<u>University of Florida – MS in Applied Data Science</u>

AI-powered Job Seek Tool – A conversational agent

By – Swetha Gendlur Nagarajan

Introduction

The proposed conversational agent would serve as a comprehensive job search and career development tool, leveraging advanced data analysis and visualization techniques to provide personalized guidance to job seekers. Upon engaging with the agent, users would input their desired role. The system would then analyze current job market trends for that position, presenting data visualizations to illustrate where the majority of opportunities are located geographically and what technology stacks are most in demand for that role.

The agent would request the user's resume and compare it against job descriptions in its database. Using natural language processing and keyword matching algorithms, it would calculate a percentage match between the user's skills and experience and the job requirements. For users deemed a good fit (based on a high match percentage), the agent would provide actionable next steps. This would include guidance on crafting effective cold emails to potential employers and providing contact information for relevant recruiters within target organizations.

If the user's qualifications fall short of the job requirements, the agent would offer a tailored improvement plan. This would involve identifying skill gaps and recommending relevant courses from platforms like Coursera to address those deficiencies.

To facilitate networking, the agent would compile a list of professionals currently working in the user's desired role, potentially leveraging data from professional networking platforms. This would allow users to connect with individuals who could provide valuable insights and potentially serve as internal referrals. Throughout the interaction, the agent would maintain a professional tone while providing detailed, data-driven insights to help users navigate their job search and career development journey effectively.

Type of Tool

The project will result in an interactive AI chatbot with natural language processing capabilities, data analysis functions, and visualization features. It will serve as a comprehensive career development platform, combining job market insights, resume analysis, and personalized recommendations.

Dataset Used

1. The Coursera dataset, sourced from Kaggle, provides a comprehensive overview of online courses available on the platform. It includes crucial information such as course titles, enrollment numbers, ratings, instructor details, skills taught, course descriptions, and difficulty levels. This dataset will be instrumental in recommending targeted learning opportunities to users looking to enhance their skill sets or bridge knowledge gaps identified during the job matching process.

2. The LinkedIn dataset, also obtained from Kaggle, offers valuable insights into professional profiles. It encompasses details such as names, current roles, work experience, skills, certifications, and contact information. While smaller in scale compared to the job description dataset, this information will be vital for the AI agent to suggest relevant networking connections and potential mentors in the user's desired field.
3. The job description dataset, acquired from Hugging Face, is significantly larger and more comprehensive than the other datasets. It contains a wealth of information about job postings, including job titles, required experience, qualifications, locations, company details, salary ranges, and detailed job descriptions. The sheer volume and depth of this dataset will allow for robust job market analysis, accurate skill matching, and the provision of detailed insights into job trends and geographical distributions.
4. The roles-based skills dataset, another Kaggle resource, focuses specifically on the skill requirements for various positions. It includes information such as company names, position titles, core responsibilities, required skills, educational requirements, and experience levels. While not as extensive as the job description dataset, it provides valuable, role-specific data that will enable the AI agent to offer tailored advice on skill development and career progression paths.

1. LinkedIn dataset:

1. **Data Loading and Preparation:**
   a. The code loads a dataset of synthetic LinkedIn profiles using the datasets library.
   b. It converts the dataset into a Pandas DataFrame for easier data manipulation.
   c. It performs initial data cleaning, such as removing unnecessary columns (like embeddings, 'About Me', 'Recommendations', etc.) and renaming columns for better clarity (e.g., renaming 'Headline' to 'Current Role').
2. **Data Cleaning and Transformation:**
   a. It further cleans the 'Current Role' column to extract the primary role from the text.
   b. It creates a new 'Contact mail' column by generating hypothetical email addresses based on first and last names.
3. **Data Filtering and Analysis:**
   a. It filters the data to focus on profiles with 'data scientist' in their current role.
   b. It extracts and analyzes the skills mentioned by data scientists, identifying the most common skills.
   c. It generates a bar chart to visualize the frequency of these top skills.
4. **Role Analysis:**
   a. It extracts and analyzes the current roles of all profiles, identifying the most common roles.
   b. It generates a bar chart to visualize the frequency of these top roles.

In essence, the code loads a dataset of LinkedIn profiles, cleans and prepares the data, focuses on data scientist profiles, analyzes their skills and roles, and visualizes the findings using bar charts.

2. Job Description Data:

Data Preprocessing Steps:

1. **Data Loading and Cleaning:**
    a. Loads a CSV file containing job descriptions into a pandas DataFrame.
    b. Removes unnecessary columns (location, Benefits).
    c. Handles missing data by imputing or removing values in columns like Company Profile, Company, Sector, Industry, Website.
    d. Converts salary information from text to numerical format, calculating average salary.
    e. Extracts numerical experience range from text and calculates average experience.
    f. Addresses potential outliers in numerical columns (e.g., Average Salary, Experience) using statistical methods.
    g. Normalizes or scales numerical features using Min-Max scaling and standardization.
2. **Data Transformation:**
    a. Converts categorical columns (e.g., Work Type, Preference, Job Title) to the appropriate data type.
    b. Identifies and removes columns with all zero values.

Exploratory Data Analysis (EDA) Steps:

1. **Descriptive Statistics:**
    a. Uses descriptive statistics like mean, median, and standard deviation to summarize the dataset.
2. **Visualizations:**
    a. Creates various visualizations (histograms, scatter plots, box plots, bar charts, heatmaps, pie charts) to explore patterns, trends, and relationships in the data.
    b. Examples include:
        i. Gender distribution by job title
        ii. Average salary by qualification and job type
        iii. Job demand over time
        iv. Geographical spread of job titles
3. **Correlation Analysis and Statistical Tests:**
    a. Identifies potential issues such as multicollinearity or skewed distributions using correlation analysis and statistical tests (not explicitly shown in the provided code snippet).

Overall Goal:

The code aims to prepare the job description dataset for further analysis (e.g., machine learning model building) by cleaning, transforming, and exploring the data. The EDA provides insights into the data, such as gender distributions in job roles, salary trends, and qualifications in demand.

3. Roles and Skills Data:

Data Loading and Preprocessing

1. **Import necessary libraries:** The code begins by importing pandas for data manipulation, json for handling JSON data, and other libraries for analysis and visualization.
2. **Load datasets:** Two datasets are loaded:

    a. job-descriptions: Contains job descriptions and related information like position titles, company names.
    b. roles-based-on-skills: Contains roles and their required skills.
3. **Data Cleaning and Transformation:**
    a. The code extracts relevant information from the 'model_response' column in job-descriptions and creates new columns.
    b. Unnecessary columns (like job_description, description_length, model_response) are dropped.
    c. Missing values are handled by filling them with the most frequent value in each column.
    d. Duplicate rows are removed based on the 'position_title' column.
    e. Data types of some columns are converted (e.g., company_name and position_title to categorical).
    f. Outliers in categorical columns (like position titles with very low frequency) are identified and printed.

Data Integration and Exploration

1. **Data Integration:** Roles from the roles-based-on-skills dataset that are not present in the job-descriptions dataset are added to the main dataframe (df_js). This expands the dataset with new roles and their required skills.
2. **Exploratory Data Analysis (EDA):**
    a. The code focuses on visualizing the relationship between the top 30 companies and position titles.
    b. It creates a heatmap using seaborn and matplotlib to show the frequency of each position title within each of the top 30 companies.

Purpose

The code aims to analyze and understand the relationship between companies, job positions, and required skills. By loading, cleaning, transforming, and integrating two datasets, it provides a foundation for understanding the job market landscape. The EDA step helps visualize the distribution of position titles within different companies, highlighting potential trends and patterns. This information can be valuable for tasks like job recommendation, skill gap analysis, or market research.

4. Coursera Course Details:

1. **Data Loading and Cleaning:**
    a. The code starts by loading a Coursera course dataset from a CSV file using pandas.
    b. It then cleans the data by:
        i. Removing unnecessary columns (Unnamed: 0, Satisfaction Rate).
        ii. Handling missing values by filling them with appropriate measures (mean, mode).
        iii. Cleaning and converting data types of certain columns (e.g., Modules/Courses, Schedule).
2. **Data Exploration and Visualization:**
    a. After cleaning, the code explores the data through:

        i. Identifying and visualizing outliers in numerical columns like rating, num_reviews, and Modules/Courses using box plots.
        ii. Visualizing the distribution of the 'enrolled' column using a histogram.

3. **Key Insights Visualization:**
    a. Finally, the code visualizes key insights:
        i. It creates a bar plot to showcase the top 20 highest-rated courses, highlighting the instructors and ratings.
        ii. It also creates a bar plot showing the top 10 longest courses (based on the number of modules) among the top 20 courses.
        iii. Another bar plot presents the top 10 courses with the highest number of reviews.

Tech Stack

1. Python for backend development and data analysis (pandas, numpy)
2. Natural Language Processing libraries (NLTK or spaCy)
3. Machine Learning frameworks (TensorFlow or PyTorch)
4. Data visualization libraries (Matplotlib, Seaborn, or Plotly)
5. Flask or Django for web framework
6. Open AI API, Langchain frameworks

Timeline

1. Feature Engineering (~ 2 weeks)

Objective: Transform raw data into meaningful predictors for resume-job matching and skill gap analysis.

- **Process**:
    - **Job Description Features**:
        - Extract skills, tools, and certifications using NLP (e.g., spaCy).
        - Create interaction terms (e.g., "Python + SQL" as a combined skill score).
        - Engineer time-based features (e.g., years of experience vs. job posting trends).
    - **Resume Features**:
        - Parse resume text into structured skills, roles, and tenure.
        - Aggregate metrics (e.g., "number of certifications relevant to target role").
    - **Coursera/LinkedIn Features**:
        - Encode course difficulty levels (label encoding).
        - One-hot encode job sectors (e.g., "Tech", "Healthcare").
- **Milestone**: Dataset with engineered features ready for modeling.

2. Feature Selection (~ 2 Weeks)

Objective: Identify the most impactful features for accurate job-resume matching.

- **Process**:
    - **Correlation Analysis**:
        - Calculate feature correlations with target variables (e.g., "job fit score").
        - Use heatmaps to visualize relationships (e.g., salary vs. certifications).
    - **Tree-Based Importance**:
        - Train a Random Forest to rank features (e.g., "Python" > "Excel" for data roles).
    - **Dimensionality Reduction**:
        - Apply PCA to skills matrices for roles with overlapping requirements.
        - Use LASSO regression to eliminate noisy features (e.g., irrelevant soft skills).
- **Milestone**: Final feature set with top 20 predictors (e.g., key skills, experience thresholds).

3. Data Modeling (~ 1 Week)

Objective: Build and optimize models for job matching and course recommendations.

- **Process**:
    - **Model Selection**:
        - Logistic Regression (baseline for job fit classification).
        - Random Forest (skill importance analysis).
        - Neural Networks (BERT-based resume-job description similarity).
    - **Hyperparameter Tuning**:
        - Grid search for optimal Random Forest depth/n_estimators.
        - Learning rate optimization for neural networks.
    - **Validation**:
        - Split data (80% training, 20% testing) stratified by job roles.
        - Cross-validate to prevent overfitting (5-fold CV).
- **Milestone**: Top-performing model with 90%+ cross-validation accuracy.

4. Evaluation & Interpretation (~ 2 Weeks)

Objective: Validate model performance and derive actionable insights.

- **Process**:
    - **Performance Metrics**:
        - Precision/Recall: Measure relevance of recommended jobs/courses.
        - ROC-AUC: Evaluate resume-job matching confidence scores.
        - F1-Score: Balance false positives/negatives in "fit vs. unfit" classification.
    - **Bias Checks**:
        - Analyze model fairness across demographics (e.g., gender-neutral skill recommendations).
        - Audit for geographic bias in job opportunity predictions.

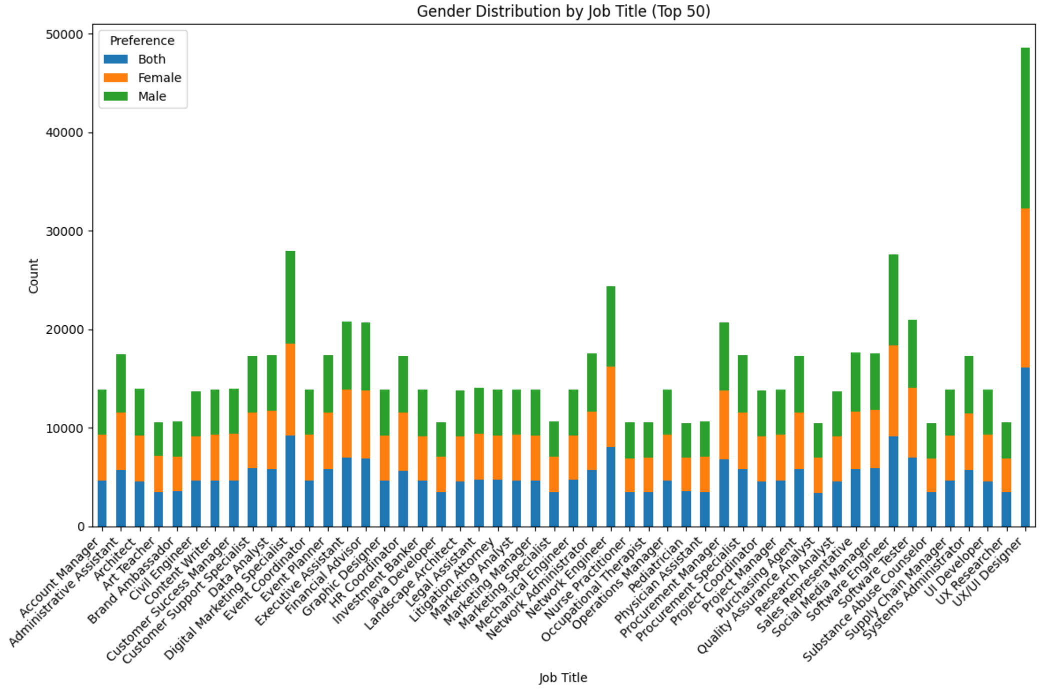
  - **Explainability**:
    - SHAP values to interpret BERT model outputs (e.g., "Python contributed 40% to fit score").
    - Visualize decision trees for skill gap recommendations.
- **Milestone**: Evaluation report with metrics, bias audit, and stakeholder-ready insights.

Key Deliverables Timeline

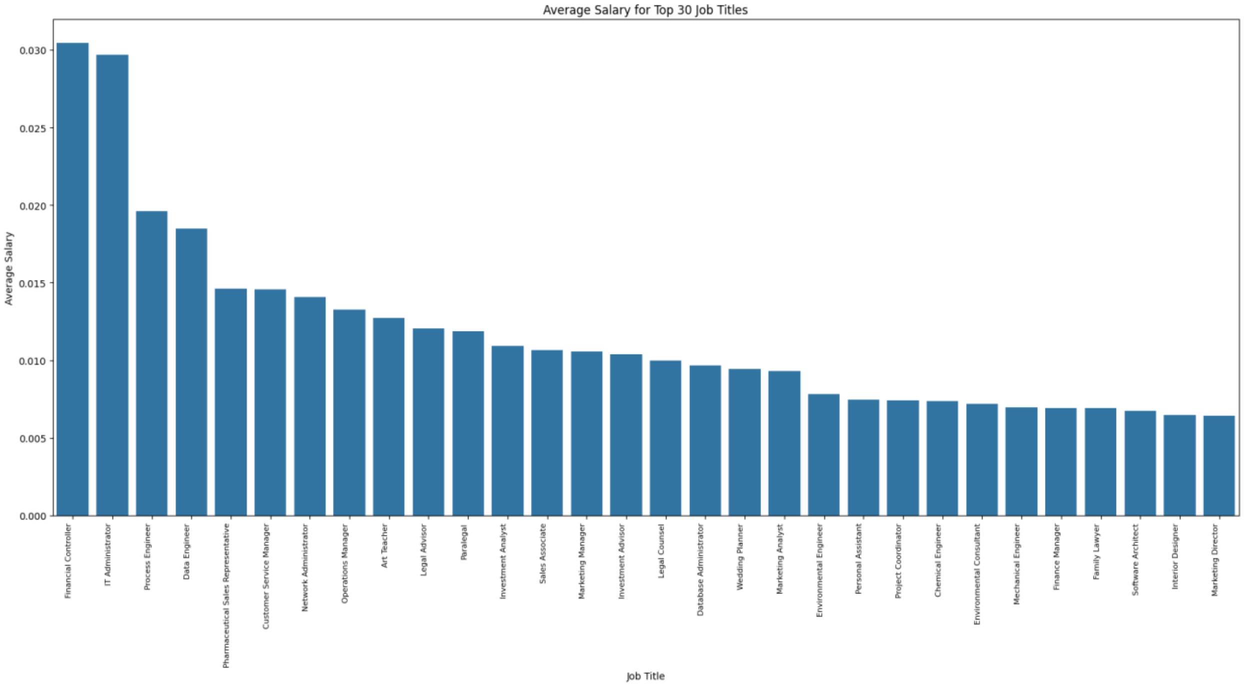
Phase	Outputs
Feature Engineering	Cleaned datasets with engineered features (skills scores, interaction terms)
Feature Selection	Heatmaps, PCA/LASSO results, final feature list
Data Modeling	Trained models (Random Forest, BERT), hyperparameter tuning logs
Evaluation	ROC curves, SHAP plots, bias audit report

Exploratory Data Analysis (Only main ones are focused)

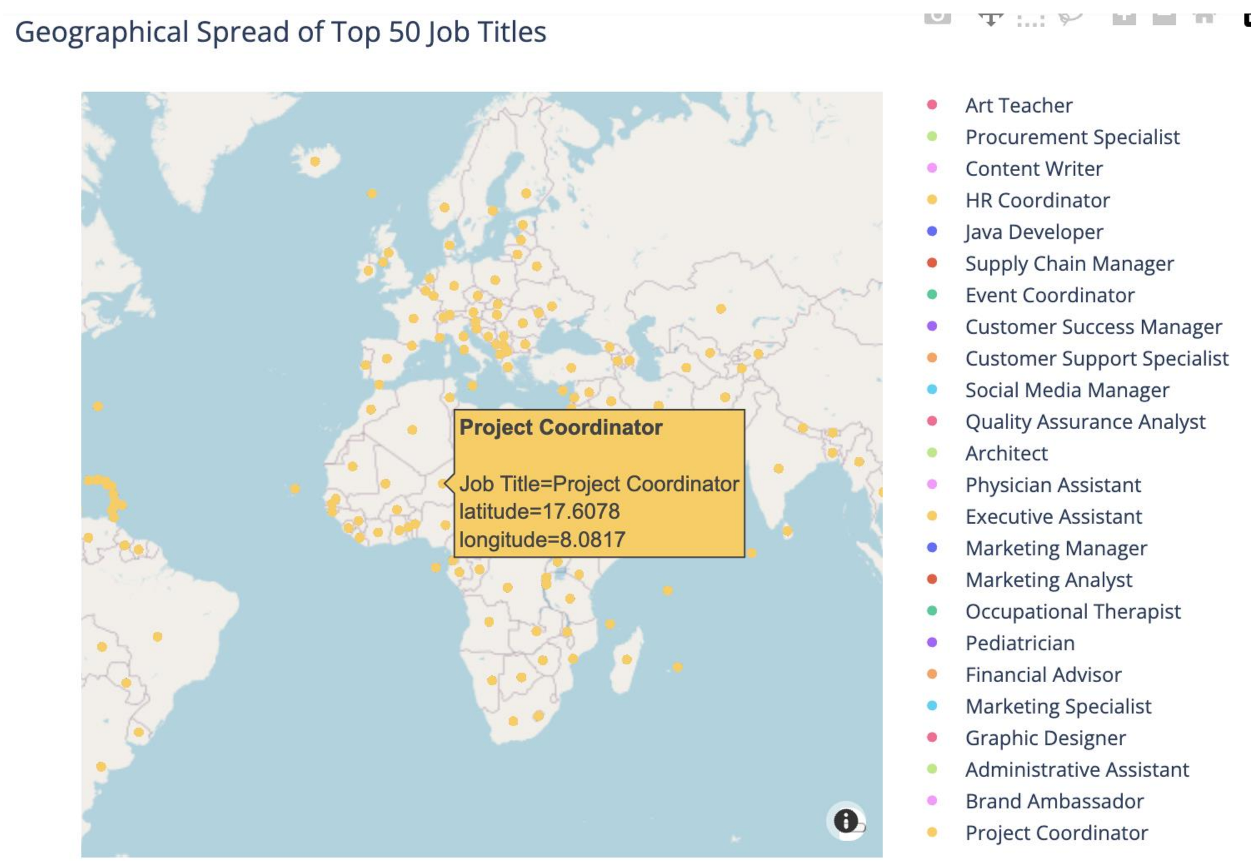
1. Job Description



This Visualization shows the Gender distribution for a particular role for top 50 roles.

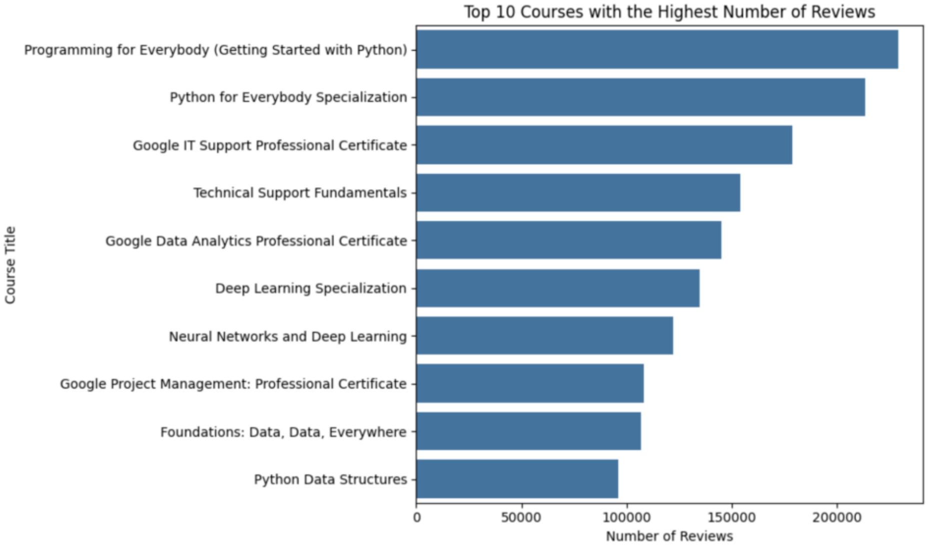


This visualization shows spread of data for the salary and the top roles from highest to lowest



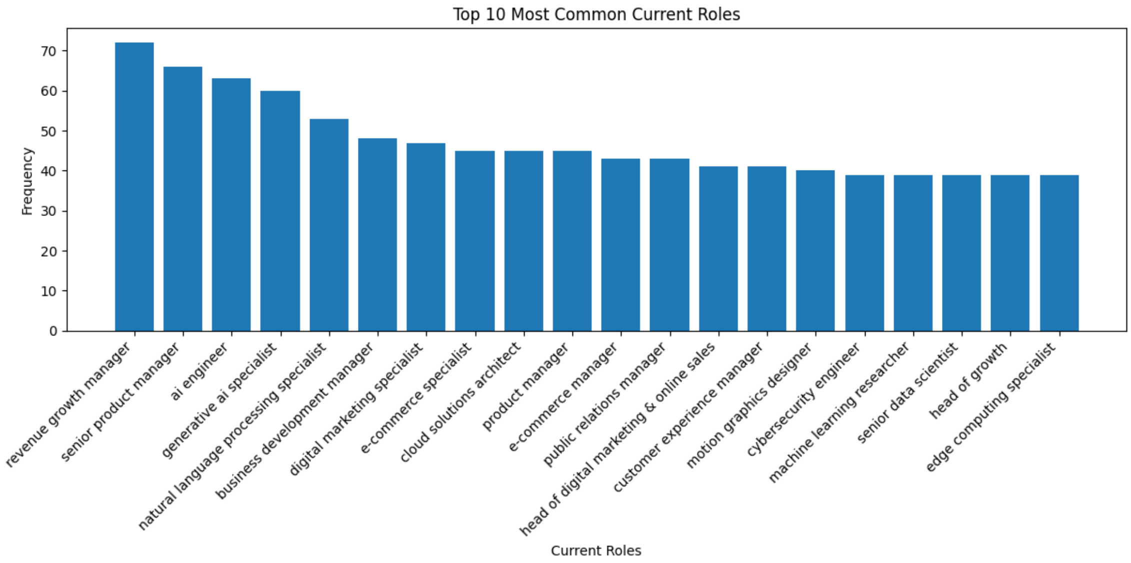
This is a Geographical Spread of Jobs

2. Coursera Details

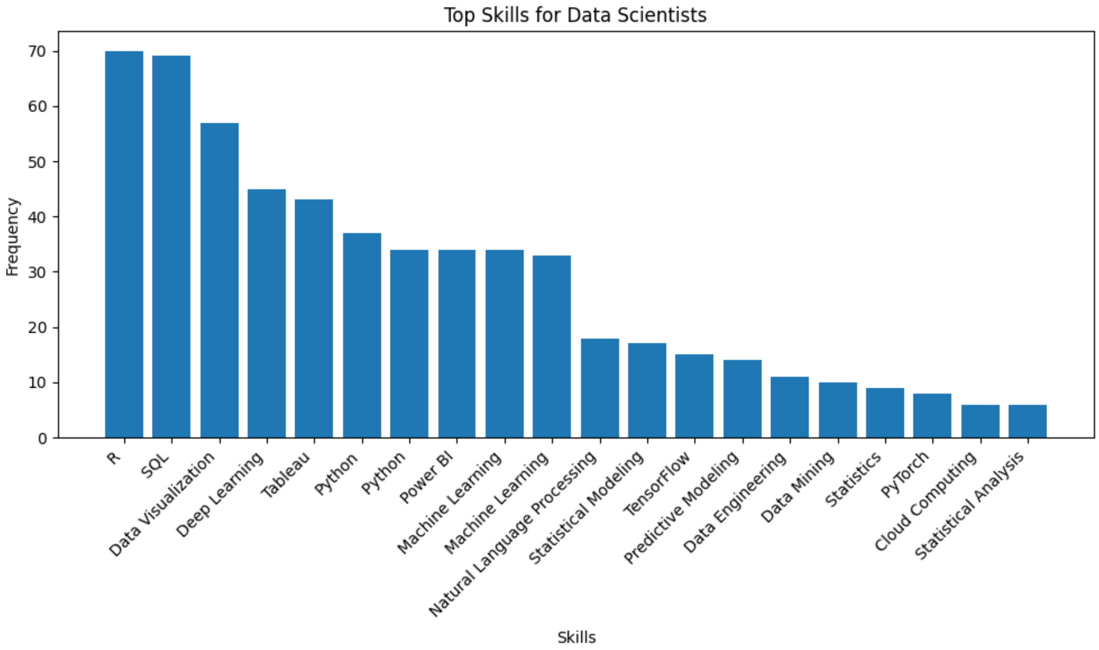


This displays the top courses according to number of Reviews and Similarly in the ipynb file according to the rating the courses have been ranked.

3. LinkedIn Data



This Visualization shows the trending current roles on demand based on the number of employees on that domain



This shows that the top skills required for the Data scientist