Big data parallel programming project report

Jayanthan subbiah

Introduction:

The aim of this project is to get understand with Spark technique and explore more openly on Machine learning libraries(Mlib). In this project I have worked on the teacher suggested project on UCI_Credit_Card data set.

Data set:

The given UCI_Credit_Card data set from Taiwan bank which has 30,000 cases and 25 different attributes.

- ID: ID of each client
- LIMIT_BAL: Given credit
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6: History of past month payment (in reverse order)from September,2005 to April,2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
- BILL_AMT1, BILL_AMT2, BILL_AMT3 BILL_AMT4, BILL_AMT5, BILL_AMT6: Amount of bill statement
- PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6: Amount of previous payment
- Default: Next month payment (1=yes, 0=no)

Task:

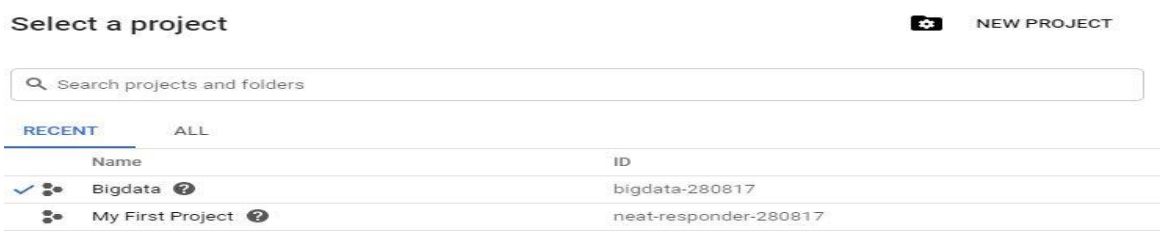
1. The program must be written using Apache spark framework
2. The program must run on any cloud platform, in this project I have used Google cloud platform.

Cloud platform:

Initially I performed all the machine learning task in the jupyter notebook then I created a project in the google cloud platform. I upload the dataset and notebook file in py format into the bucket. Then created a cluster and perform the job.

Step 1: create a project.

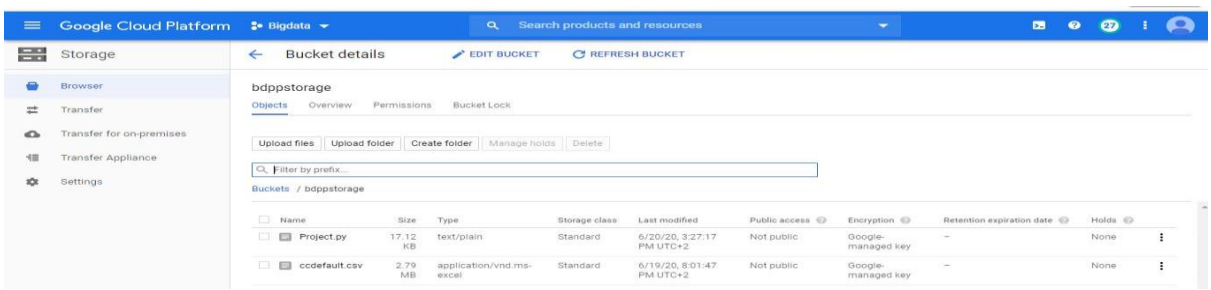
First, I logged in to the google cloud platform, and created a project in the name “Bigdata” with the provided credits to work on GCP.



We need to enable two API namely compute engine and storage to perform our cloud operation.

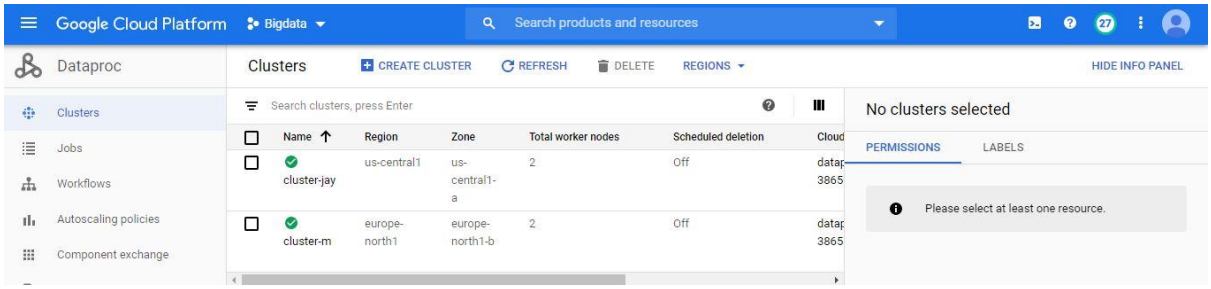
Step 2: Create a bucket:

Created a bucket under storage section to store our dataset and python file to access from the cloud storage.



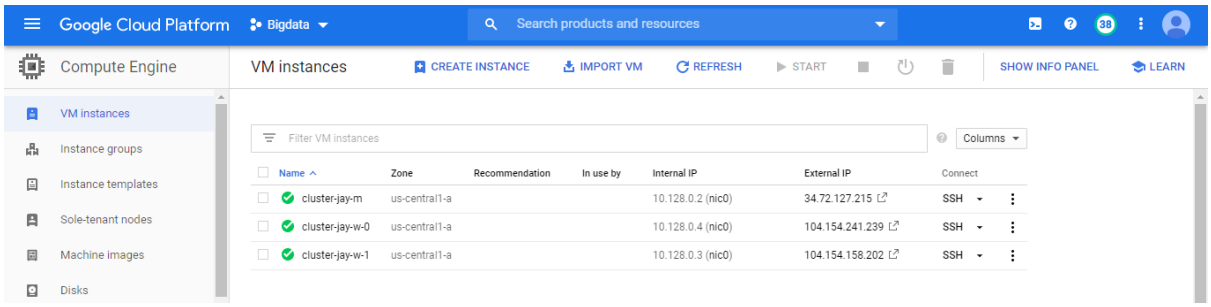
Step 3: Create Cluster:

Next, I have created cluster under Dataproc section where I assigned region as Us central 1,with 1 master and 2 worker nodes and also installed open source component such as Anaconda and Jupyter notebook while creating the cluster.



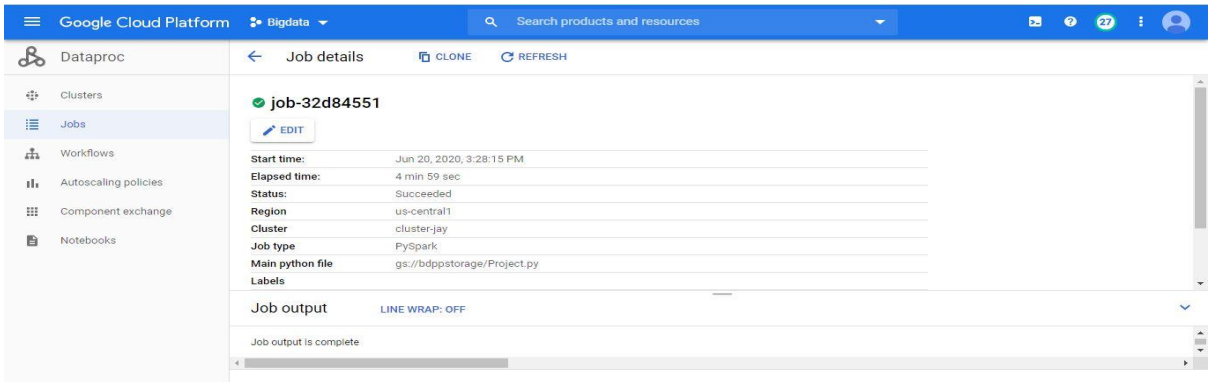
VM Instance

After creating the cluster node, we can see here there are 1 master node and 2 worker nodes in the VM instances.



Step 4 Create Job:

Created job to get the output with job type as pyspark and provided the location of main python file.



The total time taken to complete the job was 4min 59sec.

Load data:

At first, we create a spark session with an interpreter pyspark to create an application for this project. Then we load the dataset to the environment.

```
+---+---------+---+---------+--------+---+-----+-----+-----+-----+-----+-----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
| ID|LIMIT_BAL|SEX|EDUCATION|MARRIAGE|AGE|PAY_0|PAY_2|PAY_3|PAY_4|PAY_5|PAY_6|BILL_AMT1|BILL_AMT2|BILL_AMT3|BILL_AMT4|BILL_AMT5|BILL_AMT6|PAY_AMT1|PAY_AMT2|PAY_AMT3|PAY_AMT4|PAY_AMT5|PAY_AMT6|DEFAULT|
+---+---------+---+---------+--------+---+-----+-----+-----+-----+-----+-----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  1|    20000|  2|        2|       1| 24|    2|    2|   -1|   -1|   -2|   -2|     3913|     3102|      689|        0|        0|        0|        0|      689|        0|        0|        0|        0|      1|
|  2|   120000|  2|        2|       2| 26|   -1|    2|    0|    0|    0|    2|     2682|     1725|     2682|     3272|     3455|     3261|        0|     1000|     1000|     1000|        0|     2000|      1|
|  3|    90000|  2|        2|       2| 34|    0|    0|    0|    0|    0|    0|    29239|    14027|    13559|    14331|    14948|    15549|     1518|     1500|     1000|     1000|     1000|     5000|      0|
|  4|    50000|  2|        2|       1| 37|    0|    0|    0|    0|    0|    0|    46990|    48233|    49291|    28314|    28959|    29547|     2000|     2019|     1200|     1100|     1069|     1000|      0|
|  5|    50000|  1|        2|       1| 57|   -1|    0|   -1|    0|    0|    0|     8617|     5670|    35835|    20940|    19146|    19131|     2000|    36681|    10000|     9000|      689|      679|      0|
+---+---------+---+---------+--------+---+-----+-----+-----+-----+-----+-----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
only showing top 5 rows
```

Data Exploration:

I have printed schema through this it is clear that the data does not have any null value and categorical values

```
root
 |-- ID: integer (nullable = true)
 |-- LIMIT_BAL: integer (nullable = true)
 |-- SEX: integer (nullable = true)
 |-- EDUCATION: integer (nullable = true)
```

```
|-- MARRIAGE: integer (nullable = true)
|-- AGE: integer (nullable = true)
|-- PAY_0: integer (nullable = true)
|-- PAY_2: integer (nullable = true)
|-- PAY_3: integer (nullable = true)
|-- PAY_4: integer (nullable = true)
|-- PAY_5: integer (nullable = true)
|-- PAY_6: integer (nullable = true)
|-- BILL_AMT1: integer (nullable = true)
|-- BILL_AMT2: integer (nullable = true)
|-- BILL_AMT3: integer (nullable = true)
|-- BILL_AMT4: integer (nullable = true)
|-- BILL_AMT5: integer (nullable = true)
|-- BILL_AMT6: integer (nullable = true)
|-- PAY_AMT1: integer (nullable = true)
|-- PAY_AMT2: integer (nullable = true)
|-- PAY_AMT3: integer (nullable = true)
|-- PAY_AMT4: integer (nullable = true)
|-- PAY_AMT5: integer (nullable = true)
|-- PAY_AMT6: integer (nullable = true)
|-- DEFAULT: integer (nullable = true)
```

Then I have viewed default column value count. Here we can see that our given dataset is imbalanced where the classes are not represented equally.

DEFAULT	count
1	6636
0	23364

I have computed crosstab computation between default and different attributes like SEX, EDUCATION, MARRIAGE

EDUCATION_DEFAULT	0	1
0	14	0
1	8549	2036
2	10700	3330
3	3680	1237
4	116	7
5	262	18
6	43	8

MARRIAGE_DEFAULT	0	1
0	49	5
1	10453	3206
2	12623	3341
3	239	84

SEX_DEFAULT	0	1
1	9015	2873
2	14349	3763

Viewed mean of LIMIT_BAL based on MARRIAGE

MARRIAGE	avg(LIMIT_BAL)
1	182200.89318398127
3	98080.49535603715
2	156413.66073665748
0	132962.96296296295

From this we can see that singles in the MARRIAGE column acquired more credit from the bank.

Correlation matrix for SEX and DEFAULT

```
correlation matrix:
DenseMatrix([[ 1.        , -0.03996058],
             [-0.03996058,  1.        ]])
```

Correlation matrix for MARRIAGE and DEFAULT

```
correlation matrix:
DenseMatrix([[ 1.        , -0.02433922],
             [-0.02433922,  1.        ]])
```

Features Scaling:

Feature scaling is one of the most important steps in the machine learning task. Through this step we can get all the attributes to single scale. For this first I have used vector assembler to convert numerical data into a vector then I used standard scaler to scale the dataset.

```
+--------------------+--------------------+
|            features|      scaledFeatures|
+--------------------+--------------------+
|[1.0,20000.0,2.0,...|[-1.7319642067123...|
|[2.0,120000.0,2.0...|[-1.7318487385830...|
|[3.0,90000.0,2.0,...|[-1.7317332704536...|
|[4.0,50000.0,2.0,...|[-1.7316178023242...|
|[5.0,50000.0,1.0,...|[-1.7315023341948...|
+--------------------+--------------------+
only showing top 5 rows
```

Machine learning Models:

In this project I have worked on four classification techniques

- Logistic Regression
- Decision Tree classifier
- Random Forest classifier
- Gradient Boosted Tree classifier

At starting I run the four models for the initial data to know how each model is performing. Before giving the data to train the model I have used random split to split the data for training and testing on 80:20 ratio. From below image we can see how the data is split for train and test based on DEFAULT.

```
Training Data Count: 23861
Test Data Count: 6139
+-----+------------+
|label|count(label)|
+-----+------------+
|    1|        5223|
|    0|       18638|
+-----+------------+

+-----+------------+
|label|count(label)|
+-----+------------+
|    1|        1413|
|    0|        4726|
+-----+------------+
```

1. Logistic Regression
   I have used LogisticRegression to train the model and BinaryClassificationEvaluator as an evaluator and measured the ROC of the model on the test dataset.

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.86966239763636...|
|    0|       0.0|[0.82254479922521...|
|    0|       0.0|[0.74479529855262...|
|    0|       0.0|[0.71997203818794...|
|    1|       0.0|[0.67956090930478...|
+-----+----------+--------------------+
only showing top 5 rows

('Test areaUnderROC for LR model: ', 0.7297743670930633)
('Test accuracy for LR model: ', 0.8035510669490145)
```

2. Decision Tree classifier
   I have used DecisionTreeClassifier to train the model and BinaryClassificationEvaluator as an evaluator and measured the ROC of the model and MulticlassClassificationEvaluator to get accuracy on the test dataset.

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.88404886561954...|
|    0|       0.0|[0.78734984984984...|
|    0|       0.0|[0.78734984984984...|
|    0|       0.0|[0.63099921321793...|
|    1|       0.0|[0.88404886561954...|
+-----+----------+--------------------+
only showing top 5 rows

Test Area Under ROC for DT Model: 0.266955487689
('Test accuracy for DT Model: ', 0.8154422544388337)
+-----+----------+--------------------+
```

3. Random Forest classifier
   I have used RandomForestClassifier to train the model and BinaryClassificationEvaluator as an evaluator and measured the ROC of the model and MulticlassClassificationEvaluator to get accuracy on the test dataset.

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.86152764446328...|
|    0|       0.0|[0.81150409891766...|
|    0|       0.0|[0.77949427186877...|
|    0|       0.0|[0.59147618593988...|
|    1|       0.0|[0.55110795796875...|
+-----+----------+--------------------+
only showing top 5 rows

Test Area Under ROC for RF Model: 0.781711535979
('Test accuracy for RF Model: ', 0.8071347124938915)
```

4. Gradient Boosted Tree classifier
   I have used GBTClassifier to train the model and BinaryClassificationEvaluator as an evaluator and measured the ROC of the model and MulticlassClassificationEvaluator to get accuracy on the test dataset.

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.83793978299773...|
|    0|       0.0|[0.81730204614655...|
|    0|       0.0|[0.77884576143904...|
|    0|       0.0|[0.66967282836787...|
|    1|       1.0|[0.42092005574853...|
+-----+----------+--------------------+
only showing top 5 rows

Test Area Under ROC for GBT model: 0.785227344539
('Test accuracy for GBT model: ', 0.8143020035836456)
```

Data Cleaning:

In the given data set we can see some mislabelled data on some categories.

- In MARRIAGE column there is some mislabelled value 0 are clubbed as 3 “others”. After transforming we will have only 3 distinct classes.
- EDUCATION column has some mislabelled value 5,6,4 is clubbed as 0 “others”. After transforming this we will have only 4 distinct classes.
- Same as PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6 column has -2,-1 are clubbed as 0.

```
Before modifying
+-------------------------+
|count(DISTINCT EDUCATION)|
+-------------------------+
|                        7|
+-------------------------+

After modifying
+-------------------------+
|count(DISTINCT EDUCATION)|
+-------------------------+
|                        4|
+-------------------------+
```

```
Before modifying
+------------------------+
|count(DISTINCT MARRIAGE)|
+------------------------+
|                       4|
+------------------------+

After modifying
+------------------------+
|count(DISTINCT MARRIAGE)|
+------------------------+
|                       3|
+------------------------+
```

Oversampling:

As we know that our dataset is imbalanced, if we perform any classification in it the result will get more accurately. To overcome this, we are duplicating the class 1 of Default column to boost up till it reaches the same level as class 0 of Default column. This is achieved by the sample function of pyspark.

```
+-------+-----+
|DEFAULT|count|
+-------+-----+
|      1|23617|
|      0|23364|
+-------+-----+
```

Model re-training after data cleaning:

After cleaning the data set, I followed the previous feature scaling techniques , train test split data and machine learning model training. This time I dropped the ID column which has a unique id for each person, felt that it affects the performance of the model.

Data distribution for training and testing.

```
Training Data Count: 37486
Test Data Count: 9495
+-----+------------+
|label|count(label)|
+-----+------------+
|    1|       18861|
|    0|       18625|
+-----+------------+

+-----+------------+
|label|count(label)|
+-----+------------+
|    1|        4756|
|    0|        4739|
+-----+------------+
```

1. Logistic Regression

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.58333139055280...|
|    0|       0.0|[0.58561830672272...|
|    0|       1.0|[0.20043934776592...|
|    0|       1.0|[0.39504787453140...|
|    0|       0.0|[0.58201600120321...|
+-----+----------+--------------------+
only showing top 5 rows

('Test areaUnderROC for Lr Model: ', 0.7614287506759438)
('Test accuracy for Lr Model: ', 0.6966824644549763)
('total time to train_model =', 8.566664934158325)
```

2. Decision Tree classifier

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.57407407407407...|
|    0|       0.0|[0.57407407407407...|
|    0|       1.0|[0.38878732480195...|
|    0|       1.0|[0.38878732480195...|
|    0|       0.0|[0.57407407407407...|
+-----+----------+--------------------+
only showing top 5 rows

Test Area Under ROC for Dt Model: 0.529678241196
('Test accuracy for Dt Model: ', 0.6933122696155871)
('total time to train_model =', 7.083477020263672)
```

3. Random Forest classifier

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.51396426110416...|
|    0|       0.0|[0.54734873041800...|
|    0|       1.0|[0.30340420629437...|
|    0|       1.0|[0.33025474246194...|
|    0|       0.0|[0.57406264265161...|
+-----+----------+--------------------+
only showing top 5 rows

Test Area Under ROC for RF Model: 0.775758203097
('Test accuracy for RF Model: ', 0.7142706687730385)
('total time to train_model =', 7.2146031856536865)
```

4. Gradient Boosted Tree classifier

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.50918170821768...|
|    0|       0.0|[0.53667189100446...|
|    0|       1.0|[0.30705348340640...|
|    0|       1.0|[0.27807877848034...|
|    0|       0.0|[0.55557589181100...|
+-----+----------+--------------------+
only showing top 5 rows

Test Area Under ROC for GBT Model: 0.788518930387
('Test accuracy for GBT Model: ', 0.708899420747762)
('total time to train_model =', 11.691107988357544)
```

Model selection:

Model selection is another important task of machine learning model which is also known as model tuning. Here I have used CrossValidator for random forest with k=5 fold the number of trees (1, 2, 4, 5), and Instances per node (1, 2, 4, 5). BinaryClassificationEvaluator as an evaluator and measured the ROC of the model and MulticlassClassificationEvaluator to get accuracy on the test dataset.

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.55787387918792...|
|    0|       0.0|[0.57955140140162...|
|    0|       1.0|[0.21769045338445...|
|    0|       1.0|[0.29401711599966...|
|    0|       0.0|[0.57955140140162...|
+-----+----------+--------------------+
only showing top 5 rows

('Test_roc for RF Model:', 0.7730087524187301)
('Test accuracy for RF Model: ', 0.704265402843602)
('total time to train_model =', 110.32901000976562)
```

For Logistic regression I have used the same CrossValidator with k=5 fold the regParam, [0.1, 0.01]. BinaryClassificationEvaluator as an evaluator and measured the ROC of the model and MulticlassClassificationEvaluator to get accuracy on the test dataset.

```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|    0|       0.0|[0.58164445982028...|
|    0|       0.0|[0.58417912501184...|
|    0|       1.0|[0.20150973215139...|
|    0|       1.0|[0.38431413820671...|
|    0|       0.0|[0.58093667101223...|
+-----+----------+--------------------+
only showing top 5 rows

('Test_roc For LR Model:', 0.7620161851508245)
('Test accuracy For LR Model: ', 0.6983675618746709)
('total time to train_model =', 22.22335696220398)
```

Results:

Models		*Pre training*	*After data cleaning*	*After model selection*
Logistic Regression				
	ROC	0.7297	0.7614	0.7620
	Accuracy	0.8035	0.6966	0.6983
Decision Tree				
	ROC	0.3142	0.6326	
	Accuracy	0.8156	0.6934	
Random Forest				
	ROC	0.7744	0.7696	0.7730
	Accuracy	0.8167	0.7057	0.7025
Gradient Boosted Tree				
	ROC	0.7848	0.7870	
	Accuracy	0.8129	0.7080	

Time taken to train the model:

Models	**On local system** (in sec)	**On GCP** (in sec)
Logistic Regression	8:00	8:56
Decision Tree	6:60	7:08
Random Forest	8:49	7:21
Gradient Boosted Tree	13.78	11:69
Model selection Logistic Regression	26:41	22:22
Model selection Random Forest	140:38	110:32

Conclusion:

By seeing at the result table, we can see there is not high improvements in the results. I do not know much about the banking credit system so I could not be able to interrelate the attributes for future engineering. May be if I would have done the feature engineering that would have improved my results. I conclude by saying that after data cleaning and oversampling there is little improvements in ROC of logistic regression and Decision tree. But Random forest have achieved better results with and without model selection. So, I conclude that imbalance in class will affect the performance of the model.

Reference:

- https://towardsdatascience.com/machine-learning-with-pyspark-and-mllib-solving-a-binary-classification-problem-96396065d2aa
- https://www.guru99.com/pyspark-tutorial.html
- https://tudip.com/blog-post/run-jupyter-notebook-on-google-cloud-platform/