



Experiment: 1.4

Aim: Write a program to implement Fraud Detection in Financial Transactions using Logistic Regression in Business Intelligence Page.

Software Required: Anaconda, Jupiter Lab Dataset for visualization

Description: Logistic Regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class.

Algorithms:

STEP 1: Import all the necessary libraries that are required.

STEP 2: Import and read the dataset.

STEP 3: Data Preprocessing and Exploration

STEP 4: Split the Dataset into Train and Test Sets

STEP 5: Train the Logistic Regression Model

STEP 6: Make Predictions and Evaluate the Model

STEP 7: Interpret the result



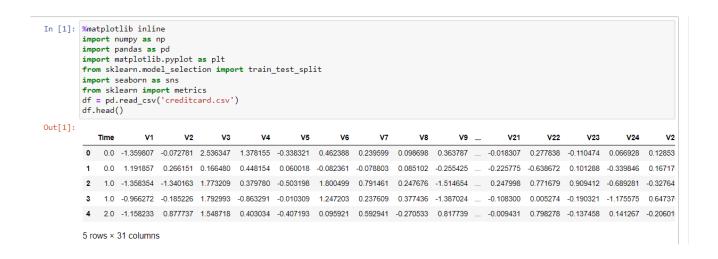
Code and Output:

```
In [1]: %matplotlib inline
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
        import seaborn as sns
        from sklearn import metrics
        df = pd.read_csv('creditcard.csv')
        df.head()
Out[1]:
                            V2 V3 V4
                                                       V5
                                                                V6
                                                                         V7
                                                                                 V8
                                                                                          V9 ...
         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
            1.0 -1.358354 -1.340163 1.773209 0.379780 -0.503198 1.800499 0.791461 0.247676 -1.514654 ...
                                                                                                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 [32]: X_train, X_test, y_train, y_test = train_test_split(data_features, data_target, train_size = 0.70, test_size = 0.30,
                                                              random_state = 1)
  In [33]: X_train.shape
 Out[33]: (199364, 29)
  In [34]: X_test.shape
 Out[34]: (85443, 29)
  In [35]: y_train.shape
 Out[35]: (199364, 1)
 In [38]: y_train = np.ravel(y_train)
  In [42]: from sklearn.linear_model import LogisticRegression
           lr = LogisticRegression(max_iter=1000)
           lr.fit(X_train, y_train)
 Out[42]:
                 LogisticRegression
           LogisticRegression(max_iter=1000)
In [44]: #Predict Regression
         lr.predict(X_test[0:10])
Out[44]: array([0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
In [47]: predictions = lr.predict(X_test)
         #Find accuracy score
         score = lr.score(X_test, y_test)
         print(score)
         0.9991573329588147
In [48]: #Print Confusion Matrix
         cm = metrics.confusion_matrix(y_test, predictions)
         print(cm)
         [[85293
          [ 57
                    78]]
```





Learning Outcome:

- How to split data into train set and test set
- · How to perform logistic regression
- How to check accuracy score
- What is confusion matrix and how to find?
- How to draw heat map