Predicting 2016 Miami House Prices using Decision Trees

Kristy Hamlin
3354342
07/18/23
CAP4612 Summer 2023

Motivation

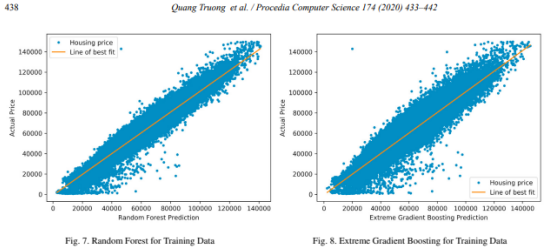


Fig. 7. Random Forest for Training Data
Fig. 8. Extreme Gradient Boosting for Training Data

- With the continued rise of the cost of living and stagnation of wages in the United States, finding affordable housing is more difficult than ever before.
- This project aims to use Machine Learning to analyze the patterns among house prices in Miami to provide insight for home buyers. What factors may they be willing to compromise on? Do they suspect a house is overpriced?
- Truong (2020) analyzed Beijing Housing Prices comparing several alternate tree-based models and achieved impressive results. (https://doi.org/10.1016/j.procs.2020.06.111)

Table 2. Prediction Results

Model	RMSLE	
	Train Set	*Test Set*
Random Forest	**0.12980**	0.16568
Extreme Gradient Boosting	0.16118	0.16603
Light Gradient Boosting Machine	0.16687	0.16944
Hybrid Regression	**0.14969**	0.16372
Stacked Generalization Regression	0.16404	**0.16350**

Truong, Q., Nguyen, M., Dang, H., & Mei, B. (2020). Housing price prediction via Improved Machine Learning Techniques. Procedia Computer Science, 174, 433–442.

https://doi.org/10.1016/j.procs.2020.06.111

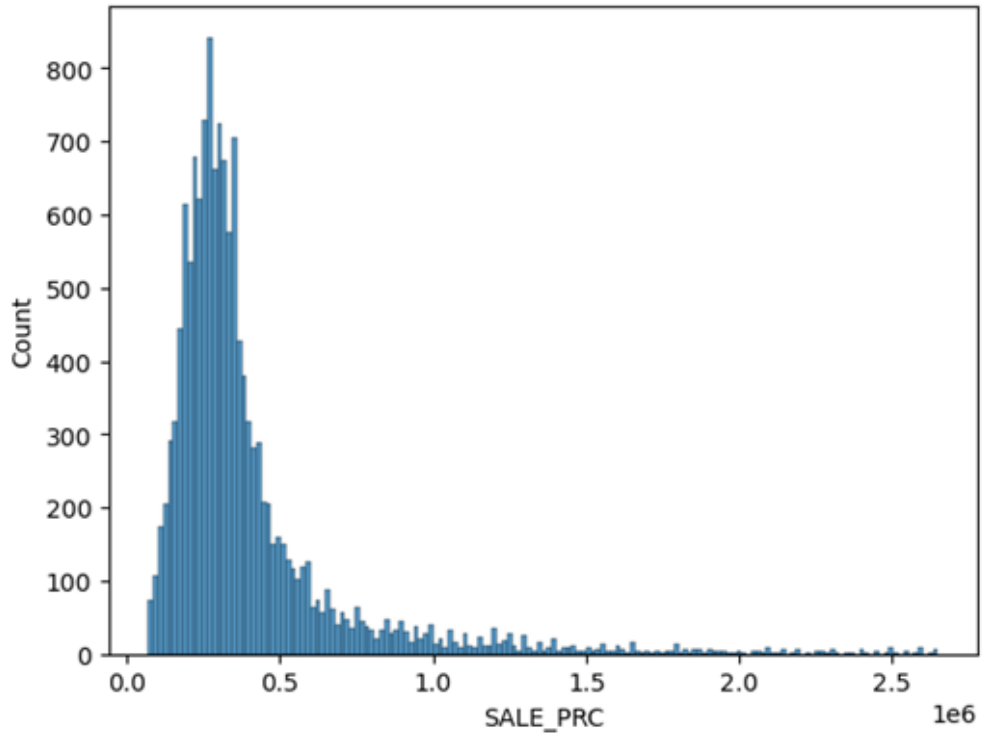
Goal

- **Original Data:** Miami Housing Dataset from Kaggle (original shape: 13,932 rows and 17 columns)
- **Original Fields**: Sale Price, Land Area, Floor Area of Home, Distance to Downtown Miami, Distance to the Ocean, Distance to nearest body of water, age, Distance to nearest Highway, Distance to nearest rail line, Boolean to indicate noise from airports nearby, …
- **Desired Output**: estimate sale price using decision tree regressor

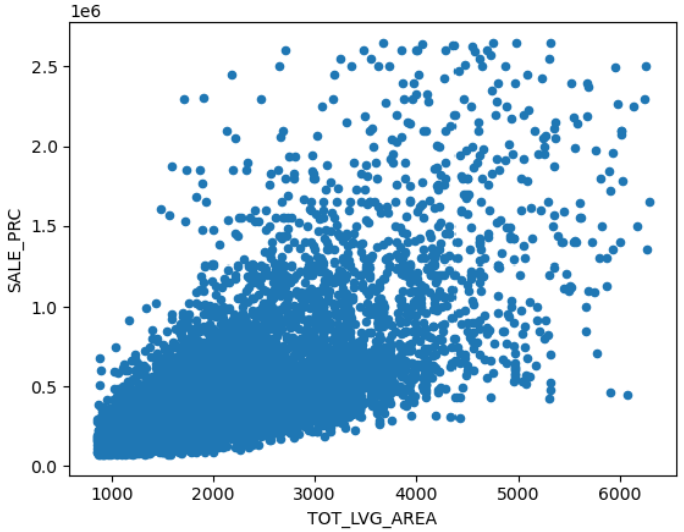
```
[5]: #My understanding is that these imports are needed to create charts and display them:
     import seaborn as sns
     import matplotlib.pyplot as plt

[6]: sns.histplot(data=df, x="SALE_PRC")

[6]: <Axes: xlabel='SALE_PRC', ylabel='Count'>
```



Implementation

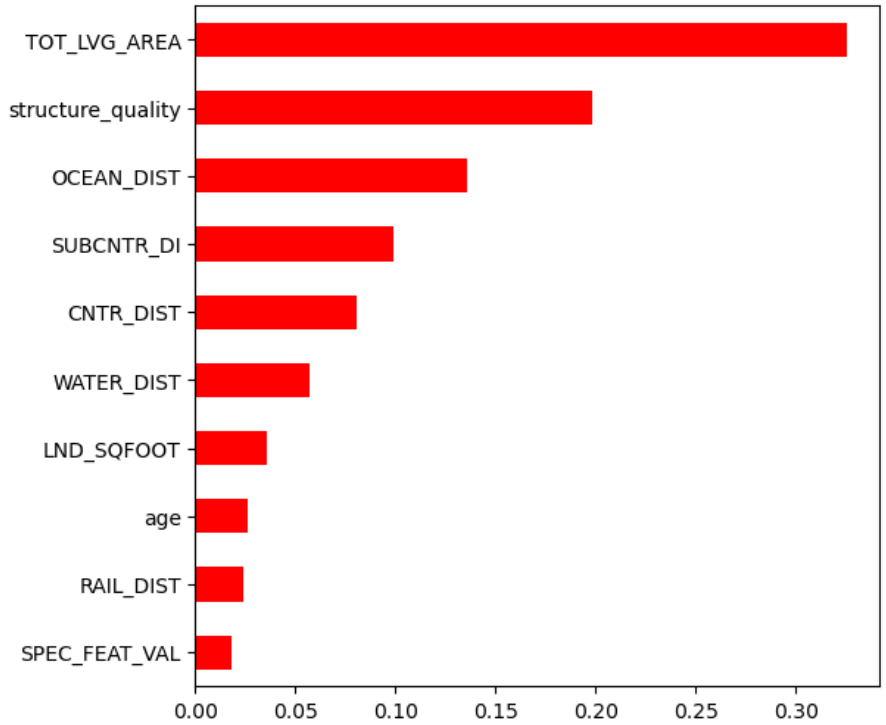


- **Environment**: Jupyter Notebooks, Python 3
- **Data Preprocessing**:
  - Removed redundant and uninformative fields (parcel number, month sold, longitude, latitude, distance to nearest highway).
  - Identified and removed very strange houses (structural quality = 3… see report).
  - Removed outliers (3 SD from mean) and normalized all fields (as done in Truong 2020, which also used decision trees to predict house prices in Beijing) – except Price
- **Final Data Shape**: 12,894 rows, 11 columns
- 80% training, 20% testing split
- Fit Decision Tree Regressor from sklearn on training data
- **Evaluation**: Calculate predictions for testing data, calculate MAE and R-squared

Experimental Results

- Used R-squared and Mean Absolute Error for ease of interpretability.
- Difficult to say whether model was successful since there is no guideline for MAE acceptance. However, R-squared was good, and compared to the standard deviation and mean of prices in the test dataset, I think the MAE was also good.
- R-squared: 0.78
- MAE: $52,870 (mean $347,000; STD $193,900)
- Running Time: training the model took much longer (~10 sec) than calculating the expected prices once the tree had been created (< 3 sec).

Relative Importance of Features in DT Model



Contribution and Future Work

- Difficult to compare my model to Truong 2020's DT models because this team used the RMSLE to compare their models, and they did not specify the units of the RMSLE or explicitly confirm if they standardized house prices.
- The relative importance of features discovered in my project is materially useful.
- I would like to see if I can lower the MAE in future iterations by refining the preprocessing of data or trying alternate regression models.