COM6012 Assignment - Deadline: 13:00 Friday 05 May 2023

Assignment Brief

Please carefully read the assignment brief before starting to complete the assignment.

Release Status:

Q1 - 14 marks:

Q2 - 12 marks: updated at 12:00, 24/03/2023

Q3 - 12 marks: released at 18:30, 29/03/2023

Q4 - 6 marks: released at 18:30, 29/03/2023

Q5 - 6 marks: released at 11:15, 30/03/2023

An FAQ will be updated when questions are raised for important clarifications/tips.

How and what to submit

A. Create a **folder YOUR_USERNAME-COM6012** containing the following:

1) **AS_report.pdf**: A report in **PDF containing answers (*including all figures and tables*) to ALL questions** at the root of the zipped folder (*like readme.txt in the lab solutions*). If an answer to a question is not found in this PDF file, you will lose the respective mark. The report should be concise. You may include appendices/references for additional information but marking will focus on the main body of the report.
2) **Code, script, and output files:** All files used to generate the answers for individual questions in the report above, **except the data,** should be included. These files should be named properly starting with the question number (separate files for the two questions): **for example**, your python code as **Q1_code.py and Q2_code.py**, your HPC script as **Q1_script.sh and Q2_script.sh**, and your output files on HPC as **Q1_output.txt and Q2_output.txt** (and Q1_figB2.jpg, etc.). The results must be generated from the HPC, **not your local machine**. We will apply a penalty if any of these files are missing, 25% for each file. Double check these files are actually included by downloading the zipped file on another machine and open to verify.

B. When you have finished ALL the questions, zip your folder **YOUR_USERNAME-COM6012** to include the above (one single report plus code, script, and output files for all questions, properly named) and upload this **YOUR_USERNAME-COM6012.zip** file to Blackboard before the deadline.

C. **NO DATA UPLOAD**: Please do not upload the data files used. Instead, use the **relative file path in your code**, assuming data files downloaded (and unzipped if needed) under **folder 'Data'**, as in the lab.

D. **Code and output**: 1) Use **PySpark 3.3.1** and Python 3.9.1 as covered in the lecture and lab sessions to complete the tasks; 2) **Submit your PySpark job to HPC** with **qsub** to obtain the output.

Assessment Criteria (Scope: Sessions 1 to 8; Total: **50 marks**)

1. Being able to use PySpark to analyse big data to answer data analytic questions.
2. Being able to perform tasks covered in Sessions 1 to 8 on large-scale data.
3. Being able to make useful observations and explain obtained results clearly.

Late submissions: We follow the Department's guidelines about late submissions, i.e., "If you submit work to be marked after the deadline you will incur a deduction of 5% of the mark each working day the work is late after the deadline, up to a maximum of 5 working days" but **NO late submission will be marked after the maximum of 5 working days** because we will release a solution by then. Please see this link.

Use of unfair means: "*Any form of unfair means is treated as a serious academic offence and action may be taken under the Discipline Regulations.*" (from the MSc Handbook). Please carefully read this link on what constitutes Unfair Means if not sure.

Question 1. Log Mining and Analysis [14 marks, set by Haiping]

You need to finish Lab 1 and Lab 2 before solving this question.

Data: Use **wget** to download the NASA access log July 1995 data (using the hyperlink ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz) to the "Data" folder. The data description is the same as in Lab 2 Task 4 Question 1 so please review it to understand the data before completing the tasks below.

Tasks:

A. Find out the **total** number of requests for 1) all hosts from Germany ending with "**.de**", 2) all hosts from Canada ending with "**.ca**", and 3) all hosts from Singapore ending with "**.sg**". Report these three numbers and visualise them using a graph of your choice. [2 marks]

B. For each of the three countries in Question A (Germany, Canada, and Singapore), find the number of **unique** hosts **and** the top 9 most frequent hosts among them. You need to report three numbers and 3 x 9 = 27 hosts in total. [3 marks]

C. For each country, visualise the percentage (with respect to the total in that country) of requests by each of the top 9 most frequent hosts and the rest (i.e. 10 proportions in total) using a graph of your choice with the 9 hosts clearly labelled on the graph. Three graphs need to be produced. [3 marks].

D. For the most frequent host from each of the three countries, produce a heatmap plot with day as the x-axis (*the range of x-axis should cover the range of days available in the log file. If there are 31 days, it runs from 1st to 31st. If it starts from 5th and ends on 25th, it runs from 5th to 25th*), the hour of visit as the y-axis (0 to 23, as recorded on the server), and the number of visits indicated by the colour. **Three** x-y heatmap plots need to be produced with the day and hour clearly labelled. [3 marks]

E. Discuss two most interesting observations from A to D above, each with three sentences: 1) What is the observation? 2) What are the possible causes of the observation? 3) How useful is this observation to **NASA**? [2 marks]

F. Your report must be clearly written and your code must be well documented so that it is clear what each step is doing. [1 mark]

Question 2 Liability Claim Prediction [set by Shuo - 12 marks].

You need to finish Lab 3 and Lab 4 before solving this question.

Data: The dataset you will use is from Kaggle and can be downloaded from this link. A Kaggle account is needed to download the data. The downloaded file is a .zip file. The uncompressed folder includes one data file: freMTPL2freq.csv, which contains risk features and claim numbers that were collected for 677,991 motor third-party liability policies (observed in a year) [1]. In total there are 12 columns:

- IDpol: The policy ID (used to link with the claims dataset). •
- ClaimNb: Number of claims during the exposure period.

- Exposure: The exposure period.
- Area: The area code.
- VehPower: The power of the car (ordered categorical).
- VehAge: The vehicle age, in years.
- DrivAge: The driver's age, in years (in France, people can drive a car at 18).
- BonusMalus: Bonus/malus, between 50 and 350: <100 means bonus, >100 means malus in France.
- VehBrand: The car brand (unknown categories).
- VehGas: The car gas, Diesel, or regular.
- Density: The density of inhabitants (number of inhabitants per km2) in the city the driver of the car lives in.
- Region: The policy regions in France (based on a standard French classification)

Tasks:

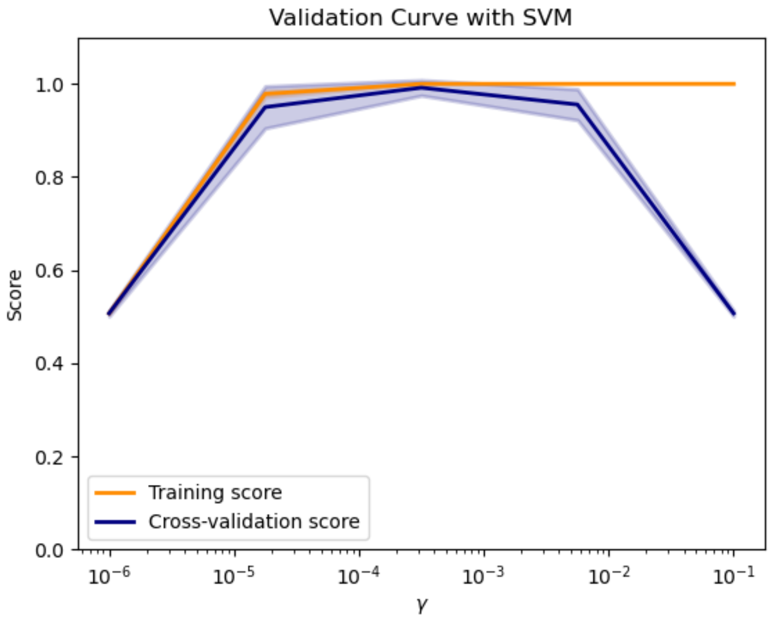
A. Create two new columns: LogClaimNb and NZClaim, where LogClaimNb = Log(ClaimNb), and NZClaim is a binary value for indicating a non-zero number of claims, the value equals 1 if ClaimNb>0, and 0 otherwise. [1 mark]

   Hint: ClaimNb contains zeros, however log(0) is undefined! Therefore, the values in ClaimNb need to be pre-processed somehow to avoid this error before taking the log. By doing this, the Poisson regression will be trained with the pre-processed ClaimNb.

B. Train predictive models with ten features: Exposure, Area, VehPower, VehAge, DrivAge, BonusMalus, VehBrand, VehGas, Density, and Region. Standardise numeric features and use one-hot encoding to transform categorical features.
   a. Split the dataset into training (70%) and test (30%) sets (use the last two digits of your registration number on UCard as the seed to split the dataset). Please use a stratified split according to the number of claims for this imbalanced dataset. [1 mark]

   b. Provide RMSE or accuracy, and model coefficients for each of the predictive models obtained from the following tasks [6 marks]:
      i. Model the number of claims (ClaimNb) conditionally on the input features via Poisson regression. [2 marks]
      ii. Model the relationship between LogClaimNb and the input features via Linear regression, with L1 and L2 regularisation respectively. [2 marks]
      iii. Model the relationship between NZClaim and the input features via Logistic regression, with L1 and L2 regularisation respectively. [2 marks]

   c. Determine the values of regParam (in [0.001, 0.01, 0.1, 1, 10]) for the above tasks automatically using a small subset of the training set (e.g. 10%). Plot the validation curves to files for the five models (one figure per model) with respect to the values of regParam. See the example below of a validation curve figure for a Support Vector Machine in Scikit-learn, where the X-axis values are the value of the hyper-parameter Gamma. [1 mark]



Hint: The train/validation scores can be obtained with PySpark first and then plot to files using visualisation tools such as matplotlib. (See Q3, A3 and Q4, A4 in the FAQs)

C. Compare the performance and coefficients obtained in Q2.B and discuss at least three observations (e.g., anything interesting), with two to three sentences for each observation. If you need to, you can run additional experiments that help you to provide these observations. [3 marks]

[1] A. Noll, R. Salzmann and M.V. Wuthrich, Case Study: French Motor Third-Party Liability Claims (November 8, 2018). doi:10.2139/ssrn.3164764

Question 3. Movie Recommendation and Cluster Analysis [set by Haiping - 12 marks]

You need to finish Lab 5 and Lab 6 before solving this question.

Data: Use **wget** to download the MovieLens 20M Dataset to the "Data" folder and unzip there. Please read the dataset description to understand the data before completing the following tasks.

Tasks:

A. Time-split Recommendation

1) Perform time-split recommendation using ALS-based matrix factorisation in PySpark on the rating data **ratings.csv**: [2 marks]

- **sort** all data by the timestamp,
- perform **splitting according to the sorted timestamp**. **Earlier time (the past) should be used for training and later time (the future) should be used for testing**, which is a more realistic setting than random split. Consider three such splits with three training data sizes: 40%, 60%, and 80%.

2) For each of the three splits above, study two versions (*settings*) of ALS using your student number (keeping only the digits) as the seed as the following [1 marks]

- **Setting 1**: The ALS setting used in Lab 5 except the seed
- **Setting 2**: Based on results (see the next step 3 below) from the first ALS setting, choose another **different ALS setting that can potentially improve the results.** Provide at least a **one-sentence justification to explain** why you think the chosen setting can potentially improve the results. [*This is to imagine a real scenario. You need to think about how the performance might be improved, provide a justification, and then make changes. This implies that failing to improve the results is **acceptable** but we expect you provide a good justification when you make changes aiming to improve the results and such justification is sound.*]

3) For each split and each version of ALS, compute three metrics: the Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE). Put these RMSE, MSE and MAE results for each of the three splits in one **Table** for the two ALS settings in the report. You need to report 3 metrics x 3 splits x 2 ALS settings = 18 numbers. Visualise these 18 numbers in ONE single figure. [2 marks]

B. User Analysis

1) After ALS, **each user** is modelled by some factors. For each of the three time-splits, use *k*-means in PySpark with ***k=25*** to cluster all the users based on the user factors learned with the ALS **Setting 2** above, and find the top five largest user clusters. Report the size of (i.e. the number of users in) each of the top five clusters in one **Table**, in total 3 splits x 5 clusters = 15 numbers. Visualise these 15 numbers in ONE single figure. [2 marks]

2) For each of the three splits in Q3 A1, consider only the *largest* user cluster in Q3B1 and do the following only on the *training* set: [3 marks]

- Considering **all users** in the largest user cluster, find all the movies that have been rated by these users and their respective average ratings, named as *movies_largest_cluster*.
- Find those movies in *movies_largest_cluster* with an average rating greater or equal to 4 (>=4), named as *top_movies*.
- Use **movies.csv** to find the genres for all the *top_movies* and and report the top ten most popular genres (*each movie may have multiple genres, separated by* '|', *where **top** refers to the number of appearances in movies*). Report these 3 splits x 10 genres = 30 genres in one **Table**.

C. Discuss two most interesting observations from A & B above, each with three sentences: 1) What is the observation? 2) What are the possible causes of the observation? 3) How useful is this observation to a movie website such as **Netflix**? Your report must be clearly written and your code must be well documented so that it is clear what each step is doing. [2 marks]

Question 4. Research Paper Visualisation [set by Haiping - 6 marks]

You need to finish Lab 7 before solving this question.

Data: Use **wget** to download the NIPS Conference Papers 1987-2015 Data Set to the "Data" folder. Please read the dataset description to understand the data before completing the following tasks.

Tasks:

There are 5811 NIPS conference papers and we want to visualise them using PCA in a 2D space. We view each of the 5811 papers as a sample, where each sample has a feature vector of dimension 11463. **Note:** you need to carefully consider the input to PCA, i.e., what should be the rows and what should be the columns.

A. Use PySpark APIs to compute the top 2 principal components (PCs) on the NIPS papers. Report the two corresponding eigenvalues and the percentage of variance they have captured. Show the first 10 entries of the 2 PCs. [3 marks]
B. Visualise the 5811 papers using the 2 PCs, with the first PC as the x-axis and the second PC as the y-axis. Each paper will appear as a point on the figure, with coordinates determined by these top 2 PCs. [2 marks]
C. Discuss the most interesting observations from the visualisation in B, with two to three sentences. Your report must be clearly written and your code must be well documented so that it is clear what each step is doing. [1 marks]

Question 5. Searching for exotic particles in high-energy physics using ensemble methods [set by Tahsin 6 marks]

You need to finish Lab 8 before solving this question.

Data: In this question, you will explore the use of supervised classification algorithms to identify Higgs bosons from particle collisions, like the ones produced in the Large Hadron Collider. In particular, you will use the HIGGS dataset.

Use **wget** to download the data using the direct link: [http://archive.ics.uci.edu/ml/machine-learning-databases/00280/HIGGS.csv.gz]. You would then need to unzip the dataset first. For this purpose, you can use a tool like **gzip**.

You will apply Random Forests and Gradient boosting over a subset of the dataset in part A and over the full dataset in part B. As performance measures use classification accuracy and area under the curve.

A. Use pipelines and cross-validation to find the best configuration of parameters for each model (4 marks).
    a. For finding the best configuration of parameters, use 1% of the data chosen randomly from the whole set. *Hint: think of proper class balancing while picking your randomly chosen subset of data*. Pick three parameters for each of the two models and use a sensible grid of three options for each of those parameters (3 marks).
    b. Use the same splits of training and test data when comparing performances among the algorithms (1 mark).

    Please, use the batch mode to work on this. Although the dataset is not as large, the batch mode allows queueing jobs and for the cluster to better allocate resources.

B. Working with the larger dataset. Once you have found the best parameter configurations for each algorithm in the smaller subset of the data, use the full dataset to compare the performance of the two algorithms in the cluster (2 marks). **Remember to use the batch mode to work on this.**

a. Use the best parameters found for each model in the smaller dataset of the previous step, for the models used in this step (1 mark).
b. Once again, use the same splits of training and test data when comparing performances between the algorithms (1 mark).

The END of the Assignment

FAQs

Q1: **How to deal with "Error: spark-submit: command not found"**

A1: Please check out the suggested solutions in Lab 1 on "Common problem: spark-submit: command not found"

Q2: **How to reset your environment if you found that you've messed it up and encountered seemingly unrecoverable errors?**

A2:

login ShARC

qrshx

resetenv

rm ~/.conda

logout fully & then back in again

Start over with Lab 1 again to install everything

Q3: **Can we use libraries other than PySpark to generate the results?**

A3: For functionalities available in PySpark, you should use PySpark, particularly for the core computational part. If functionalities are not available in PySpark, you may use other Python libraries.

Q4: Is standard deviation needed for the validation curves?

A4: You can either use a fixed validation set or do cross-validation to get the train/validation scores to determine the optimal RegParam and plot the validation curves. If using a fixed validation set, there will be no std. If you do cross-validation, you will obtain a set of train/validation scores for each hyper-parameter, then you can compute the standard deviations and add them to the figure (e.g. as error bars).

Q5: Do we have to use the CrossValidator API for generating the validation curves?

A5: No. You can use other methods in PySpark or even implement functions by yourself to construct training and validation sets.