COM6012 Assignment 2

180128022

March 2019

1 Implementation

A seed value of 50 has been used to split the original data into 70% training and 30% test data.

2 Question 1.1

In this question, pipelines and cross-validation have been used to find the best configuration of parameters for 25% of the data. The table describes it below:

Algorithm	**Param_1**	**Param_2**	**Param_3**	**Accuracy**	**AUC**
DecisionTreeClassfier	maxDepth[10]	maxBins[31]	impurity[entropy]	0.703	0.670
DecisionTreeRegressor	maxDepth[10]	maxBins[35]	NA	0.704	0.776
LogisticRegression	maxIter[15]	regParam[0.1]	NA	0.623	0.666

Below are the best parameters for each of the algorithms:

Decision Tree Classifier	**Decision Tree Regressor**	**Logistic Regression**
cacheNodeIds False		aggregationDepth 2
checkpointInterval 10	cacheNodeIds False	elasticNetParam 0.0
featuresCol features	checkpointInterval 10	family auto
impurity entropy	featuresCol features	featuresCol features
labelCol label	impurity variance	fitIntercept True
maxBins 31	labelCol label	labelCol label
maxDepth 10	maxBins 35	maxIter 15
maxMemoryInMB 256	maxDepth 10	predictionCol prediction
minInfoGain 0.0	maxMemoryInMB 256	probabilityCol probability
minInstancesPerNode 1	minInfoGain 0.0	rawPredictionCol rawPrediction
predictionCol prediction	minInstancesPerNode 1	regParam 0.1
probabilityCol probability	predictionCol prediction	standardization True
rawPredictionCol rawPrediction	seed -1407754390808368278	threshold 0.5
seed 956191873026065186		tol 1e-06

3 Question 1.2

Below table gives the performance comparison through time taken, between 10 and 20 cores.

Algorithm	**Accuracy**	**AUC**	**Cores**	**Time to Train (in seconds)**	**Cores**	**Time to Train (in seconds)**
DecisionTreeClassifier	0.704	0.681	10	88.08	20	77.78
DecisionTreeRegressor	0.704	0.776	10	25.75	20	21.29
LogisticRegression	0.622	0.666	10	15.26	20	13.39

4 Question 1.3

Below table gives three most relevant features for classification and regression for each method obtained:

Algorithm	Feature_1	Feature_2	Feature_3
DecisionTreeClassfier	_c26	_c28	_c27
DecisionTreeRegressor	_c26	_c28	_c27
LogisticRegression	_c28	_c26	_c4

5 Question 2.1.a

For this question, the rows with the missing fields have been removed.

6 Question 2.1.b

As a part of the preprocessing activity, suitable representation of the categorical values has been taken in to consideration. For example, the columns with the 'string' values have been been converted in to a numerical equivalent, which has been done through StringIndexer(). Also, the columns like *Row_ID*, *Household_ID* have been dropped as they did not add any sense to the prediction of *Claim_Amount*. This is a part of the optimisation process. As all the required columns had numerical equivalents, the data type of the columns are converted from String to Double. Thereafter, *Claim_Amount* is taken as 'label' and all the other columns as 'features'.

7 Question 2.1.c

As the data is highly imbalanced, it is imperative that it is dealt with in the correct way by using techniques like correct evaluation matrices, resampling the dataset or even clustering the abundant class. Here, RMSE has been taken in to consideration as it is less benign towards incorrectly classified elements. Now, this can be thought of as a reliable measure as it has been calculated after normalising the data. Due to this a better RMSE value is received.

8 Question 2.2.a

LinearRegression has been used as the predictive model. VectorAssembler() has been used to generate 'label' vectors before normalising them to a more interpretable value. Thereafter, RMSE has been calculated. Below table discusses the result in details for both 10 and 20 cores:

Cores	Time to Train (in seconds)	Cores	Time to Train (in seconds)	RMSE_Train	RMSE_Test
10	43.85	20	29.66	0.0032	0.0033

9 Submission

I have zipped this report and the following files as **acp18rs_180128022_AS2.zip**:
1. Q1_180128022.py that runs all the parts of Q1 one after another.
2. Q1_HPC.sh that contains the script for the above file.
3. Q1_output.txt that contains the above output.
4. Q2_180128022.py that runs all but last part of Q2 one after another.
5. Q2_HPC.sh that contains the script for the above file.
6. Q2_output.txt that contains the above output.