

~~CREDIT CARD DEFAULTERS~~

DONE BY:

V. V. S. Madhava

12. 01. 2023

(TASK – o)

NIT Hamirpur

CSE (2020-24)

Abstract:

There are many credit card users, and their number is increasing day by day. Due to user's banks have been facing an escalating credit card default rate. A person becoming default for the credit card payment means people who do not pay their bills and their dues by time. Due to this bank faces a loss, as bank's income is interests on this credit given to users. So, this credit card fraud is a severe issue in financial services area.

1.0 Problem Statement:

The goal of this problem is to predict persons who possibly become as defaulters so that bank can know which customers are likely to pay which customers are not likely to pay the bills or dues. And based on that, the bank will divide their strategy and manage their budget. And they try come up with different strategies and plans.

2.0 Market/Customer/Business Need Assessment

Banks will face a severe problem because of these defaulters. If customers do not pay their bills bank won't get money and then there plans or schedules won't go accordingly.

So, use of this model in financial sector is very much. As this model predicts the defaulters based on matrix features bank can take care

of them either alerting them to pay or changing their budget and plans necessarily.

At the same time customers also gets negative consequences due to this. If a customer does not pay his obligations, banks lose money, the customer will lose credibility in future payments, collection calls start to be made and in last resort, the case may go into the court.



3.0 Target Specification and Characterization

As from both sides i.e., customer and bank negative consequences are there our target is to predict those defaulters by proposing a Machine Learning model.

The dataset features are LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE, PAY_o to PAY_6, BILL_AMT1 to BILL_AMT6, PAY_AMT1 to PAY_AMT6 and finally dependent variable which is DEAFULT PAYMENT NEXT MONTH.

4.0 External Research

The sources I have used as reference for analysing and finding necessary features in dataset and for validation part and relevant papers are mentioned below:

- https://www.researchgate.net/publication/356563799_Credit_Card_Fraud_Detection_using_Machine_Learning_Algorithms
- <https://www.hindawi.com/journals/complexity/2021/6618841/>
- <https://www.slideshare.net/alexpnt/default-credit-card-prediction>

I modified the dataset found on Kaggle by adding few more features to it on my own thinking. And I studied about this problem

thoroughly from online research papers from sites ieeexplore.ieee.org, researchgate.com and [Hindawi](http://Hindawi.com).

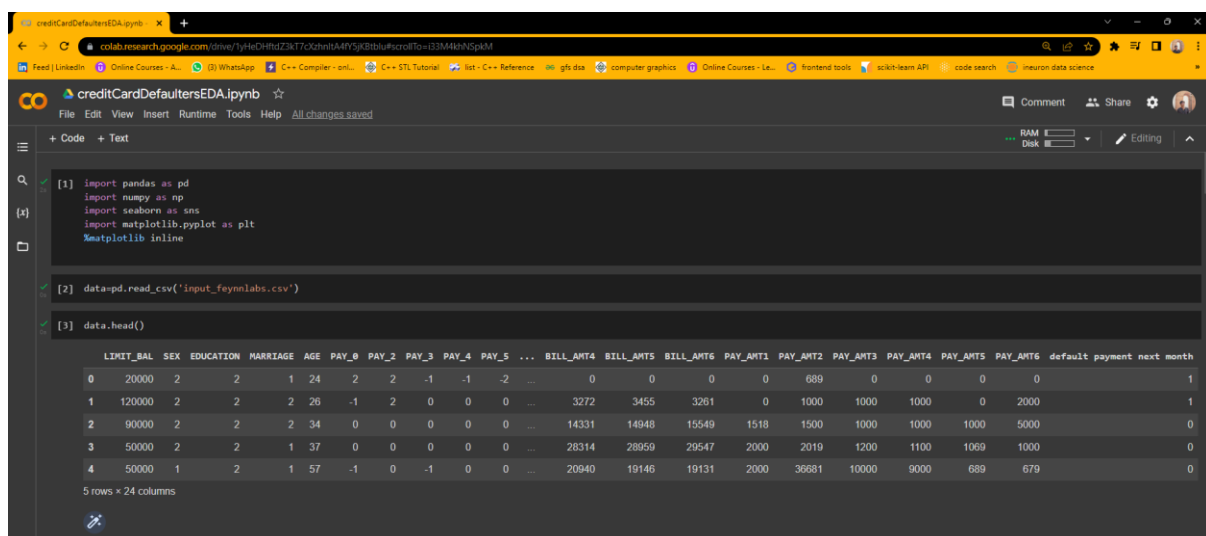
Final matrix of features in my sample dataset are:

- 1. LIMIT_BAL:** Credit limit of the person. (Continuous)
- 2. SEX:** Categorical: 1 = male; 2 = female
- 3. EDUCATION:** Categorical: 1 = graduate school; 2 = university; 3 = high school; 4 = others
- 4. MARRIAGE:** 1 = married; 2 = single; 3 = others
- 5. AGE:** number (Continuous)
- 6. PAY_0 to PAY_6:** History of past payment.
- 7. BILL_AMT1 to BILL_AMT6:** Amount of bill statements.
- 8. PAY_AMT1 to PAY_AMT6:** Amount of previous payments.

Target Label: (Whether a person shall default in the credit card payment or not)

9.default payment next month: Yes = 1; No = 0

Let's view our dataset:



```
[1] import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

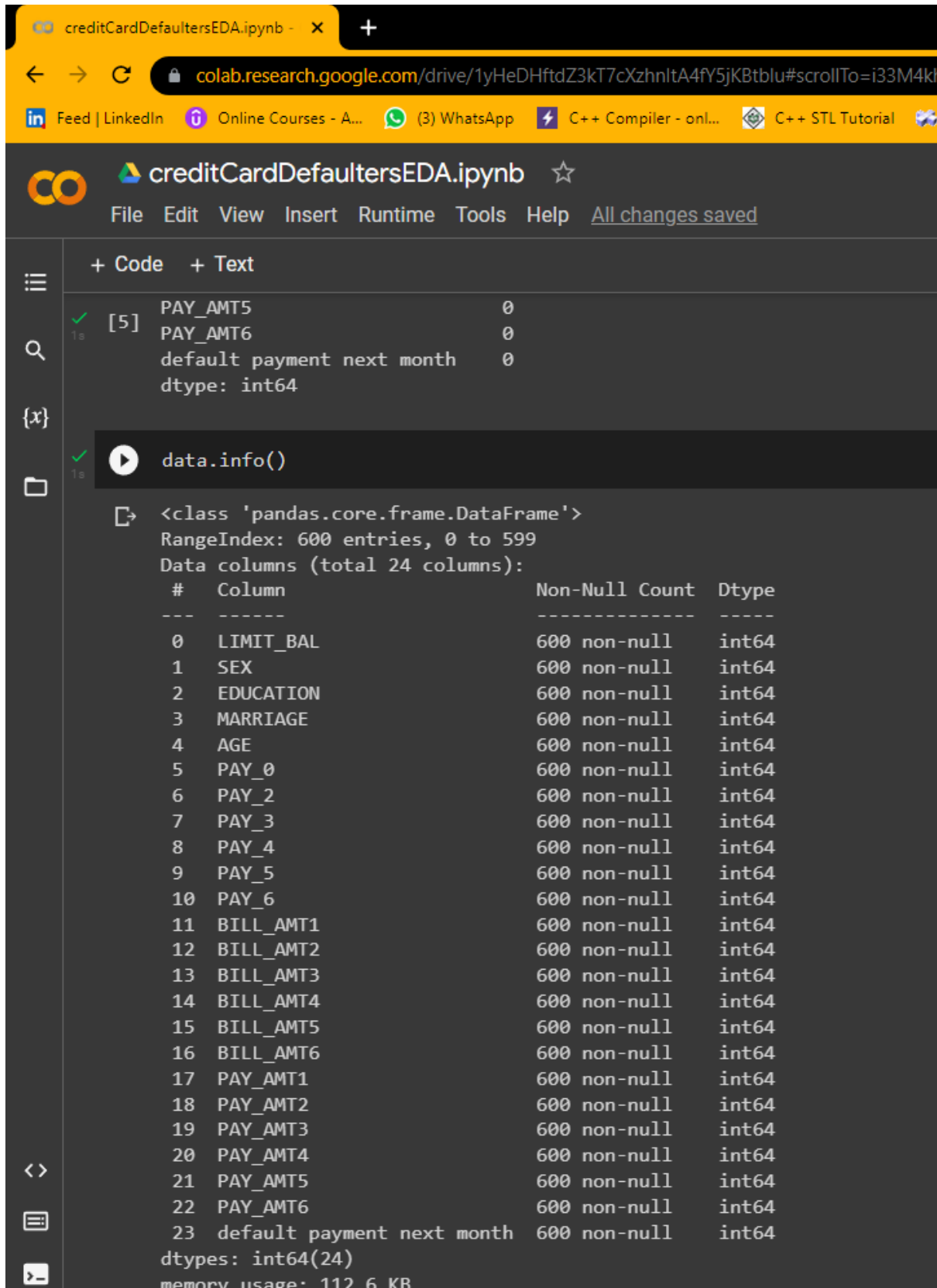
[2] data=pd.read_csv('input_feynmlabs.csv')

[3] data.head()
```

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
0	20000	2	2	1	24	2	2	-1	-1	-2	...	0	0	0	0	689	0	0	0	0	1
1	120000	2	2	2	26	-1	2	0	0	0	...	3272	3455	3261	0	1000	1000	1000	0	2000	1
2	90000	2	2	2	34	0	0	0	0	0	...	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
3	50000	2	2	1	37	0	0	0	0	0	...	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
4	50000	1	2	1	57	-1	0	-1	0	0	...	20840	19146	19131	2000	36681	10000	9000	689	679	0

5 rows x 24 columns

Let's see some information about our dataset:



The screenshot shows a Google Colab notebook titled "creditCardDefaultersEDA.ipynb". The notebook interface includes a menu bar with File, Edit, View, Insert, Runtime, Tools, Help, and a status bar indicating "All changes saved". The left sidebar shows a file explorer with a folder icon and a search icon. The main area displays a code cell with the following content:

```
[5] PAY_AMT5          0
     PAY_AMT6          0
     default payment next month  0
     dtype: int64
```

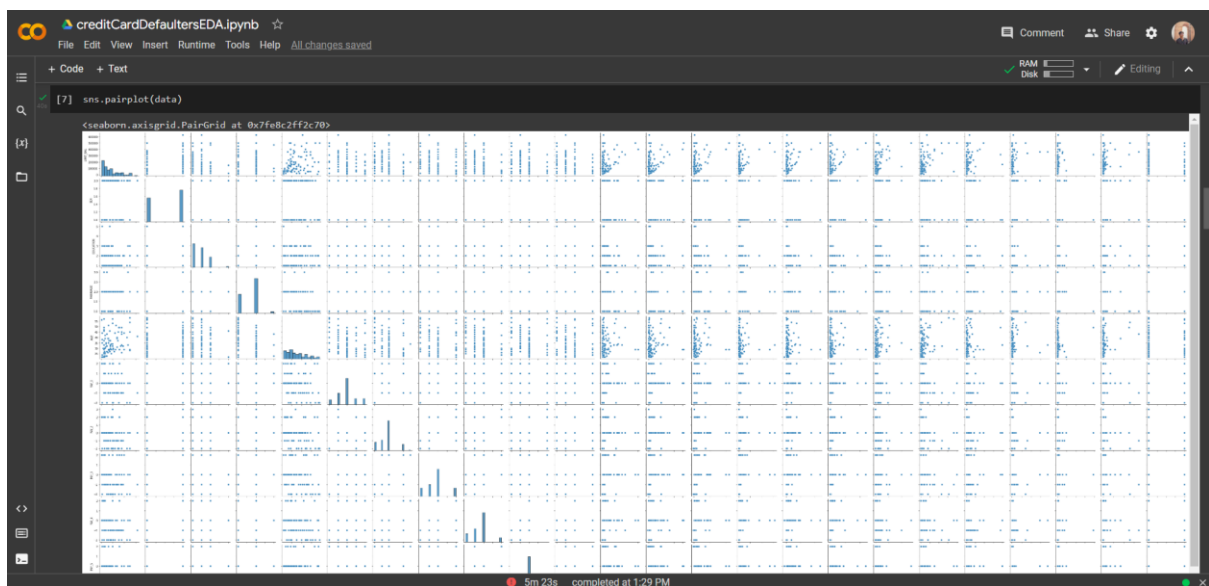
Below the code cell, the output of `data.info()` is displayed:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 24 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   LIMIT_BAL                            600 non-null    int64
1   SEX                                  600 non-null    int64
2   EDUCATION                           600 non-null    int64
3   MARRIAGE                            600 non-null    int64
4   AGE                                  600 non-null    int64
5   PAY_0                               600 non-null    int64
6   PAY_2                               600 non-null    int64
7   PAY_3                               600 non-null    int64
8   PAY_4                               600 non-null    int64
9   PAY_5                               600 non-null    int64
10  PAY_6                               600 non-null    int64
11  BILL_AMT1                           600 non-null    int64
12  BILL_AMT2                           600 non-null    int64
13  BILL_AMT3                           600 non-null    int64
14  BILL_AMT4                           600 non-null    int64
15  BILL_AMT5                           600 non-null    int64
16  BILL_AMT6                           600 non-null    int64
17  PAY_AMT1                            600 non-null    int64
18  PAY_AMT2                            600 non-null    int64
19  PAY_AMT3                            600 non-null    int64
20  PAY_AMT4                            600 non-null    int64
21  PAY_AMT5                            600 non-null    int64
22  PAY_AMT6                            600 non-null    int64
23  default payment next month          600 non-null    int64
dtypes: int64(24)
memory usage: 112.6 KB
```

5.0 Bench marking

According to my search on internet, all the models developed has not covered all useful features. I included all the useful features which will help to predict dependent variable. So, the model developed here will have more accuracy and prediction will be accurate almost.

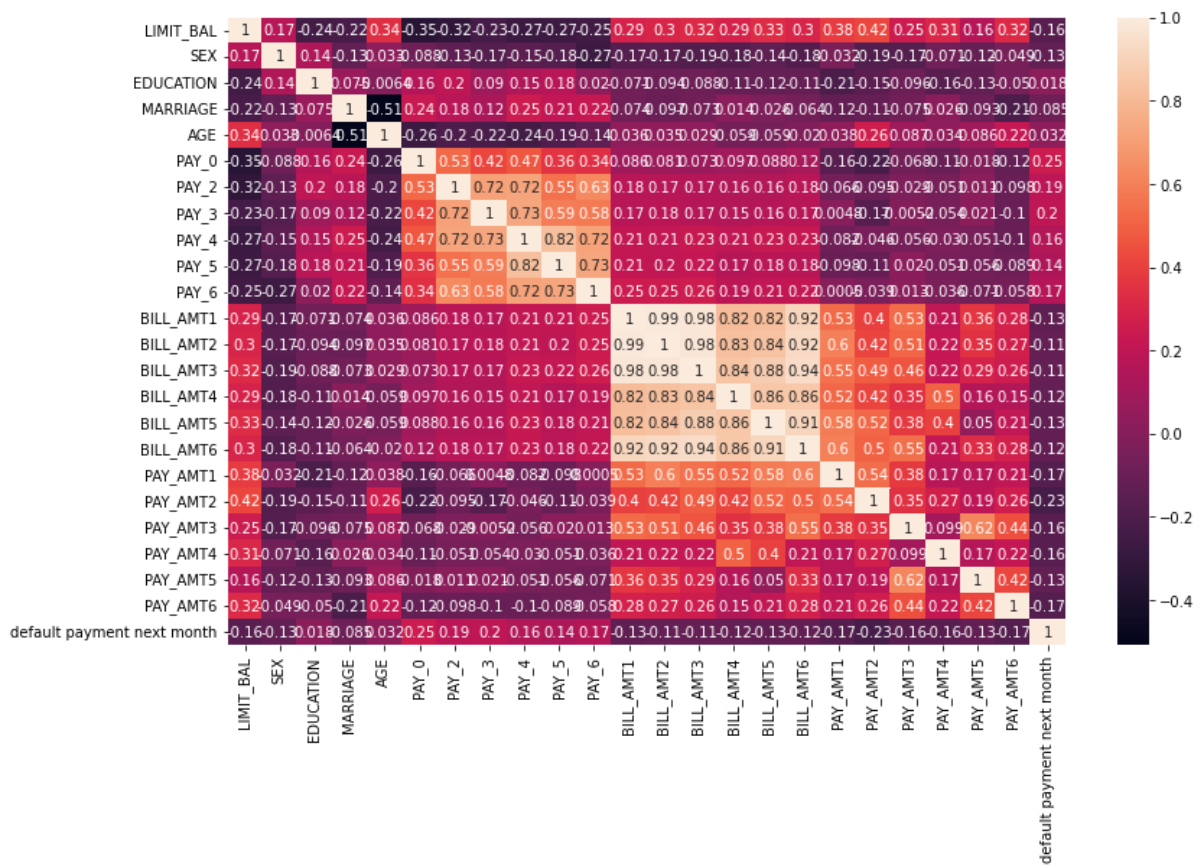
Pair-plot:



From the pair-plot above, we can see that there is some relationship between the feature columns. To confirm that we'd plot a correlation heatmap.

Correlation heatmap:

```
plt.figure(figsize=(13,8))
sns.heatmap(data.corr(), annot=True )
```



From the correlation heatmap above, it can be seen that there are some relationships between the feature columns, they are not entirely independent.

But in this scenario, there is a correlation because a customer who was not able to pay the bill for 1 month was again not able to pay it for the subsequent months and hence the correlation.

Again, for the bill amount column, the same has happened. If the customer was not able to pay the bill, then the bill amount almost remained the same, or if the customer was able to pay then the bill amount got reduced.

Here dropping the columns shall result in the loss of bill and payment history data. So, we don't need to drop any column although there is a correlation.

6.0 Applicable Patents

-> Patent of Tech/Software/Framework etc to use in your Product/Service idea.

- Method and system program for predicting credit card user default based on BP_Adaboost model with patent no/id WO2018090657A1 can be used to analyse and build our model.
- Patents on ML algorithms developed.
- Patents for the app which could be developed for the model.

7.0 Applicable Regulations

-> Government and environmental regulations imposed by countries.

- Data protection and privacy regulations (Customers). Laws related to privacy for collecting data of users from bank
- Govt Regulations for small business
- Patents on ML algorithms developed
- Review of existing work authority regulations
- Must be responsible with the scraped data: It is quite essential to protect the privacy and intention with which the data was extracted.

8.0 Applicable Constraints

-> need for space, budget, expertise.

- Continuous data collection and maintenance by banks
- Confidential credit cards data to be obtained to train the model
- Use of python, and its libraries for data validation and tuning purposes
- Use of frameworks like Flask for creating web application in deployment process
- Use of databases to insert data into it and cloud platforms to save the model trained and pushing app into cloud
- Machine learning algorithms for predictions
- Use of visualization tools like Tableau/ PowerBi for report and dashboard generations

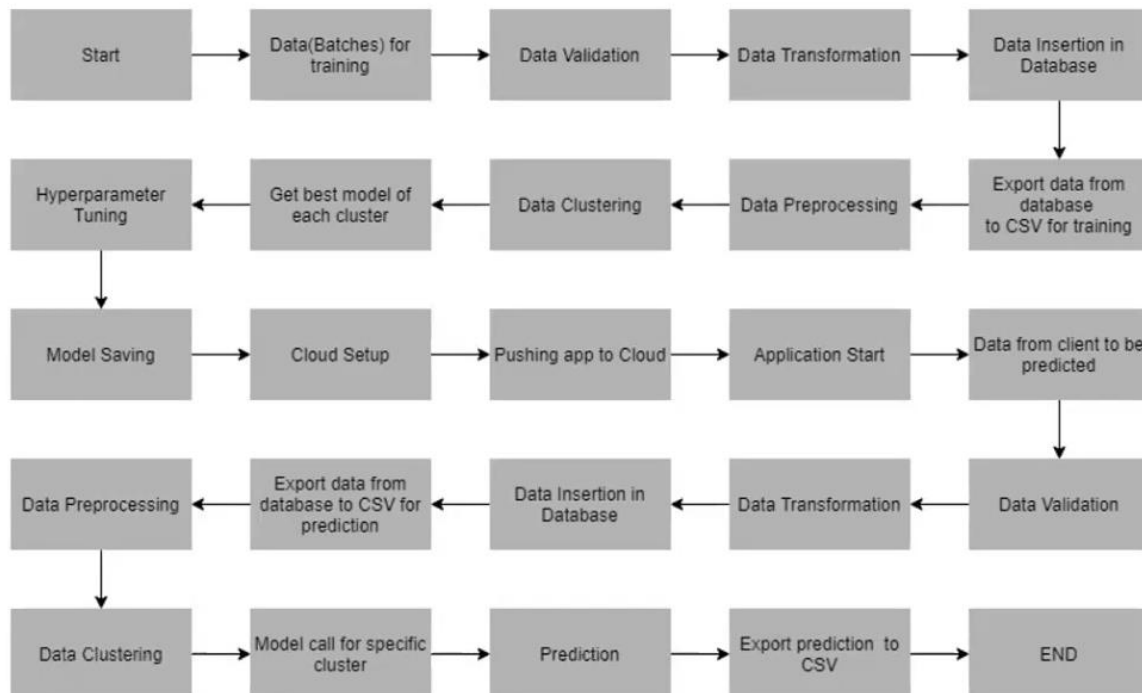
9.0 Business Opportunity

The opportunity can be seen in all financial sectors that give credit cards to users so that they get income through the interests on their credit.

So, if we build a model and help them analyse among users who are willing to pay bills and dues and who are not willing to pay. So, banks can come up with new strategies and they can change to plans and budgets accordingly.

10.0 Concept behind Model Generation

The architecture how the process goes, and the application flow is shown below:



- After getting data from client, we perform different sets of validation on the given dataset. Like Name validation, Validating the name and number of columns, datatype of columns, null values in columns etc.,
- Now a database must be created or open a connection if database is already created. And create a table in database and insert the files in that table.
- Insert only that data which passed the data validation part.

“The above is part is not necessary here as input file is already with us to train our model. From this process we actually generate final input csv file for our model training.”

MODEL TRAINING:

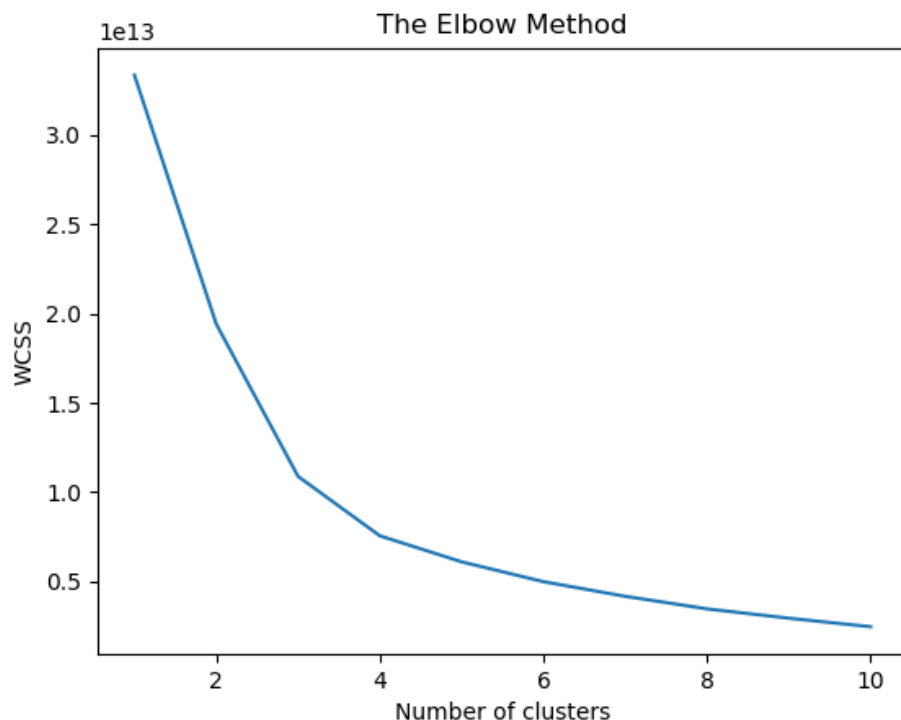
1) The data present in the input csv file will be used for the model training.

2) Data Pre-processing

- Check for null values in the columns. If present, impute the null values using the categorical imputer.
- Scale the numeric values using the standard scaler.
- Check for correlation.

3) Clustering

- Kmeans algorithm is used to create clusters in the pre-processed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using “KneeLocator” function.



- The idea behind clustering is to implement different algorithms.
- The Kmeans model is trained over pre-processed data, and the model is saved for further use in prediction.

4) Model Selection

- After the clusters have been created, we find the best model for each cluster.
- We are using two algorithms, “Naïve Bayes” and “XGBoost”. For each cluster, both the algorithms are passed with the best parameters derived from gridsearch.
- We calculate the AUC scores for both models and select the model with the best score.
- Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

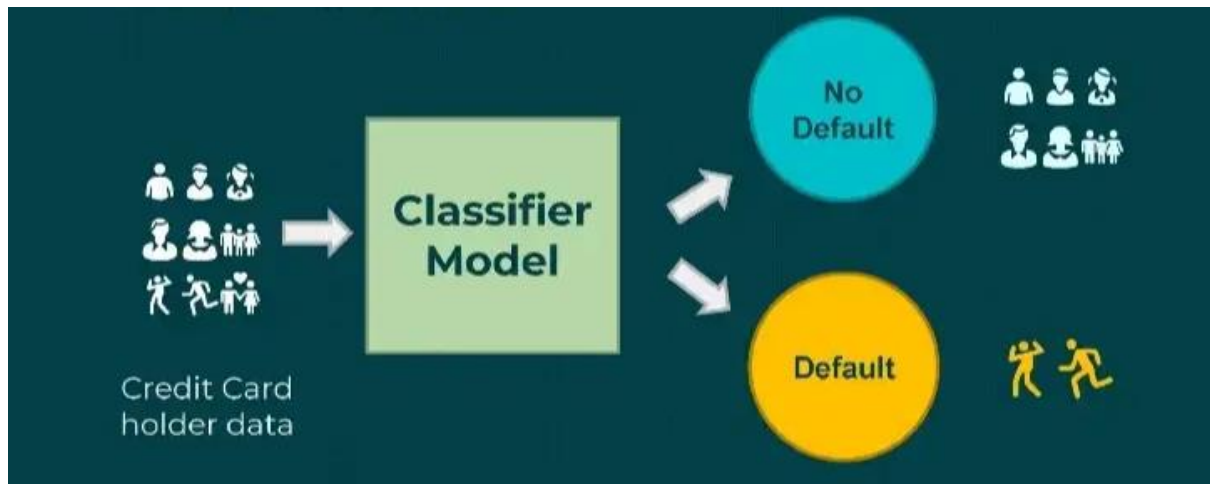
10.1 Concept Development

It is a machine learning-based model, and the best results will be extracted and presented in accordance with the datasets offered for testing. This allowed us to use feature selection to filter out the irrelevant data and extract the important information.

After performing clustering by using Kmeans algorithm for the created clusters I used two algorithms, to find the best model for each cluster. They are “Naïve Bayes” and “XGBoost”. For each cluster, both the algorithms are passed with the best parameters derived from Gridsearch.

11.0 Final Report Prototype:

The product takes the following functions to perfect and provide a good result.



Back-end:

Model or Webapp Development: This must be done before releasing the service. A lot of manual supervised machine learning must be performed to optimize the automated tasks.

As shown in flow chart above:

- Performing EDA to realize the dependent and independent features.
- Algorithm training and optimization must be done to minimize overfitting of the model and hyperparameter tuning.

Front-end:

- Different user interface: The user must be given many options to choose from in terms of parameters. This can only be optimized after a lot of testing and analysis all the edge cases,

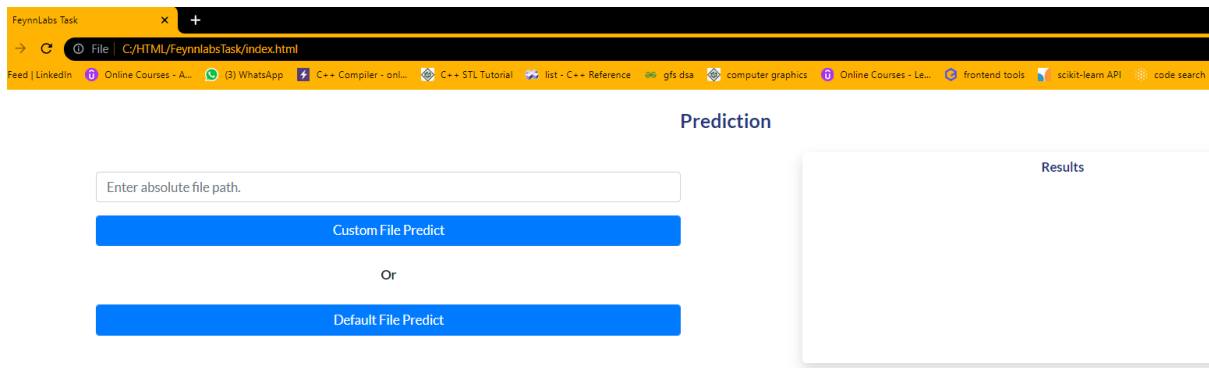
- Interactive visualization the data extracted from the trained models will return raw and inscrutable data. This must be present in an aesthetic and an “easy to read” style.
- Feedback system: A valuable feedback system must be developed to understand the customer’s needs that have not been met. This will help us train the models constantly.

12.0 Product details

I created a WEB APPLICATION using flask framework and the model trained using dataset is already saved and deployed on the **Heroku cloud platform.**

DEPLOYMENT TO HEROKU

- Go to Heroku.com and create an account and login
- Click on new to create a new app
- Give the name of the app and click ‘create app’
- After app creation, the ‘deploy’ section has all the deployment steps mentioned. We’ll download and install the Heroku CLI from the Heroku website:
<https://devcenter.heroku.com/articles/heroku-cli>
- After installing the Heroku CLI, open command prompt window and navigate to project folder
- After deployment, Heroku gives you the URL to hit the web API



- After giving input the path of the data for which we want to predict by clicking the following options we will shown results at right side.

13.0 Code Implementation

The code is implemented uploaded in my GitHub account and find the link below:

GitHub Link:

<https://github.com/madhavavvs/intern-feynnlabs-task-0>

14.0 Conclusion

The ML model created here will be very useful in market for both customers and banks as we are using optimized and hyper tuned ML algorithm our model gives good result with best accuracy. And customers can be saved from penalties and even court issues as shown in past picture.

References

- [1] Randhawa, Kuldeep, et al (2018) Credit Card Fraud Detection Using AdaBoost and Majority Voting. IEEE Access, vol. 6, pp 14277-14284. doi10.1109/access.2018.2806420
- [2] Bou, S., T. Amagasa, and H. Kitagawa. Keyword search with path-based filtering over XML streams in Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium 2014. IEEE.
- [3] https://www.researchgate.net/publication/356563799_Credit_Card_Fraud_Detection_using_Machine_Learning_Algorithms
- [4] <https://www.hindawi.com/journals/complexity/2021/6618841/>
- [5] Devi Meenakshi, Janani, Gayatri (2019). Credit card fraud detection using Random Forest. International Research Journal of Engineering and Technology (IRJET). Vol 6(3).
- [6] <https://www.slideshare.net/alexpnt/default-credit-card-prediction>
- [7] <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
