CREDIT CARD FRAUD DETECTION WITH MACHINE LEARNING

Phase 5:

In this phase, the entire project is documented

Problem Definition:

Develop a machine learning model for credit card fraud detection that effectively identifies and prevents fraudulent transactions while minimizing false alarms. The objective is to protect both cardholders and financial institutions from financial losses by accurately distinguishing between legitimate and fraudulent transactions in real-time. The model should leverage historical transaction data, incorporate advanced anomaly detection and classification techniques, and continuously adapt to evolving fraud patterns

Design Thinking:

Data Collection: Data collection in credit card fraud detection involves gathering historical transaction data and relevant information that can be used to train and evaluate machine learning models for fraud detection.

Data Preprocessing: Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.

Exploratory Data Analysis: Explore the data to understand its characteristics, identify trends, and outliers.

Statistical Analysis: Perform statistical tests to analyze vaccine efficacy, adverse effects, and distribution across different populations.

Visualization: Create visualizations (e.g., bar plots, line charts, heatmaps) to present key findings and insights.

Data collection:

The first step is to collect data that is relevant to the product demand prediction task. Once the data is collected, it needs to be cleaned and prepared for modeling.

The given dataset:

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Data Preprocessing:

Data preprocessing is the initial step in preparing data for machine learning. It involves tasks like handling missing values, removing outliers, encoding variables, and scaling features. The goal is to clean, transform, and structure the data to improve its quality and make it suitable for training predictive models, ultimately enhancing their accuracy and performance.

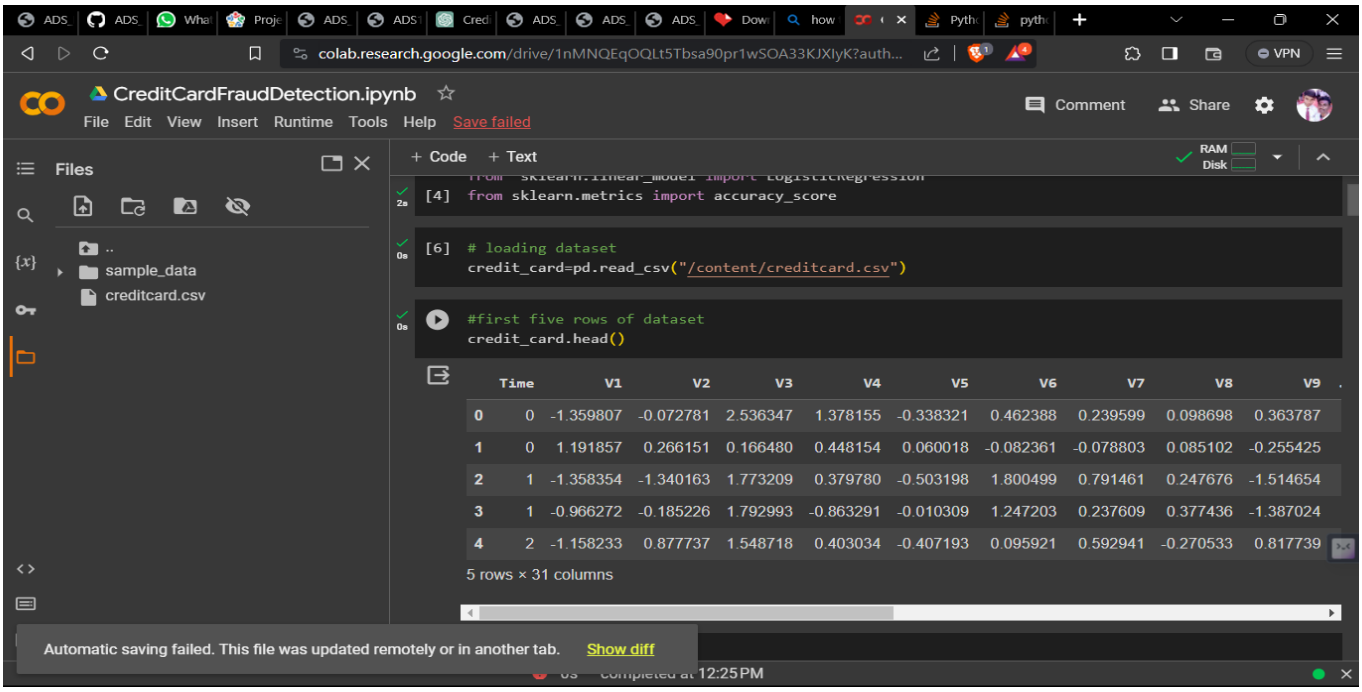
In context of creditcard fraudulent detection, we are going the loadbalance the dataset provided which consists of legitimate and illegitimate transactions.

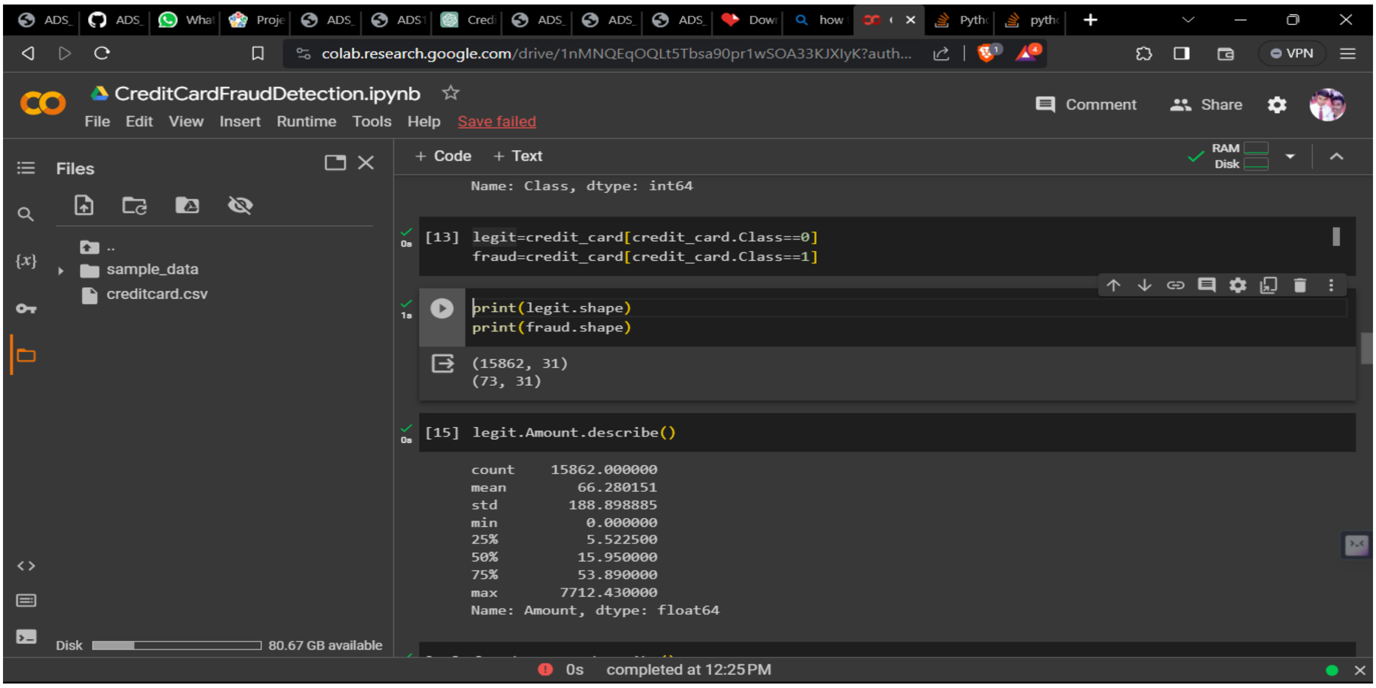
In data preprocessing of this step of this project we are going to balance the number of legitimate and illegitimate transactions by nearly bringing the transaction count equal in coding phase and then next phase is carried out

The dataset is cleaned

Exploratory Data Analysis:

Loading dataset

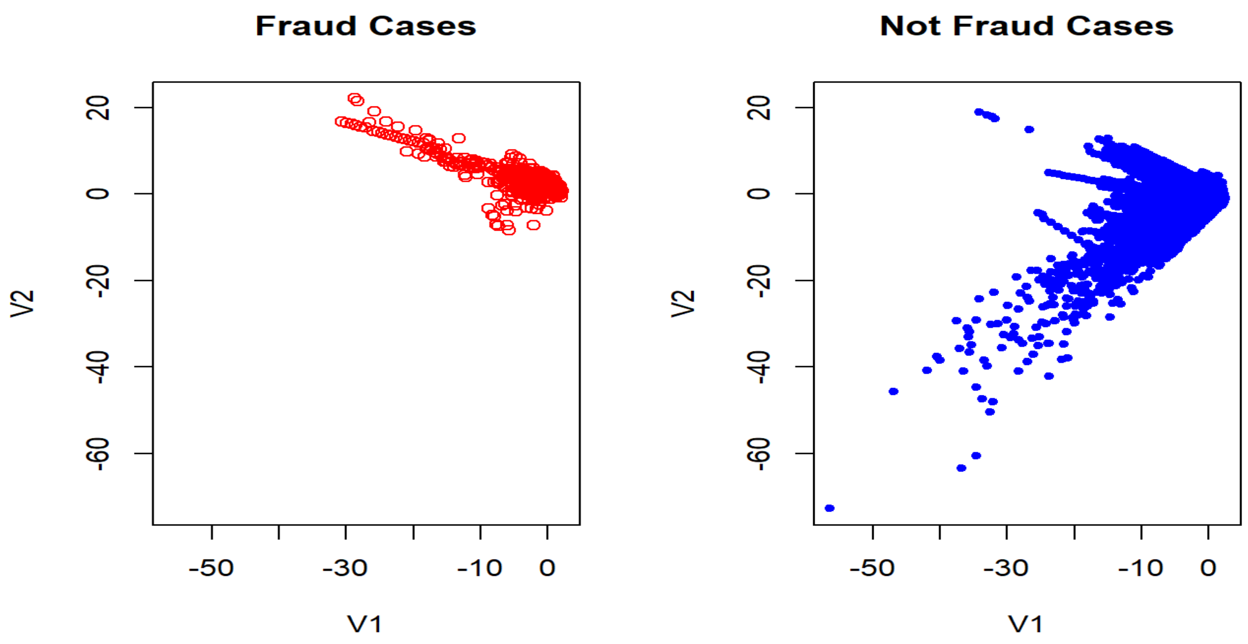




The above image represents the number of legit and fraud transactions.

Statistical Analysis

As per the statistics



Machine learning

Machine learning (ML) is a popular use of artificial intelligence since it automates the system and allows it to learn and improve from diverse experiences without being programmed. Computer programs can teach how to learn by giving them access to data and allowing them to utilize it for learning in ML. The learning process in ML begins with seeing the data through examples or instructions that humans offer; these observations enable ML to look for patterns in order to make the best predictions. Five different ML models were used to train the classifier and evaluate classification performance using the test dataset. These are discussed below.

Machine learning Techniques

There are several techniques in Machine Learning including

- Random Forest

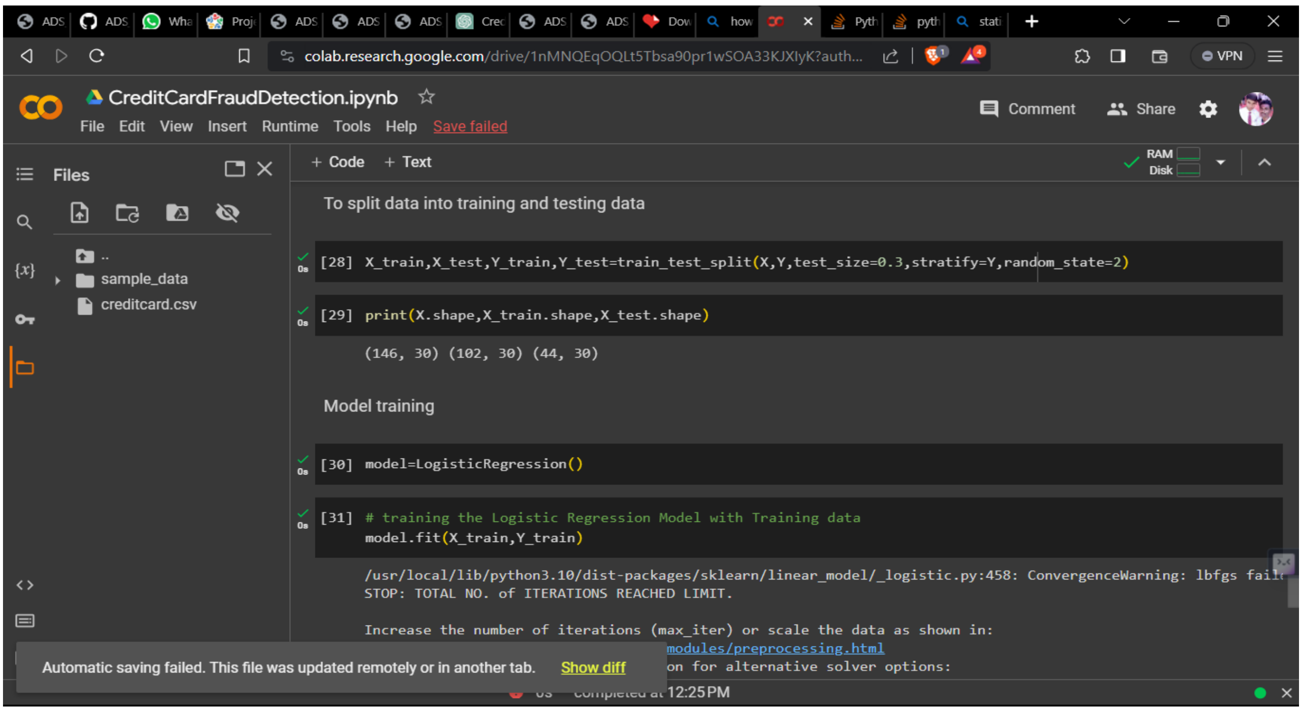
- Naive Bayes
- Decision Tree
- Logistic Regressions
- Support Vector Machine

Our choice of Machine Learning technique is Logistic Regression

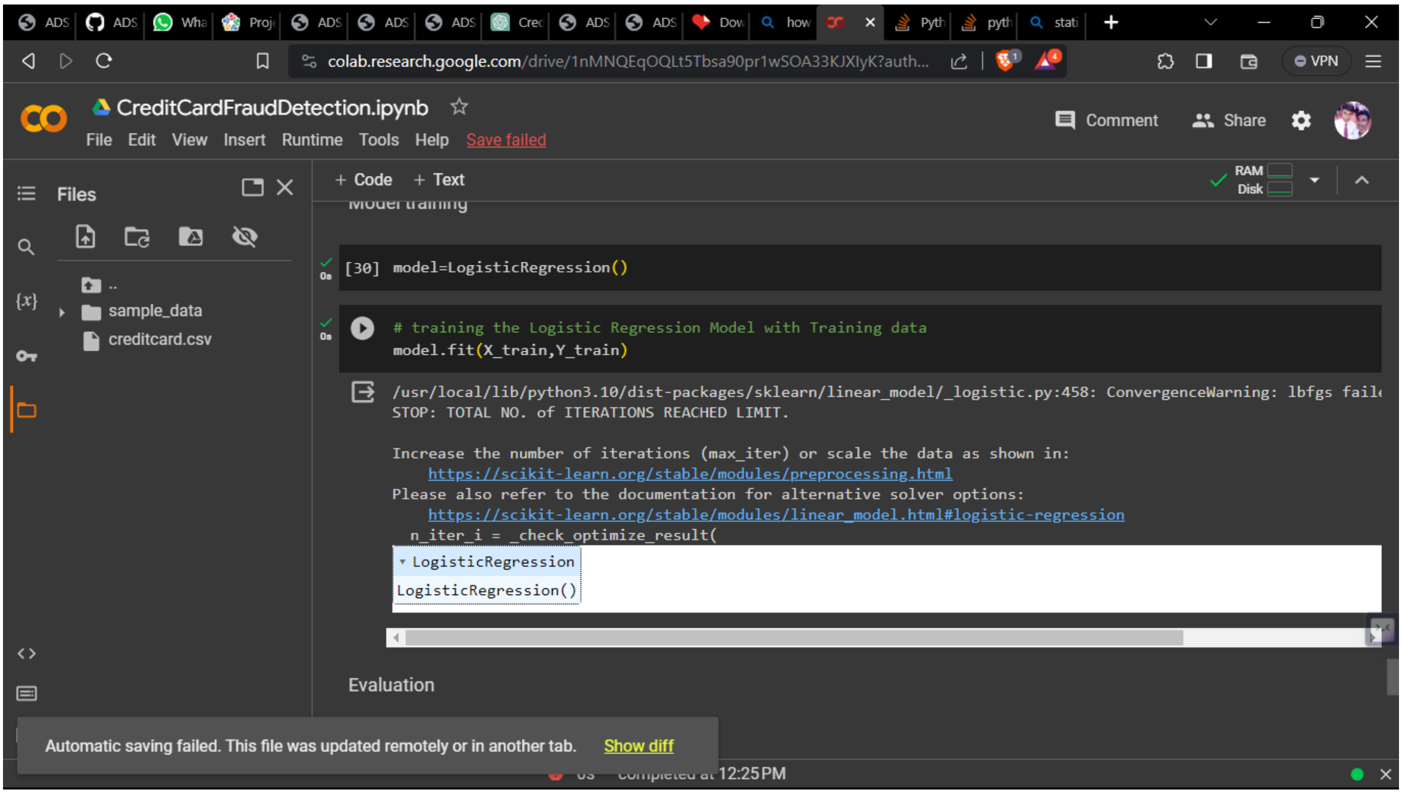
Logistic Regression

Logistic Regression is a statistical approach to data analysis in which one or more variables are utilized to determine the outcome. When the target variable is categorical, the optimum learning model to utilize is LR, which is the regression model that was used to estimate the likelihood of class members. Linear Regression uses a logistic function to estimate probabilities for the association between the categorical dependent variable and one or more independent variables.

Once the data is being pre-processed and got load balanced the entire data set is split into training and testing data and the model is trained successfully.



Model training:



Model Evaluation:

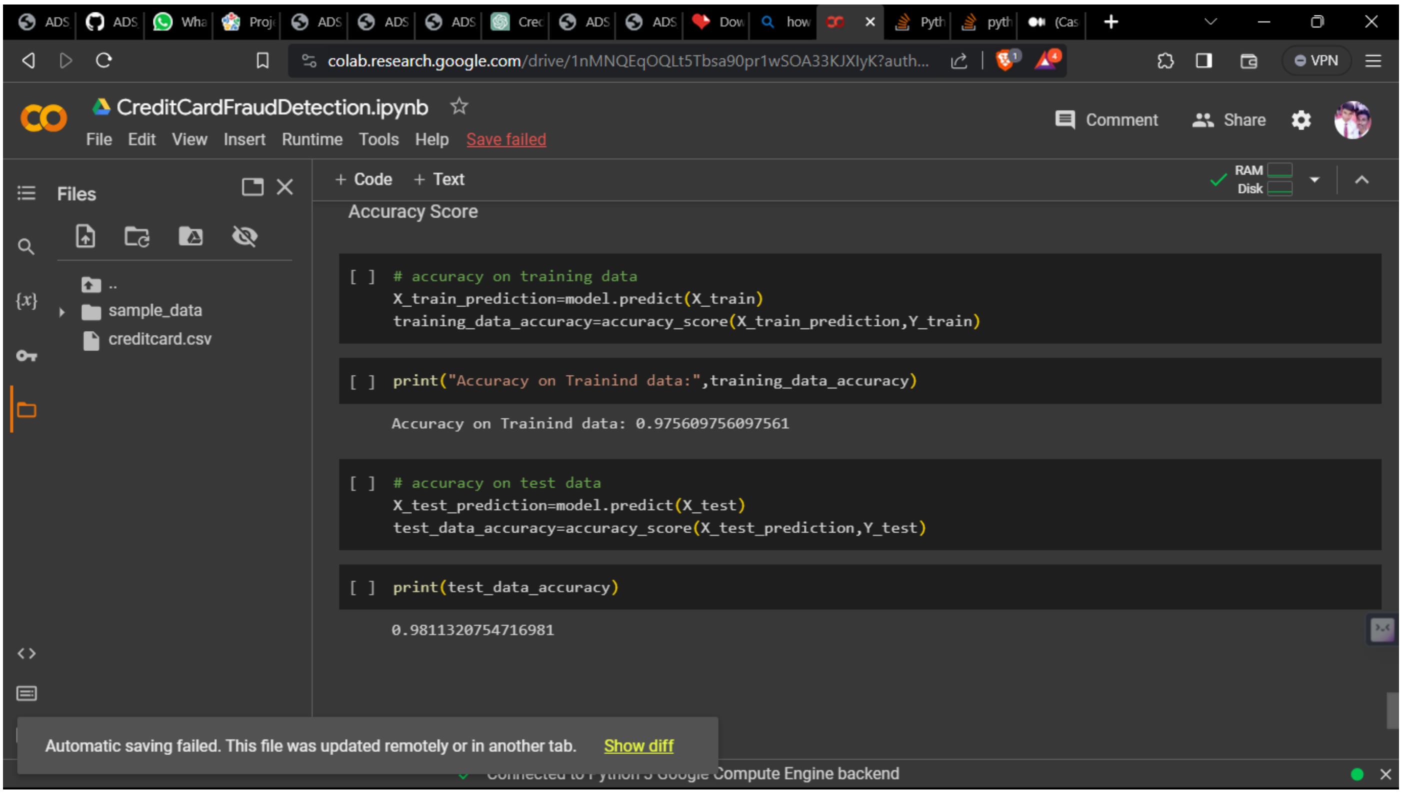
There are many evaluation metrics our choice is Accuracy score

Accuracy is a common performance metric used in machine learning, particularly for classification tasks. It measures the proportion of correctly classified instances out of the total instances in the dataset. In other words, it calculates how many predictions made by the model are correct.

The accuracy score is calculated as:

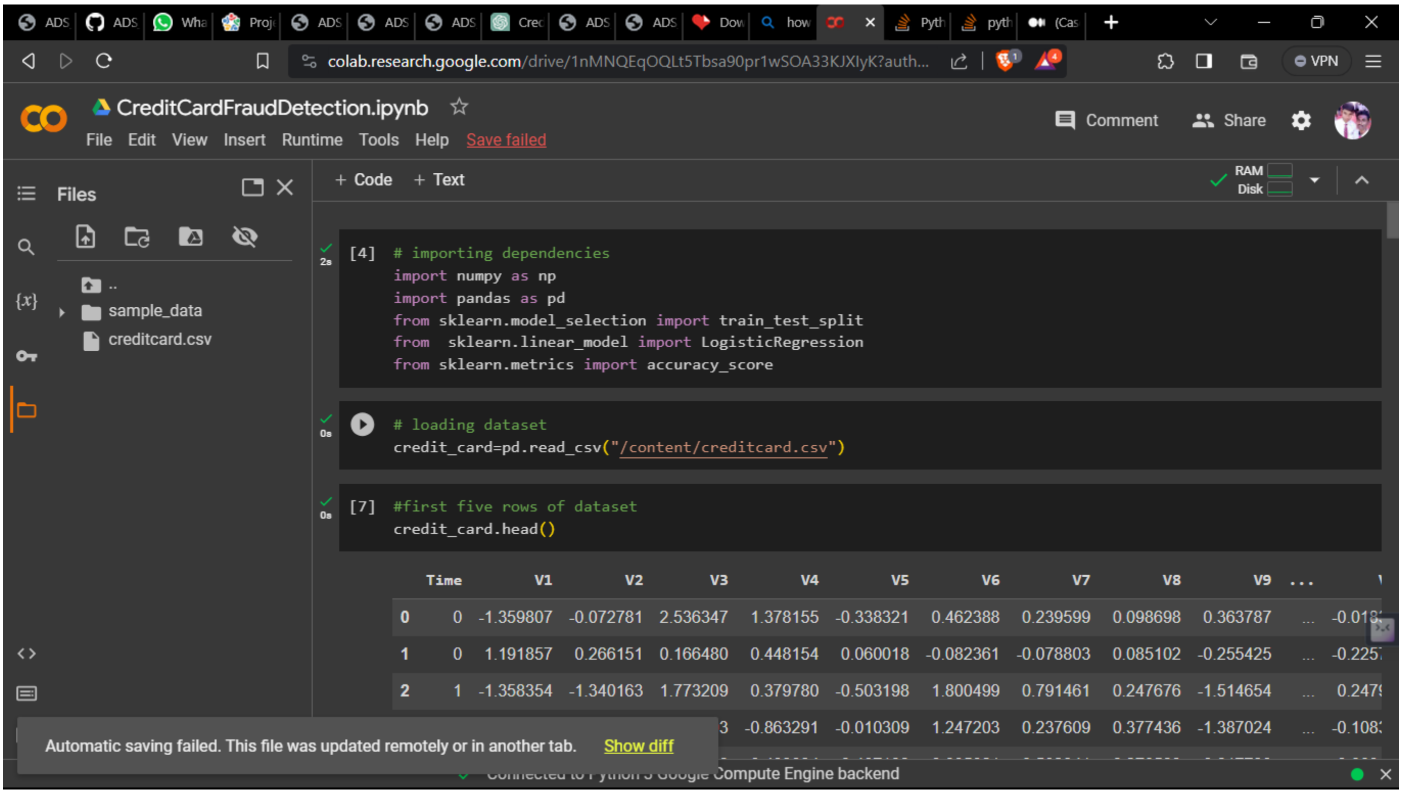
$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

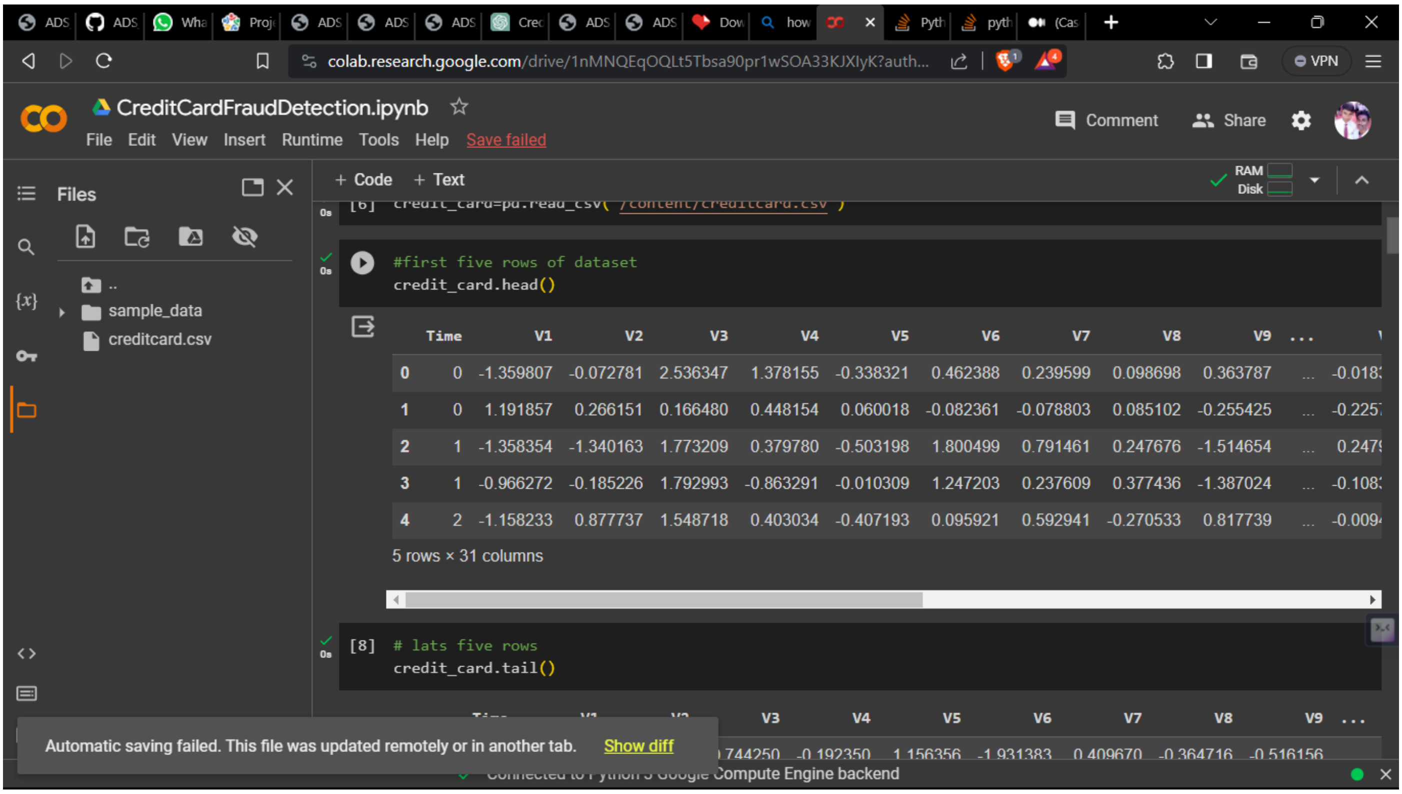
The accuracy on training and testing data as follows:

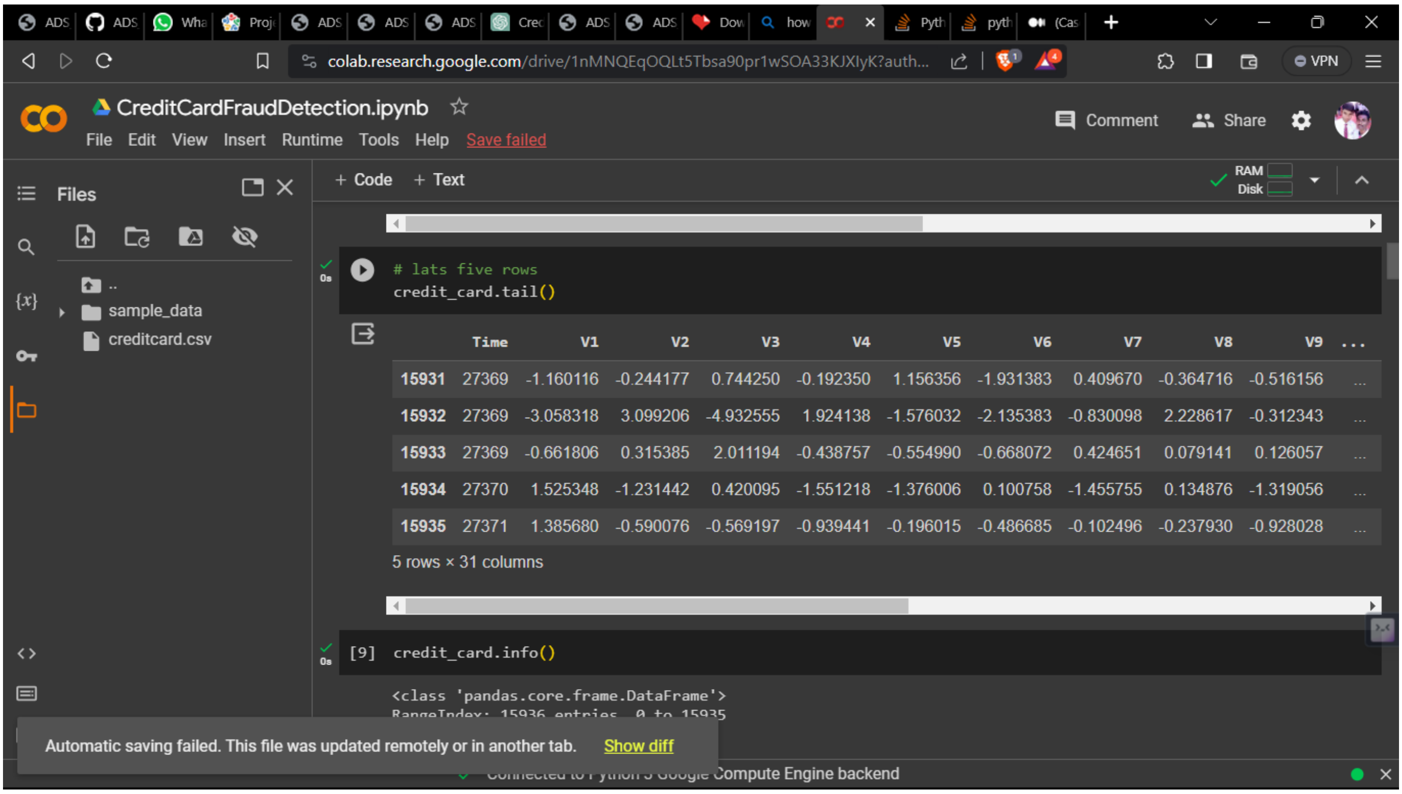


Visualization:

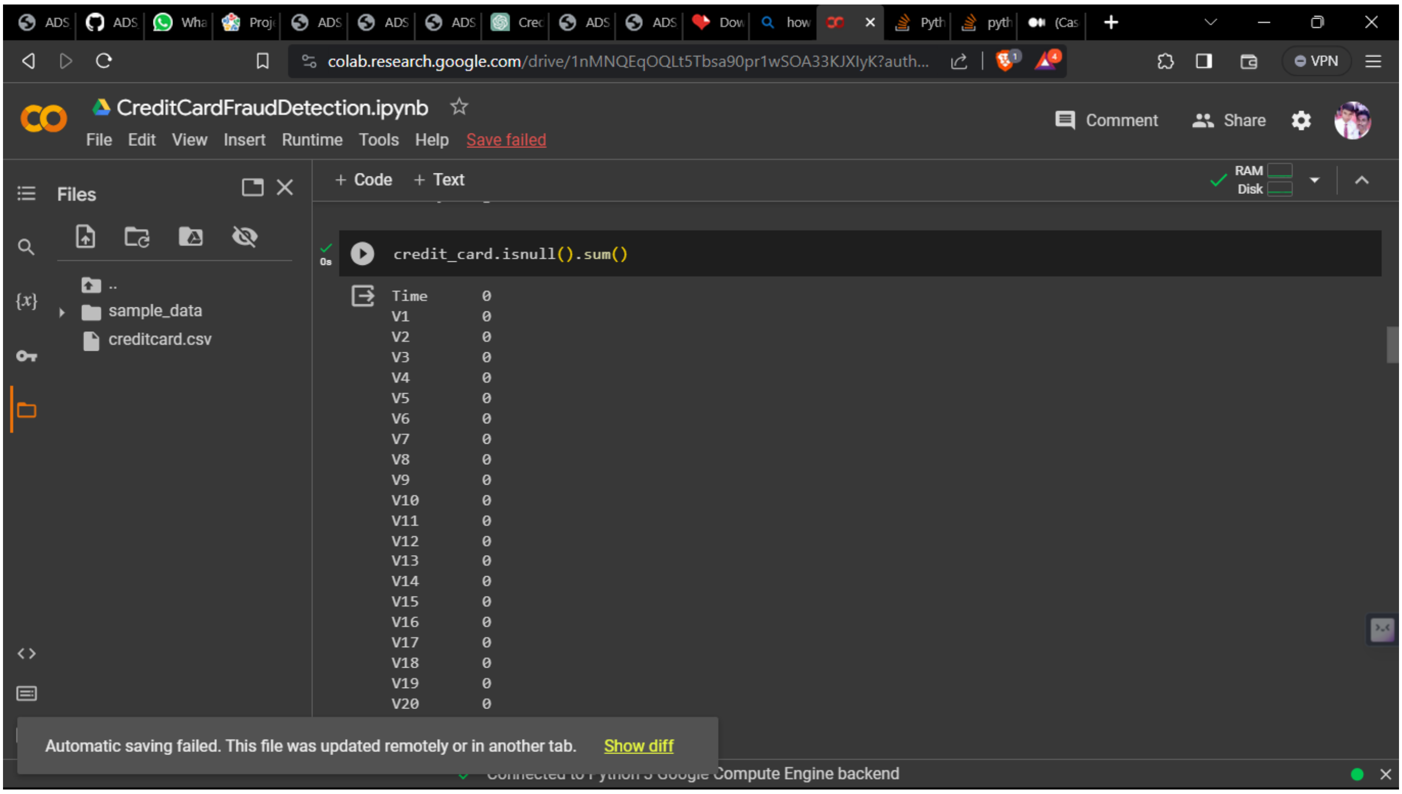
Project Overview in development phase



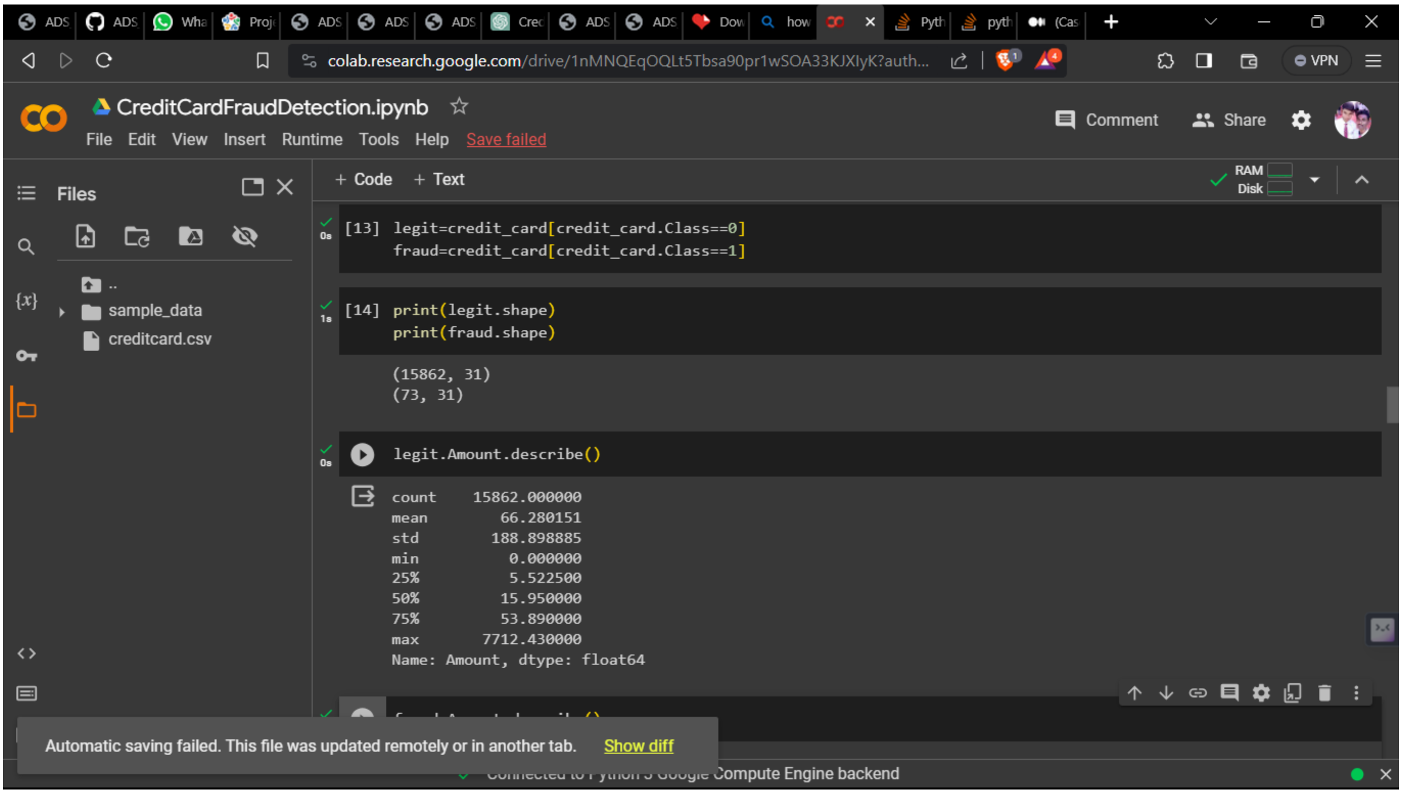




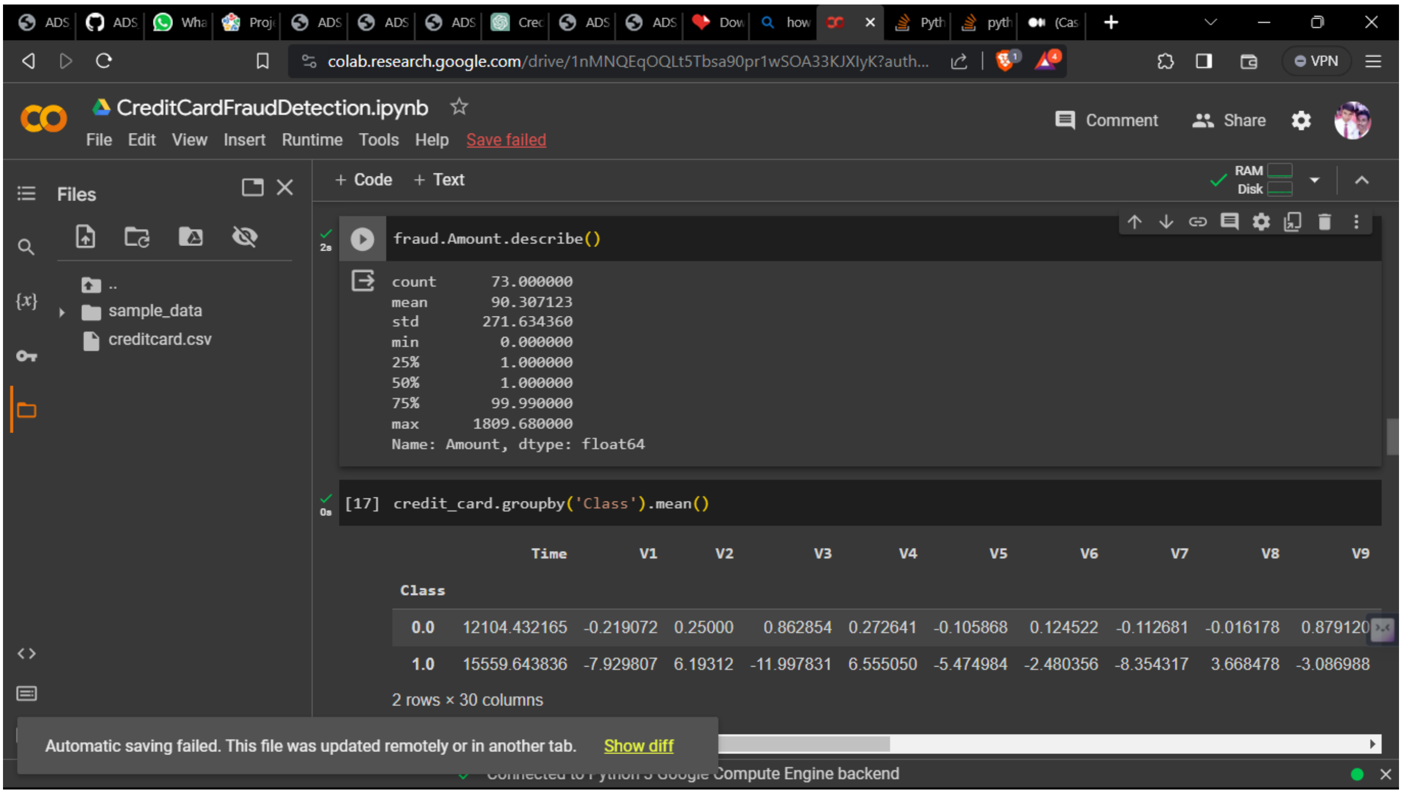
CreditCardFraudDetection.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment Share
+ Code + Text
Files
sample_data
creditcard.csv
# lats five rows
credit_card.tail()
Time V1 V2 V3 V4 V5 V6 V7 V8 V9 ...
15931 27369 -1.160116 -0.244177 0.744250 -0.192350 1.156356 -1.931383 0.409670 -0.364716 -0.516156 ...
15932 27369 -3.058318 3.099206 -4.932555 1.924138 -1.576032 -2.135383 -0.830098 2.228617 -0.312343 ...
15933 27369 -0.661806 0.315385 2.011194 -0.438757 -0.554990 -0.668072 0.424651 0.079141 0.126057 ...
15934 27370 1.525348 -1.231442 0.420095 -1.551218 -1.376006 0.100758 -1.455755 0.134876 -1.319056 ...
15935 27371 1.385680 -0.590076 -0.569197 -0.939441 -0.196015 -0.486685 -0.102496 -0.237930 -0.928028 ...
5 rows × 31 columns
[9] credit_card.info()
<class 'pandas.core.frame.DataFrame'>
Automatic saving failed. This file was updated remotely or in another tab. Show diff
Compute Engine backend



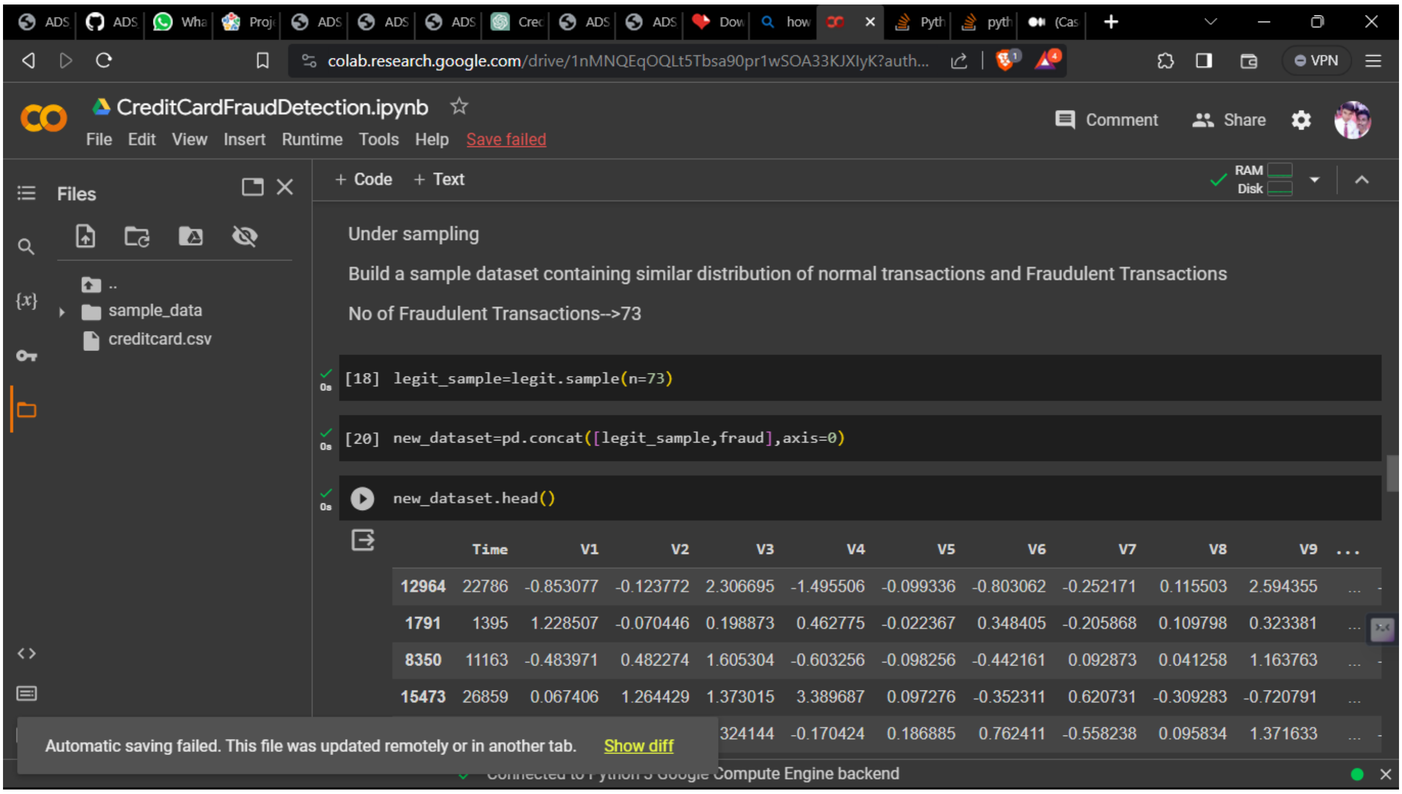
CreditCardFraudDetection.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment Share
+ Code + Text
Files
sample_data
creditcard.csv
credit_card.isnull().sum()
Time 0
V1 0
V2 0
V3 0
V4 0
V5 0
V6 0
V7 0
V8 0
V9 0
V10 0
V11 0
V12 0
V13 0
V14 0
V15 0
V16 0
V17 0
V18 0
V19 0
V20 0
Automatic saving failed. This file was updated remotely or in another tab. Show diff
Compute Engine backend



CreditCardFraudDetection.ipynb
File Edit View Insert Runtime Tools Help Save failed
[13] legit=credit_card[credit_card.Class==0]
fraud=credit_card[credit_card.Class==1]
[14] print(legit.shape)
print(fraud.shape)
(15862, 31)
(73, 31)
legit.Amount.describe()
count 15862.000000
mean 66.280151
std 188.898885
min 0.000000
25% 5.522500
50% 15.950000
75% 53.890000
max 7712.430000
Name: Amount, dtype: float64
Automatic saving failed. This file was updated remotely or in another tab. Show diff



CreditCardFraudDetection.ipynb
File Edit View Insert Runtime Tools Help Save failed
fraud.Amount.describe()
count 73.000000
mean 90.307123
std 271.634360
min 0.000000
25% 1.000000
50% 1.000000
75% 99.990000
max 1809.680000
Name: Amount, dtype: float64
[17] credit_card.groupby('Class').mean()
Time V1 V2 V3 V4 V5 V6 V7 V8 V9
Class
0.0 12104.432165 -0.219072 0.25000 0.862854 0.272641 -0.105868 0.124522 -0.112681 -0.016178 0.879120
1.0 15559.643836 -7.929807 6.19312 -11.997831 6.555050 -5.474984 -2.480356 -8.354317 3.668478 -3.086988
2 rows × 30 columns
Automatic saving failed. This file was updated remotely or in another tab. Show diff



CreditCardFraudDetection.ipynb

Under sampling

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

No of Fraudulent Transactions-->73

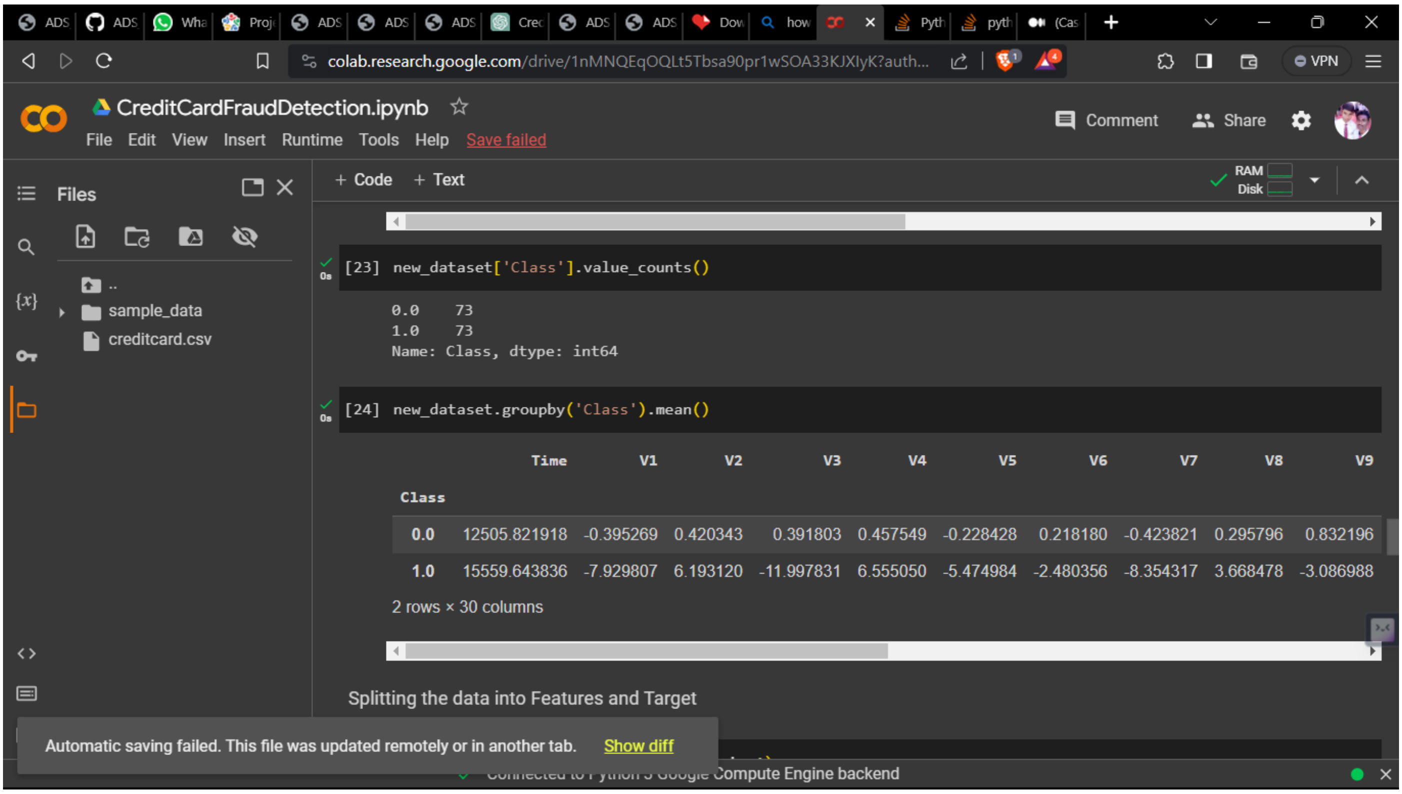
```
[18] legit_sample=legit.sample(n=73)
```

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

```
new_dataset.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...
12964	22786	-0.853077	-0.123772	2.306695	-1.495506	-0.099336	-0.803062	-0.252171	0.115503	2.594355	...
1791	1395	1.228507	-0.070446	0.198873	0.462775	-0.022367	0.348405	-0.205868	0.109798	0.323381	...
8350	11163	-0.483971	0.482274	1.605304	-0.603256	-0.098256	-0.442161	0.092873	0.041258	1.163763	...
15473	26859	0.067406	1.264429	1.373015	3.389687	0.097276	-0.352311	0.620731	-0.309283	-0.720791	...
				324144	-0.170424	0.186885	0.762411	-0.558238	0.095834	1.371633	...

Automatic saving failed. This file was updated remotely or in another tab. Show diff



CreditCardFraudDetection.ipynb

```
[23] new_dataset['Class'].value_counts()
```

```
0.0    73
1.0    73
Name: Class, dtype: int64
```

```
[24] new_dataset.groupby('Class').mean()
```

Class	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9
0.0	12505.821918	-0.395269	0.420343	0.391803	0.457549	-0.228428	0.218180	-0.423821	0.295796	0.832196
1.0	15559.643836	-7.929807	6.193120	-11.997831	6.555050	-5.474984	-2.480356	-8.354317	3.668478	-3.086988

2 rows × 30 columns

Splitting the data into Features and Target

Automatic saving failed. This file was updated remotely or in another tab. Show diff