Executive Summary

Bankrate.com offers free rate information from over 4,800 institutions on more than 300 financial products. It makes profit by receiving referral fees from the institutions to which it refers to customers. The goal of this project is to help Bankrate predict three mortgage rates (30-year fixed mortgage rate, 15-year fixed mortgage rate, 5/1 arm mortgage rate) in a short future to better recommend mortgage products to its users and to generate more referral revenue.

This report will introduce a complete process to conduct a multivariate time series forecasting utilizing modern deep learning technologies. It will cover the data collection & manipulation, feature engineering & selection, modeling methodology and results evaluation with recommendations in the final conclusions.

The analysis shows that except for the impact of modeling methods, the feature engineering and selection also play very important roles in the forecasting process. Due to the noisiness of predictors’ time series data themselves, the traditional forecasting methods may not work very well. However, for simpler task with small size of samples, the deep neural network may not be the best choice in terms of the cost-benefit balance since it’s expensive to train and tune.

Detailed Report

* Introduction and Motivation

Bankrate.com offers free rate information from over 4,800 institutions on more than 300 financial products. It makes profit by receiving referral fees from the institutions to which it refers to customers. The goal of this project is to help Bankrate predict the weekly average of three mortgage rates (30-year fixed mortgage rate, 15-year fixed mortgage rate, 5/1 arm mortgage rate) in a short future to better recommend mortgage products to users and to generate more referral revenue.

Imagine if you are planning to buy a home in the near future, you may be concerned by the rising mortgage rates since they may force you to settle for less than your ideal home or may even affect your ability to buy a home. Therefore, it’s very valuable to provide accurate forecasts of mortgage rates to help the potential buyers make better decision.

Although it’s very useful to forecast the trend and variation of mortgage rates, it’s not easy to do this, especially when a granular prediction (i.e. weekly) is required. Why? Because mortgage rates are more relevant to a long-term expectation on the economy. The 30-year fixed or 15-year fixed rates don’t change a lot in a short period. Therefore, when you see a variation of the values of average mortgage rates over weeks, a large part of this variation is due to noises, which makes it very difficult to make a weekly forecast. What’s more? The potential ‘leading indicators’ for mortgage rates, like the 10-year T-Note rate, GDP or national income level, are also very complex indicators which are hard to predict themselves. Therefore, it’s very difficult to get rid of the influence of noises and build a forecasting model that is robust enough over time.

Based on the primary study and investigation, the explanatory methods are ruled out and the analysis are focused on those predictive modeling techniques.

* Data: Source and Frequency

First, we need to determine the data sources needed to find those relevant indicators and the frequency we should use to build the model.

Thanks to the R package ‘quantmod’, we’re able to download most major economic and financial indicators from a wide range of high-quality sources like FRED and YAHOO.

The frequencies of the data vary from daily to yearly from these sources. Since we’re going to make forecasts on a weekly basis, only frequencies higher than or equal to weekly, the target frequency, are considered, which are daily or weekly data. That’s because if we want to predict a weekly variation, a monthly predictor will not be helpful, since its values keep unchanged for around four weekly data points.

After confirming the data sources and frequencies, some desktop research was conducted to discover all the candidate predictors that might have an impact on predicting the future mortgage rates. The candidates are listed as below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code** | **Name** | **Source** | **frequency** | **release date** | **start year** |
|  | **mortgage rates (targets)** |  |  |  |  |
| MORTGAGE30US | 30 years fixed | FRED | weekly | thur | 1971 |
| MORTGAGE15US | 15 years fixed | FRED | weekly | thur | 1991 |
| MORTGAGE5US | 5/1 arm | FRED | weekly | thur | 2005 |
|  | **stock market** |  |  |  |  |
| IXIC | Nasdaq index | yahoo | daily | - | 2007 |
| DJI | Dow Jones index | yahoo | daily | - | 2007 |
|  | **macro economy** |  |  |  |  |
| T10YIE | 10-Year Breakeven Inflation Rate | FRED | daily | - | 2003 |
| T5YIFR | 5-Year Forward Inflation Expectation Rate | FRED | daily | - | 2003 |
| T5YIE | 5-Year Breakeven Inflation Rate | FRED | daily | - | 2003 |
|  | **federal rate** |  |  |  |  |
| FF | Effective Federal Funds Rate | FRED | weekly | wed | 1954 |
|  | **bonds and notes** |  |  |  |  |
| VXTYN | CBOE 10-Year Treasury Note Volatility Futures | FRED | daily | - | 2003 |
| DGS10 | 10-Year Treasury Constant Maturity Rate | FRED | daily | - | 1962 |
| DFII10 | 10-Year Treasury Inflation-Indexed Security | FRED | daily | - | 2003 |
|  | **libor rate** |  |  |  |  |
| USD1MTD156N | 1-month libor rate | FRED | daily | - | 1986 |
| USD3MTD156N | 3-month libor rate | FRED | daily | - | 1986 |
| USD6MTD156N | 6-month libor rate | FRED | daily | - | 1986 |
| USD12MD156N | 12-month libor rate | FRED | daily | - | 1986 |
|  | **employment** |  |  |  |  |
| IURSA | Insured Unemployment Rate | FRED | weekly | sat | 1971 |
| CCNSA | Continued Claims (Insured Unemployment) | FRED | weekly | sat | 1967 |
|  | **housing related** |  |  |  |  |
| MORTMRGN5US | mortgage 5/1 arm margin | FRED | weekly | thur | 2005 |
| RELACBW027NBOG | Real Estate Loans | FRED | weekly | wed | 1973 |
| RHEACBW027NBOG | Real Estate Loans: Revolving Home Equity Loans | FRED | weekly | wed | 1987 |
| MBST | Mortgage-backed securities held by the Federal Reserve | FRED | weekly | wed | 2002 |
| MORTPTS30US | Origination Fees and Discount Points for 30-Year Fixed Rate Mortgage | FRED | weekly | thur | 1971 |
| MORTPTS15US | Origination Fees and Discount Points for 15-Year Fixed Rate Mortgage | FRED | weekly | thur | 1991 |
| MORTPTS5US | Origination Fees and Discount Points for 5/1-Year Adjustable Rate Mortgage | FRED | weekly | thur | 2005 |

Notice that because we exclude the monthly, quarterly or yearly frequency, some major economy indicators like GDP and CPI, which are only provided in at least monthly basis, are not included. Although these are very important information, but again, since we want to forecast mortgage rates’ variation over weeks, a monthly indicator won’t be very helpful.

* Methodology: Description of Procedures

1. Data Preprocessing

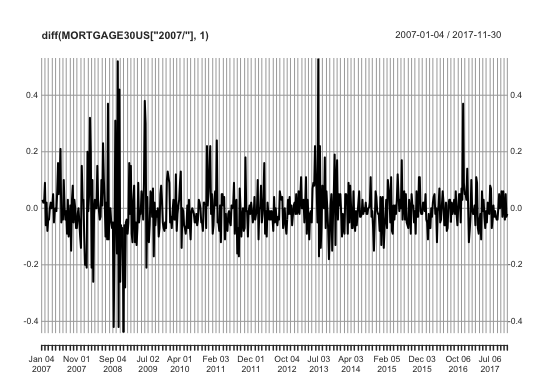
After finishing all the data collection, we set the modeling period as 2007 – 2017 (2007 – 2016 for training and 2017 for validation). That’s because we only have the 5/1 arm mortgage data from 2005 and the Nasdaq and Dow Jones index from 2007 as shown in the above table.

Before we perform further analysis on the data, some preprocessing should be conducted to address two issues: different data frequency (weekly and daily) and timing of updates for weekly data (i.e. some updated every Thursday while others updated every Wednesday). A two-step procedure is performed here:

1. Aggregate all the daily data to weekly frequency and set the cut-off at every Thursday, which is the updating day for the three target variables (30-year, 15-year and 5/1 arm mortgage rates)
2. For all the weekly data whose weekly release day is before or on Thursday, keep them as are. If their weekly release day is after Thursday, then take the lag-1 observation, which is its last week’s value, as the predictor at this moment because we cannot foresee the future.
3. Feature Engineering

Now we have all the preprocessed indicators, one way to build the model is to directly use these indicators as features or perform box-cox transformation on these indicators before feeding them into the model. However, they may not be the best way. Imagine if we calculate the correlation between a feature in week t, Xt, and a mortgage rate in week t+1, Yt+1, if Xt and Yt are highly correlated, since the mortgage rates are quite stable during the past 10 years, it’s very likely that Xt and Yt+1 are also highly correlated because Y doesn’t change a lot over 1 week. However, since we want to accurately predict the future weekly change of the mortgage rates, the indicators with high Xt and Yt+1 correlation might not be very helpful to serve this purpose. So, we performed a two-step feature engineering procedure to cut off this adjacent similarity between Yt and Yt+1 and create new features:

1. Take the lag-1 differencing for all the X and Y to reflect the weekly change. This could be interpreted as to ‘use the weekly change to predict the weekly change’.
2. Standardize the differenced values to avoid the influence of different magnitude. This could be interpreted as to ‘use the volatility of weekly change to predict the volatility of weekly change’.



We can see from the above graph that there seems to be some dependency lying in the volatility of lag-1 differenced 30-year fixed mortgage rate time series from 2007 to 2017.

Therefore, we used the standardized differenced values as the features to build the model. It turned out that it could improve the validation MAPE by 0.1-0.2% (for 30-year fixed mortgage rates, the Naïve Bayes baseline gives a MAPE of 1.6%). Since the mortgage rates are very stable over the past 10 years, this improvement could be regarded as non-trivial.

1. Feature Selection

There are mainly two factors that should be considered in terms of feature selection: the dependency between predictors and the target and the dependency between different predictors. Although there is not necessarily linear dependency, for the efficiency to compute, we performed the correlation and VIF analysis on all predictors and targets. What’s more? Since there may be different dependencies underlying different lags of predictors, we iterated the correlation and VIF analysis over lag1 to lag12 for all the predictors. If any lag of a predictor shows significant correlation with Yt+1 while passing the VIF test, it will be included in the model. After performing this selection practice, the remained features to predict the three mortgage rates are:

30-Year: T10YIE, VXTYN, DGS10, MORTGAGE30US, RELACBW027NBOG, RHEACBW027NBOG, T5YIFR, T5YIE

15-Year: T10YIE, VXTYN, DGS10, MORTGAGE30US, RELACBW027NBOG, RHEACBW027NBOG, T5YIFR

5/1 ARM: VXTYN, DFII10, RHEACBW027NBOG

1. Modeling

Two traditional methods to build a multivariate time series forecasting are:

1. First build univariate forecast for each predictor, then train a linear regression to ensemble these predictor forecasts to predict Y.
2. First train a linear regression on all predictors to predict Y, then build a univariate forecast on Y’s residual series given by the linear regression.

However, there are two drawbacks from these two methods:

1. In many cases the univariate forecasting for the predictors is as difficult as that of Y because the predictors time series are very noisy too. If we predict Y with the estimated predictor values, the results may incur large error and bias.
2. The relations between X and Y may not be linear. If there are any non-linear dependencies in the data, neither of this two methods could work very well.

Based on these thoughts, a non-linear model using features with only existing values is the idea behind the modeling method we used. How to achieve this? First, a recurrent neural network structure could be adopted because it can learn the time dependency across different time steps as well as model the non-linear relationship between predictors and the target. Second, since we’re going to forecast a short period of future directly with all the predictors we have, to make the training and evaluation process more efficient, we want to include all the target values in each row and predict them at one shot. For instance, if the forecasting period is 4 weeks, we include the values of Yt+1, Yt+2, Yt+3, Yt+4 in each row and the goal is to minimize the overall MAE or RMSE for the four predictions simultaneously.

The length of the forecasting period is set to 4 since a 4-week forecast, which is closed to a month, should be valuable for Bankrate to make suggestions to its users while also be approachable from the modeling perspective.

The complete procedures of modeling are described as below:

1. Take the lag-1 differenced values for all predictors and the mortgage rates we want to predict. Then derive next 4-week’s differenced values for the Y (that is, the change from next week to the week after next week, so and so forth). Each row of data looks like (again, all the values below are differenced values and not the original values):

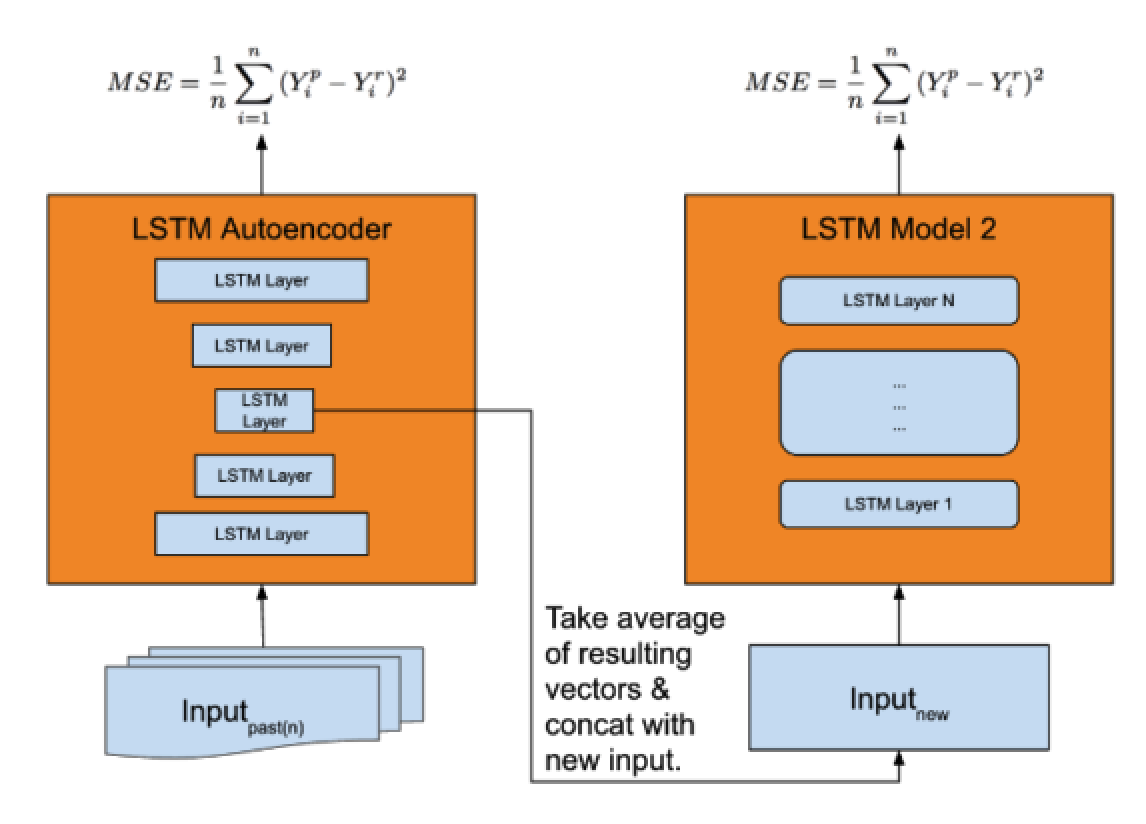
P1t, P2t, …, Pnt, Yt+1, Yt+2, Yt+3, Yt+4

1. Standardize all the predictors and Y in the data and store the parameters for future inverting.
2. Create lags for all predictors. A same degree of lags is applied to all features, this value will be tuned in next steps. Each row of data now looks like:

P1(t-m), …, P1t, P2(t-m), …, P2t, Pn(t-m), …, Pnt, Yt+1, Yt+2, Yt+3, Yt+4

1. Split the data into training and validation sets. Data from 2007 to 2016 goes to training set and data in 2017 goes to validation set.
2. Since the mortgage rates in past 10 years are quite stable. A Naïve Bayes forecast is set to be the baseline model.
3. Construct the single-layer recurrent neural network. Different neuron types are tried (LSTM, GRU) to compare the performance.
4. Invert the standardized predictions , , , in the validation set to original differenced values and add them onto the present mortgage rate value with an iterative process over the four predicted weeks. Validate the results on all usable 2017 data so far (42 observations in total).
5. Tune the parameters (degree of lags, size of the neural network, etc.) by repeating the training and validation process for 10 times (reduce the influence of variance caused by the random starts of weights)
6. Try a more sophisticated RNN structure by extracting the embedding layer from a first RNN which doesn’t include Y itself as predictor and fit a second RNN which feeds the embedding vector and Y itself as predictors and evaluate the results.

Same procedures are performed for the modeling of all the three mortgage rates. The details are skipped here.



The sophisticated RNN structure introduced in step 9 refers to the structure shown in the above picture, which is designed by the data scientists in Uber (See the second reference below). Unfortunately, this more complicated structure doesn’t lead to a better result for our project. This may be due to two reasons:

1. This model is overfitting our data.
2. Since the change for Y itself doesn’t show a more significant power to predict its future changes as shown in the correlation and VIF analysis, it may not be useful to split it out from other predictors and treat it differently.

* Results

After the tuning process, we found that the GRU neuron type and tanh activation function outperformed the LSTM cell and relu activation. Therefore, we used the GRU cell and tanh function in the three final models. Other major hyper parameters shared among all the three models are learning rate (0.01), learning rate decay (0.02), epochs (50) and batch size (52).

Furthermore, we found that removing features with relatively lower correlations with Y and only keeping those with highest correlations will not harm the validation results for all the three models. By following the rule of parsimony, only the most effective predictors are remained in the final models.

The features and other key hyper parameters that performed the best are shown as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Most Effective Predictors | Degree of lags | Dimension of RNN output |
| 30-Year | DGS10, T10YIE, VXTYN, MORTGAGE30US | 6 | 12 |
| 15-Year | DGS10, T10YIE, MORTGAGE30US | 8 | 12 |
| 5/1 ARM | VXTYN, DFII10 | 8 | 10 |

The comparison of results from the recurrent neural network and Naïve Bayes model are as below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metrics | 30-Year NB | 30-Year RNN | 15-Year NB | 15-Year RNN | 5/1 ARM NB | 5/1 ARM RNN |
| MAE | 0.0645 | 0.0551 | 0.5670 | 0.0497 | 0.0445 | 0.0439 |
| MAPE | 1.61% | 1.37% | 1.75% | 1.52% | 1.40% | 1.37% |

One of the drawbacks for the deep learning method is that it cannot directly give a prediction interval because it doesn’t make any assumption on the distribution of the estimates. One way to build a prediction interval is to perform a bootstrap process but it’s very expensive to implement so I’ll leave this for future work.

* Conclusions

This analysis mainly introduces two ways that could effectively improve the results of multivariate time series forecasting:

1. The standardized differenced features, which could be regarded as ‘the volatility of change’, are stronger indicators than the raw indicators themselves because they better reflect the trend and disconnect the similarity of adjacent observations.
2. A recurrent neural network structure could better model the time dependency as well as the multivariate dependencies in the data because it performs a non-linear transformation in each time step and stores the ‘memory’ of information from previous steps in a more sophisticated way.

Except for these advantages, we also show that a straightforward modeling methodology that only using existed values to predict the whole period could do no harm to the results. The nature behind this practice is that it completely converts a statistical modeling task to a machine learning task.

Furthermore, only keeping the most effective features in the model will not undermine the overall forecasting accuracy. So it is not necessary to put all significant features in the model even they pass the VIF test.

On the other side, there’re also some drawbacks for the deep learning methods:

1. The deep learning models cannot directly generate a prediction interval because it doesn’t make any assumption on the distribution of the estimates. We could manually generate prediction intervals by performing a bootstrap process, but it’s very expensive to do this.
2. The advantage of deep learning models is more obvious as the size of the data increases. For this project, we only have about 560 observations. The improvement on the results for such a model is not very substantial compared to a traditional model like the Naïve Bayes. Considering the cost to train and tune a deep neural network is much more expensive, it may not be the best cost-beneficial way for relatively simpler tasks.

Based on all the analysis and results above, some recommendations could be made to Bankrate:

1. The 10-Year Breakeven Inflation Rate (T10YIE), 10-Year Treasury Constant Maturity Rate (DGS10), 30-Year Fixed Mortgage Rate (MORTGAGE30US) and CBOE 10-Year Treasury Note Volatility Futures (VXTYN) are four most important leading indicators to forecast the near future of the mortgage rates.
2. A combined effort of standardized lag-1 differenced predictors (volatility of change) and a recurrent neural network structure could improve the overall forecasting accuracy by 13%-15% (this is like building an ARCH series model with deep learning technique).

* References

Jason Brownlee. Multivariate Time Series Forecasting with LSTMs in Keras, August 2017. <https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/>

Nikolay Laptev, Slawek Smyl, Santhosh Shanmugam. Engineering Extreme Event Forecasting at Uber with Recurrent Neural Network. <https://eng.uber.com/neural-networks/>