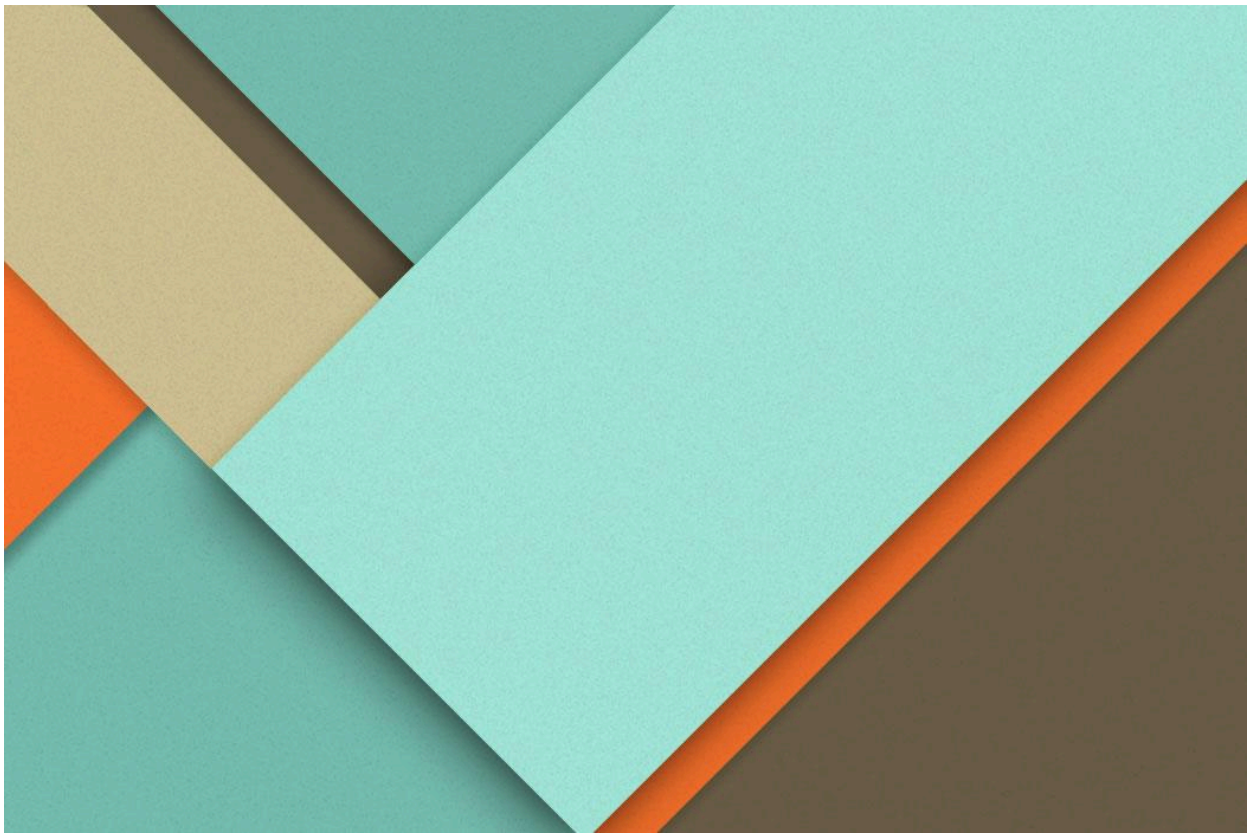

📘 AI Project Report – Module E

AI for Job Market Trend Analysis

Student & Project Details

Student Name: Y.R.Rishidhar Reddy
Mentor Name: Dr. Niranjan Deshpande
Student Code: iitrpr_ai_25010526
GitHub Repo: AI for job trend analysis
App Link: Streamlit Web app

Table of Contents

1. Problem Statement

1.1 Background & Context

The job market—especially AI and technology roles—changes rapidly across industries and countries. Students and professionals often lack **data-backed insights** into which roles are growing and where opportunities are emerging. Job portals provide listings but do not analyze **future trends**.

1.2 Importance of the Problem

Accurate job trend analysis helps:

- Students choose in-demand skills,
- Professionals plan career transition,
- Organizations understand hiring demand,

Without AI-driven analysis, such decisions rely on intuition rather than evidence.

1.3 AI Task Definition

This project performs **predictive trend analysis** using machine learning models to **rank job roles by their potential to grow ("boom")** across different countries and industries.

1.4 Objectives

- Analyze structured job market data,
- Predict a trend score for each job role,
- Rank jobs based on predicted growth,
- Visualize insights through a good dashboard,
- Allow users to **use pre-trained models or train their own models.**

1.5 Assumptions & Constraints

- Historical data reflects meaningful market patterns.
- Predictions indicate relative growth potential, not certainty.
- Ethical use: No personal or sensitive data is used.

2. Approach

2.1 System Overview

The system follows this workflow:

1. The user uploads a job dataset.
2. User selects:
    - A **pre-trained model**, or
    - **Trains a new model** using provided options
3. Data is preprocessed and encoded.
4. The model predicts a **trend score.**
5. Jobs are ranked and visualized using Streamlit.

2.2 Data Strategy

2.2.1 Dataset Size: 15,000 rows, 19 columns

2.2.2 Key Features Used:

- Job Title
- Salary (USD)
- Company Location
- Experience Level
- Employment Type
- Company Size
- Education Required
- Years of Experience
- Industry
- Posting Date

2.2.3 Preprocessing Steps

- Handling categorical features using encoding
- Creating time-based features from posting date
- Feature selection based on relevance
- Train-test split

2.3 AI / Model Design

2.3.1 Pre-trained Models Used

The system includes multiple trained regression models:

1. Linear Regression
2. Random Forest Regressor
3. Gradient Boosting Regressor
4. Extra Trees Regressor
5. XGBoost Regressor
6. LightGBM Regressor

Each model was trained in **Google Colab**, evaluated, and saved as a `.pkl` file.

2.3.2 Train-Your-Own-Model Feature (Key Enhancement)

The system also allows users to **train their own model** using the uploaded dataset.

Supported training options include:

```
train_model_classes = {
    "Random Forest": RandomForestRegressor(n_estimators=100, random_state=42),
    "Linear Regression": LinearRegression(),
    "Gradient Boosting": GradientBoostingRegressor(n_estimators=100, random_state=42),
    "Extra Trees": ExtraTreesRegressor(n_estimators=100, random_state=42)
}
```

User Flow:

- Upload dataset
- Select model type
- Train model inside the application
- Automatically save the trained model
- Use it for prediction and visualization

2.3.3 Feature Names Handling(`feature_names.pkl`)

A critical component of the system is the `feature_names.pkl` file.

Why `feature_names.pkl` is Important:

- Stores the exact feature order used during model training.
- Ensures consistency between training and prediction.
- Prevents feature mismatch errors during inference.
- Maintains reliability across different datasets and models.

Without this file:

- Models may receive incorrect feature mappings.
- Predictions become invalid or crash.

2.3.4 Tools & Technologies

- Programming Language: Python
- Data Processing: Pandas, NumPy
- Machine Learning: Scikit-learn, XGBoost, LightGBM
- Visualization: Matplotlib, Seaborn
- Dashboard: Streamlit
- Model Storage: Pickle (`.pkl`)
- Development: Google Colab, VS Code
- Version Control: GitHub

2.3.5 Design Decisions

- Regression models used to generate **trend scores.**
- Ensemble models preferred for robustness.
- Feature name tracking introduced for deployment stability.
- Modular design to support retraining and model replacement.

3. Key Results

3.1 Working Prototype

- Streamlit dashboard running locally
- Features:
    - Dataset upload
    - Model selection
    - Custom model training
    - Job ranking
    - Interactive visualizations

3.2 Model Performance Summary

Model	RMSE	R^2
Linear Regression	1.35	0.04
Random Forest	0.82	0.65
Gradient Boosting	0.89	0.58
Extra Trees	0.83	0.64
XGBoost	0.82	0.65
LightGBM	0.86	0.61

Selected Model: Random Forest

Secondary Option: XGBoost (similar performance, scalable)

3.3 Observations

- Ensemble models consistently outperform linear models
- Trend-based ranking provides actionable insights

3.4 Limitations

- No live job data
- Predictions are relative, not absolute
- Depends on historical patterns

4. Learnings

4.1 Technical Learnings

- Learned why keeping features consistent is critical for correct predictions.
- Understood how to compare models and choose the best one.
- Learned to handle and fix common deployment errors.

4.2 System & Design Learnings

- Understood how to separate training from inference and deployment to avoid feature mismatches and runtime errors.
- Learned the importance of keeping training and prediction pipelines consistent by preserving feature order and structure.
- Gained hands-on experience building clear, user-friendly AI dashboards that present complex model outputs intuitively for non-technical users.

4.3 Challenges Faced

- Feature mismatch errors.
- Streamlit control-flow issues.
- Model training issues.
- Over-Fitting Problems.

4.4 Future Improvements

- Time-series forecasting.
- Convert trend scores into categorical labels (Booming / Stable / Declining).
- Cloud deployment.
- Skill recommendation functionality.

5. References & AI Usage Disclosure

5.1 Datasets

- Main DataSet - Global AI Job Market & Salary Trends 2025
- DataSet 2 - AI Job Market Trends
- DataSet 3 - Data Science Jobs

5.2 Tools & Frameworks

- Scikit-learn
- XGBoost
- LightGBM
- Streamlit

5.3 AI Usage Disclosure

AI tools (ChatGPT and Claude) were used for coding, debugging assistance, architectural clarification, and documentation support. All model development, experimentation, and system design decisions were carried out by me.