COM6012 Assignment 2

Ziwu Manuel Worlali - 180128251

April 5, 2019

1 Question 1

To search for exotic particles in high-energy physics using classic supervised learning algorithms, data produced using Monte Carlo simulations is used to train several classification algorithms to identify **Higgs bosons** from particle collisions. A **Decision Tree Classifier**, **Decision Tree Regressor** and **Logistic Regression** model are all developed and trained using the dataset to identify the Higgs bosons.

1.1 Choosing the best configuration of parameters

These different algorithms take various types of options known as hyper-parameters and finding the best hyper-parameter for the algorithm helps increase the performance of the model. To do this, **PySpark** provides a class called *CrossValidator* that allows us to set different options for our model using a *ParamGridBuilder* to set the different options of the model and *Pipeline* to combine multiple algorithms. Using a combination of all these with the different algorithms and 25% of the data, we are able to find the best configuration of parameters for the different algorithms by measuring their performance using different metrics (accuracy, area under the ROC curve). The results are displayed in the table below:

Table 1: Best configuration of parameters

	Accuracy	Area Under ROC curve
Decision Tree Classifier	0.702706	0.701720
Decision Tree Regression	0.702707	0.701722
Logistic Regression	0.627128	0.618893

After running our cross-validator algorithm with different metrics, we obtain the best parameters for the different algorithms.

Decision Tree Classifier: *maxDepth*=10, *maxBins*=16, *impurity*=gini
Decision Tree Regression: *maxDepth*=10, *maxBins*=16, *minInfoGain*=0.0
Logistic Regression: *maxIter*=15, *regParam*=0.05

These were obtained by using a the **BinaryClassificationEvaluator** and **MulticlassClassificationEvaluator** to compute the *areaUnderROC* and *accuracy* respectively after training and testing with a 70% and 30% respectively.

1.2 Comparing classification algorithms

After obtaining, the best parameter configurations, we can apply then in the algorithm using the full dataset to classify the particles. This is done using 10 cores and 20 cores of the system and the training times recorded. We also continue to observe the performance of our models by taking the accuracy and areaUnderROC. The results are show in the table below where *Acc* stands for "Accuracy" and *AUROC* stands for "Area Under ROC".

	Accuracy		Area Under ROC		Training Time			
	10	20	10	20	10 - Acc	20 - Acc	10 - AUROC	20 - AUROC
DTree Classifier	0.704	0.704	0.703	0.702	17m 50s	17m 5s	23m 23s	22m 48s
DTree Regression	0.704	0.7038	0.703	0.703	16m 20s	13m 37s	20m 48s	13m 39s
Log. Regression	0.627	0.627	0.618	0.632	4m 21s	4m 11s	9m 23s	4m 15s

As expected, the algorithm runs faster when run with 20 cores of the system's resources. However, it is also interesting to note that, the Decision Tree Classifer and Decision Tree Regressor obtain very similar performance results. This is very important to note, seeing that the Decision Tree Regressor runs significantly faster than the Classifier. Overall, the Logistic Regression model is the fastest algorithm to run and perform classifications but its performance is lower than the other two models and produces poorer results.

It can be concluded that, the best model to use for our dataset will be the Decision Tree Regressor. A caveat of using this model, would be to carefully choose a threshold for binarization our predicted values. The result output for the 10 cores is stored in a file called: *q1_2_10.txt* and that of the 20 cores in: *q1_2_20.txt*

1.3 Most important features of dataset

The most important features, the model outputs are shown in the table below and we can see that, the different models produce similar results where some features such as m_bb, m_wwwbb appear in all model outputs. The Logistic Regression model however, outputs a feature that isn't common to the other models and the order of importance of its features clearly differs greatly from the other models.

Decision Tree Classifier	**Decision Tree Regression**	**Logistic Regression**
m_bb	m_bb	m_wwbb
m_wwbb	m_wwbb	m_bb
m_wbb	m_wbb	m_lv

2 Question 2

To develop a predictive model that accurately predicts insurance claim payments based on vehicle characteristics. Historic data of previous insurance claims is provided to assist with this task.

2.1 Data preprocessing

The data given to be used to build the model is unfortunately messy. It contains a lot of missing data in several fields that have to be dealt with. In addition to that, majority of the fields in the data are categorical and this could be quite challenging to deal with for a regression problem. Another hurdle with the data provided is that, it is highly unbalanced. The data consists of 99% of labels having claim amount of 0.0. This could pose a problem when training our model and could result in it predicting very high accuracy performances for our model when it actually doesn't perform well.

To over these, several techniques are used to pre-process the data to allow our model to give us high performances and generalize well. The following were used:

- **Overcoming data with missing fields** - Two options could have been chosen whereby a back-fill or forward-fill could have been employed on the data while taking certain attributes such as *Household_ID* or *Vehicle_ID* into account. However, considering the limited amount of system resources and the size of data to be analyzed, removal of the rows with empty fields within the dataset seemed to be the better option.

- **Dealing with categorical data** - Taking into account the use of a linear regression model, a more appropriate option was to change categorical data (usually in the form of string) where converted to integers with similar values given the same identifiers.

- **Handling unbalanced data** - The options considered when dealing with this problem was to oversample minority data by duplication or undersample majority data by removal of some rows. The latter was chosen as it would cause a reduction in the data size that would be used to train our model.

2.2 Predicting claim amount using Regression

A **Linear Regression model** is trained using the preprocessed data and used to predict a claims amount based on input data. We use the **Root-Means Squared Error(RMSE)** as our evaluation metric to check the performance of our model.
The RMSE value of our model is 69.498 after pre-processing our data. This is quite expected as our data may not exactly be linearly separable.

To deal with this, we create a binary classifier that simply tell us if the claim will be a zero or not. This is done using a **Decision Tree Classifier** because most of our data is categorical although our problem is a regression one, the task to be accomplished is a classification problem. An Area Under ROC metric is used to evaluate the classifier which had a value of 0.499997.

The predictions obtained thereof, are passed into a **Gamma Regression** model developed using a **Generalized Linear Model** class in *PySpark* to obtain a predicted claim value on the characteristics of the vehicle. The **Gamma Regression** method is used and attained an RMSE value of 253.784494.

The application was run using a system with 10 cores and one with 20 cores with their training times shown below.

	Training Times	
	10 cores	**20 cores**
Linear Regression	35m 34s	34m 23s
Decision Tree Classifier	10m 37s	10m 22s
Gamma GLM Regression	34m 45s	33m 27s