Problem Set 2

Business 35137

Spring 2025

Due: May 5th

1. Download the `topics.csv` file from canvas. This file contains labeled attention to topics from structureofnews.com. Additionally, download the `macro.csv` file from canvas. This file contains a series of financial and macroeconomic outcome variables.

   (a) Using the `mret` (for market return) column from the `macro.csv` file, fit lasso for a range of penalty parameters to the topics data. Select the penalty that yields five non-zero coefficients. Then run OLS with these five topics. What is the R^2? Interpret the topics selected.

   (b) Repeat this procedure for `vol`, `indpro`, `indprol1` (industrial production growth one-period in the future), and each the `indvol` columns. Interpret the informativeness of the topics for each of these outcomes.

   (c) Using what you learned in the first problem set, let's now try our best to forecast industrial production growth in real time. Provide some reasoning for your modeling decisions.

   (d) Next, download the `articles.pq` file from canvas. This file contains headlines from the *Wall Street Journal*. Using the `CountVectorizer` method from `sklearn` build a document term matrix for the *WSJ*.

   (e) Next, repeat the contemporaneous exercises from part (a) and (b) using the counts. How many non-zero counts do you need to recover the same R^2? What does that say about the informativeness of the counts vs. topics?

(f) Using the counts attempt to form the best forecasting model for industrial production growth. How well can you do relative to the topics?

(g) Convert the raw counts into tf-idf and repeat the exercises from part (e) and (d). Summarize the differences between the tf-idf and raw count approaches. Which terms are most important in either approach?

2. Next, using the same `articles.pq` file, we're going to explore using LLMs for generation with the `generation.py` script from canvas.

   (a) Attempt to form a prompt that generates the topics discovered in 1.a and 1.b. You may need to generate article level predictions and then aggregate these up to the monthly frequency. What R^2 can you achieve with this approach?

   (b) How much does prompt engineering change your results? Try the following:

      i. Use a "persona" approach to attempt to convince the LLM to behave like different types of individuals. For example, try to convince the LLM to behave like a "bull" or a "bear". How much does this impact your results?

      ii. Use temperature to attempt to control the randomness of the LLM. How much does this impact your results? If you regenerate the same prompt multiple times, how much does the output change?

      iii. Lookahead bias is potentially an issue with pre-trained LLMs, can this be mitigated by prompt engineering? Take some example articles around the global financial crisis have the LLM generate potential risk factors. By telling the LLM to ignore the future, can you mitigate lookahead bias?

   (c) Using the generation approach, how well can you forecast industrial production growth? Document your approach and reasoning.

3. Finally, using the same `articles.pq` file, we're going to explore using embeddings with the `embeddings.py` script from canvas.

   (a) Form embeddings for the *WSJ* headlines. Then attempt to repeat exercises 1.a and 1.b using the embeddings. How well can you do relative to the topics?

   (b) Select a couple representative topics from the `topics.csv` file. Can you recover these topics from the embeddings?

   (c) Similarly, how well can you recover the generated series from 2 using the embeddings?

   (d) Using the embeddings you've formed, how well can you forecast industrial production growth? Document your approach and reasoning.