Data Visualization for Machine Learning

Final Project Report

Housing Prices Prediction & Analysis

Prepared By: Group 9

Ahmed Al-Mahdi

Luke Gallat-Opoku

Muhammad Osama

1. Problem Definition and Dataset Selection

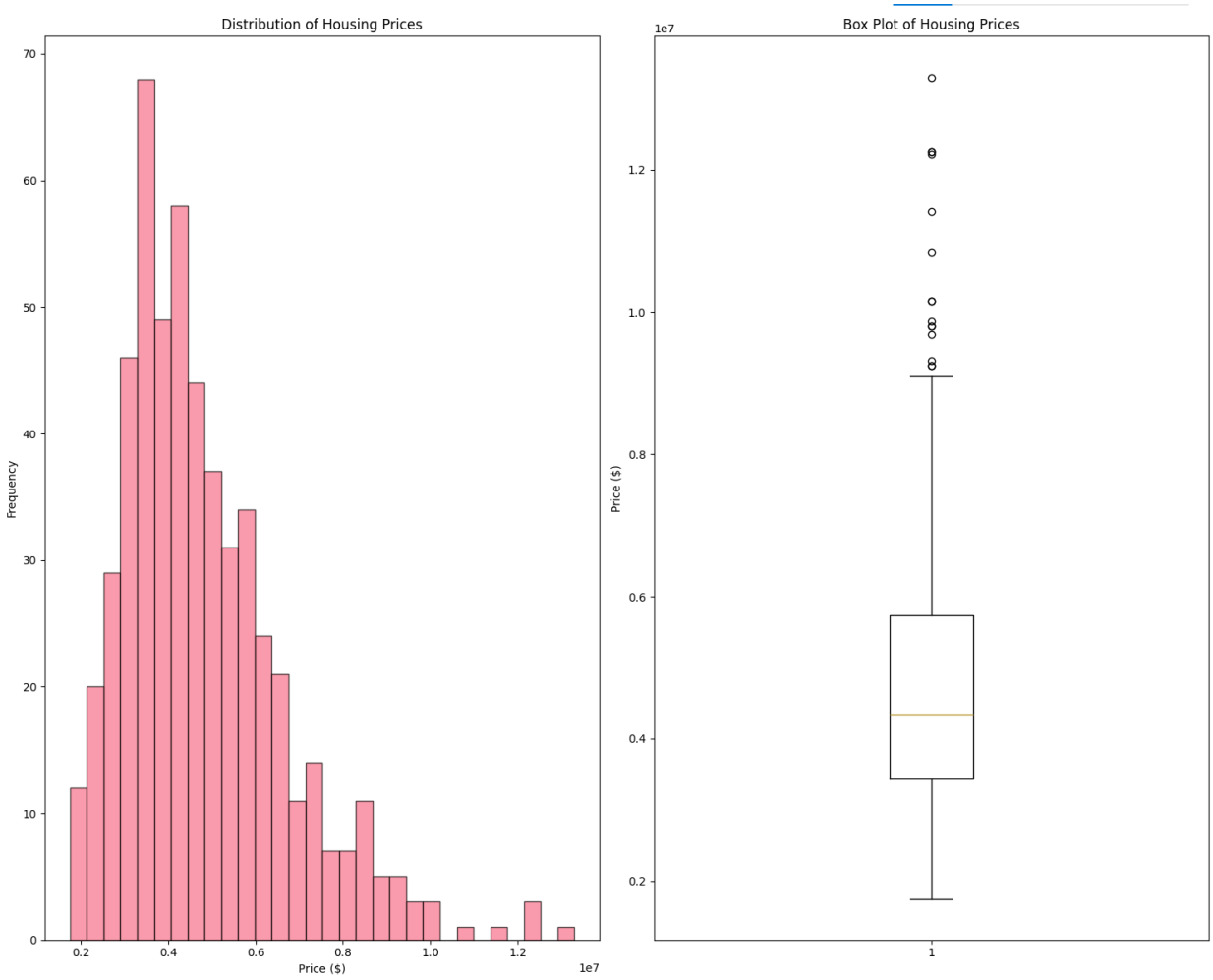
In this project we have aimed to solve housing prices prediction problem. Here the goal of our project is to predict the price of housing based on the given features. We will be using multiple types of regression models for prediction of price. Some of the factors that increase the complexity of the problem are:

- Housing prices are influenced by both quantitative (e.g., area, number of bedrooms) and qualitative (e.g., furnishing status, location preferences) variables.
- There is potential for non-linear relationships between features and the target variable.
- Interpretability is critical for real-world applications, especially in real estate.

We have selected the dataset from Kaggle, the name of the dataset is Housing Price Dataset. It contains 545 records and 13 features. The dataset has both categorical and numerical features.

- **Numerical Features**:
    - area: The total area on which house is built (continuous)
    - bedrooms: Number of bedrooms (discrete)
    - bathrooms: Number of bathrooms (discrete)
    - stories: Number of floors (discrete)
    - parking: Number of parking spaces
- **Categorical Features**: All Binary (Yes/ No)
    - mainroad
    - guestroom
    - basement
    - hotwaterheating
    - airconditioning
    - prefarea
    - furnishingstatus: Multi-class categorical (semi-furnished, unfurnished, furnished)

- **Target**:
    - price: House price in currency units (continuous)



- The Histogram and boxplot of price helps us to understand skewness, outliers, and help decide if we need any transformation or any outlier handling strategies.

2. Data Preprocessing

For the dataset preprocessing we have done the following steps:

1. **Data Cleaning:**
    - For data cleaning first we checked for missing values, however our dataset didn't have any missing values.
    - No duplicates found in the dataset.
2. **Categorical Encoding:**

- For the categorical features we label encoded them to (Yes to 1 and No to 0)
- Furnishingstatus is the only column where we used one hot encoding to handle its multiclass nature. After one hot encoding semi-furnished, unfurnished, furnished these are the new features.

3. **Data Splitting:**

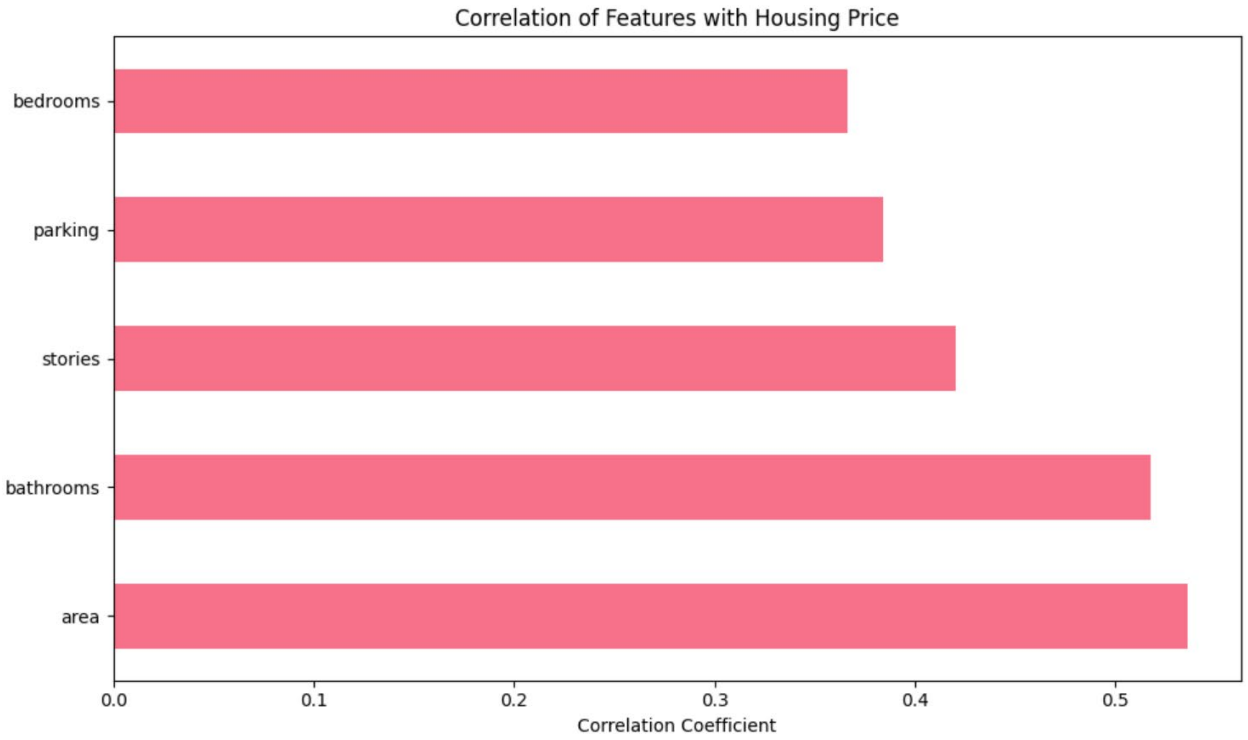
    - We have split the data into 2 parts training data which is 80 % and test data that is 20%. With a random state of 42 which ensures the codes reproducibility.

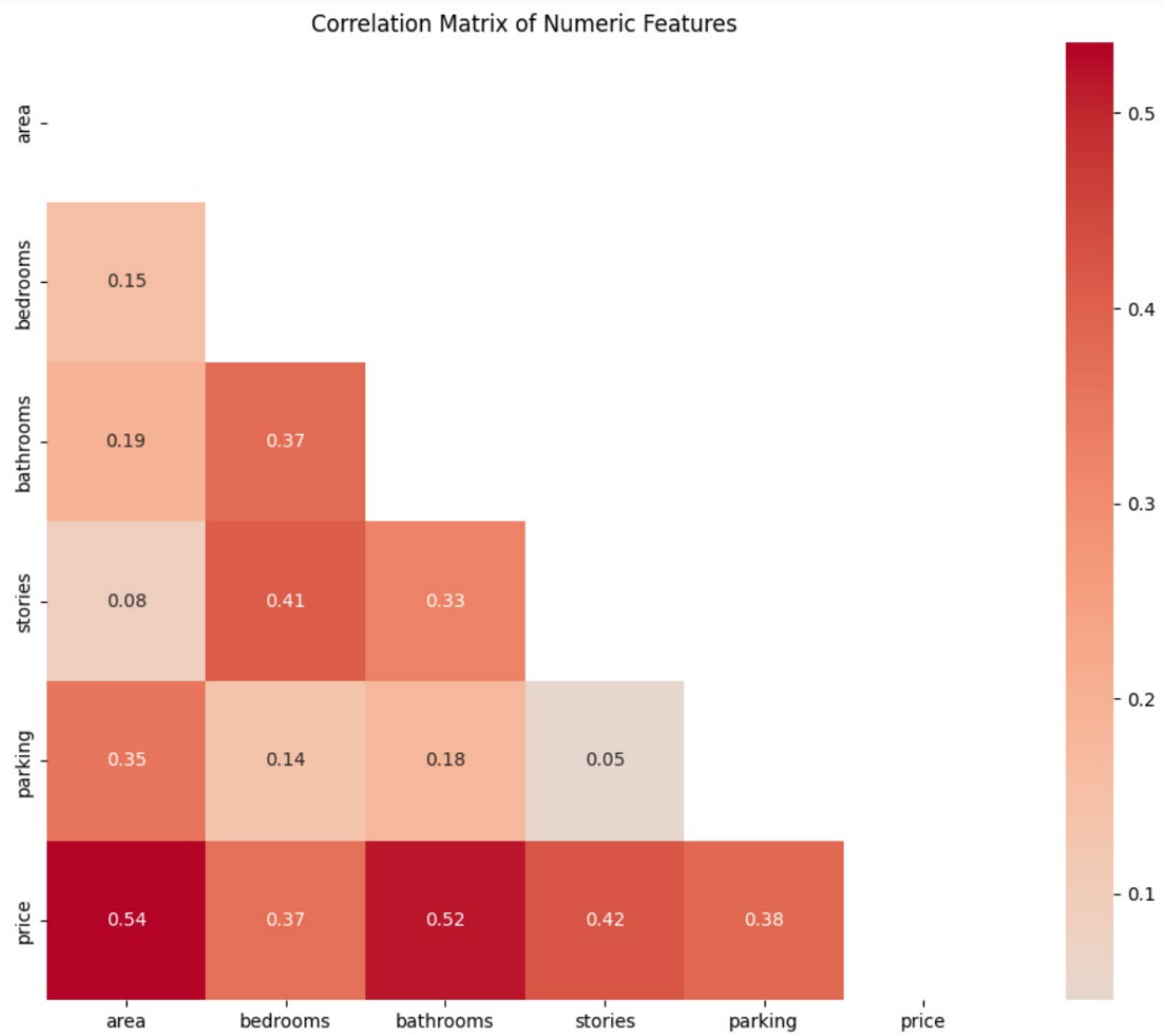
4. **Feature Engineering:**

    - We didn't have any irrelevant features in our dataset, however preserved all the column to predict the housing prices

Challenges Faced:

- In our dataset we have mixed data types which require a careful encoding technique to ensure it is compatible with our models.
- Our dataset is relatively small which brings the risk of overfitting.





- The correlation matrix shows how each feature affects the price of the house. It is an excellent choice for pairwise feature relationships. Here we can see that area and price are highly correlated.

```
Categorical variables encoded successfully!
Processed dataset shape: (545, 15)
New columns: ['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'parking', 'prefarea', 'furnishing_furnished', 'furnishing_semi-furnished', 'furnishing_unfurnished']
Missing values: 0
```

- The above snapshot shows all the preprocessing that are successfully applied on our dataset.

3. Model Selection

Our goal is to build a robust and interpretable regression pipeline, for this purpose we have selected 4 models including simple, regularized and ensemble-based model, and each

model brings a unique set of strengths which enables us to make generalized and good predictions.

1. **Linear Regression**:
    - It acts as a baseline model for our task
    - It offers high interpretability
    - Linear regression assumes a linear relation among all the features
    - However, it is sensitive to outliers
    - Useful for understanding the direct effect of features on target variable
2. **Ridge Regression**:
    - Ridge Regression introduces L2 regularization which is helpful in reducing the effect of multicollinearity.
    - It is ideal when features are correlated, which we observed in our correlation heatmap.
    - Ridge Regression helps us to achieve better generalization than simple linear regression.
3. **Random Forest Regressor**:
    - Random Forest is an ensemble method model which is based on decision trees.
    - It can capture nonlinear relations between variables and provides feature importance natively.
    - It is robust to outliers and very useful for a moderate-sized dataset like ours.
4. **Gradient Boosting Regressor**:
    - Gradient Boosting builds the model sequentially; it improves errors based on previous trees results.
    - It achieves high accuracy by combing many weak learners
    - It is less prone to overfitting than Random Forest due to boosting.

We have selected each of these models to balance performance, flexibility and interpretability to suit our dataset which has diverse feature types.

4. Project Implementation

We have implemented this project in python and below are the various libraries we have used.

- **Pandas & NumPy** for data manipulation
- **Matplotlib, Seaborn, Plotly** for visualizations
- **scikit-learn** for modeling and metrics
- **SHAP & LIME** for model interpretability
- **Dash** for interactive plots and dashboard creation

Some key components in our pipeline are:

- Data exploration and transformation
- Training pipeline with evaluation logic
- Model comparison framework
- Feature importance and explanation visualizations

These are some steps that helped us verify each step in the pipeline and provided visual confirmation of data integrity.

5. Training & Evaluation

We trained all these models using the training data and for most of the part we used the default hyperparameters so that we could establish a strong baseline. The following metrics were used to calculate performance.

- **Root Mean Squared Error (RMSE)**: Emphasizes large errors. Penalizes larger errors more, it is important for real estate pricing where a 1M overestimate is critical.
- **R^2 Score**: It indicates how well the model explains variance. Higher values suggest better predictive power.
- **MAE**: Measures average prediction deviation. It reflects average error without exaggerating large deviations.

Training/Test Performance Summary:

Model Comparison:

	Model	Train R^2	Test R^2	Train RMSE	Test RMSE	Train MAE	Test MAE
0	Linear Regression	0.6859	0.6529	984051.9237	1.324507e+06	719242.8937	9.700434e+05
1	Ridge Regression	0.6859	0.6524	984110.3938	1.325455e+06	718280.7761	9.705895e+05
2	Random Forest	0.9491	0.6121	396097.6193	1.400219e+06	281002.6647	1.019528e+06
3	Gradient Boosting	0.8684	0.6665	636997.3163	1.298372e+06	474214.3211	9.668407e+05

Best performing model: Gradient Boosting

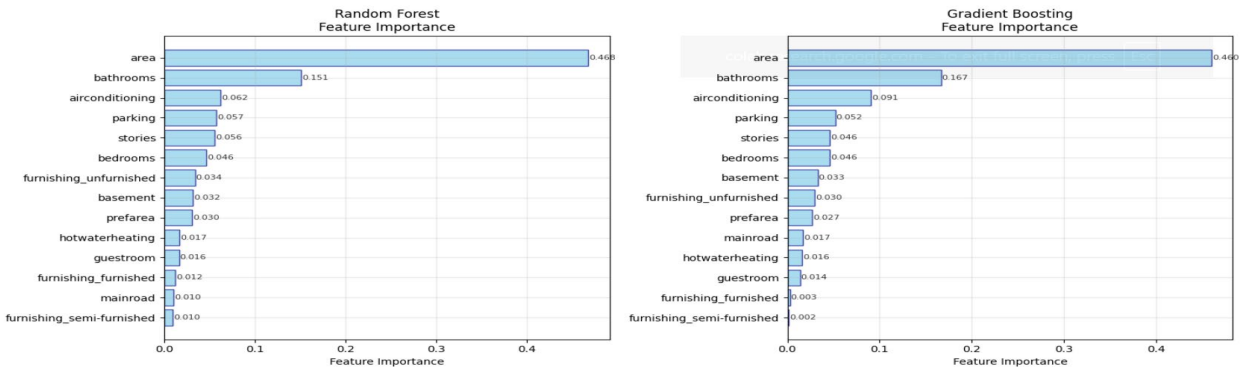
Based on the results we can see that Gradient boosting outperformed all the other models on test data, it provided the best generalization while controlling overfitting. While Random Forest showed better results on train data.

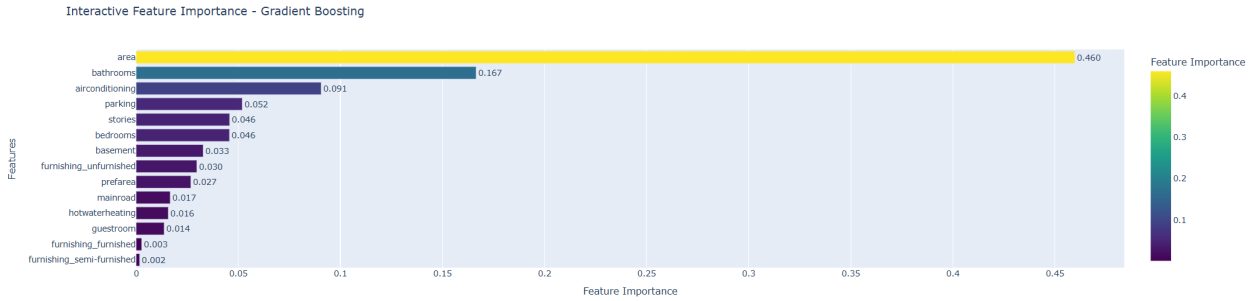
Model	Train R^2	Test R^2	Train RMSE	Test RMSE
Linear Regression	0.6859	0.6529	$984,052	$1,324,507
Ridge Regression	0.6859	0.6524	$984,110	$1,325,455
Random Forest	0.9491	0.6121	$396,098	$1,400,219
Gradient Boosting	**0.8684**	**0.6665**	$636,997	**$1,298,372**

6. Results Interpretation

6.1 Traditional Feature Importance

- Tree-based models (Random Forest & Gradient Boosting) provided intrinsic feature importance.
- area, bathrooms, and prefarea were most important.





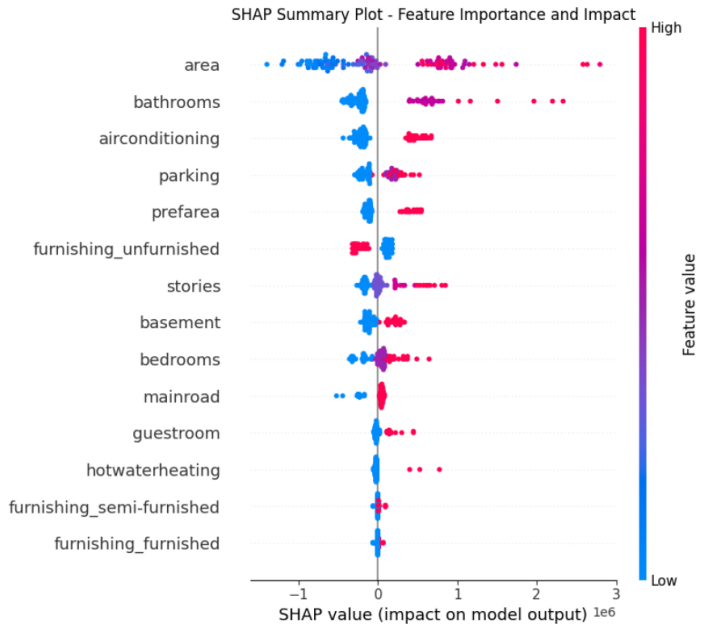
Interactive plot shows feature importance for Gradient Boosting
Test R^2 Score: 0.6665

These horizontal bar chart shows feature importance analysis. It shows how important each feature is in making a prediction. We can see in both models the top 5 important features are same. While area is contributing the most in the price prediction.

- Horizontal bar chart of feature importances: Makes it easy to communicate what drives housing prices.

6.2 SHAP Analysis for Gradient Boosting

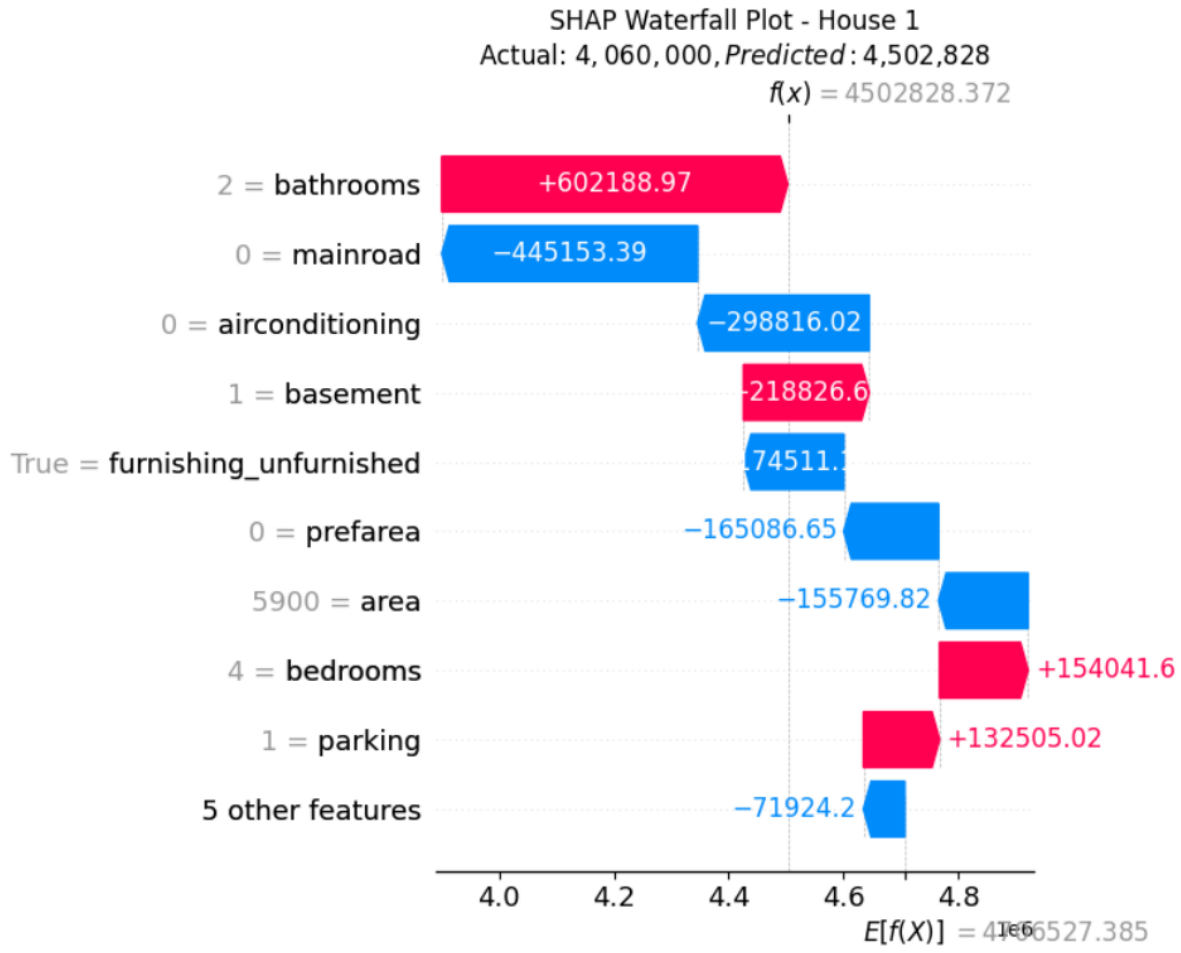
- **SHAP Summary Plot**:
    - It captures both feature importance and direction of effect.
    - It is also useful in our case as it showed bedrooms and furnishingstatus had variable effects across data points.



The SHAP summary plot shows:
- Each point represents a house from the test set
- X-axis shows the SHAP value (impact on model output)
- Color represents the feature value (red=high, blue=low)
- Features are sorted by importance (top to bottom)

- **SHAP Waterfall Plots**:
    - Visualize contribution of each feature to a single prediction.
    - Ideal for explaining individual predictions to stakeholders or customers.

```
SHAP Waterfall Plots for Individual Predictions:
Showing how each feature contributes to the prediction for specific houses
```

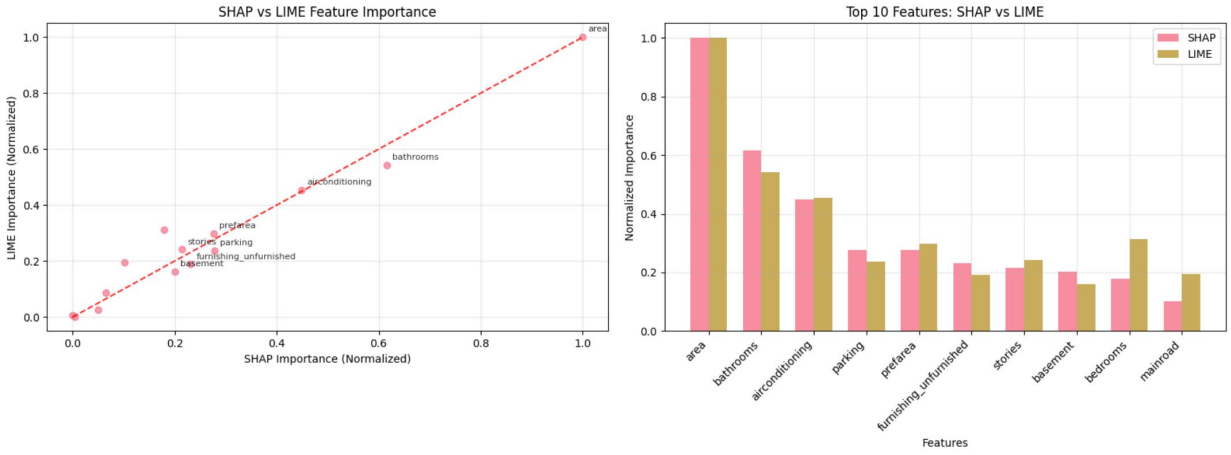



```
House 1 characteristics:
  area: 5900
  bedrooms: 4
  bathrooms: 2
  stories: 2
  mainroad: 0
  guestroom: 0
  basement: 1
  hotwaterheating: 0
  airconditioning: 0
  parking: 1
  prefarea: 0
  furnishing_furnished: False
  furnishing_semi-furnished: False
  furnishing_unfurnished: True
  Actual Price: $4,060,000
  Predicted Price: $4,502,828
  Error: $442,828
```

7. Visualizations and Their Purpose

Visualization	Purpose	Why It Matters Here
Histograms / Boxplots	Understand distribution & outliers	Helped detect skewed price distribution
Correlation Matrix	Identify linear relationships	Guided model selection and feature engineering
Count/Bar Charts	Explore categorical impact	Showed price differences based on yes/no features
SHAP Summary & Waterfall	Explain feature impact	Essential for explaining model predictions
Residual Plots	Diagnose errors	Ensured model wasn't biased
Interactive Dashboards	Explore feature relationships	Allows stakeholders to visually explore model behavior

8. Shap Vs Lime Feature Importance



It provides a comparative analysis of feature importance between SHAP and LIME, two popular model interpretability methods. On the left, the scatter plot illustrates how each feature ranks in importance according to SHAP and LIME, with the red dashed diagonal line highlighting points of agreement—features falling on this line are rated equally by both methods. Notably, features like area and bathrooms appear close to the line, suggesting

consistency between SHAP and LIME. On the right, the bar chart showcases the top 10 features as determined by each method, using pink bars for SHAP and brown bars for LIME. While there's some overlap in top features, their rankings and importance scores differ, emphasizing that interpretability techniques can yield varying perspectives on what influences the model's predictions.

9. Conclusion and Reflection

Overall, we as a team success

Our strengths included:

- Sound data exploration techniques
- Thoughtful model selection
- Balanced performance and interpretability
- Used multiple visualization to get useful insight on our dataset and predictions

We tackled encoding, data sparsity, and overfitting using regularization and ensemble methods. Visualization played a key role in explaining both data patterns and model behavior.

In terms of data selection, we could have made a better selection when it came to housing prices. While we had a solid number of features that were important in determining house prices, other impactful information such as location and currency would have been extremely insightful to have in order to provide more context. Our data just included price, and we just made the assumption that it was in dollars, however we do not know for certain if this is accurate.

We also had a few outliers, which is definitely to be expected when examining house prices, however we believe that it could have also been beneficial to create models that excluded them. This would give a more accurate representation of the typical home prices in the area. With our current models, if someone wanted to use a model to determine the price of an average house given their specifications, the price would be slightly higher due to the outliers.

Key Lessons:

- Gradient Boosting strikes a balance between performance and interpretability.
- SHAP values provide meaningful, actionable insights.
- Visualizations not only support our conclusions but enhance model transparency.

10. Future Work

- Perform hyperparameter tuning (GridSearchCV, RandomizedSearchCV)
- Evaluate advanced models like **XGBoost** or **LightGBM**
- Use **cross-validation** for more robust performance estimation
- Perform segmentation-based SHAP to explain subgroups

Link to our Dashboard: https://housing-price-dashboard.streamlit.app/

Link to Dataset: https://www.kaggle.com/datasets/saurabhbadole/housing-price-data