CREDIT CARD DEFAULT PREDICTION

MADE BY PRIYANKA

iNeuron

LLD Document

Introduction :

There are times when even a seemingly manageable debt, such as credit cards, goes out of control. Loss of job, medical crisis or business failure are some of the reasons that can impact your finances. In fact, credit card debts are usually the first to get out of hand in such situations due to hefty finance charges (compounded on daily balances) and other penalties. A lot of us would be able to relate to this scenario. We may have missed credit card payments once or twice because of forgotten due dates or cash flow issues. But what happens when this continues for months? How to predict if a customer will be defaulter in next months? To reduce the risk of Banks, this model has been developed to predict customer defaulter based on demographic data like gender, age, marital status and behavioral data like last payments, past transactions etc.

Problem Statement:

Financial threats are displaying a trend about the credit risk of commercial banks as the incredible improvement in the financial industry has arisen. In this way, One of the biggest threats faces by commercial banks is the risk prediction of credit clients. The goal is to predict the probability of credit default based on credit card owner's characteristics and payment history

Approach:

We have to build a solution that should able to predict the probability of credit default based on credit card owner's characteristics and payment history.By observing this approach we can make a classical machine learning tasks like Data Exploration ,Data Cleaning ,Feature Engineering, Model Building and Model Testing .Try out different machine learning algorithms that's best fit for our prediction by using this we can reduce the default customers in the bank. Because this model gives the ability to predict the fiture of a defaulter and non-defaulter customer .If any customer going to default then the bank don’t give them lone and secure from Financial threats

Dataset Information:

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

In this Dataset we have

['ID', 'LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_0',

'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2',

'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1',

'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6',

'default.payment.next.month'], these columns as input.

Briefly about the dataset

Content :

There are 25 variables:

ID: ID of each client

LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary = credit)

SEX: Gender (1=male, 2=female)

EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)

MARRIAGE: Marital status (1=married, 2=single, 3=others)

AGE: Age in years

PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay

for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)

PAY_2: Repayment status in August, 2005 (scale same as above)

PAY_3: Repayment status in July, 2005 (scale same as above)

PAY_4: Repayment status in June, 2005 (scale same as above)

PAY_5: Repayment status in May, 2005 (scale same as above)

PAY_6: Repayment status in April, 2005 (scale same as above)

BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)

BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)

BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)

BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)

BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)

BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)

PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)

PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)

PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)

PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)

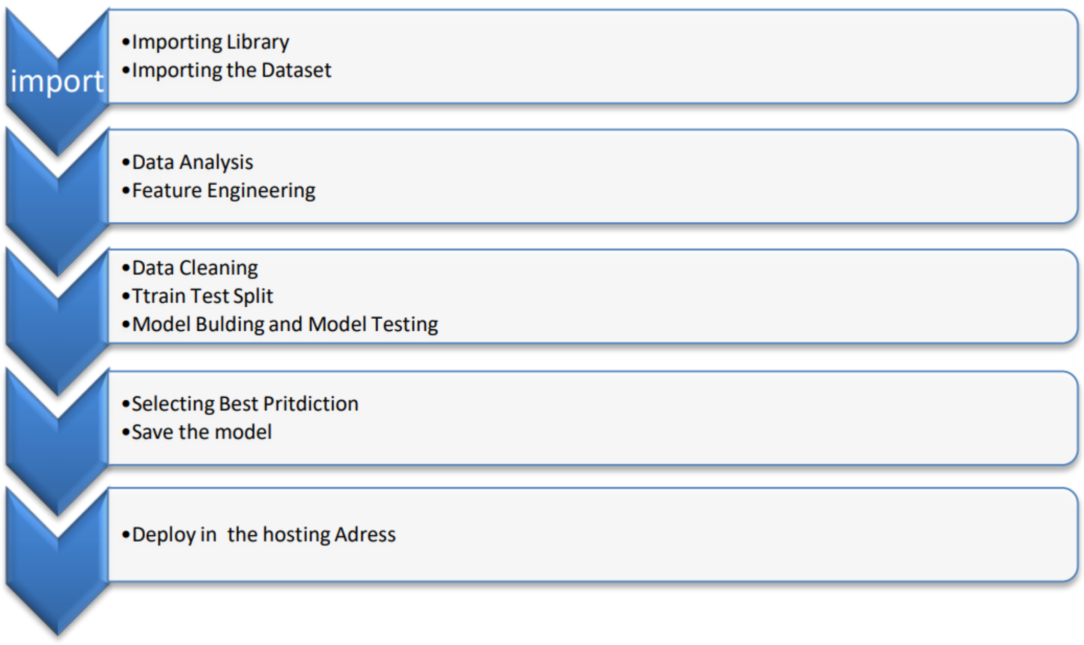
PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)

PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)

default.payment.next.month: Default payment (1=yes, 0=no)

we change “default.payment.next.month “ name to “default” in our model for beater understand

and callable .



Architecture Description :

Data Description:

The dataset was taken from Kaggle (URL: https://www.kaggle.com/uciml/defaultof-credit-card-clients-dataset), This dataset contains information on default payments, demographic factors, credit data, history of payme nt, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

Data Pre-processing:

This included importing of important libraries such as seaborn, matplot lib, pandas ,sklearn etc. We imported the same dataset mentioned above from Kaggle.

Data Analysis:

Here we handled the null values, changed the column names, plotted multiple graphs in seaborn, matplotlib and other visualization library for proper understanding of the data and the distribution of information in the same. As there were no null values in the data, we proceeded with the visualization and analysis.

import

•Importing Library

•Importing the Dataset

•Data Analysis

•Feature Engineering

•Data Cleaning

•Ttrain Test Split

•Model Bulding and Model Testing

•Selecting Best Pritdiction

•Save the model

•Deploy in the hosting Adress

For each specific feature we analysed the data using visualization, and jotted down the important key points which can impact the final predictions.

Feature Engineering :

Merging 2 or mode columns to get indepth knowledge and information regarding the data.

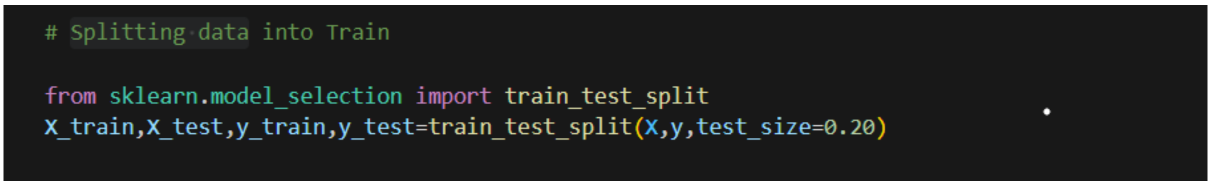
Cleaning the Data:

Remove or delete the unnecessary columns or data for better prediction and also for minimise our file size

Train/Test Split:

This library was imported from Sklearn to divide the final dataset into

the ratio of 80-20%,



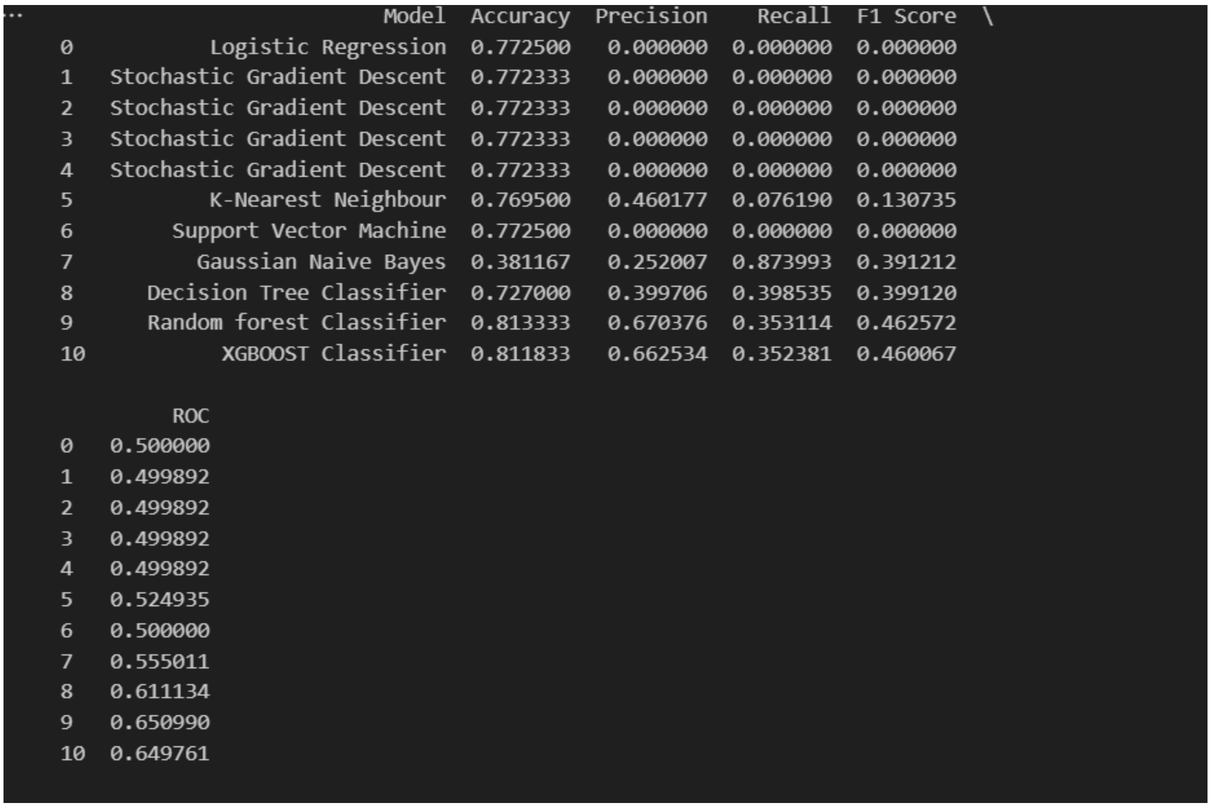
```
# Splitting data into Train

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20)
```

where 80% of the data was used to train the model and the latter 20% was used to predict or test the same model.

Selecting Model:

We tried and tested multiple models such as Stochastic Gradient Descent, K-Nearest Neighbour, Support Vector Machine, Gaussian Naive Bayes, Decision Tree Classifier, Random forest Classifier , XGBOOST Classifier AdaBoost for the model and came up with the model with the best performance, i.e the Random Forest Classifier.

Prediction:

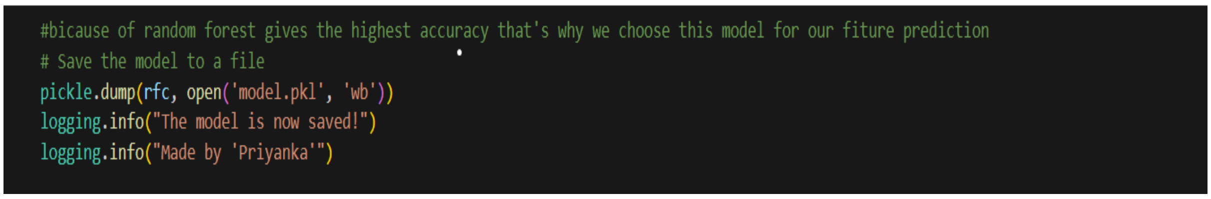


```
...                         Model  Accuracy  Precision    Recall  F1 Score  \
    0             Logistic Regression  0.772500   0.000000  0.000000  0.000000
    1     Stochastic Gradient Descent  0.772333   0.000000  0.000000  0.000000
    2     Stochastic Gradient Descent  0.772333   0.000000  0.000000  0.000000
    3     Stochastic Gradient Descent  0.772333   0.000000  0.000000  0.000000
    4     Stochastic Gradient Descent  0.772333   0.000000  0.000000  0.000000
    5             K-Nearest Neighbour  0.769500   0.460177  0.076190  0.130735
    6          Support Vector Machine  0.772500   0.000000  0.000000  0.000000
    7            Gaussian Naive Bayes  0.381167   0.252007  0.873993  0.391212
    8        Decision Tree Classifier  0.727000   0.399706  0.398535  0.399120
    9        Random forest Classifier  0.813333   0.670376  0.353114  0.462572
    10             XGBOOST Classifier  0.811833   0.662534  0.352381  0.460067

             ROC
    0   0.500000
    1   0.499892
    2   0.499892
    3   0.499892
    4   0.499892
    5   0.524935
    6   0.500000
    7   0.555011
    8   0.611134
    9   0.650990
    10  0.649761
```

The Accuracy of Random Forest was 81.9% and the F1 score was 48.2%.

Save Model:

Model was saved using the pickle library which saves the file in a binary mode.



```python
#bicause of random forest gives the highest accuracy that's why we choose this model for our fiture prediction
# Save the model to a file
pickle.dump(rfc, open('model.pkl', 'wb'))
logging.info("The model is now saved!")
logging.info("Made by 'Priyanka'")
```

Deploy in Local Host:

We created a “HTML” template and deployed the model through “Flask”. And for styling we use the “css” .