

## Association Rule Mining Process PDF Report

In order to better understand the patterns and relationships within our data, we used the analysis services multidimensional project template on Visual Studio 2019 to assist us in what's called association rule mining. This allowed us to uncover interesting connections between different types of criminal activity and their relationships to each other. Specifically, we have identified k rules that help to explain the relationships between certain types of crimes. These rules provide valuable insight into the behavior of criminals in our city and can help to inform future crime prevention strategies.

K-Rules is to a data mining technique used to discover links in data and is a type of association rule that uncovers connections between different attributes in a dataset, based on the frequency of their occurrence. It basically tries to find patterns of the form "if X and Y occur together in a certain percentage of cases, then Z is also likely to occur with X and Y." The "K" in K-Rules refers to the maximum number of items that can be present in the antecedent (the "if" part) of each rule.

Now that we have a better understanding of what association rule mining is, let's explore the steps involved in this process.

To do that, we first need to create 2 views on our SSMS, one case and one nested.

- Case view: It should have a primary key and foreign keys, all of unique values.
- Nested view: It should not have a primary key and can have duplicate values, and this in this view there should also be text columns, that are meaningful to the analysis. For example, how crime types include homicide and larceny which can be used to label and visualize the data.

It is important to note that both case and nested views should have the same foreign keys. Below is the SQL query for the case table that I used for the association rule mining process.

```
:setvar ProjectSQL "C:\Users\fredd\Documents\Uni\CITS3401 Data Warehousing\Project\"
:setvar DatabaseName "CrimeIncidents"

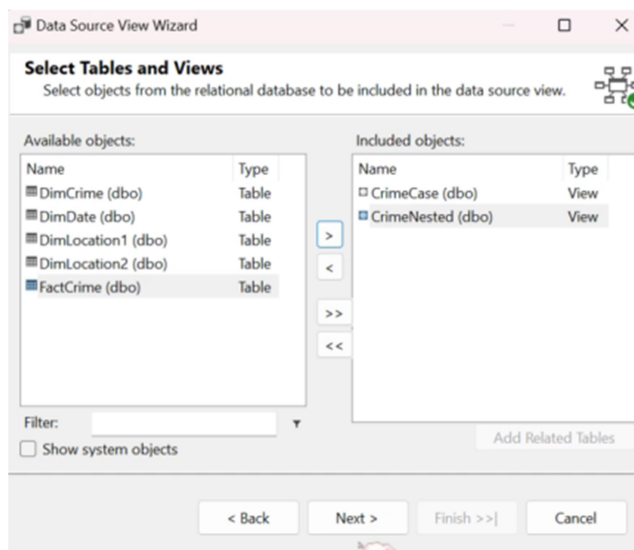
CREATE VIEW [dbo].[CrimeCase]
AS
SELECT DISTINCT
    [dbo].[FactCrime].id AS id,
    [dbo].[DimCrime].crime AS crime,
    [dbo].[FactCrime].crimekey AS crime_key,
    [dbo].[FactCrime].datekey AS date_key,
    [dbo].[FactCrime].loc1key AS loc1_key,
    [dbo].[FactCrime].loc2key AS loc2_key
FROM [dbo].[FactCrime]
INNER JOIN [dbo].[DimCrime] ON [dbo].[FactCrime].crimekey = [dbo].[DimCrime].crimekey
INNER JOIN [dbo].[DimDate] ON [dbo].[FactCrime].datekey = [dbo].[DimDate].datekey
INNER JOIN [dbo].[DimLocation2] ON [dbo].[FactCrime].loc2key = [dbo].[DimLocation2].loc2key
INNER JOIN [dbo].[DimLocation1] ON [dbo].[FactCrime].loc1key = [dbo].[DimLocation1].loc1key;
GO
```

In the code above, there are a few important things to note. The setvar function must still be there and the query should be set on SQLCMD mode, and there should be a DISTINCT statement after the SELECT statement so that it returns unique rows. The inner join statements are to join the fact table with the dim tables so that we have all our data.

```
CREATE VIEW [dbo].[CrimeNested]
AS
SELECT
    [dbo].[DimCrime].crimekey AS crime_key,
    [dbo].[DimCrime].crime AS crime,
    [dbo].[DimLocation1].neighborhood AS neighborhood,
    [dbo].[FactCrime].datekey AS date_key,
    [dbo].[FactCrime].loc1key AS loc1_key,
    [dbo].[FactCrime].loc2key AS loc2_key
FROM [dbo].[FactCrime]
INNER JOIN [dbo].[DimDate] ON [dbo].[FactCrime].datekey = [dbo].[DimDate].datekey
INNER JOIN [dbo].[DimLocation1] ON [dbo].[FactCrime].loc1key = [dbo].[DimLocation1].loc1key
INNER JOIN [dbo].[DimLocation2] ON [dbo].[FactCrime].loc2key = [dbo].[DimLocation2].loc2key
INNER JOIN [dbo].[DimCrime] ON [dbo].[FactCrime].crimekey = [dbo].[DimCrime].crimekey;
GO
```

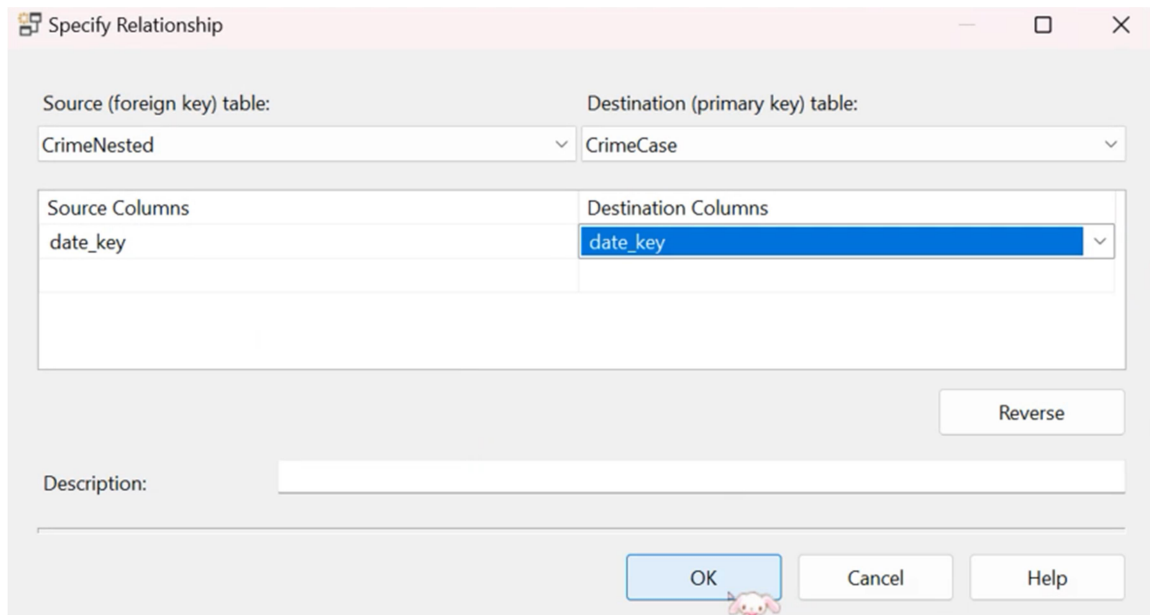
The code above is for the nested table. It has no primary key and is just full of foreign keys, as well as the crime and neighborhood text column. Unlike the case table, this one does not have unique values. In sense we could say that the case table is like dim tables, how they are all of unique value, while the nested table is like the fact table where it has recurring rows.

After executing with no errors, we head over to visual studio and select an analysis services multidimensional template. Change the host to what's on SSMS by right clicking the properties of the solution name on the right and right click on data source to create a new data source. Connect to the OLE DB SQL server and your server + database then click test connection and hen it's all good, click through until you reach a username and password and just enter your Microsoft email and password and click finish.



Below it, right click data source view to create a new view. Click through next until you see a table that has >, and click both your views and move them to the right using the > button.

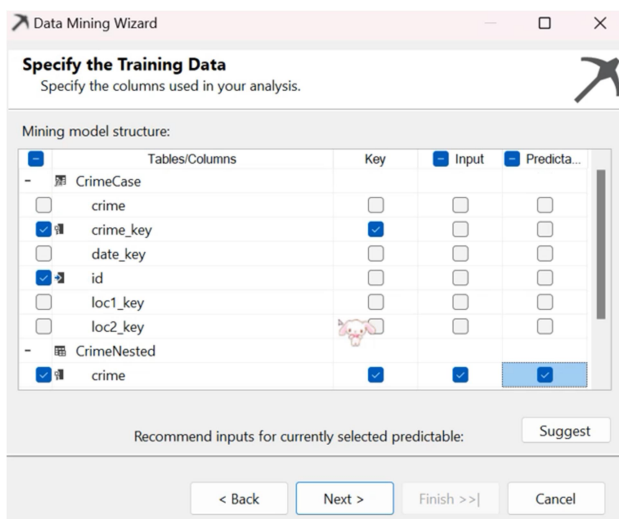
Double click on the data source view we just made and you'll see two tables of the case and nested views. We'll now need to make a relationship between them so highlight and right click one of the keys on the nested tables and click on new relationship. Once we've done that you want to match that key with the same key on the case table (which is why we needed the foreign keys to be on both tables). In this case, I used date\_key so it looks like this.



The 'Specify Relationship' dialog box shows the configuration for a relationship between two tables. The 'Source (foreign key) table' is 'CrimeNested' and the 'Destination (primary key) table' is 'CrimeCase'. Under 'Source Columns', 'date\_key' is listed. Under 'Destination Columns', 'date\_key' is also listed and highlighted in blue. A 'Reverse' button is located at the bottom right. At the bottom, there are 'OK', 'Cancel', and 'Help' buttons. A small pink cat icon is visible near the 'OK' button.

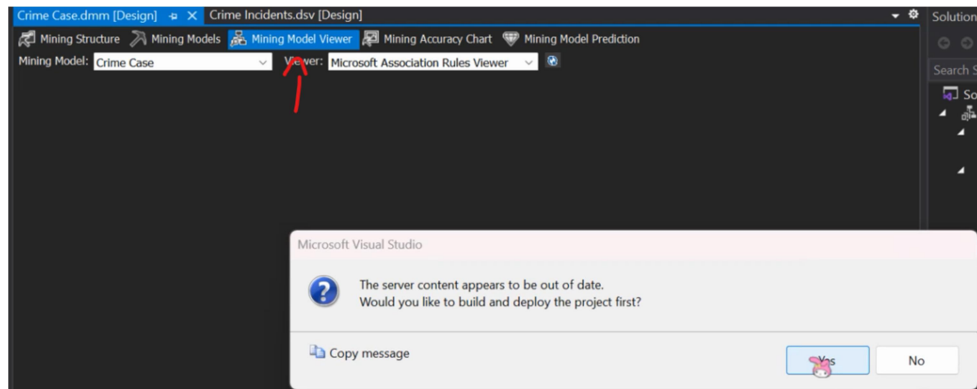
You'll know it worked if you see a thin line between the two tables.

Now, let's do the mining. Right click on mining structures in the solution explorer and click on new mining structure. Click next, then keep it on use existing tables, then you'll see a create data mining model with a drop down menu. Click 'Microsoft Association Rules' and click through next until you see an option to select which table is the nested and which is the case. After selecting, there will be a specify training data window where you'll select a key that uniquely identifies your column and highlight it as key, unselect the key tick from your primary key and tick all the boxes for one of your text columns in your nested tables. In this case, since I used the date\_key, I would like to see how it compares to crime. After this, click next and finish.

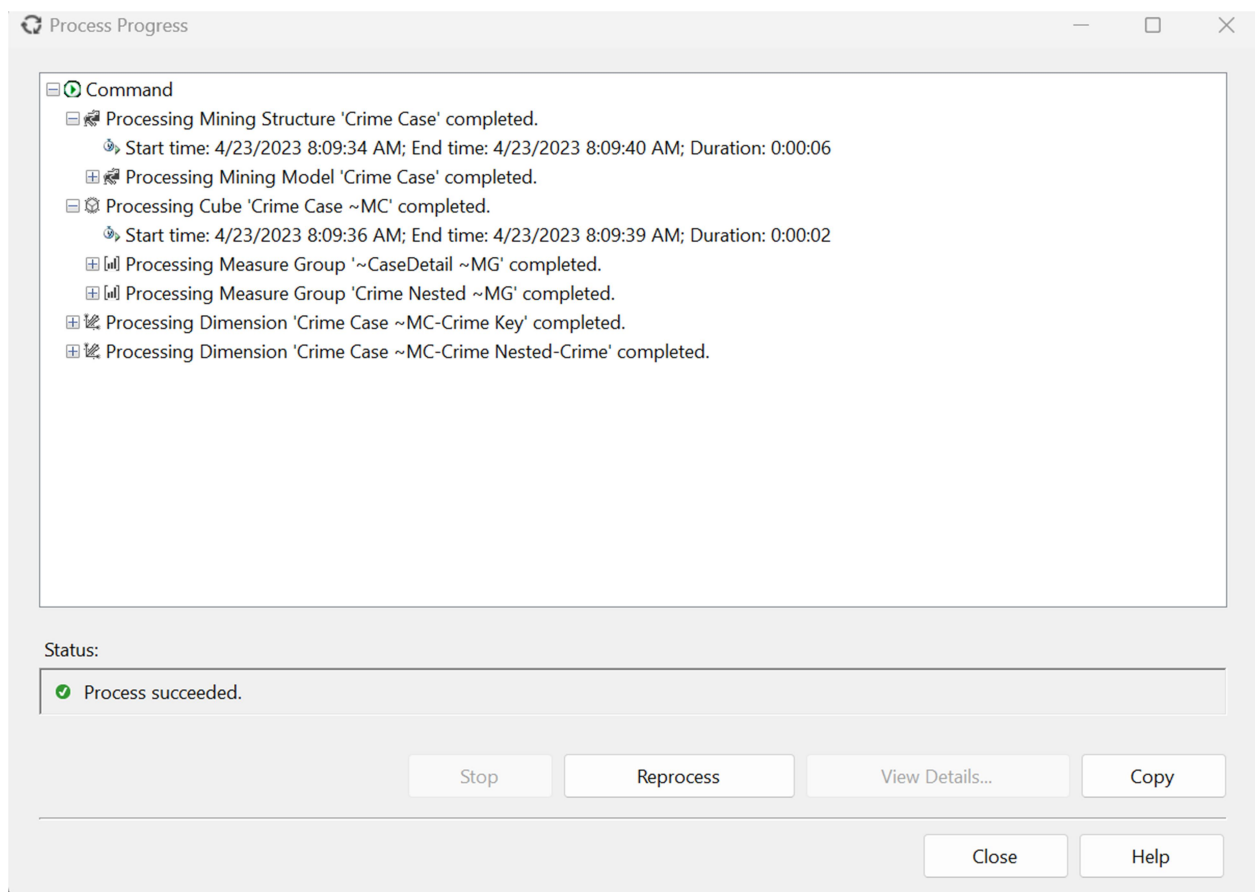


The 'Data Mining Wizard' window, 'Specify the Training Data' step, shows the mining model structure. It lists columns from 'CrimeCase' and 'CrimeNested'. The 'CrimeCase' section includes 'crime', 'crime\_key', 'date\_key', 'id', 'loc1\_key', and 'loc2\_key'. The 'CrimeNested' section includes 'crime'. The 'Key' column has checkboxes: 'crime\_key' is checked, 'id' is checked, and 'crime' is checked. The 'Input' column has checkboxes: 'crime' is checked. The 'Predictable' column has checkboxes: 'crime' is checked. A 'Recommend inputs for currently selected predictable:' button is at the bottom. At the bottom, there are '< Back', 'Next >', 'Finish >>', and 'Cancel' buttons. A small pink cat icon is visible near the 'id' row.

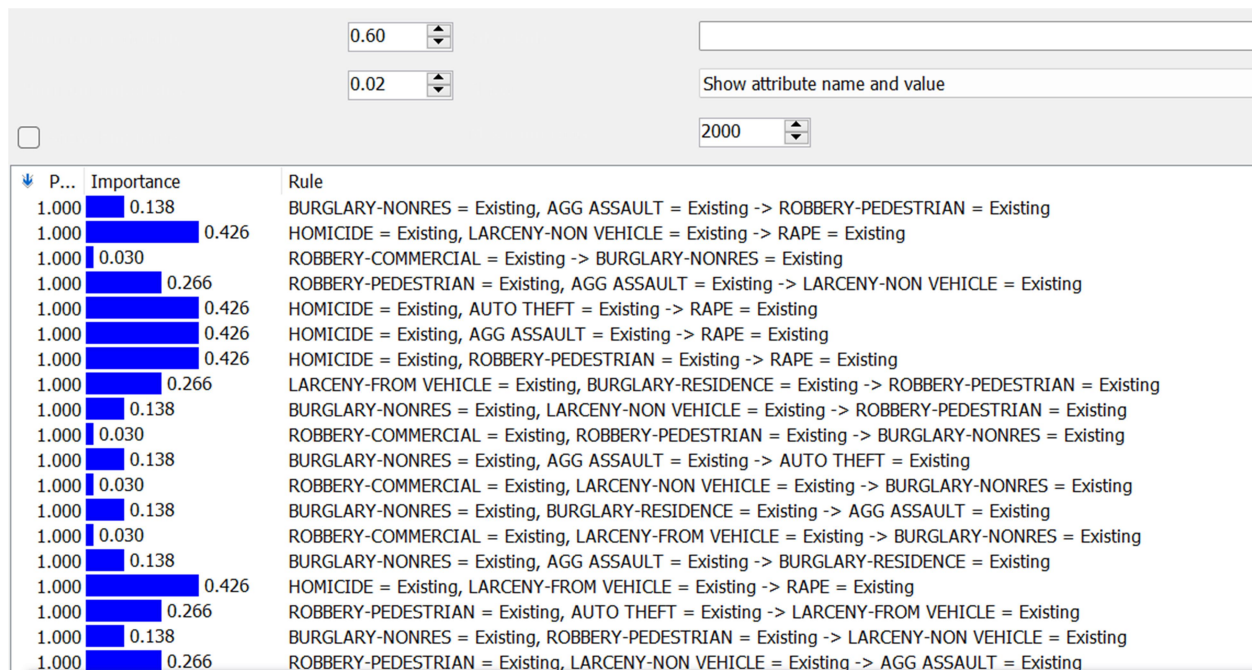
Let's process it! Click on Mining Model Viewer and click yes.



It will then deploy and process, and once it successfully deploys click run. Once succeeded, it should look like this



Clicking close will automatically show results. We are now able to identify top k rules from the 'rules' or 'dependency network' views, and changing some of the display settings and rule thresholds, will output meaningful rules.



Even though the above graph has some meaningfulness, as they all have a support of 1.0, indicating that the antecedent (left-hand side) and consequent (right-hand side) always appear together in the dataset, but we see that the importance values are all kind of the same. It's possible that this is the case because there is not much variation in the data, there only being 10 crimes and how, possibly, the algorithm may have simply memorized the patterns instead of discovering meaningful associations.

That being said, we are still able to interpret the k-rules, for example, in this row:

1.000 0.266267889404769 ROBBERY-PEDESTRIAN = Existing, BURGLARY-RESIDENCE = Existing -> LARCENY-FROM VEHICLE = Existing

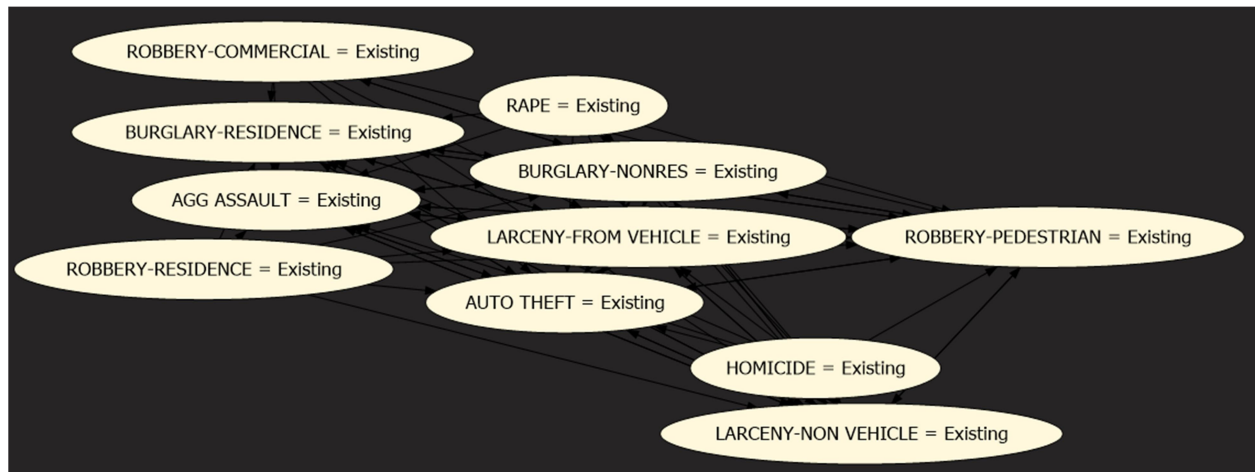
The antecedent is "ROBBERY-PEDESTRIAN = Existing, BURGLARY-RESIDENCE = Existing" and the consequent is "LARCENY-FROM VEHICLE = Existing". This rule is saying that if a pedestrian robbery and a residential burglary occur together, then there is a high probability that there will also be a larceny from a vehicle, which makes sense.

Another example is this row:

1.000 0.138302698166281 BURGLARY-NONRES = Existing, LARCENY-NON VEHICLE = Existing -> ROBBERY-PEDESTRIAN = Existing

This rule has a support of 1.000, which means that all the instances in the dataset satisfy the antecedent and consequent of the rule. The confidence of the rule is 0.138, which means that only about 14% of the instances that have both burglary-nonres and larceny-non vehicle incidents actually have a pedestrian robbery incident.

Overall, it suggests that the presence of both burglary-nonres and larceny-non vehicle incidents might be a weak predictor for pedestrian robbery incidents. However, the low confidence indicates that this rule alone may not be a strong enough predictor on its own, and more investigation is required to determine the relationship between these variables.



This is the dependency network of the graph above. The nodes all seem to be connected to one another which mean that there is a strong interdependence between all the crimes in the dataset. Even though it's not necessarily a bad thing, it may indicate that the network is too dense and needs to be simplified; while it's unlikely that a crime is related to another, it is very much possible, but in a 200,000 rowed dataset, it might be hard to find a pattern. Furthermore, the fully connected nature of this network suggests that there may be complex interactions and feedback loops among the variables, which could make it challenging to disentangle the causal relationships between them.

In conclusion, the k-rules generated from the crime dataset provide some insight into potential associations between different types of crimes. Although the results may not be entirely conclusive due to the limited number of instances in the dataset, the generated k-rules can still provide a basis for further investigation and analysis. The dependency network also highlights the complex and interdependent nature of crime patterns, which may require further research to fully understand the causal relationships between different types of crimes. Overall, the k-rules and dependency network provide a useful tool for identifying potential patterns and relationships in crime data.