*Predictable Purchases Second Draft*

Jordan Deagan

Computer Science

New College of Florida

Sarasota, US

Jordan.deagan17@ncf.edu

***Abstract*—This is the second draft of my final Report in Data Mining. It contains a partially complete version of the final report I will turn in. The Introduction contains an overview of my project, Data has my source, data tables, and the preprocessing I used, Methodology has the algorithms I use and what it does, Results will show my output and what it means, Conclusions covers the meaning of my results and the afterthoughts of my process, and Future Work goes over what I might include there depending on what happens.**

1. INTRODUCTION
   1. *Question*

The initially idea for this project was a simple association check for commonly bought items with beef jerky, trail mix, and protein bars. This has been changed to become a predictive model for purchasing certain products. I have gone from associations with Apriori to develop predictions to just using basic behaviors and if a purchase contains the item to develop a Markov Chain. When given the time and day a person makes a purchase, as well as that person’s previous purchases, can it be predicted that a specific item, or type of item, will be in their purchase.

* 1. *Reasoning*

I am asking this question because I want to find if human behavior is predictable when it comes to buying groceries. This question is very important to online stores as a predictive software would allow them to guess when someone is most likely to buy something, giving them a way to know what products to focus advertising and recommendations for, and to who they will work best on, to maximize profit.

* 1. *Overview*

Once I found my data, I preprocessed the data in order to be able to run it in less time, this meant shortening the inputs I wanted, and preorganizing and splitting all of the data as needed so I only had a few smaller tables to do less on later.

Once I am done preprocessing, I will use a premade algorithm for Markov Chains to develop a model for buying specific items. I will create two models for each product, one trained on all data, and one trained only on those who bought the product before. I will compare the results of each individually and the combined result to see which is most accurate.

1. DATA

The data I am using comes from Instacart’s open source purchases information. The data originally came as 6 data tables. Through preprocessing, I created 4 additional tables.

* 1. *Data Tables*
     1. *Store Information*

The first data tables are on the departments, aisles, and products in the store. The department data table contains the department titles and IDs. The aisle data table has the same thing for every aisle. The products data table provides the ID and name of every product, but also includes the aisle and department the product is found in. These tables were used in preprocessing to develop the lists of products from the appropriate locations. For example, Beef Jerky Dog Treats is not a product wanted in the list of Beef Jerky products, and it is not on the same aisle as other actual beef jerky products, so I do not include that aisle. When grabbing beef jerky IDs.

* + 1. *User Order*

Purchases organized by user originally came in one table containing purchase IDs, the user ID who purchased it, if the purchase is in the prior, test, or training purchase data table, the number purchase it was of this user, the day of the week it was made, the time of day it was made, and, if the purchase was made second or later, the amount of days since the last purchase by this user is recorded.

USER ORDERS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| order id | user id | eval set | order number | order dow | order hour of day | days since prior |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Fig. 1: sample of user order data table

To separate the data for training and testing, I split the purchases into the most recent purchases and all prior purchases. Due to lack of access to the test purchase data table referenced in the appropriate column, and those purchases always being the most recent purchase of that user, I would look ahead while splitting and add the current purchase to my most recent table if the next purchase was labeled test. For all other users, the purchase labeled train was the most recent purchase, and thus was used for pulling the appropriate purchase away. While I am not finished with all preprocessing due to files being bigger than expected and computers being slower than needed, I also plan on adding a value to each purchase to indicate if that purchase has the associated product in it.

* + 1. *Order purchases*

The last two tables that came with the data were of the products in each order. The tables are for the prior and train purchases. They record the purchase ID, a product ID of a product in the purchase, the order in which that product was added to the purchase, and if that product had been purchased by the user before.

ORDER PRODUCTS

|  |  |  |  |
| --- | --- | --- | --- |
| order id | product id | add to cart order | reordered |
| 1 | 49302 | 1 | 1 |
| 1 | 11109 | 2 | 1 |
| 1 | 10246 | 3 | 0 |
| 1 | 49683 | 4 | 0 |
| 1 | 43633 | 5 | 1 |
| 1 | 13176 | 6 | 0 |
| 1 | 47209 | 7 | 0 |
| 1 | 22035 | 8 | 1 |

Fig. 2: sample of order product data table

I reorganized these two tables into new files to accommodate the changes made in the user order tables, thereby shifting many purchases from prior to train, now dubbed test.

* 1. *Preprocessing*

For preprocessing, I created a list of the product IDs for all appropriate instances of beef jerky, trail mix, and protein bars. Searching through prior purchase products, I created a dictionary with purchase ID as the key and a list of product IDs as the value. On the prior user order table, I reduced the amount of individual variables in time of day by creating ranges of time to On the prior user order data table, I created the training data from if the purchase has an instance of a product in it based on the dictionary, the day of the week, time of day, and how long it has been since they last bought something.

Due to not receiving the test data table from Instacart, my data is slightly skewed.

1. METHODOLOGY

I will use K-folds cross validation to train my data. K-folds cross validation is where the data that is being used do develop a model is split into k equal parts, then the training is run k times, using all but on of the parts to develop the model and saving it. After all k runs, the k models are compared to create a final model. Using this, I will run the prior orders data sets, with purchase of desired product, day of the week, time of day, and days since last purchase, through a Markov Chain to create the weights. A Markov Chain is a predictive model used for finding patterns in randomly changing information by focusing on the current state of the data and ignoring the past.

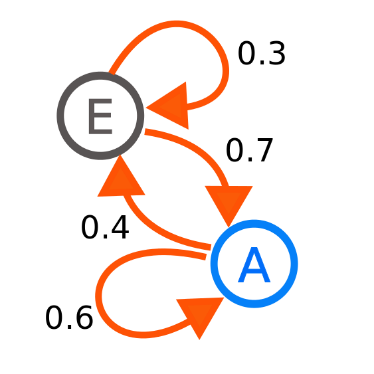


Fig. 3: sample Markov Chain

I will test this model by using the last purchase in the prior user orders to predict the probability of the user purchasing the product in the test user orders table. I will have found if these test purchases contain the product as well, since the model does not look at them, and that will make it easy to check the results. I will keep track of true positives, false positives, false negatives, and true negatives to determine how accurate the model is both during training and testing.

Once I have a definite algorithm I will go over what the algorithm does, why I chose it, and where I got it from.

1. RESULTS

As I have not finished fully with my preprocessing or began training my Markov Model, I have no results to provide. Once I have results, they will consist of the weights and accuracy of the Markov Model, a picture of the connections, and a quick analysis of how the results came about.

BEEF JERKY RESULTS

|  |  |  |
| --- | --- | --- |
|  | Purchase Beef Jerky | Not Purchase Beef Jerky |
| Purchase Beef Jerky | probability | probability |
| Not Purchase Beef Jerky | probability | probability |

Fig. 4: probabilities of purchasing beef jerky

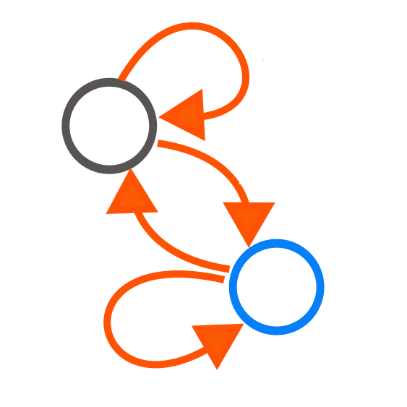


Fig. 5. Visualization of chain

TRAIL MIX RESULTS

|  |  |  |
| --- | --- | --- |
|  | Purchase Trail Mix | Not Purchase Trail Mix |
| Purchase Trail Mix | probability | probability |
| Not Purchase Trail Mix | probability | probability |

Fig. 6: probabilities of purchasing trail mix

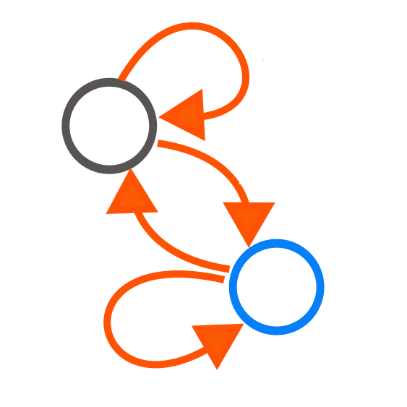


Fig. 7. Visualization of chain

PROTEIN BAR RESULTS

|  |  |  |
| --- | --- | --- |
|  | Purchase Protein Bars | Not Purchase Protein Bars |
| Purchase Protein Bars | probability | probability |
| Not Purchase Protein Bars | probability | probability |

Fig. 8: probabilities of purchasing protein bars

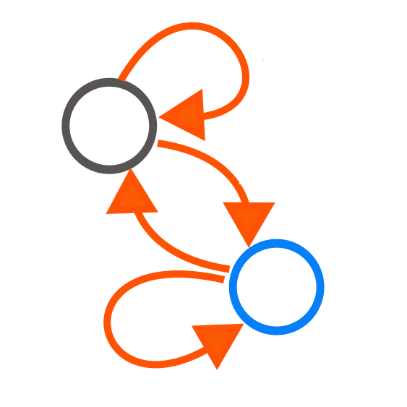


Fig. 9. Visualization of chain

For Checking accuracy, I am looking at not only the predictions being purchase above 0.5 and not purchase being below 0.5, but between 1.0 and 0.75 being a strong positive, between 0.75 and 0.5 being a weak positive, 0.5 to 0.25 being a weak negative, and from 0.25 to 0.0 being a strong negative.

I will probably have three times the amount shown, as I will create three models for each product and will need to show the results for each, whether it is in separate tables or next to each other in a single table, the size will drastically increase.

BEEF JERKY ACCURACY

|  |  |  |
| --- | --- | --- |
|  | Purchase | Not Purchase |
| Purchase Probability p>75 | Strong True Positive | Strong True Negative |
| Purchase Probability 75>p>50 | Weak True Positive | Weak True Negative |
| Purchase Probability 50>p>25 | Weak False Negative | Weak False Positive |
| Purchase Probability 25>p | Strong False Negative | Strong False Positive |

Fig. 10. The accuracy of the Markov Chain

TRAIL MIX ACCURACY

|  |  |  |
| --- | --- | --- |
|  | Purchase | Not Purchase |
| Purchase Probability p>75 | Strong True Positive | Strong True Negative |
| Purchase Probability 75>p>50 | Weak True Positive | Weak True Negative |
| Purchase Probability 50>p>25 | Weak False Negative | Weak False Positive |
| Purchase Probability 25>p | Strong False Negative | Strong False Positive |

Fig. 11. The accuracy of the Markov Chain

PROTEIN BAR ACCURACY

|  |  |  |
| --- | --- | --- |
|  | Purchase | Not Purchase |
| Purchase Probability p>75 | Strong True Positive | Strong True Negative |
| Purchase Probability 75>p>50 | Weak True Positive | Weak True Negative |
| Purchase Probability 50>p>25 | Weak False Negative | Weak False Positive |
| Purchase Probability 25>p | Strong False Negative | Strong False Positive |

Fig. 12. The accuracy of the Markov Chain

1. CONCLUSIONS

I either needed to work with a better computer from the beginning, found a smaller data set, or managed to reduce the size early. My inability to get preprocessing done early caused a lot of my intended work to not get done, but a workaround was able to be reached with time left to develop an end product. After I get results, I will go over what my probabilities mean, how I could have altered the process to get better results, what other algorithms might have done differently, and determine if my project was a success or not.

1. FUTURE WORKS

As far as dealing with this problem in the future, if I am successful this time, my main concern will be making it more efficient, accurate, or able to deal with more specific or undersampled products. The prospect of also creating the models using a few different algorithms and comparing those results to see which algorithms work better might also be a possible idea for future work. If I fail in making a decent predictor, my future work will mostly involve figuring out what part of my process went wrong and fixing it, as well as seeing if others have made similar models and compare methodologies.

REFERENCES

[1] “The Instacart Online Grocery Shopping Dataset 2017”, Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on 10/26/18