ABSTRACT

The objective of this report is to analyze the dataset "house-data.csv" from the perspective of classification, prediction, and validation using machine learning methods. The data set comprises information on various attributes of houses, such as their prices, size, location, overall condition and so on. The report is divided into four tasks, including numerical and graphical summarization, classification of houses based on their overall condition, prediction of house prices, and identifying a research sector related to housing data. The report employs various machine learning techniques, including Logistic Regression, Decision Tree, Linear Regression, Random Forest and Re-sampling methods, to accomplish the tasks. The results obtained from the analysis are presented and discussed in detail.

Word Count: 2994

1. INTRODUCTION:

Real estate is a critical and dynamic sector of the economy, where pricing, marketing, and decision-making rely on the analysis of data. Machine learning techniques provide an efficient and effective means of analyzing large amounts of data to identify patterns and make predictions. In this report, we analyze the "house-data.csv" dataset from the perspective of classification and prediction of housing market variables. The dataset contains information on various attributes of residential homes such as the number of bedrooms, lot area, sale price, overall condition and many others. Our objective is to apply machine learning methods to understand the housing market better. We begin by performing exploratory data analysis and summarizing our findings. We then divide the houses based on their overall condition into three categories and use logistic regression and a Decision tree to predict the condition of a house. Next, we predict house prices using Linear Regression and Random Forest. And then estimate test errors using cross validation and bootstrap, re-sampling methods. Finally, we identify a research questions related to the dataset and use the SVM model to answer it.

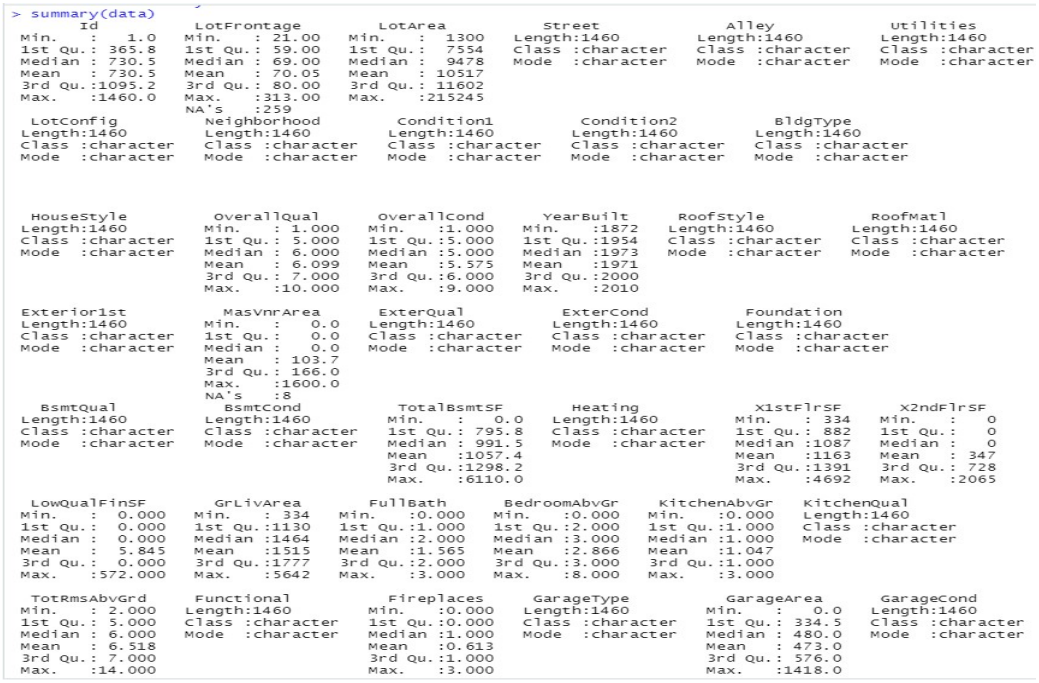
2. PRELIMINARY ANALYSIS (TASK 1):

Exploratory Data Analysis (EDA) is a statistical approach or technique for analysing data sets in order to summarize their important and main characteristics generally by using some visual aids. To describe this data set, we had to rely on statistics, in particular, descriptive statistics.

1.1 NUMERICAL SUMMARY:

Descriptive statistics is the process of evaluating data in such a way as to make them easily understandable. As can be seen from the data set, the given data set consists of 1460 rows and 51 columns.

Descriptive statistics can be summarized visually with graphs and quantitatively with numbers. The numerical representation can be divided into two segments which are the measure of the central tendency of the feature and the measure of variability. The summary of the “house-data.csv” is given in fig 1.



```
> summary(data)
       Id          LotFrontage       LotArea          Street             Alley             Utilities
 Min.   :   1.0   Min.   : 21.00   Min.   :  1300   Length:1460        Length:1460        Length:1460
 1st Qu.: 365.8   1st Qu.: 59.00   1st Qu.:  7554   Class :character   Class :character   Class :character
 Median : 730.5   Median : 69.00   Median :  9478   Mode  :character   Mode  :character   Mode  :character
 Mean   : 730.5   Mean   : 70.05   Mean   : 10517
 3rd Qu.:1095.2   3rd Qu.: 80.00   3rd Qu.: 11602
 Max.   :1460.0   Max.   :313.00   Max.   :215245
                  NA's   :259
  LotConfig         Neighborhood        Condition1         Condition2          BldgType
 Length:1460        Length:1460        Length:1460        Length:1460        Length:1460
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


  HouseStyle         OverallQual      OverallCond      YearBuilt     RoofStyle          RoofMatl
 Length:1460        Min.   : 1.000   Min.   :1.000   Min.   :1872   Length:1460        Length:1460
 Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954   Class :character   Class :character
 Mode  :character   Median : 6.000   Median :5.000   Median :1973   Mode  :character   Mode  :character
                    Mean   : 6.099   Mean   :5.575   Mean   :1971
                    3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000
                    Max.   :10.000   Max.   :9.000   Max.   :2010

 Exterior1st          MasVnrArea      ExterQual          ExterCond          Foundation
 Length:1460        Min.   :   0.0   Length:1460        Length:1460        Length:1460
 Class :character   1st Qu.:   0.0   Class :character   Class :character   Class :character
 Mode  :character   Median :   0.0   Mode  :character   Mode  :character   Mode  :character
                    Mean   : 103.7
                    3rd Qu.: 166.0
                    Max.   :1600.0
                    NA's   :8
   BsmtQual           BsmtCond          TotalBsmtSF       Heating             X1stFlrSF      X2ndFlrSF
 Length:1460        Length:1460        Min.   :   0.0   Length:1460        Min.   : 334   Min.   :   0
 Class :character   Class :character   1st Qu.: 795.8   Class :character   1st Qu.: 882   1st Qu.:   0
 Mode  :character   Mode  :character   Median : 991.5   Mode  :character   Median :1087   Median :   0
                                       Mean   :1057.4                      Mean   :1163   Mean   : 347
                                       3rd Qu.:1298.2                      3rd Qu.:1391   3rd Qu.: 728
                                       Max.   :6110.0                      Max.   :4692   Max.   :2065

  LowQualFinSF       GrLivArea       FullBath       BedroomAbvGr    KitchenAbvGr    KitchenQual
 Min.   :  0.000   Min.   : 334   Min.   :0.000   Min.   :0.000   Min.   :0.000   Length:1460
 1st Qu.:  0.000   1st Qu.:1130   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000   Class :character
 Median :  0.000   Median :1464   Median :2.000   Median :3.000   Median :1.000   Mode  :character
 Mean   :  5.845   Mean   :1515   Mean   :1.565   Mean   :2.866   Mean   :1.047
 3rd Qu.:  0.000   3rd Qu.:1777   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:1.000
 Max.   :572.000   Max.   :5642   Max.   :3.000   Max.   :8.000   Max.   :3.000

  TotRmsAbvGrd      Functional          Fireplaces     GarageType          GarageArea      GarageCond
 Min.   : 2.000   Length:1460        Min.   :0.000   Length:1460        Min.   :   0.0   Length:1460
 1st Qu.: 5.000   Class :character   1st Qu.:0.000   Class :character   1st Qu.: 334.5   Class :character
 Median : 6.000   Mode  :character   Median :1.000   Mode  :character   Median : 480.0   Mode  :character
 Mean   : 6.518                      Mean   :0.613                      Mean   : 473.0
 3rd Qu.: 7.000                      3rd Qu.:1.000                      3rd Qu.: 576.0
 Max.   :14.000                      Max.   :3.000                      Max.   :1418.0
```

***Fig 1*: Summary**

A few methods were applied to view the dataset accurately such as listing the data types using str(data) command, viewing the first and last 5 elements in the dataset using head(data) and tail(data). To perform statistical analysis on a dataset, it is often necessary to separate the variables into different categories based on their data type. One common way to do this is by classifying variables as either numerical or categorical using is.numeric() and is.character() functions.

The first part of the code involves classifying the variables in the dataset into numerical and categorical types. The select_if function is used to select columns that are either numeric or character types, and the results are stored in separate data frames for numerical and categorical variables. The code “colnames” is used to retrieve the column names for both data frames.

The next step is to count the number of missing values in each column of the dataset using the colSums and is.na functions. Subsequently, the missing values in the categorical variables of the dataset are replaced with the given descriptions. This is done using the replace function. The specific categorical variables that have missing values are Alley, PoolQC, Fence, BsmtQual, BsmtCond, GarageType, GarageCond, and MiscFeature.

The replaced values are "No Alley Access", "No Pool", "No Fence", "No Basement", "No Garage", and "None" respectively. The code colSums(is.na(data)) rechecks the null values using colSums to confirm that the replacements have been made successfully.

The dataset includes different characteristics of houses, such as sale price, garage area, overall condition, year of built, and many others. The summary statistics indicate that the average sale price is $180,921.2 with a standard deviation of $79,442.5.

1.2 GRAPHICAL SUMMARY:

The barplot to visualize the distribution of the variable 'Overall Condition' of houses on the market. It shows that the majority of houses are in 'average' condition, while a smaller number of houses are in 'excellent' or 'poor' condition (Fig 2).The bar plot was created to show the number of houses built in different years (Fig 3).

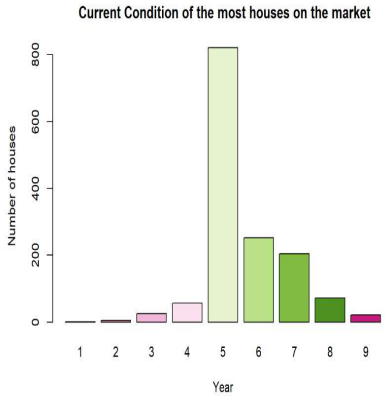


Fig 2: Barplot showing Overall condition of houses

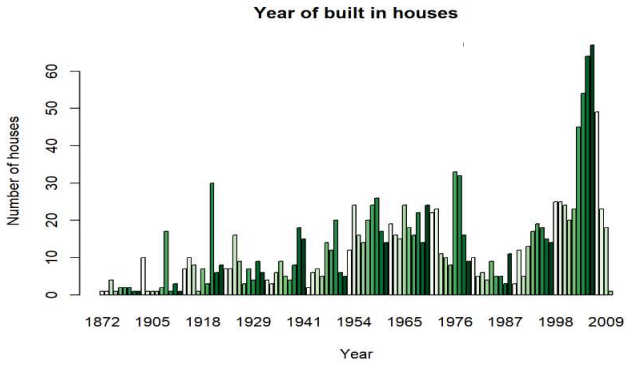


Fig 3: Barplot showing age of houses

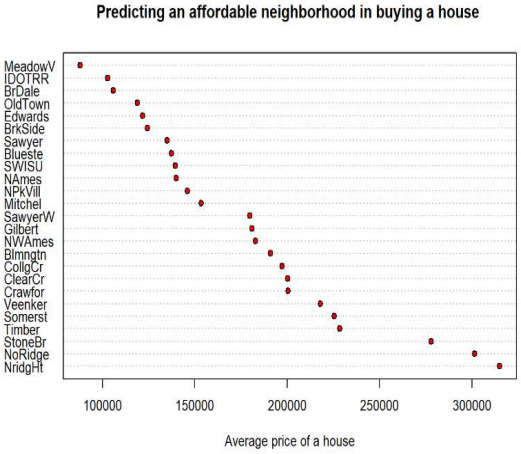


Fig 4: Predicting affordable housing location

Next, a dot chart was used to visualize the median sale prices of houses in different neighborhoods helping an analysis of finding affordable neighbourhoods (Fig 4).

To prepare the data for correlation analysis, the numerical values were filtered and null values were imputed using the 'pmm' method. A correlation matrix was then plotted to display the pairwise correlations between different variables (Fig 5).

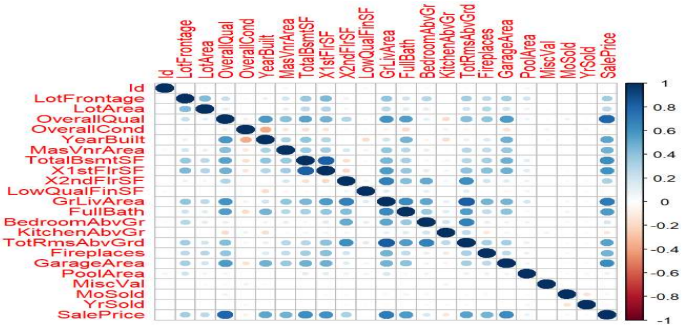


Fig 5: Correlation plot of numerical variables in the dataset

Additionally, specific correlation coefficients were calculated between the sale price and several other variables such as the lot frontage, year built, year sold, overall condition, overall quality, and number of bedrooms.

3. PREDICTING THE OVERALL CONDITION OF A HOUSE (TASK 2):

The second task in this work is to predict the overall condition of the house using any Machine Learning models. As this is a classification type of problem, many classification algorithms exists.

We have used two classification Algorithms:

i) Logistic Regression
ii) Decision Tree

After imputing missing values in numerical variables, LotFrontage and MasVnrArea using the median value, we converted the categorical variables into numerical variables using one-hot encoding and combined them with the numerical variables. The OverallCond variable was categorized as Poor, Average, or Good, based on the rating of the house's overall condition on a scale of 1 to 10. Houses with an overall condition rating of 1 to 3 were categorized as Poor, those with a rating of 4 to 6 were categorized as Average, and those with a rating of 7 to 10 were categorized as Good, before fitting in the model.

3.1 LOGISTIC REGRESSION MODEL:

The logistic regression classification model was used to predict the overall condition (OverallCond) of houses based on a dataset. The dataset was split into a training set and a testing set using the createDataPartition function from the caret package, with 80% of the data allocated to the training set and the remaining 20% allocated to the testing set. The training set consisted of 1,169 observations, while the testing set consisted of 291 observations.

A logistic regression model was then fitted to the training set using the multinom function from the nnet package, with the OverallCond variable as the response variable and all other variables in the dataset as predictors. The model was fitted using a binomial family, and the seed was set to 123 to ensure reproducibility of the results.

The summary function was used to print out the summary of the logistic regression model, which showed that the model had a final value of 256.43 and was stopped after 100 iterations.

The model was then used to predict the overall condition of the houses in the testing set, and the predicted classes were compared to the actual classes using the mean function. The accuracy of the predicted classes was found to be 78.4%. Overall, the logistic regression model performed well in predicting the overall condition of houses based on the given dataset.

3.2 DECISION TREE MODEL:

The second method to classify the house condition is decision tree classifier model. The goal of the model is to predict the overall condition of houses based on various features provided in the dataset. The first step in building the model is to create a decision tree using the rpart() function. The formula used for the decision tree is "OverallCond~ .", where OverallCond is the dependent variable and "." represents all the independent variables in the training dataset.

Once the decision tree model is created, the rpart.plot() function is used to visualize the tree (Fig 6).This helps in understanding the rules created by the model to make predictions. After visualizing the model, the next step is to make predictions on the testing dataset using the predict() function.

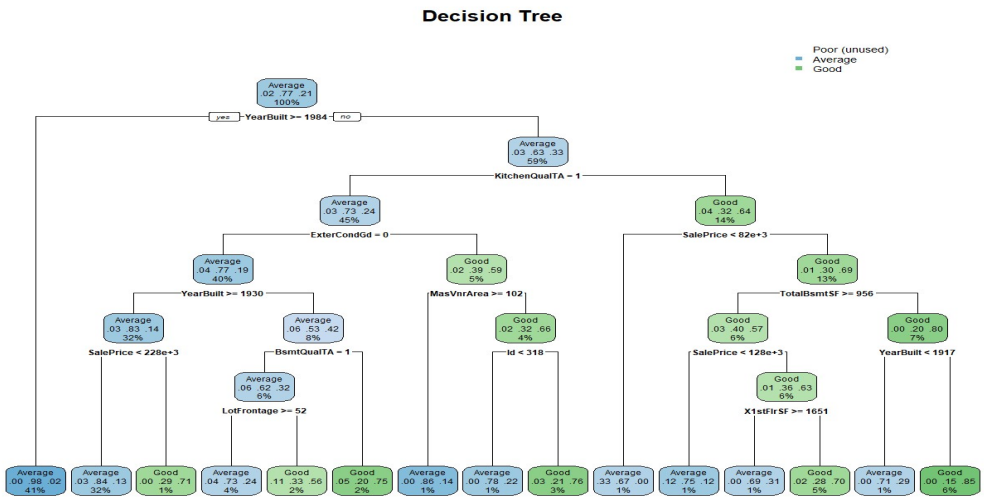


Fig 6: Decision Tree with house.csv data

The "type = 'class'" argument is used to ensure that the model predicts a class label for each observation in the testing dataset. Finally, the accuracy of the model is evaluated using a confusion matrix. The table() function is used to create a confusion matrix, which shows the number of true positive, true negative, false positive, and false negative predictions made by the model. The accuracy of the model is calculated by dividing the sum of diagonal values in the confusion matrix by the sum of all values in the matrix. The model correctly predicted the overall condition of the houses in the testing dataset with an accuracy of 83%.

4. PREDICTING HOUSE PRICES (TASK 3):

The main aim of this task is to predict the house saleprices based on features given. The analysis performed on the dataset involved the subsetting of the original dataset into training and testing data. The training set consists of 80% of the original dataset, and the remaining 20% was assigned to the testing set.

To make an effective prediction, the Outliers in data were identified and removed in the training set and new training and testing set were created. The dataset was then plotted with and without the outliers to visualize the effect of removing the outliers on the slope of the regression line (Fig 7). The change in slope was evident in the plots, indicating that removing the outliers had a significant effect on the model's performance.

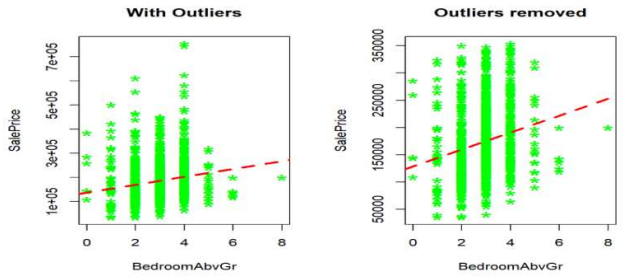


Fig 7: Data Plotting with and without outliers

Overall, this analysis shows that removing outliers can improve the performance of the model by removing any significant deviation from the data's overall trend. By removing outliers, the model's performance can be optimized, leading to more accurate predictions.

The two machine learning models used to predict house prices are:

i) Linear regression model

ii) Random forest model.

4.1 LINEAR REGRESSION MODEL:

The first model is a linear regression model, which has been fitted to the training data using the 'lm' function. The model has been evaluated using the testing dataset, and its performance has been measured using three metrics: root mean squared error (RMSE), R-squared, and mean absolute error (MAE). The metrics obtained after model fitting is RMSE = 15974.91, R-squared = 0.93, MAE = 11987.45. The R-squared value was found to be 0.93, indicating that 93% of the variability in the target variable was explained by the model.

4.2 RANDOM FOREST MODEL:

The second model is a random forest model, which has been fitted to the training data using the 'randomForest' function. This model has also been evaluated using the testing dataset, and its performance has been measured using the same three metrics: RMSE, R-squared, and MAE. The RMSE for this model was found to be 45215.31, indicating that the model's predictions were off by an average of $45,215.31. The R-squared value was found to be 0.79, indicating that 79% of the variability in the target variable was explained by the model. The MAE for this model was found to be 22426.11.

Comparing the two models, it can be observed that the linear regression model performed better than the random forest model in terms of all three-evaluation metrics: RMSE, R-squared, and MAE. The linear regression model had a lower RMSE, a higher R-squared, and a lower MAE, indicating that it was a better model for predicting house prices in this dataset.

Model	Linear Regression	Random Forest
RMSE	15974.91	45215.31
R-squared	0.93	0.79
MAE	11987.45	22426.11

Table 1: Linear Regression vs Random Forest

4.3 RE-SAMPLING METHODS:

Re-sampling methods are used to estimate the test error associated with fitting a model. The two best re-sampling methods to estimate the test error are

i) Cross-validation (CV) Method
ii) Bootstrap Method

4.3.1 CROSS-VALIDATION METHOD:

Cross-validation is a technique used to assess the performance of predictive models in machine learning and statistical modelling. It involves splitting the data into subsets, training the model on a subset, and testing it on the remaining data. The process is repeated multiple times, and the results are averaged to obtain a more robust estimate of model performance. After 10 folds of cross validation, the RMSE obtained in linear model is 43355.43. Surprisingly the original Linear model fitted best with RMSE of 15974.91. However the Root mean squared error for original Random Forest is RMSE = 45686.56 whereas the re-sampled CV model gives RMSE = 29672.2 making sampled model effective than original random forest model.

4.3.2 BOOTSTRAP SAMPLING METHOD:

Bootstrap is a re-sampling technique used to estimate the variability and distribution of a statistic from a sample of data. It involves randomly sampling the data with replacement to create multiple bootstrap samples. A statistic is calculated for each bootstrap sample, and the distribution of the statistic is estimated from the collection of bootstrap statistics. Bootstrap can be used to estimate standard errors, confidence intervals, and hypothesis tests for a wide range of statistics, including means, variances, proportions, regression coefficients, and more. As per the work, the original linear model fitted best than bootstrap, having RMSE in original as 15974.91 and Bootstrap RMSE as 41530.83. Whereas the bootstrap Random Forest model fitted effectively than original random forest model having RMSE values as 45686.56 and 29188.8.

Metric	**Linear Model**			**Random Forest**		
	Original	*CV Model*	*Boot*	*Original*	*CV*	*Boot*
RMSE	15974.91	43355.43	41530.83	45686.56	29672.2	29188.8

Table 2: RMSE of original and Sampled Models

5. RESEARCH INVESTIGATION (TASK 4):

A research was found interesting in predicting the Garage condition. Based on prior knowledge of the housing market and the factors that could influence the condition of the garage, the significant features are chosen. The selected independent features are YearBuilt, GarageType, GarageArea, Fence, and Neighborhood for predicting garage condition, as they could provide valuable insights into the age, size, location, and overall quality of the garage. The Support Vector Classification (SVM) model was taken for this prediction. Before fitting the SVM model, the dataset was pre-processed to ensure that the selected features were suitable for analysis. This involved cleaning the data and replacing missing categorical values with relevant non-null values, such as "NoGarage" and "NoFence". Additionally, categorical variables were converted to numerical values for ease of visualization and analysis.

Exploratory data analysis was conducted to gain a better understanding of the data and identify any patterns or relationships between the selected features and garage condition. A bar graph was created to visualize the distribution of garage condition values, providing a

quick overview of the frequency of each condition. Scatter plots were also generated for each selected feature, enabling the visualization of each variable's relationship with garage condition. The SVM model was chosen for its ability to handle both linear and non-linear data and its effectiveness in handling large datasets. The train-test split methodology was used to evaluate the model's performance, with 80% of the data used for training and 20% used for testing. The SVM model was fitted with the selected features, and the testing dataset was used to evaluate the model's accuracy. The confusion matrix was used to calculate the model's accuracy, which produced an accuracy of 0.969. This indicates that the model was able to correctly predict the garage condition in 96.9% of the cases.

In conclusion, the study has successfully predicted the Garage Condition using selected features from a housing dataset with a high level of accuracy. Additionally, the methods and techniques used in this study could be applied to other datasets to predict various types of property conditions.

6. CONCLUSION:

In conclusion, the analysis of the housing dataset using machine learning techniques has provided valuable insights into the factors that affect house prices and overall condition. The exploratory data analysis revealed that several variables have a significant impact on house prices, including the size of the house, the quality of the construction, and the location of the property. We used logistic regression and decision tree method to predict the overall condition of houses, and both methods performed reasonably well. A linear regression and Random Forest in a justified manner evaluated the Saleprice. In addition, we employed sampling strategies Bootstrap and Cross-Validation to tune the prediction done by previous models making reliable predictors of house prices. Finally, we identified several areas for future research, including predicting garage condition using selected features from a housing dataset with a high level of accuracy and exploring how different machine learning algorithms can be used to improve the accuracy of the predictions. Overall, the analysis has provided valuable insights into the housing market and demonstrated the usefulness of machine learning in understanding complex datasets.