SALARY PREDICTION IN DATA SCIENCE INDUSTRY USING MACHINE LEARNING

Darshkumar Patel (235841910)

CP640: Machine Learning

Master of Applied Computing,

Faculty of Graduate and Postdoctoral Studies

Wilfrid Laurier University, Waterloo ON N2L3C5, Canada

1) Abstract

This project aims to predict salary ranges for Data Science roles using machine learning regression models. The key challenges in the Data Science job market include uncertainty in salary expectations, ambiguity in job titles, and the difficulty in identifying the most in-demand skills. By leveraging a dataset containing features like job title, company rating, skills required, and location, I applied several machine learning models—Linear Regression, Lasso Regression, Random Forest Regressor, and XGBoost Regressor. These models were then evaluated using metrics like Mean Absolute Error (MAE), R-squared (R²), and Root Mean Squared Error (RMSE). After tuning the Random Forest and XGBoost models with techniques like GridSearchCV and RandomizedSearchCV, XGBoost emerged as the best performer, with the lowest MAE and RMSE. The results demonstrate that the XGBoost model provided the most accurate predictions for salary estimation, offering valuable insights for both job seekers and employers in the Data Science industry.

2) Description of Applied Problem

With technologies getting far more enhanced and data being collected over time, data science has become one of the most rapidly growing and rapidly evolving fields globally. Companies are increasingly turning toward data analysts to help them interpret raw information and come up with the next course of action. The market for data science is however vast and complex with numerous titles and meanings such as data analysts, data scientists, data engineers and the machine learning scientists. Job seekers tend to struggle with understanding the employment market scenario such as:

i) Uncertainty in Salary Expectations: It leads to some problems when the executive is unable to set reasonable limits of pay for a given employment position.

ii) Role ambiguity: There could be confusion in naming the jobs and also the responsibilities that come in with the same.

iii) Market Demand Insight: Lack of clarity on the specific talent as well as uncertainty in predicting the most valued talents.

iv) Geographical Variations: The variation in the number of jobs and quality of the jobs available according to the geographic location of an individual.

This project allows the evaluation of conduct ranges by analyzing job postings, which are characterized by several parameters.

a) Skills: Specifying the skills searching for and important features of the job.

b) Familiarize with how the size, kind and place of the firm determines its employment offer.

c) Draw employment snapshot of Data Science market based on Glassdoor job postings.

Through the development of the machine learning model for earnings prediction, the initiative seeks to assist job seekers with such knowledge so that they could demand better job offers. By employing EDA, Data Visualizations and other concepts, I will be able to gather a lot of important data from this dataset.

---

3) Description of Available Data

This project's dataset consists of Data Science employment listings gathered from Glassdoor. I found this dataset on the kaggle website. For the data science job, I will be cleaning data, analyzing data, and predicting salaries.
So, I will be developing a salary prediction model, as well as some essential insights that can supply a wealth of information, among other topics. I will also be performing data cleaning, pre-processing, feature engineering, and outlier analysis.

3.1) Dataset Characteristics:

Size: The dataset contains several thousand job postings, providing a robust sample for analysis.
Diversity: Includes a wide range of companies, from startups to established corporations, across various industries.

The description of key features I have used in the project is explained in more detail in ***Appendix A***. I have also added some features which I created on my own in the above section (Like company age).

---

4) Analysis Techniques

In order to develop a robust predictive model for estimating salaries in data science roles based on job postings, a systematic approach (general approach) involving data preprocessing, feature engineering, model selection, and hyperparameter optimization was followed. Below is a comprehensive overview of the key steps taken during the analysis.

4.1) Data Preprocessing

Prior to training the predictive models, a series of preprocessing tasks were executed to ensure the dataset was both clean and ready for analysis. The dataset, consisting of job postings with variables like job title, company rating, location, and skills, underwent the following stages:

I) Data Cleaning:

- **Removing Duplicates and Handling Missing Data:** To minimize any skew in the analysis, entries that can be considered duplicates were deleted as well as rows with all missing values. If for example, the ‘Salary Estimate’ column had missing values then imputation techniques were used or if that was not possible then the entire row was dropped.
- **Eliminating Irrelevant Features:** Features that were not needed in the process of salary prediction such as the job posting ID were removed. This was used to eliminate noise in the dataset and remove unnecessary data which in the end enhanced the quality of the model.

II) Data Transformation:

- **Salary Parsing:** At first, it was needed to upgrade the salary data, which comes in the format of a range of numbers (e.g., “137K-171K (Glassdoor est.)”). For this data to be useful in training of models, the salary range was reduced to structured numerical values. From the range, the minimum and maximum salary, and the average salary were determined as the average of these two amounts.
- **Standardizing Company Names:** A new column was created to hold a copy of the company names without any of the rating information or location codes that may be appended to the company names.
- **State Field Extraction:** In the job location field, which included the full location of employment (e.g., city and state), only the state was selected for analysis. Further, a new binary column was introduced alongside indicating whether the job was located in the same city as the company headquarters.

III) Feature Engineering:

- **Seniority Levels:** Job titles were analyzed and grouped using keywords into categories like Junior, Mid and Senior jobs. It also enabled the categorization that was used in the subsequent analysis to factor seniority into the salary determination.
- **Skill Extraction:** Some binarized features were created to indicate if the job advert includes essential skills (Python, SQL, AWS, R) or not. Each of the skills was given value 1 if it was reported, and 0 if it was not.
- **Job Title Grouping:** These multiple job names like Data Scientist and Machine Learning Engineer were grouped under one common label in order to minimize the model distortion.

And many other new features and Data Transformation were done for getting better results (which is showed in code part).

4.2) Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to better understand the relationships between variables and uncover insights that would guide feature selection and model development.

I. Correlation Analysis:

- Correlation between some of them like job title, company rating, required skills, and salary was further analyzed using a correlation heatmap. Through this heat map we can understand the relationship between the features of the dataset.

II. Outlier Detection:

- The method of detecting and deleting outliers included Basic Statistics of salary like Interquartile Range (IQR). They reached the same conclusion, adding that by eliminating high or low values within a set, the model achieved better results because it did not have to take extra variables into consideration.

III. Visualizations:

- **Salary Distribution by Job Title:** As for this approach, an analysis of variance of the salary variable was done using a barplot and tables (pivot tables) to show the salary range within and between the different job title groups. From this visualization, we found that asserting positions like Senior Data Scientist and Data Science Manager attracted higher pay packages.
- **Salary Distribution by Company Size:** An example of the use of a histogram was applied to show how salary distribution depends on the size of the company. Employer size was identified to be positively related to levels of salaries being offered and this came with a wider disparity in salary levels among larger companies.
- **Skills vs. Salary:** A use of scatter plots aimed at identifying potential correlations between the presence of specific skills (Python, R, AWS, etc.) and the estimated salary rates. In the case of the present study, a direct positive relationship term was noted, implying that the higher skills needed was correlated with the number of salaries offers.

Apart from the above visuals I have created many other visual pie charts, bars, histograms etc using different features for getting better insights from those visuals. By going through the process of data cleaning, transformation, feature engineering and basic exploration, I was able to get insights. While preparing the model in the first phase we not only made sure that our data was ready for modeling, but also identified important features required for predicting salary in any future models.

4.3) Model Selection

In this project, several machine learning models were evaluated to determine their effectiveness in predicting salaries for data science roles. Each model was selected based on its unique strengths and suitability for the dataset, which includes a mix of categorical variables, non-linear relationships, and potential multicollinearity.

I. Linear Regression

Why Linear Regression?

Linear Regression was adopted as a reference for the project with the goal of making some predictions on the salary. It implies a direct relationship between input features and the target variable, i.e., average salary, perhaps one of the reasons why it is effective in presenting a simple picture of salary trends.

Characteristics:

- It is simple to apply and analyze as well.
- The relationship between the independent and target variables is linear.
- Gives an indication as to why and how each of the individual features affects the salary.

II. Lasso Regression

Why Lasso Regression?

Lasso Regression is a further refinement of linear regression while applying L1 regularization that helps make most of the feature coefficients equal to zero. This mechanism of feature selection reduces overfitting because in the high - dimensional framework that is inherent to our dataset, some features might be of little or inconsequential predictive value in the salary prognosis.

Characteristics:

- Contains built-in, performs feature selection and shrinks the coefficient of unimportant features to zero.
- Good in managing interaction between features.
- Helps avoid cases where a model fits data excessively because of enhanced penalties for big coefficients, such that the model is more generalized.

III. Random Forest Regressor

Why Random Forest?

Random Forest is a type of the ensemble learning algorithm; it is based on the decision trees concept indicating that an ensemble of decision trees is more accurate. It performs exceptionally well when there are correlated and cyclical dependencies between features and is also robust to noise and outliers.

Characteristics:

- Resistant to noise and outliers as well as overfitted.
- Can handle interaction between features and also the non-linear data relationships.
- The basic models can be applied to fit a large number of data, containing a combination of numerical and categorical variables.

IV. XGBoost Regressor

Why XGBoost?

XGBoost (Extreme Gradient Boosting) is a high-performance algorithm, which is a sequential learning model based on a decision tree. It is well understood, especially in its ability to operate on large, high-dimensional data spaces and has been found to be among the most efficient algorithms in learning competitions.

Characteristics:

- Designed to perform the operation as quickly as well as precisely as possible.
- It performs well even with missing data as well as outliers.
- It includes the formula L1 and L2 which helps to allow the model to avoid overfitting and thereby increase the model's performance in the real world.

Each model was chosen for its ability to handle the dataset's complexity and its potential for providing meaningful predictions in salary estimation.

4.4) Model Tuning

To further enhance the performance of the selected models, hyperparameter tuning was performed for both the Random Forest and XGBoost models. Hyperparameter tuning allows for the identification of the best settings for a model, improving prediction accuracy and reducing errors.

4.4.1) Random Forest Hyperparameter Tuning

Method Used:

The hyperparameters tuning for Random Forest used a method of GridSearchCV to conduct an exhaustive search of a pre-defined hyperparameter space. This has the advantage of searching, in a structured manner, all the permutations of the values set for specific parameters, so as to guarantee an exhaustive search for the best values. To check for the optimal levels of accuracy, cross validation was employed for prediction such that over fitting on the data set was avoided.

Key Hyperparameters Tuned:

- **n_estimators:** The number of trees within the forest. A quantitative value of flow rate was tried out from the range of 10-300.
- **criterion:** The function to evaluate the quality of splits. Examined with "squared_error" (mean squared error) and "absolute_error" (mean absolute error).
- **max_features:** The number of features to look at when trying to find the best split. The options considered were "auto", where it uses all the features and "sqrt", which uses square root of the number of features for dimensionality reduction and "log2".

4.4.2) XGBoost Hyperparameter Tuning

Method Used:

Due to a higher dimensionality of hyperparameters in XGBoost, RandomizedSearchCV was employed for finding the best hyperparameters. This method randomly selects a combination for the parameters, which makes it faster than the GridSearchCV. RandomizedSearchCV is particularly relevant if there are many parameters to optimize, in this way, the time required for the search is cut back while still finding the best values.

Key Hyperparameters Tuned:

- **n_estimators:** The number of boosting rounds; that is the number of trees involved in boosting. Tested in the range from 50 to 500.
- **max_depth:** The deepest level of every tree. The ranges which were tested were ranges from 3 to 10.
- **learning_rate:** The shrinkage parameter that regulates how much each tree is going to contribute to the final model. The analysed values ranged from 0.01 to 0.3.
- **subsample:** The percentage of the actual training cases used to build each tree was tested ranged from 0.6 to 1.0.
- **reg_alpha and reg_lambda:** L1 and L2 regularization terms that help to prevent overfitting.

4.4.3) Impact of Hyperparameter Tuning:

Optimization of hyperparameters led to a clear enhancement of the models' efficiency: Random Forest was chosen specifically for it, while XGBoost was as well. This optimization procedure was instrumental in error minimization, enhancement of model performance and provided a measure of good generality on unseen data.

4.5) Model Evaluation and Comparison

The performance of each model was evaluated using three key metrics: **Mean Absolute Error (MAE)**, **R-squared (R^2)**, and **Root Mean Squared Error (RMSE)**. These metrics help assess prediction accuracy, the ability of the model to explain the variance in the data, and the magnitude of prediction errors.

Mean Absolute Error (MAE): It Measures the variability of actual or predicted values about their average. Shows on average how far is the prediction off target or how accurate the predictions made are. Higher accuracy is closer to MAE hence, lower MAE is preferred.

R-squared (R^2): Shows the extent of the dependence of the variance of the dependent variable from the features. It Explains how well the model fits the data. More value of R^2 means closer fit.

Root Mean Squared Error (RMSE): Squares the residuals, averages them, and then takes square root of the outcome. Different from MAE since it heavily punishes large errors due to square rooting. It is Helpful in determining how well the model has been developed.

a. Linear Regression

- **MAE**: 144,658.90 — A large MAE suggesting that the actual salary predictions by the model have large errors.
- **R^2**: -5.93e+08 — means that in fact the model does not even explain any of the variance in the data and does worse than just predicting the mean salary.
- **RMSE**: 831,654.26 — A high RMSE fortifying the low prediction capability of the developed model.

The Linear Regression gave a lot of errors while fitting it, and the fit was worse than anything. In this data, however, it is not good for predicting salaries because it cannot capture the relationships in the data.

b. Lasso Regression

- **MAE**: 26.81 — Comparatively much better than Linear Regression suggesting improved salary prediction.
- **R²**: Further still, we get an R² of -0.06458 — while better than Linear Regression, this shows that the model still accounts for variance in data.
- **RMSE**: 35.23 —It is lower than the Linear Regression but just above the halfway mark which means that while the model is not fully exposing all the right patterns, it is not too bad either.

When applying Lasso Regression we got better performance over Linear Regression, with more accurate predictions; however, we again observed that getting all the interactions correct was not possible.

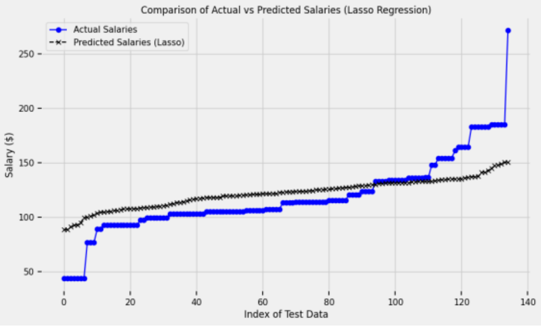

Figure 1. Actual vs Predicted Salaries(Lasso Regression)

c. Random Forest

- **MAE**: 26.87 — A little higher than Lasso, although it takes a step up from the basic models.
- **R²**: -0.11536 — Still slightly better than Lasso Regression, and the negative value of the R² indicate that the model is not a very good fit for this data.
- **RMSE**: 36.06 — Slightly elevated than Lasso and yet again depicting better performance than Linear and Lasso Regression.

Compared to Linear and Lasso Regression, Random Forest provides even higher accuracy but still has a problem incorporating the data's complexity. It has the potential, still the change needs more enhancement.

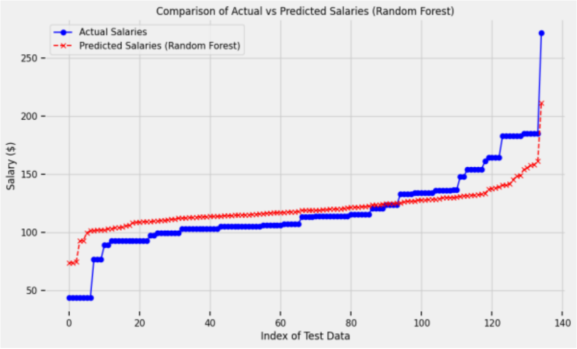

Figure 2. Actual vs Predicted Salaries(Random Forest Regressor)

d. Tuned Random Forest

- **MAE**: 26.27 — A little better because of hyperparameter tuning, executed with GridSearchCV in order to improve the choice of features and the model as a whole.
- **R²**: -0.063 — Lesser than the basic Random Forest model showing the

better settlement with the modified Rank.

- **RMSE**: 35.21 — This means that the Random Forest model has been simplified from the basic model, which supports the use of hyperparameters in the model to increase accuracy.

The Tuned Random Forest model also gives a better prediction as compared to random forest with high test accuracy and a better fitness. The optimization improved the generalizability of the model by capturing more patterns of data.

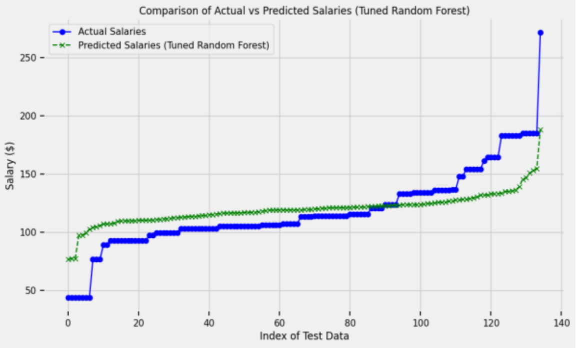

Figure 3. Actual vs Predicted Salaries (Tuned Random Forest Regressor)

e. Tuned XGBoost

- **MAE**: 25.97 — This index proves that the created model is more accurate in predicting the salary than all the other models.
- **R^2**: -0.0016 — This value is the smallest negative R^2 among all four models, which means that XGBoost shows how the model explains more variance of the data as compared to the other models.
- **RMSE**: 34.18 — The least RMSE, which supports that XGBoost yields the smallest of all prediction errors and is the most accurate model.

Tuned XGBoost is evidently the best among all the models in getting the most accurate predictions and optimal performance. The current article indicates that it is the most appropriate model for undertaking salary prediction on this dataset due to its high precision, accuracy as well as its high fitness.

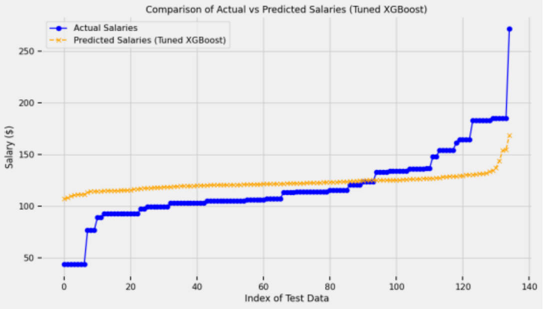

Figure 4. Actual vs Predicted Salaries (Tuned XGBoost Regressor)

5) Visualization Techniques

In this project, visualizations played a key role in both understanding the data and evaluating the model performance. The visualizations were divided into two categories: **Exploratory Data Analysis (EDA)** and **Model Results Evaluation**.

5.1) Visualizations for Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) visualizations were used to discover correlations between features and salaries, detect patterns, and improve the choice of features. Heatmaps, box plots, scatter plots, and histograms were among the visualizations I generated to help me analyze the dataset more thoroughly. These visualizations provided insights on how several characteristics, such as job title, firm size, and technical capabilities, influenced salary. Below are few examples of the EDA visuals I have used:

a) Correlation Heatmap

- **Purpose**: To visualize correlations between features and salary.
- **Insight**: The heatmap revealed strong correlations between technical skills (such as Python and SQL) and higher salaries. It also highlighted that seniority and job titles had a significant influence on salary expectations.

b) Salary Distribution by Job Title (Boxplot)

- **Purpose**: To compare salary distributions across different job titles.
- **Insight**: Senior roles, such as Senior Data Scientist, consistently had higher salaries. This helped identify roles with better compensation potential.

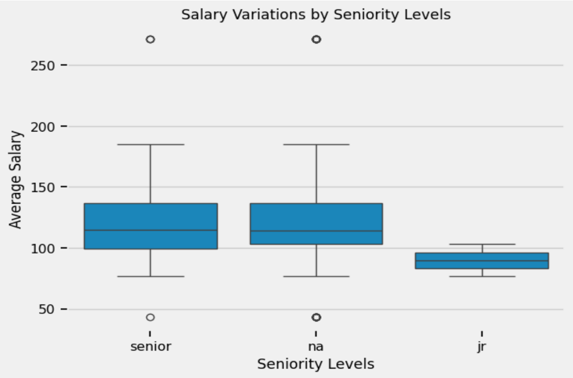

Figure 5. Salary Variation by seniority level

c) Salary Distribution by Company Size (Histogram)

- **Purpose**: To explore how salary varies by company size.
- **Insight**: Larger companies tended to offer higher average salaries, though salary variation was greater in larger firms compared to smaller companies.

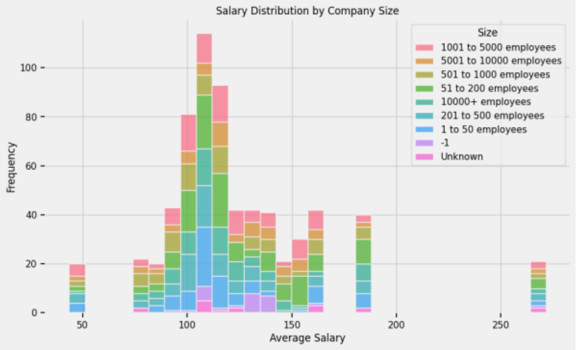

Figure 6. Salary Distribution by company size

d) Skills vs. Salary (Scatter Plot)

- **Purpose**: To analyze the relationship between specific skills (such as Python, R, AWS) and salary.
- **Insight**: High-demand skills like Python and AWS were positively correlated with higher salaries, particularly for data engineering and machine learning roles.

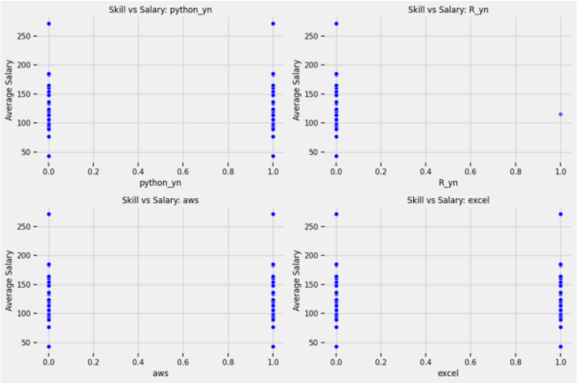

Figure 7. Salary as per skills

The above few visualizations provided critical insights into salary trends and the relationship between features, guiding feature selection and model-building decisions. I have added more in the project code.

5.2) Visualizations for Model Results

After training the models, various visualizations were created to evaluate model performance and identify areas where they excelled or faced challenges.

a) MAE and RMSE Comparison (Bar Plot)

- **Purpose**: To compare the error metrics (MAE and RMSE) across different models.
- **Insight**: XGBoost outperformed all models in both MAE and RMSE, while Linear Regression had the highest error metrics, confirming its ineffectiveness for predicting salaries in this dataset.

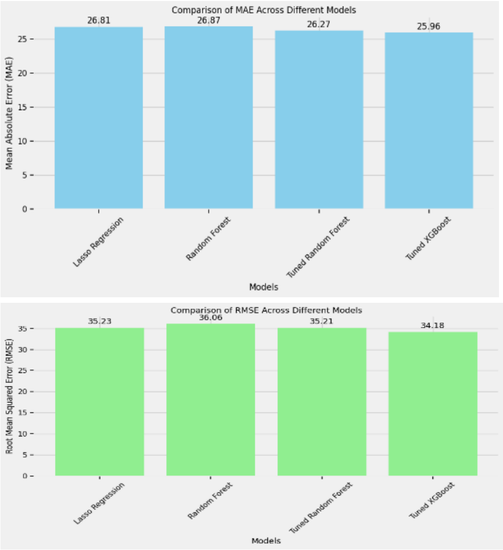

Figure 8. Comparison of RMSE and MAE using bar chart

b) Predicted vs Actual Salary (Scatter Plot)

- **Purpose**: To compare predicted salaries against actual salaries for each model.
- **Insight**: XGBoost's predictions closely matched the actual salaries, while Linear Regression's predictions were more dispersed, indicating poor model performance.

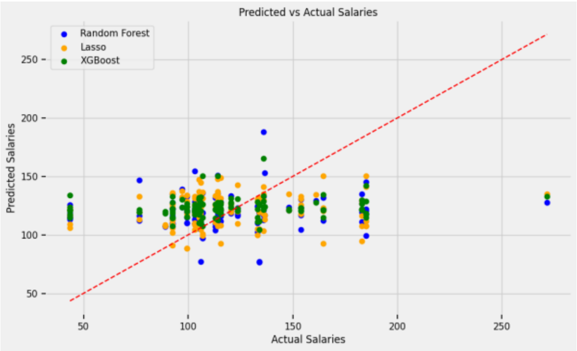

Figure 9: Predicted vs Actual Salaries for 3 models

c) Training vs Validation Loss (Line Plot)

- **Purpose**: To visualize model performance during training and check for overfitting or underfitting.
- **Insight**: Both Random Forest and XGBoost showed stable convergence with minimal overfitting. XGBoost maintained the best balance between training and validation loss, demonstrating superior generalization.

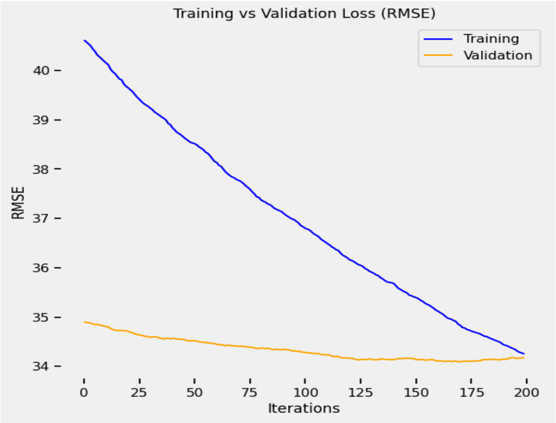

Figure 11. Training vs Validation Loss (RMSE) for XGBoost Model

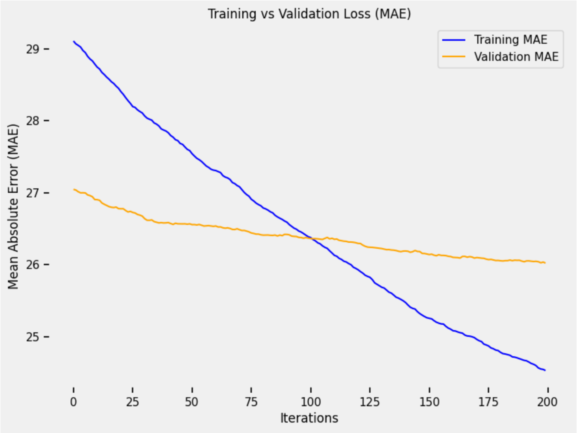

Figure 10. Training vs Validation Loss (MAE) for XGBoost Model

d) Error Distribution (Plot)

- **Purpose**: To visualize how prediction errors were distributed across different salary ranges.
- **Insight**: Both Random Forest and XGBoost performed well for mid-range salaries but struggled with extreme values, suggesting the need for further optimization to handle outliers effectively.

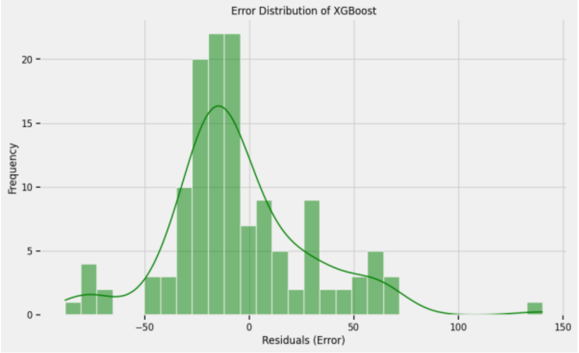

Figure 12. Error Distribution for XGBoost Model

6) Conclusion

In this project, the objective was to employ one or more machine learning algorithms to estimate salaries within the Data Science field. The features used were job title, skills, some features which were created using feature engineering and others for the Linear Regression, Lasso Regression and Random Forest models, while company size, industry, and job title were used in the XGBoost model to estimate salaries. The performance of the models was also analyzed with reference to the evaluation metrics such as MAE, R² and RMSE.

Key Findings:

- XGBoost was the most accurate of all models, in so far as accuracy of prediction was concerned. This model could estimate salaries effectively in the current dataset since it regularly presented the lowest MAE and RMSE values.
- During the EDA, certain assumptions and features of the data that is meaningful for analysis were identified. It emphasized the clear link between more technical proficiencies (such as Python, SQL) and a higher amount of pay and the role of job rank and firm size on wage rates.
- Nevertheless, it is worth emphasizing that the general performance of the models built was somewhat lower than expected. This can be attributed to factors such as the sheer size of the data, the fact that only a few features were considered by the model and the challenges of modeling the entire data based on all its feature relationships.

6.1) Future Improvements:

To further enhance model performance, several avenues for improvement can be explored:

- **Feature Expansion**: Including more features such as years of experience, education level, location, or certifications could improve model accuracy.
- **Advanced Techniques**: Exploring more advanced machine learning models, such as **neural networks** or **deep learning** techniques, could capture nonlinear relationships better.
- **External Data**: Integrating external datasets (e.g., regional salary trends or job market conditions) could provide a more holistic view of the factors influencing salaries.

7) References

1. Glassdoor Job Listings. https://www.glassdoor.com/
2. Scikit-learn Documentation. https://scikit-learn.org/
3. XGBoost Documentation. https://xgboost.readthedocs.io/
4. Pandas Library. https://pandas.pydata.org/
5. Kaggle Datasets. https://www.kaggle.com/
6. Python for Data Analysis by Wes McKinney.
7. Matbouli, Y. T., & Alghamdi, S. M. (2022). Statistical machine learning regression models for salary prediction featuring economy-wide activities and occupations. Information, 13(10), 495. https://doi.org/10.3390/info13100495
8. Jiang, W. (2024). The investigation and prediction for salary trends in the data science industry. Applied and Computational Engineering, 50, 8-14. https://doi.org/10.54254/2755-2721/50/20241102
9. Scikit-learn Documentation on Regression Metrics. https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics

Appendix A.

Key Features of the Dataset

Table 1. Description of Features

Variable	Description
***Job Title*:**	The title of the job posting.
***Salary Estimation*:**	The salary range offered for the position.
***Job Description*:**	Detailed description of job responsibilities and requirements.
Rating	Company rating based on employee reviews.
Company	Name of the hiring company.
Location	Geographical location of the job.
Headquarter	Location of the company's headquarters.
Size	Total number of employees in the company.
Type of Ownership	Classification of the company (e.g., public, private, non-profit).
***Industry and Sector*:**	The industry and sector in which the company operates.
***Revenue*:**	Company's total revenue.
***job_state*:**	U.S. state where the job is located.
***Same_state*:**	Boolean indicating if the job location is the same as the company's headquarters.
***age*:**	Age of the company in years.
***Skill Indicators*:**	Boolean columns indicating whether certain skills are required (e.g., Python, Excel, Hadoop, Spark, AWS, Tableau, Big Data).
***Seniority*:**	Boolean indicating if the position is at a senior level.