MACHINE LEARNING

PROJECT BASED REPORT

ON

LOAN PREDICTION USING TENSORFLOW

submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

180031412 - NALLA SRAVAN SAI RAHUL

180031368 – M ROHAN KANTH

180031474 – GANTA YASHWANTHINI

180031488 – M. RUTVIK SAI

Under the Esteemed Guidance of

DR.S SAGAR IMAMBI (Ph.D.)

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K L (Deemed to be) University

Green Fields, Vaddeswaram, Guntur District – 522502

(2020-2021)

K L (Deemed to be) University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is certify that the project based report entitled **"LOAN PREDICTION"** is a bonafide work done and submitted by **N. SRAVAN SAI RAHUL** bearing ID no 180031412, **M.Rohan Kanth**(180031368),**G.Yashwanthini(**180031474),**A.Sahithi**(180031259) in partial fulfilment of the requirements for the award of the degree of **BACHELOROF TECHNOLOGY** in Department of Computer Science Engineering, K L (Deemed to be University), Guntur District during the academic year **2020-2021**.

Dr.S.Sagar Imambi
FACULTY INCHARGE
DEPARTMENT OF CSE
K L (Deemed to be University)

Mr. V. HARI KIRAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF CSE
K L (Deemed to be University)

EXTERNAL EXAMINER

K L (Deemed to be) University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

This is certify that the project based report entitled **"LOAN PREDICTION"** is a bonafide work done and submitted by N. SRAVAN SAI RAHUL(180031412),M.ROHAN KANTH(180031368) , G.YASHWANTHINI(180031474),A.SAHITHI(180031259) in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in Department of Computer Science Engineering, K L (Deemed to be University), Guntur District during the academic year **2020-2021**.

N. SRAVAN SAI RAHUL – 180031412

M.ROHAN KANTH – 180031368

G.YASHWANTHINI – 180031474

A.SAHITHI – 180031259

ACKNOWLEDGEMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my guide **Dr.S.Sagar Imambi** , Associate Professor, Department of Computer Science and Engineering which cannot be expressed in words for his tremendous support, encouragement and guidance for this project.

We express our gratitude to **Mr. V. Hari Kiran,** Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project.

Finally, I sincerely thank my friends and classmates for their kind help and co-operation during our work.

N. SRAVAN SAI RAHUL – 180031412

M.ROHAN KANTH – 180031368

G.YASHWANTHINI – 180031474

A.SAHITHI – 180031259

CONTENTS

Introduction

Banking system in India has been a vital part of every ones lives personally and in business point of view. A lot of customers place request for taking loans based on different criteria and for various reasons, the most important procedure in sanctioning a loan has always been a tough job for banking system.

With the help of Machine learning techniques, we could simplify and accurately categorise the data of the customer's different parameters, so we can get the results for those eligible customers whom we can sanction the loan in a proper, safest and in the fastest way possible.

Developing countries like India have a huge population. Banking must reach out to people even in the remote fragmented locations. Banks are also suffering from diminishing employee satisfaction. Losing out on potential and valuable customer base would be one of the consequences. Top level executives and human resource departments of various banks need to spend time and effort towards retention of their key employees. Banks have also come under the scanner recently, due to various scams and malpractices. The arrest of the Chairman of Syndicate Bank is the latest case in sight.

In this particular project, we have selected different models and compared the models in order to choose the efficient model which gives accurate results and best as well.

We have typically used the TensorFlow model to analyse the bank loan system which will sanction bank loans to eligible customers.

The banking system in India has always been an important part of the Indian economy as well as the individual lives of every Indian citizen. The banks sanction loans to different works such as higher education, personal business, and sanction loans to multinational companies to grow their business rapidly which will generate instant employability to a whole lot of graduates/undergraduates probably jobless.

A country like India is still miles away from being a developed country and countries like ours definitely need a sector which we can rely upon, have our back.

There are a lot of different parameters in which the banks sanction the loans and the data set which we gathered has the elements such as employability status, kind of employability, education background, income details, and so on.

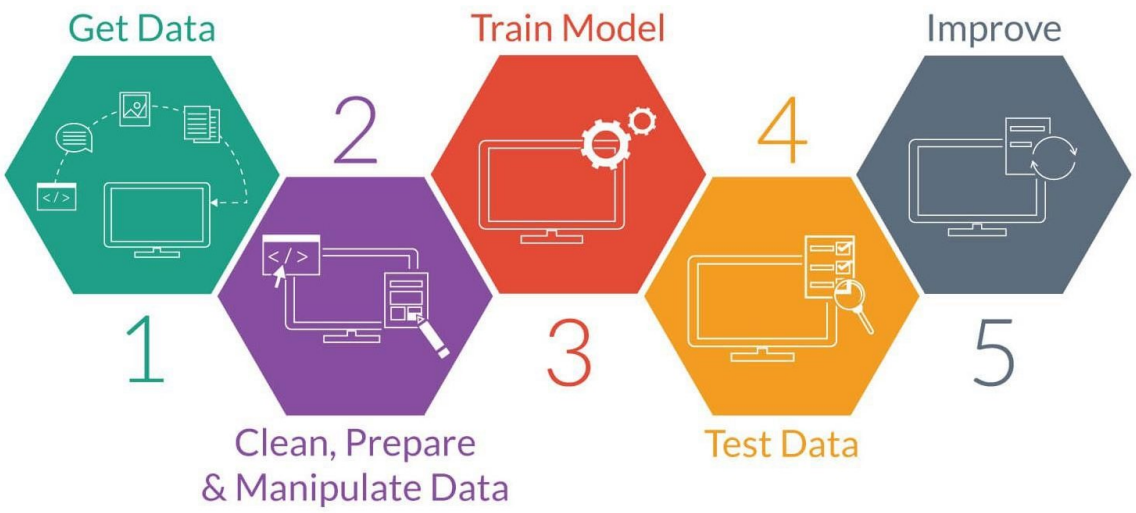
Our machine learning model will help these bankers by considering the required parameters which in turn generate the list of eligible customers whom we can sanction the loans and lay our back on the bed.

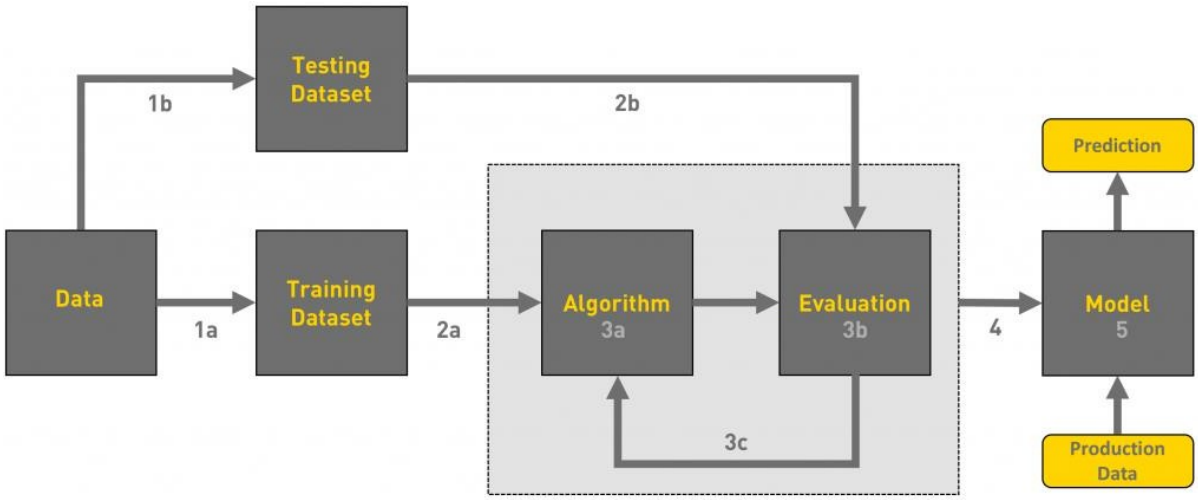
Methodology

Work-Flow

To implement any project, we need to follow certain steps to achieve the end result and so is for Machine Learning project. It completely demonstrates how to carry the work in a Machine Learning project development from start to end. Following workflow steps are applied to any kind of application-oriented Machine Learning project.

1. Data Collection or gathering data
2. Data pre-processing
3. Selection of model based on type of data
4. Training and testing the model
5. Evaluation





Data-Collection

The complete data is collected from the external source Data-Set

Data-Set Description

The training data set is now supplied to machine learning model, on the basis of this data set the model is trained. Every new applicant detail filled at the time of application form acts as a test data set. After the operation of testing, model predict whether the new applicant is a fit case for approval of the loan or not based upon the inference it concludes on the basis of the training data sets

Variable Name	Description	Type
Loan_ID	Unique Loan ID	Integer
Gender	Male/Female	Character
Maritial_Status	Applicant Married(Y/N)	Character
Dependents	Number of Dependents	Integer
Education_Qualification	Graduate/Undergraduate	String
Self_Employed	Self Employed(Y/N)	Character
Applicant_Income	Applicant Income	Integer
Co_Aplicant_Income	Co-applicant Income	Integer
Loan_Amount	Loan Amount in thousands	Integer
Loan_Amount_Term	Term of loan in months	Integer
Credit_History	Credit History meets guidelines	Integer

Property_Area	Urban/Rural/Semi-Urban	String
Loan_Status	Loan Approved	Character

Data Preparation

Data is a precious resource for every organization. But, if we don't analyse that statement further, it can negate itself.

Businesses use data for various purposes. On a broad level, it is used to make informed business decisions, execute successful sales and marketing campaigns, etc. But these cannot be implemented with just raw data.

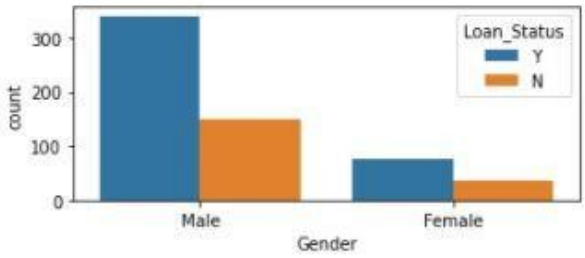
Data becomes a precious resource only if it is cleansed, well-labelled, annotated, and prepared. Once the data goes through various stages of fitness tests it then finally becomes qualified for further processing.

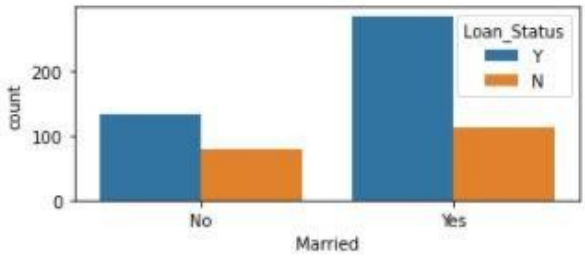
This process contains many steps like Data Extraction, Data Profiling, Data Cleaning, Data Transformation, Data Augmentation, Feature Engineering and many.

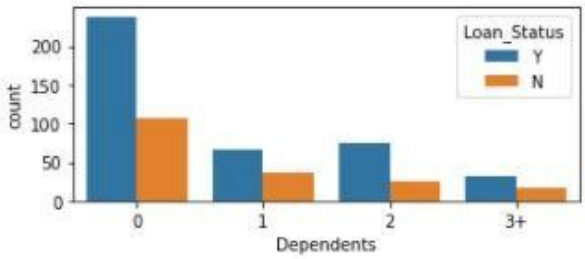
Data Analysis

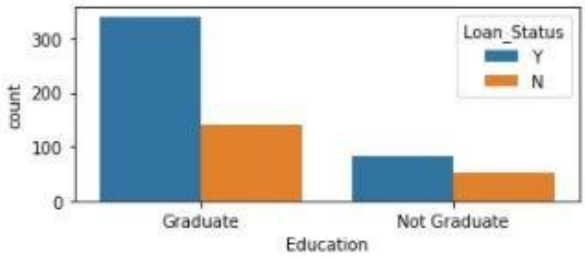
Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

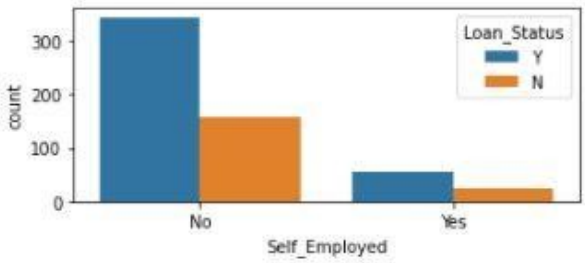
A simple example of Data analysis is whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that particular decision. This is nothing but analysing our past or future and making decisions based on it. For that, we gather memories of our past or dreams of our future. So that is nothing but data analysis. Now same thing analyst does for business purposes, is called Data Analysis.

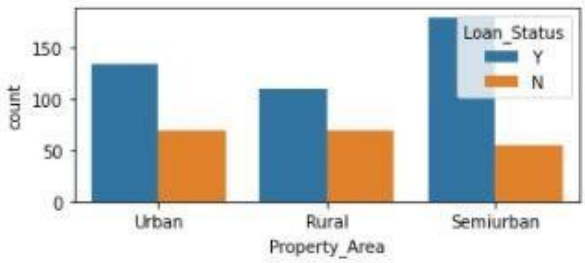












Feature Engineering

The available Techniques

1) Imputation
2) Handling Outliers
3) Binning
4) Log Transform
5) Categorical Encoding
6) Grouping Operations
7) Feature Scaling

Imputation

Missing values are one of the most common problems you can encounter when you prepare your data for machine learning. The reason for the missing values might be human errors, interruptions in the data flow, privacy concerns, etc.

Types of Missing Values:

1) Missing Completely at Random
2) Missing at Random
3) Missing Not at Random

Handling Outliers

Reasons for Outliers

1) Data-Entry Error
2) Measurement Error

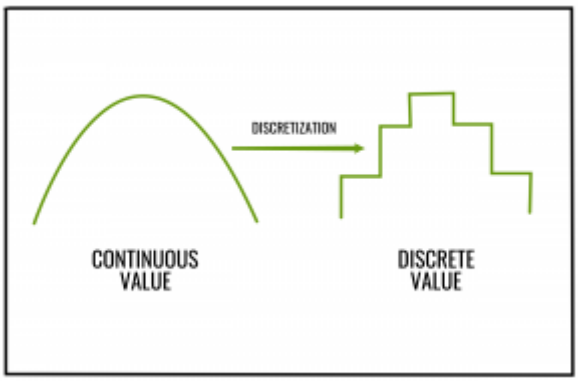
Treating Outliers

1) Deleting Outliers
2) Transforming and Binning Values
3) Imputing Outlier like Missing Values
4) Treat them separately

1. Naivye Bayes Classifier--- Not Sensitive To Outliers
2. SVM-------- Not Sensitive To Outliers
3. Linear Regression---------- Sensitive To Outliers
4. Logistic Regression------- Sensitive To Outliers
5. Decision Tree Regressor or Classifier---- Not Sensitive
6. Ensemble(RF,XGboost,GB)------- Not Sensitive
7. KNN------------------------- Not Sensitive
8. Kmeans---------------------- Sensitive
9. Hierarichal----------------- Sensitive
10. PCA------------------------ Sensitive
11. Neural Networks------------ Sensitive

Binning

The main motivation of binning is to make the model more **robust** and prevent **overfitting**, however, it has a cost to the performance.



Log Transform

Logarithm transformation (or log transform) is one of the most commonly used mathematical transformations in feature engineering. Here are the benefits of using log transform:

- It helps to handle skewed data and after transformation, the distribution becomes more approximate to normal
- It also decreases the effect of the outliers due to the normalization of magnitude differences and the model become more robust.

Categorical Encoding

Typically, any structured dataset includes multiple columns – a combination of numerical as well as categorical variables. A machine can only understand the numbers. It cannot understand the text.

Categorical encoding is a process of converting categories to numbers.

Approaches to Categorical Encoding

1) Label-Encoding
2) One-Hot Encoding

Label Encoding

- Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.

One-Hot Encoding

- One-Hot Encoding is another popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature.

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Grouping Operations

- Using a pivot table or grouping based on aggregate functions using lambda.
- Numerical columns are grouped using sum and mean functions in most of the cases.

Feature Scaling

- Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step

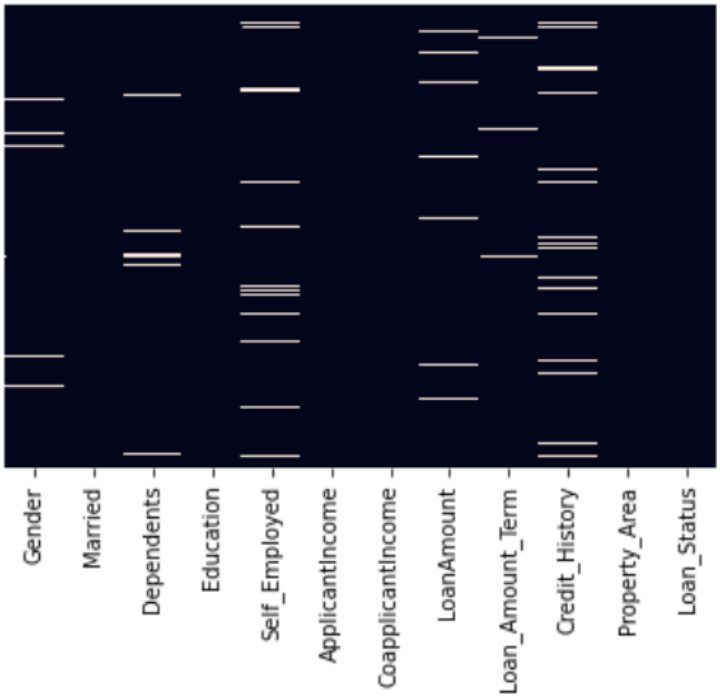
Ways of feature scaling

1) Min Max Scaler
2) Standard Scaler
3) Max Abs Scaler
4) Robust Scaler
5) Quantile Transform Scaler
6) Power Transformer Scaler
7) Unit Vector Scaler

Results of Pre-Processing Data

1) Missing Values

```
: sns.heatmap(train.isnull(),yticklabels=False,cbar=False)
: <AxesSubplot:>
```



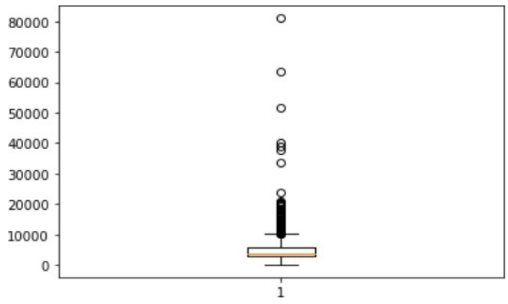
2) Categorical Encoding

```
print("Train Data Categorical Objects")
print()
categorical_features_train = (train.select_dtypes(include=['object']).columns)
for categorical_features_train in train:
    if(train[categorical_features_train].isnull().sum()>0):
        print(categorical_features_train,"=",train[categorical_features_train].isnull().sum())
```

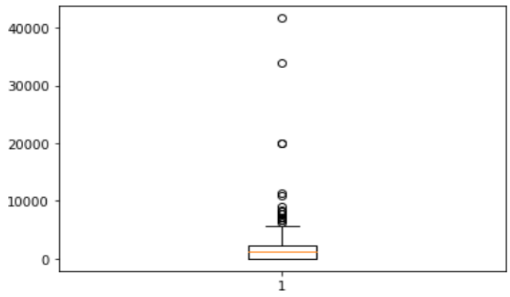
```
Train Data Categorical Objects

Gender = 13
Married = 3
Dependents = 15
Self_Employed = 32
LoanAmount = 22
Loan_Amount_Term = 14
Credit_History = 50
```

3) Outliers

```
plt.boxplot(train['ApplicantIncome'])
```




```
plt.boxplot(train['CoapplicantIncome'])
```



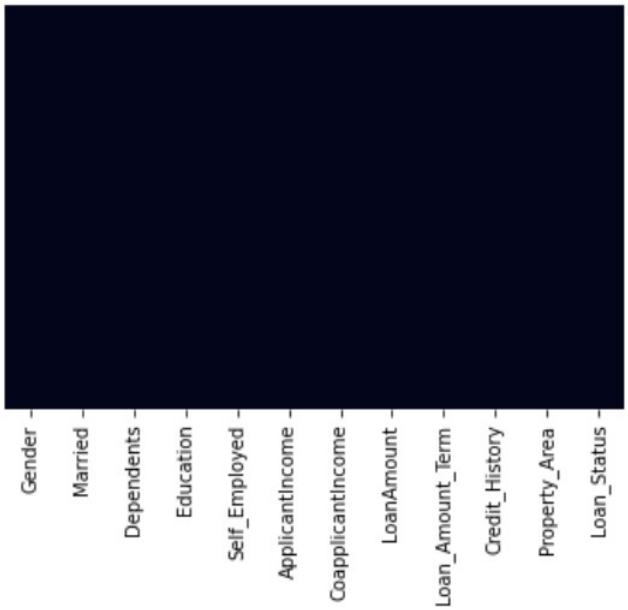
4) Handling Missing Values

```
#filling all the continuous variables
train.fillna(train.mean(),inplace=True)
```

```
#filling all the categorical variables
train['Gender']=train['Gender'].fillna(train['Gender'].mode()[0])
train['Married']=train['Married'].fillna(train['Married'].mode()[0])
train['Dependents']=train['Dependents'].fillna(train['Dependents'].mode()[0])
train['Self_Employed']=train['Self_Employed'].fillna(train['Self_Employed'].mode()[0])
train['LoanAmount']=train['LoanAmount'].fillna(train['LoanAmount'].mode()[0])
train['Loan_Amount_Term']=train['Loan_Amount_Term'].fillna(train['Loan_Amount_Term'].mode()[0])
train['Credit_History']=train['Credit_History'].fillna(train['Credit_History'].mode()[0])
```

```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False)
```

```
: <AxesSubplot:>
```



5) Handleing Categorical Features

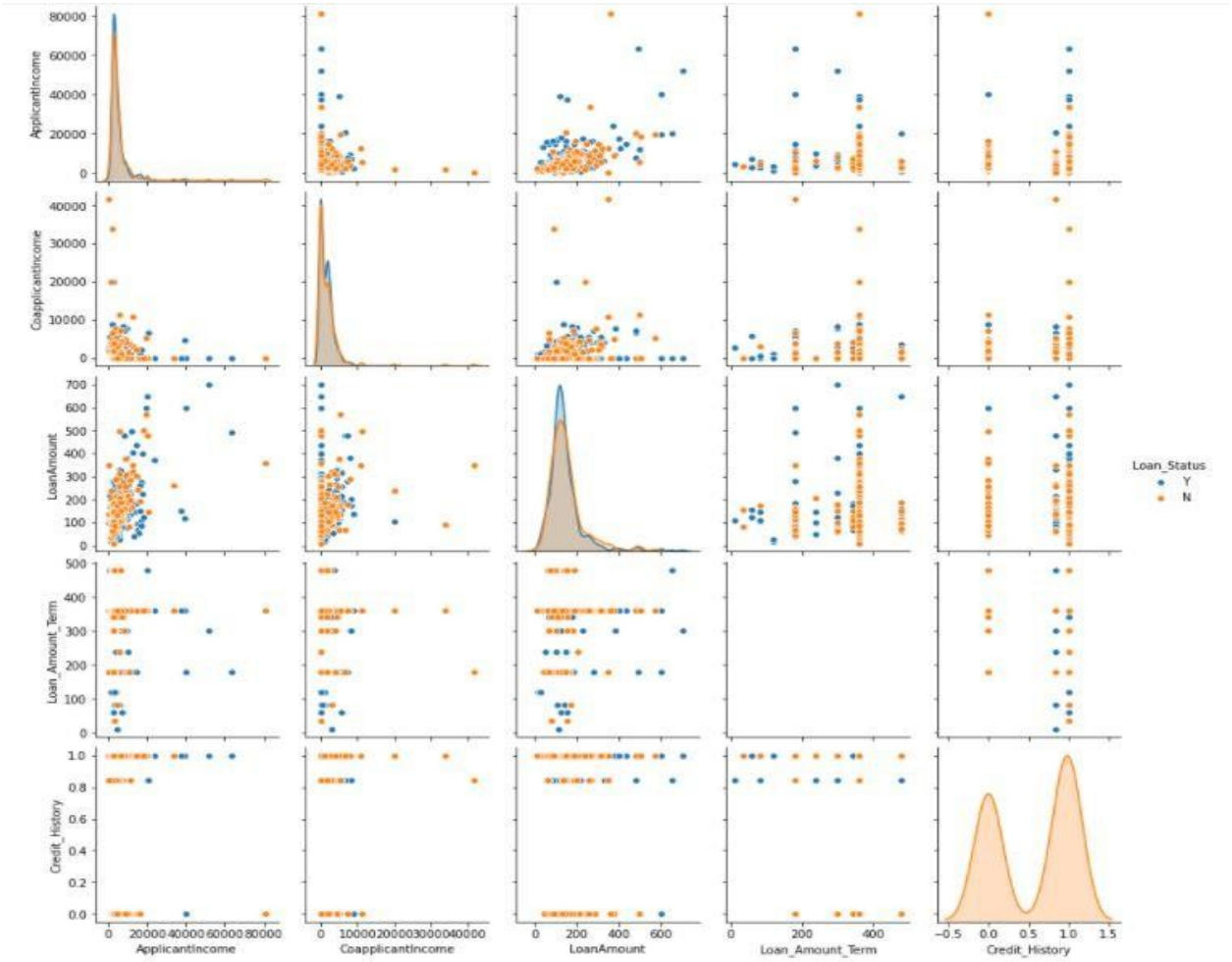
```
traintest = pd.get_dummies(traintest,drop_first=True)
```

```
traintest.dtypes
```

```
ApplicantIncome              int64
CoapplicantIncome          float64
LoanAmount                 float64
Loan_Amount_Term           float64
Credit_History             float64
Gender_Male                  uint8
Married_Yes                  uint8
Dependents_1                 uint8
Dependents_2                 uint8
Dependents_3+                uint8
Education_Not Graduate       uint8
Self_Employed_Yes            uint8
Property_Area_Semiurban      uint8
Property_Area_Urban          uint8
Loan_Status_Y                uint8
dtype: object
```

Pair Plot

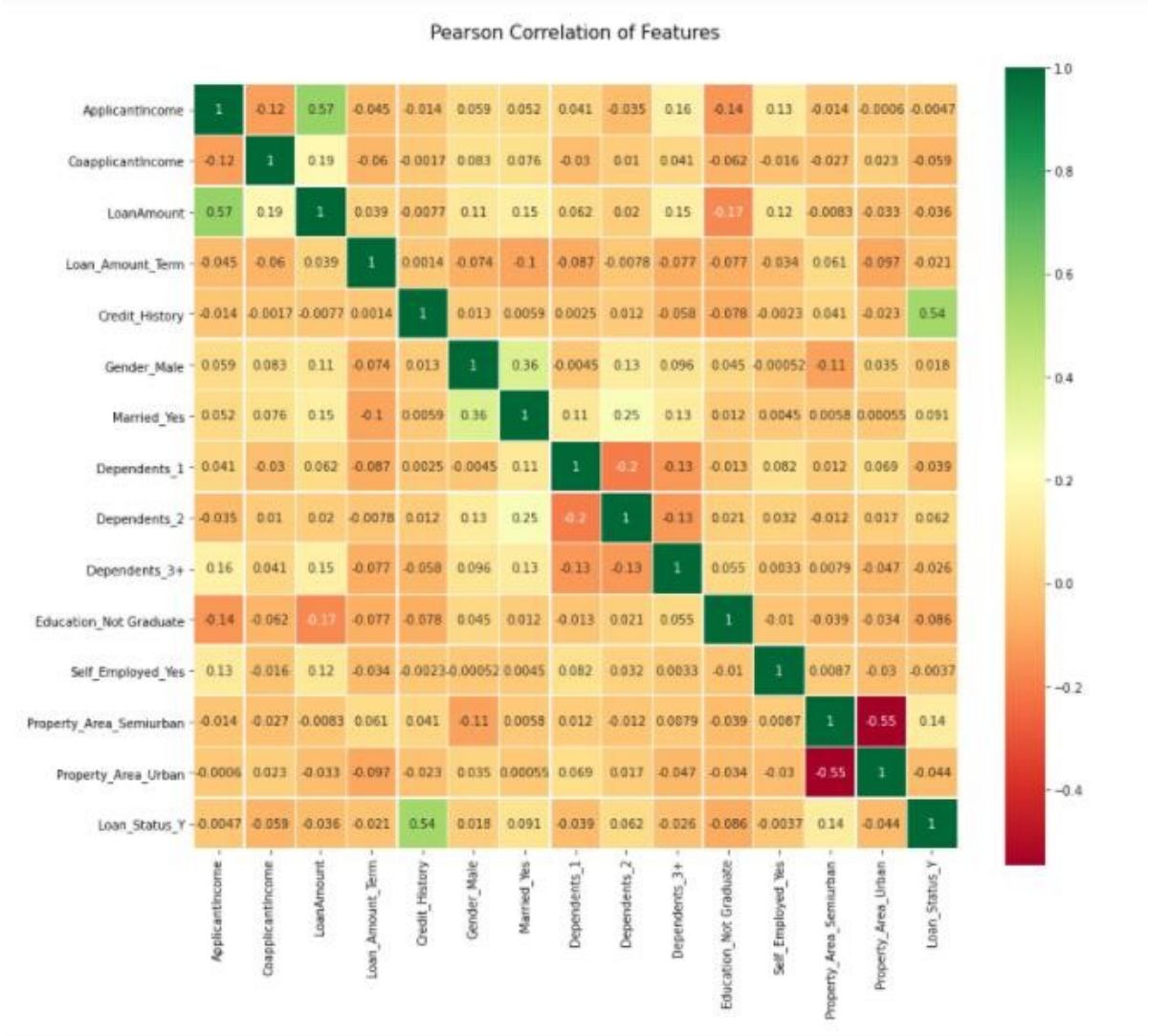
- Plot pairwise relationships in a dataset. It is also possible to show a subset of variables or plot different variables on the rows and columns.



Co-releation

- It finds the strength and direction of the linear releationship.
- These values are standarized
- It’s unit free
- The values range between -1 to +1

```
colormap = plt.cm.RdBu
plt.figure(figsize=(14,12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(train.astype(float).corr(),linewidths=0.1,vmax=1.0,
            square=True, cmap='RdYlGn', linecolor='white', annot=True)
```



Machine Learning Models

Machine learning algorithms enable the creation of a new model using existing anonymized historical data that would be used to train the model to make better predictions not only for credit risks, but also for other risks like the possibility of early repayment resulting in income losses from interest, the possible danger of money laundering etc. With a good model, financial institutions could predict the likeliness of a client repaying the loan before the maturity date and then have procedures with predefined preventive actions to be taken, prior to this happening.

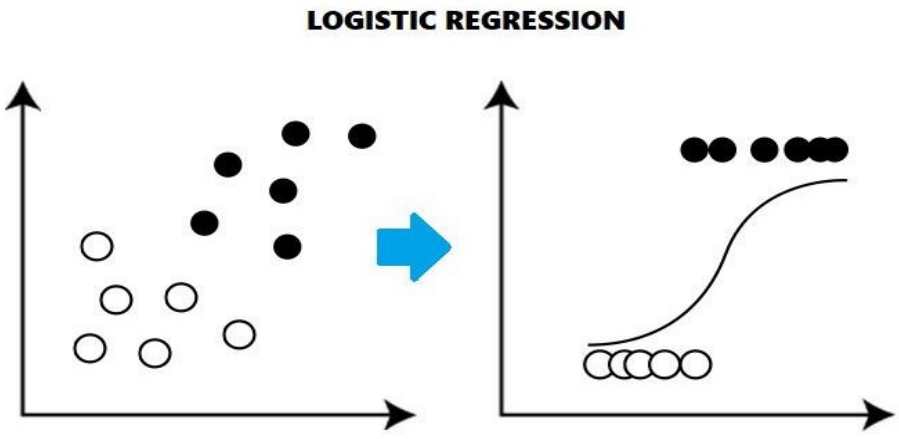
The models that we used are

1) Logistic Regression
2) K-Nearest Neighbours
3) Random Forest
4) Esembling Classifier

The brief details of each model is described below as follows

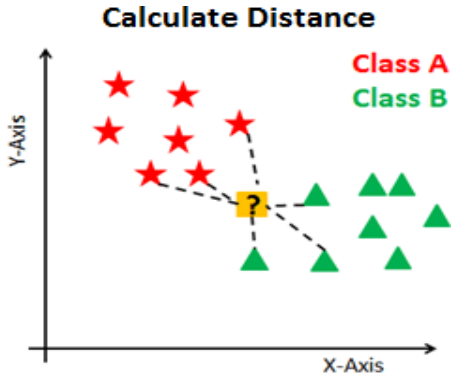
Logistic Regression

- Logistic regression is similar to linear regression but is used to model the probability of a finite number of outcomes, typically two.
- There are a number of reasons why logistic regression is used over linear regression when modelling probabilities of outcomes. In essence, a logistic equation is created in such a way that the output values can only be between 0 and 1 .



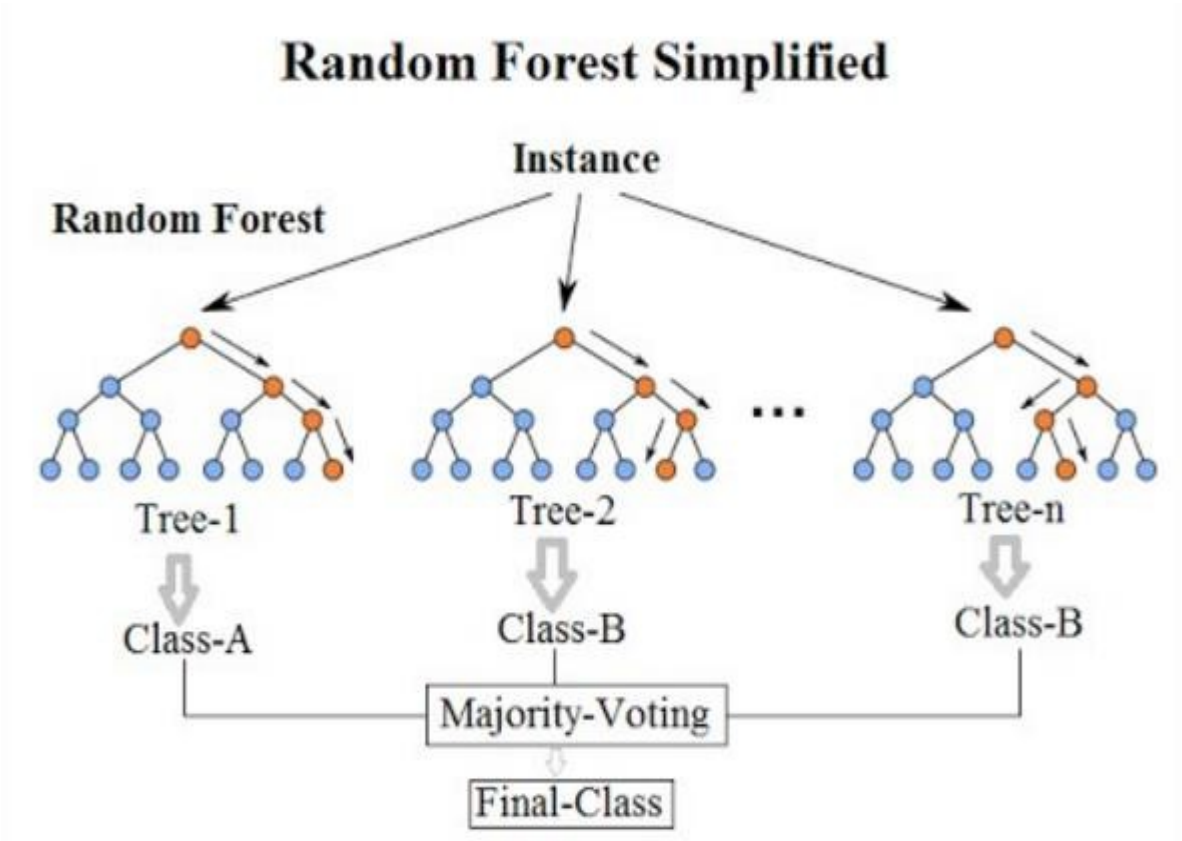
K-Nearest Neighbours

- Assumes that things exist in close proximity Output is calculated as the class with the highest frequency form k − most similar instances
- Each instance votes for their class and the class with the most votes is taken as the prediction



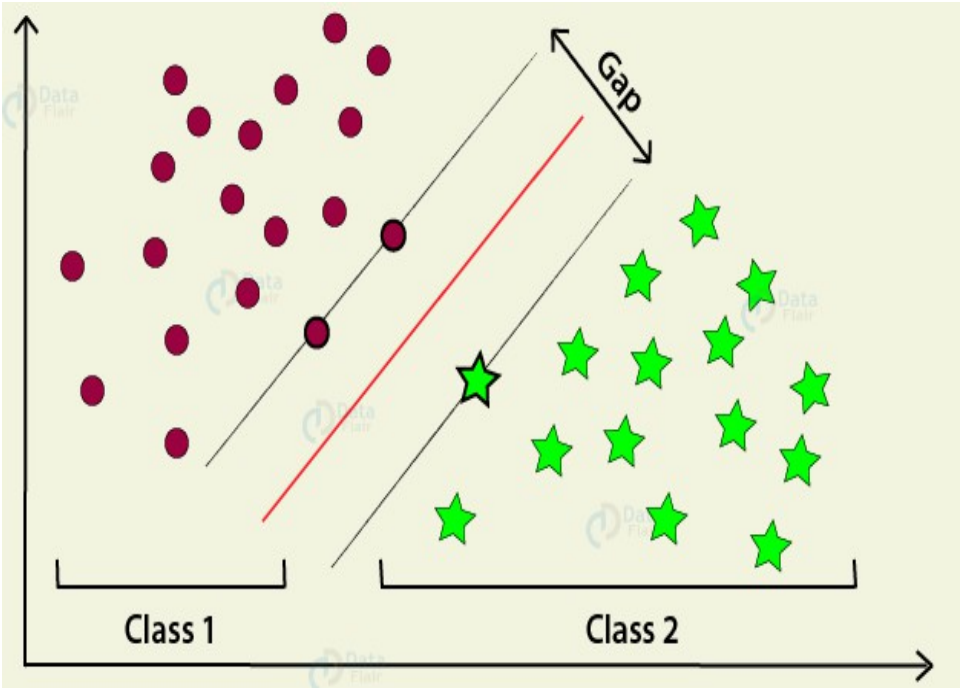
Random Forest

- **Random forests** are an ensemble learning technique that builds off decision trees.
- Random forests involve creating multiple decision trees using bootstrapped datasets of the original data and randomly selecting a subset of variables at each step of the decision tree.
- Then the model selects the mode of all the predictions of each decision tree.



Support Vector Machine

- A Support Vector Machine is a supervised classification technique that can actually get pretty complicated but is pretty intuitive at the most fundamental level.
- A support vector machine will find a **hyperplane** or a boundary between the two classes of data that maximizes the margin between the two classes
- There are many planes that can separate the two classes, but only one plane can maximize the margin or distance between the classes.



Performance Metrics

1) Confusion Matrix

- A confusion matrix is an N X N matrix, where N is the number of classes being predicted.
- Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.
- It Summarizes the performance of classification model

2) Accuracy

- Accuracy in classification problems is the number of correct predictions made by the model over all kinds of predictions made.
- Accuracy is a good measure when the target variable classes in the data are nearly balanced.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

3) Recall

- The proportion of actual positive cases which are correctly identified.
- Out of all actual positive how many are predicted as positive

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

4) Precision

- The proportion of positive cases that were correctly identified.
- Out of all the positive predictions how many are actually positive

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

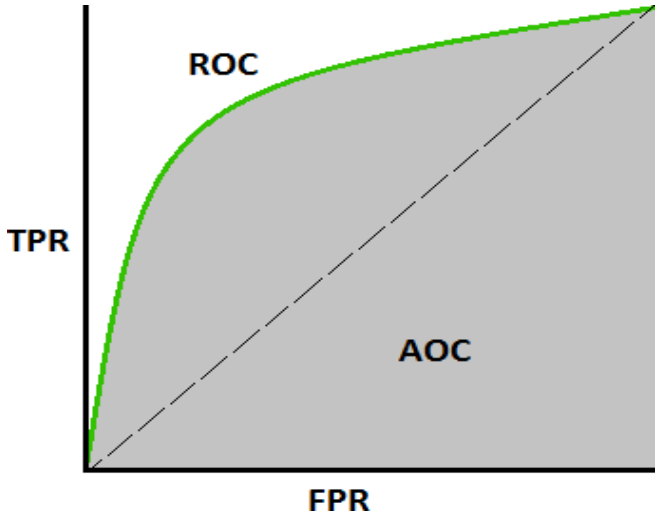
5) F-Beta Score

- F1 Score is used to measure a test's accuracy
- F1 Score is the Harmonic Mean between precision and recall.
- When ever False Positive and False Negative are equally important we use beta value as 1
- Whenever False Positive is having more impact than False Negative the we choose beta value between 0 to 1

$$F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{(\beta^2\cdot\text{precision}+\text{recall})}$$

6) Area Under Curve and ROC

- Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problem.
- The biggest advantage of using ROC curve is that it is independent of the change in proportion of responders.
- The ROC curve is the plot between sensitivity and 1- specificity



Implementation Using TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

We use TensorFlow for the following reasons :-

1) Easy model building
2) Robust ML production anywhere
3) Powerful experimentation for research

As our problem is classification model we can achieve through following models

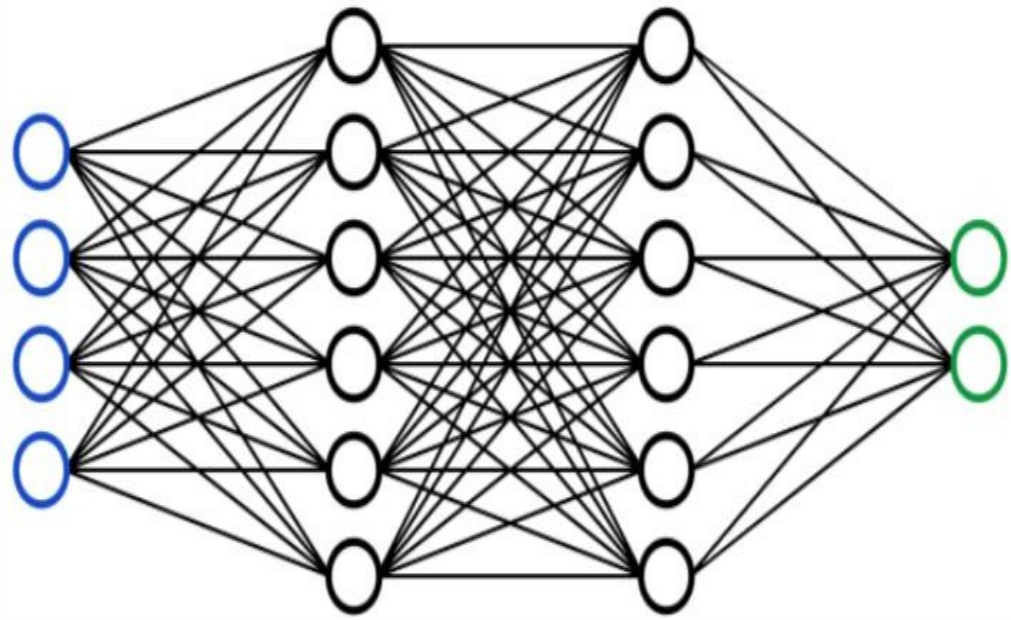
1) Linear Classifier
2) Neural Networks

In this project we are achieving our goal through Neural Networks

Neural Networks

- Neural Network is essentially a network of mathematical equations**.** It takes one or more input variables, and by going through a network of equations, results in one or more output variables
- A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.
- In this sense, neural networks refer to systems of neurons, either organic or artificial in nature
- Neural networks can adapt to changing input, so the network generates the best possible result without needing to redesign the output criteria

Mechanism



- The blue circles represent the **input layer**
- black circles represent the **hidden layers**
- green circles represent the **output layer.**
- Each node in the hidden layers represents both a **linear function** and an **activation function** that the nodes in the previous layer go through, ultimately leading to an output in the green circles.

Implementation

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
log_classifier=LogisticRegression()
print("Train Accuracy =",cv(log_classifier, cv=10).mean())
```

```
Train Accuracy = 0.8921531795435378
```

```
log_classifier.fit(X,y)
pred=log_classifier.predict(test)
print("Test Accuracy =",pred.mean())
```

```
Test Accuracy = 0.8337874659400545
```

K-Nearest Neighbours

```
from sklearn.neighbors import KNeighborsClassifier
k=KNeighborsClassifier(n_neighbors=18)
print("Train Accuracy =",cv(k).mean())
```

```
Train Accuracy = 0.8200728048786399
```

```
k.fit(X,y)
knn_pred=k.predict(test)
print("Test Accuracy =",knn_pred.mean())
```

```
Test Accuracy = 0.9237057220708447
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
random_classifier = RandomForestClassifier(n_estimators=50,max_depth=3)
print("Train Accuracy =",cv(random_classifier, cv=10).mean())
```

```
Train Accuracy = 0.8977844468507428
```

```
random_classifier.fit(X,y)
random_pred=random_classifier.predict(test)
print("Test Accuracy =",random_pred.mean())
```

```
Test Accuracy = 0.8392370572207084
```

SVM

```
from sklearn.svm import SVC
svm_classifier =SVC()
print("Train Accuracy =",cv(svm_classifier, cv=10).mean())
```

```
Train Accuracy = 0.8290343746340472
```

```
svm_classifier.fit(X,y)
svm_pred=svm_classifier.predict(test)
print("Test Accuracy =",svm_pred.mean())
```

```
Test Accuracy = 1.0
```

TensorFlow Implementation

```
model = Sequential([
    Dense(32, activation='relu', input_shape=(11,)),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid'),
])
```

```
model.compile(optimizer='sgd',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

```
hist=model.fit(X_train,Y_train,
          batch_size = 32, epochs = 100,
        validation_data = (x_val,y_val))
```

```
Epoch 11/100
370/370 [==============================] - 0s 86us/step - loss: 0.5951 - accuracy: 0.6784 - val_los
0.6709
Epoch 12/100
370/370 [==============================] - 0s 90us/step - loss: 0.5918 - accuracy: 0.6784 - val_los
0.6709
Epoch 13/100
370/370 [==============================] - 0s 86us/step - loss: 0.5889 - accuracy: 0.6784 - val_los
0.6709
Epoch 14/100
370/370 [==============================] - 0s 83us/step - loss: 0.5859 - accuracy: 0.6784 - val_los
```

```
model.evaluate(X_test,Y_test)[1]
```

```
80/80 [==============================] - 0s 411us/step

0.8374999761581421
```

Accuracy of ML Models

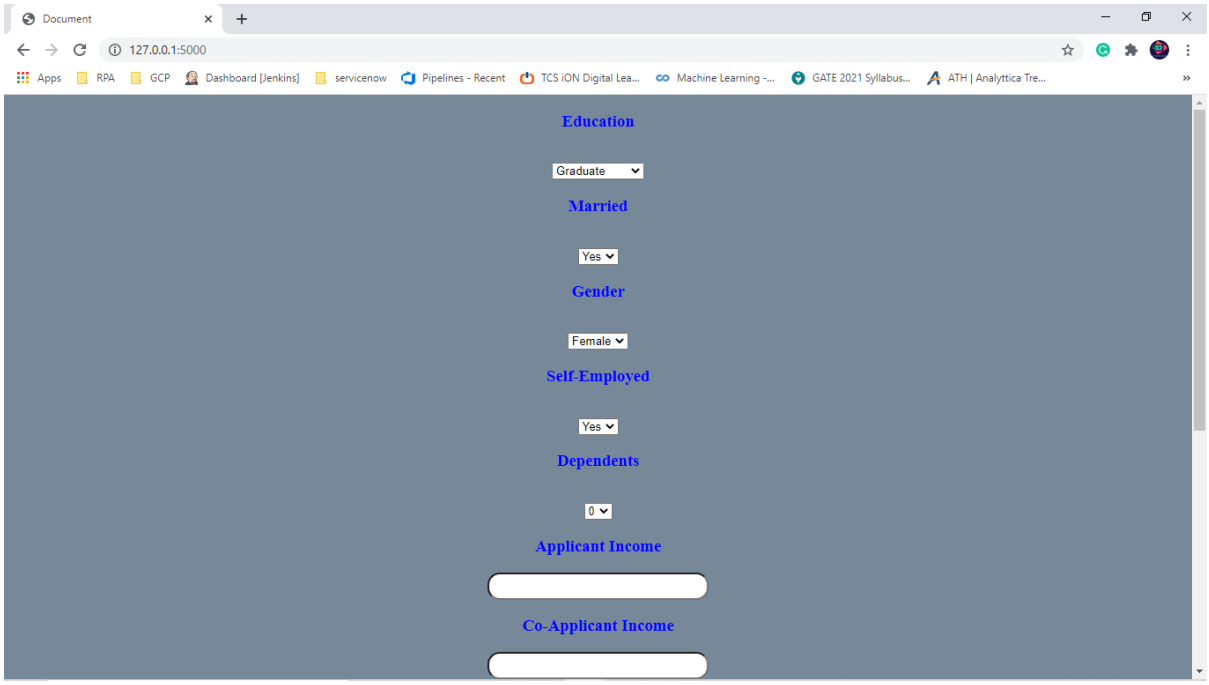
MODEL	TRAIN ACCURACY	TEST ACCURACY
LOGISTIC REGRESSION	0.8921	0.8337
K N N	0.8432	0.9237
RANDOM FOREST	0.8959	0.8392
SVM	0.8290	1.0

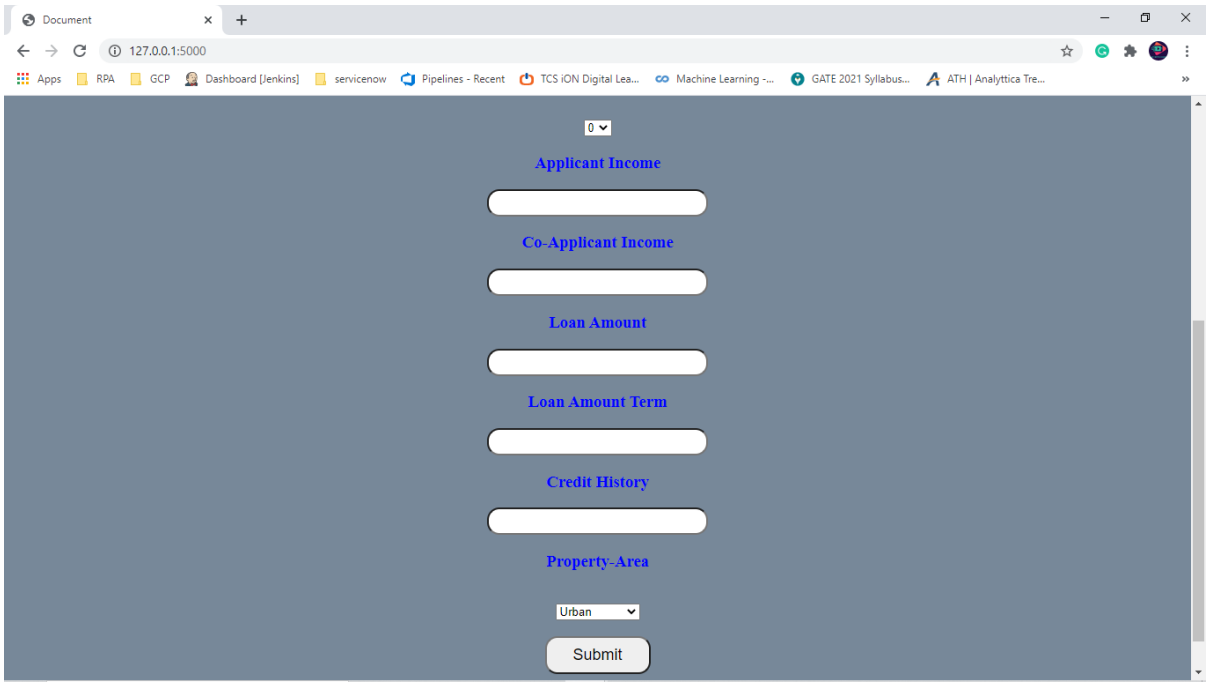
ACCURACY WE ACHIEVED BY USING TENSORFLOW IS 0.8374

Deployment

Deployment is the method by which you integrate a **machine learning** model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the **machine learning** life cycle and can be one of the most cumbersome.

- In this project deployment was done using the flask module





Conclusion and Further Improvements

- In this project we have considered an data set which actually has the criteria for creating and testing a basic machine learning model.
- In order to make the model much more efficient and reliable we actually need to consider different data parameters to sanction the loan.
- There are parameters like property evaluation which is an important step before sanctioning the loan so when we upload those documents into the model we need to use certain software step to step to recognise the documents and completing the status and property check so that we can sanction the loan.
- There are loans where the customer is willing to apply using their homes so there are furthermore evaluations like house dimensions and engineer evaluation, lawyer note so when these all data sets and verifications are successfully carried out then we can sanction the loan for the eligible customers.