Predicting House Prices using Machine Learning

Phase 1: Problem Definition and Design Thinking

Problem Definition:

The problem is to predict house prices using machine learning techniques. The objective is to develop a model that accurately predicts the prices of houses based on a set of features such as location, square footage, number of bedrooms and bathrooms, and other relevant factors. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

Design Thinking:

Data Source:

The Data set for the problem statement is collected from Kaggle.com. The dataset contains information about houses, including features like location, square footage, bedrooms, bathrooms, and price.

Dataset Link: **https://www.kaggle.com/datasets/vedavyasv/usa-housing**

Data Preprocessing:

Data preprocessing is a critical step in the data science workflow that involves cleaning, transforming, and organizing raw data into a format suitable for analysis and modeling. Proper data preprocessing is essential because the quality of the data you feed into your models significantly impacts the accuracy and reliability of your results.

Here are the typical steps involved in data preprocessing:

- **Handling Missing Data:**

- Remove or fill missing values using techniques like mean, median, or interpolation.

- **Handling Duplicates:**
  Identify and remove duplicate records.

- **Handling Outliers:**
  Detect and either remove or transform outliers using methods like z-sore, IQR, or domain knowledge.

- **Data Formatting:**
  Standardize or normalize data to ensure consistent units and scales.

- **Data Visualization:**
  Create visualizations to explore and understand the data better.

- **Feature Selection:**
  Choose the most relevant features for modeling, considering factors like feature importance, correlation, and domain knowledge.

Model Selection:

The choice of the model depends on the characteristics of your data, the underlying assumptions, and the problem you are trying to solve.

Depending on your data and problem complexity, you may need to explore more complex models, such as:

- Linear Regression
- Lasso Regression
- Decision Tree Regressor
- Random Forest Regressor
- Support Vector Regressor
- Neural Network

Model Training:

Model training is a fundamental step in data science and machine learning where you use a dataset to teach a machine learning model to make predictions or infer patterns.

Here's a step-by-step guide on how model training works:

- **Data Splitting:**
  Divide the dataset into training, validation, and testing sets to evaluate model performance.

- **Hyperparameter Tuning:**
  Machine learning models often have hyperparameters (e.g., learning rate, number of trees in a random forest) that need to be tuned for optimal performance.
  Use techniques like grid search, random search, or Bayesian optimization to find the best hyperparameters.

- **Model Training:**
  Train the selected model on the training dataset using the chosen hyperparameters.
  The model learns from the training data by adjusting its internal parameters to minimize a loss or error function.

Evaluation:

After training, evaluate the model's performance on the validation dataset.

Use appropriate evaluation metrics (e.g., accuracy, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.) depending on your problem type.

Final Model Selection:

Once you're satisfied with the model's performance on the validation set, select the best model configuration.

Assess the final model's performance on the test dataset, which it has never seen before.

This provides an unbiased estimate of how well your model will perform on new, unseen data.