# House Price Prediction

**Navpreet Singh Sidhu**
Computer Engineering, SFU
nss11@sfu.ca

**Khushnaz Rataul**
Computing Science, SFU
krataul@sfu.ca

**Guarjit Singh Grewal**
Computing Science, SFU
gsg24@sfu.ca

**Raajvansh Singh Dhaliwal**
Computing Science, SFU
rvdhaliwal@sfu.ca

## Abstract

House price Prediction is a prime topic of discussion in the field of real estate. Real estate industry is undecipherable as of today's date, this paper brings forth the machine learning techniques that are applied to analyze and discover useful models for house price fluctuations. Predicting house prices based on real-life problems such as heights of the ceilings, neighborhood, heating furnaces, pool area, locality, year built and so on. Various techniques of machine learning such as regression are used to consider all the basic amenities and parameters required to predict the price of a house. The goal of the paper is to build a model which maximizes accuracy and minimizes the error of our prediction.

## 1 Introduction:

The approach of machine learning is very widely used in different types of applications to predict outcomes in situations where it is nearly impractical to generate results using standard algorithms. Machine Learning makes this process of predicting an outcome easily achievable using a large dataset. It consists of around 80 features which includes parameters such as year sold, type of flooring, garage, utilities, street, house-style and so on. This application can be viewed as one step forward towards accurate decision making for the interested buyers. The goal of the project is to build a model which predicts the house price accurately keeping in mind the geological factors of an area or locality. Price prediction applications are also needed for financial purposes such as mortgage approvals, real estate investments, development, and construction purposes. The main approach of the project is to use two datasets, namely, training dataset and testing data set. Training data is used to teach our model and testing dataset is used to test the model.

There is a huge amount of data and resources available on today's date which can give us appropriate figures to help improve the decision- making skills of the house-buyers. The main language of coding used in the project is Python along with its various libraries such as panda, scikit-learn, NumPy, etc. The main topics and concepts used are regression, random forests, decision trees, overfitting and underfitting, training and testing. Data that is being used needs to be refined or cleaned first by removing duplicates and checking for missing data values and then identifying numerical and categorical features of the data. The training and testing datasets were imported from the Kaggle competition "House Prices-Advanced regression techniques".[1] To better understand the data that we are provided with, we conduct exploratory data analysis to highlight the main attributes of the data graphically or visually to explore its quality.

## 2 Exploratory data analysis

Representing data graphically in the form of histograms or charts gives us a better understanding of the parameters. It highlights outliers, the trends they follow, similar data patterns which helps us to summarize and analyze data which will facilitate the decision-making process. Initially the model building starts with cleaning the data that is given by removing duplicates and checking for missing data values. The most important features of the dataset on which the real estate value depends on such as (area, amenities, neighborhood, locality, etc.) are recognized from the set of 80 features which are given to us. The data is then divided into two datasets for training and testing. As the name suggests, training dataset is a subset used to train data and testing subset is the subset which is used to test data.[3] The training dataset consists of 70 of dataset to train the model and 30 of the dataset is used to test data. The test set should meet the following criteria to generalize results for any new data which is entered into the application: firstly, the dataset should be large enough to conclude correct meaningful results, secondly, the training and testing datasets should use the same characteristics in order to make predictions correctly.[3]

While preparing the dataset, we find that out of 80 given features, 18 features contain null values which means that for example, out of the total data given for the fence 401 houses does not have that feature. Out of these 18 features, we dropped the features with highest null values and build the training set accordingly.

| Feature | No. of NULL values | Feature | No. of NULL values |
|---|---|---|---|
| PoolQC | 499 | GarageQual | 29 |
| MiscFeatures | 481 | GarageCond | 29 |
| Alley | 469 | BsmtExpoure | 11 |
| Fence | 401 | BsmtFinType2 | 12 |
| FireplaceQu | 233 | BsmtCond | 11 |
| LotFrontage | 87 | BsmtQual | 11 |
| GarageYrBuils | 29 | BsmtFinType1 | 11 |
| GarageType | 29 | MasVnrArea | 1 |
| Garagefinish | 29 | MasVnrType | 1 |

Table 1: House Features that contain NULL values in dataset.

The SalePrice is taken as the target variable and correlation of other variables is calculated with respect to this target variable. In the model we see that there are 18 features that have a correlation higher than 0.3, therefore these features are correlated with the target variable and with each other as shown in the figure.

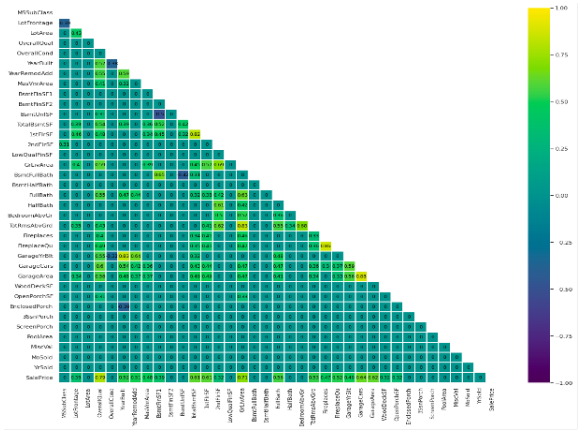

Figure 1: Heatmap of Corelation[7]

# 3 Building the model

## 3.1 Linear Regression

Regression is a machine learning technique based on supervised machine learning. It is a very simple method which is used to depict the relationship between one dependent variable and one or more independent variables. It minimizes residual sum of squares by fitting the coefficients to the data in a linear model between the target variable in the dataset. In linear regression, the estimated dependent variable is equal to the constant and regression coefficient times the independent variable, added together.

## 3.2 Random Forest

Random forest is an ensemble of many Decision Tree Classifiers. It is a machine learning algorithm which is made up of several decision trees which uses randomness to create each table. It is implemented using RandomForestClassifier from Scikit-learn machine learning library in python. It uses multiple independent decision trees in parallel to learn from data and aggregates their predictions for an outcome. Random forest training algorithm applies the technique of bootstrap aggregating, or bagging, to decision trees. Random forests construct many decision trees while training of the data, and results from all the decision trees is combined to predict the outcome. Random forests also help in acquiring low variance to reduce the complexity of the model for decision trees. The algorithm for random forest works as follows: Consider the training set to be X = x1,. . . .,xn , with responses Y=y1,. . . ,yn repeating bagging (B) times. Select a random sample with replacement of training set and fits trees to these samples, For b=1,. . . .,B: Sample, with replacement, n training examples from X,Y: call these Xb, Yb and train a classification or regression tree fb on Xb, Yb. Then after training, predictions for unseen samples x0 can be made by averaging the predictions from all the:[4]

$$f' = \frac{1}{b \sum_{b=1}^{B} fb(x_0)}$$

## 3.3 k-nearest neighbors

It is a supervised machine learning algorithm where k means nearest and as the name suggests it uses nearest data point to train its model. Random forest and k-nearest neighbors are built from a training set in order to make predictions (y') for new data-points by taking a look at the neighbor points which is mathematically written in terms of weight of the function.[4]

$$y' = \sum_{i=1}^{N} fb(W(x_i, x')y_i$$

## 3.4 XG Boost

XG boost stands for eXtreme Gradient Boosting. Regardless of the type of prediction task regression or classification, XG Boost is the most popular machine learning algorithm in today's date.[5] Boosting is an ensemble technique where new models are added to correct the errors made by existing models and the technique is added sequentially until no further improvements can be made. It is implemented into the python work environment by using XGBRegressor imported from XG boost. This boosting algorithm assumes a real-life value y and seeks an approximation F(x) in the form of weighted sum of hi(x) from class H called weak learners:[2]

$$F(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + const$$

# 4 Results

• minMax varies from 0 to 1, the closer it is to 1, the more is the accuracy of the prediction. In this case XG-Boost regression is the closest to 1.

Standard scale accuracy: Using Scaling techniques we change the standard deviation such that it becomes close to 1. Considering the formula $z = \frac{X-\sigma}{\mu}$, Machine learning algorithms exhibit good predictions when performed on similar scale

| Model | Accuracy | MinMax | StandardScalar | Normalizer |
|---|---|---|---|---|
| k-nearest Neighbour | 0.614155 | 0.714214 | 0.752954 | 0.682085 |
| Random Forest | 0.835285 | 0.834383 | 0.832793 | 0.821512 |
| XG Boost | 0.861797 | 0.860284 | 0.873529 | 0.859724 |

Table 2: This table demonstrates accuracy of different regression models.

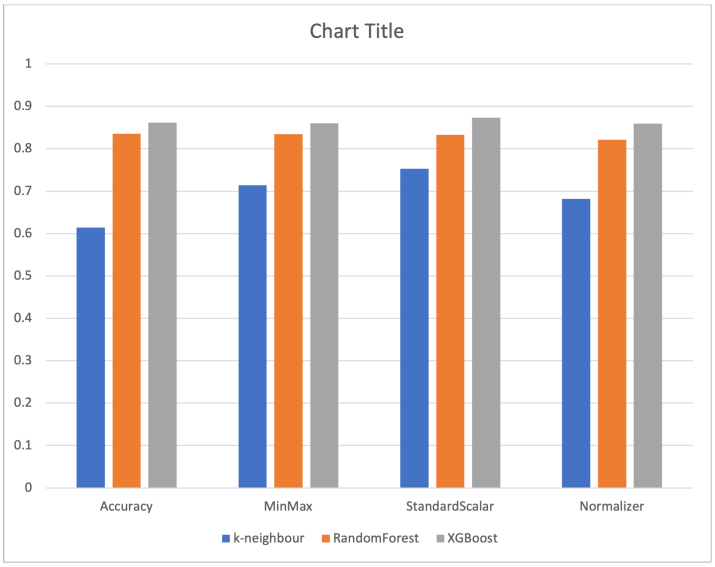


Figure 2: Accuray chart for different Models

• In the above graph the blue bar represents the accuracy, minmax, standard scaler accuracy and Normalizer accuracy of k-nearest neighbor algorithm.

• In the above graph the orange bar represents the accuracy, minmax, standard scaler accuracy and Normalizer accuracy of random forest regression model algorithm.

• In the above graph the grey bar represents the accuracy, minmax, standard scaler accuracy and Normalizer accuracy of Gradient booster model algorithm.

The results also show that random forest is more powerful and accurate than a decision try but they are a little complex because each classification decision has multiple decision paths. On the other hand, XG Boost algorithm is more accurate than random forest because in this method trees are trained to correct each other's errors.

## 5 Conclusion

In this project, the goal was to achieve an accurate house price prediction application which will be of a great use in the process of decision making while buying a house. The frequent fluctuations in the prices of real estate properties can benefit a lot from such an application as it can assist them on determining the apt time to buy a house. While making predictions about the price, the model should consider some important features such as amenities, neighborhood, market-accessible and so on, in order to give a close approximation of the price. The application is designed such that it will help buyers by giving them accurate results and saving them from investing in houses whose worth is predicted to decline or the houses which are not in their budget. Firstly, we cleaned

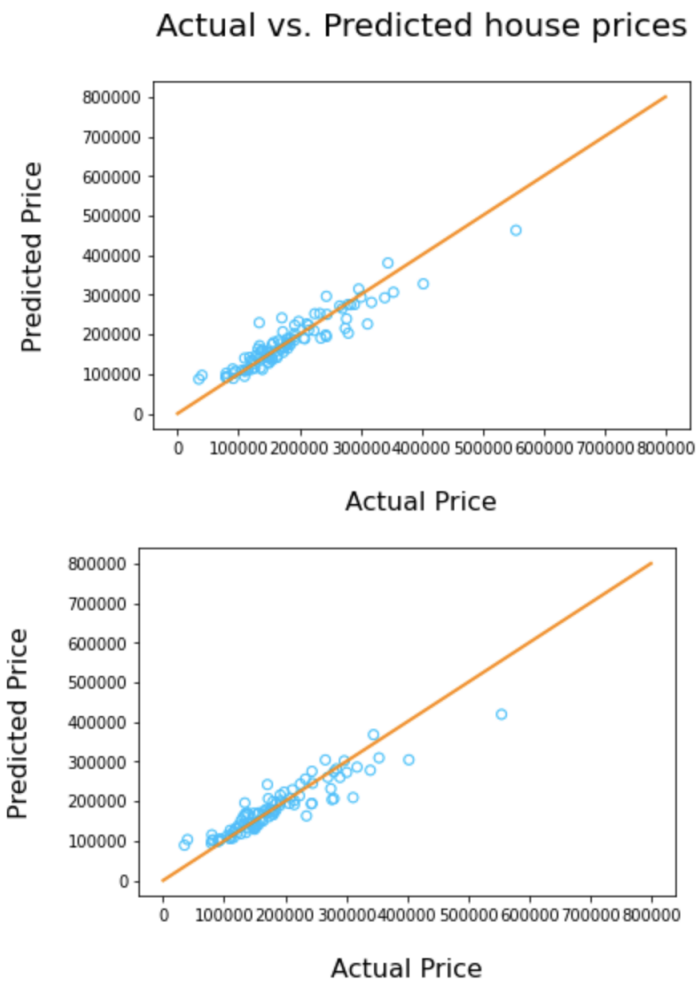


Figure 3: Real vs Predictive price of XGBoost and Random Forest respectively.

the data, removed the duplicates, handled the missing values, explored data analysis by visualising data, transformed data, and built the model. We analyzed the main attribute 'SalePrice' with other correlated variables from our training dataset and tested some statistical assumptions. To build the model we used techniques of linear regression, random forest and XG boost regression model. We evaluated the results produced by the methods used in building the model. It is found that XG boost is the most accurate model for house price prediction with an accuracy of approximately 90 percent.

## 6 Contributions

### 6.1 Khushnaz Rataul

• Research for the project

• Random forest model calculations and coding

• Contributed to writing the group report along with other members

• Contributed to analysing the final result

• Comparing the different models to find the most appropriate one.

### 6.2 Navpreet Singh Sidhu

• Coding work

• k-nearest neighbor model calculations

• gathered knowledge of various python libraries used in this project

• Contributed to analysing the final result and report formatting

• Contributed to writing the group report along with other members

### 6.3 Gurarjit Singh Grewal

• Data organizing

• Linear Regression model calculations

• Contributed to analysing the final result

• Contributed to writing the group report along with other members

### 6.4 Raajvansh Singh Dhaliwal

• XG Boost model calculations and code work

• Helped to format the report according to NeuroIPS guidelines

• Contributed to analysing the final result

• Contributed to writing the group report along with other members

## 7 References:


[1] Pmarcelino. (2019, August 23).*Comprehensive data exploration with python*. Kaggle. Retrieved April 21, 2022, from https://www.kaggle.com/code/pmarcelino/comprehensive-data-exploration-with-python

[2] *House price prediction using Machine Learning and Neural Networks*. IEEE Xplore (n.d). Retrieved April 21, 2022, from https://ieeexplore.ieee.org/document/8473231/referencesreferences

[3] *Google. (n.d.). Training and test sets: Splitting data, machine learning crash course* google developers. Google. Retrieved April 21, 2022, from https://developers.google.com/machine-learning/crash-course/training-and-test- sets/splitting-data

[4] Wikimedia Foundation. (2022, March 1).*Random Forest*. Wikipedia. Retrieved April 21, 2022, from https://en.wikipedia.org/wiki/Random$_forest$

[5] *XGboost Python tutorial: Sklearn regression classifier with code examples.* DataCamp Community. (n.d.). Retrieved April 21, 2022, from https://www.datacamp.com/community/tutorials/xgboost-in-python

[6] Brownlee, J. (2020, August 25). *How to use data scaling improve deep learning model stability and performance*. Machine Learning Mastery. Retrieved April 21, 2022, from https://machinelearningmastery.com/how-to-improve-neural-network-stability-and- modeling-performance-with-data-scaling/

[7] Kaggle Competition, *House Price Prediction(step-by-step,https://www.kaggle.com/code/adibouayjan/house-price-step-by-step-modeling*