PREDICTION OF PERSONAL LOAN APPROVAL USING MACHINE LEARNING



PROJECT

*Internship submitted in partial fulfillment’s of the requirements for the award of degree of*

Bachelor of Technology

In

Computer science and Engineering

By

Team Id : LTVIP2023TMID06434

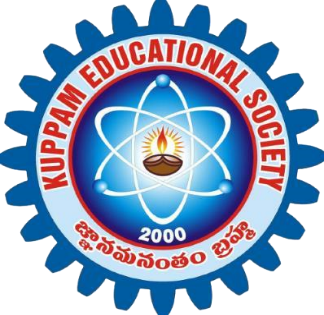
Team Leader : vuppu venkataharika

Team member : jenne Niharika

Team member : v Shruthi

Team member : Bareddy sivaprasad

Under the esteemed guidance of



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**KUPPAM ENGINEERING ANDHRA PRADESH 2022-2023**

INTRODUCTION TO MACHINE LEARNING:

Machine learning is a branch of [artificial intelligence (AI)](https://www.ibm.com/topics/artificial-intelligence) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

IBM has a rich [history](https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibm700series/impacts/) with machine learning. One of its own, Arthur Samuel, is credited for coining the term, “machine learning” with his [research](https://hci.iwr.uni-heidelberg.de/system/files/private/downloads/636026949/report_frank_gabel.pdf) (PDF) around the game of checkers. Robert Nealey, the self-proclaimed checkers master, played the game on an IBM 7094 computer in 1962, and he lost to the computer. Compared to what can be done today, this feat seems trivial, but it’s considered a major milestone in the field of artificial intelligence.

Over the last couple of decades, the technological advances in storage and processing power have enabled some innovative products based on machine learning, such as Netflix’s recommendation engine and self-driving cars.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

Machine learning algorithms are typically created using frameworks that accelerate solution development, such as TensorFlow and PyTorch.

How machine learning works

[UC Berkeley](https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/) breaks out the learning system of a machine learning algorithm into three main parts.

1. A Decision Process: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labeled or unlabeled, your algorithm will produce an estimate about a pattern in the data.
2. An Error Function: An error function evaluates the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.
3. A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this “evaluate and optimize” process, updating weights autonomously until a threshold of accuracy has been met.

Machine learning methods

Machine learning models fall into three primary categories.

### Supervised machine learning

[Supervised learning](https://www.ibm.com/topics/supervised-learning), also known as supervised machine learning, is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately. This occurs as part of the cross validation process to ensure that the model avoids [overfitting](https://www.ibm.com/topics/overfitting) or [underfitting](https://www.ibm.com/topics/underfitting). Supervised learning helps organizations solve a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, and support vector machine (SVM).

### Unsupervised machine learning

[Unsupervised learning](https://www.ibm.com/topics/unsupervised-learning), also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. This method’s ability to discover similarities and differences in information make it ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and image and pattern recognition. It’s also used to reduce the number of features in a model through the process of dimensionality reduction. Principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, k-means clustering, and probabilistic clustering methods.

### Semi-supervised learning

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labeled data set to guide classification and feature extraction from a larger, unlabeled data set. Semi-supervised learning can solve the problem of not having enough labeled data for a supervised learning algorithm. It also helps if it’s too costly to label enough data.

LITERATURE SURVEY

Amira Kamil Ibrahim Hassan, Ajith Abraham (2008) uses a prediction model which is constructed using three different training algorithms to train a supervised twolayer feedforward network. The results show that the training algorithm improves the design of loan default prediction model. Angelini (2008) used a neural network with standard topology and a feed-forward neural network with ad hoc connections. Neural network can be used for prediction model. This paper shows that the above two models give optimum results with less error. Ngai (2009) uses the classification model for predicting the future behaviour of costumers in CRM. In CRM domain, the mostly used model is neural network. He recognized eighty seven articles associated to data mining applications and techniques between 2000 and 2006. Dr. A. Chitra and S. Uma (2010) introduced a ensemble learning method for prediction of time series based on Radial Basis Function networks (RBF), K - Nearest Neighbor (KNN) and Self Organizing Map (SOM). They proposed a model namely PAPEM which perform better than individual model. Akkoç (2012) used a model namely hybrid Adaptive NeuroFuzzy Inference model, grouping of statistics and NeuroFuzzy network. A 10-fold cross validation is used for better results and a comparison with other models. Sarwesh Site, Dr. Sadhna K. Mishra (2013) proposed a method in which two or more classifiers are combined together to produce an ensemble model for the better prediction. They used the bagging and boosting techniques and then used random forest technique. Maher Alaraj, Maysam Abbod, and Ziad Hunaiti (2014) proposed a new ensemble method for classification of costumer loan. This ensemble method is based on neural network. They state that the proposed method give better results and accuracy as compared to single classifier and any other model. AlarajM , AbbodM (2015) introduced a model that are based on homogenous and heterogeneous classifiers. Ensemble model based on three classifiers that are logistic artificial neural network, logistic regression and support vector machine.

DEFINE PROBLEM:

LOANS are the major requirement of the modern world. By this only, Banks get a major part of the total profit. It is beneficial for students to manage their education and living expenses, and for people to buy any kind of luxury like houses, cars, etc.

But when it comes to deciding whether the applicant’s profile is relevant to be granted with loan or not. Banks have to look after many aspects.

So, here we will be using Machine Learning with [Python](https://www.geeksforgeeks.org/python-programming-language/) to ease their work and predict whether the candidate’s profile is relevant or not using key features like Marital Status, Education, Applicant Income, Credit History, etc.

DATA COLLECTION AND PREPARATION:

The dataset collected for foretelling loan failure clients is foretold into Training set and testing

set. Generally 8020 proportion is applied to dissociate the training set and testing set. The data model

which was created using Decision tree is applied on the training set and hung on the test take fineness,

Test set forecasting is done. Following are the attributes

DATASET LINK:

https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset

The dataset contains 13 features :

|  |  |  |
| --- | --- | --- |
| **1** | Loan | A unique id |
| **2** | Gender | Gender of the applicant Male/female |
| **3** | Married | Marital Status of the applicant, values will be Yes/ No |
| **4** | Dependents | It tells whether the applicant has any dependents or not. |
| **5** | Education | It will tell us whether the applicant is Graduated or not. |
| **6** | Self\_Employed | This defines that the applicant is self-employed i.e. Yes/ No |
| **7** | ApplicantIncome | Applicant income |
| **8** | CoapplicantIncome | Co-applicant income |
| **9** | LoanAmount | Loan amount (in thousands) |
| **10** | Loan\_Amount\_Term | Terms of loan (in months) |
| **11** | Credit\_History | Credit history of individual’s repayment of their debts |
| **12** | Property\_Area | Area of property i.e. Rural/Urban/Semi-urban |
| **13** | Loan\_Status | Status of Loan Approved or not i.e. Y- Yes, N-No |

**EXPLORATORY DATA ANALYSIS:**

Preprocessing:

The collected data may contain missing values that may lead to inconsistency. To gain better

results data need to be preprocessed and so it'll better the effectiveness of the algorithm. We should

remove the outliers and we need to convert the variables. In order to flooring these issues we use

chart function.

Train model on training data set:

Now we should train the model on the training dataset and make soothsayings for the test

dataset. We can divide our train dataset into two tract train and testimony. We can train the model on

this training part and using that make soothsayings for the testimony part. In this way, we can validate

our soothsayings as we've the true soothsayings for the testimony part (which we don't have for the

test dataset)

Correlating attributes:

Grounded on the correlation among attributes it was observed more likely to pay back their loans. The

attributes that are individual and significant can include Property area, education, loan measure, and

originally credit History, which is since by insight it's considered as important. The correlation among

attributes can be associated using corplot and boxplot in Python platform

Decision Tree:

Decision tree is a type of supervised education algorithm (having a pre- defined target variable)

that is generally used in category problems. In this approach, we disassociate the population or

sample into two or added homogeneous sets (or sub-populations) based on the most significant

splitter/ differentiator in input variables.

Decision trees use multiple algorithms to decide to disunite a bump into two or added sub- knots. The

creation of sub- knots increases the unsophistication of attendant sub- knots. In other words, we can

say that chasteness of the bump increases with respect to the target variable

**MODEL BUILDING:**

Algorithm:

1. Import all the required python modules
2. [Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe/) – To load the Dataframe
3. [Matplotlib](https://www.geeksforgeeks.org/python-matplotlib-an-overview/) – To visualize the data features i.e. barplot
4. [Seaborn](https://www.geeksforgeeks.org/introduction-to-seaborn-python/) – To see the correlation between features using heatmap
5. Import the database for both TESTING and TRAINING.
6. . Check any NULLVALUES are exists
7. If NULLVALUES exits ,fill the table with corresponding coding
8. . Exploratory Data Analysis for all ATTRIBUTES from the table
9. . Plot all graphs using MATPLOTLIB module
10. 7. Build the DECISIONTREE MODEL for the coding

. Send that output to CSV FILE

PERFORMANCE TESTING AND HYPERPARAMETER TUNING

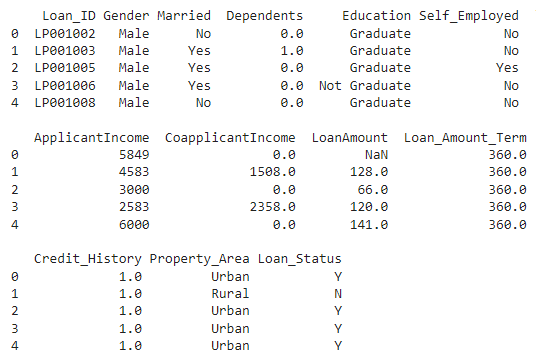
|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns    data = pd.read\_csv("LoanApprovalPrediction.csv") |

Once we imported the dataset, let’s view it using the below command.

* Python3

|  |
| --- |
| data.head(5) |

**Output:**



**Data Preprocessing and Visualization**

Get the number of columns of object datatype.

* Python3

|  |
| --- |
| obj = (data.dtypes == 'object')  print("Categorical variables:",len(list(obj[obj].index))) |

**Output :**

Categorical variables: 7

As Loan\_ID is completely unique and not correlated with any of the other column, So we will drop it using .[drop()](https://www.geeksforgeeks.org/python-delete-rows-columns-from-dataframe-using-pandas-drop/) function.

* Python3

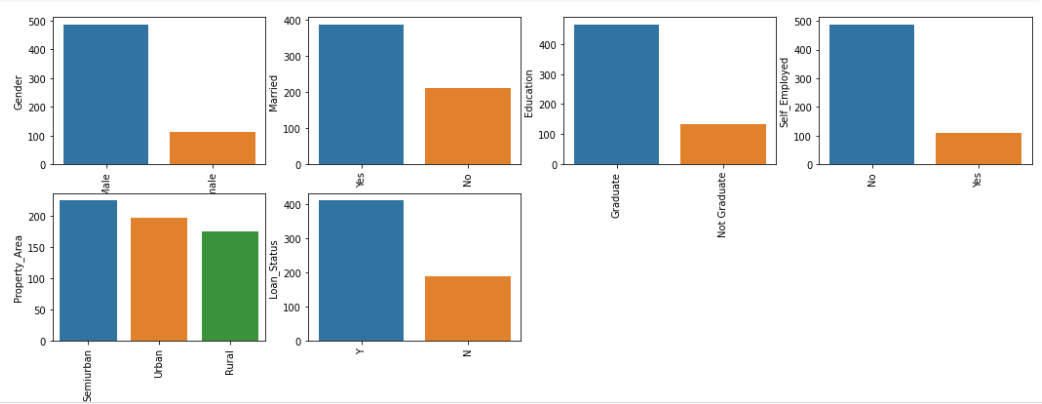
|  |
| --- |
| # Dropping Loan\_ID column  data.drop(['Loan\_ID'],axis=1,inplace=True) |

Visualize all the unique values in columns using [barplot](https://www.geeksforgeeks.org/bar-plot-in-matplotlib/). This will simply show which value is dominating as per our dataset.

* Python3

|  |
| --- |
| obj = (data.dtypes == 'object')  object\_cols = list(obj[obj].index)  plt.figure(figsize=(18,36))  index = 1    for col in object\_cols:    y = data[col].value\_counts()    plt.subplot(11,4,index)    plt.xticks(rotation=90)    sns.barplot(x=list(y.index), y=y)    index +=1 |

**Output:**



As all the categorical values are binary so we can use [Label Encoder](https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/) for all such columns and the values will change into **int** datatype.

* Python3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # Import label encoder   |  | | --- | | from sklearn import preprocessing    # label\_encoder object knows how  # to understand word labels.  label\_encoder = preprocessing.LabelEncoder()  obj = (data.dtypes == 'object')  for col in list(obj[obj].index):    data[col] = label\_encoder.fit\_transform(data[col]) |   Again check the object datatype columns. Let’s find out if there is still any left.   * Python3  |  | | --- | | # To find the number of columns with  # datatype==object  obj = (data.dtypes == 'object')  print("Categorical variables:",len(list(obj[obj].index))) |   **Output :**  Categorical variables: 0   * Python3  |  | | --- | | plt.figure(figsize=(12,6))    sns.heatmap(data.corr(),cmap='BrBG',fmt='.2f',              linewidths=2,annot=True) |   **Output:**  https://media.geeksforgeeks.org/wp-content/uploads/20220810105244/3.PNG    The above heatmap is showing the correlation between Loan Amount and ApplicantIncome. It also shows that Credit\_History has a high impact on Loan\_Status.  Now we will use [Catplot](https://www.geeksforgeeks.org/python-seaborn-catplot/) to visualize the plot for the Gender, and Marital Status of the applicant.   * Python3  |  | | --- | | sns.catplot(x="Gender", y="Married",              hue="Loan\_Status",              kind="bar",              data=data) |   **Output:**  https://media.geeksforgeeks.org/wp-content/uploads/20220810110013/4.PNG    Now we will find out if there is any missing values in the dataset using below code.   * Python3  |  | | --- | | for col in data.columns:    data[col] = data[col].fillna(data[col].mean())    data.isna().sum() |   **Output:**  Gender 0  Married 0  Dependents 0  Education 0  Self\_Employed 0  ApplicantIncome 0  CoapplicantIncome 0  LoanAmount 0  Loan\_Amount\_Term 0  Credit\_History 0  Property\_Area 0  Loan\_Status 0  As there is no missing value then we must proceed to model training.  **Splitting Dataset**   |  | | --- | | from sklearn.model\_selection import train\_test\_split    X = data.drop(['Loan\_Status'],axis=1)  Y = data['Loan\_Status']  X.shape,Y.shape    X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y,                                                      test\_size=0.4,                                                      random\_state=1)  X\_train.shape, X\_test.shape, Y\_train.shape, Y\_test.shape |  * Python3   **Output:**  ((598, 11), (598,))  ((358, 11), (240, 11), (358,), (240,))  **Model Training and Evaluation**  As this is a classification problem so we will be using these models :   * [KNeighborsClassifiers](https://www.geeksforgeeks.org/k-nearest-neighbor-algorithm-in-python/) * [RandomForestClassifiers](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/) * [Support Vector Classifiers (SVC)](https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/) * K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. * K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. * K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. * K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. * K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data. * It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. * KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data. * **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.   K-Nearest Neighbor(KNN) Algorithm for Machine Learning  Why do we need a K-NN Algorithm?  Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:  K-Nearest Neighbor(KNN) Algorithm for Machine Learning  How does K-NN work?  The K-NN working can be explained on the basis of the below algorithm:   * **Step-1:** Select the number K of the neighbors * **Step-2:** Calculate the Euclidean distance of **K number of neighbors** * **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance. * **Step-4:** Among these k neighbors, count the number of the data points in each category. * **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum. * **Step-6:** Our model is ready.   Suppose we have a new data point and we need to put it in the required category. Consider the below image:  K-Nearest Neighbor(KNN) Algorithm for Machine Learning   * Firstly, we will choose the number of neighbors, so we will choose the k=5. * Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:   K-Nearest Neighbor(KNN) Algorithm for Machine Learning   * By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:   K-Nearest Neighbor(KNN) Algorithm for Machine Learning   * As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A. * [RandomForestClassifiers](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/)   Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.  As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.  **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**  The below diagram explains the working of the Random Forest algorithm:  Backward Skip 10sPlay VideoForward Skip 10s  Random Forest Algorithm Note: To better understand the Random Forest Algorithm, you should have knowledge of the Decision Tree Algorithm.Assumptions for Random Forest Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:   * There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result. * The predictions from each tree must have very low correlations.  Why use Random Forest? Below are some points that explain why we should use the Random Forest algorithm:  <="" li="">   * It takes less training time as compared to other algorithms. * It predicts output with high accuracy, even for the large dataset it runs efficiently. * It can also maintain accuracy when a large proportion of data is missing. * [Support Vector Classifiers (SVC)](https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/)   support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.  The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.  SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:  Support Vector Machine Algorithm  **Example:** SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram  Support Vector Machine Algorithm   * [Logistics Regression](https://www.geeksforgeeks.org/understanding-logistic-regression/) * ogistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. * Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**. * Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**. * In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). * The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. * Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. * Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:   Logistic Regression in Machine Learning Note: Logistic regression uses the concept of predictive modeling as regression; therefore, it is called logistic regression, but is used to classify samples; Therefore, it falls under the classification algorithm.Logistic Function (Sigmoid Function):  * The sigmoid function is a mathematical function used to map the predicted values to probabilities. * It maps any real value into another value within a range of 0 and 1. * The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function. * In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.  Assumptions for Logistic Regression:  * The dependent variable must be categorical in nature. * The independent variable should not have multi-collinearity.  Logistic Regression Equation: The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:   * We know the equation of the straight line can be written as:   Logistic Regression in Machine Learning   * In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):   Logistic Regression in Machine Learning   * But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:   Logistic Regression in Machine Learning  The above equation is the final equation for Logistic Regression. Type of Logistic Regression: On the basis of the categories, Logistic Regression can be classified into three types:   * **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc. * **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep" * **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".   To predict the accuracy we will use the accuracy score function from [scikit-learn](https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/) library.   * Python3  |  | | --- | | from sklearn.neighbors import KNeighborsClassifier  from sklearn.ensemble import RandomForestClassifier  from sklearn.svm import SVC  from sklearn.linear\_model import LogisticRegression    from sklearn import metrics    knn = KNeighborsClassifier(n\_neighbors=3)  rfc = RandomForestClassifier(n\_estimators = 7,                               criterion = 'entropy',                               random\_state =7)  svc = SVC()  lc = LogisticRegression()    # making predictions on the training set  for clf in (rfc, knn, svc,lc):      clf.fit(X\_train, Y\_train)      Y\_pred = clf.predict(X\_train)      print("Accuracy score of ",            clf.\_\_class\_\_.\_\_name\_\_,            "=",100\*metrics.accuracy\_score(Y\_train,                                           Y\_pred)) |   **Output  :**  *Accuracy score of  RandomForestClassifier = 98.04469273743017*  *Accuracy score of  KNeighborsClassifier = 78.49162011173185*  *Accuracy score of  SVC = 68.71508379888269*  *Accuracy score of  LogisticRegression = 80.44692737430168*  **Prediction on the test set:**   * Python3  |  | | --- | | # making predictions on the testing set  for clf in (rfc, knn, svc,lc):      clf.fit(X\_train, Y\_train)      Y\_pred = clf.predict(X\_test)      print("Accuracy score of ",            clf.\_\_class\_\_.\_\_name\_\_,"=",            100\*metrics.accuracy\_score(Y\_test,                                       Y\_pred)) |   **Output :**  *Accuracy score of  RandomForestClassifier = 82.5*  *Accuracy score of  KNeighborsClassifier = 63.74999999999999*  *Accuracy score of  SVC = 69.16666666666667*  *Accuracy score of  LogisticRegression = 80.83333333333333* |

## Conclusion :

Random Forest Classifier is giving the best accuracy with an accuracy score of 82% for the testing dataset. And to get much better results ensemble learning techniques like [Bagging](https://www.geeksforgeeks.org/ml-bagging-classifier/) and [Boosting](https://www.geeksforgeeks.org/xgboost/) can also be used.