INTERNSHIP PROJECT REPORT

CREDIT CARD FRAUD DETECTION USING ML

Intern Detail

Name of Project: Credit Card Fraud Detection using ML

Name of Intern: ANJALI PATEL

Intern's Ph. no.: 7879246300

Intern's Add.: Plot no – 04 , phase 2 , Vip Nagar , Rishali , Bhilai,490006

Name of College: Chhattisgarh Swami Vivekananda Technical University

College's Add.: Newai, P.O.-Newai, District- Durg (Chhattisgarh), PIN-491107

Branch: CSE – DATA SCIENCE

Project Submission Date: 11/09/2023

Table of content

S.No.	Headings	Page No.
1	Project Application	1
2	List of hardware and software used	2
3	Flow chart	3
4	Project Description	4
5	Code	7
6	References	13

- A project to develop a machine learning model to detect fraudulent credit card transactions.
- There are a number of machine learning techniques that can be used to detect credit card fraud. Some of the most common techniques include:
- Logistic regression
- Support vector machines
- Decision trees
- Random forests
- Neural networks

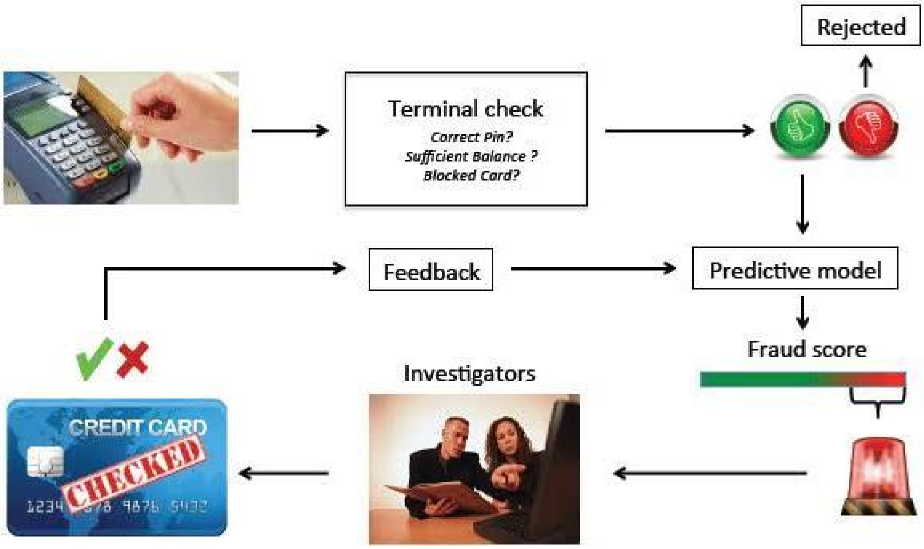


Figure: Fraud Detection Process

List of Software:

- Python Based Software
- Example : - Anaconda , Jupyter Notebook , Google Colab etc.

List of Hardware:

- OS – Windows 7, 8 , 10 and 11 (32 and 64 bit)
- RAM – 4GB

Basically, there are five steps in Credit Card Fraud Detection process
Step 1: Dataset (Credit Card Data)
Step 2: Data Pre-Processing
Step 3: Data Analysis
Step 4: Train Test split
Step 5: Logistics Regression Model
Step 6: Evaluation

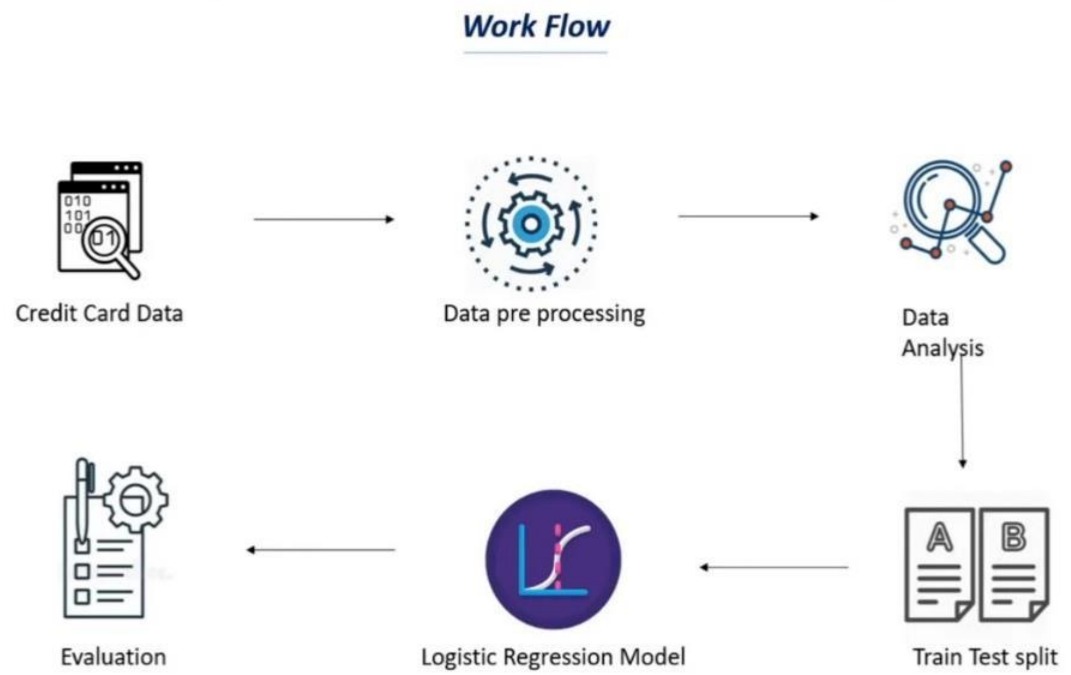


Figure: Work Flow of System

PURPOSE OF THE PROJECT

We propose a Machine learning model to detect fraudulent credit card activities in online financial transactions.

4.1 Overview

We propose a Machine learning model to detect fraudulent credit card activities in online financial transactions.Analyzing fake transactions manually is impracticable due to vast amounts of data and its complexity. However, adequately given informative features, could make it is possible using Machine Learning. This hypothesis will be explored in the project. To classify fraudulent and legitimate credit card transaction by supervised learning Algorithm such as Logistic Regression . To help us to get awareness about the fraudulent and without loss of any financially.

4.2 Dataset Description

The dataset contains transactions made by credit cards in September 2013 by European cardholders.This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Proposed Method

The proposed techniques emphasizes on detecting Credit Card Fraudulent transactions whether it is a genuine/nonfraud or a fraud transaction and the approaches used to separate fraud and non-fraud are KNN, Decision Tree, Logistic regression, Random forest and Finally we will detect credit card frauds.

The system architecture has following steps:

1. Import of Necessary Packages

2.Read the Dataset

3.Exploratory Data Analysis i.e. finding null values, duplicate values etc.

4.Selecting Features (X) and the Target (y) columns

5.Train Test Split will split the whole dataset into train and test data

6.Build the model i.e. Training the model

7.Test the model i.e. Model prediction

8. Evaluation of the system i.e. Accuracy score, F1- score etc.

Importing the Dependencies

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
# loading the dataset to a Pandas DataFrame
credit_card_data = pd.read_csv('creditcard.csv')

# first 5 rows of the dataset
credit_card_data.head()

credit_card_data.tail()

# dataset informations
credit_card_data.info()

# checking the number of missing values in each column
credit_card_data.isnull().sum()

# distribution of legit transactions & fraudulent transactions
credit_card_data['Class'].value_counts()

# separating the data for analysis
legit = credit_card_data[credit_card_data.Class == 0]
fraud = credit_card_data[credit_card_data.Class == 1]

print(legit.shape)
print(fraud.shape)

# statistical measures of the data
legit.Amount.describe()

fraud.Amount.describe()

# compare the values for both transactions
credit_card_data.groupby('Class').mean()
```

Under Sampling

Build a sample dataset containing similar distribution of normal transactions and Fraudulent Transactions

```
Number of Fraudulent Transactions --> 492
```

```
legit_sample = legit.sample(n=492)
```

Concatenating two DataFrames

```
new_dataset = pd.concat([legit_sample, fraud], axis=0)
new_dataset.head()

new_dataset.tail()

new_dataset['Class'].value_counts()

new_dataset.groupby('Class').mean()
```

Splitting Dataset into Features and Targets

```
X = new_dataset.drop(columns='Class', axis=1)
Y = new_dataset['Class']

print(X)

print(Y)
```

Split the data into Training data and Testing data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

print(X.shape, X_train.shape, X_test.shape)
```

Model Training

Logistic Regression

```
model = LogisticRegression()

# training the Logistic Regression Model with Training Data
model.fit(X_train, Y_train)
```

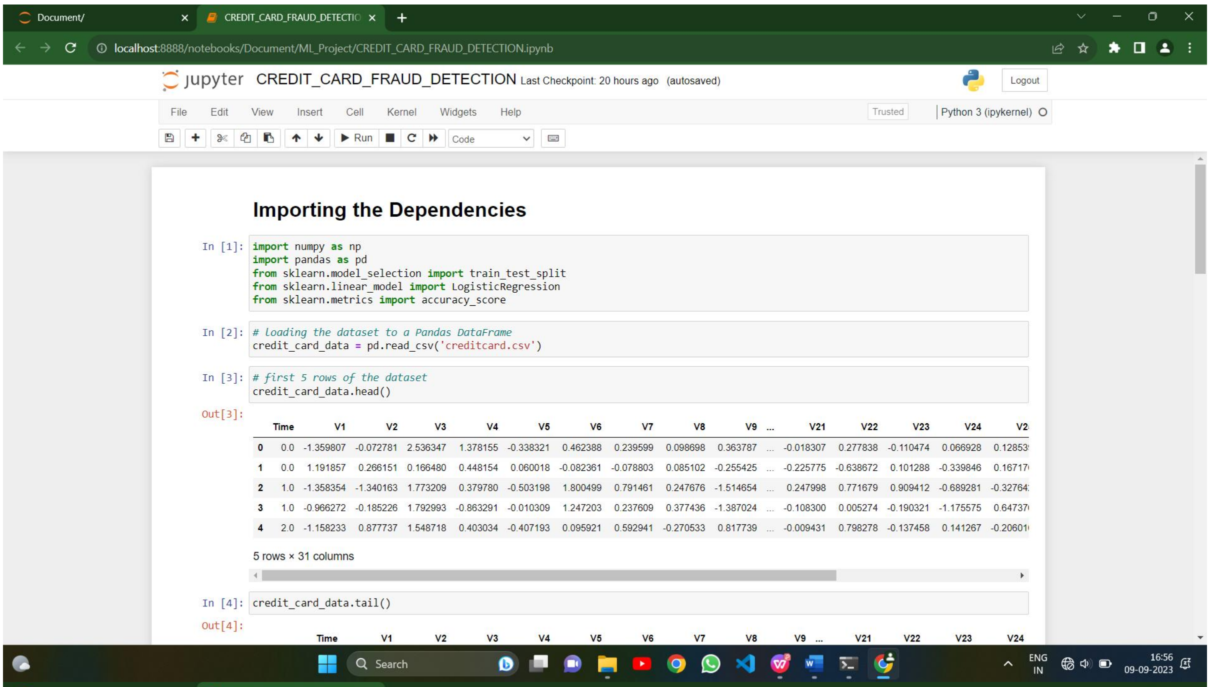
Model Evaluation

Accuracy Score

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

print('Accuracy on Training data : ', training_data_accuracy)
```

```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

print('Accuracy score on Test Data : ', test_data_accuracy)
```

IMPLIMENTATION AND SCREENSHOTS AND IMAGES OF THE RUNNING PROJECTS


Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:56 09-09-2023

Importing the Dependencies

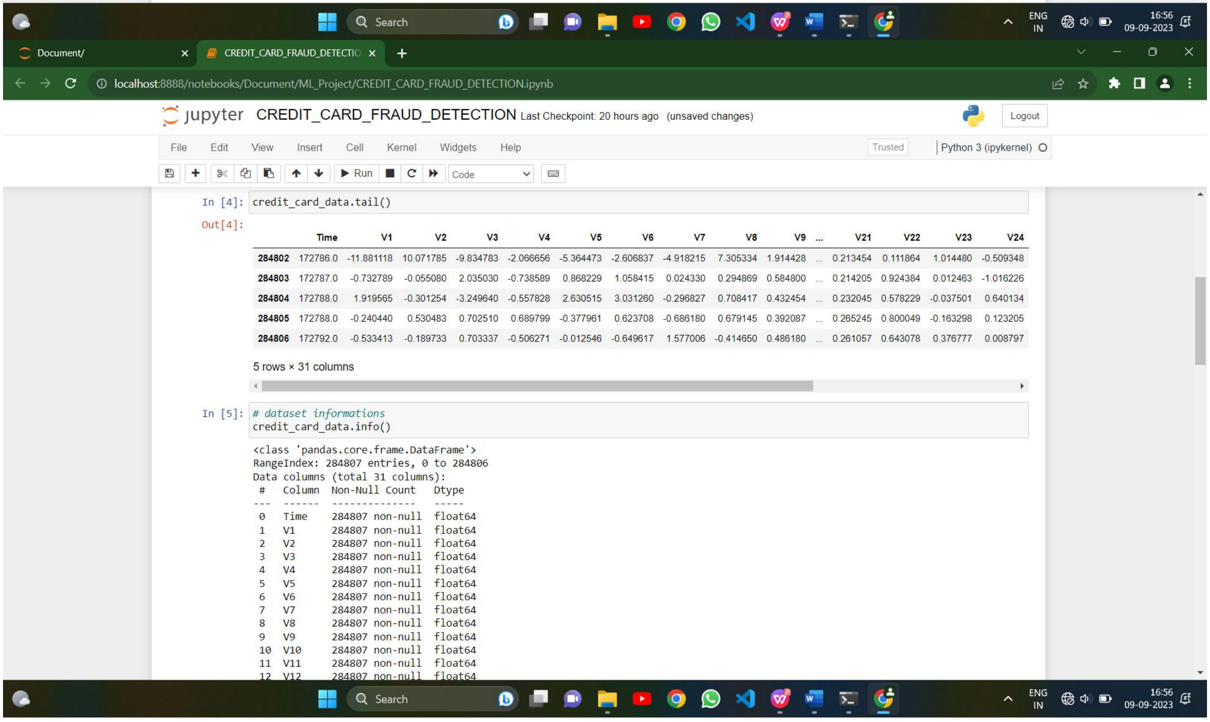
```
In [1]: import numpy as np
        import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import accuracy_score

In [2]: # loading the dataset to a Pandas DataFrame
        credit_card_data = pd.read_csv('creditcard.csv')

In [3]: # first 5 rows of the dataset
        credit_card_data.head()

Out[3]:
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601

5 rows × 31 columns

```
In [4]: credit_card_data.tail()

Out[4]:
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24

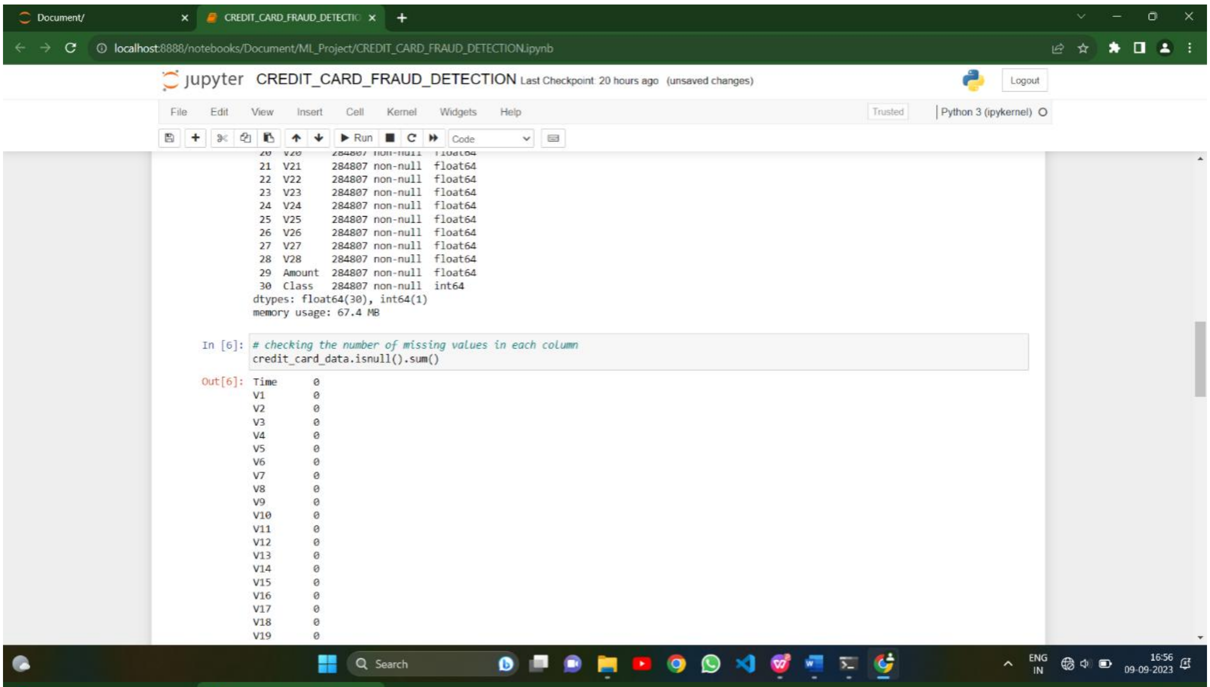

Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:56 09-09-2023

```
In [4]: credit_card_data.tail()

Out[4]:
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.123205
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.008797

5 rows × 31 columns

```
In [5]: # dataset informations
        credit_card_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
```

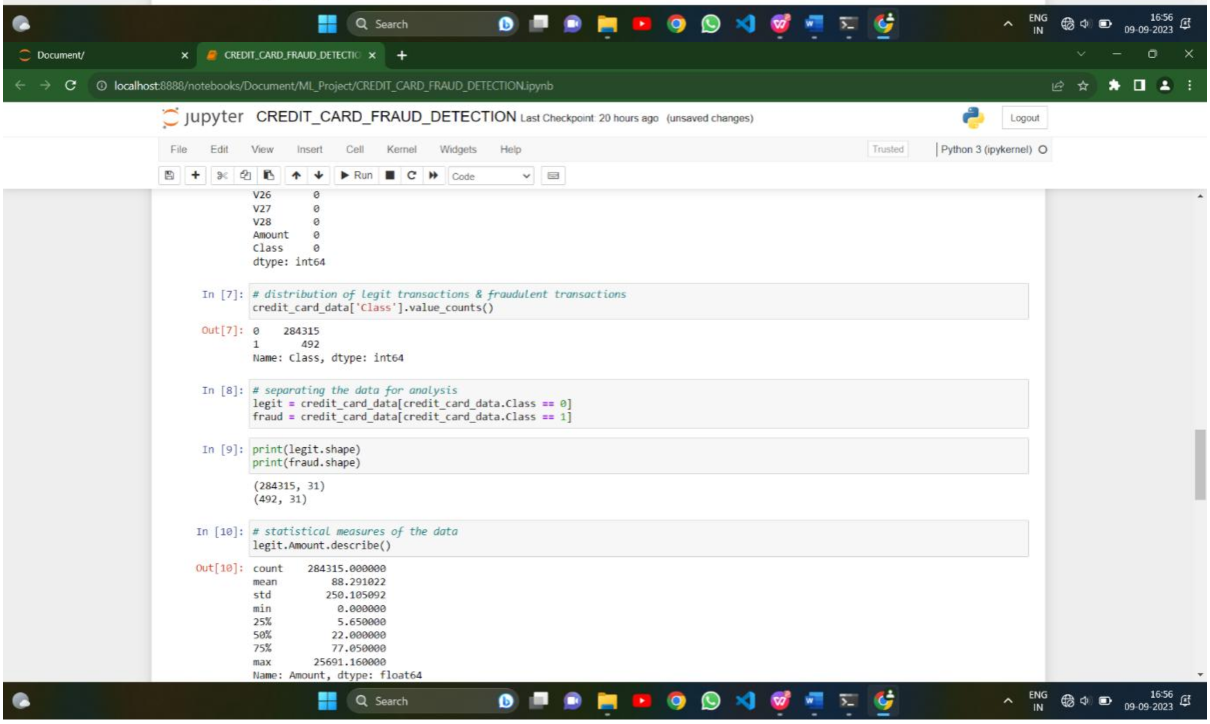


Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:56 09-09-2023

```
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

```
In [6]: # checking the number of missing values in each column
        credit_card_data.isnull().sum()
```

```
Out[6]: Time      0
        V1        0
        V2        0
        V3        0
        V4        0
        V5        0
        V6        0
        V7        0
        V8        0
        V9        0
        V10       0
        V11       0
        V12       0
        V13       0
        V14       0
        V15       0
        V16       0
        V17       0
        V18       0
        V19       0
```



Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:56 09-09-2023

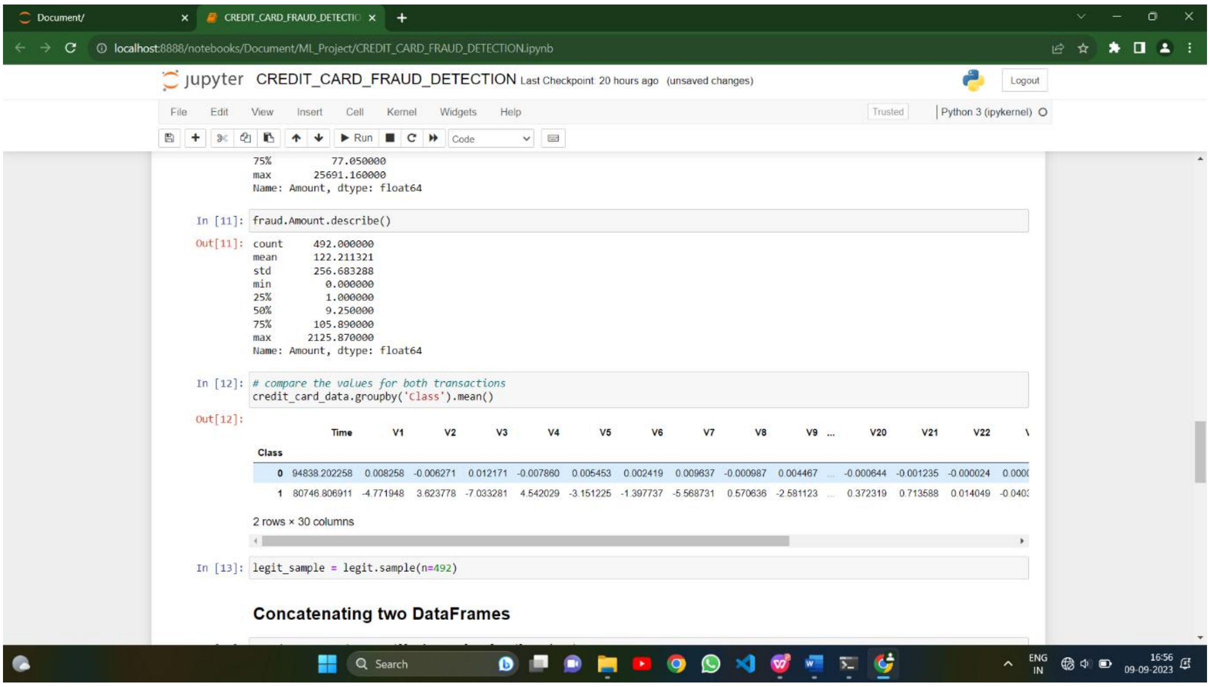
```
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

```
In [7]: # distribution of legit transactions & fraudulent transactions
        credit_card_data['Class'].value_counts()

Out[7]: 0    284315
        1       492
        Name: Class, dtype: int64
```

```
In [8]: # separating the data for analysis
        legit = credit_card_data[credit_card_data.Class == 0]
        fraud = credit_card_data[credit_card_data.Class == 1]
```

```
In [9]: print(legit.shape)
        print(fraud.shape)

        (284315, 31)
        (492, 31)
```

```
In [10]: # statistical measures of the data
         legit.Amount.describe()

Out[10]: count    284315.000000
         mean         88.291022
         std         250.105092
         min           0.000000
         25%           5.650000
         50%          22.000000
         75%          77.050000
         max       25691.160000
         Name: Amount, dtype: float64
```


Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter
CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:56
09-09-2023

```
75%          77.050000
max       25691.160000
Name: Amount, dtype: float64
```

```
In [11]: fraud.Amount.describe()
```

```
Out[11]: count     492.000000
         mean      122.211321
         std       256.683288
         min         0.000000
         25%         1.000000
         50%         9.250000
         75%       105.890000
         max      2125.870000
         Name: Amount, dtype: float64
```

```
In [12]: # compare the values for both transactions
         credit_card_data.groupby('Class').mean()
```

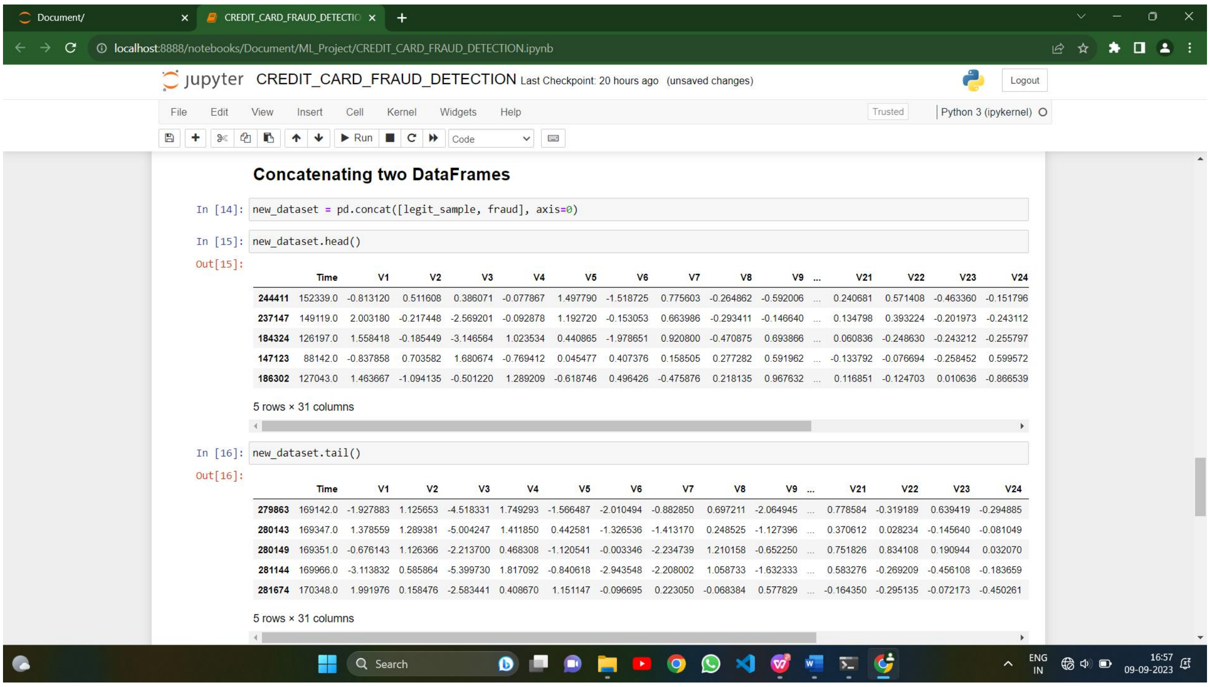
Out[12]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	V
Class															
0	94838.202258	0.008258	-0.006271	0.012171	-0.007860	0.005453	0.002419	0.009637	-0.000987	0.004467	...	-0.000644	-0.001235	-0.000024	0.000
1	80746.806911	-4.771948	3.623778	-7.033281	4.542029	-3.151225	-1.397737	-5.568731	0.570636	-2.581123	...	0.372319	0.713588	0.014049	-0.040

2 rows × 30 columns

```
In [13]: legit_sample = legit.sample(n=492)
```

Concatenating two DataFrames


Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter
CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:57
09-09-2023

Concatenating two DataFrames

```
In [14]: new_dataset = pd.concat([legit_sample, fraud], axis=0)
```

```
In [15]: new_dataset.head()
```

Out[15]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
244411	152339.0	-0.813120	0.511608	0.386071	-0.077867	1.497790	-1.518725	0.775603	-0.264862	-0.592006	...	0.240681	0.571408	-0.463360	-0.151796
237147	149119.0	2.003180	-0.217448	-2.569201	-0.092878	1.192720	-0.153053	0.663986	-0.293411	-0.146640	...	0.134798	0.393224	-0.201973	-0.243112
184324	126197.0	1.558418	-0.185449	-3.146564	1.023534	0.440865	-1.978651	0.920800	-0.470875	0.693866	...	0.060836	-0.248630	-0.243212	-0.255797
147123	88142.0	-0.837858	0.703582	1.680674	-0.769412	0.045477	0.407376	0.158505	0.277282	0.591962	...	-0.133792	-0.076694	-0.258452	0.599572
186302	127043.0	1.463667	-1.094135	-0.501220	1.289209	-0.618746	0.496426	-0.475876	0.218135	0.967632	...	0.116851	-0.124703	0.010636	-0.866539

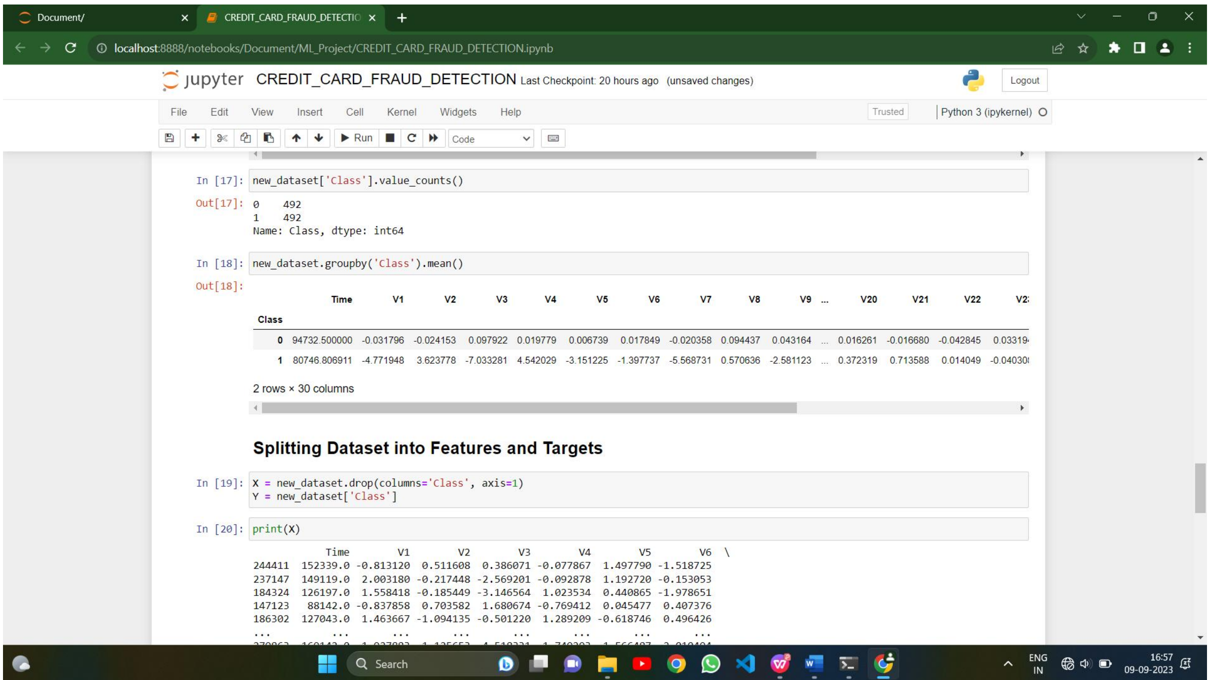
5 rows × 31 columns

```
In [16]: new_dataset.tail()
```

Out[16]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
279863	169142.0	-1.927883	1.125653	-4.518331	1.749293	-1.566487	-2.010494	-0.882850	0.697211	-2.064945	...	0.778584	-0.319189	0.639419	-0.294885
280143	169347.0	1.378559	1.289381	-5.004247	1.411850	0.442581	-1.326536	-1.413170	0.248525	-1.127396	...	0.370612	0.028234	-0.145640	-0.081049
280149	169351.0	-0.676143	1.126366	-2.213700	0.468308	-1.120541	-0.003346	-2.234739	1.210158	-0.652250	...	0.751826	0.834108	0.190944	0.032070
281144	169966.0	-3.113832	0.585864	-5.399730	1.817092	-0.840618	-2.943548	-2.208002	1.058733	-1.632333	...	0.583276	-0.269209	-0.456108	-0.183659
281674	170348.0	1.991976	0.158476	-2.583441	0.408670	1.151147	-0.096695	0.223050	-0.068384	0.577829	...	-0.164350	-0.295135	-0.072173	-0.450261

5 rows × 31 columns



Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter
Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:57
09-09-2023

CREDIT_CARD_FRAUD_DETECTION

In [17]:
```
new_dataset['Class'].value_counts()
```

Out[17]:
```
0    492
1    492
Name: Class, dtype: int64
```

In [18]:
```
new_dataset.groupby('Class').mean()
```

Out[18]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	V2:
Class															
0	94732.500000	-0.031796	-0.024153	0.097922	0.019779	0.006739	0.017849	-0.020358	0.094437	0.043164	...	0.016261	-0.016680	-0.042845	0.03319
1	80746.806911	-4.771948	3.623778	-7.033281	4.542029	-3.151225	-1.397737	-5.568731	0.570636	-2.581123	...	0.372319	0.713588	0.014049	-0.04030

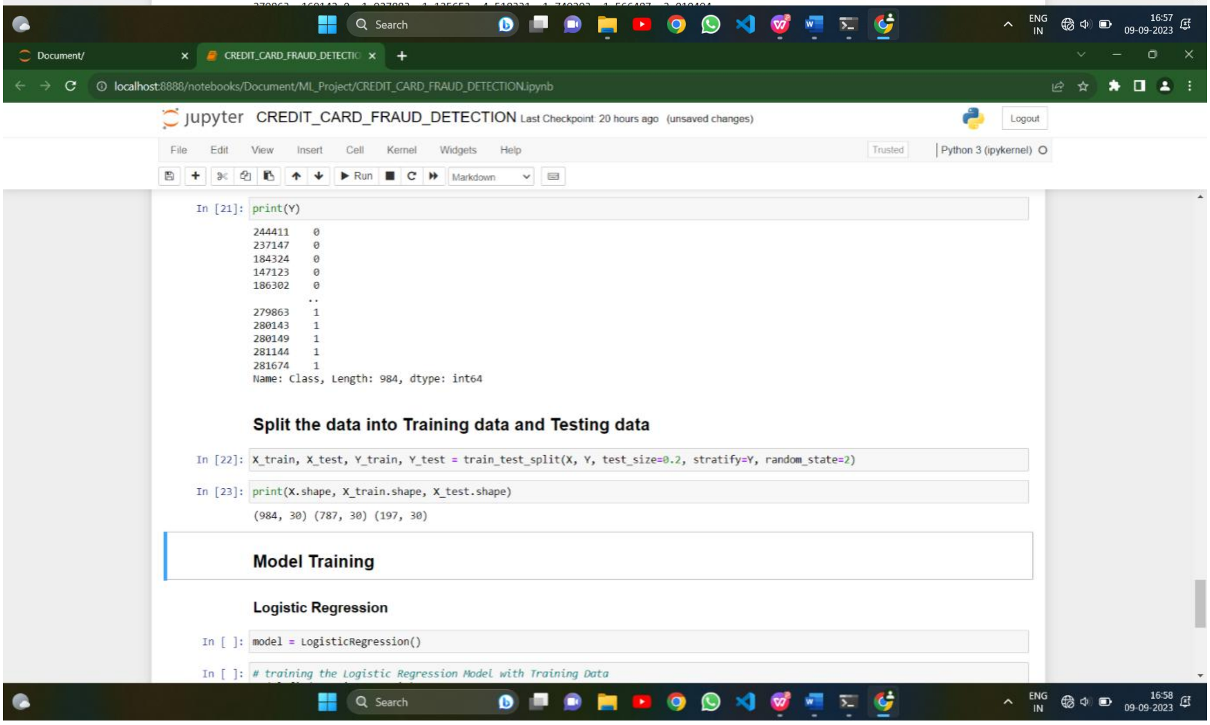
2 rows × 30 columns

Splitting Dataset into Features and Targets

In [19]:
```
X = new_dataset.drop(columns='Class', axis=1)
Y = new_dataset['Class']
```

In [20]:
```
print(X)
```

```
            Time        V1        V2        V3        V4        V5        V6  \
244411  152339.0 -0.813120  0.511608  0.386071 -0.077867  1.497790 -1.518725   
237147  149119.0  2.003180 -0.217448 -2.569201 -0.092878  1.192720 -0.153053   
184324  126197.0  1.558418 -0.185449 -3.146564  1.023534  0.440865 -1.978651   
147123   88142.0 -0.837858  0.703582  1.680674 -0.769412  0.045477  0.407376   
186302  127043.0  1.463667 -1.094135 -0.501220  1.289209 -0.618746  0.496426   
...          ...       ...       ...       ...       ...       ...       ...   
```



Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter
Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown
Search
ENG IN
16:58
09-09-2023

CREDIT_CARD_FRAUD_DETECTION

In [21]:
```
print(Y)
```

```
244411    0
237147    0
184324    0
147123    0
186302    0
         ..
279863    1
280143    1
280149    1
281144    1
281674    1
Name: Class, Length: 984, dtype: int64
```

Split the data into Training data and Testing data

In [22]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

In [23]:
```
print(X.shape, X_train.shape, X_test.shape)
```

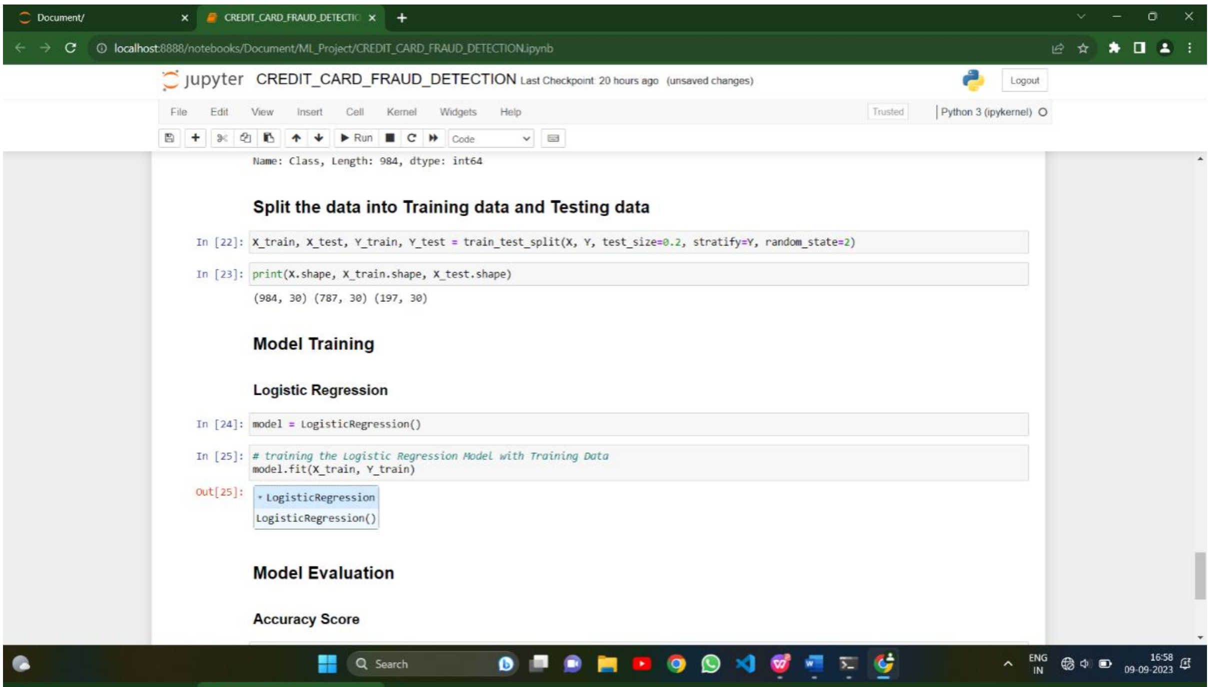
```
(984, 30) (787, 30) (197, 30)
```

Model Training

Logistic Regression

In [ ]:
```
model = LogisticRegression()
```

In [ ]:
```
# training the Logistic Regression Model with Training Data
```



Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:58 09-09-2023

```
Name: Class, Length: 984, dtype: int64
```

Split the data into Training data and Testing data

```
In [22]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

In [23]: print(X.shape, X_train.shape, X_test.shape)

(984, 30) (787, 30) (197, 30)
```

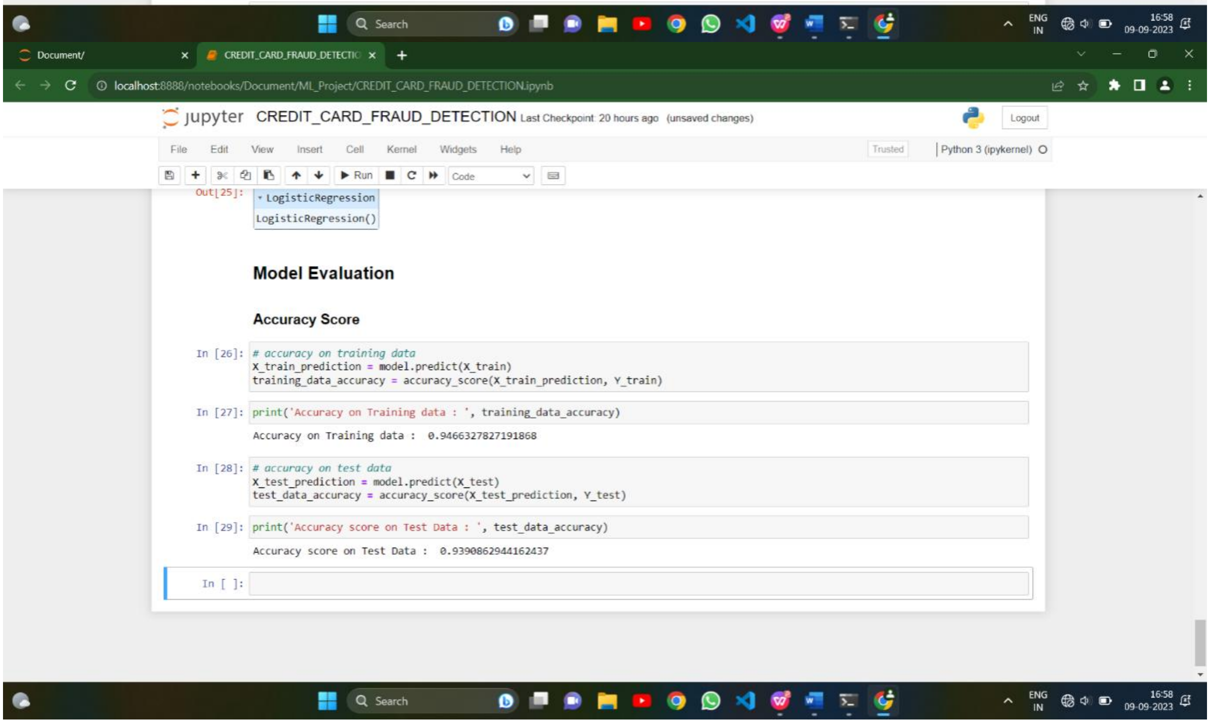
Model Training

Logistic Regression

```
In [24]: model = LogisticRegression()

In [25]: # training the Logistic Regression Model with Training Data
         model.fit(X_train, Y_train)

Out[25]: ▸ LogisticRegression
         LogisticRegression()
```

Model Evaluation

Accuracy Score



Document/
CREDIT_CARD_FRAUD_DETECTIO
localhost:8888/notebooks/Document/ML_Project/CREDIT_CARD_FRAUD_DETECTION.ipynb
jupyter CREDIT_CARD_FRAUD_DETECTION Last Checkpoint: 20 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
16:58 09-09-2023

Out[25]: ▸ LogisticRegression
LogisticRegression()

Model Evaluation

Accuracy Score

```
In [26]: # accuracy on training data
         X_train_prediction = model.predict(X_train)
         training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

In [27]: print('Accuracy on Training data : ', training_data_accuracy)

Accuracy on Training data :  0.9466327827191868

In [28]: # accuracy on test data
         X_test_prediction = model.predict(X_test)
         test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

In [29]: print('Accuracy score on Test Data : ', test_data_accuracy)

Accuracy score on Test Data :  0.9390862944162437

In [ ]:
```

List of References

[1] Machine Learning Group — ULB, Credit Card Fraud Detection (2018), Kaggle.

[2] R. J. Bolton and D. J. Hand. Unsupervised profiling methods for fraud detection. In conference of Credit Scoring and Credit Connol VII, Edinburgh. UK, Sept 5-7,2001.

[3] Quah, J. T. S., and Sriganesh, M. (2020). Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications, 35(4), 1721-1732.