Identifying Characteristics of High-salary Data Science Related Job Postings

JSC370H1 Final Report

Elaine Dai

2024-04-30

1. Introduction

In the rapidly evolving job market, the role of a data science related job has emerged as a pivotal position within companies. As a student approaching the completion of data science degree, I am preparing to step into the professional world, and get interested in understanding the landscape of data science related positions. Therefore, this project aims to dig into the characteristics of data analyst related job postings, with a specific focus on uncovering the markers that signal high-salary opportunities. By digging into this, we can gain insights into the attributes and qualifications that employers value most for.

This project seeks to answer the question: "What characteristics of data science related job postings are indicative of high-salary positions?". The dataset utilized in this analysis was acquired from Kaggle, specifically from the "LinkedIn Job Postings" dataset by user Arsh Kon[1]. This dataset comprises a wide range of United States job postings on LinkedIn in 2023, including various fields such as job title, company, location, salary, and more.

Some columns that are deemed relevant to the topic are:

- job_id: Job ID as defined by LinkedIn
- company_id: Company ID as Defined by LinkedIn
- title: Job Title
- description: Job Description
- max_salary: Maximum Salary
- med_salary: Median Salary
- min_salary: Minimum Salary
- pay_period: Pay Period
- formatted_work_type: Work Type
- location: Job Location
- formatted_experience_level: Experience Level Required

2. Methods

First I loaded libraries and the data set.

2.1 Data Cleaning and Wrangling

Upon acquiring the dataset, I examined the structure of the data: the types of each column and any missing values. The target attribute is the salary, so this preliminary examination is focused mainly on salary. The examination revealed that salary information was provided either in a range format (minimum and maximum salary) or a median value (med_salary), and not all entries had a valid salary. To address this, a new column, salary, was created by averaging the min_salary and max_salary values for each entry where med_salary was missing. 2087(the average number of work hours in a year) is multiplied to the hourly based salary, in order to convert salary information from an hourly to a yearly basis for those entries listed with hourly rates. Another modification involved extracting the state information from the column location, for future analysis.

The further cleaning entailed filtering the dataset to retain only those job postings that are directly related to data roles. This was achieved by keeping rows where the job title included the term "data". After examining the summary statistics for salary, 4 outliers with unreasonably low annual salary have been removed. And the final stage is removing all job postings with na value and selecting specific columns that were deemed relevant for the analysis.

2.2 EDA

Summary Statistics: salary I extracted the summary statistics for salary and formulated the table using the kable. To further explore the categorical variables within the dataset, I implemented a visualization function using the ggplot2 library. The function was designed to produce two types of visual representations for each categorical variable: a bar plot and a box plot.

Exploratory Graphs: word type, state, experience level The function first groups the data by the categorical variable provided, summarises the data to get counts of each category within the variable, and then creates a barplot using ggplot2 with categories on the y-axis (due to coord_flip()) and their respective counts on the x-axis. Then a boxplot is created with the categorical variable on the y-axis (again, due to coord_flip()) and salary on the x-axis. The boxes are colored based on the categorical variable to differentiate between categories visually.

Text Mining: description An additional analysis was incorporated by utilizing text mining techniques to dissect job descriptions. The tidytext package was used to tokenize the text, allowing for the identification and visualization of the most frequent tokens. Common stopwords were removed, including universally frequent English words and additional terms like "years", "will", "work", "job" and "role".etc which are expected to be recurrent in job descriptions but offer little analytical value. Additionally, any tokens containing numbers were filtered out to focus purely on textual data. The text data was then tokenized into bigrams and trigrams to facilitate a granular analysis of phrase patterns within the job descriptions. Word/phrase count bar plots and a word cloud were created for visualization.

2.3 Model Preparation

For preparing the data specifically for modeling purposes, I concentrated on our main interest, full-time job postings only. I also modified the "state" column to classify the states with too few observations into "other" to avoid splitting error. A unique identifier (doc_id) was assigned to each job posting to facilitate individual tracking through subsequent steps.

A significant portion of this phase involved processing the text data within job descriptions using TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF is a statistical measure used to evaluate the importance of a word to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

This method was chosen because it effectively highlights the most relevant words in job descriptions, which are likely indicative of the skills and responsibilities valued in higher-paying roles. Words typical across many job descriptions or irrelevant (such as common English stopwords and numbers) were filtered out to refine the analysis.

Following the computation of TF-IDF scores, these were aggregated for each document, then pivoted to create a wide-format dataset where each row represents a job posting and each column represents a word's TF-IDF score. This data was then merged back with the original dataset, preserving the essential variables such as salary, state, and experience level, and ensuring that all categorical variables were appropriately encoded as factors for analytical consistency.

Another crucial step undertaken was the normalization of the dependent variable, `Salary`. Given the significant variance observed in salary data, with a mean salary of $124,343.88, and considering the scale disparity between salary figures and the TF-IDF values (which are often less than 0.02), it was necessary to transform the salary variable to stabilize variance and bring it closer in scale to other predictors. This normalization aids in enhancing the performance of the predictive models, particularly those that assume or benefit from normality in the input distribution. The transformation applied was a logarithmic transformation, specifically `log(Salary + 1)`, which is commonly used to manage skewed data by compressing the scale of the salary distribution.

The final dataset was split into training and test sets, with 70% of the data allocated for training to build the predictive models and 30% reserved for testing to evaluate model performance.

2.4 Modeling Building

In this section, we delve into the specifics of the machine learning models developed to predict high-salary data science jobs. The following models were constructed using various statistical and machine learning approaches to capture different aspects of the data and evaluate their predictive performance. All models used `Salary` as the dependent variable, and all other features(`state`, `formatted_experience_level` and words tf-idf values) in the dataset as predictors. A Variable Importance graph was plotted for every machine learning model to identify which features contributed most to predicting Salary and provide insight into the target question.

2.4.1 Linear Regression The baseline model is a simple Linear Regression, serving as a benchmark for more complex models. Linear regression was chosen for its interpretability and efficiency in establishing a linear relationship between the independent variables and the target variable, Salary. The model was specified as: `lr_model <- lm(Salary ~ ., data = train)`.

2.4.2 Regression Tree A Regression Tree was built to handle non-linear relationships and interactions between variables better than linear models. The initial tree was constructed using the rpart package with an ANOVA method, setting a complexity parameter (cp) for pruning: `tree_model <- rpart(Salary ~ ., method = 'anova', control = rpart.control(cp=0.01), data = train)`. Optimal pruning was conducted based on the smallest cross-validated error to avoid overfitting, using the following code to determine the best cp and prune the tree.

2.4.3 Bagging The Bagging model, implemented via the `randomForest` package, aims to reduce variance and prevent overfitting by combining the predictions of multiple decision trees constructed on different subsets of the dataset: `bagging_model <- randomForest(Salary ~ ., data = train, importance = TRUE, na.action = na.omit)`.

2.4.4 Random Forest An extension of the bagging technique, the Random Forest model, uses a similar ensemble method but introduces more randomness by selecting random subsets of features for splitting in each tree: `rf_model <- randomForest(Salary ~ ., data = train, mtry = sqrt(ncol(train)), importance = TRUE, na.action = na.omit)`. The number of variables tried at each split, mtry, was set to the square root of the total number of variables in the dataset.

2.4.5 Gradient Boosting Gradient Boosting was employed to build a predictive model in a stage-wise manner. It optimizes a loss function over weak learners (trees), with each subsequent tree correcting errors made by the previous ones. The `gbm` package was used for the implementation of this model: `boosting_model <- gbm(Salary ~ ., data = train, distribution = "gaussian", n.trees = 1000, interaction.depth = 1, shrinkage = 0.01, cv.folds = 5)`. The model parameters included the number of trees (n.trees), interaction depth, and the shrinkage rate, which controls the learning rate of the procedure. A grid search was applied on the shrinkage parameter for the best training performance.

2.4.6 XGBoost Finally, the XGBoost model, an advanced implementation of gradient boosting, was utilized for its performance and speed. This model was tuned using a grid search approach over multiple hyperparameters. The `caret` package facilitated the training control and grid search to find the optimal settings for parameters such as learning rate (`eta`), tree depth (`max_depth`), and others.

3. Results

3.1 EDA Results

After cleaning and wrangling, the cleaned dataset contains 297 observations and 6 variables with no missing values. The summary statistics for the numeric variable salary and the exploratory plots are given below.

Statistic	Value
mean	124343.88
median	123133.00
sd	54841.42
min	6769.50
max	400000.00

Table 1: Summary Statistics for Salaries

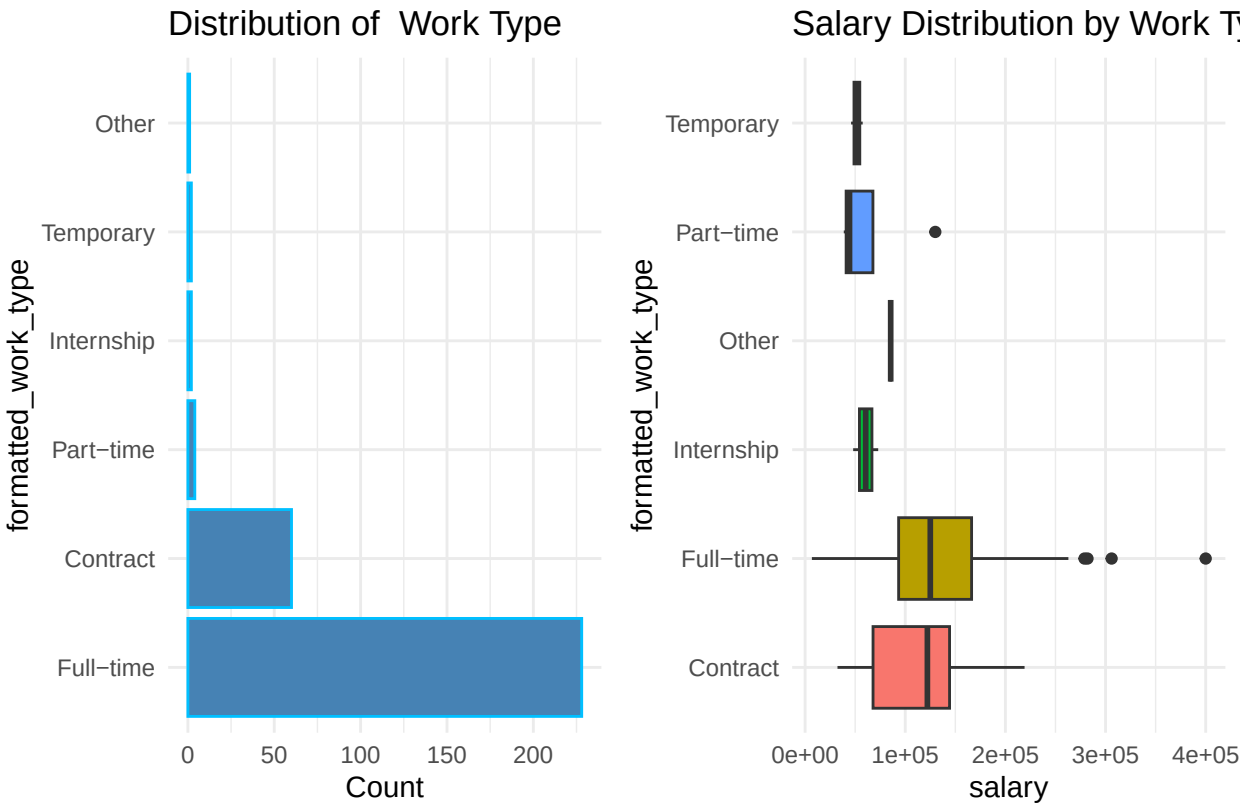

Figure 1: EDA for Work Type

In the work type category, full-time positions dominate the job market. Internships, part-time and temporary jobs are very less frequent. Salary-wise, full-time positions also lead with higher pay. Contract roles offer lower median salaries than full-time positions but are still well above temporary, part-time, and internship categories, which present the lowest pay. This is a predictable trend.

Figure 2: EDA for Experience Level

Regarding experience levels, mid-senior jobs are the most abundant, entry-level positions and associate follow in frequency. However, the high-ranking executive and director positions are rare, which reflects their specialized and leadership-focused nature.

Salaries increase notably with experience. Top-tier roles like executives enjoy the highest salaries, and understandably, those at the entry-level earn the least. This gradient in pay is expected, aligning with the increased responsibilities and expertise required at higher levels.

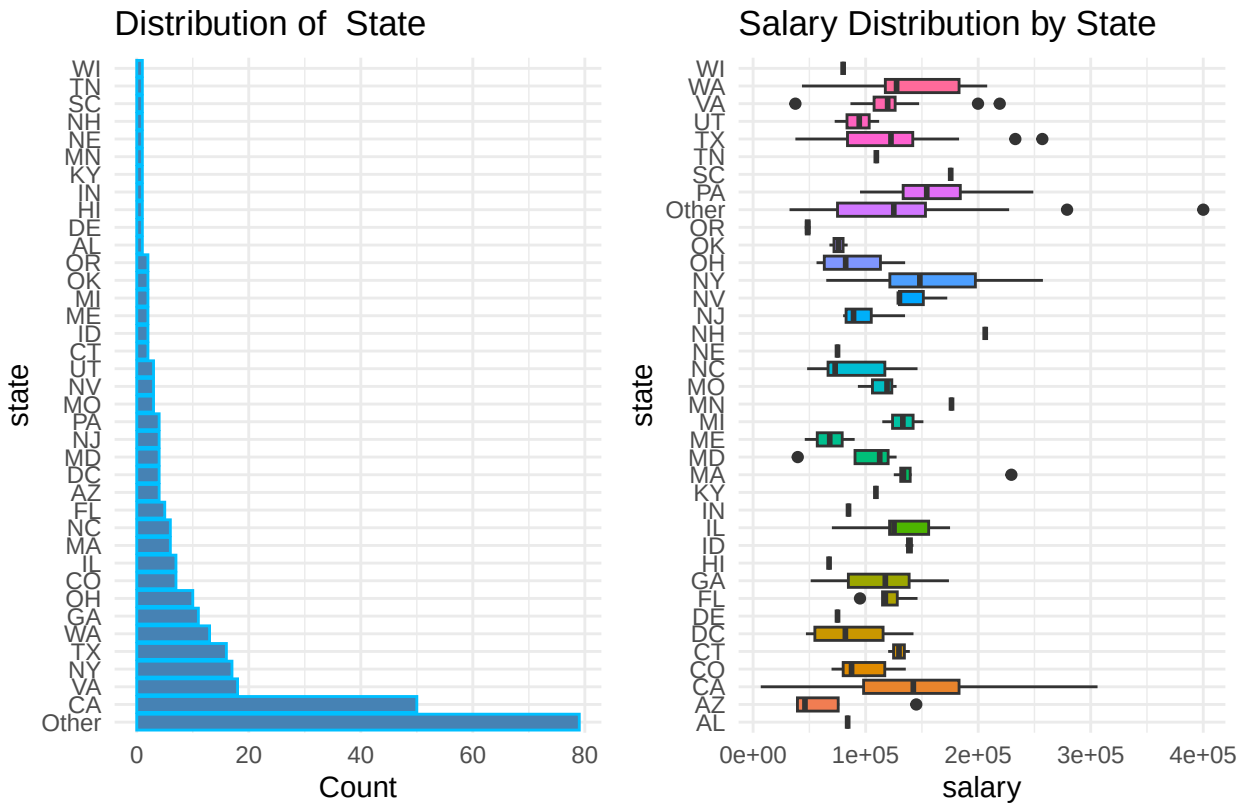

Figure 3: EDA for State

A state-wise look at job counts places California at the front, suggesting a bustling job market. New York and Texas also show significant job availability. Salaries by state reveal disparities that could be influenced by living costs or industry concentration, with places like California and New York showing higher median salaries. Other states exhibit a broad range of salaries, pointing to diverse economic landscapes and job sectors within each state.

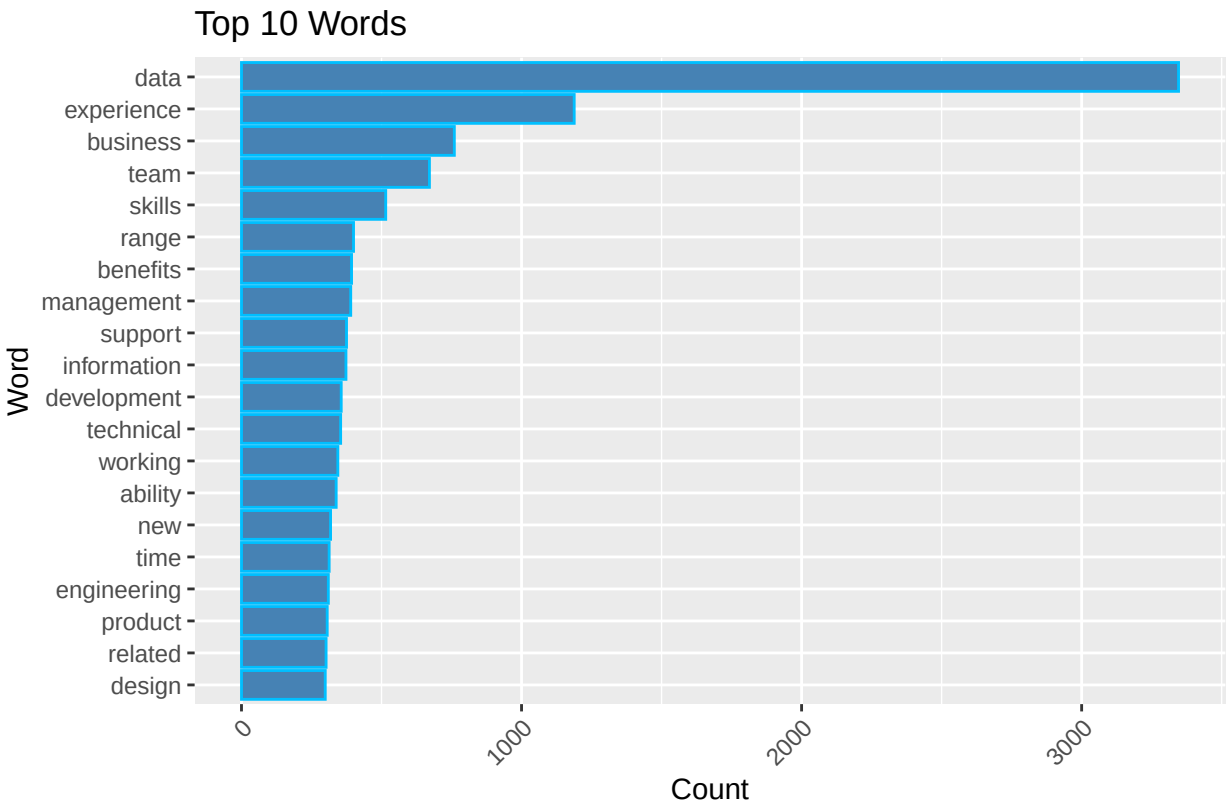

Figure 4: 10 words in job descriptions with top frequency

Figure 5: Word Cloud of Single Terms

This bar graph displays the most frequently occurring individual words within job descriptions. The words with highest frequencies includes “data”, “experience”, “business”, “team”, “skills”, highlighting the general importance of these aspects in the professional environment.

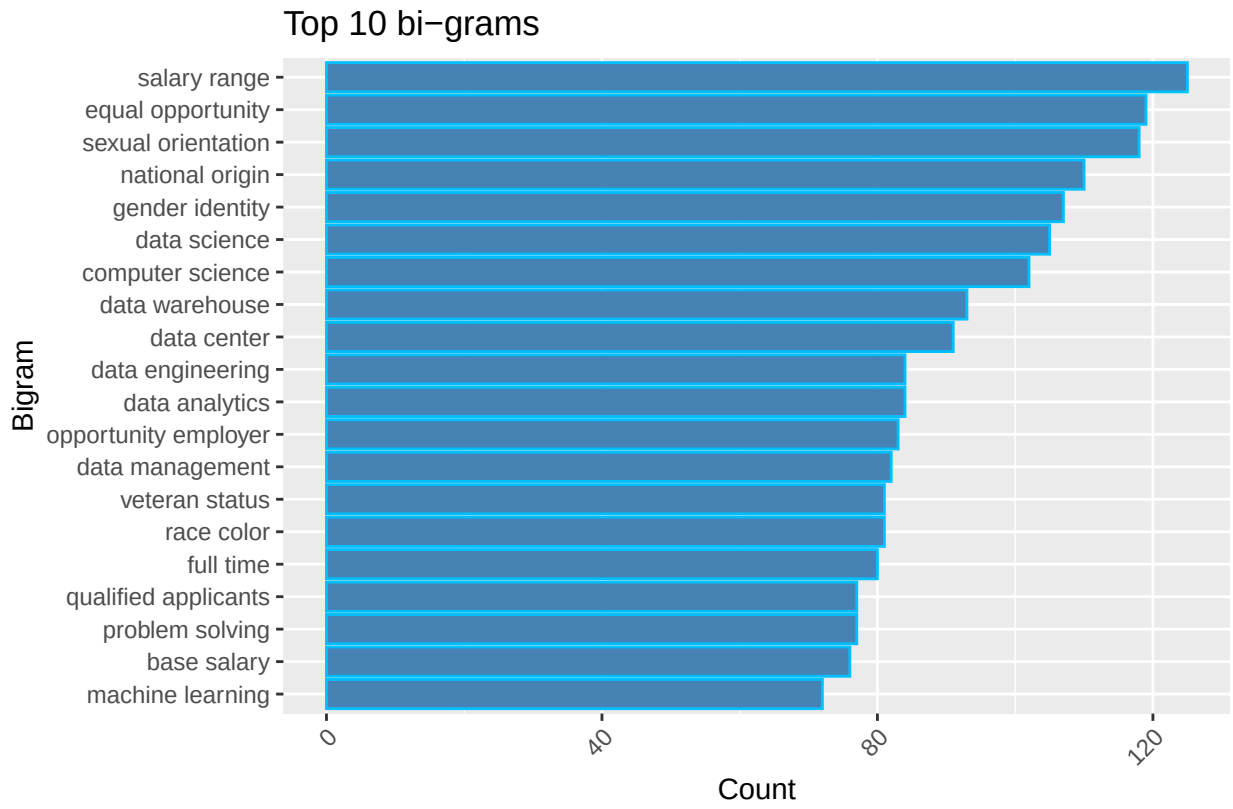

Figure 6: 10 bi−grams in job descriptions with top frequency

The bigram (two-word phrase) frequency graph sheds light on common pairings such as “machine learning”, “base salary”, “computer science” and “problem solving”. These reflect specific skills, compensation expectations, and competencies valued in the job market. The relatively even distribution suggests no overwhelming focus on a particular phrase but instead a variety of important attributes and benefits. Noticeably gender and equality concepts are brought up frequently, indicating the significant emphasis on diversity and inclusion within the job market.

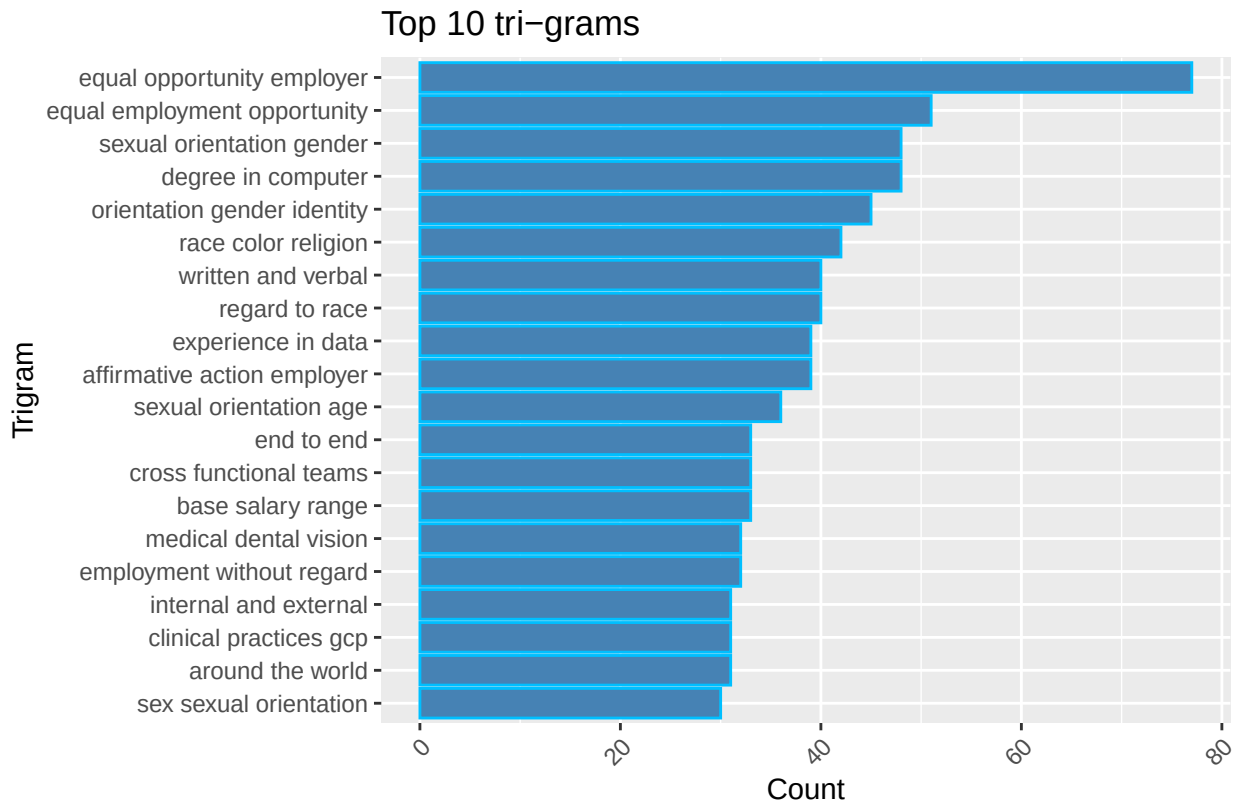

Figure 7: 10 tri−grams in job descriptions with top frequency

The trigram (three-word phrase) bar graph reveals frequent phrases with a more contextual understanding, such as 'equal opportunity employer', which denotes a commitment to workplace equality. The prevalence of phrases like 'race color religion', and 'gender identity' points to a focus on diversity and inclusion in hiring practices.

3.2 Model Results

3.2.1 Evaluating Metrics

Method	MSE	R_squared
Linear Regression	5.2522743	-16.4884891
Regression Tree	0.3273089	-0.0898400
Bagging	0.2049584	0.3175502
Random Forest	0.1957324	0.3482701
Boosting	0.2171150	0.2770726
XGBoost	0.2258173	0.2480966

Table 2: Evaluating Metrics for Model Comparison

In the evaluation of predictive models for determining high-salary positions within data science fields, a variety of advanced statistical techniques were employed. The models' performances were measured primarily using Mean Squared Error (MSE) and R-squared values, which are standard metrics for regression analysis.

Linear Regression and Regression Tree: The baseline model, Linear Regression model has the highest MSE (5.25227), which indicates that it was the least accurate model in predicting the salary. Regression Tree

showed a considerable improvement over Linear Regression, with an MSE of 0.32731. However, it still faced limitations due to its simpler, hierarchical decision-making structure, which may not capture all the nuances in the data. Linear Regression and Regression Tree models demonstrated inadequate fit, also indicated by their negative R-squared values of -16.488 and -0.089, respectively. This suggests that these models were outperformed by a trivial model that simply predicts the mean salary, indicating a poor explanatory power and lack of fit to the variance within the salary data.

Ensemble Methods – Bagging, Random Forest, and Boosting: Conversely, the ensemble methods, specifically Bagging and Random Forest, showed substantial improvements in predictive accuracy. The MSEs for Bagging (0.21406), Random Forest (0.19573), and Boosting (0.21711) were much lower, reflecting more accurate and reliable predictions. Notably, the Random Forest model achieved the lowest MSE, underscoring its robustness and effectiveness in handling diverse and complex datasets. These models achieved positive R-squared values of 0.287, 0.348, and 0.277, respectively, indicating a moderate fit to the data. Such improvements underscore the value of ensemble learning in handling complex datasets with intricate variable interactions and high variability.

XGBoost: The XGBoost model, a gradient boosting framework known for its efficiency and effectiveness, recorded an R-squared of 0.248, which, while lower than other ensemble methods, still represents an improvement over the basic regression approaches. This model's ability to handle various data shapes and types, along with its robustness to overfitting, makes it a viable option for further tuning and exploration.

3.2.2 Variable Importance Analysis of Variable Importance across models highlights critical predictors influencing salary outcomes in the data science domain.

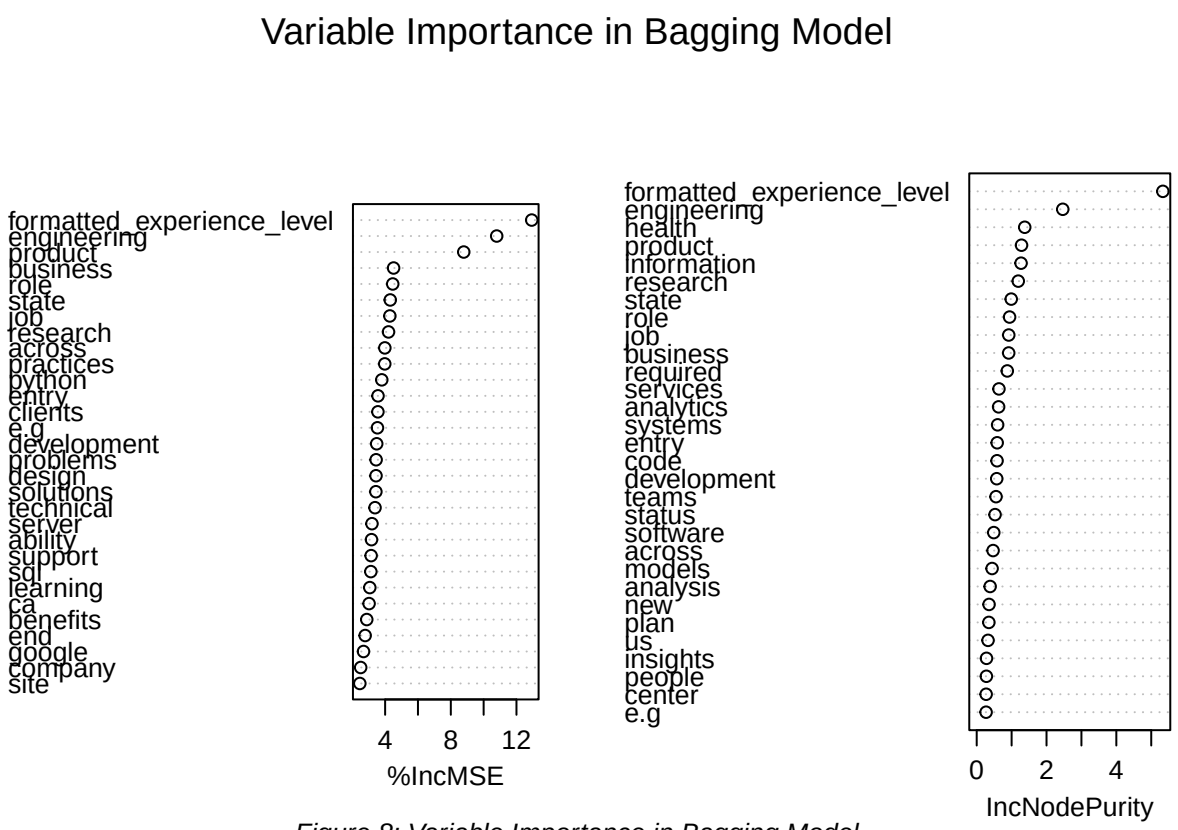

Figure 8: Variable Importance in Bagging Model

The most influential variables in the Bagging model are: formatted_experience_level, product, engineering, entry, research, role, across, IQ, status and problems. In this model, experience level and specific job functions

like product development and engineering significantly impacted salary predictions, experiences like research are also emphasized.

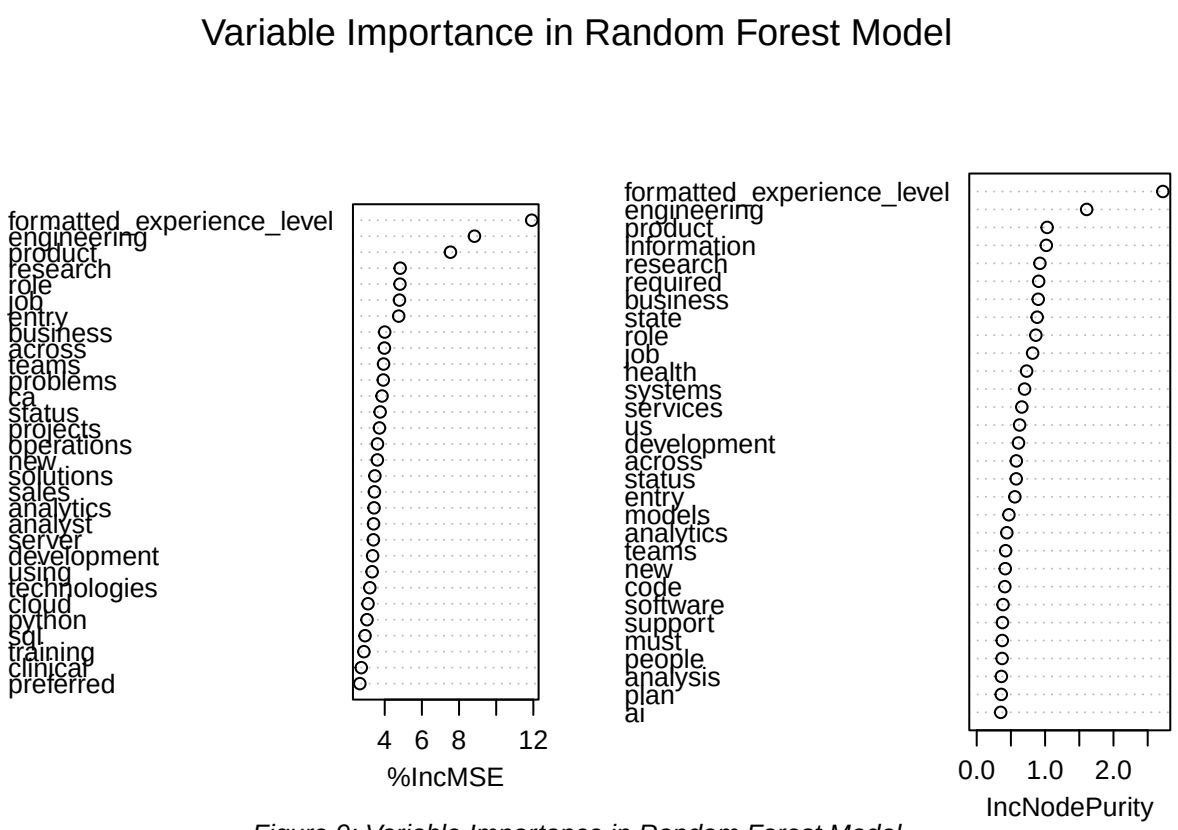

Figure 9: Variable Importance in Random Forest Model

Top variables influencing the Random Forest model included: formatted_experience_level, engineering, product, research, state, health, business, information, job and services. Similar to the Bagging model, experience level and engineering remain crucial, with the addition of location (state) and sectors (health, business) reflecting regional and industry-specific salary standards.

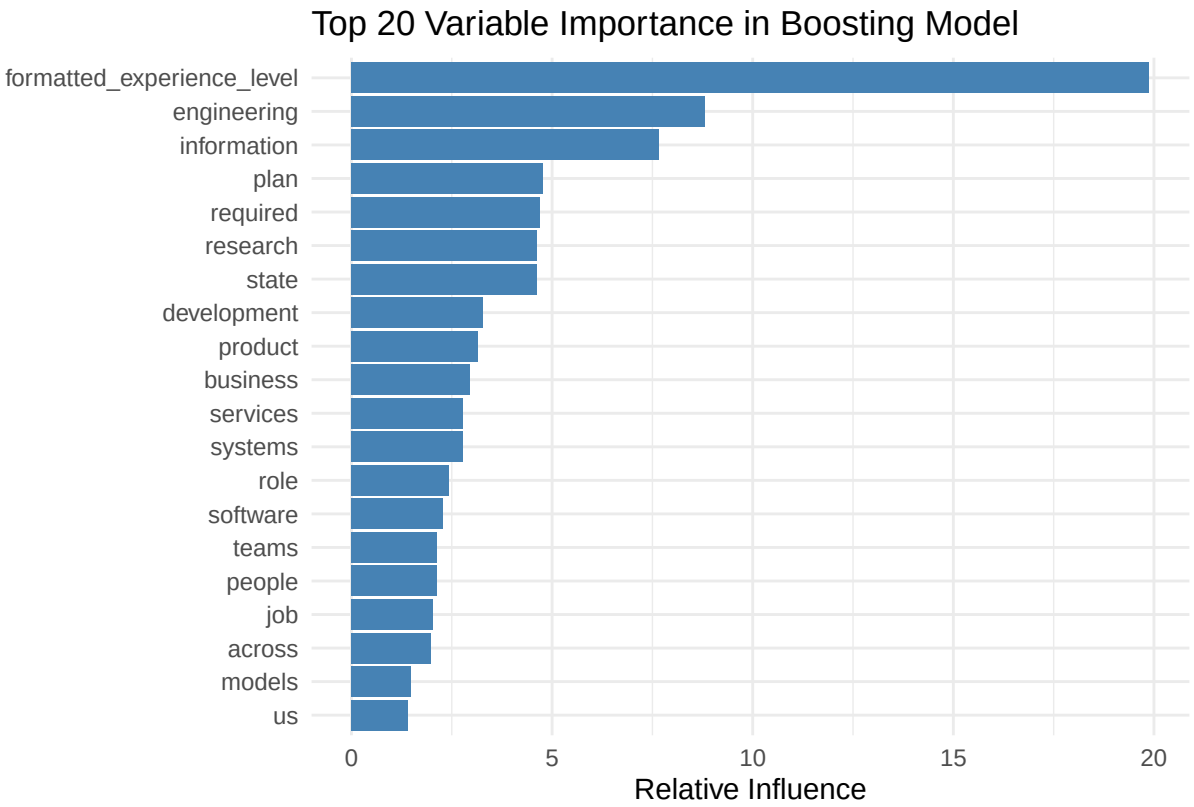

Figure 10: Variable Importance in Boosting Model

For the Boosting model, the variables with the highest relative influence were: formatted_experience_level, engineering, information, plan, research, required, state, development, systems, and software. This model emphasizes the importance of technical roles and planning capabilities, suggesting that strategic positions and comprehensive technical knowledge command higher salaries.

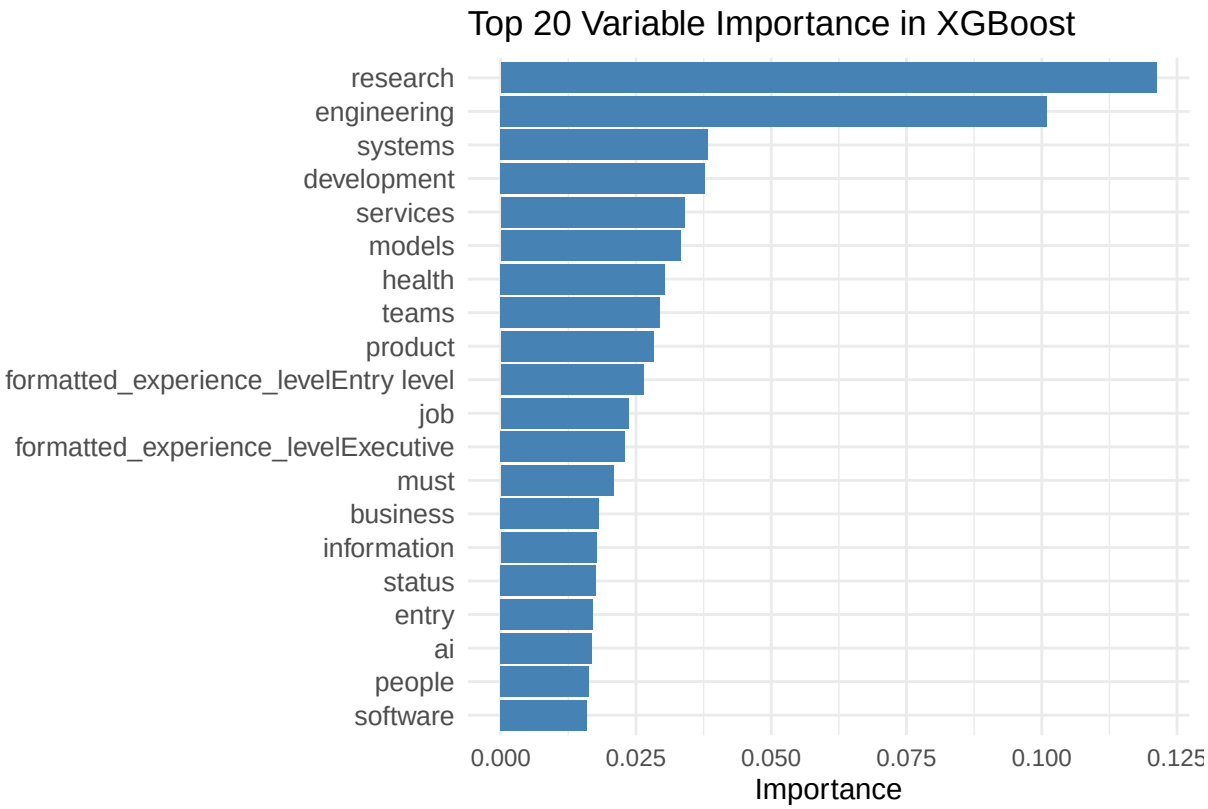

Figure 11: Variable Importance in XGBoost

The XGBoost model highlighted the following as significant: engineering, , research, services, formatted_experience_level, systems, center, information, development, health, and business. Again, technical expertise and sector-specific roles play critical roles, with engineering and research leading the list. Notably, the model distinguishes between different experience levels, directly correlating them with salary variations.

4. Conclusions and Summary

This project aimed to discover which aspects of data science job postings correlate with higher salaries. We evaluated various predictive models to determine which method and characteristics most significantly predict high-salary positions. The models ranged from simple Linear Regression to more sophisticated ensemble methods such as Random Forest and Boosting algorithms. Among all the models, Random Forest achieved lowest MSE and highest R squared value, followed by Bagging, Boosting, XGBoost models, and Regression Tree model. All emsemble models outperform the basline Linear Regression model, achieving more than 95% less MSE than the Linear Regression model.

Across all models, the recurring themes of formatted_experience_level, engineering, and sector-specific knowledge (health, business, information) as pivotal determinants of salary are evident. Some models also leverage emphasis on research experiences and software development skills. These insights align with the broader industry trends where technical proficiency, experience, and industry context are highly valued.

For organizations and professionals in the data science field, focusing on developing skills in these key areas may yield substantial career advancement and compensation benefits. Additionally, the inclusion of state in significant variables suggests regional salary disparities, which organizations must consider when designing compensation packages to attract and retain talent.

Considering the limited data resource and tuning time, the models (especially XGBoost) have the potential to achieve better predicting accuracy, and discover more and closer insight to the target question.

5. Reference

[1] Kon, A. (2023, November 5). LinkedIn job postings - 2023. Kaggle. https://www.kaggle.com/datasets/arshkon/linkedin-job-postings/data