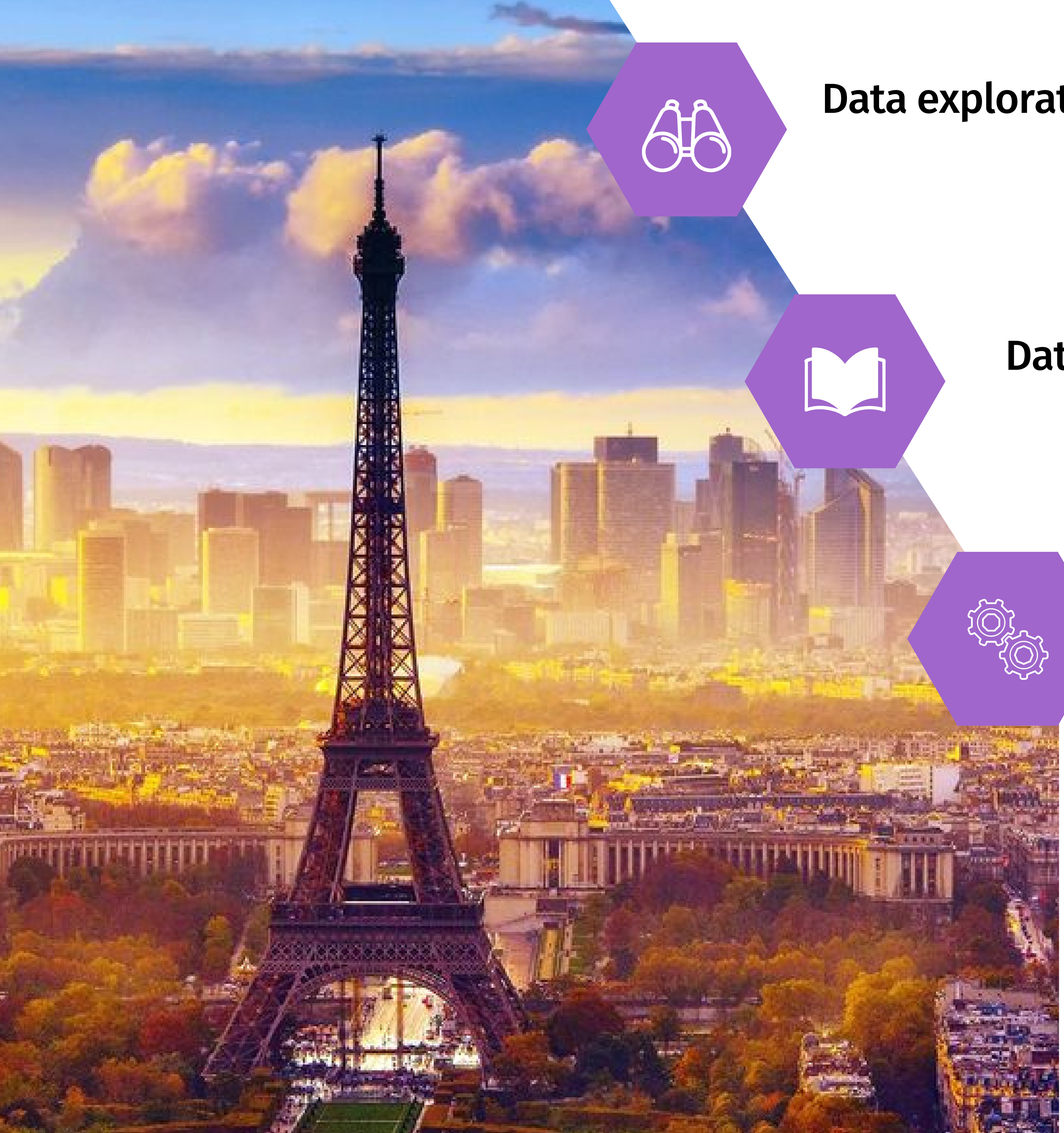
Elliot & Chi

Mid Project Presentation

Subject : Housing Price Prediction Model



Data exploration

Data visualization

Training the model

Analyzing the results

Presentation of the Dataset

Context

Our dataset is composed of :

- **houses and their caracteristics**
- **Their selling prices**

Where ?

United States
Washington state
King's county

Objective ?

To define a model that can predict the selling price.

Some key elements

The average price in King's county to buy a house.

540 296 $
--> Not very expensive if we compare some other county that are near the ocean

Luxury houses in the county

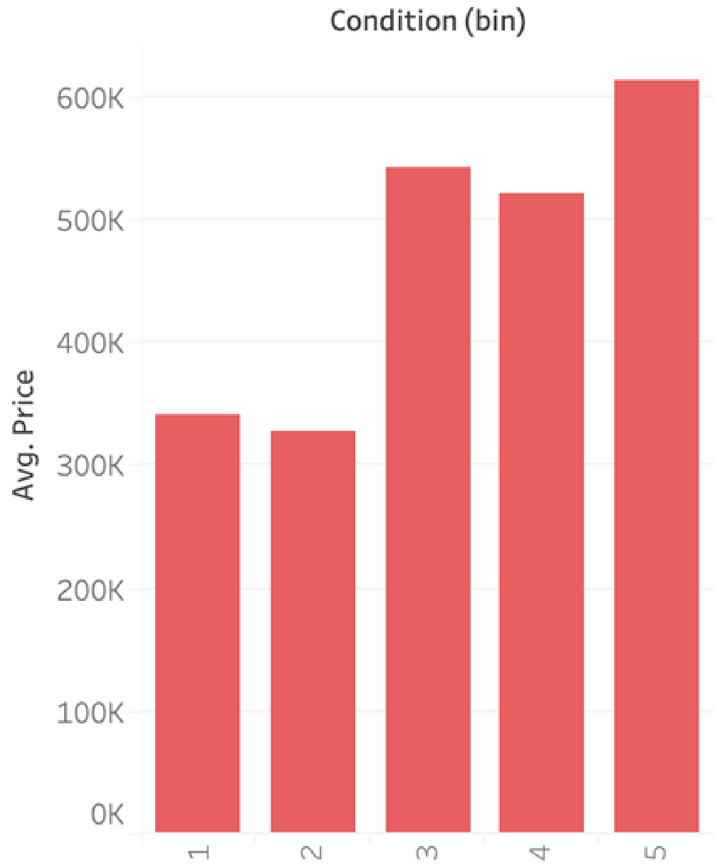
We have 1490 houses with a price that is above 1 million $ which represent only 6% of our data

Recent houses ?

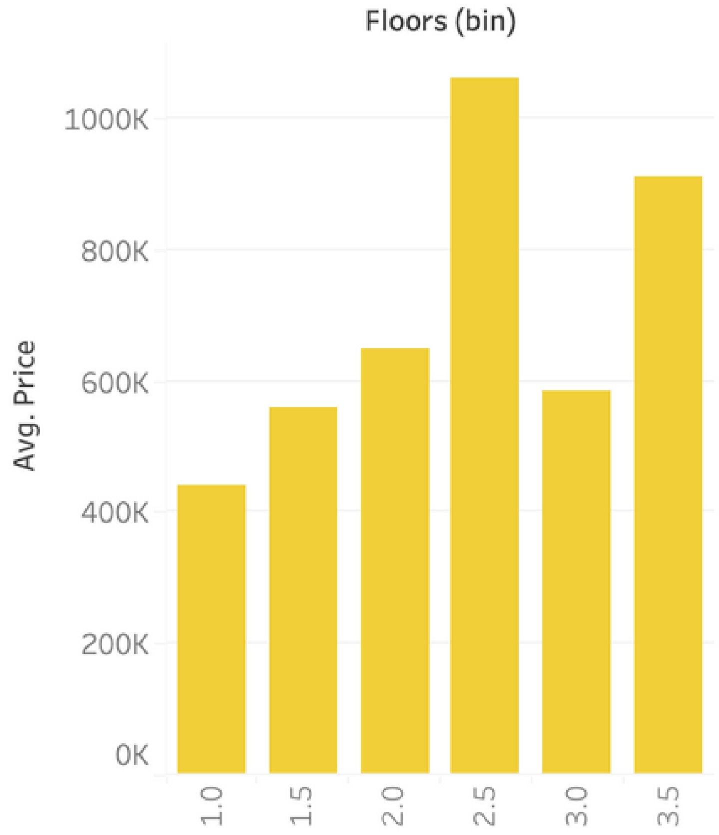
Most of our houses were built around the 70s. Something to take into consideration is that we only have approximatively 4700 houses that were built in the second millennium.
--> Not a lot of construction, does the county still attractive ?

Different figures...

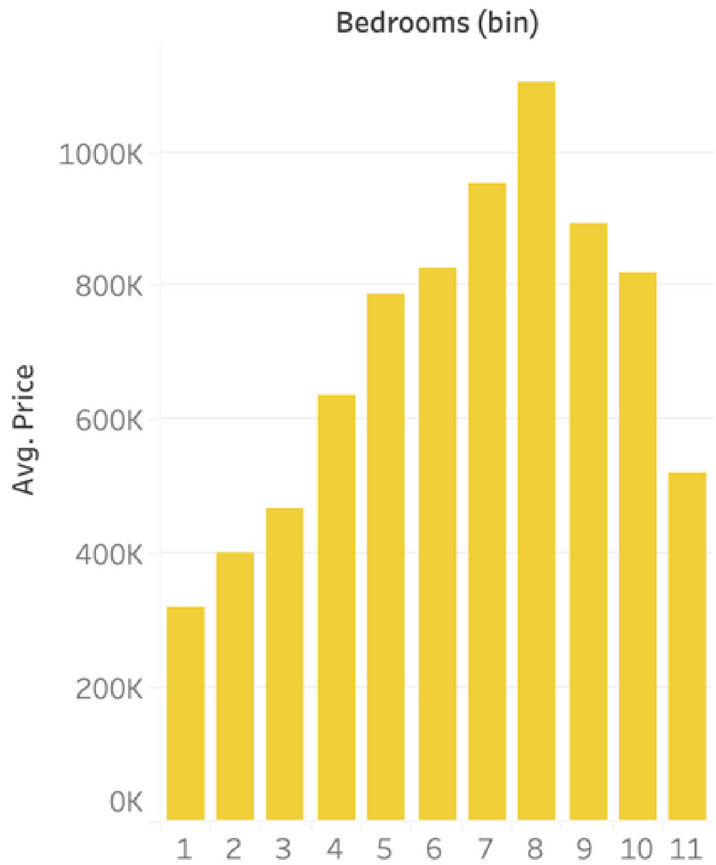
Price vs. Condition



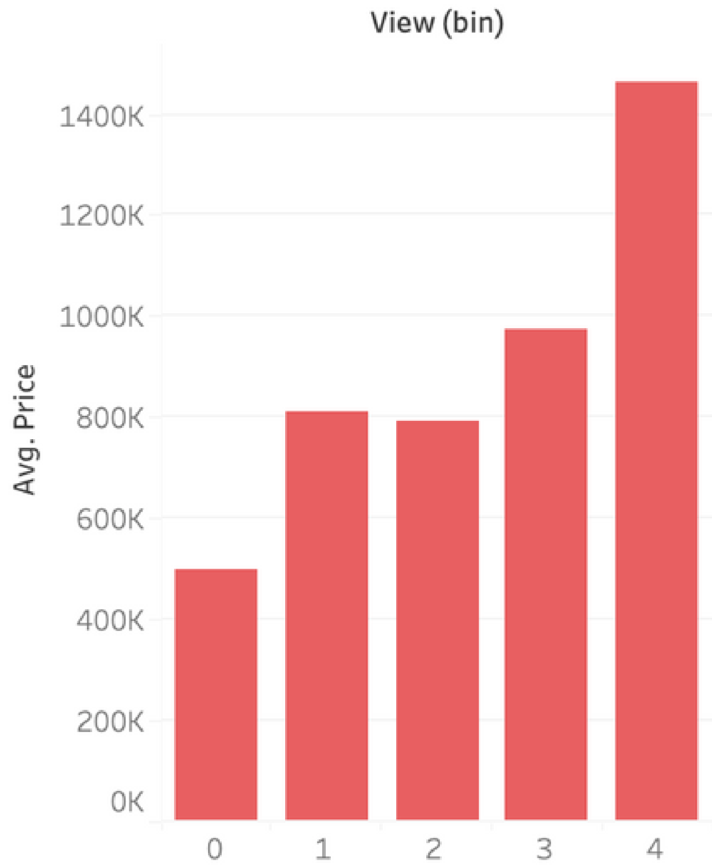
Price vs. Floors



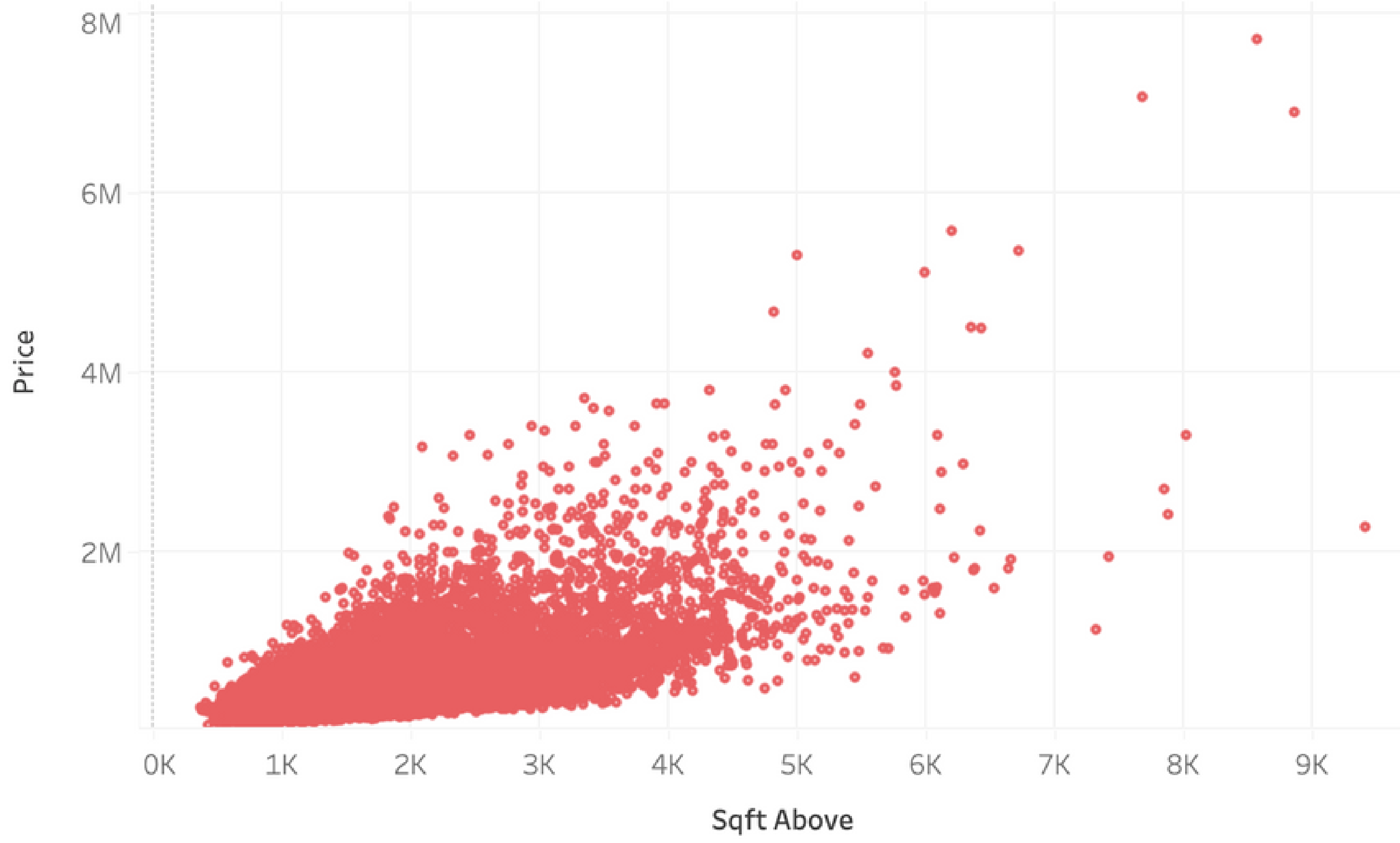
Price vs Bedrooms



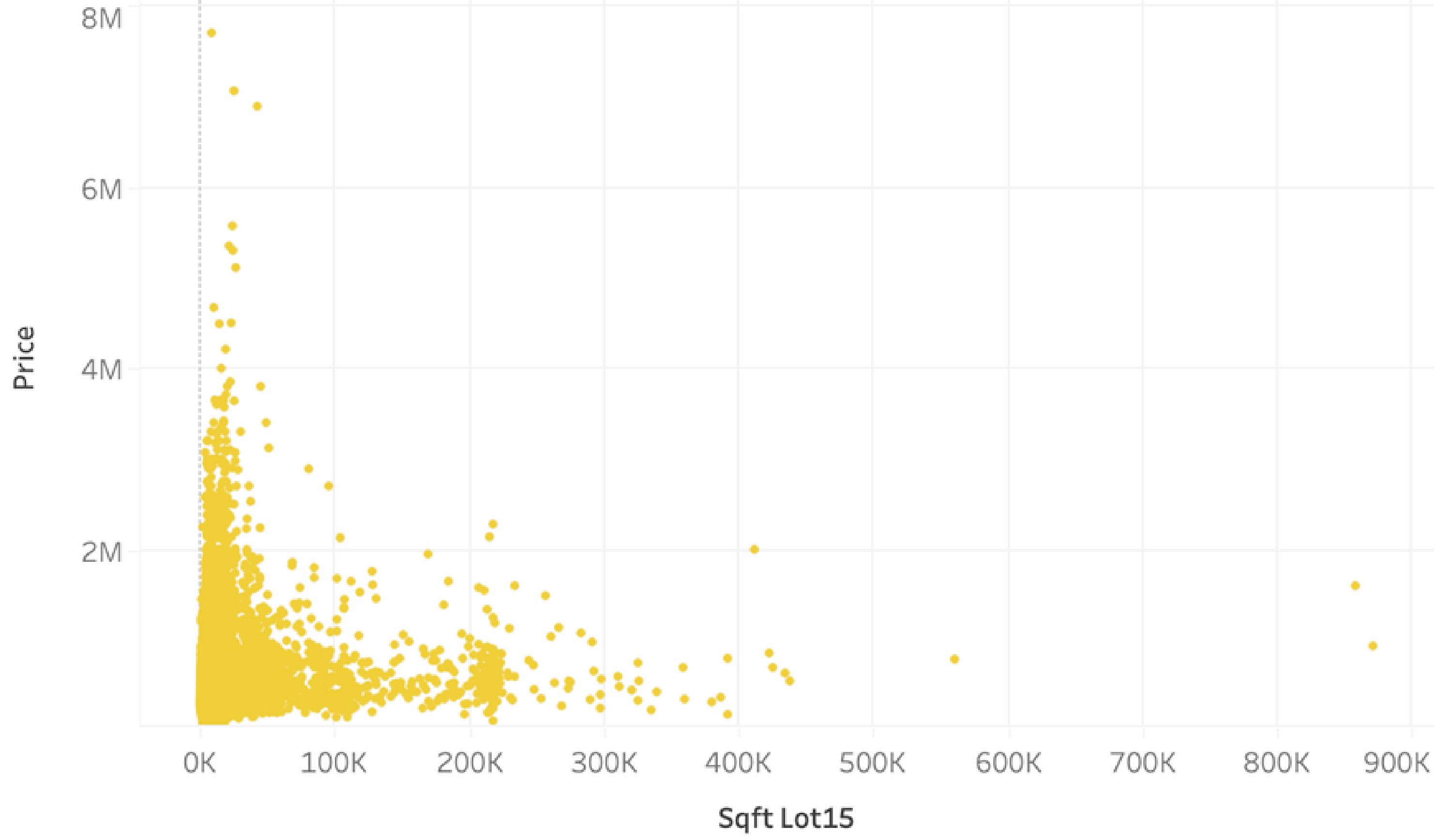
Price vs. View



Price vs. Sqft_above



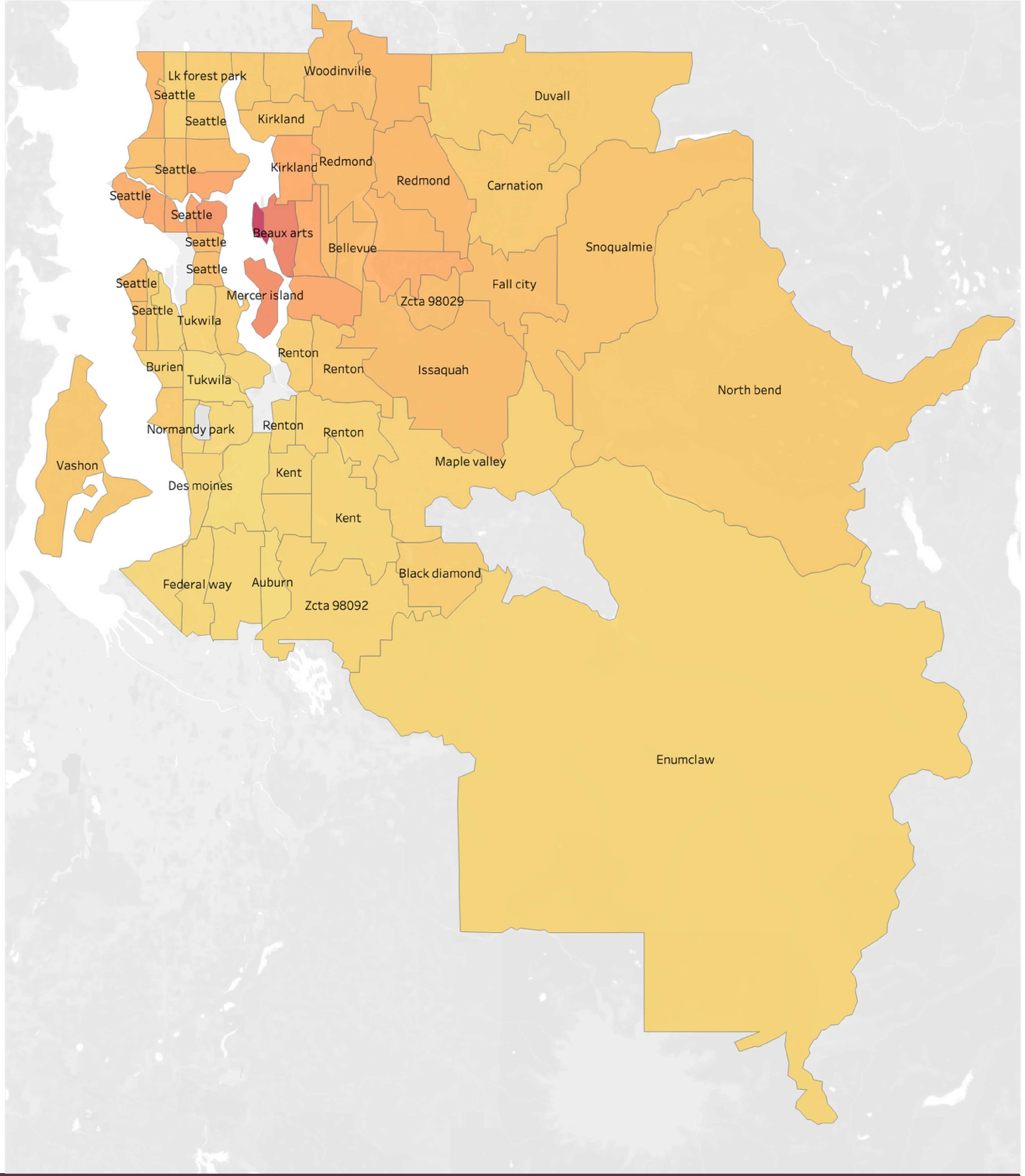
Price vs. Sqft_lot15



The map

- **Higher prices on the coast in general**
- **No significant differences between the regions**
- **North part seems richer than the south Part**

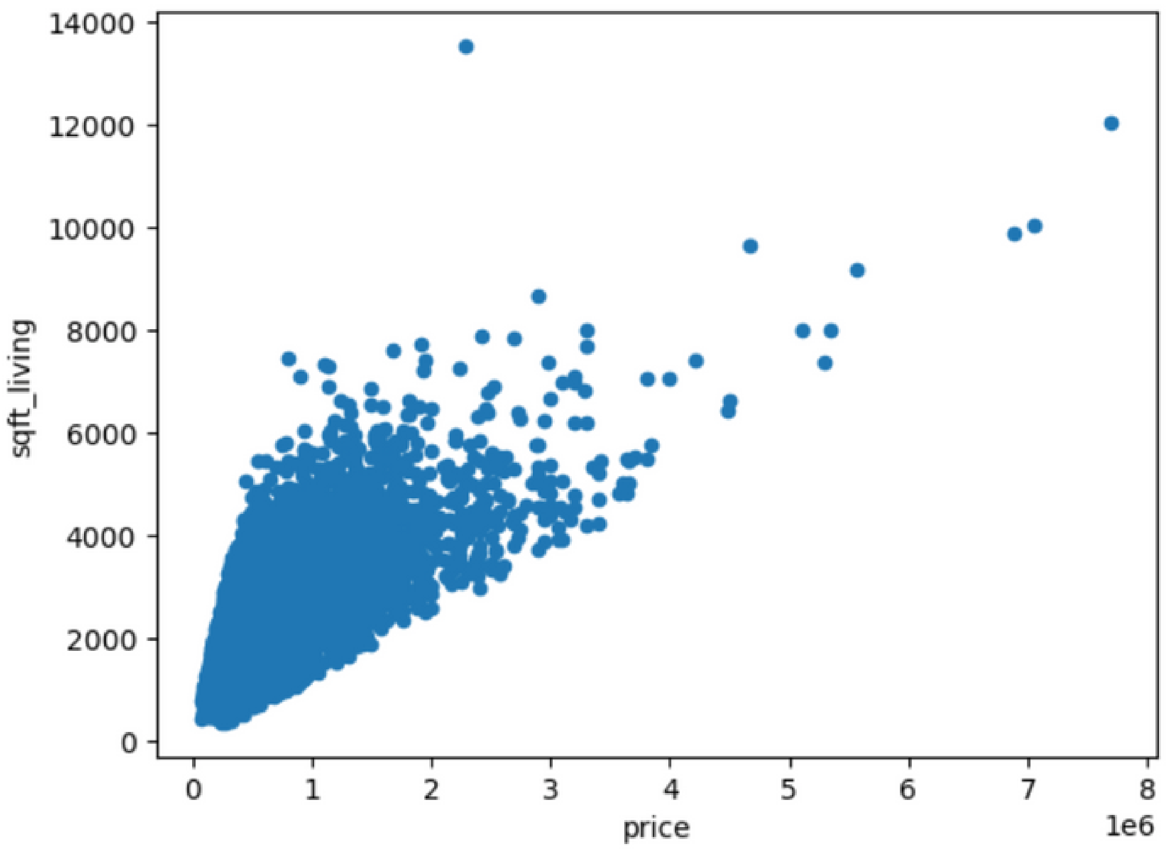
Price vs. Location

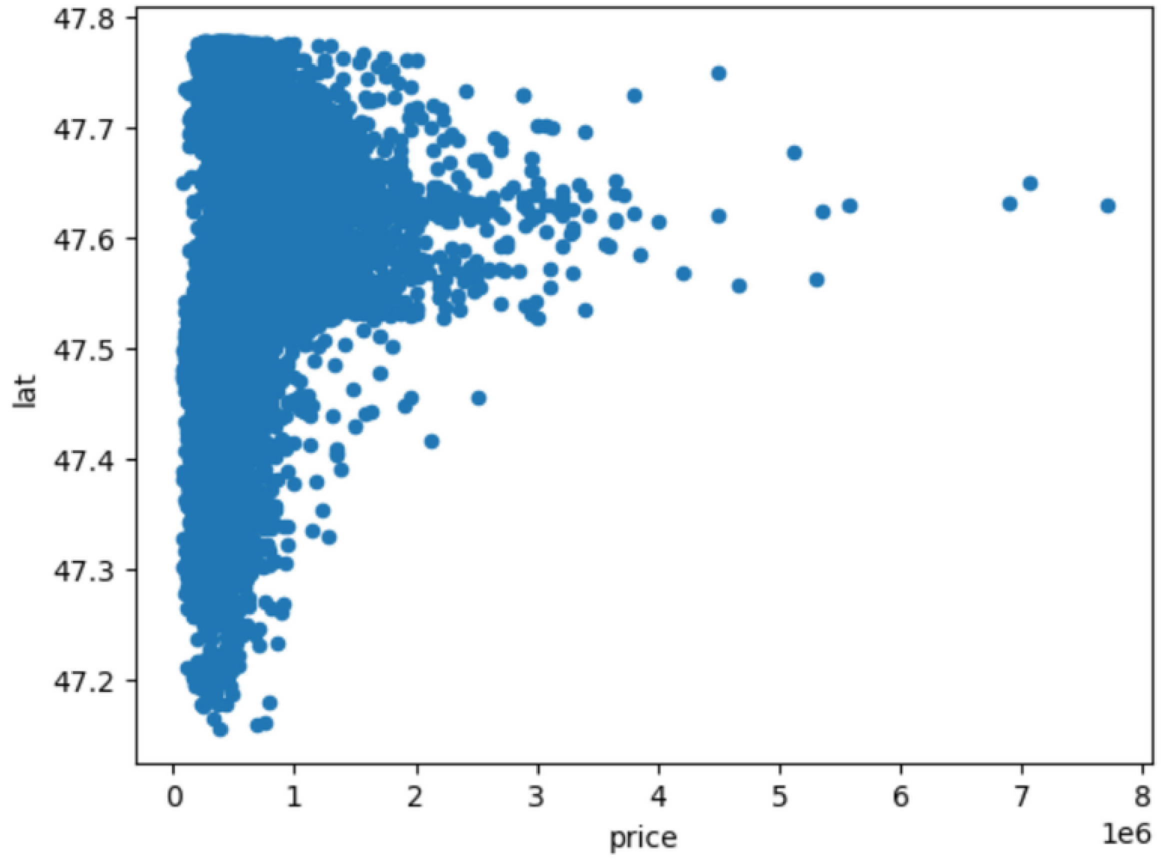


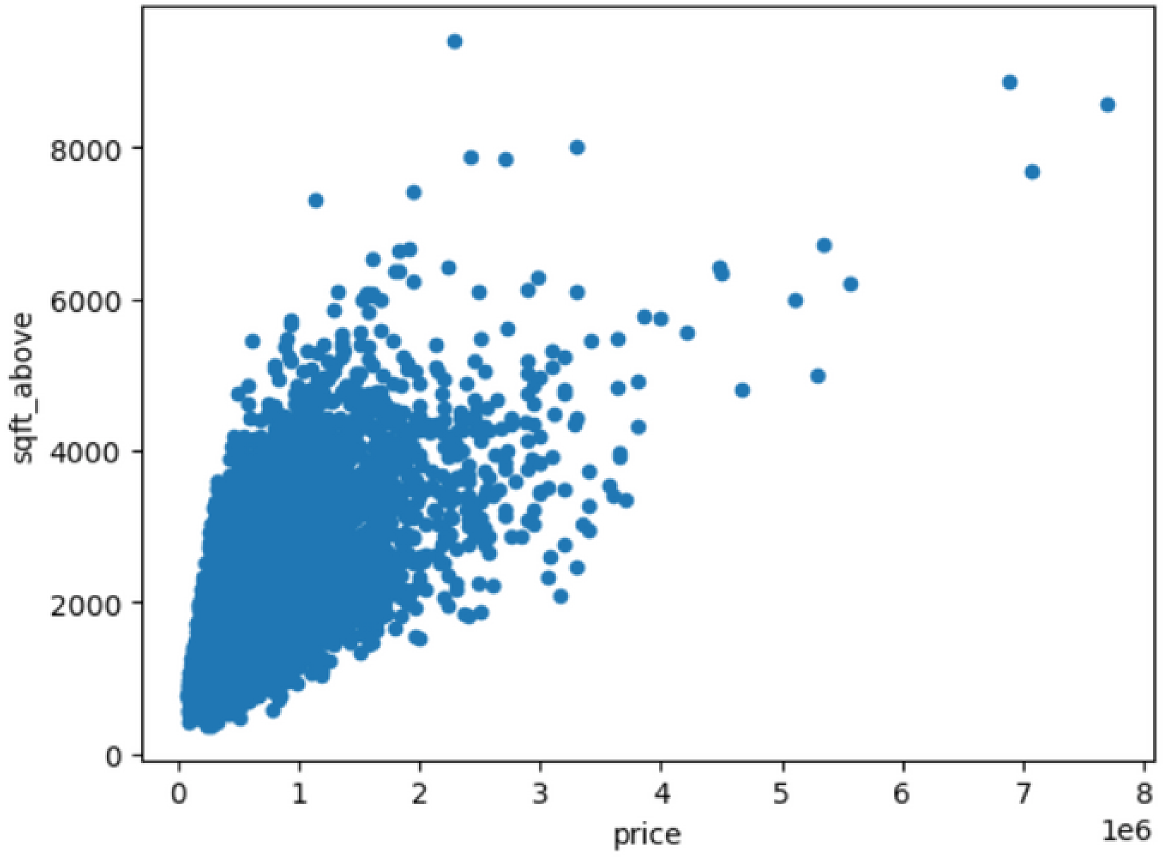
Training the model with ?

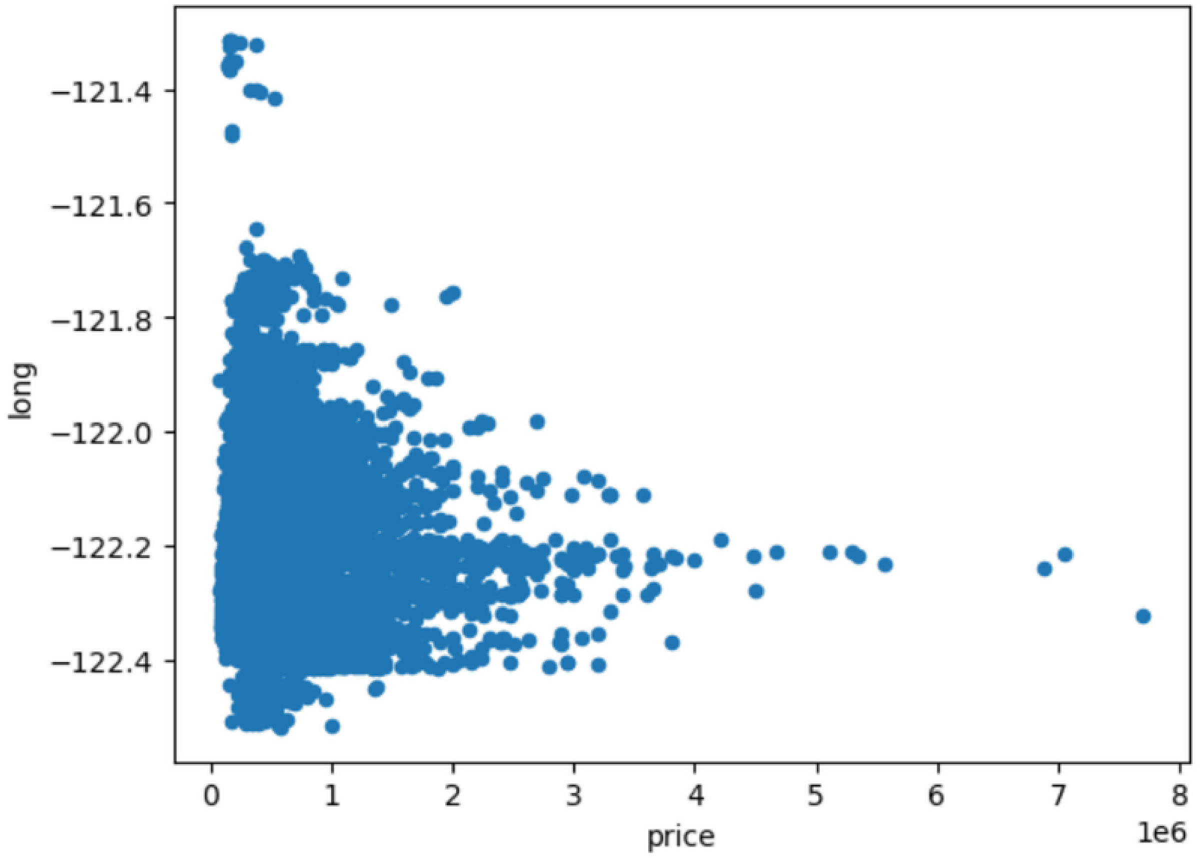
- Exploratory Data Analysis
- Introduce a new variable to the model: average_price_by_city
- Linear Regression, KNN Model, Random Forest Regression

EDA

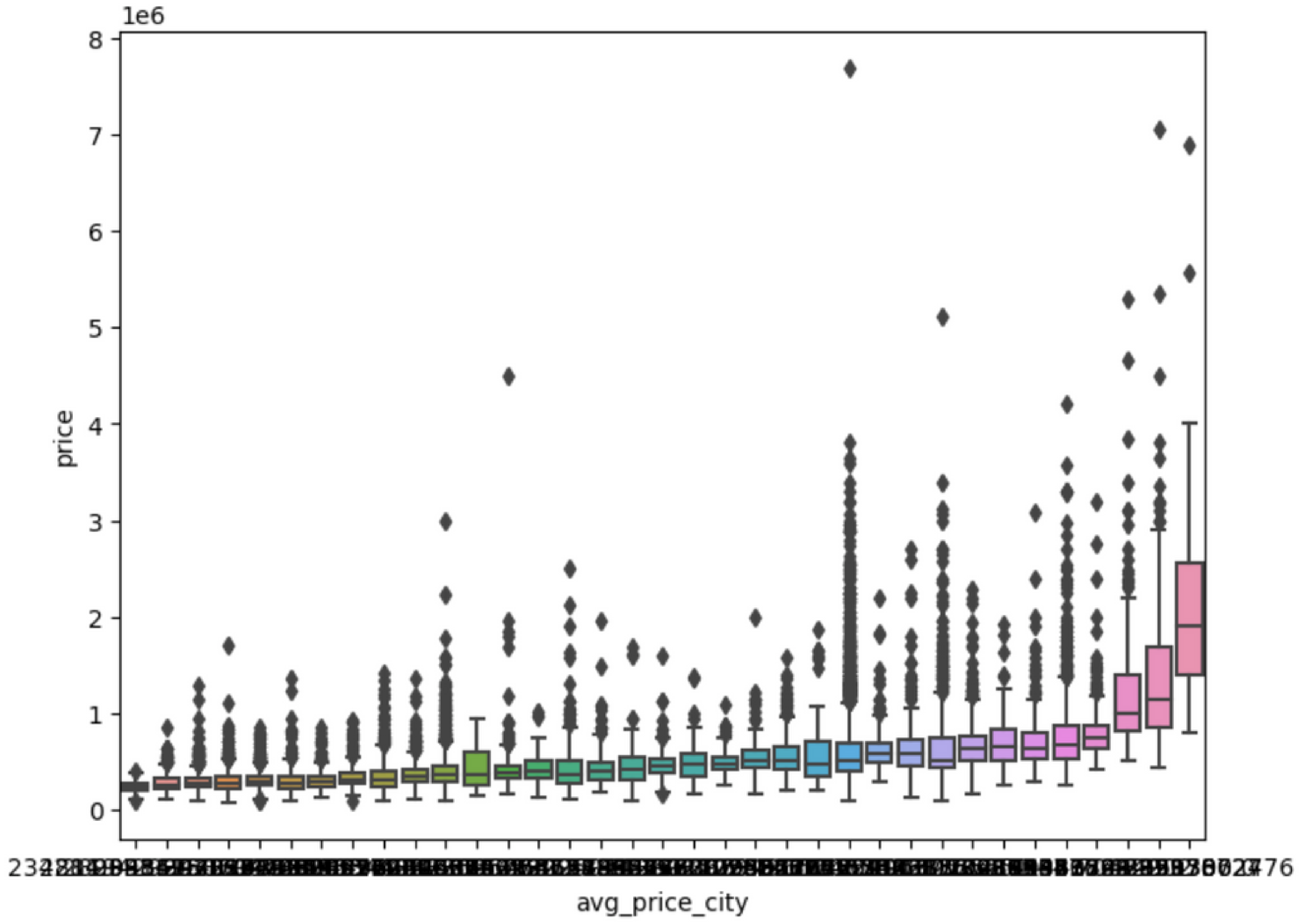

14000
12000
10000
8000
6000
4000
2000
0
sqft_living
0
1
2
3
4
5
6
7
8
price
1e6

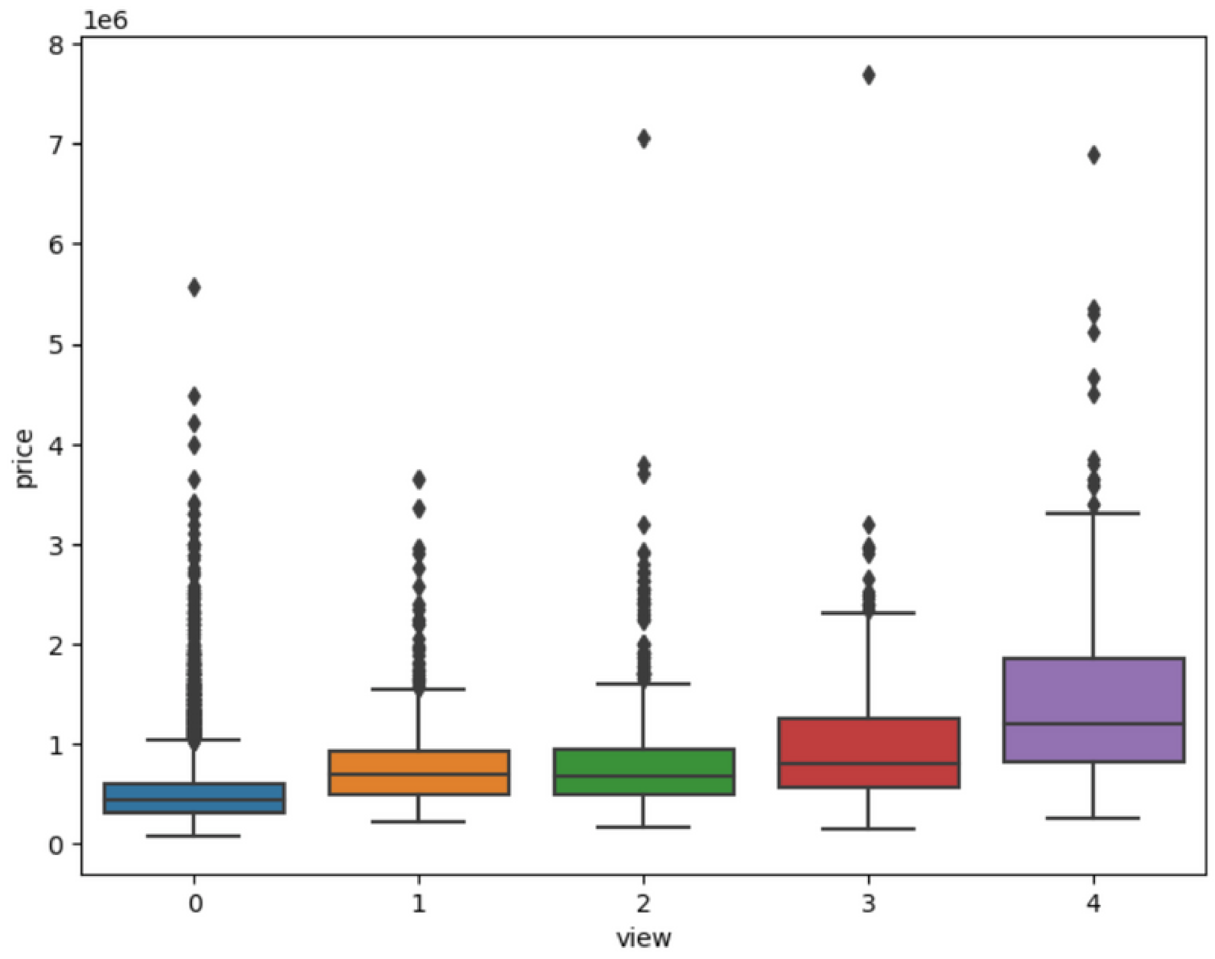

47.8
47.7
47.6
47.5
47.4
47.3
47.2
lat
0
1
2
3
4
5
6
7
8
price
1e6

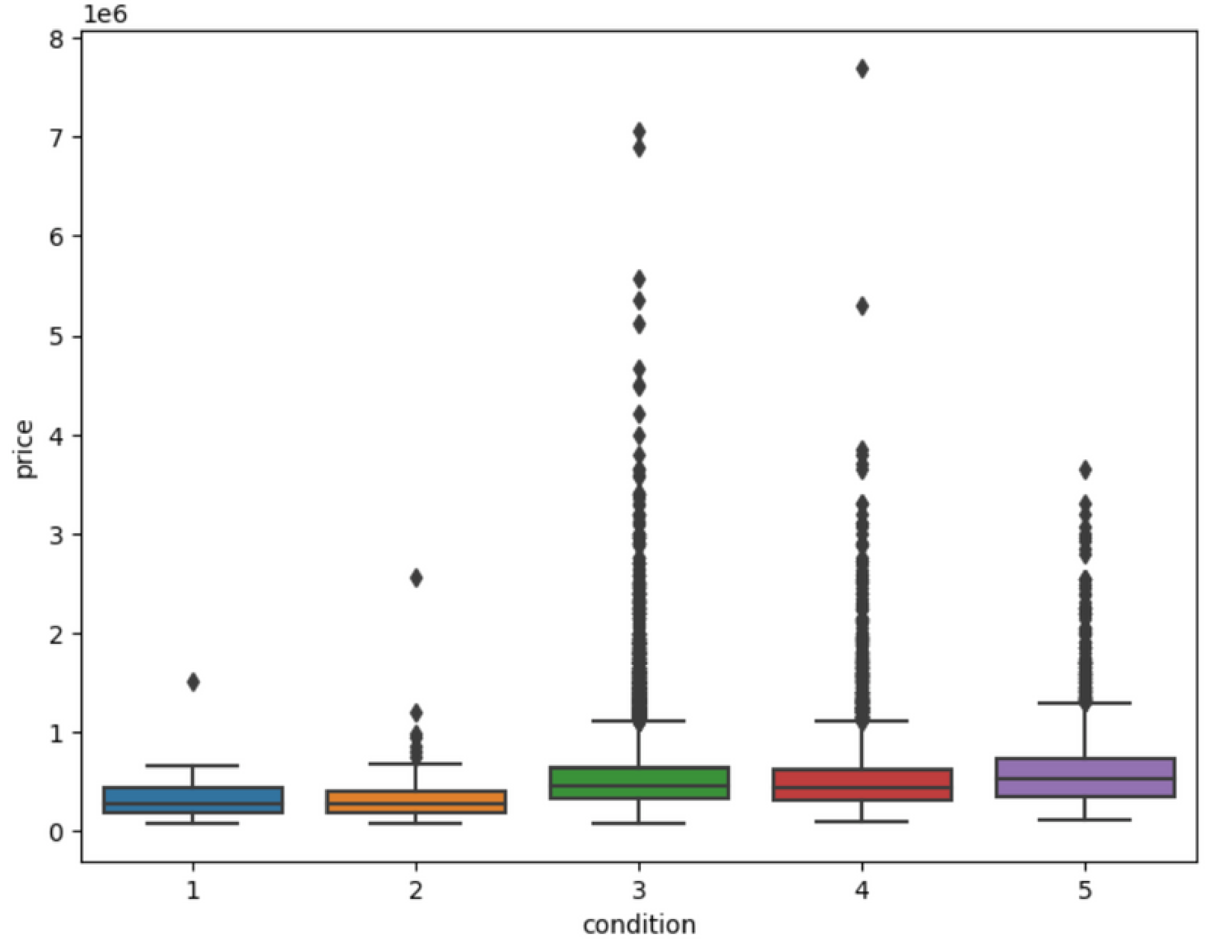

8000
6000
4000
2000
0
sqft_above
0
1
2
3
4
5
6
7
8
price
1e6


−121.4
−121.6
−121.8
−122.0
−122.2
−122.4
long
0
1
2
3
4
5
6
7
8
price
1e6

EDA

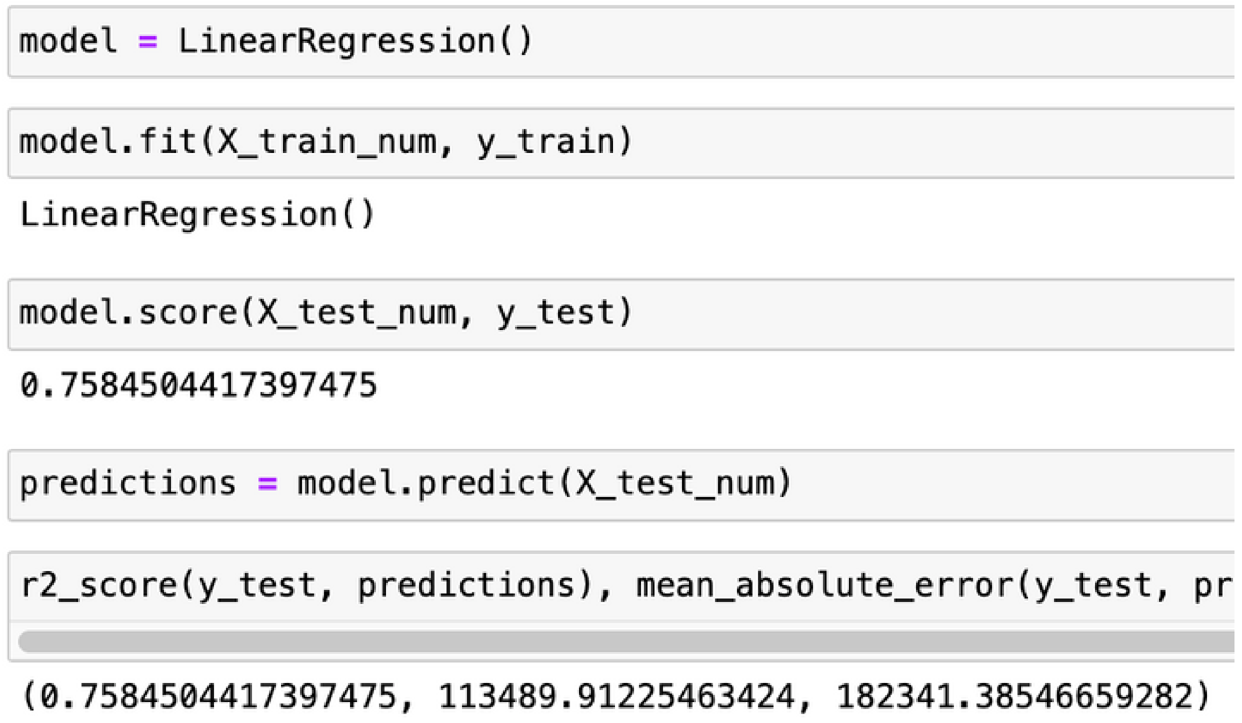

1e6
8
7
6
5
4
3
2
1
0
price
avg_price_city


1e6
8
7
6
5
4
3
2
1
0
price
0
1
2
3
4
view


1e6
8
7
6
5
4
3
2
1
0
price
1
2
3
4
5
condition

Different results

Linear Regression



```
model = LinearRegression()
```

```
model.fit(X_train_num, y_train)
```

```
LinearRegression()
```

```
model.score(X_test_num, y_test)
```

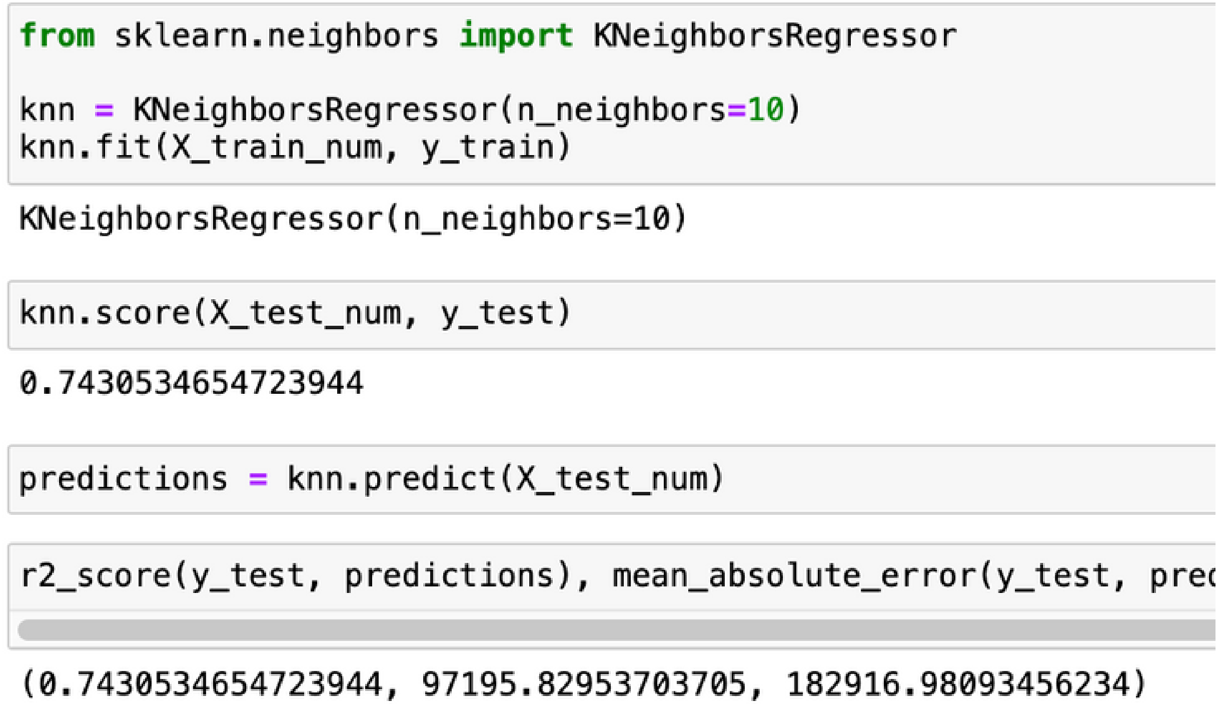
```
0.7584504417397475
```

```
predictions = model.predict(X_test_num)
```

```
r2_score(y_test, predictions), mean_absolute_error(y_test, pr
```

```
(0.7584504417397475, 113489.91225463424, 182341.38546659282)
```

KNN - K Nearest Neighbors



```
from sklearn.neighbors import KNeighborsRegressor

knn = KNeighborsRegressor(n_neighbors=10)
knn.fit(X_train_num, y_train)
```

```
KNeighborsRegressor(n_neighbors=10)
```

```
knn.score(X_test_num, y_test)
```

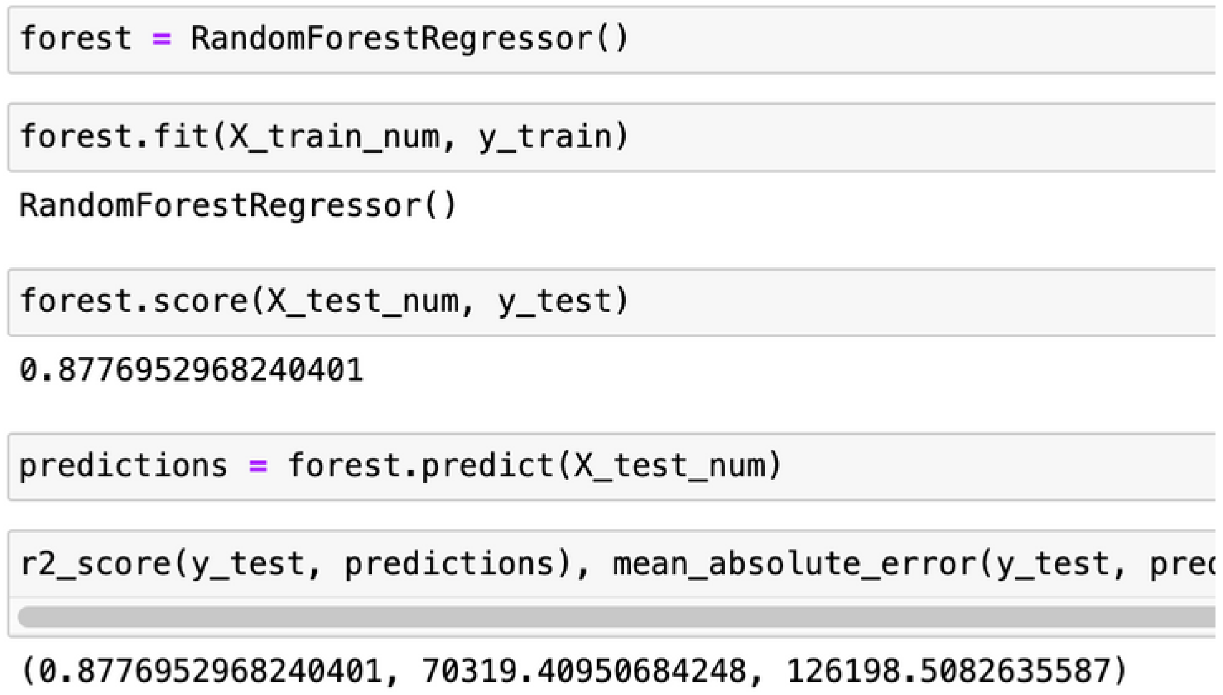
```
0.7430534654723944
```

```
predictions = knn.predict(X_test_num)
```

```
r2_score(y_test, predictions), mean_absolute_error(y_test, pred
```

```
(0.7430534654723944, 97195.82953703705, 182916.98093456234)
```

Random Forest Regression



```
forest = RandomForestRegressor()
```

```
forest.fit(X_train_num, y_train)
```

```
RandomForestRegressor()
```

```
forest.score(X_test_num, y_test)
```

```
0.8776952968240401
```

```
predictions = forest.predict(X_test_num)
```

```
r2_score(y_test, predictions), mean_absolute_error(y_test, pred
```

```
(0.8776952968240401, 70319.40950684248, 126198.5082635587)
```

Analyzing results

In term of pure analytic, some numbers

Our best prediction model result : Random forest regression

In term of business

Due to our good result, 88% of chances that we predict the good price.
We are confident that we can use our model in order to predict the future prices.
Indeed, it is now much more easier to evaluate the selling price of our houses to align with the market and increase our profits now. It will also save us a lot of time so we can focus more on different regions.