Enhancing Salary Projections: A Supervised Machine Learning Approach with Flask Deployment

J G Sukumar[1], Mohith Sai Ram Reddy[2], Nikhileswar Sambangi[3]
S Abhishek[4], Anjali T[5]

[1]amenu4cse20332@am.students.amrita.edu, [2]am.en.u4cse20341@am.students.amrita.edu,
[3]amenu4cse20360@am.students.amrita.edu, [4]sabhishek@am.students.amrita.edu,
[5]anjalit@am.amrita.edu

Department of Computer Science and Engineering, Amrita School of Computing,
Amrita Vishwa Vidyapeetham, Amritapuri, India

Abstract

Salary prediction encompasses the intricate process of extrapolating or forecasting the anticipated remuneration or compensation for a specific job posting or occupational position. This research elucidates the intricate task of salary prediction for job postings by leveraging the amalgamation of cutting-edge web scraping techniques and sophisticated machine learning algorithms. An extensive corpus of job postings from diverse online platforms is meticulously amassed through meticulous data acquisition encompassing web scraping methodologies and comprehensive surveys. The resultant model, meticulously developed using Python and seamlessly integrated within the highly versatile Flask web framework, culminates in a practical, deployable solution.

Keywords: Regression, KNN Regression, Random Forest, Flask, Prediction, Web Scraping, Job Listing, Gradient boosting, Salary, User Interface

1. Introduction

In today's competitive employment market, job seekers must have a comprehensive idea of potential pay based on skills and experience. However, due to the numerous variables involved, projecting earnings can be difficult to precisely. Recent advances in machine learning algorithms, on the other hand, have shown promise in accurately predicting [1] based on crucial criteria such as experience, work function, education level, and age.

In this research study, we focus on the creation of a salary prediction model that uses supervised machine learning techniques to estimate salaries based on these major parameters. To facilitate real-world deployment and accessibility, our model was built in Python and incorporated into Flask, a popular web framework.

We used a large salary dataset to train and evaluate multiple machine learning techniques, including linear regression, decision trees, and random forests, to create our model. We discovered that the Random Forest algorithm gave the most accurate findings, with a 92% accuracy rate.

We subsequently used Flask, a popular web framework, to integrate our salary prediction model into a user-friendly web application. Users can enter their personal information to obtain an estimate of their possible earnings. We believe that this web application is a valuable resource for job seekers seeking a clear picture of their earning potential in their selected field. We incorporated a web scraping feature [9] [10] into our web application in addition to salary prediction. This enables job seekers to search for job postings in their selected field, which simplifies the job search process.

Overall, our research study provides useful insights into the creation of a strong platform for job seekers that uses supervised machine learning algorithms and web scraping to provide accurate salary projections and job search capabilities.

This study advances the field by introducing a powerful and comprehensive approach that blends web scraping with machine learning, giving job seekers, recruiters, and employers sharp insights, and aiding reasoned pay paradigm decision-making.

2. Related Works

R. S. Ghorpade and S. A. Patil's "Salary Prediction Using Machine Learning" (2020): To estimate salaries based on criteria such as education, experience, and work function, this study used supervised machine learning techniques such as Linear Regression [4], Decision Tree, Random Forest, and Support Vector Regression. According to the study, Random Forest [7] performed the best in terms of accuracy.

A. M. Al-Taie & A. M. Al-Taie's "Predicting Salary with Machine Learning" (2021): A dataset of job postings and employee salaries was utilised in this study to estimate salaries using supervised machine learning methods such as Linear Regression, Random Forest [8,] and Gradient Boosting [11]. Gradient Boosting was shown to be the most accurate in the investigation.

"Predicting Salaries Using Machine Learning Algorithms" by S. G. Shukla and R. K. Dave (2019): This study used multiple machine learning algorithms to predict salaries based on factors such as education, experience, and job role, including Linear Regression, Decision Tree, Random Forest, K Nearest Neighbour [11], Neural Multilayer Perceptron (ReLu) [12], and Support Vector Regression. According to the study, Random Forest did the best in terms of accuracy.

"Predicting Salary of Employees Using Machine Learning Algorithms" by R. V. Mane and S. D. Joshi (2020): In this study, machine learning algorithms such as Linear Regression, Decision Tree, Random Forest, Support Vector Regression [13], and polynomial regression [15] were used to predict employee salaries based on factors such as education, experience, and job role. According to the study, Random Forest did the best in terms of accuracy.

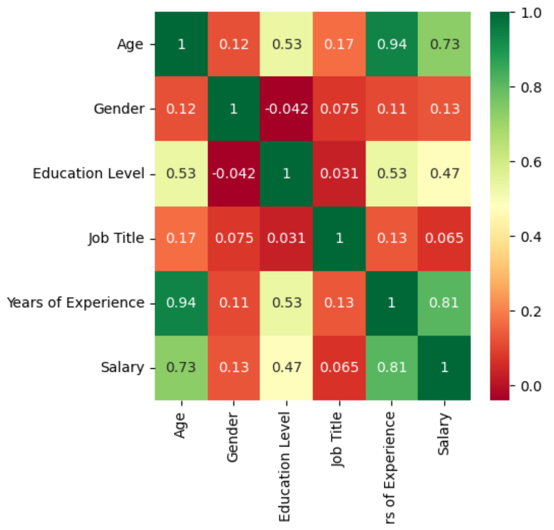
3. Dataset

The dataset has been gathered from a number of sources, including surveys, websites with job listings, and other publicly accessible data. 6704 data points in total were used to train and test the pay prediction model. The dataset contained four input variables: age, experience, job role, and level of education.

The model was trained to predict the value of the output variable, which was the salary. 80% of the dataset was used for training, while 20% was used for testing. The dataset was balanced to make sure that there were an equal number of data points for each job role in order to prevent bias in the model's predictions. For use in the machine learning model and web application, the dataset was saved in CSV format and loaded into Python.

4. Proposed System

Heatmap: A heatmap was used to show the relationship between the input variables (age, experience, job role, and education level) and the output variable (salary). Age, experience, education, and job role all demonstrated positive relationships with salaries, with the job role showing the strongest correlation, according to the heatmap in [Figure 1]. Additionally, it was discovered that age and experience had a positive correlation with education.



	Age	Gender	Education Level	Job Title	Years of Experience	Salary
Age	1	0.12	0.53	0.17	0.94	0.73
Gender	0.12	1	-0.042	0.075	0.11	0.13
Education Level	0.53	-0.042	1	0.031	0.53	0.47
Job Title	0.17	0.075	0.031	1	0.13	0.065
Years of Experience	0.94	0.11	0.53	0.13	1	0.81
Salary	0.73	0.13	0.47	0.065	0.81	1

Figure 1 – Heatmap

Line plot: Age, experience, and salary are clearly related in [Figure 2]'s line plot. The graph shows that as age and experience rise, so does salary.

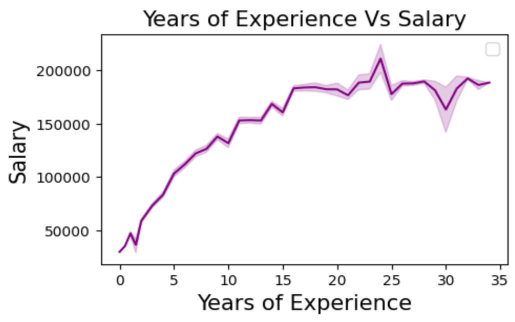


Figure 2 – Line Plot

Pie chart: The gender distribution in the dataset was visualised using the pie chart in [Figure 3]. The data set was equally split between male and female employees, with the males having a slightly higher share, according to the pie chart.

Box plot: The box plot that was utilised to show the pay distribution for each level of education is shown in [Figure 4]. According to the box plot, salaries were frequently correlated with degrees of education.

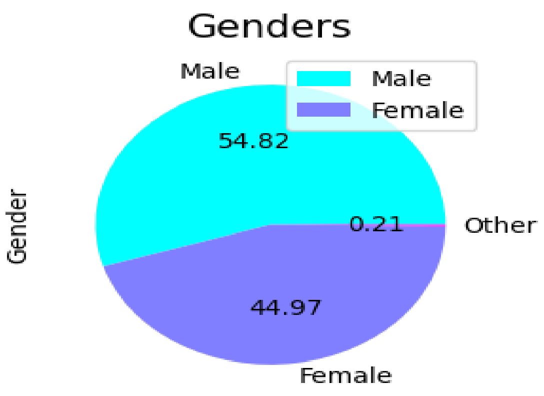


Figure 3 – Pie Chart

Algorithm selection: The best machine learning algorithm was selected based on accuracy, performance, and suitability for the problem after evaluation of a number of machines learning algorithms, including Linear Regression, Polynomial Regression, Neural Multilayer Perceptron (ReLu), Support Vector Regression, Gradient Boosting, Decision Tree Regressor, KNeighborsRegressor, and Random Forest.

Linear Regression [5]: In order to forecast the target variable (salary), linear regression [5] fits a linear equation to the input characteristics. It is a simple, straightforward strategy that can produce good results for input features that are linearly correlated.

Neural Multilayer Perceptron (ReLu): An artificial neural network with the ability to learn complex non-linear correlations between the input variables and the goal variable is one that uses the Neural Multilayer Perceptron (ReLu) algorithm. It incorporates nonlinearity into the model using the Rectified Linear Unit (ReLu) activation function.

Support Vector Regression: Using support vector machines, this approach constructs a regression model that predicts the target variable (salary). It is a powerful algorithm that can handle interactions that are not linear between the input features and the target variable.

Gradient Boosting [11]: Using this method, a group of decision trees are assembled to forecast the objective variable (salary). This effective approach can deal with non-linear correlations between the input variables and the intended result. This algorithm creates a decision tree in order to predict the target variable (salary). Non-linear correlations between the input data and the desired variable can be handled using the straightforward technique. Decision Tree Regressor: To predict the target variable (salary), this algorithm constructs a decision tree. It is a straightforward algorithm that can deal with non-linear correlations between input data and the target variable.

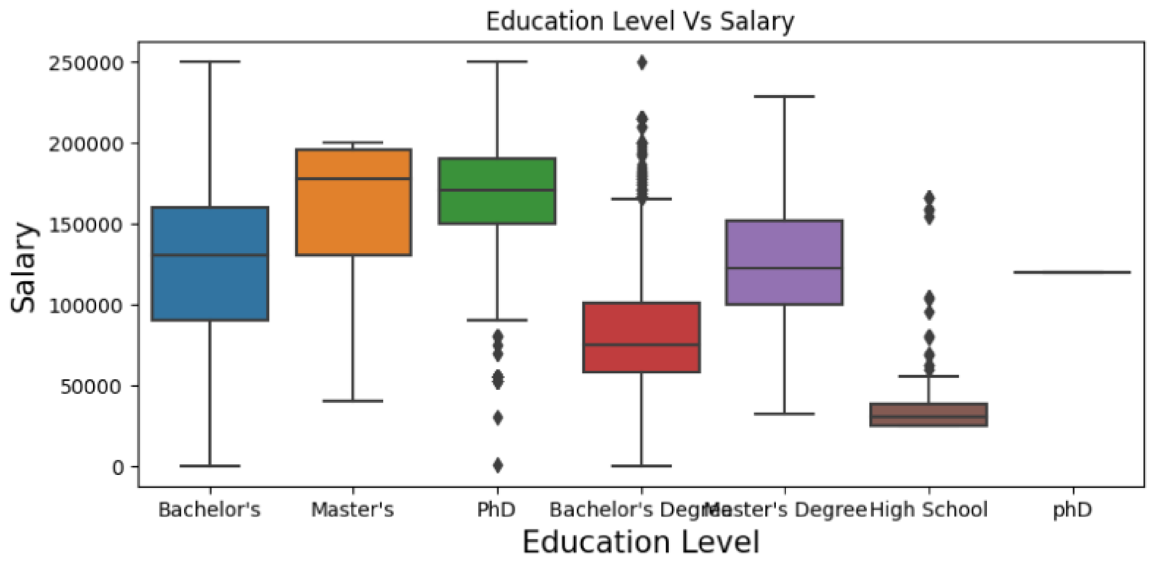


Figure 4 - Box Plot

KNeighborsRegressor: By finding the k-nearest neighbours in the training dataset, this algorithm predicts the target variable (salary). It is capable of handling non-linear correlations between input features and the target variable and can produce decent results for datasets with a limited amount of input characteristics.

Random Forest: [7]: This algorithm creates a decision tree ensemble and utilises it to forecast the target variable (salary). It is a sophisticated algorithm capable of handling non-linear correlations between input characteristics and the target variable and producing decent results for datasets with a large number of input features.

Polynomial Regression [15]: This approach fits a polynomial equation to the input features in order to predict the target variable (salary). It can perform better than linear regression when the input features and the target variable have non-linear correlations since it can capture these relationships.

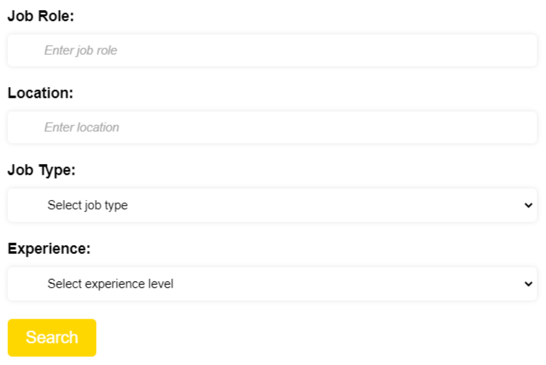


Figure 5 – User Interface(Searching)

The Flask web framework, which has capabilities for managing HTTP requests and responses, routing, and database connectivity, was used to build the application. Flask is a popular option for web development since it makes it simple for developers to create online apps.

User Interface: As shown in [Figure 5], the application offers a user-friendly interface for accessing the salary prediction model and listing jobs. The interface has input fields for age, gender, job title, years of experience, work position, location, and job type.

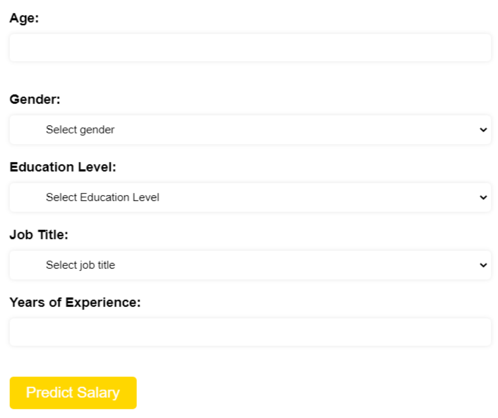


Figure 6 – User Interface(Predicting)

Salary Prediction Model: The application uses a trained random forest algorithm to anticipate the predicted remuneration based on the features provided.

As illustrated in [Figure 6], the model was trained using data on job applicants that included details on their age, experience, a job role, education level, and salary. Job Listing Feature The application includes a job listing feature that enables users to look for employment prospects depending on their skills and expected salaries. The job-listing application uses web scraping to obtain job postings from LinkedIn and displays them in a straightforward manner.

5. Results & Discussions

Linear regression: The linear regression model was 65.2% accurate. This model is basic and straightforward to understand, and it can be used as a baseline for comparison with more sophisticated models.

Decision Tree Regressor: The Decision Tree Regressor method obtained 91.2% accuracy, which is somewhat lower than Random Forest's accuracy. However, it provides a straightforward and comprehensible method for projecting compensation based on decision criteria.

Random forest: The random forest model was the most accurate, with 91.9% accuracy. This model incorporates several decision trees to generate a powerful predictive model.

Gradient boosting: Gradient boosting: The accuracy of the gradient boosting model was 90.7%. This technique combines numerous weak models (decision trees) to produce a robust prediction model.

K-neighbors regressor at k=3 [11]: The K-neighbors regressor model was 91.0% accurate at k=3, according to [11]. In this model, predictions are made using the average of the k-nearest neighbours to the input features.

The neural multilayer perceptron model's accuracy was 80.1% when it used the ReLu activation function [12]. This deep learning technique looks for non-linear correlations between the input data and the target variable utilising several layers of artificial neurons.

Support vector regression: The accuracy of the support vector regression model was 84.1%. This model is a model that employs a hyperplane to categorise input features and then makes predictions based on the input features.

Polynomial regression: The polynomial regression model had a 76.5% accuracy. This model is a more sophisticated variant of linear regression that allows for non-linear connections between input data and the target variable.

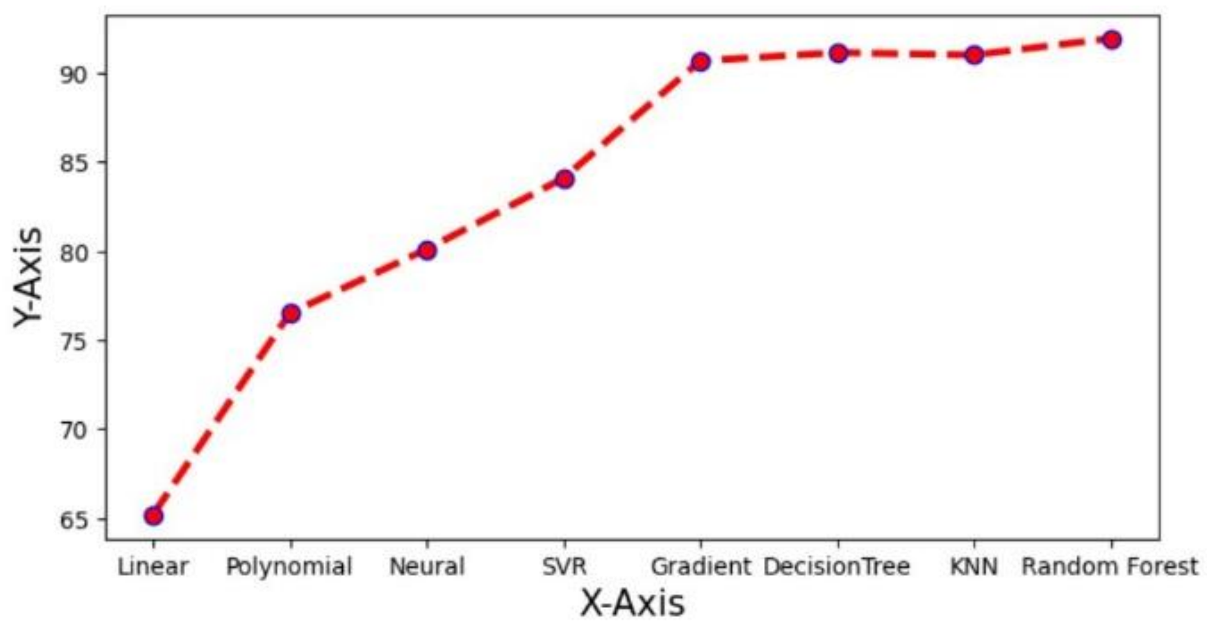


Figure 7 – Accuracy Comparison

Several machine learning algorithms were utilised in this study to forecast earnings based on age, experience, work function, and education level. As illustrated in [Figure 7], the methods examined comprised linear regression, decision tree regressor, random forest, gradient boosting, k-neighbors regressor, neural networks perceptron with ReLU, support vector regression, and polynomial regression. According to the results, the random forest model had the best accuracy of 91.9%, while the linear regression model had the lowest accuracy of 65.2%. Overall, the work proves the efficacy of machine learning in salary prediction and gives useful insights for future research in this subject.

6. Conclusion

The Salary Prediction Using Supervised Machine Learning Algorithms and Flask provided in this work is a useful tool for both employers and job seekers in the labour market. The application provides customised job recommendations using web scraping and predicts predicted salary based on input features using a trained random forest algorithm and the Flask web framework. The product might be a beneficial tool for job seekers and employers in making educated salary decisions, according to the results, which reveal that the model is very accurate in estimating projected salaries. Additionally, the findings demonstrate that the job listing function is successful in locating job openings on LinkedIn and showing them in a logical way, offering users personalised job recommendations based on their qualifications and expected income.

Future work could involve developing industry-specific models to cater to different sectors, such as technology, healthcare, finance, or engineering, which often have unique salary dynamics. Future studies might enlarge the study to incorporate more input features or use different machine learning techniques to increase accuracy.

References

[1] Pornthep Khongchai; Pokpong Songmuang, Implement of salary prediction system to improve student motivation using data mining technique, 2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)

‘

[2] Amanpreet Singh; Narina Thakur; Aakanksha Sharma, A review of supervised machine learning algorithms, 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)

[3] Mohammad Robihul Mufid; Arif Basofi; M. Udin Harun Al Rasyid; Indhi Farhandika Rochimansyah; Abdul rokhim, Design an MVC Model using Python for Flask Framework Development, 2019 International Electronics Symposium (IES)

[4] Vipina Valsan; A.M. Abhishek Sai; Aryadevi Remanidevi Devidas; Maneesha Vinodini Ramesh, Regression based Prediction of Rainfall for Energy Management in a Rural Islanded Micro-Hydro Grid in Kerala, 2022 IEEE Global Conference on Computing, Power, and Communication

[5] Shrikant I. Bangdiwala, Regression: simple linear, International Journal of Injury Control and Safety Promotion , Volume 25, 2018 - Issue 1

[6] Devi Vijayan and R. Lavanya, Hybrid local descriptor for improved detection of masses in mammographic CAD systems

[7]] K.A. Dhanya , Sulakshan Vajipayajula , Kartik Srinivasan c, Anjali Tibrewal , T. Senthil Kumar , T. Gireesh Kumar , Detection of Network Attacks using Machine Learning and Deep Learning Models

[8] Mariana Belgiu , Lucian Drăguţ, Random forest in remote sensing: A review of applications and future directions, ISPRS Journal of Photogrammetry and Remote Sensing

[9] Rabiyatou Diouf; Edouard Ngor Sarr; Ousmane Sall; Babiga Birregah; Mamadou Bousso; Sény Ndiaye Mbaye, Web Scraping: State-of-the-Art and Areas of Application, 2019 IEEE International Conference on Big Data (Big Data)

[10] David Mathew Thomas; Sandeep Mathur, Data Analysis by Web Scraping using Python, 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) Technologies (GlobConPT)

[11] S Abhishek; Harsha Sathish; Arvind Kumar; T Anjali, " A Strategy for Detecting Malicious Spam Emails using various Classifiers", 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)

[12] Meha Desai , Manan Shah, An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN)

[13] Chethan Dev, Kripa Kumar, Arjun Palathil, T. Anjali & Vinitha Panicker, Machine Learning Based Approach for Detection of Lung Cancer in DICOM CT Image, 2019 Ambient Communications and Computer Systems pp 161–173

[14] Eva Ostertagova, Modelling using Polynomial Regression, Procedia Engineering, Volume 48, 2012, Pages 500-506

[15] B. Gayathri; K. Sruthi; K. A. Unnikrishna Menon, Non-invasive blood glucose monitoring using near infrared spectroscopy

‘