Prediction of Data Science Job Salaries

Olivia Lee

Boston University MET Computer Science Department

CS 677 Data Science with Python

Final Project Report

Fall 2022

Introduction

The data set I am using contains information on data scientists and similar professions, such as Machine Learning Engineer, Data Analyst, Data Engineer, etc. In this project, I use the attributes in this dataset to predict the salary range of a particular employee.

The Dataset

The dataset has 607 rows and 12 attributes.

Attributes:

- work_year: The year the salary was paid.
- experience_level: The experience level in the job during the year with the following possible values: EN Entry-level / Junior MI Mid-level / Intermediate SE Senior-level / Expert EX Executive-level / Director
- employment_type: The type of employment for the role: PT Part-time / FT Full-time / CT Contract / FL Freelance
- job_title: The role worked during the year.
- salary: The total gross salary amount paid.
- salary_currency: The currency of the salary paid as an ISO 4217 currency code.
- salary_in_usd: The salary in USD (FX rate divided by avg. USD rate for the respective year via fxdata.foorilla.com).
- employee_residence: Employee's primary country of residence in during the work year as an ISO 3166 country code.
- remote_ratio: The overall amount of work done remotely, possible values are as follows: 0 No remote work / (less than 20%) 50 Partially remote 100 Fully remote (more than 80%)
- company_location: The country of the employer's main office or contracting branch as an ISO 3166 country code.
- company_size: The average number of people that worked for the company during the year: S less than 50 employees (small) M 50 to 250 employees (medium) L more than 250 employees (large)

Data Cleaning

1. **Filtering and selecting columns**
   I dropped the redundant index column that was included in the csv file and dropped all rows that contained NAs. In this dataset, there were none.

2. **Identifying and handling outliers**

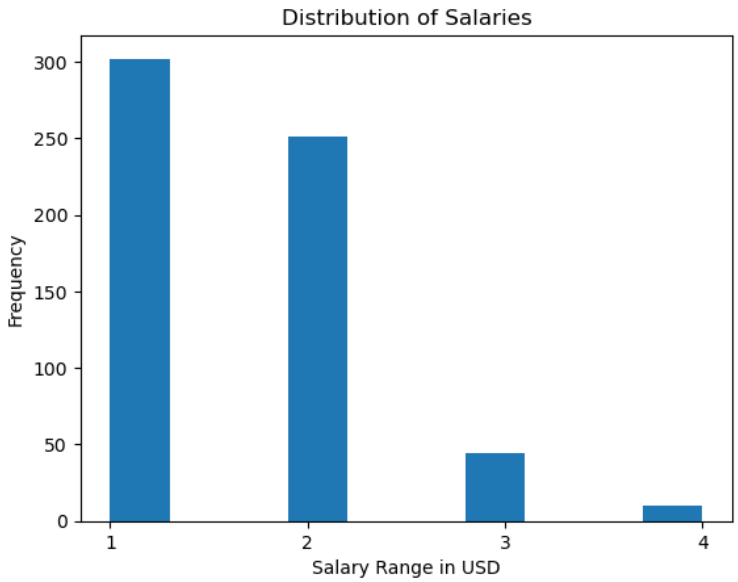
   For the attribute salary_in_usd, although there are many minimum outliers with salary prices that seem unrealistic in the US, I have decided to keep them in the dataset as they are from other countries that could have different wages. All other attributes had no outliers or incorrect entries.

Data Preprocessing

1. **Binning**

   Since salary_in_usd is a continuous variable, I employed binning to convert salary_in_usd to a categorical variable. I created a new column named salary_bin to represent the salary bin of each datapoint. In salary_bin, the values are 1, 2, 3 and 4 which represent the following bins.

   Bin 1: (2859, 100000]
   Bin 2: (100000, 200000]
   Bin 3: (200000, 300000]
   Bin 4: (300000, 600000]



2. **Label Encoding**

   For the ordinal variables in the dataset, I used label encoding to convert the string values into numerical values. They take values from 0 to 1 as the ranking matters in these attributes.

a. experience_level
   EN = 0, MI = 0.25, SE = 0.5, EX = 1

b. company_size
   S = 0, M = 0.5, L = 1

3. **One-hot Encoding**
   a. I created dummy variables for the following categorical variables where ranking does not matter.
      - employment_type
      - salary_currency
      - employee_residence
      - company_location

   b. I created new dummy variables columns for job_title.
      - job_ml: If job_title contains "Machine Learning", "AI", "Computer Vision" or "NLP", we label it 1 in job_ml. Otherwise, 0.
      - job_analyst: If job_title contains "Data Analyst" or "Analytics", we label it 1 in job_analyst. Otherwise, 0.
      - job_data_scientist: If job_title contains "Data Scientist" or "Data Scientist", we label it 1 in job_data_scientist. Otherwise, 0.
      - job_data_engineer: If job_title contains "Data Engineer" or "Architect", we label it 1 in job_engineer. Otherwise, 0.

4. **Final Dataset**
   We drop the columns salary, salary_in_usd and job_title as we will not need them in prediction. The final dataset is then the encoded dataset after dropping these columns.

5. **Splitting Test and Training Sets**
   I split the dataset into X and y, where X contains the data of the predictors and y is the predicted class, salary_bins. I split the dataset into test and training sets of equal size.

Machine Learning Algorithms

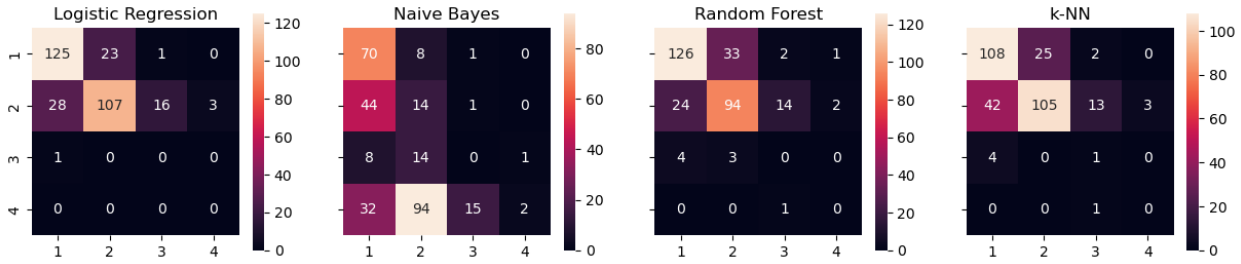
1. Logistic Regression
2. Gaussian Naive Bayes
3. Random Forest
4. K-Nearest Neighbors

Dimensionality Reduction

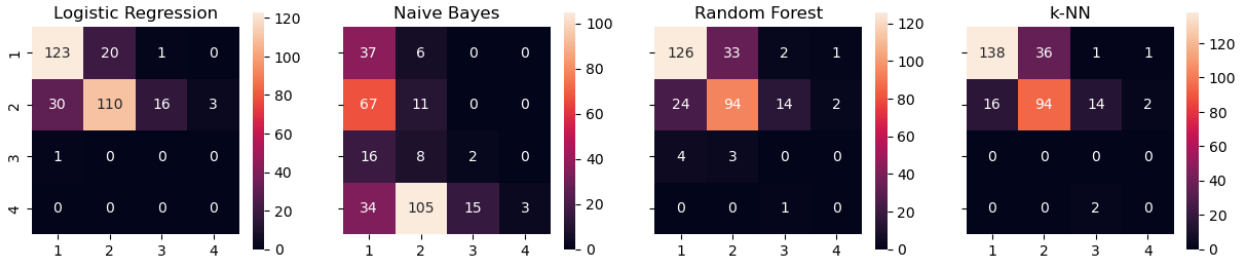
1. Chi-square test dimensionality reduction method was used to reduce the number of columns in the test and training sets. I removed any attribute that had a p-value of less than 0.1.

Model Evaluation

1. Without dimensionality reduction



2. With dimensionality reduction



3. Performance

	Model	Accuracy without Dimensionality Reduction	Accuracy with Dimensionality Reduction
0	Logistic Regression	0.763158	0.766447
1	Gaussian Naive Bayes	0.282895	0.174342
2	Random Forest	0.723684	0.756579
3	K-Nearest Neighbours	0.703947	0.763158

Conclusion

The best model for this dataset is Logistic Regression with Chi-square test dimensionality reduction method with an accuracy of 0.766447. The Chi-square test dimensionality reduction

method improved accuracy but not by a significant amount. The worst performing model is Gaussian Naive Bayes overall, with an accuracy of 0.282895 without dimensionality reduction and 0.174342 with dimensionality reduction.