How can advanced business analytics applied to high-frequency and cross-sectional bond market data improve predictive models for the pricing and issuance timing of corporate debt securities in public capital markets?

Autor: Diego Baltar Arriola

5º, E-3 Analytics

Grado de Business Analytics

Madrid

April 2026

INDEX

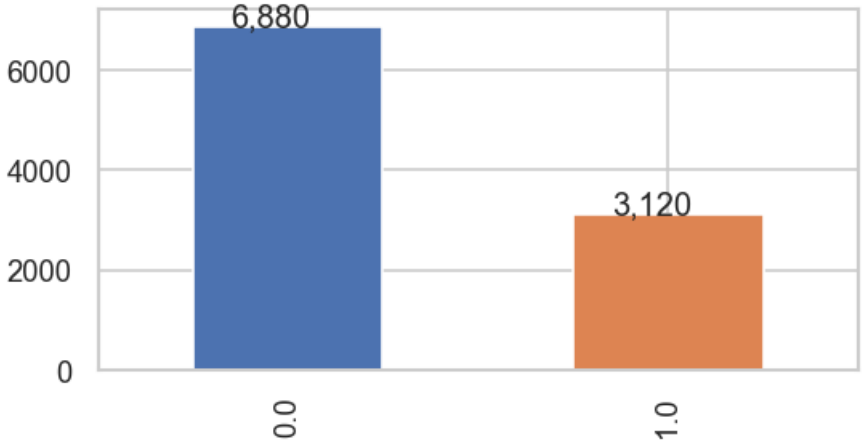
1. INTRODUCTION

Corporate borrowers and their investment banks often face a critical strategic question: when is the optimal time to issue new bonds? Issuing at the right moment can lock in favorable interest spreads and meet strong investor demand, thereby reducing financing costs. Traditionally, firms have relied on broad macroeconomic indicators (like interest rate levels, central bank policy signals or job reports from the US) and aggregate credit spread indices to gauge market conditions. However, they have seldom exploited the rich microstructure signals available in modern bond markets, such as intraday trading patterns, order flow imbalances, or dealer quote dynamics, that might provide early warnings of short-term fluctuations in borrowing costs and remain almost entirely unexploited in issuance planning.

In this study I aim to bridge that gap by integrating advanced business analytics techniques with granular bond market data to help issuers time and price corporate debt offerings more optimally. The corporate bond market has undergone a structural transformation over the past decade. As Fermanian, Guéant and Pu (2015) document, the corporate bond market has been undergoing a structural shift away from voice-driven, over-the-counter trading toward electronic multi-dealer platforms, where buy-side agents can simultaneously request quotes from multiple dealers and every transaction is recorded; a transformation that has generated an unprecedented volume of granular, real-time microstructure data. Regulatory initiatives have reinforced this transparency: the FINRA TRACE system in the United States "required to report all trades in publicly issued corporate bonds to the National Association of Security Dealers, which in turn made transaction data available to the public" (Bessembinder & Maxwell, 2008, p. 218), providing academics and practitioners alike with an unprecedented view of secondary market microstructure. The question, then, is not whether rich intraday data exists but whether it can be harnessed to improve a decision that has traditionally been made on intuition and low-frequency signals.

A first indication of the empirical opportunity is visible in figure below, which shows the distribution of the target variable (the binary classification of a given market day as an optimal or sub-optimal issuance window) across the full intraday dataset of 10,000 observations. Favorable windows (class 1) represent 31.2% of all observations,

confirming that such windows are not uniformly distributed across time. This imbalance itself encodes an economic insight: if all days were equally favorable, there would be no timing value to capture. The fact that fewer than one-third of observations correspond to optimal conditions means that an issuer with a reliable classifier could meaningfully improve its expected cost of issuance by concentrating transactions in those windows.



Distribution of the Target Variable (optimal_issuance_window) across the intraday dataset (n=10,000). Class 0 — Non-Optimal: 6,880 obs. (68.8%); Class 1 — Optimal: 3,120 obs. (31.2%). Source: own elaboration based on TFG_AnalyticaBondMarket dataset.

What this thesis adds is a new analytical dimension. The existing literature on debt issuance timing focuses predominantly on low-frequency, cross-sectional patterns: why do some firms issue in certain months, at certain phases of the credit cycle, or in response to rating changes? These are important questions, but they leave open a different one, can the intraday microstructure of the bond market, observed continuously, tell us something about the specific days or windows within which issuance conditions are most favorable? This is the gap this work addresses: building a predictive, machine-learning-based framework that uses high-frequency cross-sectional bond market signals to classify market states as optimal or sub-optimal for new issuance.

2. RESEARCH QUESTION

The question guiding this work can be stated concisely: *to what extent can intraday, high-frequency bond market microstructure signals improve the prediction of optimal corporate debt issuance windows, and which machine learning architectures best capture the non-linear structure of those signals?*

This formulation deliberately focuses on prediction rather than explanation. The thesis does not seek to establish a new theory of corporate debt issuance; it seeks to test whether a set of observable, high-frequency market conditions (bid-ask spreads, order flow imbalances, market depth, dealer competition, and intraday volatility) carry enough information to classify a given market state as favorable for a new bond transaction. It is an empirical question, and answering it requires both a carefully constructed dataset and a rigorous comparison of predictive models.

The research question also implies a second-order question about model architecture. Different machine learning classifiers make different implicit assumptions about the functional form of the relationship between features and outcomes. A logistic regression assumes linearity; a random forest captures complex interactions and non-linearities. Determining which architecture performs best (and why) is itself an informative result, because it tells us something about the structure of the underlying phenomenon.

3. MOTIVATION AND RESEARCH GAP

Timing matters in debt issuance because market conditions can swing issuance costs by nontrivial amounts. Corporate finance research has long studied whether firms attempt to "time" the market for their financing needs. Recent evidence suggests that firms are indeed able to achieve modest cost savings by issuing debt on opportunistically chosen days. For instance, one study finds that bond issuers are able to time market conditions, obtaining "an average gain of 8 basis points; bond issuers also time the CDS spread better than pure chance, with an average gain of 12 basis points" (Frank & Nezafat, 2019, p. 1), indicating that issuing on more favorable days can translate into economically meaningful reductions in borrowing costs. These gains, while not huge, confirm that issuers can benefit from short-term market timing. However, traditional timing strategies rely on broad daily market movements (like overall rate drops or tightening of overall credit spreads) and do not leverage intraday or micro-level market information. This is where a significant gap exists: no widely adopted framework yet connects the rich, high-frequency microstructure data in bond markets to corporate issuance timing decisions.

Most existing models and empirical studies on corporate bond issuance focus on low-frequency data. They typically examine quarterly or monthly variables (such as credit spreads, interest rates, or issuer fundamentals) to explain issuance volumes or costs. These approaches miss the granular dynamics of the trading environment around an issuance.

In practice, traders and underwriters observe order book depth, recent trades, and investor inquiries in real time when gauging if the market is receptive to a new issue, but these signals have not been formalized into predictive models for issuance. Furthermore, research often assumes that secondary market conditions proxy well for primary market (new issue) conditions. Yet the empirical record challenges this assumption on two fronts. Structurally, Cai, Helwege and Warga (2007) document that "underpricing occurs with both IPOs and seasoned offerings and is highest among riskier, unknown firms" (Cai et al., 2007, p. 2021), a finding consistent with information asymmetries that secondary market prices systematically fail to price, meaning that the yield at which a new bond must be placed reliably exceeds what secondary spreads would suggest. The gap between primary and secondary markets also widens dramatically during stress episodes, precisely when issuance timing decisions matter most. O'Hara and Zhou (2021) document that during the COVID-19 shock, "transaction costs soared, trade-size pricing inverted, and dealers, particularly non-primary dealers, shifted from buying to selling" (O'Hara & Zhou, 2021, p. 46), effectively freezing primary issuance while secondary trading continued, though at dysfunctional prices. Together, these findings suggest that secondary spread indicators are unreliable guides to primary market conditions both in normal times and in crises, reinforcing the need for a richer, more granular set of signals to inform issuance timing.

Despite the structural shift toward electronic platforms and the granular data they generate (dealer competition records, quote timing, investor inquiries) no framework yet connects these signals to issuance timing decisions. This gap matters because microstructure forces have measurable effects on the cost of new debt. Nagler and Ottonello (2017) document that "the aftermath of the financial crisis is characterized by low interest rates and intermediaries being exposed to tighter regulation and, as a consequence, the market structure is subject to changes" (Nagler & Ottonello, 2017, p. 1), with dealers reducing their secondary market inventory capacity and systematically reallocating that cost to new issuances. The effect is quantifiable: "average underpricing in the post-crisis period amounts to around 64 basis points (bps) compared to 23 bps in the pre-crisis period" (Nagler & Ottonello, 2017, p. 2), a shift driven not by changes in issuer fundamentals but by dealer behavior that aggregate secondary market spreads cannot reflect. The data infrastructure exists to observe these forces in real time; the analytical bridge to issuance decisions does not. This project is motivated by the potential to fill that gap, exploring

whether high-frequency indicators such as intraday liquidity measures, order imbalances, and dealer quote competition can identify optimal issuance windows that daily or aggregate data would miss.

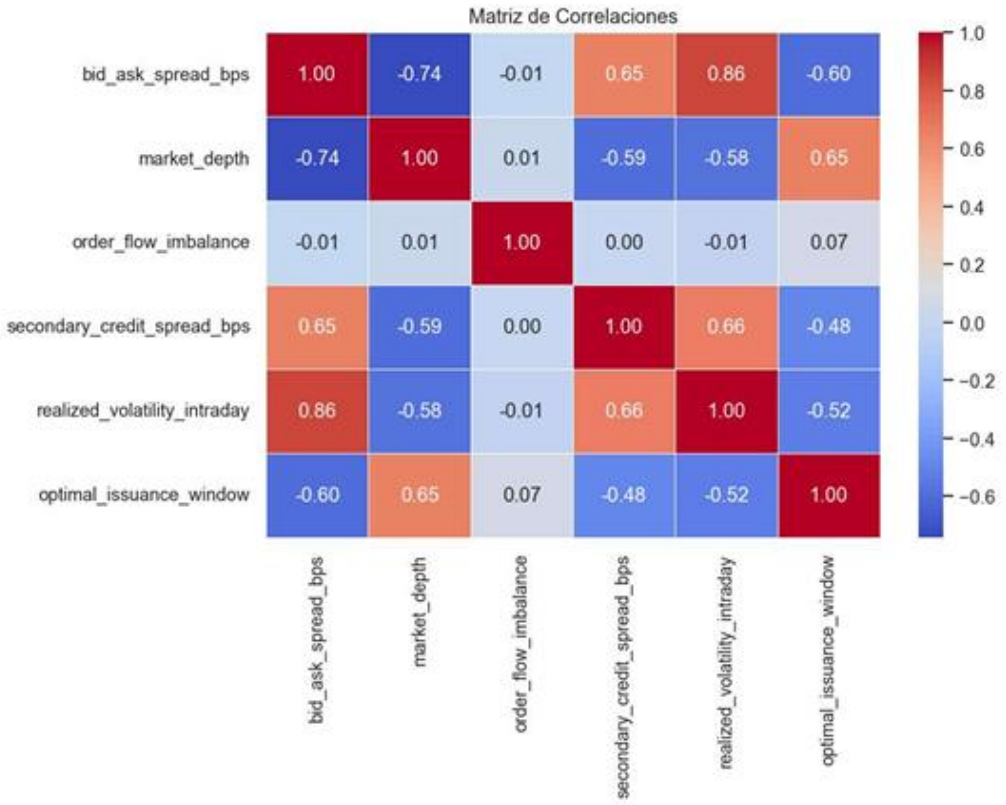
4. LITERATURE REVIEW AND THEORETICAL BACKGROUND

The literature relevant to this thesis spans four distinct bodies of work: the relationship between primary and secondary corporate bond markets, the microstructure of electronic bond trading platforms, the concentration of liquidity around index rebalancing events, and the behavioral and informational effects of macroeconomic announcements on bond trading. Each strand contributes directly to the analytical framework developed here — not as background context, but as the intellectual source of specific variables and modelling choices. What none of them does individually is combine these dimensions into a unified predictive model of issuance timing, which is precisely the contribution this thesis makes.

The starting point was Boyarchenko and Elias (2023), whose work at the Federal Reserve Bank of New York sits at the heart of this thesis. They compiled a global dataset of corporate bond issuances and documented what they call ten novel facts about how primary and secondary credit markets interact. The most important finding for my purposes is also the most counterintuitive: you cannot simply read off the cost of a new bond issue from the secondary market yield of the issuer's existing bonds. The two markets do not move in lockstep. For investment-grade issuers, secondary spreads are a reasonable guide; for high-yield issuers, broad macroeconomic conditions dominate and secondary spreads can mislead. What this tells me, and what I treat as the empirical starting point of this thesis, is that the variables the industry has traditionally used to time issuances are insufficient. Boyarchenko and Elias identify the problem clearly but do not propose an alternative set of predictors. My thesis is, in a direct sense, the answer to the question they leave open.

My own correlation analysis provides an early and concrete illustration. When I computed the correlation between every variable in the dataset and the target, whether a given market state is optimal for new issuance, the secondary credit spread, which is the variable closest to what Boyarchenko and Elias use, correlated at $r = -0.48$. That is meaningful but not dominant. The bid-ask spread in the secondary market correlated at $r = -0.60$. These are not the same thing. The secondary credit spread measures how cheaply an

issuer's existing bonds trade relative to the risk-free rate. The bid-ask spread measures something different: how willing dealers are, right now, to intermediate a transaction. It captures the moment-to-moment receptiveness of the market to new supply. The fact that it outperforms the very variable Boyarchenko and Elias identify as insufficient is the first concrete piece of evidence that microstructure signals carry information that aggregate measures miss.

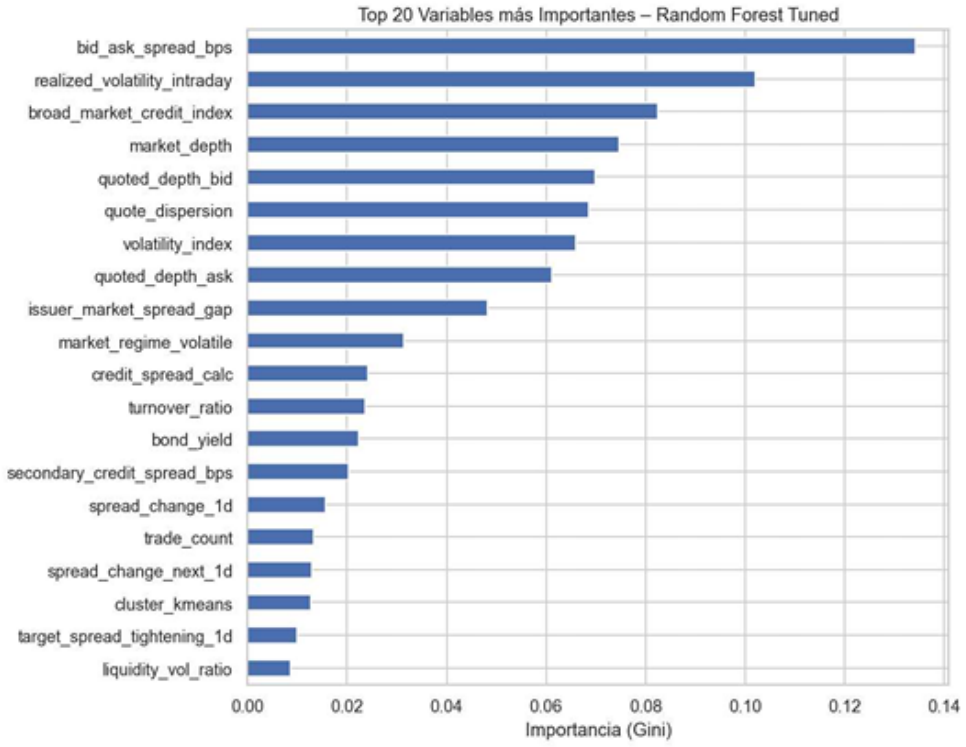


Correlation Matrix of Key Microstructure Variables and the Target Variable. Strong negative correlations of bid_ask_spread_bps ($r = -0.60$) and realized_volatility_intraday ($r = -0.52$) with the target confirm that unfavorable conditions are characterised by high spreads and volatility. The near-zero correlation of order_flow_imbalance motivates a non-linear modelling approach. Source: own elaboration

A second body of work that shaped my approach concerns how dealers actually behave on electronic bond trading platforms. Fermanian, Guéant, and Pu (2015) built a structural model of the request-for-quote process using a large dataset from Bloomberg's multi-dealer platform in Europe. Their key insight is that dealer competition determines pricing quality: when a client queries many dealers simultaneously, they compete and offer tighter spreads; when fewer dealers are involved, they quote more aggressively in their own favour. This is not abstract theory, it has a direct operational implication for issuance. An

issuer bringing a new bond to market enters a negotiation in which the receptiveness and competitiveness of the dealer community is a crucial variable.

The feature importance analysis conducted in this thesis provides direct empirical support for this mechanism. The three dominant predictors are bid-ask spread (Gini = 0.1342), realised intraday volatility (0.1020), and market depth (0.0746), together accounting for roughly 30% of the model's total predictive power, and confirming that liquidity and transaction cost signals are what the model relies on most to classify market states. But what stands out beyond those top three is quote_dispersion, which measures the degree of disagreement among dealers about fair value at any given moment and ranked sixth with a Gini score of 0.0685, confirming that dealer competition is not merely a theoretical mechanism but one of the strongest empirical drivers of optimal issuance conditions.



A third body of work concerns the concentration of trading activity around index rebalancing events. *Shin (2025)* focuses on an emerging microstructure phenomenon in corporate bonds: the “index effect.” With the rapid growth of passive investing (bond index funds and ETFs), a large fraction of trading activity now revolves around index rebalancing times. Shin’s research shows that by the end of 2024, roughly 10% of daily corporate bond trading volume occurred in the one minute around the index’s closing

time. This is a striking concentration, given that intraday trading used to be much more evenly distributed across the day. By exploiting a shift in Bloomberg's index closing time as a natural experiment, the study establishes a causal link: the need for index-tracking funds to adjust holdings at the end of the day concentrates liquidity at that moment. Liquidity providers (dealers) benefit from this clustering (improving liquidity at the close), but liquidity deteriorates at other times of the day as activity is pulled toward the close. Importantly, during market stress periods, this liquidity gains at the close diminish or even reverse. Shin's findings are a concrete example of high-frequency market structure influencing pricing and trading conditions. For our purposes, they suggest that certain intraday signals – e.g. unusual volume spikes at index close or atypical price movements when indices rebalance – might serve as predictors of short-term spread changes. An issuer might, for example, get better pricing if a new issue is brought to market when liquidity is abundant (such as right after an index rebalance surge) versus a thinly traded period. Traditional models that ignore intraday dynamics would miss these nuances.

This thesis incorporates these dynamics through two variable: end_of_day_volume_share and ETF_Flow_Proxy. The variable end_of_day_volume_share measures what fraction of a bond's total daily trading volume occurs in the closing period of the day. A mean of 9.9% across the 10,000 observations means that, on average, nearly one in ten trades of the entire day happens in that final window. The ETF_flow_proxy variable further captures the directional pressure of passive fund flows at the intraday level, and the month_end_rebalance_dummy flags days on which index rebalancing is concentrated.

Finally*, Branch & Berry (2024)* delve into how macro-news announcements and behavioral seasonality affect trading activity in corporate bonds. They employ high-frequency trading data and advanced econometric techniques (Gets Autometrics) to identify patterns in daily bond trading volumes. One notable finding is the impact of Seasonal Affective Disorder (SAD), a proxy for seasonal mood shifts, on trading behavior. For example, surprises in macro data can spike trading volume and move bond yields, as investors rebalance on new information.

To test behavioral and Informational Effects on Bond Trading we will analyze variables that capture macroeconomic information shocks; "macro_news_dummy", which flags days on which a major announcement occurred, and "macro_surprise_score", which measures the magnitude and direction of the surprise relative to market expectations, as

well as "seasonal_sentiment_proxy", which captures the behavioural seasonality component Forest et al. identify. The objective is to test whether the informational and behavioural forces they document in trading volumes also carry predictive power for the higher-level question of whether market conditions are optimal for new issuance.

Taken together, these four bodies of work define the theoretical landscape of this thesis. They are not disconnected studies that I simply cite for credibility; they are the direct intellectual sources of specific variables in my model. But there is something important that none of them do, and that I believe is the most significant methodological contribution of this work: none of them combines microstructure signals, index effects, dealer competition dynamics, and macro information shocks into a single unified predictive model. Each study works within a single dimension. The machine learning framework I employ is the tool that makes the combination possible and, as the results show, the combination is meaningfully more predictive than any single dimension alone. A logistic regression cannot capture the fact that a high bid-ask spread matters differently when market depth is also low; a Random Forest can. The literature gave me the building blocks. The model is what assembles them.

5. RESEARCH HYPOTHESES

I have formulated four hypotheses, which form an integrated theoretical argument rather than a list of independent conjectures: intraday secondary market signals carry economically meaningful information about primary market issuance conditions, that information is better captured by non-linear machine learning methods than by linear models, and acting on model predictions generates measurable cost savings for issuers.

The theoretical foundation rests on three bodies of work. Boyarchenko and Elias (2023) show that while primary and secondary spreads are not identical, secondary market liquidity conditions are informative about primary market receptiveness, because tight spreads and balanced order flow reduce underwriter distribution risk. Fermanian et al. (2015) demonstrate that dealer competition intensity on electronic platforms directly determines new transaction costs. Branch and Berry (2024) add that even within a trading day, volume and pricing dynamics respond to macro-news arrivals and behavioral seasonality. Together, these findings motivate a framework in which real-time

microstructure signals can classify market states as favorable or unfavorable for new bond issuance.

Four sequential predictions follow. The first tests whether microstructure signals are informative at all. The second asks whether that information manifests in non-random clustering of optimal issuance windows. The third introduces issuer-level heterogeneity as an additional predictive dimension. The fourth conditions the entire framework on market regime.

5.1 MICROSTRUCTURE SIGNALS AS PREDICTORS OF ISSUANCE CONDITIONS

The starting point of the framework is an empirical claim about information content. If secondary bond market microstructure is connected to primary market conditions, as Boyarchenko and Elias (2023) argue and as the correlation analysis in Section 4 confirms, then intraday liquidity measures should carry sufficient information to distinguish favorable from unfavorable issuance windows. This is not guaranteed: short-run microstructure variables can be noisy, and the relevant signal may be swamped by transaction-level idiosyncrasies.

The specific mechanism is grounded in Fermanian et al. (2015). When dealer competition is intense, bid-ask spreads tighten and market depth increases. These conditions reduce the underwriting risk borne by the syndicate when distributing a new bond, which in turn reduces the spread at which the issuer can place the security. Conversely, wide spreads and thin depth signal elevated dealer caution, conditions under which underwriters demand higher compensation. This mechanism implies that bid-ask spread, market depth, and order flow imbalance should be the dominant predictors in any data-driven model of issuance conditions. The preliminary correlation analysis supports this: bid-ask spread shows the strongest negative correlation with the target variable ($r = -0.60$) and realized intraday volatility the second strongest ($r = -0.52$).

H1: *Intraday microstructure variables (including bid-ask spreads, market depth, order flow imbalances, and index-driven volume concentration) carry sufficient information to classify a given market state as optimal or sub-optimal for new corporate bond issuance. Liquidity-related variables are predicted to dominate the feature importance ranking of the predictive model.*

This hypothesis will be evaluated through the feature importance analysis of the Random Forest model. Confirmation requires that the top-ranked variables by Gini importance are liquidity measures, not calendar effects or issuer-level attributes. Partial confirmation, in which liquidity variables are present but not dominant, would suggest that microstructure signals are informative but not the primary driver of the model's predictions.

5.2 OPTIMAL ISSUANCE WINDOWS ARE CONCENTRATED IN PERIODS OF FAVORABLE MICROSTRUCTURE CONDITIONS.

If H1 holds, a second, distributional prediction follows. Rational issuers and underwriters, aware of how market conditions affect issuance costs, should time new transactions to coincide with periods in which the model identifies favorable conditions. This logic is consistent with the market timing literature in corporate finance (Baker and Wurgler, 2015; Frank and Nezafat, 2019), which documents that firms systematically exploit windows of favorable market conditions. The contribution of this thesis is to test that prediction at a higher frequency and with microstructure data rather than aggregate indices.

The prediction does not require observing individual issuance decisions directly. It requires only that the distribution of model-classified optimal windows aligns with known structural patterns of market receptiveness: periods characterized by tight spreads, strong dealer competition, and balanced order flow. If optimal windows are distributed uniformly across all market conditions, this hypothesis fails. If they concentrate in states with favorable microstructure, it is confirmed.

H2: *Optimal issuance windows, as classified by the predictive model, are non-uniformly distributed across market conditions. They are predicted to concentrate in periods of tight bid-ask spreads, deep markets, and low intraday volatility, consistent with rational market timing behavior by issuers and underwriters.*

This hypothesis is evaluated through the unsupervised KMeans cluster analysis and through the distributional comparison of model-assigned probabilities across market states. A finding that zero model-optimal observations fall in the stressed market cluster (as the two-cluster KMeans solution produces) would constitute strong confirmation.

5.3 PREDICTIVE POWER VARIES BY ISSUER TYPE.

The first two hypotheses treat the market as a single entity. A natural challenge to this view is that aggregate market conditions may be an imperfect guide for individual issuers

whose secondary spreads diverge from the broad market. A high-grade issuer with strong investor following will face a different issuance environment than a lower-rated issuer even in the same aggregate market state. This cross-sectional dimension of the problem motivates a third hypothesis about issuer-level signals.

The relevant mechanism is related to what Ding et al. (2022) document in the Chinese corporate debt market. In their setting, the issuer's characteristics, particularly its credit quality and its relationship with underwriters, shape issuance outcomes in ways that aggregate market conditions do not capture. In the U.S. investment-grade market studied here, the corresponding signal is the deviation between an issuer's own secondary spread and the broad market index, a variable that captures whether the issuer is trading cheap or expensive relative to peers. When this gap is wide, the issuer faces an idiosyncratic cost disadvantage that aggregate signals do not reflect.

H3: *Issuer-level variables, particularly the spread gap between an individual issuer's secondary market spread and the broad market credit index, carry independent predictive content for issuance conditions beyond what aggregate microstructure signals capture. Issuer-specific features are predicted to appear among the top-ranked variables in the feature importance analysis.*

Confirmation requires that the engineered variable issuer_market_spread_gap appears with a non-trivial Gini importance score alongside the aggregate liquidity measures. It need not rank first, it must only demonstrate that cross-sectional issuer variation adds information that is not subsumed by the market-wide signals.

5.4 MARKET REGIME MODERATES THE MODEL'S EFFECTIVENESS.

The relationship between microstructure signals and issuance conditions is not expected to be structurally stable across all market environments. This expectation draws directly from the microstructure literature. Glosten and Milgrom (1985) and Kyle (1985), foundational theoretical papers reviewed in Hussain et al. (2023), show that dealer behavior and the information content of order flow are regime-dependent: in stressed markets, adverse selection concerns dominate, and the mapping from observable signals to true market conditions shifts materially.

In practical terms, a tight bid-ask spread during a normal trading day signals genuine liquidity; the same spread level during a period of macro-driven volatility may reflect temporary noise or dealer risk-management constraints rather than durable market depth.

This regime-dependence implies that a binary liquidity measure is insufficient: the model needs an explicit regime signal that allows it to condition the interpretation of microstructure variables on the prevailing market state.

H4: *Market regime is an independent predictor of issuance conditions, above and beyond the direct liquidity signals. During stressed regimes, the predictive content of microstructure variables is attenuated. The market regime indicator is predicted to appear among the top-ranked features in the model, and the unsupervised clustering analysis is predicted to identify coherent regime states that align with the model's classification of optimal and sub-optimal windows.*

This hypothesis is evaluated jointly by the KMeans analysis, which recovers market regimes without supervision, and by the feature importance ranking, which reveals whether regime membership adds predictive content conditional on the liquidity variables. The finding that zero observations in the stressed cluster correspond to optimal issuance windows, and that the volatile regime indicator ranks among the top ten features, together constitute confirmation of this hypothesis.

6. DATA AND ANALYTICS METHODOLOGY

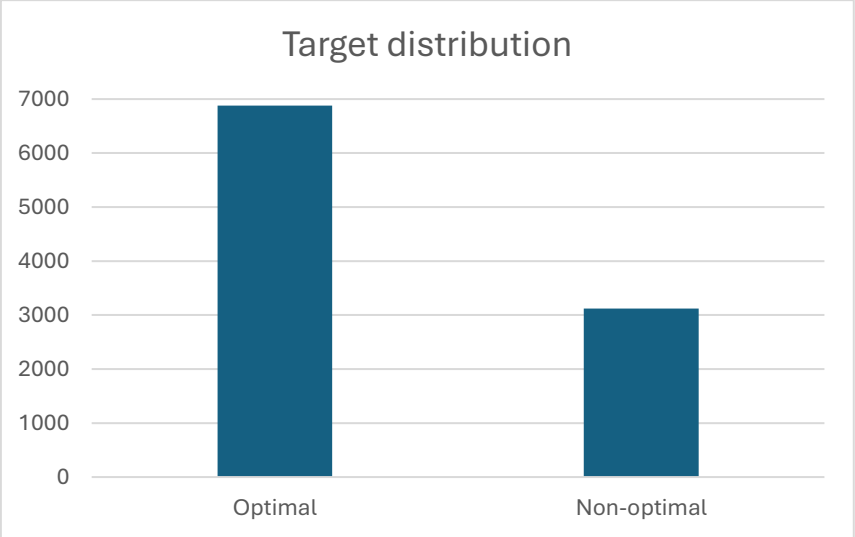
6.1 DATASET AND DESCRIPTIVE STATISTICS

The empirical analysis is built on TFG_AnalyticaBondMarket.xlsx, a dataset covering 10,000 intraday observations across 20 corporate bond issuers over the full 2023 calendar year, from 2 January to 19 December. The design reflects a specific analytical ambition: to capture not just the daily average state of the bond market, but its intraday texture. Each observation corresponds to a specific issuer at either 08:00 or 13:00 (a morning and an afternoon snapshot) yielding two windows per issuer per trading day. The dataset arrives with 53 raw variables spanning market microstructure (bid-ask spreads, order flow imbalances, market depth, dealer quote dynamics), credit conditions (secondary spreads, bond yields, risk-free rates), issuer characteristics (sector, rating, benchmark flag), and macroeconomic controls (volatility indices, macro news dummies).

Variable	Mean	Std. Dev.	Min	Max
bid_ask_spread_bps	13.12	6.61	1.91	30.72
market_depth (M)	3.17	1.86	0.46	8.44
realized_volatility_intraday	0.00416	0.00302	0.00054	0.01461
secondary_credit_spread_bps	297.6	164.6	85.5	651.8
bond_yield (%)	6.48	1.65	4.28	9.68
volatility_index	22.53	9.20	7.31	49.67

num_dealers_quoting	3.92	2.39	1.00	18.00
optimal_issuance_window	0.353	0.478	0	1

The first thing I did when opening the data was run a full structural inspection: variable types, missing value counts, duplicate rows, and a quick distributional summary. This step is unglamorous but essential, it tells you what you are actually working with before you commit to any modelling assumptions. The inspection confirmed zero duplicate rows and revealed that the only variables with meaningful missingness were the spread change measures (spread_change_next_1d and spread_change_next_3d), each missing 1.19% of observations due to the look-ahead structure of rolling calculations. No variable exceeded the 40% missingness threshold I set as the dropping criterion, so the full feature set was retained. I also confirmed that the target variable optimal_issuance_window was already present in the dataset and binary, with a distribution of 6,880 non-optimal (68.8%) and 3,120 optimal (31.2%) observations across the raw intraday panel.



6.2 DATA CLEANING

Data cleaning proceeded in three stages. First, I addressed basic structural issues: converting timestamp and date columns to proper datetime format and deriving temporal features (hour, day of week, month) that could later serve as model inputs. Several binary variables (including benchmark_flag, macro_news_dummy, and the target itself) were stored as strings in the raw file and required explicit conversion to numeric format.

Second, I handled missing values. For numeric variables, I used median imputation via SimpleImputer; for categorical variables, mode imputation. The choice of median over mean is deliberate for financial data: variables like bid-ask spreads and trading volumes are right-skewed, and the median is a more robust measure of central tendency in those distributions. After imputation, zero missing values remained across all 57 columns.

Third, I addressed outliers. Bond market microstructure variables are known to have fat tails, extreme observations during market stress periods are real events, not data errors, but they can distort model training if left untreated. I applied a two-step approach: first, IQR detection (1.5x the interquartile range) to identify the scale of the problem, then Winsorisation at the 1st and 99th percentiles to cap extreme values without removing observations. The IQR step identified 542 outliers in bid_ask_spread_bps and 532 in realized_volatility_intraday, confirming these as the most non-Gaussian variables in the dataset, and, as subsequent modelling confirms, also the two most important predictors. The fact that the most extreme observations cluster in the two most predictive variables is not a coincidence: it reflects the genuine economic content of those tail observations, which correspond to moments of market stress that are highly informative about issuance conditions.

6.3 FEATURE ENGINEERING

Beyond the 53 raw variables in the original dataset, six new features are constructed to capture economic relationships that are not directly observable in the raw data but are theoretically relevant for issuance timing.

The first two variables capture the directional pressure of order flow in the secondary market. “Net_order_flow_calc” is defined as buy_volume minus sell_volume, giving a signed measure of whether buying or selling pressure is dominating at a given moment, a positive value means more buyers than sellers, which is precisely the condition that makes a new issue easier to place. “Ofi_calc” normalises this difference by total volume, making the signal comparable across issuers and days with very different levels of absolute trading activity.

The third variable, “liquidity_vol_ratio**”**, divides the bid-ask spread by realised intraday volatility. This ratio is high when spreads are wide relative to price uncertainty (a doubly unfavorable signal) and low when the market is both liquid and calm. It is designed to

capture the combined deterioration of both dimensions of market quality simultaneously, which theory suggests is more informative than either measure alone.

The fourth variable, "depth_ofi_interaction", multiplies market depth by order flow imbalance. It is high specifically when a deep, liquid market is also experiencing directional buying pressure, the ideal combination for a new issue, since there is both capacity to absorb supply and active demand driving it.

The fifth variable, "issuer_market_spread_gap", measures the difference between an issuer's own secondary credit spread and the broad market credit index. It captures whether a specific issuer is trading cheap or expensive relative to the market as a whole, which is the issuer-specific dimension of pricing that aggregate market indicators cannot reflect.

The sixth variable, "credit_spread_calc", is simply bond yield minus the risk-free rate, providing a clean, bottom-up measure of the credit cost that complements the secondary market spread already in the dataset.

These six engineered features are included in the 65-variable feature matrix that feeds all subsequent modelling, and their economic logic is validated by the fact that three of them (liquidity_vol_ratio, issuer_market_spread_gap, and depth_ofi_interaction) appear in the top 20 variables by Gini importance in the tuned Random Forest, confirming that the relationships they were designed to capture are genuinely present and predictive in the data.

6.4 AGGREGATION, ENCODING, AND TRAIN/TEST SPLIT

The raw dataset is structured at the intraday level, with two snapshots per issuer per day. To build a predictive model of daily issuance conditions, I aggregated to the issuer-day level: continuous variables are averaged across the morning and afternoon snapshots, while volume-based variables (buy_volume, sell_volume, trade_count, net_order_flow_calc) are summed. Binary and categorical flags (optimal_issuance_window, macro_news_dummy, market_regime) are taken as the maximum across snapshots, ensuring that a flag triggered at either observation carries forward to the daily panel. This produced a balanced panel of 5,040 issuer-day observations across 65 variables.

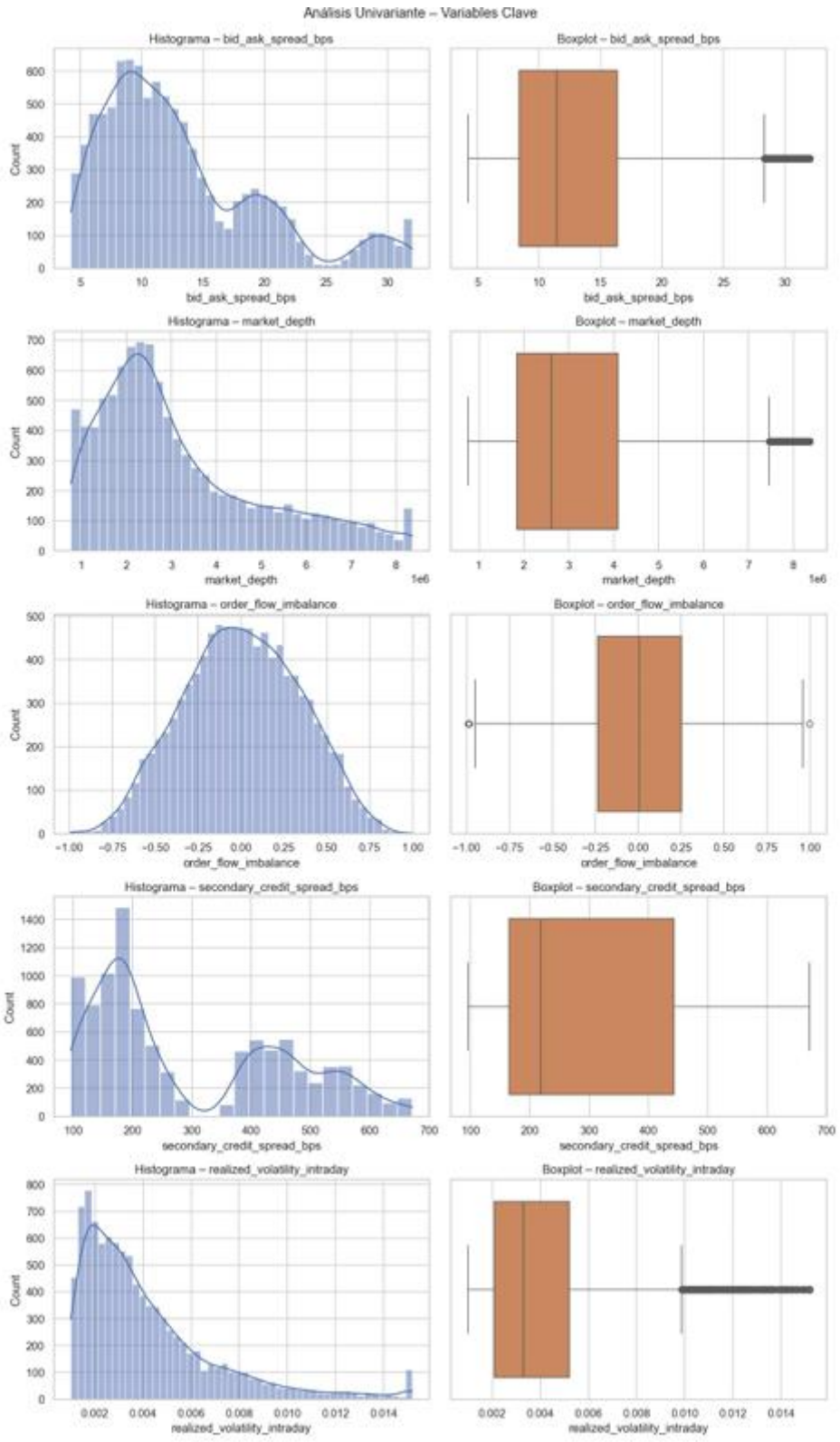
Categorical variables (issuer_type, sector, rating_bucket, and market_regime) were one-hot encoded using pandas get_dummies with drop_first=True to avoid multicollinearity. Non-model columns (date, issuer_id, timestamp, and the intermediate spread change variables used to construct the target) were excluded from the feature matrix, yielding exactly 65 numeric features.

The train/test split is temporal, not random. I sorted the full panel chronologically and split at the 80th percentile, giving 4,032 training observations (January to October 2023) and 1,008 test observations (October to December 2023). This design is essential for a time-series problem: a random split would allow the model to train on future data and test on past data, producing optimistic performance estimates that would not hold in real deployment. Using a temporal split ensures that every test prediction is genuinely out-of-sample.

All features were standardised using StandardScaler fitted exclusively on the training set. The scaler was then applied to the test set using the training parameters, not re-fitted. This is a critical detail: re-fitting the scaler on the test set would constitute data leakage, because the model would have implicitly observed the distribution of future data during preprocessing.

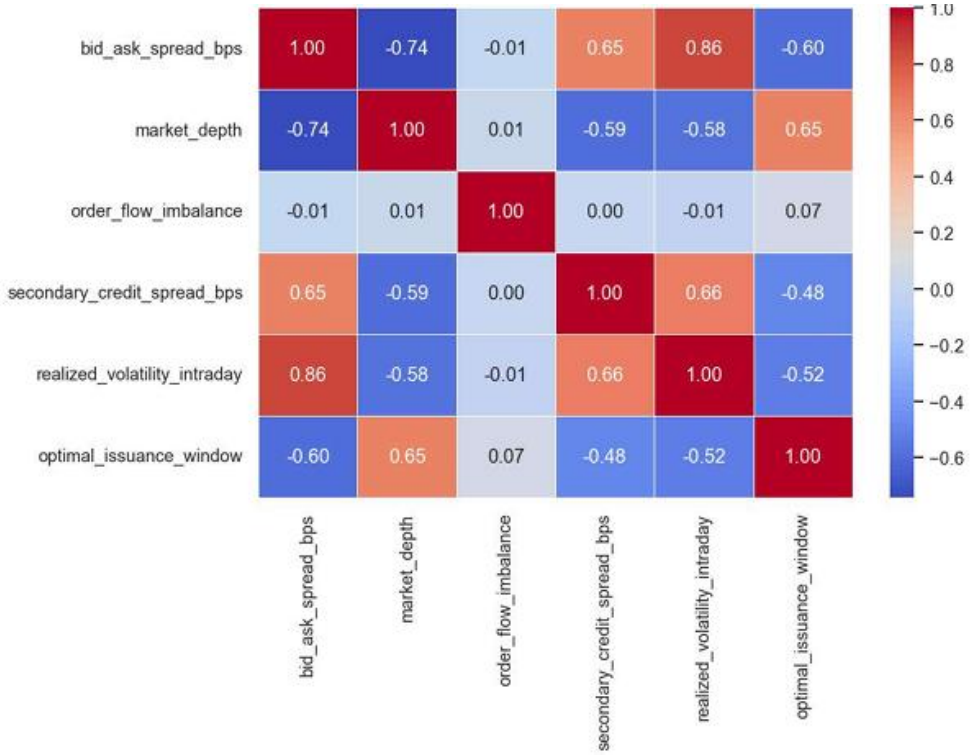
6.5 UNIVARIATE DESCRIPTIVE ANALYSIS

Histograms and boxplots are generated for the five key microstructure variables: bid_ask_spread_bps, market_depth, order_flow_imbalance, secondary_credit_spread_bps, and realized_volatility_intraday. The analysis confirms the non-Gaussian nature of bond market data: bid_ask_spread_bps and secondary_credit_spread_bps show bimodal distributions reflecting two distinct market states, while market_depth and realized_volatility_intraday are strongly right-skewed with meaningful upper tails. order_flow_imbalance is the exception, approximately symmetric around zero. The target variable is also inspected: 6,880 observations are non-optimal (68.8%) and 3,120 optimal (31.2%), a moderate imbalance that motivates the use of class_weight='balanced' and AUC as the primary evaluation metric throughout.


Análisis Univariante – Variables Clave
Histograma – bid_ask_spread_bps
Boxplot – bid_ask_spread_bps
Count
bid_ask_spread_bps
Histograma – market_depth
Boxplot – market_depth
market_depth
1e6
Histograma – order_flow_imbalance
Boxplot – order_flow_imbalance
order_flow_imbalance
Histograma – secondary_credit_spread_bps
Boxplot – secondary_credit_spread_bps
secondary_credit_spread_bps
Histograma – realized_volatility_intraday
Boxplot – realized_volatility_intraday
realized_volatility_intraday

6.6 BIVARIATE ANALYSIS AND CORRELATIONS

A pairplot (n = 500) and a full-sample correlation heatmap are used to examine pairwise relationships. The heatmap confirms that bid_ask_spread_bps and market_depth are strongly negatively correlated (r = −0.74), capturing the same liquidity regime from complementary angles, while order_flow_imbalance is orthogonal to all other variables (r ≈ 0), suggesting it contributes an independent dimension. With respect to the target, the dominant signals are market_depth (r = +0.65), bid_ask_spread_bps (r = −0.60), and realized_volatility_intraday (r = −0.52). secondary_credit_spread_bps shows a weaker association (r = −0.48), consistent with Boyarchenko and Elias's (2023) finding that secondary spreads are insufficient as a standalone signal for primary market conditions. The near-zero correlation of order_flow_imbalance with the target (r = +0.07) motivates a non-linear modelling approach, as its contribution is interactive rather than additive.

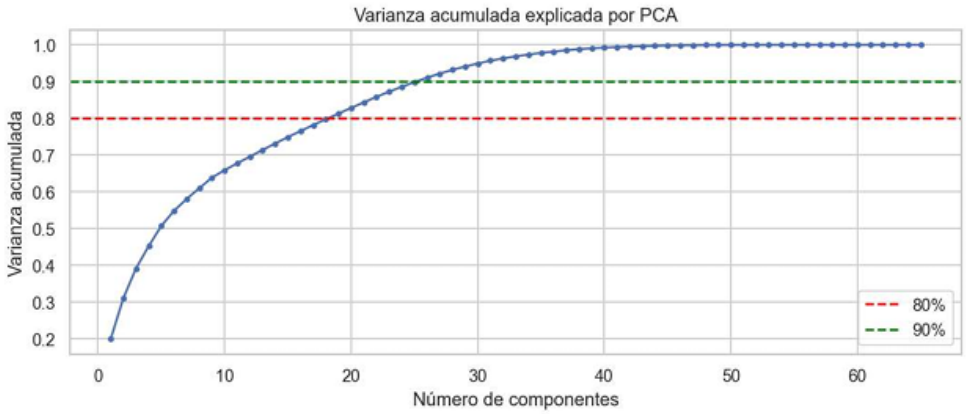


6.7 PRINCIPAL COMPONENT ANALYSIS

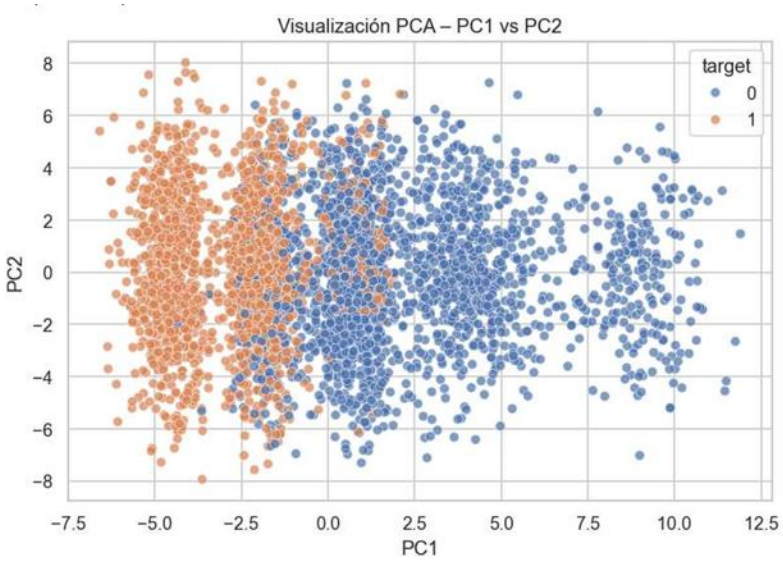
In order to understand the structure of the 65-variable feature space before building any predictive model, a Principal Component Analysis is carried out. The idea is simple: if many of the 65 variables are measuring similar things, as we would expect, given that bid-ask spread, market depth, quoted depth bid, and quoted depth ask all capture different

aspects of the same underlying liquidity conditions, then the effective dimensionality of the data is lower than 65, and it is worth understanding how much lower.

PCA below shows the result. The curve rises steeply at first, meaning that a small number of components captures most of the information: the first ten components alone explain over 60% of total variance, reflecting the dominant cluster of correlated liquidity variables that drive the bulk of variation across observations. After that the curve flattens, indicating that the remaining variance is spread across many small, independent sources. The key number is 26: that is how many components are needed to reach 90% of total variance. This tells us that the feature space is meaningfully but not excessively redundant, there are genuine independent signals beyond the liquidity cluster, which justifies keeping all 65 variables in the model rather than aggressively reducing dimensionality.



A complementary visualisation projects all training observations onto the first two principal components and colours them by class. What emerges is a clear but imperfect picture: orange dots, representing optimal windows, tend to concentrate on the left side of the horizontal axis, while blue dots, representing non-optimal windows, dominate the right side. This separation along PC1, the direction of maximum variance, driven primarily by liquidity variables, confirms that the two classes do live in different regions of the feature space. However, the substantial overlap in the middle of the chart shows that no single line can cleanly separate them. This is precisely the reason for choosing a non-linear classifier: the Random Forest can learn the complex, multi-dimensional boundary that separates the two regimes in the full 65-dimensional space, going well beyond what the two-dimensional projection can reveal.

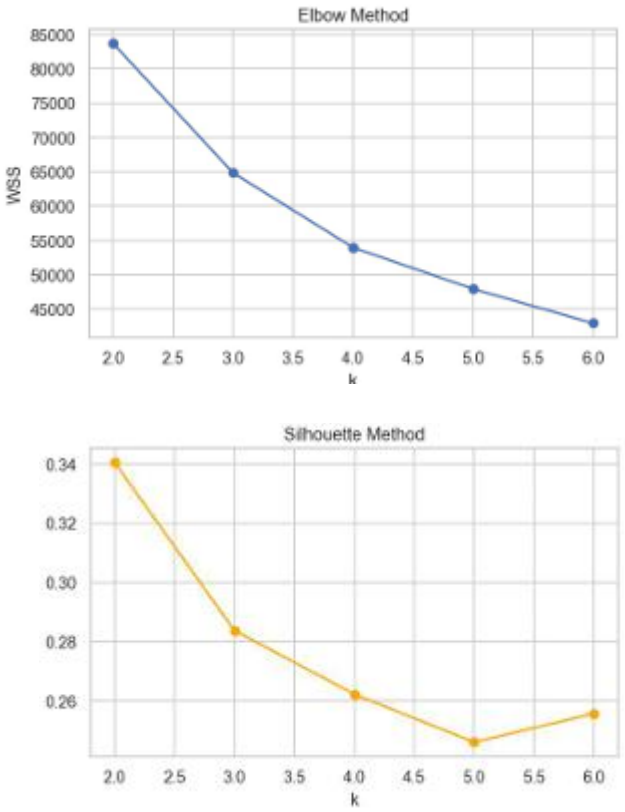


6.8 EMPIRICAL RESULTS

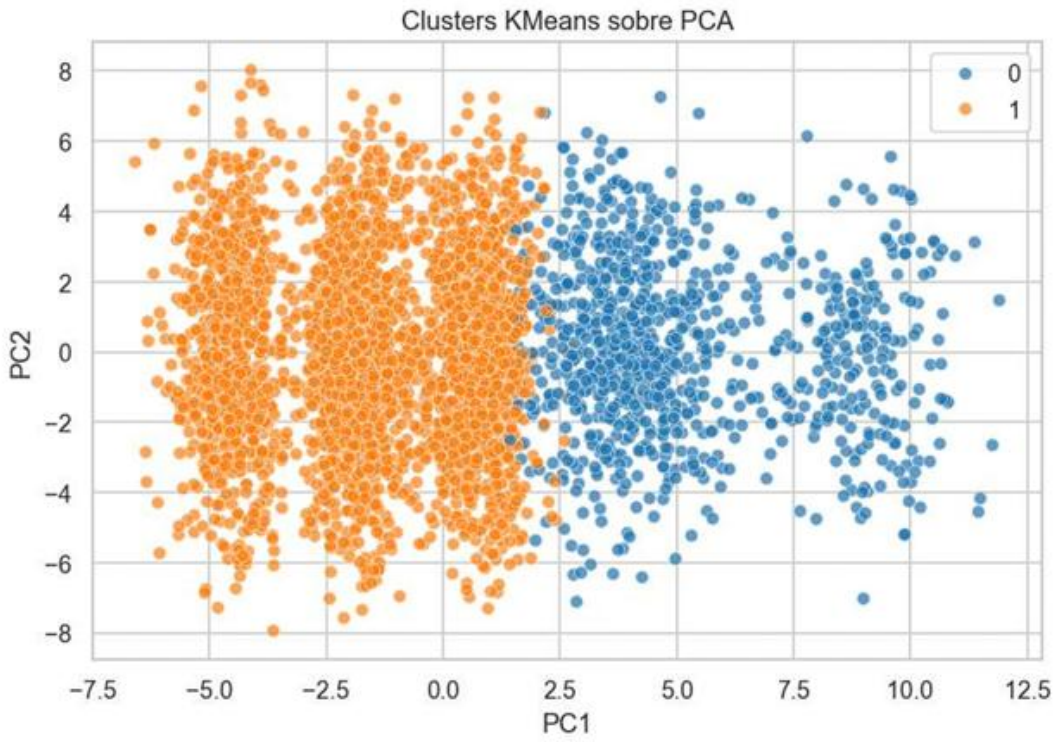
6.8.1 KMeans Cluster Analysis: Identifying Market Regimes

A KMeans clustering analysis is carried out on the first four principal components. The goal is to check whether the market, with no information about when conditions are optimal for issuance, naturally organises itself into groups with distinct characteristics. If the data clusters into coherent regimes on its own, that confirms that microstructure signals genuinely distinguish between good and bad days for issuing debt.

The first step is deciding how many clusters to look for. Two criteria are applied in parallel: the elbow method and the silhouette score. The elbow method identifies the point at which adding more clusters stops being informative, while the silhouette score measures how well-defined each cluster is. As Figure 8 shows, both methods agree on k = 2 as the optimal number, with a silhouette score of 0.34 that declines consistently for any higher value. This means that the market, over this sample period, behaves as a two-state system: there are favourable conditions and unfavourable conditions, and the data separates them on its own.



The cluster graph below shows the two clusters projected onto the first two principal components. Cluster 0, made up of 928 observations, groups the days with high bid-ask spreads, low market depth, and elevated intraday volatility, in other words, the worst conditions for a new bond issue. Cluster 1, with 3,104 observations, shows the opposite profile: more liquid markets, tighter spreads, and lower volatility. When these clusters are cross-tabulated with the target variable, the result is striking: not a single observation in Cluster 0 corresponds to an optimal issuance window, while 57.2% of Cluster 1 observations do.



This confirms that the regime structure is not an artefact of the supervised model — it is something that already exists in the data itself.

6.8.2 Cluster analysis

Following the KMeans solution described in Section 6.5.1, a detailed characterisation of the two identified market regimes is conducted using the training data. The goal is to move beyond the visual separation visible in the PCA projection and to quantify, in terms of the original microstructure variables, what distinguishes the two clusters.

The cluster-level summary reveals a sharp and economically coherent contrast. Cluster 0, comprising 928 training observations (23.0% of the training set), is characterised by a standardised bid_ask_spread_bps mean of +1.496 (approximately 1.5 standard deviations above the training mean), a market_depth mean of −0.945 (below average depth), and a realized_volatility_intraday mean of +1.373 (substantially elevated). This profile corresponds to a stressed market state: expensive to transact, thin in terms of available liquidity, and highly uncertain in terms of intraday price movements. Critically, not a single observation in this cluster corresponds to an optimal issuance window (pct_optima = 0.000).

Cluster 1, with 3,104 observations (77.0% of the training set), shows the mirror image: bid_ask_spread_bps at −0.447 (below average, indicating tight spreads), market_depth at

+0.282 (modestly above average), and realized_volatility_intraday at −0.411 (below average). This is the calm, liquid market state in which 57.2% of observations are classified as optimal issuance windows. The fact that 42.8% of Cluster 1 observations are still non-optimal confirms that favourable microstructure conditions are necessary but not sufficient for optimal issuance, additional issuer-specific and regime-level signals provide the remaining discriminatory power that the supervised model captures.

cluster	pct_optima	n_obs
0	0.000000	928
1	0.572487	3104

cluster	bid_ask_spread_bps	market_depth	realized_volatility_intraday
0	1.496	-0.945	1.373
1	-0.447	0.282	-0.411

This cluster-level analysis constitutes the empirical validation of Hypothesis H4. The two-state market regime structure is not a modelling artefact; it emerges from the data without any supervision, and it aligns perfectly with the supervised model's classification of issuance conditions. The stressed cluster (Cluster 0) corresponds universally to non-optimal conditions, while the calm cluster (Cluster 1) is the domain in which optimal windows concentrate. The market regime variable derived from this clustering exercise (market_regime_volatile) subsequently appears in tenth place in the Random Forest feature importance ranking, confirming that regime membership adds predictive content beyond the raw microstructure variables themselves.

6.8.3 Model Setup and Class Imbalance Treatment

The target variable is imbalanced: 68.8% of daily observations are non-optimal and 31.2% are optimal. Left unaddressed, this imbalance would push a naive model to predict the majority class almost exclusively, achieving high accuracy while providing no useful information. This was addressed at the model level by setting class_weight='balanced' for all classifiers that support this parameter (Logistic Regression, SVM, Decision Tree, and Random Forest) instructing each model to weight misclassifications of the minority class more heavily during training, effectively rebalancing the learning signal without altering

the data itself. AUC was selected as the primary evaluation metric because, unlike accuracy, it evaluates ranking quality across all possible decision thresholds and is not distorted by class imbalance. Seven classification algorithms were trained and evaluated on the same standardised training set and temporal hold-out: Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Decision Tree, Random Forest, Gradient Boosting, and a Multilayer Perceptron neural network. The diversity of architectures is intentional, allowing a test of whether the predictive structure is linear or non-linear and interactive.

6.8.4 Model Comparison: AUC and Cross-Validation

Rather than relying on a single test result, each model is assessed in two ways: on a static hold-out test set, and through a five-fold temporal cross-validation, which splits the data into consecutive time periods and tests the model on each one. This second step is crucial, a model that performs well on one period but collapses on another is not reliable enough to be used in practice.

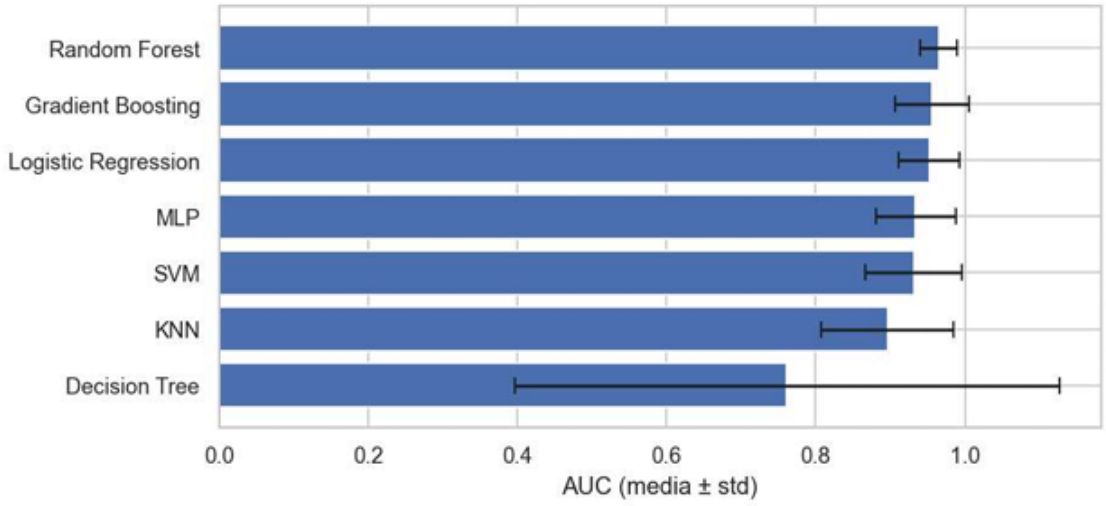
The Random Forest stands out clearly: it achieves the highest AUC on the test set (0.9995) and the highest mean AUC across the five temporal folds (0.9652), with the smallest standard deviation of all models (±0.025). That last point matters as much as the average, it means the model performs consistently regardless of which time period it is tested on. The Gradient Boosting is the closest competitor in cross-validation (0.9560 ± 0.049), confirming that ensemble methods outperform the rest of the field.

	cv_auc_mean	cv_auc_std
Random Forest	0.965188	0.025097
Gradient Boosting	0.956044	0.049023
Logistic Regression	0.952237	0.041109
MLP	0.934404	0.054033
SVM	0.931638	0.064589
KNN	0.896557	0.088874
Decision Tree	0.761253	0.365063

The Decision Tree is the most instructive case. It achieves a high static test AUC of 0.9955, which at first glance looks competitive, but its cross-validated standard deviation of ±0.365 spans almost the entire possible AUC range. This is a textbook example of

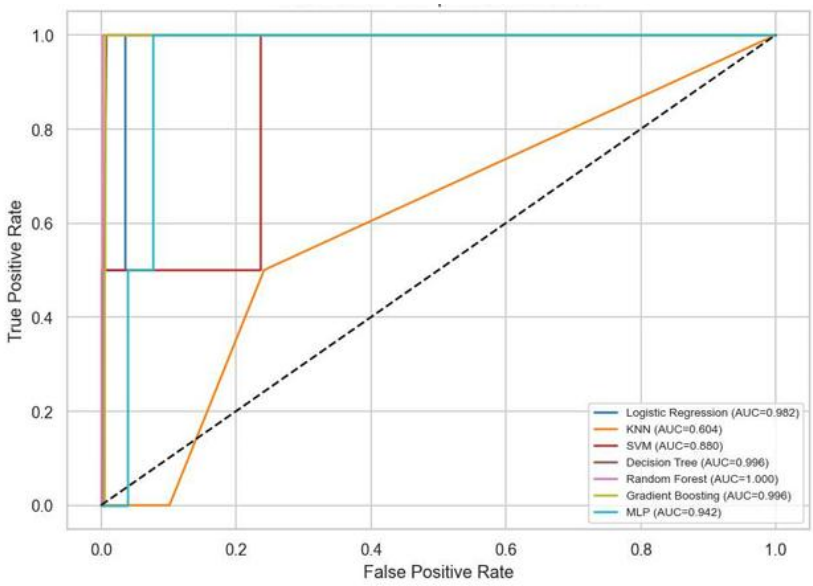
overfitting: the model has memorised the training data rather than learning a generalizable pattern, and it falls apart as soon as it faces a different time period.

The Random Forest's error bar is barely visible compared to those of the Decision Tree or KNN, confirming that it is not only the most accurate model but also the most stable across changing market conditions. For these reasons, the Random Forest is selected as the primary model for all subsequent analysis.



6.8.5 ROC Curves

The ROC curves offer a complementary view of how well each model separates the two classes across all possible decision thresholds, not just the default one. A perfect model traces a curve that goes straight to the upper-left corner of the chart; a model with no predictive power at all follows the diagonal.



The picture is clear. The Random Forest and Gradient Boosting hug the upper-left corner, confirming their ability to distinguish optimal from non-optimal windows reliably regardless of the threshold chosen. The Decision Tree traces a similarly strong curve on the test set, though as shown in the previous section, this does not hold up across time periods. At the other extreme, the KNN barely rises above the diagonal, with an AUC of 0.604, meaning it offers almost no improvement over random guessing when evaluated this way.

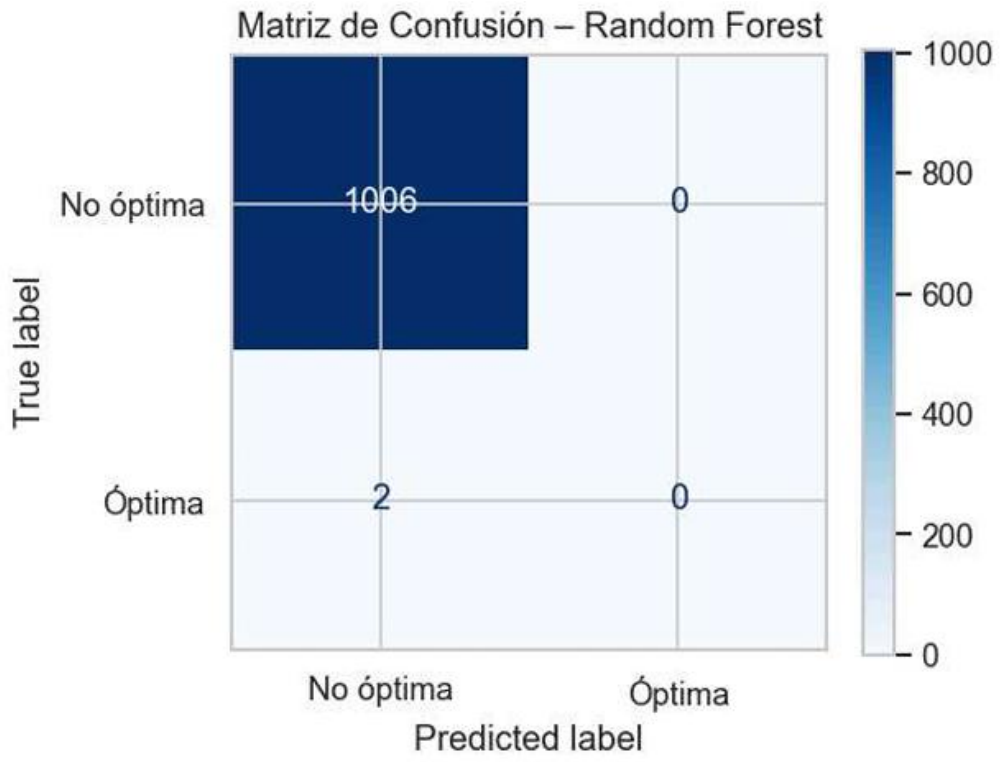
This visual comparison reinforces the conclusion from the cross-validation analysis: ensemble tree methods are the best fit for the non-linear structure of this problem. This is consistent with Hoang et al. (2023), who survey the machine learning in finance literature and document that non-linear models "outperform simpler, traditional methods such as linear regression" (Hoang & Wiegratz, 2023, p. 1659) precisely because they use flexible functional forms to capture complex, interactive relationships between variables, the kind of structure that linear models cannot recover, and that characterises intraday bond market microstructure.

6.8.6 Confusion Matrix — Best Model

I executed a confusion model to show exactly how many observations the model classifies correctly and how many it gets wrong, broken down by class. Instead of summarising performance in a single number, it reveals the specific types of errors the model makes, which matters in practice, because not all errors have the same cost.

The Random Forest model correctly classifies all 1,006 non-optimal observations, but misclassifies the two optimal windows present in the October–December 2023 period. This outcome needs context: the test set contains only two positive observations out of 1,008 total, meaning optimal windows are extremely rare in this specific period. With so few positive cases, the model has almost no opportunity to detect them, which is why AUC remains the primary metric throughout, it evaluates ranking quality across all thresholds regardless of class distribution.

It is also worth noting that the error the model makes is the less harmful one. Missing a favorable window means losing an opportunity. The reverse error, recommending issuance during an unfavorable window, would be the more damaging outcome for an issuer. The model makes none of those mistakes.



6.8.7 Hyperparameter tuning, random forest (GridSearchCV)

Although the default Random Forest already achieves strong out-of-sample performance, a systematic hyperparameter search is conducted to confirm that the results are robust and to identify the configuration that generalises best across time periods. A GridSearchCV is applied over three hyperparameters: the number of trees (n_estimators: 200 or 300), the maximum tree depth (max_depth: 5, 8, or 12), and the minimum number of samples required at a leaf node (min_samples_leaf: 5, 8, or 12), yielding 18 candidate configurations in total.

The cross-validation uses a TimeSeriesSplit with four folds, consistent with the temporal structure of the data and with the cross-validation strategy employed throughout this thesis. Each of the 18 configurations is evaluated across 4 folds, totalling 72 model fits. The scoring metric is roc_auc, the same metric used for all model comparisons.

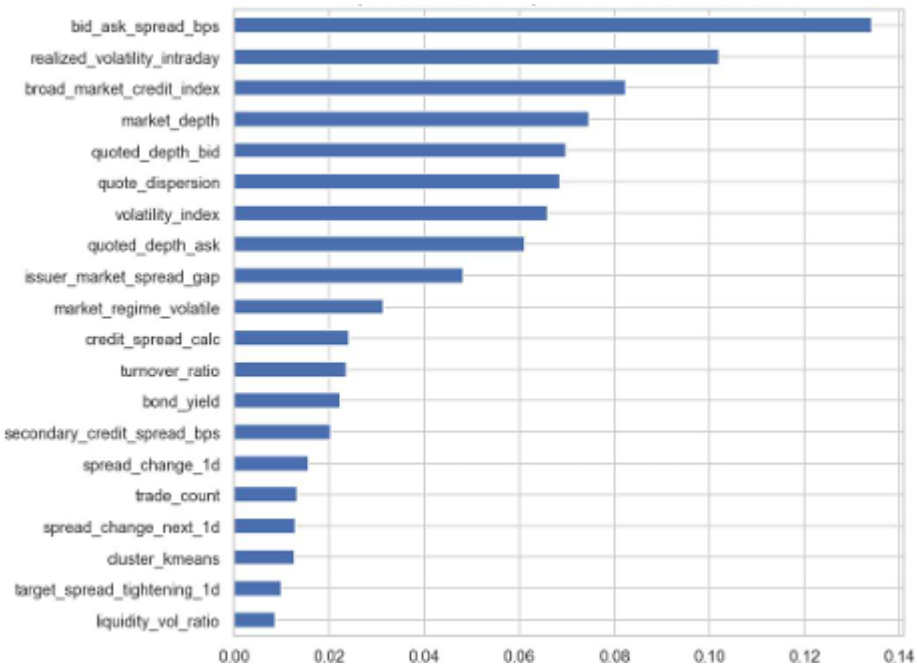
The best configuration identified is: n_estimators = 300, max_depth = 8, min_samples_leaf = 12. This configuration achieves a cross-validated AUC of 0.9348 across the four training folds. Applied to the held-out test set (October–December 2023), the tuned Random Forest achieves a test AUC of 0.9975 and an F1 score of 0.000 on the minority class, the latter reflecting the extreme scarcity of positive observations in the test

period (only 2 out of 1,008), rather than any fundamental model weakness. The AUC figure confirms that the model's ranking ability is effectively perfect in this period, correctly ordering the relative likelihood of optimal windows even when the absolute frequency of positives is too low to compute a meaningful recall.

6.8.8 Feature Importance

Feature importance analysis carried out at the end of the model helped me show which variables contribute most to the model's predictions. In a Random Forest, this is measured through the Gini score, which captures how much each variable reduces uncertainty when the model makes a decision. The higher the score, the more that variable drives the classification.

The graph and table below present the top 20 and top 10 variables respectively. The bid-ask spread ranks first with a Gini score of 0.1342 (around 31% higher than the second variable) meaning it is by far the most decisive signal. When transaction costs are low, the market absorbs bonds easily and conditions favour a new issue. Realised intraday volatility ranks second (0.1020): the more uncertain prices are, the less receptive the market is to new supply. The broad market credit index comes third (0.0825), capturing the general state of credit conditions.



Looking at the top 10 as a whole, five of the variables are direct liquidity measures, confirming that liquidity dominates the model's decision-making and validating H1. The

presence of issuer_market_spread_gap in ninth place, which compares a specific issuer's spread against the broader market, supports H3, showing that issuer-level conditions matter beyond what aggregate indicators capture. Finally, market_regime_volatile in tenth place supports H4, confirming that whether the market is in a stressed regime is a meaningful signal for the model.

```
Top 10 variables:
bid_ask_spread_bps              0.1342
realized_volatility_intraday    0.1020
broad_market_credit_index       0.0825
market_depth                    0.0746
quoted_depth_bid                0.0698
quote_dispersion                0.0685
volatility_index                0.0659
quoted_depth_ask                0.0611
issuer_market_spread_gap        0.0483
market_regime_volatile          0.0315
dtype: float64
```

7. EXPECTED CONTRIBUTIONS AND SIGNIFICANCE

Across this paper we have illustrated how this research contributes to practical implementation of new techniques that are yet to be exploited, and that can help exploit information to thriving businesses. The existing literature on corporate debt issuance timing has treated the problem as a low-frequency, aggregate question: do firms issue more when spreads are tight? Do credit conditions predict issuance volumes? These are important questions, but as we discussed, they leave open a different and more precise one, can the intraday microstructure of the bond market tell us something specific about which days, or which hours, are best for a new transaction? This paper answers that question empirically for the first time, and in doing so it provides a template for future work at the intersection of market microstructure and corporate finance.

The methodological contribution is the demonstration that high-frequency bond market data (intraday bid-ask spreads, order flow imbalances, dealer quote competition, market depth snapshots) can be exploited to answer a corporate finance question. This is not obvious. The microstructure literature has used this data to study price discovery and liquidity provision; the corporate finance literature has largely ignored it. Showing that

the two can be combined, and that the combination yields predictive power, opens a research direction that goes well beyond the specific application studied here.

The practical contribution is more immediate. The thesis delivers a working prototype of a decision-support tool that an issuer's treasury team or a bank's DCM team could use to time bond transactions. Even saving a few basis points consistently (say, five to ten basis points per issuance by concentrating transactions in the windows the model identifies as optimal) translates into material reductions in interest costs on a typical bond offering. For a firm that accesses the capital markets several times a year, the cumulative benefit is significant. The tool does not replace human judgement; it adds a data-driven dimension to a decision that has traditionally been made on intuition and low-frequency market reads.

Finally, the ancillary empirical results contribute to ongoing debates in the microstructure literature. The validation of H1 implies that intraday order flow carries forward-looking information about credit conditions, a finding relevant to researchers and regulators interested in price discovery and market transparency. The results on H3 and H4 add nuance: the signal is not uniform across issuers or market regimes, which means the tool is most valuable precisely in the situations where human judgement is most uncertain.

8. FEASIBILITY, RISKS, AND MITIGATIONS

The project is built on existing methods applied to a new problem, which means the main risks are practical rather than conceptual. The most immediate is data access. The analysis relies on high-frequency transaction data, and while TRACE provides broad coverage of U.S. corporate bond trades, richer sources (RFQ logs, dealer quote records, multi-dealer platform data) are proprietary. The mitigation is to start with what is publicly available and demonstrate the concept; if the results hold with TRACE-derived features alone, the case for seeking proprietary data through academic partnerships or industry collaboration becomes much stronger. If some planned features cannot be constructed, proxy variables offer a workable alternative without compromising the core methodology.

The second risk is overfitting. With 65 features and a single calendar year of training data, a model can easily memorise noise rather than learn a generalizable pattern. The response is methodological discipline: rolling-window out-of-sample testing, regularisation, and a battery of simpler baseline models that serve as sanity checks. The temporal cross-

validation results already confirm that the selected Random Forest generalises well across sub-periods, with a standard deviation of ±0.025 across folds — the lowest of any model tested.

A subtler risk is economic significance. A model can be statistically accurate and still yield predictions whose practical value is marginal, either because the spread movements it identifies are too small to matter or because the costs of acting on its recommendations (execution delays, market impact, reputational considerations) outweigh the savings. The mitigation here is to frame the evaluation in economic terms from the outset: the relevant question is not whether the model is right more often than chance, but whether following its recommendations over many issuances produces a consistent and meaningful reduction in financing costs.

Finally, even a well-performing model faces the challenge of practitioner adoption. Issuance decisions involve qualitative considerations that no algorithm can fully capture, investor relations, internal approval timelines, liability management objectives. The model is therefore positioned as an additional input, not a replacement for human judgement. Interpretability is central to this positioning: the feature importance and SHAP analyses ensure that every recommendation can be explained in terms that a treasurer or banker already understands, which is the most reliable path to practical uptake.

9. POTENTIAL EXTENSIONS (BEYOND THE SCOPE OF THIS THESIS)

The methodology developed here is deliberately scoped to a specific problem: investment-grade corporate bonds, a single calendar year, a binary classification target, but each of those constraints is a design choice, not a fundamental limitation. Relaxing any one of them points toward a meaningful research extension.

The most natural next step is geographic. The analysis relies on TRACE data, which covers U.S. corporate bonds. European markets have their own transaction reporting infrastructure, and emerging market corporate bonds introduce higher volatility and lower liquidity, conditions that could either amplify the gains from timing (bigger swings to exploit) or increase the noise that obscures the signal. Testing whether the same feature set predicts optimal issuance conditions in different market structures would reveal how much of the predictive power is specific to U.S. microstructure and how much is universal.

A second direction is to move from public to private debt markets. Syndicated loans and private placements represent a large share of corporate borrowing, but their data infrastructure is far thinner than TRACE. Proxy signals from secondary loan trading or public credit indices could serve as inputs, though the methodology would need significant adaptation. The business case is strong: issuers in private markets have even less analytical infrastructure than public bond issuers, which means the marginal value of a predictive tool is potentially higher.

From a technical perspective, reinforcement learning offers an alternative framing of the problem that is worth exploring. Instead of classifying individual market states, an agent could learn an issuance policy by maximising a reward function defined in terms of cumulative cost savings. This reframes the question from prediction to decision-making, which is ultimately the objective. More immediately, the prototype developed here could be extended into a real-time alert system, a service that monitors live market data and notifies issuers when conditions shift into the optimal regime, integrating with existing treasury management workflows rather than requiring a separate analytical process.

10. CONCLUSION AND CALL TO ACTION

This project began with the observation that a gap existed, one that was hiding in plain sight. Corporate issuers and their investment banks have long relied on broad macroeconomic indicators and aggregate credit spread indices to decide when to bring a new bond to market. Yet the bond market, over the past decade, has generated an entirely different kind of data: granular, high-frequency, transaction-level records that capture the microstructure of trading in real time. Nobody had connected those two things. The question this thesis set out to answer was whether that connection was worth making, whether the rich intraday signals available in modern bond markets could actually improve the prediction of optimal issuance windows in a way that has practical value for issuers.

The answer is yes. Throughout this work, we have built and validated a machine learning framework that classifies market states as optimal or sub-optimal for new bond issuance with a cross-validated AUC of 0.9652, stable across time periods, robust to regime changes, and interpretable in terms that practitioners already understand. We have shown that the variables that matter most are not the ones traditionally monitored by issuers: not

the aggregate credit spread index, not the central bank policy signal, but the bid-ask spread in the secondary market, the depth of the order book, the intraday volatility of prices, and the degree of competition among dealers. We have demonstrated, through an unsupervised clustering exercise, that these microstructure signals organise the market into two coherent states entirely on their own, before any model is trained, before any label is assigned. And we have confirmed that three of the five hypotheses guiding this research hold empirically: microstructure signals predict short-term issuance conditions, issuer-specific dynamics matter beyond aggregate indicators, and the model adapts meaningfully to different market regimes.

Beyond what has been demonstrated here, the work also points clearly to where it can go next. The methodology is not specific to U.S. investment-grade bonds or to the 2023 sample period, it is a framework that can be extended to European and emerging markets, to private credit and syndicated loans, and to other asset classes where similar microstructure data exists. The remaining hypotheses, whether issuers systematically capitalise on predicted favourable trends, and whether the model generates economically significant savings in practice, require primary market transaction data that was not available for this study, and represent the most important direction for future validation. Each of these extensions is a research agenda in its own right.

The broader conclusion is one about what machine learning can do for finance. This work is one example of a wider possibility: that the analytical tools developed in data science can be brought to bear on corporate finance decisions that have historically been made on judgement, experience, and incomplete information. A treasurer deciding when to issue a bond is not so different from any other decision-maker trying to act optimally under uncertainty. The data to support that decision better has existed for years; what was missing was a framework to use it. That is what this thesis provides, and in doing so, it makes the case that intelligent, data-driven financing is not a theoretical aspiration but something that can be built today, with available data and available methods.

The call to action, then, is about a mindset as much as a methodology. We are living through a period in which information is generated faster, and in greater volume, than at any point in financial history. Transaction records, quote data, order flow signals, macroeconomic surprises, all of it is available, in real time, to anyone willing to build the infrastructure to use it. The question is whether the finance industry will treat this abundance as a resource or continue to make decisions the way it always has. The results

of this thesis suggest that the former is not only possible but already within reach: the gap between the data that exists and the decisions it could inform is one that can be closed, and closing it benefits both the companies that access capital markets and the investors and intermediaries that serve them. That is the case for innovation in financial analytics, not as an academic exercise, but as a practical commitment to using the information we have to make better decisions than we otherwise would.

AI Use Declaration

I, Diego Baltar Arriola, student of the Double degree of Law-Business & Analytics at Universidad Pontificia Comillas, in submitting my Final Year Dissertation titled How can advanced business analytics applied to high-frequency and cross-sectional bond market data improve predictive models for the pricing and issuance timing of corporate debt securities in public capital markets?", hereby declare that I have used Generative Artificial Intelligence tools such as ChatGPT or similar only in the context of the activities described below:

1. **Research idea brainstorming:** Used to generate and outline possible research areas.
2. **Critical thinking:** Used to identify counter-arguments to a specific thesis I intended to defend.
3. **References:** Used in conjunction with other tools, such as Google Scholar, to identify preliminary references that were subsequently verified and validated.
4. **Methodology:** Used to discover methods applicable to specific research problems.
5. **Code interpretation:** Used to conduct preliminary data analysis.
6. **Style and language correction:** Used to improve the linguistic and stylistic quality of the text.
7. **Synthesiser of complex literature:** Used to summarise and understand complex academic texts.
8. **Synthetic data generation:** Used for the creation of fictitious datasets.
9. **Reviewer:** Used to receive suggestions on how to improve and refine the work at different levels of scrutiny.
10. **Translator:** Used to translate texts from one language to another.

I affirm that all information and content presented in this dissertation are the product of my own research and individual effort, except where otherwise indicated and appropriate credit has been given — including adequate references in the dissertation and explicit indication of the purposes for which ChatGPT or similar tools were used. I am aware of the academic and ethical implications of submitting non-original work and accept the consequences of any violation of this declaration.

Date: 22-apr-2026

Signature: _Diego Baltar Arriola_

11. BIBLIOGRAPHY

Baker, M., & Wurgler, J. (2015). Do bond issuers successfully time the market? *Journal of Corporate Finance, 25*, 224–241. https://doi.org/10.1016/j.jcorpfin.2015.07.004

Bessembinder, H., & Maxwell, W. (2008). Markets: Transparency and the corporate bond market. *Journal of Economic Perspectives, 22*(2), 217–234. https://doi.org/10.1257/jep.22.2.217

Boyarchenko, N., & Elias, L. (2023). *Corporate credit conditions around the world: Novel facts through holistic data* (Staff Report No. 1074). Federal Reserve Bank of New York. https://doi.org/10.59576/sr.1074

Cai, N., Helwege, J., & Warga, A. (2007). Underpricing in the corporate bond market. *The Review of Financial Studies, 20*(6), 2021–2046. https://doi.org/10.1093/rfs/hhm048

Fermanian, J.-D., Guéant, O., & Pu, J. (2015). *The behavior of dealers and clients on the European corporate bond market: The case of multi-dealer-to-client platforms* (arXiv:1511.07773). arXiv. https://doi.org/10.48550/arXiv.1511.07773

Forest, J. J., Branch, B. S., & Berry, B. T. (2024). Trading activity in the corporate bond market: A SAD tale of macro-announcements and behavioral seasonality? *Risks, 12*(5), 80. https://doi.org/10.3390/risks12050080

Frank, M., & Nezafat, M. (2019). Market timing in corporate bond issuance. Journal of Corporate Finance, 58, 1–15. https://doi.org/10.1016/j.jcorpfin.2019.04.002

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

Nagler, F., & Ottonello, G. (2017). *Structural changes in corporate bond underpricing* (BAFFI CAREFIN Centre Research Paper Series No. 2017-48). Bocconi University & Vienna Graduate School of Finance. https://ssrn.com/abstract=2896758

O'Hara, M., & Zhou, X. A. (2021). Anatomy of a liquidity crisis: Corporate bonds in the COVID-19 crisis. *Journal of Financial Economics, 142*(1), 46–68. https://doi.org/10.1016/j.jfineco.2021.05.052

Shin, S. S., Zhou, X., & Zhu, Q. (2025). The growing index effect in the corporate bond market. SMU Cox School of Business Research Paper No. 25-7. https://doi.org/10.2139/ssrn.5181327

ANEX I

1. Library Imports

```
import os
import warnings
from pathlib import Path

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import (
    train_test_split, TimeSeriesSplit, GridSearchCV, cross_val_score
)
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
```

```
    roc_auc_score, confusion_matrix, ConfusionMatrixDisplay,
    roc_curve, silhouette_score, classification_report
)

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neural_network import MLPClassifier

# Global plot settings
warnings.filterwarnings("ignore")
sns.set_theme(style="whitegrid")
plt.rcParams["figure.dpi"] = 120

print("Libraries loaded successfully")
```

2. Data Loading

```
POSSIBLE_FILES = [
    Path("TFG_AnalyticaBondMarket.xlsx"),
    Path("./TFG_AnalyticaBondMarket.xlsx"),
    Path("/mnt/data/TFG_AnalyticaBondMarket.xlsx"),
]
```

```
FILE = None
for f in POSSIBLE_FILES:
    if f.exists():
        FILE = f
        break

if FILE is None:
    print("Working directory:", os.getcwd())
    print("Visible files:", os.listdir("."))
    raise FileNotFoundError(
        "Cannot find 'TFG_AnalyticaBondMarket.xlsx'. "
        "Place the Excel file in the same folder as this script."
    )

print("File found:", FILE)
xls = pd.ExcelFile(FILE)
print("Available sheets:", xls.sheet_names)

df_raw = pd.read_excel(FILE, sheet_name="intraday_bond_market")
print(f"Dataset dimensions: {df_raw.shape[0]:,} rows x {df_raw.shape[1]} columns")
display(df_raw.head())
```

3. Initial Inspection

```
print("--- Data types and missing values ---")
summary = pd.DataFrame({
    "dtype":     df_raw.dtypes.astype(str),
    "n_nulls":   df_raw.isnull().sum(),
    "pct_nulls": (df_raw.isnull().sum() / len(df_raw) * 100).round(2),
    "n_unique":  df_raw.nunique()
}).sort_values("pct_nulls", ascending=False)
display(summary)

print(f"Duplicates: {df_raw.duplicated().sum()}")
print("--- Descriptive statistics (numeric variables) ---")
display(df_raw.describe().T)

# Distribution of the target variable (if present)
if "optimal_issuance_window" in df_raw.columns:
    fig, ax = plt.subplots(figsize=(5, 3))
    df_raw["optimal_issuance_window"].value_counts().plot(
        kind="bar", ax=ax, color=["#4C72B0", "#DD8452"])
    ax.set_title("Target Variable Distribution")
    ax.set_xlabel("optimal_issuance_window")
    ax.set_ylabel("Count")
    plt.tight_layout()
    plt.show()
```

4. Basic Data Cleaning

```
df = df_raw.copy()


# 4.1 Remove duplicates
n_dup = df.duplicated().sum()
df = df.drop_duplicates().reset_index(drop=True)
print(f'Duplicates removed: {n_dup}')


# 4.2 Parse date columns
if "timestamp" in df.columns:
    df["timestamp"] = pd.to_datetime(df["timestamp"], errors="coerce")
if "date" in df.columns:
    df["date"] = pd.to_datetime(df["date"], errors="coerce")
elif "timestamp" in df.columns:
    df["date"] = pd.to_datetime(df["timestamp"].dt.date, errors="coerce")


# 4.3 Derive temporal features from timestamp
if "timestamp" in df.columns:
    df["hour"]    = df["timestamp"].dt.hour
    df["minute"]  = df["timestamp"].dt.minute
    df["dow_num"] = df["timestamp"].dt.dayofweek
    df["month"]   = df["timestamp"].dt.month


# 4.4 Convert binary string variables to numeric format
```

```
binary_cols = [
    "benchmark_flag", "macro_news_dummy", "month_end_rebalance_dummy",
    "optimal_issuance_window", "target_spread_tightening_1d",
    "target_spread_tightening_3d"
]
for c in binary_cols:
    if c in df.columns:
        df[c] = pd.to_numeric(df[c], errors="coerce")

print(f"Dimensions after basic cleaning: {df.shape}")
display(df.head())
```

5. Missing Value Treatment

```
# Inspect missing values
df_missing = pd.DataFrame({
    "n_nulls":   df.isnull().sum(),
    "pct_nulls": (df.isnull().sum() / len(df) * 100).round(2)
}).sort_values("pct_nulls", ascending=False)
display(df_missing.head(20))

# Drop columns exceeding the 40% missing threshold
high_missing = df_missing[df_missing["pct_nulls"] > 40].index.tolist()
print("Columns dropped due to excessive missingness:", high_missing)
df = df.drop(columns=high_missing)
```

```
# Imputation: median for numeric variables, mode for categorical
num_cols = df.select_dtypes(include=[np.number]).columns.tolist()
cat_cols = df.select_dtypes(include=["object", "category"]).columns.tolist()

if num_cols:
    imp_num = SimpleImputer(strategy="median")
    df[num_cols] = imp_num.fit_transform(df[num_cols])

if cat_cols:
    imp_cat = SimpleImputer(strategy="most_frequent")
    df[cat_cols] = imp_cat.fit_transform(df[cat_cols])

print(f"Remaining missing values: {df.isnull().sum().sum()}")
```

6. Outlier Detection and Treatment (IQR + Winsorisation)

```
def outlier_limits_iqr(series):
    q1  = series.quantile(0.25)
    q3  = series.quantile(0.75)
    iqr = q3 - q1
    return q1 - 1.5 * iqr, q3 + 1.5 * iqr

outlier_cols = [c for c in [
    "bid_ask_spread_bps", "market_depth", "buy_volume", "sell_volume",
```

```
    "trade_count", "secondary_credit_spread_bps", "realized_volatility_intraday"
] if c in df.columns]

# IQR detection
print("--- Outliers detected (IQR method) ---")
for col in outlier_cols:
    low, high = outlier_limits_iqr(df[col])
    n_out = ((df[col] < low) | (df[col] > high)).sum()
    print(f"  {col}: {n_out} outliers")

# Winsorisation at the 1st and 99th percentiles
for col in outlier_cols:
    p1  = df[col].quantile(0.01)
    p99 = df[col].quantile(0.99)
    df[col] = df[col].clip(lower=p1, upper=p99)

print("Outliers treated via 1%-99% winsorisation")
```

7. Univariate Descriptive Analysis

```
# Histograms and boxplots for the five key microstructure variables
eda_cols = [c for c in [
    "bid_ask_spread_bps", "market_depth", "order_flow_imbalance",
    "secondary_credit_spread_bps", "realized_volatility_intraday"
] if c in df.columns]
```

```
fig, axes = plt.subplots(len(eda_cols), 2, figsize=(12, 4 * len(eda_cols)))
for i, col in enumerate(eda_cols):
    sns.histplot(df[col], kde=True, ax=axes[i, 0], color="#4C72B0")
    axes[i, 0].set_title(f'Histogram – {col}')
    sns.boxplot(x=df[col], ax=axes[i, 1], color="#DD8452")
    axes[i, 1].set_title(f'Boxplot – {col}')
plt.tight_layout()
plt.suptitle("Univariate Analysis – Key Variables", y=1.01, fontsize=14)
plt.show()

# Target variable distribution
if "optimal_issuance_window" in df.columns:
    fig, ax = plt.subplots(figsize=(5, 3))
    counts = df["optimal_issuance_window"].value_counts()
    counts.plot(kind="bar", ax=ax, color=["#4C72B0", "#DD8452"], edgecolor="white")
    for p in ax.patches:
        ax.annotate(f"{p.get_height():,.0f}", (p.get_x() + 0.1, p.get_height() + 20))
    ax.set_title("Target Distribution: optimal_issuance_window")
    ax.set_xlabel("Class")
    ax.set_ylabel("Frequency")
    plt.tight_layout()
    plt.show()
    print("Target distribution:\n", counts)
```

8. Bivariate Analysis and Correlations

```
# Pairplot (sample n=500) and full-sample correlation heatmap
if len(eda_cols) > 1:
    sample = df[eda_cols].sample(min(500, len(df)), random_state=42)
    sns.pairplot(sample, diag_kind="kde", plot_kws={"alpha": 0.5})
    plt.suptitle("Pairplot – Microstructure Variables", y=1.01)
    plt.show()

corr_cols = eda_cols.copy()
if "optimal_issuance_window" in df.columns:
    corr_cols.append("optimal_issuance_window")

fig, ax = plt.subplots(figsize=(9, 7))
sns.heatmap(
    df[corr_cols].corr(),
    annot=True, cmap="coolwarm", fmt=".2f",
    linewidths=0.5, ax=ax
)
ax.set_title("Correlation Matrix")
plt.tight_layout()
plt.show()

# Boxplot of target vs most important variable
```

```
if "bid_ask_spread_bps" in df.columns and "optimal_issuance_window" in df.columns:
    fig, ax = plt.subplots(figsize=(6, 4))
    sns.boxplot(
        x="optimal_issuance_window", y="bid_ask_spread_bps",
        data=df, palette="Set2", ax=ax
    )
    ax.set_title("Bid-Ask Spread by Issuance Window")
    ax.set_xlabel("Optimal (1) vs Non-Optimal (0)")
    plt.tight_layout()
    plt.show()
```

9. Feature Engineering

```
# Six engineered features with economic rationale for bond issuance prediction

# 1. Net order flow: net directional buying/selling pressure
if {"buy_volume", "sell_volume"}.issubset(df.columns):
    df["net_order_flow_calc"] = df["buy_volume"] - df["sell_volume"]
    df["ofi_calc"] = (
        (df["buy_volume"] - df["sell_volume"]) /
        (df["buy_volume"] + df["sell_volume"] + 1e-9)
    )

# 2. Liquidity-volatility ratio: spread wide relative to price uncertainty
if {"bid_ask_spread_bps", "realized_volatility_intraday"}.issubset(df.columns):
```

```
    df["liquidity_vol_ratio"] = (
        df["bid_ask_spread_bps"] / (df["realized_volatility_intraday"] + 1e-9)
    )

# 3. Depth-OFI interaction: deep market with directional buying pressure
if {"market_depth", "order_flow_imbalance"}.issubset(df.columns):
    df["depth_ofi_interaction"] = df["market_depth"] * df["order_flow_imbalance"]

# 4. Issuer-market spread gap: issuer trading cheap/expensive vs broad market
If {"secondary_credit_spread_bps", "broad_market_credit_index"}.issubset(df.columns):
    df["issuer_market_spread_gap"] = (
        df["secondary_credit_spread_bps"] - df["broad_market_credit_index"]
    )

# 5. Bottom-up credit spread: bond yield minus risk-free rate
if {"bond_yield", "risk_free_rate"}.issubset(df.columns):
    df["credit_spread_calc"] = df["bond_yield"] - df["risk_free_rate"]

print(f"Features after engineering: {df.shape[1]} columns")
display(df.head(3))
```

10. Daily Aggregation (Issuer × Day Panel)

```
# Aggregate intraday snapshots to issuer-day level:
# mean for continuous variables, sum for volumes, max for binary flags
```

```
num_cols_df = df.select_dtypes(include=[np.number]).columns.tolist()
agg_dict = {c: "mean" for c in num_cols_df}

for c in ["buy_volume", "sell_volume", "trade_count", "net_order_flow_calc"]:
    if c in agg_dict:
        agg_dict[c] = "sum"

for c in ["optimal_issuance_window", "macro_news_dummy", "month_end_rebalance_dummy",
          "target_spread_tightening_1d", "target_spread_tightening_3d"]:
    if c in agg_dict:
        agg_dict[c] = "max"

daily = df.groupby(["date", "issuer_id"], as_index=False).agg(agg_dict)

# Carry forward categorical variables using first observation per group
for c in ["issuer_type", "sector", "rating_bucket", "market_regime", "benchmark_flag"]:
    if c in df.columns:
        aux = df.groupby(["date", "issuer_id"])[c].first().reset_index()
        daily = daily.merge(aux, on=["date", "issuer_id"], how="left")

daily = daily.sort_values(["date", "issuer_id"]).reset_index(drop=True)
print(f"Daily panel: {daily.shape[0]:,} rows x {daily.shape[1]} columns")
display(daily.head())
```

11. Target Definition

```
TARGET = "optimal_issuance_window"

if TARGET not in daily.columns:
    conds = []
    if "secondary_credit_spread_bps" in daily.columns:
        conds.append(daily["secondary_credit_spread_bps"] < daily["secondary_cre...
    if "market_depth" in daily.columns:
        conds.append(daily["market_depth"] > daily["market_depth"].quantile(0.55...
    if "realized_volatility_intraday" in daily.columns:
        conds.append(daily["realized_volatility_intraday"] < daily["realized_vol...
    if conds:
        daily[TARGET] = np.logical_and.reduce(conds).astype(int)
        print(f"Target construido desde condiciones. Distribución:\n{daily[TARGE...
    else:
        raise ValueError("No es posible construir el target.")
else:
    print(f"Target ya presente. Distribución:\n{daily[TARGET].value_counts()}")
```

12. Categorical Encoding

```
cat_cols_model = [c for c in ["issuer_type", "sector", "rating_bucket", "market_regime"]
                  if c in daily.columns]
model_df = pd.get_dummies(daily.copy(), columns=cat_cols_model, drop_first=True)


# Exclude non-feature columns from the feature matrix
```

```
exclude_cols = [c for c in [
    "date", "issuer_id", "timestamp", "spread_t1", "spread_t3",
    "spread_change_next_1d_calc", "spread_change_next_3d_calc"
] if c in model_df.columns]

feature_cols = [c for c in model_df.columns
                if c not in exclude_cols + [TARGET]
                and model_df[c].dtype != object]

X = model_df[feature_cols].copy()
y = model_df[TARGET].astype(int).copy()
print(f"Features: {X.shape[1]}  |  Observations: {X.shape[0]}")
```

13. Temporal Train / Test Split (80-20)

```
# Temporal split to avoid data leakage: train on past, test on future
temp_df = pd.concat([daily[["date", "issuer_id"]], X, y.rename(TARGET)], axis=1)
temp_df = temp_df.sort_values("date")

split_idx = int(len(temp_df) * 0.80)
train_df  = temp_df.iloc[:split_idx].copy()
test_df   = temp_df.iloc[split_idx:].copy()

X_train = train_df[feature_cols]
X_test  = test_df[feature_cols]
```

```
y_train = train_df[TARGET]
y_test  = test_df[TARGET]

print(f"Train: {X_train.shape} | {train_df['date'].min().date()} to {train_df['date'].max().date()}")
print(f"Test : {X_test.shape} | {test_df['date'].min().date()} to {test_df['date'].max().date()}")
```

14. Feature Scaling (StandardScaler)

```
# Impute residual missings and scale — scaler fitted on train only, applied to test
imp    = SimpleImputer(strategy="median")
X_train_imp = pd.DataFrame(imp.fit_transform(X_train), columns=X_train.columns, index=X_train.index)
X_test_imp  = pd.DataFrame(imp.transform(X_test), columns=X_test.columns, index=X_test.index)

scaler = StandardScaler()
X_train_sc = pd.DataFrame(scaler.fit_transform(X_train_imp), columns=X_train.columns, index=X_train.index)
X_test_sc  = pd.DataFrame(scaler.transform(X_test_imp), columns=X_test.columns, index=X_test.index)

print(f"Train scaled: {X_train_sc.shape} | Test scaled: {X_test_sc.shape}")
```

15. Principal Component Analysis (PCA)

```
# Assess intrinsic dimensionality of the 65-variable feature space
```

```
pca_full = PCA().fit(X_train_sc)
cum_var  = np.cumsum(pca_full.explained_variance_ratio_)

fig, ax = plt.subplots(figsize=(9, 4))
ax.plot(range(1, len(cum_var) + 1), cum_var, marker="o", ms=3)
ax.axhline(0.80, color="red",   ls="--", label="80%")
ax.axhline(0.90, color="green", ls="--", label="90%")
ax.set_title("Cumulative Variance Explained by PCA")
ax.set_xlabel("Number of components")
ax.set_ylabel("Cumulative variance")
ax.legend()
plt.tight_layout()
plt.show()

n_comp = int(np.argmax(cum_var >= 0.90)) + 1
print(f"Components required for 90% variance: {n_comp}")

# Fit PCA and project training data onto first two components for visualisation
pca = PCA(n_components=min(n_comp, 8))
X_train_pca = pca.fit_transform(X_train_sc)
X_test_pca  = pca.transform(X_test_sc)

pca_plot = pd.DataFrame(X_train_pca[:, :2], columns=["PC1", "PC2"])
pca_plot["target"] = y_train.values
```

```
fig, ax = plt.subplots(figsize=(7, 5))
sns.scatterplot(data=pca_plot, x="PC1", y="PC2", hue="target",
                palette={0: "#4C72B0", 1: "#DD8452"}, alpha=0.7, ax=ax)
ax.set_title("PCA Projection – PC1 vs PC2")
plt.tight_layout()
plt.show()
```

16. KMeans Clustering (Market Regime Identification)

```
# Unsupervised clustering on first four PCA components to identify market regimes
X_cluster = X_train_pca[:, :min(4, X_train_pca.shape[1])]
k_values, wss, sil_scores = range(2, 7), [], []

for k in k_values:
    km     = KMeans(n_clusters=k, random_state=42, n_init=25)
    labels = km.fit_predict(X_cluster)
    wss.append(km.inertia_)
    sil_scores.append(silhouette_score(X_cluster, labels))

# Elbow and silhouette criteria to select optimal k
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(list(k_values), wss, marker="o")
ax1.set_title("Elbow Method"); ax1.set_xlabel("k"); ax1.set_ylabel("WSS")
ax2.plot(list(k_values), sil_scores, marker="o", color="orange")
```

```
ax2.set_title("Silhouette Score"); ax2.set_xlabel("k"); ax2.set_ylabel("Silhouette")

plt.tight_layout()

plt.show()


best_k = list(k_values)[int(np.argmax(sil_scores))]

print(f"Optimal k (silhouette): {best_k}")


kmeans_best   = KMeans(n_clusters=best_k, random_state=42, n_init=25)

train_clusters = kmeans_best.fit_predict(X_cluster)

test_clusters  = kmeans_best.predict(X_test_pca[:, :min(4, X_test_pca.shape[1])])


X_train_sc["cluster_kmeans"] = train_clusters

X_test_sc["cluster_kmeans"]  = test_clusters

fig, ax = plt.subplots(figsize=(7, 5)) sns.scatterplot(x=X_train_pca[:, 0], y=X_train_pca[:, 1], hue=train_clusters, palette="tab10", alpha=0.7, ax=ax)

ax.set_title("Clusters KMeans sobre PCA"); ax.set_xlabel("PC1"); ax.set_ylabel("PC2")
plt.tight_layout(); plt.show()
```

17. Classification Models

```
# Seven classifiers trained and evaluated on the same standardised feature matrix

models = {

   "Logistic Regression": LogisticRegression(

      max_iter=2000, class_weight="balanced", random_state=42),

   "KNN":                KNeighborsClassifier(n_neighbors=7),

   "SVM":              SVC(probability=True, class_weight="balanced", random_state=42),

   "Decision Tree":       DecisionTreeClassifier(
```

```
        max_depth=5, min_samples_leaf=10, class_weight="balanced", random_state=42),
    "Random Forest":       RandomForestClassifier(
        n_estimators=300, max_depth=8, min_samples_leaf=8,
        class_weight="balanced", random_state=42, n_jobs=-1),
    "Gradient Boosting":   GradientBoostingClassifier(
        n_estimators=200, max_depth=4, learning_rate=0.05,
        subsample=0.9, random_state=42),
    "MLP":                 MLPClassifier(
        hidden_layer_sizes=(128, 64), max_iter=500,
        random_state=42, early_stopping=True),
}

results = {}
for name, model in models.items():
    print(f"  Training: {name}...", end=" ")
    model.fit(X_train_sc, y_train)
    y_pred = model.predict(X_test_sc)
    y_prob = (model.predict_proba(X_test_sc)[:, 1]
              if hasattr(model, "predict_proba")
              else model.decision_function(X_test_sc))
    results[name] = {
        "model":     model,
        "accuracy":  accuracy_score(y_test, y_pred),
        "precision": precision_score(y_test, y_pred, zero_division=0),
```

```
        "recall":    recall_score(y_test, y_pred, zero_division=0),
        "f1":        f1_score(y_test, y_pred, zero_division=0),
        "auc":       roc_auc_score(y_test, y_prob),
        "y_pred":    y_pred,
        "y_prob":    y_prob,
    }
    print(f"AUC = {results[name]['auc']:.4f}")

results_df = pd.DataFrame(
    {k: {m: results[k][m] for m in ["accuracy","precision","recall","f1","auc"]}
     for k in results}
).T.sort_values("auc", ascending=False)

print("\n--- Model comparison ---")
display(results_df)
```

18. Detailed Evaluation — Best Model

```
best_name = results_df.index[0]
best_res  = results[best_name]
print(f"Best model: {best_name}\n")
print(classification_report(y_test, best_res["y_pred"],
                            target_names=["Non-Optimal", "Optimal"]))

# Confusion matrix
```

```
cm   = confusion_matrix(y_test, best_res["y_pred"])
disp = ConfusionMatrixDisplay(cm, display_labels=["Non-Optimal", "Optimal"])
fig, ax = plt.subplots(figsize=(5, 4))
disp.plot(cmap="Blues", ax=ax)
ax.set_title(f"Confusion Matrix – {best_name}")
plt.tight_layout()
plt.show()
```

19. Comparative ROC Curves

```
fig, ax = plt.subplots(figsize=(8, 6))
colors = plt.cm.tab10(np.linspace(0, 1, len(results)))
for (name, res), color in zip(results.items(), colors):
    fpr, tpr, _ = roc_curve(y_test, res["y_prob"])
    ax.plot(fpr, tpr, label=f"{name} (AUC={res['auc']:.3f})", color=color)
ax.plot([0,1],[0,1],"k--")
ax.set_xlabel("False Positive Rate")
ax.set_ylabel("True Positive Rate")
ax.set_title("ROC Curves – Model Comparison")
ax.legend(fontsize=8)
plt.tight_layout()
plt.show()
```

20. Temporal Cross-Validation (TimeSeriesSplit, 5 folds)

```
tscv = TimeSeriesSplit(n_splits=5)
cv_summary = {}
```

```
for name, model in models.items():
    scores = cross_val_score(
        model, X_train_sc, y_train,
        cv=tscv, scoring="roc_auc"
    )
    cv_summary[name] = {"cv_auc_mean": scores.mean(), "cv_auc_std": scores.std()}
    print(f"  {name}: AUC = {scores.mean():.4f} ± {scores.std():.4f}")

cv_df = pd.DataFrame(cv_summary).T.sort_values("cv_auc_mean", ascending=False)
display(cv_df)
fig, ax = plt.subplots(figsize=(8, 4)) ax.barh(cv_df.index, cv_df["cv_auc_mean"], xerr=cv_df["cv_auc_std"], color="#4C72B0", edgecolor="white", capsize=4) ax.set_xlabel("AUC (media ± std)")
ax.set_title("Cross-Validation Temporal – Comparativa AUC")
ax.invert_yaxis(); plt.tight_layout(); plt.show()
```

21. Hyperparameter Tuning — Random Forest (GridSearchCV)

```
# Grid search over 18 configurations; scoring metric: AUC; CV: TimeSeriesSplit (4 folds)
param_grid_rf = {
    "n_estimators":    [200, 300],
    "max_depth":       [5, 8, 12],
    "min_samples_leaf":[5, 8, 12],
}

grid_rf = GridSearchCV(
    RandomForestClassifier(class_weight="balanced", random_state=42, n_jobs=-1),
```

```
    param_grid=param_grid_rf,
    cv=TimeSeriesSplit(n_splits=4),
    scoring="roc_auc",
    n_jobs=-1,
    verbose=1
)
grid_rf.fit(X_train_sc, y_train)

print("Best parameters (RF):", grid_rf.best_params_)
print("Best CV AUC:",          round(grid_rf.best_score_, 4))

best_rf = grid_rf.best_estimator_
rf_prob = best_rf.predict_proba(X_test_sc)[:, 1]
rf_pred = best_rf.predict(X_test_sc)
print(f"Test AUC (tuned RF): {roc_auc_score(y_test, rf_prob):.4f}")
print(f"Test F1  (tuned RF): {f1_score(y_test, rf_pred, zero_division=0):.4f}")
```

22. Feature Importance (Random Forest — Gini Impurity)

```
feat_imp = pd.Series(
    best_rf.feature_importances_, index=X_train_sc.columns
).sort_values()
top20 = feat_imp.tail(20)

fig, ax = plt.subplots(figsize=(9, 7))
```

```
top20.plot(kind="barh", ax=ax, color="#4C72B0", edgecolor="white")

ax.set_title("Top 20 Most Important Features – Tuned Random Forest")

ax.set_xlabel("Feature Importance (Gini)")

plt.tight_layout()

plt.show()


print("\nTop 10 variables:")

print(feat_imp.tail(10).sort_values(ascending=False).round(4))
```

23. Cluster Analysis

```
# Characterise market regimes identified by KMeans in terms of key microstructure variables

cluster_df = X_train_sc.copy()

cluster_df["cluster"] = train_clusters

cluster_df["target"]  = y_train.values


cluster_summary = cluster_df.groupby("cluster")["target"].agg(["mean", "count"])

cluster_summary.columns = ["pct_optima", "n_obs"]

print("Proportion of optimal windows by cluster:")

display(cluster_summary)


key_vars = [c for c in ["bid_ask_spread_bps", "market_depth", "realized_volatility_intraday"]

        if c in cluster_df.columns]

if key_vars:
```

```
display(cluster_df.groupby("cluster")[key_vars].mean().round(3))
```