## BGD301x – Big Data: Assignment 6

The idea of finding the best model is training the Random Forest Classifier with different parameter sets (*numTrees, maxDepth, maxBins*). Evaluate will based on the following item (using [**pyspark.mllib.evaluation.MulticlassMetrics**](https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.mllib.evaluation.MulticlassMetrics.html)):

* **Accuracy:** The ratio of correctly predicted instances to the total instances. It gives a general measure of how often the classifier is correct.
* **Precision:** The ratio of correctly predicted positive observations to the total predicted positives. Averaged over all classes (weighted precision).
* **Recall:** The ratio of correctly predicted positive observations to all observations in the actual class. Averaged over all classes (weighted recall).
* **F1 Score:** The harmonic means of precision and recall, providing a single metric that balances both concerns. Averaged over all classes (weighted F1 score).
* **Confusion Matrix:** A table layout that allows visualization of the performance of the algorithm, showing the counts of actual vs. predicted classifications. (**Note:** *this criterion is too long, will not be listed in the document*)

The table below shows the parameters sets and the corresponding result when testing on the validation data (30% data of train.csv):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **numTrees** | **maxDepth** | **MaxBins** | **Additional**  **tuner** | **Accuracy/Result** |
| 10 | 4 | 32 |  | * Accuracy: 39.26% * Weighted Precision: 41.26% * Weighted Recall: 39.26% * Weighted F1 Score: 39.98% |
| 40 | 10 | 32 |  | * Accuracy: 42.99% * Weighted Precision: 47.53% * Weighted Recall: 42.99% * Weighted F1 Score: 44.21% |
| 50 | 10 | 40 |  | * Accuracy: 44.59% * Weighted Precision: 49.78% * Weighted Recall: 44.59% * Weighted F1 Score: 45.99% |
| 60 | 10 | 48 |  | * Accuracy: 45.92% * Weighted Precision: 51.16% * Weighted Recall: 45.92% * Weighted F1 Score: 47.14% |
| 80 | 10 | 64 |  | * Accuracy: 46.40% * Weighted Precision: 51.88% * Weighted Recall: 46.45% * Weighted F1 Score: 47.71% |
| 80 | 10 | 64 | Change to use the vector created by word2Vec model only. | * Accuracy: 5.21% * Weighted Precision: 1.0% * Weighted Recall: 5.21% * Weighted F1 Score: 9.91% |

**Explanation**:

* First, test with a random value: the *numTrees = 10, maxDepth = 4 and maxBins = 32* and log the evaluation results.
* Try with *numTrees = 20, maxDepth = 5 and maxBins = 32*. The result is better than the second parameter set.
* Keep increasing the numTrees and maxDepth to see if the result will be improved. But the first 10 predictions are incorrect.
* With *numTrees = 50, maxDepth = 10 and maxBins = 40,* most of predictions are correct. Increase *numTrees* and *maxBins* to see if the predictions will be more accurate.
* With *numTrees = 80, maxDepth = 10 and maxBins = 64,* the accuracy reached 46.45%. Try the same parameter tuner but instead of using the document vector created by 200 words & 100 bigrams most appearance & word2vec model, create vector 400 length by the word2vec model only. The result was bad, only 5.21% accurate.

**Conclusion**: Continue to increase the *maxBins* and more *numTrees*, the percentage of accurate will potentially keep increasing. Although there is room for improvement, the higher value of parameters the more training time and resource will be consumed.

In the scope of this assignment, I will go with the *numTrees = 80, maxDepth = 10 and maxBins = 64,* and run on the test data.

Model can correctly predict the folder for a document about 46.40% of the time may not a good performance. This may be caught by:

* **Data quality**: If some folders have significantly more documents than others, the model might be biased towards predicting the more frequent classes.
* **Inappropriate parameters**: The values for *numTrees*, *maxDepth*, and *maxBins* which were used to train the model may not be a good option.