## Abstract:

The main objective of our project is to predict the price of a house based on certain features like locality, price per sq. feet, no. of rooms, age of house etc. In this project we will develop and evaluate the performance and predictive power of a model trained and tested on data collected from different houses of different areas. Once we get a good fit, we will use this model to predict the true price value of the house based on certain factors. A model like this would be very valuable for buyers, banks for giving loans, real estate agents who could use this information provided in a daily basis. Here Machine Learning algorithms :- Linear regression algorithm and Gradient descend boosting for the data set and see if it provides good accuracy or not.



Keywords:- Linear Regression algorithm , sklearn, gradient descent boosting

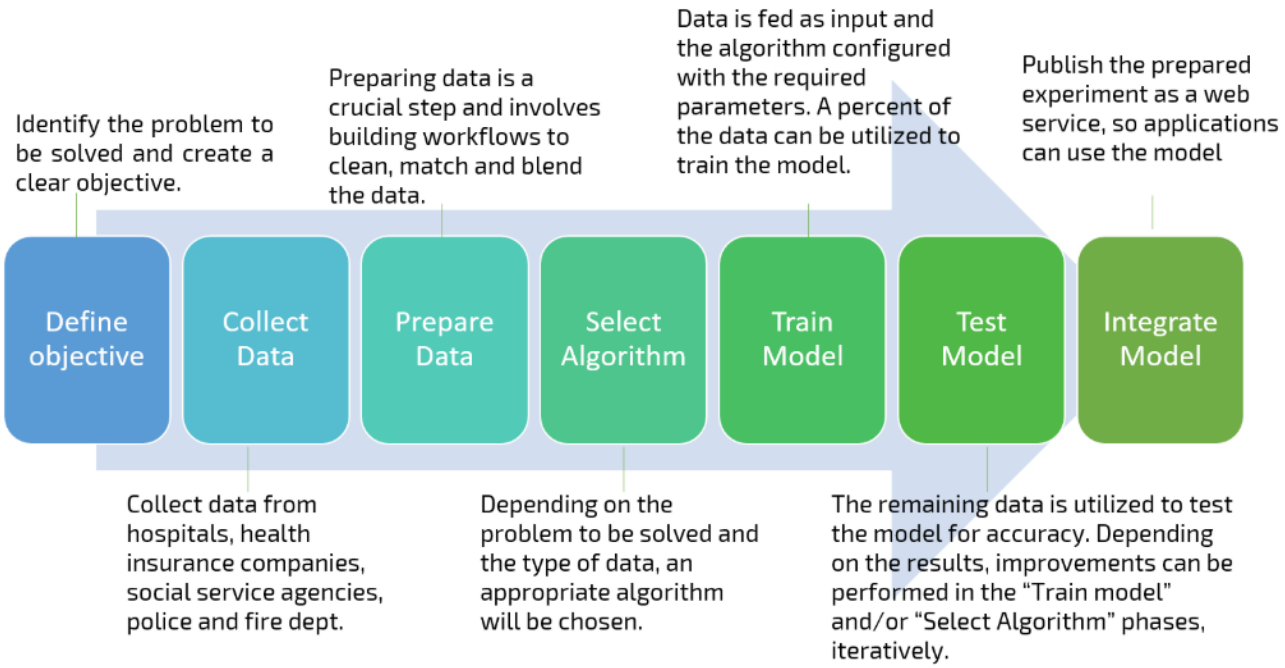

## Introduction:

An accurate prediction on the house price is important to prospective homeowners, developers, investors, appraisers, tax assessors and other real estate market participants, such as, mortgage lenders and insurers. Traditional house price prediction is based on cost and sale price comparison lacking of an accepted standard and a certification process. Therefore, the availability of a house price prediction model helps fill up an important information gap and improve the efficiency of the real estate market. Housing market is of great important for the economy activities. Housing construction and renovation boost the economy through an increase in the aggregate expenditures, employment and volume of house sales. They also simulate the demand for relevant industries such as household durables. The oscillation of house prices affects the value of asset portfolio for most households for a house is the largest single asset.

Data Science is the process of making some assumptions and hypothesis on the data, and testing them by performing some tasks. Initially we could make the following intuitive assumptions for each feature:

- The houses with more bedrooms and bathrooms are priced higher.
- A relatively new house is more expensive than an old house and a house with a garden is priced higher than one without a garden.
- Recent studies further justify the necessity of housing price analysis with a conclusion that housing sector plays a significant role in acting as a leading indicator of the real sector of the economy and assets prices help forecast both inflation and output .
- Square feet which increases the price of the house and even location influencing the prices of the house.
- Connectivity with the public transportation and other basic necessity will effect the housing prices.

Basic work flow for solving the given problem




## Preparing Data:

Here we are going to work on a dataset which consists information about the location of the house , price and other aspects such as square feet etc. When we work on these sort of data , we need to see which column is important for us and which is not. Our main is to make a model which can give us a good prediction on the price of the house based on other variables. We are going to use Linear Regression for this dataset and see if it gives us a good accuracy or not.

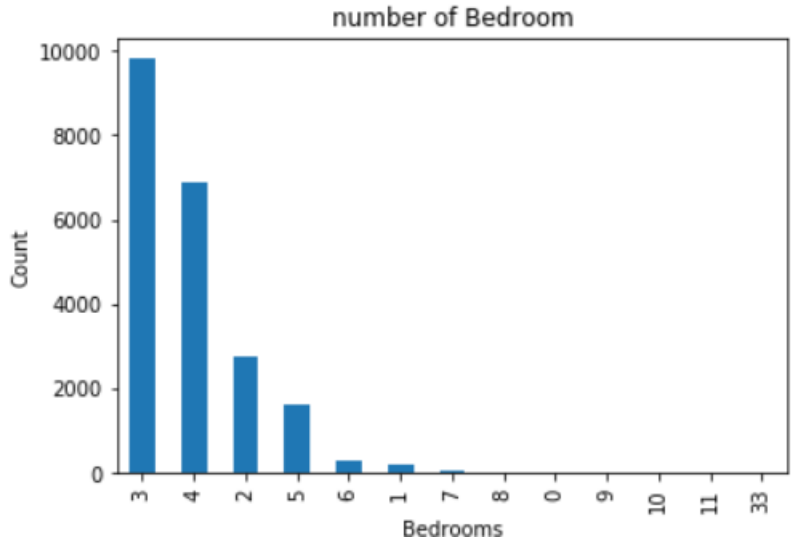
We are going to use seeborn to see some visualization and also what we can infer from visualization for given data.

By looking at this problem from a builder's perspective, sometimes it's important for a builder to see which is the highest selling house type which enables the builder to make house based on that.

Here in India , for a good locality a builder opts to make houses which are more than 3 bedrooms which attracts the higher middle class and upper class section of the society.

```
data['bedrooms'].value_counts().plot(kind='bar')
plt.title('number of Bedroom')
plt.xlabel('Bedrooms')
plt.ylabel('Count')
sns.despine
```

```
<function seaborn.utils.despine>
```




We use seaborn , and we get his beautiful visualization. Joinplot function helps us see the concentration of data and placement of data and can be really useful.

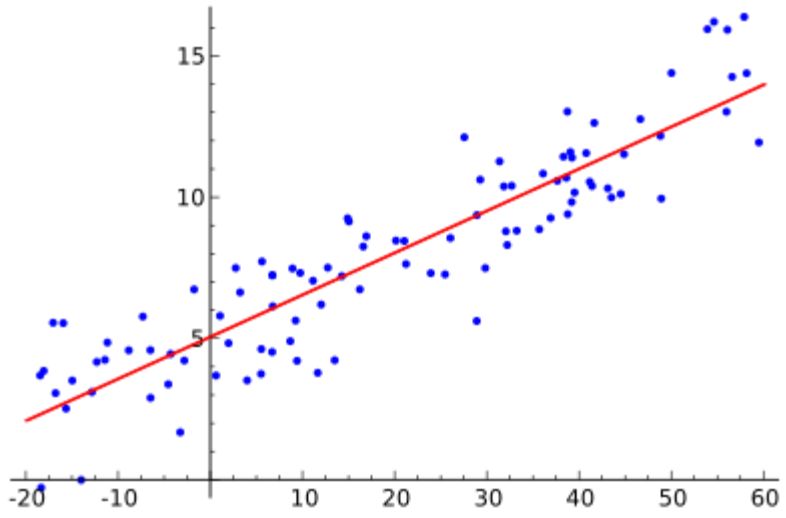
We can see relation between other factors and prices of house similarly now we the basic model is created . Here we used linear regresson model in statistics which helps us predicts the future based upon past relationship of variables.

## Machine Learning Algorithm to create a model

### Linear Regression

Linear regression is a linear approach to modeling the relationship between dependent variable and one or more independent variables. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression

Linear regression plays an important role in the field of artificial intelligence such as machine learning. The linear regression algorithm is one of the fundamental supervised machine-learning algorithms due to its relative simplicity and well-known properties.




# Regression works on the line equation , y=mx+c , trend line is set through the data points to predict the outcome.

Given a data set of *n* statistical units, a linear regression model assumes that the relationship between the dependent variable *y* and the *p*-vector of regressors **x** is linear. This relationship is modeled through a *disturbance term* or *error variable* ε — an unobserved random variable that adds "noise" to the linear relationship between the dependent variable and regressors. Thus the model takes the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where $^{\mathsf{T}}$ denotes the transpose, so that $\mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}$ is the inner product between vectors $\mathbf{x}_i$ and $\boldsymbol{\beta}$.

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

Often these *n* equations are stacked together and written in matrix notation

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

$$X = \begin{pmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_n^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

$\beta$ is the dimensional *parameter vector* Its elements are known as *effects* or *regression coefficients* (although the latter term is sometimes reserved for the *estimated* effects)

## Gradient Boosting

Gradient boosting is a greedy algorithm and can overfit a training dataset quickly.

It can benefit from regularization methods that penalize various parts of the algorithm and generally improve the performance of the algorithm by reducing overfitting.

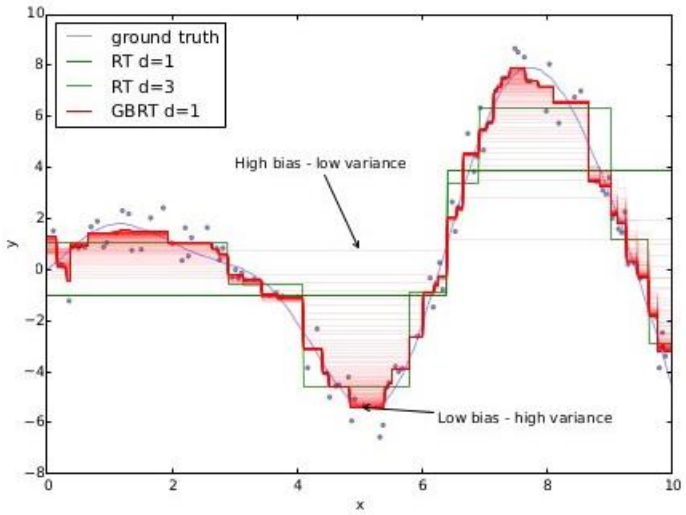
Gradient boosting involves three elements:

A loss function to be optimized.
A weak learner to make predictions.
An additive model to add weak learners to minimize the loss function.

Gradient Boosting Regressor is used as estimator and we define these parameters

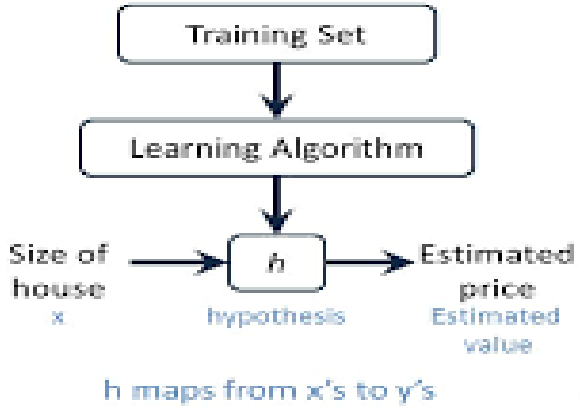
- n_estimator — The number of boosting stages to perform. We should not set it too high which would overfit our model.
- max_depth — The depth of the tree node.
- learning_rate — Rate of learning the data.
- loss — loss function to be optimized. 'ls' refers to least squares regression
- minimum sample split — Number of sample to be split for learning the data

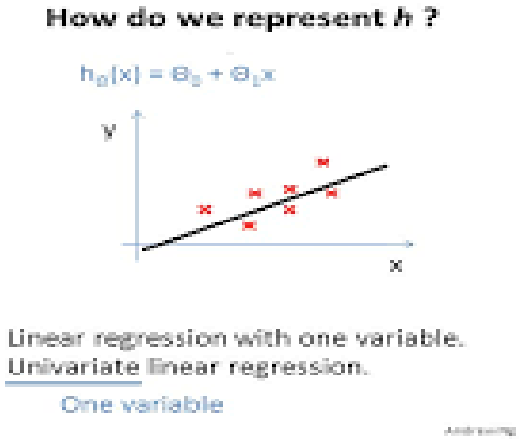
```
from sklearn.ensemble import GradientBoostingRegressor
est = GradientBoostingRegressor(n_estimators=2000, max_depth=1).fit(X, y)
for pred in est.staged_predict(X):
    plt.plot(X[:, 0], pred, color='r', alpha=0.1)
```




We then fit our training data into the gradient boosting model and check for accuracy

Final output of the Model on given data :-



References:-

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemhttp

[2] https://ieeexplore.ieee.org/document/8473231

[3] http://www.iraj.in/journal/journal_file/journal_pdf/12-477-153396274234-40.pdf

[4] https://pdfs.semanticscholar.org/782d/3fdf15f5ff99d5fb6acafb61ed8e1c60fab8.pdf

[5] https://towardsdatascience.com/machine-learning-project-predicting-boston-house-prices-with-regression-b4e47493633d

[6] https://medium.com/@ben_lau93/house-prices-prediction-using-andrew-ngs-machine-learning-algorithm-31b29a81acb8

[7] http://www.ijstr.org/final-print/dec2014/Comparative-Study-On-Estimate-House-Price-Using-Statistical-And-Neural-Network-Model-.pdf

[8] *Yan, Xin (2009), Linear Regression Analysis: Theory and Computing, World Scientific, pp. 1–2, ISBN 9789812834119,*

*[9] "Linear Regression (Machine Learning)" University of Pittsburgh.*