CP421 Data Mining - Assignment 1
Due Date: Oct 11, 2023 at 11:59 PM

Assignment Submission Guidelines

1. **File Naming**: Ensure your assignment file is named in the following format: your network login followed by the assignment number. For instance, if the username is "barn4520" and you're submitting Assignment 1, the file should be named "barn4520_a01.ipynb".
2. **Assignment Format**: All assignments should be completed using Jupyter Notebook. Once you're done, ensure you run all the cells to verify their functionality, then save your work as a ".ipynb" file. For theoretical or conceptual queries, provide your responses within the notebook using markdown cells. Coding segments must be well-documented for clarity.
3. **Submission Platform**: All submissions must be made via the MyLearningSpace website. We do not accept assignments through email.
4. **Late Submissions**: If you submit your assignment within 24 hours post the deadline, your grade will be reduced by 50%. Unfortunately, we cannot accept submissions made beyond 24 hours from the deadline, and such submissions will receive a grade of 0.
5. **Plagiarism Policy**: All submitted code will undergo a plagiarism check. Engaging in plagiarism can have significant academic consequences. Using generative AI to aid in or fully complete your coursework will be considered academic misconduct.

PART 0: Preliminary Steps

0.1 Configuring Your Software Environment

Begin by establishing the software ecosystem that we'll be utilizing throughout this course. As discussed during our sessions, you're free to set up the necessary software on your personal device. If you opt for this route, it's crucial to verify that all installed components are up-to-date.

For our programming tasks, we'll predominantly be using Python, supplemented by a few Python libraries. Most of these tools are encompassed within the SciPy stack, an open-source collection tailored for applications in mathematics, science, and engineering. For ease of installation, we recommend the Anaconda Python Distribution, which conveniently bundles the SciPy stack and is compatible across Linux, Mac, and Windows platforms.

Ensure your system has the following components installed:

- **Python**: A versatile, object-oriented programming language.
- **NumPy**: Essential for scientific computations in Python.
- **SciPy**: Designed for advanced math, science, and engineering applications.
- **Matplotlib**: Ideal for creating 2D visualizations in Python.
- **pandas**: Offers powerful data structures and tools for efficient data analysis.
- **IPython**: Provides an enhanced interactive Python computing experience.
- **scikit-learn**: A comprehensive library for machine learning in Python.

0.2 Initiate Your First Notebook

Kickstart by crafting an IPython Notebook. Incorporate the sample code provided below and feel free to introduce any personal touches (for instance, you can display your name). Ensure you tweak or append at least one line of code. To set up a new IPython Notebook, launch the Jupiter Notebook from your terminal. This action will usher you into the IPython web interface, from where you can opt for 'New Notebook'. Once you've finalized your edits, rename your assignment, and save it, which will produce an '.ipynb' file.

0.3 Testing Sample Python Code with Essential Modules

Your finalized notebook should bear resemblance to this reference:
`https://nbviewer.jupyter.org/github/wlucp421/assignment/blob/master/a1_samplecode.ipynb`.

1 PART I: 10 points

1.1 Getting statistics: 3 points

Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70. Compute the answers for the following questions using Python.

1. What is the mean, median, and mode of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).
2. Give the five-number summary and the interquartile range (IQR) of the age data.
3. Show the histogram and boxplot of the data.

1.2 Document Similarity: 7 points

Given a set of documents, we want to find pairs of documents that are similar to each other based on their content. The document similarity is determined by the Jaccard similarity of their word sets. Instead of comparing every pair of documents, which is computationally expensive, we will use Locality Sensitive Hashing (LSH) to find similar document pairs efficiently.
Task:

1. Convert the documents into sets of **words** (shingles).
2. Use MinHash to approximate the Jaccard similarity of the document pairs. Set the number of hashes as 128 and the hash function as $h = (123 \times x + 456)\%9999991$
3. Use LSH to find candidate pairs that are potentially similar. Set the number of bands as 16, and the number of rows per band as 8.
4. Compute the Jaccard similarity for the candidate pairs and output pairs that have similarity above a threshold.

Given the following set of documents, identify pairs with a Jaccard similarity greater than 0.3.

D1. Locality sensitive hashing is very useful for large scale applications.

D2. LSH is useful for large scale data processing and similarity detection.

D3. Hashing techniques are prevalent in large data processing.

D4. Locality hashing helps in efficient large scale computations.

Given another set of documents below, please answer which pairs of documents have a Jaccard similarity greater than 0.5 based on their content?

D1. Locality sensitive hashing is useful.

D2. Locality sensitive hashing is beneficial.

D3. Locality sensitive techniques are useful.

D4. Locality hashing is extremely useful.

You are required to implement the LSH algorithm on your own. Utilizing pre-existing libraries like 'datasketch' for MinHash and LSH calculations, or not adhering to the specified parameter settings, will result in a 5-mark deduction.

2 Cross-selling recommendation with Apriori: 10 points

Cross-selling involves offering related or complementary items to a customer. It's a standout strategy in the marketing world. Take, for example, a bank customer with a mortgage. The bank might suggest they consider a personal credit line or an investment option like a CD.

For online shopping platforms, a straightforward way to recommend products is by suggesting items that other customers often view together. Here's where the Apriori algorithm, which we discussed in class, comes into play. Imagine you want to propose new items to a shopper based on their browsing history on an e-commerce site. For this task, you can use the Apriori algorithm to pinpoint products that are typically viewed together using the given online behavior dataset, 'crossselling.txt', from MLS. In this dataset, each line contains eight-character strings, each representing an item ID from a particular browsing session, separated by spaces. Some sample sessions might look like:

```
ELE17451 GRO73461 DAI22896 SNA99873 FRO18919 DAI50921 SNA80192 GRO75578
ELE17451 ELE59935 FRO18919 ELE23393 SNA80192 SNA85662 SNA91554 DAI22177
ELE17451 SNA69641 FRO18919 SNA90258 ELE28573 ELE11375 DAI14125 FRO78087
```

1. (6 points) Set the minimum support s =100, i.e., product pairs need to occur together at least 100 times to be considered frequent.

   (a) Identify all pairs of items (X, Y) such that the support of $\{X, Y\}$ is at least 100. For all such pairs, compute the confidence scores of the corresponding association rules: $X \rightarrow Y$ and $Y \rightarrow X$. Sort the rules in decreasing order of **confidence** scores.

   (b) Identify item triples (X, Y, Z) such that the support of $\{X, Y, Z\}$ is at least 100. For all such triples, compute the confidence scores of the corresponding association rules: $(X, Y) \rightarrow Z$, $(X, Z) \rightarrow Y$, $(Y, Z) \rightarrow X$. Sort the rules in decreasing order of confidence scores.

   If there are any ties, resolve them based on the lexicographical order of the rule's left-hand side. In your submission, please include the following details:

   (a) Number of frequent singletons;

   (b) Number of frequent pairs;

   (c) Number of frequent triples;

   (d) Total number of rules generated from frequent pairs;

   (e) Total number of rules generated from frequent triples;

   (f) The top 5 rules sorted by confidence score.

2. (2 points) Set the minimum support s=250, repeat the above tests, and report your observations from (a) to (f).

3. (2 points) Analyze the outcomes of both experiments and describe the insights you've gathered in your own language.

Note: The aim of this question is to deepen your comprehension of the underlying principles of frequent pattern mining and association rules. Utilizing pre-existing functions from libraries like mlxtend is not allowed.