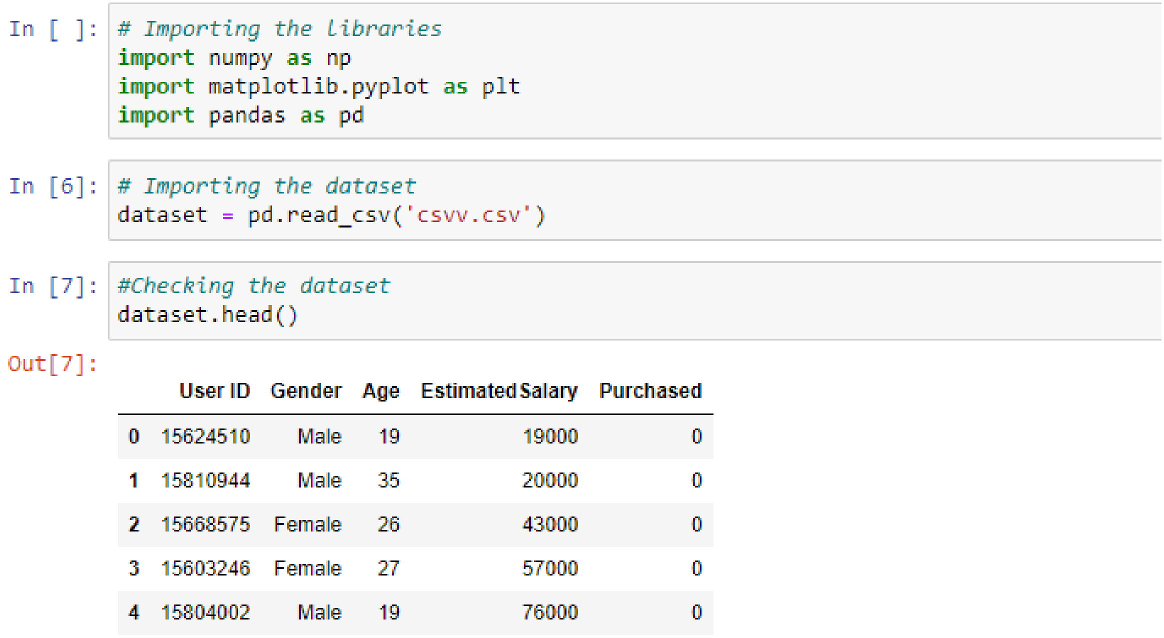


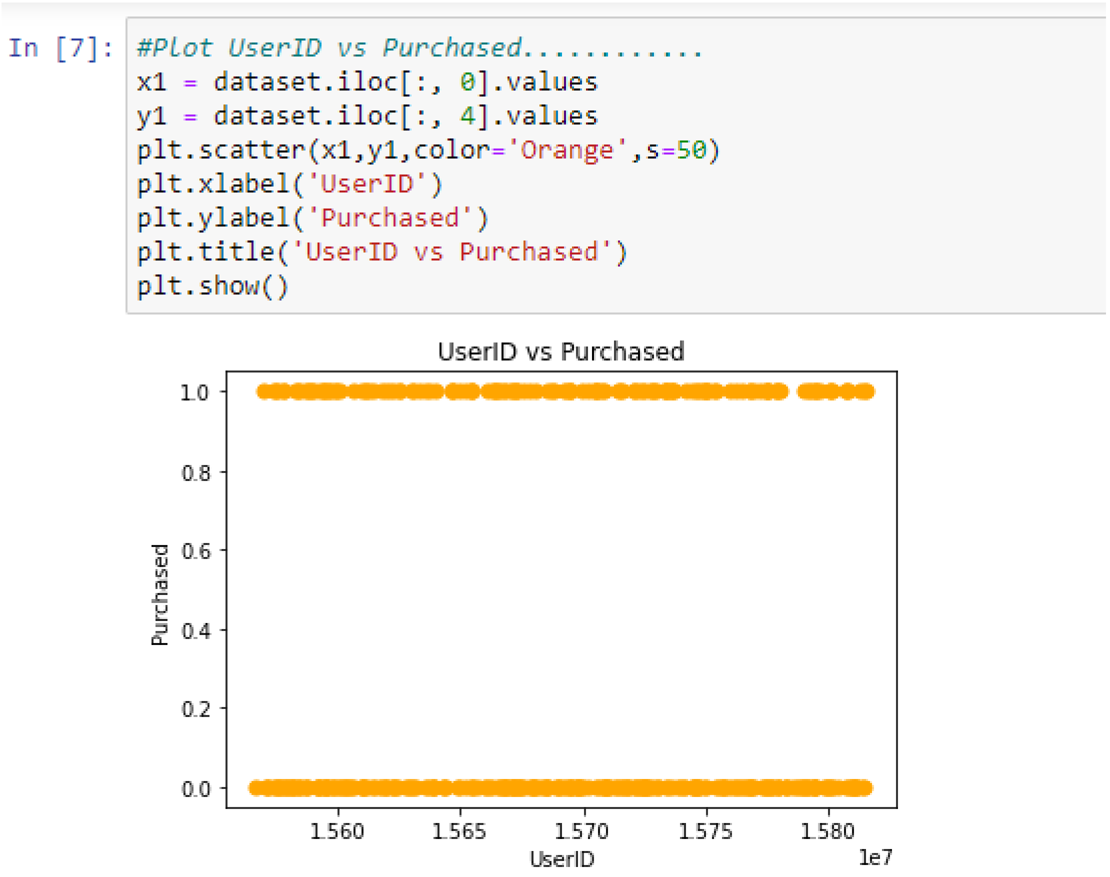
```
In [ ]: # Importing the libraries
        import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

```
In [6]: # Importing the dataset
        dataset = pd.read_csv('csvv.csv')
```

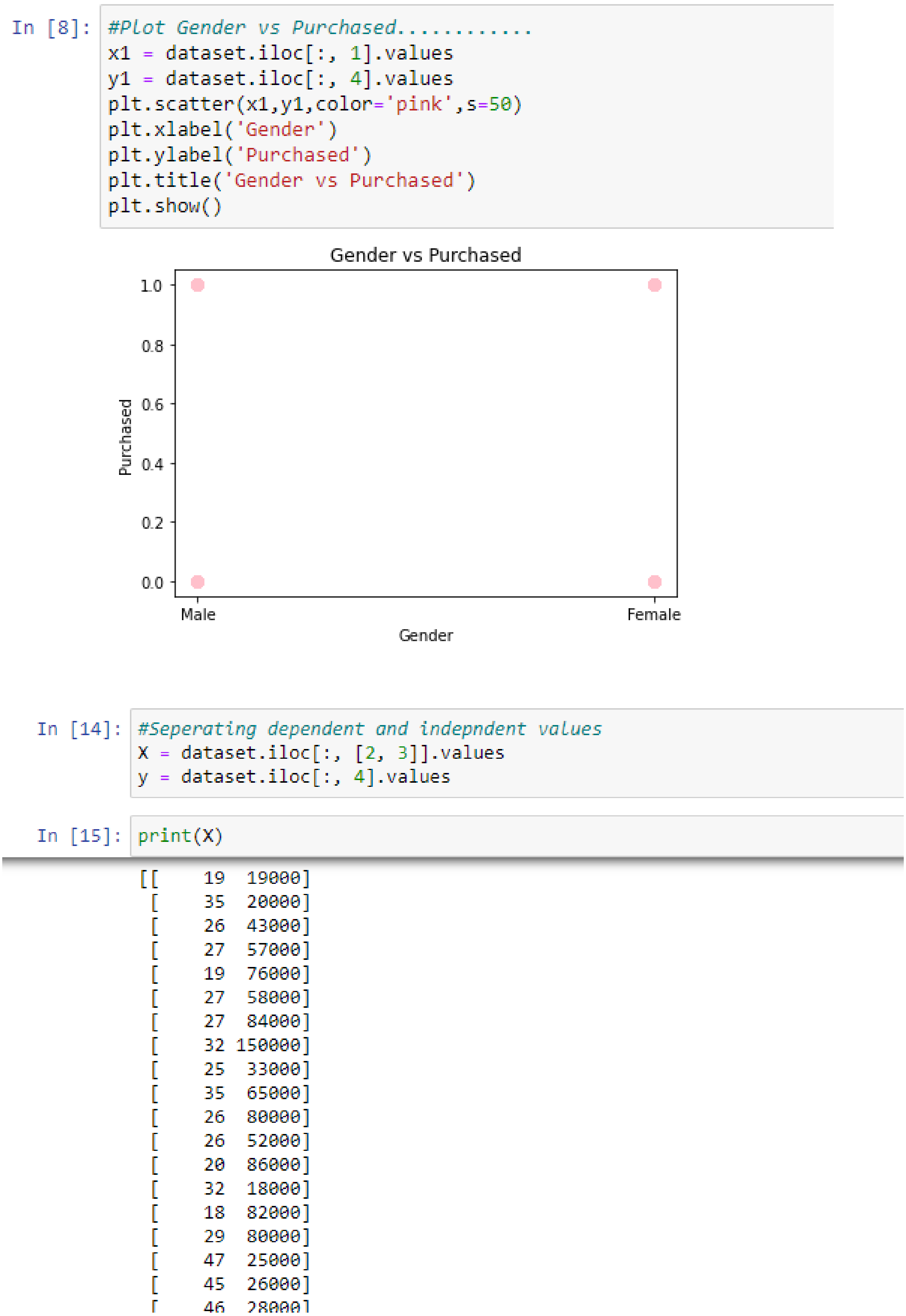
```
In [7]: #Checking the dataset
        dataset.head()
```

Out[7]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0



```
In [7]: #Plot UserID vs Purchased...........
        x1 = dataset.iloc[:, 0].values
        y1 = dataset.iloc[:, 4].values
        plt.scatter(x1,y1,color='Orange',s=50)
        plt.xlabel('UserID')
        plt.ylabel('Purchased')
        plt.title('UserID vs Purchased')
        plt.show()
```

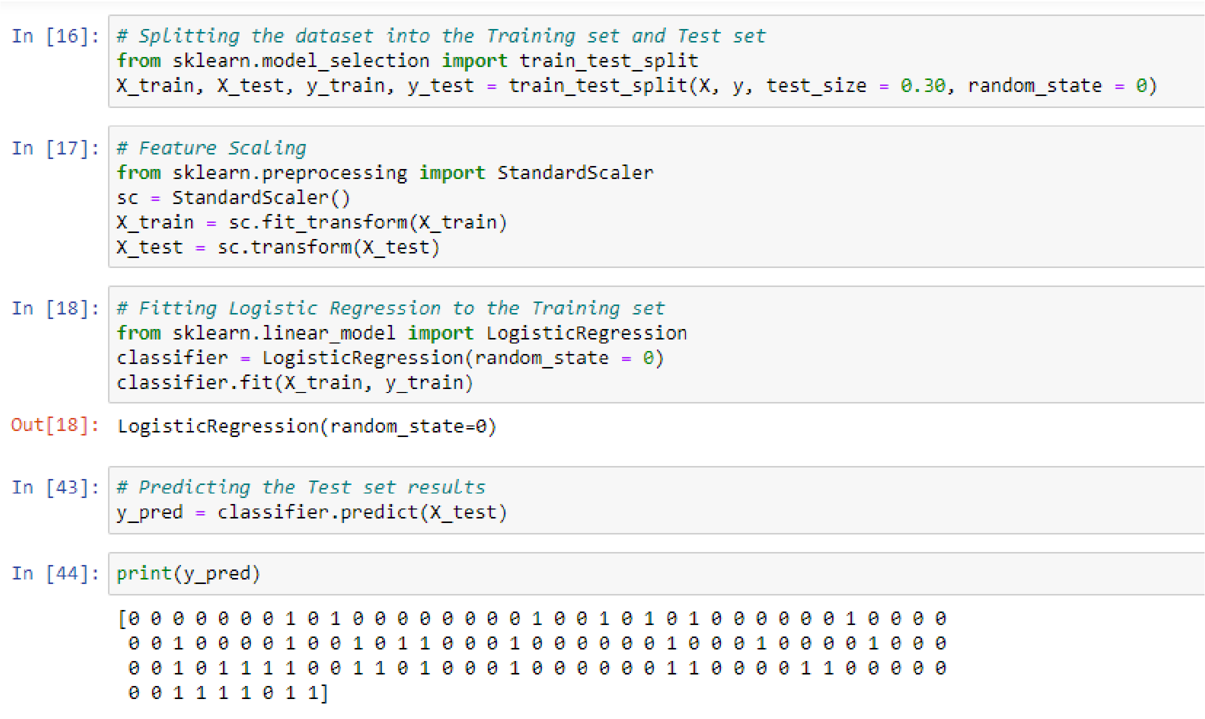


```
In [8]: #Plot Gender vs Purchased............
        x1 = dataset.iloc[:, 1].values
        y1 = dataset.iloc[:, 4].values
        plt.scatter(x1,y1,color='pink',s=50)
        plt.xlabel('Gender')
        plt.ylabel('Purchased')
        plt.title('Gender vs Purchased')
        plt.show()
```

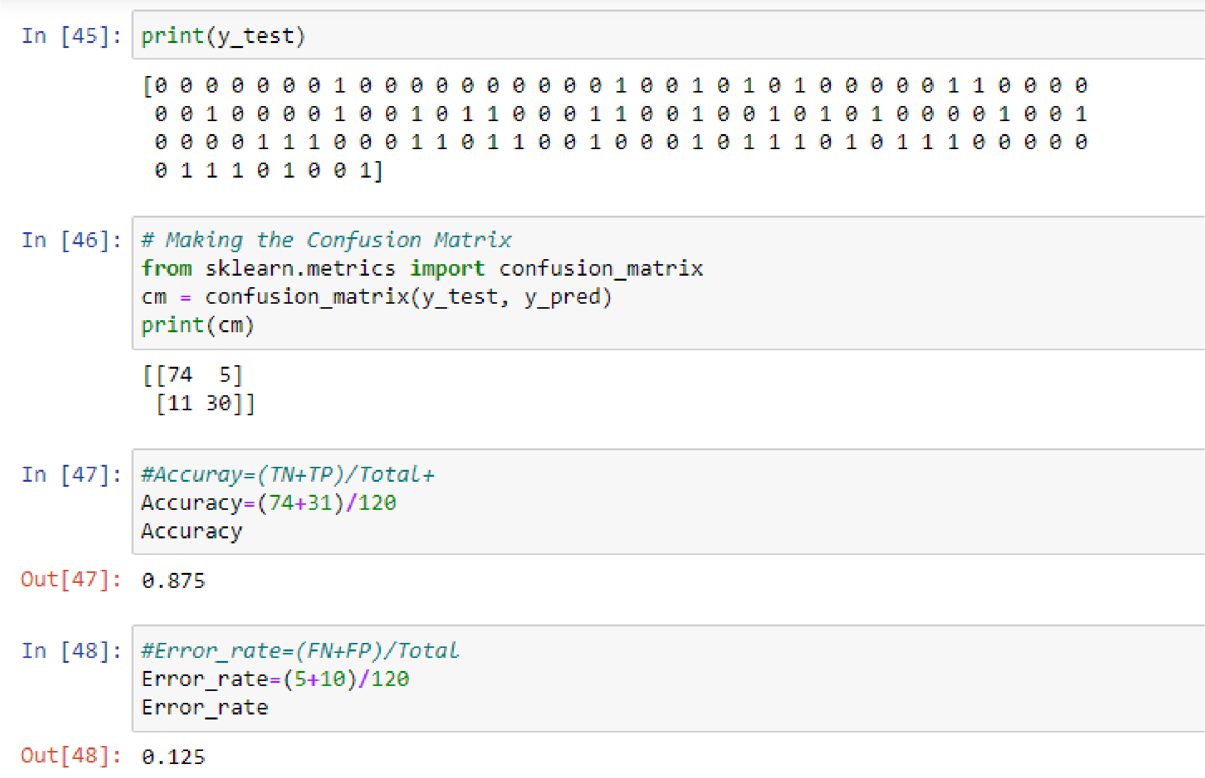
```
In [14]: #Seperating dependent and indepndent values
         X = dataset.iloc[:, [2, 3]].values
         y = dataset.iloc[:, 4].values
```

```
In [15]: print(X)
```

```
[[    19  19000]
 [    35  20000]
 [    26  43000]
 [    27  57000]
 [    19  76000]
 [    27  58000]
 [    27  84000]
 [    32 150000]
 [    25  33000]
 [    35  65000]
 [    26  80000]
 [    26  52000]
 [    20  86000]
 [    32  18000]
 [    18  82000]
 [    29  80000]
 [    47  25000]
 [    45  26000]
 [    46  28000]
```



```
In [16]: # Splitting the dataset into the Training set and Test set
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)

In [17]: # Feature Scaling
         from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         X_train = sc.fit_transform(X_train)
         X_test = sc.transform(X_test)

In [18]: # Fitting Logistic Regression to the Training set
         from sklearn.linear_model import LogisticRegression
         classifier = LogisticRegression(random_state = 0)
         classifier.fit(X_train, y_train)

Out[18]: LogisticRegression(random_state=0)

In [43]: # Predicting the Test set results
         y_pred = classifier.predict(X_test)

In [44]: print(y_pred)

         [0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0
          0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0
          0 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0
          0 0 1 1 1 1 0 1 1]
```



```
In [45]: print(y_test)

         [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0
          0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 0 0 1
          0 0 0 0 1 1 1 0 0 0 1 1 0 1 1 0 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 0 0 0 0 0
          0 1 1 1 0 1 0 0 1]

In [46]: # Making the Confusion Matrix
         from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(y_test, y_pred)
         print(cm)

         [[74  5]
          [11 30]]

In [47]: #Accuray=(TN+TP)/Total+
         Accuracy=(74+31)/120
         Accuracy

Out[47]: 0.875

In [48]: #Error_rate=(FN+FP)/Total
         Error_rate=(5+10)/120
         Error_rate

Out[48]: 0.125
```

```
In [49]: # Finding precision and recall
         #precision=TP/(TP+FP)
         from sklearn.metrics import precision_score, recall_score
         precision_score(y_test, y_pred)

Out[49]: 0.8571428571428571
```

```
In [50]: #recall=TP/(TP+FN)
         recall_score(y_test, y_pred)

Out[50]: 0.7317073170731707
```

```
In [52]: # F1 score=2*(precision*recall/precision+recall)
         from sklearn.metrics import f1_score
         f1_score(y_test, y_pred)

Out[52]: 0.7894736842105263
```