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 merchant 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.

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 SKU level by store for around 3000 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.

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. 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.

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.

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

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.

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.
**Project Overview**

**Project Importance**
- Luxury retailers make little revenue from ready-to-wear clothes.
- Approximately 90% of revenue comes from handbags, shoes, cosmetics, 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.

<table>
<thead>
<tr>
<th>Women's handbags</th>
<th>Full-price items</th>
<th>European stores</th>
<th>Retail locations</th>
<th>Gross sales</th>
<th>New Collection</th>
</tr>
</thead>
</table>

**Project Timeline**

<table>
<thead>
<tr>
<th>January - February</th>
<th>Exploratory Data Analysis</th>
<th>Initial Demand Forecasting Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>March - April</td>
<td>Feature Engineering</td>
<td>Constructing Panel Data</td>
</tr>
<tr>
<td>May - June</td>
<td>Pairwise Comparison Research</td>
<td>Efficient Algorithm Implementation</td>
</tr>
<tr>
<td>July - August</td>
<td>Sophisticated Demand Forecasting Models</td>
<td>Summer Capstone Showcase</td>
</tr>
</tbody>
</table>

**Data Overview**

**Raw Data**
- We were given four datasets:
  - Stores
  - Products
  - Transactions
  - Inventory

**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.

<table>
<thead>
<tr>
<th>Flagship Store</th>
<th>High volume, high sales and stock, high price-point.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large City Locations</td>
<td>comparable sell-through rate to the flagship store, but with fewer overall sales and less expensive sales.</td>
</tr>
<tr>
<td>Resort Locations</td>
<td>low volume store with high price-point, high sales and high revenue to square footage ratio.</td>
</tr>
<tr>
<td>Traditional City Locations</td>
<td>not carry the highest price-point, but do have high sales through and high revenue to square footage ratio.</td>
</tr>
<tr>
<td>Low Volume Stores</td>
<td>assortment of low volume store with the overall lowest price point (includes airport locations).</td>
</tr>
</tbody>
</table>

**Data Processing**

**Clustering Data using k-Prototypes**
- We applied clustering to our data using k-prototypes, which integrated 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 performance.
- These clusters helped us build new features, such as historical sales and unique store selling behaviors.

**Reducing Proportion of Null Values**
- We imputed missing values in the data frame using analytical imputation 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 features 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.

**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 few historical sales for the SKU so we added 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.

**Feature Importance**
- Below is the plot for the Fall-Winter 2016 season, for which we trained our model.
- 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.

**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.

**Recommendation: Potential Demand**
- Sales are a proxy for demand since stock-outs could have caused fewer purchases.
- 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 output 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%.

**Impact**
- Our Capstone project resulted in a better forecasting model in comparison to the client’s models. 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.
The Project

- Hopes to capture consumer sentiment and preferences
- Proposed we forecast which models to forecast with
- Used to guide buying and product development strategy

Influencers

3.5 million tweets from 1240 different accounts

Social Media Data

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...)

Google Trends

Number of Google search queries for “Backpack”

Trend Forecasting

- Using APIs to live connect to entire Google Search corpus
- Refining the appropriate category filters
- Text: data reflects demand of trends that already hit market - need earlier signals

The Team

- Mentor: Vivek Farlas
- PhD Advisor: Deeksha Sinha

The Impact

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

Project Timeline

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

Client site

Milestones

- MIT · Occasional Office Visits
- Training in SF
- BCG Boston Office

Imagery of text
‘What options would you like in your BMW i8?’

Option take-rate forecasting for BMW Group

**Model**

BMW X6 XDRIVE 50I

**Options**

- Back seat entertainment
- Display key
- Bi-LED lights

**Take-rates**

(illustrative)

**Data source:** VDWH
- Individual sales data
- Raw data is a table containing all features and options
- Raw data serves as basis for full project scope

**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: 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

**SOP + 6 months**

<table>
<thead>
<tr>
<th>MIT Model Uninformed</th>
<th>MIT Model</th>
<th>Baseline</th>
<th>BMW market forecasts</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOP + 6 months</td>
<td>Overall</td>
<td>6.5%</td>
<td>8.5%</td>
</tr>
<tr>
<td>G30 540i</td>
<td>Overall</td>
<td>5.2%</td>
<td>7.0%</td>
</tr>
<tr>
<td>G30 M550i XDRIVE</td>
<td>Overall</td>
<td>6.0%</td>
<td>7.4%</td>
</tr>
<tr>
<td>G01 X3 DRIVE 30i</td>
<td>Overall</td>
<td>4.9%</td>
<td>8.0%</td>
</tr>
</tbody>
</table>

MAE observed (in % pt take-rate)

<table>
<thead>
<tr>
<th>Existing models</th>
<th>New Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>4.1%</td>
</tr>
<tr>
<td>G30 540i</td>
<td>8.5%</td>
</tr>
<tr>
<td>G30 M550i XDRIVE</td>
<td>12.4%</td>
</tr>
<tr>
<td>G01 X3 DRIVE 30i</td>
<td>6.8%</td>
</tr>
</tbody>
</table>

**SOP + 6 - 11 months**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>6.5%</td>
<td>8.5%</td>
<td>10.5%</td>
<td>7.8%</td>
</tr>
<tr>
<td>G30 540i</td>
<td>5.2%</td>
<td>7.0%</td>
<td>9.6%</td>
<td>6.4%</td>
</tr>
<tr>
<td>G30 M550i XDRIVE</td>
<td>6.0%</td>
<td>7.4%</td>
<td>11.8%</td>
<td>8.3%</td>
</tr>
<tr>
<td>G01 X3 DRIVE 30i</td>
<td>4.9%</td>
<td>8.0%</td>
<td>12.2%</td>
<td>10.1%</td>
</tr>
</tbody>
</table>

MAE observed (in % pt take-rate)

<table>
<thead>
<tr>
<th>Data (.csv)</th>
<th>Pipeline steered by main document (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data (VDWH or BV-15)</td>
<td>Initial data frame.R</td>
</tr>
<tr>
<td>Monthly take-rates per ID</td>
<td>Produce monthly option take-rates per unique ID</td>
</tr>
<tr>
<td>Regular and smoothed take-rates per ID</td>
<td>Smoothing.R</td>
</tr>
<tr>
<td>Data including prediction and target variables</td>
<td>Perform LOESS on take-rates across full timespan</td>
</tr>
<tr>
<td>Multiple data sets including time-related variables</td>
<td>Target.R</td>
</tr>
<tr>
<td>One data set including all prediction variables</td>
<td>Create prediction target (at prediction horizon set)</td>
</tr>
<tr>
<td>Values &amp; Importances of variables in prediction</td>
<td>Re-smoothing.R</td>
</tr>
<tr>
<td>Test set RMSE and MAE and predicted future take-rates</td>
<td>preserved right information</td>
</tr>
<tr>
<td>Historic.R</td>
<td>Highlight generation transitions</td>
</tr>
<tr>
<td>Linear.R</td>
<td>Include business forecasts</td>
</tr>
<tr>
<td>Quadratic.R</td>
<td>Generation.R</td>
</tr>
<tr>
<td>ARIMA.R</td>
<td>ModelsOptions.R</td>
</tr>
<tr>
<td>BusinessInput.R</td>
<td>Similarity.R</td>
</tr>
<tr>
<td>MacroEconomics.R</td>
<td>Option price.R</td>
</tr>
<tr>
<td>Feature engineering</td>
<td>Grid Search.R</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>Join the predictors into one data frame</td>
</tr>
<tr>
<td>Optimization</td>
<td>Choose hyperparameters using cross-validation</td>
</tr>
</tbody>
</table>

**Analysis**

- Automotive industry
- BMW
- Take rates
- Product: BMW i8
- Price: €150,000

**Participants**

- Ana Lucia
- Gijs Mulder
- Jonathan Amar
- Dr. Steffen Illig
- Prof. Nikos Trichakis
- Andreea Georgescu
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.

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 fact 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.’

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. It can be thought of as similar to building segments based on the efficiency of the lookalike models. It can be thought of as similar to building segments based on the efficiency of the lookalike models. It can be thought of as similar to building segments based on the efficiency of the lookalike models.

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

Client Value / Insights

[m]Clusters

Two metrics of importance 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.

Lookalike Results

One metric of importance 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.

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 > 0.5), 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.

INTEGRATION & RESULTS

Client Value / Insights

[m]Clusters

Two metrics of importance 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.
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-occurring intents.

Results

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.
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 is MailChimp?

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

Datasets

<table>
<thead>
<tr>
<th>Store Id</th>
<th>Customer ID</th>
<th>Product Id</th>
<th>Order Id</th>
<th>Product Id</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>50156018</td>
<td>12300025</td>
<td>1740</td>
<td>2017-02-0</td>
<td>7023</td>
<td>Raven</td>
<td>Disco Dress with a high neck, low back, this dress was designed for you in all the right places.</td>
</tr>
<tr>
<td>50883273</td>
<td>11340025</td>
<td>1055</td>
<td>2017-05-0</td>
<td>7023</td>
<td>Disco</td>
<td>Jumper</td>
</tr>
<tr>
<td>50883273</td>
<td>14320754</td>
<td>1120</td>
<td>2017-06-26</td>
<td>7023</td>
<td>Disco</td>
<td>Jumper</td>
</tr>
</tbody>
</table>

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

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

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:

\[ \text{Normalized Mean Square Error (NMSE)} = \frac{\text{Mean} (\text{Model Predictions} - \text{Observed Ratings})}{\text{Observed Ratings}} \]

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.

Phase 1: Review customer feedback
Phase 2: Data Processing and Metrics
Phase 3: Choosing the right Algorithm: Soft-Impute
Phase 4: Results and Recommendations

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

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.

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.

\[ \min_{Z \in \mathbb{R}^{(m,n)}} \sum_{(u,i) \in \mathcal{I}} (x_{ui} - z_{ui})^2 \quad \text{subject to} \quad ||Z||_* \leq \tau \]

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.

\[ X = \sum_{i} \lambda_i U_i V_i^* \]

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.

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!
Routing Vehicles for the MBTA’s RIDE

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

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.

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

This figure shows the estimated daily total cost savings using our greedy algorithm. Savings were lower on weekends as 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.

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.

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

Left-to-right: Sarah Eade, Cáeline 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

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

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


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.

Our Recipe

Weeks 1-2

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

Weeks 3-4


Weeks 5-6

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.

Weeks 7

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

Weeks 7-8

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.

Weeks 9-10

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

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

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

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

**Rule-based dialog flow**
- Formulate a base dialog flow for a given use case
- Handcraft a specific series of rules from base dialog flows
- Bot leads conversation using preset question-based flow
- Bot classifies user responses using its handcrafted rules

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

**Demonstration Application**

Bot takes into account food type and neighborhood constraints

- A picture can be requested by the user
- Bot understands city switch and inquiries about new preferred area
- Alternatives can be asked for
- User can switch between cities within a conversation

**On-site internship**
- **June**
  - Release of Alpha version
- **July**
  - Example level generalizability
- **August**
  - Feature level generalizability

**Impact**
- **Customer acquisition**
  - Display advanced capabilities to prospective customers
  - Meet customer expectations
  - Adapt rapidly to new customer use cases

- **Vertical Examples**
  - User-friendly solutions bring about massive adoption

- **Churn reduction**
  - Act on customer preferences
  - Automate customer satisfaction analysis
  - Answer questions with high accuracy 24/7

- **Cost reduction**
  - Automate repetitive tasks
  - Allow exceptional people to focus on high-value problem solving
  - Scale up and down depending on customer requirements

- **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:
    - Gather and curate thousands of conversation transcripts
    - Build the corresponding informative database by scraping the web
    - Train the core deep learning modules

- **Algorithm**
  - Promoting research-stage architectural developments:
    - Memory Networks: RNN that selects and stores relevant dialog chunks in memory
    - Frames Tracking: adds 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

**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

**Project Timeline**

<table>
<thead>
<tr>
<th>On-campus research</th>
<th>May Implementing Bot modules</th>
</tr>
</thead>
</table>
| **February**
  - General literature review |
| **March**
  - End-to-end architectures |
| **April**
  - Building Informative DB |

**Impact**
- **Customer acquisition**
- **Churn reduction**
- **Cost reduction**

**Path Forward**
- **New Use Case**
- **Algorithm**
- **Infrastructure**
<combined text from the document>
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.

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)

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

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

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

NBA Sales 2014-2017
MLB Sales 2018.05-2018.08

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

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...

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

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)]

NBA Trading Profit: $6.7 Million/yr
MLB Trading Profit: >$20 Million/yr

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

**Modelling**

There are many challenges to our problem, chiefly: data imbalance (0.1% fires), wildfires can be random and skewness of fire size. 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.

**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, TerraNet, 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.

**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.

**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.

**Risk Map**

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.
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.

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.

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!