**DTCS-620: Statistics for Data Science**

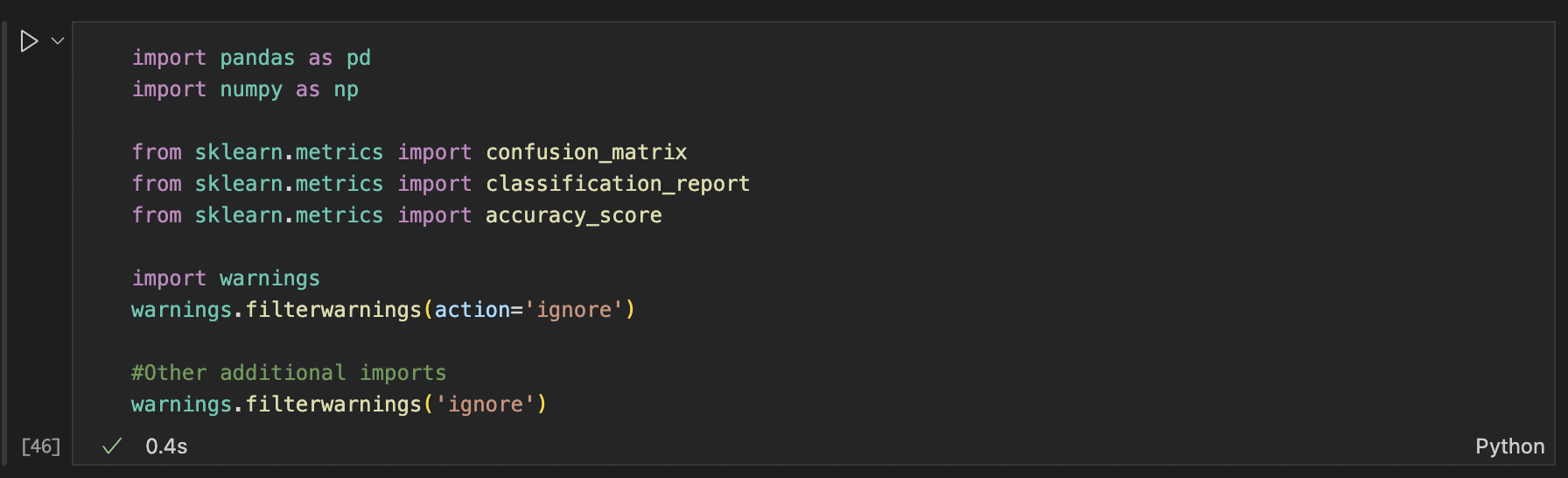
New York Institute of Technology

Project - I

Student Name: Niral Patel

Student ID: 1303276

Code Link: <https://github.com/niral-patel/-Statics-for-DS-Assignment---1/blob/main/Assingment%20-%201.ipynb>

To perform tasks, I import different important classes such as pandas, numpy, sklearn, etc. Moreover, to remove the warnings I imported the warnings class and use action method to ignore the filter warnings.

Graphical user interface, application

Description automatically generatedNow It’s time to upload the data-set and storing it in df and get more information about it by using info() method

Check the total number of Ham data and Spam data by using the pie chart and bar graph.

A screenshot of a computer

Description automatically generated with low confidence

As per requirement, I split the data in to two parts, one is train data set which cover the first 1000 rows of data to train the model and second one is rest below 1000 rows for the testing purpose. Text

Description automatically generated

Reporting Task: 1

Compare the accuracies of the Random Forest classifier as a function of the number of base learners (e.g., 10, 50, 100, 500, 1000, and 5000) and the number of features to consider at each split (e.g., auto or sqrt). Report your observations/conclusions and provide evidence to support your conclusions.

To compare the accuracy, I created the two for loop, first loop contains features which are ‘Auto’ and ‘SQRT’, and second loop contains estimators which are ‘10, 50, 100, 500, 1000, 5000’. In those two loops I perform steps to get accuracy from ‘RandomForestClassifier’ model and store them in a Data Frame.

Text

Description automatically generated

After performing this code, the output will show as below. As per the output the Maximum accuracy is gained when the number of base learners(n\_estimators) is set to 1000 and the ‘max\_features’ parameter is set to ‘sqrt’. The lowest accuracy occurs when ‘n\_estimators’ parameter is set to 10 and ‘max\_features’ is set to ‘auto’. In other words, when n\_estimator is less the accuracy will decrease and the sqrt in max\_features provide the better accuracy than auto. The accuracy result was 0.9336.

Text

Description automatically generated

Create classification report and confusion matrix to get the precision and recall information to compare the two classification (Decision tree and Random Forest).

Chart, treemap chart

Description automatically generated

Graphical user interface, text

Description automatically generatedText

Description automatically generatedTo compute the Decision Tree Classification, import the ‘Decisiontreeclassification’ model form sklearn library and tree class, store the model in variable and train the model and check the accuracy score using this model.

Preform the same functions to get the Classification report and confusion matrix as shown below.

Text

Description automatically generated

Chart, treemap chart

Description automatically generated

Reporting Task 2:

Compare of the results of all the classifiers (with the best possible parameter setting for each classifier). Use classification accuracy (# of instances correctly classified/total # of instances presented for classification), per class classification accuracy, and confusion matrix to compare the classifiers.

As demonstrated in the code, and in the table below, I obtained Random Forest classifier as the best classifier with maximum overall accuracy,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classifier | Classification Accuracy | Per Class Classification Accuracy | | Confusion Matrix |
| Precision | Recall |
| Decision Tree | 0.876 | Ham – 0.89  Spam – 0.85 | Ham – 0.90  Spam – 0.84 | array([[2001, 197],  [ 248, 1152]]) |
| Random Forest | 0.9336 | Ham – 0.93  Spam – 0.94 | Ham – 0.96  Spam – 0.89 | array([[2132, 66],  [ 125, 1275]]) |

A picture containing graphical user interface

Description automatically generated

Decision TreeDiagram

Description automatically generated

Shape

Description automatically generated with medium confidence