International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 03 | Mar 2021 www.irjet.net p-ISSN: 2395-0072

Comparison of Different Supervised Machine Learning Classifiers to Predict Credit Card Approvals

Siddhi Bansal[1], Tushar Punjabi[2]

[1]*Siddhi Bansal, Presidency University, Bangalore, India*
[2]*Tushar Punjabi, Presidency University, Bangalore, India*

Abstract - *Credit Cards have taken a crucial part in our daily financial routine. They have revolutionized the way of creating cashless payments and made making any kind of payment convenient for the customer. Accordingly, requests for Credit Cards within the banking sector is exponentially increasing, and manually considering each application is often a tedious job, also susceptible to human errors. Nowadays, Machine Learning models are used to automate such tasks. The decision of accepting a card request depends on the private and financial background of the candidate. Major factors like income, credit history, prior default, and lots of other attributes add on for the approval decision. It is very essential to seem after the credit risk and handle provocations for approval decisions because it can have conflicting results on credit management. Therefore, the judgement of approving application is vital before getting to any granting decision. This paper compares different Supervised Machine Learning models to predict how likely a Credit Card request will be approved based on the parameters like Precision, Recall, Time, Accuracy, F1-Score. The experimental results indicate that Random Forest Classifier is the best-suited model according to its F1-Score.*

Key Words: **Machine Learning, Model Evaluation, Model Performance, Credit Card Approval.**

1. INTRODUCTION

Approving for credit, e.g. payroll services and credit cards is an essential part of a developed economy. In the present interconnected world, even in developing countries like India, the use of, credit cards are no more a dream. However, for moneylenders, credit approval is still a problem as it is difficult to predict which customers represent an acceptable credit risk and should be granted credit. This is specifically valid in developing countries, as the established guidelines and models from developed countries may not be applicable. There is thus a need to research productive ways for automatic credit approval that can assist bankers in assessing consumer credit.

This paper inspects the credit application data taken from the UCI machine learning repository. Various pre-processing techniques like exploratory data analysis, data mining, and transformations like handling missing values, continuous values, and categorical values are computed [2], [38]. Data visualization techniques are also adopted to understand the data. A few Machine Learning models including Logistic Regression, Sequential Neural Network, Random Forest are generated and implemented on the data along with hyperparameter tuning using GridSearchCV.

The background of this study involves data collection, data cleaning, data analysis, data visualization, and implementing some classifiers in Python.

The objective of this paper is to find the appropriate classifier to automatically predict the approval of Credit Cards based on the attributes of the Credit Card application. The study also shows that each classifier outperforms in one or the other metric.

The main contribution of this paper is an intelligent approach to predict Credit Card approval using efficient Machine Learning models in which GridSearchCV based hyperparameters optimization is implemented to optimize certain parameters in order to increase the performance of each model. The performance of each model is evaluated based on several metrics.

The organization of the paper is structured as follows. Section II consists of related works on prediction of Credit Card approval. In Section III, information related to the experimental setup, data and its processing is provided. A brief explanation of the models used for the comparison can be seen in Section IV. Experimental analysis of Confusion Matrix is presented in Section V. Section VI shows the comparison of the classifiers on the basis of certain parameters. Section VII gives a brief information on ROC-AUC Curve, also shows the AUC value from the plot for each classifier. Concluding remarks and recommendations for further work are given in Section VIII.

2. RELATED WORKS

We collected, analyzed around 30-40 different papers published in different conferences, out of which we could find some papers which were related to our work, among these, one used the Australian Credit Card Approval dataset and performed the prediction using neural networks and went through the complex decision making process, therefore proved that ANN can be used as a reliable tool for credit card approval. Few experiments in this research were done using two and three layers with back-propagation training method [21].

To add on to this there was one more work which compared some established Machine Learning algorithms to differentiate between genuine and fraudulent transactions. Parameters like Precision, Sensitivity, and Time evaluated the performance of the models. On the basis of values obtained for sensitivity and time taken for execution, the paper concluded Decision Tree as the best classifier for Credit Card fraud detection [1].

Other related works used Genetic Programming for prediction wherein eight different Genetic Programming (GP) approaches for classification rule mining of a Credit Card application dataset used. The Australian Credit Card Approval dataset was also investigated using both strongly typed GP and Booleanizing technique. The contribution of this paper involved reduction of data pre-processing as GP was successfully implemented with missing data values [7].

Recently, a related study on profiling bank customer's behavior was also published which used k-mean, improved k-mean, fuzzy c-means, and neural networks to inspect one's profile for granting credit. The main objective of this study was to target Neural Network classification so that the clustering execution time reduces and the accuracy increases. This study resulted Neural Network classifier to be the best when accuracy ratio was taken as the parameter for the comparison [19].

Another work which compared different classification techniques, including nearest neighbor, Bayes classifier, discriminant analysis, and logistic regression to develop a web-based fraud detection model. Web services like, Representational State Transfer (REST) and Simple Object Access Protocol (SOAP) have also been implemented in this study for efficient exchange of data across the heterogeneous platforms. This research contributed an efficient classification algorithm which was successful in identifying fraud transactions [20].

Other related work checks the applicability of a new integrated model which is a combination model including techniques like Radial Basis Neural Network, Perceptron Model, Support Vector Machine, Decision tree (C4.5), Logistic Regression, and Multilayer Perceptron Model to classify credit application. The main objective of this study was to identify an integrated model combining the advantages of all the above techniques [22].

Therefore, we can observe that tackling this problem has many approaches and, also every model prepared will lead the model to predict with different accuracies.

3. EXPERIMENTAL SETUP

The first step in any study is to get the dataset and codebook. A quick analysis of the codebook gives information about the values in the dataset that have been converted to meaningless symbols to keep the data confidential.

TABLE 1: Dataset Codebook

Dataframe: 690 Observations (0-689) of 16 variables		
Gender	chr	"b", "a", "a", "b", …
Age	chr	"30.83", "58.67", "24.50", "27.83", ...
Debt	num	0.000, 4.460, 0.500, 1.540, …
Married	chr	"u", "u", "u", "u", …
BankCustomer	chr	"g", "g", "g", "g", …
EducationLevel	chr	"w", "q", "q", "w", …
Ethnicity	chr	"v", "h", "h", "v", …
YearsEmployed	num	1.25, 3.04, 1.50, 3.75, …
PriorDefault	chr	"t", "t", "t", "t", …
Employed	chr	"t", "t", "f", "t", …
CreditScore	num	1, 6, 0, 5, …
DriversLicense	chr	"f", "f", "f", "t", …
Citizen	chr	"g", "g", "g", "g", …
ZipCode	chr	"00202", "00043", "00280", "00100", ..
Income	num	0, 560, 824, 3, …
Approved	chr	"+", "+", "+", "+", …

The programming language used for the experiment is Python 3.7.6 and the editor used is Anaconda's Jupyter Notebook.

The collected data is in raw form which needs to be transformed before it is fed into the machine. We can see that the resulting values approved are '+' or '-' each for credit granted or not respectively. These character symbols are meaningless. Converting the '+' to a '1' and the '-' to a '0' will help with building Machine Learning models later in the analysis. By inspecting the above data, we can see that there are missing values which can be filled in various ways.

Missing values for numeric data is filled using the mean imputation method wherein the mean of the respective column is entered in the place of missing value. For non-numeric data, we fill in the most frequent values present in their respective columns. Once the cleaning of data is performed, label encoding comes into the picture where the data is transformed into numeric values. This is done using LabelEncoder() which is imported from sklearn.preprocessing. It is observed that Driver's license and zip code are not essential features to consider in training the model, hence they are dropped.

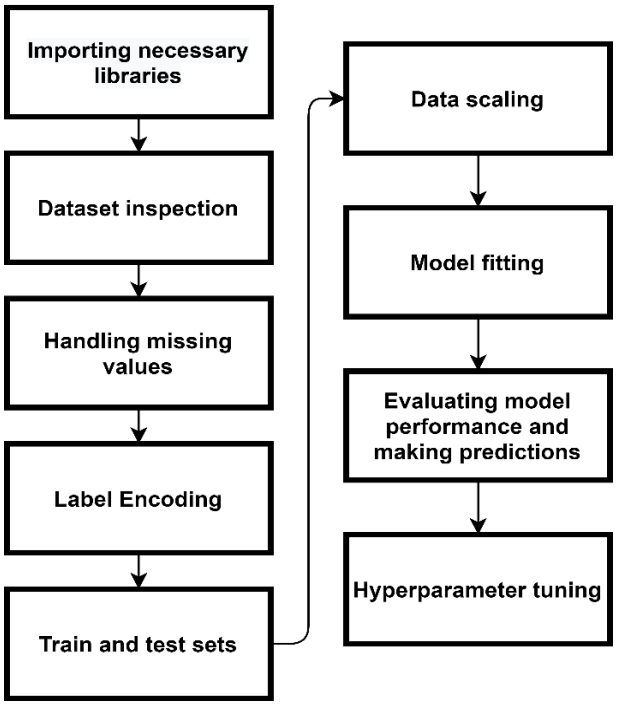

Fig -1: Experiment Flowchart

After dropping the irrelevant features and converting the data into machine language, the data is split into train and test sets by importing train_test_split from sklearn.model_selection library. As our data consists of feature variables X and y where X consists of feature columns (input variables) and y consists of the target column (output variable) with different ranges and hence, it is important to normalize the data first. The objective of normalizing the data is to change the numeric columns to a common scale without disturbing the range difference. Further, the X_train and X_test is scaled to the feature range from 0 to 1 using the MinMaxScaler [15]. This process is known as Normalization.

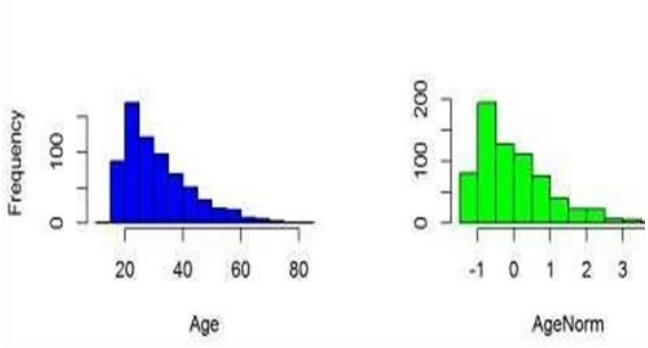

Chart -1: Distribution of values before and after normalization [38]

Further, some of the classifiers like Logistic Regression, Random Forest, Gradient Boost, XGBoost, Decision Tree, and Support Vector Machine are imported from sklearn with a random state of 24 and keras is used to import Sequential Neural Network. Once the classifiers are imported rescaledX_train and y_train are fit to the models. After model fitting, predict method with parameter rescaledX_test is used to predict the values. In order to obtain best accuracy, hyperparameter optimization is performed through GridSearchCV which is imported from the sklearn library.

Parameters which are not directly learnt within estimators are nothing but hyper parameters. It works by exhaustive search through a particular subset of hyper parameters. Candidates are exhaustively generated from a grid of parameter values specified within the param_grid argument for the grid search method. Each classifier uses different parameters as mentioned in the below table.

TABLE 2: Hyperparameters

Classifiers	Hyperparameters
Logistic Regression	tol, max_iter, solvers, penalty, c_values
Random Forest	max_depth, min_samples_leaf, min_samples_split, n_estimators
Decision Tree	min_samples_split, max_depth
Neural Network	optimizer
Support Vector	C, gamma, kernel
Gradient Boost	min_samples_split, max_depth
XGBoost	max_depth, n_estimators, learning_rate

4. MODELS USED

4.1 Logistic Regression

Our goal is to classify whether an application is approved or not, which means our target variable (Approved) should predict either 0, 1. As it is observed that our dependent variable is categorical and the value of logistic regression strictly ranges from 0 to 1, we choose Logistic Regression as one of our models to compare [3].

Logistic Regression is derived from the field of Statistics and is a technique under Machine Learning which mainly focuses on the problems with two class values [2]. To predict an output value(y), the input values(x) are combined linearly using coefficient values or weights [12].

$$y = \frac{e^{(\beta_0+\beta_1 x)}}{1 + e^{(\beta_0+\beta_1 x)}} \quad (1)$$

Where,
y is the predicted output.
β_0 is the intercept term
β_1 is the coefficient for the single input value x.
Logistic Regression uses the logistic function to transform the predictions to either 0 or 1.

4.2 Decision Tree

Decision Tree is a supervised learning method that resembles a flow chart consisting of the topmost node also referred to as Root node representing the entire training set, branches, which are used to represent the outcome of the test, and terminal node or leaf node representing decision taken. Decision Tree can be used for both numerical and categorical data. It is additionally referred to as CART (Classification And Regression Tree) [4].

The work of the decision tree is to split complex data into simpler ones by creating subtrees based on an attribute. This process of creating sub-trees is continuous until splitting no longer adds value to the predictions. Decision Tree is a powerful tool for classification because of its high accuracy, easy to build model, stability, and flexibility. There is not much computation required in Decision Tree. Decision Tree works best with simple trees and not with complex ones. It is also considered better than Logistic Regression.

4.3 Random Forest

Random Forest is a collection of Decision Tree and like Decision Tree, it can be used for both classification and regression problems. The only difference is that Decision Tree can easily over fit the data since it takes the entire dataset whereas, Random Forest selects specific rows and columns to build multiple Decision Trees. Random Forest is basically an ensemble classifier that is easy to control. One more advantage is that it is easy to specify the number of trees in a forest using n_estimators and the sample size is controlled by max_samples. Till a certain point accuracy is proportional to the number of trees but is constant after that.

Random Forest, as the name, suggests creates forest and makes it random. It takes specific features from total features, split them using best split, and repeats this until the last node is reached [8]. This process is repeated several times, hence giving us a forest.

4.4 Support Vector Machine (SVM)

Support Vector Machines a Classification algorithm which has the capability of giving the best results with a limited amount of data, at the same time is fast and dependable. SVM algorithm performs classification by finding the hyper-plane that differentiates each data item plotted as a point in n-dimensional space (where n represents number of features) and the value of each feature being the value of a particular co-ordinate [14]. The hyperplane in multi-dimensional space is constructed in an iterative manner which reduces the occurrence of error. The co-ordinates of the individual vector are termed as Support Vectors. The Classifier fits the provided data and returns the best-fit hyperplane that categorizes the data provided as input. To obtain the predicted class, some features can be fed to the classifier after the hyperplane is obtained.

4.5 Gradient Boost

Integrating various simple models into a single composite model, Machine learning defines this as “Boosting”. The simple models or weak learners are nothing but the Decision Trees which are added each at a time, to ensemble and fit to minimize the prediction errors committed by the previous models [10]. The loss function generated by the Decision Tree is basically the squared error which needs to be differentiable. As and when additional weak learners are added, the resulting model at the end becomes a strong predictor.

Gradient Boosting Regression also calculates the residual which is the difference between the predicted value and the current known target value. Further, the features of a weak model trained by Gradient Boosting Regression are mapped to the residuals. The model approaches the correct target by adding the weak model’s predicted residual to the existing model input. The overall model prediction is improved by repeating the above step again and again. Selecting a weak learner, defining the loss function, and going on minimizing the same will help Gradient Boosting Regression to approach the best prediction.

4.6 XGBoost

XGBoost or eXtreme Gradient Boosting. It is a higher version of gradient boosting. It uses decision trees because of the weak learners, and it has many

advantages over the standard gradient boosting algorithm. XGBoost has better regularization than gradient boosting. Therefore, it reduces overfitting. It allows multiprocessing, so it is much faster than standard gradient boosting. It has the inbuilt capability to handle missing data. Gradient boosting may be a greedy algorithm since it stops splitting the node as soon because it encounters a negative loss within the split whereas XGBoost splits up to the utmost depth specified and has a built-in cross-validation feature, so it is easier to work out the amount of boosting rounds at each run [9]. There are quite a few hyperparameters that require to be tuned to urge the simplest result from XGBoost algorithm.

4.7 Sequential Neural Network

Sequential Neural Network is a Neural Network with multiple layers where each layer has exactly one input tensor and one output tensor. Neural network works just like a human's brain to acknowledge patterns [29]. Similar to neurons within the human brain, neural networks are formed by interconnected neurons, also called nodes, which interact with one another through edges. In a neural network, the nodes are stacked up in layers and generally start with a broad base [28]. A neural network can be divided into three main parts: input, hidden and output layers and, these are connected through edges. The raw data is received in the first layer which is further divided into nodes to detect broad features [5]. The hidden layer(s) then analyze and process the data. Based on previous computations, the data become streamlined through the passing of each hidden layer [13]. The output is shown as the output layer. The middle layers are considered hidden layers because, like human vision, they covertly break down objects between the input and output layers.

5. CONFUSION MATRIX

Confusion matrix is a table which gives the summary of the number of correct and incorrect predictions with count values and broken down into four different combinations [27].

TABLE 3: CONFUSION MATRIX OF ALL CLASSIFIERS

Comparative techniques	TN	FP	FN	TP
Logistic Regression	87	8	30	103
Random Forest	82	13	21	112
Sequential Neural Network	84	11	29	104
Decision Tree	80	15	26	107
Support Vector Classifier	87	8	28	105
Gradient Boost	85	10	26	107
XGBoost	80	15	25	108

Where,
TN is True Negative which means the actual and the predicted value are negative (0).

FP is False Positive which means the actual value is negative (0) but the predicted value is positive (1).

FN is False Negative which means the actual value is positive (1) but the predicted value is negative (0).

TP is True Positive which means the actual and the predicted value are positive (1).

6. COMPARISON OF SUPERVISED MACHINE LEARNING MODELS ON THE BASIS OF SOME PARAMETERS

The following parameters have been considered to evaluate the performance of models.

6.1 Precision

Precision, also known as positive predicted value, is the percentage of relevant results. Precision can also be defined as the measure of positives that were correct [11].

Mathematically, Precision is the number of True Positives over the sum of True Positives and False Positives.

$$\textbf{Precision} = \textbf{TP}/(\textbf{TP} + \textbf{FP}) \quad (2)$$

Where,
TP = True Positive, and
FP = False Positive

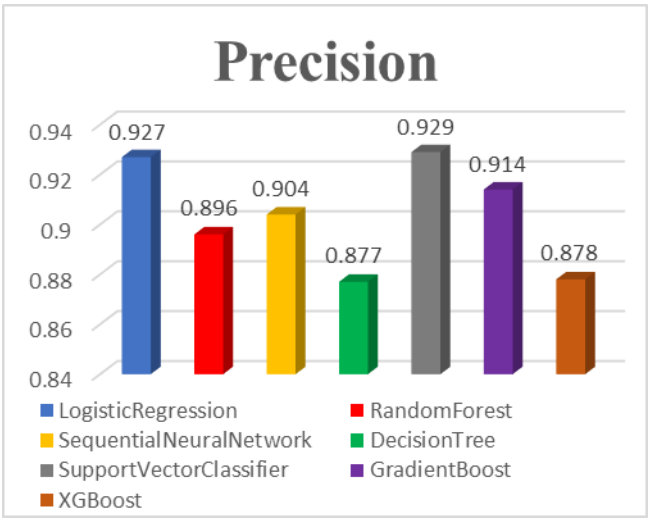

Chart -2: Comparison of Precision for all the classifiers used.

6.2 Recall

Recall, also known as Sensitivity, percentage of total relevant results correctly classified by an algorithm. Recall can also be defined as the number of correct results over the number of results that should have been returned [11].

Mathematically, Recall is the number of True Positives over the sum of True Positives and False Negatives.

$$\text{Recall} = TP/(TP + FN) \quad (3)$$

Where,
TP = True Positive, and
FP = False Positive

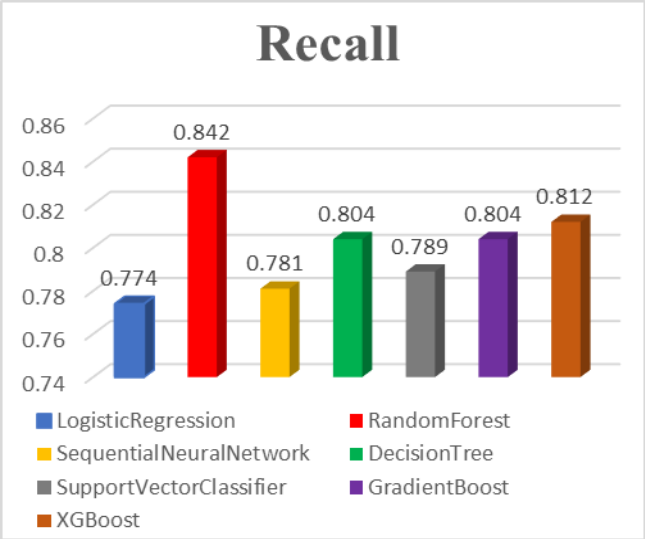

Chart -3: Comparison of Recall for all the classifiers used.

6.3 Time

Time is an important parameter for model evaluation. Time for both testing and training data is calculated for all 7 models. The execution time varies every time the code is executed and hence, the time found is approximate.

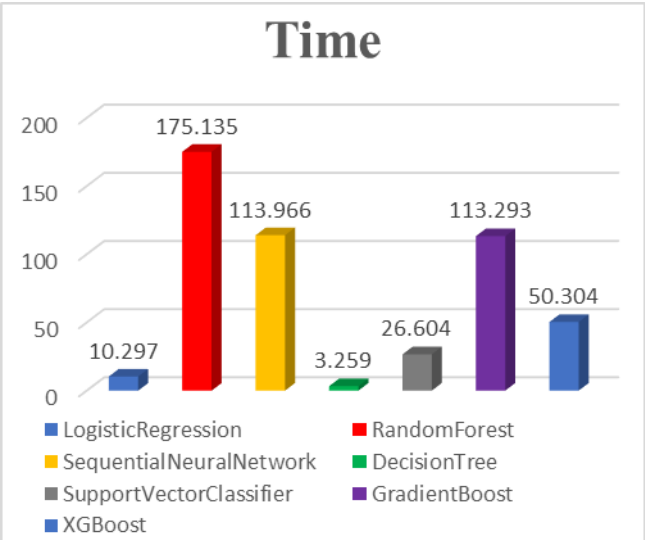

Chart -4: Comparison of Time for all the classifiers used.

6.4 F1 Score

The two important metrices for a model are Precision and Recall but it is impossible to increase the values for both at the same time [11]. To balance both of these at the same time, we have a new metric known as F_1 Score [26].

F_1 Score is the harmonic mean of Precision and Recall. Mathematically, written as

$$\text{F1} = 2 * (\text{precision} * \text{recall})/(\text{precision} + \text{recall}) \quad (4)$$

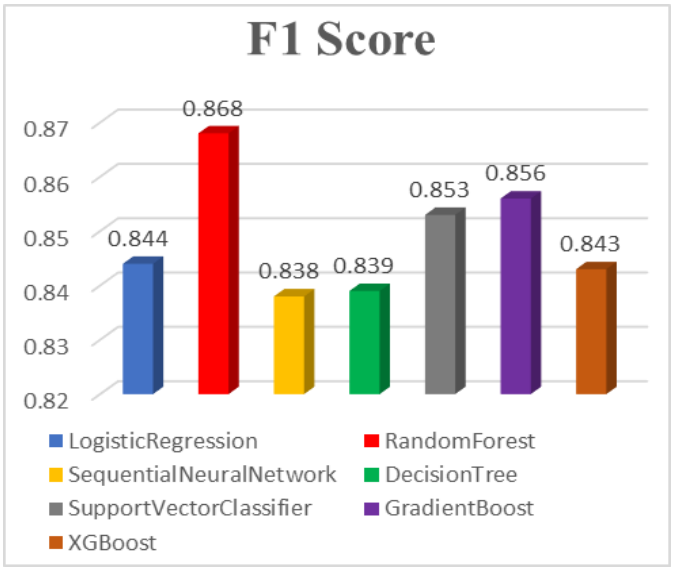

Chart -5: Comparison of F1 Score for all the classifiers used.

6.5 Best Accuracy

Accuracy is the ratio between the number of correct predictions to the total number of predictions. Mathematically, Accuracy can be calculated as

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \quad (5)$$

Where,
TP = True Positive,
TN = True Negative,
FP = False Positive, and
FN = False Negative

Increasing accuracy can fetch us more accurate predictions and can also avoid undue stress. The below graph shows the comparison of the accuracy after Hyperparameter tuning.

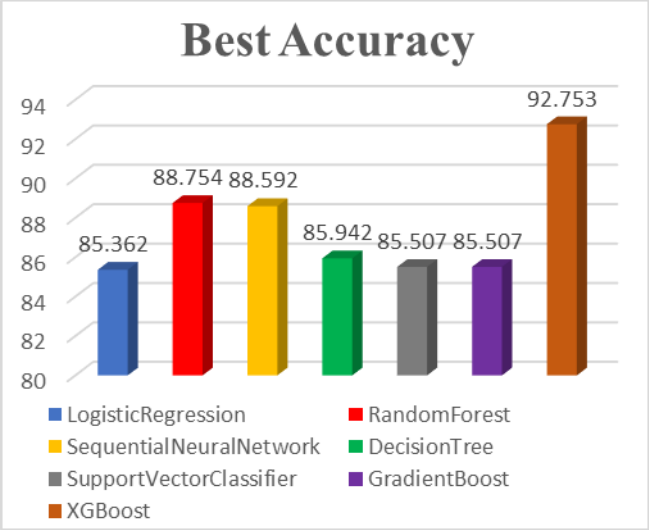

Chart -6: Comparison of best accuracy for all the classifiers used.

The below table compares all the metrics for all the classifiers.

TABLE 4: Comparison Table

MODELS	EVALUATION METRICS				
	A(%)	P	R	T(sec)	F1
Logistic Regression	85.362	0.927	0.774	10.297	0.844
Random Forest	88.754	0.896	0.842	175.135	0.868
Decision Tree	85.942	0.877	0.804	3.259	0.839
Neural Network	88.592	0.904	0.781	113.966	0.838
Support Vector	85.507	0.929	0.789	26.604	0.853
Gradient Boost	85.507	0.914	0.804	113.293	0.856
XGBoost	92.753	0.878	0.812	50.304	0.843

Where,
A = Best Accuracy,
P = Precision,
R = Recall,
T = Time, and
F1 = F1-Score

7. ROC-AUC CURVE

Performance measurement plays a vital role in Machine Learning. Every model has its own decision rule and may perform accordingly. ROC (Receiver operating characteristic) Curve is a plot which measures the performance of a binary classifier at different thresholds [26]. In a ROC Curve high value of Y-axis indicates that the value of number of True positives is greater than the number of False negatives and X-axis indicates that the value of number of False positives is greater than the number of True negatives.

Area Under Curve (AUC) is the measure of separability and ROC is the probability curve. The value of AUC lies from 0 to 1. Higher the value of AUC, better the accuracy of the model [23].

The comparison of the AUC values for the used classifiers are shown in the following plot.

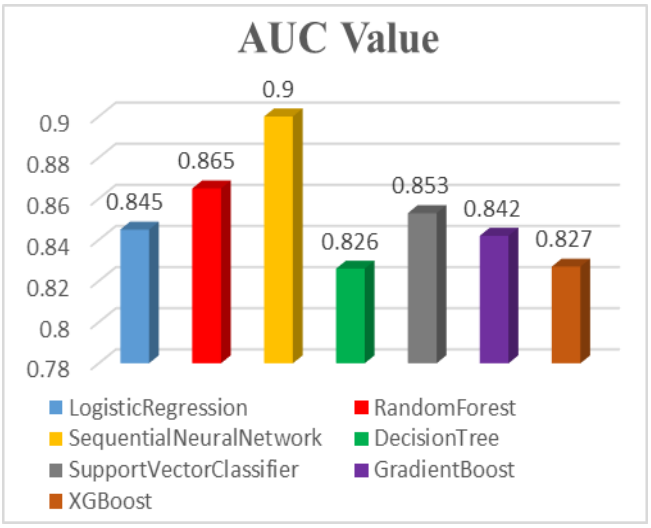

Chart -7: AUC values for all the classifiers used.

8. CONCLUSION AND FUTURE SCOPE

In this study, the dataset from the UCI Machine Learning Repository is imbalanced and therefore, considering accuracy as a parameter for comparison is not recommended. Though the accuracy of XGBoost (92.753%) is higher when compared to other models, the classifier cannot be taken as the best classifier since accuracy is not the parameter to be relied on. In place of accuracy, Precision and Recall could have been considered since these two play a vital role for the evaluation of the model. Unfortunately, it is impossible to maximize both these metrics at an equivalent time and this is where F1 Score comes into the picture to balance both Precision and Recall.

Therefore, the models are compared based on F1 Score. The classifiers for this study include Logistic Regression, Random Forest, Decision Tree, XGBoost, Gradient Boost, Support Vector Machine (SVM), Sequential Neural Network. It can be observed that Gradient Boost classifier gives a tough competition to Random Forest in both F1 Score and AUC value but fails to exceed the score of Random Forest and limits itself to be the second-best classifier.

Finally, the analysis concludes that Random Forest classifier is the best suited model for predicting Credit Card approvals based on F1 Score (0.868) with AUC value (0.865).

Future scope of this study can include the classifiers to use TensorFlow as the library to increase accuracy, precision, and the other parameters. This study uses only seven supervised Machine Learning models which can be increased and some more models including both supervised and unsupervised models can be used to find best model.

ACKNOWLEDGEMENT

The authors would like to thank Dr. P Karthikeyan, Assistant Professor, Presidency University, Bangalore for his valuable suggestions and guidance.

REFERENCES

[1] 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.

[2] A. Vaidya, "Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-6, doi: 10.1109/ICCCNT.2017.8203946.

[3] Q. Hui-lin and G. Feng, "A research on logistic regression model based corporate credit rating," 2011 International Conference on E-Business and E-Government (ICEE), Shanghai, China, 2011, pp. 1-4, doi: 10.1109/ICEBEG.2011.5882285.

[4] Yap Bee Wah and Irma Rohaiza Ibrahim, "Using data mining predictive models to classify credit card applicants," 2010 6th International Conference on Advanced Information Management and Service (IMS), Seoul, 2010, pp. 394-398.

[5] Su-Lin Pang, Yan-Ming Wang and Yuan-Huai Bai, "Credit scoring model based on neural network," Proceedings. International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, pp. 1742-1746 vol.4, doi: 10.1109/ICMLC.2002.1175335.

[6] W. Li and J. Liao, "An Empirical Study on Credit Scoring Model for Credit Card by Using Data Mining Technology," 2011 Seventh International Conference on Computational Intelligence and Security, Hainan, 2011, pp. 1279-1282, doi: 10.1109/CIS.2011.283.

[7] S. Sakprasat and M. C. Sinclair, "Classification rule mining for automatic credit approval using genetic programming," 2007 IEEE Congress on Evolutionary Computation, Singapore, 2007, pp. 548-555, doi: 10.1109/CEC.2007.4424518.

[8] 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.

[9] W. Qiu, "Credit Risk Prediction in an Imbalanced Social Lending Environment Based on XGBoost," 2019 5th International Conference on Big Data and Information Analytics (BigDIA), Kunming, China, 2019, pp. 150-156, doi: 10.1109/BigDIA.2019.8802747.

[10] 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.

[11] Shung, Koo Ping. "Accuracy, Precision, Recall or F1?" *Medium*, Towards Data Science, 10 Apr. 2020, towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9?gi=8377df893c73.

[12] Brownlee, Jason. "Logistic Regression for Machine Learning." Machine Learning Mastery, 12 Aug. 2019, machinelearningmastery.com/logistic-regression-for-machine-learning/.

[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-neural-network-python-keras/.

[14] Srivastava, Tavish SrivastavaTavish. "Support Vector Machine: SVM Classification Algorithm." *Analytics Vidhya*, 25 June 2020, www.analyticsvidhya.com/blog/2014/10/support-vector-machine-simplified/?utm_source=feed.

[15] Srinidhi, Sunny. "How to Split Your Dataset to Train and Test Datasets Using SciKit Learn." *Medium*, Medium, 9 Jan. 2020, medium.com/@contactsunny/how-to-split-your-dataset-to-train-and-test-datasets-using-scikit-learn-e7cf6eb5e0d.

[16] Lee, Elyse. "An Intro to Hyper-Parameter Optimization Using Grid Search and Random Search." *Medium*, Medium, 5 June 2019, medium.com/@cjl2fv/an-intro-

to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9834ca0a.

[17] 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.

[18] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim and A. K. Nandi, "Credit Card Fraud Detection Using AdaBoost and Majority Voting," in IEEE Access, vol. 6, pp. 14277-14284, 2018, doi: 10.1109/ACCESS.2018.2806420.

[19] E. A. E. Dawood, E. Elfakhrany and F. A. Maghraby, "Improve Profiling Bank Customer's Behavior Using Machine Learning," in IEEE Access, vol. 7, pp. 109320-109327, 2019, doi: 10.1109/ACCESS.2019.2934644.

[20] 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.

[21] Mekic, Ensar & Mekic, Emina & Ilgun, Erkan. (2014). Application of Ann in Australian Credit Card Approval. European Researcher. 69. 334-342.

[22] S. Purohit and A. Kulkarni, "Credit evaluation model of loan proposals for Indian Banks," 2011 World Congress on Information and Communication Technologies, Mumbai, 2011, pp. 868-873, doi: 10.1109/WICT.2011.6141362.

[23] Bhandari, Aniruddha. "AUC-ROC Curve in Machine Learning Clearly Explained." *Analytics Vidhya*, 16 June 2020, www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/.

[24] Song, Jong-Woo. (2008). A Comparison of Classification Methods for Credit Card Approval Using R. Journal of the Korean society for quality management. 36.

[25] Sugiyarto, I., Sudarsono, B. and Faddillah, U., 2019. Performance Comparison of Data Mining Algorithm to Predict Approval of Credit Card. *SinkrOn*, 4(1), p.149.

[26] Sun, Ting & Vasarhelyi, Miklos. (2018). Predicting credit card delinquencies: An application of deep neural networks. Intelligent Systems in Accounting, Finance and Management. 25. 10.1002/isaf.1437.

[27] Narkhede, Sarang. "Understanding Confusion Matrix." *Medium*, Towards Data Science, 29 Aug. 2019, towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.

[28] Lai, Kin Keung & Yu, Lean & Wang, Shouyang & Ligang, Zhou. (2006). Neural Network Metalearning for Credit Scoring. 403-408. 10.1007/11816157_47.

[29] Sugiyarto, Ipin & Sudarsono, Bibit & Faddillah, Umi. (2019). Performance Comparison of Data Mining Algorithm to Predict Approval of Credit Card. SinkrOn. 4. 149. 10.33395/sinkron.v4i1.10181.

[30] Chaudhuri, A., 2014. Modified fuzzy support vector machine for credit approval classification. *AI Communications*, 27(2), pp.189-211.

[31] Lu, Hongya & Wang, Haifeng & Yoon, Sang Won. (2017). Real Time Credit Card Default Classification Using Adaptive Boosting-Based Online Learning Algorithm.

[32] 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.

[33] Sariannidis, Nikolaos & Papadakis, Stelios & Garefalakis, Alexandros & Lemonakis, Christos & Kyriaki-Argyro, Tsioptsia. (2019). Default avoidance on credit card portfolios using accounting, demographical and exploratory factors: decision making based on machine learning (ML) techniques. Annals of Operations Research. 10.1007/s10479-019-03188-0.

[34] 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.

[35] Liu, R., 2018. Machine Learning Approaches to Predict Default of Credit Card Clients. *Modern Economy*, 09(11), pp.1828-1838.

[36] [1]"Credit Card Fraud Detection in Data Mining using XGBoost Classifier", *International Journal of Recent Technology and Engineering*, vol. 9, no. 1, pp. 603-608, 2020. Available: 10.35940/ijrte.f8182.059120.

[37] 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.

[38] D.Jayanthi, "CREDIT APPROVAL DATA ANALYSIS USING CLASSIFICATION AND REGRESSION MODELS", IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P- ISSN 2349-5138, Volume.5, Issue 3, Page No pp.162-169, September 2018, Available at : http://www.ijrar.org/IJRAR190B030.pdf

BIOGRAPHIES

Siddhi Bansal was born in Pali, Rajasthan, India, in 1999. She is currently pursuing B. Tech degree in Computer Engineering and Data Science from Presidency University, Bangalore, India. Her research interests include Computer Vision, Internet of Things (IOT), Machine Learning, and Cryptography.

Tushar Punjabi was born in Jodhpur, Rajasthan, India, in 2000. He is currently pursuing B. Tech degree in Computer Engineering and Data Science from Presidency University, Bangalore, India. His research interests include Data Science, Cryptography, Cyber Security, and Cloud Computing.