Credit Card Payment Defaults

Nicolai Jacobsen, Jacob Kulik, David Pogrebitskiy, Gavin Wainwright

Northeastern University, Boston, MA, USA

Abstract

Machine learning models are frequently used in the financial services industry, with one of the use-cases being predicting credit defaults. The models are able to analyze large amounts of data and learn complex patterns that are not always easily discernible by human analysts. This leads to faster decision making on credit risk assessment and saves valuable time, costs, and effort for the company. The goal of this project was to evaluate and compare various machine learning models and identify their strengths and shortcomings in predicting credit card payment defaults. Five different ML models were tried and tested, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Logistic Regression and Neural Network. Additionally, some of the algorithms were constructed manually and their results compared to their scikit-learn counterparts. The Random Forest model was the most successful, with an accuracy of around 71%, closely followed by the Decision Tree with an accuracy of around 70%. The least successful model was the K-Nearest Neighbor, which only had an accuracy of around 60%. These models were also extremely time-intensive due to their limitations. Overall, we feel that a financial institution should resort to Random Forest or Decision Tree models for credit applications or balance limits.

Introduction

The financial services industry involves bringing together individuals with money and those who need it. Traditionally, banks receive money from their clients and make a profit through investments. They also offer credit card services, where individuals and businesses can make purchases with the bank's money, paying off the services at the end of every month. Due to the ease of use, credit cards have become a fundamental component of the economy, with billions of dollars' worth of transactions being processed every day.

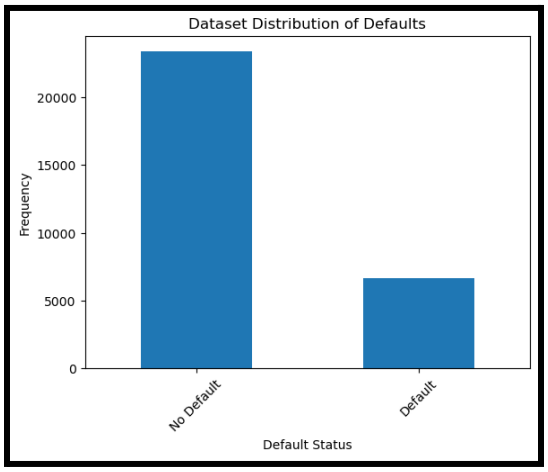
However, credit card usage also comes with inherent risks. The primary risk associated with the use of credit cards is the possibility of defaulting, or a borrower being unable to pay back the lender. Credit card defaults can have severe consequences for both borrowers and lenders, with borrowers facing financial ruin and lenders experiencing losses. In order to mitigate these risks, financial institutions utilize statistical models and risk assessment techniques to predict the likelihood of defaults.

Our dataset is taken from the UCI Center for Machine Learning and Intelligent Systems, and covers the Default of Credit Card Clients in Taiwan from April to September, 2005. There are 300,000 unique credit card holders in this dataset, a sizable sample size with more information than many other publicly available credit card datasets. This dataset also has a very clear attribute key, unlike online datasets that contain either a much smaller number of attributes or ones that are not clear or pre-screened/normalized.

With the development of machine learning technology, banks have moved away from more traditional methods in the case of balance limits or accepting new clients. However, this has brought up many ethical and legal questions regarding the fairness of these practices. The Equal Credit Opportunity Act of 1974 (ECOA) was put in place to guarantee access to credit and guard against discrimination (Klein, 2019). Here, a financial institution cannot use certain factors such as race, sex, national origin, and age against you. However, the issue of AI being unfair is still brought up, considering the lack of emotions and focus on quantitative values.

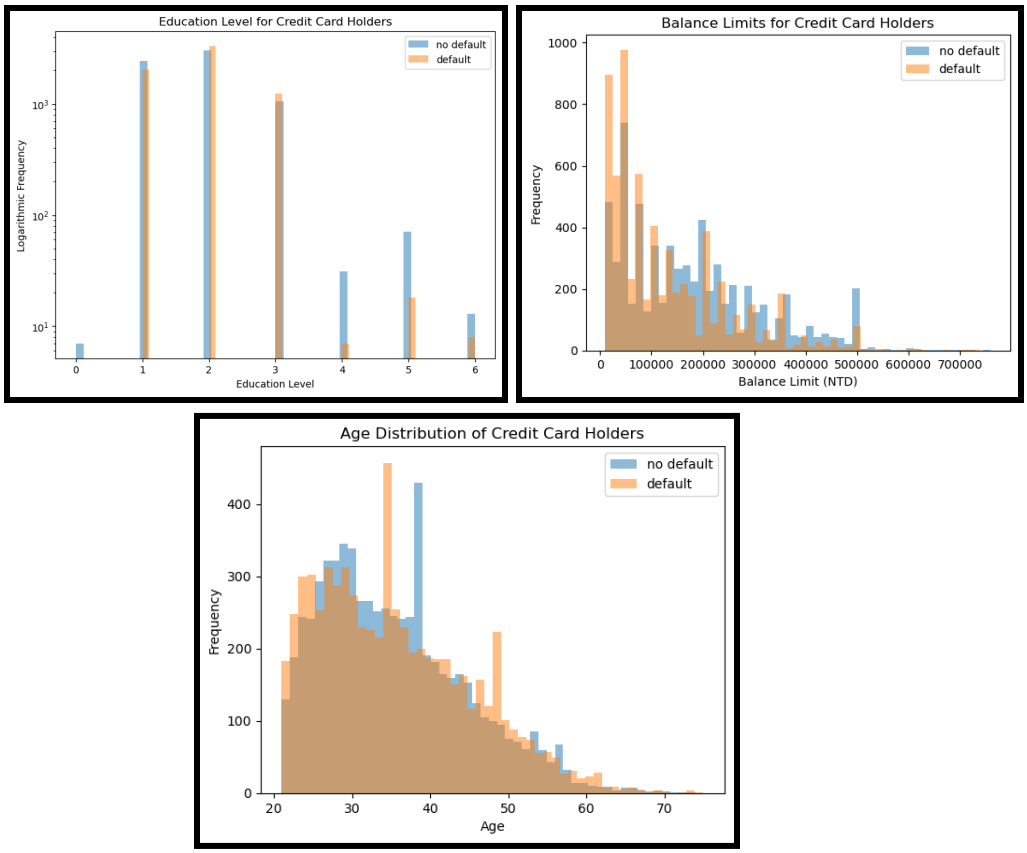
Data Analysis

The data encompasses 300,000 unique Taiwanese card holders and 23 different explanatory variables, as well as whether or not the individual defaulted. The explanatory variables range from demographic information, such as age, sex, education level, and marital status, to bill and payment history from the 6 months prior. Credit is also an important factor included in the dataset. Since the real probability of default is unknown, all factors were given an exploratory analysis.



Distribution of default data points in the dataset

Due to the imbalance distribution of defaulted vs. non-defaulted cardholders, the dataset was tweaked to get an equivalent sample of non-defaults. Factors such as age, marital status, and balance limit were explored before jumping into machine learning methods.



Dataset Visualizations

Through this basic exploratory analysis, it became clear that more older and well educated clients with higher limits defaulted at lower rates, but all card holders followed very similar trends. These distributions are not enough for a financial institution to use when evaluating a potential new client.

Methods

Feature Selection

By the above plots, we can see that the data was wildly imbalanced containing far more samples with a target of 0 than a target of 1. Knowing that this magnitude of imbalance can drastically affect a machine learning model, we decided to randomly select a set of equal size from the data with label 0 to balance out the amount of data in each class, giving us more reasonable results. Additionally, Some of our machine learning models are distance-based so each column of the feature matrix was scaled to have mean of 0 and standard deviation of 1 when necessary.

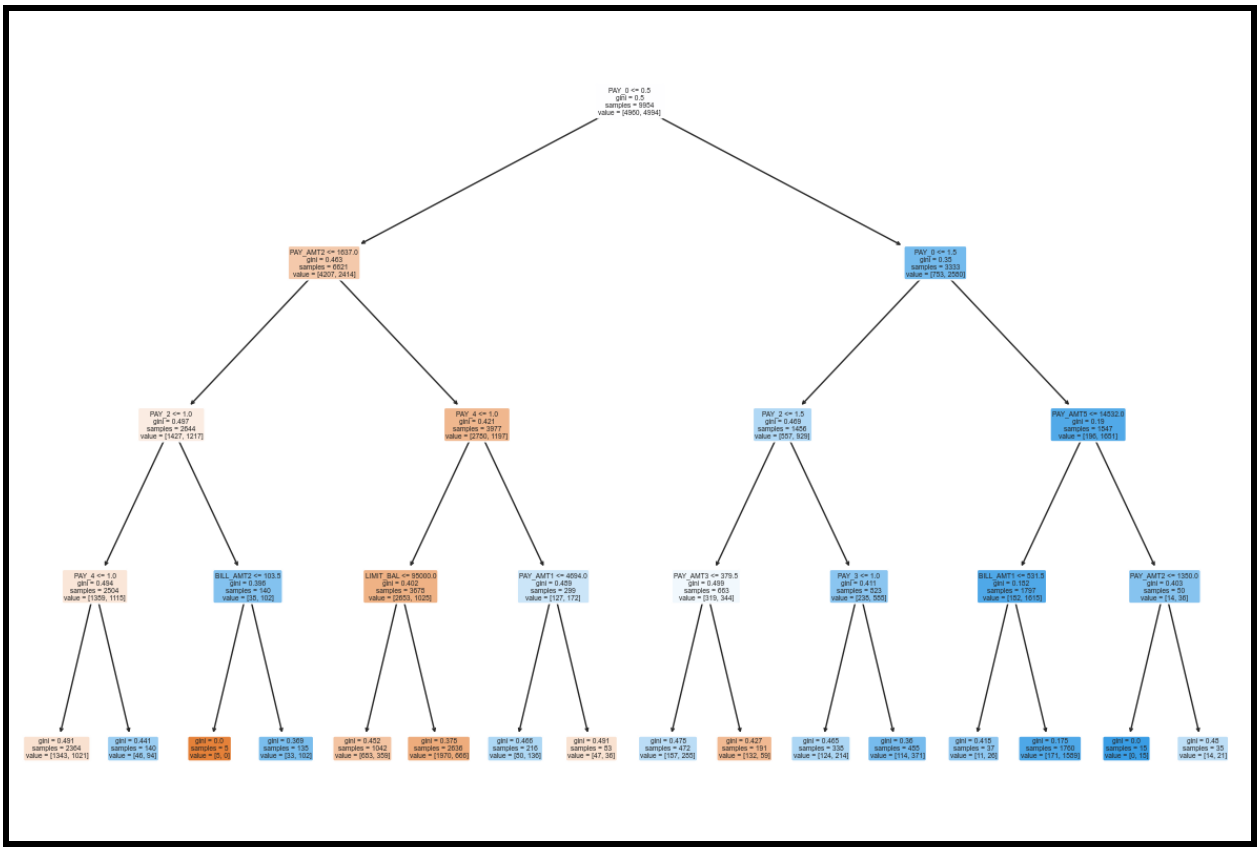
K-Nearest Neighbors

The K-Nearest Neighbors algorithm has many benefits when predicting credit card defaults. Credit default prediction is often a complex and non-linear problem, as it involves various factors without any specific distribution or relationship between the features. KNN is suitable for capturing these non-linear patterns in the data. Additionally, Credit default prediction models may need to be updated frequently as more data becomes available. This supports the use of KNN, which is considered a lazy learner, meaning that it can easily incorporate new data points without retraining the model. Initially, a KNN algorithm was constructed without the use of a library. Due to the iterative nature of the algorithm and cross-validation, the run time was high. Hence, it was important to limit the number of hyperparameters the model could take. Therefore, the algorithm was constructed to take only two hyperparameters, the value of k and the distance model. Three distance models were used, Euclidean, Manhattan, and Cosine distance. Cross-validation was employed by dividing the data in 10 separate folds, using 9 folds as training data and the remaining fold as the test set. The highest accuracy achieved by the model was 68.52% with an F1-score of 66.67%. The hyperparameters used in this result were Euclidean distance and k=23.

Without using a built-in KNN model, we noticed an extremely time-intensive run process, especially with multiple K values and distance measurements. Our next step was to compare our results to accuracies found when employing sci-kit learn. Similarly to the manual method, we used a train test split with a proportion of 0.1 to fit our classifier. Here, we made use of Grid Search CV, where the parameters of the estimator are optimized by cross-validated grid-search over a parameter grid. Our parameter grid here consisted of various K values, weights, and distance measurements. After fitting the training values onto our grid, the best parameters and associated accuracy scores were reported. With our testing, a K value of 21, weight method of distance, and metric of euclidean achieved a mean accuracy of 68.59%.

Decision Tree

Trees are data structures that are widely seen across the field of computer science and mathematics. In machine learning, we utilize trees to develop classification models known as decision trees. Decision trees are debatably one of the most interpretable machine learning algorithms out there because of its intuitive visualization and decision flow. Although more computationally expensive than other ML models, decision trees are extremely fast at churning out predictions. Utilizing a custom implementation and a pre-build Scikit-learn implementation, we have very promising, comparable results. After running Scikit-learn's GridSearchCV, we found our optimal hyperparameters to be a max depth of 4, minimum samples to split of 2, and gini evaluation criterion. This combination yields a test accuracy of 70.01% on our custom implementation and 69.77% on the Scikit-learn implementation. Up to this point, this is our most promising model.



Decision tree of depth 4 where orange nodes predict a non-defaulting customer (0) and blue nodes predict a defaulting customer (1)

Random Forest

Decision trees are often susceptible to overfitting because each split location is solely calculated from the training data itself. One common solution is utilizing the ensemble method, random forest, to build a plethora of different trees, each with random subsets of the training data, and allowing them to vote on the final classification. In the randomized process of creating trees, generalizations begin to appear about the data, increasing predictive accuracy and reducing overfitting. As all models come with tradeoffs, the biggest tradeoff of random forests is computational time. Building hundreds of decision trees is very costly and can range from minutes to days of runtime. Because of this, if your dataset is large enough, random forest can potentially be impossible to train in a timely manner. With our dataset, we were able to run GridSearchCV to find optimal parameters. With 200 trees, a max depth of 5, minimum samples to split of 2, and a gini evaluation criterion, we were able to yield a cross-validated accuracy of 71.07%.

Logistic Regression

Logistic regression is one of the most well-known models for classification in machine learning. Its popularity can be mainly attributed to its interpretability and efficiency. In terms of

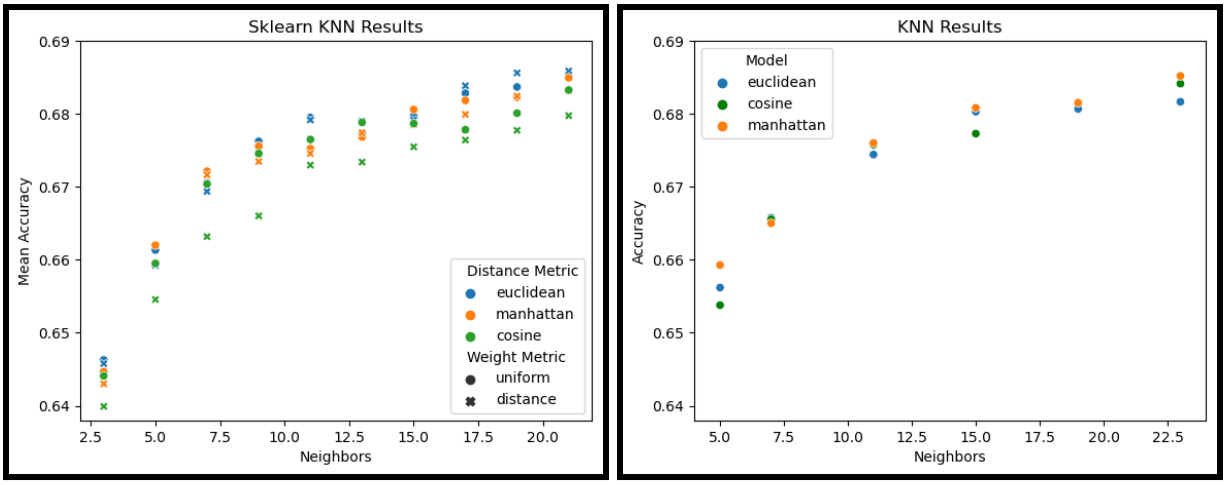
interpretability, the coefficients/weights of the linear model clearly describe the relationship between features and outputs, allowing the user to see exactly which features push the classification in a certain direction. Additionally, the probabilities that are part of the model's output show the likelihood of each class being the result. Computational efficiency is also a large reason for this model's popularity. Because of its simple and concrete mathematical foundation, gradient descent, logistic regression can be quickly trained on large datasets. Cross-entropy loss was differentiated to yield the gradient descent equation for this model. Then, through a process of iteration, the model continues to reduce loss until convergence, yielding the best weights for the linear model. After many iterations, we found that our loss function's surface was relatively simple to descend, yielding us with convergence regardless of the step size and number of epochs. In the end, we were able to utilize K-fold cross validation to prevent overfitting, yielding a cross-validated accuracy of 67.05% and an f1-score of 65.90%. We are confident in our custom implementation because it nearly mirrored the results from scikit-learn (66.31% accuracy).

Neural Network

Neural networks have gained popularity because of their ability to learn from large, complex datasets. One of the biggest advantages is their ability to combine different modeling approaches. They can utilize probabilistic approaches, proximity approaches, linear approaches, and non-linear approaches, allowing them to model complex relationships that otherwise couldn't be modeled. Additionally, unlike traditional models, neural networks have the ability to tune hyperparameters overtime, giving them the ability to fix their own mistakes and improve over time. Unfortunately, because of their ambiguous nature, neural networks require experimentation and fine-tuning of the overall structure in order to yield promising results. In our case, after many iterations of different numbers of layers, different activation functions, different optimizers, we landed on a three-layer sequential model. The first layer, a dense input layer, consisted of 25 nodes and utilized the Rectified Linear Unit (ReLU) activation function. The next layer was identical to the first, also consisting of 25 nodes with the ReLU activation function. Finally, the output layer consisted of 2 nodes, activated by the softmax function, yielding probabilities of each class. Despite the fact that the SoftMax activation function typically isn't used for classification problems, it yielded better results than a previous iteration of our neural network that contained a hidden layer activated by the Sigmoid function. With a learning rate of 0.0005, 50 epochs, a batch size of 32, and a validation split of 0.2, we retrieved a test accuracy of 70.70%, a very promising result.

Analysis

Firstly, we wanted to explore the differences in model accuracy when using sklearn versus not including it. Both models led to similar accuracy values, even though they found different distance metrics to be best. Sklearn used Euclidean distance, while KNN used Manhattan. The logarithmic trend with increasing K value was a commonality, however. The most staggering difference was the run time. While each fold took fractions of a second with sklearn, and the model had a run time less than 5 minutes, building the model from scratch led to an almost 5-hour run time.



KNN accuracies with and without sklearn

	Accuracy	Run-Time (hh:mm:ss)
K-Nearest Neighbors	68.52%	04:57:44
Scikit-learn KNN	68.59%	00:04:31
Decision Tree	70.01%	00:14:00
Scikit-learn Decision Tree	69.77%	00:01:00
Random Forest	71.07%	00:08:00
Logistic Regression	67.05%	00:00:24
Neural Network	70.70%	00:00:13

Final Results

The highest accuracy was achieved by the Random Forest model at 71.07%. This is slightly higher than the Neural Network, which has the second highest accuracy at 70.70%, closely followed by the Decision Tree at an accuracy of 70.01%. Thus, it can be concluded that the use of the random forest model, decision trees, and neural networks in regards to predicting credit card payment default proved to be the most successful. The ensemble learning of the Random Forest, which reduces the risk of overfitting, provided a marginally more accurate prediction.

The main challenge regarding the KNN algorithm was that it is sensitive to the distance model used and the value of K. If the value of K is too small, the algorithm may overfit to the training data, while if the value of K is too large, the algorithm may underfit the data. Additionally, KNN is a lazy algorithm and requires significant time and memory to compute the distance between all data points, especially with a large dataset.

Decision Trees and Random Forests are other powerful machine learning algorithms. Decision Trees work by recursively splitting the data based on the most important features until a stopping criterion is met. Random Forests, on the other hand, work by combining multiple Decision Trees and aggregating their outputs to improve the accuracy and reduce overfitting. In the case of our analysis, these two methods proved to be more accurate. Although we did not remove the gender attribute from the dataset, it was not included in the final decision trees. The advantages of Decision Trees and Random Forests are their ability to handle non-linear data and their ability to identify the most important features in the data. Although they are also relatively slow to train and have low memory requirements, they are able to produce results very quickly.

Conclusions

Predicting credit card payment default is a critical task for financial institutions, and machine learning algorithms have proven to be very useful in this regard. Through our analysis, we have shown that Decision Trees and Random Forests are effective methods of predicting credit card default. The Random Forest algorithm, in particular, performed the best with an accuracy of 71.07%. However, it is important to note that these algorithms are not foolproof, and there is always room for improvement in terms of predictive accuracy and bias mitigation. Additionally, it's important to consider the ethical and legal implications of using these algorithms in credit scoring. Financial institutions must adhere to regulations such as the ECOA to ensure fairness and equal access to credit. Additionally, careful feature selection and bias mitigation techniques must be employed to avoid potential discrimination and improve the accuracy of credit scoring models. As the financial industry continues to evolve, machine learning models and algorithms will undoubtedly play an increasingly important role in credit risk assessment, and it is important to use them responsibly and with caution.

Author Contributions

Our Credit Defaults project compared the performance of various supervised machine learning models and different parameters. Contributions were split evenly across different exploration, machine learning methods, and writing. Dave and Jacob conducted initial explorative analyses of the dataset. Nicolai and Jacob developed the KNN methods on the data, David developed the Decision Tree, Random Forest, and Logistic Regression models, and Gavin

developed the Neural Network. The report was contributed to by all members. Overall, work was split up equally based on each member's interests and skills.

References

Arora, Saurabh, et al. "Prediction of Credit Card Defaults through Data Analysis and Machine Learning Techniques." *Materials Today: Proceedings*, Elsevier, 9 June 2021, https://www.sciencedirect.com/science/article/pii/S2214785321035148.

Gui, L. (2019). Application of Machine Learning Algorithms in Predicting Credit Card Default Payment. UCLA. ProQuest ID: Gui_ucla_0031N_17808. Merritt ID: ark:/13030/m57t2mts. Retrieved from https://escholarship.org/uc/item/9zg7157q

Klein, Aaron. "Credit Denial in the Age of AI." Brookings, Brookings, 9 Mar. 2022, https://www.brookings.edu/research/credit-denial-in-the-age-of-ai/.

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.