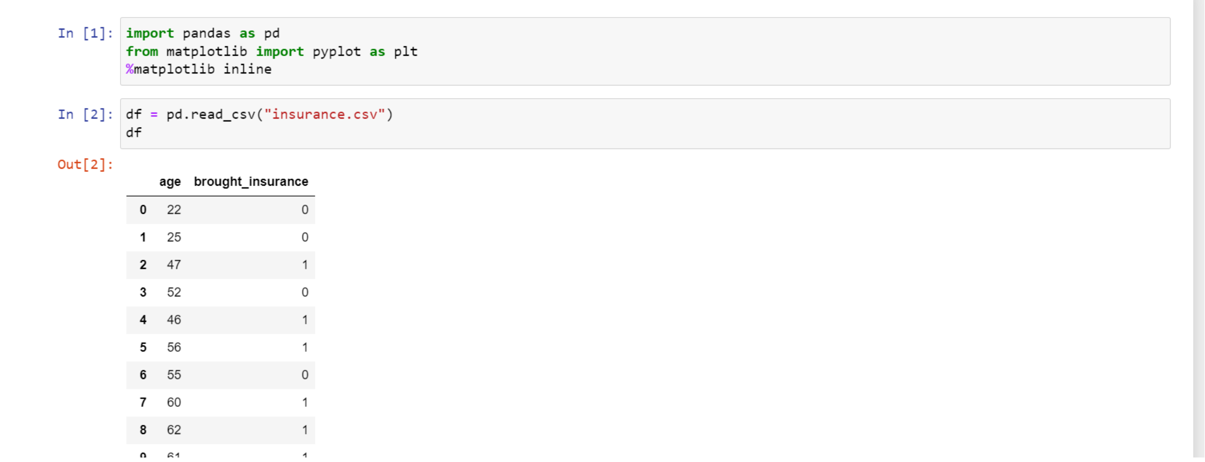
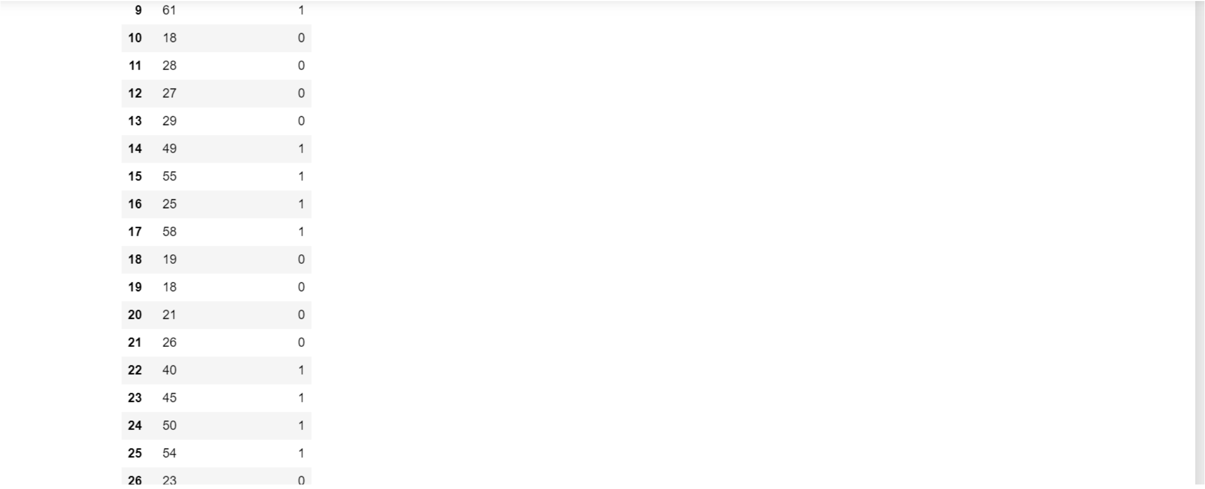
# Binary classification code



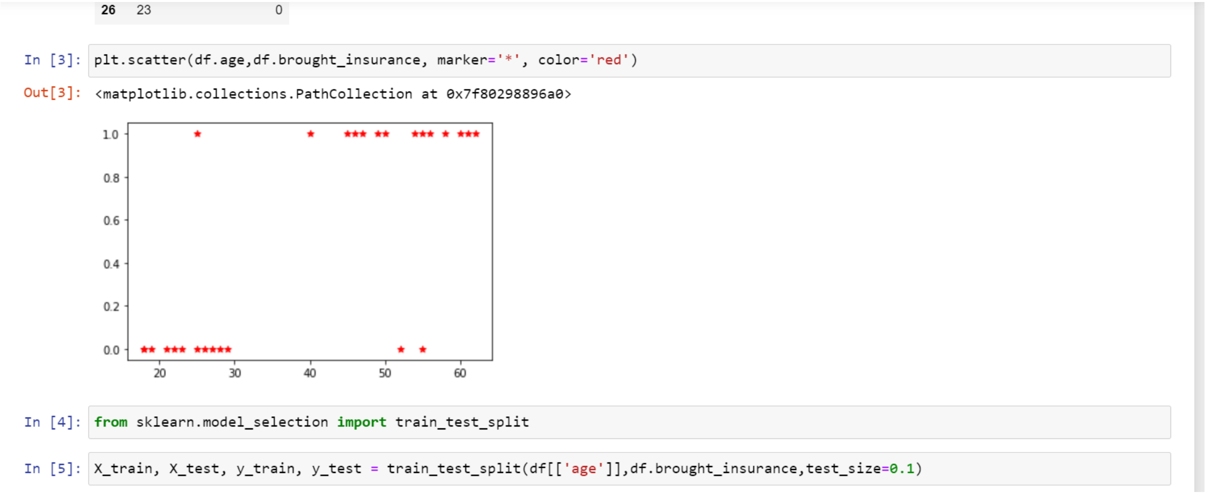
```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt
        %matplotlib inline

In [2]: df = pd.read_csv("insurance.csv")
        df

Out[2]:
```



| | age | brought_insurance |
|---|---|---|
| 0 | 22 | 0 |
| 1 | 25 | 0 |
| 2 | 47 | 1 |
| 3 | 52 | 0 |
| 4 | 46 | 1 |
| 5 | 56 | 1 |
| 6 | 55 | 0 |
| 7 | 60 | 1 |
| 8 | 62 | 1 |
| 9 | 61 | 1 |
| 10 | 18 | 0 |
| 11 | 28 | 0 |
| 12 | 27 | 0 |
| 13 | 29 | 0 |
| 14 | 49 | 1 |
| 15 | 55 | 1 |
| 16 | 25 | 1 |
| 17 | 58 | 1 |
| 18 | 19 | 0 |
| 19 | 18 | 0 |
| 20 | 21 | 0 |
| 21 | 26 | 0 |
| 22 | 40 | 1 |
| 23 | 45 | 1 |
| 24 | 50 | 1 |
| 25 | 54 | 1 |
| 26 | 23 | 0 |

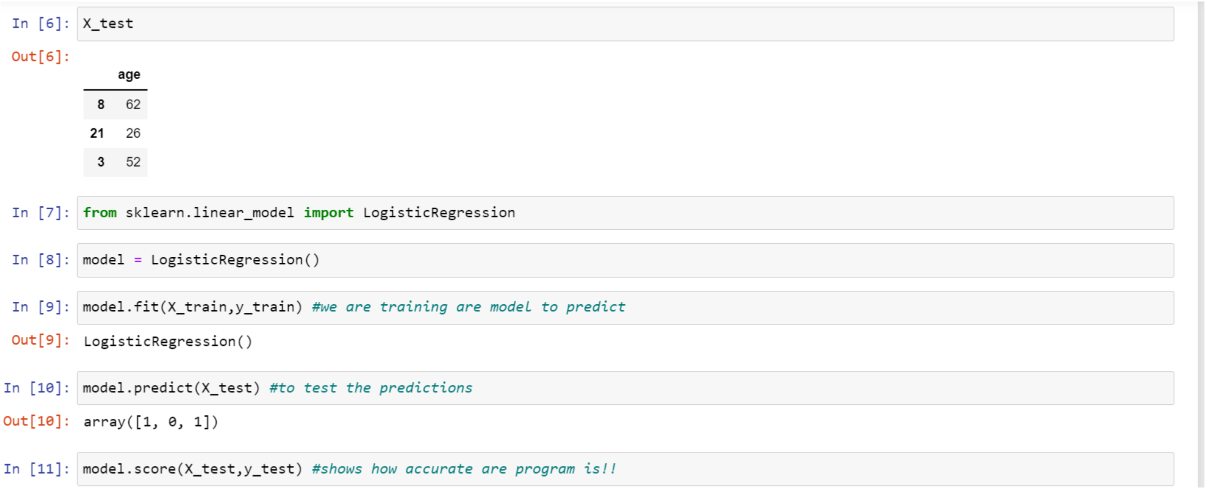


```
In [3]: plt.scatter(df.age,df.brought_insurance, marker='*', color='red')

Out[3]: <matplotlib.collections.PathCollection at 0x7f80298896a0>
```

```
In [4]: from sklearn.model_selection import train_test_split

In [5]: X_train, X_test, y_train, y_test = train_test_split(df[['age']],df.brought_insurance,test_size=0.1)
```



```
In [6]: X_test
Out[6]:
```

|  | age |
|---|---|
| **8** | 62 |
| **21** | 26 |
| **3** | 52 |

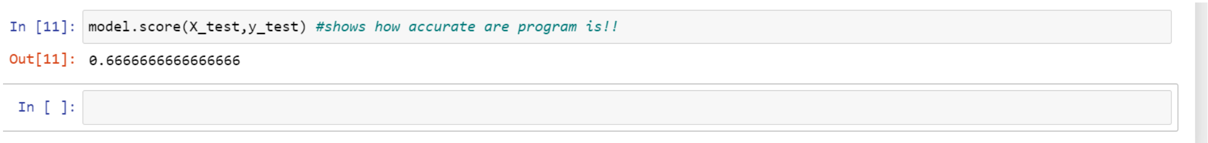
```
In [7]: from sklearn.linear_model import LogisticRegression

In [8]: model = LogisticRegression()

In [9]: model.fit(X_train,y_train) #we are training are model to predict
Out[9]: LogisticRegression()

In [10]: model.predict(X_test) #to test the predictions
Out[10]: array([1, 0, 1])

In [11]: model.score(X_test,y_test) #shows how accurate are program is!!
```

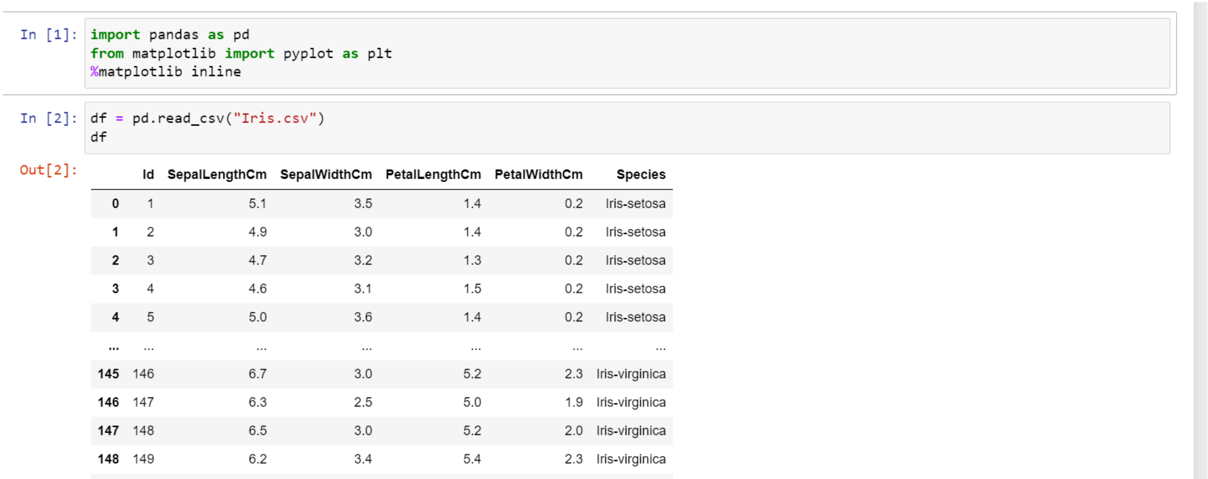


```
In [11]: model.score(X_test,y_test) #shows how accurate are program is!!
```

```
Out[11]: 0.6666666666666666

In [ ]:
```

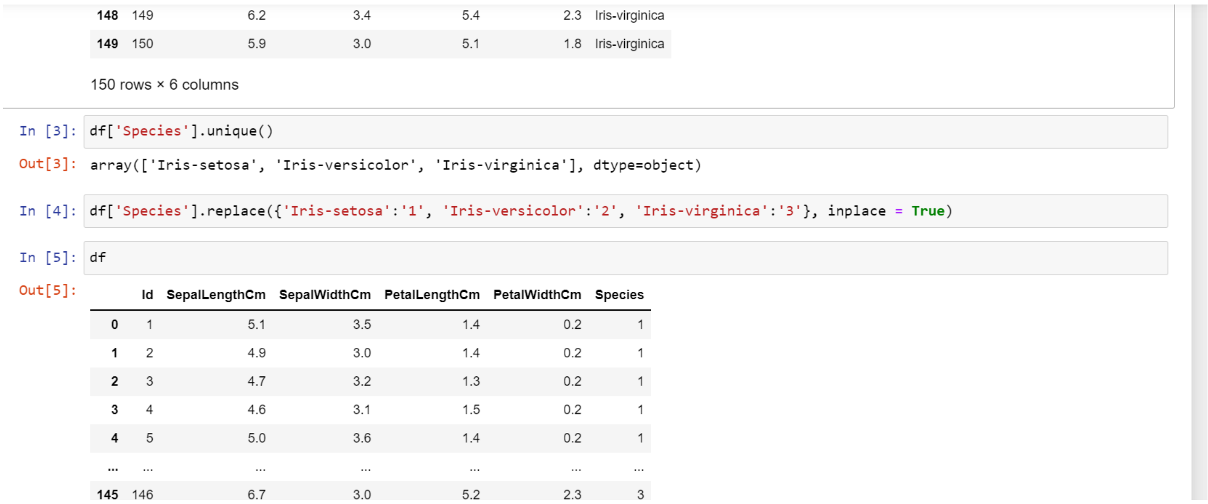
# Multiclass Classification Code:



```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt
        %matplotlib inline

In [2]: df = pd.read_csv("Iris.csv")
        df
Out[2]:
```

|  | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| **0** | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **3** | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **4** | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| **...** | ... | ... | ... | ... | ... | ... |
| **145** | 146 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| **146** | 147 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| **147** | 148 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| **148** | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |



|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| **148** | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| **149** | 150 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

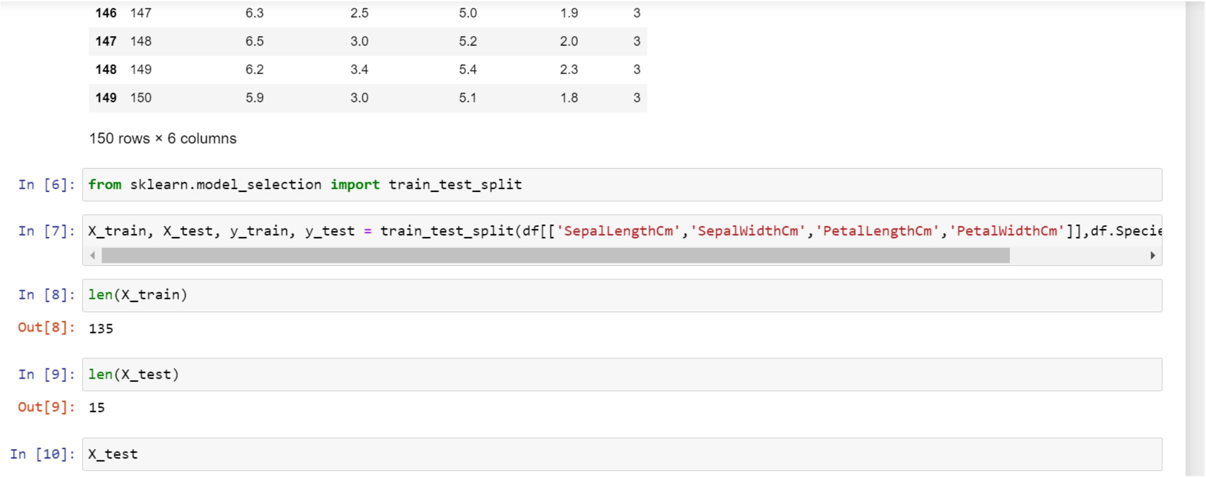
150 rows × 6 columns

```
In [3]: df['Species'].unique()
Out[3]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)

In [4]: df['Species'].replace({'Iris-setosa':'1', 'Iris-versicolor':'2', 'Iris-virginica':'3'}, inplace = True)

In [5]: df
Out[5]:
```

|  | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| **0** | 1 | 5.1 | 3.5 | 1.4 | 0.2 | 1 |
| **1** | 2 | 4.9 | 3.0 | 1.4 | 0.2 | 1 |
| **2** | 3 | 4.7 | 3.2 | 1.3 | 0.2 | 1 |
| **3** | 4 | 4.6 | 3.1 | 1.5 | 0.2 | 1 |
| **4** | 5 | 5.0 | 3.6 | 1.4 | 0.2 | 1 |
| **...** | ... | ... | ... | ... | ... | ... |
| **145** | 146 | 6.7 | 3.0 | 5.2 | 2.3 | 3 |



| | | | | | | |
|---|---|---|---|---|---|---|
| **146** | 147 | 6.3 | 2.5 | 5.0 | 1.9 | 3 |
| **147** | 148 | 6.5 | 3.0 | 5.2 | 2.0 | 3 |
| **148** | 149 | 6.2 | 3.4 | 5.4 | 2.3 | 3 |
| **149** | 150 | 5.9 | 3.0 | 5.1 | 1.8 | 3 |

150 rows × 6 columns

```
In [6]: from sklearn.model_selection import train_test_split
```

```
In [7]: X_train, X_test, y_train, y_test = train_test_split(df[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']],df.Specie
```

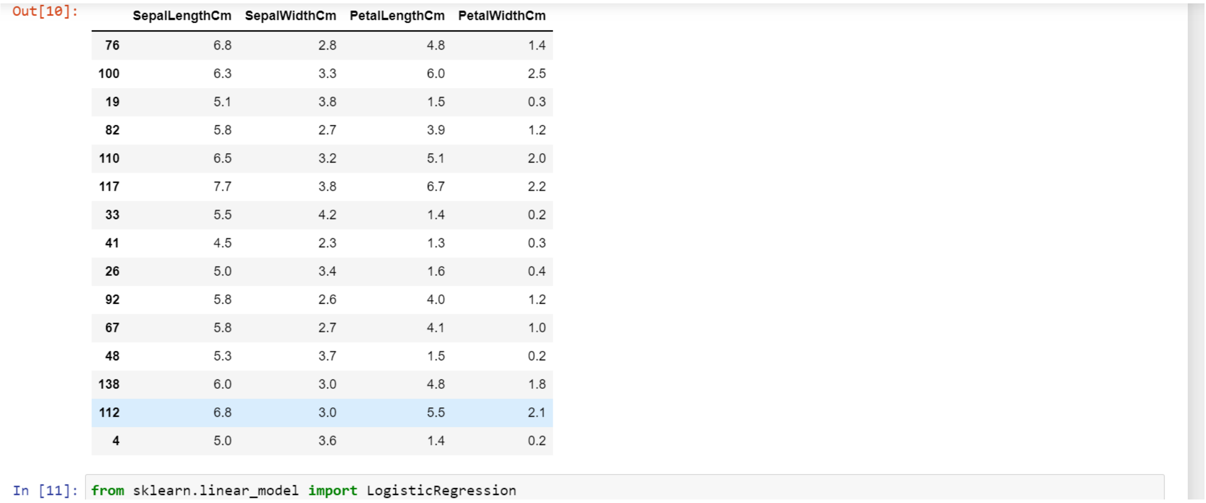
```
In [8]: len(X_train)

Out[8]: 135
```

```
In [9]: len(X_test)

Out[9]: 15
```

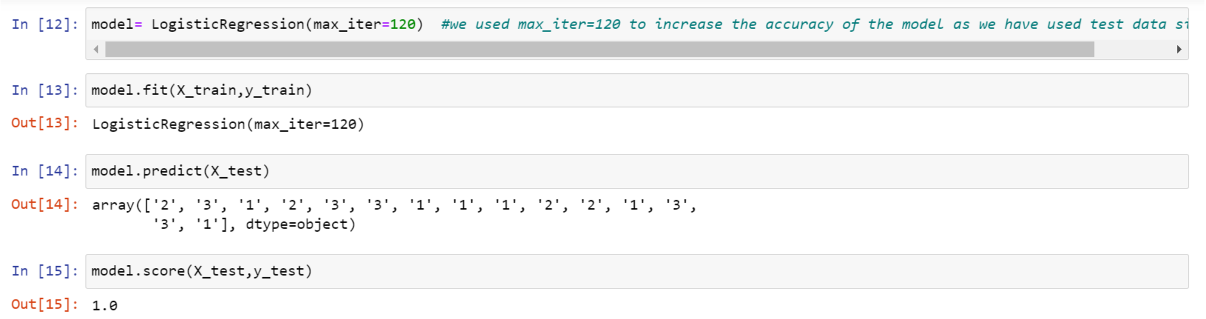
```
In [10]: X_test
```



Out[10]:

| | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm |
|---|---|---|---|---|
| **76** | 6.8 | 2.8 | 4.8 | 1.4 |
| **100** | 6.3 | 3.3 | 6.0 | 2.5 |
| **19** | 5.1 | 3.8 | 1.5 | 0.3 |
| **82** | 5.8 | 2.7 | 3.9 | 1.2 |
| **110** | 6.5 | 3.2 | 5.1 | 2.0 |
| **117** | 7.7 | 3.8 | 6.7 | 2.2 |
| **33** | 5.5 | 4.2 | 1.4 | 0.2 |
| **41** | 4.5 | 2.3 | 1.3 | 0.3 |
| **26** | 5.0 | 3.4 | 1.6 | 0.4 |
| **92** | 5.8 | 2.6 | 4.0 | 1.2 |
| **67** | 5.8 | 2.7 | 4.1 | 1.0 |
| **48** | 5.3 | 3.7 | 1.5 | 0.2 |
| **138** | 6.0 | 3.0 | 4.8 | 1.8 |
| **112** | 6.8 | 3.0 | 5.5 | 2.1 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 |

```
In [11]: from sklearn.linear_model import LogisticRegression
```



```
In [12]: model= LogisticRegression(max_iter=120)  #we used max_iter=120 to increase the accuracy of the model as we have used test data si
```

```
In [13]: model.fit(X_train,y_train)

Out[13]: LogisticRegression(max_iter=120)
```

```
In [14]: model.predict(X_test)

Out[14]: array(['2', '3', '1', '2', '3', '3', '1', '1', '1', '2', '2', '1', '3',
                '3', '1'], dtype=object)
```

```
In [15]: model.score(X_test,y_test)

Out[15]: 1.0
```