Salary Prediction Model

A project completed by Sarah Hudspeth

The Problem Defined

The task is to take information from a given description and predict a reasonable salary given job title, years experience, mile from a big city, major, and degree. The information has been take from a description and is in a csv ready to be put into dataFrame.

Once the data is cleaned, pre-processed and analyzed, the goal is to train and fit a learning model that gives a score of less than 360 MSE.

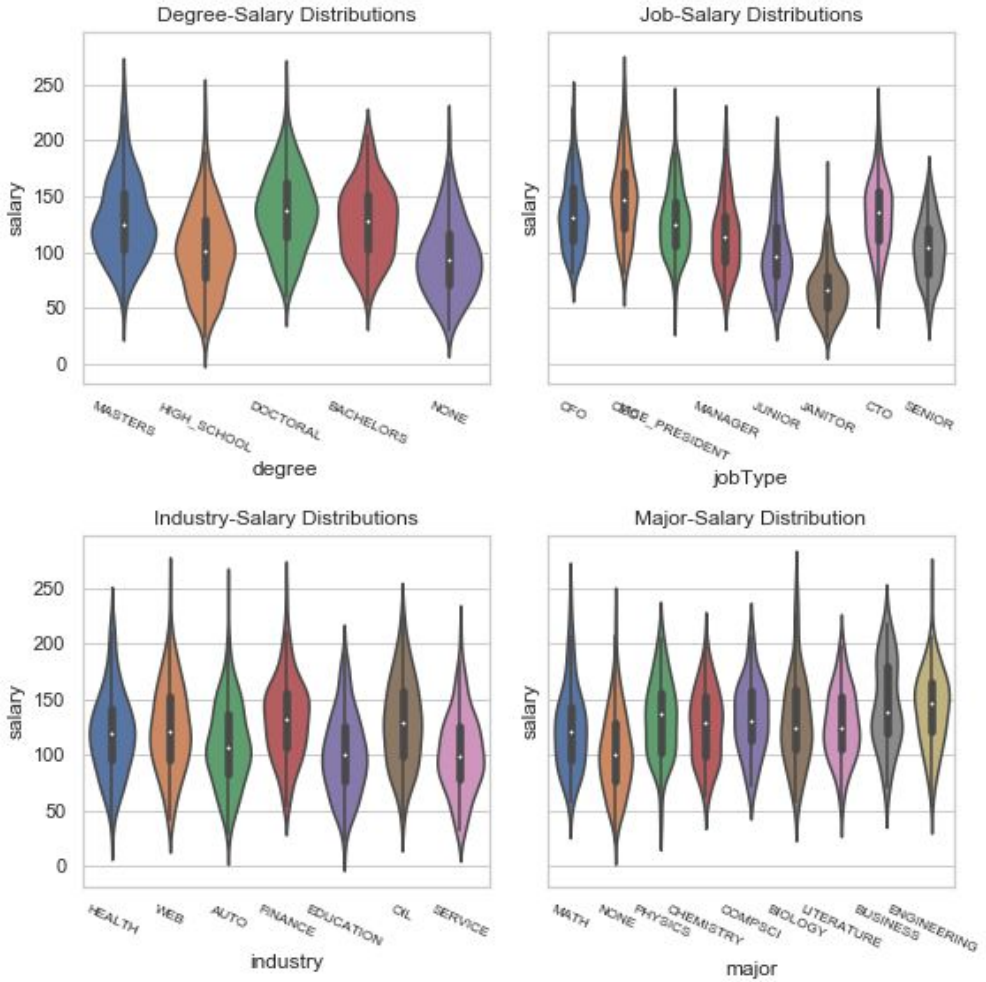
Interesting Data Points

All the categories have pretty close to normal distributions, given the violin plots shown to the right. There are a couple slighlty modal distributions (Bachelors, Finance, Engineering). However, the sampling is very evenly distributed, as shown in the graphs on the next slide.

‘None’ and ‘High School’ show slightly lower means, as does ‘Janitor’

There do seem to be some outliers and I have not accounted for them as of yet.

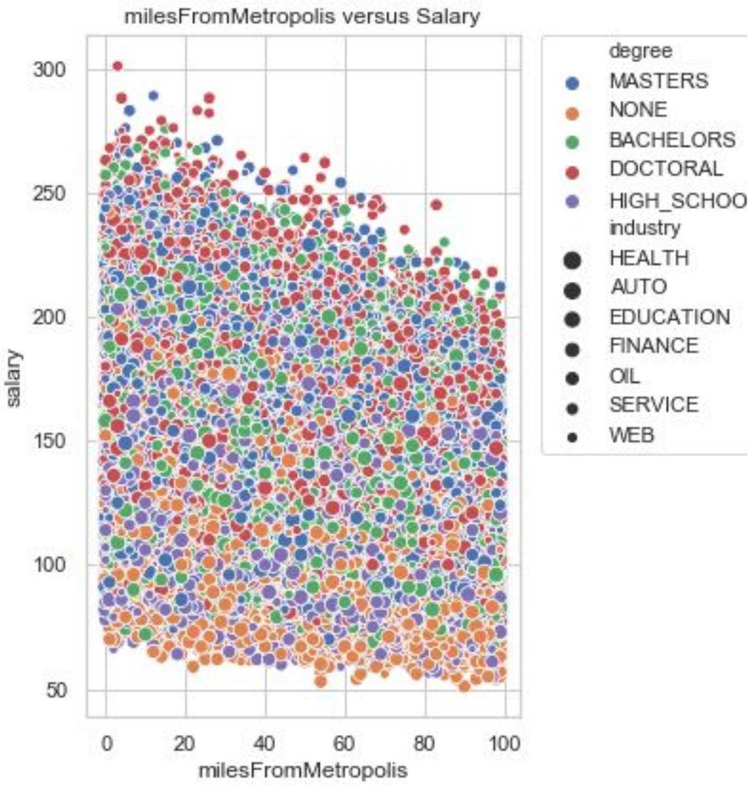
OVerall, the data is very TIDY. not much to clean.

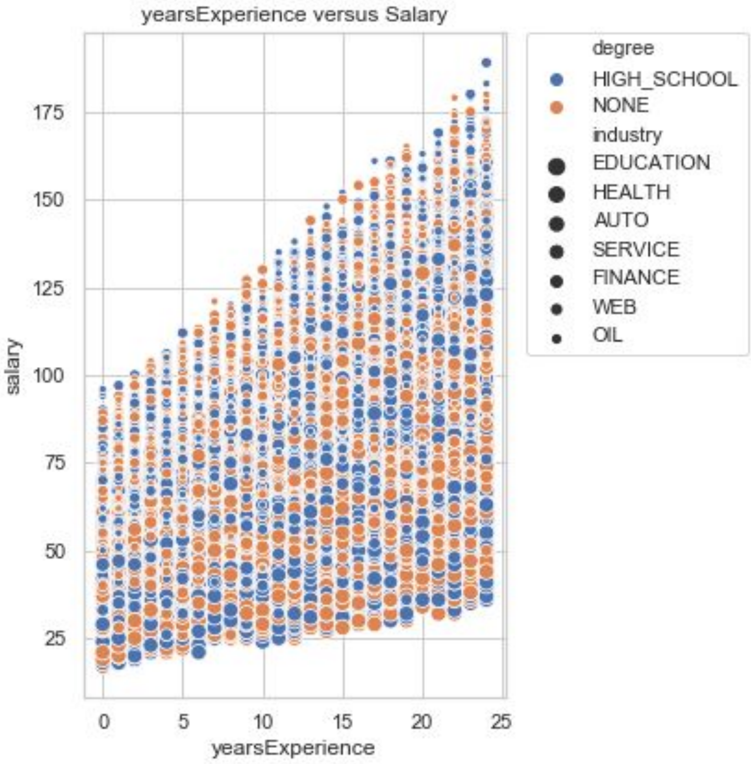


The Data

The scatterplots to the right show a sample of the weak correlations of the features, If one looks by color one can see that ‘None” orange is on the lower spectrum of salaries, but that there is a lot of overlap at all levels.

The graph to the right is specific to non-degree and shows years Ex correlates to a higher salary.

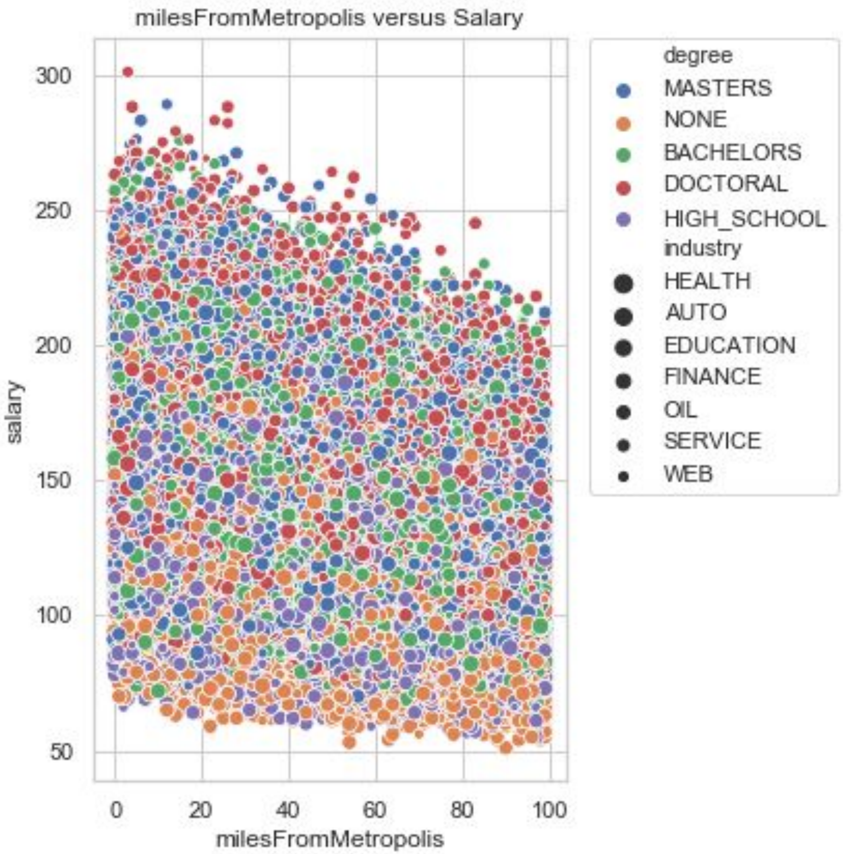




More Scatterplots

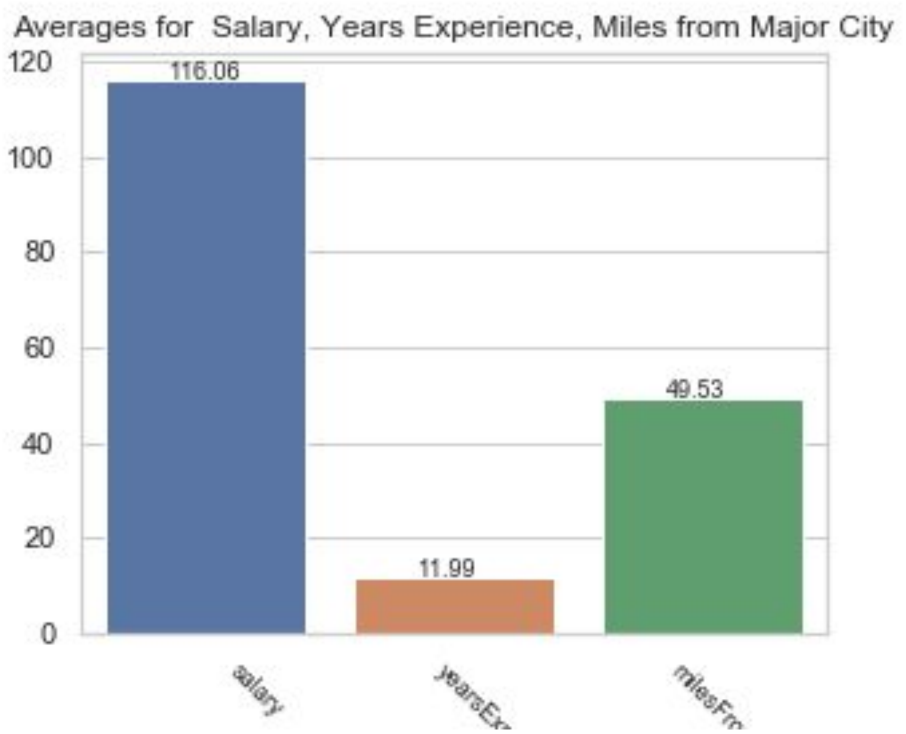
There is an option in the code to build more scatterplots to see correlations. I included three here, but I made more during EDA.

There isn't much clustering--though there is some in the "CFO" scatterplot. All degree-levels are found at all levels, but more doctoral degrees are at the top. And more 'none' are at the bottom--though everything is very interspersed!



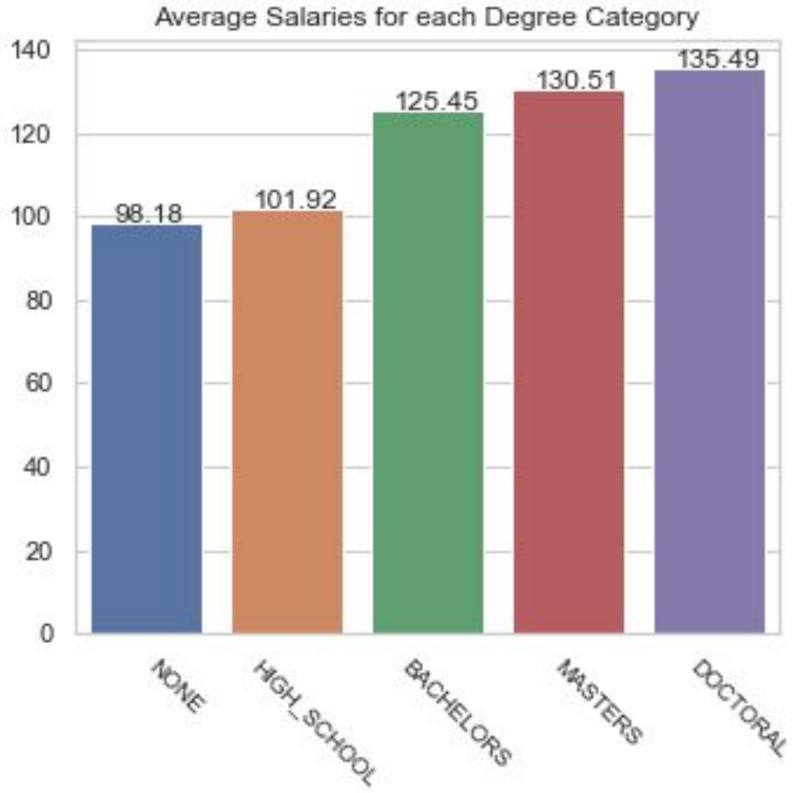
Means

Baselines for general EDA.



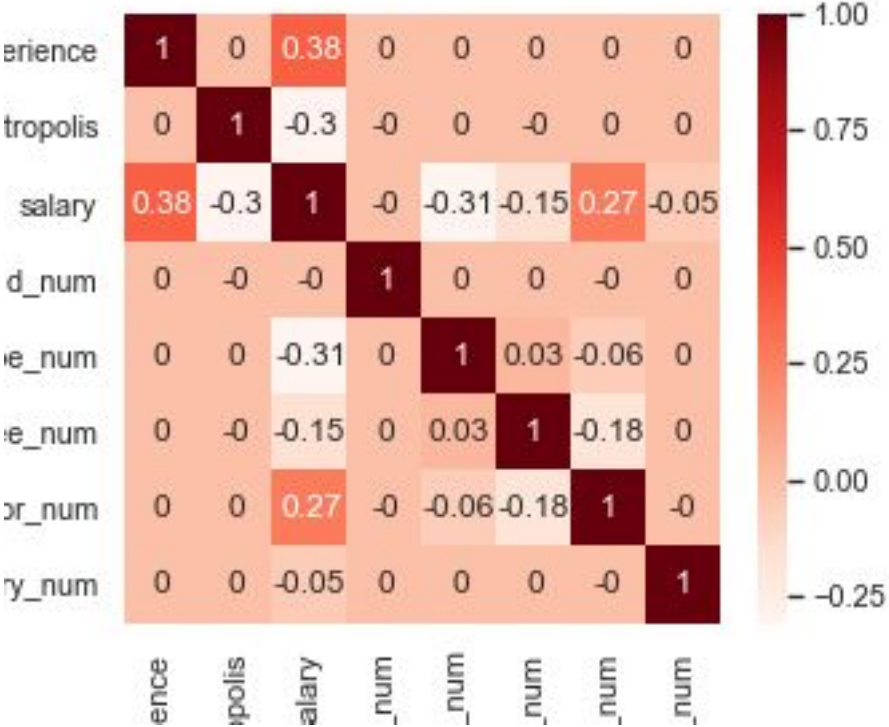
More Means

By Degree, here we can see some more stratification that will play into feature engineering and the models we choose to train the data.



Correlation Matrix

With the features originally given, here is the correlation matrix. Categorical data have been encoded numerically.



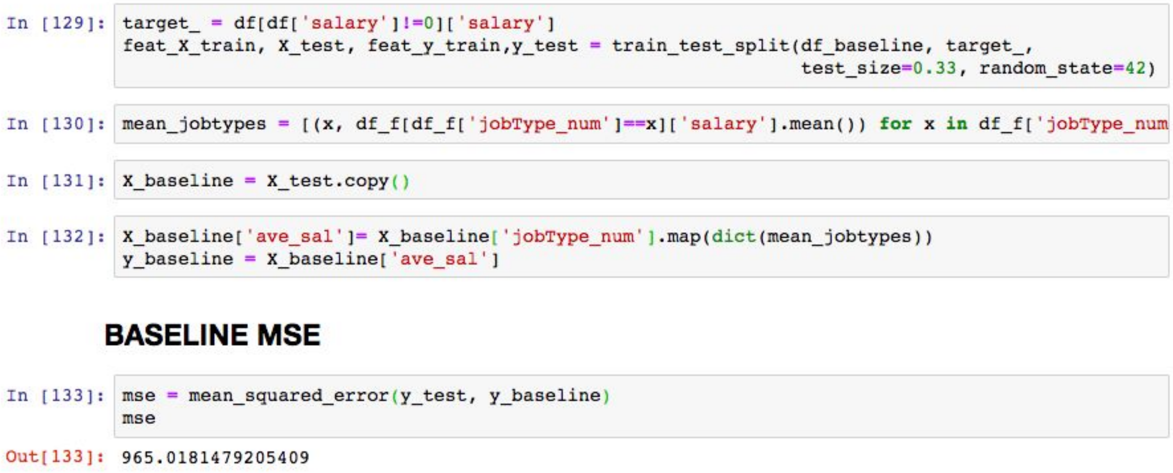
“High Weak” Correlations (+ or -) to Salary: Years Experience Major
jobType MilesfromMetropolis

Features that do not affect Salary:
Industry Company Id

BASELINE MODEL

The baseline model was created by taking the average mean of each job title and using that as a y-hat prediction. Code is shown to the right.

The baseline Mean-Squared-Error score was 965 rounded.



```
In [129]: target_ = df[df['salary']!=0]['salary']
          feat_X_train, X_test, feat_y_train,y_test = train_test_split(df_baseline, target_,
                                                              test_size=0.33, random_state=42)

In [130]: mean_jobtypes = [(x, df_f[df_f['jobType_num']==x]['salary'].mean()) for x in df_f['jobType_num

In [131]: X_baseline = X_test.copy()

In [132]: X_baseline['ave_sal']= X_baseline['jobType_num'].map(dict(mean_jobtypes))
          y_baseline = X_baseline['ave_sal']
```

BASELINE MSE

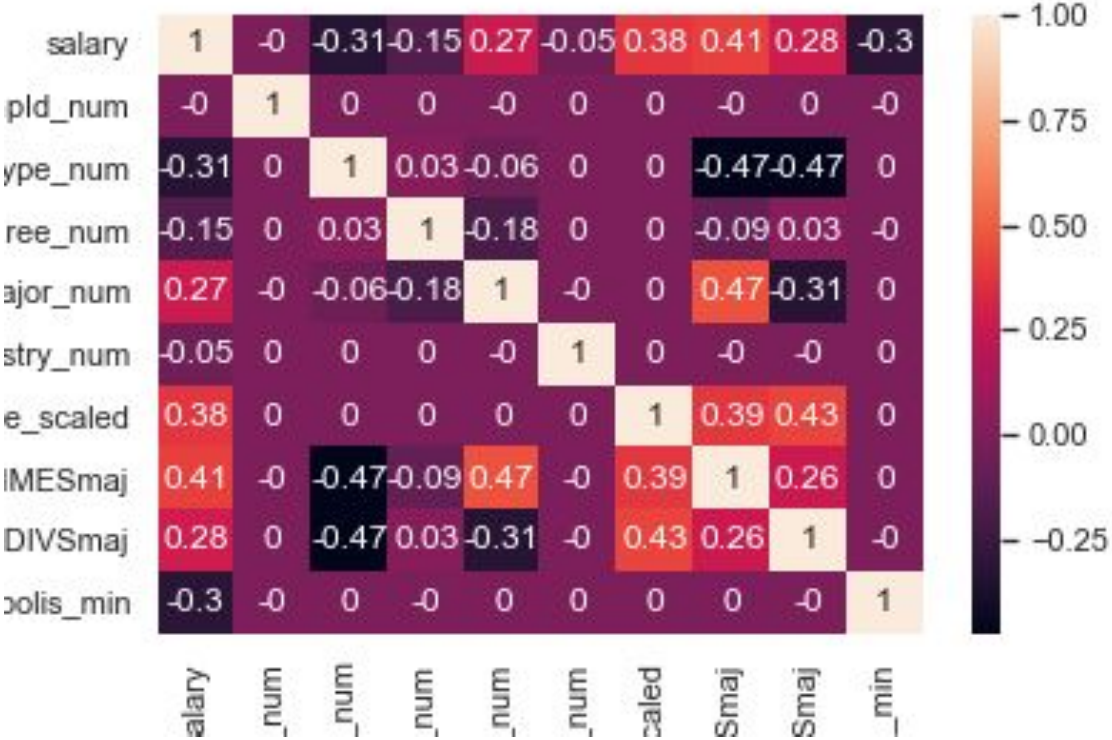
```
In [133]: mse = mean_squared_error(y_test, y_baseline)
          mse

Out[133]: 965.0181479205409
```

Hypothesizing solution

Given the correlations, boosting milesfromMetropolis, yearsExperience and ‘major’ may help the model. Squaring did not increase, but some multiplying and dividing may (spoiler: they helped some, see new feature correlation matrix).

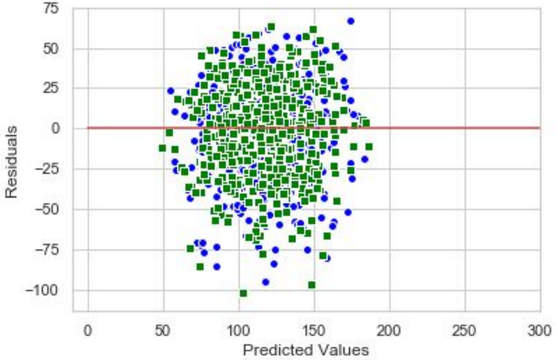
As for learning models, Random Forest Regression, Gradient Boosting, and Linear Regression seem to be a good place to start. There’s not ‘clustering’ so KNN seems illogical and a Logistic Regression causes a non-convergent error.

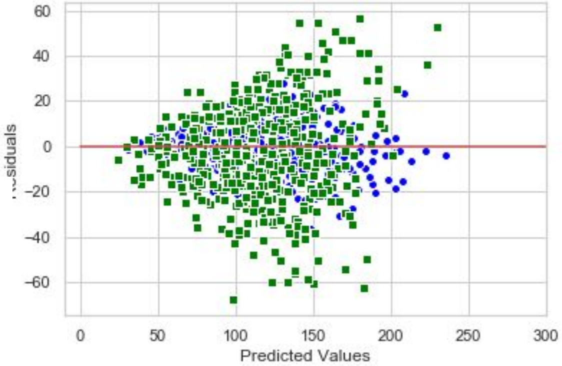


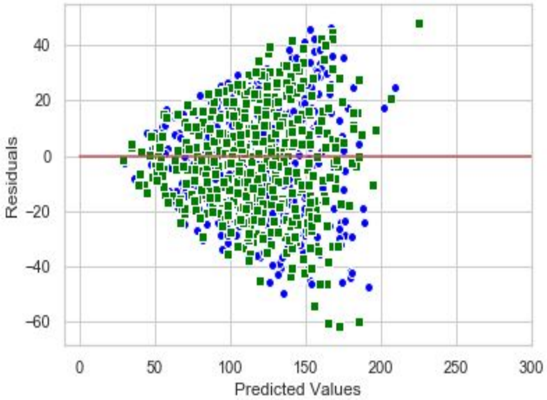
Training the Models

See the three residual plots to the right.

The Gradient Boosting Regressor gave the best MSE score. From the residual plot it seemed to not do well with ‘higher’ salary prediction. This problem an area worth exploring with more features or hyperparameter tuning.







Deploying Code

I have not timed this code or optimized it, but that is something worth exploring and improving upon for me!

Predictions!

Based on a recommendation in a forum, I did withhold some of the training data to get one last “EFFICACY” score of the data in order to judge the model and the predictions made. This MSE was 357.

I think there is room for improvement in my deployment code and also in hypertuning my parameters. I think the model does poorly predicting higher salaries--and that is easily shown in the scatterplot. It’s a bit random as to who gets the high ones! I have some more hypotheses on making more rules to parse the data that could improve the score. To Be Continued!