**PREDICTION OF HOUSE PRICES USING MACHINE LEARNING**

**Topic:** In this part continue building the house price prediction model by Feature Selection, Model Training and Evaluation.

**Introduction to Prediction of House Prices:**

* Predicting house prices using machine learning involves using algorithms and data to estimate the market value of a property based on its features, such as location, size, and amenities.
* Key steps include data pre-processing, feature engineering, model selection, and rigorous evaluation using metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE).
* Ensuring model interpretability and addressing ethical considerations, such as bias, are essential. Deployment in a user-friendly interface and continuous monitoring for updates and market trends are crucial for maintaining accuracy in real-world applications.

**Solutions:**

1. Data Collection:

* Gather a comprehensive dataset that includes relevant information about houses. This could encompass data sources such as real estate list property tax records or publicly available datasets.
* Ensure that your dataset has a variety of features including both numerical (e.g., square footages number of bedrooms) and categorical (e.g., locations type of house) variables.

1. Data Pre-processing:

* Identify and handle mission data:

Impute missing values with averages or medians for numerical features and use mode for categorical features or you may decide to remove rows or

Columns with excessive missing data .

* Outlier detection and treatment:

Identify outliers in the data and decide whether to remove them and transform them or leave them as is based on domain knowledge.

* Error Correction:

Check for data entry errors and correct them if necessary.

1. Feature Engineering:

* Utilize techniques like correlation analysis to identify relationships between features and the target variable.
* Employ features importance scores from tree-based models like Random Forests or Gradient Boost to select the most relevant features.

1. Model Training:

* Split your dataset into training and testing sets to access model perfomance .
* Train the selected model on the training data using the chosen algorithm. The model will adjust its parameters to learn patterns in the data.

1. Model Evaluation:

* Use regression evaluation metrics:
* Mean Absolute Error (MAE): Measures the average squared difference between predicted predicted and actual values.
* Root Mean Squared Error (RMSE): RMSE is the square root of MSE and provides a more interpretable measure.
* Perform cross-validation to access how well the model generalizes to new data and to dataset overfining.

1. Hyper Parameter Tuning:

* Experiment with different hyper parameter to optimize model performance. You can use techniques like grid search or random search to find the best combination of hyper parameters.

1. Deployment:

* Once you have a well-perform models deploy it as a service or interate it into a web application where users can input house features and net price predictions.

1. Monitoring and Maintenance:

* Continuously monitor the models performance in a real-world and retrain it periodically with new data to keep it up to date.

**Tools used:**

Prediction of house prices with machine learning involves several keys steps and tools. First, you need to gather and pre-process your dataset, using libraries like Pandas and Numpy for data manipulation and cleaning. Features engineering is crucial, as it helps select and create relevant features that can improve the model’s

accuracy. Next, you can employ popular machine learning libraries such as Scikit-Learn for traditional regression models or Tensorflow and PyTorch for deep learning-based approaches. Ensemble methods like XGBoost and LightGBM can enhance predictive power. Data visualization with Matplotlib and Seaborn can provide insights into dataset. Throughput this process, Jupter Notebooks offer an interactive environment for model development and testing. Overall, success in predicting house prices hinges on data quality, feature engineering, and the appropriate choice of algorithm for the specific task.

**Program:**

**# Import necessary libraries**

**import pandas as pd**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.linear\_model import LinearRegression**

**from sklearn.metrics import mean\_squared\_error**

**# Load your dataset (replace 'your\_dataset.csv' with your actual dataset file)**

**data = pd.read\_csv('your\_dataset.csv')**

**# Split the data into features (X) and target (y)**

**X = data.drop('Price', axis=1)**

**y = data['Price']**

**# Split the data into training and testing sets**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# Create and train the Linear Regression model**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**# Make predictions**

**y\_pred = model.predict(X\_test)**

**# Evaluate the model**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**printf("Mean Squared Error: {mse}")**

**Evaluation :**

1. **Import Libraries:**

**The code begins by importing necessary libraries. `pandas` is used for data manipulation, `train\_test\_split` from `sklearn.model\_selection` for splitting the data, `LinearRegression` from `sklearn.linear\_model` for creating and training the model, and `mean\_squared\_error` from `sklearn.metrics` for calculating the Mean Squared Error.**

**2.Data Splitting:**

**The dataset is divided into features (X) and the target variable (y). In this context, features represent attributes of the houses (e.g., square footage, number of bedrooms), and the target variable represents house prices.**

**3. Data Splitting (Training and Testing):**

**The dataset is further divided into training and testing sets using `train\_test\_split`. In this example, 80% of the data is used for training (`X\_train`, `y\_train`), and 20% is used for testing (`X\_test`, `y\_test`).**

**4. Model Creation and Training:**

**A Linear Regression model is created using `LinearRegression()`, and it's trained on the training data with `model.fit(X\_train, y\_train)`.**

**5. Prediction:**

**The model is used to make predictions on the testing data using model.predict(X\_test)`, and the predicted house prices are stored in `y\_pred`.**

**6. Evaluation (Mean Squared Error):**

**The code calculates the Mean Squared Error (MSE) as a measure of the model's performance. MSE is computed using ` mean\_squared\_error(y\_test, y\_pred)`, which compares the predicted house prices (`y\_pred`) to the actual house prices (`y\_test`). A lower MSE indicates better predictive accuracy**

**Conclusion:**

The code presented is a concise python script for predicting house prices using machine learning. It encompasses data loading,splitting,model training, prediction, and Mean Squared Error (MSE) evaluation.