# Indeed data exercise questions – Alan Au (4/12/2018)

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

It took me about 4 hours of actual work, plus extra time waiting for my model to fit and predict.

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

I used Python via a Jupyter notebook. For model-building, I used the scikit-learn package.

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

I did some basic exploration of the data to check for missing or poorly-specified data, and to check the data distributions. As part of pre-processing the data, I chose to one-hot encode my categorical variables, which was probably the bulk of my preparation.

I did some minimal cleaning to remove a few listings where the salary was listed as $0, but generally the data looked pretty good. (I did choose to retain the ‘NONE’ listings as a separate categorical entry, rather than excluding or replacing them.)

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

I chose to use a random forest regression model. This had many advantages, including the easy combination of categorical and continuous features. It also gave me built-in cross validation, and made it easy to extract feature importance. I didn't spend much time considering other models.

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

Random forest is a tree-based model, so it uses differences in the feature space to split the dataset until it reaches an outcome. During forest construction, it subsets both the input data and the feature space, comparing the outcome for each of the generated sub-trees, and then combining them together into a final aggregate model.

**6. What features did you use? Why?**

I ended up using almost all of the fields except for ‘jobID’. I debated whether or not to include ‘companyId’ as an input, but decided that because it was a finite and relatively small number, it might be worthwhile including. I also debated whether or not to encode job level as continuous variable instead of a categorical level, but decided that there might be difference in how different listings defined those levels, and it would be ordinal at best. In addition, the executive-level positions would be difficult to rank.

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

I trained my model using a standard feature matrix. Initially, I was a bit concerned that the test data might contain values unseen in the training data, but I checked for that specifically, and it looked okay.

I was also uncertain how much the inclusion of 'NONE' values for 'degree' and 'major' might influence the model, but I ended up splitting out all of the categorical variables anyway as part of encoding them. Ideally I would have preferred to find an encoding that grouped the categorical data in a better way, mostly to keep the feature space manageable, and to better gauge the overall impact of each category.

I didn't do a lot of testing of the hyper-parameters to see how the number and depths of trees affected my model. Given more time, I might try more combinations (e.g. a grid search). However, because it took a while to re-train the model, I decided against it.

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

I used the provided out-of-bag score, roughly equivalent to an R2 score assessing squared error. This is fairly standard for model evaluation. There are some other metrics that provide comparable scores for regression models.

**9. 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 generated a feature importance report from my random forest. Curiously, if the listing was for a ‘JANITOR’ position, that seemed to have the highest impact on salary. After that, ‘yearsExperience’ and ‘milesFromMetropolis’ seemed be the most important. Generally, the ‘jobType’ feature seemed to have a pretty strong impact, except when evaluating CTO and CFO positions. In general, ‘companyId’ did not strongly impact salary.