California Housing Price Forecasting

1. Motivation

Predicting housing price has always been a hot topic in data analytics field. During our undergraduate studies, we have witnessed a great change in housing prices in California. For instance, in San Diego where we spent 4 years studying, the housing prices have increased 50% from 2015 to 2019. Thus, we are pretty curious about if we can apply Machine Learning techniques on correctly forecasting sales price of houses. On the other hand, since many real estate businesses rely on deciding the appropriate prices to sell their houses, we believe our project can potentially help them in this regard. Moreover, investors who are interested in real estates and individual householder who wants to buy/sell house can benefit from our results as well.

2. Data Collection

All variables collected are monthly continues data for California state from 3 sources:

Zillow: https://www.zillow.com/research/data/

Name	Time Period	Description
Sale Price	2009/1 –2019/9	A seasonally adjusted median sale price for all housing types
ZHVI-All Home	2009/1 –2019/9	A seasonally adjusted median estimated home value for all home
Sale Count	2009/1 –2019/9	A seasonally adjusted monthly sale count number for all housing types
ZHVI-Condo	2009/1 –2019/9	A seasonally adjusted median estimated home value for condo
ZHVI-Single Family	2009/1 –2019/9	A seasonally adjusted median estimated home value for single family house
ZHVI-1Bedroom	2009/1 –2019/9	A seasonally adjusted median estimated home value for 1-bedroom home
ZHVI-2Bedroom	2009/1 –2019/9	A seasonally adjusted median estimated home value for 2-bedroom home
ZHVI-3Bedroom	2009/1 –2019/9	A seasonally adjusted median estimated home value for 3-bedroom home
ZHVI-4Bedroom	2009/1 –2019/9	A seasonally adjusted median estimated home value for 4-bedroom home
ZHVI-5Bedroom	2009/1 –2019/9	A seasonally adjusted median estimated home value for 5+bedroom home
Median home value per sq.ft	2009/1 –2019/9	A seasonally adjusted median estimated home value per square feet for all home
Buyer & Seller Index	2010/11 – 2019/9	A measure of the balance between sellers and buyers in a given market
Monthly For-Sale Inventory	2013/1 –2019/9	A seasonally adjusted for- sale listings data for all home
Zillow Rent Index-All Homes	2010/09-2019/9	A seasonally adjusted measure of the median estimated market rate rent for all home
Sale to List Ratio	2010/1 –2019/9	Median of the ratio between the sale price and the list price for all homes
Days on Zillow	2010/1 –2019/9	The median days on market of homes sold within a given month

Fred Economics Data:

https://fred.stlouisfed.org/categories/27286?cid=27286&et=&ob=pv&od=desc&pageID=4&t=monthly%3Bsa

Unemployment rate	2009/1 –2019/9	A seasonally adjusted monthly unemployment rate for California
Leading Index	2009/1 –2019/9	A seasonally adjusted leading index for California, which represents the economy situation of the state

Macro Trend: https://www.macrotrends.net/2604/30-year-fixed-mortgage-rate-chart

Note: Zillow provides the data of each feature for all states in one file. Thus, in order to get above data for California, we extracted the features from 16 individual files.

30 Years Fixed Mortgage Rate	2009/1 –2019/9	A monthly 30 years fixed mortgage Rate

3. Data Prepossessing

3.1. Missing Data within Time Period

Data of Zillow Rent Index in 2014/02 is missing. From the graph (a) in Appendix, the missing data is in a decreasing trend. Since the data is seasonally adjusted and we want to get a smooth curve, we decided to predict the value by averaging the values of its previous and next month. The result curve is in graph (b).

3.2. Large increase in the Sale Price

From the graph (c), the Sale Price increase fast through time. Also, the growth of sale price is meaningful. So, we applied log transformation to sale price and our dependent variable becomes log(Sale Price).

3.3. Higher degree independent variables

In order to increase the predictive power of the dataset, we added eight high degree independent variables by creating perfect power of existing numerical features. We chose ZHIV-All home and Median home value per square feet to create high degree variables. For each of them, we generated 4 high degree variables by taking it to the power of 2, 3, 4, 5 respectively.

3.4. Time Series Data Structure

To determine the predictive power of past values towards dependent variable, we generated the PACF graph to find out the correlation between the past and current values using training data. From the graph (d), the lag 1 value has a statistically significant impact and lag 2 value's impact is not significant but still obvious. So, in order to use this predictive power in other models, we added two new features SPLM(log(Sale Price) with lag 1) and SPL2M(log(Sale Price) with lag 2) to capture the predictive powers of these past values.

3.5. Different Time Period of Features

From the above, there are 7 features - 2 features are past values of dependent variables and last 5 features from Zillow - whose start date is not 2009/01. If we keep all the features, we need to give up 36 observations and start our data from 2013/01. If we decide to keep almost all the observations, we need to sacrifice 5 features from Zillow. The price of keeping 2 past values is just deleting 2 earliest observations, which is pretty cheap. In this stage, we didn't know which method can give us better predictions. Thus, we decided to separate our dataset into two subsets dataA & dataB: dataA contains all the observations and dataB contains all the features. We proceed our works with these two data sets separately.

3.6. Split Train and Test Dataset

Since this is a time series dataset, we cannot randomly split the dataset. In fact, we splitted using year. We considered all the observations earlier than 2018 as training set, and all other observations as testing test. Then, the test set for both subsets has 21 observations.

3.7. Different Range of Features

Since the features in the data set have different ranges, it is necessary to do normalization before applying machine learning algorithms. We normalized the training data and used the mean and standard deviation of training data to normalize testing data.

After data prepossessing, our main dataset has 127 observations and 30 variables including one variable for the date. Indeed, dataA has 127 observations and 25 variables, and dataB has 81 observations and 30 variables.

4. Model & Result

Since our dependent variable is continuous and our goal is prediction, we have applied 9 models: Basic Linear Model, Time Series Model, Random Forest, Boosting, Principal Component Regression, Ridge, Lasso, Forward Stepwise Regression, and Blending Model to both datasets. The following are the details of analysis for each dataset.

4.1 DataA Model Analysis

1. Basic Linear Model

As the graph (c) shows, Sales price has a linear relation. Therefore we firstly applied the Basic Linear Regression model, selecting all the other variables against the dependent variable Sales.price with logarithm. From the model, the most significant variables ($p <= 0.001$) are SPLM, SPL2M, unemployment rate, ZHIV-3Bedrooms, and ZHIV-Condo. The following is the performance on the test set.

OSR^2	MAE	RMSE
0.9808901	0.04905144	0.05573186

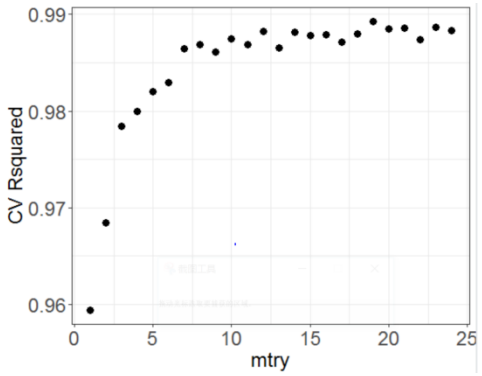
2. Time Series Model

Since it is time series data, we applied the ARIMA model. We use the auto.arima function to pick up the best parameter. From the result from auto.arima, we choose parameter lag term = 1, degree of differencing = 1, and moving average = 0. That is, the model differenced the dependent value once to make it stationary, and used the lag 1 term to predict, which shows the predictive power of SPLM and SPL2M. The following is the performance on the test set.

OSR^2	MAE	RMSE
0.9970986	0.01912432	0.02171598

3. Random Forest and Boosting

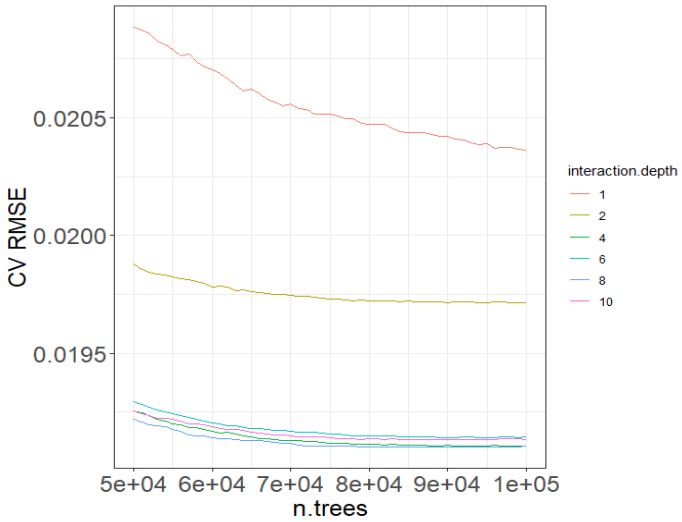
Next we considered to apply more sophisticated model with other variables as well as time series features. We firstly applied Random Forest model based on two lag terms features and all the other variables. Use 5-fold cross-validation to pick up the appropriate parameter mtry and set other parameters with default value.



We selected mtry=19 to give us highest R^2 in terms of the plot above. The result of Random Forest model is as follows.

OSR^2	MAE	RMSE
0.9699813	0.06791442	0.06985057

Then we move to the Boosting model. With a similar idea, we applied 5-fold cross-validation to select parameter n.trees and interaction.depth, set shrinkage = 0.002 and n.minobsinnode = 10.



We built final model with 92000 iterations and interaction.depth=8 to in order to lower error in terms of the plot above. Then, we get the predictive quality results.

OSR^2	MAE	RMSE
0.8735047	0.141986	0.1433874

Compared with previous models, the results from both Random Forest and Boosting are not so ideal. One reason is that these two models are not good at predicting the value out of the range of trainset. Following two graphs visualize the predictions of two models. The red curve is the original data and the black one is the prediction from the model. The left graph is Random Forest and left one is Boosting. We can see that the prediction is within the range of trainset, but the real data is increasing through time.

4. Random Forest and Boosting using Residue

As we mentioned before, decision trees are not good at predicting extreme data, so a good practice is replacing the current dependent variable with the residue of ARIMA model which is stationary. Then, we used the same models to find the relation between new dependent variable and feature without two past values because we have used them in the ARIMA model.

For Random Forest, the mtry for the final model is 10. The following is the performance on the test set.

OSR^2	MAE	RMSE
0.9913779	0.03014456	0.0374352

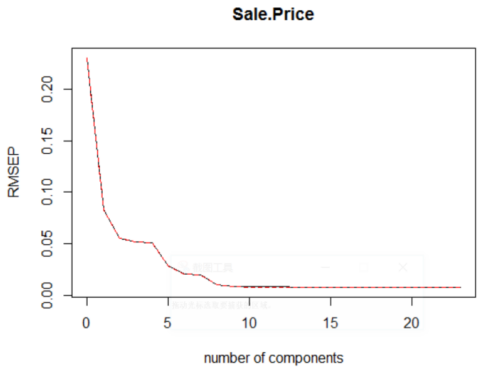
For Boosting, the n.trees = 20 , and the interaction.depth = 2 in the final model. Then, the result is as following.

OSR^2	MAE	RMSE
0.9920387	0.02908872	0.03597201

Compared with previous results, the current models have improved a lot. However, if we compared with the performance of ARIMA, the performance of these two models decreased. That is, they did not generate any useful information from residues.

5. PCR

Since the ratio between the number of features and observations is relatively large, we consider to apply models which can keep the important information and avoid overfitting. We applied PCR model with cross-validation to select the number of PCs as follows.



We can see that ncomp=10 is a reasonable choice which means that we keep the most

important 10 components in the model. Train the PCR model and get the performance on test set.

OSR^2	MAE	RMSE
0.999659	0.00585739	0.007444726

6. Ridge Regression

Consider adding a penalty term on the loss function to shrinkage variable selection. We applied Ridge regression on the data set. In order to select the rational regularization parameter, we conduct cross-validation. We used cv.glmnet method with alpha = 0. The lambda gave us minimum error is 0.02281001. The performance on test set is as follows.

OSR^2	MAE	RMSE
0.989584	0.0388927	0.04114577

7. Lasso Regression

We repeated the steps above with Lasso shrinkage penalty. We used cv.glmnet method with alpha = 1. The lambda gave us minimum error is 0.001034406. The performance on test set is as follows. The model with 5 non-zero coefficients: Sale Count, Unemployment Rate, Leading Index, 30 Years Fixed Mortgage Rate, and SPLM. The performance is as follows.

OSR^2	MAE	RMSE
0.9996831	0.005727009	0.007177147

8. Forward Stepwise

Finally, we applied Forward Stepwise model. At each step, the model tested each variable for addition until all variables are considered so that to build the best model. There are four features in the final model: SPLM, SPL2M, Unemployment Rate, and Leading Index. The performance on test set is as follows.

OSR^2	MAE	RMSE
0.9998046	0.00411799	0.005636024

9. Blending

Now we consider blending well-performed models above. Since Basic Linear, Ridge, Lasso, Forward Stepwise Regressions are all linear models, we just choose Forward Stepwise to build the blending. Since ARIMA has combined with RF on Residue and Boosting on Residue, we will not use it in blending model. Since Random Forest and Boosting cannot predict out of range value, we will also remove them from blending features. Then, we will use Forward Stepwise, PCR, RF on Residue, Boosting on Residue to build blending model. The final model shows that the significant predictions are RF on Residue and Boosting on Residue, which are out of our expectation. The performance on the testset is following.

```
lm(formula = logSP ~ . - 1, data = bl_train)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0067846 -0.0020459 -0.0003466  0.0020547  0.0144149

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
be            -0.65692    0.08484  -7.743 7.32e-12 ***
logSP.10.comps  0.00157    0.10414   0.015    0.988
re             1.53922    0.08926  17.244  < 2e-16 ***
fd             0.11624    0.09934   1.170    0.245
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.003424 on 102 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 3.643e+08 on 4 and 102 DF,  p-value: < 2.2e-16
```

OSR^2	MAE	RMSE
0.9984918	0.01565688	0.01313898

4.2 DataA Result

The best model of DataA is Forward Stepwise model with the highest OSR^2 and lowest error. The featured that most frequently used in the models are SPLM, SPL2M, Unemployment Rate, and Leading Index.

	OSR^2	MAE	RMSE
Linear Regression	0.9808901	0.04905144	0.05573186
ARIMA	0.9970986	0.01912432	0.02171598
Random Forest	0.9699813	0.06791442	0.06985057
Boosting	0.871523	0.1432571	0.1445062
RF on Residue	0.9913779	0.03014456	0.0374352
Boosting on Residue	0.9920387	0.02908872	0.03597201
PCR	0.999659	0.00585739	0.00744473
Ridge	0.989584	0.0388927	0.04114577
Lasso	0.9996831	0.00572701	0.00717715
Forward Stepwise	0.9998046	0.00411799	0.00563602
Blending	0.9984918	0.01565688	0.01313898

4.3 DataB Model Analysis

Since we applied the same process to analyze the dataB, we will state process concisely and focus more on the difference between the results from A and B.

1. Basic Linear Regression

The R^2 for the model is 0.9994, but the OSR^2 is 0.05903775. This means that the model is overfitting, so the model did not work well in the dataB. The summary of model is in graph (e). The following is the performance on the test set.

OSR^2	MAE	RMSE
0.05903775	0.1967228	0.2108252

2. Time Series Model

The result of auto.arima is (0,1,2), which means that past values have no predictive power to current values. However, in linear model before, SPLM and SPL2M are statistically significant, which means the lag term should not be 0. Therefore, we manually test several arima parameters and finally choose arima(1,1,1) which gave the highest OSR^2. It means the model used the lag 1 term of past value and prediction error to predict differenced dependent value. The following is the performance on the test set.

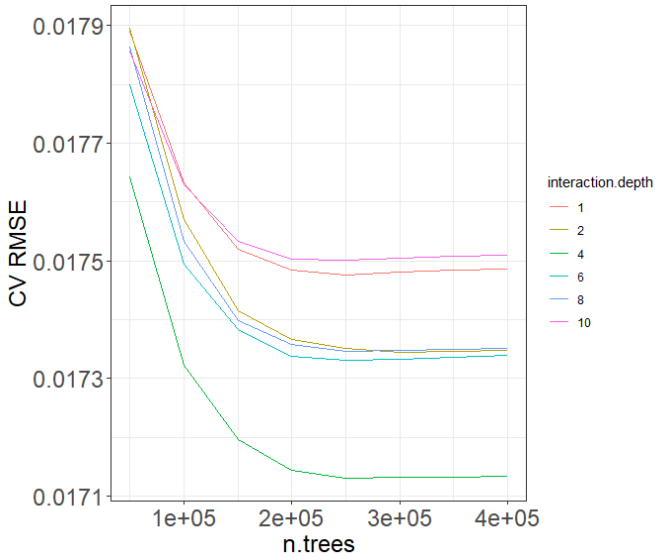
OSR^2	MAE	RMSE
0.9937797	0.01462164	0.01714129

3. Random Forest and Boosting

For Random Forest, we used 5-fold cross-validation to pick up mtry from 1 to 28 . Based on the graph (f), the final model used mtry=12, which is smaller than the one in dataA. The result of Random Forest model is as follows.

OSR^2	MAE	RMSE
0.8709654	0.07580276	0.07807095

For Boosting model, with a similar idea, we applied 5-fold cross-validation to select parameter n.trees and interaction.depth, set shrinkage = 0.002 and n.minobsinnode = 10.



Based on the plot above, we built the final model with n.trees = 2*10^5 and interaction.depth=4 to lower RMSE . Then, we get the predictive quality results.

OSR^2	MAE	RMSE
0.8058443	0.09283113	0.09576605

Compared to dataA, both Random Forest and Boosting performed worse in dataB. Compared with previous models, the results from both Random Forest and Boosting are not so ideal for the same reason as dataA. Therefore, we run RF and Boosting on residue of ARIMA model in the next step.

4. RF and Boosting on Residue

For Random Forest, the mtry for the final model is 16. The following is the performance on the test set.

OSR^2	MAE	RMSE
0.9948073	0.01290733	0.01566148

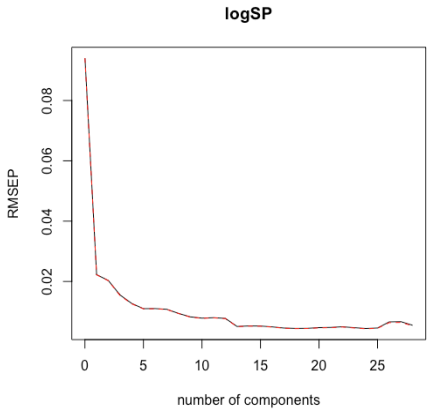
For Boosting, the n.trees = 2 , and the interaction.depth = 1 in the final model. Then, the result is as following.

OSR^2	MAE	RMSE
0.9947495	0.01296334	0.01574838

Compared with previous results, the current models have improved a lot. In dataB, after we run RF and Boosting on residue, the model's performance is slightly better than ARIMA model which is different to dataA's model performance.

5. PCR

We applied PCR model with cross-validation to select the number of PCs as follows.



Based on the graph above, we can see that ncomp=13 is a reasonable choice which means that we keep the most important 13 components in the model. Train the PCR model and get the performance on test set.

OSR^2	MAE	RMSE
0.9849262	0.02580919	0.0266838

6. Ridge Regression

We applied Ridge regression on the data set. We used cv.glmnet method with alpha = 0. The lambda gave us minimum error is 0.009223. The performance on test set is as follows. In the following graph, the red line represents the original data and the blakc line represents

our prediction.

OSR^2	MAE	RMSE
0.995873	0.01255816	0.01396218

7. Lasso Regression

We repeated the steps above but with the Lasso shrinkage penalty. We used cv.glmnet method with alpha = 1. The lambda gave us minimum error is 0.000137. The performance on test set is as follows. There are 10 non-zero coefficients, which is much larger than the one in dataA. Compared with dataA, 30 Years Fixed Mortgage Rate is dropped from the model, and ZHIV^5 and ZHIV-5bedrooms becomes important. For the 5 new features, only the Zillow Rent Index is dropped and all other 4 features are used in the model. That is , from Lasso's result, the new features have predictive power to dependent variable.The performance on test set is as follows.

OSR^2	MAE	RMSE
0.9989489	0.005480174	0.007046097

8. Forward Stepwise

Finally, we applied Forward Stepwise model. There are eight features in the final model: SPLM, SPL2M, ZHVI-Condo, ZHVI-2 bedroom, ZHVI-All home, ZHVI-1 bedroom, ZHVI-3 bedroom, and Leading Index. Compared with dataA, it has 4 more features. Also, 5 new features are not chosen by model. The performance on test set is as follows.

OSR^2	MAE	RMSE
0.9953598	0.01043063	0.01480484

9. Blending

Following similar idea, we chose RF on Residue, Boosting on Residue, PCR, and Lasso to build blending model. From the graph (g), the final model shows that the significant predictions are RF on Residue and Boosting on Residue, which is same with dataA. The performance on the testset is following.

OSR^2	MAE	RMSE
0.9958187	0.01085869	0.01405369

4.4 DataB Result

The best model of DataB is Lasso model with the highest OSR^2 and lowest error. The models used relatively different variables to do predictions, compared with dataA. Only the SPLM and SPL2M showed the significant power in all models. Based on the Lasso, the new features also helped to predict.

	OSR^2	MAE	RMSE
Linear Regression	0.05903775	0.1967228	0.2108252
ARIMA	0.9937797	0.01462164	0.01714129
Random Forest	0.8709654	0.07580276	0.07807095

Boosting	0.8058443	0.09283113	0.09576605
RF on Residue	0.9948073	0.01290733	0.01566148
Boosting on Residue	0.9947495	0.01296334	0.01574838
PCR	0.9849262	0.02580919	0.0266838
Ridge	0.995873	0.01255816	0.01396218
Lasso	0.9989489	0.005480174	0.007046097
Forward Stepwise	0.9953598	0.01043063	0.01480484
Blending	0.9958187	0.01085869	0.01405369

4.5 Final Model & Conclusion

The best model for our dataset is forward stepwise from data. The OSR^2 is 0.9998046, the RMSE is 0.005636024, and MAE is 0.00411799. Back to our dependent variable, the absolute average error is e^0.00411799 percentage. The following graph shows its prediction to dependent variable.

We used the bootstrap to assess the performance of the final model on test set. The bias, std. error, and 95% of confidence interval for each metric is as follows:

RMSE: -7.817053e-05, 1.074243e-03, (0.0036, 0.0078)

MAE: 1.841090e-05, 8.352736e-04, (0.0024, 0.0056)

OSR^2: -9.891208e-07, 7.234604e-05, (0.0024, 0.0056)

In general, we are pretty confident about our results. From the bootstrap results, the biases are nearly zero. From the data collection aspects, we have gathered accurate and recent data from three different sources; we have also included economic data such as unemployment rata and leading index. From the model selection aspects, we have tested 9 different models with parameter tunings; we compared different metrics such as RMSE, OSR^2 and MAE.

However, there's one missing data in the original source. We averaged the data before and after that data point in order to predict that missing value, which could be different from what the data actually is. Although we tried to collect a large set of data, our number of observations is relatively small. If we could collect more data, our model will have more predicting power.

Also, we could include more features to show the impact exerted by the government policies on housing prices. In terms of models, we have so far included 9 machine learning models. We could further add deep learning techniques such as recurrent neural network.

In terms of comparing results for dataA and dataB, we have found that our model performs better on test set when trained on dataA. Since dataA has 40 more observations than dataB, dataB has 5 more features than data, we can safely conclude that the predicting power of observations is greater than features'. Indeed, this is a time series data set, naturally past data has significant predicting power on our test set. On the other hand, only one model considered these 5 features as significant, which means that these features didn't have obvious predictive power.

5. Impact

- Real estate businesses could adjust their strategies based on our results in order to maximize profits.
- Our model could help investors and householders who consider to purchase/sell homes to seize the right time to get the best deals in the housing market.
- From the past experience, the sale price of housing is a good indicator of inflation rate, thus our results could help people to understand the economic situation in the near future.
- What's more, we can further scale up this model to nationwide data instead of California only so that the results can benefit more people.
- Our model and forecasting techniques used in this data set could be applied to other time series data with similar characteristics as well.

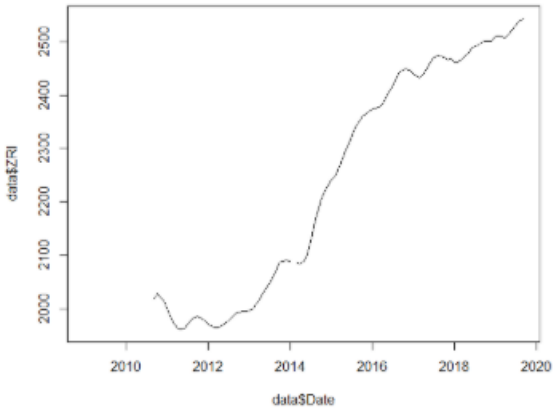
6. Appendix

Code on google drive:
https://drive.google.com/file/d/1rYEm2z2VUWgo35mqv6gJzcHLPZT5xZof/view?usp=sharing

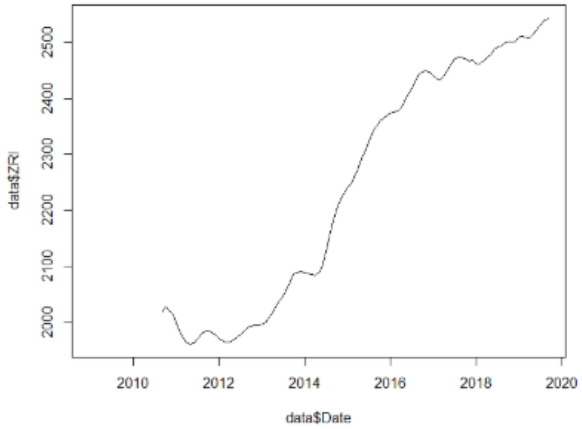
Data file on google drive:
https://drive.google.com/file/d/1dnMO2fJK-6MJ5GCcRFXHuJJxqxzQWYD2/view?usp=sharing

Load the data into R and install packages as written in the code, set seed at 242, then the results can be reproduced.

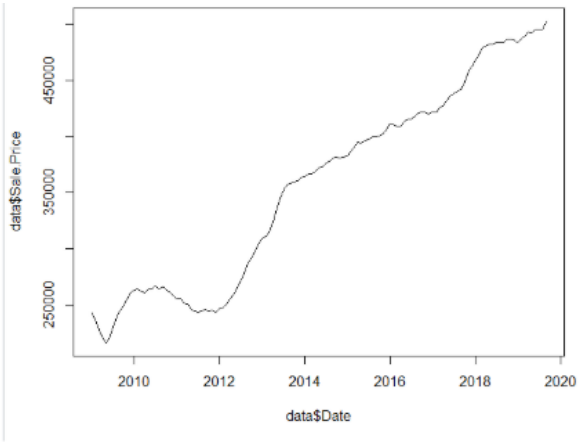
Graphs:



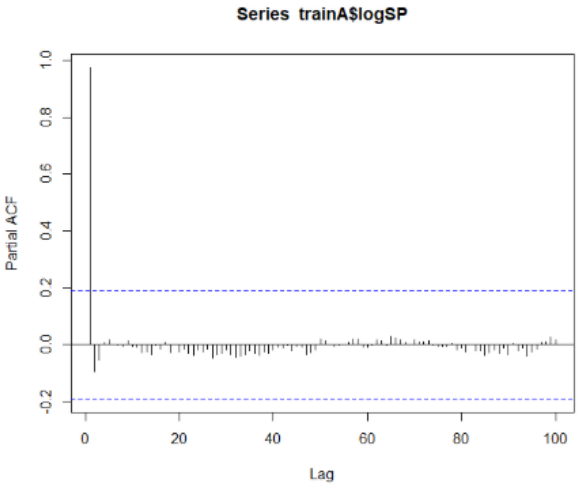
a



b



c



d

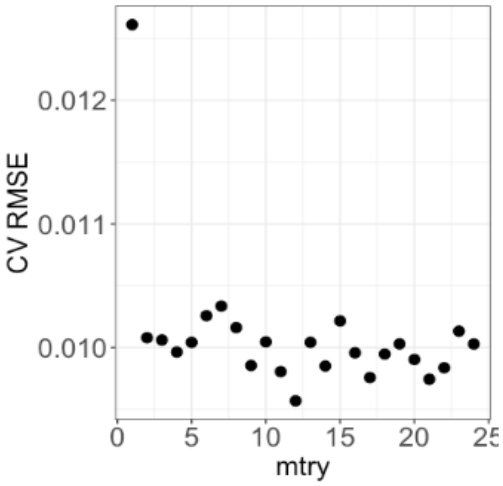
```
lm(formula = logSP ~ ., data = trainB)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0052310 -0.0015699 -0.0001036  0.0011863  0.0079517

Coefficients:
                Estimate Std. Error   t value Pr(>|t|)
(Intercept)    1.288e+01  4.094e-04 31451.458  < 2e-16 ***
ZHVI.All.home  4.876e+02  2.773e+02     1.758  0.08856 .
ZHVI2         -2.106e+03  1.135e+03    -1.856  0.07294 .
ZHVI3          3.422e+03  1.754e+03     1.951  0.06010 .
ZHIV4         -2.479e+03  1.213e+03    -2.045  0.04948 *
ZHIV5          6.754e+02  3.164e+02     2.134  0.04083 *
Sale.Count     1.814e-04  1.639e-03     0.111  0.91261
ZHVI.Condo    -8.403e-02  1.687e-01    -0.498  0.62187
ZHVI.SF        4.934e-01  2.351e-01     2.099  0.04404 *
ZHIV.1B        7.459e-02  9.489e-02     0.786  0.43781
ZHIV.2B       -2.688e-02  1.639e-01    -0.164  0.87077
ZHIV.3B        6.174e-02  1.690e-01     0.365  0.71737
ZHIV.4B        4.968e-02  2.171e-01     0.229  0.82051
ZHIV.5B       -2.874e-02  1.165e-01    -0.247  0.80673
MHVPS         -5.074e+02  2.826e+02    -1.795  0.08237 .
MHVPS2         2.176e+03  1.159e+03     1.878  0.06980 .
MHVPS3        -3.511e+03  1.794e+03    -1.957  0.05944 .
MHVPS4         2.525e+03  1.244e+03     2.030  0.05100 .
MHVPS5        -6.821e+02  3.252e+02    -2.098  0.04419 *
BSI            2.172e-04  2.174e-03     0.100  0.92108
MSInventory   -4.116e-03  3.167e-03    -1.300  0.20334
ZRI            2.324e-02  1.783e-02     1.304  0.20189
SLR            4.341e-03  2.236e-03     1.941  0.06138 .
Days           6.962e-04  1.541e-03     0.452  0.65464
Unemployment  -3.552e-02  3.905e-02    -0.910  0.37007
LI            -2.546e-03  1.360e-03    -1.873  0.07056 .
FMR30          1.131e-03  9.954e-04     1.136  0.26446
SPLM           4.905e-02  1.770e-02     2.771  0.00936 **
SPL2M         -3.670e-02  1.604e-02    -2.288  0.02909 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.003171 on 31 degrees of freedom
Multiple R-squared:  0.9994,    Adjusted R-squared:  0.9988
F-statistic:  1822 on 28 and 31 DF,  p-value: < 2.2e-16
```

e



f

```
lm(formula = logSP ~ . - 1, data = bl_train)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0079375 -0.0008871  0.0000745  0.0012841  0.0039750

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
be             -0.57035    0.07070  -8.067 5.99e-11 ***
logSP.13.comps -0.12556    0.12262  -1.024    0.310
re              1.64625    0.11945  13.781  < 2e-16 ***
X1              0.04967    0.07267   0.684    0.497
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00198 on 56 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 6.344e+08 on 4 and 56 DF,  p-value: < 2.2e-16
```

g