House Price Prediction

HackAI

Triple C

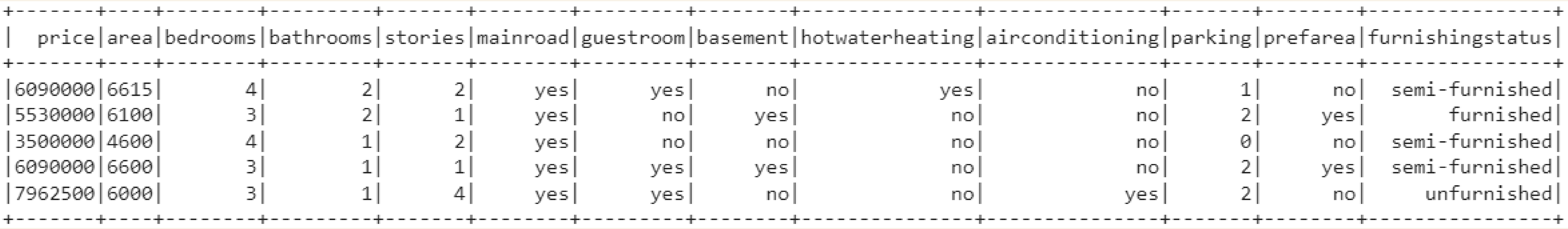
Hao-Chun Chou , Lillian Chen, Larry Chen

1 - Introduction

- **Objective:** To predict house prices based on attributes and find out what features have the most significant impact.
- **Dataset:** 327 houses
- **Features:** Price, Area, Number of Bedrooms, Number of Bathrooms, Stories, Main Road Access, Guest Room Availability, Basement, Hot Water Heating, Air Conditioning, Parking Capacity, Preference Area, and Furnishing Status.
- **Tools:** PySpark, Scikit-learn

1 - Introduction

Dataframe:



```
+-------+----+--------+---------+-------+--------+---------+--------+---------------+---------------+-------+--------+----------------+
|  price|area|bedrooms|bathrooms|stories|mainroad|guestroom|basement|hotwaterheating|airconditioning|parking|prefarea|furnishingstatus|
+-------+----+--------+---------+-------+--------+---------+--------+---------------+---------------+-------+--------+----------------+
|6090000|6615|       4|        2|      2|     yes|      yes|      no|            yes|             no|      1|      no|   semi-furnished|
|5530000|6100|       3|        2|      1|     yes|       no|     yes|             no|             no|      2|     yes|        furnished|
|3500000|4600|       4|        1|      2|     yes|       no|      no|             no|             no|      0|      no|   semi-furnished|
|6090000|6600|       3|        1|      1|     yes|      yes|     yes|             no|             no|      2|     yes|   semi-furnished|
|7962500|6000|       3|        1|      4|     yes|      yes|      no|             no|            yes|      2|      no|      unfurnished|
+-------+----+--------+---------+-------+--------+---------+--------+---------------+---------------+-------+--------+----------------+
```

Unique values:
price: 168 / area: 203 / bedrooms: 6 / bathrooms: 3 / stories: 4 / mainroad: 2 / guestroom: 2 / basement: 2 / hotwaterheating: 2 / airconditioning: 2 / parking: 4 / prefarea: 2 / furnishingstatus: 3

2 - Data Cleaning

Data Quality Check:

- **Consistency:** No inconsistencies detected within the dataset.
- **Completeness:** Zero missing values or null entries across all features.
- **Uniqueness:** Confirmed absence of duplicate rows, ensuring data integrity.

Outlier Analysis:

- **Initial Approach:** Considered removing outliers based on the 10th to 90th quantile range to enhance model accuracy.
- **Decision:** Due to the dataset's limited size, removing outliers was deemed potentially detrimental to the model's ability to learn from a comprehensive range of data. Consequently, all data points were retained to maximize the dataset's utility and representativeness.

3 - Feature Engineering: PySpark

Categorical to Numeric:
Utilized StringIndexer to convert categorical columns (mainroad, guestroom, basement, hotwaterheating, airconditioning, prefarea, furnishingstatus) into indexed numerical columns to facilitate their use in machine learning models.

Log Transformation: Converted price and area to their logarithmic values to normalize their distribution and potentially enhance model performance.

Feature Vector Assembly:
Employed VectorAssembler to combine all feature columns into a single vector, streamlining the input for model training.

3 - Feature Engineering: Scikit-learn

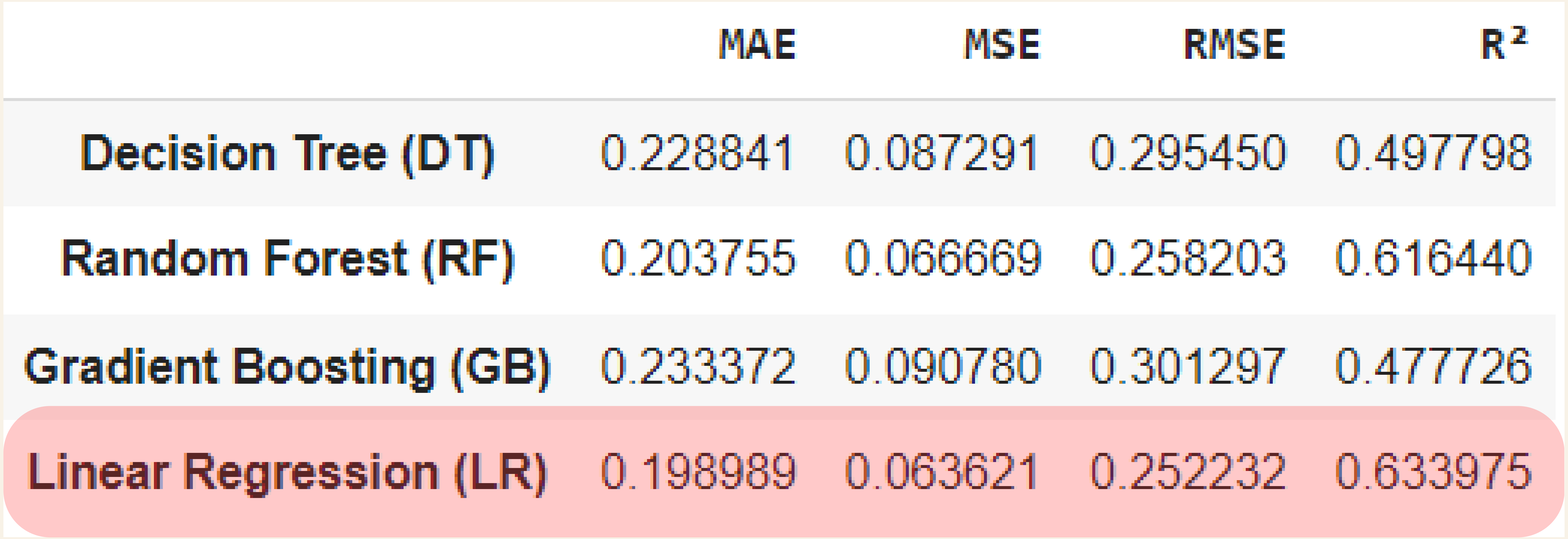
Binary Encoding: Transformed binary categorical columns (mainroad, guestroom, basement, hotwaterheating, airconditioning, prefarea) into binary (1/0) format based on their values.

Ordinal Encoding: Applied a mapping to the furnishingstatus column to reflect its ordinal nature (unfurnished = 0, semi-furnished = 1, furnished = 2).

Log Transformation: Converted price and area to their logarithmic values to normalize their distribution and potentially enhance model performance.

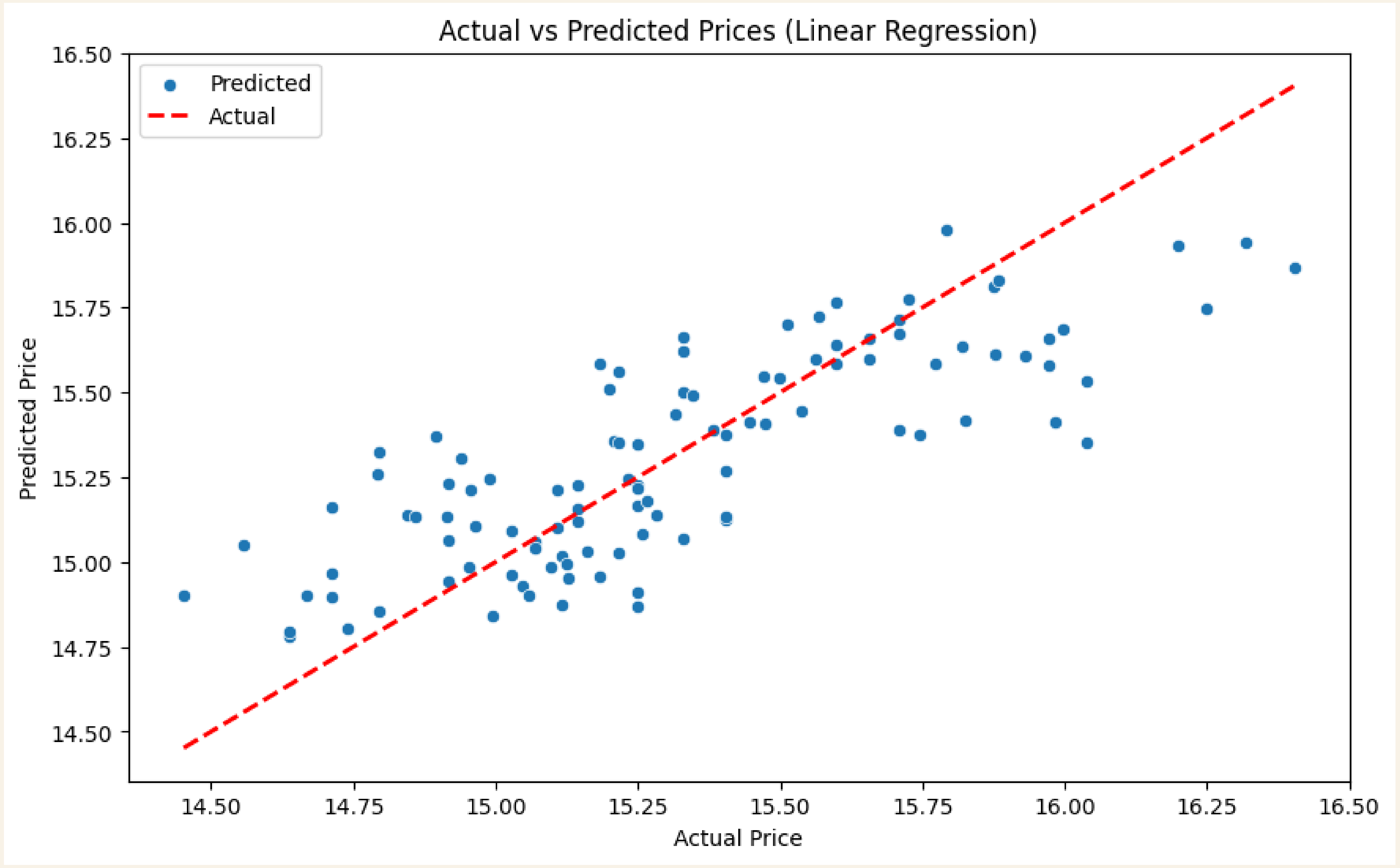
Feature Selection: Dropped the hotwaterheating feature due to its low correlation (0.087) with the target variable, aiming for a more efficient and relevant feature set.

4 - Model Performance: PySpark

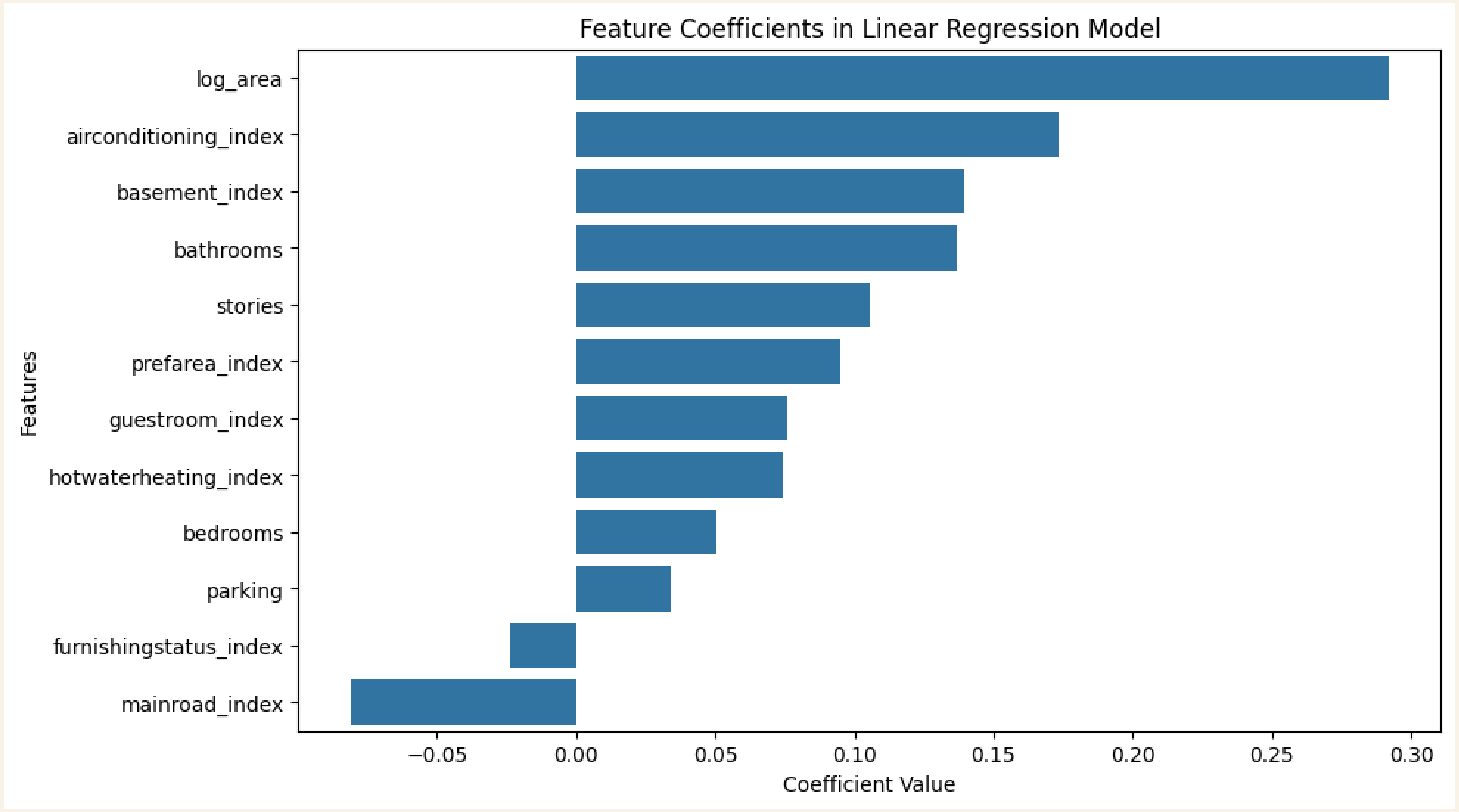


	MAE	MSE	RMSE	R²
Decision Tree (DT)	0.228841	0.087291	0.295450	0.497798
Random Forest (RF)	0.203755	0.066669	0.258203	0.616440
Gradient Boosting (GB)	0.233372	0.090780	0.301297	0.477726
Linear Regression (LR)	0.198989	0.063621	0.252232	0.633975

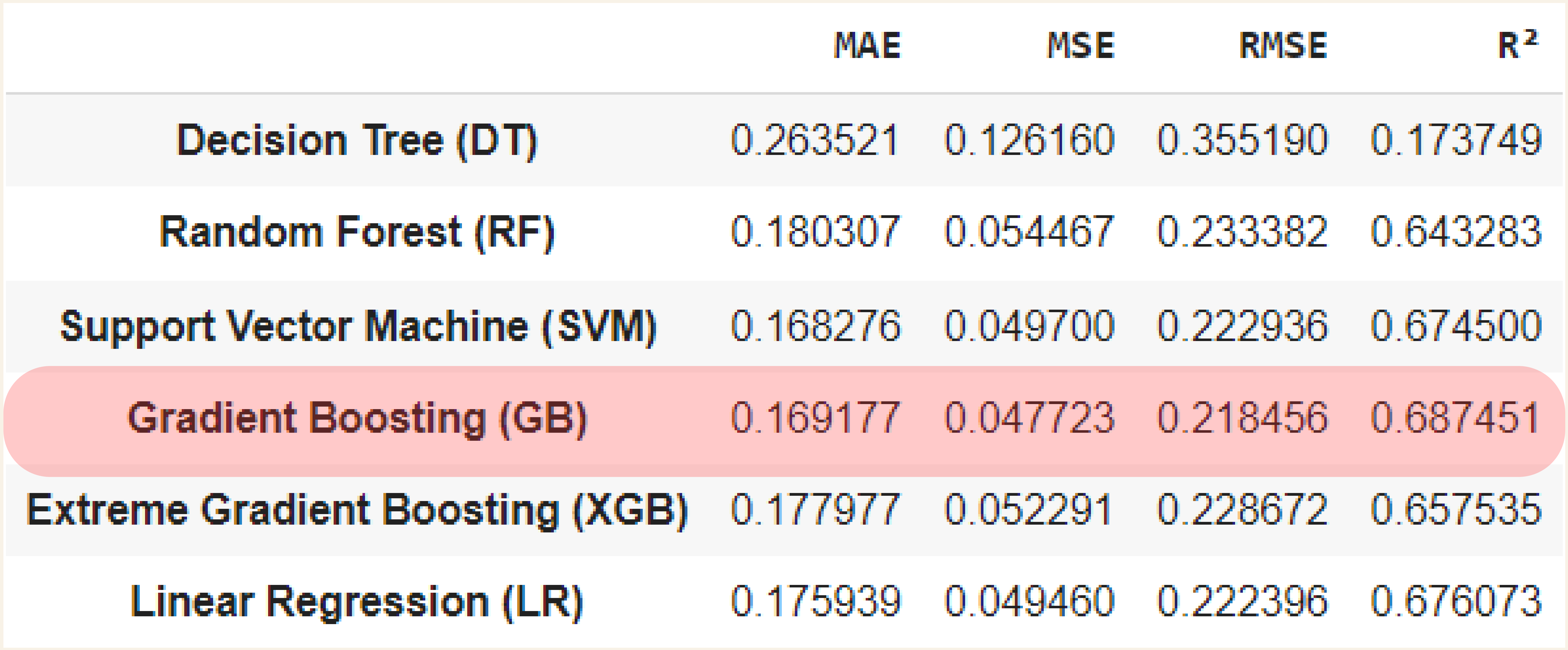
4 - Model Performance: PySpark



4 - Model Performance: PySpark

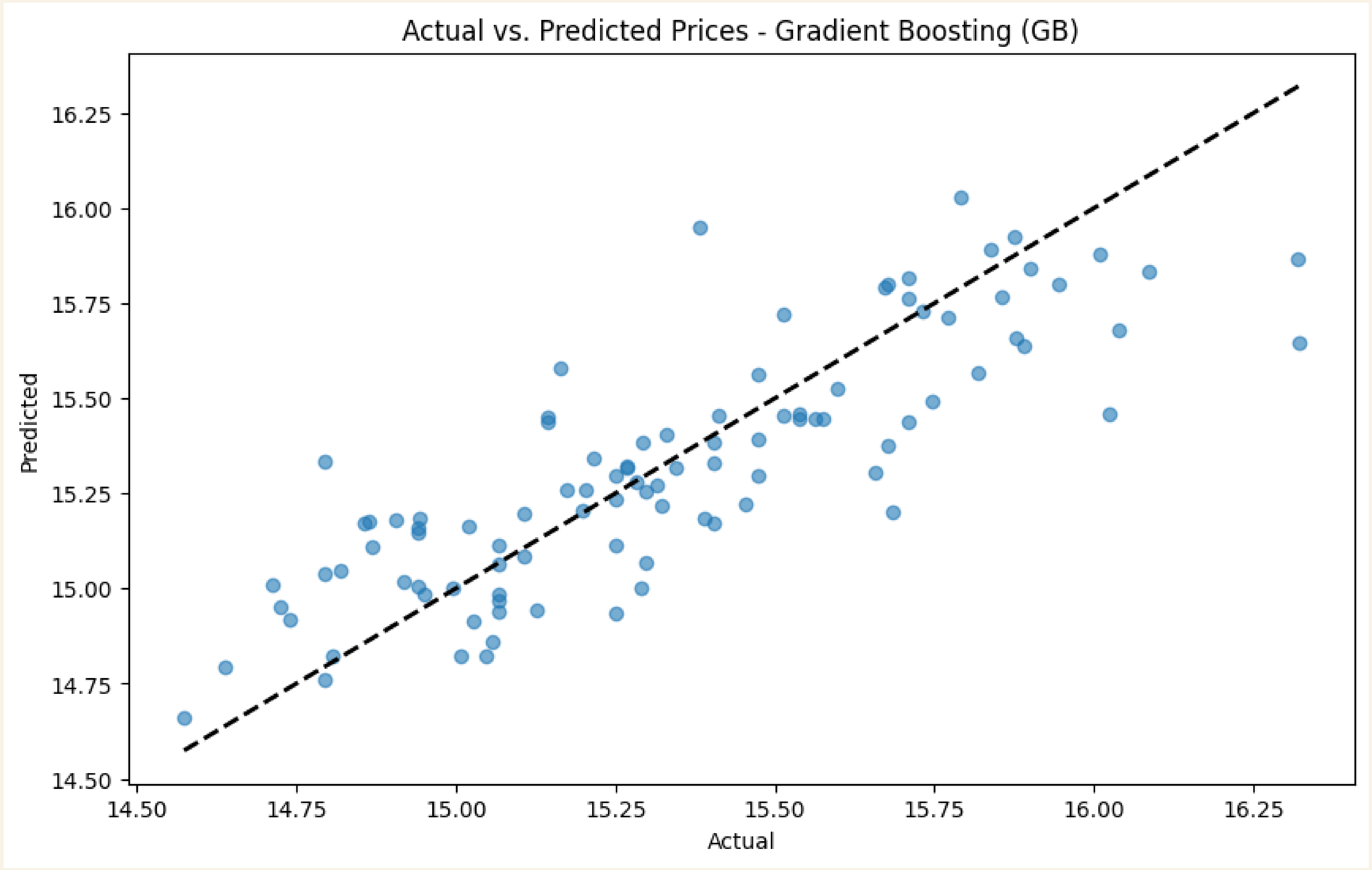


4 - Model Performance: Scikit-learn

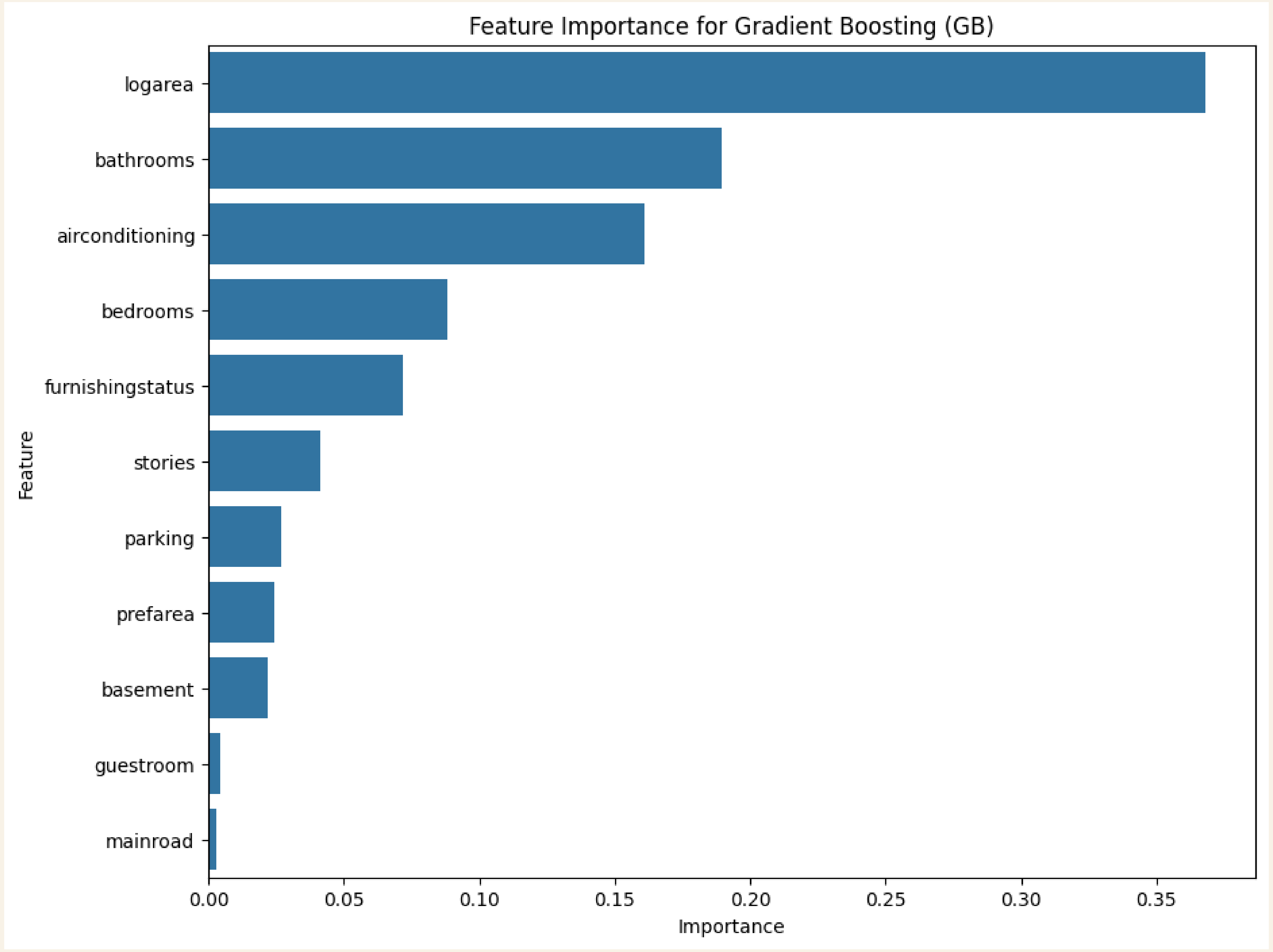


	MAE	MSE	RMSE	R^2
Decision Tree (DT)	0.263521	0.126160	0.355190	0.173749
Random Forest (RF)	0.180307	0.054467	0.233382	0.643283
Support Vector Machine (SVM)	0.168276	0.049700	0.222936	0.674500
Gradient Boosting (GB)	0.169177	0.047723	0.218456	0.687451
Extreme Gradient Boosting (XGB)	0.177977	0.052291	0.228672	0.657535
Linear Regression (LR)	0.175939	0.049460	0.222396	0.676073

4 - Model Performance: Scikit-learn



5 - Model Performance: Scikit-learn



6 - Conclusions

Best Performing Model:
Our analysis found that the Gradient Boosting model implemented with Scikit-learn achieved the highest R^2 score of 0.687. This indicates a moderately strong ability to predict house prices based on the features provided.

Feature Engineering Insights:
During our experimentation, we introduced a "price per sqft" feature, which initially appeared to significantly improve model accuracy. However, upon further consideration, we recognized that including both "area" and "price per sqft" as features essentially leaks the target variable (price) into our model. This realization led us to exclude the "price per sqft" feature to ensure our model's practical applicability in real-world scenarios, where such direct information may not be available beforehand.

6 - Conclusions

Challenges with Deep Learning:
Attempts to employ neural networks resulted in an R^2 score of -14, signaling a poorly fitting model. This outcome underscores the challenge of using deep learning techniques on relatively small datasets. Our experience suggests that the complexity of neural networks requires more extensive data to effectively capture and generalize the underlying patterns without overfitting.

Final Thoughts:
Our journey through various models and feature engineering techniques has highlighted the importance of matching model complexity with data availability and the relevance of features to the task at hand. The Gradient Boosting model stands out for its balance of complexity and performance, offering valuable predictions that can inform real estate decisions.

6 - Conclusions

Future Directions:
Improving our models further could involve exploring more sophisticated feature engineering, incorporating additional data sources, or applying more advanced ensemble techniques. Additionally, data augmentation strategies might be considered to better support complex models like neural networks.

Thanks