**Salary Prediction**

# Introduction:

**Short description of your data:** Our data set comprises information related to job listings, including attributes such as Job Title, Salary Estimate, Company Name, Location, Industry, Sector, and various skills (e.g., proficiency in Python, R, AWS, Excel). The primary goal is to predict the average salary (response variable) based on these attributes (predictor variables).

**Research question:** The key question we aim to answer is: *How can we accurately predict the average salary for job listings based on the given attributes?*

# Methods

In this study, we employed two different modeling approaches to predict average salary: **Linear Regression** and **Random Forest**. These models were chosen due to their complementary strengths in handling different types of data relationships and providing various levels of interpretability.

## Linear Regression Model:

**Reason for using Linear Regression:** Linear regression is a well-established statistical method used for understanding and predicting the relationship between a dependent variable (average salary) and multiple independent variables (predictors). It is particularly useful for its simplicity and ease of interpretation.

**Model Formula:** job\_state

**Considerations while fitting the model:**

* **Predictor Selection:** We examined the p-values of each predictor to determine their inclusion. Predictors with high p-values were excluded due to lack of statistical significance or low relevance from a domain knowledge perspective.
* **Model Assumptions:** Diagnostic plots were used to check for linearity, homoscedasticity, and normality of residuals. This ensured that the assumptions of linear regression were not violated.
  + **Residuals vs. Fitted:** To check for non-linearity and unequal error variances.
  + **Q-Q Plot:** To assess the normality of residuals.
  + **Scale-Location:** To check for homoscedasticity.
  + **Residuals vs. Leverage:** To identify influential data points.

**Data Splitting and Training:**

* **Data Split:** The full dataset was randomly split into 80% for training and 20% for testing to ensure the model's ability to generalize to new data.
* **Cross-Validation:** This process was repeated 10 times using different random splits to ensure robustness, and the mean test prediction error was calculated. The set.seed() function was utilized to ensure that both models worked on the same 10 train/test data subdivisions for fair comparison.

**Advantages:**

* **Simplicity and Interpretability:** The model is straightforward to implement and interpret. Coefficients directly show the impact of each predictor on the response variable.
* **Computational Efficiency:** Linear regression is computationally efficient, making it suitable for large datasets.
* **Statistical Significance Testing:** Provides statistical significance for each predictor, helping in understanding the relevance of each feature.

**Disadvantages:**

* **Assumption Dependence:** Linear regression assumes a linear relationship between predictors and the response, which might not always be the case.
* **Sensitivity to Outliers:** Outliers can significantly affect the model’s performance.
* **Multicollinearity:** High correlation between predictors can lead to unstable estimates and affect the model's reliability.

A screenshot of a computer code

Description automatically generated

## Random Forest:

**Reason for using Random Forest:** Random Forest is an ensemble learning method that enhances predictive accuracy and robustness by building multiple decision trees and combining their outputs. It is well-suited for handling complex interactions between variables and reducing overfitting.

**Model Formula:**

avg\_salary ~ Rating + Founded + job\_state+ Sector +job\_state +same\_state+ python\_yn…+excel

**Considerations while fitting the model:**

* **Parameter Tuning:** We focused on selecting the optimal number of trees and the number of variables to try at each split. Different configurations were tested to balance model accuracy and interpretability.
  + **Number of Trees:** We tested models with 100, 200, 300, and 500 trees to find the optimal balance between accuracy and computational cost.
  + **Variables at Each Split:** We experimented with different numbers of variables considered at each split to enhance model performance.
* **Model Interpretability:** Variable importance plots were generated to identify the most significant predictors influencing the average salary.

**Data Splitting and Training:**

* **Data Split:** Similar to the linear regression model, the dataset was randomly split into 80% for training and 20% for testing.
* **Cross-Validation:** The same 10-fold cross-validation was applied, recording the mean test prediction error for different numbers of trees to identify the best configuration. The set.seed() function ensured consistency across model comparisons.

**Advantages:**

* **Handling Non-Linearity:** Random Forest can model complex, non-linear relationships between predictors and the response variable.
* **Robustness to Overfitting:** By averaging multiple trees, Random Forest reduces the risk of overfitting compared to single decision trees.
* **Variable Importance:** Provides insights into the importance of each predictor, aiding feature selection and interpretation.

**Disadvantages:**

* **Computationally Intensive:** Building and combining multiple trees require more computational resources and time compared to linear regression.
* **Less Interpretability:** The model's interpretability is lower compared to linear regression, as it involves many trees and complex interactions.
* **Parameter Tuning:** Requires careful tuning of hyperparameters (e.g., the number of trees, the number of variables tried at each split) to achieve optimal performance.

By leveraging both models, we aim to capitalize on the strengths of each approach: the interpretability and simplicity of linear regression and the flexibility and robustness of the Random Forest model. This combination allows for a comprehensive analysis of the factors influencing the average salary while ensuring reliable and accurate predictions.



# Results

## Linear Regression Results

**Model Performance:**

* **Mean MSE across 10 iterations**: 1294.625

The linear regression model was evaluated using 10-fold cross-validation. The mean squared error (MSE) for each iteration was as follows:

* MSE 1: 977.8026
* MSE 2: 1577.58
* MSE 3: 1225.935
* MSE 4: 1497.378
* MSE 5: 1487.562
* MSE 6: 1043.323
* MSE 7: 1543.658
* MSE 8: 1032.426
* MSE 9: 1261.761
* MSE 10: 1298.829

The mean MSE across all iterations was 1294.625, indicating the model's prediction accuracy on unseen data.

**Diagnostic Plots:** The diagnostic plots provided insights into the model’s assumptions and potential issues:

* **Residuals vs. Fitted:** This plot checks for non-linearity and unequal error variances. The residuals are randomly scattered around the horizontal axis, indicating a linear relationship.
* **Q-Q Plot:** The quantiles of the residuals are plotted against the quantiles of a normal distribution. The points lie approximately along the line, suggesting normality of the residuals.
* **Scale-Location Plot:** This plot checks for homoscedasticity (constant variance of residuals). The residuals appear randomly spread, supporting the assumption of homoscedasticity.
* **Residuals vs. Leverage:** This plot identifies influential data points. Points with high leverage and standardized residuals are identified, indicating potential outliers.

**Summary:** The linear regression model provided reasonable prediction accuracy with a mean MSE of 1294.625. The diagnostic plots confirmed that the model assumptions were generally met, although some influential points were identified.

**Diagnostic Plots:**

A group of graphs showing different values

Description automatically generated

## Random Forest Results

**Model Performance:**

* **Mean MSE across different configurations:**
  + 100 trees: 1232.145
  + 200 trees: 1147.989
  + 300 trees: 1047.64
  + 500 trees: 1199.992

The Random Forest model was evaluated using different numbers of trees to find the optimal configuration. The lowest MSE was achieved with 300 trees, yielding an MSE of 1047.64.

**Variable Importance:** The variable importance plot shows the predictors that had the most significant impact on the average salary:

* **Job State:** The state where the job is located was the most significant predictor.
* **Sector:** The sector of the industry had a substantial impact on salary predictions.
* **Founded:** The year the company was founded also contributed significantly to the predictions.
* **Rating:** Company rating influenced salary predictions.
* **Python\_yn, Excel, Same\_State, Spark, AWS, R\_yn:** Skills and other attributes were also important predictors.

**Error vs. Trees Plot:** The error rate decreased as the number of trees increased, stabilizing around 300 trees. This indicates that increasing the number of trees beyond 300 does not significantly improve the model’s performance.

**Summary:** The Random Forest model outperformed the linear regression model with a lower mean MSE of 1047.64 when using 300 trees. The model effectively captured complex interactions between predictors and provided valuable insights through variable importance measures.

**Variable Importance Plot and Error vs. Trees Plot:**

A graph of a state

Description automatically generated with medium confidence

# Conclusion:

From the results, it is evident that the Random Forest model is more effective in predicting average salaries compared to the Linear Regression model. The Random Forest model not only provided a lower mean MSE but also highlighted the most significant predictors, offering deeper insights into the factors influencing job salaries.

# Bibliography:

The dataset for this project is from Kaggle: <https://www.kaggle.com/datasets/thedevastator/jobs-dataset-from-glassdoor/?select=salary_data_cleaned.csv>