2020_1125_Decision_Tree_&_Random_Forest_Classification

November 27, 2020

1 Classifying Pulsars from the High Time Resolution Universe Survey (HTRU2) - Decision Tree & Random Forest Classification

1.1 Overview & Citation

In this code notebook, we attempt to classify pulsars from the High Time Resolution Universe Survey, South (HTRU2) dataset using decision tree and random forest classification. The dataset was retrieved from the UC Irvine Machine Learning Repository at the following link: https://archive.ics.uci.edu/ml/datasets/HTRU2#.

The dataset was donated to the UCI Repository by Dr. Robert Lyon of The University of Manchester, United Kingdom. The two papers requested for citation in the description are listed below:

- R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, J. D. Knowles, Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach, Monthly Notices of the Royal Astronomical Society 459 (1), 1104-1123, DOI: 10.1093/mnras/stw656
- R. J. Lyon, HTRU2, DOI: 10.6084/m9.figshare.3080389.v1.

1.2 Import the Relevant Libraries

[1]:
```
# Data Manipulation
import pandas as pd
import numpy as np

# Modeling & Evaluation
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix,classification_report
```

1.3 Import & Check the Data

[2]:
```
df = pd.read_csv('2020_1125_Pulsar_Data.csv')
pulsar_data = df.copy()
```

[3]:
```
pulsar_data.head()
```

```
[3]:       IP_Mean   IP_StdDev  IP_Kurtosis  IP_Skewness   DM_Mean  DM_StdDev  \
     0  140.562500  55.683782    -0.234571    -0.699648  3.199833  19.110426
     1  102.507812  58.882430     0.465318    -0.515088  1.677258  14.860146
     2  103.015625  39.341649     0.323328     1.051164  3.121237  21.744669
     3  136.750000  57.178449    -0.068415    -0.636238  3.642977  20.959280
     4   88.726562  40.672225     0.600866     1.123492  1.178930  11.468720

        DM_Kurtosis  DM_Skewness  Class
     0     7.975532    74.242225      0
     1    10.576487   127.393580      0
     2     7.735822    63.171909      0
     3     6.896499    53.593661      0
     4    14.269573   252.567306      0
```

1.4 Train Test Split

The following train test split will be used for both the decision tree and random forest classifications below:

```
[4]: X = pulsar_data.drop('Class',axis=1)
     y = pulsar_data['Class']
```

```
[5]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
      ↪random_state=42)
```

1.5 Decision Tree Classification

1.5.1 Build and Test the Model

```
[6]: tree = DecisionTreeClassifier()
     tree.fit(X_train,y_train)
```

```
[6]: DecisionTreeClassifier()
```

```
[7]: y_pred = tree.predict(X_test)
```

1.5.2 Model Evaluation

```
[8]: confusion = confusion_matrix(y_test,y_pred)
     print(f'CONFUSION MATRIX:
      ↪\n\n{confusion[0][0]}\t{confusion[0][1]}\n{confusion[1][0]}\t{confusion[1][1]}')
```

```
CONFUSION MATRIX:

4003    67
58      347
```

```
[9]: print(f'CLASSIFICATION REPORT:\n\n{classification_report(y_test,y_pred)}')
```

```
CLASSIFICATION REPORT:

              precision    recall  f1-score   support

           0       0.99      0.98      0.98      4070
           1       0.84      0.86      0.85       405

    accuracy                           0.97      4475
   macro avg       0.91      0.92      0.92      4475
weighted avg       0.97      0.97      0.97      4475
```

The dataset contains a total of 1,639 actual pulsars out of 16,259 instances in the dataset (approximately 10%). This means that we have an unbalanced classification problem, and accuracy is not a good metric. Therefore, the most important metrics for predicting a pulsar with this model are: * Precision = 0.84 * Recall = 0.86 * F1-Score = 0.85

Let's save this data to a .csv file for future comparison with the other classification models:

[10]:
```
with open("2020_1125_Decision_Tree_Results.csv","w") as file:
    file.write('Model,Accuracy,Precision,Recall,F1-Score\n')
    file.write('Decision_Tree,0.97,0.84,0.86,0.85\n')
```

1.6 Random Forest Classification

1.6.1 Build and Test the Model

[11]:
```
forest = RandomForestClassifier(n_estimators=100)
forest.fit(X_train,y_train)
```

[11]:
```
RandomForestClassifier()
```

[12]:
```
y_pred = forest.predict(X_test)
```

1.6.2 Model Evaluation

[13]:
```
confusion = confusion_matrix(y_test,y_pred)
print(f'CONFUSION MATRIX:
 ↪\n\n{confusion[0][0]}\t{confusion[0][1]}\n{confusion[1][0]}\t{confusion[1][1]}')
```

```
CONFUSION MATRIX:

4049    21
69      336
```

[14]:
```
print(f'CLASSIFICATION REPORT:\n\n{classification_report(y_test,y_pred)}')
```

```
CLASSIFICATION REPORT:
```

```
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.83      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.94      4475
weighted avg       0.98      0.98      0.98      4475
```

We see that the random forest classifier has performed significantly better than the decision tree classifier in terms of precision and f1-score. Let's see if we can improve the performance of our random forest by experimenting with the number of estimators in the random forest.

1.6.3 Improving Performance

[15]:
```python
# Test the data with random forest classifiers of variable n_estimators.
# Please note that this may take a few minutes to run.

for i in range(100,1600,100):
    forest_loop = RandomForestClassifier(n_estimators=i)
    forest_loop.fit(X_train,y_train)
    y_pred_loop = forest_loop.predict(X_test)

    print(f'CLASSIFICATION REPORT FOR {i} ESTIMATORS:
 ↪\n\n{classification_report(y_test,y_pred_loop)}\n\n')
```

```
CLASSIFICATION REPORT FOR 100 ESTIMATORS:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.83      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.94      4475
weighted avg       0.98      0.98      0.98      4475



CLASSIFICATION REPORT FOR 200 ESTIMATORS:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.93      0.83      0.88       405

    accuracy                           0.98      4475
```

```
   macro avg       0.96      0.91      0.93      4475
weighted avg       0.98      0.98      0.98      4475


CLASSIFICATION REPORT FOR 300 ESTIMATORS:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.83      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.94      4475
weighted avg       0.98      0.98      0.98      4475


CLASSIFICATION REPORT FOR 400 ESTIMATORS:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.82      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.93      4475
weighted avg       0.98      0.98      0.98      4475


CLASSIFICATION REPORT FOR 500 ESTIMATORS:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.83      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.93      4475
weighted avg       0.98      0.98      0.98      4475


CLASSIFICATION REPORT FOR 600 ESTIMATORS:

              precision    recall  f1-score   support
```

0	0.98	0.99	0.99	4070
1	0.94	0.83	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.91	0.94	4475
weighted avg	0.98	0.98	0.98	4475

CLASSIFICATION REPORT FOR 700 ESTIMATORS:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4070
1	0.94	0.83	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.91	0.94	4475
weighted avg	0.98	0.98	0.98	4475

CLASSIFICATION REPORT FOR 800 ESTIMATORS:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4070
1	0.94	0.83	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.91	0.93	4475
weighted avg	0.98	0.98	0.98	4475

CLASSIFICATION REPORT FOR 900 ESTIMATORS:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4070
1	0.94	0.83	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.91	0.93	4475
weighted avg	0.98	0.98	0.98	4475

CLASSIFICATION REPORT FOR 1000 ESTIMATORS:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4070
1	0.94	0.84	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.92	0.94	4475
weighted avg	0.98	0.98	0.98	4475

CLASSIFICATION REPORT FOR 1100 ESTIMATORS:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4070
1	0.94	0.83	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.91	0.94	4475
weighted avg	0.98	0.98	0.98	4475

CLASSIFICATION REPORT FOR 1200 ESTIMATORS:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4070
1	0.94	0.83	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.91	0.94	4475
weighted avg	0.98	0.98	0.98	4475

CLASSIFICATION REPORT FOR 1300 ESTIMATORS:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4070
1	0.94	0.83	0.88	405
accuracy			0.98	4475
macro avg	0.96	0.91	0.93	4475

```
weighted avg       0.98      0.98      0.98      4475



CLASSIFICATION REPORT FOR 1400 ESTIMATORS:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.83      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.94      4475
weighted avg       0.98      0.98      0.98      4475



CLASSIFICATION REPORT FOR 1500 ESTIMATORS:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      4070
           1       0.94      0.83      0.88       405

    accuracy                           0.98      4475
   macro avg       0.96      0.91      0.93      4475
weighted avg       0.98      0.98      0.98      4475
```

The best random forest model used 1000 estimators and yielded the following metrics: * Accuracy = 0.98 * Precision = 0.94 * Recall = 0.84 * F1-Score = 0.88

Let's save this in a .csv file for future reference:

[16]:
```
with open("2020_1125_Random_Forest_Results.csv","w") as file:
    file.write('Model,Accuracy,Precision,Recall,F1-Score\n')
    file.write('Random_Forest,0.98,0.94,0.84,0.88\n')
```

1.7 Conclusions

We conclude that the random forest classifier performed better than the decision tree classifier. We also conclude that increasing the number of estimators (number of trees in the random forest) does not appreciably improve the predictive power of the random forest, at least when tested over range(100,1600,100).