MSSP Portfolio

Ajay Krishnakumar

April 2024

Table of Contents:

Survey Validation: Autism Spectrum Disorder in Indian Children

Introduction

Our client, a post-doctoral researcher at the Child and Family Health Lab, was interested in what parents of children with autism felt about the effects and success of Speech Language Therapy (SL). In order to tie these with other variables such as age and income, they circulated a survey with questions about both groups of variables. Our client came to us wanting to check the reliability of the survey results and also that the questions about Access to and Quality of SL did actually measure those variables.

Data and Methods

We started by cleaning our client's data, clarifying variable meanings and how each one pertained to their objectives. As we did this, we identified some missing data and some inconsistencies in responses. We then used EDA to look at patterns in the data, to visualize differences between the two groups in the study: those whose children went to Speech Language Therapy and those whose children did not.

In addition to the EDA, we conducted research to figure out how to apply statistical methods of establishing reliability and validity (we focused on Cronbach's alpha and factor analysis) to a small sample size. Our client had 110 data points split roughly 60-50 between two groups. Further the questions were different between the different groups so these tests would have to be conducted on each group separately, further exacerbating the sample size problem. We found that for sample sizes between 50 and 100, we would need to carry out principal component analysis, remove any components with a loading less than 0.4 and then re-run the PCA. If the eigenvalue of the first principal component was then above 4(Yurdugül, H. (2008) Minimum sample size for Cronbach's coefficient alpha: A Monte-Carlo study. Hacettepe Üniversitesi Eğitim Fakültesi Dergisi, 35, pp. 397-405) and if at least four loadings were above 0.8 (Guadagnoli, E. and Velicer, W. F. (1988) Relation to sample size to the stability of component patterns. Psychological Bulletin, 103(2), pp. 265-275.) we could proceed with running Cronbach's alpha and factor analysis.

Results

Unfortunately, the principal component analysis did not satisfy the requirements that would have allowed our statistical tests of reliability (Cronbach's alpha) and validity (confirmatory factor analysis). We nevertheless identified ways in which the client might still interpret their data, including giving them the visualizations we created. We were sure to emphasize that we were not saying that the client's data was not reliable and not valid, only that we could not say with any degree of certainty that they were. This was a very important learning experience for

us, both in terms of interpreting these kinds of statistical results but also in terms of the ethics of statistical practice and balancing those with the needs of clients.

Analyzing Large Language Model Performance across Document Retrieval Tasks

Introduction:

With the increasing permeation of Large Language Models into facets of work and academic life and because of their ability to simplify a lot of potentially tedious tasks, it's only natural to wonder what applications, if any, they might have to business. Our Clients, the Data Science COE at Fidelity Investments were interested in and working on precisely such applications. They came to us hoping to learn more about the range of LLM products available, restrictions on their use, differences in pricing, limitations and their relative strengths and weaknesses. Specifically, they were interested in how these models might perform on Document Retrieval tasks. Broadly speaking, these are tasks that involve reading or parsing a document and being able to pick up and convey key information from them, often in the form of answers to questions. Another question our Fidelity Partners were interested in answering concerned the size of the models. Model size refers to the number of parameters used to train the model and can range from a few billion to hundreds of billions. Are larger models always better than smaller models in performance, in practicality, and in the possible extra value they bring given extra cost?

Data and Methods:

We started by using Huggingface as a central repository of information about models, their parameter size, token limits(which limit how many characters can be fed into the LLM and which translate to characters at a ratio of roughly 4:3) and cost. We then looked at the Github pages and/or websites of each model to confirm the details. From this we shortlisted thirteen models we would use for our evaluation, including models from Intel, Meta, Mistral, and others.

We identified three categories of documents we would use, designed to encompass varying styles, levels of technicality, syntax and semantic relationships. These were: Short Stories, Code and Latex Documents. For each document category, we came up with three general questions we could ask LLMs. For example, for short stories, we would ask the LLM to tell us who the main character was, what person the story was written in and whether it had any dialogue. For code, we asked LLMs to identify what and how many parameters a function had, and what the return types was. We made sure to pick documents whose token size would not exceed the lowest maximum token size for the models we were considering.

At the same time, we looked into methods of prompting large language models, exploring a variety of methods to optimize the responses. These included few shot questions, chain of thought and assigning a role to the LLM. Ultimately, for the sake of balance between ease of API querying and getting good responses, we used a combination of one shot and role assigning.

Once we had identified the models, documents, questions, and prompts. we created API endpoints on Huggingface to which we could pass queries. Models which we couldn't access using the Huggingface API, we instead deployed on the BU Shared Computing Cluster (SCC). We

wrote a script to automate querying the LLM and retrieving its responses. This was repeated at three different temperatures for each model (temperature is a measure of 'creativity' of the model's responses. Loosely the higher the temperature, the more the LLM's response is likely to change each time you repeat the question, and the more it will get simple tasks like information retrieval wrong. In a little more detail given a series of words, the model will 'predict' the next word by choosing from a set of likely next words determined by its training. At a low temperature it will pick the most likely next word most of the time, at a higher temperature, it might pick less likely words). In this fashion, we generated roughly 35,000 data points.

We developed a response checking script in Python which used string matching and regex to identify if each of our 35,000 responses were accurate. We also picked several hundred points at random and manually checked that the responses were accurate.

We then used this data to examine three things:

1. The differences in response accuracy between large models (> 10 billion parameters) and small models (<10 billion parameters).
2. The differences in accuracy at different temperatures.
3. The difference in rate of blank responses between deployed models and API accessed models.

Results

We found that the accuracy of large models for document retrieval tasks was higher on average than that of small models, but the best performing model was a small model with 7 billion parameters.

For information retrieval tasks, a lower temperature is preferred to higher temperatures.

While using an API resulted in faster querying and easier implementation than deploying the model, it also returned a lot of blank responses, while deployment on the SCC returned no blank responses.

Conclusions

For simple, straightforward tasks, the additional monetary and time cost of using large models is not worth the improved performance they might bring, especially given that choosing the right small model will provide results that outperform those of larger models. That said, the overall accuracy we observed was not - to put it mildly – great. Part of that can be rectified by more efficient prompting but finetuning a model might also be worth the investment if it returns significantly better accuracy.

Predicting CO2 Emissions Auction Prices under EU Cap and Trade Regulation

Introduction

The European Union's cap-and-trade regulation allows companies, factories, and power plants a certain allowance of emissions of CO2 each year. One allowance equates to a ton of CO2 or equivalent amount of other greenhouse gas. A large percent of this allowance is allocated to the organizations in question at 'no cost' and this constitutes their respective caps. For example, the allowance the EU set for carbon emissions across the aviation industry in 2021was 24.5 million, i.e., 24.5 million tons of CO2. 20.7 million was issued for free, and the amount that each entity received constituted its 'cap' on emissions. The remaining 3.8 million allowances were auctioned. Heavy fines are levied by the EU on any entity that breaches its cap - in the order of a 100 Euros per ton of CO2 equivalent emitted above the cap.

The auctions are structured as follows: The bids are sorted in descending order of price. The bid volumes are added up going down that list and the price at which the total volume bid for crosses the total available is the clearing price.

My goal was to predict this auction price and to identify what, if anything, important predictors of it might tell us about the system and the motivations of covered organizations.

Data and methods:

Data for the model came from the EU Emissions Trading System, who publish yearly data on the auction, clearing prices, maximum bid price, auction volume and average number of bids per bidder. After examining correlations of all the predictors in the data set and after conducting EDA, I chose not to use predictors in the data set that weren't the ones listed above. To get more useful information, especially of macroeconomic trends that might affect auction prices, I pulled brent crude oil prices and the value of the FTSE-100 index, joining them with my dataset by date. The data had EU-wide auctions, German auctions, Polish auctions and aviation allowance auctions and these would be the random effect 'levels' in my mixed-effects model.

The first challenge with the data was the time-series nature of auction prices, FTSE-100 values and oil prices. I used partial autocorrelation plots to identify that each of those variables had an AR (1) structure and calculated a lag for each which I used as predictors in the model.

The training set constituted data from 2017 up till July 8th, 2022. The validation set was the remaining data up until December 2023, which amounted to making an 80-20 split.

I fit no pooling models, a complete pooling model and a partial pooling model that varied intercept (I found that varying slope did not provide improved interpretation or predictive power) for each random effect. Then I calculated the mean squared error for each model using

my validation data and repeated this with a null model that just used the previous price as a predictor.

Results:

The predictive accuracy of the complete pooling model was best of the three I examined but none performed as well as the null model. This makes sense considering the heavily autocorrelated structure of the variables. I also found that across all the models, oil prices, FTSE and maximum bid price were very significant predictors.

Conclusions and Discussion:

The results of the predictive power of the models did not come as a surprise given the time-series nature of the data and use of more sophisticated time series analysis and forecasting would yield better results. This is an avenue I would like to pursue in future work.

Oil prices and FTSE-100 were very significant predictors of the CO2 auction prices and go a long way to explaining how the price increased from around €15 per ton in 2017 to over €90 per ton in 2023. They were good long-term predictors of CO2 emissions auction price. The maximum bid price was also a strong predictor, and this was especially interesting given that a viable strategy to win the auction is to bid as high as you can because the clearing price is inevitably lower. This speaks a little to the value polluting corporations place on emissions because you would only bid the maximum price if you needed to guarantee winning emissions allowances; i.e: you are exceeding your cap significantly.

Post Stroke Capillary Stalling Dynamics in Mice

Introduction

Our client, a biomedical engineering researcher at BU, aimed to investigate the effects of stroke on capillary stalling events in mice. A capillary stalling event is a small, temporary blockage in a capillary and happens all the time when cells in the blood stick to the walls of the capillary. The client's research investigates how heightened incidence of capillary stalling is indicative of recovery from conditions such as a stroke. The client's study involved data collected from 8 mice, each measured 5 times to track capillary stalling events: once pre-stroke and then at various intervals post-stroke (2 days, 1 week, 2 weeks, and 4 weeks).

The primary goal of our analysis was to evaluate the client's assumptions while conducting the analysis and to help statistically demonstrate differences between pre-stroke and post-stroke survival analysis of capillary stalling events that our client had conducted.

Data and Methods

Several capillaries of interest were identified per mouse. These were the capillaries which stalled and un-stalled during the ten-minute measurement period. These capillaries were not the same across measurements. It is important to note that capillaries which did not stall at all and capillaries which might have been stalled for the entire ten minutes were not included. For each capillary, binary data time series were obtained over a roughly 10-minute observation period. The ten-minute period was split into frames, with seven seconds between frames. The capillaries were marked as stalled (1) or not stalled (0) for each of these frames. These series indicated the occurrence of stalling events within the capillaries.

To address the client's objectives, our analysis included the following key steps:

1) Confirming Assumptions of the Log Rank Test: Prior to applying any statistical tests, we verified the assumptions of the log rank test to ensure its suitability to the clients data. This step was critical to evaluating the approach used for comparing survival curves between pre-stroke and post-stroke measurements.

2) Evaluating the need for, and suitability of, survival analysis in the first place: It's not readily obvious that survival analysis techniques are the best way to capture information about the differences in stalling dynamics across pre and post stroke measurements. We discuss this further below.

3) Providing alternatives to Kaplan-Meier tests and cumulative incidence curves: We looked at creating alternative aggregates and using clustering methods to see if we might find/learn anything new from them.

Results and Conclusions

The assumptions of a log-rank test held for the analysis our client did and served as a good method of distinction between the Kaplan-Meier curves he created.

An examination of the data and the analysis conducted by our client reveals that a survival analysis approach, by examining only the first time a capillary stalls, loses a lot of information about stalling dynamics. Furthermore, choosing a different ten-minute period might easily lead to a different shape to the cumulative incidence curves and the Kaplan Meier curves (shape here refers to the rate of ascent/descent and not the overall plateauing shape which will always be typical of these kinds of curves).

We therefore concluded that instead of survival analysis there might be other aggregates more appropriate to capturing the dynamics of stalling. We looked first at simple cumulative curves, where we add up the number of stalling events over time. This suffers from a number of problems, such as the fact that choosing the granularity of these curves (per capillary or per measurement window) seems like an arbitrary choice. The major issue with this however, was that it didn't provide any information that our client might use to further his analysis.

A look at the data suggested that we might try clustering the capillaries and also take a look at stalling lengths and intervals between stalls as being more representative of the stalling dynamics. All the same, these results yielded little useful information. The clustering would have been helpful provided there was information about other characteristics these capillaries had (proximity to stroke site or major blood vessels etc.). In the absence of this information and in the same capillaries being selected for each measurement, the clustering is of limited value.

While stalling length and interval between stalls differed between pre and post stroke measurements, it doesn't really provide new information or information that the client was necessarily interested in for this study. All the same, he expressed interest in having those metrics and we provided him with a function that would create those metrics from matrices of his data.

Proof of Concept: Using Large Language Models to Accurately Summarize Changes in Accounting Standards and Contracts

Introduction

Motivated by the extreme time cost of summarizing the frequent changes to Accounting standards and to various contracts which they use or to which they are beholden, Fidelity's Data Science COE were interested in using Large Language Models to streamline efficient that process. We were tasked with creating a pipeline that, starting with the documents or with the regular changes posted on the FASB (Financial Accounting Standards Board) website, delivered a summary generated by a Large Language model that was suitably accurate, coherent, and useful. This involved (a) identifying the best way to feed a lot of information to an LLM in a way that both reflected the changes being made and, to some extent, mimicked the basic logic a human would use in solving the same problem; and (b) identifying how to generate faith in the LLM generated summary – how can we establish that what it outputs is accurate and reliable?

Data and Methods

With FASB standards, the changes are codified as struck-through text for removals and underlined text for addition while text with no markups constitute text that hasn't been changed. We parsed these documents into HTML, used HTML tags to identify the changed sections and generated 'before' and 'after' documents. For contracts, because we couldn't access Fidelity's own contracts for confidentiality reasons, we sourced exemplar Advisory Contracts and created versions with changes that our partners at Fidelity told us were the sorts of changes typically seen in the contracts they were interested in.

Once we had 'before' and 'after' documents for each of our use cases, we focused on processing the documents in a way consistent with satisfying the objectives (a) and (b) outlined above. First, this involved chunking. We wrote code to assign a score to text based on font size, bolding and underlining, and based on scores, divided spans of text into headings, subheadings, and text. We further chunked the text by paragraphs, combining very small paragraphs or breaking up long ones. This way, the LLMs' token limit would not be exceeded, and the models would have an easier time identifying changes to small chunks of text instead of pages-long documents.

After creating dataframes with the 'old' and 'new' content in different columns, we used measures of edit distance to identify changes. This included Levenshtein distance, longest common subsequence and a simple comparison of string length. These changed chunks were fed into GPT 3.5 using the Open AI API. The list of summarized changes was then fed to the model to condense into one summary. These summaries were compared with 'ground truth' summaries which in the case of FASB documents were given to us by Fidelity's Accounting Department and in the case of contracts was generated by us.

We used G-Eval to evaluate the LLMs summaries against the human generated summaries on the basis of relevance, coherence and consistency. We also used ROUGE frameworks, which directly compare the two and assign an f-score, recall and precision. Since we used edit distance measures to optimize our prompting, we used a Mann-Whitney U test to establish that the distribution of rouge scores were not different for each prompting group. This told us that our selective prompting was actually effective in what it was trying to achieve.

Results and Conclusion

While we were able to deliver the proof of concept and use LLMs to summarize changes to the documents we were interested in, the most important findings concerned the level of architecture we had to build around the models to both create the summaries and establish faith in their accuracy and reliability. This included optimizing prompts, combining scoring metrics with statistical tests to establish differences in performance, as we did with ROUGE scores, and chunking and parsing documents so that LLMs can effectively interpret the information we give to them.