**RANDOM FORESTS**

NORTHEASTERN UNIVERSITY - SILICON VALLEY

SUBMITTED TO: SUBMITTED BY:

YIQING WANG ANUBHAV RASTOGI

(PROFESSOR) (STUDENT)

**INTRODUCTION**

Machine Learning is one of the most significant and amazing branches of Artificial Intelligence. It involves techniques which help a machine to learn by itself on the basis of recognized patterns thereby producing the results with less human interaction. In simple words, Machine learning involves the creation of a system which helps a machine to learn and predict automatically without being explicitly programmed. There are many algorithms which are used for making predictions in machine learning. Some of these algorithms are KNN, Naive Bayes, Decision tree, Random Forest etc. The present assignment includes the Random Forest algorithm which is performed on two different datasets.

Random Forest algorithm is a supervised machine learning algorithm widely used for predicting and analyzing the data. It is used for both the regression as well as the classification. It merges the multiple decision trees to create a forest which leads to accurate results. It is extremely simple and highly flexible to use as compared to other machine learning algorithms. Some of its applications include banking and finance sector, health care sector, e-commerce, stock market etc. This algorithm averages several decision trees and uses straight forward hyper-parameters just like the decision tree and also avoids the major problems of machine learning i.e. over-fitting of data. However, in a real-life situation, it overfits easily, therefore proper care is to be taken while fitting the random forest. Another thing which is to be kept in mind is that it makes slow real-life predictions due to the presence of a large number of trees. Therefore, to speed up the process and to improve the model performance, boosting is used.

The present assignment contains two parts: Part A and Part B. Part A contains the glass identification dataset and Part B contains the famous wine dataset. In both the parts, the Random Forest algorithm is applied and predictions are made.

**PART A**

Part A of the assignment is performed in Python via Jupyter notebook. Firstly, the necessary packages are imported and the data file is extracted. Then, the Data head is printed to have a look at the dataset. After seeing the dataset, the features are loaded and the target value is set. Here we have to predict the glass type, so glass\_types column is taken as y variable and all the other columns of the dataset are taken as X variable. After this, the train and test split is performed in the ratio of 4:1. Then, the model is fit and predictions are made. The accuracy of the model 74.42% which is not that great. The confusion matrix is as follows:

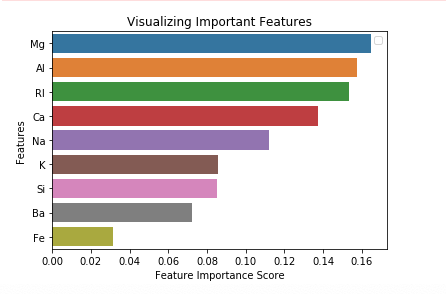
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Predicted Type 1 | Predicted Type 2 | Predicted Type 3 | Predicted Type 5 | Predicted Type 6 | Predicted Type 7 |
| Actual Type 1 | 11 | 1 | 0 | 0 | 0 | 0 |
| Actual Type 2 | 4 | 7 | 0 | 1 | 0 | 0 |
| Actual Type 3 | 3 | 1 | 0 | 0 | 0 | 0 |
| Actual Type 5 | 0 | 0 | 0 | 2 | 0 | 1 |
| Actual Type 6 | 0 | 0 | 0 | 0 | 3 | 0 |
| Actual Type 7 | 0 | 0 | 0 | 0 | 0 | 9 |

The classification report is as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Type 1 | Type 2 | Type 3 | Type 5 | Type 6 | Type 7 |
| Precision | 0.61 | 0.78 | 0.00 | 0.67 | 1.00 | 0.90 |
| Recall | 0.92 | 0.58 | 0.00 | 0.67 | 1.00 | 1.00 |
| F1-score | 0.73 | 0.67 | 0.00 | 0.67 | 1.00 | 0.95 |
| Support | 12 | 12 | 4 | 3 | 3 | 9 |

It can be seen from above that the error rate is 25.58% and the confusion matrix and classification report depict ill-defined precision and recall. Also, it can be seen that Type 3 is not recalled at all. Plus, the report does not tell us anything about Type 4.

Now, the feature importance has been done depicting the significance of all the features. A graph has been plotted to visualize the same. It appears that Aluminum has maximum importance and iron has the least.



Now, X\_train, X\_test, y\_train, and y\_test have been printed to have a look. In order to improve the model, XG Boost has been performed. To avoid the error while performing the XG Boost, the labels must between 0 and class numbers. Here all the labels are starting from 1, so we have done minus 1 from all the y\_train and y\_test so that labels start from 0. As a result, Class 1 became Class 0, Class 2 became Class 1, Class 3 became Class 2, Class 4 became Class 3, Class 5 became Class 4, Class 6 became Class 5 and Class 7 became Class 6.

The XG boost is a gradient boosting machine learning algorithm which is way quicker than the other classifiers. It has a wide number of parameters which can be tuned to produce more accurate and quick results. Now, the XG Boost is performed by setting the parameters and predictions are made. It can be seen that the accuracy of the model has been increased to 81.40%. The classification report and confusion matrix also reflect on the increased accuracy.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Type 0 | Type 1 | Type 2 | Type 4 | Type 5 | Type 6 |
| Precision | 0.77 | 0.86 | 0.00 | 0.67 | 1.00 | 0.90 |
| Recall | 0.83 | 1.00 | 0.00 | 0.67 | 0.67 | 1.00 |
| F1-score | 0.80 | 0.92 | 0.00 | 0.67 | 0.80 | 0.95 |
| Support | 12 | 12 | 4 | 3 | 3 | 9 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Predicted Type 0 | Predicted Type 1 | Predicted Type 2 | Predicted Type 4 | Predicted Type 5 | Predicted Type 6 |
| Actual Type 0 | 10 | 1 | 1 | 0 | 0 | 0 |
| Actual Type 1 | 0 | 12 | 0 | 0 | 0 | 0 |
| Actual Type 2 | 3 | 1 | 0 | 0 | 0 | 0 |
| Actual Type 4 | 0 | 0 | 0 | 2 | 0 | 1 |
| Actual Type 5 | 0 | 0 | 0 | 1 | 2 | 0 |
| Actual Type 6 | 0 | 0 | 0 | 0 | 0 | 9 |

Thus, the XG boost has improved the model as compared to the random forest.

**PART B**

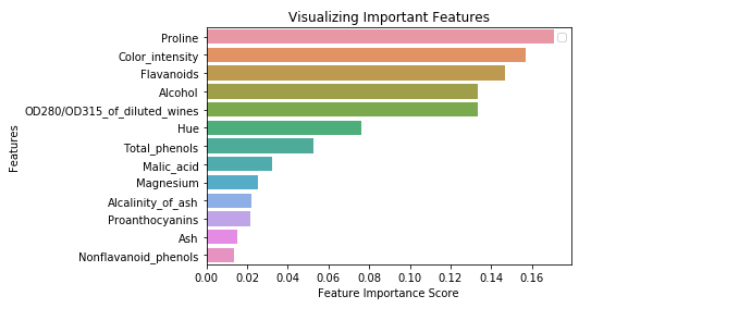
Part B of the assignment involves the performance of random forest on the wine dataset. This part of the assignment is also performed on Jupyter notebook in Python language. The procedure for performing the algorithm is exactly the same as in Part A. Firstly, the relevant packages are imported, and the data set is extracted. Data head is printed to have a look at the data. After this, the features are loaded, and the target value is set. Here, our target value is wine types as we are supposed to predict the types of wine. So, y variable is Wine\_types column and other columns are in X variable.

Now, the splitting of train and test has been done in the ratio of 4:1. After the split is done and the model is fit, predictions are made. It can be seen that the accuracy of the model is pretty high which is a good sign. The error rate is small i.e. 5.56%. Also, the classification report and confusion matrix suggest high accuracy and great model performance. The classification report and confusion matrix are given below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted Type 1 | Predicted Type 2 | Predicted Type 3 |
| Actual Type 1 | 12 | 0 | 0 |
| Actual Type 2 | 1 | 15 | 1 |
| Actual Type 3 | 0 | 0 | 7 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Type 1 | Type 2 | Type 3 |
| Precision | 0.92 | 1.00 | 0.88 |
| Recall | 1.00 | 0.88 | 1.00 |
| F1-score | 0.96 | 0.94 | 0.93 |
| Support | 12 | 17 | 7 |

Now feature importance is done and the graph is plotted. The plotted graph is as follows:



It depicts that Proline has maximum importance and Non-flavonoid Phenols have the least importance. Therefore, we have dropped the column of the least important feature. Also, before training the XG boost, we have made labels between 0 to class numbers by the help of a function. We have subtracted 1 from each class as a result, Class 1 became Class 0, Class 2 became Class 1 and Class 3 became Class 2. After setting the parameters of XG Boost, the predictions have been made. The accuracy of the model remains same i.e. 94.44%. The confusion matrix and classification report are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted Type 0 | Predicted Type 1 | Predicted Type 2 |
| Actual Type 0 | 12 | 0 | 0 |
| Actual Type 1 | 1 | 15 | 1 |
| Actual Type 2 | 0 | 0 | 7 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Type 0 | Type 1 | Type 2 |
| Precision | 0.92 | 1.00 | 0.88 |
| Recall | 1.00 | 0.88 | 1.00 |
| F1-score | 0.96 | 0.94 | 0.93 |
| Support | 12 | 17 | 7 |

Although the Random Forest model shows great accuracy, XG Boost has been done to show if there exist any further chance of improvement. The results, however, are similar depicting both the models have performed equally well. This can be because the dataset is normally distributed.

**CONCLUSION**

This brings an end to the report. On the basis of the analysis, it can be said that Random Forest is a supervised machine learning algorithm which can be used for both regression and classification. Apart from making predictions, this algorithm is used for assessing the importance of a variable. This algorithm involves the creation of multiple trees and then merging them together to form a forest. This results in higher accuracy as compared to the decision tree and also provides randomness to feature space. Random Forest has the same hyper-parameters as that of decision tree but provides more diversity and flexibility. Although, it handles the problem of over-fitting well, but may require cross-validation as the model tends to over-fit in real life situations.

In the present assignment, the random forest is performed on 2 different data sets. The accuracy of the model in Part A is not great, therefore XG Boost is performed to improve the model performance. On the other hand, in Part B, the model accuracy of Random Forest is extremely good such that the same accuracy is given by XG Boost. This implies that Random Forest has worked well on the Wine Dataset in Part B than the Glass Identification Dataset in Part A as the accuracy is higher in Part B than Part A. This also suggests that the Wine Dataset is more normalized leading to higher accuracy of Random Forest Model.

Thus, it can be concluded that Random Forest is a powerful tool for analyzing and predicting the data and can be performed easily with the help of Scikit Learn. This algorithm is very useful for developing a model in a short span of time and also highlights the importance of the features pretty well. Despite several advantages, it has a major disadvantage of slow real-life predictions.

**REFERENCES**

Saimadhu Polamuri. (2017, May 22). How The Random Forest Algorithm Works In Machine Learning. Retrieved from <http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/>

Synced. (2017, October 24). How Random Forest Algorithm Works in Machine Learning. Retrieved from <https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674>

Niklas Donges. (2018, February 22). The Random Forest Algorithm. Retrieved from <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>