Predictive Modeling for Credit Card Default Detection: A Supervised Machine Learning Approach

Vamsi Deekshit Kanakavety
Dept. of Computer Science
Arizona State University
Tempe, AZ 85282
vkanakav@asu.edu

Sharan Bajjuri
Dept. of Computer Science
Arizona State University
Tempe, AZ 85282
sharanb@asu.edu

Sathish Reddy Gathpa
Dept. of Computer Science
Arizona State University
Tempe, AZ 85282
sgathpa@asu.edu

Prashanth Reddy Tummala Palli
Dept. of Computer Science
Arizona State University
Tempe, AZ 85282
ptummal3@asu.edu

Jahnavi Kolli
Dept. of Computer Science
Arizona State University
Tempe, AZ 85282
jkolli@asu.edu

1 Introduction

The primary motivation for identifying credit card defaulters is to minimize financial risks and safeguard the interests of credit card issuers and lenders, while also adhering to stringent industry standards and avoiding potential legal and financial repercussions. Timely detection of defaulters enables the companies to take necessary actions to maintain customer relationships and, if required, pursue legal remedies. By providing more accurate risk assessments, credit card companies can make informed decisions about lending money or extending credit, while minimizing potential losses, and preserving the financial health of the organizations.

Early detection of defaulting not only prevents financial losses for both customers and the company but also plays a significant role in preserving customer trust and loyalty. By customizing credit offerings to individual risk profiles, companies can strike a balance between profitability and risk management, resulting in overall enhanced profitability for credit card firms. The credit card industry thrives in a data-rich environment with vast amounts of transaction and customer data. Analyzing customer actions and payment trends assists financial institutions in optimizing their lending methods and improving their evaluation of risk.

2 Project Description

The main goal of the project is to develop a robust credit card default prediction system using supervised machine learning techniques. It involves the analysis of historical credit card transactions and customer information to pre-process and engineer features that will improve the models' predictive capabilities. The objective is to explore a wide range of supervised algorithms to classify the data, and these algorithms will be carefully fine-tuned to achieve optimal performance. As we achieve this, another main objective is to conduct a thorough evaluation of the models, including the use of metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, to help in the selection of the most accurate model. The project harnesses the capabilities of machine learning to create a robust system that not only reduces default rates but also enables more accurate credit assessments, ultimately benefiting both lenders and borrowers.

3 Technical Background

3.1 Python Libraries

NumPy. NumPy is essential for numerical computations and data manipulation, providing a foundation for working with arrays and matrices required for machine learning.

Pandas. Pandas is an open-source Python tool for manipulating and analyzing data. It is used for data cleaning, exploration, and transformation. Dataframes organize and manipulate the dataset efficiently.

Scikit-Learn (sklearn). Scikit-Learn is a comprehensive Python library for machine learning. It includes a variety of algorithms, model evaluation tools, and pre-processing techniques that the project will employ for model development, training, testing, and evaluation.

Matplotlib. Matplotlib is a 2D plotting library for Python, for creating static, interactive, and animated visualizations, complementing the data analysis process by fine-tuning visual representations of data.

Seaborn. Seaborn is a statistical data visualization library, constructed over Matplotlib, and offers a streamlined interface to craft visually appealing and informative statistical graphics.

3.2 Machine Learning Models

The project entails the utilization of supervised machine learning algorithms, specifically tailored for classification tasks. Within this framework, various commonly employed classifiers play a crucial role. These classifiers encompass random forests, logistic regression, support vector machines, decision trees, and neural networks, collectively contributing to the project's analytical approach. Incorporating k-fold cross-validation significantly bolsters the reliability of model performance evaluations.

3.3 Evaluation Metrics

To gauge the effectiveness of the credit card default prediction models, a range of evaluation metrics are applied. The metrics include recall, accuracy, F1-score, precision, and Receiver Operating Characteristics (ROC) curve.

3.4 Development Environment

Google Colab. The project development is being carried out using Google Colab, a cloud-based Jupyter Notebook platform. This platform allows users to access GPU and TPU provided by Google, which can speed up the training of machine learning models for free. It is especially useful for training deep learning models, which can take a long time on regular computers. Moreover, it has features that allow people to work together as a team.

4 Related Work

The landscape of credit card default prediction has evolved significantly, driven by both traditional statistical methods and innovative machine-learning techniques. Extensive research has been conducted on various techniques for credit card default detection. Early approaches, as discussed by Yue[1], primarily centered on foundational models like Logistic Regression, laying the groundwork for subsequent advancements. Sayjadah et al.[2] provides a comprehensive review of machine learning algorithms, including Random Forests, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Deep Neural Networks (DNN). This study meticulously explores the strengths and limitations of each algorithm, offering valuable insights into their applicability in credit card default prediction scenarios. Building upon this foundation, Wang et al.[3] introduces novel data preprocessing techniques, notably the Synthetic Minority Oversampling Technique (SMOTE), addressing the challenge of imbalanced datasets. This significantly enhances predictive accuracy. These studies collectively contribute to the evolution of credit card default prediction techniques, encompassing historical context, diverse machine learning algorithms, effective data preprocessing, and the delicate balance between precision and recall. Our project leverages and extends these contributions to create a robust credit card default prediction system, aiming to strike the optimal balance between predictive accuracy, interpretability, and proactive identification of potential defaulters.

5 Contributions

The successful completion of the project on predicting credit card default is a collaborative effort where each team member contributed their expertise to various aspects of the project. Beginning with the literature review, the team collaboratively surveyed the papers [1], [2], [3], [4] which laid the foundation for understanding the problem domain. Sharan and Vamsi played pivotal roles in finding the right dataset collection and the team has decided to use "Default of credit card clients" data from the UCI machine learning repository. Feature Engineering, a crucial step in cleaning the data and enhancing the model performance was handled by Sathish and Prashanth. Furthermore, they performed One-Hot encoding to handle categorical variables, implemented feature scaling and feature selection using a covariance matrix and achieved feature transformation to improve predictive accuracy. The data is split into training data (70%) and test data (30%) for training and testing the model, which was managed by Jahnavi.

In the second phase of the project, we began with implementing the models. Prashanth and Sharan delved into K-Nearest Neighbors (KNN) showcasing their skills in proximity classification and achieving 81.2% accuracy. Moreover, improved the model performance to 82% accuracy by applying K-fold cross-validation to the KNN model. The statistical modeling, logistic regression along with cross-validation were developed by Sathish and Vamsi which achieved an accuracy of 82%. Jahnavi collaborated with Sharan on Random Forest yielding an effective ensemble model with 81.8% accuracy. Sathish and Prashanth focused on non-linear classification, i.e., Support Vector Machines. Sharan and Vamsi explored the complexity of Deep Neural Networks and implemented the network. In the third phase, the team focused on evaluating the models. The documentation was a collective effort from the team members to learn the LaTeX format and contribute to various sections of the final project report. Together, the team achieved a holistic understanding of the project, leveraging diverse skills and delivering an impactful project.

6 Accomplished Tasks

6.1 Dataset

An open-source dataset titled "Default of credit card clients" was obtained from the UCI machine learning repository[4], encompassing 30,000 distinct credit card clients in Taiwan. The dataset, observed between April and September 2005, includes 24 attributes for each observation. The first set of variables pertains to the client's personal information. Subsequent attributes comprise data related to payment delay history, providing insights into clients' past payment behaviors. The crucial variable for prediction is "default.payment.next.month," denoting whether clients will default in the following month (coded as 1 for defaulters and 0 for non-defaulters). With 6,636 out of 30,000 clients expected to default, this constitutes a binary classification challenge on an imbalanced dataset. The primary objective is to discern clients prone to credit card default in the upcoming month based on these diverse features.

6.2 Data Preprocessing and Feature Engineering

6.2.1 Handling Categorical Variables - One-Hot Encoding

One of the critical aspects of data preprocessing and feature engineering is handling the categorical variables. Categorical variables in our dataset include attributes such as "SEX," "EDUCATION," and "MARRIAGE." It's essential to encode these variables appropriately as the categories they represent do not have any inherent ordinal relationships. The objective is to remove any unintended orders that might be inferred by the model when using integer encoding. In the case of credit card default prediction, it is crucial to treat these variables as distinct categories without any ordinal significance. One-hot encoding impeccably transforms these nominal categories into binary representations, ushering in a new era of dummy variables that encapsulate each category's presence or absence. This utilization serves as a vital safeguard against any inadvertent misinterpretation of ordinal relationships that are fundamentally absent within the categorical variables.

6.2.2 Feature Scaling

To ensure a fair comparison among the attributes, a normalization process was employed to standardize their scales. The input variables often possess different units, causing variations in scale and magnitude. The MinMaxScaler() was used to rescale the numerical features, ensuring that their values fell within a consistent range from 0 to 1 (see Appendix A.1). This rescaling method involved mapping the minimum value of each numerical feature to 0 and the maximum value to 1, while proportionally transforming other values within the range of 0 to 1. Categorical variables, which had already been converted into one-hot vectors, were similarly rescaled to a uniform range from 0 to 1.

6.2.3 Feature Selection

The effectiveness of machine learning models can be notably influenced by the existence of robust correlations among features. Detecting and addressing feature correlations not only improves model performance but can also simplify the model by representing the same information with fewer attributes. Pearson's correlation coefficient, a widely employed tool for gauging the strength of linear association between pairs of random variables, spans a scale from -1 to 1 (see Appendix A.2), providing insights into the strength of linear correlations. It's important to note, however, that Pearson's coefficient primarily captures linear trends and may not fully capture nonlinear relationships.

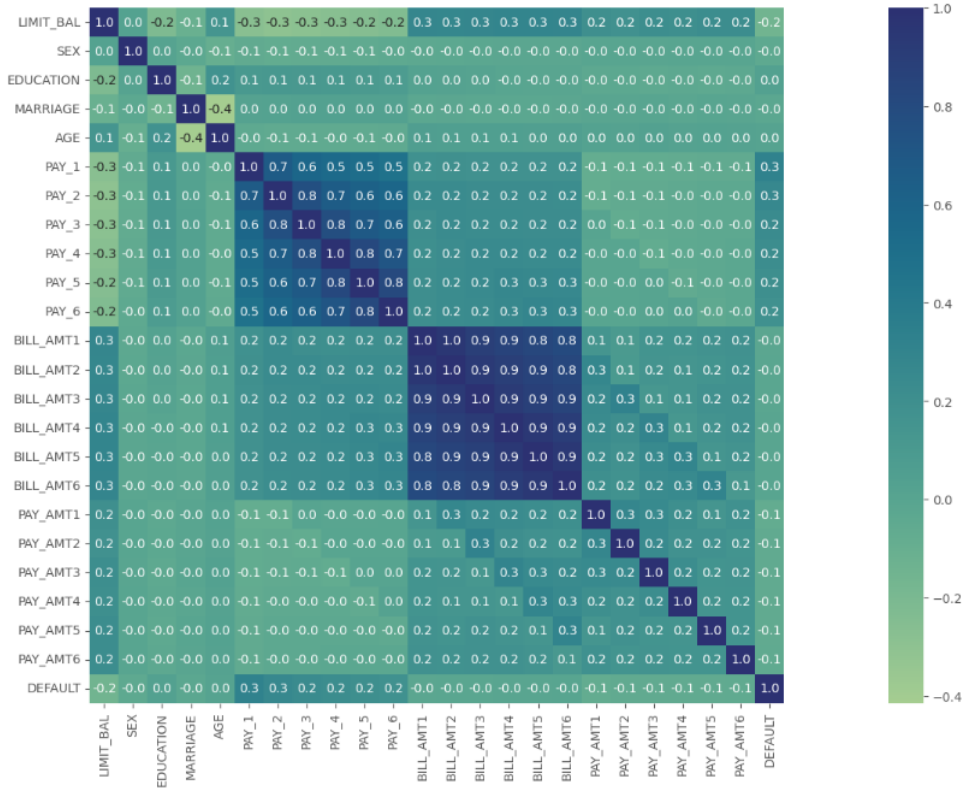

Figure 1: Correlation between features: 1 denotes the strongest correlation, while -1 signifies the weakest correlation between any given pair of features.

Upon examining the correlation matrix shown in Figure 1, it becomes evident that there are strong linear correlations among certain features within the dataset. This suggests that retaining all of these correlated features may not be justified, as they likely contain duplicative information. As a solution, we can consider removing or omitting any features with a Pearson correlation coefficient (denoted as P) equal to or greater than 0.92 concerning other predictors.

6.3 Data Splitting

The original dataset is split into a training set and a test set. The model is then exclusively trained on the samples from the training set and evaluated using unseen data, ensuring that its generalization ability is rigorously tested. To tackle the challenge of imbalanced data sets, a stratified sampling strategy is used during the data split process. This helps maintain the proportion of defaulters and non-defaulters in the final sets as close as possible to the initial distribution, which is important when working with highly skewed data.

Cross-validation, is another re-sampling method that enhances evaluation by repeatedly drawing samples from the training set. This approach enhances the robustness of the model's performance. The dataset is partitioned into k subsets or folds. In each iteration, $k-1$ subsets constitute the training set, while one subset serves as the validation set for performance assessment. This iterative process is repeated k times, with the validation subset changing in each round. Ultimately, the k performance estimates are averaged, reducing variability and providing a more reliable evaluation.

6.4 Evaluation Metrics

In classification tasks, evaluating a model involves computing various metrics using the predicted and true values, through a matrix referred to as confusion matrix as shown in Figure 2, visually depicting the performance of a classification model for better comprehension of the data. This matrix provides a clear and concise summary of true positives, true negatives, false positives, and false negatives, offering a visual tool to assess the effectiveness of the model's predictions.

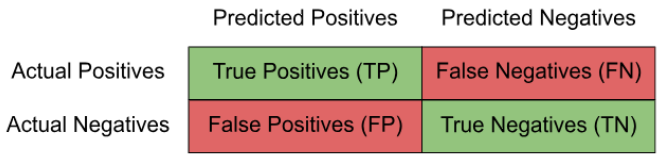

Figure 2: Confusion Matrix

The evaluation metrics used in the classification are:

Accuracy: The ratio of correct predictions to the total number of classified data points.

Precision: This metric calculates the percentage of cases that are correctly predicted among those that have a positive label. When it comes to minimizing false positives, precision is important.

Recall: Recall measures the extent to which a model identifies actual positives by correctly labeling them as True Positives. This metric becomes crucial when the emphasis is on minimizing false negatives.

F1-score: The harmonic mean of precision and recall, offering a balanced metric that considers both false positives and false negatives.

(see Appendix A.3)

The three aforementioned metrics are specific to individual classes, and deriving a global metric involves averaging these values across all classes in the dataset. This approach proves particularly beneficial when confronted with significantly unbalanced datasets, as it takes into account the learning performance for each class. In our analysis, the primary focus is on identifying customers who may default, with the positive class capturing the classifier's attention. As mentioned earlier, we will utilize the F1-score to determine the optimal configuration for each model in the subsequent sections.

6.5 Classification Algorithms

Now that we have standardized and preprocessed the data, it is ready to feed as an input to the models which can classify the data as default or non-default. To the current state of the art, we have approached two main classification models.

6.5.1 Logistic Regression

Logistic Regression, a binary classification algorithm, employs a parametric and discriminative approach, making it a vital component of the Generalized Linear Model, wherein the response variables follow a distribution based on the Bernoulli distribution. The training data is used to train this logistic regression model, which is subsequently utilized to make predictions on the test dataset. To optimize and assess the model performance, a k-fold cross-validation technique was employed on the training dataset with an optimal k value of 15.

In the context of our credit card default prediction study, Logistic Regression proves to be a compelling choice due to its interpretability and efficiency in handling large datasets. Despite its simplicity compared to more complex machine learning models, it showcases notable performance. Furthermore, its robustness is evident in the balanced precision, recall, and F1-score metrics, signifying its ability to effectively identify default cases while minimizing false positives or negatives.

6.5.2 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm is a supervised binary classification approach that utilizes proximity information to classify or predict the grouping of individual data points. In this study, Euclidean Distance is utilized to calculate the proximity between predicted and true labels. The model undergoes training on the training data and is tested across a varied range of k values, facilitating thorough performance evaluation and k selection.

A graph depicting the error rate for various values of k is presented in Figure 3. This plot aids in the determination of the ideal k value, which, in this instance, has been determined to be 150.

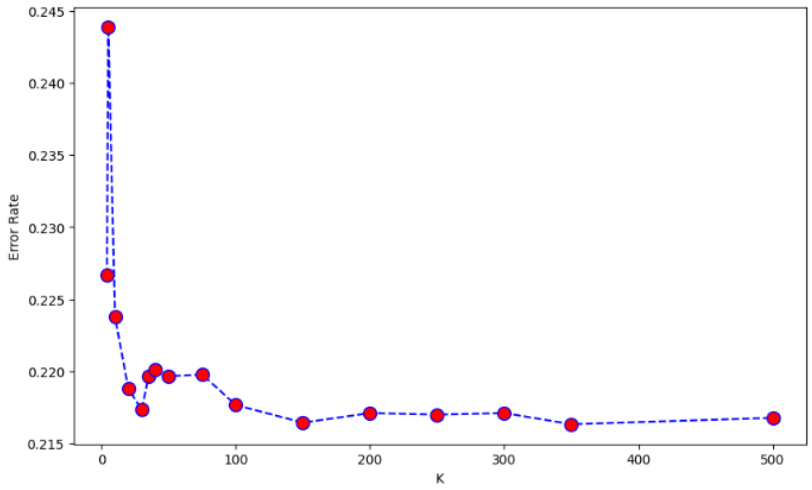

Figure 3: Error rate analysis for different k values in KNN.

6.5.3 Random Forests

Random Forest, an ensemble method, employs multiple decision trees trained on random subsets of data and features. Combining their predictions through averaging or voting, it creates stable results by minimizing noise and outliers. It also evaluates feature importance by assessing how shuffling or removing features impacts training, aiding tasks like predicting credit card defaults. Its ability to handle big datasets, capture nonlinear relationships, and highlight key features makes Random Forest a strong choice for predictive tasks.

Its ensemble approach, leveraging multiple decision trees with varied subsets of data and features, is expected to demonstrate robust performance, offering a promising solution to predict credit card defaults effectively. This method's versatility and ability to address overfitting while providing feature-importance insights make it a vital component in the comparison and selection of the most suitable model for this predictive task.

6.5.4 Support Vector Machine

The Support Vector Machine (SVM) is one of the most widely used classification algorithms designed to distinguish between two linear classes of data by introducing a hyperplane in the data space. Numerous hyperplanes could potentially separate the classes, but the primary goal is to identify one with the maximum margin — the greatest distance between data points of both classes. By maximizing this margin, the algorithm gains robustness, enhancing confidence in classifying future data points. Once the hyperplane is determined, predictions are made by assessing which side of the hyperplane a test point falls on. The data points that lie on the margin are known as support vectors.

Hard Margin: In the primal optimization problem, the objective is to maximize the margin when linear separation of the data is possible; otherwise, it is considered infeasible (see Appendix A.4). This formulation is pivotal in determining the optimal hyperplane for linearly separable data.

Soft Margin: In the dual optimization problem, we adopt the flexibility of allowing some data points to be predicted incorrectly to maximize the margin (see Appendix A.5). The introduction of a hyper-parameter, denoted as ξ, that is designed to relax the misclassification constraint, enabling the model to make some incorrect predictions and consequently expand the width of the margin.

6.5.5 Deep Neural Networks

Deep Neural Networks (DNNs) are a class of artificial neural networks that consist of multiple layers, enabling them to learn complex representations of data. These networks are designed to mimic the human brain's structure, utilizing an interconnected series of nodes or neurons across various layers. Its ability to automatically discover intricate patterns within the data and learn hierarchical representations allows us to handle sophisticated tasks such as classification, regression, and feature extraction. By harnessing the depth and interconnectedness of these networks, we aim to enhance the accuracy and robustness of our model, ensuring it can effectively generalize to unseen data and deliver more insightful predictions or classifications.

Regarding the specific DNN utilized in our project, it comprises an input layer, two hidden layers with 128 and 256 neurons, respectively, and an output layer. Each layer employs Rectified Linear Unit (ReLU) activation functions, known for their effectiveness in introducing non-linearity to the model. The output layer, employing the softmax activation function, is particularly tailored for multi-class classification tasks. Trained using the Adam optimizer and sparse categorical cross-entropy loss, the DNN demonstrates promising performance. After training over 10 epochs and evaluating against a test dataset, the model achieved a test loss of approximately 1.188 with a commendable accuracy of around 77.86%. These results highlight the effectiveness of the DNN in learning complex patterns within the data and making accurate predictions, establishing its viability for our project's objectives.

7 Results

Table 1: Performance Evaluation for discussed methods

Model	Accuracy	Precision	Recall	F1 Score	ROC
Logistic Regression	0.821	0.679	0.323	0.438	0.640
K-Nearest Neighbors	0.812	0.634	0.313	0.419	0.631
Random Forests	0.818	0.641	0.370	0.469	0.656
Support Vector Machine	0.819	0.659	0.343	0.451	0.647
Deep Neural Networks	0.778	0.416	0.053	0.094	0.516

In this machine learning project, we implemented a range of supervised learning algorithms to predict credit card default. The assessment involved implementing algorithms including Logistic Regression, Support Vector Machines, K-Nearest Neighbors, Random Forest, and a Deep Neural Network. Additionally, the utilization of k-fold cross-validation enhanced the accuracy of the models, underscoring the efficacy of this technique in improving overall performance. The test results aligned closely with the validation scores, indicating robust and reliable model performance. These models were trained and evaluated using a diverse set of metrics, including accuracy, precision, recall, F1-Score, and ROC-AUC.

Table 2: Performance Evaluation for discussed methods with k-fold Cross Validation

Model	Accuracy	Precision	Recall	F1 Score	ROC
Logistic Regression (15 folds)	0.819	0.687	0.334	0.450	0.645
K-Nearest Neighbors (22 folds)	0.820	0.658	0.319	0.430	0.636
Random Forests (10 folds)	0.8153	0.644	0.369	0.469	0.655
Support Vector Machine (10 folds)	0.815	0.645	0.370	0.455	0.656
Deep Neural Networks (10 folds)	0.780	0.425	0.056	0.098	0.524

Table 1 displays the evaluation results for the applied classification algorithms, and Table 2 displays the results with cross-validation. Logistic Regression achieved an accuracy of 82.1%, demonstrating a balanced performance with a precision of 67.9%, and a recall of 32.3%. Furthermore, K-fold cross-validation is implemented for logistic regression with 15 folds which improved the F1-Score to 0.450. Although the KNN algorithm with 22-fold cross-validation has shown the optimal accuracy (82%), its F1-Score is less than logistic regression. The Random forest seems to have a consistent F1-score with an accuracy of 81.8%. The support vector machine achieved 81.9% which was slightly reduced when k-fold (10) cross-validation was integrated. Finally, a Deep Neural Network with 2 hidden layers has achieved 78% accuracy which is comparatively less than other supervised algorithms.

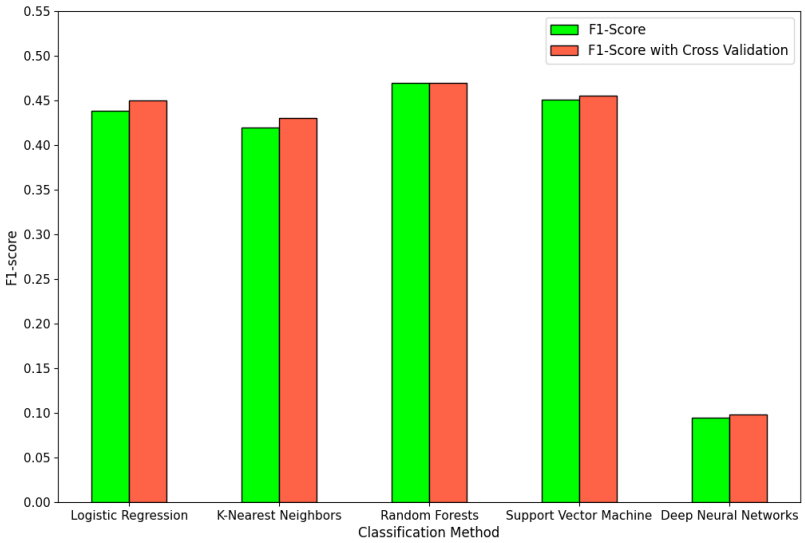

Figure 4: Comparative analysis of F1-scores across classification methods.

The barplot shown in Figure 4 gives us the summary of the results of F1-scores given by the different classification algorithms whose models are trained with their best hyperparameters.

8 Conclusion

This project involved the exploration and implementation of various supervised algorithms on the UCI dataset to construct a classification model for predicting whether credit card clients will default in the next month. The data preprocessing step is crucial for enhancing results compared to the original dataset. Among all the assessed algorithms, Random Forests emerged as the most effective for our case, exhibiting high accuracy, precision, and the best F1-score.

Our project successfully achieved all the objectives outlined in the initial proposal. The project not only shed light on the intricacies of predictive modeling in finance but also emphasized the importance of understanding the trade-offs among various machine learning methodologies. Moving forward, this project's findings pave the way for further refinements in model architectures, feature engineering strategies, and data preprocessing techniques, fostering a continuous quest for more accurate credit card default prediction systems.

Appendix A

A.1 Feature Scaling

The feature scaling is given by the following formula,

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

where X_{min} is the minimum value on the column, X_{max} is the maximum value on the column, and X_{scaled} is the scaled input data.

A.2 Feature Selection

Pearson's correlation coefficient is given by,

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \in [-1, 1] \subset \mathbb{R}$$

where $Cov(X, Y)$ is the Covariance and σ_X and σ_Y are the standard deviations of X and Y respectively.

A.3 Evaluation Metrics

With True Positives as TP, True Negatives as TN, False Positives as FP, and False Negatives as FN, the metrics accuracy, precision, recall, and F1-score are calculated using the below,

$$Accuracy = \frac{Number\ of\ correct\ samples}{Total\ samples\ in\ the\ dataset} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{Number\ of\ samples\ correctly\ assigned\ to\ class\ c}{Number\ of\ samples\ assigned\ to\ class\ c} = \frac{TP}{TP + FP}$$

$$Recall = \frac{Number\ of\ samples\ correctly\ assigned\ to\ class\ c}{Number\ of\ samples\ actually\ belonging\ to\ class\ c} = \frac{TP}{TP + FN}$$

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

A.4 Hard Margin in SVM

$$\min_{w,b} \frac{1}{2}||w||^2 \qquad s.t.\ y_i(w^T x_i + b) \geq 1, \forall i$$

where w is a vector normal to the hyper plane and b is an offset and for the positive classes $(w^T x_i + b)$ is > 1 and y_i is > 1 and for the negative classes $(w^T x_i + b)$ is < 1 and y_i is < 1, hence for every class the product $y_i(w^T x_i + b) > 1$.

A.5 Soft Margin in SVM

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i}^{n} \xi_i \qquad s.t.\ y_i(w^T x_i + b) \geq 1 - \xi_i\ and\ \xi_i \geq 0, \forall i$$

The variable ξ is a hyperparameter introduced to relax the misclassification constraint.

References

[1] Y. Yu, "The Application of Machine Learning Algorithms in Credit Card Default Prediction," 2020 International Conference on Computing and Data Science (CDS), Stanford, CA, USA, 2020, pp. 212-218, doi: 10.1109/CDS49703.2020.00050.

[2] Y. Sayjadah, I. A. T. Hashem, F. Alotaibi and K. A. Kasmiran, "Credit Card Default Prediction using Machine Learning Techniques," 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), Subang Jaya, Malaysia, 2018, pp. 1-4, doi: 10.1109/ICACCAF.2018.8776802.

[3] T. Wang and Y. Zhao, "Credit Card Fraud Detection using Logistic Regression," 2022 International Conference on Big Data, Information and Computer Network (BDICN), Sanya, China, 2022, pp. 301-305, doi: 10.1109/BDICN55575.2022.00064.

[4] Yeh, I-Cheng. (2016). default of credit card clients. UCI Machine Learning Repository. https://doi.org/10.24432/C55S3H.