6.867 Final Project

AirBnB and The Boston Housing Market

David Yang Stephanie Pavlick Israel Ridgley

December 2017

Abstract

AirBnB is an online marketplace through which people are able to lease or rent short-term housing. Since the company's founding in 2008, it has altered the way people travel, and has also made it easier for people to rent out properties short-term. The variety of rentals through AirBnB vary greatly in their amenities and layouts, making pricing highly variable, and the effect of AirBnBs in communities on home values is still largely unclear. As rental prices are set by the AirBnB host, they are subject to market forces and also the personal opinion of the host. For our final project, we wished to determine whether we could predict price and popularity of an AirBnB unit based on amenities and ratings given by previous guests of the unit. We also wished to assess whether fluctuations in home prices could be predicted based on AirBnB stock in a given area.

1 Introduction

As AirBnBs are fundamentally apartments available for short-term rental, they vary greatly in price and amenities. Additionally, AirBnBs take away from the existing permanent housing stock in a city. Therefore, increasing numbers of AirBnBs in an area can decrease the amount of available permanent housing and potentially drive up the home prices in the area [9]. We were interested in characterizing what makes an AirBnB attractive and how AirBnBs effect real estate values.

For this project, we focused on evaluating AirBnBs within the city of Boston only. Our dataset included AirBnB listings throughout Boston between September 2014 and September 2017. These data were taken from the Inside AirBnB project [3] website and were scraped directly from the AirBnB website. We focused on using our available data to predict three separate pieces of information: median home price in a neighborhood, AirBnB price per night, and the frequency of an AirBnB being booked. Our available features included whether an AirBnB provided certain amenities, such as a refrigerator or washing machine, the occupancy of a unit, the unit's rating from previous travelers, and the neighborhood of Boston in which that unit is located, among others. Our dataset also included a calendar that listed whether an AirBnB was available for booking each day of the year. However, the calendar does not distinguish between being booked and being unavailable (e.g. for cleaning, renovation, etc).

2 Project Contributions

To split up the work for this project, each group member was responsible for tackling one question related to AirBnBs. Each group member applied various machine learning techniques to their respective problems and assessed which were best suited to predict outcomes. David primarily addressed how to predict an AirBnB's nightly price, Stephanie investigated the frequency with which AirBnBs were booked throughout Boston, and Israel worked to predict the median neighborhood rent prices throughout Boston.

For our implementation, we used Python packages Scikit-learn[6] for machine learning algorithms, Pandas[5] and Numpy[8] for data processing, as well as Statsmodels[7] for time series forecasting.

3 Predicting AirBnB price per night

Examining all the listings on AirBnB, it is clear that there is a wide range of AirBnBs, each with its own set of unique features. We are interested in determining if we can accurately predict what an AirBnB price per night will be listed for by considering the features of that AirBnB.

To tackle this problem, a fully connected neural network (NN) with ReLu activations, a linear output layer and a square loss function was implemented ('MLPRegressor' function in [6]) to predict AirBnB prices. An L_2 penalty on the weights in each layer was implemented as well to control overfitting. Additionally, all hidden layers consider the same number of units.

Due to the stochastic nature of NN training, all results were averaged over 10 trainings of the considered NN architecture in each section. The authors note that even without averaging, results were very similar across individual trainings.

The listing data was pre-preprocessed to transform the amenity and neighborhood list into T/F fields for each amenity/neighborhood and all components of the input vector were normalized to the range [0,1] based on the maximum value in each field across the data.

3.1 Neural network regression on minimally processed data

We first consider using the minimally processed data for prediction. The data set was split into training, validation and test sets at a 60%, 20%, and 20% proportion respectively at random. A grid search was conducted for L_2 penalty λ (0.0001 - 1), hidden layer number (1-4) and units per layer (10-1000). The optimal set of parameters were found to be 0.001, 2 and 405 respectively. Note that based on these results, $\lambda = 0.001$ is used for all results moving forward. However, even in this optimal case, the error is high with an absolute error of \$64.34 or 35.9% (average AirBnB listing price of \$179.05). It was hypothesized that the length of the input vector (109) hinders performance and reducing the input dimension would improve the predictor accuracy. The first method considered was to replace the neighborhood components with the median price in that neighborhood for the class of apartment/house of the AirBnB listing. Median neighborhood prices are taken from Zillow research [10] which splits median prices according to bedroom number. For example, for a 1 bedroom apt in Cambridge, the median 1 bedroom apartment price taken from Zillow was used. This approach was deemed appropriate as housing price per neighborhood should capture many qualitative aspects of that neighborhood. For listings that corresponded to missing housing prices in the Zillow data, the listings were removed; total number of listings removed from the data set in this manner was <1%. Additional methods considered are feature selection by variance thresholding, feature selection by mutual info regression, using human knowledge to remove irrelevant features, and dimension reduction through principal component analysis (PCA).

3.2 Augmenting with Zillow data

For the Zillow-revised input data, a grid search to determine the values for hidden layers (1-3), and units per layer (10-10000). We were limited to these ranges due to computational limitations; there is minor evidence that deeper networks may slightly improve predictor performance however. Several of the results are shown in table 1. It can be seen that the mean absolute errors and R2 scores are 1) similar among all NN architectures and 2) improved from the minimally processed data by about 20%, showing the additional pre-processing was warranted. For hidden layer numbers of 1, 2, and 3, the optimal number of units were all 10000. Moving forward, these NN architectures were used for comparison purposes.

Layers	Units per layer	Mean Error	R2 value
1	100	54.08	000
2	100	51.19	0.36
3	100	55.28	0.32
1	10000	53.51	0.35
2	10000	**49.76**	**0.45**
3	10000	51.82	0.44

Table 1: NN predictor error for various architectures after Zillow data augmentation

3.3 Data pre-processing for dimensionality reduction

To determine the effect of additional pre-processing to reduce the dimensionality of the input vector, we considered the following methods: variance thresholding, removal by mutual information regression, reduction by principal component analysis (PCA), and pruning features using human knowledge/intuition. For brevity, the results are not presented but overall feature selection efforts did not improve NN predictor performance and in most cases, actually resulted in a predictor with worse performance. The exception was when human knowledge was used to remove what were deemed irrelevant features and this improved predictor performance by about 1-2%. In this case, 12 features were removed and included features such as whether or not the listing included a carbon monoxide detector, smoke detector, etc. These were chosen based on whether or not the authors would consider these while booking an AirBnB. This result suggests that the NN has enough data to learn the importance of each feature and the poor performance is not necessarily due to the input vector length. Additionally, it may be postulated that in certain situations, the injection of human knowledge for pre-processing data can improve the performance of the resulting machine learning predictor.

3.4 Outlier removal

Next, we investigate if there are outliers in the data that could be influencing the predictor. In this case we are concerned with outliers in terms of AirBnB price; if there are listings that have prices that are unrealistically high, the predictor will be highly influenced by these listings as it uses a squared loss function. To investigate if this is the case, the prices in the data are sorted and plotted against percentile, shown in figure 1. It can be seen that there is a sharp increase in

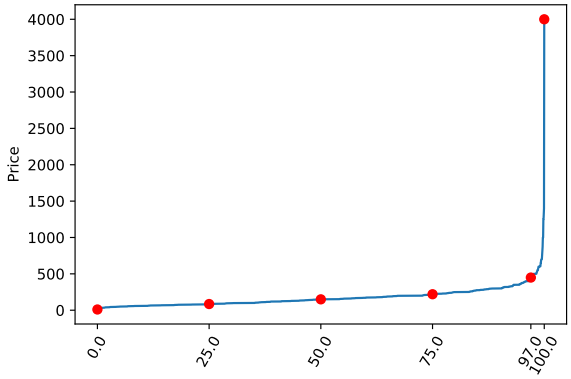

Figure 1: Sorted Prices vs. Percentile

price at the 97th percentile, corresponding to $450. We consider these listings outliers and remove them from the data set. We then retrain the NN predictor with manual pruning as the pre-processing step; the average absolute errors and R2 values are shown in table 2. It

Architecture	Mean absolute error	R2 value
1 HL, 10000 units	36.75	0.66
2 HL, 10000 units	33.92	0.70
3 HL, 10000 units	**33.57**	**0.72**

Table 2: NN predictor errors after price outlier removal

can be seen that removing the outliers greatly improves the performance of the NN predictor, reducing the average absolute error by roughly 30% and increasing the R2 value by 0.3. To further determine if other outliers exist in the data (not just in price), we cluster the training data through a density based method, DBScan [4], which takes as input minimum distance for association (ϵ) and minimum number of points per cluster (N_{min}). Points that do not fall into a cluster are considered outliers. DBScan was chosen as the true number of clusters

is unknown (eliminating the use of K-means or similar algorithms) and it is able to find outliers automatically. Through experimentation, a good clustering was found with $\epsilon = 4$ and $N_{min} = 5$; this results in 36 clusters, 112 outliers and can be seen in figure 2 with T-SNE used for visualization.

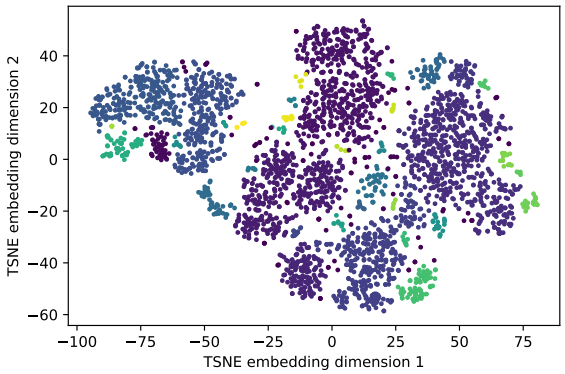

Figure 2: DBScan clustering

We now further remove these outliers from the training set, retrain the NN and calculate error on the test set. Results can be seen in table 3.

Architecture	Mean absolute error	R2 value
1 HL, 10000 units	36.56	0.66
2 HL, 10000 units	34.15	0.70
3 HL, 10000 units	**33.31**	**0.72**

Table 3: NN predictor errors after price and DBScan outlier removal

It can be seen that NN predictor error is only slightly (<1%) better than before, suggesting that the main outliers that influence the predictor are those in price.

4 Predicting AirBnB Occupancy

AirBnBs are highly variable in the type of amenities they offer, the size of the unit, the price, and many other features. However, it is unknown what types of AirBnBs are most popular for guests. This sections seeks to answer the questions of whether AirBnB guests are consistent in their demands for AirBnBs based on amenities, and which amenities are most important in determining the likelihood of a unit to be occupied.

4.1 Pre-Processing

Unfortunately, the occupancy rates of the AirBnBs are not provided as a part of the dataset. To estimate these rates, we used a method used by the city of San Francisco to evaluate AirBnB's presence in the city [2]. We assumed guests write reviews at a rate of 72%, in line with this model, and the average booking length of 3.6 days for AirBnBs in Boston[1]. We then used the number of reviews for a unit to estimate the proportion of the year it was occupied, with an upper cap of 70% occupancy, a number that reflects hotel occupancy rates as detailed by the San Francisco study.

All features were normalized to fall between 0 and 1. A total of 82 different features were used in prediction for each of the methods. In addition to the amenities and other features found in the dataset from Inside AirBnB, the median home price for the size of home as the AirBnB in that unit's neighborhood was included as a feature. These values were found using data provided by Zillow [10].

4.2 Linear Regression

We experimented with using linear regression models with both LASSO regularization and ridge regression to predict the occupancy rates of the AirBnB units. Ridge regression makes use of the L_2 norm for regularization, while LASSO implies use of the L_1 norm.

Figure 3 shows the error rate on the validation dataset as a function of the regularization parameter, λ, for ridge regression. Figure 3 shows that smaller λ give better performance on the validation dataset. The value of $\lambda = 0.1$ was chosen for evaluation on the test dataset. The error rate when applied to the test dataset was 0.1451.

Figure 4 depicts the error rate on the validation set as a function of λ when using LASSO regularization.

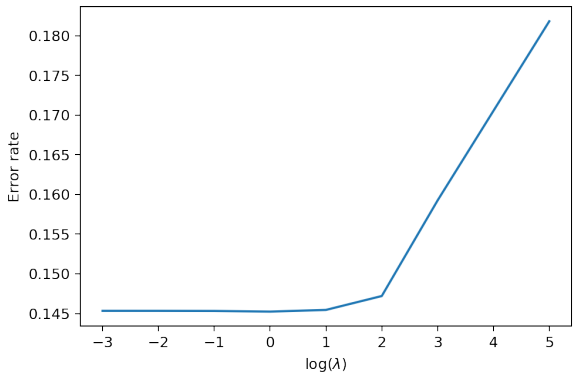

Figure 3: Error rate vs λ for ridge regression

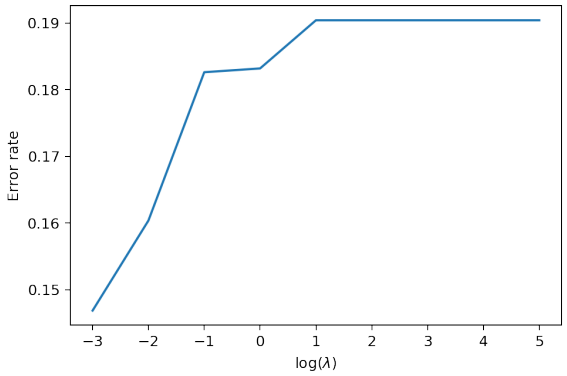

Figure 4: Error rate vs λ for LASSO regression

Like the performance for ridge regression, we found that smaller regularization terms were best for LASSO regularization. However, unlike ridge regression, LASSO regression error rates seem to level off with larger λ values. The value $\lambda = 0.001$ was used to evaluate performance on the test data. For this λ value, an error rate of 0.1453 was achieved.

4.3 Regression Trees

We also used boosted gradient regression trees to predict occupancy rates of AirBnBs. These trees use stochastic gradient descent to train many decision trees on the same dataset, then create a decision tree from the average of all the decision trees found when training. Table 4 shows the error rates on the validation dataset for boosted gradient regression trees with various values for η, the learning rate, and different numbers of estimators, or the number of boosting stages to perform. For each of these iterations, the maximum depth of the regression tree was set to 4 nodes. From Table 4, we can see that, generally, larger numbers of estimators and larger learning rates resulted in a lower error rate. Additionally, an inverse relationship exists between the number of estimators and the learning rate. A learning rate of 0.1 and a total of 1000 estimators were used when evaluating performance of this method on the test dataset. With these values, an error rate of 0.10604 on the test data was achieved.

η	**Estimators**	**Error**
10e-3	10	0.1894
10e-3	10e3	0.1395
10e-2	10e2	0.1394
10e-2	10e4	0.1085
0.1	10	0.1382
0.1	10e3	0.1077
1.0	10e2	0.1236
1.0	10e4	0.1347

Table 4: Error rates on the validation dataset for varying learning rates and number of estimators

5 Forecasting Boston Rent Prices

AirBnB has received enormous controversy over the fact that it bypasses hotel regulations and reduce housing stock in the cities in which it operates. Those that criticize AirBnB claim that its impact on housing stock results in negative externalities for the residents of these cities - they claim that the prevalence of AirBnBs makes it harder to find an apartment in the city and the reduction in supply results in increased rent prices. This section aims to determine the validity of the second of these claims by using Autoregressive models to forecast

the median rent price of various Boston neighborhoods based on the number of AirBnBs. If the inclusion of AirBnB data improves the predictions of these models, then there is reason to believe that the housing market does respond to the prevalence of AirBnBs.

5.1 Data Preprocessing and the Zillow Rent Index

In order to estimate the amount of housing stock off of the market for AirBnB use, we considered an AirBnB listing off of the market between the time of its first and last review. Furthermore, since a studio and a house are not equivalent, we estimated the amount of stock each AirBnB represented to be the mean of its number of bedrooms and bathrooms. We calculated the amount of stock this way because AirBnBs will often advertise a small apartment as accommodating a whole family or count living rooms as bedrooms. Using the number of bathrooms gives a more realistic and conservative estimate of how many people could live in the listed dwelling.

Due to the limitations of the dataset and the annual nature of its data scrapes, some listings that are present in the older datasets have been removed from AirBnB's website and are thus not present in the newer data set. The effect of this, shown in Figure 5 is plateaus of housing stock growth as listings disappear at dataset edges. Additionally, given that the datasets were officially scraped starting in 2015, the amount of housing stock devoted to AirBnB before then is surely an underestimate. Because of this, we decided to only train our model starting at month 50, saving 8 months at the end for testing.

For this section the Zillow Rent Index is used as a proxy for actual rent prices for a neighborhood. The rent index is an estimate of median rent prices and it incorporates the rents of various property types into a single index, making it easy to gauge the cost of housing. Figure 5 shows that the Zillow Rent Index and

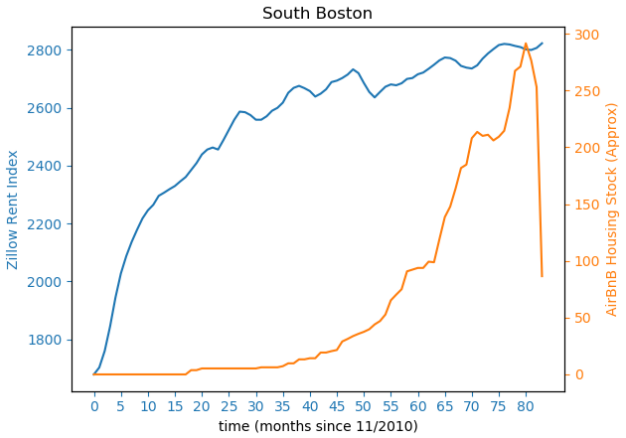

Figure 5: Twin plot of Zillow Rent index and AirBnB Housing Stock for South Boston since 11/2010

AirBnB stock are positively correlated, indicating the possibility of a causal relationship.

5.2 Autoregressive Models

Since time-series data is causal and time-varying in nature, the methods from class cannot be directly applied. Instead the order of the data must be respected. Therefore we decided to use the linear ARX (Autoregresive with exogenous input) model is used. The ARX model given by:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{i=0}^{p} \theta_i X_{t-i} + \epsilon_t \qquad (1)$$

where Y is the time-series variable of interest, c is a constant, p is the lag order of the model, ϕ_i and θ_i are parameters of the model to be learned, X is the time-series exogenous input variable, and ϵ_t is the realization of white noise at the current time step. The ARX model is implemented using the ARIMA class in statsmodels, with I and MA parameters set to 0, and the documentation can be seen for more information about the model implementation. Essentially, the Autoregressive model incorporates lagged values of the observed and input variables to predict future values. It does this by fitting the data at each time step using the maximum likelihood estimate implemented with a

Kalman Filter. A Neural Network could be used as a nonlinear implementation with the lagged values as inputs for each time step; however, due to the limited training data that we have, training a meaningful NN would be in-feasible.

We initially fit our ARX model with a lag parameter of order 5 based on the 99% confidence of the rent index auto correlation in Figure 6. After training the model we used a prediction horizon of 8 months given our data size and that the AR predictions regress to the mean relatively quickly. After tuning our model, we arrived at a lag parameter of order 2.

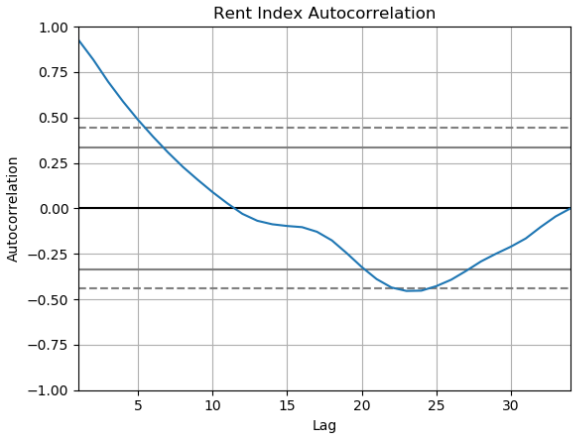

Figure 6: The autocorreltion of rent index for South Boston, dashed line is 99% confidence, solid line is 95% confidence

5.3 Prediction of the Zillow Rent Index

Given the number of neighborhoods in Boston, we only concerned ourselves with predicting the rent index of South Boston in order to make visualization and analysis easier. Figure 7 shows the forecasting results of our model. The blue line shows the actual rent index over the test interval, while the orange and green lines show predicted index. The orange line is a benchmark of using no input data and only the lagged values of rent index and predicted rent index. The figure shows that when the model is trained using the South Boston AirBnB data (green), it is much better at predicting the trend of the rent index.

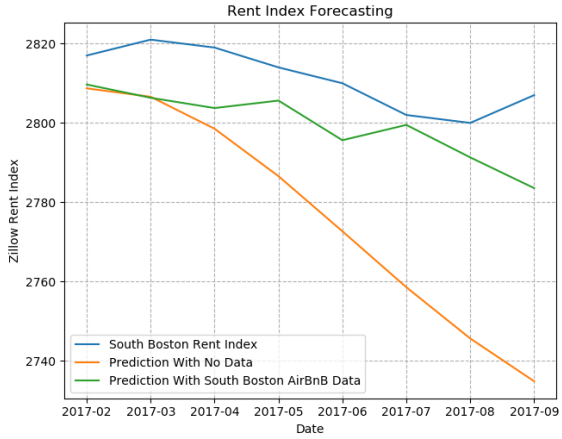

Figure 7: ARX rent index forecasting of South Boston based on South Boston AirBnB data as input, using no input data as a control

Next we used the data of other neighborhoods of Boston, as well as the aggregate AirBnB data for all of Boston, to make predictions. Figure 8 shows the model results for different neighborhoods as input data. We chose to use the North End (orange) and Chinatown (red) because they represent a neighborhood in a similar price bracket and a much lower price bracket respectively. The fact that predictions using the North End data track the South Boston rent index well, while the Chinatown predictions do not, indicates that there may be a relationship between the housing markets of certain neighborhoods. The aggregate data (purple), formed by adding up the AirBnB housing stock across all neighborhoods, did not perform particularly well compared to individual neighborhoods.

6 Discussion

6.1 AirBnB Price Prediction

Overall, the AirBnB price prediction problem proved difficult with mean prediction error on the test set never dropping below 18.5% and an R2 value never exceeding 0.72. However, we found that predictor performance

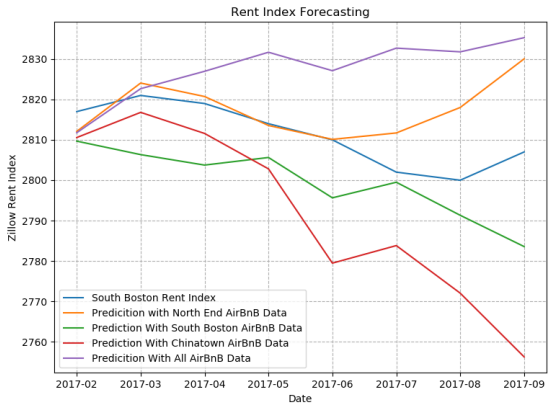

Figure 8: ARX rent index forecasting of South Boston based on various neighborhood input variables

could be significantly improved by combining the Zillow data with the listing data and removing any outliers in the data. Each of these steps improved performance by 20%.

Additionally in this setting, we found that most dimensionality reduction techniques were unsuccessful in increasing predictor accuracy with only human knowledge feature pruning reducing predictor error. This suggests that 1) almost all of the features provide some information for price prediction (and reducing this amount of information results in a worse predictor) and 2) human knowledge/intuition in data pre-processing still has the potential to improve the performance of machine learning techniques.

We can conclude that with the currently considered data set, prediction still has significant error which 1) may be due to the fact that AirBnB pricing is inherently noisy since it is the product of human intuition/choice or 2) there are important features not represented in the data set. One example of #2 is picture data for each listing. Many people use pictures to choose which AirBnBs to book and pay for and this could be a significant feature. Future work would focus on possibly implementing a convolutional neural network to classify pictures according to the sentiment they convey, using this output as an additional feature for prediction purposes.

6.2 AirBnB Occupancy Prediction

The results show that AirBnB occupancy rates can be approximately predicted using regression methods. Linear regression performed somewhat poorly, with error rates reaching as high as 20%, and regularization doing little to improve performance on the test set. Boosted regression trees were more effective at predicting occupancy rates of a unit with about a 10% error. This suggests that AirBnB occupancy can be somewhat accurately predicted using the data provided from the AirBnB online listing. Additionally, the inclusion of the Zillow data in the feature set increased the accuracy of regression.

6.3 Rent Price Forecasting

The results of the ARX model would suggest that, at least from a prediction standpoint, a relationship between housing stock consumed by AirBnBs and rent prices exists. Furthermore, the results indicate that these effects are not isolated but that AirBnBs can affect neighborhoods with similar housing markets. Non-local AirBnB effects are balanced by the fact that AirBnBs in dissimilar markets can actually negatively affect forecast accuracy when compared to no data. These asymmetries thus explain why using the aggregate AirBnB data was not fruitful - the predictive power of some neighborhoods is balanced by the misinformation of others. Future work should be focused on determining the network of AirBnB effects in the housing market and creating a more complex model that incorporates non-linearity with a Neural Network. Overall our results give credence to those that would criticize AirBnB's role in driving up rent prices.

References

[1] *Airbnb Citizen*. URL: https://www.airbnbcitizen.com/data/boston-2015/.

[2] San Francisco Budget and Legislative Analyst's Office. "Analysis of the impact of short-term rentals on housing". In: (2015). URL: http://sfbos.org/sites/default/files/FileCenter/Documents/52601-BLA.ShortTermRentals.051315.pdf.

[3] Murray Cox and John Morris. *Inside AirBNB*. URL: http://insideairbnb.com/index.html.

[4] Martin Ester et al. "A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231. URL: http://dl.acm.org/citation.cfm?id=3001460.3001507.

[5] Wes McKinney. "pandas: a Foundational Python Library for Data Analysis and Statistics". In: (2011).

[6] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[7] J.S. Seabold and J. Perktold. "Statsmodels: Econometric and Statistical Modeling with Python". In: *Proceedings of the 9th Python in Science Conference*. 2010.

[8] Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30. DOI: 10.1109/MCSE.2011.37.

[9] Curt Woodward. "Airbnb rentals trigger debate over housing stock". In: *Boston Globe* (2016).

[10] *Zillow Data*. URL: https://www.zillow.com/research/data/.