Report on Predicting House Prices Using Multiple Regression

This report details the methodology for creating a predictive model aimed at estimating house prices based on various characteristics, such as size, number of bedrooms, age, and distance from downtown. The intention is to provide real estate professionals with a tool for more precise property valuation. The analysis employs a dataset containing multiple properties along with their respective attributes and sale prices.

Data Exploration and Visualization

Exploratory Data Analysis (EDA)

The dataset comprises the following variables:

- **Size (sq. ft.)**: The total area of the house measured in square feet.
- **Bedrooms**: The count of bedrooms in the property.
- **Age**: The age of the house expressed in years.
- **Proximity to Downtown (miles)**: The distance from the house to the downtown area.
- **Price**: The selling price of the house, represented in thousands of dollars.

The first step involved importing the dataset and conducting descriptive statistics to grasp the data distribution. Various visualizations were created to examine the relationships between features and house prices:

1. **Scatter Plots**: These were generated for each feature against the price to visually evaluate correlations.
2. **Correlation Matrix**: A heatmap was produced to quantify the relationships among features.

Findings from EDA

- **Size (sq. ft.)**: A strong positive correlation with price was observed, indicating that larger homes typically command higher prices.
- **Bedrooms**: The number of bedrooms also exhibited a positive correlation with price, albeit less pronounced than that of size.
- **Age**: An inverse relationship was noted, with older houses generally priced lower.
- **Proximity to Downtown**: Properties situated closer to downtown areas tended to have higher prices.

Data Preprocessing

Handling Missing Data

The dataset was scrutinized for any missing values, and none were detected, eliminating the need for imputation.

Normalization

To ensure uniformity across feature scales, the data was standardized using the StandardScaler from Scikit-learn:

```
from sklearn.preprocessing import StandardScaler

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

Encoding Categorical Variables

Since all features in the dataset are numerical, no encoding was necessary.
Model Development

Implementing Multiple Regression

The LinearRegression class from Scikit-learn was utilized to implement the multiple regression model. The dataset was divided into training (70%) and testing (30%) subsets:

```
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, y, test_size=0.3,
random_state=42)

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

Feature Selection

The model's coefficients were analyzed to determine the most impactful 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 effectiveness was assessed 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)
X_test_sm = sm.add_constant(X_test)
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 valuable insights into the significance of each predictor. The coefficients indicated the expected change in house price for a one-unit increase in each feature while holding other variables constant.

Visualization of Model Accuracy

A scatter plot was created to compare predicted prices against 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**: While the dataset was clean, real-world datasets often contain missing values and outliers. Techniques such as imputation or outlier detection are essential in such cases.
2. **Feature Selection**: Determining the most significant predictors can be complex, especially in larger datasets. Statistical methods and evaluation metrics assisted in this process.
3. **Model Assumptions**: The linear regression model presumes a linear relationship between features and the target variable. If this assumption does not hold, alternative modeling techniques may be required.

Conclusion

The multiple regression model developed in this analysis effectively predicts house prices based on various features. The findings indicated that size, number of bedrooms, age, and proximity to downtown are significant predictors of house prices.

Applicability in Real-World Scenarios

This model can serve as a valuable tool for real estate agents and investors, aiding them in making informed decisions regarding property pricing. By understanding the factors influencing house prices, stakeholders can better assess market conditions and property values.

Potential Limitations

Although the model demonstrated strong performance, it is crucial to recognize that real estate markets are influenced by external factors such as economic conditions, interest rates, and local market trends. Additionally, the model's accuracy may vary across different datasets or geographical areas.

In summary, this project highlights the practical application of multiple regression techniques in the real estate sector, laying the groundwork for further exploration and enhancement of predictive modeling approaches.