# Predicting Data Scientist Salaries based on Levels.fyi Data

*Patrick* Brown, *Aksharan* Saravanan, *Vivek* Saravanan, and *Axel* Sagundo

## 1 Dataset & Exploratory Analysis

The dataset we have chosen is Data Science and STEM salaries from Kaggle [1]. This dataset scraped data from Levels.fyi, a job salary and comparison website. We explored many options, but this dataset had the largest combination of features and was easy to work with. Among the 29 features, the ones we used are timestamp, company, title, digital marketing id (identification for regions for marketing purposes), city id, years of experience, tenure, degree achieved (BS, MS, Ph.D., some college, high school), tag (short description of role), city/state/country, position level, and of course salary.

### 1.1 Dataset Visualization

The dataset contained a total of 62,300 data points, however, there were 59,996 data points once we filtered out entries with no reported salaries. The first property we examined was the distribution of companies. As seen in Figure 1, a few companies had more data points than the rest. The largest of the companies can be seen in Figure 2.

Next, we looked at the distribution of salaries. As always, there are always outliers and it's difficult to know how accurate the user-inputted fields are, but as we can see in Figure 3, the distribution was roughly Gaussian centered around 0.15e6, or 150,000.

Since there was a timestamp included, we also decided to graph the salaries over time, using a sliding window of 500 entries. As seen in Figure 4, the first reported date was 06/07/2017 and the last was 07/17/2021. The sliding window calculated the average, starting at around 155,000 and then falling down to around 130,000.

We also know that salary is directly proportional to experience tenure, so we plotted salary vs experience as can be seen in Figure 5. In the calculation, we used rounded salaries and removed outliers, showing a general upward trend.

For the degrees, there were 12505 entries with a BS degree, 15298 with an MS degree, 12505 with a PhD degree, 318 with a college degree, 1793 with a high school degree, and 17577 that had no specification. Of course, there could and would be many overlaps with degrees (with people earning both a BS and MS, for example), while many people don't even enter their degrees and just report their salary and company.

Lastly, we plotted the average salary by salary title and the most popular cities. As seen in Figure 6, the popular cities are likely to correlate with the popular companies mentioned above, and seem to center around technology hubs like the Greater Seattle Area and the San Francisco Bay Area.

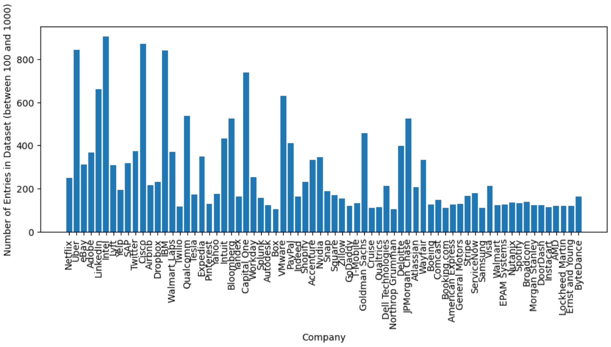

**Figure 1.** Number of datapoints per company. We can see that there is a distribution, but all have a significant representation.

## 2 Predictive Task

As seen in the exploration section, this data has a lot of features that can have some significance in predicting the salary of a given job position. The ones we chose to explore were time, years of experience, highest levels of education, company, job position, level, and primary location of the job. We took each of these into consideration and created separate feature vectors that are described below.

### 2.1 Feature Identification

When graphing the salary over time for the dataset, we noticed there was a definite downward trend as well as a periodic shape of the average salary using a sliding window. Those aspects led us to first consider a simple temporal model using the sliding window technique. For this, we maintained a sliding window of 1000 entries and computed the average of those to predict the next entry.

We first combined the timestamp and salary into a tuple, then sorted the resulting list by increasing time. Then, for the first 1000 entries, we simply set the salary prediction to be the average of those 1000 entries, to act as a starting point. Then for each next entry, the predicted salary was the average of the previous 1000 entries, storing the prediction in a list. When implementing this approach,

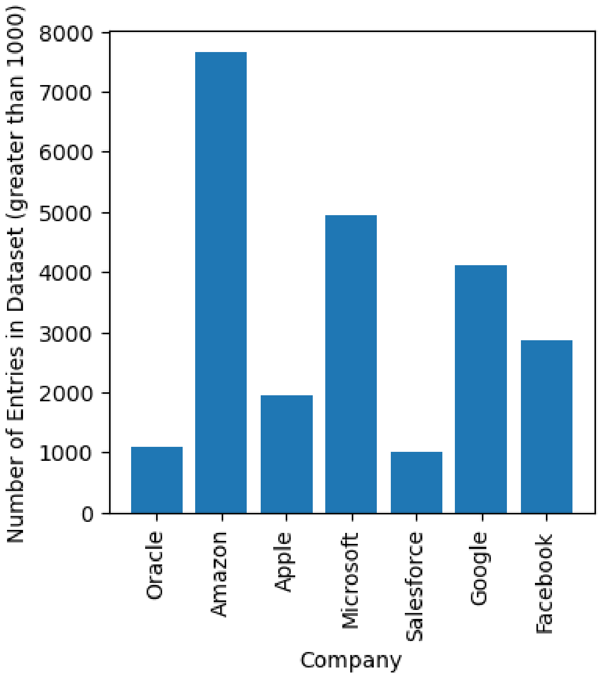

**Figure 2.** Number of datapoints per company for companies with more than 1,000 datapoints.

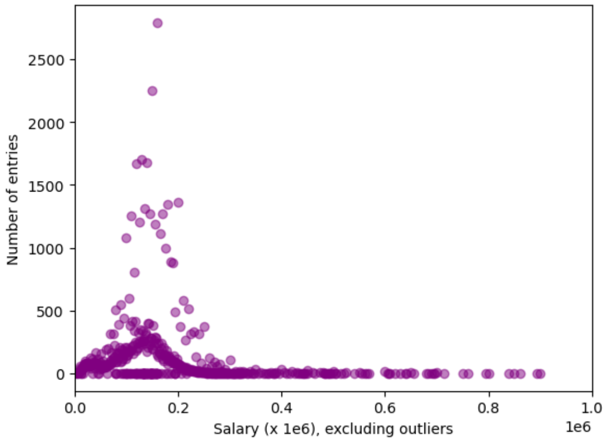

**Figure 3.** Distribution of salaries when bucketed to the nearest 1,000

we found that it was not very successful, so we opted for a different time-based approach by encoding the time. We included one-hot encodings for the year, day, month, and hour. We made sure to disregard 1 entry for each feature since it would be included with the constant 1 entry in the combined feature vector.

Next, we used general knowledge as well as our literature review to realize that years of experience and education are vital to the salary outcomes of a position. As such, the three features that encoded experience were years of working experience, tenure at the company, and level of education. In the dataset, the level of education was indicated categorically (PhD, Master's, Bachelor's, Some College, High School). We decided to transform this into

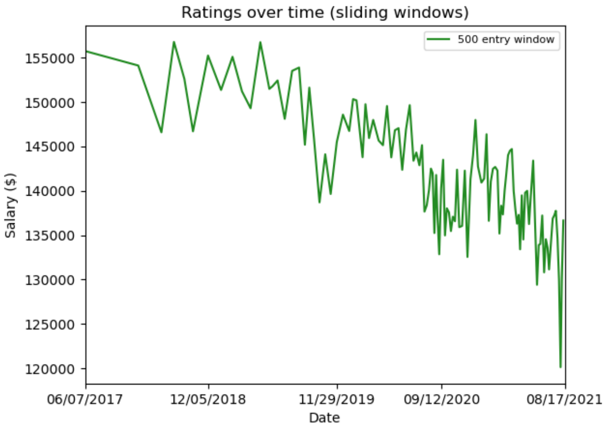

**Figure 4.** Salaries reported over time can be seen to be decreasing

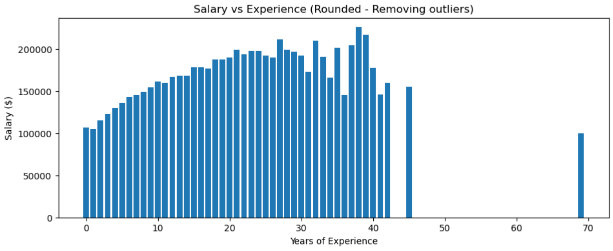

**Figure 5.** Distribution of salaries when plotted against number of years of experience

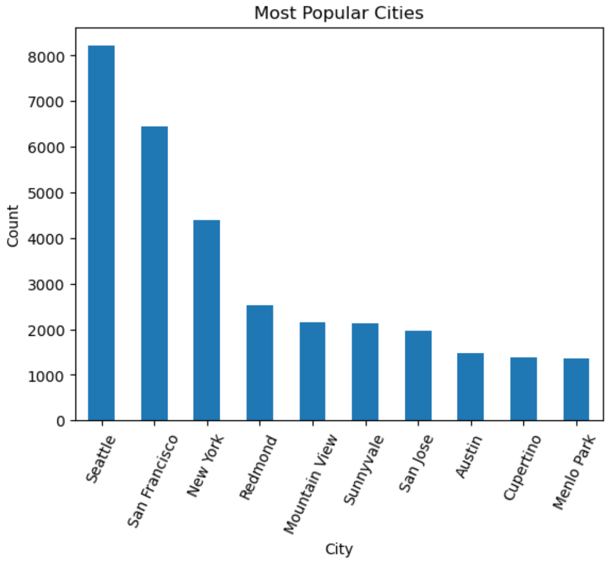

**Figure 6.** Most popular locations heavily correlate with the most popular companies and economic hotspots

years of education for the purposes of our model. For example, PhD would correlate to 21 years of education. Additionally, since a lot of the data was missing values for education, we decided to impute it with the years of education for a bachelor's since we assumed that jobs in data science and stem would require at least a bachelor's degree.

Next, we also understood that the role of the job, company and level of the position is also vital pieces of information in predicting salary. To do this, we developed word-based analysis models to extract additional information from the company, job position, and level. First and foremost, we implemented a bag of words model that tracked the occurrences of each word in the job title, level, and company name. We sorted this list by popularity and tracked the number of occurrences of these words for the feature of each datum. Upon further analysis increasing the size of the dictionary continued to increase the accuracy of the model, which is expected, but also increased the time it took to build the features and train the model. For this reason, we limited the size of the dictionary to 500.

Lastly, we also learned from the literature review that most counties have economic hotspots where high-paying jobs are more common. We wanted to create a feature based on this concept, so we used the Digital Market Area Identification number, or DMA ID, which is a numeric value that separates the United States into geographical regions. We utilized one-hot encoding on the 150 unique identification numbers to generate this feature. This is preferable over simple indexing the city name because it creates a coarser grain that allows adjacent cities to share economic significance.

### 2.2 Error Measurement

In order to assess the validity of the model, we chose to use Root Mean Square Error (RMSE), and the Mean Average Percent Error (MAPE). Ultimately we wanted to use Mean Square Error as an error metric because we wanted to disproportionately punish salary predictions that were very far from the correct value. However, because we were predicting salaries greater than 100,000, we were noticing values that were in the millions. In order to make this more understandable and consistent, we decided to use Root Mean Square Error. Additionally, we used Mean Average Percent Error for multiple reasons. First, we wanted a second error metric to compare rather than just Mean Average Percent Error to report more comprehensive benchmarks. Then, we chose Mean Average Percent Error because it is a very intuitive way to visualize the difference between predicted and actual salary.

### 2.3 Baseline Model

In order to verify that these features were significant, we produced a simple baseline model that included what we believed to be the most important piece of information: years of experience. This model processes the data by bucketing the data points based on the number of years reported. In order to predict the salary of the job position, we took the average of the salaries with the same number of years of experience and predicted that value. If there was a value we had not seen before, we would predict the average overall salary. This benchmark produced an RMSE of 47258 and an MAPE of 41.28.

## 3 Models

### 3.1 Time-based Models

We first started with a time-based model, with the strategy used to predict with this model discussed in the predictive task section above. Again, we decided to attempt using time because there seemed to be a downward trend in the graphed salary data as well as periodic effects. The sliding window and one-hot encoding models seemed to lend themselves well to the above situations. After creating the average predicted salaries list, we used dynamic programming to compute the sum by subtracting the oldest data point and adding in the newest data point. This model had an RMSE (root MSE) of 56,051, an MAE (mean absolute error) of 39,534, and a MAPE (mean absolute percentage error) of 47. We computed a similar model using a sliding window of size 500 but disregarding the most recent data point (to possibly account for huge swings in nearby salaries), and this had an RMSE of 59,947, an MAE of 43,096, and MAPE of 43. The strengths of this model are that it can capture trends within a recent time range and the weaknesses are that the average may not accurately describe the complexity and variance in salaries. Another option, to optimize, would be to explore temporal latent factor models

The next time-based model we tried was to try and fit the approximately piecewise function using a Linear Regressor with one-hot encodings for time. To reiterate, we created four features for this model based on the one-hot encodings for the day, month, and year. We trained on different combinations of those 4 features, with the best model on the training set having all 4 included. On the test set, the RMSE was 55,347, the MAE was 39,450, and the MAPE was 45. This model is tied to the fact of periodicity in the dataset for a particular time resolution, so if the data does indeed have periodicity, for example by the month, then it would be effective, but if there's no clear pattern, then this model wouldn't be as effective. The above approaches didn't have any issues with scalability (running on datasets of 40,000-50,000) points, though since we were randomizing the dataset, there may have been an uneven distribution of time in the train/validation/test sets, causing issues when fitting a regressor. Overall, it seems that using time may not be the best option for this dataset, as it was on par with the baselines, and due to the fact that the correlation between timestamp and the salary a user entered may be entirely random and not necessarily indicative of any periodic trends.

### 3.2 Experience-based Models

The second main model we analyzed was that which encoded years of experience, tenure, and maximum levels of education for each data point. Once these features were created as described in the previous section, we then created a feature vector that included all of these features and trained it using linear regression. We found that years of experience was the most significant feature with tenure playing a slight factor. In our model, the level of education was not that significant, and this can be explained by its absence in a lot of the data. On the test set, the RMSE (root mean squared error) was 51,305 and had a MAPE (mean absolute percent error) of 44. Overall, this model slightly outperformed the baseline, and it was profoundly clear that we would need to include more features to achieve better accuracy in predicting salary.

### 3.3 Word-based Models

Next, we further analyzed our word-based models. Again, there were two primary modeling techniques we used to fit this information: a standard bag-of-words model and a TF-IDF model, which I will discuss individually. We then fed these feature vectors for each datum into a Ridge regression model. After running this model on the test set, we found that this model achieved an RMSE of 36,237, and MAPE of 31.61. Next, we implemented a TF-IDF model. It is important to note here that we ran into significant performance issues with this model, as to create a comprehensive TF-IDF model you must calculate similarity scores between every possible combination of data points, which scales quadratically. Thus, to design a model that could succeed with our given computation equipment, we reduced the number of similarity scores calculated to 20,000 per data point. With this model, we achieved an RMSE of 59782. Comparing the metrics, we found that the bag-of-words-based model significantly outperformed the TF-IDF model for this data.

### 3.4 Location-based Model

The last partial model we built before our final model was the one that considered the one-hot encoded DMA Id. When feeding this feature vector into a linear regressor, it yielded an average percentage error rate of 32% and an RMSE of 44,799. We then set out to improve this prediction model and added another column to our feature vector by adding the user's total years of experience. Now, our improved model considers the user's relative location and their total years of experience to predict their salary. This model significantly improved on our previous model and yielded an average percentage error rate of 24% and an RMSE of 39,289 on our test set. Overall, considering a user's dma_id location and their total number of years of experience proved to be a relatively successful model.

### 3.5 Final Model

Our final prediction model combines the most important and useful features of each model that we created. This model essentially combines the bag of words features, the dma_id one hot encoding features, the time features, and the education features into a single feature vector that would prove useful to predict STEM salaries. Once we had combined these features, we trained a ridge regression model to predict the salaries in the test set. This model outperformed every single one of our previous models and yielded an RMSE of 29,672, an MAE of 19,342, and a mean absolute percentage error of 21%.

## 4 Literature Review

### 4.1 Salary Prediction in the IT Job Market with Few High-Dimensional Samples: A Spanish Case Study

In the first piece of literature we analyzed, researchers at the University of Trento, and Universidad Carlos III de Madrid, developed and implemented various models to predict the relative salary of job postings scraped from online sources. From the job postings, they gathered feature vectors with roughly 2,000 data points each that represent different important aspects of the job posting, such as minimum education requirements and skills mentioned in the job description. They then fed these feature vectors into different machine learning models and were able to achieve an accuracy of 84% when predicting whether the salary would be low, medium-low, medium-high, or high.

Although we are taking a different approach to the prediction model specifically, there is a lot that we learned here about the specific domain knowledge that should be used in the analysis of job posting-related data. First and foremost, we learned that to improve our model further, we should take into account the approximate size of the company as a feature. They also introduced the "per capita gross product" for the geographic region of each company. This allows the model to take into consideration economic hotspots or more expensive regions that would provide relatively higher salaries. We also learned that we should convert the highest level of education for each entry to the number of years of education required to reach the said level of education. We assume that each entry has an education level bachelors. For bachelor's, we use 16 years, for master's we use 18 years, and for doctorates, we use 21. Although the paper indicates that this would be a powerful way to differentiate the experience required for each position, our dataset handles Data Science positions. These types of jobs generally require higher than average education levels and minimize the distribution of education levels in our dataset.

### 4.2 Statistical Machine Learning Regression Models for Salary Prediction Featuring Economy Wide Activities and Occupations

In the paper "Statistical Machine Learning Regression Models for Salary Prediction Featuring Economy Wide Activities and Occupations" the authors Yasser T. Matbouli and Suliman M. Alghamdi outlined various models that estimate average salaries across many industries in Saudi Arabia. They sourced their labor compensation data from professional associations, industrialists, public databases, and third-party companies that specialized in salary surveys. Even though our models honed in on predicting STEM salaries mainly in the United States, the models presented in this article were somewhat similar in their approach to ours. For example, their models for predicting salary were based on a specific economic sector where they also considered occupation and skill level titles in their feature vectors. Similar to most of our models, they also ignored demographic information about their users such as gender and ethnic group. Something they did differently was that they also considered the size of a company by classifying the company as small, medium, or large which was interesting in itself since we did not have that information in our dataset. Overall, they tested many linear regression models as well as non-linear ones which resulted in varying results. They found that nonlinear machine learning techniques improved results over multiple linear regression models. Their first improvement of the linear regression model was to use an artificial neural network to predict mean annual salary across industries. Over multiple models that they tested, the best model they came up with was a Bayesian-based Gaussian process regression model which brought their MAE from 4397 (using linear regression) to 526.

### 4.3 Time series regression model for infectious disease and weather

The next piece of literature relates to the use of time in models. Researchers published the paper to evaluate time series regression (TSR) models for evaluating the correlation between weather and infectious diseases. Although this evaluation was not related to salary prediction, it highlighted some of the aspects of dealing with time in predictive models. The datasets used were cholera cases and rainfall in Bangladesh and influenza cases and temperature in Tokyo. The first dataset was gathered from lab cases from a hospital in Dhaka and a meteorology agency from 1996 to 2008. The second dataset was acquired from Japan's public influenza dataset and meteorology agency from 1999 to 2009. The data for influenza-like illness and El Tor cholera cases were graphed over the time period. The first showed periodic spikes every year, presumably around flu season, while the rest of the year was relatively flat. The second showed more variations with more spikes and valleys, with no clear pattern. The researchers started with a common TSR model, resembling a latent factor model, which aimed to "identify how time-varying factors like temperature explain variation in disease occurrence." The equation included regression coefficients beta and a time function and included a logarithmic term, where the final result was approximately a Poisson distribution. They explored various aspects of the topic including the role of an immune population, the lagging effect of disease in response to weather (autocorrelation), and separate functions for seasonality and long-term trends. ARIMA models and wavelet models were mentioned as alternatives. The results of the research were that "there was a difference in magnitude but not the direction for the effect of temperature in TSR and non-TSR models, suggesting also including other models". However, "the fit of the TSR model improved when including autocorrelation and immunity terms." The discussion in this paper demonstrated how time series regression can be viable for prediction, and can be a key tool in identifying environmental relationships. Though there wasn't a salary-prediction time model to compare to, it can be seen that time can be a key factor.

### 4.4 Salary Prediction via Sectoral Features in Turkey

One piece of literature that attempted to solve a similar predictive task was put together by researchers at Galatasaray University in Turkey. In this paper, they started with a dataset of over 500k entries with 11 features. This data was anonymized and included significant elements of a job posting. Some examples of the features include the position name, company code, record date, and salary amount. They then cleaned and transformed the data, and subsequently fed, deployed and evaluated several machine learning methods that estimated salary. The results from their exploration was that gradient boosting and artificial neural networks fared much better than other methods like linear regression. In comparison to our task, there were a lot of insights from this paper that proved useful in our analysis. Firstly, the researchers had standardized annual salaries to account for inflation, but had found that this proved to have no significant effect on performance. This informed our decision on whether we should transform the time data. Further, it showed that other kinds of models beyond regression such as tree-based models and artificial neural networks could be a better way to predict salary.

## 5 Results and Conclusion

In the end, our final model outperformed the alternative models we had developed. This was an expected outcome as we had intentionally picked the most useful features from our previous models. When comparing our chosen metric: MAPE, our final model yielded 21%. This was significantly lower than what we observed in our previous models (time-based model: 45%, experience-based model: 44%, word-based model: 31.61%, location-based based model: 32%). The feature vector was created by appending the best features across all models and had a length of 724. Further, from the results of our previous models, we can infer that the combination of the feature vectors one-hot encoding of the DMA id and the bag of words model contributed the most to the result we obtained from the final model. This model succeeded in comparison because we used the best resulting techniques from each of the models we tried and because its features summarized more of the entire dataset better than any individual model. In conclusion, we explored different techniques and models and used these insights to design a better model that predicted the salary of data science and STEM employees.

## References

1. https://www.kaggle.com/datasets/jackogozaly/data-science-and-stem-salaries

2. Martín, Nacho & Mariello, Andrea & Battiti, Roberto Hernández, José. (2018). Salary Prediction in the IT Job Market with Few High-Dimensional Samples: A Spanish Case Study. International Journal of Computational Intelligence Systems. 11. 1192. 10.2991/ijcis.11.1.90.

3. Ş. Demir İnan Özer, B. Ülke, F. Serhan Daniş and G. Keziban Orman, "Salary Prediction via Sectoral Features in Turkey," 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2022, pp. 1-6, doi: 10.1109/INISTA55318.2022.9894130.

4. Chisato Imai, Ben Armstrong, Zaid Chalabi, Punam Mangtani, Masahiro Hashizume, Time series regression model for infectious disease and weather, Environmental Research, Volume 142, 2015, Pages 319-327, ISSN 0013-9351, https://doi.org/10.1016/j.envres.2015.06.040.

5. Matbouli YT, Alghamdi SM. Statistical Machine Learning Regression Models for Salary Prediction Featuring Economy Wide Activities and Occupations. Information. 2022; 13(10):495. https://doi.org/10.3390/info13100495