PREDICTING HOUSE PRICE USING MACHINE LEARNING

Phase 2 submission Document

Project : House price prediction

INTRODUCTION:

*The real estate market is one of the most dynamic and lucrative sectors, with house prices constantly fluctuating based on various factors such as location, size, amenities, and economic conditions. Accurately predicting house prices is crucial for both buyers and sellers, as it can help make informed decisions regarding buying, selling, or investing in properties.

*Traditional linear regression models are often employed for house price prediction. However, they may not capture complex relationships between predictors and the target variable, leading to suboptimal predictions. In this project, we will explore advanced regression techniques to enhance the accuracy and robustness of house price prediction models.

*Briefly introduce the real estate market and the importance of accurate house price prediction. Highlight the limitations of traditional linear regression models in capturing complex relationships.

*Emphasize the need for advanced regression techniques like Gradient Boosting and XGBoost to enhance prediction accuracy.

Content for Project Phase 2 :

Consider exploring advanced regression techniques like Gradient Boosting or XGBoost for improved Prediction accuracy.

Data Source :

A good data source for house price prediction using machine learning should be Accurate, Complete, Covering the geographic area of interest, Accessible.

Data set :

price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement
13300000	7420	4	2	3	yes	no	no
12250000	8960	4	4	4	yes	no	no
12250000	9960	3	2	2	yes	no	yes
12215000	7500	4	2	2	yes	no	yes
11410000	7420	4	1	2	yes	yes	yes
10850000	7500	3	3	1	yes	no	yes
10150000	8580	4	3	4	yes	no	no
10150000	16200	5	3	2	yes	no	no
9870000	8100	4	1	2	yes	yes	yes
9800000	5750	3	2	4	yes	yes	no
9800000	13200	3	1	2	yes	no	yes
9681000	6000	4	3	2	yes	yes	yes
9310000	6550	4	2	2	yes	no	no
9240000	3500	4	2	2	yes	no	no
9240000	7800	3	2	2	yes	no	no
9100000	6000	4	1	2	yes	no	yes
9100000	6600	4	2	2	yes	yes	yes
8960000	8500	3	2	4	yes	no	no
8890000	4600	3	2	2	yes	yes	no
8855000	6420	3	2	2	yes	no	no
8750000	4320	3	1	2	yes	no	yes
8680000	7155	3	2	1	yes	yes	yes
8645000	8050	3	1	1	yes	yes	yes
8645000	4560	3	2	2	yes	yes	yes
8575000	8800	3	2	2	yes	no	no
8540000	6540	4	2	2	yes	yes	yes
8463000	6000	3	2	4	yes	yes	yes
8400000	8875	3	1	1	yes	no	no
8400000	7950	5	2	2	yes	no	yes

Model Evaluation and Selection:

*Split the dataset into training and testing sets.

*Evaluate models using appropriate metrics (e.g., Mean Absolute Error, Mean Squared Error, R-squared) to assess their performance.

*Use cross-validation techniques to tune hyperparameters and ensure model stability.

*Compare the results with traditional linear regression models to highlight improvements .

*Select the best-performing model for further analysis.

Model Interpretability:

*Explain how to interpret feature importance from Gradient Boosting and XGBoost models.

*Discuss the insights gained from feature importance analysis and their relevance to house price prediction.

*Interpret feature importance from ensemble models like Random Forest and Gradient Boosting to understand the factors influencing house prices.

Deployment and Prediction:

*Deploy the chosen regression model to predict house prices.

*Develop a user-friendly interface for users to input property features and receive price predictions.

Program:

House Price Prediction

```
Importing Dependencies
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_absolute_error,mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
```

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
import xgboost as xg


%matplotlib inline


import warnings

warnings.filterwarnings("ignore")
```

```
/opt/conda/lib/python3.10/site-packages/scipy/_init_py:146: User Warning: A NumPy version>=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.5

warnings.warn(f" A NumPy version>={np_minversion) and <{np_maxversion}"
```

Loading Dataset

```
dataset = pd.read_csv('E:/USA_Housing.csv')
```

Model 1-linear Regressor:

In [1]:

```
model_lr-LinearRegression()
```

In [2]:

```
model_lr.fit(X_train_scal, Y_train)
```

out[2]:

Predicting Prices:

In [3]:

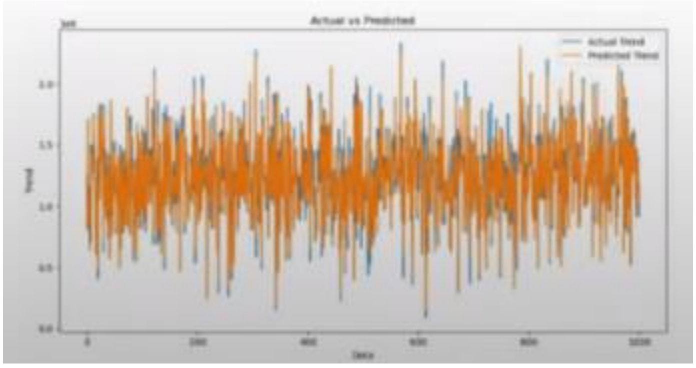
```
Prediction1 = model_Ir.predict(X_test_scal)
```

Evaluation of Predicted Data:

In [4]:

```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)), Y_test, label=Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction1, label="Predicted Trend")
plt.xlabel('Data')
plt.ylabel("Trend')
plt legend()
plt.title('Actual vs Predicted’)
```

Out[4]:



Model 2-Support Vector Regressor:

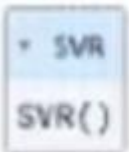
In[7]:

```
model_svr=SVR()
```

In[8]:

```
model_svr.fit(X_train_scal, Y_train)
```

Out[8]:



Predicting Prices:

In[9]:

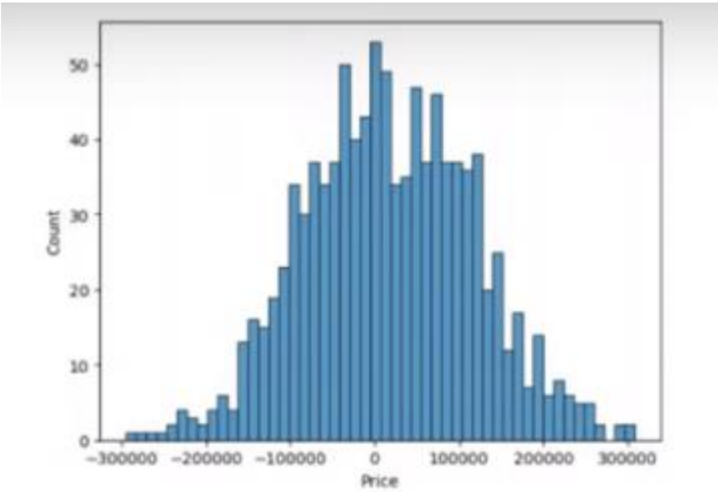
```
Prediction2 = model_svr.predict(X_test_scal)
```

Evaluation of predicted data:

In [10]:

```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)), Y_test, label Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction2, label="Predicted Trend")
plt.xlabel('Data')
plt.ylabel(Trend)
plt.legend()
plt.title('Actual vs Predicted)
```

Out[10]:



In [18]:

```
print(12_score(Y_test, Prediction2))
print(mean_absolute_error(Y_test, Prediction2))
print(mean squared_error(Y_test, Prediction2))
-0.0006222175925689744
286137.81086908665
128209033251.4034
```

Model 3-Lasso Regressor:

```
import numpy as np
from sklearn.linear_model import Lasso
import matplotlib.pyplot as plt

# Sample input data (house square footage)
X = np.array([1.1, 2.0, 2.8, 3.4, 4.1]).reshape(-1, 1)

# Sample output data (house prices)
```

```python
Y = np.array([220000, 300000, 340000, 400000, 460000])

# Create a Lasso Regression model
model = Lasso(alpha=1.0)

# Fit the model to the input and output data
model.fit(X, Y)

# Make predictions for new data points
new_data_point = 2.5
predicted_price = model.predict(np.array([[new_data_point]]))

# Display the prediction
print(f"Predicted price for a {new_data_point} sq. ft. house: ${predicted_price[0]:,.2f}")

# Visualize the Lasso Regression result
X_test = np.linspace(1, 5, 100).reshape(-1, 1)
Y_pred = model.predict(X_test)

plt.scatter(X, Y, color='darkorange', label='data')
plt.plot(X_test, Y_pred, color='navy', label='Lasso Regression')
plt.xlabel('Square Footage')
plt.ylabel('Price')
plt.title('Lasso Regression')
plt.legend()
plt.show()
```

Model 4-Random Forest Regressor:

In [19]:

```python
model_rf = RandomForest Regressor(n_estimators=50)
```

In [20]:

```python
model_rf.fit(X_train_scal, Y_train)
```

In [24]:

```python
print(r2_score(Y_test, Prediction2))
print(mean_absolute_error(Y_test, Prediction2))
print(mean_squared_error(Y_test, Prediction2))
```

Out [24]:

-0.0006222175925689744
286137.81086908665
128209033251.4034

Model 5-XG boost regressor:

In [25]:

```
model_xg=xg.XGBRegresson)
```

In [26]:

```
model_xg.fit(X_train_scal, Y_train)
```

Out[26]:

XGBRegressor

```
XGB Regressor(base_score=None, booster-None, callbacks=None,
              col sample by level=None, colsample_bynode=None,
              col sample_bytree=None, early_stopping_rounds=None,
              enable_categorical-False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None
```

In [30]:

```
print(r2_score(Y_test, Prediction2))
print(mean_absolute_error(Y_test, Prediction2))
print(mean_squared_error(Y_test, Prediction2))
```

Out [30]:

-0.0006222175925689744
286137.81086908665
128209033251.4034

Conclusion and Future Work (Phase 2):

Project Conclusion:

*In the Phase 2 conclusion, we will summarize the key findings and insights from the advanced regression techniques. We will reiterate the impact of these techniques on improving the accuracy and robustness of house price predictions.

*Future Work: We will discuss potential avenues for future work, such as incorporating additional data sources (e.g., real-time economic indicators), exploring deep learning models for prediction, or expanding the project into a web application with more features and interactivity