613 -0.52119996 6.025668e-01
CAGR_revenue_fallback 0.362834106 0.79987005  0.45361632 6.503944e-01
CAGR_EBITDA_fallback -0.375825121 0.21446630 -1.75237380 8.061096e-02
Growth_accel_revenue  0.446037487 0.77805897  0.57326952 5.668412e-01
EBITDA_margin_CY1     0.004801117 0.06124051  0.07839772 9.375578e-01
EBIT_margin_CY1      -0.019785619 0.05845026 -0.33850350 7.351927e-01
Volatility_EBITDA     0.010207850 0.40746367  0.02505217 9.800280e-01
ROA_CY1              -2.069895097 1.69494313 -1.22121802 2.228496e-01
Equity_ratio_CY1      1.244687916 2.15258043  0.57823062 5.634912e-01
Sponsor_backed_dummy  2.803931366 0.94601469  2.96394061 3.251552e-03
```

Ridge

```
X_train_mat <- as.matrix(ebitda_train[, feature_cols_final])
y_train <- ebitda_train$Transaction_EV_EBITDA

# the entity-grouped folds from above, so no company is split across the
# penalty search either
set.seed(42)
ridge_cv <- cv.glmnet(X_train_mat, y_train, alpha = 0,
                      foldid = ebitda_train$fold, type.measure = "mae")
plot(ridge_cv)
```




```
cat("Optimal lambda:", round(ridge_cv$lambda.min, 4), "\n")
Optimal lambda: 50.5815
cat("Conservative lambda:", round(ridge_cv$lambda.1se, 4), "\n")
Conservative lambda: 4008.497
ols_coef <- coef(model_ols3)[-1]
ridge_coef <- as.vector(coef(ridge_cv, s = "lambda.min"))[-1]

coef_compare <- data.frame(
  Variable = feature_cols_final,
  OLS = round(ols_coef, 4),
  Ridge = round(ridge_coef, 4),
  Sign_change = sign(ols_coef) != sign(ridge_coef))
print(coef_compare)
                              Variable     OLS   Ridge Sign_change
Log_total_assets_CY1   Log_total_assets_CY1 -2.5668 -0.3632       FALSE
Log_employees_CY1         Log_employees_CY1  1.7537  0.1866       FALSE
CAGR_revenue_fallback CAGR_revenue_fallback  0.8149  0.0088       FALSE
CAGR_EBITDA_fallback   CAGR_EBITDA_fallback  0.0498 -0.0137        TRUE
Growth_accel_revenue   Growth_accel_revenue  1.1479  0.3028       FALSE
EBITDA_margin_CY1         EBITDA_margin_CY1  0.0714 -0.0101        TRUE
EBIT_margin_CY1             EBIT_margin_CY1 -0.0984 -0.0148       FALSE
Volatility_EBITDA         Volatility_EBITDA  0.6885  0.0055       FALSE
ROA_CY1                             ROA_CY1 -4.5286 -1.0464       FALSE
Equity_ratio_CY1           Equity_ratio_CY1  3.8173  0.5947       FALSE
Sponsor_backed_dummy   Sponsor_backed_dummy  8.7261  2.8618       FALSE
```

EV/Revenue regression

```
formula_revenue <- as.formula(paste("Transaction_EV_Revenue ~",
                                    paste(feature_cols_final,
                                          collapse = " + ")))
model_ols_revenue <- lm(formula_revenue, data = revenue_train)

cat("Adj R2:", round(summary(model_ols_revenue)$adj.r.squared, 4), "\n")
Adj R2: 0.028
cat("F p-value:", round(pf(summary(model_ols_revenue)$fstatistic[1],
                           summary(model_ols_revenue)$fstatistic[2],
                           summary(model_ols_revenue)$fstatistic[3],
                           lower.tail = FALSE), 4), "\n")
F p-value: 0.0307
print(summary(model_ols_revenue)$coefficients)
                        Estimate Std. Error    t value    Pr(>|t|)
(Intercept)           1.96122160 4.07850343  0.4808680 0.630904092
Log_total_assets_CY1  0.05874418 0.47582698  0.1234570 0.901814495
Log_employees_CY1    -0.92731507 0.52970780 -1.7506162 0.080868594
CAGR_revenue_fallback 3.63859232 1.23164767  2.9542477 0.003341602
CAGR_EBITDA_fallback  0.24818813 0.33693618  0.7366028 0.461846354
Growth_accel_revenue  2.87847737 1.19395281  2.4108804 0.016417211
EBITDA_margin_CY1    -0.03904638 0.09506273 -0.4107433 0.681506527
EBIT_margin_CY1       0.08570144 0.09105860  0.9411680 0.347253416
Volatility_EBITDA     0.54532835 0.60214271  0.9056464 0.365732165
ROA_CY1              -1.76410118 3.03299584 -0.5816365 0.561177579
Equity_ratio_CY1     -0.65894142 3.33082185 -0.1978315 0.843289197
Sponsor_backed_dummy  3.09510069 1.43887150  2.1510612 0.032140106
print(sort(vif(model_ols_revenue), decreasing = TRUE))
      EBIT_margin_CY1     EBITDA_margin_CY1  Log_total_assets_CY1     Log_employees_CY1
             3.912665              3.864057             2.391571              2.231647
 Growth_accel_revenue CAGR_revenue_fallback     Volatility_EBITDA               ROA_CY1
             1.952044              1.916623             1.852680              1.289754
     Equity_ratio_CY1  CAGR_EBITDA_fallback  Sponsor_backed_dummy
             1.177786              1.133402             1.033973
bp_rev <- bptest(model_ols_revenue, studentize = TRUE)
print(bp_rev)

    studentized Breusch-Pagan test

data:  model_ols_revenue
BP = 19.642, df = 11, p-value = 0.0505

# borderline at 0.051 so report HC3 as well
print(coeftest(model_ols_revenue,
               vcov = vcovHC(model_ols_revenue, type = "HC3")))

t test of coefficients:

                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            1.961222   1.992820  0.9841  0.32571
Log_total_assets_CY1   0.058744   0.215975  0.2720  0.78578
Log_employees_CY1     -0.927315   1.030097 -0.9002  0.36861
CAGR_revenue_fallback  3.638592  10.938890  0.3326  0.73961
CAGR_EBITDA_fallback   0.248188   0.442738  0.5606  0.57544
Growth_accel_revenue   2.878477   3.751842  0.7672  0.44346
EBITDA_margin_CY1     -0.039046   0.098663 -0.3958  0.69252
EBIT_margin_CY1        0.085701   0.092884  0.9227  0.35680
Volatility_EBITDA      0.545328   0.774134  0.7044  0.48162
ROA_CY1               -1.764101   2.919641 -0.6042  0.54608
Equity_ratio_CY1      -0.658941   1.639298 -0.4020  0.68795
Sponsor_backed_dummy   3.095101   1.843637  1.6788  0.09406 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ridge on revenue as well, same settings and the same grouped folds
X_train_rev <- as.matrix(revenue_train[, feature_cols_final])
y_train_rev <- revenue_train$Transaction_EV_Revenue

set.seed(42)
ridge_cv_rev <- cv.glmnet(X_train_rev, y_train_rev, alpha = 0,
                          foldid = revenue_train$fold,
                          type.measure = "mae")
cat("Optimal lambda:", round(ridge_cv_rev$lambda.min, 4), "\n")
Optimal lambda: 44.5145
coef_compare_rev <- data.frame(
  Variable = feature_cols_final,
  OLS = round(coef(model_ols_revenue)[-1], 4),
  Ridge = round(as.vector(coef(ridge_cv_rev, s = "lambda.min"))[-1], 4))
print(coef_compare_rev)
                                 Variable     OLS   Ridge
Log_total_assets_CY1   Log_total_assets_CY1  0.0587 -0.0254
Log_employees_CY1         Log_employees_CY1 -0.9273 -0.1256
CAGR_revenue_fallback CAGR_revenue_fallback  3.6386  0.4616
CAGR_EBITDA_fallback   CAGR_EBITDA_fallback  0.2482  0.0874
Growth_accel_revenue   Growth_accel_revenue  2.8785  0.1535
EBITDA_margin_CY1         EBITDA_margin_CY1 -0.0390  0.0105
EBIT_margin_CY1             EBIT_margin_CY1  0.0857  0.0121
Volatility_EBITDA         Volatility_EBITDA  0.5453  0.0223
ROA_CY1                             ROA_CY1 -1.7641 -0.0463
Equity_ratio_CY1           Equity_ratio_CY1 -0.6589  0.0279
Sponsor_backed_dummy   Sponsor_backed_dummy  3.0951  0.7273
```

Augmented regressions

```
aug_ebitda_train <- read_excel(
  paste0(DATA, "Reconciliation_ML_Augmented_Train_v2.xlsx"),
  sheet = "EBITDA_Train")
aug_revenue_train <- read_excel(
  paste0(DATA, "Reconciliation_ML_Augmented_Train_v2.xlsx"),
  sheet = "Revenue_Train")

# comp-set mean and count are left out, the mean is collinear with the
# median and the count is a data quality flag rather than a price signal
feature_cols_aug_ebitda <- c(feature_cols_final, "Comps_Median_EV_EBITDA",
                             "STOXX600_at_closing", "Rate_10Y_at_closing")
feature_cols_aug_revenue <- c(feature_cols_final,
                              "Comps_Median_EV_Revenue",
                              "STOXX600_at_closing", "Rate_10Y_at_closing")

cat("Augmented EBITDA features:",
    length(feature_cols_aug_ebitda), "\n")
Augmented EBITDA features: 14
cat("Augmented Revenue features:",
    length(feature_cols_aug_revenue), "\n")
Augmented Revenue features: 14
formula_aug_ebitda <- as.formula(paste("Transaction_EV_EBITDA ~",
                                       paste(feature_cols_aug_ebitda,
                                             collapse = " + ")))
model_aug_ebitda <- lm(formula_aug_ebitda, data = aug_ebitda_train)
cat("Adj R2:", round(summary(model_aug_ebitda)$adj.r.squared, 4), "\n")
Adj R2: 0.0289
cat("F p-value:", round(pf(summary(model_aug_ebitda)$fstatistic[1],
                           summary(model_aug_ebitda)$fstatistic[2],
                           summary(model_aug_ebitda)$fstatistic[3],
                           lower.tail = FALSE), 4), "\n")
F p-value: 0.0398
print(summary(model_aug_ebitda)$coefficients)
```

```
                         Estimate  Std. Error     t value     Pr(>|t|)
(Intercept)           -3.56606082 13.63969523 -0.26144725 0.7938996676
Log_total_assets_CY1  -2.46278407  0.93639814 -2.63006083 0.0089096315
Log_employees_CY1      1.76734231  1.03811687  1.70245025 0.0895488475
CAGR_revenue_fallback  1.28057417  2.44931130  0.52283030 0.6014194150
CAGR_EBITDA_fallback  -0.01939094  0.66379326 -0.02921231 0.9767117213
Growth_accel_revenue   1.27958360  2.36468311  0.54112265 0.5887633960
EBITDA_margin_CY1      0.13899535  0.18897012  0.73554139 0.4624968870
EBIT_margin_CY1       -0.15472817  0.18074734 -0.85604677 0.3925509691
Volatility_EBITDA      0.55775771  1.19278019  0.46761148 0.6403501778
ROA_CY1               -5.19130081  5.17300089 -1.00353758 0.3162870143
Equity_ratio_CY1       2.05795034  6.51196488  0.31602602 0.7521690056
Sponsor_backed_dummy   9.57710962  2.86256085  3.34564403 0.0009089094
Comps_Median_EV_EBITDA 0.66101423  0.60579389  1.09115366 0.2759473828
STOXX600_at_closing    0.05844878  0.02867793  2.03811007 0.0422818176
Rate_10Y_at_closing   -2.39710128  1.62089164 -1.47887818 0.1400621307

print(sort(vif(model_aug_ebitda), decreasing = TRUE))
       EBIT_margin_CY1      EBITDA_margin_CY1   Log_total_assets_CY1
              3.852138              3.828581              2.417714
     Log_employees_CY1   Growth_accel_revenue  CAGR_revenue_fallback
              2.212284              1.995674              1.952531
     Volatility_EBITDA    STOXX600_at_closing    Rate_10Y_at_closing
              1.918089              1.845525              1.826912
               ROA_CY1       Equity_ratio_CY1 Comps_Median_EV_EBITDA
              1.231278              1.189401              1.182657
  CAGR_EBITDA_fallback   Sponsor_backed_dummy
              1.146071              1.065468
print(bptest(model_aug_ebitda, studentize = TRUE))

    studentized Breusch-Pagan test

data:  model_aug_ebitda
BP = 9.6379, df = 14, p-value = 0.7881
formula_aug_revenue <- as.formula(paste("Transaction_EV_Revenue ~",
                                        paste(feature_cols_aug_revenue,
                                              collapse = " + ")))
model_aug_revenue <- lm(formula_aug_revenue, data = aug_revenue_train)
cat("Adj R2:", round(summary(model_aug_revenue)$adj.r.squared, 4), "\n")
Adj R2: 0.0224
cat("F p-value:", round(pf(summary(model_aug_revenue)$fstatistic[1],
                           summary(model_aug_revenue)$fstatistic[2],
                           summary(model_aug_revenue)$fstatistic[3],
                           lower.tail = FALSE), 4), "\n")
F p-value: 0.0755

print(summary(model_aug_revenue)$coefficients)
                           Estimate Std. Error     t value    Pr(>|t|)
(Intercept)             1.207845809 6.97842933  0.17308276 0.862685018
Log_total_assets_CY1    0.042112053 0.47825049  0.08805439 0.929883100
Log_employees_CY1      -0.907643782 0.53242713 -1.70472864 0.089119944
CAGR_revenue_fallback   3.640193500 1.23886593  2.93832723 0.003515603
CAGR_EBITDA_fallback    0.269458188 0.33947609  0.79374717 0.427872863
Growth_accel_revenue    2.853031863 1.19818268  2.38113262 0.017786201
EBITDA_margin_CY1      -0.034200462 0.09584594 -0.35682744 0.721432993
EBIT_margin_CY1         0.081498294 0.09187701  0.88703687 0.375659586
Volatility_EBITDA       0.493935038 0.60685616  0.81392440 0.416233863
ROA_CY1                -1.714543787 3.05322810 -0.56155116 0.574776189
Equity_ratio_CY1       -0.755716264 3.35292787 -0.22538996 0.821805618
Sponsor_backed_dummy    3.217103877 1.45825074  2.20613903 0.028014226
Comps_Median_EV_Revenue -1.141658686 2.35180447 -0.48543946 0.627664099
STOXX600_at_closing     0.007799959 0.01486396  0.52475657 0.600079962
Rate_10Y_at_closing    -0.736142183 0.80801121 -0.91105441 0.362884905
print(sort(vif(model_aug_revenue), decreasing = TRUE))
       EBIT_margin_CY1      EBITDA_margin_CY1   Log_total_assets_CY1
              3.960638              3.905631              2.402241
     Log_employees_CY1   Growth_accel_revenue  CAGR_revenue_fallback
              2.241784              1.954709              1.928116
     Volatility_EBITDA    STOXX600_at_closing    Rate_10Y_at_closing
              1.871086              1.790135              1.698492
               ROA_CY1       Equity_ratio_CY1   CAGR_EBITDA_fallback
              1.299579              1.186677              1.144004
Comps_Median_EV_Revenue   Sponsor_backed_dummy
              1.131098              1.055967
```

```
print(bptest(model_aug_revenue, studentize = TRUE))

    studentized Breusch-Pagan test

data:  model_aug_revenue
BP = 19.717, df = 14, p-value = 0.1393

# same folds carried over to the augmented files
aug_ebitda_train$fold <- entity_folds[aug_ebitda_train$BvD_ID]
aug_revenue_train$fold <- entity_folds[aug_revenue_train$BvD_ID]

missing_entities_aug <- unique(aug_revenue_train$BvD_ID[
  is.na(aug_revenue_train$fold)])
if (length(missing_entities_aug) > 0) {
  set.seed(42)
  extra_folds_aug <- sample(1:5, length(missing_entities_aug),
                            replace = TRUE)
  names(extra_folds_aug) <- missing_entities_aug
  for (e in missing_entities_aug) {
    aug_revenue_train$fold[aug_revenue_train$BvD_ID == e] <-
      extra_folds_aug[e]
  }
}
aug_revenue_train_clean <- aug_revenue_train[
  !is.na(aug_revenue_train$Transaction_EV_Revenue), ]

cat("rows with no fold assigned, aug EBITDA:",
    sum(is.na(aug_ebitda_train$fold)), "\n")
rows with no fold assigned, aug EBITDA: 0
cat("rows with no fold assigned, aug Revenue:",
    sum(is.na(aug_revenue_train_clean$fold)), "\n")
rows with no fold assigned, aug Revenue: 0
```

ML feature sets

```
# ML-A is the regression specification, so any difference in performance
# comes from the functional form and not from the information
features_A <- c("Log_total_assets_CY1", "Log_employees_CY1",
                "CAGR_revenue_fallback", "CAGR_EBITDA_fallback",
                "Growth_accel_revenue", "EBITDA_margin_CY1",
                "EBIT_margin_CY1", "Volatility_EBITDA", "ROA_CY1",
                "Equity_ratio_CY1", "Sponsor_backed_dummy")

# ML-B puts back what the regression could not carry, trees do not mind
# collinearity
features_B <- c(features_A,
                "CAGR_revenue_horizon_years",
                "CAGR_EBITDA_horizon_years",
                "Log_total_assets_CY1_missing",
                "Log_employees_CY1_missing",
                "CAGR_revenue_fallback_missing",
                "CAGR_EBITDA_fallback_missing",
                "Growth_accel_revenue_missing",
                "EBITDA_margin_CY1_missing",
                "EBIT_margin_CY1_missing",
                "Volatility_EBITDA_missing",
                "ROA_CY1_missing")

cat("ML-A features:", length(features_A), "\n")
ML-A features: 11
cat("ML-B features:", length(features_B), "\n")
ML-B features: 22
cat("Missing from train - ML-A:",
    paste(features_A[!features_A %in% names(ebitda_train)],
          collapse = ", "), "\n")
Missing from train - ML-A:
cat("Missing from train - ML-B:",
    paste(features_B[!features_B %in% names(ebitda_train)],
          collapse = ", "), "\n")
Missing from train - ML-B:
```

```
market_vars_ebitda <- c("Comps_Median_EV_EBITDA", "STOXX600_at_closing",
                        "Rate_10Y_at_closing")
market_vars_revenue <- c("Comps_Median_EV_Revenue", "STOXX600_at_closing",
                         "Rate_10Y_at_closing")

features_aug_A_ebitda <- c(features_A, market_vars_ebitda)
features_aug_B_ebitda <- c(features_B, market_vars_ebitda)
features_aug_A_revenue <- c(features_A, market_vars_revenue)
features_aug_B_revenue <- c(features_B, market_vars_revenue)

cat("Aug ML-A EBITDA:", length(features_aug_A_ebitda), "\n")
Aug ML-A EBITDA: 14
cat("Aug ML-B EBITDA:", length(features_aug_B_ebitda), "\n")
Aug ML-B EBITDA: 25
```

Grid search

```
run_cv_fold <- function(train_df, features, target, fold_col,
                        model_type, params) {
  fold_maes <- numeric(5)
  for (k in 1:5) {
    tr <- train_df[train_df[[fold_col]] != k, ]
    va <- train_df[train_df[[fold_col]] == k, ]
    X_tr <- as.matrix(tr[, features])
    y_tr <- tr[[target]]
    X_va <- as.matrix(va[, features])
    y_va <- va[[target]]

    if (model_type == "rf") {
      m <- ranger(x = X_tr, y = y_tr,
                  num.trees = 500,
                  mtry = params$mtry,
                  min.node.size = params$min.node.size,
                  max.depth = params$max.depth,
                  seed = 42, num.threads = 1)
      preds <- predict(m, data = X_va)$predictions

    } else if (model_type == "xgb") {
      dtrain <- xgb.DMatrix(X_tr, label = y_tr)
      dval <- xgb.DMatrix(X_va, label = y_va)
      # the seed inside params is ignored by the R interface and kept for
      # documentation only; R's own generator drives the row and column
      # subsampling, so it is seeded here. Same stream for every fold and
      # every grid combination, with nthread = 1 removing the last of the
      # scheduling nondeterminism
      set.seed(42)
      m <- xgb.train(
        params = list(objective = "reg:squarederror",
                      max_depth = params$max_depth,
                      eta = params$eta,
                      min_child_weight = params$min_child_weight,
                      subsample = 0.8,
                      colsample_bytree = 0.8,
                      seed = 42, nthread = 1),
        data = dtrain,
        nrounds = params$nrounds,
        verbose = 0)
      preds <- predict(m, dval)
    }
    fold_maes[k] <- mean(abs(y_va - preds))
  }
  return(mean(fold_maes))
}
```

```r
# RF, 3 mtry by 3 node sizes by 2 depths. NULL means unlimited and drops
# out of the list, which is what ranger expects
rf_params_list <- list(
  list(mtry = 4, min.node.size = 5,  max.depth = NULL),
  list(mtry = 4, min.node.size = 10, max.depth = NULL),
  list(mtry = 4, min.node.size = 20, max.depth = NULL),
  list(mtry = 6, min.node.size = 5,  max.depth = NULL),
  list(mtry = 6, min.node.size = 10, max.depth = NULL),
  list(mtry = 6, min.node.size = 20, max.depth = NULL),
  list(mtry = 8, min.node.size = 5,  max.depth = NULL),
  list(mtry = 8, min.node.size = 10, max.depth = NULL),
  list(mtry = 8, min.node.size = 20, max.depth = NULL),
  list(mtry = 4, min.node.size = 5,  max.depth = 4),
  list(mtry = 4, min.node.size = 10, max.depth = 4),
  list(mtry = 4, min.node.size = 20, max.depth = 4),
  list(mtry = 6, min.node.size = 5,  max.depth = 4),
  list(mtry = 6, min.node.size = 10, max.depth = 4),
  list(mtry = 6, min.node.size = 20, max.depth = 4),
  list(mtry = 8, min.node.size = 5,  max.depth = 4),
  list(mtry = 8, min.node.size = 10, max.depth = 4),
  list(mtry = 8, min.node.size = 20, max.depth = 4))

xgb_params_list <- list(
  list(nrounds = 100, max_depth = 2, eta = 0.05, min_child_weight = 5),
  list(nrounds = 100, max_depth = 2, eta = 0.10, min_child_weight = 5),
  list(nrounds = 100, max_depth = 3, eta = 0.05, min_child_weight = 5),
  list(nrounds = 100, max_depth = 3, eta = 0.10, min_child_weight = 5),
  list(nrounds = 200, max_depth = 2, eta = 0.05, min_child_weight = 5),
  list(nrounds = 200, max_depth = 2, eta = 0.10, min_child_weight = 5),
  list(nrounds = 200, max_depth = 3, eta = 0.05, min_child_weight = 5),
  list(nrounds = 200, max_depth = 3, eta = 0.10, min_child_weight = 5),
  list(nrounds = 100, max_depth = 2, eta = 0.05, min_child_weight = 10),
  list(nrounds = 100, max_depth = 2, eta = 0.10, min_child_weight = 10),
  list(nrounds = 100, max_depth = 3, eta = 0.05, min_child_weight = 10),
  list(nrounds = 100, max_depth = 3, eta = 0.10, min_child_weight = 10),
  list(nrounds = 200, max_depth = 2, eta = 0.05, min_child_weight = 10),
  list(nrounds = 200, max_depth = 2, eta = 0.10, min_child_weight = 10),
  list(nrounds = 200, max_depth = 3, eta = 0.05, min_child_weight = 10),
  list(nrounds = 200, max_depth = 3, eta = 0.10, min_child_weight = 10))

cat("RF combinations:", length(rf_params_list), "\n")
RF combinations: 18
cat("XGB combinations:", length(xgb_params_list), "\n")
XGB combinations: 16
# EV/EBITDA, pure
rf_cv_A_ebitda <- sapply(rf_params_list, function(p) {
  run_cv_fold(ebitda_train, features_A, "Transaction_EV_EBITDA",
              "fold", "rf", p)
})
best_rf_A_ebitda <- rf_params_list[[which.min(rf_cv_A_ebitda)]]
cat("Best RF ML-A EBITDA: mtry", best_rf_A_ebitda$mtry,
    "min.node", best_rf_A_ebitda$min.node.size,
    "| CV MAE", round(min(rf_cv_A_ebitda), 3), "\n")
Best RF ML-A EBITDA: mtry 4 min.node 20 | CV MAE 13.31
xgb_cv_A_ebitda <- sapply(xgb_params_list, function(p) {
  run_cv_fold(ebitda_train, features_A, "Transaction_EV_EBITDA",
              "fold", "xgb", p)
})
best_xgb_A_ebitda <- xgb_params_list[[which.min(xgb_cv_A_ebitda)]]

rf_cv_B_ebitda <- sapply(rf_params_list, function(p) {
  run_cv_fold(ebitda_train, features_B, "Transaction_EV_EBITDA",
              "fold", "rf", p)
})
best_rf_B_ebitda <- rf_params_list[[which.min(rf_cv_B_ebitda)]]

xgb_cv_B_ebitda <- sapply(xgb_params_list, function(p) {
  run_cv_fold(ebitda_train, features_B, "Transaction_EV_EBITDA",
              "fold", "xgb", p)
})
best_xgb_B_ebitda <- xgb_params_list[[which.min(xgb_cv_B_ebitda)]]
```

```
# EV/Revenue, pure
rf_cv_A_revenue <- sapply(rf_params_list, function(p) {
  run_cv_fold(revenue_train_clean, features_A, "Transaction_EV_Revenue",
              "fold", "rf", p)
})
best_rf_A_revenue <- rf_params_list[[which.min(rf_cv_A_revenue)]]

xgb_cv_A_revenue <- sapply(xgb_params_list, function(p) {
  run_cv_fold(revenue_train_clean, features_A, "Transaction_EV_Revenue",
              "fold", "xgb", p)
})
best_xgb_A_revenue <- xgb_params_list[[which.min(xgb_cv_A_revenue)]]

rf_cv_B_revenue <- sapply(rf_params_list, function(p) {
  run_cv_fold(revenue_train_clean, features_B, "Transaction_EV_Revenue",
              "fold", "rf", p)
})
best_rf_B_revenue <- rf_params_list[[which.min(rf_cv_B_revenue)]]

xgb_cv_B_revenue <- sapply(xgb_params_list, function(p) {
  run_cv_fold(revenue_train_clean, features_B, "Transaction_EV_Revenue",
              "fold", "xgb", p)
})
best_xgb_B_revenue <- xgb_params_list[[which.min(xgb_cv_B_revenue)]]

cv_summary <- data.frame(
  Target = c("EV/EBITDA", "EV/EBITDA", "EV/EBITDA", "EV/EBITDA",
             "EV/Revenue", "EV/Revenue", "EV/Revenue", "EV/Revenue"),
  FeatureSet = c("ML-A", "ML-A", "ML-B", "ML-B",
                 "ML-A", "ML-A", "ML-B", "ML-B"),
  Model = c("RF", "XGB", "RF", "XGB", "RF", "XGB", "RF", "XGB"),
  Best_CV_MAE = round(c(min(rf_cv_A_ebitda), min(xgb_cv_A_ebitda),
                        min(rf_cv_B_ebitda), min(xgb_cv_B_ebitda),
                        min(rf_cv_A_revenue), min(xgb_cv_A_revenue),
                        min(rf_cv_B_revenue), min(xgb_cv_B_revenue)), 3))

print(cv_summary)
      Target FeatureSet Model Best_CV_MAE
1  EV/EBITDA       ML-A    RF      13.310
2  EV/EBITDA       ML-A   XGB      13.730
3  EV/EBITDA       ML-B    RF      13.455
4  EV/EBITDA       ML-B   XGB      13.793
5 EV/Revenue       ML-A    RF       3.532
6 EV/Revenue       ML-A   XGB       3.365
7 EV/Revenue       ML-B    RF       3.340
8 EV/Revenue       ML-B   XGB       3.336
write.csv(cv_summary, paste0(RES_ML, "ML_CV_Summary.csv"),
          row.names = FALSE)
```

Grid search, augmented

```
rf_cv_augA_ebitda <- sapply(rf_params_list, function(p) {
  run_cv_fold(aug_ebitda_train, features_aug_A_ebitda,
              "Transaction_EV_EBITDA", "fold", "rf", p)
})
best_rf_augA_ebitda <- rf_params_list[[which.min(rf_cv_augA_ebitda)]]

xgb_cv_augA_ebitda <- sapply(xgb_params_list, function(p) {
  run_cv_fold(aug_ebitda_train, features_aug_A_ebitda,
              "Transaction_EV_EBITDA", "fold", "xgb", p)
})
best_xgb_augA_ebitda <- xgb_params_list[[which.min(xgb_cv_augA_ebitda)]]

rf_cv_augB_ebitda <- sapply(rf_params_list, function(p) {
  run_cv_fold(aug_ebitda_train, features_aug_B_ebitda,
              "Transaction_EV_EBITDA", "fold", "rf", p)
})
best_rf_augB_ebitda <- rf_params_list[[which.min(rf_cv_augB_ebitda)]]

xgb_cv_augB_ebitda <- sapply(xgb_params_list, function(p) {
  run_cv_fold(aug_ebitda_train, features_aug_B_ebitda,
              "Transaction_EV_EBITDA", "fold", "xgb", p)
})
best_xgb_augB_ebitda <- xgb_params_list[[which.min(xgb_cv_augB_ebitda)]]
rf_cv_augA_revenue <- sapply(rf_params_list, function(p) {
  run_cv_fold(aug_revenue_train_clean, features_aug_A_revenue,
              "Transaction_EV_Revenue", "fold", "rf", p)
})
best_rf_augA_revenue <- rf_params_list[[which.min(rf_cv_augA_revenue)]]

xgb_cv_augA_revenue <- sapply(xgb_params_list, function(p) {
  run_cv_fold(aug_revenue_train_clean, features_aug_A_revenue,
              "Transaction_EV_Revenue", "fold", "xgb", p)
})
best_xgb_augA_revenue <- xgb_params_list[[which.min(xgb_cv_augA_revenue)]]

rf_cv_augB_revenue <- sapply(rf_params_list, function(p) {
  run_cv_fold(aug_revenue_train_clean, features_aug_B_revenue,
              "Transaction_EV_Revenue", "fold", "rf", p)
})
best_rf_augB_revenue <- rf_params_list[[which.min(rf_cv_augB_revenue)]]

xgb_cv_augB_revenue <- sapply(xgb_params_list, function(p) {
  run_cv_fold(aug_revenue_train_clean, features_aug_B_revenue,
              "Transaction_EV_Revenue", "fold", "xgb", p)
})
best_xgb_augB_revenue <- xgb_params_list[[which.min(xgb_cv_augB_revenue)]]
cv_full <- data.frame(
  Target = c(rep("EV/EBITDA", 8), rep("EV/Revenue", 8)),
  Variant = rep(c("Pure", "Pure", "Pure", "Pure",
                  "Augmented", "Augmented", "Augmented", "Augmented"), 2),
  FeatureSet = rep(c("ML-A", "ML-A", "ML-B", "ML-B"), 4),
  Model = rep(c("RF", "XGB"), 8),
  CV_MAE = round(c(
    min(rf_cv_A_ebitda), min(xgb_cv_A_ebitda),
    min(rf_cv_B_ebitda), min(xgb_cv_B_ebitda),
    min(rf_cv_augA_ebitda), min(xgb_cv_augA_ebitda),
    min(rf_cv_augB_ebitda), min(xgb_cv_augB_ebitda),
    min(rf_cv_A_revenue), min(xgb_cv_A_revenue),
    min(rf_cv_B_revenue), min(xgb_cv_B_revenue),
    min(rf_cv_augA_revenue), min(xgb_cv_augA_revenue),
    min(rf_cv_augB_revenue), min(xgb_cv_augB_revenue)), 3))

print(cv_full)
```

```
       Target   Variant FeatureSet Model CV_MAE
1   EV/EBITDA      Pure       ML-A    RF 13.310
2   EV/EBITDA      Pure       ML-A   XGB 13.730
3   EV/EBITDA      Pure       ML-B    RF 13.455
4   EV/EBITDA      Pure       ML-B   XGB 13.793
5   EV/EBITDA Augmented       ML-A    RF 13.498
6   EV/EBITDA Augmented       ML-A   XGB 13.948
7   EV/EBITDA Augmented       ML-B    RF 13.557
8   EV/EBITDA Augmented       ML-B   XGB 14.044
9  EV/Revenue      Pure       ML-A    RF  3.532
10 EV/Revenue      Pure       ML-A   XGB  3.365
11 EV/Revenue      Pure       ML-B    RF  3.340
12 EV/Revenue      Pure       ML-B   XGB  3.336
13 EV/Revenue Augmented       ML-A    RF  3.485
14 EV/Revenue Augmented       ML-A   XGB  3.342
15 EV/Revenue Augmented       ML-B    RF  3.340
16 EV/Revenue Augmented       ML-B   XGB  3.313
```

```
# the parameter set behind each of the sixteen minima, so the retained
# values are read off the run rather than taken on trust

fmt_params <- function(p, model) {
  if (model == "RF") {
    d <- if (is.null(p$max.depth)) "unlimited" else p$max.depth
    paste0("mtry ", p$mtry, ", min.node ", p$min.node.size,
           ", depth ", d)
  } else {
    paste0(p$nrounds, " rounds, depth ", p$max_depth,
           ", eta ", p$eta, ", mcw ", p$min_child_weight)
  }
}

best_list <- list(
  best_rf_A_ebitda,     best_xgb_A_ebitda,
  best_rf_B_ebitda,     best_xgb_B_ebitda,
  best_rf_augA_ebitda,  best_xgb_augA_ebitda,
  best_rf_augB_ebitda,  best_xgb_augB_ebitda,
  best_rf_A_revenue,    best_xgb_A_revenue,
  best_rf_B_revenue,    best_xgb_B_revenue,
  best_rf_augA_revenue, best_xgb_augA_revenue,
  best_rf_augB_revenue, best_xgb_augB_revenue)

cv_full$Retained <- mapply(fmt_params, best_list, cv_full$Model)

print(data.frame(
  Spec = paste(cv_full$Target, cv_full$Variant,
               cv_full$FeatureSet, cv_full$Model),
  CV_MAE = cv_full$CV_MAE,
  Retained = cv_full$Retained), right = FALSE)
```

```
   Spec                          CV_MAE Retained
1  EV/EBITDA Pure ML-A RF        13.310 mtry 4, min.node 20, depth 4
2  EV/EBITDA Pure ML-A XGB       13.730 100 rounds, depth 2, eta 0.05, mcw 10
3  EV/EBITDA Pure ML-B RF        13.455 mtry 8, min.node 20, depth 4
4  EV/EBITDA Pure ML-B XGB       13.793 100 rounds, depth 2, eta 0.05, mcw 10
5  EV/EBITDA Augmented ML-A RF   13.498 mtry 4, min.node 20, depth 4
6  EV/EBITDA Augmented ML-A XGB  13.948 100 rounds, depth 2, eta 0.05, mcw 5
7  EV/EBITDA Augmented ML-B RF   13.557 mtry 4, min.node 20, depth 4
8  EV/EBITDA Augmented ML-B XGB  14.044 100 rounds, depth 2, eta 0.05, mcw 5
9  EV/Revenue Pure ML-A RF        3.532 mtry 4, min.node 20, depth 4
10 EV/Revenue Pure ML-A XGB       3.365 100 rounds, depth 2, eta 0.05, mcw 10
11 EV/Revenue Pure ML-B RF        3.340 mtry 4, min.node 20, depth 4
12 EV/Revenue Pure ML-B XGB       3.336 100 rounds, depth 2, eta 0.05, mcw 10
13 EV/Revenue Augmented ML-A RF   3.485 mtry 4, min.node 20, depth 4
14 EV/Revenue Augmented ML-A XGB  3.342 100 rounds, depth 2, eta 0.05, mcw 10
15 EV/Revenue Augmented ML-B RF   3.340 mtry 4, min.node 20, depth 4
16 EV/Revenue Augmented ML-B XGB  3.313 100 rounds, depth 2, eta 0.05, mcw 10
```

```
write.csv(cv_full, paste0(RES_ML, "ML_CV_Summary_Full.csv"),
          row.names = FALSE)
```

Scoring functions

Every method is measured the same way on the same transactions, so the three scoring helpers are defined once, before anything from the test set is read.

```
metrics <- function(actual, predicted, label) {
  err <- abs(actual - predicted)
  cat("\n---", label, "---\n")
  cat("MAE  :", round(mean(err), 3), "\n")
  cat("RMSE :", round(sqrt(mean((actual - predicted)^2)), 3), "\n")
  cat("MedAE:", round(median(err), 3), "\n")
}

metrics_clean <- function(df, err_col, label) {
  clean <- df[!df$is_extreme, ]
  cat(sprintf("\n--- %s (n=%d excl. extremes) ---\n",
              label, nrow(clean)))
  cat("MAE  :", round(mean(clean[[err_col]]), 3), "\n")
  cat("RMSE :", round(sqrt(mean(clean[[err_col]]^2)), 3), "\n")
  cat("MedAE:", round(median(clean[[err_col]]), 3), "\n")
}

bucket <- function(actual) {
  ifelse(actual < 0, "Negative (artifact)",
  ifelse(actual <= 15, "Typical (0-15x)",
  ifelse(actual <= 50, "Elevated (15-50x)", "Extreme (>50x)")))
}
```

Opening the test set

Every specification decision is now closed: the regression feature set, the ridge penalty, and the retained hyperparameters for all sixteen machine learning models. Nothing below feeds back into a modelling choice, so the test files open at this single point and each method is scored on them once.

```
ebitda_test <- read_excel(
  paste0(DATA, "Reconciliation_ML_Test_v2.xlsx"), sheet = "EBITDA_Test")
revenue_test <- read_excel(
  paste0(DATA, "Reconciliation_ML_Test_v2.xlsx"), sheet = "Revenue_Test")
aug_ebitda_test <- read_excel(
  paste0(DATA, "Reconciliation_ML_Augmented_Test_v2.xlsx"),
  sheet = "EBITDA_Test")
aug_revenue_test <- read_excel(
  paste0(DATA, "Reconciliation_ML_Augmented_Test_v2.xlsx"),
  sheet = "Revenue_Test")

# no company may appear on both sides of the split
cat("entities in both EBITDA train and test:",
    length(intersect(ebitda_train$BvD_ID, ebitda_test$BvD_ID)), "\n")
entities in both EBITDA train and test: 0
cat("entities in both Revenue train and test:",
    length(intersect(revenue_train$BvD_ID, revenue_test$BvD_ID)), "\n")
entities in both Revenue train and test: 0

# the entity structure the grouped split and the grouped folds rest on
all_ids <- c(ebitda_train$BvD_ID, ebitda_test$BvD_ID)
cat("unique entities, EBITDA train and test:",
    length(unique(all_ids)), "\n")
unique entities, EBITDA train and test: 460
cat("of which with more than one transaction:",
    sum(table(all_ids) > 1), "\n")
of which with more than one transaction: 31
```

```
# the augmented frames must be row-aligned with the pure ones, since the
# actuals are taken from one and the identifiers from the other
stopifnot(identical(aug_ebitda_test$Transaction_ID,
                    ebitda_test$Transaction_ID))
stopifnot(identical(aug_revenue_test$Transaction_ID,
                    revenue_test$Transaction_ID))

y_ebitda <- ebitda_test$Transaction_EV_EBITDA
y_revenue <- revenue_test$Transaction_EV_Revenue
```

Regression on the test set

```
pred_ols_ebitda <- predict(model_ols3,
  newdata = ebitda_test[, feature_cols_final])
pred_ols_revenue <- predict(model_ols_revenue,
  newdata = revenue_test[, feature_cols_final])

X_test_ebitda <- as.matrix(ebitda_test[, feature_cols_final])
pred_ridge_ebitda <- as.vector(predict(ridge_cv, newx = X_test_ebitda,
                                       s = "lambda.min"))

pred_aug_ebitda <- predict(model_aug_ebitda,
  newdata = aug_ebitda_test[, feature_cols_aug_ebitda])
pred_aug_revenue <- predict(model_aug_revenue,
  newdata = aug_revenue_test[, feature_cols_aug_revenue])

X_test_revenue <- as.matrix(revenue_test[, feature_cols_final])
pred_ridge_revenue <- as.vector(predict(ridge_cv_rev,
                                        newx = X_test_revenue,
                                        s = "lambda.min"))

metrics(y_ebitda, pred_ols_ebitda, "OLS - EV/EBITDA")
```

```
--- OLS - EV/EBITDA ---
MAE  : 14.213
RMSE : 35.299
MedAE: 7.996
```

```
metrics(y_ebitda, pred_ridge_ebitda, "Ridge - EV/EBITDA")
```

```
--- Ridge - EV/EBITDA ---
MAE  : 14.34
RMSE : 35.433
MedAE: 7.805
```

```
metrics(y_ebitda, pred_aug_ebitda, "OLS Aug - EV/EBITDA")
```

```
--- OLS Aug - EV/EBITDA ---
MAE  : 14.243
RMSE : 35.497
MedAE: 7.239
```

```
metrics(y_revenue, pred_ols_revenue, "OLS - EV/Revenue")
```

```
--- OLS - EV/Revenue ---
MAE  : 3.096
RMSE : 5.743
MedAE: 1.682
```

```
metrics(y_revenue, pred_ridge_revenue, "Ridge - EV/Revenue")
```

```
--- Ridge - EV/Revenue ---
MAE  : 2.917
RMSE : 5.765
MedAE: 2.089
```

```
metrics(y_revenue, pred_aug_revenue, "OLS Aug - EV/Revenue")
```

```
--- OLS Aug - EV/Revenue ---
MAE  : 3.217
RMSE : 5.739
MedAE: 1.937
```

```r
results_ebitda <- data.frame(
  Transaction_ID = ebitda_test$Transaction_ID,
  Actual = y_ebitda,
  OLS_Predicted = pred_ols_ebitda,
  Ridge_Predicted = pred_ridge_ebitda,
  OLS_AbsErr = abs(y_ebitda - pred_ols_ebitda),
  Ridge_AbsErr = abs(y_ebitda - pred_ridge_ebitda),
  OLS_SqErr = (y_ebitda - pred_ols_ebitda)^2,
  Sector = ebitda_test$Sector)

results_revenue <- data.frame(
  Transaction_ID = revenue_test$Transaction_ID,
  Actual = y_revenue,
  OLS_Predicted = pred_ols_revenue,
  Ridge_Predicted = pred_ridge_revenue,
  OLS_AbsErr = abs(y_revenue - pred_ols_revenue),
  Ridge_AbsErr = abs(y_revenue - pred_ridge_revenue),
  OLS_SqErr = (y_revenue - pred_ols_revenue)^2,
  Sector = revenue_test$Sector)

results_aug_ebitda <- data.frame(
  Transaction_ID = aug_ebitda_test$Transaction_ID,
  Actual = y_ebitda,
  AugOLS_Pred = pred_aug_ebitda,
  AugOLS_AbsErr = abs(y_ebitda - pred_aug_ebitda),
  Sector = aug_ebitda_test$Sector)

results_aug_revenue <- data.frame(
  Transaction_ID = aug_revenue_test$Transaction_ID,
  Actual = y_revenue,
  AugOLS_Pred = pred_aug_revenue,
  AugOLS_AbsErr = abs(y_revenue - pred_aug_revenue),
  Sector = aug_revenue_test$Sector)

write.csv(results_ebitda,
          paste0(RES_LR, "Regression_EBITDA_Predictions.csv"),
          row.names = FALSE)
write.csv(results_revenue,
          paste0(RES_LR, "Regression_Revenue_Predictions.csv"),
          row.names = FALSE)
write.csv(results_aug_ebitda,
          paste0(RES_LR, "Regression_Aug_EBITDA_Predictions.csv"),
          row.names = FALSE)
write.csv(results_aug_revenue,
          paste0(RES_LR, "Regression_Aug_Revenue_Predictions.csv"),
          row.names = FALSE)

# per sector and with the extremes taken out
print(aggregate(OLS_AbsErr ~ Sector, data = results_ebitda, FUN = mean))
                                Sector OLS_AbsErr
1                    Commercial Printing   8.051775
2           Diversified Support Services  16.695530
3  Environmental and Facilities Services   8.965640
4 Human Resource and Employment Services  20.389793
5           Office Services and Supplies  41.928847
6       Research and Consulting Services   8.467601
7            Security and Alarm Services   8.688029
print(aggregate(OLS_AbsErr ~ Sector, data = results_revenue, FUN = mean))
                                Sector OLS_AbsErr
1                    Commercial Printing   1.712845
2           Diversified Support Services   3.275368
3  Environmental and Facilities Services   2.081342
4 Human Resource and Employment Services   5.011102
5           Office Services and Supplies   2.225921
6       Research and Consulting Services   3.085767
7            Security and Alarm Services   3.684576
extreme_ebitda <- results_ebitda$Actual > 50 | results_ebitda$Actual < 0
cat("Extreme in EBITDA test:", sum(extreme_ebitda), "\n")
Extreme in EBITDA test: 7
cat("OLS MAE all:", round(mean(results_ebitda$OLS_AbsErr), 3), "\n")
OLS MAE all: 14.213
cat("OLS MAE excl:",
    round(mean(results_ebitda$OLS_AbsErr[!extreme_ebitda]), 3), "\n")
OLS MAE excl: 8.407
```

```
cat("OLS MedAE all:", round(median(results_ebitda$OLS_AbsErr), 3), "\n")
OLS MedAE all: 7.996
cat("OLS MedAE excl:",
    round(median(results_ebitda$OLS_AbsErr[!extreme_ebitda]), 3), "\n")
OLS MedAE excl: 7.603

# same for the augmented specification
extreme_aug <- results_aug_ebitda$Actual > 50 | results_aug_ebitda$Actual < 0
cat("Aug OLS MAE all:",
    round(mean(results_aug_ebitda$AugOLS_AbsErr), 3), "\n")
Aug OLS MAE all: 14.243
cat("Aug OLS MAE excl:",
    round(mean(results_aug_ebitda$AugOLS_AbsErr[!extreme_aug]), 3), "\n")
Aug OLS MAE excl: 8.504
cat("Aug OLS MedAE all:",
    round(median(results_aug_ebitda$AugOLS_AbsErr), 3), "\n")
Aug OLS MedAE all: 7.239
cat("Aug OLS MedAE excl:",
    round(median(results_aug_ebitda$AugOLS_AbsErr[!extreme_aug]), 3), "\n")
Aug OLS MedAE excl: 6.911
```

Buckets for the regression

```
results_ebitda$Bucket <- bucket(results_ebitda$Actual)
results_revenue$Bucket <- bucket(results_revenue$Actual)
results_aug_ebitda$Bucket <- bucket(results_aug_ebitda$Actual)
results_aug_revenue$Bucket <- bucket(results_aug_revenue$Actual)

b1 <- aggregate(cbind(OLS_AbsErr, Actual) ~ Bucket, data = results_ebitda,
                FUN = function(x) round(mean(x), 3))
b1$N <- as.vector(table(results_ebitda$Bucket))
b1$Method <- "OLS Pure"
b1$Target <- "EV/EBITDA"
names(b1)[names(b1) == "OLS_AbsErr"] <- "MAE"

b2 <- aggregate(cbind(AugOLS_AbsErr, Actual) ~ Bucket,
                data = results_aug_ebitda,
                FUN = function(x) round(mean(x), 3))
b2$N <- as.vector(table(results_aug_ebitda$Bucket))
b2$Method <- "OLS Augmented"
b2$Target <- "EV/EBITDA"
names(b2)[names(b2) == "AugOLS_AbsErr"] <- "MAE"

b3 <- aggregate(cbind(OLS_AbsErr, Actual) ~ Bucket, data = results_revenue,
                FUN = function(x) round(mean(x), 3))
b3$N <- as.vector(table(results_revenue$Bucket))
b3$Method <- "OLS Pure"
b3$Target <- "EV/Revenue"
names(b3)[names(b3) == "OLS_AbsErr"] <- "MAE"

b4 <- aggregate(cbind(AugOLS_AbsErr, Actual) ~ Bucket,
                data = results_aug_revenue,
                FUN = function(x) round(mean(x), 3))
b4$N <- as.vector(table(results_aug_revenue$Bucket))
b4$Method <- "OLS Augmented"
b4$Target <- "EV/Revenue"
names(b4)[names(b4) == "AugOLS_AbsErr"] <- "MAE"

bucket_summary <- rbind(b1, b2, b3, b4)
print(bucket_summary)
              Bucket     MAE  Actual   N        Method     Target
1  Elevated (15-50x)  10.812  24.446  20      OLS Pure  EV/EBITDA
2     Extreme (>50x) 111.259 131.336   7      OLS Pure  EV/EBITDA
3    Typical (0-15x)   7.912   7.619  97      OLS Pure  EV/EBITDA
4  Elevated (15-50x)  11.361  24.446  20 OLS Augmented  EV/EBITDA
5     Extreme (>50x) 110.166 131.336   7 OLS Augmented  EV/EBITDA
6    Typical (0-15x)   7.914   7.619  97 OLS Augmented  EV/EBITDA
7  Elevated (15-50x)  25.143  29.882   4      OLS Pure EV/Revenue
8    Typical (0-15x)   2.362   1.892 120      OLS Pure EV/Revenue
9  Elevated (15-50x)  24.466  29.882   4 OLS Augmented EV/Revenue
10   Typical (0-15x)   2.509   1.892 120 OLS Augmented EV/Revenue
```

```
write.csv(bucket_summary,
          paste0(RES_LR, "Bucket_Analysis_Regression.csv"),
          row.names = FALSE)
```

Final ML models, pure

```
rf_final_A_ebitda <- ranger(
  x = as.matrix(ebitda_train[, features_A]),
  y = ebitda_train$Transaction_EV_EBITDA,
  num.trees = 500,
  mtry = best_rf_A_ebitda$mtry,
  min.node.size = best_rf_A_ebitda$min.node.size,
  max.depth = best_rf_A_ebitda$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)

rf_final_B_ebitda <- ranger(
  x = as.matrix(ebitda_train[, features_B]),
  y = ebitda_train$Transaction_EV_EBITDA,
  num.trees = 500,
  mtry = best_rf_B_ebitda$mtry,
  min.node.size = best_rf_B_ebitda$min.node.size,
  max.depth = best_rf_B_ebitda$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)

rf_final_A_revenue <- ranger(
  x = as.matrix(revenue_train_clean[, features_A]),
  y = revenue_train_clean$Transaction_EV_Revenue,
  num.trees = 500,
  mtry = best_rf_A_revenue$mtry,
  min.node.size = best_rf_A_revenue$min.node.size,
  max.depth = best_rf_A_revenue$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)

rf_final_B_revenue <- ranger(
  x = as.matrix(revenue_train_clean[, features_B]),
  y = revenue_train_clean$Transaction_EV_Revenue,
  num.trees = 500,
  mtry = best_rf_B_revenue$mtry,
  min.node.size = best_rf_B_revenue$min.node.size,
  max.depth = best_rf_B_revenue$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)
```

```r
# same reason as in the CV function, xgboost draws its subsamples from
# R's own generator
set.seed(42)
xgb_final_A_ebitda <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_A_ebitda$max_depth,
                eta = best_xgb_A_ebitda$eta,
                min_child_weight = best_xgb_A_ebitda$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(as.matrix(ebitda_train[, features_A]),
                     label = ebitda_train$Transaction_EV_EBITDA),
  nrounds = best_xgb_A_ebitda$nrounds, verbose = 0)

set.seed(42)
xgb_final_B_ebitda <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_B_ebitda$max_depth,
                eta = best_xgb_B_ebitda$eta,
                min_child_weight = best_xgb_B_ebitda$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(as.matrix(ebitda_train[, features_B]),
                     label = ebitda_train$Transaction_EV_EBITDA),
  nrounds = best_xgb_B_ebitda$nrounds, verbose = 0)

set.seed(42)
xgb_final_A_revenue <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_A_revenue$max_depth,
                eta = best_xgb_A_revenue$eta,
                min_child_weight = best_xgb_A_revenue$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(as.matrix(revenue_train_clean[, features_A]),
                     label = revenue_train_clean$Transaction_EV_Revenue),
  nrounds = best_xgb_A_revenue$nrounds, verbose = 0)

set.seed(42)
xgb_final_B_revenue <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_B_revenue$max_depth,
                eta = best_xgb_B_revenue$eta,
                min_child_weight = best_xgb_B_revenue$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(as.matrix(revenue_train_clean[, features_B]),
                     label = revenue_train_clean$Transaction_EV_Revenue),
  nrounds = best_xgb_B_revenue$nrounds, verbose = 0)
pred_rf_A_ebitda <- predict(rf_final_A_ebitda,
  data = as.matrix(ebitda_test[, features_A]))$predictions
pred_rf_B_ebitda <- predict(rf_final_B_ebitda,
  data = as.matrix(ebitda_test[, features_B]))$predictions
pred_xgb_A_ebitda <- predict(xgb_final_A_ebitda,
  xgb.DMatrix(as.matrix(ebitda_test[, features_A])))
pred_xgb_B_ebitda <- predict(xgb_final_B_ebitda,
  xgb.DMatrix(as.matrix(ebitda_test[, features_B])))

metrics(y_ebitda, pred_rf_A_ebitda, "RF ML-A - EV/EBITDA")

--- RF ML-A - EV/EBITDA ---
MAE  : 13.964
RMSE : 34.305
MedAE: 7.698
metrics(y_ebitda, pred_rf_B_ebitda, "RF ML-B - EV/EBITDA")

--- RF ML-B - EV/EBITDA ---
MAE  : 13.784
RMSE : 34.109
MedAE: 7.617
metrics(y_ebitda, pred_xgb_A_ebitda, "XGB ML-A - EV/EBITDA")
```

```
--- XGB ML-A - EV/EBITDA ---
MAE  : 14.753
RMSE : 35.85
MedAE: 7.935
metrics(y_ebitda, pred_xgb_B_ebitda, "XGB ML-B - EV/EBITDA")

--- XGB ML-B - EV/EBITDA ---
MAE  : 14.705
RMSE : 35.478
MedAE: 7.926
pred_rf_A_revenue <- predict(rf_final_A_revenue,
  data = as.matrix(revenue_test[, features_A]))$predictions
pred_rf_B_revenue <- predict(rf_final_B_revenue,
  data = as.matrix(revenue_test[, features_B]))$predictions
pred_xgb_A_revenue <- predict(xgb_final_A_revenue,
  xgb.DMatrix(as.matrix(revenue_test[, features_A])))
pred_xgb_B_revenue <- predict(xgb_final_B_revenue,
  xgb.DMatrix(as.matrix(revenue_test[, features_B])))

metrics(y_revenue, pred_rf_A_revenue, "RF ML-A - EV/Revenue")

--- RF ML-A - EV/Revenue ---
MAE  : 3.65
RMSE : 8.734
MedAE: 1.503
metrics(y_revenue, pred_rf_B_revenue, "RF ML-B - EV/Revenue")

--- RF ML-B - EV/Revenue ---
MAE  : 3.53
RMSE : 8.445
MedAE: 1.636
metrics(y_revenue, pred_xgb_A_revenue, "XGB ML-A - EV/Revenue")

--- XGB ML-A - EV/Revenue ---
MAE  : 3.816
RMSE : 9.083
MedAE: 1.526
metrics(y_revenue, pred_xgb_B_revenue, "XGB ML-B - EV/Revenue")

--- XGB ML-B - EV/Revenue ---
MAE  : 3.927
RMSE : 9.408
MedAE: 1.504
```

```
results_ml_ebitda <- data.frame(
  Transaction_ID = ebitda_test$Transaction_ID,
  Sector = ebitda_test$Sector,
  Actual = y_ebitda,
  RF_A_Pred = pred_rf_A_ebitda,
  RF_B_Pred = pred_rf_B_ebitda,
  XGB_A_Pred = pred_xgb_A_ebitda,
  XGB_B_Pred = pred_xgb_B_ebitda,
  RF_A_AbsErr = abs(y_ebitda - pred_rf_A_ebitda),
  RF_B_AbsErr = abs(y_ebitda - pred_rf_B_ebitda),
  XGB_A_AbsErr = abs(y_ebitda - pred_xgb_A_ebitda),
  XGB_B_AbsErr = abs(y_ebitda - pred_xgb_B_ebitda))
results_ml_ebitda$Bucket <- bucket(results_ml_ebitda$Actual)

results_ml_revenue <- data.frame(
  Transaction_ID = revenue_test$Transaction_ID,
  Sector = revenue_test$Sector,
  Actual = y_revenue,
  RF_A_Pred = pred_rf_A_revenue,
  RF_B_Pred = pred_rf_B_revenue,
  XGB_A_Pred = pred_xgb_A_revenue,
  XGB_B_Pred = pred_xgb_B_revenue,
  RF_A_AbsErr = abs(y_revenue - pred_rf_A_revenue),
  RF_B_AbsErr = abs(y_revenue - pred_rf_B_revenue),
  XGB_A_AbsErr = abs(y_revenue - pred_xgb_A_revenue),
  XGB_B_AbsErr = abs(y_revenue - pred_xgb_B_revenue))
results_ml_revenue$Bucket <- bucket(results_ml_revenue$Actual)

write.csv(results_ml_ebitda,
          paste0(RES_ML, "ML_Pure_EBITDA_Predictions.csv"),
          row.names = FALSE)
write.csv(results_ml_revenue,
          paste0(RES_ML, "ML_Pure_Revenue_Predictions.csv"),
          row.names = FALSE)
```

Final ML models, augmented

```
rf_aug_final_A_ebitda <- ranger(
  x = as.matrix(aug_ebitda_train[, features_aug_A_ebitda]),
  y = aug_ebitda_train$Transaction_EV_EBITDA,
  num.trees = 500, mtry = best_rf_augA_ebitda$mtry,
  min.node.size = best_rf_augA_ebitda$min.node.size,
  max.depth = best_rf_augA_ebitda$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)

rf_aug_final_B_ebitda <- ranger(
  x = as.matrix(aug_ebitda_train[, features_aug_B_ebitda]),
  y = aug_ebitda_train$Transaction_EV_EBITDA,
  num.trees = 500, mtry = best_rf_augB_ebitda$mtry,
  min.node.size = best_rf_augB_ebitda$min.node.size,
  max.depth = best_rf_augB_ebitda$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)

rf_aug_final_A_revenue <- ranger(
  x = as.matrix(aug_revenue_train_clean[, features_aug_A_revenue]),
  y = aug_revenue_train_clean$Transaction_EV_Revenue,
  num.trees = 500, mtry = best_rf_augA_revenue$mtry,
  min.node.size = best_rf_augA_revenue$min.node.size,
  max.depth = best_rf_augA_revenue$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)

rf_aug_final_B_revenue <- ranger(
  x = as.matrix(aug_revenue_train_clean[, features_aug_B_revenue]),
  y = aug_revenue_train_clean$Transaction_EV_Revenue,
  num.trees = 500, mtry = best_rf_augB_revenue$mtry,
  min.node.size = best_rf_augB_revenue$min.node.size,
  max.depth = best_rf_augB_revenue$max.depth,
  importance = "permutation", seed = 42, num.threads = 1)
```

```
set.seed(42)
xgb_aug_final_A_ebitda <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_augA_ebitda$max_depth,
                eta = best_xgb_augA_ebitda$eta,
                min_child_weight = best_xgb_augA_ebitda$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(as.matrix(aug_ebitda_train[, features_aug_A_ebitda]),
                     label = aug_ebitda_train$Transaction_EV_EBITDA),
  nrounds = best_xgb_augA_ebitda$nrounds, verbose = 0)

set.seed(42)
xgb_aug_final_B_ebitda <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_augB_ebitda$max_depth,
                eta = best_xgb_augB_ebitda$eta,
                min_child_weight = best_xgb_augB_ebitda$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(as.matrix(aug_ebitda_train[, features_aug_B_ebitda]),
                     label = aug_ebitda_train$Transaction_EV_EBITDA),
  nrounds = best_xgb_augB_ebitda$nrounds, verbose = 0)

set.seed(42)
xgb_aug_final_A_revenue <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_augA_revenue$max_depth,
                eta = best_xgb_augA_revenue$eta,
                min_child_weight = best_xgb_augA_revenue$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(
    as.matrix(aug_revenue_train_clean[, features_aug_A_revenue]),
    label = aug_revenue_train_clean$Transaction_EV_Revenue),
  nrounds = best_xgb_augA_revenue$nrounds, verbose = 0)

set.seed(42)
xgb_aug_final_B_revenue <- xgb.train(
  params = list(objective = "reg:squarederror",
                max_depth = best_xgb_augB_revenue$max_depth,
                eta = best_xgb_augB_revenue$eta,
                min_child_weight = best_xgb_augB_revenue$min_child_weight,
                subsample = 0.8, colsample_bytree = 0.8,
                seed = 42, nthread = 1),
  data = xgb.DMatrix(
    as.matrix(aug_revenue_train_clean[, features_aug_B_revenue]),
    label = aug_revenue_train_clean$Transaction_EV_Revenue),
  nrounds = best_xgb_augB_revenue$nrounds, verbose = 0)
pred_rf_augA_ebitda <- predict(rf_aug_final_A_ebitda,
  data = as.matrix(aug_ebitda_test[, features_aug_A_ebitda]))$predictions
pred_rf_augB_ebitda <- predict(rf_aug_final_B_ebitda,
  data = as.matrix(aug_ebitda_test[, features_aug_B_ebitda]))$predictions
pred_xgb_augA_ebitda <- predict(xgb_aug_final_A_ebitda,
  xgb.DMatrix(as.matrix(aug_ebitda_test[, features_aug_A_ebitda])))
pred_xgb_augB_ebitda <- predict(xgb_aug_final_B_ebitda,
  xgb.DMatrix(as.matrix(aug_ebitda_test[, features_aug_B_ebitda])))

metrics(y_ebitda, pred_rf_augA_ebitda, "RF Aug-A - EV/EBITDA")

--- RF Aug-A - EV/EBITDA ---
MAE  : 13.532
RMSE : 34.245
MedAE: 7.166
metrics(y_ebitda, pred_rf_augB_ebitda, "RF Aug-B - EV/EBITDA")

--- RF Aug-B - EV/EBITDA ---
MAE  : 13.579
RMSE : 34.603
MedAE: 7.41
metrics(y_ebitda, pred_xgb_augA_ebitda, "XGB Aug-A - EV/EBITDA")
```

```
--- XGB Aug-A - EV/EBITDA ---
MAE  : 13.599
RMSE : 34.036
MedAE: 6.657
metrics(y_ebitda, pred_xgb_augB_ebitda, "XGB Aug-B - EV/EBITDA")

--- XGB Aug-B - EV/EBITDA ---
MAE  : 13.356
RMSE : 33.751
MedAE: 6.338
pred_rf_augA_revenue <- predict(rf_aug_final_A_revenue,
  data = as.matrix(aug_revenue_test[, features_aug_A_revenue]))$predictions
pred_rf_augB_revenue <- predict(rf_aug_final_B_revenue,
  data = as.matrix(aug_revenue_test[, features_aug_B_revenue]))$predictions
pred_xgb_augA_revenue <- predict(xgb_aug_final_A_revenue,
  xgb.DMatrix(as.matrix(aug_revenue_test[, features_aug_A_revenue])))
pred_xgb_augB_revenue <- predict(xgb_aug_final_B_revenue,
  xgb.DMatrix(as.matrix(aug_revenue_test[, features_aug_B_revenue])))

metrics(y_revenue, pred_rf_augA_revenue, "RF Aug-A - EV/Revenue")

--- RF Aug-A - EV/Revenue ---
MAE  : 3.627
RMSE : 8.712
MedAE: 1.51
metrics(y_revenue, pred_rf_augB_revenue, "RF Aug-B - EV/Revenue")

--- RF Aug-B - EV/Revenue ---
MAE  : 3.491
RMSE : 8.006
MedAE: 1.587
metrics(y_revenue, pred_xgb_augA_revenue, "XGB Aug-A - EV/Revenue")

--- XGB Aug-A - EV/Revenue ---
MAE  : 3.915
RMSE : 9.296
MedAE: 1.462
metrics(y_revenue, pred_xgb_augB_revenue, "XGB Aug-B - EV/Revenue")

--- XGB Aug-B - EV/Revenue ---
MAE  : 3.927
RMSE : 9.41
MedAE: 1.476
```

```
results_aug_ml_ebitda <- data.frame(
  Transaction_ID = aug_ebitda_test$Transaction_ID,
  Sector = aug_ebitda_test$Sector,
  Actual = y_ebitda,
  RF_A_Pred = pred_rf_augA_ebitda,
  RF_B_Pred = pred_rf_augB_ebitda,
  XGB_A_Pred = pred_xgb_augA_ebitda,
  XGB_B_Pred = pred_xgb_augB_ebitda,
  RF_A_AbsErr = abs(y_ebitda - pred_rf_augA_ebitda),
  RF_B_AbsErr = abs(y_ebitda - pred_rf_augB_ebitda),
  XGB_A_AbsErr = abs(y_ebitda - pred_xgb_augA_ebitda),
  XGB_B_AbsErr = abs(y_ebitda - pred_xgb_augB_ebitda))
results_aug_ml_ebitda$Bucket <- bucket(results_aug_ml_ebitda$Actual)

results_aug_ml_revenue <- data.frame(
  Transaction_ID = aug_revenue_test$Transaction_ID,
  Sector = aug_revenue_test$Sector,
  Actual = y_revenue,
  RF_A_Pred = pred_rf_augA_revenue,
  RF_B_Pred = pred_rf_augB_revenue,
  XGB_A_Pred = pred_xgb_augA_revenue,
  XGB_B_Pred = pred_xgb_augB_revenue,
  RF_A_AbsErr = abs(y_revenue - pred_rf_augA_revenue),
  RF_B_AbsErr = abs(y_revenue - pred_rf_augB_revenue),
  XGB_A_AbsErr = abs(y_revenue - pred_xgb_augA_revenue),
  XGB_B_AbsErr = abs(y_revenue - pred_xgb_augB_revenue))
results_aug_ml_revenue$Bucket <- bucket(results_aug_ml_revenue$Actual)

write.csv(results_aug_ml_ebitda,
          paste0(RES_ML, "ML_Aug_EBITDA_Predictions.csv"),
          row.names = FALSE)
write.csv(results_aug_ml_revenue,
          paste0(RES_ML, "ML_Aug_Revenue_Predictions.csv"),
          row.names = FALSE)
```

Buckets and extremes

```
ml_bucket_ebitda <- aggregate(
  cbind(RF_A_AbsErr, RF_B_AbsErr, XGB_A_AbsErr, XGB_B_AbsErr) ~ Bucket,
  data = results_ml_ebitda, FUN = function(x) round(mean(x), 3))
ml_bucket_ebitda$N <- as.vector(table(results_ml_ebitda$Bucket))
print(ml_bucket_ebitda)
```

```
             Bucket RF_A_AbsErr RF_B_AbsErr XGB_A_AbsErr XGB_B_AbsErr  N
1 Elevated (15-50x)      10.962      11.310       10.880       11.895 20
2    Extreme (>50x)     106.776     106.774      112.208      110.259  7
3   Typical (0-15x)       7.886       7.584        8.518        8.389 97
ml_bucket_revenue <- aggregate(
  cbind(RF_A_AbsErr, RF_B_AbsErr, XGB_A_AbsErr, XGB_B_AbsErr) ~ Bucket,
  data = results_ml_revenue, FUN = function(x) round(mean(x), 3))
ml_bucket_revenue$N <- as.vector(table(results_ml_revenue$Bucket))
print(ml_bucket_revenue)
```

```
             Bucket RF_A_AbsErr RF_B_AbsErr XGB_A_AbsErr XGB_B_AbsErr   N
1 Elevated (15-50x)      26.779      26.784       28.440       27.771   4
2   Typical (0-15x)       2.879       2.755        2.995        3.132 120
aug_bucket_ebitda <- aggregate(
  cbind(RF_A_AbsErr, RF_B_AbsErr, XGB_A_AbsErr, XGB_B_AbsErr) ~ Bucket,
  data = results_aug_ml_ebitda, FUN = function(x) round(mean(x), 3))
aug_bucket_ebitda$N <- as.vector(table(results_aug_ml_ebitda$Bucket))
print(aug_bucket_ebitda)
```

```
             Bucket RF_A_AbsErr RF_B_AbsErr XGB_A_AbsErr XGB_B_AbsErr  N
1 Elevated (15-50x)       9.107       8.980       10.531       10.760 20
2    Extreme (>50x)     106.773     110.457      102.677      103.202  7
3   Typical (0-15x)       7.715       7.536        7.804        7.407 97
aug_bucket_revenue <- aggregate(
  cbind(RF_A_AbsErr, RF_B_AbsErr, XGB_A_AbsErr, XGB_B_AbsErr) ~ Bucket,
  data = results_aug_ml_revenue, FUN = function(x) round(mean(x), 3))
aug_bucket_revenue$N <- as.vector(table(results_aug_ml_revenue$Bucket))
print(aug_bucket_revenue)
```

```
             Bucket RF_A_AbsErr RF_B_AbsErr XGB_A_AbsErr XGB_B_AbsErr   N
1 Elevated (15-50x)      26.420      26.667       28.083       27.404   4
2   Typical (0-15x)       2.867       2.718        3.109        3.144 120
```

```
write.csv(ml_bucket_ebitda, paste0(RES_ML, "ML_Bucket_EBITDA.csv"),
          row.names = FALSE)
write.csv(ml_bucket_revenue, paste0(RES_ML, "ML_Bucket_Revenue.csv"),
          row.names = FALSE)
write.csv(aug_bucket_ebitda, paste0(RES_ML, "ML_Aug_Bucket_EBITDA.csv"),
          row.names = FALSE)
write.csv(aug_bucket_revenue, paste0(RES_ML, "ML_Aug_Bucket_Revenue.csv"),
          row.names = FALSE)

# same metrics with the seven extremes taken out
results_ml_ebitda$is_extreme <- results_ml_ebitda$Actual > 50 |
                                results_ml_ebitda$Actual < 0
results_aug_ml_ebitda$is_extreme <- results_aug_ml_ebitda$Actual > 50 |
                                    results_aug_ml_ebitda$Actual < 0

metrics_clean(results_ml_ebitda, "RF_A_AbsErr", "RF ML-A Pure")
```

```
--- RF ML-A Pure (n=117 excl. extremes) ---
MAE  : 8.412
RMSE : 10.389
MedAE: 7.587
```

```
metrics_clean(results_ml_ebitda, "RF_B_AbsErr", "RF ML-B Pure")
```

```
--- RF ML-B Pure (n=117 excl. extremes) ---
MAE  : 8.22
RMSE : 10.325
MedAE: 7.261
```

```
metrics_clean(results_ml_ebitda, "XGB_A_AbsErr", "XGB ML-A Pure")
```

```
--- XGB ML-A Pure (n=117 excl. extremes) ---
MAE  : 8.922
RMSE : 11.55
MedAE: 7.363
```

```
metrics_clean(results_ml_ebitda, "XGB_B_AbsErr", "XGB ML-B Pure")
```

```
--- XGB ML-B Pure (n=117 excl. extremes) ---
MAE  : 8.988
RMSE : 11.815
MedAE: 7.642
```

```
metrics_clean(results_aug_ml_ebitda, "RF_A_AbsErr", "RF Aug-A")
```

```
--- RF Aug-A (n=117 excl. extremes) ---
MAE  : 7.953
RMSE : 9.897
MedAE: 6.923
```

```
metrics_clean(results_aug_ml_ebitda, "RF_B_AbsErr", "RF Aug-B")
```

```
--- RF Aug-B (n=117 excl. extremes) ---
MAE  : 7.783
RMSE : 9.51
MedAE: 7.162
```

```
metrics_clean(results_aug_ml_ebitda, "XGB_A_AbsErr", "XGB Aug-A")
```

```
--- XGB Aug-A (n=117 excl. extremes) ---
MAE  : 8.27
RMSE : 11.218
MedAE: 6.021
```

```
metrics_clean(results_aug_ml_ebitda, "XGB_B_AbsErr", "XGB Aug-B")
```

```
--- XGB Aug-B (n=117 excl. extremes) ---
MAE  : 7.98
RMSE : 10.888
MedAE: 5.891
```

Feature importance

```
# permutation for RF and gain for XGB, the two are on different scales so
# each ranking is read on its own

imp_rf_A_ebitda <- sort(importance(rf_final_A_ebitda), decreasing = TRUE)
imp_rf_B_ebitda <- sort(importance(rf_final_B_ebitda), decreasing = TRUE)
imp_rf_A_revenue <- sort(importance(rf_final_A_revenue),
                         decreasing = TRUE)
imp_rf_B_revenue <- sort(importance(rf_final_B_revenue),
                         decreasing = TRUE)

imp_rf_augA_ebitda <- sort(importance(rf_aug_final_A_ebitda),
                           decreasing = TRUE)
imp_rf_augB_ebitda <- sort(importance(rf_aug_final_B_ebitda),
                           decreasing = TRUE)
imp_rf_augA_revenue <- sort(importance(rf_aug_final_A_revenue),
                            decreasing = TRUE)
imp_rf_augB_revenue <- sort(importance(rf_aug_final_B_revenue),
                            decreasing = TRUE)

print(round(imp_rf_A_ebitda, 3))
```

```
      EBIT_margin_CY1     EBITDA_margin_CY1  Log_total_assets_CY1     Volatility_EBITDA
               40.161               29.982               12.656               11.904
 Sponsor_backed_dummy     Log_employees_CY1              ROA_CY1 CAGR_revenue_fallback
                7.973                1.009               -1.355               -1.568
     Equity_ratio_CY1  Growth_accel_revenue  CAGR_EBITDA_fallback
               -2.704               -3.383               -4.422
```

```
print(round(imp_rf_augA_ebitda, 3))
```

```
Comps_Median_EV_EBITDA    Rate_10Y_at_closing        EBIT_margin_CY1
                33.927                 20.553                 14.903
  Sponsor_backed_dummy               ROA_CY1      EBITDA_margin_CY1
                13.954                 11.355                  6.415
   STOXX600_at_closing   Log_total_assets_CY1      Volatility_EBITDA
                 1.880                  1.571                 -0.768
  CAGR_EBITDA_fallback   Growth_accel_revenue  CAGR_revenue_fallback
                -1.362                 -2.830                 -4.288
     Equity_ratio_CY1      Log_employees_CY1
                -4.971                -13.080
```

```
print(round(imp_rf_A_revenue, 3))
```

```
      EBIT_margin_CY1     EBITDA_margin_CY1              ROA_CY1     Log_employees_CY1
               24.375               22.878                2.929                1.840
 Sponsor_backed_dummy  Growth_accel_revenue     Volatility_EBITDA      Equity_ratio_CY1
                1.313                0.088               -0.036               -1.049
 Log_total_assets_CY1 CAGR_revenue_fallback  CAGR_EBITDA_fallback
               -2.395               -3.581               -6.993
```

```
imp_xgb_A_ebitda <- xgb.importance(feature_names = features_A,
                                   model = xgb_final_A_ebitda)
imp_xgb_B_ebitda <- xgb.importance(feature_names = features_B,
                                   model = xgb_final_B_ebitda)
imp_xgb_A_revenue <- xgb.importance(feature_names = features_A,
                                    model = xgb_final_A_revenue)
imp_xgb_B_revenue <- xgb.importance(feature_names = features_B,
                                    model = xgb_final_B_revenue)

imp_xgb_augA_ebitda <- xgb.importance(
  feature_names = features_aug_A_ebitda,
  model = xgb_aug_final_A_ebitda)
imp_xgb_augB_ebitda <- xgb.importance(
  feature_names = features_aug_B_ebitda,
  model = xgb_aug_final_B_ebitda)
imp_xgb_augA_revenue <- xgb.importance(
  feature_names = features_aug_A_revenue,
  model = xgb_aug_final_A_revenue)
imp_xgb_augB_revenue <- xgb.importance(
  feature_names = features_aug_B_revenue,
  model = xgb_aug_final_B_revenue)

print(imp_xgb_A_ebitda)
```

	Feature	Gain	Cover	Frequency
	<char>	<num>	<num>	<num>
1:	Log_total_assets_CY1	0.26173130	0.24927206	0.23550725
2:	Log_employees_CY1	0.25345378	0.17591568	0.17028986
3:	Sponsor_backed_dummy	0.08525010	0.09877910	0.07246377
4:	Equity_ratio_CY1	0.08177562	0.10082245	0.10869565
5:	CAGR_revenue_fallback	0.07857228	0.07260715	0.09420290
6:	EBITDA_margin_CY1	0.05385879	0.06373559	0.05434783
7:	CAGR_EBITDA_fallback	0.05219878	0.08059325	0.07971014
8:	ROA_CY1	0.04401306	0.04205902	0.04710145
9:	Volatility_EBITDA	0.03464239	0.05574948	0.05434783
10:	EBIT_margin_CY1	0.03445797	0.04362559	0.04347826
11:	Growth_accel_revenue	0.02004593	0.01684064	0.03985507

```
print(imp_xgb_augA_ebitda)
```

	Feature	Gain	Cover	Frequency
	<char>	<num>	<num>	<num>
1:	Log_total_assets_CY1	0.249604423	0.239392396	0.21245421
2:	Equity_ratio_CY1	0.126631544	0.060081466	0.09890110
3:	Rate_10Y_at_closing	0.124266471	0.125865580	0.11721612
4:	Log_employees_CY1	0.110609726	0.111982349	0.10256410
5:	STOXX600_at_closing	0.097110576	0.064935506	0.08058608
6:	Comps_Median_EV_EBITDA	0.096968060	0.112813985	0.09890110
7:	Sponsor_backed_dummy	0.075256128	0.082942974	0.07692308
8:	EBIT_margin_CY1	0.026423132	0.037135098	0.03296703
9:	EBITDA_margin_CY1	0.024116271	0.028360489	0.02930403
10:	Volatility_EBITDA	0.020368084	0.049066531	0.04029304
11:	CAGR_revenue_fallback	0.015537146	0.042786830	0.04029304
12:	ROA_CY1	0.014616419	0.012661236	0.01831502
13:	CAGR_EBITDA_fallback	0.012573643	0.024711473	0.02564103
14:	Growth_accel_revenue	0.005918376	0.007264087	0.02564103

```
print(imp_xgb_A_revenue)
```

	Feature	Gain	Cover	Frequency
	<char>	<num>	<num>	<num>
1:	Log_employees_CY1	0.332163308	0.3163887410	0.318627451
2:	CAGR_EBITDA_fallback	0.326004362	0.2600078998	0.259803922
3:	CAGR_revenue_fallback	0.296325634	0.2661903862	0.259803922
4:	EBITDA_margin_CY1	0.015622122	0.0005667279	0.004901961
5:	Sponsor_backed_dummy	0.011396213	0.0447199849	0.044117647
6:	Growth_accel_revenue	0.007280728	0.0492709818	0.049019608
7:	EBIT_margin_CY1	0.003843101	0.0050490306	0.004901961
8:	Log_total_assets_CY1	0.003237326	0.0292637689	0.029411765
9:	Volatility_EBITDA	0.002164455	0.0142884130	0.014705882
10:	ROA_CY1	0.001962750	0.0142540658	0.014705882

```
write.csv(data.frame(Variable = names(imp_rf_A_ebitda),
                     Importance = imp_rf_A_ebitda),
          paste0(RES_IMP, "Importance_RF_A_EBITDA.csv"),
          row.names = FALSE)
write.csv(data.frame(Variable = names(imp_rf_B_ebitda),
                     Importance = imp_rf_B_ebitda),
          paste0(RES_IMP, "Importance_RF_B_EBITDA.csv"),
          row.names = FALSE)
write.csv(data.frame(Variable = names(imp_rf_A_revenue),
                     Importance = imp_rf_A_revenue),
          paste0(RES_IMP, "Importance_RF_A_Revenue.csv"),
          row.names = FALSE)
write.csv(data.frame(Variable = names(imp_rf_B_revenue),
                     Importance = imp_rf_B_revenue),
          paste0(RES_IMP, "Importance_RF_B_Revenue.csv"),
          row.names = FALSE)
write.csv(data.frame(Variable = names(imp_rf_augA_ebitda),
                     Importance = imp_rf_augA_ebitda),
          paste0(RES_IMP, "Importance_RF_augA_EBITDA.csv"),
          row.names = FALSE)
write.csv(data.frame(Variable = names(imp_rf_augB_ebitda),
                     Importance = imp_rf_augB_ebitda),
          paste0(RES_IMP, "Importance_RF_augB_EBITDA.csv"),
          row.names = FALSE)
write.csv(data.frame(Variable = names(imp_rf_augA_revenue),
                     Importance = imp_rf_augA_revenue),
          paste0(RES_IMP, "Importance_RF_augA_Revenue.csv"),
          row.names = FALSE)
write.csv(data.frame(Variable = names(imp_rf_augB_revenue),
                     Importance = imp_rf_augB_revenue),
          paste0(RES_IMP, "Importance_RF_augB_Revenue.csv"),
          row.names = FALSE)

write.csv(imp_xgb_A_ebitda,
          paste0(RES_IMP, "Importance_XGB_A_EBITDA.csv"),
          row.names = FALSE)
write.csv(imp_xgb_B_ebitda,
          paste0(RES_IMP, "Importance_XGB_B_EBITDA.csv"),
          row.names = FALSE)
write.csv(imp_xgb_A_revenue,
          paste0(RES_IMP, "Importance_XGB_A_Revenue.csv"),
          row.names = FALSE)
write.csv(imp_xgb_B_revenue,
          paste0(RES_IMP, "Importance_XGB_B_Revenue.csv"),
          row.names = FALSE)
write.csv(imp_xgb_augA_ebitda,
          paste0(RES_IMP, "Importance_XGB_augA_EBITDA.csv"),
          row.names = FALSE)
write.csv(imp_xgb_augB_ebitda,
          paste0(RES_IMP, "Importance_XGB_augB_EBITDA.csv"),
          row.names = FALSE)
write.csv(imp_xgb_augA_revenue,
          paste0(RES_IMP, "Importance_XGB_augA_Revenue.csv"),
          row.names = FALSE)
write.csv(imp_xgb_augB_revenue,
          paste0(RES_IMP, "Importance_XGB_augB_Revenue.csv"),
          row.names = FALSE)
```

Analyst benchmark

The benchmark was built in Excel. Comparable-set medians were computed per sector and closing year with a PivotTable using MEDIAN as the summary statistic, after trimming each cell with QUARTILE.INC to obtain Q1 and Q3 and excluding observations outside Q1 minus 1.5 IQR and Q3 plus 1.5 IQR. The discount was calibrated on the training sheet with a column of candidate values from 0 to 60 per cent in steps of 0.1, the median absolute error evaluated for each through a MEDIAN and IF array formula, and the optimum selected with MATCH on MIN. That is 84 sector-year cells and 601 candidate discounts; Tables 9 and 10 report the two calibration criteria and the test-set sensitivity, computed on those sheets. The calibrated values were applied unchanged to the test sheet. What follows is the scoring of that output in R, so that the benchmark and the models are measured the same way on the same transactions.

```
analyst_ebitda <- read_excel(
  paste0(DATA, "Methode_Analyste_TrainTest.xlsx"), sheet = "EBITDA_Test")
analyst_revenue <- read_excel(
  paste0(DATA, "Methode_Analyste_TrainTest.xlsx"), sheet = "Revenue_Test")

analyst_ebitda$AbsErr <- abs(
  analyst_ebitda$Transaction_EV_EBITDA -
  analyst_ebitda$Comps_Median_EV_EBITDA * (1 - 0.215))
analyst_revenue$AbsErr <- abs(
  analyst_revenue$Transaction_EV_Revenue -
  analyst_revenue$Comps_Median_EV_Revenue * (1 - 0.035))

analyst_ebitda$Bucket <- bucket(analyst_ebitda$Transaction_EV_EBITDA)
analyst_revenue$Bucket <- bucket(analyst_revenue$Transaction_EV_Revenue)
analyst_ebitda$is_extreme <-
  analyst_ebitda$Transaction_EV_EBITDA > 50 |
  analyst_ebitda$Transaction_EV_EBITDA < 0

cat("EBITDA  MAE:", round(mean(analyst_ebitda$AbsErr), 3),
    "| RMSE:", round(sqrt(mean(analyst_ebitda$AbsErr^2)), 3),
    "| MedAE:", round(median(analyst_ebitda$AbsErr), 3), "\n")
EBITDA  MAE: 12.373 | RMSE: 36.867 | MedAE: 4.072

cat("Revenue MAE:", round(mean(analyst_revenue$AbsErr), 3),
    "| RMSE:", round(sqrt(mean(analyst_revenue$AbsErr^2)), 3),
    "| MedAE:", round(median(analyst_revenue$AbsErr), 3), "\n")
Revenue MAE: 2.251 | RMSE: 6.167 | MedAE: 0.576
ae_clean <- analyst_ebitda$AbsErr[!analyst_ebitda$is_extreme]

cat("EBITDA excl. extremes, n =", length(ae_clean),
    "| MAE:", round(mean(ae_clean), 3),
    "| RMSE:", round(sqrt(mean(ae_clean^2)), 3),
    "| MedAE:", round(median(ae_clean), 3), "\n")
EBITDA excl. extremes, n = 117 | MAE: 5.697 | RMSE: 9.003 | MedAE: 3.808

print(aggregate(cbind(AbsErr, Transaction_EV_EBITDA) ~ Bucket,
                data = analyst_ebitda,
                FUN = function(x) round(mean(x), 3)))

             Bucket  AbsErr Transaction_EV_EBITDA
1 Elevated (15-50x)  17.187               24.446
2    Extreme (>50x) 123.963              131.336
3   Typical (0-15x)   3.328                7.619
print(table(analyst_ebitda$Bucket))

Elevated (15-50x)    Extreme (>50x)   Typical (0-15x)
               20                 7                97

print(aggregate(cbind(AbsErr, Transaction_EV_Revenue) ~ Bucket,
                data = analyst_revenue,
                FUN = function(x) round(mean(x), 3)))

             Bucket AbsErr Transaction_EV_Revenue
1 Elevated (15-50x) 29.055                29.882
2   Typical (0-15x)  1.358                 1.892
print(table(analyst_revenue$Bucket))

Elevated (15-50x)   Typical (0-15x)
                4               120
e_sec <- aggregate(AbsErr ~ Sector, data = analyst_ebitda,
                   FUN = function(x) round(mean(x), 3))
e_sec$N <- as.vector(table(analyst_ebitda$Sector))

print(e_sec)
                                Sector AbsErr  N
1                   Commercial Printing  8.040  6
2           Diversified Support Services 15.037 34
3  Environmental and Facilities Services  6.498 23
4 Human Resource and Employment Services 20.412 13
5           Office Services and Supplies 45.539  8
6       Research and Consulting Services  4.995 35
7            Security and Alarm Services  4.161  5
```

```
r_sec <- aggregate(AbsErr ~ Sector, data = analyst_revenue,
                   FUN = function(x) round(mean(x), 3))
r_sec$N <- as.vector(table(analyst_revenue$Sector))
print(r_sec)
                                Sector AbsErr  N
1                   Commercial Printing  1.700  6
2           Diversified Support Services  1.602 32
3  Environmental and Facilities Services  0.964 22
4 Human Resource and Employment Services  4.603 15
5           Office Services and Supplies  2.652  8
6       Research and Consulting Services  2.514 35
7            Security and Alarm Services  3.037  6

# the Office Services error is carried by two transactions above 50x
office <- analyst_ebitda[
  analyst_ebitda$Sector == "Office Services and Supplies", ]

cat("the transactions above 50x:",
    paste(round(sort(office$Transaction_EV_EBITDA[office$is_extreme],
                     decreasing = TRUE), 1), collapse = ", "), "\n")
the transactions above 50x: 255.4, 66.3

cat("excluding them, n =", sum(!office$is_extreme),
    "| MAE:", round(mean(office$AbsErr[!office$is_extreme]), 3), "\n")
excluding them, n = 6 | MAE: 9.33
```

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Louvain School of Management

Place des Doyens, 1 bte L2.01.01, 1348 Louvain-la-Neuve
Boulevard Emile Devreux 6, 6000 Charleroi, Belgique
Chaussée de Binche 151, 7000 Mons, Belgique

www.uclouvain.be/lsm