CREDIT RISK ANALYSIS BASED ON LOGISTIC REGRESSION & OTHER MACHINE LEARNING MODELS, ON AMERICAN EXPRESS DATASET

Souhardya Mitra

Abstract

Credit risk analysis is a crucial task for financial institutions as it enables them to determine the likelihood of default for potential borrowers. In this paper, we analyze credit risk using logistic regression and other machine learning models on the American Express dataset. Our aim is to identify the best performing model in predicting credit card defaults and to determine the most important variables in credit risk analysis. Our study shows that XGBoost is the best performing model, with an accuracy of 0.97, precision of 0.91, F1-score of 0.91, AUC value of 0.92. Logistic Regression and other models also performed well, but not as well as XGBoost. Our findings indicate that the most significant variables in predicting credit card defaults are credit score, credit limit utilization, and number of days employed. Furthermore, we find that the age of the borrower is not a significant factor in predicting credit card defaults. This highlights the importance of considering other variables when analyzing credit risk. Our study provides practical implications for financial institutions in improving their credit risk analysis models. By using machine learning techniques such as XGBoost, they can better identify and manage credit risk, thus reducing their losses due to defaults.

Keywords: Credit risk analysis, logistic regression, XGBoost, machine learning.

1. Introduction

Credit risk analysis is a critical task for financial institutions, as it allows them to identify potential borrowers who may be more likely to default on their loans or credit card payments. To categorize applicants as good or bad, the banking system assesses the probability of default. The applicants who fall into the good category are deemed to have a high likelihood of repaying their loans to the bank, whereas those in the bad category are viewed as having a low probability of repayment and are considered defaulters. The benefits of the reliable credit risk dataset is it reduces the cost of credit scoring, good decision making in very less time and avoid less risk associates with loan collection (Pandey et al, 2017).

Traditionally, financial firms have employed classical linear, logit, and probit regressions to model credit risk (Altman, 1968). In recent years, the use of machine learning models for credit risk analysis has gained significant attention due to their ability to handle large amounts of data and complex relationships among variables. In 2000, Galindo and Tamayo used CART decision tree, KNN and probit models to detect defaults

on mortgage-loan data. Khandani et al. (2010) suggest a machine-learning method that utilizes decision trees and SVM in consumer lending. The authors test this technique on real lending data and find that it results in cost savings of up to 25%. In 2016, Butaru et al. examine consumer delinquency in six different banks by employing logistic regression, decision trees, and random forest algorithms.

In this paper, we focus on the application of logistic regression and other machine learning models to analyze credit risk using the American Express dataset. The dataset contains information on a sample of credit card holders, including their demographic characteristics, credit card usage patterns, and repayment history. We aim to compare the performance of different machine learning models in predicting credit risk and assess the factors that contribute to credit card default. Specifically, we will use logistic regression, decision trees, random forests, KNN and gradient boosted models to build predictive models and evaluate their accuracy and interpretability.

The results of our analysis can provide valuable insights for financial institutions in managing credit risk and designing targeted interventions to prevent default. Additionally, our study contributes to the literature on the use of machine learning for credit risk analysis and provides a practical example of its application to a real-world dataset.

2. Theoretical background

2.1. Models

2.1.1. Logistic regression

Logistic regression is a type of regression analysis that is applied when the dependent variable is a binary variable with two possible values, typically indicating the presence or absence of an outcome event. The independent variables can be continuous, categorical, or a combination of both. Additionally, it does not make assumptions about normal distributions for either the dependent variable or the error terms.

The form of the model is:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

A logistic function is used to convert the predictor variable values into probabilities. This has been widely used in credit scoring applications due to its simplicity (Satchidananda et al, 2006). Binary logistic regression has been utilized by Altman & Sabato (2007), and Yazdanfar & Nilsson (2008) to anticipate the likelihood of SMEs' default. Despite proof that artificial intelligence methods have improved the results of credit risk analysis in comparison to traditional statistical techniques (Abellan &

Castellano, 2017), logistic regression is still popular because of its straightforward implementation and balanced error distribution.

2.1.2. Decision trees

Decision tree is a predictive model which maps the observation about an item represented in branches to conclusion about a target value represent in leaf nodes (Pandey et al, 2017). The decision tree consists of interior nodes, each corresponding to a variable, and arcs to a child represent a possible value of that variable. A leaf represents the predicted value of the target variable based on the values of the variables represented by the path from the root. In decision tree learning, the decision tree describes a tree structure where the leaves represent classifications and branches represent combinations of features that lead to those classifications. A decision tree is created by dividing the source set into subsets based on an attribute value test.

The use of decision trees has become prevalent in the credit risk analysis sector owing to their capacity to manage both categorical and numerical data, as well as their interpretability and ability to handle nonlinear connections between variables. Decision trees are also a dependable tool for credit risk analysis because they can handle missing data and outliers.

2.1.3. Random forest

Breiman (2000) proposed a method called random forests, which involves building a collection of decision trees in randomly selected subspaces of the data to create a predictor ensemble. This is a well-known ensemble learning method that enhances model accuracy and stability by combining various decision trees. A random forest is a classifier combining of a set of tree-structured classifiers. The performance of a random forest model is determined by the quality of its constituent tree classifiers and the degree of correlation among them. The model is capable of predicting the output value, either 1 or 0, for a given input X by utilizing the appropriate number of trees. The prediction error is minimized by using a stopping criterion.

Random forests have been widely used in diverse fields, including finance, healthcare, and marketing, owing to their capability to manage high-dimensional data with complex connections between characteristics. It is suggested that random forest models are more effective than traditional approaches like logistic regression or even k-nearest neighbors. Similarly, Gahlaut & Singh (2017) concluded that after comparing various algorithms such as decision tree, support vector machine, adaptive boosting model, linear regression, random forest, and neural network for constructing predictive models, the most effective algorithm for risky credit classification is the random forests algorithm.

2.1.4. Gradient boosted machines

Decision trees are the central element of gradient-boosted machines, a machine learning algorithm that is widely used for its efficiency, accuracy, and interpretability. This algorithm, also known as gradient boosting decision tree (GBDT), is considered a top-performing technique for various machine learning tasks, including multi-class classification, click prediction, and learning to rank.

There are some famous gradient boosting models commonly used such as XGBoost, CatBoost or LightGBM. XGBoost is an abbreviation for Extreme Gradient Boosting, and it is the machine learning model that is frequently employed for predictive tasks. This model was developed by associate professor Chen (2016) at Carnegie Mellon University, USA. CatBoost stands for Category Boosting, which is one of the latest models in gradient boosting library. The algorithm assigns a group of possible feature-split pairs to the leaf as the split, and selects the split that incurs the smallest penalty. Balanced trees offer advantages such as faster computation and overfitting control. LightGBM is a machine learning model that is also called Light Gradient Boosted Machine. Like XGBoost, it uses an asymmetric decision tree. However, unlike XGBoost, it grows the tree leaf-wise, rather than level-wise. This means that LightGBM produces smaller and faster models. This model was developed by Microsoft in 2017, it was first introduced in the paper of G.Ke et al. in same year.

2.1.5. K-nearest neighbors (KNN)

According to Pandey et al (2017), KNN is the non-parametric method used for classification and regression. It involves a training set of both positive and negative cases. If KNN is utilized for classification, the predicted class can be determined by selecting the class that occurs most frequently among the k-most similar instances. Essentially, each instance provides a vote for its class, and the class with the most votes is considered the prediction.

Although it can be utilized for regression or classification problems, it is usually used as a classification algorithm, operating on the assumption that comparable points can be found near each other. For classification problems, a class label is assigned based on a majority vote, which is technically referred to as "plurality voting," but the term "majority vote" is more frequently used in literature. This involves using the label that appears most frequently around a given data point.

The KNN algorithm is both simple and accurate, making it one of the initial classifiers that a new data scientist learns. It is also adaptable to changes, meaning that it adjusts itself to new data added to the model. Furthermore, the algorithm has a limited

number of hyperparameters, requiring only a k-value and a distance metric, which is less than other machine learning algorithms.

2.2. Weight of Evidence (WOE) and Information value (IV)

2.2.1. Weight of Evidence (WOE)

The weight of evidence (WOE) is a measure that determines how influential an independent variable is in predicting the dependent variable. Originally used in the credit scoring industry, it is commonly used to assess the distinction between good and bad customers. In this context, "bad customers" are those who have failed to repay a loan, while "good customers" are those who have successfully repaid their loans.

The formula for WOE is:

$$WOE = \ln\left(\frac{Distribution\ of\ Goods}{Distribution\ of\ Bads}\right)$$

To calculate the Weight of Evidence (WOE) for a continuous variable, begin by dividing the data into 10 groups (or fewer, based on the distribution). Next, determine the number of events and non-events in each group (bin). Calculate the percentage of events and non-events in each group. Finally, compute WOE by taking the natural logarithm of the ratio of the percentage of non-events to the percentage of events.

To calculate the Weight of Evidence (WOE) for a categorical variable, do not need to split the data into different parts. Instead, skip the first step and proceed directly to calculating the number of events and non-events in each group (bin), calculating the percentage of events and non-events in each group, and finally, computing WOE by taking the natural logarithm of the ratio of the percentage of non-events to the percentage of events.

WOE has several advantages, including the ability to handle outliers. For example, if a continuous variable such as annual salary has extreme values over 500 million dollars, these values can be grouped into a class of (for instance) 250-500 million dollars, and WOE scores can be used instead of raw values. Additionally, missing values can be binned separately, making it easy to handle them. Because WOE transformation handles categorical variables, there is no need for dummy variables. Moreover, WOE transformation allows to establish a strict linear relationship with log odds, which is difficult to achieve using other transformation methods like log or square root. In conclusion, if do not use WOE transformation, we may need to try out several transformation methods to establish this relationship.

2.2.2 Information value (IV)

Information Value or IV is a useful tool to determine the importance of variables in a predictive model. It helps to rank the variables based on their significance. The formula to calculate IV is as follows:

$$\text{IV} = \sum(\%\ \text{of non\_events} - \%\ \text{of events}) \times \text{WOE}$$

Table 1 provides conventional values of IV statistics (Siddiqi, 2006)

Table 1. Conventional Interpretation of IV

Information Value (IV)	Predictive Power
< 0.02	Useless
0.02 - 0.1	Weak predictors
0.1 - 0.3	Medium predictors
0.3 - 0.5	Strong predictors
> 0.5	Suspicious

3. Data

3.1. Data description

The data used in this paper comes from the "AmExpert 2021 CODELAB - Machine Learning Hackathon" competition hosted on the online coding platform, HackerEarth. The dataset can be accessed **here**, belongs to American Express, a company that provides customers with various payment products and services.

The original dataset consisted of 45528 rows and 19 columns, but for this study, a subset of 30000 rows and 19 columns was used. Table 2 provides a detailed explanation of each column in the dataset.

Table 2. Columns Description

No	Variable	Meaning
1	customer_id	Customer ID
2	name	Name of customer
3	age	Age of customer
4	gender	Gender of customer: Female (F), Male (M)
5	owns_car	Whether own car or not? (T/F)
6	owns_house	Whether own house or not? (T/F)
7	no_of_children	Number of children
8	net_yearly_income	Net yearly income
9	no_of_days_employed	Number of days employed
10	occupation_type	Occupation type
11	total_family_members	Total family members
12	migrant_worker	Is migrant worker or not? (0/1)
13	yearly_debt_payments	Yearly debt payments
14	credit_limit	Credit limit
15	credit_limit_used(%)	Percentage of credit limit used

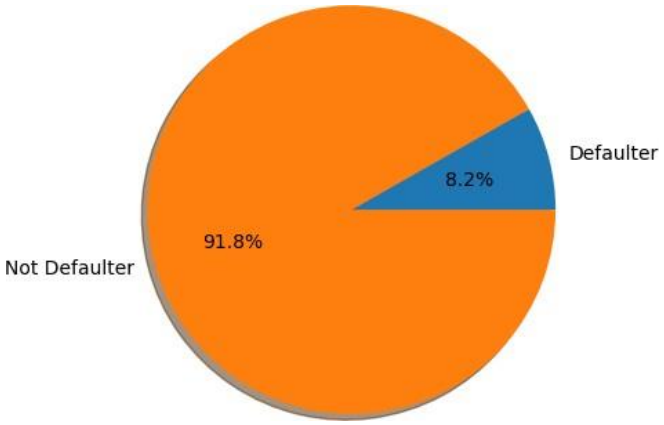
16	credit_score	Credit score
17	prev_defaults	Number of previous defaults
18	default_in_last_6months	Whether default in last 6 months or not? (0/1)
19	credit_card_default	Target variable: Credit card default or not? (0/1)

The target variable of our data frame is “credit_card_default”, which is a binary variable, whose values are 0 and 1. Credit card default risk is the chance that companies or individuals will not be able to return the money lent on time. Data frame has 6 categorical features and 13 numeric features.

3.2. Exploratory Data Analysis (EDA)

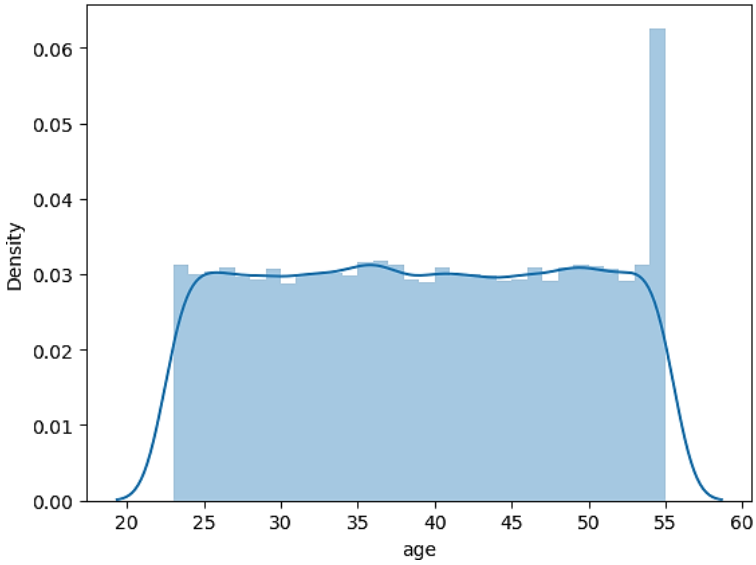
When examining the "credit_card_default" target variable, we observe a disproportion between defaulters and non-defaulters. The number of individuals who have defaulted accounts for only 8.2%, whereas those who have not defaulted make up 91.8%.

Figure 1. Defaulter ratio



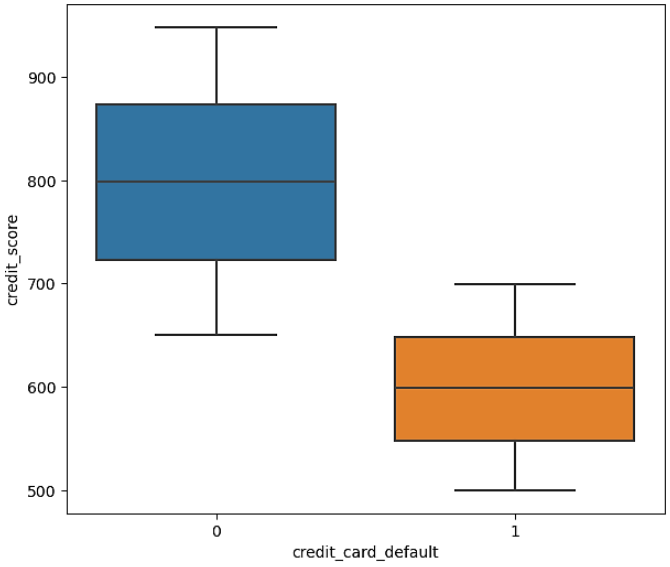
In the dataset, age of customers ranges from 23 to 55. It means that all of them are of working age and likely had at least one job when they borrowed credit.

Figure 2. Distribution of age



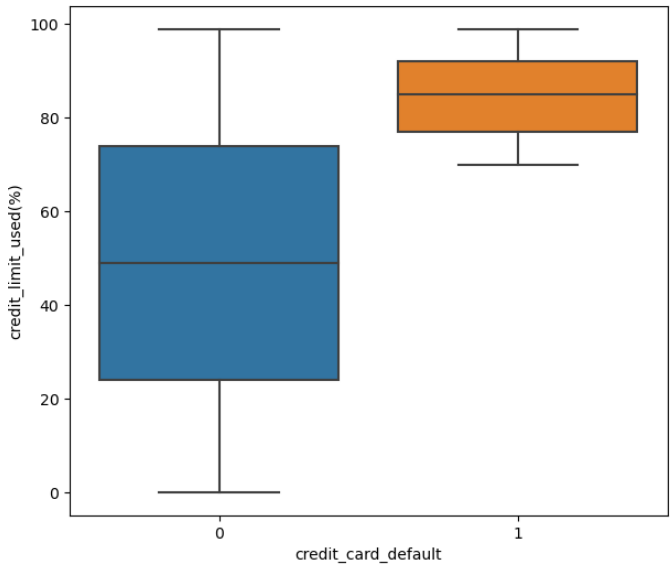
Upon examining the credit score, a noteworthy observation can be made. Individuals with higher credit score have a lower tendency to default on their credit cards, whereas those with lower credit score are more likely to default. This suggests that credit score could be a significant variable in predicting credit card default.

Figure 3. Boxplot of credit score on credit card default



The same conclusion could be drawn from the analysis of credit limit used with respect to credit card default variable. Individuals who utilize a significant portion of their credit limit are more susceptible to default, whereas those who use a lesser amount are less likely to default.

Figure 4. Boxplot of credit limit used on credit card default



Additional details regarding gender, occupation types, and other continuous features can be found in the *appendix.*

3.3. Data Preprocessing

3.3.1. Missing values handling

After EDA, we note that, there are missing values mostly less than 2%, so that, we impute all missing values with statistical way. That is, in most cases, we fill null by mode if it is categorical variable and by median if it is numerical variable.

In terms of categorical features, there are only two variables having missing values: “owns_car” and “gender”. In “owns_car”, there are 369/30000 rows have null values. As stated above, we fill all missing values of this variable by mode. In column “gender”, there is only one observation having null. Since the name of customer in this observation is “Ernard”, we change the missing value into “M” (Male).

About numerical features, there are six variables having missing values. We impute “no_of_days_employed” with median based on “occupation_type”, “yearly_debt_payments” with median based on “credit_card_default” and “credit_card_default” with respect to target variable. In the group of “migrant_worker”, “total_family_members” and “no_of_children”, we fill missing values by mode of its own.

3.3.2. Features selection with IV

The research employed IV estimation, a popular technique in credit scoring problems, to identify the relationship between each characteristic and the outcome variable (Zdravevski et al., 2014). The IV computations for continuous variables were performed after grouping them into bins, and the results are presented in the table below:

Table 3. IV values of variables

Variable	IV
age	0.000002
owns_house	0.000070
yearly_debt_payments	0.000080
credit_limit	0.000470
no_of_children	0.000976
total_family_members	0.002461
net_yearly_income	0.003464
owns_car	0.004524
prev_defaults	0.006260
migrant_worker	0.012561
gender	0.048503
occupation_type	0.096320
no_of_days_employed	0.120459
credit_score	0.422381
default_in_last_6months	0.600895
credit_limit_used(%)	1.148924

With IV threshold equals 0.02, 10 variables would be dropped from the dataset. The list of removals are “age”, “owns_house”, “yearly_debt_payments”, “credit_limit”, “no_of_children”, “total_family_members”, “net_yearly_income”, “owns_car”, “prev_defaults”, “migrant_worker”. Until now, we have 6 variables left to move to the next step.

3.3.3. Variables encoding

Encoding techniques are essential to ensure that all variables are represented accurately in a predictive model. One-hot encoding is a popular technique that creates a new binary variable for each category in the original categorical variable. This technique is widely used and can be implemented using Scikit-learn's OneHotEncoder or Pandas' get_dummies method. One-hot encoding is particularly useful when dealing with nominal categorical variables with no inherent order.

Label encoding, on the other hand, is another technique that is commonly used to convert categorical variables into numerical ones. It assigns a unique integer value to each category in the variable. However, it should be used with caution as it can lead to incorrect interpretations, especially with ordinal categorical variables. For example, if we assign integer values to different income levels, the model may interpret a higher income as having a greater impact than a lower income, which may not be true.

In general, it is crucial to select the appropriate encoding technique based on the type of categorical variable and the specific problem being addressed. In some cases, a combination of encoding techniques may be required to achieve optimal results in the model.

In our cases, we use binary encoding for the “gender” variable and using get_dummies method from Pandas to make dummies variables for “occupation_type”.

3.3.4. Data transformation

Data standardization is a common preprocessing technique used in machine learning to rescale features to the same scale to improve model performance. One such method is scaling, which transforms data to have a mean of zero and standard deviation of one. There are various types of scalers, including StandardScaler, MinMaxScaler, and RobustScaler. StandardScaler is a technique that standardizes features by removing the mean and scaling to unit variance. It assumes that the data is normally distributed, and hence is not suitable for data with skewed distributions. On the other hand, RobustScaler is a technique that scales features using statistics that are robust to outliers. It uses the interquartile range (IQR) to scale features and hence is suitable for data with outliers. It scales the data such that the IQR is 1 and the median is 0.

Another commonly used scaler technique in Sklearn is MinMaxScaler, which scales features to a given range (default is 0-1). It works by subtracting the minimum value of a feature and then dividing by the range of the feature. This technique is useful when the distribution of the data is not necessarily Gaussian or when there are outliers in the data. However, it is important to note that MinMaxScaler is sensitive to outliers and hence, the data needs to be pre-processed before applying this technique.

According to Maruma et al (2022), this technique is commonly used in credit risk analysis to ensure that numerical variables are on the same scale and have equal importance in the machine learning model. So that, we apply MinMaxScaler from Sklearn to our dataset.

3.3.5. Data imbalance handling

Data imbalance is a common problem in machine learning where the number of instances in one class is significantly higher or lower than the number of instances in other classes. This imbalance can have a significant impact on the performance of machine learning algorithms, particularly in classification tasks. In cases where the minority class is of greater interest, the algorithm may fail to identify patterns in the minority class, leading to lower accuracy and recall for that class. On the other hand, when the majority class is of greater interest, the algorithm may become biased towards that class, leading to lower precision and recall for the minority class.

To overcome this challenge, one approach is to use the SMOTE (Synthetic Minority Over-sampling Technique) method, which is a popular technique for dealing with class imbalance. This method involves generating synthetic samples for the minority class by interpolating existing instances. This helps to balance the class distribution and improve the performance of machine learning algorithms. SMOTE works by randomly selecting a sample from the minority class and finding its k-nearest neighbors. It then generates synthetic samples by interpolating between the selected sample and its neighbors. This process is repeated until the desired balance is achieved.

After using SMOTE, the target variable “credit_card_default” are equally distributed between the value 0 and 1.

3.4. Modeling

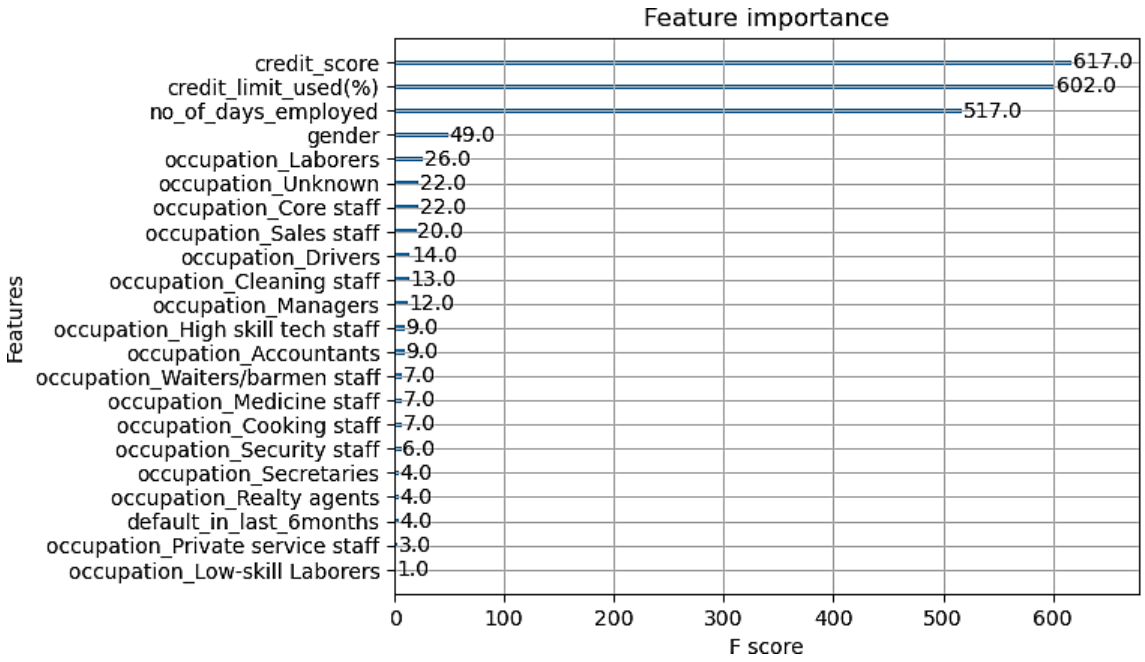
After preprocessing step, we split the dataset into two parts with ratio 70/30: training set and test set. Training set has 21000 rows, test set has 9000 rows. The training set is a large dataset used to train a machine learning model. This is the dataset from which the

model will learn and extract important features to remember. The test set is a dataset used to verify if the results achieved by the model after training are truly effective or not.

As stated above, with this dataset, there are 7 algorithms we use to train model: Logistic Regression, Decision Tree, Random Forest, CatBoost, XGBoost, LightGBM and KNN. In Logistic Regression, we also fit the WOE binning dataset to see the performance.

After the training process, we used feature_importances from the sklearn library to examine and rank the attributes of the customer based on their influence on the predictive performance of the model. Figure X shows that, the credit score, the percentage of credit limit used and number of days employed are among the most important features.

Figure 5. Features importance



4. Results

4.1. Metrics

To measure a model's performance, several metrics such as precision rate, recall rate, F1 value, AUC value, and accuracy can be used. The main evaluation metrics in this study include AUC value, F1 value, precision rate, accuracy, and recall rate. In order to define these metrics, we first establish four variables: (i) TP, which represents the number of positive samples that are correctly predicted by the classification model, (ii) FN, which represents the number of positive samples that are incorrectly predicted as a negative class by the model (also known as Type II error), (iii) FP, which represents the number of negative samples that are incorrectly predicted as a positive class by the model (also known

as Type I error), and (iv) TN, which represents the number of negative samples that are correctly predicted by the model.

Table 4. Confusion Matrix

		Predict	
		0	1
True	0	TN	FP
	1	FN	TP

From the Confusion Matrix, several metrics can be computed to assess the classifier's predictive performance. One of the most widely used metrics is accuracy, which measures the ratio of correctly classified instances to the total number of instances. Accuracy is computed by the following formula:

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

In default risk analysis, identifying default users is of utmost importance to banks. In this regard, recall and precision are two evaluation indicators that hold significant value. Precision is calculated by determining the ratio of true positive samples that are predicted by the classifier. On the other hand, recall is defined as the ratio of positive samples that are correctly predicted by the classifier.:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Accuracy and recall are measures utilized for evaluating a classifier's ability to make accurate predictions on reliable data. However, there can be discrepancies between the two metrics, which can make it difficult to improve both simultaneously. Consequently, to assess a classifier's predictive power, we employ precision and recall calculating the F1 score, which factors in both the classifier's recall and accuracy. The F1-score computation method is shown in the following equation:

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

Another important metric to control overall performance of machine learning model is the AUC, which stands for “Area Under the Curve”. The AUC is a metric commonly

used in evaluating the performance of a binary classifier. It represents the area under the Receiver Operating Characteristic (ROC) curve, which is a plot of the true positive rate against the false positive rate at various threshold settings. A perfect classifier will have an AUC of 1, while a random classifier will have an AUC of 0.5. The AUC provides a single number that summarizes the performance of a classifier across all possible threshold settings, making it a useful metric for comparing different classifiers or models.

4.2. Results

The below table includes information about name of model and the corresponding metrics of each. The information of the logistic regression with the WOE binning dataset is not included in the table. This model achieved the AUC value 0.9809 on the test set. The following figures have been rounded off to easily observe.

Table 5. Models and prediction metrics result

Model	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.9464	0.80	0.96	0.86	0.9585
Decision Tree	0.9663	0.87	0.92	0.90	0.9244
Random Forest	0.9650	0.86	0.94	0.90	0.9378
LightGBM	0.9668	0.87	0.94	0.90	0.9437
KNN	0.9681	0.88	0.93	0.90	0.9259
CatBoost	0.9683	0.88	0.93	0.90	0.9328
XGBoost	0.9734	0.91	0.92	0.91	0.9178

XGBoost model achieves highest accuracy score while the logistic regression has the largest AUC value. In terms of precision and F1-score, XGBoost are still the best model at prediction with this dataset. Therefore, XGBoost is the model with the least incorrect predictions.

5. Conclusions and discussions

In this paper, we conducted a comprehensive analysis of credit risk using various machine learning models, including XGBoost, CatBoost, LightGBM, and Logistic Regression, among others. Our study was conducted on the American Express dataset, which is a widely used dataset in the field of credit risk analysis. We evaluated the performance of each model using several metrics such as accuracy, precision, recall, F1-score, and AUC to identify the best performing model. Our results showed that the XGBoost model outperformed all other models, demonstrating the effectiveness of gradient boosting techniques in credit risk analysis.

The practical implications of our study are significant for financial institutions such as credit card companies, banks, and other lending institutions. By using machine learning models to analyze credit risk, these institutions can better identify potential borrowers who are likely to default, thus reducing their losses. Additionally, by using the variables that were found to be most significant in our study, such as credit score, credit limit utilization, and number of days employed, financial institutions can further refine their credit risk analysis models to improve accuracy and precision.

Future research in this area could focus on comparing the performance of other machine learning models such as Support Vector Machines, Neural Networks, and Deep Learning, among others. This can help in identifying which models perform best in different scenarios and can further improve the accuracy and precision of credit risk analysis. Additionally, future studies can explore the use of alternative data sources such as social media and other online activities to further improve credit risk analysis.

In conclusion, this study demonstrates the effectiveness of machine learning techniques in credit risk analysis. By using the best performing model, XGBoost, financial institutions can improve their credit risk analysis models, which can lead to significant financial savings in the long run. Further research can be conducted in this area to refine the existing models and develop new models that incorporate novel data sources and features, thus improving the accuracy and precision of credit risk analysis.

References

Abellán, J., & Castellano, F.J. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. Expert Syst. Appl., 73, 1-10.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23(4), 589–609.

Altman, E.I., and Sabato, G. (2007). Modelling Credit Risk for SMEs: Evidence from The US Market. ABACUS, 43 (3), 332-357.

Breiman, Leo. 2000. Some Infinity Theory for Predictors Ensembles. Technical Report; Berkeley: UC Berkeley, vol. 577.

Butaru, Florentin, Qingqing Chen, Brian Clark, Sanmay Das, Andrew W. Lo, and Akhtar Siddique. 2016. Risk and risk management in the credit card industry. Journal of Banking and Finance 72: 218–39.

Chen, T.; Guestrin, C., (2016), "XGBoost: A Scalable Tree Boosting System". 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16.

Gahlaut, A., Tushar, & Singh, P.K. (2017). Prediction analysis of risky credit using Data mining classification models. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-7.

Galindo, Jorge, and Pablo Tamayo. 2000. Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications. Computational Economics 15: 107–43.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

Khandani, Amir E., Adlar J. Kim, and Andrew W. Lo. 2010. Consumer credit-risk models via machine-learning algorithms. Journal of Banking and Finance 34: 2767–87

Maruma, C., Tu, C., Nawej, C. (2022). Banking Credit Risk Analysis using Artificial Neural Network. In: Yang, XS., Sherratt, S., Dey, N., Joshi, A. (eds) Proceedings of Seventh International Congress on Information and Communication Technology. Lecture Notes in Networks and Systems, vol 447. Springer, Singapore. https://doi.org/10.1007/978-981-19-1607-6_76

Pandey, T. N., Jagadev, A. K., Mohapatra, S. K., & Dehuri, S. (2017, August). Credit risk analysis using machine learning classifiers. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 1850-1854). IEEE.

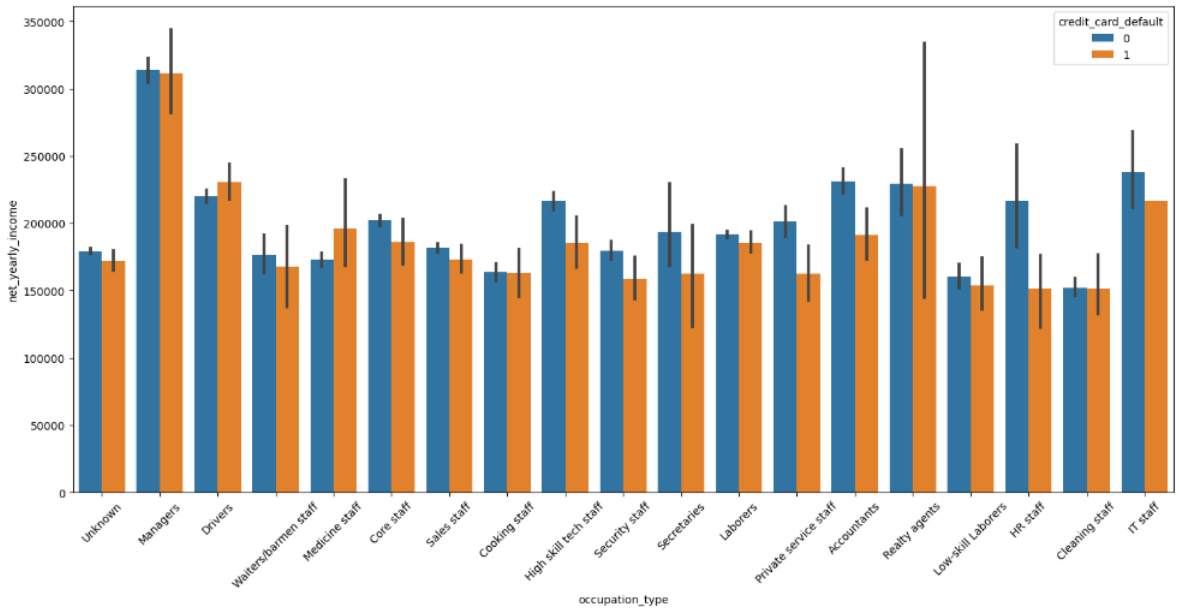
Satchidananda, S. S., & Simha, J. B. (2006). Comparing decision trees with logistic regression for credit risk analysis. International Institute of Information Technology, Bangalore, India.

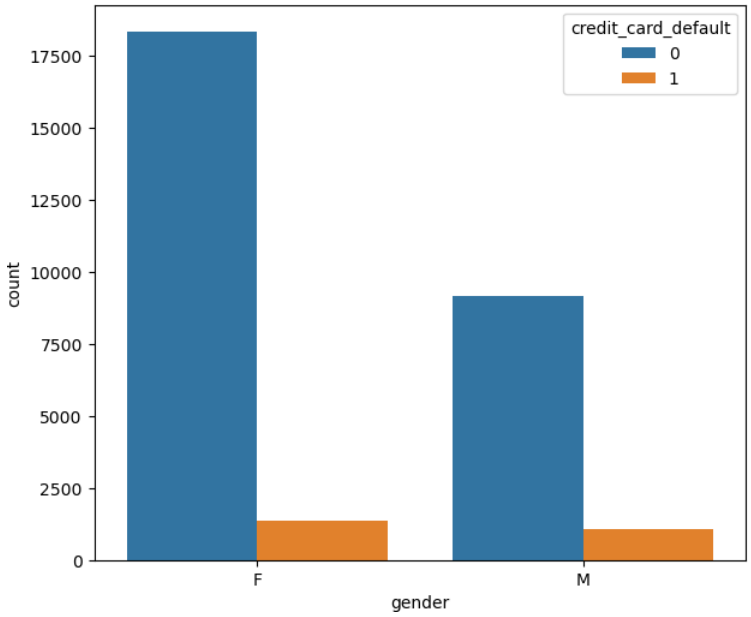
Siddiqi, N. (2006). Credit Risk Scorecards, Developing and Implementing Intelligent Credit Scoring. Hoboken, NJ: John Wiley & Sons, Inc.

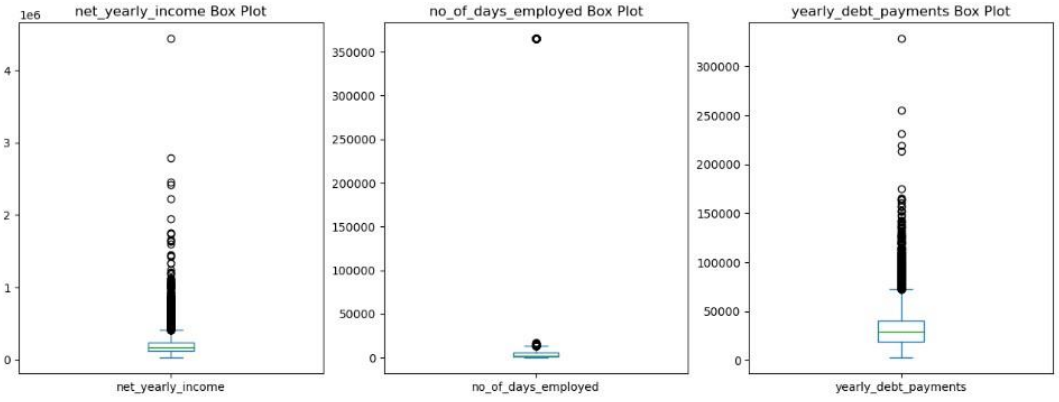
Yazdanfar, D., and Nilsson, M. (2008). The Bankruptcy Determinants of Swedish SMEs. Institute for Small Business and Entrepreneurship, Belfast, Ireland.

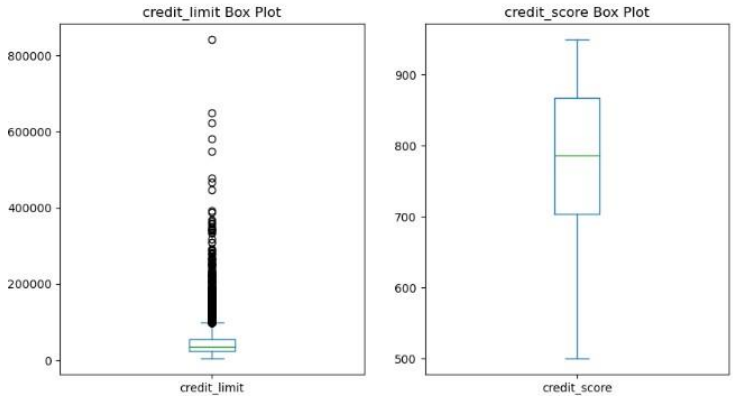
Zdravevski, E., Lameski, P., Kulakov, A., & Gjorgjevikj, D. (2014). Feature selection and allocation to diverse subsets for multi-label learning problems with large datasets. 2014 Federated Conference on Computer Science and Information Systems, 387-394.

Appendix




credit_card_default
0
1
count
gender
F
M


net_yearly_income Box Plot
1e6
net_yearly_income
no_of_days_employed Box Plot
no_of_days_employed
yearly_debt_payments Box Plot
yearly_debt_payments


credit_limit Box Plot
credit_limit
credit_score Box Plot
credit_score