Laidlaw Scholarship - overview

Summer 2019

- FutureLearn data analysis, ML, graph creation – with Ahmed Alamari
- FutureLearn platform fully webscraped
- Kaggle Eye dataset ML competition – ranked top 300 globally
- NLP Bert – multi GPU training

Summer 2020

- NLP Bert on twitter dataset for SMM4H20 – with Tahir and Jialin
- Further NLP dataset analysis – guided by Tahir and Jialin
- Japan e-book dataset analysis, graph creation – with Jingyun Wang

FutureLearn data analysis

For this project, I worked alongside a PhD student to develop insights into student behaviour on MOOCs. We started by developing a pipeline for ETL to restructure the dataset. After we created a structed dataset, we started analysing the trends, asking and answering questions we had. We split students into two categories, completers and dropouts. Completers are students who complete more than 80% of the course, the rest are dropouts. Dropouts are the biggest problem in MOOCs, hence we strive to identify these students as early as possible.

We ran 20-30 ML experiments on predicting dropouts through various strategies. I created a jupyter notebook with approximately 15 algorithms including AdaBoost, XGBoost, Random Forests and more. We focused heavily on feature engineering and creating additional columns of data for each student with some meaning to it. We reached 80% accuracy on predicting dropout from the first week of student activities.

We decided to create some graphs to visually analyse the student behaviour patterns. I wrote the automated graph generation code using Python. We spent about a month tweaking the complexity of the output graph, with advice from many members from the CS department.

The output of this project was a paper titled, Is MOOC Learning Different for Dropouts? A Visually-Driven, Multi-granularity Explanatory ML Approach. This paper was accepted through peer review to ITS 2020.

FutureLearn web scraping

I wrote some web scraping code to scrape all of the content from FutureLearn for all courses, including video files, transcripts, discussions and articles. This data is on OneDrive and has been used by multiple PhD students to add additional pieces of data and understand the structure of the courses easily. The data has been normalised in a sense to

third normal form and in a csv file format. The web scraping code is also on OneDrive, so it can be run in the future easily to account for new courses and content.

Kaggle Eye dataset competition

Millions of people suffer from diabetic retinopathy, the leading cause of blindness among working aged adults. Aravind Eye Hospital in India hopes to detect and prevent this disease among people living in rural areas where medical screening is difficult to conduct. The models that result from the Kaggle competition will improve the hospital's ability to identify potential patients.
Competiton url: https://www.kaggle.com/c/aptos2019-blindness-detection/overview

One of my main reasons for entering this competition is to utilise the resources given to me, specifically the RTX 2080ti. Also to expand upon my image classification model building skills.

For this competition, I tried many different models and training types, across PyTorch, TensorFlow and Keras. The best results from my initial testing came from using EfficientNet. After a lot of tuning, I achieved a weighted kappa score of 0.797 using EfficientNet B5 on the public leaderboard. The final results are calculated from the private testing leaderboard where my model achieved a weighted kappa score of 0.9105 and the best model achieved a 0.936. The best model was created by a Kaggle GrandMaster, just like the entire top 10.

I presented this project at the UCL Laidlaw conference as part of the 3 minute thesis competition, it was received well.

NLP Bert multi-GPU training

I was asked to advise on another PhD student's execution of training Bert. After a lot of reading and analysing the training of transformer models I advised to try training on multiple GPUs or a RTX 2080ti rather than a GTX 1080ti. I went on to adapt the code for multi-GPU training. However, it was towards the end of summer so I did not get a chance to stick around for its execution and to my knowledge it wasn't executed.

It did give me a lot of insight into NLP tokenisation and pre-processing steps as well as Bert and the hardware issues that arise from trying to train it.

NLP Bert – tweet classification – SMM4H2020

I worked alongside two PhD students on Task 1 of the social media mining for health applications shared tasks. The task was to classify tweets based on whether contained information on medication or not. The task was challenging as only 0.2% of the dataset has tweets that contained medication data. This high imbalance required many strategies working together to counter. We created a new workflow for pretraining, by using an intermediary pretraining stage with an external balanced dataset that we acquired from previous years of the same competition. The scoring metric was the F1 score of the positive class.

I wrote a major portion of the first draft of the paper as well as tweaking the phrasing and explanations in the final drafts. The paper was accepted through peer review to the conference, which has been delayed to December due to Covid-19.
We ranked in the top 4 for the competition out of approximately 10 groups, we need to wait for the conference to know the exact results in full. Our model performed better than the average of all 10 models.

Further NLP dataset analysis

Towards the final days of the SMM4H competition, we shared and discussed a lot of ideas specific of highly imbalanced datasets. This triggered a train of taught for me, on how we could artificially imbalance a dataset to understand if these claimed techniques actually worked and how imbalance actually affects algorithms. It was an attempt to really understand and create a benchmark that could in theory inspire more research into the area.

After sharing my thoughts, I discussed the ideas with two other PhD students who understood the potential of the research. They guided me to certain resources and ideas that added to the research idea.

I wrote some code to test some initial ideas using PyTorch Lightning and experimented using the twitter sentiment analysis dataset Sentiment140, which is heavily researched and is perfectly balanced. My initial experiments included splitting the datasets into different imbalances like 60/40, 70/30, 80/20, 90/10; also analysing these splits with both the postivie and negative classes as the imbalanced category.

The preliminary results are as follows:

Class b is denoted as positive class and class a is the negative class, classes a and b:
Experiment 1 - benchmark:
100a 100b pos=0.869 neg=0.87

Experiment 2 66/34 split bothways with all sets of data:
100a - 50b pos=0.908 neg=0.815
100a - other 50b pos=0.911 neg=0.818

50a - 100b pos=0.812 neg=0.91
Other 50a - 100b pos=0.821 neg=0.9

These scores show that independent of the which subsection of data is used for the training the results are very consistent. This result allows us to confidently create a higher imbalance and experiment further.

Japan e-book dataset analysis

Working alongside Professor Jingyun Wang from Kyushu University I created some basic data analysis scripts to understand the dataset. Due to restriction I was not able to view or obtain a copy of the dataset, I was working with only the schema of the dataset. This led to many technical issues like mismatches in column type, column names etc. These bugs would

normally only take 5 minutes to fix but due to the workflow and the distance it would take a day to fix a bug and get it tested to make sure the fix was working.

It was an interesting project due to the nature of the experiments involved and all of the data that had been logged by students across multiple courses. Similar, to the project completed on FutureLearn data we wanted to analyse student behaviour and patterns.

The first 10 scripts helped me understand the dataset better and come up with the concept of linear and non-linear students. Linear students are those that read the book in a linear fashion, whereas non-linear students search for a topic, go back and forth etc. We made the decision to draw some graphs on student progression to see if we could stop some interesting patterns and come to some visual conclusions.

Professor Wang's time at Kyushu University was coming to an end during the summer so instead of generating the graphs which was a slow process going back and forth we decided to generate some intermediary data which get through all the restriction on taking data outside of Japan.

This process was very interesting because of two reasons, the looming deadline and the lack of a powerful server. The initial algorithm estimated 20 hours to complete per course, we had 6 to get through. Step by step, I optimised then reoptimized to create a script which completed the same task in under 30 minutes. The input dataset was fairly large and pandas had trouble handling it in the limited ram so as the algorithm processed all of a student's activities the code dynamically deleted that student from the dataset. This mean the file size got smaller as the number of students processed increased. I sorted the students such that those with the most activities were processed first. The next bottleneck was the output file that was being created. I ended up creating a file per student with a list of all of their activities such that the file was never large and was no longer a bottleneck. I thoroughly enjoyed the process of optimising the intermediate data creation script, applying concepts learnt theoretically.

I have been working with this intermediary dataset locally to generate graphs, we are close to finishing this section of the project. I have a little bit more work to do on tweaking the visual aesthetics of the graphs and some of the logic in its creation.