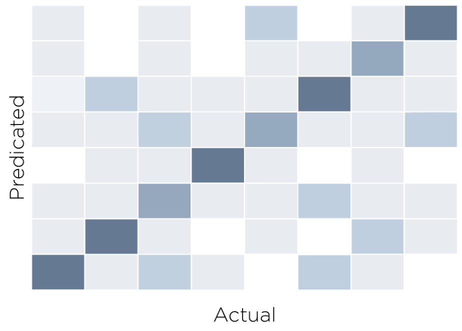
PRO-C117

Confusion Matrix



What is our GOAL for this MODULE?

The goal of this module is to check for the accuracy of the prediction model in linear and multi linear regression using the confusion matrix.

What did we ACHIEVE in the class TODAY?

We checked for the accuracy of the prediction model in linear regression and multi linear regression using the confusion matrix and heatmap

Which CONCEPTS/CODING BLOCKS did we cover today?

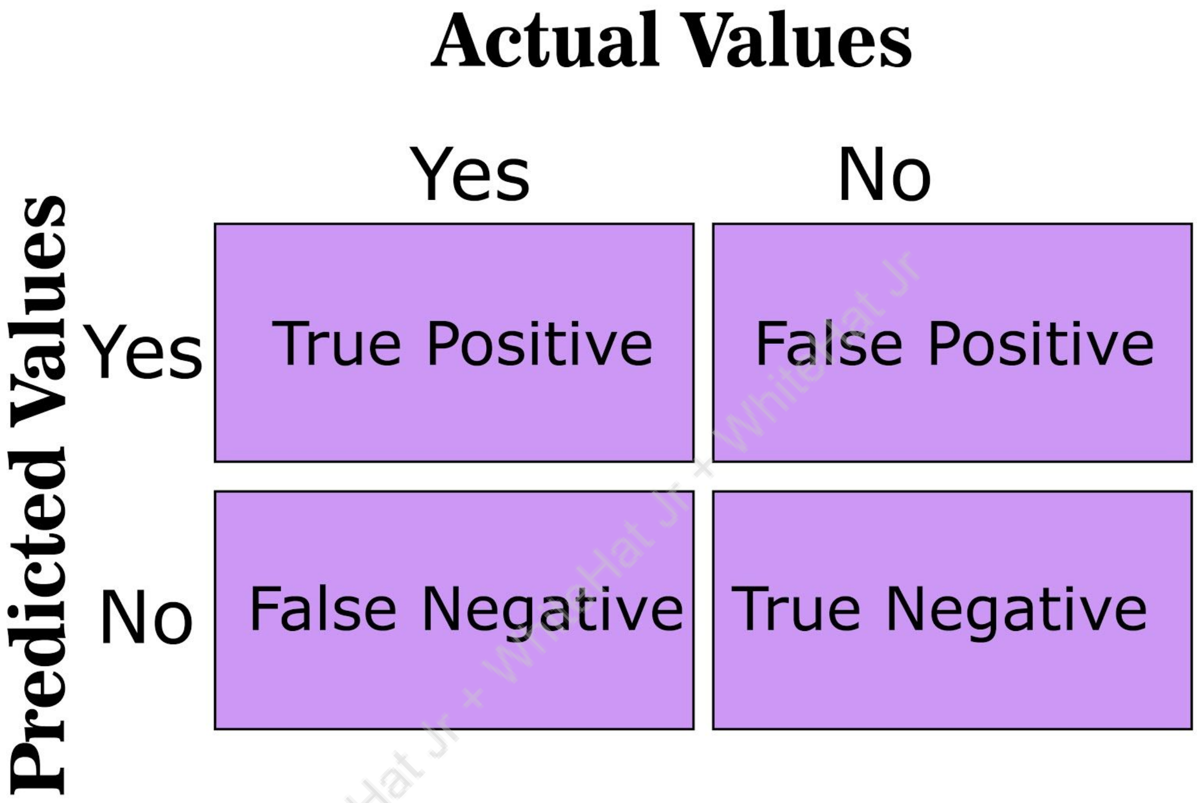
- Usage of sklearn library
- Training and testing of the prediction model
- Using the heatmap and creating a confusion matrix

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

How did we DO the activities?

1. We learned about the confusion matrix.

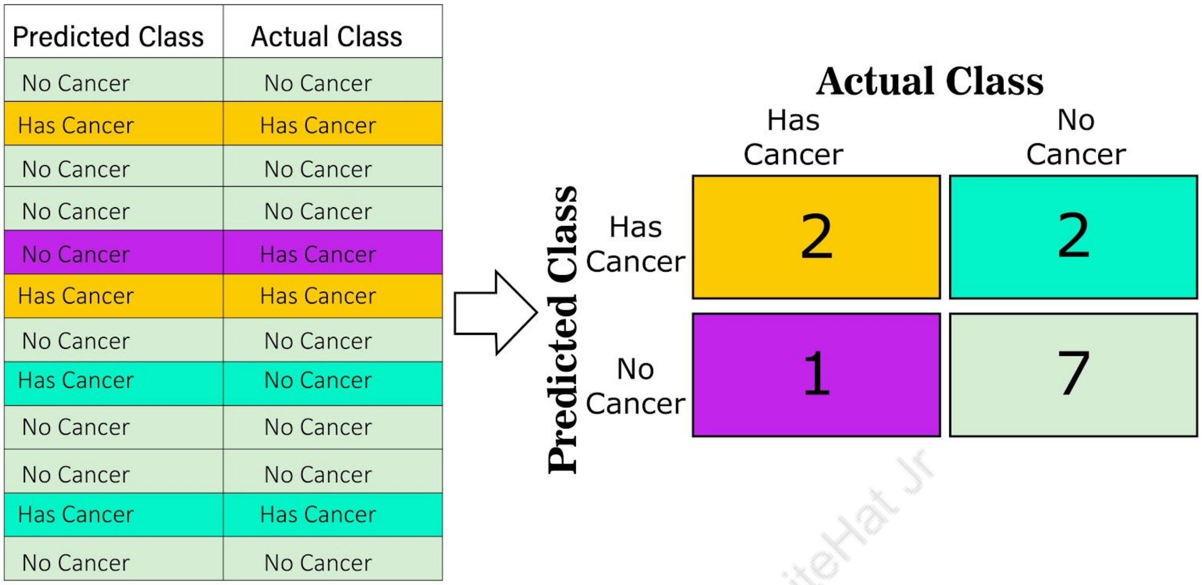


		Actual Values	
		Yes	No
Predicted Values	Yes	True Positive	False Positive
	No	False Negative	True Negative

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

2. We saw how our prediction matrix looks like.

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

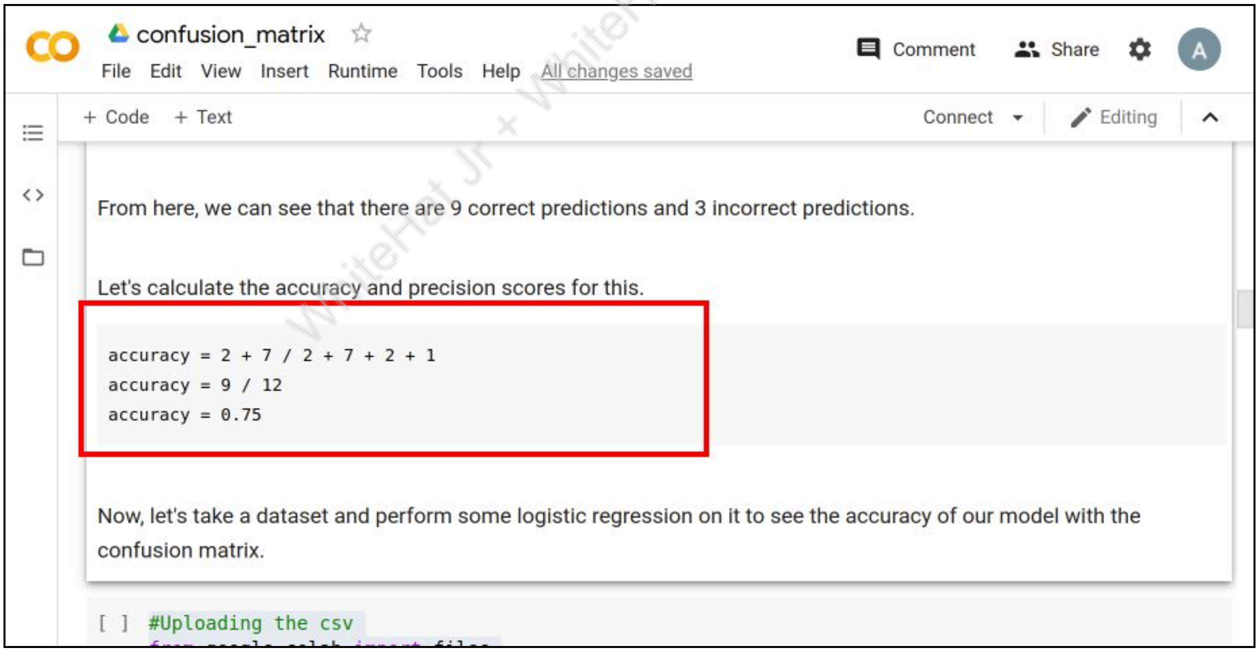


Predicted Class	Actual Class
No Cancer	No Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
No Cancer	No Cancer
No Cancer	Has Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
Has Cancer	No Cancer
No Cancer	No Cancer
No Cancer	No Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer

		Actual Class	
		Has Cancer	No Cancer
Predicted Class	Has Cancer	2	2
	No Cancer	1	7

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

3. We calculated the accuracy of the prediction model.



From here, we can see that there are 9 correct predictions and 3 incorrect predictions.

Let's calculate the accuracy and precision scores for this.

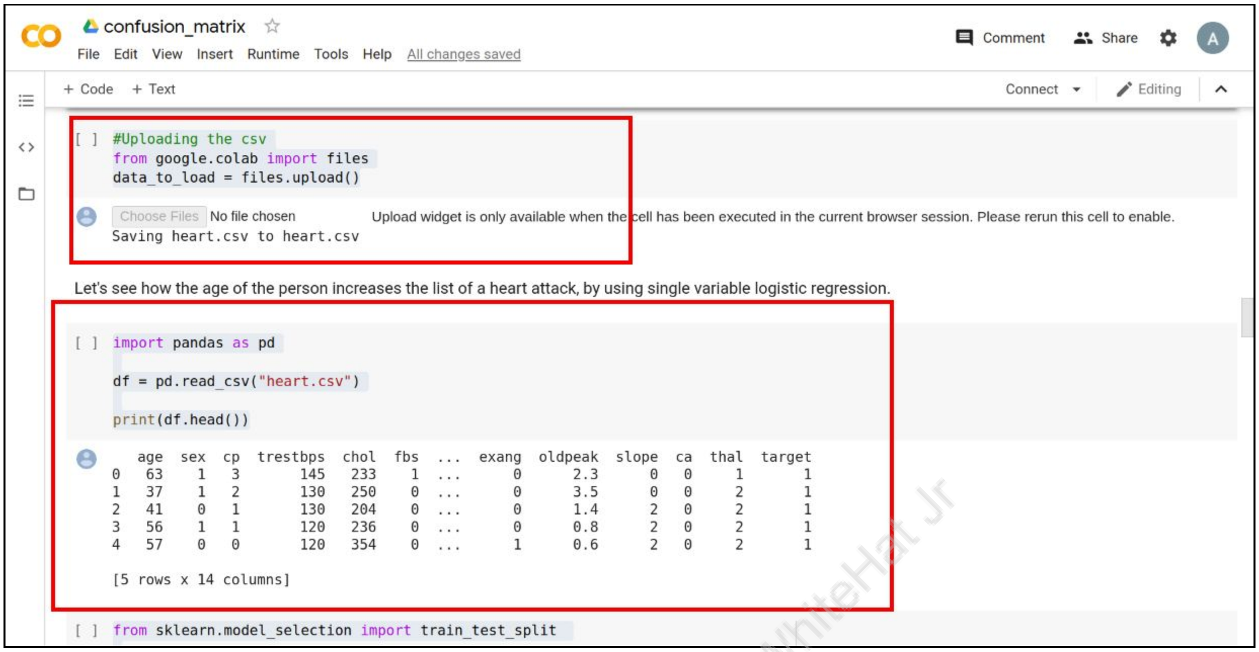
```
accuracy = 2 + 7 / 2 + 7 + 2 + 1
accuracy = 9 / 12
accuracy = 0.75
```

Now, let's take a dataset and perform some logistic regression on it to see the accuracy of our model with the confusion matrix.

```
[ ] #Uploading the csv
    from google.colab import files
```

4. We uploaded the heart attack rate csv file in the colab notebook and using the pandas library we read it and print it's content.

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.



```
[ ] #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 session. Please rerun this cell to enable.

```
Saving heart.csv to heart.csv
```

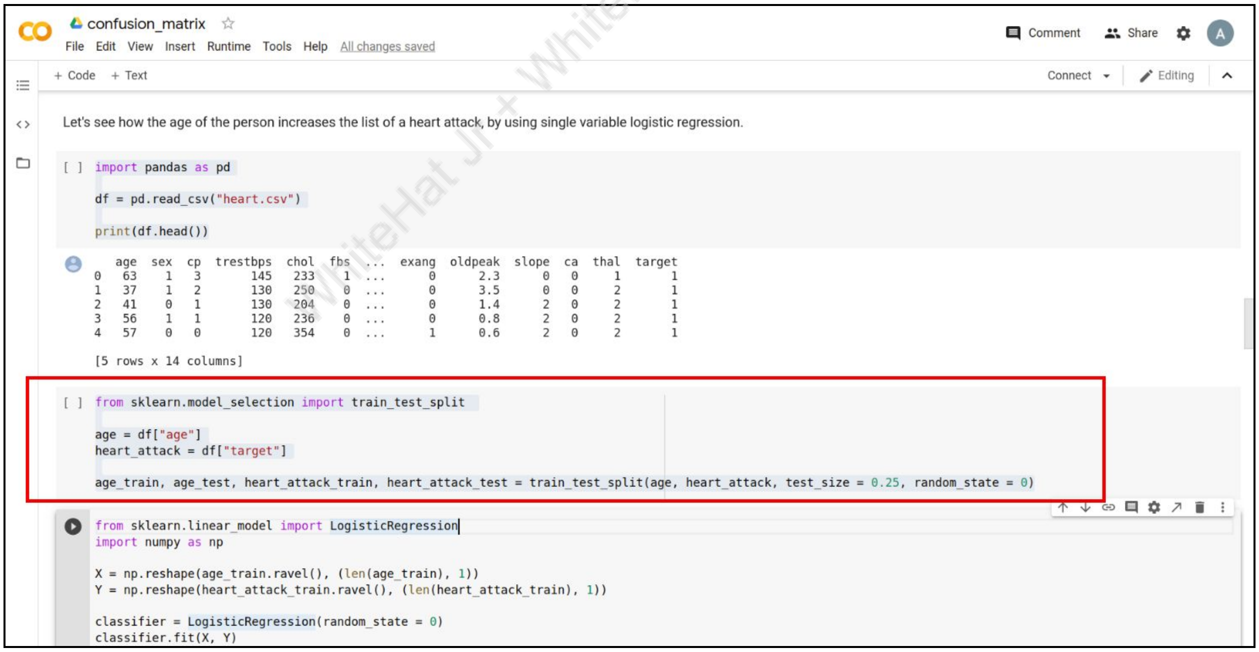
Let's see how the age of the person increases the list of a heart attack, by using single variable logistic regression.

```
[ ] import pandas as pd

    df = pd.read_csv("heart.csv")

    print(df.head())
```

```
   age  sex  cp  trestbps  chol  fbs  ...  exang  oldpeak  slope  ca  thal  target
0   63    1   3       145   233    1  ...      0      2.3      0   0     1       1
1   37    1   2       130   250    0  ...      0      3.5      0   0     2       1
2   41    0   1       130   204    0  ...      0      1.4      2   0     2       1
3   56    1   1       120   236    0  ...      0      0.8      2   0     2       1
4   57    0   0       120   354    0  ...      1      0.6      2   0     2       1

[5 rows x 14 columns]
```

```
[ ] from sklearn.model_selection import train_test_split
```

5. Then we split the data into 75% and 25% to train and test the prediction model.



Let's see how the age of the person increases the list of a heart attack, by using single variable logistic regression.

```
[ ] import pandas as pd

    df = pd.read_csv("heart.csv")

    print(df.head())
```

```
   age  sex  cp  trestbps  chol  fbs  ...  exang  oldpeak  slope  ca  thal  target
0   63    1   3       145   233    1  ...      0      2.3      0   0     1       1
1   37    1   2       130   250    0  ...      0      3.5      0   0     2       1
2   41    0   1       130   204    0  ...      0      1.4      2   0     2       1
3   56    1   1       120   236    0  ...      0      0.8      2   0     2       1
4   57    0   0       120   354    0  ...      1      0.6      2   0     2       1

[5 rows x 14 columns]
```

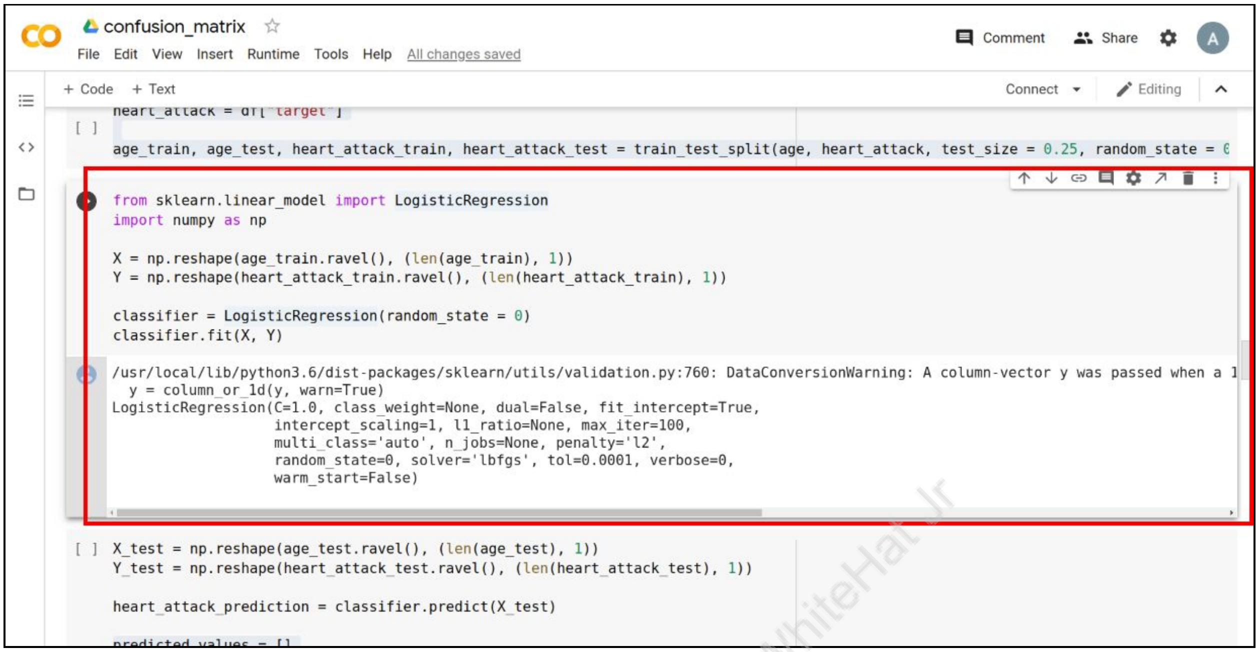
```
[ ] from sklearn.model_selection import train_test_split

    age = df["age"]
    heart_attack = df["target"]

    age_train, age_test, heart_attack_train, heart_attack_test = train_test_split(age, heart_attack, test_size = 0.25, random_state = 0)
```

```
from sklearn.linear_model import LogisticRegression
import numpy as np

X = np.reshape(age_train.ravel(), (len(age_train), 1))
Y = np.reshape(heart_attack_train.ravel(), (len(heart_attack_train), 1))

classifier = LogisticRegression(random_state = 0)
classifier.fit(X, Y)
```

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

Training the data:

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.



```
heart_attack = df['target']
[ ]
age_train, age_test, heart_attack_train, heart_attack_test = train_test_split(age, heart_attack, test_size = 0.25, random_state = 0
```

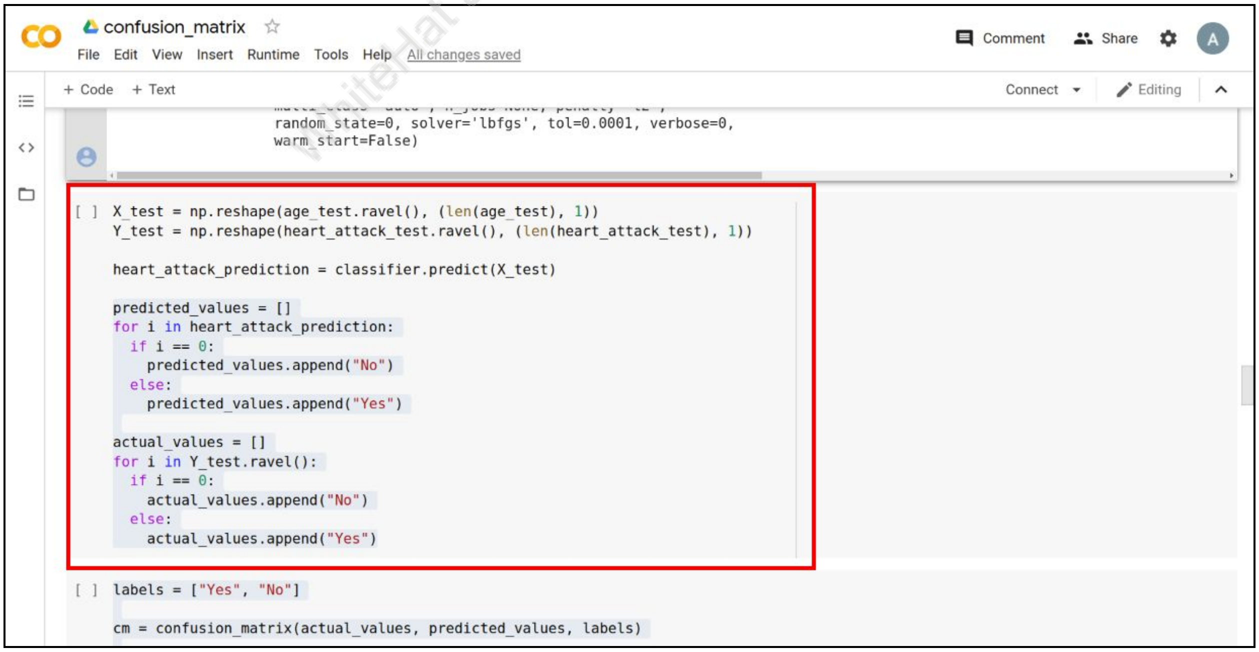
```
from sklearn.linear_model import LogisticRegression
import numpy as np

X = np.reshape(age_train.ravel(), (len(age_train), 1))
Y = np.reshape(heart_attack_train.ravel(), (len(heart_attack_train), 1))

classifier = LogisticRegression(random_state = 0)
classifier.fit(X, Y)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:760: DataConversionWarning: A column-vector y was passed when a 1
  y = column_or_1d(y, warn=True)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
[ ] X_test = np.reshape(age_test.ravel(), (len(age_test), 1))
    Y_test = np.reshape(heart_attack_test.ravel(), (len(heart_attack_test), 1))

    heart_attack_prediction = classifier.predict(X_test)

    predicted_values = []
```

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

6. Then we test the model using the remaining data and substitute the value of 0 as No and 1 as Yes and create 2 arrays for the values in heart attack predictions and actual_values and create a labels array with values Yes and No for the confusion matrix.



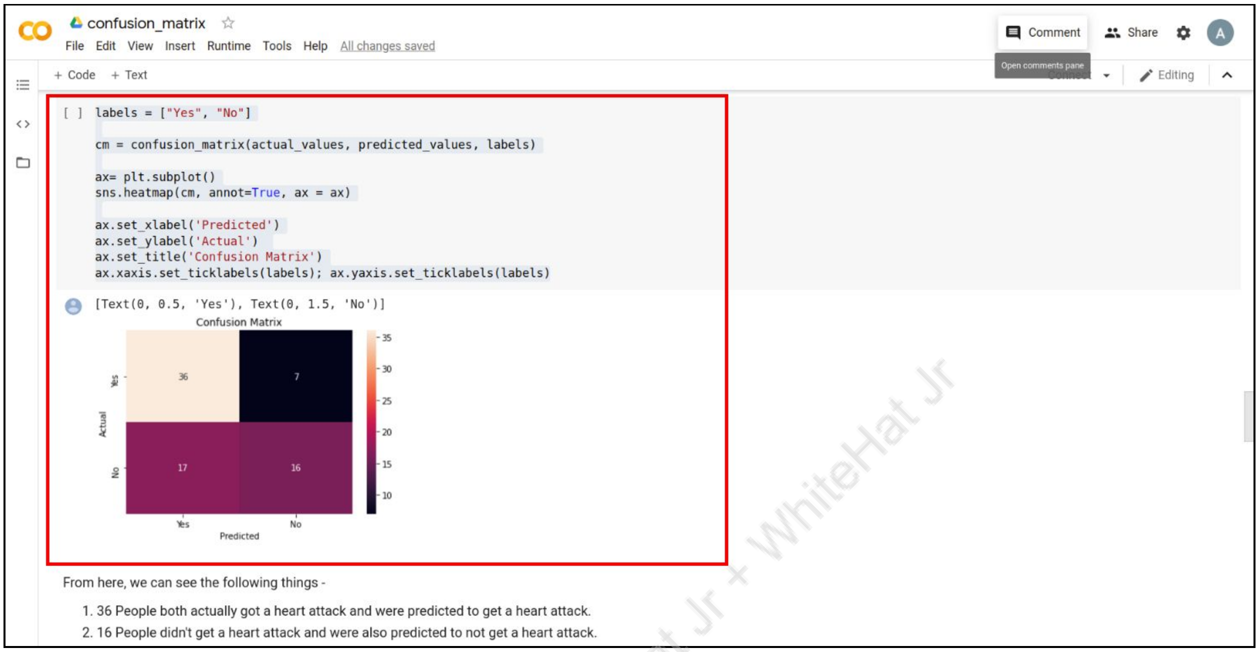
```
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
[ ] X_test = np.reshape(age_test.ravel(), (len(age_test), 1))
    Y_test = np.reshape(heart_attack_test.ravel(), (len(heart_attack_test), 1))

    heart_attack_prediction = classifier.predict(X_test)

    predicted_values = []
    for i in heart_attack_prediction:
      if i == 0:
        predicted_values.append("No")
      else:
        predicted_values.append("Yes")

    actual_values = []
    for i in Y_test.ravel():
      if i == 0:
        actual_values.append("No")
      else:
        actual_values.append("Yes")
```

```
[ ] labels = ["Yes", "No"]

    cm = confusion_matrix(actual_values, predicted_values, labels)
```

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

7. We plotted the heatmap for the data and calculated its accuracy.

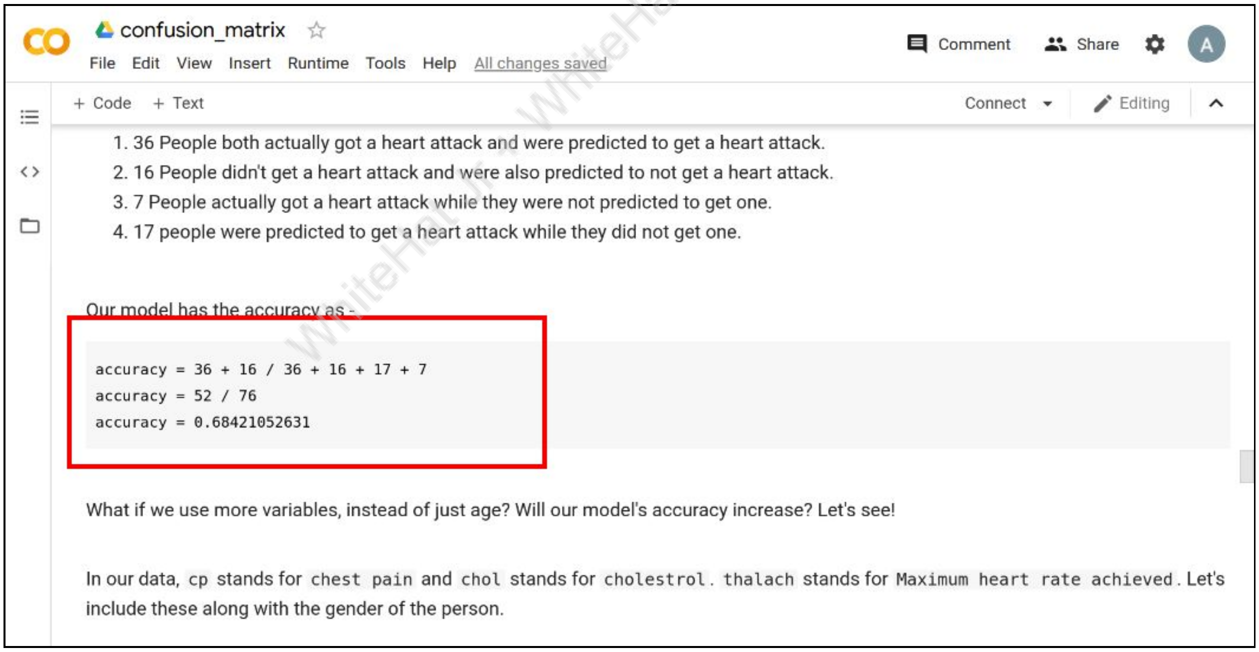


```
labels = ["Yes", "No"]

cm = confusion_matrix(actual_values, predicted_values, labels)

ax= plt.subplot()
sns.heatmap(cm, annot=True, ax = ax)

ax.set_xlabel('Predicted')
ax.set_ylabel('Actual')
ax.set_title('Confusion Matrix')
ax.xaxis.set_ticklabels(labels); ax.yaxis.set_ticklabels(labels)
```

```
[Text(0, 0.5, 'Yes'), Text(0, 1.5, 'No')]
```

From here, we can see the following things -

1. 36 People both actually got a heart attack and were predicted to get a heart attack.
2. 16 People didn't get a heart attack and were also predicted to not get a heart attack.



1. 36 People both actually got a heart attack and were predicted to get a heart attack.
2. 16 People didn't get a heart attack and were also predicted to not get a heart attack.
3. 7 People actually got a heart attack while they were not predicted to get one.
4. 17 people were predicted to get a heart attack while they did not get one.

Our model has the accuracy as -

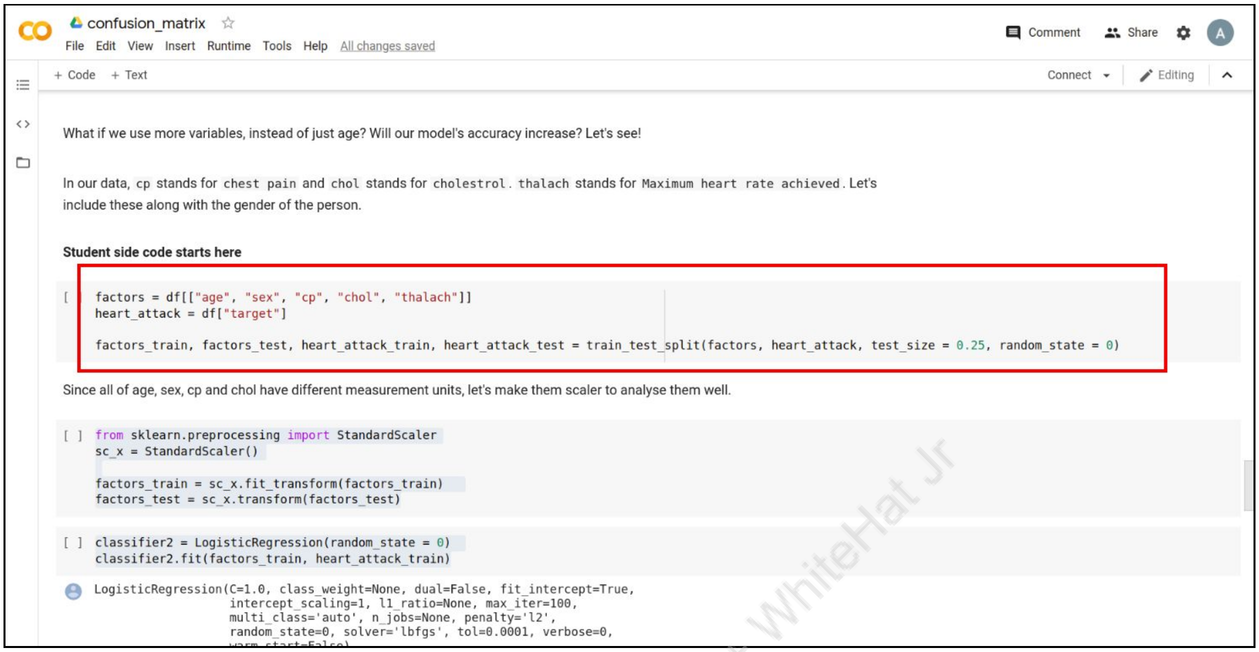
```
accuracy = 36 + 16 / 36 + 16 + 17 + 7
accuracy = 52 / 76
accuracy = 0.68421052631
```

What if we use more variables, instead of just age? Will our model's accuracy increase? Let's see!

In our data, `cp` stands for `chest pain` and `chol` stands for `cholestrol`. `thalach` stands for `Maximum heart rate achieved`. Let's include these along with the gender of the person.

8. We also checked how multiple factors(variables) affect the accuracy of a prediction model.
9. We took multiple factors and created two different data frames and then split that data into 75% and 25%.

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.



What if we use more variables, instead of just age? Will our model's accuracy increase? Let's see!

In our data, `cp` stands for `chest pain` and `chol` stands for `cholestrol`. `thalach` stands for `Maximum heart rate achieved`. Let's include these along with the gender of the person.

Student side code starts here

```
[ ] factors = df[["age", "sex", "cp", "chol", "thalach"]]
    heart_attack = df["target"]

    factors_train, factors_test, heart_attack_train, heart_attack_test = train_test_split(factors, heart_attack, test_size = 0.25, random_state = 0)
```

Since all of age, sex, cp and chol have different measurement units, let's make them scaler to analyse them well.

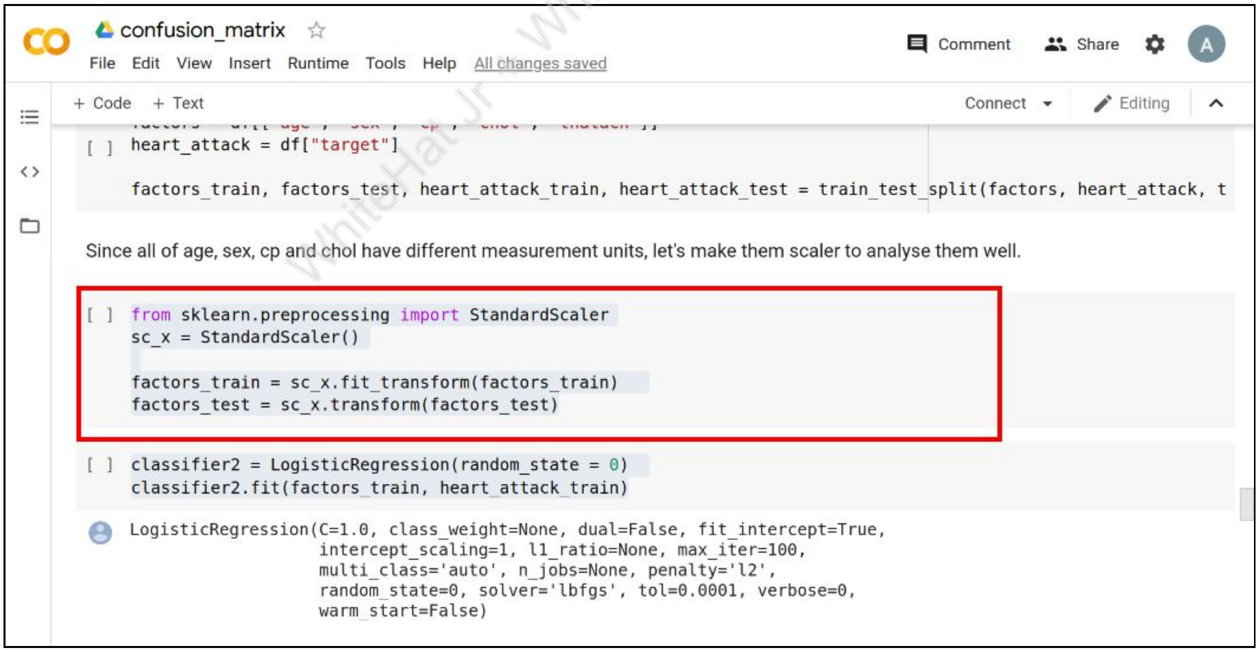
```
[ ] from sklearn.preprocessing import StandardScaler
    sc_x = StandardScaler()

    factors_train = sc_x.fit_transform(factors_train)
    factors_test = sc_x.transform(factors_test)
```

```
[ ] classifier2 = LogisticRegression(random_state = 0)
    classifier2.fit(factors_train, heart_attack_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
```

10. Then we made all the factors scalar as all had different measurements.

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr



```
[ ] heart_attack = df["target"]

    factors_train, factors_test, heart_attack_train, heart_attack_test = train_test_split(factors, heart_attack, t
```

Since all of age, sex, cp and chol have different measurement units, let's make them scaler to analyse them well.

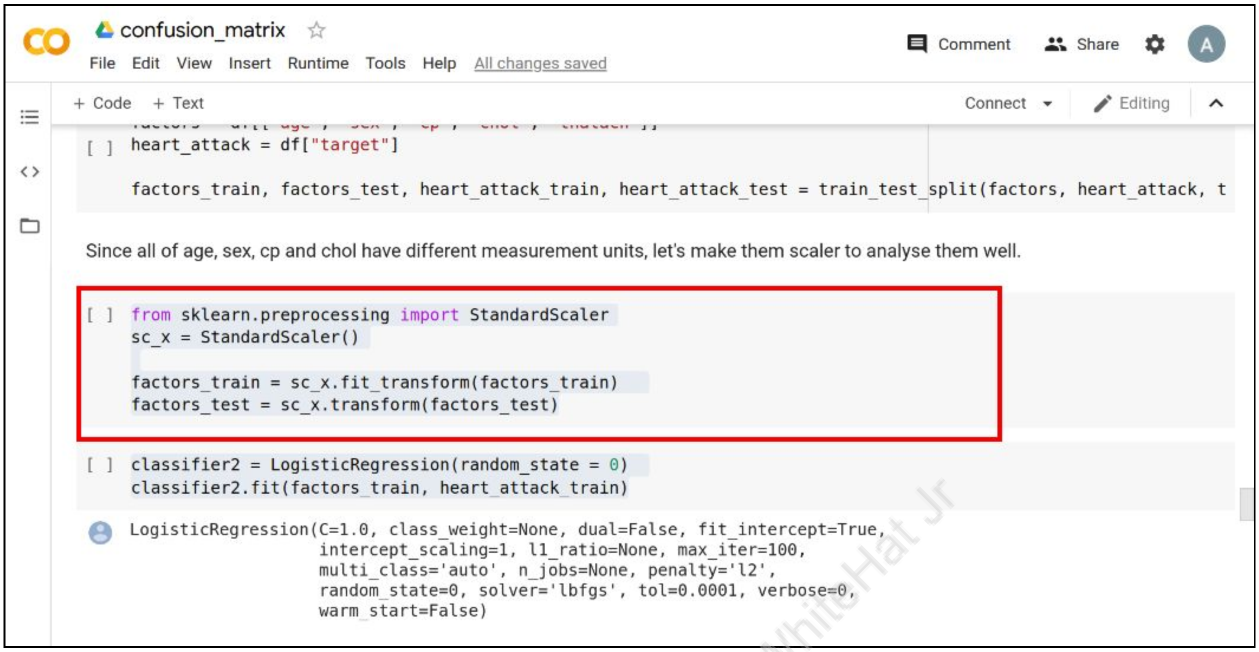
```
[ ] from sklearn.preprocessing import StandardScaler
    sc_x = StandardScaler()

    factors_train = sc_x.fit_transform(factors_train)
    factors_test = sc_x.transform(factors_test)
```

```
[ ] classifier2 = LogisticRegression(random_state = 0)
    classifier2.fit(factors_train, heart_attack_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

11. We trained our model on the remaining data.

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.



```
heart_attack = df["target"]

factors_train, factors_test, heart_attack_train, heart_attack_test = train_test_split(factors, heart_attack, t
```

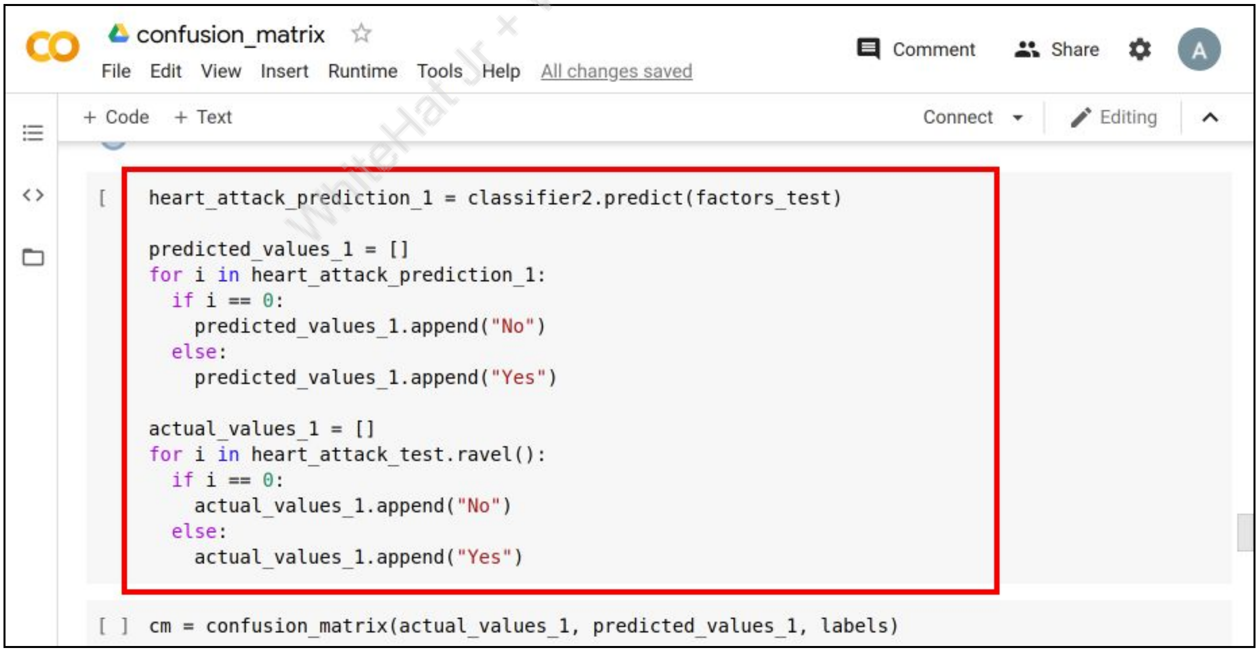
Since all of age, sex, cp and chol have different measurement units, let's make them scaler to analyse them well.

```
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()

factors_train = sc_x.fit_transform(factors_train)
factors_test = sc_x.transform(factors_test)
```

```
classifier2 = LogisticRegression(random_state = 0)
classifier2.fit(factors_train, heart_attack_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

12. We then substituted the values of 0 by No and 1 by Yes and created it’s list for the heat map.



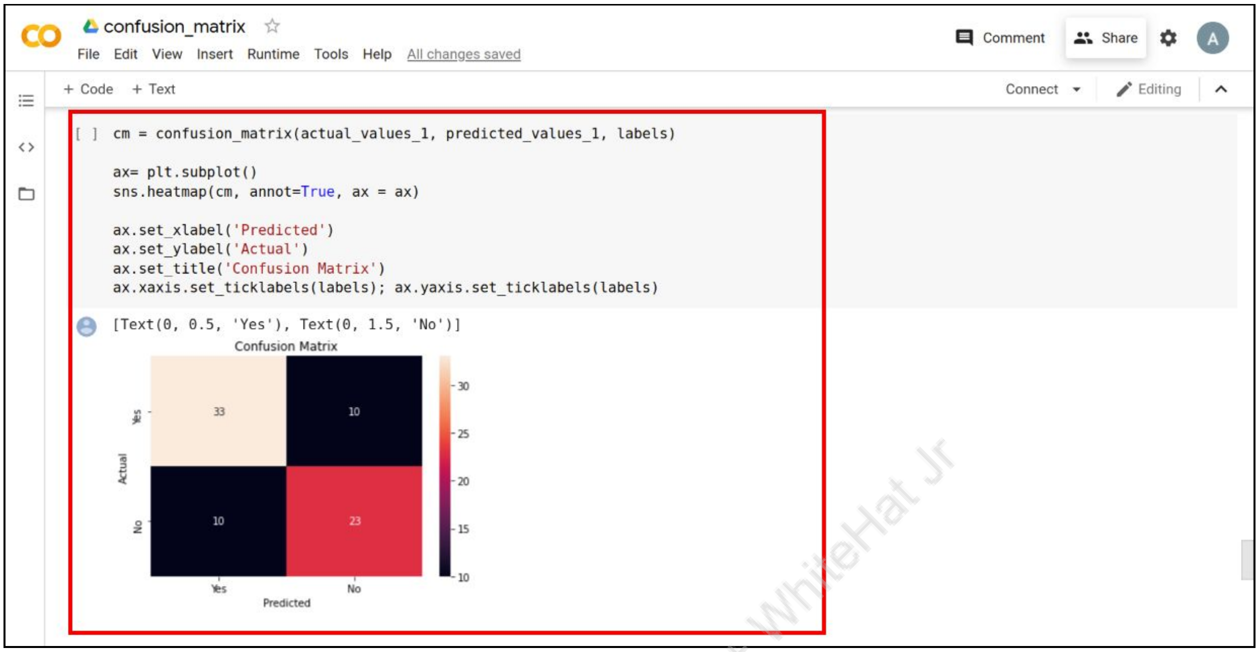
```
heart_attack_prediction_1 = classifier2.predict(factors_test)

predicted_values_1 = []
for i in heart_attack_prediction_1:
  if i == 0:
    predicted_values_1.append("No")
  else:
    predicted_values_1.append("Yes")

actual_values_1 = []
for i in heart_attack_test.ravel():
  if i == 0:
    actual_values_1.append("No")
  else:
    actual_values_1.append("Yes")
```

```
cm = confusion_matrix(actual_values_1, predicted_values_1, labels)
```

13. We plotted the heatmap.

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

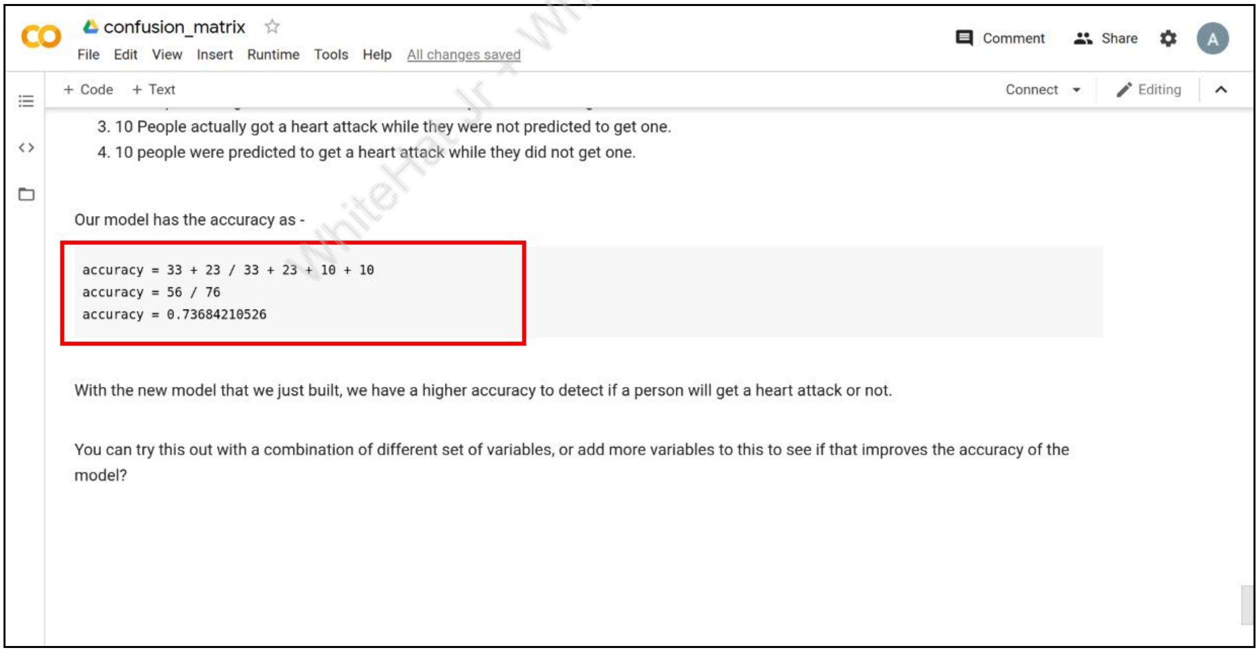
© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.



```
cm = confusion_matrix(actual_values_1, predicted_values_1, labels)

ax= plt.subplot()
sns.heatmap(cm, annot=True, ax = ax)

ax.set_xlabel('Predicted')
ax.set_ylabel('Actual')
ax.set_title('Confusion Matrix')
ax.xaxis.set_ticklabels(labels); ax.yaxis.set_ticklabels(labels)
```

```
[Text(0, 0.5, 'Yes'), Text(0, 1.5, 'No')]
```

14. Calculated the accuracy of the model.



3. 10 People actually got a heart attack while they were not predicted to get one.
4. 10 people were predicted to get a heart attack while they did not get one.

Our model has the accuracy as -

```
accuracy = 33 + 23 / 33 + 23 + 10 + 10
accuracy = 56 / 76
accuracy = 0.73684210526
```

With the new model that we just built, we have a higher accuracy to detect if a person will get a heart attack or not.

You can try this out with a combination of different set of variables, or add more variables to this to see if that improves the accuracy of the model?

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

We concluded that more number of variables affects the accuracy of the model.

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

What's NEXT?

In the next class, we will learn about clustering.

EXTEND YOUR KNOWLEDGE:

Learn more about the confusion matrix from the following doc:
https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.