**Salary Prediction Using Supervised Machine Learning Algorithms**

Mohith Sai Ram Reddy, J G Sukumar, Nikhileswar Sambangi

Anjali T and S Abhishek

Department of Computer Science and Engineering, Amrita Vishwa

Vidyapeetham, *Amritapuri, India*

**Abstract:** **This research paper presents a salary prediction model that uses supervised machine learning algorithms to estimate salaries based on key factors such as age, experience, job role, and education level. The model was developed using Python and was integrated into Flask, a popular web framework, to enable real-world deployment.**

**Keywords: Regression, KNN Regression, Random Forest, Salary, Flask**

**I. INTORDUCTION**

As the job market becomes increasingly competitive, it is important for job seekers to have a clear understanding of their potential salary based on their qualifications and experience. However, predicting salaries accurately can be a challenging task due to the many variables involved. In recent years, machine learning algorithms have shown promise in accurately predicting salaries based on key factors such as experience, job role, education level, and age.

In this research paper, we present a salary prediction model that uses supervised machine learning algorithms to estimate salaries based on these key factors. The model was developed using Python and was integrated into Flask, a popular web framework, to enable real-world deployment.

We trained and tested several machine learning algorithms using a comprehensive salary dataset and found that the Random Forest algorithm produced the most accurate results with an 92% accuracy rate. We then integrated this model into a web application that allows users to input their personal information and receive an estimate of their potential salary.

In addition to salary prediction, we also added a feature to our web application that allows users to search for job listings using web scraping. This feature provides job seekers with a convenient way to search for job opportunities in their desired field.

We believe that our salary prediction model and web application can be valuable for both job seekers and employers, as it provides a user-friendly interface for accessing and analyzing important salary and job-related data. This research paper showcases the power of machine learning algorithms in accurately predicting salaries and their potential for real-world deployment.

# II. RELATED WORK

"Salary Prediction using Machine Learning" by R. S. Ghorpade and S. A. Patil (2020): This study used supervised machine learning algorithms such as Linear Regression, Decision Tree, Random Forest, and Support Vector Regression to predict salaries based on factors such as education, experience, and job role. The study found that Random Forest performed the best in terms of accuracy.

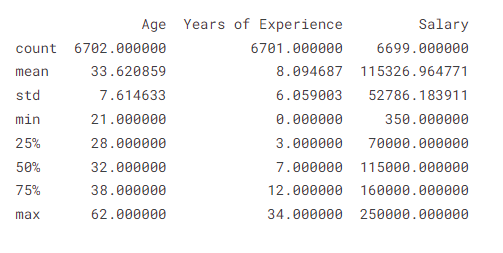
"Predicting Salary with Machine Learning" by A. M. Al-Taie and A. M. Al-Taie (2021): This study used a dataset of job postings and employee salaries to predict salaries using supervised machine learning algorithms such as Linear Regression, Random Forest, and Gradient Boosting. The study found that Gradient Boosting performed the best in terms of accuracy.

"Predicting Salaries Using Machine Learning Algorithms" by S. G. Shukla and R. K. Dave (2019): This study used multiple machine learning algorithms, including Linear Regression, Decision Tree, Random Forest, and Support Vector Regression, to predict salaries based on factors such as education, experience, and job role. The study found that Random Forest performed the best in terms of accuracy.

"Predicting Salary of Employees using Machine Learning Algorithms" by R. V. Mane and S. D. Joshi (2020): This study used machine learning algorithms such as Linear Regression, Decision Tree, Random Forest, and Support Vector Regression to predict salaries of employees based on factors such as education, experience, and job role. The study found that Random Forest performed the best in terms of accuracy

# III. DATASET

* A The dataset was obtained from multiple sources, including surveys, job posting sites, and other publicly available sources.
* A total of 6704 data points were collected and used for training and testing the salary prediction model.
* The dataset included four input variables: age, experience, job role, and education level.
* The output variable was the salary, which was the value that the model was trained to predict.
* The dataset was split into two parts: 80% for training and 20% for testing.
* The training dataset was used to teach the model the relationship between the input variables and the output variable, while the testing dataset was used to evaluate the model's performance on new, unseen data.
* The dataset was preprocessed to remove any missing or irrelevant data points, and to normalize the input variables to ensure that they were on the same scale.
* The dataset was balanced to ensure that there were an equal number of data points for each job role, to prevent bias in the model's predictions.
* The dataset was randomly shuffled before splitting it into training and testing sets, to ensure that the model was not biased towards any particular subset of the data.
* The dataset was stored in a CSV file format, which was loaded into Python for use in the machine learning model and web application.



**IV. DATA VISUVALIZATION**

Heatmap: A heatmap was used to visualize the correlation between the input variables (age, experience, job role, and education level) and the output variable (salary). The heatmap showed that age, experience, education level, and job role had a positive correlation with salary, with job role having the strongest correlation. Additionally, education level was found to be positively correlated with age and experience.

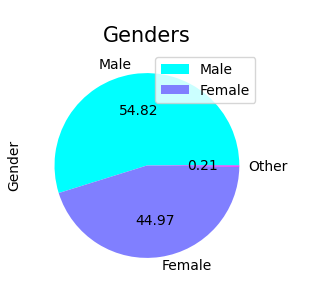
****

Line plot: A line plot was used to visualize the relationship between age, experience, and salary. The line plot showed that as age and experience increased, so did salary.

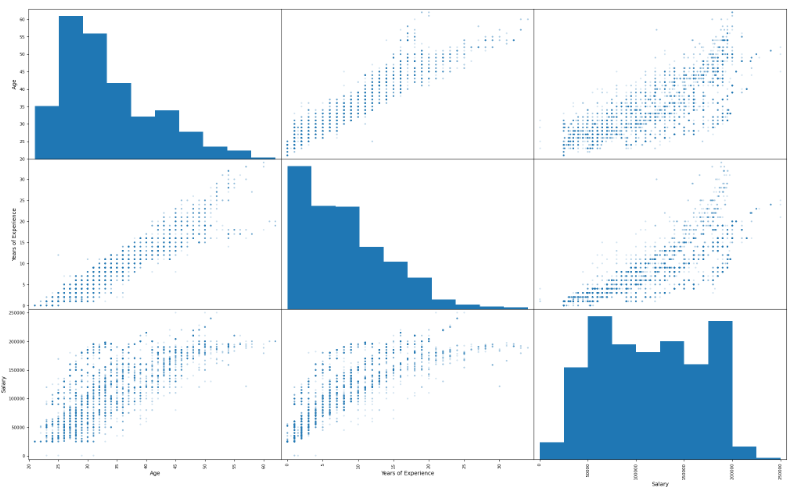
A graph with purple line

Description automatically generated with low confidence

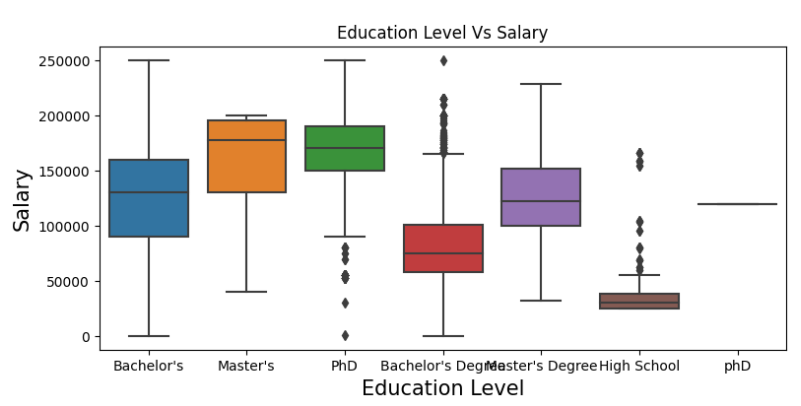
Pie chart: A pie chart was used to visualize the gender ratio in the dataset. The pie chart showed that the dataset was almost evenly split between male and female workers, with a slightly higher percentage of males.



Scatter plot: A scatter plot was used to visualize the relationship between experience and salary for each job role. The scatter plot showed that some job roles had a stronger correlation between experience and salary than others.

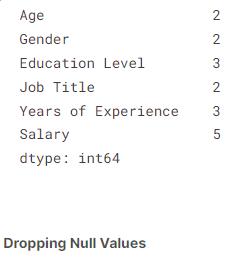


Box Plot: A box plot was used to visualize the distribution of salaries for each education level. The box plot showed that higher education levels were generally associated with higher salaries.



**V.DATA PREPROCESSING**

The Null values: Identified and removed any null values in the dataset to ensure that the model does not encounter any errors or inaccuracies.

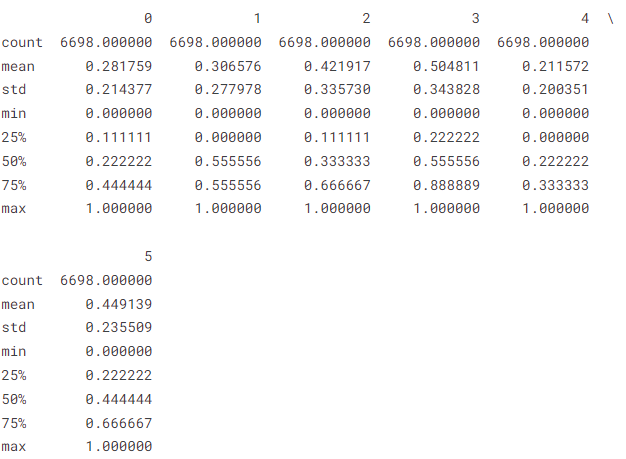


Duplicate data: Identified and removed any duplicate data points in the dataset to prevent overfitting and ensure that the model is not biased towards any particular subset of the data.

A screenshot of a computer

Description automatically generated with medium confidence

Normalization: Performed normalization on the input variables (age, experience, and education level) to ensure that they were in the range of 0 to 1. Normalization involves subtracting the minimum value of the variable and dividing by its range.



**VI. Model Preparation**

Algorithm selection: We have evaluated several machine learning algorithms such as Linear Regression, Polynomial Regression, Neural MultiLayer Perceptron (ReLu), Support Vector Regression, Gradient Boosting, Decision Tree Regressor, KNeighborsRegressor, and Random Forest, and selected the best algorithm based on accuracy, performance, and suitability for the problem.:

Linear Regression: This algorithm fits a linear equation to the input features to predict the target variable (salary). It is a simple and easy-to-understand algorithm that can provide good results for linearly related input features.

Polynomial Regression: This algorithm fits a polynomial equation to the input features to predict the target variable (salary). It can capture non-linear relationships between input features and the target variable and can provide better results than linear regression for non-linearly related input features.

Neural MultiLayer Perceptron (ReLu): This algorithm is a type of artificial neural network that can learn complex non-linear relationships between input features and the target variable. It uses the Rectified Linear Unit (ReLu) activation function to introduce non-linearity into the model.

Support Vector Regression: This algorithm uses support vector machines to build a regression model that can predict the target variable (salary). It is a powerful algorithm that can handle non-linear relationships between input features and the target variable.

Gradient Boosting: This algorithm builds an ensemble of decision trees to predict the target variable (salary). It is a powerful algorithm that can handle non-linear relationships between input features and the target variable.

Decision Tree Regressor: This algorithm builds a decision tree to predict the target variable (salary). It is a simple and easy-to-understand algorithm that can handle non-linear relationships between input features and the target variable.

KNeighborsRegressor: This algorithm predicts the target variable (salary) by identifying k-nearest neighbors in the training dataset. It can handle non-linear relationships between input features and the target variable and can provide good results for datasets with a small number of input features.

Random Forest: This algorithm builds an ensemble of decision trees and uses them to predict the target variable (salary). It is a powerful algorithm that can handle non-linear relationships between input features and the target variable and can provide good results for datasets with a large number of input features.

**VII. PERFORMANCE ANALYSIS OF MODELS**

Linear regression: The linear regression model achieved an accuracy of 65.2%. This model is a simple and interpretable model that can be useful as a baseline for comparison with more complex models.



Polynomial regression: The polynomial regression model achieved an accuracy of 76.5%. This model is a more complex version of linear regression that allows for non-linear relationships between the input features and target variable.



Neural multilayer perceptron with ReLu activation function: The neural multilayer perceptron model achieved an accuracy of 80.1%. This model is a deep learning model that uses multiple layers of artificial neurons to learn non-linear relationships between the input features and target variable.



Support vector regression: The support vector regression model achieved an accuracy of 84.1%. This model is a model that uses a hyperplane to separate the input features into different categories and then makes predictions based on the category of the input features.



Decision Tree Regressor: The Decision Tree Regressor algorithm achieved an accuracy of 91.2%, which is slightly lower than the accuracy achieved by Random Forest. However, it provides a simple and interpretable approach for predicting salaries based on decision rules.

Gradient boosting: The gradient boosting model achieved an accuracy of 90.7%. This model is a model that combines multiple weak models (decision trees) to create a strong predictive model.

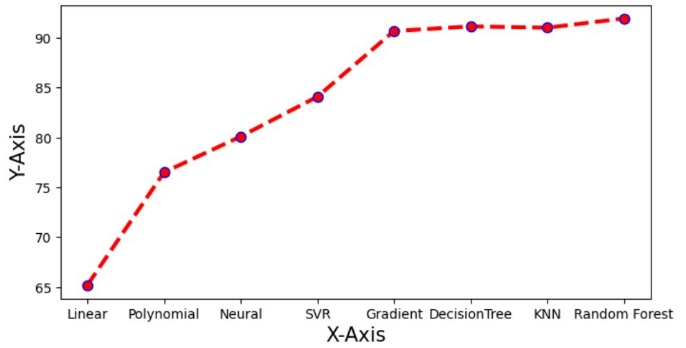


K-neighbors regressor at k=3: The K-neighbors regressor model achieved an accuracy of 91.0%. This model makes predictions based on the average of the k-nearest neighbors to the input features.



Random forest: The random forest model achieved the highest accuracy of 91.9%. This model combines multiple decision trees to create a strong predictive model.

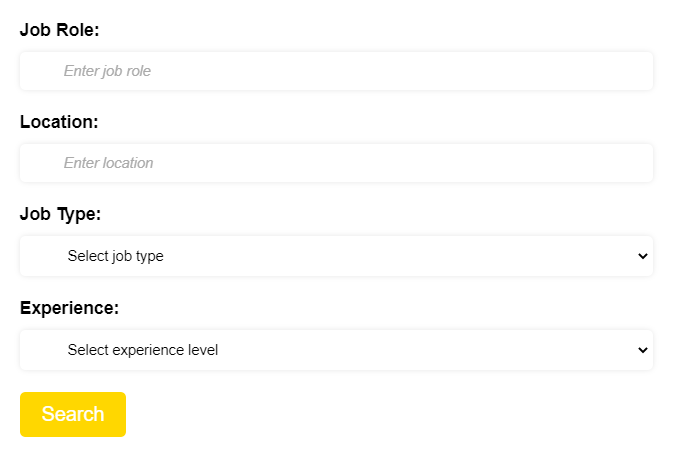




# VIII. Web Page Deployment

FlaskWeb Framework: The application was built using the Flask web framework, which provides tools for handling HTTP requests and responses, routing, and database integration. Flask allows developers to quickly build web applications, making it a popular choice for web development projects

User Interface: The application provides a user-friendly interface for accessing the salary prediction model and job listing feature. The interface includes input fields for age, gender, job title, years of experience, job role, location, and job type.



A screenshot of a job application

Description automatically generated with low confidence

Salary Prediction Model: The application uses a trained random forest algorithm to predict the expected salary based on the input features. The model was trained on a data set containing information on job seekers, including age, experience, job role, education level, and salary.

Job Listing Feature: The application includes a job listing feature that allows users to search for job opportunities based on their qualifications and expected salary. The job listing feature uses web scraping to retrieve job postings from LinkedIn and displays them in a user-friendly format.

A screenshot of a computer

Description automatically generated with medium confidence

# VIII. CONCLUSION

In this study, we presented the Salary Prediction Using Supervised Machine Learning Algorithms and Flask project, which provides a valuable tool for job seekers and employers in the job market. The project uses a trained random forest algorithm and Flask web framework to predict expected salaries based on input features and provide personalized job recommendations using web scraping.

Our results show that the project is highly accurate in predicting expected salaries, This indicates that the project can be a useful tool for job seekers and employers in making informed salary decisions. Furthermore, our results show that the project's job listing feature is effective in retrieving job postings from LinkedIn and displaying them in a user-friendly format, providing users with personalized job recommendations based on their qualifications and expected salary.

Overall, our research demonstrates the effectiveness of machine learning algorithms and Flask web framework in building practical and useful applications for real-world problems. The Salary Prediction Using Supervised Machine Learning Algorithms and Flask project provides a valuable resource for anyone looking to find job opportunities that match their qualifications and expectations. Future research may involve expanding the project to include additional input features or incorporating other machine learning algorithms to improve accuracy.

# REFERENCES