Data:

One is from CMS member database

another is CMS provider database

3rd is dataworld.com

compiled all of it and we have calculated **PMPM using claims data**.

we are preprocessing the data.

So the way we are preprocessing it for ARM and subgroup is little different.

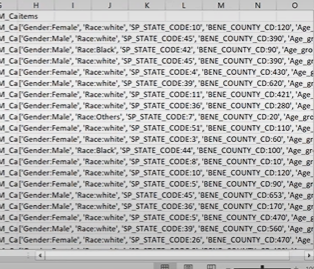
association rule mining :

treats **each of the category of each variable as a separate entity** like because it was originally developed to deal in a Market Basket kind of

there is nothing called as variable concept in this algorithm.

**We have to give item sets**.

So it is like **gender, female race white**.

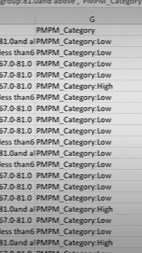


So this way we had to bring the data into because otherwise we had data like gender, male, female like this but then so for gender you can understand but **for other categories it was like difficult to identify what are the.**

r example for **disability indicator and all, yes no.** But for what? Yes, what? No. So we had to have.

this **variable name followed by the category**.

So to bring it in this picture in this way, uh is what we have to do in the preprocessing step. We have **done this for all features that we have say for state code, county code** and then we are whatever information is available in different columns. **We are storing it and also PMPM category and we have calculated PMPM category using quantiles.**



After PMPM numeric PMPM that **we have calculated so then we are storing it as an item**

\code how we did the preprocessing.

config file for perform plus

original data set path, then a path for storing preprocessed data.

And then **we are storing uh rules that we are obtaining from ARM in another part**.

We are giving cohorts and all and then which objects to change, how to rename and so on will be stored in config file then in preprocessing what I showed you. **So our preprocessing code will bring the data in this format**

written multiple uh functions, which we'll do this preprocessing.

CMS data gender was in coded format one and two, we are first bringing it into a categorical format. Similarly for this age group we are **creating age groups** then for **disease codes** we are making it instead of 01 yes no. And then we are **preparing cohort according to the cohort listed in the config file.**

So if you want to run your code for some other disease cohort you'll have to change the cohort in the config file and then run the code as it is.

So then the preprocessing will happen like changing of data types and then categories and all renaming the categories. Then we are creating PMPM categorical variable.

beginning we started by experimenting with high, medium and low.

However, we were getting many rules and most of them were like **it was difficult to arrive at a decision which one of them are more important because they like statistically** they had higher performance in terms of confidence and.

we decided to go with high end low PMPM categories only. OK then we are creating the data in the format like joining the.

So first we are **dropping those columns which are not associated with our current cohort**

we are doing feature selection using feature with algorithm

it will give you top features which are contributing so that we can utilize only those features to run the algorithm.

Then we are **recording the columns to make output of arm more understandable** and then we are **dropping any values to make columns of lists where each list is 1 transaction**. So because it is like it began with Market Basket analysis you can see.

This **what we have stored in a list format is essentially 1 transaction** or it is like in one go water what all occur together is what we are giving to the algorithm so that is why we are bringing it into a transaction kind of set.

So this **we did from FP growth package**. So since we are using **FP growth algorithm to do this to get the association rules.**

**\**  
FP growth package itself, we have uh renamed it. Or uh, repurposed it in a way that will be helpful for us. And that's what we are utilizing here. Then we are preparing the data.

, feature selection Recode. So here we were writing the codes for doing all these steps and then prepared data will so.

0:8:2.350 --> 0:8:4.450  
Rutuja Kore  
Uh only prepared data will be.

0:8:5.210 --> 0:8:6.560  
Rutuja Kore  
Uh, call.

0:8:7.550 --> 0:8:8.500  
Uma Maheswari K  
The main function.

0:8:5.540 --> 0:8:9.740  
Varsha Khamitkar  
So here main function is prepared data is it?

0:8:9.320 --> 0:8:9.900  
Uma Maheswari K  
Yeah.

0:8:10.790 --> 0:8:12.690  
Uma Maheswari K  
They'll be calling each and every function.

0:8:9.720 --> 0:8:12.820  
Rutuja Kore  
Yes, correct. And all other functions I told.

0:8:13.730 --> 0:8:16.860  
Varsha Khamitkar  
OK. All other other supporting ones.

0:8:17.700 --> 0:8:18.60  
Varsha Khamitkar  
OK.

0:8:17.210 --> 0:8:18.270  
Rutuja Kore  
Correct, correct.

0:8:21.620 --> 0:8:22.120  
Uma Maheswari K  
OK.

0:8:21.740 --> 0:8:22.350  
Varsha Khamitkar  
Umm.

0:8:19.180 --> 0:8:24.50  
Rutuja Kore  
So which will be part of prepare data, so those will be called in this function.

0:8:25.160 --> 0:8:25.580  
Uma Maheswari K  
OK.

0:8:25.220 --> 0:8:25.670  
Varsha Khamitkar  
OK.

0:8:26.290 --> 0:8:29.590  
Rutuja Kore  
And then the preprocessed data will be exported.

0:8:30.550 --> 0:8:31.600  
Rutuja Kore  
In this format.

0:8:32.680 --> 0:8:33.140  
Varsha Khamitkar  
Hmm.

0:8:36.590 --> 0:8:37.340  
Uma Maheswari K  
OK.

0:8:36.920 --> 0:8:37.450  
Varsha Khamitkar  
OK.

0:8:33.870 --> 0:8:48.900  
Rutuja Kore  
At the location given in the config file. OK then then it is pretty simple. So since we had a huge data because you can see that each category of each variable is treated as a separate entity.

0:8:49.770 --> 0:8:50.200  
Varsha Khamitkar  
Hmm.

0:8:49.460 --> 0:9:20.110  
Rutuja Kore  
Ohh even if I I have considered only four five features here, I'm getting a lot of combinations, permutations and combinations. That was the reason why we eventually dropped off a lot of columns because it was computationally difficult and it was like more difficult to interpret. That is why we drop them and that's why you don't have it here. Even though we had it in the original data, we did multiple experiments but finally to show the proof of.

0:9:32.610 --> 0:9:32.940  
Varsha Khamitkar  
Mm-hmm.

0:9:20.190 --> 0:9:40.40  
Rutuja Kore  
Concept we thought, OK, we'll take only demographic for five features and since according to the perform plus Team, State and county were more important we we kept them and we applied filter on the basis of disease and that is why I remember now we are dropping the disease related columns here.

0:9:42.240 --> 0:9:44.970  
Rutuja Kore  
OK, so uh, so that it?

0:9:43.770 --> 0:9:45.390  
Varsha Khamitkar  
Ohh sorry.

0:9:45.890 --> 0:9:46.430  
Rutuja Kore  
Yes.

0:9:46.140 --> 0:9:46.690  
Varsha Khamitkar  
Mm-hmm.

0:9:47.500 --> 0:9:55.990  
Varsha Khamitkar  
So you are applying on the disease column and that's why we have a dropped the diabetic and other one feature is it.

0:10:19.800 --> 0:10:20.370  
Varsha Khamitkar  
Yeah.

0:9:56.490 --> 0:10:27.320  
Rutuja Kore  
Yes, yes and uh, we could have kept other diseases as well, but then the information that they were adding in comparison with the computational expensive city, it was adding two. So that is why we thought, OK, we'll drop it as of now actually the problem is that since association rule, mining is very computationally expensive. Our machines used to hang. So I had tried on data bricks as well and then multiple.

0:10:32.940 --> 0:10:33.710  
Varsha Khamitkar  
Hmm.

0:10:44.320 --> 0:10:44.690  
Varsha Khamitkar  
Umm.

0:10:27.530 --> 0:10:50.100  
Rutuja Kore  
Different options we had explored and finally we decided to use Pyspark where things will be like we can use multiple cores at a time and then we have done it like we imported pie growth from PYSPARK but eventually I I I would tell you this will be difficult like.

0:10:57.800 --> 0:10:58.380  
Varsha Khamitkar  
OK.

0:10:50.880 --> 0:11:4.870  
Rutuja Kore  
Like my machines configuration was decent. I had 16GB RAM and all but still it was failing. It worked only on Databricks because it was like freely available. We utilized it but you would need a.

0:11:14.880 --> 0:11:15.570  
Varsha Khamitkar  
OK.

0:11:5.990 --> 0:11:34.440  
Rutuja Kore  
For database like server to do this it will be then only it is efficient otherwise it is very difficult to do it with local machine. So here essentially we are just calling the data then the algorithm will be applied FP growth, minimum support and minimum confidence you can change so you can see I've kept minimum confidence .5 so all the rules which have confidence about .5 only those will be considered.

0:12:4.800 --> 0:12:5.130  
Varsha Khamitkar  
Umm.

0:11:36.310 --> 0:12:6.350  
Rutuja Kore  
Then support is for each individual item, so it is like the support will. What is the support for gender females, gender, male race, white race, Black County Code 90 and so on. So and then the support for rule is also calculated. So in order to give a chance to even like less frequent observations we have kept minimum support a 0.01 you can experiment with it you can.

0:12:6.480 --> 0:12:11.420  
Rutuja Kore  
Try changing it according to the need at the hand at when you will run it.

0:12:12.60 --> 0:12:13.280  
Rutuja Kore  
Uh, and then?

0:12:13.140 --> 0:12:13.600  
Varsha Khamitkar  
Welcome.

0:12:14.390 --> 0:12:20.700  
Rutuja Kore  
These rules are stored generated rules and then they are exported in CSV format.

0:12:21.760 --> 0:12:43.270  
Rutuja Kore  
So this is all about ARM, like generating the rules. Then there are multiple ways in which you can visualize the rules. But in Python there are not many options available. I had explored a few like one of them I had shown in the like best of them was shown in the PPT that we discussed last week.

0:12:45.610 --> 0:12:48.80  
Rutuja Kore  
In the kedari which we had presented, but then.

0:13:3.250 --> 0:13:3.680  
Varsha Khamitkar  
OK.

0:12:50.380 --> 0:13:7.120  
Rutuja Kore  
If you have lesser number of rules you can get better visualizations so that experiments I have stored, But I'll I'll look into the notebook like my folder and I'll share them with you. So this was about association rule mining. Is it OK till here?

0:13:6.310 --> 0:13:16.190  
Varsha Khamitkar  
The result of the uh result sheet is not there in this code. Is it like like a final rule set you have stored it into the CSV know?

0:13:16.810 --> 0:13:19.960  
Rutuja Kore  
Yes it is. It should be here. Just a minute. I'll see.

0:13:36.130 --> 0:13:40.720  
Rutuja Kore  
OK, I have it I it is not in what I have shared with you just.

0:13:47.450 --> 0:13:47.910  
Varsha Khamitkar  
OK.

0:13:44.670 --> 0:13:47.940  
Rutuja Kore  
I look for it and I'll share it with you. I have it with me.

0:13:49.140 --> 0:13:49.790  
Varsha Khamitkar  
OK, sure.

0:13:50.950 --> 0:13:52.610  
Rutuja Kore  
It is there, but it is like.

0:13:51.680 --> 0:13:58.40  
Varsha Khamitkar  
They're just doing this to put it in here and you can share the updated ZIP kind.

0:14:5.460 --> 0:14:7.320  
Varsha Khamitkar  
OK. Yeah.

0:14:15.740 --> 0:14:16.200  
Varsha Khamitkar