Report on Predicting House Prices with Multiple Regression

Introduction

This report outlines the process of developing a predictive model to estimate house prices based on various features, including size, number of bedrooms, age, and proximity to downtown. The goal is to assist real estate agents in making more accurate price estimations. The dataset used for this analysis includes several properties with their respective attributes and prices.

Data Exploration and Visualization

Exploratory Data Analysis (EDA)

The dataset consists of the following columns:

- **Size (sq. ft.)**: Size of the house in square feet.
- **Bedrooms**: Number of bedrooms in the house.
- **Age**: Age of the house in years.
- **Proximity to Downtown (miles)**: Distance of the house from the downtown area.
- **Price**: Actual price of the house (in thousands of dollars).

The initial step involved loading the dataset and performing descriptive statistics to understand the distribution of the data. The following visualizations were created to explore the relationships between features and house prices:

1. **Scatter Plots**: Plots were created for each feature against the price to visually assess correlations.
2. **Correlation Matrix**: A heatmap was generated to quantify the relationships between features.

Findings from EDA:

- **Size (sq. ft.)**: There is a strong positive correlation with price, indicating that larger houses tend to be more expensive.
- **Bedrooms**: The number of bedrooms also shows a positive correlation with price, though the relationship is less pronounced than size.
- **Age**: Older houses generally have lower prices, suggesting a negative correlation.
- **Proximity to Downtown**: Houses closer to downtown tend to have higher prices.

Data Preprocessing

Handling Missing Data

The dataset was checked for missing values, and none were found. Therefore, no imputation was necessary.

Normalization

To ensure that all features were on a similar scale, the data was standardized using the `StandardScaler` from Scikit-learn:

python

```
from sklearn.preprocessing import StandardScaler

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

Encoding Categorical Variables

In this dataset, all features are numerical, so no encoding was required.

Model Development

Implementing Multiple Regression

The `LinearRegression` class from Scikit-learn was used to implement the multiple regression model. The dataset was split into training (70%) and testing (30%) sets:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

X = data[['Size (sqft)', 'Bedrooms', 'Age', 'Proximity to Downtown (miles)']]
y = data['Price']

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

model = LinearRegression()
model.fit(X_train, y_train)
```

Feature Selection

The coefficients of the trained model were examined to identify the most significant predictors:

```
coef = model.coef_
feature_importances = pd.Series(coef, index=X.columns)
feature_importances.sort_values(ascending=False)
```

Model Evaluation

Performance Metrics

The model's performance was evaluated using Mean Squared Error (MSE), R-squared, and Adjusted R-squared:

```
from sklearn.metrics import mean_squared_error
import statsmodels.api as sm

y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = model.score(X_test, y_test)

X_train_sm = sm.add_constant(X_train)
model_sm = sm.OLS(y_train, X_train_sm).fit()
adjusted_r2 = model_sm.rsquared_adj
```

Interpretation of Model Coefficients

The summary of the OLS model provided insights into the significance of each predictor. The coefficients indicated the expected change in house price for a one-unit increase in each feature, holding other variables constant.

Visualization of Model Accuracy

A scatter plot was created to visualize the predicted prices against the actual prices:

```
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], 'r--')
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Actual vs Predicted House Prices')
plt.show()
```

Challenges Faced

1. **Data Quality**: Although the dataset was clean, in real-world scenarios, missing values and outliers are common. To address this, techniques such as imputation or outlier detection and removal would be necessary.
2. **Feature Selection**: Identifying the most significant predictors can be challenging, especially in larger datasets. Using statistical methods and model evaluation metrics helped streamline this process.
3. **Model Assumptions**: The linear regression model assumes a linear relationship between features and the target variable. If the relationship is non-linear, alternative modeling techniques may need to be considered.

Conclusion

The multiple regression model developed in this project demonstrates the ability to predict house prices based on various features effectively. The analysis showed that size, number of bedrooms, age, and proximity to downtown are significant predictors of house prices.

Applicability in Real-World Scenarios: This model can assist real estate agents and investors in making informed decisions about property pricing. By understanding the factors that influence house prices, stakeholders can better evaluate market conditions and property values.

Potential Limitations: While the model performed well, it is essential to recognize that real estate markets can be influenced by external factors such as economic conditions, interest rates, and local market trends. Additionally, the model's performance may vary with different datasets or geographical locations.

Overall, this project illustrates the practical application of multiple regression techniques in the real estate domain, providing a foundation for further exploration and refinement of predictive modeling approaches.