# House Price Prediction

Adrianne Fu, Grant Liu, and Linyi Lyu

ORIE 5741: Learning with Big Messy Data

Cornell University

December 6, 2021

## 1 Introduction

Both individuals and institutions hope to invest in the real estate market with the best interest rate and minimal risk, and hope that it will be the best investment for the long term. Housing prices have maintained an increase of around 5% for many years, however, during the pandemic, there have been double-digit price increases, and the current housing supply is at its lowest level since the 1970s. This phenomenon arises increasing concerns about the ability to buy houses while decision makers are being inundated with extremely messy data.

The purpose of this project is to build a model that helps predicting house price in the future and find the factors that influence the change in house price.

## 2 Data Overview

This part contains a detailed description of the datasets we used to solve the problem. They are collected from different sources.

### 2.1 Variables

The house price data we used is from Zillow website. It includes the region information (state, county) and Zillow Home Value Index (ZHVI) in each month. ZHVI reflects a smoothed, seasonally adjusted measure of the typical home value within the 35th to 65th percentile in a given region. ZHVI will be the response variable that we are interested in predicting in this project. The unemployment data includes the unemployment rate in different regions in each month. The hospital data lists all registered hospitals in a given region together with the rating for each hospital. Similar to the hospital data, the school data contains information of all the schools in each region. We also collected the consumer price index (CPI) and interest rate data since they reflect the national economics level. The income data includes the average income and income inequality score. The education background data shows the percentage of residents with a bachelor's degree or higher and a high school degree or higher. The population data includes the estimated migration and total residents size.

Except that state and county are categorical, all the other variables are numerical, and their timeliness are classified as follows:

A. Variables constant in all months for a given county: hospital count, hospital rating, private schools count, public schools count.

B. Variables constant in all counties for a given year: CPI, interest rate.

C. Variables with different values in different months and different counties: ZHVI, unemployment.

D. Variables with different values in different years and different counties: personal income, income inequality, percentage of population with Bachelor's Degree or higher, percentage of population with high school graduation or higher, net migration rate, resident population.

### 2.2 Data Processing

There are a total of more than 280 thousand observations in original house price data available on Zillow, including regions that are not covered by American Community Survey (ACS). Besides, some variables such as personal income are missing federal census records of 2020 possibly due to the pandemic. Thus, we only retained regions and times with valid ACS reports – a total of 93835 rows of per county per month house price data from Jan. 2010 to Dec. 2019 of 781 distinct counties across the United States, along with according variables listed above.

### 2.2.1 Imputation

For missing values of ZHVI in house price data, we used both forward and backward filling across date because some rows miss house price information from the start of 2010 and some at the end of 2019, only using one of the sequential filling method might not be able to impute all missing values. For the regions with too many continuous missing entries across time (>2 years) under the ZHVI column, we chose to drop those regions.

After getting a cleaned house price data, we merged all the other datasets with it by state, county, and date using these columns in the house price data as primary keys. The merged dataset has 288406 rows and 24 columns. As classified in 2.1 some variables are relatively constant and do not update by date (eg. hospital or school resources), and some only update yearly (eg. income, population), so we impute them with the constant value or the value in the same year.

### 2.2.2 Encoding

We one-hot-encoded the categorical variables state and county. This step increased the number of columns from 24 to 641.

## 2.3 EDA

In this section, we implemented exploratory data analysis (EDA) to investigate the data distribution and the relationship among variables.

Figure 1 shows Spearman's rank correlations (better in finding nonlinear correlation) between each pair of numeric variables in our dataset. We can see that some features like proportion of population with higher education degree, and income are positively correlated with ZHVI, while unemployment rate is negatively correlated with ZHVI. It should also be noticed that some explanatory features are also strongly correlated such as residence population and income, CPI and year, unemployment rate, and interest rate.

We are also interested in the overall trend of the house price change and time. Their relationship is plotted in figure 2. It can be observed that before 2012, the overall house price is declining probably due to the impact of 2008 economic crisis, but after 2012, the house price has been keeping a stable rising trend.

Figure 3 plots how the house price index changes by month. It shows that except in year

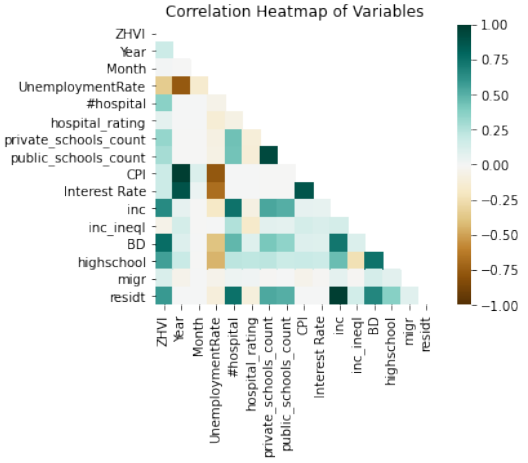


Figure 1: Correlation heatmap of variables

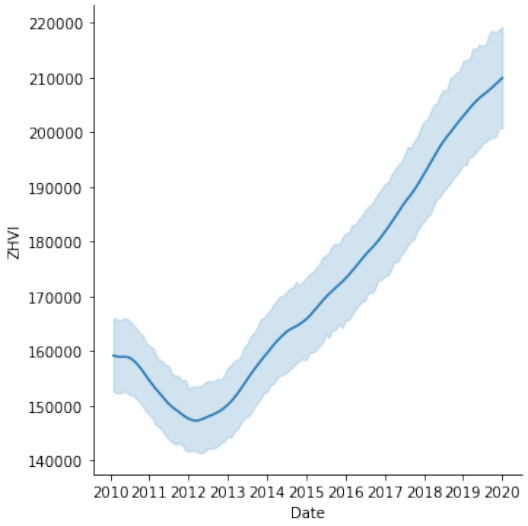


Figure 2: ZHVI vs. date

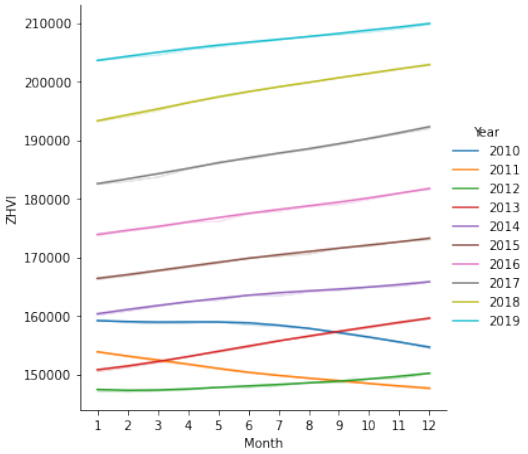


Figure 3: ZHVI vs. month

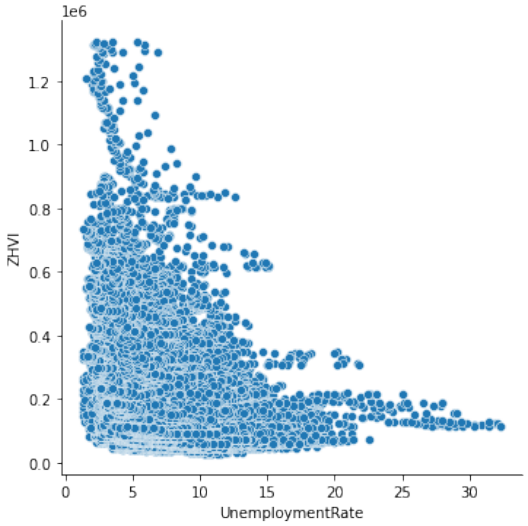


Figure 4: ZHVI vs. unemployment rate

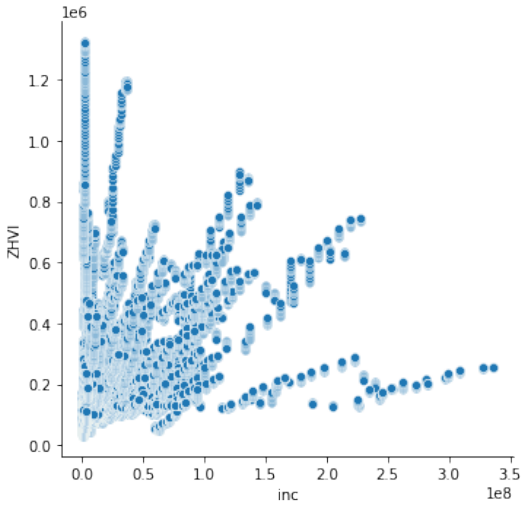


Figure 5: ZHVI vs. income

2010 and 2011, the overall house price increases stably in a year, and there seems no seasonal impact on ZHVI. This result is not surprising because as we stated in 2.1, ZHVI is already smoothed and seasonally adjusted.

Figure 4 to 7 shows the pairwise relationship between ZHVI and the features that revealed a strong correlation with ZHVI in the correlation heatmap which are unemployment rate, income, population ratio with bachelor's degree or higher, and residence population. It seems that the relationships are not linear which implies that linear models will probably not have a good performance on this dataset.

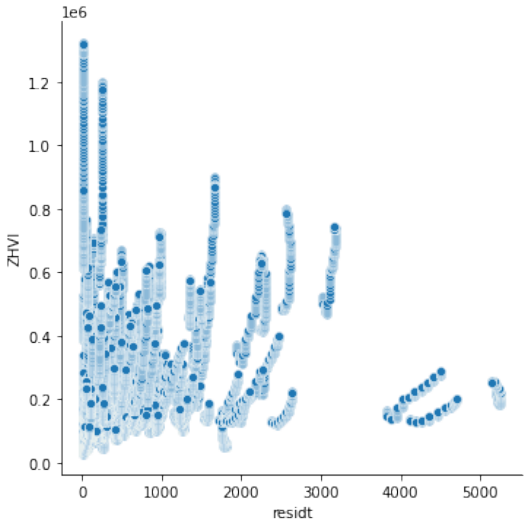


Figure 6: ZHVI vs. residence population

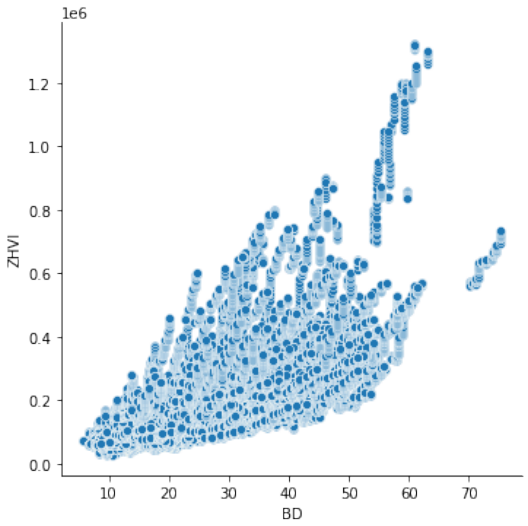


Figure 7: ZHVI vs. population ratio with bachelor's degree or higher

| Variable | Coefficient |
|---|---|
| Intercept | 943e11 |
| ZHVI | 226453.53 |
| Previous ZHVI | -119387.84 |
| Year | 2339.42 |
| Month | -436.47 |
| CPI | -4399.16 |
| Unemployment Rate | -1483.12 |
| High School Degree | 273.37 |
| Bachelor's Degree | 714.61 |
| Income Inequality | 125.35 |
| Private School Num | 202e13 |
| Public School Num | -190e13 |
| Hospital Num | -1181.94 |
| Hospital Rating | -61.97 |
| Migration | -159.64 |

Table 1: Coefficients of linear regression

# 3 Models

In this section, we will talk about how we fit different models to predict the house price. Since we are interested in predicting the house price in the future, we can only use the data available in the past (i.e. we cannot use the data collected from the same year/month to predict the house price in that year). However, it makes sense to use the history house price data in the prediction. Thus we added two columns of the ZHVIs from the previous month and next month for each observation. Our response variable then became the ZHVI in the next month, and the ZHVI in the current month and previous month can be used as predictors. To verify the out-of-sample accuracy of the models, we randomly split the dataset into train (80%) and test data (20%). We will show how we fit models with the train data, prevented overfitting through validation, and models' performance on the test data in the following subsections.

## 3.1 Linear Regression

Although the results in EDA suggested that linear models might not be suitable for our data, we still fit them as comparison to other models. For the linear regression and lasso regression coming later, we standardized the numerical features by the mean and standard deviation in the train data to avoid the influence of different units of different features.

We were concerned about outliers when fitting linear regression because that affects our choice of loss function. Figure 8 plots ZHVI in the current month and ZHVI in the next month. It is not surprising that their values are close

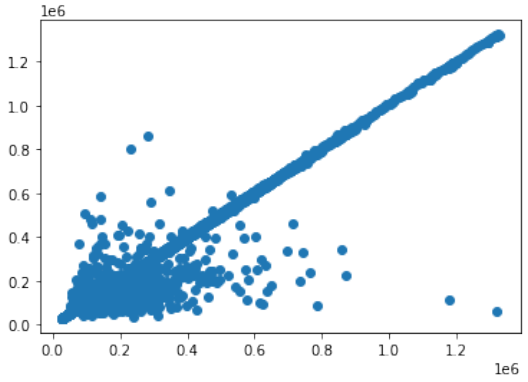


Figure 8: ZHVI vs ZHVI in the next month

which means that most of the time, there is only a slight difference in consecutive months. But we can also observe the existence of outliers that might appeared in some special regions or years. Since we are fitting a model for all regions and years, we want the model to take these outliers into consideration instead of being insensitive to them. Thus we chose to use quadratic loss for the linear and lasso regression. The coefficients of a sample of features are shown in table 1.

## 3.2 Lasso Regression

By observing the coefficients of the linear regression, we found that the weight of ZHVI of the previous two months are huge while the other coefficients seem to play a relatively minor role in the prediction. Therefore, it seems unnecessary to include all of the 640 features into the regression model. To select important features and prevent overfitting, we decided to add $\ell 1$ regularization to the linear regression. To tune the hyperparameter alpha, the penalty constant, we used k-fold cross validation with 5 folds. The optimized value of alpha is 12. Table 2 ranked the features by their absolute weights.

## 3.3 Boosting

Besides linear regression models, we also considered tree-based models because most features in the dataset do not show a linear relationship with the response variable. We first fit gradient descent boosting. The major hyperparameter that we are concerned with is the number of boosting stages. According to the scikit-learn documents, in most cases, the larger the value is, the better the performance will be because boosting seldom overfits the data. Thus we split a validation dataset and ran a few trial runs with different numbers. The result shows

| Variable | Coefficient |
|---|---|
| ZHVI | 19754.61 |
| Previous ZHVI | 90399.24 |
| State CA | 2520.88 |
| CPI | -2195.85 |
| Year | 1822.63 |
| State CO | 1037.83 |
| State MA | 1008.06 |
| State OR | 937.21 |
| State OK | -927.64 |
| Unemployment Rate | -806.27 |
| State WA | 749.87 |
| State NJ | 680.37 |
| Resident Rate | 661.07 |
| Month | -608.36 |
| Income | -567.29 |

Table 2: Coefficients of lasso regression

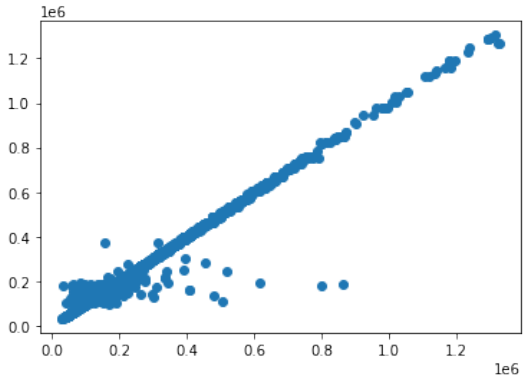


Figure 9: Boosting: true values vs. prediction

that although the increment in boosting stages will enhance the accuracy in the validation set, the difference is really small while the duration of the training process increases fast. Thus we chose to use the default value 100 which has a good enough accuracy.

## 3.4 Bagging

Next, we looked at using Bagging with Decision trees as estimators. The bagging algorithm uses bootstrap datasets from the training data to fit multiple tree models and ensemble the results to prevent overfitting. For tuning the hyperparameters, we used grid search with cross validation to explore different combinations. We primarily focused on tuning the number of decision trees used in the ensemble (tested values:10, 50, 100) and the size of the bootstrap sample(tested values: 0.05, 0.1, 0.5).

For the cross validation strategy, we chose to use Repeated K-fold, which has the benefit

```
Best: -78686716.910711 using {'max_samples': 0.5, 'n_estimators': 100}
-154276494.537587 (72127133.553199) with: {'max_samples': 0.05, 'n_estimators': 10}
-139770097.127740 (69528325.077937) with: {'max_samples': 0.05, 'n_estimators': 20}
-134897672.185902 (65710662.775867) with: {'max_samples': 0.05, 'n_estimators': 100}
-130683454.716426 (67879277.976282) with: {'max_samples': 0.1, 'n_estimators': 10}
-122639561.009455 (69105302.522163) with: {'max_samples': 0.1, 'n_estimators': 20}
-113764610.326420 (63682374.194556) with: {'max_samples': 0.1, 'n_estimators': 100}
-92071717.961235 (51853717.814928) with: {'max_samples': 0.5, 'n_estimators': 10}
-87557815.936916 (49217705.260336) with: {'max_samples': 0.5, 'n_estimators': 20}
-78686716.910711 (52207943.819873) with: {'max_samples': 0.5, 'n_estimators': 100}
```

Figure 10: Bagging: grid search results

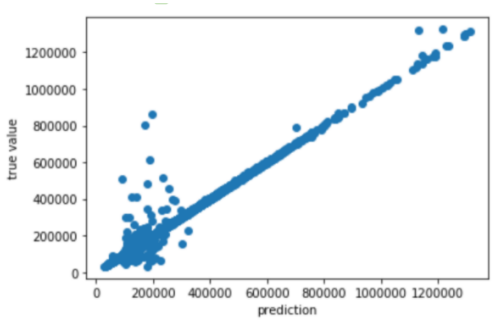


Figure 11: Bagging: true values vs. prediction

of improving the estimate of average model performance. This could inevitably introduce more model fitting and evaluations. MSE is used as a scoring strategy to evaluate the model performance during cross validation. Figure 10 shows the results for grid search.

Since we are using 'neg mean squared error' as our scores (this is by default of sklearn), we will choose the set of parameters that gives the lowest absolute value from the above graph, which is 0.5 and 100. Then we use these parameters to fit the model again. Figure 11 is true values plotted against the predicted values.

The prediction is quite accurate for house prices in the middle (400k to 1000k) The model does not work well to predict house prices around the lowest and the highest levels (obvious cluster on lower left).

## 3.5 Random Forest

The last model we fit is random forest (prediction result shown in figure 12). Similar to bagging, random forest uses bootstrap data to fit multiple trees and ensemble them together. The main difference is that in random forest, we can also select the number of features used for different bootstrap data-sets. Adopting the experience from boosting and bagging, we set the number of trees to be 100. Then the main hyper-parameters left to tune are the size of bootstrap samples (tested values: 0.05, 0.5, 1) and the number of features used for each sample (tested values: 'auto', 'sqrt'). After cross validation, the optimized values we got are 1 and 'auto'. 'Auto' makes use of the full feature

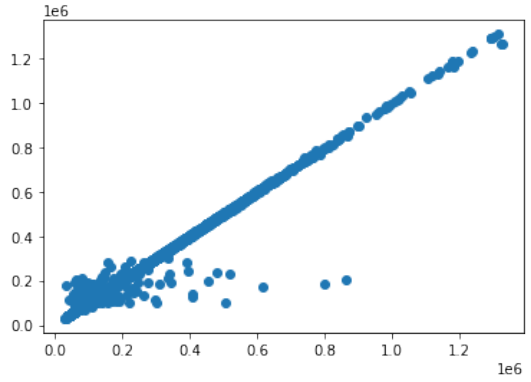


Figure 12: Random forest: true values vs. prediction

| | Linear regression | Lasso regression | Gradient descent boosting | Bagging | Random forest |
|---|---|---|---|---|---|
| Train data MAE | 2312.36 | 2391.27 | 1540.35 | 623.84 | 379.48 |
| Test data MAE | 2334.75 | 2378.76 | 1727.74 | 1123.03 | 1112.24 |
| Test data trend accuracy | 56.24% | 43.66% | 63.37% | 84.26% | 84.79% |

Figure 13: Model performance

space in each bootstrap tree. So it seems to imply that the optimal random forest model is basically the same as the bagging model.

# 4 Results

We will discuss model performance and important features in this section.

## 4.1 Model Performance

We applied all the models to the test data. We used mean absolute error as the measurement since it makes more intuitive sense for house price. Besides the quantitative error. We also care about how accurately the models can predict the increasing or decreasing trend of house price. This would be even more important than MAE in the scenario since by predicting the correct trend, we can at least make sure that someone who uses our model will not lose money when the market is going down or miss investment opportunity when the market is thriving.

Figure 13 shows the result in a table. When comparing across models, we can see that nonlinear models(tree-based models) have better performance on test dataset compared to linear models(ridge and lasso). The models with the best performance on test data are bagging and random forest. Actually their MAE and trend

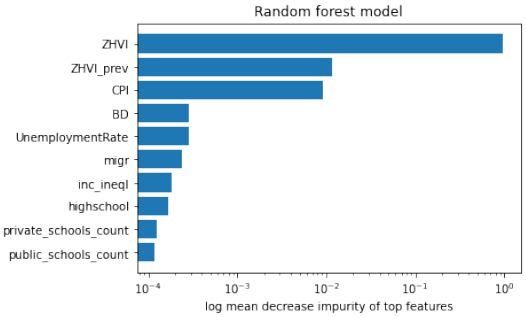


Figure 14: Feature Importance

accuracy are very close. As we explained earlier, this is because the optimal hyperparameter we got for random forest makes it basically the same as bagging. The result illusstrates that the models with higher complexity can better capture the non-linearity between features and the target variable.

However, although bagging and random forest have the best performance, they also overfit the data the most even after we tried to avoid it through cross-validation. Notice that the difference between train MAE and test MAE are big for bagging and random forest while quite small for linear regression and lasso regression.

## 4.2 Feature Importance

In order to provide a better interpretability of data, we calculated the feature importance (according to the decrease in node impurity weighted by the probability of reaching that node).

Figure 14 shows the top ten features with the highest importance scores. We logged the importance score in the graph for better visualization. This indicates that for predicting ZHVI for next month, features like current month ZHVI, previous month ZHVI, CPI, bachelor's degree percentage, unemployment rate are among the most important ones whereas other variables like county labels have minor influence on house price prediction. The feature importance calculated from random forest and bagging are very similar to each other, so we only included the figure for random forest.

Given that the importance scores have a large gap between CPI and bachelor's degree percentage, we tried using only the top three features to predict next-month ZHVI with random forest model. The MAE on the test data is 1154.35 and the trend accuracy is 84.37%

which are very close to the random forest with the whole feature space. This provides a strong evidence that only a few features in the data is significant for predicting house price.

# 5 Discussion

In this section we will discuss the effectiveness of the models and fairness consideration.

## 5.1 Effectiveness

We calculated the MAE and trend accuracy on test data. The MAE of our best model is around 1100, but we do not have any benchmark to compare to. The trend accuracy seems to be a more intuitive measurement to most people. The trend accuracy around 84% seems not bad as a byproduct of the regression model. However, if we want to apply it into real-world business use, it probably needs further improvement.

## 5.2 Fairness

Prediction models that serve for economic objective and involve demographic information might generate bias against certain regions or population groups. In our project, a biased model may attract investors to fund the house market in certain regions while ignoring other regions. Features that showed importance in the random forest model include bachelor's degree percentage and unemployment rate. These features might be correlated with race distribution in the country though they are not directly included in our dataset. Another feature subject to correlation with race distribution is county. However, county is not playing a significant role in the final model. The sparse model we fit in the last only used ZHVI from the previous two months and CPI which is constant nationwide. Thus, overall, it seems like our model will not play against fairness. Although history house price may reflects existing bias in the past, it seems inevitable in time series analysis.

# 6 Conclusions

In this project, we employed various machine learning techniques, including fitting linear and non-linear models, to predict next-month house prices by counties. We considered factors such as income levels, unemployment rate, number of schools and hospitals that cover both economic and society conditions. We also incorporate the effect of time as house price varies over time. The predictions of our project could assist investors to make decisions on buying and selling houses. The importance of features acquired from random forest model also provide information on what factors have the largest impact on house prices.

# 7 Future Work

Although we have 640 columns in our train matrix, we are taking into account categorical variables (county) after one-hot encoding. The actual number of features we are considering is 24. Moreover, by fitting different models, we found that lots of features we collected did not contribute to the prediction of future house price. Our models is subject to the risk of underfitting due to the lack of significant features. Thus, it is intuitive and reasonable to incorporate more indicators as predictive features. For example, supply and demand, GDP, and government laws can be used to add more information to our model construction. Exploring the interaction effect among different features is also worth trying. We expect better model performance with more data to train on.

We are also considering predicting house prices one year later by making predictions based on predictions, since real estate is often regarded as a long-term investment. One concern is that some features like CPI, which has a relatively large correlation with house price, varies over time. Therefore, we might want to fit a model to predict CPI first. This can be accomplished by sophisticated statistical models. Alternatively, since inflation trends tend to be persistent, one can use lags of CPI to forecast future values.