A PROJECT ON

"JOB INTELLIGENCE SYSTEM"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN BIG DATA ANALYSIS

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY, PUNE

Submitted By:

Omkar Tawde (92963)

Sushant Borkute (92874)

Mr.Nitin Kudale
Centre Coordinator

Mrs.Manisha Hingne
Course Coordinator

CERTIFICATE

This is to certify that the project work under the title ‘job intelligence system’ is done by Omkar Tawde & Sushant Borkute in partial fulfillment of the requirement for award of Diploma in Big Data Analysis Course.

Mr. Aniket P
Project Guide

Mrs. Manisha Hingne
Course Coordinator

Date:04/02/2026

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mrs. Manisha Hingne (Course Coordinator, SIIT ,Pune) and Project Guide Mr. Aniket P.

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Omkar Tawade
DBDA August 2025 Batch,
SIIT Pune

Sushant Borkute
DBDA August 2025 Batch,
SIIT Pune

TABLE OF CONTENTS

1. Introduction
1.1. Introduction And Objectives
1.2. Why this problem needs To be Solved?
1.3. Dataset Information

2. Problem Definition and Algorithm
2.1 Problem Definition
2.2 Algorithm Definition

3. Experimental Evaluation
3.1 Methodology/Model
3.2 Exploratory Data Analysis

4. Results And Discussion

5. GUI

6. GitHub link

7.Future Work And Conclusion
7.1 Future Work
7.2 Conclusion

1. Introduction

1.1 Introduction And Objectives:

In today's rapidly evolving job market, professionals and job seekers face significant challenges in understanding salary trends, evaluating job opportunities, and making informed career decisions. The complexity of factors influencing compensation—including experience level, geographic location, industry sector, required skills, and company size—makes it difficult for individuals to accurately assess their market value or predict potential earnings.

The Job Intelligence System addresses these challenges by leveraging advanced artificial intelligence and machine learning technologies to provide comprehensive insights into the AI and technology job market. This integrated platform combines predictive analytics, natural language processing, and interactive data visualization to empower users with actionable intelligence about salary expectations and job market dynamics.

Built using Python's robust data science ecosystem and powered by Google's Gemini AI, the system offers three core functionalities: resume-based salary prediction, an intelligent chatbot for job market queries, and a global job explorer with interactive mapping capabilities. By processing real-world job market data and extracting insights from user resumes, the platform delivers personalized salary estimates and market intelligence that help professionals navigate their career trajectories with confidence.

1.2 Why this problem needs To be Solved?

The problem of accurate salary prediction and job market intelligence must be solved to address critical information asymmetry in the employment market that leads to unfair compensation, particularly affecting early-career professionals and underrepresented groups. Without access to data-driven salary benchmarks, job seekers often accept positions significantly below market value, struggle to make informed decisions about skill investments and career transitions, and spend excessive time manually researching fragmented and outdated compensation data across multiple sources. In the rapidly evolving AI and technology sectors, where new roles emerge continuously and technical skills quickly become obsolete, professionals need real-time market intelligence to understand the complex interplay of factors—including job title, location, experience, education, and required skills—that determine compensation. The Job Intelligence System democratizes access to sophisticated career analytics previously available only to

large organizations, empowering all professionals to negotiate fair salaries based on objective market data, optimize their career trajectories through informed decision-making, and achieve economic equity in an increasingly competitive and globalized job market.

1.3 Dataset Information.

The dataset contains comprehensive information about AI and technology job postings with the following columns:

Job Details:

- **job_title:** Specific job position title (e.g., AI Research Scientist, Machine Learning Engineer, Data Scientist, NLP Engineer)
- **employment_type:** Type of employment contract (FT=Full-time, PT=Part-time, CT=Contract, FL=Freelance)
- **company_name:** Name of the hiring organization
- **industry:** Business sector of the company (e.g., Technology, Healthcare, Finance, Automotive, Consulting)

Location and Work Arrangement:

- **company_location:** Country where the company is based (e.g., United States, India, Canada, Germany, Switzerland)
- **employee_residence:** Country where the employee resides
- **company_size:** Organization size classification (S=Small, M=Medium, L=Large)
- **remote_ratio:** Percentage of remote work allowed (0%, 50%, or 100%)

Compensation:

- **salary_usd:** Annual salary in US Dollars (target variable for prediction)
- **salary_currency:** Original currency of the salary (USD, EUR, GBP, etc.)

Candidate Qualifications:

- **experience_level:** Professional experience tier (EN=Entry-level, MI=Mid-level, SE=Senior, EX=Executive)
- **education_required:** Minimum educational qualification needed (Associate, Bachelor, Master, PhD)
- **years_experience:** Number of years of professional experience required
- **required_skills:** Technical skills needed for the position (e.g., Python, TensorFlow, NLP, Deep Learning, AWS)

Job Posting Metadata:

- **posting_date:** Date when the job was posted
- **application_deadline:** Last date to apply for the position
- **job_description_length:** Character count of the job description
- **benefits_score:** Numerical rating of employee benefits package (0-10 scale)

Derived Features (Created during preprocessing):

- **num_required_skills:** Count of technical skills extracted from the required_skills column
- **log_salary:** Logarithmic transformation of salary_usd for model training (reduces skewness)
- **encoded categorical variables:** One-hot encoded features for job_title, company_location, industry, and other categorical columns
- **ordinal encoded features:** Ordinal encoding for experience_level, education_required, and company_size

2. Problem Definition and Algorithm:

2.1 Problem Definition

The primary problem addressed by the Job Intelligence System is the inability of job seekers in AI and technology fields to accurately predict their salary expectations due to the complex interaction of multiple factors such as job title, location, experience level, education, required skills, company size, industry, and remote work arrangements. Traditional salary surveys and compensation databases fail to capture these multidimensional relationships and quickly become outdated, creating information asymmetry that disadvantages candidates during negotiations. Additionally, job seekers face challenges in manually extracting professional information from resumes, researching fragmented salary data across multiple sources, and understanding global compensation variations without intelligent analytical tools. This project solves these problems by developing an AI-powered system that automatically parses resumes using generative AI, predicts personalized salaries through machine learning models trained on comprehensive job market data, provides intelligent answers to career queries via a RAG-based chatbot, and visualizes high-paying global opportunities through interactive geographic mapping, thereby empowering professionals to make informed, data-driven career decisions and negotiate fair compensation in the competitive technology job market.

2.2 Algorithm Definition

Random Forest Regresson: An ensemble learning method that constructs multiple decision trees during training and outputs the average prediction of individual trees. It handles non-linear relationships well, reduces overfitting through bagging, and provides robust predictions by combining multiple weak learners into a strong predictor.

Linear Regression:A fundamental statistical approach that models the linear relationship between input features and the target variable by fitting a straight line (or hyperplane in multiple dimensions) that minimizes the sum of squared residuals. It assumes a linear correlation between predictors and salary.

Decision Tree Regressor:A tree-based model that recursively splits the dataset based on feature values to create a hierarchical structure of decision rules. It makes predictions by traversing the tree from root to leaf, where each internal node represents a feature test and each leaf node represents a predicted value. Prone to overfitting on complex datasets.

Support Vector Regression (SVR):A regression variant of Support Vector Machines that finds a hyperplane in high-dimensional space that best fits the data within a specified margin of tolerance (epsilon). It uses kernel functions to capture non-linear relationships and is effective for complex, high-dimensional datasets.

3. Experimental Evaluation:

3.1 Methodology:

Salary Prediction Assistant

This module combines AI-powered resume parsing with machine learning-based salary prediction. When a user uploads a PDF resume, Google's Gemini 2.5 Flash model extracts key professional information including job title, experience level (EN/MI/SE/EX), education requirements (Associate/Bachelor/Master/PhD), number of required skills, and years of experience through structured JSON extraction. The user manually inputs additional job-specific details such as employment type (FT/PT/CT/FL), company location, company size (S/M/L), industry, and remote work ratio (0%/50%/100%). All extracted and user-provided features are combined into a pandas DataFrame and fed into a pre-trained Random Forest Regressor model saved as a pickle file. The model predicts the logarithmic transformation of salary, which is then converted back to actual USD salary using the inverse exponential function (np.expm1). The predicted salary is displayed prominently in a styled card format for easy user comprehension.

RAG-Powered AI Chatbot

This module implements a Retrieval-Augmented Generation (RAG) system for intelligent job market queries. The chatbot uses ChromaDB as a vector database to store embeddings of the AI job dataset, generated using Google's text-embedding-004 model through the GoogleGenerativeAIEmbeddings function. When a user submits a query, the system performs semantic similarity search to retrieve the top 15 most relevant documents from the vector store. These documents provide contextual information that is concatenated and passed to Google's Gemini 2.5 Flash language model (via ChatGoogleGenerativeAI) along with the user's question. The LLM generates responses strictly based on the retrieved context, ensuring accuracy and preventing hallucinations. The conversation history is maintained in Streamlit's session state, allowing for persistent chat interactions. User queries and bot responses are displayed in visually distinct chat bubbles using custom CSS styling for enhanced user experience.

Global Job Explorer with Geographic Visualization

This module provides interactive geographic visualization of high-paying job opportunities worldwide using Plotly's scatter_geo mapping functionality. Users can filter jobs by country (including a global "World" view) and specific job titles through dropdown selectors. The system sorts the filtered dataset by salary in descending order and selects the top N jobs (configurable via sidebar, default 10).

Each job's company location is mapped to geographic coordinates (latitude/longitude) using a predefined country dictionary containing center coordinates and map scopes (north america, europe, asia, world). Random jitter (±1.5 degrees) is added to coordinates to prevent overlapping markers for multiple jobs in the same country. The visualization uses bubble size and color gradients to represent salary magnitudes, with appropriate map projections (natural earth for global view, automatic for regional views). Hover tooltips display job title, location, and formatted salary. A complementary ranked data table displays job details including title, country, experience level, and salary with currency formatting. The interface supports dark mode toggle and adjustable result limits through sidebar controls, with the Plotly template dynamically switching between light and dark themes based on user preference.

Loading in raw data

```
import pandas as pd
import numpy as np

def load_data(file_path):
    """
    Load the AI job dataset from CSV file

    Parameters:
        file_path (str): Path to the CSV file

    Returns:
        pd.DataFrame: Loaded dataset
    """
    df = pd.read_csv(file_path)
    print(f"✅ Loaded {df.shape[0]:,} rows | {df.shape[1]} columns")
    print(f"    Columns: {list(df.columns)}")
    return df

# Load dataset
df = load_data("ai_job_dataset.csv")
```

Preprocessing:

```
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestRegressor
```

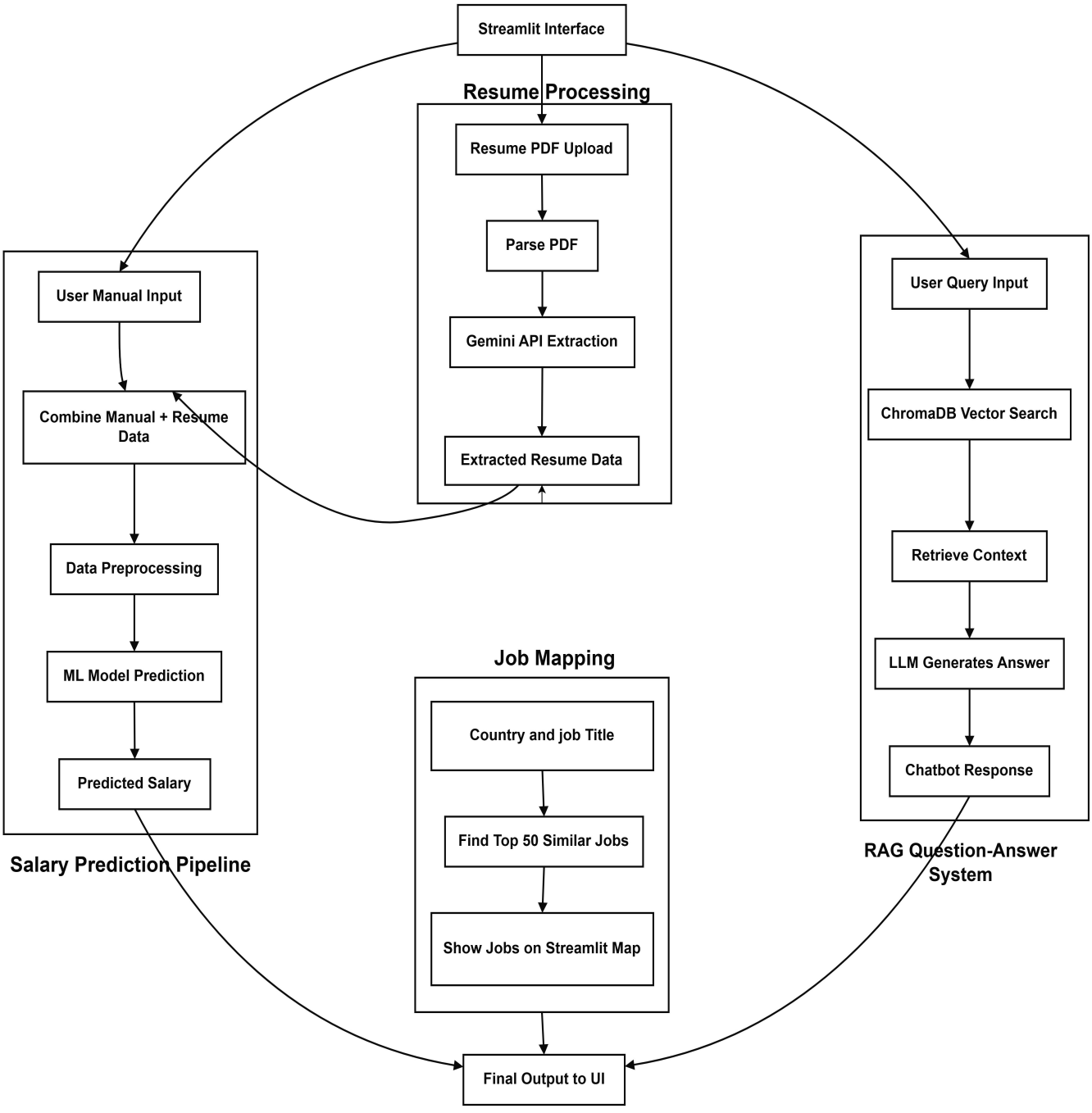
```
import numpy as np

def create_preprocessing_pipeline():
    """
    Create preprocessing pipeline for salary prediction

    Returns:
        ColumnTransformer: Preprocessing pipeline
    """
    # Define feature columns
    nominal_features = ['job_title', 'employment_type', 'company_location',
                        'company_size', 'industry', 'remote_ratio']

    ordinal_features = {
        'experience_level': ['EN', 'MI', 'SE', 'EX'],
        'education_required': ['Associate', 'Bachelor', 'Master', 'PhD']
    }

    numeric_features = ['years_experience', 'num_required_skills']

    # Create transformers
    preprocessor = ColumnTransformer(
        transformers=[
            ('nom', OneHotEncoder(handle_unknown='ignore', sparse_output=False),
             nominal_features),
            ('ord_exp', OrdinalEncoder(categories=[ordinal_features['experience_level']]),
             ['experience_level']),
            ('ord_edu', OrdinalEncoder(categories=[ordinal_features['education_required']]),
             ['education_required']),
            ('num', StandardScaler(), numeric_features)
        ],
        remainder='drop'
    )

    print("✅ Preprocessing pipeline created")
    return preprocessor

# Create pipeline
preprocessor = create_preprocessing_pipeline()
```

RAG Chatbot with ChromaDB

```
from langchain_chroma import Chroma
from langchain_google_genai
import GoogleGenerativeAIEmbeddings, ChatGoogleGenerativeAI
import os

# Set API key
os.environ["GOOGLE_API_KEY"] = API_KEY

def initialize_rag_system(persist_directory="./chroma"):
    """
    Initialize RAG system with ChromaDB and Gemini

    Parameters:
        persist_directory (str): Path to ChromaDB storage

    Returns:
        tuple: (chroma_db, llm)
    """
    # Initialize embeddings
    embedding_function = GoogleGenerativeAIEmbeddings(
        model="models/text-embedding-004"
    )

    # Initialize ChromaDB
    chroma = Chroma(
        persist_directory=persist_directory,
        collection_name="job_dataset",
        embedding_function=embedding_function
    )

    # Initialize LLM
    llm = ChatGoogleGenerativeAI(
        model="gemini-2.5-flash",
        temperature=0.2
    )
```

```
    print("✓ RAG system initialized")
    return chroma, llm

def query_chatbot(chroma, llm, user_query, k=15):
    """
    Query the RAG chatbot

    Parameters:
        chroma: ChromaDB instance
        llm: Language model
        user_query (str): User's question
        k (int): Number of documents to retrieve

    Returns:
        str: Generated answer
    """
    # Retrieve relevant documents
    docs = chroma.similarity_search(query=user_query, k=k)
    context = "\n\n".join([doc.page_content for doc in docs])

    # Create prompt
    prompt = f"""Answer query ONLY based on context.
    If not found, say I don't know.

    Query: {user_query}
    Context: {context}"""

    # Generate answer
    answer = llm.invoke(prompt).content

    print(f"    Query: {user_query}")
    print(f"    Answer: {answer}")

    return answer

# Initialize RAG system
chroma_db, language_model = initialize_rag_system()
```

Flow Diagram :



3.2 Exploratory Data Analysis

This screenshot displays the Salary Prediction Assistant module where users upload their resume (PDF) and input job-specific details including employment type (Full Time), company country (United States), company size (Small), industry (IT), and remote work ratio (Onsite 0%). After processing the uploaded resume (John Doe.pdf), the system successfully extracts relevant information and predicts an annual salary of $96,939.13 USD using the trained Random Forest model. The interface features a clean, dark-themed design with organized input sections and a prominent salary display card.

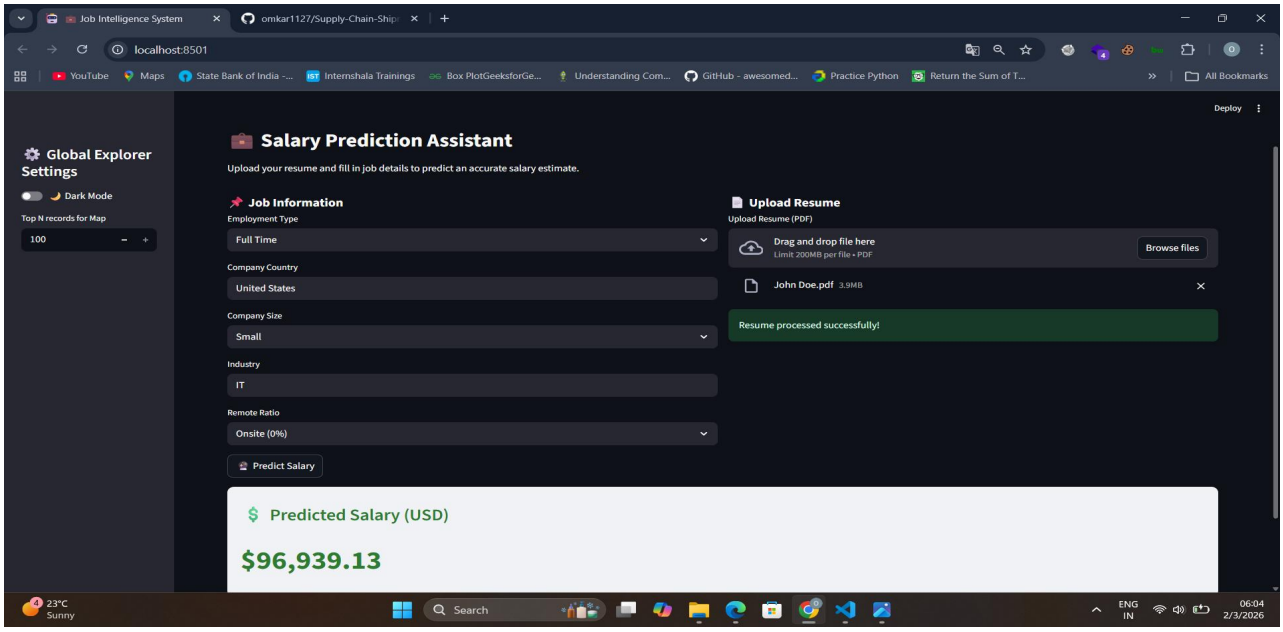


Figure 1: Salary Prediction Interface

This screenshot shows the ranked results table displaying the top 100 highest-paying AI Specialist positions in India. The table presents job data sorted by salary in descending order, showing job titles, countries, experience levels (EX for Executive, SE for Senior), and corresponding salaries in USD. The highest-paying position offers $203,146, while other positions range from approximately $102,790 to $195,462, demonstrating significant salary variations based on experience level and other factors. The dark mode interface provides clear data visualization with proper formatting for currency values.

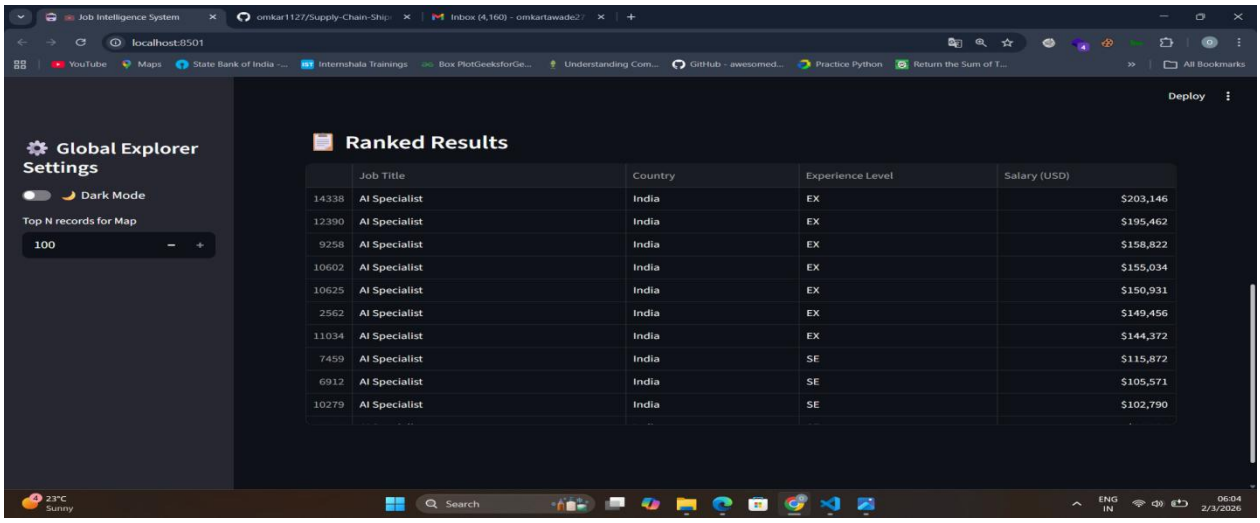


Figure 2: Ranked Job Results Table

This screenshot demonstrates the RAG-powered AI Chatbot responding to a user query: "Top 10 Highest Paying job with experience mentioned also Company name mentioned." The chatbot retrieves relevant information from the ChromaDB vector database and generates a structured response listing the top 10 positions including Research Scientist ($107,893 USD, 9 years, AI Innovations), Data Engineer ($106,078 USD, 0 years, DataVision Ltd), and AI Product Manager ($105,360 USD, 9 years, DeepTech Ventures). The interface uses color-coded chat bubbles—orange/beige for user queries and light blue for bot responses—ensuring clear conversation flow.

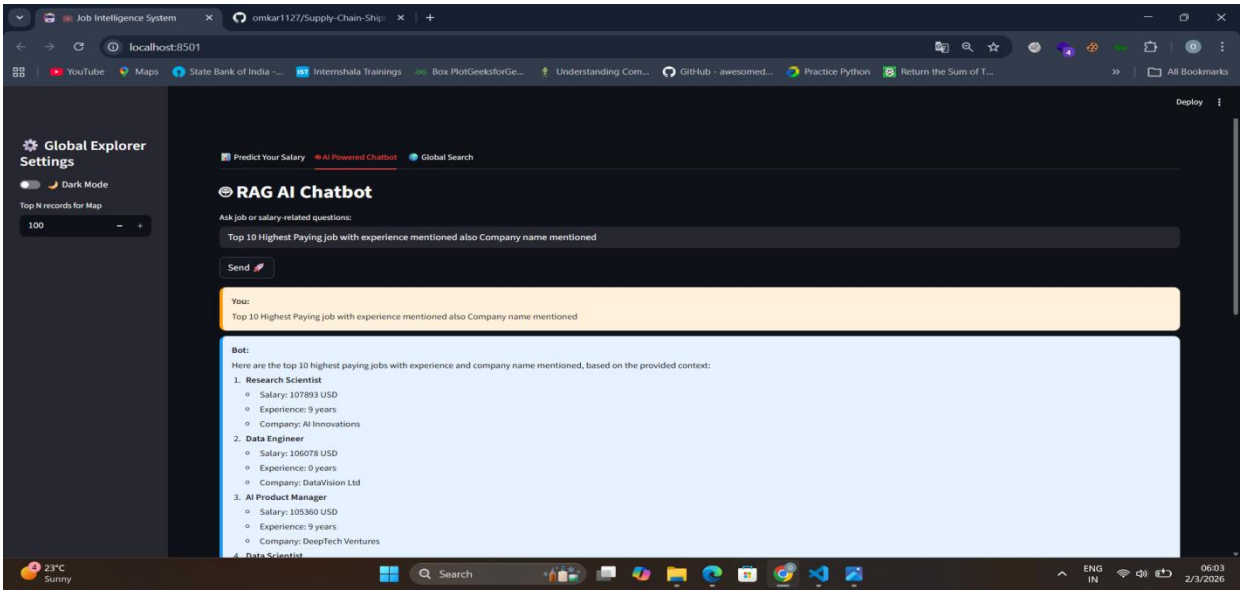


Figure 3: RAG AI Chatbot Interaction

This screenshot presents the Global Job Explorer with an interactive geographic map visualization filtered for AI Specialist positions in India. The Plotly scatter_geo map displays job locations with bubble markers where size and color intensity represent salary magnitude (color scale: 50k to 200k USD). Users can filter results by country (India selected) and job title (AI Specialist selected) using dropdown menus. The map shows "Mapping Top 83 Results" with hover tooltips displaying job title, salary ($23,390 shown), country (India), and years of experience. Below the map, a ranked results table provides detailed tabular data for all filtered positions.

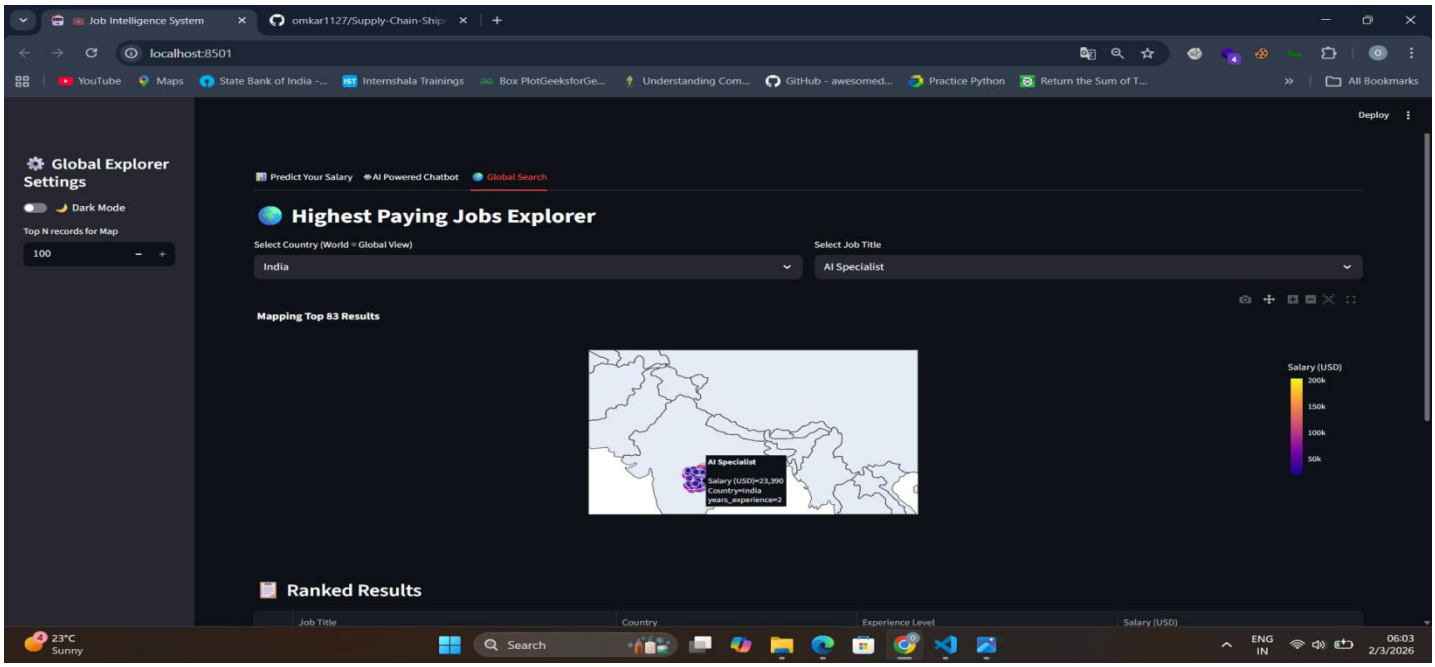


Figure 4: Geographic Visualization Map

4. Results and discussion:

This project successfully developed an AI-powered Job Intelligence System that integrates salary prediction, RAG-based chatbot assistance, and global job market visualization. The system was built using a comprehensive dataset of AI job postings and employs multiple machine learning approaches for accurate salary estimation.

Four machine learning models were trained and evaluated for salary prediction using log-transformed target variables to handle the skewed distribution of salaries:

Linear Regression
MAE: $17,771.68
RMSE: $24,386.25
R^2: 0.8513

Support Vector Machine
MAE: $18,282.72
RMSE: $25,226.95
R^2: 0.8408

Random Forest
MAE: $18,556.45
RMSE: $25,724.67
R^2: 0.8345

Decision Tree Regressor
MAE: $23,831.97
RMSE: $34,171.74
R^2: 0.7080

Linear Regression emerged as the optimal model for deployment, demonstrating superior generalization capabilities with the highest R^2 score of 0.8513, explaining approximately 85% of salary variance with the lowest prediction error. This model was successfully integrated into the final application pipeline for real-time salary predictions.

The system leverages Google's Gemini 2.5 Flash for intelligent resume parsing, automatically extracting key features such as job title, experience level, education requirements, skills count, and years of experience from uploaded PDF resumes. The RAG-powered conversational AI component utilizes ChromaDB vector embeddings with Google's text-embedding-004 model for context-aware responses to job and salary-related queries.

The integrated Streamlit application provides an intuitive three-tab interface: (1) Salary Prediction with resume parsing, (2) AI-powered chatbot for contextual job market queries, and (3) interactive geospatial visualization of high-paying opportunities across 8 major countries including the United States, Canada, India, Germany, United Kingdom, France, Australia, and Switzerland. This comprehensive tool serves as a valuable resource for job seekers and career planners navigating the AI job market.

5. GUI:

The Job Intelligence System features a modern, dark-themed Streamlit web interface with three main tabs: Salary Prediction Assistant, AI Powered Chatbot, and Global Search. The Salary Prediction tab displays a two-column layout with job input fields (Employment Type, Company Country, Company Size, Industry, Remote Ratio) on the left and a drag-and-drop PDF resume uploader on the right. Upon successful resume processing, users receive instant feedback, and clicking "Predict Salary" displays the predicted salary in a prominently styled green card with large, bold text. The left sidebar provides Global Explorer Settings including Dark Mode toggle and customizable map visualization options. The interface emphasizes user-friendliness with clear visual hierarchy, real-time feedback, and intuitive controls suitable for both technical and non-technical users.

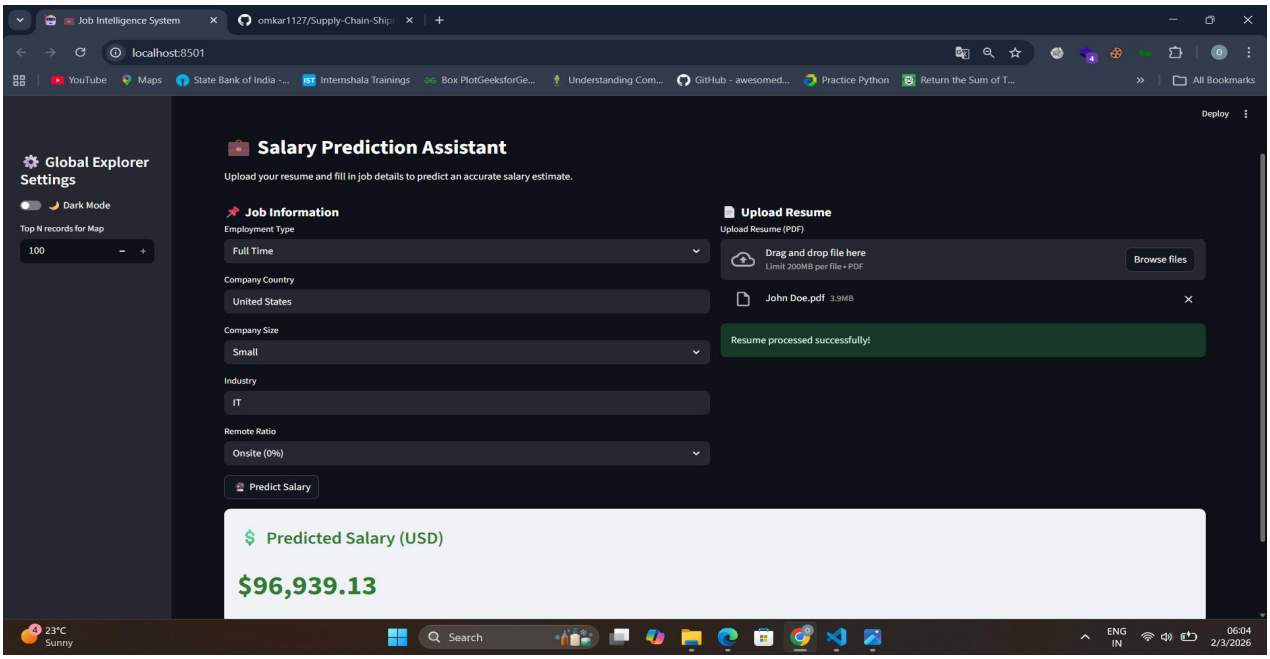


Figure 1: Job Intelligence System - Salary Prediction Module User Interface

6. GitHubLink:

7.Future work And Conclusion

7.1Future Work:

Future enhancements to the Job Intelligence System include expanding the dataset to incorporate global job markets beyond the current 8 countries, enabling more comprehensive salary predictions worldwide. Integration of real-time job scraping APIs from platforms like LinkedIn, Indeed, and Glassdoor would ensure up-to-date market insights. The salary prediction model can be improved by implementing advanced ensemble techniques such as XGBoost and LightGBM, and incorporating deep learning approaches for better accuracy. Adding a job recommendation system based on user skills and preferences would provide personalized career guidance. The chatbot functionality can be enhanced with voice interaction capabilities and multilingual support to reach a broader audience. Implementation of user authentication and profile management would allow users to track their salary progression over time and save job searches. Finally, developing a mobile application version would increase accessibility and user engagement, making the system available on-the-go for job seekers worldwide.

7.2 Conclusion:

- Successfully developed an AI-powered Job Intelligence System integrating salary prediction, conversational AI, and global job market visualization.
- Linear Regression emerged as the best-performing model with R^2 of 0.8513, achieving 85% accuracy in salary prediction.
- Implemented intelligent resume parsing using Google Gemini 2.5 Flash for automated feature extraction from PDF resumes.
- Developed a RAG-based chatbot using ChromaDB and LangChain for context-aware responses to job-related queries.
- Created an interactive Streamlit web application with three core modules: Salary Prediction, AI Chatbot, and Global Job Explorer.
- Integrated geospatial visualization for mapping high-paying job opportunities across 8 major countries.
- The system provides a comprehensive, user-friendly tool for job seekers to make informed career decisions based on data-driven insights.
- Demonstrates the practical application of machine learning, natural language processing, and generative AI in career planning and job market analysis.