Enhancing Credit Risk Models at Revolut by Combining Deep Feature Synthesis and Marginal Information Value

Federico Spinella
Revolut Group Holdings Ltd, London, United Kingdom
Global Credit Management - Data Science
Senior Data Scientist
federico.spinella@revolut.com

Tadas Krisciunas
Revolut Group Holdings Ltd, Vilnius, Lithuania
Global Credit Management - Data Science
Lead Data Scientist
tadas.krisciunas@revolut.com

August 29, 2025

Abstract

In the domain of credit risk modelling, the generation of predictive features from complex relational datasets is paramount for accurate risk assessment. Traditional methods often involve manual feature engineering, which can be time-consuming and may not capture intricate relationships within the data. Deep Feature Synthesis (DFS) emerges as a powerful automated technique that addresses these limitations by automatically constructing features from relational and temporal data. This paper explores combining it with Marginal Information Value (MIV) based feature selection to derive an automated credit acquisition scorecard generation process.

The ability to identify insightful features from relational data automatically is particularly crucial for financial institutions such as Revolut, which handle vast amounts of transactional data across diverse global markets. Automated feature engineering with DFS and feature selection with MIV allow for the systematic exploration of potential predictors, uncovering complex patterns and interactions that might be missed through manual approaches. This is of significant interest, as it can lead to more robust and accurate credit risk models, ultimately impacting crucial business decisions related to lending, pricing, and risk management, especially for individuals with limited traditional credit history.

The paper will explore how the combination of DFS and MIV allows our models to capture nuanced patterns in credit, transactional data and user behaviour, leading to noticeable improvements in model performance and business outcomes, while at the same time automating large parts of the model development process.

1 Introduction

1.1 Business Motivation

An accurate assessment of credit risk is the cornerstone of modern financial institutions. The ability to correctly predict the probability of default for new applicants is critical for prudent risk management and sustainable profitability [24].

The development of robust acquisition PD models traditionally relies on a combination of expert knowledge and statistical techniques to engineer predictive features. However, the increasing complexity and volume of data, particularly from transactional and alternative sources, present

a significant challenge to manual feature engineering approaches. These processes are often resource-intensive and may fail to uncover subtle, non-linear relationships within large relational datasets. In a multi-national banking organisation engaging in retail lending across progressively more jurisdictions such as Revolut, a streamlined, automated and generalised methodology to develop candidate credit scorecards is required if the growth in the headcount of data scientists is to be moderated.

To address these limitations, Revolut has developed a sophisticated and automated methodology to develop PD models. This methodology is centred on Deep Feature Synthesis (DFS), an algorithm for automated feature generation from relational data [1, 11]. DFS automates feature generation by systematically exploring the relational structure of the data, thereby creating a rich set of candidate features. It then further distils the comprehensive and large set of features generated by DFS utilising Marginal Information Value (MIV) based feature selection. Our methodology is an instance of automated machine learning (AutoML), which seeks to automate the end-to-end process of applying machine learning fine-tuned to be applicable in a regulated environment where model explainability is key [24], and to be suitable to be applied across geographies and jurisdictions with differing credit regulations such as the EU, the UK and US.

1.2 Related Work

The challenge of creating predictive features from raw data is a central theme in machine learning and particularly in credit risk modelling. While this paper details an applied methodology combining Deep Feature Synthesis (DFS) and Marginal Information Value (MIV), it is useful to position this approach within the broader context of automated feature engineering and selection techniques. The paragraphs that follow provide a cursory overview of the state-of-the-art techniques and their applicability to the credit modelling area.

1.2.1 Feature Generation

DFS automates the generation process by systematically applying mathematical primitives (e.g., sum, average) across relational data paths. Its strength lies in creating a vast and rich set of candidate features whose lineage is explicit and therefore interpretable. This contrasts with other automated feature generation techniques that are described below.

Genetic Programming (GP) GP uses evolutionary algorithms to "evolve" new features as mathematical expressions [15, 16]. While highly flexible, GP can often produce overly complex and unintuitive features, posing a significant challenge for model validation and regulatory review in a domain where transparency is paramount.

Large Language Models (LLMs) Given the increasingly wide adoption of Large Language Models (LLMs) in generating code [19], it is natural to explore the usage of LLMs in new feature generation processes. This is a promising avenue of research, but is in early stages [17, 18]. The topic is outside of the scope of this paper, but is noted as a future research area.

1.2.2 Feature Selection

Following feature generation, a selection process is crucial. Our approach uses MIV, a greedy forward-selection method rooted in information theory that quantifies the additional predictive power a variable provides, given the variables already in the model. Other common methods include[20] the ones briefly described below.

Recursive Feature Elimination (RFE) A wrapper method that starts with all candidate features, builds a model, and recursively removes the least important feature until a desired number remains [20]. RFE can be more computationally intensive than MIV, especially with a large initial feature set, making it less scalable.

Regularization-based Selection Methods like LASSO (L1 regularization) perform feature selection implicitly by shrinking the coefficients of less important features to zero [21]. This is often effective but integrates the selection and modelling steps, making it more difficult to perform residual monitoring as described in Sec. 3.

SHAP-based Selection Modern techniques leverage game-theoretic concepts like SHAP (SHapley Additive exPlanations)[22, 23] to measure feature importance. By training a model and aggregating the SHAP values for each feature, one can discard those with low overall impact. This is a powerful, model-agnostic technique, though often applied to more complex "black-box" models. Besides, like RFE, it is more computationally complex than the approach discussed in the paper.

1.2.3 Conclusions

The combination of DFS and MIV is uniquely suited to the credit risk domain for several reasons. First, the entire pipeline prioritizes transparency and interpretability. DFS creates features with a clear, traceable logic, and the subsequent WoE transformation and MIV selection are directly tied to logistic regression, a highly transparent final model favored by regulators, especially in an IRB context [24]. This avoids the "black-box" problem while still permitting the utilisation of alternative data sources such as Revolut's mobile app interactions. Second, the methodology is *efficient and scalable*, allowing for the rapid development of robust scorecards across multiple jurisdictions, a key requirement for a global institution like Revolut. Finally, it is built upon metrics like Information Value, which are well-established and trusted within the credit scoring industry, facilitating straightforward validation and governance.

1.3 Outline

This paper provides a comprehensive overview of this automated model development and monitoring lifecycle. Section 2 details the core stages of the process: defining the target variable, feature generation and preprocessing, feature selection, and model specification. Section 3 describes the architecture and results of an automated feature monitoring system. Section 4 presents case studies which illustrate the practical business benefits of this approach. Finally, Section 5 discusses future research directions.

2 Model Development Methodology

Having motivated the methodology, we provide details and a more nuanced mathematical grounding of it. The model development process is structured into distinct sequential phases: feature generation, feature selection, model training and benchmarking, ultimately leading to a final model. Fig. 1 showcases the schematic representation of the model development lifecycle, and the sections below go into depth discussing the specific steps outlined in the schema. To begin with, the overall model structure is discussed. Then, feature generation approach utilising DFS opted for in Revolut is outlined. Subsequently, the feature selection approach adopting MIV is detailed. Finally, some remarks about the model training and benchmarking are provided.

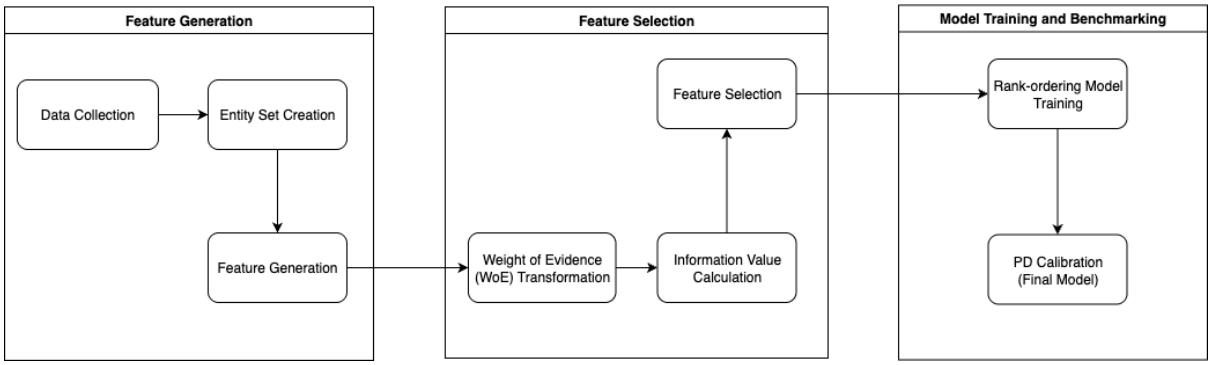

Figure 1: An overview of the Probability of Default (PD) Model Development Lifecycle

2.1 Model Structure

There might not be a one-size-fits-all model structure that ensures best performance in all edge cases, across differing geographical jurisdictions utilising different data sources for credit risk assessment, and it is best to not commit to a structure without understanding if it will guarantee best performance in all of the disparate cases. However, there are boundaries on what types of models Revolut would consider and some broad-strokes structural limitations have been set on models to permit for automation, as described below. The core principle is splitting the rank-ordering score, responsible for ranking customers in terms of credit risk form highest to lowest, and the PD calibration, responsible for accurate probability of default assessments as is a relatively standard practice in internal rating systems [12]. The two distinct model components are developed separately and their performance is assessed using different metrics, as detailed in Sec. 2.4. Taking this key principle in mind, the following are the key elements of model design that apply to PD models at Revolut. The model structural components listed below are also presented schematically below in Fig. 2.

1. A rank-ordering application scorecard.
   (a) Definition of a good and a bad loan to be used as a target in model development.
   (b) The choice of the model development, hold-out/test and out-of-time samples.
   (c) Identification of a set of factors predictive of the targets to be used as inputs into the model, and preparation thereof for modelling.
   (d) Optimal choice, fitting and calibration of the statistical or machine learning model.
2. Probability of Default model/calibration/assessment.
   (a) Definition of Default (DoD) used. It must meet regulatory guidelines for the respective jurisdiction.
   (b) Definitions and cut-off points for risk grades.
   (c) The choice of the sample to perform PD calibration on.
   (d) (Only applies if (1) and (2) are separate models.) A probability calibration exercise to transform rank-ordering scores produced in (1) to PD assessments.

Keep in mind that (1) and (2) can be the same model, in which case (1a), (1b) would be equivalent to (2a), (2c) respectively.

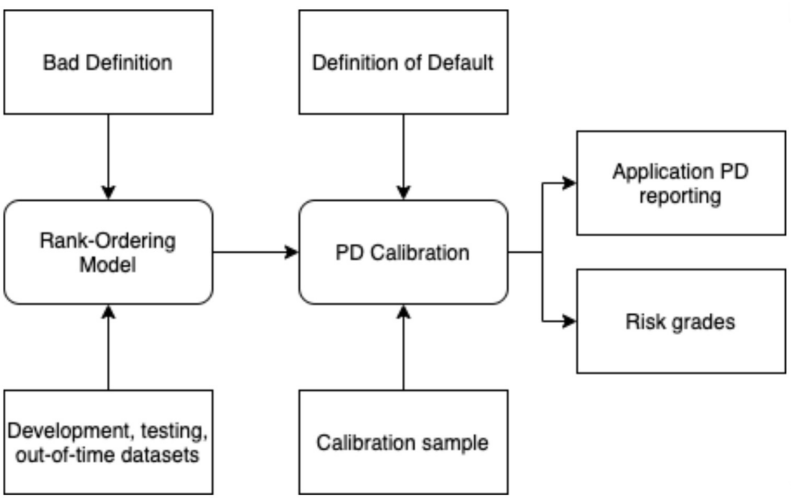

Figure 2: The structural representation of Revolut's probability of default (PD) model components

2.2 Feature Generation

The section outlines Revolut's approach to feature generation. It begins by overviewing the process used to select the model target variable, moves into covariate generation, and ends with the preprocessing steps applied to the generated covariates.

2.2.1 Target Variable Definition

A robust definition of the target variable is fundamental for scorecard development. It consists of two key parameters: the bad/default event and the prediction horizon.

2.2.2 Bad Definition vs. Definition of Default

A distinction is made between a "Bad Definition" used for developing the rank-ordering component of a model and the formal "Definition of Default" (DoD) used for PD calibration. The DoD is subject to regulatory scrutiny and must align with internal policies, which are localised for different jurisdictions. However, the Bad Definition is independent of regulatory considerations and can be chosen to optimise the statistical properties of the rank-ordering model.

The primary desideratum for a bad event is that it should be terminal; once an account experiences it, there should be a high probability of rolling over to worse statuses with little chance of curing. This is typically assessed using roll rate or transition matrix analysis, which examines the probability of moving between different delinquency statuses over various time horizons (e.g. 1, 6, 12 months).

2.2.3 Prediction Horizon and Censoring

The prediction horizon (or time window) should be long enough to capture the majority of bad events and ensure the bad rate stabilises over vintages [2]. This is analysed using cumulative and marginal bad rate curves over the product's lifetime; see Fig. 3 for an illustration. Accounts that have been observed for less than the full prediction horizon are considered "censored" and should be excluded from model development to avoid introducing severe biases.

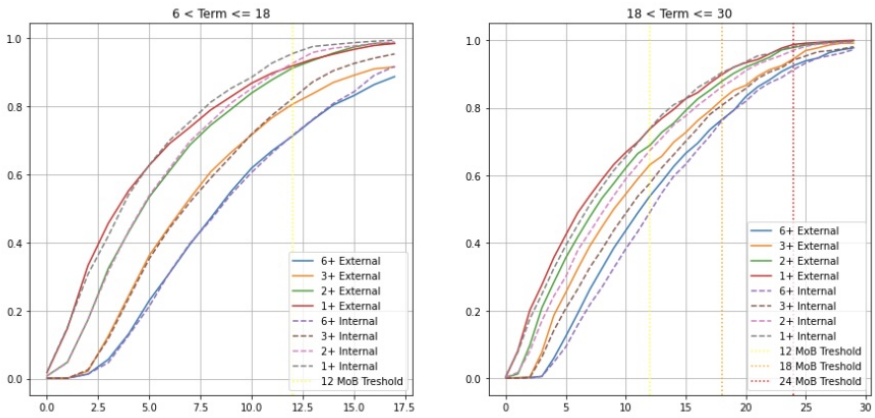

Figure 3: Cumulative Default Rate Curves. Shows the proportion of bads (y-axis) recorded up to the given month-on-book (x-axis) for various bad definitions (legend).

2.2.4 Candidate Covariate Generation with DFS

Deep Feature Synthesis (DFS) is an automated feature engineering technique that generates features from relational data by stacking simple mathematical operations (e.g., sum, average, etc.) across paths in a set of relational entities [11].
The input for DFS is a collection of interconnected entities and their corresponding data tables. Features are generated at two levels: the entity level and the relational level. Entity features (**efeat**) are computed for each entry within a single entity table, often involving element-wise operations or transformations of existing features, such as converting a categorical string to a numeric value or extracting components from a timestamp. Relational features are derived by analyzing two related entities. These include "direct features" (**dfeat**), which transfer features directly from a related entity via forward relationships, and "relational features" (**rfeat**), which apply mathematical functions (e.g., min, max, sum, count) over backward relationships to aggregate values from related entities. The feature generation process is recursive, where **rfeat** and **dfeat** features for a target entity are synthesized using features from its backward and forward related entities, respectively. After these are generated, **efeat** features can then be created using the original and newly synthesized features. This recursive process terminates when a predetermined depth is reached or when there are no further related entities.
An example illustrating this recursive generation is provided in Fig. 4. The illustration shows how a feature counting the number of employers a customer applying for credit has could be derived automatically. The DFS algorithm manages the order of computation, ensuring that the necessary features from related entities are generated before they are used to create features for the target entity.

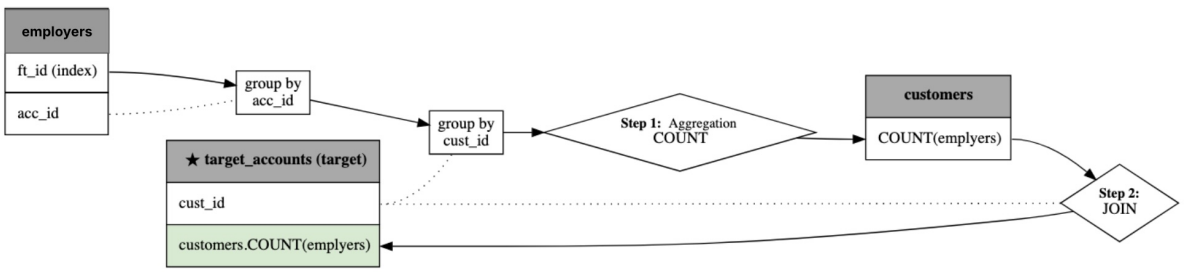

Figure 4: An example of a feature calculation using DFS

By traversing the relationships between entities, DFS enables scalable domain-agnostic fea-

ture generation, often producing richer representations than manual feature engineering.

2.2.5 Variable Preprocessing: Coarse Binning

To improve variable stability and handle non-linear relationships in models such as logistic regression, it is recommended to bin continuous variables into a smaller number of categories (i.e. coarse binning) [2, 9]. A decision-tree-based approach is used to split the numeric part of a feature, minimising Gini impurity within each bin while ensuring each bin contains a minimum percentage (e.g., 5%) of observations. Subsequently, bins with similar bad rates or insufficient observations are merged based on statistical tests (e.g., bootstrapped Z-test, Chi-square test, etc.) or qualitative criteria such as subcategory comparison tests. The latter involves assessing that the rank-ordering based on the bad rate across bins is the same in each subcategory of choice. If two contiguous bins present a reversal in trend, they are merged.

2.2.6 Variable Transformation: Weight of Evidence (WoE)

For binned or categorical variables, a Weight of Evidence (WoE) transformation is recommended over one-hot encoding. WoE replaces each bin with a single numeric value that represents the log-odds ratio of bad to good outcomes within that bin [2, 9].
This transformation linearises the relationship between the feature and the log-odds of the target, making it suitable for logistic regression. It also reduces the number of coefficients to estimate compared to one-hot encoding, leading to more stable and interpretable models. Combined with binning, WoE transformation allows model developers to capture non-linear relationships in the feature.

$$\text{WoE}(X = a) = \log\left(\frac{P(X = a|\text{Bad})}{P(X = a|\text{Good})}\right) \text{[2, 9, 5]}$$

where:

- $P(X = a|\text{Bad})$ is the fraction of bads in bin a
- $P(X = a|\text{Good})$ is the fraction of goods in bin a

The sign and value of WoE can be interpreted as follows:

- Positive WoE indicates that the ratio of bads to goods is greater than 1. Namely, that the proportion of bads is higher than the proportion of goods in that category, which indicates higher than average risk,
- Negative WoE indicates that the ratio of bads to goods is less than 1. Namely, that the proportion of bads is lower than the proportion of goods in that category, which indicates lower than average risk
- WoE near zero indicates that the ratio of bads to goods is close to 1. This indicates that the category presents little to no power to distinguish between bad and good customers.

2.3 Feature Selection

The sections below outline the automated approach used in feature selection. Firstly, reject inferencing and referencing is reviewed. Then, the concept of marginal information value (MIV) is introduced, building on top of the concepts introduced earlier in the paper.

2.3.1 Reject Inference and Referencing

In cases where a model is built on an "accept" population, it may not generalise well to "rejects" (applicants who were not granted credit). To address this, reject referencing or reject inference may be used [4, 8]. Reject referencing utilises external data (e.g., from credit bureaus) to obtain performance outcomes for rejected applicants who obtained credit elsewhere. Reject inference uses statistical techniques to infer the outcomes for the reject population. Reject referencing is preferred when available, as it provides a more accurate view of the risk profile of the reject population [8]. It is critical to seek and identify if either reject inferencing or referencing can be done, and the feature selection process should be performed on the population incorporating rejected application information whenever possible.

2.3.2 Marginal Information Value (MIV)

Information value (IV) is a measure based on the Kullback-Leibler divergence that quantifies the predictive power of a variable [2, 5].

$$\mathrm{IV}(X) = \sum_{a \in X} \left[(P(X = a|\mathrm{Bad}) - P(X = a|\mathrm{Good})) \times \mathrm{WoE}_{\mathrm{observed}}(X = a) \right] [2, 5]$$

The higher the IV the stronger the predictive power of the variable (in a univariate sense). Different ranges of IV can be interpreted as follows [2]:

- Less than or equal to 0.02: poor predictive power
- From 0.02 to 0.1: weak predictive power
- From 0.1 to 0.3: medium predictive power
- From 0.3 to 0.5: strong predictive power
- Greater than 0.5: very strong predictive power

The MIV process begins by selecting the feature with the highest individual Information Value (IV).
In subsequent steps, the feature with the highest MIV is added. MIV assesses the additional information that a new feature provides, given the information already captured by the features in the current model iteration.

$$\mathrm{MIV}(X) = \sum_{a \in X} \Bigg[(P(X = a|\mathrm{Bad}) - P(X = a|\mathrm{Good})) \times (\mathrm{WoE}_{\mathrm{observed}}(X = a) - \mathrm{WoE}_{\mathrm{expected}}(X = a)) \Bigg] [5]$$

where $\mathrm{WoE}_{\mathrm{expected}}$ is calculated based on the current model's scores.
The process continues until model performance on a test set plateaus or the highest MIV falls below a set threshold (e.g. 2%). A pairwise Pearson correlation threshold (e.g. 40%-60%) is also applied to ensure low multicollinearity among the final set of features.
Even if the process is fully automated, the model developer will review the outcome of the process to ensure the conceptual soundness of the variables selected, including their meaning and trends. Moreover, the model developer will further scrutinise the variables to confirm that they are not proxies for protected characteristics.
This "human-in-the-loop" approach ensures that variables selected make business sense and there are no concerns from a legal and compliance perspective to go ahead with the next stages of model development.

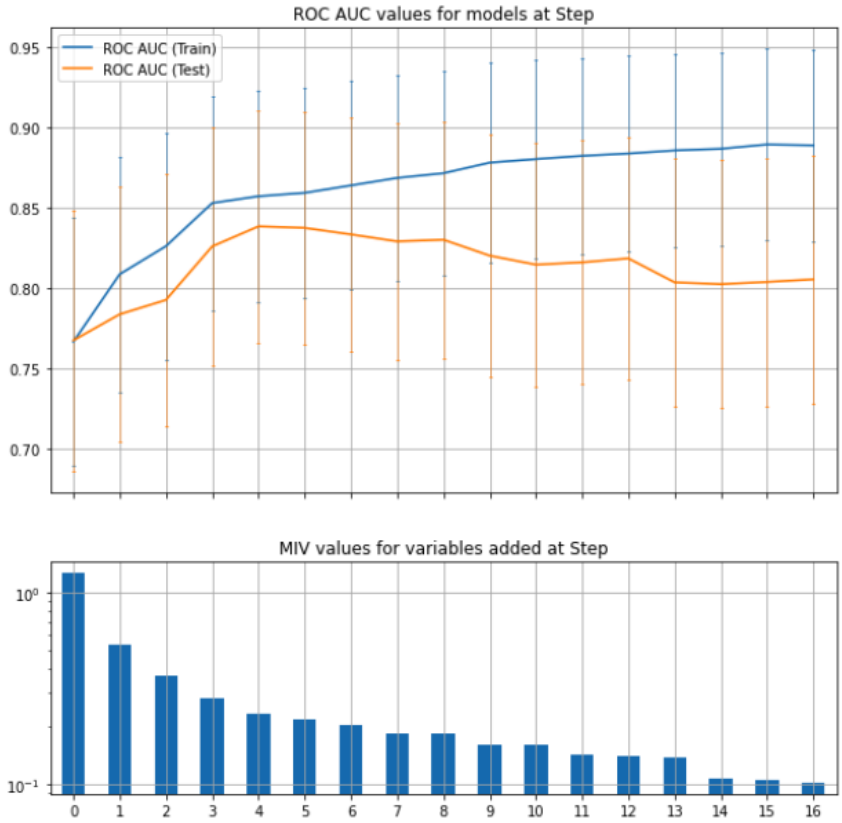

Figure 5: Illustration of MIV stepwise feature selection process.

2.4 Model Training and Benchmarking

While the focus of the paper are the feature generation and selection steps, it is worthwhile to provide remarks on the model training and benchmarking to showcase the performance of the chosen generation and selection techniques. The section starts by overviewing different model performance calculation techniques, and concludes by showcasing internal benchmarking results.

2.4.1 Performance Evaluation

The model performance evaluation is split into the evaluation of the rank-ordering and PD accuracy performance. Each is reviewed in turn in the paragraphs that follow.

Rank-ordering The rank-order scorecard performance is evaluated based on its ability to rank-order risk. The primary metric for this is the Gini coefficient, which is derived from the Lorenz curve [3]. The Lorenz curve is illustrated in Fig. 6.
A good credit scoring model is expected to have a Gini value of at least 50%, although this can vary depending on the risk spectrum of the population. Performance must be stable across time on development, test, and out-of-time samples.

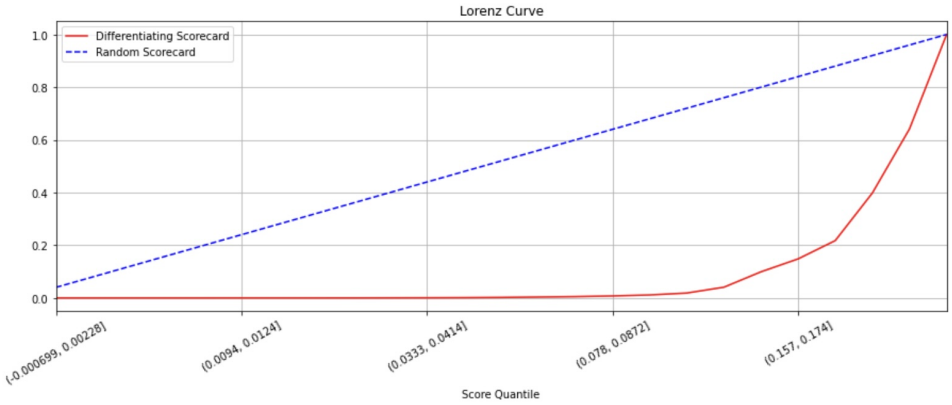

Figure 6: An illustration of the Lorenz Curve. The x-axis shows the quantiles of the rank-order score, and the y-axis shows the fraction of bads falling below the given cut-off.

Calibration A separate calibration step is used to transform the rank-ordering scores into accurate PD estimates. Techniques such as "Platt Scaling" (which uses logistic regression) [10, 12] or isotonic regression are employed [12]. The choice depends on the data volume and the characteristics of the score distribution, with isotonic regression being preferred for larger datasets, as it can capture non-monotonic relationships without requiring input transformations.
The calibration quality is generally assessed using score calibration plots [12], an illustration of which is available in Fig. 7, and the Brier score [2].

Brier Score The Brier score for binary classification problems can be computed as

$$\text{Brier Score} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{p}_i)^2 [13]$$

where:

- y_i is the observed outcome, which takes up a value of 1 if the event occurred and 0 otherwise
- $\hat{p}_t$ is the probability of occurrence predicted by the model, with values ranging between 0 and 1

The Brier Score measures the mean squared difference between the actual outcome and the predicted probability: the smaller the value the better.

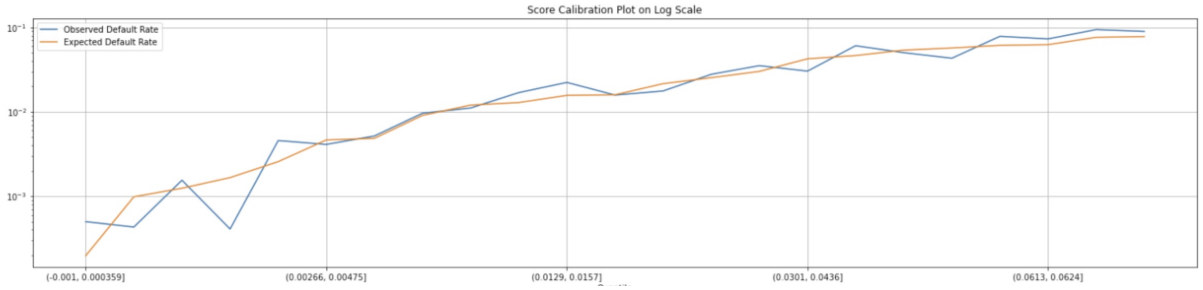

Figure 7: An illustration of the calibration curve. The x-axis shows the quantile of the calibrated score, and the y-axis shows the default rate.

2.4.2 Model Choice and Benchmarking

The MIV feature selection technique is theoretically related to logistic regression, and benchmarking shows that this combination frequently performs in line with, or better than, more complex models like tree ensembles. For example, while building acquisition models for personal loan and credit card portfolios, logistic regression models achieved a higher or equal Gini coefficient compared to tree ensembles[1].

Table 1: Benchmarking of Tree Ensemble vs. Logistic Regression **Gini**

Model Name	Tree Ensemble	Logistic Regression	Final Model Chosen
Personal Loans A	54%	58%	Logistic Regression
Credit Cards A	63%	63%	Logistic Regression
Personal Loans B	64%	65%	Logistic Regression
Credit Cards B	70%	70%	Logistic Regression
Alternative Data Model	55%	43%	Tree Ensemble

While internal benchmarking shows that more advanced deep learning approaches such as recurrent neural networks outperform both tree ensemble and logistic regression approaches, the WoE-based logistic regression models are able to, in general albeit with some exceptions, perform on par with standard non-linear models such as tree ensembles (gradient boosted trees, random forests) thanks to the WoE transformation. As mentioned in 2.2.6, WoE transformation linearises variable relationships with the target in the log-odds space and captures non-linearities at the variable level.

3 Automated Value-Adding Feature Monitoring

The combination of DFS and MIV permits to establish an automated risk-splitting, value-adding feature monitoring along the lines of the famed, but highly manual, residual monitoring (ReMo) approach employed in the financial industry [14].

3.1 Background and Architecture

To ensure that models remain predictive and to identify new valuable features, an automated feature monitoring system has been created. This system continuously checks whether any of the features generated via DFS add marginal information value to the incumbent acquisition PD scorecards.

The system architecture relies on several key components: an orchestrator which takes care of computing new features using DFS on internal data; Revolut internal scoring engine; Revolut internal centralised data lakehouse to store model scores and target events (delinquency outcomes); and cloud storage to store the outputs of the automated process. As a final step, the orchestrator calculates the MIV for the full set of generated features against a selected delinquency target (e.g., 14 days past due at 3 months on book). Results are refreshed on a regular basis for all portfolios.

An example output of the process is illustrated in Fig. 9. It shows that higher travel-related spending in the current account of the user leads to statistically significantly lower delinquency rates when looking at 14 days past due at 3 months on book. The x-axis shows the coarse risk grades customers are allocated to based on the incumbent model. The y-axis shows the observed early (14 days past due in the first 3 months-on-book) delinquency rate on average in each grade, and for the groups having the variable values in the ranges as per the legend.

[1]Names have been anonymised to protect business sensitive information

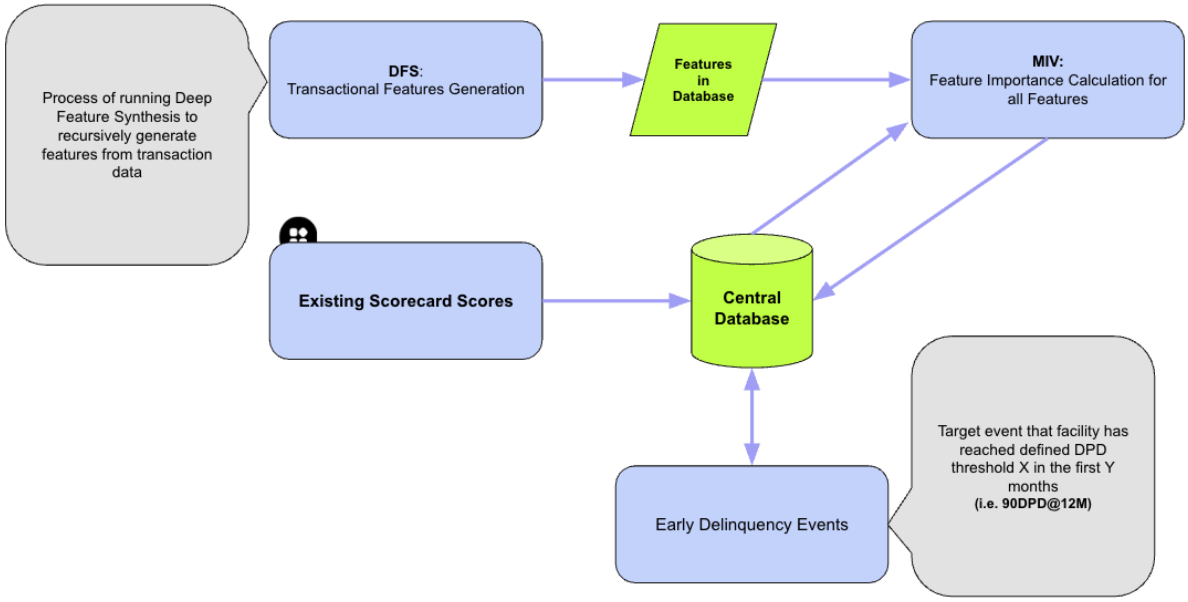

Figure 8: Schematic of the Automated Value-Adding Feature Monitoring system

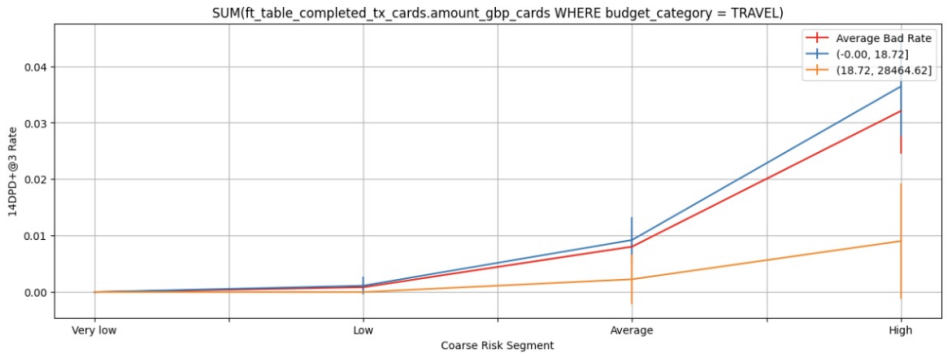

Figure 9: An example of residual monitoring outcomes. The x-axis shows the coarse risk grades customers are allocated to based on the incumbent model. The y-axis shows the observed early (14 days past due in the first 3 months-on-book) delinquency rate.

3.2 Results and Limitations

The current system is a minimum viable product and has limitations. It only uses disbursed facilities for its target (excluding rejected applications), considers only some of our internal data, and does not yet incorporate reject referencing. Future improvements include onboarding more data sources and incorporating analysis on the rejected population.

4 Case Studies and Business Benefits

The section illustrates two case studies of how the combination of MIV and DFS led to better business outcomes in terms of approval rate, sales and profitability. The section will explain how, while the initial investment in building the automated DFS and MIV pipeline was significant, the long-term reduction in manual data science hours for new market entry and model refreshes, coupled with the performance gains, has resulted in a strong positive return on investment.

4.1 Case Study: Personal Loans and Positive Selection

By utilising our own internal data, especially in markets where Revolut is a dominant player in the banking industry, or alternative data sources not commonly used in credit risk management, we can identify user segments that are lower risk but under-served in the market. This allows for more competitive pricing for these desirable customers, leading to an effect called "positive selection," where better users choose the Revolut's credit offers over competitors'.

When a new acquisition model incorporating transactional data was rolled out for a personal unsecured personal loans portfolio, it resulted in an approximately 30% increase in sales volume. This was driven primarily by a higher offer take-up rate rather than an increased approval rate, and was achieved while simultaneously reducing the portfolio's delinquency rates. Analysis on disbursals indicates that loan amounts increased by +22.71% for customers who would have been accepted by the old model, and by +9.75% for those who would have been rejected, leading to an overall increase of +24%. Further sales uplift was realised after relaxing credit policy rules that were no longer deemed required given a stronger acquisition PD model, increasing the approval rate, and bringing the total effect of the model roll out to approximately 30%.

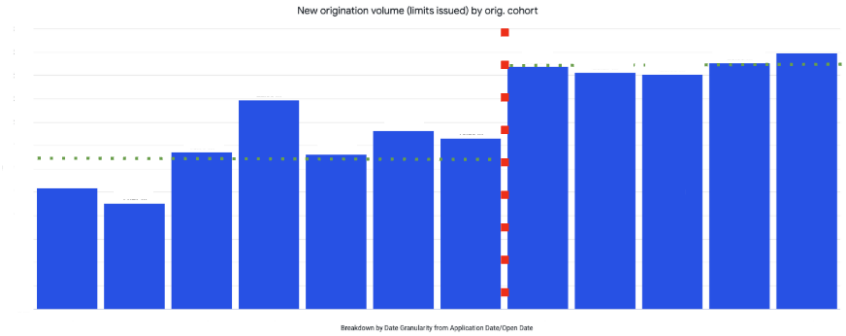

Figure 10: New origination volume for personal loans portfolio.
The red dotted line indicates when the new model went live.

The model was first rolled out as part of an A/B testing for a period of 3 months. During A/B testing, customers were randomly assigned to be scored by either the new or the incumbent model. The average increase in loan amount and overall disbursed amount observed during A/B testing were consistent in magnitude with what was observed after the end of the A/B testing. The increase in sales while maintaining profitability and risk appetite in one portfolio showcased above is sufficient to justify the investment in the development of the MIV and DFS system on its own, but comparable commercial improvements can be seen across multiple portfolios.

4.2 Case Study: Credit Cards and RAROC

Identifying features that add predictive power to an acquisition PD model is identical to identifying user segments where we under- or over-predicted risk, as showcased in the previous sections. Therefore, identifying new data sources and predictive features allows to more accurately allocate a broader user base to risk grades, increasing Risk-Adjusted Return on Capital (RAROC) of the credit originations. In other words, a more accurate risk assessment allows for more appropriate risk-based pricing, which increases profitability. By more accurately allocating customers to risk grades, the RAROC of credit originations increases. Data from a credit card portfolio demonstrates this increase in RAROC after the implementation of a new model utilising more internal transaction and application usage data.

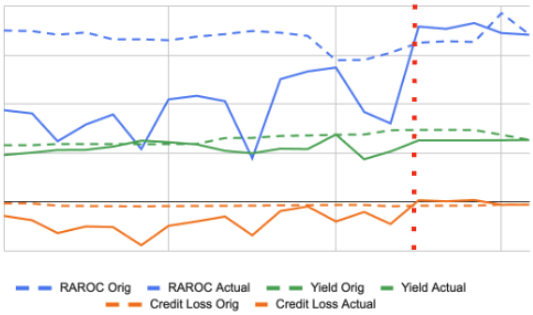

Figure 11: Impact on RAROC on a credit card portfolio.
The red dotted line indicates when the new model went live.

5 Conclusion and Future Directions

The presented methodology provides a robust, scalable, and automated framework for the development and monitoring of acquisition PD models. By combining Deep Feature Synthesis with a disciplined, MIV-based feature selection process, Revolut develops highly predictive models that drive tangible business value through positive selection and enhanced profitability. DFS enables a more comprehensive use of internal and external data sources, and while usage of non-bureau data for credit origination is widespread in Latin America and Asia, the current entrants still have a first mover advantage in many European countries. Arguably, alternative or more enriched, while still relevant and appropriate, data is key to a successful IRB modelling strategy, too [24].

Future work will focus on expanding the data sources used, incorporating bureau and more mobile application usage data, and implementing reject referencing to account for rejected applications where it was not yet done. This will provide a more complete view of the applicant pool and further enhance model accuracy and fairness. Additionally, research into leveraging Large Language Models (LLMs) and embeddings for more nuanced feature generation is planned.

Acknowledgements

The authors thank Natalia Semina and Francisco Antonio Teixeira Mendonca on the work done in developing new acquisition models using Deep Feature Synthesis and Mark Fukson and his team (Marcin Maslany, Manvydas Kriauciunas) in developing MIV-based automated feature monitoring. The work presented in this paper has been contributed to by multiple members of the Revolut organisation and we apologise if we have inadvertently missed some.

References

[1] James Kanter and Kalyan Veeramachaneni (2015) *Deep Feature Synthesis: Towards Automating Data Science Endeavors*, IEEE International Conference on Data Science and Advanced Analytics (DSAA).

[2] Naeem Siddiqi (2017) *Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards*, John Wiley & Sons, 2nd ed.

[3] Mamdouh Refaat (2012) *Credit Risk Scorecards: Development and Implementation Using SAS*

[4] David J. Hand and William E. Henley (1997) *Statistical classification methods in consumer credit scoring: a review*, Journal of the Royal Statistical Society: Series A (Statistics in Society), Vol. 160, No. 3, pp. 523-541.

[5] Gerard Scallan (2011), *Class(ic) Scorecard, Selecting Characteristics and Attributes in Logistic Regression*, Edinburgh Credit Scoring Conference

[6] David Opitz and Richard Maclin (1999) *Popular ensemble methods: An empirical study*, Journal of artificial intelligence research, Vol. 11, pp. 169-198.

[7] Erik Strumbelj and Igor Kononenko (2014) *Explaining prediction models and individual predictions with feature contributions*, Knowledge and information systems, Vol. 41, pp. 647-665.

[8] John Banasik and Jonathan Crook (2007) *Reject inference, augmentation, and sample selection*, Journal of the Operational Research Society, Vol. 58, No. 7, pp. 842-851.

[9] Lyn C. Thomas (2009) *Consumer credit models: pricing, profit and portfolios*, OUP Oxford.

[10] Platt John (1999) *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*, Advances in Large Margin Classifiers, 10 (3), pp. 61-74

[11] Alteryx (2025), Available at: `https://featuretools.alteryx.com/en/stable/index.html`

[12] Nehrebecka, Natalia. *Probability-of-default curve calibration and validation of internal rating systems*. Vol. 43. Bank for International Settlements, 2017.

[13] Glenn W. Brier (1950) *Verification of Forecasts Expressed in Terms of Probability*, Monthly Weather Review 78 (1), 1-3

[14] Clemons, Eric K., and Matt E. Thatcher. "Capital One: Exploiting an information-based strategy." *Proceedings of the Thirty-first Hawaii International Conference on System Sciences*. Vol. 6. IEEE, 1998.

[15] Guo, Hong, Lindsay B. Jack, and Asoke K. Nandi. "Feature generation using genetic programming with application to fault classification." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 35.1 (2005): 89-99.

[16] Tackett, Walter Alden. "Genetic Programming for Feature Discovery and Image Discrimination." ICGA. 1993.

[17] Chandra, Dharani. "Applications of Large Language Model Reasoning in Feature Generation." arXiv preprint arXiv:2503.11989 (2025).

[18] Gong, Nanxu, et al. "Evolutionary large language model for automated feature transformation." *Proceedings of the AAAI conference on artificial intelligence.* Vol. 39. No. 16. 2025.

[19] Wang, Jianxun, and Yixiang Chen. "A review on code generation with LLMs: Application and evaluation." 2023 *IEEE International Conference on Medical Artificial Intelligence (MedAI)*. IEEE, 2023.

[20] Visalakshi, S., and V. Radha. "A literature review of feature selection techniques and applications: Review of feature selection in data mining." *2014 IEEE international conference on computational intelligence and computing research.* IEEE, 2014.

[21] Tibshirani, Robert. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996): 267-288.

[22] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).

[23] Strumbelj, Erik, and Igor Kononenko. "An efficient explanation of individual classifications using game theory." *The Journal of Machine Learning Research* 11 (2010): 1-18.

[24] Hurlin, Christophe, and Christophe Pérignon. "Machine Learning and IRB Capital Requirements: Advantages, Risks, and Recommendations." *Risks, and Recommendations* (December 5, 2023) (2023).