COM6012 - Assignment 2

Question 1. Supervised classification algorithms to identify Higgs bosons

1. Model Tuning with subset of the Data

a. The Dataframe is created from the HIGGS.csv.gz file downloaded and a subset is created using 1% of the data from the whole dataset as specified in order to find the best configuration of parameters for each classification model.
Then the subset is used to create the train-test set split by 70-30 ratios and there are 77474 rows in the training set, and 33146 rows in the test set. Using the train and test sets data parquets are created and saved to the disk so that same training and test data will be used for all the models.

b. Using pipeline, paramGrid and CrossValidator, the best parameters for each predictive model are extracted to further build the models on the whole dataset. The options for each parameter are chosen according to the range and possible values suggested in the Spark API's.

- In case of Randomforest classifier, a set of low values than the default are chosen for maxBins and subsamplingRate and high values for number of trees are selected as it might require more trees to build at small subsampling rate and for combination of bins.
- Similar options are used for the Gradinet-Boosted Tree classifier as well and for maxIter parameter a combination of low and high value is given to see if the model is optimized in minimizing the loss for a lower or higher number of iterations.
- In case of MultilayerPerceptron or Neural Network classifier a combination of network architecture like number of hidden layers and their sizes (nodes) are used and higher values for stepSize are provided to see if the model would converge better at higher rates for the given number of iterations.

All the various options provided can be seen in the code of 'Q1_Code.py' file and the performance metrics of the best tuned models obtained on the test subset are as shown below.

Classifier Algorithm	Accuracy	AUC
Random Forest	0.709407	0.708318
Gradient-Boosted Tree	0.710312	0.708999
Neural Network (MPC)	0.682013	0.679708

c. As mentioned above the same splits of training and test data that are taken from the parquets are used for the cross validation step and the performance metrics are evaluated for the algorithms.

2. Classification Models on the whole Dataset

a. The best parameters extracted for each model from the data subset are used to build the models again on the whole dataset.

b. To use the same splits of training and test data for the algorithms, data parquets for the training and test set are created from the whole set and stored on the disk. The performance of all three models on the test set for different cores are shown below.

Classifier Algorithm	5 CORES		10 CORES	
	Accuracy	AUC	Accuracy	AUC
Random Forest	0.708191	0.706905	0.708469	0.707137
Gradient-Boosted Tree	0.710837	0.709736	0.710044	0.70882
Neural Network (MPC)	0.684092	0.681968	0.684717	0.68253

c. The training times when using 5 CORES and 10 CORES are shown in the table below.

Classifier Algorithm	Training Times (secs)	
	5 CORES	10 CORES
Random Forest	659.425	631.26
Gradient-Boosted Tree	155.006	147.442
Neural Network (MPC)	1047.19	983.15

3. Relevant features

The three most relevant features for Randomforest and Gradient-Boosted Tree classifier are reported in the table below.

Random Forest	Gradient-Boosted Tree
lepton_pT	lepton_pT
lepton_eta	lepton_eta
lepton_phi	lepton_phi

4. Observations

1. The Accuracy and Area under the curve (AUC) values for both sample and the whole dataset are very close.

2. The training times of the models are lower when using 10 cores when compared to 5 cores.

3. The top three most relevant features for both Random Forest and GBT came out to be similar.

Note: Please note that the warning messages that are being generated in the output file (Q1_Output.txt) while running the cross validation have been removed so as to make the output look clear and reduce the size of the file.

Question 2. Supervised classification algorithms to identify Higgs bosons

1. Pre-processing

a. The dataset has been downloaded from kaggle and a dataframe is created using the train.csv. After replacing the '?' values with Null it is found that Cat2, Cat4-5 and Cat7 features has more than 35% missing values so they are dropped. Then the features Blind_Make, Blind_Model, Blind_Submodel are also dropped as they have more classes or levels considering it would not be allowed for the classifier to work with more than 32 levels (bins).
The features such as Row_ID, Household_ID, Vehicle looks redundant so they are also dropped along with Cat1, Cat6, Cat11, Cat12, OrdCat and NVCat as these categorical features have more classes and with missing data. Finally, the rows with some missing values are dropped as they account for very small percentage of the whole data.
As the final step of the data cleaning process, the correlation between the target feature Claim_Amount and the rest of the numerical features is checked and dropped Var8 feature since it has very small negative correlation and has got the maximum outliers (observed from the previous term assignment).

b. As part of the data preparation both the Model_Year and Calendar_Year features are casted as string or object type and all the categorical features are transformed to numeric values using String Indexer and One-Hot encoder for the models to be build.

c. As the data is highly unbalanced, a resampling method is used to balance out the zero and non-zero claim amounts. Random under-sampling technique is used here and instead of resampling the data to pure balanced set, slightly high proportion of 0's are retained than the non-zero values.

2. Prediction using linear regression.

a. Regression model, linear regression is built on the prepared data to predict the Claim_Amount using the training and test set split with 70-30 ratio and the regression metrics on the test set are reported below.

Classifier Algorithm	MSE	MAE
Linear Regression	75437.52	115.42

b. The training times when using 5 CORES and 10 CORES are as shown below.

Classifier Algorithm	Training Times (secs)	
	5 CORES	10 CORES
Linear Regression	3.401	2.768

3. Prediction using a Tandem Model.

a. For the binary classifier, Gradient-Boosted Tree algorithm is selected and cross validation is performed to fetch the best parameters for the classifier. The model with the hyper parameters is fitted to the same training set from the parquet and an overall accuracy of **61.5%** is obtained. It does not indicate a good fit as the accuracy is low, the model has not performed well in separating the zero and non-zero claims.

b. For the second model, Gamma regressor (a GLM) model is fitted on the subset of the training set with only non-zero claims. The strategy used for the tandem model is to apply both the classifier and the regressor on the whole test set and then replace the final predicted claim amount values with zero if the classifier predicts it to be a zero and the rest will remain the same. The performance of the model is then calculated with the updated predictions and the metrics obtained are as shown below.

Classifier Algorithm	MSE	MAE
GBT + Gamma regressor	90008.05	204.29

c. The training times when using 5 CORES and 10 CORES are obtained as shown below.

Classifier Algorithm	Training Times (secs)	
	5 CORES	10 CORES
Tandem Model (GBT + GLM)	14.86 + 1.93	11.69 + 1.64

4. Observations

1. The standalone Linear Regression model has performed better than the tandem model.

2. The training times for both Liner model and the Tandem model are low when using 10 cores than the 5 cores.

3. The proportion of non-zero claims is very low in the dataset, even after resampling the data the subset is very small when compared to the whole dataset so the subset obtained would have impact on the performance.

References:

1. COM6012 - Labs 2, 6, 7, 8 & 9.
2. COM6509 - Assignment-1.