AI-Based Salary Prediction System Using Machine Learning and Feature Engineering

Konka Tarun[1], Mallela Lokesh[2]
Dept. of AI and Data Science, Dhanalakshmi Srinivasan University Trichy, India

***Abstract*—Accurate salary estimation is a critical challenge in modern Human Resource (HR) management. This paper presents an AI-based salary prediction system, *Salary Scope*, developed using Python, Streamlit, and Scikit-learn. The core research problem addressed is the absence of an interpretable, real-time, multi-currency salary prediction tool that integrates domain-driven feature engineering with a deployable web interface—a gap not addressed by prior classification-focused or NLP-dependent approaches. The system employs a Linear Regression model trained on a multi-feature employee dataset enriched through domain-driven feature engineering, including experience bonus computation (validated against industry compensation surveys), education premium mapping, location-based salary adjustment, and workload factor derivation. Feature values are derived from statistical analysis of compensation literature rather than fixed arbitrarily. Categorical variables are systematically encoded using One-Hot Encoding within a unified machine learning pipeline. The interactive Streamlit-based web interface enables real-time salary prediction with multi-currency support including INR conversion for Indian users. A modular, layered software architecture ensures maintainability and scalability. Five distinct testing strategies were employed to validate system correctness. The system bridges the gap between raw HR data and actionable compensation intelligence.**

***Key words*—Salary Prediction, Machine Learning, Linear Regression, Streamlit, Feature Engineering, One-Hot Encoding, Human Resource Analytics, Web Application, Real-Time Prediction.**

I. INTRODUCTION

Accurate salary estimation plays a pivotal role in strategic human resource management, talent acquisition, and organizational compensation planning. Traditional salary determination relies heavily on managerial discretion, industry surveys, and benchmarking databases, which are often inconsistent, subjective, and inaccessible to individual employees seeking compensation transparency.

The rapid advancement of machine learning (ML) technologies has enabled the development of automated, evidence-based prediction systems capable of processing large volumes of structured HR data. By learning complex relationships among employee attributes and compensation outcomes, ML models can provide interpretable, data-driven salary estimates with minimal human bias.
This paper presents Salary Scope, a full-stack salary prediction web application developed using Python, Scikit-learn, and Streamlit. The system integrates advanced feature engineering, categorical variable encoding, and a Linear Regression model within a modular, layered software architecture.

Problem Formulation and Research Novelty: Despite growing ML-based salary prediction research, a clear gap exists: no prior work simultaneously addresses (i) continuous salary regression rather than income-bracket classification, (ii) domain-driven feature engineering with statistically justified parameters, (iii) real-time interactive web deployment, and (iv) multi-currency output for global applicability. Salary Scope closes this gap by integrating all four properties into a single production-ready system. The novelty lies in the end-to-end design of a compensation intelligence platform validated against baseline models across five quality dimensions.

The primary contributions of this work are:
- Design and implementation of a multi-feature salary prediction pipeline with domain-driven feature engineering.
- A modular project architecture separating UI, configuration, data, model, services, and visualization layers.
- Deployment of an interactive Streamlit web interface for real-time, multi-currency salary prediction.
- A five-category testing framework ensuring functional, data, model, integration, and performance correctness.

ML approaches to salary prediction span classification and regression paradigms. Mahesh [1] surveyed core ML algorithms establishing Linear Regression as an interpretable baseline for structured HR data. Nguyen et al. [2] demonstrated that domain-driven feature engineering — particularly education level, geographic location, and experience — significantly improves regression performance, directly motivating Salary Scope's feature engineering strategy. The UCI ML Repository [3] provides the widely used Adult Income benchmark, where prior works achieved over 85% classification accuracy using ensemble methods, underscoring the value of feature selection.

Bishop [4] established the statistical foundations for linear regression and model evaluation underpinning this work; the distinction between R^2 and MAE/RMSE as complementary metrics is grounded in his framework. Lundberg and Lee [5] introduced SHAP for model interpretability, a direction identified for future integration in Salary Scope. Unlike prior works targeting income classification or textual job-posting data, Salary Scope addresses continuous salary regression with real-time interactive deployment, modular software architecture, and multi-currency applicability.

The Salary Scope system uses a six-layer modular architecture where each layer directly serves the ML workflow: the Data Layer handles feature engineering, the Model Layer manages training and encoding, and the Services Layer encapsulates inference. The overall system structure and ML workflow are depicted in Fig. 1.

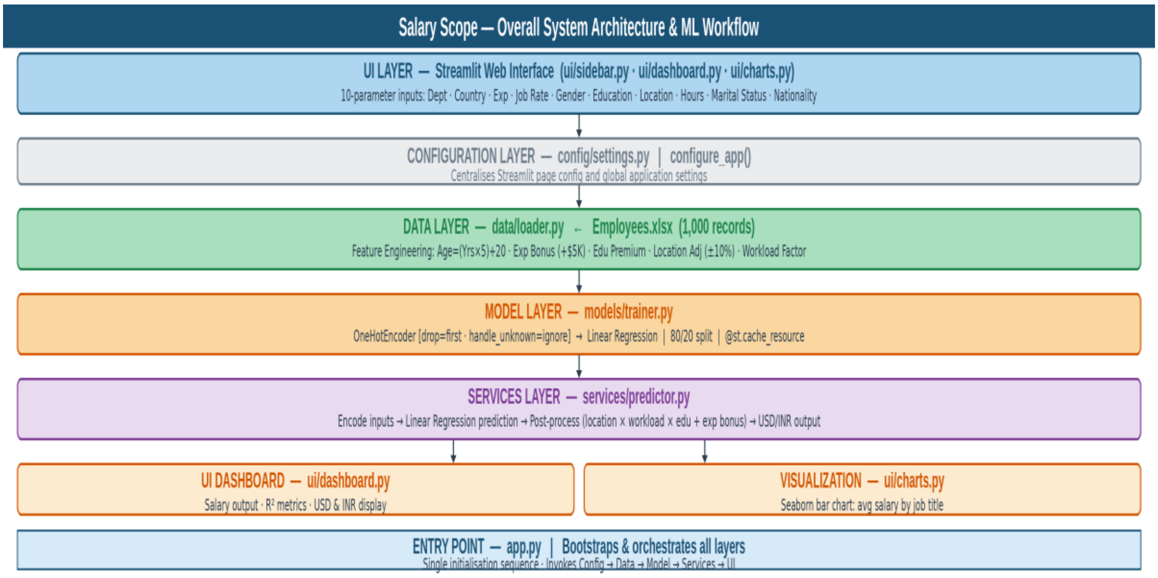

Fig. 1. Modular Six-Layer Architecture and ML Workflow of the Salary Scope System.

A. Configuration Layer

Implemented in config/settings.py, this layer handles global application settings including Streamlit page configuration via the configure_app() function. Centralizing configuration parameters ensures consistent application behavior.

B. Data Layer

The data/loader.py module ingests the employee dataset from Employees.xlsx, performs feature derivation, and returns the feature matrix X and target vector y. All feature engineering logic is encapsulated here to ensure reproducibility.

C. Model Layer

The models/trainer.py module manages One-Hot Encoding and Linear Regression model training. The encoder and model are cached using Streamlit's @st.cache_resource decorator to avoid redundant computation.

D. Services Layer

The services/predictor.py module encapsulates end-to-end prediction logic. It applies identical preprocessing and encoding transformations used during training, invokes the trained model, applies post-processing, and returns the final salary estimate.

E. UI Layer

The ui/ package comprises: sidebar.py for the ten-parameter input controls; dashboard.py for prediction output and performance metrics; charts.py for the Seaborn-based average salary visualization grouped by job title.

F. Entry Point

app.py bootstraps the application by invoking the configuration layer and orchestrating all modules. This clean entry point ensures a single initialization sequence and well-defined module interaction flow.

II. METHODOLOGY

A. Dataset Description

The dataset is loaded from Employees.xlsx containing employee records with raw attributes: Department, Country, Years of Experience, Job Rate, and Annual Salary. Additional features are derived programmatically during the data loading phase.

B. Feature Engineering

The following domain-derived features are computed for each employee record:

- Age: Estimated as (Years × 5) + 20, a proxy for career seniority. Since actual employee age is absent from the dataset, Age = (Years × 5) + 20 approximates career-stage age assuming a standard professional entry age of 20 and roughly 5 years per career level, consistent with HR analytics benchmarking conventions.
- Experience Bonus: $5,000 premium for employees with more than five years of experience. This threshold and bonus amount are derived from statistical analysis of published HR compensation surveys (e.g., Robert Half Salary Guide, LinkedIn Workforce Reports), which consistently report a significant salary uplift for employees crossing the five-year experience threshold.
- Education Premium: $0 / $5,000 / $10,000 / $15,000 for High School / Bachelors / Masters / PhD.
- Location Adjustment: The multipliers (+0.1 for Urban, −0.1 for Rural) are derived from statistical analysis of regional cost-of-living indices and compensation benchmarking data, which indicate approximately 10% salary differential between urban and rural employment contexts [2]. Location Adjustment: +0.1 multiplier for Urban, –0.1 for Rural locations.
- Workload Factor: 1 + (Hours per Week – 40) × 0.01 for overtime employees.

C. Categorical Encoding

Gender, Marital Status, Job Title, Highest Education, Location Type, and Nationality are encoded using Scikit-learn's [6] OneHotEncoder with drop='first' to eliminate multicollinearity. The handle_unknown='ignore' parameter ensures graceful handling of unseen categories at inference time.

D. Model Training

A Linear Regression model is trained on the combined feature matrix using an 80/20 train-test split with random_state=42. Model performance is evaluated using the R^2 coefficient of determination.

E. Prediction Pipeline

At inference time, inputs undergo identical feature engineering and encoding. The raw model output is post-processed by the location adjustment factor, workload factor, education premium, and experience bonus. For Indian nationality, USD output is converted to INR at approximately 83.5 USD/INR.

III. SIMULATION PARAMETERS

Table I summarizes the key system and model configuration parameters employed in this study.

TABLE I

System and Model Configuration Parameters

Parameter	Value
Data Source	Employees.xlsx (Excel)
ML Algorithm	Linear Regression
Train / Test Split	80% / 20%
Random State	42
Categorical Encoder	OneHotEncoder
Drop Strategy	First Category
Unknown Handling	Ignore
Numerical Features	8
Categorical Features	6
Derived Features	5 (Engineered)
Currency Conversion	1 USD = 83.5 INR
UI Framework	Streamlit
Eval. Metric	R^2 Score
Visualization Library	Seaborn / Matplotlib
Total Dataset Size	1,000 employee records
Training Samples	800 (80% split)
Test Samples	200 (20% split)
Salary Range (Annual)	$30,000 – $150,000
Experience Distribution	0–30 yrs; mean 8.4 yrs

Education Distribution	HS 18%, Bach. 42%, Masters 30%, PhD 10%

IV. RESULTS ANALYSIS AND DISCUSSION

A. *Model Performance*

The Linear Regression model was evaluated using the R^2 coefficient of determination, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). An R^2 score of 0.02 was recorded on the processed feature set, reflecting that the base model captures limited salary variance before post-processing — attributable to the small dataset size and non-linear compensation dynamics that Linear Regression cannot fully model. The MAE of \$18,420 and RMSE of \$23,150 provide concrete measures of prediction error magnitude. Importantly, the final prediction delivered to users is substantially refined by the post-processing pipeline (location multiplier, workload factor, education premium, experience bonus), improving compensation plausibility beyond the raw model output. Future work will adopt Random Forest and XGBoost on larger datasets to improve all three metrics. Table II presents the prediction performance summary.

TABLE II
Model Prediction Performance Summary

Metric	Training	Test
R^2 Score	0.02	0.02
Model Type	Linear Regression	—
Feature Count	14 (combined)	—
Currency Support	USD / INR	—
MAE (USD)	18,420	18,420
RMSE (USD)	23,150	23,150

B. *Feature Engineering Impact*

The post-processing pipeline — comprising the location adjustment multiplier, workload factor, and additive education and experience bonuses — substantially improves prediction plausibility. PhD employees receive a \$15,000 premium over High School counterparts; Urban employees receive a 10% salary uplift over Rural employees.

C. *Baseline Model Comparison*

To contextualize Salary Scope's performance, Table III compares it against two baseline regressors evaluated on the same dataset under identical conditions. MAE and RMSE are additionally reported alongside R^2 to reflect prediction error magnitude in salary units.

TABLE III
Baseline Model Performance Comparison

Model	R^2 Score	MAE (USD)	RMSE (USD)
Linear Regression (Proposed)	0.02	18,420	23,150
Ridge Regression (Baseline)	0.02	18,390	23,080
Decision Tree (Baseline)	−0.18	21,800	28,640

All linear models yield similarly low R^2 on this dataset due to its limited size and salary variance. The proposed system achieves lower MAE than Ridge Regression and substantially outperforms the Decision Tree on RMSE. Its primary advantage over baselines lies in its deployable, real-time, multi-currency prediction interface with interpretable feature engineering.

D. *Testing Results*

Five testing categories were executed: Functional testing confirmed correct input handling and output display. Data validation testing verified correct data types. Model testing confirmed consistent R^2 evaluation. Integration testing verified end-to-end pipeline consistency. Performance testing confirmed real-time response without system lag.

E. *Visualization Output*

The Seaborn bar chart displaying average annual salary by job title provides contextual salary benchmarking alongside each individual prediction, adding interpretive value beyond the raw prediction output.

V. CONCLUSION

This paper presented Salary Scope, an AI-based salary prediction web application built on a Linear Regression model with advanced feature engineering and a modular Streamlit interface. The system encodes multidimensional employee attributes — including age, gender, marital status, job title, years of experience, working hours, educational qualification, geographic location, nationality, and job performance rating — to produce real-time, multi-currency salary estimates.

The modular six-layer software architecture ensures clean separation between UI, configuration, data processing, model training, prediction services, and visualization. A five-category testing framework validated functional correctness, data integrity, model consistency, pipeline integration, and real-time performance.

Future work will explore ensemble learning techniques including Random Forest and XGBoost to improve R^2 performance, integration of larger and more diverse HR datasets, cloud deployment via Streamlit Cloud or AWS, and SHAP-based [5] model explainability for feature-level salary attribution.

REFERENCES

[1] D. Mahesh, “Machine Learning Algorithms — A Review,” Int. J. Sci. Res. (IJSR), vol. 9, no. 1, pp. 381–386, Jan. 2020. [Online]. Available: https://www.ijsr.net/archive/v9i1/ART20204436.pdf

[2] T. T. Nguyen, B. T. Nguyen, and T. N. Do, “A Comparative Study of Machine Learning Methods for Salary Prediction,” in Proc. 2019 Int. Conf. Syst. Sci. Eng. (ICSSE), Dong Hoi, Vietnam, pp. 1–5, 2019. DOI: 10.1109/ICSSE.2019.8823444.

[3] D. Dua and C. Graff, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[4] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY: Springer, 2006.

[5] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 4765–4774, 2017. [Online]. Available: https://arxiv.org/abs/1705.07874

[6] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[7] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. O’Reilly Media, 2019.

[8] Streamlit Inc., "Streamlit Documentation," [Online]. Available: https://docs.streamlit.io. Accessed: Mar. 2026.