PREDICT CREDIT CARD APPROVAL USING MACHINE LEARNIG APPROACH

1. Introduction

Predicting a good client is vital for a bank in today's world. It helps them manage risk, operate more efficiently, enhance customer satisfaction, reduce bad debt, gain a competitive advantage, meet regulatory requirements, and potentially expand financial inclusion. By using data-driven approaches, banks can make better-informed decisions about whom to lend to and under what terms, ultimately benefiting both the institution and its clients.

Predicting good clients is essential for a bank as it supports prudent risk management, reduces bad debt, optimizes resource allocation, fosters a competitive edge, ensures compliance with regulations, enhances customer satisfaction, promotes financial inclusion, and facilitates data-driven decision-making. This, in turn, contributes to the bank's financial stability and long-term success.

The integration of predictive analytics in the banking sector holds significant implications. Mainly, it strengthens risk management by minimizing loan defaults, so ensuring greater financial stability and public trust. Additionally, predictive analytics optimizes operational efficiency, making the lending process more efficient. However, challenges exist, including data quality assurance, regulatory compliance, skill development, and technological upgrades. Addressing these gaps is crucial for Indian banks to harness the potential of predictive analytics effectively. Once overcome, this approach allows banks to make informed lending decisions, reduce risks, enhance operational efficiency, and expand financial services, aligning with financial inclusion goals and bolstering competitiveness in the dynamic Indian market.

1.1 Hypothesis

" Identifying the model that achieves the highest accuracy in predicting credit card approval through the exploration of various models and techniques."

2. Literature review

The banking industry has undergone significant transformation with the advent of information technology, particularly in the realm of credit card and online banking transactions. However, this progress has also exposed the industry to heightened vulnerabilities, as cyberattacks on banks have surged, putting vast amounts of customer data at risk. Cybersecurity has become a paramount concern for banks, with the global cyber security market expected to reach $170.4 billion in 2022. The cost of cybercrime is projected to increase by 15% annually over the next three years, potentially exceeding $10.5 trillion per year by 2025. In response, the banking sector is prioritizing the enhancement of cyber fraud protection, particularly in credit card transactions. To combat the evolving nature of threats, banks are continually investing in cutting-edge technologies and systems for detecting and preventing cyber fraud. The widespread adoption of credit card and online payments has led to a surge in various forms of credit card cyber fraud, making it imperative for the industry to stay at the forefront of cyber fraud management and security technologies.

3. Machine learning Algorithms

The machine learning models used in this project are Logistic Regression, Decision Tree classification, Random Forest classification, Support Vector Machine classification, K Nearest Neighbor classification, XGBoost classification.

3.1 Logistic Regression

Logistic Regression is a fundamental and widely used statistical model in machine learning for binary classification tasks. It's particularly suitable for problems where the outcome is binary (e.g., yes/no, 1/0), making it applicable to various real-world scenarios, including predicting customer churn, spam detection, or, in the context of banking, assessing creditworthiness. The model estimates the probability of an observation belonging to one of two classes, mapping its input features to a sigmoid-shaped curve that ranges between 0 and 1. This curve effectively separates the two classes, and a threshold can be applied to determine the final classification. Logistic Regression is valued for its simplicity, interpretability, and efficiency, making it a go-to choice for many classification tasks.

3.2 Decision Tree classification

A Decision Tree classification model is a popular and interpretable machine learning algorithm used for solving classification problems. It represents a tree-like structure, where each internal node denotes a feature or attribute, and each branch represents a decision rule or split based on that feature. The leaves of the tree correspond to the class labels or outcomes. Decision Trees are highly intuitive, making them accessible for non-technical users to understand. They are known for their ability to capture complex decision boundaries and interactions between features. However, they can be prone to overfitting, which can be mitigated by techniques like pruning and ensembling. Decision Trees are valuable in a wide range of applications, including risk assessment, medical diagnosis, and customer segmentation.

3.3 Random Forest classification

A Random Forest classification model is a powerful ensemble learning technique in machine learning. It combines multiple decision trees to make accurate and robust predictions. Each decision tree in the ensemble is constructed independently, making random decisions about which features to use and how to split the data. By aggregating the predictions from these trees, the Random Forest model reduces the risk of over-fitting and offers a more stable and reliable classification outcome. This makes it particularly effective for tasks such as predicting client creditworthiness for a bank, as it can handle complex, high-dimensional datasets and provide insights into the most influential factors affecting classification decisions.

3.4 Support Vector Machine

A Support Vector Machine (SVM) is a powerful supervised machine learning model used for classification and regression tasks. In classification, an SVM aims to find the optimal hyper-plane that best separates data points of different classes. It does so by identifying support vectors, which are the data points closest to the decision boundary. SVMs are effective in high-dimensional spaces and are versatile in handling linear and non-linear classification tasks through kernel functions. Their primary objective is to maximize the margin between classes, making them robust against over-fitting and capable of handling complex data

distributions. SVMs are widely used in various fields, including image recognition, text classification, and financial risk assessment, due to their ability to make accurate and efficient predictions.

3.5 K Nearest Neighbor classification

The K Nearest Neighbors (KNN) classification model is a straightforward and intuitive supervised machine learning algorithm used for both classification and regression tasks. In classification, KNN classifies a data point by considering the class labels of its K nearest neighbors in the feature space. It operates on the principle that similar data points tend to belong to the same class. By adjusting the value of K, one can control the model's sensitivity to local variations in the data. KNN is non-parametric, meaning it doesn't make assumptions about the underlying data distribution, making it versatile and suitable for a wide range of applications. However, its performance can be sensitive to the choice of K and the feature scaling, and it might not be ideal for high-dimensional data due to the "curse of dimensionality."

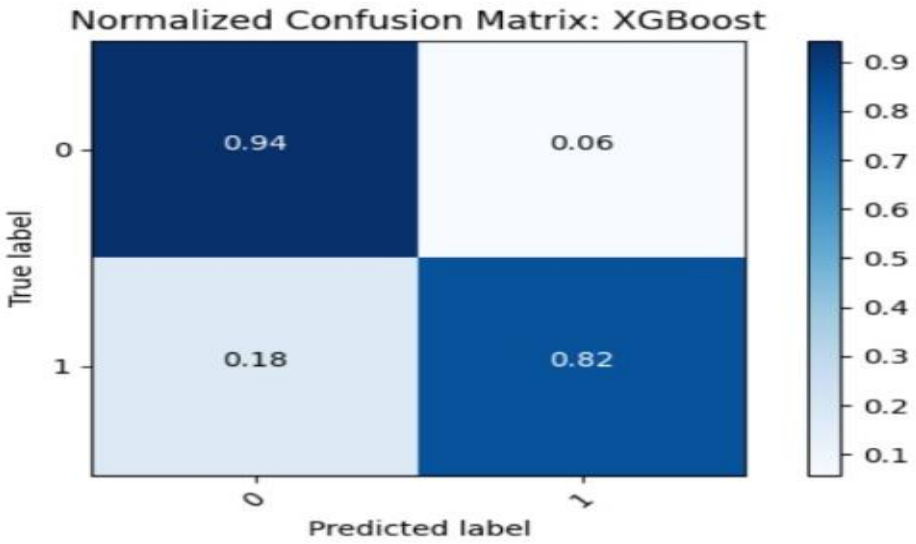
3.5 XGBoost classification

XGBoost, short for "Extreme Gradient Boosting," is a powerful machine learning algorithm known for its exceptional performance in classification tasks. It is an ensemble learning method that combines the predictions of multiple decision trees to achieve high accuracy and predictive power. XGBoost excels in handling structured data and is widely used in various applications, including predicting good clients for banks, detecting fraud, and many more. Its key features include the ability to handle missing data, robustness against overfitting, and the capability to provide feature importance scores, making it a favored choice in data science and machine learning for accurate and efficient classification tasks.

By conducting a comprehensive evaluation of various classification models, and the results indicate that the XGBoost classification model stands out as the best performer, achieving an impressive accuracy rate of 88.32%. XGBoost is renowned for its versatility and ability to excel in classification and regression tasks while mitigating overfitting. This outcome suggests that, in the context of predicting good clients for a bank, XGBoost proves to be the most effective and suitable model. It's a popular choice in real-world applications due to its

robustness, interpretability, and its capability to capture intricate patterns within the data.

By presenting and interpreting the confusion matrix, a comprehensive justification of the XGBoost classification model's performance can be provided.It allows stakeholders to assess the model's effectiveness in correctly classifying good clients while understanding its trade-offs, and it demonstrates the model's suitability for the task at hand.



4. Data analysis

4.1 Data Collection

Gathered and analyzed the data using suitable techniques, including statistical tests, EDA and machine learning algorithms. The dataset used is available publicly in the website of kaggle. It has two data set one having 1548 rows and 18 columns and other having 1548 rows and 2 columns.

4.2 Feature Engineering

Imputation

Utilizing mode imputation to address missing values in the dataset involves replacing the missing values with the mode, which is the most frequently occurring value for the respective feature. This approach is especially effective for categorical variables or discrete numerical variables where missing values can be imputed with a value that aligns with the majority of the data.

Handling Outliers

Addressing outliers through the technique of 'median imputation for outliers' entails the identification of data points that deviate notably from the usual value range, commonly referred to as outliers. Subsequently, these outlier data points are substituted with the median value calculated from the outlier subset.

Feature selection

Selecting features based on 'value counts' involves choosing variables by assessing how many different values or categories they have. This method focuses on features with a wide range of unique values, which can be beneficial, especially when working with categorical data. The goal is to pick features that exhibit diversity in their value counts, potentially making them more valuable for analysis or modeling

Encoding:

Label Encoding converts categorical data into numerical values by giving each category a distinct label or integer. This method is frequently employed to change categorical features into a format suitable for machine learning algorithms, which typically operate with numerical data. The label encoder assigns a unique numeric code to each category, simplifying data analysis and the construction of predictive models.

Feature Scaling:

Min-max scaling was used to standardize or normalize the values of features in a dataset. Min-max scaling ensures that the values of each feature are transformed to a common range, typically between 0 and 1, making it easier to compare and work

with these features in a consistent manner. This technique is often employed to prevent certain features from dominating others in various data analysis or modeling tasks.

Approach:

The chosen data analysis approach, incorporating techniques like feature engineering, median imputation for outliers, and min-max scaling, is well-justified for several reasons. It ensures data is prepared consistently and is ready for machine learning models while maintaining interpretability. Handling missing data pragmatically through median imputation and managing outliers prevents extreme values from unduly influencing results. Additionally, feature scaling enhances the performance of certain models and prevents one feature from dominating others. This approach aligns with the practical needs of the problem, enhancing data quality and reliability, which is vital for accurate credit card approval predictions.

4.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a powerful tool for detecting outliers that can have a substantial impact on the analysis. One effective method to visually identify and assess outliers is by creating box plots. Box plots provide a clear graphical representation of the distribution of data, allowing us to observe the presence of extreme values (outliers) that fall outside the typical range of the data.

EDA allows us to explore the relationships between variables. Correlation matrices helped to identify whether certain factors, like children_count or family_member are strongly correlated with credit card approval, validating the importance of these features.

EDA allows us to identify any data discrepancies or errors, which is crucial for ensuring data quality. Resolving such discrepancies justifies the data cleaning and preprocessing steps.

EDA revealed that some data transformations, like min-max scaling, have been beneficial in improving the distribution of features and normalizing data, enhancing the performance of machine learning models.

5. Methodology

5.1 Machine learning approach

Logistic Regression: Logistic regression is a linear model suitable for binary classification tasks like credit card approval. It's known for its simplicity and interpretability, making it an excellent choice for understanding the impact of individual features on approval decisions.

Decision Tree Classification: Decision trees are non-linear models that can capture complex relationships within the data. They divide the data into segments based on feature values, creating a tree-like structure that can help identify key decision points in the credit card approval process.

Random forest classification: Random forests are an ensemble method that combines multiple decision trees. They are effective at handling high-dimensional data and can improve predictive accuracy by reducing overfitting.

Support Vector Machine Classification: Support vector machines aim to find the optimal hyperplane that separates different classes in the data. They work well for binary classification tasks and can handle both linear and non-linear data patterns.

K Nearest Neighbor Classification: K-nearest neighbors classify data points based on the similarity to their neighbors. This method is especially useful for identifying patterns and making predictions based on the characteristics of neighboring data points.

XGBoost Classification: XGBoost is a gradient boosting algorithm that has gained popularity for its high predictive accuracy. It can handle complex relationships in the data and often delivers competitive results in various classification tasks.

5.2 Hyper-parameter Tuning

Conducting k-fold cross-validation is a valuable approach to bolster the accuracy and reliability of credit card approval prediction model. This technique evaluates the model's performance by dividing the dataset into k subsets, training and testing the model on various combinations of these subsets. It ensures that the model's accuracy is not merely due to overfitting, offering a more realistic assessment of its generalization ability. K-fold cross-validation also stabilizes performance estimates, accounts for hyperparameter tuning, aids in model selection, detects overfitting or underfitting, and provides insights into the model's bias and variance trade-off. Overall, it's a crucial step to ensure model excels in making accurate credit card approval predictions across diverse data scenarios.

After conducting k-fold cross validation the average accuracy of each models are listed below

Model	Average Accuracy
Logistic regression	88.69%
Decision Tree Classification	81.52%
Random forest classification	87.91%
Support Vector Machine Classification	88.71%
K Nearest Neighbor Classification	84.17%
XGBoost Classification	86.88%

6. Result

Logistic regression

```
Logistic Model Accuracy :  61.496350364963504 %

Confusion matrix :
[[153 121]
 [ 90 184]]

Classification report:
              precision    recall  f1-score   support

           0       0.63      0.56      0.59       274
           1       0.60      0.67      0.64       274

    accuracy                           0.61       548
   macro avg       0.62      0.61      0.61       548
weighted avg       0.62      0.61      0.61       548
```

Decision Tree Classification

```
Decision Tree Model Accuracy :  81.2043795620438 %

Confusion matrix :
[[239  35]
 [ 68 206]]

Classification report:
              precision    recall  f1-score   support

           0       0.78      0.87      0.82       274
           1       0.85      0.75      0.80       274

    accuracy                           0.81       548
   macro avg       0.82      0.81      0.81       548
weighted avg       0.82      0.81      0.81       548
```

Random forest classification

```
Random Forest Model Accuracy :  83.94160583941606 %

Confusion matrix :
[[253  21]
 [ 67 207]]

Classification report:
              precision    recall  f1-score   support

           0       0.79      0.92      0.85       274
           1       0.91      0.76      0.82       274

    accuracy                           0.84       548
   macro avg       0.85      0.84      0.84       548
weighted avg       0.85      0.84      0.84       548
```

Support Vector Machine Classification

```
Support Vector Classifier Accuracy :  73.54014598540147 %

Confusion matrix :
[[208  66]
 [ 79 195]]

Classification report:
              precision    recall  f1-score   support

           0       0.72      0.76      0.74       274
           1       0.75      0.71      0.73       274

    accuracy                           0.74       548
   macro avg       0.74      0.74      0.74       548
weighted avg       0.74      0.74      0.74       548
```

K Nearest Neighbor Classification

```
KNN Model Accuracy :  75.18248175182481 %

Confusion matrix :
[[215  59]
 [ 77 197]]

Classification report:
              precision    recall  f1-score   support

           0       0.74      0.78      0.76       274
           1       0.77      0.72      0.74       274

    accuracy                           0.75       548
   macro avg       0.75      0.75      0.75       548
weighted avg       0.75      0.75      0.75       548
```

XGBoost Classification

```
XGBoost Model Accuracy :  88.32116788321169 %

Confusion matrix :
[[258  16]
 [ 48 226]]

Classification report:
              precision    recall  f1-score   support

           0       0.84      0.94      0.89       274
           1       0.93      0.82      0.88       274

    accuracy                           0.88       548
   macro avg       0.89      0.88      0.88       548
weighted avg       0.89      0.88      0.88       548
```

7. Conclusion

The choice of XGBoost is the most suitable model for credit card approval prediction, with an accuracy of 88.32%, is well-justified by considering additional evaluation metrics. Precision, which measures the ability to minimize false positive errors, and the F1 score, which balances precision and recall, both contribute to the model's effectiveness. A high precision score ensures fewer false approvals, crucial for risk management in financial services. Meanwhile, a strong F1 score highlights a balanced approach between correctly identifying positive cases and minimizing errors. The examination of the confusion matrix provides comprehensive insights into the model's performance, helping identify potential issues and the consequences of false approvals or rejections.