Naive bayes

What is our GOAL for this MODULE?

The goal of this module is to explore the dependencies of the variables through Naive Bayes algorithm.

What did we ACHIEVE in the class TODAY?

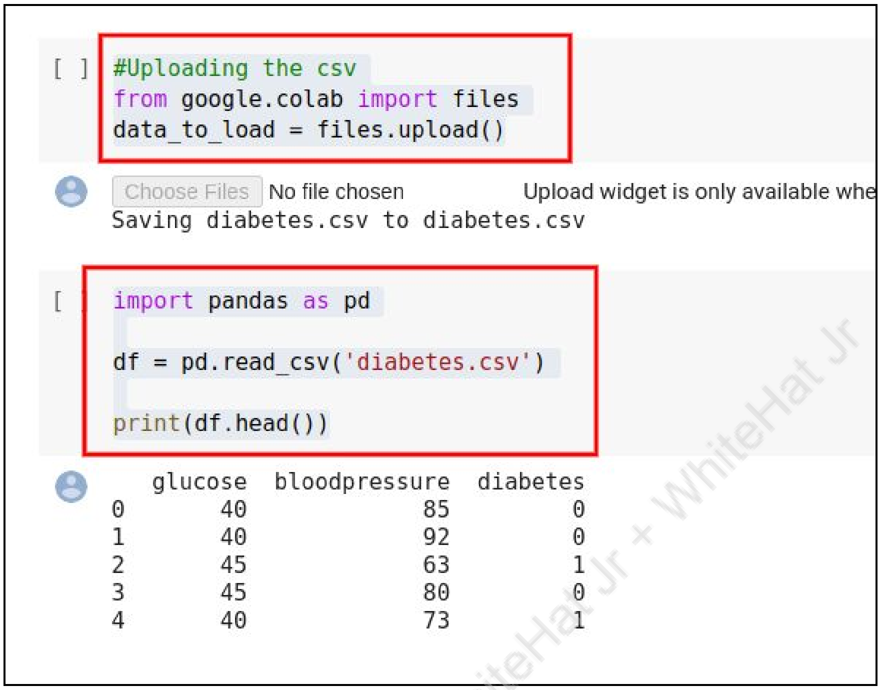
- We explored the concept of Naive Bayes.
- We learned about Bayes law.
- Compared Naive Bayes and Logistics regression and made conclusions on the output.

Which CONCEPTS/CODING BLOCKS did we cover today?

- Naive Bayes algorithm
- Bayes law
- Logistic regression

How did we DO the activities?

1. We explored the concept of Naive Bayes.
2. We took data of people who had diabetes, uploaded it and printed it.



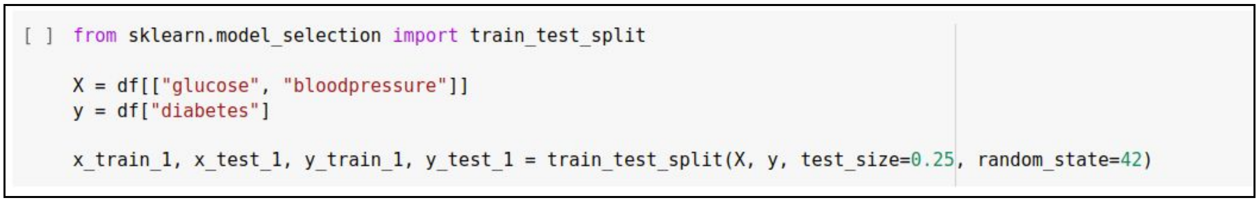
```
#Uploading the csv
from google.colab import files
data_to_load = files.upload()
```

```
Saving diabetes.csv to diabetes.csv
```

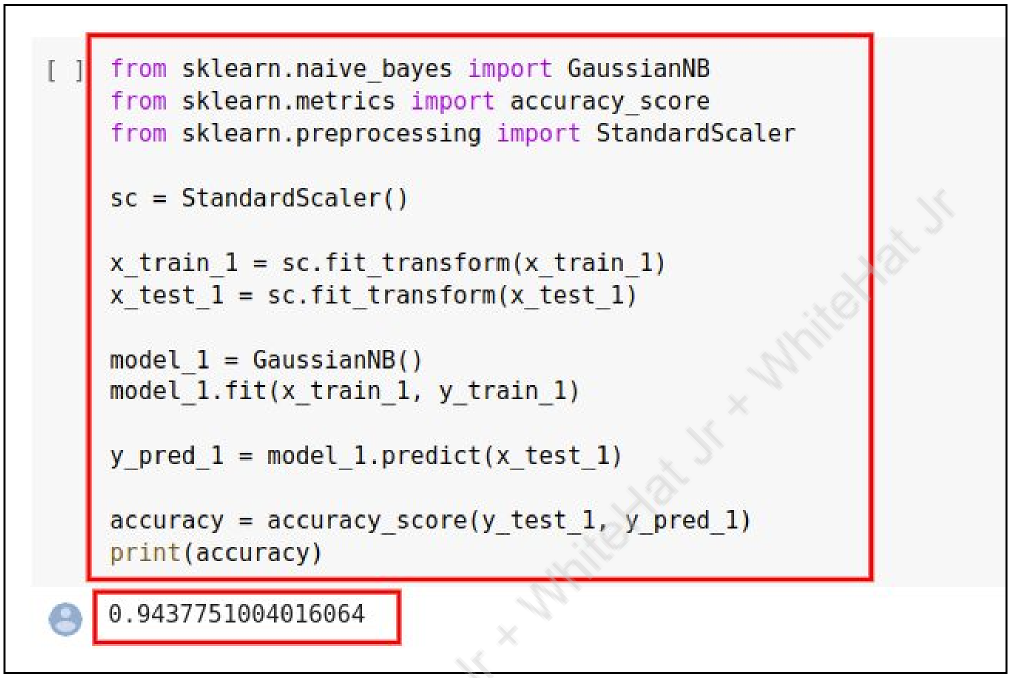
```
import pandas as pd

df = pd.read_csv('diabetes.csv')

print(df.head())
```

```
   glucose  bloodpressure  diabetes
0       40             85         0
1       40             92         0
2       45             63         1
3       45             80         0
4       40             73         1
```

3. We split the data to train and test the Naive Bayes model.



```
[ ]  from sklearn.model_selection import train_test_split

     X = df[["glucose", "bloodpressure"]]
     y = df["diabetes"]

     x_train_1, x_test_1, y_train_1, y_test_1 = train_test_split(X, y, test_size=0.25, random_state=42)
```

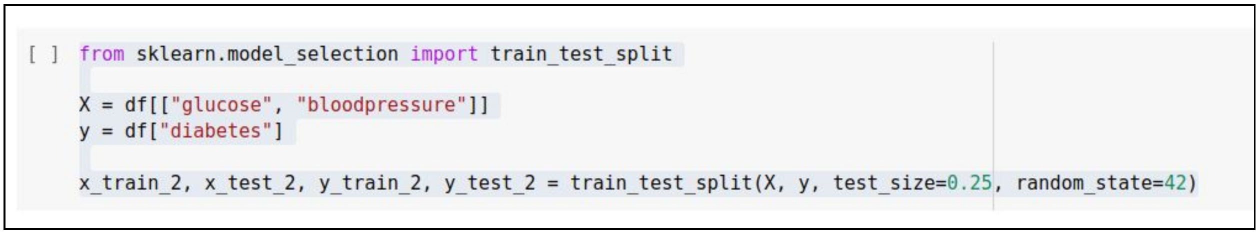


```
[ ]  from sklearn.naive_bayes import GaussianNB
     from sklearn.metrics import accuracy_score
     from sklearn.preprocessing import StandardScaler

     sc = StandardScaler()

     x_train_1 = sc.fit_transform(x_train_1)
     x_test_1 = sc.fit_transform(x_test_1)

     model_1 = GaussianNB()
     model_1.fit(x_train_1, y_train_1)

     y_pred_1 = model_1.predict(x_test_1)

     accuracy = accuracy_score(y_test_1, y_pred_1)
     print(accuracy)
```

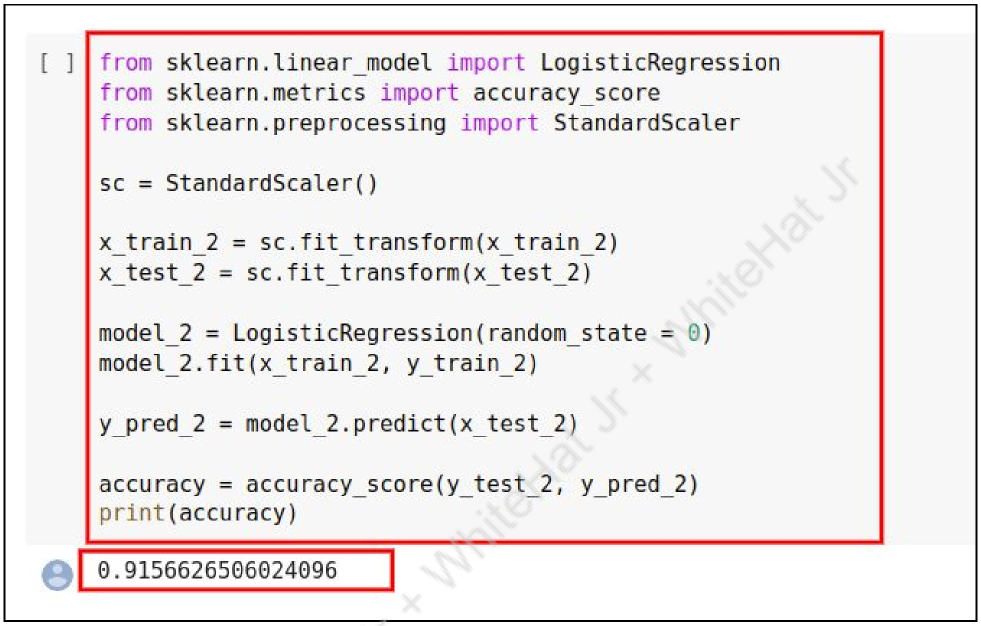
```
0.9437751004016064
```

4. We got the accuracy of 94%.

5. Then we split the data to train and test the Logistics regression model.



```
[ ] from sklearn.model_selection import train_test_split

    X = df[["glucose", "bloodpressure"]]
    y = df["diabetes"]

    x_train_2, x_test_2, y_train_2, y_test_2 = train_test_split(X, y, test_size=0.25, random_state=42)
```

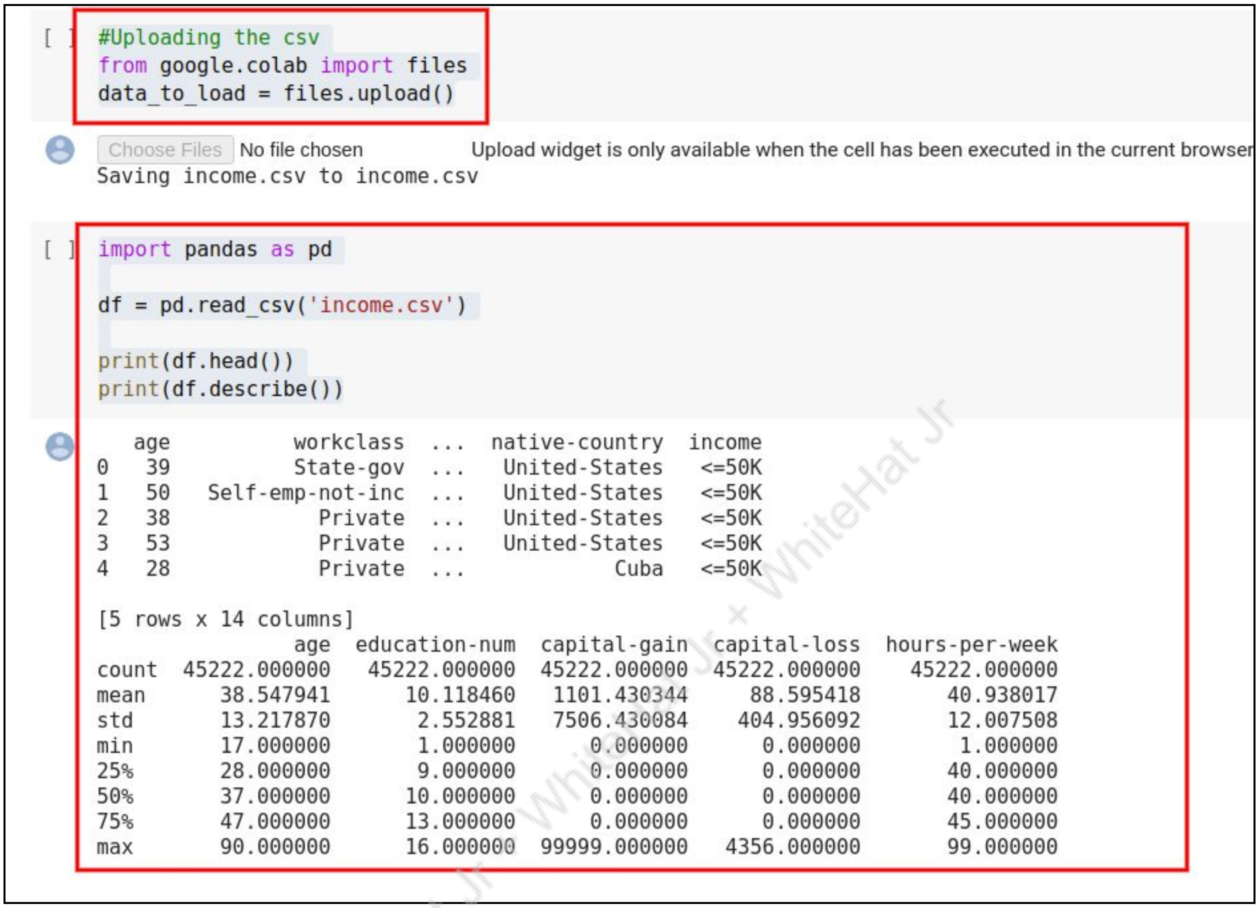


```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score
    from sklearn.preprocessing import StandardScaler

    sc = StandardScaler()

    x_train_2 = sc.fit_transform(x_train_2)
    x_test_2 = sc.fit_transform(x_test_2)

    model_2 = LogisticRegression(random_state = 0)
    model_2.fit(x_train_2, y_train_2)

    y_pred_2 = model_2.predict(x_test_2)

    accuracy = accuracy_score(y_test_2, y_pred_2)
    print(accuracy)

    0.9156626506024096
```

- Here we got the accuracy of 91.6% . We know that the Naive Bayes system outperformed the Logistics regression model.

6. We used another data set of income of people.

7. We uploaded it and read the data.



```
#Uploading the csv
from google.colab import files
data_to_load = files.upload()
```

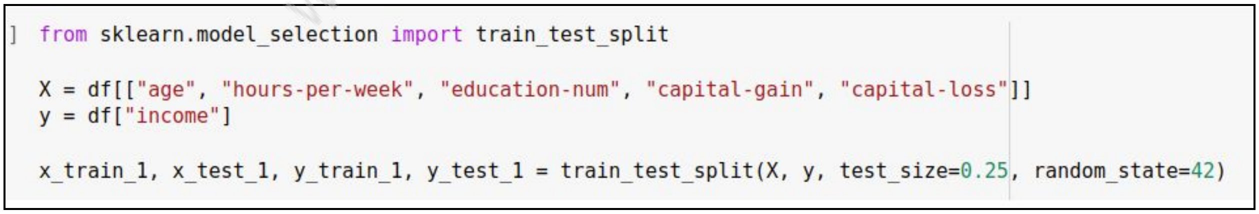
Choose Files
No file chosen
Upload widget is only available when the cell has been executed in the current browser

```
Saving income.csv to income.csv
```

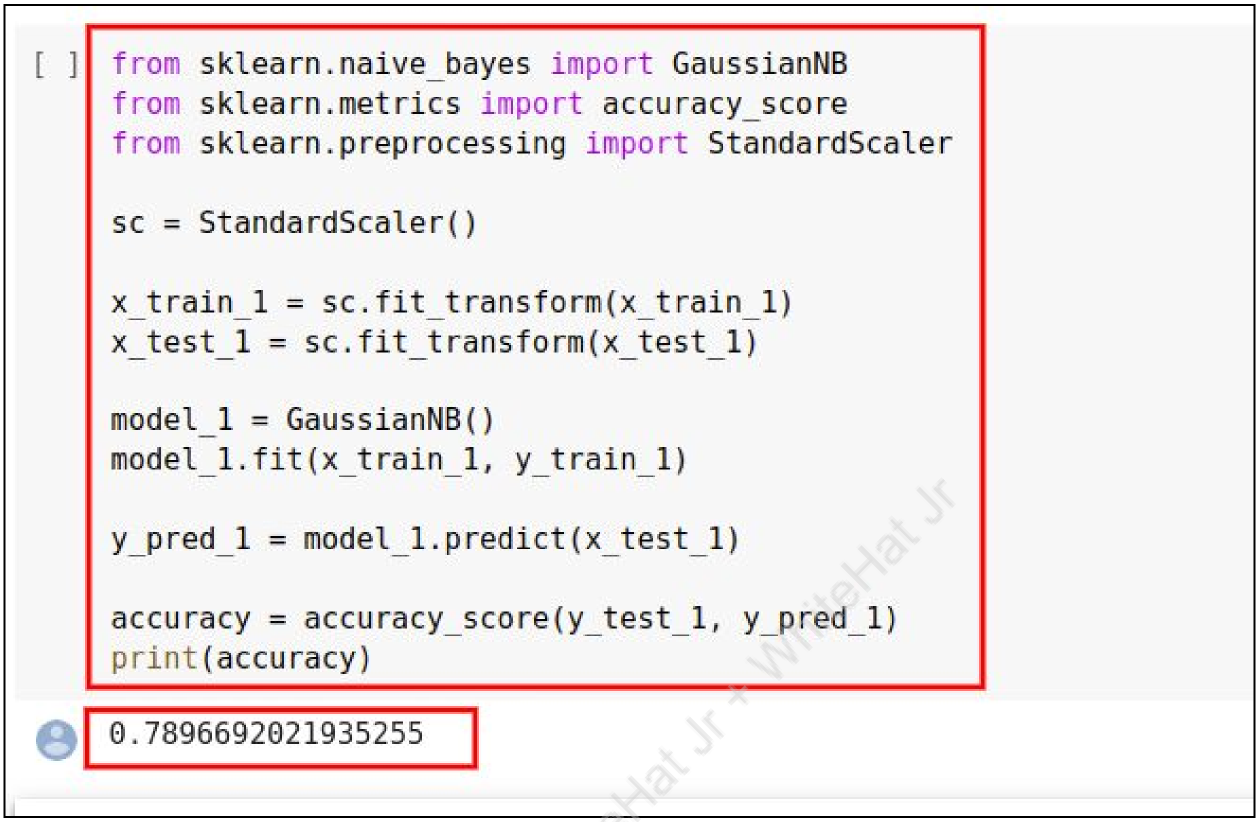
```
import pandas as pd

df = pd.read_csv('income.csv')

print(df.head())
print(df.describe())
```

```
   age         workclass  ...  native-country  income
0   39         State-gov  ...   United-States   <=50K
1   50  Self-emp-not-inc  ...   United-States   <=50K
2   38           Private  ...   United-States   <=50K
3   53           Private  ...   United-States   <=50K
4   28           Private  ...            Cuba   <=50K

[5 rows x 14 columns]
                age  education-num  capital-gain  capital-loss  hours-per-week
count  45222.000000   45222.000000  45222.000000  45222.000000    45222.000000
mean      38.547941      10.118460   1101.430344     88.595418       40.938017
std       13.217870       2.552881   7506.430084    404.956092       12.007508
min       17.000000       1.000000      0.000000      0.000000        1.000000
25%       28.000000       9.000000      0.000000      0.000000       40.000000
50%       37.000000      10.000000      0.000000      0.000000       40.000000
75%       47.000000      13.000000      0.000000      0.000000       45.000000
max       90.000000      16.000000  99999.000000   4356.000000       99.000000
```

8. Then we split the data and trained the Naive Bayes model.



```
from sklearn.model_selection import train_test_split

X = df[["age", "hours-per-week", "education-num", "capital-gain", "capital-loss"]]
y = df["income"]

x_train_1, x_test_1, y_train_1, y_test_1 = train_test_split(X, y, test_size=0.25, random_state=42)
```

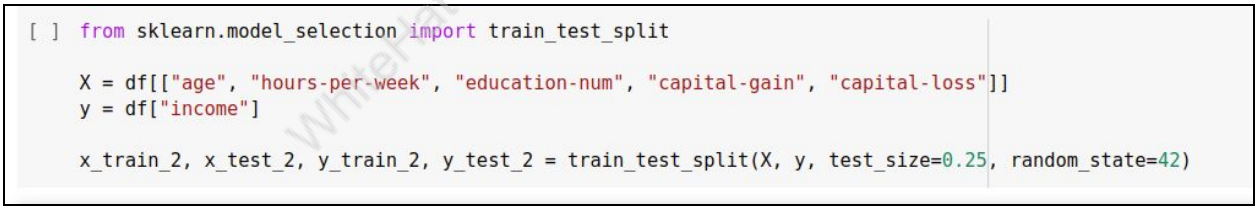


```
[ ] from sklearn.naive_bayes import GaussianNB
    from sklearn.metrics import accuracy_score
    from sklearn.preprocessing import StandardScaler

    sc = StandardScaler()

    x_train_1 = sc.fit_transform(x_train_1)
    x_test_1 = sc.fit_transform(x_test_1)

    model_1 = GaussianNB()
    model_1.fit(x_train_1, y_train_1)

    y_pred_1 = model_1.predict(x_test_1)

    accuracy = accuracy_score(y_test_1, y_pred_1)
    print(accuracy)
```

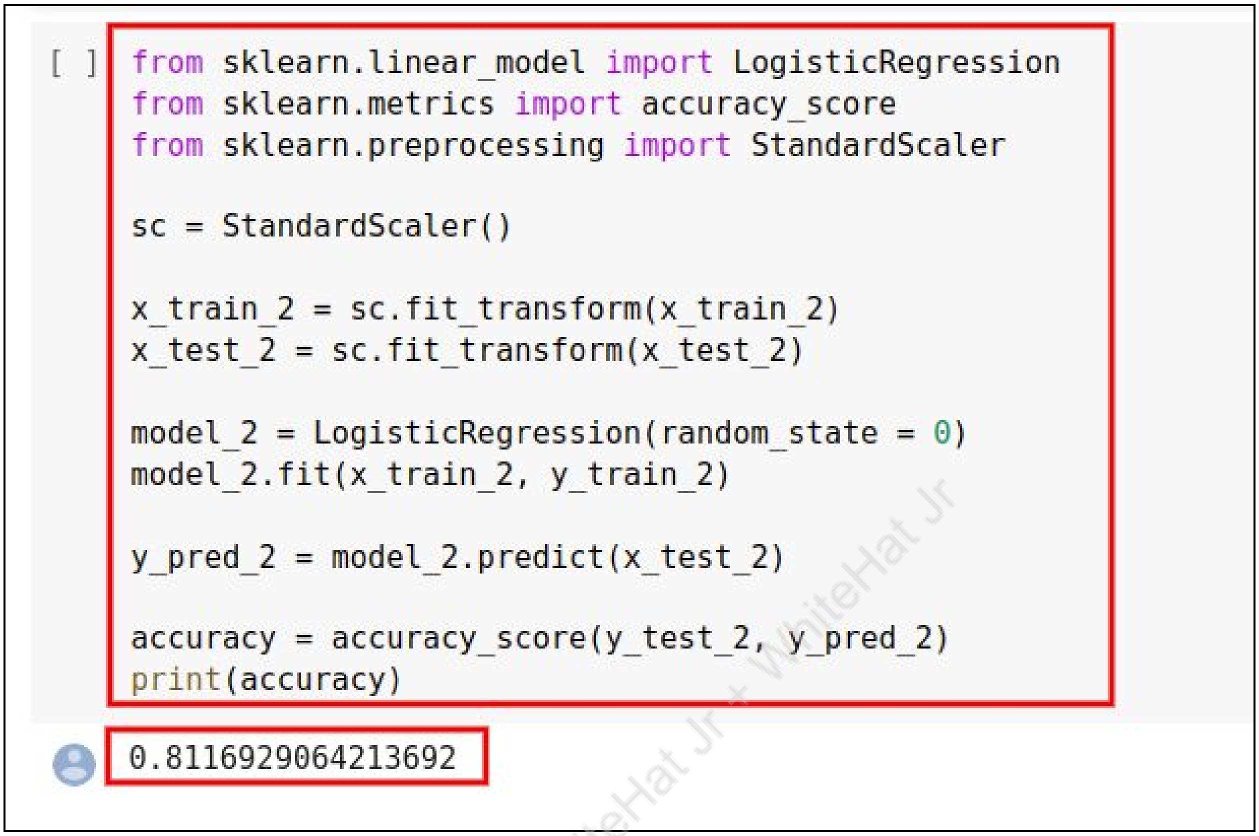
```
0.7896692021935255
```

- We got 78% accuracy here.

9. Then we again split the data to train and test the Logistics model.



```
[ ] from sklearn.model_selection import train_test_split

    X = df[["age", "hours-per-week", "education-num", "capital-gain", "capital-loss"]]
    y = df["income"]

    x_train_2, x_test_2, y_train_2, y_test_2 = train_test_split(X, y, test_size=0.25, random_state=42)
```



```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x_train_2 = sc.fit_transform(x_train_2)
x_test_2 = sc.fit_transform(x_test_2)

model_2 = LogisticRegression(random_state = 0)
model_2.fit(x_train_2, y_train_2)

y_pred_2 = model_2.predict(x_test_2)

accuracy = accuracy_score(y_test_2, y_pred_2)
print(accuracy)
```

```
0.8116929064213692
```

10. We saw that the logistic regression outperformed the Naive Bayes theorem.

What's NEXT?

In the next class, we will learn about neural networks. Next class will be a capstone class so don't forget to bring your parents.

EXTEND YOUR KNOWLEDGE:

Learn more about the Naive Bayes from the following link:
https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/