

FitForward

Team members

Dinesh Anand Thulasiraman
Sailesh Baabu Suresh Babu
Sanjayram Raja Srinivasan

DATS 6312_11

Prof. Ning Rui

Agenda

I.Introduction

II.Exploratory Data Analysis (EDA)

III.Data Cleaning

IV.Web Scraping

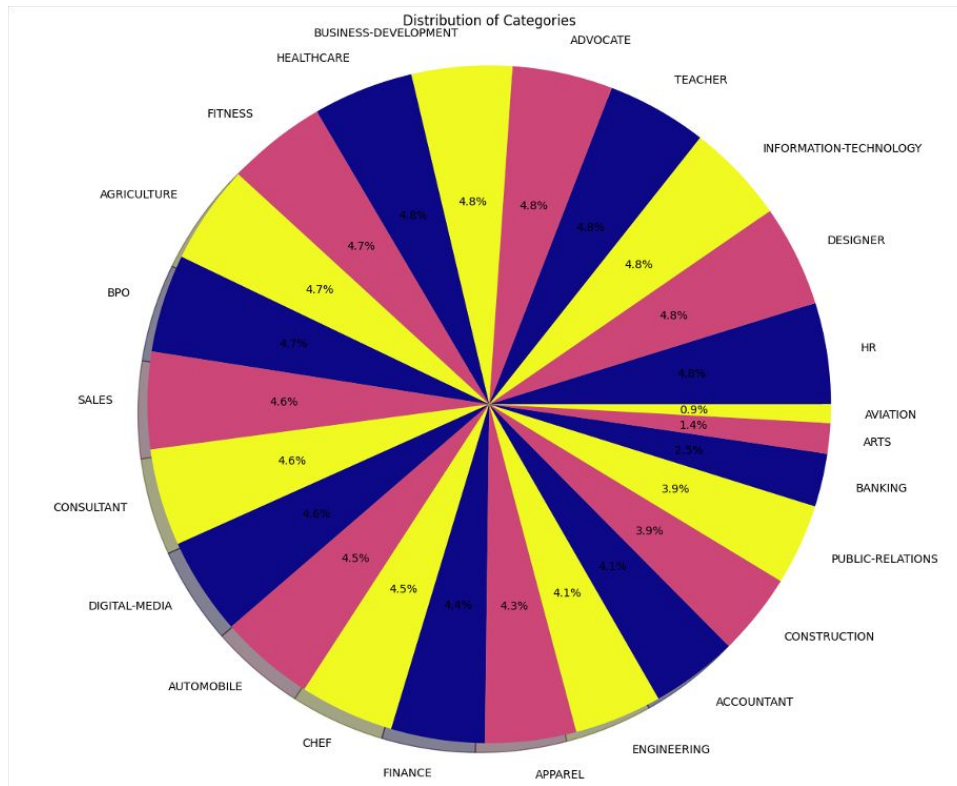
V.Deployment

Introduction

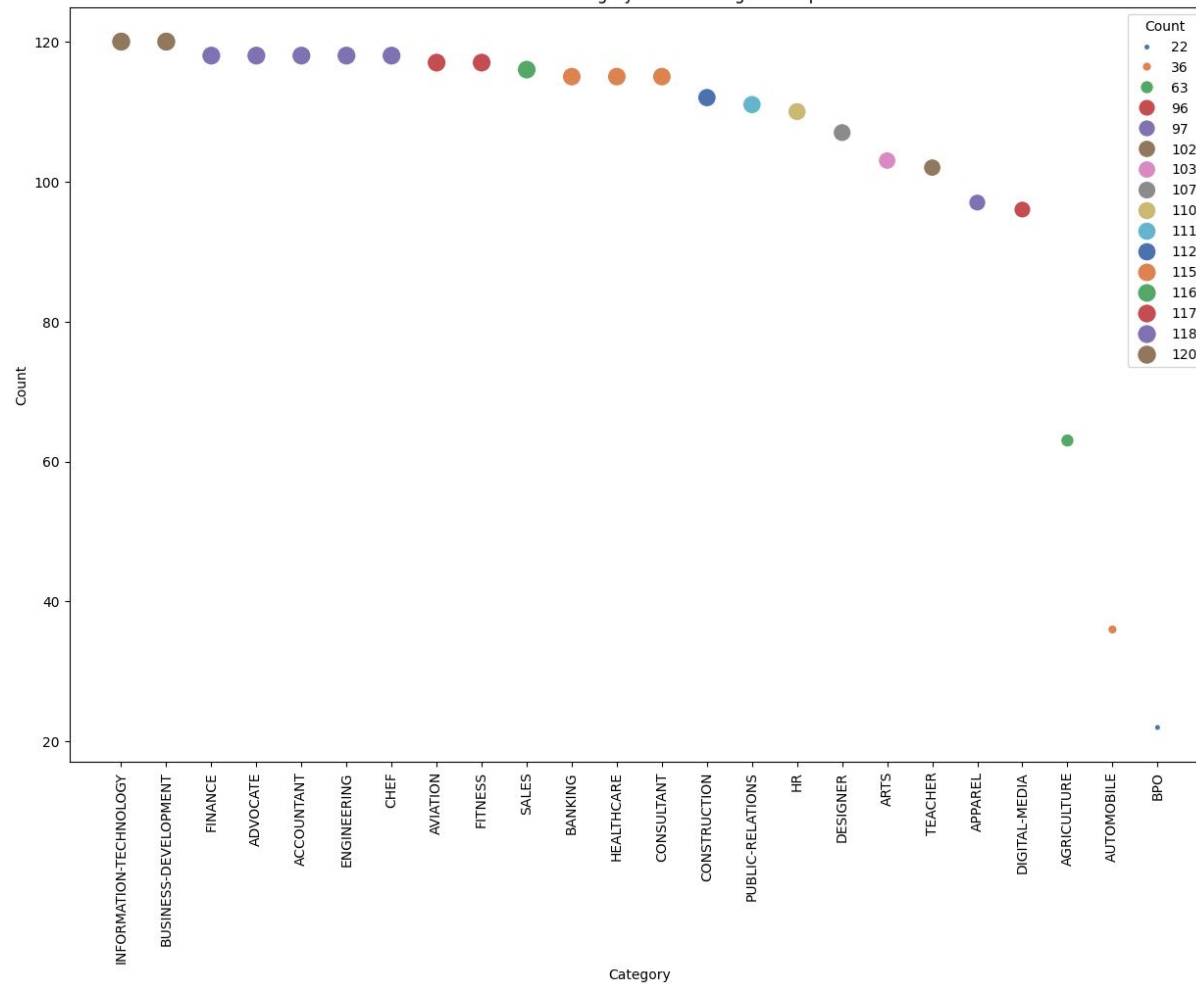
Objectives

- Resume-Based Job Finder simplifies job searching by analyzing resumes and finding matching job postings.
- Use AI to extract key insights from resumes and leverage keywords to find tailored job recommendations.
- KeyBERT for keyword extraction, Streamlit for app development, and JobSpy for web scraping job postings.

Exploratory Data Analysis



Distribution of Category Counts using Scatterplot



Data Cleaning

- Text preprocessing removes noise (e.g., punctuation, numbers, unnecessary spaces) to ensure data quality.
- Steps include normalizing text by converting to lowercase and cleaning unnecessary elements using `re.sub()`.
- Preprocessing ensures accurate and relevant keyword extraction by KeyBERT.

Web Scrapping

Definition:

Web Scrapping is the process of extracting data from websites automatically. In this project , we used the **jobspy** library to scrape job posting from platform like Indeed, LinkedIn and Glassdoor.

Implementation:

- The scrape_jobs function was used to collect job listings based on the selected or custom job title.
- **Key parameters configured:**
 - Search term: Job Title
 - Location: Remote (or any custom location)
 - Platforms: Indeed, LinkedIn, Glassdoor, etc.
 - Time range: Job posted within the last 72 hours.

The scraped data included:

- Job title, Company name, Description and application link.

Feedback Generation

Purpose:

Provide actionable insights to users to optimize their resumes for specific job postings.

Process:

1. Extract Resume Keywords:

- i. Utilized the KeyBERT model to extract top keywords (unigrams and bigrams) from the resume text.

2. Extract Job Keywords:

- i. Applied the same model on job descriptions scraped from the web to identify the top keywords.

3. Keyword Comparison:

- a. Compared the resume keywords with job-specific keywords to identify:
 - i. Matching Keywords: Skills or qualifications already present in the resume.
 - ii. Missing Keywords: Skills or qualifications in the job description but absent in the resume.

4. Feedback Example:

Highlight missing keywords like “cloud computing” or SQL and recommend their inclusion in the resume.

Cosine Similarity Score

Definition:

Cosine similarity is a measure of similarity between two text documentation, calculated as the cosine of the angle between their vector representations.

How It Works:

1. **Text Vectorization:**
 - a. Used TF-IDF to convert textual data into numeric vectors.
2. **Similarity Calculation:**
 - a. Computed the cosine similarity between the resume vector and job description vectors.
3. **Score Range:**
 - a. 0 to 1: Closer to 1 indicated a higher similarity between the resume and job description.

Implementation:

- Used the `cosine_similarity` function from `sklearn.metrics.pairwise`.
- Calculated similarity for all job descriptions and sorted the results to identify the top 5 most relevant jobs.

Deployment

- The application is implemented using **Streamlit**, a Python-based framework that simplifies the development of interactive web applications.
- Streamlit provides an intuitive user interface for tasks such as resume uploads, text extraction, and displaying feedback in real time.

Key Libraries:

- **PyPDF2**: Enables efficient extraction of text content from PDF resumes, ensuring compatibility with a widely used file format.
- **python-docx**: Facilitates text extraction from DOCX files, broadening the application's support for standard resume formats.

Code Design:

- **Text Extraction**: Processes and extracts raw content from resumes in various formats.
- **Text Preprocessing**: Cleans and normalizes extracted text for subsequent analysis.
- **Result Display**: Dynamically visualizes extracted keywords and job alignment scores.

Future Work and Enhancements

- 1. Interview Feedback Module**
- 2. Cover Letter Feedback**
- 3. Skill Gap Analysis and Training Recommendations**
- 4. Behavioral and Cultural Fit Analysis (Advanced AI Integration)**

References

1. KeyBERT Documentation: <https://github.com/MaartenGr/KeyBERT>
2. Streamlit Documentation: <https://docs.streamlit.io/>
3. JobSpy GitHub Repository: <https://github.com/Bunsly/JobSpy>
4. BeautifulSoup Documentation: <https://www.crummy.com/software/BeautifulSoup/>

Github Repo Link : <https://github.com/sanjayram01/Final-Project-Group-FitForward>

Streamlit Site: <https://fitforward.streamlit.app/>