

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
NPTEL
NPTEL ONLINE CERTIFICATION COURSE
Business Analytics & Data Mining Modeling
Using R – Part II
Lecture-02
Association Rules – Part II
With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

Business Analytics & Data Mining Modeling Using R - Part II

Lecture-02 Association Rules-Part II



With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

Welcome to the course Business Analytics and Data Mining Modeling Using R Part 2, so as we talked about in the previous lecture, the first lecture that this is continuation of a previous course on business analytics and data mining modeling using R.

Association Rules

- Also called
 - Affinity Analysis
 - Market Basket Analysis
 - Due to its origin from the studies of customer purchase transactions databases
- Main Idea is
 - To identify item associations in transaction-type databases and
 - Formulate probabilistic association rules for the same
 - “what goes with what”



So in the previous lecture we started about discussion on association rules, so let's continue that. So we talked about the basic idea of association rules and you know what we try to achieve using this particular technique, so that part were discussed in the previous lecture.

Association Rules

- Market Basket databases
 - Large no. of transaction records
 - Each record consists of all the items purchased by a customer in a single transaction
- If we can find item groups which are consistently purchased together, such info could be used for
 - Store layouts, cross selling, promotions, catalog design, and customer segmentation



We've talked about the market basket databases, we talked about how this particular technique can be applied in various, for various business challenges, analytics challenges, so all that discussion was done we talked about association rules and the if then statements that are used to formulate these rules to express these rules, we talked about the two stage process that is used in association rules, first one rule generation, second one assessment of rule strength.

Association Rules

- Association rules
 - “if-then” statements computed from data
 - Example: online recommendation systems or recommender systems in online shopping websites of e-commerce companies like Amazon, Flipkart, and Snapdeal
- Two-stage process
 - Rule generation
 - Apriori Algorithm
 - Assessment of rule strength



Association Rules

- Example: mobile phone cover purchase
 - What colors of covers customers are likely to purchase together?
 - Database of ten transactions
 - Open RStudio
- Candidate Rules generation
 - Examine all possible rules between items in “if-then” format
 - Select rules which are most likely to capture the true association



We also started our discussion on rule generation part of it, and few things that we were able to discuss, we used this particular example mobile phone cover purchased, and we talked about

Association Rules

- “If-then” format
 - “If” part is called antecedent
 - “then” part is called consequent
- Antecedent and consequent are
 - Disjoint sets of items or item sets
 - Example: mobile phone cover purchase
 - “if red then white”If red cover is purchased, a white cover is also purchased



candidate rules generation and few things, and we were able to discuss the, we talked about

Association Rules

- Antecedent and consequent
 - Example: “if red and white then green”
- Rule generation
 - No. of distinct items in a database = p
 - In mobile phone cover purchase example, $p=6$
 - All possible combinations
 - Single items, pairs of items, triplets of items, and so on
 - High computation time

Association Rules

- Rule generation
 - Look for high frequency combinations
 - Called frequent item sets
- Define frequent item set
 - ‘concept of support’
 - Support of a rule is
 - No. of transactions with both antecedent and consequent item sets
 - Measures the degree of support the data provides for the validity of the rule
 - Expressed as a percentage of total records

antecedent and consequent item sets. Then we discussed few things about rule generation and we talked about the problems that are associated with the steps, and then we talked about the importance of frequent item set and how it could be used for, you know, next steps, so then we talked about the concept of support which is again associated with this, actually required for us to create these frequent item set. Then we ended our discussion, you know at this point and so let's carry forward this discussion, the next thing that we are going to discuss is Apriori algorithm, so this is the classic algorithm given by Agarwal which is typically used for association rules.

Association Rules

- Apriori Algorithm
 - Generate frequent item sets with one-item sets
 - Compute support for one-item sets
 - Drop the sets having support below user specified minimum support
 - Remaining sets are the frequent one-item sets
 - Recursively generate frequent item sets with two items
 - Use frequent one-item sets to generate two-item sets
 - Since larger size item sets containing non-frequent one-item sets will also be non-frequent item sets
 - Compute support for two-item sets
 - Drop the sets having support below user specified minimum support
 - Remaining sets are the frequent two-item sets



So we are talking about the rule generation process, rule generation stage of this association rules process, so Apriori algorithm if we look at the overall major steps then the first step is going to be a generate frequent item sets with one item sets. Then recursively generate frequent item sets with two item sets, then we move to the same thing with 3 items, and so one for all sizes, so this process is to be done for, you know, item sets with you know all sizes.

So now let's look at the first step in a bit more detail, so generate frequent item sets with one item sets, so it is easier to you know identify the item sets you know one item having just one item, so once we are able to list all those item sets which are having just the single item, then we can you know compute support for these one item sets, so support how do we compute that we have already talked about, let's revisit it again, so we talked about the support of rule being number of transaction with both antecedent and consequent item sets, so in this fashion we can compute the support value for all the item set which are having just the single item.

Now as we talked about that, you know user has to specify a minimum support level that is required for us to you know select a you know to consider a item set as a frequent item set, so we need to drop them sets having support below that user specified minimum support, once that is done the remaining sets can be called as frequent one item sets, so it is the user specified minimum support level which gives us the criteria for deciding whether a one item set is going to be consider as frequent you know one item set or not, so once this process is completed then we'll end up with the frequent item sets with the you know, just one item. So this process as we talked about the second step is again in the second steps where two items are involved, so we are dealing with item sets having two items, so here in this case we start with what we have already created, so we have already created frequent item sets with one item, so we start with that list, and use that while creating, while generating frequent item sets with two items, so as you can see in the first you know sub step in this is used frequent one item sets to generate two item sets, so why this is done?

So as it is noted here, since larger size item sets containing non-frequent one item set, sets will also be non-frequent item sets, so if you know, if we have a non-frequent item set having just the single item and it is included, it is used to create an item set with the two items then of

course it is also going to be the non-frequent, so therefore if we just work on the list of frequent one item sets then that would actually lead us to frequent two item sets, so this is why we have, we start with frequent one item sets.

Now once we have, once we have listed all the, you know all these two item sets then we can compute support for them, so these two item sets as I talked about have to be generated using the frequent one item sets, so once these two item sets are there, then we can compute support for them and then drop the sets having support below user specified minimum support, so once that is done the remaining sets they would be the frequent two item sets, so if you look at the process it is easy to understand what we are going to do in the next step, so to you know review these two steps again, so first we start with, in the first step we try to create frequent item sets having just single item, so these are one item sets, so easier to identify these one item sets then we compute support, then we dropped those one item sets which are having support less than user specified minimum level, and then the remaining sets are used for the next step.

Now in the next step you know the item sets with the two items we only considers, we only consider two item sets which are based on frequent one item sets, so once again we compute supports and then drop the one's, to drop the one's having support below or minimum user specified minimum level, and then the remaining set would serve as frequent 2 item set, and the same process is repeated with 3 items, and item sets with 3 items and more, so if we look at the main idea here is we typically generate K item sets using frequent k-1 item sets, so this actually allows us, allows our algorithm to be computationally you know cheaper, in the sense the algorithm, Apriori algorithm is quite fast because we have to go through just a one pass through the database, so as you can see this point is noted here, so Apriori algorithm is quite fast even for a large number of unique items, so each step requires a single run through the database.

Association Rules

- Apriori Algorithm
 - To generate k-item sets, use frequent (k-1)-item sets
 - Then with three items
 - And so on for all sizes
- Apriori algorithm is quite fast even for a large no. of unique items
 - Each step requires a single run through the database

So now to again look at, look back at this Apriori algorithm, so we start, we are trying to, first we try to create you know item sets, frequent item sets with one item, then use this list to create frequent item sets with 2 items, then use this list to create frequent item sets with 3 items and then so on, so in this process, in this fashion we are able to reduce the number of computational

steps and the algorithm is quite fast, so this is the Apriori algorithm that is used for rule generation.

Association Rules

- Assessing rule strength
 - Idea is to identify rules which capture strong association between antecedent and consequent item sets
 - Metrics to measure strength of this association as implied by a rule
 - Confidence
 - Ratio of no. of transactions with antecedent and consequent item sets to the no. of transactions with antecedent item set
 - Lift ratio



So let's move forward, so next stage of the process is assessing rule strength, so how do we determine, how do we compute rule strength? Now we look at the main idea, so as you can see the main idea is to identify rules which capture a strong association between antecedent and consequent item sets, so that is the main idea, so we are interested in finding or capturing the strong association between these two item sets antecedent and consequent item sets, because as we have talked about our rules go like this that if X has been purchased then Y is also going to be purchased, so here if you see that there has to be some sort of association between item set X and item set you know Y, so therefore our assessment of rule strength should be able to capture this strength, so matrix that can be used to measure this, measure the strength of disassociation as implied by rule, so first one is confidence, so what we mean by this matrix? So confidence is ratio of number of transactions with antecedent and consequent item sets to the number of transactions with the antecedent item set, so if we really look at this formula so from the group of you know within the transaction which are having antecedent item set, we are looking to find out the proportion of these transactions which are also having the consequent item sets.

So we focus on the transaction which are having the antecedent item sets, and then we try to identify, they try to compute the proportion which are also having the consequent item sets, so this value gives us the confidence you know number, so as you can see that through confidence for a given rule if the confidence is higher than probably the strength of the association between antecedent and consequent is also on the higher side, because if the confidence value is higher than the appearance of or co-occurrence of antecedent and consequent item set is on the higher side that means whenever there are more chances that whenever antecedent item set is present, the consequent item set is also going to be present, so we can see if the confidence value is higher than the strength of this association is also going to be on the higher side, so this is one matrix that is used, then other matrix is lift ratio, so lift ratio is typically as we have covered in the previous course, typically we compare it with you know average case or bench mark scenario or case, and then see how much the lift is being provided by the model, so in this

particular technique also it has the same sense, same meaning, so we'll discuss this in later in this particular lecture.

Association Rules

- Revisit Support and Confidence
 - Support as $P(\text{antecedent and consequent})$
 - Confidence as $\frac{P(\text{antecedent and consequent})}{P(\text{antecedent})}$
or $P(\text{antecedent} | \text{consequent})$
- Typically, high value of confidence means strong association rule
 - It might fail in cases where antecedent and/or consequent item sets have high support leading to high confidence despite no real association



Now at this point let's revisit the two important matrix that we have talked about support and confidence, so if we you know support can also be thought of as probability of antecedent and you know consequent items you know occurrence of antecedent and consequent items among the transactions in the database, so we can also think about support as this probability value.

Similarly if we look at the confidence, so confidence can also be thought of as probability of antecedent and consequent item sets divided by probability of antecedent item sets, so in this we can also think about these matrix support and confidence in probability terms.

If we look at the confidence, once we think about confidence in terms of probability we can see that confidence can also be rewritten as the conditional probability where we are trying to compute the probability of you know, probability of antecedent given the consequent, so typically high value of confidence as we talk about, high value of confidence means strong association rules, so however this might not be true in all the cases because in some cases where antecedent or consequent item sets have high support, it might lead to high confidence despite no real association, for example if you have, for example if you have you know some transactions where the customers almost all the transactions involved it and almost all the transaction involved some snack, so both these items will have high support and therefore we're also end up having high confidence value, so there might be cases you know where you know particular you know antecedent item set and consequent item set they have high support and that might be leading to high confidence despite you know there is, despite no you know clear association between the items. So typically we can rely on this high value of confidence, meaning strong association however there might be cases where this might not be true.

Now let's talk about the next metric so as a lift ratio, so as we talk about we try to, as we have discussed in previous course as well that typically we compare, with the benchmark scenario or average case scenario, so here also lift ratio compares the confidence of a rule with its benchmark value.

Association Rules

- Lift ratio

- Compare the confidence of a rule with its benchmark value
- Benchmark value of a rule is confidence value computed by assuming no association between antecedent and consequent item sets

- We assume that antecedent and consequent item sets occur independently

$$P(\text{antecedent and consequent}) = P(\text{antecedent}) \times P(\text{consequent})$$

$$\text{Confidence (benchmark)} = \frac{P(\text{antecedent}) \times P(\text{consequent})}{P(\text{antecedent})} = P(\text{consequent})$$

$$\text{Benchmark confidence} = \frac{\text{No. of transactions with consequent item set}}{\text{total no. of transactions}}$$



Now what do we mean by benchmark value in this case, so you can see the next point is the benchmark value of a rule is confidence value computed by assuming no association between antecedent and consequent item sets, so when we assume this then we compute the confidence value and this confidence value is treated as a benchmark value, and then we compare it with the confidence of the rule and we get the ratio, so let's look at this particular process and more detail, so we assume that antecedent and consequent item sets occur independently, this assumption is to compute the benchmark value. So if this assumption is there then the probability of antecedent and consequent item sets occurring can also be rewritten as probability of antecedent multiplied by probability of consequent, so if this is true then we can plug this particular you know, expression in this confidence formula and we can compute the confidence in the benchmark case, so you would see that in the numerator part of it will have the probability of antecedent multiplied by probability of consequent, and then this whole expression is divided by probability of antecedents, so and then we'll get the probability of antecedent, so you would see that confidence, the benchmark value of confidence is nothing but the probability of antecedent happening. Now this benchmark confidence we can rewrite as number of transactions with consequent item set divided by total number of transactions. Let's revisit the concept of support and confidence the two metrics that we talked about in probability terms, so support can also be thought of as probability of occurrence of antecedent and consequent item sets, similarly confidence can also be thought of as probability of occurrence of antecedent and consequent item sets, in the numerator divided by probability of antecedent item set, so this can also be rewritten as the probability of conditional probability of consequent given the antecedent item set, so in this fashion we can rethink these two metrics, support and confidence in probability terms, and also we can think about that probability of in all the transactions that we have in the database, support can be thought of as the you know occurrence of item sets belonging to antecedent and consequent in the whole database among all the transactions.

Similarly confidence can be thought of as probability of occurrence of antecedent and consequent item sets among the antecedent item sets, so we can always think about these metric as support and confidence.

Now as we talked about that typically high value of confidence means a strong association but sometimes it might not be true, so sometimes the antecedent or consequent item sets they might have higher support and which might lead to higher value for confidence, so even though there is no real association. For example if there are you know two items that are being purchased by most of the customers from a café and those two items don't have any association, but because these two items are frequently purchased by most of the customers so they will have high support and therefore the confidence valuable also be on the higher side, so there are some situations where the typically high value of confidence might not be the strong association.

The next metric that we are going to discuss is lift ratio, so as we talked about in the previous course also lift ratio has been used in many techniques where we always try to compare the performance of the model with the benchmark case or the average scenario case, so similarly here also lift ratio is about comparison of confidence of a rule with its benchmark value.

Now what do we mean by benchmark value in this case? So as you can see in this, and the point that is mentioned in this particular slide, the benchmark value of a rule is a confidence value computed by assuming no association between antecedent and consequent item sets, so here to compute the benchmark value we are assuming that antecedent and consequent item sets they occur independently, so what this means is that the probability of occurrence of antecedent and consequent item sets can also be rewritten as the probability of antecedent item set occurring multiplied by probability of occurrence of consequent item sets, this is just like the independent events probability, two events are considered to be independent, there probability can be computed as the you know, we can multiply the probability of individual events occurring, right, so in this fashion the confidence, the benchmark confidence value can be rewritten, you can see the numerator part, here the numerator changes to P, probability of antecedent multiplied by probability of consequent, and divided by probability of antecedent, so you can see that it will result into the probability of consequent. So you can see that the benchmark confidence value essentially becomes the probability of occurrence of consequent item sets, so we can also rewrite this particular expression benchmark confidence as, in the numerator number of transactions with consequent item set divided by total number of transaction, so in this fashion we can use this particular formula and compute the benchmark confidence value, and then it could be used later on to compute the lift ratio.

Association Rules

- Lift ratio of a rule is defined as:

$$\frac{\text{Confidence}}{\text{Benchmark Confidence}}$$

- Usefulness of a rule
 - Lift ratio > 1.0
 - Larger the value, greater the strength of the association

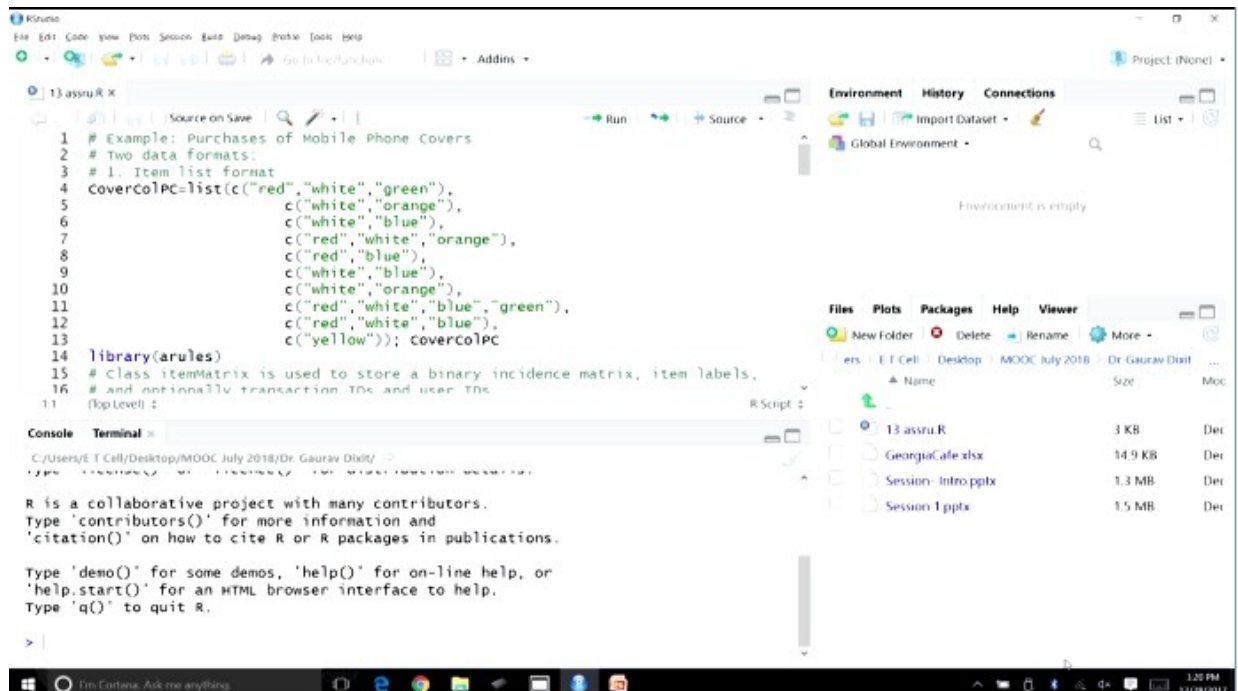
So how do we define lift ratio? So as you can see here lift ratio can be computed using this formula, so in the numerator we have the confidence that is going to be divided by the benchmark confidence value, how do we compute benchmark confidence? We have already discussed confidence formula we already understand, so in this fashion lift ratio we can compute.

Now how this lift ratio can be used to find out the usefulness of a rule, so as you can see here if the lift ratio is greater than 1, that means the confidence of a rule is greater than its value, the value of benchmark is, value of benchmark scenario where we assume that there is no association between antecedent and consequent item set, so lift ratio value of greater than 1 means that in comparison to the scenario where antecedent and consequent item sets are assumed to be independent, the confidence value is higher than that. So typically larger the value, larger the lift ratio value, greater the strength of the association, however lift ratio is able to overcome that problem that we talked about, the problem related to confidence formula where the high value of confidence typically means the stronger association or stronger association between those sets antecedent and consequent item sets and we talked about that in some scenarios it might not be though we had given the example of you know two snacks being purchased from a café, so that problem can be overcome using the lift ratio metric where we are comparing with the benchmark confidence, so this seems to be a much better metric in terms of judging the strength of association rules.

Association Rules

- Transaction Data formats
 - Item list format
 - Each row contains a list of purchased items and represents a transaction
 - Binary matrix format
 - Rows represent transactions
 - Columns represent items
 - Cells have either a 1 or a 0 indicating presence or absence of the item in the transaction
 - Open RStudio

Now let's discuss the transaction data formats, so there are two popular data formats that are available to store the, or to store the transaction data, transactions of the database, so first one is called item list format, so in this item list format each row contains a list of purchased items and represents a transaction, so all the rows represent the particular transaction and each row will contain list of the items where essentially item names, so this is called the item list format. The second format is binary matrix format, in this particular format rows represent transactions, so again rows are still representing transaction, but the column represents items. So now when the column represents items and rows are representing transactions the cells can have either 1 or 0 value that would indicate the presence or absence of that particular item in the transactions, so for a typical you know any column a particular column would represent you know a particular you know item, and if the value for that cell in a particular row, in a particular transaction is 1 it will mean the presence of it, otherwise if the value is 0 then it will mean the absence of it, so this is the binary matrix format, so these are the two formats that are typically popular and used in the association rules mining and related discussion.

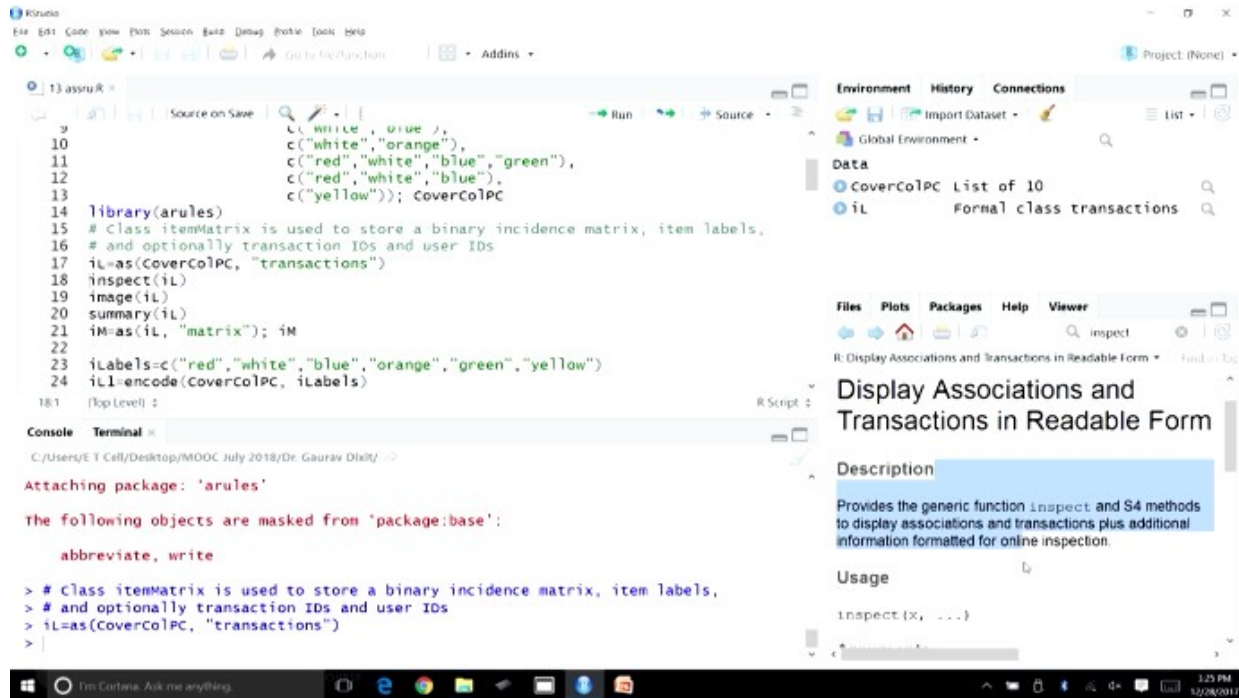


So let's you know, in the previous lecture we had talked about the mobile phone cover example, so let's go back to that example, let's open R studio and let's discuss these two formats, so let's revisit this example, so here as we can see that the example purchase of mobile phone covers, we talked about this particular example, we have 10 transactions here, so you can see the first item list format here, so you can see the item list format here and we can also, you can also the 10 transactions that we talked about in the previous lecture as well, so this is essentially the item list format you can see, right now we have created it you know using the list in R, and this can be executed, let's run this, now you can see from here, that you can see this list has been created, you can see each row is representing a transaction and the item names are given, red, white, green for the first transaction, for the second transaction white, orange. The third transaction white, blue, so in this fashion for all of these transaction, total 10 transactions we can see the item names are given, so in this case item names are being represented by the colors, because as we talked about this particular example is about, if a customer is purchasing mobile phone cover which you know, which colors they are more likely to purchase together, right, so in a particular transaction we can see the covers with different colors that are being purchased in a single transactions, so this is the item list format.

To look at the binary matrix format we'll have to use this particular package, library rules, so let's load this particular package, so right now this is, this package is not installed in the system so let's install this, this is the command that we are going to use is, is the install.packages, and we will use the package name in double codes A rules, enter, so this package is being installed, so A rules this is the most popular package that is available in all, available for our environment, so this is used for association rules modeling, so once this is, once this is installed and loaded we'll be able to create this binary matrix, you know, format for our data set, this mobile phone purchase data set.

Now this installation is almost complete, now let's load this, so we'll go back to our code, let's run this, you can see the package has been loaded now, so now as we talked about the class item matrix is part of this particular package and this can be used to store a binary incidence matrix, item labels, and optionally transaction ID's and user ID's, so this particular class that is part of

this package A rules can be used to store data in various formats, so right now we are interested in you know creating the binary incidence matrix, so let's run this, so as you can see this list that we have already created it has the all the 10 transaction in the list format, now we are going to use this as function which is going to you know covers this list into the transaction format, so let's run this.

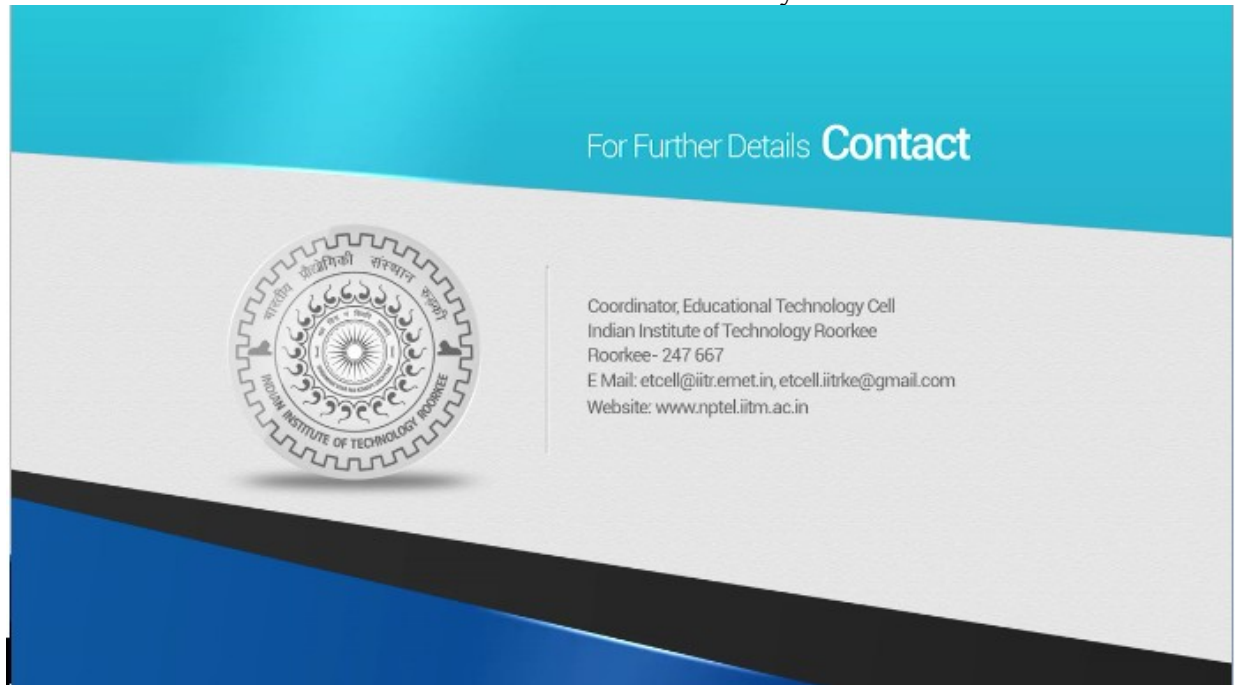


Now inspect the function that can be used to actually look at this particular, you know, this particular format, so more detail about the inspect function you can find out from the help section, so here again you can type in the help section inspect and you would be able to find more detail about this particular function. So as you can see provides the generic function inspects for you know to display association and transactions plus additional information formatted for online inspection, so this is the function that we are going to use here, so let's run this particular code, and you can see here or the transactions listed in these braces, so items you can see first transaction green, red, white, second one orange, white, so all these are you know being us own in curly braces, so each row is representing a transaction and the name of item, so this is essentially the item list format. So now we can also display the same information in image format, so the function that we are going to use is image, so let's run this, so you can see in the plot section we have on the row side we have items, so as we talked about in the previous lecture in this particular database we have just 6 distinct items, so you can see 6 you know columns here, and each row is representing a transaction, so these colors are indicating the presence or absence of that particular item in that transaction, so you can see here that for example transaction number 1, we have item 2, item 4 and 5 present, right, so in this fashion you know we can have a visual representation of this particular database, this particular transaction database as well.


Then we can use summary function as well to find out a bit more detail about this, now this particular, once this, we have this transaction format with us we can create a matrix format so you can see as function is being used and how the format is matrix as you can see in the second argument, matrix is being passed over here, so let's run this and we'll get the matrix format. Now let's look at the output, as you can see that you know the column names are being

represented by the colors, different colors for the, you know covers, mobile covers you can see blue, green, orange, red, white, yellow, here you would also notice that these colors are, right now these item names are arranged in the alphabetical order, you can see blue comes first, then green, and orange, so they're arranged in alphabetical order, how about you might want a different order, so that can also be done.

So for this we'll have to create these labels for these items, so you can see the next line of code, you can see, this is the order that we want, first red, then white, then blue, orange, which is not necessarily the, we might not want you know the alphabetical order sometimes, so in this fashion we will have to create the labels and then we can use in code function to change the names of column labels, right, so in this fashion we can move forward, so let's run this and then in code function can be used to change these label names, and you would see we have run the inspect of this new variable, and you can see this is right now, this is you can run the image function, if you can say here now we can create our new matrix, now if you scroll you can see the column names the order has changed now depending on the order that you want, you can create in this fashion, and this is the binary matrix format that we have, so two matrix format, item list format and binary matrix format that we have been able to cover, so we'll stop here, and we'll continue our discussion in the next lecture. Thank you.



For Further Details **Contact**



Coordinator, Educational Technology Cell
Indian Institute of Technology Roorkee
Roorkee - 247 667
E Mail: etcell@iitr.ernet.in, etcell.iitrke@gmail.com
Website: www.nptel.iitm.ac.in

For Further Details Contact
Coordinator Educational Technology Cell
Indian Institute of Technology Roorkee
Roorkee – 247 667
E Mail: etcell@iitr.ernet.in, iitrke@gmail.com
Website: www.nptel.iitm.ac.in

Acknowledgement

Prof. Ajit Kumar Chaturvedi
Director, IIT Roorkee
NPTEL Coordinator
IIT Roorkee

Prof. B. K Gandhi

Subject Expert

Dr. Gaurav Dixit

Department of Management Studies

IIT Roorkee

Produced by

Mohan Raj.S

Graphics

Binoy V.P

Web Team

Dr. Nibedita Bisoyi

Neetesh Kumar

Jitender Kumar

Vivek Kumar

Dharamveer Singh

Gaurav Kumar

An educational Technology cell

IIT Roorkee Production

© Copyright All Rights Reserved

WANT TO SEE MORE LIKE THIS

SUBSCRIBE