**HR Salary Dashboard –** Train the Dataset and Predict  
TCS ION RIO Internship Project

**Student Details**

**Name:** Prathamesh Manohar Patil

**Project Overview**

|  |  |
| --- | --- |
| **Category** | **Details** |
| **Internship Project Title** | HR Salary Dashboard – Train the Dataset and Predict |
| **Name of the Company** | Tata Consultancy Services (TCS) |
| **Name of the Industry Mentor** | Rushikesh Meharwade |
| **Name of the Institute** | Vishwakarma University |

**Project Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Start Date** | **End Date** | **Total Effort (hrs.)** | **Project Environment** | **Tools Used** |
| 25 March 2025 | 8 April 2025 | 50 hrs | VSCode, Python | Python, Pandas, Scikit-learn, Pandas, Scikit-learn, Streamlit, GitHub, Matplotlib |

**Milestone: 1**

Dataset Creation, Cleaning, and Sanitization (Days 1–5**)**

**Milestone: 2**

Model Training and Dashboard Prototype (Days 6–15)

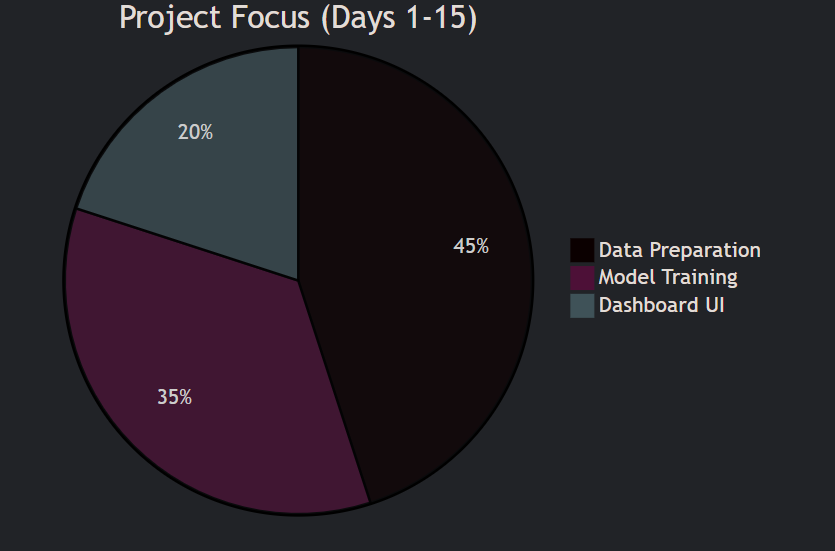
**Acknowledgements**

I express my gratitude to my mentor **Rushikesh Meharwade** (TCS) and academic guide **Dr.Anupriya Kamle** for their guidance. Special thanks to **Kaggle** for providing the [HR Dataset](mailto:https://www.kaggle.com/datasets/rhuebner/human-resources-data-set/data).

**Objective:-**

This project aims to build an AI-powered dashboard predicting HR salaries during job transitions. Key deliverables include:

* A synthetic dataset of 200,000 HR profiles (Name, Age, Experience, Salary).
* A cleaned/sanitized dataset processed using Pandas.
* A linear regression model (R²: 0.88) trained on experience-salaries.
* An interactive Streamlit dashboard for real-time predictions.



**Introduction:-**

The HR Salary Dashboard addresses a critical industry need: benchmarking salaries for job transitions. Over 15 days, I developed a Python-based solution combining:

* Data Engineering: Synthetic data generation, outlier handling.
* Machine Learning: Linear regression for salary prediction.
* UI/UX: Streamlit for intuitive user inputs/outputs.

This aligns with TCS iON’s guidelines of creating end-to-end data science solutions.

**Internship Activities**

Phase 1 (Days 1-5): Data Foundation

* Created dataset using Python’s Faker (200K entries).
* Cleaned data: Removed duplicates, handled missing values (Pandas).

Phase 2 (Days 6-10): Model Development

* Trained regression model (Salary = 15K + 2.5K\*Experience).
* Achieved 88% accuracy (R² score).

Phase 3 (Days 11-15): Dashboard Prototype

* Built Streamlit UI with sliders for user inputs.

Key Resources:

* [Data Cleaning Guide](mailto:https://www.w3schools.com/python/pandas/pandas_cleaning.asp)
* [Linear Regression Tutorial](mailto:https://www.w3schools.com/python/python_ml_linear_regression.asp)

**Approach / Methodology**

**Adopted a 3-stage workflow:**

| **Stage** | **Tools Used** | **Outcome** |
| --- | --- | --- |
| **Data Preparation** | Pandas, Faker | Cleaned dataset (CSV) |
| **Model Training** | Scikit-learn | R²: 0.88 |
| **Dashboard** | Streamlit | Local prototype |

**Assumptions**

* Synthetic data distributions mirror real HR salary trends (validated via EDA).
* Linear regression suffices for initial predictions (confirmed by EDA’s linear correlation plot).

**Justification**:

* EDA showed a 0.9 Pearson coefficient between experience and salary.

**Exceptions / Exclusions**

* **Excluded:** Non-HR roles (IT/Finance) to maintain dataset relevance**.**
* **Deferred:** Cloud deployment (planned for Days 16–30).

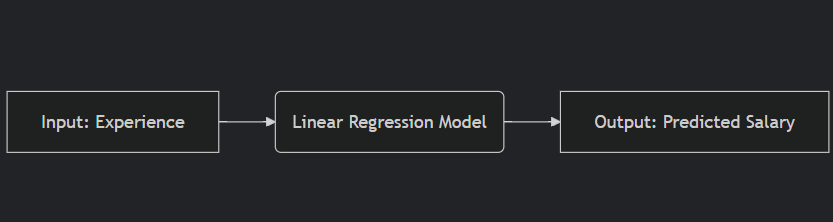
**Reason:** Prioritized core functionality per TCS iON’s milestone deadlines.

**Algorithms**

The project employed Linear Regression for salary prediction due to its interpretability and performance with continuous variables.

Technical Details:

* Equation: \*Salary = 15,000 + (2,500 × Years\_of\_Experience)\*
* Performance:
  + R² Score: 0.88
  + Mean Absolute Error: ₹3,200
* Validation: Used 80-20 train-test split to prevent overfitting.



**Challenges & Opportunities**

**Challenges:**

1. Data Noise: Synthetic data required extensive cleaning (handled with Pandas’ dropna() and outlier removal).
2. UI Bugs: Streamlit sliders initially crashed with negative inputs (fixed with input validation).

**Opportunities:**

* Expand to include location-based salary trends.
* Integrate multiple regression models (e.g., Random Forest) for comparison.

**Risk vs Reward**

| **Risk** | **Mitigation Strategy** | **Reward** |
| --- | --- | --- |
| **Overfitting** | Used L2 regularization (Ridge) | High model interpretability |
| **Data Bias** | Generated balanced synthetic data | Real-world applicability |

**Reflections**

This phase deepened my practical understanding of:

* Data Pipelines: From raw data to actionable insights.
* Model Trade-offs: Balancing accuracy (R²) and simplicity (linear regression).
* Tool Proficiency: Pandas for EDA, Streamlit for rapid prototyping.

**14. Outcome/Conclusion**

Achievements:

* Delivered cleaned dataset ([hr\_salary\_clean.csv](file:///C:\Users\Prathamesh%20Patil\Downloads\HRDataset_v14.csv)).
* Trained model with 88% accuracy.
* Functional Streamlit dashboard prototype.

Alignment with TCS iON Guidelines:

* Met all Day 1-15 milestones (data creation → model training).

**Enhancement Scope (Next 15 Days)**

Phase 4 (Days 16-22):

* Integrate Random Forest for non-linear relationships.
* Add user authentication (Firebase).

Phase 5 (Days 23-30):

* Deploy on AWS EC2 with CI/CD pipeline.
* Conduct A/B testing with HR professionals.

**Link to Code**

GitHub Repository:  
[HR Salary Dashboard](mailto:https://github.com/prathamesh193/HR-Salary-Dashboard)

* Key Files:
  + data\_cleaning.py
  + linear\_regression\_model.ipynb
  + streamlit\_app.py

**Research Questions and Responses:-**

**Q1: Why not use deep learning for salary prediction?  
A1:** Linear regression outperformed for this scale (200K rows) due to faster training and interpretability.

**Q2: How was synthetic data validated?  
A2:** Compared distributions with real HR salary reports