Credit Card Approval Prediction

Jay Patel (824672518), Nadeem Masani (824660441), Harsh Javia (824649625)

Abstract*: *In the banking sector, every banking infrastructure contains an enormous dataset for customers' credit card approval which requires customer profiling. The customer profiling means collection of data related to what customers need. It depends on customers' basic information like field of work, address proof, credit score, salary details, etc. This process mainly concentrates on predicting approval of credit cards to customers using machine learning. Machine Learning is the scientific study of algorithms and statistical models that computers use to perform specific tasks without any external instructions or interference. In the current trend this process is possible using many algorithms like "K-Mean, Improved K-Mean and Fuzzy C-Means". This helps banks to have a high profitability. The proposed system aims at improvising the accuracy ratio while using only few algorithms.

Keywords: Credit Score, Machine Learning, Supervised Learning, Unsupervised Learning.

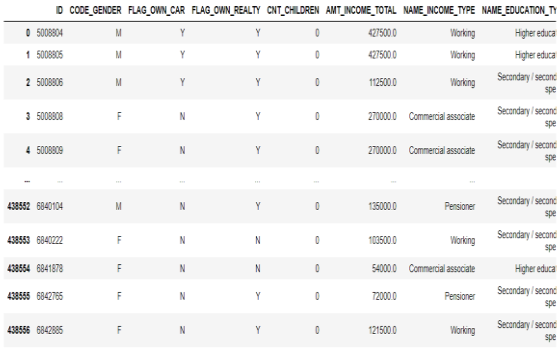
I. INTRODUCTION

The decision of approving a credit card or loan is majorly dependent on the personal and financial background of the applicant. Precisely, age, gender, income, employment status, credit history and other attributes contributes to the approval decision. Credit Analysis involves the statistical – quantitative and qualitative measure to investigate the probability of a third party to pay back the loan to the bank on time and predict its default characteristic. Analysis focus on recognizing, assessing and reducing the financial/other risks involved which may otherwise results in the losses incurred by the company while lending. The risk can be business loss by not approving the good candidate or can be financial loss by approving the candidate who is at bad risk. It is very important to manage credit risk and handle challenges efficiently for credit decision as it can have adverse effects on credit management. Therefore, evaluation of credit approval is significant before jumping to any granting decision.

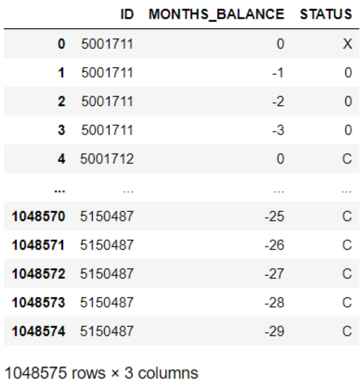
The primary objective of this analysis is to implement the data mining techniques on credit card approval dataset and prepare models for prediction of approval decision using machine learning models. Therefore, the risks can be identified while lending, appropriate conclusions can be elicited about probability of repayment and recommendations can be put forward. The techniques used for analysis are data visualization to get better insight of data, data and dimension reduction to locate essential attributes, supervised and unsupervised learning for preparing models. The classification algorithms we used are Logistic Regression, Decision Tree, Random Forest Classification, Support Vector Machine, and KNN to examine the prediction accuracy. These models are evaluated and compared using confusion matrix, Area Under the Curve performance metrics.

II. DATASET

The dataset analyzed in this report is the Credit Card Approval dataset taken from Kaggle. In its initial, unaltered form, the dataset contains more than 400K cases (some repetitive cases), and 17 attributes. We don't have the outcome column which is binary classified, as if the credit card application for the given individual should be approved or not. But, we do have data about each individual's credit history in second csv file. We will classify individual as 'good' or 'bad' clients based on their credit history for our output(result) column. We will not consider credit history as part of feature to predict if the client is 'good' or 'bad'. The first look at the structure of the dataset showed as following:



	ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	NAME_INCOME_TYPE	NAME_EDUCATION_TY
0	5008804	M	Y	Y	0	427500.0	Working	Higher educat
1	5008805	M	Y	Y	0	427500.0	Working	Higher educat
2	5008806	M	Y	Y	0	112500.0	Working	Secondary / second spe
3	5008808	F	N	Y	0	270000.0	Commercial associate	Secondary / second spe
4	5008809	F	N	Y	0	270000.0	Commercial associate	Secondary / second spe
...	...	...	...	...	...	...	...	
438552	6840104	M	N	Y	0	135000.0	Pensioner	Secondary / second spe
438553	6840222	F	N	N	0	103500.0	Working	Secondary / second spe
438554	6841878	F	N	N	0	54000.0	Commercial associate	Higher educat
438555	6842765	F	N	Y	0	72000.0	Pensioner	Secondary / second spe
438556	6842885	F	N	Y	0	121500.0	Working	Secondary / second spe

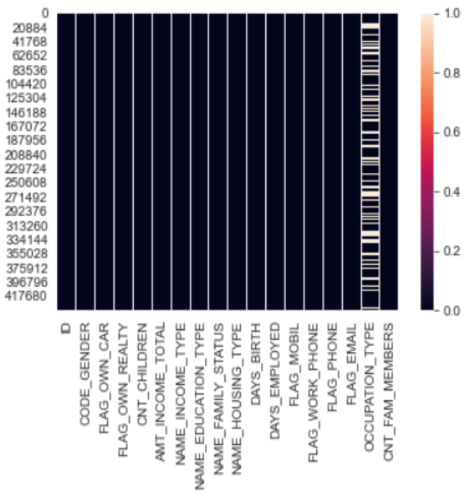


	ID	MONTHS_BALANCE	STATUS
0	5001711	0	X
1	5001711	-1	0
2	5001711	-2	0
3	5001711	-3	0
4	5001712	0	C
...	...	...	...
1048570	5150487	-25	C
1048571	5150487	-26	C
1048572	5150487	-27	C
1048573	5150487	-28	C
1048574	5150487	-29	C

1048575 rows × 3 columns

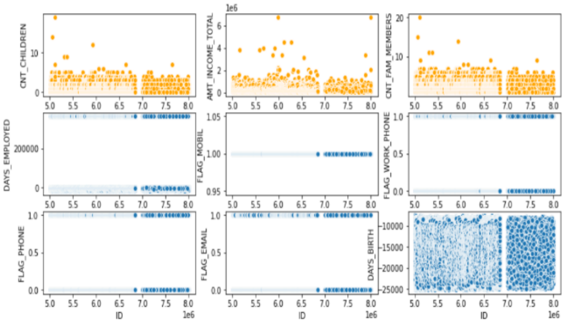
III. ANALYSIS

As the dataset is very large, there is high probability of missing data. As we can see from below heatmap image that the occupation_type has many NULL values, so will discard that attribute from our dataset.



Also, there is a probability of having duplicate instances, to overcome this issue we will drop duplicates using ID as reference variable for an instance.

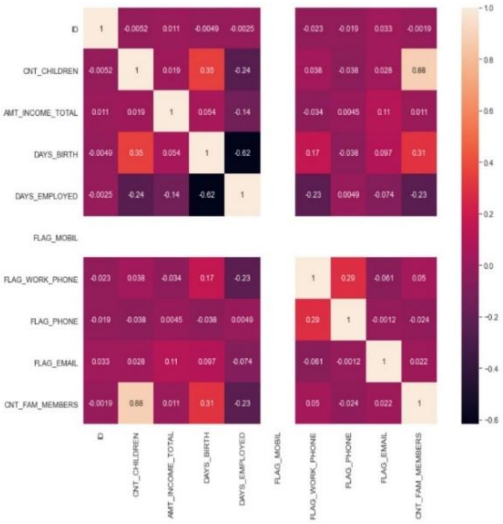
We will filter the columns that have non-numeric values, to convert them to numeric values. We counted total values of classification of each class and found that each feature is well classified as they have similar number of instances. We have used LabelEncoder method available in sklearn module to convert those non-numeric values to numeric values. LabelEncoder encode labels with a value between 0 and n_classes-1 where n is the number of distinct labels. If a label repeats it assigns the same value to as assigned earlier.



From the above graph, we can say that features cnt_children, amt_income_total, cnt_fam_members have outliers.

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. We need to remove these outliers to make sure they do not affect our model results. We have used quantile function available in pandas to remove these outliers.

The attributes with continuous values are each on same scale with low variances among them. This expedites the analysis process as the results will not deviate and comparison for finding correlation can be done among variables. We will plot the correlation between the features to check if we have considered correct features for prediction or not. By plotting the correlation, we found out that the has_mobil feature is not related to any other feature as its value is always same irrespective of the instances.



To create the output column for ‘good’ or ‘bad’ clients, we classified ‘good’ clients as those who have 60 or less days overdue on their payment and all others as ‘bad’. The total number of classifications as ‘good’ are much higher than ‘bad’. 99% of data have 0(‘good’) value and only 1% are 1(‘bad’) which results in oversampling issue. The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on the minority class, although typically it is the performance on the minority class that is most important because the minority class represents ‘bad’ clients.

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don’t add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique, or SMOTE.
The approach is effective because new synthetic examples from the minority class are created that are plausible, that is, are relatively close in feature space to existing examples from the minority class.

IV. MACHINE LEARNING MODELS

A. Logistic Regression

Regression models are useful for predicting continuous (numeric) variables. However, the target value in Approved is binary and can only be values of 1 or 0. The applicant can either be issued a credit card or denied- they cannot receive a partial credit card. We could use linear regression to predict the approval decision using threshold and anything below assigned to 0 and anything above is assigned to 1. Unfortunately, the predicted values could be well outside of the 0 to 1 expected range. Therefore, linear or multivariate regression will not be effective for predicting the values. Instead, logistic regression will be more useful because it will produce probability that the target value is 1. Probabilities are always between 0 and 1 so the output will more closely match the target value range than linear regression.

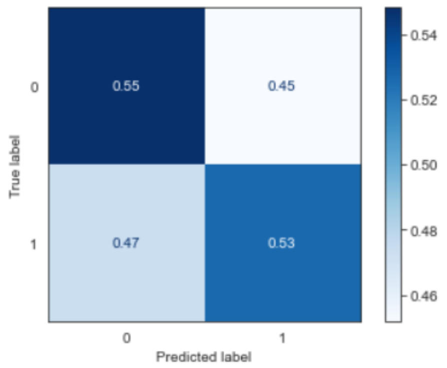
Input values (x) are combined linearly using weights to predict an output value (y). Below is an example logistic regression equation:

$$\log(p/1-p)=\beta_0+\beta_1 x_1+\cdots+\beta_q x_q$$

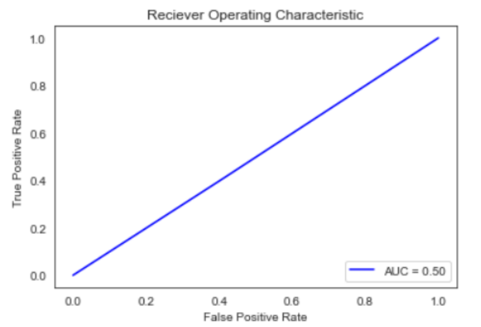
The parameter C in Logistic Regression is our regularization parameter, where C = 1/ λ. Lambda (λ) controls the trade-off between allowing the model to increase it's complexity as much as it wants with trying to keep it simple.

Parameter C will work the other way around. For small values of C, we increase the regularization strength which will create simple models which underfit the data. For big values of C, we low the power of regularization which implies the model is allowed to increase it's complexity, and therefore, overfit the data.

We tried different values of C like [1, 0.1, 0.01, 100]. The maximum accuracy we get on training data was of 65% with the value C=0.4. We were able to get around 53% of accuracy on testing data, and the result of confusion matrix of this model were as follows:



Area Under the curve/ROC is another performance metrics used to evaluate the trade off between sensitivity (True Positive Rate) and Specificity (False Positive Rate). Higher the AUC, better is the model. Area Under the curve for our model is as follows:



B. Decision Tree

Decision Trees can be used for both classification and regression. The methodologies are a bit different, though principles are the same. The decision trees uses the CART algorithm (Classification and Regression Trees). In both cases, decisions are based on conditions on any of the features. The internal nodes represent the conditions and the leaf nodes represent the decision based on the conditions.

In our dataset, attributes are first converted into word vectors with its importance. It creates a tree and calculates the cost for all path from root to leaf. The split with lowest cost will be chosen. This algorithm is recursive in nature as the groups formed can be sub-divided using same strategy. This algorithm is also known as greedy algorithm.

Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.

The formula for finding Gini or Gini Index is as follows:

$$Gini = 1 - \sum_{j} p_j^2$$

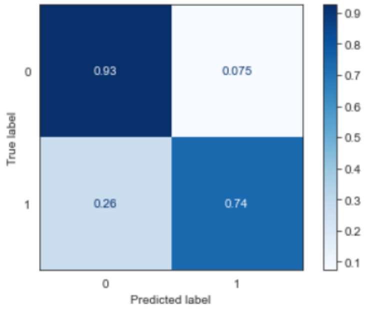
As a problem usually has a large set of features, it results in large number of split, which in turn gives a huge tree. Such trees are complex and can lead to over fitting. So, we need to know when to stop? One way of doing this is to set a minimum number of training inputs to use on each leaf. Another way is to set maximum depth of your model. Maximum depth refers to the length of the longest path from a root to a leaf.

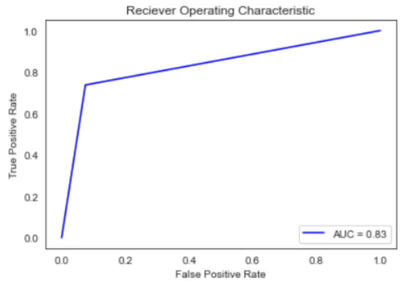
$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

The performance of a tree can be further increased by pruning. It involves removing the branches that make use of features having low importance. This way, we reduce the complexity of tree, and thus increasing its predictive power by reducing over fitting. Pruning can start at either root or the leaves. The simplest method of pruning starts at leaves and removes each node with most popular class in that leaf, this change is kept if it doesn’t deteriorate accuracy. Its also called reduced error pruning. More sophisticated pruning methods can be used such as cost complexity pruning where a learning parameter (alpha) is used to weigh whether nodes can be removed based on the size of the sub-tree. This is also known as weakest link pruning.

In our project, We have used grid search to apply different set of hyperparameters. The grid search will use all combinations of hyperparameters and chooses best for the validation and testing data. We have done hyperparameter tuning to get the optimal criterion and max depth.

We have achieved 83% of accuracy for prediction. Confusion Matrix and Area Under the curve for our model is as follows:



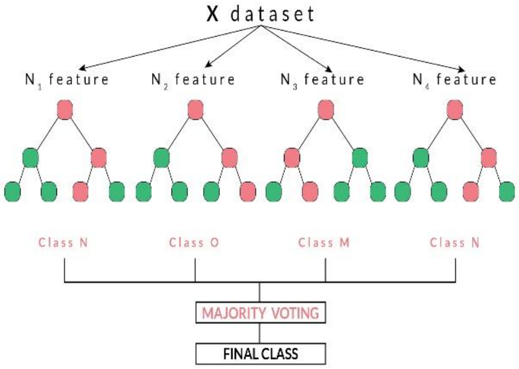


C. Random Forest

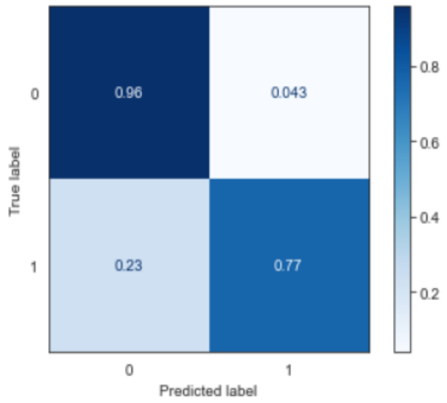
Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

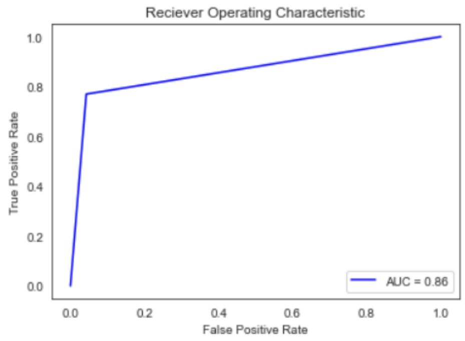
The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.



While applying this algorithm, We have used grid search to apply different set of hyperparameters. The grid search will use all combinations of hyperparameters and chooses best for the validation and testing data. We have also used hyperparameter tuning to get the best result. There is the n_estimators hyperparameter, which is just the number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions. We tested it on test set and result was almost similar to decision tree. We have achieved 85% accuracy with random forest. Confusion matrix and Area under curve for this model was as follows:



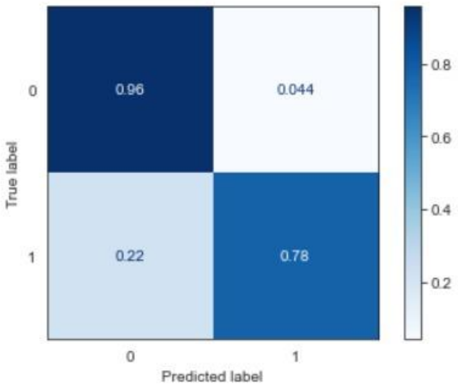


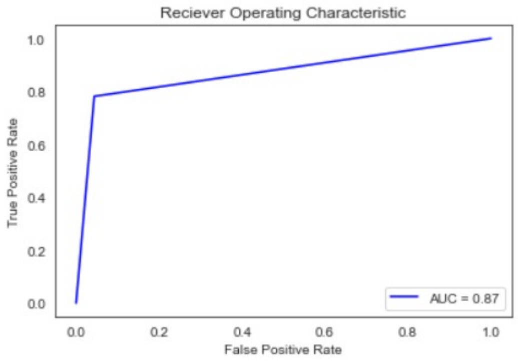
D. Support Vector Machine

We decided to pick SVM classifier as one of our classifier since determining the boundary that distinguishes the 'good' clients from the 'bad' ones is the fundamental problem posed by this dataset. SVM's are perfect for creating a hyperplane between the two categories.

C parameter adds a penalty for each misclassified data point. If c is small, the penalty for misclassified points is low so a decision boundary with a large margin is chosen at the expense of a greater number of misclassifications. If c is large, SVM tries to minimize the number of misclassified examples due to high penalty which results in a decision boundary with a smaller margin. Penalty is not same for all misclassified examples. It is directly proportional to the distance to decision boundary.

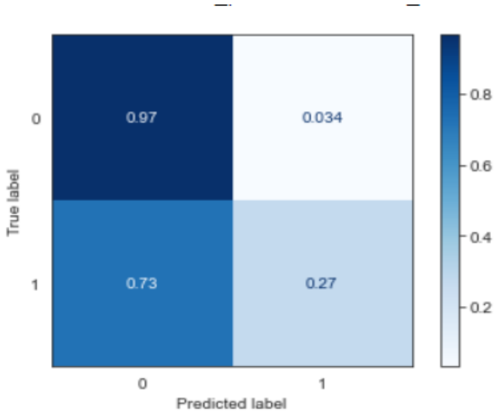
After lot of tweaking and tuning for gamma and C, we found that setting gamma as 0.01 and C as 100 provided a better model. The accuracy we get from SVM model was 86%, which is best we got. Confusion Matrix and Area Under curve for the SVM is as follow:

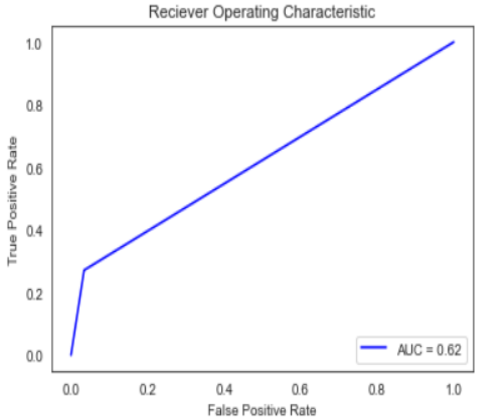




E. K-nearest neighbors

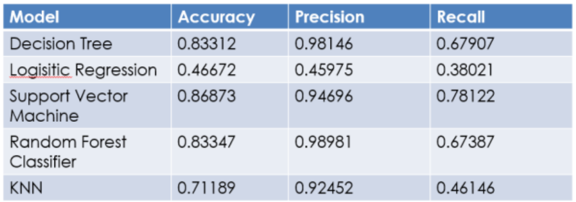
k-NN algorithm predicts the values of test dataset from the training datasets following distance (Euclidean or Hamming) similarity measure. k-NN is a non-parametric classification method which does not make assumptions about the distribution of the data and is distribution independent. This classifier when given with unknown test case finds pattern from training cases and these cases acts as neighbors to test cases. The similar instances of prediction attribute to unseen test cases are summarized and acts as prediction for the unseen cases. It is also referred as lazy algorithm. The selection of the value of k for the algorithm is important and can be done experimentally by starting with k=1. Smaller value will have noise and higher value will be computationally expensive. The value of k with least error rate is optimal and can be found by incrementing it or using cross validation. After applying different n_neighbors values to our model, we found the highest accuracy with K = 7. The Confusion Matrix and Area Under Curve for the model is as follows:





V. MODEL EVALUATION RESULT

The following table shows the results(accuracy) we get from different machine learning algorithms that we applied on our dataset.



Model	Accuracy	Precision	Recall
Decision Tree	0.83312	0.98146	0.67907
Logisitic Regression	0.46672	0.45975	0.38021
Support Vector Machine	0.86873	0.94696	0.78122
Random Forest Classifier	0.83347	0.98981	0.67387
KNN	0.71189	0.92452	0.46146

VI. CONCLUSION

From our analysis, we are able to conclude that the most significant attributes in determining the outcome of a credit application are PayScale, Days of Employment, and Education Level. This result was consistent over all the techniques applied – visualization, dimension and data reduction, supervised and unsupervised learning. Correspondingly, the other variables in the dataset – Age, Gender, own_car, own_realty, no_children, Income_type, Family Status, House_type, has_phone, has_mobile, has_email, member_count– did not have a significant effect on whether an application was approved. Of the models built to predict credit application outcomes, SVM and Random Forest provided the most accurate results and had a misclassification rate of only 14%. Future work on this and similar datasets could include combination of two or more techniques to produce a classification model with a higher degree of accuracy. An improved model can

greatly reduce the risk of granting credit to potential defaulters.

REFERENCES

[1] Song, Jungwoo. "A comparison of Classification Methods For Credit Card Approval Using R"
[2] Sudhamathy G: Credit Risk Analysis and Prediction Modelling of Bank Loans Using R, vol. 8, no-5, pp. 1954-1966.
[3] Fu,Zhoutong, and Liu Zhedi. "Classifier Comparisons On Credit Approval Prediction".
[4] Aida Krichene, Abdelmoula."Bank credit risk analysis with k-nearestneighbor classifier: Case of Tunisian banks"

[5] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.
[6] Huang, J., Wang, H., Wang, W., & Xiong, Z. (2013). A Computational Study for Feature Selection on Customer Credit Evaluation. 2013 IEEE International Conference on Systems, Man, and Cybernetics.

STATEMENT OF CONTRIBUTION

Jay Patel: Responsible for preprocessing and analyzing data and training and testing Random Forest Model.
RedID: 824672518

Nadeem Masani: Responsible for preprocessing data and training, testing and evaluating Support Vector Machine and Decision Tree model.
RedID: 824660441

Harsh Javia: Responsible for preprocessing data and training, testing, and evaluating Logistic Regression and K-Nearest Neighbors.
RedID: 824649625