12/1/2020

Data and web mining
(HDSDASEPOL_YR2)

Data Mining CA1

Marion Bonnard
X19161964

Part1

Tidy Data

Preparing a dataset is the first and most critical step before completing any analysis. As per Dasu and Johnson, 80% of the task is used to clean and prepare a dataset, and it represents 80% of the value of the analysis. (Dasu T, 2003) We can interpret this last figure by saying that, if a dataset is not well prepared, the result might not be accurate.

To make sure that our dataset is ready for analysis, we are applying in the below report the principles taught by Hadley Wickham. The first step is to understand how the dataset is composed in order to determine the "messy" part and the irregularities.

Dataset description

The original Groceries dataset contains 1499 observations and 35 variables which displays a list of grocery items. The columns are labelled from V1 to V35 and the items are sparse across the dataset and displayed as a list. The first column contains dates which are stuck with item (strings) and the last column (V35) contains only N/A values. We can see that not all cells contain values because each transaction has various number of items.

Tidying Methodology

To following methodology is also described on the separate code "Rscript_CA1_Part1_Groceries_Marion_Bonnard"; each below step number corresponds to the steps in the code.

1. The first column (V1) has two values per cell; we can see a date and an item stuck together with no space. As per Wickham's third principle: "Every cell is a single value" (Wickham, s.d.) therefore we must separate the 2 values and create an additional column.
2. The last column V35 contains only NA values, and since "Every row is an observation" (Wickham, s.d.) we will delete the entire column that contains no value.
3. It is also advised to identify each observation with a unique ID or primary key. By creating a column "Transaction ID", each transaction is uniquely identified, and it can be used to connect tables in a normalization step.
4. Since the "Items" variables are spread across the table, it means that the first principle "Every column is a variable" is not respected. To get the variables into columns, we must use the melt function from the Reshape2 Library; this way each Item will have a matching "Transaction ID" and "Transaction date". Following this step, the new named dataset "Tidy_Groceries", contains 50966 observations and 4 variables ("Transaction ID", "Transaction dates", "variable", "value").
5. Our new dataset "Tidy_Groceries" is almost ready, however we still need to remove the rows of the cells that do not contain any value in column 4, to respect Wickham's third principle. After this step, the dataset ("Tidy_Groceries2") contains now 29063 observations and 4 variables.
6. Finally, to get a proper dataset with a minimum of noise, we can remove the column "variable" that will not be in any use for the analytic part. The final and tidied version of the dataset is called "Tidy_Groceries2" and is saved under "Tidy_Groceries_Dataset.csv"; it will be used to complete the Association Rules.

Association Rules

Applying Association Rules to Market Basket analysis, is a technique to find interesting relationships between items and predict a purchase pattern. The goal is to determine if there is a pattern when buying certain item that could lead to the purchase of an associated object in a set. To find Associations Rules, we are using the "Apriori" algorithm; it uses parameters such as support, confidence, and lift as a result. The support is the frequency or the popularity of an item in the total transactions. The confidence of X→Y is the "likelihood of item Y being purchase when items X has been purchased" (Augmented Startups, 2017). The lift is the chance to purchase an item Y (relatively to its purchase rate), knowing that another item has been bought; it is part of the result of an Association Rule and measures the performance of the rules.

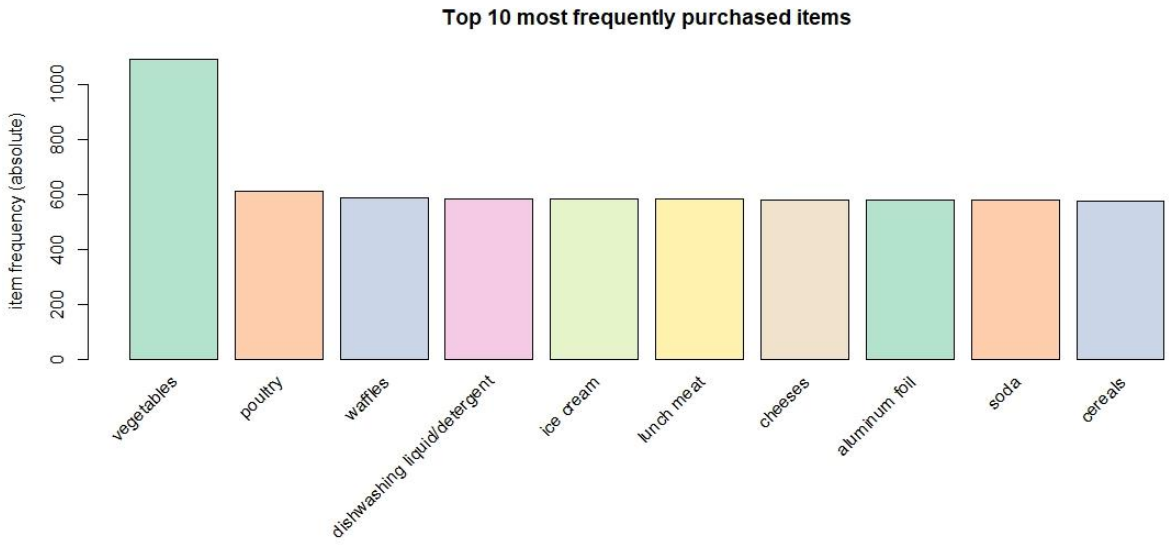
Preparation

Prior to starting the analysis, we need to transform the dataset "Tidy_Groceries_Dataset.csv" from the Tidy Data part to make it readable as "transaction". First, we are pivoting the items to integrate them all in rows per transaction ID. It can be considered as the opposite of melting. Then, we remove the columns Transaction ID and Transaction date to get a list of transactions. Finally, we rename the header as Items and save the dataset as "AR_Groceries.csv"

Data description

The dataset "AR_dataset" is read as transactions and contains 1499 transactions and 38 columns with a density of 0.3836 which means that, despite the empty cells due to the dataset being sparse, there are 21850 items in this dataset. The most purchased item is "vegetables" (1089) followed by "poultry" (613) and "waffles" (587) and we can see the top 10 most purchased items in figure 1. The minimum of items purchased is 4 and the maximum is 27 per transaction; and on average clients purchase about 15 items.

Figure 1



"Apriori" algorithm & Results

By applying the generic *Apriori* rule, the parameter used shows support = 0.1, confidence= 0.8 and we get 443 rules. When sorting this *Rules_1* by lift and selecting top 10 rules, we see that the highest lift is 1.2. A lift higher than 1 is a good sign and shows a strong rule; this means that the higher the lift is, the greater

are the chances to buy the associated item of the rule. Table 1 shows the result of *Rules_1* with the strongest rule being {dinner rols, eggs} → {vegetables}

Table 1

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{dinner rolls,eggs}	=>	{vegetables}	0.1440961	0.8780488	0.1641094	1.208627	216
[2]	{eggs,sandwich bags}	=>	{vegetables}	0.1227485	0.8761905	0.1400934	1.206069	184
[3]	{cheeses,eggs}	=>	{vegetables}	0.1454303	0.8755020	0.1661107	1.205122	218
[4]	{aluminum foil,sugar}	=>	{vegetables}	0.1260841	0.8709677	0.1447632	1.198880	189
[5]	{eggs,sandwich loaves}	=>	{vegetables}	0.1200801	0.8695652	0.1380921	1.196950	180

When changing the confidence parameter to 0.85, we obtain a set of 36 rules for vector *Rules_2.* However, when we sort by lift, the result is approximately the same as we can see on Table 2.

Table 2

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{dinner rolls,eggs}	=>	{vegetables}	0.1440961	0.8780488	0.1641094	1.208627	216
[2]	{eggs,sandwich bags}	=>	{vegetables}	0.1227485	0.8761905	0.1400934	1.206069	184
[3]	{cheeses,eggs}	=>	{vegetables}	0.1454303	0.8755020	0.1661107	1.205122	218
[4]	{aluminum foil,sugar}	=>	{vegetables}	0.1260841	0.8709677	0.1447632	1.198880	189
[5]	{eggs,sandwich loaves}	=>	{vegetables}	0.1200801	0.8695652	0.1380921	1.196950	180

On the other hand, if we lower the support down to 0.008 and confidence stays 0.80, we obtain a high number of rules (742,444) with a higher lift (see Table 3). It means that the items below are not showing frequently in the transactions (less than 1%) and if all the items on the left-hand side are purchased there is a 100% chance that hand soap will be purchased. However, having to count on these 7 items, does not help in determining a clear pattern.

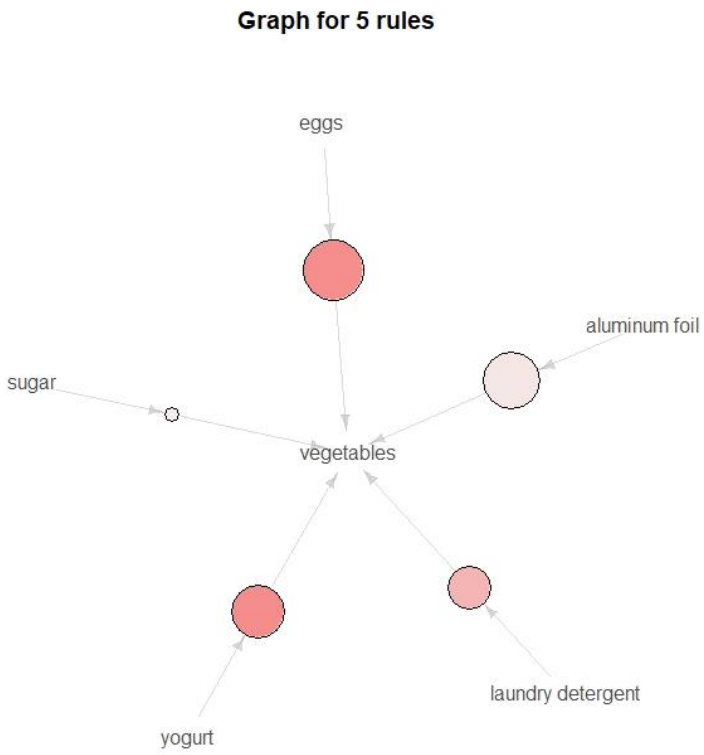
Table 3

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{all- purpose,dinner rolls,flour,mixes,paper towels,sandwich bags}	=>	{hand soap}	0.008005337	1	0.008005337	2.866157	12
[2]	{dinner rolls,flour,lunch meat,milk,mixes,yogurt}	=>	{hand soap}	0.008005337	1	0.008005337	2.866157	12
[3]	{eggs,pasta,pork,soda,spaghetti sauce,tortillas}	=>	{hand soap}	0.008005337	1	0.008005337	2.866157	12
[4]	{lunch meat,pasta,pork,soda,spaghetti sauce,tortillas}	=>	{hand soap}	0.008672448	1	0.008672448	2.866157	13
[5]	{beef,butter,coffee/tea,juice,poultry,shampoo}	=>	{hand soap}	0.008672448	1	0.008672448	2.866157	13

After trying with several parameters, the choice has been made to use a support of 0.25 and confidence of 0.80. The result shows 5 set of rules. All these rules are suggesting that vegetables will be purchased with a minimum lift of 1.1, which means that the rules are strong enough and that the buyer is 1.1 time more likely to buy vegetables if yogurt or eggs or detergent is bought. (See table 4) A representation of the 5 rules are displayed on Figure 2, it shows that eggs and yogurt have a strong chance to imply a purchase of vegetables.

Table 4

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{yogurt}	=>	{vegetables}	0.3082055	0.8148148	0.3782522	1.121586	462
[2]	{eggs}	=>	{vegetables}	0.3108739	0.8146853	0.3815877	1.121408	466
[3]	{laundry detergent}	=>	{vegetables}	0.3042028	0.8099467	0.3755837	1.114885	456
[4]	{aluminum foil}	=>	{vegetables}	0.3095397	0.8013817	0.3862575	1.103096	464
[5]	{sugar}	=>	{vegetables}	0.2935290	0.8000000	0.3669113	1.101194	440



Finally, since we noted that vegetables are the most purchased items, we can create a rule that will determine what item is going to be associated with vegetables in the basket. By swapping the items around and creating a rule using the vegetables on the left-hand side, we obtain the result in table 5. For this rule, the support selected is 0.31 and the confidence 0.43. The result implies that there is a 44% chance that a customer buys poultry when having bought vegetables.

Table 5

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{vegetables}	=> {poultry}	0.3202135	0.4407713	0.7264843	1.077841	480

To conclude, we can say that there is a high correlation between eggs and vegetable or poultry and vegetables. The choice of parameters depends on the shop strategy; do we want to look at the most frequent purchased items or do we want to upsell some items that are neglected by the consumer with a low support. In future analysis of this dataset, I would recommend removing the vegetables item to see what other meaningful association we can discover.

Part2

The second part of this report is using a data mining technique called Classification. It is a supervised technique used in machine learning. The goal of this method is to "assign an observation to a category or a class" (Gareth James, 2013) For example, Classification is often used in medicine to predict a condition following an observation of symptoms. It is a reliable technique where the algorithm is trained on known data and label which makes it easier to measure the success of its model.

Dataset description

The renamed dataset "Pulsar stars.csv" was retrieved from the UCI Machine Learning Repository website and is originally called HTRU2, as it was originally collected by the *High Time Resolution Universe Collaboration.* It contains a sample of pulsar candidate which are "a rare type of Neutron star that produce radio emission detectable here on Earth" (University of Manchester Centre for Doctoral Training in Computer Science, s.d.) The sample collected is the measures of the radio emission pattern from a pulsar beam emission as it rotates.

The dataset contains 9 variables without headers and 17,898 observations. The first 8 variables are the mean, standard deviation, kurtosis and skewness of the integrated profile and the dispersion measures of the signal to noise ratio and are continuous and numeric. The ninth variable which is the *class*, is an integer and binary type of data (1 = Pulsar and 0= Non-Pulsar). It has been decided to use this dataset as it is already labeled with the independent variable C*lass*, which means that the model is built using data already known.

Methodology

The method used in this classification is KNN (K Nearest Neighbor) which is also called lazy learner. It helps to predict the category of a new observation by comparing the features of this new item with the known and labelled groups using the Euclidean distance or the nearest proximity. The challenge in this method is to determine the value of K which is the number of nearest neighbors selected to make a prediction. (Augmented Startups, 2017).

The first step requires to prepare the dataset and we start by adding headers to the data frame. We then check for any missing values, which is not the case for this dataset. A normalization step is necessary to obtain a common scale and get values between 0 and 1 to get a more accurate result. (Towards AI, 2019) During this phase, we are also excluding the independent variable *Class* and creating a separate vector for this attribute.

The next step is to sample the data by shuffling the rows of the normalized dataset (*Pulsar_norm)* and the *Class* variable. We then create the training and test datasets for both the features and the *Class* (target). To determine the parameter *K* we will use the square root of the training dataset as a start which is 120.

The third step is to create the model using KNN Algorithm which is a function part of the "class" library. The function only requires the training and test dataset as well as the trained target and the value of K (120).

Results and Evaluation

The performance of the model is evaluated in the below confusion matrix (Table 1). As a "two-class" classification model, the table contains 2 categories for the predicted model and 2 for the actual or true class. As explained in the data description, "0" means "Non-Pulsar" and "1" means that it is a "Pulsar". The result obtained shows the correct decision in the diagonal; we correctly predicted that 3249 candidates are non-Pulsar and 246 are Pulsars. Unfortunately, the model wrongly predicted that 78 instances were non-Pulsar and that 7 were Pulsars. The total accuracy is 97.63% (Table 2) which means that the misclassification rate is 2.37%. The "sensitivity" of 75.9% is the percentage of the prediction correctly classified. The percentage of negative observations correctly classified or "specificity" is 99.8%. This means that the model is doing a better job in predicting the Non-Pulsar observations than predicting the actual Pulsars.

	N=3580	Predicted Model (Non-Pulsar)	Predicted Model (Pulsar)	
		0	1	
Actual *Class* Non-Pulsar	0 (negative class)	3249 = TN	7 = FP	3256
Actual *Class* Pulsar	1 (positive class)	78 = FN	246=TP ☆	324
		3327	253	

Table 1

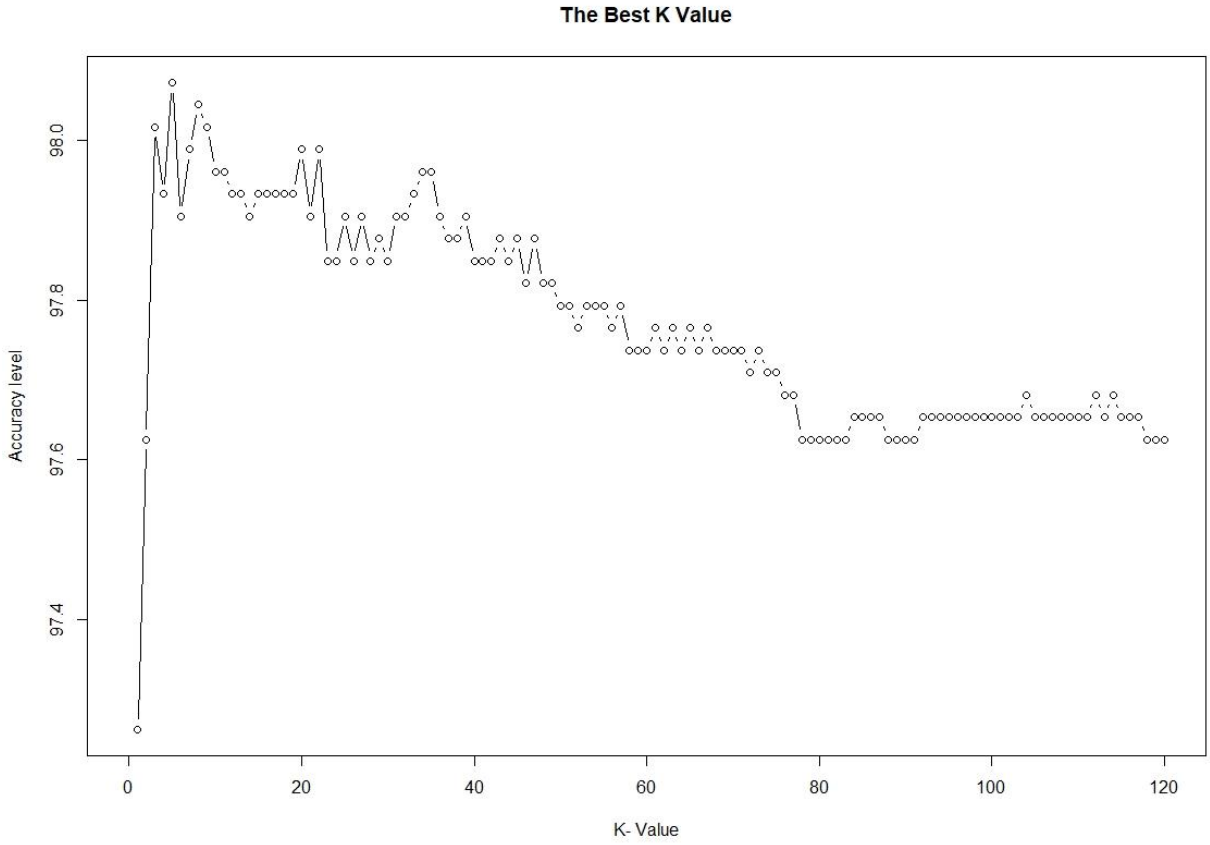
Table 2: Calculation Table

Total Accuracy	(3249+246)/3580 = 0.9763
Misclassification Rate	1 - 0.97 = 0.0237
True Positive Rate or Sensitivity or Recall	246/324 = 0.7593
False Positive Rate	7/3256 = 0.002
True Negative Rate or Specificity	3249/3256 = 0.9978
Precision	246/253 = 0.9723

Recommendations and Conclusion

We found that the model as an accuracy of 97.63% and is not that good at predicting Pulsars compared to predicting non-Pulsars. It means that the model could be improved to predict the Pulsar observations better. By using an iteration in a "for" loop to find the best value for K, we find that if K=5, the accuracy rises to 98.07% (see plot on figure 1). Therefore, we can re-evaluate the model and see if the sensitivity gives a better rate. Alternatively, we could also use other classification techniques such as Random Forest to check which method gives the best accuracy.

Figure 1


The Best K Value
Accuracy level
97.4
97.6
97.8
98.0
0
20
40
60
80
100
120
K- Value

References

Augmented Startups. (2017, 10 31). *Apriori Algorithm (Associated Learning) - Fun and Easy Machine Learning*. Retrieved from Youtube.com: https://www.youtube.com/watch?v=WGlMlS_Yydk

Augmented Startups. (2017, 08 07). *K - Nearest Neighbors - KNN Fun and Easy Machine Learning*. Retrieved from Youtube: https://www.youtube.com/watch?v=MDniRwXizWo

Dasu T, J. T. (2003). Exploratory Data Mining and Data Cleaning. Wiley.

Gareth James, D. W. (2013). *An Introduction to Statistical Learning.* New York: Springer.

Jabeen, H. (2018, 08 21). *Market Basket Analysis using R*. Retrieved from Datacamp.com: https://www.datacamp.com/community/tutorials/market-basket-analysis-r

NA, I. C. (n.d.). Retrieved from creativecommons.org: https://creativecommons.org/licenses/by-nc/3.0/

Swinburne Pulsar Group. (n.d.). *HL Survey*. Retrieved from astronomy.swin.edu.au: https://astronomy.swin.edu.au/pulsar/?topic=hlsurvey

Towards AI. (2019, 05 16). *Data Science*. Retrieved from Towardsai.net: https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff

University of Manchester Centre for Doctoral Training in Computer Science. (n.d.). *HTRU2 Data Set*. Retrieved from Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/HTRU2#

Wickham, H. (n.d.). *Tidyr*. Retrieved from Tidyverse.org: https://tidyr.tidyverse.org/