JobSelect:
A Custom Job Recommendation System

Information Retrieval Project | IIIT Delhi

Table of Contents

01

Problem Statement

This project aims to create a system that recommends suitable job opportunities to job seekers based on their skills and preferences.

Problem Statement

Our recommendation System will help job seekers find jobs that match their qualifications and career goals more quickly and easily while also helping employers find more qualified and motivated candidates.

The system will use natural language processing and machine learning algorithms to analyze job seeker data, such as resumes and job applications, and suggest relevant job profiles.

AIM OF THE PROJECT

MOTIVATION

The job recommendation system can benefit job seekers and employers by making the job search and recruitment processes more efficient and effective.

Motivation : Benefits of Job Recommendation System

Saves Time

Job seekers often have to go through a large number of job postings to find relevant opportunities, which can be time-consuming and overwhelming.

Improvising Resume

Candidates often want to build their resume according to a specific job role so they can tweak their profile until their recommended job profile matches their choice.

Hiring Motivated Candidates

This system can help employers by matching them with more qualified and motivated candidates.

Related Work

We reviewed few papers which focuses on the work related to job recommendation systems. Some of them are:

1. Learning-Based Matched Representation System for Job Recommendation
2. A smart Geo-Location Job Recommender System Based on Social Media Posts
3. Investigating Natural Language Processing Techniques for a Recommendation System to Support Employers, Job Seekers and Educational Institutions
4. NLP-Based Bi-Directional Recommendation System: Towards Recommending Jobs to Job Seekers and Resumes to Recruiters
5. JobFit: Job Recommendation using Machine Learning and Recommendation Engine

Our Methodology

01 Data Extraction

02 Data Cleaning

03 Natural Language Processing

04 Recommending Job Profiles

Data Extraction

1. We have used two datasets in this project : **Job Postings Data (Training Data)** and **User Resume (Testing Data)**
2. Job postings and user resume data has been extracted from **LinkedIn**.
3. **Selenium** has been used for scraping the data from these websites. Job seekers are expected to provide the resume from which the information (skillset) will be extracted.

Data Cleaning

1. After data extraction, all the redundant information was removed from the data to make sure that only relevant data is present.
2. For each user resume in the test data, we assigned **three job profiles** ranked in order of descending order of their relevance.

Natural Language Processing

NLP techniques used for the processing of the scraped data are:

- Tokenization
- Stop words removal
- Lemmatization and Stemming to retrieve the key terms/skills from resume and job profiles.

Word Net and Porter algorithms are used for the lemmatization and stemming respectively.

Recommending Job Profiles

Count Vectorizer and TF-IDF is used to learn the embeddings (matrix of token counts) of the text documents. Thus, this pre-trained deep learning model has been used to match the expected skill set for a job profile and skills present in candidate profile. TF-IDF method is also used to learn the word embeddings from the text documents. Classification techniques like **Random Forest, Stacking Classifier** are used for recommendation of most suitable jobs based on matching between candidate and job profile.

Baseline Model

Count Vectoriser + Random Forest Classifier

The model is trained to analyze job descriptions and recommend the most suitable job profiles.

- The Count Vectorizer method breaks down the job description into smaller pieces and analyzes them.
- This method represents the job description as a bag-of-words, which means it breaks the text into individual words and counts the frequency of each word. This allows the model to capture the most important words that are relevant to the job profile.
- Random forest is then applied to the matrix, built using count vectoriser, which classifies new documents based on their word frequencies. The algorithm works by creating an ensemble of decision trees, where each tree is trained on a different subset of the data. Each decision tree splits the data into smaller and smaller subsets based on the features (words) in the data, until it reaches a leaf node, which represents a classification decision.

Other Models

TF-IDF + Stacking Classifier

The model is trained to analyze job descriptions and recommend the most suitable job profiles.

- Using the TF-IDF approach, each word in a text is given a weighted significance score depending on how frequently it appears there and throughout the corpus as a whole.
- The fundamental principle behind TF-IDF in the context of job recommendations is to determine the weighting of keywords in a job description and a candidate's profile. The more frequently a keyword appears in a job description, the more crucial it is probably to that position.
- Stacking Classifier combines the predictions of multiple machine learning models to make a more accurate prediction. In other words, it takes the output of several models and trains another model on top of them to make a final prediction. This can help to reduce errors and increase the accuracy of the prediction. The machine algorithms used in the Stacking classifier are Random Forest and XGBoost algorithm.

Other Models

Count Vectoriser + Stacking Classifier

The model is trained to analyze job descriptions and recommend the most suitable job profiles.

- Count Vectorizer was again used to create word embeddings (matrix of token counts) of the documents. Then this token count matrix was given input to the stacking classifier (consisting of Random Forest and XGBoost Classifier).

Advanced Models

Glove + Stacking Classifier

- "GloVe" (Global Vectors) was used for Word Representation. It is an unsupervised learning algorithm for obtaining vector representations (embeddings) for words based on their co-occurrence statistics.

- It builds a global co-occurrence matrix from the corpus, which records how frequently pairs of words appear together in the same context. Then, the algorithm factorizes this matrix to obtain low-dimensional vectors for each word that capture its semantic meaning and contextual usage.

Advanced Models

BERT + Stacking Classifier

- BERT (Bidirectional Encoder Representations from Transformers,) is a type of transformer-based model that can be fine-tuned on a wide range of NLP tasks, such as text classification, named entity recognition, and question-answering.
- BERT was fine-tuned on our data which involved training the model by adding a layers on top of the pre-trained BERT model and updating the parameters through backpropagation.

Performance Analysis

	Precision@3	Recall@2	Mean Average Precision (MAP)
Count Vectoriser + Random Forest Classifier	0.645	0.763	0.680
Count Vectoriser + Stacking Classifier	0.693	0.852	0.736
TF-IDF Vectorizer + Stacking Classifier	0.659	0.802	0.684
BERT + Stacking Classifier	0.519	0.577	0.474
GLOVE + Stacking Classifier	0.609	0.670	0.461

- **Based on the results, the Stacking Classifier method with Count Vectorizer appears to perform the best in terms of precision, recall, and mean average precision. This suggests that this model is the most accurate in predicting job recommendations based on the job description and profile.**

User Interface

A user interface named **JOBSELECT** has been created for the models created.

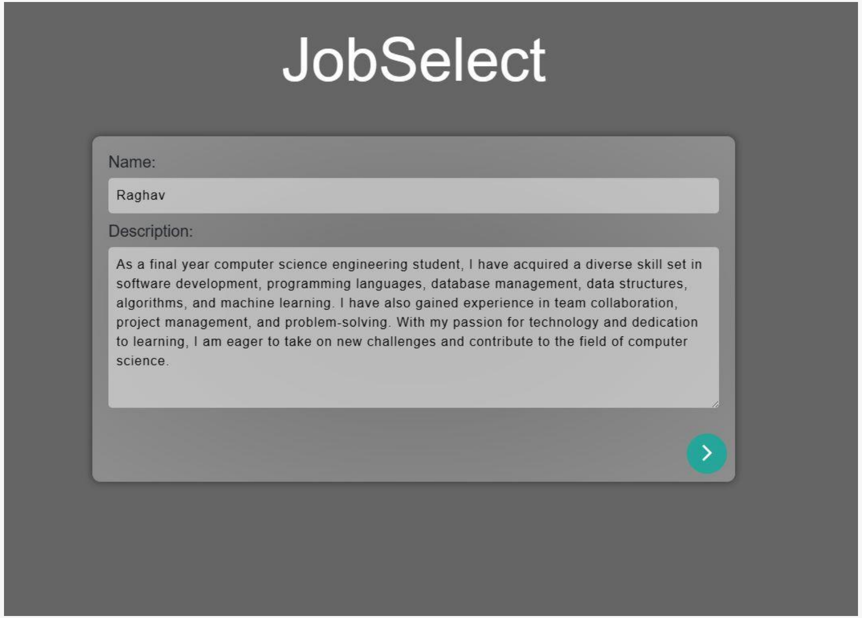
The candidate goes to the portal and enters his/ her name, description, courses done, skills, experiences and education details.

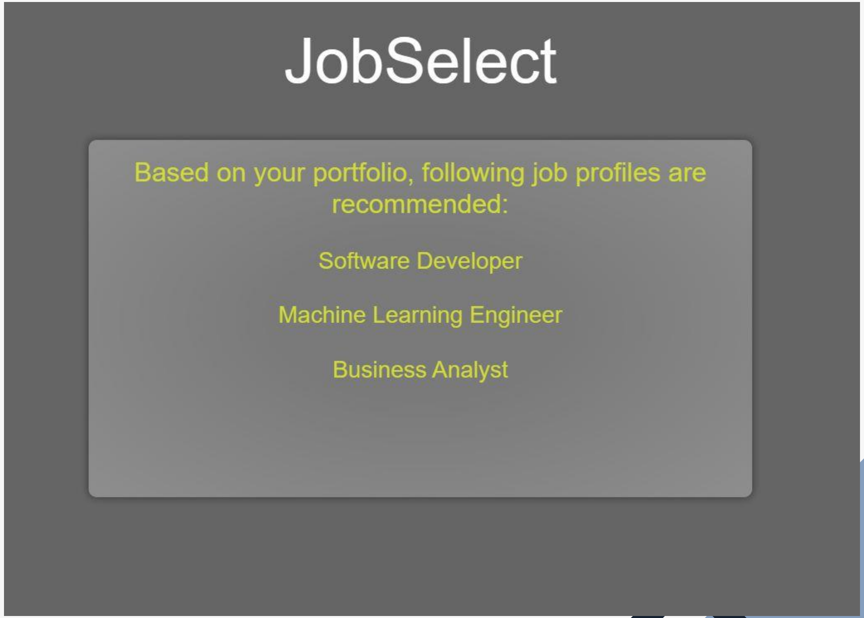
Then the portal recommends the candidate top **3 job profiles** matching best to the candidate.

The algorithms implemented at the backend are **Count Vectorizer for learning embeddings of the textual data from the user and then Stacking Classifier** (Random Forest and XGBoost Classifier with final estimator as the Logistic Regression).

Case 1 - Machine Learning written many times in job profile

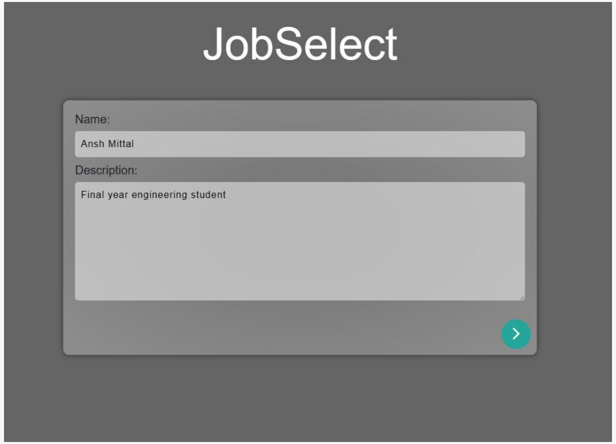
In this case, "machine learning" occur many times in job profiles and the top 3 predicted profiles comes out to be Software Developer, Machine Learning Engineer and Business Analyst.

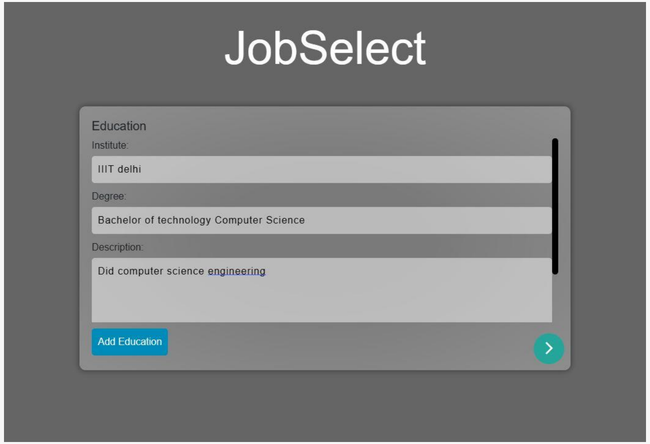




Case 2 - No job description just education details

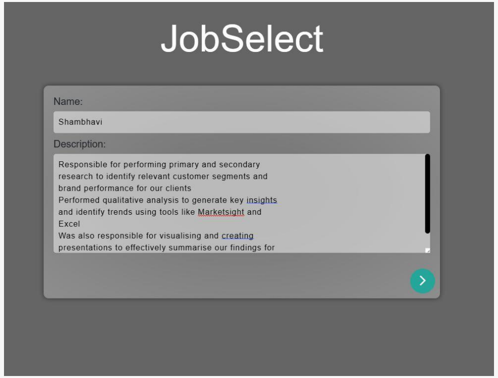
In this case, user enter only education details after which predicted profiles come out to be Software Developer, Machine Learning Engineer and Business Analyst.

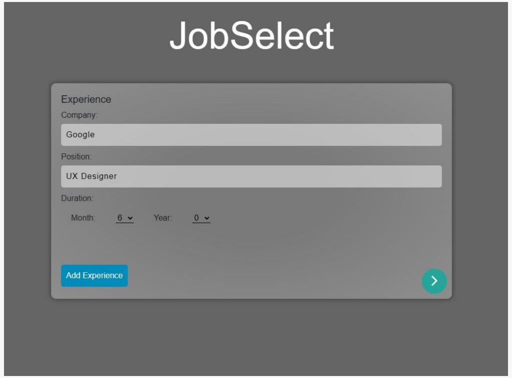


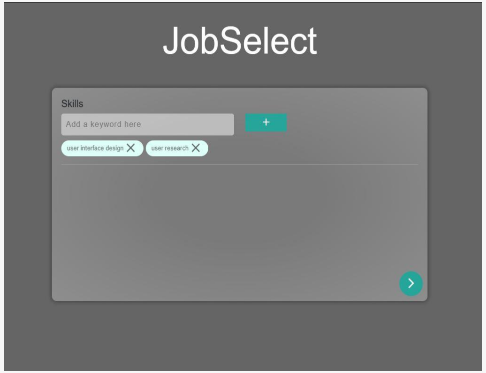


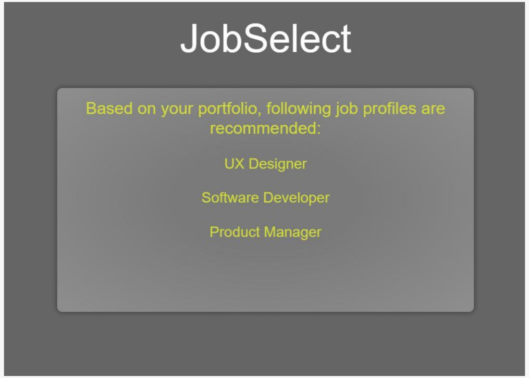
Case 3 - UX designer profile

In this case, we simply enter the profile details of a UX designer and the predicted profiles come out to be UX Designer, Software Developer and Product Manager.









Conclusion

- Various NLP and machine learning models **(BERT, GLOVE, Count Vectorizer, TF-IDF, Stacking Classifier and Random Forest Classifier)** were trained to analyze job descriptions and recommend the most suitable job profiles.

- The simple models like **Count Vectorizer and TF-IDF** are found to be performing better than the pre trained deep learning models trained on huge datasets like Wikipedia, etc.

- The reason for this is limited training data. If we have limited training data like in our case (training size = 914), the models like **BERT and Glove might overfit** on the training data.

- Also our data has short or medium simple textual data with titles in it which is considered ideal case for Count Vectorizer to perform.

- Also the Count Vectorizer approach seems better than the more sophisticated approach like TF-IDF and the reason can be because of occurrence of many stop words in the datasets which is in our Job Description case.

- The pre-trained models like BERT and Glove are better than Count Vectorizer when the problem is more **complex and versatile in nature**.

Individual Contributions

Data Web scraping	Rishita Chauhan
Natural Language Processing	Raghav Bhalla & Md. Zaid
Resume Information Extraction	Rishita Chauhan & Harshita Gupta
Data CleanUp	Harshita Gupta
Classification Algorithms & Evaluation	Ansh Mittal & Kartikey Gupta
User Interface	Ansh Mittal & Kartikey Gupta
Report & PPT	Raghav Bhalla & Rishita Chauhan
Video	Ansh Mittal & Rishita Chauhan

Efforts By:

Ansh Mittal
Harshita Gupta
Kartikey Gupta
Raghav Bhalla
Rishita Chauhan
Md. Zaid

Thank You!