**Milestone 3 - Extension**

**Grace, Natalie, Mikaela, Yash**

For our Milestone 3 extension, we decided to implement a form of concept bottleneck models, which hopefully will help create a more interpretable model than just using a single GPT model. Concept bottleneck models have been especially popular for image classification. These models begin by coming up with a set of concepts, either created by a human or by a model (more advanced work uses GPT), collects labels for the presence/absence of these concepts to build a supervised model, and combine concept scores. It is one of the most high performing, interpretable models, however it still tends to perform worse than its black-box counterparts.

*Our Implementation:*

For this milestone, we attempted to implement a concept bottleneck model, which was explained in “Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification.” The idea is that we generate features from our text data and use these features to run a simple machine learning model to predict our target (a list of professors). In feature\_generation.ipynb, we use a gpt-3.5 model combined with few-shot prompting to generate features. We used this list to come up with 93 features, which ranged from being general to very niche. This was by far the most time-consuming part of the project, since from here, we created a manual dataset in which we assigned feature values for every professor on our list.

Next, in feature\_transformation\_for\_test.ipynb, we created a model using gpt-4 and some prompting to transform our dev set to return 0 or 1 for all of our features. This leaves us with a matrix in which each row corresponds to a question and each column represents a feature. From here, we were able to use a traditional machine learning model to predict a professor or list of professors for a given question. We used Random Forest, SVC, and KNN and compared the results for each in the *Results* section.

In score.py, we used the same score technique as we did in Milestone 2 but changed the prompt slightly to account for the similarity between professors that we found in grouping\_professors.ipynb, which uses K-means to group professors into six different groups.

*Results*:

Concept Bottleneck Model + Random Forest (y = one professor): .456

Concept Bottleneck Model + SVC (y = list of four professors): .598

Concept Bottleneck Model + KNN (y = list of four professors): .608

As you can see, our model that uses a KNN model performs best at .608. In many answers we received a 4, meaning that in the list of professors returned by the chatbot, we didn’t return the one we were expecting, but many of the professors in the list fall into the same group as the professor we were expecting. Our chatbot answers appear high quality as well.

Future Work:

Overall, this was a very interesting approach to creating a more informed model. We think there is a lot of room for improvement in adjusting our prompts and features to obtain better results. For next steps, we think creating a pipeline that includes our Milestone 2 model and this one will give us optimal results.

*Sources:*

Yang, Yue, et al. “Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification.” *ArXiv.org*, 25 Apr. 2023, arxiv.org/abs/2211.11158.