Decision Management Community Challenge Oct-2025

"Decision with two objectives"

https://dmcommunity.org/challenge/challenge-oct-2025/

After Jacob used Copilot to solve this challenge, he asked me to implement the same approach in OpenRules. Copilot used the following greedy algorithm to select the most valuable features for the website design:

1) For each feature compute **Value-to-Cost ratio**:

$$\text{Ratio} = \frac{\text{Value}}{\text{Cost}}$$

2) Sort all features by their rations in the descending order.
3) Select features from this sorted array until budget runs out.

Here are the Copilot's results:

Selected Features: 5, 4, 6, 3, 7, 8, 9
Total Cost: $10,000
Total Value: 3 + 4 + 2 + 5 + 1 + 1 + 1 = **17**
Number of Features: 7

Below is my implementation of the same approach. I created an OpenRules project "WebsiteDesign" using mainly Excel. Here is my Glossary:

Glossary glossary				
Variable Name	**Business Concept**	**Attribute**	**Type**	**Used As**
Budget	**Design**	budget	double	in
Features		features	Feature[]	in
Selected Features		selectedFeatures	String[]	out
Total Value		totalValue	double	out
Total Cost		totalCost	double	out
Name	**Feature**	name	String	in
Cost		cost	double	in
Value		value	double	in
Ratio		ratio	double	

Here is my test data:

DecisionData Feature features		
Name	Cost	Value
1	$5,000	7
2	$4,000	6
3	$3,000	5
4	$2,000	4
5	$1,000	3
6	$1,000	2
7	$1,000	1
8	$1,000	1
9	$1,000	1
10	$1,000	1

DecisionTest testCases		
#	Define	Define
Test #	**Features**	**Budget**
1	features	$10,000

Here are the decision tables that implement 3 steps described by Copilot above:

1) Define Feature Ratios

Decision DefineFeatureRatios [for each Feature in Features]
Conclusion
Ratio of Feature
Value of Feature / Cost of Feature

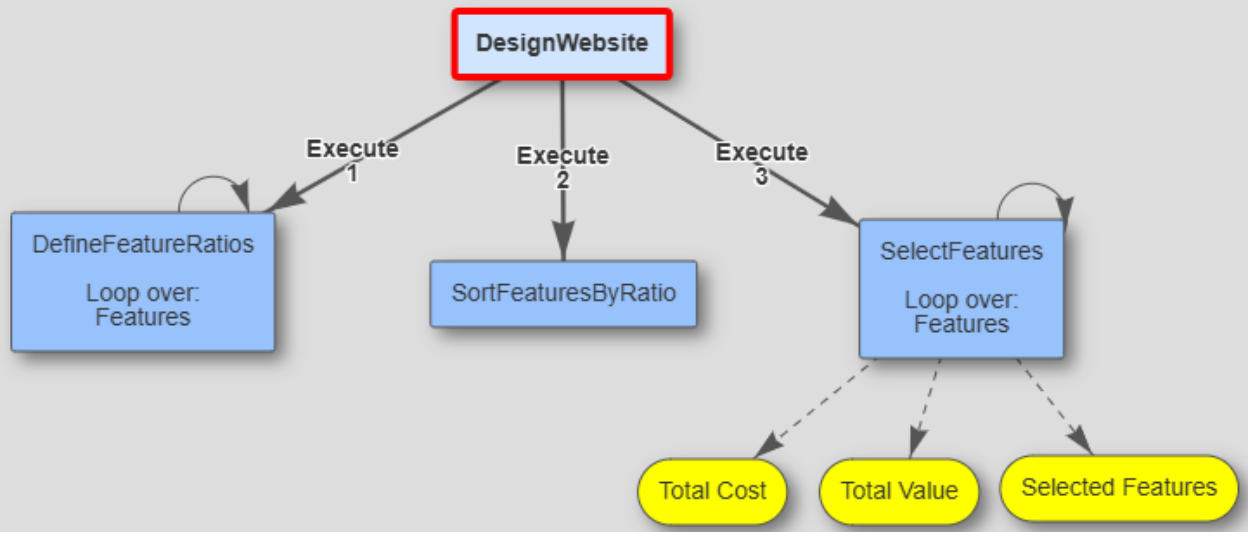
2) Sort Features By Ratio

DecisionTable **SortFeaturesByRatio** [sort Features]		
Condition		**ActionPrefer**
Ratio of Feature1		**Feature**
>	**Ratio of Feature2**	Feature1
=	**Ratio of Feature2**	Feature1
<	**Ratio of Feature2**	Feature2

3) Select Features

Decision **SelectFeatures** [for each Feature in Features]							
Condition		**Conclusion**		**Conclusion**		**Conclusion**	
Budget		**Total Cost**		**Total Value**		**Selected Features**	
>=	**Total Cost + Cost of Feature**	+	**Cost of Feature**	+	**Value of Feature**	**Add**	**Name of Feature**

When I opened this decision model in OpenRules Explorer, I received this decision diagram:



When I executed this decision model, I received the same results as Copilot:

Selected Features: [5, 6, 4, 3, 10, 9, 8]

Total Cost: 10000.0

Total Value: 17.0

I wanted this model to become available as a decision service to work with different inputs (features and budget) provided in the JSON format. So, with one click on the standard "runLocalServer.bat" I deployed this decision model as a REST service on my local server:

Then I tested this decision service using POSTMAN: