# IS 567 - Text Mining Final Project Report

## Recommendation System

**Team members:**

1. Anish Shetty (anishhs2)
2. Asmita Dabholkar (avd6)
3. Saish Desai (sbdesai2)

**Introduction:**

Recommendation Systems, in a simple sense, are algorithms that are aimed at suggesting relevant items to users. The recommendations can vary from use case to use case, and these algorithms may also be designed in a different way as per their use cases. Examples could be cases like product recommendations, movie recommendation etc. Recommendation Systems can be pivotal in various industries as not only they would direct relevant choices to people, but they would also be generating a huge amount of income for companies which deploy such systems to direct their customers towards relevant products. Generally, what we came across in our research is that there are two kinds of recommendation systems, i.e Collaborative Recommendation Systems and Content based Recommendation Systems. Collaborative Methods solely rely on past interactions recorded between users and items to derive new recommendations. Content based recommendation systems on the other hand do not only rely on the user-item interaction, but also rely on the additional information pertaining to these users and items, and essentially work very well when compared to collaborative methods where a new user will have a difficulty in the start relating to recommendations. As graduate students, one of our daily hardships would be the process of job and internship applications. We realized that for a given dataset containing information about job postings, applicant profile and skillset and relevant search history for an applicant, we would be able to derive a recommendation system based on such textual data directing relevant job postings for a particular applicant's skillset, profile, experience, and search history. Such a system could also be designed from a recruiter's perspective where for a given posting, this system would be able to recommend relevant applicants based on their features as well.

**Data:**

(Source: https://www.kaggle.com/datasets/kandij/job-recommendation-datasets)

We used a job recommendation dataset from Kaggle for the project. The complete data consists of four individual files –

a) Experience – Includes title, location, duration, and description of the jobs that the candidate has worked.
b) Job Views – Includes the job ID, title, location, duration, and description of the jobs that the candidate has viewed.
c) Positions of Interest – Includes the applicant ID and job title of the positions that the candidate is interested in.
d) Combined Jobs Final – Includes the title, location, and description of the jobs that are currently open.

**Methods:**

We applied the standard cleaning techniques to our data. For the Combined Jobs dataset, we selected the relevant columns such as job ID, title, position, company, city, employment type, and job description. Except job ID, all the columns contain textual data. We checked for missing values in the data which were filled using imputation wherever applicable. The null values in the 'City' column were imputed with the headquarters of the respective company. The null values in the 'Employment.Type' were imputed with 'Full-Time/Part-Time'. The textual data was case-folded and converted to lower case and non-alphabetical values were removed. The title, position, company, city, employment type and description were combined to form a long string for each job ID.

The user data was divided into three files – experience, job views and positions of interest. These files were cleaned using similar techniques which were used to clean the jobs data. We combined all three files into one dataframe so we could get the relevant user information in one text corpus.

While converting the text data into vectors for further analysis, we had to decide which transformation would best represent the text into numerical format.

Count Vectorizer: It converts text into numerical data based on the frequency of words in the text. The more frequent words will have higher numerical values. As a result, the most frequent words will have a higher statistical significance.

TF-IDF: It stands for Term Frequency-Inverse Document Frequency. Along with word frequency, it also assigns importance to the uniqueness of the words within a document. It is based on the logic that the words localized in a particular document from the corpus will be given higher weightage than the words prevalent across the entire corpus. The words with higher scores have higher importance and the words with lower scores have lower importance.

We decided to use TF-IDF to transform our jobs and user text corpus into numerical vectors.

We used cosine similarity to measure the similarities between the vectors of user data and jobs data. Similarity is determined using the cosine of the angle between two given word vectors. If the angle is small, the cosine value will be closer to 1 and the similarity would be higher. If the angle is large, the cosine value will be close to -1 and the vectors would be dissimilar.

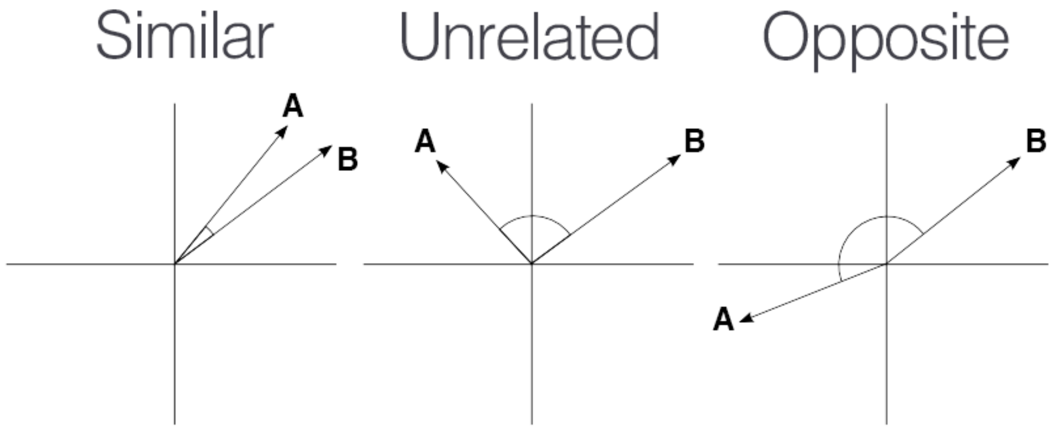


Fig. 1 Cosine Similarity

For job recommendation, we took the applicant ID of the user as input. We measured the cosine similarity between the given user record and job data. The job records with the top ten highest scores were displayed as output.

```
Enter the applicant id: 3
```

| | Applicant.ID | text | select_pos_com_city | Position.Of.Interest | Position.Name |
|---|---|---|---|---|---|
| 1 | 3 | marketing intern server prep cook | | | marketing intern server prep cook |

Fig. 2 User input for job recommendation

| | ApplicantID | JobID | title | score | Job.ID |
|---|---|---|---|---|---|
| 0 | 3 | NaN | Banquet Servers and Prep Cooks @ Snelling | 0.53616 | 141798.0 |
| 1 | 3 | NaN | Grill Cook / Prep Cook / Chef @ First Class Wo... | 0.434776 | 145401.0 |
| 2 | 3 | NaN | Restaurant Cook - Prep - Bartender - Barback -... | 0.43311 | 137536.0 |
| 3 | 3 | NaN | COOK / COOK SUPERVISOR @ Arbor Terrace of Midd... | 0.350865 | 147306.0 |
| 4 | 3 | NaN | Hiring All Restaurant Positions - Servers - Co... | 0.349388 | 137581.0 |
| 5 | 3 | NaN | Hiring All Restaurant Positions - Servers - Co... | 0.349326 | 142265.0 |
| 6 | 3 | NaN | Hiring Restaurant Positions - Servers - Cooks ... | 0.348948 | 144106.0 |
| 7 | 3 | NaN | Hiring All Restaurant Positions - Servers - Co... | 0.348678 | 144105.0 |
| 8 | 3 | NaN | Hiring All Restaurant Positions - Servers - Co... | 0.348067 | 144111.0 |
| 9 | 3 | NaN | Hiring All Restaurant Positions - Servers - Co... | 0.347434 | 144108.0 |

Fig. 3 Job Recommendations

For candidate recommendation, we took the job ID of the requested position as input and measured the cosine similarity between the job record and all the candidate data. The candidates with top 10 scores were displayed as output.

```
Enter the job id: 111
```

| | Job.ID | text | Title |
|---|---|---|---|
| 0 | 111 | server tacolicious palo alto part time tacolic... | Server @ Tacolicious |

Fig. 4 User input for candidate recommendation

| | JobID | ApplicantID | keywords | score |
|---|---|---|---|---|
| 0 | 111 | 9135 | cook denny franchise milpitas cook denny franc... | 0.455783 |
| 1 | 111 | 601 | retail sale consultant retail bay area associa... | 0.442293 |
| 2 | 111 | 12664 | lucy activewear sale associate palo alto lucy ... | 0.423059 |
| 3 | 111 | 6808 | server tacolicious palo alto server tacoliciou... | 0.414649 |
| 4 | 111 | 12481 | office assistant officeteam palo alto server c... | 0.382295 |
| 5 | 111 | 9204 | software intern first data palo alto experienc... | 0.359377 |
| 6 | 111 | 13247 | macy seasonal retail sale macy appointment hol... | 0.327709 |
| 7 | 111 | 9686 | medical lab technician hill regional hospital ... | 0.282837 |
| 8 | 111 | 9704 | driving partner uber minneapolis driving partn... | 0.271391 |
| 9 | 111 | 13399 | lucy activewear assistant store manager palo a... | 0.264478 |

Fig. 5 Candidate Recommendations

**Conclusion:**

Recommendation System can be considered as a useful tool for recruiters and applicants. Leveraging past employment and job search data can enable improvement of performance of content-based recommendation systems. This application can serve as an important functionality within an Application Tracking Software for bridging the gap between job search and available career opportunities.

**References:**

https://medium.com/geekculture/cosine-similarity-and-cosine-distance-48eed889a5c4

https://medium.com/@armandj.olivares/building-nlp-content-based-recommender-systems-b104a709c042

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

https://medium.com/@armandj.olivares/building-nlp-content-based-recommender-systems-b104a709c042