Predictive Analysis for a credit card approval

Siri Shreshta Reddy Baddam
Department of Information System of University of Maryland Baltimore County

Harshith Babu Martha
Department of Information System of University of Maryland Baltimore County

Rohit Sankar Srinivas Gadi
Department of Information System of University of Maryland Baltimore County

***Abstract*— The banking sector's decisions about whether or not to approve a credit card applicant are never simple. They consistently do thorough investigation into applicants' creditworthiness. The goal of this research is to use machine learning to create a predictive model for credit card acceptance. By using predictive analysis, the models which better fits to understand whether to approve or decline a credit card is analyzed which are based on applicant's income total, income type, possessions etc. Decision tree, Logistic Regression, K-nearest neighbor, and random forest Machine Learning Models are used for this analysis.**

I. Introduction

The acceptance of credit cards is a crucial step for banks and other financial organizations. It entails assessing applicants' creditworthiness in light of a number of variables, including their income, credit history, and debt-to-income ratio. Credit card acceptance decisions have historically been determined using manual procedures and subjective standards, which may be slow, prone to prejudice, and time-consuming. It is difficult to emphasize the significance of precise credit card approval decisions. An incorrectly authorized credit card application might result in unpaid obligations for the financial institution, while an incorrectly refused application can cause lost income and unhappy customers.

However, the development of predictive models that may automate and enhance the credit card acceptance process has now become possible thanks to the development of machine learning techniques and the accessibility of vast amounts of data. With the use of machine learning algorithms, we want to create a predictive model for credit card acceptance in this project and assess its effectiveness using a variety of criteria. Our objective is to show how machine learning approaches may increase the effectiveness and precision of decision-making related to credit card approval.

As a part of our analysis, we are performing four machine learning models (Logistic Regression, KNN, Decision Tree and Random Forest). The performance metrics of these models are used to compare them such as accuracy, precision, recall, and F1 score. The Random Forest model results are proven to be more accurate in the end.

II. Literature Review

For decades, several research studies have emphasized the use of data analytical methods in business applications, including machine learning, statistical methods, and so on. Predictive analysis has garnered a significant amount of attention in the financial services sector, especially in relation to credit card approval processes. The utilization of predictive analytics allows for a more accurate assessment of applicants, leading to better risk management and profitability for financial institutions [5]. This literature review explores the latest research in this area, focusing on the application of different predictive analytic models and their effectiveness. Traditionally, credit scoring has involved the use of statistical models such as logistic regression and discriminant analysis. These models are capable of predicting the probability of default based on a set of financial and personal characteristics [8]. However, the evolution of machine learning and artificial intelligence has paved the way for more sophisticated and accurate models.

Machine learning models, such as decision trees, random forests, and support vector machines, have been used for credit scoring. These models can handle large amounts of data and identify complex patterns that are not easily detectable by traditional statistical models [8]. For instance, [1] compared the performance of different machine learning models for credit scoring and found that support vector machines performed the best.

With the rise of deep learning, neural networks have been increasingly applied to credit scoring. Neural networks have the advantage of being able to model non-linear relationships and can handle high-dimensional data. For instance, [7] demonstrated the effectiveness of neural networks in predicting credit card defaults. Similarly [10] proposed a fraud detection system using an Artificial Neural Network (ANN), which showed promising results in terms of both detection rate and false alarm rate. With the advent of big data, predictive analytics for credit scoring has advanced to new heights. [2] highlighted how big data technologies can enhance predictive analytics for credit scoring. They emphasized that big data, along with machine learning algorithms, can enhance the predictive power of credit scoring models.

Ensemble learning techniques, which combine multiple machine learning models to achieve better predictive performance, have been increasingly used in credit scoring [11] applied an ensemble learning approach that combined multiple decision trees, which proved to be more effective than single decision tree models. In addition to model selection, feature selection plays a crucial role in credit scoring. [4] proposed a two-stage feature selection method using Support Vector Machines (SVM) and Genetic Algorithms (GA). Their results demonstrated that feature selection could significantly improve the prediction performance of the SVM model.

The use of predictive analytics has also been extended to peer-to-peer lending platforms [9] applied various machine learning algorithms to predict loan defaults on the LendingClub platform and found that Random Forests outperformed other models. With the advent of machine learning, the fairness of credit scoring models has come into question. [12] pointed out potential issues of discrimination in machine learning-based credit scoring models and proposed methods to address these issues.

The interpretability of machine learning models is another key issue in credit scoring. [3] proposed a method for interpreting Random Forest models in credit scoring. Their method allowed for a better understanding of the decisions made by the model, which is crucial for regulatory compliance. More recently, hybrid models that combine different machine learning techniques have been proposed to improve the accuracy of credit scoring. For instance, [6] proposed a hybrid model that combines genetic algorithms and support vector machines. They found that their hybrid model outperformed the individual models in terms of accuracy. Predictive analysis in credit card approval has evolved significantly over the years. While traditional statistical models have served their purpose, the emergence of machine learning and deep learning models has provided a new avenue for improving the accuracy of credit scoring.

III. METHODOLOGY

A. Data Description

We are using the Credit Card Dataset available on Kaggle which has (https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction).

It consists of two records i.e., application record and credit records. Application_record.csv contains appliers personal information, which you could use as features for predicting. Credit_record.csv records users' behaviors of credit cards. Id is the common entry for both records.

Id is the common attribute among both the records and credit record has only 3 attributes Id , Months_Balance and Status

Below is the description of each attribute present in the dataset

1. ID: It is the unique identification number given to the applicant
2. CODE_GENDER: Describes the gender of the applicant
3. FLAG_OWN_CAR: It indicates whether the applicant owns a car or not
4. FLAG_OWN_REALTY: Indicates whether an applicant owns a property or not
5. CNT_CHILDREN: Indicates the number of Children
6. AMT_INCOME_TOTAL: Annual Income
7. NAME_INCOME_TYPE: Income Category
8. NAME_EDUCATION_TYPE: Educational level
9. NAME_FAMILY_STATUS: Indicates marital status
10. NAME_HOUSING_TYPE: Way of living
11. DAYS_BIRTH: Days lived
12. DAYS_EMPLOYED: Mentions about start date of employment
13. FLAG_MOBIL: Indicates whether an applicant owns a mobile phone or not
14. FLAG_WORK_PHONE: Indicates whether an applicant owns a work phone or not
15. FLAG_PHONE: Indicates whether an applicant has a work phone or not
16. FLAG_EMAIL: Is there an email or not
17. OCCUPATION_TYPE: Occupation Type
18. CNT_FAM_MEMBERS: Family size
19. MONTHS_BALANCE The month of the extracted data is the starting point, backward, 0 is the current month, -1 is the previous month, and so on
20. STATUS: 0-Customers who have paid their amount or didn't take a credit card or just have a due of 29 days ; 1-Customers who has an overdue after 29 days

B. Data Preprocessing

In our data preprocessing steps, we merged the "application_record" and "credit_record" datasets, resulting in a combined dataset with 777,715 entries. This merging process allowed us to incorporate credit-related information into our analysis, providing a more comprehensive view of the applicants' profiles.

Furthermore, we performed additional data cleansing procedures, including the removal of duplicate values, resulting in a dataset with enhanced accuracy and reliability. We initially identified 412,393 duplicate values within the dataset. By removing these duplicates, we obtained a cleaned dataset comprising 365,322 unique entries. The elimination of duplicates mitigated potential biases and inaccuracies that could have affected our subsequent analyses and findings.

Additionally, we addressed null values in the "Occupation type" variable by assigning them the label "Others." This step ensured that all entries in the dataset had a defined value for the "Occupation type" variable, maintaining dataset integrity and completeness. This cleaned dataset enables us to draw meaningful conclusions, make informed decisions, and gain valuable insights based on accurate and comprehensive information.

C. Data Analysis

Correlation refers to a statistical measure that indicates the strength and direction of a relationship between two variables. It provides insight into how changes in one variable are associated with changes in another variable. Status is our Target variable for further analysis. The correlation coefficient, often denoted as "r," ranges between -1 and 1. A positive correlation (ranging from 0 to 1) means that as one variable increases, the other tends to increase as well. Conversely, a negative correlation (ranging from -1 to 0) indicates that as one variable increases, the other tends to decrease.

A correlation coefficient of 1 or -1 represents a perfect correlation, indicating that the variables are completely related and move together in a linear fashion. A coefficient of 0 suggests no correlation, implying that the variables are independent of each other and do not exhibit a linear relationship.

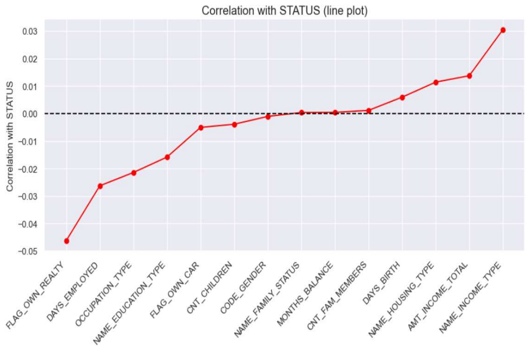


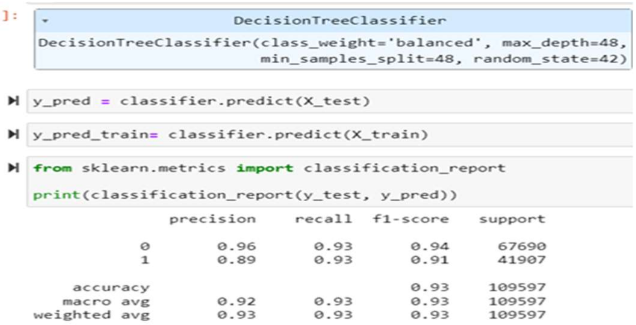
Fig. 1. Correlation Graph

The correlation coefficient, often denoted as "r," ranges between -1 and 1. A positive correlation
(ranging from 0 to 1) means that as one variable increases, the other tends to increase as well. From Fig.1 it is clear that Days_Birth, Name_Housing_ Type, Amt_income_total, name_income_type are all positively co related to the status variable.Conversely, a negative correlation (ranging from -1 to 0) indicates that as one variable increases, the other tends to decrease. From the above figure it is evident that Flag_own_reality, days_employed, occupation_type, name_education_type, Flag_own_car are all negatively co-related to the status variable. This indicates that these attributes are completely related and move together in a linear fashion. Name_Family_Status suggests 0 coefficient, implying that the variables are independent of each other and do not exhibit a linear relationship.

D. Model Evaluation

1) Decision tree :

In the project on predicting credit card approval, we used decision tree model as one of the machine learning algorithms to assess the credit card approval rate of applicants. Decision trees are intuitive and powerful models that facilitate decision-making by dividing data into smaller, more manageable subsets based on specific features and their thresholds. The decision tree model utilized in this project aimed to create a tree-like structure of decision rules that could classify credit card applications as approved or rejected. The model learned from a labeled dataset that consisted of various applicant attributes, such as amt_income_total, name_income_type, name_housing_type, days_birth etc.,

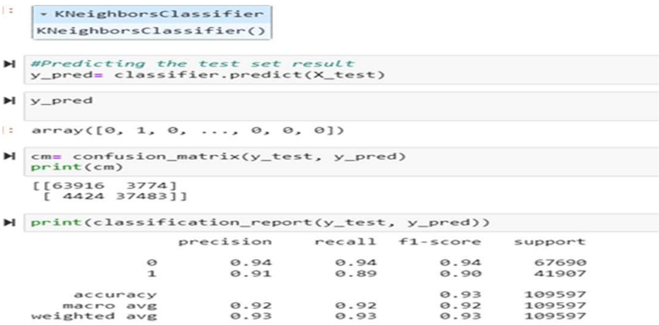


```
]:  DecisionTreeClassifier
    DecisionTreeClassifier(class_weight='balanced', max_depth=48,
                           min_samples_split=48, random_state=42)

y_pred = classifier.predict(X_test)

y_pred_train= classifier.predict(X_train)

from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.96      0.93      0.94     67690
           1       0.89      0.93      0.91     41907

    accuracy                           0.93    109597
   macro avg       0.92      0.93      0.93    109597
weighted avg       0.93      0.93      0.93    109597
```

Fig. 2. Confusion matrix and classification report of decision tree classifier algorithm

From Fig.2. The decision tree model performed commendably with its 93% accuracy which is similar to the KNN model, indicating its effectiveness in classifying credit card applications as approved or rejected. Moreover, decision trees are relatively resilient to missing data and can handle both categorical and numerical features without extensive preprocessing.

2) K-nearest neighbors :



```
]:  KNeighborsClassifier
    KNeighborsClassifier()

#Predicting the test set result
y_pred= classifier.predict(X_test)

y_pred

]:  array([0, 1, 0, ..., 0, 0, 0])

cm= confusion_matrix(y_test, y_pred)
print(cm)

[[63916  3774]
 [ 4424 37483]]

print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.94      0.94      0.94     67690
           1       0.91      0.89      0.90     41907

    accuracy                           0.93    109597
   macro avg       0.92      0.92      0.92    109597
weighted avg       0.93      0.93      0.93    109597
```

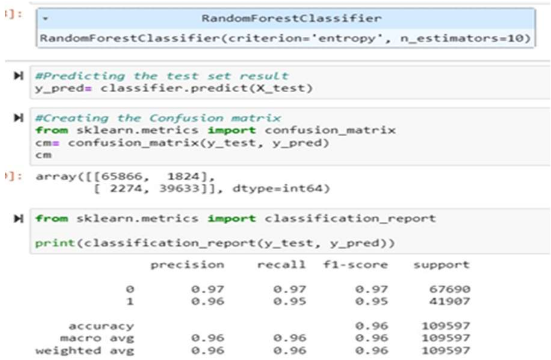
Fig. 3. Confusion matrix and classification report of k-nearest neighbor's algorithm

From Fig.3 the KNN model exhibited a competitive accuracy of 93%. KNN is a non-parametric algorithm that classifies new data points based on the majority vote of their neighboring points. While it lacks the interpretability of decision trees, it can be computationally efficient and can capture complex relationships in the data.

3) Random forest :

The random forest algorithm generates a "forest" of decision trees. A randomly chosen subset of the training data and a randomly chosen subset of characteristics are used to construct each tree. The risk of overfitting is decreased, and the trees' diversity is increased because to this random selection.

Each decision tree in the random forest separately learns to forecast credit card approval based on various subsets of the data and attributes during the training process. When making predictions, the random forest averages (for classification) or combines (for regression) the results of each individual tree, resulting in the final prediction.



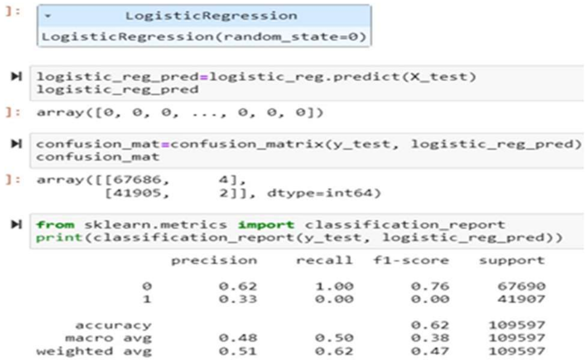
```
]:  RandomForestClassifier
    RandomForestClassifier(criterion='entropy', n_estimators=10)

#Predicting the test set result
y_pred= classifier.predict(X_test)

#Creating the Confusion matrix
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
cm

]:  array([[65866,  1824],
           [ 2274, 39633]], dtype=int64)

from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     67690
           1       0.96      0.95      0.95     41907

    accuracy                           0.96    109597
   macro avg       0.96      0.96      0.96    109597
weighted avg       0.96      0.96      0.96    109597
```

Fig. 4. Confusion matrix and classification report of random forest algorithm

From Fig. 4 The random forest model achieved the highest accuracy of 96%, outperforming both the decision tree and KNN models. It uses the ensemble learning method that combines multiple decision trees to make predictions. By aggregating the predictions of individual trees, random forests can mitigate the limitations of single decision trees and achieve higher accuracy. However, random forests may be more complex and less interpretable compared to decision trees.

4) Logistic regression:

Logistic regression is applied to this dataset. Since the Logistic Regression model just models the output probability in accordance with the input, it does not actually perform a classification. By selecting a cutoff value, the model can be utilized to create a classifier. The inputs with probability greater than the cut-off value can be categorized into one class to actualize the binary classifier, while the inputs with probability less than the cut-off value can be classed into a different class.



```
]:  ▾        LogisticRegression
    LogisticRegression(random_state=0)

▶|  logistic_reg_pred=logistic_reg.predict(X_test)
    logistic_reg_pred
]:  array([0, 0, 0, ..., 0, 0, 0])

▶|  confusion_mat=confusion_matrix(y_test, logistic_reg_pred)
    confusion_mat
]:  array([[67686,     4],
           [41905,     2]], dtype=int64)

▶|  from sklearn.metrics import classification_report
    print(classification_report(y_test, logistic_reg_pred))
                  precision    recall  f1-score   support

               0       0.62      1.00      0.76     67690
               1       0.33      0.00      0.00     41907

        accuracy                           0.62    109597
       macro avg       0.48      0.50      0.38    109597
    weighted avg       0.51      0.62      0.47    109597
```

Fig. 5. Confusion matrix and classification report of logistic regression algorithm

In Fig. 5 we can observe that the accuracy of the Logistic Regression algorithm is 0.62% which is the least among all the analyses. The accuracy of this classification is very less because there is no centrality in the dataset. That was the reason it does not predict the accuracy correctly

IV. RESULT

From the entire data available 70% of the data is used to train the models, while 30% of the data is utilized to test the models. all the performance criterions are found and roc and precision-recall curve is plotted for all the models. The results indicate all three models decision tree, KNN, and random forest were, to varied degrees, accurate in predicting credit card approval. While the accuracy of the decision tree and KNN models were both 93%, the random forest model fared better than both of them, with an accuracy of 96%. The most effective model for predicting credit card approval will be chosen based on the application's requirements and restrictions, interpretability, and computational efficiency.

Model	Precision	Recall	F1-Score	Accuracy
Decision tree	0.96	0.93	0.94	0.93
Random forest	0.97	0.97	0.97	0.96
KNN	0.94	0.94	0.94	0.93
Logistic Regression	0.62	1.00	0.76	0.62

Table 1: Comparison between different performance metrics of the model

A. *ROC curve:*

The ROC curve is a graphical representation of the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at various classification thresholds. It helps to visualize the model's performance across different threshold settings. The ROC curve plots the true positive rate (TPR) on the y-axis against the false positive rate (FPR) on the x-axis. A higher area under the curve (AUC) indicates better discrimination and performance.

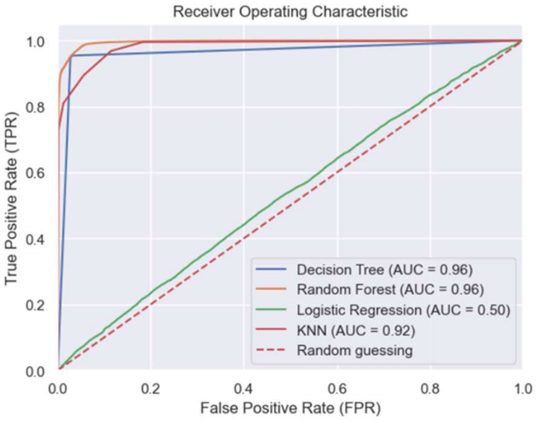


Fig. 6. Comparison graph between receiver operating characteristics of models

From Fig. 5 It is clear to see that the AUC of the random forest is more compared to others. So Receiver Operating Characteristic curve of random forest at all classification thresholds is better than all other classification models.

B. *Precision-recall curve:*

The Precision-Recall (PR) curve, on the other hand, illustrates the trade-off between precision (positive predictive value) and recall (sensitivity) at different classification thresholds. The PR curve is especially useful when dealing with imbalanced datasets where the positive class (approved applications) is rare. It focuses on the model's ability to correctly classify positive instances. The PR curve plots precision on the y-axis against recall on the x-axis. A higher area under the curve indicates better precision and recall balance.

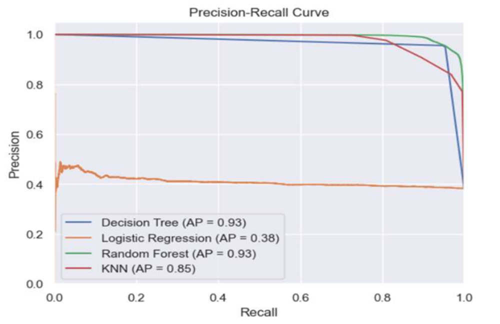


Fig. 7. Comparison graph between precision-recall curves of models

From fig.7 The PR curve focuses on the models' precision-recall balance, particularly important in imbalanced datasets. A higher area under the PR curve indicates better precision and recall trade-off. we can observe that the precision-recall curves of all the models analyzed, and we can find the random forest method has more AUC compared to others. So we can say that random forest performed well.

V. CONCLUSION

In conclusion, our analysis focused on the prediction of credit card approvals using a combination of machine learning algorithms such as decision tree, random forest, KNN and logistic regression. The objective was to develop a reliable model that could effectively predict the approval rate of applicants and make accurate predictions regarding credit card approval.

The results demonstrated that the random forest model achieved a high level of accuracy in determining credit card approval. The model successfully captured complex patterns and relationships within the data, enabling it to make reliable predictions and minimize the risk of approving applicants with low approval rate or rejecting potential creditworthy individuals. It is crucial to remember that the performance of the model should be regularly assessed and improved as new data become available. Additionally, other features and algorithms that can improve the model's predictive skills might be explored through further study and experimentation.

These outcomes have significant implications for credit card issuers and financial institutions, as it provides them with a powerful tool for automating and optimizing the credit approval process. By leveraging machine learning techniques, lenders can streamline their operations, improve decision-making, and reduce the potential for human bias. The prediction of credit card approval successfully demonstrated the potential of machine learning in the realm of credit risk assessment, offering a more efficient and accurate approach to credit card approval decisions.

REFERENCES

[1] Abdou, H., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: a review of the literature. Intelligent Systems in Accounting, Finance & Management, 18(2-3), 59-88.

[2] Baesens, B., Bapna, R., Marsden, J. R., Vanthienen, J., & Zhao, J. L. (2016). Transformational Issues of Big Data and Analytics in Networked Business. MIS Quarterly, 40(4).

[3] Carvalho, A., Pereira, R., & Cardoso, J. (2019). Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics, 8(8), 832.

[4] Chen, L., Li, J., & Zhou, X. (2012). A two-stage SVM method to predict credit scores. 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, 2, 641-645.

[5] Hand, D. J., & Henley, W. E. (1997). Statistical Classification Methods in Consumer Credit Scoring: a Review. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523-541.

[6] Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. Expert systems with applications, 33(4), 847-856.

[7] 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.

[8] 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

[9] Malekipirbazari, M., & Aksakalli, V. (2015). Risk assessment in social lending via random forests. Expert Systems with Applications, 42(10), 4621-4631.

[10] Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications, 40(15), 5916-5923.

[11] Zhou, L., Wang, M., & Chen, Q. (2010). Ensemble learning for credit scoring: An ensemble of support vector machine classifiers. 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 3, V3-641.

[12] Žliobaitė, I. (2017). Measuring discrimination in algorithmic decision making. Data Mining and Knowledge Discovery, 31(4), 1060-1089.