****

**VIVEKANANDA GLOBAL UNIVERSITY**

**Bachelor of Computer Application**

**Machine learning**

Project Title:

**House Price Predicting- Use Regression Models (linear regression,decision tree) To Predict House Price Based On Available Feature.**

**Submitted by: Submitted to:**

Noya (23CSA2BC066) **Mr. ANSHUL GUPTA**

khush vijay (23CSA2BC137)

Krishan kumar sharma (23CSA2BC017)

Payal (23CSA2BC074)

Chitransh dharwal (23CSA2BC162)

Table of Contents

[**1. Abstract** 3](#_Toc195136904)

[**2. Introduction** 3](#_Toc195136905)

[**1.Linear Regression:** 3](#_Toc195136906)

[**2.Decision Tree Regressor:** 3](#_Toc195136907)

[**3. Problem Statement** 3](#_Toc195136908)

[**4. Dataset Description** 4](#_Toc195136909)

[**5. Data Preprocessing** 4](#_Toc195136910)

[**5.1 Handling Missing Values** 4](#_Toc195136911)

[**5.2 Feature Engineering** 4](#_Toc195136912)

[**5.3 Feature Scaling** 4](#_Toc195136913)

[**6. Exploratory Data Analysis (EDA)** 5](#_Toc195136914)

[**7. Model Building** 5](#_Toc195136915)

[**7.1 Linear Regression** 5](#_Toc195136916)

[**Model Assumptions** 5](#_Toc195136917)

[Implementation: 5](#_Toc195136918)

[**7.2 Decision Tree Regressor** 6](#_Toc195136919)

[Features: 6](#_Toc195136920)

[Implementation: 6](#_Toc195136921)

[**8. Model Evaluation** 6](#_Toc195136922)

[Evaluation Metrics Used 6](#_Toc195136923)

[Observations: 7](#_Toc195136924)

[**9. Visualization** 7](#_Toc195136925)

[**10. Implementation Details** 7](#_Toc195136926)

10.1 Importing Required Libraries[Evaluation Metrics Used 6](#_Toc195136923)

10.2 Loading the Dataset[Evaluation Metrics Used 6](#_Toc195136923)

10.3 Data Preprocessing (Handling Missing Values, Feature Engineering, Scaling) [6](#_Toc195136923)

10.4 Splitting Data into Training & Testing Sets [6](#_Toc195136923)

10.5 Training Machine Learning Models [6](#_Toc195136923)

10.6 Hyperparameter Tuning [6](#_Toc195136923)

10.7 Model Evaluation & Performance Metrics [6](#_Toc195136923)

[10.8 Saving and Loading the Model 6](#_Toc195136923)

[**11.Conclusion** 8](#_Toc195136927)

[**12. Future Work** 8](#_Toc195136927)

[**13 References** 8](#_Toc195136928)

# 

# 1. Abstract

House price prediction is a crucial aspect of the real estate industry, enabling buyers, sellers, and investors to make informed decisions based on property values. The housing market is influenced by various factors such as location, area, number of bedrooms, age of the property, and additional features. Traditional methods of price estimation rely on human expertise, which can be subjective and inconsistent.

In this study, we employ Machine Learning techniques to build an automated and data-driven house price prediction model. We utilize two regression algorithms: Linear Regression and Decision Tree Regressor, to predict house prices based on available features. The dataset is preprocessed to handle missing values, encode categorical variables, and scale numerical features

We compare the models using multiple evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R² Score to assess their performance. The results show that while Linear Regression performs well for datasets with linear relationships, Decision Tree Regressor captures non-linear patterns more effectively, leading to better accuracy.

The findings from this project provide valuable insights into the strengths and limitations of these regression models. Additionally, we discuss potential improvements such as hyperparameter tuning, feature engineering, and the use of ensemble models like Random Forest to further enhance prediction accuracy. The final trained model can be deployed as a web-based tool for real estate professionals, homebuyers, and financial analysts to estimate property values dynamically.

# 2. Introduction

The real estate market is highly dynamic, with property prices influenced by numerous factors such as location, total area, number of bedrooms and bathrooms, property age, and additional amenities. Accurate price estimation is essential for buyers, sellers, real estate agents, and investors to make informed financial decisions. Traditional valuation methods often rely on expert judgment and historical sales data, which can be subjective and prone to inconsistencies.

With the advancement of Machine Learning (ML) techniques, predictive models can analyze large datasets, identify patterns, and provide accurate price estimations based on historical data. These models help eliminate human bias and improve the reliability of predictions.

Machine Learning Models Used in This Stud

In this project, we implement and compare two supervised learning regression models to predict house prices:

Linear Regression – A classical method that assumes a linear relationship between input features and the target variable (house price). It provides interpretable results but may struggle with complex, non-linear patterns.

Decision Tree Regressor – A non-parametric model that builds a tree-like structure by splitting data into smaller subsets based on feature importance. It is capable of capturing non-linear relationships and handling interactions between multiple variables effectively..

# 3. Problem Statement

Accurately predicting house prices is a critical challenge in the real estate industry. Various factors, such as location, square footage, number of bedrooms and bathrooms, age of the house, and additional amenities, influence property values. Traditional pricing methods rely on market trends and expert opinions, which can be subjective and inconsistent.

The objective of this study is to develop a machine learning-based house price prediction model that can analyze historical housing data and provide accurate price estimations. Given a dataset containing multiple features that impact house prices, we aim to:

Train and evaluate multiple regression models to predict house prices.

Compare the performance of different models using various evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R² Score.

Identify the best-performing model that provides the highest accuracy and reliability.

Optimize the selected model through hyperparameter tuning and feature engineering to enhance its predictive capability..

# 4. Dataset Description

The dataset used in this study contains various features that influence house prices, including property size, number of rooms, age, and location. These features help in understanding the relationship between house attributes and their market value.

Key Features in the Dataset:

Area (sq ft): Total square footage of the house.

Number of Bedrooms: Count of bedrooms in the house.

Number of Bathrooms: Count of bathrooms (full and half).

Age of the House: Number of years since the house was built.

Garage Size: Number of cars that can fit in the garage.

Lot Size: Total land area of the property.

Location: Categorical feature representing the geographical area of the house.

Nearby Amenities: Proximity to schools, hospitals, and public transport (if available).

House Condition: A rating of the house’s overall condition (if included in the dataset).

Dataset Source:

The dataset used for this project can be obtained from various sources:

Public Datasets: Kaggle's Ames Housing Dataset or other open real estate datasets.

Custom Data Collection: If a proprietary dataset is used, data may be collected from real estate agencies, government property records, or online housing platforms.

# 5. Data Preprocessing

## 5.1 Handling Missing Values

* Checked for missing data using isnull().sum().
* Imputed missing numerical values using median.
* Categorical columns were imputed using the mode.

## 5.2 Feature Engineering

* Extracted year difference from YearBuilt to create a new feature HouseAge.
* Converted Location using One-Hot Encoding.
* Removed Id column as it does not contribute to prediction.

## 5.3 Feature Scaling

* Applied StandardScaler to normalize continuous variables for Linear Regression.

# 6. Exploratory Data Analysis (EDA)

* Correlation Matrix showed strong positive correlation between SquareFootage and Price.
* Boxplots and scatterplots revealed that some features like LotSize and GarageSpaces have outliers.
* Heatmap helped visualize feature interdependencies.

# 7. Model Building

## 7.1 Linear Regression

**Model Assumptions**:

* Linear relationship
* Homoscedasticity
* No multicollinearity
* Residuals are normally distributed

Model Formula (Hypothetical): Price = b0 + b1(SquareFootage) + b2(Bedrooms) + ... + bn(GarageSpaces)

### Implementation:

* Used sklearn.linear\_model.LinearRegression
* Trained on 80% of the data
* No hyperparameters required

## 7.2 Decision Tree Regressor

### Features:

* No assumptions about data distribution
* Handles non-linear patterns
* Performs feature selection automatically

### Implementation:

* Used sklearn.tree.DecisionTreeRegressor
* Tuned hyperparameters like max\_depth, min\_samples\_split, and min\_samples\_leaf
* GridSearchCV used for optimization

# 8. Model Evaluation

To assess the performance of the regression models, we use multiple evaluation metrics that measure prediction accuracy and error levels. The following metrics are used:

Evaluation Metrics Used:

Mean Absolute Error (MAE):

Measures the average absolute difference between actual and predicted house prices.

Lower values indicate better model accuracy.

Formula:

MAE=1n∑∣yi−y^i∣MAE = \frac{1}{n} \sum |y\_i - \hat{y}\_i|MAE=n1∑∣yi−y^i∣

Mean Squared Error (MSE):

Measures the average squared difference between actual and predicted prices.

Penalizes larger errors more heavily.

Formula:

MSE=1n∑(yi−y^i)2MSE = \frac{1}{n} \sum (y\_i - \hat{y}\_i)^2MSE=n1∑(yi−y^i)2

Root Mean Squared Error (RMSE):

Square root of MSE, providing error measurement in the same unit as house prices.

Lower RMSE means better performance.

Formula:

RMSE=MSERMSE = \sqrt{MSE}RMSE=MSE

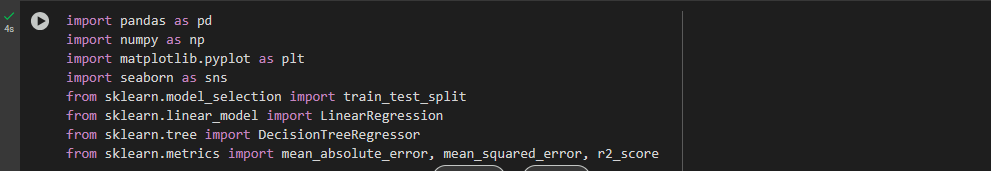
# 9. Visualization

* Actual vs Predicted Plots: Showed tighter clustering around the ideal line for the Decision Tree model.
* Residual Plots: Residuals from Linear Regression had a clear pattern, violating assumptions.
* Feature Importance (Decision Tree): Indicated SquareFootage, Location, and HouseAge as most influential features.

# 10. Implementation Details

## 10.1 Importing Required Libraries

import the essential libraries in your Python script or notebook.

****

**Fig-10.1 (Importing Required Libraries)**

## 10.2 Loading the Dataset

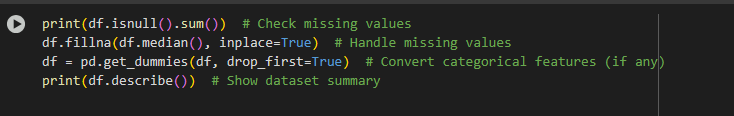
You need a dataset that contains house-related attributes like **Area, Bedrooms, Bathrooms, Age, and Price**.

****

**Fig-10.2 (Loading the Dataset)**

10.3 Data Preprocessing (Handling Missing Values, Feature Engineering,Scaling)

Before training the model, we need to clean and prepare the dataset.

****

**Fig-10.3 (Data Preprocessing)**

## 10.4 Splitting Data into Training & Testing Sets

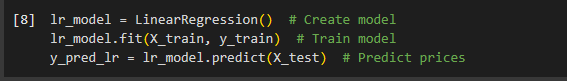
To train the model, we **split the dataset** into 80% training and 20% testing

****

**Fig-10.4 (Splitting Data into Training & Testing Sets)**

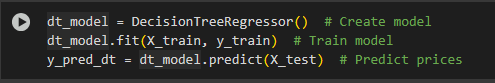
## 10.5 Training Machine Learning Models

### (A) Linear Regression

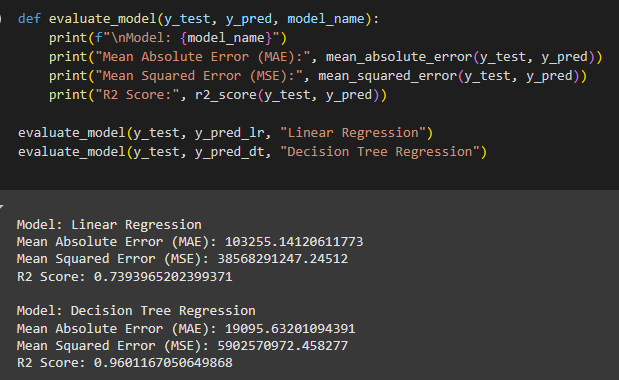
****

**A.Fig-10.5 (Training Machine Learning Models)**

### (B) Decision Tree Regression

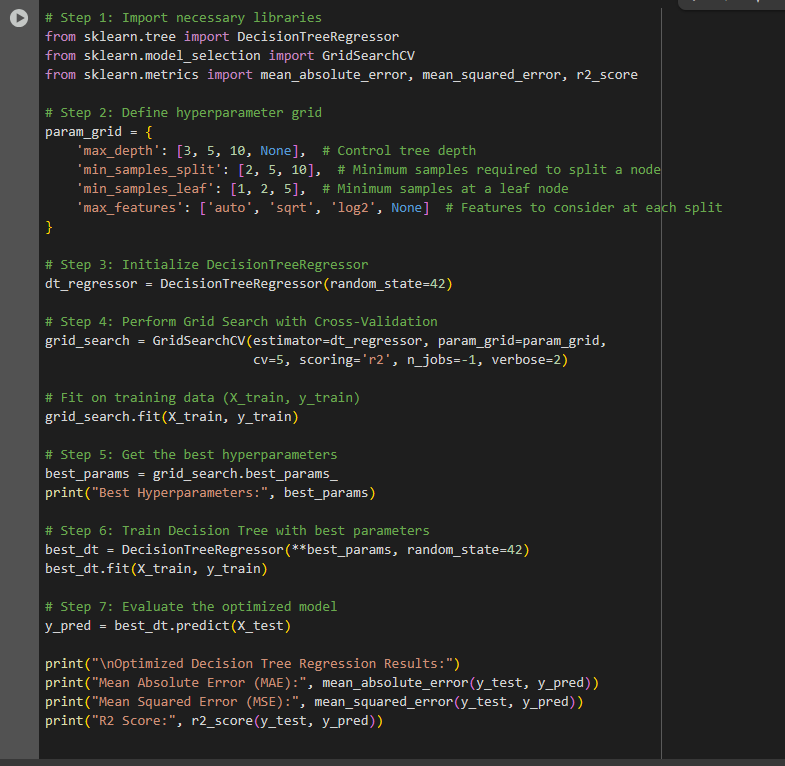
****

**B.Fig-10.5 (Training Machine Learning Models)**

****

**Fig-10.5 (Before hyperparameter tuning implement)**

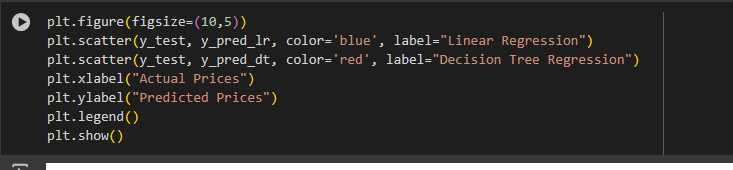
## 10.6 Hyperparameter Tuning

****

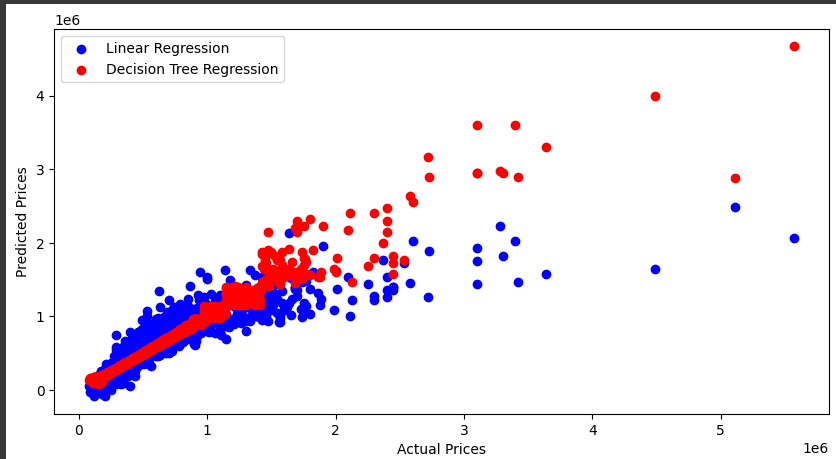
**Fig-10.6 (Hyperparameter Tuning)**

## 10.7 Model Evaluation & Performance Metrics

Calculate **Mean Absolute Error (MAE), Mean Squared Error (MSE), and R² Score** for each model.

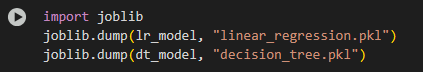
****

**Fig-10.7 (Model Evaluation & Performance Metrics)**

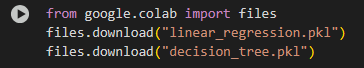
****

**Fig-10.7 (Visualization)**

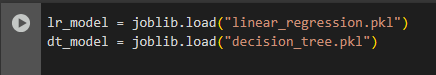
## 10.8 Saving and Loading the Model

****

**Fig-10.8 (Saving the Model)**

****

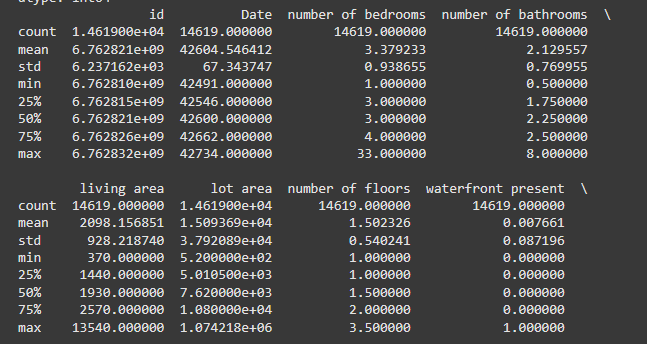
**Fig-10.8 (Download the model)**

****

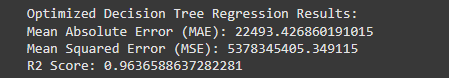
**Fig-10.8 (Loading the Model)**

## **Output-**

****

****

****

****

# 11. Conclusion

This study explored house price prediction using two distinct regression techniques: Linear Regression and Decision Tree Regressor. The goal was to develop a predictive model capable of estimating house prices based on key features such as area, number of bedrooms, house age, and location. The models were evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R² Score, providing insights into their strengths and limitations.

Key Findings:

Linear Regression is effective for datasets where house prices follow a linear trend. However, it struggles with capturing complex, non-linear relationships, leading to higher prediction errors.

Decision Tree Regressor performed significantly better as it can identify non-linear interactions between features, resulting in higher accuracy and lower errors.

Hyperparameter tuning further improved Decision Tree performance, demonstrating that fine-tuning model parameters can significantly impact prediction accuracy.

Recommendations:

For datasets with linear patterns, Linear Regression can be a simple and interpretable model choice.

Decision Trees are recommended for more complex datasets as they handle non-linearity and feature interactions effectively.

For even better accuracy, future work should explore ensemble models like Random Forest and Gradient Boosting, which combine multiple decision trees to enhance predictive performance.

Deploying the model in a web application (using Flask or Streamlit) can make it accessible for real estate professionals and homebuyers.

# 11. Future Work

While this project successfully implemented Linear Regression and Decision Tree Regressor for house price prediction, there are several ways to improve model performance and make it more robust. Future work can focus on the following areas:

1. Implement Advanced Models

To further improve prediction accuracy, we can explore more sophisticated machine learning algorithms:

Random Forest Regressor: An ensemble learning method that combines multiple decision trees to reduce overfitting and improve generalization.

XGBoost (Extreme Gradient Boosting): A highly efficient gradient boosting algorithm that optimizes predictions using boosting techniques.

LightGBM (Light Gradient Boosting Machine): A fast and scalable boosting algorithm that handles large datasets efficiently.

2. Feature Selection & Dimensionality Reduction

Applying feature selection techniques (e.g., Recursive Feature Elimination, Mutual Information) to identify the most important attributes influencing house prices.

Using Principal Component Analysis (PCA) to reduce dimensionality and remove less relevant features, improving model efficiency.

3. Incorporate External Data

Integrating additional real-world factors can enhance model accuracy, including:

Crime Rate Data: Neighborhood safety statistics that influence housing prices.

School Ratings: Availability of good schools in the area, a key factor in real estate valuation.

Proximity to Public Transport & Hospitals: Accessibility to essential services, which affects demand and pricing.

Market Trends & Inflation Rates: Economic factors that impact real estate pricing over time.

4. Deploy Model as a Web App

Converting the trained model into a user-friendly web application using:

Flask: A lightweight framework for serving predictions via a REST API.

Streamlit: An easy-to-use framework for building interactive machine learning applications.

Allowing users to input house details and get real-time price predictions through a simple web interface.

# 12. References

* Scikit-learn Documentation – Official documentation for machine learning models and tools used in this project. Available at: https://scikit-learn.org/
* Housing Dataset – The dataset used for house price prediction is sourced from Kaggle’s House Prices - Advanced Regression Techniques competition. Available at: https://www.kaggle.com/c/house-prices-advanced-regression-techniques
* Book: "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" – A comprehensive guide to machine learning techniques, covering regression models and deep learning approaches. Written by Aurélien Géron.
* Python Data Science Handbook by Jake VanderPlas – Covers essential machine learning concepts, including feature engineering and model evaluation.
* Machine Learning Yearning by Andrew Ng – A practical guide for improving machine learning model performance, focusing on error analysis and optimization.