**REPORT ON**

**MACHINE LEARNING PROJECT:**

**LOAN APPROVAL PREDICTION**

Submitted by **GROUP 5**

|  |  |
| --- | --- |
| **NAME OF THE STUDENT** | **ROLL NUMBER** |
| P ABHIRAM REDDY | AM.EN.U4AIE22162 |
| SANVIA SAJ PS | AM.EN.U4AIE22163 |
| MADHAV A DAS | AM.EN.U4AIE22131 |
| M T SURYA NARAYAN | AM.EN.U4AIE22130 |

*Department of Computer Science*

*Amrita School of Engineering   
Amrita Vishwa Vidyapeetham,*

*Amritapuri, Kollam, India*

Submitted to:

Sarath S, Assistant Professor (Sr. Gr.) CSE,

[saraths@am.amrita.edu](mailto:saraths@am.amrita.edu),

9895893606.

**ABSTRACT**

In this machine learning project, we were asked to experiment with various machine learning models like classification models, and regression models based on the topic we chose. The idea behind this project is to make students familiar with machine learning models to gain experience working with those models and understand how each model works. Working on this project helps gain major knowledge regarding data preprocessing. Once the project is done you will have a clear idea about various data preprocessing techniques and you will also be in a situation where you can wisely choose appropriate techniques.

We had to choose three datasets under one particular topic and perform data preprocessing for those three datasets and since this is a team project with four members in each team, we were asked to implement 4 models under each preprocessed dataset. We will then have to do a good analysis of the results and state what we have understood through the analysis done.

**INTRODUCTION**

We do know that machine learning is a subset of Artificial Intelligence and machine learning is nothing but a process where we provide intelligence and impose decision-making capability to the systems. The ultimate aim is to make the machine learning model learn the hidden patterns behind the data and target variables. The useful patterns that can influence or impact the decision-making capacity in the machine learning model can be said as ‘signals’. The unwanted patterns can be called with the term ‘noise’.

In my project, I chose to do a classification problem and that is the reason I chose Loan Approval Prediction. The motivation behind choosing this topic is because of a problem statement: these days loans have become common for individuals whether it be either for education or business. When we apply for a loan, it takes 3 or more days to get approval for it, and most of us are uncertain whether it will be approved or not. This loan approval prediction model can be integrated with the available loan processing system so that when a loan application is sent, it automatically takes the data into the model and be able to predict the approval status. Using this, applicants can receive faster responses about their loan status. It will also help in saving time by reducing the need for manual work.

Loan Approval Prediction is one of the major topics that comes to our mind instantly when we think about classification problems. It is a supervised technique which means we will have the target label. We will give the model both the data and the corresponding label which helps the model gain the capability to predict well by learning the underlying patterns in such a way that the data maps to the corresponding label.

We obtained three datasets for loan approval prediction in kaggle.com and each of them was in different scales. The four models that we chose to implement are Logistic Regression, Random Forest, Support Vector Classifier (SVC) and K Nearest Neighbors (KNN). If we look at our dataset we sure have different kinds of features in each dataset, from this, we have understood that each dataset should be handled in different ways when it comes to data preprocessing. We even implemented the Forward Selection code without the use of its built-in function.

In real life when in a situation to decide whether to approve or reject an application loan, features like cibil score or credit history play a major role and we will understand it sooner when looking at the accuracies of the models implemented in one of our datasets and also when looked at the result of the forward selection.

Now let’s know more about our features because then we can understand if we have any unnecessary features that might not be needed to predict the loan approval status.

**Dataset 1:**

Source link: [Kaggle dataset - 1](https://www.kaggle.com/datasets/ninzaami/loan-predication)

(https://www.kaggle.com/datasets/ninzaami/loan-predication)

Dataset description: The dataset contains information relevant to loan approval

Features and their description:

**Loan\_ID**: It’s a unique id value for each loan application

**Gender**: The gender of the applicant

**Married**: Marital status of the applicant

**Dependents**: Number of individuals financially dependent on the applicant

**Education**: It’s a feature informing whether the applicant is a graduate or not

**Self\_Employed**: Feature that informs whether the applicant is self-employed or not

**ApplicantIncome**: Income of the applicant

**CoApplicantIncome**: Income of an individual who applies for a loan jointly with the primary applicant

**LoanAmount**: Amount of the loan requested by the applicant

**Loan\_Amount\_Term**: The length of time over which a loan is scheduled to be repaid

**Credit\_History**: Refers to a record of an individual's or entity's borrowing and repayment behaviour.

**Proprty\_Area**: Type of area where a property is situated (urban or rural)

**Loan\_Status**: This is our TARGET VARIABLE informing whether the loan is approved or not.

**Dataset 2:**

Source link: [Kaggle dataset - 2](https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset)

(https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset)

Features and their description:

**loan\_id**: It’s a unique id values for each loan application

**no\_of\_dependents**: Number of individuals financially dependent on the applicant

**education**: It’s a feature informing whether the applicant is a graduate or not

**self\_employed**: Feature that informs whether the applicant is self-employed or not

**income\_annum**: Income of the applicant

**loan\_amount**: Amount of the loan requested by the applicant

**loan\_term**: The length of time over which a loan is scheduled to be repaid

**cibil\_score**: Refers to a credit score generated by the Credit Information Bureau Limited

**residential\_assets\_value:** Estimated value of residential properties owned by the applicant

**commercial\_assets\_value:** Estimated value of commercial properties owned by the applicant

**luxury\_assets\_value:** Estimated value of luxurious assets properties owned by the applicant

**bank\_asset\_value:** Refers to the total value of assets held by a bank

**loan\_status:** This is our TARGET VARIABLE informing whether the loan is approved or not.

**Dataset 3:**

Source link: [Kaggle dataset - 3](https://www.kaggle.com/datasets/chilledwanker/loan-approval-prediction)

(https://www.kaggle.com/datasets/chilledwanker/loan-approval-prediction)

Features and their description:

**person\_age:** Age of the applicant

**person\_income:** Income of the applicant

**person\_home\_ownership:** Refers to the homeownership status of the applicant

**person\_emp\_length:** Refers to the length of time an individual has been employed

**loan\_intent:** Refers to the purpose or intention behind the loan application.

**loan\_grade:** Refers to the grade which indicates the risk level of the loan

**loan\_amnt:** Amount of the loan requested by the applicant

**loan\_int\_rate:** Refers to the interest rate associated with the loan

**loan\_status:** This is our TARGET VARIABLE informing whether the loan is approved or not.

**loan\_percent\_income:** represents the ratio of the loan amount applied by the applicant to the applicant’s income (in %)

**cb\_person\_default\_on\_file:** Indicates whether the person has a record of defaulting on a loan according to their credit bureau file

**cb\_person\_cred\_hist\_length**: Refers to the length of the credit history for the individual, as recorded in their credit bureau file

We assume that the datasets used for training our model represents the real-world data and nobody messed around with the data randomly. We also assume that the loan applications are independent of each other, which means that the approval or denial of an application doesn’t show any impact on other applications. Finally the assumption is that the criteria followed by the lenders to approve or deny a loan remain constant and don’t change over time.

**METHODOLOGY**

Regarding the methodology, let’s look at the code in detail and understand the reasons behind the decisions made during data preprocessing. We do know that in data preprocessing we need to find the missing values and fill them with suitable, appropriate imputation techniques after that we need to remove outliers and finally convert categorical values into numerical values.

**IMPUTATIONS:**

In **Dataset 1** we have 614 records and 13 unique features. 7 features out of 13 had missing values, so there is a requirement to find a suitable imputation technique for those 7 features. The features names are Gender, Married, Dependents, Self\_Employed, LoanAmount, Loan\_Amount\_Term, Credit\_History.

|  |  |
| --- | --- |
| **Feature Name** | **Number of missing values** |
| Gender | 13 |
| Married | 3 |
| Dependents | 15 |
| Self\_Employed | 32 |
| LoanAmount | 22 |
| Loan\_Amount\_Term | 14 |
| Credit\_History | 50 |

Here is the table with the feature name and its corresponding number of missing values in dataset 1.

There is nothing complex here, we can easily understand that the missing values of categorical values can be filled with mode since we can’t apply mean or median imputations to them. Loan\_Amount\_term and Credit\_History also can be imputed using mode since there aren’t many unique values in these features. Coming to the Loan\_Amount, there is no way that it could be filled using mode imputation since there will be many unique values and each applicant wants to take a different amount of loan. So we can either use mean or median imputation but, I chose median imputation because if we select mean imputation there is a chance that outliers impact the mean of the data, No one wants outliers to be considered or let them influence the imputation result. so, it is better to choose median imputation.

In **Dataset 2** there are 4269 records and 13 features. None of them had any missing values. So, there are no imputations performed on this dataset.

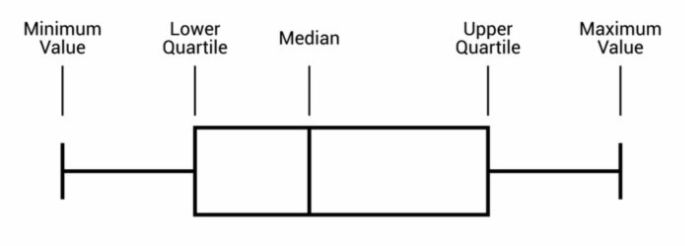
In **Dataset 3**, there are 32581 records and 12 features out of which 2 features had missing values. The feature names are person\_emp\_length, loan\_int\_rate.

|  |  |
| --- | --- |
| **Feature Name** | **Number of missing values** |
| person\_emp\_length | 895 |
| loan\_int\_rate | 3116 |

Here is the table with the feature name and its corresponding number of missing values in dataset 3. Both of these features are filled using median imputation for the same reason as LoanAmount in dataset 1. When looking into our dataset 1, it is clearly noticed that person\_emp\_length surely has outliers because the value in the first record under the feature person\_emp\_length is surely in another scale compared to the later values in the further records. So to avoid letting the outliers influence the imputation value, Median imputation is chosen.

**REMOVING OUTLIERS:**

Once the imputations are done, we will now remove outliers. How do we do this? The answer would be using a Box plot and IQR method. The box plot is a way of visualizing the values in a feature and seeing if that feature contains any outliers or not. We can remove outliers using the IQR method. Outliers are the data points that are below the lower limit and above the upper limit.



* Minimum: It is the minimum value in the dataset excluding the outliers
* First Quartile or Lower Quartile (QI) - 25% of the data lies below the First (lower) Quartile.
* Median (Q2): It is the mid-point of the dataset. Half of the values lie below it and half
* Third Quartile or Upper Quartile (Q3): 75% of the data lies below the Third (Upper) Quartile.
* Maximum: It is the maximum value in the dataset excluding the outliers.

After plotting the box plots for the numerical features like ApplicantIncome, CoapplicantIncome etc., we will then decide whether we need to remove the outliers of this feature or not. How do we decide that? Let’s look at two examples.

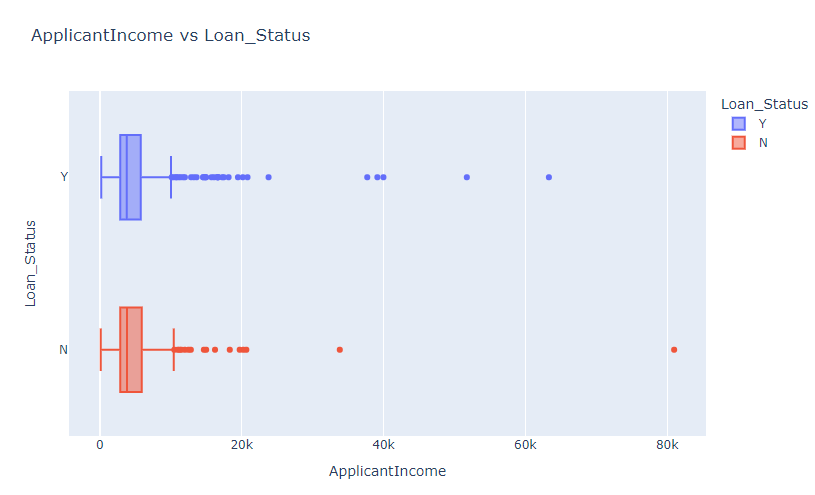


Fig 1

From Fig 1, we could understand that the median of the data would be around 4K and there are points even in ranges 50k, 60k and 80k. So these must be considered as outliers and those records have to be removed.

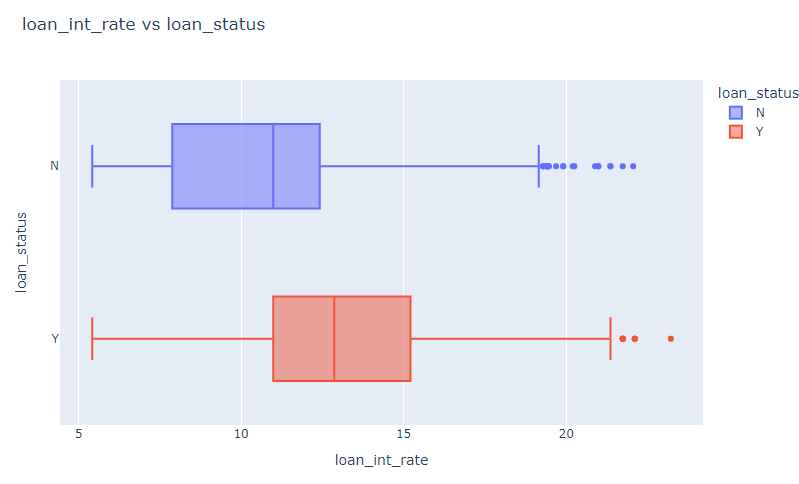


Fig2

In Fig 2, even though there are few outliers they are just outside the upper limit of the box plot. There is no compulsion that these points have to be considered as outliers and removed from the data. Similarly, we do it for all the numerical features and decide whether to remove the outliers or not.

Since we are removing the records with outliers there will surely be a change in the shape of the data.

Let's look at how removing outliers affected the shape of our datasets.

|  |  |  |
| --- | --- | --- |
| **DATASET** | **Number of records before removing outliers** | **Number of records after removing outliers** |
| 1 | 614 | 520 |
| 2 | 4269 | 4173 |
| 3 | 32581 | 28287 |

**ENCODING:**

Encoding techniques are the techniques used to convert categorical values in a feature into numerical values. This can be done in a lot of ways like, label encoding, one-hot encoding, ordinal encoding, mean/target encoding, frequency encoding etc., however in this project I have only used label encoding and one-hot encoding based on the type of the feature.

In **Dataset 1** we have 7 categorical features including the target variable ‘Loan\_Status’.

They are ‘Gender’,’Married’,’Dependents’,’Education’,’Self\_Employed’,’Property\_Area’,

’Loan\_Status’. Here is a table specifying the encoding technique used for the corresponding features.

|  |  |
| --- | --- |
| **Feature Names** | **Encoding Technique** |
| Gender’  ’Married’  ’Education’  ’Self\_Employed’  ’Property\_Area’ | One-hot encoding |
| ’Dependents’  ’Loan\_Status’. | Label Encoding |

The reason for choosing label encoding for ‘Loan\_status’ is just that the target variable should be having positive class(1) and negative class(0). So, to maintain that notation that feature is encoded with label encoding. Coming to the ‘Dependents’ feature, if we look into the unique values of that feature then we could notice it has ‘0’,’1’,’2’, and ’3+’. Which means this feature has become categorical because it has the string value ‘3+’. If you look mathematically then you will understand that the feature is meant to have ordinality so, as a label encoder can impose ordinality, we need to encode the ‘Dependents’ feature with a label encoder.

The Remaining features are encoded using one-hot encoding because those features are not meant to have ordinality and one-hot encoding does not impose ordinality. One-hot encoding is even efficient and simple with features having a less unique values.

In **Dataset 2** there were only 3 categorical features. They are ‘education’,’self\_employed’, and ‘loan\_status’. These features were present even in the dataset 1 so, without thinking much the same technique used for those features in previous dataset is used here.

|  |  |
| --- | --- |
| **Feature Names** | **Encoding Technique** |
| ’education’  ’self\_employed’ | One-hot encoding |
| ’loan\_status’. | Label Encoding |

In **Dataset 3** there are 5 categorical features and those are ‘person\_home\_ownership’, ‘loan\_intent’,‘loan\_grade’,’loan\_status’, and ’cb\_person\_default\_on\_file’. Here is a table specifying the encoding technique used for the corresponding features.

|  |  |
| --- | --- |
| **Feature Names** | **Encoding Technique** |
| ‘loan\_intent’  ‘person\_home\_ownership’ | One-hot encoding |
| ’loan\_grade’  ’loan\_status’  ‘cb\_person\_default\_on\_file’ | Label Encoding |

This time there was a need for ordinality in ‘loan\_grade’ as they are the grades but in alphabetical order ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, and ’G’. So, ‘loan\_grade’ is encoded with label encoding. ‘cb\_person\_default\_on\_file’ has only two unique value so, if we use label encoding it will just become a binary feature. The other features are encoded using one-hot encoding since there are more than two unique values in those features. So, if we use label encoding it might impose ordinality which is unnecessary.

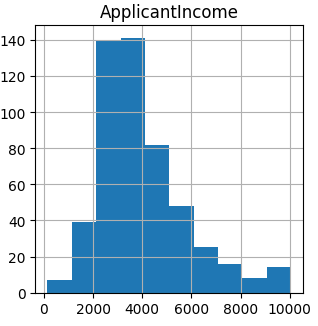
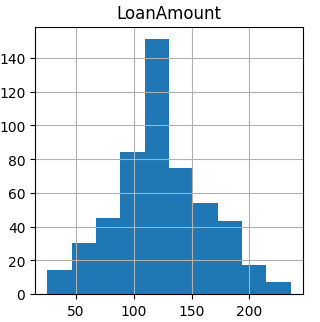
Different encoding methods will have different disadvantages. Keeping those in mind it is required to choose an appropriate encoding technique so that those disadvantages don’t affect us much. For example disadvantage of one-hot encoding is that increases dimensionality but it is ok to choose one-hot encoding where there aren’t many unique values and disadvantage of label encoding is that it imposes ordinality so, you can use that technique where you want ordinality to be imposed and make use of that disadvantage.

**FEATURE SCALING:**

Feature scaling is a method that is used for adjusting the range of independent variables in our dataset. In machine learning feature scaling is crucially important because it ensures that all the features are equally contributing to the model, improving the performance.

In my Loan Approval Prediction, I chose Standardization over Normalization. The reason is Normalization preserves the shape of the data and I did not find that necessary for my project. Instead Standardization preserves the relation between the data points. For this project preserving the relation between the data points is important than preserving the shape of the data so, Standardization is applied to the independent features. Standardization can also be applied to data in Gaussian distribution, there are features in my data that are normally distributed so, even that is one of the reasons to choose Standardization.

Under these circumstances, one good way to make the data points relatively comparable to one another is by transforming the data to have a mean of 0 and a standard deviation of 1. This helps the machine learning models perform better with properly scaled features. Overall, standardization was the best choice for my project in ensuring that relationships between those data points are preserved and model performance is optimized.



**RESULTS**

Coming to the results we will look at the performance of the models used in each dataset. We will see how accurately a model is predicting loan approval. Along with accuracy, there are other metrics like precision, recall, f1-score, and Receiver Operating Characteristic (ROC) curve of each machine learning model. As mentioned in the introduction, the models used in this project are Logistic Regression, Support Vector Classifier, K Nearest Neighbors, and Random Forest. We will also look into the results of Forward Selection.

Let’s see a summary of the models that are being used:

* **K Nearest Neighbors**

K-Nearest Neighbors is a machine learning algorithm that classifies data points by the similarity of a data point to its nearest examples. In the case of classifying a new data point, a KNN algorithm finds its closest neighbors and takes their classes. By choosing a value K for the number of neighbors to consider, KNN decides on a class for a new point based on the majority of classes of the neighbors.

* **Logistic Regression**

Logistic Regression, in very few words, is a way of predicting the probability that something occurs, for example, a yes or no answer. Logistic Regression is therefore used to estimate probabilities, meaning these are the numbers between 0 and 1, based on input data. The categories for a new data point are based on these probabilities. It does so using a very special math formula called the logistic function in order to transform input values into probabilities. If the probability is above a set threshold it predicts one category and if below, it predicts the other. This is a very intuitive, easy-to-understand method and easily the most popular in carrying out classification tasks in machine learning where we have two groups of data.

* **Support Vector Machine**

A Support Vector Classifier finds the best way to divide data into groups. It seeks for a line or boundary that separates data points into different categories. That is, this line shall be farthest from the nearest points of each group, known as support vectors. SVC uses mathematical tricks to deal with cases where data points cannot be separated by a simple line using different shapes or kernels. The optimal boundary enables an SVM classifier, in a new dataset, to correctly classify new data points based on the side of the line it falls to.

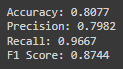
* **Random Forest**

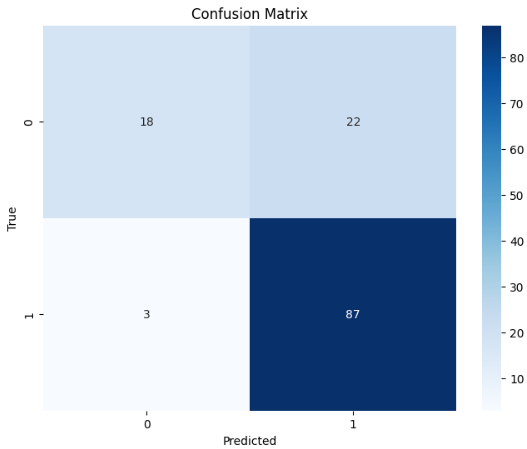
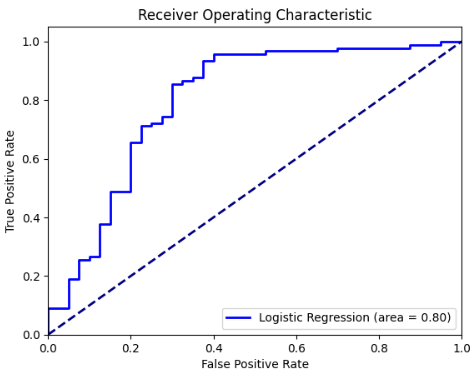
Random Forest works on the committee of decisions principle. At Random Forest, each of its decision-makers acts as a small decision tree, considering different things in the problem and making an individual decision. After that, they vote on the best choice. Like group decision-making, the random forest takes a final vote to make its decision as a combination. Since it is applied by machine learning to make predictions of the collection, Random Forest is robust and accurate for many problem types.

**DATASET 1:**

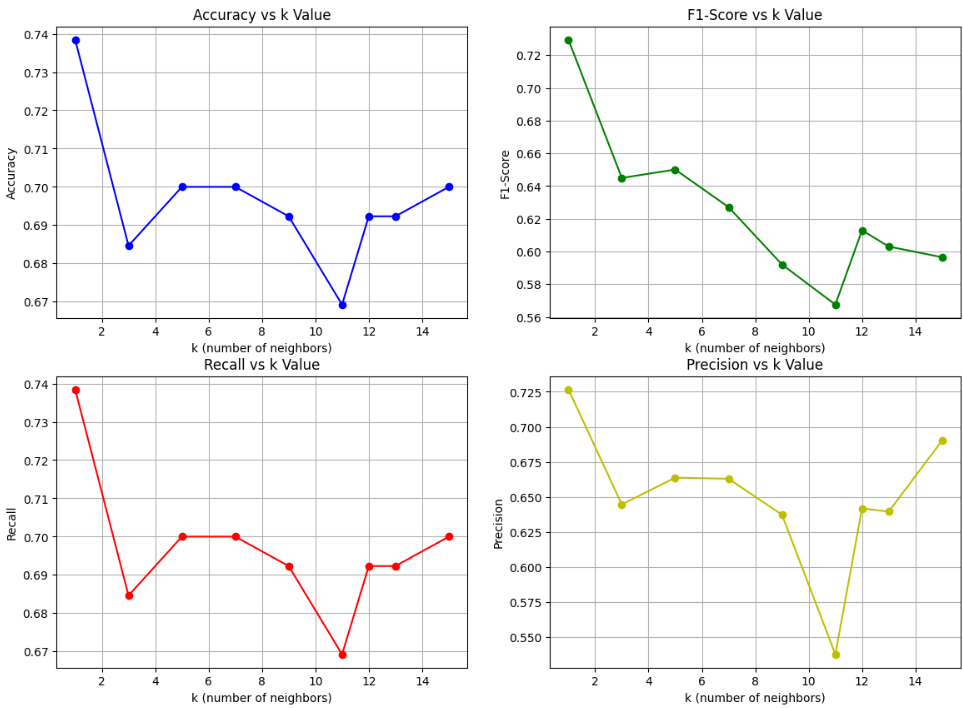
Let’s look at different metrics, confusion matrix and ROC curve

**Logistic Regression:**



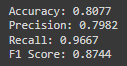
 

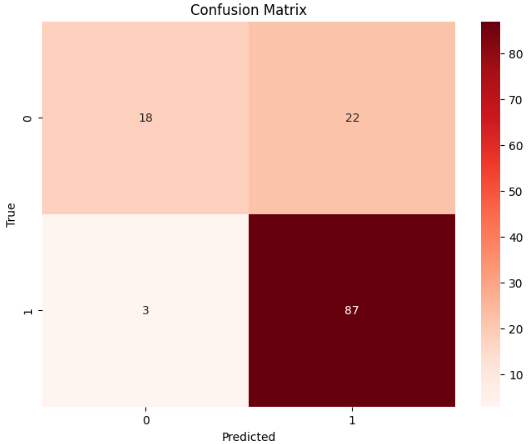
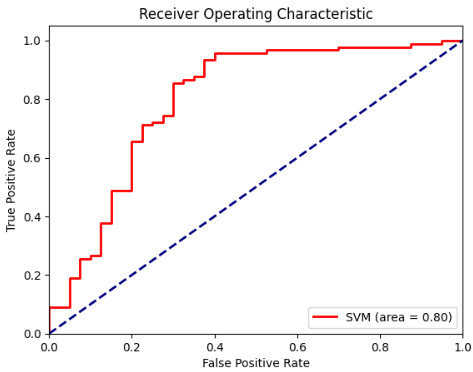
**KNN:**



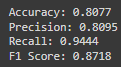
These plots show how good the accuracy, precision, recall and f1-score are for different k values used in KNN.

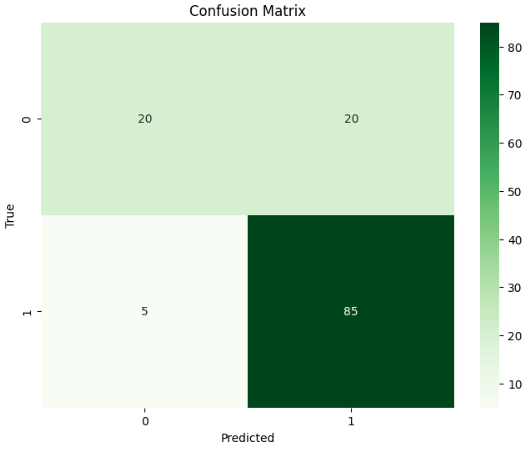
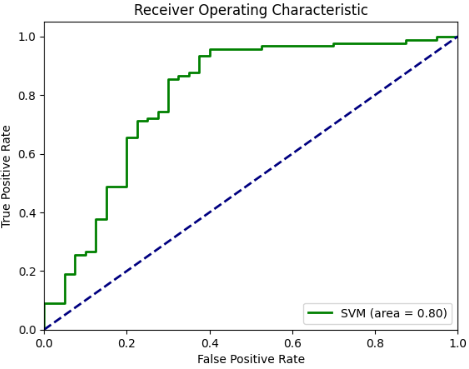
**Support Vector Classifier:**

****

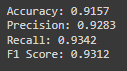
****  ****

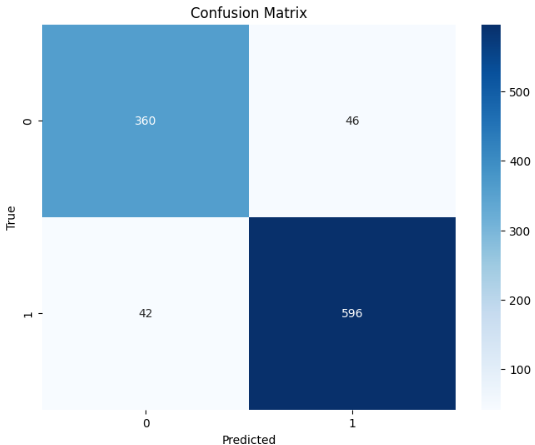
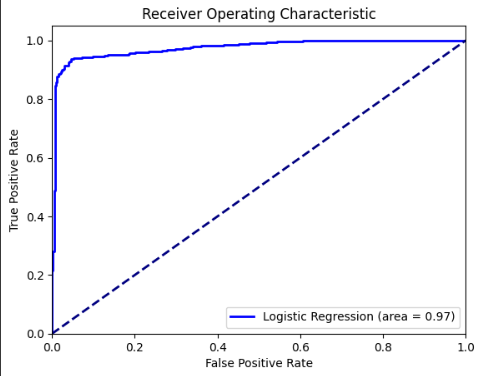
**Random Forest:**

****

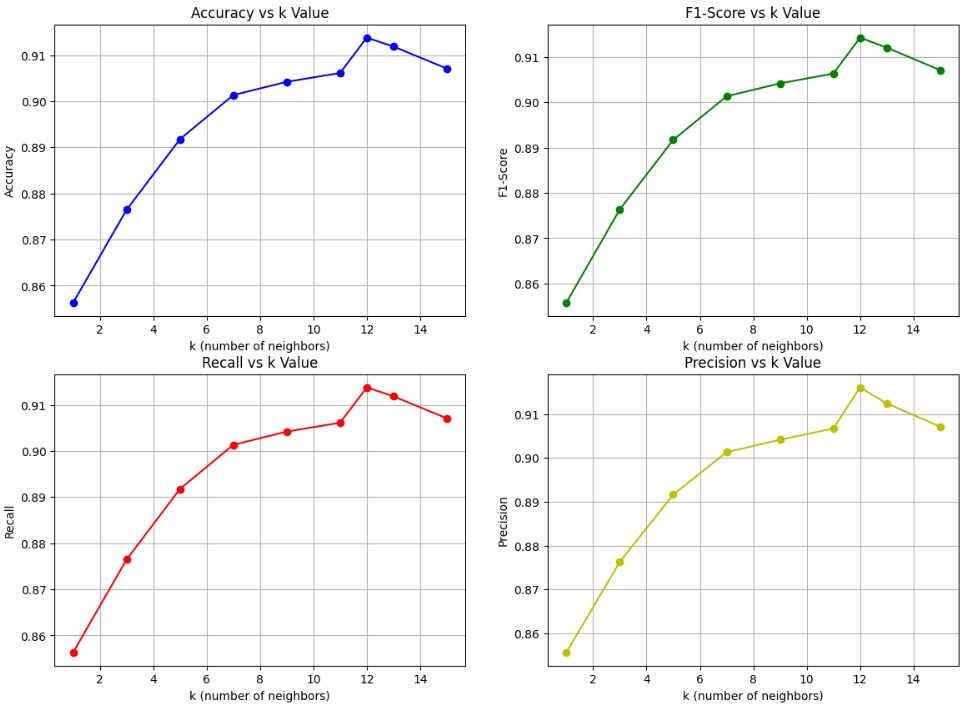
** **

**DATASET 2:  
 Logistic Regression:**

****

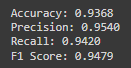
**** ****

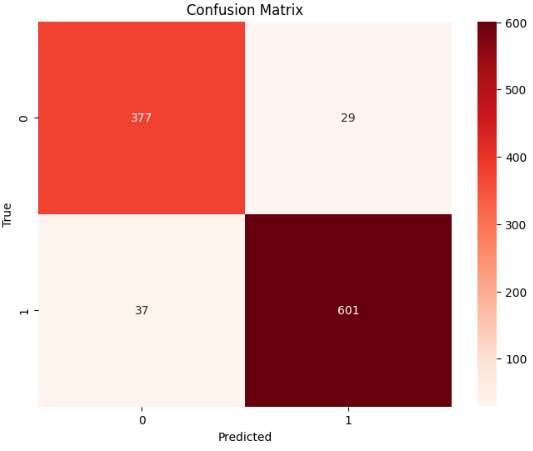
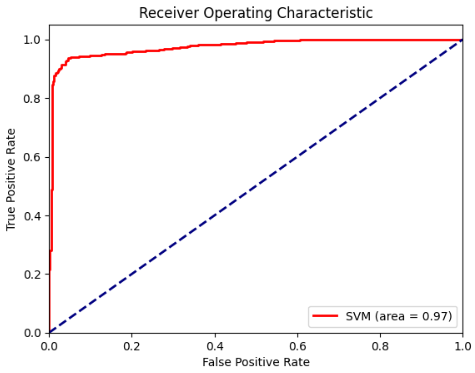
**KNN:**

****

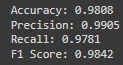
These plots show how good the accuracy, precision, recall and f1-score are for different k values used in KNN.

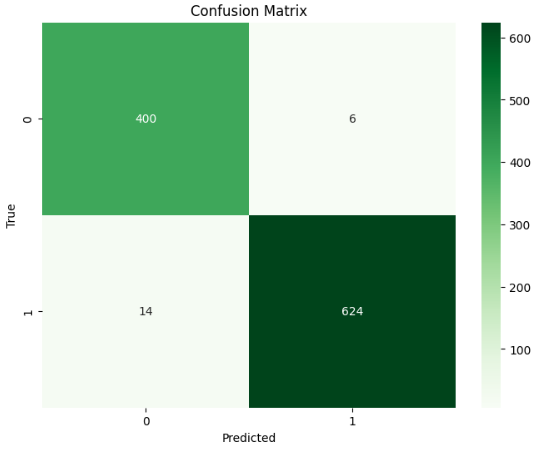
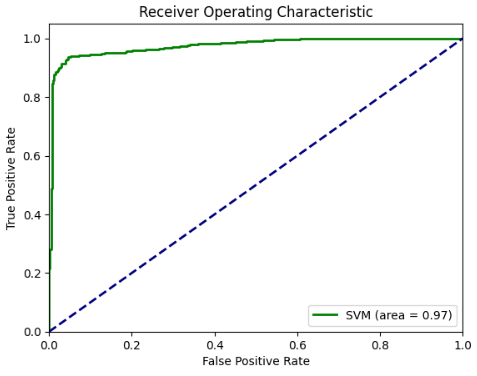
**Support Vector Classifier:**

****

**** ****

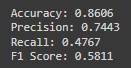
**Random Forest:**

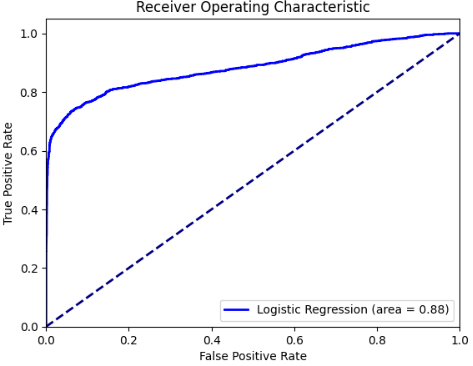
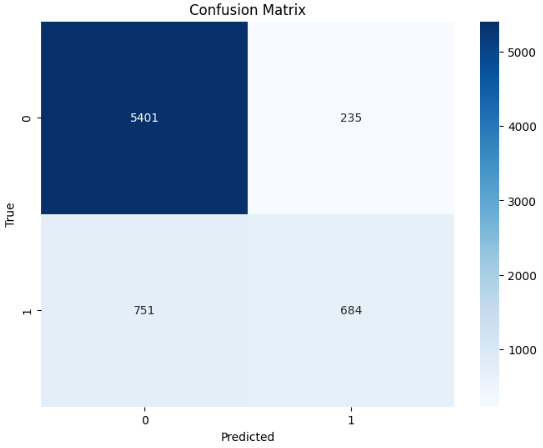
****

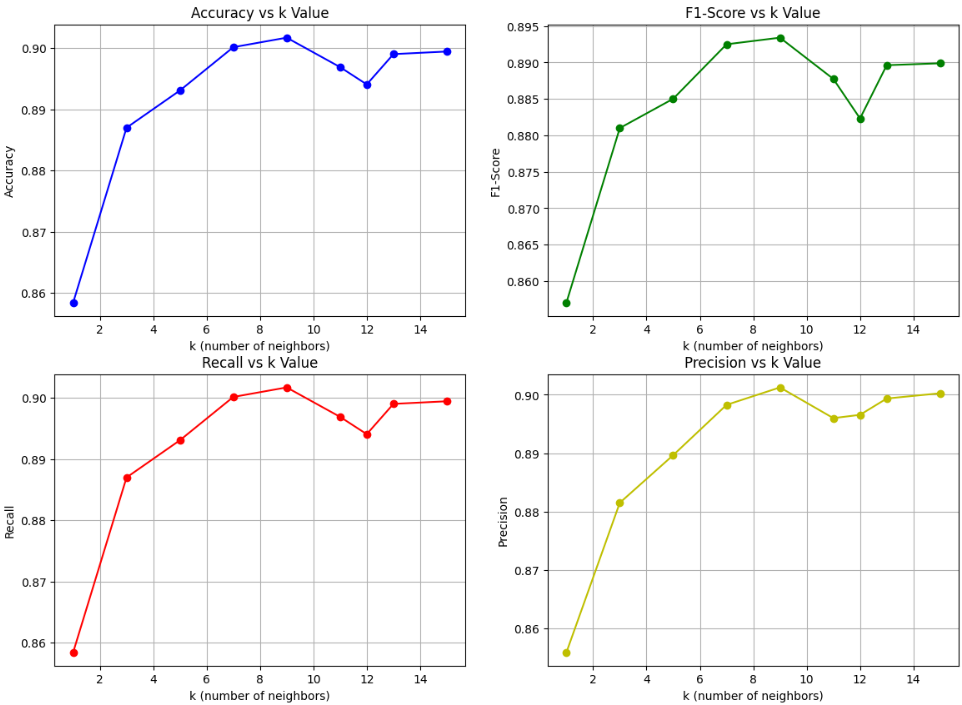
**** ****

**DATASET 3:**

**Logistic regression:**

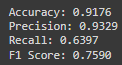
****

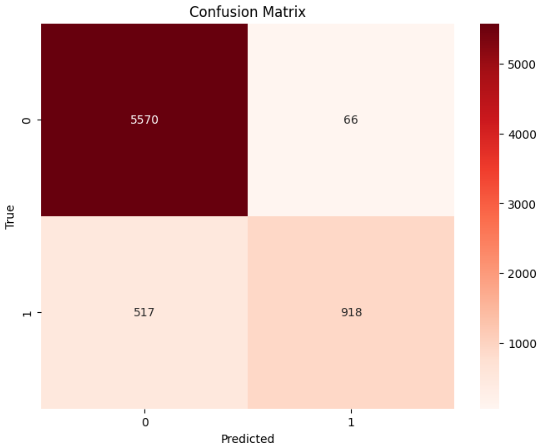
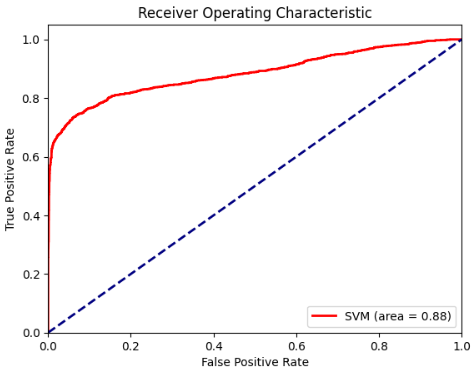
**KNN:**

****

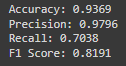
These plots show how good the accuracy, precision, recall and f1-score are for different k values used in KNN.

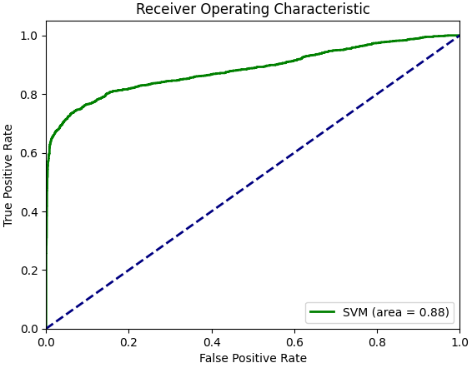
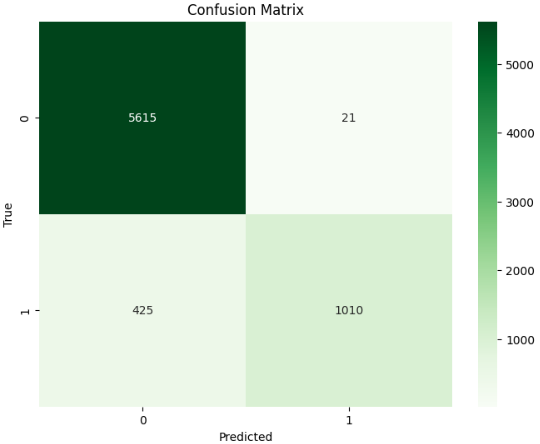
**Support Vector Machine:**

****

**** ****

**Random Forest:**

****

****

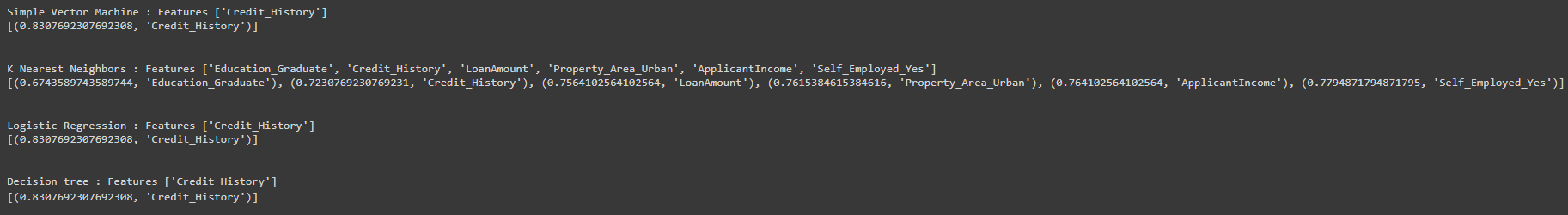
**FORWARD SELECTION:**

Basically, forward selection in machine learning is a feature selection method according to which the model adds features one by one, including only those that improve model performance. It starts from an empty set of features, iteratively selecting the best-performing feature in each step until the stopping criterion such as when the model reaches optimal performance or until a certain predetermined number of features is met. Though computationally intensive, this method efficiently ensures that only the most relevant features are identified for given models.

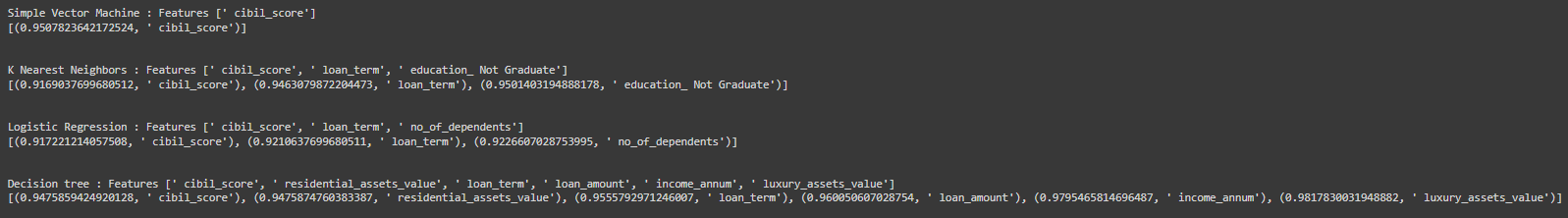
In my project, I implemented a forward selection method without using inbuilt libraries for the Logistic Regression, K-Nearest Neighbors, Support Vector Classifier, and Random Forest models.

Lets look at the results of the forward selection implementation:

**DATASET 1:**



**DATASET 2:**



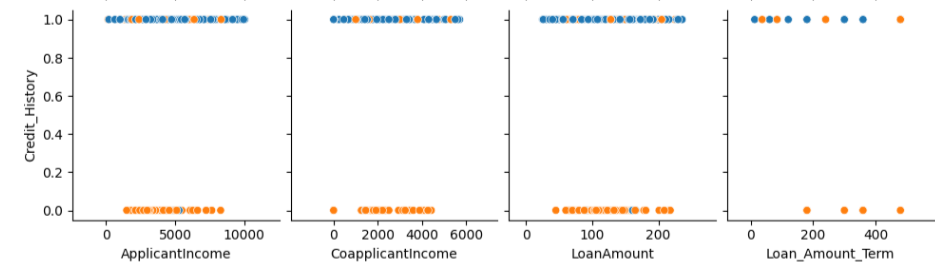
**DATASET 3:**

Since there are 30k+ records in dataset 3, we couldn’t get the output of the forward selection even after running it for 2 hours. It might need some High-performance computing systems to do it. So there are no results of forward elimination for dataset 3.

We could see that Random Forest is performing well on every dataset and holds the highest accuracy when compared to other models executed in this project. But if you look in dataset 1, we can see that the accuracy is exactly the same for logistic regression, Simple vector classifier and random forest. The reason is that the data points are linearly separable when compared with a feature that influences the prediction the most.

If we look at the output of the forward selection above, we can understand that features like cibil\_score and Credit\_History plays a major role in predicting the output.

So let's look at the data under this feature:



Since the data is perfectly linearly separable, this phenomenon of each model having same accuracy has occurred and it is because each model is able to draw a clear boundary between classes without error, leading to identical performance metrics across these classifiers. it’s a valid reason for the question that why the accuracies of different models under the same dataset are exactly the same and that too including the decimals.