Artificial Intelligence One Year Project

Assignment - 08

Classification Assignment

1.) Identify Your Problem Statement: -

Classification

2.) Tell Basic Info About The Dataset (Total Number Of Rows, Columns)

399 rows, 25 columns

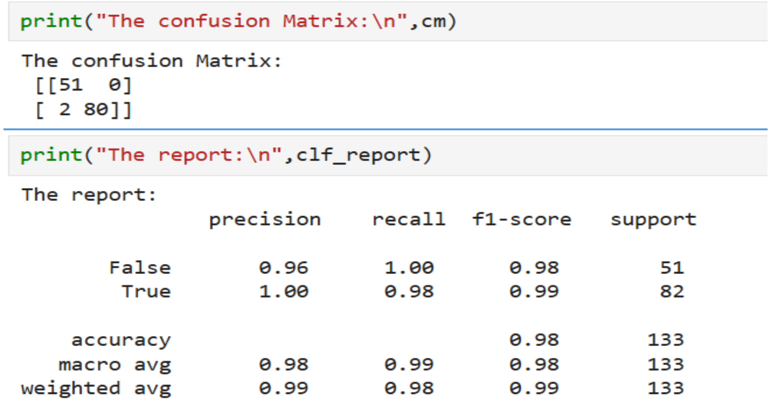
3.) Mention The Pre-Processing Method If You're Doing Any (Like Converting String To Number – Nominal Data)

get dummies (.get_dummies)

4.) Develop A Good Model With Good Evaluation Metric. You Can Use Any Machine Learning Algorithm; You Can Create Many Models. Finally, You Have To Come Up With Final Model.

Finaly, I choose this model RF-Grid-Classification gave the best performance

1. Logistic-Grid-Classification report:



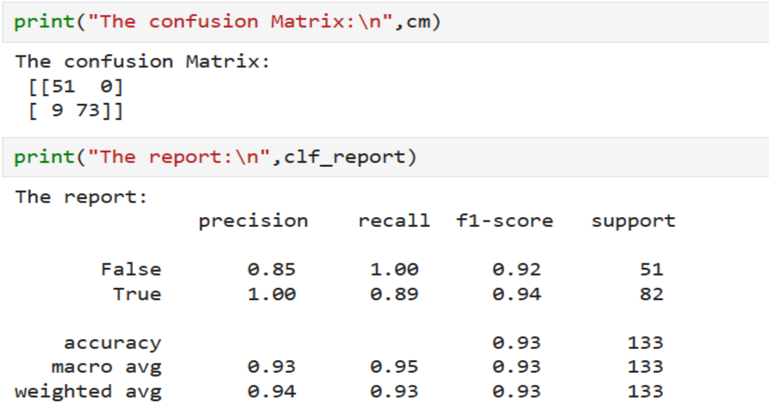
```
print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:
 [[51  0]
 [ 2 80]]
```

```
print("The report:\n",clf_report)
```

```
The report:
               precision    recall  f1-score   support

       False       0.96      1.00      0.98        51
        True       1.00      0.98      0.99        82

    accuracy                           0.98       133
   macro avg       0.98      0.99      0.98       133
weighted avg       0.99      0.98      0.99       133
```

2. DC-Grid report:



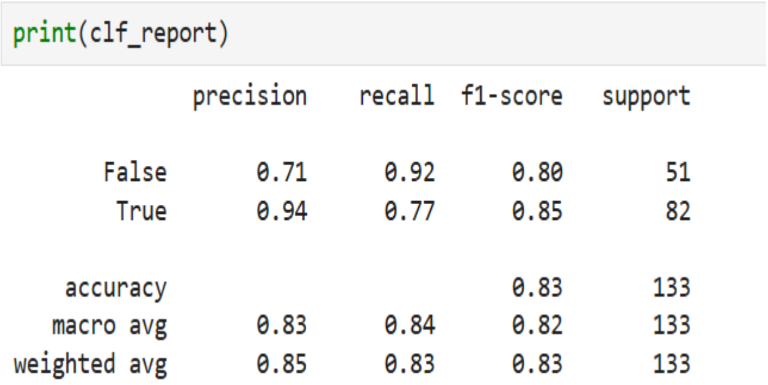
```
print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:
 [[51  0]
 [ 9 73]]
```

```
print("The report:\n",clf_report)
```

```
The report:
               precision    recall  f1-score   support

       False       0.85      1.00      0.92        51
        True       1.00      0.89      0.94        82

    accuracy                           0.93       133
   macro avg       0.93      0.95      0.93       133
weighted avg       0.94      0.93      0.93       133
```

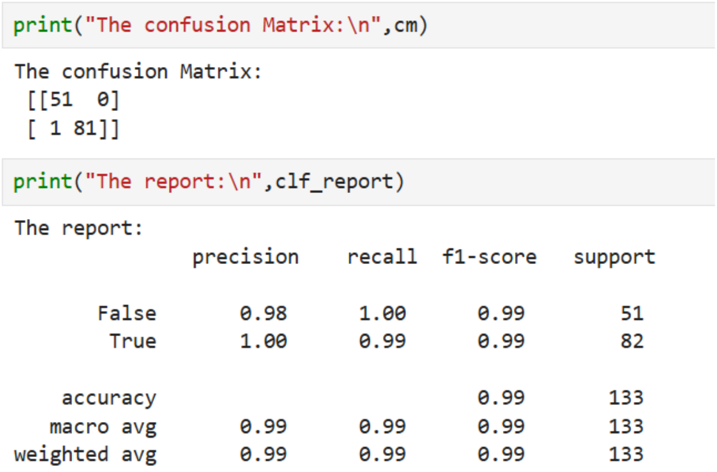
3. KNN report:



```
print(clf_report)
```

```
              precision    recall  f1-score   support

       False       0.71      0.92      0.80        51
        True       0.94      0.77      0.85        82

    accuracy                           0.83       133
   macro avg       0.83      0.84      0.82       133
weighted avg       0.85      0.83      0.83       133
```

4. RF-Grid-Classification report:



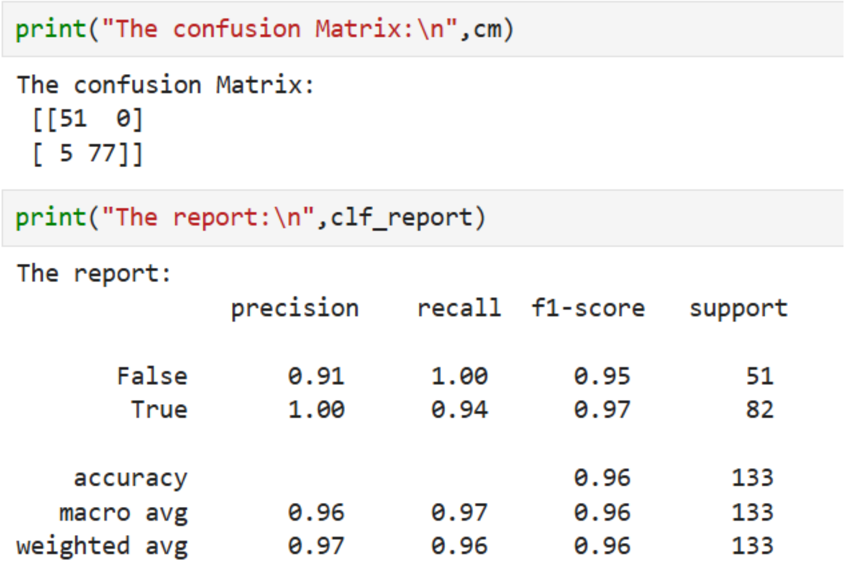
```
print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:
 [[51  0]
 [ 1 81]]
```

```
print("The report:\n",clf_report)
```

```
The report:
               precision    recall  f1-score   support

       False       0.98      1.00      0.99        51
        True       1.00      0.99      0.99        82

    accuracy                           0.99       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

5. SVM-Grid-Classification report:



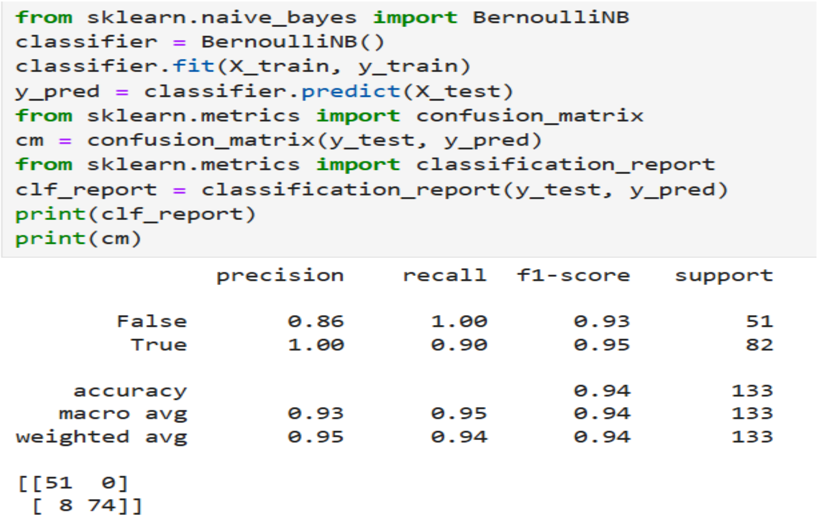
```
print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:
 [[51  0]
 [ 5 77]]
```

```
print("The report:\n",clf_report)
```

```
The report:
               precision    recall  f1-score   support

       False       0.91      1.00      0.95        51
        True       1.00      0.94      0.97        82

    accuracy                           0.96       133
   macro avg       0.96      0.97      0.96       133
weighted avg       0.97      0.96      0.96       133
```

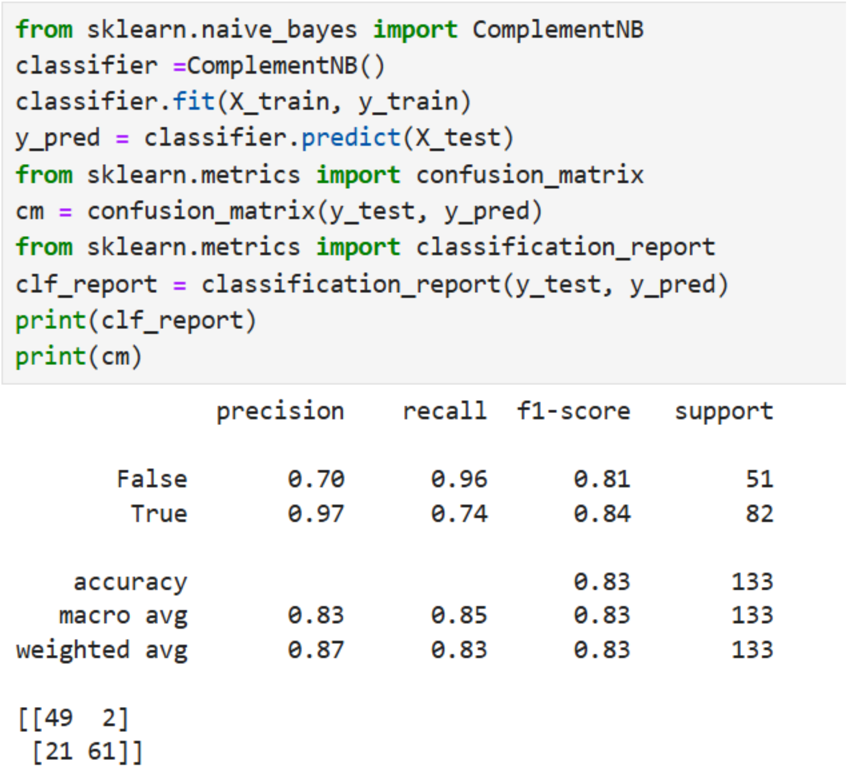
6. **Naive Bayes – BernoulliNB report:**



```
from sklearn.naive_bayes import BernoulliNB
classifier = BernoulliNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

```
              precision    recall  f1-score   support

       False       0.86      1.00      0.93        51
        True       1.00      0.90      0.95        82

    accuracy                           0.94       133
   macro avg       0.93      0.95      0.94       133
weighted avg       0.95      0.94      0.94       133

[[51  0]
 [ 8 74]]
```

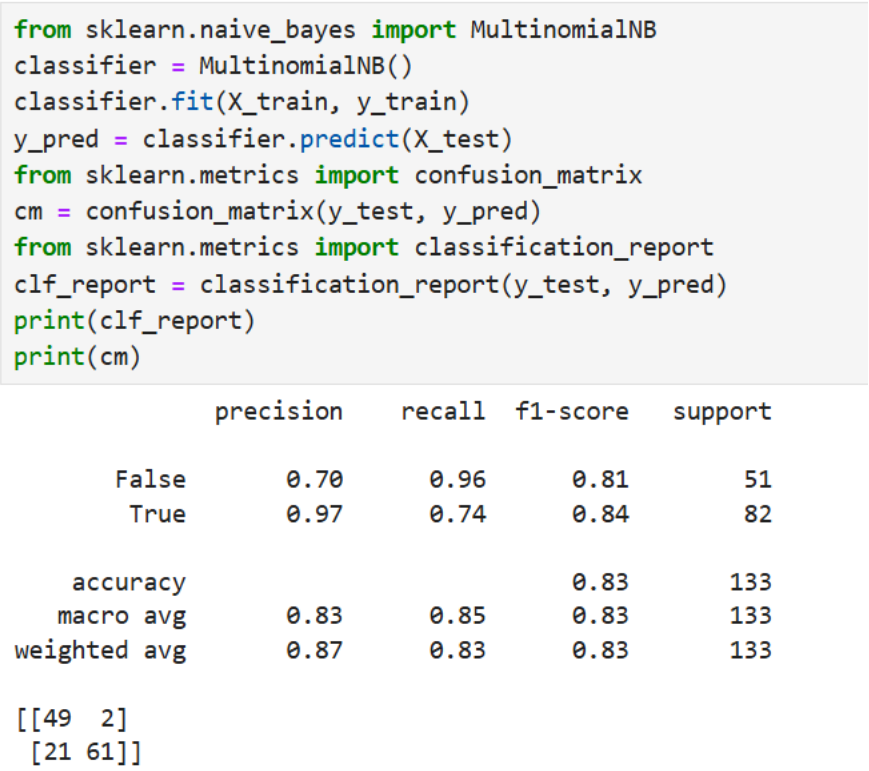
7. **Naive bayes ComplementNB report:**



```
from sklearn.naive_bayes import ComplementNB
classifier =ComplementNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

```
              precision    recall  f1-score   support

       False       0.70      0.96      0.81        51
        True       0.97      0.74      0.84        82

    accuracy                           0.83       133
   macro avg       0.83      0.85      0.83       133
weighted avg       0.87      0.83      0.83       133

[[49  2]
 [21 61]]
```

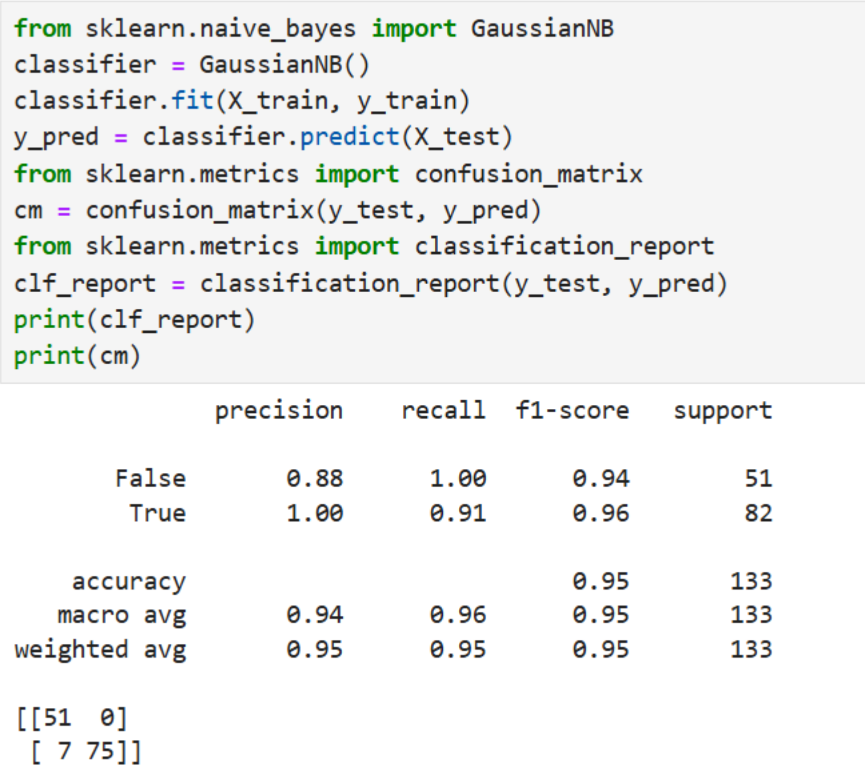
8. Naive bayes MaltinomialNB report:



```
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

```
              precision    recall  f1-score   support

       False       0.70      0.96      0.81        51
        True       0.97      0.74      0.84        82

    accuracy                           0.83       133
   macro avg       0.83      0.85      0.83       133
weighted avg       0.87      0.83      0.83       133

[[49  2]
 [21 61]]
```

9. Naive bayes GaussianNB report:

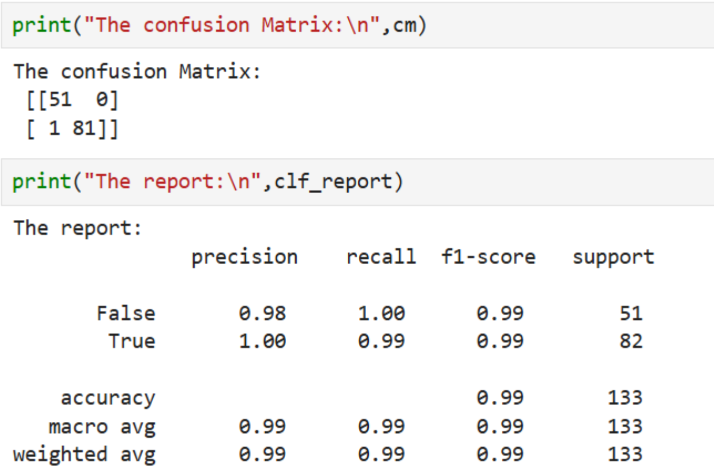


```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import classification_report
clf_report = classification_report(y_test, y_pred)
print(clf_report)
print(cm)
```

```
              precision    recall  f1-score   support

       False       0.88      1.00      0.94        51
        True       1.00      0.91      0.96        82

    accuracy                           0.95       133
   macro avg       0.94      0.96      0.95       133
weighted avg       0.95      0.95      0.95       133

[[51  0]
 [ 7 75]]
```

5.) All The Research Values Of Each Algorithm Should Be Documented. (You Can Make Tabulation Or Screenshot Of The Results.)

6.) Mention Your Final Model, Justify Why U Have Chosen The Same.



```
print("The confusion Matrix:\n",cm)
```

```
The confusion Matrix:
 [[51  0]
 [ 1 81]]
```

```
print("The report:\n",clf_report)
```

```
The report:
               precision    recall  f1-score   support

       False       0.98      1.00      0.99        51
        True       1.00      0.99      0.99        82

    accuracy                           0.99       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

RF-Grid-Classification Gave The Best Performance, F1 Score And Accuracy =0.99; The Model Also Produced Good Results On The Test Data. That’s Why I Selected RF-Grid-Classification As The Final Model.