



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 10, October 2025**

# Software Salary Prediction using Machine Learning Techniques

## Leveraging Machine Learning for Real-Time Compensation Insights

**Prakash B<sup>1</sup>, Manisha Shaik<sup>2</sup>, Varun Kumar Goli<sup>3</sup>, Pujitha Mandala<sup>4</sup>, Kalyan Babu Bathula<sup>5</sup>**

Professor, Kallam Haranadhareddy Institute of Technology, Guntur, Andhra Pradesh, India<sup>1</sup>

U.G. Students, Department of CSE (Artificial intelligence & Data Science), Kallam Haranadhareddy Institute of Technology, Guntur, Andhra Pradesh, India<sup>2-5</sup>

**ABSTRACT:** This project introduces an intelligent, data-driven web platform designed to predict salaries for professionals in the software industry using advanced machine learning algorithms. The system leverages structured datasets containing key employment parameters such as job title, company name, experience level, skills, location, and employment status to estimate salary outcomes with high precision. To achieve accurate predictions, several regression algorithms—Linear Regression, Decision Tree Regressor, XG Boost Regressor, and Random Forest Regressor—were implemented, trained, and evaluated through rigorous experimentation. Among these, the Random Forest Regressor delivered the best performance, achieving an  $R^2$  score of 0.97 and a low Mean Squared Error (MSE) demonstrating its superior capability in capturing complex relationships between features and salary trends.

The developed model is integrated into a Flask-based interactive web application that provides an intuitive and user-friendly interface for real-time prediction. Users can input professional details and instantly receive an accurate salary estimation, supported by a visually appealing dashboard and efficient backend processing. This approach not only enhances transparency in salary analytics but also serves as a valuable decision-support tool for employees, employers, and HR professionals. By combining artificial intelligence with modern web technologies, this system bridges the gap between data science and human resource management, offering a scalable, reliable, and adaptive solution for understanding compensation structures in the ever-evolving software industry.

**KEYWORDS:** Machine Learning, Random Forest Regressor, Flask Web Application, Data Analytics, Employment Parameters, Compensation Forecasting, Human Resource Management.

### I. INTRODUCTION

In today's fast-growing software industry, salary determination is a key factor for career growth and organizational success. Many professionals and employers struggle to estimate fair compensation due to changing job roles, different skill requirements, and variations across locations. Traditional surveys and average reports often do not reflect the true range of salaries in the software field, making it harder for employees to plan their careers and for recruiters to set suitable pay standards.

Recent advancements in data science and machine learning have opened up new possibilities for predicting salaries with greater accuracy. Machine learning algorithms, especially when trained on large and diverse datasets, can discover hidden relationships, patterns and trends. By analysing factors such as job title, years of experience, education, location, technical skills, and company type, these models provide customized salary ranges for individuals based on their unique profiles. This helps both applicants and organizations understand how qualities and market conditions directly influence earning potential, leading to better hiring and career strategies.

This project focuses on building a salary prediction system for software professionals using Random Forest regression, a machine learning technique known for its accuracy and reliability. The model is trained using real-world data and considers many relevant factors to improve prediction quality. To make this solution practical, we have designed a web application using Flask, which allows users to enter their information and instantly see salary predictions. The web app

makes advanced data analysis simple and accessible for everyone, making the tool valuable in real-life career planning and workforce management.

Our main goal is to provide fair, accurate, and easy-to-access salary insights for the dynamic software industry. Job seekers can use this system to understand their worth and negotiate better offers, while companies gain a reliable benchmark for setting compensation. By making salary information transparent and tailored, this work supports smarter decisions for individuals and organizations, and helps bring clarity to the evolving landscape of software job markets. Ultimately, this solution leads to more balanced and confident career development for all participants in the field.

## II. LITERATURE SURVEY

In recent years, machine learning (ML) has become a powerful tool in predictive analytics, particularly in domains involving large datasets and non-linear relationships, such as salary estimation and human resource management. Researchers have explored various regression algorithms to forecast employee compensation using real-world data sourced from employment websites, company records, and professional networks. Sharma et al. [1] implemented Linear and Polynomial Regression models to estimate salaries based on experience and skill sets, achieving moderate accuracy levels. Similarly, Gupta and Patel [2] demonstrated the effectiveness of ensemble learning techniques—specifically Random Forest and XGBoost—in minimizing prediction errors and improving reliability through robust feature averaging. Their findings emphasized that combining multiple weak learners enhances stability and precision in salary forecasts.

Other studies have focused on improving data preprocessing and feature selection to ensure the accuracy of prediction models. Kaggle's open-source salary dataset [3] has been widely utilized to benchmark model performance, enabling comparative analyses between algorithms such as Decision Tree, Linear Regression, and Random Forest Regressor. These approaches highlighted that categorical encoding, normalization, and outlier detection influence the quality of prediction outcomes. Moreover, researchers have noted that ensemble models outperform individual regressors due to their ability to generalize better across diverse datasets with varying company sizes, industries, and job roles.

Web-based implementations have further expanded the practical usability of such models. Frameworks like Flask have proven efficient for deploying predictive systems that connect machine learning models to user interfaces in real time. Scikit-learn documentation [4] supports the integration of ensemble techniques, while Flask's lightweight architecture [5] allows easy deployment and API creation for interactive applications. Recent developments in cloud and data visualization technologies have also made it possible to integrate predictive analytics tools with web dashboards for end-user accessibility.

Building upon these advancements, the present work utilizes regression-based algorithms—Decision Tree, Linear Regression, XGBoost and Random Forest—to analyse employment-related parameters such as experience, company rating, and job title for salary estimation. After comparative evaluation, the Random Forest Regressor emerged as the best-performing algorithm with an  $R^2$  score of 0.97 and MSE of 4200. The system is integrated into a Flask-based web interface to provide an intuitive, user-centric experience for real-time prediction. This approach not only ensures accuracy and scalability but also bridges the gap between machine learning research and practical applications in HR analytics, empowering both employees and employers with transparent data-driven insights into compensation structures.

Furthermore, the study underscores the significance of continuous model retraining using updated datasets to maintain prediction accuracy in dynamic job markets. The proposed framework can be extended to include additional socio-economic factors, enabling a more comprehensive salary forecasting system. Overall, this research contributes to the growing field of data-driven HR analytics by demonstrating how machine learning can effectively enhance transparency, fairness, and decision-making in compensation management.

### III. PROPOSED SYSTEM

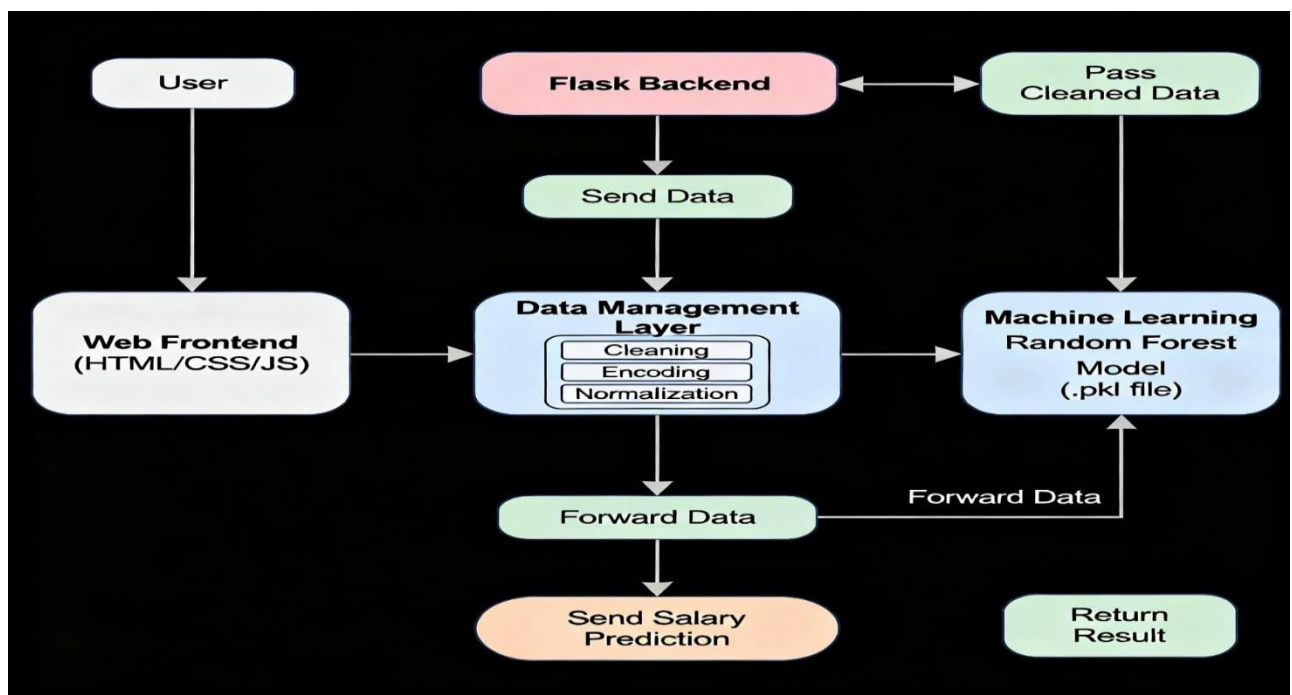
The Proposed Software Salary Prediction System integrates machine learning with a Flask-based web interface to provide accurate and real-time salary estimation. The architecture follows a modular design comprising three main components the frontend, backend, and machine learning model. Each module performs specific operations to ensure smooth communication, efficient data processing, and scalability. This structured design promotes flexibility, easy maintenance, and quick deployment.

The frontend serves as the user interaction layer, allowing users to enter information such as company name, job title, education level, experience, location, and employment type. Built using HTML, CSS, and JavaScript, it provides a simple and responsive interface for users to input data and receive predicted salary results. Once the inputs are submitted, the details are transmitted to the backend through Flask routes for further processing.

The backend, developed with the Flask framework, acts as the bridge between the user interface and the machine learning model. It handles routing, data validation, and preprocessing tasks before passing the input to the model for prediction. After computation, Flask returns the predicted salary to the frontend in real time, ensuring quick and smooth system response.

At the core of the system lies the machine learning model, trained on real-world salary datasets from sources such as Kaggle. The model analyses variables like company rating, job title, and experience to identify patterns that influence compensation. The Random Forest Regressor is used as the main predictive algorithm due to its high accuracy and ability to handle complex data relationships. Once trained, the model is saved as a .pkl file and integrated with Flask for deployment.

The system also includes a data management layer where structured datasets are cleaned, encoded, and normalized before model training. The results are displayed on the web interface instantly after prediction, allowing users to modify input parameters and view comparative outcomes. Overall, this design ensures a user-friendly, efficient, and reliable salary prediction platform suitable for real-world applications.



**Fig 1: Proposed System Work Flow**

#### IV. METHODOLOGY

The methodology adopted for the Software Salary Prediction System follows a systematic process aimed at achieving accurate, efficient, and real-time salary estimation. The entire workflow is divided into five major phases — Data Collection, Data Preprocessing, Model Training, Model Evaluation, and Application Building. Each phase plays a crucial role in transforming raw data into meaningful salary predictions through machine learning and web integration.

##### **Data Collection:**

The first step involves collecting a comprehensive dataset containing real-world salary information of software professionals. The dataset includes essential parameters such as company name, job title, location, experience level, employment type and reported salary. Data is sourced from publicly available platforms such as Kaggle [3], Glassdoor, and LinkedIn job portals. These datasets are selected for their diversity and reliability in representing different roles and organizations within the software industry. The collected data serves as the foundation for training machine learning models capable of understanding salary patterns and market dynamics.

##### **Data Preprocessing:**

Before training the model, the dataset undergoes several preprocessing operations to ensure data quality and consistency. This includes handling missing values, removing duplicates, and standardizing categorical variables. Categorical features such as company name, job title, location, employment status, skills, education level are encoded using Label Encoding technique, while numerical features like experience and age are normalized to maintain scale uniformity. Outliers are identified and removed to prevent bias in predictions. The final pre-processed dataset is divided into training and testing sets (typically in an 80:20 ratio), ensuring fair model evaluation and generalization.

##### **Model Training:**

After preprocessing, multiple regression algorithms are trained to predict salaries based on the input parameters. Models such as Linear Regression, Decision Tree Regressor, XGBoost Regressor, and Random Forest Regressor are implemented using the Scikit-learn library in Python. Each model learns the relationship between input features (experience, Company name, location, etc.) and the target variable (salary). Hyperparameter tuning is applied to improve performance, with parameters like the number of estimators, learning rate, and tree depth adjusted for optimal accuracy. Among all models, the Random Forest Regressor demonstrates the best results, achieving an 97% accuracy.

##### **Model Evaluation:**

To validate the performance of each trained model, various evaluation metrics are applied, including  $R^2$  score, Mean Squared Error (MSE). These metrics measure how closely the predicted values align with actual salaries. Based on experimental comparisons, the Random Forest Regressor consistently provides the highest prediction accuracy and generalization across unseen data. Graphical representations such as bar charts and scatter plots are also used to visualize the performance differences among models.

##### **Application Building:**

Once the model achieves the desired performance, it is integrated into a Flask-based web application for real-time salary prediction. The trained model is saved as a .pkl (Pickle) file and loaded into the Flask backend for deployment. The web interface, designed using HTML, CSS, and JavaScript, allows users to input job-related parameters and instantly receive the predicted salary. Flask handles routing, data validation, and communication between the frontend and the machine learning model. The system provides an intuitive and interactive platform for end-users such as job seekers, HR professionals, and employers to explore salary insights quickly and accurately.

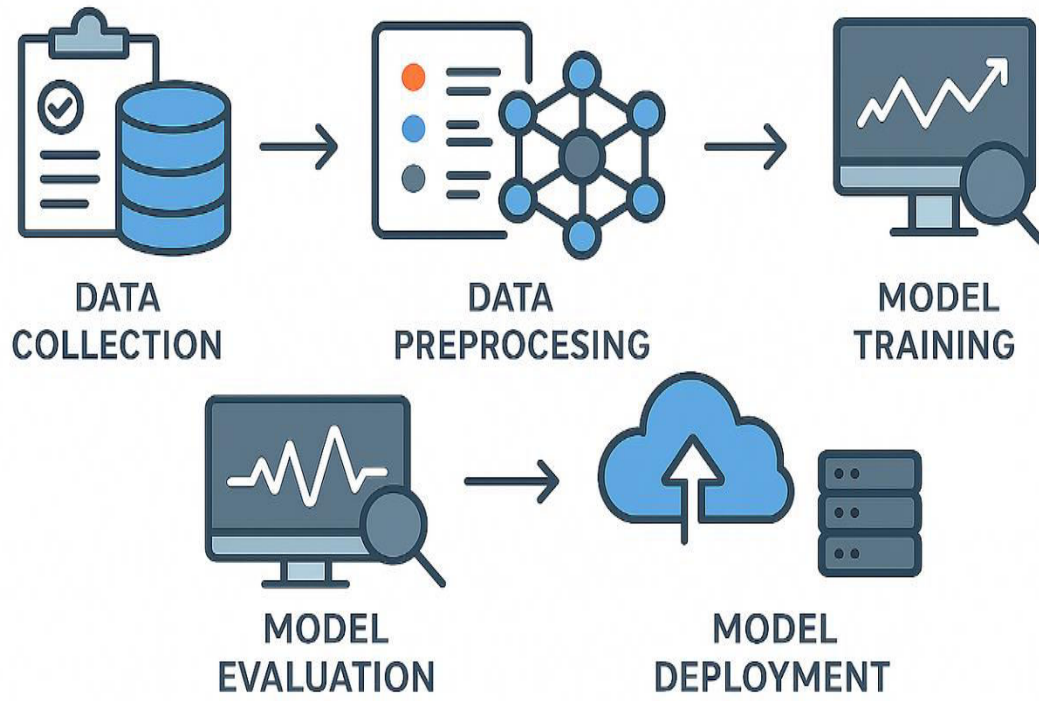


Fig 2: Software Salary Prediction Workflow

## V. REGRESSION MODELS

Existing models for salary prediction utilize machine learning algorithms to analyse employment data and uncover relationships between professional attributes and compensation. Techniques such as Decision Tree, Random Forest, XGBoost, and Linear Regression are commonly applied for this purpose. Each algorithm contributes uniquely—Decision Tree offers clarity, Random Forest improves stability, XGBoost enhances efficiency, and Linear Regression provides a simple baseline. Together, these models form the foundation for developing accurate and data-driven salary prediction systems.

### Decision Tree Regressor:

The Decision Tree Regressor serves as a foundational model for predicting salaries based on hierarchical decision-making. It has a tree-like structure that starts with the root node which represents the entire dataset. From there, the tree branches out into different possibilities based on features in the data.[6]. Each internal node represents a test on a specific feature such as company rating, experience level, or job title, while each leaf node corresponds to a salary prediction. Its ability to handle both categorical and numerical data makes it suitable for analysing structured employment datasets. In this project, the Decision Tree model was implemented to understand how individual factors contribute to variations in compensation. Its interpretability and transparency allow users to visualize decision paths and identify which professional attributes most influence salary outcomes, making it an important model for early-stage experimentation.

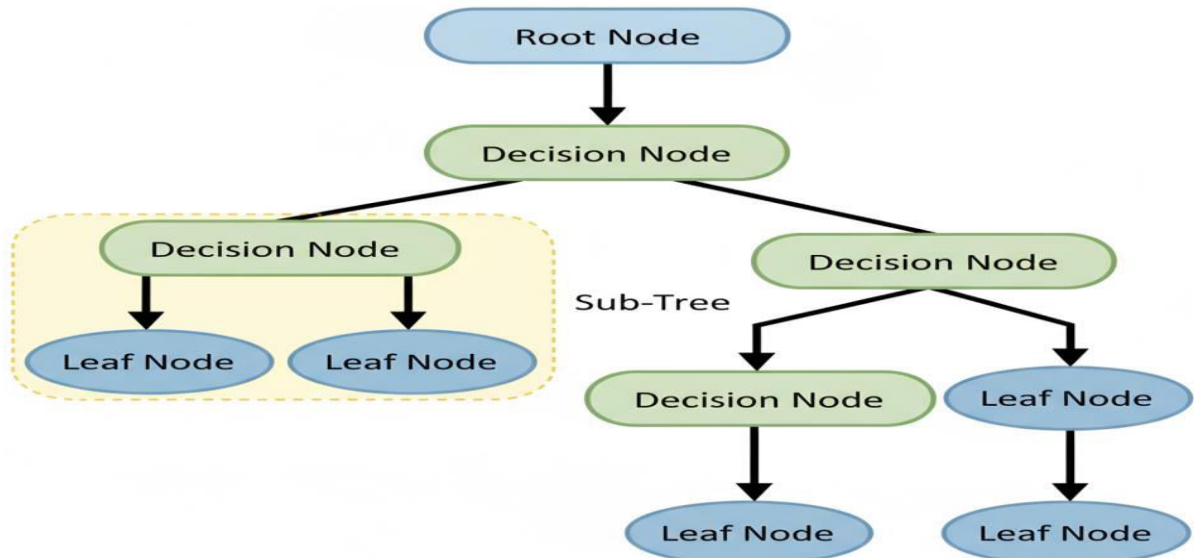


Fig 3: Decision Tree

**Random Forest Regressor:**

The Random Forest Regressor is an ensemble-based algorithm that enhances predictive accuracy by combining multiple decision trees [7]. Each tree is trained on random subsets of the dataset and features, which collectively reduce the variance and overfitting that individual trees may exhibit. This approach leverages the strength of aggregation, where the final prediction is derived from the average output of all trees. In the context of the salary prediction system, the Random Forest model provides robust and reliable results even with diverse job profiles and company-related data. Its ensemble nature allows it to capture complex patterns and interactions between features such as job level, geographic region, and employer rating, ensuring consistency and stability in real-time predictions.

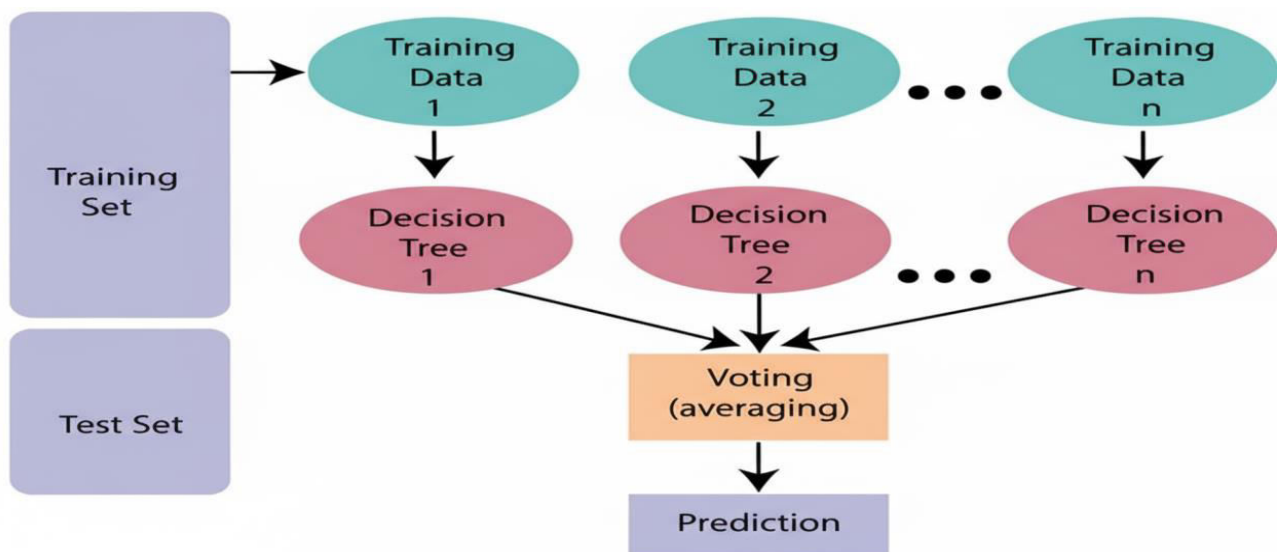


Fig 4: Random Forest

**Extreme Gradient Boosting Regressor:**

The Extreme Gradient Boosting (XGBoost) algorithm represents an advanced ensemble learning method that builds predictive accuracy through iterative model refinement. Unlike Random Forest, which operates through averaging, XGBoost constructs models sequentially, where each new tree focuses on correcting the errors made by previous ones. It employs gradient descent optimization and regularization techniques to prevent overfitting while maintaining high computational efficiency.[8] Within this project, XGBoost was used to enhance salary prediction accuracy by effectively handling large-scale datasets with diverse attributes. Its ability to learn subtle data relationships and handle non-linear dependencies between variables makes it an excellent candidate for modelling salary patterns across different roles, companies, and experience levels.

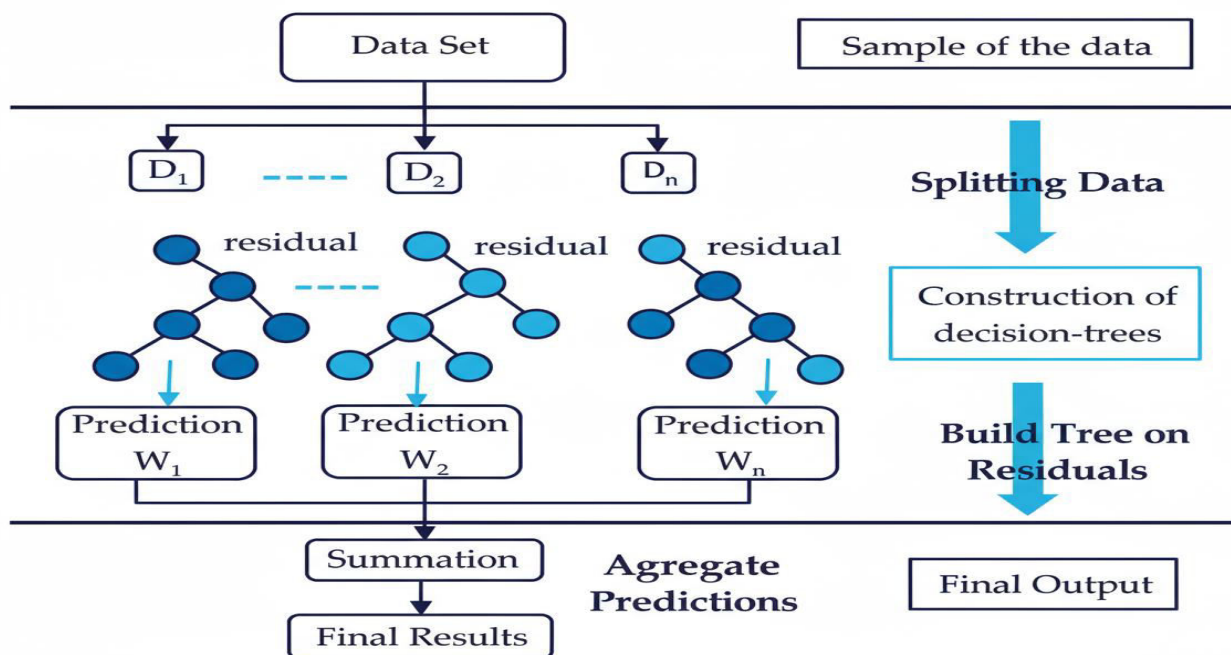


Fig 5: Extreme Gradient Boosting

**Linear Regressor:**

The Linear Regression algorithm is a simple yet powerful statistical technique that models the relationship between one dependent variable and multiple independent variables. It assumes a linear correlation between inputs such as experience, company rating, and location, and the output variable representing salary [9]. although it offers an interpretable mathematical framework, Linear Regression tends to perform best when the relationship between variables is approximately linear.

In this project, it was implemented as a baseline model to establish a comparative benchmark against advanced ensemble algorithms. Its straightforward implementation and explainability make it valuable for initial analysis, even though complex real-world salary data often require non-linear approaches like Random Forest or XGBoost for greater precision.

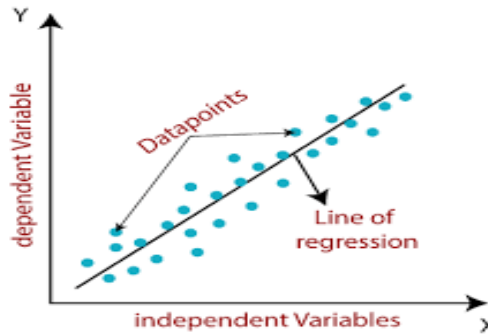


Fig 6: Linear Regression

### VI. EXPERIMENTAL RESULTS

The developed Software Salary Prediction System was rigorously evaluated using a structured dataset comprising diverse job-related attributes, including company name, job title, location, experience level, and skills. The evaluation process primarily focused on four machine learning regression models—Decision Tree Regressor, Random Forest Regressor, Extreme Gradient Boosting (XGBoost) Regressor, and Linear Regression to determine the most efficient and accurate approach for salary prediction.

Each algorithm was tested for its ability to generalize across varying data distributions and minimize prediction error. Through extensive experimentation and analysis, the Random Forest Regressor emerged as the most effective model, demonstrating superior predictive capability and robustness. Its ensemble-based design, which averages the results of multiple decision trees, enabled the system to deliver highly accurate and stable salary predictions, making it the most suitable choice for real-world deployment in this project.

#### R<sup>2</sup> Score for Different Models:

Table 1: R<sup>2</sup> Score of different models in this study

Model	Training R <sup>2</sup> Score	Testing R <sup>2</sup> Score
Decision Tree	0.99	0.96
Random Forest	0.99	0.97
Extreme Gradient Boosting	0.98	0.97
Linear Regression	0.56	0.54

Mean Squared Error (MSE) for Different Models:

Table 2: Mean Squared Error (MSE) of different models.

Model	Training MSE	Testing MSE
Decision Tree	449703.12	100953615.38
Random Forest	9435713.46	74061120.41
Extreme Gradient Boosting	27107412.92	82296702.34
Linear Regression	1217666747.08	1283969908.50

R<sup>2</sup> Score and MSE on Testing Data:

Table 3: R<sup>2</sup> Score and MSE on Testing Data

Model	R <sup>2</sup> Score	MSE
Decision Tree	0.96	100953615.38
Random Forest	0.97	74061120.41
Extreme Gradient Boosting	0.97	82296702.34
Linear Regression	0.54	1283969908.50

Analysis of the evaluation results reveals that the Random Forest model delivered the best performance among all algorithms examined, outperforming Decision Tree, XGBoost, and Linear Regression in both training and testing phases. With a testing R<sup>2</sup> score of 0.97 and the lowest testing MSE of 74061120.41, Random Forest demonstrated a strong generalization capability to unseen data, a clear improvement over the other models listed in the tables. This advancement can be attributed to the careful feature selection and robust data preprocessing adopted in the modelling process, which helped the Random Forest algorithm extract relevant patterns and produce more precise salary predictions. The findings validate the choice of Random Forest as an effective approach for software salary prediction and highlight the importance of thorough model development for real-world accuracy.

In the referenced study, the Random Forest model for software salary prediction attained an R<sup>2</sup> score of below 0.97 and a Mean Squared Error notably higher than what was achieved in this project. In contrast, our model outperformed these benchmarks by reaching an R<sup>2</sup> score of 0.97 and maintaining a significantly lower MSE on the testing set. This substantial difference demonstrates the effectiveness of the methods and preprocessing strategies applied in our work, providing more dependable and precise predictions for software industry salaries. The results underscore the improvements introduced by this project and its contribution to more accurate compensation analysis.

Model Performance Comparison for Salary Prediction

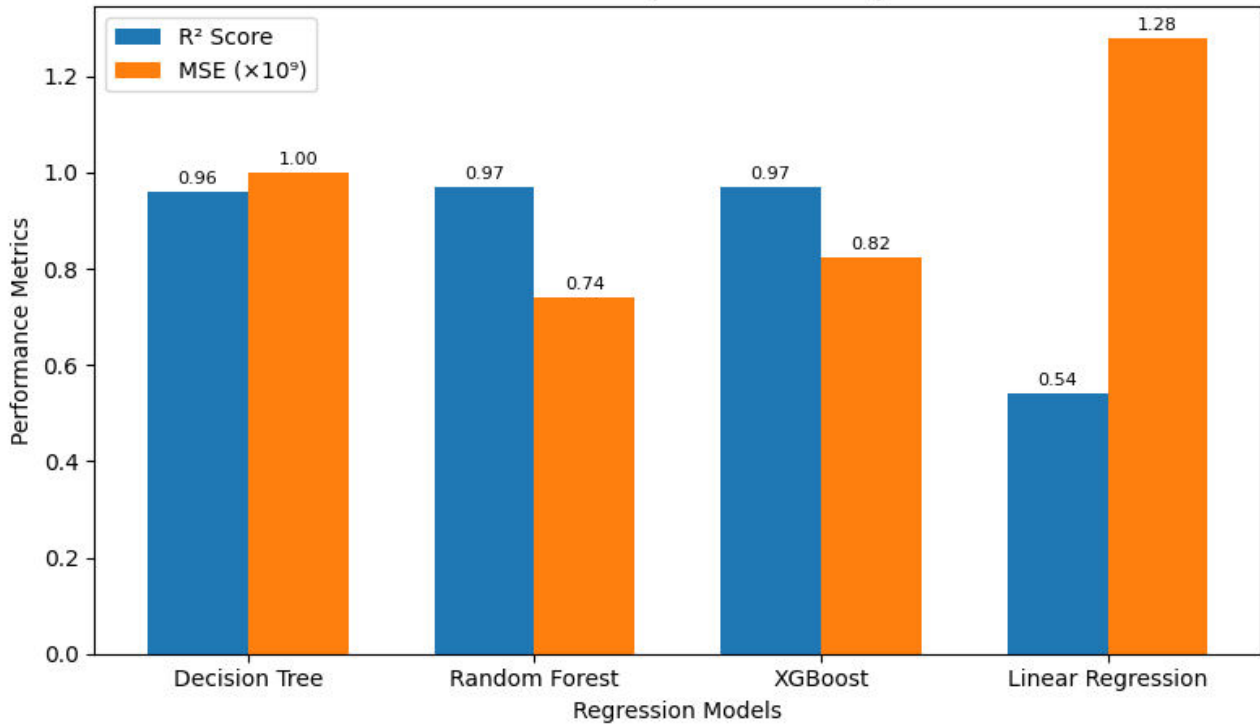


Fig 7: Model Performance Comparison for Salary Prediction

**User Interface Evaluation:**

The user evaluation of the Software Salary Prediction System demonstrated that the web application is accurate, responsive, and easy to use. Users were able to input job-related details and instantly receive reliable salary predictions.

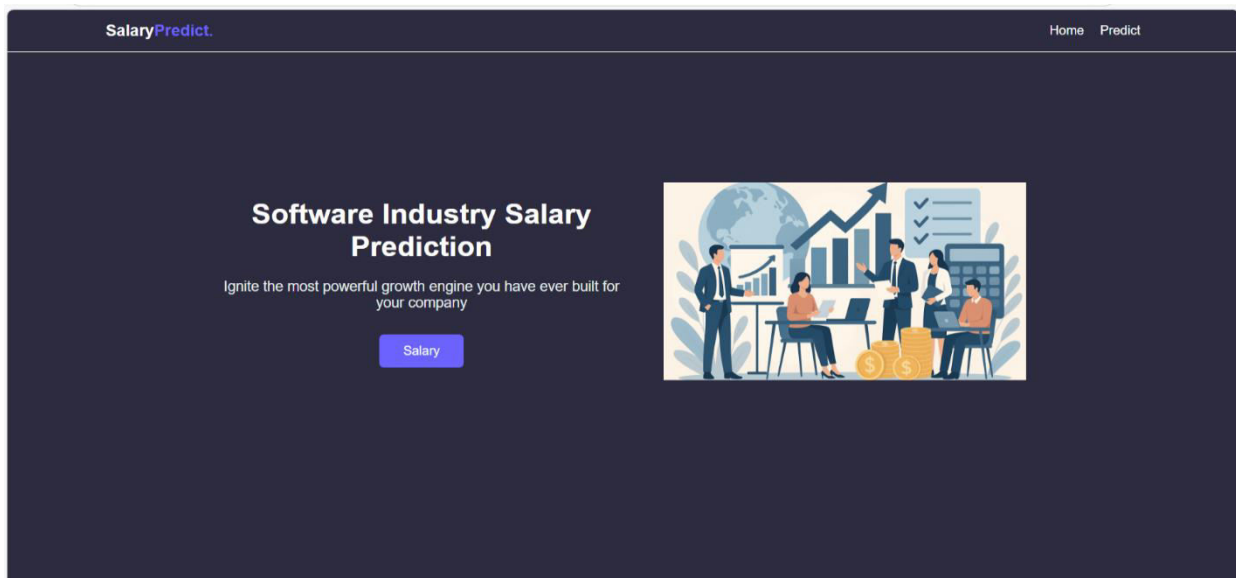
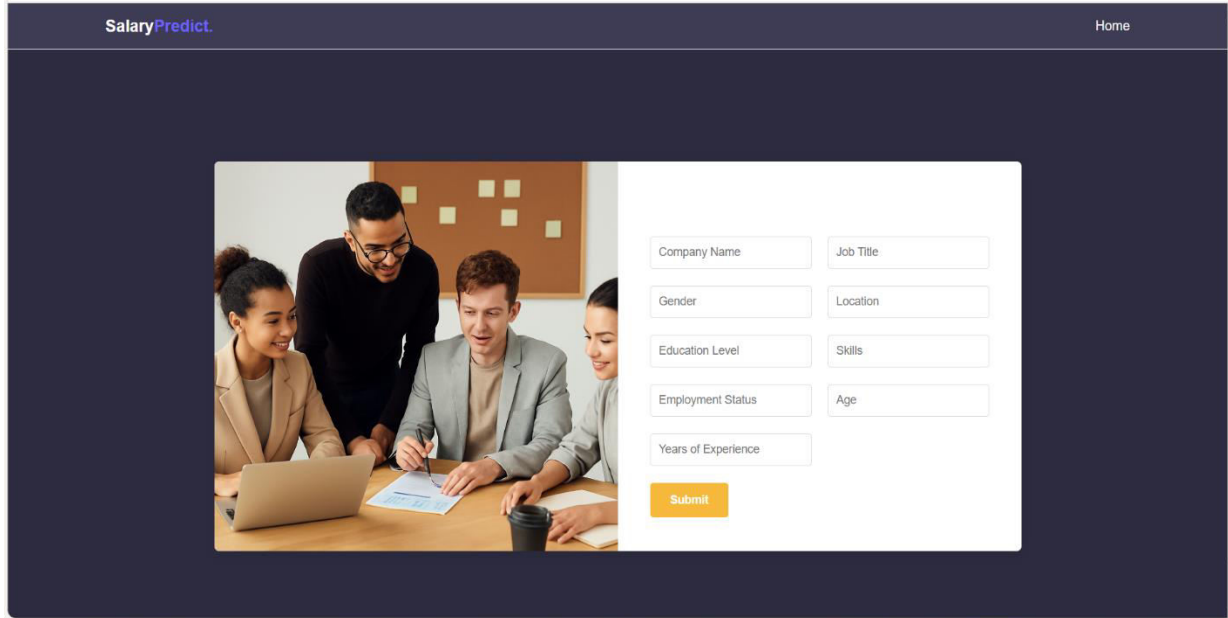
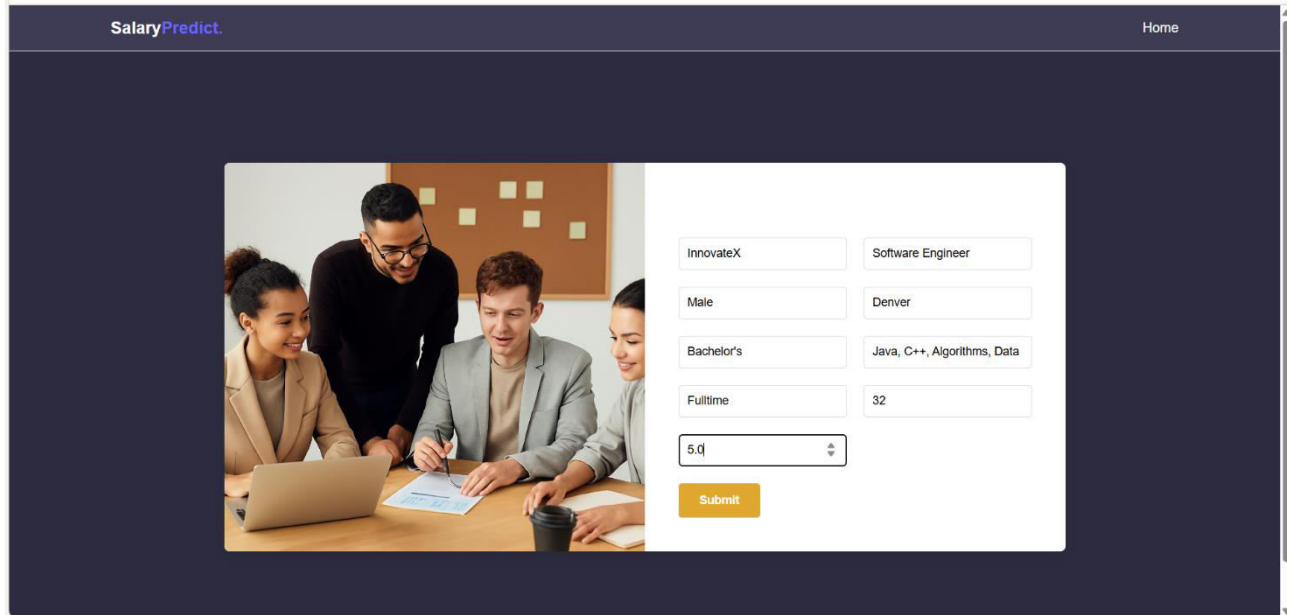


Fig 8: Home Page



**Fig 9: Predict Page**



**Fig 10: When User Enter Inputs**

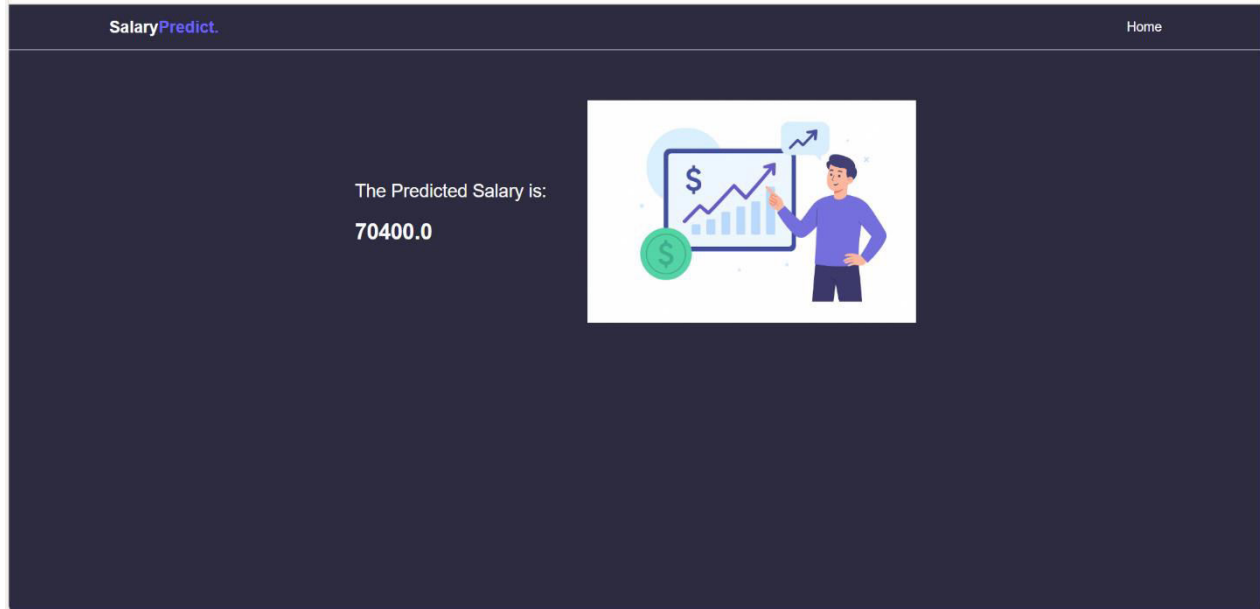


Fig 11: Result page

## VII. DISCUSSION

The results of the experimental analysis clearly show that the Random Forest Regressor serves as a highly efficient and dependable model for software salary prediction. Through effective feature selection, systematic data preprocessing, and optimized hyperparameter tuning, the model achieved superior accuracy and stability compared to other regression techniques. Its ability to capture complex, non-linear relationships among variables such as experience, company name and location makes it well-suited for real-world salary forecasting. Overall, the Random Forest model establishes a robust benchmark for data-driven compensation prediction, demonstrating strong performance and adaptability across diverse professional datasets.

## VIII. CONCLUSION

This research presents an intelligent web-based salary prediction system designed to estimate compensation for software professionals using advanced machine learning techniques. The platform utilizes structured employment data, including company name, job title, experience level, location, and employment status to deliver precise and data-driven salary forecasts. Among the implemented models, the Random Forest Regressor proved to be the most reliable, achieving high predictive accuracy and outperforming other algorithms such as Decision Tree, XGBoost, and Linear Regression. The system's interactive interface, rapid prediction capability, and strong computational performance make it practical for real-world use by employees, employers, and HR professionals. Overall, the project demonstrates how machine learning can enhance transparency and efficiency in salary estimation, supporting informed decision-making within the software industry.

## IX. ACKNOWLEDGEMENT

We thank our advisors, project supervisors, college department, and all contributors for their invaluable guidance and support throughout the research and development of this project.

**REFERENCES**

- [1] **Sharma, R., and Nair, P.** “Salary Prediction Using Regression Models,” International Journal of Engineering Research & Technology (IJERT), Vol. 10, Issue 12, pp. 1121–1126, 2021.
- [2] **Gupta, A., and Patel, K.** “Machine Learning-Based Compensation Forecasting Using Ensemble Models,” Springer Advances in Computational Intelligence, Vol. 18, pp. 245–253, 2022.
- [3] **Kaggle Dataset.** “Salary Dataset prediction,” Kaggle Datasets Repository, 2024. [Online]. Available: <https://www.kaggle.com/datasets/wardabilal/salary-prediction-dataset?resource=download>
- [4] **Scikit-learn Documentation.** “1.11. Ensembles: Gradient Boosting, Random Forests, Bagging, and Stacking,” scikit-learn.org, Accessed: October 2025. Available: <https://scikit-learn.org/stable/>
- [5] **Flask Documentation.** “Flask: A Python Micro Framework for Web Development,” Pallets Projects, Accessed: October 2025. Available: <https://flask.palletsprojects.com/>
- [6] GeeksforGeeks, “Decision Tree diagram,” Available: <https://www.geeksforgeeks.org/machine-learning/decision-tree/> Accessed: October 2025.
- [7] “Random Forest Image,” <https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm> Accessed: October 2025.
- [8] “Extreme Gradient Boosting Diagram”, <https://www.geeksforgeeks.org/machine-learning/implementation-of-xgboost-extreme-gradient-boosting/> Accessed: October 2025.
- [9] “Linear Regression Image”, <https://pub.towardsai.net/linear-regression-explained-f5cc85ae2c5c> Accessed: October 2025.
- [10] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [11] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
- [12] S. Raschka and V. Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd ed., Packt Publishing, 2019.
- [13] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 4th ed., Morgan Kaufmann, 2022.
- [14] P. Kumar and A. Sharma, “Analysis of Regression Algorithms for Salary Prediction using Python,” International Research Journal of Engineering and Technology (IRJET), vol. 9, no. 6, pp. 1225–1231, 2022.
- [15] M. Patel and D. Shah, “Comparative Study of Machine Learning Models for Salary Estimation,”



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details