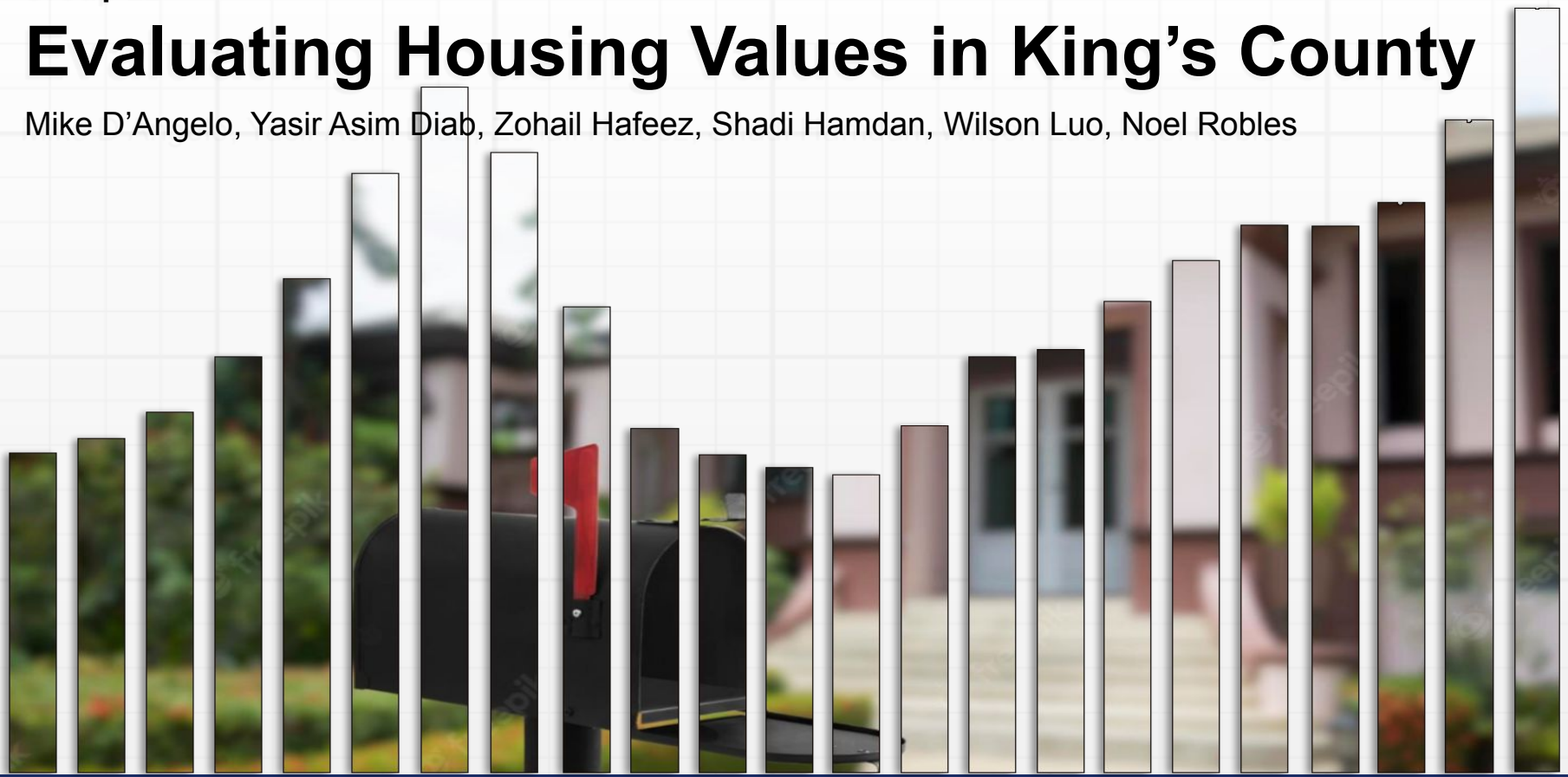
Group 22

Evaluating Housing Values in King's County

Mike D'Angelo, Yasir Asim Diab, Zohail Hafeez, Shadi Hamdan, Wilson Luo, Noel Robles



Introduction:

- The average American has **70%** of their net worth tied into a primary residence.
- In the last decade, the U.S. housing market has doubled to over **$43 trillion.**
- Buying a home is important, so there should be **adequate transparency** when evaluating real estate.
- We **hypothesize** that **square foot** and **location** affect home value the **strongest**.
- What affects the value of a house?
    - Number of Bedrooms & Bathrooms
    - Year Built
    - Location
    - Size
    - And Many More!

Introduction:

Problem:

- **Lack of transparency** in the evaluation of real estate, especially for **buyers**.
- Lenders will often **not move forward** with a transaction if the **appraisal is less than the selling price**. Therefore, it is important that buyers receive accurate pricing on their properties.
- Current solutions, like **Zillow's "Zestimate"** generally do not factor in the current condition of the property, and can accept inaccurate seller data which may lead to **inflated prices**.

REDFIN

StreetEasy

Zillow

Introduction:

Solution:

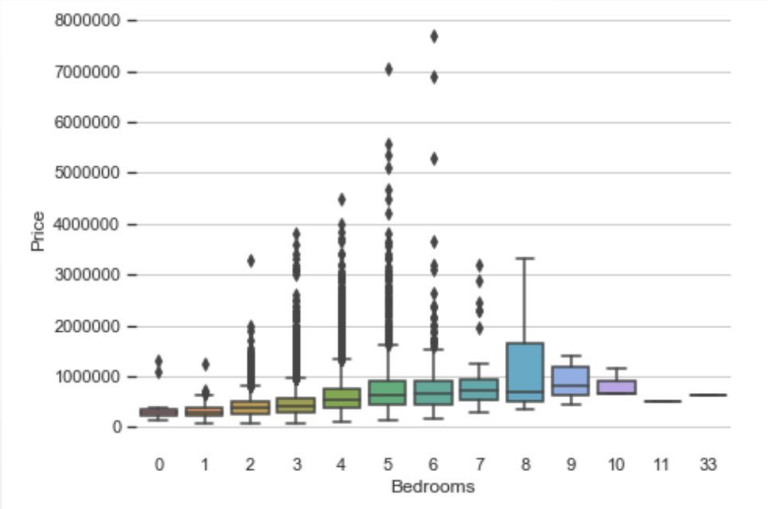
- We aim to provide a **model that allows users to input features associated with a specific property and receive a reasonable evaluation** based on similar homes in the area. This should help **provide transparency** to potential homebuyers.

Methods: Dataset Selection

- Looking at dataset of housing prices in King's County, Washington
- 21,600 Samples
- 19 Features
- No missing feature data

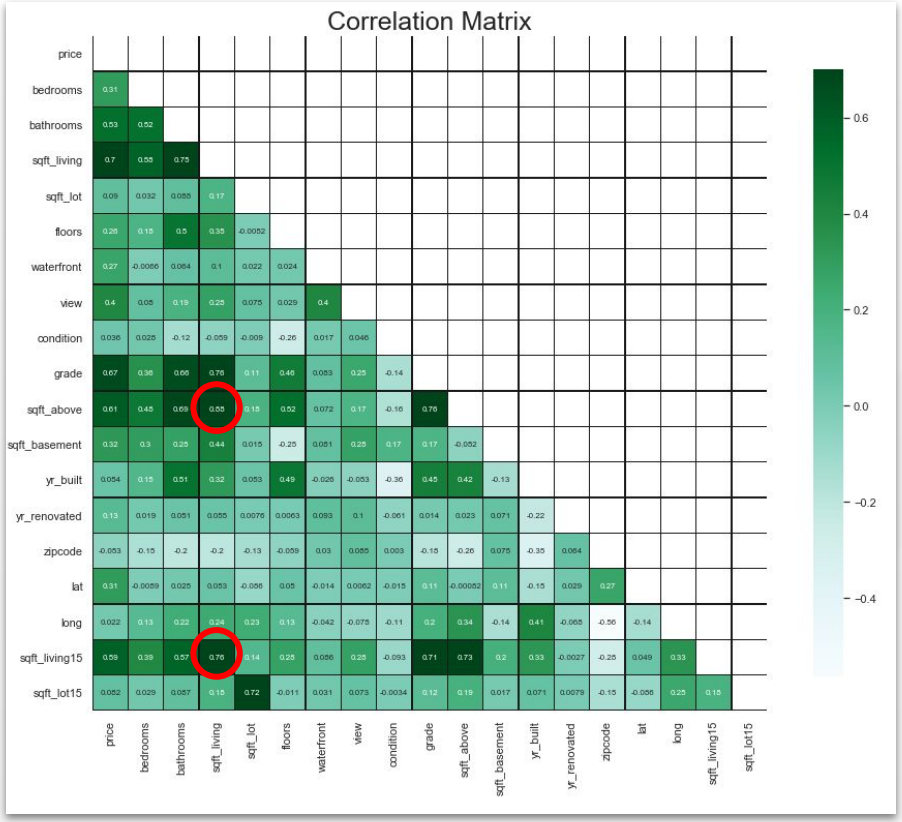
	Total Null	% of Total	Data Type
id	0	0.00	int64
date	0	0.00	object
age_rnv_binned_26-50	0	0.00	uint8
age_rnv_binned_1-25	0	0.00	uint8
age_rnv_binned_less_than_1	0	0.00	uint8
age_binned_51_plus	0	0.00	uint8
age_binned_26-50	0	0.00	uint8
age_binned_1-25	0	0.00	uint8
age_binned_less_than_1	0	0.00	uint8
age_rnv	0	0.00	float64
age	0	0.00	int64
sales_yr	0	0.00	object
sqft_lot15	0	0.00	int64
sqft_living15	0	0.00	int64
long	0	0.00	float64
lat	0	0.00	float64
zipcode	0	0.00	int64
yr_renovated	0	0.00	int64
yr_built	0	0.00	int64
sqft_basement	0	0.00	int64
sqft_above	0	0.00	int64
grade	0	0.00	int64
condition	0	0.00	int64
view	0	0.00	int64
waterfront	0	0.00	int64
floors	0	0.00	float64

Methods: Analysis & Feature Engineering



Skewing to the right with an outlier of $770,000, which we expect from a good housing dataset since there are luxury houses included.

Methods: Analysis & Feature Engineering



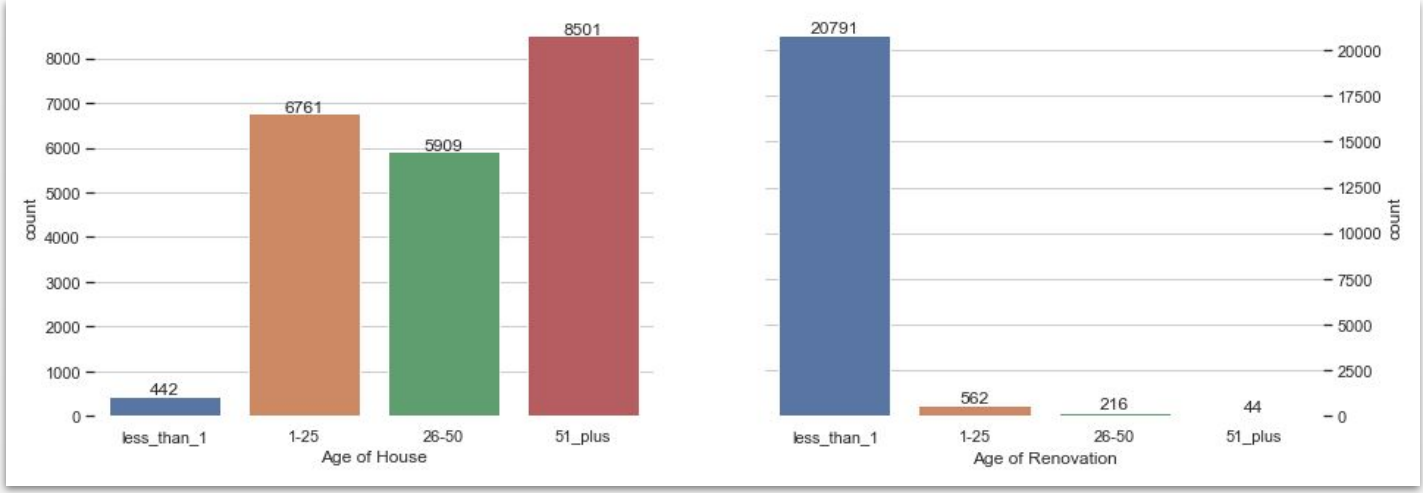
For high correlation to sqft_living, following features removed:

1- Sqft_above

2 - Sqft_living_15

Methods: Analysis & Feature Engineering - Binning

- Creating two new features and binned them:
  1. **Age of house** which is based off of year built.
  2. **Renovation** Age based on renovation year.
  3. Determined **quartile partitions** to be best.



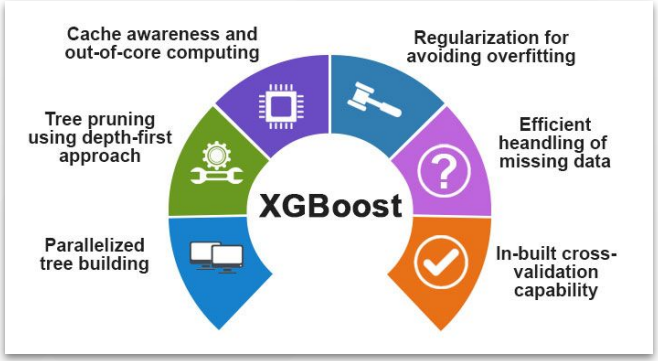
Methods: Model Comparisons - Tuning

Random Forest vs XG Boost - Model Selection Rationale

- Random Forest Regression
  - Randomly generates decision trees
  - Output is average of predictions

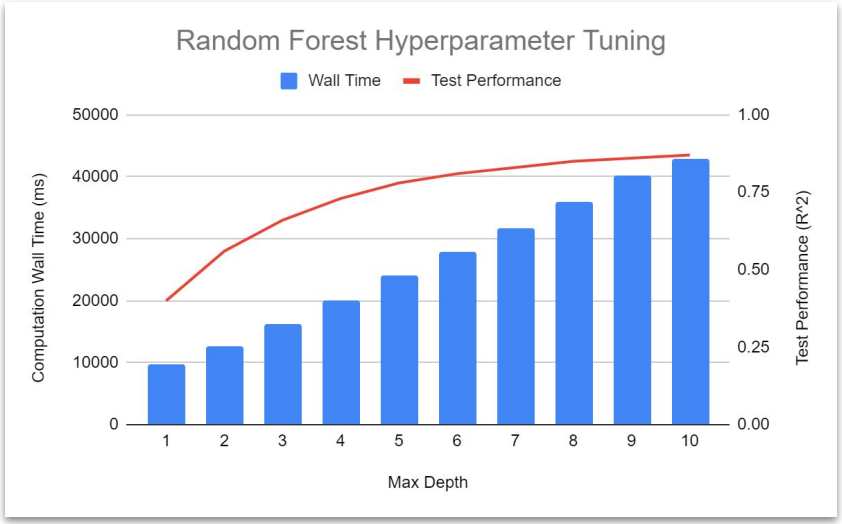
<u>Max Depth</u>
Hyperparameter tuned to prevent overfitting

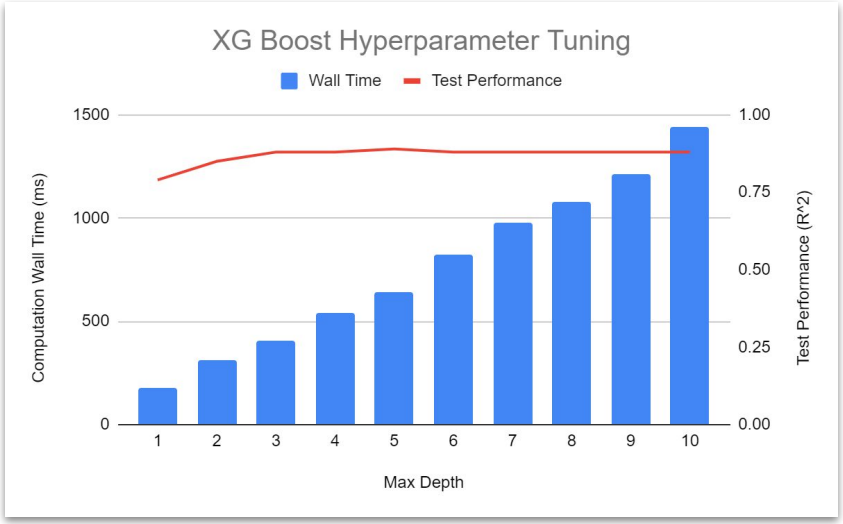
- XG Boost - Implementation of Gradient Boost
  - Evaluates results of each decision tree as they are generated
  - Unlike Random Forest, each new tree generated minimizes the gradient of the loss function



Methods: Model Comparisons

Random Forest vs XG Boost - Model Selection Rationale

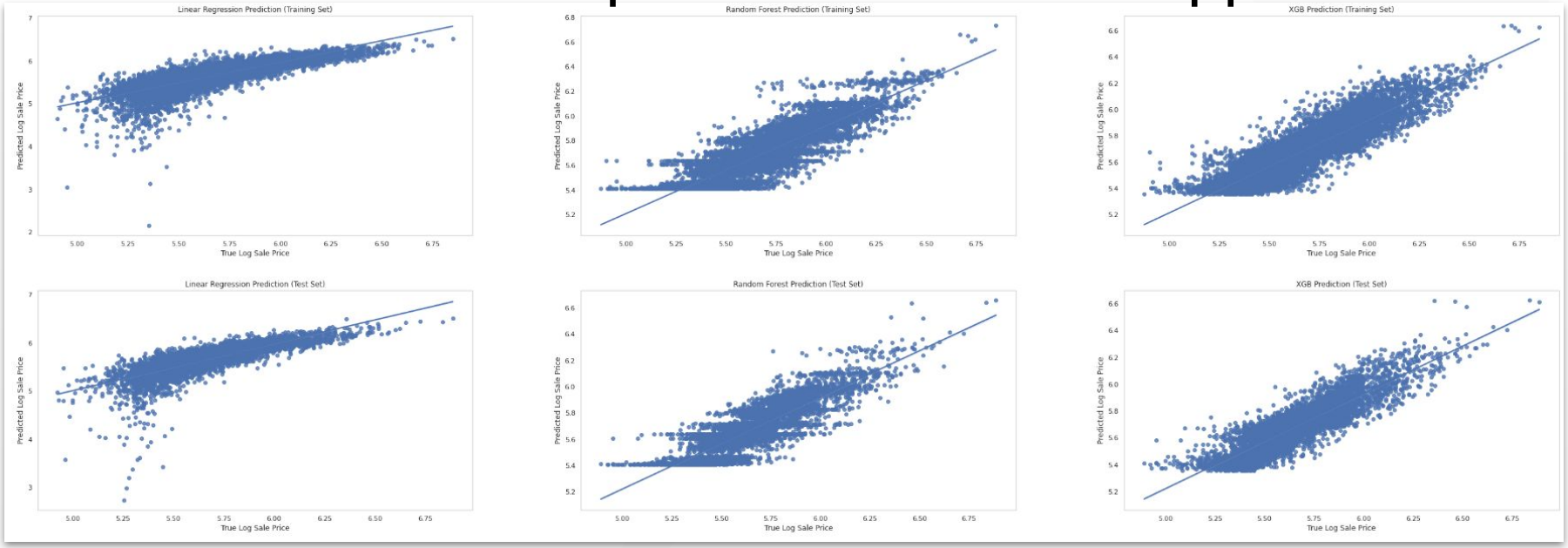




NOTE VERTICAL SCALE

XG Boost reduces wall time by greater than an order of magnitude

Results: Model Comparisons/Alternate Approaches



Model 1: Linear Regression
Performance on train set: 0.71
Performance on test set: 0.70
Speed: 18.5 ms

Model 2: Random Forest
Performance on train set: 0.76
Performance on test set: 0.73
Speed: 20.9 s

Model 3: XGBoost
Performance on train set: 0.89
Performance on test set: 0.86
Speed 337 ms

Results: Model Comparisons/Alternate Approaches

- Accuracy (Training/Test)
  - Linear Regression: 0.71/0.7
  - Random Forest: 0.76/0.73
  - XGBoost: 0.89/0.86
- Speed
  - Linear Regression: 18.5 ms
  - Random Forest: 20.9 s
  - XGBoost: 337 ms
- Data Partition
  - Train Set: 70%
  - Test Set: 30%
- All three models provide accurate results (R2 > 0.7)
- Training and Test Accuracy results are similar - models are not overfitted
- Linear Regression has the fastest computation time but least accurate
- XGBoost is the most accurate model - slower than linear regression but much faster than random forest

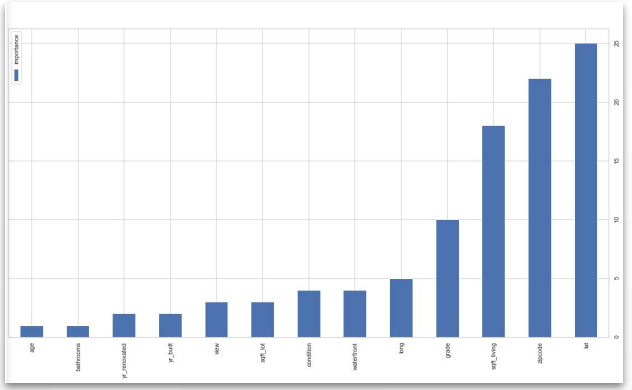
Results & Discussion:

Potential Problems:

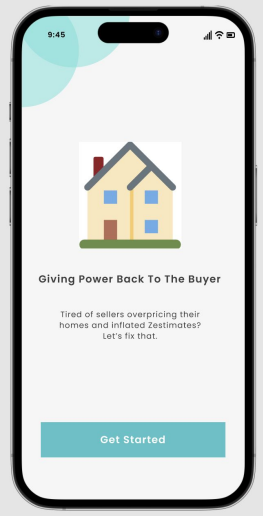
- Dataset does **not** include **supply** and **interest rates.**
- Data has sale prices from 2015, would need more recent data to be applicable to 2022, as just applying average 7.7% appreciation to 2015 price, would not account for rapid appreciation and the migration into rural suburbs seen post-covid.

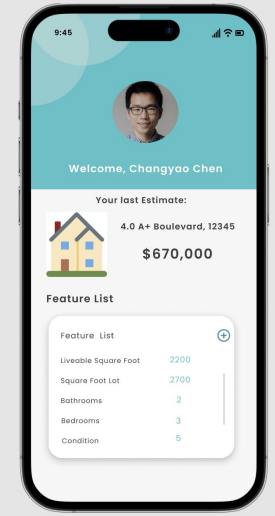
Future Extensions, Actionable Results & New/Alternate Approaches:

- **Latitude/zip code** is shown to be the strongest feature impacting price. We intend to create a emerging neighborhood feature that tracks number of businesses nearby and median income, as neighborhoods are rapidly changing, a once undesirable zip code can be “up and coming” (I.E Williamsburg/Bushwick).
- Expand dataset to include more locations, as the dataset is specific to King County.
- Factor in features such as supply & interest rates.
- Build out user friendly application (FIGMA prototype shown).



Feature Importance from XGBoost





Any Questions?

Sources:

- https://yourorlando.com/buying-a-home-in-the-orlando-florida-area
- https://www.nerdwallet.com/article/mortgages/fed-mortgage-rates
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.cut.html
- https://towardsdatascience.com/data-preprocessing-with-python-pandas-part-5-binning-c5bd5fd1b950
- https://www.creditkarma.com/home-loans/i/average-home-value-increase-per-year#:~:text=Since%201991%2C%20the%20average%20annual,significantly%20from%20state%20to%20state.
- https://xgboost.readthedocs.io/en/stable/tutorials/model.html