**Stat 602 Class Project Weekly Updates**

**Competition: NYC Real Estate Price Competition**

**Team Kaggle Name (if Relevant): k fold k times**

**Team Members:** Reid Vincent Paris (Vinny),Subrata Pal, Amin Shirazi

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Week 12 (April 12-16) Summary of Activity and Progress:**

**Amin:**

We spent most of our time on understanding the data, cleaning the data, and dealing with NA’s in the data to come up with tidy data. One major problem was the abundance of zeros in the data for ‘residential units’, ‘commercial units’, ‘land square feet’, and ‘gross square feet’. Instead of removing them from the data, I tried some naïve imputation approaches. I tried to replace the zeros with the median of the variables in the same area. i.e., I first filtered on the properties in the neighborhood in which there are non-zero values for the four variables I mentioned, and then tried to take their medians to come up with some values for the zeros. I am not sure if that is the best approach, but this might be helpful. I also worked on the NA’s for factors in the data such as tax class and building class. We ran a multiple linear regression on the raw data first (where we put aside all rows with NA) and then on the data after replacing the zeros and NA’s.

**Vinny:**

I focused mostly on data exploration, especially in reference to the residential/commercial/total units and year built. For the year built there appears to be a cut off year where the variance past it (1959 ish) becomes much more stable. As for residential units it turns out that log(res\_units + 1) actually seems useful although co-ops muddle the water since it appears that routine apartment/condo/townhouse sales can/will be posted with the co-op statistics (eg 800 units in total). A cheat around this is to fit two models to residential units (one before a certain number of units and one after, found minimizing SEL (for log(price + 1)). For both residential units and year built a binary variable was created to mark the change in the model and to prep for interaction terms in our model. Total and commercial units seem to have little to no effect. Also, I grabbed inflation data for Queens County NY single family housing from the Federal Reserve Bank (#EndTheFed #RonPaul) although I’m uncertain if it will be worth estimating/using for moving only one year into the future.

**Subrata:**

First, we started Exploratory Data Analysis of the data and looked for anomalies or other problems in the data format or data values. Histograms of some covariates(areas in square feet) were highly skewed, and transformations (such as log-transformation) gave them more uniformity or ‘Gaussianity.’ Sometimes it can be seen that the same building was sold many times in a short interval, or there are buildings in a very close neighborhood. Vinny inferred that those would be the co-ops with high sq feet area. There were minor differences in input data format for some of the covariates, like the building class, which were taken care of. Other minor works include the change of date formats. We also studied a Multi-Dimensional-Scaling of the numerical training data covariates with color related to the price to look at possible cluster formation.

**Week 13 (April 19-April 23) Summary of Activity and Progress**

**Amin:**

This week we focused more on cleaning the data, feature engineering and implementing different ML algorithms on the data. I used zip code to extract longitude and latitude information about the properties. These were helpful features as we believe that the most important thing in determining house price is the location. I tried extreme gradient boosting on the data but did not get promising results. Neural network was the next algorithm I considered. The prediction error I got on the training set was 0.18, and on the validation set was 0.19; however, the error on test case was crazy. I don’t think the overfit is that huge. I should have made mistakes somewhere in transformation on the test set. A challenging part of the feature engineering is the categorical variables which should be transformed to factors first, and the large number of factors is troublesome to some extent.

Vinny: I’ve been playing around with knn and trying to sort out how to deal with NA values. Knn in the caret package does not allow for “NA” values (not even na.action = na.pass will return the full data set for predictions). The best model I have is a knn with fit with covariates that occasionally have NA values (notably SALES.DATE). Dropping those covariates appears to have a strong negative result on the predicitons. My current plan is to use Subrata’s gbm model (our best so far) to predict the cases with NA present, train a knn model with the full covariates for complete case data, predict on the test data for the compelte cases, and then use the gbm for all incomplete cases.

**Subrata:**

We tried mostly tree based and boosting based methods like ‘gbm’, ‘rpart’, ‘random forest’; modified linear regressions like PCR and PLS; spline based methods like ‘MARS’ and SVM. Among them, Partial Least Square and Principle Component Regression had a bad cross validation error than other models mentioned – possibly modeling using linear functions directly is not very effective for this data. From the CV error, it seems that the decision trees (‘rpart’) requires more hyper-parameter tuning. Random forest, ‘MARS’ and ‘SVM’ CV errors were not bad either. Some of the tree based methods can inherently handle the NA values and usually, for those cases, CV-error was smaller. The best CV-error happened for Gradient Boosted Machine(GBM) with some specific tuning parameters which currently have best error score.

**Week 14 (April 26-30) Summary of Activity and Progress**

**Vinny:** I finished the knn plan I had posted last week with lackluster results. Beyond that I worked some on estimating inflation for 2006. I grabbed the data we downloaded earlier from the federal reserve and messed around some on predictions. From there I corrected the knn values since they did not contain sales date. Because we had a free submission about to expire I threw in another correction for the gbm half of the data and saw a sizable improvement. Since then I have been running a sequential computer experiment on inflation for whenever we don’t have new models to submit. There has been a small but present improvement (probably), could be non-LOOCV messing up things.

Amin: Neural network did not do as expected, so I gave up on that at this time and maybe the prediction could be useful in building and ensemble model. I worked on latitude and longitude one more time as we saw unusually positive and negative values. I should have probably done that the first time I added them to the data. I fixed the issues in the hope of getting better results. My effort on extreme gradient boosting was not successful and not an improvement on the prediction was made. Again, this could be saved for an ensemble model.

**Subrata:** I tried Support Vector Machine with Gaussian Kernel but it did not gave improved results. I am also struggling with some errors from ‘cubist’ method. Apart from that, I mostly tried different fine tuning to the existing models whenever the chosen hyperparameter is close to the boundaries of hyperparameter sets. However, the improvements in the train CV error was minimal. Currently, I am trying to other ways of using inflation and would rely on ensemble models next. Also, we noted lat-long and some problems in the existing procedures which might make predictions better.

**Week 15 (May 3-May 7) (Finals Week) Summary of Activity and Progress**

**Subrata:** We tried to implement the stacked ensemble models. The stacking was done with the help of a glm model(logistic) and with the help of a random forest model for the data with lat-long and clustering index included. The base models contained multiple linear regression, random forest, kNN, gbm, decision tree and MARS. Among these, kNN predictions were relatively uncorrelated with others. Both stacked models were not that good and have not improved the scores. The simple average ensemble was unfortunately not good either. We are little puzzled about why these methods are practically not working very much. Somehow the training data is possibly over-fitting or/and we have to do better feature engineering.