Starbucks DDT Recommendations Project

Anant Rajeev, Corina Geier, Christie Lee Gan, Leena Elamrawy

Project Proposal

Introduction

The objective of this project is to increase contextual relevance of product recommendation headlines to current store conditions. Not only will this increase screen engagement compared to the current baseline of static headlines but it will also result in higher recommended product conversion rates. Additionally, we will use reinforcement learning and in particular multi-arm bandits to adapt content based on customer responsiveness.

Problem Statement

Starbucks recently deployed a new product recommendation system on their drive-thru screens at 4,000 Starbucks locations. This recommendation system has been in use for the last year. Now the company would like to do a redesign of the interface and has tasked a UX team to refresh the screens. The ultimate goal of this project is to both optimize the conversion rate of Starbucks customers and to better communicate why the customer should purchase the items of interest to increase the likelihood of the product being purchased. The new design recommendations include more dynamic content to increase screen engagement and sales. The hypothesis is that if better communication is provided through headlines that show the customer why they are seeing particular products, there will be a higher revenue.

The deliverables to achieve this goal include predicting context labels for a specific store, predicting optimal headline template choices, and ensuring product recommendations align with the predicted headline. An example headline for a store might be "Sunny in Seattle" or "Afternoon Treat". Some factors that we will have to consider in this process include weather, such as "sunny" or "snowy", time attributes, and customer behaviors at a store location. With 600 products total, we will need to consider categories such as sweet vs. savory food and drinks, how big the store footprint is, the taste profiles of that store, such as a preference for hot or iced drinks, as well as the nutrition facts of the item. For our purposes the product list might be scoped down.

Background / Literature review

Our project will involve reinforcement learning and in particular multi arm bandits. In a multi-arm bandit experiment, there are multiple slot machines with different probabilities of payout with potentially different amounts. If a user were to pull each slot machine's arm an equal number of times, eventually the winning probabilities of each slot machine is determined the more times each round of arms are being pulled. This approach is particularly useful in the context of recommender systems when you don't know much about the customer or don't understand which features of a product are particularly relevant (Chumley). One drawback of the multi-armed bandit algorithm that Shenghao Xu points out in *BanditMF: Multi-Armed Bandit Based Matrix Factorization Recommender System* is that it is relatively slow because it requires extensive exploring before the system has learned enough to draw confident probabilities. Most of these algorithms are not familiar with a new user in the initial period, so the algorithm tends to explore more items to understand the user, but then the user may be confronted with a number of items that are not as relevant. The items only become relevant the more the user interacts with the recommender system. For example, as the user selects more items in their basket, the recommendations become more accurate to the user's preferences, but it takes time to gather that initial data (Xu). This inherent nature of recommendation/reinforcement learning systems can be considered both a positive feature and a drawback when it comes to predicted contextual headlines for products. It works well when new customers come through the Starbucks drive through and we have no initial data on the customer. However it takes a while for the predictions to become more accurate over time when we collect more data on the customer. We plan on using filtering and dynamic weights in the multi-armed bandit algorithm, along with any other dedicated algorithms for certain strings/key words to speed up the algorithm. This will allow us to predict headlines as best as possible given what we know.

Literature:

Chumley, Janel Roland. "Bandits for Recommender System Optimization." *Medium*, 11 Sept. 2021, towardsdatascience.com/bandits-for-recommender-system-optimization-1d702662346e.

Xu, Shenghao. "BanditMF: Multi-Armed Bandit Based Matrix Factorization Recommender System." *ArXiv:2106.10898 [Cs]*, 23 June 2021. *arxiv*, arxiv.org/abs/2106.10898. Accessed 15 Dec. 2021.

Data Exploration

- *String template data*

String Template (target ~5 words)
{weather_state} in {store_city}
{daypart} {preferred_customer_mode}

{weather_state} {preferred_customer_mode}
{daypart} in {store_city}
{weather_state} {daypart} in {store_city}

Above are the sample outline of strings that we will use as part of the recommendation system on screen in the drive-thrus.

- *Context Element Key mapped to Context Element Value with Product Discordancy Notes*

Context Element Key	Context Element Value	Product Discordancy Notes
Hour store_city	{store num mapping}	
weather_state	Sunny	No hot drinks
weather_state	Chilly	No iced drinks
weather_state	Snowy	No iced drinks
weather_state	Rainy	No iced drinks
weather_state	Pleasant	
daypart	Morning	
daypart	Lunch	
daypart	Afternoon	
daypart	Evening	
preferred_customer_mode	Light Pick-Me-Up	Calorie threshold/weight potential
preferred_customer_mode	Treat	Sugar threshold/weight potential
preferred_customer_mode	Boost	Caffeine threshold/weight potential
preferred_customber_mode	Flavor	

Above are the categories we place the individual stores into categories with regards to which products to advertise at the drive thru. For example, when the weather context is set to "sunny", we want our system to ensure that it won't recommend hot drinks. Our product discordancy notes denote that condition.

At the time of submission, we do not have access to all of the data on the VM, but we have detailed information on the features of 4 tables that will come in the format of parquet files and will be critical to our analysis and recommendations:

Table Name	**Columns**	**Grain**	**Purpose**
action reward - labels	1) **CheckNumber** 2) **StoreNumber** 3) **recommendationId** 4) **_localdate** 5) **BusinessDate** 6) **_localtime** 7) **parentSKUs_final** 8) **nonZeroCart_old**	**BusinessDate, StoreNumber, CheckNumber, productNumber, form, (hour)**	map impressions to successful conversions (boolean)

store - labels	1) **STORE_NUM** 2) **UrbanityType** 3) **CountryCode** 4) **PLANNED_SQ_FEET** 5) **hasDriveThru** 6) **hasMOP** 7) **hasWifi** 8) **hasClover** 9) **hasMercato** 10) **hasNitro**	**BusinessDate, StoreNumber**	store-level features
weather - labels	1) **StoreNumber** 2) **HourInDay** 3) **TempAvgDeseas** 4) **TempAvgRatio** 5) **ExtInd_TempAvgDeseasM05** 6) **ExtInd_TempAvgDeseasM95** 7) **ExtInd_RainSumM95** 8) **ExtInd_SnowSumM95**	**BusinessDate, HourInDay, StoreNumber**	store-hour-level deseasonalized/ normalized weather features
product - labels	1) **productNumber** 2) **productType** 3) **config_category** 4) **form_codes** 5) **prod_num_name** 6) **str_ingredients**	**productNumbe r**	map impressions to successful conversions (boolean)

Work-to-date / Data Pipeline

Our data will be in the form of parquet files where each parquet file is representative of a specific day. The benefits of using parquet files is that the data types are specified and hardcoded in the columns of data, is the standard for high-load cloud computing systems, and easy to distribute especially when using multiple clusters. Additionally, Starbucks has provided string templates to use in order to construct the contextual headlines, such as "Sunny in Seattle". The templates have four keys with different potential values as follows:

- Preferred_customer_mode: *flavor, boost, treat, light pick me up*
- Daypart: *morning, lunch, afternoon, evening*
- Weather_state: *sunny, chilly, snowy, rainy, pleasant*
- Store_city: *store number mapping*

The data will comprise three feature vectors - Product, Store, and Weather. Additionally, there will be one label set. Each row in a dataframe represents one hour, with all transactions made in that hour combined in a single row.

The data will reside on a virtual machine on the UW server. An Azure instance was created for our data storage and computing needs. We will use Python to access the data, utilizing pandas (pandas.read_parquet), pyarrow and numpy to extract information from the parquet files. The crux of our data cleaning will be to change the parquet files into csv files.

Proposed Solutions

Final Deliverables/System Requirements are:

1) Ability to predict context labels to a requesting store
2) Ability to predict optimal headline template choice (strategy to address cold start here will be key)
3) Ensuring product recommendation aligns with with chosen headline

Code-stack of combining all three of these ^ how we will turn these 3 abilities into a system (2 sets of predictions/mapping)

Some of the challenging aspects of this project that we are looking forward to tackle are:

1. Understanding and implementing Reinforcement Learning including Multi-Armed Bandits to predict headlines and optionally product recommendations.
2. Associating dynamic headlines with specific stores based on the store conditions.
3. Ensuring that the dynamic headlines we create align with the recommended products. For example, not recommending breakfast items with a headline such as "Afternoon Treat". We will evaluate the use of filters, dynamic weights, and adjunct algorithms.
4. Predicting accurate context labels for a given headline template.
5. Considering whether the headline should come first and then matching the products to this headline, or if the products should come first and then an appropriate headline should be chosen to match these.
6. Testing our solution, such as through A/B testing, to see how effective it is and how well it supports the initial hypothesis.

Any potential A/B test would be at a time after submission but the POC presentation will happen in March.

Risks & Benefits of proposed solution

A benefit of using multi armed bandit tests is that you are able to use information gleaned from the exploratory period for exploitation. In this way you are able to take advantage of what has already been learned from the model without having to wait a long time to understand what the best choice is. This is especially important as it is costly to collect data. Additionally, multi-armed bandit tests are important as they allow for changing conditions and changes in customer behaviour and preferences by constantly considering the most effective options. One risk of using multi armed bandit testing is that it can take a while to reach significance and they can be affected by biases in the system inputs.

Implementation Updates

Our sponsors on this project reinforced the idea that the individual stores would determine the products that are on-screen. Our implementation would then include 2 main things:

1. A filtering mechanism to make sure that the headlines displayed on the drive-thru screen actually matched the products that were being displayed
2. A multi-armed bandit model that recursively learns, based on the customer conversion rate, about which headline is the most successful to display based on the products that they recommend. This model will continue to update based on the data that is gathered.

Schedule

All team members are responsible for each task:

Task	Due Date	Dependencies	Actual Date of Completion
Problem Statement (Class)	November 11	N/A	November 11
Team Bios Page (Class)	November 17	N/A	November 17
Non-Disclosure Agreements	November 10	Starbucks Law Department	November 25
Data Access	November 24	NDAs	November 24
Data Pipeline (Class)	November 24	Data Access	Waiting
Data Exploration (Class)	December 1	Data Access	Waiting
Access Data Package	December 10	Production Server Access	Waiting
Proposal Submission (Class)	December 15	Data Access	December 15

AFTER DATA INGESTING:

Join all parquet files	February 1		February 1
Data Preprocessing, Analysis, Modeling	February 10		February 15

Multi-Armed Bandit Implementation	March 1		March 3
Categorize products for weather/time of day	March 1		March 1
Finish Flyer	March 8		March 7
Finish Poster	March 1		March 6
Finish Writeup	March 10		March 14
Measuring Success/Testing	*Starbucks*		*Starbucks*

Meet The Team

Leena Elamrawy

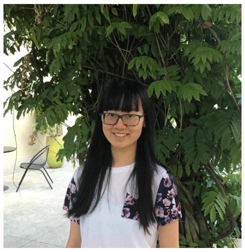
Leena graduated with a Bachelor's in Data Science and a minor in Economics from UC San Diego. She gained work experience in the Data Science field through her internships in Fintech and Tech companies. She's experienced in workflow automation in Python, database management in PostgreSQL, personalizing web experiences using BI tools, data visualization and natural language processing. Leena will be the team's coordinator.

Corina Geier

Corina graduated with a Bachelor's in Math from the University of Washington. She has gained job experience in technical project management and data analytics roles. She's experienced in SQL, Python, and R for data analysis and machine learning, and Power BI tools for data visualization and business process automation. Corina will be the team's project manager and note taker.

Anant Rajeev

Anant graduated from the University of Washington with a B.S. in Informatics concentrating in Data Science and minoring in Mathematics. Over the years, he has had experience interning in Fintech and Biotech companies working in data science, analytics, and project management roles. He is very well versed in Python, R, and SQL doing statistical analysis, machine learning and predictive modeling, and data visualization. Anant will be working as the team's facilitator.



Christie Gan

Christie graduated from the University of Washington with a B.S. in Informatics, concentrating in Data Science. She has experience in data science and product management within the aerospace industry. Her skills include Java, Python, R, SQL, and well as data visualizations, process automation, and predictive analytics. Christie will be the team's contact focal.

Emily Yamauchi

Emily is a data scientist currently working at Alaska Airlines. Prior to this position, she was at a mobility tech company working with traffic and geodata, and helped launch its service expansion into Japan. Prior to starting the MS in Data Science program at the University of Washington, she worked as an analyst at several Japanese public equity funds. Her skills include python, R, Java, and is passionate about using data and data insight in ways that can benefit society as a whole.