Joanna Chou, Alex Buntaran, Aditya Manikonda
CSC 466
Dr. Rwebangira

Predictive Modeling of Salaries

Overview

This paper explores the application of various machine learning algorithms in predicting salaries, aiming to provide insights into their performance and applicability in real-world scenarios.

Motivation

The outcomes of this project are anticipated to assist job seekers or recent graduates in negotiating their income for either existing or new positions. Employers can also benefit by employing predictive modeling to ensure equitable compensation for their workforce. By comparing multiple algorithms, our goal is to pinpoint the most efficient algorithm for salary prediction.

Previous Work

Several projects have tackled salary prediction and constructed salary analyses utilizing machine learning techniques. For instance, Analytics Vidhya employed Linear Regression to forecast salaries based on factors such as education level and experience [1]. GitHub user anillava1999 utilized Random Forest to predict salaries considering a broader array of features, including 'company_rating', 'company_founded', 'competitors_count', 'company_sector', 'company_ownership', 'job_title', 'job_in_headquarters', 'job_seniority', and 'job_skills' [2]. These projects exemplify the diverse approaches in salary prediction using machine learning methods.

The Dataset

The dataset is named “Salary_Data” and was curated by user Mohith Sai Ram Reddy and two other collaborators on Kaggle [3]. According to the author, the dataset was retrieved from multiple sources, including surveys, job posting sites, and other publicly available sources. With a comprehensive compilation of 1,786 cleaned entries comprising general salary details, we opted for this dataset because of its comprehensive and diverse range of information sources. The attributes of the dataset are as follows:

1. Age - The age of the individual.
2. Gender - The gender of the individual (male, female, and other).
3. Education level - The level of education of the individual (Bachelor’s, Master’s, PhD, and highschool).
4. Job title - The job title or role associated with the reported salary.
5. Years of experience - The experience of the individual.

6. Salary - The monthly salary of the individual in Indian rupees.

After cleaning the data, distribution plots were made using Python library Seaborn to visualize the frequency and distribution of values for each of the six variables in the dataset.

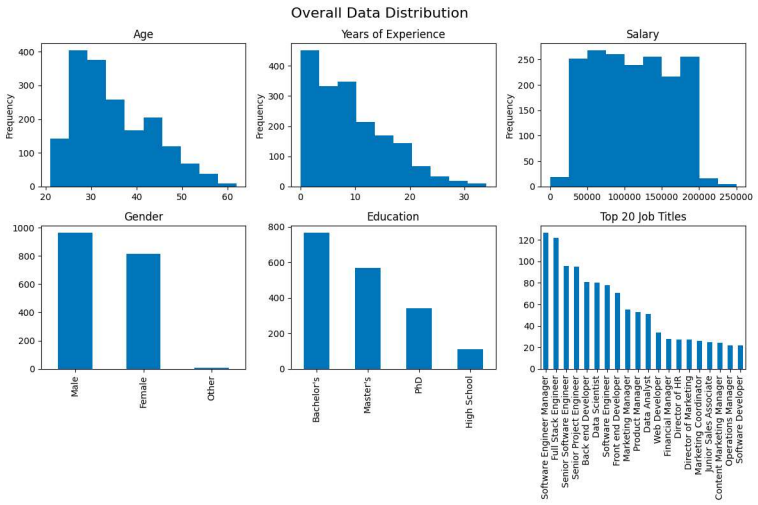

Fig. 1. Data Distribution

We noticed that the majority demographic of those who partook in the survey were 25-35 aged employed in technology-related professions. Notably, more than half of the top 20 job titles were directly associated with the technology sector. Since we were planning on using models for salary prediction, we proceeded to visualize the data distribution of the five other variables - gender, education level, age, and years of experience - against salary.

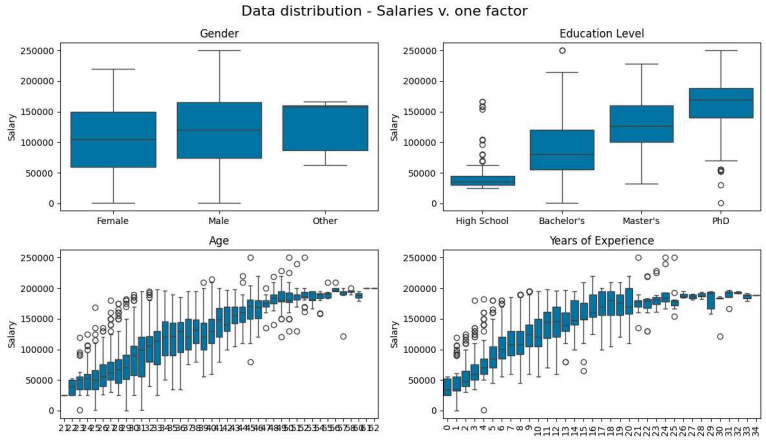

Fig. 2. Data Distribution of Variables Against Salary

As expected, several factors had a positive correlation with salary. Specifically, higher levels of education, age, and years of experience were associated with an increase in the income of the individual. However, we did notice that these relationships displayed a non-linear trend,

resembling more of a logarithmic curve as age or years of experience increased to the upper bound, the income growth would start to plateau.

Methodology

Linear Regression

To explore the relationship between variables and salary, we initially visualized a linear regression model using two variables. The resulting figures depicted R^2 scores of 0.67 and 0.72, indicating a moderately strong correlation between the variables chosen - age, years of experience, and education level - and salary.

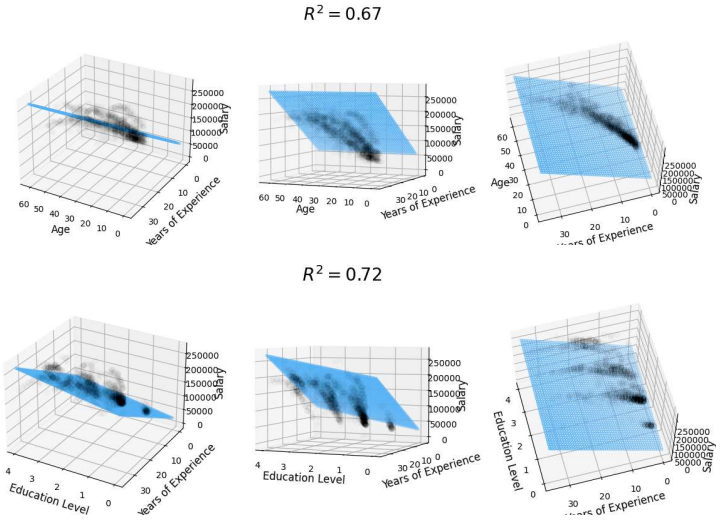

Fig. 3. 2-Variable Linear Regression Model

To create a more sophisticated model, we proceeded with a multivariate linear regression approach, incorporating five variables from the dataset to predict salaries. This involved a series of preprocessing steps including hyperparameter tuning, column transforming, one hot encoding, and standard scaling. Hyperparameter tuning aims to optimize the model's parameters for better performance, while column transforming, one hot encoding, and standard scaling are techniques used to prepare the data for modeling by addressing issues like categorical variables and feature scaling.

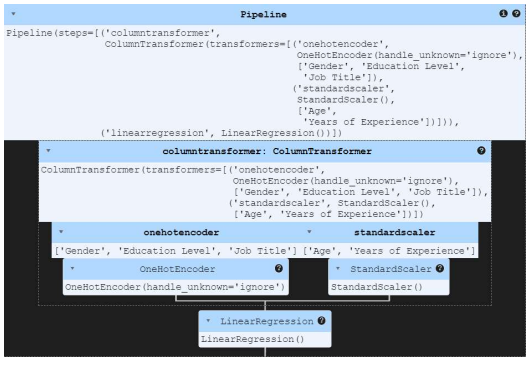

Fig. 4. Linear Regression Preprocessing Pipeline

Furthermore, the dataset was divided where 80% of the data was used to train the model and 20% of the data was used to test the model's performance. The resultant linear regression model achieved an R^2 value of 0.823.

Fig. 5. Actual vs. Predicted Salaries of the Linear Regression Model

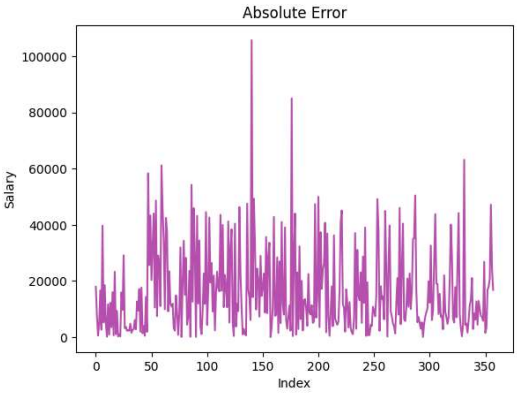

Fig. 6. Absolute Error of the Linear Regression Model

The model was used to predict salaries within the test data, and these predictions were compared against the actual results of the test data. Figure 5 visually represents the absolute differences between the actual and predicted results. Notably, the mean error was approximately 22,000, with particularly inaccurate estimates reaching as high as 100,000.

K-Nearest Neighbors

We started by creating the initial K-Nearest Neighbors model, which used euclidean calculations for its distance metric. We proceeded by creating a pipeline that incorporated a column transformer to standardize the data. All the categorical features were one hot encoded, while the numerical features were standardized with a standard scaler.

Fig. 7. K-Nearest Neighbors Preprocessing Pipeline

To optimize the model's predictive performance, we utilized hyperparameter tuning. A grid search cross validation with 5 folds was used to exhaustively search for the best n-neighbors and features to use for the best K-Nearest Neighbors model.

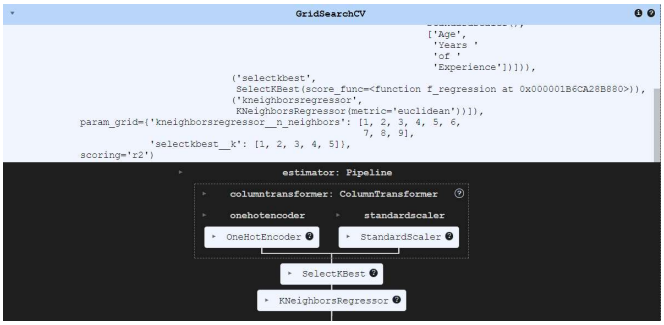

Fig. 8. GridSearchCV for Hyperparameter Tuning

The grid search cross validation found that the following parameters would produce the model with the best R2 score:

```
Best N-Neighbors: 9
Best Features: ["Age",  "Gender",  "Education Level", "Job
Title", "Years of Experience"]
```

Once the optimized parameters were used to train the final K-Nearest Neighbors model, the model was evaluated by using an 80:20 train:test split. The model was used to predict salaries within the test data, and these predictions were compared against the actual results of the test data.

Fig. 9. Actual vs. Predicted Salaries of the K-Nearest Neighbors Model

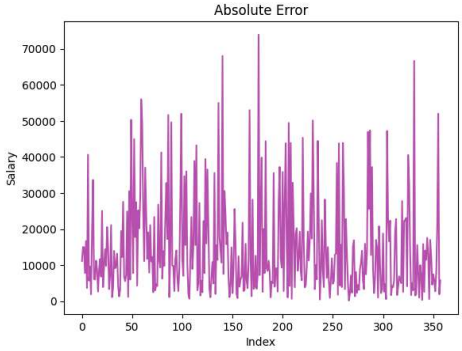

Fig. 10. Absolute Error of the K-Nearest Neighbors Model

Figure 10 visually represents the absolute differences between the actual and predicted results. Notably, the mean error was approximately 20,000, with particularly inaccurate estimates reaching as high as 70,000.

Random Forest

We started the Random Forest model by first creating an initial version of it using arbitrary hyperparameters. Using the same column transformer as before to standardize our data, we performed another grid search with the parameters needed for a RandomForestRegressor model. We started by using 100 decision trees to construct the random forest, and didn't bother to restrict the depth of each tree, the minimum samples to split a tree, etc. for simplicity's sake. Using 5-fold cross validation with an R^2 scoring metric, we fit our data on an 80:20 training-testing split and then extracted the hyperparameter values that made our model perform best.

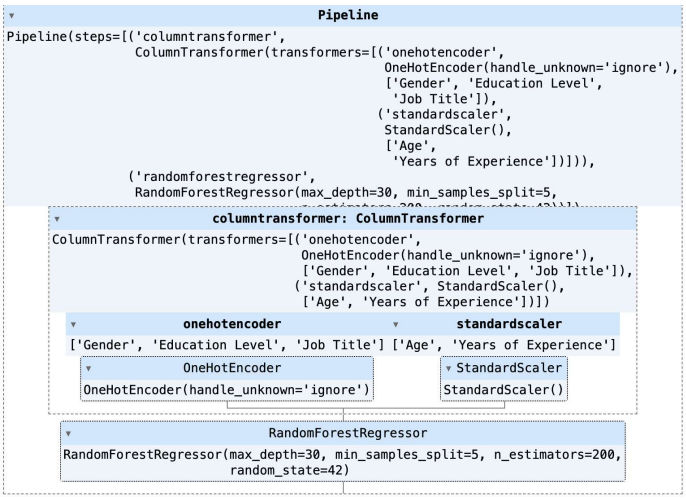

Fig. 11. Random Forest Preprocessing Pipeline

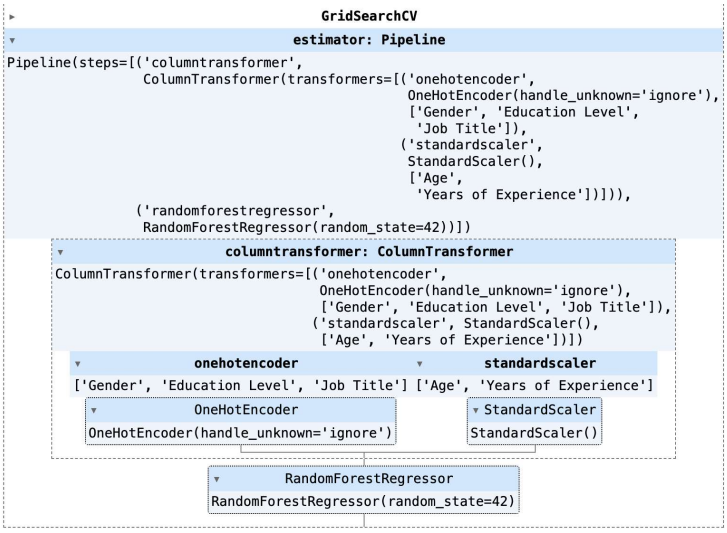

Fig. 12. GridSearchCV for Hyperparameter Tuning

In a similar fashion to our KNN model, the 3 most optimal hyperparameters that offered the highest information gain were years of experience, age, and job title.

Furthermore, here is what a sample decision tree looks like within the random forest model:

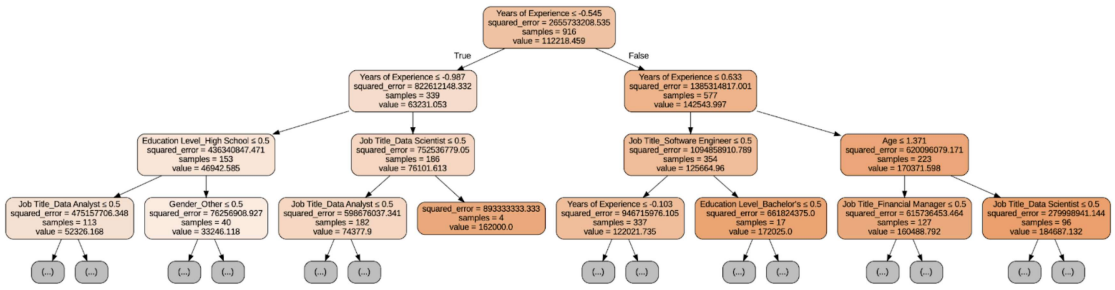

Fig. 13. Sample Decision Tree Within Random Forest

As we covered in class, the attributes in each path are determined by which ones have the most information gain. The root node is associated with Years of Experience because it has the highest predictive value, and has the value of -0.545. This represents how many standard deviations from the mean the tree should split the data by. In other words, if the z-score of the Years of Experience is less than -0.545, then the tree would select the left branch over the right one. In the figure, the depth is limited to 3 levels, but this process would continue for all the relevant features. Moreover, decision trees inherently perform feature selection through this process of selecting attributes with the highest information gain, so no further steps were necessary. It is also worth to note that the color intensity of each node tells us how much higher or lower a target variable within a branch is compared to the overall value of that variable in the dataset For example, we can see that the nodes on the third level progressively get darker from left to right. This means that paths that contain the nodes on the right side tend to have higher predicted salaries than the ones that take the left nodes.

We then proceeded to create a new model using the new hyperparameters we found and evaluated it on our training and testing data from before using an 80:20 split once again. If we plot the actual salaries against the predicted salaries like the previous two models then we can see that the trends are similar to the other two models, but there is slightly more overlap compared to the other models, indicating that the random forest model is a bit more accurate. The absolute error plot yields similar results to the KNN model.

Fig. 14. Actual vs. Predicted Salaries of the Random Forest Model

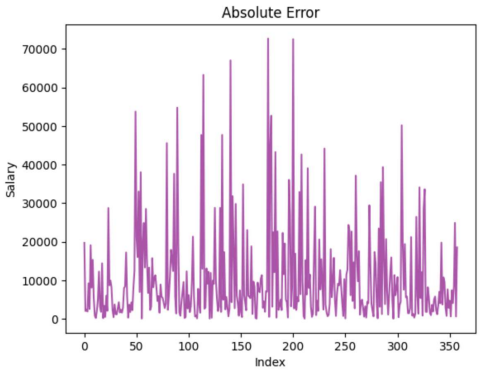

Fig. 15. Absolute Error Plot With Random Forest Mode

Conclusion

In all three models, it was clear that there was a correlation between using more features and a better R^2 score. Therefore, each feature in the dataset had a positive correlation with the salary target column.

Model	MSE	RMSE	R^2 Score
Linear Regression	503620589.51475126	22441.492586607317	0.8231840419395484
K-Nearest Neighbors	419640080.43175393	20485.11851153793	0.8526687264045383
Random Forest	269086535.38912493	16403.857332625303	0.905526512325799

From the evaluation scoring results, it was clear that there was a positive trend between the evaluation scores of Linear Regression and K-Nearest Neighbors and between K-Nearest Neighbors and Random Forest. Notably, the R^2 score from Linear Regression to K-Nearest Neighbors improved by 0.03, and the R^2 score from K-Nearest Neighbors to Random Forest improved by 0.05. Therefore, Random Forest was the best model to predict salaries based on age, gender, education level, job title, and years of experience, with a tolerance of about +/- 16,000 and a correlation fit of about 91%.

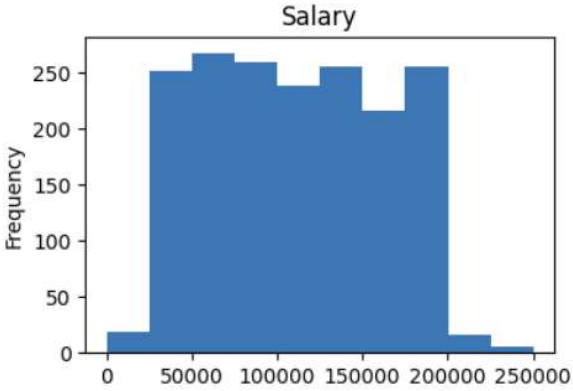

Fig. 16. Distribution of Salaries

Another thing to note is that while MSE scores in the hundred millions may seem high, it is quite small after noting that the RMSE scores are in the ten thousands. For instance, job sites, such as GlassDoor's "Embedded Systems Engineer Salaries" [4], provide salary data with ranges up to $70,000. While age, gender, education level, job title, and years of experience are excellent features to predict salary on, there are still some discrepancies within each of those cohorts. For instance, Figure 15 shows how many of the data points have salaries that are concentrated between 50,000 to 200,000. Therefore, having an RMSE score in the ten thousands, which reflects a tolerance of +/- the RMSE score, is a reasonable score.

References

[1] “Datahour: Salary analysis and prediction using ML,” Analytics Vidhya, https://datahack.analyticsvidhya.com/contest/datahour-salary-analysis-and-prediction-using-ml/.

[2] “Data Science Job Salary Predictions,” GitHub, https://github.com/anillava1999/Data-Scientist-Salary-Prediction/blob/main/Data%20Science%20Salary%20Prediction%20.ipynb.

[3] M. S. R. Reddy, “Salary_data,” Kaggle, https://www.kaggle.com/datasets/mohithsairamreddy/salary-data/data.

[4] “Embedded Systems Engineer Salaries,” GlassDoor, https://www.glassdoor.com/Salaries/embedded-systems-engineer-salary-SRCH_KO0,25.htm.