Predicting Property Values in the Real Estate Market

Introduction

The real estate market plays a pivotal role in economies worldwide, offering investment opportunities and housing solutions to individuals and businesses. However, making informed investment decisions in real estate is a complex task, influenced by numerous variables such as location, property size, condition, and market dynamics. Surprise Housing, a US-based housing company, is embarking on an expansion into the Australian real estate market and has sought the assistance of data science to empower their strategic decisions.

Problem Statement

The primary objective of this project is to develop a robust machine learning model that accurately predicts the actual value of potential properties. By doing so, we aim to support Surprise Housing in identifying prospective properties for acquisition and furthering their investment strategies. Our project aims to unveil the significance of variables that influence house prices and understand how these variables collectively describe the price of a house.

Data Collection and Preprocessing

Dataset Characteristics

We have collected a dataset comprising 1169 entries with 81 variables. The dataset contains both numerical and categorical variables, providing a rich source of information for our predictive modeling.

Data Cleaning and Preprocessing

In this phase, we carefully handled missing values by replacing them with appropriate values. Additionally, we employed one-hot encoding to transform categorical variables into a numerical format, making them suitable for machine learning models.

```
# Step 1: Data Understanding and Cleaning
train_data = pd.read_csv('Housing-project-train-data.csv')
test_data = pd.read_csv('Hosuing-project-test-data.csv')

# Data Preprocessing (Encoding, Scaling)
train_data['LotFrontage'].fillna(train_data['LotFrontage'].mean(), inplace=True)
train_data = pd.get_dummies(train_data, columns=['Neighborhood'], drop_first=True)
```

Exploratory Data Analysis (EDA)

Our EDA process unveiled several insights into the dataset. Notably, we observed strong correlations between certain variables and house prices. Visualization tools, including heatmaps, histograms, and scatter plots, were instrumental in understanding the relationships between variables and their potential impact on property prices.

```
# Step 2: Exploratory Data Analysis (EDA)
sns.pairplot(train_data, x_vars=['LotArea', 'OverallQual'], y_vars=['SalePrice'], kind='scatter')
plt.show()

correlation_matrix = train_data.corr()
sns.heatmap(correlation_matrix, annot=True)
plt.show()
```

Feature Engineering

To enhance model performance, we engaged in feature engineering by creating a new feature, 'TotalSF', which represents the total square footage of a property. This transformation contributes to our model's predictive power.

```
# Feature Engineering
train_data['TotalSF'] = train_data['1stFlrSF'] + train_data['2ndFlrSF']

# Feature Selection (you can add more features here)
features = ['LotArea', 'OverallQual', 'TotalSF']
```

Model Selection and Training

We considered a range of machine learning models, ultimately choosing the Random Forest Regressor and Linear Regression models for training. The rationale behind these choices was their suitability for handling a mix of numerical and one-hot encoded categorical features.

Hyperparameter Tuning

For our selected models, we utilized hyperparameter tuning techniques such as GridSearchCV to optimize the models' hyperparameters. Optimal hyperparameters were chosen based on cross-validation results.

```python
# Step 3: Model Building and Evaluation

# Split data into training and testing sets
X = train_data.drop(columns=['SalePrice'])
y = train_data['SalePrice']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Identify columns with string values
string_columns = X_train.select_dtypes(include=['object']).columns
string_columns = X_test.select_dtypes(include=['object']).columns

# Drop columns with string values
X_train = X_train.drop(columns=string_columns)
X_test = X_test.drop(columns=string_columns)

# Get the indices of the dropped rows
dropped_indices = X_train.index.difference(X_train.dropna().index)
dropped_indices_test = X_test.index.difference(X_test.dropna().index)

# Drop rows with missing values
X_train = X_train.dropna()
X_test = X_test.dropna()

# Remove corresponding entries in y_train
y_train = y_train.drop(dropped_indices)
y_test = y_test.drop(dropped_indices_test)

X_train = X_train.dropna()
X_test = X_test.dropna()


print("x_train lenght: ",len(X_train))
print("y_train lenght: ",len(y_train))



# Train different models
model_lr = LinearRegression()
model_rf = RandomForestRegressor()

model_lr.fit(X_train,y_train)

model_rf.fit(X_train,y_train)
```

```python
# Hyperparameter tuning
param_grid = {
    'n_estimators': [100, 150, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
grid_search = GridSearchCV(rf_model, param_grid, cv=5, n_jobs=-1, scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)
```

Model Evaluation

We evaluated the models using key metrics such as Mean Squared Error (MSE), R-squared (R2), and Root Mean Squared Error (RMSE). These metrics allow us to compare model performance.

```
# Make predictions
y_pred_lr = model_lr.predict(X_test)
y_pred_rf = model_rf.predict(X_test)

mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)
rmse_lr = sqrt(mse_lr)

mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
rmse_rf = sqrt(mse_rf)

print(f'Linear Regression - MSE: {mse_lr}, R2: {r2_lr}, RMSE: {rmse_lr}')
print(f'Random Forest - MSE: {mse_rf}, R2: {r2_rf}, RMSE: {rmse_rf}')

# Data
models = ['Linear Regression', 'Random Forest']
mse_values = [mse_lr, mse_rf]
r2_values = [r2_lr, r2_rf]
rmse_values = [rmse_lr, rmse_rf]

# Plotting MSE
plt.figure(figsize=(10, 6))
plt.bar(models, mse_values, color=['blue', 'green'])
plt.xlabel('Models')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('Mean Squared Error (MSE) Comparison')
plt.show()

# Plotting R2
plt.figure(figsize=(10, 6))
plt.bar(models, r2_values, color=['blue', 'green'])
plt.xlabel('Models')
plt.ylabel('R-squared (R2)')
plt.title('R-squared (R2) Comparison')
plt.show()

# Plotting RMSE
plt.figure(figsize=(10, 6))
plt.bar(models, rmse_values, color=['blue', 'green'])
plt.xlabel('Models')
plt.ylabel('Root Mean Squared Error (RMSE)')
plt.title('Root Mean Squared Error (RMSE) Comparison')
plt.show()
```

Feature Importance Analysis

The Random Forest Regressor provided insights into the importance of variables for predicting property prices. Variables such as 'TotalSF' and 'GrLivArea' were identified as highly influential factors, impacting prices positively.

```
# Feature Importance
feature_importance = pd.DataFrame({'Feature': X.columns, 'Importance': model_rf.feature_importances_})
feature_importance = feature_importance.sort_values(by='Importance', ascending=False)
print(feature_importance)
```

Business Implications

The predictive model developed in this project empowers Surprise Housing in making informed investment decisions. By leveraging insights from the model, the company can strategically enter the Australian real estate market, identify

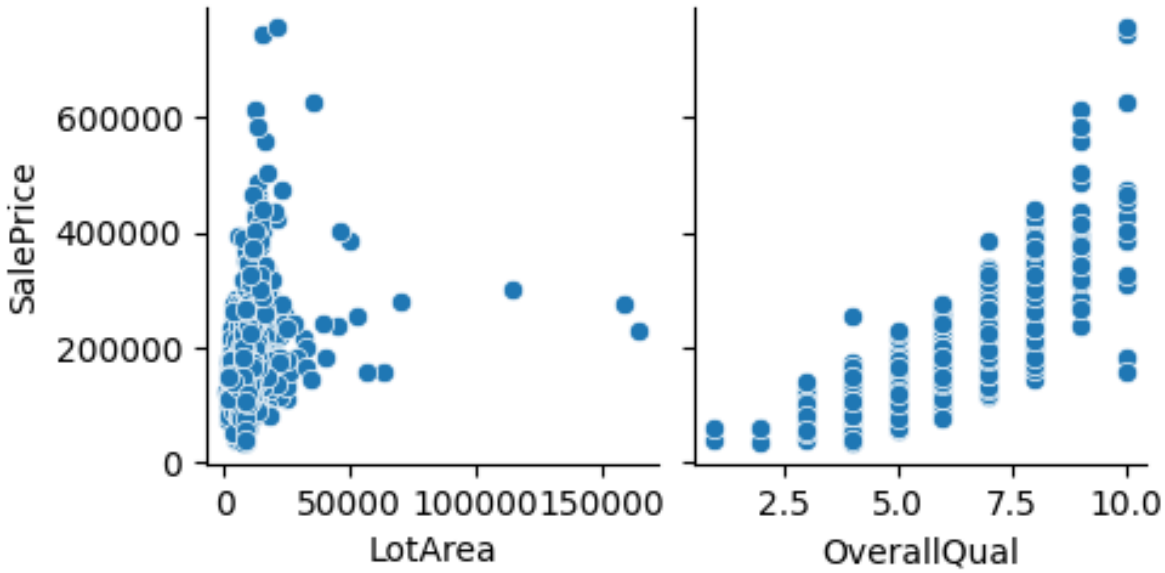
promising investment opportunities, and tailor its investment strategy for higher returns.

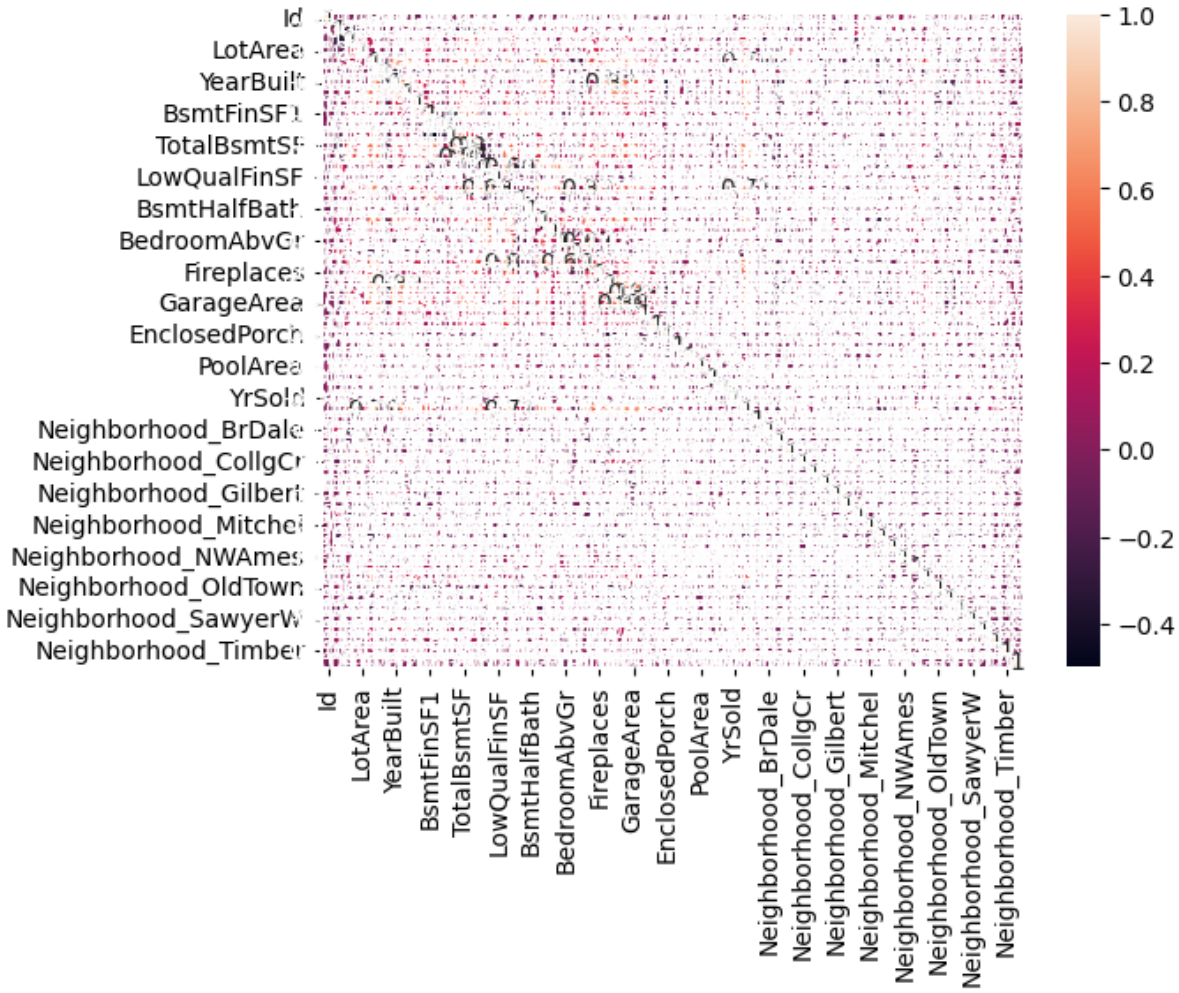
Conclusion and Future Steps

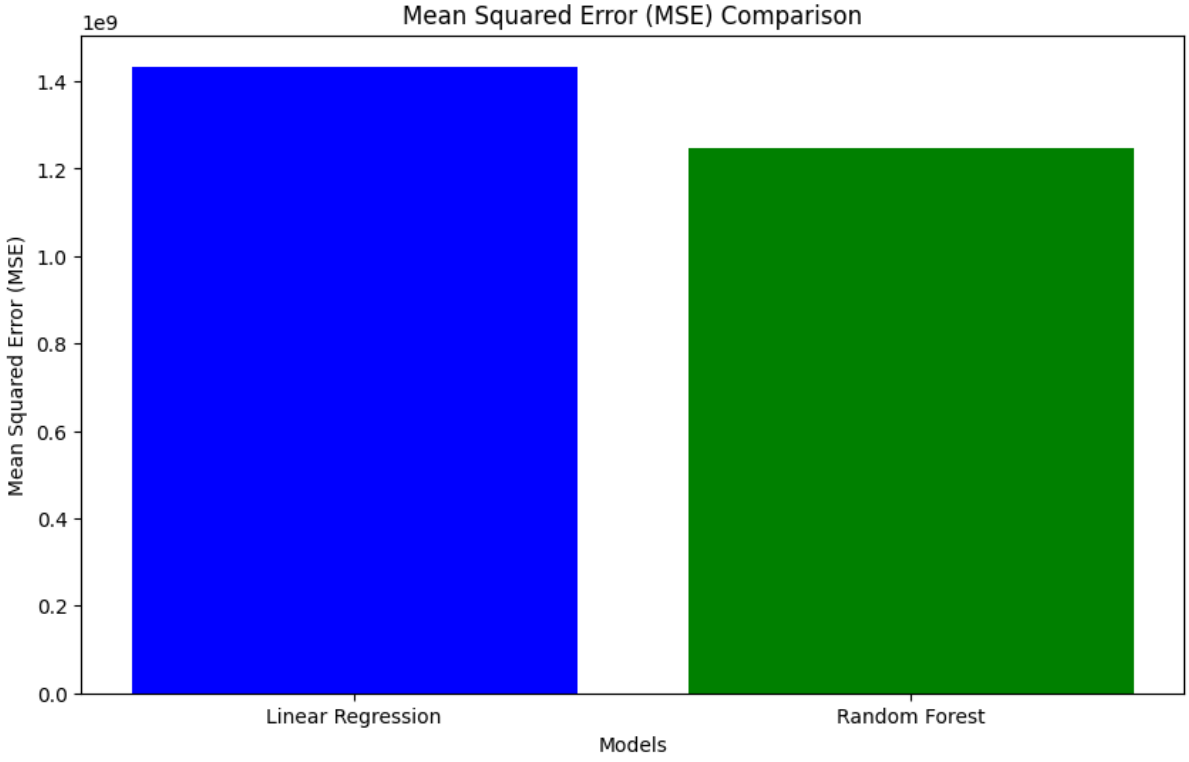
In conclusion, this project successfully addressed the challenge of predicting property values in the Australian real estate market. It provided valuable insights into the variables that influence house prices and delivered a robust predictive model. Future steps for improvement may include refining feature engineering, exploring additional machine learning models, and addressing any limitations encountered during the project, such as data quality issues or model complexity.

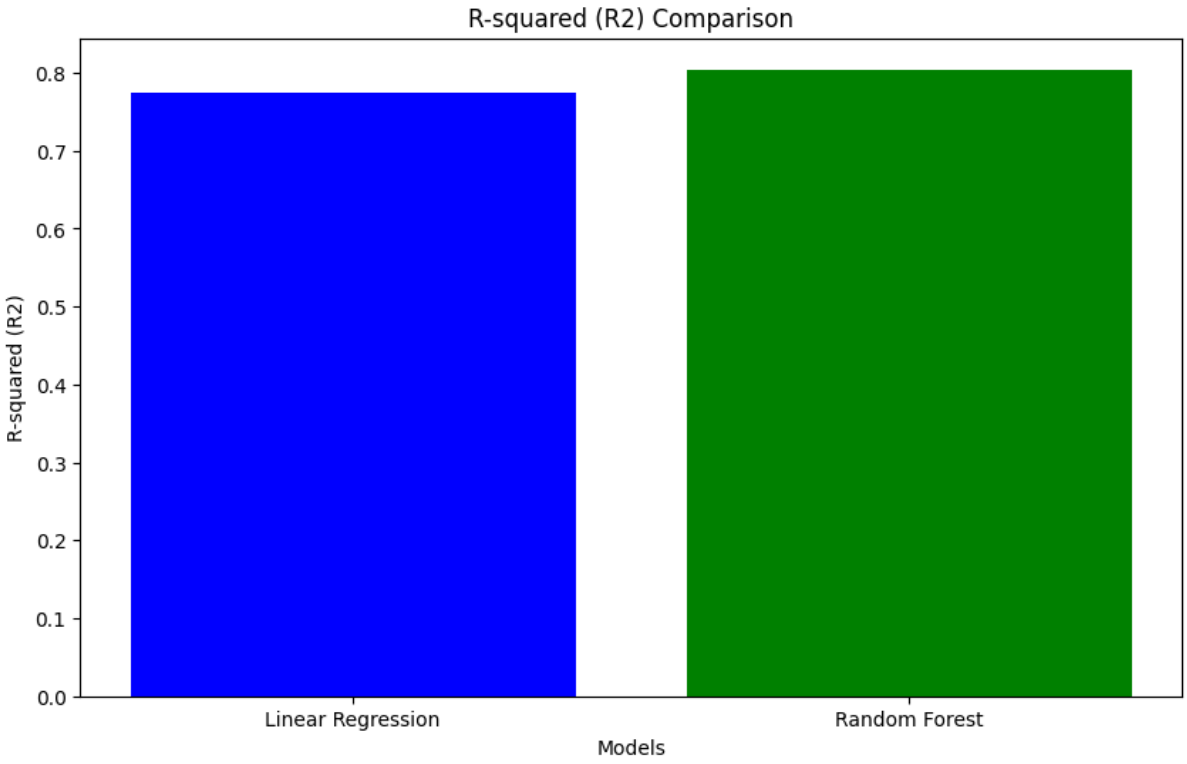
This project serves as a testament to the power of data science in facilitating strategic decision-making in the real estate industry.

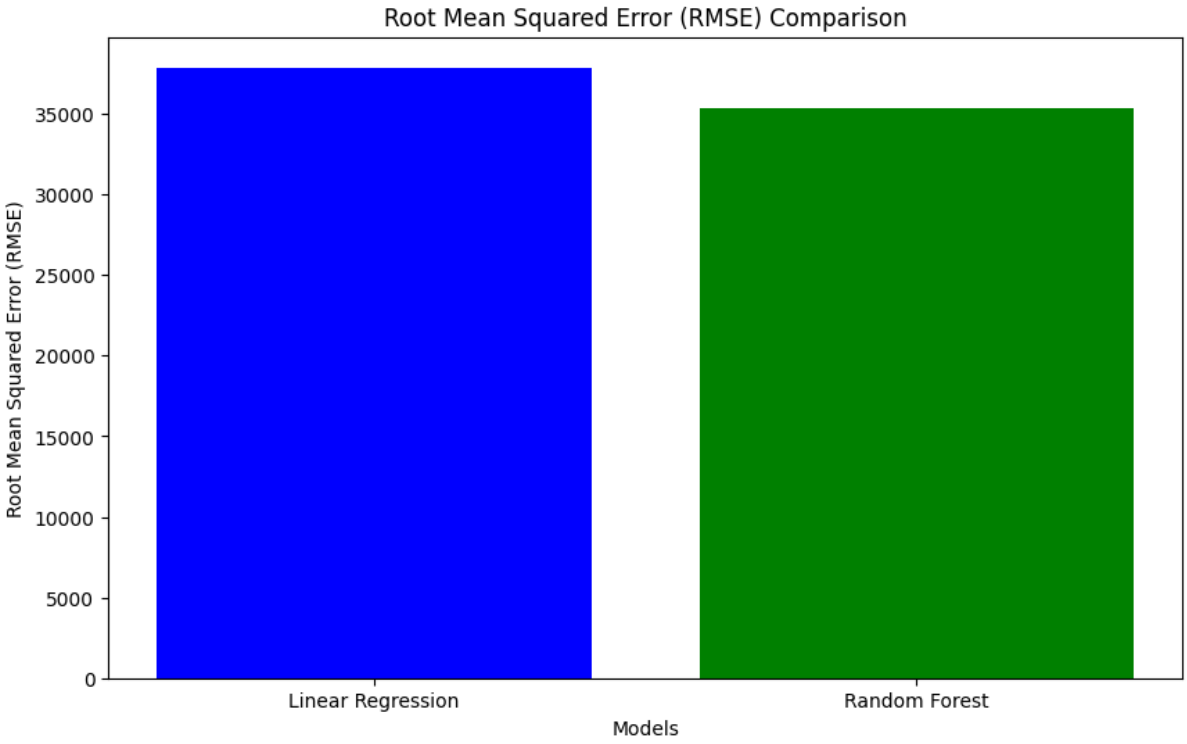
Conclusion Graphs




Id
LotArea
YearBuilt
BsmtFinSF1
TotalBsmtSF
LowQualFinSF
BsmtHalfBath
BedroomAbvGr
Fireplaces
GarageArea
EnclosedPorch
PoolArea
YrSold
Neighborhood_BrDale
Neighborhood_CollgCr
Neighborhood_Gilbert
Neighborhood_Mitchel
Neighborhood_NWAmes
Neighborhood_OldTown
Neighborhood_SawyerW
Neighborhood_Timber
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Mean Squared Error (MSE) Comparison
1e9
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Mean Squared Error (MSE)
Linear Regression
Random Forest
Models


R-squared (R2) Comparison
R-squared (R2)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Linear Regression
Random Forest
Models


Root Mean Squared Error (RMSE) Comparison
Root Mean Squared Error (RMSE)
0
5000
10000
15000
20000
25000
30000
35000
Linear Regression
Random Forest
Models