House Price Prediction Using Machine Learning
COMP 562: Introduction to Machine Learning, Final Project (Spring 2023)

Jad Jarkas, Reshmasai Malleedi, Janet Mbugua, Samuel Stone

Department of Computer Science, University of North Carolina at Chapel Hill

May 9, 2023

Abstract

The real estate industry is a complex and ever-evolving sector that requires accurate and efficient methods for predicting house prices. In recent years, the use of machine learning models for this purpose has gained popularity, as they can leverage both numerical and location data to accurately predict house prices. In this report, we present our findings from an investigation into the use of various machine learning models for predicting house prices. The random forest regression model was found to be the most accurate with a mean absolute percentage error of 5%.

1. Introduction

Our study focuses on the effectiveness of these models in accurately predicting house prices and identifying factors that affect the value of a property. We evaluate several machine learning models, such as Support Vector Machines (SVMs), Random Forest Regression, and others using various metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared. Our report provides insights into the potential impact of these models on the real estate industry, as well as the challenges associated with their development and deployment. We argue that the use of machine learning models for house price prediction has the potential to revolutionize the real estate industry, and our study provides a valuable contribution to the growing body of research in this area.

1.1. The Data

The 'House Prices and Images- SoCal' dataset is a comprehensive dataset that includes numerical and string data of residential properties in Southern California. The dataset contains over 15,000 observations, with features such as the number of bedrooms, bathrooms, square footage, lot size, year built, and sales price. The address of each entry is stored in Street-City format as string entries, unlike the previously mentioned columns, which are composed of floats.

1.2. Application and Motivation

Predicting house prices can be useful for homeowners, real estate companies, and investors. Homeowners can use predictive models to estimate the value of their property and make informed decisions about selling or renovating their home. Real estate companies can use predictive models to estimate the value of properties accurately and make informed decisions about buying and selling properties. Investors can use predictive models to identify undervalued properties and make informed investment decisions. Overall, predicting house prices can improve decision-making, save time and resources, and identify trends and patterns in the real estate market.

2. Literature Survey and Related Works

1. **Paper**: *Housing Price Prediction using Machine Learning Algorithms: The Case of Melbourne City, Australia.* [1]
   Description: House price forecasting is an important topic of real estate. The literature attempts to derive useful knowledge from historical data on property markets. Machine learning techniques are applied to analyze historical property transactions in Australia to discover useful models for house buyers and sellers.

2. **Paper**: *Real Estate Value Prediction Using Linear Regression.* [2]
   Description: The study of real estate value is felt critical to help the choices in urban arranging. The land framework is a precarious stochastic process. Financial specialists' choices depend on available patterns to procure the most extreme returns. Designers are intrigued by knowing the future patterns for their basic

leadership. To precisely gauge real estate costs what's more, future patterns, a vast measure of information that impacts arrival cost is required for examination, demonstrating, and determining.

3. **Paper**: *A Hybrid Regression Technique for House Prices Prediction.* [3]
   Description: How to use machine learning algorithms to predict house prices? It is a challenge to get as close as possible results based on the model built. A specific house price is determined by location, size, house type, city, country, tax rules, economic cycle, population movement, interest rate, and many other factors which could affect demand and supply. For local house price prediction, there are many useful regression algorithms to use. For example, support vector machines (SVM), Lasso (least absolute shrinkage and selection operator), Gradient boosting, Ridge, and Random forest.

3. Approach

3.1. Data Processing

Once the dataset is loaded, it must be cleaned and processed to gain a better understanding of the variables and information the dataset provides. Data cleaning and processing is an essential step in the machine learning pipeline as it allows the data to be transformed into a format that is suitable for training a machine learning model.

We trained our models on Google Colaboratory using a free GPU hardware accelerator. This offered much less computational power than required for training other models to predict house prices. Thus, our main challenge was to build and train an accurate model under the constraint of low computational power.

Here is a brief summary of the steps we followed:

1. **Data Pre-processing**: we categorize the features depending on their datatype (int, float, object) and then calculate the number of them.

2. **Exploratory Data Analysis**: we deeply analyze the data so as to discover different patterns and spot anomalies.

3. **Data Cleaning**: we improvise the data and remove incorrect, corrupted or irrelevant data.

4. **Data Conversion**: we alter data stored as string values into numerical data.

5. **One Hot Encoder**: we convert categorical data into binary vectors.

6. **Training/Testing Split**: we split the dataset into training and testing data following the split of our X and Y variables; Y being the 'price' of the house and X being the rest of the columns.

7. **Modeling**: we train different models to determine the continuous values, in the case of regression models.

8. **Accuracy**: to calculate loss we will be mainly using the 'mean absolute percentage error' module from the sklearn library. The formula for Mean Absolute Error is:

$$\sum_{i=1}^{D} |x_i - y_i|$$

3.2. Models

Regression models, such as Support Vector Machines (SVMs), Random Forests, and Linear Regression, were chosen for the development of machine learning models for house price prediction due to their ability to handle continuous target variables, such as house prices. SVMs are a popular choice for regression tasks as they can handle both linear and non-linear relationships between input variables and target variables. Random Forests are another popular choice as they are capable of capturing non-linear relationships and interactions between features. Linear Regression is a simple yet powerful regression model that can provide insight into the relationships between input variables and target variables. The choice of these models was based on their effectiveness in handling regression tasks and their ability to capture complex relationships between features and target variables. By using a combination of some of these models, researchers can create more robust machine learning models for house price prediction that can provide valuable insights for stakeholders in the real estate industry.

3.2.1. Random Forest Regression Model

We used the Scikit-learn library to build and evaluate a Random Forest Regression (RFR) model. The dataset we use for this model is with a one-hot encoder, a technique to convert categorical variables into numerical data. We start by initializing a Random Forest Regression model with 10 decision trees, specified by the 'n_estimators' parameter. The 'X_train_OH' represents the feature matrix and 'Y_train_OH' represents the target variable. We then fit the Random Forest Regression model to the training data, where 'X_train_OH' represents the feature matrix and 'Y_train_OH' represents the target variable. Finally, we generate predictions on the validation set ('X_valid_OH') using the trained model, and store them in 'Y_pred_OH'.

We evaluate the accuracy of the model by its mean absolute percentage error between the predicted and actual values for the validation set, once again. The result of

approximately 0.05 indicates that the average percentage difference between the predicted and actual house prices is 5%. In other words, on average, the predicted house prices are off by 5% of the actual house prices. This means that the Random Forest Regression model is highly accurate in predicting house prices.

3.2.2. Gradient Boosting Machine Model

We used the Scikit-learn library to evaluate a Gradient Boosting Regressor to predict the prices of houses based on quantitative data provided in the given dataset. This process is achieved by using the correlation matrix to identify the strongest correlations between the other quantitative columns. In the case of this given data, the column that had the strongest correlation coefficient with the price column was the 'sqft' column at 0.583457.

We then utilized Scikit-Learn libraries' transformer function called the OneHotEncoder which takes the integer columns and converts them into vectors. We then regulate the scales of the numerical values by normalization which is min-maxing the numerical attributes. This was achieved by using the Scikit-learn library's StandardScaler function for this process. To aid in the process in which we transformed the data, we are also using the Pipeline class provided in the same library. This is commonly used to make two Pipeline objects that keep all current transformations in order as we change them.

After all this preparation, we then test the data to evaluate the preciseness of the model, taking the mean squared error (MSE) with each prediction which comes to a score of roughly 80% accuracy.

3.2.3. Linear Regression With Geopy Point Data

Utilizing Skickit-learn again, we experiment with converting address data to numerical data, as the Scikit-learn LR model is not designed to accept string-type inputs. To this end, we utilized Nominatim, an open-source tool to search OpenStreetMap (OSM) for numerical data related to addresses through a process known as geocoding.

Two issues arose from this approach, however. Firstly, Nominatim limits the rate at which requests are accepted by the server. To avoid being timed out, we limited address requests to a rate of 1/second. This meant that obtaining the information for 15,000 addresses took 4.3 hours. Secondly, we realized Nominatim is not a fully comprehensive dataset, as a significant percentage of the dataset (approximately 23% of addresses) was given a return value of None. Manual searches confirmed that Nominatim does not contain latitude/longitude data for many physical locations. Both of the issues above raised question as to the scalability of the model, as well as real-world applicability. The end result of a 37% mean percentage error (a mere 3% decrease compared to the previous Linear Regression model) did not justify itself in our eyes, and we moved on to other approaches.

3.2.4. Linear Regression Model

We used the Scikit-learn library to build and evaluate a Linear Regression (LR) model. The dataset we use for this model is not with a one hot encoder, but rather without the categorical variables present in the dataset. We start by initializing a Linear Regression model and then fitting it to the training data, where 'X_train_OH' represents the feature matrix and 'Y_train_OH' represents the target variable. Finally, we generate predictions on the validation set ('X_valid_OH') using the trained model, and store them in 'Y_pred_OH'.

We evaluate the accuracy of the model by its mean absolute percentage error between the predicted and actual values for the validation set. The result of approximately 0.40 indicates that the average percentage difference between the predicted and actual house prices is 40%. In other words, on average, the predicted house prices are off by 40% of the actual house prices. This means that the LR model may not be very accurate in predicting house prices.

3.2.5. Support Vector Machine (SVM) Model

We used the Scikit-learn library to build and evaluate a Support Vector Regression (SVR) model. The dataset we use for this model is with a one-hot encoder, a technique to convert categorical variables into numerical data. We start by initializing an SVR model using the default hyperparameters. We then fit the SVR model to the training data, where 'X_train_OH' represents the feature matrix and 'Y_train_OH' represents the target variable. Finally, we generate predictions on the validation set ('X_valid_OH') using the trained model, and store them in 'Y_pred_OH'.

We evaluate the accuracy of the model by its mean absolute percentage error (MAPE) between the predicted and actual values for the validation set. The result of approximately 0.45 indicates that the average percentage difference between the predicted and actual house prices is 45%. In other words, on average, the predicted house prices are off by 45% of the actual house prices. This means that the SVM model may not be very accurate in predicting house prices.

3.2.6. Decision Tree Model

This model uses the Scikit-learn library to build and evaluate a decision tree. This decision tree was trained to predict if a given house was worth 1 million dollars or more. Data entries are assigned "Yes" if the price of the home is equal to or greater than 1 million dollars. Otherwise, the entries were assigned "No."

We start by initializing a decision tree with a maximum depth of 8, a minimum of 6 sample leafs, and 101 random states. Then we start fitting it to the training data, where 'X_train' represents the feature matrix and 'Y_train' represents the target variable. Finally, we generate predictions on the testing set ('X_test') using the trained model and stores them in the 'Y_pred' variable. We use the 'accuracy_score' function to compare the results of the 'y_pred' and 'Y_test' variables. The returned value of 89.66% represents the accuracy score.

3.2.7. Decision Tree Regression Model

This model uses the Scikit-learn library to build and evaluate a decision tree regression model. This model was trained to predict if a given house was worth 1 million dollars or more. Data entries are assigned a value of 1 if the price of the home is equal to or greater than 1 million dollars. Otherwise, the entries were assigned a value of 0.

We start by initializing a decision tree regression model with a maximum depth of 8, a minimum of 6 sample leaves, and 101 random states. Then we start fitting it to the training data, where 'X_train_b' represents the feature matrix and 'Y_train_b' represents the target variable. Finally, we generate predictions on the testing set ('X_test_b') using the trained model and store them in the 'Y_pred_dt' variable.

We evaluate the accuracy of the model by its mean square error (MSE) and root mean square error (RMSE) between the predicted and actual values for the validation set. The MSE came out as 0.08301 which represents a model with low levels of error. A model with MSE of 0 is a perfect model so the closer the MSE is to 0 the better the model is, generally. The RMSE is 0.288115 which is a relatively acceptable mark for this metric. This means that 28% is the average rate of error.

4. Conclusion

The 'House Prices and Images- SoCal' dataset is heavily used in the predictions of house prices and the real estate industry. This paper aimed to address the benefits of different machine learning models to successfully predict the price of houses based on numerical data.

We first constructed several regression models and then tried incorporating other types of machine learning techniques in hopes of increasing our accuracy score. When comparing the data retrieved from the various models, we received varying results concerning the accuracy of the model to the given set of data. Our models also tested accuracy in different ways such as using the mean absolute percentage error and accuracy score functions and tried methods of adding other factors such as considering a model predicting prices of houses estimated to be over 1,000,000 dollars. Accuracy in our context means how close the predicted house price values were to the actual house prices.

Starting from the most to least precise predictions, the following was our approximate model results: Random Forest Regression with 95% accuracy and 5% error, Gradient Boosting Regression at 80% accuracy and 20% error. Linear Regression with Geopy Point Data with 63% accuracy and 37% error, Linear Regression with 60% accuracy and 40% error, Support Vector Machine with 55% accuracy and 45% error, Decision Tree Regression with 28% accuracy and 72 % error,

Despite the relatively high accuracy of our final model, the Random Forest Regression, there is room for further research and improvement. Since our models for house price prediction are heavily based on numerical data, we could potentially incorporate image classification based on the images of the exteriors of houses to further increase our prediction accuracy using convolutional neural networks.

5. Acknowledgements

We would like to thank Professor Jorge Silva and the Teaching Assistants for their instruction in COMP 562: Introduction to Machine Learning. We drew upon the foundational knowledge gained from the class to construct and train the machine-learning models in this paper.

6. References

[1] Phan, D. *Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia*. *2018 International Conference on Machine Learning and Data Engineering*, pp. 35-42, Sydney, Australia, 2018.

[2] Ghosalkar, N., Dhage, S., *Real Estate Value Prediction Using Linear Regression*. *2018 Fourth International Conference on Computing Communication Control and Automation*, pp. 1-5, Pune, India, 2018.

[3] Lu, S.,Li, Z., Qin, Z., Yang, X., Siow, R., and Goh, M. *A hybrid regression technique for house prices prediction*. *2017 IEEE International Conference on Industrial Engineering and Engineering Management*, pp.319-323, Singapore, 2017.