Statistics and distance based features

Some data for CTR task

	User_id	Page_id	Ad_price	Ad_position
0	4	6	165.977125	Bottom_right
1	4	6	34.5395640	Bottom_right
2	4	6	29.1963786	Bottom_left
3	4	6	79.4373729	Bottom_left
4	4	6	290.534595	Bottom_right
5	4	6	314.412660	Bottom_right
6	4	6	138.9007639	Bottom_right
7	4	6	107.4711914	Bottom_right
8	4	6	242.1089786	Bottom_left
9	4	7	27.16719836	Bottom_left
10	4	7	413.5421978	Bottom_right

Useful features

	User_id	Page_id	Ad_price	Ad_position	Max_price	min_price	Min_price _position
0	4	6	95.874252	Bottom_right	474.63772	73.711548	Bottom_left
1	4	6	215.751007	Bottom_right	474.63772	73.711548	Bottom_left
2	4	6	474.637726	Bottom_left	474.63772	73.711548	Bottom_left
3	4	6	73.711548	Bottom_left	474.63772	73.711548	Bottom_left
4	4	6	79.288841	Bottom_right	474.63772	73.711548	Bottom_left
5	4	6	271.391785	Bottom_right	474.63772	73.711548	Bottom_left
6	4	6	296.529053	Bottom_right	474.63772	73.711548	Bottom_left
7	4	6	96.030029	Bottom_right	474.63772	73.711548	Bottom_left
8	4	6	130.175064	Bottom_left	474.63772	73.711548	Bottom_left
9	4	7	35.465202	Bottom_left	121.54219	35.465202	Bottom_left
10	4	7	121.542191	Bottom_right	121.54219	35.465202	Bottom_left

Useful features: implementation

In [22]:
```
gb = df.groupby(['user_id','page_id'],as_index=False).agg(
    {'ad_price':{'max_price':np.max, 'min_price':np.min}})
gb.columns = ['user_id','page_id','min_price','max_price']
gb
```

Out[22]:

	user_id	page_id	min_price	max_price
0	4	6	73.711548	474.637726
1	4	7	35.465202	121.542191

In [23]:
```
df = pd.merge(df,gb,how='left',on=['user_id','page_id'])
```

More features

- How many pages user visited
- Standard deviation of prices
- Most visited page
- Many, many more

Neighbors

- Explicit group is not needed
- More flexible
- Much harder to implement

Neighbors

- Number of houses in 500m, 1000m,..
- Average price per square meter in 500m, 1000m,..
- Number of schools/supermarkets/parking lots in 500m, 1000m,..
- Distance to closest subway station

Neighbors

- Explicit group is not needed
- More flexible
- Much harder to implement

KNN features in Springleaf

- Mean encode all the variables
- For every point, find 2000 nearest neighbors using

Bray-Curtis metric

$$\sum |u_i - v_i| / \sum |u_i + v_i|$$

- Calculate various features from those 2000 neighbors

KNN features in Springleaf

- Mean target of nearest 5,10,15,500, 2000 neighbors
- Mean distance to 10 closest neighbors
- Mean distance to 10 closest neighbors with target 1
- Mean distance to 10 closest neighbors with target 0

Thank you