Andre Warsaw

Prof. Lori Perine

Data 205

CRN # 21844

Alcohol and Beverage Services Buying Patterns

Abstract:

The purpose of this project was to assist the Alcohol and Beverage Services (or ABS for short) of Montgomery County with determining a better methodology for observing the buying patterns between all 27 locations with a final goal of improving quarterly financial performance. As of currently, ABS is observing their buying patterns by two metrics, that being an individual store's size and financial performance. I aimed to optimize this methodology by first observing inventory performance both locally (via every individual store) and globally (all 27 locations collectively). Upon observing the inventory performance, I was able to conclude that there were no notable differences on a local level, thus deciding to solely observe inventory performance on a global level. Using variable engineering I constructed a new variable named "priority" which was used to categorize inventory performance into three separate tiers. Finally, I used machine learning modeling to create a predictive model which assists in optimizing inventory management, focusing on placing inventory into all stores that will perform well. My final product, an inventory management system, produces a csv file with recommended inventory to stock in each store along with metrics to help keep track of quotas with the purpose with saving money on inventory fulfillment, and thus improving ABS' quarterly financial performance.

Introduction:

Alcohol and Beverage Services of Montgomery County has tasked me with optimizing their buying patterns methodology, which currently consists of store size and store tiers (store performance). The chart below will help give a breakdown of how each category were defined.

Store Size Identifier	Definition	Store Tier Identifier	Definition
A	Stores > 6000 sq ft	1	Stores performing > $8 million in the SY
B	Stores between 4000 and 6000 sq ft	2	Stores performing between $5 million and $8 million in the SY
C	Stores < 4000 sq ft	3	Stores performing < $5 million in the SY

There is a total of 27 locations observed in Montgomery County, MD in which I will include a legend identifying the stores by their location and store number. To conduct this research, I will be utilizing R Studio as my sole source of all analysis and modeling of the transaction data set provided by ABS. My main goals are to do the following:

- Use ABS' current methodology to observe inventory performance
- Determine a better methodology using machine learning modeling

Data Set Breakdown and Preparation:

There are a total of 18 variables and roughly 12 million observations spanning from 2023 to 2025 in the transactions data set provided by ABS, which I will give a breakdown of below. However, there will also be additional variables that I will also include for this data set and will be using throughout this project which will be mentioned below as well.

Below are the variables originally included in the data set:

Variable	Description	Variable	Description
transdate	This is the physical date that the transaction took place. ABS has provided transactions starting from July 1, 2023 to June 30, 2025 (roughly 2 years worth of data)	description	This is the product description
transactionid	This is a unique identifier for each individual transaction (likely to not be used for this project)	netamount	This is the total cost of the line item / product purchased (discounts included) without tax
store	This is the store number where the transaction took place	lineqty	This is quantity sold of each individual line item in a transaction. So if 3 bottles of wine were sold, this would be 3. If 2 6-packs were sold, this would be 2
storename	This is the formatted store name where the transaction took place	packunit	This is the selling unit for the line item. This is the unit as how the product was sold. This would either be Btl (single bottle) or 4pk, 6pk, etc. Some lines may only show 04 or 06 and this relates to 4pk or 6pk. This is more to show the calculation of the totalqty column, which represents how many actual bottles were sold to a customer
itemid	This is the unique identifier for products. Item dimension attributes do not change. If an item's pack unit or bottles per case changes a new code is created.	totalqty	This is the total quantity of items sold for that line item. (Line quantity * pack unit)

itemtag	This is the warehouse designation inventory tag. Some items may be NULL as tags are updated on a daily basis and at the time this data set was taken offline, it may have converted some TAGS to NULL. (note that I will only be observing itemtags that identify as "ST", which is Standard Stock)	classificationcategory	This is the group category the product belongs too (Level 3). This further references the category within the type of items. i.e.: LIQ > WHISKEY > IRISH WHISKEY.
tagdesc	This is the formatted description of the warehouse inventory tags	bottlespercase	This is the total number of bottles in a case. This is important for our warehouse partners when calculating overall case volume of an item. totalqty / bottlespercase = Case(s) sold
classificationdepartment	This is the department the product belongs too (Level 1). The only departments are BEER, WINE, LIQ, MISC.	size	This is the size of the product. There are standard sizes of many products and often one size of a product will outperform another for a variety of reasons. i.e. 750 mL vs 1.75 L
classificationtype	This is the group type the product belongs too (Level 2). This references the type of product within the department. i.e.: LIQ > WHISKEY	custaccount	This designates if the transaction was for a retail/public customer or a licensee (This will only be used to filter out all licensee transactions as I will only be observing public customers)

Alongside the original variables, these are the additional variables I have added into the Data Set prior to doing any further analysis:

Variable	Description	Variable	Description
transdate2	This is used as a separately formatted date variable for visualization purposes upon cleaning the data set		

year	Another created variable used for observational purposes to organize the data set specifically by year	cost	raw cost of a given transaction (not including tax) based on netamount / lineqty
storesizecat	This is added for observing the data by the store size categorization created by ABS, only "A", "B", or "C" will populate the rows corresponding to its respective store in each transaction	storetiercat	This is added for observing the data by the stores financial performance based on ABS' parameters, only showing "1", "2", or "3" in the rows corresponding to its respective store in each transaction

As mentioned previously I have used R Studio exclusively for my research, utilizing a total of 5 libraries for all functions needed including:

- tidyverse – for data cleaning, wrangling, and EDA
- DBI – for database setup for SQL queries
- RSQLite – for SQL query functions
- nnet – for my first multinomial model observations
- caret – for all machine learning modeling and analysis

Before starting all observations I started with loading in the transaction data set, followed by fixing all column variables to lowercase for simplicity, filtering all licensee transactions to prevent my observations from being skewed by outliers (as advised by the ABS team), and also removing all stock that were not considered standard as that would make observing the inventory very hard to do as the ABS team has noted that inventory frequently gets converted from one item tag, such as special, to another on a daily basis depending on the scenario and when doing so it also changes the itemid even if it is the exact same item. After cleaning all variables, I then added in the additional variables as noted in the table above, followed by filtering out all transactions that were voided and refunded as that would also negatively impact my research. After this I was ready to begin all observations planned.

<u>Statistical Methods and Observations:</u>

To observe ABS' current buying patterns methodology, I decided to first observe their classification variables, that being classification department, classification type, and classification category (Please refer to chart for further explanations on said variables). The goal was to see if there were any variation in inventory performance from a local level (store-to-store) vs. a global level (all 27 stores, collectively). This was done by first observing the inventory performance by classification department of all 27 stores collectively. Based on the observation (see chart A) liquor easily performed the best followed by wine, beer, and miscellaneous. Afterwards I decided to observe the top two performing classification departments, liquor and wine, by classification type to determine if there are any notable variation in inventory performance between all 27 individual stores followed by all stores collectively. With one exception, that being wine sales in one specific store, there were little to no variation in the item performance distribution between all 27 locations collectively and individually. To confirm this, I used SQL queries to observe inventory performance year by year by store tier's and it further confirmed that there are little to no variation in inventory performance as items that performed well in one store, performed well across all stores and the same for items that perform poorly in the given year.

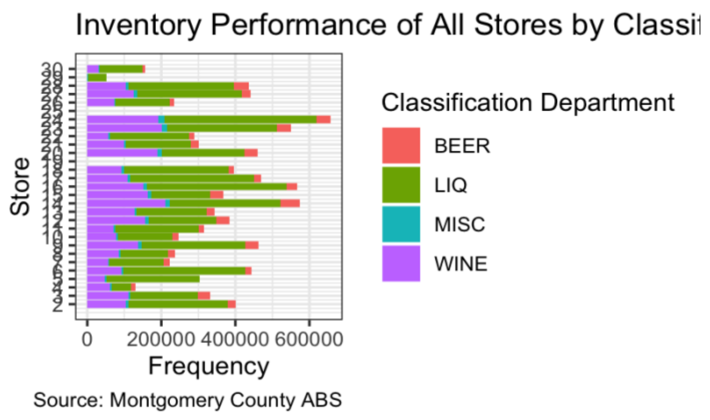


Chart A:

Inventory Performance by Classification Dept.

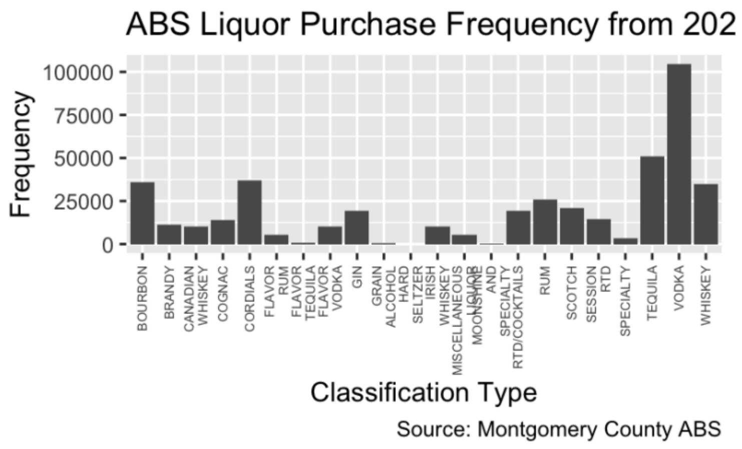
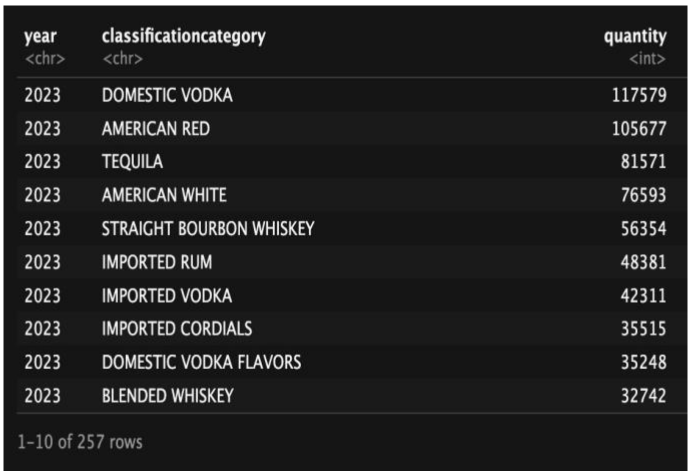


Chart B:

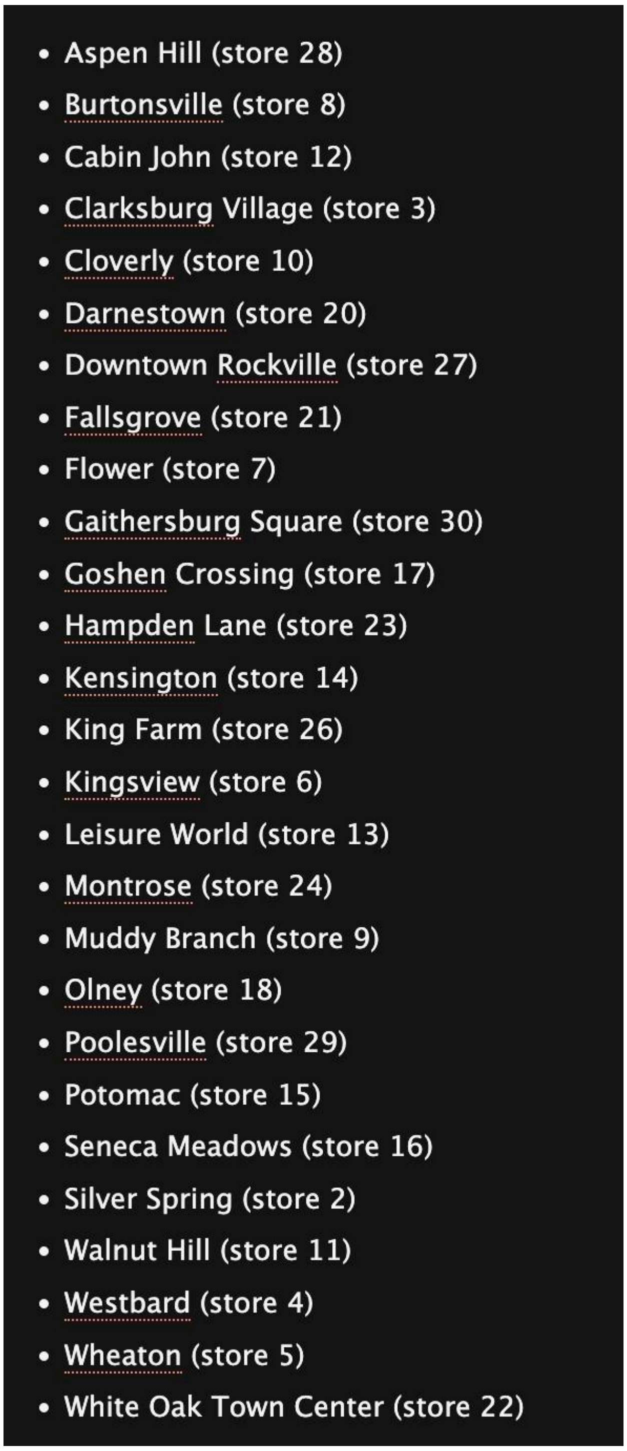
Inventory Performance by Classification Type – Liquor (All stores collectively)



```
year   classificationcategory        quantity
<chr>  <chr>                            <int>
2023   DOMESTIC VODKA                  117579
2023   AMERICAN RED                    105677
2023   TEQUILA                          81571
2023   AMERICAN WHITE                   76593
2023   STRAIGHT BOURBON WHISKEY         56354
2023   IMPORTED RUM                     48381
2023   IMPORTED VODKA                   42311
2023   IMPORTED CORDIALS                35515
2023   DOMESTIC VODKA FLAVORS           35248
2023   BLENDED WHISKEY                  32742
1-10 of 257 rows
```

Chart C:

Sample SQL query output showing performance of inventory by classification category and quantity



- Aspen Hill (store 28)
- Burtonsville (store 8)
- Cabin John (store 12)
- Clarksburg Village (store 3)
- Cloverly (store 10)
- Darnestown (store 20)
- Downtown Rockville (store 27)
- Fallsgrove (store 21)
- Flower (store 7)
- Gaithersburg Square (store 30)
- Goshen Crossing (store 17)
- Hampden Lane (store 23)
- Kensington (store 14)
- King Farm (store 26)
- Kingsview (store 6)
- Leisure World (store 13)
- Montrose (store 24)
- Muddy Branch (store 9)
- Olney (store 18)
- Poolesville (store 29)
- Potomac (store 15)
- Seneca Meadows (store 16)
- Silver Spring (store 2)
- Walnut Hill (store 11)
- Westbard (store 4)
- Wheaton (store 5)
- White Oak Town Center (store 22)

Master List of all Stores for reference along with store numbers

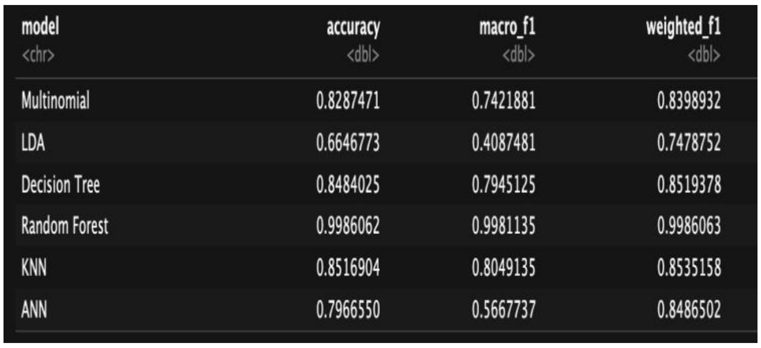
Variable Engineering and Predictive Modeling:

Upon confirming that inventory performance has little to no variation between all 27 locations, it was safe to confirm that I could observe the inventory on a global level, and that ABS is able to optimize their buying patterns methodology by observing inventory performance. To assist with observing inventory performance, I engineered a new variable named "priority" which categorizes inventory into three separate columns—that being high, medium, and low priority. This was done by grouping all inventory globally followed by separating them using quartile functions so that the top 25% performing items are categorized as high priority, middle 50% as medium priority, and the lowest 25% items as low priority. The goal with creating this variable was to use machine learning to help with creating a predictive model which could accurately predict the future priority of an item based on the previous year's performance. To do so I will be using the following formula for predicting the future priority of an item:

next_priority ~ total_netamount + total_lineqty + avg_cost + n_transactions + storetiercat + storesizecat + year

Where "total_netamount" is the sum amount a product has achieved in a year, "total_lineqty" is the frequency a product has been purchased, "avg_cost" is the mean cost overall of the product, "n_transactions" is the number of transactions the product shown up in, and the remaining variables are explained in the variable charts above. I tested this formula on six different machine learning models along to observe their statistics and determine the best model moving forward. The models I tested on are listed below along with the confusion matrix which showcases their performance:

- Multinomial
- Linear Discriminate Analysis
- Decision Tree
- Random Forest
- K Nearest Neighbors
- Artificial Neural Networks



model <chr>	accuracy <dbl>	macro_f1 <dbl>	weighted_f1 <dbl>
Multinomial	0.8287471	0.7421881	0.8398932
LDA	0.6646773	0.4087481	0.7478752
Decision Tree	0.8484025	0.7945125	0.8519378
Random Forest	0.9986062	0.9981135	0.9986063
KNN	0.8516904	0.8049135	0.8535158
ANN	0.7966550	0.5667737	0.8486502

Chart D:
Confusion Matrix showing the final statistics of all 6 models

As seen in the confusion matrix above (see Chart D), the random forest model was by far the best choice for a predictive model for ABS' business problem as it has a high accuracy score of 99.86% and an F1 score of 0.998 out of 1.

Final Product:

Now that I have confirmed the predictive model that best predicts the priority of an item, my final step was to produce an inventory manager which uses predictive modeling to produce a csv file that recommends what items should be stocked in what store based on priority. To do so, I first had to receive as close to accurate storefront SKU counts for each store from the ABS team to know how many items needed to be fulfilled per store. I also needed to determine parameters for how the stores needed to be stocked based on priority. To do so I chose to organize the stores by store tier as it is a convenient grouping method alongside it offering an opportunity to make a logical quota for all stores. When deciding the priority mix for all three tiers, I based it on the following logic:

- Tier 3 stores should not hold any merchandise that could be considered risky, prioritizing only merchandise that regularly perform either very well or generally well
- Tier 2 stores should focus mainly on holding mostly high and medium priority merchandise, only considering low priority in these categories if the store has a high quantity of SKUs available
- Tier 1 stores have more flexibility, giving more option to experiment with new inventory (which would theoretically start at low priority) along with keeping available select top performing low priority items if additional storage is available at a given store

I have included a chart below (see Chart E) which summarises how the predictive model will recommend inventory fulfilment for each store. When creating the inventory manager, I have tasked it to include store information (such as store name, the store tier, and the SKU count of said store), item information (such as item ID number and priority label), predictive information (such as recommended number of priority per store, fill rates and percentages per priority), and quotas (including the pass/fail quota for low, medium, and high priority items per store, and an indicator for if the store as a whole passed its' quota). See below (see Chart F) for a final sample of the output csv file the inventory manager would provide when given an updated data set to observe.

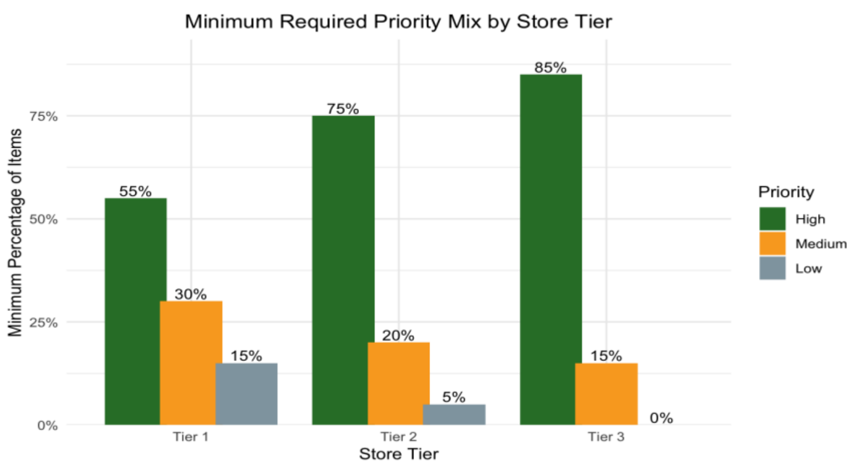
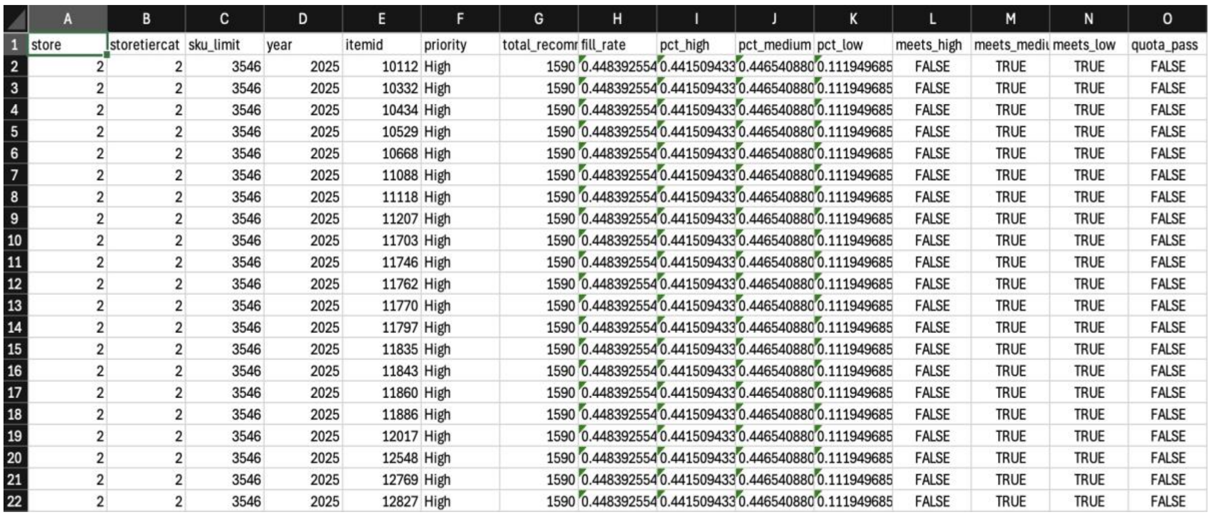


Chart E:
Priority Mix Parameters for Inventory Manager based on Store Tier



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	store	storetiercat	sku_limit	year	itemid	priority	total_recomr	fill_rate	pct_high	pct_medium	pct_low	meets_high	meets_mediu	meets_low	quota_pass
2	2	2	3546	2025	10112	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
3	2	2	3546	2025	10332	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
4	2	2	3546	2025	10434	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
5	2	2	3546	2025	10529	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
6	2	2	3546	2025	10668	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
7	2	2	3546	2025	11088	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
8	2	2	3546	2025	11118	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
9	2	2	3546	2025	11207	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
10	2	2	3546	2025	11703	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
11	2	2	3546	2025	11746	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
12	2	2	3546	2025	11762	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
13	2	2	3546	2025	11770	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
14	2	2	3546	2025	11797	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
15	2	2	3546	2025	11835	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
16	2	2	3546	2025	11843	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
17	2	2	3546	2025	11860	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
18	2	2	3546	2025	11886	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
19	2	2	3546	2025	12017	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
20	2	2	3546	2025	12548	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
21	2	2	3546	2025	12769	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE
22	2	2	3546	2025	12827	High	1590	0.448392554	0.441509433	0.446540880	0.111949685	FALSE	TRUE	TRUE	FALSE

Chart F:
Sample output of csv file from inventory manager

Final Remarks:

As I reached the conclusion of my project, there are a few noticeable concerns with my inventory manager that I would love to update if given more time. When observing Chart F, you can see that none of the stores have passed their quota. This is purely due to all stores failing their high priority quota as the parameter I have set for high priority is unrealistic as the top 25% of items globally does not contain enough items to fulfill said quota. I also noticed that the final product could be improved in other ways as well, to help with accuracy and further solidify its' usefulness as a utility. I would resolve these issues by doing the following:

- Recalibrate the high priority variable and quotas I have created for all store tiers to a more realistic amount
    - High Priority being top 33%, top 40% or potentially getting the model to assist with determining the parameters
- Include quantity counts for each recommended item instead of only mentioning items that should be stocked
- Adjust the model to observe data on a shorter cycle that is both more accurate and beneficial to ABS team, including either a monthly cycle or a timeframe ABS currently performs inventory fulfillment (if not monthly)

By making these changes, I am certain that the product I have provided to the ABS team would be assist with saving money by optimizing inventory fulfilment and thus improving financial performance overall.

References and Acknowledgments:

I would like to thank the following people for their assistance throughout this process:

- ABS team for keeping me on track and assisting me every week with understanding the data set they have provided and stopping me from going the wrong direction with my research whenever it was needed
- Data Montgomery team for helping with organizing meetings, and all communication and helping with finding supplementary data even if we never ended up using it
- My capstone professor, Prof. Lori Perine for advising me on how to move forward with my machine learning models and suggesting the R studio library caret to me
- Finally, my fellow Arash Nateghian as his research on the same business problem also helped sharpen my approach