FT5005: Assignment 2 (10 Marks)

**Due: 11:59pm of March 31 Sunday 2024**

**Submission Formatting Requirements**

1. Please submit one zip file that includes all \*.ipynb of your codes for Questions #1-3.
2. You can add comments to help TA understand your codes. Avoid including unnecessary codes or comments.
3. Remember to set random seeds so TA can verify your results when needed.
4. TA can deduct your points up to 2 out of 10 if you included unnecessary codes and comments.

**Q1 (3 marks) Dictionary Approach**

Please use the data in “zacks\_arguments.csv” to practice Q1. The dataset includes part of the content from stock analysts’ reports on Zacks.com. This is a tiny dataset to save your time in testing your codes. The columns are

1. ID: This is just a sequential ID.
2. report\_name: filename. This is not useful in A3.
3. ticker: This is for your reference to find the real company name. This is not useful in A3.
4. report\_date: This is just for your reference.
5. arguments\_clean: This is the most important column. **This column includes reasons to buy or reasons to sell in each report.**
6. label: reasons to buy or reasons to sell (250 Buy and 193 Sell, 443 rows in total).

Please pre-process the “arguments\_clean” by the steps covered in Week 6. This includes

1. Convert all letters to lowercase
2. Remove all special characters. This includes extra spaces. For example “A B” => “A B”. This is one thing that I forgot to cover in slides/sample codes and is useful in practice when your dictionary has compound words “United States”, which cannot be an exact match to “United States”. Several more comments below:
   1. This dataset is very messy because data are extracted from pdf files without structure. You may need to be more careful in data pre-processing in this kind of case in the future.
   2. Remember to apply the same pre-processing to dictionaries. In A3, the list of positive words is in LM2018P.csv, and the list of negative words is in LM2018N.csv. These two are from the most famous dictionaries for financial usage. If you have time, you can also try comparing results to other dictionaries. Most evidence shows that LM dictionaries are indeed better for financial documents but dictionary-based approach has its limit in accuracy.
   3. Our two lists of words are unigram and relatively clean. But they are capitalized.
3. Tokenization. Please use unigram.
   1. If your dictionary has words like “United\_States”, you may need to pre-process your document so that all “United States” are converted to “United\_States” so you can count accurately.
4. Apply stopword list
   1. For the dictionary approach, you may need to try whether you should apply stopword list. Some of your dictionary words could be in the stopword list too. In that case, you may need to judge whether those words should be counted or not.
5. Apply Lemmatization
   1. Many people do not apply this step when using the dictionary approach. From my observation, I see quite a number of applications that should be conducted with this step.
6. Again, you may need to make sure the dictionary is also pre-processed in the same procedure.

Now please use the LM2018 positive word list and negative word list to calculate the word count and sentiment score of all 443 arguments. For word counts, just use exact match after pre-processing. For formulas, please try (P-N)/(P+N+1). I added 1 to the denominator to avoid the issue about divided by 0.

In our task, we have the label for comparison. Please compute 3 cases of results. The 3 cases are

1. This formula with Lemmatization, without negate
2. This formula with Lemmatization and also negate (if there is a negate word right before the positive/negative word, we count it as the opposite sentiment.)
3. NLTK Vader’s compound score (see the sample code).

In each one of the 3 cases, first sort your sentiment scores and report how many of your top 100 positive arguments (by the ranking of your own sentiment score) are indeed positive arguments. Ideally, all your top 100 words should be “Buy” cases. Which one out of the six methods is the best method?

For this question, I also provide a template of codes from my RA. I intentionally hide the last part of the codes for you to complete it. You can modify that version or use any other sample codes.

**Q2 (4 marks) Text Classification**

Now we use another larger dataset to practice text classification and topic modelling. Different from the dictionary approach, these two methods are more applicable to a larger dataset. This dataset is “startups.xlsx”. More information is below.

* Due to limitation of Canvas, please download the data from https://docs.google.com/spreadsheets/d/1jErr0bwPcaJzSplqiVMbMhZQVCcBpmWq/edit?usp=sharing&ouid=117113194800900672794&rtpof=true&sd=true
* There are 60,089 start-ups and the business description of those start-ups.
* The main textual content is in the column “description”.
* Two labels are “industry” and “industry2”. “industry” has only 3 types of industries. “industry2” has 10 types of industries.
* This dataset is created by: first I only select 3 industries. Also, I exclude the description that includes less than (<=) 200 letters. Also, in the description, the last few words are company keywords. This may not happen in other datasets. But this can improve the classification performance and the topic modelling performance and could make A2 easier.

First, please apply all data-pre-processing steps in Q1. Next,

1. Please try using TF-IDF as the vector representation of your document
2. The classification method is LightGBM
3. The label is “Industry”. There are only 3 types.
4. Please use GridSearchCV with 5-fold cross-validation to tune your LightGBM.
5. Report your 5-fold cross-validation accuracy. In other words, simple accuracy is the classification metric that you pay attention to.
6. Grading will be partially dependent on your accuracy value. TA can deduct up to **2 points** if your accuracy is much worse than the class: worse than 80% of the class and also the accuracy is so poor that clearly something is wrong. If your prediction performance is unreasonably high, we will also deduct up to 2 points because there is a bug in your code. Most of the grading will be based on the correctness of your code in implementing text classification and the associated data pre-processing.

**Q3 (3 marks) Topic Modeling**

Please apply topic modeling on the same dataset after the same data pre-processing in Q2.

1. You can modify the sample code in Week 7 to convert each document to “bag of words” list.
2. Apply the classical LDA topic-modeling on this corpus. The number of topics is
   1. Try 3 topics.
   2. Try 10 topics.
3. (**1 mark**) For the 3-topic case, report the topic that is most similar to industry=”Information Technology”. Please report the precision and recall of (industry=”Information Technology”). Also, print the top 10 keywords.
   1. Your topic modeling output is the predicted industry type. In general, we cannot calculate precision-recall for unsupervised learning problems. You need to read the keywords to decide the naming of each topic. But in this case, you also have the actual industry type. So you can calculate the precision and recall as the performance metric and we can objectively observe the clustering performance of topic modelling.
4. **(1 mark)** For the 10-topic case, report the topic that is most similar to industry2=”Computer Software and Services”. Report the precision and recall of your topic modelling output of (industry2=”Computer Software and Services”). Print the top 10 keywords.
5. **(1 mark)** Write a grid search CV code with LDA by the “Example\_LDA\_2” sample code to decide 3 or 10 topics is a better model by perplexity.

Optional Practices:

1. (optional) Results of classical LDA may not be good. You can move those useless top keywords (especially those in top 10 words in several dictionaries) to stop-word list and re-run again to see whether you can improve the clustering performance.
2. (optional) There are many other topic-modelling packages, and you can practice trying those packages on this dataset. I included a sample code from my RA for your reference and practice. For this sample, you may need to install several new packages to make it work. See the optional sample code about guidedlda.