INTELLIGENT SYSTEMS LAB-9 (25/10/2021)

NAME: Harshvardhan Agarwal | SEC: A | REG NO.: 201800524

PROBLEM STATEMENT

Implement a random forest supervised learning technique.

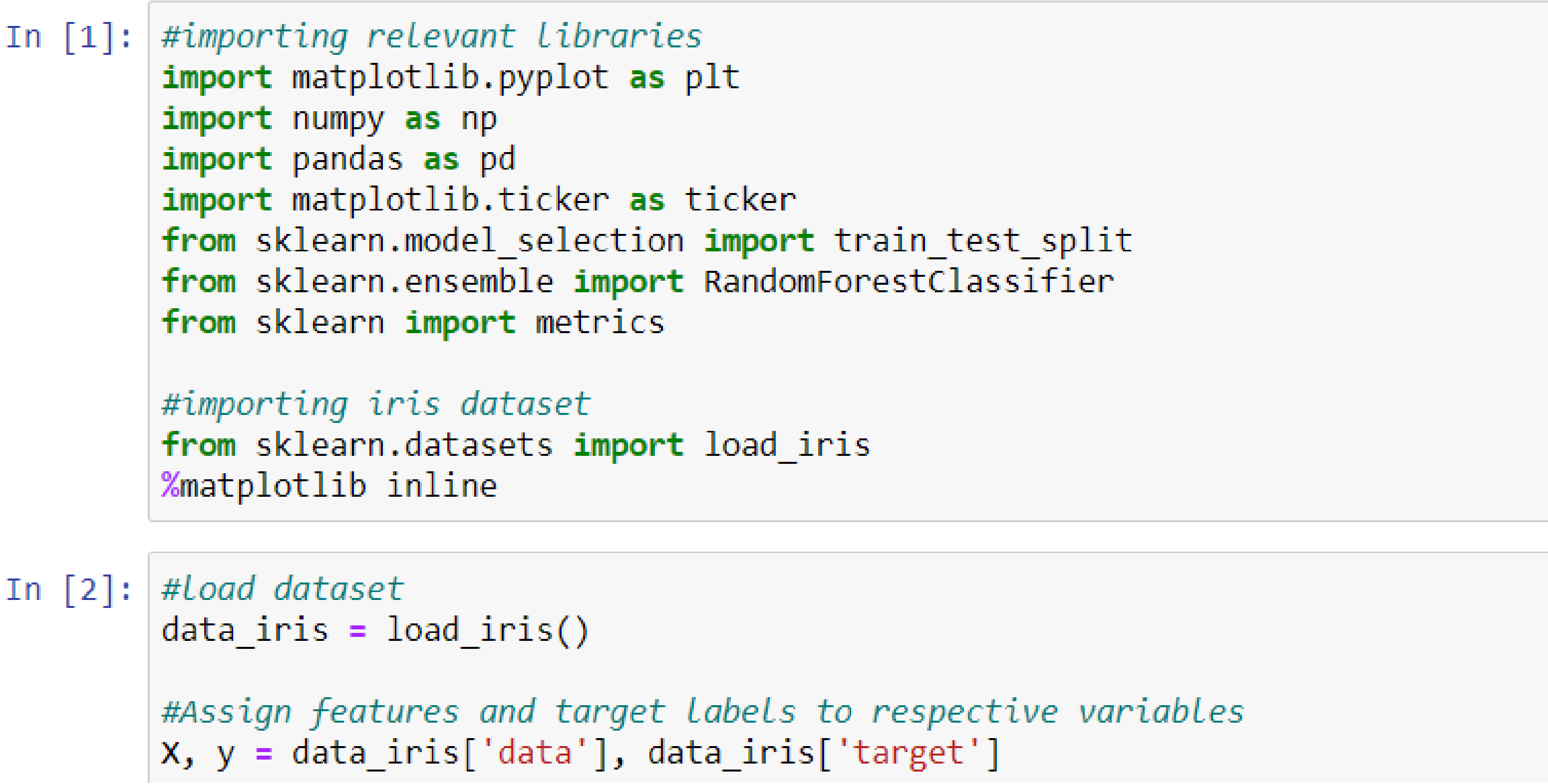
1. Submit pdf file

2. Perform and analyze random forest by varying the test and training set percentage.

3. Analyze random forest by varying the number of decision trees.

PROBLEM SOLUTION

SOURCE CODE AND OUTPUT

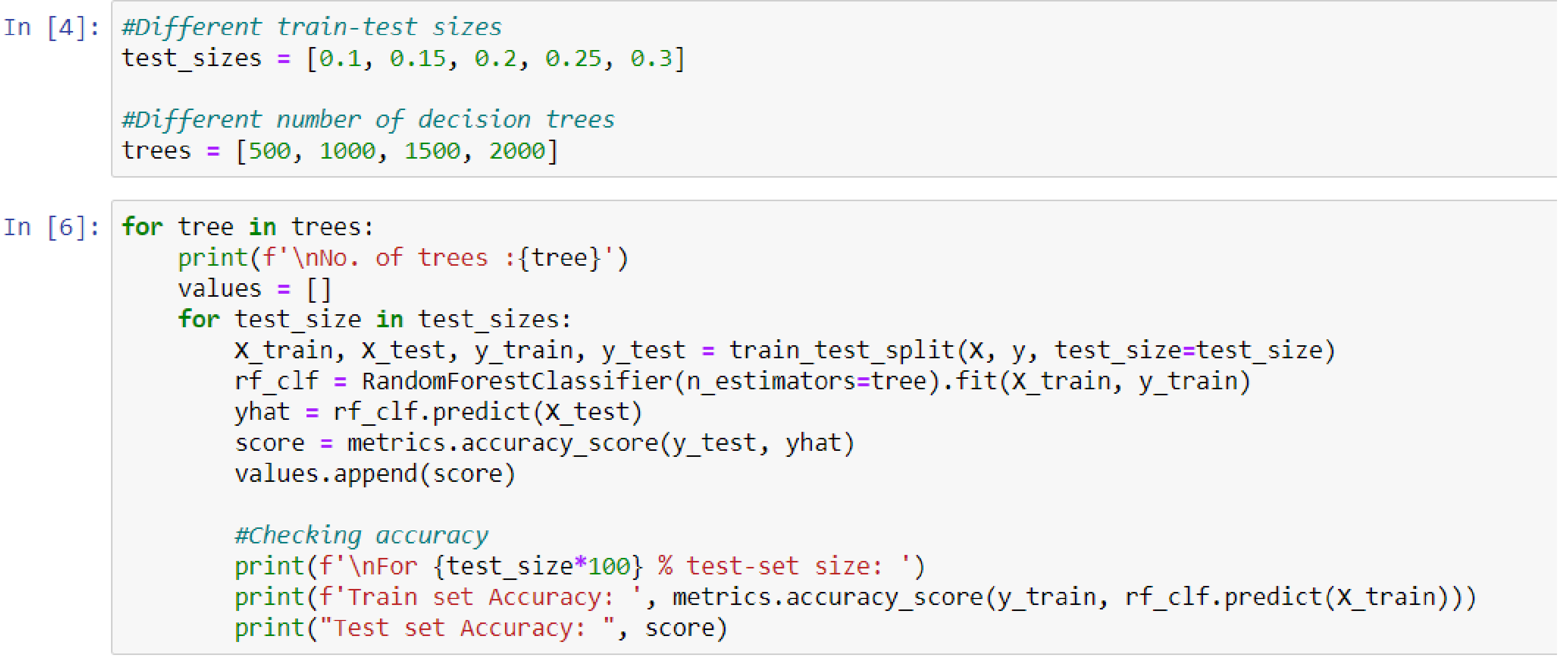


```
In [1]: #importing relevant libraries
        import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd
        import matplotlib.ticker as ticker
        from sklearn.model_selection import train_test_split
        from sklearn.ensemble import RandomForestClassifier
        from sklearn import metrics

        #importing iris dataset
        from sklearn.datasets import load_iris
        %matplotlib inline
```

```
In [2]: #load dataset
        data_iris = load_iris()

        #Assign features and target labels to respective variables
        X, y = data_iris['data'], data_iris['target']
```

Test dataset sizes : 10%, 15%, 20%, 25%, 30%

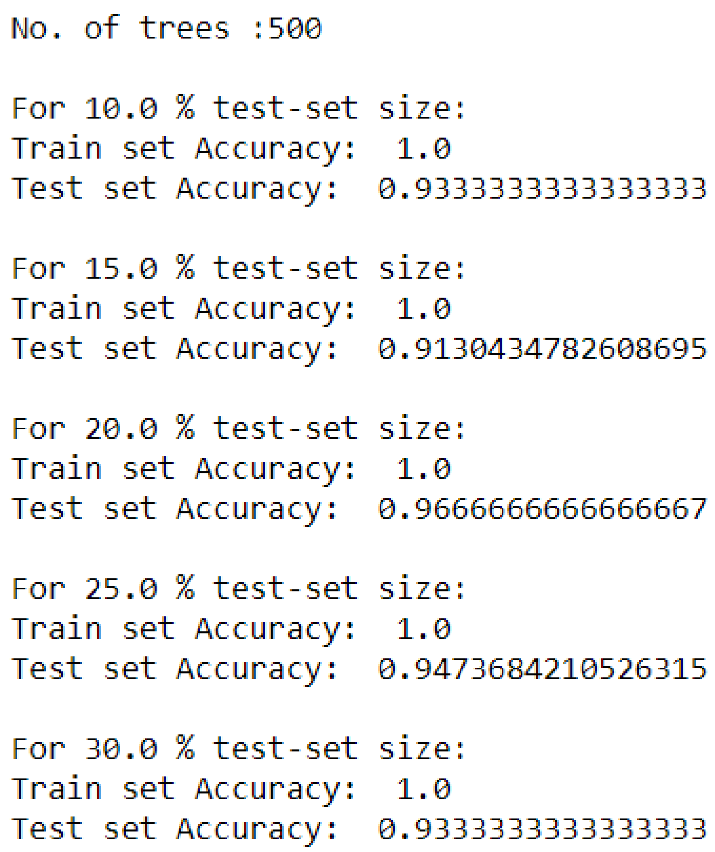
Number of decision trees : 500, 1000, 1500, 2000



```
In [4]: #Different train-test sizes
        test_sizes = [0.1, 0.15, 0.2, 0.25, 0.3]

        #Different number of decision trees
        trees = [500, 1000, 1500, 2000]
```

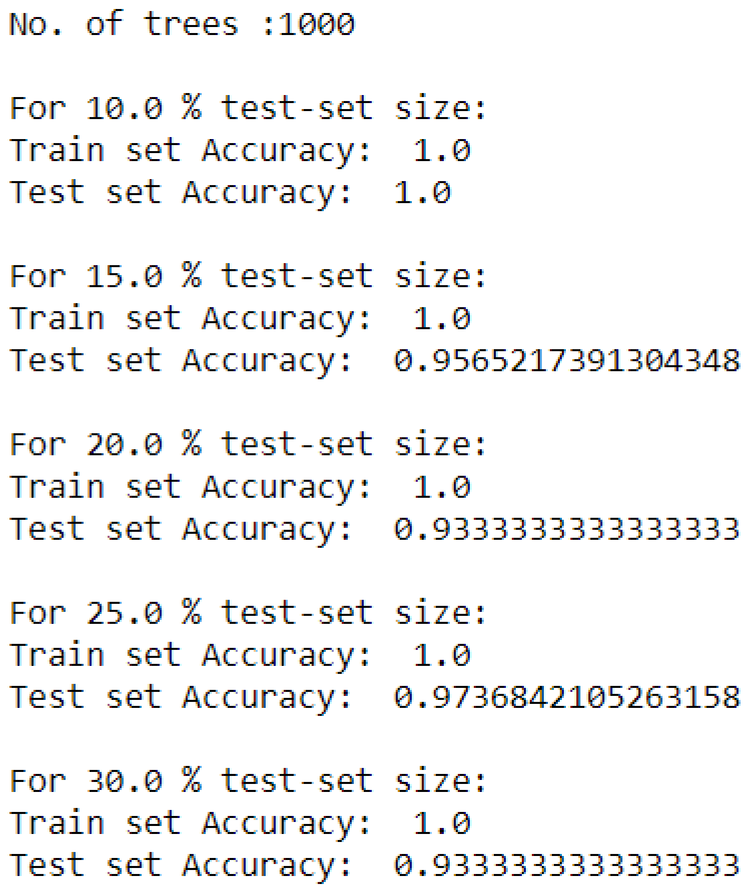
```
In [6]: for tree in trees:
            print(f'\nNo. of trees :{tree}')
            values = []
            for test_size in test_sizes:
                X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size)
                rf_clf = RandomForestClassifier(n_estimators=tree).fit(X_train, y_train)
                yhat = rf_clf.predict(X_test)
                score = metrics.accuracy_score(y_test, yhat)
                values.append(score)

                #Checking accuracy
                print(f'\nFor {test_size*100} % test-set size: ')
                print(f'Train set Accuracy: ', metrics.accuracy_score(y_train, rf_clf.predict(X_train)))
                print("Test set Accuracy: ", score)
```

n_estimators = 500



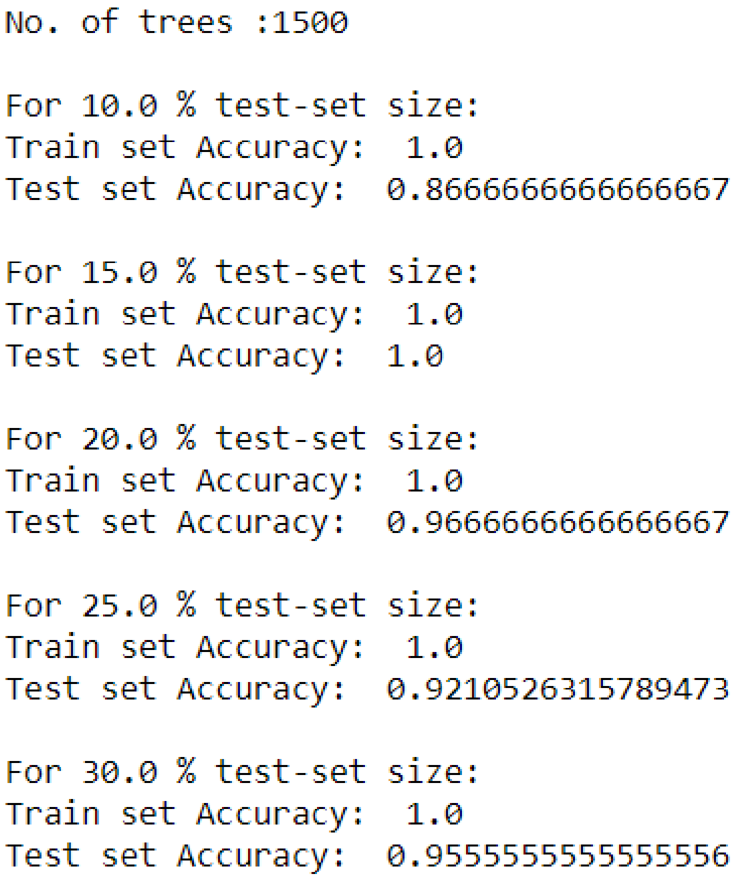
```
No. of trees :500

For 10.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9333333333333333

For 15.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9130434782608695

For 20.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9666666666666667

For 25.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9473684210526315

For 30.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9333333333333333
```

n_estimators = 1000



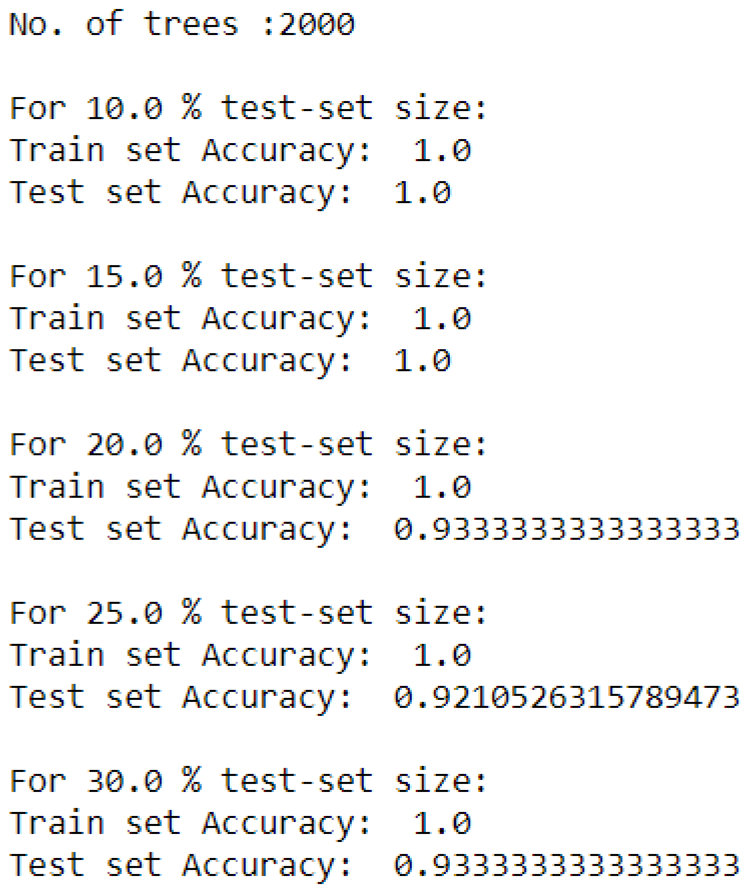
```
No. of trees :1000

For 10.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  1.0

For 15.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9565217391304348

For 20.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9333333333333333

For 25.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9736842105263158

For 30.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9333333333333333
```

n_estimators = 1500



```
No. of trees :1500

For 10.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.8666666666666667

For 15.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  1.0

For 20.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9666666666666667

For 25.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9210526315789473

For 30.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9555555555555556
```

n_estimators = 2000



```
No. of trees :2000

For 10.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  1.0

For 15.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  1.0

For 20.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9333333333333333

For 25.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9210526315789473

For 30.0 % test-set size:
Train set Accuracy:  1.0
Test set Accuracy:  0.9333333333333333
```