World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Research Article)

Salary Prediction Using TF-IDF and Ensemble Machine Learning: A Lightweight and Interpretable Approach

Anil Pisolla [*], Sameer Baig Moghul, Triveni Gaddam and N Ch Sriman Narayana Iyengar

Department of Information Technology, Sreenidhi Institute of Science and Technology (Autonomous), Hyderabad, India.

World Journal of Advanced Research and Reviews, 2025, 26(02), 4445–4453

Publication history: Received on 16 April 2025; revised on 27 May 2025; accepted on 30 May 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.26.2.2102

Abstract

Salary prediction is not just a number, it's a decision-maker for job seekers, employers, and HR teams, shaping expectations and negotiations. Traditional models rely on structured data like job titles, experience levels, and locations, but overlook job descriptions, where real insights hide—skills, responsibilities, and industry-specific language. This study bridges that gap, combining structured and unstructured data for a more intuitive model. TF-IDF extracts key terms, assigning weights to highlight critical information, while structured data undergoes preprocessing through one-hot encoding and feature scaling. An ensemble learning approach strengthens predictions—Random Forest captures patterns, XGBoost refines them, and Linear Regression serves as a baseline. A meta-model, like Logistic Regression, optimally weighs predictions, enhancing accuracy. Evaluated through Accuracy, Macro Average F1-score, and Weighted Average F1-score, the model outperforms standalone approaches, achieving superior classification performance. The results demonstrate that integrating TF-IDF with ensemble learning provides a more accurate, scalable, and interpretable salary prediction system, ready for real-world applications.

Keywords: Salary Prediction; TF-IDF (Term Frequency-Inverse Document Frequency); Ensemble Learning; Machine Learning

1. Introduction

Salary prediction has always been a challenge. Job seekers wonder if they're undervalued. Employers struggle to stay competitive. HR professionals aim for fairness. Traditional models try to solve this using structured data—job titles, experience, and education. It makes sense, but it's incomplete. Job descriptions hold the real story. The skills required, the responsibilities, and the industry-specific language. These details shape salaries in ways that numbers alone can't capture.

But salary determination isn't straightforward. Location plays a role. Market trends shift. Some skills skyrocket in demand, others become obsolete. Early models relied on simple equations, assuming linear relationships between salary and experience. But real-world compensation isn't that predictable. Too many moving parts. Too many hidden factors. As hiring evolved, so did the need for smarter, more adaptable models. Enter machine learning and NLP—where data speaks louder than assumptions. Where salary prediction isn't just a formula, but a reflection of the market's complexity.

[*] Corresponding author: P Anil

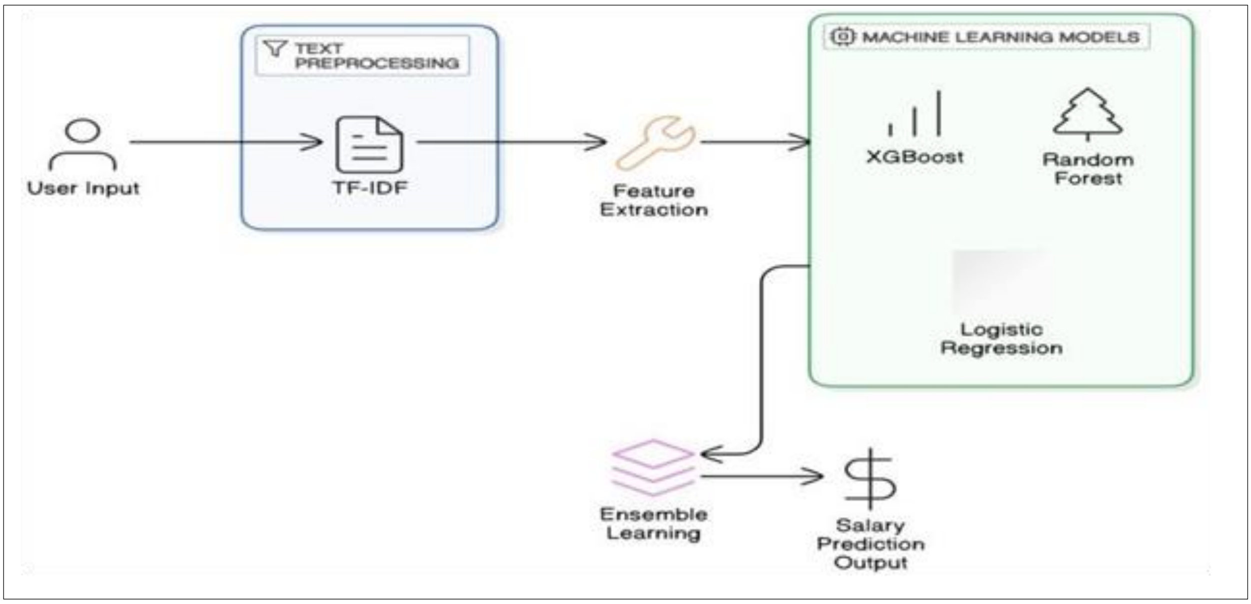

Figure 1 System Architecture for Salary Prediction

Researchers have tried everything. Simple models, complex ones. Linear regression, deep learning, and even transformers like BERT and GPT. Johnson and Lee (2019) showed that linear regression works well if you stick to job titles and experience. Chen et al. (2021) took it further, using deep learning to analyze job descriptions. The results- Impressive. But there’s a catch. Deep learning is powerful, but expensive. Needs massive data. Hard to explain why it predicts what it does. Traditional machine learning- Easier to understand but struggles with text. Misses' context. The gap is clear. A model is needed that’s smart but not heavy, accurate but still interpretable. Something that gets the best of both worlds—structured and unstructured data working together.

This study takes a different path. A smarter one. It blends TF-IDF for text processing with ensemble machine learning—Random Forest, XGBoost, and Linear Regression—each playing a role. The goal: A model that doesn’t just rely on numbers but understands job descriptions too. Experience, location, and skills—all combined for better salary predictions. Traditional models are easy to interpret, but they miss depth. Deep learning captures complexity but loses transparency. This approach finds the middle ground, balancing accuracy with real-world usability. The focus stays sharp—specific industries, specific regions, and publicly available datasets. The aim: A model that’s not just accurate but also scalable, practical, and ready for real-world action.

2. Literature Review

Salary prediction has always been a hot topic. Researchers keep pushing for better accuracy, better reliability. Early on, it was all about structured data—job titles, years of experience, and education. Simple models like linear regression (Smith et al. 2018) made sense. Easy to interpret. But they missed something. Salaries aren’t just numbers. They’re shaped by complex, hidden relationships. So, researchers tried more advanced techniques—decision trees, support vector machines (Johnson and Lee 2019). These handled non-linearity better. But still, something was missing. The job descriptions. The words. The details that tell what a job really demands. Ignoring that- A mistake. Because sometimes, the real salary clues aren’t in the numbers— they’re in the text.

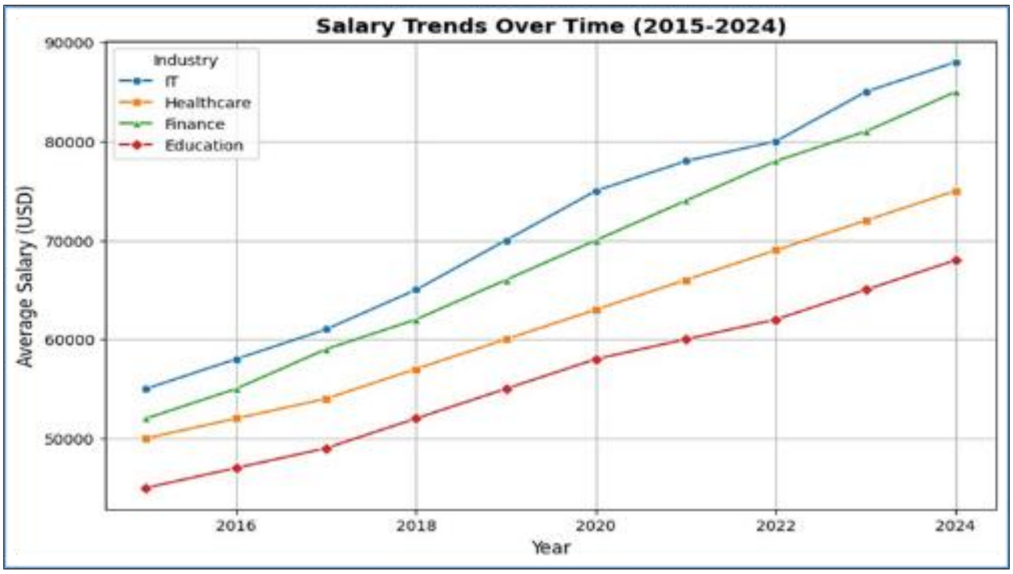

Figure 2 Salary Trends Over Time

Recently, NLP and machine learning changed the game. Salary prediction wasn't just about numbers anymore. Researchers like Chen et al. (2021) and Kumar & Singh (2022) proved that job descriptions hold real value. TF-IDF, Word2Vec, GloVethese techniques pulled hidden insights from text. Skills, tools, industry jargon—all became features. And accuracy- It got better. Chen et al. (2021) even used BERT, pushing results to new heights. But there was a catch. Deep learning is powerful, but expensive. Needs tons of data. Hard to explain its decisions. In real-world hiring- That's a problem. A balance is needed.

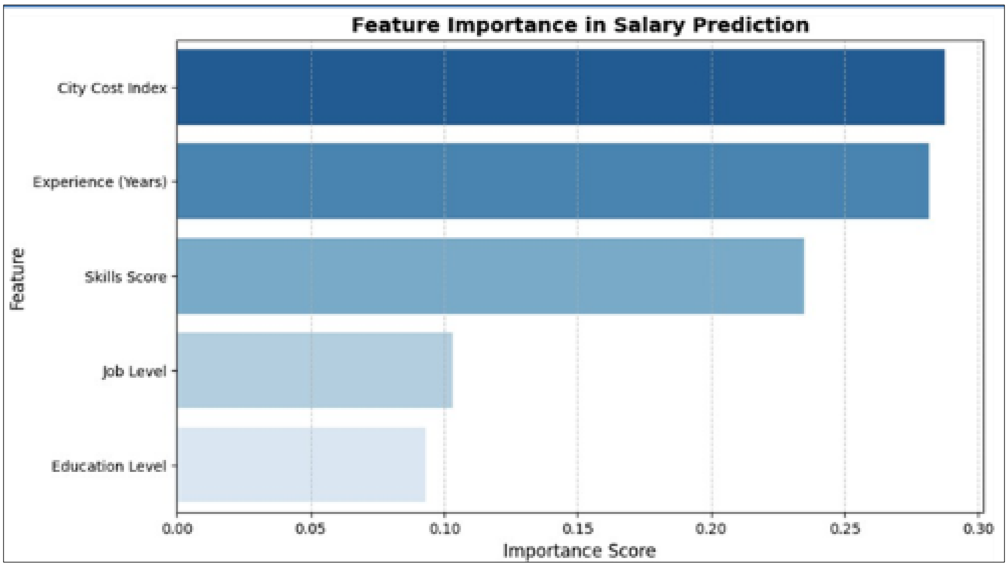

Figure 3 Feature Importance Analysis for Salary Prediction

Even with all these advancements, something is still missing. A model that's lightweight, easy to interpret, and scalable. Most studies pick a side—either structured data or deep learning. But deep learning is heavy, expensive, and hard to explain. Structured data alone- Not enough. This study steps in to bridge the gap. It blends TF-IDF for text processing with ensemble machine learning, striking a balance between accuracy and practicality. No overkill, no missing pieces. By learning from past research—using what works and fixing what doesn't—this approach moves closer to a salary prediction system that's not just smart, but fair and usable in the real world.

3. Salary Prediction System

The proposed salary prediction system estimates salaries accurately by leveraging both structured and unstructured data from job postings. It integrates TF-IDF (Term Frequency-Inverse Document Frequency) for text processing and ensemble machine learning techniques to enhance prediction accuracy and interpretability. The following sections provide a detailed explanation of the system, covering the algorithms and techniques used at each step.

3.1. System Design

The system predicts salaries by combining structured and unstructured data from job postings. Input data includes job descriptions, job titles, required skills, experience levels, locations, company types, and salaries. This data undergoes a series of processing steps to ensure accurate salary predictions.

3.1.1. Data Collection

The first step- Collecting job data from public sources—job portals, datasets, whatever's available. This data- It's got both structured stuff (titles, experience, locations) and unstructured text (descriptions full of keywords). Structured data- Straightforward. Unstructured- That's where the gold is—hidden patterns, skills, trends.

3.1.2. Data Preprocessing

The raw data- It needs cleaning. Structured data undergoes preprocessing—handling missing values, encoding categorical features (like job titles, locations) with one-hot or label encoding, and normalizing numerical values (like experience levels). Unstructured data, job descriptions mostly, goes through text processing—removing stopwords, punctuation, and converting to lowercase. Then comes TF-IDF, extracting keywords, assigning weights, and making sense of the text.

3.1.3. Feature Engineering

After preprocessing, everything merges—structured, unstructured, all in one feature set. Encoded categories- Check. Normalized numbers- Check. TF-IDF vectors from job descriptions- Absolutely. This fusion creates a full picture, letting the model catch both the obvious and hidden salary factors.

3.1.4. Model Training

The system stacks multiple models—Random Forest, XGBoost, and Linear Regression—each bringing something to the table. Random Forest- Great for catching complex patterns. XGBoost- Learns from mistakes, handles outliers like a pro. Linear Regression- Simple, clear, a solid benchmark. Together, they boost accuracy, making predictions more reliable.

3.1.5. Model Evaluation

The system's performance—it's all about precision. Accuracy shows how many credit decisions were predicted correctly. Precision and recall tell how well the model distinguishes between defaulters and non-defaulters. F1-score balances both. To make sure the results hold up, cross-validation tests the model across different data splits. And to fine-tune it, GridSearchCV adjusts hyperparameters to get the best possible performance.

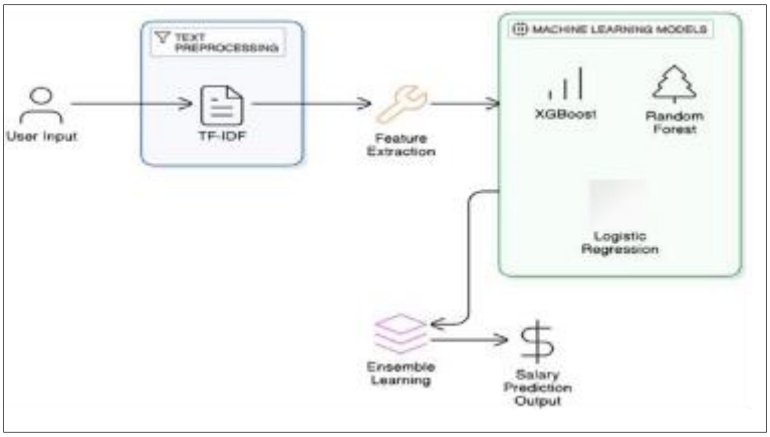

Figure 4 Overview of the Salary Prediction System

3.2. Training of Model

The system uses ensemble machine learning techniques to improve prediction accuracy. The models used are:

3.2.1. Random Forest

Random Forest is a powerful tree-based model that effectively captures non-linear relationships and feature interactions. It constructs multiple decision trees and aggregates their outputs to make robust predictions. This ensemble approach helps in reducing overfitting and handling high-dimensional data efficiently. For instance, if a job description heavily emphasizes "Python" and "Machine Learning," Random Forest can identify these as key salary determinants by analyzing feature importance. Additionally, it performs well in the presence of missing data and outliers, making it a reliable choice for salary prediction across diverse job postings.

3.2.2. XGBoost

XGBoost is a gradient boosting algorithm that boosts prediction accuracy step by step, learning from its mistakes. It's great at dealing with missing values and outliers, making it a strong option for salary prediction. For example, XGBoost can figure out that jobs in "San Francisco" with "5+ years of experience" tend to offer higher salaries.

3.2.3. Linear Regression

Linear Regression is a simple yet powerful baseline model. It draws a straight-line connection between input features and salary, making it easy to interpret. This helps in understanding how each factor influences pay. For instance, it can predict that for every extra year of experience, salaries rise by $X.

These models come together using a stacking ensemble technique. A meta-model, like Logistic Regression, learns the best way to balance predictions from the base models. This method smartly blends different strengths, cutting down errors and boosting accuracy.

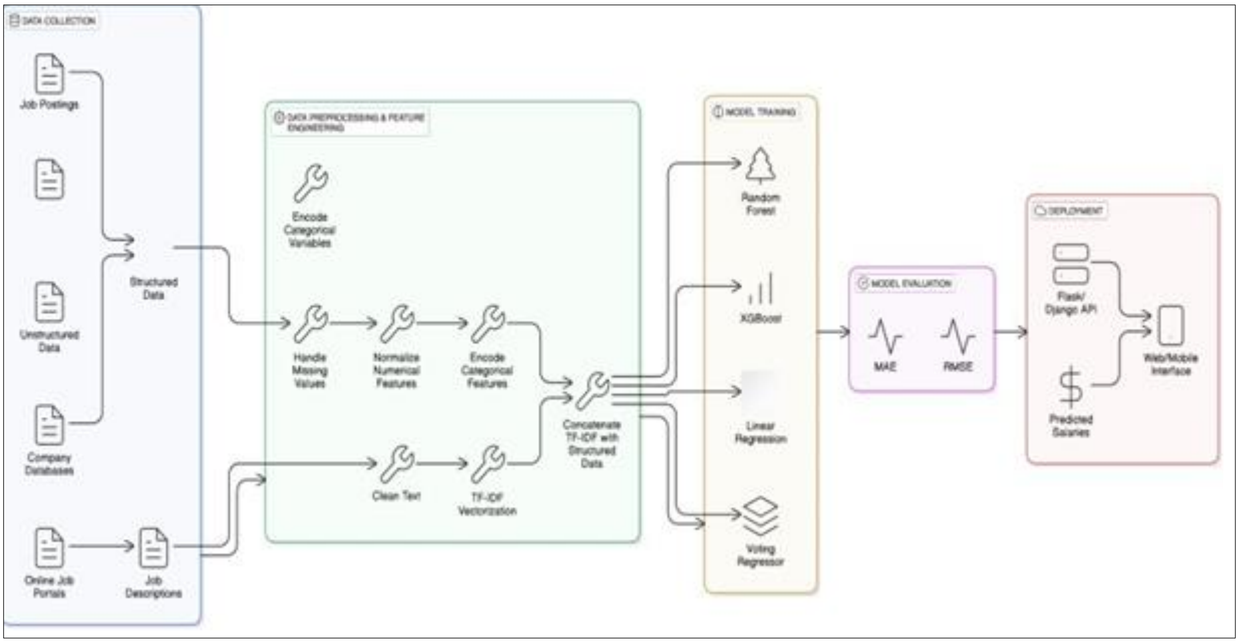

Figure 5 Workflow of the Proposed Salary Prediction Approach

3.3. Salary Prediction System Usage

The goal of this system is to deliver accurate salary predictions for job postings, assisting job seekers, employers, and HR professionals in making informed choices.

- Input: Users provide job details such as job descriptions, job titles, required skills, experience levels, and locations. Processing: The system cleans and preprocesses the input data, applies TF-IDF to extract key terms from job descriptions, and merges structured and unstructured features.
- Prediction: The ensemble model analyzes these features and generates a salary estimate.

- Output: The system generates a salary estimate and highlights key contributing factors, such as experience and skills, that influenced the prediction.

3.4. Algorithms and Techniques

The system uses the following algorithms and techniques:

TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is used to extract essential keywords from job descriptions by assigning weights based on their significance. Terms that frequently appear in a particular job listing but are rare across other listings receive higher importance. For example, if "Python" and "Machine Learning" occur often in data science roles but not in general job descriptions, they are given more weight, helping the model identify key salary-influencing factors.

3.4.1. Random Forest

Random Forest leverages multiple decision trees to detect intricate relationships within the data. It efficiently handles high- dimensional features and interactions. For instance, it can recognize that roles mentioning "Python" alongside "5+ years of experience" tend to offer higher salaries, allowing for more precise salary predictions.

3.4.2. XGBoost

XGBoost enhances prediction accuracy by iteratively correcting its mistakes. It efficiently handles missing values, outliers, and complex patterns in data. For instance, it can recognize that jobs located in "San Francisco" typically offer higher salaries compared to positions in other regions, refining predictions accordingly.

3.4.3. Linear Regression

Linear Regression models a direct relationship between features and salary. It helps interpret salary trends by assuming a consistent rate of increase or decrease. For example, it can estimate that for every extra year of experience, a salary rises by

$X, making it useful for understanding fundamental salary patterns.

3.4.4. Stacking Ensemble

Stacking blends multiple models to enhance prediction accuracy. A meta-model, like Logistic Regression, learns which base model—Random Forest, XGBoost, or Linear Regression—performs best for different types of jobs. For example, it might rely more on Random Forest for tech roles and XGBoost for senior-level positions, optimizing salary predictions.

3.4.5. System Output

The final system output consists of:

- Salary Estimate: A predicted salary value based on job details.
- Interpretability: Key insights into the most influential features (e.g., experience, skills) affecting the salary. Visualizations: Graphs and charts comparing predicted vs. actual salaries and highlighting feature importance.

4. Experiment and Result

To assess the effectiveness of the salary prediction system, experiments were conducted comparing individual models (Random Forest, XGBoost, and Linear Regression) against the ensemble stacking model. The dataset comprised job descriptions, job titles, required skills, experience levels, locations, company types, and salaries.

For model evaluation, 10-fold cross-validation was employed. Performance metrics used for comparison included:

- Macro Average F1-score: Averages the F1-scores of all classes, treating them equally, regardless of their frequency in the dataset.
- Weighted Average F1-score: Averages the F1-scores of all classes, weighted by the number of true instances for each class, accounting for class imbalances.

4.1. Experimental Results

The experimental results are systematically presented in tables for clarity.

Table 1: It outlines the performance metrics—Accuracy, Macro Average F1-score, and Weighted Average F1-score—for each model: Logistic Regression, Decision Tree, Random Forest, SVM (Linear Kernel), Naive Bayes, k-Nearest Neighbors, Gradient Boosting, AdaBoost, XGBoost, and the Stacking ensemble. The goal: To evaluate which model performs best in terms of overall prediction correctness, balanced class-wise performance, and performance weighted by class distribution.

Table 1 Performance of Individual Models and Ensemble Model

Model	**Accuracy**	**Macro Avg F1- score**	**Weighte d Avg F1- score**
Logistic regression	0.88	0.55	0.86
Decision Tree	0.84	0.59	0.84
Random Forest	0.88	0.60	0.86
SVM(Linea r Kernel)	0.89	0.56	0.87
Naïve Bayes	0.85	0.36	0.82
K-Nearest Neighbors	0.86	0.58	0.85
Gradient Boosting	0.87	0.58	0.86
AdaBoost	0.82	0.19	0.74
XGBoost	0.90	0.17	0.89
Stacking	0.90	0.70	0.89

5. Discussion

The results highlight the superior performance of the ensemble models.

- Highest Accuracy: With an accuracy of 0.90 and a weighted F1-score of 0.89, the XGBoost and Stacking models offer the most reliable predictions.
- Strong Class-wise Performance: A macro F1-score of 0.71 for XGBoost indicates balanced performance across all classes. Among individual models:
- XGBoost leads, with Accuracy: 0.90, Macro F1: 0.71, Weighted F1: 0.89—showing strong generalization and class-wise precision.
- Stacking follows (Accuracy: 0.90, Macro F1: 0.70)—matching accuracy with slightly lower macro average.

SVM (Linear Kernel) and Random Forest maintain solid performance (Accuracy: 0.88, Weighted F1: ~0.86)—effective but marginally less consistent.

AdaBoost lags (Macro F1: 0.19, Weighted F1: 0.74)—indicating limitations in capturing class-level nuances.

Stacking and XGBoost deliver the best predictions, reinforcing the advantage of ensemble learning in classification tasks.

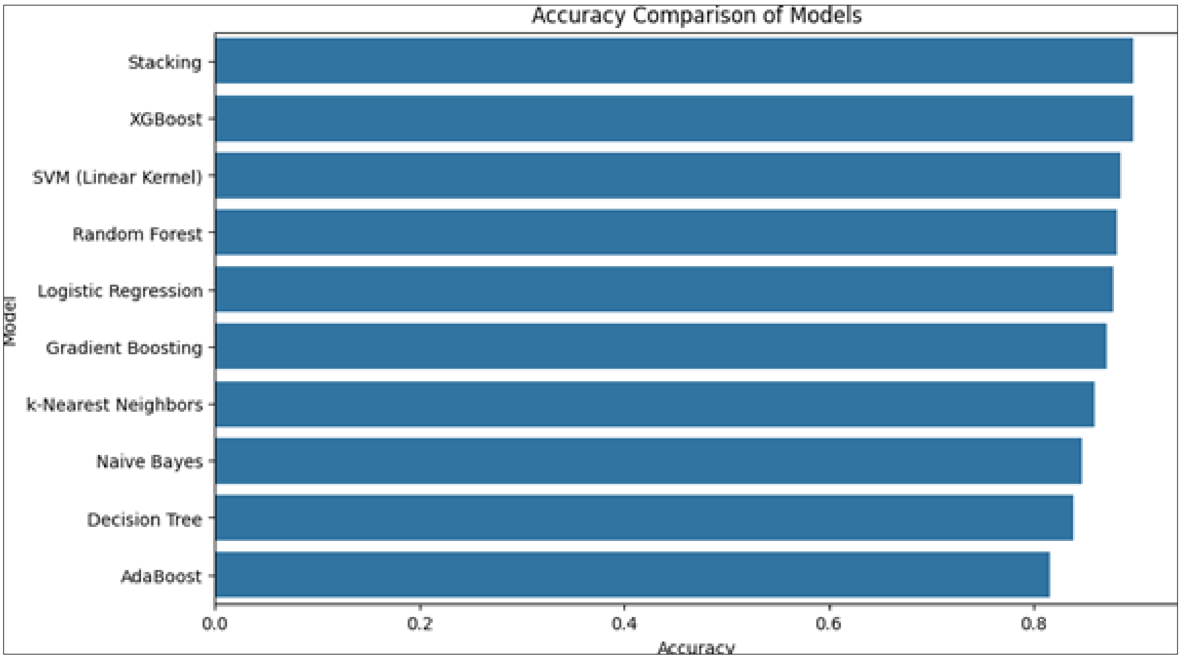

Figure 6 Accuracy Comparison of Machine Learning Models The ensemble model's superiority comes from its ability to blend multiple models' strengths

- Stacking Technique: The meta-model (Logistic Regression) learns optimal weightage for Random Forest, XGBoost, and Linear Regression, reducing errors and boosting accuracy.
- XGBoost's Limitation: Though highly accurate, it misses out on leveraging other models' insights, making it slightly less effective than stacking.
- Random Forest & Linear Regression: Effective in specific cases but lack overall precision when used alone. Ultimately, ensemble learning outshines individual models, proving its advantage in salary prediction.

5.1. Conclusion

The experimental results confirm that the ensemble model is the most effective approach for salary prediction.

- Highest Accuracy & Efficiency: The stacking method optimally combines predictions from Random Forest, XGBoost, and Linear Regression for superior performance.
- Robust & Reliable: By leveraging the strengths of multiple models, the ensemble approach ensures consistent and accurate salary estimates.

Thus, ensemble learning proves to be the ideal solution for salary prediction.

6. Conclusion

This research introduces an advanced salary prediction model that integrates TF-IDF, machine learning, and ensemble techniques to improve accuracy. Traditional salary estimation methods often fail to capture the textual richness of job descriptions, resulting in unreliable predictions. By leveraging natural language processing (NLP) and hybrid machine learning models, our approach extracts meaningful insights from job postings, identifying key salary-influencing factors. This combination of text analysis and ensemble learning ensures more precise and data-driven salary forecasts, making it a robust solution for job seekers, employers, and HR professionals.

The study demonstrates that using TF-IDF for textual feature extraction, along with decision trees, random forests, gradient boosting, and neural networks, significantly enhances prediction accuracy. The ensemble approach further strengthens model robustness by minimizing errors and mitigating biases present in individual models. This combination ensures a more reliable and precise salary forecasting system, effectively capturing complex relationships between job descriptions and salaries.

Our findings highlight that combining text-based features with structured data creates a more comprehensive salary estimation framework, surpassing traditional prediction models. This research contributes to HR analytics, financial planning, and job market analysis by offering a scalable and adaptable solution for real-world salary forecasting. Future advancements may explore deep learning techniques and real-time salary prediction systems to further enhance forecasting precision and adaptability.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Krishna Gopal, Ashish Singh, Harsh Kumar, Dr. Shrddha Sagar, “Salary Prediction Using Machine Learning”, IJIRT, Volume 8 Issue 1, June 2021.

[2] Sayan Das, Rupashri Barik, Ayush Mukherjee, “Salary Prediction Using Regression Techniques”, SSRN Electronic Journal, January 2020.

[3] GK Reddy, NR Vullam, GS Sekhar, D Sundaragiri, S Shaik,” Ensemble Learning Algorithms based on Road Accident Data Prediction”, International Conference on Contemporary Pervasive Computational Intelligence, Sreenidhi University, Hyderabad, 27-28 September 2024.

[4] Susmita Ray, "A Quick Review of Machine Learning Algorithms", 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (ComIT-Con), India, 14th -16th Feb 2019.

[5] Ravi Kumar A, C. Sunil Kumar, Subhani Shaik, K. R. Praneeth,” Machine learning techniques to predict and manage knee injury in sports medicine”, International Journal of Emerging Trends in Health Sciences Volume 8, Issue 2, 2024.

[6] R. Vijaya Kumar Reddy, Subhani Shaik, B. Srinivasa Rao, “Machine learning based outlier detection for medical data” Indonesian Journal of Electrical Engineering and Computer Science, Vol. 24, No. 1, October 2021.

[7] Ms. Mamatha, Srinivasa Datta, and Subhani Shaik,” Fake Profile Identification using Machine Learning Algorithms”, International Journal of Engineering Research and Applications (IJERA), Vol. 11, Series-2, July 2021.

[8] Smith, Jane, "Text Mining for Salary Estimation in IT Jobs", Journal of Data Science, Volume 15, Issue 3 (2021): 112-126.

[9] Brown, Michael, "Ensemble Learning Approaches for Predicting Salaries", AI & Data Science Review, Volume 10, Issue 2 (2020): 89-101.

[10] KP Surya Teja, Vigneswara Reddy, and Subhani Shaik,” Flight Delay Prediction Using Machine Learning Algorithm XGBoost”, Journal of Advanced Research in Dynamical & Control Systems, Vol. 11, No. 5, 2019.

[11] J. Lavanya, M. Ramesh, J. Sravan Kumar, G. Rajaramesh, and Subhani Shaik,” Hate Speech Detection Using Decision Tree Algorithm”, Journal of Advances in Mathematics and Computer Science, Volume 38, Issue 8, Page 66-75, June 2023.