19ECE363 MACHINE LEARNING

Coding Assignment 5

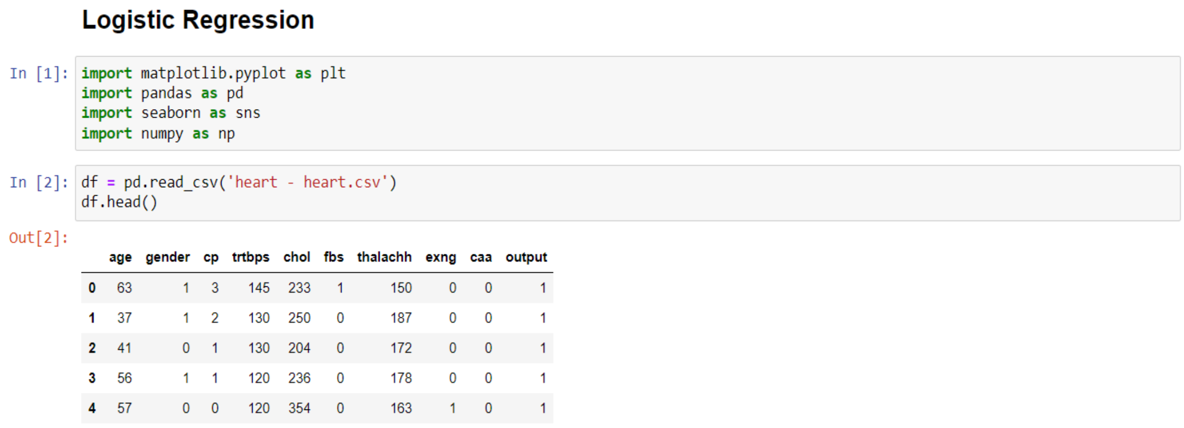
Logistic Regression

- **This assignment will help you understand the implementation and working of a logistic regression ML module**
- **The dataset will be provided to you for this assignment**
- **You have to build a logistic regression module that can predict the chances of heart attack for an individual based on their medical data**

Understanding the dataset

- Import the dataset and try to understand the feature set.
- The dataset contains information about the individuals body vitals
- The columns in the given dataset are as follows:
- age
- gender
- cp: chest pain type
- Value 1: typical angina
- Value 2: atypical angina
- Value 3: non-anginal pain
- Value 4: asymptomatic
- trtbps - resting blood pressure (in mmHg)
- chol - cholestoral (in mg/dl)
- fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- thalachh - maximum heart rate achieved
- exng - exercise induced angina (1 = yes; 0 = no)
- caa - number of major vessels (0-3)
- output - target
    - 0= less chance of heart attack
    - 1= more chance of heart attack

● Identify the target feature in the dataset



Logistic Regression

```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        import seaborn as sns
        import numpy as np
```

```
In [2]: df = pd.read_csv('heart - heart.csv')
        df.head()
```

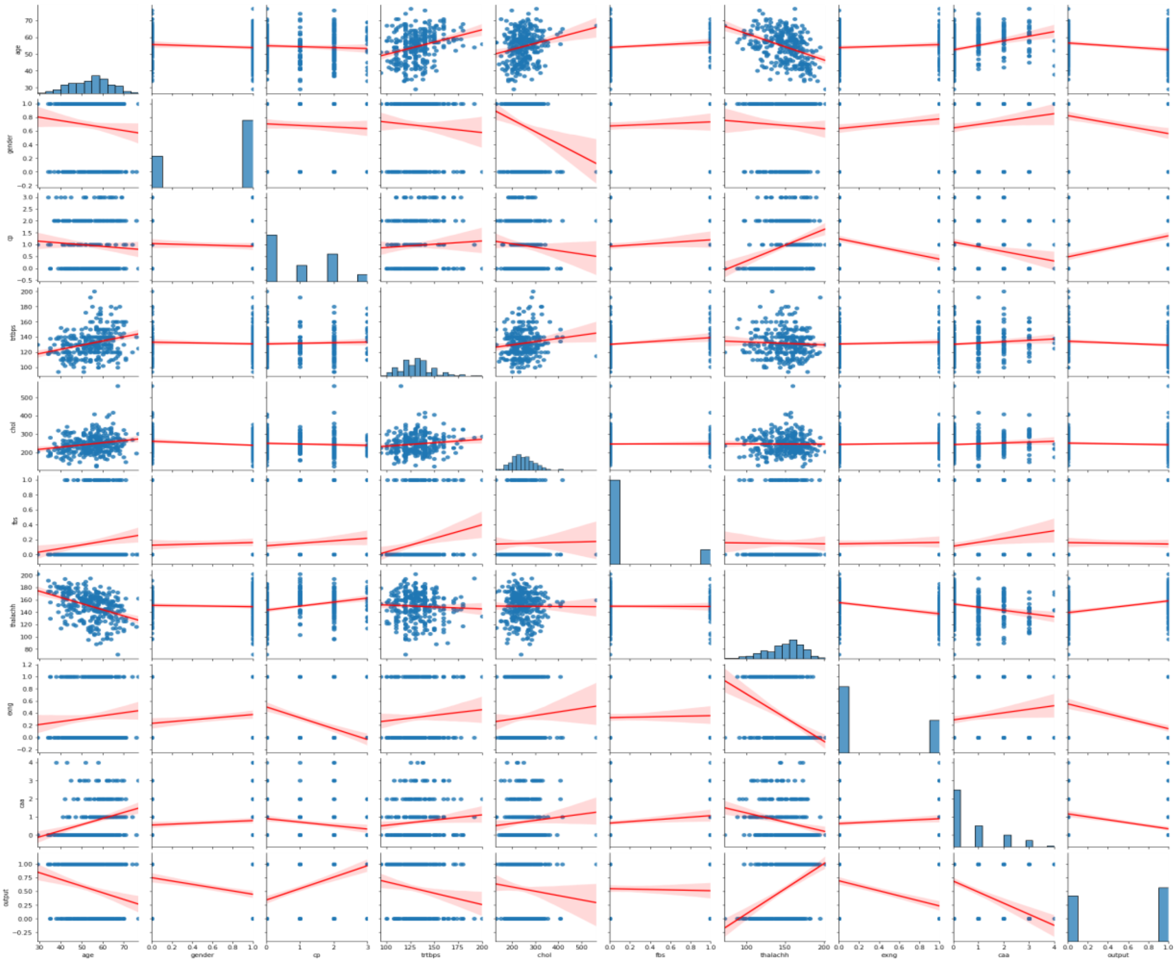
Out[2]:

	age	gender	cp	trtbps	chol	fbs	thalachh	exng	caa	output
0	63	1	3	145	233	1	150	0	0	1
1	37	1	2	130	250	0	187	0	0	1
2	41	0	1	130	204	0	172	0	0	1
3	56	1	1	120	236	0	178	0	0	1
4	57	0	0	120	354	0	163	1	0	1

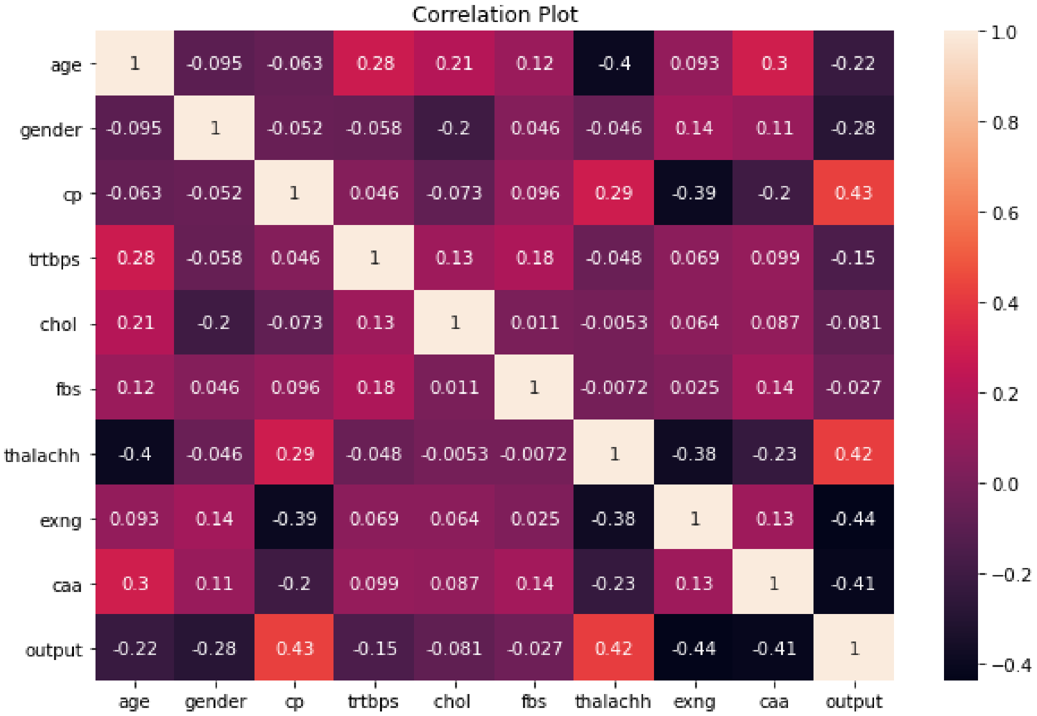
Output is the target feature.

● Do a visual data analysis and try to get an insight about the relationship of feature set with the target feature

Pair Plot:

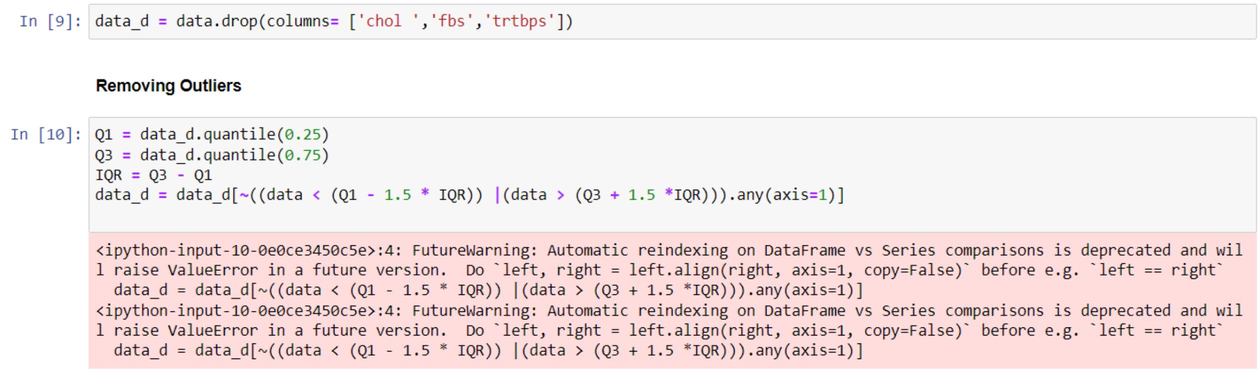


Correlation Plot:



	age	gender	cp	trtbps	chol	fbs	thalachh	exng	caa	output
age	1	-0.095	-0.063	0.28	0.21	0.12	-0.4	0.093	0.3	-0.22
gender	-0.095	1	-0.052	-0.058	-0.2	0.046	-0.046	0.14	0.11	-0.28
cp	-0.063	-0.052	1	0.046	-0.073	0.096	0.29	-0.39	-0.2	0.43
trtbps	0.28	-0.058	0.046	1	0.13	0.18	-0.048	0.069	0.099	-0.15
chol	0.21	-0.2	-0.073	0.13	1	0.011	-0.0053	0.064	0.087	-0.081
fbs	0.12	0.046	0.096	0.18	0.011	1	-0.0072	0.025	0.14	-0.027
thalachh	-0.4	-0.046	0.29	-0.048	-0.0053	-0.0072	1	-0.38	-0.23	0.42
exng	0.093	0.14	-0.39	0.069	0.064	0.025	-0.38	1	0.13	-0.44
caa	0.3	0.11	-0.2	0.099	0.087	0.14	-0.23	0.13	1	-0.41
output	-0.22	-0.28	0.43	-0.15	-0.081	-0.027	0.42	-0.44	-0.41	1

● Do a proper and detailed data cleaning and transformation so that your model is well equipped with a very good accuracy



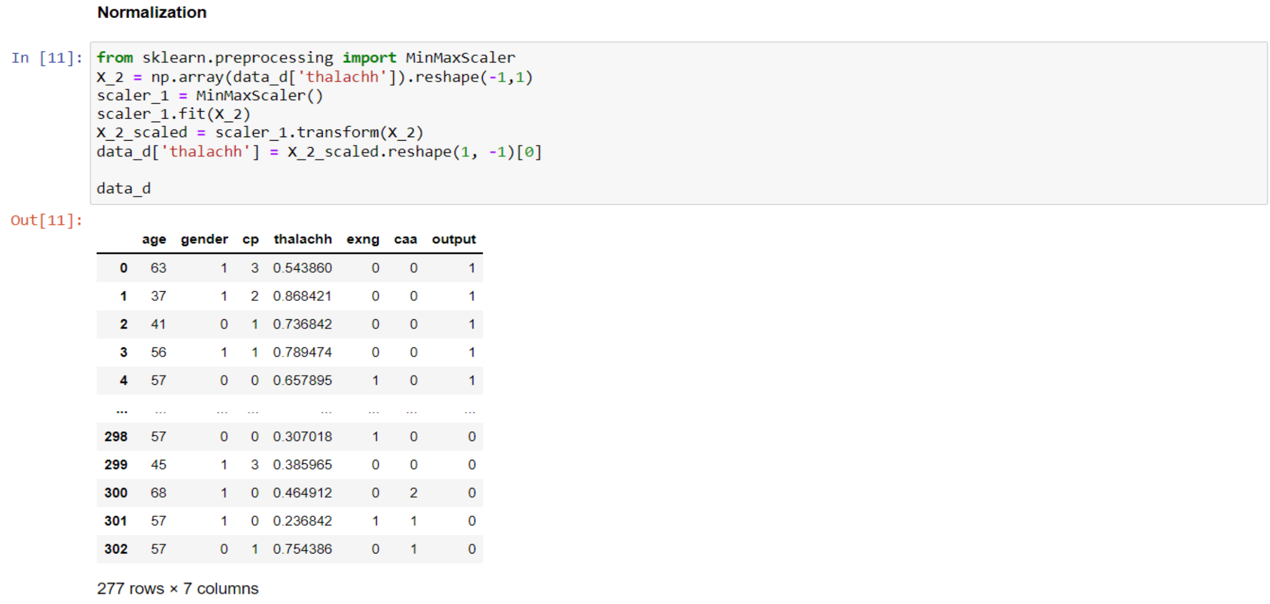
```
In [9]: data_d = data.drop(columns= ['chol ','fbs','trtbps'])
```

Removing Outliers

```
In [10]: Q1 = data_d.quantile(0.25)
         Q3 = data_d.quantile(0.75)
         IQR = Q3 - Q1
         data_d = data_d[~((data < (Q1 - 1.5 * IQR)) |(data > (Q3 + 1.5 *IQR))).any(axis=1)]
```

```
<ipython-input-10-0e0ce3450c5e>:4: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and wil
l raise ValueError in a future version.  Do `left, right = left.align(right, axis=1, copy=False)` before e.g. `left == right`
  data_d = data_d[~((data < (Q1 - 1.5 * IQR)) |(data > (Q3 + 1.5 *IQR))).any(axis=1)]
<ipython-input-10-0e0ce3450c5e>:4: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and wil
l raise ValueError in a future version.  Do `left, right = left.align(right, axis=1, copy=False)` before e.g. `left == right`
  data_d = data_d[~((data < (Q1 - 1.5 * IQR)) |(data > (Q3 + 1.5 *IQR))).any(axis=1)]
```

The reason for dropping the above three features if we observe the both pair plot and correlation plot the three features are not correlated with the output.



Normalization

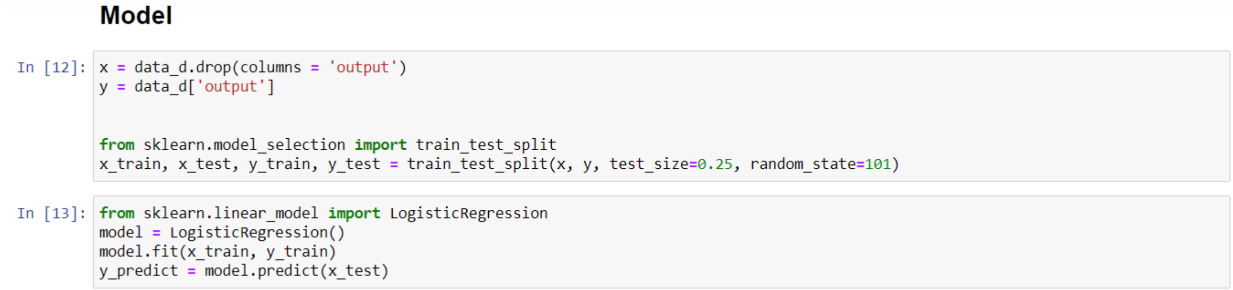
```
In [11]: from sklearn.preprocessing import MinMaxScaler
         X_2 = np.array(data_d['thalachh']).reshape(-1,1)
         scaler_1 = MinMaxScaler()
         scaler_1.fit(X_2)
         X_2_scaled = scaler_1.transform(X_2)
         data_d['thalachh'] = X_2_scaled.reshape(1, -1)[0]

         data_d
```

Out[11]:

	age	gender	cp	thalachh	exng	caa	output
0	63	1	3	0.543860	0	0	1
1	37	1	2	0.868421	0	0	1
2	41	0	1	0.736842	0	0	1
3	56	1	1	0.789474	0	0	1
4	57	0	0	0.657895	1	0	1
...	...	...	...	...	...	...	...
298	57	0	0	0.307018	1	0	0
299	45	1	3	0.385965	0	0	0
300	68	1	0	0.464912	0	2	0
301	57	1	0	0.236842	1	1	0
302	57	0	1	0.754386	0	1	0

277 rows × 7 columns

1. Your task in this assignment is to build a logistic regression model with a good accuracy



Model

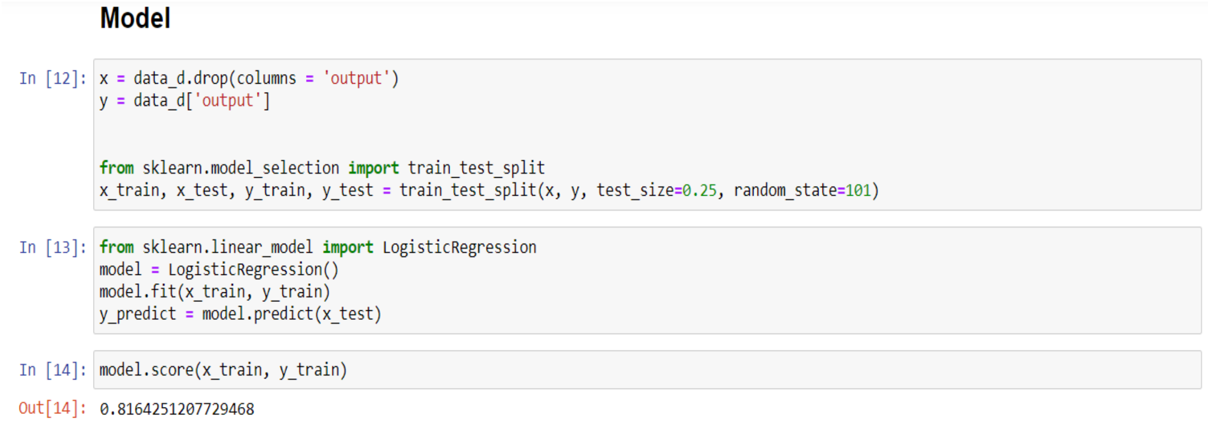
```
In [12]: x = data_d.drop(columns = 'output')
         y = data_d['output']


         from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=101)
```

```
In [13]: from sklearn.linear_model import LogisticRegression
         model = LogisticRegression()
         model.fit(x_train, y_train)
         y_predict = model.predict(x_test)
```

2. List out the features you selected for this model and why

Feature	**Reason For Selection**
age	If we observe from above though it doesn’t have strong correlation (0.22) with the output it is slightly positively correlated with output
Cp(chest pain type)	Cp is strongly correlated with output
gender	Though gender is negatively correlated it has good relation with the target feature
Exng(exercise induced angina)	Exng is negatively correlated **(-0.41)** with the output feature
Caa(number of major vessels)	Caa is also negatively correlated **(-0.44)** with target feature.

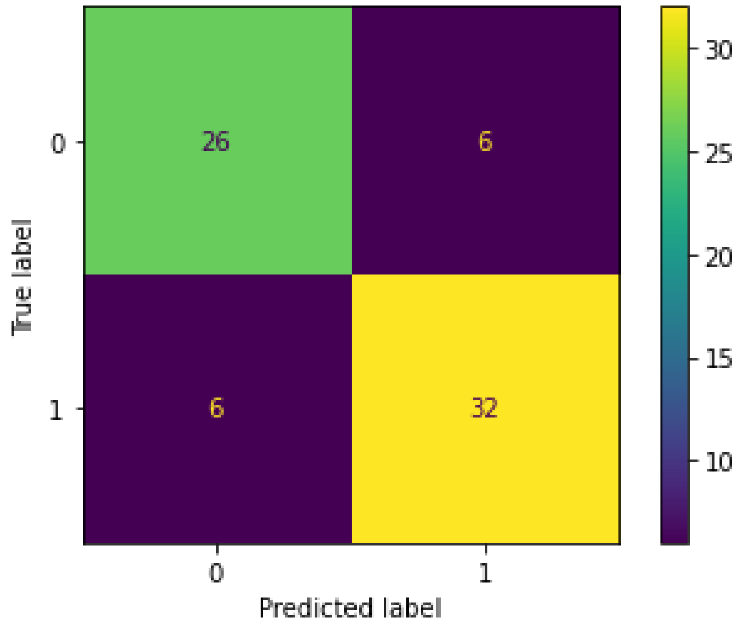
3. Implement the logistic regression model with the selected feature set



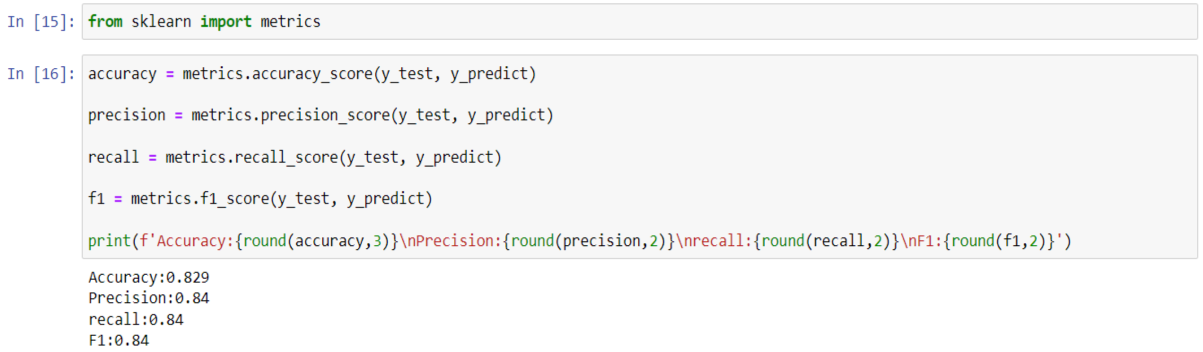
Model

```
In [12]: x = data_d.drop(columns = 'output')
         y = data_d['output']


         from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=101)

In [13]: from sklearn.linear_model import LogisticRegression
         model = LogisticRegression()
         model.fit(x_train, y_train)
         y_predict = model.predict(x_test)

In [14]: model.score(x_train, y_train)

Out[14]: 0.8164251207729468
```

4. Show the confusion matrix and list out the values for TP, FP, TN, FN



5. Find out the accuracy score, precision, recall and F1 Score for your model



```
In [15]: from sklearn import metrics

In [16]: accuracy = metrics.accuracy_score(y_test, y_predict)

         precision = metrics.precision_score(y_test, y_predict)

         recall = metrics.recall_score(y_test, y_predict)

         f1 = metrics.f1_score(y_test, y_predict)

         print(f'Accuracy:{round(accuracy,3)}\nPrecision:{round(precision,2)}\nrecall:{round(recall,2)}\nF1:{round(f1,2)}')

         Accuracy:0.829
         Precision:0.84
         recall:0.84
         F1:0.84
```

Group:

Poludasu Paneendra -AM.EN. U4ECE19044

Koduru Madhusudhan Reddy -AM.EN. U4ECE19127

Lagumsani Vamsi Krishna -AM.EN. U4ECE19131

R.S.V. Mukhesh -AM.EN. U4ECE19147

Rudra Charith Chandan -AM.EN. U4ECE19148

Colab Notebook: Assignment-5