

# PHASE 2 SUBMISSION

## Predicting House Price using Machine Learning

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)**

Follow the steps given below to perform the prediction of house prices using machine learning –

We have used Kaggle `kc_house_data` dataset.

Import the required libraries and modules, including pandas for data manipulation, scikit-learn 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.

## Example

```
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:",predicted_price[0])
```

### **Output:**

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, ultimately maximizing their opportunities and outcomes.