**Capstone Project**

**House Price Prediction**

**1. Introduction**

**Problem Statement:**

A house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. For example, you want to sell a house and you don’t know the price which you may expect, it can’t be too low or too high. One would like to buy a house at the best rate and minimum risk and would like it to be the best investment for the future. Various online websites, real estate agents and realtors try to guide home buyers by letting them compare different houses available for purchase. To find house price you usually try to find similar properties in your neighborhood and based on gathered data you will try to assess your house price.

**Need of the study/project:**

House prices increase every year, so there is a need for a system to predict house prices in the future. The aim of the project is to provide the best-selling price of the house that can help the national real estate developer, individual buyer looking for a house to purchase.

Take advantage of all of the feature variables available below, use it to analyze and predict house prices.

|  |
| --- |
| 1. **cid:** a notation for a house 2. **dayhours:** Date house was sold 3. **price:** Price is prediction target 4. **room\_bed:** Number of Bedrooms/House 5. **room\_bath:** Number of bathrooms/bedrooms 6. **living\_measure:** square footage of the home 7. **lot\_measure:** square footage of the lot 8. **ceil:** Total floors (levels) in house 9. **coast:** House which has a view to a waterfront 10. **sight:** Has been viewed 11. **condition:** How good the condition is (Overall) 12. **quality:** grade given to the housing unit, based on grading system 13. **ceil\_measure:** square footage of house apart from basement 14. **basement\_measure:** square footage of the basement 15. **yr\_built:** Built Year 16. **yr\_renovated:** Year when house was renovated 17. **zipcode:** zip 18. **lat:** Latitude coordinate 19. **long:** Longitude coordinate 20. **living\_measure15:** Living room area in 2015(implies-- some renovations) This might or might not have affected the lot size area 21. **lot\_measure15:** lot Size area in 2015(implies-- some renovations) 22. **furnished:** Based on the quality of room 23. **total\_area:** Measure of both living and lot |
|  |

**Understanding business/social opportunity:**

Based on house prices predicted one can invest in real estate, find a county house better suited for their needs where they can buy a house. House buying and selling decision would become easier with the prediction done by this data science project.

1. **EDA and Business Implication**

**Univariate analysis:**

**Data Distribution for Continuous Data:**

Graphical user interface, chart, bar chart

Description automatically generated

Graphical user interface, chart, application

Description automatically generated

Inference:

* From above plot it is evident that there is skewness in the data.
* Variables such as **basement,ceil\_measure,living\_measure, living\_measure15,lot\_measure15** are positively skewed.
* **Yr\_build** is negatively skewed.

**House Price distribution:**

Chart, scatter chart

Description automatically generated

From above plat it is evident that as the price increases number of buyers reduces and most of buyers interested to buy houses with moderate price.

**Distribution of house based on No of bedrooms/bathroom/floors/grading:**

Graphical user interface, chart

Description automatically generated

Inference:

* From plot it is evident that as number of bedrooms increases, buyers decreases and most of the people would like buy flats with 3 or 4 bedrooms.
* Flats with two bathrooms are most popular and then one bathroom. But it seems customers are less interested to buy house with more than 3 bathrooms.
* Majority of the house are with one floor, but other choice is house with two floors.
* Most of the houses are of quality 7.

**Distribution based on waterfront view/house condition/sight and furnished:**

Graphical user interface

Description automatically generated

Inference:

* Majority of the houses are with waterfront view.
* Majority of the houses has moderate house condition.
* Most of the people have not seen the house.
* Most of the houses are not furnished.

**Distribution on year build:**

Chart, bar chart, histogram

Description automatically generated

Inference:

* From above plot it is evident that highest number of houses have built on 2014 and lowest build date of house least for 1934.

**Bi-variate / Multi-variate analysis:**

**House Price based vs Number of Floors:**

Chart, box and whisker chart

Description automatically generated

Inference:

* From above box plot it is evident that second floor has the high price of the house with many outliers.
* Highest price of the house has 2.5 floors.

**House Price Vs Sight:**

Chart, box and whisker chart

Description automatically generated

Inference:

* From above plot it is evident that range of the house price is more for houses have been seen four times and there are many outliers in the price.

**House Price Vs Condition:**

Chart, box and whisker chart

Description automatically generated

Inference:

* House with moderate condition has highest prices.

**House Price vs Grade:**

Chart, box and whisker chart

Description automatically generated

Inference:

* From above box plot it is evident that as the grade increases, price range of the house also increases and it highest with grade 13.

**Price vs Month:**

Chart, line chart

Description automatically generated

A picture containing text, indoor, receipt

Description automatically generated

Inference:

* From above plot between month and price it is evident that price trend increases from second month to fourth month and then there is decrease in price.
* Price is lowest for second month and heights for fourth month.
* Most of the property have highest mean price observed during April - 2015, May -2015 and June-2014.

**Price Vs Year:**

Chart, line chart

Description automatically generated

* There is increasing trend of price from year 2014 to 2015. As the year increases price of house also boosts up.

**Home Price Vs Home Sqft/Lot Sqft/Basement Sqft/Celi Measure**

Graphical user interface, scatter chart

Description automatically generated

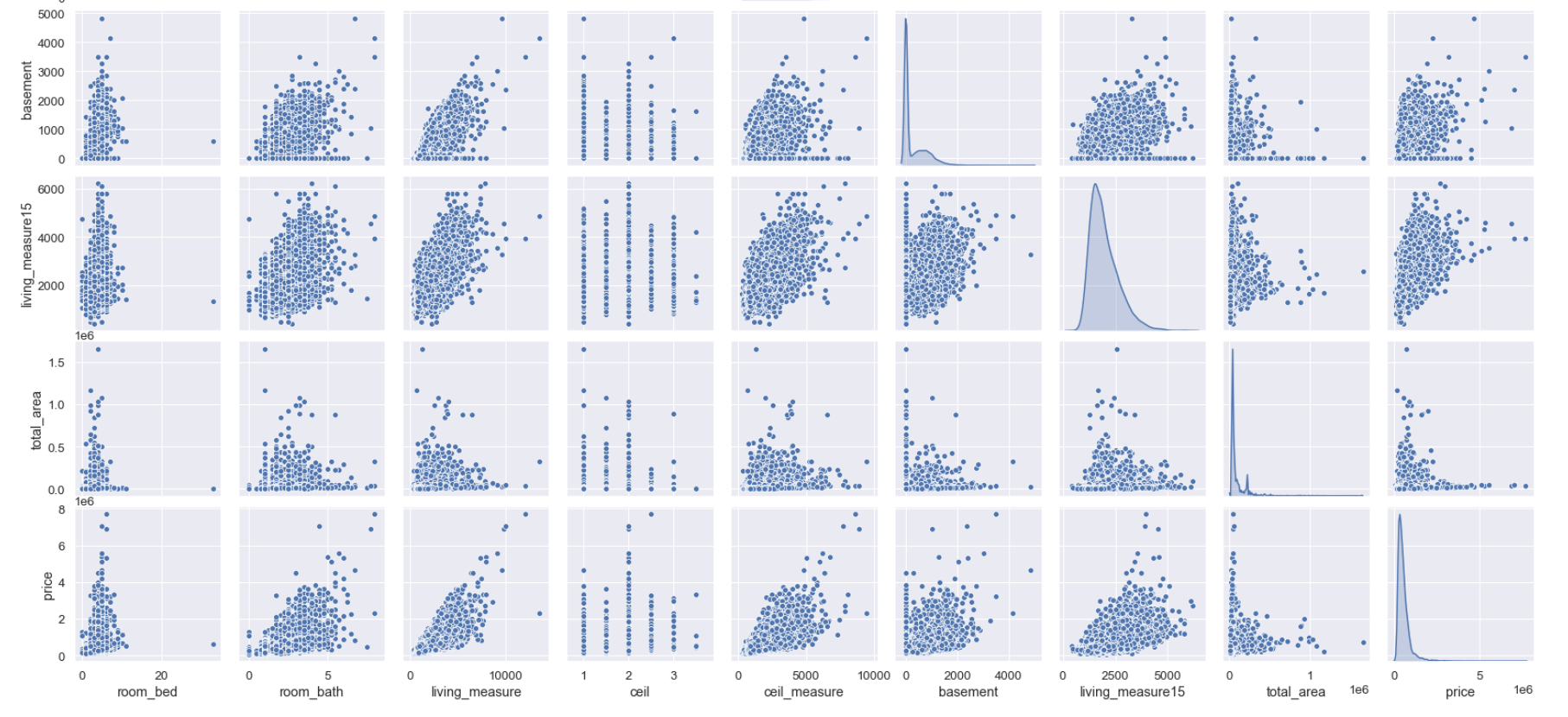
Inference:

* From above scatter plot it is evident that as the square foot of the house increases, price also increases. There are few exceptions where price is low for higher square foot of the house.
* For lot measure, as it increases prices decreases.
* Data is more scatter for basement but overall there is increase in price as house with basement area increases and that is more evident from 2000 sqft or more.

**Pair Plot:**

A close - up of a calendar

Description automatically generated with low confidence



Observations:

* For regression problems analysis we will analyze cross diagonal plots of pair plots.
* As the price of the property increases, there is significant increase in living\_measure,celi\_measure but total area deceases i.e. price is seems to be higher for less area then for larger area.
* Ceiling measure and living measures seems to be highly corelated.
* It is also evident that as number of bathrooms increase there is increase in number of bedrooms

**Heatmap:**

Chart

Description automatically generated

Observation:

* From above heatmap it evident that living\_measure and celi\_measure is highly corelated.
* There are also high amount of correlation between price and living\_measure,celi\_measure.
* Total area does not have any good correlation with other features.
* Room\_bath has good correlation with living\_measure.
* Target variable price is good corelated with living\_measure,celi\_measure and room\_bath but other features like basement,room\_bed and celi are not good corelated.

**business insights from EDA:**

* As the number of bed rooms increase, prices also increase upto certain point and then there is drop.
* Room\_bath increases with price and then it drops for few numbers.
* House with a basement has higher price compared to house without basement.
* There is increasing trend of price from year 2014 to 2015. As the year increases price of house also boosts up.
* Also price of house increase with squarefoot increase.
* Renovated house has more price then non renovated house.
* Price of house is less between Jan -Feb and Nov-Dev so we can recommend customers to buy house during these periods.

**3. Data Cleaning and Pre-processing**

**Check for missing values:**

**Table

Description automatically generated**

There are missing values for most of features. We need to treat missing values based on categorical and continuous variables separately.

**Imputing missing values for categorical variables:**

Filling with mode for below missing categorical variables:

Coast

Sight

Condition

Quality

Furnished

yr\_built

**Imputing Missing Values for Numeric Variables:**

Checking if numerical data has outliers.

**Graphical user interface, diagram

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

Inference:

* From above boxplot it is evident that except celi there are prominent outliers in all the continuous variables.
* If there are outliers in the data, filling null with mean will create skewness in the data hence will be imputing null with median of the data.
* For celi there are no outliers, so we can impute mean for nulls.

**Outlier treatment**

**Checking outliers:**

Boxplot Shape before Outliers Treatment (21613, 25)

Chart, box and whisker chart

Description automatically generated

A picture containing chart

Description automatically generated

**Inference:**

1. In house price prediction, we need to predict the price which is continuous variable hence we need to apply Linear Regression Model.
2. There are positive outliers in living\_measure,celi\_measure and total\_area variable so data will be positively skewed.
3. There are positive and negative outliers in room\_bed and room\_bath.

Boxplot Shape after Outliers Treatment (21613, 25)

**Chart, bar chart

Description automatically generated**

**Chart, bar chart

Description automatically generated**

**Variable transformation:**

* Converting categorical features to Object type – Below categorical variables are converted into object types.

**'coast','sight','condition','quality','furnished','ceil'**

* Separating categories from numerical variables:

Categorical variables: 'ceil', 'coast', 'sight', 'condition', 'quality', 'furnished'

Numerical variables: 'cid', 'dayhours', 'price', 'room\_bed', 'room\_bath', 'living\_measure', 'lot\_measure', 'ceil\_measure', 'basement', 'yr\_built', 'yr\_renovated', 'zipcode', 'lat', 'long', 'living\_measure15', 'lot\_measure15', 'total\_area', 'month', 'year'

**Addition of new variables:**

* We had to transform **dayhours** into year and month for further feature analysis. Hence two new variables have been added to dataframe.

**Removal of unwanted variables :**

Following are the features that we should drop from the data frame:

**cid** - IDs are not needed for training. It is just a surrogate key to identify customer.

**Dayhours-**  The date in this particular dataset are only limited to 2014 and 2015 and will not likely to contribute to price.

**living\_measure15,lot\_measure15 –** These features might not affect price as it is only renovation for 2015.

**Zipcode/long/lat/yr\_build/yr\_renovated** – these features are not required for training model.

**Sample records after dropping unwanted features:**

Table

Description automatically generated

1. **Model building**

# Feature Scaling:

# It standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.

# Scaling gives us unitless quantity and centers all the data on x-bar and y-bar in linear regression so as a result intercept will become zero.

# By scaling overall accuracy score will not be impacted. But it centers the independent variables.

* Feature scaling will help us see all the variables from the same lens (same scale), it will also help our models learn faster.
* The data in our data set are spread across a wide range of values, which might result in various features affecting the final result more than the other features and hence we will be using zscalar technique to normalize the dataset.

**Split the data into Train and Test:**

Splitting the train and test data based on 80:20 ratio. Splitting the data using train\_test\_split sklearn python literary where 20% of data we are keeping for Test and rest 80% of data for Train the Model.

Text

Description automatically generated

**Apply Multiple Linear Regression Model on Unscaled Data :**

* Multiple Linear Regression is an extension of Simple Linear Regression and assume that there is a linear relationship between a dependent variable Y and independent variables X.

**Coefficient of Independent Attributes:**

# Text Description automatically generated

Linear Regression Formula: y = mx + c

m: coefficient

c: intercept

**Interpretation of Coefficient:**

* Coefficient represents that, for every one unit increase or decrease in x how much y will increase or decrease while other variables are constant.
* one unit increase in total\_area,, there is a increase about 49840 units in target (charges).
* **Living measure and ceil\_measure** have negative coefficient, even though there are positive correlation between these variables and target variable. This is due to the effect of **multicollinearity** in variables. These variables are interdependent on each other.

**Interpretation of Intercept:**

* A linear regression creates a model that assumes a linear relationship between the inputs and outputs. The higher the inputs are, the higher (or lower, if the relationship was negative) the outputs are. What adjusts how strong the relationship is and what the direction of this relationship is between the inputs and outputs are our coefficients. The first coefficient without an input is called the intercept, and it adjusts what the model predicts when all your inputs are 0.
* **Intercept** represents that when values of weights and other independent variables are zero, the value of price is 

A computer screen capture

Description automatically generated with low confidence

**Interpretation of regression result:**

* From above regression results, it is observed that R squared and Adjusted R Squared are same hence there are no effect of multicollinearity in the model and we can say that model is reliable.
* Coefficient of living\_measure,celi\_measure is negative, but it is observed from pair plot that there is positive correlation with Target variable. This is due to the presence of multicollinearity among these variables.
* Below is the scatter plot between Actual Price and predicted Price. From plot it is clear that there are some noise(variance) in the prediction specially when price is high but overall actual and predicted values are very much close to each other.

Chart, scatter chart

Description automatically generated

* **Visualizing residuals:** by visualizing the residual we can see that is normally distributed (proof of having linear relationship with the dependent variable).

Chart, histogram

Description automatically generated

* Distribution of plot of predicted values between actual data and predicted data: It is clear that actual and predicted price are not in same scale.

Chart, line chart

Description automatically generated

**Interpretation of Hypothesis Testing:**

* Null Hypothesis (H0) claims that, there is no relationship between target variable (Price) and any other independent variables that means coefficients should be 0 for the sample taken from universe.
* F-statistics is the P value of the Model and its value 0 is less then threshold alpha of 0.05 which means, we will be rejecting the null hypothesis i.e. independent variables used in the model describes the target variables and it is not as good as any other models.
* P value of variables are the probability of finding coefficients for different independent variables if the sample is drawn from null hypothesis universe and assuming null hypothesis to be true.
* Probability of getting all coefficients except lot\_measure variables are very less i.e. probability is 0 which is less than 0.05, hence we will reject null hypothesis for these variables i.e. these variables can be considered as good predictor for predictive modeling.
* So overall P value is less then alpha, so rejecting null hypothesis (H0) and accepting alternate hypothesis (Ha) and model will be reliable after eliminating the useless attribute which is lot\_measure.

**Performance Matrix:**

**Mean Absolute Error (MAE**): and Root mean squared error (RMSE) are two of the most common metrics used to measure accuracy for continuous variables. -Mean Absolute Error (MAE): MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It’s the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. -a small MAE suggests the model is great at prediction, while a large MAE suggests that your model may have trouble in certain areas. A MAE of 0 means that your model is a perfect predictor of the outputs (but this will almost never happen).

**Mean Square Error (MSE**): Because we are squaring the difference, the MSE will almost always be bigger than the MAE. For this reason, we cannot directly compare the MAE to the MSE. We can only compare our model’s error metrics to those of a competing model. The effect of the square term in the MSE equation is most apparent with the presence of outliers in our data. While each residual in MAE contributes proportionally to the total error, the error grows quadratically in MSE. This ultimately means that outliers in our data will contribute to much higher total error in the MSE than they would the MAE. Similarly, our model will be penalized more for making predictions that differ greatly from the corresponding actual value. This is to say that large differences between actual and predicted are punished more in MSE than in MAE.

**Root Mean Square Error (RMSE)**: It is a quadratic scoring rule that also measures the average magnitude of the error. It’s the square root of the average of squared differences between prediction and actual observation. We will often use RMSE to convert the error metric back into similar units, making interpretation easier. Since the MSE and RMSE both square the residual, they are similarly affected by outliers. The RMSE is analogous to the standard deviation (MSE to variance) and is a measure of how large your residuals are spread out. Both MAE and MSE can range from 0 to positive infinity, so as both of these measures get higher, it becomes harder to interpret how well your model is performing.

**Interpretation :**

**R square of train and test sets:**

* **Accuracy score** for train and test datasets are **0.57** and **0.56** respectively so it is clear that model is not overfitting or underfitting.
* There are 57 percent of the variance in the price is explained by predictors in the model for train set and 56 percent for test set.

**RMSE on Test and Train set:**

* RMSE of train and test datasets are 240905 and 233191 respectively and it indicates that there are no issue of overfitting and underfitting. RMSE values here are in unit of target variables.

**Multiple Linear Regression (Apply Scaling):**

* Try to improve the model performance by applying z score since independent attributes have different unit and scales of measurement.
* Intercept of the model is very close to zero (3.363455917187121e-17).

**R square of train and test sets:**

* Regression model **accuracy score** for train and test datasets are 0.57 and 0.569 respectively which is same as before hence model accuracy did not improve.

**RMSE on Test and Train set:**

* RMSE of train and test datasets are 0.65 and 0.647 respectively and it indicates that there are no issue of overfitting and underfitting. RMSE values here less because input dataset is scaled.

**MSE and MAE of test and train set:**

* Here, we have obtained two error metric Mean Absolute Error (MAE) and Mean Squared Error (MSE) to determine the accuracy of our model.

Text

Description automatically generated

* Below is the scatter plot between Actual Price and predicted Price. From plot it is clear that there are some noise(variance) in the prediction. Overall actual and predicted values are close but scatter plot on original dataset (without scaling) is better.

Chart, scatter chart

Description automatically generated

* Distribution of plot of predicted values between actual data and predicted data: It is clear that actual and predicted price on the same scale.

Chart, line chart

Description automatically generated

**Building Support Vector Regression (SVR)**

* The model we will be implementing is the SVR model (using the RBF kernel).
* Fit scaled data on SVR model.

base\_svr\_model = SVR()

**Performing K-fold cross validation:**

Usually, we split the data set into training and testing sets and use the training set to train the model and testing set to test the model. We then evaluate the model performance based on an error metric to determine the accuracy of the model. This method, however, is not very reliable as the accuracy obtained for one test set can be very different to the accuracy obtained for a different test set. K-fold Cross Validation (CV) provides a solution to this problem by dividing the data into folds and ensuring that each fold is used as a testing set at some point.

**Evaluating a ML model using 10-Fold CV:**

scoring = {'abs\_error': 'neg\_mean\_absolute\_error',

'squared\_error': 'neg\_mean\_squared\_error'}

scores = cross\_validate(base\_svr\_model, x\_train\_scaled, y\_train\_scaled, cv=10, scoring=scoring, return\_train\_score=True)

**MSE & MAE on Test and Train set:** Here, we have obtained two error metric Mean Absolute Error(MAE) and Mean Squared Error(MSE) to determine the accuracy of our model. This is done by defining a custom scorer. This scorer is passed to the cross\_validate() function of sklearn, which performs 10-fold cross validation

Text

Description automatically generated

**Model Tuning Support Vector Regression (SVR):**

* First, we pass the features(X) and the dependent(y) variable values of the data set, to the method created for the Support Vector regression model.
* We then use the [grid search cross validation method](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)  from the *sklearn* library to determine the optimal values to be used for the hyperparameters of our model from a specified range of values.

Here, we have chosen the three hyperparameters.

Identified best PARAM with below Hyperparameter values.

param\_grid={

'C': [0.1, 1, 100, 1000],

'epsilon': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10],

'gamma': [0.0001, 0.001, 0.005, 0.1, 1, 3, 5]

}

We have used the three hyper-parameters.

C, epsilon and gamma, to be optimized.

**C:** Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.

**Epsilon:** Epsilon in the epsilon-SVR model. It specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value.

**Gamma:** Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’.

**Best Param:** {'C': 100, 'epsilon': 0.1, 'gamma': 0.01}

**MSE & MAE on Test and Train set:** Here, we have obtained two error metric Mean Absolute Error(MAE) and Mean Squared Error(MSE) to determine the accuracy of our model. This is done by defining a custom scorer. This scorer is passed to the cross\_validate() function of sklearn, which performs 10-fold cross validation

Text

Description automatically generated

From above scores for MSE and MAE it is evident that model performance is not improved by tuning.

**Ridge Regression:**

* **Ridge** regression performs ‘**L2 regularization** ‘, i.e., it adds a factor of sum of squares of coefficients in the optimization objective. Thus, ridge regression optimizes the following:

Objective = RSS + α \* (sum of square of coefficients)

Here, α (alpha) is the parameter which balances the amount of emphasis given to minimizing RSS vs minimizing sum of square of coefficients. α can take various values:

**α = 0:**

The objective becomes same as simple linear regression.

We’ll get the same coefficients as simple linear regression.

**α = ∞:**

The coefficients will be zero. Why? Because of infinite weightage on square of coefficients, anything less than zero will make the objective infinite.

**0 < α < ∞:**

The magnitude of α will decide the weightage given to different parts of objective.

The coefficients will be somewhere between 0 and ones for simple linear regression.

* **As the value of alpha increases, the model complexity reduces**. Though higher values of alpha reduce overfitting, significantly high values can cause underfitting as well (eg. alpha = 5). Thus, alpha should be chosen wisely. A widely accept technique is cross-validation, i.e., the value of alpha is iterated over a range of values and the one giving higher cross-validation score is chosen.

**MSE & MAE on Test and Train set:** Here, we have obtained two error metric Mean Absolute Error(MAE) and Mean Squared Error(MSE) to determine the accuracy of our model. This is done by defining a custom scorer. This scorer is passed to the cross\_validate() function of sklearn, which performs 10-fold cross validation

Text

Description automatically generated

**Lasso Regression:**

* Lasso regression performs **L1 regularization**, i.e. it adds a factor of sum of absolute value of coefficients in the optimization objective. Thus, lasso regression optimizes the following:

#### Objective = RSS + α \* (sum of absolute value of coefficients)

Here, α (alpha) works similar to that of ridge and provides a trade-off between balancing RSS and magnitude of coefficients. Like that of ridge, α can take various values. Lets iterate it here briefly:

* α = 0: Same coefficients as simple linear regression
* α = ∞: All coefficients zero (same logic as before)
* 0 < α < ∞: coefficients between 0 and that of simple linear regression

**MSE & MAE on Test and Train set:** Here, we have obtained two error metric Mean Absolute Error(MAE) and Mean Squared Error(MSE) to determine the accuracy of our model. This is done by defining a custom scorer. This scorer is passed to the cross\_validate() function of sklearn, which performs 10-fold cross validation

**Text

Description automatically generated**

**Ensemble modelling:**

**Building Random Forest Regression:**

* A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging. Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement.
* The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

**Implementing random forest model with default parameters:**

* **Import sklearn library, it** is a machine learning library which features various classification, regression and clustering algorithms
* First, we pass the features(X) and the dependent(y) variable values of the data set, to the method created for the random forest regression model.
* Implementing random forest with default parameters.
* After creating a random forest regressor object, we pass it to [the cross\_val\_score(](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html)) function which performs K-Fold cross validation the given data and provides as an output, an error metric value, which can be used to determine the model performance.

**MSE & MAE on Test and Train set:** Here, we have obtained two error metric Mean Absolute Error (MAE) and Mean Squared Error(MSE) to determine the accuracy of our model. This is done by defining a custom scorer. This scorer is passed to the cross\_validate() function of sklearn, which performs 10-fold cross validation

Text

Description automatically generated

Here test and train MSE and MAE are not matching hence model seems to be a underfitting.

# Implementing random forest model with Hyperparameter Tuning:

* First, we pass the features(X) and the dependent(y) variable values of the data set, to the method created for the random forest regression model.
* We then use the [grid search cross validation method](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) from the sklearn library to determine the optimal values to be used for the hyperparameters of our model from a specified range of values.
* Here, we have chosen the five hyperparameters.

max\_depth, n\_estimators, min\_sample\_leaf, min\_sample\_split and n\_estimators, to be optimized.

* According to [sklearn documentation,](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html) max\_depth refers to the maximum depth of the tree and n\_estimators, the number of trees in the forest. Min\_sample\_split is the minimum number of samples required to split an internal node.
* We then use the [grid search cross validation method](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) from the sklearn library to determine the optimal values to be used for the hyperparameters of our model from a specified range of values.
* Ideally, we can expect a better performance from our model when there are more trees. However, we must be cautious of the value ranges we specify and experiment using different values to see how our model performs.
* After creating a random forest regressor object, we pass it to [the cross\_val\_score(](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html)) function which performs K-Fold cross validation (refer to [this article](https://medium.com/datadriveninvestor/k-fold-cross-validation-6b8518070833) for more information on K-Fold cross validation) on the given data and provides as an output, an error metric value, which can be used to determine the model performance.

**MSE & MAE on Test and Train set:**

**Text

Description automatically generated**

Here test and train scores are same so there is no issue with underfitting or overfitting.

# Building Gradient Boost Regression:

* "Boosting" in machine learning is a way of combining multiple simple models into a single composite model. This is also why boosting is known as an additive model, since simple models (also known as weak learners) are added one at a time, while keeping existing trees in the model unchanged. As we combine more and more simple models, the complete final model becomes a stronger predictor. The term "gradient" in "gradient boosting" comes from the fact that the algorithm uses gradient descent to minimize the loss.
* [Decision trees](https://medium.com/@dhiraj8899/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a) are used as the weak learners in gradient boosting. Decision Tree solves the problem of machine learning by transforming the data into tree representation. Each internal node of the tree representation denotes an attribute, and each leaf node denotes a class label. The loss function is generally the squared error (particularly for regression problems). The loss function needs to be differentiable.
* Also like linear regression we have concepts of **residuals** in Gradient Boosting Regression as well. Gradient boosting Regression calculates the difference between the current prediction and the known correct target value.  
  This difference is called residual.
* **Import sklearn.ensemble liberary** ,It is a machine learning library which features various classification, regression and clustering algorithms
* First, we pass the features(X) and the dependent(y) variable values of the data set, to the method created for the Gradient Boost regression model.
* Implementing random forest with default parameters.
* After creating a random forest regressor object, we pass it to [the cross\_val\_score(](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html)) function which performs K-Fold cross validation the given data and provides as an output, an error metric value, which can be used to determine the model performance.

**MSE & MAE on Test and Train set:**

Text

Description automatically generated

**Observation:** Since MSE on test and train are almost same so there is no overfitting or underfitting issue with model.

# Implementing Gradient Boost model with Hyperparameter Tuning:

* We have used the [grid search cross validation method](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) from the sklearn library to determine the optimal values to be used for the hyperparameters of our model from a specified range of values.
* Here, we have chosen the five hyperparameters.

max\_depth, n\_estimators, min\_sample\_leaf, learning\_rate and max\_features, to be optimized.

* According to [sklearn documentation,](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html) max\_depth refers to the maximum depth of the tree and n\_estimators, the number of trees in the forest. Min\_sample\_leaf is the minimum number of samples required to be at a leaf node. Learning rate shrinks the contribution of each tree by learning\_rate.

**MSE & MAE on Test and Train set:**

Text

Description automatically generated

1. **Model Validation**

**Compare all the model parameters:**

* We are using The Mean Squared Error, Mean absolute error and RMSLE for model evaluation.
* Mean Squared Error (MSE) and Root Mean Square Error penalizes the large prediction errors vi-a-vis Mean Absolute Error (MAE).
* Mean Squared Error (MSE) and Root Mean Square Error penalizes the large prediction errors vi-a-vis Mean Absolute Error (MAE) hence the lower value of MAE, MSE, and RMSE implies higher accuracy of a regression model.

# Table Description automatically generated

* Above table which has all the models scores for MSE and MAE of test and train dataset. If we compare all the model’s prediction accuracy, it is evident that least MAE is almost same for Support Vector Regressor and Gradient Boost Regressor, so we have to look at model Error and penalizing the model for large magnitude error will help us here to choose the appropriate model.
* Two regression models with error values

**SVR R-squared errors :** array([-0.29931186, -0.29822226, -0.30040686, -0.29937442, -0.29744586,

-0.2921246 , -0.29316399, -0.28454881, -0.29698458, -0.30011442])

**Gradient Boost R-squared Errors :**array([-0.22853585, -0.2285429 , -0.23205995, -0.23165985, -0.23079607,

-0.22458547, -0.22738451, -0.2239404 , -0.22632339, -0.23230371])

* By squaring the difference between the actual and predicted values, we are able to consider only negative error values and penalize higher error values.
* From above it is evident that Gradient Boost model is slightly higher magnitude of error and so it is getting penalized by GBR by a bit. Hence by these observations Gradient Boost Regression seems to be better fir for our use case then SVR.

**Root Mean Squared Logarithmic Error:**

* Observations of the target variables are huge in magnitude for both actual and predicted values and error for that pair is going to be large compared for other smaller magnitude of the observations.
* For example, if a model predicts small condos worth $100,000 as $50,000 then it is off by a lot but if the same model predicts a mansion’s price as $900,000 instead of $850,000 we can consider it close. The same error value of $50k is both massive and also insignificant in the same data set.
* Logarithms are usually a convenient way to express large numbers in much smaller magnitude. When the regression models’ Y and Ŷ values vary widely, higher magnitude numbers increase error in RMSE, MSE & MAE significantly.
* In such cases, to avoid such relatively large differences between actual and predicted value contributing to error, we use **RMSLE.**

**Compare RMSLE Score:**

Table

Description automatically generated

**Conclusion:**

* Here in our use case, we want to penalize the model for underestimates more than overestimates and RMSLE score for GBR and SVR is almost same and GBR error is slightly large in magnitude then SVR, hence we conclude that **Gradient Boost Regression** will be most optimal model for our use case.

1. **Final interpretation / recommendation:**

Recommendations:

* We want to provide estimate for a house and end customer is real estate broker, we might want our estimates to be off by a bit rather than extremely accurate or inaccurate. In that case, penalizing the model for larger magnitude errors will help us choose the appropriate model. We could do that by calculating MSE.
* Independent variables like ‘ceil measure, ‘coast’, and ‘sight’ , ‘basement’, ‘quality have maximum influence on ‘price. Hence, the real estate companies can focus on the above variables to increase their profit share.
* Renovated house is more in demand so real estate agents can renovate the house to get more selling price.
* Price of house is less between Jan -Feb and Nov-Dev so we can recommend customers to buy house during these periods and real estate companies can run offers/campaign during this period to attract more customers.
* As the number of bed rooms increase, prices also increase upto certain point and then there is drop.
* There is increasing trend of price from year 2014 to 2015. As the year increases price of house also boosts up.Also price of house increase with **squarefoot** increase.