# HOUSE PRICE PREDICTION

A project report submitted in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Technology**

In

**Computer Science and Engineering**

**By**

A.Sai Sowmya(16011A0525)

K.Shravani(16011A0532)

M.Shreya(16011A0533)

Under the guidance of

**Prof. Dr.D.Vasumathi**

Department of Computer Science and Engineering

Jawaharlal Nehru Technological University Hyderabad

JNTUH College of Engineering Hyderabad

Kukatpally, Hyderabad – 500085

2019

**Department of Computer Science and Engineering**
**Jawaharlal Nehru Technological University Hyderabad**
**JNTUH College of Engineering, Hyderabad – 500085**

## DECLARATION BY THE CANDIDATES

We, A.SaiSowmya(16011A0525), K.Shravani(16011A0532), M.Shreya(16011A0533) here by certify that the mini-project report entitled "**House Price Prediction**", carried out under the guidance of **Dr.D.Vasumathi**, is submitted in partial fulfilment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science and Engineering***. This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced/ copied from any source and have not been submitted to any other University or Institute for the award of any other degree or diploma.

**A.Sai Sowmya(16011A0525)**
**K.Shravani(16011A0532)**
**M.Shreya(16011A0533)**
Department of Computer Science & Engineering,
JNTUH College of Engineering,
Hyderabad.

**Department of Computer Science and Engineering**
**Jawaharlal Nehru Technological University Hyderabad**
**JNTUH College of Engineering, Hyderabad – 500085**


# CERTIFICATE BY THE SUPERVISOR

This is to certify that the project report entitled "**House Price Prediction**", being submitted by **A.Sai Sowmya**(16011A0525), **K.Shravani**(16011A0532), **M.Shreya** (16011A0533) in partial fulfilment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science and Engineering***, is a record of bonafide work carried out by them. The results are verified and found satisfactory.

**Dr.D.Vasumathi,**
Professor,
Department of Computer Science and Engineering,
JNTUH College of Engineering,
Hyderabad.

Date:

**Department of Computer Science and Engineering**

**Jawaharlal Nehru Technological University Hyderabad**

**JNTUH College of Engineering, Hyderabad – 500085**


# CERTIFICATE BY THE HEAD OF DEPARTMENT

This is to certify that the project report entitled "**House Price Prediction**", being submitted by **A.Sai Sowmya**(16011A0525), **K.Shravani**(16011A0532), **M.Shreya** (16011A0533) in partial fulfilment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science and Engineering***, is a record of bonafide work carried out by them. The results are verified and found satisfactory.

**Dr.R. Sri Devi,**
Professor & Head of the Department,
Department of Computer Science and Engineering,
JNTUH College of Engineering,
Hyderabad.

Date:

## ACKNOWLEDGEMENT

We would like to express sincere thanks to our Supervisor **Dr.D.Vasumathi** mam, Professor for her admirable guidance and inspiration both theoretically and practically and most importantly for the drive to complete project successfully. Working under such an eminent guide was our privilege.

We owe a debt of gratitude to **Dr.R.SriDevi** mam, Professor & Head of the department of Computer Science & Engineering, for her kind considerations and encouragement in carrying out this project successfully.

We are grateful to the Project Review Committee members and Department of Computer Science & Engineering who have helped in successfully completing this project by giving their valuable suggestions and support.

We express thanks to our parents for their love, care and moral support without which we would have not been able to complete this project. It has been a constant source of inspiration for all our academic endeavours. Last but not the least, we thank the Almighty for making us a part of the world.

**A.Sai Sowmya(16011A0525)**
**K.Shravani(16011A0532)**
**M.Shreya(16011A0533)**

# ABSTRACT

The relationship between house prices and the economy is an important motivating factor for predicting house prices. For a house price prediction, more accurate method based on location, house type, size, build year, local amenities, and some other factors which could affect house demand and supply. It is important to predict housing prices without bias to help both the buyers and sellers make their decisions.

For building a prediction model, **gradient boosting regression is used.** It is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically trees.

The Python library sklearn is used to define gradient boost regressor variable, then we set parameters and check for accuracy by fitting the training data into the model.

# TABLE OF CONTENTS

# INTRODUCTION

The relationship between house prices and the economy is an important motivating factor for predicting house prices. There is no accurate measure of house prices .A property's value is important in real estate transactions. House prices trends are not only the concerns for buyers and sellers, but they also indicate the current economic situations. Therefore, it is important to predict the house prices without bias to help both buyers and sellers make their decisions.
A good housing price prediction would better prepare them for what to expect before they make one of the most important financial decisions in their lives.

House sellers and buyers are increasingly turning to online research in order to estimate house price before contacting real estate agents . Researching how much the house you are interested in is worth on your own can be difficult for multiple reasons. One particular reason is that there are many factors that influence the potential price of a house, making it more complicated for an individual to decide how much a house is worth on their own without external help. This can lead to people making poorly informed decisions about whether to buy or sell their houses and which prices are reasonable. Because houses are long term investments, it is imperative that people make their decisions with the most accurate information possible.

# OBJECTIVE

This project aims to create a house price prediction model using regression to obtain optimal prediction results. Regression is used to determine the optimal coefficient in prediction. Evaluate the performance of the developed regression models in time series data. Prediction house prices are expected to help people who plan to buy a house so they can know the price range in the future, then they can plan their finance well. In addition, house price predictions are also beneficial for property investors to know the trend of housing prices in a certain location.

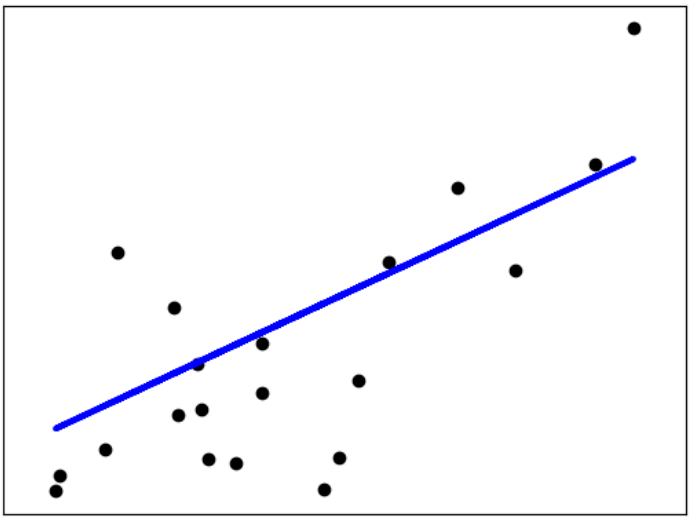
# MODELS USED :

## Linear Regression:

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

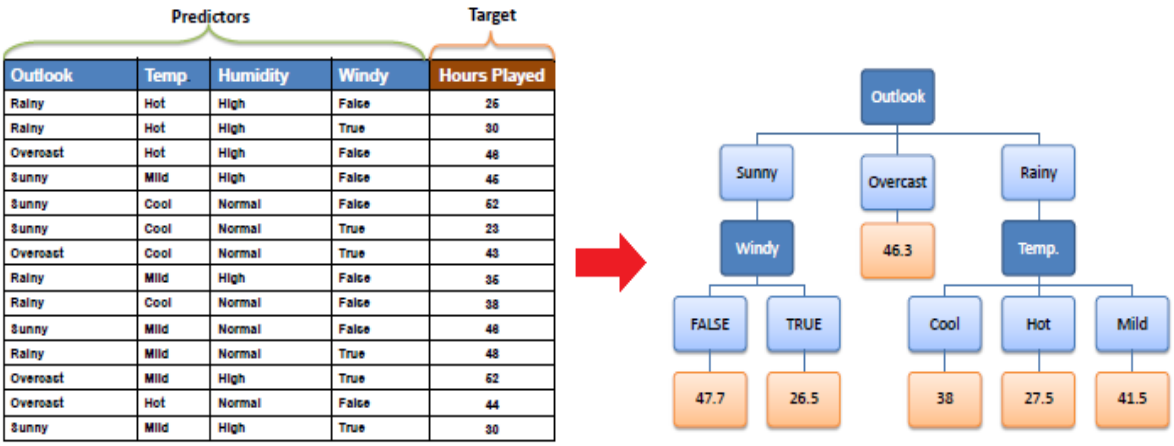
Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form $\boldsymbol{Y = a + bX}$, where $\boldsymbol{X}$ is the explanatory variable and $\boldsymbol{Y}$ is the dependent variable. The slope of the line is $\boldsymbol{b}$, and $\boldsymbol{a}$ is the intercept (the value of $\boldsymbol{y}$ when $\boldsymbol{x} = 0$).



## Decision Tree Regression:

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.



## Gradient Boost Regression:

**Gradient boosting** is a machine learning , technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Gradient boosting involves three elements:

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

**1. Loss Function:**

The loss function used depends on the type of problem being solved.

It must be differentiable, but many standard loss functions are supported and you can define your own.

For example, regression may use a squared error and classification may use logarithmic loss.

A benefit of the gradient boosting framework is that a new boosting algorithm does not have to be derived for each loss function that may want to be used, instead, it is a generic enough framework that any differentiable loss function can be used.

**2. Weak Learner:**

Decision trees are used as the weak learner in gradient boosting.

Specifically regression trees are used that output real values for splits and whose output can be added together, allowing subsequent models outputs to be added and "correct" the residuals in the predictions.

Trees are constructed in a greedy manner, choosing the best split points based on purity scores like Gini or to minimize the loss.

Initially, such as in the case of AdaBoost, very short decision trees were used that only had a single split, called a decision stump. Larger trees can be used generally with 4-to-8 levels.

It is common to constrain the weak learners in specific ways, such as a maximum number of layers, nodes, splits or leaf nodes.

This is to ensure that the learners remain weak, but can still be constructed in a greedy manner.

**3. Additive Model:**

Trees are added one at a time, and existing trees in the model are not changed.

A gradient descent procedure is used to minimize the loss when adding trees.

Traditionally, gradient descent is used to minimize a set of parameters, such as the coefficients in a regression equation or weights in a neural network. After calculating error or loss, the weights are updated to minimize that error.

Instead of parameters, we have weak learner sub-models or more specifically decision trees. After calculating the loss, to perform the gradient descent procedure, we must add a tree to the model that reduces the loss (i.e. follow the gradient). We do this by parameterizing the tree, then modify the parameters of the tree and move in the right direction by (reducing the residual loss.

Generally this approach is called functional gradient descent or gradient descent with functions.

*One way to produce a weighted combination of classifiers which optimizes [the cost] is by gradient descent in function space*

The output for the new tree is then added to the output of the existing sequence of trees in an effort to correct or improve the final output of the model.

A fixed number of trees are added or training stops once loss reaches an acceptable level or no longer improves on an external validation dataset.

## Packages:

- Pandas
- Scikit-learn
- Numpy
- Seaborn
- Matplotlib

## Pandas:

*pandas* is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
Python has long been great for data munging and preparation, but less so for data analysis and modeling. *pandas* helps fill this gap, enabling you to carry out your entire data analysis workflow in Python without having to switch to a more domain specific language like R.

- A fast and efficient **DataFrame** object for data manipulation with integrated indexing;
- Tools for **reading and writing data** between in-memory data structures and different formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format;
- Intelligent **data alignment** and integrated handling of **missing data**: gain automatic label-based alignment in computations and easily manipulate messy data into an orderly form;
- Flexible **reshaping** and pivoting of data sets;
- Columns can be inserted and deleted from data structures for **size mutability**;
- Aggregating or transforming data with a powerful **group by** engine allowing split-apply-combine operations on data sets;
- High performance **merging and joining** of data sets

## Scikit-learn:

Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. It's built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib!
The functionality that scikit-learn provides include:

- **Regression**, including Linear and Logistic Regression
- **Classification**, including K-Nearest Neighbors
- **Clustering**, including K-Means and K-Means++
- **Model selection**
- **Preprocessing**, including Min-Max Normalization

## Numpy:

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

## Seaborn:

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

- A dataset-oriented API for examining relationships between multiple variables
- Specialized support for using categorical variables to show observations or aggregate statistics
- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
- Automatic estimation and plotting of linear regression models for different kinds dependent variables
- Convenient views onto the overall structure of complex datasets
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
- Concise control over matplotlib figure styling with several built-in themes
- Tools for choosing color palettes that faithfully reveal patterns in your data

## Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code.

## METRICS :

## Mean Absolute Error :

***Mean Absolute Error***, also known as MAE, is one of the many metrics for ***summarizing and assessing the quality*** of a machine learning model.

Prediction Error → Actual Value - Predicted Value

This prediction error is taking for each record after which we convert all error to positive. This is achieved by taking Absolute value for each error as below;

Absolute Error → |Prediction Error|

Finally we calculate the mean for all recorded absolute errors (Average sum of all absolute errors).

MAE = Average of All absolute errors

$$mae = \frac{\sum_{i=1}^{n} abs\left(y_i - \lambda(x_i)\right)}{n}$$

## R2 score:

In statistics, the **coefficient of determination**, denoted $R^2$ or $r^2$ and pronounced "R squared", is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.

If $y_m$ is the mean of the observed data:

$$y_m=1/n \sum y_i$$

then the variability of the data set can be measured using three sums of squares formulas:

- The total sum of squares (proportional to the variance of the data):
  $$SS_{tot} =\sum(y_i-y_m)^2$$
- The regression sum of squares, also called the explained sum of squares:
  $$SS_{reg}=\sum(f_i-y_m)^2$$
- The sum of squares of residuals, also called the residual sum of squares:
  $$SS_{res}=\sum(y_i-f_i)^2$$

  The most general definition of the coefficient of determination is

$$R^2=1-SS_{res}/SS_{tot}$$

## Technologies Used:-

**Anaconda**:

Anaconda is a free and open-source[5] distribution ofthe Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system *conda*.[6] The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS.

**Anaconda distribution** comes with more than 1,500 packages as well as the conda package and virtual environment manager. It also includes a GUI, **Anaconda Navigator**[7], as a graphical alternative to the command line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, warning if this cannot be done.

Open source packages can be individually installed from the Anaconda repository[8], Anaconda Cloud (anaconda.org), or your own private repository or mirror, using the **conda install** command. Anaconda Inc compiles and builds all the packages in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

**Jupyter Notebook:**

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyternbconvert" command line interface in a shell.

To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

Jupyter Notebook interface

Jupyter Notebook provides a browser-based REPL built upon a number of popular open-source libraries:

- IPython
- ØMQ
- Tornado (web server)
- jQuery
- Bootstrap (front-end framework)
- MathJax

# IMPLEMENTATION:

## DataSet:

Data about home prices in Melbourne, Australia

1. Suburb: Suburb
2. Address: Address
3. Rooms: Number of rooms
4. Price: Price in Australian dollars
5. Method: S - property sold; SP - property sold prior; PI - property passed in; PN - sold prior not disclosed; SN -sold not disclosed; NB - no bid; VB - vendor bid; W - withdrawn prior to auction; SA - sold after auction; SS -sold after auction price not disclosed. N/A - price or highest bid not available.
6. Type: br - bedroom(s); h - house,cottage,villa, semi,terrace; u - unit, duplex; t - townhouse; dev site -
7. development site; o res - other residential.
8. SellerG: Real Estate Agent
9. Date: Date sold
10. Distance: Distance from CBD in Kilometres
11. Regionname: General Region (West, North West, North, North east ...etc)
12. Propertycount: Number of properties that exist in the suburb.
13. Bedroom2 : Scraped # of Bedrooms (from different source)
14. Bathroom: Number of Bathrooms
15. Car: Number of carspots
16. Landsize: Land Size in Metres
17. BuildingArea: Building Size in Metres
18. YearBuilt: Year the house was built
19. CouncilArea: Governing council for the area
20. Lattitude: Self explanitory
21. Longtitude: Self explanitory

22.Postcode: Postal Code

## Data Analysis:

### Data-exploration:

Load and explore the data.

read the data and store data in DataFrame titled melbourne_data

```
melbourne_data = pd.read_csv(melbourne_file_path)
```

print a summary of the data in Melbourne data

```
melbourne_data.describe()
```

The results show 8 numbers for each column in your original dataset. The first number, the **count**, shows how many rows have non-missing values.

### Dealing with missing values:

There are many ways data can end up with missing values. For example,

- A 2 bedroom house won't include a value for the size of a third bedroom.
- A survey respondent may choose not to share his income.

Most machine learning libraries (including scikit-learn) give an error build a model using data with missing values.

A Simple Option: Drop Columns with Missing Values¶

The simplest option is to drop columns with missing values.

| Bed | Bath |
|---|---|
| 1.0 | 1.0 |
| 2.0 | 1.0 |
| 3.0 | 2.0 |
| NaN | 2.0 |

→

| Bath |
|---|
| 1.0 |
| 1.0 |
| 2.0 |
| 2.0 |

Unless most values in the dropped columns are missing, the model loses access to a lot of (potentially useful!) information with this approach. As an extreme example, consider a dataset with 10,000 rows, where one important column is missing a single entry. This approach would drop the column entirely!

2) A Better Option: Imputation

Imputation fills in the missing values with some number. For instance, we can fill in the mean value along each column.

| Bed | Bath |
|---|---|
| 1.0 | 1.0 |
| 2.0 | 1.0 |
| 3.0 | 2.0 |
| NaN | 2.0 |

→

| Bed | Bath |
|---|---|
| 1.0 | 1.0 |
| 2.0 | 1.0 |
| 3.0 | 2.0 |
| 2.0 | 2.0 |

## Data Cleaning:

Cleaning the data is the most important step in data analysis. It is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. One of the things that can be noticed after visualizing the data is the existence of 'NaN' (Not a Number) values a.k.a NULL values, i.e. undefined or unrepresentable value. It is not possible to work with a data set having NULL values that is why we need to deal with them, either by giving them some defined values, or just removing them. Some of the features contain NULL values more than non-NULL values. These features are considered 'useless' and won't contribute much to the prediction later on.

## Data Visualization:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Statistical analysis is a process of understanding how variables in a dataset relate to each other and how those relationships depend on other variables. Visualization can be a core component of this process because, when data are visualized properly, the human visual system can see trends and patterns that indicate a relationship.

**Bar plots**

A familiar style of plot that accomplishes this goal is a bar plot. In seaborn, the barplot() function operates on a full dataset and applies a function to obtain the estimate (taking the mean by default).

**Scatter plot**

The scatter plot is a mainstay of statistical visualization. It depicts the joint distribution of two variables using a cloud of points, where each point represents an observation in the dataset. This depiction allows the eye to infer a substantial amount of information about whether there is any meaningful relationship between them.

**Line plot**

Scatter plots are highly effective, but there is no universally optimal type of visualization. Instead, the visual representation should be adapted for the specifics of the dataset and to the question you are trying to answer with the plot.

## Feature Selection:

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve. Irrelevant or partially relevant features can negatively impact model performance. Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output **.**Having irrelevant features in your data can decrease the accuracy of the models and make the model learn based on irrelevant features.

## Model Building:

Regression models are used to predict a continuous value. Predicting prices of a house given the features of house like size, price etc is a examples of Regression. It is a supervised technique.

Data is split into training and testing data. Training data is used while building the models. Linear regressions, Decision tree regressor, Gradient Booting Regressor are used.

## Model Evaluation:

After building each model r2 score and mean absolute error are calculated. Higher r2 score indicates higher accuracy.MEA indicates the error in the predicted value.

## Improve Accuracy:

Fine tuning machine learning predictive model is a crucial step to improve accuracy of the forecasted results.

Hyperparameter Tuning:

Hyperparameters in Machine Learning are user-controlled "settings" of your ML model. They influence how your model's parameters will be updated and learned during training. Finding the best hyper-parameters is usually done manually. It's a simple task of trial and error, with some intelligent guesstimating.

GridSearchCV:

Grid search is the process of performing hyper parameter tuning in order to determine the optimal values for a given model. This is significant as the

performance of the entire model is based on the hyper parameter values specified.
Grid search is an approach to parameter tuning that will methodically build and evaluate a model for each combination of algorithm parameters specified in a grid.

Log-transformation:
The log transformation, a widely used method to address skewed data, is one of the most popular transformations. The log transformation is, arguably, the most popular among the different types of transformations used to transform skewed data to approximately conform to normality. If the original data follows a log-normal distribution or approximately so, then the log-transformed data follows a normal or near normal distribution. The log transformation is to reduce the variability of data, especially in data sets that include outlying observations. Once the data is log-transformed, many statistical methods, including linear regression, can be applied to model the resulting transformed data.

# Conclusion and Future Scope:-

With the use of a variety of analytical and graphical tools, we were able to evaluate the predictive performance of various housing price models applied to real data of Melbourne. Furthermore, we were able to improve our models' prediction accuracy by fine-tuning the model. The models used in this project were compared and assessed using median absolute error as performance criteria.

The development of our project till now is just bound to predicting housing prices based on features that do not change with time. In addition to these features, there are various other factors in the market that affect the prices. Parameters like Economy, the inflation rate of an area may result in increase or decrease in the prices. If these data sets could be obtained, it would give clues to how these predictors affect housing prices. This information could prove useful for future investments and exploration .The further project development will be focusing on including these features thereby giving a more precise prediction of prices.

# REFERENCES


i. https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab
ii. https://www.kaggle.com/
iii. https://towardsdatascience.com/tagged/linear-regression
iv. https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html
v. https://docs.scipy.org/doc/numpy/user/quickstart.html
vi. https://scikit-learn.org/stable/tutorial/basic/tutorial.html