Project 1 - 25 points

In this project we will be working with Apache Spark on a dataset around true and fake job postings.

1. Setup (1 point)

As a first step, you need to install (if you don't have it already) Spark in your local machine. We will be using both scala and python (PySpark) for this project.

There are some tutorials available here: http://cgi.di.uoa.gr/~antoulas/books/ but you are encouraged to find and study additional relevant resources online. The overall project may require some research online on relevant topics for each question from your part as well as trial and error.

As a first step, visit https://spark.apache.org/downloads.html to check what version you would need, download and install it. Here are some non-authoritative (i.e., run at your own risk) installation instructions for Spark and Scala for Linux, Windows and MacOS. Please also install PySpark (https://pypi.org/project/pyspark/#files). All you would normally need to do is `pip install pyspark` on your machine.

Once you are done you should be able to run both the commands `spark-shell` and `pyspark` in your command line and see something like this:

```
Spark context Web UI available at http://127.0.0.1:4040
Spark context available as 'sc' (master = local[*], app id = local-1585922064592).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.5
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_91)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

You can check to see a GUI of the jobs running at http://127.0.0.1:4040 in your browser in this case.

2. Dataset (1 point)

We will be using the real/fake jobs data from the following Kaggle dataset:
https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction
Please register in Kaggle and download the dataset. Before starting, please ensure that you have read the dataset description on Kaggle's Web page. Unzip the file once you've downloaded it.

3. Simple Statistics in Scala (5 points - 1 point each question)

Start spark-shell and write scala code to read the csv[1] file in order to compute the following:
a) Number of lines in the csv file
b) Number of fake job postings

[1] Please ensure you are reading the file correctly into spark. It contains quotes and you will have to specify the right parameters.

c) Number of real job postings
d) top-10 most required education (e.g. Bachelor's Degree) in fake job postings
e) top-10 most required education in real job postings

4. **Analytics in PySpark (8 points - 2 points each question)**

Start PySpark and read the csv file. You are also welcome to use IPython with a spark notebook if you like, but you **cannot use SQL** commands to solve this part.

If there is any row in the data that has null in it you should ignore it (not consider it zero). Please compute the following:

f) Compute the average and standard deviation of the maximum salary in the range in fake job postings
g) Compute the median of the minimum salary in the range in real job postings
h) If you sorted the data for g) is what you computed correct? (Hint: you may have to do some transformation in the data). Are there outliers in the data, what are they? (Hint: there are three kinds of outliers). Correct/clean your data from the outliers and recompute f) and g)
i) Compute the 10 most popular bi-grams (i.e., sequences of 2 words) and tri-grams (i.e., sequences of 3 words) for fake and real job postings using the description field.

5. **Machine Learning in Python (10 points - 5 points each question)**

In this section we will be trying to build a machine learning model (specifically a classifier) to distinguish real vs fake job postings in Python. You don't have to use spark (but you can if you want to) and you are welcome to use existing libraries such as pandas, numpy, scikit-learn[2], etc.

j) Using only the telecommuting feature, build a naive Bayes classifier to distinguish between real and fraudulent postings. What is the precision, recall and f1 score of your model? Using the has_company_logo and has_questions features in addition to the telecommuting, build two additional classification models (different from naive Bayes) of your choice and report their precision, recall and f1 scores. What do you observe from the results?

k) Use the description feature of the dataset to create a classifier that predicts whether a job posting is fake (binary classification). Please note that the description is a text feature so it needs special handling. Experiment with additionally using the rest of the features in the dataset as well as derived features of your own. Examples of derived features can be: splitting up minimum and maximum salary, whether salary is not null, length of benefits, etc. This question is up to you to experiment and improve your model. You can also (but it is not expected to) use boosting or bagging (or any other technique of your choice) to deal with the class imbalance. Train two (different from j)) classification models with all the features and report their precision, recall and f1 score.

[2] You may find the following short tutorial useful:
https://www.digitalocean.com/community/tutorials/how-to-build-a-machine-learning-classifier-in-python-with-scikit-learn

Deliverables

- All files for your project need to be zipped in **one single zip file**. The file name should be in the following format: Lastname_Firstname_Project_X.zip (change X with the respective project number i.e., 1 for project 1, 2 for the second project and so on).
- Please email the zip file to the instructor by the deadline. **There are absolutely no exceptions and no extensions.**
- Inside the zip file, please include a) your source code b) any results that are part of the project c) a README text file that explains how to compile (if needed) and run your code and d) a short and concise report (preferably pdf - no more than 10 pages) describing your work and your approach to solving the project. **Please do not include the original or any derived data in the zip file.**

Important Notes

- Please check the class Web site: http://www.di.uoa.gr/~antoulas/m111 regularly for announcements and/or clarifications to the project. Announcements will show up there and will also be sent to the mailing list.
- You are free to make any assumptions you may need to along the way as long as you document them in your report.
- The project is meant to be worked on by the student submitting and only that student. In the event that there is a submission not worked on by the student, then the student fails the class.
- Although students are expected (and encouraged) to chat among them on potential solutions and approaches, sharing code and solutions is strictly not allowed. In the event that two or more students provide submissions that have common pieces, everyone involved (regardless of who is at fault or who copied from whom) fails the class.