Kaggle Team Member: Josh Wu, Shang Shi, Jingru Cheng
Kaggle Team Name: Jingru Cheng

Our public leaderboard score is 13327490964.91300, but later on we got a better public score 12194058688.40840 with a private score 4716673391.62684, which does not show up on the leaderboard somehow…(We did submit the prediction by deadline.)

Description:
At first we split the data into a training set and a test set, using lasso regression to do variable selection and prediction. However, the prediction results have a relatively high test error rate and low r squared value, which may be due to the fact that some predictors have a non-linear relationship with P while lasso regression is based on least squares (linear relationship) and regularization. Considering that lasso can simplify the model by returning coefficients as 0s for predictors that is not much linearly correlated with P, we still decided to apply it to all of the data to do a variable selection. As a result, we had Y, R, S, YB, Sch, and L as important variables.

Next, we used polynomial regression to explore the degree of polynomial for the variables that were selected by lasso, which gave the results as below.

Variable	Degree of Polynomial
Y	5
R	1
S	2
Sch	2
L	2

Since R and Sch are both qualitative variables, we thought that fitting a GAM can automatically create dummy variables for them, which may give better results than simply fitting a polynomial model. Therefore, we used smoothing splines for the other quantitative variables with a degree of freedom same as degree of polynomial and left the qualitative variables as what they are. Thus, we had the model gam(P ~ s(Y,5) + R + s(S, 2) + Sch + s(L,2), data = house). By running ANOVA test, we found that the GAM with smoothing splines truly did better than the polynomial regression lm(P ~ poly(Y,5) + R + poly(S,2) + poly(Sch,2) + poly(L,2), data = house). Plotting the GAM, we found the non-linear relationship between these variables and P, which further confirms that this model could work relatively better.

In the end, we calculated the test error and r squared values of this GAM by randomly splitting the data into different sizes of training sets and test sets. We got a stable average test error of 3~4E9 and an average r squared value of ~0.95. As a result, we used this model gam(P ~ s(Y,5) + R + s(S, 2) + Sch + s(L,2), data = house) to predict the P in the test data and achieved our lowest test error rate.