Salary Analysis and Prediction of ACME Corporation

Group 11

Daniel Dapula, Ishaan Gupta, Jun Kim, Ranjini Nair, Luke Wu

1. Introduction

Salary is one of many important qualities when workers are on job search. Job prospects are interested in the expected salary range to make informed decisions about whether the pay aligns with their financial expectations and needs. It provides a benchmark for comparing a company's compensation practices with industry standards, allowing individuals to also gauge whether the current salary aligns with the market trends. People would not prefer working at a company where they underpay the workers.

Therefore, our group wanted to explore ACME Corporation's salary structure for workers who hold different job titles and create a predictive model that predicts the salary and understand different variable correlations. People who use this model can get a grasp of the potential earnings that they could get at a company to help them make strategic decisions about their career paths and the companies that they target.

2. Data

Our original, uncleaned dataset contains 375 rows and 6 columns: Age, Gender, Education Level, Job Title, Years of Experience, Salary. The data types per column are diverse, ranging from float64 (64 bits) to object (strings) and int64 (64 bits).

Before diving into the training and testing model, we cleaned the data by handling extraneous values, standardizing the format of the data, and incorporating rationales behind each decision related to data cleaning. First, we created a multiclass encoding for the Education Level column by labeling different degrees with different numerical values. By locating the Education Level column and appending numbers using conditional statements, we created a new column: Education_Bin. Next, we converted the columns — Gender and Education Level — such that each object has its own labeled numerical values in order to use machine learning models such as random forest regressor. Our group chose the Gender_Bin for binary encoding because the column only contained two values: Male and Female. We created conditional statements such that objects are labeled as 1 if the string was equal to ‘Female’ and 0 if otherwise. Finally, we cleaned the raw dataset by removing any missing, unnecessary values. Almost all of the values and elements were relevant except for the missing values, so we simply used the .dropna function to remove any missing (NaN) values from the two-dimensional tabular data structure that we have in Pandas. Our final shape of the cleaned dataset now consists of 373 rows and 8 columns.

3. Methodology

Heatmaps

After cleaning data, we used heatmaps to determine the correlation between all the different relevant numerical variables in our dataset. After parsing the data into numerical values for Gender and Education, we used the .corr() function to find the correlation of all variables. This visualization helped us create appropriate assumptions and gain context of the overall dataset.

As expressed in the code, we dropped categories ‘Gender’, ‘Job Title’, and ‘Education Level’ simply because we assigned numerical values for ‘Gender’ and ‘Education Level’. This created a new data frame with variables of a given numerical hierarchy, which were either straight from the ‘SalaryData.csv’, or hot encoded during cleaning procedures. The newly defined numerical data frame is now only represented by ‘Age’, ‘Years of Experience’, ‘Gender’, ‘Education’, and the observed value of ‘Salary’. We then proceed with the numerical data frame and extract from it a correlation matrix ([numeric_df.corr()]). This method will create a new dataframe ‘corr’, which was created from the objective columns of ‘numeric_df’. After defining these new data frames, we then access the seaborn library to initiate a heat map visualization with the ‘corr’ data frame as its main parameter. In this same line we initialize variables ‘vmin’ and ‘vmax’, putting numerical values to the color concentrations we will eventually formulate. ‘vmax’ is set to a positive 1 color concentration, signifying complete correlation. Complete correlation will only be when variables are being compared to themselves. The contradictory color concentration, noted by ‘vmin=-1’, expresses a complete inverse relationship between the two variables compared. With’ vmin’ and ‘vmax’, seaborn will apply a gradient bar which will be pivotal when analyzing correlation values.

Linear Regression

Our first methodology of comparison is searching linearity through compared variables. We started with our numerical values and predicted the most proportional trends. Our goal in linear regression was to determine the associated correlation between salary and other determining variables.

In order to understand the general correlation for Salary, we included an additional scatter plot from the data gathered. We included a scatter plot of Age vs Salary shown below on the left, and Years of Experience vs Salary shown below on the right:

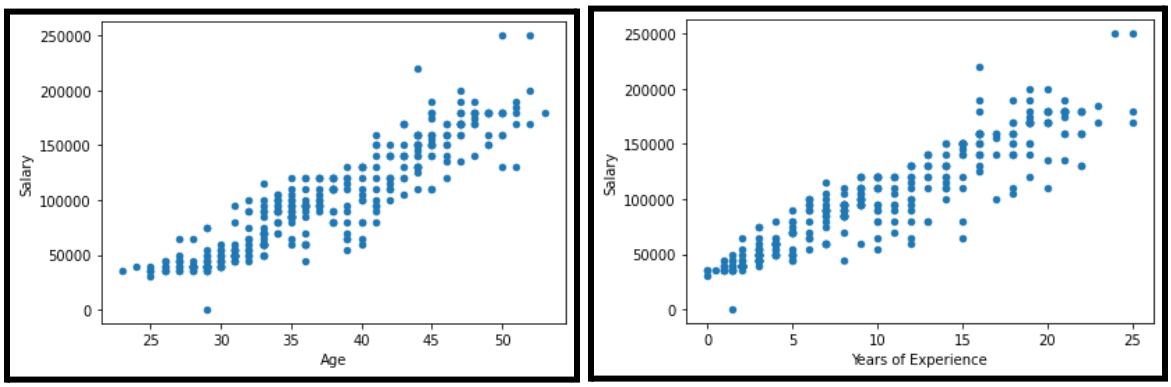

Figure 1. Scatterplot of Age vs Salary and Years of Experience vs Salary

Age and Years of Experience have an already increasing numerical hierarchy to begin with, which are contrary to "bin-like" categorical variables such as Education, Job title, Gender, etc. This made both Age and Years of experience optimal for the scatterplot visuals noted above. If compared to more categorical variables, the plots listed would resemble closer to a bar graph or frequency based data sets. We were also curious about the density distribution of the Salary data. We implemented Seaborn's joint plot function. The function indicated that a large proportion of employees from ACME Corporation that are between the ages of 25-35 have an associated salary of 50,000 to 100,000.

To calculate Linear Regression, we first implemented the method on the entire data set for values under Age and Years of Experience. We now apply linear regression to predict the salary of ACME Corporation employees based on the employee age and years of experience. We used the scikit learn function to split the training and testing data for the linear regression prediction. Our testing data was 20% of the entire dataset, and our training data was 80% of the entire dataset. Using scikit learn's implicit linear regression modeling, we were able to predict the associated salary values.

Random Forest Regression

Our third methodology of data visualization was random forest regression in order to provide the most accurate prediction of salary of ACME Corporation. Aggregating the salary

predictions of multiple decision trees, it reduces the likelihood of high variance and leads to better generalization of the unseen data.

First, we performed encoding to represent the categorical variables (Gender and Education Level) as vectors such that random forest can handle numerical and one-hot encoded categorical features. For gender, we created binary values as there were only two objects (Male and Female) in the dataset. We created multiclass values (0, 1, 2) for education level given that there were more than two unique objects in the column. Along with the encoding, we created two functions — get_field and get_seniority — that determine the field and seniority from the respective columns. Incorporating conditional statements to construct the functions, we applied the functions to the respective columns and created a new data frame with all of the encoded features. This allowed us to convert all necessary categorical variables into a one-hot encoded format. Because we concatenated the new encoded data frame to the original one, we removed the unnecessary columns and missing values (NaN) to finally perform random forest with the clean, appropriate dataset.

	Age	Gender	Education Level	Years of Experience	Salary	Field_Engineering	Field_Executive	Field_Marketing	Field_Operations	Field_Other	Field_Researcher	Field_Sales	Seniority_Director	Seniority_Executive	Seniority_Junior
0	32.0	1.0	0.0	5.0	90000.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	28.0	0.0	1.0	3.0	65000.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2	45.0	1.0	2.0	15.0	150000.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
3	36.0	0.0	0.0	7.0	60000.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
4	52.0	1.0	1.0	20.0	200000.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
368	44.0	0.0	2.0	16.0	160000.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
369	33.0	1.0	0.0	4.0	60000.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
370	35.0	0.0	0.0	8.0	85000.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
371	43.0	1.0	1.0	19.0	170000.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
372	29.0	0.0	0.0	2.0	40000.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0

Figure 2. One-Hot Encoding for Job Title

Before performing random forest analysis, we split the dataset into both training and testing data. We kept 80% of filtered data for training and 20% for testing, and we made sure to pass random_state = rng_seed, one of many arguments in scikit-learn's train_test_split function. It is important to understand how rng_seed works because it dictates the repeatability of the random processes. Setting a seed — in this case rng_seed = 42 — ensures that the random forest algorithm runs with the same dataset and seed, outputting the same results every time.One of the benefits of using random forest is that increasing the number of n-estimators, i.e the number of decision trees, can improve the Out of Bag Score and the Mean Squared Error. The value of 15 n-estimators was chosen as increasing the number of n-estimators had a small increase at the cost of very high runtime. Finally, we used the RandomForestRegressor function from scikit-learn to

find the total number of classifying decision trees on Salary and determine the predictive accuracy. Finally, we found the actual prediction of salary by using .predict of the testing data and compared the performance metric of both predicted and actual salary. Our group also included out of bag (OOB) performance and mean squared error to show the difference between the predicted and actual salary.

4. Results and Analysis

Heatmaps

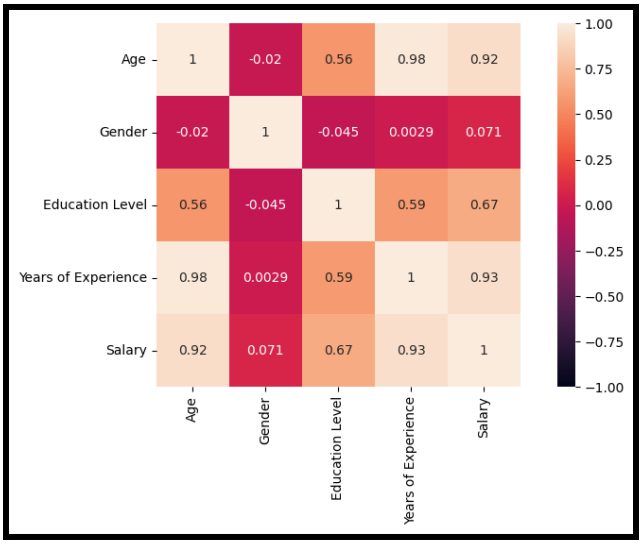

Figure. 3 Heatmap with Correlation Coefficient (Annotated)

As shown by the gradient bar above, the lightest concentrations nearing the diagonal of 1's are what we will assume to have the greatest chance of correlation. For example, it would make sense to assume a more linearly represented model when comparing years of experience and age. In contrast, as expected “binn-ed” values are the least likely to correlate. As highlighted by their darker shading, these values are the farthest from the 1:1 diagonals and are considered to be independent of one another. Therefore, we can assume to be wary of comparing variables that did not have a set numerical hierarchy to begin with.

In all, the heatmap gives correlations between all relevant variables to help understand trends and identify patterns in the data.

Linear Regression

We then computed the least squares estimates of the parameters and determined the R^2 value for the data: 86.9%. In addition to the square of the correlation, we computed the associated variance on the parameters. Shown below is the linear regression line for Age vs Salary and Years of Experience vs Salary.

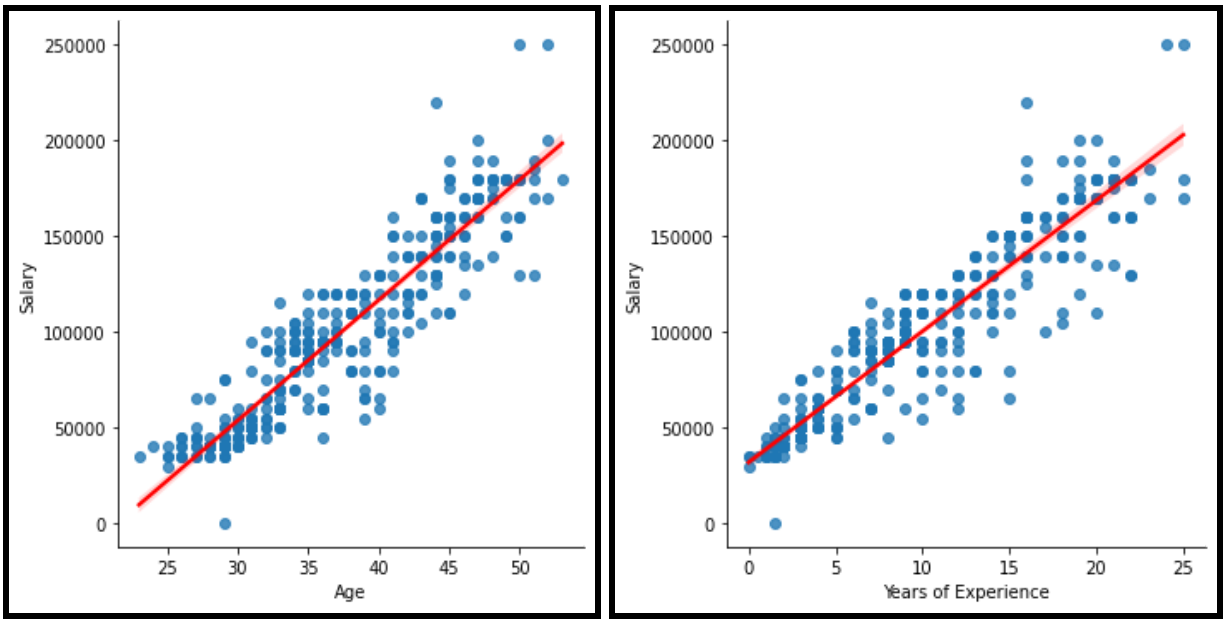

Figure 4. Linear Regression of Scatterplots (Age vs Salary and Years of Experience vs Salary)

We also included the plot illustrating the comparison for the predicted and the testing data from the linear regression model. As the plot is generally on the $x = y$ line, we can visualize the accuracy of our linear regression model.

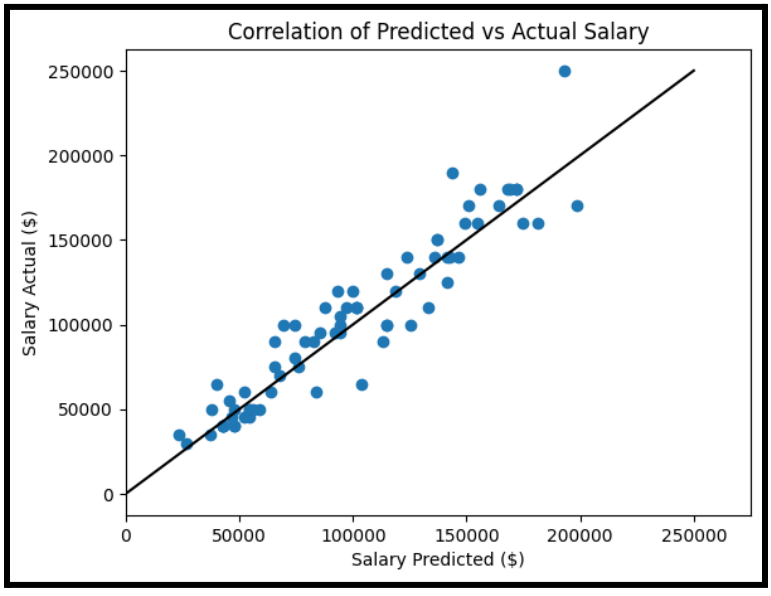

Figure 5. Comparison of Predicted vs Actual Salary ($)

To determine the accuracy of our linear regression model, we found the associated Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

MSE: 206562823 | **MAE:** 10975.473 |**MAPE:** 0.1276

We found the Root Mean Squared Error (RMSE) to be 14372.29, which is larger than the MAE, indicating that there must be large errors in the dataset. Our model additionally indicates we have low, but acceptable accuracy with a MAPE of 12.76%, which is greater than 10%. The model is not as accurate as Random Forest as we will illustrate in the proceeding section.

Random Forest

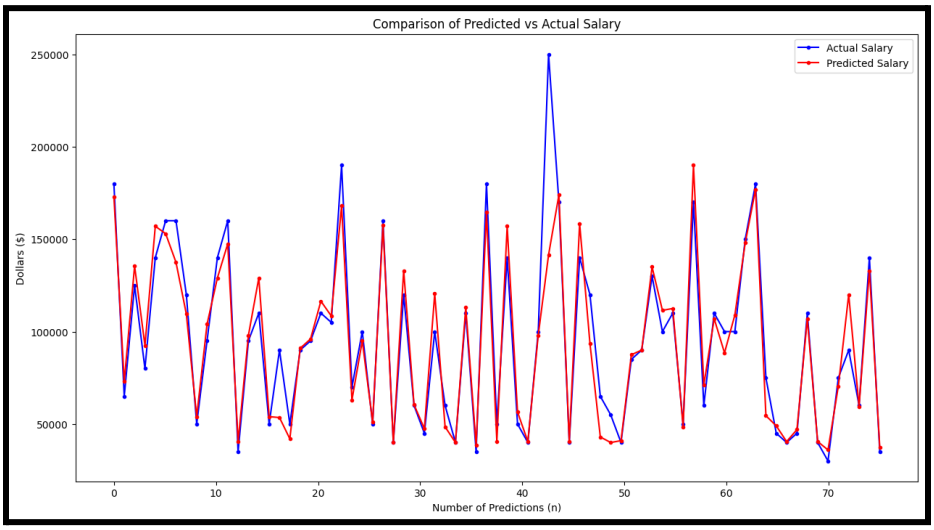

Figure 6. Comparison of Predicted vs Actual Salary ($)

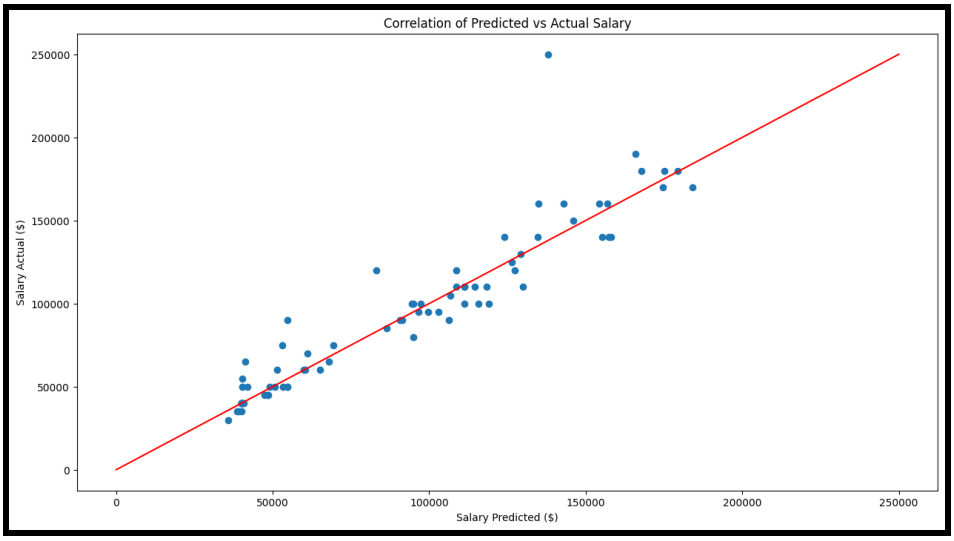

Figure 7. Linear Regression Comparing Predicted vs Actual Salary

Out of Bag Score: 0.906882 | **MAE**: 9226.666666666666 | **MSE**: 295031111.1111111
RMSE: 17176.46969289997 | **MAPE**: 0.09131026739495261

Across the board, the Random Forest Regression is an improvement over the linear regression model, and this improvement is most reflected in its MAPE, which is under 10%. This indicates that we have a relatively accurate model. With the exception of the single outlier in this test data set, the model was able to accurately estimate the salary of an employee at ACME.