```
# connect google drive to colab
from google.colab import drive
drive.mount('/content/drive',force_remount=True)
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('/content/drive/MyDrive/ML/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] # indicate true when the condition is met
fraud = credit_card_data[credit_card_data.Class == 1]
print(legit.shape)
print(fraud.shape)
legit.Amount.describe()
fraud.Amount.describe()
# compare the values for both transactions
```

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

legit_sample = legit.sample(n=492)

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()

X = new_dataset.drop(columns='Class', axis=1)

Y = new_dataset['Class']

print(Y)

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 = LogisticRegression()

# training the Logistic Regression Model with Training Data

model.fit(X_train, Y_train)

# 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)
```