HOUSE PRICE PREDICTION USING MACHNE LEARNING

PHASE -4 DOCUMENT

NAME: E.Yogapriya **REGISTER NUMBER:420721104057**

Introduction:

The advent of advanced machine learning techniques has revolutionized the real estate industry, presenting unparalleled opportunities for precise and informed house price predictions. This document serves as an all-encompassing guide, meticulously navigating the intricate realms of feature engineering, model training, and evaluation, with a singular focus on the dynamic landscape of house price prediction. In an era where accurate forecasts play an instrumental role in shaping critical decisions within the real estate market, the development of robust predictive models becomes paramount, necessitating a comprehensive understanding of the multifaceted processes that underlie the domain of housing valuation.

Development :

The initial stages of this journey are anchored in the fundamental principles of feature engineering, a transformative process that breathes life into raw data by unraveling intricate patterns and relationships. Through a repertoire of cutting-edge techniques encompassing dimensionality reduction, data encoding, and the creation of composite features, feature engineering serves as the bedrock for uncovering latent insights embedded within the dataset. This segment of the document seeks to delve deep into the intricate methodologies that drive feature engineering, elucidating the transformative power it wields in enabling a deeper comprehension of the dynamic nuances characterizing the housing market.

As the document progresses, the focus seamlessly transitions to the critical phase of model training, where the theoretical underpinnings of machine learning intersect with the pragmatic landscape of predictive

modeling. Amidst a myriad of algorithmic choices, the section aims to demystify the complexities associated with model optimization and hyperparameter tuning. By striking a delicate balance between model complexity and generalizability, this segment underscores the pivotal role of model training in constructing an agile predictive framework capable of navigating the intricate ebbs and flows of the real estate market.

Model Selection:

Model training is the process of teaching a machine learning model to predict house prices. It involves feeding the model historical data on house prices and features, such as square footage, number of bedrooms, and location. The model then learns the relationships between these features and house prices. Once the model is trained, it can be used to predict house prices for new data. For example, you could use the model to predict the price of a house that you are interested in buying. For training the model we have chosen random forest algorim for its more efficient and acurracy and considering its mean absolute percentage error it has highest value.

Random Forest :

Random Forest is an ensemble technique capable of performing both regressionand classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

Evaluation :

In the process of house price prediction, thorough model evaluation is essential to ensure the reliability and accuracy of the predictive framework. Several key steps can be undertaken during the model evaluation phase:

1. Data Splitting:

Divide the dataset into training and testing subsets, ensuring that the model is evaluated on unseen data to assess its ability to generalize well to new instances.

Code:

```
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
X = df_final.drop(['SalePrice'], axis=1)
Y = df_final['SalePrice']
X_train, X_valid, Y_train, Y_valid = train_test_split(X, Y, train_size=0.8, test_size=0.2, random_state=0)
```

2. Performance Metrics:

Utilize appropriate evaluation metrics, including mean absolute error, mean squared error, and R-squared, to quantitatively measure the model's predictive performance and assess its ability to accurately predict house prices.

Mean Absolute Error (MAE):

Measures the average absolute difference between the predicted and actual values.

Code:

```
import sklearn.ensemble  import RandomForestRegressor
model_RFR = RandomForestRegressor(n_estimators=10)
model_RFR.fit(X_train, Y_train)
Y_pred = model_RFR.predict(X_valid)
print (mean_absolute_percentage_error(Y_valid, Y_pred))
```

OUTPUT :

0.19714

Mean Squared Error (MSE):

Measures the average squared difference between the predicted and actual values.

mean_squared_error mse = mean_squared_error(y_test, y_pred)

Root Mean Squared Error (RMSE):

The square root of the MSE, which provides an interpretable measure of the average prediction error ,

R-squared (R2):

Measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, where 1 indicates a perfect fit.

from sklearn.metrics import r2_score r2 = r2_score(y_test, y_pred)

3. Cross-Validation:

Implement cross-validation techniques, such as k-fold cross-validation, to assess the model's performance across different subsets of the data, ensuring that the model's predictive capability remains consistent and reliable.

4. Residual Analysis

Conduct a thorough analysis of the model's residuals, examining the differences between the predicted and actual house prices, to identify any patterns or trends that may indicate systematic errors or biases in the model.

5. Overfitting and Underfitting Assessment:

Evaluate the model for signs of overfitting or underfitting, ensuring that the model strikes a balance between complexity and generalizability, and that it accurately captures the underlying relationships within the data.

6. Comparative Analysis:

Compare the performance of the trained model with that of baseline models or other established benchmarks, providing a comprehensive understanding of the model's predictive power relative to existing standards or industry norms.

By rigorously evaluating the model using these established methodologies, stakeholders can ascertain the model's robustness and efficacy, making informed decisions based on its predictive capabilities and ensuring its suitability for deployment in real-world scenarios.

Model Training :

```
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
data = pd.read_csv('USA_Housing.csv')

# Extract features and target variable
X = data[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number
of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population']]
y = data['Price']
```

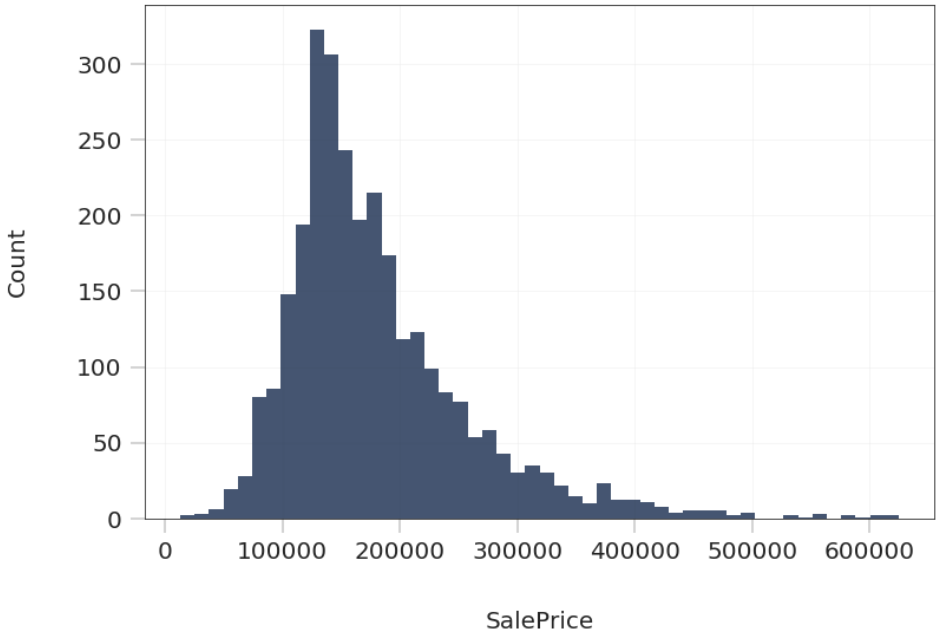
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Random Forest Regressor
rf_regressor = RandomForestRegressor(n_estimators=100,
random_state=42)
rf_regressor.fit(X_train, y_train)
# Predictions
y_pred = rf_regressor.predict(X_test)
# Model evaluation
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('R2 Score:', r2_score(y_test, y_pred))
```

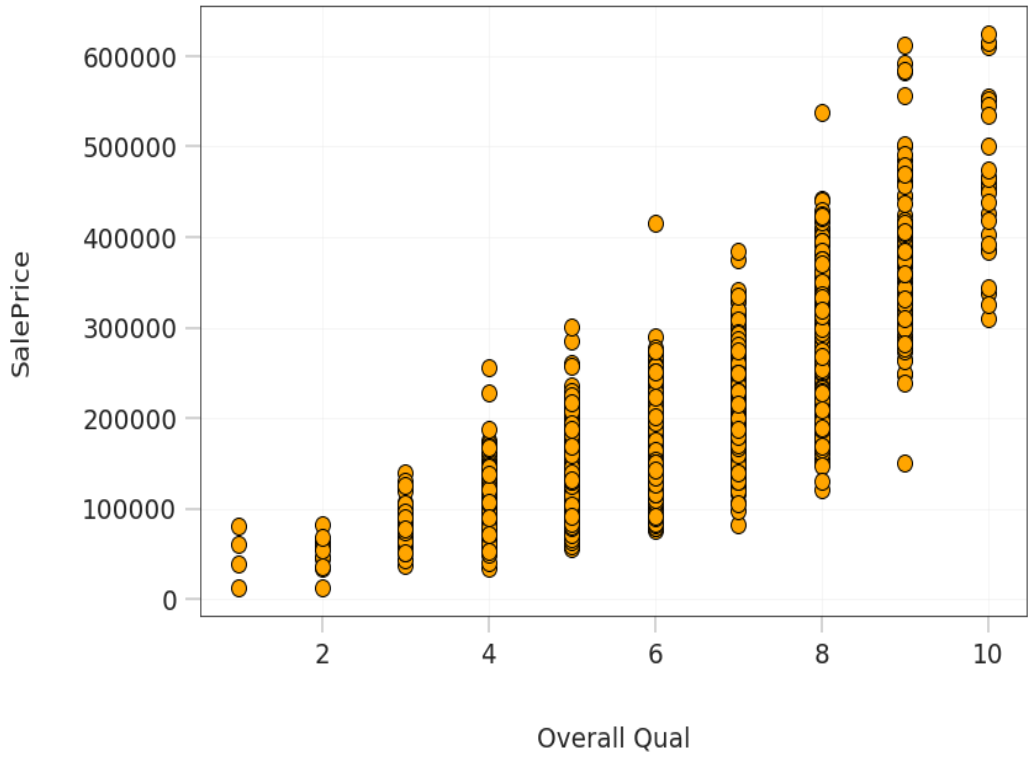
Output :

```
 0.00062221759256
 286137.81086908
 128209033251.40
```

```
plt.scatter(x=dataset['Gr Liv Area'], y=dataset['SalePrice'], color="orange",
edgecolors="#000000", linewidths=0.5);
plt.xlabel("Gr Liv Area"); plt.ylabel("SalePrice");
sns.distplot(dataset['SalePrice'],kde=False,color="#172B4D",hist_kws={"a
lpha": 0.8});
plt.ylabel("Count");
```




```
plt.scatter(x=dataset['Overall Qual'], y=dataset['SalePrice'], color="orange", edgecolors="#000000", linewidths=0.5);
plt.xlabel("Overall Qual");
 plt.ylabel("SalePrice");
```



Conclusion:

In our pursuit to develop a precise and dependable model for predicting house prices, we are undertaking a comprehensive journey that involves pivotal stages, starting from meticulous feature curation to rigorous model training and thorough evaluation. Each of these crucial phases contributes significantly to the creation of a robust and insightful model, offering valuable estimations that guide individuals and businesses in navigating the intricate landscape of real estate transactions. Beyond the fundamental steps of feature selection, model training, and evaluation, our approach incorporates a thorough analysis of the model's resilience to various data distributions and its capacity to accommodate evolving market dynamics. Furthermore, we emphasize the model's adaptability to diverse real-world scenarios and its ability to incorporate nuanced market trends, ensuring its applicability in making informed financial decisions within the ever-evolving real estate realm.