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
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

2) Load Dataset:

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

3) Handle Missing Values:

```
#handling the missing values
df['Age'].fillna(df['Age'].mean(), inplace=True)
df.isnull().sum().sort_values(ascending=False)
```

Data Overview:

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

7) Define Target Variable:

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

8) Define Features:

```
# Define features (independent variables)
df = df.drop(['PassengerId', 'Name'], axis=1)
df.head(5)
  # 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:**

```
    df[['VIP', 'CryoSleep', 'FoodCourt', 'ShoppingMall', 'Spa',
'VRDeck']] = df[['VIP', 'CryoSleep', 'FoodCourt', 'ShoppingMall',
'Spa', 'VRDeck']].fillna(value=0)
    df.isnull().sum().sort_values(ascending=False)
```

11) **Feature Scaling:**

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

12) **Train-Test Split**:

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

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

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

14) **Make Predictions**:

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

13) **Evaluate the Model**:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error

# Assuming df is already loaded
X = df.drop(columns=['Transported'])  # Features
y = df['Transported']  # Target

# Handle missing values in X
# Fill numerical columns with mean
numerical_cols =
X.select_dtypes(include=['float64']).columns
X[numerical_cols] =
X[numerical_cols].fillna(X[numerical_cols].mean())

```

```
    # Fill categorical columns with "Unknown"
    categorical_cols = X.select_dtypes(include=['object']).columns
    X[categorical_cols] = X[categorical_cols].fillna("Unknown")

    # One-hot encode categorical columns
    X = pd.get_dummies(X, drop_first=True)

    # Convert target variable to numeric if needed
    y = y.astype(int)

    # Train-test split
    X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.2, random_state=42)

    # Train Random Forest model
    rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
    rf_model.fit(X_train, y_train)

    # Make predictions
    y_pred = rf_model.predict(X_valid)

    # Evaluate the model
    mae = mean_absolute_error(y_valid, y_pred)
    print("Mean Absolute Error:", mae)

```

52) **Print Evaluation Metrics**:

```
    from sklearn.metrics import accuracy_score, mean_absolute_error, r2_score

    y_pred_binary = (y_pred > 0.5).astype(int)

    accuracy = accuracy_score(y_valid, y_pred_binary)
    print("Classification Accuracy:", accuracy)

    mae = mean_absolute_error(y_valid, y_pred)
```

```
print("Mean Absolute Error (MAE):", mae)

r2 = r2_score(y_valid, y_pred)
print("R² Score:", r2)
```

65) **Prepare Submission File**:

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