UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Application of One Class Models for Financial Risk Classification

Author:
Ana REY DAVILA

Supervisor:
Oriol PUJOL VILA

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2025

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc in Fundamental Principles of Data Science

Application of One Class Models for Financial Risk Classification

by Ana REY DAVILA

This project explores the use of One Class Classification methods to predict credit risk in highly imbalanced financial datasets. Unlike traditional supervised models, OCC approaches focus only on the majority class, in this case, customers with good payment behaviour, and aim to detect unusual patterns that might suggest a higher risk of default.

The study is divided into three experimental phases. The first phase uses a limited set of 13 variables, selected and categorised by experts based on risk. The second phase removes this expert selection and uses all available features. In the third phase, a hybrid strategy is tested by adding the anomaly scores generated by OCC models as extra input variables to supervised models.

The models are evaluated using ROC AUC and PR AUC, two metrics well suited for imbalanced classification problems. The main goal is to analyse whether anomaly detection techniques can support or improve current risk assessment strategies in a real business setting.

However, the results did not confirm the initial hypothesis, as One Class models and hybrid approaches did not outperform traditional supervised methods.

Acknowledgements

I would like to express my sincere gratitude to all the people who have supported me throughout this project.

In particular, I want to thank Dr Oriol Pujol for his continuous academic guidance. His help during moments of doubt and his trust in my work have been key to moving forward.

I would also like to thank Pol, for his professional support, his openness to discussing ideas, and for always being helpful and available whenever I had questions or needed guidance.

Their support, in different but complementary areas, has been essential to completing this project.

Finally, I want to thank my family and friends, who have always stood by me throughout this journey. Your encouragement during the toughest moments and your belief in me meant more than words can say.

Contents

Chapter 1

Introduction

1.1 Context and Motivation

In recent years, due to the latest financial crises, the financial sector has become more controlled by strict regulatory requirements and transparency standards. For this reason, it is very important to build strong models that can assess credit risk in an effective and reliable way.

In this situation, building scoring models to evaluate credit risk has become a particularly difficult task. Identifying high risk customers correctly is crucial for avoiding financial losses and ensuring efficent business management. Historically, financial instituions have relied on scoring models based on logistic regression due to their statistical simplicity, interpretability and high degree of explainability

However, logistic regression and other traditional supervised models have significant limitations when dealing with highly imbalanced datasets (Abd Rahman and Ong, 2020), which are common in credit risk management. Typically, most customers have good payment behaviour, while only a small minority defaulta, and, due to this imbalance, traditional models tend to focus more on the majority class. This causes the minority class, which is more important in the context of credit risk, to be left in the background. As a result, it becomes difficult to correctly identify customers with a high risk of default. From a technical point of view, these models often overfit the majority class and do not generalise well to rare cases.

This limitation highlights the critical need to explore alternative strategies capable of effectively handling highly imbalanced data. One Class Classification models, that can be used for anomaly detection, emerge as a promising alternative for credit risk assessment in such circumstances. Unlike traditional supervised methods, these models are trained exclusively on data representing normal or majority class behaviour, that is, customers who consistently fulfill their payment obligations. The main goal of One Class models is to accurately detect anomalies or significant deviations that could indicate a higher risk of default.

An additional advantage of One Class models is that they do not require labelled examples of defaulters. This is especially important in financial datasets where the number of defaults is very small or the labels are not always reliable.

Within these alternative approaches, methods like One Class Support Vector Machines (Schölkopf et al., 1999), Isolation Forest (Liu, Ting, and Zhou, 2009) or Aproximate Polytope Ensemble (Casale, Pujol, and Radeva, 2014) are considered strong options, as they focus solely on patterns representing good payers. This strategy could help institutions detect potentially problematic customers more accurately and improve early detection of credit risk.

In summary, evaluating the potential of One Class models not only addresses the methodological need for effectively dealing with imbalanced datasets but also offers a strategic opportunity for improving risk management practices in financial institutions.

1.2 Objectives of the Study

The main objective of this project is to evaluate how effective One Class Classification models are for predicting financial risk, especially in situations where the data is highly unbalanced.

To reach this goal, the study follows these specific steps:

- Compare the performance of different One Class models (One Class SVM, Isolation Forest, and Approximate Polytope Ensemble) with traditional supervised models such as logistic regression and linear SVM.
- Test the models under two different settings: one using a reduced set of categorised variables selected by experts, and another using the full set of variables without any manual selection or categorisation.
- Assess the models using evaluation metrics that are more appropriate for unbalanced data, including the ROC curve and the Precision–Recall curve.
- Analyse whether the anomaly scores produced by One Class models can be used as additional features to improve the performance of supervised models.

1.3 Preview of Results

The initial hypothesis of this study was that One Class models, by focusing exclusively on good customers, could be more effective in detecting defaulters in highly imbalanced financial data. The idea was that these models would learn the typical behaviour of reliable payers and identify risk by spotting unusual cases, without needing labelled examples of default.

However, after testing different strategies across the experimental phases, the results showed that traditional supervised models, such as logistic regression and linear SVM, still performed better in this specific use case. Even when using a richer feature set or combining anomaly scores with supervised models, One Class approaches did not consistently outperform the baseline.

These findings suggest that, despite the theoretical appeal of treating defaulters as anomalies, One Class models may face important limitations in real credit scoring environments. Factors such as the nature of the input variables, the difficulty of modelling complex customer behaviour, or the lack of clear structural differences between good and bad payers may reduce their effectiveness in practice.

However, the study also highlighted that combining supervised and unsupervised signals can be a promising direction. While the results did not improve in this project, hybrid approaches remain an area worth exploring further.

1.4 Structure of the Thesis

This thesis is structured in a way that follows the natural development of the project, from the initial motivation to the final conclusions. After this introduction, the next part provides the theoretical foundations needed to understand the models and techniques used throughout the work. This includes an explanation of what anomaly detection is, the main types of anomalies, and the selected One Class models to be applied in this context.

Once the theoretical background is established, the thesis moves on to describe the specific use case that motivates the project. This part explains the real business situation in which the models are applied, the limitations of the current approach, and the reasons for considering a different strategy based on anomaly detection. The proposed modelling strategy is introduced here.

The next section focuses on the methodology used to carry out the study. It includes a detailed description of the dataset, the preprocessing steps applied to the variables, and the experimental design.

After the methodological part, the results of the experiments are presented and analysed. This includes performance metrics for each model, comparisons between supervised and One Class approaches, and observations on how the models behave in different settings.

Finally, the thesis ends with a conclusion that summarises the main contributions of the work. It also reflects on the limitations of the current approach and suggests ideas for future research or practical improvements.

Chapter 2

Theoretical Foundations

This chapter presents the key theoretical foundations that support the project. First, it explains what anomalies are and the different types that exist. Then, it discusses how the availability of labels influences the design of the models and describes the main strategies used to detect anomalies. Finally, it introduces the three One Class models evaluated in this study.

2.1 Anomalies in Data

In most datasets, the majority of entries follow common patterns. However, some observations behave very differently from the rest. These are known as anomalies, they are samples that appear to deviate markedly from other members of the group in which they occur (Grubbs, 1969).

Anomalies can be grouped into three main types, depending on how they appear and what makes them different (Chandola, Banerjee, and Kumar, 2009):

- **Point anomalies:** A single data point is clearly different from the rest of the data. For example, in a person's credit card history, a very large transaction that does not match their usual spending may be a point anomaly.
- **Contextual anomalies:** A data point is only unusual within a specific context. For instance, a temperature of 10°C might be normal during the winter but unusual during the summer. Time, in this case, provides the context.
- **Collective anomalies:** A group of values is anomalous when considered together, even if individual values are normal. For example, a specific sequence of actions on a computer may indicate an attack, even if each action alone seems normal.

In this project, we focus on point anomalies, which occur when individual data points strongly differ from typical behaviour. These are especially relevant in situations where rare but significant cases must be identified within a large volume of normal data.

Detecting such anomalies is essential because they often signal events of high importance, such as fraud, technical malfunctions, or unexpected user behaviour. Because of this, anomaly detection plays a key role in areas like finance, cybersecurity, healthcare, and industrial monitoring (Matthew, Jude, and James, 2025).

2.1.1 Supervised, Semi-Supervised, and Unsupervised Detection

The way anomaly detection models are trained depends on the availability of labelled data. In many real world problems, getting reliable and representative labels is expensive and difficult. Because of this, different methods have been developed depending on how much label information is available.

Anomaly detection methods can be grouped into three main categories, depending on how labels are used (Chandola, Banerjee, and Kumar, 2009):

- **Supervised anomaly detection:** These techniques assume that the training data includes labels for both normal and anomalous instances. The goal is to learn a classification model that can separate the two. However, in practice, this setup is limited by two major challenges: strong class imbalance, as anomalies are far fewer than normal examples, and difficulty in obtaining enough representative examples of all possible anomalies.
- **Semi-supervised anomaly detection:** In this setting, the model is trained using only examples of normal behaviour. The goal is to learn the structure of the normal class and then flag any new instance that deviates significantly in the test set.
- **Unsupervised anomaly detection:** These methods do not need labelled data. They assume that anomalies are rare and clearly different from normal cases. The model tries to learn general patterns in the data and separates any samples that do not fit. This approach is flexible but can lead to many false positives if the assumption about rarity is not correct.

2.2 One Class Classification

One Class Classification (OCC) is a modelling strategy that fits naturally within the semi-supervised framework of anomaly detection. These models are trained using only data from the normal class and aim to learn its underlying structure. The core idea is to identify the region of the feature space that best describes normal behaviour and then flag any observation that lies outside this region as a potential anomaly. This approach is particularly useful when anomalous instances are too scarce, poorly defined, or highly diverse to be reliably included in the training process (Perera, Oza, and Patel, 2021).

The following sections describe the specific One Class models evaluated in this project: One Class Support Vector Machine, Isolation Forest, and Approximate Polytope Ensemble.

2.2.1 One Class Support Vector Machine (OCSVM) (Schölkopf et al., 1999)

One Class SVM is a method designed to detect unusual observations by learning the structure of normal data. It is based on the idea of separating the normal class from the origin in a high-dimensional space, using kernel functions.

The algorithm maps the input data $x_i \in \mathbb{R}^n$ into a feature space $\mathcal{F}$ using a kernel function $\phi(x)$. Then it tries to find a hyperplane that maximises the distance from the origin and contains most of the mapped data. This is done by solving the following optimisation problem:

$$\min_{\mathbf{w},\rho,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \tag{2.1}$$

subject to:

$$\mathbf{w} \cdot \phi(x_i) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n \tag{2.2}$$

Here:

- $\nu \in (0, 1]$ is a parameter that controls the trade-off between the fraction of outliers and support vectors.
- ξ_i are slack variables that allow some violations.
- ρ is the offset defining the threshold.

The decision function is defined as:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i K(x_i, x) - \rho\right) \tag{2.3}$$

where $K(x_i, x)$ is the kernel function (often RBF), and the α_i are obtained by solving the dual problem.

This model works well when the normal data is compact and has a regular structure. However, it also has some limitations:

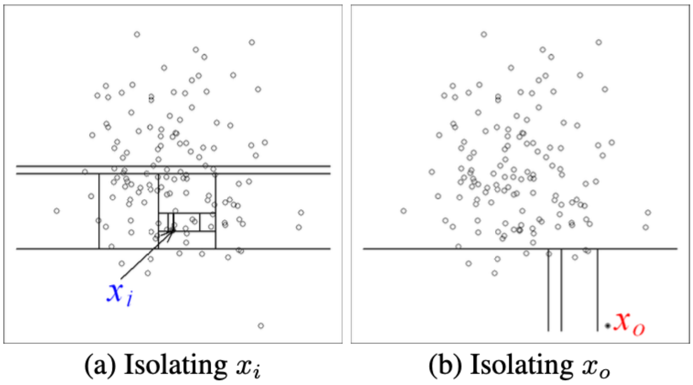
- It is sensitive to the choice of kernel and hyperparameters.
- It may not perform well in high dimensional or noisy data.
- The decision boundary may not be flexible enough if the data distribution is complex.

Even with these drawbacks, One Class SVM is a solid method for anomaly detection.

2.2.2 Isolation Forest (Liu, Ting, and Zhou, 2009)

Isolation Forest is an algorithm for anomaly detection that works by isolating observations instead of profiling normal behaviour. Its key idea is that anomalies are easier to isolate because they are few and different.

Unlike other models that build a profile of normal data, iForest creates many random trees (iTrees). Each tree is built by randomly selecting a feature and then a split value, continuing this process until every instance is isolated. In general, anomalies are separated faster and require fewer splits, so they appear closer to the root of the tree. Figure 2.1 illustrates this intuition: the normal point x_i (left) requires more random cuts to be isolated compared to the anomaly x_o (right), which can be separated with fewer partitions due to its sparse location.



(a) Isolating x_i (b) Isolating x_o

FIGURE 2.1: Isolation process in iForest (Liu, Ting, and Zhou, 2009).

To detect anomalies, iForest measures the average path length $E(h(x))$ of each instance across all trees. The path length is the number of splits required to isolate the instance. Then it computes an anomaly score $s(x, n)$, defined as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2.4}$$

where $c(n)$ is the average path length of unsuccessful search in a Binary Search Tree and is approximated as:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \quad \text{with} \quad H(i) \approx \ln(i) + 0.5772 \tag{2.5}$$

Interpretation of the score:

- If $s(x) \approx 1$, x is very likely an anomaly
- If $s(x) \approx 0.5$, x is similar to normal instances
- If $s(x) \ll 0.5$, x is very likely a normal point

The method is efficient and scalable:

- It uses random sub-samples of the data
- It has linear time complexity
- It does not require distance or density computations

2.2.3 Approximate Polytope Ensemble (APE) (Casale, Pujol, and Radeva, 2014)

The Approximate Polytope Ensemble (APE) is a One Class classification method that uses geometrical ideas to define the boundary of the normal class. It is based on the concept of the convex hull, which is the smallest shape that contains all the data points. In APE, this shape is used to represent the normal class, and any point outside of it is considered an anomaly.

However, computing the convex hull in high-dimensional data is very expensive. To solve this, APE uses random projections: the data is projected many times into low dimensional spaces (1D or 2D), where convex hulls are easy to compute. A point is considered an outlier if it falls outside the expanded convex hull in at least one of the projections, as we can see in Figure 2.2.

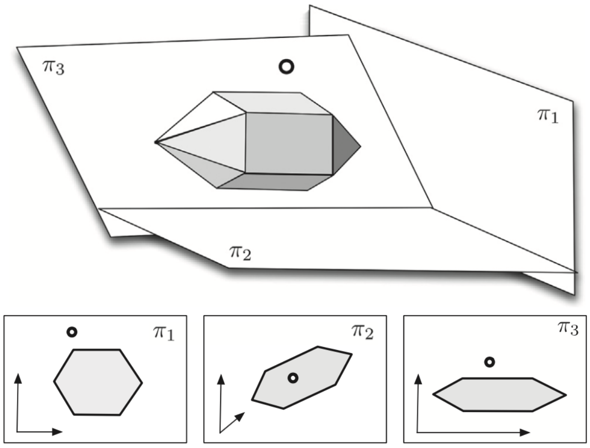


FIGURE 2.2: Example of how APE projects data into low dimensional spaces to build convex hulls (Casale, Pujol, and Radeva, 2014).

To avoid overfitting and allow more flexibility, APE does not use only the original convex hull. Instead, it builds expanded or shrunken versions of the hull, controlled by a parameter α. For example, if $\alpha > 0$, the polytope is enlarged; if $\alpha < 0$, it is shrunk. Each vertex of the extended convex polytope is computed as:

$$v_\alpha = v + \alpha \cdot \frac{v - c}{\|v - c\|} \tag{2.6}$$

where v is a vertex of the convex hull and c is the centre of the data points. This structure helps tune the model to be more or less strict when detecting outliers.

An alternative way to introduce flexibility into the method is through subset sampling. Instead of projecting the full dataset, APE selects small random subsets of the data for each projection. This approach is based on the assumption that the majority class tends to be densely concentrated around a core region, while anomalies are more scattered. By using different subsets in each projection, the method increases its chances of capturing the core distribution of the normal class while reducing sensitivity to outliers. This not only improves computational efficiency and also enhances the robustness of the hulls constructed in each projection.

Chapter 3

Use Case and Proposed Approach

Identifying customers who are likely to default is a difficult task in credit scoring, especially when the data is very unbalanced. This chapter explains the real world case that motivated this project, focusing on the weaknesses of the current models and why new strategies are needed. It also presents the proposed solution, which uses anomaly detection methods to identify unusual behaviours related to credit risk.

3.1 Business Context

The use case presented in this project comes from a financial company that offers credit solutions for purchasing products through a e-commerce platform. Specifically, the company provides short term financing that allows customers to split the payment into four instalments over a 90 day period. Since the financed items are not typically high value products, the system must deliver fast and accurate credit decisions to ensure a efficient purchasing process for the customer.

To perform this, the company currently uses a scoring model based on logistic regression. It was developed using expert knowledge and works with a set of 13 variables. These variables include information about the applicant's profile and the amount requested among others. Each variable was grouped into risk based categories, and a Weight of Evidence (WoE) encoding was applied to make them suitable for modelling. This encoding technique will be explained in more detail later in the Methodology Chapter 4.

Still, the model has some limitations that can affect its overall performance. The next section discusses these challenges in more detail.

3.2 Limitations of Logistic Regression in Imbalanced Data

Logistic regression is one of the most common models used in credit scoring. It is easy to interpret, efficient to train, and suitable for production systems. The model estimates the probability that a customer will repay the credit by learning a linear relationship between the input features and the log odds of the positive class.

The formula of logistic regression is:

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{3.1}$$

where $x_1, \ldots, x_n$ are the input features and $\beta_0, \ldots, \beta_n$ are the model coefficients. The output P represents the estimated probability that the target variable given the input features (Peng, Lee, and Ingersoll, 2002).

While logistic regression works well in many situations, it has some important limitations when the data is highly imbalanced (Abd Rahman and Ong, 2020). In credit scoring, only a small fraction of the customers are defaulters, which means the model sees very few examples of the risky class. This can lead to several problems:

- The model focuses too much on predicting the majority class correctly, since most observations are from that group.
- It may learn decision boundaries that are too simple to detect the rare patterns of defaulters.
- The probabilities produced by the model can be misleading, especially for customers close to the decision boundary.

In practice, this means that the model can achieve high accuracy while still failing to detect many defaulters. This is dangerous in credit scoring, where identifying risky customers is the main goal.

To deal with these problems, other solutions like resampling, changing the threshold, or using alternative metrics (such as Precision Recall AUC) are often used. However, these adjustments do not always solve the core issue: the lack of information about the minority class during training.

Because of this, different modelling approaches that do not rely on having many examples of both classes were explored. One of these alternatives is anomaly detection, and its motivation is introduced in the next section.

3.3 Motivation for Using Anomaly Detection

In credit scoring, defaulters represent a very small fraction of the population, but they carry a high financial impact. This creates a common problem in machine learning: imbalanced data, where the minority class is both rare and highly important. Missing these cases can lead to losses, while detecting them early improves the company's ability to manage risk.

Anomaly detection provides a different way to approach this challenge. Instead of building a model that learns to separate defaulters from non defaulters, it focuses on understanding the structure of normal behaviour. The goal is to detect unusual patterns that may indicate risk, even if those patterns have not been seen before.

This perspective is useful in credit risk scenarios for several reasons. First, the number of defaulters is very small, which limits the available training data for the risky class. Second, customer behaviour may change over time, making expert designed rules or past labels less reliable. In these situations, detecting deviations from the norm can provide a more flexible and adaptive solution.

By treating defaults as anomalies, One Class models can detect unusual cases that may go unnoticed in traditional systems. This makes them a valuable complement to supervised approaches.

3.4 Modelling Strategy and Experimental Design

Based on the motivation explained in the previous section, this project follows a different strategy to model credit risk by using anomaly detection methods. Instead of using only labelled examples of defaulters, the main idea is to learn the usual behaviour of good payers and find cases that are different from this pattern. This type of modelling works well in situations where there is a strong class imbalance, and where risky cases are both rare and hard to label clearly.

To explore this idea, the evaluation was structured into three experimental phases.

3.4.1 Purpose of the Multi Phase Structure

The multi phase structure was developed to evaluate the trade offs between expert-driven modelling and data driven strategies. In Phase 1, the goal was to test whether anomaly detection models could perform competitively when using the same input variables as the company's current production model. Phase 2 removed expert selection and used the full set of available features, allowing the models to uncover patterns without prior assumptions. In Phase 3, we introduced a hybrid strategy, using the anomaly scores produced by One Class models as additional features for supervised classifiers.

This progressive setup from replication of expert defined strategies to purely data driven learning and hybrid integration were not just used to compare performance, but also to understand how different types of information, such as expert knowledge, raw features, and anomaly scores, can help in predicting credit risk.

3.4.2 Comparative Modelling Objectives

Each experimental phase was guided by a specific modelling question:

- Can One Class classifiers for anomaly detection achieve better results than logistic regression when using the same expert selected variables?
- Does removing expert selection and using the full set of features allow unsupervised and supervised models to detect riskier patterns more effectively?
- Can the anomaly scores generated by One Class classifiers help improve the performance of supervised models when used as additional features?

These comparisons aimed not only to identify which models perform better, but also to understand the added value of expert-driven variables, the potential of raw feature patterns, and the benefits of combining both. Ultimately, they helped clarify when expert inputs are necessary and when purely data driven information offer an advantage.

Chapter 4

Methodology

This chapter presents the complete methodology followed in the project, from data preparation to model training and evaluation. The process was designed to ensure fair and consistent comparisons between modelling strategies, including both supervised classifiers and One Class anomaly detectors. In the following sections, each step of the experimental workflow is described in detail, including the structure of the phases, data preprocessing, feature engineering, and the evaluation criteria applied across all models.

4.1 Structure of the Experimental Process

In the first stage, we prepared the raw dataset by selecting the population of interest and defining the target variable. The population included credit applications that were initially accepted and not defunded during the financing process. This ensured that the models were trained and evaluated only on cases where the customer had completed the credit cycle and a final outcome was available. The original class imbalance between good and bad customers was preserved to reflect the real world distribution of credit outcomes.

In the second stage, we applied specific feature transformations based on the type of variable and the experimental phase. To ensure consistency across all models, these transformations were integrated into a unified preprocessing pipeline.

The third stage consisted of training both supervised and One Class models. These models were evaluated following the three experimental phases introduced in Section 3.4:

- Using expert defined variables.
- Using the full set of features.
- Combining both approaches by using anomaly scores as additional inputs for supervised models

Finally, the models were evaluated using ROC AUC and PR AUC scores. Cross validation was applied during training to ensure robust results, and all final models were tested on a separate test set

This workflow made sure that all models were tested under the same conditions, so the results reflect real challenges in credit scoring.

4.2 Data Preparation

The dataset used in this study contains 503,429 credit applications submitted between January 2021 and February 2022. Each row corresponds to a financing request that was approved. Applications that were rejected are not included, and the same customer may appear more than once if they submitted multiple approved requests during this period.

The dataset includes a variety of features related to each application. These features cover four main categories: personal information about the customer, device related data from which the request was submitted, geographical indicators, and financial details such as the requested amount and the resulting monthly instalment.

To define the study population, we applied several filters during the data extraction process. Specifically, we removed all records flagged as fraud and all contracts that were defunded, that is, cases where the financed purchase was cancelled before the end of the 90 day period. These filters were applied early to reduce dataset size and optimise later processing, due to constraints on computation and storage.

The target variable is binary and was already present in the internal databases used for the current production model in the company. A customer is labelled as BAD if they entered legal debt collection procedures, which occurs when their Bad Debt Rate (BDR) exceeds 25% within six months. The BDR at time t is defined as:

$$BDR(t) = \frac{\text{Cumulative unpaid amount by time } t}{\text{Cumulative expected repayments by time } t} \tag{4.1}$$

This ratio indicates the proportion of expected repayments up to month t that remain unpaid. It provides a measure of how much of the scheduled debt has not been recovered at a given point in the credit lifecycle. If the BDR at six months is greater than 0.25, the customer is considered to have defaulted. Otherwise, the record is labelled as GOOD.

The overall class distribution is highly imbalanced: approximately 98% of the records are labelled as GOOD and only 2% as BAD. This imbalance is the main motivation to explore alternative approaches to traditional binary classification methods.

The train-test split was also defined within the company's internal setup, using a 70/30 ratio. This split was stratified to maintain the original proportion of GOOD and BAD cases and remained unchanged for all experiments in this project. All preprocessing steps were applied after the split.

4.3 Description of the Experimental Phases

The three phase structure introduced in Section 3.4, was implemented to assess the effectiveness of different modelling strategies in detecting credit risk under imbalanced conditions. Each phase followed a distinct configuration in terms of input variables and model composition. This section describes the technical implementation of each phase.

- **Phase 1** aims to replicate the modelling setup currently used by the company, which relies on 13 features categorised according to their relationship with credit risk and transformed using Weight of Evidence (WoE) encoding. This phase allows us to assess whether One Class models can perform competitively using the same inputs.

- **Phase 2** removes expert selection and uses the full set of available features. This gives the models access to a richer and more detailed view of the data.

- **Phase 3** explores a hybrid strategy, where the anomaly scores produced by the One Class models are used as new features for supervised models. This phase aims to combine the strengths of both approaches.

4.4 Feature Engineering

The feature engineering process differed across the experimental phases, depending on the nature and preparation level of the input variables.

In Phase 1, no additional feature selection was performed, as the goal was to replicate the methodology currently used in the company's production model. The dataset included 13 variables categorised according to credit risk, originally selected by domain experts. Although the company's model applies Weight of Evidence (WoE) encoding to these variables, the available dataset contained only the original categorical values. For this reason, we applied the WoE transformation ourselves, reproducing the same approach used in the production system.

WoE is a technique used to transform categorical variables into numerical values based on their relationship with the target variable. It is particularly useful in credit scoring (Seitshiro and and, 2024), as it provides a way to quantify the predictive power of each category in terms of default risk. The WoE value for a category c is computed as:

$$WoE(c) = \ln\left(\frac{\text{Proportion of GOOD in } c}{\text{Proportion of BAD in } c}\right) \tag{4.2}$$

A higher WoE indicates a stronger association with good repayment behaviour, while a lower WoE suggests higher risk.

In Phases 2 and 3, we worked with the complete set of available features, which included a mix of binary, categorical, numerical and date type variables. To ensure consistency across experiments, we implemented a preprocessing pipeline using `ColumnTransformer` from `scikit-Learn`. This pipeline applied appropriate transformations to each group of variables:

- Binary variables were converted into 0/1 format.
- Categorical variables were handled according to their cardinality:
  - High-cardinality features, such as country of birth, nationality, or postal code, were encoded using target encoding. To avoid data leakage, this transformation was fitted only on the training set.
  - Low-cardinality features were transformed using One-Hot Encoding to create binary indicator variables.
- Date variables, including hour, minute, second, month, and day of the month, were encoded using sine and cosine transformations. This approach tries to capture cyclical patterns, such as seasonality or behavioural trends over time.
- Numerical features were standardised using `StandardScaler` to normalise their range and improve model training stability.

4.5 Evaluation Protocol

The goal of the evaluation is to compare how well each modelling strategy can identify risky customers, under the constraint of strong class imbalance. For this reason, choosing the right metrics is very important. For example, if we used metrics like accuracy, a model that simply classified all customers as good would achieve 98% accuracy, which might look like a great result, but in reality, it would not be detecting any risky customers.

The problem is that using an inappropriate metric can hide the real performance of the model in the cases that matter most, which are few but very important. For this reason, instead of focusing on how often the model is correct in general, it is more appropriate in our context to use metrics that evaluate how well the model can separate the two classes, especially when detecting correctly the minority class can have a strong impact on the business.

To assess model performance, we used two metrics that are well suited for imbalanced classification tasks (Holzmann and Klar, 2024):

- **ROC AUC (Area Under the Receiver Operating Characteristic Curve)** measures the trade off between true positive rate (TPR) and false positive rate (FPR) at different classification thresholds. One of the advantages of the ROC curve is that it is independent of the class imbalance ratio. This makes it a stable reference for evaluating how well the model separates the two classes across the full range of possible decision thresholds.

  However, in highly imbalanced problems, ROC AUC can be too optimistic. When the number of negative examples is much higher, a low FPR may still correspond to many false positives in absolute terms. For this reason, we also included a second metric that focuses more directly on the minority class.

- **PR AUC (Area Under the Precision-Recall Curve)** plots precision (the fraction of true positives among all positive predictions) against recall (TPR). Unlike ROC AUC, the PR curve is sensitive to the proportion of positive cases (π), and therefore gives a more realistic picture in scenarios where positive cases are rare. As shown in the literature, when π is very small, as in our case with only 2% defaulters, precision becomes more difficult to maintain, especially when the model aims for high recall. This trade-off is clearly reflected in the PR curve.

Using both ROC AUC and PR AUC, we obtain a more complete view of how the models behave. Additionally, these curves allow us to explore different operating points. This is important in a real business context, where the company may need to choose between reducing false positives or maximizing detection of defaulters. ROC and PR curves make it possible to adjust the decision threshold based on the acceptable balance between risk and business opportunity.

These metrics were applied consistently across all models and experimental phases. For supervised models, the scores were based on predicted probabilities for the BAD class. For One Class models, the anomaly scores were used directly without converting them to binary predictions.

4.6 Model Configuration

In previous sections, we introduced the different experimental phases and outlined the general training and evaluation methodology. This section focuses on the individual configuration of each model used in the experiments.

With the exception of the APE model, which was adapted and implemented based on the method described in (Casale, Pujol, and Radeva, 2014), all other models were built using standard Python libraries, such as `scikit-learn`. Each model was configured to avoid computational issues due to the large dataset size and, to ensure fair comparison, we maintained consistent data splits, prepocessing pipelines and hyperparameters across all phases.

As detailed in Section 4.5, all models were retrained on the full training set after cross validation and then tested on the same test set using the same metrics.

4.6.1 Parameter Settings and Assumptions

Both the One Class SVM and Isolation Forest models include a parameter to set the expected level or the limit of anomalies in the data. In our case, it was set to 0.02, matching the known proportion of defaulters in the dataset. This approach follows a semi-supervised methodology, as the models are trained only with examples of normal behaviour. This setup is especially useful if we assume that the proportion of defaults stays stable over time.

4.6.2 Logistic Regression

Logistic regression was selected as one of the supervised baselines, since it is the model currently used in the company's production scoring system. The goal here was to replicate it as closely as possible to provide a fair baseline for comparison with the alternative approaches explored in this study.

The model was implemented using the `LogisticRegression` class from the `scikit-learn` library. To handle the class imbalance, we set the `class_weight` parameter to `balanced`, and increased the `max_iter` value to 10,000 to ensure proper convergence on the large dataset.

4.6.3 Linear Support Vector Machine (SVM)

The Linear SVM model was implemented using the `LinearSVC` class from `scikit-learn`. This algorithm was selected for its computational efficiency and performance on high dimensional data, making it suitable for our. The model configuration included setting `class_weight= balanced` to address the class imbalance, `max_iter=10000` to ensure convergence, and `dual=False` for improved performance when the number of features exceeds the number of samples. A fixed `random_state` was used to ensure reproducibility.

Since `LinearSVC` does not produce probability estimates by default, we added a calibration step to make its outputs compatible with our evaluation metrics. For this, we used `CalibratedClassifierCV` with the `sigmoid` method, which transforms the raw decision scores into calibrated probabilities. This step transforms raw decision scores into probability estimates required for consistent metric evaluation across all models. Calibration was applied during training using 5-fold cross validation on the validation data, following the same logic as described in Section 4.5.

4.6.4 One Class Support Vector Machine (OCSVM)

The One Class SVM was implemented using the `OneClassSVM` class from `scikit-learn`. As explained in Section 2.2.1, this model belongs to the family of unsupervised anomaly detection methods and was trained exclusively on GOOD customers to learn the profile of normal behaviour.

The model was configured with:

- `gamma = auto`, which sets the kernel coefficient based on the number of input features

- `nu = 0.02`, a value aligned with the known proportion of defaulters in the dataset. This parameter acts as an upper bound on the expected fraction of anomalies.

During training, the model was fitted using only GOOD samples from each fold. Anomaly scores were obtained by applying the negative of the `decision_function`, so that higher values indicate a greater likelihood of deviation from normal behaviour. These scores were used directly to evaluate model performance using ROC AUC and PR AUC.

4.6.5 Isolation Forest

Isolation Forest, as introduced in Section 2.2.2, is an algorithm that isolates anomalies by building an ensemble of random trees that recursively partition the data. Samples that are easier to isolate, are considered more anomalous, or in our context, more likely to be defaulters.

The model was implemented using the `IsolationForest` class from `scikit-learn`, with the number of trees set to `n_estimators=100`. The `contamination` parameter was fixed at 0.02 to match the known share of defaulters in the dataset.

As with the One Class SVM, the model was trained only on GOOD customers. Anomaly scores were computed using the negative of the `decision_function`, so that higher scores correspond to higher deviation from normal behaviour. These scores were then evaluated using ROC AUC and PR AUC.

4.6.6 Aproximate Polytope Ensemble (APE)

The Approximate Polytope Ensemble (APE) is a One Class classifier based on geometric principles. The method was originally proposed for One Class Classification problems in Casale, Pujol, and Radeva, 2014, and the implementation used in this project was adapted to fit our high dimensional dataset.

The core idea of APE is to construct a convex hull that encloses the positive class, in our case the good customers, and to flag as anomalous any point outside this region. However, in dimensional spaces, computing convex hulls is computationally infeasible. To address this, APE performs repeated projections of the data into lower dimensional spaces, in this project we use two dimensions.

Each projection is performed using a normal random projection matrix $P \in \mathbb{R}^{d \times 2}$, where d is the number of original features. Its two columns are orthonormalised to ensure numerical stability. This normal random projection approach allows the method to explore multiple geometric perspectives of the data in a consistent and low dimensional space, making it computationally efficient while preserving relevant structure from the original high dimensional space.

Within each 2D subspace, a convex hull is built around the GOOD samples from a randomly selected subset of the training data. A data point is considered an inlier in that projection if its 2D projection falls within the convex hull.

Given the number of samples in our dataset, projecting all data points in each iteration would still be too costly. To solve this problem, each projection uses only a small random subset, we choose 1% of the training data. With many projections, this strategy ensures that the model progressively covers the entire dataset without needing to process all samples at once.

APE is configured with 100,000 projections. For each of them:

- A random 1% is sampled from the training data.

- The subset is projected into 2D using a normal random projection P.
- A convex hull is computed over the projected GOOD points.

For every data point, APE keeps track of how many times it has been evaluated (seen) and how many times it does not fall inside the convex hull (out of hull). The anomaly score is then computed as the ratio between the number of projections in which the point was not inside the hull and the total number of projections in which the point was included.

By default, APE adopts a strict criterion: a point is considered anomalous if it falls outside the convex hull in at least one projection. However, in this project we opted for a more flexible interpretation by using the ratio of times a point falls outside as a continuous anomaly score. This value estimates how unusual each point is, making it more suitable for evaluation using ROC AUC and PR AUC.

At test time, the score is computed using the full test set without subsampling, to ensure that all instances are evaluated consistently across all projections.

Chapter 5

Results

5.1 Introduction

The goal of this chapter is to present and analyse the results obtained from the different modelling strategies tested in this project. As described in previous chapters, the main objective was to compare how well various models can detect risky customers under strong class imbalance.

The results are organised according to the three experimental phases introduced and described earlier in Sections 3.4 and 4.3. To ensure a fair and consistent comparison, all models were evaluated using the same two metrics: ROC AUC and PR AUC. These metrics were selected for their ability to measure model performance in imbalanced classification tasks, as discussed in Section 4.5.

For each phase, the results include both numerical summaries and visualisations such as ROC and Precision-Recall curves, to help interpret the behaviour of each model in more detail.

5.2 Phase 1 – Expert Defined Features

In this first experimental phase, we replicated the modelling pipeline currently used in the company's credit scoring system. All models were trained using the same 13 categorized variables selected by domain experts, already transformed using Weight of Evidence (WoE) encoding, as described in Section 4.4.

This setup provides a solid and controlled baseline for comparison. It allows us to analyse whether One Class models can match or even improve the performance of traditional supervised approaches when given the same input variables.

Table 5.1 shows the evaluation results for all models. As expected, considering the nature of the input data, the supervised classifiers achieved the best performance. Logistic Regression and Linear SVM reached almost identical scores, and among the One Class models, Isolation Forest showed the highest results.

TABLE 5.1: Model performance in Phase 1 (expert defined features)

Model	ROC AUC	PR AUC
Logistic Regression	0.772	0.085
Linear SVM	0.771	0.085
One Class SVM	0.531	0.038
Isolation Forest	0.694	0.059
APE	0.660	0.056

These results reflect the advantages of supervised learning when both classes are available and the input features are designed to capture credit risk. The Weight of Evidence (WoE) encoding includes expert knowledge by comparing the frequency of good and bad payers in each category and generating scores that reflect this relationship. Supervised models like logistic regression and linear SVM can take advantage of this encoding because they learn from both classes and can understand which values represent higher or lower risk.

However, this type of encoding is not suitable for One Class models. These models are trained only with data from good customers and do not see any negative examples during training. As a result, they cannot interpret WoE values correctly, since these scores depend on a comparison with the bad class, which is not available to the model. Instead of representing natural patterns of normal behaviour, WoE encodes differences between classes, something that One Class models cannot use. This makes it harder for them to learn a useful representation of what a good customer looks like, and limits their ability to detect anomalies. For this reason, Phase 1 is a challenging environment for unsupervised methods, which explains their lower performance.

To better visualise the results, Figure 5.1 shows the performance curves for each model and highlights the notable differences between supervised and One Class approaches.

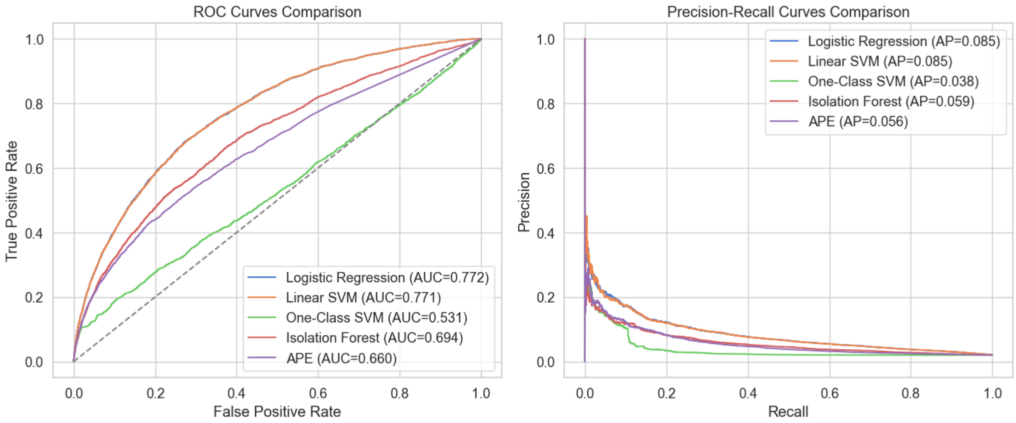


FIGURE 5.1: Comparison of ROC (left) and Precision-Recall (right) curves for all models in Phase 1.

5.3 Phase 2 – Full Feature Set

In this second experimental phase, all expert filters were removed and the models were trained using the complete set of available features. The main goal was to explore whether using a broader and unbiased feature space could improve performance, especially for the One Class models, which had struggled in Phase 1 due to the use of WoE encoding.

To prepare the data, we used the full preprocessing pipeline described in Section 4.4, applying transformations based on the type of variable: binary, categorical, numerical or temporal. This configuration was applied consistently across all models to ensure a fair comparison.

The evaluation results are shown in Table 5.2. As in the previous phase, the supervised models achieved the highest scores, with very similar results between logistic regression and linear SVM. The One Class models again showed lower performance, with no improvements compared to Phase 1.

TABLE 5.2: Model performance in Phase 2 (full feature set)

Model	ROC AUC	PR AUC
Logistic Regression	0.763	0.071
Linear SVM	0.765	0.073
One Class SVM	0.550	0.029
Isolation Forest	0.586	0.027
APE	0.564	0.026

Although models had access to more information in this phase and a wider range of features, supervised classifiers still performed better. This may be because they can learn more complex relationships between variables and the target. On the other hand, even in a more favourable setup than Phase 1, One Class models still found it difficult to build a stable profile of normal behaviour due to the high variability in the feature space.

Figure 5.2 shows the ROC and Precision-Recall curves for all models. The pattern is similar to Phase 1: supervised models show steeper curves, while One Class methods have flatter curves, reflecting a weaker ability to separate risky cases from the rest.

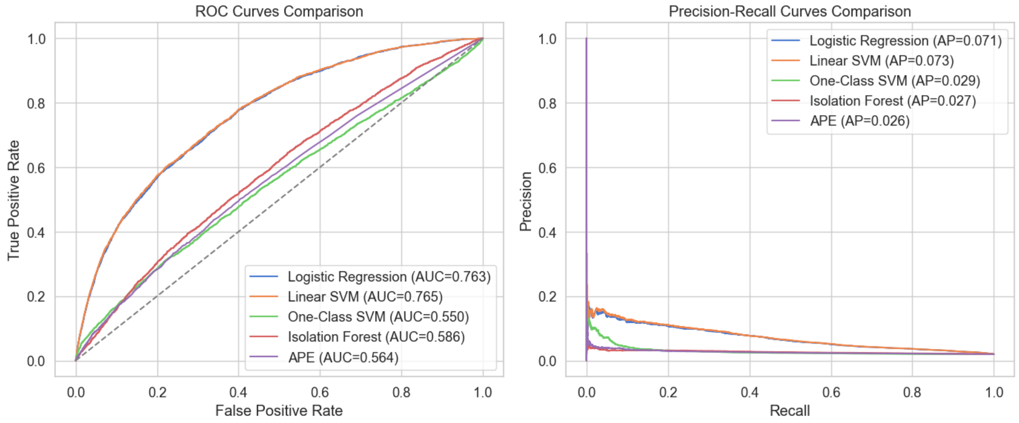


FIGURE 5.2: Comparison of ROC (left) and Precision-Recall (right) curves for all models in Phase 2.

5.4 Phase 3 - Hybrid Models

The third experimental phase explores whether the anomaly scores obtained from One Class models can enhance the performance of supervised models. Instead of replacing the existing modelling pipeline, this phase focuses on integrating anomaly detection outputs as additional features.

Two variants of the hybrid model were tested:

- Using the 13 expert-based features from Phase 1 along with the anomaly scores from OCSVM, Isolation Forest, and APE.
- Using the full feature set from Phase 2, combined with the anomaly scores computed from the same One Class models trained on the full input space.

In both cases, the anomaly scores were added as three new features to the input of Logistic Regression and Linear SVM models. The aim was to evaluate whether this extra information could help the supervised models improve their classification performance.

The results for both configurations are presented below.

5.4.1 Hybrid Models with Phase 1 Variables

On the test set, the hybrid models trained with the 13 risk-based variables and the anomaly scores achieved the following results:

TABLE 5.3: Performance of hybrid models in Phase 3 (13 expert-defined variables + anomaly scores)

Model	**ROC AUC**	**PR AUC**
Logistic Regression + anomaly scores	0.773	0.084
Linear SVM + anomaly scores	0.772	0.083

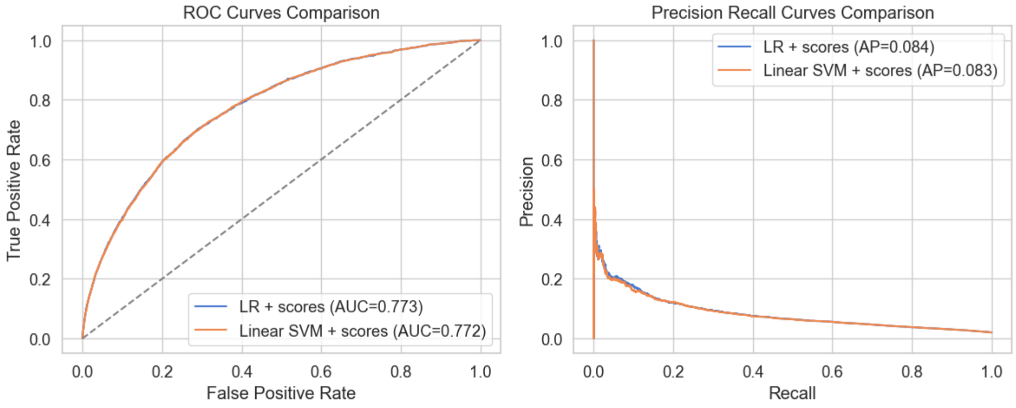


FIGURE 5.3: ROC and PR curves for hybrid models trained with Phase 1 variables and anomaly scores.

5.4.2 Hybrid Models with Full Feature Set

When trained with the complete set of input variables and the anomaly scores, the models produced slightly lower results:

TABLE 5.4: Performance of hybrid models in Phase 3 (full feature set + anomaly scores)

Model	**ROC AUC**	**PR AUC**
Logistic Regression + anomaly scores	0.762	0.071
Linear SVM + anomaly scores	0.765	0.073

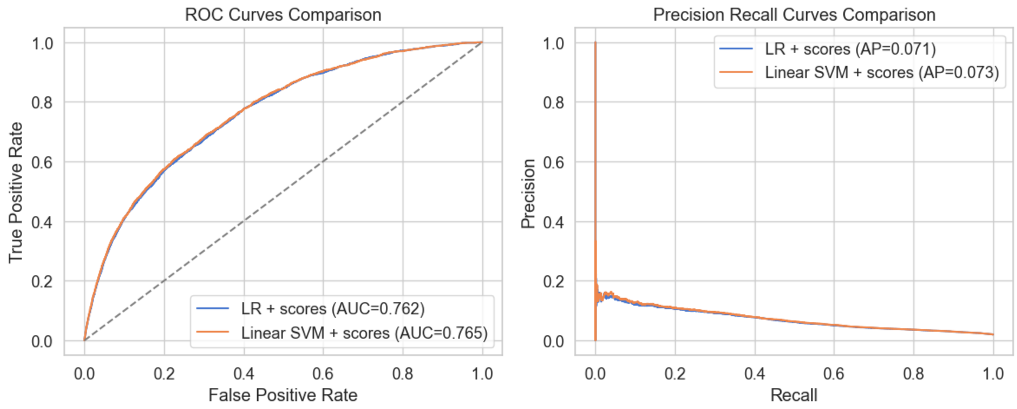


FIGURE 5.4: ROC and PR curves for hybrid models trained with full feature set and anomaly scores.

5.4.3 Final Remarks

These results suggest that, in this case, the anomaly scores did not provide enough additional information to improve the performance of the supervised models. Although the idea of combining both approaches is promising, its effectiveness strongly depends on the quality and usefulness of the scores generated by the One Class models.

Chapter 6

Conclusions

6.1 General Conclusions

This project explored different modelling strategies to predict credit risk in a highly imbalanced classification setting. The main goal was to evaluate whether anomaly detection methods, particularly One Class classifiers, could be a competitive alternative to traditional supervised models. To do this, we followed a three phase experimental design that gradually expanded the feature set and included hybrid model configurations.

The phased structure was one of the most valuable parts of the project, as it helped organise the analysis in a clear and progressive way. It also allowed us to adapt the following steps based on the results of each phase and explore the impact of different feature transformations on model performance.

One of the key lessons from this work was understanding how the design of the features affects different types of models. In Phase 1, we saw that transformations like Weight of Evidence (WoE), which are designed to reflect risk using the target variable, clearly benefit supervised models. These models can learn which values are linked to higher or lower risk. However, this kind of encoding is not useful for One Class models, which only see positive examples and do not have the context needed to correctly interpret WoE scores.

Even though One Class models did not outperform the supervised ones, their inclusion was necessary to test the initial idea that defaulters could be treated as anomalies. This hypothesis seemed promising at the beginning but was not supported by the results.

At the start of the project, One Class models were expected to perform well with imbalanced data, but the experiments showed that their performance was lower and that they are more difficult to use in real world systems. Therefore, we must reject the initial hypothesis and conclude that, at least for credit scoring, traditional models are still the most suitable choice.

This is especially important in the financial sector, where transparency and regulatory compliance are essential. In this context, models like logistic regression remain the preferred option due to their interpretability and stability.

Still, One Class models may have more potential in other areas such as fraud detection, where anomalies are more clearly separated from normal behaviour.

6.2 Limitations and Future Work

Although the results obtained in this study provide useful insights, there are several limitations that should be considered.

First, all One Class models were evaluated using feature representations that were originally designed for supervised approaches. Techniques such as Weight of Evidence or target encoding rely on information from the target variable, which may not be appropriate for unsupervised algorithms. As a result, these models may have been limited in their ability to learn useful patterns.

In addition, One Class models selected work directly on the data representation in a semi supervised way. Their performance strongly depends on how the observations are distributed in the feature space. If the data contains many irrelevant variables, noise, or a structure that does not help separate normal and abnormal behaviour, these models may fail to identify outliers correctly. In the case of Isolation Forest, the random partitions become less effective when the space includes many low-information dimensions. Similarly, One Class SVM tries to learn a boundary that surrounds the normal data. If the data is poorly scaled or noisy, this boundary may not represent the data well and will generalise poorly.

In the case of APE, although the method reduces dimensionality by projecting the data into random 2D spaces, its performance still depends on the quality of the original feature space. Since no dimensionality reduction was applied before the projections, many of them may include noise or non-informative combinations of features. This lowers the quality of the convex hulls used by the model and can affect its ability to detect anomalies.

Moreover, the accuracy of APE depends on the number of random projections performed. Increasing the number of projections would likely improve its reliability, but due to computational constraints, this was not feasible in the current study.

Therefore, exploring dimensionality reduction techniques, such as PCA, could be a promising way to improve model performance. These transformations help remove noise and concentrate relevant information in fewer dimensions, making it easier for detection models to work. This would be especially useful in Phases 2 and 3, where all available variables were used without any prior selection, which may have introduced more noise than useful information.

Another limitation is that no hyperparameter tuning was performed. All models used fixed values during all experimental phases to ensure consistency and fair comparison. While this decision helped keep the experimental design controlled, it may have prevented some models from reaching their best performance. A possible future step could be to apply systematic hyperparameter search strategies.

Another line of future work could involve the use of deep learning models to capture complex interactions among variables. Neural network based architectures may be better suited to discover latent structures and subtle patterns in high dimensional financial data. However, it is important to consider the constraints of transparency and explainability that apply in the financial sector, where model decisions must often be interpretable and auditable.

Finally, this study focused only on one modelling goal: predicting default risk. However, One Class models could be more suitable for other financial applications, such as fraud detection, where anomalies are more evident and structurally different. These types of tasks may offer a better environment for unsupervised approaches.

In summary, future research could extend this work by testing new data representations, adjusting model configurations, applying dimensionality reduction techniques, and exploring other use cases where anomaly detection could have greater impact.

Appendix A

Supplementary Figures

This appendix presents the ROC and Precision Recall (PR) curves of each model individually, across Phase 1 and Phase 2. These figures complement the comparative plots shown in the main body of the thesis and allow for a more detailed analysis of model behaviour.

A.1 Phase 1

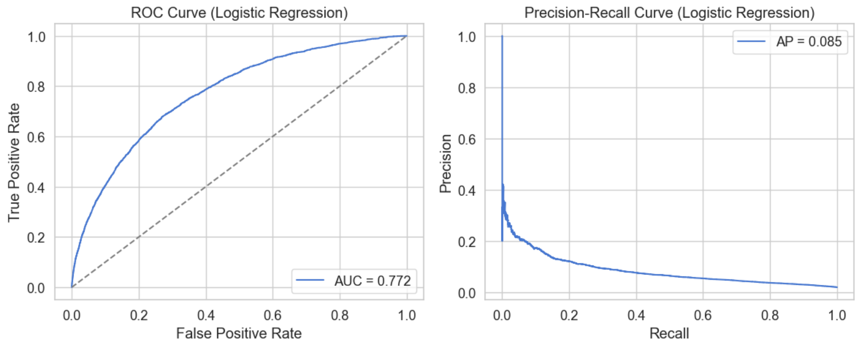


FIGURE A.1: Logistic Regression – Phase 1

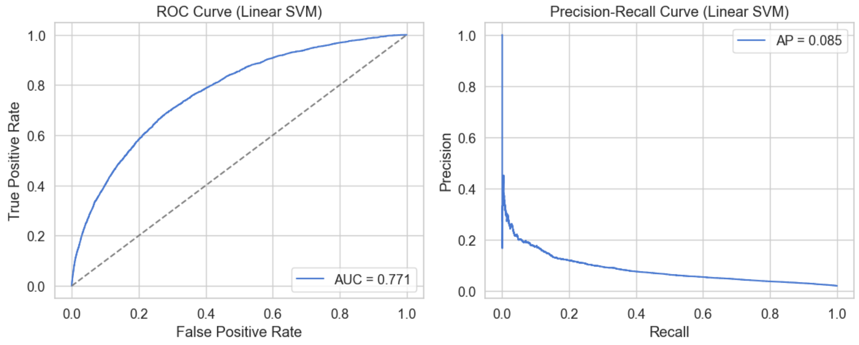


FIGURE A.2: Linear SVM – Phase 1

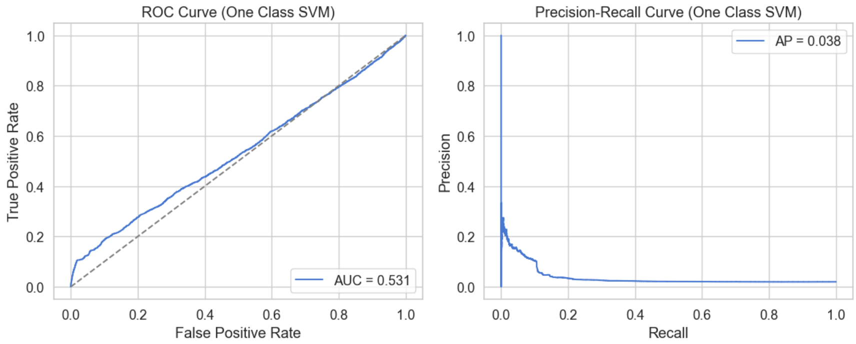


FIGURE A.3: One Class SVM – Phase 1

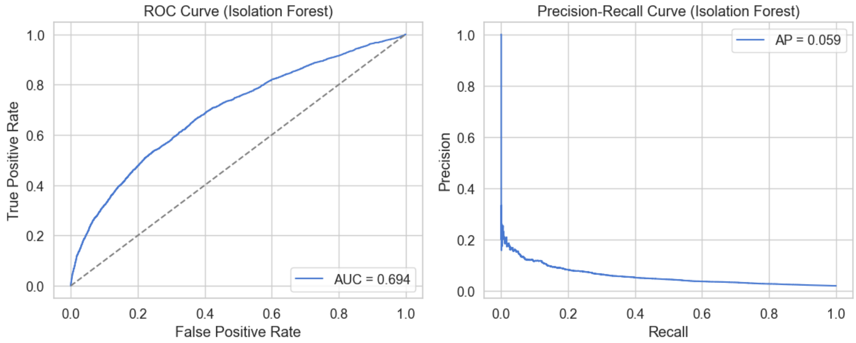


FIGURE A.4: Isolation Forest – Phase 1

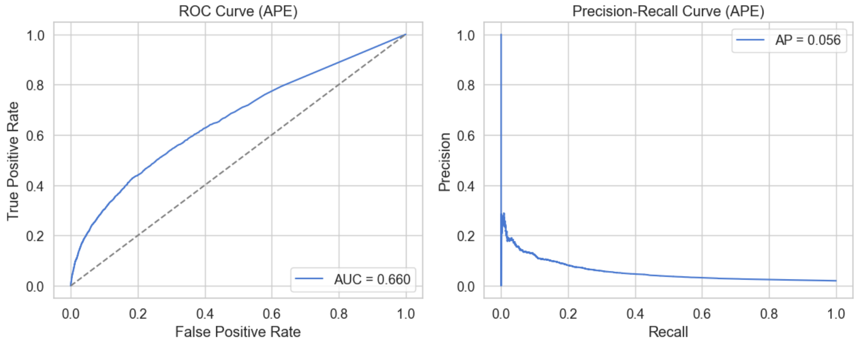


FIGURE A.5: Approximate Polytope Ensemble – Phase 1

A.2 Phase 2

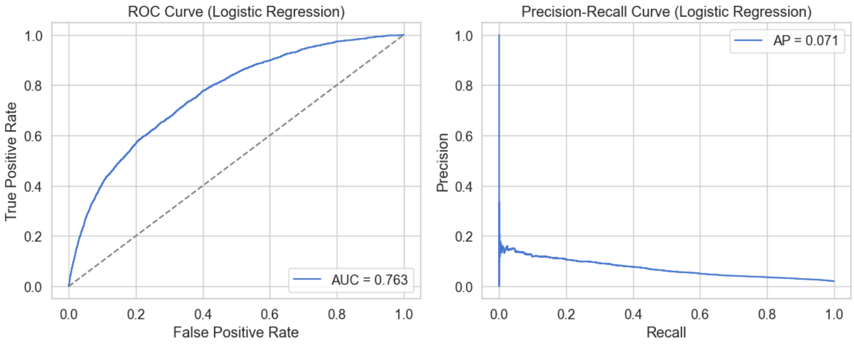


FIGURE A.6: Logistic Regression – Phase 2

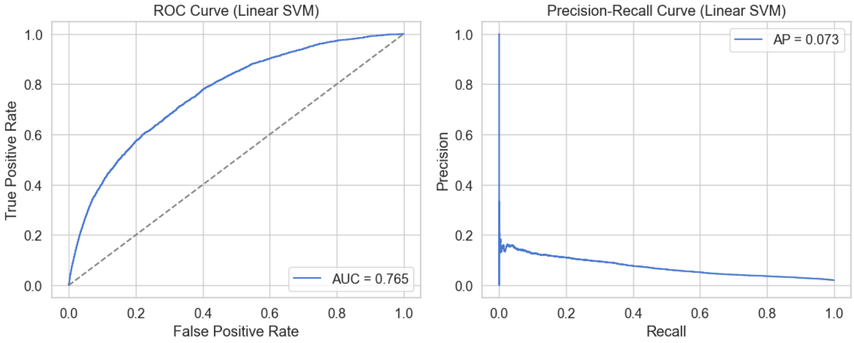


FIGURE A.7: Linear SVM – Phase 2

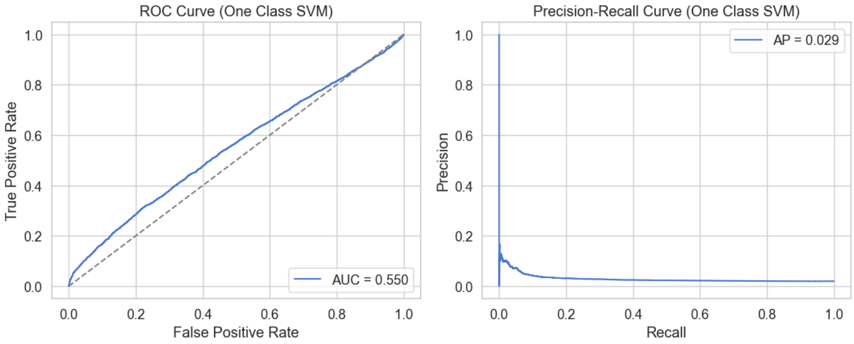


FIGURE A.8: One Class SVM – Phase 2

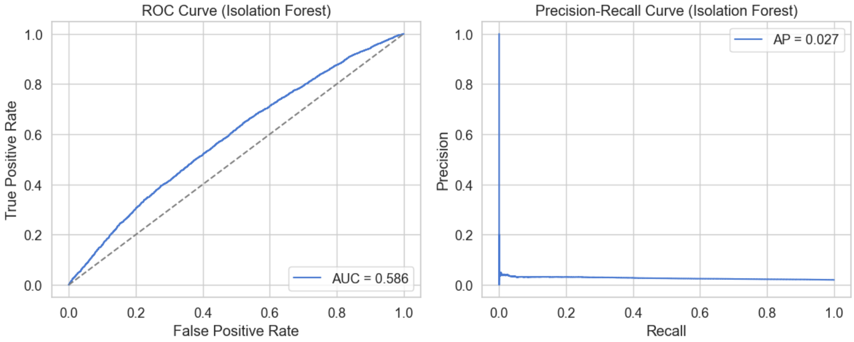


FIGURE A.9: Isolation Forest – Phase 2

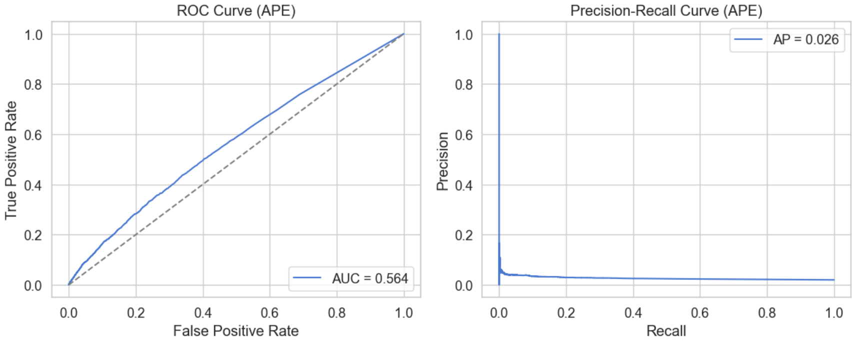


FIGURE A.10: Approximate Polytope Ensemble – Phase 2

Appendix B

Source Code Repository

The code used in this project is available at the following GitHub repository:

https://github.com/areydavila/Application-of-One-Class-Models-for-Financial-Risk-Classification

Bibliography

Abd Rahman, Hezlin Aryani and Seng-Huat Ong (2020). “Predictive Performance of Logistic Regression for Imbalanced Data with Categorical Covariate”. In: *Pertanika Journal of Science and Technology* 28. DOI: 10.47836/pjst.28.4.02.

Schölkopf, Bernhard et al. (1999). “Support Vector Method for Novelty Detection”. In: *NIPS* 12, pp. 582–588.

Liu, Fei Tony, Kai Ting, and Zhi-Hua Zhou (2009). “Isolation Forest”. In: pp. 413 – 422. DOI: 10.1109/ICDM.2008.17.

Casale, Pierluigi, Oriol Pujol, and Petia Radeva (2014). “Approximate polytope ensemble for one-class classification”. In: *Pattern Recognition* 47.2, pp. 854–864. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2013.08.007.

Grubbs, Frank E. (1969). “Procedures for Detecting Outlying Observations in Samples”. In: *Technometrics* 11.1, pp. 1–21. DOI: 10.1080/00401706.1969.10490657.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar (2009). “Anomaly Detection: A Survey”. In: *ACM Comput. Surv.* 41. DOI: 10.1145/1541880.1541882.

Matthew, Bamidele, John Jude, and Michelle James (2025). “Applications of Anomaly Detection”. In.

Perera, Pramuditha, Poojan Oza, and Vishal M. Patel (2021). *One-Class Classification: A Survey*. DOI: 10.48550/arXiv.2101.03064. arXiv: 2101.03064 [cs.CV].

Peng, Joanne, Kuk Lee, and Gary Ingersoll (2002). “An Introduction to Logistic Regression Analysis and Reporting”. In: *Journal of Educational Research - J EDUC RES* 96, pp. 3–14. DOI: 10.1080/00220670209598786.

Seitshiro, Modisane B. and Seshni Govender and (2024). “Credit risk prediction with and without weights of evidence using quantitative learning models”. In: *Cogent Economics & Finance* 12.1, p. 2338971. DOI: 10.1080/23322039.2024.2338971.

Holzmann, Hajo and Bernhard Klar (2024). *Robust performance metrics for imbalanced classification problems*. arXiv: 2404.07661 [stat.ML].