Time Series Final Project Write-up

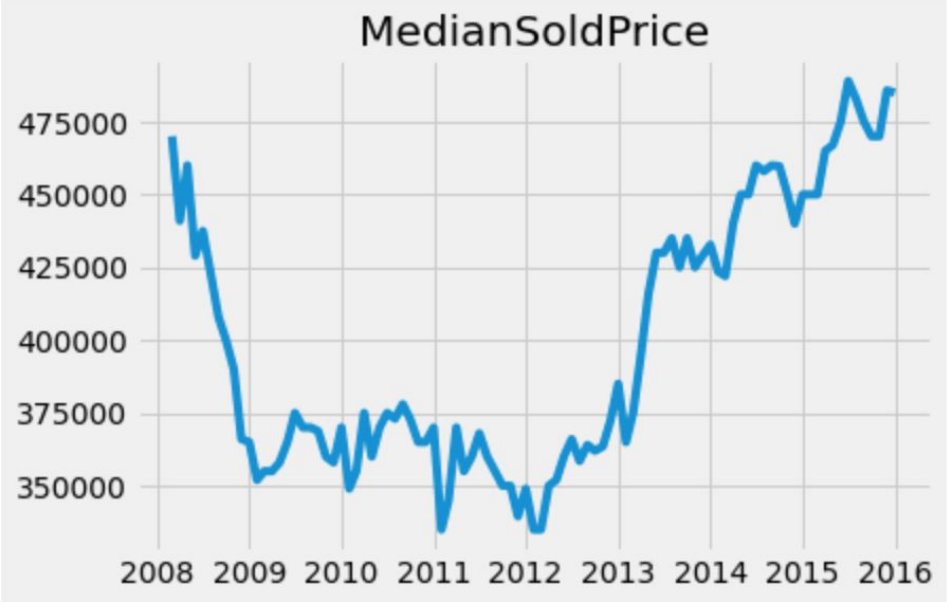
Sophie Wang, Anni Liu

Project description:

The Zillow dataset recorded Feb 2008 - Dec 2015 monthly median sold price for housing in California, Feb 2018 - Dec 2016 monthly median mortgage rate and Feb 2008 - Dec 2016 monthly unemployment rate. The goal is to predict the monthly median sold price for Jan-Dec 2016.

Features in the datasets are:Date,MedianSoldPrice,MedianMortageRate,UnemploymentRate

First, we need to take a look at the variable that we need to predict. There is a down trend at the beginning of all history data from 2008 to 2009, and then an upward trend begins from year 2012.



We first split Feb 2008- Dec 2015 data into training set (Feb 2008 - Dec 2014) and validation set (Jan 2015 - Dec 2015) and fit the models below on the training set and compute the metric (RMSE and MAPE) on the validation set.

Modeling methods:

1.Univariate models:

a.ARIMA
b.SARIMA
c.ETS
d.Prophet

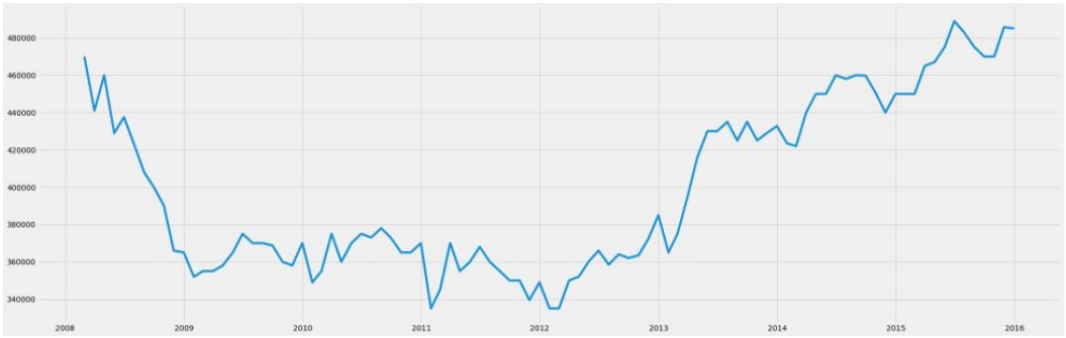
2.Multivariate models:

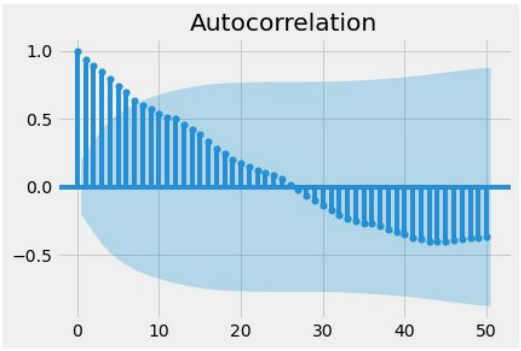
a.VAR
b.SARIMAX
c.LSTM

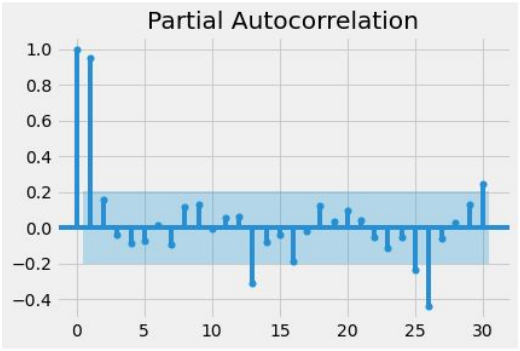
Univariate models:

1. **ARIMA**

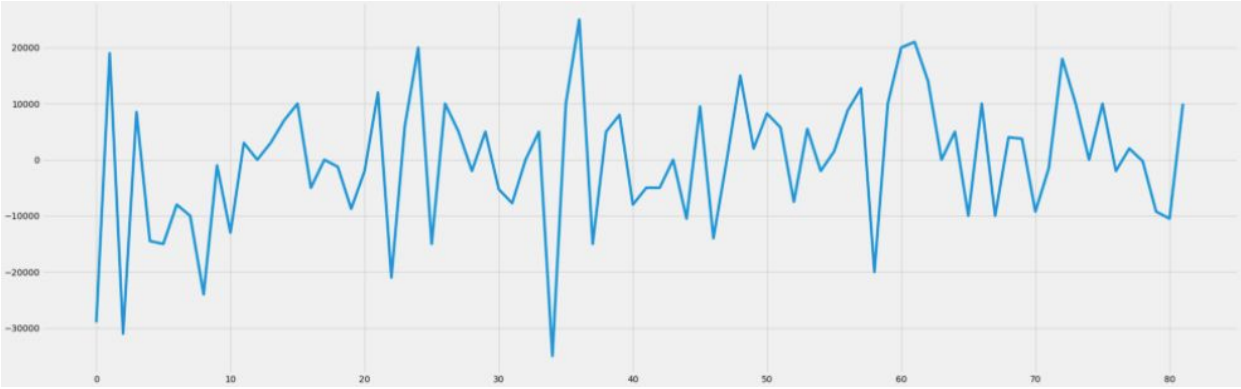
   For (S)ARIMA family models, first thing to do is to plot the time series plot, ACF, PACF plots as well as ADF test to check whether the time series data is stationary or not:

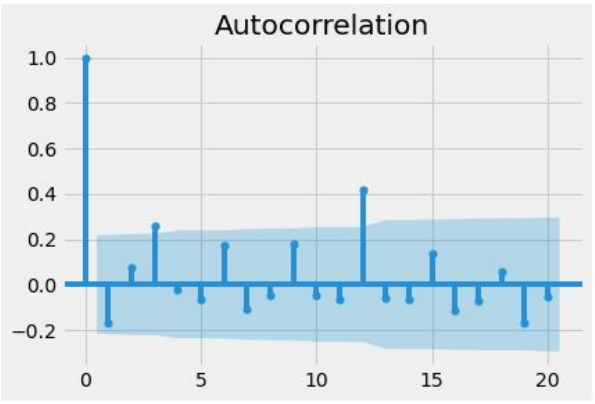


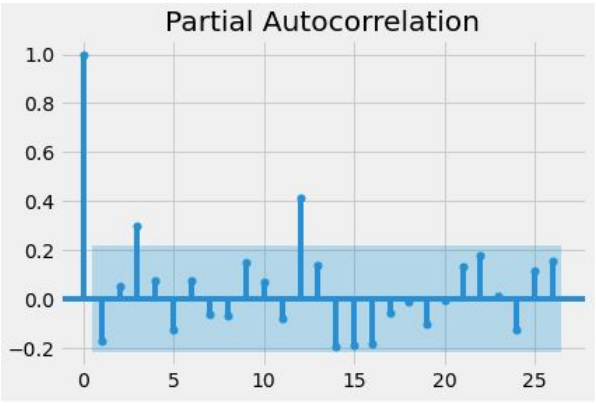




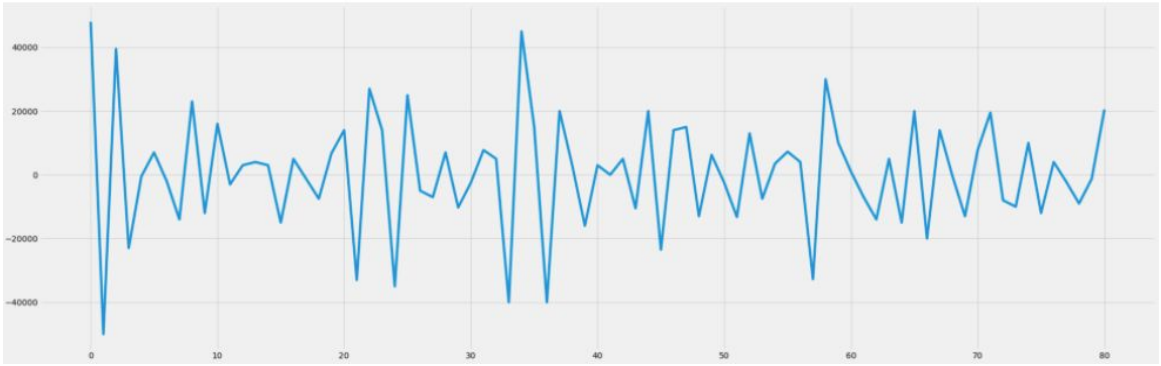
From the plots above, the data does not seem stationary. The p-value from the ADF test on the training set is 0.95, which confirms that it's not stationary. So we started differencing the trend once and plotted the same three plots:

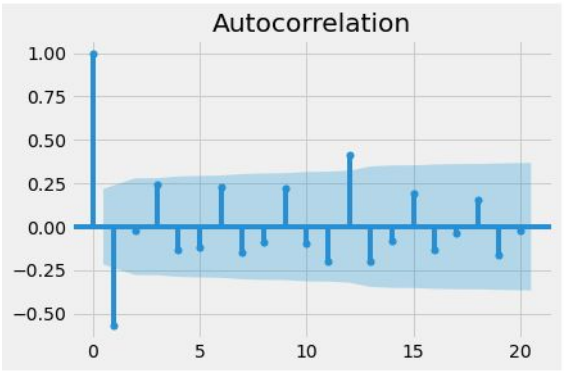


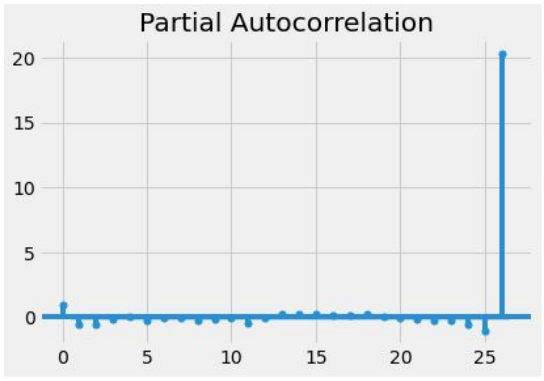




The data looks stationary from the plots now, however the ADF test shows p-value being 0.05, which is on the threshold. So to be more confident, we differenced the trend once more and got the following plots:







Now p-value from the ADF test is 8.499491e-09, which also confirms data being stationary. So we know that parameter d in ARIMA model is 2 and we searched for p and q parameter based on BIC criteria and got our first model: ARIMA(1,2,1)

2. SARIMA

From the original time series plot on the training data, we cannot tell for sure whether seasonality exists, so we also fit a SARIMA model which includes seasonality

components and then check performance on the validation set. By using auto search for SARIMA orders, we got the SARIMA(1,1,0),(0,1,0,10).

```
# Try fit a SARIMA model which include seasonality

model=auto_arima(history, # gdp as endogenous
                 start_p=0, start_q=0,
                 max_p=4, max_q=4,
                 max_d=2,
                 m=10, D=1, max_P=3, max_Q=3,
                 trace=True,
                 error_action='ignore',
                 suppress_warnings=True,information_criterion='oob', n_jobs = -1)
```

```
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(1,1,1)[10]             : AIC=inf, Time=0.15 sec
 ARIMA(0,1,0)(0,1,0)[10]             : AIC=1607.290, Time=0.01 sec
 ARIMA(1,1,0)(1,1,0)[10]             : AIC=1607.486, Time=0.04 sec
 ARIMA(0,1,1)(0,1,1)[10]             : AIC=1607.358, Time=0.06 sec
 ARIMA(0,1,0)(1,1,0)[10]             : AIC=1610.232, Time=0.03 sec
 ARIMA(0,1,0)(0,1,1)[10]             : AIC=1611.033, Time=0.04 sec
 ARIMA(1,1,0)(0,1,0)[10]             : AIC=1606.424, Time=0.02 sec
 ARIMA(1,1,0)(0,1,1)[10]             : AIC=1607.311, Time=0.13 sec
 ARIMA(1,1,0)(1,1,1)[10]             : AIC=inf, Time=0.18 sec
 ARIMA(2,1,0)(0,1,0)[10]             : AIC=1608.458, Time=0.02 sec
 ARIMA(1,1,1)(0,1,0)[10]             : AIC=1607.639, Time=0.08 sec
 ARIMA(0,1,1)(0,1,0)[10]             : AIC=1606.479, Time=0.02 sec
 ARIMA(2,1,1)(0,1,0)[10]             : AIC=1610.816, Time=0.13 sec
 ARIMA(1,1,0)(0,1,0)[10] intercept   : AIC=1607.782, Time=0.02 sec

Best model:  ARIMA(1,1,0)(0,1,0)[10]
Total fit time: 0.939 seconds
```

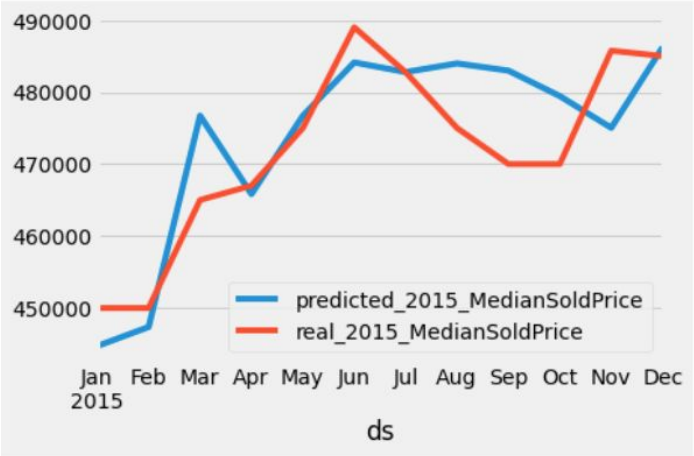
3. **ETS**

   The next univariate model we fitted is exponential smoothing ETS. Since from the time series plot, we don't have a linear trend and no obviously consistent seasonality pattern. We tried one ETS model with trend parameter being "multiplicative", seasonal parameter being 'multiplicative" , period =10 and damped = True.

4. **Prophet**

   We also fitted the Facebook Prophet model using default parameters. Since we have monthly data, we set freq = 'm'. Below the plot illustrating model performance on training set (before Dec 2014 inclusive) and validation set (after Jan 2015):

A closer look at 2015 predicted data and real validation set below (looks pretty good!):



Multivariate models:

1. **VAR**

   The chart below is a correlation between all the variables: We can fit a VAR model If we want to model everything together and assume every variable is impacting each other. We want to see how other variables can affect MedianSoldPrice, so we fit three VAR models, one for VAR-all the variables, one for VAR-MedianSoldPrice and MedianMortageRate, and one for VAR-MedianSoldPrice and UnemploymentRate.

   Best VAR

   rmse = 43561, mape = 0.075

Correlation matrix of residuals

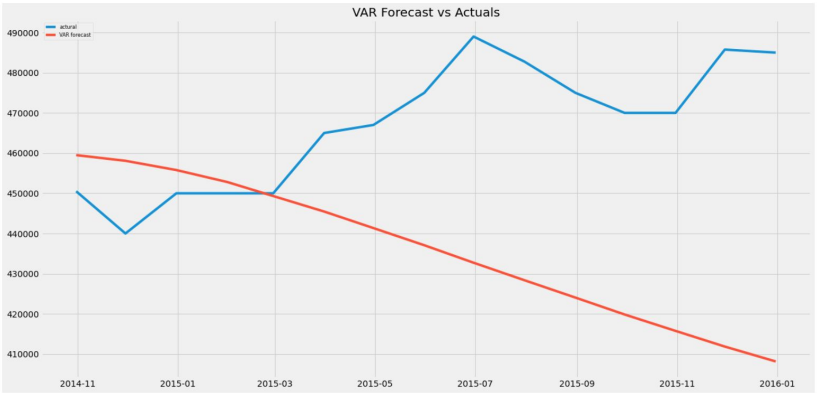
	MedianSoldPrice	MedianMortageRate	UnemploymentRate
MedianSoldPrice	1.000000	-0.130223	0.231505
MedianMortageRate	-0.130223	1.000000	0.002340
UnemploymentRate	0.231505	0.002340	1.000000

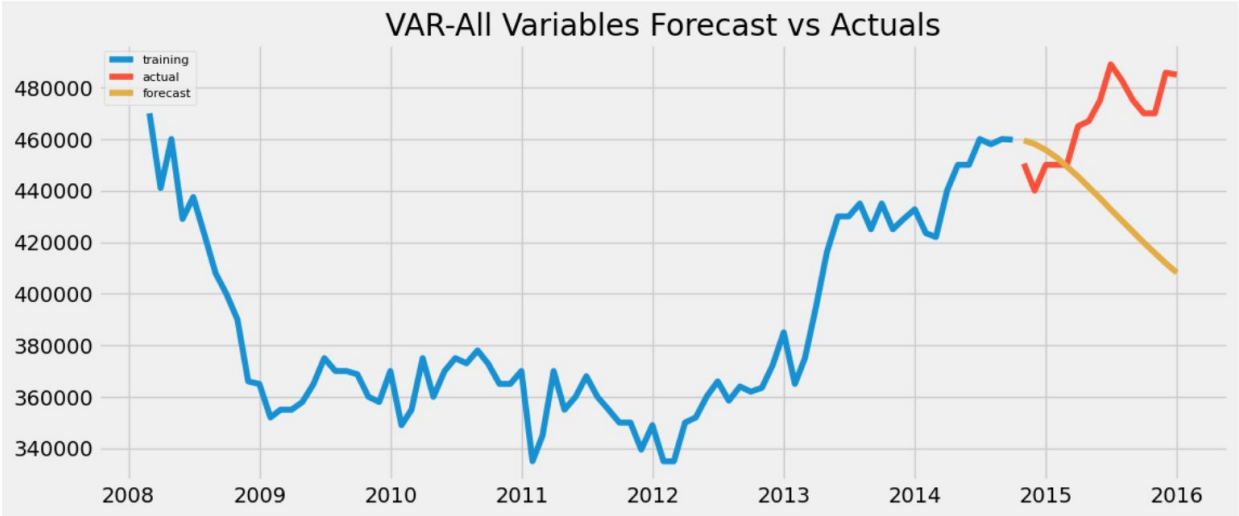
This is the forecasting plot for all the variables in the VAR model.

a. VAR-MedianSoldPrice,MedianMortageRate,UnemploymentRate

rmse = 43561

mape = 0.075

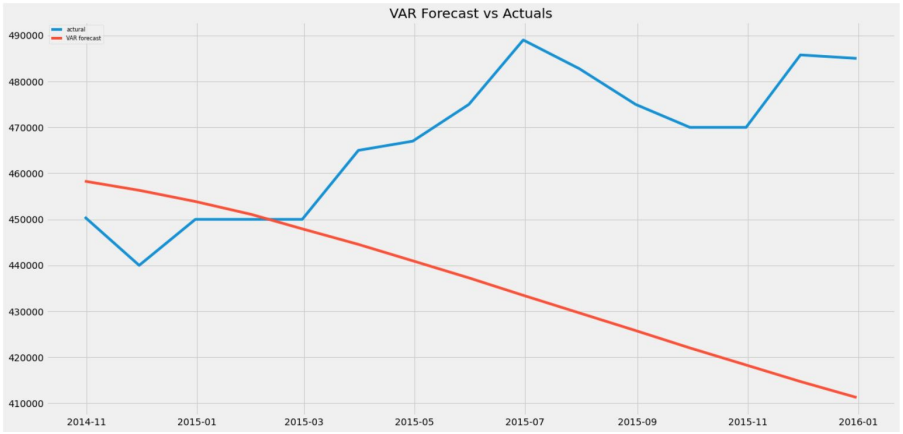


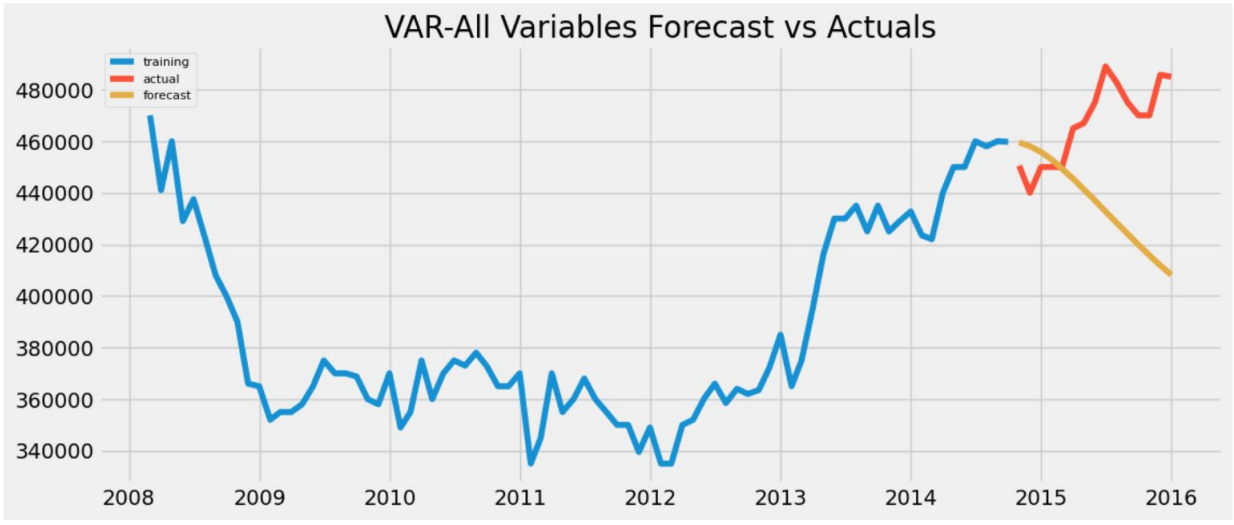


b.VAR-MedianSoldPrice, MedianMortageRate

rmse = 42138

mape = 0.128

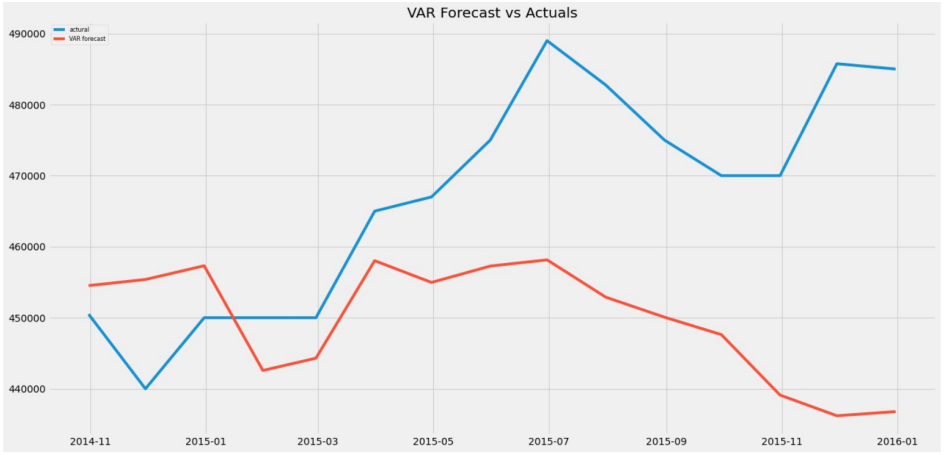


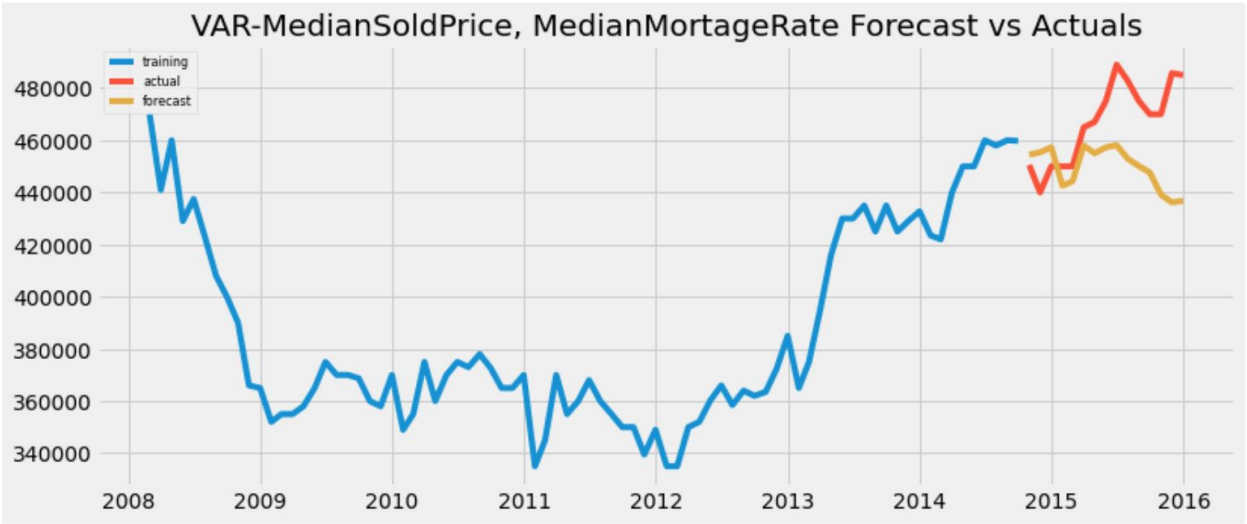


c.VAR-MedianSoldPrice,UnemploymentRate

rmse = 25297

mape = 0.187





2. **SARIMAX**

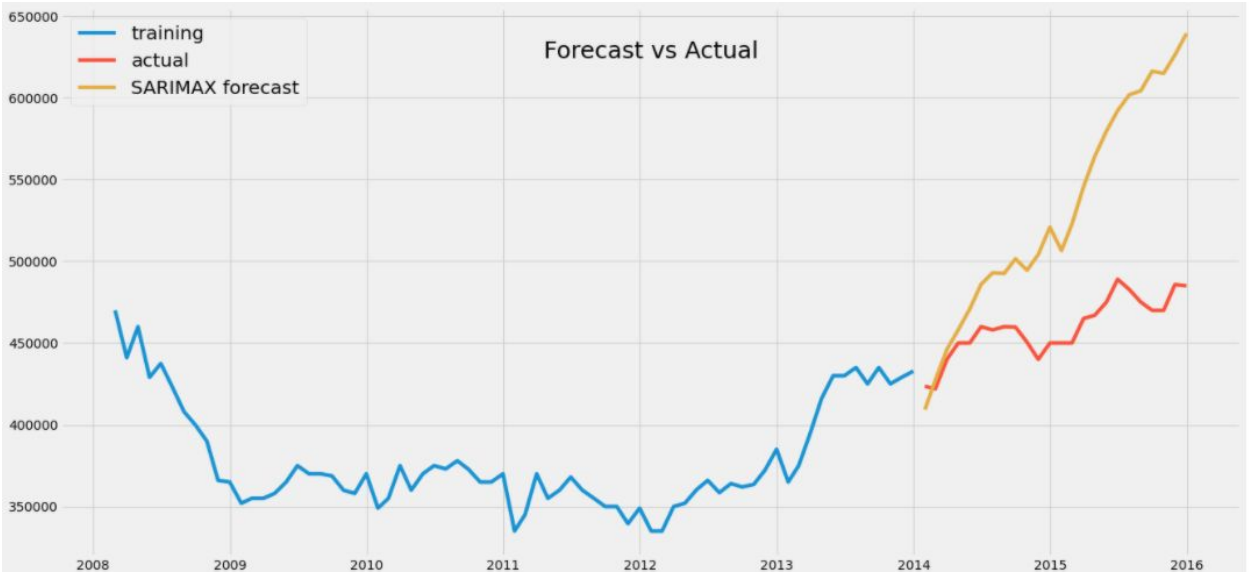
In SARIMAX, the endogenous outcome is MedianSoldPrice,the endogenous features are the variables other than MedianSoldPrice. We perform an auto arima search for the SARIMAX and the best model is the ARIMA(3,1,0)(0,1,1,12) intercept model. Based on the result of this model, we identify the differencing d=1 to remove the trend presented in the data, and the D=1 to remove seasonality. The model was evaluated using the validation set and the evaluation scores are:

rmse=86457

mape=0.153

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(1,1,1)[12]             : AIC=-31.915, Time=0.45 sec
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=-37.588, Time=0.09 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=-35.442, Time=0.24 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=-34.641, Time=0.36 sec
 ARIMA(0,1,0)(1,1,0)[12]             : AIC=-35.618, Time=0.16 sec
 ARIMA(0,1,0)(0,1,1)[12]             : AIC=-35.626, Time=0.15 sec
 ARIMA(0,1,0)(1,1,1)[12]             : AIC=-33.728, Time=0.60 sec
 ARIMA(1,1,0)(0,1,0)[12]             : AIC=-37.306, Time=0.11 sec
 ARIMA(0,1,1)(0,1,0)[12]             : AIC=-36.628, Time=0.08 sec
 ARIMA(1,1,1)(0,1,0)[12]             : AIC=-35.835, Time=0.27 sec
 ARIMA(0,1,0)(0,1,0)[12] intercept   : AIC=-40.367, Time=0.05 sec
 ARIMA(0,1,0)(1,1,0)[12] intercept   : AIC=-38.378, Time=0.26 sec
 ARIMA(0,1,0)(0,1,1)[12] intercept   : AIC=-38.385, Time=0.17 sec
 ARIMA(0,1,0)(1,1,1)[12] intercept   : AIC=-36.445, Time=0.61 sec
 ARIMA(1,1,0)(0,1,0)[12] intercept   : AIC=-42.534, Time=0.10 sec
 ARIMA(1,1,0)(1,1,0)[12] intercept   : AIC=-42.229, Time=0.27 sec
 ARIMA(1,1,0)(0,1,1)[12] intercept   : AIC=-46.637, Time=0.45 sec
 ARIMA(1,1,0)(1,1,1)[12] intercept   : AIC=inf, Time=0.51 sec
 ARIMA(1,1,0)(0,1,2)[12] intercept   : AIC=inf, Time=1.03 sec
 ARIMA(1,1,0)(1,1,2)[12] intercept   : AIC=inf, Time=1.05 sec
 ARIMA(2,1,0)(0,1,1)[12] intercept   : AIC=-47.192, Time=0.45 sec
 ARIMA(2,1,0)(0,1,0)[12] intercept   : AIC=-42.364, Time=0.25 sec
 ARIMA(2,1,0)(1,1,1)[12] intercept   : AIC=inf, Time=0.70 sec
 ARIMA(2,1,0)(0,1,2)[12] intercept   : AIC=inf, Time=1.39 sec
 ARIMA(2,1,0)(1,1,0)[12] intercept   : AIC=-42.628, Time=0.39 sec
 ARIMA(2,1,0)(1,1,2)[12] intercept   : AIC=inf, Time=1.84 sec
 ARIMA(3,1,0)(0,1,1)[12] intercept   : AIC=-49.511, Time=0.63 sec
 ARIMA(3,1,0)(0,1,0)[12] intercept   : AIC=-47.910, Time=0.26 sec
 ARIMA(3,1,0)(1,1,1)[12] intercept   : AIC=-49.059, Time=0.99 sec
 ARIMA(3,1,0)(0,1,2)[12] intercept   : AIC=inf, Time=1.57 sec
 ARIMA(3,1,0)(1,1,0)[12] intercept   : AIC=-47.562, Time=0.69 sec
 ARIMA(3,1,0)(1,1,2)[12] intercept   : AIC=-47.048, Time=1.99 sec
 ARIMA(4,1,0)(0,1,1)[12] intercept   : AIC=-47.510, Time=0.60 sec
 ARIMA(3,1,1)(0,1,1)[12] intercept   : AIC=-47.350, Time=0.64 sec
 ARIMA(2,1,1)(0,1,1)[12] intercept   : AIC=-47.469, Time=0.78 sec
 ARIMA(4,1,1)(0,1,1)[12] intercept   : AIC=-45.542, Time=0.76 sec
 ARIMA(3,1,0)(0,1,1)[12]             : AIC=-48.013, Time=0.47 sec

Best model:  ARIMA(3,1,0)(0,1,1)[12] intercept
```



3. **LSTM**

We also tried the deep learning model LSTM on this time series data. We used four hidden layers and a fully-connected dense layer. We chose 100 epochs and 7 as batch-size.. The RMSE and MAPE on my validation set (Jan_2015 - Dec 2015) didn't beat the univariate prophet model. From the plot below, it seems like LSTM catch the underlying data trend but is systematically underpredicting:

```
plt.plot(inv_y,color='green')
plt.plot(inv_yhat,color='red')  # red is the predicted value from LSTM
```

```
[<matplotlib.lines.Line2D at 0x7fcbe893bdf0>]
```

Here is a summary table oRMSE and MAPE on validation set (Jan 2015 - Dec 2015) across all the univariate and multivariate time series models mentioned above:

	RMSE	***MAPE***
ARIMA(1,2,1)	25079.515	0.0444
SARIMA(1,1,0)(0,1,0,10)	13888.247	0.0243
ETS	13965.578	0.0244
Prophet	7388.590	0.0125
SARIMAX(3,1,0)(0,1,1,12)	86457.800	0.1527
VAR	43561.757	0.0750
LSTM	20522.241	0.0396

Winner model: Prophet

Next we train the prophet model on the entire training set (up to Dec 2015) and forecast on the 2016 test set (Jan-Dec 2016). Below is the RMSE/MAPE from the final model and the visualization of performance on the 2016 test set:

Prophet model	RMSE = 8919.060	MAPE = 0.0128