US Beer Assortment Optimization

2018 Capstone Project with Anheuser-Busch InBev
New York, New York

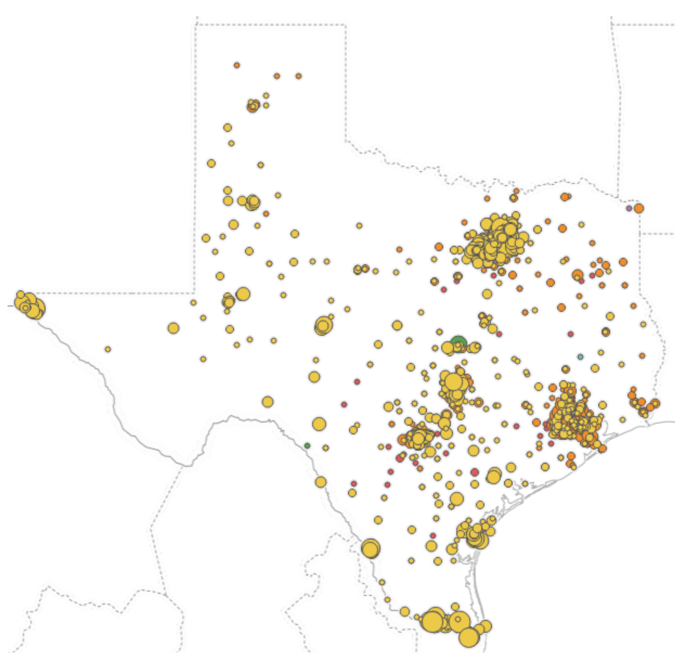
Team: Daria Brauner and Thomas Littrell
Advisor: Prof. Dimitris Bertsimas
PhD collaborators: Brad Sturt & Chris McCord
Thanks to: Brent Hayden, Emily Shapiro, Siyu Wang, and Debarshi Indra

1. Challenge

What products should we put into a store to make the most money while accounting for demand substitution?

2. Scope and Timeline

We worked with the U.S. business and focused on making assortment recommendations for chain accounts in Texas. These accounts are where we have the most data and can have the most immediate impact given that ABI representatives help chains design assortments every 6 months.



Feb.: Project start — Jun.: Full time in NYC — Aug. 2nd: Presentation at RM conference — Aug. 17th: Project end

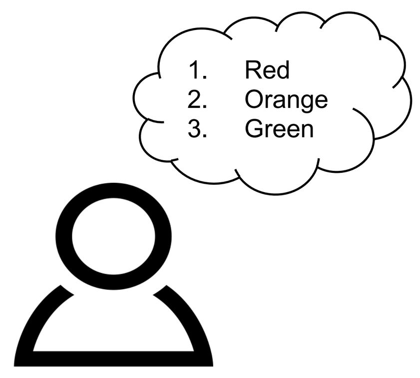
3. Data

Our dataset is a combination of internal and third party data. The main data source we used is the data provider JDA's beer retail data set, which provides sales information at the SKU level by store for around 3000 chain accounts in Texas. We supplemented the sales data with information on both stores and products. In particular, we used data provider IRI's sales data to impute price information for all products in our data.

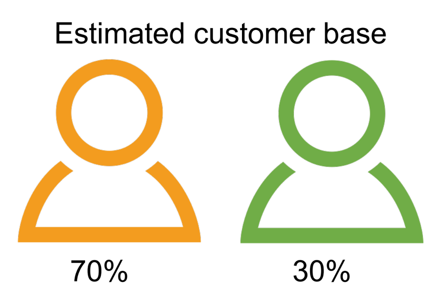
For computation reasons, we aggregated all products to product groups, providing the highest level of visibility for the largest and fastest growing brands.

4. Methodology

Our methodology consists of two pieces.[1] First, we estimate a consumer preference model to capture substitution patterns between products. Second, we use the consumer preference model as an input to a constrained optimization that searches over all the possible assortments and picks the one that gives the most expected revenue while respecting business constraints.



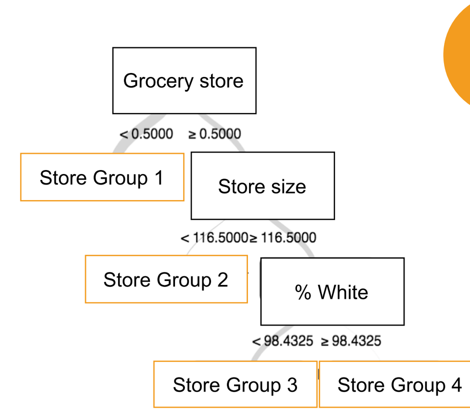
1 We assume that customers have a ranking, which they come to for any reason, of all products and buy their most preferred product from all the available products in a store.



2 Different rankings define different customer types and we solve a linear program to find the proportion of each type in a store's customer base that best explains sales data.

Beer	Share
Beer 1	45%
Beer 2	17%
Beer 3	38%

3 Given the customer base and rankings, we can estimate shares for new assortments. Changes in shares as assortment changes captures substitution.



4 Since stores have different customer bases, we train an optimal tree that learns from the data how to best cluster stores based on demographics for the choice model.

5 A mixed integer optimization problem looks at the expected value of all assortments (predicted share times price summed over products) and picks the best one.

Business rules:
- Assortment size
- Package size
- Disallowed products
- ABI only
- Number of changes

6 Working with subject matter experts, we add additional (optional) constraints to reflect business rules and improve the usefulness of our recommendations.

[1]Model taken from Bertsimas, Dimitris, and Velibor V. Mišic. "Data-driven assortment optimization." submitted to Management Science (2015).

5. Results

The model outputs recommendations for what to swap into and out of a chain store's current assortment. Below, we show the recommendations made for different numbers of swaps for a convenience store and highlight how demand substitution affects our recommendations. After letting the model make as many changes as it wants, we also observe that relatively few assortment changes can realize most of the overall benefit.

+1%

Estimated opportunity to increase unites moved for a typical store

The Miller pack pulls demand from ABI products

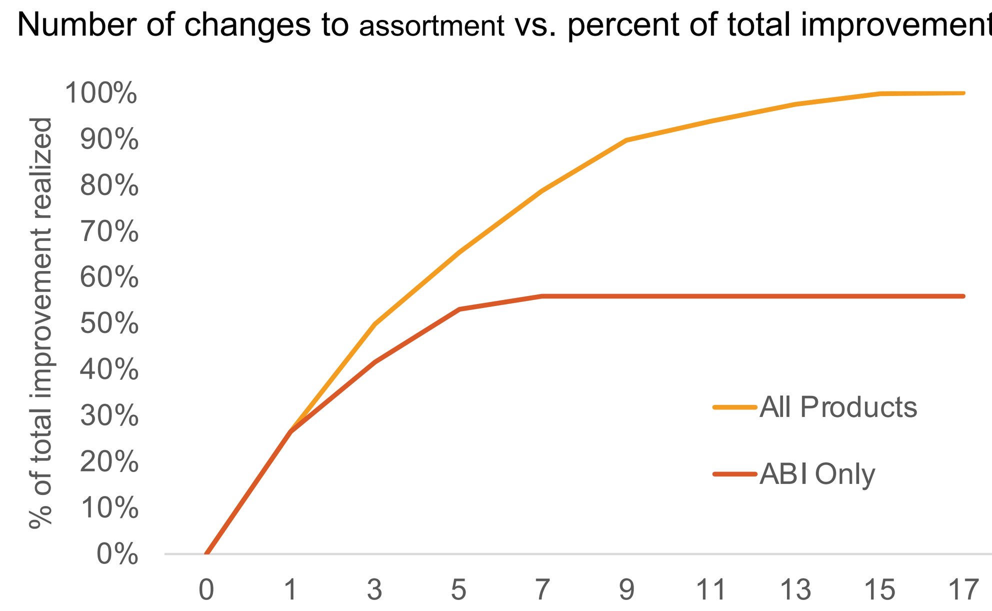
Michelob Ultra is a higher price brand

Swaps	Products to add	Products to remove
1	Bud Light large*	Miller Lite 18/12C
3	Bud Light large* Bud Light 15/16C Bud Light 24/12C	Bud Light 12/12B Miller Lite 18/12C Miller Lite 12/12B
5	Bud Light large* Michelob Ultra large* Bud Light 24/12C Bud Light 15/16C Bud Light 24/12BBB	Budweiser medium* Miller Lite 18/12C Budweiser small* Miller Lite small* Bud Light 12/12B

*Aggregate product

Demand substitution patterns show a willingness to buy larger SKUs

Number of changes to assortment vs. percent of total improvement



6. Next Steps

There are four areas in which the model can be extended:

1. Adding information on the market segment served by each chain
2. Using unit movement in the objective function
3. Using more refined space constraints
4. Accounting for inventory requirements

After refining the model further, we recommending running a field experiment with a partner chain in Texas.

Demand Forecasting for a Luxury Fashion Retailer

BCG Team: Arun Ravindran & Anton van Pamel
MIT Mentor: Robert Freund
Location: Boston, Massachusetts

Project Overview

Project Importance

- Luxury retailers make little revenue from ready-to-wear clothes.
- Approximately 90% of revenue comes from handbags, shoes, accessories, and fragrances.
- Gross margins for handbags are often the highest across all departments, so an accurate demand forecast is crucial.

Project Scope

- Our project was forecasting demand for women's handbags in their European stores.
- Specifically, we created a model to predict demand for handbags that are part of the new seasonal collection, meaning they have no historical sales.

Below is the criteria on which we filtered the data.

- Women's handbags
- Full-price items
- European stores
- Retail locations
- Gross sales
- New Collection

Project Timeline

January - February	→ Exploratory Data Analysis → Initial Demand Forecasting Models
March - April	→ Feature Engineering → Constructing Panel Data
May - June	→ Pairwise Comparison Research → Efficient Algorithm Implementation
July - August	→ Sophisticated Demand Forecasting Models → Summer Capstone Showcase

Data Overview

Raw Data

We were given four datasets:

Stores | Products | Transactions | Inventory

- We merged these four datasets and then filtered the data to reflect the scope of our project.
- Our merged dataframe had approximately 45,000 rows. Within that dataframe, there are over 1,300 unique stock keeping units (SKUs) and about 120 unique store locations.

Store Clusters

The client provided us with five store clusters, labeled A through E. We analyzed each of these clusters and created a short description for each.

- A — **Flagship Store:** highest amount of stock and sales with the most expensive purchases
- B — **Large City Locations**: comparable sell-through rate to the flagship store, but with fewer overall sales and less expensive sales
- C — **Resort Locations:** low stock and low sell-through rate, but high prices and located in resort towns
- D — **Traditional City Locations:** do not carry the highest price-point, but has a high sell-through rate and high revenue to square footage ratio
- E — **Low Volume Stores:** assortment of low volume store with the overall lowest price point (includes airport locations)

Data Processing

Clustering Data using *k*-Prototypes

- We applied clustering to our data using *k*-prototypes, which integrates *k*-means and *k*-modes algorithms to cluster both continuous and categorical variables.
- We selected the number of clusters by validating on the model's overall performance.
- These clusters helped us build new features, such as historical sales and stock-made by cluster.

Reducing Proportion of Null Values

- We imputed missing values in the dataframe using analytical expertise and ETL techniques.
- Using our analytical expertise, for example, we inspected the data and replaced null values with zero for binary features.
- We used ETL techniques to create an aggregated feature, and by merging datasets on this aggregated feature, the number of null values was dramatically reduced.

Dummifying Data and Deleting a Degree of Freedom

- We dummified the categorical variables and, when doing so, we deleted the extra degree of freedom.
- This approach decreased the complexity of the data and increased the performance of our model.

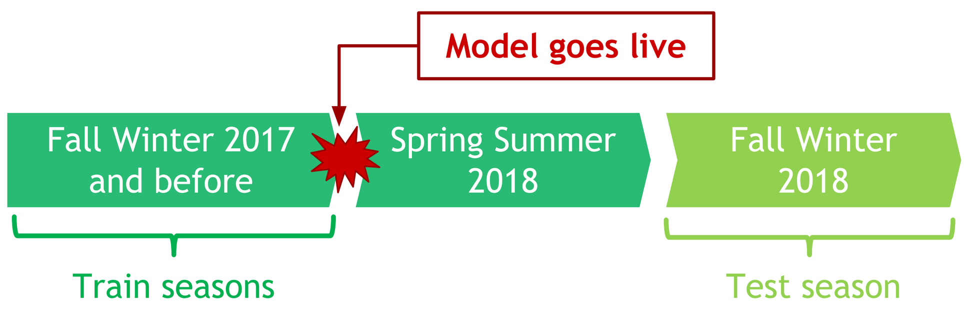
For example, the variable `Ornaments` has four levels: `Pearls`, `Studs`, `Swarovski`, and `None`. If we reduced these dummified features to `Pearls on handbag`, `Studs on handbag`, and `Swarovski on handbag`, all of the information can be captured.

Project Challenges

We faced two main challenges:
- Forecasting is done in advance to allow for manufacturing.
- There are few similarities between the train and test set.

Manufacturing Timeline

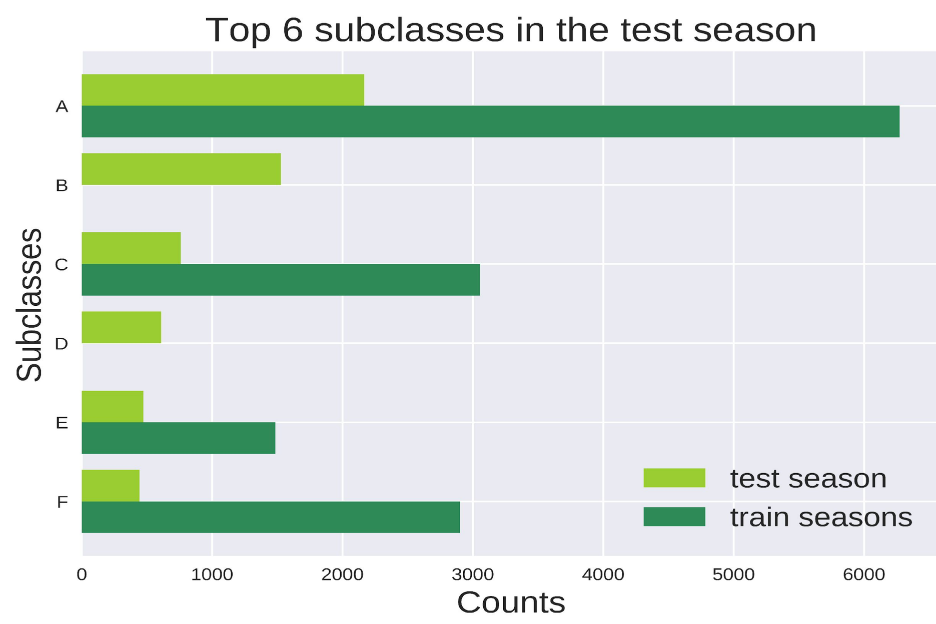
- Demand forecasting must be done six months before the season begins because of the manufacturing timeline.
- We predicted handbag demand for the Fall-Winter 2018 season.
- We trained our modeling using data from the Fall-Winter 2017 season and earlier.
- However, to allow for manufacturing and shipping, our model has to goes live at the beginning of the Spring-Summer 2018 season in order to predict demand for Fall-Winter 2018.



Dissimilarities between Train and Test

- Significant discrepancies exist between the train and test sets, which makes accurate predictions difficult for the test set.
- We inspected the number of SKUs made per subclass for the two datasets to asses the dissimilarity.

Below is a plot showing that some subclasses are prevalent in the test set but absent from the train set.



Feature Engineering

Historical Features

We created historical features by lagging the last two seasons of data. Because these SKUs are part of the new collection, however, we have no historical sales for the SKUs so we lagged on product category features (i.e. type of material, color of bag, etc.).

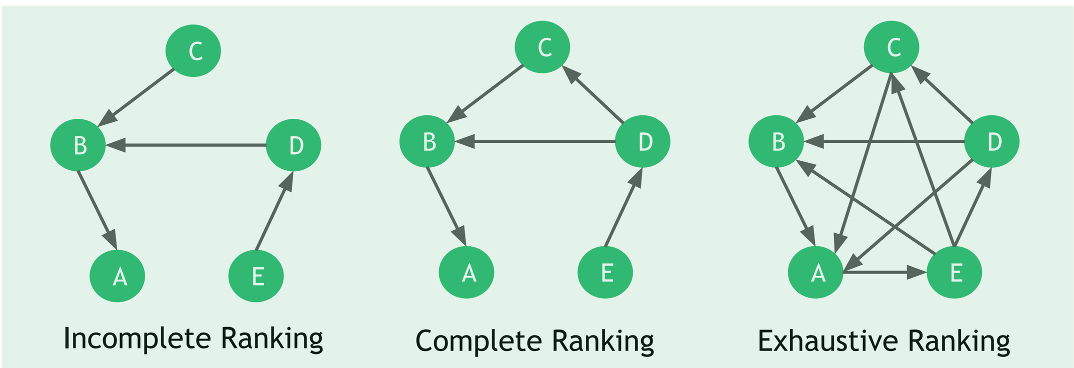
- Sales for that category for season-1
- Sales for that category for season-2
- Stock-made for that category for season-1
- Stock-made for that category for season-2
- Sell-through rate for that category for season-1
- Sell-through rate for that category for season-2

Product and Store Features

- Aggregated style and color features were created to decrease dissimilarity between the train and test sets.
- Consider a granular five-digit color code for a green bag, where the first three digits indicate that it is green, the next digit indicates the brightness of the shade, and the final digit signifies the exact hue of green.
- By reducing this feature to an aggregated three-digit code, we are able to find more similarities between the train and test set.

SKU Popularity: A Bayesian Approach

- We created the SKU popularity feature and included it in the model before its training, like in a Bayesian framework.
- To create this feature, store managers will perform pairwise comparisons of SKUs, allowing us to include human intelligence to our machine learning model.
- It is too time consuming to compare every pair of SKUs to obtain a global ranking. Therefore, we use an adaptive ranking algorithm to select the next pair to compare in order to minimize the total number of comparisons needed.



- In our algorithm, we use directed graphs: each node represents a SKU and an edge is added between two nodes when those SKUs have been compared.
- Our ranking can be obtained if all nodes are connected in our directed graph, as shown above.

Model and Results

Model Selection: Random Forest

- We tried three models: Elastic Net, CART, and Random Forest.
- We evaluated these three models on mean absolute error (MAE) and mean absolute percentage error (MAPE).
- We selected the Random Forest model not only because it has the best performance, but also because it is interpretable.

	Elastic Net	CART	Random Forest
MAE	6.08	5.64	**4.74**
MAPE	138%	147%	**130%**

Feature Importance

- Below are the most significant features in our model.
- We created all of these top features except for SKU price.
- This emphasized to us the importance of feature engineering to extract important signals from the data to feed into the model.

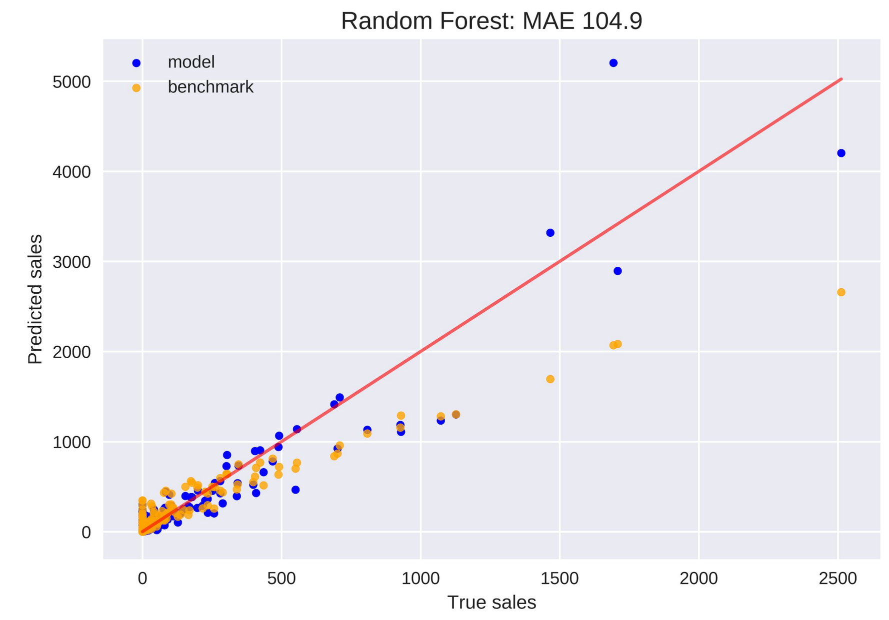
Rank of Importance	Feature
1	SKU popularity
2	Historical sales
3	Store popularity
4	Number of competing SKUs
5	SKU price
6	SKU launch month

Performance Compared to Benchmark

- In order to convince the client of the validity of our model, we compared its performance to the benchmark, which is the amount of stock made by the client per SKU for each season.
- We assessed the performance of our model and the benchmark using MAE and price MAE, which is MAE weighted by SKU price.

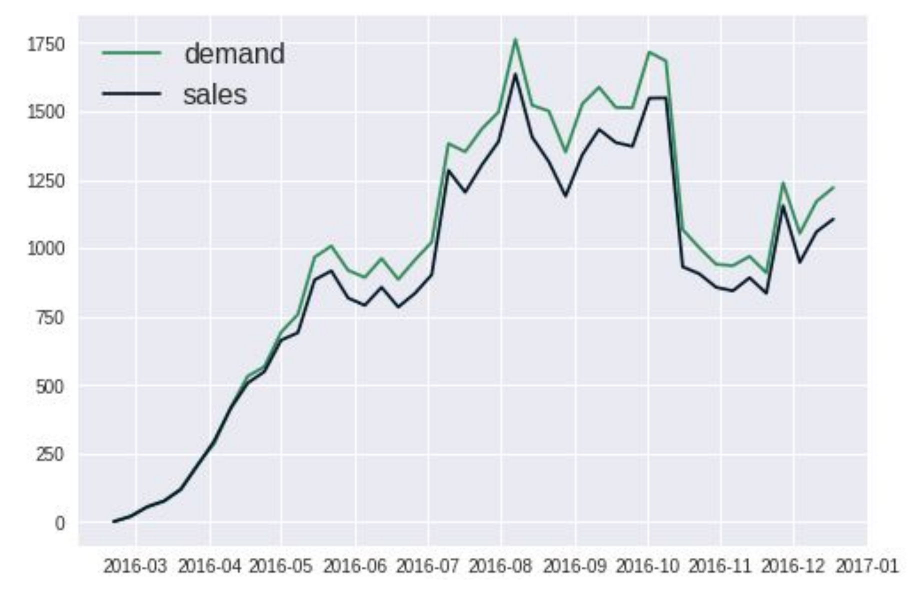
	Benchmark	Model
MAE	126.8	104.9
Priced MAE	204k €	168k €

- The client wants to make twice as much stock as they expect to sell to buffer for supply-chain logistics and ensure that stores are fully stocked. Therefore the objective of our model, which is the red line below, is to predict *2 x sales*.
- In the figure below, we plot our model's predictions in blue and the benchmark's predictions in orange.



Recommendation: Potential Demand

- Sales are a proxy for demand since stock-outs could have caused fewer sales to occur.
- We trained a Random Forest model to predict sales for which there were no stock-outs and then predicted sales for weeks in which there were stock-outs.
- Below is the plot for the Fall-Winter 2016 season, for which we predict that demand is 10.01% higher than sales. The MAE of this model is 0.461 and MAPE is 27.3%.



- Demand and sales are equal at the beginning of the season because no stock-outs have yet occurred, so the client is meeting all demand. Later in the season, however, there are many instances in which SKUs are out of stock.

Impact

Our Capstone project resulted in a better performing forecasting model in comparison to the client's model. This superior performance is the result of our data insights, feature engineering, and model selection. Ultimately, better forecasting improves the organizational and business performance, resulting in the following benefits:

- **Fewer missed sales:** accurately forecasting demand will ensure that inventory is in the right place at the right time.
- **Lower working capital:** the client can operate with less inventory because of confidence in demand projections.
- **Less waste:** the client is more likely to sell stock at full-price, without having to discount it because it is no longer part of the new season's collection.
- **Improved customer service:** with a deeper understanding of customer demand and unique store selling behaviors, the client can effectively deploy inventory to provide higher sell-through rates, improved on-time availability, and fewer stock-outs.

BCG Gamma | Trend Forecast

Location: Boston, MA (U.S.)

The Project

CLIENT

- Hopes to capture consumer sentiment and preferences
- Proposed we forecast which trends will hit market in a year
- Used to guide buying and product development strategy

BCG GAMMA

- How is consumer sentiment quantified?
- Which models to forecast with?
- Team comprised of data scientists and consultants

The Team

Faculty Mentor: Vivek Farias
PhD Advisor: Deeksha Sinha

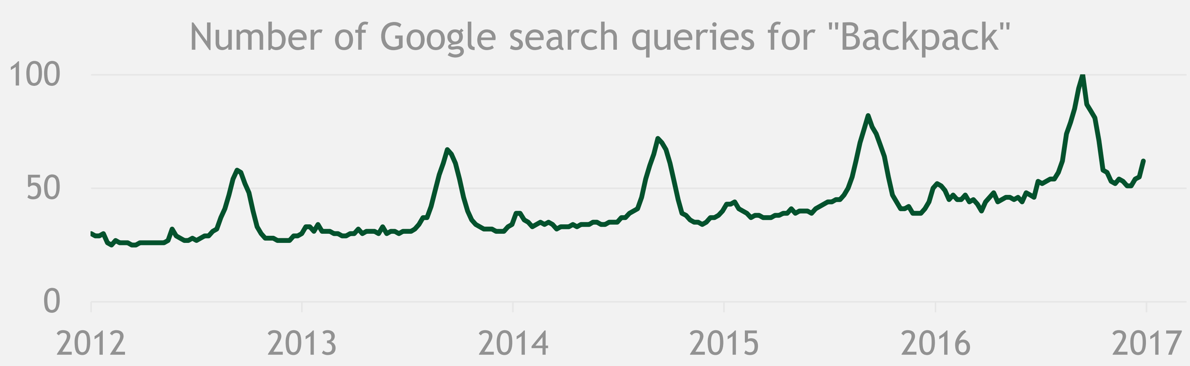
MIT MANAGEMENT SLOAN SCHOOL | BCG GAMMA

Kenza Sbai, Tim Valicenti — MIT Summer Consultants & Data Scientists

Jit Tan — BCG Project Leader

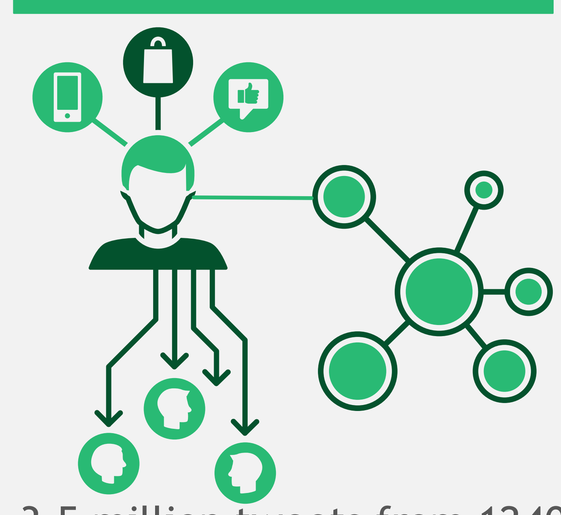
Julien Bohne, Sithan Kanna — BCG Consultants & Data Scientists

Google Trends



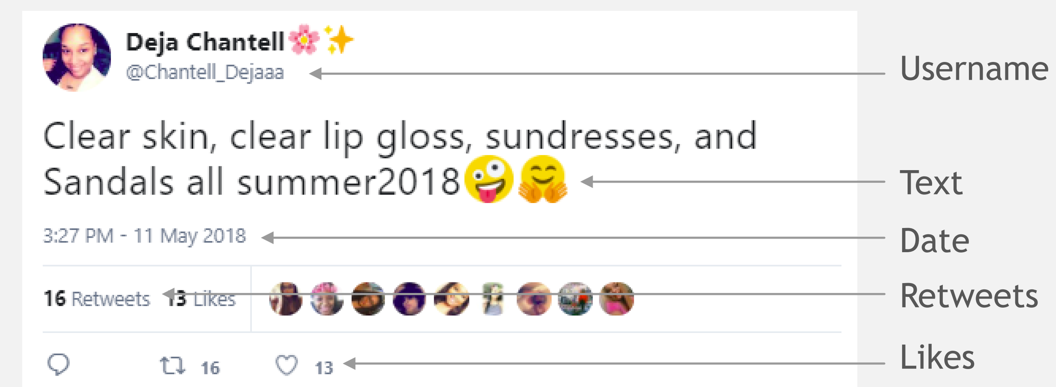
1. Used APIs to live connect to entire Google Search corpus – refined using the appropriate category filters
2. Used a small corpus of 450 terms to test our first forecasting and clustering methods
3. *Issue: data reflects demand of trends that already hit market – need earlier signals

Influencers



3.5 million tweets from 1240 different accounts

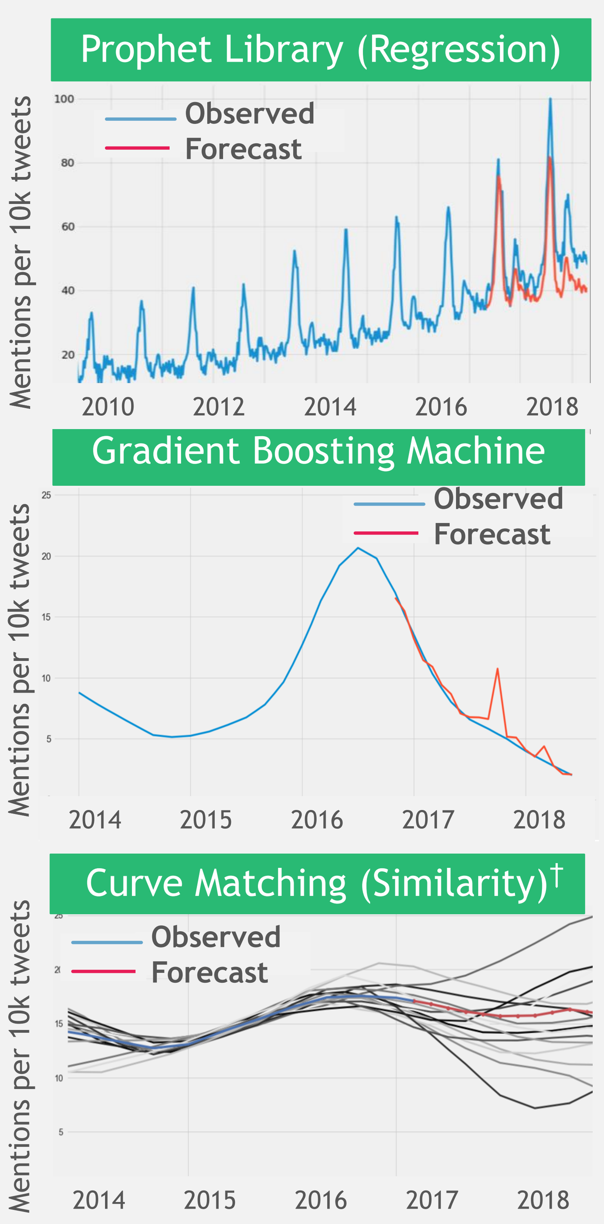
twitter



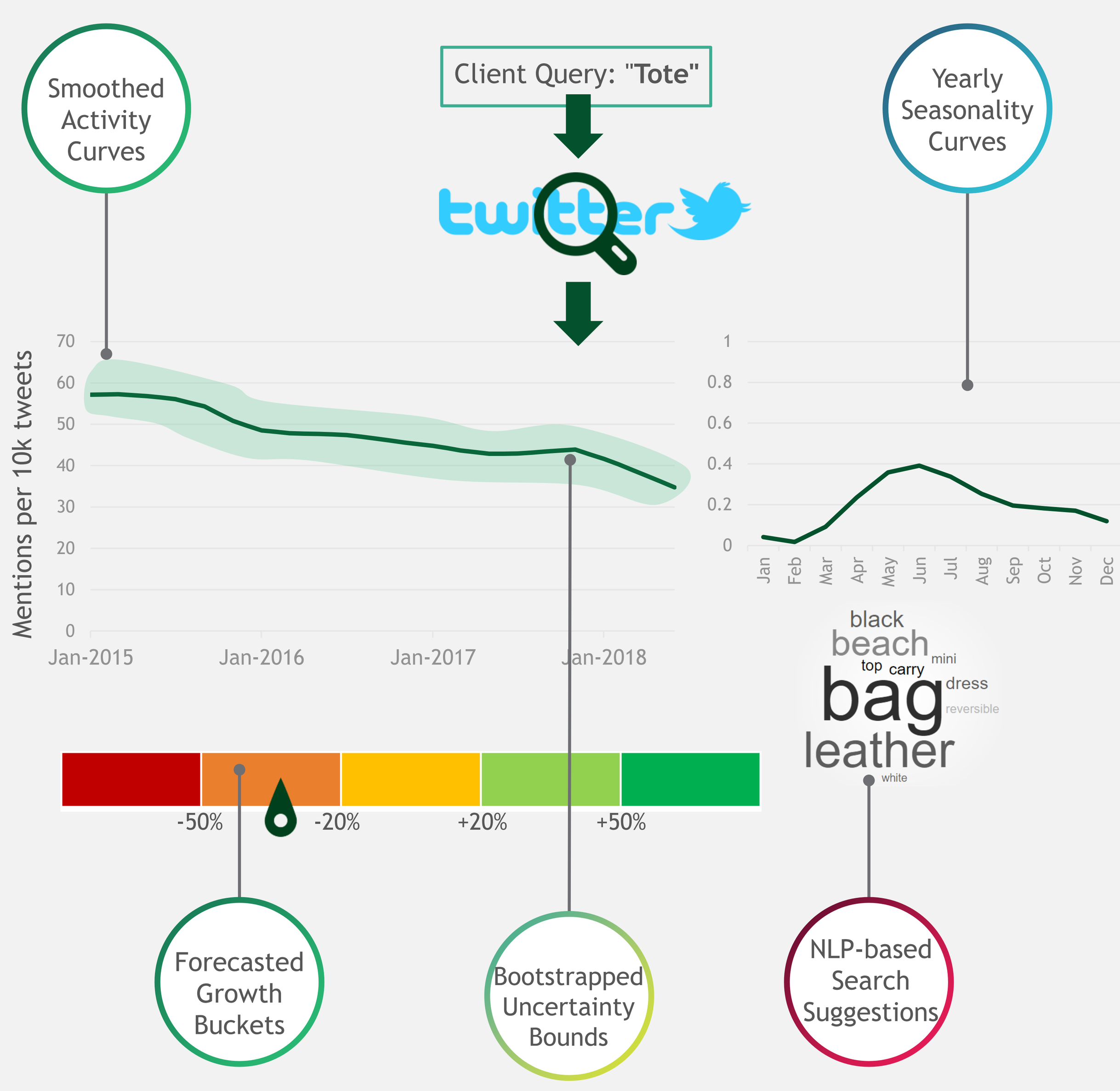
1. Using a seed of selected relevant influencers
2. Curating their mutual friends on Twitter to keep those who are focused on the same segment
3. Collect the users Tweets (text, date, likes…)

Social Media Data

Trend Forecasting

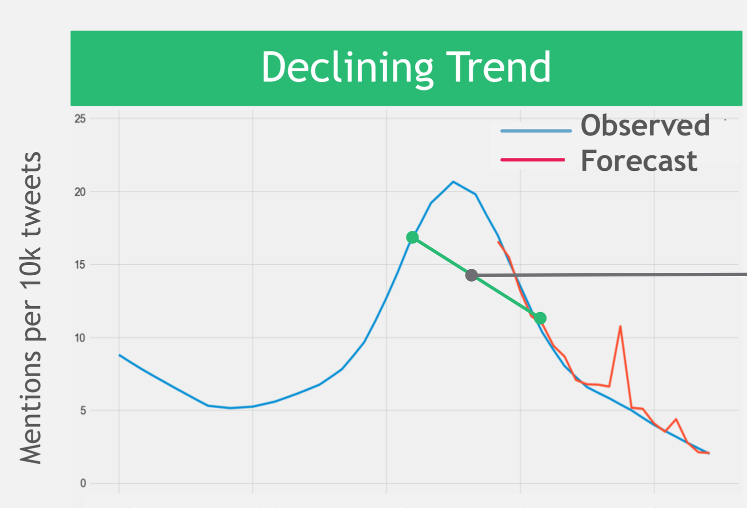


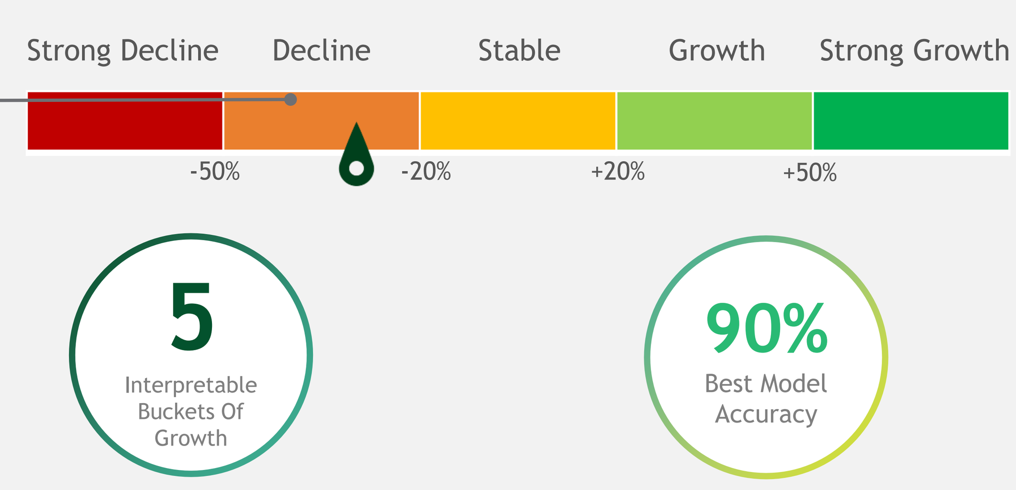
† Grey-scale curves show the 15 most similar segments

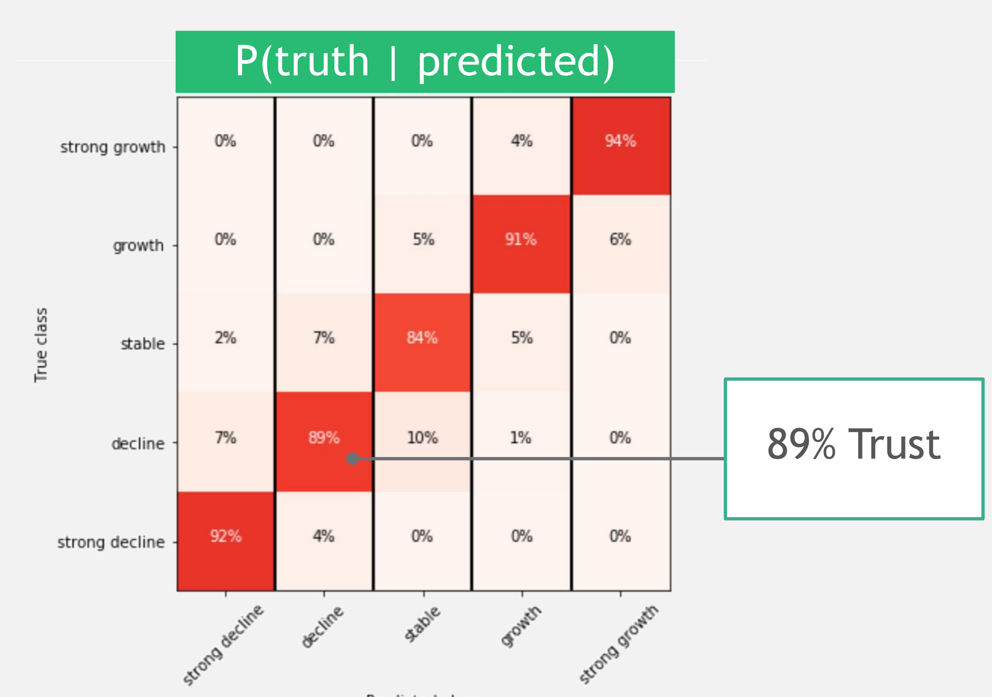


Trend Guidance

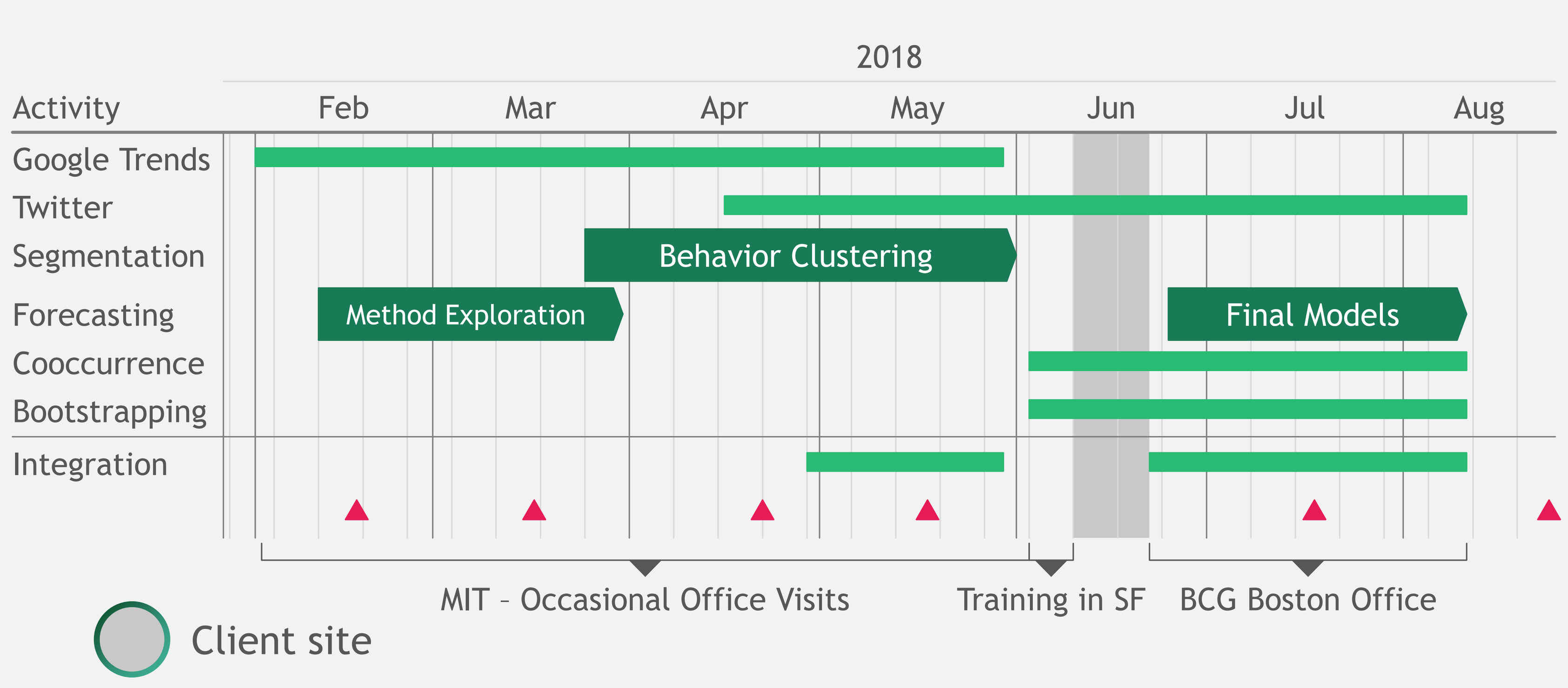
Growth Bucketing







Project Timeline



The Impact

BCG GAMMA

- Designed features for the overall solution provided to the client
- Integrated our work to the solution through an ETL pipeline

CLIENT

- Provided real-time insights on consumer trends
- Improved trends analysis for product developers and buyers' decision making

- User-test the solution with client users
- Continue aggregating Twitter data to get the most relevant analysis
- Evaluate the effectiveness of decisions made based on our solution for future development cycles

BMW GROUP

'What options would you like in your BMW i8?'

Option take-rate forecasting for BMW Group

Ana Lucia Perez Sanchez

Gijs Mulder

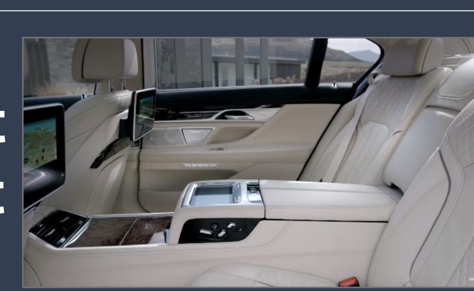
Model

BMW X6 XDRIVE 50I

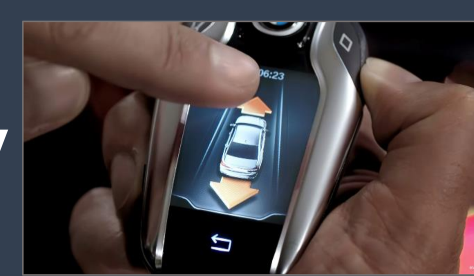
Options

+

Back seat entertainment



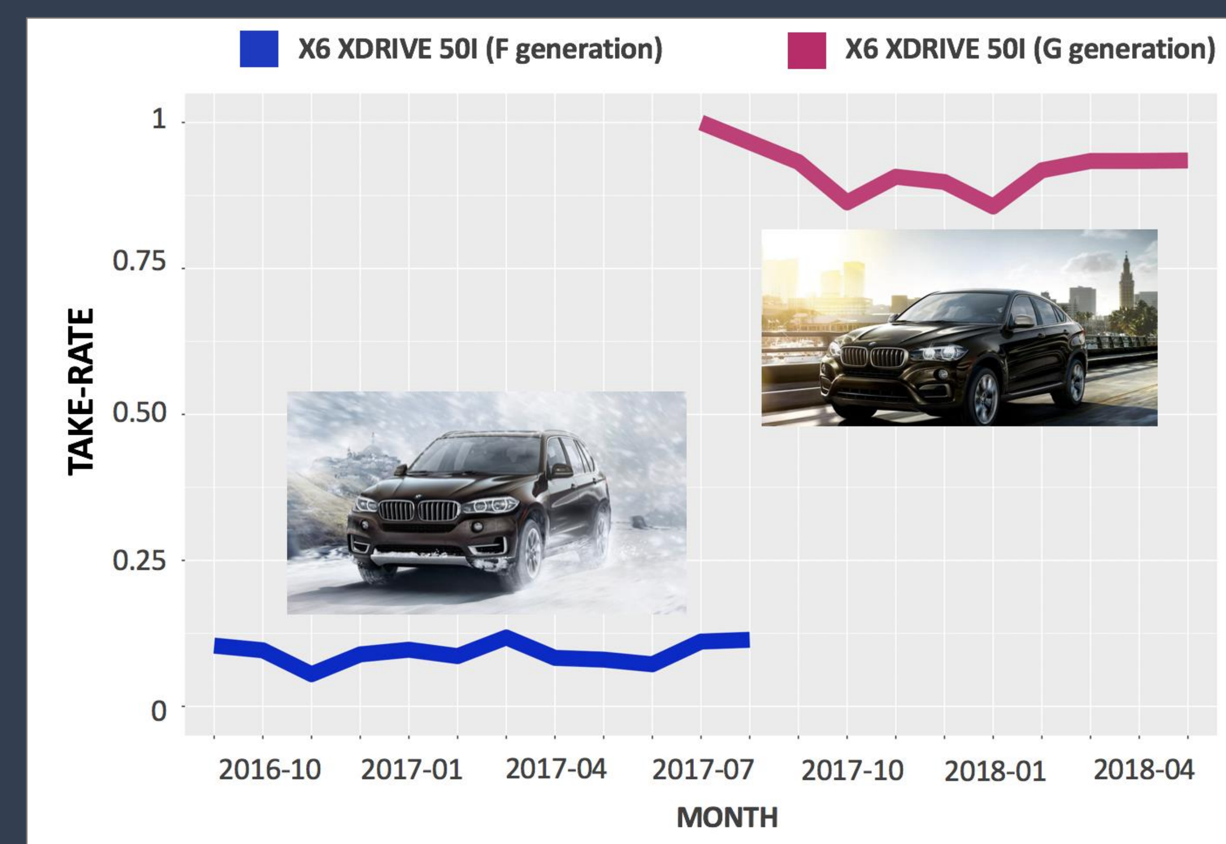
Display key

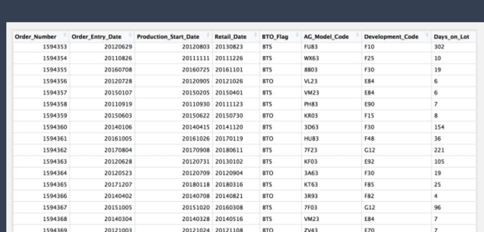


Bi-LED lights

Take-rates

(illustrative)





Data source: VDWH

- Individual sales data
- Row is a sale containing all features and options
- Data serves as basis for full project scope

1 Feb. 2018

1 Jun. 2018

17 Aug. 2018

US Build-to-stock optimization

First part of project

- Goal: Perform alternative to current BMW approach
- Define and test choice modeling approach
- Compare results to ML-approach from BMW

Option take-rate forecasting

Part two and content of this poster

- Goal: Establish Option Take-rate forecasting model
- Define approach, features and model from scratch
- Hand-over to BMW/BCG Gamma for implementation

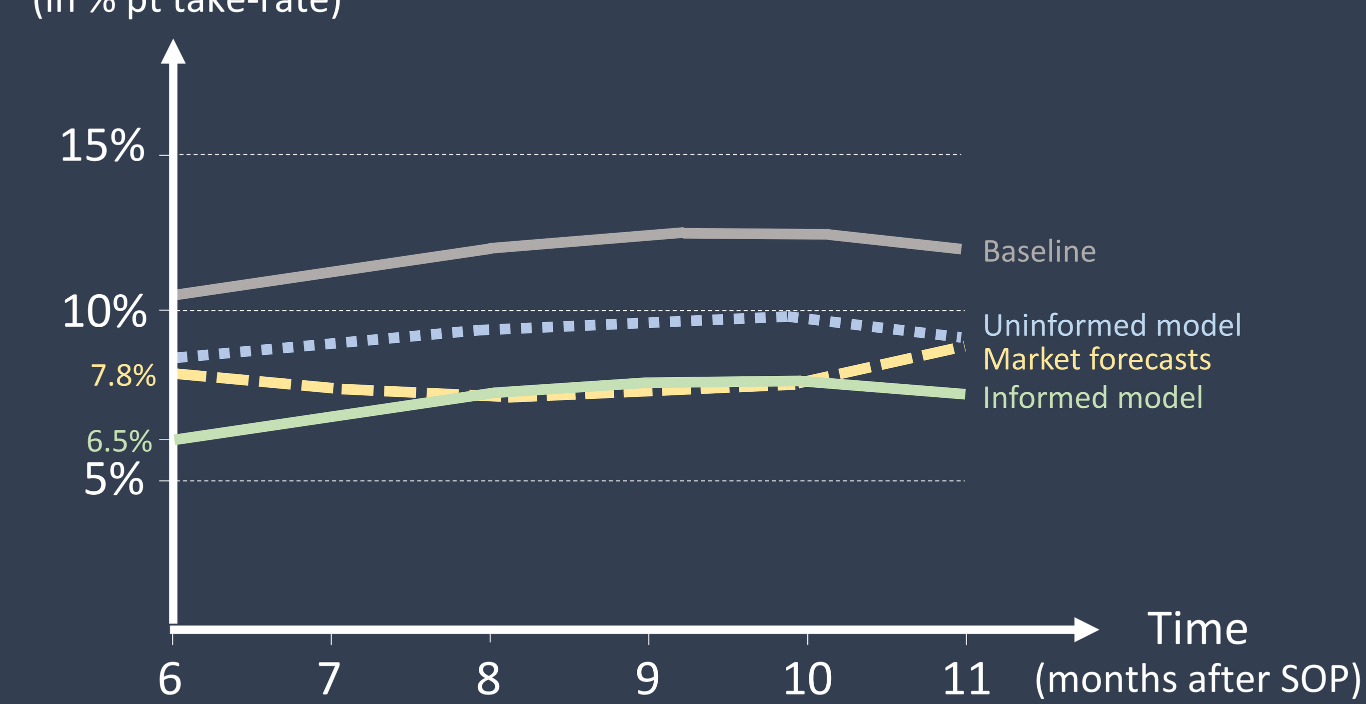
BCG GAMMA

SOP + 6 months

MAE observed (in % pt take-rate)

		MIT Model (Informed by markets)	MIT Model (Uninformed)	Baseline	BMW market forecasts
1 Existing models	Overall	NA	4.1% ↓	6.9%	NA
2 New Models	Overall	6.5% ↓	8.5% ↑	10.5%	7.8%
	G30 540I	5.2% ↓	7.0% ↑	9.6%	6.4%
	G30 540I XDRIVE	6.0% ↓	7.4% ↓	11.8%	8.3%
	G30 M550I XDRIVE	10.9% ↑	12.4% ↑	12.2%	10.1%
	G01 X3 XDRIVE 30I	4.9% ↓	8.0% ↑	8.5%	6.8%

SOP + 6 - 11 months



Data (.csv)

Pipeline steered by main document (R)

Preparation

- Raw data (VDWH or BV-15) → **Initial data frame.R**
  - Produce monthly option take-rates per unique ID
- Monthly take-rates per ID → **Smoothing.R**
  - Perform LOESS on take-rates across full timespan
- Regular and smoothed take-rates per ID → **Target.R**
  - Create prediction target (at prediction horizon set)
  - → **Re-smoothing.R**
    - Re-smooth training set to preserve right information
- Data including prediction and target variables

Time-series

- **Historic.R** •Set historic take-rates
- **Linear.R** •Fit linear trends
- **Quadratic.R** •Assess TR convexity
- **ARIMA.R** •Forecast ARIMA model
- **BusinessInput.R** •Include business forecasts
- **Generation.R** •Highlight generation transitions
- Multiple data sets including time-related variables

Feature engineering

- **Option price.R** •Show price of the option
- **ModelsOptions.R** •Indicate model series & option types
- **Similarity.R** •Show take-rates in similar models
- **MacroEconomics.R** •Input Macro-economic variables
- **Joining.R**
  - Joins the predictors into one data frame
- One data set including all prediction variables

Machine Learning

- **Grid Search.R**
  - Optimize hyperparameters using cross-validation
  - → **XGB.R**
    - XGBoost on input and prediction data
- Values & importances of variables in prediction

Optimization

- **Rule-based Optimization.R**
  - Avoids rule violation: rules as constraints
  - Outcome optimally close to ML output
- Test set RMSE and MAE and predicted future take-rates

Dr. Steffen Illig
Prof. Nikos Trichakis
Andreea Georgescu
Jonathan Amar

BMW Group
FG-24
ITZ - 6 Bremer Straße
München, Germany

[m]clusters

AUDIENCES FIRST

groupm

Copenhagen, Denmark

GroupM Mentor: Kristjan Brødreskift

Capstone Project By

Will Fein and Gerard Woytash

January - August, 2018

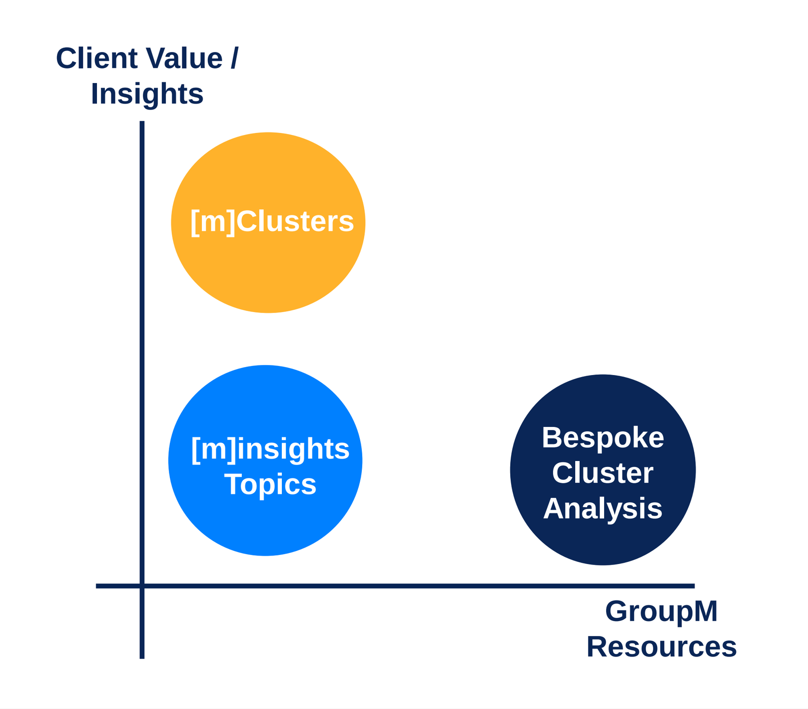
MIT Advisor: Karen Zheng

Who is GroupM?

GroupM is a leading global Media Agency. Advertising agencies make ads, but Media Agencies *place* them, and as online advertising and personalization increasingly dominate the marketplace, ad placement becomes more and more important. GroupM is focused on showing the right ads to the right people at the right time, and to this end has become a leader in data-driven solutions.

What are [m]Clusters?

[m]Clusters are segments of the population that are defined by particular online behaviors. GroupM clients can score their website's visitors against these segments, and can also target these segments for future advertising.



GroupM's Proprietary Data

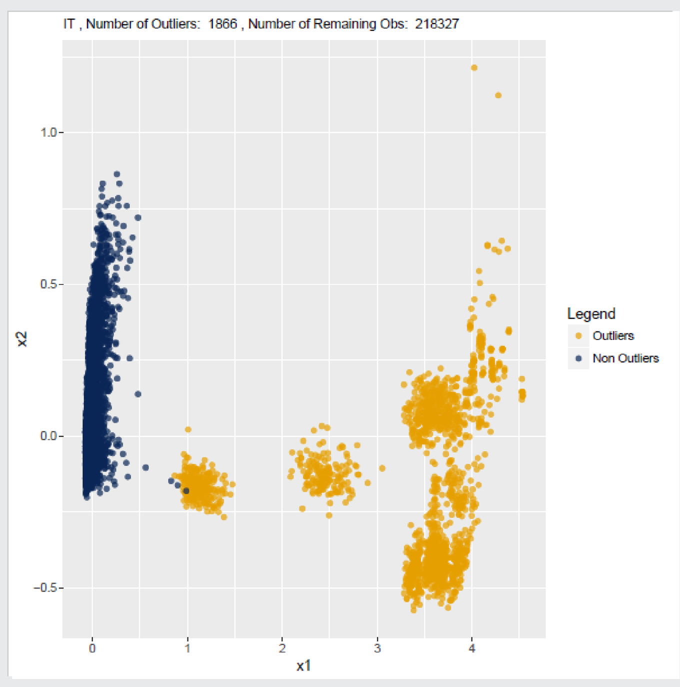
GroupM gave us access to their extraordinary and proprietary user interest data, among the most comprehensive in the world. As GroupM participates in display ad auctions, they record site visits for nearly all online users and nearly all webpages. These webpages are classified by a semantic engine, and then the counts of visits are converted to binary tables specifying whether a given user is "interested in" a given webpage type. It was these behavior datasets – and no additional demographic data – that we used to make our behavior-based segmentation.

visitor_id	Accessories	Adult_Education	Adventure_Travel	Advertising	Africa	Agriculture
ID_103[redacted]	0	1	0	0	0	0
ID_103[redacted]	1	0	0	0	0	0
ID_103[redacted]	1	1	0	1	0	0

PROCESS

Automated Data Cleaning

Our datasets had *bad users*, which we can either attribute to online bots or to failures of the semantic engine. Including these *bad users* hurt our clustering results. To make our entire process replicable, however, we couldn't leave behind any steps that centered around our ability to find and judge outliers. Our automated data cleaning notebook uses drastic dimensionality reduction with MCA and DBScan clustering to automatically detect outliers.



Dimensionality Reduction

High-dimensional data and binary data are both poorly suited to clustering analysis. The first suffers from the curse of dimensionality, and the latter makes distance calculations difficult. So, we used PCA to compress the user interest data and convert binary interests into continuous features. The curse of dimensionality – the notion that distance measurements converge in high dimensions – is not just a theoretical problem. Its business consequence is too many users placed in a 'leftover cluster.'

Ask Me About…

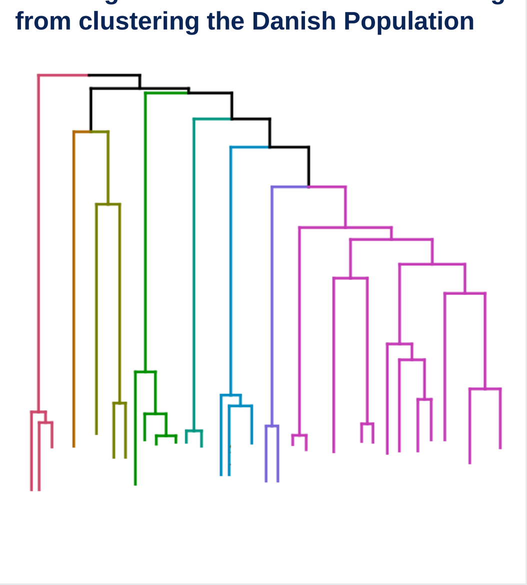
Things we'd love to chat about that didn't make this poster:

- Statistics for evaluating clustering algorithms
- Synthesizing user interests datasets from recoverable truth
- Dimensionality reduction with binary data
- Finding pre-determined clusters with supervised learning
- Ad buying and levels of personalization
- GDPR and demographic user data
- Copenhagen, Denmark or the Nordic Region!

Clustering

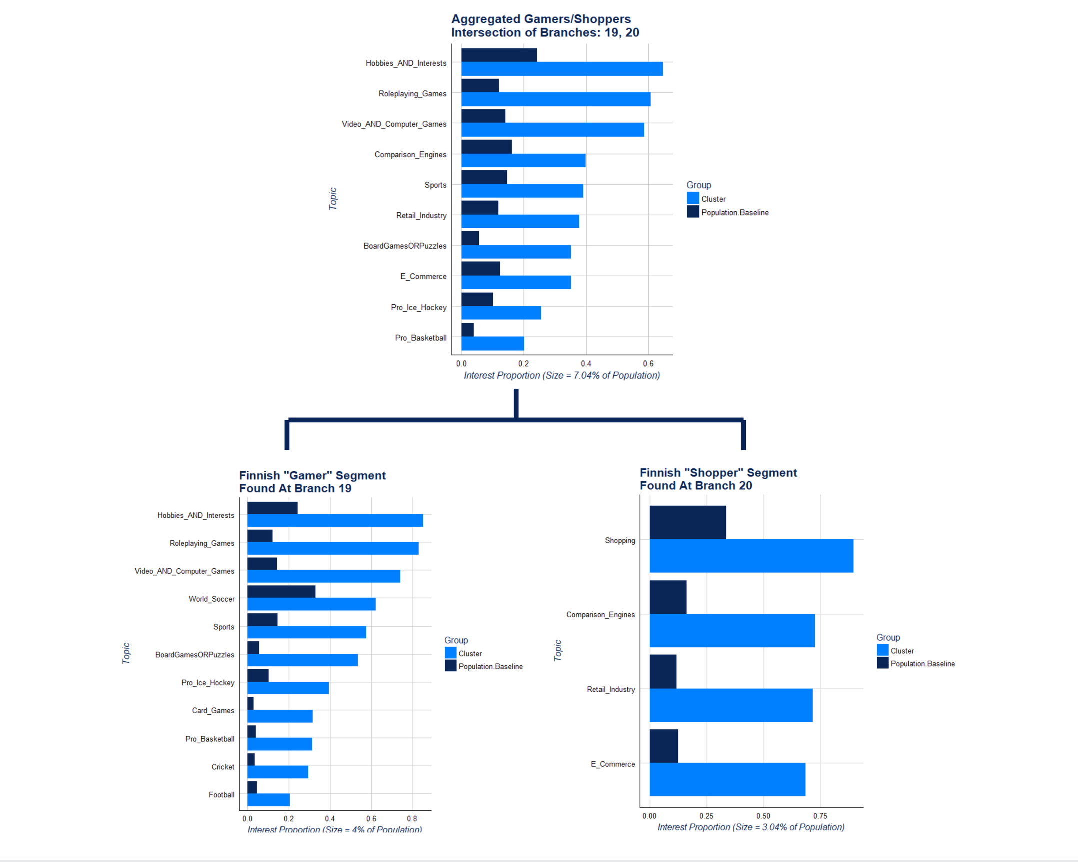
The next was to perform agglomerative hierarchical clustering with linkage determined by Ward's Method. We kept the first 30 branches of these trees.

Dendrogram with 30 leaf nodes resulting from clustering the Danish Population



Tree Pruning

"Choosing K" is always a difficult step in clustering, and we included business experts for this step of the process. In this business context, it's a question of aggregation. Which splits of the tree separated distinguishable audiences, and which split a single audience into two? The importance of this question centers on our ability to sell the segments to GroupM client managers, and on their ability to sell the segments to their clients. So, we asked local experts in each country to help us choose the level of aggregation for their market.



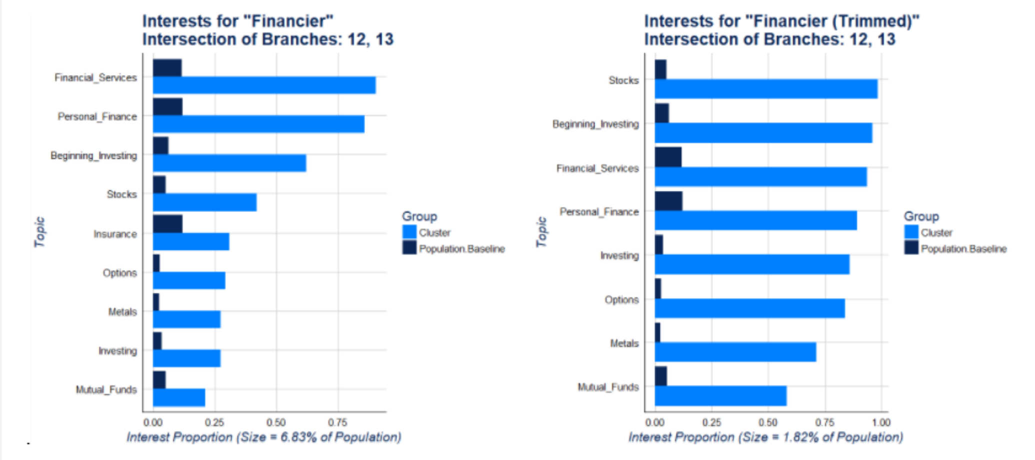
INTEGRATION & RESULTS

Final Segments

Audience Segment	Denmark	Sweden	Norway	Finland
The Attentive Parent	✓		✓	
The Car Aficionado	✓	✓	✓	✓
The Car Buyer	✓	✓		✓
The Do-It-Yourselfer		✓	✓	✓
The Driven Professional			✓	✓
The Engaged Citizen	✓	✓	✓	✓
The Fashionista	✓	✓		
The Financier	✓	✓	✓	✓
The Foodie	✓	✓	✓	✓
The Gamer	✓	✓	✓	✓
The Health Enthusiast		✓	✓	
The Interior Decorator		✓	✓	✓
The Mainstream Media Consumer	✓	✓		✓
The New Parent	✓	✓		
The News Junkie		✓	✓	✓
The Office Computer	✓	✓	✓	✓
The Online Shopper	✓			✓
The Real Estate Buyer	✓	✓		✓
The Sports Fan	✓	✓	✓	✓
The Student	✓	✓	✓	✓
The Stylish Parent	✓	✓	✓	✓
The Traveler	✓		✓	✓

Trimming Clusters

When it came time to upload our "segments," we had to define each segment in the GroupM platform by passing in a list of user ids. We chose these user ids carefully, building algorithms that essentially trim the clusters and amplify the signals.



Lookalike Results

One metric of important to GroupM is the ability of their bidding and insights engine to identify the right users from the population to match a segment. They evaluate this with a statistic called GRP, which measures the efficiency of the lookalike models. It can be thought of as similar to recall, except that smaller GRP numbers mean that the models are more successful. MIT segments outperformed nearly all benchmark models.

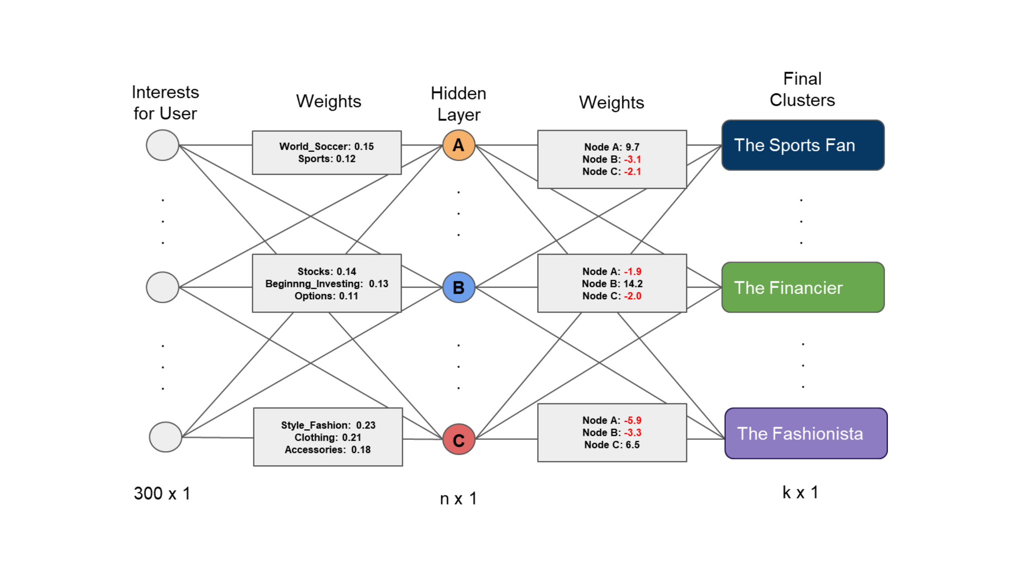
Buying Trial Results

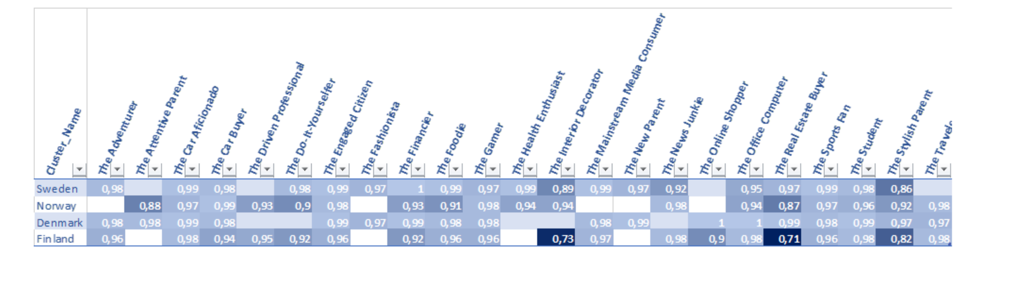
We activated our segmentation for three different clients, who each selected certain segments to target. These selections were based on segments "over-indexed" for their current customers, and the results shown below demonstrate the monetary value of [m]Clusters.

Client Industry	Buying Strategy	Click Through Rate	Conversion Rate	Cost Per Conversion	Impressions
Automotive	Broad Strategy	1%	NA	1	>345,000
	[m]Cluster Activation	1.25%	NA	0.94	>16,000
Telecom	Broad Strategy	1%	1%	1	>398,000
	[m]Cluster Activation	1.36%	2.07%	0.77	>202,000
Amusement Park	Broad Strategy	1%	0.01%	1	>24,000
	[m]Cluster Activation	1.01%	9.84%	0.09	>2,400

Neural Networks for Cluster Scoring and latent Sparse Dimensions

Neural networks allowed us to find sparse latent dimensions that could explain the hierarchical clustering tree in the original feature space and define the segments with few, shared dimensions. They also allowed to score our segments on how easy they were to distinguish from the population and from other segments. Our results were almost uniformly excellent (precision and recall > 0.9), and where they were not they were informative. In Finland, it seems our Interior Decorator and Real Estate Buyer segments may contain many similar individuals.



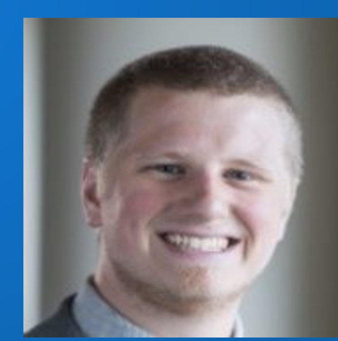


IBM WATSON

Cambridge, USA

Intent Classification from Unlabeled Dataset

Stephen Albro
salbro@mit.edu



Chuanquan Shu
c.shu@mit.edu

IBM Supervisor: **Robert Yates**; MIT Faculty Advisor: **Patrick Jaillet;** PhD Advisors: **Konstantina Mellou**, **Chong Yang Goh**

Business Problem

Businesses can customize Watson Assistant to recognize common requests (intents) that their customers frequently make. IBM invests a lot of energy into helping its clients train chatbots that are specific to their businesses. Our work falls into this effort.

In terms of machine learning, we want to empower IBM business users to train a *classifier* to recognize each of their customer intents. Text classification traditionally requires an extensive labeled data set of examples, but this places a burden upon IBM's business users. Hand-labeling requires hundreds of hours of manual labor and can only be done by a subject matter expert.

Our Solution

Our capstone aims to use machine learning to most efficiently tap into the subject matter expertise of an IBM business user, such that a quality custom classifier can be produced from an unlabeled dataset. We develop a browser-based process, in which the machine honors the time constraints of the user. It does this by surfacing the most relevant words and phrases to the user and then adapting to the user's response. The human and machine work together until the user is satisfied.

Data: Customer Utterances

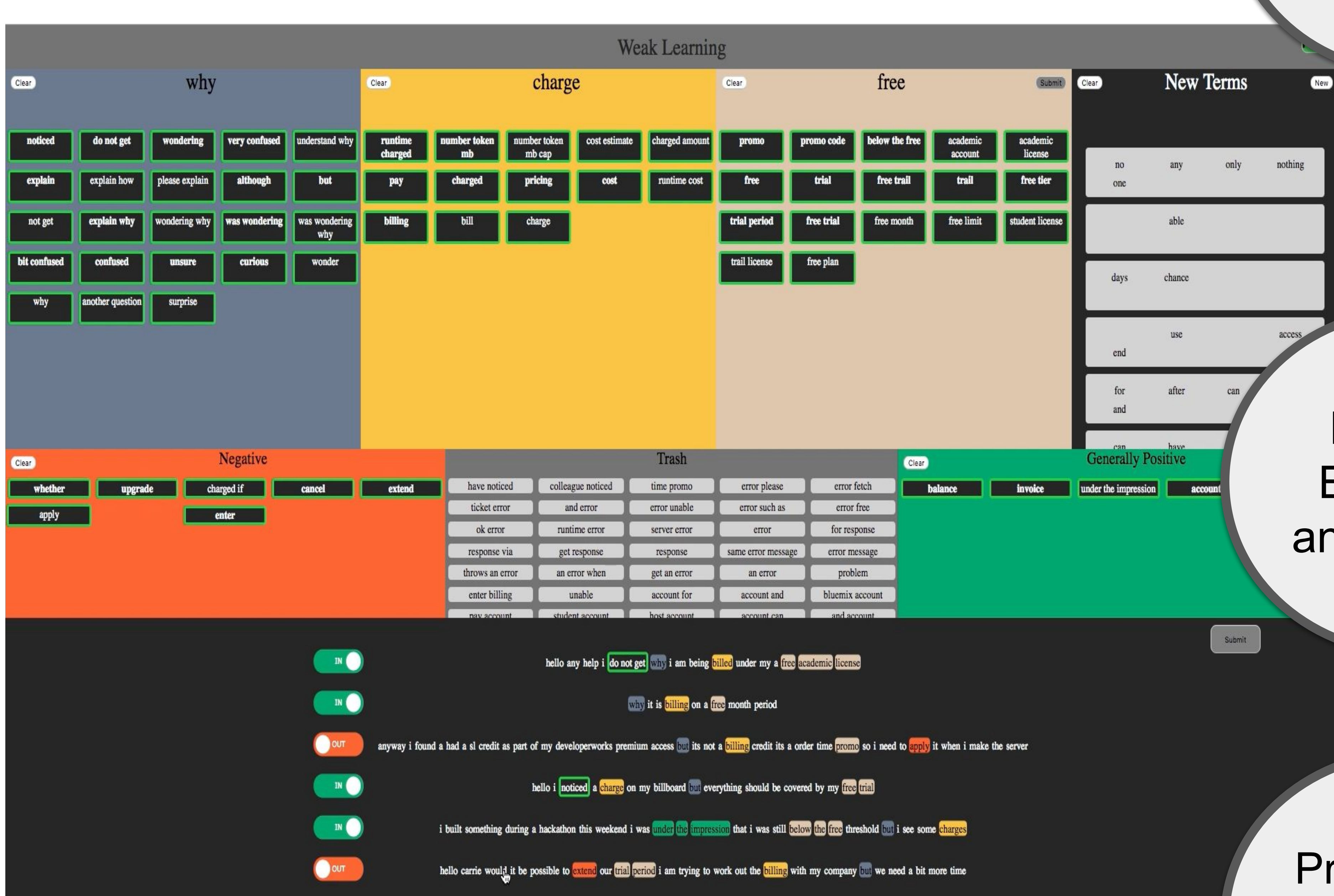
When training Watson Assistant, business users provide data sets of customer chat logs. IBM provided its own, containing 55,000 customer utterances with nine commonly-occuring intents.

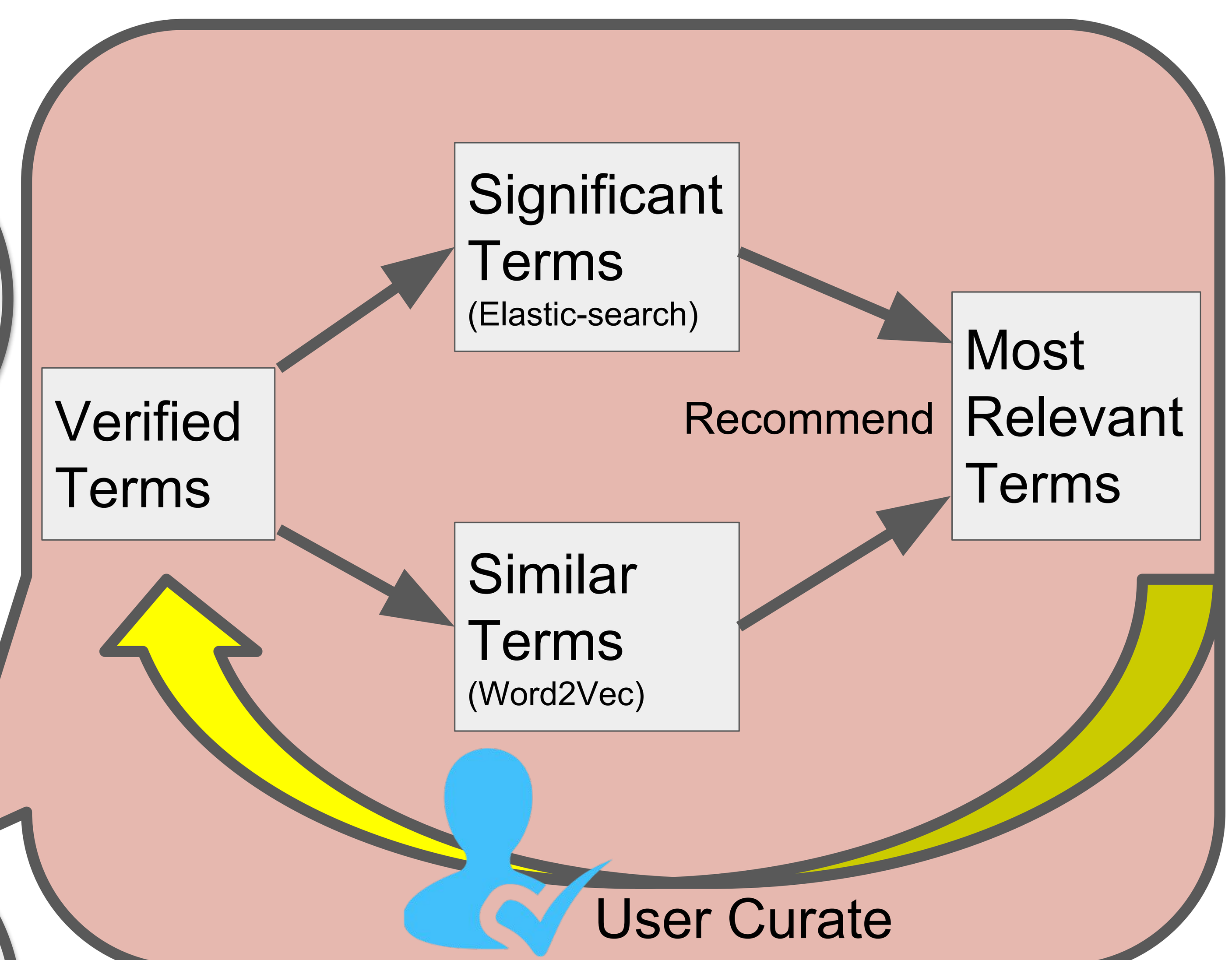
Matched with IBM Watson and received the project (2018 Feb) → Offsite: Data Exploration, Preliminary Analyses and Logic Development (2018 Feb - May) → Onsite: Logic & Methods Developments (2018 May - July) → Onsite: Browser App Development (2018 July - Aug) → Internal & External Presentations (2018 Aug)

Intent Understanding Tool

Intent:
"*Why am I seeing a charge? I have a free account.*"

User Specifies Intent

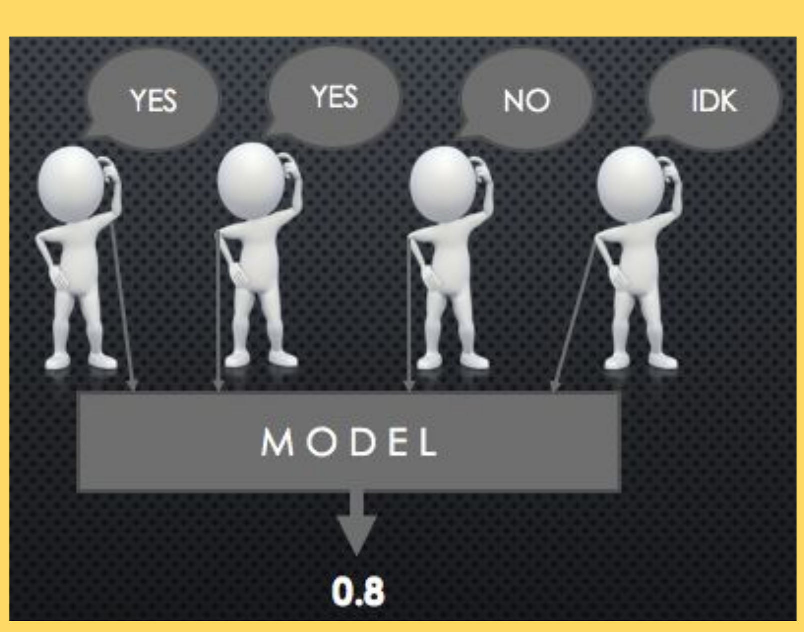




User Defines, Explores, and Refines Intent

Probabilistic Labels Generated

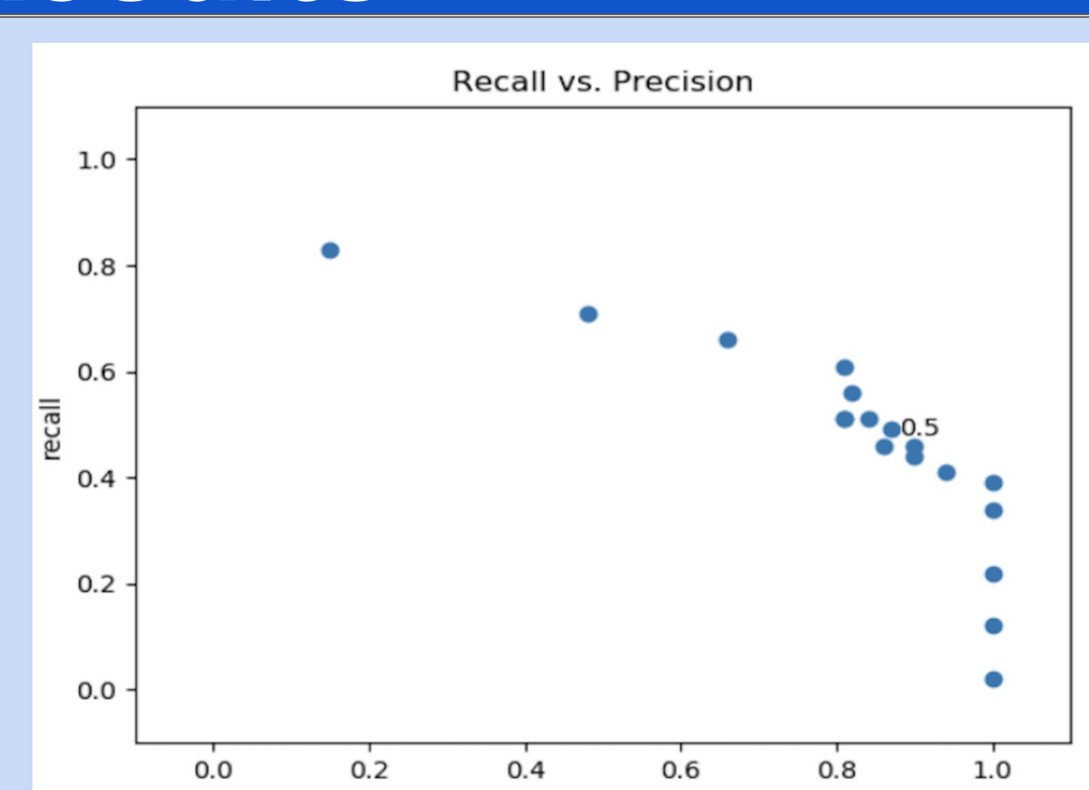
- After the words and phrases are grouped, the machine uses heuristics to "vote" on whether the intent is present.
- The votes of each utterance are synthesized into a single *probability*, which acts as the training label.



wisdom of the crowd

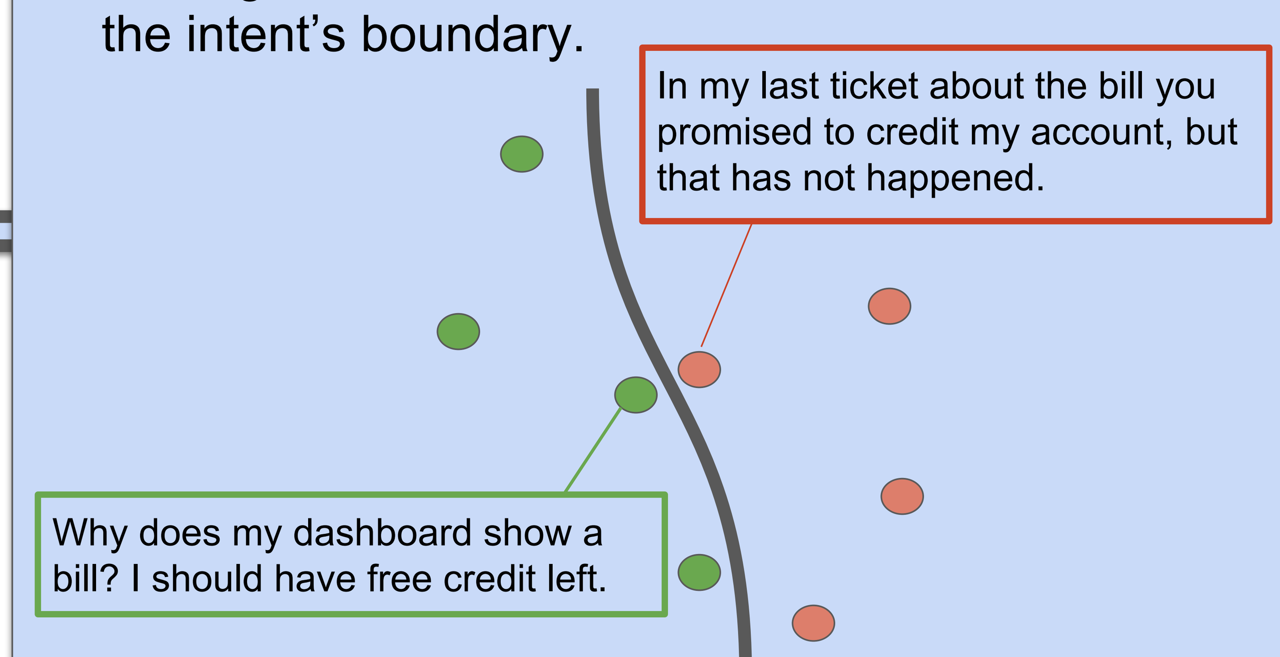
Text-Classifier Trained on Probabilistic Labels

Results



Results for Why-Free-Charge Intent (Above)

We transformed dozens of hours of hand-labeling into a 20-minute, low-cognitive-load experience leading to labels that carve out the user's idea of the intent's boundary.



Capstone: Generating Product Recommendations for small businesses at scale

Subhashree Rengarajan

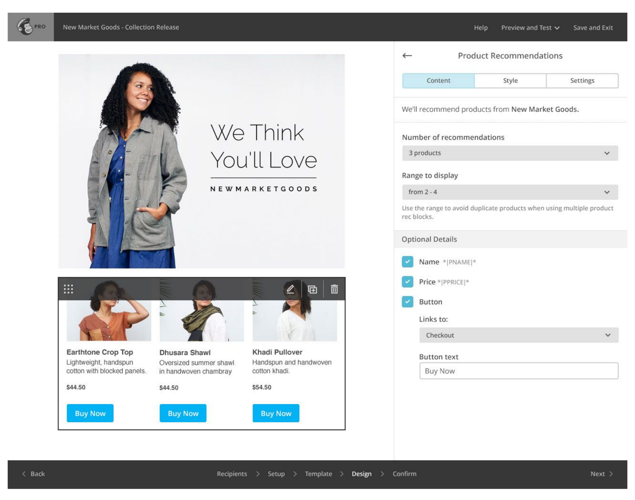
Yingtian Yang

Advisors and Mentors: Neel Shivdasani, Rahul Mazumder, Hussein Hazimeh

What is MailChimp?

Mailchimp is the world's leading marketing automation platform for small businesses. To this end, the platform offers services including marketing automation, landing pages, email templates and product recommendations (affectionately known as P-REX).

MailChimp's goals are to publish the right content to the right person at the right place at the right time.



What are Personalized Product Recommendations?

Using the purchase history of each customer to make smart, data-driven predictions about what they'll want to buy in the future.

Our 1st few weeks were reviewing customer feedback about the existing system, understanding pain points, and seeing if there were ways we could improve the existing P-REX system.

Central Business Question: Can we improve the relevance of P-REX for consumers who are the recipients of Product Recommendations from MailChimp customers?

Datasets

Tens of thousands of customers use product recommendations each month.

Raw Data:
- Sample of ~1,000 stores
- Historical transactions for 3 years
- Product details, including text descriptions

Store ID	Customer ID	Product ID	Ordered At
59165197	1358525	1795	2017-02-21
59165197	1274065	1802	2017-02-01
56892273	NaN	115	2017-08-20
56892273	1432704	112	2017-09-07

Product ID	Title	Description
7523	Raven Disco Jumper	With a **high neck**, **low back**, this **dress** was designed for you in all the right places.

Cleaning and Processing:
- Removed NA's, aggregated sales for the same customer, and same products
- Transformed the datasets into user * product matrices

Purchases	Hats	Socks	Hoodies
Ben		✅	
Dan			✅
Neel		✅	✅
Mitch			✅
Jasmine	✅		

Phase 1: *Review customer feedback*

Phase 2: *Data Processing and Metrics*

Phase 3: *Choosing the right Algorithm:* ***Soft-Impute***

Phase 4: *Results and Recommendations*

Recommender Systems: Solving the problem of missing data

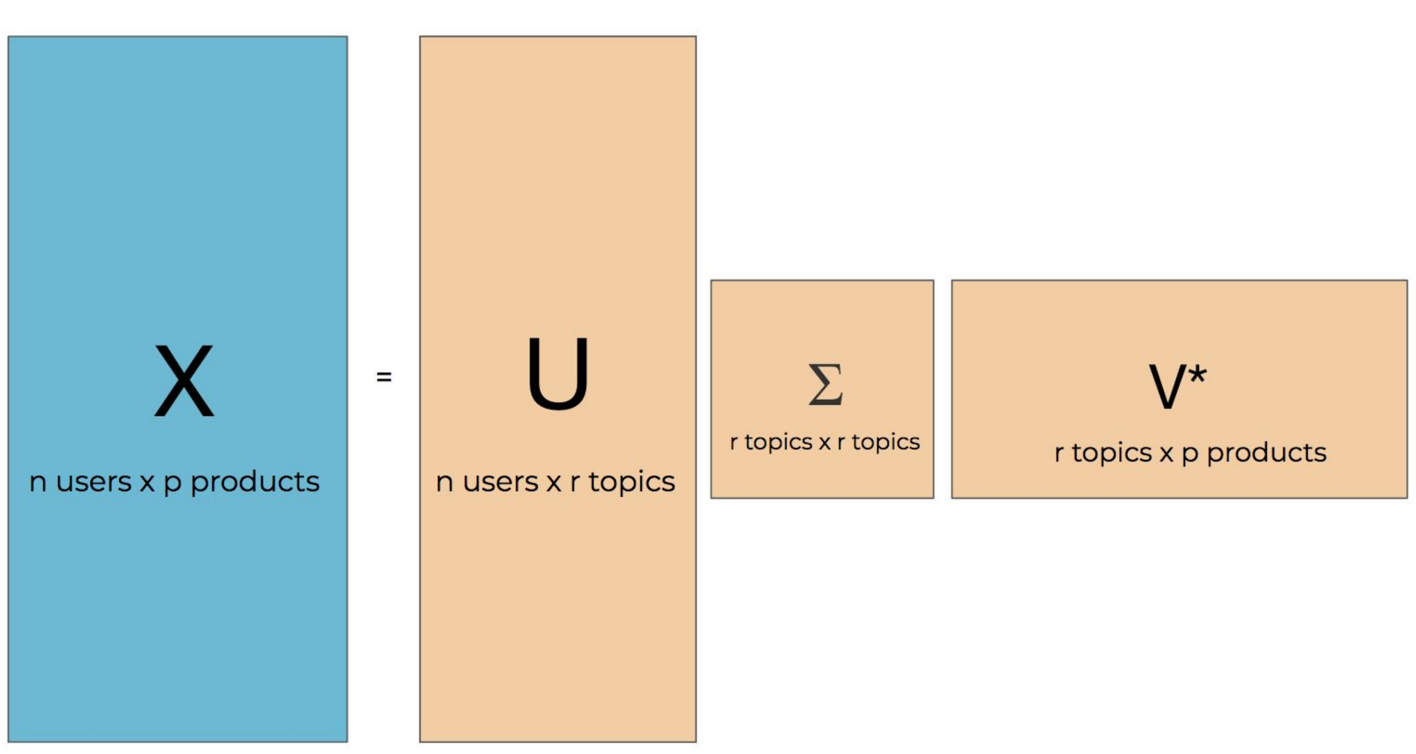
After exploring multiple machine learning algorithms used in recommender systems (BM25, Weighted Alternating Least Squares), we settled on Soft-Impute as it requires the fewest parameters to tune.

Soft-Impute: The idea is to impute the missing values where people have not bought anything with educated guesses while also minimizing the error on the observed values.

$$\underset{Z}{\text{minimize}} \sum_{\text{Observed}(i,j)} (X_{ij} - Z_{ij})^2 \quad \text{subject to } ||Z||_* \leq \tau$$

$$||Z||_* = \sum_j \lambda_j(Z)$$

The algorithm is based on singular value decomposition, the breakdown of a matrix into 3 submatrices, which reduces the dimensionality as well as providing some interpretability to the system.



The only tuning parameter: λ, as a penalization coefficient. Similar to the penalization parameter in LASSO, here λ is a penalty on the nuclear norm $||Z||^*$. Once we generate our approximation Z, we're able to make estimations on what people will like and dislike.

Purchases	Hats	Socks	Hoodies
Ben	💖	✅	👎
Dan	👎	💖	✅
Neel	👎	✅	✅
Mitch	👎	👎	✅
Jasmine	✅	💖	💖

Testing and Results:

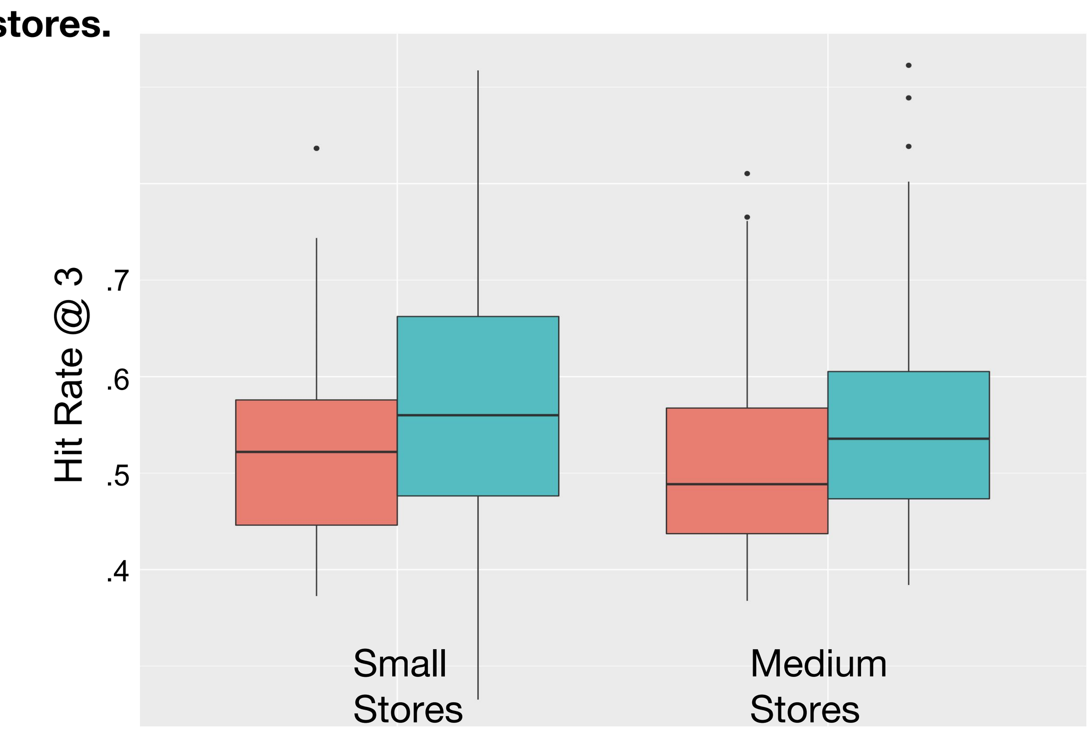
We tested cosine similarity, which is MailChimp's current method, and SoftImpute on small and medium stores*. The metrics we used to train the model and tune λ is NMSE:

$$Normalized\ Mean\ Square\ Error\ (NMSE) = \frac{mean(\sum Model\ Predictions - Observed\ Ratings)}{(Observed\ Ratings)^2}$$

With the optimal λ, we masked 20% of the purchase matrix and tested recommendations using Hit Rate @ 3: how many items the model can detect as being purchased i.e. the top 3 items likely to be purchased by the user.

We have run the Soft-Impute methodology over 74 small stores and 112 medium stores using a stratified sample,*

For both the small and medium stores, Soft-Impute outperforms the Cosine Similarity recommender system, but the difference is only statistically significant for small stores.



Recommendations:

Business Impact: Expansion of the P-REX feature will give MailChimp's customers a greater ability to grow their small business by using personalized e-commerce tailored to their consumers.

For MailChimp, we have observed that the most benefit would be applying Soft-Impute to the small-stores who are not already able to generate recommendations. We see a net benefit to expanding this feature to more small businesses who may not qualify for P-REX under the current schema.

Thanks for a wonderful summer in Atlanta, Georgia!

* Small stores contain 1-29 products (not including variants of size or color), medium stores contain 30 - 100 products

Routing Vehicles for the MBTA's RIDE

SPONSORED BY

Massachusetts Bay Transportation Authority

INTRODUCTION

- The RIDE is MBTA's transportation service for mobility-impaired people
- This service is mandated by the federal government as part of ADA guidelines
- It serves 55k people / year
  - 5000 - 6000 rides on a weekday, 2500 rides on a weekend
  - 20% are in a wheelchair

The RIDE's operational costs exceed **$100 million** annually

PROJECT GOALS

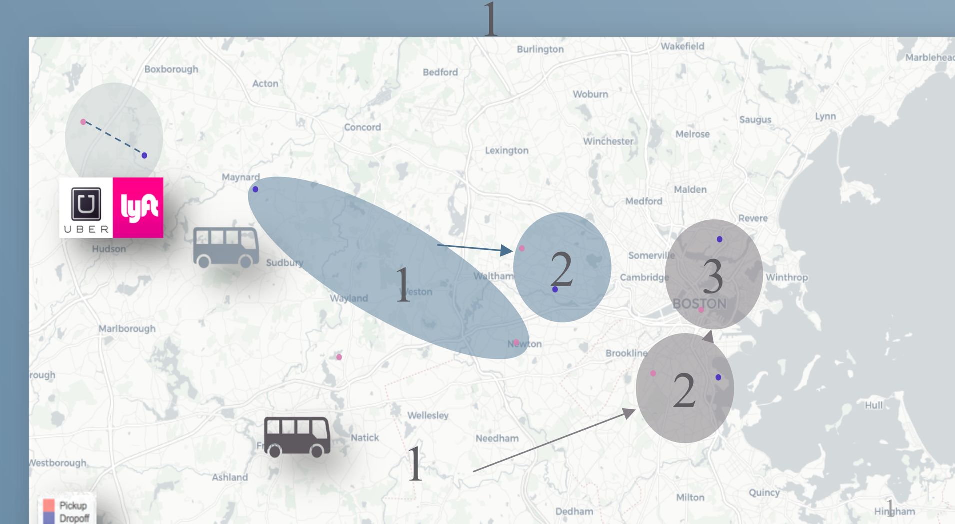
Assess their historical efficiency and the capabilities of their current software

Provide a systematic way to group similar rides together

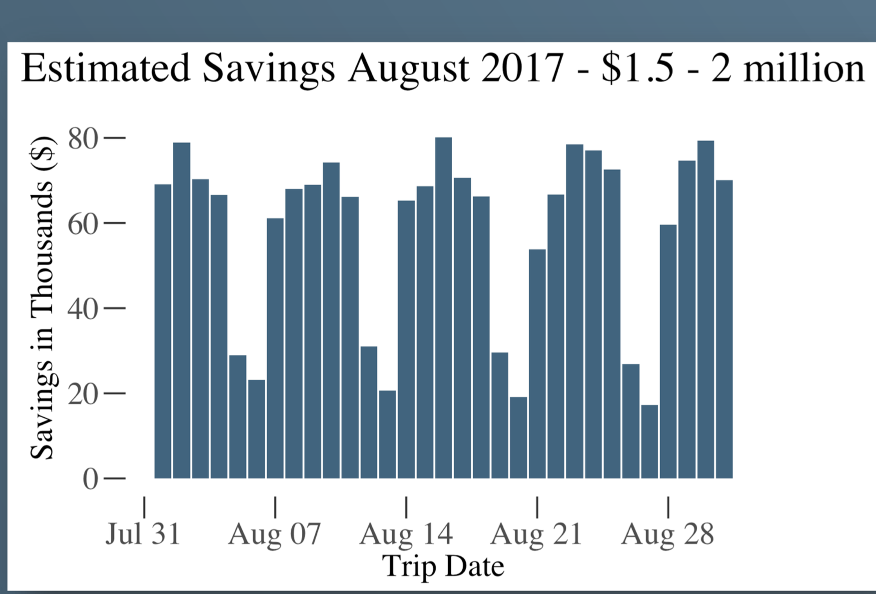
Provide an algorithmic way to assign trips to non-dedicated service providers

METHODOLGY

INPUT RIDE REQUESTS → FORM "MINI-CLUSTERS" → REMOVE UBER/LYFT RIDES → SCHEDULE AMONG CLUSTERS → OUTPUT DRIVER ROUTES

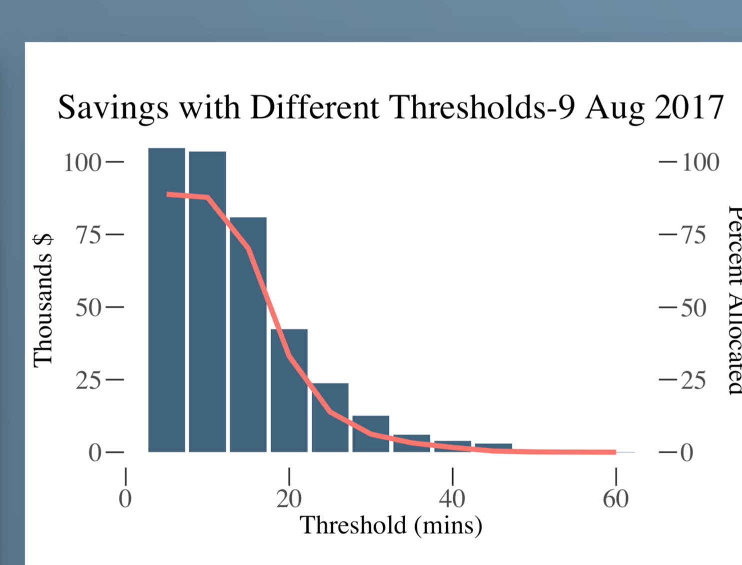


RESULTS

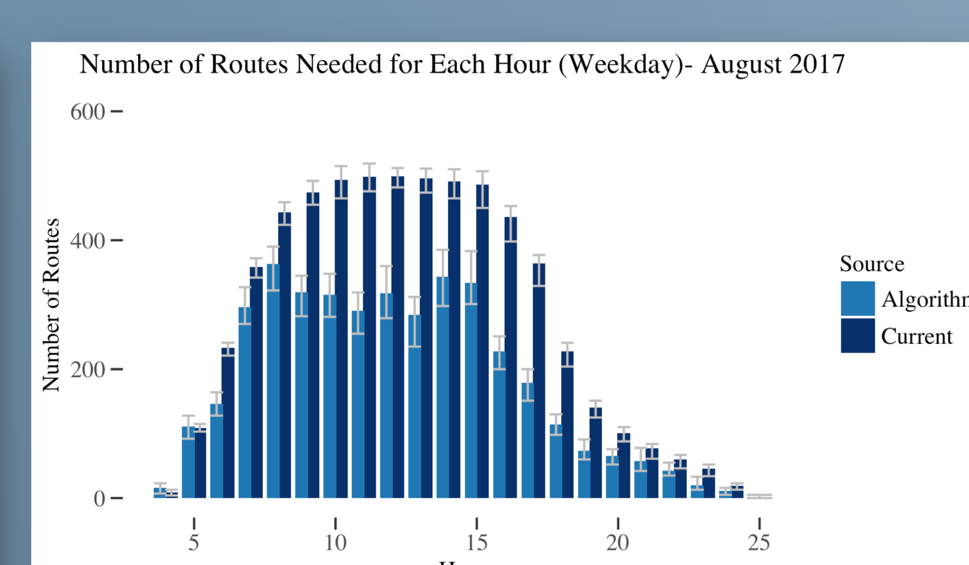


This figure shows the estimated daily total cost savings using our greedy algo rithm. Savings were lower on weekends as since there were fewer trips

Blue bars show results where our algorithm outperformed Adept, red bars the contrary. Generally, the difference between the two algorithms is not significant



The savings by allocating trips to TNCs are shown in bars, and the percentage of allocated trips is shown in red.



There is a large gap between the number of required cars and the number of available cars between 9 AM and 6 PM. Potential issues occur early in the morning at 4 and 5 AM, as well as after 9 PM

NEXT STEPS

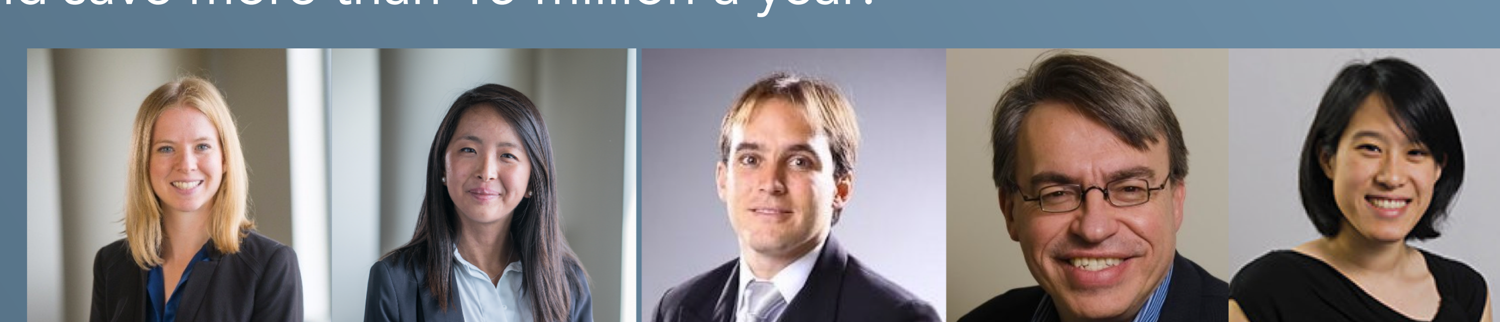
- Continue with legal steps to introduce Non-Dedicated Service Provider allocation. Begin at a small scale to work out technology and user satisfaction and then expand.
- Reduce the number of routes
- Integrate our algorithm in daily operations
- Investigate potential root cause of inefficient routing

CONCLUSION

The MBTA's RIDE service is a costly operation for the department, and the goal was to identify areas to reduce costs. There is significant savings to be had by allocating trips to non-dedicated service providers, at a higher cost savings than efficiently routing, so we strongly urge the MBTA to work towards this change as its first priority. Additionally, we showed that inefficient routing has led to excessive costs and if the MBTA was to improve this routing, they could save more than 15 million a year.

MIT MANAGEMENT BUSINESS ANALYTICS
OPERATIONS RESEARCH CENTER

Left-to-right: Sarah Eade, Céline Guo, Diogo Lousa (MBTA sponsor), Prof Dimitris Bertsimas (advisor), Julia Yan (mentor)



WHAT ARE LARGE ORGANISATIONS HUNGRY FOR?

MIT MBAn CAPSTONE (Sponsor: McKinsey & Company)
Benjamin Lim, Rita Yuan | Mentored by Carine Simon, Chris McCord

404 Wyman St
Waltham MA
Open Mon-Fri:
730am-530pm

About Us

We are the world's first data-science restaurant run by recovering consultants hailing from China, Germany, USA, and Singapore. Disclaimer: Our food does not contain any HiPPOs*.

**Highest Paid Person's Opinion*

Our Mission

To identify what is top-of-mind for large organizations using **topic modelling**, so as to lead knowledge acquisition efforts within McKinsey. Finding out what organizations care about helps us to **highlight knowledge gaps**. We also model relationships between different topics to **uncover cross-functional synergies** within the firm. To date, we have partnered with two Practices to derive insights using our tool.

The Ingredients

Our Recipe

1) Processing: Melt documents to boil off any uninformative words and confidential information.

2) Model Topics: Train and compare Latent Dirichlet Allocation, Biterm Topic Models and Correlated Topic Models.

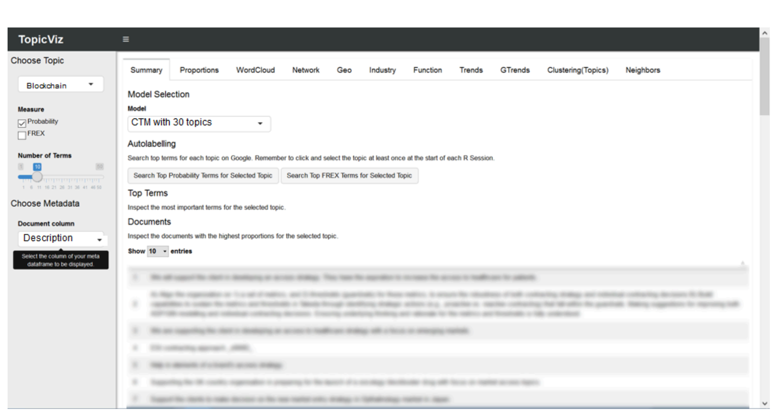
3) Label Topics: Apply auto-labelling algorithms to derive labels for topics and quantify their quality. Topics with low-quality auto-labels are manually labelled.

4) Enrich Topics: Add metadata (the function, industry, and geography of a document) to allow for tailored analyses and cross-functional comparisons.

5) Visualize Models: Build application for end-users to easily understand what each topic means, how documents are related, and explore how topics change across time and space.

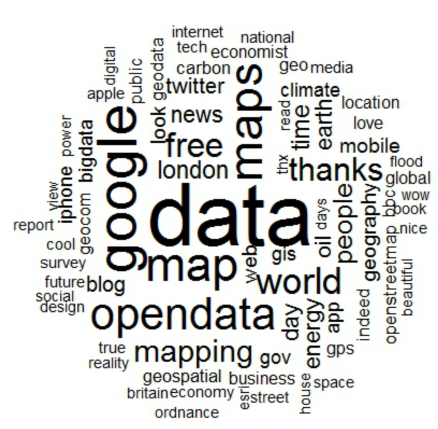
6) Derive Insights: Partner with specific Practices to build custom models and generate actionable insights.

Appetizers



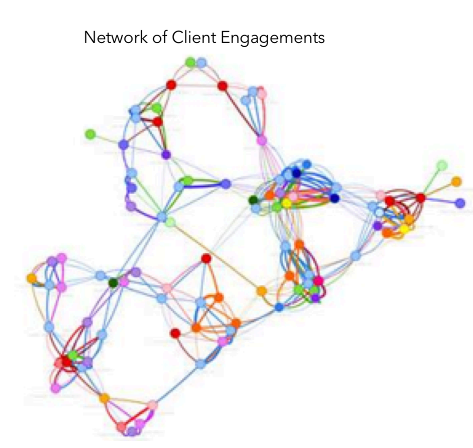
Document Exploration

We display the documents most representative of each topic.



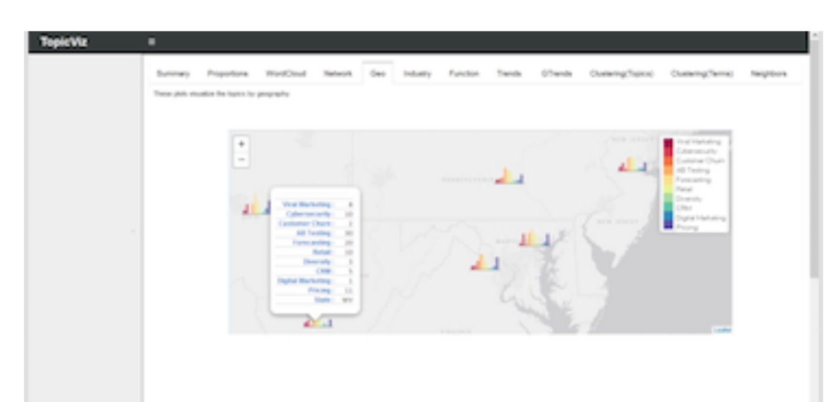
Word Cloud

Words most representative of each topic are shown in a word cloud.



Network Analysis

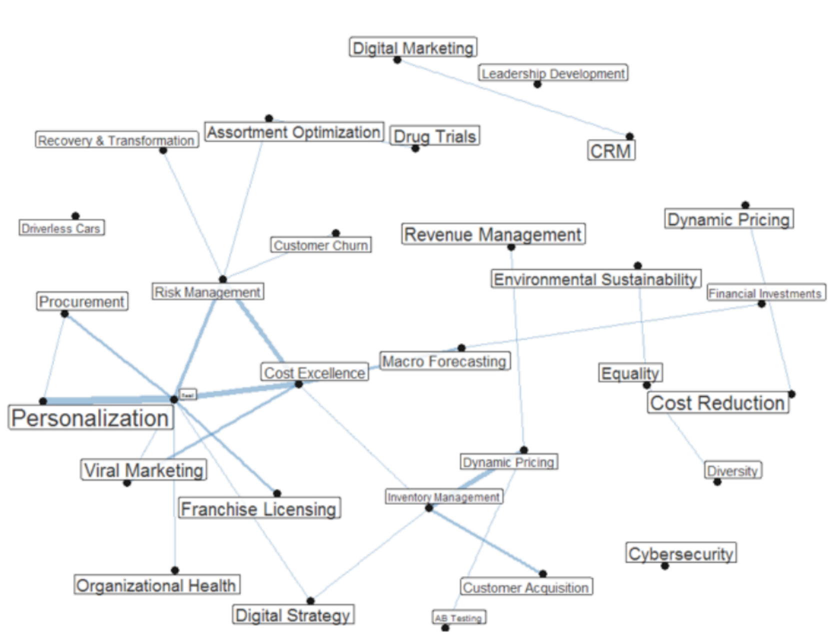
Each document is a node, and the edge widths represent similarities between documents.



Geospatial Analysis

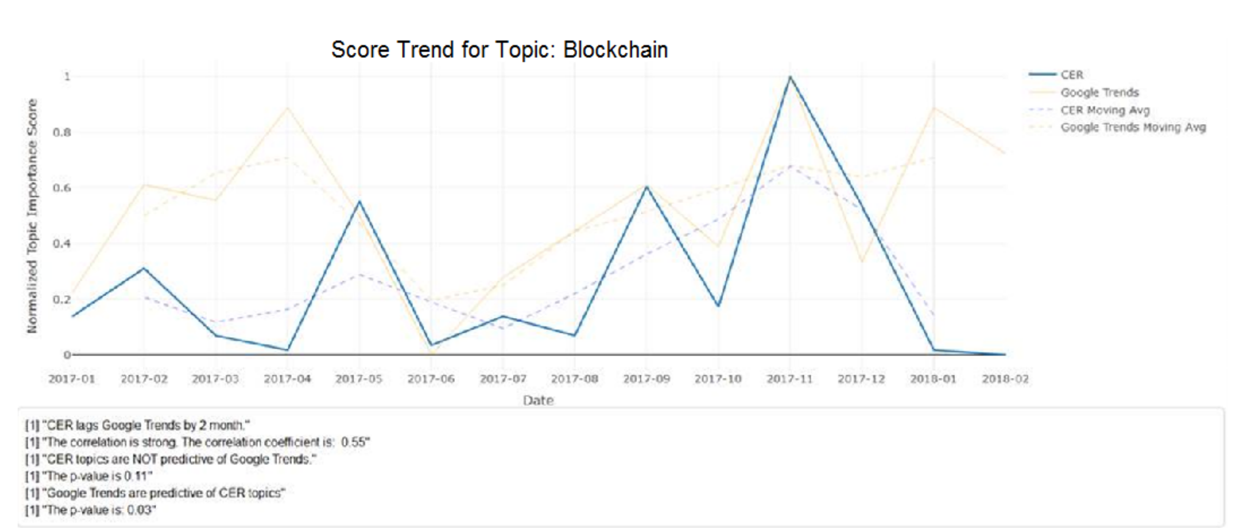
Interactive map showing how the composition of topics vary by region.

Chef's Recommendations



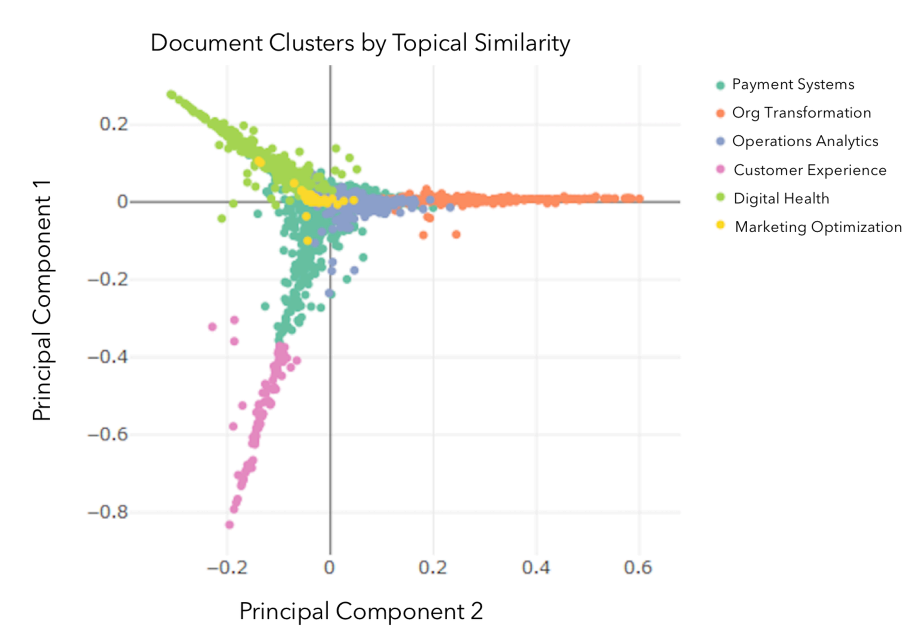
Topic Network

Topical relationships are shown in a network, where highly correlated topics have a thick edge.



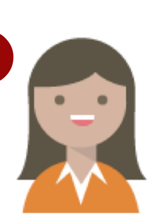
Internal vs. External Signals

We run statistical tests to see if topical trends within the firm lead or lag topical trends from external sources.



Document Clusters

We perform K-means, Hierarchical and DBSCAN clustering on the documents to uncover tribes within the firm.



"The topic network helped my team **deliver better expertise** to my client by identifying correlated topics. For example, I found out that clients seeking solutions for Revenue Management were often want to better understand Personalised Advertising services.

Finding out that Google Trends closely tracked our internally trending topics allowed our Practice to use it as an indicator of **when and where to grow knowledge acquisition efforts**.

The clustering analysis was helpful in **facilitating knowledge-sharing**. It enabled me to find colleagues who worked on similar topics and allowed me to tap into their expertise."

All graphics and quotes are purely illustrative for confidentiality reasons

Our Contributions

- Designed **robust text cleaning** procedures that preserve topics while protecting client confidentiality
- Built **reproducible topic models** for diverse data sources and defined methods for evaluating them
- Created an **original heuristic that finds the optimal number of topics** for any topic modelling algorithm
- Implemented **auto-labelling algorithms** that reduce the need for manual labelling by up to 45 percent
- Developed an **app that facilitates easy topic analysis** across a wide range of business use cases
- Partnered with two Practices within the firm to **operationalise our tool and derive actionable insights**

Source: Graphics were taken from www.freepik.com

Introducing Ratatouille: a Generalizable Goal-Oriented Dialog Bot

McKinsey & Company | MIT Management Business Analytics | Operations Research Center

Team M. Amram – J. Toledano
Faculty N. G. des Mesnards – T. Zaman
Company L. Gerdes – R. Sehgal – I. Pyzow

Problem Statement

Commercial solutions use **human** workforce to frame dialog with **rules**

Domain knowledge
- **Expertise** required to formulate business use case

Conversation transcripts
- Sample **dialogs** required to scope bot features

Business analysts
- Formulate a **base dialog flow** for a given use case
- **Handcraft** a specific series of rules from base dialog flows

Rule-based dialog flow
- Bot leads conversation using preset **question-based flow**
- Bot **classifies** user responses using its handcrafted rules

Our solution leverages **deep learning** to improve **generalizability**

Structured knowledge
- **Database** of structured information required to answer user requests

Extensive conversational data
- Thousands of **labeled conversation** transcripts required to use deep learning

Deep architecture
- Deep learning algorithms **infer patterns** from textual data to frame any dialog

Generalizable model
Can be **extended** by:
- **Switching** database
- **Incorporating** new features by **generating** new conversations
- **Curating** transcripts for any business **use case**

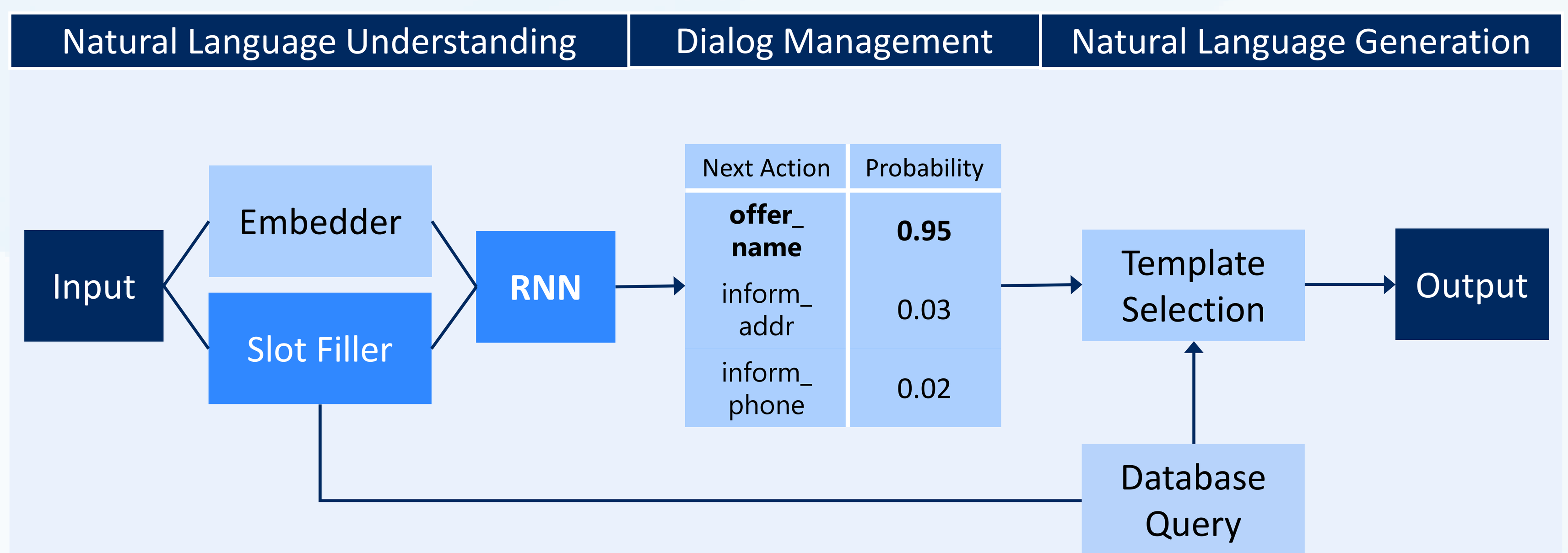
Data Integration & Architecture

Two **enhanced sources** fuel the restaurant recommendation task

Structured Database
- Information about 1,000 restaurants in **Boston**, **Cambridge**, and **Waltham**
- Data **collected** using APIs from Yelp, Zomato, and OpenTable
- Set of scripts **automates** data integration and cleaning

Transcripts
- More than 3,000 **open-source** conversation transcripts published by University of Cambridge
- Augmented with **new features** and **automatically generated sentences** by bespoke parsers

Our **end-to-end** architecture **predicts** the bot's next response



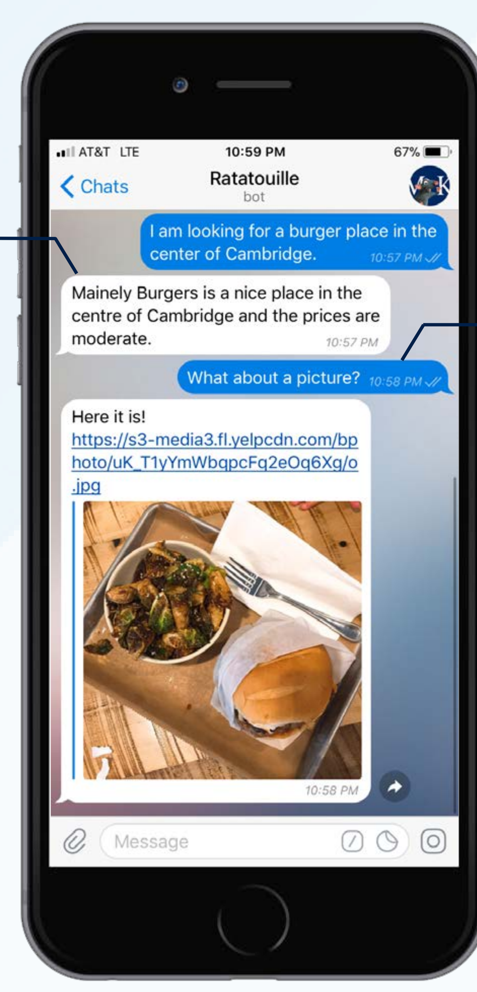
Demonstration Application

Bot takes into account **food type constraint** to suggest a restaurant

Phone number can be retrieved and clicked on

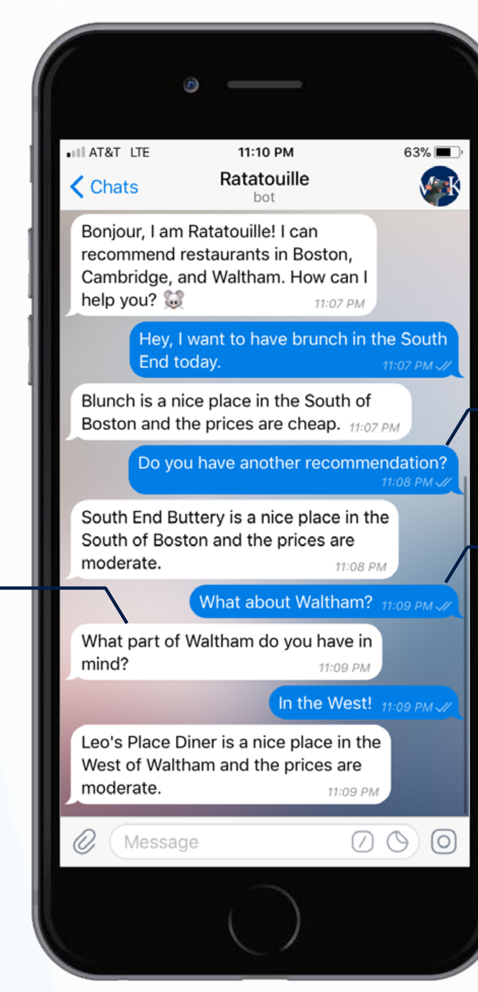
More details, such as the **address**, can be requested

Bot deals with **food type** and **neighborhood constraints**



A **picture** can be requested by the user

Bot understands city switch and inquires about **new preferred area**



Alternatives can be asked for

User can **switch between cities** within a conversation

Project Timeline

On-campus research				On-site internship		
February General literature review	**March** End-to-end architectures	**April** Building Informative DB	**May** Implementing Bot modules	**June** Release of Alpha version	**July** Example level generalizability	**August** Feature level generalizability

Impact

	Customer acquisition	Churn reduction	Cost reduction
Vertical	• Display **advanced capabilities** to prospective customers • Meet customer expectations • **Adapt** rapidly to new customer use cases	• Act on customer **preferences** • Automate customer **satisfaction** analysis • Answer questions with high accuracy **24/7**	• Automate repetitive tasks • Allow exceptional people to **focus** on high-value problem solving • Scale up and down depending on customer requirements
Examples	• **User-friendly** solutions bring about massive adoption	• **Brands** use bots to retain tech-savvy customers	• **Large-scale** implementations have a proven track record for generating **value**

Path Forward

New Use Case

Methodology to apply the architecture to a new business use case:
- Formulate the business use case as **recommendation task**
- Gather and **curate** thousands of conversation transcripts
- Build the corresponding informative database by **scraping** the web
- **Train** the core deep learning modules

Algorithm

Promising research-stage architectural developments:
- **Memory Networks:** RNN that selects and stores relevant dialog chunks in memory
- **Frames Tracking:** adding a memory module to rewind the dialog
- **Reinforcement Learning:** takes into account the future turns of the conversation to optimize the local dialog state

Infrastructure

From a prototype to production-ready solution:
- Training the core RNN with **GPU** reduces training time from 7 hours to 30 minutes
- **Cloud hosting** allows the bot to communicate with several users simultaneously to improve scalability

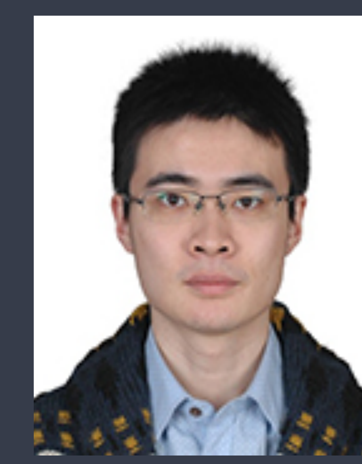
Machine Learning Methods in Credit Risk

2018 Capstone Project (Boston)

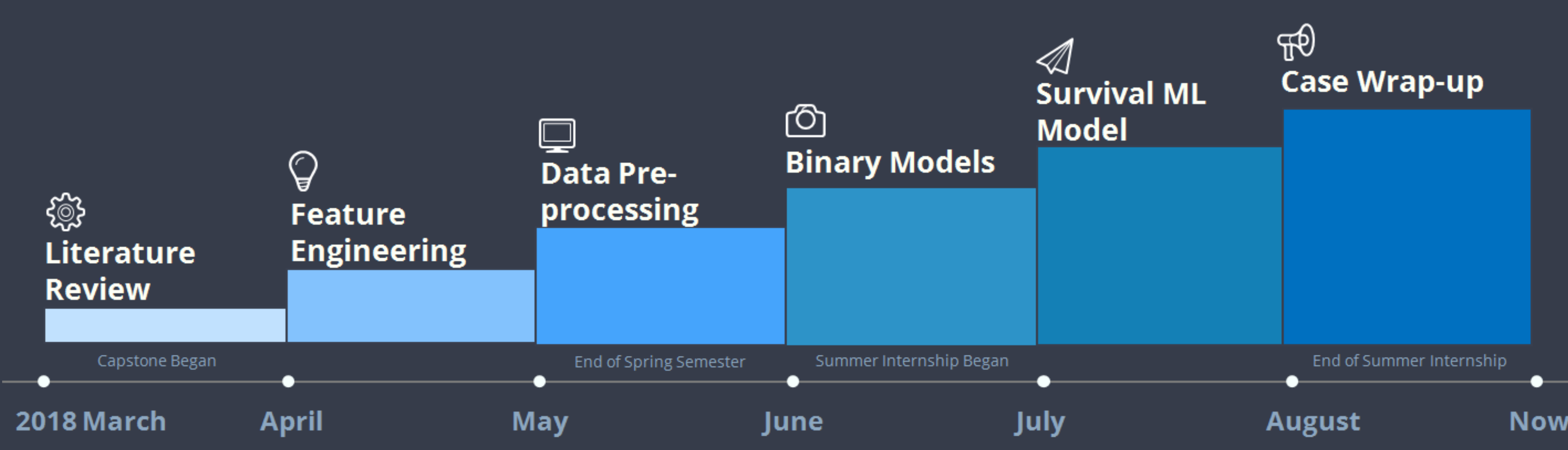

Scott Wang
MBAn
hswang@mit.edu


Tim Yang
MBAn
timsyang@mit.edu


Colin Fogarty
Faculty Advisor
cfogarty@mit.edu


Sean Lu
PhD Mentor
haihao@mit.edu

Capstone Timeline



Problem Statement

The main interest was to help bank determine whether to grant loan depending on the risk of the mortgage. Our goal was to develop a robust model to predict default using available data at the time of the house mortgage application.

Definition of Default

Data Sources

Fannie Mae

Mortgage Loan with terms 10-30 years, acquired by Fannie Mae **2004-2013**

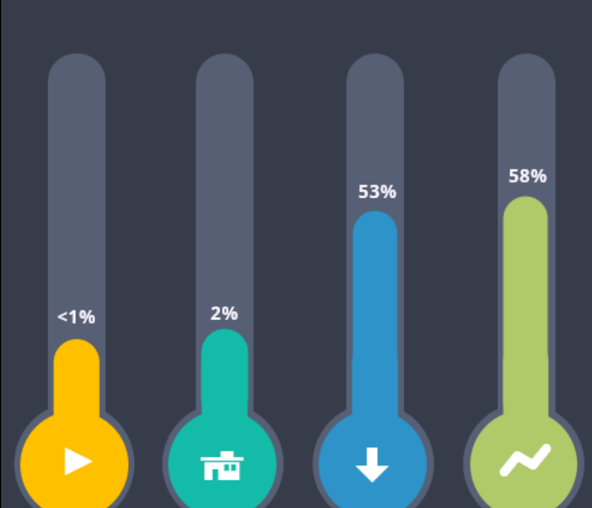
Housing Price Index

Financial Market (stock, bond, gold price)

Labor Market Data

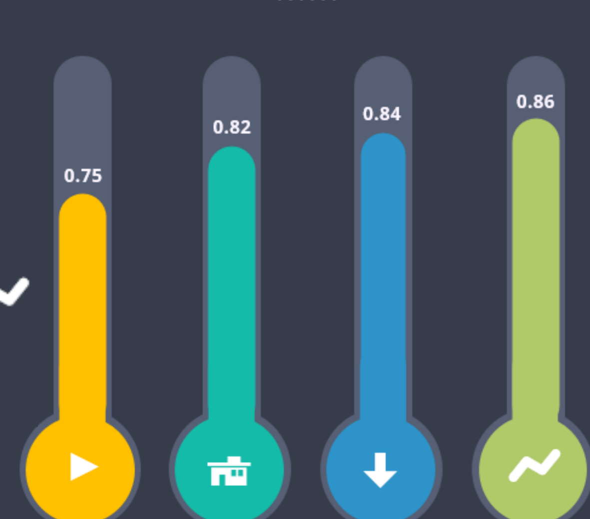
Methodology, Data Processing And Performance

In Penalized LR
Sensitivity

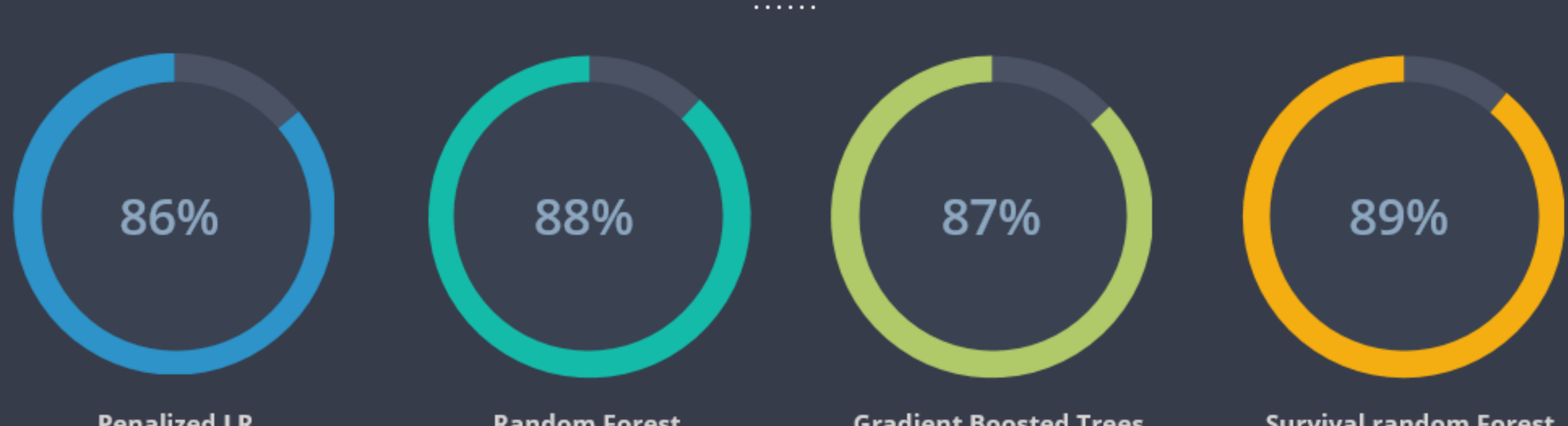


Baseline
Penalized LR (Elastic net)
Added Feature (HPI)
House Price Index
Downsample
House Price Index
Added more features & data
Labor Statistics, Macroeconomic Variables (Stock, Bond, Gold Price)

In Penalized LR
Out-of-Sample AUC

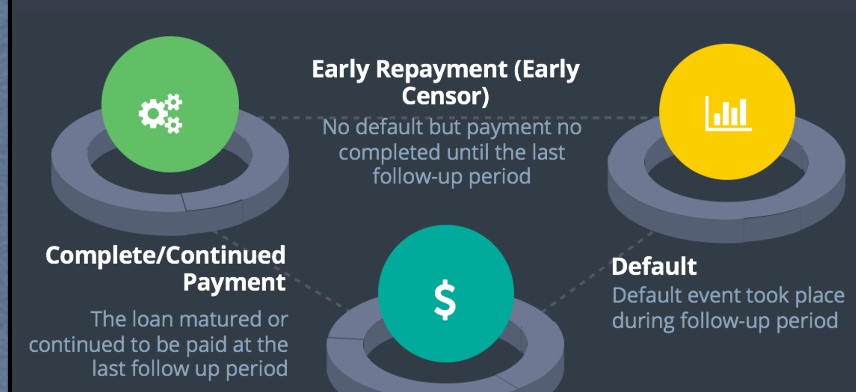


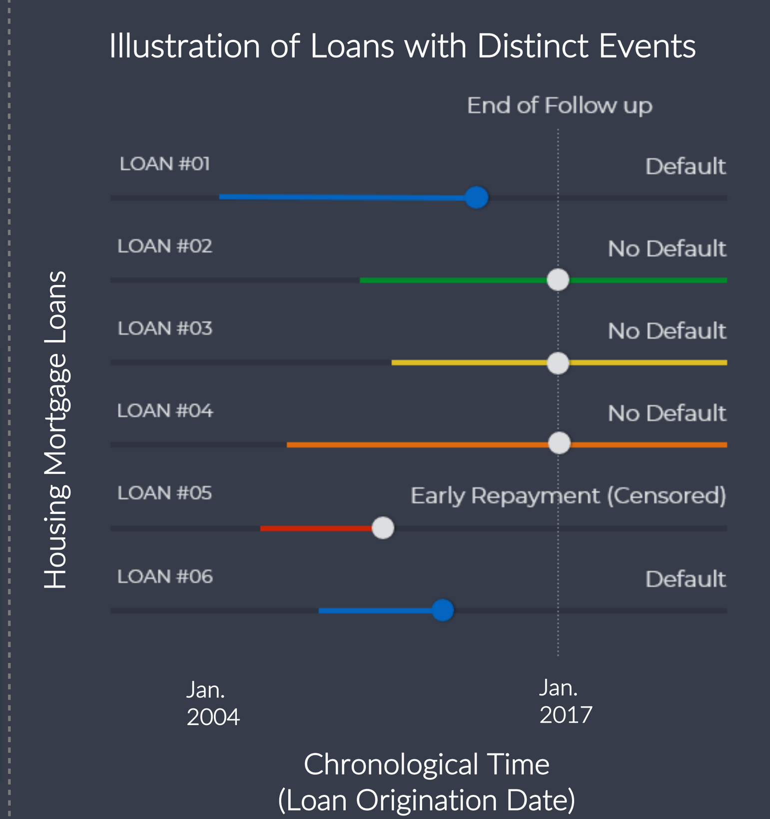
In AUC
Algorithm Performance



Survival Analysis in Default Prediction

- **Classification** (Predict the state): Event/no event/censored
- **Regression** (Predict time to default): Time to event/no event/censor





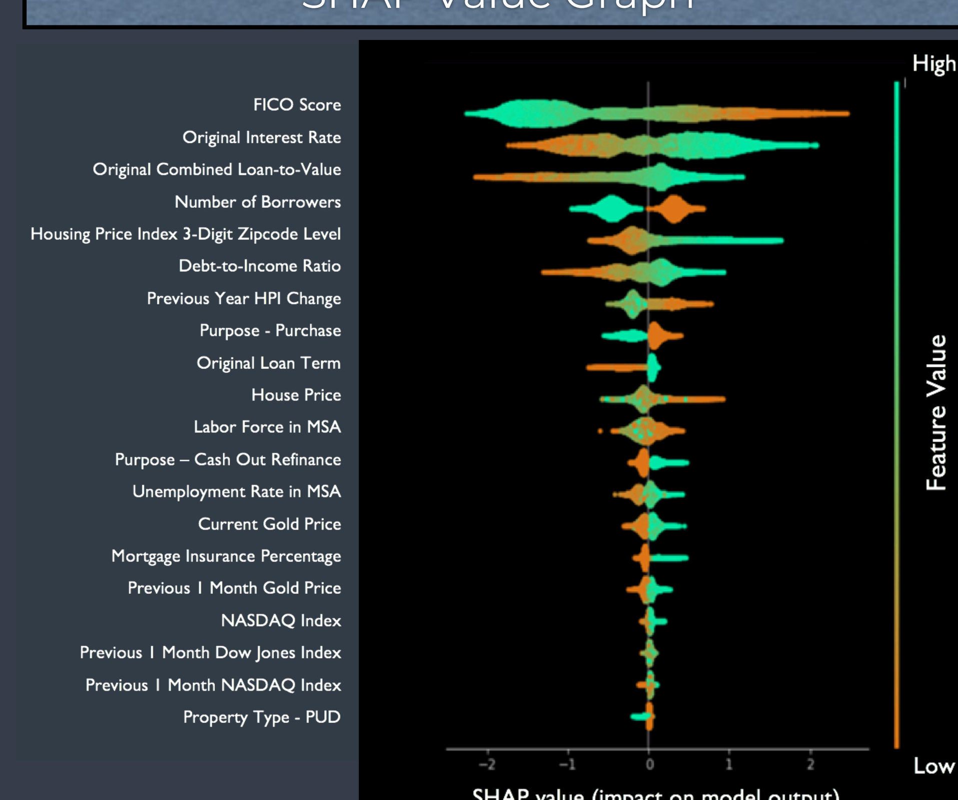
Interpret Machine Learning Models

	Method	Description
Classical Feature Importance Methods	Weight (Split Count)	Calculated as the total decrease in node impurity (weighted by the probability of reaching that node) averaged over all trees of the ensemble.
	Gain (Mean Decrease Gini)	Calculated as the total decrease in node impurity (weighted by the probability of reaching that node) averaged over all trees of the ensemble.
	Permutation	Calculated as the decrease of accuracy of the model predicted on intact OOB samples and OOB samples where the values of a specific variable are randomly permuted.
	Cover	Calculated as the average coverage (the number of samples affected by the split) of the feature when it is used in trees.
Newer Feature Importance Methods	LIME (Local Interpretable Model-Agnostic Explanations)	Globally breaks down the feature components and fits in local (single data-point) level to visualize the effect and direction for each variable on this single data point.
	Tree SHAP (Shapley Additive Explanation)	Ranks feature importance by calculating mean absolute value of SHAP values, which average differences in predictions over all possible orderings of features for each individual observation.

Classic Feature Importance

Cons:
- No magnitude/direction
- Not consistent
- Hard to validate with prior knowledge

SHAP Value Graph



Color → **Magnitude**

SHAP Value → **Direction**

Thick → **#Stacked Individuals**

Pros:
- Has magnitude/direction
- Accurate and consistent
- Easier to interpret complex relationships

Business Impact

Key Conclusions

Machine learning models especially survival models add value and valuable insights

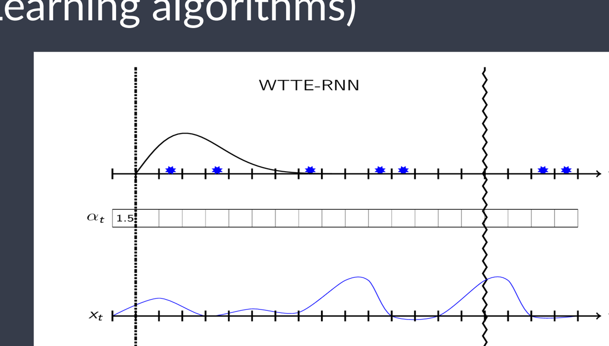
The results of this study will be used to build comprehensive and accurate credit risk models for future customers

Economic Impact via Optimization

Baseline	Machine Learning	Survival Machine Learning
Ground-Truth Economic Loss	Cut 80% Ground-Truth Loss	1% Further Improvement

Future Directions

- Choose cutoffs via portfolio loss optimization
- Real-time portfolio risk monitor with time-varying covariates (through Deep Learning algorithms)



Dynamic Time to Default Estimation

Segmenting Retail Advisors and Optimizing Coverage Model

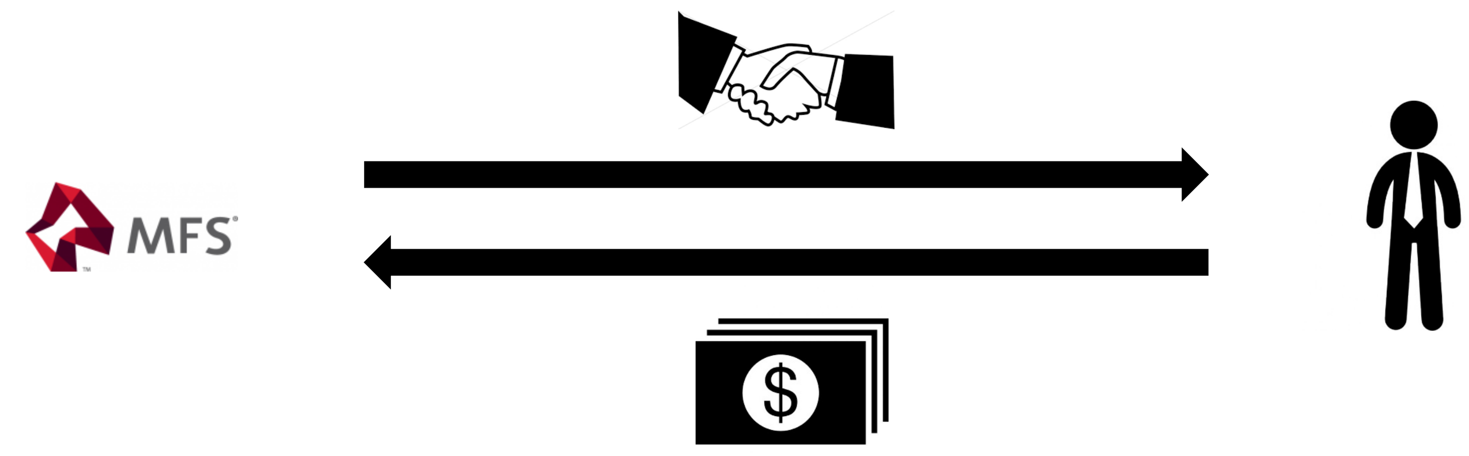
William McEntee[A]; Chinmay Jha[A]; Dimitris Bertsimas[B]; Ryan Cory – Wright[C]; Nadine Kawkabani[D]; Brian Shaw[D]; Brendan Mannix[D]
[A]MIT MBAn 2018; [B]Faculty Advisor, MIT; [C]PhD Student – mentor, MIT; [D]Mentor, MFS Investment Management, Boston, US

Abstract

Clients of Massachusetts Financial Services' (MFS) US Retail business include 300,000 financial advisors spread across the US. With a salesforce of 150 representatives, MFS can only service 7.5% of all the financial advisors effectively.

First, we explore how accurately we can predict the transactions from a financial advisor across various MFS funds in the next six months. Second, we address the problem of optimal resource allocation using an optimization framework to prescribe interaction levels for every advisor using the predictive model.

Third, we identify new approaches for MFS to grow its business by identifying new funds to recommend to advisors. Finally, we propose an extension to the slice recovery algorithm to recommend funds to new advisors.



Market Share & Performance	Activities	Transactions	Advisor Information
MFS's market share across **~13000** client offices	**~1 million** activities(emails, meetings, phone calls) from 2013-16	**~15 million** transactions across MFS funds between 2014-17	Information on **~19000** advisors

- January — *Matched with MFS*
- February — *Developed a deeper understanding of the problem*
- March — *Got access to data and initial exploratory analysis*
- April — *Continued exploratory analysis; First meeting with MFS senior management to set project scope*
- May — *Data preprocessing and preliminary predictive modeling*
- June — *Incorporated feedback on models; Presented preliminary results to MFS President's Council*
- July — *Expanded modeling to prescription and recommendation*
- August — *Delivered final results; Presentation at MIT*

Prediction

How accurately can we predict flows from advisors in the next six months?

- Data: transactions, advisor-specific information, activities, and fund performance
- Methods: regression trees, boosted trees, optimal trees, and classify-then-predict
- Evaluation metric: R^2, mean absolute error (MAE) compared against mean absolute deviation (MAD)

Prescription

Which interactions should we prescribe for an advisor based on the predictive model?

- Data: transactions, advisor-specific information, activities, and fund performance
- Methods: optimal trees and optimization formulation for prescriptive approach
- Evaluation metric: % lift over predicted flows

Recommendation

Which new funds should we recommend to existing advisors?

- Data: purchase history observed across time slices of six months
- Methods: slice recovery, user-based collaborative filtering, item-based collaborative filtering, and matrix factorization
- Evaluation metric: % of new funds purchased which were correctly recommended

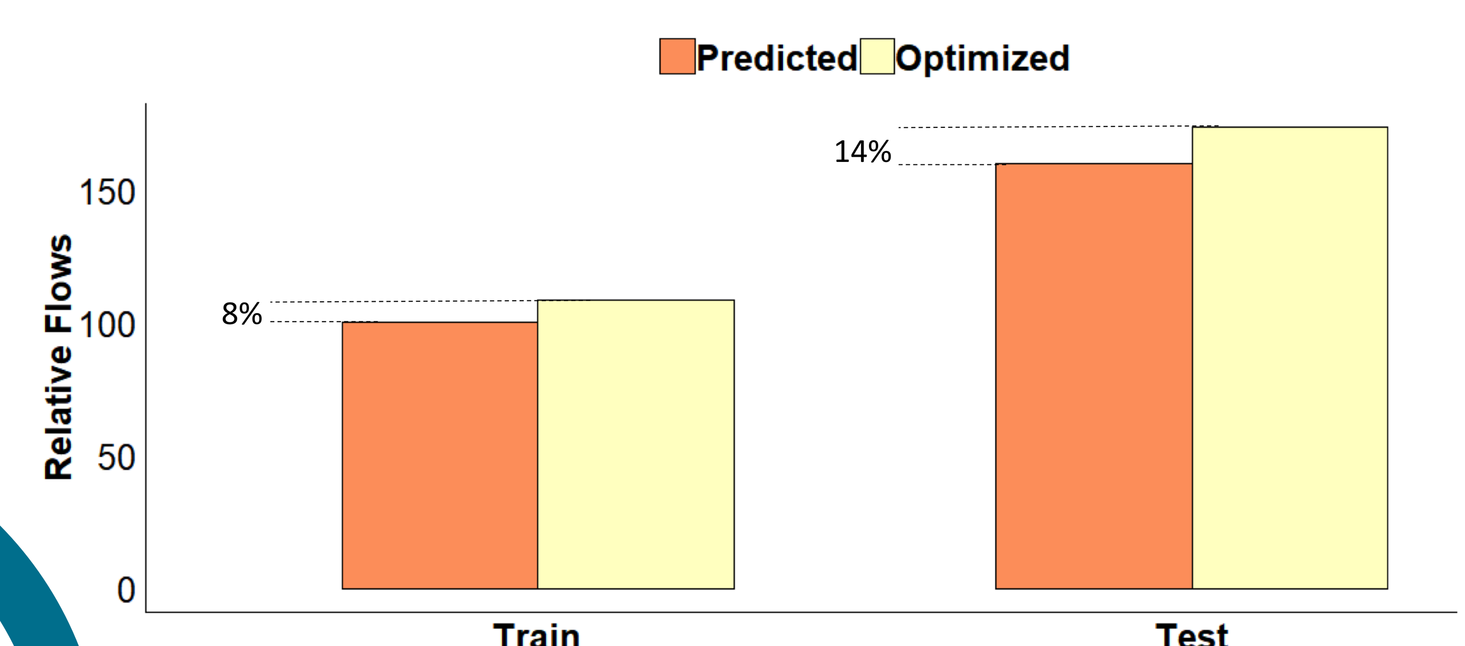
Extrapolation

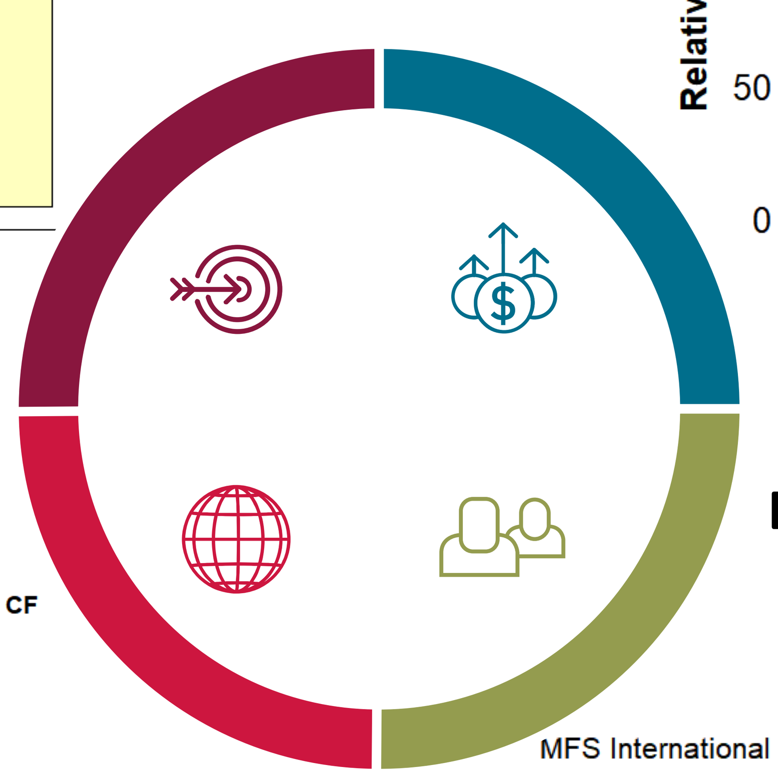
Which new funds should we recommend to new advisors?

- Data: purchase history observed across time slices of six months, advisor-specific information
- Methods: slice recovery and nearest neighbors approach
- Evaluation metric: % of new funds purchased which were correctly recommended

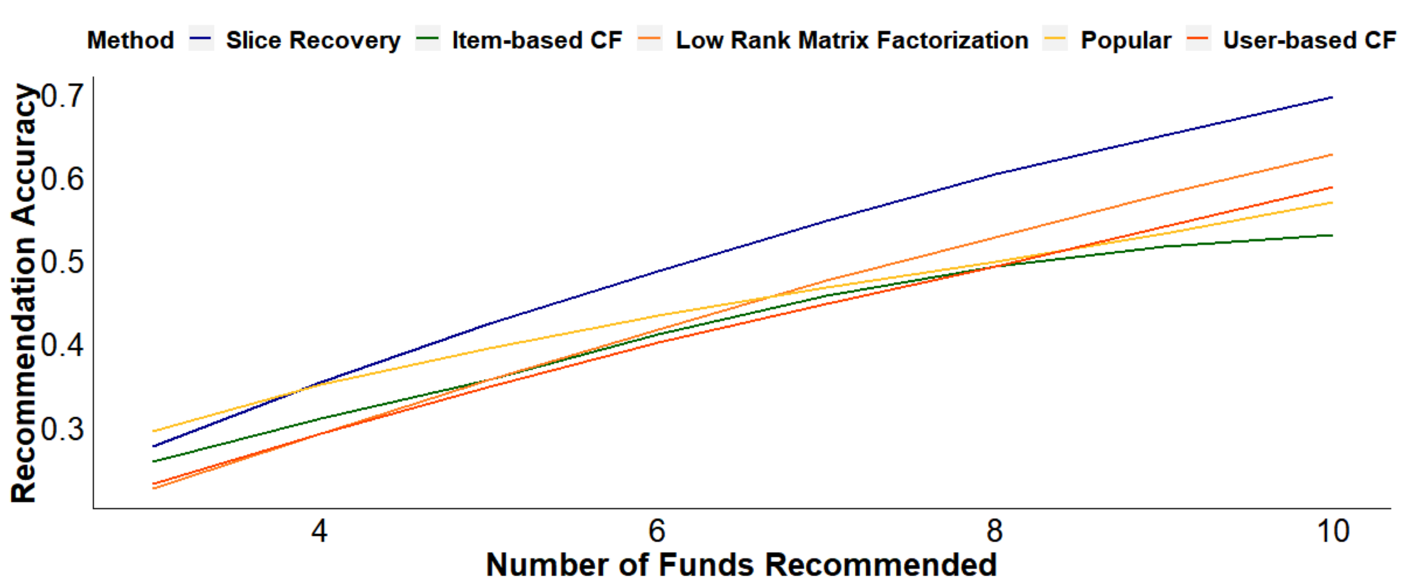
Optimal trees' out-of-sample R^2 is at par with boosted trees

Prescription approach gives lifts over the predicted flows

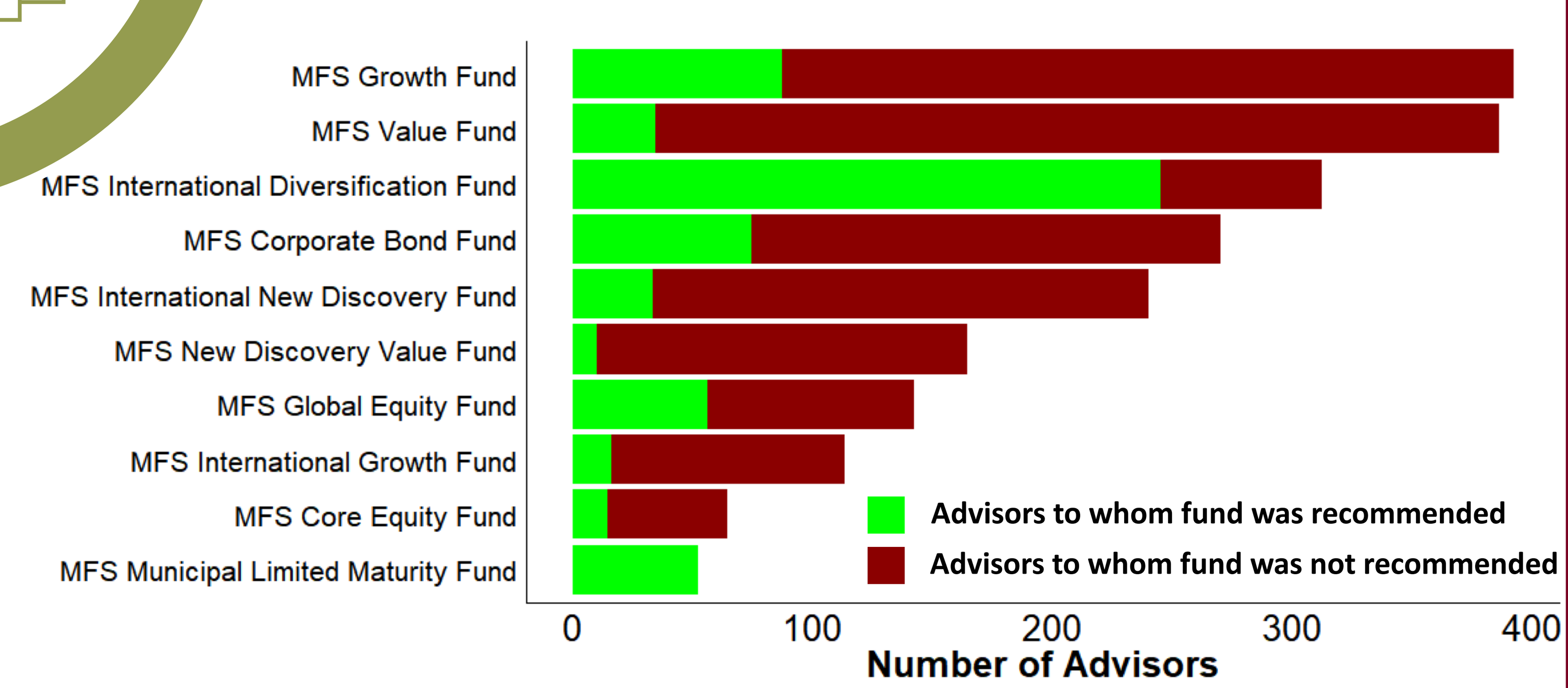




Slice recovery beats incumbent approaches



Extrapolation has consistent performance across funds



Classified 80% advisors correctly as high or low-value

8% lift over predicted flow levels

Recommended 70% of new funds purchased

30% recommendation accuracy for new advisors

1. D. Bertsimas, J. Dunn. **"Optimal Classification Trees"**. *Mach. Learn., 2017*
2. A. Li, V. Farias. **"Learning Preferences with Side Information"**. *Mang. Sci. (to appear)*
3. D. Bertsimas, N. Kallus. **"From Predictive to Prescriptive Analytics"**. Mang. Sci. (under review)

StubHub

San Francisco, USA

Automated Ticket Trading

Michael Li   Charles Hermann

Advised by: Prof. Georgia Perakis, Max Briggs, and Rim Hariss

NBA Sales 2014-2017

MLB Sales 2018.05-2018.08

2018.2-2018.4

Feature Generation

- The ticket reselling market is constantly changing, demanding market awareness
- Generated 20 market features, 6 game features and 3 environment state features
- Eg. Median listed price in game, win/loss ratio of home team, total value sold in section, etc…

2018.7-2018.8

Covariate Unbiasing

- We would like to buy high and sell low, but the price variable is confounded with others
- Created novel estimation method "Dual Machine Learning" to debias price
- Price sensitivity of resulting model almost doubled

2018.5-2018.7

Estimation of Sales

- We try to predict whether a ticket eventually sold on StubHub or not as classification
- Tested 5 different prediction methods ranging from logistic regression to neural networks
- Random Forest + Gradient Boosted Trees performed best [AUC: 0.86 (NBA) / 0.81 (MLB)]

2018.6-2018.8

Price Optimization

- We would like to optimize our tickets over price to achieve best revenue
- We introduced multiple variance constraints to control for uncertainty
- Variance estimation was explored but eventually removed – scalability remains weak

NBA Trading Profit: $6.7 Million/yr

MLB Trading Profit: >$20 Million/yr

Project Phoenix – Wildfire Prediction in Canada

OPERATIONS RESEARCH CENTER
MIT MANAGEMENT SLOAN SCHOOL
Zakaria El Hjouji, Louis Lecluse
Dimitris Bertsimas, Lea Kapelevich, Carine Simon
Christian Klose, Nataliya Le Vine
Swiss Re
New York, NY

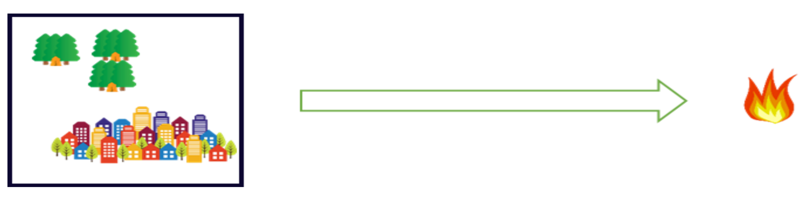
Problem

Intro: Wildfires are very rare and costly events. As of today, wildfires have cost the (re)insurance industry billions of dollars. For example, Fort McMurray's fire in 2016 is expected to cost more than $9 billion. While some people think that such events are one-off events, others believe that there are common atmospheric and geographic patterns that lead up to wildfires.

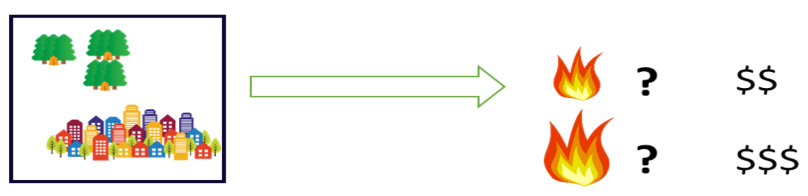
Project Statement: In this project, we hope to harness the power of Machine Learning and Artificial Intelligence to recognize those patterns. Our goal is to understand the risk of wildfires for any region in Canada in time and space through predictive modelling.

Our model can be broken down into two:

- **Fire Occurrence Model:** For each location *(x,y),* this model predicts whether such location will experience a fire in one month, two months, … , up to fifteen months.

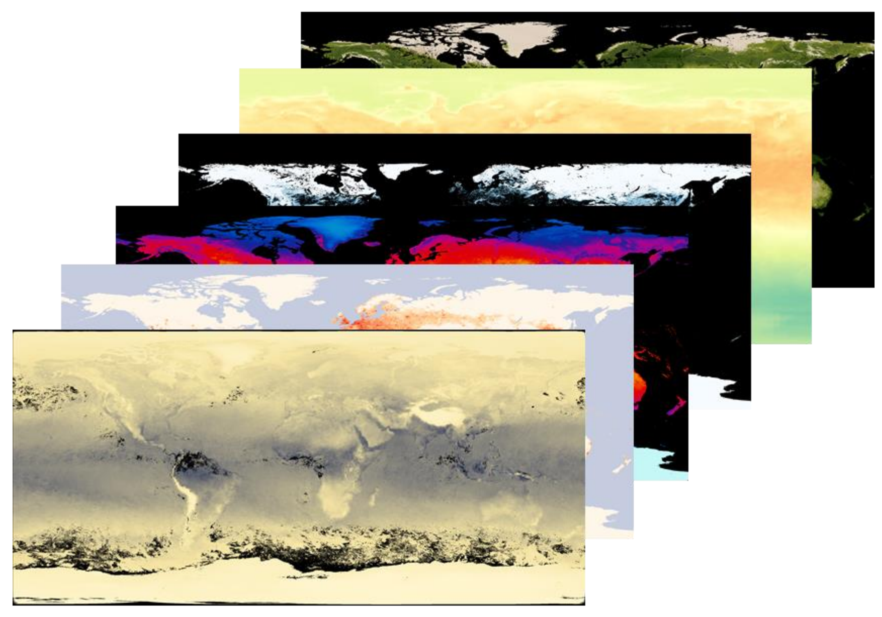


- **Fire Severity Model** : For each location *(x,y),* this model predicts the size of the fire such location might have in one month, two months, … , up to fifteen months.



Our Data is Heterogeneous in the following ways:

- **Different Sources:** Our data comes from different sources such as NASA Earth Science, Swiss Re's proprietary data and other publicly available data.
- **Different Time & Space scale:** Our data comes in different scales. For instance, some features are at 0.1 degree scale *(10 km),* while others at 1 degree scale. Features also cover different timespans.
- **Different Forms:** Our data comes in both Structured and Unstructured Format (*i.e. Satellite Images).*



Data

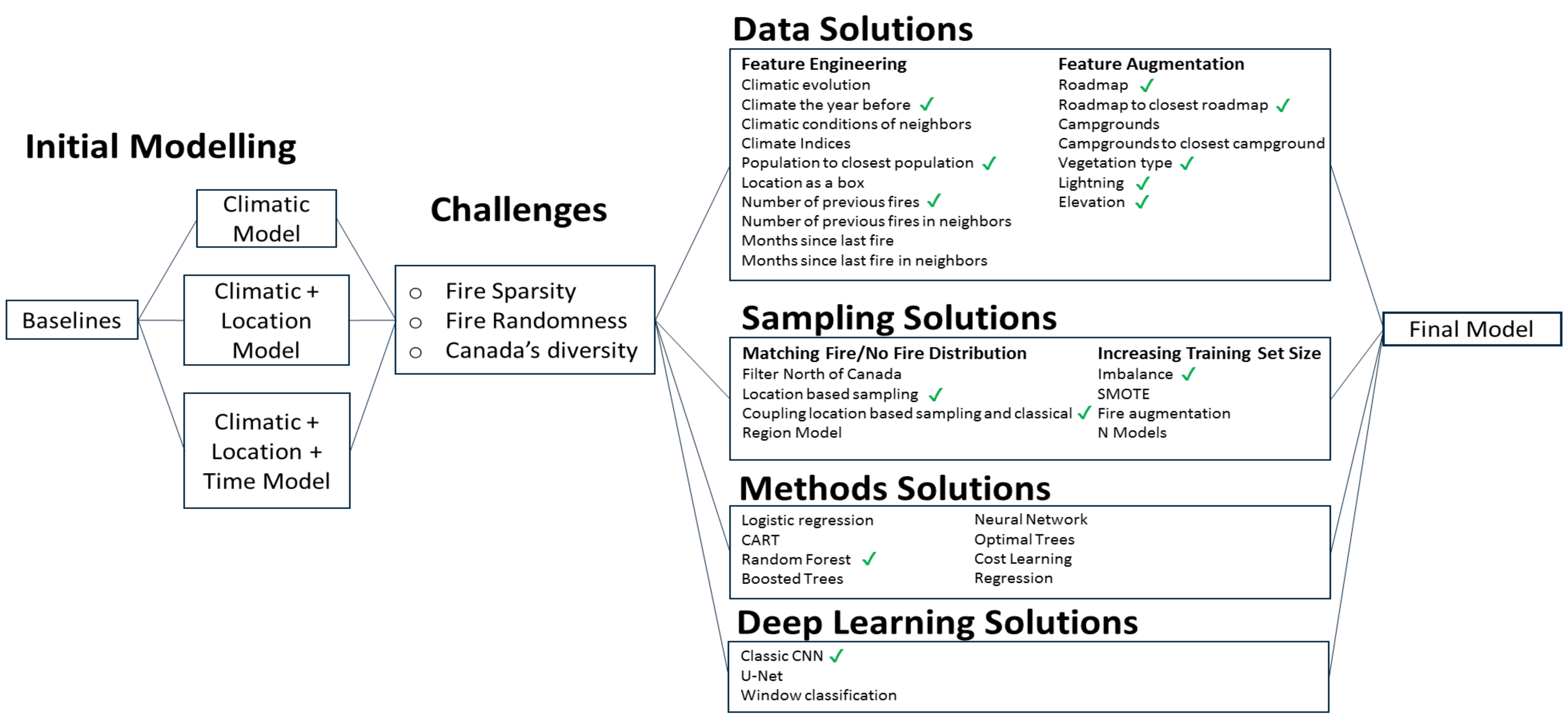
Our Data encompasses major wildfire predictors. They can be broken down into four different categories:

- **Climatic features:** Such features are important as they allow the model to capture climatic patterns under which wildfires occur. For example, wildfires occur frequently in dry areas with high Surface Temperature.
- **Geographical Features:** Wildfires occur under specific geographical settings. For instance, wildfires occur in places with high vegetation and low elevation.
- **Sources of Ignition:** These features help the model capture some of the randomness that triggers fires. For example, in June 2018, lightning sparked nearly 100 wildfires in British Columbia in 24 hours. Hence, taking into account the lightning activity in each region is key
- **Fire History:** Some areas might have high wildfire activity, however, our features are unable to set such regions apart. Using the history of fires as a feature allow the model to form a prior about this region's risk.

Climatic	Geo	Sources of Ignition	Fire History
Temperature	Elevation	Lightning	Number of Past Fires
Wind speed and Direction	Vegetation Index	Campground	Month Since last Fire
Drought Index	Vegetation Type	Roadmaps	
Water Vapor	Snow Cover		
Net Radiation			

Modelling

There are many challenges to our problem, chiefly: data imbalance (0.1% fires), wildfires can be random and skewness of fire sizes. To overcome those challenges, we explored different modelling approaches. We started with strong baselines and initial modelling attempts providing us with insights and performance references. We then increased our performance by closely exploring our features and varying our sampling methods and modelling techniques.

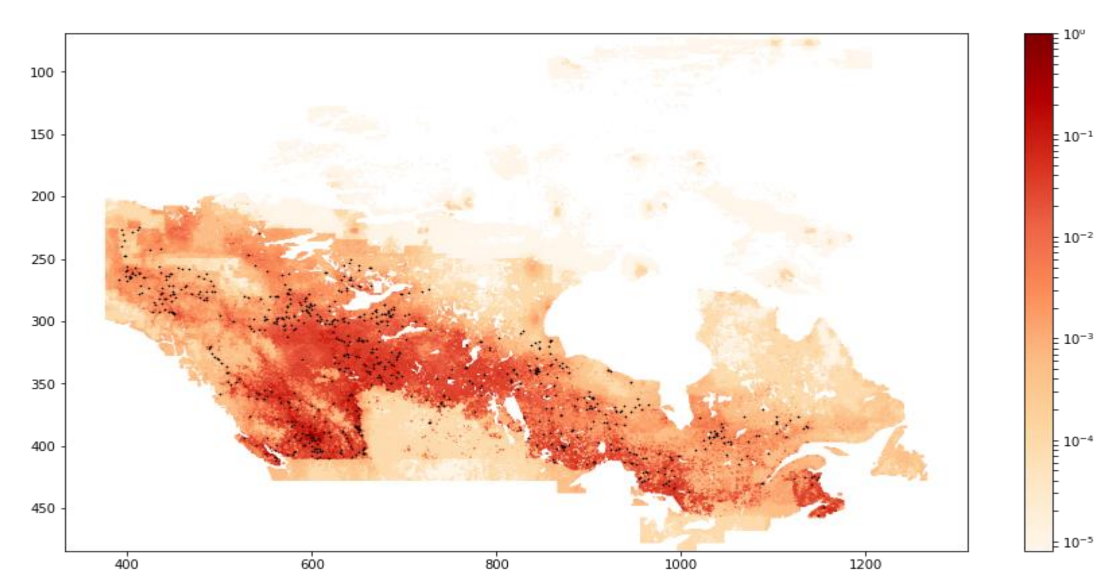


Best Occurrence Model

Through our modelling journey, we identified the key features, the model architecture (Random Forest) and sampling methods (Imbalance and location-based sampling) that yielded the best out-of-sample performance for the occurrence model. Below are the features selected.

Performance: This model has the best Average Performance Score: 10% (baseline: 3%) with a recall of 88%. Its performance remain strong as we predict further in the future. It is able to predict with good performance 15 months into the future.

Climatic conditions	Vegetation Index	Lightning	Number of past fires
Climatic conditions year before	Snow cover	Roadmap	Number of past fires in neighbors
Climatic conditions evolution	Vegetation type	Closest roadmap	Month since last fire
Climatic conditions of neighbors	Elevation	Population	Month since last fire in neighbors
Climate indices		Closest population	

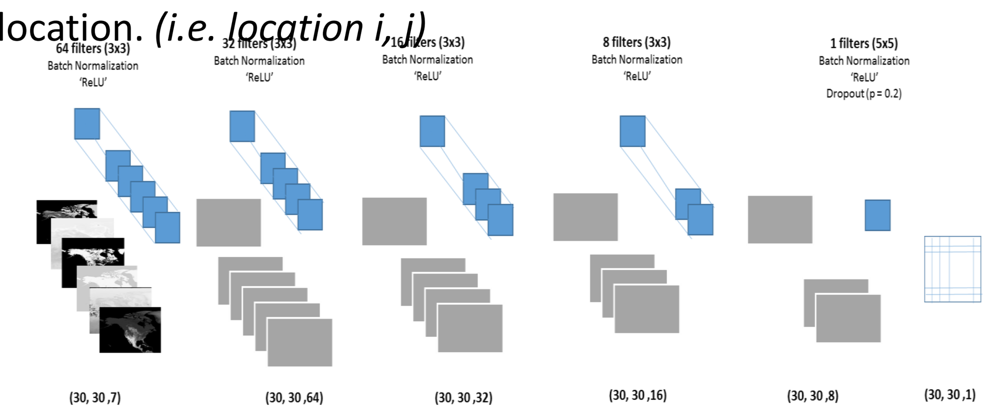


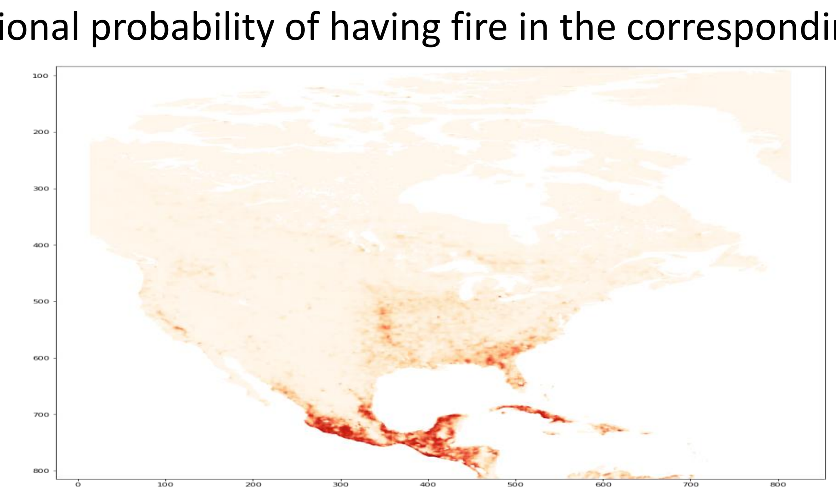
Feature map

Deep Learning

Motivation: Random Forest and Structured Data models are sometimes unable to capture complex patterns, mainly when it comes to spatial correlations. Also, given the nature of our data (i.e. satellite images) and the recent success of Deep Learning in computer vision, we believe that it is important to explore such models.

Models: We explored various models and architectures, spanning from Classical CNN to semantic segmentation architectures *(e.g. U-net, TernausNet, etc..*). The model that delivered the best out-of-sample performance is a CNN that takes as input a 3D-matrix (30x30) with 7 channels (each representing a different feature), and passes it through a series of convolutions with same padding and outputs a 2D-matrix such that each element *(i, j)* represents the conditional probability of having fire in the corresponding location. *(i.e. location i, j)*





Performance: This model was trained on North America data and delivered the best performance. The Average Precision score was 34% with a recall of 81%. Unlike the structured data model, this model can be easily scaled to the global scale.

Impact

When an underwriter needs to understand the risk associated with wildfire for a particular region, they use Classic (probabilistic) models. However, such models are based primarily on wildfire history in the region, which becomes cumbersome when such data is not readily available. In addition to that, such models provide static risk scores and cover regions at a macro-level, which does not allow underwriters to build risk scores at a granular-level, or asset-level. Our model uses state-of-the art Machine Learning methods to help underwriters build a **forward-looking** view of the wildfire risk on a monthly basis and at a micro region *(10x10km).*

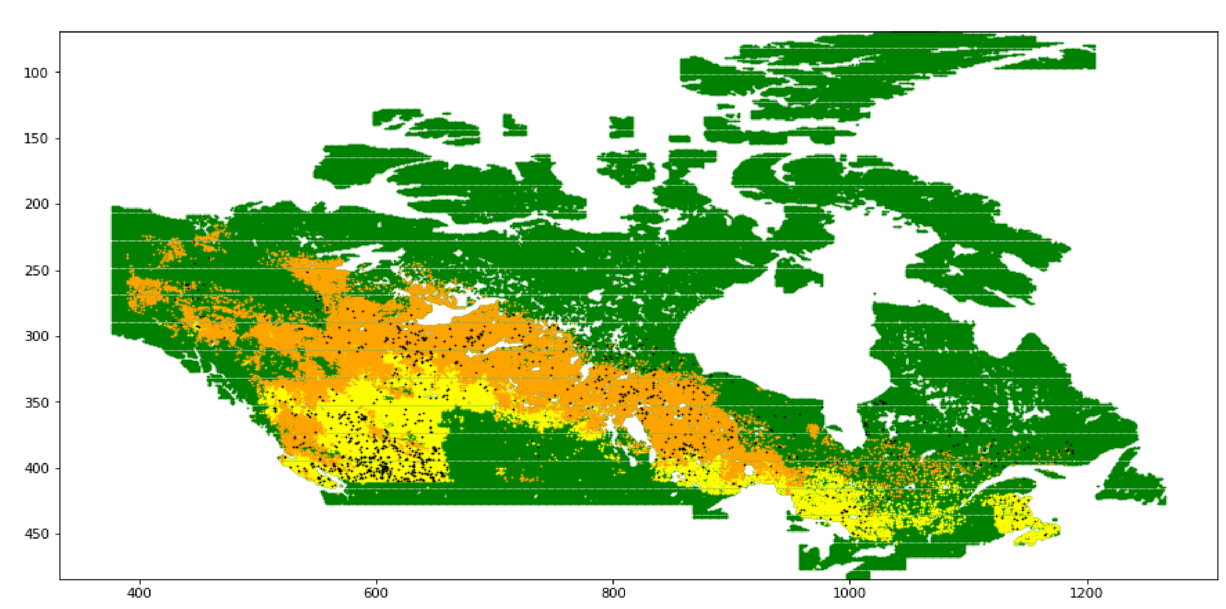
Loss Frequency Curve: Loss Frequency curves depict the distribution of area burnt by wildfires on a particular region. Using our model with distribution fitting techniques, one can develop such curves at a pixel- and monthly-level. These can then be aggregated to cover larger regions and time periods.

Fort Mc-Murray: Our model accurately detects the 2016 Fort McMurray event: it predicts a 2% increase in hazard for the May-June-July 2016 period with respects to 2015 levels.

Best Severity Model

To obtain a thorough evaluation of wildfire risk, it is necessary to estimate the size of the wildfire event. The severity model builds on the occurrence model to predict size of wildfires. We broke down the size of wildfires into two: Small and Large.

Performance: This model has strong performance, it is able to catch more than 50% of the potentially costly wildfires with a relatively high precision.



Walmart eCommerce Capstone Project

Creating a Tool To Diagnose Out Of Stock Causes

Rachel Insoft and Sam Smith

MIT Faculty Mentor: Steve Graves | MIT PhD Mentor: Li Wang

Project Location: Jet HQ, Hoboken, NJ, USA

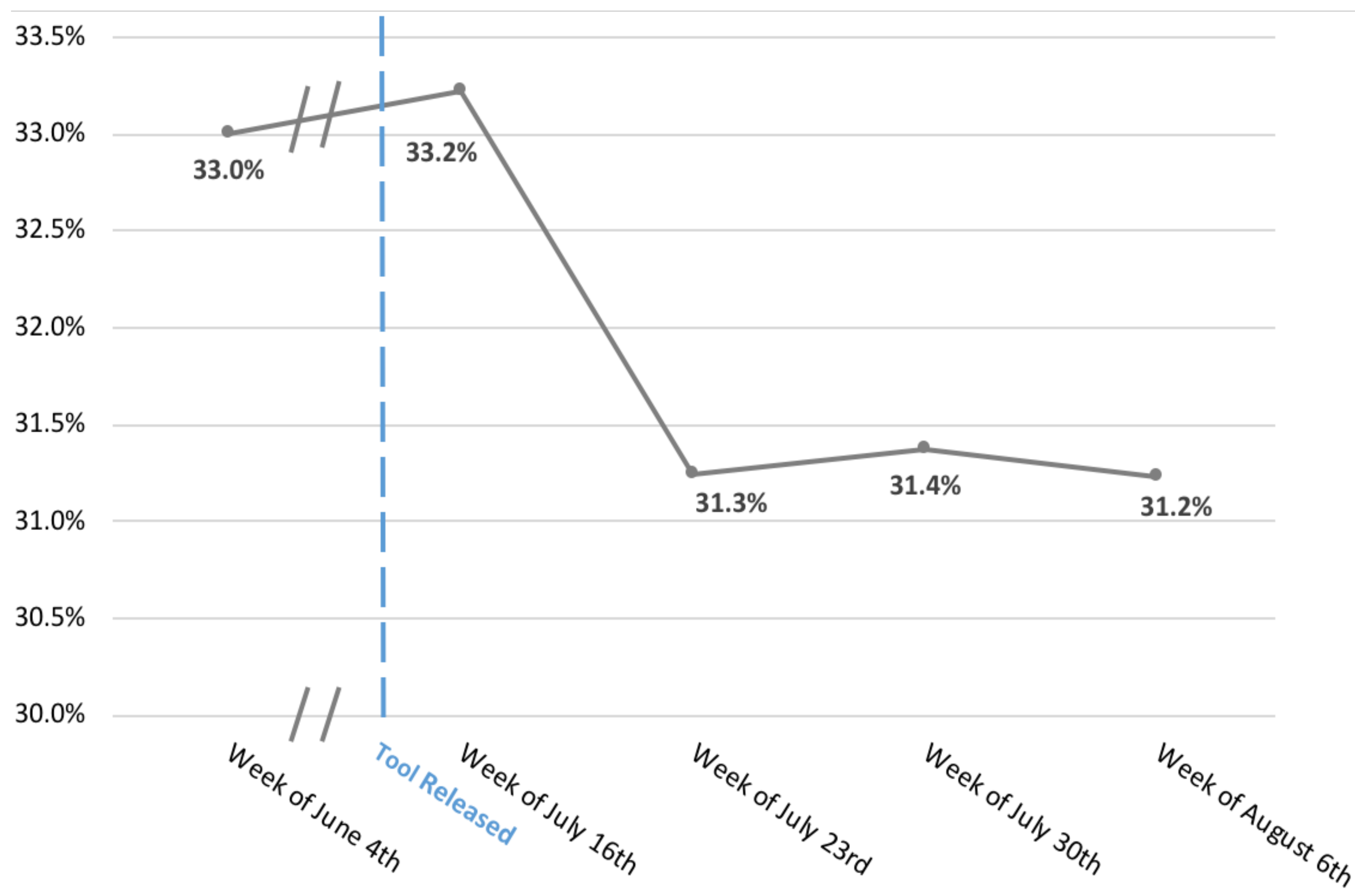
Project Scope

Our team's role within Walmart was within the Supply Chain Product Management and Analytics team. We were tasked to create a tool which could help supply chain managers diagnose why their items were going out of stock.

This is an important and dire problem for Walmart – when we first released roughly 33% of item-node (fulfillment center) pairs were flagged as being out of stock across the six main Walmart fulfillment centers.

Our team's final iteration of this tool had 10 different views to show various weighted and unweighted curs of the network's out of stock situations, as well as over 20 filters.

Walmart Item-Node Out of Stock %

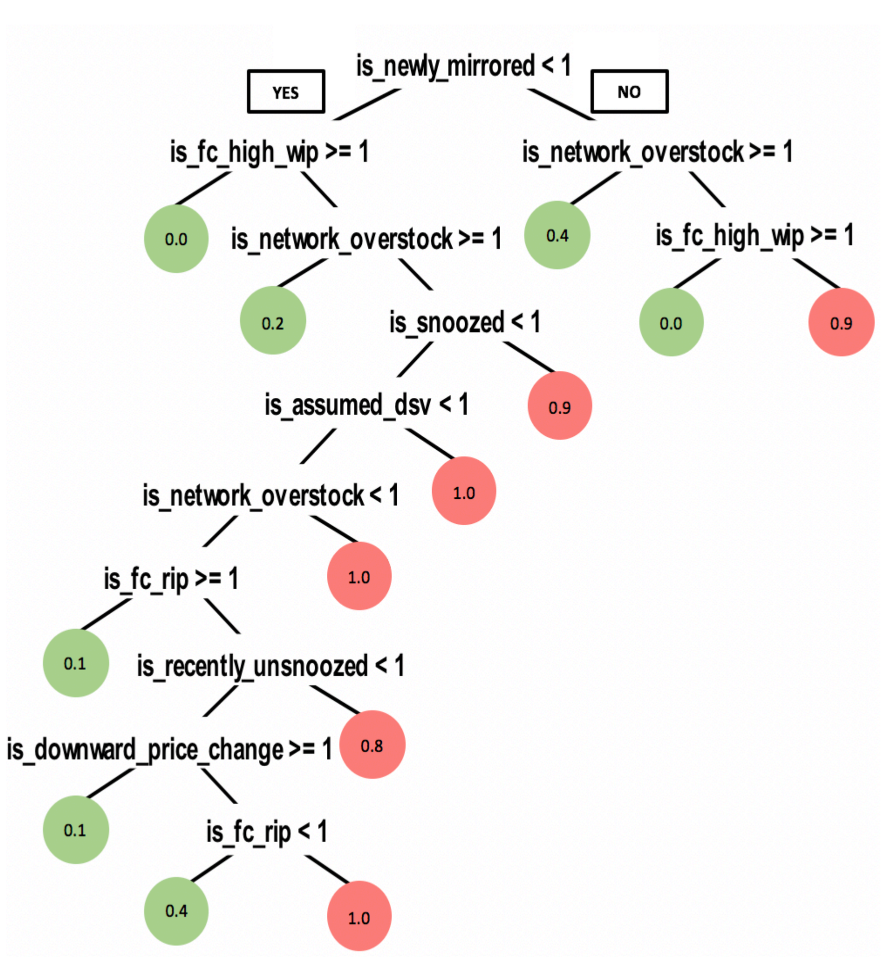


Every 10-basis point increase in item-node in-stock percentage affects roughly $37,440 of demand per week top-line. By the end of our time with Walmart, the out of stock percentage had **dropped by 200 basis points.**

This is an average increase in-stock value of nearly **$750,000** in weekly demand.

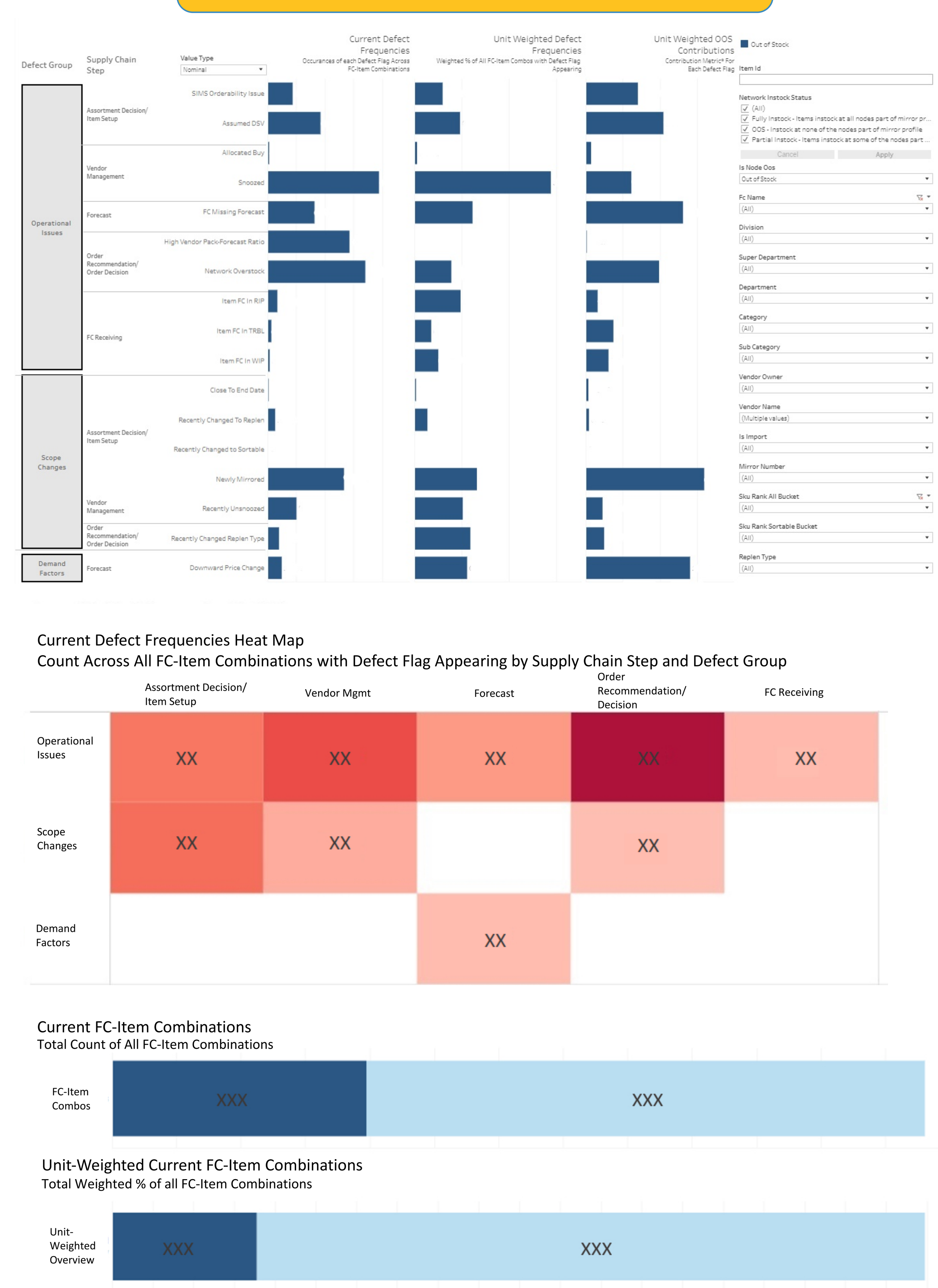
Data Engineering and Modeling

Utilizing 2,500+ lines of Hive queries and a Hadoop architecture, we engineered an **ETL ("extract, transform load") data pipeline** that refreshes automatically each day. The final product of this process were two tables: a warehouse that shows a summary of all relevant stockout metrics as defined by our team at the item-node level as well as a database that breaks each item-node combination down by defect and can be leveraged for more flexibility in data visualization.



We used CART models to predict out of stock situations for every division of items within the Walmart network. Our CART results showed us that primary splits (variables most predictive of out of stock situations aka "root causes") varied greatly by division. An example for the Everyday Living division is shown at left.

Sample Dashboard Views



February – May: Preliminary Data Exploration/Modeling → June: In-Depth Hive Querying & Scripting → June 29: Initial Beta Version Tool Release → July-Early Aug: Tool Updates & Root Cause Modeling → August 6: Final Tool Version Released

Special thanks to the entire Walmart/Jet team for being incredible sponsors and giving us a great summer, to the entire MIT team for their help and support in the organization of this project, and to all others who made this fun, relevant, and impactful project possible!