ADM

Data Mining Project

Done by:
Mhd_Mohsen_121343

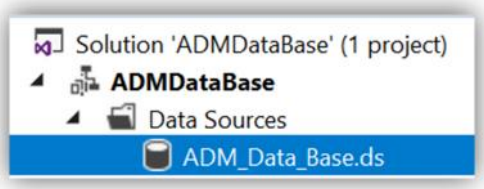
Mohammad_121485

Ahmad_121295

Contents

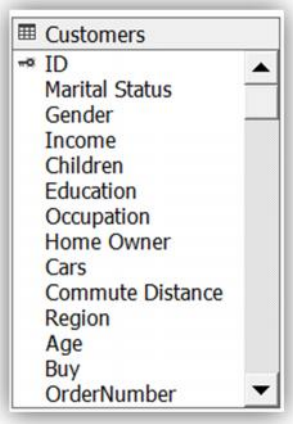
1- Introduction

To create this data mining project we had at first to create a new Analysis services Project in visual studio, once we created it, we created a new data source related to our SQL database and then we started to use this data source to build our needed models.



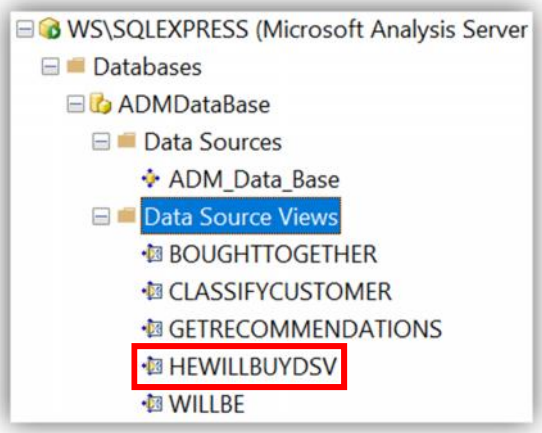
2- He will buy

This mining model should be build based on Byes algorithm, so at first we created data source view, using our data source, we name it He Will Buy DSV “HEWILLBUYDSV”, the data source clearly should contain the customers table to calculate the predicted percentage.

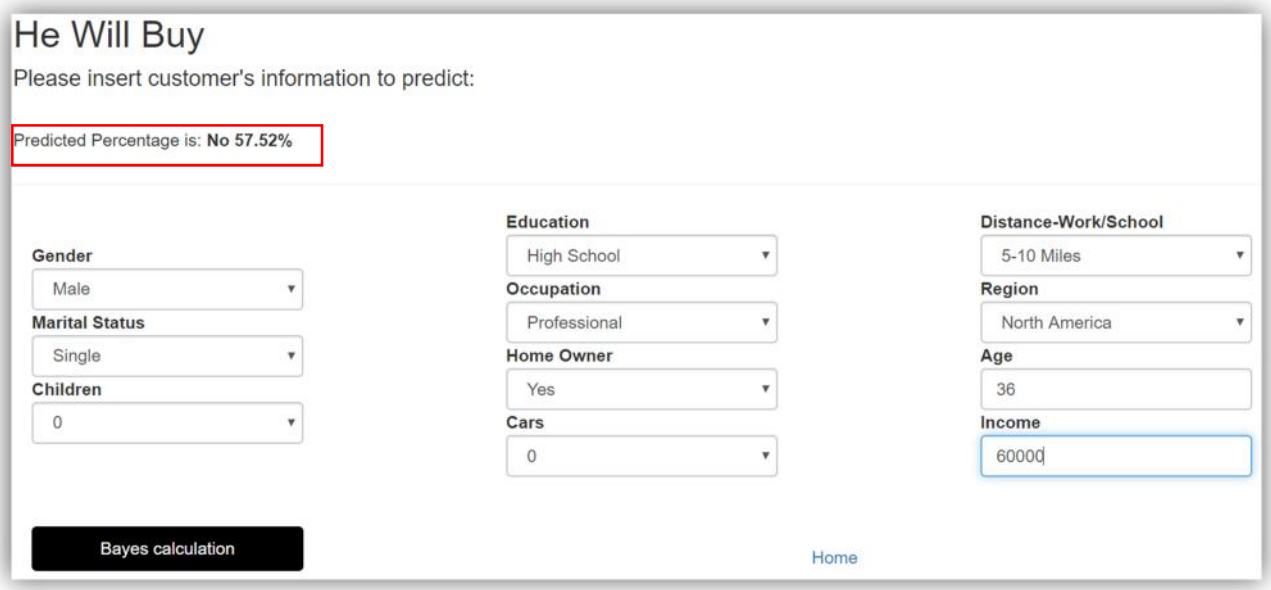


Then we created new mining structure model, from type Naïve Bayes, there were no need to edit any parameters.

We went to our SQL server and we make sure that the model created under Analysis services, which it did.



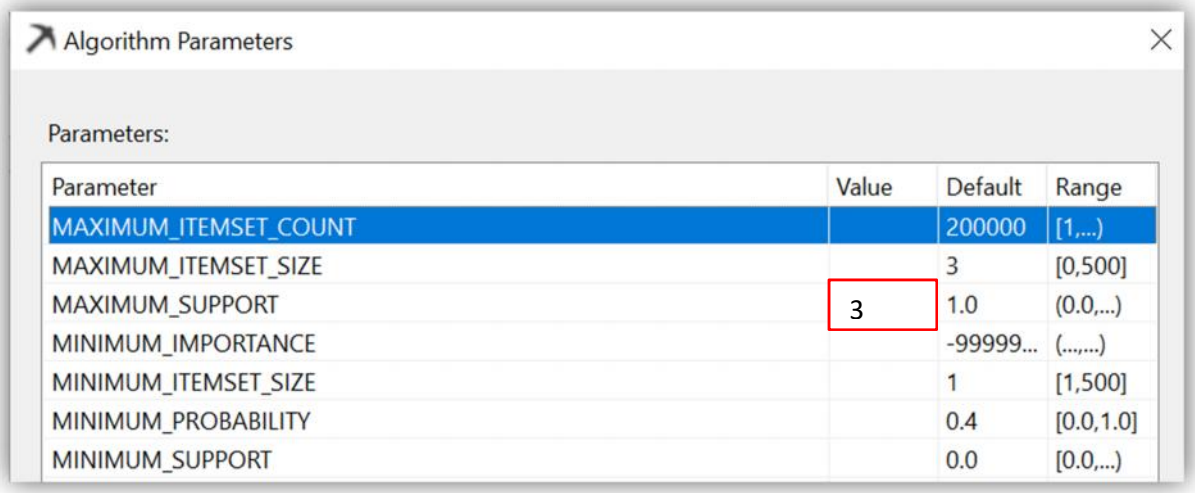
Then we went to our created Web Application Solution and started to make new AdomdConnectin to our DB and we write the correct query to retrieve the information from that data mining model. Then we designed our page to meet our needs, so when the admin inserted correct characteristics of a customer, he will get the predicted percentage for that customer if he is going to buy or no.



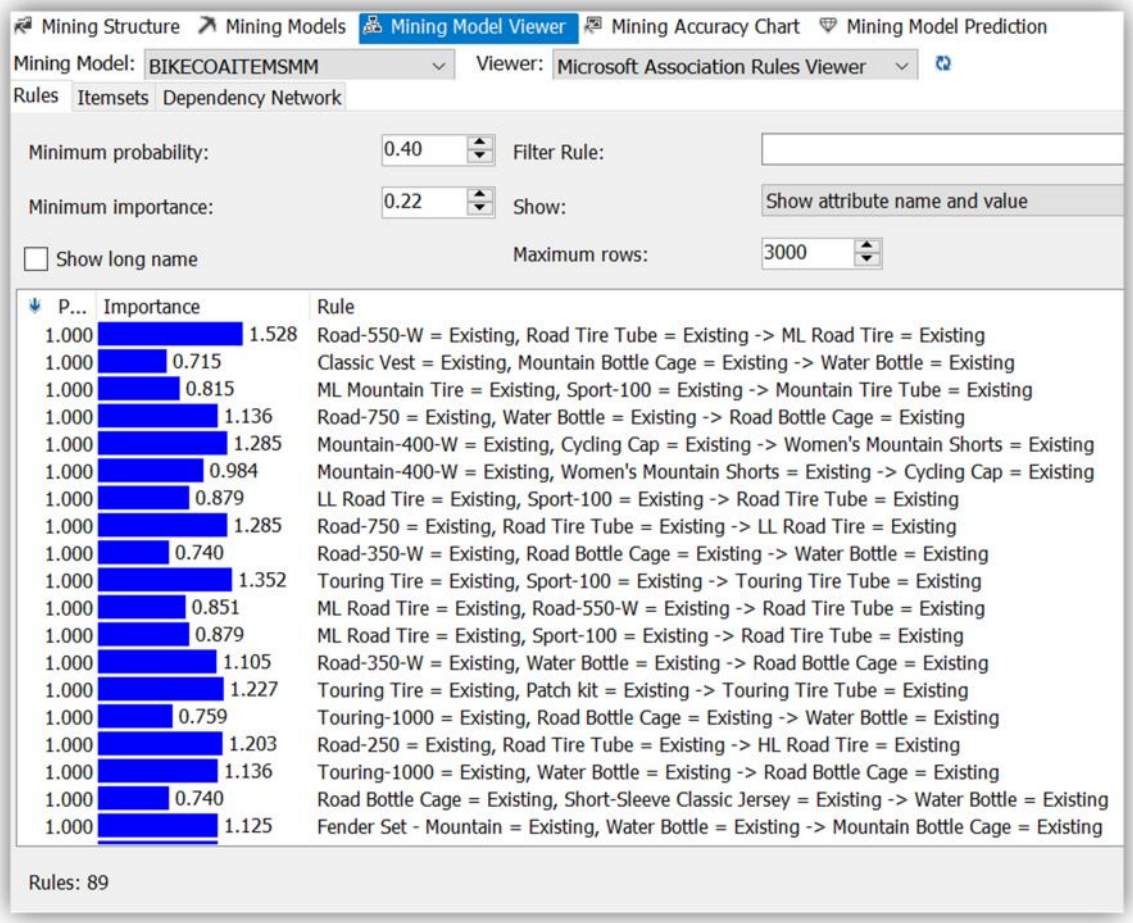
3- Frequently bought together

This data mining model should depend on Association algorithm so to create it we do the following. After creating new data source view related to both orders table and ordersdetails table we had to set the parameters of the association algorithm.

The most important parameter is the “minimum support”, which we need in our case 3.

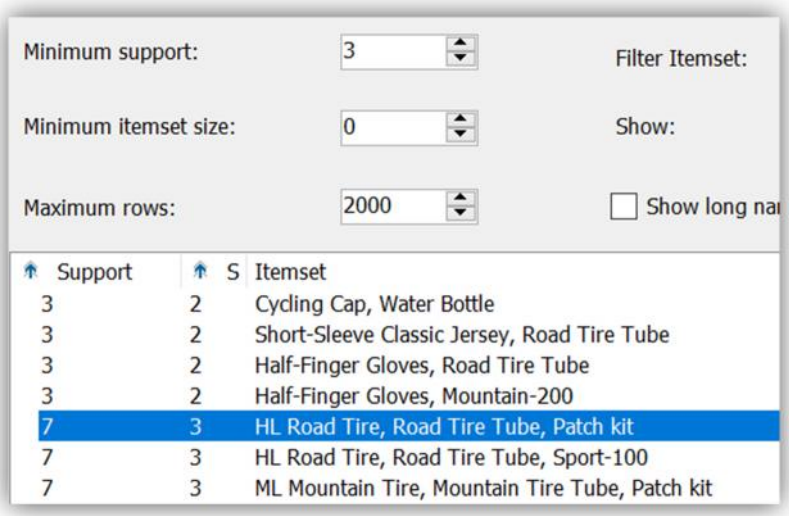


For the minimum probability parameter and as we have only 37 items “small number” so we leave it to its default, which is 0.4. And after testing 0.5 till 0.9 of “minimum probability” we found it’s the best value as it give us the most accurate number of rules “89 rule”.

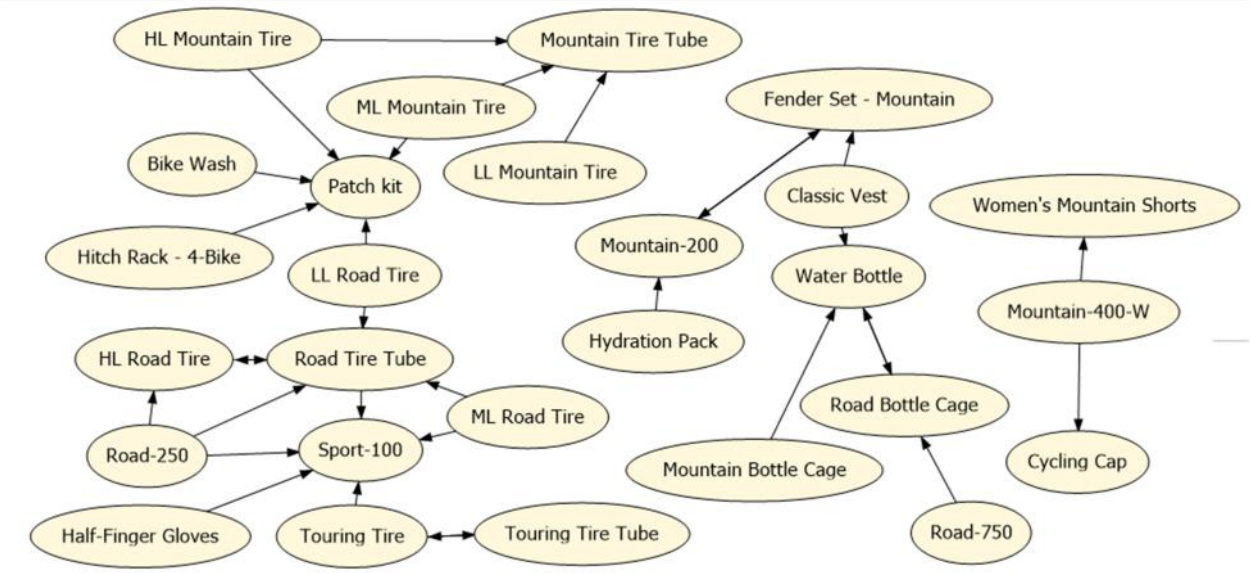


In the “frequent item set” we noticed 161 results showing us the size of each redundancy, for example its shows that these three items are repeated 3 times in the DB:

Size	Itemset
3	HL Road Tire, Road Tire Tube, Patch kit.



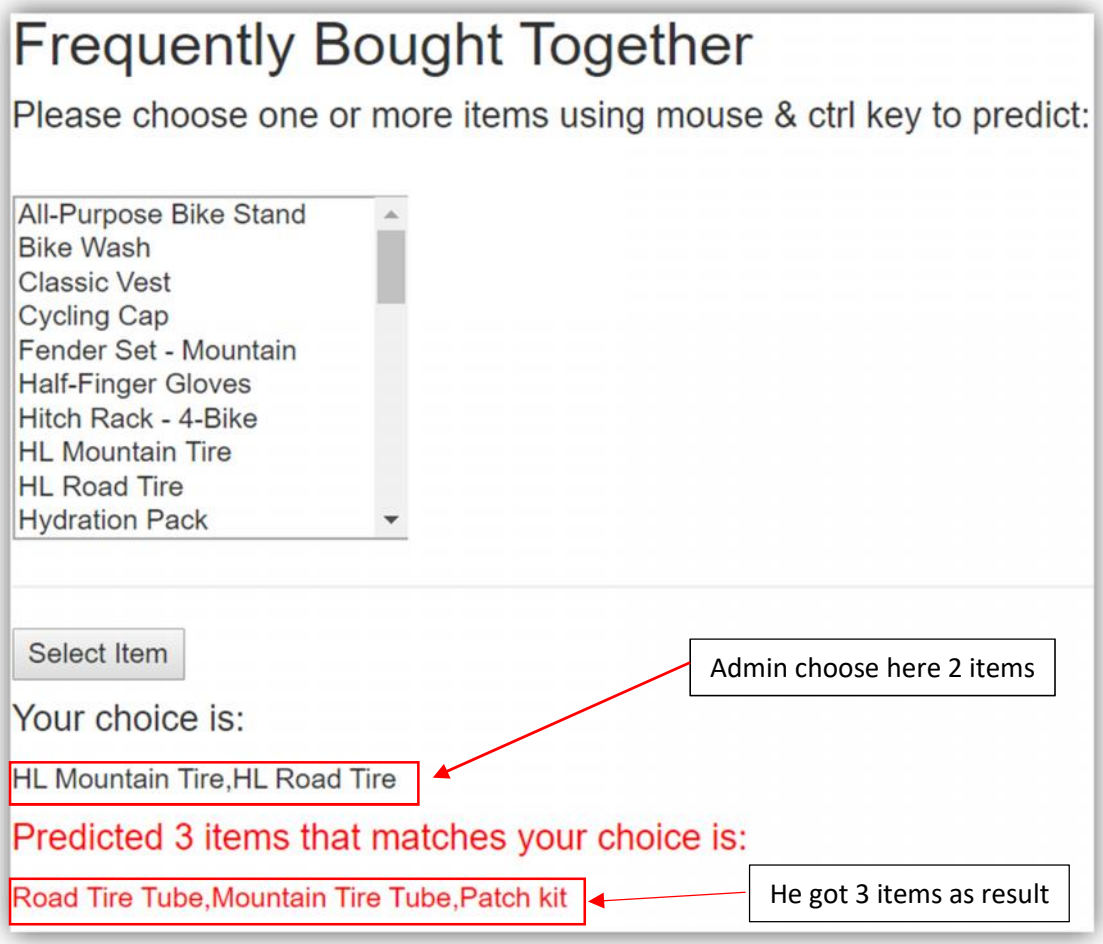
At last in the viewer we can check the dependency relations, we noticed that there is 3 main groups are related to each other according to our requirements.



The most powerful relation was between the “Touring tire and the Touring tire tube” for example, and the most weak relation was between the sport 100 helmet and the touring tire, which make “sense”.

One more thing we noticed that there is no relations between the Cycling cap and Women’s shorts with other products in general, hence we can say “likely” that the “fashion products” in the company considered to be a new “work sector” that it not really related to the main company work sector, or simply that means that the bike “Mountain-400” is really a “girlish” bike as it’s the only product that it related to women’s fashion products.

After implementing the code and design to our web solution we had this page to be used by the admin.

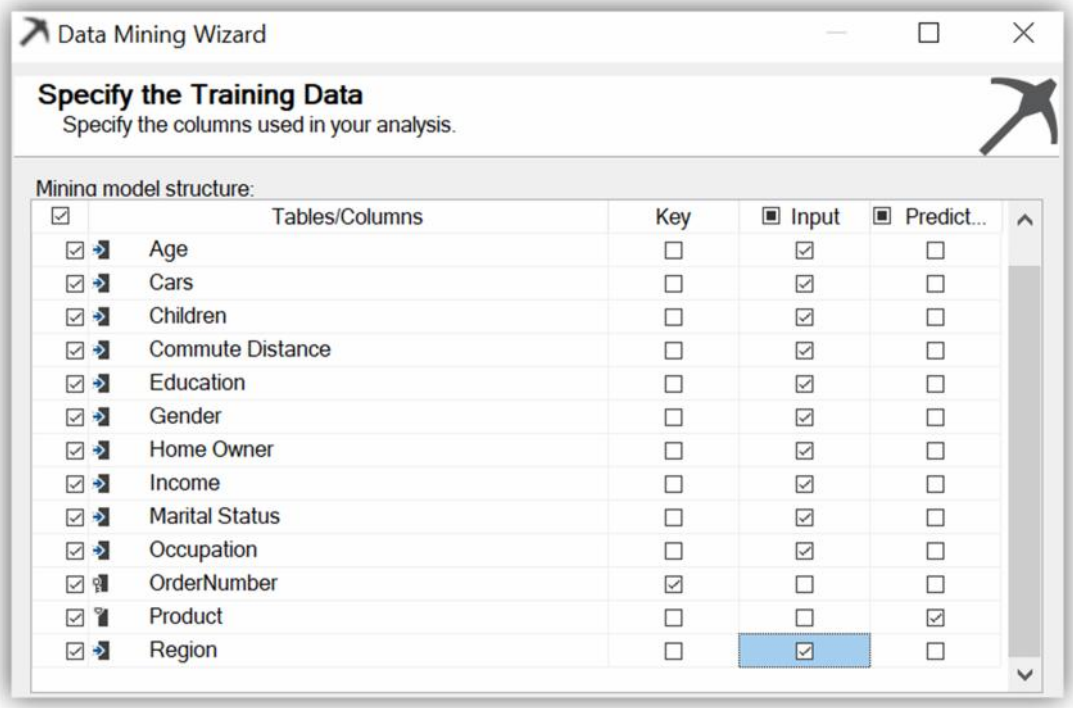


User Admin here can select the items from the list box, ”One or many” and check what is the best 3 items are bought with the items he had chosen.

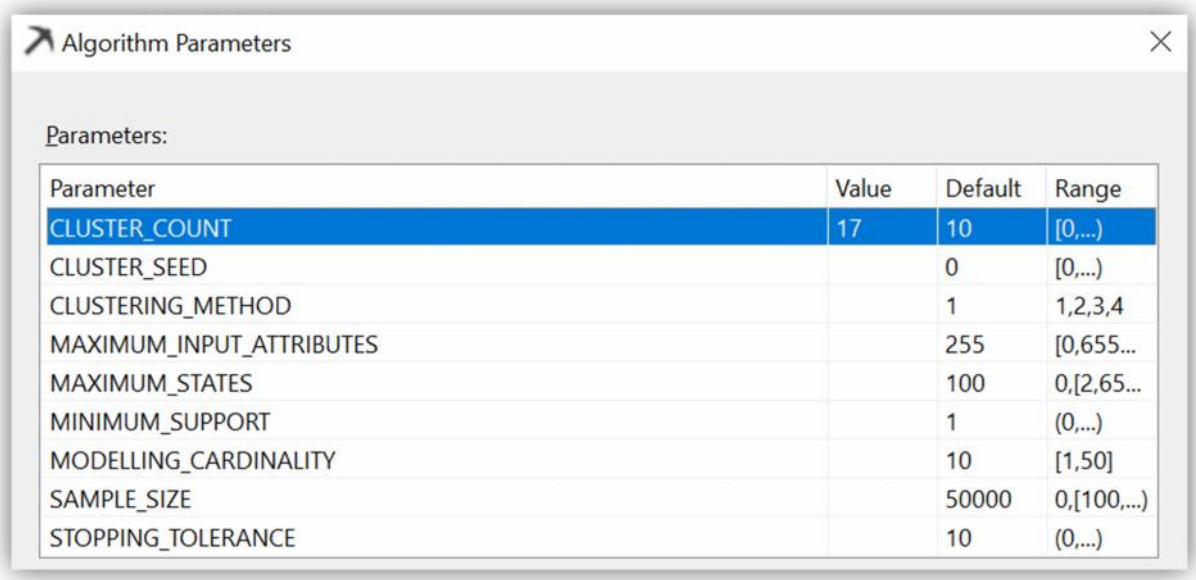
4- Get personalized recommendations

Here we will use the clustering algorithm to show the products that might customer’s buys based on their characteristics.
So after creating related view that binds customers table with orders table and orders details table in database, we created a new data source view (customers - products).

Then we inserted the inputs and the predicted value (which is products in our case.)

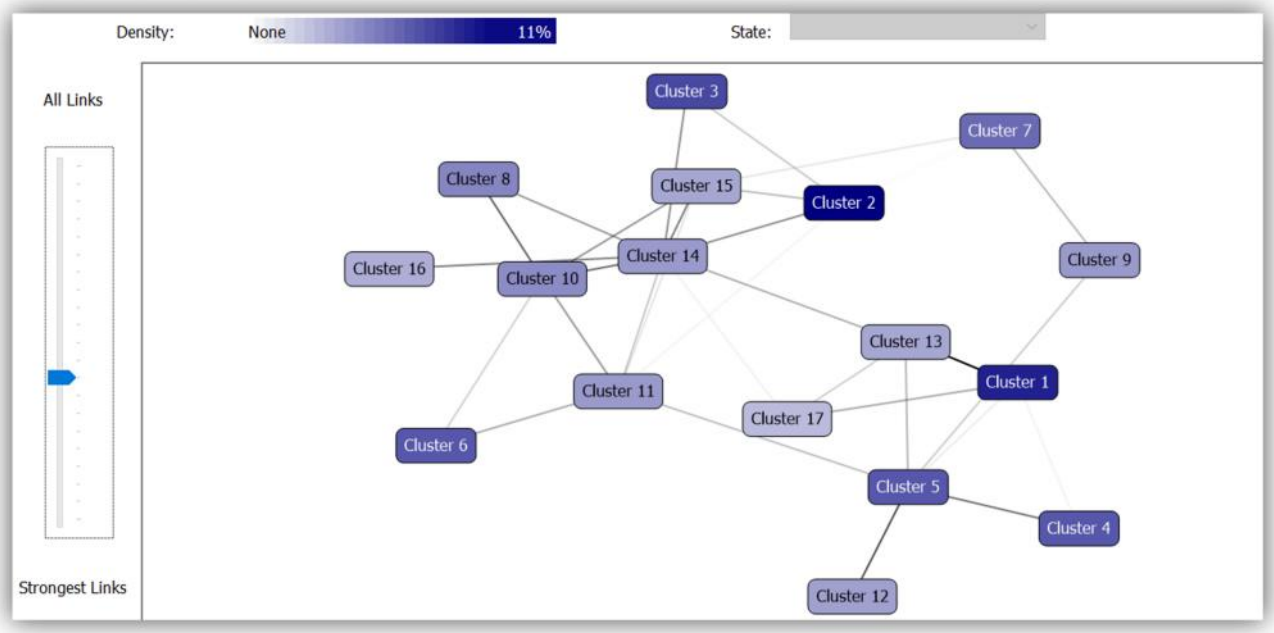


The percentage of tested data was 30% as we have a lot of cases in our case.
Now there is an important parameter to edit in our case, which it the cluster count, we decided to make the clusters count: 17 which it the our categories count ”logical sense”.

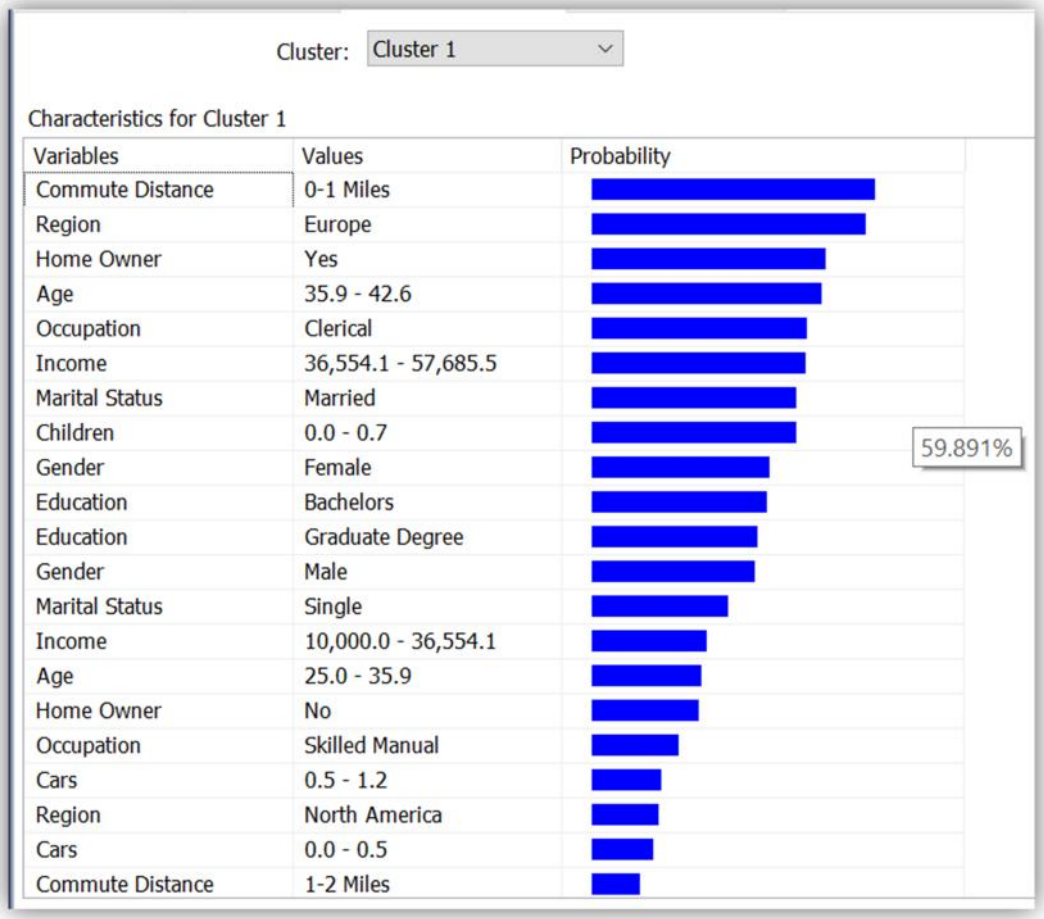


Parameter	Value	Default	Range
CLUSTER_COUNT	17	10	[0,...)
CLUSTER_SEED		0	[0,...)
CLUSTERING_METHOD		1	1,2,3,4
MAXIMUM_INPUT_ATTRIBUTES		255	[0,655...
MAXIMUM_STATES		100	0,[2,65...
MINIMUM_SUPPORT		1	(0,...)
MODELLING_CARDINALITY		10	[1,50]
SAMPLE_SIZE		50000	0,[100,...)
STOPPING_TOLERANCE		10	(0,...)

Now we processed the model and we checked the cluster diagram:



Then we went to cluster characteristics and here we choose cluster 1 for example:

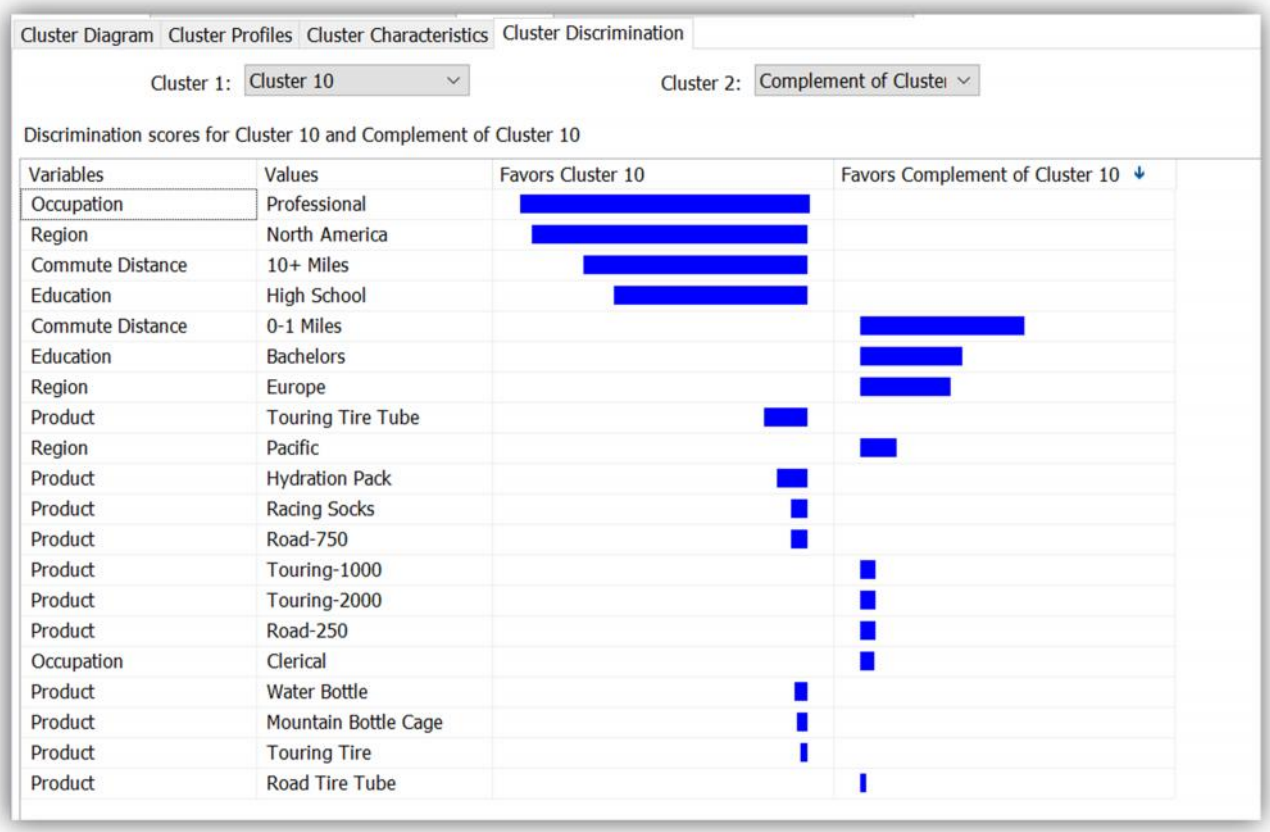


Cluster: Cluster 1

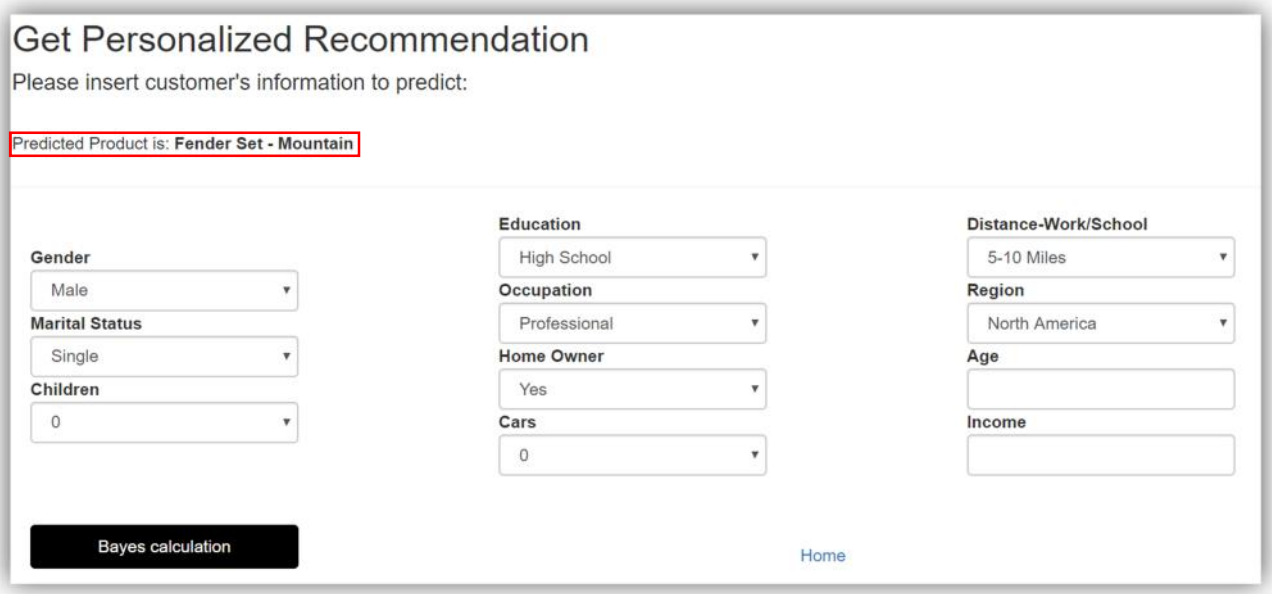
Characteristics for Cluster 1

Variables	Values	Probability
Commute Distance	0-1 Miles	
Region	Europe	
Home Owner	Yes	
Age	35.9 - 42.6	
Occupation	Clerical	
Income	36,554.1 - 57,685.5	
Marital Status	Married	
Children	0.0 - 0.7	59.891%
Gender	Female	
Education	Bachelors	
Education	Graduate Degree	
Gender	Male	
Marital Status	Single	
Income	10,000.0 - 36,554.1	
Age	25.0 - 35.9	
Home Owner	No	
Occupation	Skilled Manual	
Cars	0.5 - 1.2	
Region	North America	
Cars	0.0 - 0.5	
Commute Distance	1-2 Miles	

Now there is a good comparison tab “cluster Discrimination” that showing us the variance for each cluster according to the customers characteristics value, for example here you can see that products in cluster 10 are not likely being bayed by region of North America:

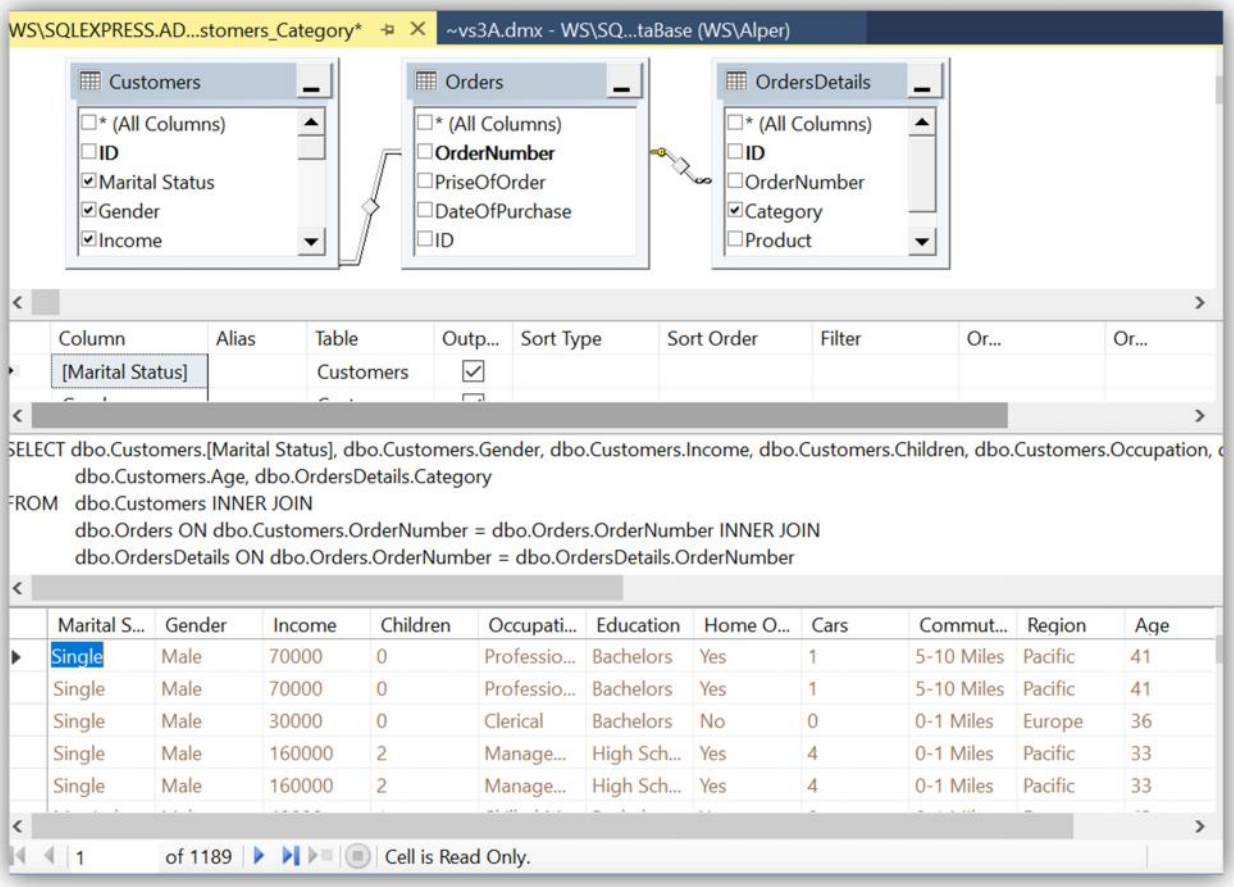


Later on we configure our web application, with needed coded and design and the result was like the following:

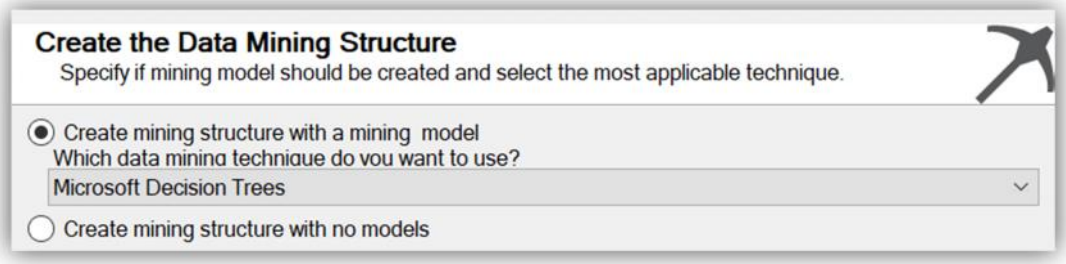


5- Classify customer

Now we need to create a new data source view to create the data mining model to classify customers to what predicted categories they might buy.
The data source view should be related to a “SQL View” that we build earlier that it join the customers characteristics (from customers table) to the categories (from oredersdetails table), through the order number table.



Then we have to build the mining model as a decision tree model.

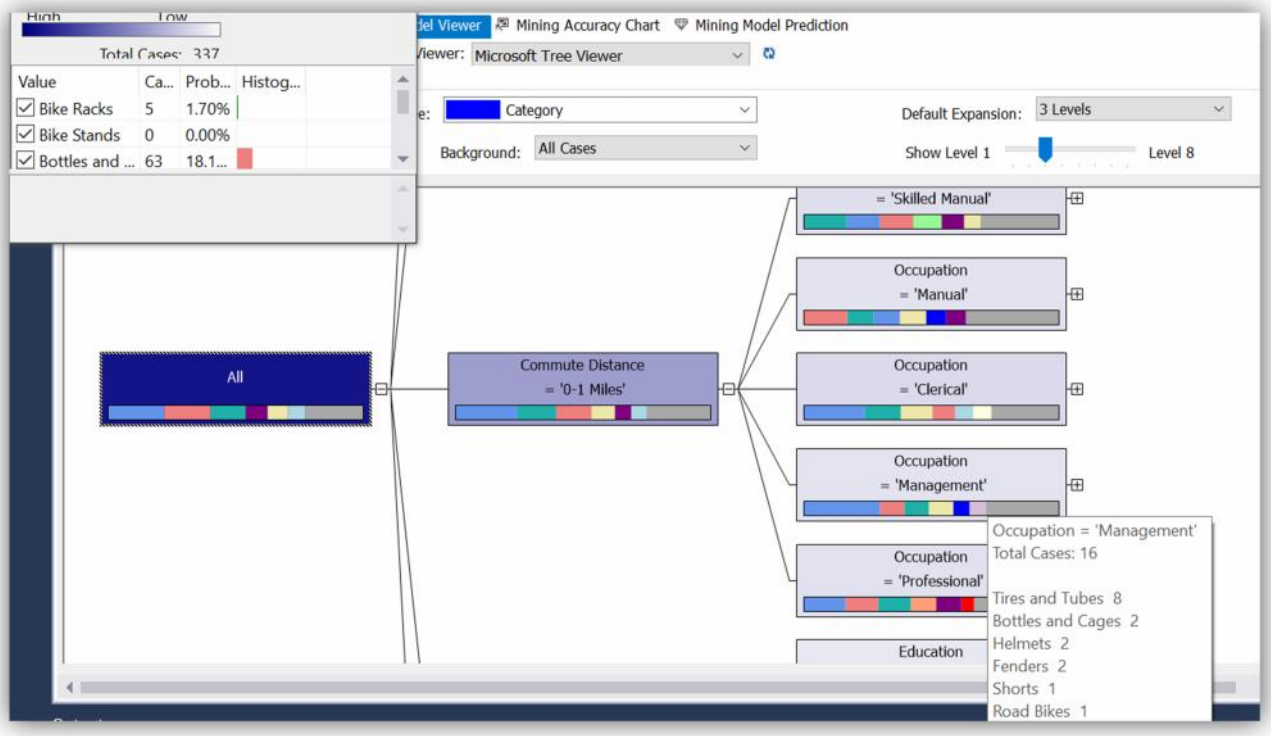


And we do left the percentage of the tested data to 30% as there is a huge number of records.

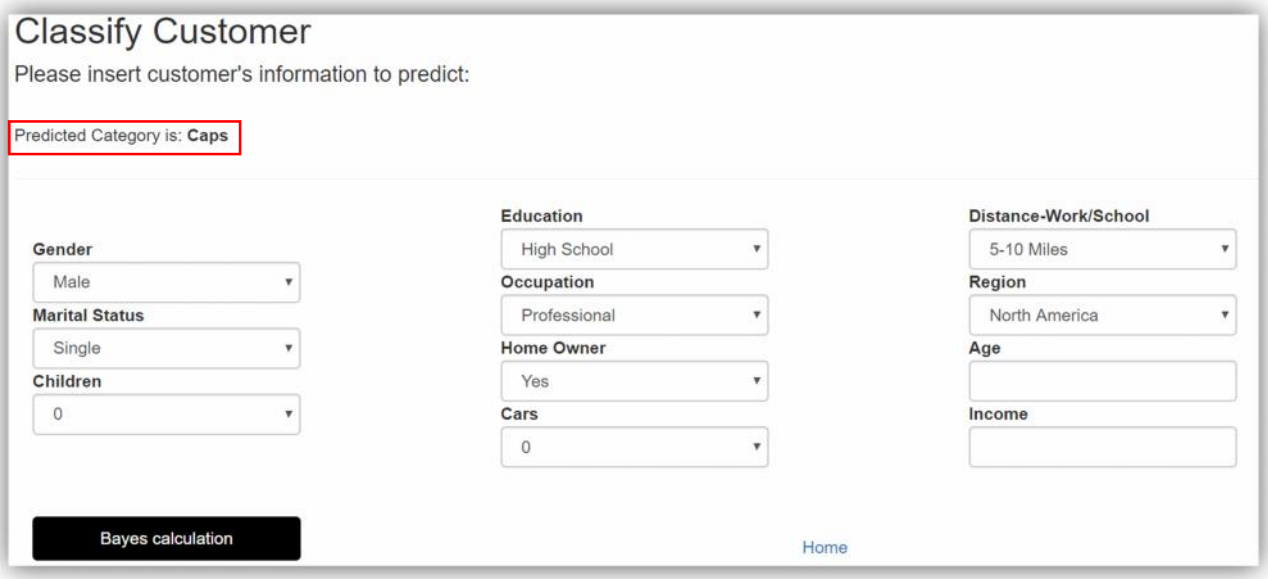
Other parameters we choose to edit was:

MINIMUM_SUPPORT	1	10.0	(0.0,...)
SCORE_METHOD	1	4	1,3,4
SPLIT_METHOD	3	3	[1,3]

And the final result of the tree was:



Then after implementing the code of our web application, we created this page to be used by admin.

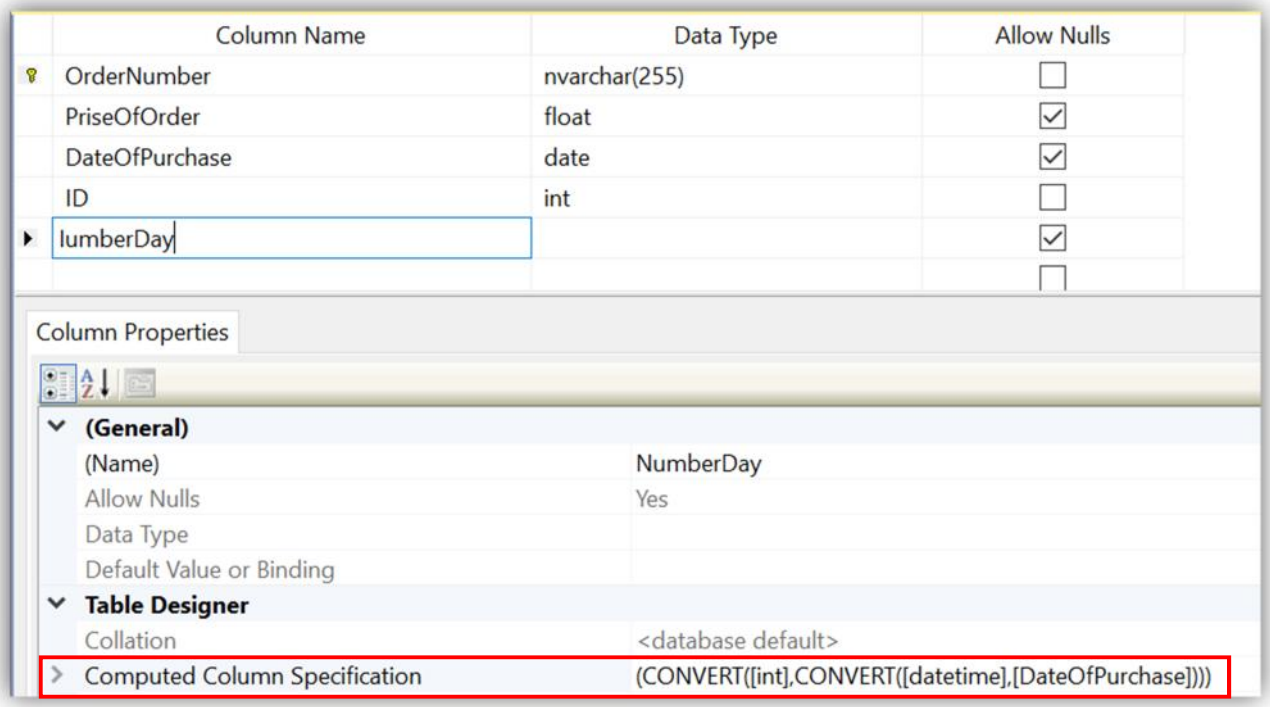


6- Time series

In this algorithm we have to create a column to handle the date issue, it should be “int” though.
So we created a calculated column related to the real date column.

	OrderNumber	PriseOfOrder	DateOfPurchase	ID	NumberDay
473	SO61740	29.99	2018-07-07	4...	43286
474	SO61741	14.98	2018-05-15	4...	43233
475	SO61742	68.97	2018-04-30	4...	43218
476	SO61743	14.98	2018-12-26	4...	43458
477	SO61744	14.98	2018-08-12	4...	43322
478	SO61745	69.99	2018-07-14	4...	43293
479	SO61746	78.98	2018-03-16	4...	43173
480	SO61747	609.98	2018-11-26	4...	43428
481	SO61748	609.98	2018-10-21	4...	43392
482	SO61749	4.99	2018-12-18	4...	43450
483	SO61750	7.95	2019-12-11	4...	43808

To change the date to integer we used the computed column specification like:



Column Name	Data Type	Allow Nulls
OrderNumber	nvarchar(255)	☐
PriseOfOrder	float	☑
DateOfPurchase	date	☑
ID	int	☐
lumberDay		☑
		☐

Column Properties

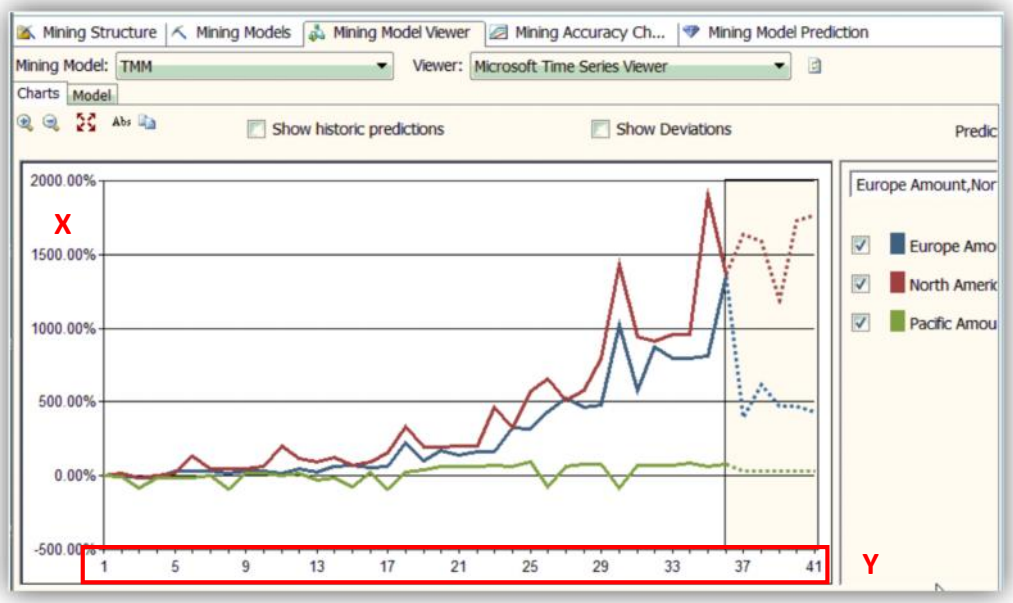
(General)	
(Name)	NumberDay
Allow Nulls	Yes
Data Type	
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	(CONVERT([int],CONVERT([datetime],[DateOfPurchase])))

Note that the first value 1, is at day 1-1-1900 as Microsoft standards, so any other new records will be changed to “int” into specific column in order to solve time series data mining in real cases and real time of purchasing.
We suggested that all old purchases orders were purchased during 2018, and as we already have 482 records, so we shuffled their purchasing date into 2018 only, which make sense to our algorithm.
We set the percentage of tested data to 30% as we have a lot of records.

NOTE:
We discovered mini error in our treatment to time series algorithm in our last lab, to make my point clear I will replicate our lab issue then I will explain my point.
In our lab we had 3 columns:

- Year/month int as an input.
- ID int as key.
- Sales (North America Amount, Pacific Amount and Europe Amount) as predicted.

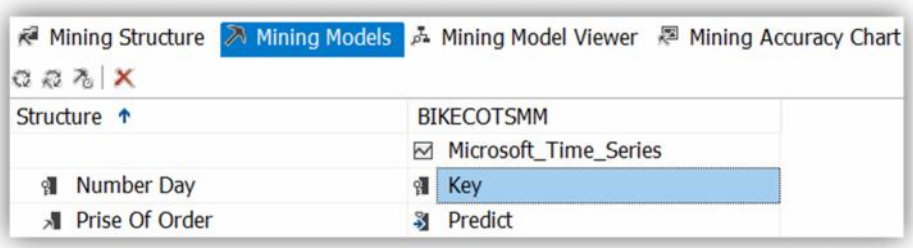
and the chart was like the following:



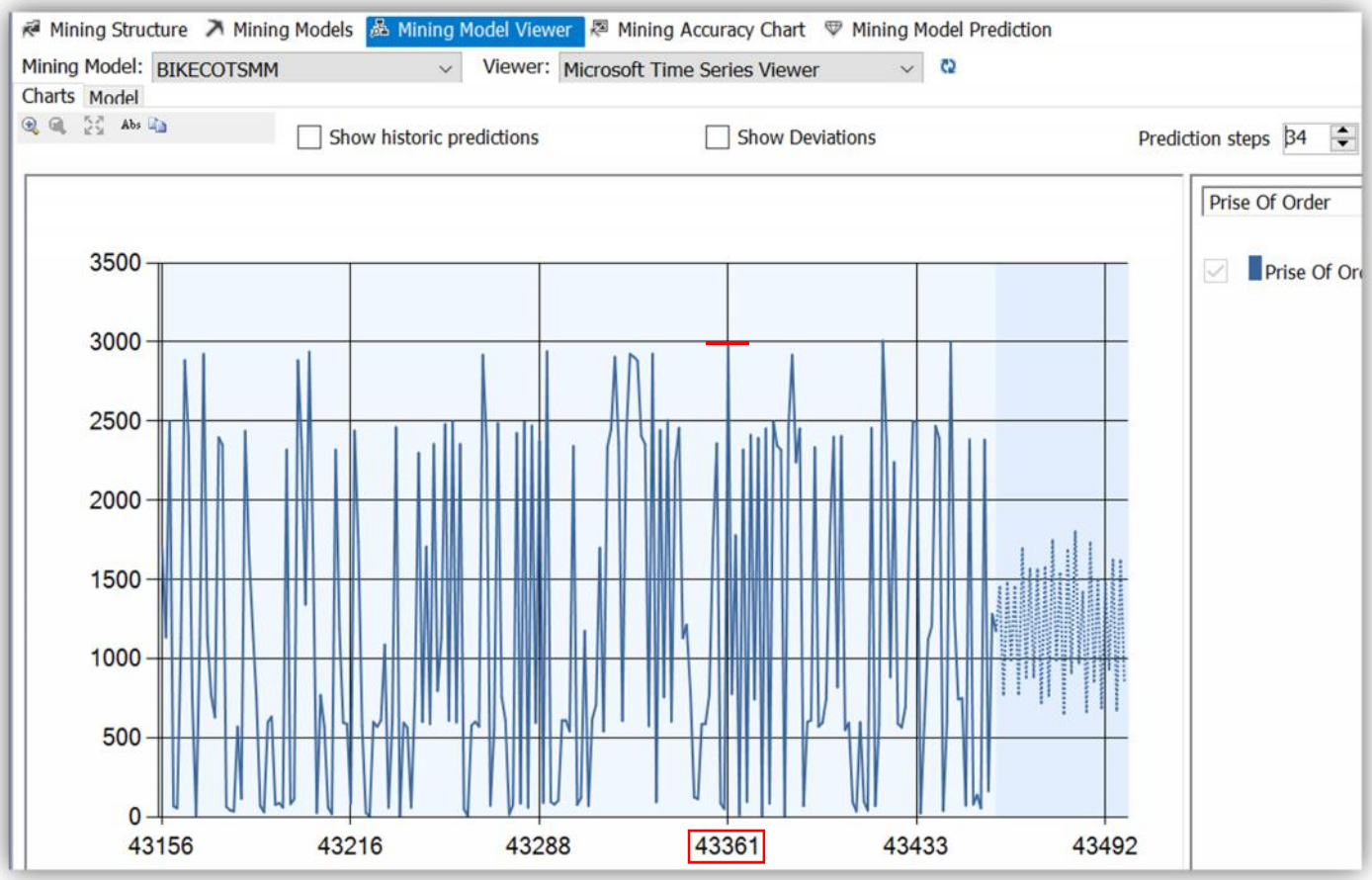
Now let's analyze the "Axis", the X axis is the amount of sales, and Y axis that started from 1 till 36, and continue predicting till 41, that's in-fact is the "ID" column, and it's not logical to predict the sales based on continuity of an ID, we need to predict based on the date value not in the Id value.

So in my case I changed the values of my mining model to:

- Year/month/day (NumberDay) as integer and it should be the Time-key or simply the key as we want to "predict" based on this key "predict during Time-key".
- Sale (PriseofOrder) as an input, because we want to study the amount of sales during the time.
- Sales (PriseofOrder) again as a predicted value, because I want to predict the sales amount during next periods.



Note that (PriseofOrder) is predict so its input and predict, but we can make predict only, and that will not help our case as we will have no input to study and learn from.
And here is the histogram that I got:

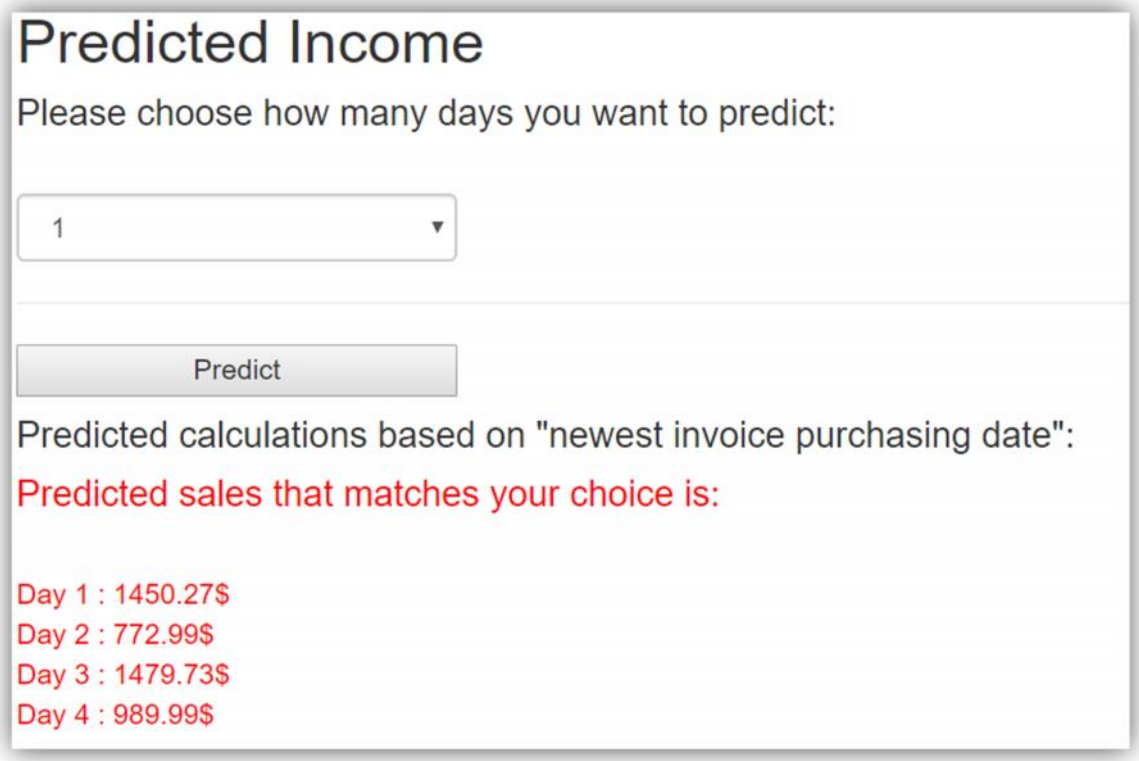


And to prove my point, for example you can notice here that the day number (43361) stated that the sales was near 3000, so I went to my data base to check that date sales value, and here is the result.

Results | Messages

	OrderNumber	PriseOfOrder	DateOfPurchase	ID	NumberDay
352	SO61495	49.99	2018-09-18	228	43359
353	SO61522	2983.34	2018-09-20	255	43361
354	SO61439	2493.34	2018-09-20	172	43361

The amount was 2983.44 which it exactly the real situation and its matching the mining model that I built, so as a notice: the int (time) has to be the key not the input, and the amount should be the input and the predicted factors.
After making sure of our calculations we went to our website to apply the algorithm, we build and designed the needed page, and here is how to use it.



7- Notes

- To use the project, open the main page, head to products tab, login using:
  Admin credentials:
  username: email@test.com password: 123
  Regular user account:
  username: User@Bike.co password: 15
  DB credentials:
  username: sa password: Admin12345678 (just in-case).
  The website is hosted online without the ADM part, as we couldn't upload it
  www.bikeco.somee.com
- The project solution (both App and Analysis) is attached to this file.
- The code of related functions is attached.
- The used DB is attached.