Ex2 : Python operations on real time application

Problem identified in real time

Description:
In a real-time scenario, the problem revolves around predicting real estate prices based on various property features. The goal is to develop a predictive model that can assist potential buyers, sellers, and real estate agents in estimating property values accurately. This model leverages machine learning, particularly linear regression, to make these predictions.

Scope:
The scope of this project includes:
- Collecting a dataset containing real estate property information and corresponding prices.
- Preprocessing the dataset to handle missing values, outliers, and feature scaling.
- Building a linear regression model to predict property prices based on features like house age, distance to amenities, and more.
- Evaluating the model's performance using metrics such as R2 score, Mean Squared Error (MSE), and Mean Absolute Error (MAE).
- Visualizing relationships between different features and property prices.
- Potentially deploying the model to provide real-time price predictions through a web interface or mobile app.

Python Packages Used:
- pandas: Used for data manipulation and analysis, reading CSV files, and managing dataframes.
- numpy: Used for numerical operations and calculations.
- seaborn and matplotlib: Used for data visualization and creating scatter plots to analyze relationships between features and prices.
- scikit-learn: Used for splitting the dataset, training the linear regression model, making predictions, and evaluating the model's performance.

Code:

```
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.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn import preprocessing

df = pd.read_csv('/content/Real estate.csv')
df.head()

df.shape

df.drop('No',inplace= True, axis= 1)

df.shape

df.columns.values

#Step 3: Create a scatterplot to visualize the data
sns.scatterplot(x='X4 number of convenience stores',
               y='Y house price of unit area', data= df)

# Step 3b addition----univariate analysis vs Price of Unit Area---->
sns.scatterplot(x = 'X3 distance to the nearest MRT station',
              y='Y house price of unit area', data = df)

# 'X2 house age'
sns.scatterplot(x = 'X2 house age', y='Y house price of unit area',data = df)

#creating a feature variables
X= df.drop('Y house price of unit area',axis= 1)
y= df['Y house price of unit area']
X

y

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_state= 3)

lr = LinearRegression()
lr.fit(X_train, y_train)
predictions = lr.predict(X_test)

print("R2 score", r2_score(y_test,predictions))
```