Machine Learning Project Report: House Price Prediction

1. Define the Problem

In today's real estate market, property prices vary significantly depending on several factors such as area, location, number of bedrooms, presence of amenities, and more. Manually estimating the price of a house based on these features can be inaccurate and time-consuming.

Objective: Build a machine learning model capable of predicting house prices based on input features.

Why Machine Learning? This is a **supervised learning** problem because the dataset contains labeled data: input features (e.g., area, number of bedrooms) and a target value (price). The model will learn from this labeled data and then generalize to predict prices for new data.

Goal: Develop a regression model that accurately estimates the price of a house.

2. Data Collection

Machine learning models rely heavily on high-quality data. Collecting relevant and clean data is the first major step.

Common Sources of Data:

- Ready datasets: CSV, Excel, SQL databases
- Public platforms:
  - *Kaggle* (general-purpose datasets)
  - *Roboflow* (images)
  - *HuggingFace* (NLP and text data)
- APIs: Used to pull real-time data (e.g., Twitter, weather APIs)
- Web scraping: Extracting information from websites
- Manual entry: Collecting data manually when it's not available online

The dataset was sourced from Kaggle and loaded using a standard data import method.

3. Data Representation

Before feeding the data into a machine learning model, we must represent it in a structured form.

What is Data Representation? It means organizing raw data into a format that is readable and useful for ML models, such as:

- Tables (rows = instances, columns = features)
- Numeric encoding of categorical variables

The dataset is stored in a pandas DataFrame with features like:

- area
- bedrooms
- bathrooms
- furnishingstatus
- price (target)

4. Data Wrangling

This phase involves cleaning and preparing the dataset to remove inconsistencies or irrelevant data.

Common Steps:

- Remove unnecessary columns
- Drop duplicate rows
- Handle missing values

The dataset was cleaned to remove missing values, duplicates, and irrelevant records.

5. Data Analysis

This step explores the data to find patterns, relationships, and insights.

Univariate Analysis:

- **Price Distribution**: Most houses are priced between 3.5 to 4.5 million.
- **Area Distribution**: Most properties range from 1,000 to 8,000 sq. meters.
- **Bedrooms/Bathrooms/Stories/Parking**:
    - Most have 3 bedrooms
    - 1 bathroom is most common
    - 1-2 stories
    - Usually 0 parking spaces
- **Amenities (Mainroad, Guestroom, etc.):**
    - Most are on the main road
    - Guest rooms and basements are rare
    - Hot water heating and air conditioning are not common but increase price

Bivariate Analysis:

- **Price vs Area**: Positive correlation – larger areas tend to cost more
- **Price vs Bedrooms**: Slight upward trend
- **Price vs Bathrooms**: More bathrooms usually mean higher prices
- **Price vs Stories**: Higher stories can increase price range
- **Price vs Mainroad / Guestroom / Air Conditioning / Heating / Furnishing**:
    - Properties with these features tend to have higher prices

Multivariate Analysis:

Examining interactions between multiple features:

- (price, area, bedrooms, bathrooms)
- (price, stories, mainroad, guestroom, basement)
- (price, hotwaterheating, airconditioning, parking, furnishingstatus)

Tools Used: .describe(), matplotlib, seaborn, various plots (scatter, box, violin, bar)

Regression Models Overview

🔷 Linear Models:

LinearRegression: Fits a straight line to model the relationship between features and the target.

SGDRegressor: Uses stochastic gradient descent, suitable for large-scale datasets.

Lasso (L1 Regularization): Shrinks some feature coefficients to zero, promoting feature selection.

Ridge (L2 Regularization): Penalizes all coefficients, reducing overfitting.

🔷 Support Vector Machines:

SVR: Effective for capturing complex, non-linear relationships using support vectors.

LinearSVR: A faster, linear-only variant of SVR.

🔷 Distance-Based Models:

KNeighborsRegressor: Predicts a value by averaging the outcomes of the K nearest data points.

🔷 Tree-Based Models:

DecisionTreeRegressor: Splits the data based on decision rules; effective for non-linear patterns.

🔷 Ensemble Methods:

Bagging (Bootstrap Aggregation):

BaggingRegressor: Combines multiple models trained on random subsets of the data.

RandomForestRegressor: An ensemble of decision trees; reduces overfitting and increases accuracy.

ExtraTreesRegressor: Introduces extra randomness, leading to faster training and better variance handling.

Boosting (Sequential Learning):

XGBoost: Builds models sequentially, where each one corrects the errors of the previous.

CatBoost: Efficiently handles categorical variables without preprocessing.

LightGBM: A gradient boosting framework optimized for speed and memory efficiency on large datasets.

Hybrid Methods:

VotingRegressor: Averages predictions from multiple models to improve overall accuracy.

StackingRegressor: Combines several base models and feeds their outputs into a final model for improved predictions.

8. Conclusion

A complete machine learning pipeline was developed to predict house prices using structured tabular data. Every step, from data collection and preprocessing to modeling and evaluation, contributed to building an effective prediction model.

Key Learnings:

- Effective data cleaning and exploration are essential before modeling
- Feature encoding and scaling have a noticeable impact on prediction quality
- Comparing multiple algorithms helps select the most accurate one

Future Improvements:

- Implement cross-validation and grid search for hyperparameter tuning
- Incorporate external data sources (e.g., neighborhood info, zip code demographics)
- Deploy the best model via an interactive web interface using tools like Streamlit or Flask