Vacation Home Price Predicting Case

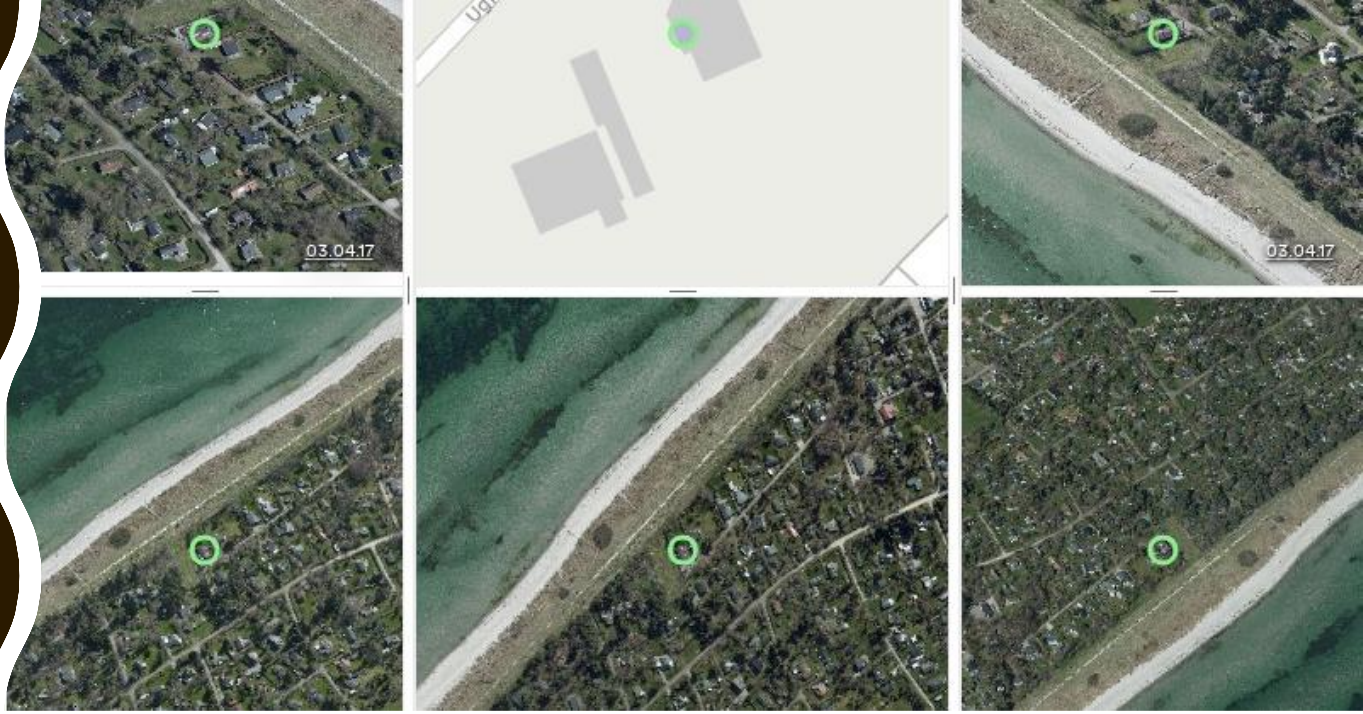
YIHAO SUN

02.January.2020

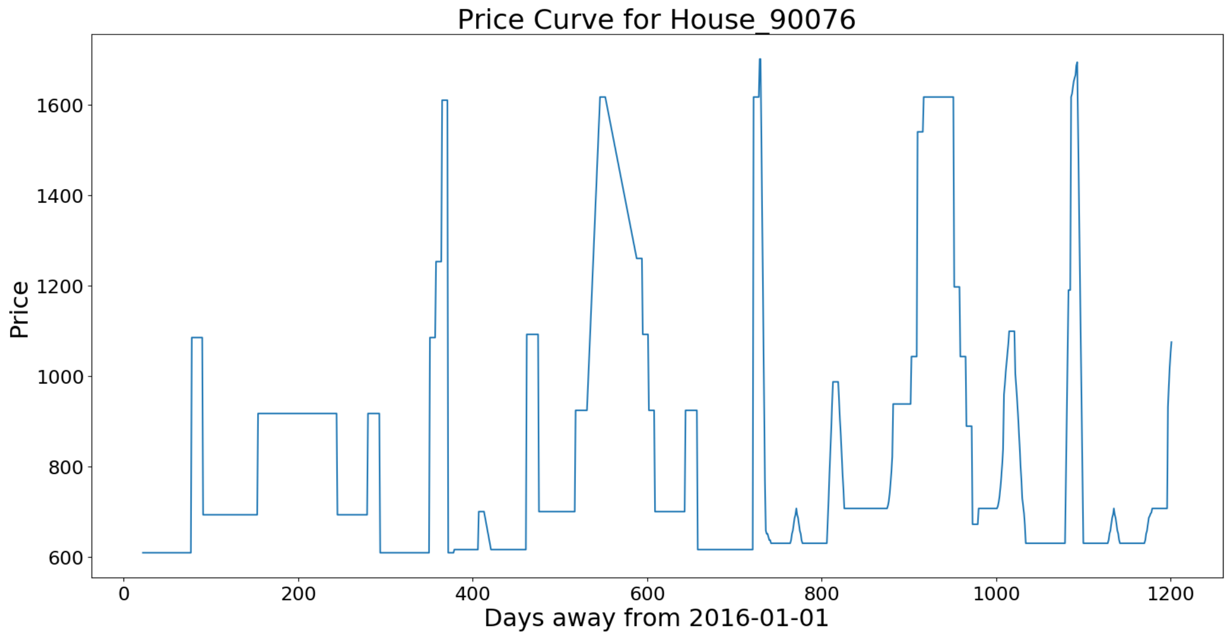
WHAT REALLY DETERMINES THE PRICE OF A VACATION HOME?

FACTOR ONE:

YEAR & CALENDAR



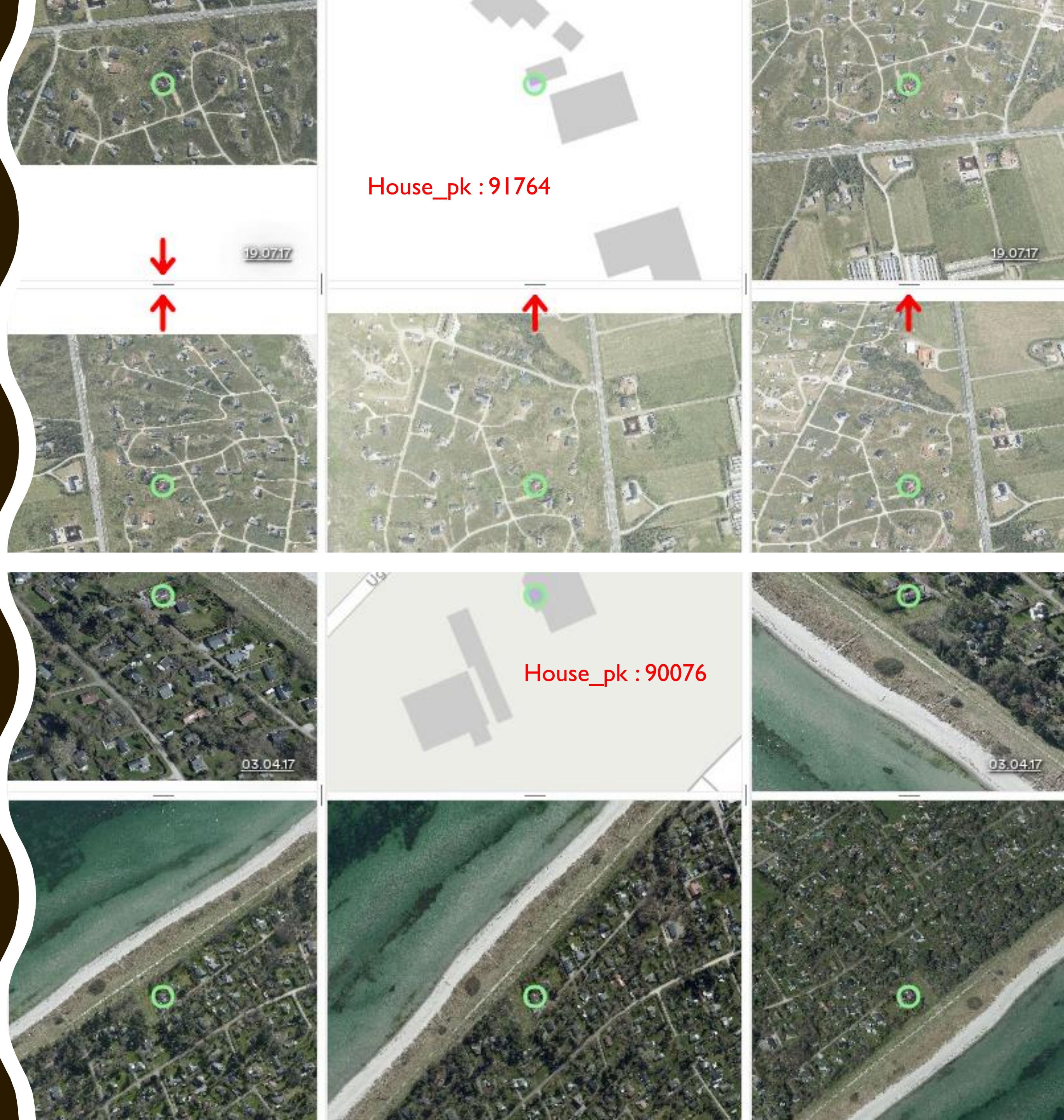
House_pk : 90076



WHAT REALLY DETERMINES THE PRICE OF A VACATION HOME?

FACTOR TWO:

GEOGRAPHICAL ATTRIBUTES



STRATEGY

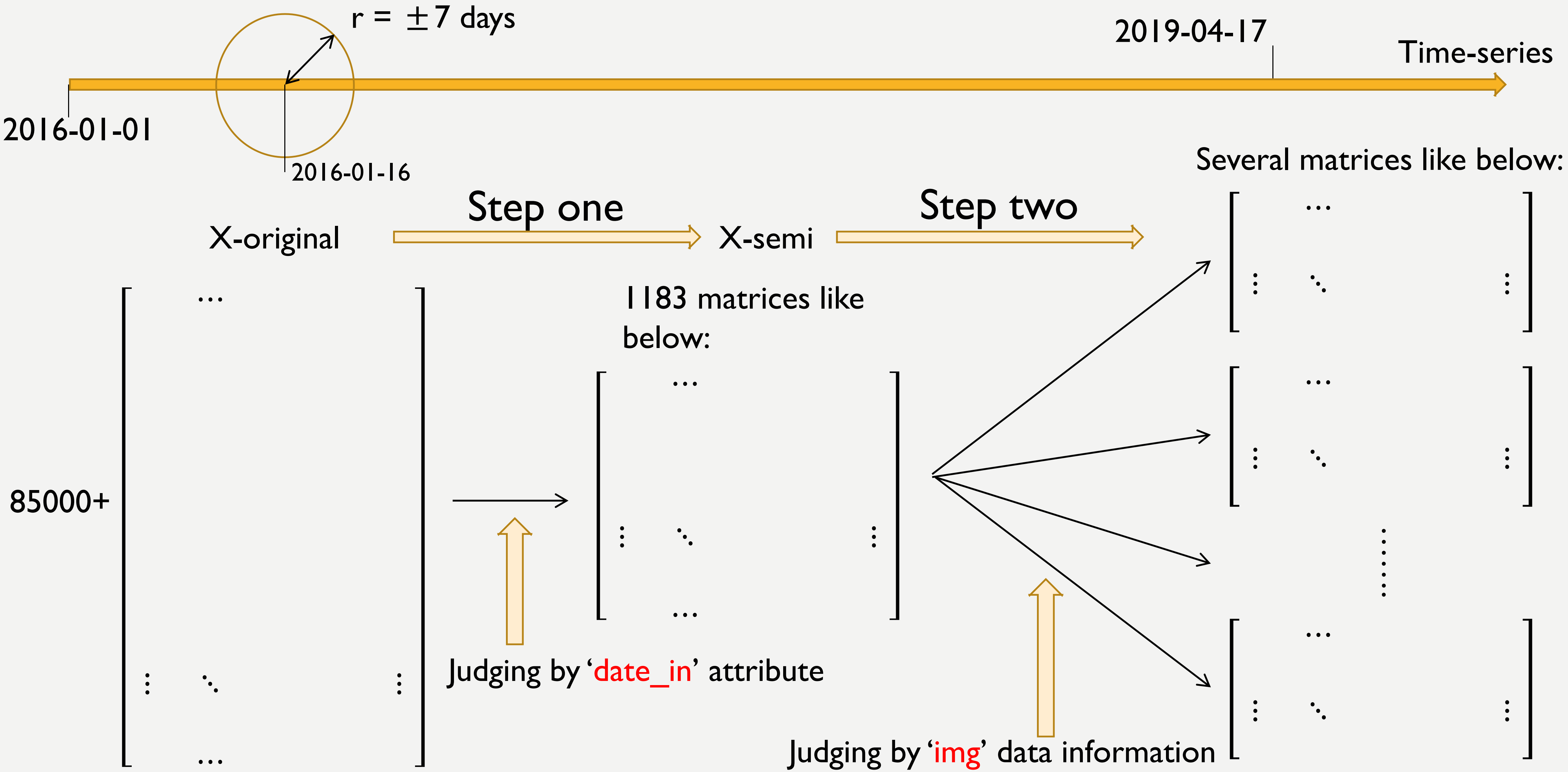
- A time series model

With the consideration of periodic

- Geographical attributes

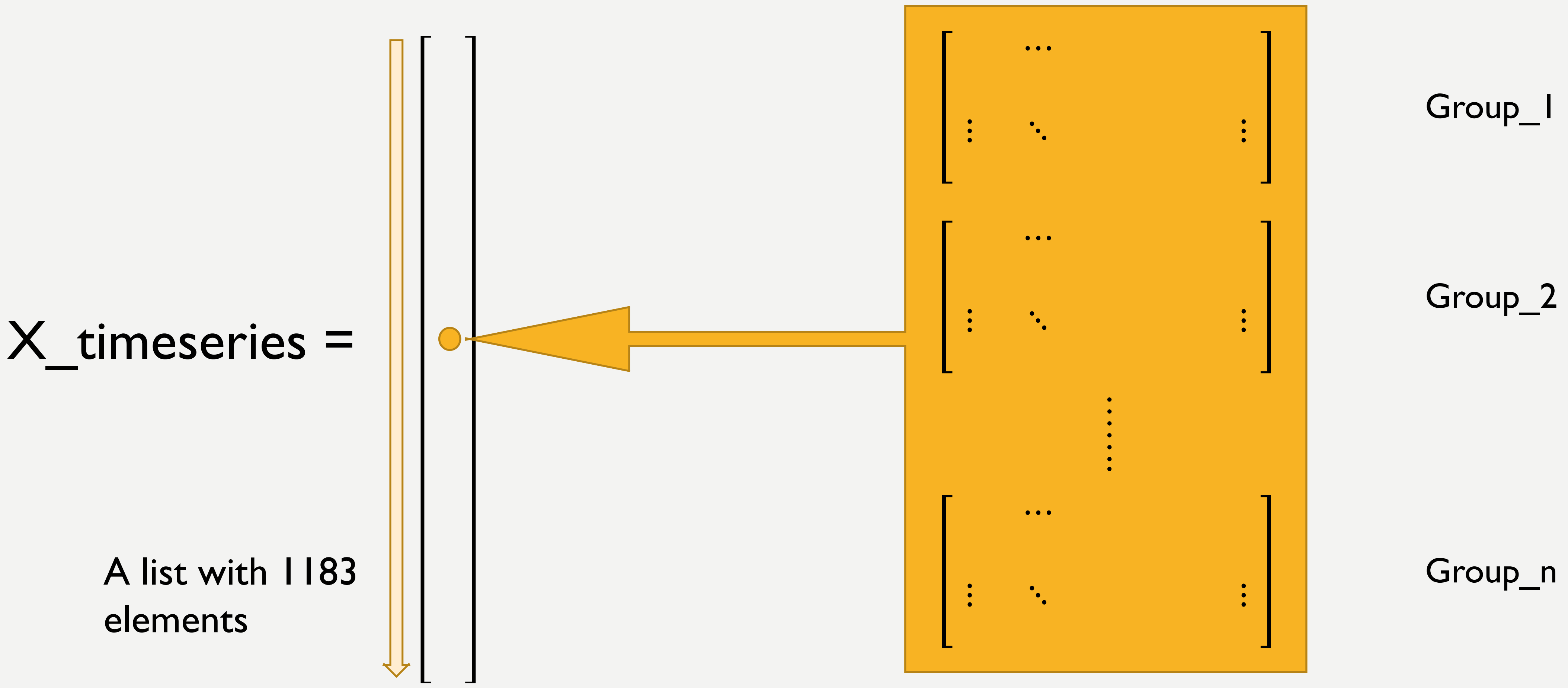
With the consideration of house-clustering

MODEL STRUCTURE



MODEL STRUCTURE

Data for training the model looks like:



DABULAR DATA PRE-PROCESS

Min: 2016-01-16
Max: 2019-04-17
+ Set base as 2016-01-01

Inherent attributes

house-pk	date_in	price	dis_water_real	dis_shopping	....	agency_rating	img_FC2_vector
.							
.							
.							
.							
.							

Transfer to the difference between
this day and 2016-01-01(base)

IMAGE DATA PRE-PROCESS

- Load the image and makes their names into 'dict'
- Transfer learning on a **VGG-16** model, who was pre-tained for image data Cosine distance. Calculate the distance matrix, which can be imported by *distance_matrix[house_pk_dict[90076], house_pk_dict[90002]]*
- By using the **sklearn.cluster.DBSCAN** for the distance matrix, to cluster the house into different groups
- Track back the index of each image in each group

Reference: https://medium.com/@jeff.lee.1990710/image-similarity-using-vgg16-transfer-learning-and-cosine-similarity-98571d8055e3

XGBOOST

After splitting the data by following the strategy mentioned as previous, 90% as training set, and 10% as testing set, the data can be directly poured into the XGBoost.

```
####################################################################
Attributes used for XGBoost prediction, only for importance checking:
Attribute name: date_in ,  column index: 0
Attribute name: dis_water_real ,  column index: 1
Attribute name: dis_shopping ,  column index: 2
Attribute name: no_bedrooms ,  column index: 3
Attribute name: max_persons ,  column index: 4
Attribute name: house_size ,  column index: 5
Attribute name: land_size ,  column index: 6
Attribute name: build_year ,  column index: 7
Attribute name: renovation_year ,  column index: 8
Attribute name: apartment ,  column index: 9
Attribute name: indoor_pool ,  column index: 10
Attribute name: spa ,  column index: 11
Attribute name: internet ,  column index: 12
Attribute name: pets_allowed ,  column index: 13
Attribute name: water_view ,  column index: 14
Attribute name: fire_stove ,  column index: 15
Attribute name: agency_rating ,  column index: 16
```

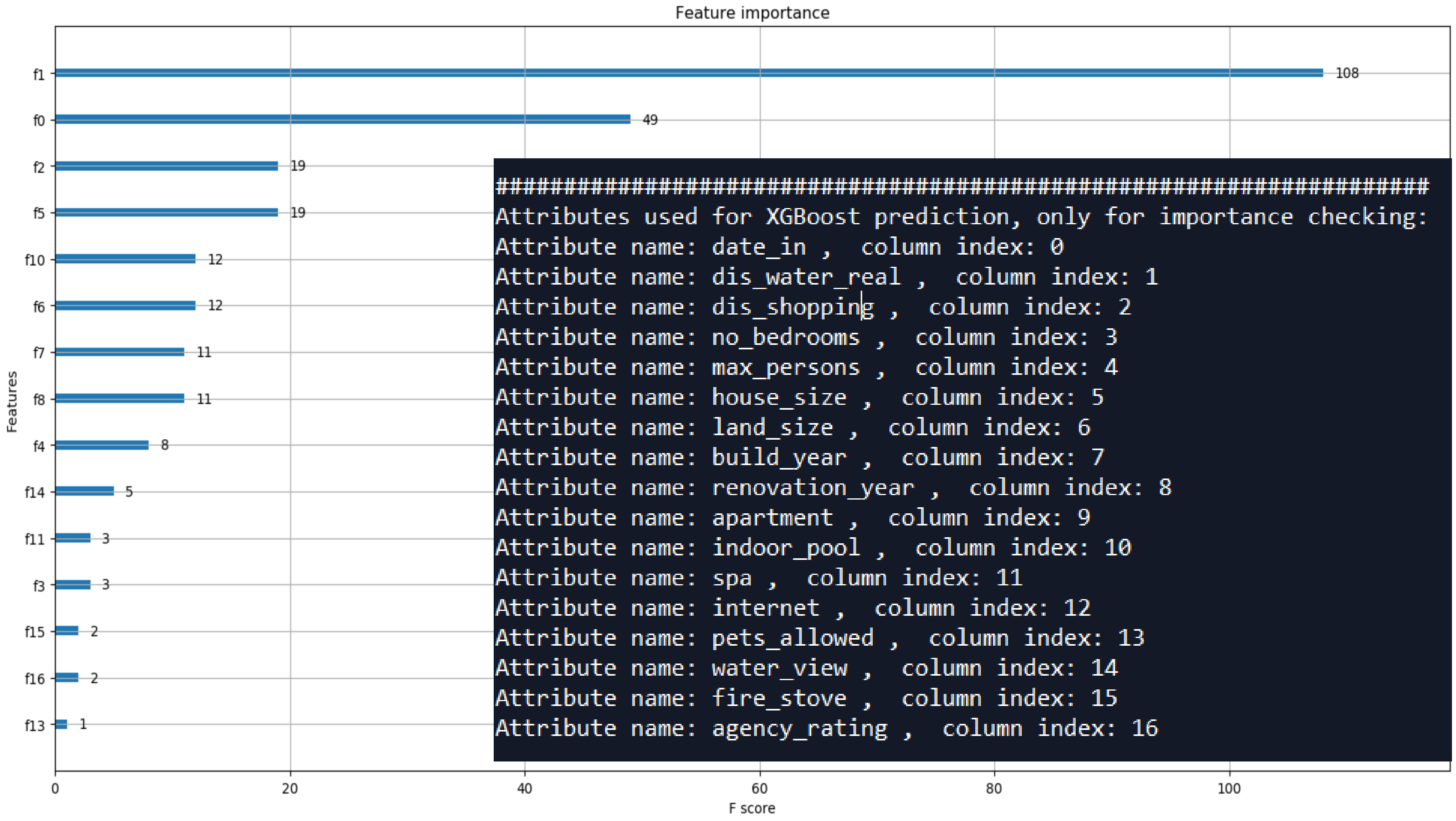
```
##################
Number of clusters:  6
##########
Cluster -1:
[  1108   6611   7602   7603   7604   7605  27735  27743  83780  83838
  84077  84197  84357  84408  84561  85669  85866  86867  88541  88704
  90002  90076  90574  90682  91466  91513  91750  91751  91757  91764
  95470  96240  96875  97500  97657  97729  98140  98416  98913  99569
  99669  99672 100104 103705 105135 105341 107919 108607 108696 108701
 109206 111214 111301 113364 115373]

##########
Cluster 0:
[ 22604  27742  84270  84271  84517  84638  86192  86741  86772  86872
  87030  87123  87425 100357 100914 101007 102480 102686]

##########
Cluster 1:
[ 84280  85021  85248  85799  91760 115610 116663]

##########
Cluster 2:
[ 84336  88586  90137  90879  91250  91969  92178  92188  92534  95746
 110945]

##########
Cluster 3:
[ 91857  92358  95903 103646 114674]

##########
Cluster 4:
[ 87503 103030 108367 110117]
```

Take an example from the results, lets' say for '2019-01-23':

```
####################
Under the  2019-01-23
For Group -1
[0]     train-rmse:34.3145      eval-rmse:43.4845
[1]     train-rmse:5.74174      eval-rmse:4.97465
[2]     train-rmse:0.610215     eval-rmse:0.465314
[3]     train-rmse:0.162665     eval-rmse:0.051119
[4]     train-rmse:0.084393     eval-rmse:0.020979
[5]     train-rmse:0.040898     eval-rmse:0.017177
[6]     train-rmse:0.020136     eval-rmse:0.010914
[7]     train-rmse:0.010633     eval-rmse:0.009816
[8]     train-rmse:0.00541      eval-rmse:0.009637
[9]     train-rmse:0.002708     eval-rmse:0.009302
####################
Under the  2019-01-23
For Group 0
[0]     train-rmse:61.6069      eval-rmse:48.9227
[1]     train-rmse:5.78264      eval-rmse:3.51121
[2]     train-rmse:1.16435      eval-rmse:0.356607
[3]     train-rmse:0.281646     eval-rmse:0.045736
[4]     train-rmse:0.069975     eval-rmse:0.006857
[5]     train-rmse:0.017467     eval-rmse:0.001106
[6]     train-rmse:0.004363     eval-rmse:0.00022
[7]     train-rmse:0.001095     eval-rmse:0.000129
[8]     train-rmse:0.000299     eval-rmse:0.000129
[9]     train-rmse:0.000147     eval-rmse:0.00011
####################
Under the  2019-01-23
For Group 1
[0]     train-rmse:54.7822      eval-rmse:59.0569
[1]     train-rmse:5.93111      eval-rmse:11.5073
[2]     train-rmse:1.09478      eval-rmse:2.30056
[3]     train-rmse:0.228668     eval-rmse:0.460076
[4]     train-rmse:0.049335     eval-rmse:0.092014
[5]     train-rmse:0.01091      eval-rmse:0.018407
[6]     train-rmse:0.002474     eval-rmse:0.003691
[7]     train-rmse:0.00062      eval-rmse:0.000576
[8]     train-rmse:0.000455     eval-rmse:0.000702
[9]     train-rmse:0.000391     eval-rmse:0.001046
```

```
####################
Under the  2019-01-23
For Group 2
[0]     train-rmse:36.7229      eval-rmse:27.8867
[1]     train-rmse:2.85963      eval-rmse:1.88659
[2]     train-rmse:0.208638     eval-rmse:0.111366
[3]     train-rmse:0.014832     eval-rmse:0.008984
[4]     train-rmse:0.000895     eval-rmse:0.000539
[5]     train-rmse:0.000111     eval-rmse:0.000107
[6]     train-rmse:0.000111     eval-rmse:0.000107
[7]     train-rmse:0.000111     eval-rmse:0.000107
[8]     train-rmse:0.000111     eval-rmse:0.000107
[9]     train-rmse:0.000111     eval-rmse:0.000107
####################
Under the  2019-01-23
For Group 3
[0]     train-rmse:63.0069      eval-rmse:59.905
[1]     train-rmse:4.35943      eval-rmse:4.321
[2]     train-rmse:0.304183     eval-rmse:0.314803
[3]     train-rmse:0.021417     eval-rmse:0.023146
[4]     train-rmse:0.001538     eval-rmse:0.001728
[5]     train-rmse:0.0001       eval-rmse:0.00011
[6]     train-rmse:9.8e-05      eval-rmse:0.000109
[7]     train-rmse:9.8e-05      eval-rmse:0.000109
[8]     train-rmse:9.8e-05      eval-rmse:0.000109
[9]     train-rmse:9.8e-05      eval-rmse:0.000109
####################
Under the  2019-01-23
For Group 4
[0]     train-rmse:22.4643      eval-rmse:22.4643
[1]     train-rmse:1.60458      eval-rmse:1.60458
[2]     train-rmse:0.114624     eval-rmse:0.114624
[3]     train-rmse:0.008179     eval-rmse:0.008179
[4]     train-rmse:0.00058      eval-rmse:0.00058
[5]     train-rmse:3.1e-05      eval-rmse:3.1e-05
[6]     train-rmse:0    eval-rmse:0
[7]     train-rmse:0    eval-rmse:0
[8]     train-rmse:0    eval-rmse:0
[9]     train-rmse:0    eval-rmse:0
#####################################
```

Take an example from the results, lets' say for '2019-01-23':




```
####################################################################
Attributes used for XGBoost prediction, only for importance checking:
Attribute name: date_in ,  column index: 0
Attribute name: dis_water_real ,  column index: 1
Attribute name: dis_shopping ,  column index: 2
Attribute name: no_bedrooms ,  column index: 3
Attribute name: max_persons ,  column index: 4
Attribute name: house_size ,  column index: 5
Attribute name: land_size ,  column index: 6
Attribute name: build_year ,  column index: 7
Attribute name: renovation_year ,  column index: 8
Attribute name: apartment ,  column index: 9
Attribute name: indoor_pool ,  column index: 10
Attribute name: spa ,  column index: 11
Attribute name: internet ,  column index: 12
Attribute name: pets_allowed ,  column index: 13
Attribute name: water_view ,  column index: 14
Attribute name: fire_stove ,  column index: 15
Attribute name: agency_rating ,  column index: 16
```

Thanks for watching.
Lets' go for the scripts !