Machine Learning Engineer Challenge

Hi there, welcome to my submission for the Machine Learning Engineer Challenge! I hope you enjoy it as much as I did. The thing I really enjoyed about this challenge is that it was vague, and left a lot of room for creativity. Hopefully this document shows an analysis of the downloaded datasets, summarizes the design choices I made, and some future directions for more advanced models.

1 Description of the Dataset

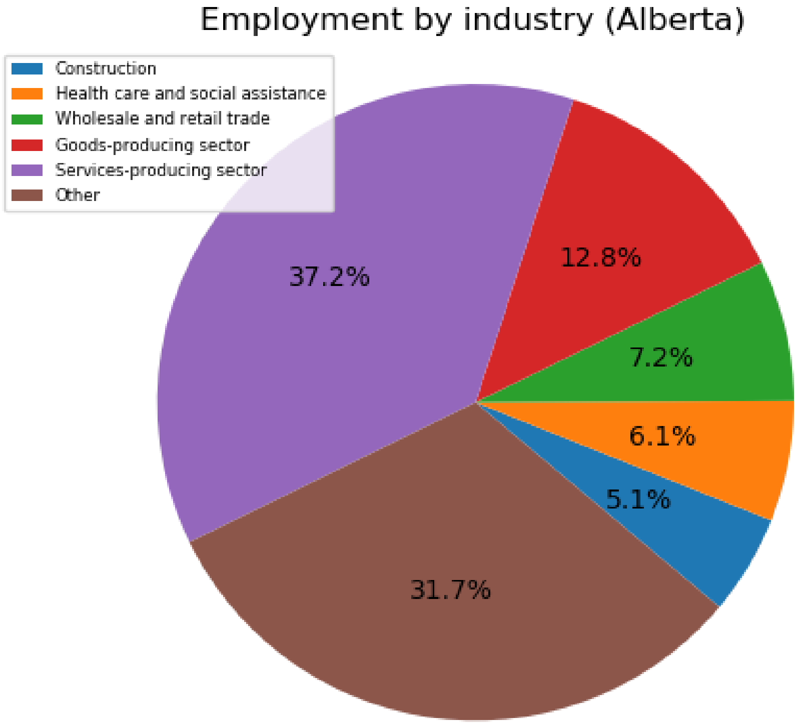
The most important component of any machine learning or AI project is the data! So, that was the first thing I looked at.

I downloaded two sets of data from the linked Statistics Canada website. This included employment data for each province by age group and sex, and by industry. Also, the provided job dataset (*job_skills.csv*) was downloaded. These datasets were preprocessed (split by province, and some other minor tweaks), and the final datasets were stored as csv files. You can find all of the files in the 'Employment Data' folder on my Github repository.

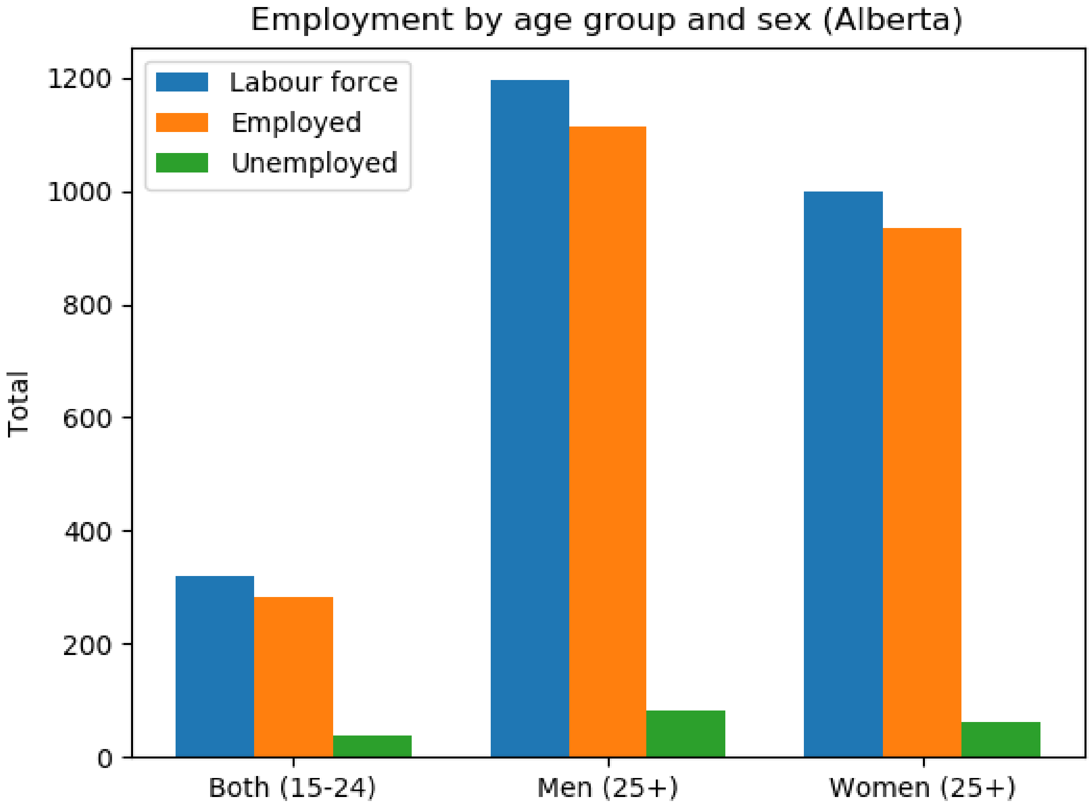
2 Data Analysis

Once the datasets were downloaded and preprocessed, I wanted to get a better picture of what this data represented. To do that, I constructed several charts (pie and bar charts) for each dataset, per province. I wrote the Python script, which you can find as *visualize_data.py* in my Github repository, to construct these graphs. I found some interest results from the data.

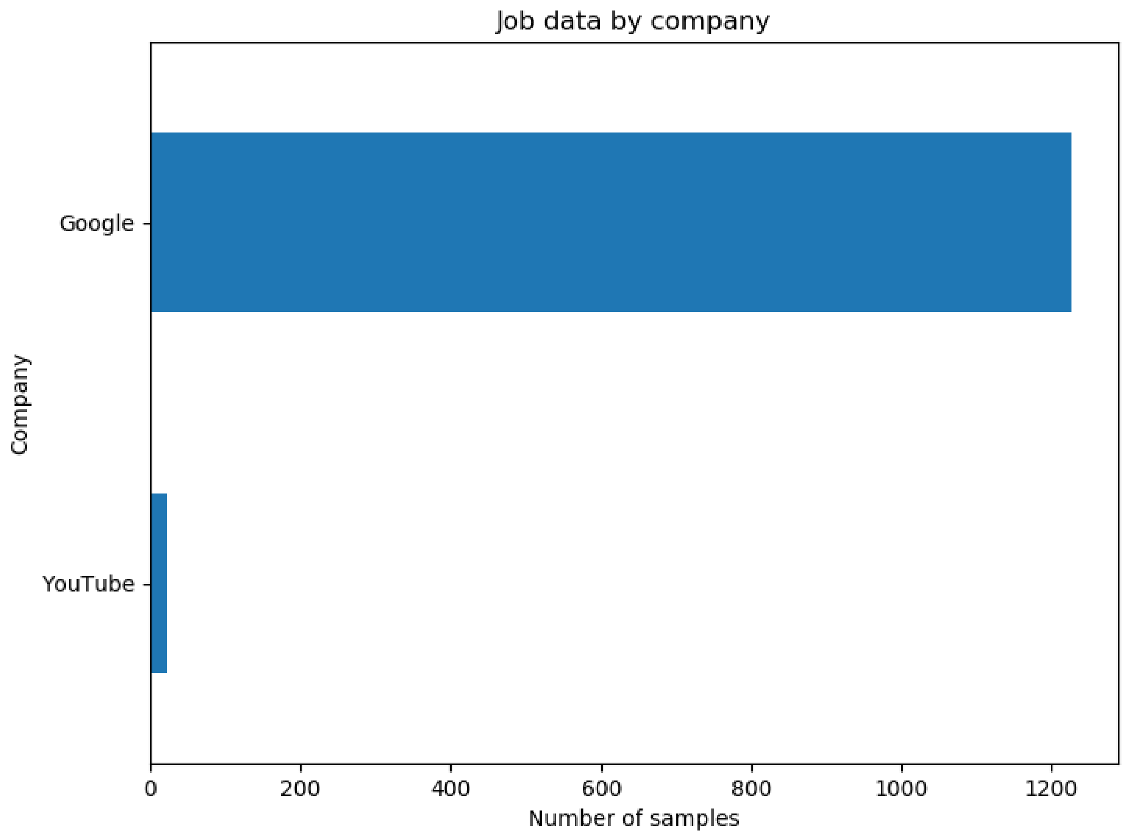
In the figure below, I have attached a pie graph I made for the province of Alberta. From this picture, we can see what the most common industries are in Alberta. This can help identify bias in any predictions, as training examples typically should be balanced. So, for instance, a model for predicting industry in Alberta would be heavily skewed towards predicting the "Services-producing sector", since it is the most common label. It is always important to note biases like this that may emerge in your model. I constructed these pie charts for each province, and they are kept in the 'Plots' folder in the repository.



To analyze the construction of the workforce, I then turned towards the employment data by age group and sex. Again, for each province, I created bar charts that looked at different age groups and sexes. I found some more interesting results, specifically by comparing the charts for each province. As you might expect, the labour force varies significantly for different provinces. As a result, any model built will obviously be biased towards more populated provinces, as it is more representative of the data. Also, another important note is variance in the unemployment-to-labour force ratio. By comparing provinces, you can see how unemployment fluctuates, and this is another interesting finding from the data. An example of the constructed bar charts is shown below for Alberta, and all of the constructed plots are stored in the ‘Plots’ folder in the repository.



Finally, the last analysis I performed was on the provided job_skills dataset. As shown in the figure below, a majority of the examples provided comes from Google, while some are from YouTube. For this reason, I decided to stay away from predicting the company, since the dataset is so heavily skewed towards Google.



3 Predicting Category from Responsibilities

Once I dove into the data and understood its construction, I decided to build a basic predictive model. In my data analysis, I found that the 'Category' section was pretty balanced, and the 'Responsibilities' column seemed pretty interesting.

I decided to use a basic Natural Language Processing (NLP) method of analyzing text known as 'tf-idf'. Term frequency inverse document frequency (tf-idf) is a very basic tool in NLP that treats an inputted string as a 'bag of words', and does not consider the context or semantic meaning of the words used. I chose this method because it is easy to implement, and is often a good baseline model that can be improved. I showed this result in my undergraduate thesis this year, where I built a deep learning model to predict legal citations using text.

Using tf-idf, I wrote a basic script that found the most common unigrams and bigrams for each job category in the dataset. So with these bigrams, in theory, you could make predictions about the category of the job, simply by knowing the responsibilities. The script is saved as *predictions.py*, and the results for each category are stored in *results.txt* in the repository.

It is important to note some limitations of the model. As shown in research, bigrams are typically more powerful than unigrams. However, the dataset is so small that it probably doesn't make a significant difference. So, one limitation is the size of the dataset. Another limitation, as mentioned above, is the ability of tf-idf. Much more information can be gained beyond tf-idf, and future models could use pre-trained word embeddings that already have built-in information. Finally, the model built is static, so future models would in theory perform better if they learned from the data (i.e. deep learning).