House Price Prediction using Machine

Learning

Phase 5

Introduction:

With the introduction of the power of machine learning in predicting house prices using Python has revolutionized the real estate industry. In this article, we explore the dynamic world of house price prediction using cutting-edge machine-learning techniques. By harnessing the vast potential of data analysis, feature engineering, and model training in Python, we aim to provide a comprehensive guide that equips readers with the tools to make informed decisions in the ever-changing housing market.

Linear regression for house price prediction

Linear regression is a mainly used technique for the prediction of house prices due to its simplicity and interpretability. It assumes a linear relationship between the independent variables (such as how many bedrooms, number of bathrooms, and square footage) and the dependent variable (house price). By fitting a linear regression model to historical data, we can estimate the coefficients that represent the relationship between the target variable and the features. This enables us to make predictions on new data by multiplying the feature values with their respective coefficients and summing them up. Linear regression provides insights into the impact of each feature on the house price, enabling us to understand the significance of different factors and make informed decisions in the real estate market.

House price prediction using machine learning

Machine learning involves training a computer to recognize patterns and make predictions based on data. In the case of house price prediction, we can use historical data on various features of a house, such as its location, size, and amenities, to train a machine-learning model. Once the model is trained, it can analyze new data on a given house and make a prediction of its market value.

House price prediction using machine learning(Linear regression model)

* Import the required libraries and modules, including pandas for data manipulation, scikitlearn for machine learning algorithms, and LinearRegression for the linear regression model.

* Loading the required dataset with pd.read\_csv and select the features we want to use for prediction (e.g., bedrooms, bathrooms, sqft\_living, sqft\_lot, floors, and zip code), as well as the target variable (price).

* Split the data into a training set and a test set using the train\_test\_split function, with 80% of the data used for training and 20% for testing.
* Create an instance of the linear regression model using LinearRegression(). We then perform the model training by calling the function fit() with the training data.

* Once the model is trained, we make predictions for the test data set using predict and store the results in y\_pred.

To evaluate the performance of the model, we calculate the R^2 score using the score for the test set.

* Demonstrate how to predict the price of a new house by creating a new dataframe new\_house with the features of the house. We pass this dataframe to the model’s prediction function to obtain the predicted price.

.

. code :

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression import pandas as pdd

# Loading the dataset data\_h = pdd.read\_csv('kc\_house\_data.csv')

# Selecting the features and target variable

|  |  |
| --- | --- |
| Features1 = ['bedrooms', 'bathrooms', 'sqft\_living', 'sqft\_lot', | |
| 'floors', 'zipcode'] |  |

target = 'price' X1 = data\_h[features1] y1 = data\_h[target]

# We will perform the data splitting into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X1, y1, test\_size=0.2, random\_state=42)

# instance of the Linear Regression model creation model = LinearRegression()

# Training the model model.fit(X\_train, y\_train)

# Making predictions on the test set y\_pred = model.predict(X\_test)

# Evaluating the model score = model.score(X\_test, y\_test) print("Model R^2 Score:", score) # Predicting the price of a new house

new\_house = pdd.DataFrame({'bedrooms': [2], 'bathrooms': [2.5],

|  |  |
| --- | --- |
| 'sqft\_living': [600], 'sqft\_lot': [600], 'floors': [2], 'zipcode': | |
| [98008]}) |  |

predicted\_price = model.predict(new\_house) print("Predicted Price:",

Out put :

Model R^2 Score: 0.5152176902631012

Predicted Price: 121215.61449578404

Conclusion:

In conclusion, using machine learning in Python is a powerful tool for predicting house prices. By gathering and cleaning data, visualizing patterns, and training and evaluating our models, we can make informed decisions in the dynamic world of real estate.

By leveraging advanced algorithms and data analysis, we can make accurate predictions and inform decision-making processes. This approach empowers buyers, sellers, and investors to make informed choices in a dynamic and competitive market, ultimate