***DATA 606: ASSIGNMENT***

***Team: Giridhar Sriram Janga, Sai Bhargav Karnati, Jayanth Rachuri***

**Project – 1:**

**AI-Powered Job Recommendation & Career Insights Dashboard**

**Dataset:**

Scraped Job Postings (Primary Dataset):

* Source: Glassdoor (scraped biweekly).
* Fields: job title, company, location, salary estimate, company rating, job description.
* Stored in AWS S3 and updated automatically.

Personal Resume & Portfolio (User Dataset):

* Parsed and analyzed to extract skills, experiences, and education.
* Used as input for personalized job recommendations.

**Link:** https://drive.google.com/drive/folders/1V4gAjJtaaUJTQU8QA8JzfpMF7GS4ohTS?usp=sharing

**Problem Statement:**

Job seekers often struggle with two major challenges: filtering through countless postings to find roles that genuinely match their skills and experience, and staying informed about industry trends such as in-demand skills, salary benchmarks, and career paths. This project addresses both issues by developing an AI-powered system that not only recommends jobs ranked by their alignment with a candidate’s resume but also provides an interactive dashboard that delivers live, weekly insights into the job market—offering newcomers to the data industry a clearer, data-driven view of opportunities and trends.

**Planned Analysis and Method:**

**Step 1: Data Pipeline & Storage**

* Scraper collects new postings → stores in AWS S3.
* Data cleaned and transformed into structured form.
* Historical records maintained for trend analysis.

**Step 2: NLP for Resume & Job Description Matching**

* Skill Extraction: Use NER or rule-based extraction to pull skills from job descriptions and resume.
* Semantic Similarity: Use cosine similarity or BERT embeddings to compare resumes with job descriptions.
* Job Ranking: Sort jobs by percentage match to candidate’s profile.

**Step 3: ML Component**

* Classification Model: Predict whether a job is a “good fit” for the candidate (based on skill match, experience level, etc.).
* Trend Forecasting: Use Time Series models (Prophet, ARIMA) to forecast demand for top skills.

**Step 4: Career Insights Dashboard**

* Built in Tableau/PowerBI.
* Features:
  + Top skills in demand (weekly update).
  + Salary trends by role and location.
  + Geographic distribution of opportunities.
  + Resume-to-job match scores (for the user).

**Performance Evaluation Metrics:**

* **Job Matching Model:**
  + Precision@K, Recall@K, and cosine similarity scores.
  + F1-score (if classification model is used).
* **Skill Forecasting:**
  + RMSE, MAE, or MAPE on predicted vs. actual demand.
* **Dashboard Quality:**
  + Qualitative evaluation based on interpretability and usability.

**PROJECT-2:**

**Document Search with Retrieval-Augmented Generation**

**Dataset**  
The arXiv dataset is used - link - https://www.kaggle.com/datasets/Cornell-University/arxiv. It has research papers from arXiv.org. The dataset is large with over 1.7 million records. Each record has details such as the paper ID, title, abstract, authors, categories, and sometimes a journal reference. The files are provided in CSV and JSON formats. Since working with full papers would be too heavy, I plan to focus mainly on the titles and abstracts. The data covers many subject areas like computer science, physics, and mathematics.

**Planned Analysis**

The main goal is to create a system that lets users search large amounts of text more effectively.

Instead of just using keyword searches, I will use Retrieval-Augmented Generation (RAG), which mixes document retrieval with a language model. The project will include:

1. Finding the most relevant documents or abstracts for a query using embeddings.

2.Using a language model to generate answers or short summaries based on the retrieved text.

3.Comparing results from embedding-based retrieval with traditional methods like BM25

**Evaluation Metrics**

To compare performance, I'll take into consideration:-

1. Precision@k, Recall@k, and Mean Reciprocal Rank (MRR) to measure how good the system is at retrieving relevant documents.

2. ROUGE or BLEU scores to compare output responses to reference responses (if available).

3. Basic system metrics like response time to questions.

**Expected Results**

The outcome will be a prototype search and QA system running on the arXiv dataset. Along with the tool, I plan to contrast different retrieval methods and illustrate trade-offs between accuracy and efficiency.