MIE 1624 Introduction to Data Science and Analytics – Winter 2023

Assignment 3
Due Date: 11:59pm, March 31, 2023
Submit via Quercus

Introduction:

For this assignment, you are responsible for answering the questions below based on the dataset provided. You will then need to submit a 3-page report in which you present the results of your analysis. In your report, you should use visual forms to present your results. How you decide to present your results (i.e., with tables/plots/etc.) is up to you but your choice should make the results of your analysis clear and obvious. In your report, you will need to explain what you have used to arrive at the answer to the research question and why it was appropriate for the data/question. You must interpret your final results in the context of the dataset for your problem.

Background:

Data science, analytics, AI, big data are becoming widely used in many fields, that leads to the ever-increasing demand of data analysts, data scientists, ML engineers, managers of analytics and other data professionals. Due to that, data science education is now a hot topic for educators and entrepreneurs.

In this assignment, you will need to design a course curriculum for a new "Master of Business and Management in Data Science and Artificial Intelligence" program at University of Toronto with focus not only on technical but also on business and soft skills. Your curriculum would need to contain optimal courses (and topics covered in each course) for students to obtain necessary technical and business skills to pursue a successful career as data scientist, analytics and data manager, data analyst, business analyst, AI system designer, etc. You are required to extract skills that are in demand at the job market from job vacancies posted on http://indeed.com web-portal and apply clustering algorithms to group/segment skills into courses.

You are provided with a sample Python code to web-scrape job postings from http://indeed.com web-portal, that you would need to modify for your assignment. You can decide on the **geographical locations** of the **job postings** (e.g., Canada, USA, Canada and USA) and job roles (e.g., "data scientist", "data analyst", "manager of analytics", "director of analytics") of the posting that you will be web-scraping, but your dataset should contain at least 1000 unique job postings.

Experiment with different Natural Language Processing (NLP) algorithms to extract skills (features) from the web-scarped job postings. You may manually define your own list of keywords/key-phrases (N-grams) that represent skills, e.g., "Python", "R", "deep learning", "problem solving", "communications", "teamwork", or use pre-trained NLP algorithms, such as ChatGPT, that automatically extract skills/features from your dataset.

Finally, you will need to use skills extracted from job postings as features and run two clustering algorithms to create clusters of skills that can be interpreted as courses. First clustering algorithm that you are required to use is **hierarchical clustering algorithm with one feature**, where that feature represents a distance between each pair of skills. The idea is that if a pair of skills is found together in many job postings, those two skills would be required together on the job, and it makes sense to teach those skills (topics) together within the same course (cluster). Using this idea, you will need to define your own distance measure, create a dendrogram (see slides 43-44 of "Lecture 8 – Advanced Machine Learning" for an example), and interpret each cluster as a course. For the second clustering algorithm you can **choose between k-means and DBSCAN**. You will be required to use **at least 10 features as inputs for your second clustering algorithms.** As in the first case, you are required to interpret each cluster as a course.

Based on your first and second clustering analysis separately, create a sequence of **8-12 courses. For each course include 3-8 topics (based on skills) that should be taught in each course.** Please list your courses in a logical order, i.e., a course that requires another course as a pre-requisite should be listed after the pre-requisite course. You can use your own judgement for deciding about a logical sequence of courses or try to interpret your clustering results for that. For visualizing your course curriculum, feel free to use Python or any other software like Tableau and Power BI. As a bonus, you are asked to combine your two course curricula into one course curriculum that you propose to be taught at the master program.

Learning objectives:

1. Understand how to clean and prepare data for machine learning, including transforming unstructured web-scaped data into structured data for analysis. Convert categorical features into numerical features and perform data standardization/normalization, if necessary, prior to modeling.
2. Understand how to apply unsupervised machine learning algorithms (clustering) to the task of grouping similar items.
3. Interpret your modeling results and visualize those.
4. Improve on skill and competencies required to collate and present domain specific, evidence-based insights.

Questions:

The following sections should be included but the order does not need to be followed. The discussion for each section is included in that section's marks.

1. **[1 pt] Data collection and cleaning:**
    a. Adapt provided web-scraping code
        i. Decide on geographical location and job role/title
        ii. Scrape Indeed job postings for data (must include at least job title, location, company name, job description, and salary)

2. **[3 pts] Exploratory data analysis and feature engineering:**

    a) Engineer features for clustering analysis
        i. Define skills manually from your own knowledge and by using ChatGPT. Use OpenAI's API to access ChatGPT (`gpt-3.5-turbo` model) and request a list of skills to use in extraction. For your analysis combine your skills list generated manually and those generated with ChatGPT.
        ii. Extract skills using N-grams or pre-trained NLP algorithms.
    b) Visualize key information
        i. Generate at least two visual depictions of the information you've collected and extracted (e.g., count of each skill vs. job titles or company name, average salary of skill, etc.)

3. **[3 pts] Hierarchical clustering implementation:**

    a) Implement **hierarchical clustering** algorithm
        i. Generate a distance matrix for to describe the relationship between skills.
        ii. Develop a course curriculum based on clustering results (8-12 courses with at least 3 skills/topics covered in each)

4. **[3 pts] K-means or DBSCAN clustering implementation:**

    a) Implement **k-means clustering** algorithm or **DBSCAN clustering** algorithm
        i. Engineer 10 unique features to describe each skill for clustering (e.g., skill frequency, average salary for skill, binary indication of soft or hard skill, etc.). Your distance matrix used in Section 3 may be incorporated but only counts as one feature.
        ii. Develop a course curriculum based on clustering results (8-12 courses with at least 3 skills/topics covered in each)
    b) Use the elbow method to determine the optimal *k* number of clusters for **k-means clustering** or *eps* value if using **DBSCAN clustering**

5. **[3 pts] Interpretation of results and visualizations:**

    a) Generate a **dendrogram** from **hierarchical clustering** algorithm
    b) Generate a labeled scatterplot from **k-means clustering** algorithm or **DBSCAN clustering** algorithm
    c) Include visualization of elbow method used to find optimal *k* number of clusters for **k-means clustering** or *eps* value if using **DBSCAN clustering**

6. **[2 pts] Discussion and final course curriculum:**

    Discuss your results. Present and justify your final course curriculum. You may select curriculum from Section 3 (hierarchical clustering algorithm) or from Section 4 (second clustering algorithm) as your final course curriculum. Insufficient discussion will lead to the deduction of marks.

7. **[+1 bonus pt] OpenAI to describe clustering results:**

   If you can further implement OpenAI's API to describe your clustering results (e.g.,, asking ChatGPT to describe what is common amongst the clusters) then you will get one bonus point (your max assignment mark cannot exceed 15 pts including bonus). For ChatGPT API, use `gpt-3.5-turbo` or newer model (newer model may be available if GPT-4.0 API is released before the assignment deadline).

Submission:

1) Produce an IPython Notebook (.ipynb file) detailing the analysis you performed to answer the questions based on your data set.

2) Produce a 3-page report explaining your response to each question for your data set and detailing the analysis you performed. When writing the report, make sure to explain for each step, what you are doing, why it is important, and the pros and cons of that approach.

Tools:

- **Software:**
  - **Python Version 3.X** is required for this assignment. Make sure that your Jupyter notebook runs on Google Colab (https://colab.research.google.com) portal. All libraries are allowed but here is a list of the major libraries you might consider: Numpy, Scipy, Sklearn, Matplotlib, Pandas, NLTK, OpenAI.
  - No other tool or software besides Python and its component libraries can be used to collect your data and touch the data files. For instance, using Microsoft Excel to clean the data is not allowed. Please dump your web-scraping results into a file, submit it with the assignment, and comment your web-scraping code in the notebook.
  - Upload the required data file to your notebook on Google Colab – for example,
    ```
    from google.colab import files
    uploaded = files.upload()
    ```
  - You are allowed to use any software for visualizing your course curricula (Tableau, Power BI), but you should use Python for everything else.

- **Required data files to be submitted:**
  - **webscraping_results_assignmnet3.csv**: file to be submitted with the assignment
  - The notebook will be run using the local version of this data file. Do not save anything to file within the notebook and read it back.

- Auxiliary files:
  - **Indeed_webscraping.ipynb**: the code used to web-scrape job postings from Indeed web-portal. Please modify this code for your own needs.

- In order to get around Indeed's anti-scraping service **download geckodriver** (HTTP API for Firefox and other Gecko browsers) from https://github.com/mozilla/geckodriver (Code>ReadMe>Downloads>Releases) or find your own alternative method. Note that geckodriver requires you to use Firefox or another Gecko based browser.

What to submit:

1. Submit via Quercus a Jupyter (IPython) notebook containing your implementation and motivation for all the steps of the analysis with the following naming convention:

   lastname_studentnumber_assignment3.ipynb

   Make sure that you **comment** your code appropriately and describe **each step** in sufficient detail. Respect the above convention when naming your file, making sure that all letters are lowercase and underscores are used as shown. **A program that cannot be evaluated because it varies from specifications will receive zero marks.**
2. Submit via Quercus a csv file with your web-scraping results with the following naming convention:

   webscraping_results_assignmnet3.csv
3. Submit a report in PDF (up to 3 pages) including the findings from your analysis. Use the following naming conventions **lastname_studentnumber_assignment3.pdf**.

Late submissions will receive a standard penalty:

- up to one hour late - no penalty
- one day late - 15% penalty
- two days late - 30% penalty
- three days late - 45% penalty
- more than three days late - 0 mark

Other requirements and tips:

1. A large portion of marks are allocated to analysis and justification. Full marks will not be given for the code alone.
2. Output must be shown and readable in the notebook. The only file that can be read into the notebook is the file **webscraping_results_assignmnet3.csv** with your web-scraping results.
3. Ensure the code runs in full before submitting. Open the code in Google Colab and navigate to Runtime -> Restart runtime and Run **all** Cells. Ensure that there are no errors.
4. You have a lot of freedom with how you want to approach each step and with whatever library or function you want to use. As open-ended as the problem seems, the emphasis of the assignment is for you to be able to ***explain the reasoning behind every step***.