Predicting house price in machine learning

AI

Program Dataset:

```
import pandas as pd
import numpy
import random
random.seed(4)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import Lasso, Ridge
from sklearn.model_selection import cross_val_score
from sklearn.metrics import mean_squared_error, r2_score
df = pd.read_csv("/content/drive/MyDrive/Housing.csv")
df.head()
df.info()
df.describe()
df.isnull().sum()
plt.figure
sns.histplot(df['price'], kde=True)
plt.title('House Price Distribution')
plt.xlabel('Price')
plt.show()
sns.barplot(x=df['airconditioning'],y=df['bedrooms'],hue=df["furnishingstatus"])
sns.scatterplot(y=df['price'],x=df['area'],hue=df['furnishingstatus'])
sns.boxplot(x = 'furnishingstatus', y = 'price', hue = 'airconditioning', data = df)
fig, axs = plt.subplots(2,3, figsize = (10,5))
plt1 = sns.boxplot(df['price'], ax = axs[0,0])
plt2 = sns.boxplot(df['area'], ax = axs[0,1])
plt3 = sns.boxplot(df['bedrooms'], ax = axs[0,2])
plt1 = sns.boxplot(df['bathrooms'], ax = axs[1,0])
plt2 = sns.boxplot(df['stories'], ax = axs[1,1])
plt3 = sns.boxplot(df['parking'], ax = axs[1,2])

plt.tight_layout()
sns.pairplot(df)
```

```
plt.show()
X = df.drop(['price'],axis=1)
y = df['price']
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=42)
sns.heatmap(df.corr(numeric_only=True), annot = True)
plt.show()
X_train.head(), y_train.head()
numeric_preprocessor = Pipeline(
    steps=[
        ("imputation_mean", SimpleImputer(missing_values=numpy.nan, strategy="mean")),
        ("scaler", StandardScaler()),
    ]
)
categorical_preprocessor = Pipeline(
    steps=[
        (
            "imputation_most_frequent",
            SimpleImputer(fill_value="missing", strategy="most_frequent"),
        ),
        ("onehot", OneHotEncoder(handle_unknown="ignore", drop='first')),
    ]
)

categorical_col = ['mainroad', 'guestroom', 'basement', 'prefarea', 'furnishingstatus']
numerical_col = ['price', 'bedrooms', 'bathrooms', 'bedrooms', 'stories']
preprocessor = ColumnTransformer(
    [
        ("categorical", categorical_preprocessor, categorical_col),
        ("numerical", numeric_preprocessor, numerical_col),
    ]
)
preprocessor
X_train.head(), y_train.head()
```