Develop a linear regression model to predict house price based on features such as the number of rooms, location, size and other relevant factors. Collect a suitable dataset from Kaggle, preprocess it, and train the model to make accurate predictions.

Project Title: House Price Prediction Using Linear Regression

1. Introduction

This project aims to develop a linear regression model that predicts house prices based on various property features such as number of rooms, location, size, and quality. Accurate price prediction can assist buyers, sellers, and real estate professionals in making better-informed decisions.

2. Dataset Description

The dataset used is the Kaggle "House Prices - Advanced Regression Techniques" dataset, consisting of 1460 samples and 81 features. The data includes both numerical attributes (e.g., year built, lot area, total rooms) and categorical attributes (e.g., neighborhood, exterior quality). The target variable is SalePrice.

3. Data Preprocessing and Feature Engineering

- **Missing values were filled (numerical features with median, categorical features with "None").**
- **The target variable SalePrice was log-transformed to stabilize variance.**
- **New features such as total living area and total number of bathrooms were created.**
- **Ordinal categorical variables like ExterQual and KitchenQual were encoded into ordered numerical values.**
- **Nominal categorical variables were one-hot encoded.**

4. Model Development

A pipeline was constructed combining:

- **Data preprocessing (imputation, ordinal encoding, scaling, one-hot encoding).**
- **A linear regression model was trained on the processed data.**
  Data was split into training and validation sets for evaluation.

5. Model Evaluation

The model's performance was evaluated with:

- **Root Mean Squared Error (RMSE) on the log-transformed target.**
- **R^2 score indicating the proportion of variance explained.**
  The final model achieved an RMSE of approximately 0.13 (log scale) and R^2 of 0.86 on the validation set.

6. Future Work

Potential improvements include:

- **Trying regularization approaches like Ridge or Lasso regression.**
- **Implementing ensemble methods e.g., Random Forest or Gradient Boosting.**
- **More advanced feature engineering and hyperparameter tuning.**

7. Conclusion

The project successfully demonstrates building an effective linear regression model for house price prediction using real-world dataset and thorough preprocessing, achieving good prediction accuracy.

Code :

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error, r2_score

from sklearn.preprocessing import OneHotEncoder, StandardScaler

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline


# Load the dataset (assuming CSV format from Kaggle)
```

```
data = pd.read_csv('house_prices.csv')


print(data.head())

print(data.info())

print(data.describe())


for col in data.select_dtypes(include=[np.number]).columns:

    data[col].fillna(data[col].median(), inplace=True)

for col in data.select_dtypes(include=['object']).columns:

    data[col].fillna(data[col].mode()[0], inplace=True)


data['TotalArea'] = data['GrLivArea'] + data['TotalBsmtSF']


num_features = ['OverallQual', 'TotalArea', 'YearBuilt', 'FullBath', 'GarageCars',
'GarageArea', 'TotRmsAbvGrd']

cat_features = ['Neighborhood', 'ExterQual', 'KitchenQual']


X = data[num_features + cat_features]

y = data['SalePrice']


numeric_transformer = StandardScaler()


categorical_transformer = OneHotEncoder(handle_unknown='ignore')


preprocessor = ColumnTransformer(
```

```
    transformers=[
        ('num', numeric_transformer, num_features),
        ('cat', categorical_transformer, cat_features)
    ])

model = Pipeline(steps=[('preprocessor', preprocessor),
                        ('regressor', LinearRegression())])

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

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"Model Performance:")
print(f"RMSE: {rmse:.2f}")
print(f"R^2: {r2:.2f}")

plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, alpha=0.7)
```

```
plt.xlabel("Actual Prices")

plt.ylabel("Predicted Prices")

plt.title("Actual vs Predicted House Prices")

plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')

plt.show()
```

References

- **Kaggle House Prices - Advanced Regression Techniques Dataset**
- **Scikit-learn documentation on preprocessing and modeling**