Project 4 executive summary

This project was split into three basic parts which is quite typical of data science projects. The first part was acquiring the data and making it suitable for use. Once the data was acquired, I needed to explore the data in depth to see what insights I may be able to gain to shape the approach for next step. Once I felt comfortable I knew enough about the data to build models with it my final step was to fit appropriate models and access them.

The data I chose to use for this project was from the website Indeed.com. The reason I went with this website was simply because I believed I would be able to gather enough data to carry out the project. The tools I used to acquire the data included python, Selenium and Beautifusoup. Python was used to run the modules needed to scrape the necessary data. The reason I used Selenium was due to the fact that most of the Indeed’s website is loaded using javascript. This meant that I needed to load an instance of the website before obtaining the html code as it would not load otherwise. In this way Selenium was used simply as a headless browser to imitate loading of the website on every query. Once I reached the necessary page I used Beautifulsoup to parse the html and convert it into text I would be able to use. I focused on three main job titles spanning all over Australia. These titles were Business Analyst, Data analyst and Data scientist. Using these main search terms I returned a total of 9000 jobs many of which did not fit the scope of my question but I did not address them until later on. For each job, I obtained the job title, the location of the job, the full description and the salary if it was given.

Once I was happy with the amount of jobs that I found, I moved onto the second part of my project, exploratory analysis. To begin, all of the data needed to be “cleaned”. This basically means I needed to make the data uniform in the type of information I was getting and any missing values needed to be checked. The cleaning was done through python and after removing all the non-relevant job titles and cleaning the data, I was left with approximately 1000jobs for me to do my analysis on. I decided to create a tableau dashboard to visualise my exploratory data analyst. I focused on information that would be useful if I was looking for a job such as location, title and salary level. Most of the jobs I found interestingly were on the East Coast of Australia. Using the dashboard I can explore down any level of granularity I like.

I also explored the descriptions themselves, grouping them into their appropriate job titles, I could seem some interesting patterns arise. For example, I could see that data and business analyst roles seem to have more words within their description looking for strong communication skills and customer interaction. Data scientist roles focused more on the tools such as machine learning, SQL, working with big data and python. Interestingly, all the job roles no matter what seniority level, there was a large focus on experience as a whole. This however may need more investigation as it is quite plausible that “experience” could very well just be over representation due to it being a common header within the descriptions.

After I felt I had a good grasp on the data I had and I was done drawing any inferences on face value, I focused on building the appropriate models to answer the questions within the project. For question 1 the goal is to predict the salary of a job based off its description. To answer this required a lot of pre-processing. The reason for this is because of my 1000 jobs, only 300 had salaries listed. I decided to use these 300 salaries to impute what the missing salaries would be instead of dropping jobs which had no salary. The reason I chose to do this was I wanted to answer this question using regression techniques because I planned to tackle question 2 as a classification problem. What this means is I am attempting to predict a continuous number for the salaries. I imputed the missing salaries by grouping the jobs appropriately into the three job titles and three different seniority levels, junior, unspecified and senior. I then took the median values for each of the categories and used them to fill any in that were missing.

The first model I chose to fit was a linear regression model, specifically I used Lasso Regression. The reason I chose to use this first over any other was simply due to the large amount of data I had and I wanted to reduce the amount of variables I would need to predict the salary with. For the text analytics I decided to use CountVectoriser and TFIDFVectoriser to compare between the two and I also implemented some basic stopwords. CountVectoriser simply converts the descriptions into numbers and does a count for each of the words the idea here is that the more common a word within a description the more important it is. TFIDFVectoriser does something similar but once I have the frequency, it returns the inverse document frequency instead. It takes into account not only the number of times a word appears within a descriptions but also compares it across all the descriptions. Stopwords were used because these words may have a high frequency within a description but won’t actually give any information such as “for” and “in”. This first model did not perform as well as I hoped, returning an R2 score of 0.15 which means only 0.15 of the variance within the salary can be explained by the description and the job location.

I then fit a random forest regression model which is an example of an ensemble model (a model made of several models all working together). This model returned an R2 of 0.22 which was an improvement but still had a root mean squared error of 27916 which means that on average my predictions were off by $27, 916. The major limitation of the models I fit for this question were due to the nature of that data I had, since I was predicting salaries where most the salaries were not provided, by imputing the medians I effectively created a classification problem and it was no longer a regression problem. To show the effect of this I fit one more model, but this time used classification. The model fit was a random forest classifier and I used the CountVectoriser as it performed better with my other models also. The baseline was 0.64 and I achieved an F1 score of 80 which is much better a predictor than my first models.

For the second question, I fit 2 models again and attempted to predict the three different titles within my dataset, Data analyst, Business analyst and Data scientist. My best predictive model was XGboost, an ensemble model using decision trees but I gained the most insights from a logistic regression with an L1 penalty. With the logistic regression it was clear that it was predicting data and business analysts based heavily on commination and storytelling skills. Data science roles were distinguished more by technical skills such as machine learning, big data and strong fundamental mathematics. These findings are quite in line with what I expected before analysis but also were expected from the EDA findings.