Kaggle competition I

Regression Challenge

Baby Peanut’s Fan Club

**Team Members: Alexander Haynie**

**Benjamin Goldstein**

**Carlos Prieto**

After all of the libraries were imported in, we began to train with the train data. From the x train data, we removed the id, price and date. We removed the id because a house’s id does not have an effect on the housing price. We removed the date because all of the houses were sold in the month of May, and a few days difference does not seem significant enough to warrant us to involve the date. We removed price from the train data because it is the factor for the y train data. After we removed those elements, we split the lists of the data into random train and test subsets using the sklearn library function “train\_test\_split.” From this we used the x\_train and y\_train on a linear regression function. This resulted in a mean squared error of 38508988841.80 and a regression score of .7. The mean squared error was so high and regression score so low because a linear function was not a good representation of the data with so many factors. To improve the MSE and regression score we decided to utilize ensemble from sklearn. Ensemble contains boosting methods that combine several weak models to produce a powerful ensemble. Ensemble has a function called Gradient Boosting Regressor that we used in our code. Gradient boosting regressor builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function. In our case, we optimized the least squares regression using ‘ls’. With this regressor created, we fit our x and y training data to the regressor. The score that was returned from this method was .9 which is dramatically greater than .7 and resulted in a mean squared error of 113743.68. That is a 3850785098 difference. We used this method to predict the housing prices of the validation data and exported the results to a .csv file.

# 