Time Series Final Project

Sandhya Kiran Reddy Donthireddy, Yihong Shen, Tiance Tan

Introduction

Data description

The Zillow dataset (modified) recorded Feb 2008- Dec 2015 monthly median sold price for housing in California, Feb 2008-Dec 2016 monthly median mortgage rate, and Feb 2008-Dec 2016 monthly unemployment rate.

Models used

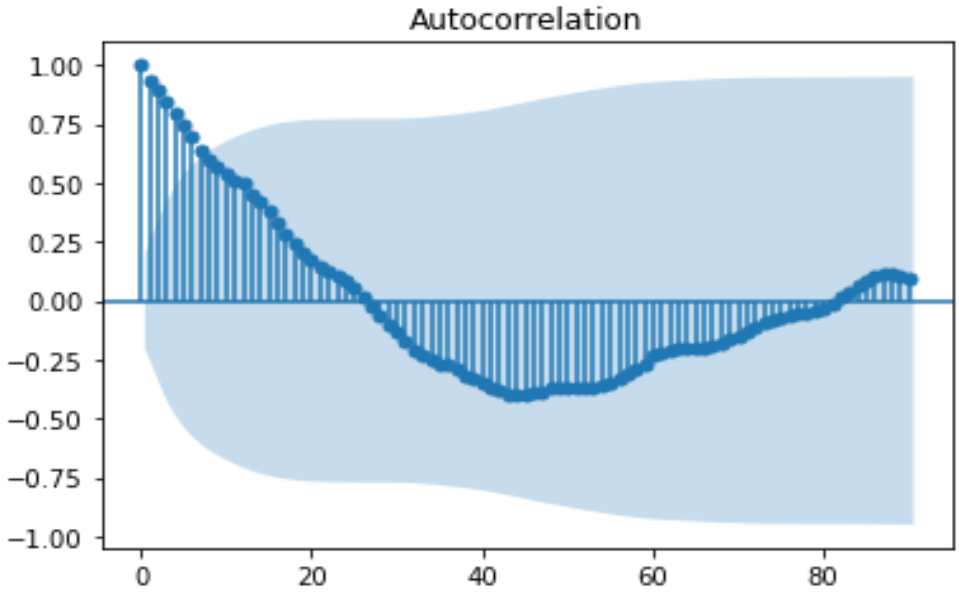
1. ARIMA
   Autoregressive Integrated Moving Average Model
2. SARIMA
   Seasonal Autoregressive Integrated Moving Average
3. ETS
   The ETS models are a family of time series models consisting of a level component, a trend component (T), a seasonal component (S), and an error term (E)
4. Multivariate
   Multivariate time series has more than one time-dependent variable
5. LSTM
   Long short-term memory - a type of recurrent neural network
6. Prophet
   An open source software released by Facebook's Core Data Science team

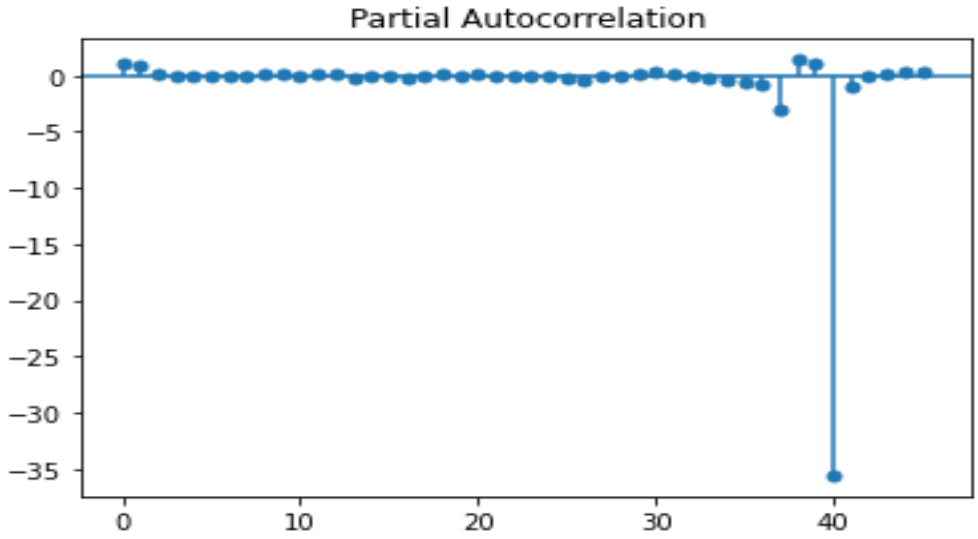
ARIMA Model:

Autoregressive Integrated Moving Average (ARIMA) model is a generalization of an Autoregressive Moving Average (ARMA) model. It is used to fit the time series data and forecasting.ARIMA model is used when there is a trend in the data.Since our data (monthly median sold price) has a trend component, we can explore the ARIMA model to fit the data.

After the initial visualisation of Time series plot, ACF Plot and PACF Plot, we have come to the conclusion that there is a trend in the data but we were not sure about the seasonality period in the data.

The Plots are:





In the time series plot, the trend is a curve and our initial assumption is d = 2(differencing).So we differenced the data twice and checked whether the data was stationary or not using the ADF test and the p- value came to be at 8.027576e-11. But the p-value for the ADF test when we differenced it only once came to be at 0.027443. So, differencing twice makes sense.

We used arma_order_select_ic method from the statsmodels.tsa.stattools package and selected model using BIC, the order selected for ARIMA came to be (0,1) and with d = 2, we have our first model as **ARIMA (0,2,1)**

We also used the bic_sarima function, which basically iterates over all the chosen p, q and d values and using the BIC as the metric, we got our second model and it came to be **ARIMA (0,2,4)**

For these two models, we evaluated the models using rmse_sarima function, which uses one step cross validation with 80 % split and RMSE as a metric. The RMSE for model 1 ARIMA (0,2,1) came to be 10323.72411408035 and RMSE for model 2 ARIMA (0,2,4) came to be 10563.062912056892. So, from these two models, it seems that model 1 ARIMA (0,2,1) is a better model.

SARIMA Model:

The SARIMA model takes into consideration the seasonality in the data and could model the data more appropriately than the ARIMA if seasonality exists. So, to the check the seasonality of the data (monthly median sold price), we did the initial check by differencing the data with m=3,12(quarterly and yearly data) after already differencing it for the trend component and also looking at the plots(Time series plot, ACF and PACF plots)

We were not able to decide on 'm' value since both had low p values when the ADF test was done. The D value chosen is 1. Since there seems to be no multiplicative increase in seasonality, we didn't take the log values of data and continued with the actual data values.

We used the bic_sarima function which which basically iterates over all the chosen p, q and d values and using the BIC as the metric, we got our first SARIMA model and it came to be **SARIMA (2, 2, 2), (0, 1, 2, 12)**

To reduce the computation time while iterating over the P,Q,p,q,d,m values we divided the values into two components and checked for the best model twice. In this way we got another SARIMA model and it came to be **SARIMA (3, 2, 4), (3, 1, 4, 12)**

We also combined the best P,Q,p,q,d,m values from both the iterations and we got the first SARIMA model as the best of both. But to do model evaluation, we used rmse_sarima function, which uses one step cross validation with 80 % split and RMSE as a metric. The RMSE for model 1 **SARIMA (2, 2, 2), (0, 1, 2, 12)** came to be 11487.43526426531 and RMSE for model 2 **SARIMA (3, 2, 4), (3, 1, 4, 12)** came to be 11542.284366872449. So, from these two models, it seems that model 1 **SARIMA (2, 2, 2), (0, 1, 2, 12)** is a better model.

LSTM Model

- **Model Description:**

  Long short-term memory (LSTM) is an artificial recurrent neural network architecture used in the field of deep learning. LSTM networks are well-suited to classifying, processing and making predictions based on time series data.

- **Set up for input and output:**

  Because LSTM is a model for multivariate time series analysis, thus we try to combine the test data with the extra information in the training dataset.

	Median House Price	MedianMortageRate	UnemploymentRate
2016-01-31	476250	3.91	5.0
2016-02-29	466000	3.96	4.9
2016-03-31	485000	3.60	5.0
2016-04-30	501000	3.60	5.0
2016-05-31	501000	3.59	4.8
2016-06-30	505000	3.59	4.8
2016-07-31	507000	3.46	4.9
2016-08-31	510000	3.46	4.8
2016-09-30	510000	3.42	5.0
2016-10-31	523000	3.36	5.0
2016-11-30	506000	3.47	4.8
2016-12-31	510000	4.07	4.7

The first column is from the test set, and the other two are from the training set.

- **Normalize and transform data:**

  In an effort to make data fit in the neural network, we want to feed the model with the input of the previous month's house price, median mortgage rate, and employment rate; therefore the input has three columns, while the output has one column which is the current house price. The combination matrix of the input and output is also normalized between 0-1 for the best potential outcome.

	var1(t-1)	var2(t-1)	var3(t-1)	var1(t)
1	0.876623	0.729927	0.363636	0.688312
2	0.688312	0.784672	0.340909	0.811688
3	0.811688	0.777372	0.386364	0.610390
4	0.610390	0.795620	0.363636	0.665584
5	0.665584	0.843066	0.340909	0.571429

The first five rows of the input and output combination matrix.

- **Split on the train and validation dataset:**

We decided to use the last 12 months' house prices as the validation dataset since the neural network would perform better with more data, such that we can make a concrete decision on the best combination of parameters on batch size and hidden layer. For the parameter tuning part, we make batch size 1, 3, 6, 12 to indicate the potential monthly/quarter/semester/annual trend, and hidden layer in the range of 8, 16, 24, 32 in order to capture the 3 input as there would be 2**3 in a total of 8 gates.

batch_size	hidden_layer	mape	rmse
6	32	0.0047	2953
6	24	0.0047	3001
12	32	0.0046	3035
12	24	0.0047	3221
3	8	0.0047	3446
6	16	0.0049	3448
12	16	0.0051	3630
1	24	0.0053	3695
1	8	0.0051	3727
3	32	0.0055	3772
3	16	0.0055	3940
1	32	0.0059	3993
3	24	0.0059	4036
6	8	0.0074	4819
1	16	0.008	5105
12	8	0.0111	6497

The combination of mape and rmse values for different combinations of parameters.

Prophet Model

- **Model Description:**
  Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

- **Set up for input and output:**
  Since the prophet model has the specific requirement for the feature name as 'ds' and label name as 'y', we rename the original data frame to fit in the model.

	ds	y
0	2008-02-29	470000.0
1	2008-03-31	441000.0
2	2008-04-30	460000.0
3	2008-05-31	429000.0
4	2008-06-30	437500.0
...	...	...

The date needs to be in the ‘ds’ column and the predicted value is renamed as ‘y’

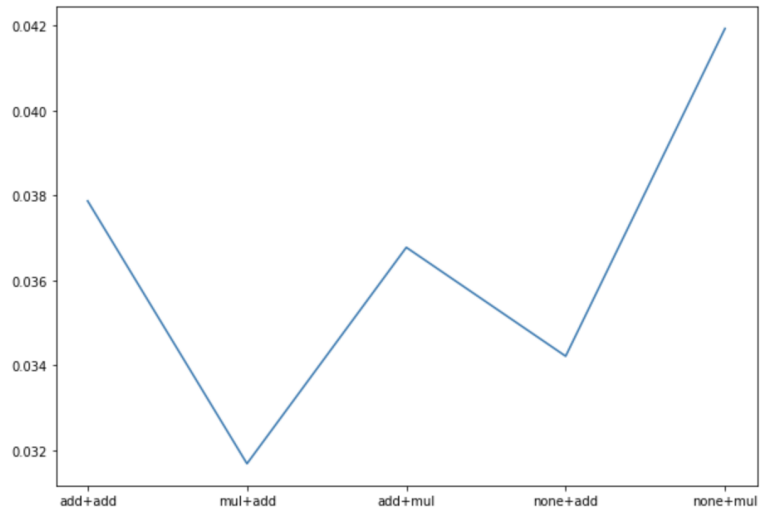
- **Split on the train and validation dataset:**
  We decided to use the last 12 months’ house prices as the validation dataset to find the best potential prophet model. Without any seasonality and holidays adding to the base model, it gives us an outstanding score of rmse 7388 and mape 0.0124. We also try the parameter tuning on the frontier order, seasonality, and period but the scores do not show a significant difference.

ETS model

The ETS models are a family of time series models consisting of a level component, a trend component (T), a seasonal component (S), and an error term (E)

ETS model uses exponential smoothing and we can use it if there is a trend or seasonality in the data. Based on the visualization plots we can see there is a trend. So we can use the ETS model.

Then we fit 6 models with combinations of trend = None/additive/multiplicative and Seasonality = additive/multiplicative.



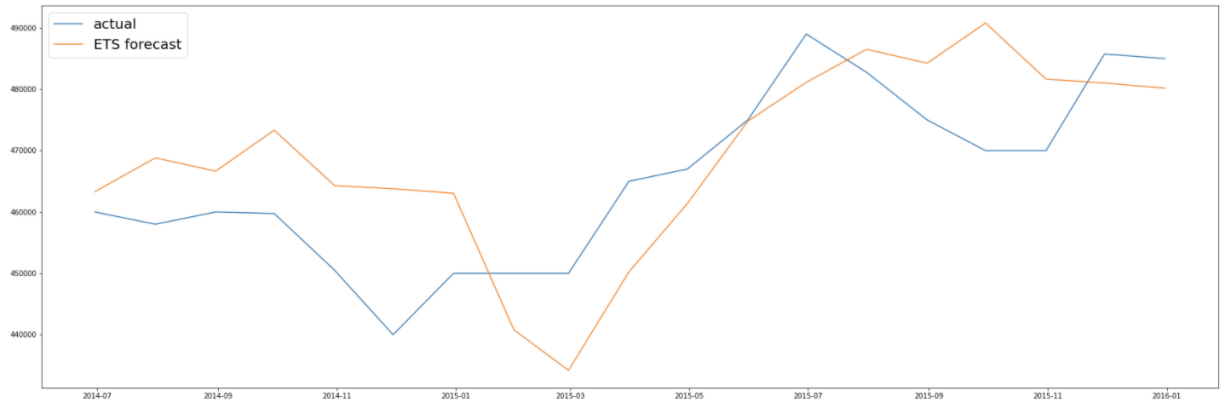
By finding the mean absolute percentage error for each model, we realize the model with multiplicative trend and additive seasonality has the lowest mean absolute percentage error value, which means it is the best ETS model.

Then we fit the best ETS model and predicted the result for the future and calculated the MAPE and RMSE between the actual values and predicted values.

Result:

The MAPE score is 0.0221
The RMSE score is 11795.34, which is high and reflects the poor ability of the model to accurately predict the data.

The plot of predicted values and actual values is below

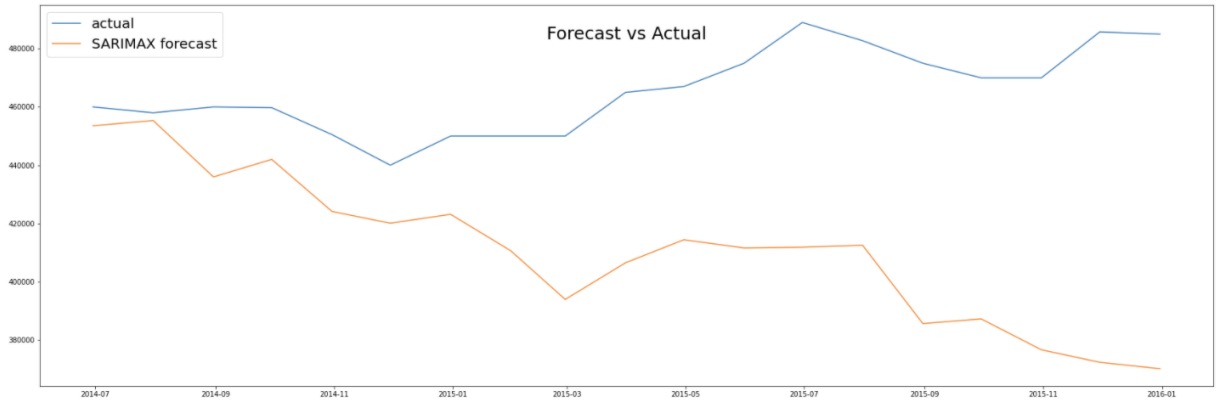


Multivariate Time Series

The data we are looking at has three variables that were recorded with same time steps. Since they may have a relationship/correlation with each other, we use multivariate time series models to make the prediction.

We include all three variables MedianMortageRate, UnemploymentRate, MedianSoldPrice and build SARIMAX models.
Then we choose the model with the lowest AIC scores, which is ARIMA(4,1,0)(0,1,0)[12] with AIC=-140.796

Result:
The MAPE score is 0.1155
The RMSE score is 64152.90. We can see this model does not perform better than the ETS model.
The plot of predicted values and actual values is below



RMSE values from different models on validation set:

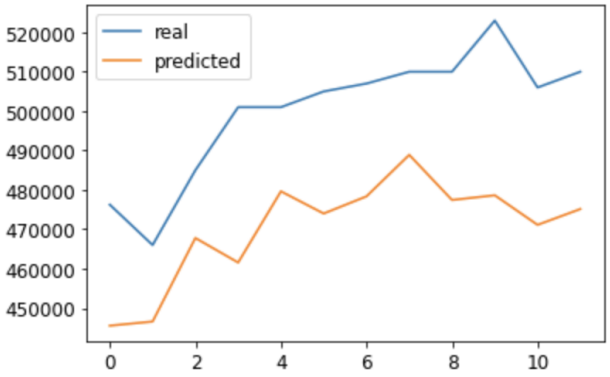
Model	RMSE values
ARIMA	10323
SARIMA	11487
ETS	11795
Multivariate	64152
LSTM	2953
Prophet	**7388**

Final Choice:

Since the Prophet model has the second lowest RMSE value, our Final model was chosen as the Prophet model. The LSTM model is not chosen as we are not confident enough on how its hidden layer would perform on the test set. Neural network model probably has a potential problem of overfitting, in a result little change on the test data would impact model's performance on rmse, so we decide not to risk our best model on lstm but keep it as an evidence we explore the model option.

- **Final result and forecast:**

  With the above indications, we set the basic prophet model as the best one because adding seasonality and holiday does not improve the model performance. Though the actual prices always exceed the predicted one, the difference in the gap makes it a decent prediction. The final model on the test dataset has a score of **RMSE 8919 and MAPE 0.0129**.



The visualization for real house prices and the predicted ones.

- **Table of the predicted values of 12 months:**

Date	Forecasting
2016-01-31	445537.61
2016-02-29	446583.25
2016-03-31	467789.99
2016-04-30	461561.39
2016-05-31	479635.83
2016-06-30	474002.58
2016-07-31	478357.38
2016-08-31	488899.62

2016-09-30	477438.63
2016-10-31	478636.08
2016-11-30	471125.28
2016-12-31	475143.30