```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import re
```

```
df=pd.read_csv('/content/drive/MyDrive/train_nlp_pro',encoding='ISO-8859-1')
df
```

	S. No.	Message_body	Label
0	1	Rofl. Its true to its name	Non-Spam
1	2	The guy did some bitching but I acted like i'd...	Non-Spam
2	3	Pity, * was in mood for that. So...any other s...	Non-Spam
3	4	Will ü b going to esplanade fr home?	Non-Spam
4	5	This is the 2nd time we have tried 2 contact u...	Spam
...	...	...	...
952	953	hows my favourite person today? r u workin har...	Non-Spam
953	954	How much you got for cleaning	Non-Spam
954	955	Sorry da. I gone mad so many pending works wha...	Non-Spam
955	956	Wat time ü finish?	Non-Spam
956	957	Just glad to be talking to you.	Non-Spam

957 rows × 3 columns

```
df1=pd.read_csv('/content/drive/MyDrive/test_nlp_pro',encoding='ISO-8859-1')
df1
```

	S. No.	Message_body	Label
0	1	UpgrdCentre Orange customer, you may now claim...	Spam
1	2	Loan for any purpose £500 - £75,000. Homeowner...	Spam
2	3	Congrats! Nokia 3650 video camera phone is you...	Spam
3	4	URGENT! Your Mobile number has been awarded wi...	Spam
4	5	Someone has contacted our dating service and e...	Spam
...	...	...	...
120	121	7 wonders in My WORLD 7th You 6th Ur style 5th...	Non-Spam
121	122	Try to do something dear. You read something f...	Non-Spam
122	123	Sun ah... Thk mayb can if dun have anythin on....	Non-Spam
123	124	SYMPTOMS when U are in love: "1.U like listeni...	Non-Spam
124	125	Great. Have a safe trip. Dont panic surrender ...	Non-Spam

125 rows × 3 columns

```
dff=pd.concat([df,df1],axis=0,ignore_index=True)
dff
```

	S. No.	Message_body	Label
0	1	Rofl. Its true to its name	Non-Spam
1	2	The guy did some bitching but I acted like i'd...	Non-Spam
2	3	Pity, * was in mood for that. So...any other s...	Non-Spam
3	4	Will ü b going to esplanade fr home?	Non-Spam
4	5	This is the 2nd time we have tried 2 contact u...	Spam
...	...	...	...
1077	121	7 wonders in My WORLD 7th You 6th Ur style 5th...	Non-Spam
1078	122	Try to do something dear. You read something f...	Non-Spam

```
dff=dff.drop('S. No.',axis=1)
```

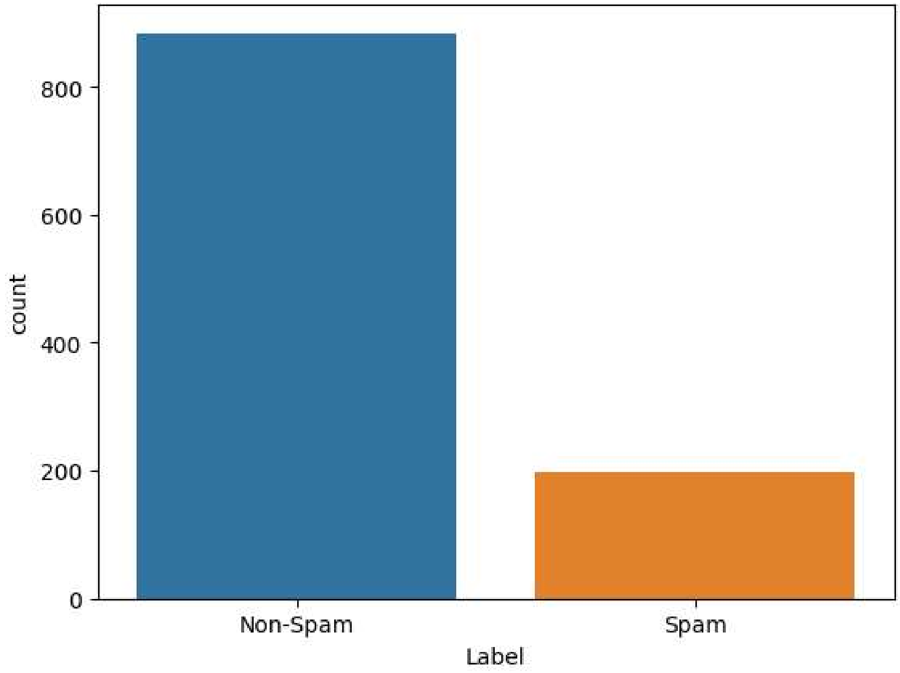
1080	124	SYMPTOMS when U are in love: "1.U like listeni...	Non-Spam

```
df['Label'].value_counts()
```

```
Non-Spam    835
Spam        122
Name: Label, dtype: int64
```

```
sns.countplot(x='Label',data=dff)
```

```
<Axes: xlabel='Label', ylabel='count'>
```



```
dff['Label']=dff['Label'].map({'Non-Spam':1,'Spam':-1})
dff
```

	Message_body	Label
0	Rofl. Its true to its name	1
1	The guy did some bitching but I acted like i'd...	1
2	Pity, * was in mood for that. So...any other s...	1

```
df.dtypes
```

```
S. No.           int64
Message_body    object
Label           object
dtype: object
```

```
nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
mess=dff.Message_body
mess
```

```
0                              Rofl. Its true to its name
1       The guy did some bitching but I acted like i'd...
2       Pity, * was in mood for that. So...any other s...
3                    Will ü b going to esplanade fr home?
4       This is the 2nd time we have tried 2 contact u...
                              ...
1077    7 wonders in My WORLD 7th You 6th Ur style 5th...
1078    Try to do something dear. You read something f...
1079    Sun ah... Thk mayb can if dun have anythin on....
1080    SYMPTOMS when U are in love: "1.U like listeni...
1081    Great. Have a safe trip. Dont panic surrender ...
Name: Message_body, Length: 1082, dtype: object
```

```
# preprocessing

from nltk.tokenize import TweetTokenizer
tk=TweetTokenizer()
mess=mess.apply(lambda x:tk.tokenize(x)).apply(lambda x:" ".join(x))
mess
```

```
0                                          rofl true name
1       guy bitch act like interest buy someth els nex...
2                                    piti mood ani suggest
3                                    b go esplanad fr home
4       2nd time tri 2 contact u u 750 pound prize 2 c...
                              ...
1077    7 wonder world 7th 6th ur style 5th ur smile 4...
1078                     tri someth dear read someth exam
1079    sun ah thk mayb dun anythin thk book e lesson ...
1080    symptom u love 1 u like listen song 2 u get st...
1081                   great safe trip dont panic surrend
Name: Message_body, Length: 1082, dtype: object
```

```
# special charachter remove
# re : regular expression we remove special charachters

mess=mess.str.replace('[^a-zA-Z-0-9]+',' ')
mess
```

```
<ipython-input-46-4ab931728cb5>:4: FutureWarning: The default value of regex will change from True to False in a future
  mess=mess.str.replace('[^a-zA-Z-0-9]+',' ')
0                               Rofl Its true to its name
```

```
     1       The guy did some bitching but I acted like i d...
     2       Pity was in mood for that So any other suggest...
     3                      Will b going to esplanade fr home
     4       This is the 2nd time we have tried 2 contact u...
                                   ...
     1077    7 wonders in My WORLD 7th You 6th Ur style 5th...
     1078    Try to do something dear You read something fo...
     1079    Sun ah Thk mayb can if dun have anythin on Thk...
     1080    SYMPTOMS when U are in love 1 U like listening...
     1081     Great Have a safe trip Dont panic surrender all
     Name: Message_body, Length: 1082, dtype: object
```

```
# stemming or lematization

from nltk.stem import SnowballStemmer
ss=SnowballStemmer('english')
mess=mess.apply(lambda x:[ss.stem(i.lower()) for i in tk.tokenize(x)]).apply(lambda x:" ".join(x))
mess
```

```
     0                                 rofl it true to it name
     1       the guy did some bitch but i act like i d be i...
     2          piti was in mood for that so ani other suggest
     3                           will b go to esplanad fr home
     4       this is the 2nd time we have tri 2 contact u u...
                                   ...
     1077    7 wonder in my world 7th you 6th ur style 5th ...
     1078       tri to do someth dear you read someth for exam
     1079    sun ah thk mayb can if dun have anythin on thk...
     1080    symptom when u are in love 1 u like listen son...
     1081        great have a safe trip dont panic surrend all
     Name: Message_body, Length: 1082, dtype: object
```

```
# stopwords

nltk.download('stopwords')
from nltk.corpus import stopwords
sw=stopwords.words('english')
mess=mess.apply(lambda x:[i for i in tk.tokenize(x) if i not in sw]).apply(lambda x:' '.join(x))
mess
```

```
     [nltk_data] Downloading package stopwords to /root/nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
     0                                          rofl true name
     1       guy bitch act like interest buy someth els nex...
     2                                   piti mood ani suggest
     3                                   b go esplanad fr home
     4       2nd time tri 2 contact u u 750 pound prize 2 c...
                                   ...
     1077    7 wonder world 7th 6th ur style 5th ur smile 4...
     1078                     tri someth dear read someth exam
     1079    sun ah thk mayb dun anythin thk book e lesson ...
     1080    symptom u love 1 u like listen song 2 u get st...
     1081                   great safe trip dont panic surrend
     Name: Message_body, Length: 1082, dtype: object
```

```
# vectorization

from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()
train_data=vec.fit_transform(mess)
print(train_data)         #x
```

```
       (0, 1852)     0.5086856793431559
       (0, 2734)     0.5352804139572925
       (0, 2264)     0.6743246681420617
       (1, 1191)     0.19084717659108363
       (1, 2794)     0.2620897628588603
       (1, 1236)     0.3166286972359124
       (1, 2881)     0.22002695063463382
       (1, 1882)     0.25587622919424974
       (1, 1035)     0.29329608266677626
       (1, 2455)     0.26551480891862445
       (1, 677)      0.26551480891862445
       (1, 1478)     0.307577621142851
       (1, 1626)     0.20980773882403927
       (1, 396)      0.3419878575694143
       (1, 607)      0.36211655551990307
       (1, 1309)     0.2588858462402129
       (2, 2555)     0.5165656915002457
       (2, 463)      0.36716239650585775
       (2, 1805)     0.5469696796701571
       (2, 2044)     0.5469696796701571
       (3, 1386)     0.3883344606933877
       (3, 1187)     0.630740525885995
       (3, 1063)     0.5956800313099777
       (3, 1265)     0.3106896135077221
       (4, 1858)     0.30932958639486785
       :     :
       (1079, 1822)  0.2633677871797729
       (1079, 2560)  0.2544650331411059
       (1079, 1614)  0.2410408066920934
       (1079, 628)   0.23101411970886698
       (1079, 2654)  0.4820816133841868
       (1079, 1737)  0.21063371947658105
       (1079, 1008)  0.21634301112384327
       (1079, 424)   0.23101411970886698
       (1080, 589)   0.37025990523411034
       (1080, 2588)  0.37025990523411034
       (1080, 1637)  0.3237491068715447
       (1080, 462)   0.3237491068715447
       (1080, 2459)  0.33507584451064965
       (1080, 2525)  0.2208005656522519
       (1080, 1676)  0.2249749608215637
       (1080, 2331)  0.21572034231434808
       (1080, 1253)  0.36602572795540395
       (1080, 1626)  0.21452593732655792
       (1080, 1852)  0.27931042764086844
       (1081, 2573)  0.4791624063199324
       (1081, 1989)  0.4791624063199324
       (1081, 983)   0.30819278764115643
       (1081, 2288)  0.4525275695553008
       (1081, 2731)  0.3734390895085233
       (1081, 1296)  0.31872563060534453
```

```
train_data.shape
```

```
     (1082, 3005)
```

```
y=dff['Label'].values
y
```

```
     array([1, 1, 1, ..., 1, 1, 1])
```

```
# train test split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(train_data,y,test_size=0.30,random_state=42)
print(x_train)
```

```
       (1, 2979)     0.7480661157830715
       (1, 2650)     0.6636242057197945
       (2, 101)      0.2646384845778062
       (2, 2199)     0.20546178792598968
       (2, 1680)     0.2646384845778062
       (2, 2172)     0.2646384845778062
       (2, 819)      0.2450149523023616
       (2, 246)      0.5292769691556124
       (2, 882)      0.18475606226073812
       (2, 2963)     0.23208926203167904
       (2, 2249)     0.4539192204143587
       (2, 2035)     0.18988571408523777
       (2, 1789)     0.1641839552513953
       (2, 2846)     0.19176000409098287
       (2, 687)      0.11700723359841446
       (3, 2992)     0.4744252349259739
       (3, 628)      0.39876568684043007
       (3, 3001)     0.3683374213716801
```

```
  (3, 583)	0.36358600039116745
  (3, 2857)	0.28947411942667833
  (3, 1339)	0.3987656868404307
  (3, 687)	0.209762326807933б
  (3, 1265)	0.24744659066934233
  (4, 1820)	0.1985749666919
  (4, 1566)	0.3971499333838
  :	:
  (755, 2021)	0.20391520180466627
  (755, 1711)	0.17643010269024142
  (755, 1075)	0.22217277260949916
  (755, 2972)	0.21378143666117558
  (755, 1277)	0.1607211279702954
  (755, 2596)	0.19610690236832126
  (755, 545)	0.1896444559736829
  (755, 2857)	0.31613109434209863
  (755, 1329)	0.19853388002142303
  (755, 2913)	0.19853388002142303
  (755, 2731)	0.21378143666117558
  (755, 1676)	0.1666711577830679
  (756, 2181)	0.3286987682985329
  (756, 429)	0.3286987682985329
  (756, 1342)	0.3286987682985329
  (756, 2946)	0.2974640673221203
  (756, 564)	0.2974640673221203
  (756, 1940)	0.2791929385296804
  (756, 975)	0.2791929385296804
  (756, 2654)	0.2722465777498161
  (756, 704)	0.24101187677340344
  (756, 1591)	0.20519062894156473
  (756, 473)	0.2272507236230591
  (756, 1664)	0.21488400980986913
  (756, 1035)	0.2662293663457076
```

```
# model creation
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
s_model=SVC()
n_model=MultinomialNB()
k_model=KNeighborsClassifier()
r_model=RandomForestClassifier()
d_model=DecisionTreeClassifier()
lst_model=[s_model,n_model,k_model,r_model,d_model]
```

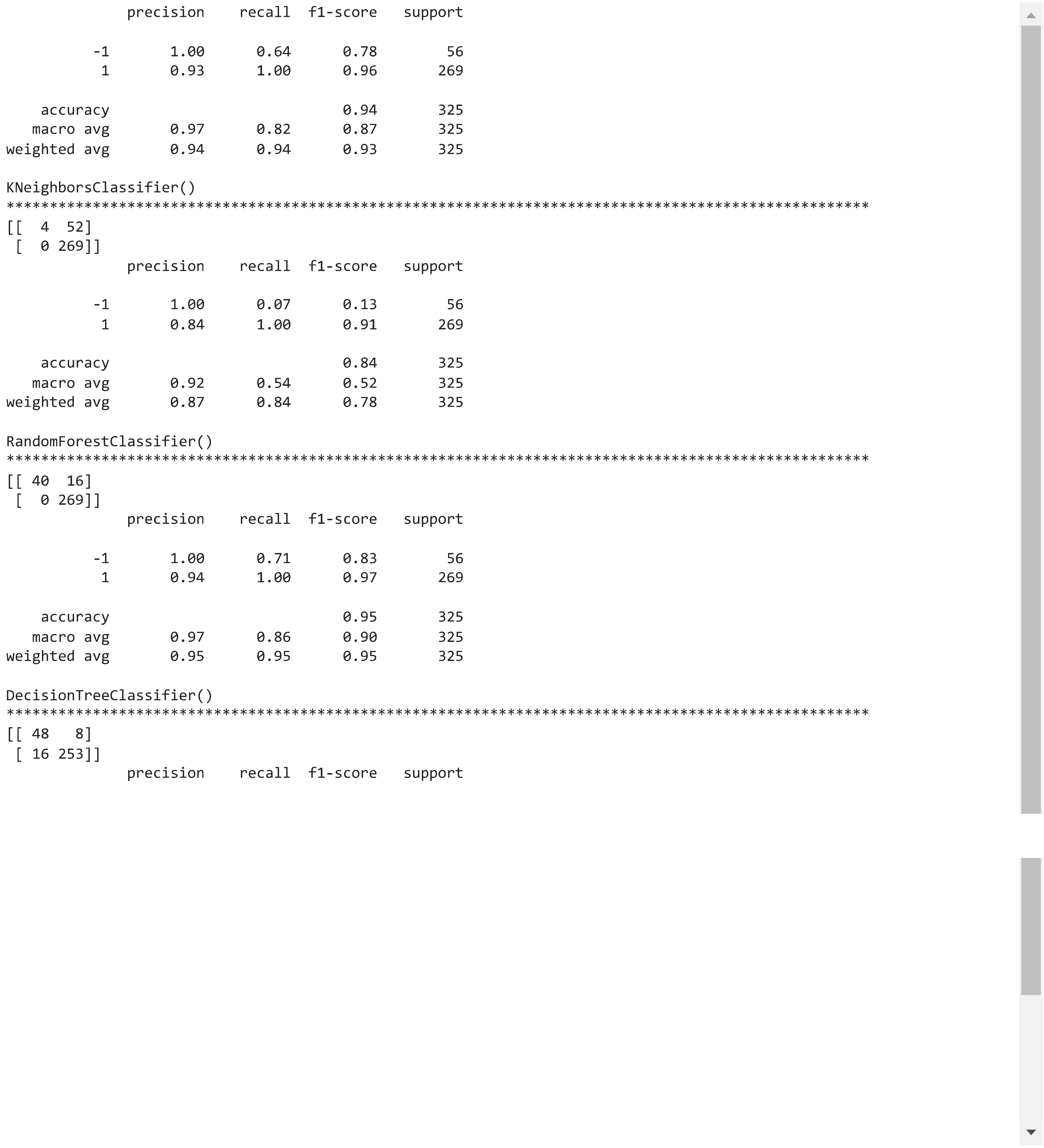
```
for i in lst_model:
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  print(i)
  print('*'*100)
  print(confusion_matrix(y_test,y_pred))
  print(classification_report(y_test,y_pred))
```

```
SVC()
****************************************************************************************************
[[ 33  23]
 [  0 269]]
              precision    recall  f1-score   support

          -1       1.00      0.59      0.74        56
           1       0.92      1.00      0.96       269

    accuracy                           0.93       325
   macro avg       0.96      0.79      0.85       325
weighted avg       0.93      0.93      0.92       325

MultinomialNB()
****************************************************************************************************
[[ 36  20]
 [  0 269]]
```



```
              precision    recall  f1-score   support

          -1       1.00      0.64      0.78        56
           1       0.93      1.00      0.96       269

    accuracy                           0.94       325
   macro avg       0.97      0.82      0.87       325
weighted avg       0.94      0.94      0.93       325

KNeighborsClassifier()
****************************************************************************************************
[[  4  52]
 [  0 269]]
              precision    recall  f1-score   support

          -1       1.00      0.07      0.13        56
           1       0.84      1.00      0.91       269

    accuracy                           0.84       325
   macro avg       0.92      0.54      0.52       325
weighted avg       0.87      0.84      0.78       325

RandomForestClassifier()
****************************************************************************************************
[[ 40  16]
 [  0 269]]
              precision    recall  f1-score   support

          -1       1.00      0.71      0.83        56
           1       0.94      1.00      0.97       269

    accuracy                           0.95       325
   macro avg       0.97      0.86      0.90       325
weighted avg       0.95      0.95      0.95       325

DecisionTreeClassifier()
****************************************************************************************************
[[ 48   8]
 [ 16 253]]
              precision    recall  f1-score   support
```