BATTLE OF ZURICH

Coursera Capstone by Markus Kessler

The Problem

- The Importance of price-predicting is obiquitous
- Nowing what price will come out based on the location can give extreme insight.
- Allows i.e. a bank or a coorporation to decide if they should invest in a real estate project.
- But what exactly has an impact on the price of a flat?

My Motivation

“Working with all of this loacation based data in the previous lessions, i wonderd if i could implement a linear regression model to predict the pricing of flats around the city of Zurich, Switzerland.”

The gathered Data

Biggest real estate platform of Switzerland: ImmoScout24.ch.

Features:

- Price
- Surface
- Room number
- Title of the Advertisement
- Latitude / Longitude

The generated Data

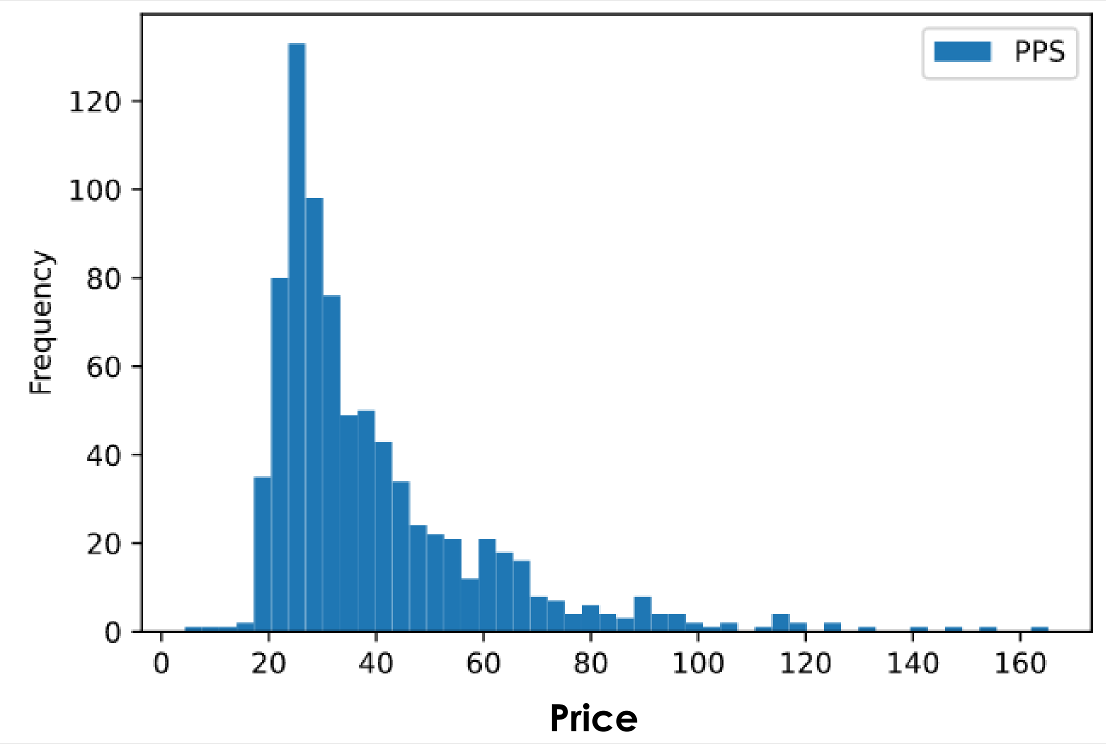
- Number of Foursquare Points-Of-Interest
- Distance to Main Station
- Keyword-checking from Title like “Lakeview” or “new”

The Preprocessing

- Filtering the Title for Keywords to fill in 4 onehot features:
  - Vista, Bright, New, Furnished

- Calculating the distance to the mainstation with latitude / longitude and the help of the Geopy distance function.

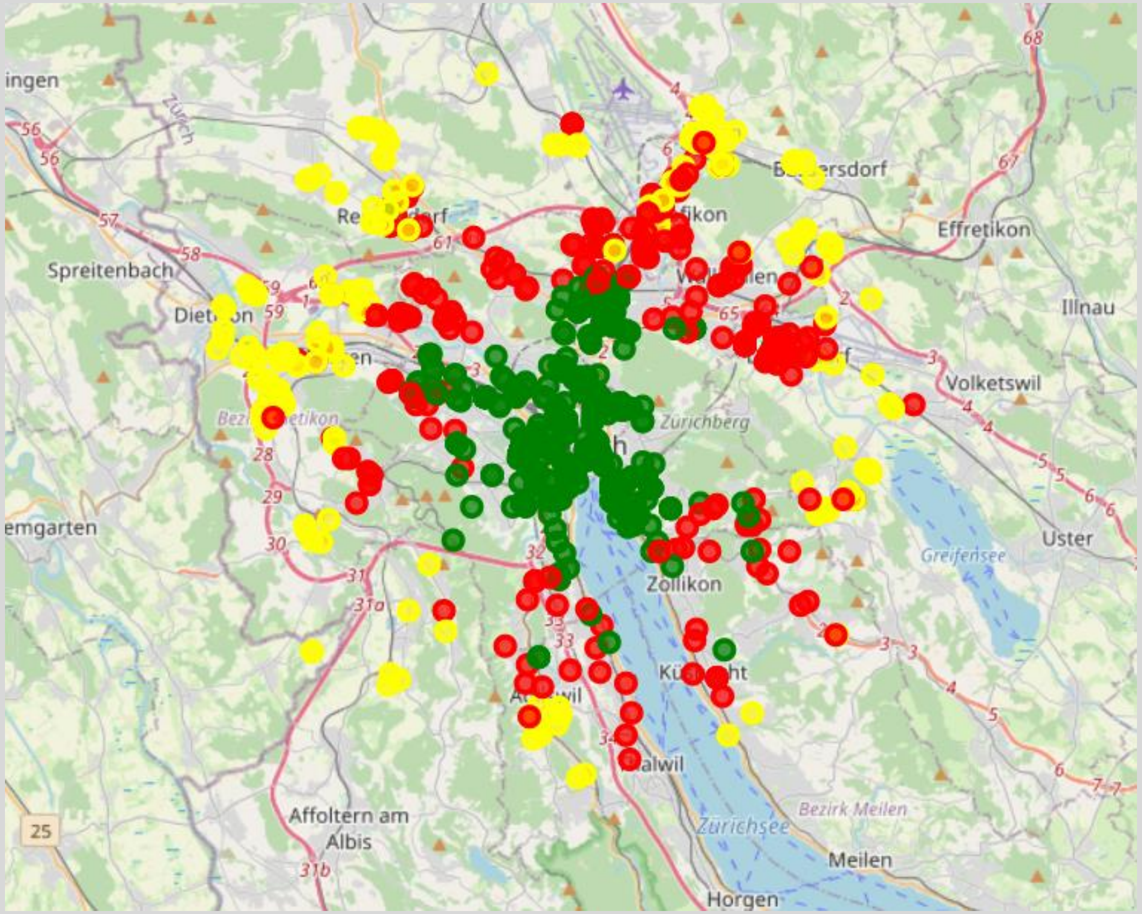
- Calculating Price per squaremeter as this will be the target variable. Distribution shown right.



Digression: Zurich in Clusters

- Visualizing flats as Kmeans clusters
- Features: PPS, Station Distance, Vista

- Result: Lake, river Limmat and region Oerlikon seem to attract more expensive flats (green), not only distance to center.



Syphoning Folium

- Get 50 nearest POIs in a radius ot 500m for every flat
- Store as onehot encoding to allow filtering
- Add number of (important) POIs

ID	Accessories Store	Advertising Agency	Airport	Airport Lounge	Airport Terminal	American Restaurant
2141913	0	0	0	0	0	0
2926781	0	0	0	0	0	0
3093372	0	0	0	0	0	0
3181771	0	0	0	0	0	0
3181854	0	0	0	0	0	0
...	...	...	...	...	...	...
6316267	0	0	0	0	0	0
6316268	0	0	0	0	0	0
6316270	0	0	0	0	0	1

Testing and Results (1/2)

Checking the correlation of all features towards Price per squaremeter:

- Every feature above 0.5 will be used
- Sadly, no keyword-checking correlates strongly enough
- «Furnished» might be worth of some further insight

```
Out[123]: Price          0.156006
          RoomNr        -0.574463
          Surface       -0.529985
          Latitude      -0.212653
          Longitude      0.005043
          Vista         -0.044556
          Bright        -0.107253
          New           -0.142452
          Furnished      0.427571
          StationDist   -0.573536
          PPS            1.000000
          Cat            0.044625
          POIS           0.560316
          Name: PPS, dtype: float64
```

Testing and Results (2/2)

- Data split into train and test set
- Fitted into linear regression model by SciKit Learn
- R-squared returns value of 0.6. Might be a best-case score but shows potential of the model

```
In [120]: Y_hat = linMod.predict(X_test)
          print("Residual sum of squares: %.2f"
                % np.mean((Y_hat - Y_test) ** 2))
          print('Variance score: %.2f' % linMod.score(X_test, Y_test))

          Residual sum of squares: 154.02
          Variance score: 0.60
```

Summary

- Basic prediction of price is possible
- Only features needed where position, room number and surface

“It’s self explaining that this project is way too less detailed to actually produce a linear regression model that would be usable in the real world. But i think i could show some interesting aspect and angles one could use to approach this problem.”