Predictive Modeling and Analysis of Housing Prices

Puja Saha
University of Guelph
Engineering Department
psaha03@uoguelph.ca

Parya Abadeh
University of Guelph
Engineering Department
pabadeh@uoguelph.ca

Om Bhosale
University of Guelph
Engineering Department
obhosale@uoguelph.ca

Sanchi Sanchi
University of Guelph
Engineering Department
ssanchi@uoguelph.ca

***Abstract*— One of the fundamental needs for individuals worldwide is housing and accommodation. However, the challenge of acquiring a home has become prevalent due to financial constraints in various countries. This study addresses the difficulties faced by individuals globally in finding and purchasing a house. Our objective is to enhance predictive accuracy for house prices by employing various regression techniques, including Random Forest, KNN, Linear Regression, Lasso Regression, Gradient Boosting Regression, Support Vector Regression and Ridge Regression. This research aims to provide valuable insights that can benefit prospective buyers and real estate advisors in comparing properties or making informed decisions on new acquisitions. Additionally, the study considers pertinent factors influencing housing costs, such as physical conditions, concept, and location.**

Keywords—regression, house prediction, home, price.

I. INTRODUCTION

This report explores the latest research predictions and trends in conjunction with forecasting economic dynamics. The primary focus of the project, "Forecasting on House Price," is to optimize house price predictions by employing suitable algorithms and determining the most effective approach with a minimal error rate. The significance of this endeavor lies in its relevance to a widespread and impactful concern – the buying and selling of homes. As analysts in the realm of house prices, this investigation offers valuable insights into the housing market, empowering individuals to make well-informed decisions.

The analysis presented in this paper draws extensively from datasets sourced from Kaggle, a renowned platform recognized for its accurate and comprehensive datasets. Our objective is to achieve the most accurate house price predictions, leveraging techniques such as Linear Regression (LR), Ridge Regression (RR), Lasso Regression (LR), Gradient Boosting Regression (GBR), Random Forest Regression (RFR), Support Vector Regression (SVR) , and K Nearest Neighbors (KNN). The performance of each algorithm was evaluated based on scores, Mean Square Error (MSE) and R-Squared. The implementation of these techniques is facilitated through Python programming, and the corresponding code is accessible on GitHub at https://github.com/pariyaab/House_Prices_Regression_Models.

Figure 1 visually represents the data flow and processing intricacies involved in employing diverse regression techniques, encapsulating the essence of our methodology in predicting house prices with precision.

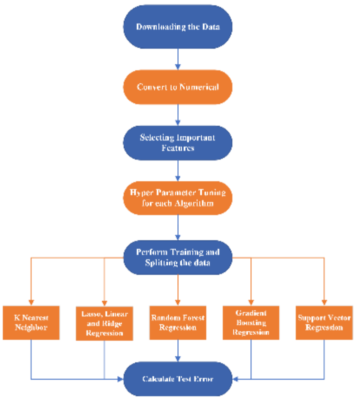
Downloading the Data
Convert to Numerical
Selecting Important Features
Hyper Parameter Tuning for each Algorithm
Perform Training and Splitting the data
K Nearest Neighbor
Lasso, Linear and Ridge Regression
Random Forest Regression
Gradient Boosting Regression
Support Vector Regression
Calculate Test Error

Fig1 - Data Flow Diagram

II. RELATED WORKS

In this study [1] , the authors conducted a comparative analysis of various regression techniques for house price prediction. The objective of the research was to forecast house prices based on financial provisions and aspirations of non-house holders. The study utilized different regression techniques including Multiple Linear Regression, Ridge Regression, LASSO Regression, Elastic Net Regression, Ada Boosting Regression, and Gradient Boosting Regression. The authors collected house sales data from the King County dataset. The performance of each algorithm was evaluated based on scores, Mean Square Error (MSE), and Root Mean Square Error (RMSE). The results showed that the Gradient Boosting Regression algorithm had the highest accuracy for house price predictions. The study aimed to assist sellers in estimating the selling cost of a house accurately and help people predict the right time to buy a house. The research can be cited as a valuable reference for predicting house prices using regression techniques.

III. METHODOLOGY

Before In this section, we aim to discuss the methodologies that have been implemented, shedding light on their underlying mechanisms and providing a comprehensive understanding of their application.

A. *K Nearest Neighbours (KNN)*

The K Nearest Neighbors (KNN) algorithm is versatile, serving purposes in both classification and regression tasks. Its fundamental mechanism involves modeling each data point in the space. When new data is introduced, the algorithm explores the distances between the new data point and all other existing data points. Subsequently, it selects the k nearest neighbors based on these calculated distances. For instance, if k is set to 10, the algorithm identifies the 10 closest neighbors, considering their distances.

In the context of a classification task, the algorithm determines the most frequent class among these neighbors. In contrast, for a regression task, it calculates the average value of each data point within the selected neighbors. This approach provides a robust foundation for decision-making in various analytical scenarios.

In this study, our initial step involves preprocessing the data, followed by its division into training, testing, and validation sets (specific details will be elaborated in the experimental results section). The optimization process for determining the optimal k-value is conducted, and it is found to be 44 (the methodology for this determination will be thoroughly discussed in the experimental results section). We designate "SalePrice" as the target variable for prediction, utilizing the average value derived from each set of 44 nearest neighbors. The outcomes of these computations are visually presented in Figure 1. The evaluation metrics are mean squared error and R-squared, with their formulas and calculation methods as follows:

Mean Squared Error (MSE):

Mean Squared Error is a measure of the average squared difference between predicted values and actual values. It is calculated using the formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y'_i)^2 \qquad (1)$$

where n is the number of data points, y_i is the actual value, and y'_i is the predicted value.

R-Squared (R2):

R-squared, also known as the coefficient of determination, represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It is calculated by the formula:

$$R^2 = 1 - \frac{SS_{res}}{SS_{totall}} \qquad (2)$$

Where SS_{res} is the sum of squared residual and SS_{tot} is the total sum of squared.

For our project, we first split our data into train and test by only 80 and 20 ration and then train our model by best k and test on test data. Finally, our scores for these two metrics are as follows:

Mean Squared Error ≅ 0.003078 and R-squared ≅ 0.39378.

Which are the acceptable results for our model and show its ability to predict the house prices.

B. *Linear Regression, Lasso Regression, Ridge Regression*

WRITE OTHER TEAM DESCRIPTION

IV. Experimental results

In this section, our aim is to elucidate the various steps, including the introduction of the dataset, the preprocessing methodology, and the hyperparameter tuning approach. These steps are essential for facilitating a comprehensive comparison of results..

A. *Dataset*

We obtained our dataset from the Kaggle website, and it is publicly available at the following link: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data. The dataset encompasses various features related to houses, with "SalePrice" as our target variable. A selection of these features, along with their descriptions, is provided in Table 1.

MSSubClass	The Building Class	
LotFrontage	Linear feet of street connected to property	
LotArea	Lot size in square feet	
Street	Type of road access	
LotShape	General shape of property	
LandContour	Flatness of the property	
MSZoning	The general zoning classification	
Utilities	Type of utilities available	

Table1- some of the features in our dataset.

Street	Type of road access to property
Grvl	Gravel
Pave	Paved

Table2 – A description of "Street" feature.

B. *Preprocessing Data*

As evident in Table 2, some of our features are categorical, and since regression algorithms require numerical data, we must convert categorical features into numerical ones. One of the most effective methods at this stage is one-hot encoding. This technique involves transforming categorical features into numeric ones by adding columns equal to the number of categories to the dataset. For instance, considering the "street" feature with only two categories, we introduce two columns, each corresponding to a type of street. We assign a value of 1 in the respective column if the house's street category is 1, and repeat this process for all data points. Following this step, our original dataset, initially with dimensions (2919, 81), expands to (2919, 2262).

Subsequently, we proceed to the feature selection step to identify the most informative features among the multitude. This process will be expounded upon in the next paragraph.

In the feature selection phase, we adopt the Mutual Information Gain method based on the principles outlined in [2]. Mutual Information (MI), synonymous with Information Gain (IG), quantifies the dependence or shared information between two random variables [3]. It is rooted in Shannon's definition of entropy, expressed as:

$$H(X) := -\sum_{x} p(x)\log\bigl(p(x)\bigr) \qquad (3)$$

This definition captures the uncertainty inherent in the random variable X. In the context of feature selection, our goal is to maximize the mutual information, signifying relevance, between a given feature and the target variable.

The Mutual Information (MI) is defined as the relative entropy between the joint distribution and the product distribution:

$$MI(X;T) := \sum_{X}\sum_{T} p(x^j,t)\log\frac{p(x^j,t)}{p(x^j)p(t)} \qquad (4)$$

Here $p(x^j,t)$ represents the joint probability density function of feature x^j and target t, while $p(x^j)\ and\ p(t)$ denote the marginal density functions. The MI is zero or greater than zero for independent or dependent X and Y, respectively. To maximize MI, a greedy step-wise selection algorithm is employed.

The algorithm initializes a subset of features, denoted by matrix S, with one feature. Features are added one by one to this subset, and the index of the selected feature is determined using:

$$j := \arg\max\ MI(Y_{S\,\cup x^j}\ ;t) \qquad (5)$$

This equation is also used for selecting the initial feature. The selected features are assumed to be independent. The addition of new features stops when the highest increase in MI at the previous step is achieved. It is crucial to note that even if a variable appears redundant or uninformative in isolation, it may still contribute valuable information when combined with another variable. This approach effectively reduces the dimensionality of features without significantly compromising performance [4].

For our study, we set a threshold of 0.01 for our features. Subsequently, out of 2262 features, we selected 56 features deemed most informative. Our algorithm is then executed, and the dataset is split, considering only these selected features – effectively constituting our new dataset. In Table 3, you can find some of the essential features. Notably, "Built Year" emerges as a crucial feature significantly influencing house prices.

Feature Name	Score
MSSubClass	0.158
YearBuilt	0.087343
Neighborhood	0.1158657413
MasVnrArea	0.084393

Table 3 – some selected features along with their score.

C. Experimental Setup

Following the selection of important and informative features, the subsequent step involves hyperparameter tuning for each algorithm through dataset splitting and cross-validation. Take the K-Nearest Neighbors (KNN) algorithm as an example: initially, we shuffle our data to mitigate bias or indexing issues. Subsequently, we partition our dataset into three sets - train, test, and validation, with an 80-10-10 ratio, respectively. This process is repeated across ten distinct partitions.

For each value of k ranging from 1 to 51, we execute a ten-fold cross-validation. In the first iteration, segments 0-7 are assigned as the training set, 8 as the validation set, and 9 as the test set. In the subsequent iteration, segments 1-8 serve as the training set, 9 as the validation set, and 0 as the test set and the rest of the iterations are the same.

Throughout each iteration, the model is trained on the training set and tested on the validation set to identify the optimal k. This process is repeated ten times, and the errors for each k are averaged. The results are then sorted, and the minimum errors are selected as the best k. Figure 2 visually represents the errors corresponding to different k values, utilizing Mean Squared Error (MSE). Following the determination of the best k, ten additional iterations are conducted. The model is constructed solely with the optimal k, trained on the training set, and tested on the test set. Subsequently, the mean score over ten tests is computed, and the best k is reported, considering the mean error ± standard deviation, as illustrated in Table 4.

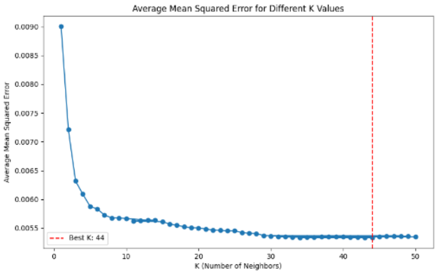

Figure2 – show MSE over different K

Algorithm	Best Parameter	Errors
KNN	K = 44	0.0053±0.002
RFR		
SVR		
LR(Linear)		
GBR		
RR		
LR(Lasso)		
RFR		

Table4 – the error of hyper parameter tuning

D. Experimental Results

In this section, we want to have a overview of the different algorithms and compare them together. You can find complete result and errors for each algorithm in Table 5.

Algorithm	MSE	R-Squared
KNN	0.003078	0.39378
RFR		

SVR		
LR(Linear)		
GBR		
RR		
LR(Lasso)		
RFR		

Table5- comparison of different algorithms

ACKNOWLEDGMENT *(Heading 5)*

We would like to express our heartfelt gratitude to our instructor, Benaymin Ghojoogh, at the University of Guelph, Canada. His guidance, encouragement, and insightful feedback played a pivotal role in shaping the course of our research.

Our sincere thanks also go to the University of Guelph for providing us with an enriching academic environment and valuable resources that facilitated our study. We extend appreciation to our fellow students and colleagues who contributed to discussions and shared insights that enhanced the quality of our work. Additionally, we want to acknowledge the support from the University of Guelph, which includes any relevant departments, for their role in fostering an environment conducive to research and learning.

REFERENCES

[1] Madhuri, C. R., Anuradha, G., & Pujitha, M. V. (2019, March). House price prediction using regression techniques: A comparative study. In *2019 International conference on smart structures and systems (ICSSS)* (pp. 1-5). IEEE.

[2] Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., & Crowley, M. (2019). Feature selection and feature extraction in pattern analysis: A literature review. *arXiv preprint arXiv:1905.02845*.

[3] T. J. Cover TM, Elements of Information Theory. 2nd edn, Wiley-Interscience, New Jersey, 2006, vol. 2.

[4] N. Nicolosiz, “Feature selection methods for text classification,” Department of Computer Science, Rochester Institute of Technology, Tech. Rep., 2008.

[5] …

[6] …