Data Science – Job Market Analysis

Emilia Hoang

Contents

1. Introduction

The project aims to analyze a dataset encompassing company and job information to extract valuable insights. The objectives include understanding salary and rating trends by examining the relationship between factors like company size, industry, and job type with salary estimates.

2. Data description

The dataset is a CSV. File scrapped from the Glassdoor website of Data Science job listings: https://www.kaggle.com/datasets/vincenttu/glassdoor-joblisting?select=joblisting.csv

Dataset has 2573 records and size of 4.4 MB under csv format.

'company'	A string with or without a company rating
'job title'	A string of the job title
'headquarters'	A string of the company's headquarters location
'salary estimate'	A string of the provided estimated salary or salary range
'job type'	A string for the job type (full-time, part-time, internship, etc)
'size'	A string estimating the number of employees at a job
'founded'	An int specifying the year of company founding
'type'	A string for company type (public, private, etc)
'industry'	A string for their industry
'sector'	A string for the sector
'revenue'	A string for the revenue the company generated
'job description'	A string description of the job and/or the company

NOTE: null values are represented as a -1

3. Data processing

R script: 1.handle_rawdata.R

Challenge: the dataset is uncleaned and we will have to deal with texts that belong to a varying number of categories. It is quite noisy and contains a rather high number of missing and unknown values which are represented as a -1.

The purpose of this data handling is to clean and preprocess a dataset for further analysis. The dataset, named "joblisting.csv," is read into R and several cleaning and transformation steps are applied to it. The goal of these operations is to ensure the dataset's quality, consistency, and readiness for analysis.

Loading the dataset:	the process begins with setting the working directory and reading the raw dataset "joblisting.csv" into R using the read.csv() function.
Removing unnecessary column:	the 'index' column is removed from the dataset using the $ operator and the NULL assignment.
Replacing -1 Values:	a loop is used to iterate through each column in the dataset. For each column, all occurrences of -1 values are replaced with NA (missing values). This ensures consistency and accuracy in the dataset.
Handling company data:	'Company Name' column is split at newline characters to separate the company name and rating. The company name is stored in a new 'company.name' column, and the rating is stored in a 'rating' column. The original 'company' column is then dropped. Additionally, a swap is performed on misunderstanding data from 'company.name' to 'job.title' and 'headquarters' to maintain data integrity.
Handling headquarters data:	'Headquarters' column is processed to extract the city and state information. 'HQ_city' and 'HQ_state' columns are created to store this information. The 'Headquarters' column is then dropped.
Calculating company age:	the current year is calculated, and 'company.age' column is created by subtracting the 'Founded' year from the current year.
Handling job type and company type data:	'Job Type' and 'Company Type' columns are cleaned by removing unnecessary prefixes. 'Job Type' values of "N/A" are replaced with NA.
Handling size data:	'Size' column is cleaned by removing 'employees' and replacing ' to ' with a hyphen. 'unknown' and '1-50' values are replaced with NA.
Handling revenue data:	'Revenue' column is cleaned by replacing 'Unknown / Non-Applicable' with NA and removing 'USD', parentheses, and other special characters. 'to' is replaced with a hyphen.
Handling title data:	'Job Title' column is cleaned by removing specified strings and creating a 'Seniority' column based on the presence of 'Senior' in the title.
Handling salary data:	'Salary Estimate' column is cleaned by removing "(Glassdoor est.)" and "Employer Provided Salary:". It is then split into 'min_salary_estimate' and 'max_salary_estimate' columns, which are further cleaned to remove non-digit characters. These values are converted to numeric, multiplied by 1000, and averaged to calculate 'Avg_salary_estimate'. 'Salary Estimate' column is dropped.

Writing cleaned data to csv:	the final cleaned dataset is written to a CSV file named "cleaned_rawdata.csv."

The provided R code performs a series of data preprocessing steps to clean, transform, and prepare the raw dataset for analysis. By following these steps, the dataset's quality and consistency are improved, ensuring that it can be effectively used for further exploration and insights. The resulting cleaned dataset, "cleaned_rawdata.csv" is ready for subsequent analysis tasks.

4. Analytic models

Linear Regression model: use linear regression to predict the salary estimate based on various variables such as company size, industry, and job type. This model assumes a linear relationship between the predictors and the target variable.

Classification model: Given labeled data, use classification algorithms to predict company rating by using these models:

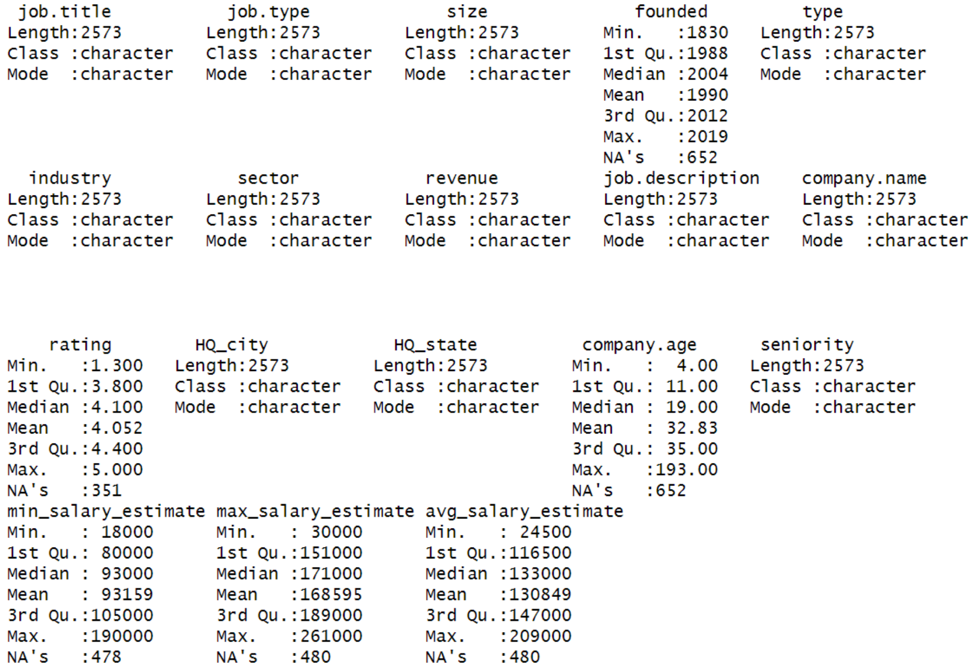
- Logistic Regression model
- Normalized K-Nearest Neighbors (KNN)
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)
- Naive Bayes

5. Insightful findings

4.1 Descriptive analysis

R script: 2.descriptive_analysis.R

Calculate summary statistics (mean, median, standard deviation) for variables like salary estimate, company size, and company age. This will provide a general understanding of the data distribution and central tendencies.



```
   job.title            job.type             size              founded         type          
 Length:2573        Length:2573        Length:2573        Min.   :1830   Length:2573       
 Class :character   Class :character   Class :character   1st Qu.:1988   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Median :2004   Mode  :character  
                                                          Mean   :1990                     
                                                          3rd Qu.:2012                     
                                                          Max.   :2019                     
                                                          NA's   :652                      
   industry            sector             revenue          job.description    company.name      
 Length:2573        Length:2573        Length:2573        Length:2573        Length:2573       
 Class :character   Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  



     rating         HQ_city            HQ_state          company.age       seniority        
 Min.   :1.300   Length:2573        Length:2573        Min.   :  4.00   Length:2573       
 1st Qu.:3.800   Class :character   Class :character   1st Qu.: 11.00   Class :character  
 Median :4.100   Mode  :character   Mode  :character   Median : 19.00   Mode  :character  
 Mean   :4.052                                         Mean   : 32.83                     
 3rd Qu.:4.400                                         3rd Qu.: 35.00                     
 Max.   :5.000                                         Max.   :193.00                     
 NA's   :351                                           NA's   :652                        
 min_salary_estimate max_salary_estimate avg_salary_estimate
 Min.   : 18000      Min.   : 30000      Min.   : 24500     
 1st Qu.: 80000      1st Qu.:151000      1st Qu.:116500     
 Median : 93000      Median :171000      Median :133000     
 Mean   : 93159      Mean   :168595      Mean   :130849     
 3rd Qu.:105000      3rd Qu.:189000      3rd Qu.:147000     
 Max.   :190000      Max.   :261000      Max.   :209000     
 NA's   :478         NA's   :480         NA's   :480        
```

- Company information: the dataset includes data on company names, job types, company sizes, industries, sectors, revenue levels, and company ratings. Companies in the dataset have been founded as early as 1830 and as recently as 2019. The majority of companies were founded around the early 2000s, with the median founding year being 2004.
- Job listings: the dataset contains job titles, descriptions, seniority levels, and salary estimates for the listed jobs. Jobs have varying levels of seniority, indicating opportunities for different experience levels.
- Company locations: the dataset provides information about the city and state where the company headquarters are located.
- Company age: companies' ages vary widely, with an average age of around 32.83 years. the dataset includes some companies that are relatively young (around 4 years old) and some that are quite old (up to 193 years).
- Salary estimates: the dataset provides estimated salary information for the job listings, including minimum, maximum, and average salary estimates. The average estimated salary for the jobs in the dataset is approximately $130,849.
- Missing data: there are missing values in various attributes, such as founded year, company rating, company age, and salary estimates. These missing values should be handled appropriately during analysis.
- Diversity: the dataset covers a wide range of industries, job types, and company sizes, suggesting a diverse set of job opportunities.
- Company ratings: the average company rating is around 4.052, indicating that the companies in the dataset generally have favorable ratings.

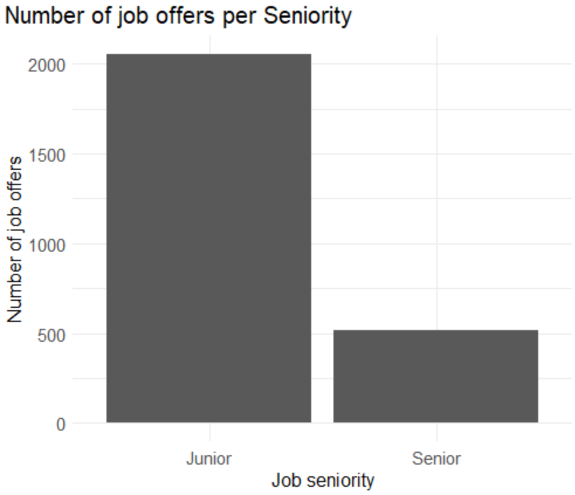
4.2 Exploratory analysis

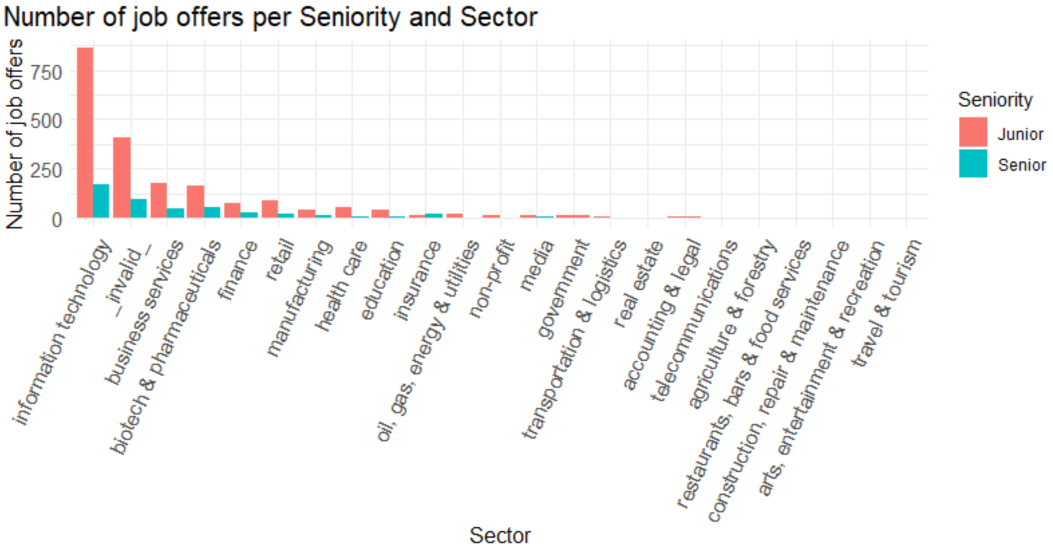
Visualize the relationships between variables using charts and graphs. For example: plot the distribution of job types, compare salaries across different industries or sectors, or examine the relationship between company size and revenue.

R script: 3.EDA.R

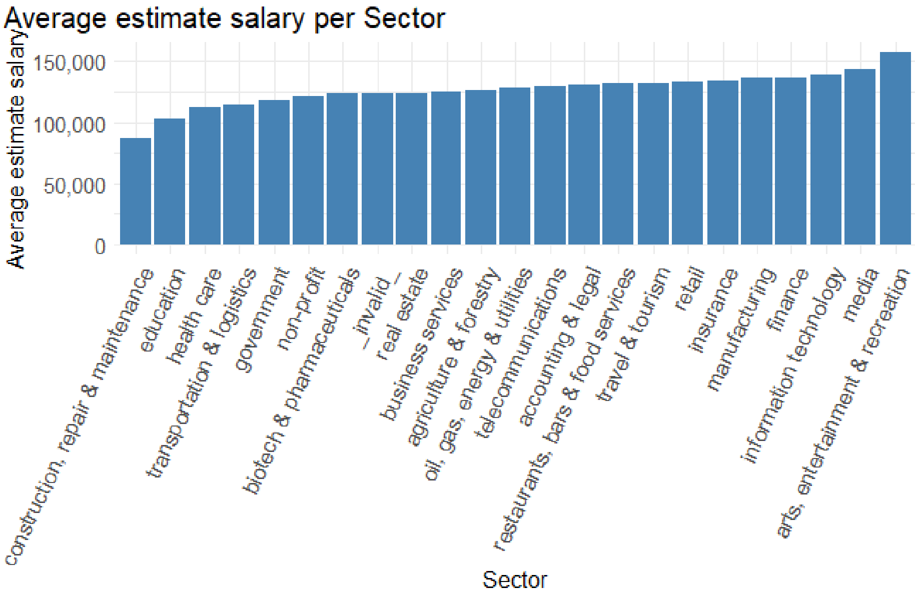
Analyzing the impact of company factors on salary: explore the relationship between company size, industry, revenue, and job type with the salary estimate. This analysis can help understand how different company attributes affect compensation levels.

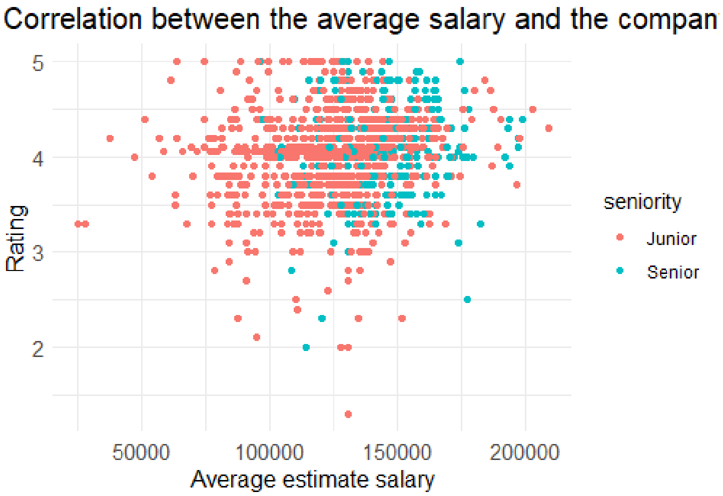
SENIORITY

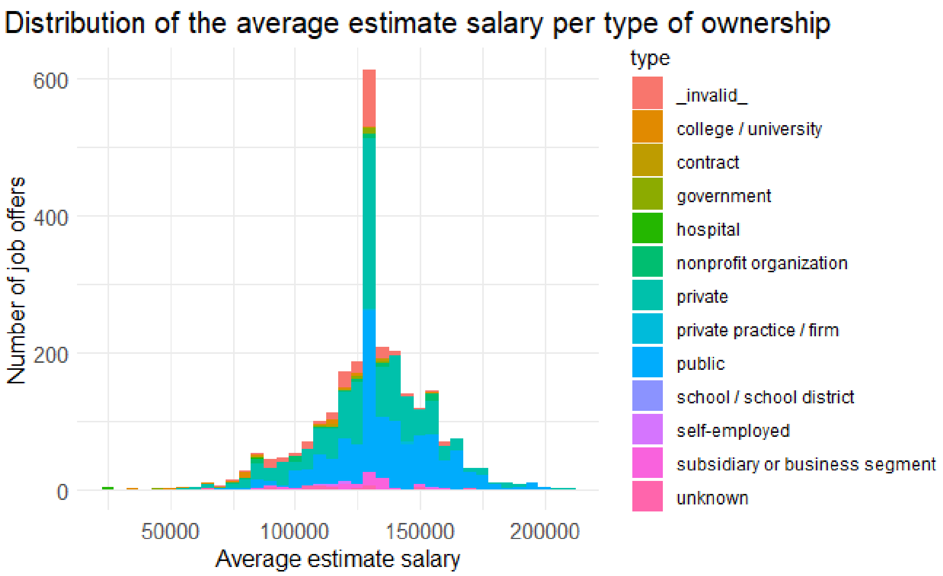


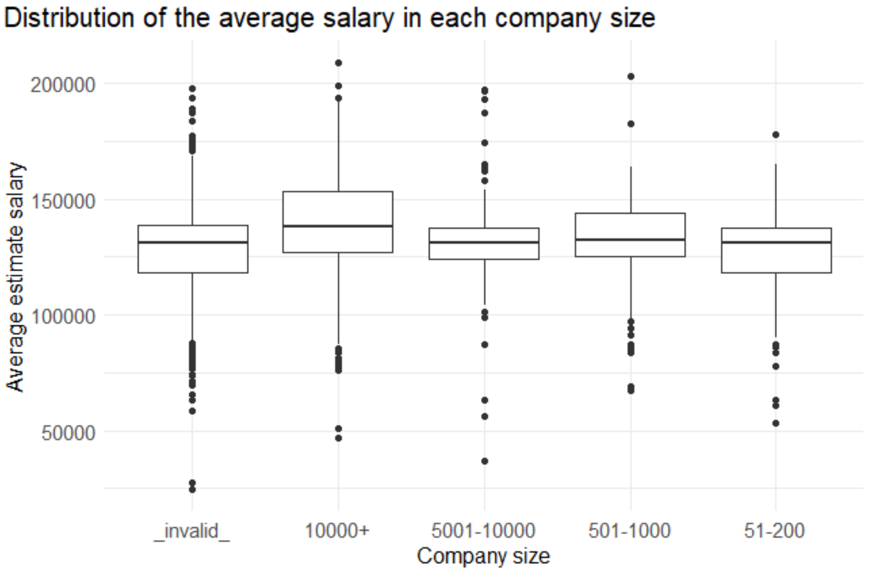


SALARY

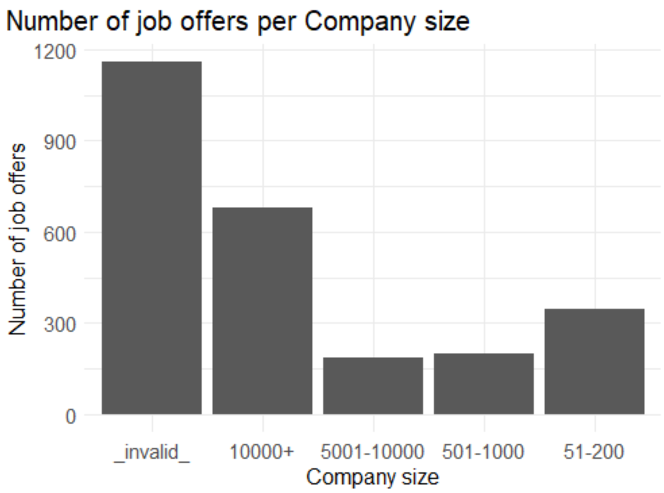

Average estimate salary per Sector
Average estimate salary
150,000
100,000
50,000
0
construction, repair & maintenance
education
health care
transportation & logistics
government
non-profit
biotech & pharmaceuticals
invalid
real estate
business services
agriculture & forestry
oil, gas, energy & utilities
telecommunications
accounting & legal
restaurants, bars & food services
travel & tourism
retail
insurance
manufacturing
finance
information technology
media
arts, entertainment & recreation
Sector


Correlation between the average salary and the compan
Rating
5
4
3
2
50000
100000
150000
200000
Average estimate salary
seniority
● Junior
● Senior

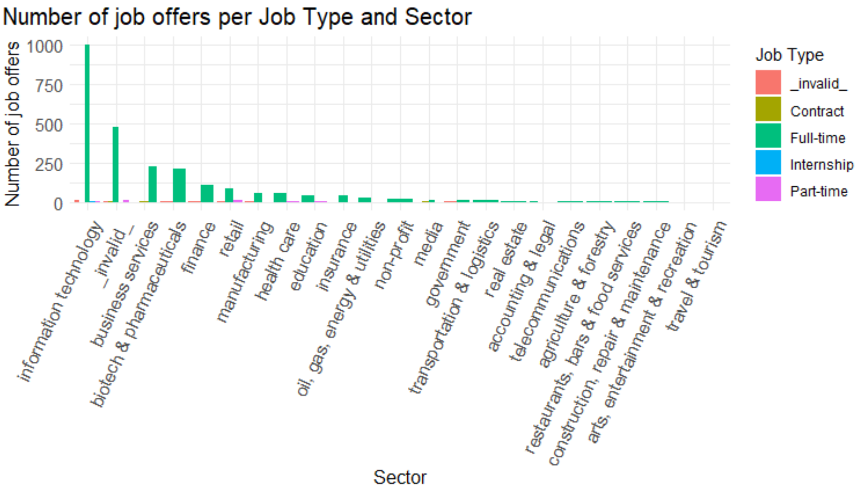




COMPANY SIZE



SECTOR



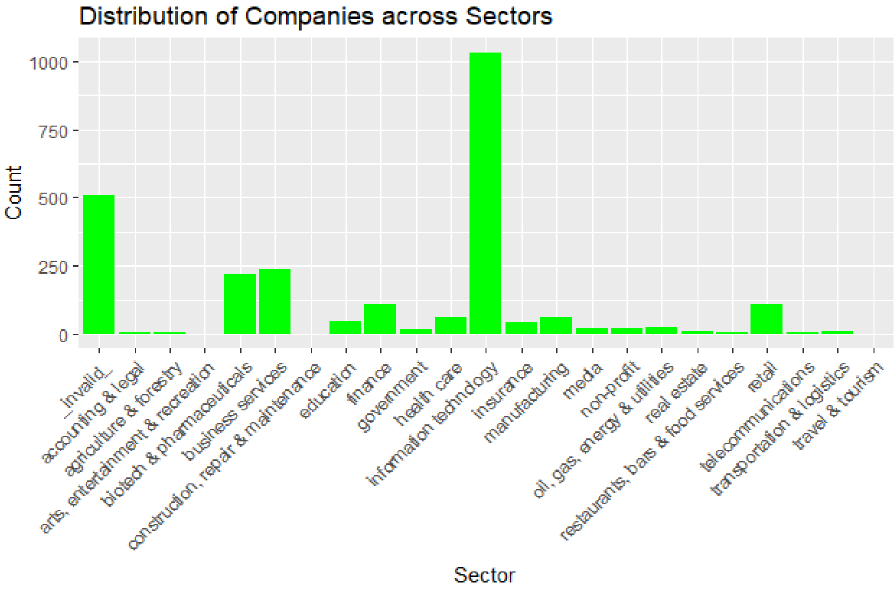
Junior job openings for Data Scientists outnumber senior positions, which is promising for aspiring young professionals in this field. The second graph illustrates the Information Technology sector's strong demand for Data Scientists, which is expected due to its tech-oriented nature. For experienced Data Scientists seeking roles, focusing on Information Technology, Biotech & Pharmaceuticals, and Business Services sectors is recommended, as they offer more opportunities.

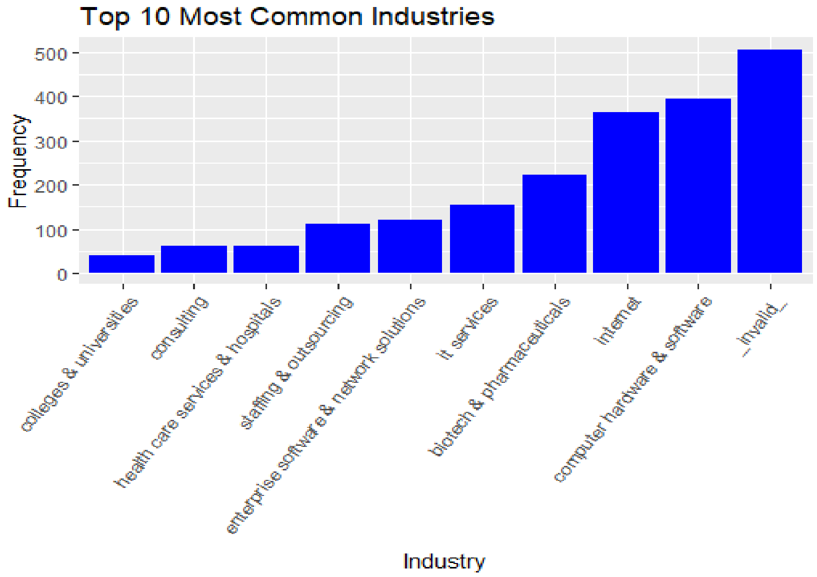
The Arts, Entertainment & Recreation sector boasts the highest average entry salary, while the Construction, Repair & Maintenance sector offers the lowest. Interestingly, there seems to be no significant correlation between a company's rating and the average estimated salary for its Data Scientist roles, as indicated by the scatterplot analysis. The majority of job offers fall within the salary range of $100,000 to $150,000 on average.

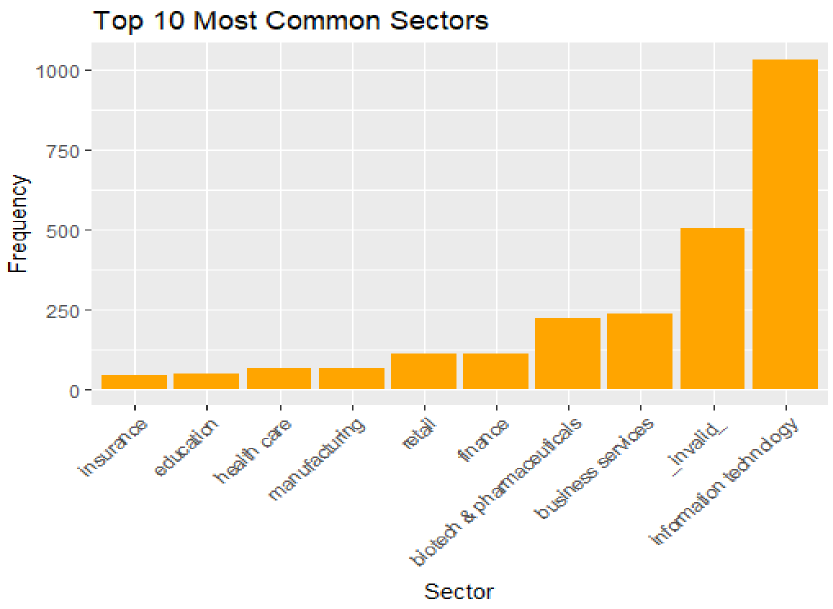
Companies with strong ratings (between 4 and 5 points) that fall under both private and public categories tend to offer Data Scientist positions with average salary estimates ranging from $100,000 to $150,000. Notably, companies with a workforce size exceeding 1000 employees exhibit a significant number of job openings for Data Scientists. The majority of these job offers specify a requirement for full-time commitment from prospective candidates.

<u>R script:</u> 3a.industry_sector_trend_analysis.R

Investigating industry and sector trends: analyze the distribution of companies across different industries and sectors. Identify the most common industries or sectors in the dataset and explore any trends or patterns within them.



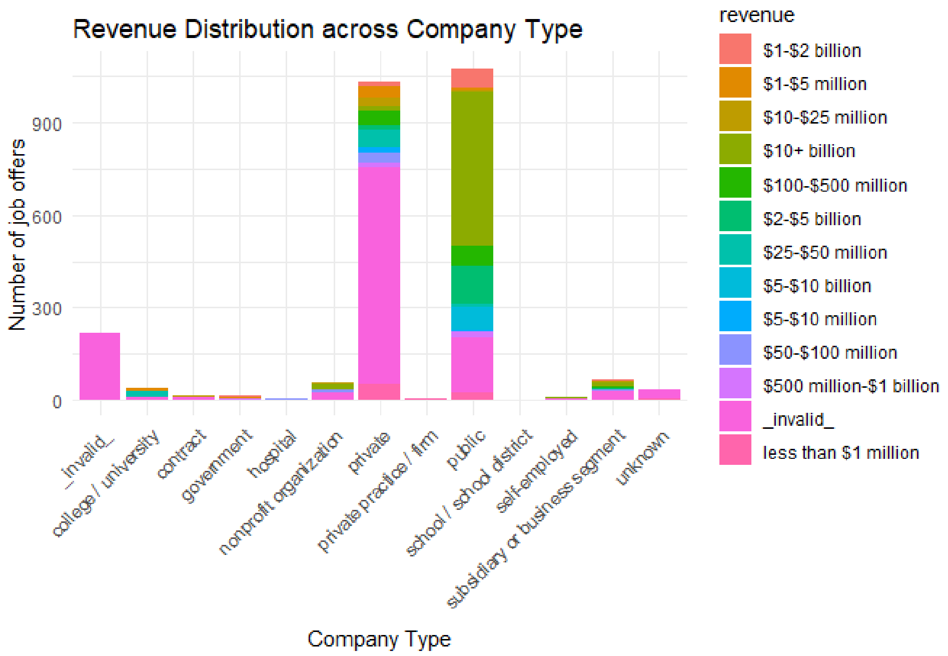


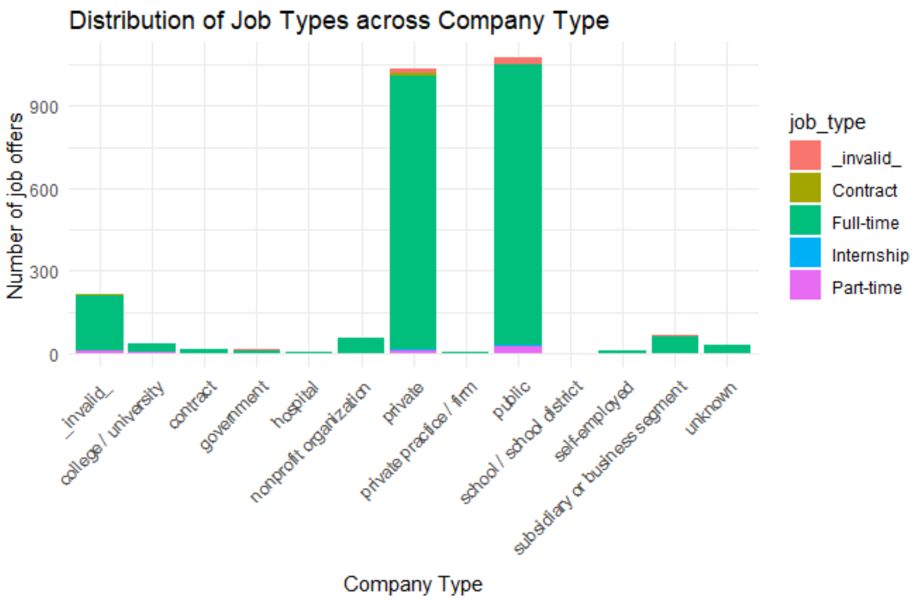


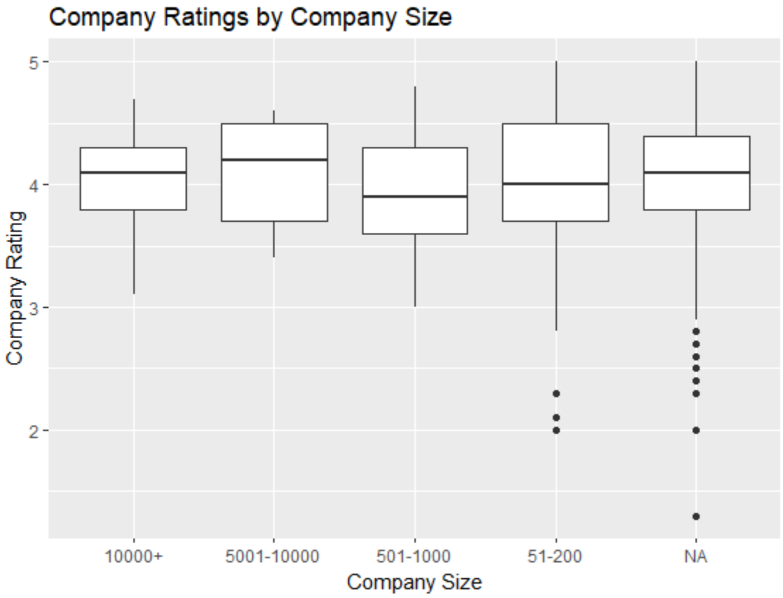
In terms of industries, positions for data science are most abundant in sectors such as computer hardware & software, internet, and biotech & pharmaceuticals. When considering sectors, information technology and business services emerge as the leading sectors with the highest number of available data science positions.

R script: 3b.company_size_analysis.R

Comparing company types: compare the characteristics and attributes of public and private companies in terms of size, revenue, and job types. This analysis can provide insights into the differences between these types of organizations.







The majority of companies seeking data science professionals require candidates to hold full-time positions. Additionally, both private and public company types exhibit the greatest number of data science job opportunities. It's worth noting that companies falling within the revenue range of $5 billion to $10 billion are also prominently offering these positions.

4.3 Predictive analysis

4.3.1 Linear regression - Salary prediction

R script: 4.salary_prediction_linear.R

Predicting salary ranges: build linear regression model to predict the salary estimate based on variables such as job title (seniority), company size, industry, and headquarters location. This can help job seekers or employers gain insights into salary expectations for different positions.

The analysis involves loading, preprocessing, outlier removal, correlation analysis, model creation, prediction, and cross-validation.

1. Loading and preprocessing:

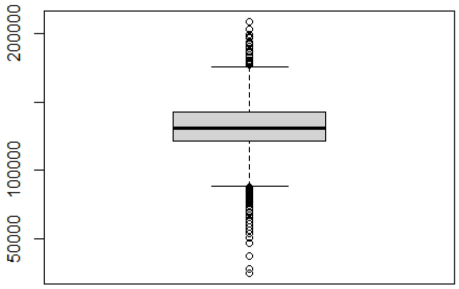
The analysis begins with loading a cleaned dataset named "cleaned_rawdata.csv." The data is loaded using the read.csv() function. To avoid bias, the "Temporary" job type is renamed to "Contract." The 'job.description' column is dropped from the dataset. A summary of the 'avg_salary_estimate' column is displayed.

2. Handling missing data:

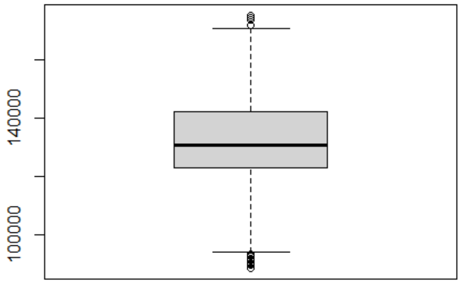
Missing data in the dataset is identified using the count_missing function. The vtreat library is used to design a treatment plan for missing data and prepare the data using this plan. The prepared data contains no missing values.

3. Removing outliers:

Outliers are identified and removed using the Interquartile Range (IQR) method. A boxplot is used to visualize the distribution of 'avg_salary_estimate' before and after outlier removal.



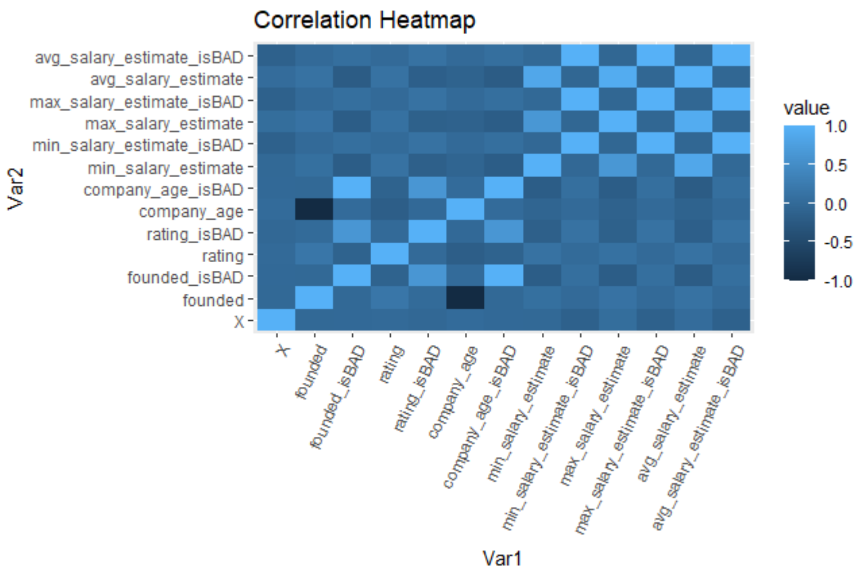
Before moving outliers



After moving outliers

4. Correlation analysis:

The correlation between numeric variables in the dataset is calculated using the cor() function. The relationships between variables are further visualized using a heatmap.



5. Model creation:

The dataset is split into training and testing sets using a random group assignment. A linear regression model is created using the 'avg_salary_estimate' as the response variable and 'size', 'type', 'revenue', 'seniority', 'job_type', and 'company_age' as predictor variables. A formula is constructed using the wrapr library.

6. Model summary and prediction:

The linear regression model is summarized using the summary() function. Predictions are made on both the training and testing datasets using the created model.

7. Evaluation R-squared and Root Mean Squared Error (RMSE):

R-squared and Root Mean Squared Error (RMSE) are calculated to evaluate the model's performance on both the training and testing sets.

R-squared measures the proportion of the variance in the response variable that is explained by the predictors. RMSE quantifies the difference between predicted and actual values.

8. Cross-validation to improve the regression model:

Cross-validation is performed using the caret package. A repeated cross-validation method with 5-fold repetition is used. The train() function is used to create a linear regression model, and mean R-squared and RMSE values from the cross-validation are calculated.

9. Result analysis:

```
> rsq(dtrain$avg_salary_estimate, dtrain$pred_salary)
[1] 0.207557
> rsq(dtest$avg_salary_estimate, dtest$pred_salary)
[1] 0.1936407
```

- R-squared for the training dataset (dtrain): 0.207557

- R-squared for the testing dataset (dtest): 0.1936407

These values indicate that the model explains approximately 20.76% of the variance in the training dataset and 19.36% of the variance in the testing dataset. This means that the predictors in the model collectively account for this proportion of the variability in the average salary estimates.

```
> rmse(dtrain$avg_salary_estimate, dtrain$pred_salary)
[1] 15139.52
> rmse(dtest$avg_salary_estimate, dtest$pred_salary)
[1] 15680.86
```

- RMSE for the training dataset (dtrain): 15139.52
- RMSE for the testing dataset (dtest): 15680.86

These values indicate the average magnitude of the differences between the predicted and actual average salary estimates in the model. A lower RMSE suggests that the model's predictions are closer to the actual average salary estimates.

The model's R-squared values indicate that while there is some level of explanation of variance, there is still a significant amount of unexplained variance in both the training and testing datasets. Additionally, the RMSE values provide information about the average magnitude of errors in the prediction. It's important to consider these metrics in the context of specific analysis goals and the nature of the dataset. Further refinement of the model and feature selection might be considered to improve its performance.

Peform cross validation to improve the model. However, the cross-validated RMSE values are not significantly different from the RMSE values calculated on training and test datasets.

```
mean(model$resample$Rsquared)
[1] 0.1928606
> mean(model$resample$RMSE)
[1] 15501.73
```

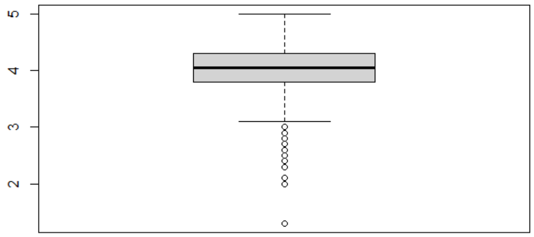
4.3.2 Linear regression - Rating prediction

R script: 5.rating_prediction_linear.R

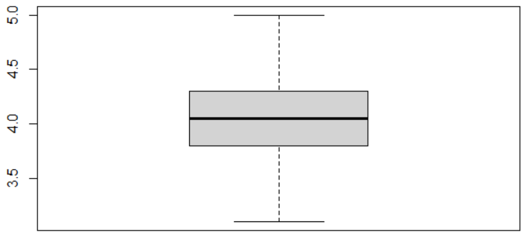
Exploring company ratings: investigate the relationship between company ratings and variables such as company revenue, size, type. This analysis can provide insights into the factors influencing company reputation or employee satisfaction.

1. Data preprocessing: The initial step involves loading the dataset named "cleaned_rawdata.csv" from the specified directory. This dataset is then subjected to data preprocessing to handle missing values. The function count_missing() is utilized to identify columns with missing data, and the Vtreat library is used to treat missing values by replacing them with mean values of their respective columns. The resulting cleaned data is then written to a new CSV file named "data_prepared.csv" for further use.
2. Remove outliers: the cleaned data is reloaded from the newly created CSV file, "data_prepared.csv". A boxplot is generated to visualize the distribution of the "rating" column.

Outliers in the "rating" column are removed using the interquartile range (IQR) method. The boxplot is displayed again to illustrate the absence of outliers in the new dataset, denoted as "new_data".

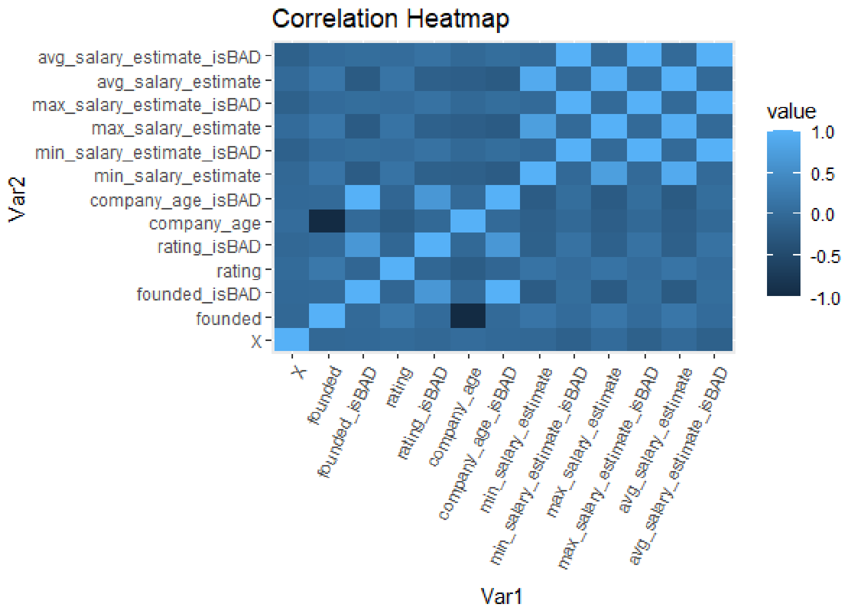


Before moving outliers



After moving outliers

3. Correlation analysis: a correlation matrix is calculated for the numeric variables in the "new_data" dataset. The relationships between these variables are visualized using a heatmap, which provides insights into the degree of correlation between pairs of variables.



4. Build model: perform linear regression model to salary prediction model with ‘rating’ as target variable and ‘seniority’, ‘revenue’, ’size’, ‘type’, ‘founded’, ‘avg_salary_estimate’ dependent variables.
5. Model Prediction and Evaluation: The fitted model is used to make predictions on both the training and testing datasets. Predicted values are added to the datasets as "pred_rating" columns. The model's performance is evaluated using two metrics: R-squared, RMSE
6. Examine the result:

```
> rsq(dtrain$rating, dtrain$pred_rating)
[1] 0.2136301
```

```
> rsq(dtest$rating, dtest$pred_rating)
[1] 0.1858178
```

- R-squared for the training dataset (dtrain): 0.207557
- R-squared for the testing dataset (dtest): 0.1936407

These calculated R-squared values signify the proportion of variance that the model is able to explain within the respective datasets. Specifically, the model accounts for approximately 20.76% of the variance in the training dataset and 19.36% of the variance in the testing dataset. These values reflect the extent to which the predictor variables in the model collectively capture the variability present in the rating.

Additionally:

```
> rmse(dtrain$rating, dtrain$pred_rating)
[1] 0.3614939
> rmse(dtest$rating, dtest$pred_rating)
[1] 0.3524048
```

- RMSE for the training dataset (dtrain): 15139.52
- RMSE for the testing dataset (dtest): 15680.86

Lower RMSE values indicate that the model's predictions are closer to the actual values, with the training dataset having an RMSE of 15139.52 and the testing dataset an RMSE of 15680.86.

The slightly higher RMSE in the testing dataset could suggest that the model's performance on unseen data is slightly less accurate compared to the training data.

These metrics offer valuable insights into the model's strengths and limitations. To enhance its predictive capabilities, we might consider refining the model through exploring alternative algorithms.

4.3.3 Logistic regression & other classification methods

R script: 6.rating_prediction_categorical.R

The goal is to build and evaluate predictive models that can accurately classify companies into high and low rating categories based on selected features.

1. Data Loading and Preprocessing:

The analysis begins with loading the dataset from a CSV file named "cleaned_rawdata.csv" located in the specified directory. The selected columns for rating analysis include 'rating', 'seniority', 'revenue', 'size', 'type', 'founded', and 'avg_salary_estimate'. These columns form a new dataframe called new_data.

Missing data in both numerical and categorical columns are handled simultaneously using the vtreat package. This package helps prepare the data for modeling by treating missing values and rare occurrences. The resulting treated dataset is stored as data_treated.

2. Target Variable Transformation:

The 'rating' column is transformed into a binary target variable with the threshold set at 4.0. Companies with ratings greater than or equal to 4.0 are labeled as 'high_rating', while those below are labeled as 'low_rating'.

3. Data Scaling and Splitting:

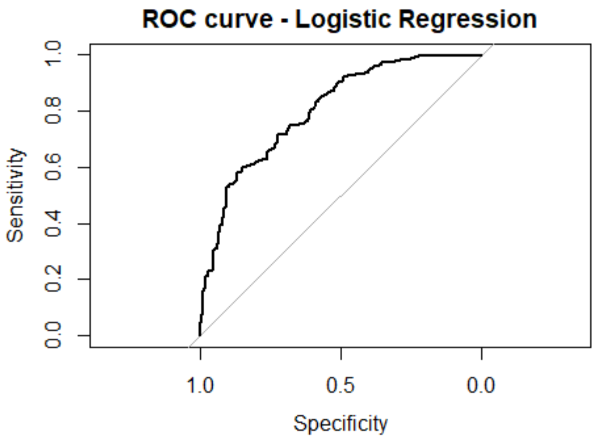
The data is scaled using the preProcess function from the caret package. The scaled dataset is named norm_cleaned_data.

The dataset is then split into a training set (70% of the data) and a test set (30% of the data) using stratified sampling based on the binary target variable. The training set is further converted to a dataframe, and the target variable is converted to a factor.

4. Model Building and Evaluation:

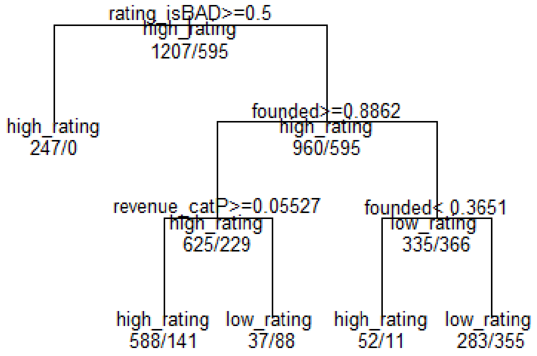
Various classification algorithms are implemented and evaluated using the training and test sets. The algorithms explored are as follows:

- Logistic Regression: A logistic regression model is built using the glm function with a binomial family. Predictions are made using a probability threshold of 0.5. The confusion matrix and accuracy for both training and test sets are computed.



The ROC curve is "to the left," it suggests that the model is performing well in terms of its true positive rate (sensitivity) while keeping the false positive rate (1 - specificity) relatively low. In other words, the model is effectively distinguishing between the positive and negative classes without producing a large number of false positives.

- Normalized K-Nearest Neighbors (KNN): A K-Nearest Neighbors model with k=5 is implemented using the knn function from the class package. Confusion matrices and accuracies are calculated for both training and test sets.
- Decision Tree: A decision tree classifier is created using the rpart package. Two versions of the model are considered: one without complexity control and one with a complexity parameter of 0.05. Confusion matrices and accuracies are computed for both versions.



- Random Forest: A random forest model is built using the randomForest package. Predictions are made based on the response variable probabilities. Confusion matrices and accuracies are calculated.
- Support Vector Machine (SVM): SVM models are implemented using the svm function from the e1071 package. Both linear and radial kernel SVMs are tested with different regularization parameters (C=0.1 and C=10). Confusion matrices and accuracies are computed.
- Naive Bayes: A Naive Bayes model is constructed using the naiveBayes function from the class package. Confusion matrices and accuracies are calculated.

5. Cross-Validation and Model Selection:

Cross-validation is performed using the train function from the caret package to improve the logistic regression model's performance. A repeated cross-validation scheme (5 repetitions of 5-fold cross-validation) is used, and accuracy is used as the evaluation metric. The results of cross-validation are examined to assess the model's performance and stability across folds.

6. Result analysis:

```
                                method accuracy_train accuracy_test
1                                logit      0.2375139     0.2321660
2                 Normalized KNN (K=5)      0.8623751     0.8274968
3                        Decision Tree      0.8451720     0.8274968
4 Decision Tree with complexity control      0.7380688     0.7431907
5                        Random Forest      0.8984462     0.8780804
6                 SVM (Linear, C=0.1)      0.7208657     0.7250324
7                  SVM (Linear, C=10)      0.7330744     0.7172503
8                  SVM (Radial, C=10)      0.7746948     0.7704280
9                          Naive Bayes      0.4178690     0.4228275
```

From the table provided, it appears that the logistic regression model has significantly lower accuracy compared to other models on both the training and test sets. There are several possible reasons for the low accuracy of the logistic regression model:

- Model assumptions: Logistic regression assumes that the relationship between the independent variables and the log-odds of the target variable is linear. If the true relationship is non-linear, the model may not fit the data well.
- Imbalanced dataset: If the dataset is imbalanced, where one class is much more prevalent than the other, the model may be biased towards predicting the majority class, leading to low accuracy.
- Feature selection: The choice of features (independent variables) in the logistic regression model might not capture the underlying patterns well.

Other methods like KNN, decision trees, random forests, and support vector machines (SVM) are capable of capturing more complex relationships in the data compared to logistic regression.

To improve the performance of the logistic regression model, a cross validation method with 5 folds is employed with better result of accuracy of 0.738:

```
   Accuracy     Kappa   Resample
1 0.7708738 0.4472942 Fold1.Rep1
2 0.7320388 0.3494737 Fold2.Rep1
3 0.7641326 0.4198622 Fold3.Rep1
4 0.7572816 0.3902439 Fold4.Rep1
5 0.6679612 0.1926198 Fold5.Rep1

> mean(model$resample$Accuracy)
[1] 0.7384576
```

4.4 Clustering analysis

R script: 7.clustering_citygroup.R

Apply K-means clustering to identify how different types of job opportunities are distributed across various geographical regions within California State.

1. Initial analysis:

The analysis begins by loading the dataset named "data_prepared.csv". A summary table of the frequency of headquarters' cities is generated using the table() function to gain an initial understanding of the data.

2. Feature selection and data transformation:

Columns relevant to the clustering analysis are selected: "HQ_city," "rating," "company_age," and "avg_salary_estimate." These columns form a new dataframe named new_data.

Using the dplyr package, summary statistics are calculated for each headquarters' city. The mean rating, median company age, and mean average salary are computed, resulting in a summarized dataset named group_data.

3. K-Means clustering:

The features are standardized by applying the scale() function to the selected columns, resulting in the pmatrix. K-Means clustering is performed on the standardized data using the kmeans() function with kbest_p set to 5. Cluster centers and sizes are extracted from the clustering results for further interpretation.

4. Interpreting results:

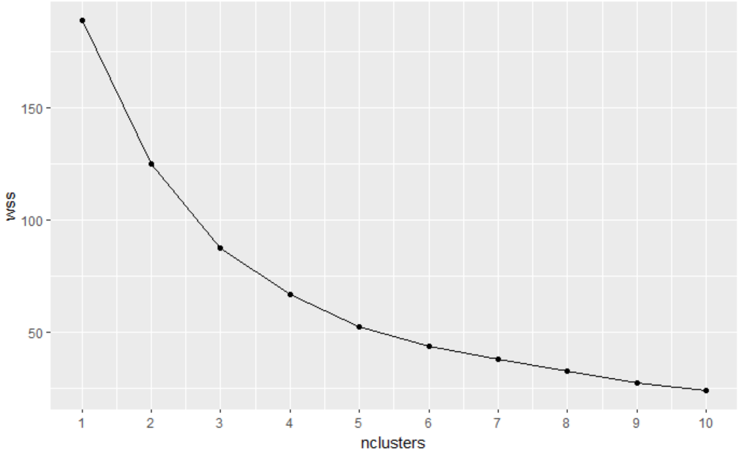
A function print_clusters() is defined to extract and print details of each cluster, including the HQ city, mean rating, median company age, and mean average salary. This function is applied to the clustered data using the computed cluster assignments. The details of each cluster are presented for analysis.

5. Evaluating clustering quality: Within-Cluster Sum of Squares (WSS):

To assess the quality of the K-Means clustering, the WSS metric is calculated. Functions to calculate squared Euclidean distance (sqr_edist()) and WSS for each cluster (wss_each_cluster()) are defined. The total WSS for the entire clustering is calculated using the wss_total() function.

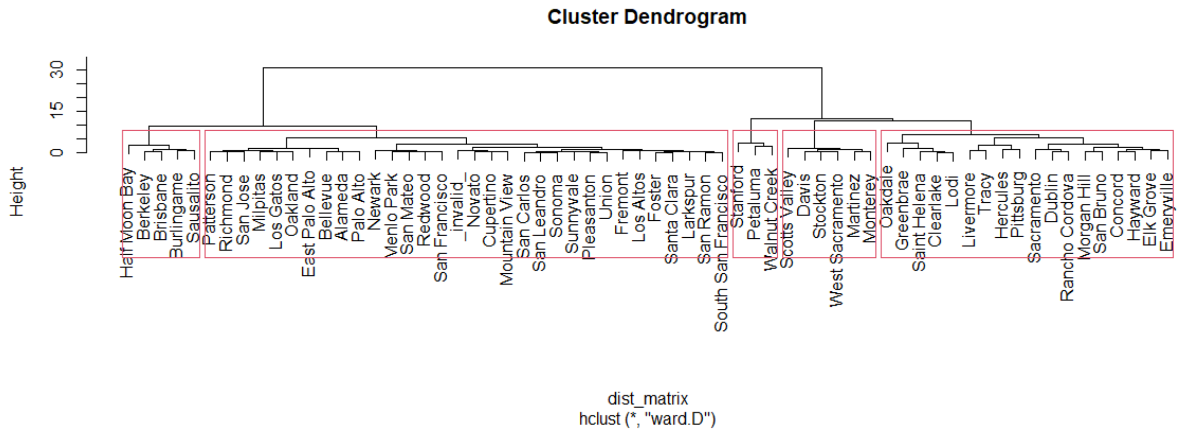
6. Determining optimal number of clusters:

The elbow method is used to determine the optimal number of clusters for K-Means. A function get_wss() is defined to compute the WSS values for different numbers of clusters. The results are plotted using the ggplot2 package, where the x-axis represents the number of clusters, and the y-axis represents the corresponding WSS values. This visualization aids in identifying the "elbow point," which suggests the optimal number of clusters. Result from visualization shows that K = 2 and K = 3 are optimal.



7. Hierarchical clustering:

Hierarchical clustering is performed using the hclust() function with the Ward linkage method. A dendrogram is plotted to visualize the clustering hierarchy, and a specific number of clusters (k=5) is highlighted using the rect.hclust() function.



8. Summary insights:

Data science innovation hub - Cluster label: 5

- Cities in this cluster have high average ratings, relatively new companies, and high average salary estimates.
- These cities likely represent data science and technology innovation hubs within California.
- Aspiring data scientists looking for cutting-edge opportunities and competitive compensation should focus on this cluster.

Diverse data opportunities - Cluster label: 3

- This cluster comprises cities with diverse company ages, moderate ratings, and varying average salary estimates.
- It suggests a mix of data science roles across industries, making it suitable for data scientists interested in various domains.
- Job seekers with a strong desire for versatility and cross-disciplinary experience may find this cluster appealing.

Emerging data markets - Cluster label: 2

- Cities in this cluster have lower average ratings, young companies, and moderate average salary estimates.
- These areas may be emerging data science markets, offering opportunities for those seeking entry-level or growth-oriented positions.
- Data enthusiasts aiming to establish their careers and grow within evolving data ecosystems could explore this cluster.

Data-driven business centers - Cluster label: 4

- This cluster includes cities with older companies and high average salary estimates.
- It likely represents established business centers with strong demand for experienced data scientists.
- Seasoned data professionals looking for stable, well-compensated roles in established industries may gravitate toward this cluster.

Coastal analytics excellence - Cluster label: 1

- Cities in this cluster have high average ratings, relatively new companies, and high average salary estimates.
- These coastal areas may offer exceptional opportunities for data scientists, combining quality of life and high-impact data roles.
- Data practitioners seeking a balanced lifestyle and impactful data roles can consider positions in this cluster.

In summary, the clustering analysis provides insights tailored to the data science job market within different California city clusters. By aligning their career goals with these insights, data science job seekers can make informed decisions about the type of roles and regions that best match their aspirations and preferences.

7. References

Mount, J., & Zumel, N. (2019). Practical data science with R (2nd Edition). Manning.

Burger, S. V. (2018). Introduction to machine learning with R: Rigorous mathematical analysis. O'Reilly Media, Inc.

https://github.com/khaledimad/Predictive-Model-for-Job-Salaries

https://www.ibm.com/docs/vi/spss-statistics/beta?topic=features-exploratory-factor-analysis

https://rkabacoff.github.io/datavis/

https://www.geeksforgeeks.org/data-visualization-in-r/

https://www.geeksforgeeks.org/k-means-clustering-introduction/