Table of Contents

[Abstract 1](#_Toc175356867)

[Acknowledgements 3](#_Toc175356868)

[Chapter 1: Introduction 4](#_Toc175356869)

[I. Overview 4](#_Toc175356870)

[II. Research Questions 4](#_Toc175356871)

[Chapter 2: Background 5](#_Toc175356872)

[I. Literature Review 5](#_Toc175356873)

[II. Algorithms 8](#_Toc175356874)

[a. Linear Regression 8](#_Toc175356875)

[b. Support Vector Machine 8](#_Toc175356876)

[c. Decision Tree Regressor 9](#_Toc175356877)

[d. Random Forest Regressor 9](#_Toc175356878)

[e. Gradient Boosting Regressor 10](#_Toc175356879)

[f. AdaBoost Regressor 10](#_Toc175356880)

[g. XGBoost Regressor 10](#_Toc175356881)

[III. Metrics: 11](#_Toc175356882)

[Chapter 3: Methodology 12](#_Toc175356883)

[I. Tools and Techniques 12](#_Toc175356884)

[II. EDA and Visualization 12](#_Toc175356885)

[Chapter 4: Results and Conclusion 17](#_Toc175356886)

[I. Process Flow 17](#_Toc175356887)

[II. Results 17](#_Toc175356888)

[III. Conclusion 21](#_Toc175356889)

[IV. Future Work 21](#_Toc175356890)

[Chapter 5: Legal, Ethical and Professional Issues 23](#_Toc175356891)

[References 24](#_Toc175356892)

[Appendices 26](#_Toc175356893)

[I. Images 26](#_Toc175356894)

[II. Code 29](#_Toc175356895)

# Abstract

The present study aims at comparing the predictive performances of the single and collective models of the machine learning framework on the record from King County, USA. Several models such as Linear Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, Gradient Boosting, AdaBoost, and XGBoost are used and their performances are compared on three different splits of data (80:20, 70:30, and 60:40). The results of the models are assessed with essential measures such as, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and R-squared (R²), the accuracy and reliability of the models. Comparing the errors and the R-squared values of all models, we can note that Linear Regression and XGBoost models have, as a rule, the lowest errors and the highest R-squared values that proves their effectiveness in identifying patterns in the data on housing. AdaBoost also fared badly, specifically in RMSE and R-squared meaning that it is not suitable for this specific dataset. The findings draw attention to the fact that feature engineering, data preprocessing and selection of right model all matter in achieving high accuracy. This research provides suggestions for the next studies and the real estate practice, stressing that more reliable models are required to predict the housing prices.

# Acknowledgements

# Chapter 1: Introduction

## Overview

Ensemble models are a special type of machine learning models that use a multiple individual machine learning models to increase the performance (Singh, 2018). This results in capturing more patterns in the data and generalizes better than individual algorithms. There are multiple variations of these ensemble models, a few most used types are bagging and boosting. Bagging is used in Random Forest algorithm where multiple individual models (Decision Tree) are trained simultaneously and an aggregate of their predictions is used as the final output (Kumari, 2024). Where are boosting trains multiple individual algorithms sequentially where each new individual algorithm reduces the error of the previous algorithm (Ray, 2015). These ensemble algorithms usually take longer to train than the individual algorithms.

Individual models are easier to understand and easier to implement than ensemble models. The understandability is one major area where the ensemble models lack. The complexity of these models works as a double-edged sword, where on one side it captures complex relationships of the data and on the other side it becomes harder and harder to interpret the result.

In this research these algorithms will be put to test on a ‘House Sales‘ data from King County, USA([link](https://www.kaggle.com/datasets/harlfoxem/housesalesprediction)). Every algorithm needs data to be trained on, this dataset is complex enough with around 20 columns which each represent a feature that helps in predicting the house price. The data also has enough observations ~20k for the model to train on. More information on the dataset can be found in the further sections.

## Research Questions

How do ensemble learning methods compare to individual machine learning models in predicting house prices, and what are the effective performance benefits through ensemble algorithms?

# Chapter 2: Background

## Literature Review

The strategies that are used in this paper (Ahtesham, Bawany and Fatima, 2020) are concerned with the machine learning technique known as XGBoost that aims to predict the house prices in Karachi considering the dataset from the Open Real Estate Portal of Pakistan. The original dataset has 168,447 instances and 20 attributes but only data of Karachi was used, so after preprocessing records are 38,961 and attributes are 14. Dubious records were stripped off based on missing values and the unfruitful features; the resultant records were 38,961 and features were 14. The implementation section elaborates on the employment of the several Python packages including but not limited to pandas, numpy, scikit-learn, Matplotlib, Seaborn, and finally the XGBoost. Some of the entries or features were not filled up and hence such data records were omitted in order to provide a proper set of data for learning. Categorical data was split from numerical data and associated changes performed while features with negligible correlation to the target variable, sale price, were deleted using Pearson’s correlation function.

As a result, feature selection was carried out as a means of providing generalizable feature-data to the model. The process involved the usage of “crcols. remove” in removing features perceived as not elemental to the prediction as a way of narrowing down to the most important features. This set was then divided into training and testing data with the different proportions (60:40, 50:50 and 70:30) to test the models. The reasons for selecting XGBoost include versatility, fast learning time, and accuracy that are most suitable for a tabular format and both binomial and multinomial classification as well as regression problems. For training the model XGBRegressor is used and for the validation the testing dataset is used. It was established that learning rate would be constant at 0 always. 01 for all experiments. It is seen that whether the train/test split is done in 60:40, 50:50, 70:30 split, the model is highly accurate scoring to 98% and the Mean Absolute Error (MAE) of 22502. 0824694.

These results show that the method used in this study, that is the XGBoost model, is valid, reliable and consistent. Thus, the model outcome remains well-maintained of performance indicators, although the proportion of data used for training and testing is not constant. This affirms the effectiveness of the preprocessing steps, an ability in feature selection, as well as the XGBoost model in the processing of the given dataset for house price prediction in Karachi. The train/test ratios used in the experiments show that the model is accurate enough and has relatively low MAE while using unseen data to achieve the modified goal and become quite a precise tool for predicting house prices. Conclusively, based on the preprocessing, features selection and evaluation techniques the methodology of XGBoost model is used and validated to obtain accurate prediction of house prices.

The methodology in this research paper (Chaurasia and Haq, 2023) focuses on generating a machine learning model which relates to using linear regression in order to estimate the housing prices. The study uses data gathered from the ‘housing\_data. csv’ data set freely available at Kaggle; it possesses features as the average income, the age of houses, the number of rooms and bedrooms, population, and price of the house. First of all, non-informative columns such as ‘Address’ are often dropped in data preprocessing step and the dataset is then divided into predictor variables (or independent variables, denoted by X) and the dependent variable (denoted by y) using the Pandas module.

The next utilized algorithm could be the linear regression one originating from the SciKit-Learn library. This algorithm is selected due to its ability to find regressed relationships and carry out prediction activities based on the patterns identified. The model is trained on the training data set and then tested on the given test data set and the best achievable performance of the model is determined using metrics such as MAE, RMSE, VIF and R squared. These metrics serve as benchmarks to gauge the model's accuracy in predicting house prices: the desirable outcomes hence are small values of MAE and RMSE for business, high R squared value, showing closeness of fit of the model to the data.

The findings of the study based on the experimental results indicate that linear regression model performs with MAE of 82,288. The last two evaluations stand for 22, which is the mean absolute error of the program and indicates the average absolute difference between the actual and predicted house prices. According to the model, the approximate error charge is ±6 on average. As for its usage in practical applications in setting real estate prices it registered a 67% which is not too small. For the purpose of diagnosing the given dataset and additionally for the confirmation of the efficiency of the designed models, this research also applies the kinds of graphs including histograms, scatter plots, and heatmaps. They complement the analysis of the result of modelling by offering information about the distribution of the data, and the relationship between them.

In summary, the studies show that machine learning, with linear regression as an example, is effective at estimating the price of housing according to numerous characteristics of the socio-demographic nature. In particular, it underlines the importance of the data quality for the improvement of model efficiency and outlines the possible directions of further researches focused on increasing predictive capacity in real estate field. Thus, employing state-of-the-art analytical tools and approaches, the research provides a set of insights into the ML approaches usage in order to solve the problems in the context of the housing price forecasting, being an important and rather complex domain.

Thus, the scope of this research paper focuses on establishing a strong foundation to forecast housing prices using the regression models that are helpful for developers and prospective buyers. This paper relies on the use of Python modules to develop highly complex machine learning algorithms with the ultimate goal of bringing efficiency in the housing price prediction. Moreover, it is also concerned with different graphical and numerical techniques necessary for prediction.

This study by (Kiran *et al.*, 2023), makes a start by stressing on the need to predict the prices of housing since the real estate industry is dynamic. It underlines three factors, namely; the physical state of the house as well as the design and location that have a profound bearing with house prices. They are indispensable not only for prognosis of the cash flows necessary for the future financial management but also for the analysis of the tendencies at the market and possible permanently available offers.

One of the most important components of the research analysis is to implement the linear regression model and evaluate a dataset collected purposefully for research purposes. The following critical information is obtained from raw data in this regard, aided by efficient real estate data mining: The predictions about house values pertinent to specific essential property features and pertinent demographic data. When it comes to prediction, the literature turn in the study shows that the most preferable models are ANN, SVM, and regression models, especially because of superior performance in complex market like housing markets.

Regarding the future work, the methodology section describes linear regression to be applied as the key predictive model. This entails feeding the model to the split data, where half of the data is used in training the model and the other part in testing the model in order to get more reliable results of the model’s performance. Being as the aim of the model is to predict the price of a house, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the R squared value are used as measures of the accuracy of the model. The use of aids such as Correlation Heatmap and graphical outputs make it easier to understand the data relations and outcomes of the models.

Thus, the proposed system combines users’ and administrative features that allow input, model training, and validation. For instance, the graphical user interface provides the registration process as well as data entry process; the control panel handles the activation of the users as well as dataset management. This setup enables the effective implementation of the machine learning algorithms to generate accurate housing prices meeting the users’ variety and convenience.

In the final section of the study, the author reiterates the applicability of these conclusions to the field and to its many interested stakeholders, including housing developers, scholars, and others in the realm of real estate. Thus, the study advances knowledge regarding the identification of critical factors affecting house prices and analysis of the effectiveness of different machine learning models to achieve high levels of accuracy. Thus, it shows how enhancing the predictions in the housing market can be accomplished using Python-based data mining techniques, and the possible contributions to the future development of this type of study.

In other words, this paper outlines a worked-out and detailed approach to modelling the housing prices using labour-intensive machine learning techniques and stresses the theoretical background of the methods used as well as the practical consequences for the key players in the field. Therefore, with the help of complex analytical instruments and methods the research is to resolve the issues connected with housing prices prediction and contribute to more rational decision-making regarding housing market.

In this research paper by (The Danh Phan, 2018), the main purpose of the methodology section is to identify how through the use of a number of machine learning algorithms procedures, a model that could effectively predict the price of houses could be created. The data used for the study is obtained from Kaggle as Melbourne Housing Market Data containing 34557 observations and 21 variables actual sold house transactions in Melbourne from year 2016 to 2018. It’s in these variables that SH data of transactional details, location predictors, and house features like the number of bedrooms, bathrooms, car slots, and land size exist.

The raw dataset is pre-processed since it contains missing data and outliers that must be addressed first. It is standard practice to remove columns with more than 55% of the data missing; likewise, any observation that has a missing value for the dependent variable (price) is also removed in order to avoid introducing any bias. For predictors that have insignificant missing data amounts, imputation is done by methods such as carrying out Google’s API for geographical data and median imputation for the land size depending on the house types and suburbs. Outliers that are observations that are considered to be extremely different from the entire set of the given dataset are also considered.

The cleansed data consist of 11 predictor and more than 21,000 records for creating and assessing the prediction models. Analysing data with the help of descriptive analysis the author concludes that the examined houses had three bedrooms, one bathroom, a land size more than 5000 square meters, and costs approximately 900,000 dollars. About the dependent variable, the values are normalized using the logarithm of the price (log(price)) in order to get a good fit in the model.

In order to improve interpretability and, at the same time, achieve high levels of prediction error, several procedures for data reduction are used, which are Stepwise selection and Principal Component Analysis (PCA). Backward elimination identifies the most relevant predictors to the dependant variable, and these they are: the number of rooms in a house, the distance from a house to CBD, latitude and longitude of a geographical location and type of house. Curiously, the factors such as the size of the piece of land and the number of car spots play a very minimal role in the prices of the houses.

Linear Regression, Polynomial Regression, Regression Tree, Artificial Neural Network, and Support Vector Machine are adjusted in the study and the models are enhanced with or without PCA integration. Out of the several ways of measuring the performance of the models, Mean Squared Error (MSE) has been used to judge the model on the training and evaluation dataset and Linear regression model on which the models are built is used for comparison.

Regression analysis results show that the programs Regression Tree and Polynomial Regression has equally offered close prediction with least error. Compared to other algorithms, Neural Network, which seems to be more powerful and generalized, possesses comparatively a less effective result for this dataset. PCA and tuned SVM have better accuracy rate but usually have overfitting problem because the gap between the evaluation MSE and training MSE is larger. The resulting tuned Version of the Stepwise selected set of Variables, feeding it into SVM, meantime proves to be the best setup of all these models yielding the least overall error on this dataset.

Regarding execution time, pure linear and polynomial regression models are non-iterative and thus provide a response in a blink of an eye while non-nearby models like Neural Network and SVM models take fairly a good amount of time. Hence, it is seen that the proposed method of invoking Stepwise in conjunction with SVM is more efficient than the combined method of invoking PCA in conjunction with SVM, as it gives optimum accuracy in reasonable time.

The research also pinpoints the need to explain the results and, thus, opts for interpretation; simple models such as Linear Regression and Decision Trees are clear while models such as Neural Networks and SVM are not. In order to look into such features, it is recommended to apply the Stepwise-SVM and Polynomial Regression on the historical datasets sourced from various cities in Australia in an attempt to increase model’s effectiveness and precision.

In conclusion, this research presents successful algorithms for the house price forecasting, with the emphasizes on the Stepwise-SVM as one of the superior ways. The findings of the research are useful to understand the Melbourne housing market, and pave the way for extending the application of these approaches to other areas.

## Algorithms

### Linear Regression

A linear regression analysis the strength of the relationship between the dependent and one or more independent variables on the assumption that the relationship is straight. To estimate the method, it uses a regression line in which its suitability is based on the minimum sum of squared deviations of the actual observation and predicted values. By using coefficients, including the slope and intercept which are obtained for the best fit line, prediction on to new data is made. When the relationships are non-linear, first the data must be put in linear terms. Simple regression analysis includes data on one independent variable, but multiple regression uses more than one such variable. It is simple to use, requires little pre-processing of data and is useful for predicting and for drawing causal conclusions. However, it has a problem of declining performances for non-linear relation and issues like multicollinearity which is injurious to overfitting. Also, for valid inference the predictors are required to be normally distribution and should meet the condition of homoscedasticity. Failure to understand these requirements means that the forecasts made are wrong and the relationship that exists between two variables is not clear (*What is linear regression?*, 2010).

### Support Vector Machine

The standard SVM is a method of solving a supervised learning problem for classification and regression. The main objective of SVM is to determine the right hyperplane which correctly classes the features of the points on the N-dimensional space. This hyperplane separates with the largest margin the closest points belonging to different classes, the so-called support vectors and it is good at generalizing on new data. To solve non-linearly separable problems, SVM employs a method called Kernel method that maps data-set into a second higher-dimensional feature space where data is easier to be separated. Different kinds of kernel functions such as polynomial, radial basis function, and sigmoid etc may be used (Fagbuyiro, 2023).

Some merits associated with SVM include its ability to work within the high-dimensional space, and has a small space complexity because of its support vectors. But, deciding the kernel function and tuning parameters is more of an art and SVM’s scales poorly with large noisy datasets and tends to overfit them. Furthermore, models based on SVM are less transparent, and the training time may be very large when working with a large quantity of data.

### Decision Tree Regressor

Decision Tree Regressor is a supervised learning model dealing with regression type of problems the construction of which is done by splitting the data based upon conditions related to the features to minimize the loss. This forms a tree like structure to arrive at a conclusion of the classification and this includes the internal node which tests on the features, branch which tests on an outcome and the last node which is a leaves node that predicts on the classification. The choice of splits at different nodes is carried out based on some factors such as variance, the Gini index, or information gain so that at every step, the information gain is at its maximum, and the process continues to the next node is pure or a pre-specified depth is reached. The above forecast is made after a path from the root node leads to a leaf node under the following considerations. To reduce overfitting random forests and boosted trees are used which are trees that are combined in the process (The Click Reader, 2021).

Advantages of decision trees are easy to interpret, works good with non-linear data, requires little data pre-processing and suitable for both nominal as well as continuous data. But they tend to have a high variance with deep trees, and the accuracy also fluctuates according to the chosen features to some extent based on the changed dataset.

### Random Forest Regressor

Random Forest Regressor is a supervised learning algorithm that belongs to the ensemble methods for regression problems. It joins several decision trees that are made from the different bootstrapping of the training data set to make a model. In this process of selecting data points, the possibility of repeating some data point is permitted while others may not be chosen so this increases the diversity of the models. Random Forest takes it to another level and limits the number of features that individual trees can consider in the construction of the tree. Random Forest trees are unpruned and fully grown trees while single regression trees are pruned. Averaging the results obtained from all the trees provides an additional level of variance reduction and makes the final prediction stable (Koehrsen, 2017).

Plus side include robustness in working with Numerical and categorical datasets, it handle missing values and outliers, and it is resistant to overfitting. It also offers a measure of feature importance as well as learns to deal with non-linear relationships. However, it can be hard simply to visualize, leads to poor results when the data have a large number of features or can be sparse, and could be biased in its estimate of probabilities.

### Gradient Boosting Regressor

Gradient Boosted Regressor, one of the boosting models that is used in the family of machine learning algorithms with the aim of reducing the level of errors in the models as a result of repeated use of weak models particularly the decision tree models. That is the model is gradually constructed by starting with a simple initial model and then incorporating a number of weak predictors to reduce the error between the predicted and actual values. Every single learner aims then at reducing its error for poorly predicted observations and thus improves the model. Some of many tuning parameters are the learning rate that determines the contribution of each tree and the number of estimators affecting the model’s accuracy and variance (Dhiraj, 2019).

Advantages are high precision, flexibility with different loss functions, ability to do feature selection depending on the model, and insensitivity to outliers. However, it can over fit if not well adjusted it may take time to train because it uses iteration in training and last but not least it is less explainable than more simple models. Also, it is not easy to implement it across many systems and machines; thus, large-scale implementations are quite testing.

### AdaBoost Regressor

AdaBoost or Adaptive Boosting is an ensemble learning technique regarding learning the basis for classification or regression analysis and by combining many weak learners into one mighty learner. It mainly provides each training instance with an equal significance and then trains a weak learner usually a decision tree of only one seat deep. Much like evaluation of errors, AdaBoost modifies the weights of misclassified instances, when the next weak learner is trained. This process is repeated and the final output is a kind of voting across all of the weak learners where all of them are averaged out (*Adaboost for Regression - Example*, 2024).

Some of the benefits of AdaBoost include the following: Overfitting is a major problem but AdaBoost handles it effectively, there is hardly any need to adjust the parameters of the model and during the iterations irrelevant features are assigned low weights by the algorithm. However, because of the noise the performance is lowered, or it underfits if the learning rate is high or the iterations are less. Also, due to the fact that AdaBoost is sequential, nonoptimal first weak learning functions can decrease the overall performance, and diagnostic of several models is difficult.

### XGBoost Regressor

XGBoost or eXtreme Gradient Boosting is one of the efficient boosting algorithms used in Machine learning particularly for regression operations. XGBoost Regressor constructs a multiple tree structure in each step in order to reduce errors. It begins with a single regression tree and subsequently recognizes the residual or errors in training data after which it incorporates new trees to rectify the errors. Every new tree counts more, as it corresponds to important mistakes, while regularization is applied for non-overfitting (Jain, 2016).

The advantage of XGBoost is high efficiency especially for sparse data, the model is capable of handling missing values without imputation and it can be trained on GPU. It is highly flexible with many parameters that can be adjusted and can usually post high accuracy on large machine learning tasks. Nevertheless, it could be extremely prone to overfitting if not regularized appropriately, it tends to require so much time and effort in adjusting its hyperparameters and it is less transparent than its predecessor. Moreover, it requires much computational power and skill level coding to utilize it to the full potential.

## Metrics:

Root Mean Square Error (RMSE) is a well-known measure in the context of regression models which estimates the standard deviation of the residuals between the observed and the predicted values. RMSE is preferred because it focuses greater errors and positive as well as negative errors neither prioritized. A lower RMSE indicates to some extent, the closeness of the prediction to the real value, hence making the model better (Padhma, 2021). However, RMSE have a problem on sensitivity to outliers and it can lead to the skewed assessment, thus, this measure is commonly used together with the measures such as Mean Absolute Error (MAE), which gives the same weight to every error measured. MAE is easier to compute and does not take into account the direction of errors and is less influenced in case of outliers (20\_\_80\_\_, 2018). However, R-squared (R²) describes the amount of variance in the dependent variable that is attributed to the independent variables, and the higher the R² it will be, the better fist it will be. The standard of R² is an imperative for comparing models but creates problems of overfitting hence the use of Adjusted R² in complex models (Orulluoğlu, 2023).

# Chapter 3: Methodology

## Tools and Techniques

* Pandas
* Matplotlib and Seaborn
* Scikit-learn
* Random Forest
* Gradient Boosting and XGBoost
* Decision Tree and Linear Regression
* Train-Test Split
* One-Hot Encoding
* Feature Importance Analysis
* Volume Analysis
* Heatmaps
* Box Plots
* Cross-Validation
* Model Performance Comparison
* Feature Engineering

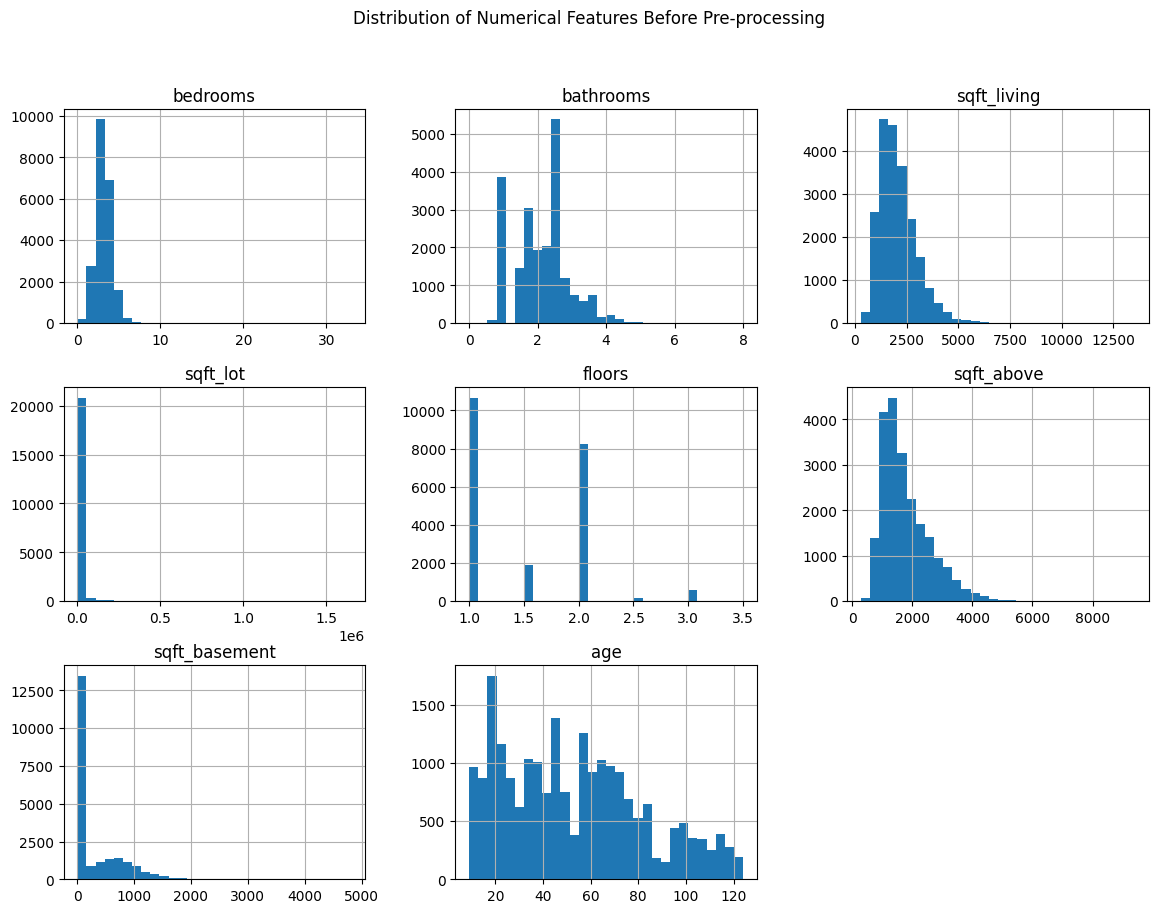
## EDA and Visualization

Exploratory data analysis of the housing dataset consists of calculating, monitoring, analyzing both numerical and categorical variables to outline the data patterns, distribution, or feature that has an impact on models’ performance.

**Numerical Features Distribution:**

Density of the numerical features before performing any kind of data preprocessing was visualized using histograms. They are the basic characteristics of the house like number of bedrooms, number of bathrooms, size of living area (in sq ft.), size of the lot, number of floors, total area of the house above the ground, total area of the house under the ground (basement), and age of the house. The histograms reveal several key insights.

* Bedrooms and Bathrooms: The distribution of the number of bedrooms and number of bathrooms is positively skewed this implies that the majority of the houses possesses a small number of these amenities the majority of the houses possesses a small number of these amenities while a few of them possess more of the amenities.
* Living Area and Above-Ground Square Footage: Has a right-skewed distribution with a peak in frequency for most houses and then little humps in the tails for the extremely high figures for the above-ground square footage for the living area.
* Age of the House: Despite the outliers, the age distribution of the houses seems to be fairly uniform demonstrating that the age of the houses varies significantly in the studied dataset.
* Lot Size and Basement Square Footage: Distribution of the lot size is positively skewed, as most of the values are located in the lower part of the scale, which may suggest the occurrence of outliers, or a necessity to transform values to logarithm scale. Moreover, the same applies to the basement square footage which is also rather inflated, though not to the same degree.

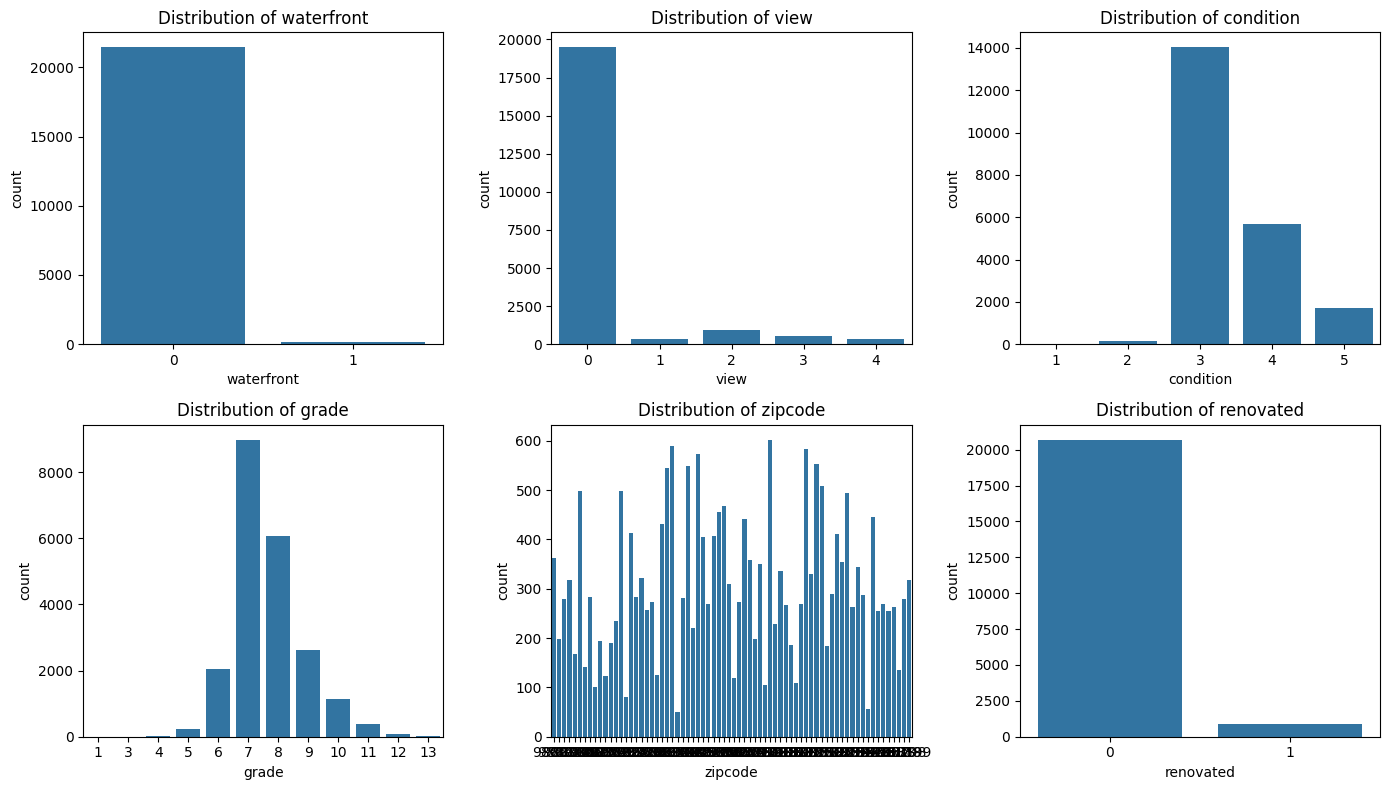


**Missing Values**

The dataset was also made sure free of missing values which is important step in data pre-processing. This indicates that if the missing values are not handled well, the model will be affected and could have biases or inaccurate information. The assessment showed the existence of instances containing missing values in some of the features, which called for imputation or elimination of data records in the pre-processing stage.

**Categorical Features Distribution**

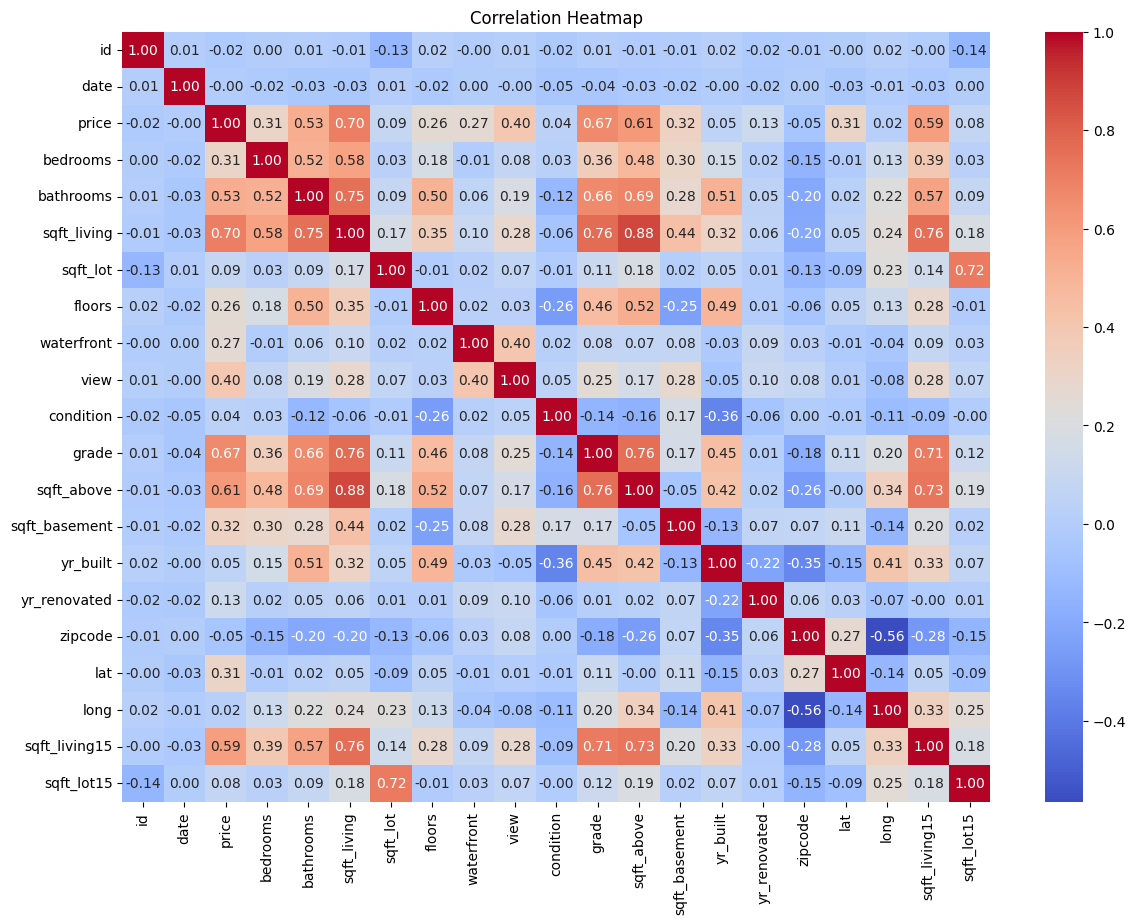
* Concerning the categorical variables, the presence of waterfront, view rating, house condition, grade, and renovation, the count plots were employed in the analysis of the data set. The analysis revealed the following:
* Waterfront and View: A large number of houses do not have a sea frontage, for a rather small number of houses have a sea frontage. Likewise, the magnitude of view also reveals that most houses have low view scores while a few houses score high view scores.
* Condition and Grade: The condition of the houses is, therefore, mainly average with little proportion of substandard or superior houses. The grade distribution of the house proves that the quality of construction and finishes are quite standard, and no extreme quality houses are observed in the dwellings.
* Renovation Status: Over the years, most of the houses featured in the dataset have not been renovated with few percentages of the houses having been renovated.



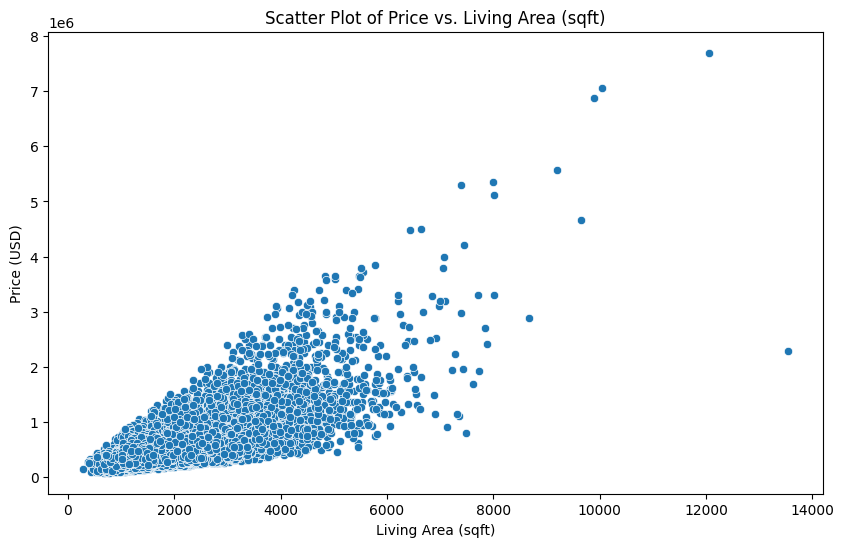
**Correlation Analysis**

Since this research seeks to identify patterns and associations between the chosen features and the target variable that is the price of the house, a correlation heatmap was produced. The heatmap highlights several strong correlations:

* Living Area and Price: The result of the analysis also shows those living area amount, specifically the square feet and house price are positive and strong, meaning greater living space is likely to cost more.
* Grade and Price: Just like in the previous case, the grade of the house also increases with the price, further meaning the higher the quality or luxuriousness of the house, the higher its price will be.
* Bathrooms and Price: Where the number of bathrooms is concerned, there is moderate positive relationship indicating that increase in the number of bathrooms will favor the increase in the price of the house.



Thus, the next type of plot to draw is the one that represents the dependence between the price and the Living area.



Price against living area also demonstrates the same positive direct relationship through a scatter plot. Surprisingly, the models indicate that the size of the living area is directly proportional to the price of the house, although there are certain cases of less expensive houses that overshadow the trend and could be affected by other factors like location and construction quality among others.

**Boxplot of House Prices**

Boxplot of prices further shows us that house price has a number of outliers which implies that although the majority of the houses can be sold at a certain price range or bracket, there are some houses that can only be sold at extremely high prices. This may be due to the fact that such values may require to be handled during the preprocessing phase in order to prevent their impact on the model’s predictive ability.



Therefore, the EDA offers a good start as it offers sufficient information on the structure in the particular dataset and then infers that there will be a need to address issues like missing values, skewness of the data, and the presence of outliers that may require either transformation or deletion. From the results of this process, we can move on to the exercises that make up the construction of a strong predictive model for house prices.

# Chapter 4: Results and Conclusion

## Process Flow

This research adopts a systematic, quantitative approach to evaluate machine learning algorithms for predicting house prices, divided into four key stages: Introduction to feature engineering, Data preprocessing, Model Construction and Validation with emphasis on visualization.

Feature Engineering: The process starts from feature extraction where new variables say ‘age’ and ‘renovated’ are engineered to capture more of the predictive variability. ‘Age’ is derived as the difference between the current year equal to 2024 and year of construction while ‘renovated’ is a dummy variable with a value of one if the house has undergone a renovation.

Data Preprocessing: Data can be analyzed into Numerical Features and Categorical Features. Columns like number of bedrooms, number of baths, size, etc, are normalized with StandardScaler while other qualitative data like building condition, zipcode etc are encoded with OneHotEncoder for a better fine tuning of the model.

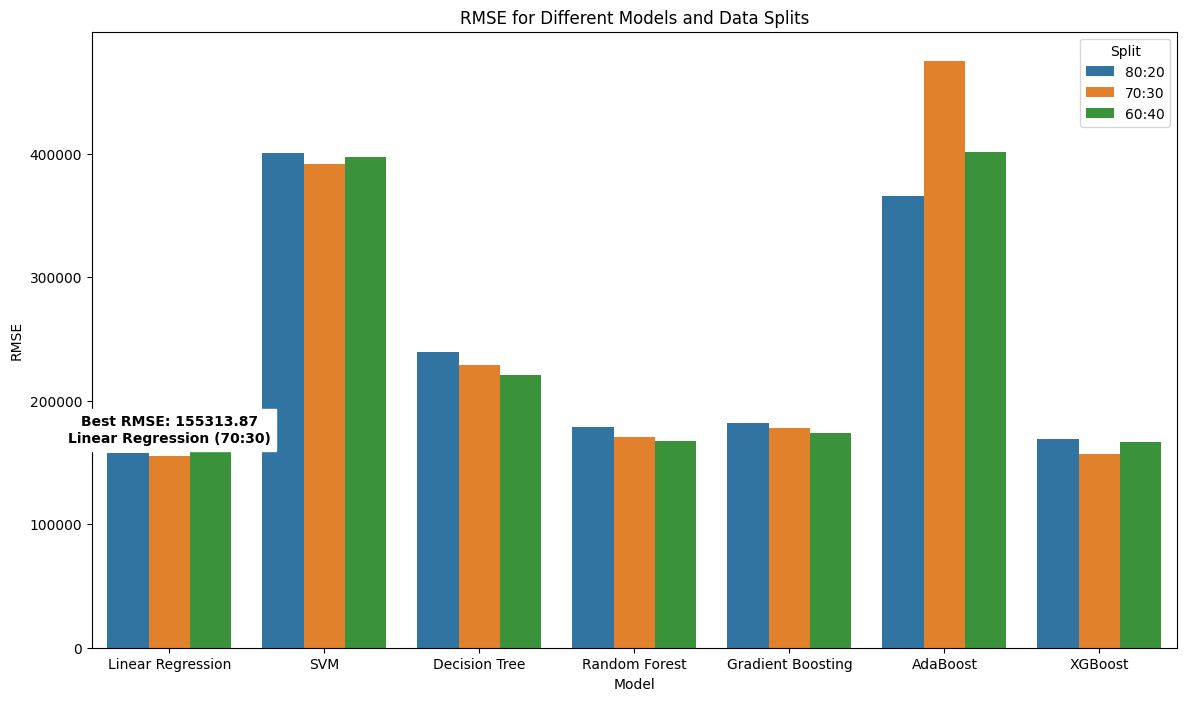
Model Building: About the data splits, the former is as follows: 80% of the data for training purposes, and the remaining 20% for testing. Linear Regression is trained and tested as well as SVR, Decision Tree, Random Forest, and several boosting algorithms are also trained and tested.

Model Evaluation and Visualization: It is often used in model performances assessment where RMSE, MAE, R² and other are employed. A bar plot has been adopted to represent the performance of the data to enable comparison with other models and other data splits to easily identify the best model and data split for every metric.

## Results

The results from the analysis of different machine learning models on the house price prediction task in King County, USA, using three different data splits (80:20, 70:30, and 60:40), provide detailed insights into the performance of these models across various metrics: These measures are Root Mean Square Error or RMSE, Mean Absolute Error or MAE, and Coefficient of Determination or R-squared.

**Root Mean Squared Error (RMSE)**

****

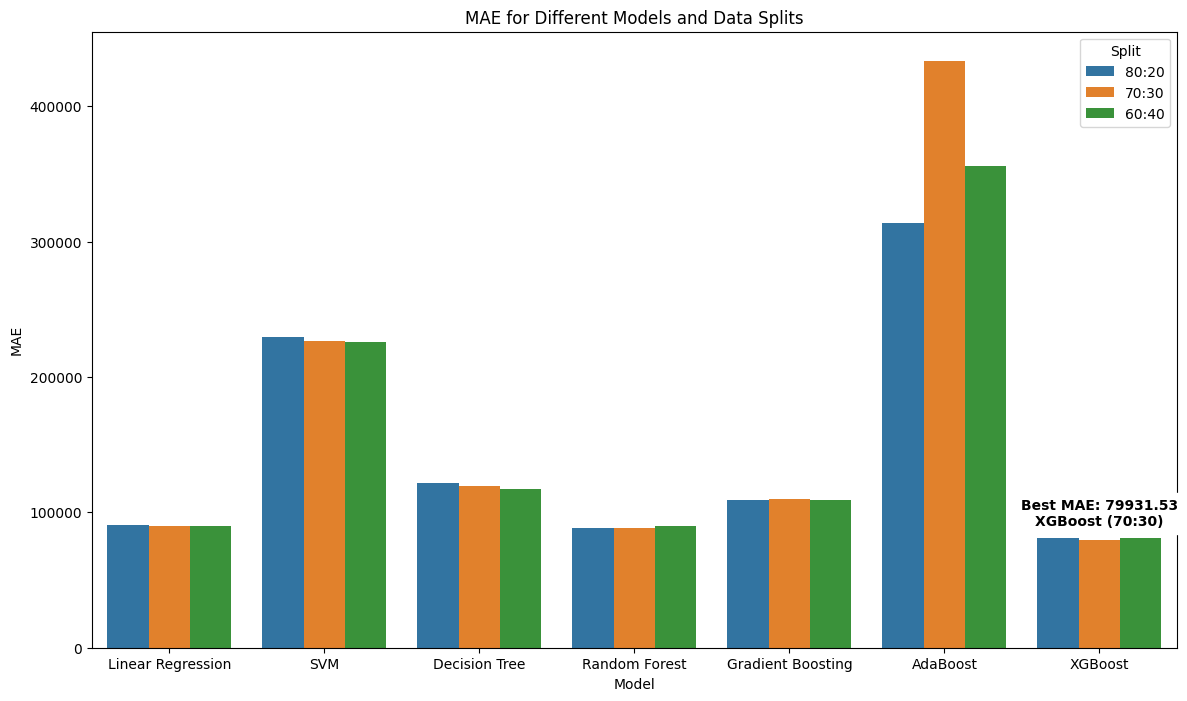
Explained in simple terms, RMSE is an effective measure of the accuracy of the models used in the regression tasks, such as estimating house prices. The objective of making predictions is to get the closest possible estimate to the actual value and the smaller the RMSE value the better are the predictions. In this analysis:

Linear Regression was observed to have the lowest RMSE in every case of data split; the lowest being 155313.; In the 70:30 split, it was 87. This means that though it is simple to use Linear Regression did well in reducing the prediction errors in this dataset.

XGBoost and Random Forest also proved promising yielding an RMSE of 156,980 with XGBoost. for 70:30 in this cross-sectional study, the K-Nearest Neighbor yielded an MAE of 99 in the 187584, 72 in the 70:30 split, Random Forest noted an RMSE of 167,455. 88 where the 60:40 split is attained. These values point towards the idea that this method and ensemble created can detect the intricacies in the housing data of King County.

AdaBoost presented the highest RMSE scores in all the cases, mainly at 70:30 split where RMSE was 475,113. 43. From this, it can be deduced that AdaBoost performed poorly in an ability to capture the relations within the data.

**Mean Absolute Error (MAE)**



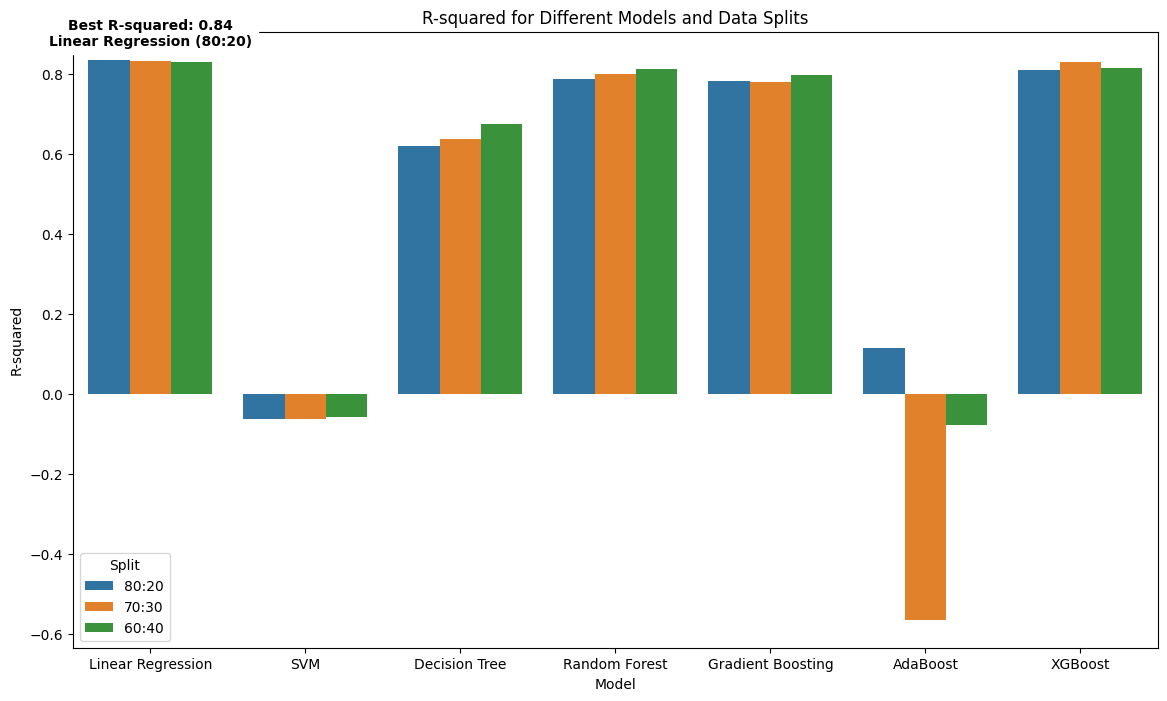
MAE is more interpretable than MSE because it accommodates the average prediction errors while disregarding their direction. The results reveal that:

XGBoost had the lowest MAE of 79,931. 70:30 of employee-client interactions, by achieving an average error rate of 53 out of 100. This takes XGBoost a notch higher in performance as compared to other models making it one of the best in this analysis.

Linear Regression also made quite a good impression and for this model MAE was 89,825 at its best. 70 in the 70:30 split, thus proving that it does not depart significantly from the mark in its predictions.

Specifically, the higher MAEs were observed for SVM and AdaBoost, which amounted to 433,363 for AdaBoost. 26 in the 70:30 split, which just goes on to prove how unfit the 70:30 split ratio is for this dataset.

**R-squared (R²)**

****

R-squared may be defined as a measure that part of the variance outside the line of perfect prediction in the dependent variable (here, house prices) is explained by the variance in the independent variables. Models with higher R-squared ratios display a better fit of the model. The analysis shows:

Linear Regression depicted the highest value of test set R-squared of 0. 8356 in the 80:20; therefore, indicating the extent to which the company is capable of analyzing 83% of the ratios. 31% of the amendment to the variation in King County House prices. This goes further to indicate that Linear Regression is extremely efficient in identifying the facets most influential in the determination of house prices in this area.

The next best model which was XGBoost gave an R-squared value of 0. 8293 for the split 70:30 thus confirmed the model’s effectiveness in explaining the variations in house prices.

Contrary to them, AdaBoost which was also examined in the study had the lowest and even negative R-squared values like -0. 70:30 split reached 5636, which means the model was worse than random as well as detrimental to the performances as compared to the mean-based approach.

**General Observations**

From the analysis of the results above, Linear Regression and XGBoost can be regarded as the most efficient models for house price forecasting in the King County, USA. The performance of both models was stable for all the evaluation criteria implemented, which made it possible to select them as the most suitable for further improvement and application. In sharp contrast, AdaBoost remained below par, meaning that the algorithm is rather ill-suited to this specific data set or it would take much fine-tuning to generate better results with it.

The analysis also shows that the difference in various data splits in terms of the models’ performance was not substantial for the best-performing models, which implies that these are models that are relatively stable and should yield a good margin of generality depending on the amount of data used in their training. Nevertheless, it was also clearly revealed that even models like AdaBoost where specifically sensitive to the data split observed, which strongly pointed to their instability and, thus, poor generalization in this case.

Therefore, the application of the discussed models in practice should be carried out using their performance in the context of certain markets, including the housing market of King County. Consequently, Linear Regression and XGBoost are more feasible to fine-tune, compared to the models such as AdaBoost which could be significantly altered or should not be applied to this type of task.

## Conclusion

This research work deals with the use of different types of machine learning to forecast the price of houses in King County, USA. Thus, the study based on multiple splits in Linear Regression, XGBoost, Random Forest, SVM, Decision Tree, Gradient Boosting, AdaBoost helps to uncover the strengths and weaknesses of each model. The studies show that as a result, Linear Regression and XGBoost models yielded the lowest errors in RMSE, MAE and the highest R-squared values. Given these results, these models are useful tools in real life given that they are able to identify the underlying pattern of housing data.

Other models such as AdaBoost present a very poor performance moving from one data split to another and very poor capability to generalize information. This indicates that even though AdaBoost may work in some circumstances, there may be a necessity of calibration, so the method may not be optimal for the complicated field of house prices in King County.

In addition to the report, the web application using the framework Streamlit was created with regard to the possibility of making interactive predictions of house prices. In the part of this application, users are expected to provide house features such as the number of bedrooms, the number of bathrooms, size of the house among other factors. The model and preprocessor were imported from the ‘joblib,’ and the application uses such inputs to produce forecasts depending on the most suitable machine learning model presented in the research.

It is easy to navigate the designed interface; there is use of drop-downs and number inputs to include attributes such as the condition, grade, and waterfront. When the ‘Predict’ button is pressed on the application, the entered data goes through the preprocessing step in order to shift it to the calling application and show the provided estimate of the house price. This web tool keeps the users involved and engaged by offering the bandwidth to understand the effect of various features on house prices and remains an excellent steppingstone between machine learning and real estate application.

In summary, the findings of this study accord with the view that, for typical predictive tasks improved input pre-processing procedures, cautious selection of features, and choice of the appropriate model are prerequisites to obtaining high levels of accuracy. The findings acquired in this research could be valuable for further works and practical usage in the sphere of real estate analytics and applied machine learning to use such effective algorithms to make predictions.

## Future Work

There are same few directions that topic can be developed in the future to improve the predictive model of house prices even more: The first possibility for the future work is to expand the set of characteristics that define objects and take into account more detailed property characteristics as well as more specific characteristics of neighborhoods, previous price fluctuations, and key economic factors that affect the housing market in certain locations. There is also the possibility of utilizing further techniques such as ‘geographical information systems’ (GIS) integration to investigate the location-specific components of prices further.

One of the potential research areas is perhaps using CNNs or RNNs as they may better learn the data dynamics and temporal features of the housing data compared to most of the conventional algorithms. Also, especially within ensemble techniques that look at the possibility of using more than one model in an attempt to combine their capabilities, we could possibly see enhanced prediction performance.

Still, given the rising need for explainability in machine learning, and since this paper did not address the interpretability of the models, future research could also aim at building interpretable models which will help the decision makers in real estate the key factors that influence the prices of the houses.

Thus, to subsequently introduce other regions or countries into the analysis and improve the reliability of the models would refine them based on regionally unique factors that affect housing markets.

# Chapter 5: Legal, Ethical and Professional Issues

When it comes to house price prediction of the house using machine learning models several legal, ethical and professional issues to be addressed. From a legal point of view, one of the problematic issues is discrimination, for example, redlining. The models that use location-based features that are associated with the protected characteristics such as race or ethnicity may thus end up discriminating in violation of the FH Act. An example of such an issue in the real world is Facebook’s targeted advertising which enabled housing ads to be filtered in a way that resultant in discrimination arising from race, religion, and other characteristics of protected nature. Furthermore, access to and utilization of; copyrighted or patented data and/or algorithms in their decision-making processes is in violation of, and demands proper licensing thus exposing the concerned parties to hefty legal consequences, if not dealt with carefully.

Example: [Facebook Issue with housing prices.](https://www.cnbc.com/2022/06/21/doj-settles-with-facebook-over-allegedly-discriminatory-housing-ads.html)

In other words, ethically, these models have to be applied in real estate with the highest standards of honesty and non-deception. This is because it is only right that if buyers and sellers are to be affected by or make decisions based on the machine learning model’s prediction of property prices then they should be privy to the factors that make up the assessment. An example of ethical issue can be referred to the 2008 Ovation Credit case in which opaque mortgage risk evaluation brought severe credit crunch. This shows why it is critical that there is an open system that checks on unfair actions that may be made by means of a predictive model and to avoid worsening the situation of marginalized groups. At the same time, there is a possibility that the application of such models will strengthen the cycle of negative bias (Intelegain Technologies, 2024) and keep marginalizing those who are in unfavorable conditions.

On the same note, in their professional practice, data scientists and real estate professionals always have to deal with the fact that the forecasts entail uncertainty. These models should therefore not be used as standalone ‘pricing’ tools, but should merely be part of a ‘decision support system.’ Abuse or over reliance without adequate testing could result in millions going up in smoke and companies’ reputations suffering major blows. One of the examples of generating false information is Ziller, which was criticized for the flaws in the Zestimate home price estimator. This case highlights the criterion for validation and the necessity to disclose the imperfections of the considered models to consumers.

# References

20\_\_80\_\_ (2018) *Mean absolute error ~ MAE [Machine Learning(ML)]*, *Medium*. Available at: https://medium.com/@20\_\_80\_\_/mean-absolute-error-mae-machine-learning-ml-b9b4afc63077 (Accessed: August 8, 2024).

*Adaboost for Regression - Example* (2024) *Datamapu.com*. Available at: https://datamapu.com/posts/classical\_ml/adaboost\_example\_reg/ (Accessed: August 8, 2024).

*Deep learning vs. machine learning* (2020) *Zendesk*. Available at: https://www.zendesk.com/in/blog/machine-learning-and-deep-learning/ (Accessed: June 8, 2024).

Dhiraj, K. (2019) *Implementing gradient boosting regression in Python*, *Paperspace Blog*. Available at: https://blog.paperspace.com/implementing-gradient-boosting-regression-python/ (Accessed: August 8, 2024).

Fagbuyiro, D. (2023) *Understanding the support vector machine (SVM) model*, *Medium*. Available at: https://medium.com/@davidfagb/understanding-the-support-vector-machine-svm-model-c8eb9bd54a97 (Accessed: August 8, 2024).

Goyal, S. (2021) *Evaluation metrics for regression models - analytics Vidhya - medium*, *Analytics Vidhya*. Available at: https://medium.com/analytics-vidhya/evaluation-metrics-for-regression-models-c91c65d73af (Accessed: June 8, 2024).

Jain, A. (2016) *XGBoost parameters tuning: A complete guide with python codes*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/ (Accessed: August 8, 2024).

Koehrsen, W. (2017) *Random forest in python*, *Towards Data Science*. Available at: https://towardsdatascience.com/random-forest-in-python-24d0893d51c0 (Accessed: August 8, 2024).

Orulluoğlu, O. (2023) *R-square in machine learning: A powerful tool for evaluating model performance*, *Medium*. Available at: https://medium.com/@bayramorkunor/r-square-in-machine-learning-a-powerful-tool-for-evaluating-model-performance-f90b43e23d9b (Accessed: August 8, 2024).

Padhma, M. (2021) *A comprehensive introduction to evaluating regression models*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/ (Accessed: August 8, 2024).

Ray, S. (2017) *Top 10 machine learning algorithms to use in 2024*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/ (Accessed: June 8, 2024).

The Click Reader (2021) *Decision tree regression explained with implementation in python*, *Medium*. Available at: https://medium.com/@theclickreader/decision-tree-regression-explained-with-implementation-in-python-1e6e48aa7a47 (Accessed: August 8, 2024).

*What is linear regression?* (2010) *Statistics Solutions*. Available at: https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-linear-regression/ (Accessed: August 8, 2024).

Yadav, A. (2018) *Support vector machines(svm)*, *Towards Data Science*. Available at: https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589 (Accessed: August 8, 2024).

Zhang, Z. (2019) *Understand data normalization in machine learning*, *Towards Data Science*. Available at: https://towardsdatascience.com/understand-data-normalization-in-machine-learning-8ff3062101f0 (Accessed: June 8, 2024).

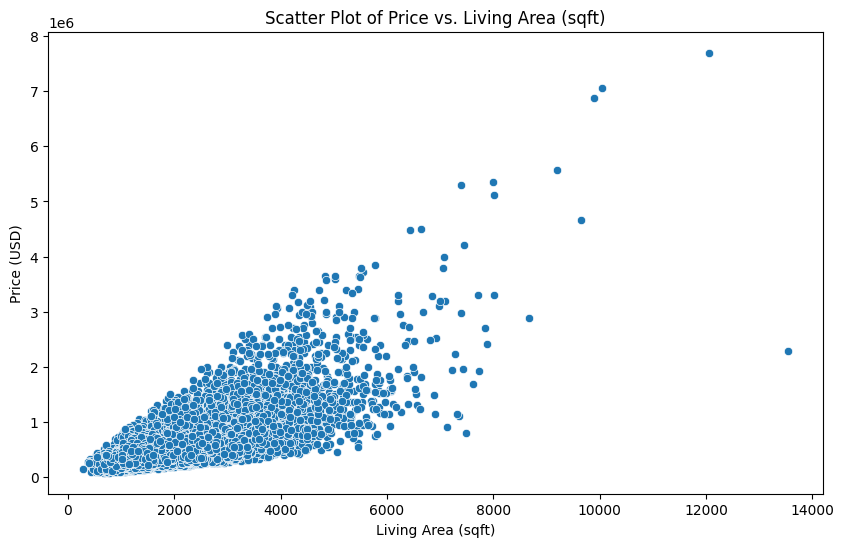
Feiner, L. (2022) *DOJ settles lawsuit with Facebook over allegedly discriminatory housing advertising*, *CNBC*. Available at: https://www.cnbc.com/2022/06/21/doj-settles-with-facebook-over-allegedly-discriminatory-housing-ads.html (Accessed: August 23, 2024).

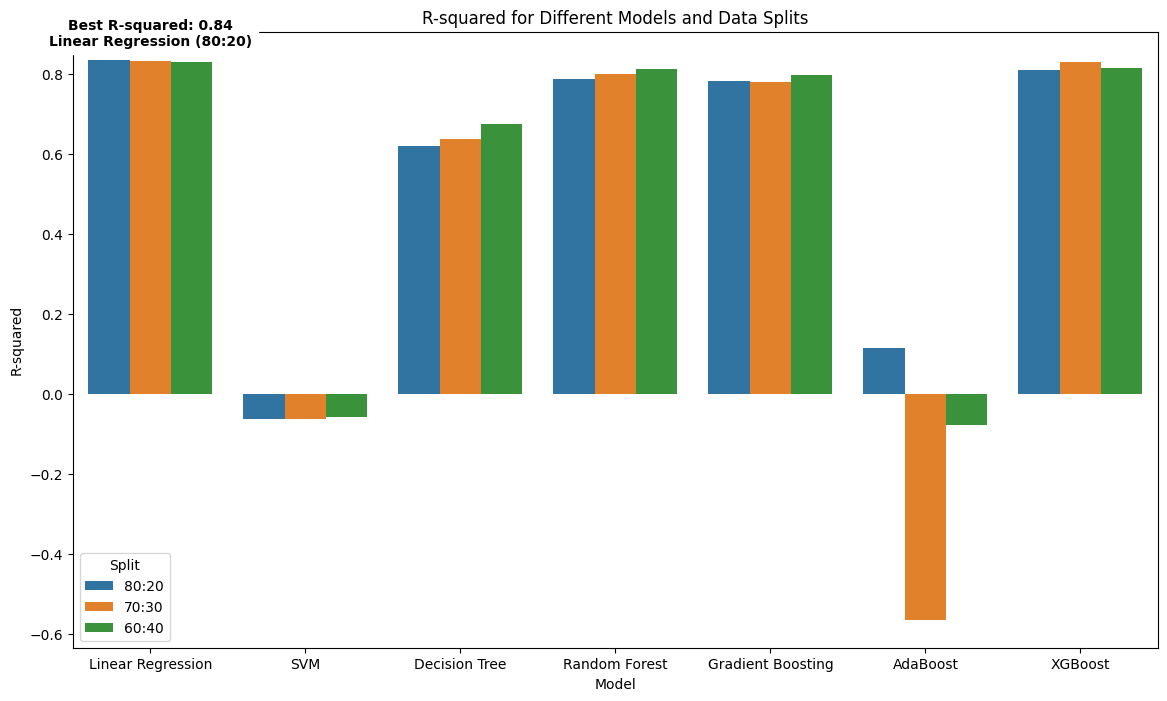
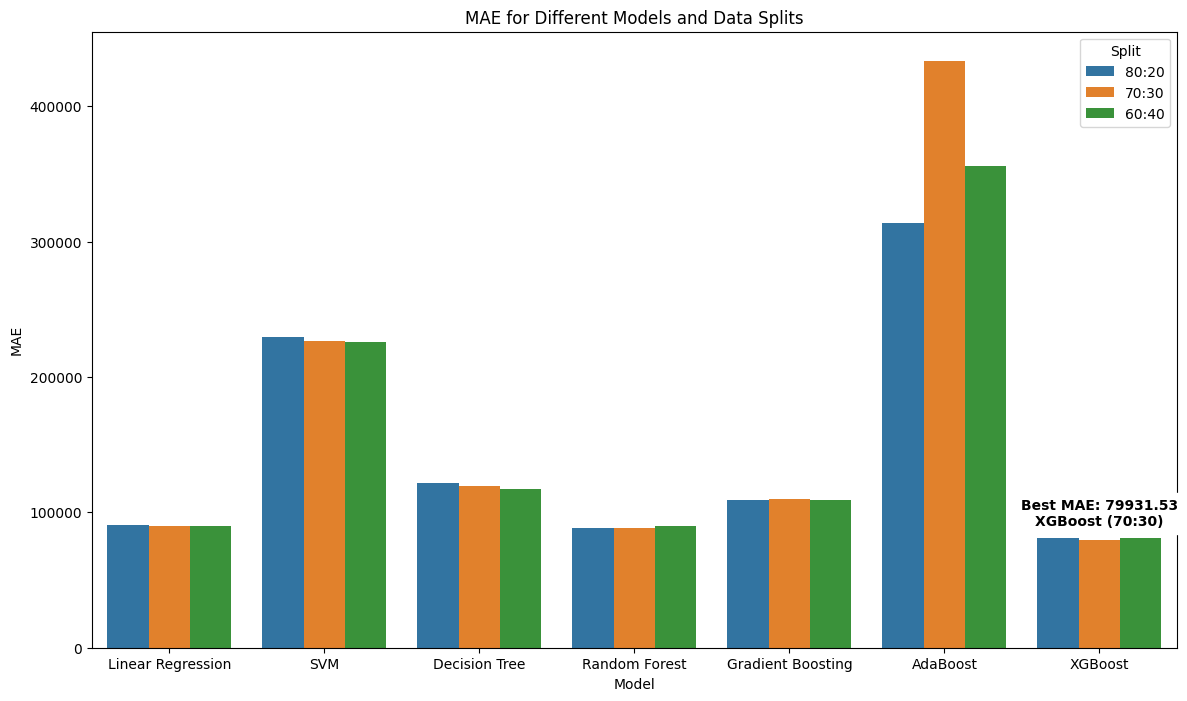
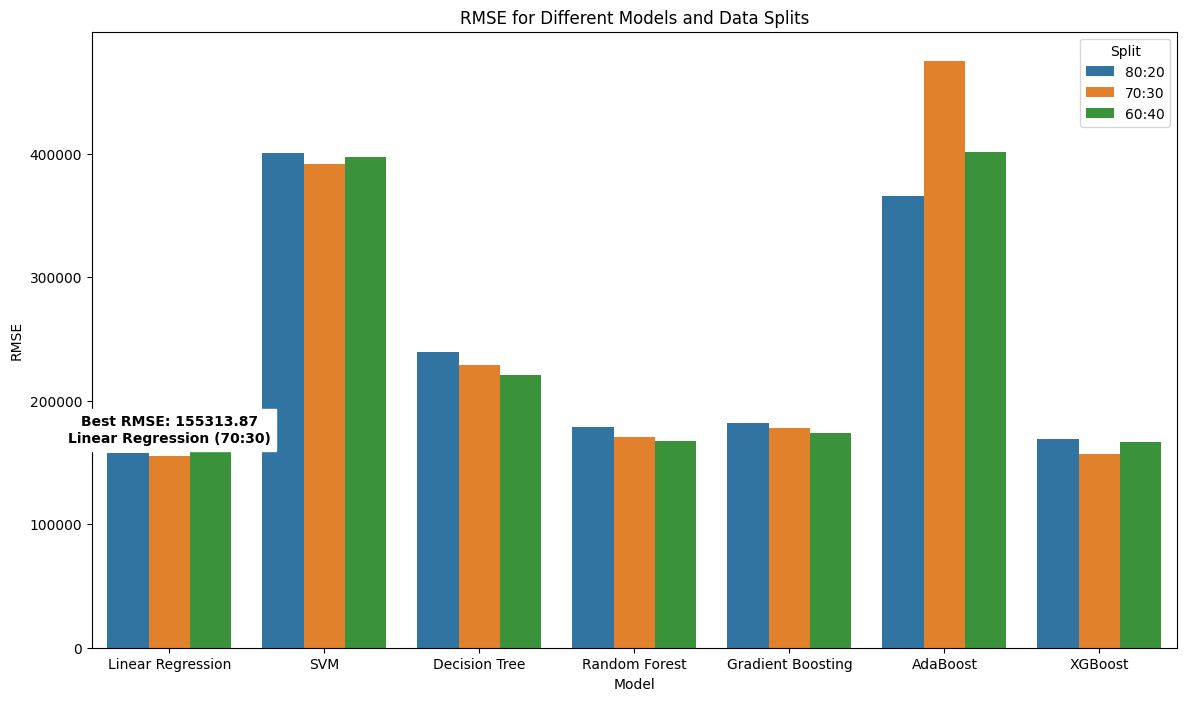
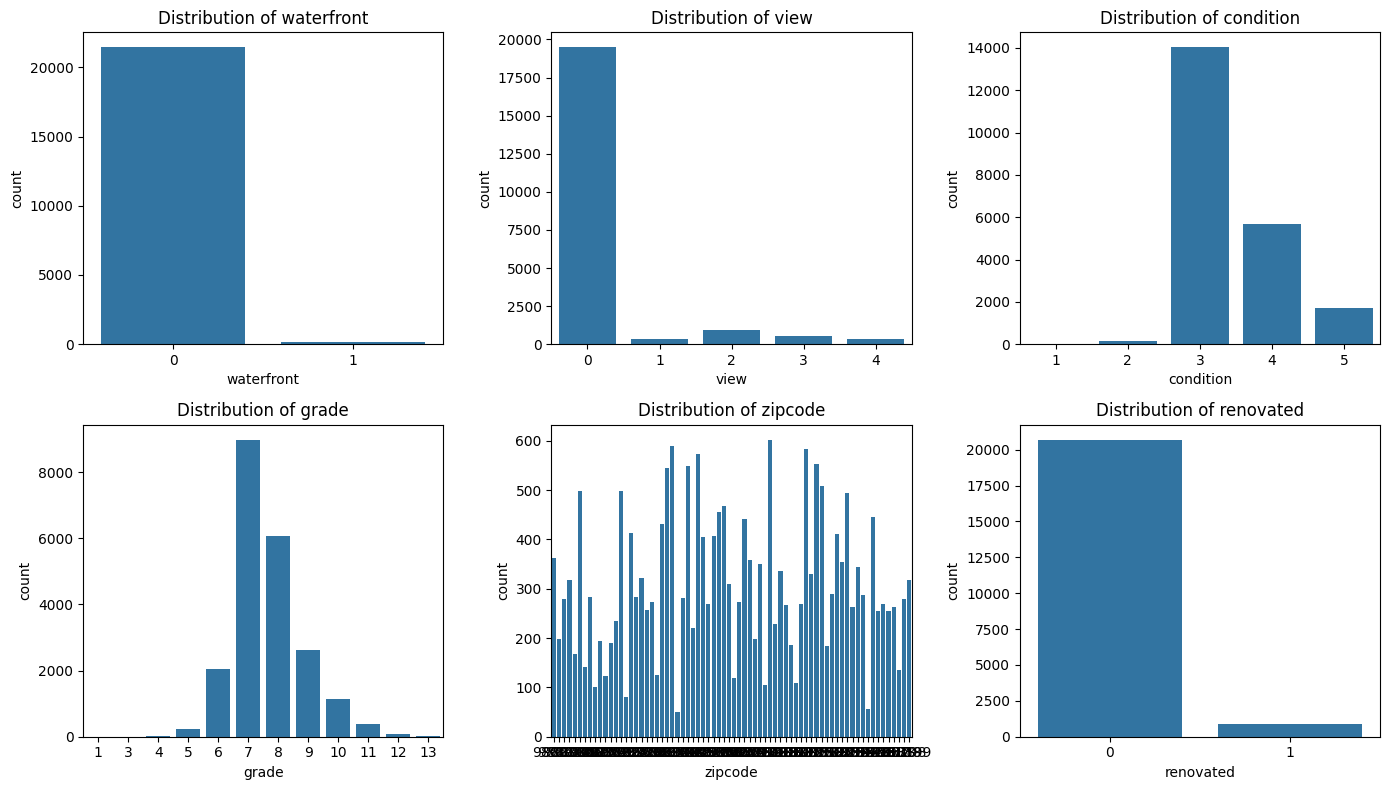
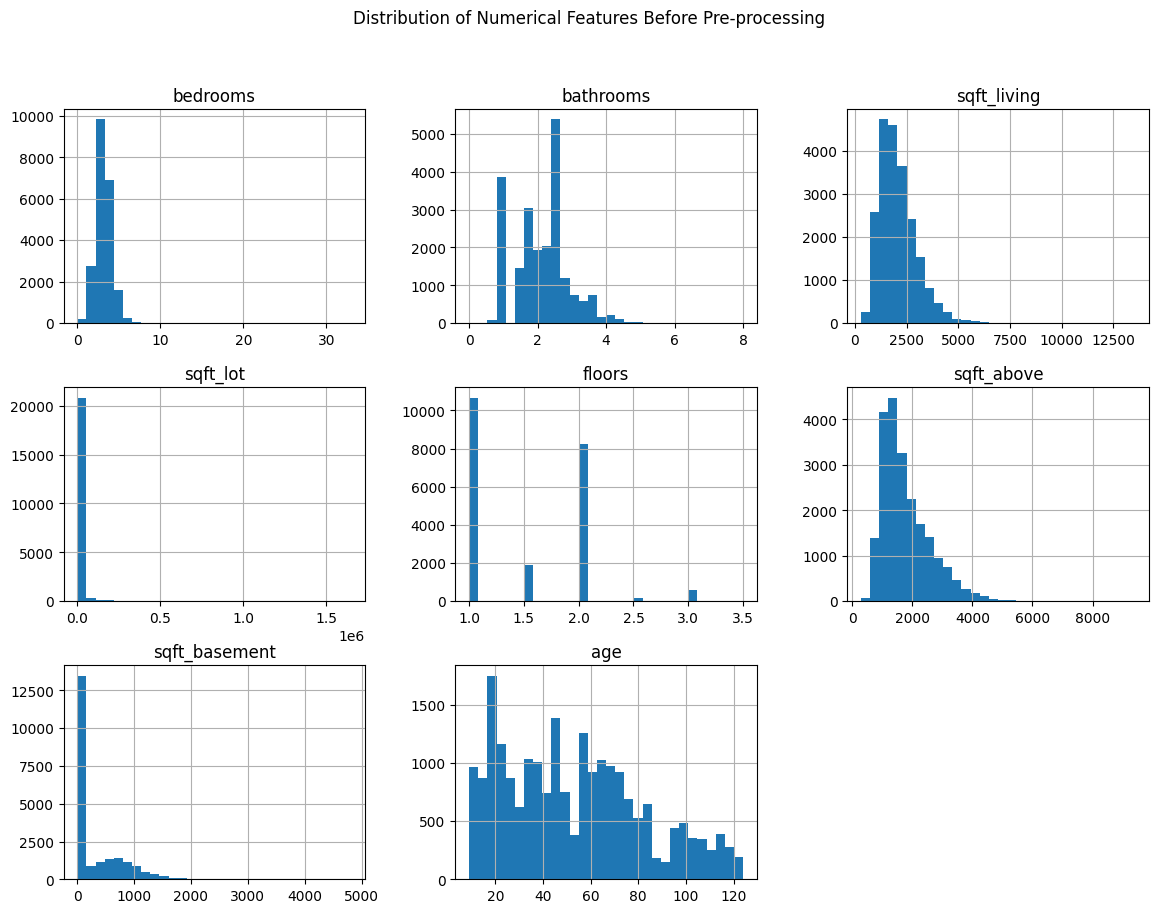
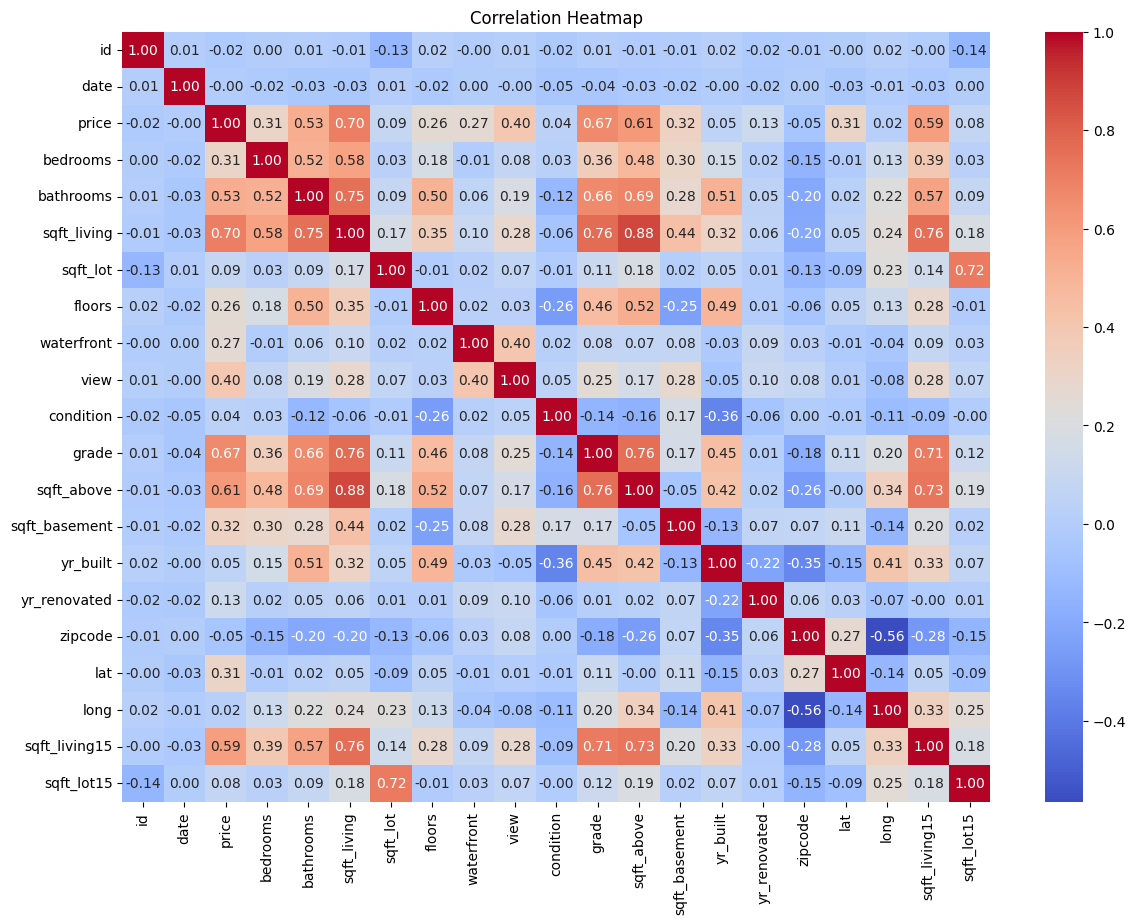
Intelegain Technologies (2024) “Ethical considerations in AI & Machine Learning,” *Intelegain - Enterprise Mobile & Web Application Development company*. Intelegain, 26 March. Available at: https://www.intelegain.com/ethical-considerations-in-ai-machine-learning/ (Accessed: August, 2024).

# Appendices

## Images







## Code I – Research Analysis

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Load the dataset

houses = pd.read\_csv("kc\_house\_data.csv")

houses['date']=pd.to\_datetime(houses['date'])

# Boxplot for price

plt.figure(figsize=(10, 6))

sns.boxplot(x=houses['price'])

plt.title('Boxplot of House Prices')

plt.show()

# Scatter plot for price vs sqft\_living

plt.figure(figsize=(10, 6))

sns.scatterplot(x=houses['sqft\_living'], y=houses['price'])

plt.title('Scatter Plot of Price vs. Living Area (sqft)')

plt.xlabel('Living Area (sqft)')

plt.ylabel('Price (USD)')

plt.show()

# Correlation plot

plt.figure(figsize=(14, 10))

sns.heatmap(houses.corr(), annot=True, fmt='.2f', cmap='coolwarm')

plt.title('Correlation Heatmap')

plt.show()

# Distribution of numerical features before preprocessing

houses[numeric\_features].hist(bins=30, figsize=(14, 10))

plt.suptitle('Distribution of Numerical Features Before Pre-processing')

plt.show()

# Check for missing values

missing\_values = houses.isnull().sum()

print("Missing Values in Dataset:")

print(missing\_values[missing\_values > 0])

# Distribution of categorical features

plt.figure(figsize=(14, 8))

for i, feature in enumerate(categorical\_features):

plt.subplot(2, 3, i + 1)

sns.countplot(x=feature, data=houses)

plt.title(f'Distribution of {feature}')

plt.tight\_layout()

plt.show()

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.impute import SimpleImputer

from sklearn.ensemble import RandomForestRegressor

from sklearn.linear\_model import LinearRegression

from sklearn.svm import SVR

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import GradientBoostingRegressor, AdaBoostRegressor

from xgboost import XGBRegressor

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

# Feature engineering: Create a new feature - house age

houses['age'] = 2024 - houses['yr\_built']

houses['renovated'] = houses['yr\_renovated'].apply(lambda x: 1 if x > 0 else 0)

# Define features and target variable

features = ['bedrooms', 'bathrooms', 'sqft\_living', 'sqft\_lot', 'floors', 'waterfront', 'view',

'condition', 'grade', 'sqft\_above', 'sqft\_basement', 'zipcode', 'lat', 'long', 'age', 'renovated']

target = 'price'

X = houses[features]

y = houses[target]

# Preprocessing: Scaling and encoding

numeric\_features = ['bedrooms', 'bathrooms', 'sqft\_living', 'sqft\_lot', 'floors', 'sqft\_above', 'sqft\_basement', 'age']

categorical\_features = ['waterfront', 'view', 'condition', 'grade', 'zipcode', 'renovated']

preprocessor = ColumnTransformer(

transformers=[

('num', StandardScaler(), numeric\_features),

('cat', OneHotEncoder(handle\_unknown='ignore'), categorical\_features)

])

# Data splitting functions

def split\_data(X, y, test\_size):

return train\_test\_split(X, y, test\_size=test\_size, random\_state=42)

splits = {

'80:20': split\_data(X, y, 0.2),

'70:30': split\_data(X, y, 0.3),

'60:40': split\_data(X, y, 0.4)

}

models = {

'Linear Regression': LinearRegression(),

'SVM': SVR(),

'Decision Tree': DecisionTreeRegressor(random\_state=42),

'Random Forest': RandomForestRegressor(random\_state=42),

'Gradient Boosting': GradientBoostingRegressor(random\_state=42),

'AdaBoost': AdaBoostRegressor(random\_state=42),

'XGBoost': XGBRegressor(random\_state=42)

}

results = []

# Training and evaluation

for split\_name, (X\_train, X\_test, y\_train, y\_test) in splits.items():

X\_train = preprocessor.fit\_transform(X\_train)

X\_test = preprocessor.transform(X\_test)

for model\_name, model in models.items():

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred))

mae = mean\_absolute\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

results.append({

'Split': split\_name,

'Model': model\_name,

'RMSE': rmse,

'MAE': mae,

'R-squared': r2

})

# Convert results to DataFrame

results\_df = pd.DataFrame(results)

import seaborn as sns

import matplotlib.pyplot as plt

# Function to annotate the best metric

def annotate\_best\_metric(ax, metric, results\_df):

best\_value = results\_df[metric].min() if metric in ['RMSE', 'MAE'] else results\_df[metric].max()

best\_row = results\_df[results\_df[metric] == best\_value].iloc[0]

text = f"Best {metric}: {best\_value:.2f}\n{best\_row['Model']} ({best\_row['Split']})"

ax.annotate(

text,

xy=(best\_row['Model'], best\_value),

xytext=(0, 10),

textcoords="offset points",

ha='center',

color='black',

fontsize=10,

weight='bold',

backgroundcolor='white'

)

# Visualization of RMSE

plt.figure(figsize=(14, 8))

ax = sns.barplot(x='Model', y='RMSE', hue='Split', data=results\_df)

plt.title('RMSE for Different Models and Data Splits')

plt.ylabel('RMSE')

annotate\_best\_metric(ax, 'RMSE', results\_df)

plt.show()

# Visualization of MAE

plt.figure(figsize=(14, 8))

ax = sns.barplot(x='Model', y='MAE', hue='Split', data=results\_df)

plt.title('MAE for Different Models and Data Splits')

plt.ylabel('MAE')

annotate\_best\_metric(ax, 'MAE', results\_df)

plt.show()

# Visualization of R-squared

plt.figure(figsize=(14, 8))

ax = sns.barplot(x='Model', y='R-squared', hue='Split', data=results\_df)

plt.title('R-squared for Different Models and Data Splits')

plt.ylabel('R-squared')

annotate\_best\_metric(ax, 'R-squared', results\_df)

plt.show()

results\_df

## Code II – Streamlit Application

import streamlit as st

import joblib

import numpy as np

import pandas as pd

# Load the best model and preprocessor

model = joblib.load('best\_model.pkl')

preprocessor = joblib.load('preprocessor.pkl')

# Streamlit application title

st.title('House Price Prediction App')

# Input fields for the user

bedrooms = st.number\_input('Bedrooms', min\_value=0, max\_value=10, value=3)

bathrooms = st.number\_input('Bathrooms', min\_value=0.0, max\_value=10.0, value=2.0)

sqft\_living = st.number\_input('Living Area (sqft)', min\_value=0, max\_value=10000, value=2000)

sqft\_lot = st.number\_input('Lot Area (sqft)', min\_value=0, max\_value=100000, value=5000)

floors = st.number\_input('Floors', min\_value=0.0, max\_value=5.0, value=1.0)

waterfront = st.selectbox('Waterfront', options=[0, 1])

view = st.selectbox('View', options=[0, 1, 2, 3, 4])

condition = st.selectbox('Condition', options=[1, 2, 3, 4, 5])

grade = st.selectbox('Grade', options=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13])

sqft\_above = st.number\_input('Sqft Above', min\_value=0, max\_value=10000, value=1500)

sqft\_basement = st.number\_input('Sqft Basement', min\_value=0, max\_value=5000, value=500)

zipcode = st.number\_input('Zipcode', min\_value=0, max\_value=99999, value=98000)

lat = st.number\_input('Latitude', min\_value=0.0, max\_value=100.0, value=47.5)

long = st.number\_input('Longitude', min\_value=-180.0, max\_value=180.0, value=-122.0)

yr\_built = st.number\_input('Year Built', min\_value=1800, max\_value=2024, value=1990)

yr\_renovated = st.number\_input('Year Renovated', min\_value=0, max\_value=2024, value=0)

# Compute additional features

age = 2024 - yr\_built

renovated = 1 if yr\_renovated > 0 else 0

# Create a DataFrame for the input features

input\_features = pd.DataFrame({

'bedrooms': [bedrooms],

'bathrooms': [bathrooms],

'sqft\_living': [sqft\_living],

'sqft\_lot': [sqft\_lot],

'floors': [floors],

'waterfront': [waterfront],

'view': [view],

'condition': [condition],

'grade': [grade],

'sqft\_above': [sqft\_above],

'sqft\_basement': [sqft\_basement],

'zipcode': [zipcode],

'lat': [lat],

'long': [long],

'age': [age],

'renovated': [renovated]

})

# Add a predict button

if st.button('Predict'):

# Preprocess the input features

input\_features\_preprocessed = preprocessor.transform(input\_features)

# Make a prediction

predicted\_price = model.predict(input\_features\_preprocessed)

# Display the prediction

st.subheader(f'Predicted House Price: ${predicted\_price[0]:,.2f}')