Cole Biehl, Jonathan Fong, Xiaoming Huang

Dr. Ghosh

20 March 2024

Machine Learning for Salary Prediction

Introduction:

Salary prediction has advanced with the emergence of machine learning. It provides a scheme for refined, accurate insights into salary trends. This project seeks to apply the KNN algorithm to predict salaries pertaining to data science related roles.

Background:

Salaries are influenced by a variety of factors that include location, company size, industry, and skill set. The KNN algorithm, given its efficacy in adapting multidimensional data and ability to handle non-linear relationships, presents a new opportunity in salary prediction.

The dataset from Kagge, Salary Prediction, includes job listings for data science positions and a variety of features such as job title, salary estimates, company ratings, location, etc. that are pivotal in determining salary. This dataset was derived from listings on Glassdoor.com, with salary estimates cleaned and standardized to aid in analysis. Furthermore, the dataset includes derived features such as average salary, company age, and indicators for key skills.

Design and Implementation:

Our model development process utilized Google Colab as the primary platform, leveraging Python libraries such as Pandas and Numpy for data manipulation, Matplotlib and Seaborn for visualization, and Scikit-Learn for conducting the analysis. We initially loaded two versions of the dataset for examination: “salary data cleaned” and “eda data”. After reviewing the difference between them, we chose to proceed with the “salary data cleaned” since it omitted extraneous columns present in the alternative dataset.

Data Preprocessing

Our next step was data preprocessing, which included handling missing values, data type conversions, and normalization. Missing values were addressed either through removal or imputation. For example, the revenue column had missing values, and we decided to impute the median. Data type conversion was performed to ensure that all features were in the appropriate format for modeling. Normalization techniques were also applied to scale and prevent any feature from dominating the model.

The data preprocess step predominantly included feature removal and feature engineering. Feature removal was applied to eliminate redundant or irrelevant features simplifying the model and potentially enhancing its performance.

1) Salary Estimate: The dataset already contained min_salary, max_salary, and avg_salary as numerical columns, which provided a usable representation of salary information.
2) Job Description: Job descriptions were, in our case, too complex to use. Features such as job title and the others related to having certain data science skills captured essential job information.
3) Rating: Rating had a negligible correlation (0.01) with the target variable (average salary) indicating that company ratings do not significantly impact salary levels.
4) Company Name: With 343 unique values, encoding this feature would have significantly increased the dimensionality and sparsity of the data, which would negatively affect performance. Salaries are more closely associated with the role's responsibilities and the required experience and skills than with the employer's brand.

5) Headquarters: The location of a company's headquarters was found to be irrelevant to salary considerations, which are more directly influenced by the job's geographic location. For example, the fact that Amazon has headquarters in Seattle does not have an effect on the salary of a data scientist working at Amazon in the Bay Area.
6) Size: We consider the size of the company to have a limited effect on the salary since for example a small startup company may have provided larger salaries than older established companies.
7) Founded: The year a company was founded showed a very low correlation (-0.02) with the target variable (average salary) and was therefore considered irrelevant for predicting salary.
8) Industry: Industry was removed due to its redundancy with the sector feature, which provides similar information in a more generalized form.
9) Competitors: 62% of the data marked as -1, indicating missing information, this feature was deemed unreliable and therefore removed.
10) Hourly and Employer Provided: 97% of the values were 0, suggesting that they do not contribute meaningfully to the salary.
11) Company_txt: This feature, which is the company name, includes redundant information since the feature company name exists.
12) Same State: Same state describes whether the applicant lived in the same state as the job location. Understanding salaries are determined by the job market and role requirements rather than the applicant's location, we decided to remove it.
13) Age: The age of the company showed very low correlation (0.02) with the target variable and therefore considered irrelevant for predicting salary.

14) R_yn: All the values of whether R was required skill was 0 and therefore would not affect salary levels.

Feature engineering, an essential aspect of machine learning, involves creating new features or modifying existing ones to improve model performance.

1) Job Title: Diverse job titles across different companies resulted in a lot of unique job title values. Recognizing that many job titles represent similar roles, we aimed to categorize these positions by identifying common keywords and synonyms that signify similar job functions. We used regular expressions to search for these keywords, enabling us to group similar roles into broader categories. Specifically, we created binary features for the terms "data," "scientist," and "analyst," which are indicative of the job's focus area. For instance, a job title containing any synonyms of "scientist," such as "researcher," was flagged as "scientist."To distinguish leadership roles, we added a "Manager" feature. If the job title had the words "senior," "directory," "VP," and several others, it was considered to be a managerial role. After the new features were created, the original job title feature was removed.
2) Location: The dataset contained over 200 unique values for the "location" feature, in the form city, state. This deterred us from using one-hot encoding due to the potential increase in dimensionality and sparsity, which could adversely affect the performance of the KNN model. To address this, we chose to represent location information through geographical coordinates, specifically longitude and latitude. We used the Google Maps API to obtain the coordinates for each location and introduced two new features, "latitude" and "longitude," to the dataset. This approach allowed us to replace the original "location" feature with these more model-friendly numerical features.

3) Sectors: Given the manageable number of unique values for the "sector" feature, standing at 25, and acknowledging the significant impact of the industry on salary levels—where, for example, individuals in the biomedical field tend to earn more than those in agriculture—we opted for one-hot encoding of the sector data.
4) Type of Ownership: The types of ownership were encoded into the top five predefined types: "Company - Private," "Company - Public," "Nonprofit Organization," "Subsidiary or Business Segment," and "Government." A "Other" type was encoded for those records not matching the aforementioned ones. These categories were chosen since a large percentage of the records had one of these types of ownerships. Once the most common type of ownership was chosen, each row was checked to determine if it fell under one of those categories or not.
5) Revenue: Originally, revenue was a string of the range of the company's revenue. We hardcoded the ranges of the revenue into two columns, min and max. For any string "$10+billion (USD)," we decided to make the min and max ten-billion. If any record had "Unknown" or "-1" in its revenue, the max and min were marked as "None."

Data Analysis

We used the KNN model for predicting the average salary. KNN was selected for its proficiency in managing classification problems, which can also be applied to regression tasks such as predicting salary values. The model's underlying principle—that similar data points are likely to have similar outcomes—made it particularly suitable for our dataset, where features like geographical location, sector, and required skills play a significant role in determining salary levels.

First we divided the dataset into training and testing subsets. In our training phase, we experimented with various k values. For each instance in the testing set, the model predicts the average salary by calculating the mean of the average salaries of the kth nearest neighbors. Since these test instances had known actual salaries, we were able to compare our model's predicted salaries against these true values. The error between the predicted and actual salaries is quantified using the Root Mean Square Error (RMSE). By applying KNN across a range of k values, we identify the optimal k that minimizes the RMSE.

Results and Interpretation:

Quantitative Results

The KNN model, with k=5, provided the following Root Mean Squared Error (RMSE) values, which are key indicators of our model's predictive performance:

- Training Set RMSE:
  - Min Salary: 18.78 (thousands of dollars)
  - Max Salary: 26.87 (thousands of dollars)
  - Avg Salary: 22.37 (thousands of dollars)
- Testing Set RMSE:
  - Min Salary: 21.15 (thousands of dollars)
  - Max Salary: 32.58 (thousands of dollars)
  - Avg Salary: 26.37 (thousands of dollars)

These values represent the average prediction error on both the training and testing datasets, highlighting the model's generalization capabilities.

Model Learning Evaluation

The learning of the model was assessed using the RMSE values from the training and testing sets. The slight discrepancy between these errors suggests room for improvement, potentially through feature engineering or alternative modeling techniques. Learning curves and additional evaluation metrics were not presented here but can be included in a more detailed technical report if necessary.

Discussion of Results

The RMSE values reported are significant when considered relative to the unit of measurement (thousands of dollars). The model's performance might be influenced by several factors:

- Outliers in Salary Data: KNN is sensitive to outliers, which can disproportionately affect average predictions and result in higher RMSE values.
- Feature Representation: The predominance of binary features derived from one-hot encoding might dilute the model's sensitivity to subtle differences among data points, leading to a higher error rate.

Conclusions and Recommendations

While the model shows promise, the RMSE values indicate that the accuracy of salary predictions could be improved. For future iterations, we recommend exploring outlier detection and removal, experimenting with more sophisticated models like ensemble methods or neural networks, and potentially incorporating more diverse and continuous features. Further investigation into feature weighting within the KNN algorithm may also yield improvements.

By addressing these areas, we aim to refine our model to provide more precise salary predictions, which could have significant applications in job market analysis and individual career planning.