Seattle Washington Housing Market 2018 - 2022

Team # 1 - MACK 2.0
Nathan Karr, Ivy Jones, Caroline Sudhakar, Kristen Pollok, Allan Ivey

Real Estate Price Analysis -Seattle, Washington

- Previous project focused on the Austin housing market - data only covered 2021
- Found Seattle dataset from Kaggle - dataset covers 1999 - 2022
- **Goal:**
  - **Analyze single family home prices in Seattle**
  - **Generate model with R^2 fit greater than 65%**
  - **Achieved 68% or 80% dependant on model**
- **Overview:**
  - **Study/clean data**
  - **Generate SQL database**
  - **Pull in data to flask app with new Seattle data**
  - **Analyse with ML models**
  - **Generate graphics using Tableau**

Data Cleaning & Database

- Study the data to understand how to define a “single family home”
  - Seattle area is defined by zoning that govern property development for single family homes
  - Define zoning codes
- Used Lat/Long data to generate zipcode using Geopy function

```python
def get_zipcode(df_test, geolocator, latitude, longitude):
    location = geolocator.reverse((df_test['latitude'], df_test['longitude']))
    return location.raw['address']['postcode']

geolocator = geopy.Nominatim(user_agent="4321")

zipcodes = df_test.apply(get_zipcode, axis=1, geolocator=geolocator, latitude='Lat', longitude='Lon')
```

- Generate schema to build database tables

```
57
58  CREATE TABLE monthly_seattle_sales(
59      id serial PRIMARY KEY,
60      Years INT,
61      Months INT,
62      Average_Sale_Price FLOAT,
63      Median_Sale_Price FLOAT,
64      Total_Houses_sold INT,
65      interest_rate FLOAT
66  );
```

Query | Query History

```
1   --DROP TABLE IF EXISTS seattle_sales
2
3   CREATE TABLE seattle_sales(
4       id INT PRIMARY KEY,
5       sale_id VARCHAR(50) NOT NULL,
6       pinx VARCHAR(255),
7       sale_date DATE,
8       sale_price FLOAT NOT NULL,
9       sale_nbr VARCHAR(50),
10      sale_warning VARCHAR(255),
11      join_status VARCHAR(50),
12      join_year INT,
13      latitude FLOAT,
14      longitude FLOAT,
15      area INT,
16      city VARCHAR(100),
17      zoning VARCHAR(50),
18      subdivision VARCHAR(100),
19      present_use INT,
20      land_val FLOAT,
21      imp_val FLOAT,
22      year_built INT,
23      year_reno INT,
24      sqft_lot INT,
25      sqft INT,
```

Machine Learning Model

- Took in the cleaned data and trimmed it down a little more
- Used value_counts() and describe() to find columns that had little relevant data or a low amount of non zeroes like 'garb_sqft''gara_sqft', etc
- Converted alphabetical columns like Zoning to numerical values
- First intended to use linear regression but data was too complicated to draw linear relations, got R^2 scores of .3-.4
- Moved to random forest with much better results, R^2 began to be in the .6s
- R^2 was getting pretty good but the Root Mean Squared Error(RMSE) was too high to be as useful to a human, with model being off by as much as 300,000
- Tried using only rows with sale prices less than and greater than $1,000,000
- Worse R^2 but better root mean squared error

Machine Learning Model

- Tried to bucket further, but the data source was too small for certain values
- Final Algorithm used was Gradient Boosting
- Similar to random forest except that the trees are trained one after the other rather than independently
- Each new tree tries to fix the errors of the last
- Using gradient boost with bucketing got us a decent R^2 of .682 along with the more reasonable root mean squared error of $94,000

	Actual Price	Predicted Price	Absolute Difference
0	879000.00	943537.87	64537.87
1	892500.00	843567.28	48932.72
2	675000.00	783233.13	108233.13
3	940000.00	887688.78	52311.22
4	899999.00	893442.92	6556.08
...	...	...	...
3054	369000.00	472114.27	103114.27
3055	745000.00	759528.36	14528.36
3056	918000.00	874475.26	43524.74
3057	622500.00	603481.06	19018.94
3058	830000.00	734844.31	95155.69

[3059 rows x 3 columns]
Root Mean Squared Error: 94005.49
R-squared: 0.6824328980247552

Machine Learning Model

- Finally, did new gradient boost for whole data
- Reduced analysis to 2020 - 2022
- Very good R^2
- Not as ‘user-friendly’ due to high range of error
- Appears to do better due to larger amount of rows with gradient boosting

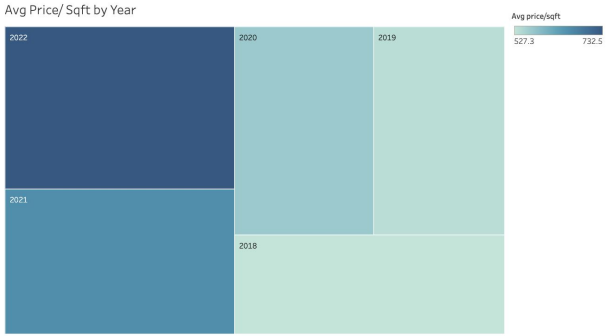
	Actual Price	Predicted Price	Absolute Difference
13329	810000	672568.19	137431.81
16534	600000	596047.38	3952.62
20185	875000	991144.81	116144.81
17441	1015000	842392.89	172607.11
223	993000	954858.33	38141.67
...	...	...	...
10731	1015000	1139517.30	124517.30
18017	1655000	1521980.76	133019.24
24521	1550000	1645759.63	95759.63
2566	820000	1120768.38	300768.38
22767	1165000	1083553.58	81446.42

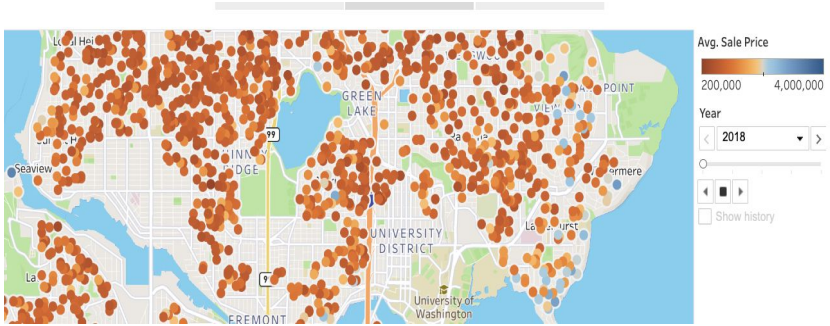
[3146 rows x 3 columns]
Root Mean Squared Error: 308869.92
R-squared: 0.8042023263816688

Visualizations and Map

The visualizations we created through Tableau:

- Average Price/Sqft by Year Visualization
- Square Footage of Home Graph
- Sales Price Distribution Graph
- Zip Codes by Average Price/sqft Graph
- Average Sales Price Map
- Zip Codes by Average Price/sqft Map





Task List

Tasks for Final project

- Clean data for ML model
  - 'seattle_sales_cleaned_data'
  - Mostly cleaned
- **Caroline/Nathan -** Build ML models multiple linear regression model
  - Visualizations for ML models
  - Linear Regression
  - Predictions
  - Add report ML analysis into the readMe file
- **Kristen Pollok/Allan -** Make HTML "fancy"
  - Flask api
  - Database revisions
  - JS/HTML
- **Ivy Jones -** Tableau generated plots
  - Dashboard w/ 3 different visualizations
  - layers/pages for years 2018-2022
  - Maps using lat/long data

- **All -** Create google slides for presentation

Comments

- City: Seattle
- Years 2018 - 2022
- Home Type: Single Family Home