Get To Know Your Right Customers

By G7 Consultancy

Liu Yue & Zhu Yilin

Motivation of G7's CAPS Customer Acquisition Prioritization System

Customer Acquisition Prioritization

Marketers ask...

Who are my potential new customers?

How to get access to them?

External data sources!

Customer Acquisition Prioritization

Marketers ask the following questions

Which segment of potential customer is more profitable?

Their demographics?

Their preferences?

Likelihood to become one?

Professional Data Analytics!

Our Missions

To get you a larger ***customer base***,

To ***save*** your marketing budget,

To help you attain greater ***profits***,

With our ***analytics*** service.

Our Beliefs

Customer purchase behaviours extend from ***past*** to the ***future***.

Your ***competitors***' customers may become yours.

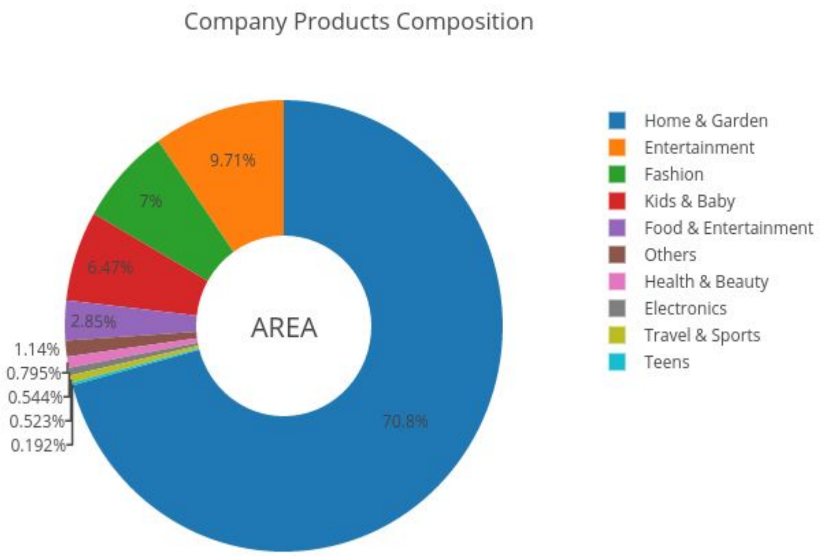
You can always ***strengthen*** on your specializations.

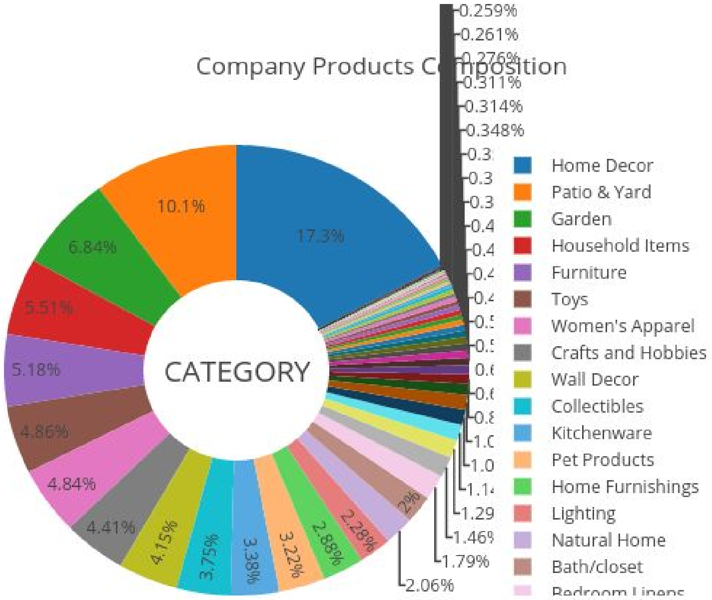
Demo

Our Data Science

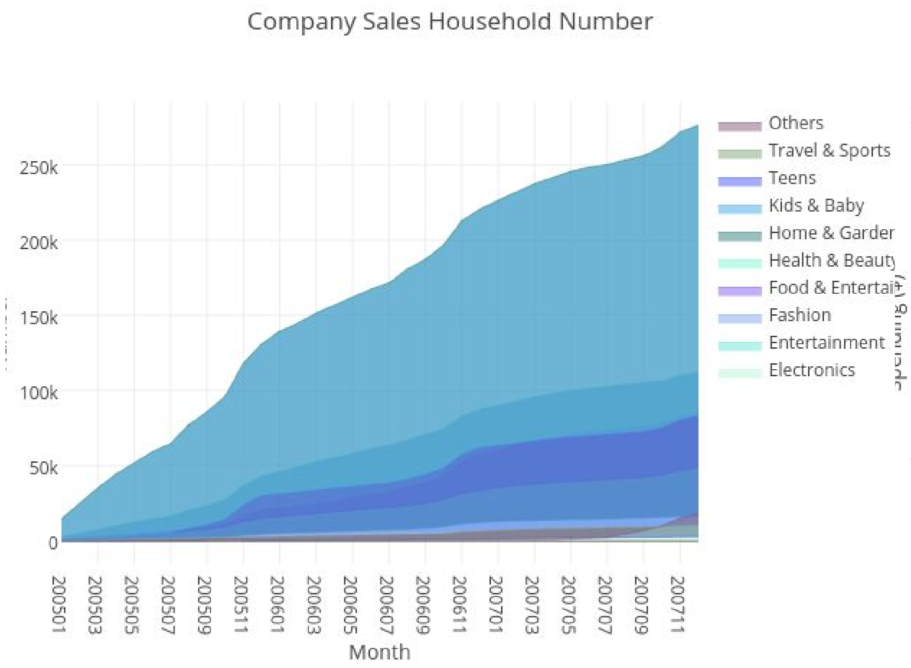
- *Descriptive Analytics*
- *Recommender Algorithms*
- *Dataset*
- *Tools*
- *Models*

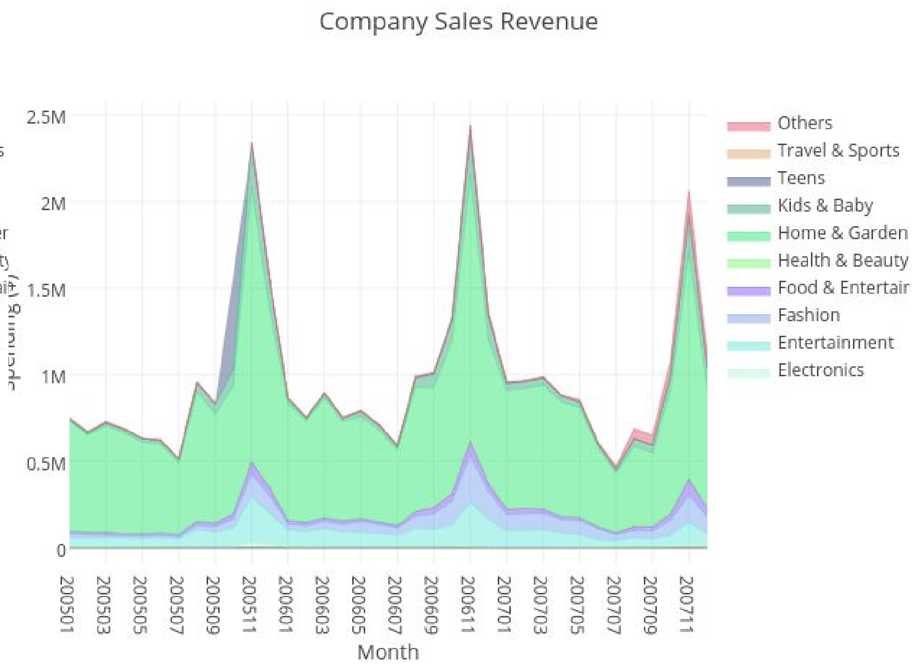
Descriptive Analytics





Descriptive Analysis - Company





Recommender Algorithms

Non-personalized

- Popularity-Based Recommendations
  - Customers who spend more in the past, are likely to spend more in the future
  - “人傻，钱多，快来！“ (marketers pursue for customers with high value)

Personalized for a company

- Popularity-Based Recommendations (proven accuracy)
- User-kNN (traditional)
- SVD popular (popular)
- Co-Clusters (dynamic, near real-time)

DATASET

Household purchasing data, January, 2005 - December, 2007

2.5 million households, 36+million-record database for 207 companies

3 main data files

DMEFLines3Dataset2.csv	-- Line orders
DMEF3YrBase.csv	-- ZIP Code for all households
MajorCatReference.csv	-- More detailed classification of the product purchased

Datasource Credit: Marketing_EDGE Data, Data Set 11

TOOLS

MongoDB v3.4.2

- Original csv files are large with useless columns
- Easy to dump and access data
- Relatively faster to load data than MySQL + XAMPP

Python v3.5.2

Scikit-Surprise v1.0.2

Ploty v2.0.7

MongoDB Screenshot

In [2]:
```
from pymongo import MongoClient
client = MongoClient('mongodb://localhost:27017/')
db = client.BT4221_DB
orders = db.Orders
```

In [3]:
```
import pprint
pprint.pprint(orders.find_one())
```

```
{'Channel': 'C',
 'CompanyID': 837,
 'Dollars': 116,
 'HH_ID': 1,
 'OrderDate': 20051212,
 'OrderNum': 2604929,
 'PaymentType': '',
 '_id': ObjectId('58bc3802bbc6d9a22238d61e')}
```

Models

Standard Collaborative Filtering

- Score: $, log($), scale($), scale_log(#)
    - User-kNN
    - Co-Clustering
    - SVD [with #, log(#)]

Popularity-Based Recommendations

- Not Weighted, or Weighted by by the Company's Product Type

User-kNN

Aim:

- Predict a **worthiness score** for each potential customer

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in N_i^k(u)} \text{sim}(u,v) \cdot r_{vi}}{\sum\limits_{v \in N_i^k(u)} \text{sim}(u,v)}$$

Training data:

- Total $ spent / Total # of purchases
- Actual number / Log scale

User-kNN

Model Setup:

- k=40: The max number of neighbors to take into account for aggregation.

- Similarity measures:

cosine	Compute the cosine similarity between all pairs of users
msd	Compute the Mean Squared Difference similarity between all pairs of users
pearson	Compute the Pearson correlation coefficient between all pairs of users
pearson_baseline	Compute the (shrunk) Pearson correlation coefficient between all pairs of users using baselines for centering instead of means

SVD

Aim:

- Predict a **worthiness score** for each potential customer

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

- By minimizing regularised error via SGD

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2 \right)$$

CF based on Co-Clustering

Aim:

- Predict a **worthiness score** for each potential customer

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i}),$$

- $\overline{C_{ui}}$ average rating of co-cluster
  $\overline{C_u}$ average rating of uu's cluster
  $\overline{C_i}$ average rating of ii's cluster

Parameters setting

- Users and items are assigned 3 clusters and co-clusters.

DATA PROCESSING - CF

Collaborative Filtering --Training Data

Line orders of training period (20050101 - 20070630)

Aggregate Dollars and OrderNum by HH_ID over the period, and transform to log scale

	HH_ID	CompanyID	OrderNum	Dollars	logOrder	logDollar
0	1	837	3	268	1.098612	5.590987
1	3	661	1	24	0.000000	3.178054
2	3	837	3	173	1.098612	5.153292

DATA PROCESSING

Collaborative Filtering --Testing Data

Line orders of from 20070701 to 20071231 for CompanyID = 36

For HH_ID being a customer for companies other than CompanyID = 36 in training period,

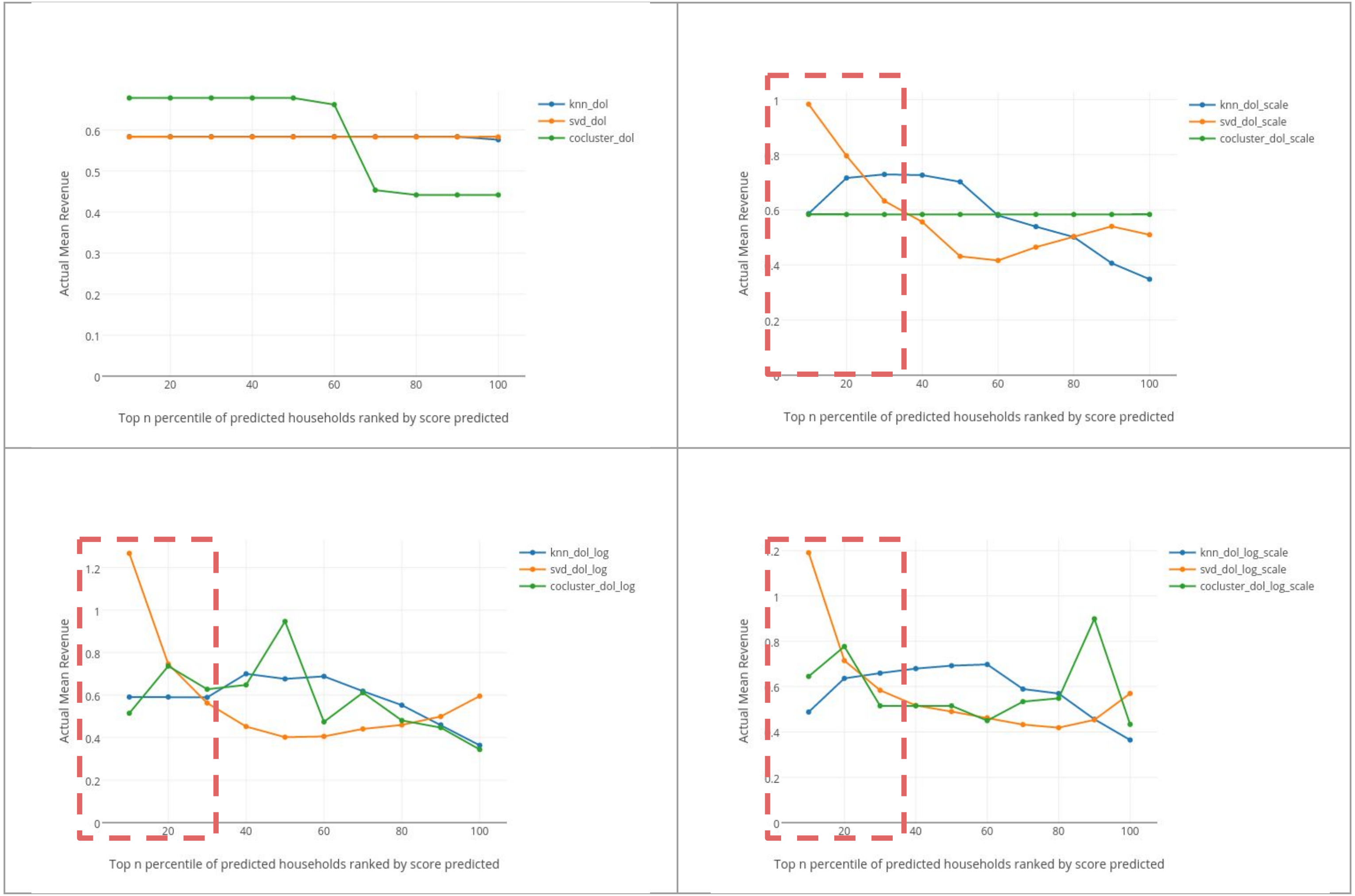
aggregate Dollars and OrderNum by HH_ID over the period, and transform to log scale

	HH_ID	CompanyID	OrderNum	Dollars	logOrder	logDollar
3	343	36	1.0	28.0	0.000000	3.332205
0	338	36	0.0	0.0	-inf	-inf
4	346	36	0.0	0.0	-inf	-inf

PERFORMANCE - CF

	dol		dol_scale		dol_log		dol_log_scale	
	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time
kNN	9.9981	31:58	0.0166	06:49	4.3425	07:19	1.4845	04:33
SVD	9.9993	27:26	0.0155	39:31	3.7073	43:19	1.2045	30:50
Co-clustering	9.5782	21:18	0.0156	33:16	3.5792	34:53	0.9245	21:57

Table 1. Performance for algorithms under each expenditure/revenue-related rating design



knn_dol
svd_dol
cocluster_dol
Actual Mean Revenue
Top n percentile of predicted households ranked by score predicted
knn_dol_scale
svd_dol_scale
cocluster_dol_scale
Actual Mean Revenue
Top n percentile of predicted households ranked by score predicted
knn_dol_log
svd_dol_log
cocluster_dol_log
Actual Mean Revenue
Top n percentile of predicted households ranked by score predicted
knn_dol_log_scale
svd_dol_log_scale
cocluster_dol_log_scale
Actual Mean Revenue
Top n percentile of predicted households ranked by score predicted

CF - IMPROVEMENT

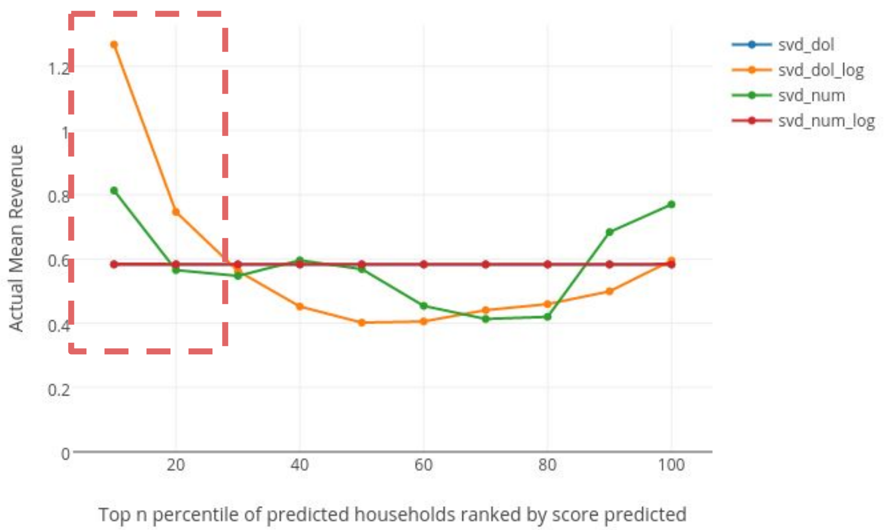
SVD

- Unscaled ratings - #, log(#)

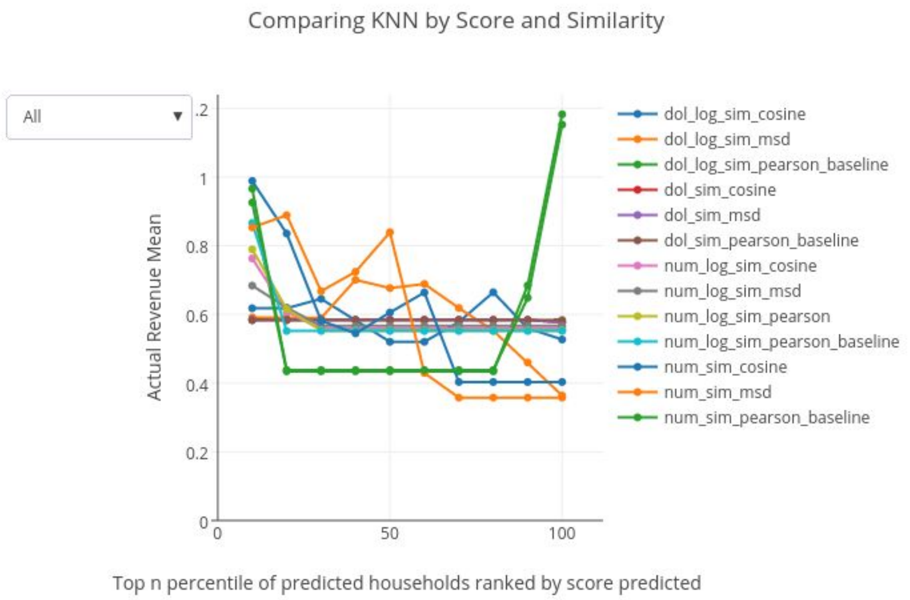
KNN

- Unscaled ratings - #, log(#)
- Different similarity measures
    - cosine, msd, pearson, pearson_baseline

SVD



KNN



To see more in interactive Demo

Popularity

Aim:

- Predict a **worthiness score** for each potential customer

Description:

- Non-personalized baseline
- **General** Popularity Score =

    sum ($ spent in the past 30 months for a customer)

Customised Popularity_By Area

Aim:

- Predict a **worthiness score** for each potential customer

Description:

- Select top **3 of 10** areas by company revenue
- **TopThreeArea** Popularity Score =

  sum ($ spent in the past 30 months for a customer in a area)

  * Percentage revenue of the Company in a area

Customised Popularity_By Category

Aim:

- Predict a **worthiness score** for each potential customer

Description:

- Select top **10 of 69** categories by company revenue
- **TopTenCategory** Popularity Score =

  sum ($ spent in the past 30 months for a customer in a category)

  * Percentage revenue of the Company in a category

DATA PROCESSING - Popularity

Company Revenue Sources

Aggregate Dollars by CompanyID, Area/Category, compute revenue% in each Area/Category.

	Others	Gift Wrapping	Audio/music	Baby Furniture	Baby Gear	Bar & Cigar	Bath/closet	Bathroom Linens	Beauty	Bedroom Decor	...
CompanyID											
27	0.001347	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
32	0.005267	0.000089	0.001875	0.0	0.004999	0.015582	0.007081	0.011475	0.002188	0.000168	...
36	0.011430	0.000000	0.000983	0.0	0.003481	0.001181	0.020015	0.010541	0.001687	0.000463	...

DATA PROCESSING - Popularity

Household Spendings

Aggregate Dollars by HH_ID [, Area, Category] in training period (20050101 to 20070630) for the selected top 3 Area/ 10 Category in the Company Description

	CraftsandHobbies	Furniture	Garden	HomeDecor	HouseholdItems	PatioYard	Toys	WallDecor	WomensApparel
HH_ID									
411835	0.0	30.0	0.0	105.0	33.0	10.0	0.0	0.0	345.0
1285805	15.0	0.0	30.0	0.0	50.0	0.0	0.0	0.0	0.0
1026883	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1586.0

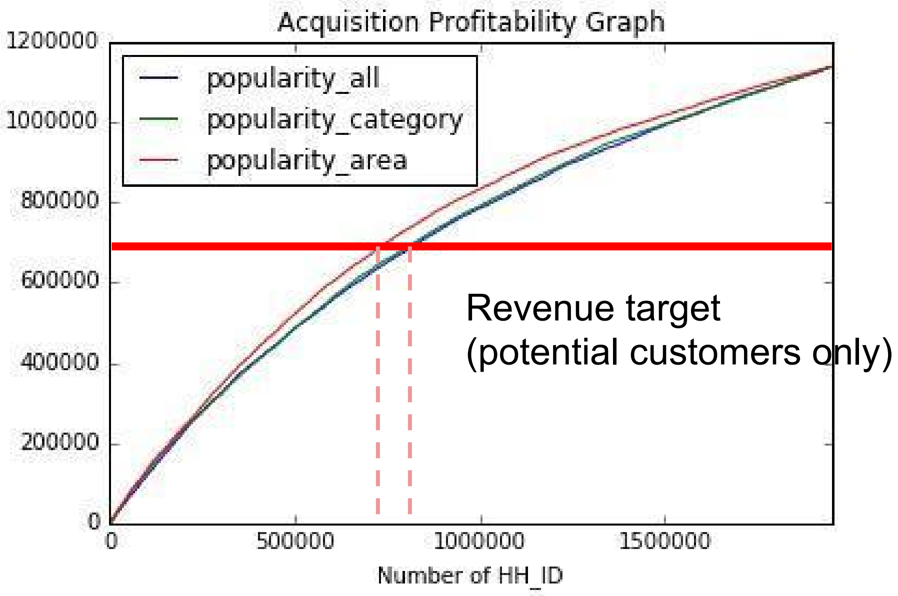
DATA PROCESSING - Popularity

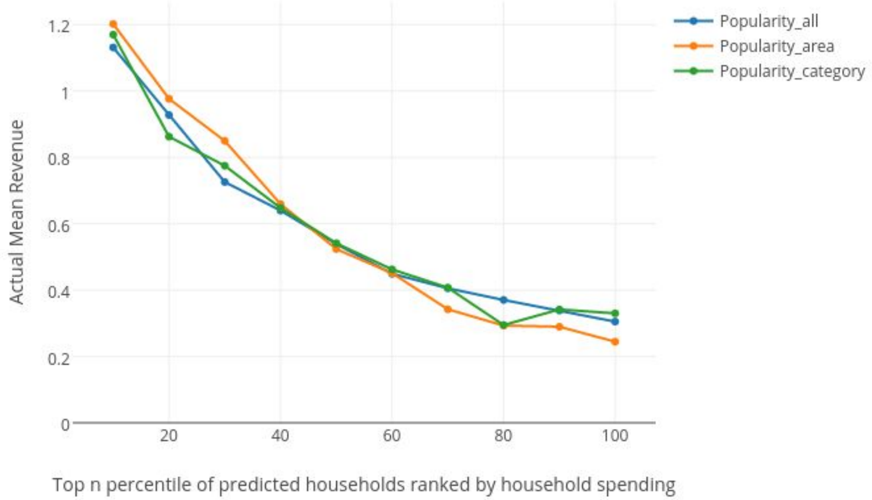
Compare computed score with actual CompanyRevenue in the testing period:

TestingRevenue =

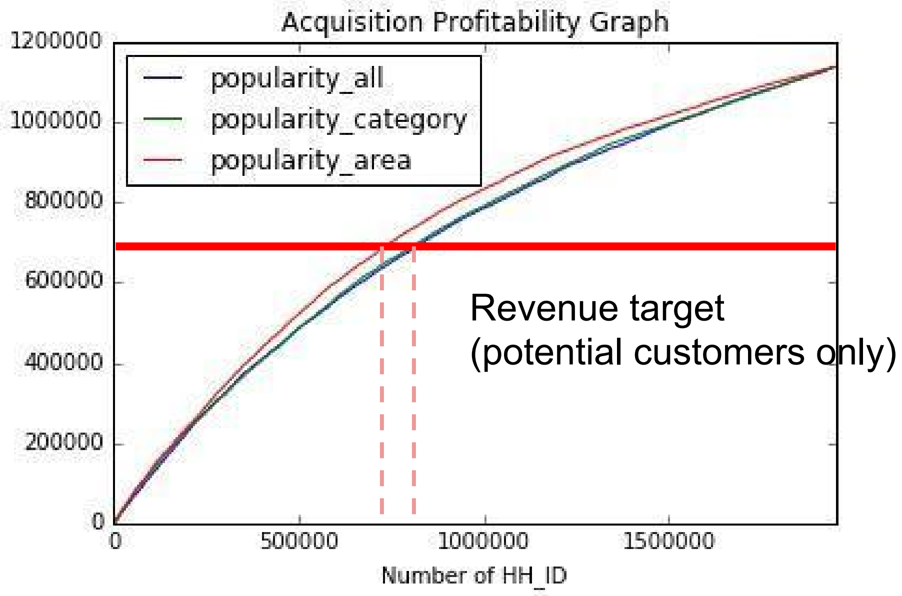
Aggregate Dollars by HH_ID for CompanyID = 36 in testing period (20070701 to 20071231)

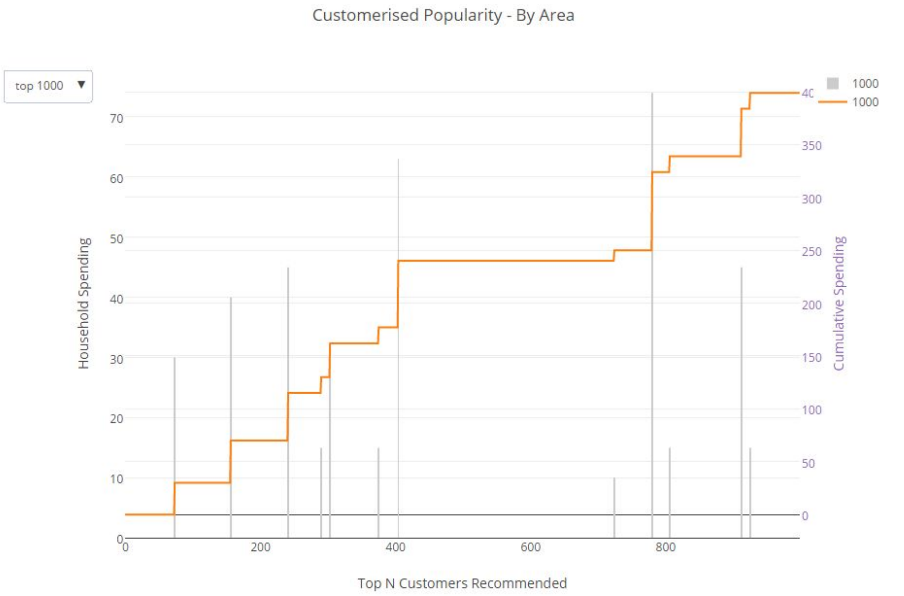
	HH_ID	TopTenCategoryPopularity	TestingRevenue
45150	53255	5067.966423	0.0
1373900	1792511	4858.641617	0.0
1003992	1286456	3954.722506	0.0


Acquisition Profitability Graph
popularity_all
popularity_category
popularity_area
Revenue target
(potential customers only)
Number of HH_ID
0
500000
1000000
1500000
200000
400000
600000
800000
1000000
1200000

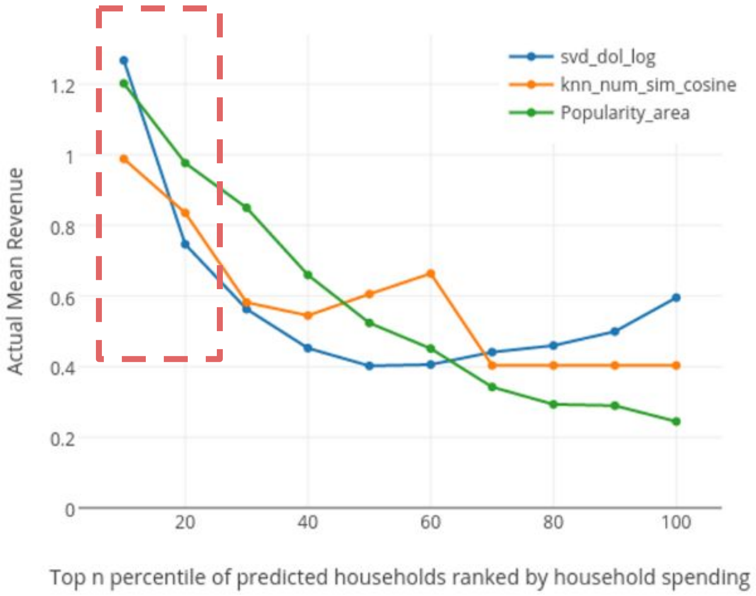

Popularity_all
Popularity_area
Popularity_category
Actual Mean Revenue
Top n percentile of predicted households ranked by household spending
0
0.2
0.4
0.6
0.8
1
1.2
20
40
60
80
100

More on customer's details:
Actual purchases in Jun-Dec 2007





Conclusions



Decile by rui	Actual Revenue Mean (6 months)		
	SVD (dol_log)	Popularity (Top3 areas)	kNN (num, sim: cosine)
1	**1.267**	**1.202**	**0.988**
2	**0.747**	**0.977**	**0.835**
3	0.563	0.850	0.582
4	0.452	0.659	0.545
5	0.402	0.524	0.605
6	0.406	0.451	0.664
7	0.441	0.342	0.403
8	0.460	0.293	0.403
9	0.499	0.289	0.403
10	0.595	0.245	0.403

Limitation & Discussion

- Subset of customers for SlopOne and Item-kNN
  - Computationally expensive: 2.5 million * 2.5 million household similarity matrix
- Better design of worthiness score of households
  - Possible additional data: US census data by ZIP code

ZIP	PctUrban	PopPerSQM	PctUnemployed	AvgHHInc	AvgHValue	PctAge0_4	PctAge5_17
	% Urban Population	Persons Per Sq Mile	% Unemployed Persons	Average Household Income	Average House Value	% Under 5	% 5 to 17
601	57.8	287	30.9	15,892	$63,884	7.7	24.4
602	100	1358.8	22.1	18,915	$66,312	7.7	22.1
604	100	1829.7	27.3	18,756	$77,088	7.3	19.5
606	44	176.1	28.9	16,959	$53,833	8	24.3
610	89.8	755.4	23.2	19,598	$69,374	7.7	20.8
613	93.4	980.3	19.1	21,194	$78,974	7.2	19.1
616	88.2	668.4	24.8	17,101	$58,048	6.8	19.6
617	95.2	1086.9	24	20,425	$61,228	8.6	20.9
622	46.6	273.1	15.9	24,769	$85,508	7.1	16.5

Limitation & Discussion

- GridSearch for better parameters in Co-Clustering
  - More user clusters, item clusters and co-clusters, computationally costly

class `surprise.evaluate.GridSearch`(*algo_class, param_grid, measures=[u'rmse', u'mae'], verbose=1*)

The `GridSearch` class, used to evaluate the performance of an algorithm on various combinations of parameters, and extract the best combination. It is analogous to GridSearchCV from scikit-learn.

Parameters:
- **algo_class** (`AlgoBase`) – A class object of of the algorithm to evaluate.
- **param_grid** (*dict*) – The dictionary has algo_class parameters as keys (string) and list of parameters as the desired values to try. All combinations will be evaluated with desired algorithm.
- **measures** (*list of string*) – The performance measures to compute. Allowed names are function names as defined in the `accuracy` module. Default is `['rmse', 'mae']`.

THANK YOU

Questions&Answer