**Home Assignment for Data Scientist**

**[Data Science and Analytics Team Prague]**

Q1. PySpark assessment:

You are given data in the bigdata environment, and you need to train a model in the cluster environment. Imagine you chose PySpark to train the model and make prediction on bigdata. (Please add code snippets and the outputs)

1. Setup a local environment to run PySpark.
2. Download the Census Income data at <http://archive.ics.uci.edu/ml/datasets/Census-Income+(KDD)>
3. Print the data Schema, Summary, # of columns and # of rows .
4. Print a table that distinct values of all columns.
5. Make exploratory data analysis and visualize your findings from data.
6. Create a binary target variable to predict income above 50k and below 50k.
7. Select the most promising features without using any ML model (assume that dataset contains many features, and you cannot train a model on all of them). Hint: you may use some heuristics, significance tests etc. to select the predictive variables. You may not rely on model-based feature reduction techniques like Stepwise selection, Regularization. The goal here is to not rely on modeling algorithms to select features, as this may be very costly.
8. Should the raw model dataset be randomly distributed to Train/Test before or after identifying the most predictive features?
9. Perform any feature engineering and explain the reasons behind. e.g. New features from existing, re-grouping categorical variable categories, meaningful encoding, binning etc.
10. Choose a ML algorithm to build an acceptable predictive model in PySpark and explain the reasons for the selection of such algorithm.
11. Print the model performance measurements such as ROC, AUC, Confusion Matrix for both train and test data.
12. Explain the steps taken to avoid model overfitting.
13. What changes do you propose to make this modeling process more effective, if you are to train this in a cluster with 1,000 nodes, 300 million rows and 5,000 features in AWS?

Q2. SQL skills assessment:

1. You have the following table at disposal (called *campaign\_comms\_user\_level*):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **campaign\_id** | **user\_id** | **date\_sent** | **opened\_ email** | **clicked\_on\_ link** | **visited\_landing\_ page** | **converted** |
| 183946 | 68324 | 20180705 | 1 | 1 | 0 | 0 |
| 183946 | 65715 | 20180712 | 1 | 0 | 0 | 0 |
| 421564 | 46546 | 20180702 | 1 | 1 | 1 | 1 |
| 654654 | 86451 | 20180801 | 0 | 0 | 0 | 0 |
| 421564 | 65158 | 20180807 | 1 | 0 | 0 | 0 |
| … | … | … | … | … | … | … |

Write SQL queries to:

1. Calculate a “funnel” per campaign (= ordered milestones with proportion of successes in each milestone based on the successes in the previous milestone).
2. Find the most successful campaign in terms of   
    i) proportion of opened (among sent) emails   
    ii) proportion of conversions among emails opened.
3. Find all users who were contacted with at least 5 campaigns and   
    i) converted in more than 75% of cases   
    ii) never converted.

1. Regarding the metrics calculated in the previous task, is there any methodological issue / something we need to take care of? How would you assess the quality of the data stored in *campaign\_comms\_user\_level*?

1. Write an SQL script which creates the above table (*campaign\_comms\_user\_level*) using only the information from the table below (called *campaign\_comms*):

|  |  |  |  |
| --- | --- | --- | --- |
| **date** | **campaign\_id** | **user\_id** | **event** |
| 20180722 | 183946 | 657464654 | opened\_email |
| 20180801 | 421564 | 6874654651 | converted |
| … | … | … | … |

Where event can have the following values:

* ‘email\_sent’, ‘email\_opened’, ‘clicked\_on\_link’, ‘visited\_landing\_page’, ‘converted’

Q3: Modelling Assessment:

For this task use the Bank Marketing dataset available on the following address:

<http://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

Build 2 binary classification models using any 2 of the following methods (in R or Python):

1. Logistic Regression
2. Random Forest
3. GBM
4. Xgboost
5. Neural Network

Try and minimalize overfitting. Compare the performance of both models using ROC graphs, AUCs, confusion matrices.

Provide also the full source code and description of any variable transformations or balancing performed.