ECE 5984 Project: Glassdoor Job Title Prediction based on the Job Description

Team members

Eugene Jung Jo

Chang Kyu Kim

What have you done so far? / What is left to do?

The scope of this project is to create a data pipeline that takes the raw web-scraped job posting data from Glassdoor, and build a text classification model that can predict the job title. This dataset provides several features such as salary range, company name, size, industry and many others, but we are mainly using the job title as the label data and the raw job description as the feature data to build the classifier. We have chosen the Glassdoor job posting data as we are interested in Natural Language Processing and this dataset is realistic data we would work with when building a NLP data pipeline in the future. Before building the data pipeline with Airflow, we have been working with the data in local development. We have worked through the Exploratory Data Analysis on the job title and job descriptions and performed the initial data cleaning and preprocessing. With the initial transformed data, we have run through 11 different ML models and compared the mean scores and accuracy.

We want to improve the model accuracy by further cleaning the training data and once we are satisfied with the model performance, we will implement the whole process in Airflow and write the final report with the visualization of findings.

What pipeline, project, and dataset are you using?

We chose the batch data ingestion and ML model training pipeline as this is the most common data pipeline we see in the field. As mentioned previously, we are both interested in NLP projects and this job title prediction with Glassdoor dataset (Data Science Job Posting on Glassdoor) was the best fit to our interest. Our data pipeline will start by downloading the raw job posting data from Kaggle, cleaning and preprocessing the job title and job description field, and training 11 different ML models and recording the classifier performance.

Does your data need data cleaning or preprocessing? If so, what?

- Text Normalization
    - Case Normalization
    - Punctuation Removal
    - Stop word removal
    - Lemmatization

For the job title, we first grouped the job titles to see the distribution and look for the outliers. The predominant job title was Data Scientist, 337 from a total of 672 entries and Data Engineer with 26 entries. The main two variation we needed to tackle for better grouping was the seniority (Senior, Sr., Jr., Principal, etc.) and specific industry included in the title (e.g. Data Scientist - Algorithms).

By reviewing the outlier cases, we have identified following job title category:

- data scientist
- senior data scientist
- data engineer
- senior data engineer
- data analyst
- senior data analyst
- machine learning engineer
- machine learning scientist
- data science manager

We have decided to group the job titles with following keywords into a single job title called data science manager:

- manager
- management
- director
- vp
- president

We also decided to group the following keywords under senior title for each job category:

- senior
- sr
- experienced

- staff
- lead
- principal

After cleaning the job title, the total of 172 unique job titles was reduced down to 69, and the number of job descriptions matching the 9 groups is 583 from the total of 672 (86.7%). The following is the statistics about the normalized job titles.

```
count                   672
unique                   69
top          data scientist
freq                    390
```

For the job description, we tokenized the text using `nltk` library's `word_tokenize` method, removed english stopwords and lemmatized the remaining word tokens. On top of the `nltk` 's built-in english stopwords, we used additional stopwords from https://www.ranks.nl/stopwords and https://www.kaggle.com/datasets/rowhitswami/stopwords to increase effectiveness of the categorization. It generated average of 316.74 tokens. The following is the statistics about the numbers of the tokens per entry.

```
mean    316.744048
std     146.196395
min       8.000000
25%     225.000000
50%     306.000000
75%     390.000000
max     998.000000
```

Will you perform Exploratory Data Analysis? Which methods are you going to use?

We have taken the raw data and converted it into a Pandas DataFrame for EDA. We have looked for the data anomaly such as duplicate, missing data, and inspected the job title and job description data. We did not need to explore the other numerical features such as salary or the categorical data like industry and company name as our project is focused on job title and description.

What information about data provenance have you listed? Answer the characteristic data provenance questions addressed in Module 5

The raw data is available from Kaggle and the author scrapped the data science related job postings from glassdoor's website. The author provides a raw and cleaned version of the data and we are using the raw version. We have downloaded the raw data and only keep the two fields that are relevant for our model: Job Title and Job Description. This data was created about 3 years ago, and it has not been updated since. The original author performs the data cleaning process and provides a raw and cleaned version of the data. We believe this data is trustworthy as we can validate the entries by looking up the company names to make sure it is an actual company and comparing with the web archive of the job posting data if it is available. The data is also likely authentic and we can validate by comparing the data with the web archive version.

Are there any untested assumptions or other reasons that would prevent you from completing your project?

As the data is 3 years old, more recent job descriptions might be different from the training dataset as the industry has evolved and the demand for the roles have changed over time. While we do not think this will prevent us from completing the project, the model performance might be insufficient with the latest job descriptions and we may need to acquire more up-to-date dataset for training.

What are your next steps?

We want to iterate on additional data cleaning and preprocessing to improve the model performance and also work on visualizing the model performance for the report. Once we are satisfied with the tuning, we will implement it with Airflow to create the final data pipeline.