# Report on

**Predicting House Prices Using Supervise Learning**

**By**

**Prabhav Kansal – 202410116100143**

**Priyanshu Kansal – 202410116100155**

**Sagar Gupta – 202410116100174**

# Session:2024-2025 (II Semester)

Under the supervision of

# **Ms. Komal Salgotra (Assistant Professor)**

****

**DEPARTMENT OF COMPUTER APPLICATIONS**

**KIET GROUP OF INSTITUTIONS,DELHI-NCR,**

**GHAZIABAD- 201206**

**INTRODUCTION**

In the world of real estate, accurately predicting house prices is a crucial task for buyers, sellers, investors, and real estate agents. One effective way to achieve this is through the use of supervised learning, a type of machine learning where a model is trained on a labeled dataset—that is, a dataset where the input features and corresponding target outputs (house prices, in this case) are known. The model learns the relationship between the inputs and outputs so that it can predict future values accurately.

Supervised learning involves two main types of problems: classification and regression. House price prediction falls under regression, as the goal is to predict a continuous numeric value. The process typically begins by collecting a dataset of historical house sales. This data usually includes various features (also called independent variables) that influence the price of a house, such as the number of bedrooms, size in square feet, location, age of the property, number of bathrooms, presence of a garage, and more. The target variable (dependent variable) is the sale price of the house.

Once the data is collected, it is cleaned and preprocessed. This step involves handling missing values, converting categorical data into numerical form (using techniques like one-hot encoding), scaling the data, and removing outliers. After preprocessing, the dataset is split into training and testing sets. The training set is used to teach the model how the features relate to house prices, while the testing set is used to evaluate how well the model performs on unseen data.

There are several supervised learning algorithms that can be used for house price prediction. Linear regression is a common starting point due to its simplicity and interpretability. It assumes a linear relationship between the features and the price. More advanced models like decision trees, random forests, gradient boosting machines (e.g., XGBoost), and neural networks can capture more complex relationships and often provide better accuracy.

During training, the model adjusts its internal parameters to minimize the difference between the predicted prices and the actual prices in the training set. This difference is quantified using error metrics such as Mean Squared Error (MSE) or Mean Absolute Error (MAE). After training, the model is tested on the testing data to evaluate its generalization performance.

The effectiveness of a house price prediction model depends on various factors, including the quality of the data, the choice of features, and the algorithm used. Feature engineering—selecting or creating meaningful variables—plays a critical role in improving model performance. Additionally, hyperparameter tuning and cross-validation can be applied to further refine the model.

**OBJECTIVES OF THE PROJECT**

This project aims to achieve the following objectives:

1. **Build a Predictive Model:**  
   Develop a machine learning model to predict house prices using supervised learning.

**2. Feature Identification:**  
Identify and analyze important features that impact house pricing.

**3. Algorithm Implementation:**  
Apply and compare different regression algorithms such as Linear Regression, Decision Trees, and Random Forests.

**4. Data Preprocessing:**  
Clean the dataset, handle missing values, encode categorical variables, and normalize features.

**5. Model Evaluation:**  
Evaluate model accuracy using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R²).

**6. Hyperparameter Tuning:**  
Optimize model performance using cross-validation and hyperparameter tuning techniques.

**7. Data Visualization:**  
Use visual tools (e.g., graphs and charts) to explore data and present model results.

**8. User Interface (Optional):**  
Create a simple user interface for users to input house features and receive price predictions.

**METHODOLOGY**

The methodology for exploring healthcare data in Python involves a structured approach to understanding, cleaning, and analyzing the dataset. Here’s a step-by-step guide:

### **1. Problem Definition**

The first step is clearly defining the problem. The objective of this project is to build a machine learning model that can predict the price of a house based on input features such as the number of bedrooms, size of the property, location, and other relevant factors. Since the target output is a continuous value (price), this is a **regression problem**, a subtype of supervised learning.

### **2. Data Collection**

Data is the foundation of any machine learning model. In this step, a dataset containing historical house sales is collected. This dataset includes various features (independent variables) like square footage, location, number of rooms, number of bathrooms, year built, and more, along with the target variable (dependent variable), which is the house price.

### **3. Data Preprocessing**

Real-world data often contains inconsistencies, missing values, or unformatted entries. Preprocessing ensures the data is clean and suitable for training. This step includes:

* **Handling Missing Values:** Replacing or removing rows/columns with null values.
* **Encoding Categorical Variables:** Converting categorical data (e.g., location, house style) into numerical format using techniques like label encoding or one-hot encoding.
* **Feature Scaling:** Standardizing numerical features to ensure they are on the same scale.
* **Removing Outliers:** Identifying and removing outliers that could distort model learning.

### **4. Feature Selection and Engineering**

Not all features contribute equally to price prediction. In this step, important features are selected based on correlation analysis, and new features may be engineered by combining or transforming existing ones. This enhances the model's learning capability and reduces complexity.

### **5. Splitting the Dataset**

The dataset is split into two subsets: a **training set** and a **testing set**, typically using an 80:20 or 70:30 ratio. The training set is used to train the model, while the testing set evaluates the model's performance on unseen data to simulate real-world predictions.

### **6. Model Selection**

Multiple regression algorithms are considered for model development. Common algorithms include:

* **Linear Regression:** A basic model assuming a linear relationship between features and price.
* **Decision Tree Regressor:** A tree-based model that captures non-linear relationships.
* **Random Forest Regressor:** An ensemble model that combines multiple decision trees for improved accuracy and generalization.

Each model is tested, and the best-performing one is selected for final predictions.

### **7. Model Training**

The selected algorithm is trained using the training dataset. During this process, the model learns the patterns and relationships between the features and house prices by minimizing the error between predicted and actual prices.

**CONCLUSION**

This project demonstrates the practical application of supervised learning techniques in predicting house prices based on historical housing data. By leveraging machine learning algorithms such as Linear Regression, Decision Trees, and Random Forests, the project successfully establishes a model capable of estimating house prices with reasonable accuracy.

The process involved several critical steps, starting from data collection and preprocessing to model selection, training, evaluation, and optional deployment. Each step played a vital role in enhancing the model’s performance and ensuring reliable predictions. Through feature selection and engineering, the model was able to focus on the most influential factors affecting house prices, such as location, size, number of rooms, and other property characteristics.

Evaluation using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R²) ensured that the model was not only accurate but also generalizable to unseen data. Hyperparameter tuning further helped optimize the models for better results.

In conclusion, supervised learning provides an effective and scalable solution for real estate price prediction. The outcomes of this project highlight the power of data-driven decision-making and its potential to assist buyers, sellers, and investors in making informed choices. With further enhancements, such as incorporating real-time data and deploying the model through a user-friendly interface, this system can serve as a valuable tool in the real estate industry.

In conclusion, supervised learning offers a reliable and efficient approach to house price prediction. As the housing market continues to evolve, such data-driven tools will become increasingly important for buyers, sellers, investors, and agents. Future enhancements can include integrating more dynamic datasets, using deep learning for further accuracy, and deploying the model on a web or mobile platform to increase accessibility and usability.

**CODE AND OUTPUT**

import pandas as pd

df=pd.read\_csv("kc\_house\_data.csv", delimiter= ",")

df.head()

df.shape

df.info()

df.isnull()

df.isnull().sum()

import numpy as np

df["bathrooms"]=np.round(df["bathrooms"])

df['date'] = pd.to\_datetime(df['date'])

df["bedrooms"].value\_counts()

df[df['bedrooms'] == 33]

df.loc[15870, 'bedrooms'] = 3

df["bedrooms"].value\_counts()

import matplotlib.pyplot as plt

import seaborn as sns

plt.title("Histogram of Different Number of Bedrooms")

plt.xlabel("Number of Bedrooms")

df["bedrooms"].plot.hist()

plt.title("Histogram of Different Number of Bathrooms")

plt.xlabel("Number of Bathrooms")

df["bathrooms"].plot.hist()

plt.title("Histogram of Different Number of Floors")

plt.xlabel("Number of Floors")

df["floors"].plot.hist()

plt.title ("Histogram of Different Square Footage of the Living Space")

plt.xlabel("Square Footage of the Living Space")

df["sqft\_living"].plot.hist()

sns.distplot(df["price"], bins=50, hist=True, kde=True)

sns.boxplot(x= "bedrooms", y="price", data=df)

sns.boxplot(x= "floors", y="price", data=df)

new\_data= df.copy()

new\_data= new\_data.drop(["id", "date"], axis=1)

new\_data.head()















