iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XI **Month of publication:** November 2022

DOI: https://doi.org/10.22214/ijraset.2022.47369

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

Credit Card Approval Prediction using Classification Algorithms

Naman Dalsania[1], Devang Punatar[2], Deep Kothari[3]
[1]*Information Technology Department, NMIMS University, Mumbai,* [2,3]*Computer Engineering Department, Thakur College of Engineering and Technology, Mumbai*

Abstract: ***Credit risk as the boards in banks basically revolves around determining the probability of default or the creditworthiness of a customer, collapse, and the cost, assuming it happens. It is important to consider key factors and anticipate the likelihood of consumer default, given the circumstances.***
This is where machine learning models come into play. This allows banks and large financial institutions to predict whether their customers will default on their loans. This project uses Python to create machine-learning models with the highest possible accuracy.
First, we load the dataset and take a glimpse. The data set is a combination of mathematical and non-mathematical elements, with various ranges of values and some missing points. We pre-process the dataset so that the selected ML model meets high expectations.
Once the information looks good, an exploratory information check is performed to glean instincts. Finally, we created a machine learning model that can predict whether an individual's credit card application will be approved. This project uses the Jupyter Python programming notebook to create a machine-learning model. This project used data analytics and machine learning to determine the most important parameters for credit card approval. The machine learning model we built is based on the idea that a credit card will get approved or not, considering various factors listed in the credit cardholder's application. We have analysed three algorithms using precision measures including F1 score, precision, and recall. We got the highest accuracy of 90% from Gradient Boosting Classifier out of the other two models that we applied i.e., Support Vector Classifier and Adaboost classifier.
Keywords: Machine Learning, Data Analysis, Credit Card Approval, Credit Score.

I. INTRODUCTION

Credit approval, such as for credit cards, is crucial to the modern economy. In today's interconnected globe, even in developing nations such as India, the use of credit cards is no longer a fantasy.
Credit acceptance remains a challenge for moneylenders, as it is difficult to forecast whether consumers pose an acceptable credit risk and should be granted credit. This is especially true in emerging nations, where established rules and models from industrialized nations may not apply. Therefore, productive methods for automatic credit approval that can aid bankers in analysing consumer credit must be investigated.
Each bank receives tens of thousands of credit card applications each month. Banks have to manually skim through each of these applications, while paying close attention to these factors to determine whether the applicant is to be granted a credit card
or not. Due to the time-intensive nature of this activity and the growing likelihood of error as the number of applications increases, banks are seeking prediction-based algorithms that can do this task effectively and accurately.
In this paper, we predict if an applicant will be approved for a credit card or not using few machine learning algorithms. To begin with, we pre-processed the data and performed thorough EDA to better comprehend the factors that are crucial for training the model. In addition, we have implemented ten machine learning algorithms on these pre-processed data to identify the model that provides the most accurate results given the precision-recall trade-off.
The essay has been organized in the following way. In Section II, we have described our findings from the literature review we performed. In Section III, we explain the entire system in detail. We also demonstrated and analysed the results and compared them with a different view. Finally, we concluded the outcomes and observations.

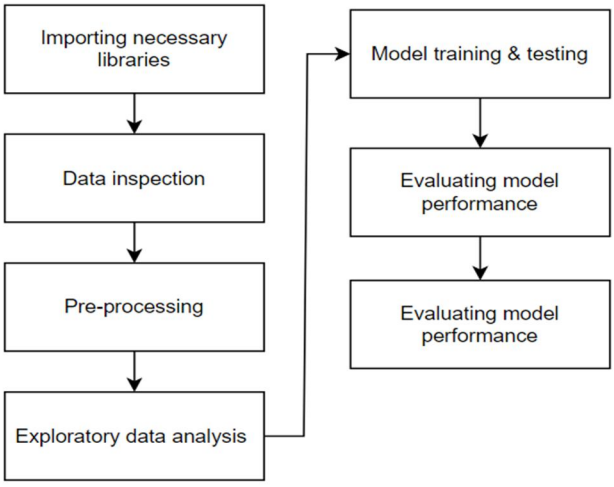


Fig. 1 Experiment Flowchart

II. LITERATURE REVIEW

A. *Comparison of Different Supervised Machine Learning Classifiers to Predict Credit Card Approvals (IRJET)*

This study contrasts various supervised machine learning models to forecast the likelihood that a credit card request will be accepted based on various criteria like Precision, Recall, Time, Accuracy, and F1 Score. The aim here was to identify the best classifier for automatically predicting credit card approval based on the characteristics of credit card applications. The analysis also demonstrates that every classifier performs better in one or more metrics. To improve the performance of each model, the method used hyperparameter optimization based on GridSearchCV to optimise certain parameters. The UCI Machine Learning Repository dataset used in this work was unbalanced and hence F1 score was relied on up to test the models. The classifiers used in this study are Logistic Regression, Random Forest, Decision Tree, XGBoost, Gradient Boost, Support Vector Machine (SVM), and Sequential Neural Network. Finally, based on F1 Score and AUC value, the research finds that Random Forest classifier is the best model for predicting Credit Card approvals with a F1 score of 86%. Although the research tries multiple machine learning models to test the dataset, there was no attempt made to balance out the data for better results.

B. *Predicting Credit Card Approval of Customers Through Customer Profiling using Machine Learning (IJEAT)*

This study focuses on forecasting credit card approval for users using a limited number of algorithms. The data was taken from bank customers in 2 ways, primary data and secondary data and then combined into one. These customer datasets were fully gathered, evaluated, and trained. These trained datasets helped in predicting whether credit card applications from customers will be approved. Since only a small number of variables were employed to determine the final decision, the training and testing accuracy of both decision tree and k nearest neighbour algorithms were roughly 99.7% and 99.6%, respectively. The training and testing accuracies of the decision and knn algorithms would alter in real time as more datasets are trained and tested and as the variables for the final choice are raised. This study however falls short in testing various other classification algorithms that could show better results in the future when more variables are taken in consideration.

C. *Credit Card Approval Predictions Using Logistic Regression, Linear SVM and Naïve Bayes Classifier (IEEE)*

This paper compares the prediction accuracy of Logistic Regression, Linear SVM and Naïve Bayes Classifier in the credit card approval process, with the Balanced Accuracy as the performance criteria. The dataset contains 2 types of features, numerical and categorical. Some of them include debt, age, income, education, income, etc. Credit applicants are split into "good credit" and "poor credit" categories according to the credit scoring algorithm. Based on the model implementation, Linear SVM has showcased the best prediction performance among the models, with a Balanced Accuracy of around 89%. However, the performance for each model would fluctuate slightly depending on the data processing, parameter tuning process and data features. One of the limitations to this paper is that further comprehensive factors such as the computational efficiency, reject inference and outlier handling to assess the prediction performance are not included.

III. PROPOSED SYSTEM

1) *Dataset:* The dataset has been taken from Kaggle's Credit Card Approval Prediction page. We have merged two datasets containing application and credit records of the applicants on primary key 'ID'. After merger, our columns contain variety of information of the applicant through which the lending corporation can easily take a decision whether to lend out to a particular candidate.

application_record.csv		
Feature name	Explanation	Remarks
ID	Client number	
CODE_GENDER	Gender	
FLAG_OWN_CAR	Is there a car	
FLAG_OWN_REALTY	Is there a property	
CNT_CHILDREN	Number of children	
AMT_INCOME_TOTAL	Annual income	
NAME_INCOME_TYPE	Income category	
NAME_EDUCATION_TYPE	Education level	
NAME_FAMILY_STATUS	Marital status	
NAME_HOUSING_TYPE	Way of living	
DAYS_BIRTH	Birthday	Count backwards from current day (0), -1 means yesterday
DAYS_EMPLOYED	Start date of employment	Count backwards from current day(0). If positive, it means the person currently unemployed.
FLAG_MOBIL	Is there a mobile phone	
FLAG_WORK_PHONE	Is there a work phone	
FLAG_PHONE	Is there a phone	
FLAG_EMAIL	Is there an email	
OCCUPATION_TYPE	Occupation	
CNT_FAM_MEMBERS	Family size	

Fig. 2 application_record.csv

credit_record.csv		
Feature name	Explanation	Remarks
ID	Client number	
MONTHS_BALANCE	Record month	The month of the extracted data is the starting point, backwards, 0 is the current month, -1 is the previous month, and so on
STATUS	Status	0: 1-29 days past due 1: 30-59 days past due 2: 60-89 days overdue 3: 90-119 days overdue 4: 120-149 days overdue 5: Overdue or bad debts, write-offs for more than 150 days C: paid off that month X: No loan for the month

Fig. 3 Credit_record.csv

2) *Pre-processing:* The dataset had column names in the camel case format which we converted into more readable format. In addition, we have conducted feature engineering, in which we have addressed skewness in our data, executed one-hot encoding, and min-max scaling, among other things.

```
def full_pipeline(df):
    # Create the pipeline that will call all the class from OutlierRemoval to OversampleSMOTE in one go
    pipeline = Pipeline([
        ('outlier_remover', OutlierRemover()),
        ('feature_dropper', DropFeatures()),
        ('time_conversion_handler', TimeConversionHandler()),
        ('retiree_handler', RetireeHandler()),
        ('skewness_handler', SkewnessHandler()),
        ('binning_num_to_yn', BinningNumToYN()),
        ('one_hot_with_feat_names', OneHotWithFeatNames()),
        ('ordinal_feat_names', OrdinalFeatNames()),
        ('min_max_with_feat_names', MinMaxWithFeatNames()),
        ('change_to_num_target', ChangeToNumTarget()),
        ('oversample', Oversample())
    ])
    df_pipe_prep = pipeline.fit_transform(df)
    return df_pipe_prep
```

Fig. 4 Data pre-processing pipeline

3) *Exploratory Data Analysis:* This part has been extensively covered to graphically plot and examine the values and correlations of all variables so that banks and lending institutions can identify the variables they wish to include and provide specific importance in their model. To analyse the relationships between numerous features, ANOVA and chi-square tests were conducted.

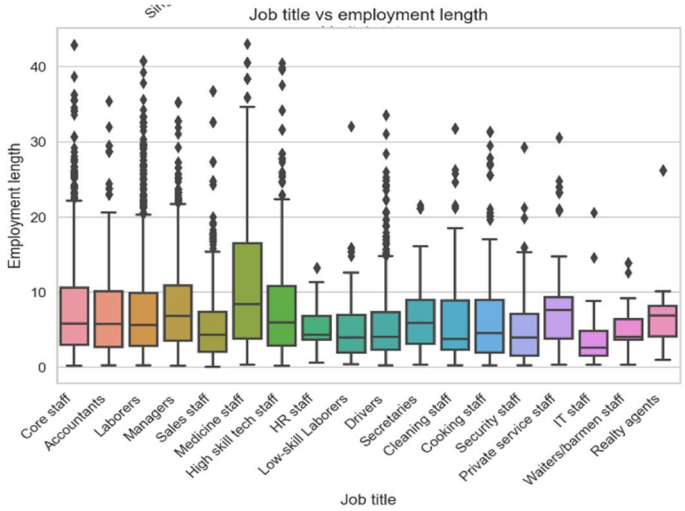


Fig. 5 ANOVA chart for age vs job title

4) *Development of Models:* We have used 10 different classification models to find the one which suits our data the best. We have explained the useful ones below.

One more thing, we cannot consider accuracy to be a reliable parameter. Precision and recall may have been considered instead of accuracy. These two play an important role in model evaluation. Unfortunately, it's impossible to maximize both metrics at the same time. The F1 score helps balance both precision and recall. Therefore, models are compared based on F1 scores.

A. *Support Vector Classifier*

The goal of the SVM algorithm is to create optimal lines or decision boundaries that can divide the n-dimensional space into classes so that new data points can be easily placed in the correct category in the future. This optimal decision boundary is called a hyperplane. SVM chooses extreme points/vectors that help create hyperplanes. These extreme cases are called support vectors, and the algorithm is called a support vector machine.

---------- support_vector_machine ----------

	precision	recall	f1-score	support
0	0.87	0.82	0.85	23272
1	0.83	0.88	0.86	23272
accuracy			0.85	46544
macro avg	0.85	0.85	0.85	46544
weighted avg	0.85	0.85	0.85	46544

Fig. 6 Accuracy table for SVC

We also plotted a Receiver Operator Characteristic (ROC) curve as it is a graphical plot used to show the diagnostic ability of binary classifiers. Classifiers that give curves closer to the top-left corner indicates a better performance.

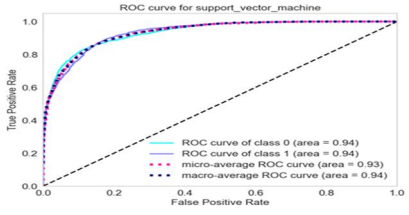


Fig. 7 ROC curve for SVC

As we can see here that our curve is at the top-left corner which indicates that our model performed well.

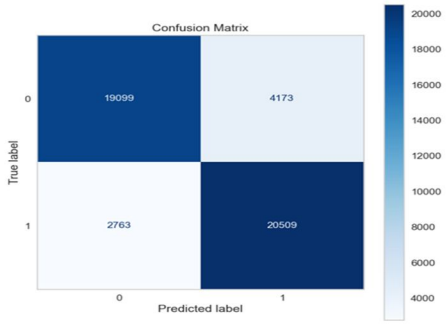


Fig. 8 Confusion matrix for SVC

From the confusion matrix, we can see that Support Vector Classifier predicted only a small percentage of values as False Positives and False Negatives. This shows that SVM performed well but our motive is to reduce the number of False Positives and False Negatives.

B. Adaboost Classifier

The AdaBoost classifier combines multiple poorly performing classifiers to build a strong classifier, providing a highly accurate strong classifier. The basic concept behind Adaboost is to set classifier weights and train data samples at each iteration to ensure accurate prediction of anomalous observations.

---------- adaboost ----------

	precision	recall	f1-score	support
0	0.77	0.75	0.76	23272
1	0.75	0.78	0.77	23272
accuracy			0.76	46544
macro avg	0.76	0.76	0.76	46544
weighted avg	0.76	0.76	0.76	46544

Fig. 9 Accuracy table for Adaboost Classifier

The F1 score of this model is less than SVC.

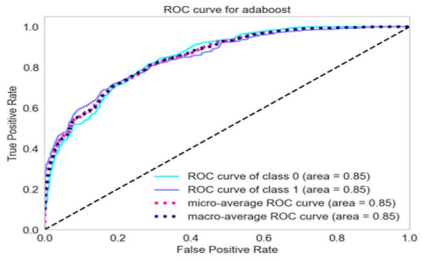


Fig. 10 ROC curve for Adaboost Classifier

The ROC curve of this model is not that good as compared to SVC.

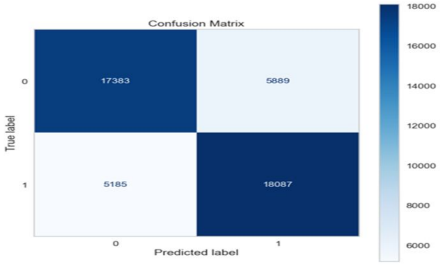


Fig. 11 Confusion matrix for Adaboost Classifier

Here, the number of false positives and false negatives increased which indicates this model did not perform well.

C. Gradient Boosting Classifier

The principle behind boosting algorithms is first we build a model on the training dataset, then a second model is built to rectify the errors present in the first model. The objective of gradient boosting classifier is to is to minimize this loss function by adding weak learners using gradient descent.

---------- gradient_boosting ----------

	precision	recall	f1-score	support
0	0.90	0.90	0.90	23272
1	0.90	0.90	0.90	23272
accuracy			0.90	46544
macro avg	0.90	0.90	0.90	46544
weighted avg	0.90	0.90	0.90	46544

Fig. 12 Accuracy table for Gradient Boosting Classifier

The F1 score of Gradient Boosting is better than SVC.

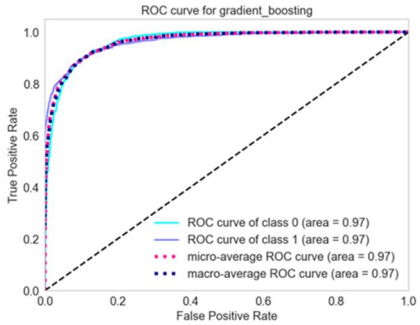


Fig. 13 ROC curve for Gradient Boosting Classifier

The ROC curve of Gradient boosting is a little better than the ROC curve of SVC.

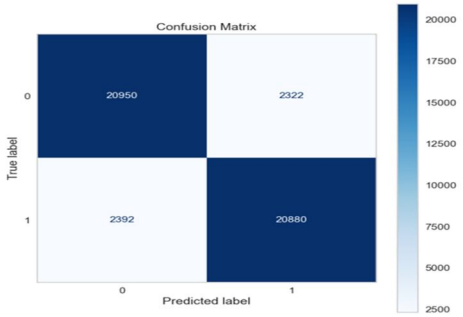


Fig. 14 Confusion matrix for Gradient Boosting Classifier

Here, the number of False Positives and False Negatives decreased. This model performed well than SVC.

Since the purpose of this problem is to minimise the risk of loan default for the financial institution, the criteria that should be employed depend on the present economic climate:

1) During a bull market (when the economy is expanding), people feel prosperous and are typically employed. Typically, money is inexpensive, and the danger of default is low. Since the financial institution can manage the risk of default, it is not overly stringent when extending loans. The financial organisation can accommodate a small number of undesirable customers so long as most applicants are desirable (aka those who payback their credit). Ideal in this situation is a high recall (sensitivity) rate.

2) People lose their employment and their money through the stock market during a bear market (when the economy is contracting). Numerous individuals struggle to fulfil their financial obligations. Therefore, the financial institution tends to be more careful when extending credit or loans. The financial firm cannot afford to extend credit to customers who will be unable to repay it. The financial organisation would prefer to have fewer good customers, even if it means denying credit to some of them, than to have any bad customers. In this circumstance, precision (specificity) is desired.

Since we are currently in the longest bull market (excluding the flash crash in March 2020), we will utilise recall as our gauge.
In conclusion, gradient boosting classifier is the best performing model using ROC curve and recall.

IV. RESULTS

We are considering only those values whose class is '1' i.e., when the credit card gets approved.

Table I: Comparing Accuracies Obtained For Different Algorithms

Model	Accuracy	Precision	Recall	F1-Score
Adaboost Classifier	0.76	0.75	0.78	0.77
Support Vector Classifier	0.85	0.83	0.88	0.86
Gradient Boosting Algorithm	0.90	0.90	0.90	0.90

V. CONCLUSION

In this paper, we have mentioned various machine learning methods to predict whether a credit card will be approved for an individual or not. Several parameters were taken into consideration as these parameters make the model more effective and help institutions make better decisions to avoid fraud and losses. We applied a lot of data pre-processing techniques as good amount of data pre-processing contributes effectively to developing better performance of traditional machine learning models. During Exploratory Data Analysis, we plotted a lot of graphs and charts to study the dataset deeply so that we can get a better understanding of the dataset. This was done so that we can decide which models to apply which can perform well on this dataset and can correctly predict whether to approve a credit card or not. This prediction system can be helpful to various banks as it makes their task easier and increases efficiency as compared to the manual system which is currently used by many banks and this system is cost effective.

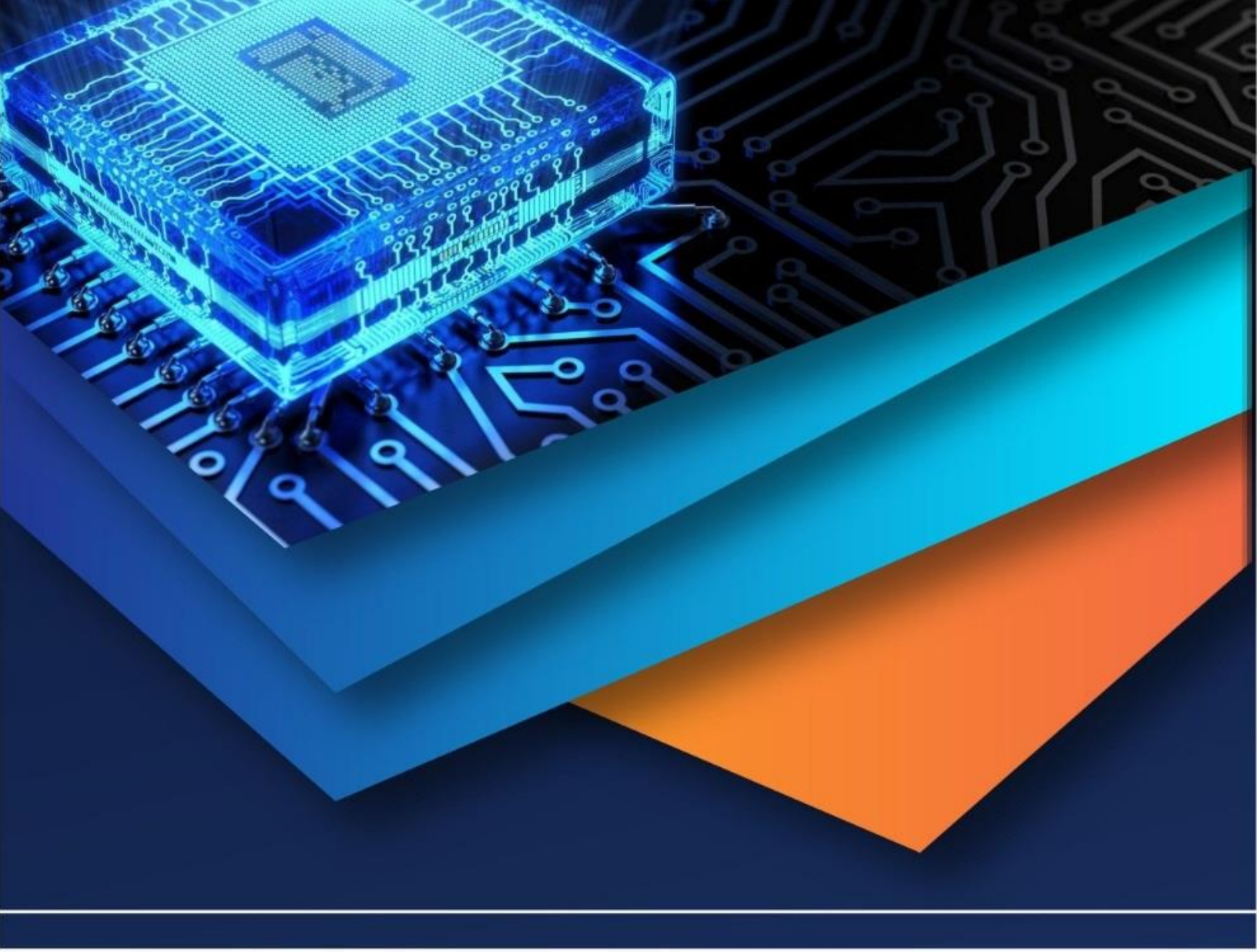
VI. FUTURE SCOPE

To further improve our system, we can use deep learning models as it can increase our accuracy. Neural networks can be used as it can discover hidden patterns and correlations in raw data, cluster and classify it, and continuously learn and improve over time. In the future, this credit card approval system will be able to be optimized and implemented in an artificial intelligence environment. By displaying the prediction result on a web or desktop application, the system can also be automated. Thus, this work has a good future scope and can be enhanced by adding other various feature for better predictions.

REFERENCES

[1] Siddhi Bansal, Tushar Punjabi Comparison of Different Supervised Machine Learning Classifiers to Predict Credit Card Approvals IRJET-, Volume: 08 Issue: 03 2021
[2] Arokiaraj Christian St Hubert, R. Vimalesh, M. Ranjith, S. Aravind Raj Predicting Credit Card Approval of Customers Through Customer Profiling using Machine Learning IJEAT- Volume-9 Issue-6, April 2021.
[3] Yiran Zhao University of Toronto Credit Card Approval Predictions using Logistic Regression, Linear SVM, and Naïve Bayes Classifier 2022 International Conference on Machine Learning and Knowledge Engineering.
[4] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," in IEEE Access, vol. 8, pp. 25579-25587, 2020, doi: 10.1109/ACCESS.2020.2971354.Niloy, NH. (2018). Naïve Bayesian Classifier and Classification Trees for the Predictive Accuracy of Probability of Default Credit Card Clients. American Journal of Data Mining and Knowledge Discovery. 3. 1. 10.11648/j.ajdmkd.20180301.11.
[5] Husejinovic, Admel & Kečo, Dino & Mašetić, Zerina. (2018). Application of Machine Learning Algorithms in Credit Card Default Payment Prediction. International Journal of Scientific Research. 7. 425. 10.15373/22778179#husejinovic.
[6] D. J. C. MacKay, "Comparison of Approximate Methods for Handling Hyperparameters," in Neural Computation, vol. 11, no. 5, pp. 1035-1068, 1 July 1999, doi: 10.1162/089976699300016331.
[7] Song, Jong-Woo. (2008). A Comparison of Classification Methods for Credit Card Approval Using R. Journal of the Korean society for quality management. 36.
Narkhede, Sarang. "Understanding Confusion Matrix." Medium, Towards Data Science, 29 Aug. 2019, towardsdatascience.com/understanding-confusionmatrix-a9ad42dcfd62.

[8] D. Prusti and S. K. Rath, "Web service-based credit card fraud detection by applying machine learning techniques," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 492-497, doi: 10.1109/TENCON.2019.8929372
[9] Liu, R., 2018. Machine Learning Approaches to Predict Default of Credit Card Clients. Modern Economy, 09(11), pp.1828-1838.
[10] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, "Random Forest for credit card fraud detection," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, 2018, pp. 1-6, doi: 10.1109/ICNSC.2018.8361343.
[11] Shung, Koo Ping. "Accuracy, Precision, Recall or F1?" Medium, Towards Data Science, 10 Apr. 2020, towardsdatascience.com/accuracy-precision-recall-orf1-331fb37c5cb9?gi=8377df893c73.
[12] S. Khatri, A. Arora and A. P. Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 680-683, doi: 10.1109/Confluence47617.2020.9057851.
[13] Brownlee, Jason. "Your First Deep Learning Project in Python with Keras Step-By-Step." Machine Learning Mastery, 16 Apr. 2020, machinelearningmastery.com/tutorial-first-neuralnetwork-python-keras/.
[14] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," in IEEE Access, vol. 8, pp. 25579-25587, 2020, doi: 10.1109/ACCESS.2020.2971354.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY