Predicting house prices using machine learning :

INTRODUCTION

1. Data Collection:

Gather a dataset that contains relevant features like square footage, number of bedrooms, location, etc., as well as the corresponding house prices.

2. Data Cleaning:

Check for missing values, outliers, and inconsistencies in your dataset. Decide how to handle missing data, either by imputing values or removing rows with missing data. Address outliers appropriately.

3. Feature Selection/Engineering:

Select the most relevant features for your model. This can involve removing irrelevant features and creating new ones, like feature scaling or one-hot encoding for categorical variables.

4. Data Splitting:

Divide your dataset into a training set and a testing set. This allows you to train your model on one portion of the data and evaluate its performance on another.

5. Feature Scaling:

Normalize or standardize your numerical features to bring them to a similar scale. This can be important for some machine learning algorithms.

6. Handling Categorical Data:

If your dataset contains categorical variables, encode them using techniques like one-hot encoding to make them usable for machine learning algorithms.

7. Addressing Imbalanced Data:

If your dataset has an imbalance in the target variable, you may need to apply techniques like oversampling or undersampling to balance the classes.

8. Data Splitting:

Split your data into training and testing sets to evaluate your model's performance accurately.

9. Model Selection:

Choose an appropriate machine learning model for regression, such as linear regression, decision trees, random forests, or gradient boosting.

10. Model Training:

Train your selected model on the training data. Ensure you tune hyperparameters and cross-validate if necessary.

11. Model Evaluation:

Evaluate your model's performance on the testing data using appropriate regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

12. Hyper parameter Tuning:

Fine-tune your model's hyperparameters to optimize its performance.

13.Prediction:

Once you're satisfied with your model's performance, you can use it to make predictions on new, unseen data.

14.Deployment:

If you intend to use your model in a real-world application, deploy it for serving predictions.