Bridging the Gap: Enhancing Loan Approval Processes with Machine Learning and Soft Information Integration

Author: M.R. Weenink
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands

June 27, 2024

Abstract,

The incorporation of machine learning algorithms into financial technology has drastically revolutionized lending procedures by combining soft and hard information. However, the use of soft information such as human characteristics, company prospects, and subjective assessments by loan officers is still underexplored. This study investigates the use of machine learning techniques, notably Support Vector Machines (SVM), to improve loan approval procedures by exploiting soft data. The study uses data from a microfinance organization including both hard and soft information gathered through direct borrower contacts. The data were preprocessed and features were engineered with Term Frequency-Inverse Document Frequency (TF-IDF). Various SVM models with different kernels were trained and assessed for accuracy and the Matthews Correlation Coefficient (MCC), with the model using the linear kernel achieving the highest MCC of 0.726. The findings demonstrate that integrating soft information into SVM models enhances the accuracy of loan disbursement forecasts. Feature extraction revealed that business-related features were the most influential in the model's decisions, followed by financial and entrepreneur features. A rigorous comparison of actual loan disbursements with model predictions showed that incorporating soft information improves prediction accuracy and reduces repayment difficulties. These findings emphasize the importance of soft information in credit assessments and demonstrate the efficacy of SVM models in real-world lending scenarios, thereby promoting more inclusive and accurate credit evaluation processes.

Graduation Committee Members:

First supervisor: PhD candidate F. Koefer
Second supervisor: Prof. Dr. M. Ehrenhard

Keywords

Financial technology, Soft information, Machine learning, Credit risk assessment, Support vector machine, Loan approval

---

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

CC-BY-NC

1 Introduction

The landscape of lending has been fundamentally transformed by advancements in financial technology, particularly through the integration of machine learning algorithms. This integration promises improved decision-making by incorporating both soft and hard information (Y. Li, 2024). However, the integration of soft information—non-quantitative data about personal traits, business potential, and subjective assessments by loan officers—into the loan approval process, when combined with machine learning algorithms, still lags behind (Chen et al., 2013).

By collecting soft information, engaging with customers applying for loans, and listening to their explanations, financial inclusiveness can be achieved. This approach, known as relationship lending, involves maintaining close, personal interactions with borrowers to gather soft information (Berger and Udell, 1995). Loan officers provide soft information through evaluations that assess various aspects of the businesses seeking loans.

To process these evaluations, we explore the use of machine learning techniques for making informed loan approval decisions. Following several preparatory steps, we test a machine learning model and then conduct a comprehensive analysis.

1.1 Problem Statement

Research on the process of integrating soft information with the lending process is limited, with few studies having explored how it can systematically be combined with hard information to enhance loan approval accuracy. The challenge lies in the subjective nature of soft information, whose relevance is highly context-dependent (Z. Wang et al., 2020), and in the difficulties associated with standardizing and quantitatively incorporating this data into machine learning models (Liberti and Mian, 2009). This knowledge gap has hindered the development of more holistic and inclusive lending models that leverage the full spectrum of available information to make lending decisions (Berger and Frame, 2007).

Most existing literature focuses on the application of hard information, which includes quantitative measures that are easy to process. Attigeri et al. (2017) present a comparison of various machine learning techniques used to evaluate credit risk based on hard information, and a study by Bao et al. (2019) proposes a strategy for integrating machine learning techniques to improve the performance of credit scoring models based on hard information. While this approach is objective and scalable, it fails to capture the nuanced understanding of the potential and risk posed by the borrower. The elaborate contextual perspectives provided by soft information offer a much more holistic view of the credit evaluation process, especially for small and medium-sized enterprises and individual borrowers who would otherwise be limited by purely quantitative metrics (D. Campbell et al., 2019).

1.2 Research Question

The purpose of this research is to integrate machine learning algorithms to assess and utilize the soft information provided by lenders to enhance the accuracy of lending approval processes. The objective is to determine whether machine learning methods can effectively leverage non-quantitative, soft information in loan assessments. This study combines the analytical capabilities of machine learning to process large datasets and make predictions (Sharifani and Amini, 2023), with the rich, contextual insights provided by soft information (Chen et al., 2013).

The following research question has been formulated to achieve this objective:

To what extent can machine learning models leverage soft information provided by loan officers to improve loan disbursement and repayment outcomes?

In the remainder of this thesis, the theoretical framework supporting the research question will be detailed, followed by the methodology used to conduct the study. Subsequently, the results will be presented and discussed. The paper will conclude with a summary of the findings and recommendations for further research.

1.3 Contributions

Incorporating soft information into credit risk assessment presents significant intellectual challenges for traditional models, which have historically relied on objective, numerical data. Sole reliance on hard information overlooks the complex potential and risks associated with borrowers. Several studies have demonstrated how the combination of soft and hard information can improve loan acceptance accuracy (Chen et al., 2013; Cornée, 2019; Z. Wang et al., 2020). The primary challenge lies in the systematic integration of both types of data. The subjective nature of soft information, which varies across contexts, complicates its standardization and inclusion in machine learning models. Addressing this knowledge gap offers an opportunity for academic research to transform credit assessment procedures by incorporating a broader range of borrower data (Das and Chen, 2007; Filomeni et al., 2021; Goddard et al., 1999; Liberti and Petersen, 2019; Uchida et al., 2012).

This research contributes to the financial technology literature by demonstrating the value of integrating soft information into credit models, thus enriching our understanding of its impact on loan assessment. It highlights how machine learning models can effectively incorporate soft information, advancing the integration of machine learning in financial technology.

The practical implications of this research are significant for financial institutions aiming to refine their loan approval processes. By identifying efficient machine learning approaches to evaluate soft information, lending institutions can foster a more dynamic and responsive lending environment. This comprehensive approach enables more informed decisions by considering both soft and hard information, thus providing a complete picture of an applicant's risk profile and potential (Liberti and Petersen, 2019; Uchida et al., 2012). This could potentially reduce the rate of repayment issues (Agarwal and Hauswald, 2010) and enhance the inclusivity and fairness of lending practices (Bouwens and Kroos, 2019; Cornée, 2019). Such methods help to democratize credit access for small and medium-sized enterprises and individual borrowers, who have often been overlooked by traditional credit scoring algorithms.

2 Theoretical Framework

A comprehensive literature review on relevant topics is essential to develop an effective research methodology for addressing the research question formulated above. This literature review will cover theories on hard- and soft information, the relation between soft information and credit evaluations, soft information criteria and the usage of machine learning models in credit evaluations.

2.1 Soft and Hard Information Classification

In financial literature, a distinction is made between soft and hard information. Hard information is measurable, impersonal, and easy to store, while soft information comprises opinions and words, such as projections of market or economic trends (Liberti and Petersen, 2019). Hard information is quantitative and straightforward, whereas soft information requires contextual understanding, making its interpretation subjective and potentially unreliable when transformed into hard metrics, like credit scores (Liberti and Petersen, 2019). Although hard information is often treated as a clear-cut category, experts, including Liberti and Petersen (2019) and earlier studies by Goddard et al. (1999), argue that it exists on a continuum with soft information. Despite this, most research tends to classify them nominally into two distinct groups. The methods for collecting hard and soft information differ significantly. Hard information can be gathered impersonally and is standardized, leading to lower transaction costs and easier processing and transmission. In contrast, soft information requires personal collection and context-specific understanding, making it challenging to verify and transmit. This difference results in hard information being more durable and easier to maintain, whereas soft information incurs higher transaction costs and potential information loss when attempting to harden it (Flögel, 2018; Liberti and Petersen, 2019).

2.2 Soft Information Criteria

An empirical article on soft information value that is specifically done for a Taiwanese finance company as part of the study of Chen et al. (2013), identifies several types of soft information that are helpful for predicting loan defaults:

1. Employee Loyalty and Satisfaction: Firms with high employee loyalty and satisfaction are less likely to default. For direct evidence, it means that employee loyalty and satisfaction, in all probability representing a sounder organizational culture and better operational stability, turn into soft information of considerable value for lenders.

2. Long-Term Customer Relationships: The existence of long-term customers also showed a negative relationship with default rates. This means the long-lasting and stable customer relationships held by businesses may be having a less risky and portend-positive pictures of the credit standing.

3. Loan Officers' Subjective Judgment on Borrowers' Leverage and Profitability: It means that loan officers' subjective judgments, in respect to leverage and profitability, made significant adjustments to the scores based on the financial ratios and were significant predictors of defaults. This suggests that understanding and processing the subjective knowledge of loan officers, such as a borrower's financial standing beyond what is evident from financial statements, could be crucial soft information.

4. Management Skills and Business Strategies: The nature of business strategies and management skills is of utmost importance to the owner and CEO of the borrowing firm. Therefore, the assessment of such attributes by a loan officer might provide some insight regarding the firm's future prospects and ability to combat adversity, thereby impacting the likelihood of loan default.

Another paper by Gabbi et al. (2020) states that empirical analysis on small and medium enterprises shows that strategic perspectives, approximated through soft variables, include managerial skills, business diversification, vision, competition intensity, quality accreditation, bank-firm relationship, and borrowing requirements. Such soft variables are feasible to estimate the risk and further the survival probability of firms.

Kaplan and Norton (1992) developed a balanced scorecard model that considered both hard and soft information research variables. According to the balanced scorecard model by Kaplan and Norton (1992), the assessment of firm strategy includes but is not limited to the following perspectives: (a) customers; (b) business processes; (c) financial; (d) learning and growth.

The perspective of the customers concentrates on the firm's ability to meet customers' needs. An efficient quality management system aims to offer better products and services while improving process efficiency according to Flynn et al. (2007), Kaynak (2003), and Samson and Terziovski (1999). The perspective of business processes concentrates on enhancing fundamental business processes. Key business processes including product innovation, the distribution process, and after-sales support should all be optimized by the company. According to Hamilton and Fox (1998), the financial perspective examines the financial equilibrium that is taken into account to determine whether the company can consistently guarantee a sufficient level of financial coverage. The firm's mindset when establishing improvement targets for its capacity to carry out the operations that generate value for clients and other stakeholders is related to the learning and growth perspective.

2.3 Risks and Benefits of Soft Information in Credit Risk Assessment

There is limited knowledge on how soft information directly affects the credit approval process. Some studies, such as Lipshitz and Shulimovitz (2007), investigate the role of loan officers' intuition in their decisions, highlighting that financial data is merely the starting point, and personal impressions and intuition developed through interactions and observations play a crucial role.

Integrating soft information, such as credit analysts' subjective opinions, greatly improves the quality of credit rating forecasts as compared to models based purely on hard information. This empirical investigation found that models incorporating soft information were better at predicting defaults (Lehmann, 2003).

A survey of Portuguese banking specialists underlined the relevance of soft information variables in credit risk assessment, such as managerial experience and reliability, which showed strong negative correlations with bank default rates (Soares et al., 2011).

A study investigating small enterprises' qualitative actions to predict criticalities in the Italian credit market found that including soft information aspects into hard information rating processes improves credit rating accuracy (Gabbi et al., 2020).

Gavalas and Syriopoulos (2014) presents an integrated credit rating model that includes both hard and soft information elements to help banks make lending decisions. The findings suggest that incorporating soft information increases the accuracy and reliability of credit ratings.

These sources show that incorporating soft information considerably enhances the forecast quality of credit ratings, providing useful insights for financial institutions and strengthening credit risk assessment models.

However, the use of soft information in credit risk assessment can introduce significant challenges due to its subjective and manipulable nature. Soft information, unlike hard information which is quantitative and verifiable, relies heavily on qualitative assessments and personal judgments, which can be easily influenced or biased. For instance, in the study by Godbillon-Camus and Godlewski (2005), it was shown that while soft information can provide a more precise estimation of debtor quality, it also creates opportunities for credit officers to manipulate the information to their advantage, potentially leading to improper risk evaluations (Godbillon-Camus and Godlewski, 2005). Furthermore, the reliance on soft information can lead to inconsistencies in credit decisions, as different credit officers may interpret the same qualitative data differently, reducing the reliability of the risk assessment process (Acheampong and Elshandidy, 2021). Additionally, incorporating soft information requires significant organizational changes and specific incentive structures to mitigate manipulation risks, which can be complex and costly to implement effectively (Cornée, 2019).

2.4 Risks and Benefits of Machine Learning in Credit Risk Assessment

Caridad et al. (2019) describes in his study how AI approaches can be used to combine soft information from public sources with economic and financial data to provide precise credit rating forecasts. The study demonstrates that such integration greatly improves forecast quality. A research using machine learning to generate a comprehensive measure of credit risk using soft information from conference calls and management discussion sections conducted by Donovan et al. (2021), found that soft information considerably improves credit event prediction. Machine learning approaches auto-

mate loan approval by analyzing a large dataset of previous loan applications that contains both financial and personal information. Integrating soft information considerably improves prediction accuracy (Diwate et al., 2021). Evaluating the effectiveness of various data science methods, such as feature selection and data resampling, highlights that correlation-based feature selection and random forest classifiers deliver superior performance when incorporating soft information (Ziemba et al., 2021). Deconstructing loan officers' choices into components driven by hard and soft information indicates that machine learning models can surpass loan officers in hard data processing while effectively capturing vital soft data (M. Liu, 2022). These resources provide a thorough overview of how machine learning algorithms can successfully integrate soft information to enhance the accuracy and reliability of loan assessments.

S Nair et al. (2015) used a hybrid technique combining rule-based and machine learning to undertake an experiment on sentiment analysis of Malayalam cinema reviews. Conditional Random Field (CRF) and a support vector machine (SVM) model were employed in this investigation with a strategy that combines each with a rule-based approach, and it was discovered that SVM outperformed CRF with the best accuracy of 91%. Cortes and Vapnik (2009) introduces SVMs and outlines its theoretical underpinnings, establishing SVMs as powerful tools for classification tasks, including those with high-dimensional feature spaces typical of text data. According to multiple researchers, Hsu (2020), Islam and Sultana (2018), Kanakaraddi et al. (2020), and Y. Liu et al. (2017), SVM models perform the best for comparable cases among multiple machine learning algorithms like decision tree and naïve Bayes.

Machine learning techniques, despite their advanced capabilities, present several limitations when applied to credit risk assessment. One major issue is the "black box" nature of many machine learning models, which makes it difficult to interpret and understand the decision-making process. This lack of transparency can be problematic for regulatory compliance and for gaining trust from stakeholders (Noriega et al., 2023). Additionally, machine learning models often require large amounts of high-quality data to perform accurately. However, financial datasets can be noisy, incomplete, or imbalanced, which can significantly impact the model's performance (Attigeri et al., 2017). Moreover, the implementation of machine learning models necessitates substantial computational resources and expertise, making it a costly and time-consuming process. There are also concerns about overfitting, where models perform well on training data but fail to generalize to unseen data, leading to inaccurate risk assessments (Suhadolnik et al., 2023). Lastly, while machine learning can enhance predictive accuracy, it often does so at the expense of model interpretability and simplicity, which can complicate its integration into existing risk management frameworks (D. Li, 2023).

2.5 Hypotheses

Based on the literature review, there are two relevant hypotheses.

2.5.1 Hypothesis 1: Soft information contains a significant relevance in credit risk assessment

Soares et al. (2011) underlines the relevance of soft information variables in credit risk assessment. Studies by Lehmann (2003), Gabbi et al. (2020), and Gavalas and Syriopoulos (2014) all show that incorporating soft information considerably enhances the forecast quality of credit ratings. Therefore, I hypothesize that soft information contains value for credit risk assessments.

2.5.2 Hypotheses 2: Support vector machine models have significant predictive power in credit risk assessment

Donovan et al. (2021) states that a comprehensive measure of credit risk generated by machine learning using soft information considerably improves credit event prediction. Ziemba et al. (2021)'s study shows higher performances of several data science strategies using soft information. Other studies have found that the integration of soft information into machine learning, using hard information, results in improved prediction accuracy (Diwate et al., 2021). A study conducted by M. Liu (2022) illustrates that machine learning can even outperform loan officers in processing hard data while gathering valuable soft data. The studies of Burges (1998), Cortes and Vapnik (2009), Cristianini and Shawe-Taylor (2000), and S Nair et al. (2015) show the Support Vector Machine model to be suited for our dataset. Multiple studies conducted by Hsu (2020), Islam and Sultana (2018), Kanakaraddi et al. (2020), and Y. Liu et al. (2017) show an SVM model outperforming other machine learning models in comparable cases. Therefore, I hypothesize that a support vector machine model can have significant predictive power in credit risk assessment.

3 Methodology

3.1 Research Design

In order to measure to what extent machine learning could be effective in predicting loan application ap-

proval, we used different parameters for the Support Vector Machine (SVM) model and tested which one would work best. We first had to install Python and Visual Studio Code to run the codes. We started with the most basic default SVM model. Later, we extended our research by adjusting various parameters, hoping to find the best combination of parameters to predict loan approval. After engineering our SVM model, we conducted feature extraction to analyze which features contribute most to the decision-making process of the SVM model. Besides the engineering of our SVM model and feature extraction, we performed a descriptive analysis of the actual data and the predictions made by the SVM model.

3.2 Data Collection

The data used in this research is provided by a non-profit micro-finance institution (MFI). Its mission is to increase financial inclusiveness by providing access to credit for small entrepreneurs who are underserved by traditional banking systems. The organization aims to help these entrepreneurs start and grow their businesses by offering not only loans but also training sessions and coaching services. The MFI is dedicated to serving clients on ideological grounds, aiming to fill the gap left by traditional banks and promoting entrepreneurship among those with limited access to financial resources. The MFI has an average loan size of around €21,000, which is relatively small. This focus on micro-loans helps support small business ventures that often lack sufficient guarantees or financial data to obtain credit from traditional banks. These clients are typically considered too small or too risky by larger financial institutions (Guo et al., 2014; Orser et al., 1994).

The MFI employs relationship lending, where personal attention and service are emphasized. This involves direct interaction between loan officers and applicants to gather in-depth soft information, helping to assess the true potential and risks associated with each loan application. When the application passes the initial assessment at the application intake, a loan officer conducts an interview with the applicant. This phase involves gathering both hard and soft information to form a comprehensive view of the applicant's business and entrepreneurial capabilities, summarized in an evaluation (Berger and Udell, 1995).

These evaluations are the data used in our ML model, containing textual data divided into the following sections: Loan Officer Conclusion, Risk Manager Conclusion, Business Activities, Entrepreneur Description, Market Description, Explanation Private, and Explanation Financial Analysis.

Other variables used in the data are "Disbursed" or "Not to be disbursed" as labels in our ML model. For the comparison of the ML model's predictions with the actual data, the following other variables were used: Capital requirement for the business, Amount disbursed, Credit Score, Repayment Issues, Face-to-face meeting, Age, Gender, Nationality, Education, and Marital Status.

3.3 Data Preprocessing

In order to use the data in our machine learning model and for data analysis, a preprocessing process had to be conducted. The preprocessing of the data involves cleaning and transforming the data to make it suitable for support vector machines and data analysis. The preprocessing consists of the following aspects: text aggregation, cleaning, tokenization, normalization, removing questions, and lemmatization, according to Hapke et al. (2019) and others like Yang and Cui (2021).

The preprocessing will be performed using pre-built libraries and tools in Python.

3.4 Feature Engineering

A statistical measure called Term Frequency-Inverse Document Frequency (TF-IDF) is used to assess a word's significance to a document within a collection or corpus, according to Manning et al. (2009). A word's relevance increases in direct proportion to its frequency in the document, but this is offset by the word's frequency across the corpus. Text mining and information retrieval extensively use this technique. Inverse Document Frequency (IDF) measures the importance of the term within the entire corpus. The idea is that if a word is common and appears in many documents, it is likely not a good discriminator. The IDF of a term is calculated as the logarithm of the number of documents divided by the number of documents that contain the word. TF-IDF can be extended to consider word sequences, or n-grams, in addition to single words, known as unigrams. By considering phrases or word combinations, this method takes into account the context in which terms appear and can enhance the functionality of text mining and information retrieval applications. Unigrams are single words; bigrams are sequences of two adjacent words, capturing pairs of words that often appear together, providing more context than unigrams. Trigrams are sequences of three adjacent words, offering even richer context. By using n-grams, TF-IDF better captures the structure and meaning of the text (Manning et al., 2009).

When paired with Support Vector Machines (SVMs), TF-IDF creates a potent technique especially well-suited for text classification tasks such as senti-

ment analysis, document classification, and spam detection. TF-IDF is a preprocessing technique that is supported by both theoretical knowledge and practical validation. Its purpose is to convert text data into a numerical format that SVM can process. This narrative and its sources describe how TF-IDF and SVM work in concert. Joachims (1998) provides empirical evidence supporting the use of SVM for text categorization. It specifically highlights the effectiveness of SVM in handling high-dimensional feature spaces created by TF-IDF, demonstrating superior performance in text classification tasks. Therefore, we use the TF-IDF feature engineering technique in combination with the SVM model. The feature engineering will be conducted using pre-built libraries and tools in Python.

3.5 Support Vector Machine Model Construction

Different SVM models will be tested to find the best working model for predicting loan approval. The vector machine models will differ based on kernel choice and parameter settings. Various kernels will be tested and evaluated, and parameter optimization will be conducted to find the most effective method. The SVM model will be built using pre-built libraries and tools in Python.

Before training the model, the dataset will be split into training and test sets. The split ratio used is 80% for training and 20% for testing. The training process involves feeding the model with labeled data so it can learn to predict defaults. The testing process involves comparing the model's predictions with the data's labels. To ensure the robustness and generalizability of our machine learning models, we employ a technique called 5-fold cross-validation. Both the training and testing of the model will be conducted using pre-built libraries and tools in Python.

The models will be evaluated based on accuracy, known as the percentage of accurately predicted instances among all of the dataset's instances, and the Matthews Correlation Coefficient (MCC), which takes into consideration all four categories of the confusion matrix and scales from -1 to 1, where 1 denotes flawless prediction, 0 a random guess, and -1 a complete discrepancy between the predicted and actual results (Chicco et al., 2021). A comprehensive analysis of the model's performance will be provided via a confusion matrix:

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	TN	TN

Table 1: Confusion Matrix

This matrix provides insights for improvement by assisting in the identification of specific error categories (*Data Mining*, 2011).

Although accuracy offers a straightforward and understandable way to gauge performance, it can be deceptive in datasets that are unbalanced and have a notably higher frequency of one class than the other. The MCC value provides a balanced measure that can be applied in cases where the classes have significantly different sizes. The confusion matrix offers a thorough analysis of the model and provides insights into areas that require development by helping to pinpoint the exact types of mistakes the model is making. The evaluation metrics will be calculated using pre-built libraries and tools in Python.

3.6 Feature Extraction

Based on the literature review, the features significantly contributing to the decision-making of the SVM model will be linked to one of the important criteria stated in prior literature.

The features with a significant contribution to the decision-making will be filtered using a threshold of one standard deviation added to the mean. This threshold filters the significant features and helps to avoid overly simplistic decision boundaries, creating a more conservative decision boundary and reducing false positives.

The feature weights are extracted from the SVM model using pre-built libraries in Python. The significant features are then manually linked. The criteria containing the different significant features are then compared based on the number of features within the categories and their average and combined weights, stating their importance to the decision-making process.

3.7 Variables

A descriptive analysis is conducted on the dependent variables of loans disbursed, not disbursed, disbursed with repayment issues, and disbursed without repayment issues. For the descriptive analysis, the dataset initially uses the actual data indicating whether a loan was disbursed or not. For the analysis of predicted outcomes, this actual data is replaced with the model's predictions.

The independent variables in our dataset consist of Capital Requirement for Business, Amount Disbursed, Credit Score, Repayment Issues, Face-to-Face Meeting, Age, Gender, Nationality, Education, and Marital Status, as stated in the data collection paragraph. The correlations between the status of the loans (disbursed or not, and with or without repayment issues) and the independent variables are compared. The correlation is found by calculating the distribution of the independent variables within the dependent variables.

4 Results

4.1 Model Performance

Kernel:	MCC:
Linear	0.607
RBF	0.716
Sigmoid	0.722
Poly	0.064

Table 2: Model Performances

Table 2 shows three different models, each using a different kernel. The parameters of the models are all set to default: C set to 1, Gamma to scale, Probability to False, and Class Weight to none. The table contains, besides their kernel usage, the MCC evaluation metrics that indicate the predictive power of a machine learning model. The MCC value is a number between -1 and 1, where 1 denotes a perfect prediction, 0 denotes no prediction, and -1 denotes a negative prediction. We see that the model using the Sigmoid kernel achieves the highest MCC, indicating the highest predictive power, closely followed by the model using the RBF kernel. The model using the Linear kernel achieves a sufficient MCC score of 0.607, indicating a certain predictive power of the machine learning models. Lagging behind is the model using the Poly kernel, with an MCC close to zero, meaning it has small predictive power.

Kernel:	C:	MCC	Accuracy
Linear	0,1	0.726	0.937

Table 3: Best Performing Model

After performing parameter optimization for the promising models with the Linear, RBF, and Sigmoid kernels, we found the best performing model in terms of MCC to be the Linear SVM with the regularization parameter C set to 0.1.

	Actual Disbursed:	Actual Not Disbursed:
Predicted Disbursed	2408	14
Predicted Not Disbursed	166	246

Table 4: Confusion Matrix of the best performing model

The confusion matrix in table 4 shows the performance of the best-performing model in terms of MCC on the test dataset, providing a visualization of its performance. The visualization indicates that the model correctly predicted 2408 samples to be disbursed while incorrectly predicting 14 samples to be disbursed. Additionally, the model correctly predicted 246 samples to not be disbursed while incorrectly predicting 166 samples to not be disbursed.

4.2 Feature Extraction Weights

The weights assigned to the features containing the TF-IDF by the Support Vector Machine using the Linear kernel indicate the importance of those features. I was unable to manually link the features to the criteria stated in prior studies due to the nature of the features. Therefore, we have categorized the features into somewhat comparable categories regarding finance, entrepreneur, and overall business. We have analyzed these categorized features to determine their importance in the decision-making process of the SVM model.

Categories:	Number of Features:	Average Weight	Combined Weight
Financial	49	0.338	16.571
Entrepreneur	44	0.332	14.603
Overall Business	62	0.354	21.913

Table 5: Positive Feature Categories

Table 5 shows the number, average weight, and combined weight of the categories for the positive significant features. We can conclude from this data that the category of business-related features has more influence on the decision-making of our model, with its higher average weight, combined weight, and number of features. The category of financial features has a slightly higher positive influence on decision-making compared to the category of entrepreneurial features.

Categories:	Number of Features:	Average Weight	Combined Weight
Financial	7	-0.200	-1.397
Entrepreneur	11	-0.216	-2.372
Overall Business	11	-0.194	-2.129

Table 6: Negative Feature Categories

Table 6 shows the same analysis but for the negative significant features. We see that the features within the entrepreneur category have a slightly higher influence on decision-making, with higher combined and average weights. The features of the financial category have the least negative influence on decision-making, with a lower number of features and lower average and combined weights.

4.3 Data Analysis

A comprehensive descriptive analysis was conducted on the dataset to examine the correlations between various variables. The dataset comprised 14,166 samples after data processing. The samples were categorized based on loan disbursement status (disbursed or not disbursed) and, within the disbursed category, further divided into those with or without repayment issues. The actual results of the data analysis can be found in the appendices (see Appendix Table 7).

Comparing disbursed to non-disbursed samples, a slight difference in credit scores was observed, with disbursed loans having a higher average credit score. Several borrower characteristics were also found to influence loan disbursement: younger age, being female, and holding Dutch nationality all had a positive impact on loan approval.

Education (see Appendix Table 8) and marital status (see Appendix Table 9) had minimal impact on loan disbursement. However, possessing a high school or university education or living together with a partner or having a registered partnership negatively influenced loan disbursement. The main income source (see Appendix Table 10) also showed little variation. The loan product (see Appendix Table 11) significantly influenced loan approval, with small and medium-sized enterprise (SME) loans facing a substantial negative effect.

When comparing disbursed loans with and without repayment issues (see Appendix Table 7), notable insights emerged. Loans disbursed without repayment issues had an average credit score half a point higher than disbursed loans with repayment issues, indicating that a higher credit score generally leads to fewer repayment problems. The average amount disbursed was also significantly higher for loans without repayment issues. Additionally, having no benefit, being male, and holding Dutch nationality correlated with fewer repayment issues.

The entrepreneur's education level (see Appendix Table 8) did not significantly affect repayment issues. However, marital status (see Appendix Table 9) did: living alone correlated with more repayment issues, while being married in community of property correlated with fewer repayment issues. In terms of the loan product (see Appendix Table 11), SME loans experienced fewer repayment problems compared to the company loan product and the EZK loan product.

From this analysis, we conclude that the SME loan product negatively impacts loan approval. Additionally, male entrepreneurs and those with a stable marital status tend to have fewer repayment issues. Notably, the SME loan product, while disbursed less frequently, tends to result in fewer repayment issues.

4.4 Anlysis of the predicted data

The same descriptive analysis was conducted on the dataset using the predictions generated by the SVM model. The samples were categorized based on whether loans were disbursed or not, and among the disbursed loans, further categorized based on the presence or absence of repayment issues. The actual results of the data analysis can be found in the appendices (see Appendix Table 12).

The analysis (see Appendix Table 12) revealed a slight difference in credit scores, with disbursed loans having a higher average credit score. Additionally, several borrower characteristics were found to positively influence loan disbursement: younger age, receiving benefits, being female, and holding Dutch nationality.

Education (see Appendix Table 13) and marital status (see Appendix Table 14) had minimal impact on loan disbursement, with the exception that cohabitation positively affected loan disbursement, while living alone had a negative impact. High school education negatively influenced loan disbursement compared to lower education levels. The main income source showed little variation. The loan product (see Appendix Table 16) significantly influenced loan approval, with small and medium-sized enterprise (SME) loan products experiencing a substantial negative effect, in contrast to the company loan product.

When comparing disbursed loans with and without repayment issues (see Appendix Table 12), several notable insights emerged. Loans without repayment issues had a slightly lower average credit score. The average amount disbursed was higher for loans without repayment issues. Furthermore, loans without repayment issues were more often associated with male entrepreneurs

who received benefits and had Dutch nationality. Higher age also correlated with fewer repayment issues.

Examining the educational level of entrepreneurs (see Appendix Table 13), it was observed that higher education levels correlated with fewer repayment issues. Similarly, living together with a partner or being married correlated with fewer repayment issues, whereas living alone correlated with more repayment issues (see Appendix Table 14). Entrepreneurs with a primary income from their enterprise or job faced fewer repayment issues (see Appendix Table 15). The loan products (see Appendix Table 16) of SME and EZK were associated with fewer repayment issues compared to the company loan product, which correlated with more repayment problems.

This analysis of the predictions of disbursed and not disbursed loans concludes that a loan product for small-sized enterprises negatively impacts loan disbursements. Conversely, Dutch nationality has a positive effect on loan disbursement. Entrepreneurs receiving benefits or being male tend to face fewer repayment issues. Cohabitation or marriage correlates with fewer repayment issues, while living alone correlates with more. Entrepreneurs with a primary income from their enterprise or job face fewer repayment issues. Finally, the loan products of SME and EZK result in fewer repayment issues despite fewer loans being disbursed, whereas loans with the company loan product are more prone to repayment problems.

4.5 Comparison of the two analyses

Comparing the descriptive analysis of the actual data with the analysis of the data replaced by the model's predictions reveals several noteworthy insights. The model predicts a higher number of loan disbursements compared to the actual data, with only 9% predicted as not disbursed, against 14% in the actual data. This suggests that the SVM model using a linear kernel has a higher acceptance rate. Additionally, 50% of the actual disbursed loans were without repayment issues, whereas the model predicts 53% of the disbursed loans to be without repayment issues, indicating improved predictive accuracy regarding the potential for repayment issues.

Furthermore, the model results in fewer loans disbursed to entrepreneurs receiving benefits or those with non-Dutch nationality compared to actual disbursements. Conversely, entrepreneurs not receiving benefits are more frequently predicted to be disbursed loans, and these loans are less likely to encounter repayment issues. There are minimal differences in the distribution of education level, marital status, and main source of income between the actual and predicted disbursements, both with and without repayment issues. However, the loan product for small to medium-sized enterprises shows an even greater negative impact on the prediction of loan disbursements compared to the actual data.

Overall, these findings suggest that the model shows a slight improvement in predicting loans without repayment issues. It also highlights certain demographic and loan product biases in loan disbursement predictions, emphasizing the need for careful consideration of these factors in predictive modeling.

The existing literature highlights the importance of soft information in credit assessments. Studies emphasize that soft information, such as borrower characteristics and subjective judgments, provides nuanced insights that hard information alone cannot capture. The SVM model's high accuracy in predicting loan disbursements when soft information is included demonstrates the practical value of these qualitative data points. This aligns with the hypothesis that soft information is critical for accurate loan assessment. Therefore, I conclude my first hypothesis stating that soft information contains significant relevance in credit risk assessment to be true.

The results show that the SVM model may efficiently use soft information provided by loan officers. The significant weights assigned to soft information features indicate that the SVM model uses these inputs to predict loan evaluations. The SVM model's predictions, which included soft information, had a strong correlation with actual disbursement results. This confirms the model's capacity to successfully use soft information. Thus, the SVM model's ability to forecast loan evaluations based on soft information emphasizes its usefulness in real-world credit evaluation settings. Therefore, I conclude my second hypothesis stating that Support Vector Machine models have significant predictive power in credit risk assessment to be true.

5 Discussion

The value of soft information in credit risk assessment is well-documented in the literature. Studies such as those by Liberti and Petersen (2019) and Chen et al. (2013) emphasize that soft information provides nuanced insights that hard information alone cannot capture. The results demonstrate that soft information significantly influences loan disbursement and repayment outcomes with the model's performance. The credit assessment of the machine learning model using soft information as variables is sufficient, stating the importance of soft information in credit assessment. This aligns with the literature's emphasis on the importance of soft information in credit assessments, validating its critical role in

enhancing decision-making. The results also show significant positive and negative weights for features derived from soft information categories. These findings demonstrate that soft information substantially influences the SVM model's decision-making process, affirming its critical value in credit assessment.

The literature extensively discusses the benefits of using machine learning models in credit risk assessment, highlighting their ability to process large datasets and uncover complex patterns that traditional methods might miss. Studies by Caridad et al. (2019) and Donovan et al. (2021) suggest that machine learning models should effectively integrate both hard and soft information to improve prediction accuracy. The integration of soft information into the SVM model, as reflected in the results, indicates that the model can effectively utilize soft information provided by loan officers. The significant weights assigned to soft information features confirm that the SVM model leverages these inputs to predict loan evaluations. The SVM model's predictions, which incorporated soft information, showed a high correlation with actual disbursement outcomes. This confirms the model's capability to effectively utilize soft information. This performance aligns with the literature, which suggests that machine learning models, when properly trained and tuned, can significantly enhance credit risk assessments by leveraging both quantitative and qualitative data.

A major concern in the literature regarding machine learning models is the "black box" nature, which makes it difficult to interpret and understand the decision-making process (Noriega et al., 2023). This lack of transparency can be problematic for regulatory compliance and stakeholder trust. However, our approach includes feature extraction with their weights, which provides insight into the model's decision-making criteria. By identifying the significant features and their respective weights, we address the black box issue to some extent, aligning with the literature's call for more interpretable and transparent machine learning models in financial applications.

Due to the nature of the features, I was unable to manually link the significant features to the criteria stated in prior literature. While our findings on important feature weights do not fully align with the literature, they offer valuable insights into the model's decision-making process. The literature, such as studies by Kaplan and Norton (1992) and Gabbi et al. (2020), suggests certain soft information criteria as critical, but our feature weight extraction showed different influences. For instance, business-related features had the highest positive influence, whereas entrepreneur features had the highest negative impact. This discrepancy highlights the complexity of integrating soft information into machine learning models. Our approach to feature weight extraction, however, provides a practical method to understand and interpret the contributions of different types of information in credit risk assessment.

5.1 Theoretical Implications

This study's findings challenge standard ideas that concentrate on hard information in loan approval processes, indicating that including soft information in machine learning (ML) models significantly improves forecast accuracy. This contributes to the financial technology and credit risk assessment literature streams. Our findings align with and extend the work of scholars such as Gavalas and Syriopoulos (2014), who have explored the importance of the integration of soft information in credit evaluations. Similarly, Udell and Berger (2002) have emphasized the relevance of soft information in small business financing, underscoring the potential benefits of our research.

It lends weight to the developing belief that combining hard and soft information provides a more holistic assessment of creditworthiness, contributing to the behavioral finance literature stream. Furthermore, this study establishes the framework for future research to investigate and develop these methodologies, thereby improving the broader area of credit risk assessment. By demonstrating the efficacy of incorporating soft information in ML models, our research provides a foundation for scholars like Caridad et al. (2019) and Donovan et al. (2021), who study venture capital decision-making processes, to enhance their models and predictions.

5.2 Practical Implications

The findings of this study have important practical implications for the financial industry, particularly in improving the loan approval process. By using machine learning models that leverage soft information, lending institutions can change their decision-making processes, resulting in a more inclusive loan assessment process. The incorporation of soft information into machine learning models may improve the ability to evaluate loan applications more accurate.

Using machine learning models in loan assessments provides the possibility to decrease the bias inherent in manual evaluations performed by loan officers. Human decision-makers may unintentionally add personal biases due to subjective judgments or lack of expertise. In contrast, machine learning models handle data objectively, ensuring that loan decisions are made based on a thorough review of both hard and soft information. This results in more equitable lending practices.

The use of machine learning models can greatly reduce the operational costs involved in the loan approval process. Loan officers' manual reviews take a long time and require significant resources. By automating this process with machine learning algorithms, banks and financial institutions can process a larger volume of loan applications more efficiently. This cost reduction might be especially useful for smaller financial institutions that may not have as many resources available.

5.3 Future Research

Future research should examine a broader range of machine learning models to see if other approaches perform better at utilizing soft information. Investigating advanced models such as deep learning or ensemble approaches may yield significant insights.

Investigating various feature engineering strategies may improve the processing and exploitation of soft information. Word embedding, sentiment analysis, and more advanced natural language processing (NLP) methods should be examined.

Collecting more extensive datasets with an equal number of positive and negative samples might increase the models' generalizability. Having a balanced sample of loan applications can help construct more robust predictive models.

The development of standardized techniques for collecting and evaluating soft information can help eliminate subjectivity and bias. Creating criteria for loan officers to follow while reviewing applications can result in more consistent and objective data for model training.

Reevaluating and restructuring loan approval processes to better incorporate machine learning techniques can result in more efficient and equitable decision-making. Future studies should investigate how organizational changes and process optimizations might aid in the effective use of ML models in lending.

5.4 Limitations

This study has aimed to provide a comprehensive examination of the effectiveness of machine learning algorithms in predicting loan assessments. However, several limitations should be considered when interpreting the results of this study.

There is a possibility of model selection bias, as other machine learning models may have outperformed the one used for this study. Different models could produce different results, affecting the overall findings and conclusions.

The selection of feature engineering approaches may have induced bias. Alternative feature engineering strategies could have been more effective in improving the performance and accuracy of the machine learning models utilized in this study.

The dataset used in this investigation included a small number of negative samples. This imbalance may have an impact on the model's capacity to generalize and properly predict loan disbursements because it has not been exposed to a representative distribution of outcomes.

The soft information utilized in machine learning models is based on loan officers' subjective assessments. These assessments can be skewed, reflecting human judgments that do not always align with objective standards.

The machine learning models used in this work evaluate data by counting words and determining their significance. However, they do not understand the true meaning of the words, which can limit their ability to capture the full context and nuances of the soft information provided.

The validation of our model through cross-validation techniques lends credibility to our findings. The model's predictions showed a high correlation with actual loan disbursement outcomes, reinforcing the reliability of our approach.

However, our data analysis is limited as it primarily calculates the distribution of variables within the disbursed or not disbursed, and repayment issues or no repayment issues categories, without employing more rigorous statistical tests like T-tests. This limitation means that while our findings provide useful descriptive insights, they lack the statistical rigor that could further validate the significance of the observed patterns.

Further validation with diverse datasets, additional evaluation metrics, and more comprehensive statistical analyses would be beneficial to confirm the model's generalizability and effectiveness in various scenarios.

6 Conclusion

This study finds that the integration of machine learning models, particularly Support Vector Machines (SVM), into loan approval processes significantly enhances predictive accuracy when soft information is included. SVM models, especially those with linear kernels, efficiently utilize qualitative data, improving the prediction of loan disbursements and repayment concerns. The findings emphasize the need to combine hard and soft data to develop a more comprehensive and inclusive credit evaluation methodology. This strategy not only improves the decision-making process but also promotes financial inclusion by taking into account the nuances of borrowers' profiles.

7 References

References

Acheampong, A., & Elshandidy, T. (2021). Does soft information determine credit risk? Text-based evidence from European banks. *Journal of International Financial Markets, Institutions and Money*, *75*, 101303. https://doi.org/10.1016/j.intfin.2021.101303

Agarwal, S., & Hauswald, R. (2010). Distance and Private Information in Lending. *The Review of Financial Studies*, *23*(7), 2757–2788. https://doi.org/10.1093/rfs/hhq001

Alpaydin, E. (2004). *Introduction to Machine Learning*. MIT Press.

Attigeri, G., Pai, M., & Pai, R. (2017). Credit Risk Assessment Using Machine Learning Algorithms. *Advanced Science Letters*, *23*, 3649–3653. https://doi.org/10.1166/asl.2017.9018

Bao, W., Lianju, N., & Yue, K. (2019). Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. *Expert Systems with Applications*, *128*, 301–315. https://doi.org/10.1016/j.eswa.2019.02.033

Berger, A. N., & Frame, W. S. (2007). Small Business Credit Scoring and Credit Availability* [Publisher: Routledge _eprint: https://doi.org/10.1111/j.1540-627X.2007.00195.x]. *Journal of Small Business Management*, *45*(1), 5–22. https://doi.org/10.1111/j.1540-627X.2007.00195.x

Berger, A. N., & Udell, G. F. (1995). Relationship Lending and Lines of Credit in Small Firm Finance [Publisher: University of Chicago Press]. *The Journal of Business*, *68*(3), 351–381. Retrieved June 21, 2024, from https://www.jstor.org/stable/2353332

Bouwens, J., & Kroos, P. (2019). The effect of delegation of decision rights and control: The case of lending decisions for small firms. *Management Accounting Research*, *43*, 29–44. https://doi.org/10.1016/j.mar.2018.07.004

Burges, C. J. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167. https://doi.org/10.1023/A:1009715923555

Campbell, C., & Ying, Y. (2011). *Learning with Support Vector Machines*. Springer International Publishing. https://doi.org/10.1007/978-3-031-01552-6

Campbell, D., Loumioti, M., & Wittenberg-Moerman, R. (2019). Making sense of soft information: Interpretation bias and loan quality. *Journal of Accounting and Economics*, *68*(2), 101240. https://doi.org/10.1016/j.jacceco.2019.101240

Camps-Valls, G., Martín-Guerrero, J., Rojo-Álvarez, J. L., & Olivas, E. (2004). Fuzzy sigmoid kernel for support vector classifiers. *Neurocomputing*, *62*, 501–506. https://doi.org/10.1016/j.neucom.2004.07.004

Caridad, D., Hanclova, J., Hosn el Woujoud, B., & Caridad López del Río, L. (2019). Corporate rating forecasting using Artificial Intelligence statistical techniques. *Investment Management and Financial Innovations*, *16*, 295–312. https://doi.org/10.21511/imfi.16(2).2019.25

Chen, Y., Huang, R., Tsai, J., & Tzeng, L. (2013). Soft Information and Small Business Lending. *Journal of Financial Services Research*, *forthcoming*. https://doi.org/10.1007/s10693-013-0187-x

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113–126. https://doi.org/10.1016/S0893-6080(03)00169-2

Chicco, D., Warrens, M., & Jurman, G. (2021). The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2021.3084050

Cornée, S. (2019). The Relevance of Soft Information for Predicting Small Business Credit Default: Evidence from a Social Bank [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jsbm.1231 *Journal of Small Business Management*, *57*(3), 699–719. https://doi.org/10.1111/jsbm.12318

Cortes, C., & Vapnik, V. (2009). Support-vector networks. *Chem. Biol. Drug Des.*, *297*, 273–297. https://doi.org/10.1007/%2FBF00994018

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press. https://doi.org/10.1017/CBO9780511801389

Das, S., & Chen, M. (2007). Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. *Management Science*, *53*, 1375–1388. https://doi.org/10.1287/mnsc.1070.0704

Data Mining: Practical Machine Learning Tools and Techniques. (2011). Elsevier. https://doi.org/10.1016/C2009-0-19715-5

Diwate, Y., Rana, P., & Chavan, P. (2021). Loan Approval Prediction Using Machine Learning. *08*(05).

Donovan, J., Jennings, J., Koharki, K., & Lee, J. (2021). Measuring credit risk using qualitative disclosure. *Review of Accounting Studies*, *26*(2), 815–863. https://doi.org/10.1007/s11142-020-09575-4

Filomeni, S., Udell, G. F., & Zazzaro, A. (2021). Hardening soft information: Does organizational distance matter? [Publisher: Taylor & Francis Journals]. *The European Journal of Finance*, *27*(9), 897–927. Retrieved April 11, 2024, from https://ideas.repec.org//a/taf/eurjfi/v27y2021i9p897-927.html

Flögel, F. (2018). Distance and Modern Banks' Lending to SMEs: Ethnographic Insights from a Comparison of Regional and Large Banks in Germany [Publisher: Oxford University Press]. *Journal of Economic Geography*, *18*(1), 35–57. Retrieved April 11, 2024, from https://ideas.repec.org//a/oup/jecgeo/v18y2018i1p35-57..html

Flynn, B., Schroeder, R., & Sakakibara, S. (2007). The Impact of Quality Management Practices on Performance and Competitive Advantage. *Decision Sciences*, *26*, 659–691. https://doi.org/10.1111/j.1540-5915.1995.tb01445.x

Gabbi, G., Giammarino, M., & Matthias, M. (2020). Die Hard: Probability of Default and Soft Information [Number: 2 Publisher: Multidisciplinary Digital Publishing Institute]. *Risks*, *8*(2), 46. https://doi.org/10.3390/risks8020046

Gavalas, D., & Syriopoulos, T. (2014). An integrated credit rating and loan quality model: Application to bank shipping finance. *Maritime Policy & Management*, *42*, 1–22. https://doi.org/10.1080/03088839.2014.904948

Godbillon-Camus, B., & Godlewski, C. J. (2005). Credit Risk Management in Banks: Hard Information, Soft Information and Manipulation. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.882027

Goddard, M., Mannion, R., & Smith, P. C. (1999). Assessing the performance of NHS hospital trusts: The role of 'hard' and 'soft' information. *Health Policy (Amsterdam, Netherlands)*, *48*(2), 119–134. https://doi.org/10.1016/s0168-8510(99)00035-4

Guo, X., Wang, J., & Wang, F. (2014). Research On Credit Guarantee Problems In Small And Micro Enterprises' Credit Financing [ISSN: 1951-6851]. https://doi.org/10.2991/icetis-14.2014.23

Hamilton, R., & Fox, M. (1998). The Financing Preferences of Small Firm Owners. *International Journal of Entrepreneurial Behaviour & Research*, *4*, 239–248. https://doi.org/10.1108/13552559810235529

Hapke, H., Howard, C., & Lane, H. (2019, March). *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python* [Google-Books-ID: 9zczEAAAQBAJ]. Simon; Schuster.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. https://doi.org/10.1007/978-0-387-84858-7

Hsu, B.-M. (2020). Comparison of Supervised Classification Models on Textual Data [Number: 5 Publisher: Multidisciplinary Digital Publishing Institute]. *Mathematics*, *8*(5), 851. https://doi.org/10.3390/math8050851

Islam, M., & Sultana, N. (2018). Comparative Study on Machine Learning Algorithms for Sentiment Classification. *International Journal of Computer Applications*, *182*, 1–7. https://doi.org/10.5120/ijca2018917961

Joachims, T. (1998). Text Categorization with Support Vector Machines. *Proc. European Conf. Machine Learning (ECML'98)*. https://doi.org/10.17877/DE290R-5097

Kanakaraddi, S. G., Chikaraddi, A. K., Gull, K. C., & Hiremath, P. S. (2020). Comparison Study of Sentiment Analysis of Tweets using Various Machine Learning Algorithms. *2020 International Conference on Inventive Computation Technologies (ICICT)*, 287–292. https://doi.org/10.1109/ICICT48043.2020.9112546

Kaplan, R. S., & Norton, D. P. (1992). The Balanced Scorecard—Measures that Drive Performance [Section: Balanced scorecard]. *Harvard Business Review*. Retrieved April 11, 2024, from https://hbr.org/1992/01/the-balanced-scorecard-measures-that-drive-performance-2

Kaynak, H. (2003). The Relationship Between Total Quality Management Practices and Their Effects on Firm Performance. *Journal of Operations Management*, *21*. https://doi.org/10.1016/S0272-6963(03)00004-4

Kotsiantis, S., Zaharakis, I., & Pintelas, P. (2006). Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, *26*, 159–190. https://doi.org/10.1007/s10462-007-9052-3

Lehmann, B. (2003). Is It Worth the While? The Relevance of Qualitative Information in Credit Rating. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.410186

Li, D. (2023). Evaluating Various Machine Learning Techniques in Credit Risk Area. *BCP Business*

& Management, *38*, 2836–2844. https://doi.org/10.54691/bcpbm.v38i.4198

Li, Y. (2024). *Essays in financial technology: Banking efficiency and application of machine learning models in Supply Chain Finance and credit risk assessment* [PhD]. University of Glasgow. https://doi.org/10.5525/gla.thesis.84159

Liberti, J. M., & Mian, A. R. (2009). Estimating the Effect of Hierarchies on Information Use. *The Review of Financial Studies*, *22*(10), 4057–4090. https://doi.org/10.1093/rfs/hhn118

Liberti, J. M., & Petersen, M. A. (2019). Information: Hard and Soft. *The Review of Corporate Finance Studies*, *8*(1), 1–41. https://doi.org/10.1093/rcfs/cfy009

Lipshitz, R., & Shulimovitz, N. (2007). Intuition and Emotion in Bank Loan Officers' Credit Decisions. https://doi.org/https://doi.org/10.1518/155534307X232857

Liu, M. (2022). Assessing Human Information Processing in Lending Decisions: A Machine Learning Approach. *Journal of Accounting Research*, *60*. https://doi.org/10.1111/1475-679x.12427

Liu, Y., Bi, J.-W., & Fan, Z.-P. (2017). Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. *Expert Systems with Applications*, *80*, 323–339. https://doi.org/10.1016/j.eswa.2017.03.042

Manning, C., Raghavan, P., & Schuetze, H. (2009). Introduction to Information Retrieval.

Noriega, J., Rivera, L., & Herrera, J. (2023, August). *Machine Learning for Credit Risk Prediction: A Systematic Literature Review*. https://doi.org/10.20944/preprints202308.0947.v1

Orser, B. J., Riding, A. L., & Swift, C. S. (1994). Banking experiences of canadian micro-businesses [Publisher: World Scientific Publishing Co.]. *Journal of Enterprising Culture*, *01*(03n04), 321–345. https://doi.org/10.1142/S0218495894000033

Pattern Recognition and Machine Learning. (2006). Springer New York. https://doi.org/10.1007/978-0-387-45528-0

S Nair, D., Jayan, J., R R, R., & Elizabeth, S. (2015). Sentiment Analysis of Malayalam film review using machine learning techniques, 2381–2384. https://doi.org/10.1109/ICACCI.2015.7275974

Samson, D., & Terziovski, M. (1999). The relationship between total quality management practices and operational performance. *Journal of Operations Management*, *17*(4), 393–409. https://doi.org/10.1016/S0272-6963(98)00046-1

Sharifani, K., & Amini, M. (2023). Machine Learning and Deep Learning: A Review of Methods and Applications. *10*(07).

Soares, J., Pina, J., Ribeiro, M., & Catalão-Lopes, M. (2011). Quantitative vs. Qualitative Criteria for Credit Risk Assessment. *Frontiers in Finance and Economics*, *8*, 69–87.

Suhadolnik, N., Ueyama, J., & Da Silva, S. (2023). Machine Learning for Enhanced Credit Risk Assessment: An Empirical Approach. *Journal of Risk and Financial Management*, *16*, 496. https://doi.org/10.3390/jrfm16120496

Uchida, H., Udell, G., & Yamori, N. (2012). Loan officers and relationship lending to SMEs [Publisher: Elsevier]. *Journal of Financial Intermediation*, *21*(1), 97–122. Retrieved April 11, 2024, from https://econpapers.repec.org/article/eeejfinin/v_3a21_3ay_3a2012_3ai_3a1_3ap_3a97-122.htm

Udell, G., & Berger, A. (2002). Small Business Credit Availability and Relationship Lending: The Importance of Bank Organization Structure. *Economic Journal*, *112*, 32–32. https://doi.org/10.2139/ssrn.285937

Vinge, R., & Mckelvey, T. (2019). Understanding Support Vector Machines with Polynomial Kernels, 1–5. https://doi.org/10.23919/EUSIPCO.2019.8903042

Wang, W., Xu, Z., Lu, W., & Zhang, X. (2003). Determination of the spread parameter in the Gaussian kernel for classification and regression. *Neurocomputing*, *55*(3-4), 643–663. https://doi.org/10.1016/S0925-2312(02)00632-X

Wang, Z., Jiang, C., Zhao, H., & Ding, Y. (2020). Mining Semantic Soft Factors for Credit Risk Evaluation in Peer-to-Peer Lending. *Journal of Management Information Systems*, *37*, 282–308. https://doi.org/10.1080/07421222.2019.1705513

Yang, Y., & Cui, X. (2021). Bert-Enhanced Text Graph Neural Network for Classification. *Entropy*, *23*(11), 1536. https://doi.org/10.3390/e23111536

Zhou, D.-X., & Jetter, K. (2006). Approximation with polynomial kernels and SVM classifiers. *Advances in Computational Mathematics*, *25*(1-3), 323–344. https://doi.org/10.1007/s10444-004-7206-2

Ziemba, P., Becker, J., Becker, A., Radomska-Zalas, A., Pawluk, M., & Wierzba, D. (2021). Credit Decision Support Based on Real Set of Cash Loans Using Integrated Machine Learning Algorithms [Number: 17 Publisher: Multidisciplinary Digital Publishing Institute]. *Electronics*, *10*(17), 2099. https://doi.org/10.3390/electronics10172099

8 Appendices

8.1 Codes

8.1.1 Data Preprocessing Code

```
import msoffcrypto
import io
import pandas as pd
import nltk
import re
import spacy
from spacy.lang.nl.stop_words import STOP_WORDS
from tqdm import tqdm

# Pre-download necessary NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Load spaCy Dutch language model
nlp = spacy.load('nl_core_news_sm')

# Function Definitions
def process_text(text):
    """Clean and lemmatize text using regex and spaCy."""
    text = re.sub(r'[^a-zA-Z\s]', '', text)  # Remove non-alphabet characters
    text = re.sub(r'\s+', ' ', text).strip()  # Remove extra spaces and strip
    doc = nlp(text)
    return [token.lemma_.lower() for token in doc if token.text.lower() not in STOP_WORDS
     and not token.is_punct and not token.is_space]

def join_tokens(token_list):
    """Join a list of tokens into a single string."""
    return ' '.join(token_list)

# List of unwanted tokens
unwanted_tokens = ['xd', 'x', 'redacted', 'nan', 'ad']

def remove_unwanted_tokens(text):
    """Remove specific unwanted tokens."""
    tokens = text.split()
    filtered_tokens = [token for token in tokens if token.lower() not in unwanted_tokens]
    return ' '.join(filtered_tokens)

# Data Loading and Processing
file_path = r'C:\Users\mauri\Downloads\BA Circle 2.9 Dataset v2.2.xlsx'
password = 'YNqT‘8R@'

# Decrypt the file
file = msoffcrypto.OfficeFile(open(file_path, "rb"))
decrypted_stream = io.BytesIO()
file.load_key(password=password)
file.decrypt(decrypted_stream)

# Read the decrypted file
decrypted_stream.seek(0)
df = pd.read_excel(decrypted_stream, header=None)

```

```
# Clean up the header
new_columns = df.iloc[1].tolist()
df.columns = new_columns
column_mapping = {
    'Evaluations': 'LoanOfficerConclusion',
    'Kolom1': 'RiskManagerConclusion',
    'Kolom2': 'BusinessActivities',
    'Kolom3': 'EntrepreneurDescription',
    'Kolom4': 'MarktDescription',
    'Kolom5': 'ToelichtingPrive',
    'Kolom6': 'ToelichtingFinancieleAnalyse',
    'Kolom7': 'RequestId',
    'Kolom8': 'CreditLabel',
    'Kolom9': 'ApplicationDate',
    'Kolom10': 'Capital requirement for Business',
    'Kolom11': 'Disbursed/ not disbursed',
    'Kolom12': 'Amount disbursed',
    'Kolom132': 'Interest charged',
    'Kolom13': 'KredietScore (to be filled out)',
    'Kolom14': 'Defaulted (as of June 2023)',
    'Kolom15': 'Repayment Delays and Issues (as of June 2023)',
    'Kolom1510': 'Number of days delay repayment (categorical variable)',
    'Kolom158': 'Loan repaid (fully or partially) with repayment issues',
    'Kolom159': 'Loan repaid fully without issues',
    'Kolom156': 'Deferment of repayment due to Corona (as of June 2023)',
    'Kolom157': 'Deferment of repayment general (Uitstel avlossing as of June 2023)',
    'Kolom155': 'Investment into New Business (1 -New Business; 0- Existing Business)',
    'Kolom154': 'Purpose of Investment- Aquisition',
    'Kolom153': 'Purpose of Investment- Refinancing',
    'Kolom152': 'Corona Credit (Overbruggingscredit)',
    'Screening Method': 'Face-to-face meeting (instead of video) 1-Face-to-face; 0- Video
    ',
    'Details Screener': 'Screener gender',
    'Kolom16': 'Screener Experience Years Banking',
    'Kolom17': 'Experience Years at Qredits',
    'Details Entrepreneur and Business': 'Uitkering',
    'Kolom18': 'Age',
    'Kolom19': 'Gender (0-Male; 1-Female)',
    'Kolom20': 'Nationality',
    'Kolom21': 'Education',
    'Kolom22': 'Burgelijke Staat',
    'Kolom23': 'VoornaamstenInkomstbrom',
    'Kolom24': 'BedrijfsVorm',
    'Kolom24': 'Branche',
}
df.rename(columns=column_mapping, inplace=True)
df = df.iloc[3:].reset_index(drop=True)

# Drop rows with NaNs in the target column
df = df.dropna(subset=['Disbursed/ not disbursed'])

# Ensure columns to process are strings
columns_to_process = ['LoanOfficerConclusion', 'RiskManagerConclusion', '
    BusinessActivities','EntrepreneurDescription','MarktDescription','ToelichtingPrive','
    ToelichtingFinancieleAnalyse']

for col in columns_to_process:
    df[col] = df[col].astype(str)

```

```
# Apply text processing with progress bars
tqdm.pandas(desc="Processing Text Data")

for col in columns_to_process:
    df[col] = df[col].progress_apply(process_text)

# Apply join_tokens function to each column
for col in columns_to_process:
    df[col] = df[col].progress_apply(join_tokens)

# Apply remove_unwanted_tokens function to each column
for col in columns_to_process:
    df[col] = df[col].progress_apply(remove_unwanted_tokens)

# Apply transformations
df['Uitkering'] = df['Uitkering'].apply(lambda x: 0 if x == 'Geen' else 1)
df['Nationality'] = df['Nationality'].apply(lambda x: 0 if x == 'Nederlandse' else 1)

# Mapping categorical variables to numerical values
value_mapping0 = {
    'Qredits': 0,
    'MKB': 1,
    'EZK': 2,
}
df['CreditLabel'] = df['CreditLabel'].map(value_mapping0)

value_mapping1 = {
    'Voortgezetonderwijs': 0,
    'LBO': 1,
    'MBO': 2,
    'HBO': 3,
    'WO': 4,
}
df['Education'] = df['Education'].map(value_mapping1)

value_mapping2 = {
    'Samenwonend': 0,
    'Gehuwd onder huwelijkse voorwaarden': 1,
    'Gehuwd in gemeenschap van goederen': 2,
    'Alleenstaand': 3,
    'Gescheiden' : 4,
    'Geregistreerd Partnerschap' : 5
}
df['Burgelijke Staat'] = df['Burgelijke Staat'].map(value_mapping2)

value_mapping3 = {
    'NULL': 0,
    'Inkomen uit eigen onderneming': 1,
    'Inkomen uit uitkering': 2,
    'Inkomen uit loondienst': 3,
    'Anders' : 4,
    'Pensioen ' : 5,
    'Geen inkomen' : 6,
    'Studiefinanciering' : 7
}
df['VoornaamstenInkomstbrom'] = df['VoornaamstenInkomstbrom'].map(value_mapping3)

# Create a combined column based on several criteria
df['Combined'] = (df['Defaulted (as of June 2023)'] |
```

```
                    df['Repayment Delays and Issues (as of June 2023)'] |
                    df['Loan repaid (fully or partially) with repayment issues'] |
                    df['Deferment of repayment due to Corona (as of June 2023)'] |
                    df['Deferment of repayment general (Uitstel avlossing as of June 2023)'
    ]).astype(int)

# Saving data
df.to_pickle('preprocessed_data.pkl')  # Save the whole dataframe after processing as a
    pickle file
df.to_excel('preprocessed_data.xlsx')  # Save the whole dataframe after processing as an
    Excel file
```

data_preprocessing_final.py

8.1.2 Feature Engineering Code

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
import tqdm

# Load the DataFrame
df = pd.read_pickle('preprocessed_data.pkl')

# Combine the text columns into a new column 'Combined1'
df['Combined1'] = df[['LoanOfficerConclusion', 'RiskManagerConclusion', '
    BusinessActivities', 'EntrepreneurDescription', 'MarktDescription', 'ToelichtingPrive
    ', 'ToelichtingFinancieleAnalyse']].apply(lambda x: ' '.join(x), axis=1)

# Columns to be processed
columns_to_process = ['LoanOfficerConclusion', 'RiskManagerConclusion', '
    BusinessActivities', 'EntrepreneurDescription', 'MarktDescription', 'ToelichtingPrive
    ', 'ToelichtingFinancieleAnalyse', 'Combined1']

# Create an empty DataFrame to store the results
result_df = pd.DataFrame()

tfidf = TfidfVectorizer(ngram_range=(1, 3), max_features=1000)  # Unigrams, bigrams, and
    trigrams

for col in tqdm.tqdm(columns_to_process):
    # Perform the TF-IDF transformation
    tfidf_transformed = tfidf.fit_transform(df[col])
    # Create a DataFrame from the sparse matrix and add it to the result_df
    tfidf_df = pd.DataFrame(tfidf_transformed.toarray(), columns=[f"{col}_{feature}" for
    feature in tfidf.get_feature_names_out()])
    result_df = pd.concat([result_df, tfidf_df], axis=1)  # Concatenate the TF-IDF
    DataFrame with the result_df

# Saving data
result_df.to_pickle('tf_idf_results.pkl')  # Save the whole DataFrame after processing as
     a pickle file
```

feature_engineering_tf_idf_final.py

8.1.3 SVM Parameter Optimalisation Code

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score,
    roc_auc_score, confusion_matrix, matthews_corrcoef
import pickle

print('Starting')

# Load the data
df = pd.read_pickle('preprocessed_data.pkl')
feature = pd.read_pickle('tf_idf_results.pkl')

# Labels
y = df['Disbursed/ not disbursed']

# Ensure that the labels are discrete values
if y.dtype == 'object' or y.dtype == 'float64':
    y = LabelEncoder().fit_transform(y)

print('Splitting')

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(feature, y, test_size=0.2,
    random_state=42)

print('Training with parameter tuning')

# Define the parameter grid for grid search
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': ['scale', 'auto'],
    'kernel': ['linear', 'rbf', 'sigmoid', 'poly']
}

svm_model = SVC(class_weight='balanced', probability=True)

# Perform grid search with the specified parameter grid
grid_search = GridSearchCV(svm_model, param_grid, cv=5)

grid_search.fit(X_train, y_train)

print('Testing')

# Test the model
y_pred = grid_search.predict(X_test)

print('Evaluating')

# Calculate and print precision, recall, F1-score, and ROC-AUC
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)
mcc = matthews_corrcoef(y_test, y_pred)

```

```
# Collect results in a DataFrame
results = pd.DataFrame([{
    'Kernel': grid_search.best_estimator_.kernel,
    "C": grid_search.best_estimator_.C,
    "Gamma": grid_search.best_estimator_.gamma,
    "Class weight": grid_search.best_estimator_.class_weight,
    "Probability": grid_search.best_estimator_.probability,
    'Accuracy': accuracy,
    'Precision': precision,
    'Recall': recall,
    'F1-score': f1,
    'ROC-AUC': roc_auc,
    'MCC': mcc
}])

# Print results
print(results)

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)

# Save the trained model
with open('svm_tfidf_best.pkl', 'wb') as file:
    pickle.dump(grid_search.best_estimator_, file)
```

svm_parameter_opt_final.py

8.1.4 SVM Model Code

```
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score, StratifiedKFold
from sklearn.svm import SVC
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score,
    roc_auc_score, confusion_matrix, matthews_corrcoef
import pickle
import matplotlib.pyplot as plt
from sklearn.inspection import DecisionBoundaryDisplay

print('Starting')

# Load the data
df = pd.read_pickle('preprocessed_data.pkl')
feature = pd.read_pickle('tf_idf_results.pkl')

# Labels
y = df['Disbursed/ not disbursed']

# Ensure that the labels are discrete values
if y.dtype == 'object' or y.dtype == 'float64':
    y = LabelEncoder().fit_transform(y)

print('Splitting')

# Split the data into training and test sets
```

```
X_train, X_test, y_train, y_test = train_test_split(feature, y, test_size=0.2,
    random_state=42)

print('Cross validation')

# Initialize the SVC with the specified parameters
svm_model = SVC(kernel='linear', C=0.1, gamma='scale', probability=False)

# Perform cross-validation
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cv_results = cross_val_score(svm_model, X_train, y_train, cv=cv, scoring='accuracy')

print('Cross-validation results:')
print(cv_results)
print('Mean cross-validation score:', cv_results.mean())

print('Training')

# Train the model
svm_model.fit(X_train, y_train)

print('Testing')

# Test the model
y_pred = svm_model.predict(X_test)

print('Evaluating')

# Calculate and print precision, recall, F1-score, and ROC-AUC
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)
mcc = matthews_corrcoef(y_test, y_pred)

# Collect results in a DataFrame
results = pd.DataFrame([{
    'Kernel': svm_model.kernel,
    "C": svm_model.C,
    "Gamma": svm_model.gamma,
    "Class weight": svm_model.class_weight,
    "Probability": svm_model.probability,
    'Accuracy': accuracy,
    'Precision': precision,
    'Recall': recall,
    'F1-score': f1,
    'ROC-AUC': roc_auc,
    'MCC': mcc
}])

# Print results
print(results)

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)

```

```
# Save the trained model
with open('svm_tfidf_linear_final.pkl', 'wb') as file:
    pickle.dump(svm_model, file)
```

svm_final.py

8.1.5 SVM Comparison Code

```
import pandas as pd
import pickle
from sklearn.preprocessing import LabelEncoder

print('Starting')

# Load the data
df = pd.read_pickle('preprocessed_data.pkl')
feature = pd.read_pickle('tf_idf_results.pkl')

# Labels
y = df['Disbursed/ not disbursed']

# Ensure the labels are discrete values
if y.dtype == 'object' or y.dtype == 'float64':
    y = LabelEncoder().fit_transform(y)

print('Loading model')

# Load the pre-trained SVM model
with open('svm_tfidf_default2.2.pkl', 'rb') as file:
    svm_model = pickle.load(file)

print('Predicting')

# Predict using the model for the entire dataset
y_pred = svm_model.predict(feature)

print('Creating Excel sheet')

# List of columns to include in the output
columns_to_include = [
    'RequestId', 'CreditLabel', 'ApplicationDate', 'Capital requirement for Business', '
    Disbursed/ not disbursed',
    'Amount disbursed', 'KredietScore (to be filled out)', 'Interest charged', 'Combined'
    , 'Defaulted (as of June 2023)',
    'Repayment Delays and Issues (as of June 2023)', 'Number of days delay repayment (
    categorical variable)',
    'Loan repaid (fully or partially) with repayment issues', 'Loan repaid fully without
    issues',
    'Deferment of repayment due to Corona (as of June 2023)', 'Deferment of repayment
    general (Uitstel avlossing as of June 2023)',
    'Investment into New Business (1 -New Business; 0- Existing Business)', 'Purpose of
    Investment- Aquisition',
    'Purpose of Investment- Refinancing', 'Corona Credit (Overbruggingscredit)', 'Face-to
    -face meeting (instead of video) 1-Face-to-face; 0- Video',
    'Screener gender', 'Screener Experience Years Banking', 'Experience Years at Qredits'
    , 'Uitkering', 'Age',
    'Gender (0-Male; 1-Female)', 'Nationality', 'Education', 'Burgelijke Staat', '
    VoornaamstenInkomstbrom'
```

```
]

# Create a DataFrame for the entire data with the required columns and add the prediction
    column
df_with_predictions = df[columns_to_include].copy()
df_with_predictions['Prediction'] = y_pred

# Save the DataFrame to an Excel file
df_with_predictions.to_excel('data_with_predictions.xlsx', index=False)

print('Excel sheet created')
```

svm_comparison_final.py

8.1.6 SVM Feature Extraction Code

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import pickle
import re

# Load the data
df = pd.read_pickle('preprocessed_data.pkl')
feature = pd.read_pickle('tf_idf_results.pkl')
y = df['Disbursed/ not disbursed']

# Ensure labels are discrete values
if y.dtype == 'object' or y.dtype == 'float64':
    y = LabelEncoder().fit_transform(y)

# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(feature, y, test_size=0.2,
    random_state=42)

# Load the saved SVM model
with open('svm_linear_01.pkl', 'rb') as file:
    svm_model = pickle.load(file)

# Get the feature weights
feature_weights = svm_model.coef_[0]

# Create a DataFrame for feature importance
feature_names = feature.columns if isinstance(feature, pd.DataFrame) else feature.
    get_feature_names_out()

# Clean up the feature names by removing trailing 'xd'
clean_feature_names = [re.sub(r'xd+$', '', name) for name in feature_names]

feature_importance_df = pd.DataFrame({'Feature': clean_feature_names, 'Weight':
    feature_weights})

# Calculate the absolute weights
feature_importance_df['AbsWeight'] = feature_importance_df['Weight'].abs()

# Calculate the threshold for influential features
mean_weight = feature_importance_df['AbsWeight'].mean()
std_weight = feature_importance_df['AbsWeight'].std()
threshold = mean_weight + std_weight

# Identify influential features
influential_features_df = feature_importance_df[feature_importance_df['AbsWeight'] >
    threshold]

# Remove duplicates by keeping the highest absolute weight for each feature
influential_features_df = influential_features_df.sort_values(by='AbsWeight', ascending=
    False).drop_duplicates(subset='Feature')

# Sort by absolute weight
influential_features_df = influential_features_df.sort_values(by='AbsWeight', ascending=
    False)
```

```
52
53 # Save the influential features to an Excel file
54 influential_features_df.to_excel('influential_features_threshold.xlsx', index=False)
55 print("Influential features and their weights have been saved to '
       influential_features_threshold.xlsx'.")
56
57 # Plot the influential features
58 plt.figure(figsize=(10, 15))
59 plt.barh(influential_features_df['Feature'], influential_features_df['Weight'], color='
       skyblue')
60 plt.xlabel('Weight')
61 plt.ylabel('Feature')
62 plt.title('Influential Features')
63 plt.gca().invert_yaxis()
64 plt.show()
```

svm_feature_extraction_final.py

8.2 Data Analysis of the Actual Data

Overview	Count	Percentage	Average CreditScore	Average Amount Disbursed	% Face to Face	Average Age	% Benefits	% Male	% Non Dutch
All	14166	100%	6.56	19761,30	32%	41	14%	69%	6%
Disbursed	12171	86%	6.56	23000,46	31%	40	15%	69%	6%
Not Disbursed	1995	14%	6.60	0,00	35%	41	14%	72%	8%
Problems	6087	50%	6.35	21028,71	34%	40	16%	66%	7%
No Problems	6084	50%	6.77	24973,18	29%	41	13%	72%	6%

Table 7: Actual Data Overview

Education	Voortgezet onderwijs	LBO	MBO	HBO	WO
Disbursed	5%	3%	49%	35%	8%
Not Disbursed	7%	2%	45%	36%	10%
Problems	5%	3%	50%	34%	7%
No Problems	5%	3%	47%	36%	9%

Table 8: Education Overview

Marital Status	Cohabating	Married under Prenuptial agreement	Married in community of property	Single	Divorced	Registered Partnership
Disbursed	21%	10%	28%	31%	5%	4%
Not Disbursed	18%	12%	27%	32%	5%	6%
Problems	20%	8%	26%	35%	5%	4%
No Problems	21%	11%	31%	28%	5%	4%

Table 9: Marital Status Overview

Main Income Source	Own Business	Benefits	Employment	Other	Pension	No Income	Student Grants
Disbursed	65%	10%	20%	1%	0%	2%	1%
Not Disbursed	65%	10%	19%	1%	0%	2%	0%
Problems	65%	12%	18%	1%	0%	2%	1%
No Problems	64%	9%	22%	0%	0%	1%	1%

Table 10: Main Income Source Overview

Loan Product	Company Loan Product	MKB	EZK
Disbursed	95%	4%	0%
Not Disbursed	84%	1%	0%
Problems	96%	3%	0%
No Problems	94%	5%	0%

Table 11: Loan Product Overview

8.3 Data Analysis of the Predicted Data

Overview	Count	Percentage	Average CreditScore	Average Amount Disbursed	% Face to Face	Average Age	% Benefits	% Male	% Non Dutch
All	14166	100%	6.56	19761,30	32%	41	14%	69%	6%
Disbursed	12893	91%	6.58	21636,20	32%	40	15%	69%	6%
Not Disbursed	1273	9%	6.44	772,23	34%	41	12%	72%	10%
Problems	6065	47%	6.35	21027,91	34%	40	16%	66%	7%
No Problems	6828	53%	6.78	22176,52	30%	41	13%	72%	6%

Table 12: Prediction Overview

Education	Voortgezet onderwijs	LBO	MBO	HBO	WO
Disbursed	5%	3%	48%	35%	8%
Not Disbursed	7%	2%	46%	36%	9%
Problems	5%	3%	50%	34%	7%
No Problems	5%	3%	47%	36%	9%

Table 13: Education Overview

Marital Status	Cohabating	Married under prenuptial agreement	Married in community of property	Single	Divorced	Registered Partnership
Disbursed	21%	10%	28%	31%	5%	4%
Not Disbursed	17%	11%	26%	34%	6%	5%
Problems	20%	8%	26%	35%	6%	4%
No Problems	21%	11%	31%	27%	5%	5%

Table 14: Marital Status Overview

Main Income Source	Own Business	Benefits	Employment	Other	Pension	No Income	Student Grants
Disbursed	64%	11%	20%	1%	0%	2%	1%
Not Disbursed	67%	10%	17%	1%	0%	2%	1%
Problems	65%	12%	18%	1%	0%	2%	1%
No Problems	64%	9%	22%	0%	0%	1%	1%

Table 15: Main Income Source Overview

Loan Product	Company Loan Product	MKB	EZK
Disbursed	95%	5%	1%
Not Disbursed	82%	17%	1%
Problems	97%	3%	0%
No Problems	93%	6%	1%

Table 16: Loan Product Overview

8.4 Example of Feature Extraction

Feature	Weight	Category
bevoegdheid	1.783	Entrepreneur
eb	1.765	Business
akkoord	1.734	Other
omzetting	1.104	Business
verstrekking	0.930	Financials
bkr	0.885	Business
acceptatiescore	0.864	Financials
flex	0.830	Other
aanvulling	0.812	Other
c	0.797	Business
overname	0.797	Business
beperken	0.793	Other
taakstelling	0.785	Entrepreneur
technisch	0.778	Business
rating	0.747	Other
lening	0.715	Financials
ts	0.692	Other
financieringsverzoek	0.689	Financials
rr	0.685	Other
corona	0.677	Other
covid	0.648	Other
zien	0.631	Other
k	0.612	Other
conform	0.610	Other
partner	0.604	Entrepreneur
priv	0.599	Entrepreneur
toelichting	0.598	Other
financiering	0.592	Financials
beheer	0.582	Financials

8.5 Analysis of the Feature Extraction

Categories	Number of features	Average Weights	Combined Weight
Financials	49	0.338	16.571
Entrepreneur	44	0.332	14.603
Business	62	0.354	21.920
Other	263	0.288	75.816

Table 18: Positive Categories

Categories	Number of features	Average Weights	Combined Weight
Financials	7	-0.200	-1.397
Entrepreneur	11	-0.216	-2.373
Business	11	-0.194	-2.129
Other	23	-0.196	-4.512

Table 19: Negative Categories

8.6 Additional Explanation of Concepts of Machine Learning

The technique of optimizing the performance of a system by programming with historical data and experiences is known as machine learning Alpaydin, 2004. When human expertise is lacking, machine learning is required to do data analysis, which may be completed quickly over a vast quantity of data. For instance, anyone would typically be able to tell the difference between a man's and a woman's voice after just one listen. However, it can take some time for someone to finish this assignment if there are 100 voice recordings to identify. By designing and running machine learning algorithms that are run on a computer, this can be done automatically and therefore is beneficial to save time and resources compared to doing manual work carried out by humans.

Machine learning is used everywhere and in a variety of fields. Researchers and practitioners have been using this paradigm to assist in the solution of real-world issues, such as forecasting consumer behavior in grocery stores, evaluating credit applications, identifying faces and voices, diagnosing illnesses, optimizing networks, and so forth. Now and again, the machine learning tasks leverage the availability of labeled datasets. This is frequently referred to as a classification problem, in which the chosen algorithm must determine which class or categories the new data belongs in.

The machine learning algorithm can be trained with this labeled data to help comprehend the attributes and traits of various photographs that correlate with the related tags. This type of process, where the data instances are labeled with the appropriate output, is known as supervised learning Kotsiantis et al., 2006.

8.6.1 Data Preprocessing

In order to use the date in our machine learning model a preprocessing process had to be conducted. The preprocessing of the data consists of cleaning and transforming the data to make it suitable for analysis and selecting the variables relevant to our study. The preprocessing consists of the following aspects according to Hapke et al., 2019 and others like Yang and Cui, 2021

1. Text Aggregation: Combine the different textual fields for each loan application into a single text document. This ensures that the neural network considers all available information.
2. Cleaning: Standardize the text by removing special characters, HTML tags, and any irrelevant information that does not contribute to understanding the loan's risk.
3. Tokenization: Convert the cleaned text into tokens (words or phrases).
4. Stop Word Removal: Eliminate common words that add little to no value in understanding sentiment or context.
5. Normalization: Converting text to a uniform case (usually lowercase) for consistency.
6. Questions: Using terms like "when," "when," "who," "how," and so on won't help to clarify polarity.
7. Lemmatization: Reducing words to their lemma to ensure different forms of a word are analyzed as one.

8.6.2 Feature Engineering

A statistical measure called Term Frequency-Inverse Document Frequency (TF-IDF) is used to assess a word's significance to a document inside a collection or corpus according to Manning et al., 2009. A word's relevance rises in direct proportion to how frequently it occurs in the document, however this is countered by the word's frequency in the corpus. Text mining and information retrieval are two fields that make extensive use of this technology.

Term Frequency, TF, measures how frequently a term occurs in a document. It is calculated as:

$$TF(t,d) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document\ d}{Total\ number\ of\ terms\ in\ the\ document\ d}$$

Inverse Document Frequency, IDF, measures the importance of the term within the entire corpus. The idea is that if a word is common and appears in many documents, it's likely not a good discriminator The IDF of a term is calculated as the logarithm of the number of documents divided by the number of documents that contain the word.

$$IDF(t) = \log\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ term\ t}\right)$$

The TF-IDF score is the product of TF and IDF.

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

TF-IDF can be extended to consider word sequences, or n-grams, in addition to single words, unigrams. By taking into account phrases or word combinations, this method considers the context in which terms appear and can enhance the functionality of text mining and information retrieval applications. Unigrams are single words. Bigrams are sequences of two adjacent words. Bigrams capture pairs of words that often appear together, providing more context than unigrams. Trigrams are sequences of three adjacent words offering even richer context. By using n-grams, TF-IDF can better capture the structure and meaning of the text Manning et al., 2009.

8.6.3 Support Vector Machine Model

One of the newest supervised machine learning methods is called Support Vector Machine (SVM)(Kotsiantis et al., 2006. The model learns to identify new labels for a set of data samples that have been classified into one of two categories. Therefore, the model is referred to as a binary linear classifier also known as the "kern machine". It is a maximum margin method that is also represented by the total influence of a subset of the training cases Alpaydin, 2004. An optimal separating hyperplane that splits the data instances into two sides with the greatest margin is attempted to be established by the algorithm, assuming that certain data points are defined in n-dimensional vector space. That being said, this hyperplane will be defined in (n-1) dimensional vector space.

Support Vector Machines (SVMs) aim to find a hyperplane that optimally separates a dataset into two classes, forming a decision boundary. However, datasets are often complex and not linearly separable. To classify such datasets effectively, it is necessary to move beyond a two-dimensional perspective and consider a higher-dimensional space. This can be visualized by imagining two sets of colored balls on a flat surface. If the surface is elevated, causing the balls to be lifted into the air, the surface can then be used to separate them while they are airborne. This elevation process represents mapping data into a higher-dimensional space, a technique known as kerneling.

Given our transition to three dimensions, the hyperplane used for classification can no longer be represented by a simple line; instead, it must now take the form of a plane. The fundamental idea is that the data will be mapped into progressively higher dimensions until a hyperplane can be constructed to effectively separate it. This iterative process of dimension mapping continues until the data is linearly separable by the hyperplane.

S Nair et al., 2015 used a hybrid technique combining rule-based and machine learning to undertake an experiment on sentiment analysis of Malayalam cinema reviews. Conditional Random Field and a SVM were employed in this investigation with a strategy that combines each with a rule-based approach, and it was discovered that SVM outperformed CRF with the best accuracy of 91%.

Cortes and Vapnik, 2009 introduce SVMs and outline it's theoretical underpinnings, establishing SVMs as powerful tools for classification tasks, including those with high-dimensional feature spaces typical of text data.

8.6.4 Kernel

When data is not linearly separable, SVM uses the "kernel trick" to map the original data into a higher-dimensional space where it is linearly separable. Common kernels include polynomial, radial basis function (RBF), and sigmoid. The kernel function implicitly computes the dot product in the higher-dimensional space without explicitly performing the transformation, which is computationally efficient. C. Campbell and Ying, 2011

The linear kernel is the simplest type, suitable for linearly separable data. It maps the input features into a space where they are linearly separable. The linear kernel is, when usable, very efficient and computationally inexpensive.

$$K(x, y) = x \cdot y$$

Here, x and y are input vectors. The dot product represents the similarity between the two vectors.

The RBF kernel, also known as the Gaussian kernel, is suitable for non-linearly separable data. It maps the input features into an infinite-dimensional space. The RBF kernel is highly effective in high dimensional spaces. W. Wang et al., 2003

$$K(x, y) = \exp(-\gamma \|x - y\|^2)$$

$||x - y||$ is the Euclidean distance between the vectors x and y, γ is a parameter that defines the spread of the kernel.

The Sigmoid kernel, also known as the hyperbolic tangent kernel, is inspired by neural networks. It can handle non-linear relationships in the data. Useful in scenarios where data resembles patterns that neural networks can capture. Camps-Valls et al., 2004

$$K(x, y) = \tanh(\alpha \cdot x \cdot y + c)$$

tanh is the hyperbolic tangent function. α and c are kernel parameters that need to be tuned.

The polynomial kernel maps the input features into a higher-dimensional space defined by polynomials. Suitable for problems where the relationship between features is polynominal. Vinge and Mckelvey, 2019

$$K(x, y) = (x \cdot y + c)^d$$

$x \cdot y$ is the dot product, c is a coefficient to trade off the influence of higher-order versus lower-order terms and d is the degree of the polynomial.

Kernel	Formula	Suitable For
Linear	$K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$	Linear relations
RBF	$K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$	Non-linear relations
Sigmoid	$K(\mathbf{x}, \mathbf{y}) = \tanh(\alpha \mathbf{x} \cdot \mathbf{y} + c)$	Neural network like relations
Polynomial	$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + c)^d$	Polynomial relations

Table 20: Comparison of different SVM kernel functions

8.6.5 Parameters

The SVM classifier's regularization parameter is represented by the float parameter C, which has a default value of 1.0. The regularization effect decreases as C increases and vice versa since the regularization strength is inversely related to the value of C. This parameter applies a squared L2 penalty and needs to be strictly positive. Cherkassky and Ma, 2004

The gamma parameter defaults to "scale," but it can also take the values "auto," "scale," or a float. This coefficient pertains to the 'poly','sigmoid', and 'rbf' kernels. Gamma is computed as

$$1/(n_features * X.var())$$

when set to 'scale'. If 'auto' is selected, gamma is computed as

$$1/n_features$$

. If one is given a float, it has to be non-negative. Cherkassky and Ma, 2004

The degree of the polynomial kernel function (or "poly") is specified by the degree parameter, an integer with a default value of 3. This parameter is ignored by all other kernel types and needs to be non negative.Zhou and Jetter, 2006

A boolean parameter called probability signals whether to activate probability estimates, which have to be activated before the fit procedure may be called. Because the fit technique internally use 5-fold cross-validation. 5-fold cross-validation is a method used to evaluate the performance of a machine learning model by dividing the

dataset into five equal parts or "folds." The model is trained and validated five times, each time using a different fold as the validation set and the remaining four folds as the training set. The purpose of this internal cross-validation is to ensure that the probability estimates are well-calibrated and reliable. However, this additional step requires more computational resources and time *Pattern Recognition and Machine Learning*, 2006.

The $class_weight$ option has a default value of None and can be modified to a dictionary or "balanced." This option sets $class_weight[i] * C$ as the parameter C for each class i. All classes are taken to have a weight of one if it is not stated otherwise. The "balanced" mode computes weights as

$$n_samples/(n_classes * np.bincount(y))$$

, dynamically adjusting them inversely proportional to class frequencies in the input data. This adjustment ensures that minority classes contribute more significantly to the decision boundary, helping the model to better recognize instances of these underrepresented classes Hastie et al., 2009; *Pattern Recognition and Machine Learning*, 2006.

8.6.6 Evaluations

The percentage of accurately predicted instances among all of the dataset's instances is known as accuracy.

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

Where TP stands for true positives, TN for true negatives, FP for false positives, and FN for false negatives.

Although accuracy is simple, unbalanced datasets might lead to misleading results Alpaydin, 2004

The precision of a model indicates how well it predicts positive instances based on the ratio of genuine positive predictions to all anticipated positives. The equation is:

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

When the cost of false positives is significant, a low number of false positives is indicated by high precision.Alpaydin, 2004

The model's recall, also known as sensitivity, gauges its capacity to recognize every true positive case. It is computed as follows:

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

When the cost of false negatives is high, high recall is significant because it shows that the model is successful in capturing all positive cases.Alpaydin, 2004

The F1 score is a single statistic that balances both precision and recall, calculated as the harmonic mean of the two. It is provided by:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1 score is particularly useful for imbalanced datasets, as it considers both precision and recall. *Pattern Recognition and Machine Learning*, 2006

A comprehensive analysis of the model's performance is given via a confusion matrix:

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	TN	TN

Table 21: Confusion Matrix

This matrix provides insights for improvement by assisting in the identification of particular error categories.*Data Mining*, 2011

Taking into consideration all four confusion matrix categories, Matthews Correlation Coefficient, MCC, is a gauge of the accuracy of binary classifications. It is computed as follows:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

The MCC value scale goes from -1 to 1, where 1 denotes a flawless prediction, 0 a random guess, and -1 a complete discrepancy between the predicted and actual results.Chicco et al., 2021

8.7 Results of Parameter Optimization

Kernel	C	Gamma	Probability	Class_Weight	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC
RBF	1	scale	false	none	0.935	0.932	0.996	0.963	0.786	0.716
RBF	1	scale	true	none	0.935	0.932	0.996	0.963	0.786	0.716
RBF	1	scale	true	balanced	0.929	0.934	0.988	0.960	0.787	0.688
RBF	1	scale	false	balanced	0.929	0.934	0.988	0.960	0.787	0.688
RBF	1	auto	false	none	0.855	0.855	1.000	0.922	0.500	0.000
RBF	1	auto	false	balanced	0.145	0.000	0.000	0.000	0.500	0.000
RBF	1	auto	true	balanced	0.145	0.000	0.000	0.000	0.500	0.000
RBF	1	auto	true	none	0.145	0.000	0.000	0.000	0.500	0.000
Sigmoid	1	scale	false	none	0.936	0.936	0.994	0.964	0.796	0.722
Sigmoid	1	scale	true	none	0.936	0.936	0.994	0.964	0.796	0.722
Sigmoid	1	scale	true	balanced	0.863	0.937	0.900	0.919	0.773	0.502
Sigmoid	1	scale	false	balanced	0.863	0.937	0.900	0.919	0.773	0.502
Sigmoid	1	auto	true	none	0.855	0.855	1.000	0.922	0.500	0.000
Sigmoid	1	auto	true	balanced	0.145	0.000	0.000	0.000	0.500	0.000
Sigmoid	1	auto	false	none	0.855	0.855	1.000	0.922	0.500	0.000
Sigmoid	1	auto	false	balanced	0.145	0.000	0.000	0.000	0.500	0.000

Table 22: Model Performance Metrics

Kernel	C	Gamma	Probability	Class_weight	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC
Linear	1	n/a	false	n/a	0.909	0.932	0.964	0.948	0.777	0.607
Linear	1	n/a	true	n/a	0.909	0.932	0.964	0.948	0.777	0.607

Table 23: Linear Kernel Model Performance Metrics

Kernel	C	Gamma	Probability	Class_weight	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC
Linear	1	scale	false	none	0.909	0.932	0.964	0.948	0.777	0.607
RBF	1	scale	false	none	0.935	0.932	0.996	0.963	0.786	0.716
Sigmoid	1	scale	false	none	0.936	0.936	0.994	0.964	0.795	0.722
Poly	1	scale	false	none	0.855	0.855	1.000	0.502	0.502	0.064

Table 24: Kernel Model Performance Metrics

Kernel	C	Gamma	Probability	Class_weight	Accuracy	Precision	Recall	F1-score	ROC-AUC	MCC
Linear	0.1	n/a	false	n/a	0.937	0.936	0.995	0.964	0.797	0.726
Linear	1	n/a	false	n/a	0.909	0.932	0.964	0.948	0.777	0.607
Linear	10	n/a	false	n/a	0.874	0.928	0.924	0.926	0.751	0.497
RBF	0.10	scale	false	none	0.897	0.893	0.999	0.943	0.647	0.506
RBF	1.00	scale	false	none	0.935	0.932	0.996	0.963	0.786	0.716
RBF	10	scale	false	none	0.931	0.865	0.992	0.961	0.781	0.694
Sigmoid	0.10	scale	false	none	0.867	0.865	1.000	0.928	0.541	0.267
Sigmoid	1.00	scale	false	none	0.936	0.936	0.994	0.964	0.796	0.722
Sigmoid	10.00	scale	false	none	0.901	0.928	0.958	0.943	0.761	0.571

Table 25: Kernel Model Performance Metrics