**Project Write-up: Houses Sold Over List Price Prediction**

*By: Salma Katri*

**Abstract**

The goal of this project is to predict whether a house will sell for over the list price using available Redfin listing data for houses sold in the Seattle area. The over/under ask is calculated as the difference between the list price and final sale price. The binary variable for our classification is whether a house is sold over or at/under list price. This aims at helping home buyers to better understand features affecting sale price, plan according to their budget and strategize when placing their offers. As Redfin estimate was also used as an independent variable, the model informs on how accurate this estimate is.

**Design**

The data for this project was scraped from Redfin available sold homes data in the Seattle area. All variables used in the model are obtained directly or derived from the Redfin scraped data. The model aims to predict whether the final sale price will be higher using all available Redfin data to simulate homebuying decision making.

**Data**

The dataset scraped from Redfin consisted of 2530 rows representing the most houses sold in Seattle, Kirkland and Bellevue. The data spanned mainly October and September 2021 with few entries from August and November. 24 different variables were obtained from Redfin for each house. Some variables were dropped due to many missing values, ending up with 16 variables that were considered for further exploration in the model. Most important features included list price, redfin estimate, sqft, number of times house was favorited and days on the market.

**Algorithms**

*Feature Engineering*

* Selecting variables with significant impact on model
* Creating more meaningful variables out of available ones such as age, renovation status, days on the market and average school ratings, difference between Redfin estimate and list price
* Converting categorical features into dummy variables and creating groups for neighborhoods and house style to reduce number of features

*Models*

* Logistic regression
* KNN
* Random Forest
* XGBoost

In tuning my model, I wanted to prioritize recall, so I would rather have houses sold at or under list price to be predicted to be sold over rather than the other way around. The reason is from a buyer perspective, if they think a house might sell over, they can add an escalation cost and will still pay the list price if no competing offer is present, but if their initial offer is lower than list price or they don’t include an escalation cost, they risk of being outbid and losing the house,

I considered different features and chose to tune the parameters for random forest model using Grid search since it performed better compared to other baseline models. Final model had a recall of 0.956, precision of 0.916, accuracy of 0.92 and F1 score of 0.936.

**Tools**

Numpy and Pandas for data manipulation

Matplotlib and Tableau for visualization

Sklearn, XGBoost for modeling

**Communication**

In addition to the presentation slides, I also plan on further developing a tableau dashboard for visualization