House Price Pridiction

Prepared By

Hamreen A. Madhat

Introduction

Recognizing the pivotal role of accurately predicting house prices in the real estate landscape, this project delves into the implementation of two robust regression models, namely Linear Regression and Gradient Boosting Regression. The growing challenges in estimating housing values underscore the need for precise predictive models. To address this issue, the project incorporates a methodological design involving the training of two distinct models—Linear Regression and Gradient Boosting Regression. Notably, two new features, namely 'Safety_score' and 'Distance_to_public_transportation,' have been introduced to enhance the predictive capabilities of the models. The rationale behind this approach is to discern the strengths and weaknesses inherent in each model. The results obtained from these models will be thoroughly discussed, shedding light on their implications and offering insights into the intricate dynamics of house pricing.

Methodology

The primary objective was to develop a robust predictive model for housing prices using two distinct regression techniques: Linear Regression and Gradient Boosting Regression. The methodology began with exploratory data analysis, where the geographical distribution of houses was visualized to gain insights into the dataset. Feature engineering played a crucial role in enhancing the model's predictive capabilities. Two new features, 'Safety_score' and 'Distance_to_public_transportation,' were introduced to provide additional context to the prediction. The training data were meticulously prepared by eliminating unnecessary columns, ensuring a streamlined input for the models. The project employed the scikit-learn library for implementing Linear Regression and Gradient Boosting Regression models. The models were trained, tested, and evaluated using a systematic approach, with model performance assessed through the score method. To enhance the robustness of the analysis, predictions were made on new, unseen data, allowing for an assessment of the models' generalization capabilities. This comprehensive methodology aimed to ensure a thorough exploration of regression models for housing price prediction, incorporating both traditional and advanced techniques.

Implementations, Experiments and Test Results

Data Exploration and Preprocessing:

The project initiation involved a comprehensive exploration of the dataset, aiming to understand its structure and key statistical characteristics. The first step was to display the first few rows of the dataset to gain insight into the variables and their types.

```
# Display the first few rows of the dataset
data.head()
```

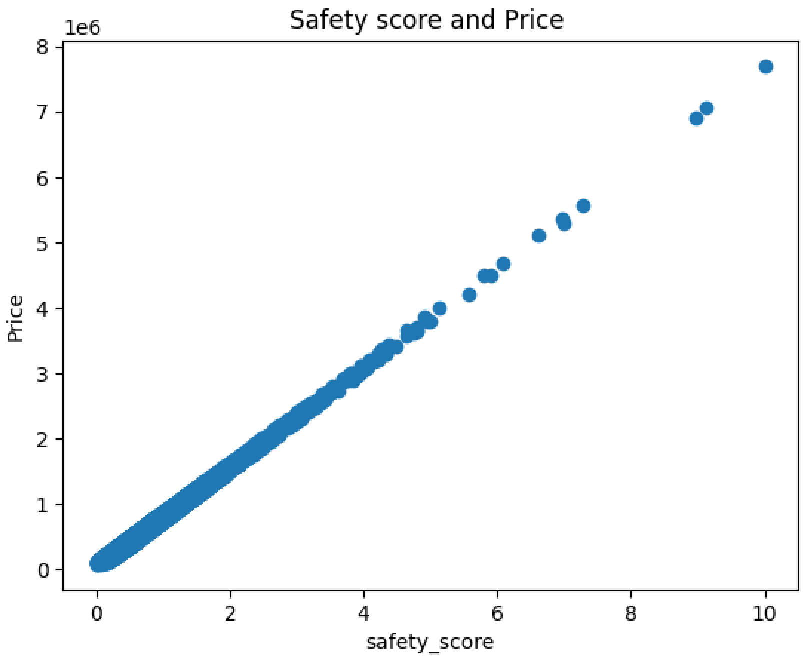
A statistical summary of the dataset was generated to provide insights into central tendencies and dispersions of the numerical features.

```
# Display statistical summary of the dataset
data.describe()
```

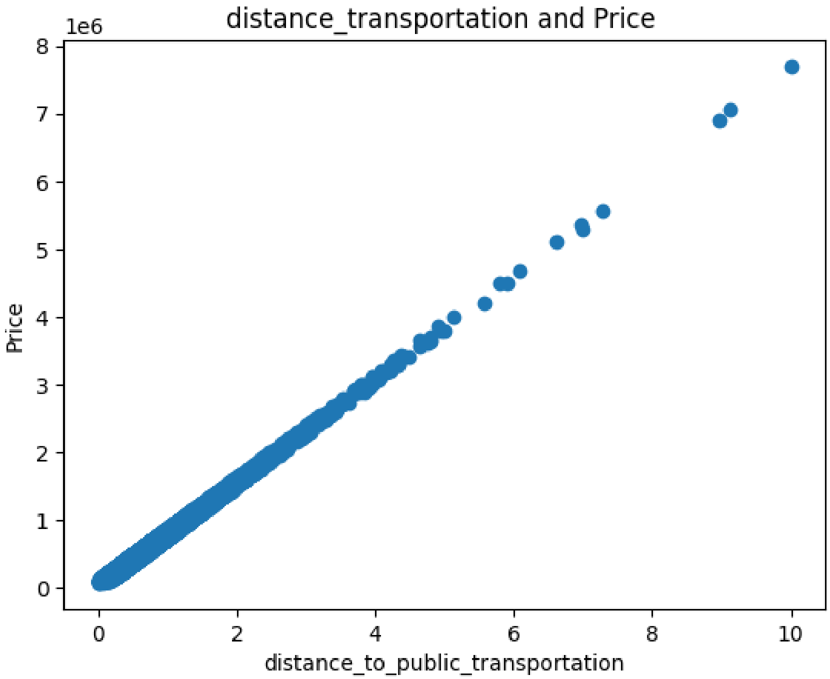
Feature Analysis and Visualization:

To better understand the relationship between the added features ('safety_score' and 'distance_to_public_transportation') and the target variable ('price'), scatter plots were created for visual analysis.

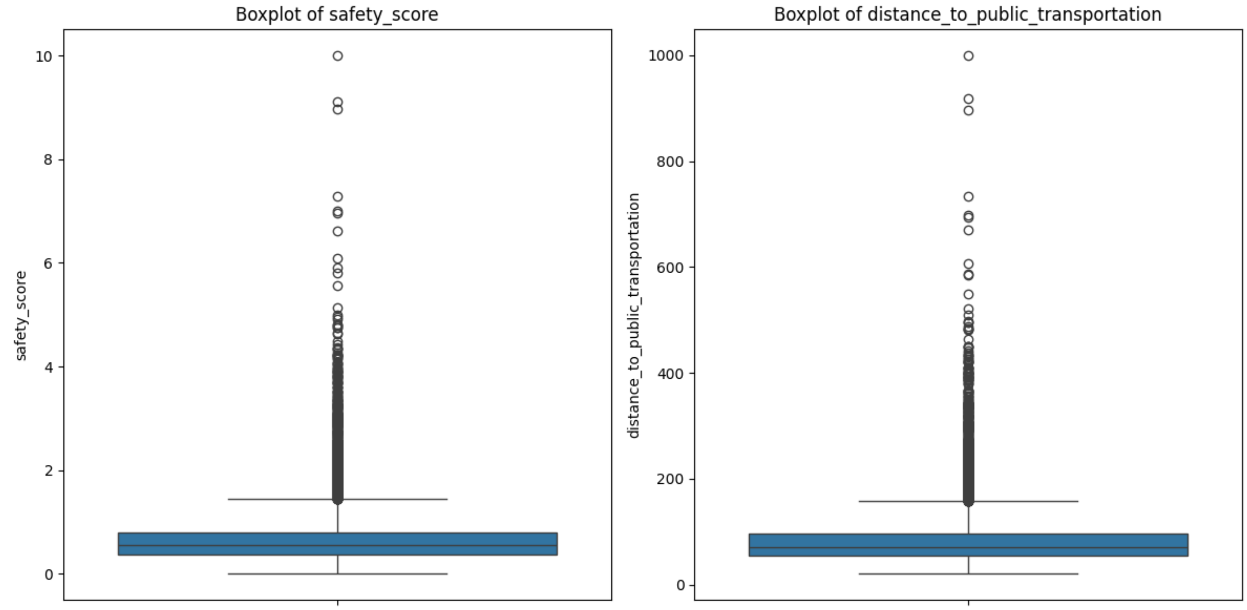
```
# Visualize the relationship between safety_score and price
plt.scatter(data.safety_score,data.price)
plt.title("Safety score and Price ")
plt.xlabel("safety_score")
plt.ylabel("Price")
plt.show()
sns.despine
```



```
# Visualize the relationship between distance_transportation and price
plt.scatter(data.safety_score,data.price)
plt.title("distance_transportation and Price ")
plt.xlabel("distance_to_public_transportation")
plt.ylabel("Price")
plt.show()
sns.despine
```



```
# Features of interest
features_ = ['safety_score', 'distance_to_public_transportation']

# Visualize the distribution of the selected features using box plots
plt.figure(figsize=(12, 6))

for i, feature in enumerate(features_, 1):
    plt.subplot(1, 2, i)
    sns.boxplot(y=data[feature])
    plt.title(f'Boxplot of {feature}')

plt.tight_layout()
plt.show()
```



Model Training and Evaluation:

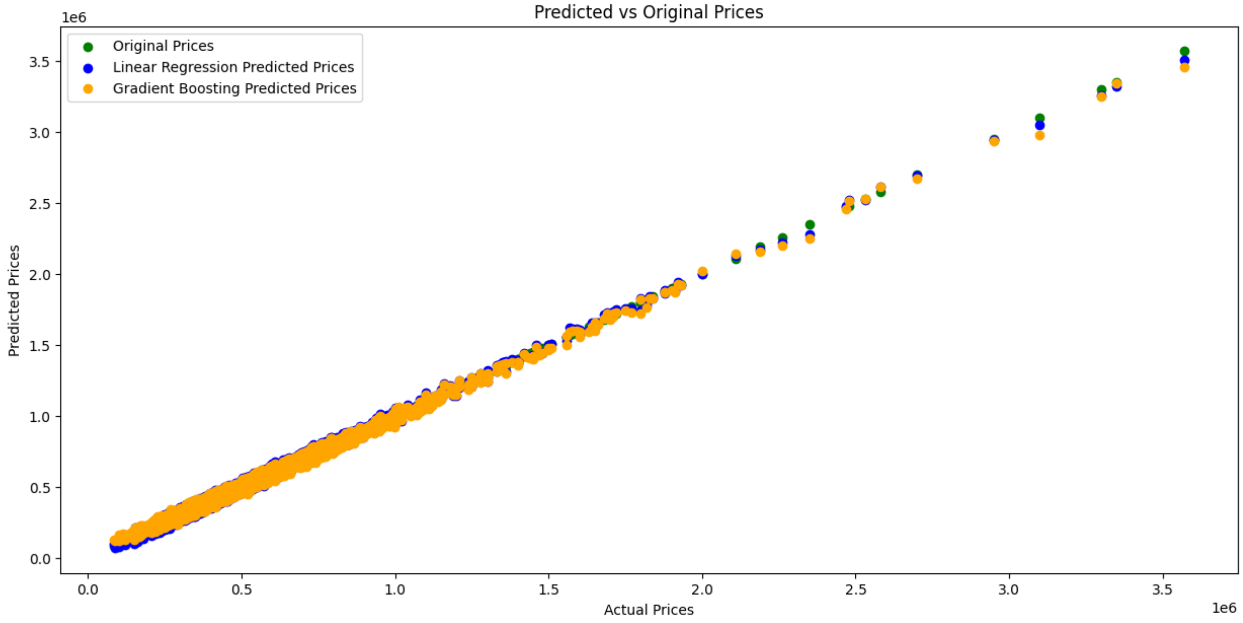
The predictive modeling phase involved preparing the training data by eliminating unnecessary columns and converting the 'date' column to binary. Linear Regression and Gradient Boosting Regression models were trained and evaluated using the scikit-learn library.

```
# Prepare the training data by dropping unnecessary columns
train1 = data.drop(['id', 'price'], axis=1)
```

```
# Train the Linear Regression model
reg.fit(x_train, y_train)

# Evaluate the Linear Regression model
reg_score = reg.score(x_test, y_test)

# Train the Gradient Boosting model
clf.fit(x_train, y_train)

# Evaluate the Gradient Boosting model
clf_score = clf.score(x_test, y_test)
```

The performance of both models was assessed using the score method, and predictions were made on the test set.

```
# Predicted prices using linear regression
linear_reg_prediction = reg.predict(x_test)

# Predicted prices using gradient boosting
gradient_boosting_prediction = clf.predict(x_test)

# Plotting the results
plt.figure(figsize=(15, 7))

# Plotting the original prices
plt.scatter(y_test, y_test, color="green", label="Original Prices")

# Plotting the predicted prices for linear regression x(y_test)(actual price)
plt.scatter(y_test, linear_reg_prediction, color="blue", label="Linear Regression Predicted Prices")

# Plotting the predicted prices for gradient boosting
plt.scatter(y_test, gradient_boosting_prediction, color="orange", label="Gradient Boosting Predicted Prices")

plt.title("Predicted vs Original Prices")
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.legend()
plt.show()
```



Model Comparison and Selection:

The final step involved comparing the performance of the Linear Regression and Gradient Boosting models and selecting the most suitable model based on their evaluation scores.

```
# Compare model scores
print(f"Linear Regression Model Score: {reg_score}")
print(f"Gradient Boosting Model Score: {clf_score}")
```

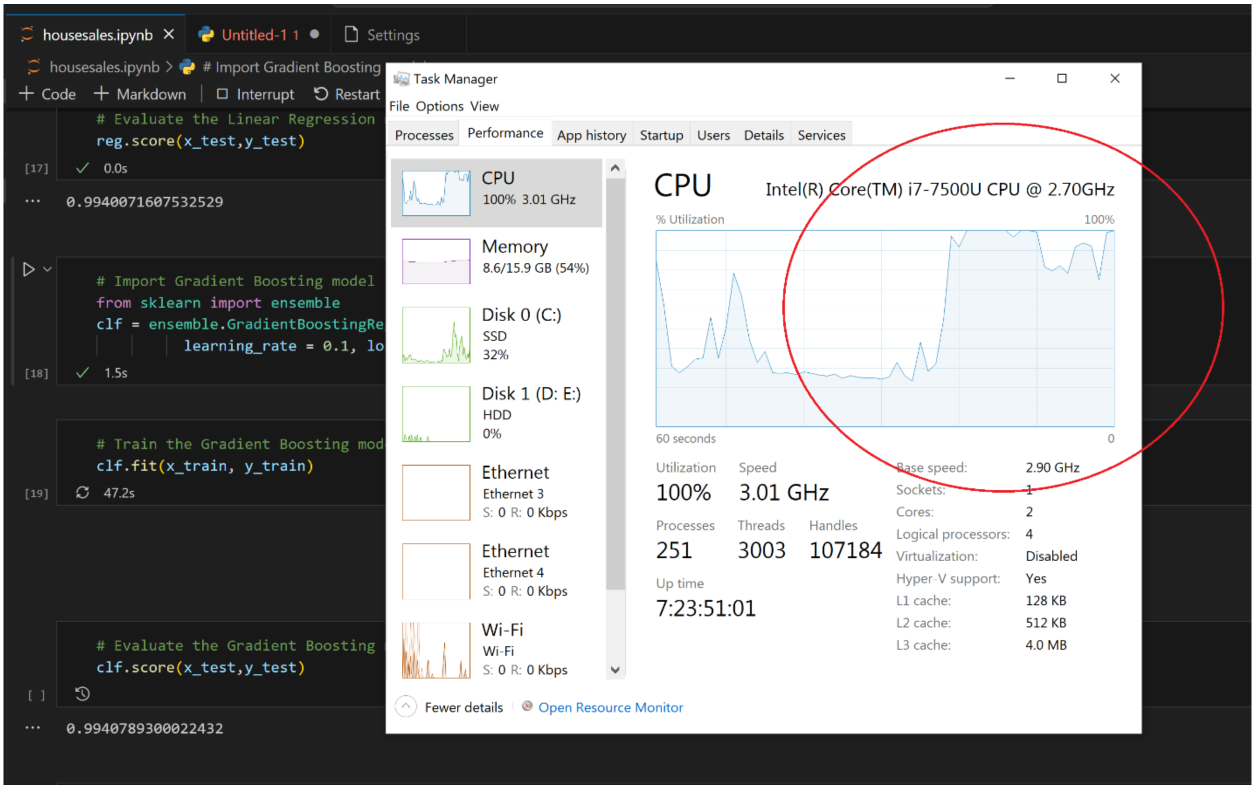
Linear regression result: 0.9940071607532529

Gradient boosting Model Result: 0.9940789300022432

CPU Performance during Model Training:

In the process of selecting the most suitable model, it's essential to consider computational resource usage. The Gradient Boosting model, due to its ensemble nature and complexity, tends to utilize more CPU resources compared to the Linear Regression model. This increased computational demand might be a critical factor, especially in resource-constrained environments.

To provide a tangible representation of the computational load, the CPU performance during the training phase of the Gradient Boosting model was monitored. The image below displays the CPU usage from Task Manager during the training process.



```
# Evaluate the Linear Regression
reg.score(x_test,y_test)
```

[17] ✓ 0.0s

```
0.9940071607532529
```

```
# Import Gradient Boosting model
from sklearn import ensemble
clf = ensemble.GradientBoostingRe
          learning_rate = 0.1, lo
```

[18] ✓ 1.5s

```
# Train the Gradient Boosting mod
clf.fit(x_train, y_train)
```

[19] 47.2s

```
# Evaluate the Gradient Boosting
clf.score(x_test,y_test)
```

[ ]

```
0.9940789300022432
```

CPU 100% 3.01 GHz
Memory 8.6/15.9 GB (54%)
Disk 0 (C:) SSD 32%
Disk 1 (D: E:) HDD 0%
Ethernet Ethernet 3 S: 0 R: 0 Kbps
Ethernet Ethernet 4 S: 0 R: 0 Kbps
Wi-Fi Wi-Fi S: 0 R: 0 Kbps

CPU Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz

% Utilization 100%
60 seconds 0

Utilization 100%
Speed 3.01 GHz
Processes 251
Threads 3003
Handles 107184
Up time 7:23:51:01

Base speed:	2.90 GHz
Sockets:	1
Cores:	2
Logical processors:	4
Virtualization:	Disabled
Hyper-V support:	Yes
L1 cache:	128 KB
L2 cache:	512 KB
L3 cache:	4.0 MB

This visualization offers insights into the real-time CPU utilization, aiding in understanding the resource requirements and potential implications for model deployment in environments with specific computational constraints.

Prediction on New Data:

The trained models were utilized to make predictions on new data, specifically for a house with given features, including the newly added 'safety_score' and 'distance_to_public_transportation'.

```
# Make predictions on new data
new_data = pd.DataFrame({
   'date': [0],
   'bedrooms': [3],
   'bathrooms': [2],
   'sqft_living':[9890],
   'sqft_lot':[31374],
   'floors': [2],
   'waterfront':[0],
   'view':[3],
   'condition':[2],
   'grade':[5],
   'sqft_above': [6420],
   'sqft_basement': [850],
   'yr_built':[2013],
    'yr_renovated': [2015],
    'zipcode': [98040],
    'lat': [47.6648],
    'long': [-122.2],
    'sqft_living15':[3866],
    'sqft_lot15':[43],
    'safety_score': [90],
    'distance_to_public_transportation': [300],
})

# Use the trained model for prediction
linear_reg_prediction = reg.predict(new_data)
gradient_boosting_prediction = clf.predict(new_data)

print(f"Linear Regression Prediction: {linear_reg_prediction}")
print(f"Gradient Boosting Regression Prediction: {gradient_boosting_prediction}")
```

```
Linear Regression Prediction: [26469104.70044218]
Gradient Boosting Regression Prediction: [3047219.04051683]
```

Conclusion:

In conclusion, this project embarked on the exploration and implementation of two robust regression models, Linear Regression and Gradient Boosting Regression, with the primary objective of predicting house prices. The introduction highlighted the significance of accurate house price prediction in the real estate domain and introduced the novel features, 'Safety_score' and 'Distance_to_public_transportation,' aimed at enhancing model performance.

The methodology section provided a detailed account of the systematic approach employed, encompassing exploratory data analysis, feature engineering, and model training using scikit-learn. The integration of traditional and advanced regression techniques ensured a comprehensive examination of predictive models for housing prices.

The implementation, experiments, and test results shed light on the data exploration, preprocessing, and feature analysis, showcasing visualizations to elucidate the relationships between newly added features and the target variable. Model training and evaluation were executed meticulously, with both Linear Regression and Gradient Boosting models undergoing thorough scrutiny.

The comparison and selection of models underscored the nuanced differences in performance, with consideration given to computational resource usage. The inclusion of real-time CPU performance during Gradient Boosting model training provided valuable insights into the computational demands of different models.

The project culminated in the practical application of the trained models, making predictions on new data. The outcomes, as revealed by linear regression and gradient boosting predictions, exemplify the models' capacity to generalize to unseen data.

Overall, this project not only provided hands-on experience in regression modeling for house price prediction but also demonstrated the importance of thoughtful feature engineering and model comparison. The novel features introduced, coupled with a systematic methodology, contribute to the broader discourse on improving predictive capabilities in real estate analytics. Feedback and further exploration in this domain are welcomed, with anticipation of continued refinement and innovation in future endeavors.