**Predicting Credit Risk in Financial Lending: A Machine Learning Approach**

**Authors:** Nhan Yến Trang - 11219290

# **ABSTRACT**

Predicting credit risk is a vital task for financial institutions. Advances in machine learning technology, alongside the availability of extensive datasets and computational power, have led to the development of more sophisticated credit risk prediction models within the finance industry. In this project, we employed various Supervised Machine Learning Models, including Logistic Regression, Random Forest Classifier, CatBoost Classifier, and LightGBM Classifier, on a synthesized dataset modeled on real data to preserve customer privacy. Among the tested models, the CatBoost Classifier emerged as the most suitable for our dataset, demonstrating exceptional performance with an accuracy of 0.93, precision of 0.96, F1-score of 0.82, specificity of 0.99, and an AUC of 0.95. This indicates the CatBoost Classifier's superior ability to balance precision and recall, effectively managing the trade-offs inherent in credit risk prediction tasks.

Keywords: Supervised Machine Learning, Credit risk.

# **INTRODUCTION**

In the realm of financial services, assessing credit risk accurately is paramount for institutions to maintain solvency and profitability. The advent of machine learning (ML) technologies has revolutionized many aspects of the financial industry, including credit risk assessment. These technologies offer the potential to predict credit risk with greater accuracy than traditional statistical methods by processing vast amounts of data and identifying complex patterns that may not be immediately apparent to human analysts. Recent technological advancements, such as increased computing power and the availability of large datasets, have further facilitated the development and deployment of advanced Machine Learning models in this sector.

Pande and colleagues undertook a study on credit risk using machine learning classifiers, including Artificial Neural Networks (ANN), k-Nearest Neighbors (KNN), and Naive Bayes (NB), leveraging the German credit dataset. The primary metric for evaluating the ML models' performance was accuracy. Their findings revealed accuracies of 77.45% for ANN, 77.20% for NB, and 72.20% for KNN. Despite these promising results, the study did not extend its evaluation to include other critical metrics like F1-Score or the Area Under the Curve (AUC) score.

In a separate study, Nasser and Maryam devised a credit risk assessment system for customers using Artificial Neural Networks (ANNs), focusing on the Gradient Descent learning method. The main metric for evaluating the effectiveness of their system was accuracy, applied to credit risk datasets from Australia, Japan, and Germany. Their system demonstrated accuracies of 78.11%, 76.87%, and 68.26% across these datasets, respectively.

Furthermore, Ha and colleagues developed a refined model for predicting credit risk in online peer-to-peer (P2P) lending, incorporating a feature selection (FS) method and deep learning (DL). Their approach started with data preprocessing, followed by feature selection with Restricted Boltzmann Machines (RBMs), and then model implementation using techniques such as Linear Discriminant Analysis (LDA), Artificial Neural Networks (ANN), k-Nearest Neighbors (KNN), and Random Forest (RF). When tested on the Australian and German credit datasets, the models achieved accuracies of 85.80%, 71.45%, 65.94%, and 67.72% for LDA, and 76.50%, 75.8%, 67.10%, and 67.72% for RF, respectively, on the German dataset. Despite these advancements, the research did not explore other important performance metrics like precision, recall, and AUC.

The aforementioned studies illustrate the growing reliance on machine learning models to predict credit risk, highlighting the field's dynamic evolution. Despite these advances, there remains a noticeable gap in the comprehensive evaluation of these models beyond accuracy. Metrics such as precision, recall, F1-Score, and AUC offer valuable insights into model performance, emphasizing the importance of a multi-faceted assessment approach.

This study aims to bridge this gap by deploying several Supervised Machine Learning Models on a synthesized dataset modeled after real data. My objective is not only to assess these models based on accuracy but to extend my evaluation to include precision, F1-Score, specificity, and AUC. This comprehensive approach will allow me to determine the most effective model for credit risk prediction, contributing to the optimization of credit risk management strategies within the financial sector.

The significance of this study lies in its potential to enhance the precision of credit risk assessments, enabling financial institutions to make more informed lending decisions. By adopting a broader set of performance metrics, this research endeavors to provide a nuanced understanding of model efficacy, paving the way for more reliable and robust credit risk prediction methodologies.

# **THEORETICAL BACKGROUND**

## Methodology

### 1.1. Weight of Evidence (WOE)

The weight of evidence measures the predictive power of an independent variable in relation to the dependent variable. It has its roots in the credit scoring world and it tells the degree of the separation of good and bad customers. “Good Customers” refers to the customers who pay back loans (non-events) and “Bad Customers” refers to the customers who fall behind with paying a loan (events).

The formula for calculating WOE is given by:

Where,

Distribution of Goods: % of Good Customers in a particular group

Distribution of Bads : % of Bad Customers in a particular group

A positive Weight of Evidence (WOE) value indicates that the Distribution of Goods (non-default cases) exceeds the Distribution of Bads (default cases) within a particular category. This suggests that the category is more likely associated with good outcomes. Conversely, a negative WOE value signifies that the Distribution of Goods is less than the Distribution of Bads, implying that the category is more commonly linked with bad outcomes.

The Weight of Evidence (WOE) is indispensable in credit risk modeling, primarily serving to transform categorical variables into numerical values. This transformation is crucial for integrating qualitative data into logistic regression and other machine learning algorithms, ensuring that the rich information encapsulated within categorical variables is not lost but instead utilized to predict outcomes more accurately. WOE's ability to maintain the contextual integrity of data while facilitating its analytical use underscores its vital role in enhancing model development and analysis in the financial sector.

The advantages of WOE are manifold, notably its capacity to reduce data variance, effectively handle missing data, and streamline the variable selection process. By quantifying the predictive strength of variables and offering a clear framework for data transformation, WOE not only improves model interpretability but also boosts the overall accuracy and reliability of predictions. Its straightforward application and the interpretability of its outcomes make WOE an invaluable tool, optimizing predictive models and ensuring more informed decision-making in credit risk assessment.

### 1.2. Information Value

Information Value gives a measure of how variable X is good in distinguishing between a binary response (e.g. “good” versus “bad”) in some target variable Y. Low Information Value of a variable X means that it may not classify the target variable on a sufficient level and should be removed as an explanatory variable.

The formula for calculating IV is given by:

Rules related to Information Value

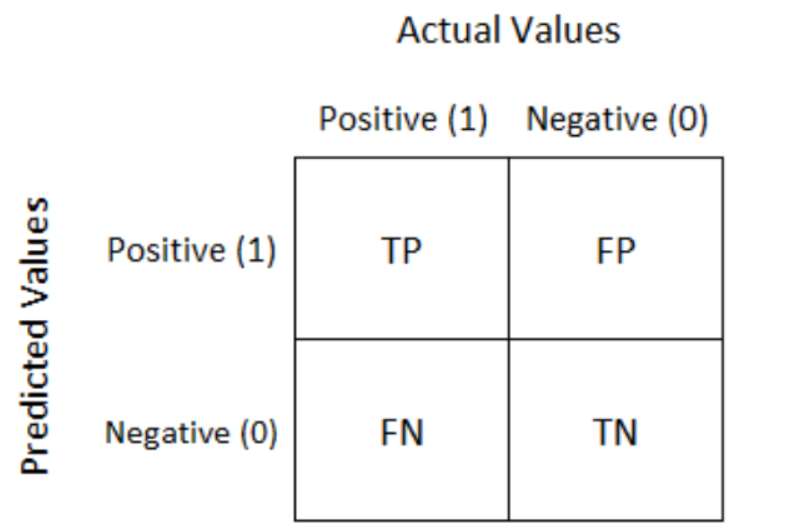
***Table 1. Conventional Interpretation of IV***

| **Information Value** | **Variable Predictiveness** |
| --- | --- |
| Less than 0.02 | Not useful for prediction |
| 0.02 to 0.1 | Weak predictive Power |
| 0.1 to 0.3 | Medium predictive Power |
| 0.3 to 0.5 | Strong predictive Power |
| > 0.5 | Suspicious Predictive Power |

### 1.3. Confusion matrix

The confusion matrix serves as a fundamental tool for evaluating the performance of machine learning classification models, applicable when the output involves two or more categories. It is essentially a table that displays the intersections of the predicted and actual classes.

***Figure 1. Confusion Matrix illustration***



Defined as a matrix, it facilitates the performance assessment of a classification algorithm on a test dataset whose true values are known. It proves invaluable for quantifying Recall, Precision, Accuracy, and the AUC-ROC curve. To comprehend these metrics, it's essential to introduce four fundamental variables:

* True Positive (TP): The count of positive instances accurately identified by the model.
* True Negative (TN): The count of negative instances accurately identified by the model.
* False Positive (FP): The count of negative instances incorrectly identified as positive by the model, also known as a Type I error.
* False Negative (FN): The count of positive instances incorrectly identified as negative by the model, also referred to as a Type II error.

Accuracy represents the ratio of correctly predicted observations to the total observations:

Precision, or the positive predictive value, calculates the ratio of correctly predicted positive observations to the total predicted positive observations:

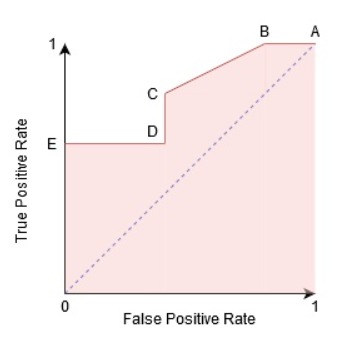
Recall measures the ratio of correctly predicted positive observations to all observations in the actual class:

The F1 Score represents the harmonic mean of Precision and Recall, balancing both metrics:

Specificity measures the proportion of actual negative instances accurately identified by the model, reflecting the fraction of true negative outcomes from all genuine negative cases.

The ROC Curve plots the true positive rate against the false positive rate at various threshold levels, with the AUC representing the area under the ROC Curve. This area measures the model's classification quality; the larger the area, the better the model's performance. An AUC of 1 implies perfect distinction between positive and negative classes, 0 implies complete misclassification, and 0.5 indicates an inability to distinguish between classes.

***Figure 2. The ROC curve***



## Machine learning models

### 2.1. Logistic Regression Classifier

Logistic Regression is a widely used machine learning classification model designed to predict the outcome of a binary dependent variable based on one or more independent variables. This model operates by employing a logistic function, a sigmoid function, to estimate the probability that a particular instance falls into a specific class. The strengths of Logistic Regression include its ability to handle non-linear relationships between independent and dependent variables and its capability to provide explicit probabilities for classifications, which is beneficial for making decisions based on probability thresholds. Logistic Regression is commonly applied in fields such as healthcare, finance, and many others where classification and risk estimation are crucial.

The sigmoid function is referred to as an activation function for logistic regression and is defined as:

### 2.2. Random Forest Classifier

The Random Forest Classifier is an ensemble machine learning algorithm distinguished for its proficiency in both classification and regression tasks. By constructing multiple decision trees during the training phase and aggregating their predictions, Random Forest effectively handles large datasets with complex structures, ensuring high accuracy in diverse applications. Its mechanism of averaging predictions across trees helps mitigate the risk of overfitting, a notable advantage that enhances the model’s generalization capabilities.

A key strength of Random Forest lies in its ability to provide insights into feature importance, allowing for a deeper understanding of the data. This attribute, coupled with its robustness against noise and high dimensional data, makes it invaluable across various sectors, including finance for fraud detection and healthcare for disease diagnosis. The versatility and interpretability of Random Forest solidify its status as a fundamental tool in machine learning, aiding in both predictive modeling and data exploration.

### 2.3. Gradient Boosting Decision Tree

The Gradient Boosting Decision Tree (GBDT) is a potent machine learning technique that builds on the concept of boosting multiple weak models (typically decision trees) into a strong predictive model. This method incrementally adds decision trees to the ensemble, where each new tree aims to correct the prediction errors made by the preceding ones. Utilizing gradient optimization techniques to identify and minimize errors, GBDT efficiently tackles complex problems across large and varied datasets.

Key strengths of GBDT include its capability to manage high-dimensional data, its flexibility in addressing both classification and regression problems, and its adeptness at automatically handling both categorical and continuous variables without the need for explicit data transformation. Additionally, it offers highly interpretable model outcomes and often achieves superior performance across diverse data types. However, careful parameter tuning is required to avoid overfitting, and training on very large datasets can be time-consuming.

### 2.4. LightGBM Classifier

LightGBM, standing for Light Gradient Boosting Machine, is a sophisticated iteration of the gradient boosting framework designed for speed and effectiveness in classification and regression tasks. It excels in processing large datasets and managing high-dimensional data with remarkable efficiency. This efficiency is primarily due to innovative techniques such as gradient-based one-side sampling and exclusive feature bundling, which streamline the node-splitting process, reducing the computational load without compromising accuracy.

One of the key features of LightGBM is its direct handling of categorical variables, simplifying the data preparation phase. Moreover, it boasts support for parallel and GPU learning, significantly cutting down training durations while ensuring robust model performance. Importantly, LightGBM balances speed with interpretability, offering valuable insights into which features most influence the model's decisions.

However, LightGBM's robust capabilities necessitate meticulous parameter optimization to prevent overfitting, particularly with smaller datasets. Its ability to quickly generate and refine models positions it as a favored tool among data scientists for various applications, from detecting fraudulent activities to understanding customer behaviors.

### 2.5. CatBoost Classifier

CatBoost (Categorical Boosting) is an open-source gradient boosting library developed by Yandex that specializes in handling categorical data directly. As a machine learning algorithm, it is designed for classification and regression tasks and stands out for its exceptional handling of categorical variables without the need for extensive data preprocessing. CatBoost achieves high accuracy through sophisticated algorithms that reduce overfitting, making it highly effective even with datasets that have a lot of noise.

One of the distinctive features of CatBoost is its innovative approach to processing categorical data, utilizing an ordered boosting method and one-hot encoding with a minimal number of distinct values. This allows CatBoost to efficiently manage high cardinality features and deliver robust model performance. Additionally, it offers built-in support for GPU acceleration, which speeds up the training process significantly.

CatBoost is renowned for its ease of use, providing automatic parameter tuning and detailed feature importance analysis, which enhances model interpretability. Its versatility and user-friendly nature make it a popular choice among data scientists and researchers for a wide array of applications, ranging from recommendation systems to predictive analytics. Despite its power, users are encouraged to fine-tune model parameters carefully to achieve the best possible performance tailored to their specific data characteristics.

# **DATA AND DESCRIPTIVE ANALYSIS**

## Data Description

The dataset employed in this project is a synthetic one, specifically designed for the purpose of credit scoring. The use of such a dataset stems from the confidential nature of customer credit data, which is typically safeguarded by financial institutions. This particular dataset was sourced from [kaggle](https://www.kaggle.com/datasets/laotse/credit-risk-dataset/data).

Comprising 32,581 entries and 12 attributes, the dataset includes a mix of 8 numerical and 4 categorical features. Central to the dataset is the "target" variable, a binary indicator with possible values of 0 or 1, distinguishing between the two classes for the prediction task.

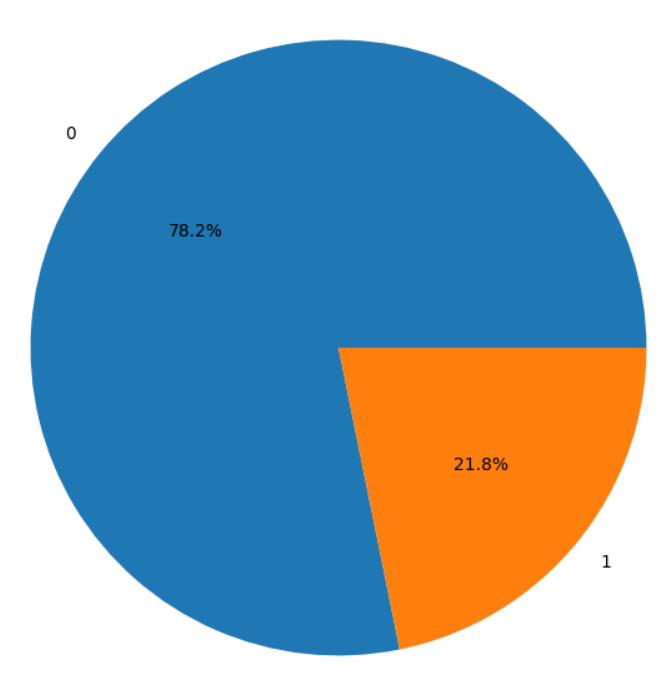
***Table 2. Columns Description***

| **No** | **Variable** | **Meaning** |
| --- | --- | --- |
| 1 | age | The person’s age in years |
| 2 | annual\_income | The person’s annual income |
| 3 | home\_ownership | The type of home ownership (RENT, OWN, MORTGAGE, OTHER) |
| 4 | empl\_length | The person’s employment length in years |
| 5 | loan\_intent | The person’s intent for the loan (PERSONAL, EDUCATION, MEDICAL, VENTURE, HOMEIMPROVEMENT, DEBTCONSOLIDATION) |
| 6 | loan\_grade | The loan grade (A, B, C, D, E, F, G) |
| 7 | loan\_amnt | The loan amount |
| 8 | loan\_int\_rate | The loan interest rate |
| 9 | loan\_percent\_income | The percentage of person’s income dedicated for the mortgage |
| 10 | default\_on\_file | If the person has a default history (Yes, No) |
| 11 | credit\_history\_length | The person’s credit history |
| 12 | target | Shows whether the loan is currently in default with 1 being default and 0 being non-default |

## Exploratory Data Analysis

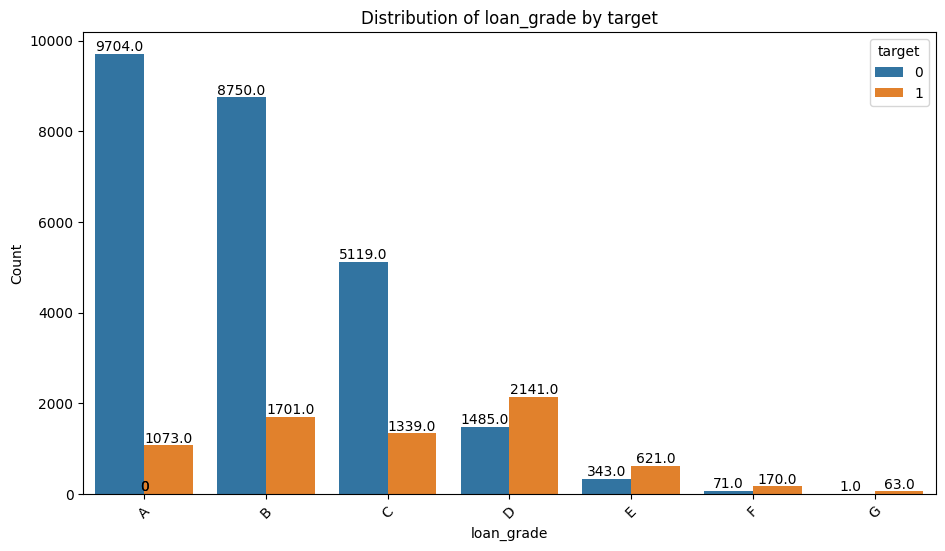
Figure 1 depicts a marked imbalance in the distribution of the 'target' variable, illustrating that 78.2% of loans are currently not in default (class '0'), while only 21.8% are in default (class '1'). This distribution indicates that the dataset is skewed towards non-defaulting loans.

***Figure 3. Distribution of Target Variable***



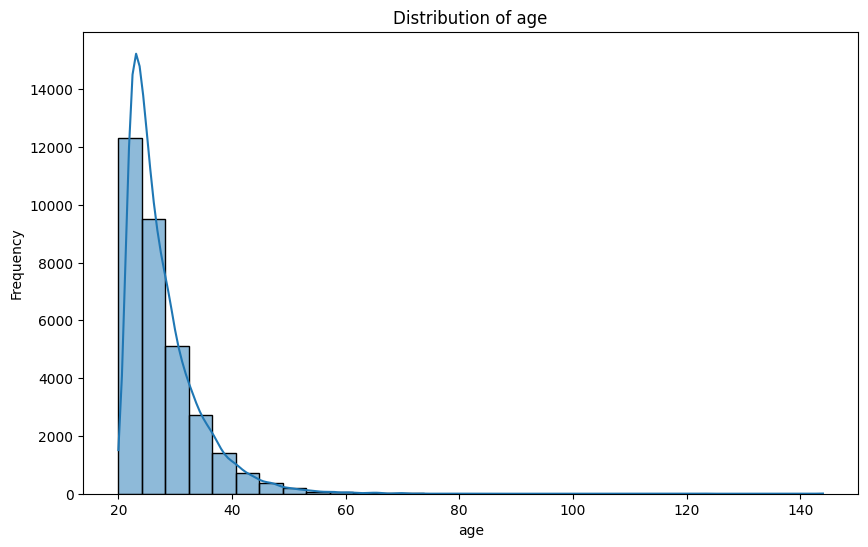
The bar chart demonstrates a clear trend across loan grades: higher grades like 'A' and 'B' have more non-defaults, while lower grades 'E' to 'G' see a significant rise in defaults. This suggests that lower loan grades, indicative of riskier credit profiles, are more likely to default.

***Figure 4. Distribution of Loan grade by Target***



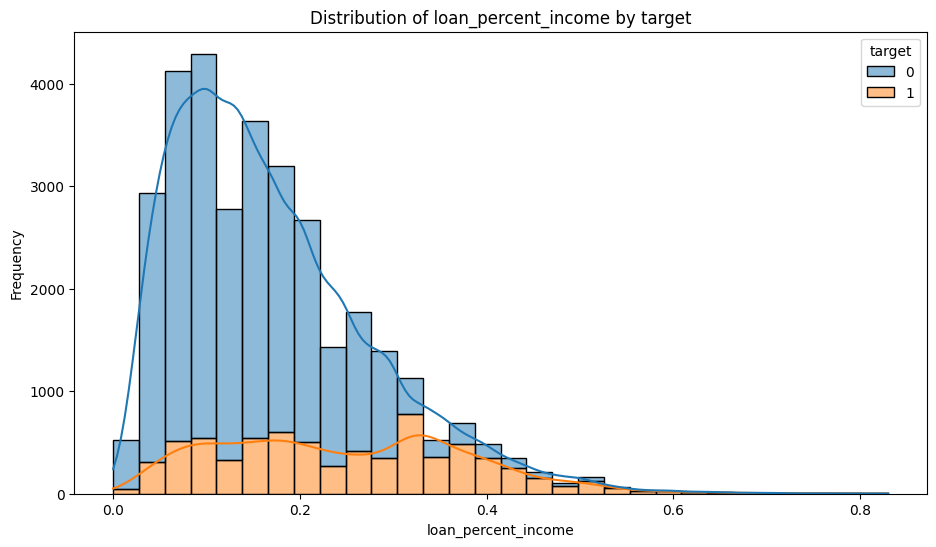
The distribution of the Age is right-skewed. This means that most of the values lie on the right side of the scale. In other words, considering the age, the dataset is populated mostly by young people.

***Figure 5. Distribution of Age***



The data skews towards the lower end, suggesting that most loans constitute a smaller percentage of borrowers' income. Notably, defaults (represented by the orange color) have a higher frequency at increased loan percentages of income, indicating a trend where a higher loan relative to income correlates with a greater likelihood of default. This trend may imply that loans consuming a larger share of income are riskier.

***Figure 6. Distribution of Loan percent income by Target***

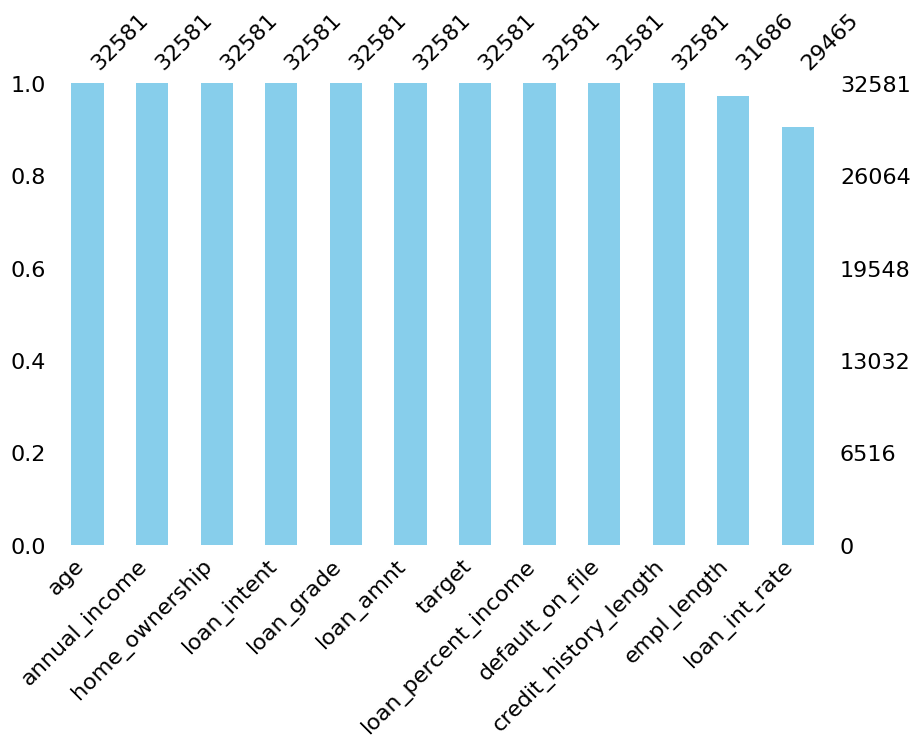


## Data Preprocessing

### 3.1. Missing values handling

During the exploratory data analysis (EDA), I employed the Missingno library to visualize the amount of missing data in each column. Notably, there were two numerical features with missing values: ‘empl\_length’ was missing 895 values, and ‘loan\_int\_rate’ was missing 3,116 values.

***Figure 7. Missing values chart***



To address this issue, I opted to use the median value to fill in the gaps. This approach is designed to maintain the central tendency of the data without being skewed by extreme values. Using the median is particularly effective for numerical data because it is less sensitive to outliers than the mean, which ensures that the overall distribution of the data remains intact. This method is a common practice in data preprocessing as it prepares the dataset for more robust and accurate analysis, especially when the missing data is not biased towards high or low extremes.

### 3.2. Features engineering

Feature engineering is a critical step in the data science pipeline, involving the creation of new variables or modification of existing ones to uncover additional insights or enhance the predictive power of a model.

In the context of loan data analysis, three newly engineered features provide nuanced perspectives on a borrower's financial status and risk profile.

The first feature, the Loan-to-Income Ratio, gauges the scale of the loan request against the borrower's earnings. A high ratio here might flag a potential strain on the borrower's financial resources, suggesting they are seeking a significant sum relative to their income, which could pose a higher risk of default.

The second feature, the Loan-to-Employment Length Ratio, contrasts the duration of employment with the loan amount. It's a measure of employment stability in the face of financial obligation. A lower ratio might be a sign of concern, indicating a relatively large loan amount compared to the length of employment, perhaps suggesting less job stability or a newer entrant to the workforce.

Lastly, the Interest Rate-to-Loan Amount Ratio Links the cost of the loan to the principal amount. This ratio can signal the lender's perception of risk, as higher interest rates often correlate with riskier loans. A smaller loan with a relatively high-interest rate might be seen as less favorable to the borrower, reflecting a higher risk perceived by the lender.

By integrating these engineered features into the analysis, one can derive more sophisticated insights and make more informed decisions, whether in the realm of credit scoring, risk assessment, or tailored financial product offerings.

### 3.3. Features selection with IV

The study utilized Instrumental Variable (IV) estimation, a commonly used technique in credit scoring research, to examine the relationship between various characteristics and the outcome variable (Zdravevski et al., 2014). Continuous variables underwent IV computations, which were conducted by categorizing them into bins. The findings of these computations are summarized in the table provided below:

***Table 3. IV values of variables***

| **Variable** | **IV** | **Classification** |
| --- | --- | --- |
| credit\_history\_length | 0.004989 | Not useful for prediction |
| age | 0.011422 | Not useful for prediction |
| int\_rate\_to\_loan\_amt\_ratio | 0.027559 | Weak predictor |
| empl\_length | 0.066925 | Weak predictor |
| loan\_intent | 0.093669 | Weak predictor |
| loan\_amnt | 0.094354 | Weak predictor |
| loan\_to\_emp\_length\_ratio | 0.122540 | Medium predictor |
| default\_on\_file | 0.174384 | Medium predictor |
| home\_ownership | 0.385031 | Strong predictor |
| annual\_income | 0.458351 | Strong predictor |
| loan\_int\_rate | 0.693564 | Suspiciously strong |
| loan\_percent\_income | 0.881044 | Suspiciously strong |
| loan\_to\_income\_ratio | 0.924024 | Suspiciously strong |
| loan\_grade | 0.928253 | Suspiciously strong |

Variables such as loan\_grade, loan\_int\_rate, loan\_to\_income\_ratio, and loan\_percent\_income show high Information Value (IV), indicating strong predictive potential for the target variable in credit scoring. Conversely, age and credit\_history\_length demonstrate very low IV, suggesting minimal predictive capability for the target.

### 3.4. Variables encoding

When it comes to preparing categorical data for the rigor of machine learning models, several encoding techniques stand ready to translate categories into a numerical language that algorithms can digest. Among these techniques, we have label encoding, which assigns each category a unique integer. While neat and compact, label encoding assumes an order that may not exist, which can confuse models into thinking one category ranks higher than another. Then there's one-hot encoding, which creates a binary column for each category, eliminating any notion of hierarchy and allowing the model to appreciate each category's unique contribution to the pattern it tries to learn.

In the realm of one-hot encoding, `get\_dummies` from the Pandas library is a star player. This method takes each category in a column and creates a new column for it, assigning a 1 or 0 to indicate the presence or absence of that category in the original data. It's like taking the essence of a category and giving it its own flag to wave, without muddling the message with numeric size or importance.

In our case, `get\_dummies` was called upon to shed light on the diverse landscape of our loan dataset. It marched through the columns, `loan\_grade`, `home\_ownership`, `loan\_intent`, and `default\_on\_file`, and for each unique category, it unfurled a new binary column. With this technique, we sidestepped the risk of misinterpreted rankings while enriching our model's language to better understand the nuanced dance of factors that sway the world of lending.

### 3.5. Data transformation

In the realm of data preprocessing, transforming numerical data is crucial for harmonizing scales and bringing different variables to a common ground. Many machine learning algorithms perform better or converge faster when features reside on a similar scale. Without this step, features with larger magnitudes can unduly influence the model, overshadowing the smaller scale variables.

For this purpose, the StandardScaler from Scikit-learn is a go-to technique. It works by subtracting the mean and dividing by the standard deviation for each feature. Each transformed feature now has a mean of 0 and a standard deviation of 1. This scaling doesn't distort differences in the range of values or lose information from the original data, which is why it's such a staple in the data science process.

I applied the StandardScaler to my dataset, particularly to features like `age`, `annual\_income`, `empl\_length`, and others encapsulated in the `num\_cols` list. Each of these features, which vary in units and range, was carefully standardized, ensuring that no single feature would dominate the predictive modeling due to its scale. The transformed data now presents a level playing field, allowing the nuances and strengths of each feature to contribute more equally to the predictive insights I aim to uncover.

### 3.6. Data imbalance handling

Dealing with data imbalance is a common conundrum in machine learning, especially when it comes to classification problems. It's like trying to learn the rules of a game by watching it played over and over, but one team hardly ever gets the ball—it's tough to understand their strategies or predict their plays. This imbalance often leads models to be biased towards the majority class, while the rare but important minority class gets overlooked.

To even out the playing field, techniques like SMOTE, or Synthetic Minority Over-sampling Technique, come into play. It's a bit like giving the underrepresented team some extra practice sessions. Specifically, SMOTE generates synthetic examples of the minority class by creating new, similar-but-slightly-different instances. It does this by finding the nearest neighbors within the minority class and interpolating new points between them.

I employed SMOTE on my training data to help mitigate the imbalance issue. By setting `k\_neighbors` to 5, I instructed SMOTE to consider five nearby instances of the minority class to craft the new synthetic data points. The `random\_state` set to 42 ensures that every time the code runs, it generates the same synthetic points, which is great for reproducibility of the results.

By resampling the dataset in this way, I was essentially giving the model a more balanced perspective. This doesn't just help the algorithm learn better; it can also improve performance metrics, making the resulting predictions more reliable and fair, particularly for the minority class.

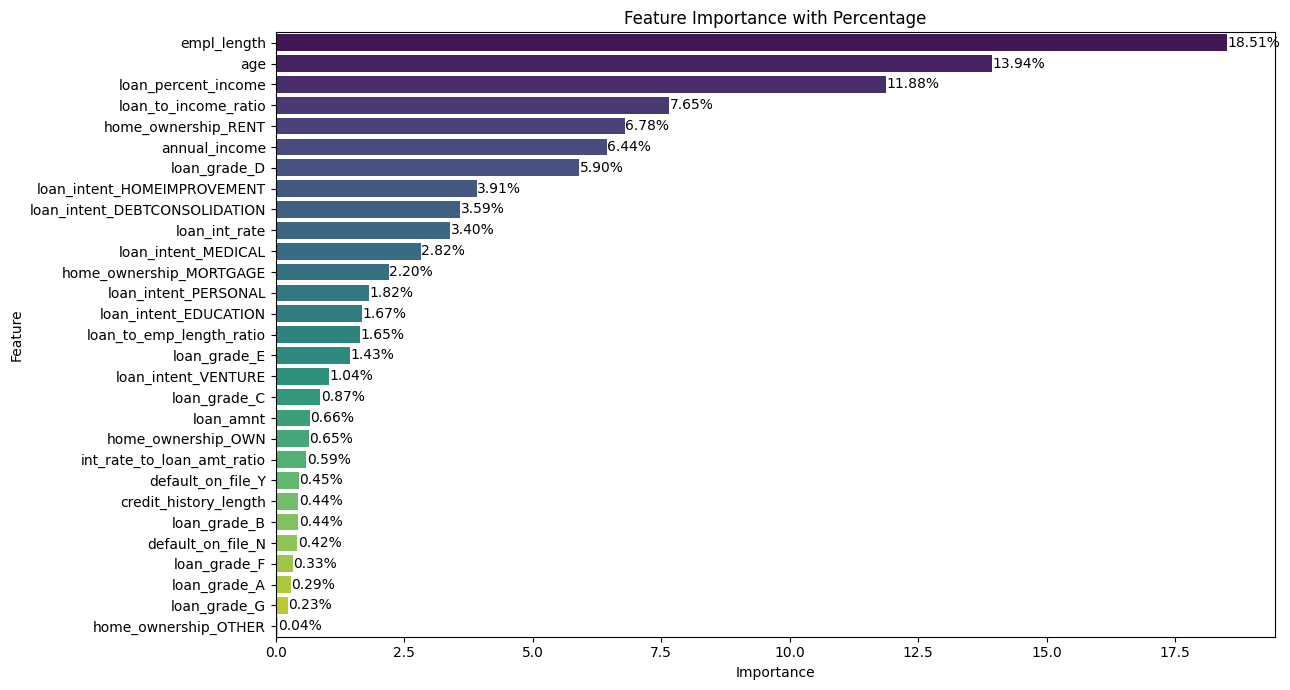
## Features importance

After preprocessing the data, I split the dataset into an 80% training set and a 20% testing set. In machine learning, the training set is employed to train the model on how to make predictions. Conversely, the test set is utilized to assess the model's performance.

Once the datasets were divided, I proceeded to train the model using 12 different algorithms. Moreover, with Logistic Regression, I also fitted the WOE (Weight of Evidence) binned dataset to observe the performance.

Following the training process, the CatBoost Classifier emerged as the top-performing model. I employed the feature\_importances\_ from the sklearn library to inspect and rank the influence of the variables on the model's performance.

Figure 6 reveals that empl\_length, age, and loan\_percent\_income are among the most significant features.

***Figure 6. Features importance ***

# **RESULTS**

The table below includes information on the top 4 models with the highest results and the corresponding parameters for each model. The information regarding the logistic regression with the WOE binning dataset is not included in the table since its AUC score is only 0.88.

***Figure 7. Top 4 models with the highest results***

| Model | Accuracy | Precision | Recall | Specificity | F1 Score | AUC |
| --- | --- | --- | --- | --- | --- | --- |
| CatBoost Classifier | 0.933239 | 0.969903 | 0.721300 | 0.993739 | 0.827329 | 0.945124 |
| LightGBM Classifier | 0.932449 | 0.969337 | 0.707581 | 0.993739 | 0.818030 | 0.941658 |
| XGBoost Classifier | 0.931187 | 0.945591 | 0.727798 | 0.988285 | 0.822521 | 0.944599 |
| Random Forest Classifier | 0.923769 | 0.965381 | 0.704693 | 0.992931 | 0.814691 | 0.927592 |

The CatBoost model has the highest accuracy. Additionally, LightGBM also has an accuracy nearly equal to that of CatBoost. However, in terms of precision, F1-score, and AUC, CatBoost is the model with the best predictive capability.

# **CONCLUSIONS AND DISCUSSIONS**

This study has demonstrated the robust capabilities of various supervised machine learning models in predicting credit risk, leveraging a synthesized dataset modeled on real-world data. Among the various models tested, the CatBoost Classifier distinguished itself as the most effective, showcasing the highest accuracy, precision, F1-score, specificity, and AUC scores. These results underscore the model's ability to manage the inherent trade-offs in credit risk prediction, such as balancing sensitivity (recall) and specificity effectively.

The investigation into different machine learning strategies, including Logistic Regression, Random Forest, LightGBM, and XGBoost, highlighted that no single model fits all scenarios. Each model brings unique strengths to different aspects of the predictive process, and the choice of model can significantly influence the outcome of credit risk assessments.

The use of advanced feature engineering and selection techniques like WOE and IV analysis further enriched the predictive models, allowing for more nuanced insights into the factors that influence credit risk. This approach not only enhanced model accuracy but also offered a deeper understanding of the underlying patterns in the data.

The findings of this study contribute to the broader discourse on machine learning applications in financial services, particularly in the context of credit risk assessment. The superior performance of the CatBoost Classifier, especially in handling categorical data and complex interactions within the dataset, suggests that this model could be an invaluable tool for financial institutions aiming to optimize their credit risk management strategies.

However, the study also acknowledges the limitations associated with using a synthetic dataset. While this ensures privacy and facilitates extensive testing without ethical concerns, the translation of these findings to real-world scenarios may require additional validation to account for the nuances and variability of real customer data.

In conclusion, this research has not only highlighted the effectiveness of comprehensive machine learning approaches to credit risk prediction but also paved the way for future investigations that might integrate more granular data analysis techniques or newer models that could offer even better predictive performance.

# **REFERENCES**

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

[2] Mohammadi, N., & Zangeneh, M. (2016). Customer credit risk assessment using artificial neural networks. International Journal of Information Technology and Computer Science, 8(3), 58–66.

[3] Ha, V. S., Lu, D. N., Choi, G. S., Nguyen, H. N., & Yoon, B. (2019). Improving credit risk prediction in online peer-to-peer (P2P) lending using feature selection with deep learning. In 21st International Conference on Advanced Communication Technology (pp. 511–515).

[4] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics.

[5] Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. Journal of Banking & Finance, 34(11), 2767-2787.

[6] Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 247(1), 124-136.

[7] Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems with Applications, 39(3), 3446-3453.

[8] Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. Management Science, 49(3), 312-329.

The code of this paper can be found [link](https://github.com/IamYtrang/Credit-Risk-Analysis)