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

Around 4 hours. I spent around an hour exploring the data, an hour building the model, an hour tweaking the model, and an hour training the model.

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

I used python and mostly sklearn within python.

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

I did not find anything too weird with the data and everything seemed to be within normal boundaries. I normalized the ‘miles from metropolis’ and ‘years of experience’ to be between 0 and 1. I dealt with categorical that was labeled as ‘none’ by treating this as a separate category. Alternatively, I could have simply not had a feature if the label was ‘none’

4. What algorithmic method did you apply? Why? What other methods did you consider?

I used simple linear regression. I could have done feature engineering on the input vectors or I could have also tried using ridge regression or lasso with linear regression. I could have also used a simple feed forward neural network to automatically feature engineer. I chose the linear regression because it is simple and does not take long to train. Given that the assignment should be done in 4 hours, I chose this simple method that will run quickly.

5. Describe how the algorithmic method that you chose works?

Linear regression has the form:

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where the beta’s are the fit parameters and x’s are the features. The model assumes a linear relationship between features and output (salary). The parameters are fit by minimizing the mean square error.

6. What features did you use? Why?

I transformed the categorical data (company id, job type, etc.) into dummy variable vectors. I also normalized the numerical data to range between 0 and 1. I did not try to further engineer these features due to limited time.

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

I used the sklearn library to train the model. I was most concerned about overfitting the model. I did not do a full cross validation check but only a single test-train split. I was also concerned about the amount of time that it took to train the model. I have an older laptop so the training did take some time on the data set.

8. How did you assess the accuracy of your predictions? Why did you choose that method? Would you consider any any alternative approaches for assessing accuracy?

I used mean square error to access the model. I compared to the value of a simple average model to understand this value better. I also made a plot of the predicted salary versus the actual salary. I could have also measured the R value since this is a linear fit. Additionally, I could also measure the mean square error per company, per industry, etc. to assess where my model is falling the greatest. I could then decide ways of improving the model given this additional information.

9. Which features had the greatest impact on salary? How did you identify these to be the most significant? Which features had the least impact on salary?

The features that had the most significance was years experience and miles from metropolis. I ran the model with each of the feature sets removed and calculated the error for each. All of the rest of the features seemed to change the error very little. This was a bit surprising to me and if I had more time I would definitely dig further into this problem.