Lim Yu Zheng
General Assembly (Data Science Immersive 11)
Nov19-Jan20

Capstone Project: E-commerce end-to-end Marketing Strategy

Customer Segmentation, Recommendation System, Market Basket Analysis

Outline

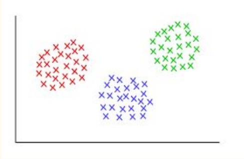
1. 3 Business Problems
2. Data Analytics/Science
    2.1. EDA, Data Cleaning, Feature Engineering
    2.2. Modelling & Evaluation
    2.3. Conclusion
3. Deployment

Dataset: UCI Retail Dataset

542k rows: https://archive.ics.uci.edu/ml/datasets/online+retail (used for this project), or
1.07m rows: https://archive.ics.uci.edu/ml/datasets/Online+Retail+II . 8 features for both

1. Business Problems

P1. Customer segmentation:
How can we **segment customers**, into different clusters, so as to **deliver tailored marketing strategies**, so as to **minimise promotional costs/retain loyal customers/raise revenue**?



P2. Product Recommendation:
How can we also **make personalised product recommendations**, so as to **raise revenue**?

P3. Continued customer engagement:
How can we FURTHER **engage customers through email/promotion campaigns**, so as to **raise revenue**?

P1. Customer Segmentation

Approach: RFM, Kmeans, DB-Scan

Evaluation: Elbow method, Silhouette Score

Conclusion: **Kmeans’ 5 clusters best (Silhouette score: 0.58)**

Marketing Dashboard deployed online on ‘Tableau Public’

RFM:
Split into eg. 4 quartiles, then rank based on %tile

	Recency (days since last purchase)	R	Freq (no. of transactions)	F	Monetary (total spend)	M	RFM combined
Person A	0	1	10	1	$10	1	111
Person B	3	2	7	2	$7	2	222
Person C	7	3	3	3	$3	3	333
Person D	10	4	0	4	$0	4	444

EDA, Data cleaning, Feature Engineering

Original dataset:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	2010-01-12 09:41:00	27.50	14527	United Kingdom
154	C536383	35004C	SET-OF-3-COLOURED--FLYING-DUCKS	-1	2010-01-12 09:49:00	4.65	15311	United Kingdom
235	C536391	22556	PLASTERS-IN-TIN-CIRCUS-PARADE	-12	2010-01-12 10:24:00	1.65	17548	United Kingdom
236	C536391	35004	SET-OF-3-COLOURED--FLYING-DUCKS	-24	2010-01-12 10:24:00	0.29		United Kingdom
237	C536391	22556	PLASTERS-IN-TIN-CIRCUS	-24	2010-01-12 10:24:00	0.29	17548	United Kingdom

-removed nulls for CustomerID, because without ID we can't identify them

-removed duplicated rows

-identical products have variations in StockCode/Description, hence commonized StockCode, Description for such products

-Some product purchases have a mix of +ve/-ve Quantities, hence summed Customers' purchases for each product, and removed those that are still sum -ve

-created TotalPrice column, via UnitPrice x Summed Quantity

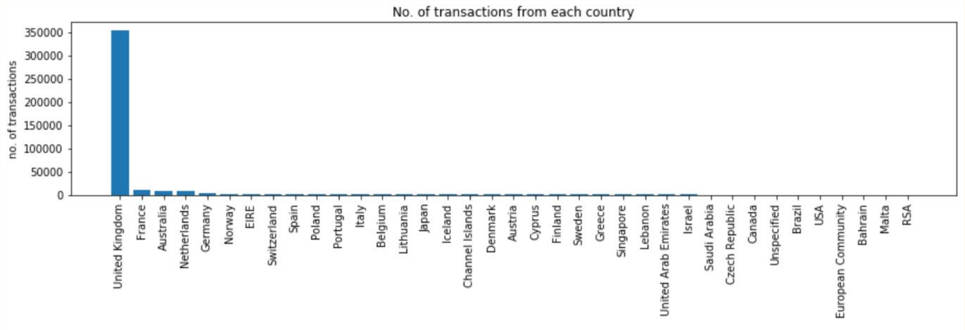
-created Year, Month columns from InvoiceDate

-created R,F,M columns for each customer

-scaled R,F,M values (for Kmeans, DB-Scan)

Post-cleaning EDA

```
no. of unique transactions: 21235
no. of unique items: 3620
no. of unique Countries bought from: 37
no. of unique customers: 4326
no. of unique years: 2
no. of unique months: 12
```

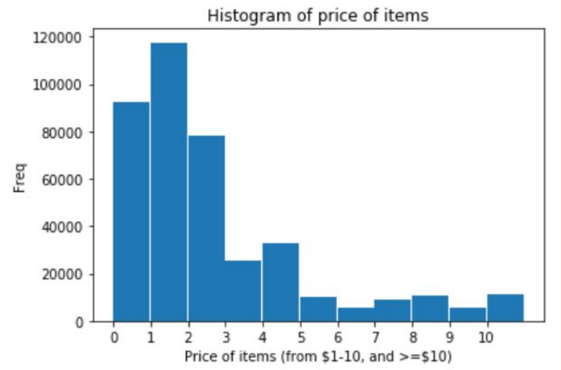



```
Min date: 2010-01-12 08:26:00
Max date: 2011-12-10 17:19:00
```

```
Distribution of years for all transactions:

absolute numbers...
 2011    371347
2010     25936
Name: Year, dtype: int64

normalised...
 2011    0.934717
2010    0.065283
Name: Year, dtype: float64
```



RFM (baseline model)

CustomerID	recency	frequency	monetary
12347	41	182	4310.00
12348	77	31	1797.24
12349	20	73	1757.55
12350	312	17	334.40
12352	74	89	1755.55

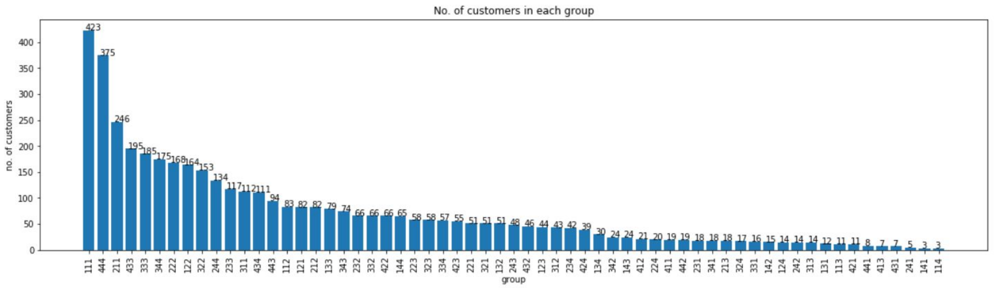
CustomerID	recency	frequency	monetary	combined
12347	2	1	1	211
12348	3	3	1	331
12349	1	2	1	121
12350	4	4	3	443
12352	3	2	2	322

Downsides:

-very similar spenders at edge of eg. M tiers 1&2, may wrongfully be sorted into different groups

-HUUUGE no. of groups (up to 4x4x4=64 combinations) to manage for marketing team

No. of groups: 61

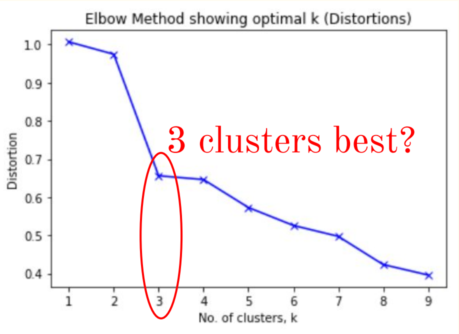


Evaluate other models…

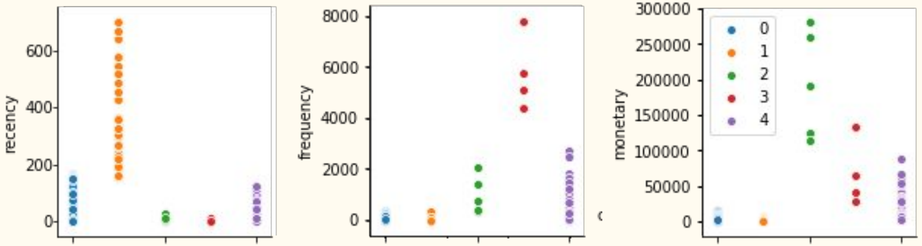
Kmeans

Gridsearch. Try 1-10 clusters…

Elbow:



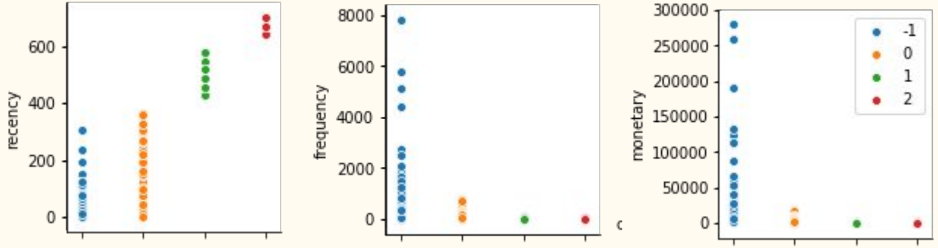
Silhouette score: 5 clusters best? (best score of 0.58)



DB-Scan

Gridsearch. Try eps: [0.02, 0.1, 0.4], min_samples: [2,5,9]

Silhouette score: 3 clusters best? (best score of 0.62, best params eps=0.4, min_samples=9)



BUT!!!
As rule of thumb, 3 clusters is too few! Typically ~8!

SO… Use 5 clusters.

Rank clusters' median RFM into RFM tiers...

P1. Conclusion & Recommendations

Best model: Kmeans

```
no. of members per cluster:
0    3019
1    1062
4     236
2       5
3       4
Name: cluster, dtype: int64

median of RFM for each cluster:
cluster          0       1          2          3        4
recency      40.00  244.00       4.00      3.000    13.00
frequency    50.00   19.00     772.00   5456.000   417.00
monetary    808.87  304.92  189575.53  52530.255  6705.05

median of RFM for each cluster (R/F/M binned into 4 quartiles):
cluster    0  1  2  3  4
recency    3  4  1  1  2
frequency  4  4  2  1  3
monetary   4  4  1  2  3
```

Industry's marketing-strategy recommendations for each cluster:

A. Core - Your Best Customers (none from kmeans!)
- RFM group: 111
- Who They Are: Highly engaged customers who have bought the most recent, the most often, and generated the most revenue.
- Marketing Strategies: Focus on loyalty programs and new product introductions. These customers have proven to have a higher willingness to pay, so don't use discount pricing to generate incremental sales. Instead, focus on value added offers through product recommendations based on previous purchases.

B. Loyal - Your Most Loyal Customers (kmeans cluster 3!)
- RFM group: X1X
- Who They Are: Customers who buy the most often from your store.
- Marketing Strategies: Loyalty programs are effective for these repeat visitors. Advocacy programs and reviews are also common X1X strategies. Lastly, consider rewarding these customers with Free Shipping or other like benefits.

C. Whales - Your Highest Paying Customers (kmeans cluster 2!)
- RFM group: XX1
- Who They Are: Customers who have generated the most revenue for your store.
- Marketing Strategies: These customers have demonstrated a high willingness to pay. Consider premium offers, subscription tiers, luxury products, or value add cross/up-sells. Don't waste margin on discounts.

D. Promising - Faithful customers (none from kmeans!)
- RFM group: X13, X14
- Who They Are: Customers who return often, but do not spend a lot.
- Marketing Strategies: You've already succeeded in creating loyalty. Focus on increasing monetization through product recommendations based on past purchases and incentives tied to spending thresholds (pegged to your store AOV).

E. Rookies - Your Newest Customers (none from kmeans!)
- RFM group: 14X
- Who They Are: First time buyers on your site.
- Marketing Strategies: Most customers never graduate to loyal. Having clear strategies in place for first time buyers such as triggered welcome emails will pay dividends.

F. Slipping - Once Loyal, Now Almost Gone (none from kmeans!)
- RFM group: 44(1/2)
- Who They Are: Great past customers who haven't bought in awhile.
- Marketing Strategies: Customers leave for a variety of reasons. Depending on your situation price deals, new product launches, or other retention strategies.

G. Lost - Not worth keeping (kmeans cluster 0,1!)
- RFM group: (3/4)44
- Who They Are: Lost customers who spent little, infrequently
- Marketing Strategies: Don't waste huge marketing efforts on them

H. Others - (kmeans cluster 4!)
- RFM group: others
- Who They Are: others
- Marketing Strategies: No particular strategy. Maintain status quo

https://www.barilliance.com/rfm-analysis/#tab-con-1

P2. Product Recommendation

Approach: Collaborative Filtering (data only has purchased qty: 'implicit feedback')
Evaluation: Mean Precision@k (& less importantly Mean Recall@k, Mean AUC)
Conclusion: **Best model achieved <u>mp@k=88%</u>** (& mr@k=51%, mAUC=100%)

Mock E-commerce website with recommendation engine deployed on 'http://uci-retail-recommender.herokuapp.com/'

Used fashion company Lyst's LightFM library, for implicit feedback for recommendation engines.

-matrix factorisation

-inbuilt mp@k, mr@k, mauc scoring functions

-inbuilt prediction/recommendation functions

-outperforms both collaborative and content-based models in cold-start or sparse interaction data scenarios

EDA, Feature Engineering

	CustomerID	StockCode	Description	UnitPrice	Quantity	CustomerID_cat	StockCode_cat
0	12347	16008	SMALL-FOLDING-SCISSOR(POINTED-EDGE)	0.25	24	0	24
1	12347	17021	NAMASTE-SWAGAT-INCENSE	0.30	36	0	86
2	12347	20665	RED-RETROSPOT-PURSE	2.95	6	0	129
3	12347	20719	WOODLAND-CHARLOTTE-BAG	0.85	40	0	165
4	12347	20780	BLACK-EAR-MUFF-HEADPHONES	4.65	12	0	204
5	12347	20782	CAMOUFLAGE-EAR-MUFF-HEADPHONES	5.49	6	0	206

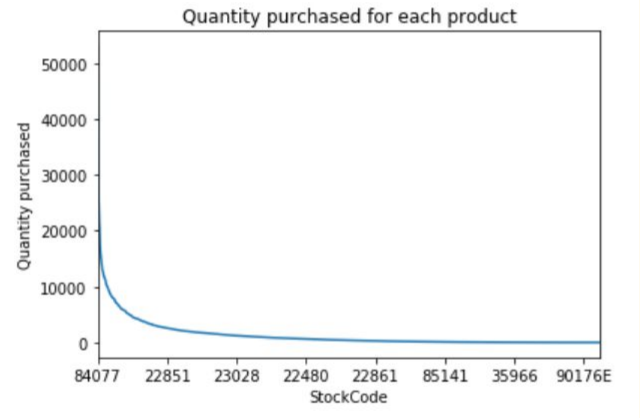
-already cleaned prior

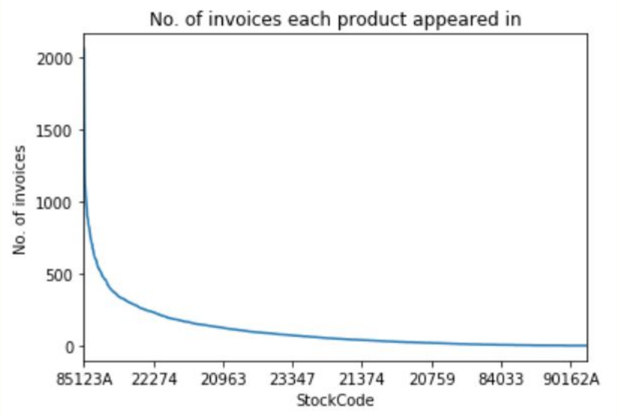
-aggregated each customer's purchases of the same item, together, then binarized purchases into 1 (bought that product) and 0 (didn't buy it)

-converted to sparse matrix (required by lightfm)

-noticed long tail effect (only a few products have interactions with customers)

Many products not purchased in large amounts...





...nor frequently

BPR (baseline model)

	mp@k	mr@k	mauc
BPR (baseline)	**0.37**	0.06	0.84

-BPR model: it optimises AUC score

-Baseline because: AUC measures the quality of the overall ranking (though **without regard for top-k rankings**). That by and large, most recommendations are relevant.

BUT!!!

-**p@k is more important**, because customers won't scroll to page 99 to find your recommendations. Need to **prioritize relevant items at top of 1st page**, to best ensure their purchase. Hence, use...

-WARP, k-OS WARP models: they optimise mp@k by prioritizing top-k recommended items

P2. Model evaluation & Conclusion

-set k=10. Meaning, ‘Serve top 10 recommendations’

-Baseline BPR vs the rest.

  -default hyperparameters

  -10 epochs

	mp@k	mr@k	mauc
BPR (baseline)	**0.37**	0.06	0.84
WARP	**0.34**	0.06	0.89
k-OS WARP	**0.27**	0.05	0.85

-Gridsearch. Hyperparameter tuning of WARP, k-OS WARP

  -models: [WARP, k-OS WARP]

  -no. of latent features: [100,150,200]

  -learning schedules: [adagrad, adadelta]

  -50 epochs

Best model is...

WARP, 200, adadelta	**0.88**	0.51	0.100

Test our recommendation engine!

Eg. Customer 5

Previously bought:

```
userid: 5

bought_prod with desc:
   StockCode                          Description
0      22890   NOVELTY-BISCUITS-CAKE-STAND-3-TIER
1      37446   MINI-CAKE-STAND-WITH-HANGING-CAKES
2      37449   CERAMIC-CAKE-STAND-&-HANGING-CAKES
3      37450    CERAMIC-CAKE-BOWL-&-HANGING-CAKES
```

'Closeness' scores of recommended products...

Recommended products:

```
scores for top recommended products (higher is better):
 [1.5638829469680786, 1.5414295196533203, 1.5290290117263794, 1.4790804386138916, 1.4465149641036987, 1.4418790340423584,
1.4328478574752808, 1.3941409587860107, 1.3775039911270142, 1.3757057189941406]

top recommended products:
   StockCode                          Description
0      37450    CERAMIC-CAKE-BOWL-&-HANGING-CAKES
1      22890   NOVELTY-BISCUITS-CAKE-STAND-3-TIER
2      37446   MINI-CAKE-STAND-WITH-HANGING-CAKES
3      37449   CERAMIC-CAKE-STAND-&-HANGING-CAKES
4      22055   MINI-CAKE-STAND--HANGING-STRAWBERY
5      22649         STRAWBERRY-FAIRY-CAKE-TEAPOT
6      21232        STRAWBERRY-CERAMIC-TRINKET-POT
7      37448       CERAMIC-CAKE-DESIGN-SPOTTED-MUG
8      22063  CERAMIC-BOWL-WITH-STRAWBERRY-DESIGN
9      37447     CERAMIC-CAKE-DESIGN-SPOTTED-PLATE
```

Max possible p@k of 40% obtained!
(repeat & take mean of everyone's p@k)

Other recommendations also highly cake-related! Looks sensible!

Finally, scrape images using product Descriptions, & deploy on Heroku

Compare with baseline model...

Eg. Customer 5

Previously bought:

```
userid: 5

bought_prod with desc:
   StockCode                          Description
0      22890  NOVELTY-BISCUITS-CAKE-STAND-3-TIER
1      37446  MINI-CAKE-STAND-WITH-HANGING-CAKES
2      37449  CERAMIC-CAKE-STAND-&-HANGING-CAKES
3      37450   CERAMIC-CAKE-BOWL-&-HANGING-CAKES
```

‘Closeness’ scores of recommended products...

Recommended products:

```
scores for top recommended products (higher is better):
 [1.2551584243774414, 1.2278729677200317, 1.179606556892395, 1.1503195762634277, 1.1157306432724, 1.099589467048645, 1.0934269428253174, 1.090859055519104, 1.088451862335205, 1.0861166715621948]

top recommended products:
   StockCode                       Description
0      22998        TRAVEL-CARD-WALLET-KEEP-CALM
1      22115              METAL-SIGN-EMPIRE-TEA
2      22996  TRAVEL-CARD-WALLET-VINTAGE-TICKET
3      21175       GIN-AND-TONIC-DIET-METAL-SIGN
4          M                              Manual
5      82580                BATHROOM-METAL-SIGN
6      21165        BEWARE-OF-THE-CAT-METAL-SIGN
7      21034        REX-CASH&CARRY-JUMBO-SHOPPER
8      82551              LAUNDRY-15C-METAL-SIGN
9      21172                    PARTY-METAL-SIGN
```

Absolutely no hits! Nor seemingly sensible recommendations!

P3. Continued Customer Engagement

Approach: Apriori algorithm. Used for Market Basket Analysis (MBA)

Evaluation: Support, Lift

Conclusion: See generated **list of highly associated products.** **<u>For use in email/marketing campaigns (AND on website too) to bundle product recommendations together</u>**

Deployed in email-marketing campaign, & mock e-commerce website.

Different from content-based/collaborative filtering!

Evaluation metrics for MBA

-Support (range [0,1]): fraction of all transactions that contain both products A and B.

Higher better. Hence more data to draw conclusions about their relationship.

-Lift (range [0,infinite)): greater lift indicate stronger associations between A and B.

>1 better. Indicates that it's not just a coincidence, and there is indeed a high association between items A and B. High chance of buying B if the customer has already bought A.

Product associations generated...

Product A

Product B

Sort support from high to low

Seek lift>1

Associated products!

No. of associations: 131330

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
3120	((22086, PAPER-CHAIN-KIT-50-S-CHRISTMAS))	((22910, PAPER-CHAIN-KIT-VINTAGE-CHRISTMAS))	0.141239	0.107721	0.074896	0.530278	4.922712
3121	((22910, PAPER-CHAIN-KIT-VINTAGE-CHRISTMAS))	((22086, PAPER-CHAIN-KIT-50-S-CHRISTMAS))	0.107721	0.141239	0.074896	0.695279	4.922712
4553	((22423, REGENCY-CAKESTAND-3-TIER))	((22699, ROSES-REGENCY-TEACUP-AND-SAUCER))	0.201110	0.097319	0.073047	0.363218	3.732263
4552	((22699, ROSES-REGENCY-TEACUP-AND-SAUCER))	((22423, REGENCY-CAKESTAND-3-TIER))	0.097319	0.201110	0.073047	0.750594	3.732263

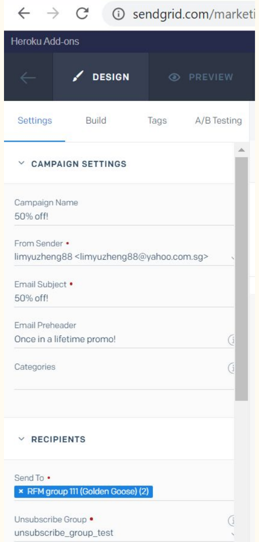
P3. Conclusions & Recommendations

Marketing strategies! When promoting Product A, also cross-sell its associated Product B, to boost overall sales, for:

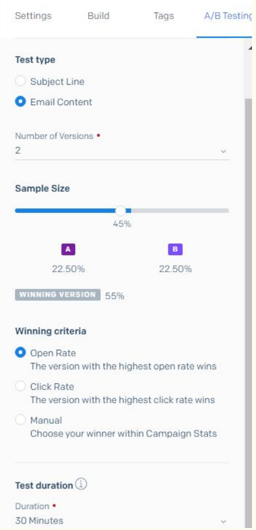
1. Email marketing
2. General marketing
3. 'Checkout' page on e-commerce website

Email-marketing campaign demo

-Heroku's 'Sendgrid Marketing Campaign' add-on. A/B testing & analytics

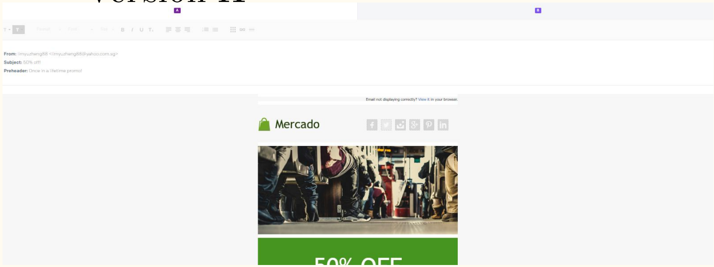


Choose customer segment to send to

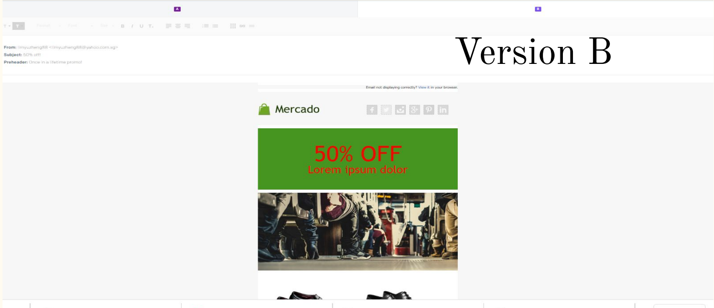


A/B testing: vary winning criteria, campaign duration etc

Version A



Vary email content/design/header etc

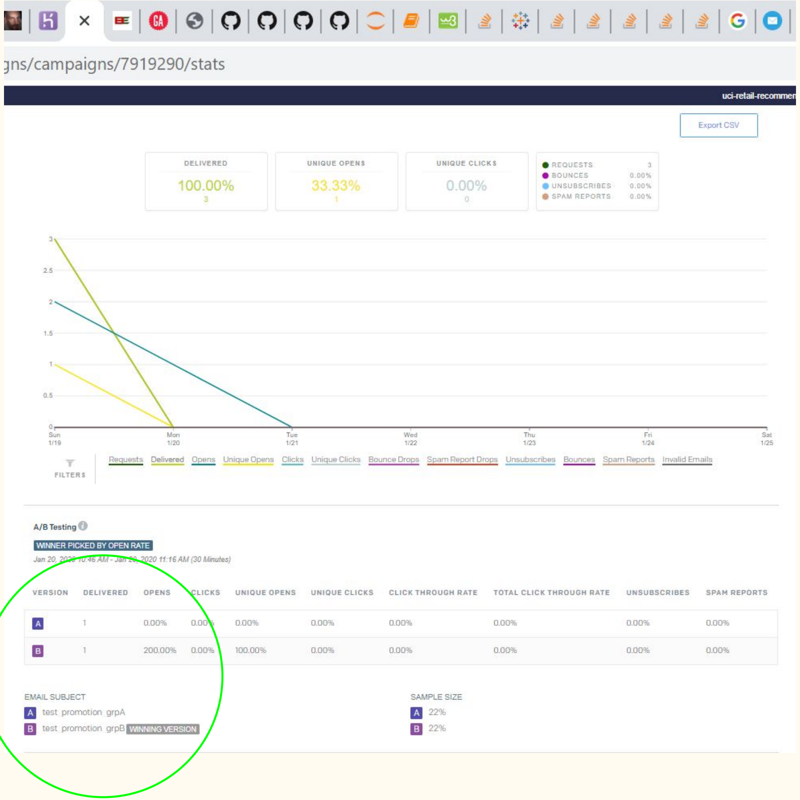


Version B

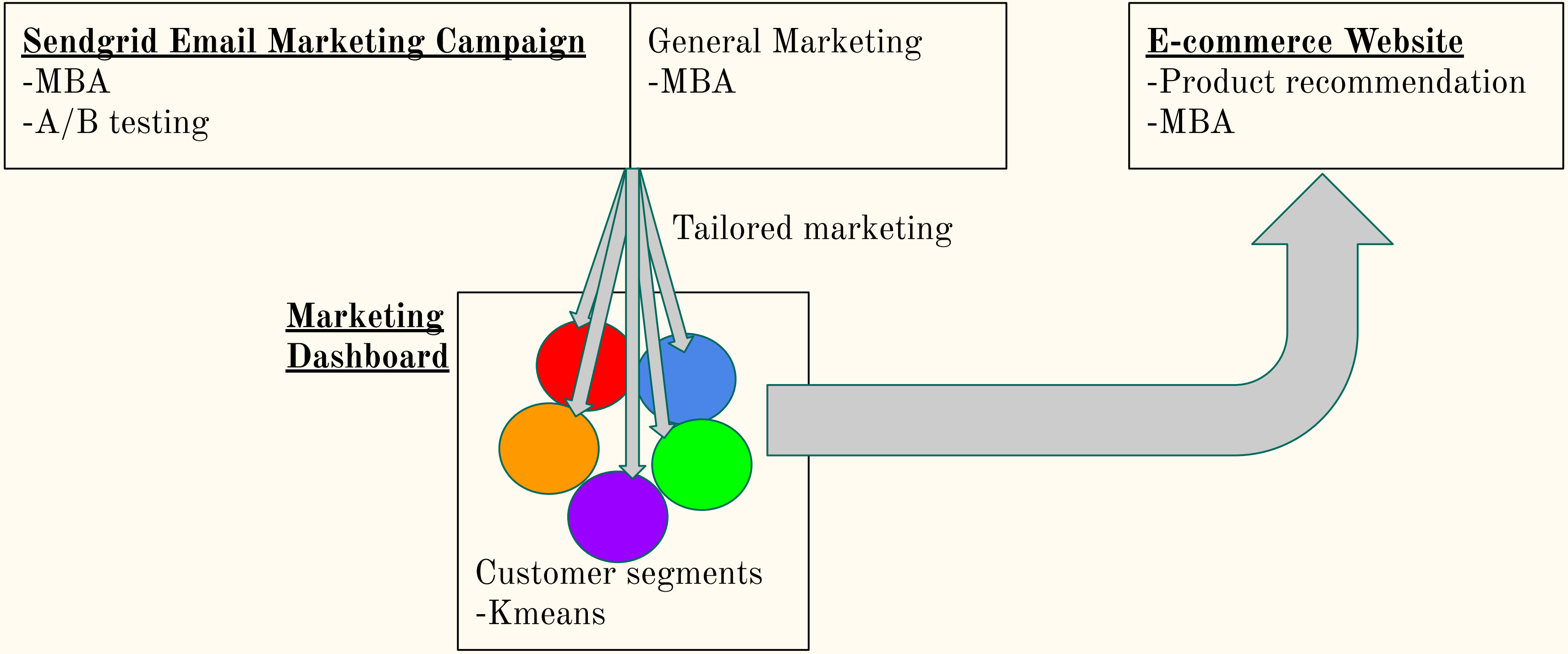
Winner...

View winner of email campaign A/B testing

Version B wins!



Putting everything together...



Deployment

-Marketing dashboard:
https://public.tableau.com/profile/lim.yu.zheng#!/vizhome/UCI-retail-recommender Marketing-Dashboard/Story1?publish=yes

-Mock e-commerce website with collaborative-filtering recommendation engine, and Market Basket Analysis recommendations:
http://uci-retail-recommender.herokuapp.com/

-Email marketing campaign control page is accessed by my id & pswd, not shareable