Predicting Housing Prices using Machine Learning Algorithms

Daniel Arantes Ventura
Faculty of Business
Humber College
Toronto, Canada
N01468881

Rutuja Bhagatsing Kadam
Faculty of Business
Humber College
Toronto, Canada
N01468983

Mariana Reyes
Faculty of Business
Humber College
Toronto, Canada
N01468759

Maria Paula Rodriguez
Faculty of Business
Humber College
Toronto, Canada
N01469279

Mughda Vivek Bhatwadekar
Faculty of Business
Humber College
Toronto, Canada
N0146889

***Abstract*— The housing market involves multiple stakeholders looking to maximize their real estate investments. This report aims to provide a framework to understand the influence of certain house features on the price and evaluate three machine learning algorithms to find a model that will accurately predict the value of a property. This paper will discuss the application of Multiple Linear Regression, K-Nearest Neighbors and Random Forest algorithms to a dataset containing historical sales data from residential properties in Washington, US. For every model implemented, this report will comprehensively show a performance and accuracy evaluation to recommend the best-performing solution.**

Keywords— House prediction; regression analysis; k-nearest neighbors, random forest, machine learning.

I. INTRODUCTION

The housing market refers to the number of residential properties available in the market for sale or purchase. It brings together multiple parties, such as homeowners, real estate investors, contractors, and brokers looking to maximize their investment in every property. With inflation and rent prices skyrocketing in the US, some experts advise people to invest in real estate properties as a hedge against inflation, as the home value tends to increase in response to other surrounding costs that are on the rise [1]. Given this scenario, predicting home prices is critical to take advantage of opportunities in the local market. It will provide potential homeowners and investors with a tool to determine the price of a house and the right time to buy it.

This report aims to provide a framework to accurately predict house prices by using predictive analytics techniques and machine learning regression algorithms. Along with the macro external attributes such as the state of the economy, interest rates and fluctuations of household income that influence house prices, there are several specific structural house attributes that define the value of a property. The focus of this research is to understand the influence of these common structural attributes that describe a property and generate a prediction taking into consideration the variables that strongly have an effect on the price.

This paper is organized under the following structure: the first section describes the data used in this research, followed by a brief discussion of the three machine learning models used to forecast the price. To finalize with a discussion of the results and comparison of the three methods concluding with the one that is more suitable to fulfill our prediction task.

II. UNDERSTANDING THE DATA

The data set used to conduct the house prices model contained historical data of houses located in Washington, US, between 2014 and 2015. Along with our variable target price, the data set contained eighteen house features, an Id for each house and the date on which the sale record was created. The dataset initially contained 21.613 data points. Table I provides a description of the variables.

TABLE I. VARIABLES DESCRIPTION

VARIABLE	DESCRIPTION	VARIABLE	DESCRIPTION
Id	Unique key for each house	Condition	House condition (Values:1 to 5)
Date	Record created date	Grade	House grade (Value:1 to 13)
Price	House price	Sqft_above	Square footage above ground
Bedrooms	Number of bedrooms	Sqft_basement	Basement square footage
Bathrooms	Number of bathrooms	Yr_built	Year in which the house was built
Sqft_living	Square footage of living area	Yr_renovated	Year in which the house was renovated
Sqft_lot	Square footage of lot	Zip code	Zip code – Location
Floors	Number of floors	Lat	Latitude
Waterfront	Access to body of water (dummy variable)	Long	Longitude
View	Type of view (Values: 1 to 4)	Sqft_living15	Updated square footage of the house.

After exploring the data using statistical methods and functions provided in the library pandas, we found that the data did not contain any missing values; it had duplicates that were further eliminated. Regarding outliers, max and min values were analyzed for each variable and then, using boxplots, several outliers were detected in the price, bedrooms, bathrooms, sqft_living, sqft_above and sqft_basement variables. For the purpose of this research, outliers were removed to avoid potential bias in the research.

III. DISCUSSION

A. Initial Variable Selection

After data exploration and cleaning, an initial feature selection was conducted to simplify the following processes.

As already mentioned, the original dataset contained three date columns: record date, construction date, and renovation date. As previously executed by Sharma (2021) [2], the age of the house was computed by subtracting the record year and the construction year, adding a new feature to the dataset. The same process was followed with the renovation year, adding the renovation age as a feature. The record date, construction date, and renovation date were removed from the dataset after obtaining the house age and renovation age. The reason behind this procedure is that variables computed as a number of years are numeric and more suitable for regression analysis.

Furthermore, interaction among dependent variables was reviewed to detect and eliminate variables with high correlation. Pearson coefficient was revised in a range of 0.8 to 0.95 (which typically indicates a strong correlation). None of the variables were correlated to this degree. Thus, none were removed through this method.

B. Regression Analysis

The first machine learning algorithm to predict housing prices in this data set was Multiple Regression. As explained by Aljohani (2021), Multiple Regression is a variation of Linear Regression in which one dependent variable (y) is a function of more than one independent variable (x). [3] The model can be described by the following equation (1):

$$y=b0+b1*x1+b2*x2+b3*x3 \ldots \quad (1)[3]$$

Multiple Regression is suitable for the characteristics of the dataset. The model can predict a numeric variable (in this case, housing price) using a combination of numeric and categorical variables.

The following diagram (Fig.1) illustrates the steps taken in this revision for the implementation of Multiple Regression:

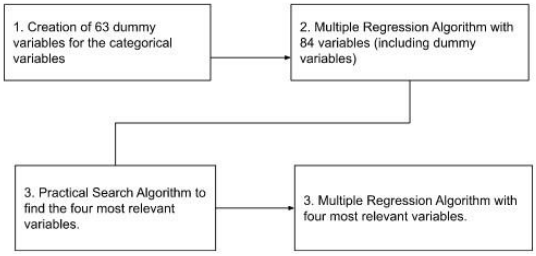

Fig.1 Multiple Regression Algorithm Flow Chart

As illustrated in Fig.1, the first step before executing the Multiple Regression algorithm was creating dummy variables for the categorical variables in the dataset. The dataset contains two categorical variables: waterfront and zip code. The first step consisted on creating 63 dummy variables to account for the possibility of a waterfront view plus all the zip code possibilities. This procedure yielded 84 variables in total, an increase of more than 60 versus the initial count.

Next, the first iteration of the Multiple Regression algorithm was carried out using all 84 variables. The dataset was partitioned into a training set and a validation set where the training set accounted for 60%, as previously done by Verma, et. al. (2022) [4] in a similar machine learning study predicting housing prices in India through Multiple Regression

Following the first iteration, the Forward Selection practical search algorithm was utilized to identify the most significant variables in the model. According to this algorithm, the four specifications that appear to be the most important for predicting housing prices are grade, lat, age, and sqft_living15.

Forward Selection was the chosen search method for feature reduction because, as mentioned in a bibliographical review conducted by Kotsiantis (2011), there are few to no advantages against Backward Selection. In this same study Kotsiantis (2011), highlights that Stepwise Selection is the best method among these three, because it accounts for more combinations or possibilities [5]. In the case of the present study, Stepwise Selection yielded the same results as Forward Selection.

After defining the four most relevant features, a second multiple regression iteration was performed utilizing only those four variables (grade, lat, age, sqft_living15) as independent inputs. Once again, the dataset was divided into a training set accounting for 60% training data and 40% validation data as done by Verma, et. al. (2022) [4]. To test the second iteration, the price of two houses in the data set was estimated using the model.

To assess the models, summary statistics and adjusted R-square were calculated for both iterations. Results for multiple regression iterations and the price estimation computed for two houses in the model are discussed in the following section.

C. K-Nearest Neighbors

According to the author Shmueli et tal. (2019), predicting a categorical outcome refers to a classification model. On the other hand, when predicting a numerical outcome, it refers to a regression model. This paperwork section intends to discuss the k Nearest Neighbors algorithm, which can predict both types of outcomes, categorical or numerical, depending on the dataset. The prediction is based on the closest neighbours of focal data, and the number of these neighbours is represented by the letter k.

In like manner, the nearest point of the focal data indicates the similarity between these points. After calculating the k, the model can recognize classes of data, which means that each class has a k number of points referring to the closest similarity [6]. Moreover, the most common equation (2) to calculate the nearest neighbors is the Euclidean distance:

$$distance = \sqrt{(x_1^* - x_1)^2 + (x_2^* - x_2)^2 + \cdots + (x_d^* - x_d)^2} \quad (2)[7]$$

To find the k value, the first step is to subset the dataset into train and validation, separating features and targets. It is necessary to preprocess the train features subset, which means standardizing the data. Note that the validation feature subset also must be standardized to be used further in the k calculation.

For this present work, the technique used was GridSearchCV from Scikit Learn, as indicated by the author Albon (2018). Then, a k-NN classifier was created using the algorithm also from Scikit Learn. Also, it is created as a pipeline to refer to the classifier and the standardized train subset. This pipeline was input in the grid search, and finally, the model found the k that produces the best model [7].

In this case, k = 2. Consequently, the algorithm was run again, considering the new nearest neighbours [7]. The result was a predictor. At this point, the work was to run the predictor with the validation subset and compare the numbers retrieved. The model evaluation numbers will be brought up later in this paper.

D. Random Forest Regression

A supervised learning algorithm called Random Forest Regression implements ensemble learning, a technique that combines predictions from various machine learning algorithms to provide predictions that are more accurate than those from a single regression model (Breiman, 2011) [8].

Random Forest is a collaborative technique capable of performing regression and classification tasks using multiple decision trees and a practice called Bootstrap and Aggregation, also known as bagging. Instead of depending solely on individual decision trees to determine the final output, the main idea behind this is to aggregate several decision trees (Breiman, 2011) [8].

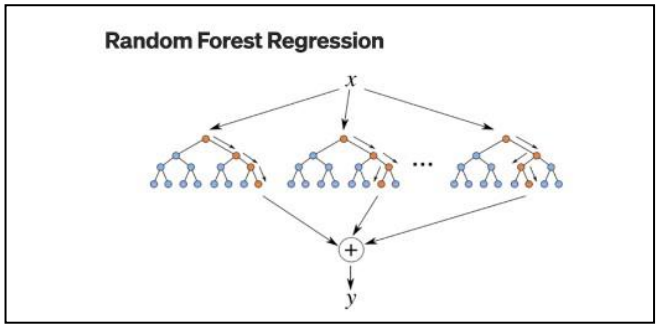

Fig.2 Random Forest Regression Model Overview. [9]

The Fig.2 shows the structure of a Random Forest. The trees run in parallel without any interaction. During training, a Random Forest builds many decision trees and outputs the mean of the classes as the prediction of all the trees (Breiman, 2011) [8].

The following steps serve to better understand the Random Forest algorithm: [8]

- Each tree is created from a different sample of rows and at each node, a different sample of features is selected for splitting.
- Each of the trees makes its own individual prediction.
- These predictions are then averaged to produce a single result.

A Random Forest's accuracy and overfitting are improved over a single Decision Tree because of the averaging process. A Random Forest Regression model is effective and precise. On many issues, including those involving non-linear relationships, it typically delivers excellent results [8].

For the purpose of this study, Random Forest Regression was computed two times. The first iteration includes 16 predictors and one outcome (price). However, the second iteration includes only four predictors, resulting in the most relevant features in the Forward Selection algorithm discussed in the previous section. The model was executed two times in order to be able to compare performance results and metrics between them and with the other models that are the subject to this study. Both iterations were performed on 1,000 entries. The dataset was partitioned into a training set and a validation set, where the training set accounted for 80%. This data was used to initialize the Random Forest Regression model.

As mentioned by Srivastava (2020) [10], the estimator of the Random Forest Regression model is the number of trees required before calculating averages of predictions. Although higher estimator numbers make the algorithm more complex, the higher the number of predictions, the better the performance. Srivastava (2020) [10] recommends using two estimators, 1 and 50, to confirm model stability. For the purpose of this study, the model was run with both estimators (1 and 50) to compare results.

The performance metrics of both iterations of Random Forest Regression are presented in the following section.

IV. RESULTS

A. Regression Analysis

As discussed in the previous section, two iterations of Multiple Regression were executed. One iteration was carried out with the complete set of variables and the second iteration was executed with only the four most relevant variables found through the practical search Forward Selection algorithm. In Table II, the results for both iterations are presented.

TABLE II. MULTIPLE REGRESSION PERFORMANCE METRICS

	Multiple Regression First Iteration		**Multiple Regression Second Iteration**	
	Training Set	***Validation Set***	***Training Set***	***Validation Set***
Mean Error (ME)	0.0	-259.8	0.0	-39.6
Root Mean Squared Error (RMSE)	79132.5	79984.9	122564.0	122659.9
Mean Absolute Error (MAE)	58397.3	59014.7	93327.3	92964.2
Mean Percentage Error (MPE)	-2.4%	-2.6%	-6.6%	-6.6%
Mean Absolute Percentage Error (MAPE)	14.0%	14.2%	22.6%	22.5%
Adjusted R-squared	83.8%	83.2%	62%	60.9%

Firstly, by comparing the performance metrics of the training set and validation set in the first iteration, it is evident that the model is capable of maintaining the same performance across both sets. For instance, the Mean Absolute Percentage Error (MAPE) for the training set is 14% versus 14.2% in the validation set. The same is also true for the performance metrics in the second iteration, where the results are similar between the training and validation set.

However, there are significant differences in the performance metrics between iterations. The model in the first iteration is more statistically accurate in predicting the price using 84 independent variables versus the model in the second iteration with only the four most relevant variables. The MAPE for the first iteration is 14%, while the same metric for the second iteration is 22.5%. Likewise, the adjusted R-squared for the first iteration is 83.3%, while for the second iteration is 60.9%. The result in this metric means that the first model is capable of explaining 83.8% percent of the variance in price, while the second model accounts for only 60.9% of the variance in price.

Besides calculating the performance metrics for the models, both models were executed to predict the price of two houses in the validation portion of the data set. The results are shown in Table III. Although it is difficult to draw insights from only two data points, the two values predicted by the first model are closer to the actual value than those predicted by the second model. To further understand the difference, 100% of the data points in both validation sets were utilized to predict housing prices using the models, and the residual between the actual value and the predicted value of every house was calculated. Results show that the standard deviation of the residuals in the first model is $79,990.2, which is smaller than the standard deviation of the residuals calculated for the second model which is $122,668.7. Even though there seems to be less variation in the predictions of the first model, the standard deviation among the residuals is still more than the minimum price in the data set ($78,000). Therefore, the error of both models can be higher than the price of a house in the data set.

TABLE III. HOUSING PRICE PREDICTION WITH MULTIPLE REGRESSION

Multiple Regression First Iteration			
Index	Predicted Value	Actual Value	Residual
13585	397,720.6	389,999	-7,721.6
2103	536,464.5	560,000	23,535.4
Multiple Regression Second Iteration			
219	370,682.4	285,000	-85,682.4
10181	267,416.5	149,500	-117,916.5

B. K-Nearest Neighbors

As explained on this model section, it was needed to run the model a couple of times. Initially, there was the necessity to find the k value, which was performed by running the model with a chosen k, and then, the outcome was the best k = 2. Subsequently, a second iteration took place considering the found k.

TABLE IV. REGRESSION STATISTICS COMPARING VALIDATION AND PREDICTION TARGETS

	First Iteration (k=5)	**Second Iteration (k=2)**
Mean Error (ME).	84,916.54	43,693.39
Root Mean Squared Error (RMSE).	141,722.70	123,017.53
Mean Absolute Error (MAE) .	102,478.53	85,991.51
Mean Percentage Error (MPE).	15.75	6.33
Mean Absolute Percentage Error (MAPE).	21.51	18.80

The statistics displayed in Table IV refer to the comparison of validation and prediction values for the target (price). Notably, the statistics are favourable for the second iteration, which ratifies the best k=2. For instance, the mean percentage error for the second iteration is less than half of the first iteration, which is k=5

C. Random Forest Regression

As previously discussed, the algorithm for Random Forest Regression was carried out two times, each time with different predictors. For the first iterations all the predictors in the data set were used. For the second iteration only the four most relevant predictors according to Forward Selection were used. Each iteration was repeated two times, with estimator 1 and 50 to assess model stability. The results for all iterations and repetitions are shown in Table V and VI.

TABLE V. RANDOM FOREST REGRESSION MODEL FIRST ITERATION

	Estimator	
	1	50
MAE	93,052.6	61,757.4
MSE	19,260,000,000	8,441,000,000
RMSE	138,776.4	91,874.5
R^2	0.5	0.78
Adjusted R^2	0.45	0.76
ME	-5,108.4	-2,792.48
MAPE	21.1%	14.7%

TABLE VI. RANDOM FOREST REGRESSION MODEL SECOND ITERATION

	Estimator	
	1	50
MAE	92,914.6	68,590.7
MSE	15,740,000,000	9,671,000,000
RMSE	125,475.2	98,340.7
R^2	0.59	0.75
Adjusted R^2	0.58	0.74
ME	2,072.3	-3.326.4
MAPE	21.3%	16.4%

The performance metrics show that for both iterations the error decreased when the estimator increased from 1 to 50, indicating a more accurate and stable model. For instance, the Root Means Squared Error decreased by approximately 30,000 - 40,000 when the estimator was 50.

When comparing both iterations, the performance metrics indicate that reducing the number of predictors to only four didn't drastically impact the performance. The adjusted R-squared decreased from 0.76 to 0.74 from the first iteration to the second one. The variation in MAPE was less than 2% from one iteration to the other. Although there is a slight indication that error increases and adjusted R-squared decreases when reducing the predictors to four, the differences in performance are not considerable. Thus, executing this model with four predictors reduces complexity and yields approximately the same results.

V. CONCLUSION

Based on the results provided in the past section and shown in Table VII, it can be concluded that for the Multiple Regression model, the best performance results were obtained when 84 predictors (including 66 dummy variables) were used, yielding a MAPE of 14.2% and an Adjusted R-squared of 83.2%. The same is true for Random Forest Regression, where the iterations with all predictors yielded slightly better results. However, there is no considerable difference in the performance between the Random Forest Regression model with all predictor variables and the one with only four predictors. Thus, it can be concluded that for Random Forest Regression, it might be more convenient to use only four variables since it reduces complexity without considerably affecting performance.

When it comes to K-NN it can be concluded that the best k for the model is 2, since the MAPE was 6.3%, compared to 15.8% when k is 5.

When comparing the three models, the Multiple Regression and Random Forest Regression models obtained similar performance metrics in their best-performing iteration. Although metrics are similar, Multiple Regression has better results regarding RMSE, MAE, and MAPE. These results can be observed in the Fig.3, Fig. 4, Fig.5, respectively. In general, K-NN was the worst-performing model when compared to the other two.

Overall, in this study the Multiple Regression model is the best performing model statistically. However, the fitness of the performance of a model completely depends on the application and the acceptable error for such application. Although in this study the best performing models obtained a MAPE of approximately 14% this amount of error can account for the minimum price in this data set and might not be suitable for certain applications. Therefore, more information might be needed to emit a judgment in terms of how successful the model was in reaching the desired performance.

TABLE VII. MODEL PERFORMANCE COMPARISION

Model	**Multiple Regression**	**KNN**	**Random Forest**
ME	-2,182.92	43,693.39	-2,792.48
RMSE	80,047.79	123,017.5	91,874.49
MAE	58,908.61	85,991.51	61,757.45
MPE	-2.68	6.33	-4.57
MAPE	14.42	18.8	14.71
Adjusted R^2	83%	-	76%

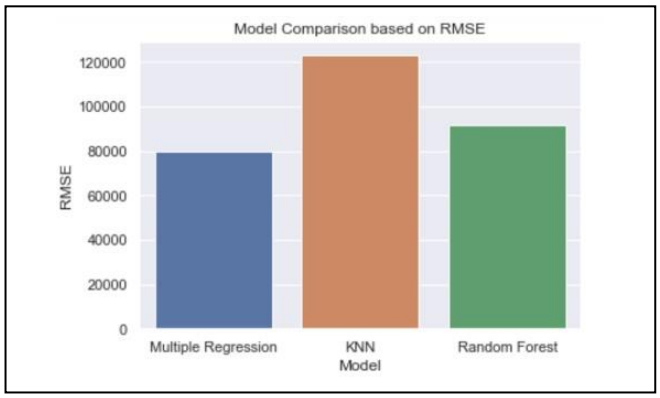

Fig. 3. Model Comparison based on RMSE

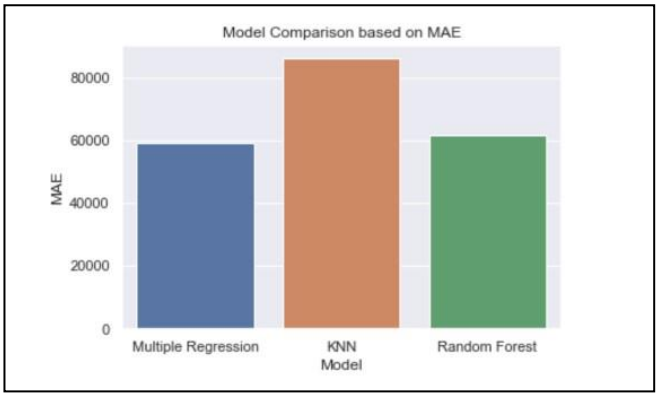

Fig. 4. Model Comparison based on MAE

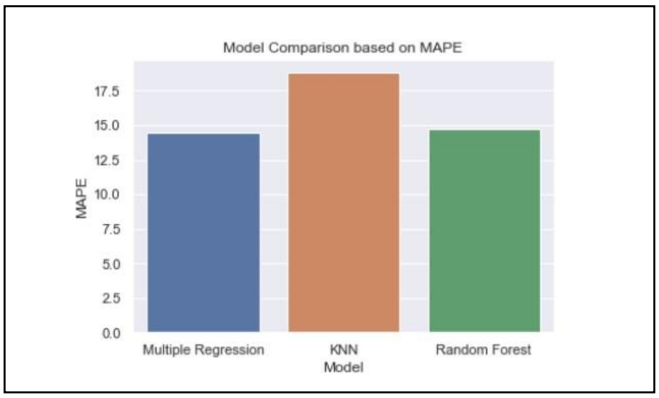

Fig. 5 Model Comparison based on MAPE

VI. REFERENCES

[1] Is Real Estate a Good Hedge Against Inflation? (2022, April 27)

[2] Sharma, S. (2021). House Prices Regression Predictive Analysis. House Prices - Advanced Regression Techniques..

[3] Aljohani, O. (2021, December). Developing a stable house price estimator using regression analysis. In The 5th International Conference on Future Networks & Distributed Systems (pp. 113-118).

[4] Verma, A., Nagar, C., Singhi, N., Dongariya, N., & Sethi, N. (2022, April). Predicting House Price in India Using Linear Regression Machine

[5] Kotsiantis, S. (2011). Feature selection for machine learning classification problems: a recent overview. Artificial intelligence review, 42(1), 157-176.

[6] Shmueli, G., Bruce, P. C., Gedeck, P., & Patel, N. R. (2019). Data Mining for Business Analytics: Concepts, Techniques and Applications in Python (1st ed.). Wiley.

[7] Albon, C. (2018). Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning (1st ed.). O'Reilly Media.

[8] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32

[9] RandomForestRegression. (2020, June 8). Level up Coding.

[10] Srivastava, T. (2015). Tuning the parameters of your Random Forest model. Analytics Vidhya, 9