FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Detection of fraudulent financial statements in KB
Student:	Michal Šolc
Supervisor:	Ing. Josef Ditrich, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2021/2022

Instructions

- Discuss detailed specifications and the meaning of the input data with the supervisor and other team members.
- Analyze existing approaches for fraud detection from both time series and internal relations within on statement perspective.
- Suggest your own approach.
- Using AI and data processing methods, describe, implement, and evaluate chosen approaches.
- Propose a way to improve your results.

Electronically approved by Ing. Karel Klouda, Ph.D. on 7 January 2021 in Prague.

Bachelor's thesis

Detection of fraudulent financial statements in KB

Michal Šolc

Department of applied mathematics
Supervisor: Ing. Josef Ditrich, Ph.D.

May 13, 2021

Acknowledgements

I would like to sincerely thank to KB for providing me the opportunity to create this thesis with their help and data. I namely thank to Josef Ditrich for supervising, Tomáš Pazák for mentoring and Matěj Stránský for initial organization.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on May 13, 2021 …………………

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Šolc, Michal. *Detection of fraudulent financial statements in KB*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2021.

Abstrakt

Tato práce zkoumá možnosti detekce podvodných finančních výkazů. Cílem práce je najít vhodný přístup k detekci podvodných finančních výkazů v Komerční bance. V rešeršní části jsou ukázany přístupy k detekci navržené několika autory. Poté je představeno několik machine learningových modelů a přístupů. Na základě tohoto je navržen přístup, který bude sloužit jako základ pro budoucí výzkum v této oblasti v KB. Přístup je navržen s přihlédnutím k měnění hustoty minoritní třídy v trénovací množině a toto jo testováno a komentováno na několika klasifikátorech a metodách předzpracování dat. Tato práce přinesla několik modelů a pozorování, které budou v Komerční bance sloužit dále pro výzkum.

Klíčová slova Podvodný finanční výkaz, klasifikační algoritmy, podvzrokování, předzpracování dat, banky, finanční podvod, Komerční banka

Abstract

This thesis surveys the possibilities of automated financial statement fraud detection. Main goal of this work is to find an appropriate approach to this problem in Komerční banka. In the research part, current approaches are shown. Machine learning models and techniques are introduced and with that, custom approach, that will serve as a basis for future research in this filed in KB. Approach is designed with regards to changing the density of minority class in training set and is tested and commented on top of various classifiers and data preprocessing methods. This work brought various models and observations, that will serve for future research in KB.

Keywords Fraudulent financial statement, classification algorithm, undersampling, data preprocessing, banks, financial fraud, Komerční banka

Contents

List of Figures

List of Tables

Introduction

Frauds are as old as humanity itself. First known financial fraud is dated back to 300 B. C. when a Greek merchant named Hegestratos attempted something we would now call ”insurance fraud.“

Nowadays, there are many types of financial fraud, starting with money laundering and ending with cutting taxes. There are many regulatory policies, which require banks to monitor the behavior and prevent such frauds. When banks provide a loan, it is in their best interest to have an overview of the client for themselves. This urge of overview led to a document form named financial statement. Thanks to such document, banks should be able to understand the financial situation of a given entity with help of a list of assets, liabilities, gains, losses, and cash flow. Based on that information, they can propose the entity a loan within their risk tolerance.

This mechanism creates an environment for a fraud. A typical fraud person or entity aims to get something he would normally not reach if he reflected the truth in his statement. Maybe he wants access to money which he then plans to use for different purposes than stated or wants a higher loan than he could normally get. Both such types of fraud can lead the client to a “lossy default”, which means that the client ends up in bankruptcy and the bank writes off some money from this client because he is unable to pay it back.

There are control mechanisms for such fraudulent behavior, yet there are too many clients ending in “lossy default” causing the banks lose a lot of money. The control mechanisms are subject of business secret, thus is not publicly available. But most of the time, they are controlled manually or by some set of defined rules created by humans when it comes to detecting fraudulent financial statements.

Objectives

In this thesis, I will propose a machine learning approach for the detection of fraudulent financial statements in KB. I will also focus on the importance of data preprocessing – e.g., balancing data and dataset transformations.

Thesis structure

This thesis consists of four chapters. The first chapter gives an introduction to current approaches for financial statement fraud (FSF). The second chapter gives a theoretical basis for used machine learning. In chapter three, custom approach is proposed, implemented and evaluated. Chapter four discusses the obtained results and proposes future improvements.

Chapter 1

Existing Solutions

General overview

Models for Automated fraudulent statement fraud (FSF) detection in banks on its clients are subject of business secret an thus not publicly available. As of KB and FSF, there are mostly predefined sets of rules created by a human expert as so far it outperforms any ML-based model.

Generally speaking, FSF using machine learning was mostly done on big corporate clients whose reports are publicly available.

The study from 2020 [1] defines various implementation tasks and based on survey of existing solutions concludes their handling in a table 1.1. In the second column, there are most used solutions along with their proportion in surveyed studies.

Implementation issue	**Handling**
Fraud definition	Investigation by authorities (63%)
Data features	Financial ratios (52%)
Data imbalance	Match fraud firms with non-fraud ones (71%)
Data region	USA (38%) and Taiwan (13%)
Data size	min (27), mean (2 365), max (49 039)
Methods used	ANN, logistic regression and SVM
Feature selection	Filter based approaches (69%)
Missing data treatment	Not specified or deleted records (94%)
Performance measures	Classification accuracy (35%)
Learning approach	Supervised classification (97%)
Best detection method	Varies across data sets

Table 1.1: Implementation handling [1, p. 100, Table 7]

According to the study, current approaches use 2 main definitions of a fraud. First is ”An investigation by authorities” and second is a ”Qualified audit opinion”. Some papers use combination of both.

As the problem is mostly handled as a supervised classification task, therefore supervised machine learning models are used in the approaches. In studies surveyed in [1], most frequent models are artifical neural network (23%), logistic regression (18%), support vector machines (13%) and decision trees (12%).

Defining a fraud is a key phase for supervised learning methods as when labeled wrongly even in a few cases, it has a non negligible impact on the models performance due to a high class imbalance.

As a financial statement has a lot of fields, It brings another problem, that is a high number of dimensions for the classifiers. Most of the studies use filter based approaches to reduce dimensionality. Those methods are based on statistical analysis and not using wrapper or projection methods.

One of the problems stated by the author of [1] is that time series approach is not taken into consideration. There is only one study from a 2005 exploring time series relations in financial fillings from statistical perspective [2].

Models interpretability, as confirmed by my supervisor, is one of a key aspects of the final model. The decision should be interpretable as there is always a need to provide a reason for a decision, which discriminates some black box models, such as neural network, and favors models as decision trees. However, scenarios when the black box model functions as a highlighter of a suspicious statements can be useful as it at least provide initial warning for future investigation by a human expert. Model interpretability is mostly ignored in the literature.

In general, automated FSF detection is highly imbalanced problem, because there are more non-fraudulent financial statements. As author also pointed out, the most common performance measure (Classification accuracy) is wrong method to use due to high class imbalance in this field.

As for the imbalance itself, 71% of observed studies from [1] match the fraudulent firms with non fraudulent ones, creating balanced dataset. Authors sometimes use oversampling [4] of a minority class or undersampling [3] of a majority class to train the model on different prior fraud probabilities, which brings notable results.

Specific example

In a mentioned study from 2011 [3], author examined different prior probability levels, meaning that while training the model, he undersampled the training data and evaluated the model on original distribution. In this study author concluded that logistic regression and support vector machines outperforms rest of the models, measured by estimated relative error. Author also found out that changing prior fraud probabilities has an impact on the models performance.

Komerční banka

Publicly available studies however use big clients mostly listed on exchanges, who are required to have much more detailed financial statements with certified audits. As of clients in KB, that is not the case because clients of the bank have much smaller turnover. As the standards for financial statements for bigger exchange listed companies differ from the clients of KB, for example text mining approaches cannot be used as the statements consists of financial variables only.

Chapter 2

Theoretical Basics

In this chapter, I will provide theoretical basis for this thesis on which I build the approach later on.

2.1 Machine learning

Machine learning (ML) is mostly defined as a study of self-learning algorithms. In general, it is a process where computer tries to understand the structure of the data and is able to do a decision based on that understanding. So, far, it is the best attempt to imitate human process of learning.

There are 2 main categories of ML: Unsupervised and supervised learning. Unsupervised learning is used when data are not labeled and is used to understand the structure of the data and it's internal relations. Supervised learning on the other hand expects labeled data and then tries to understand the relation between the input variables and the target. As the task of fraudulent financial statement detection is a classification problem, I will use supervised learning in this thesis.

2.2 Supervised learning

Supervised learning is a machine learning approach focused on mapping input values to output values. Let's consider the column Y from dataset N such as $Y = N_{:}, j$ often referred to as a target variable. Let X be the dataset without column Y: $X = N \setminus Y$. Then supervised models try to find function
f: $X \mapsto Y$ [5].

2.2.1 Decision tree

Decision tree is easy-to-interpret machine learning model. The tree consists of nodes. There are 2 types of nodes in decision tree, internal node and leaf node. This model splits the dataset to two subsets recursively in each node, until stopping criterion is met.

Splitting techniques

In a tree, each node looks for an optimal split. For that purpose, two main metrics exists. Those are *Entropy* and *Gini*. Let p_i be number of samples belonging to i^{th} class of target variable. Having n classes (2 in our case of binary classification) then:

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$$Gini = 1 - \sum_{i=1}^{n} p_i^2$$

The entropy measure the disorder of the features with the target. Gini measures the degree of inequality in a distribution. [6]

Stopping criterion

Stopping criterion solves the overfitting [1] problem. Without that criterion, the model would copy the train data and will not work properly on unseen data. Examples of stopping criterions are maximum tree depth, minimum samples on leafs etc. [6]

2.2.2 Ensemble methods

The idea behind ensemble methods is taking multiple ML models, for example decision trees. Ensemble methods are of 2 kinds: Bagging and boosting. [7]

Bagging

From input dataset N divide this dataset to multiple datasets $N_1, \ldots, N_n$ with the bootstrap method. We then use dataset N_i to train i-th tree in the model. When evaluating decision, run the sample through all the trees, that then vote for final decision. This structure is called a *random forest*.

Boosting

Boosting in contrary to bagging learns sequentially, which means that based on evaluation of the first model, second model is tweaked as it iteratively adjust the weight of observations based on last classification.

[1] Overfitting is a high assimilation to train data and the model does not work properly on test data.

2.2.3 Support vector machines

Support vector machines is a supervised learning model. It uses transformation to map the data to a higher dimension. When in higher dimension, it is easier to find a hyperplane which separates the data with maximal margin, therefore providing better classification results. Support vector machines used for classification are called support vector classifiers (SVC) [8].

2.2.4 Logistic regression

Logistic regression is supervised learning technique. Unlike other binary classification models, it tries to estimate the likelihood of target variable being equal to 1. The probability is denoted as follows:

$$P(Y = 1|x, w)$$

where

$$x = (x_1, \ldots, x_n)$$

are the features. And

$$w = (w_0, \ldots, w_n)$$

are coefficients. We need to constrain this linear expression:

$$w_0 + w_1 x_1 \ldots + w_n x_n$$

to interval [0, 1]. We can achieve this by putting those numbers to Sigmoid function.

$$f(x) = \frac{\mathrm{e}^{\mathrm{x}}}{1 + \mathrm{e}^{\mathrm{x}}} = \frac{1}{1 + \mathrm{e}^{-\mathrm{x}}}$$

To get the coefficients, use MLE (Most Likelihood Method).

MLE

$$P(Y = 1|x, w) = \frac{\mathrm{e}^{\mathrm{w}^{\mathrm{T}}\mathrm{x}}}{1 + \mathrm{e}^{\mathrm{w}^{\mathrm{T}}\mathrm{x}}}$$

$$P(Y = 0|x, w) = 1 - \frac{\mathrm{e}^{\mathrm{w}^{\mathrm{T}}\mathrm{x}}}{1 + \mathrm{e}^{\mathrm{w}^{\mathrm{T}}\mathrm{x}}}$$

Having i-th data point with the target variable Y_i and feature vector x, the probability of such point is noted as follows:

$$p_{Y i}(x_i, w)$$

Assuming that the data points are independent. Then the probability of the dataset is:

$$L(w) = \prod_{i=1}^{N} p_{Y i}(x_i, w)$$

By maximizing this function, we gain parameters w of the model. [10]

2.2.5 Artificial neural network

Artificial neural network is a highly used and complex model. Neural network model consists of neurons. Output of neuron is gained by activation function f calculated from other neuron inputs. Artifical neural network (ANN) is then set of those neurons connected in multiple layers.

2.3 Data preprocessing

Data preprocessing is one of a key phases of ML model development. This part of process handles multiple problems often encountered in the real world datasets.

2.3.1 Imbalanced classes

In this thesis, I need to deal with imbalanced class distribution as number of fraudulent financial statements is by an order of magnitude smaller than non fraudulent statements.

Imbalanced target class distribution can be handled in 2 ways [13]:

Undersampling

Undersampling is deleting samples from majority class in different ways. Examples are Tomek Links or a random undersampling.

Oversampling

Oversampling the minority class, i.e. creating synthetic data. Common approach is random oversampling or a SMOTE algorithm.

2.3.2 Missing values

Missing values can be a big problem for classification, as classifiers often cannot handle them. There are solutions for missing values imputation, for example deleting missing values, substitute missing data with a constant, impute a mean or predict missing values. [11]

2.3.3 Dimensionality reduction

Dimensionality reduction is a technique used to reduce the number of input variables for the classifier. The high number of dimensions can lead to a phenomenon called *Curse of Dimensionality.* As with growing dimension, the pairwise distance of the datapoints converges to the same value and thus makes it harder for some classifiers to make proper predictions. [12] Also, models with too many input features tend to overfit.

Dimensionality reduction approaches can be divided into 2 categories: [9]

1. Projection methods
    - Principal component analysis
    - Autoencoders
2. Feature selection
    - Unsupervised methods
        - Random feature selection
        - Mutual feature correlation
    - Supervised methods
        - Wrapper - Recursive feature selection
        - Filter - Statistical methods

PCA

Principal component analysis is method for dimensionality reduction by transforming the dataset in a lower dimensionality space by identifying what parts the data holds most information. To compute the parts, a covariation matrix is needed. Let $X_1 \dots X_p$ be features from the dataset X. Covariation matrix is then denoted as follows:

$$cov(X_j, X_i) = \frac{1}{N-1}\sum_{k=1}^{N}(x_{k;i} - \bar{x_i})(x_{k;j} - \bar{x_j})$$

From this covariation matrix, it's eigenvectors and eigenvalues are computed creating the basis of space R^p. Then represent the data in that basis and choose first n vectors with largest eigenvalues to obtain n new features with highest variance. [14] According to [15], standardization and normalization helps when using PCA. Simply put, it is because of lowering gaps between columns variations. For example if column ranges from 100 to 10000, it will have naturally bigger variance than column which ranges from 0 to 1.

Univariate analysis

Univariate feature selection approach examine one feature at a time with respect to the target variable. When having numerical input and categorical output, we can use Analysis of Variance (ANOVA) statistical method. [16]

Recursive feature selection

Recursive feature selection is a wrapper method in which there are features selected based on performance of a classifier.Example of such classifier is a decision tree. It is then trained using all features, for those features, it computes their importance level and drops the least important ones. This process is recursively repeated until desired number of features is obtained. [17]

Correlation feature selection

This approach aims to merge highly correlated features and thus removing redundant information in dataset.

2.3.4 Dataset transformations

Dataset transformation is a procedure where problems in data are removed. Some of the most used methods are desribed below. [18]

Normalization

Normalization rescales the data to range [0, 1] feature-wise. Consider feature vector x, then normalized vector X is computed as follows:

$$X = \frac{x - min(x)}{max(x) - min(x)}$$

Standardization

Standardization rescales all features to have mean of 0 and standard deviation of 1. Feature vector x, then new vector X is:

$$X = \frac{x - \mu}{\sigma}$$

where μ is a mean a nd σ is a standard deviation.

Binning

Binning is a process to create discrete values from continuous ones. We say that we sort them into *bins*. There are 2 main approaches to binning. *Equal width* aims to have the bins equally wide, meaning that the feature is partitioned into k equally wide intervals. In contrary *equal depth* having same number of points in each bin, maintaining equal depth.

2.4 Model evaluation

Training is one of a key phases of ML model development. But once the model is trained, we need to recognize how well it will behave on data that it has never seen before. That is what evaluation is for. For the sake of evaluation, we split dataset into 3 parts.

- **Training set** is used to train the model.
- **Validation set** is used for fine-tuning the model, for example to tune it's hyperparameters. [2]
- **Test set** is set used for evaluation purpose as the model tries to predict something it has never seen before.

When it comes to scoring the quality of a model for a binary classification, the models prediction belongs to one of those classes:

- **True positive (TP)** means that the sample belongs to positive category and the model recognized it.
- **True negative (TN)** means that the sample does not belong to positive category and the model recognized it.
- **False positive (FP)** means that the model predicted positive, but sample belongs to negative class.
- **False negative (FN)** means that the model predicted negative, but the sample belongs to positive class.

[2] Hyperparameter is a model parameter which controls how the model learns. For decision tree, it is for example the maximum depth or minimum samples in a leaf. For neural network, it could be the sizes of hidden layers etc.

Based on those classes, many metrics for binary classification are build:

Accuracy

Accuracy shows how many samples are classified correctly.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

Precision

Precision shows how many positives classified by the model are actually positives:

$$Precision = \frac{TP}{TP + FP}$$

Recall

Recall shows how many positives the model recognized:

$$Recall = \frac{TP}{TP + FN}$$

F_1-score

F_1 score is a harmonic mean of recall and precision and thus being a robust metric [19]:

$$F_1 = \frac{2}{Recall^{-1} + Precision^{-1}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

For f_1-score, f-score is used later in the text.

Confusion matrix

Confusion matrix is a visual representation of models prediction.

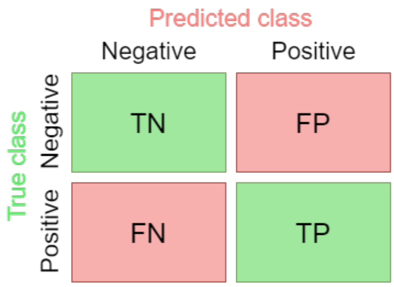


Figure 2.1: Confusion matrix

2.4.1 K fold cross-validation

K fold cross-validation is a statistical method used to estimate the capabilities of ML model. Algorithm first splits dataset into k folds. In k iterations it trains the data using k-1 folds and evaluates the model on the remaining fold as shown in figure 2.2.

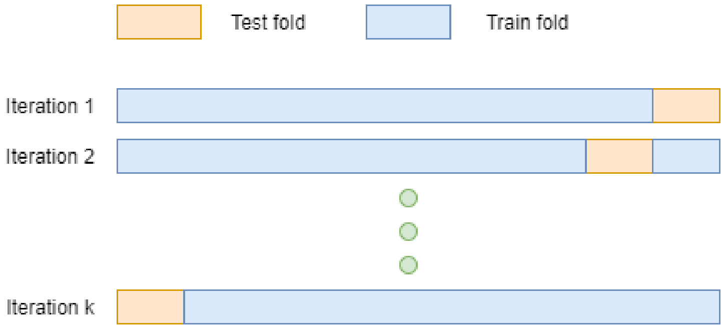


Figure 2.2: Cross validation

Cross validation can be used as a substitute of a validation set when tuning models hyperparameters in order to have more data to train on. In such scenario, dataset is firstly divided into *train* and *test* parts. Then, using cross validation to tune hyperparameters on train data and finally evaluate the model on test data. [20]

Modified cross validation

In this thesis, I use *stratified* cross validation with undersampling as I work with prior fraud probabilities and have highly imbalanced dataset. The purpose of stratification is to maintain equal class distribution among folds. In each iteration, the dataset is split into train and test folds. Then it is undersampled to desired distribution. Preprocessors are fit, classifier is trained and model is evaluated on a test fold as shown in 2.3.

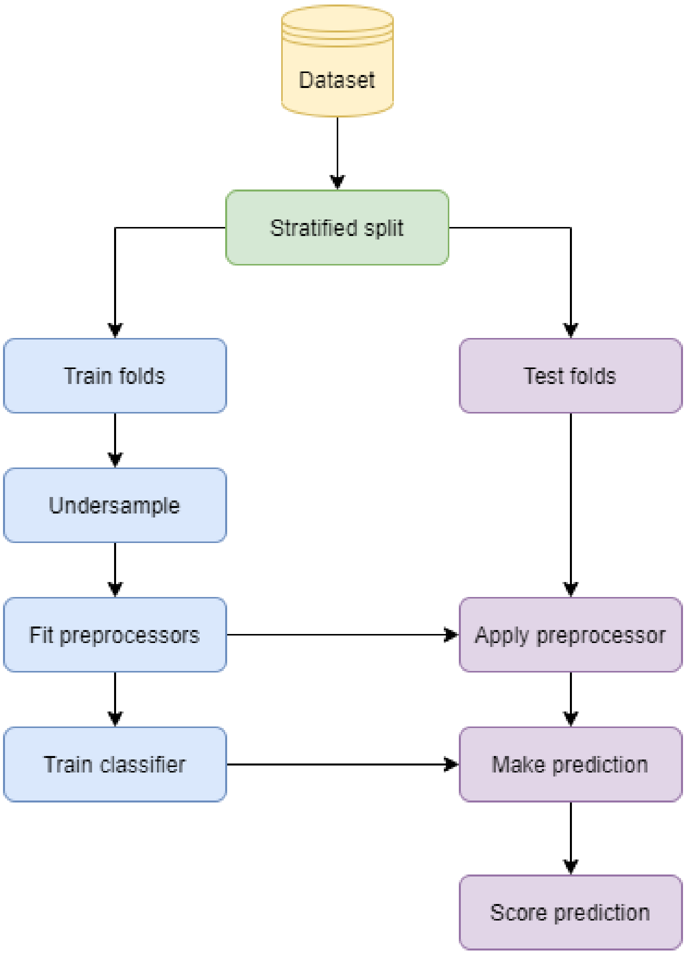


Figure 2.3: Stratified cross validation with undersampling

Chapter 3

Realization

3.1 Tech stack

Python

Python is a commonly used interpreted programming language among data scientists. It is an open source project with many extensions from community or development teams called *packages* or *libraries*. [21]

Jupyter

Project jupyter is an open-source project providing interactive environment for various programming languages including python. [22]

scikit-learn

scikit-learn is an open source library used for machine learning. It implements various data preprocessing methods, models and evaluation techniques. [23]

Matplotlib

Matplolib is a library in python for visualizations. [24]

NumPy

Numpy is a high-performance open source package used for matrix operations in python. [26]

Pandas

Pandas is a library build on to of numpy. It provides various high-performance data structures, functions for data loading from files, or even from databases by SQL queries. [25]

3.2 My approach

In this section, I propose a way of solving the issue of detecting financial fraudulent statements from data available in KB. Based on the research and consultations, a process pipeline was designed (3.1). As in [3], I will explore training on different prior fraud probabilities, but with regards to different preprocessing methods and even their combinations. Projection methods will be also examined and tailored approach with digits distribution is also implemented. The approach is rather quantitative as this is a specific dataset for a specific environment and can serve as a basis for further research in KB and automated FSF detection.

3.2.1 Fraud definition

For this thesis, I recognize 2 definitions of a fraudulent financial statement.

Fraud by charge off

First type is that statement, after which the client ended up in lossy default. This shows that the bank was not able to score that client properly and thus ending in a loss for the bank, therefore may indicate successful fraudulent behavior not captured early enough. This technique marks successful frauds, but can also mark a non-fraudulent behaviour.

Marked fraud

Are clients marked by KB fraud department as frauds. Then theirs financial statements is labeled as fraudulent. This labeling technique labels all frauds considered by KB experts, but some successful frauds can be missed due to a systematic human error.

3.2.2 Model usage scenarios

High precision

Models with high precision can function as a nice support for human analyst as when they predict fraud, it is most likely correctly predicted and the statement require further attention.

High recall

In contrary, models with high recall can reveal most of the frauds and therefore there is a low probability that a fraud stays undetected.

High F score

Models with high f-score are generally robust as they have decent recall and precision combination. Thus first objective is to maximize f-score during development.

3.2.3 Introduction of concepts

In this section, I will introduce concepts that I use later in the text.

Dataset of charged-off clients is called *dataset A*. Dataset with expertly marked frauds is called *dataset B*.

When referring to *prior fraud probability*, I speak of frauds/non-frauds ratio in training folds during training phase. In following text, this will be referred as a *prior*.

When stating *evaluation*, I refer to 10 fold stratified cross-validation with undersampling as introducted in 2.4.1.

I run every evaluation on those priors 5%, 10%, 25%, 50% for dataset A, and 1%, 5%, 10%, 25%, 50% for dataset B.

When mentioning *classifiers*, I mean those: Artificial neural network (ANN), Decision tree (DT), Random forest (RF), Logistic regression (LR), Support vector classifier (SVC).

As a *preprocessor*, I refer to any entity capable of changing the dataset. I use following preprocessors in 3 categories:

- Missing values
  - Single value imputation
  - Mean value imputation
  - Median value imputation
- Dimensionality reduction
  - Recursive featrue elimination (RFE)
  - Univariate analysis with ANOVA
  - Feature mutual correlation
  - Principal component analysis (PCA)

- Dataset transformations
  - Standardization
  - Normalization
  - Binning

As a model *evaluation metric*, I use precision, recall and f-score.

As a result of evaluation, I generate a *report* consisting of:

- Mean values for f-score, precision, recall for each preprocessor.
- Mean values for f-score, precision, recall for each classifier.
- Top 10 classifiers, priors and preprocessors by f-score.
- Top 10 classifiers, priors and preprocessors by precision.
- Top 10 classifiers, priors and preprocessors by recall.
- Metrics score as a function of a prior for each preprocessor–classifier combination.

I use mean values to see how each model is consistent. I also observe maximum values to find out whether some model is good with specific combination and also to see on which prior provides the best results for a given metric. Based on that report, I select combination of preprocessing, classifier and prior fraud probability for final model tuning.

During *model tuning*, the whole dataset is first split into train and test folds in 70:30 ratio, the exact number of fraudulent and non-fraudulent clients for each dataset is shown in tables below. On the train part, 5-fold stratified cross validation with undersampling is used to find best hyperparameters. Classifier is then trained using the train set with best hyperparameters and tested on the test set.

	Fraud	Non-fraud
Train	503	11306
Test	216	4796

Table 3.1: Dataset A

	Fraud	Non-fraud
Train	78	11308
Test	33	4794

Table 3.2: Dataset B

A *digits distribution* in a statement were computed and all classifiers were evaluated using that features as well. The idea behind that was to reveal whether the statement was manipulated using only the distribution of digits in it.

3.2.4 Data description

The data for financial statements are for corporate clients in KB. I have 719 fraudulent statements for the dataset A and 111 statements for dataset B. To those frauds, I mined 16,102 non-fraudulent statements from KB's data warehouse teradata [28] using pyodbc library. [27]

I had 3 main categories of statements in my dataset differing by accounting type.

- Simple accounting
- Accounting
- International Financial Reporting Standards (IFRS)

Simply put, those categories overlap in the terms of available fileds, but some fileds are not required for every category.

Dataset A had 4.3% ratio of fraudulent statements, whereas dataset B had 0.7% ratio.

Initially, both datasets had around 250 features corresponding to observed fields from financial statements. After performing analysis of redundant features, or columns with no data, I ended up having 160 columns for both datasets.

Example of observed values in financial statement in KB are:

- Gearing - measures the ratio of dept to equity
- Revenues
- Short term dept
- Long term dept
- Earnings Before Interest, Taxes, Depreciation and Amortization (EBITDA)

Both datasets had approximately 19% of missing data, that was caused by the difference between each accounting category and thus some statements had missing values in non-required fields.

3.2.5 Pipeline description

Based on previous research and data exploration. With the help of my supervisor and mentor, I created following pipeline for a model selection.

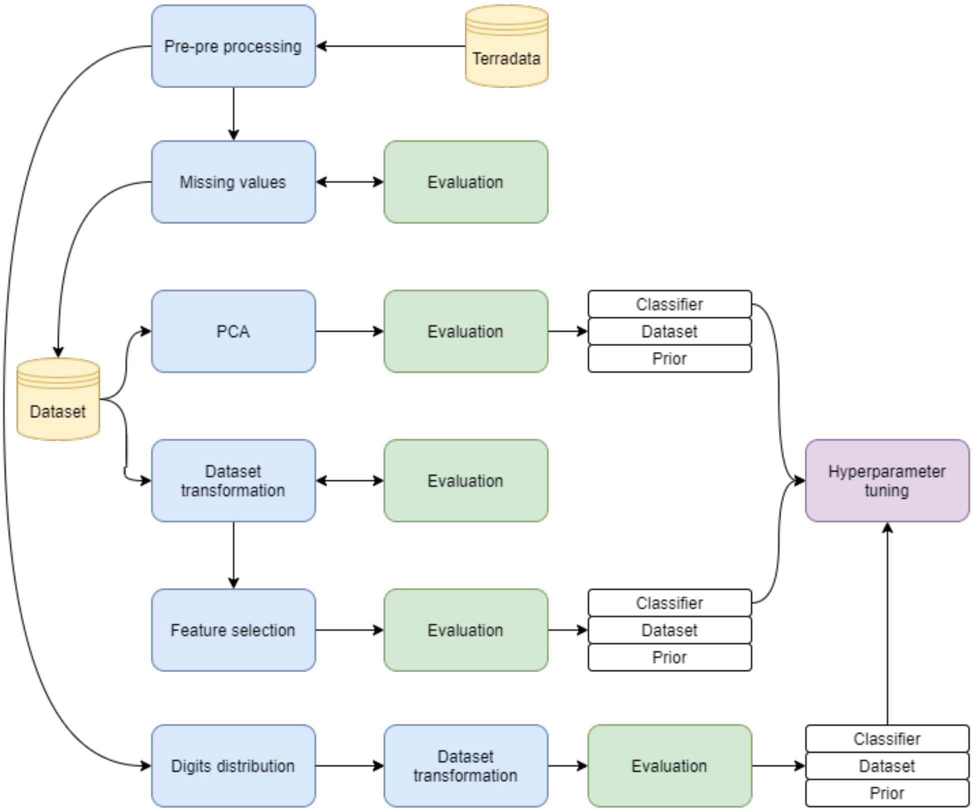


Figure 3.1: Model selection pipeline

I divided the pipeline to several steps. At first, I retrieved the data and transformed them to proper format from the data warehouse. Then, I dealt with missing values. I used imputed values as a basis for next experiments. In each experimental part, I performed preprocessing illustrated by a blue box. Those steps were then evaluated and best combinations of model, preprocessor and prior fraud probability were selected for final model tuning.

3.3 Experiments

This section contains performed experiments and their results. First, I decribe missing values handling. Then I examine PCA. After that I use various data transformation techniques and feature selection methods. Lastly, classifiers are evaluated on digits distribution from both datasets.

The results are firstly presented, briefly commented and then discussed in the last part of this section.

3.3.1 Missing values

At first, missing values count was noted and added as a new feature. Then a threshold for maximum percentage of missing values in each column was evaluated. Then, the threshold resulting in maximum f-score was chosen and columns above that threshold were removed. For both datasets, optimal threshold was 70%.

For missing values imputation, following approaches were evaluated:

- mean imputation
- median imputation
- constant value imputation for -1 and 0

Those methods were then evaluated and the method with overall best f-score was selected.

	f-score	**prec.**	**recall**
Mean	0.20	0.22	0.41
Med.	0.19	0.24	0.39
-1	0.23	0.24	0.43
0	0.22	0.24	0.43

Table 3.3: Imputation for dataset A

	f-score	**prec.**	**recall**
Mean	0.08	0.10	0.36
Med.	0.07	0.09	0.35
-1	0.08	0.09	0.37
0	0.08	0.10	0.37

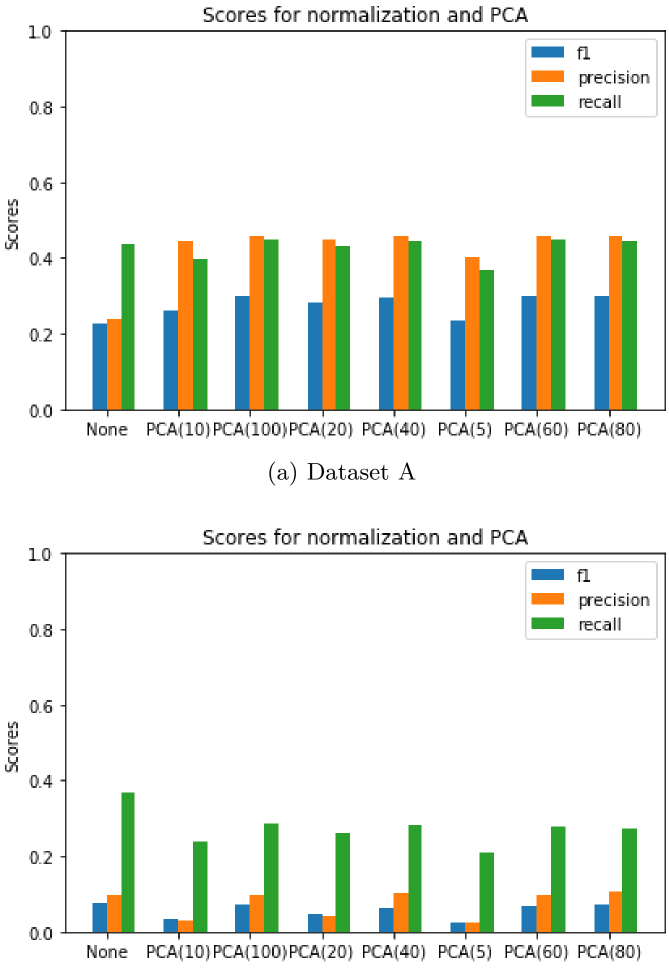
Table 3.4: Imputation for dataset B

Based on those results, imputation of a constant -1 is chosen for dataset A for following steps. Experiment on dataset B resulted in choosing constant 0.

3.3.2 PCA

This section provides results for first technique examined, that was a PCA with normalization.

Dataset A showed big improvements in both precision and f-score even for 5 main components. For dataset B, PCA generally did not help with one exception shown later in this section. Overall results can be seen in a figure 3.2, where the first 3 columns in each graph are scores with no preprocessing applied.



(a) Dataset A

(b) Dataset B

Figure 3.2: PCA comparison

Now best performing models based on report were tuned further for both datasets.

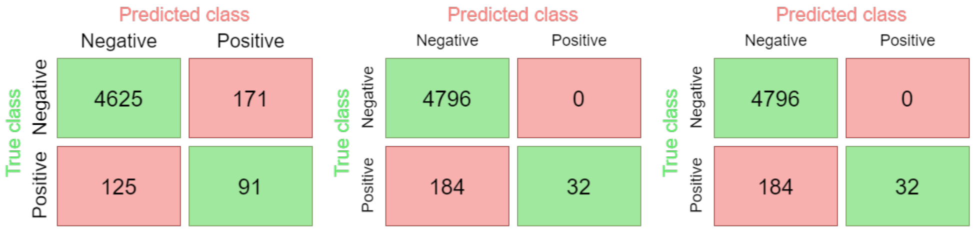
Dataset A

Based on the generated report, I selected 3 models in two categories:

- *Best f-score* (0.4) reached by ANN with 100 principal components on a 10% prior.
- *Best precision* (1.0) reached by RF and SVC on 40 principal components. Both on a 5% prior.

I chose not to take recall into consideration as with high recall was connected extremely low precision values resulting in low f-score.

Confusion matrices (3.3) along with observed metrics (3.5) are shown below:



(a) ANN, 10% prior (b) SVC, 5% prior (c) RF, 5% prior

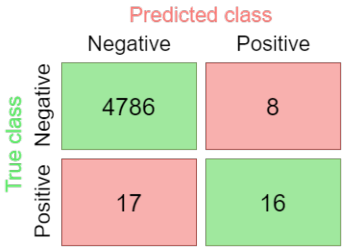
Figure 3.3: Dataset A confusion matricies after PCA

Model	F-score	Precision	Recall
a) ANN, 10%	**0.38**	0.35	**0.42**
b) SVC, 5%	0.26	**1.00**	0.15
c) RF, 5%	0.26	**1.00**	0.15

Table 3.5: Metrics for dataset A and PCA after tuning

Dataset B

From dataset B, I chose ANN for final tuning due to it's high precision (0.78) and decent recall (0.20) on 80 principal components. Model achieved those results on a 1% prior. After tuning, I obtained those results (3.4):



		Predicted class	
		Negative	Positive
True class	Negative	4786	8
	Positive	17	16

Figure 3.4: ANN confusion matrix after PCA on dataset B

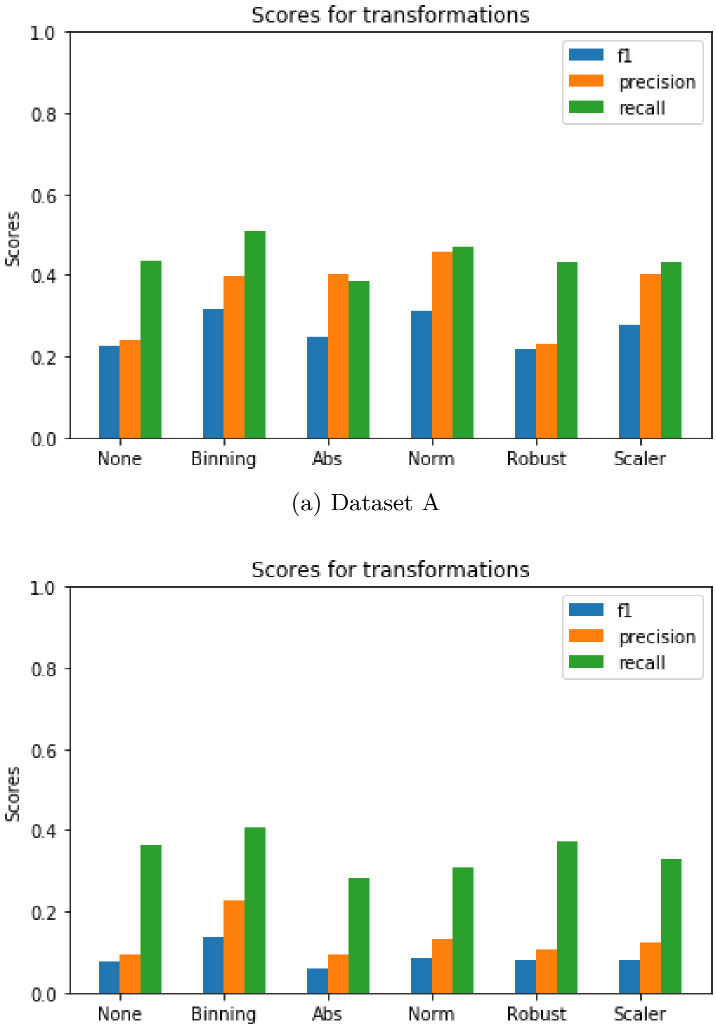
F1: 0.56
Precision: 0.67
Recall: 0.48

This result shows that there is some hidden pattern in dataset B as it was discovered only by neural network and this result totally outstands the others as seen in figure 3.2, where mean value for f-score on dataset B is around 0.1 levels.

3.3.3 Dataset transformations

The goal of this section is to select the best preprocessing technique.

Various techniques implemented in scikit-learn [23] were examined as shown in 3.5. Namely binning, absolute scaling (Abs), normalization (Norm), robust scaling (Robust) and standard scaling (Scaler).



(a) Dataset A

(b) Dataset B

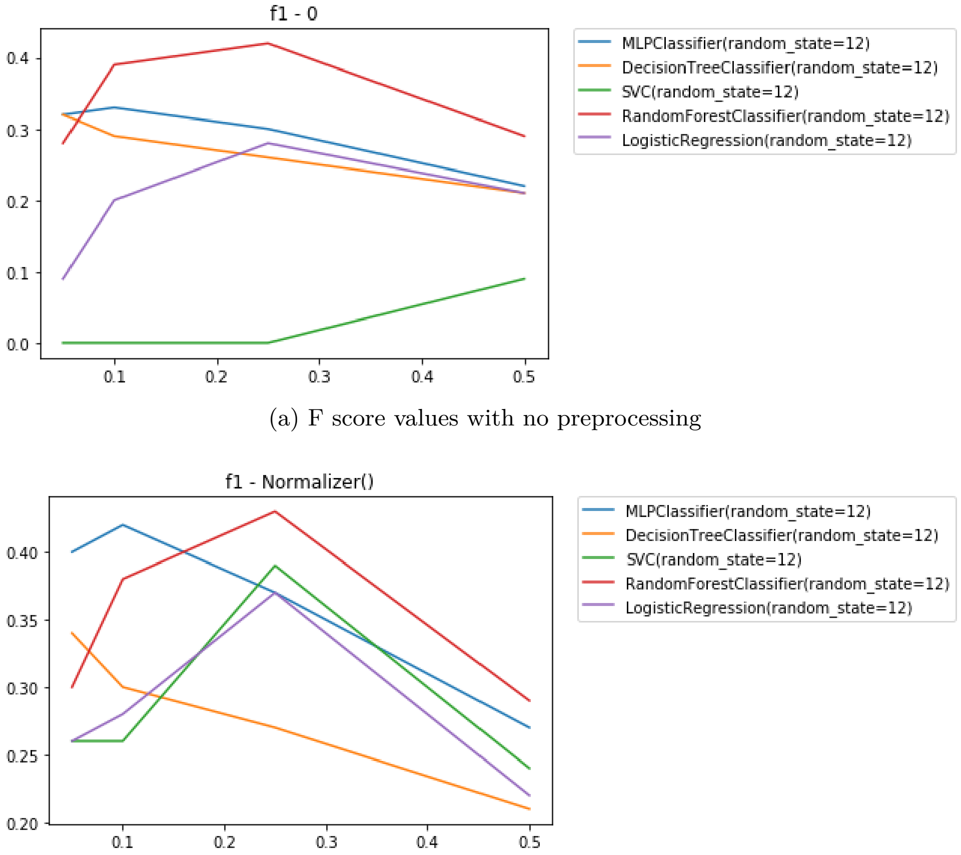
Figure 3.5: Mean values for transformations

Dataset A

When using normalization as a preprocessor for dataset A, there are improvements in f-score shown in 3.6 and in precision 3.7 for all prior levels. It is interesting to observe, that the model is most robust at a prior of 25% with exception of neural network, that achieved best f-score at a 10% prior visualized in 3.6.

More precise, however, were all models on original prior (5%), when models SVC, RF, LR achieved precision close to one after normalization, shown in figure 3.7 (b).

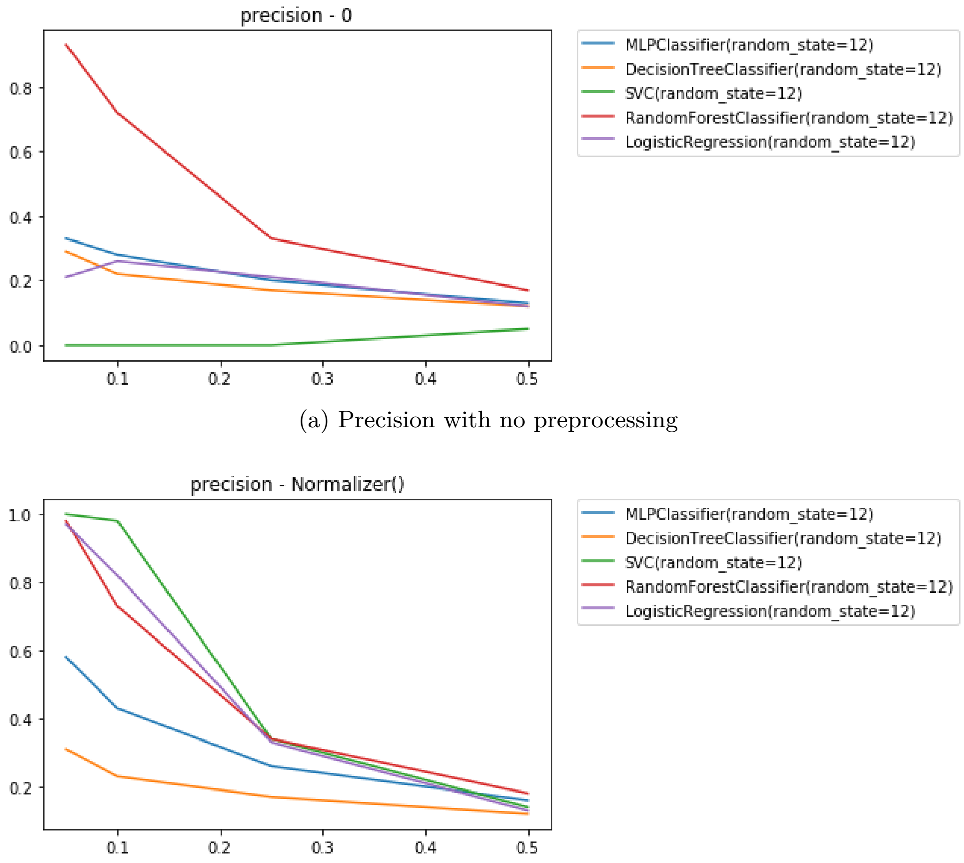
Neural network showed outstanding results for binning with ordinal encoding as it had decent precision and f-score for a 5% prior.



(a) F score values with no preprocessing

(b) F score values after normalization

Figure 3.6: F-score comparison after normalization



(a) Precision with no preprocessing

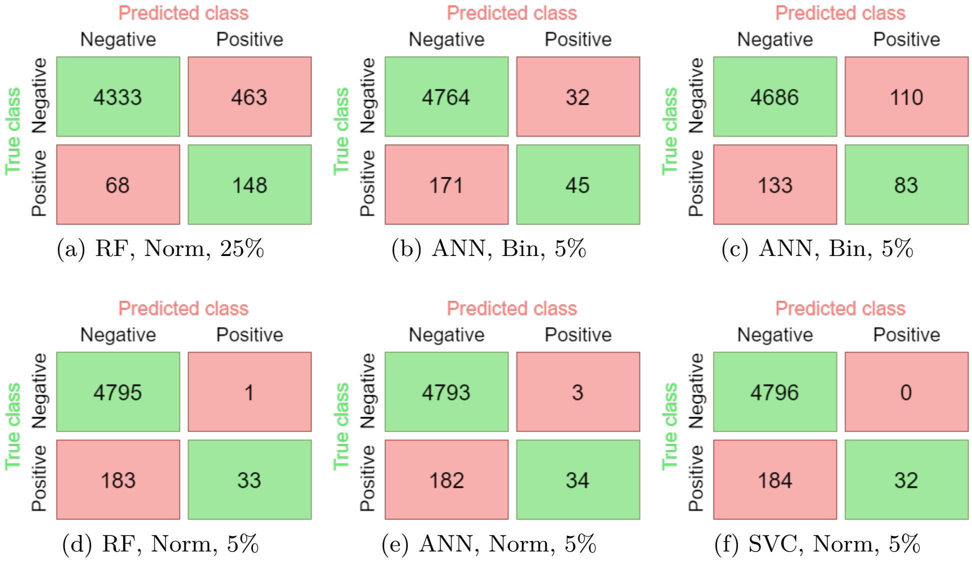
(b) Precision after normalization

Figure 3.7: Precision comparison after normalization

Model tuning

Based on the above figures and generated report that can be found in enclosed media, 6 models, preprocessors and priors were selected for further model tuning with following results.

Confusion matricies for each of selected models are displayed in 3.8. RF stands for random forest, ANN for artifical neural network and SVC for support vector classifier. Norm is normalization and Bin stands for equal width binning with ordinal encoding. The last number under each matrix is a training prior. Note that models c, b are equal, however they were trained using different hyperparameters.



		Predicted class	
		Negative	Positive
True class	Negative	4333	463
	Positive	68	148

(a) RF, Norm, 25%

		Predicted class	
		Negative	Positive
True class	Negative	4764	32
	Positive	171	45

(b) ANN, Bin, 5%

		Predicted class	
		Negative	Positive
True class	Negative	4686	110
	Positive	133	83

(c) ANN, Bin, 5%

		Predicted class	
		Negative	Positive
True class	Negative	4795	1
	Positive	183	33

(d) RF, Norm, 5%

		Predicted class	
		Negative	Positive
True class	Negative	4793	3
	Positive	182	34

(e) ANN, Norm, 5%

		Predicted class	
		Negative	Positive
True class	Negative	4796	0
	Positive	184	32

(f) SVC, Norm, 5%

Figure 3.8: Dataset A confusion matricies after preprocessing

Observed metrics are shown in table below with the best value for each metric highlighted.

Model	F-score	Precision	Recall
a) RF, Norm, 25%	0.36	0.24	**0.69**
b) ANN, Bin, 5%	0.31	0.58	0.21
c) ANN, Bin, 5%	**0.41**	0.43	0.38
d) RF, Norm, 5%	0.26	0.97	0.15
e) ANN, Norm, 5%	0.27	0.92	0.16
f) SVC, Norm, 5%	0.26	**1.0**	0.15

Table 3.6: Metrics for transformations after tuning on dataset A

Dataset B

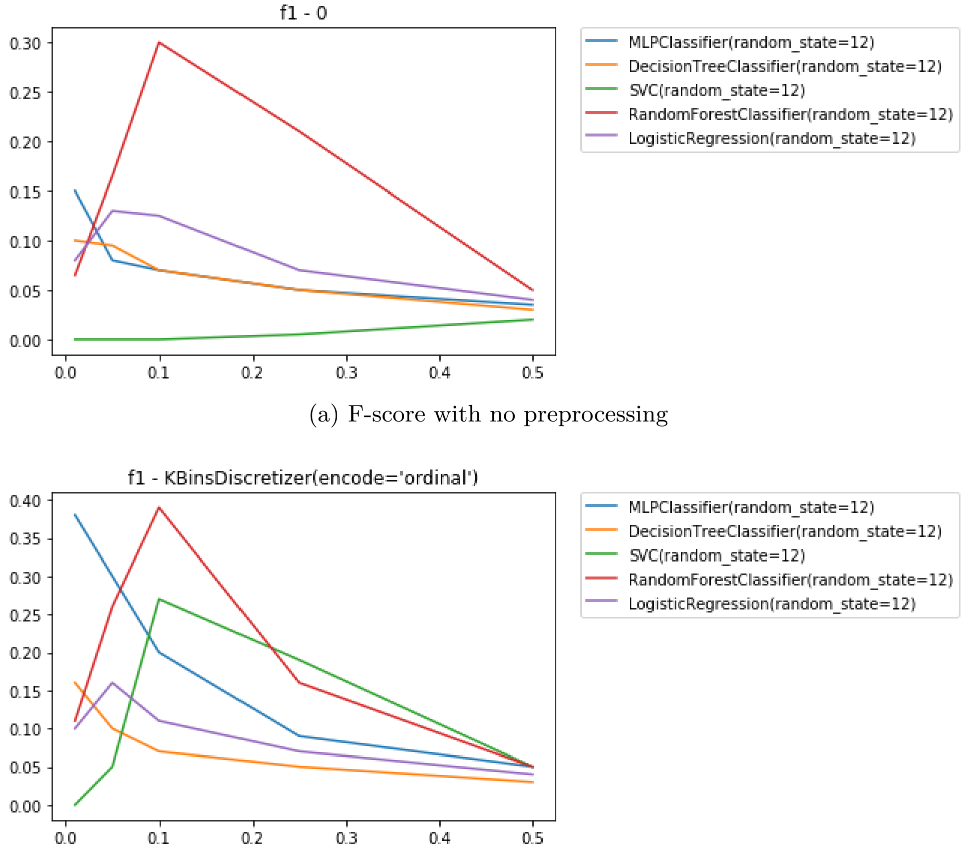
For dataset B is interesting to observe that the best results were achieved on a 10% prior, meaning that it was trained on 17x more dense target than in the original distribution (0.6%).

Best results for f-score were achieved with binning with values 0.39, 0.62 and 0.3 for f score, precision and recall respectively.

However more precise were results with RF on a 5% prior.

MLP with 0.6 precision is also worth mentioning as it was the only model with decent f-score on 1% prior.

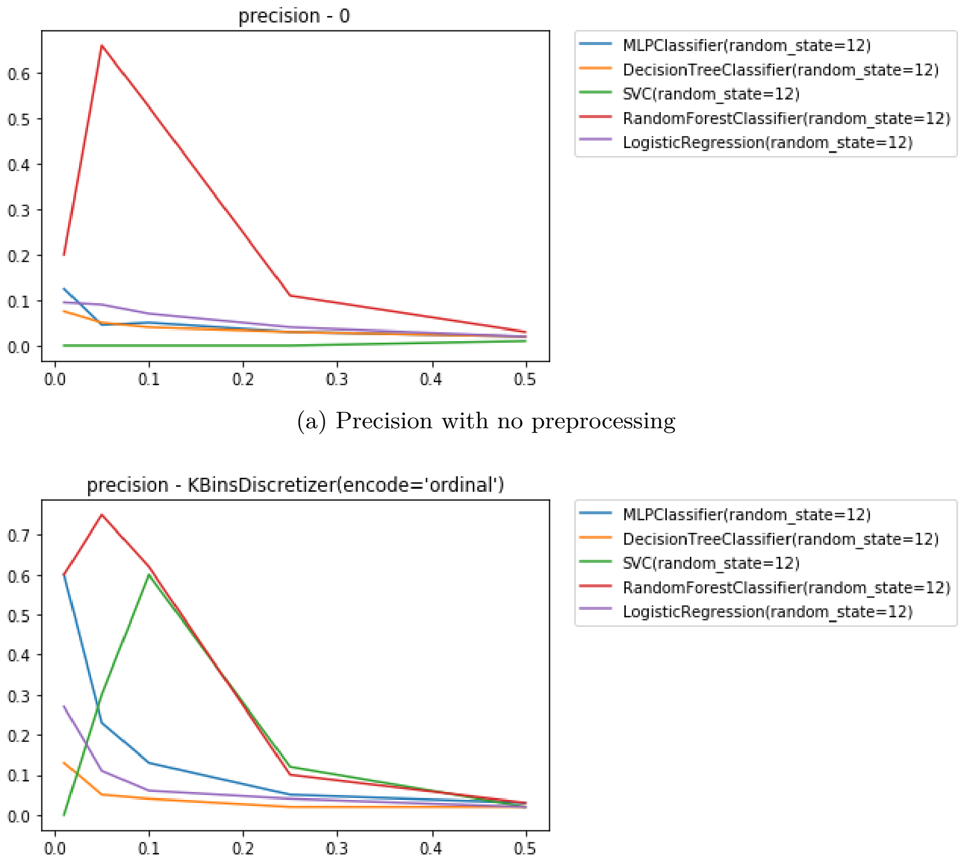
The figure 3.9 shows how the binning improved f-score for all classifiers. Precision improvement is shown in 3.10.



(a) F-score with no preprocessing

(b) F-score values after binning

Figure 3.9: F-score with dataset B after binning comparison



(a) Precision with no preprocessing

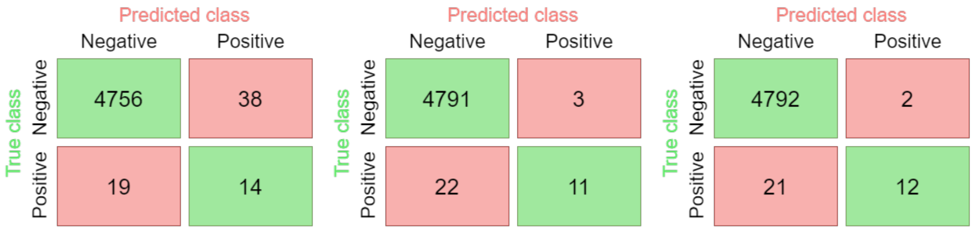
(b) Precision values after binning

Figure 3.10: Precision with dataset B after binning comparison

Model tuning

Based on generated report and figures above, 3 models, preprocessors and priors were selected for further model tuning with following results.

Confusion matricies for selected models are in figure 3.11 and achieved metrics 3.7 are shown below. RF stands for Random Forest, ANN for artifical neural network, KBins is equal width binning with ordinal encoding.



	Negative	Positive
Negative	4756	38
Positive	19	14

(a) RF, KBins, 5%

	Negative	Positive
Negative	4791	3
Positive	22	11

(b) RF, KBins, 10%

	Negative	Positive
Negative	4792	2
Positive	21	12

(c) ANN, KBins, 1%

Figure 3.11: Dataset B confusion matricies

Model	**F-score**	**Precision**	**Recall**
a) RF, KBins, 5%	0.33	0.27	**0.42**
b) RF, KBins, 10%	0.47	0.79	0.33
c) ANN, KBins, 1%	**0.51**	**0.86**	0.36

Table 3.7: Metrics for transformation after tuning for dataset B

3.3.4 Feature selection

For feature selection, 3 approaches were selected and implemented using scikit-learn [23]:

- Deleting correlated features
- Wrapper method - RFE
- Filter method - Univariate analysis using ANNOVA

For the wrapper methods, 3 different estimators were examinded:

- Decision tree
- Logistic regression
- Random forest

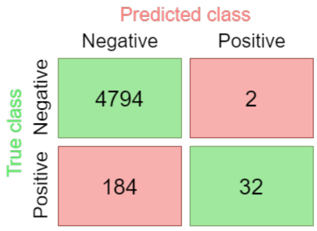
Selected feature selection methods did not provide any further improvements neither as a standalone or in combination with data transformations from previous section. Feature selection was therefore not considered for further model tuning and tuning was done in previous section using only transformations. Full generated report can be found in enclosed medium in a file experiments/Feature_Selection and in experiments/Experimental_Preprocessing.

3.3.5 Digits distribution

As stated before in the text 3.2.3, digit distribution was computed and evaluated. As there were 10 features to examine, each feature ranging from 0-100, no data preprocessing was applied. Results are presented in this section for dataset A. For dataset B, digits distribution did not provide any notable results.

Dataset A

For dataset A, the first report shows interesting results with regards to previous observations. More to be discussed in following section 3.4. Nevertheless, based on report and with regards to model interpretability, random forest on 5% prior was selected, as along with SVC and ANN, it had the best precision and f-score. Model was then tuned and produced following confusion matrix and scores:



		Predicted class	
		Negative	Positive
True class	Negative	4794	2
	Positive	184	32

Figure 3.12: RF on digits distribution

F-score: 0.26
Precision: 0.94
Recall: 0.15

The similarity to best precision models selected in 3.3.2 and also in 3.3.3, is obvious as it detected the same number of positive clients with very low false positive rate.

3.4 Experiment assesment

The main goal of the experimental part was to, in broad perspective, evaluate machine learning approaches for automated FSF detection using different prior fraud probabilities, classifiers, preprocessing methods and even fraud definitions.

In general, I discovered that training the models on modified priors can be useful and is a way to go in a future. Along with that, importance of dataset preprocessing was demonstrated on various examples.

I discussed 3 use case scenarios for the resulting models 3.2.2. I want to asses the experiment for each dataset separately with regards to those scenarios.

Dataset A

For dataset A and use case scenarios, the results varied across methods and priors. Notable result, fulfilling use case 2 (High recall) was achieved by RF after normalization on 25% prior as seen in as seen in 3.8. With recall of 0.69, it detects majority of frauds of the first type.

Result fulfilling the use case 3 (High f-score) was achieved by a neural network in combination with PCA on 10% prior and binning on 5% prior, both achieving f-score of 0.4 with balanced precision and recall.

Arguably most fulfilled case was the first one (high precision). Creating models across classifiers, preprocessing techniques or even priors with 100% precision can serve as a good marking technique because of theirs low FP rate and thus when classified as fraud, investigation by human expert is advised. Among others, this score was achieved by a random forest, therefore creating on option to be interpreted. As the same precision was also achieved by simply exploring the digits distribution, it can lead to an assumption that it is connected with missing values and therefore with different statements categories, as the distribution with more missing values gets more dispersed.

Dataset B

As for dataset B, that was interesting in a way that the big picture, it tries to implement the decision process of KB experts into ML model. Regarding the fact, that density of a target was 0.7%, results were decent. In case of a model from section 3.3.3, where the neural network achieved precision of 0.86, it fulfilled the expectations from the first scenario and can be used for that purpose. Along with that, RF also achieved decent precision and thus providing the possibility of exploring it's decision process and achieving the desired model interpretability.

Chapter 4

Future Improvements

In the follow up work, I suggest to use even more prior fraud probability levels to tweak the range of most suitable prior probability range for different classifiers.

Some advanced undersampling techniques can be examined instead of random undersampling. Oversampling/data augmentation can be explored in order to create more fraudulent samples to train the model on.

As PCA proved itself useful, exploring other projection methods such as autoencoders or manifold learning can provide further improvements.

The results from dataset A suggests that there is some strong pattern for approximately 15% of the clients (as precision was close to one and recall around 0.15). Discovering this pattern can provide useful information about those clients. Removing such clients from the dataset could allow the classifiers to find the patterns that were always overshadowed by this one.

Combining introduced fraud definitions, or even coming up with new one can also provide future improvements.

Conclusion

In the first chapter, existing solutions for automated fraudulent financial statement detection were introduced with one specific example and with current status in KB.

Second chapter provided theoretical basis to support following decisions in next chapter. It described basic ML models along with advanced model evaluations techniques.

In chapter 3, goal for the approach was set, two different fraud definitions were introduced and possible usage scenarios were defined. Based on that, experiments pipeline was designed and experiments were carried out with decent results and interesting observations creating a starting point for future research of automated financial fraudulent statement detection in KB.

Chapter 4 provided ideas for further future development.

In conclusion, ML as a discipline has something to offer in this field and is certainly worth exploring further.

Bibliography

[1] MONGWE, Wilson MALAN, Katherine. (2020). A Survey of Automated Financial Statement Fraud Detection with Relevance to the South African Context. [online]; South African Computer Journal. 32. Available from DOI: http://dx.doi.org/10.18489/sacj.v32i1.777

[2] KIEHL, Thomas Hoogs, Bethany Lacomb, Christina Senturk, Deniz. [online]; Evolving Multi-Variate Time-Series Patterns for the Discrimination of Fraudulent Financial Filings. [last accessed on 25.04.2021]; Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.588.2095&rep=rep1&type=pdf

[3] PEROLS, Johan. (2010). Financial Statement Fraud Detection: An Analysis of Statistical and Machine Learning Algorithms. Auditing A Journal of Practice Theory. 30. DOI:10.2308/ajpt-50009; Also available from: https://www.researchgate.net/publication/256045322_Financial_Statement_Fraud_Detection_An_Analysis_of_Statistical_and_Machine_Learning_Algorithms

[4] Huanzhuo Ye and Lin Xiang and Yanping Gan, (2019), [online]; Detecting Financial Statement Fraud Using Random Forest with SMOTE. Available from DOI: https://doi.org/10.1088/1757-899x/612/5/052051

[5] BROWNLEE, Jason; Supervised and Unsupervised Machine Learning Algorithms; Machine Learning Mastery [online]; 16.03.2016; [accessed on 15.04.2021]; Available from: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms.

[6] KLOUDA, Karel, Rozhodovací stromy [online], 2020, Prague: CTU in Prague, [accessed on 20.03.2021], available from: https://courses.fit.cvut.cz/BI-VZD/@B201/lectures/index.html

[7] KLOUDA Karel, Ensamble metody (rozhodovací lesy, AdaBoost) [online], 2020, Prague: CTU in Prague, [accessed on 20.03.2021], available from: `https://courses.fit.cvut.cz/BI-VZD/@B201/lectures/index.html`

[8] ROHITH Gandhi, Support Vector Machine — Introduction to Machine Learning Algorithms [online]; 07.06.2018; [accessed on 01.05.2021], available from: `https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47`

[9] BROWNLEE, Jason, How to Choose a Feature Selection Method For Machine Learning [online]; 27.01.2019; [accessed on 01.04.2021], available from: `https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/`

[10] KLOUDA Karel, Logistická regrese [online], 2020, Prague: CTU in Prague, [accessed on 23.03.2021], available from: `https://courses.fit.cvut.cz/BI-VZD/@B201/lectures/index.html`

[11] SATYAM Kumar, 7 Ways to Handle Missing Values in Machine Learning [online]; 24.07.2020; [accessed on 01.04.2021], available from: `https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e`

[12] Great Learning Team, Understanding Curse of Dimensionality [online]; 01.10.2020; [accessed on 21.03.2021], available from: `https://www.mygreatlearning.com/blog/understanding-curse-of-dimensionality/`

[13] KORDÍK, Pavel JIŘINA, Marcel, Lecture 7: Data reduction (2nd part); [online], 2020 Prague: CTU in Prague, [accessed on 25.03.2021], available from: `https://courses.fit.cvut.cz/MI-PDD/lectures/index.html`

[14] VAŠATA, Daniel. Redukce dimenzionality [online], 2020, Prague: CTU in Prague, [accessed on 23.03.2021], available from: `https://courses.fit.cvut.cz/BI-VZD/@B201/lectures/index.html`

[15] HOWLEY, Tom Madden, Michael O'Connell, Marie-Louise Ryder, Alan. (2005). [online]; The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data. 209-222. DOI: 10.1007/1-84628-224-1_16. Available from: https://www.researchgate.net/publication/220804070_The_Effect_of_Principal_Component_Analysis_on_Machine_Learning_Accuracy_with_High_Dimensional_Spectral_Data

[16] sampath kumar gajawada, ANOVA for Feature Selection in Machine Learning [online]; 19.10.2019; [accessed on 25.03.2021], available from: https://towardsdatascience.com/anova-for-feature-selection-in-machine-learning-d9305e228476

[17] BROWNLEE, Jason Recursive Feature Elimination (RFE) for Feature Selection in Python [online]; 25.05.2020; [accessed on 25.03.2021], available from: https://machinelearningmastery.com/rfe-feature-selection-in-python/

[18] KORDÍK, Pavel JIŘINA, Marcel, Lecture 5: Problems in data, data cleaning; [online] Prague: CTU in Prague, 2018/2019, [accessed on 26.03.2021], available from: https://courses.fit.cvut.cz/MI-PDD/lectures/index.html

[19] SASAKI, Yutaka. (2007). The truth of the F-measure. Teach Tutor Mater. [online], [acessed on 23.4.2021]. Available from: https://www.researchgate.net/publication/268185911_The_truth_of_the_F-measure

[20] KLOUDA, Karel, Metoda nejbližších sousedů, křížová validace [online] (2020), Prague: CTU in Prague, [accessed on 28.03.2021], available from: https://courses.fit.cvut.cz/BI-VZD/@B201/lectures/index.html

[21] Python Software Foundation; Python [software]; ©2001-2021; Available from: https://www.python.org/.

[22] Project Jupyter development team; Project Jupyter [software]; ©2021; Available from: https://jupyter.org.

[23] scikit-learn development team; scikit-learn [software]; ©2007-2021; Available from: https://scikit-learn.org/stable/.

[24] The Matplotlib development team; Matplotlib [software]; ©2012-2021; Available from: https://matplotlib.org/.

[25] Pandas development team; Pandas [software]; ©2008-2021; Available from: https://pandas.pydata.org/.

[26] NumPy development team; NumPy [software]; ©2019-2021; Available from: `https://www.numpy.org/`.

[27] pyodbc development team; pyodbc [software]; Available from: `https://github.com/mkleehammer/pyodbc`.

[28] teradata development team; teradata [software]; ©2021; Available from: `https://www.teradata.com/`.

Appendix A

Acronyms

ML Machine learning

FSF Financial statement fraud

PCA Principal component analysis

RFE Recursive feature elimination

ANN Artifical neural network

SVM Support vector machines

SVC Support vector classifier

DT Decision tree

RF Random forest

TP True positives

FP False positives

TN True negatives

FN False negatives

Appendix B

Contents of enclosed CD

readme.txt	Contents description
experiments	Directory of performed experiments
html	Exported .ipynbs in html.
results	Results of hyperparameter tuning
text	Text of thesis.
text.pdf	Pdf version of thesis.