Case-Shiller Housing Index Prediction

Project Category: Finance and Commerce

Name: Ari Webb
SUNet ID: arijwebb
Department of Computer Science
Stanford University
arijwebb@stanford.edu

Name: Laywood Fayne
SUNet ID: lfayne
Department of Computer Science
Stanford University
lfayne@stanford.edu

Name: Mattheus Wolff
SUNet ID: mbwolff
Department of Computer Science
Stanford University
mbwolff@stanford.edu

1 Relevant Information

We have no external collaborators and are not sharing this project with any other classes.

2 Introduction

In this paper we used various monthly municipal and macro-economic metrics to predict housing market performance in American cities.

In our research, we came across an index that measures single family house prices called the Case-Shiller index. The Case-Shiller index uses repeat sales of single-family homes to track trends in home pricing. It is normalized to 100 for the year 2000, meaning values before 2000 (for the most part) are under 100, while values after go above 100.

We collected city level data including unemployment data, consumer price index data (CPI), crime rates and patent data. We also used data that plots more general economic trends, such as US stock market data.

We used three types of models to predict the Case-Shiller index. We decided to evaluate model performance by the mean absolute percentage error of predictions for each city twelve months into the future from various thousands of starting points. Each model could use any data from before that starting point to make the prediction. Our first model was linear regression, which used varying months worth of the city's data (previous unemployment rates, crime rates, etc.) as each feature to make its prediction about that city's Case-Shiller index. Our second model was a neural network that was trained on twelve months of only Case-Shiller data, essentially stripping the other features from our examples. Our final model was a Vector Autoregression model, which used time series modeling.

3 Related Work

Maccarrone et al. (1) used K-Nearest Neighbors (KNN), linear regression and several time series models to predict real U.S. GDP. Overall economic growth is tied to the housing market, so we can use some of the same features such as CPI and unemployment. They found that KNN performed best when predicting "one step ahead," but failed to predict longer time horizons well. The time series methods showed different levels of performance for predicting forward for different amounts of time. The regression model managed to be the best when forecasting farther into the future. We thought it would be interesting to apply some similar methods to another macroeconomic index. In principle, "one step ahead" forecasting is not too helpful for determining any policy or making real life decisions, so we also wanted to focus on making predictions several time steps in the future.

Schindler and Conover (2) used a time series model to predict the US housing market via the Case-Shiller index, but only utilized CPI data in doing so. We thought it would be interesting to see if other factors that go into deciding where to buy a home, such as crime rates, would have an impact.

Kaboudan (3) used a multi-agent system to forecast the Case-Shiller indices for San Francisco and San Diego over six months. Kaboudan's approach used both genetic programming and neural networks, but their neural networks only had one to two layers. We though we would could also use a neural network, but experiment with more layers, more cities, and forecasting farther into the future.

Plakandaras et al. (4) used Vector Autoregression (VAR) as a comparison model to their new methodology. They claim that their new model outperforms VAR by producing half the error, and further claim that their model can be used as a warning system for sudden housing price drops. We believe that the simplicity and ease of generalization of VAR may make it a better candidate for applications with many variables, which is why we chose it for this project.

Schimbinsch et al. (5) advocated usage of VAR for traffic forecasting in large urban areas. Schimbinsch demonstrated that VAR has low generalization error and outperforms "baselines and univeriate equivalents over two large area datasets." We thought these aspects of VAR would make it a good model for forecasting the Case Shiller against many other variables.

4 Dataset and Features

We chronologically split our dataset into a training set (75%), validation set (15%), and test set (10%), with the training set starting with the earliest data points for each city, and the test set ending with the latest data points for each city.

Case-Shiller was used as both a feature and label (6), with previous Case-Shiller indices being used to predict future indices. Case-Shiller data was only available for major US cities, prompting us to narrow our project scope to seventeen select US cities. The data is also only available starting in 1990 for most cities, limiting our time frame to the years 1990-2020 in most locales.

City names were encoded as a feature so the models could give specific economic metrics more weight in predicting the index for particular cities. The city names were represented as one-hot representation vectors in each data point, with a 1 in the position of the city that the data belongs to and a 0 in all of the other positions.

The NASDAQ, Dow Jones, and S&P 500 (7) were the only nationwide economic metrics we used as features. For a month's data point, we averaged over the mean price of the indexes for each day of the month. To calculate the mean price of an index for a given day, we calculated the average between the opening and closing price for the index on that day.

We used CPI data for motor fuel (8), medical care (9), food and beverages (10), and all items more broadly (11) so that our models could gauge the cost of living in different cities. Each month's data point for a particular city contained one feature for each of the aforementioned categories. We also calculated and included the per capita number of employees in educational and health services (12) for each city so that the models could gauge the quality of educational and health services. Our intuition was that the higher that number, the more attention teachers and doctors could give to individuals, resulting in better quality of life and consequently higher home prices.

We added crime (13) and unemployment rates (14) as features in our dataset so that the models could use them to determine the undesirability of housing markets. We figured people would not be moving in as much to cities that were experiencing high crime and unemployment, and thus would drive down the costs of homes in the area.

Resident population numbers (15) were used so our models could track the rate of growth of a city, and thus be able to predict higher housing prices with the resulting larger demand for homes. Per capita personal income (16) was also utilized so that the model could recognize cities that had very attractive job markets, and monthly numbers of new patent assignments (17) were incorporated so the model could identify areas that were the source of lots of technical innovations and thus new business opportunities that would attract employees.

For linear regression, our tuned data points contained each of the aforementioned features on a monthly basis for eighteen months, for a total 287 features. That is seventeen features for the city one-hot vector, and eighteen months of data, with fifteen features per month. For the neural network, each data point contained only the Case-Shiller index feature on a monthly basis for an entire year, for a total of 12 features. The corresponding label was the Case-Shiller index for that city twelve months into the future. For the vector autoregression, each data point contained one month's worth of features for a total of thirty-two features, as that model was a time series. If data was not available in a monthly format, but was rather annualized for example, we linearly extrapolated the data to get an approximation for each month.

5 Methods

First, we ran linear regression as a base line. We ran the regression on our dataset including all of our processed features. To tune this model, we changed the number of prior months that made up the features of one example (e.g. including 6/12/24 consecutive months of data to predict a year into the future from the last month).

Next, we decided to use a neural network as another prediction method. Building off of linear regression, we wanted to see if we could generate a model that was effectively city independent, only using the Case-Shiller data. This is helpful in the real world because we would theoretically be able to use this model to predict housing market performance for cities not contained within our dataset if we are able to develop some sort of correlation metric to the seventeen cities that we are representing as features.

Our neural network contained dense hidden layers of varying sizes and then a single real number for the output. It was optimized using Mean Absolute Percentage Error. The particular choices for relevant hyperparameters are discussed in the next section.

Finally, we used a time series based model called Vector Autoregression (VAR). The VAR model is a generalization of the single-variable autoregressive model, where a variable at time t is predicted as a sum of past values of that variable multiplied by trained coefficients. The VAR of variable X at time t includes past values of X, past values of other variables Y and an error term (18). We used a stationary VAR test.

We implemented VAR as follows. First, we split the data into time series' for each city. Then, we conducted the Granger Causality test on each time series. Before doing the VAR model, each time series' needed to be stationary, meaning that its mean and variance does not change over time. To achieve this, we conducted an Augmented Dickey-Fuller test, followed by taking the difference of the time series' twice, resulting in stationary data. Finally, we ran the VAR model on the stationary data, inverted the forecast output, and plotted that against the actual test data.

Granger Causality Statistics determine whether the lagged values of one variable help to predict another variable (19). Causality here is not synonymous with cause. Granger Causality is better defined as "precedence", or the belief that a particular variable comes before another in a time series.

The output of the Granger Causality test is a matrix of p-values. We call this matrix the Granger matrix. The p-value for any given element in the matrix measures the null hypothesis of whether "variable X fails to Granger-cause variable Y." The Granger Matrix's diagonal entries are all 1, as every variable must fail to Granger-cause itself. The rest of the entries take values between 0 and 1. Values that are below .05 meet the thresh hold to reject the null hypothesis, while values above .05 do not reject the null hypothesis. The result of this step is that variables X that cannot reject the null hypothesis for Y are not useful in the VAR model for variable Y, and so can be omitted.

The Augmented Dickey Fuller (ADF) test determines whether a time series has a unit root. The presence of a unit root means that the data is non-stationary (20) and thus cannot be used in a VAR model. The null hypothesis for this test is that there is a unit root. If the test results in a p-value of less than .05, then we can reject the null hypothesis and claim that the data is stationary.

We took the ADF test of each time series, followed by taking the first and second derivatives of every time series. This resulted in all our data being stationary. Finally, we ran the VAR model on the stationary data, then integrated the data twice to undo the derivatives we took and compared the values to the test data.

To evaluate our models, we used Mean Absolute Percentage Error. We chose this over the standard Mean Squared Error because MSE is scale dependent. The Case-Shiller indices between the different cities vary quite significantly. There are other more nuanced, loss metrics that we considered, but ultimately decided against, which we will discuss later.

6 Experiments/Results/Discussion

After tuning the linear regression model by changing the amount of months of previous data we included as features, we found that eighteen was the optimal number of consecutive months to include within our set of features. With this hyperparameter tuned, linear regression yielded a 7.533% test error.

For the neural network, we started by using the dataset with one hot encodings for city names and twelve consecutive months of data for features (for a total of 161 elements per sample). We used twelve months instead of eighteen because we wanted to see if this more complex model could make better predictions with a more limited amount of data. We processed this data through NNs with different layers, layer sizes, batch sizes, and activation types. We found that models were consistently getting MAPEs of around 10% on the validation set, significantly worse than our results for linear regression. Since the NN doesn't have any sense of time and is treating each feature essentially independently, we believe that it was creating complex connections between features that shouldn't exist, which effectively confused the model.

From here, we decided to look closer into the features themselves. We ran Kendall correlation on the training features, comparing them with their corresponding label. We chose this statistic because we cannot assume that the data is normally distributed and because it does not require continuous data. In addition, it measures a purely monotonic relationship between variables (rather than a linear relationship like with Pearson correlation). The most correlated features were the prior Case-Shiller values (in descending order from the latest to the earliest), with correlations all above 0.75. The other features all reported correlations below 0.52 and were largely grouped together irrespective of time. The next five in descending order were the DOW Jones, income, S&P 500, motor fuel, and CPI.

With this knowledge, we ran the NN with varying parameters on different subsets of the features. Beginning with just the Case-Shiller data (cutting out the one hot city representations and other real numbered values), we achieved validation errors of around 5.5%, a significant improvement over using the whole dataset. When adding in different groupings of the other features to train on, the neural network's performance worsened, with errors jumping up to the 7-8% range. Based on these findings, we decided to restrict the dataset we used for the rest of our work with our NN model to the past twelve months of Case-Shiller index values.

We then looked to optimize the hyperparameters of our neural network. The aspect we chose to focus on (tested options in parentheses) were the type of network optimizer (Adam, Nadam, Adamax, Adadelta, SGD, and RMSprop from the Keras library), number of layers (1, 2, 5, 10), batch size (16, 32, 64), number of neurons (3, 6, 12, 24), and activation layer type (ReLU, Leaky ReLU). These initial lists were determined empirically through choices that performed well during our previous analysis. We trained models on every combination of these parameters and sorted the results by their best performance measured by MAPE for the validation set. We selected the prominent parameters from the most successful models and re-ran the analysis using a narrower range around these parameters. Through this tuning, we were able to evaluate overfitting to the training set and choose a more generalizable model.

We found that the optimal parameters were the optimizer Adam, a batch size of 16, and three Leaky ReLU layers with 24 neurons each. It makes sense that Adam was chosen over SGD and RMSprop because our implementation used just 50 epochs with a patience of 5 in order to be computationally efficient, so it is likely that there was too much noise for effective convergence. The remaining four optimizers have pretty similar and even overlapping implementations, so the choice of our hyperparameter optimization script validates common sentiment and recommendation that Adam tends to be the best optimizer (21). Similarly, the batch size is an interesting feature of our dataset and model, and seems reasonable since our training dataset is on the order of 1000 samples. Three layers of 24 neurons each allow for adequately complex predictions to be made with the data, without overfitting or developing invalid relationships similar to when we used all of our features. The Leaky ReLU selection makes sense since we greatly narrowed our number of features, so we have less latitude to lose any information because of dead neurons caused by a ReLU. This model returned a test error of 2.358%. On a relative percentage basis, this result is $100\% \cdot (7.533 - 2.358)/7.533 = 68.7\%$

better than that for linear regression. Our initial inquiry was confirmed since the neural network was able to produce a noticeably better prediction with significantly less features.

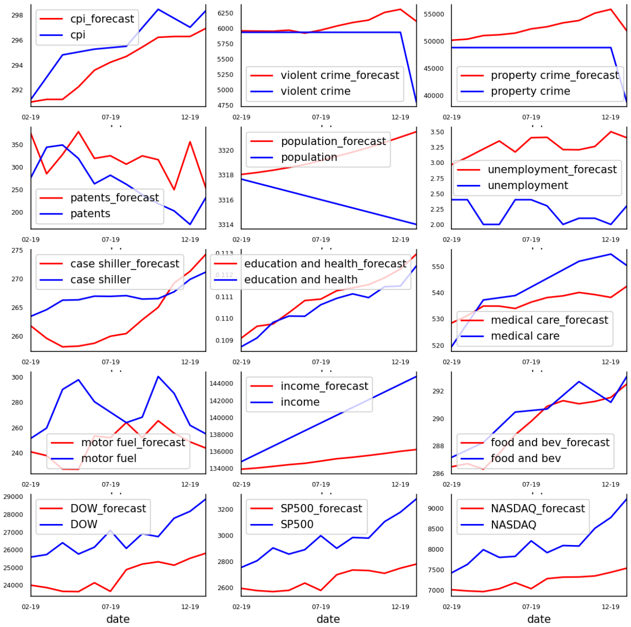

Figure 1: San Francisco Projections

The VAR model had test error of 9.235%, which was computed in the same way as above. We believe the VAR model could have had a lower test error if we spent more time calibrating the lag, and had more computational resources to thoroughly tune this hyperparameter. We also believe that there could have been more work done evaluating which variables had causality, and eliminating/scaling variables that did not have causality. Figure 1 (above) shows forecasts for all the variables for San Francisco projecting until January 2020. It is interesting to see how some trends were predicted, such as some of the shape of the unemployment data, albeit shifted. Although, it is clear that there are varying levels of success with each feature and that there is more to investigate with regards to under/overprediction.

7 Conclusion/Future Work

The neural net performed the best at predicting the Case Shiller index twelve months into the future. Linear regression did second best, followed by the VAR model with the highest error. We have a few ideas about how to expand our work in the future.

For linear regression and the neural network, we could use a weighted loss function that penalizes overprediction more than underprediction. We could implement this with a constant multiplier or one that increases with the size of the underprediction. This is relevant for investment strategies in general since people tend to be risk averse and would rather not lose money. This is ultimately an empirical decision though, since everyone has different preferences.

For VAR, we could employ more rigorous measures to determine causality of variables before running the model. We could also do more experimentation with different lag values.

We could also try working with other time series forecasting models, such as Long Short-Term Memory (LSTM), AutoRegressive Integrated Moving Average (ARIMA) and Trigonometric, Box-Cox transform, ARMA errors, Trend and Seasonal components (TBATS).

8 Contributions

Matt Wolff: Data acquisition, data formatting, data compilation, linear regression modeling, write-up.
Laywood Fayne: Data formatting, data compilation, neural network modeling, write-up.
Ari Webb: Data cleaning, VAR modeling, write-up.

References

[1] M. Giovanni, M. Giacomo, and S. Sara, "Gdp forecasting: Machine learning, linear or autoregression?," *Frontiers in Artificial Intelligence*, vol. 4, 2021.

[2] F. Schindler, "Predictability and persistence of the price movements of the s&p/case-shiller house price indices," *The Journal of Real Estate Finance and Economics*, vol. 46, no. 1, pp. 44–90, 2013.

[3] M. Kaboudan and M. Conover, "A three-step combined genetic programming and neural networks method of forecasting the s&p/case-shiller home price index," *International Journal of Computational Intelligence and Applications*, vol. 12, no. 01, p. 1350001, 2013.

[4] V. Plakandaras, R. Gupta, P. Gogas, and T. Papadimitriou, "Forecasting the u.s. real house price index," *Economic Modelling*, vol. 45, pp. 259–267, 2014.

[5] F. Schimbinschi, L. Moreira-Matias, V. X. Nguyen, and J. Bailey, "Topology-regularized universal vector autoregression for traffic forecasting in large urban areas," *Expert Systems with Applications*, vol. 82, pp. 301–316, 2017.

[6] FRED, "Case-shiller home price indices." https://fred.stlouisfed.org/release/tables?rid=199&eid=243552, 2022.

[7] WSJ, "U.s. stocks." https://www.wsj.com/market-data/stocks?mod=nav_top_subsection, 2022.

[8] BLS, "Bls data finder 1.1 - motor fuel." https://beta.bls.gov/dataQuery/find?removeAll=1&q=Motor+Fuel, 2022.

[9] BLS, "Bls data finder 1.1 - medical care." https://beta.bls.gov/dataQuery/find?st=0&r=20&more=0&q=Medical+Care, 2022.

[10] BLS, "Bls data finder 1.1 - food and beverages." https://beta.bls.gov/dataQuery/find?removeAll=1&q=Food+and%20Beverages, 2022.

[11] FRED, "Consumer price index by expenditure category." https://fred.stlouisfed.org/release/tables?rid=10&eid=35849, 2022.

[12] FRED, "All employees: Education and health services." https://fred.stlouisfed.org/searchresults/?st=All%20Employees%3A%20Education%20and%20Health%20Services&t=msa&ob=sr&od=desc, 2022.

[13] FBI, "Federal bureau of investigation crime data explorer." https://crime-data-explorer.fr.cloud.gov/pages/explorer/crime/crime-trend, 2022.

[14] BLS, "Bls data finder 1.1 - unemployment rate." https://beta.bls.gov/dataQuery/find?st=0&r=20&more=0&q=Unemployment+Rate, 2022.

[15] FRED, "Annual population by location." https://fred.stlouisfed.org/release/tables?rid=119&eid=162982, 2022.

[16] FRED, "Personal income by county and metropolitan area." https://fred.stlouisfed.org/release?rid=175, 2022.

[17] FRED, "New patent assignments." https://fred.stlouisfed.org/release?rid=432, 2022.

[18] *Vector Autoregressive Models for Multivariate Time Series*, pp. 385–429. New York, NY: Springer New York, 2006.

[19] J. H. Stock and M. W. Watson, “Vector autoregressions,” *Journal of Economic Perspectives*, vol. 15, pp. 101–115, December 2001.

[20] R. Harris, “Testing for unit roots using the augmented dickey-fuller test: Some issues relating to the size, power and the lag structure of the test,” *Economics Letters*, vol. 38, no. 4, pp. 381–386, 1992.

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference for Learning Representations*, January 2017.

(6) (13) (7) (17) (11) (15) (14) (16) (8) (9) (10) (12) (2) (19) (20) (18) (5)