Jhon Carlo F. Ledesma
BSCS-2A

CSST 102
Mr. Barnardino

Report: Predicting House Prices with Multiple Regression

1. Introduction

- In this machine learning problem, we aim to predict house prices based on features such as size, number of bedrooms, age, and proximity to downtown using multiple linear regression. This task involves several steps, including data preprocessing, exploratory data analysis (EDA), model development, and evaluation.

2. Process Overview

2.1 Data Exploration and Visualization

- **Loading the Data** - The dataset was imported using pandas to examine the general structure, including missing values and basic statistical summaries.

```
df = pd.read_csv('datasets_house_prices.csv')
print(df.info())
print(df.describe())
print(df.isnull().sum())
```

- **EDA** - We first explored the dataset to better understand the relationships between the various features (size, age, proximity to downtown, etc.) and house prices. The correlation matrix revealed important relationships, helping us determine which variables had stronger associations with price.
- **Visualizing the Data** - A set of plots, including line plots, histograms, and bar plots, provided insights into how each feature is related to house price. The KDE plot of the price distribution showed its skewness, which helped us understand the nature of the target variable.

```
plt.figure(figsize=(12, 8))
sns.lineplot(data=df, x='Size (sqft)', y='Price', marker='o', color='lightcoral')
plt.subplot(2, 2, 2)
sns.barplot(data=df, x='Bedrooms', y='Price', ci=None, color='lightgreen')
```

```
plt.tight_layout()

plt.show()
```

2.2 Data Preprocessing

- **Handling Missing Values** - We filled missing values with the mean of each respective column, a simple yet effective imputation method when only a small percentage of data is missing.

```
df.fillna(df.mean(), inplace=True)
```

- **Feature Scaling** - To ensure all features had equal influence on the model, we normalized the continuous variables using the StandardScaler from sklearn.

```
scaler = StandardScaler()

X = scaler.fit_transform(df[['Size (sqft)', 'Bedrooms', 'Age', 'Proximity to Downtown (miles)']])

y = df['Price']
```

- **Encoding Categorical Variables** - Although the dataset in this task did not contain categorical features, encoding would be necessary if we encountered variables such as house type or neighborhood categories.

2.3 Model Development

- **Modeling with Linear Regression** - After splitting the dataset into training and testing sets (30% for testing), we trained a multiple linear regression model.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

model = LinearRegression()

model.fit(X_train, y_train)
```

- **Coefficients Interpretation** - The model's coefficients showed the impact of each feature on the house price. For instance, "Size" had the largest positive influence, meaning larger houses were associated with higher prices.

coefficients = pd.DataFrame(model.coef_, ['Size (sqft)', 'Bedrooms', 'Age', 'Proximity to Downtown (miles)'], columns=['Coefficient'])

2.4 Model Evaluation

- **Metrics** - To assess model performance, we used the Mean Squared Error (MSE) and R-squared (R^2) score. The MSE provided an absolute measure of the prediction error, while the R^2 score indicated how well the features explained variance in house prices.

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)

- **Plotting Actual vs. Predicted Prices** - A scatter plot visualizing the actual vs. predicted prices helped us visually inspect how well the model was performing. The diagonal line represents perfect predictions.

plt.scatter(y_test, y_pred, marker='o', color='plum')

plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='black')

plt.show()

3. Challenges Faced

- **Multicollinearity** - Initially, multicollinearity among features (such as size and number of bedrooms) was suspected. We addressed this by examining the correlation matrix and ensuring that features with high correlations were carefully selected or transformed.
- **Outliers** - Extreme house prices affected the model's predictions. We applied robust scaling techniques and inspected residuals to manage these outliers better.
- **Skewness of Price Distribution** - The target variable (house price) was skewed, which could reduce model accuracy. We considered applying log transformations to handle this but found that normalizing the features improved the results without needing transformations.

4. Conclusion

- **Real-World Applicability** - The model showed reasonable performance in predicting house prices and provided valuable insights into the factors affecting house pricing. In practice, this model could be useful for real estate agents, buyers, and sellers for price estimation based on known house features.
- **Limitations** - One limitation is that the model assumes linear relationships between features and price, which may not hold for all factors. Also, external factors like market trends or interest rates are not accounted for, which limits the model's accuracy in volatile conditions.
- **Future Improvements** - Enhancing the model with non-linear algorithms (e.g., decision trees, random forests) could improve accuracy. Adding more features, such as neighborhood quality or historical price trends, would also make predictions more robust.