Code:

```
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/credit_data.csv')

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

#dataset informations
credit_card_data.info()

#dataset informations
credit_card_data.info()

# 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()

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()

#Splitting the data into Features & Targets
X = new_dataset.drop(columns='Class', axis=1)
Y = new_dataset['Class']
print(X)
print(Y)
```

```
#Split the data into Training data & 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 Training data :  0.9415501905972046

# 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)
#Accuracy score on Test Data :  0.9390862944162437
```