naive_bayes

November 8, 2022

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import category_encoders as ce
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import RobustScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

[2]:
```
# initializing df
df = pd.read_excel("naive_bayes_algorithm/test-data.xlsx")
print(df.head().to_string())

# Dimensions of df
print(df.shape)

# iloc controls which rows are used.
set_row = df.iloc[0:2]
print(set_row.to_string())
```

```
                Name                 Date Venue Result           Squad
Opponent  SoTA  Saves  Save%   CS  PSxG  Opposition XG   GA
0  kasper schmeichel 2021-08-13 23:00:00  Home  W 1-0  Leicester City
Wolves   3.0    3.0  100.0  1.0   0.3            1.1  0.0
1  kasper schmeichel 2021-08-22 23:00:00  Away  L 1-4  Leicester City
West Ham   7.0    3.0   42.9  0.0   3.4            2.5  4.0
2  kasper schmeichel 2021-08-27 23:00:00  Away  W 2-1  Leicester City
Norwich City   3.0    3.0  100.0  0.0   1.1            1.6  1.0
3  kasper schmeichel 2021-09-10 23:00:00  Home  L 0-1  Leicester City
Manchester City   8.0    7.0   87.5  0.0   1.3            2.9  1.0
4  kasper schmeichel 2021-09-18 23:00:00  Away  L 1-2  Leicester City
Brighton   3.0    2.0   66.7  0.0   1.1            1.4  2.0
(401, 13)
                Name                 Date Venue Result           Squad  Opponent
```

```
SoTA  Saves  Save%   CS  PSxG  Opposition XG   GA
0  kasper schmeichel 2021-08-13 23:00:00  Home  W 1-0  Leicester City    Wolves
3.0    3.0  100.0  1.0   0.3            1.1  0.0
1  kasper schmeichel 2021-08-22 23:00:00  Away  L 1-4  Leicester City  West Ham
7.0    3.0   42.9  0.0   3.4            2.5  4.0
```

[3]:
```
# Getting categorical columns
categorical = [var for var in df.columns if df[var].dtype == 'O']
print('There are {} categorical variables\n'.format(len(categorical)))
print('The categorical variables are :\n\n', categorical)
print(f'\n{df[categorical].isnull().sum()}')

# Getting numerical columns
numerical = [var for var in df.columns if df[var].dtype != 'O']
print('There are {} numerical variables\n'.format(len(numerical)))
print('The numerical variables are :\n\n', numerical)
```

```
There are 5 categorical variables

The categorical variables are :

 ['Name', 'Venue', 'Result', 'Squad', 'Opponent']

Name        1
Venue       1
Result      1
Squad       1
Opponent    1
dtype: int64
There are 8 numerical variables

The numerical variables are :

 ['Date', 'SoTA', 'Saves', 'Save%', 'CS', 'PSxG', 'Opposition XG', 'GA']
```

[4]:
```
# Replacing N/a in save% with 0.0 and dropping date
df = df.fillna(0.0)
df = df.drop(['Date'], axis=1)

# Declare feature vector amd target variable
X = df.drop(['GA'], axis=1)
print(X)
y = df['GA']
print(y)
```

```
                  Name Venue Result           Squad         Opponent  SoTA  \
0    kasper schmeichel  Home  W 1-0  Leicester City           Wolves   3.0
1    kasper schmeichel  Away  L 1-4  Leicester City         West Ham   7.0
```

```
2    kasper schmeichel  Away  W 2-1  Leicester City     Norwich City   3.0
3    kasper schmeichel  Home  L 0-1  Leicester City  Manchester City   8.0
4    kasper schmeichel  Away  L 1-2  Leicester City         Brighton   3.0
..                 …   …    …              …               …   …
396        Hugo Lloris  Away  D 1-1       Tottenham        Liverpool   3.0
397        Hugo Lloris  Home  W 3-0       Tottenham          Arsenal   4.0
398        Hugo Lloris  Home  W 1-0       Tottenham          Burnley   1.0
399        Hugo Lloris  Away  W 5-0       Tottenham     Norwich City   0.0
400                0.0   0.0    0.0             0.0              0.0   0.0

     Saves  Save%   CS  PSxG  Opposition XG
0      3.0  100.0  1.0   0.3           1.1
1      3.0   42.9  0.0   3.4           2.5
2      3.0  100.0  0.0   1.1           1.6
3      7.0   87.5  0.0   1.3           2.9
4      2.0   66.7  0.0   1.1           1.4
..     …    …  …   …           …
396    2.0   66.7  0.0   0.4           1.2
397    4.0  100.0  1.0   0.5           0.4
398    1.0  100.0  1.0   0.0           0.7
399    0.0    0.0  1.0   0.0           0.3
400    0.0    0.0  0.0   0.0           0.0

[401 rows x 11 columns]
0      0.0
1      4.0
2      1.0
3      1.0
4      2.0
      …
396    1.0
397    0.0
398    0.0
399    0.0
400    0.0
Name: GA, Length: 401, dtype: float64
```

[5]:
```python
# Spliting Data into sep training sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
print(X_train)
print(X_test)

# Getting Categorical/numerical columns in training set
print(X_train.dtypes)
categorical = [col for col in X_train.columns if X_train[col].dtypes == 'O']
print(f'Categorical:\n{categorical}')
numerical = [col for col in X_train.columns if X_train[col].dtypes != 'O']
```

```
print(f'\nNumerical:\n{numerical}')
```

```
                Name Venue Result           Squad         Opponent  SoTA  \
302    Jordan Pickford  Away  L 0-1         Everton        Brentford   3.0
334            Allison  Home  D 2-2       Liverpool         Brighton   6.0
249            Jose Sa  Away  L 0-1          Wolves    Newcastle Utd   3.0
18   kasper schmeichel  Home  L 2-3  Leicester City        Tottenham   8.0
248            Jose Sa  Home  W 2-1          Wolves      Aston Villa   4.0
..                 …    …     …              …               …    …
361            Allison  Home  W 3-1       Liverpool           Wolves   5.0
267       Emi Martinez  Home  L 1-2     Aston Villa  Manchester City   7.0
196      Illan Meslier  Away  L 0-7    Leeds United  Manchester City  15.0
378        Hugo Lloris  Away  D 1-1       Tottenham      Southampton   2.0
226            Jose Sa  Away  D 1-1          Wolves     Leeds United   3.0

     Saves  Save%   CS  PSxG  Opposition XG
302    3.0  100.0  0.0   1.1            1.2
334    3.0   66.7  0.0   1.4            1.1
249    3.0  100.0  0.0   1.2            1.6
18     5.0   62.5  0.0   3.7            3.7
248    4.0  100.0  0.0   2.0            2.1
..     …    …  …   …             …
361    4.0   80.0  0.0   1.3            1.1
267    5.0   71.4  0.0   0.9            0.9
196    8.0   53.3  0.0   4.2            3.6
378    1.0   50.0  0.0   0.7            0.4
226    3.0  100.0  0.0   1.0            1.9

[320 rows x 11 columns]
                Name Venue Result            Squad         Opponent  SoTA  \
107       David de Gea  Away  L 0-4   Manchester Utd         Brighton   5.0
128            Ederson  Away  W 2-1  Manchester City          Arsenal   2.0
28   kasper schmeichel  Home  W 2-1   Leicester City  Crystal Palace   2.0
191      Illan Meslier  Away  L 1-2     Leeds United        Tottenham   4.0
39      Aaron Ramsdale  Home  W 3-1          Arsenal        Tottenham   4.0
..                 …    …     …               …               …    …
389        Hugo Lloris  Away  W 2-0        Tottenham         Brighton   0.0
121            Ederson  Away  W 2-1  Manchester City     Aston Villa   3.0
10   kasper schmeichel  Away  D 1-1   Leicester City    Leeds United   5.0
217      Illan Meslier  Away  W 2-1     Leeds United        Brentford   5.0
288       Emi Martinez  Home  L 1-2      Aston Villa        Liverpool   5.0

     Saves  Save%   CS  PSxG  Opposition XG
107    1.0   20.0  0.0   2.7            2.8
128    1.0   50.0  0.0   0.9            0.9
28     1.0   50.0  0.0   0.9            1.3
191    1.0   50.0  0.0   1.8            2.0
```

```
39     2.0   75.0  0.0   1.1            1.0
..     …    …  …   …             …
389    0.0    0.0  1.0   0.0            0.7
121    2.0   66.7  0.0   0.6            0.9
10     4.0   80.0  0.0   1.1            1.2
217    4.0   80.0  0.0   1.2            1.2
288    3.0   60.0  0.0   1.5            1.9

[81 rows x 11 columns]
Name              object
Venue             object
Result            object
Squad             object
Opponent          object
SoTA             float64
Saves            float64
Save%            float64
CS               float64
PSxG             float64
Opposition XG    float64
dtype: object
Categorical:
['Name', 'Venue', 'Result', 'Squad', 'Opponent']

Numerical:
['SoTA', 'Saves', 'Save%', 'CS', 'PSxG', 'Opposition XG']
```

[6]:
```
# encode remaining variables with one-hot encoding
encoder = ce.OneHotEncoder(cols=['Name', 'Venue', 'Result', 'Squad',␣
 ↪'Opponent'])
X_train = encoder.fit_transform(X_train)
X_test = encoder.transform(X_test)
print(X_train.head(2).to_string())
print(X_test.head(2).to_string())


# Feature Scaling
cols = X_train.columns
scaler = RobustScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
X_train = pd.DataFrame(X_train, columns=[cols])
X_test = pd.DataFrame(X_test, columns=[cols])
print(X_train.head(2).to_string())
```

```
     Name_1  Name_2  Name_3  Name_4  Name_5  Name_6  Name_7  Name_8  Name_9
Name_10  Name_11  Name_12  Venue_1  Venue_2  Venue_3  Result_1  Result_2
Result_3  Result_4  Result_5  Result_6  Result_7  Result_8  Result_9  Result_10
```

```
Result_11  Result_12  Result_13  Result_14  Result_15  Result_16  Result_17
Result_18  Result_19  Result_20  Result_21  Result_22  Result_23  Result_24
Result_25  Result_26  Result_27  Result_28  Result_29  Result_30  Result_31
Result_32  Result_33  Result_34  Result_35  Result_36  Result_37  Result_38
Result_39  Result_40  Result_41  Result_42  Result_43  Result_44  Result_45
Result_46  Result_47  Result_48  Squad_1  Squad_2  Squad_3  Squad_4  Squad_5
Squad_6  Squad_7  Squad_8  Squad_9  Squad_10  Squad_11  Squad_12  Opponent_1
Opponent_2  Opponent_3  Opponent_4  Opponent_5  Opponent_6  Opponent_7
Opponent_8  Opponent_9  Opponent_10  Opponent_11  Opponent_12  Opponent_13
Opponent_14  Opponent_15  Opponent_16  Opponent_17  Opponent_18  Opponent_19
Opponent_20  Opponent_21  SoTA  Saves  Save%   CS  PSxG  Opposition XG
302       1       0       0       0       0       0       0       0       0
0        0        0        1        0        0         1         0         0
0         0         0         0         0         0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0        1        0        0
0        0        0        0        0        0         0         0         0
1           0           0           0           0           0           0
0           0            0            0            0            0            0
0            0            0            0            0            0            0
3.0    3.0  100.0  0.0   1.1            1.2
334       0       1       0       0       0       0       0       0       0
0        0        0        0        1        0         0         1         0
0         0         0         0         0         0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0        0        1        0
0        0        0        0        0        0         0         0         0
0           1           0           0           0           0           0
0           0            0            0            0            0            0
0            0            0            0            0            0            0
6.0    3.0   66.7  0.0   1.4            1.1
     Name_1  Name_2  Name_3  Name_4  Name_5  Name_6  Name_7  Name_8  Name_9
Name_10  Name_11  Name_12  Venue_1  Venue_2  Venue_3  Result_1  Result_2
Result_3  Result_4  Result_5  Result_6  Result_7  Result_8  Result_9  Result_10
Result_11  Result_12  Result_13  Result_14  Result_15  Result_16  Result_17
Result_18  Result_19  Result_20  Result_21  Result_22  Result_23  Result_24
Result_25  Result_26  Result_27  Result_28  Result_29  Result_30  Result_31
Result_32  Result_33  Result_34  Result_35  Result_36  Result_37  Result_38
Result_39  Result_40  Result_41  Result_42  Result_43  Result_44  Result_45
Result_46  Result_47  Result_48  Squad_1  Squad_2  Squad_3  Squad_4  Squad_5
Squad_6  Squad_7  Squad_8  Squad_9  Squad_10  Squad_11  Squad_12  Opponent_1
Opponent_2  Opponent_3  Opponent_4  Opponent_5  Opponent_6  Opponent_7
```

```
Opponent_8  Opponent_9  Opponent_10  Opponent_11  Opponent_12  Opponent_13
Opponent_14  Opponent_15  Opponent_16  Opponent_17  Opponent_18  Opponent_19
Opponent_20  Opponent_21  SoTA  Saves  Save%   CS  PSxG  Opposition XG
107       0       0       0       0       0       0       0       0       1
0        0        0        1        0        0         0         0         0
0         0         0         0         0         0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          1
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0        0        0        0
0        0        0        0        0        1         0         0         0
0           1           0           0           0           0           0
0           0           0           0           0           0           0
0            0            0            0            0            0            0
5.0    1.0   20.0  0.0   2.7            2.8
128       0       0       0       0       0       0       0       0       0
0        1        0        1        0        0         0         0         0
1         0         0         0         0         0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0
0          0          0          0          0        0        0        0
0        0        0        0        0        0         0         1         0
0           0           0           0           0           0           0
0           0           1           0           0           0           0
0            0            0            0            0            0            0
2.0    1.0   50.0  0.0   0.9            0.9
  Name_1 Name_2 Name_3 Name_4 Name_5 Name_6 Name_7 Name_8 Name_9 Name_10 Name_11
Name_12 Venue_1 Venue_2 Venue_3 Result_1 Result_2 Result_3 Result_4 Result_5
Result_6 Result_7 Result_8 Result_9 Result_10 Result_11 Result_12 Result_13
Result_14 Result_15 Result_16 Result_17 Result_18 Result_19 Result_20 Result_21
Result_22 Result_23 Result_24 Result_25 Result_26 Result_27 Result_28 Result_29
Result_30 Result_31 Result_32 Result_33 Result_34 Result_35 Result_36 Result_37
Result_38 Result_39 Result_40 Result_41 Result_42 Result_43 Result_44 Result_45
Result_46 Result_47 Result_48 Squad_1 Squad_2 Squad_3 Squad_4 Squad_5 Squad_6
Squad_7 Squad_8 Squad_9 Squad_10 Squad_11 Squad_12 Opponent_1 Opponent_2
Opponent_3 Opponent_4 Opponent_5 Opponent_6 Opponent_7 Opponent_8 Opponent_9
Opponent_10 Opponent_11 Opponent_12 Opponent_13 Opponent_14 Opponent_15
Opponent_16 Opponent_17 Opponent_18 Opponent_19 Opponent_20 Opponent_21 SoTA
Saves  Save%   CS      PSxG Opposition XG
0    1.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0     0.0     0.0
0.0     1.0    -1.0     0.0      1.0      0.0      0.0      0.0      0.0
0.0      0.0      0.0      0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
```

```
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0     1.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0      0.0      0.0      0.0        1.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0  0.0  0.333333  0.500  0.0  0.076923           0.1
1    0.0    1.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0     0.0     0.0
0.0     0.0     0.0     0.0      0.0       1.0      0.0      0.0      0.0
0.0      0.0      0.0      0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0     0.0     1.0     0.0     0.0     0.0     0.0     0.0     0.0
0.0      0.0      0.0      0.0        0.0        1.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0  1.0  0.333333 -0.166  0.0  0.307692           0.0
```

[7]:
```
# Training our df
gnb = GaussianNB()
gnb.fit(X_train, y_train)

# Predicting results
y_pred = gnb.predict(X_test)
print(y_test.head(5))
print(y_pred)
print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, y_pred)))
y_pred_train = gnb.predict(X_train)
print(y_pred_train)
print('Training-set accuracy score: {0:0.4f}'. format(accuracy_score(y_train,␣
 ↪y_pred_train)))
```

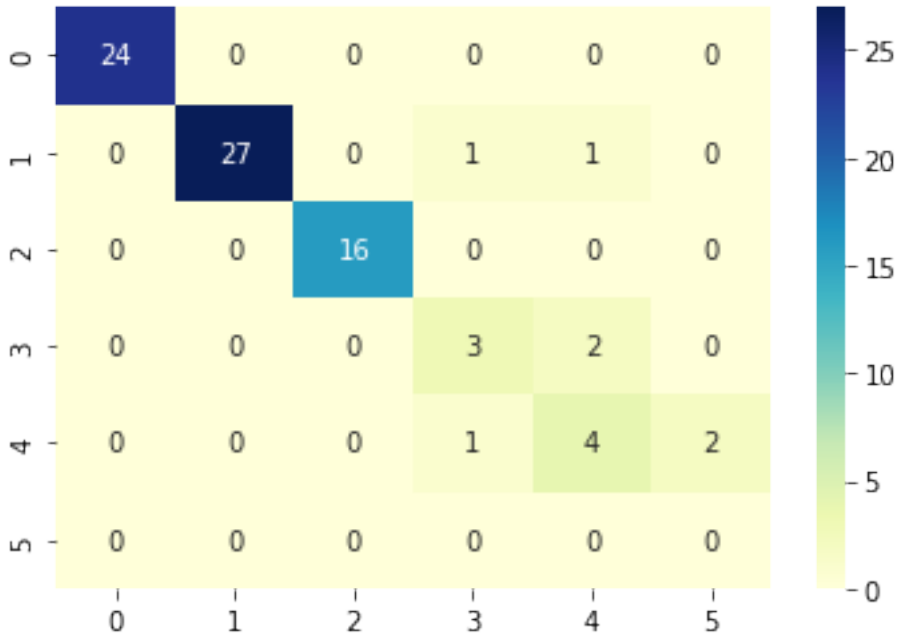
```
107    4.0
128    1.0
28     1.0
191    2.0
39     1.0
Name: GA, dtype: float64
[4. 1. 1. 2. 1. 1. 5. 0. 2. 2. 0. 3. 4. 0. 1. 0. 2. 2. 0. 1. 0. 1. 1. 2.
 0. 0. 4. 2. 2. 0. 1. 1. 0. 5. 0. 0. 1. 0. 2. 0. 3. 1. 0. 3. 0. 0. 2. 1.
 0. 2. 0. 1. 2. 3. 1. 1. 1. 1. 1. 2. 0. 1. 1. 0. 0. 4. 2. 0. 4. 1. 4. 4.
 2. 3. 1. 1. 0. 1. 1. 1. 2.]
Model accuracy score: 0.9136
[1. 2. 1. 3. 1. 1. 3. 3. 1. 1. 0. 3. 0. 1. 1. 0. 5. 0. 1. 1. 1. 3. 0. 4.
 0. 1. 1. 1. 1. 2. 0. 0. 1. 0. 3. 1. 0. 0. 0. 1. 1. 0. 0. 0. 3. 2. 0. 2.
 0. 1. 0. 3. 0. 0. 0. 1. 1. 2. 1. 1. 2. 2. 0. 4. 0. 0. 0. 0. 0. 0. 3. 3.
 4. 1. 0. 1. 0. 0. 6. 0. 2. 1. 1. 1. 2. 3. 0. 1. 3. 4. 2. 0. 1. 6. 1. 1.
```

```
 0. 1. 1. 0. 2. 2. 0. 3. 0. 1. 2. 1. 3. 3. 0. 2. 0. 2. 0. 1. 1. 1. 0. 0.
 1. 0. 3. 0. 2. 0. 1. 1. 2. 1. 1. 1. 1. 2. 1. 0. 1. 1. 4. 1. 0. 2. 2. 0.
 0. 2. 1. 5. 0. 0. 1. 0. 0. 1. 3. 1. 1. 2. 0. 1. 2. 1. 0. 2. 1. 2. 1. 0.
 0. 3. 1. 1. 3. 1. 1. 2. 0. 1. 0. 0. 2. 0. 4. 1. 0. 4. 0. 2. 2. 3. 1. 1.
 0. 0. 3. 1. 2. 1. 1. 2. 2. 0. 0. 5. 1. 2. 0. 0. 2. 1. 3. 0. 1. 2. 0. 0.
 2. 1. 0. 1. 2. 5. 1. 3. 3. 0. 2. 0. 1. 2. 0. 0. 0. 0. 1. 2. 2. 2. 1. 1.
 2. 1. 0. 2. 1. 3. 1. 3. 1. 0. 0. 0. 0. 0. 1. 2. 2. 1. 1. 0. 0. 1. 2. 0.
 1. 0. 1. 2. 3. 0. 0. 0. 1. 0. 2. 1. 5. 1. 0. 0. 1. 2. 0. 3. 2. 1. 1. 2.
 2. 3. 0. 0. 1. 1. 0. 2. 1. 0. 0. 4. 1. 0. 3. 4. 0. 3. 0. 1. 1. 2. 0. 0.
 0. 3. 2. 1. 2. 7. 1. 1.]
Training-set accuracy score: 1.0000
```

[8]:
```
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
print('Confusion matrix\n\n', cm)
cm_matrix = pd.DataFrame(data=cm)
heat = sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='YlGnBu')
plt.show()
print(classification_report(y_test, y_pred))
```

```
Confusion matrix

 [[24  0  0  0  0  0]
 [ 0 27  0  1  1  0]
 [ 0  0 16  0  0  0]
 [ 0  0  0  3  2  0]
 [ 0  0  0  1  4  2]
 [ 0  0  0  0  0  0]]
```



```
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        24
         1.0       1.00      0.93      0.96        29
         2.0       1.00      1.00      1.00        16
         3.0       0.60      0.60      0.60         5
         4.0       0.57      0.57      0.57         7
         5.0       0.00      0.00      0.00         0

    accuracy                           0.91        81
   macro avg       0.70      0.68      0.69        81
weighted avg       0.94      0.91      0.93        81
```

```
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-
packages/sklearn/metrics/_classification.py:1334: UndefinedMetricWarning: Recall
and F-score are ill-defined and being set to 0.0 in labels with no true samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-
packages/sklearn/metrics/_classification.py:1334: UndefinedMetricWarning: Recall
and F-score are ill-defined and being set to 0.0 in labels with no true samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-
packages/sklearn/metrics/_classification.py:1334: UndefinedMetricWarning: Recall
and F-score are ill-defined and being set to 0.0 in labels with no true samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[ ]: