# Questions

1. How long did it take you to solve the problem?

Almost four hours.

2. What software language and libraries did you use to solve the problem? Why did you

choose these languages/libraries?

I used Python 3.8.10, using the following python libraries for my analysis. These are well-known libraries for the given problem. Pandas provide an excellent means to handle spreadsheet data. It is a bit slow than NumPy but very convenient.

* Numpy - Data handling and storing
* Pandas – Data handling and storing
* Matplotlib - Visualisation
* Seaborn - Visualisation
* Sklearn – For model building

3. What steps did you take to prepare the data for the project? Was any cleaning

necessary?

Please refer to the attached Jupiter notebook for more details on it. The data was reasonably clean and needed almost no cleaning.

4. a) What machine learning method did you apply?

b) Why did you choose this method?

c) What other methods did you consider?

Since the data and task are relatively simple with a few independent features, I decided to use a simple solution following Occam’s Razor principle. I selected Linear regression and a reasonably complex Random Forest Regressor for my analysis. Random Forest Regressor achieved better results than simple linear regressor. Other regressors such as polynomial regressor, elastic regressor, lasso regressor, ridge regressor, SVM and artificial neural networks could also be used.

5. Describe how the machine learning algorithm that you chose works.

The linear regressor tries to fit a straight line such that the error is minimum. The main task is to find the sloop and constant of the line equation that fits the data with minimum error.

Random forest is a more complex regressor and works like a black box. It uses several decision trees and averages their results to produce a non-biased and median solution.

6. Was any encoding or transformation of features necessary? If so, what

encoding/transformation did you use?

The dataset consisted of 4 categorical features and two numerical features. We excluded jobid and companyid as they are meaningless for our analysis. The categorical data is usually converted to numerical values before the model training. Sometimes, converting categorical data to numerical introduce order or bias in the data. That’s why I choose the one-hot encoding to convert categorical data to dummy variables with binary input. One hot encoding is known to increase the dimensionality of the data, but in our case, the dataset was simple enough to tolerate more features resulting from one-hot encoding.

7. Which features had the greatest impact on salary? How did you identify these to be

most significant? Which features had the least impact on salary? How did you identify

these?

I hadn’t got the time to look into these aspects.

8. How did you train your model? During training, what issues concerned you?

I split the data into 70-30 training and testing. Since the data was big enough, there was no need for cross-validations. We could use grid search for optimal hyper-parameter tuning, but I hadn’t looked into it and trained the models with default values.

9. a) Please estimate the RMSE that your model will achieve on the test dataset.

The random forest regressor I trained had achieved RMSE = 10.23 on the training dataset. The model is expected to reach RMSE = 50+

b) How did you create this estimate?

I have split the training features into 70-30 training and testing datasets. I estimate these results based on my model’s performance on the unseen test dataset.

10. What metrics, other than RMSE, would be useful for assessing the accuracy of salary

estimates? Why

The error metrics are all driven from one value. Mean absolute error, Mean squared error, and Root mean squared error is just alternative metrics. The most distinctive one is the R-squared. It is suitable for model comparison as its values always lie between 0 and 1. I have used All these metrics for my analysis.