Project 4 - Executive Summary

For this project, I had to scrape data from a job listings website and then make a model to predict whether a job would have a salary or low salary and then create another model to predict whether a job would be a data scientist or business analyst, based on the data that I scraped.

I scraped my data from au.indeed.com. I scraped job listings for three different roles (data scientist, data analyst, and business analyst) and the five main cities in Australia (Sydney, Melbourne, Brisbane, Perth, and Adelaide. The data that I scraped included the location, job title, company, salary, summary (short text summary), and description (long text summary/description).

Before I could create the model, I had to first clean and analyse the data. The salaries that I scraped were not all annual salaries. Many were hourly, daily, and weekly salaries. I scaled these non-annual salaries to an annual salary. Once I had scaled the salaries, I looked at the mean annual salary for each type (yearly, weekly, daily, and hourly). Daily and hourly salaries had the highest mean annual salary and I therefore concluded that these were likely contract jobs. Weekly salaries had by far the lowest mean annual salary, well below the national Australian average even for senior positions. Therefore, it is likely that these are part-time roles. When looking at the individual cities, Brisbane had the highest mean annual salary but not the highest median annual salary. Upon closer inspection, it was revealed that Brisbane has a higher proportion of the high-paying hourly and daily jobs than the other cities, which explains why it had the highest mean salary of all five cities.

For my first model, which predicted high vs low salary, I used the mean salary of the jobs that I scraped to form the boundary of high and low salaries. Because most of the jobs did not have a listed salary, I could only use jobs that did list a salary.

The models require numeric inputs. Apart from the salaries, the data that I scraped was text. I had to use a process known as Natural Language Processing to look for key words in the job titles, companies, job summaries, and job descriptions. I could then look at whether certain words appeared more in high salary jobs or low salary jobs and use these for the model. With the model requiring a numeric input, I engineered the features such that the value would be a 1 if the word was present or 0 if it was not present. Other features (or predictors) that I included were the locations/cities. The target that my model was trying to predict was: 1 for high salary; 0 for low salary.

The model performed reasonably well, with an accuracy score of 72%. In regards to incorrect classifications, the model was more likely to incorrectly classify a high salary job as a low salary job than incorrectly classify a low salary job as high salary job. This is a desired outcome because if you were to predict the salary of a job that someone was applying for, it would be better to predict and tell them incorrectly that they would get a low salary when they would end up with a high salary (which would make them happy) than tell them they would get a high salary when they end up receiving a low salary (which would disappoint them).

The features that had the most affect on the model were the words from the descriptions. The presence of certain words were strong predictors of high or low salaries. This shows that descriptions include technical roles such as data engineering and architecture, software like aws, and involvement in the business processes are more likely to predict high salary. Descriptions that include less technical roles such as administration, customer service, and assistant are more likely to predict low salary.

Once I had my model, I was able to use it to predict whether the jobs that did not list a salary had a high salary or low salary. To reiterate, the accuracy was just over 70%.

With part 1 of the project complete, I could move onto part 2. For part 2, I would be creating a model that would predict if a job is a data scientist or a business analyst. As with the first model, for part 2 I needed to use natural language processing to determine which words/phrases more

commonly appeared in the summaries/descriptions of data scientists than business analysts. Once I had done this and engineered the features based on these words, I ran the model.

The model got a score of around 87%, which seems good. However the baseline accuracy (i.e. what I measure the model's accuracy against) was 86%, so the model was not much of an improvement. The reason for the high baseline was that 86% of the jobs were business analysts, meaning you would get an accuracy score of 86% just by guessing business analyst for every job. Upon further inspection of the model, I noticed that the model was predicting very few data scientists and jobs that should have been data scientists were being classified as business analysts. The type of model that I used is known as a decision tree. With my model struggling to distinguish between data scientist and business analyst, I tried two other types of models, know as logistic regression and KNN. Both of these performed slightly worse than the original decision tree model. Even though the decision tree was the best performer, the overall performance was still not very good. Therefore, I conclude that the data I scraped is not sufficient for being able to predict whether a job is a data scientist or business analyst.