Customer segmentation using decision trees marketing essay

Business, Marketing

This chapter will continue to keep a technical focus on statistical techniques, but will switch to a more advanced set of methods. These methods are different than the ones studied in the previous chapter, in the way that we do not start with firm, well-defined assumptions about the models and their outcomes. There are some basic assumptions involved, that based on the data available we will have some working classifications. But this is far from having a well-defined hypothesis stating, for example, that sales for a store are a function of the size of the city where it is located, the number of competing stores, the incomes of the buyers and the season. Here we have much less of that; our task is to explore and seek out relationships between variables and come up with our models based on what we will discover. Often one of the ‘ dependent’ variables (as defined in a regression model) may be used as explanatory variable. While these methods have a statistical background and draw a lot from classical statistical and econometric methods, they grew into rather distinct fields called data mining and machine learning where the focus is more towards discovery of new relationships, and dealing quite often with large amounts of data. Due to these characteristics, these methods often come with specific approaches that are suited to their exploratory nature. This means that, compared to basic statistical techniques, these techniques have a heavy exploratory focus. Another issue is that these techniques are easily prone to what is called ‘ overfitting’. That is, we may get classifications which fit very well the data at hand, but, because of this, it will not work well when using other similar data, and lead us to fallacious analysis and poor business decisions. Therefore, using these techniques is essentially an exploratory process, when initial results are examined, revised and ammended. This is not entirely different from the regression work we did last course, but now we do not benefit from having that many clear-cut diagnostics (such as R2, p-values and residual plots) to say we are on the right track or not, and we do not have a fairly precise idea about the outcomes of our analysis. Thus, we now have to follow several specific steps to apply any of the methods described in this chapter. Remember from our previous course that we will use statistical ‘ tools’ which will not give us the proper solutions unless they are properly applied. These steps are a living proof of that. Initial results may be fine from the statistical point of view, but not good enough for using them in practice. And because of that, techniques that belong to data mining or machine learning fields come with procedures and steps to account for that. In data mining, a first run of the model is performed in order to get a first round of estimates, and then an analysis step follows. After that, when the initial estimates are produced and assessed, there comes a third phase, when the model is run again, based on the results from the first step, but using data that is slightly different from the one used in the first step. Then, after inspecting the subsequent results and running the models again based on the results obtained, the final assessment and conclusions are drawn. This chapter will present three of the most widely used advanced business analytics techniques used: decision trees, to classify customers and predict in which class new customers will fall into, clustering techniques used to group together items that have similar characteristics and are different from other groups or clusters, and market basket analysis, to identify which products are likely to be purchased together.

4. 1 Customer segmentation using decision trees/classification trees

Customer segmentation stems for a basic need to appropriately classify clients so as to target and manage them better. Most products and services are purchased by a wide variety of customers, with different characteristics. Even for highly personalized products, uniquely made for specific customers, there is the need to figure out their characteristics in order to anticipate and meet customer needs, and have the needed skills and materials available to do it. In order to gain a better understanding of customer segmentation, it is useful to review the basic segmentation variables that are used. In his book, ‘ Marketing Management’, Kotler (2006) identifies the main segmentation variables for the US consumer market as follows: geographic region, size of the city, area type (urban, suburban and rural), clime, age, family size, lifecycle stage (bachelor, married without children, etc), gender, income, profession, education, religion, race, generation, social class, lifestyle, personality, usage behavior, features sought after in a product or service, usage proficiency, frequency of use, loyalty, attitudes towards products, etc. This is a rather exhaustive list, and very often detailed information for customers is not available, or just not relevant. In some cases, no one has ever though collecting information about their customers, or this was just not feasible. In other cases, the information is not reliable. Let’s take the example of subscriber information for cell phone customers. In my case, I remember never having to do an update of my personal information except for my billing address, so data gathered from my initial contract might have changed quite a bit (income, employment, marital status, etc.). Then, apart from assuming that everything about me was unchanged except for a different address, how could the cell phone company profile me and get a picture of what are my characteristics as a customer? In most cases, usage and payment data complete the picture. Based on the calls made, the company could infer that I use this phone for personal use, since most calls were made during evenings and weekends. The phone numbers called matched against the customer database could tell how many of the calls are made with their customers, how many long-distance calls are done, how much text messages I do, and so on. Also, based on the cell phone antennas that my phone connected to, the company could tell where I usually go after work, and where do I live most of the time. And based on the IMEI number (a number that uniquely identifies any cellular phone) the company could tell what type of phone I used, whether I used the same phone from the initial contract and for how long, and whether I now use a smartphone for which I may need a data plan. In most cases, usage data are the best and most reliable resource, and most variables used in segmentation contain factual information about the customers. Other information is usually publicly available, the size of a city, the area of a city, average or, better, median incomes for a specific neighborhood, unemployment rates, inflation. In this case, the challenge is to merge customer usage data with location data using a key variable, as explained in the second chapter. Let us work through a practical example of customer segmentation, where we only have basic data. We will use the car usage summary data to classify customers based on the price paid on their cars. The first step is to grow a decision tree, that is to do a first run so as to unveil the basic features of the customers based on their relevant characteristics as incorporated in the variables. In order to do so, instructions and code will be provided, along with step-by-step guidance through the entire process. First, the rpart package needs to be loaded, and then the dataset cu. summary from the rpart package. Then the tree will ‘ be grow’ or, in plain language, a first run of the model will be performed, with the following command[1]: The rpart package is used to do decision trees. Decision trees are also known by other names, classification trees or regression trees, depending on the method used in bulding them.

Decision trees: Purpose and use

Decision trees is a classical data mining method to predict the value of one outcome (or target) variable as a function of several input variables. The prediction model resembles a tree, or more precisely a branch, where nodes are independent variables and the leafs show the value of the target variable given the values of input variables that describe each and every branch, from the leaf up to the root, as in Figure 4. 4. However, when building a decision tree, all observations of a data set become classified under each and every branch, based on the significant values of the input variables. This, along with the information of the value of the outcome variable, and the characteristics of the input variables, are essential in defining groups of observations which have similar characteristics, and obtaining a better understanding of them, as further analysis will focus on more homogeneous data. The use of decision trees for classification of customers is fairly widespread, and presented as such in several books (e. g. Bramer, 2007) . The first part of the formula after the opening bracket gives the model to be used: Price~ Mileage + Country + Reliability + Type. This is pretty important in defining the outcome variable. Another distinctive feature of this technique is that here the working hypothesis is that price is the relevant outcome variable, while other variable, say mileage, may be equally suited in describing customer behaviour. This is contrary to the regression model where the dependent variable would not change (unless we come up with a well-defined hypothesis as in the examples in chapter 3). Therefore we use the term outcome variable or target variable to refer to this key variable on which our estimation will be based. The rpart command has two methods, class and anova. Class is used to produce the classification tree, which is mainly based on categorization of the data. In our case, there are three variables that are categorical, country, reliability and type, so class may be a good candidate as an option for running our decision/classification tree. On the other hand, we have the Anova method, which will give us a regression tree. This tree is very similar to the classification tree, with one difference: instead of using a classification model, similar to the logistic regression, it will use a regression model (linear regression or another type of regression where the dependent variable is continuous) to build the tree. This is because in the dataset cu. summary we have two continuous variables, that can take a wide range of values: mileage and price. Two of the three categorical variables mentioned above can also be accomodated by the analysis using the regression method by assigning values to them. After running the decision tree, a review of the results will be carried out by using a few commands. The command printcp(a)gives the basic statistics on the regression tree. The most important are the ones which give the variables used in growing the tree (in our case country and type), the number of splits, or, in other words, each level of branches in the tree, and the error and standard deviation split, which will tell us how much misclassification we can get if we go to one of the splits listed in the output. The table shown in the output below goes from 0 splits to the largest number of splits, in our case 6. On the second column, CP stands for threshold complexity parameter or cost complexity factor, and essentially tells us what is the gain obtained in terms of a more precise classification by adding another split to the tree. This value will be very useful later in deciding whether, and how, we should revise and improve the tree. The relative error, rel error, is computed as 1-R2 , with R2 defined similarly to the regression analysis, and tells how well the model fits the data. The cross-validation error, xerror, gives the misclassification error after performing cross-validations of the data, and xstd gives the standard deviation of the cross-validation data. Regression tree: rpart(formula = Price ~ Mileage + Country + Reliability + Type, data = cu. summary, method = " anova")Variables actually used in tree construction:[1] Country TypeRoot node error: 7407472615/117 = 63311732n= 117CP nsplit rel error xerror xstd1 0. 250522 0 1. 00000 1. 02719 0. 1617342 0. 148359 1 0. 74948 0. 90639 0. 1636053 0. 087654 2 0. 60112 0. 75054 0. 1257644 0. 062818 3 0. 51347 0. 58233 0. 1031355 0. 010519 4 0. 45065 0. 52292 0. 0976086 0. 010308 5 0. 44013 0. 54202 0. 0999507 0. 010000 6 0. 42982 0. 54202 0. 099950

Cross-validation: intuition and purpose

Cross-validation is a standard procedure performed by data mining and machine learning algorithms. The basic concept behind it is to run the model obtained on similar data, that is not exactly the same as the data we have analyzed. This can be done by randomly taking a number of observations/rows from the data, and then applying the model. This will give an indication as to whether the model is overfitting[2]the data or not, and, more important, whether the model has been obtained so as to be the best fit for our purpose. In other words, cross-validation results give us the confidence that we can run the model on similar data and get reliable results each and every time we do so. The command plotcp(a) will give a picture of the splits and the classification error associated with each split. Its result can be seen in Figure 4. 1. Note that split means the level of branches in the tree; that is a split of 2 corresponds to a tree size of three. That is, from the core node, which contains all the observations, there are two levels of ‘ branches’ in which the observations are classified. Figure 4. 1 Relative error plot of the initial decision treeHere we start forming an idea about the usefulness of the obtained results, and on the size of the tree to be used later on. We can see that when we reach a size tree of 5, the error is the smallest, and here is the point where we can decide to shorten or ‘ prune’ the tree. The following commands will offer us an additional set of diagnostics to help us decide the optimal length of a tree, and later on, to decide how to ‘ prune’ the tree. par(mfrow= c(1, 2))rsq. rpart(a)In the left graph in Figure 4. 2 we have the R2 of the classification tree, both for the initial run of the model shown in the Actual line, and for the cross-validation run shown in the X Relative line. Here we can see whether our model is or not prone to overfitting and how good the fit is. The left graph shows the relative error on the cross-validation data as the number of splits increase. Here, it looks that the optimal number of splits is 4, after that it makes little sense have more branches, or, in other words, more classes. Why? The gain in the fit will be minimal (and decreasing for cross-validation data) and may not be relevant for similar data other than the ones used for building this. Also, the relative error will be increasing after node 4, which confirms the conclusions drawn after inspecting the left graph. Figure 4. 2 Diagnostic plots of the initial decision treeFinally, the print(a) command shows the basic structure of our tree, and allows us to see which branches are most important. Its results are shown in the output below. n= 117node), split, n, deviance, yval* denotes terminal node1) root 117 7407473000 15743. 4602) Type= Compact, Small, Sporty, Van 80 3322389000 13035. 0104) Country= Brazil, France, Japan, Japan/USA, Korea, Mexico, USA 69 1426421000 11555. 1608) Type= Small 21 50309830 7629. 048 *9) Type= Compact, Sporty, Van 48 910790000 13272. 83018) Country= Japan/USA, Mexico, USA 29 482343500 12241. 550 *19) Country= France, Japan 19 350528000 14846. 890 *5) Country= Germany, Sweden 11 797004200 22317. 730 *3) Type= Large, Medium 37 2229351000 21599. 5706) Country= France, Korea, USA 25 1021102000 18697. 28012) Type= Medium 18 741101600 17607. 440 *13) Type= Large 7 203645100 21499. 710 *7) Country= England, Germany, Japan, Sweden 12 558955000 27646. 000 *While this is not the most intuitive way to showcase the results, it could be useful for the inspection of the obtained tree, and for checking if the tree graphs are correctly plotted.

After the node number, there comes a description of the classification variable used (e. g. Type= Large for node 13), and how many observations are in the node (e. g. 7 for node 13). Deviance here stands for the corrected error, and Yval is the mean price for the car values that were classified in a particular node. Finally we will use the commands below to get the tree, as shown in figure 4. 3. plot(a, uniform= TRUE, main=" Regression Tree for Price ")text(a, use. n= TRUE, all= TRUE, cex=. 8)Figure 4. 3 Basic decision tree plotBut this is not yet very informative since we do not have a clear description of all variables, type and country. As a remark, you can see now that the decision tree algorithm has converted the categorical variables, type and country, into values shown as letters. In order to obtain an understandable picture of the tree, we will create a nice postscript file which will show the tree in all its details. After using this command to output the filepost(a, file = " D:/tree2. ps", title = " Regression Tree for Price ")you can visualize the exported postscript file in figure 4. 4. I use this site http://view. samurajdata. se/ to convert it into a gif image, which is shown below. Figure 4. 4 Detailed decision tree plotHere is the final result of the initial tree. It contains details on all the branches: how many observations are in there shown as n=, the composition of each node, as shown by the variables type and country, and the average price of the cars (the target variable) for each node. After obtaining the initial result, we will need to question whether the decision tree obtained is the best one, and whether a better one can be obtained based on the results gotten from the first run of the model. Before deciding whether to prune the tree, a closer look needs to be taken at the diagnostic results obtained earlier (shown below for convenience). CP nsplit rel error xerror xstd1 0. 250522 0 1. 00000 1. 02719 0. 1617342 0. 148359 1 0. 74948 0. 90639 0. 1636053 0. 087654 2 0. 60112 0. 75054 0. 1257644 0. 062818 3 0. 51347 0. 58233 0. 1031355 0. 010519 4 0. 45065 0. 52292 0. 0976086 0. 010308 5 0. 44013 0. 54202 0. 0999507 0. 010000 6 0. 42982 0. 54202 0. 099950If we go to the 5th and 6th splits, we notice that both cross-validation classification error and standard error increase. This is shown in the graphs as well, and raises the question as to how well the model was computed. If the rpart package runs well, both errors should go down gradually. It is obvious that in this case this does not happen. Introducing a cp parameter in the initial formula to grow the tree and changing it will fix the problem. In this case we have used a cp of 0. 011, obtained after trying several different values for it. The results are now: Regression tree: rpart(formula = Price ~ Mileage + Country + Reliability + Type, data = cu. summary, method = " anova", cp = 0. 011)Variables actually used in tree construction:[1] Country TypeRoot node error: 7407472615/117 = 63311732n= 117CP nsplit rel error xerror xstd1 0. 250522 0 1. 00000 1. 02718 0. 1600222 0. 148359 1 0. 74948 0. 88013 0. 1483813 0. 087654 2 0. 60112 0. 76696 0. 1495904 0. 062818 3 0. 51347 0. 66869 0. 1361365 0. 011000 4 0. 45065 0. 50473 0. 092565Now we obtained a cross-validation error of 0. 50473, and a standard error of 0. 092565 for the last split, which are better than the previous results. It looks like the 4th split is the cutoff point as before, and now the cross-validation scores are higher than before, which means that misclassification is lower. In the graph showing the size of the tree, we obtain the optimal size of the tree to be 4 (Figure 4. 5) as shown by the dotted line. We see that the cross-validation cutoff line is 0. 6, slightly below the value obtained before. Figure 4. 5 Relative error plot of the revised decision treeFigure 4. 6 Diagnostic plots of the revised decision treeDiagnostic plots also confirm that this is the optimal size of a tree. At the 4th split the actual (labeled Apparent) and the cross-validation error reach the highest point in the left graph of figure 4. 6, and the cross-validation relative error in the right graph shows a significant decrease between the 3rd split and the 4th split. Thus, we are confident that we have obtained a good classification model and that we are now able to segment our customer base of car owners based on the price of their cars. Based on it, let us produce the final version of our classification. For this, we will now need to prune the tree, choosing a cp value that will get a tree size of 4 (or three splits). Given our previous results, this should be a value of above 0. 062818.

How to make a final decision on a model. Discussion for decision trees

Deciding which model is best is rather an art than a science. It is often the case that we have several diagnostic measures, as for example R2 and relative error. They will only give a picture of the fit of the model, and the misclassification error that is a given for any statistical model. Can we base our decision on these parameters alone? The answer is no. While we can have an excellent fit of the model for our data (including cross-validation results), these results may not be very relevant, and even not robust enough. This will result later in potentially larger misclassification errors, or to obtain results that have little relevance. Let’s just look at the example above, and let us also look at the final classification tree produced below in Figure 4. 7. Based on the R2 and relative error, the optimal size of the tree is

5. However, if we look at the complexity factor, we see that the gain for the last split is quite small, a bit more than 1%. By definition, this threshold complexity parameter tells us the additional gain obtained by adding another split. This is bound to contradict the R2 and relative error results in our case, and raises a question mark on the usefulness of our classification. Should we keep this tree structure, or cut the tree to a higher value of the complexity parameter? In this case, an examination of the final outcome in Figure 4. 7 becomes useful. We see that for two of the four nodes obtained with a three-node split we get about 10% of the observations. This is a relatively small part of our customer base, which can become even smaller if extra splits are added. Would this be relevant for an overall description of our customer base? Can these segments prove useful in making decisions about the customers that are grouped into them? The answer is often found in one's knowledge and intuition about the business, and the data that describes the customers. However, in this case, I decided that having an extra level of detail would amount to ' seeing the trees rather than the wood' and creating small segments that are not very useful for making groups of customers and addressing their needs due to their extremely small size. Choosing a value of 0. 07, we will prune the tree and get a picture of it using the commands: plot(af, uniform= TRUE, main=" Regression Tree for Price ")text(af, use. n= TRUE, all= TRUE, cex=. 6)post(af, file = " D:/tree. ps", title = " Final Regression Tree for Price ")Now we got the final version of the classification of customers using the price of their cars. There are now two segments consisting of large and medium car owners with an average price of $21, 600[3], and another one of compact, small, minivan and sport cars with an

average price of $13, 400. Then, compact, small, minivan and sport car owners are split into two segments, owners of German and Swedish cars with an average price of $22, 320, and owners of cars made in other countries with an average price of $11, 560. Similarly, owners of large and medium cars classify into two different segments, owners of more expensive English, German Japanese and Swedish cars, and owners of less expensive Korean, French and US cars. Figure 4. 7 Detailed decision tree plotWhy is this classification better than the ones obtained before? If we look at the first model with the unpruned tree (Figure 4. 4), we see, for example, that owners of compact, small, minivan and sport cars made in Brazil, France, Japan, US Korea and Mexico, can be split into a segment of small car owners group with an average price of $7. 600, and owners of more expensive compact, minivan and sport cars with an average price of $13. 000. There are also several other nodes generated that further split our customer base into finer classes. Wouldn’t this classification be much better that the final result in Figure 4. 7? One reason why it isn’t so is that, at the level of three splits, we get no more possible splits for two of the terminal nodes (the lowest ones in the above picture). More specifically, the right nodes of more expensive cars, both large and medium and compact, small, minivan and sport cars cannot be split further. Another one is that the size of the data for some nodes is fairly small, 11 for the second terminal node and 12 for the fourth terminal node in Figure 4. 7. We only got a total of 117 customers. If we will not get significantly more customers in the future, does it make sense to refine our segmentation and devise strategies to treat them differently, strategies that will cost money and not bring significant benefits? On the other hand, there are reasons to go for an extended tree. Misclassification of customers based on price is a bit high at 0. 6, and adding a node could make it go down to 0. 5. Also, the first node with less expensive cars compact, small, minivan and sport cars may benefit from isolating the small cars group with an average price of $7. 600, and allow us to concentrate the efforts on the other customers, that may be willing to spend more on your services. While there are statistical benchmarks used by some authors to tell us where to prune the tree, I believe that one’s experience can guide you better in making decisions. However, apart for a good knowledge of the business, it is essential to understand and play a bit around with such a classification model and correctly interpret the results obtained, in order to get a segmentation model you trust is useful for your purposes. 4. 2 Clustering based on multiple criteria. The RFM frameworkA similar method to classify customers and certain transactions is clustering. Formally, clustering is the method that allows us to classify observations (e. g. customer records) in a way that observations belonging to the same cluster are more similar to the ones that belong to other clusters. Thus, clustering can also be used in segmenting customers by grouping them into classes, or clusters. But clustering can be used to do other analyses as well. Clustering can group similar transactions, in order to analyze them and draw conclusions about them. As an example, let’s take a series of daily sales for a period. We can of course group them using quartiles and percentiles, and compute the mean. But it is often more informative to group them using clustering, making use of several variables at the same time, so as to get homogeneous groups and then start analyzing their features. Clustering can also be used to group certain products and services based on their features. This will allow us to see the similarities and dissimilarities between them, and based on that, to fine-tune sales, marketing and procurement activities with the purpose to increase sales and/or reduce costs. In analyzing customers (including a segmentation of the entire customer base), one of the best known analysis frameworks is the RFM. RFM stands for Recency, Frequency and Monetary. These three criteria combined will give you the necessary information to maximize the value of your customers and increase profitability. Based on this framework, there are three main factors to take into consideration when analyzing customers. The first one is recency, which tells what was the last time a given customer did business with you. There is often a relationship between the sales for a given customer, and the time passed since his/her last transaction. In some cases, for fast moving consumer goods (FMCG), this tells the likelihood that the customer makes frequent purchases for you. In other cases, for durable goods (e. g. vacuum cleaner), this may indicate that he/she is not likely to purchase the same good for a while, but that he/she might be interested in purchasing other durable goods such as a washing machine. The second main factor is frequency. This tells you how likely is a customer to do business with you in the future, and can be an indicator of how much will he/she spend in the future on your products or services. For example, a cell phone user who uses his/her phone to make frequent calls is more likely to be a customer in the future and spend money on airtime. Based on usage and plan, we can infer how much he/she is likely to spend in the future. On the other hand, a less frequent user is less likely to spend less on our services, and maybe more likely to go to the competition if there is a less expensive plan that fits his low usage pattern. The third and perhaps the most trivial factor is the monetary one. This one classifies customers into those who make large purchases, sustain sales, and may be interested in discounts or deals, and customers who do not spend a lot, do not generate many sales, may also buy from competition to fulfill their consumption needs, and could even drive up costs as it takes time to be attended by sales and marketing people, while the sales generated may barely cover the cost of the time spend by your employees. Let us do an example to see how clustering works in practice by taking a file with customer sales, and cluster the customers according to the last transaction date (Rec), number of items purchased, and total amount of transactions done in a given period of time. First, the package cluster needs to be installed and loaded. Then an initial clustering using the dataset CDNOW_sample. xls can be run. This dataset is an adaptation of a dataset made available by Bruce Hardie, and contains CD sales data for two consecutive years. In the remainder of this subchapter, the imported dataset will be labeled cd. First the ID column must be deleted so that customer numbers are not used in doing the clustering. From the menu Data, Manage variables in active data set, Delete variables in the data set, you need to select the ID variable and then click OK[4]With the following commands we will be able to visualize and assess the results in Figure 4. 8: layout(matrix(c(1, 2), 1, 2))plot(a)Figure 4. 8 Initial Cluster Analysis ResultsThe results of clustering shown in the left table look a bit strange. A closer look at the data reveals that this is mainly due to the fact that it refers to CD sales with prices which are fairly standard. This result is different from a textbook example, where data is more heterogeneous and clusters are fairly round and distinct. But the message that the first two components explain 99% of the point variables confirms that our clustering fits the data well. This is the first indicator that clustering works well on the data you have. A low value here, of below 70%, may indicate some data issues, such as having outliers[5]in the data, which may bias the classification results, and eventually yield a misleading result. This is especially important for the cluster analysis, since we have several variables in our analysis dataset, and the impact of the outliers can be harder to assess. The right plot confirms that. It is called the silhouette plot and shows how well the clusters are formed, and whether we have the right number of clusters. The key statistic here is the average silhouette width, which shows how well the model chosen manages to classify the data into clusters. A value below 0. 5 shows that the clustering structure is not good and needs revision, or another analysis method (e. g. decision trees) may be more suitable. A value of 0. 5 to 0. 7 indicates a reasonable cluster structure, and above 0. 7 we have a strong structure. Another important feature of the silhouette plot is the way it shows the silhouette widths for each and every cluster, and the misclassification which occurs when the silhouette plot goes negative, as for clusters 5 and 6. The average silhouette widths of 0. 32 and 0. 21 for the last two clusters shows that clustering is fairly poor.

Getting the best cluster structure, an educated trial-and-run procedure

When doing the analysis and choosing the best clustering solution, you will notice that, in general, the less clusters you have, the more likely it is to have a higher average silhouette width. However, one should not fall into the trap of getting too few clusters just by aiming to get a high average silhouette width value. Rather, the average silhouette width should be assessed in conjunction with the misclassification shown in the silhouette plot, along with silhouette width values obtained for the clusters. In doing several runs of the clustering algorithm, an analyst will need to develop a feel for the dataset analyzed, and compare several results until he/she feels comfortable with the final choice made. In some cases it may be impossible to get extremely good average silhouette widths (above 0. 7), but if there is little misclassification present, this may be just fine. In many cases, a decision based on all three parameters, average silhouette widths for the overall cluster structure, average silhouette widths for the individual clusters and the misclassification seen in the negative region of the silhouette plot for the clusters, would help in choosing the best cluster structure. It will also allow one to avoid creating few clusters, or to create clusters with poor parameters. In some cases it is worth assigning the clusters to the dataset and do an inspection, as shown at the end of this section, to see if the clusters make sense. This is the best way to gain more confidence in explaining them, and do an ultimate validation of the chosen solution, in terms of number of clusters. In conclusion, all results from the silhouette plot in the right of figure 4. 8 point out that it would be better to change the number of clusters by reducing them, in the hope to obtain clusters for which classification yields better results and little or no misclassification. Figure 4. 9 Final Cluster Analysis ResultsOnce we have decided on the cluster structure that is deemed to be the best one, we will need to add the cluster numbers to the dataset so that we can classify all customers in our database with the commandIf we now convert the cluster number to a factor with the command[6], as shown in the output below: Variable: Datemean sd 0% 25% 50% 75% 100% n1 263. 4436 146. 3472 101 130 221. 0 316. 00 731 15962 1296. 7816 233. 3063 706 1121 1308. 5 1512. 25 1630 760Variable: Freqmean sd 0% 25% 50% 75% 100% n1 3. 011905 4. 011799 1 1 2 3 64 15962 14. 860526 17. 021850 2 5 9 18 129 760Variable: VALmean sd 0% 25% 50% 75% 100% n1 44. 9409 60. 80212 0. 00 14. 96 27. 94 50. 6850 990. 28 15962 218. 1757 251. 36101 17. 26 71. 88 133. 06 258. 0325 1943. 58 760This is how the basic characteristics of the obtained clusters look like. In the first cluster we have customers with fairly old purchases. The numbers for the mean and the quintiles are not relevant per se, since they are computed from last purchase dates for two years with a date format of YMMDD, but they show that the first cluster groups customers with the least recent transactions, while the second shows customers with more recent transactions. Those in the second cluster have made more purchases during the analysis period, about 15 on average, than those in the first with a mean of only 3. Also, total purchases for the period for customers grouped in the second cluster is, on average, more than five times higher than those for customers in the first cluster (44. 49 versus 218. 17). A final summary note when doing the analysis and choosing the best clustering solution. First, one needs to see if the clustering algorithm fits the data well, and remove outliers if needed. Then, one needs to use the silhouette plot to decide if you get a good overall fit, and good results for the individual clusters obtained. After obtaining the final cluster solution, it is recommended to assign the clusters to the dataset and do an inspection as shown above to see if the clusters make sense, and be able to describe them in a more intuitive way to your audience/boss/customer.

Decision Trees vs. Cluster Analysis

We have studies so far two methods of classification, decision trees and clustering, and used them to group customers, and, respectively, transactions, into specific classes. Now which of them methods is better and serves best different analysis purposes pertains to the features of the data sets and the specific focus of the analysis. Decision trees are more amenable to an analysis that focuses on one outcome variable at a time, and tries to pinpoint how its evolution is explained by other variables. Thus, its narrow focus, and the enhanced ability to predict into which group (node) will a new element fall into makes it the method of choice when both of these qualities are needed. On the other hand, cluster analysis is more flexible with respect to the variables included. Here there is no target variable, and all variables considered contribute to defining the clusters inasmuch as their variation is significant. This, and the more precise diagnostics available for assessing the fit of the obtained clusters recommends it for a more exploratory analysis where discovery is more important and assumptions less strict. Also, by running summary statistics, it is possible to classify observations into a specific class, albeit not as well as in the case of decision trees. 4. 3 Market Basket/Association AnalysisThe last and maybe most interesting topic for many practicing analysts is designing a market basket. This is a powerful data mining tool, with immediate applicability for sales, which tells what items are most frequently purchased together with others. Let’s take a sandwich shop for example. It would be useful to know what kind of drink is purchased together with each of the sandwiches provided, or what other products are bought together with expensive coffees (cappuccino, moccacino, etc.). This is useful for designing attractive display layouts, designing bundle offers and, for online stores, even designing recommender systems for online sales. Perhaps the most famous example of a practical application of association analysis, or market basket analysis, is the " beer and diapers" story, where a supermarket chain has discovered that customers that buy diapers are also likely to buy beer. As a result, diapers were put on shelves close to beer shelves, which lead to an increase in sales for both items. Doing a market basket, or association analysis, requires sales data obtained from the POS terminals, or from the items inputted in a cash register, or just taken down by the sales clerks, or transcribed from issued receipts. In our example we will use a dataset containing simulated data for convenience store sales. The package to be used is arules. You will need to install and load arules as shown in the previous classes. Now you will need to get the data into a format suitable for market basket analysis. More specifically, the data often comes as a table with transaction ID and item purchased as shown below: Table 4. 1 Typical layout for basic sales dataTransaction IDItem1Candy1Pop1Chips1GumInstead, what is needed is a data structure which shows whether any of the items shown above are present or absent from a transaction in a structure like the one in table 4. 2. Table 4. 2 Data format for market basket analysisTransactionCandyPopBagelChipsGumBread10111012000101301011140011015010101One easy way to do this in Excel is to do a pivot table, which will put ones as the count for the items that appear in a transaction. You will need to fill in zeros for the missing items with a replace command (after copying and pasting the pivot table as values only, and replacing the missing data with zeros with Ctrl+H command). The resulting data is available in the dataset customers. xls, which will be used to do the example below. Once you will load the data, you will need to do one more operation: convert all variables to factors. This is done by going to the menu option Convert numeric variables to factors, selecting all variables, checking the option use numbers, and then clicking OK to replace all variables with numbers. Figure 4. 10 shows you how to do this. Figure 4. 10 Converting dataset variables into factorsNow the market basket rules analysis can be run with the following command: Note that the parameter sup stands for support and shows the probability of occurrence for the transactions that contain the items for which association rules are generated. For example a 0. 2 probability for the items bagel and candy shows that in 20% of the cases customers have bought both items. conf stands for confidence and shows the probability that a transaction containing an item will contain the other item. Confidence is very similar to correlation, and its values should be interpreted more like those for the correlation coefficients explained in chapter 3. Here, both parameters are set so as the association rules algorithm will choose only combinations of products that are significant on both counts. If the algorithm ran correctly the output generated will look like this: parameter specification: confidence minval smax arem aval originalSupport support minlen maxlen target0. 7 0. 1 1 none FALSE TRUE 0. 05 1 10 rulesext FALSEalgorithmic control: filter tree heap memopt load sort verbose0. 1 TRUE TRUE FALSE TRUE 2 TRUEapriori - find association rules with the apriori algorithmversion 4. 21 (2004. 05. 09) (c) 1996-2004 Christian Borgeltset item appearances ...[0 item(s)] done [0. 00s]. set transactions ...[397 item(s), 885 transaction(s)] done [0. 00s]. sorting and recoding items ... [10 item(s)] done [0. 00s]. creating transaction tree ... done [0. 00s]. checking subsets of size 1 2 3 4 5 6 done [0. 00s]. writing ... [276 rule(s)] done [0. 00s]. creating S4 object ... done [0. 00s]. The following command will give you the information on the strongest rules found inspect(head(sort(a, by =" support"))The resulting output is: lhs rhs support confidence lift1 {Candy= 0} => {Gum= 0} 0. 3830508 0. 9495798 2. 3024062 {Gum= 0} => {Candy= 0} 0. 3830508 0. 9287671 2. 3024063 {Bagel= 0} => {Gum= 0} 0. 3129944 0. 9518900 2. 3080074 {Gum= 0} => {Bagel= 0} 0. 3129944 0. 7589041 2. 3080075 {Bagel= 0} => {Candy= 0} 0. 3107345 0. 9450172 2. 3426906 {Candy= 0} => {Bagel= 0} 0. 3107345 0. 7703081 2. 342690These are the top rules with highest support, confidence and lift. Let us look at them into more detail. A description of the rule is given in the first columns. {Candy= 0} => {Gum= 0} means that customer who did not purchase candies did not purchase gum either. This relationship occurs in 38% of all transactions, as shown by the support parameter. The confidence parameter shows that we can be 95% percent sure that if candies have not been purchased, gum is not likely to have been purchased either. The new measure, the lift, shows how good the association rule found is in explaining joint purchases for the two items compared with the average purchase. In the example above, this gives the probability of a purchase not to contain both gum or bagel, versus the probability of a purchase of not containing any of these items. A value of 1 and below shows that the rule is not good. In other words, it shows that we are not able to predict an association between the items any more than by picking a random transaction containing (or, in our case, not containing) any of the two items. The higher the value of the lift, the stronger is the association rule found. In this case, the likelihood of not purchasing both items is 2. 3 times higher than the likelihood of a transaction which does not involve any of the two items. Now, by analyzing these rules, we see that in fact they are of little or no interest, as they do not refer to any purchase being made. Remember from table 4. 2 that 0 stands for no purchase. They merely show that someone not purchasing candy is not likely to purchase bagels or gum, and that people that are not buying bagels are highly unlikely to buy gum. This should not discourage one in any way! In real life, association relationships are often not that obvious and do not occur that frequently; if this were the case, people will notice them right away from the data, hence there will be little or no need to run an association analysis. Luckily, we have a total of 276 rules generated for the analyzed dataset, that are not reported by the summary command above. Since we are interested in associations for which customers purchased at least one item, which may help us find some rules valid for joint purchases of items, we will need to isolate them and inspect the results obtained. The way to get information on these rules is by subsetting the results, with a command like the one below: With this command we selected transactions which include purchases of chips, and with a lift indicating that this association rule does a good job in predicting associations between purchases of chips and of other items.

Running the command summary(rulesChips)we see that we have a total of 13 rules selected using this criteria. Running the inspect command on the subsetted rules, which are now stored in rulesChips we get again the top 6 rules in the output below: lhs rhs support confidence lift1 {Pop= 1, Bread= 1} => {Chips= 1} 0. 09039548 0. 8333333 3. 2065222 {Pop= 1, Gum= 0, Bread= 1} => {Chips= 1} 0. 08135593 0. 8181818 3. 1482213 {Candy= 0, Pop= 1, Bread= 1} => {Chips= 1} 0. 07683616 0. 8095238 3. 1149074 {Candy= 0, Pop= 1, Gum= 0, Bread= 1} => {Chips= 1} 0. 06779661 0. 7894737 3. 0377575 {Pop= 1} => {Chips= 1} 0. 14124294 0. 7575758 2. 9150206 {Pop= 1, Gum= 0} => {Chips= 1} 0. 12768362 0. 7434211 2. 860555Now a close inspection of these will tell that essentially only two association rules matter here: purchases of chips are likely to be made together with purchases of pop and bread, and purchases of chips are likely to be made jointly with purchases of pop. We get more association rules for these, as this is the way the algorithm computing association rules runs. It takes every item (variable) or groups of items (e. g {Candy= 0, Pop= 1, Bread= 1}, shown in the left-hand side (lhs) and then computes association rules between it and the other items taken either alone or together, in pairs of two, three or more, shown in the right-hand side (rhs). This feature of the association rule algorithm can be very useful when there is some precedence or conditioning relationship between the items for which we do the association analysis. In some cases, the association between a first occurrence of an item, shown in the left-hand side (transaction or another event for which we try to discover association patterns) and the subsequent occurrence of a second item, is not the same as the occurrence of the second item first, followed by the occurrence of the first item. An example is the purchase of spare parts or accessories for cars. A high proportion of people (say 90%) who purchase tires and auto accessories also get automotive services done. However, a smaller proportion of people who get automotive services will purchase tires and accessories (say 60%) because, in many cases, these services consist of major repairs on breaks, gearboxes, suspensions etc., cases in which the customer does not care much about getting items of relatively smaller importance. However, to conclude our discussion, in our example, and in many real-life situations, the order of occurrence of an item does not matter much. Aside from the common knowledge and anecdotal evidence, this can be usually seen in the rules generated by the algorithm. If results for a rule A => B are the same, or almost the same as the results for the rule B => A, there is living proof that the order of occurrence of both items in a transaction does not matter much. Back to our example, it would be useful to have a look at all the rules which contain chips on the right-hand side, and isolate the ones that would make most sense and could be used further in designing sales strategies based on them. For more information on these rules, you may want to get them all and look at all of them. The commandWRITE(rulesChips, file = " D: data. csv", sep=",", col. names= NA)exports all association rules found into a csv file. Results will appear rather messy in the csv file and require some manual work to get them in a more appealing form as in the table below, but contain all the information you need to see, for all the rules. The results below show other strong association rules between purchases of chips and purchases of bagels and bread taken together. And they also show that chip buyers are unlikely to buy gum or candies, so it may not be a good idea to make a combo offer for chips, gum or candy to increase sales. Table 4. 3 Association rules

Rulessupportconfidencelift8{Pop= 1} => {Chips= 1}0. 1412430. 7575762. 9150229{Bagel= 1Bread= 1} => {Chips= 1}0. 0621470. 7333332. 82173959{Pop= 1Bread= 1} => {Chips= 1}0. 0903950. 8333333. 20652265{Pop= 1Bagel= 0} => {Chips= 1}0. 0960450. 7142862. 74844767{Candy= 0Pop= 1} => {Chips= 1}0. 1242940. 7432432. 85987169{Pop= 1Gum= 0} => {Chips= 1}0. 1276840. 7434212. 860555116{Candy= 0Bagel= 1Bread= 1} => {Chips= 1}0. 0542370. 7164182. 756652119{Bagel= 1Gum= 0Bread= 1} => {Chips= 1}0. 0587570. 7323942. 818126156{Candy= 0Pop= 1Bread= 1} => {Chips= 1}0. 0768360. 8095243. 114907158{Pop= 1Gum= 0Bread= 1} => {Chips= 1}0. 0813560. 8181823. 148221173{Candy= 0Pop= 1Gum= 0} => {Chips= 1}0. 1118640. 7279412. 800991222{Candy= 0Bagel= 1Gum= 0Bread= 1} => {Chips= 1}0. 0508470. 7142862. 748447245{Candy= 0Pop= 1Gum= 0Bread= 1} => {Chips= 1}0. 0677970. 7894743. 037757A similar exercise in an assignment at the end of this chapter will require finding association rules for other items, say for candy. One should not get discouraged if no meaningful results are obtained the first time the algorithm is run, but should instead try to change the confidence and support parameters to be able to get meaningful relationships. This practice is similar to the ones used for decision trees and clustering methods, and is tied to the knowledge discovery process. In some cases one may get no results at all, or unreliable results, but again this should not discourage anyone. No reliable result obtained can also have informational value, showing that there is no meaningful relationship to be obtained for associations between certain items. Translating a lack of an association relationship in practice can be trickier, but it can also have its rewards. In some cases, where sales are highly price sensitive, making a combo offer on these items may have a trigger effect on sales for other items. What if a bundle offer for gum and bagels can increase the sales of chips? Given the fact that we know that customers buying bagels are likely to also buy chips, this could be likely. However, the offer of such a bundle needs some testing, and other additional information, such as price elasticity and price points, that will be explored in the next chapter, combined with some trial offers, may tell us whether the offer is successful or not. 4. 4 Exercises and Review Questions1. What parameters should one look at when making a decision tree? Would information other than statistical parameters be useful in deciding the optimal segmentation of customers? 2. Please indicate all statistical measures that must be used in order to decide on the optimal cluster structure, and attempt to rank them according to their importance. Can you draw a decision making pattern out of it (for example a diagram)? 3. Can you notice some important differences, between association analysis and both decision trees? (Hint: how does the analysis approach differ, and how do we use the results?)4. Can we use cluster analysis to segment our customer base? Why would that be better than decision tree, and what would it not do well? 5. Using the data from the cluster analysis example, please do a segmentation of the clients. Compare the results with the analysis obtained with the cluster analysis. What are your conclusions? 6. With the association rules data, please run the analysis for Bread and show the strongest association rules. Please write a brief summary of results. Then try to run the analysis for candy. What results do you obtain, and what conclusions can you draw from them? 7. Suppose you have an analysis assignment for a bank. What usage data are likely to be available from the bank statement? What other data should be available for a particular customer? Based on your answers, please indicate what variables could be used in segmenting the customer base? 8. Based on the answers for question seven, and on your personal experience, can you identify and propose some opportunities/situations for doing an association analysis? What could be the possible outcomes of your proposals, and what improvements/suggestions could be done based on them?