**Weekly Update**

**Oct 26- Oct 30**

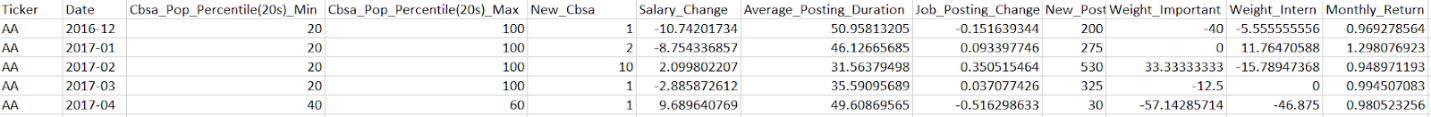
Summary of work done:

Each one of us has completed engineering his or her part of X (predictor) variables. We organized an internal meeting and shared each person’s procedure and methods he/she used in transforming the data to obtain these predictor variables. We also agreed upon certain assumptions that can help us join our individual works into one cohesive dataset for future training of the model.

The main methods we used and some corresponding assumptions are summarized below:

1. Filter out all the job posting data that were posted before 01/31/20 using SQL commands on the server (This cutoff is chosen to get rid of the COVID-19 impact on the job market.)
2. Based on our assumption, we treated each company’s data in each month as a single observation. More specifically, we calculated all job-related or stock-related aggregated values in the monthly interval for each company ticker.
3. We each wrote separate python code to turn the raw data into potential features that we can use in the modeling stage. The features (predictors/response) (for each company and month) are summarized below:
   1. **Job posting change** (in percentage): built a dictionary for each ticker with month as key and monthly count of active job postings as value. Then calculated the change in percentage for each month.
   2. **Average cumulative duration days**: constructed a dictionary for each job posting which tracks its cumulative duration days. Then used another list to store all cumulative duration days for each month and calculate the average.
   3. **New job postings**: used ‘GROUP BY’ statement in SQL to obtain number of new job postings in each ‘observation’
   4. **Average salary change** (in percentage): first calculated the average salary grouped by month and company, then used window functions in SQL to calculate salary change
   5. **Stock price change** (in percentage): scraped the data frame AlphaVantage (a web API), then wrote a python code to clean the scraped data and compute close price and close price change in each month.
   6. **Monthly return**: calculated through dividing the closing price on the last day of month by the closing price on the last day of the previous month.
   7. Dropped zip code level information and only used CBSA level information. Grouped records based on ticker and month date, so that each record has minimum and maximum population percentiles of the cities that a company is hiring for that period, and it also has a number of postings to new CBSAs for the company.
   8. **“Important” job postings change** (in percentage): allow us to track the open positions a company is hiring for in regards to roles such as VP, supervisors, directors, and chief officers, as well as those requiring more than 10 years of experience. Selected through regular text expressions of desired titles and tags for jobs of interest.
   9. **“Entry” job postings change** (in percentage): allow us to track the open positions a company is hiring for in regards to roles such as interns, entry level positions, as well as those where training is needed or training is provided. Selected through regular text expressions of desired titles and tags for jobs of interest.

After we finished our individual parts, we discussed how we are going to join our separate tables into one during our meeting. We decided to join our tables and predictor variables based on the company ‘ticker’ column and the ‘month’ column, we also agreed on the format and data types of the two columns to reduce the possibility of errors.



Plans for this week (11/1 - 11/7):

1. Combine our individual tables into one table
2. After joining the tables, we plan to do an initial screening of the data to look for abnormalities and outliers, as well as general exploratory data analysis of the final dataset.
3. Then we will split the data into training and testing data (potentially split based on Date) and also start testing sevel models on the training set
   1. Model: use the x predictors engineered using the data Greenwich provided as well as the stock return data that we scraped from the web API as our Y variable.

Model Planning

* SVM model: predict the trend of the stock
* Regression model: predict the percentage change in the stock price/monthly return
  + Random forest regression
  + Multiple linear regression

Model Tuning

* Use **feature reduction** techniques including lasso regression, best subset selection to reduce multicollinearity and select significant predictor variables
* **Cross validation**: train and test split
* **Tuning hyper parameters**

Model Evaluation

* **Criteria for “accurate”:** If the actual increase in stock price is larger than or equal to the predicted increase in stock price or the absolute value of actual decrease in stock price is larger than or equal to the absolute value of predicted decrease in stock price, then we can call it an accurate prediction.
* Discrepancy between the predicted stock price the actual amount (in percentage)