

**M I N I P R O J E C T**

# HOUSE SALE PRICE

PREDICTION MODEL

**USING PYTHON**

By- Sahil Gupta Course- B.Tech Section- AI&DS Class Rollno.- 60

Student ID- 20021853

**Problem Statement**

Machine learning model to predict “**HOUSE *SALE PRICE***” using Python to determine a

better price with the help of different variables like views, parking, bedrooms and much

more.

**Motivation**

Today, we all face the problem to find a good home with many features like good location,

better locality, a big house with great view, with great leisure, comfort and parking. And to

find so we have to deal with the middle-man or property dealer and give a part to them.

With no knowledge of surroundings and price according to it people have to give a huge

amount of money in exchange of comfort. For this we really need a machine which can give

best price according to the features.

So, I created a model using python which take inputs like Number of bathrooms,

Number of bedrooms, waterfront, view, parking, location, maintenance, year of built,

etc. and predict a good price to us.

I would like to thank my mentor “***Mr. VIKAS TOMAR***” who gave me this opportunity

To work on this and gain some extra knowledge on python and machine learning. I am

Glad that I work under sir.

**About the dataset**

The dataset I used is taken from [www.Kaggle.com](http://www.Kaggle.com) . The data consist of date the house

build in the locations, Number of bedrooms (1-5), Number of bathrooms

(1-3), Square foot living, square foot lot, Number of floors (1-2), Waterfront, View,

condition, grade, Square above, square basement, year built, year renovated,

Zip-code (Seattle-King County), latitude, longitude, square foot living15

(Average **house square** **footage of the 15 closest houses**), Square

foot lot15(sized **for a single house**).

**Modules and Libraries used**

* **NumPy**

NumPy is a python library used for working with arrays, linear algebra and matrices.

These includes how to create NumPy arrays, use broadcasting, access values, and

manipulate arrays.

* **Pandas**

Pandas is widely used for data analysis. pandas is **a Python package providing fast,**

**flexible, and expressive data structures** designed to make working with “relational” or

“labeled” data both easy and intuitive.

* **Matplotlib.pyplot**

Matplotlib is a python library used for data visualization. it offers a viable open

source alternative to MATLAB.

* **Seaborn**

Seaborn is a python library used for making statistical graphics.  It builds on top of

matplotlib and integrates closely with pandas’ data structures. Seaborn helps you explore

and understand your data.

* **Mpl toolkits**

Mpl\_toolkits provide some basic 3D plotting tools.  Not the fastest or most feature

complete 3D library out there, but it ships with Matplotlib and thus may be a lighter weight

solution for some use cases.

* **Scikit-learn**

Sklearn is a python library used for machine learning. Scikit-learn is **a free**

**machine learning library for Python**. It features various algorithms like support vector

machine, random forests, and k-neighbors, and it also supports Python numerical

and scientific libraries like NumPy and SciPy.

* **Linear Regressor**

Linear regression is a machine learning algorithm.  linear regression is **a**

**regression model that estimates the relationship between one independent**

**variable and one dependent variable using a straight line**. Both variables should

be quantitative.

* **Train\_test\_split**

It is a function in Sklearn model selection for splitting data into train and

test subsets. train\_test\_split is **a function in Sklearn model selection for splitting**

**data arrays into two subsets**: for training data and for testing data. With this function,

you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make

random partitions for the two subsets.

* **Ensemble**

Ensemble method is a machine learning technique that combines several

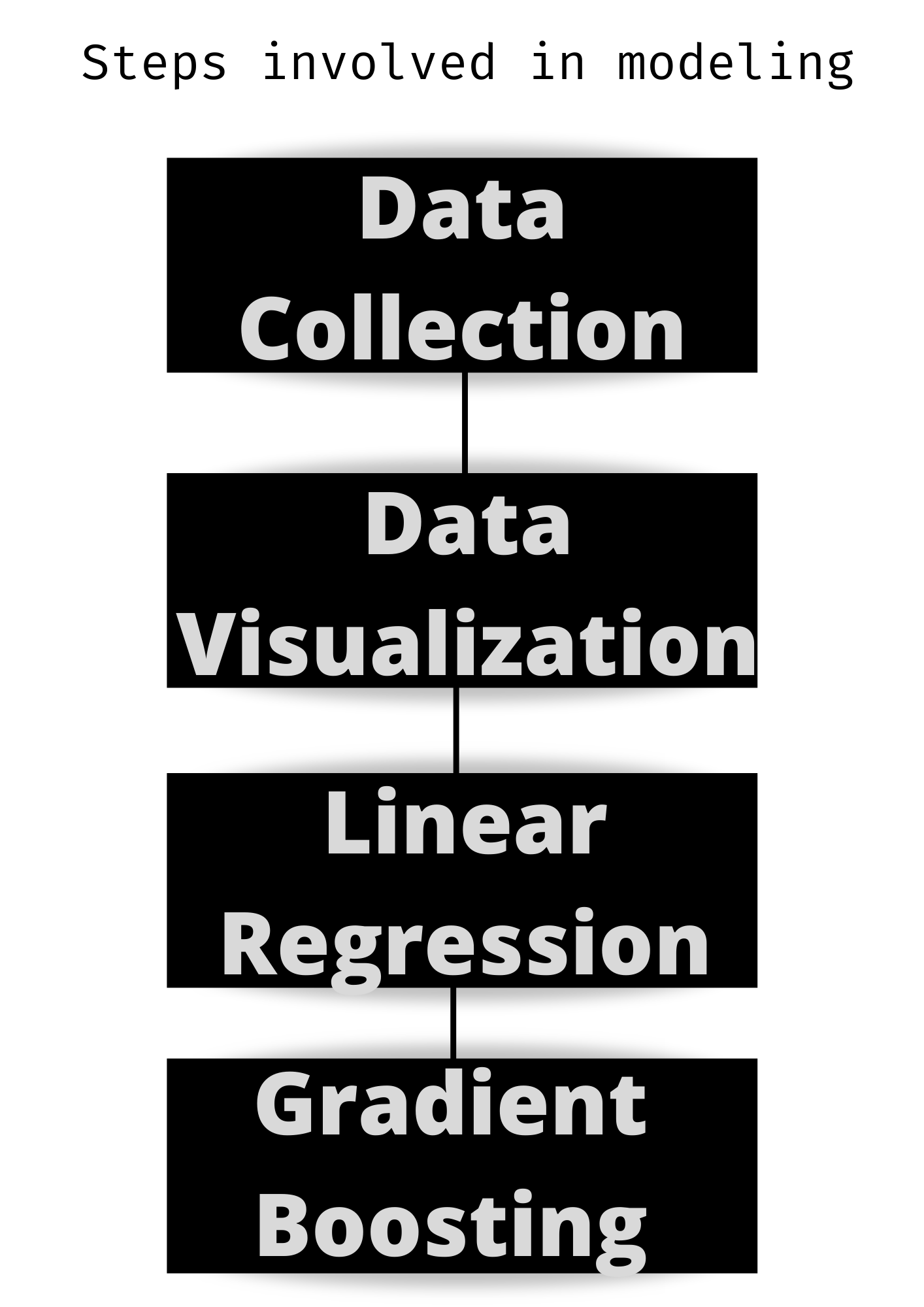
Base models in order to produce one optimal predictive. Ensemble modeling is **a**

**process where multiple diverse models are created to predict an outcome**, either by

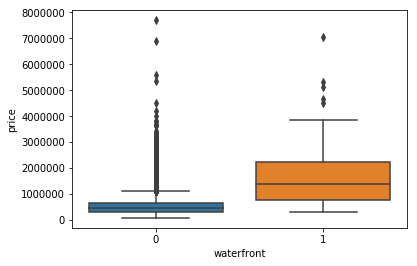
using many different modeling algorithms or using different training data sets.

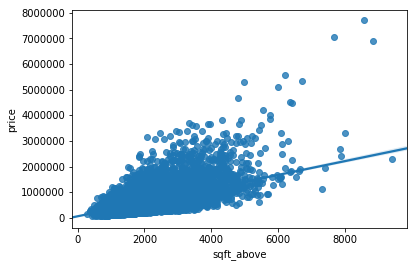
**Tools Used**

* + Google collab
  + Visual studio code
  + Canva
  + Python
  + Html
  + CSS
  + GitHub



**VISUALIZATION**

****

****

**Methodology**

In this project I have first load the data into google collab and then check their

Data-types. I have also checked their ranges and their factors.

Then I go for the correlations of a variables like price with other variables and

visualize them like PRICE VS NO. OF BEDROOMS, PRICE VS NO. OF BATHROOMS, PRICE VS PARKING etc. After the visualizing I move towards modelling.

The machine learning model I use is linear regression because as I see there is a linear

Relation between price and other factors. So, this was a little easy work for me as

the mathematics behind the model was “Y**=M\*X + C**”.

Linear regression is a linear model, **that assumes a linear relationship between**

**the input variables (x) and the single output variable (y)**. Different techniques

can be used to prepare or train the linear regression equation from data, the most common of which is called Ordinary Least Squares.

After training the model I have checked the accuracy which was about **74%** and

it was less than 85% so to make the model more efficient I use another python

library “**Ensemble”** which is basically a Gradient booster which is use to increase

the efficiency of the model and after using it the score of model Increases to **89%**

which I think is a good score and make my model more effective and accurate.

**Discussion of Result**

The end result of the model is to show the best and nearest actual price based on the

Previous data of price and houses which I think will be very helpful for anyone who is facing

the problem of house prices.

**Link for project files:- https://github.com/Sahilgupta4103/MINIPROJECT.git**