Classification: Miami Housing

By: Avery Peterson
Date: June 14th, 2022

Abstract

The goal of this project was to produce a classification model to assist a Miami real estate agency in creating a more efficient client management process. Prior, this agency provided equal service and agent skilling to all prospective clients regardless of property value. To boost client retention and improve customer experience for high-valued clients, this agency wants to restructure their service by separating high-valued clients from normal clients and allocating better skilled agents to these high valued clients. This classification model will predict whether a potential client will be a high-valued client (property value above $1 mil) or a normal client based on the properties features.

Design

The data for this project originate from Kaggle's Miami Housing dataset which includes 13,932 single-family homes sold in Miami in 2016. Each entry represents a single sale and includes a list of numeric features such as sales price, land square footage, distance from the ocean, age, structural quality, etc. I added a binary feature that indicates whether a home value is above or below $1 mill (represented by 1 and 0 respectively) and will serve as the target feature for this classification model.

Data

The data includes 13,932 sale entries with 17 features. The target feature (sales greater than $1 mil) represent a minority of 5.12% of the entire data set.

Algorithms

Feature Engineering

1. Standardized the data to represent features as continuous values between 0 and 1. This ensures all features are on the same scale.
2. Create a dummy variable to represent sale prices greater or less than $1 mill (defined as "Target")
3. Upsampled the minority Target data 3 times it's original amount to rebalance the data (now representing a 16% minority)

Models

K-Nearest Neighbors, Logistic Regression, and Random Forest were chosen as my base models before settling with Random Forest as my final model. These models were chosen based on their relatives simplicity and interpretability. Random forest's feature importance tool was used to determine the least and most important features in the model refinement process.

Model Evaluation and Selection

The entire data was split into training, validation, and test sets (60%, 20%, 20% respectively) where the models were trained on the training set and evaluated on the validation set. The test set was saved for the end of the model evaluation to assess overall performance.

I opted to use the F1 score as my evaluation metric as it assesses overall performance (though precision and recall were relatively similar across all models). The random forest model produced the highest initial F1 score, so I tuned the hyper parameters and dropped the insignificant features to increase the F1 score and model performance.

Validation Scores

- Precision: 0.809
- Recall: 0.7986
- F1: 0.8040

Final Test Scores

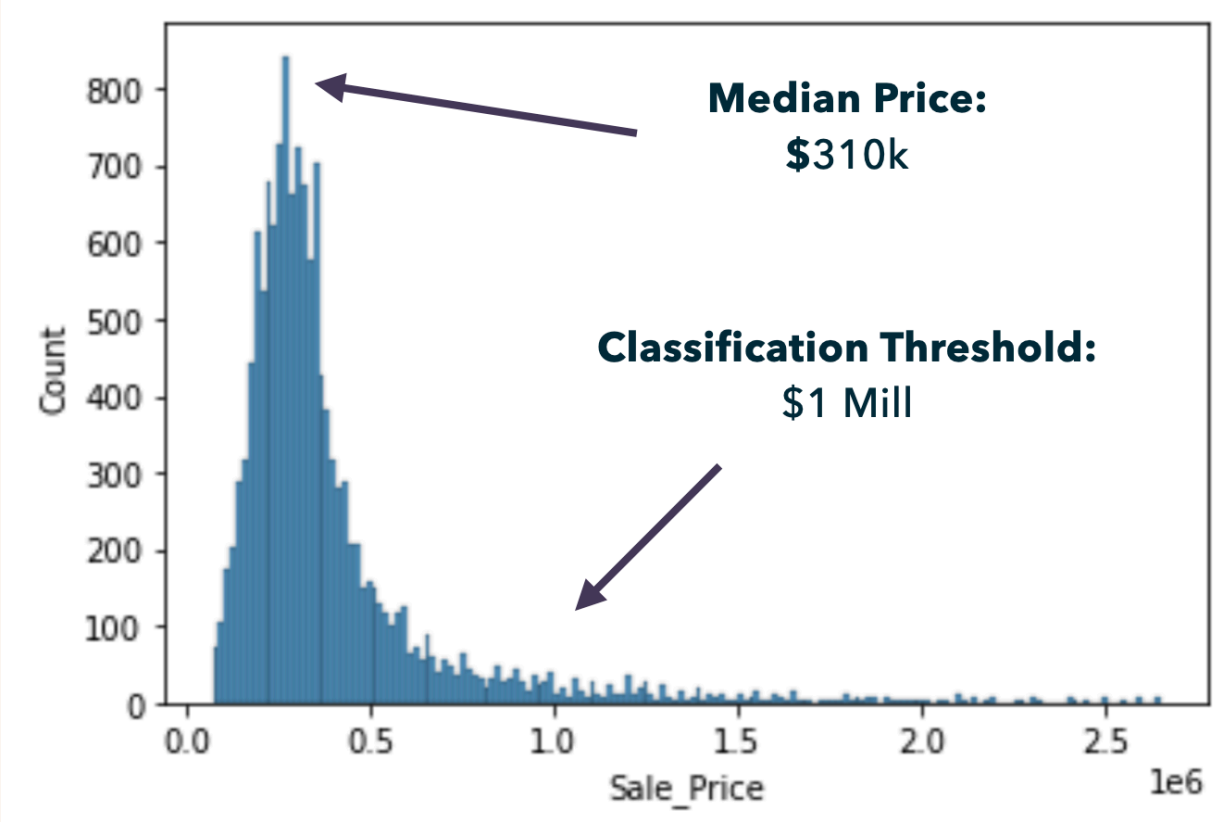
- Precision: 0.8356
- Recall: 0.8133
- F1: 0.8243

Tools

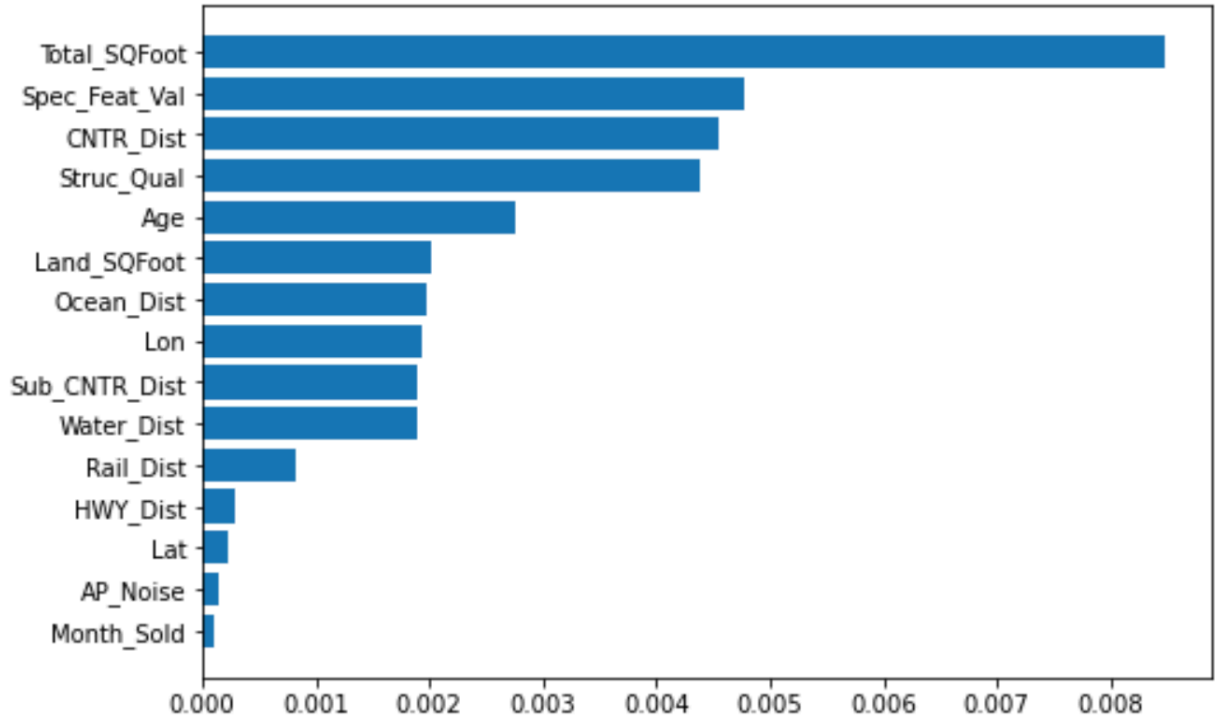
- Numpy and Pandas for data manipulation
- Scikit-learn for modeling
- Matplotlib for plotting

Communications

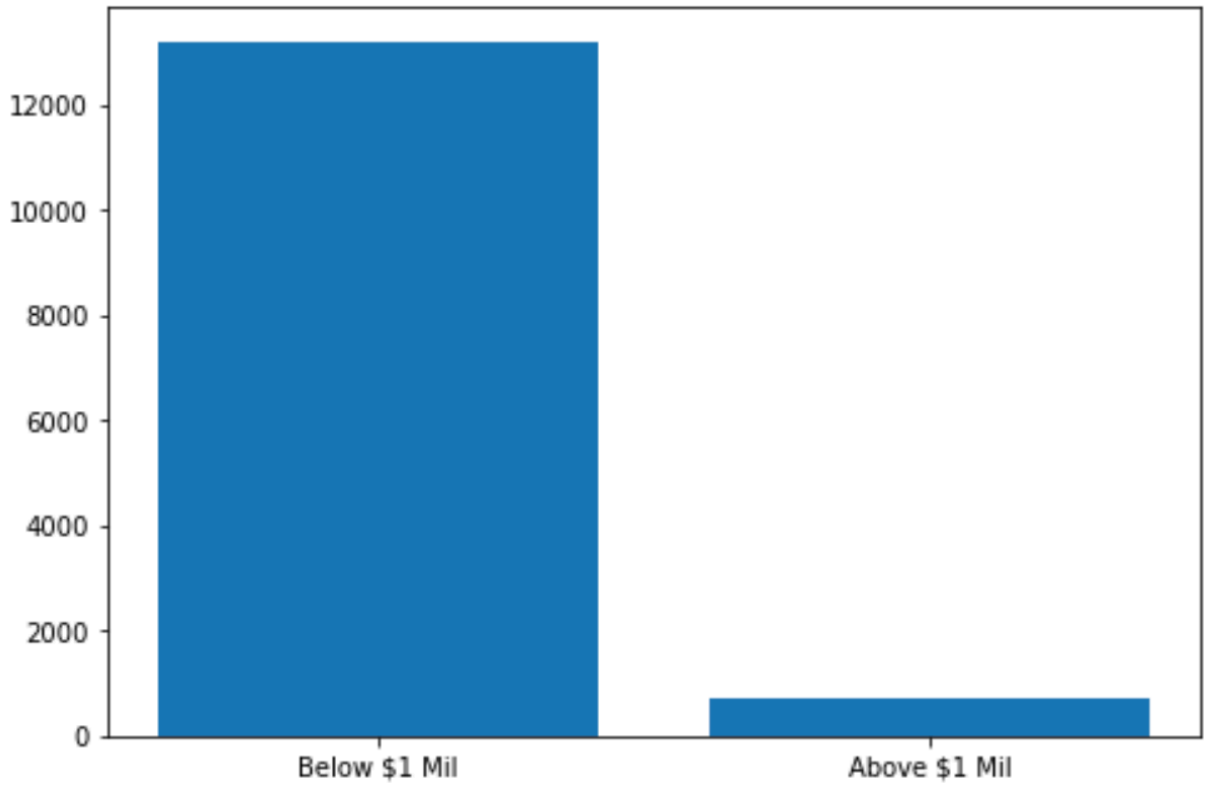
Sales Prices Distribution



Feature Importance



Target Data Imbalance



Upsampled Data