Final Report: House Price Prediction in Atlanta

Shahrokh Shahi, Zichen Wang, Wenqing Shen, Yixing Li, Dong Gao, Xiangyi Yan

Georgia Tech

1. Introduction

House purchase comes with different considerations, of which two important factors are price and location. The traveling time from houses to popular shopping centers is a good measure of how convenient it is for people to live in an area, and can affect people's purchase decision and houses' market prices. Although models for house price prediction have been researched and evaluated extensively in the past, an interactive web-based system that provides house price prediction and traveling time information has not been developed. Our goal for this project is to build an interactive web application that predicts house prices in the Atlanta Metropolitan Area, meanwhile providing traveling information to nearby shopping centers. Considering the significance of traveling time, it was considered as a feature in our price prediction training.

2. Problem Definition

This project is to build an interactive application that could predict house price and provide traveling information based on users' input. From the back-end, we need to develop a machine learning (ML) regression model. As traveling time to popular places affects people's decision on where to live, traffic information should be included as features in machine learning model training. Prior to modeling training, necessary data need to be collected and cleaned. As for the front-end, given users' specific housing requirements, the application should present the output data in a direct way.

3. Survey

To consider the traveling time for the prediction model, one classical model is standard urban model (SUM), which assumes one center for a city and correlate the house price with the distance to the center of the city [1]. Larger cities have higher house price volatility due to lower elasticity of resources and there is higher spatial heterogeneity for the submarkets in larger cities [2]. In suburban regions, the house supply has high elasticity and the price tends to be stationary [3]. The house price prediction model can be implemented in a location-based recommendation system (RS). Such system provides predicted values based on location information and hence increases accuracy [4]. Park et al. [5] developed a RS taking in spatial variation using Bayesian Network [6, 7] which can efficiently provide the conditional probability distributions of results. It has also been suggested that shopping centers have significant impact on the property prices in the vicinity [15]. Sirpal et al. argues that local shopping centers contribute positively to the prices on local houses. Therefore, we will introduce the distances to the top 20 shopping centers in Atlanta as training features. No previous machine learning model on house price prediction has included this feature to our best knowledge.

Vladimir et al. [8] proved that Random Forest is powerful generating accurate results when making prediction. However, since it's hard to see the exact structure of each tree, we cannot intuitively see the relationships between features and predictions [9]. As for Ridge Regression, [10] it can proficiently avoid overfitting by adding penalty, also Marquardt et al. [11] displayed the "ridge trace" to better tune parameters based on sensitivities between coefficients and data sets. But these articles also show the shortcoming that there still exist impacts of irrelevant features since it doesn't enforce coefficient to 0. Neural network is robust to error and gives good performance as a prediction model according to works by Odom [12], while the tradeoff is larger computational burden and higher possibility to overfitting [13]. Gradient Boosting keeps searching for steepest descent to minimize loss function, which is an efficient way [14]. In this project, several machine learning models are developed and compared.

4. Methodology

The general design of the methodology can be visualized in figure 1. We have collected data from Zillow, cleaned the raw data, introduced travelling time to local centers, ran different ML models, and created a visualization web application. Details will be discussed in sub-sections.

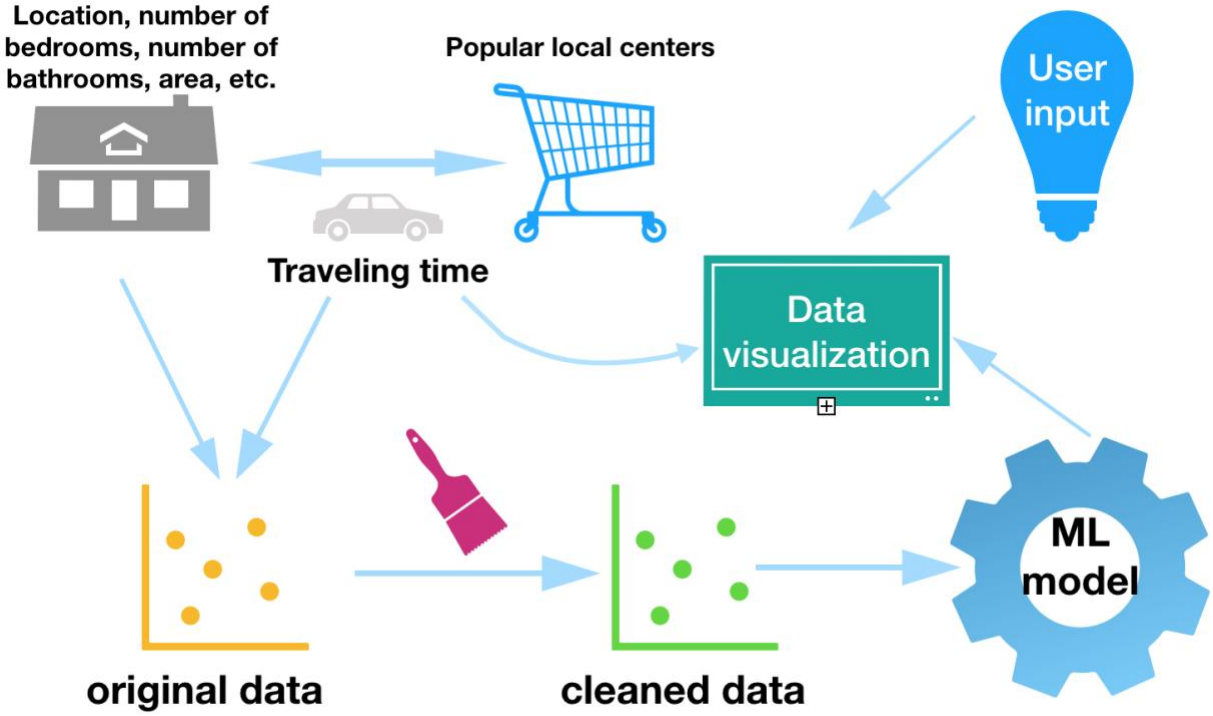

Figure 1. Framework of this project

Based on the previous works in related fields, our approach involves the following innovations:

1. interactive map

2. comparison between different ML models
3. providing traveling time information.

4.1. Data Collection and Preprocessing

The real estate data used in this project are obtained from Kaggle and Zillow, containing ten thousand entries. The Zillow data are extracted using a web scraper, including both recently-sold and for-sale houses data. The property details include property location (e.g. City, State, longitude and latitude etc.), features (i.e., number of bedrooms and bathrooms and sqft) and price. The scraping logic, briefly speaking, is first to construct an URL of the search results page from Zillow manually. Then download the entire HTML page using Python Requests. Parse the page using LXML which lets you navigate the HTML tree structure using Xpaths. Consequently, the data needed are selected out and got saved. After scraping all the listings on the first page, a new URL of next results page is generated based on the information stored in the web elements. The data on the new page are extracted. These scraping procedures will continue until all search results are scrapped.

Google Maps API is utilized to get the locations of popular local centers. The direct distances between each property and local centers are representative of traveling information, which are used in subsequent machine learning training.

The data obtained are cleaned using excel and OpenRefine. Further information of postcode / city and corresponding latitude / longitude is incorporated to the original data by joining multiple tables.

4.2 Front-end Development

The user interface consists of a basic front-end developed using HTML, CSS, and JavaScript. Bootstrap is the main framework, which has been employed for easier and faster front-end development. Another library used to develop the front-end is Material Design for Bootstrap (MDB), which is mainly based on Bootstrap and provides a collection of flexible components. Some of these elements are implemented to design a typical page layout including header, navigation bar, and footer sections to obtain a simple but functional and efficient interface.

The user interface mainly consists of two distinct parts: the input section and the output section. The input section includes an input panel in the right to take the input values from user, and an interactive map in the left, which enables user to pick a desirable point to search. The Google Maps JavaScript API has been employed to create and tailor such interactive map for the purpose of this project. By clicking on any point on this map, a marker surrounded by a modifiable circle will appear on the map, and the correspondence information (latitude and longitude of the point, and the specified search radius) will be prepared to send to the back-end. By pressing the “Search” button, all the information gathered in the input section will be transferred to the backend for processing. As soon as the process is completed, the results will be posted on the corresponding section in the text form.

To be more descriptive, a d3 visualization has been augmented to the front-end which consists of a histogram of the available data points in our database for the requested information in the input

panel. Moreover, a transparent range slider is provided to enable the user to limit the price range. At the same time, the associated markers are displayed on the interactive map which gives the user more information and a better sense of data.

4.3 Back-end Establishment

Back-end is developed using Flask, and is composed of two parts. The first part receives the user input and selects from data the records that satisfy requirements. The back-end then sends records to the front-end, which employs d3 as visualization tool. For the second part, back-end receives zip code, area, number of bedrooms and number of bathrooms from the front-end. It then proceeds to calculate the distance from selected zip code to all local centers and calls machine learning codes. Machine learning will return jsonified results, including estimated price to Flask, and then to the "results" column on the Web page. Both parts work independently, and invalid input for one part (e.g. wrong zip code) does not influence the output for the other. Back-end also incorporates wtform module to restrict input. For illegitimate input, the website will not proceed but instead flash specified error message.

4.4 Machine learning

4.4.1 How to include traveling time information

In this project, training process needs to consider traveling information. Standard urban model assumes one center for each city, which might not work well for big cities like Atlanta. Big cities usually have several centers with high population density. Another problem for implementing the standard urban model is how to select the centers properly. A simple way is based on population density. Popularity of a place depends not only on population, but also on safety, surrounding utilities, etc. Shopping centers are usually located at popular centers, providing a shortcut to find proper centers. Also as discussed in the survey section, previous work has shown having shopping centers in the neighborhood has positive impact on the house prices. Therefore we decide to include distance to the shopping centers as additional features. 20 shopping centers are selected for Metro Atlanta. Getting traveling time from each house to each center could be done by Google API. This is plausible for small datasets, while for data sets containing thousands of houses, the query exceeds Google's free limit. The direct distance is used for the time being. For practical business use, premium API account could be used to provide more accurate traveling information. The features we have decided to train are: square footage, number of bedrooms, number of bathrooms, latitude and longitude, city, and the 20 shopping malls in Metro Atlanta area.

4.4.2 Machine Learning Models

Several machine learning models are tested, including Random Forest, Neural Network, SVC, and Ridge Linear Regression. Decision tree has problems of overfitting. The Random Forest builds trees based on multiple sub-samples and then uses averaging to help avoid over-fitting. The bootstrap method is applied. Neural Network is efficient for image and model with shallow

layers could be applied to our regression. SVC is effective for multiple-dimension space and cost less memory. Ridge Linear Regression uses ridge regularization to select useful features. All above mentioned models are tested and evaluated in this project, and the best model would be selected for the final use.

4.4.3 Evaluation/Scoring

To provide direct observation about prediction accuracy, mean absolute relative difference score is used.

$$\text{Score}(y_i, y_{predict}) = -\frac{1}{n}\sum_{i=1}^{n} \left| \frac{y_i - y_{predict}}{y_i} \right|$$

For example, if a house is $100,000, a prediction of $90,000 or $110,000 gives score of 1-0.1=0.9, which indicates the error range is within 10%.

4.4.4 Model Selection

Grid search cross validation is used for parameter tuning. It was implemented by sklearn package. Shuffle split was used for cross validation, with split number of 10 and test size of 10%. Best model and parameter set should have low prediction error for the test data set and acceptable fitting time.

5. Experiments/ Evaluation

5.1 Data collection

It is always a good practice to know the distribution of data before using it. To check if the property data make sense, a histogram of data is given in the figure below to show the distribution of the prices in the studied area. By looking at the distribution graph, the data points with extreme high prices are detected and removed. The final house prices are constrained in a reasonable range.

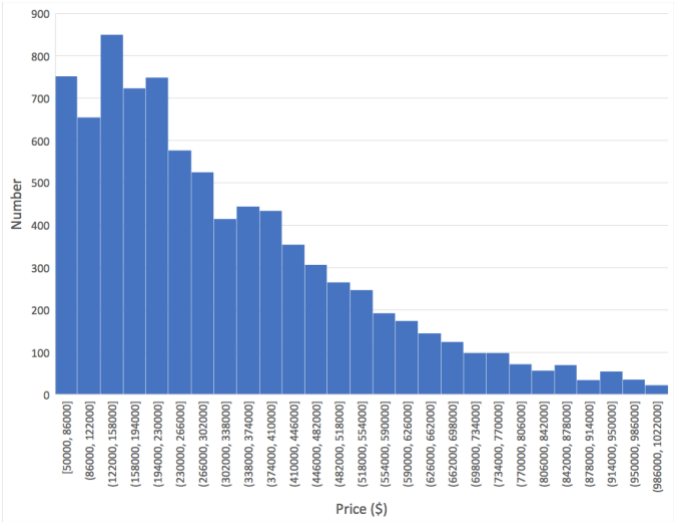

Figure 2. Distribution of price data

Local centers that are chosen are plotted in **Figure 3**. Note that these are local centers, more than merely supermarkets or cinemas. Some shops and restaurants only operate at certain local centers. They are all within the Interstate 285.

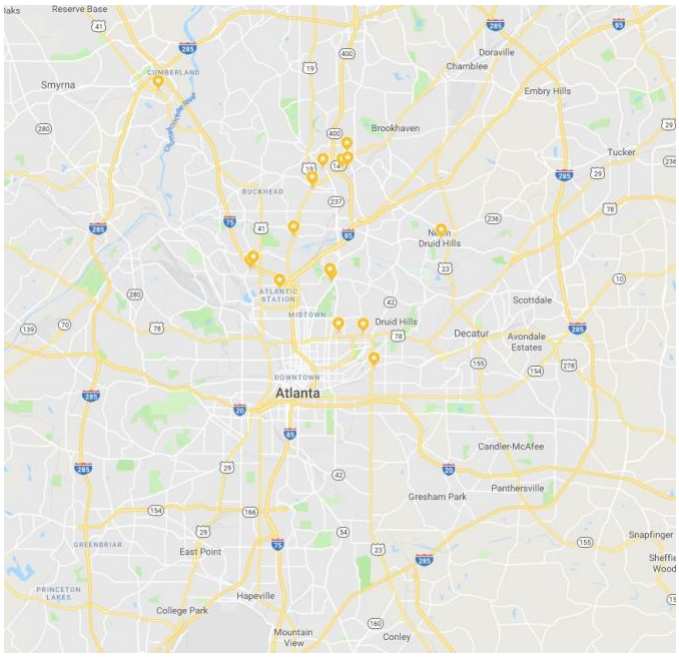

Figure 3. Location of local centers

Average house price for each zip code is also visualized in a stand-alone visualization tool (not incorporated in the website) using d3 and topojson. Figure on the left shows the visualized price. Circle in red roughly depicts Interstate 285, known as the Perimeter. Regions in white are regions where there is no price information. This house price visualization tool (not the website) can also shows the name of the region and average house price when mouse is hovering over the region, shown on the right.

Figure 4. Distribution of price data

5.2 Front-end

To make sure that every part in the front-end works fine, different levels of testing have been done involving checking the functionalities of the webpage elements and the associated JavaScript code.

Unit testing:
To ensure that all functions are working as they are expected, unit tests are written to cover the codebase as much as possible, and the testing framework Mocha is employed to run the test functions.

Integration testing:
Integration tests ensure that different units work together correctly. In this project, the functionality of interactive map and d3 visualization modules, as well as the integrity between these two components are particularly investigated in a set of integration tests.

5.3 Back-end

To ensure that back-end works fine, tests are carried out. For abnormal inputs, the back-end is able to recognize and respond with proper warning messages, rather than breaking down directly.

5.4 Machine Learning

5.4.1 Testbed: questions to answer

For the price prediction model, the predicted prices are used to validate our design. Some questions are:

- Which model is the best fit for house price prediction?
- Which parameter set for the model predict the most accurately?
- How does traveling time affect price prediction?

5.4.2 Model selection for house price prediction

To compare the accuracy of different machine learning models, each model was first tuned to find the best parameter set. Grid search cross validation was used and parameters with largest mean test score was chosen as the best set. The tuned parameters and their options are listed in Table 1. The tuning results including best parameter, best score and fitting time are also listed in the same table.

Among the tested models with tuned parameter, Random Forest has the best score of -0.20 and each fitting could be done within a few seconds. The score of Random Forest is much better than others. Best parameters for Random Forest are 'max_features': 'sqrt', 'min_samples_split': 2, 'n_estimators': 150, 'max_depth': 20. We also tested larger n_estimators and max_depth, and found no improvement.

5.4.3 Evaluation of including traveling time feature

We used two methods to validate that the traveling time information can improve the prediction performance. One is to compare the Random Forest cross validation score with/without traveling

time in the dataset, while using the best parameters from grid search with traveling information. The score for model without traveling time is 0.77, while the score for model with traveling time is 0.80. By adding the traveling information, the error decreased by 13%.

Another method is by analyzing the feature importance in the trained forest model. Figure 3 shows the importance of each feature. The most important five features are area, d11, d18, latitude, zip code. The traveling distance features for 20 centers are marked by 'd' following the center index. From the results, it's obvious that the distance between house and center 11 & 18 is helpful to predict prices.

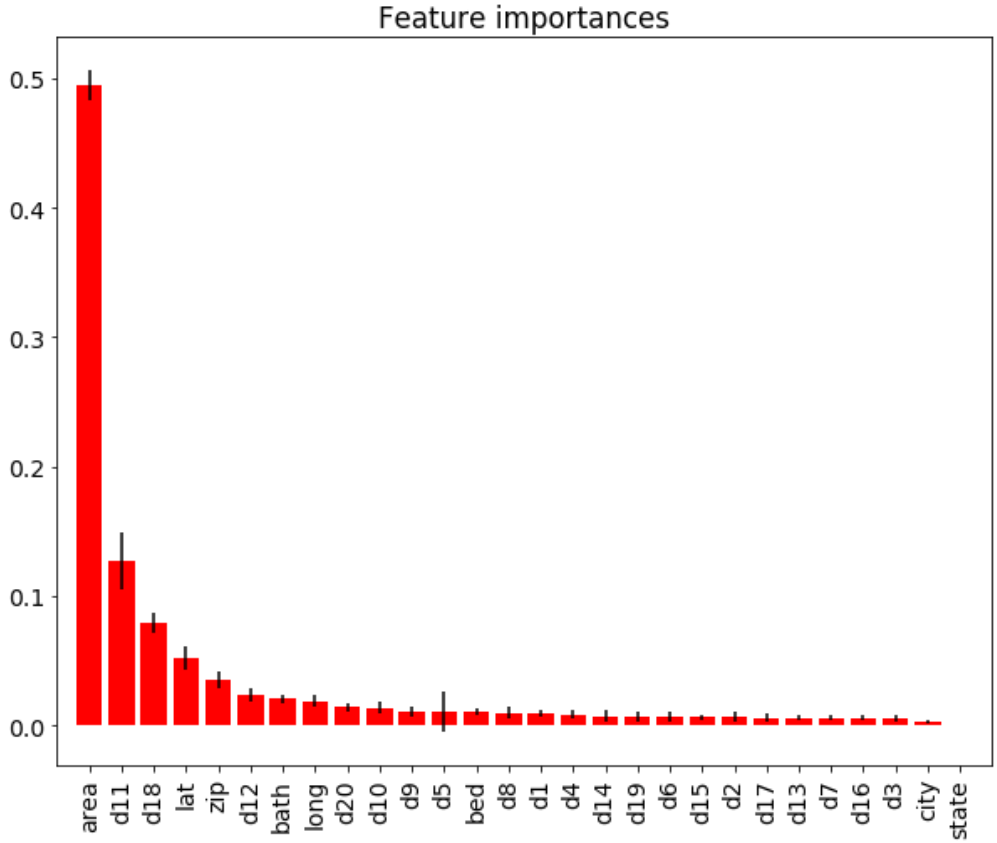

Figure 5. Feature importances for Random Forest model

Table 1. Model tuning for price prediction

	Random Forest	Neural Network	SVC	Ridge
Tuned Parameters	'n_estimators':[50,100,150], 'max_depth':[10,15,20], 'max_features':['auto','sqrt','log2'], 'min_samples_split':[2,10,20]	'hidden_layer_sizes':[(2,),(20,),(50,),(100,)],'alpha':[0.0001,0.01,1.0,100]	C':[0.001,0.1,10]	'alpha':[0.0001,0.001,0.01,0.1,1,10,100]

Best Parameter	'max_features': 'sqrt', 'min_samples_split': 2, 'n_estimators': 150, 'max_depth': 20	'alpha': 0.01, 'hidden_layer_sizes': (100,)	C:10	'alpha': 0.0001
Best Score (Negative Mean Relative Difference)	0.80	0.54	0.71	0.62
Mean Fitting Time	5.936	57.64	32.89	0.029

*NOTE that the scoring is using method in Section 4.4.3.

6 Conclusions and discussion

In this project, an interactive web application was developed to assist people find houses that meet their requirements. Specifically, the application could present travel information at the same time predicting house prices. Multiple super shopping centers were chosen as 'city centers' and the distances between houses to city centers were included as features in training dataset.

Parameters were tuned for different models including Random Forest, Neural Network, Ridge Linear Regression and SVC. Taking the best parameters for each model, Random Forest provide the most accurate prediction. Random Forest with the best parameters was trained and serves as prediction model for the application.

7 Distribution of team member effort

Plan of activities are shown in table 2. All team members contributed similar effort.

Table 2. Working Distribution

Group Member	Effort
Zichen Wang	ML, Data Cleaning
Wenqing Shen	ML training and model selection, Final report
Yixing Li	Data Cleaning, Back-end, Code Coordination
Dong Gao	Data Cleaning, Scraping, Final report
Xiangyi Yan	Data Visualization
Shahrokh Shahi	Front-end

References:

1. Muth R. *Cities and housing: The spatial patterns of urban residential land use.* University of Chicago, Chicago. 1969;4:114-23.
2. Bogin, A., W. Doerner, and W. Larson, *Local house price dynamics: New indices and stylized facts.* Real Estate Economics, 2016.
3. Glaeser, E.L., et al., *Housing dynamics: An urban approach.* Journal of Urban Economics, 2014. **81**: p. 45-56.
4. Horozov, T., N. Narasimhan, and V. Vasudevan, *Location-based recommendation system*. 2006, Google Patents.
5. Park, M.-H., J.-H. Hong, and S.-B. Cho. *Location-based recommendation system using bayesian user's preference model in mobile devices*. in *International Conference on Ubiquitous Intelligence and Computing*. 2007. Springer.
6. *Bayesian network.* February 28, 2018]; Available from: https://en.wikipedia.org/wiki/Bayesian_network.
7. Friedman, N., D. Geiger, and M. Goldszmidt, *Bayesian network classifiers.* Machine learning, 1997. **29**(2-3): p. 131-163.
8. Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003). Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *Journal of Chemical Information and Computer Sciences,43*(6), 1947-1958. doi:10.1021/ci034160g
9. Wiesmeier, M., Barthold, F., Blank, B. et al. Plant Soil (2011) 340: 7. https://doi.org/10.1007/s11104-010-0425-z
10. Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Applications to Nonorthogonal Problems. *Technometrics,12*(1), 69-82. doi:10.1080/00401706.1970.10488635
11. Marquardt, D., & Snee, R. (1975). Ridge Regression in Practice. The American Statistician, 29(1), 3-20. doi:10.2307/2683673
12. Odom, M., & Sharda, R. (1990). A neural network model for bankruptcy prediction. *1990 IJCNN International Joint Conference on Neural Networks*. doi:10.1109/ijcnn.1990.137710
13. Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology,49*(11), 1225-1231. doi:10.1016/s0895-4356(96)00002-9
14. Death, G. (2007). Boosted Trees For Ecological Modeling And Prediction. *Ecology,88*(1), 243-251. doi:10.1890/0012-9658(2007)88[243:btfema]2.0.co;2
15. R. Sirpal (*1994*) Empirical Modeling of the Relative Impacts of Various Sizes of Shopping Centers on the Values of Surrounding Residential Properties. Journal of Real Estate Research: 1994, Vol. 9, No. 4, pp. 487-505.