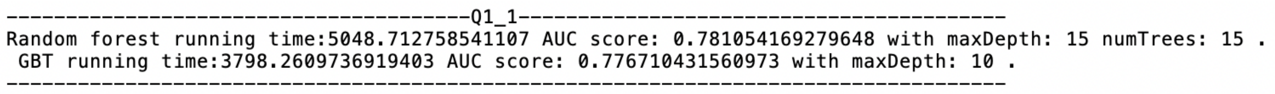
Assignment 2

Question 1:

Firstly, as for random forest, the best hyper-parameters are 15 (maxDepth), 15 (numTrees). As for gradient boosted tree, the best hyper-parameter is 10 (maxDepth). The next figure displays the AUC score based on small dataset.



```
--------------------------------------Q1_1----------------------------------------
Random forest running time:5048.712758541107 AUC score: 0.781054169279648 with maxDepth: 15 numTrees: 15 .
 GBT running time:3798.2609736919403 AUC score: 0.776710431560973 with maxDepth: 10 .
-----------------------------------------------------------------------------------
```

Secondly, training random forest as well as gradient boosted tree based on whole dataset. The AUC score and accuracy based on three different strategies of feature selection will be displayed as follow.

Feature	Model	Accuracy	AUC	training time (10)
21 low-level	RF	0.6479	0.7039	1273
	GBT	0.6473	0.7034	403
7 high-level	RF	0.7085	0.7859	**1014**
	GBT	0.7023	0.7788	**289**
All	RF	**0.7280**	**0.8071**	1517
	GBT	**0.7222**	**0.8009**	549

According to the table above, the performance using all features is better than 7 high-level features that is higher than 21 lower-level features. In contrast, the training time will be decreased with the decreasing of number of features. As a consequence, all features were used to achieve the best AUC and accuracy although it needs longer training time. The next table shows the running time based on 10 cores and 20 cores.

	Model	Training time	Program running time
10 cores	RF	1517	About 45 min
	GBT	549	
20 cores	RF	1198	About 35 min
	GBT	494	

Thirdly, the next table illustrates three most relevant features.

Features	RF	GBT
1	m_bb (0.269)	m_bb (0.156)
2	m_wwbb (0.132)	m_wbb (0.103)

Features	RF	GBT
3	m_wbb (0.129)	m_jlv (0.100)

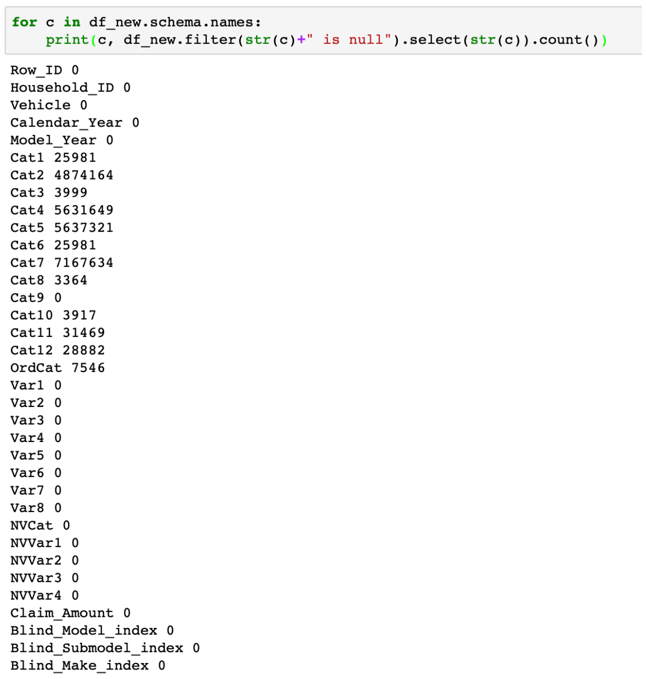
* all those result was from Q1_output.txt.

Observation:

1. High-level features are more important for classification.
2. There is no big improvement of time when using 20 cores compared with 10 cores.
3. RF and GBT have similar performance of accuracy or AUC. However, GBT is less time-consuming.
4. Accuracy metric is lower than AUC.

Question 2:

Firstly, rows consisted of missing value were removed from the dataset. Before, I explain why I choose this way. Let's check the situation of missing value in every columns. The next figure displays the amount of missing value in every columns.



```
for c in df_new.schema.names:
    print(c, df_new.filter(str(c)+" is null").select(str(c)).count())
```

```
Row_ID 0
Household_ID 0
Vehicle 0
Calendar_Year 0
Model_Year 0
Cat1 25981
Cat2 4874164
Cat3 3999
Cat4 5631649
Cat5 5637321
Cat6 25981
Cat7 7167634
Cat8 3364
Cat9 0
Cat10 3917
Cat11 31469
Cat12 28882
OrdCat 7546
Var1 0
Var2 0
Var3 0
Var4 0
Var5 0
Var6 0
Var7 0
Var8 0
NVCat 0
NVVar1 0
NVVar2 0
NVVar3 0
NVVar4 0
Claim_Amount 0
Blind_Model_index 0
Blind_Submodel_index 0
Blind_Make_index 0
```

According to the result above, the amount of missing value is huge in this dataset. In this case, I remove the rows that is consisted of missing value directly, which seems better than replacing them with max category of every columns. To be specific, replacing missing value with its max category in every columns might change the original distribution of dataset. This phenomenon could be more serious if there are huge missing value. Therefore, removing rows consisted of missing value might be a better idea although many useful information might be lost.

Secondly, there are two approach to deal with imbalance dataset. The first is using under sample for balance dataset. The second is to change the weight of imbalanced dataset, which could be implemented by logistic regression ("weightCol" parameter) in Spark. See below:

```
+--------------------+------+------------------+------------+
|            features|labels|           weights|class_labels|
+--------------------+------+------------------+------------+
|[0.16691404566748...|   0.0|0.9924646452339733|         0.0|
|[-0.7435864231954...|   0.0|0.9924646452339733|         0.0|
|[0.16691404566748...|   0.0|0.9924646452339733|         0.0|
|[1.07741451453041...|   0.0|0.9924646452339733|         0.0|
|[0.16691404566748...|   0.0|0.9924646452339733|         0.0|
+--------------------+------+------------------+------------+
```

As for the first approach, Let's compare the MAE and MSE of linear regression model with both balanced and imbalanced dataset.

	Balance	imbalance
MAE	111.413	2.617
MSE	61084	1283

It seems that model with imbalanced dataset gets a higher mark than balanced dataset. The reason might be that unbalanced dataset is not a biggest problem for regression model. Therefore, let's use imbalance dataset for the second and third question. The next table displays the performance of model.

	MAE	MSE	10 cores	20 cores
Linear regression	2.617	1283	259.137	221.441

Thirdly, the next table displays the result of AUC, MAE, MSE of logistic regression and gamma regression.

AUC	MAE	MSE	10 cores	20 cores
0.596	143.382	115732	2222	2076

Observation:

1. Under sample might not be a good idea to balance dataset for regression problem.
2. MSE is much bigger than MAE. It seems that MSE could be impacted by outliers and MAE is more tolerant to outliers.
3. Gamma regression is worse than common linear regression, which might be that gamma is not a prior distribution or small dataset is available for gamma regression.
4. Performance of classification is not good, which might be caused by imbalanced dataset although weighted dataset was used.