Assessment 2 Reflection

Tom Blain

For the project we decided we would like to generate our own dataset to work towards creating a model we would all be interested in using. There was limited literature on the task we chose, however, one article [https://link.springer.com/chapter/10.1007/978-3-030-51156-2_125] seemed very aligned to our approach and used Latent Dirichlet allocation topic extraction on the movie plots, which made our task promising.

The first step was to create the dataset for the project. I began by researching how we could obtain the necessary data through an API. The official IMDb API was locked behind an AWS paywall, however, https://omdbapi.com/ was free and openly available and could provide us with all the data we needed. We needed to give it a film title and it would return data, hence we would need to generate a list of film titles to use with the API. We couldn't find a dataset of movie titles online so decided to use a web scraping approach and found that there were fairly complete wikipedia articles which documented a large amount of movies by country and by year, which would be ideal for the dataset.

After generating the dataset, our first task was to remove some of the films which the API had not called correctly - issues with calling the wrong films e.g. API call for 'Gone' could return 'gone with the wind' or 'gone girl' or 'Gone' (2012), or issues with the web crawler which gave incorrect titles to the API. We next needed to decide on an approach towards the actors column. For each film we had about 3-5 of the main cast, and this seeemed like it could be a very good predictor for the overall rating; we all know some terrible actors. We got together as a team to brainstorm this, thinking about dimensionality reduction methods. Eventually, we decided that only actors who has been in more than a few films would actually be useful as predictors, and the rest might just add noise to the dataset. Setting the minimum amount of films that an actor has to appear in to 5 massively reduced the amount of actors we had and it became a valid option to One hot encode the remaining actors.

The next major task in the project was to find an approach with the text data, such as 'Title' and 'Plot'. Our previous reading has mentioned LDA topic modelling as effective for this task. I wanted to also look into word embeddings such as word2vec or some pretrained transformer models. Initially we researched how to preprocess the text effectively for each of these models, attempting to only keep in the important meaningful words. We can then tokenize the sentances to prepare them to be fed into the models. Daniel wanted to look into synonyms, how two words can mean very similar things. We were unsure as to if this would have an effect on our final model.

LDA managed to produce a very promising looking vector for each movie plot which we

would hope would add additional information to a feature like the genre. The transformer approach seemed like much more of a brute force with a pretrained tool designed for any task you throw at it. Embedding whole movie plots however added a lot more complexity to the model, which is often not a good thing.

With 3 different approached to the text data, we then ran these through a boosting model to evaluate the performance of our approaches. We chose to implement an XGBoost model since it has been a high performance model in our previous data science experience. For our rating prediction task we wanted the model to have the ability to handle complex data as well as being fast and efficient. For our loss function we choose to use MeanAbsoluteError(MAE). This is a standard loss function for this task and does not penalise outlier data as strongly as MeanSquaredError(MSE). This provides more interpretability to our results. We needed to create a baseline to give us an idea of how good our performance is, especially since not many people have done this project before. I chose to make two baselines, one which would simply predict the mean rating of the training set (around 6.1) for each target, and another which would be a boosting model without any text processing, just the originally gathered data with one hot encoded actors and 'Plot' and 'Title' removed.

We were not expecting for the text models to perform worse than our boosting model baseline, since to us, the plot adds a lot of information about a film and is often the reason we choose to watch a film on something like netflix where you cant see a rating. We would really need to do a lot more research into why none of our text approaches worked well, but this project was over a short time period so we didn't get a chance to try again.

My overall contribution to the project was the dataset creation (web crawler, imdb api), initial data processing, and actor processing. I then created the huggingface transformer text embedding model, the xgboost models for all of our models, and the evaluation with daniel.

The team was good in this project however we definitely did not get as much opportunity to brainstorm and problem solve together as the previous project. It was less effective for individuals to be working on their own areas with less collaboration, but the deadline of this project was shorter than the previous so it was a difficult thing to balance.

The project has been a good introduction to some NLP methods which I hope to explore more soon. It would be good to begin a personal project where I can try and find a way to make these film plots effective predictors. I'm also interested in generative AI and feel this data would be good for trying to generate new film plots with GANs or VAEs. My main learning was that you absolutely need a well thought out and well tailored approach to using text data.