Name :

Awais Ahmad

Roll-No:

SU92-BSAIM-F23-132

Section:

BSAI-4C

Task:

(1)

Home Price Prediction Model:

1) Import Libraries:

For data manipulation (pandas), model training (sklearn), and evaluation are imported.

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeRegressor
```

2) Load Dataset:

```
df = pd.read_csv("train.csv")
# print("frist 5 rows:")
print (df.head(5))
```

3) **Handle Missing Values:**

```
# Drop columns with any missing values
df = df.dropna(axis=1)
```

4) **Data Overview:**

```
df.describe()
df.isnull().sum()
df.info()
```

5) Define Target Variable:

```
# Define target (dependent variable)
y = df['SalePrice']  # Target column
```

6) **Define Features**:

```
# Define features (independent variables)
X = df.drop(columns=['SalePrice'])  # Drop the target column from dataset
```

7) **Check for Remaining Missing Values**:

```
df.isnull().sum().sum()
df.isnull().sum().to_frame().transpose()
```

8) **Fill Missing Values**:

```
numeric_columns = df.select_dtypes(include=['number']).columns
df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].mean())
```

9) **Feature Scaling**:

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

10) **Train-Test Split**:

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)
```

11) **Initialize and Train the Model**:

```
model = RandomForestRegressor(n_estimators=100, random_state=42
model.fit(X_train, y_train)
```

12) **Make Predictions**:

```
y = df.SalePrice
features = ['LotArea', 'YearBuilt', '1stFlrSF', '2ndFlrSF', 'FullBath',
'BedroomAbvGr', 'TotRmsAbvGrd']
X = df[features]
```

13) **Evaluate the Model**:

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

14) **Print Evaluation Metrics**:

```
print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"R²: {r2}")
```

15) **Prepare Submission File**:

```
output = pd.DataFrame({'Id': .Id,'SalePrice': test_preds})
output.to_csv('submission.csv', index=False)
```