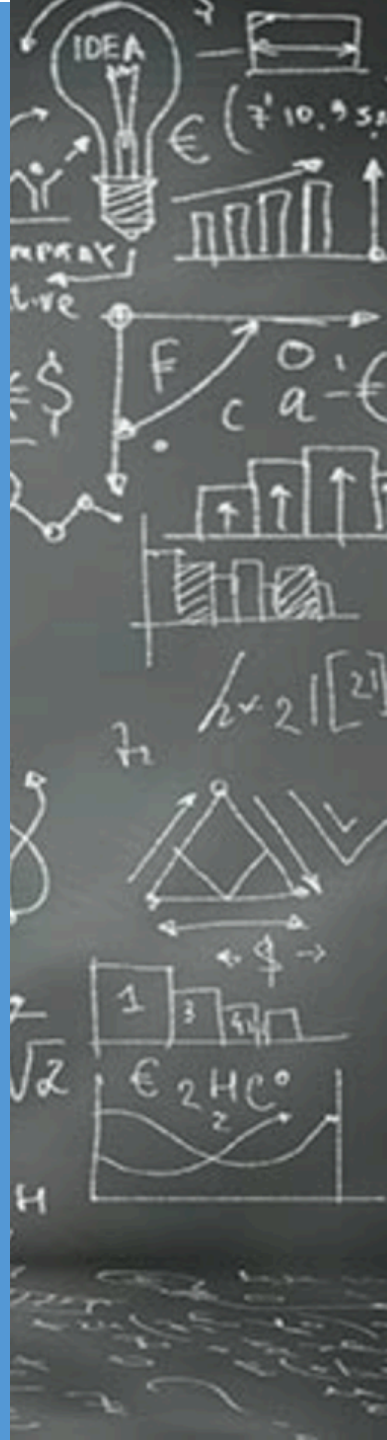
Anomaly models
Methods and examples

Yann Gouedo
Data Scientist Leader – Machine Learning / Artificial Intelligence
Marketing / Risk / Fraud / Maintenance
IBM Certified Senior Data Scientist & IBM Certified Senior Architect



Agenda

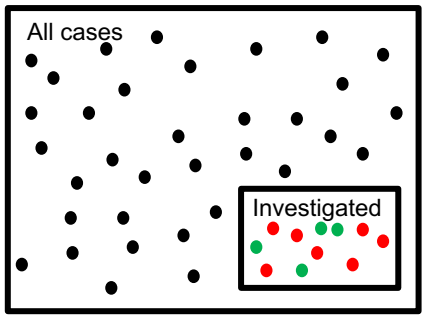
- Introduction

- Anomaly models
  - Creating univariate rules from aggregations
  - Creating multivariate rules from aggregations
  - Rank based anomaly models
  - Cluster based anomaly detection
  - PCA based anomaly detection
  - Permutation based anomaly detection
  - Association model based anomaly detection
  - Nearest neighbor based anomaly detection

- Setting up the data to find the ‘right’ anomalies

Introduction

- In fraud research, known and investigated fraud cases are often challenging to work with.
  - Fraudsters are finding new holes in the system and as such, fraud patterns change all the time. Historic patterns are fixed with a rule based system.
  - The investigated fraud cases are not numerous (in the 1000's, as compared to millions or billions events) and the data regarding those cases is often
    - Not accessible
    - Does not contain the exact state of the data when the case presented itself
    - The timing of data is often unknown (i.e. there can be a police flag, but this is set after fraud was discovered.)



- Anomaly detection model are proven to be helpful
  - Although the outcome of an anomaly model does not necessarily mean fraud, often those cases are interested to look at.

- In this presentation, a series of approaches to anomaly detection is shown.
  - The different approaches serve different types of data and each method has its own pros and cons.
  - For each approach, a industry relevant example is created on synthetic data
    - Synthetic data is used so the examples can be studied without being connected to a database containing actual data
  - The IBM SPSS Modeler streams are provided so the user can gain practical experience

Use cases discussed

What is being investigated	Why this is interesting to look at
Uncommon IP protocol used	Hacking/out of policy behavior
Frequent changes of IP address	Hacking or malware
Exorbitant call duration	Theft or abuse
Number of different ports connected to in a time frame	Hacking/botnet/malware
Number of different server IP address connected to in a time frame	Hacking/botnet/malware
Unexpected increase in revenue	Theft/abuse
Infrequent combinations of IP, App and Parent protocol	Hacking
Upload behavior of customers	Looking for those small group utilizing all bandwidth.
Last day voice vs data usage.	Looking for very infrequently occurring combinations indicating upcoming issues
Last 10 days of voice usage	Theft/abuse
Up vs. download volumes.	Looking for patterns indicating running a commercial webserver or bot/malware activity
Average data usage last month vs last day	Theft/abuse
Data usage over number of consecutive days	Theft/abuse
A set of low level connection variables like 'encrypted yes/no' and 'packet type start/middle/end'	Hacking
Data usage compared against a set of reference group variables.	Theft/abuse

Data requirements and use of the presented techniques

Technique	Continuous data	Categorical data	Scoring options
Creating univariate rules from aggregations	1 variable	1 variable	Creates rules
Creating multivariate rules from aggregations	Create bins first	Multiple variables	Creates rules
Rank based anomaly models	1 variable		Creates rules/scoring
Cluster based anomaly detection	Multiple variables	Multiple variables	Scoring
PCA based anomaly detection	Multiple variables		Scoring
Permutation based anomaly detection	Multiple variables		Scoring
Association model based anomaly detection		Multiple variables	Creates rules/scoring
Nearest neighbor based anomaly detection		Multiple variables	Scoring

* Scoring refers to the method of providing new and unseen data to the model and receiving an anomaly score in return. This typically happens in real-time.

Creating rules from (univariate) aggregations

- **The (type of) data**: both categorical features as well as continuous features
- **The method**
  - For categorical features: create a frequency table. Select common categories as good or uncommon as anomalies
  - For continuous features: create a histogram or boxplot. Select cases outside acceptable range as anomalies
- **Advantages**
  - Very simple and fast
  - The method generates rules that can be used in a rule engine
- **Disadvantages**
  - No multivariate method, only looking at single variables
- **Examples**
  - Looking for IP protocol use, IP changes, and call duration.
  - Observing regular behavior and determining undesirable behavior.
  - Create rules accordingly

Categorical example:
IP protocol use

	ip_protocol	record count
1	1	8563246
2	6	546356
3	17	232443
4	47	3
5	50	1

ALERT IF
Exclusion rule: IP not in (1,6,17)
Inclusion rule: IP in (47,50)

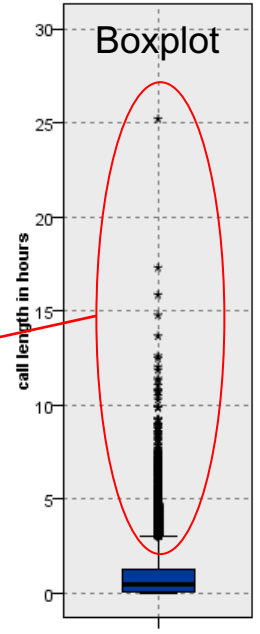
Integer count example:
IP change

	change	Record_Count
1	1	33452
2	2	3951
3	3	1015
4	4	311
5	5	96
6	6	40
7	7	21
8	8	9
9	9	6
10	12	1

ALERT IF:
IP change>10

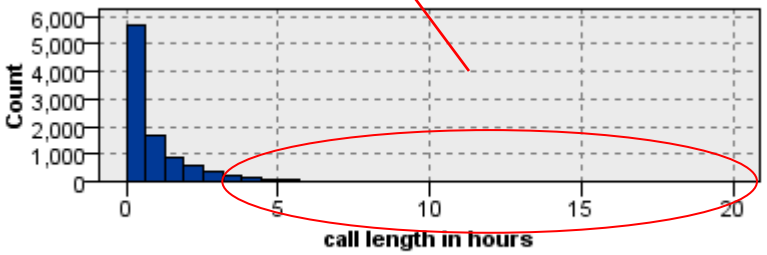
Continuous example:
Call duration in hours

ALERT IF:
Call duration >3.5

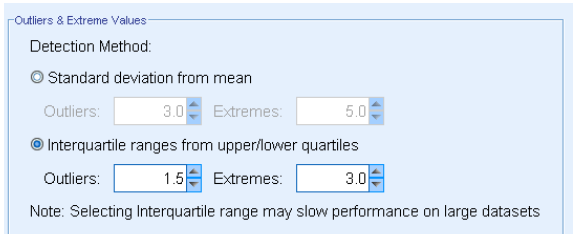


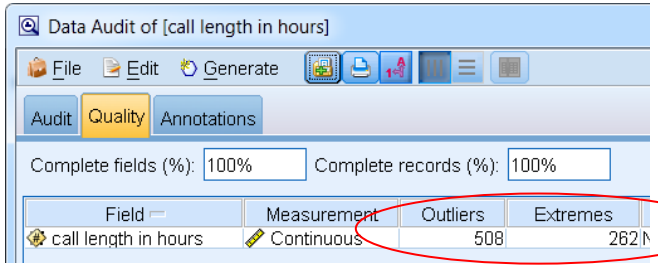
Outliers can be inspected using visual methods, but can be computed using outlier statistics (next slide)

Histogram



Continuous univariate outliers in Modeler

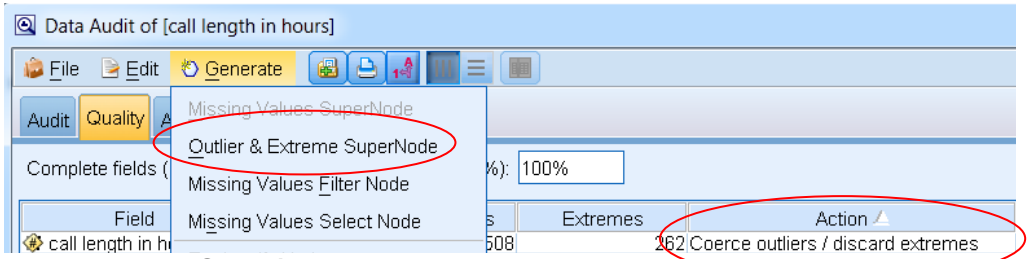




Step 1: Select Data audit

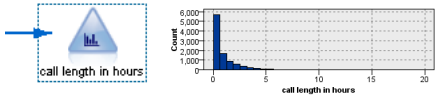
Step 2: Select and configure outlier method

Step 3: Observe outliers and extremes

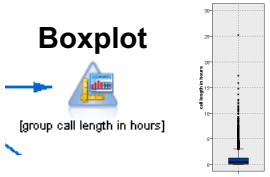


Step 4:
Define outliers and extreme action
Run Outlier and Extreme SuperNode

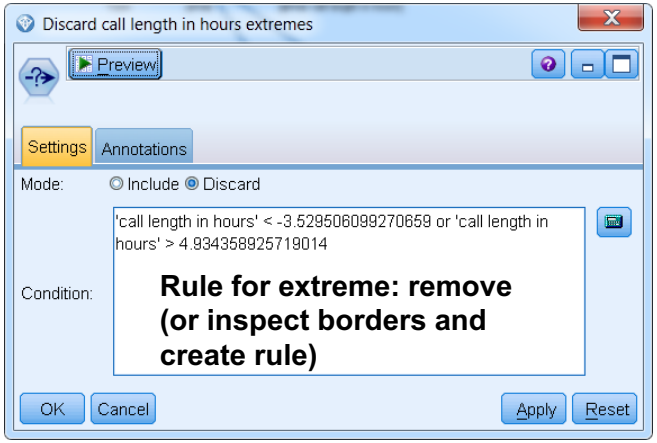
Histogram



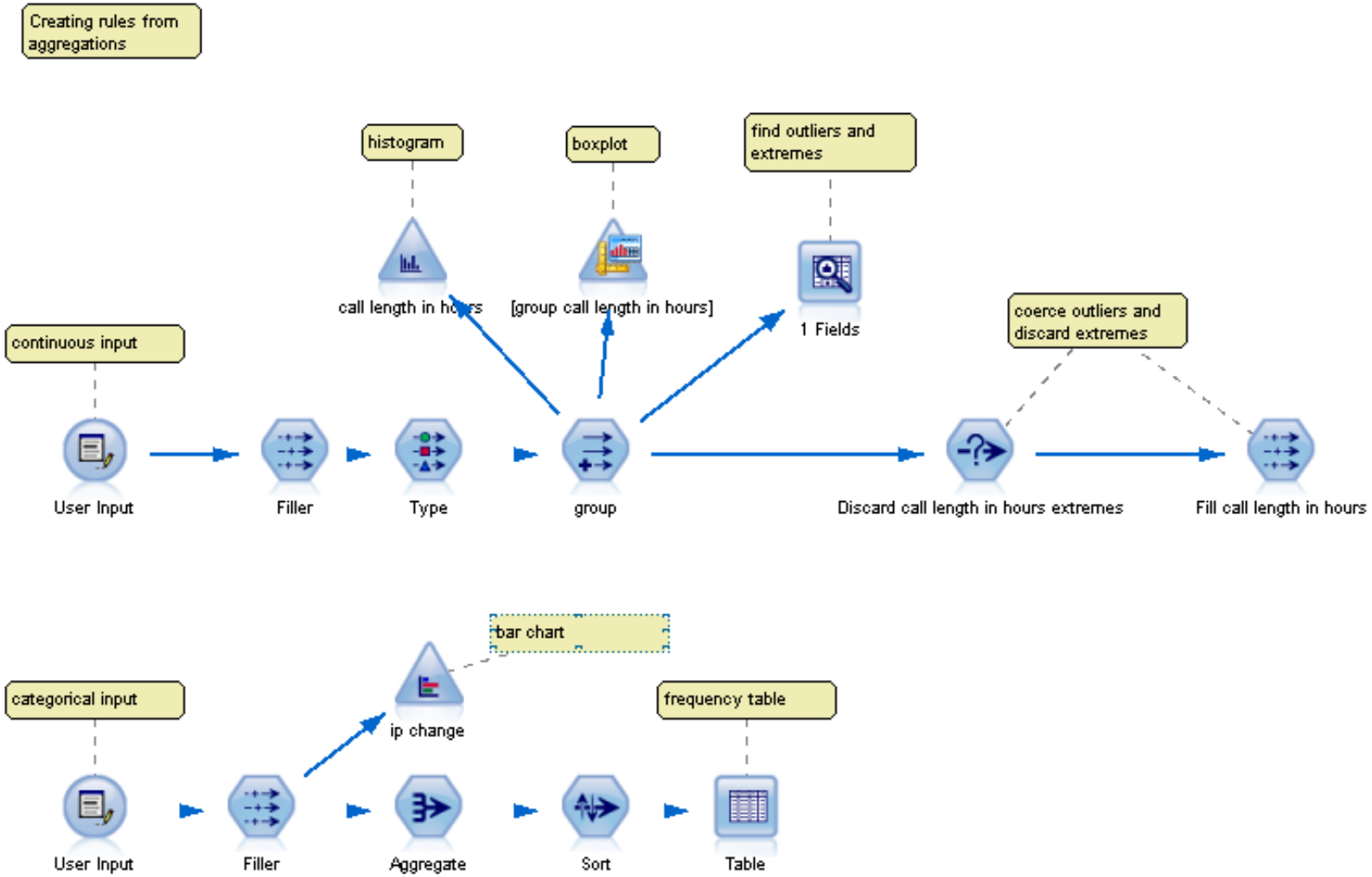
Boxplot



Step 5 (optional): Create the graphics



Example Modeler stream



Direct applications

- # ip changes in last 24 hours
- #different ports (under 1024) targeted in last 24 hours
- #different servers connected to in last 24 hours
- Uncommon IP protocols used
- Call duration in hours
- Monthly revenue to date
- Various other fields like:

SESSN_CONT_CD
HA_MIPADDR_ID
SRVNG_PCF_IPADDR
BS_MSC_ID
FNDMTL_FRAME_SIZE_CD
FRWD_FNDMTL_RC_CD
RVRS_FNDMTL_RC_CD
IP_TCHNY_CD
RELS_CD
AIR_LINK_QOS_NBR

Creating rules from (multivariate) aggregations

- **The (type of) data**: categorical features
- **The method**
  - Create a table by using SQL group by (aggregate) on all fields involved as key, keep the count of occurrence. Select high frequency combinations as good or low frequency combinations as anomalies
- **Advantages**
  - Very simple and fast
  - The method generates rules that can be used in a rule engine
  - Multivariate
- **Disadvantages**
  - No method to indicate which combinations should be looked at. Trial and error.
- **Examples**
  - Without knowing the exact meaning of the protocols, the multivariate aggregation of 3 protocol fields yield a small table with a highly skewed frequency distribution.
    - The uncommon events do **not** imply fraud, but…
    - On a large number of users, events that happens only so infrequently are interesting to look at.

Three protocols and their occurrence

	sn_app_protocol	sn_parent_protocol	ip_protocol	Record_Count
1	0	0	6	810341
2	5	0	6	526738
3	14	0	17	457489
4	6	0	6	343452
5	29	0	6	80636
6	29	0	17	69148
7	0	0	17	55127
8	15	0	1	19160
9	0	0	41	217
10	35	0	47	14
11	35	425151192	47	4
12	33	0	6	3
13	0	0	50	1
14	35	449733800	47	1

A high number here is uncommon

Protocol 35 and 33 are uncommon

Protocol 47 and 50 are uncommon

A = sn_app_protocol
B = sn_parent_protocol
C = ip_protocol

ALERT IF

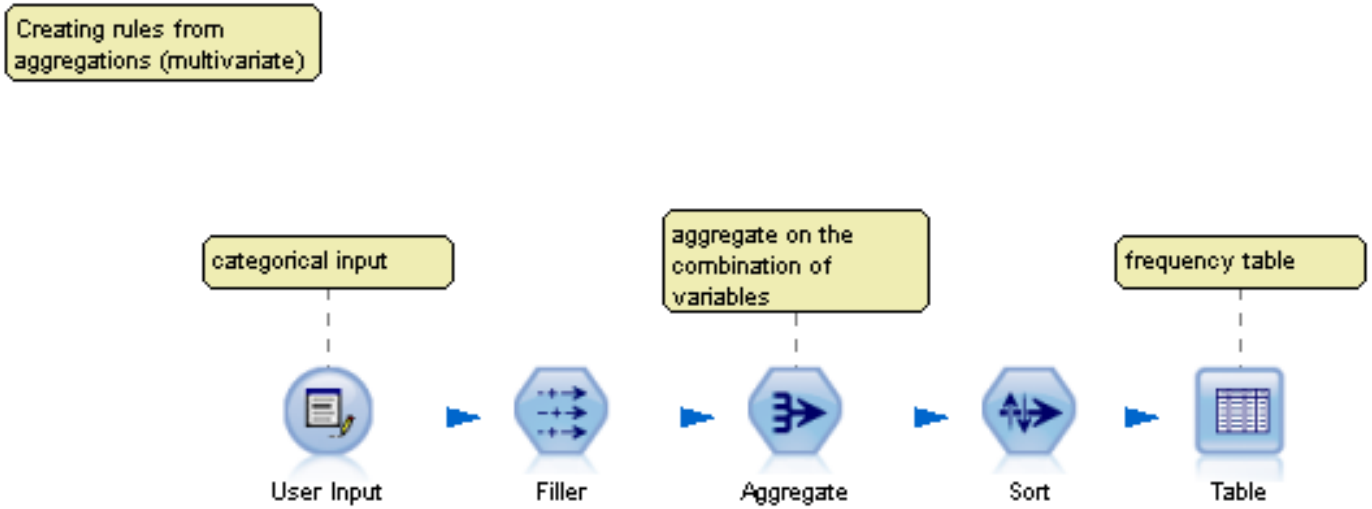
Inclusion scoring

A,B,C in[(0,0,6),(5,0,6),(14,0,17)]

Exclusion scoring

A,B,C in[(35,0,47),(35,425151192,47), etc]

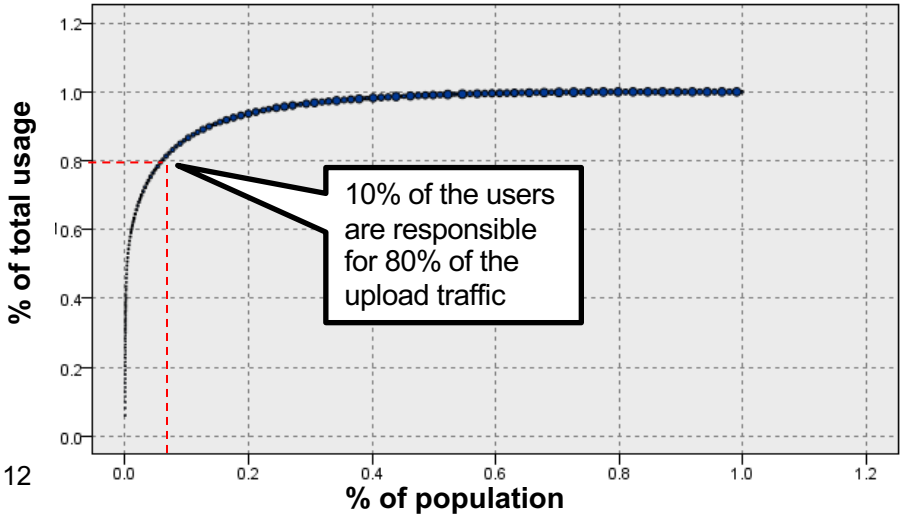
Example Modeler stream



Rank based anomaly models

- **The (type of) data**: continuous features
- **The method**
  - Compute a percentage as individual contribution to the total, sort descending and compute a cumulative percentage for both the location of the individual in the group as well as the cumulative quantity to be measured. This results in statements like “the top 20% uses if responsible for 80% of the use”
- **Advantages**
  - Easy to compute
  - Easy to understand/explain
  - Relates directly to under/over performers
- **Disadvantages**
  - Single variable method
- **Examples**
  - Upload behavior of customers. Looking for those small group utilizing all bandwidth.

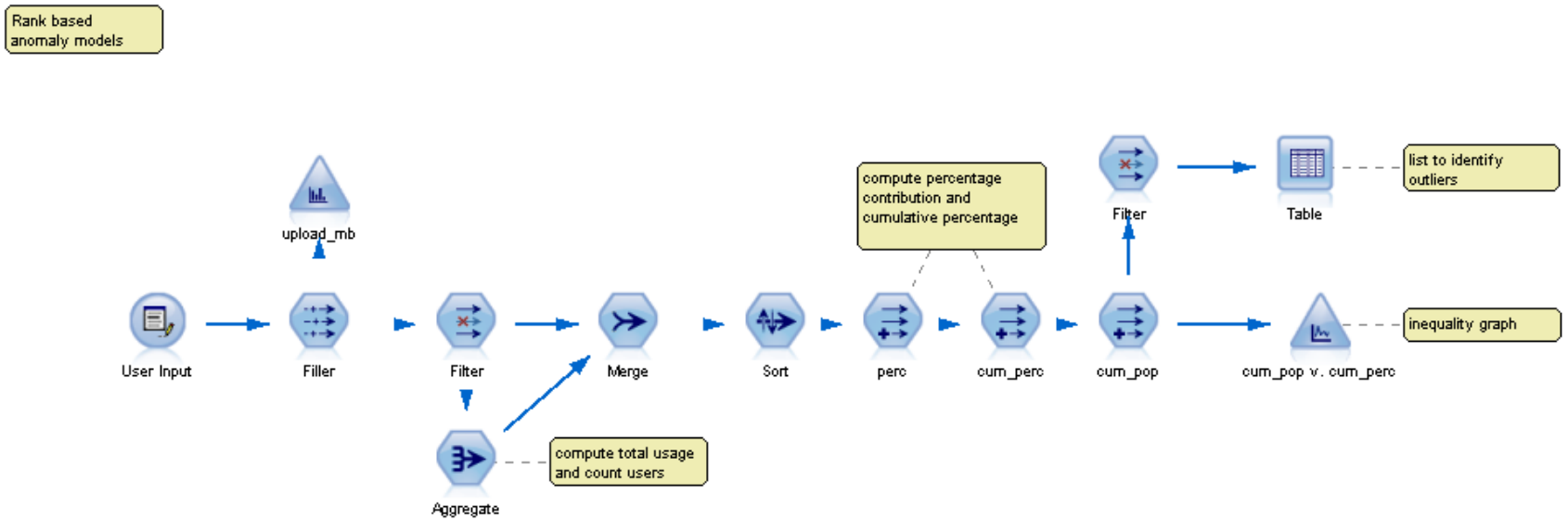
Graphical display of the inequality of use



Selecting the top 25 users responsible for 33% of traffic

	radius_user_name	uplink_Sum	cum_perc	cum_pop
1	9021555178A5@sprintpcs.com	222771679	0.031	0.000
2	1C1448BA1081@sprintpcs.com	211792180	0.060	0.000
3	64A769030EC4@sprintpcs.com	154829715	0.082	0.000
4	00A0D52C7E2D@sprintpcs.com	123956750	0.099	0.000
5	38E7D8705046@sprintpcs.com	115931654	0.115	0.000
6	0015FFF0CD3F@sprintpcs.com	100836909	0.129	0.000
7	001A200B705A@sprintpcs.com	92224145	0.142	0.000
8	00A0D509CFF4@sprintpcs.com	91438022	0.154	0.000
9	64A76918111B@sprintpcs.com	89232499	0.167	0.000
10	64A769029A28@sprintpcs.com	89177760	0.179	0.000
11	7C6193256F3B@sprintpcs.com	89159981	0.192	0.000
12	0018418208FA@sprintpcs.com	89123589	0.204	0.000
13	00A0D52DA085@sprintpcs.com	84386246	0.216	0.000
14	CC7D3720C2BC@sprintpcs.com	82738929	0.227	0.000
15	64A769A7437B@sprintpcs.com	75991749	0.238	0.000
16	00A0D506B5C3@sprintpcs.com	72132060	0.248	0.000
17	64A769067943@sprintpcs.com	69159233	0.257	0.000
18	00A0D50AFECA@sprintpcs.com	68202740	0.267	0.000
19	00A0D50BFBEB@sprintpcs.com	68191982	0.276	0.000
20	00184182D37F@sprintpcs.com	61407026	0.285	0.000
21	D4206D985E9F@sprintpcs.com	59874009	0.293	0.000
22	84518110D704@sprintpcs.com	54692205	0.301	0.000
23	64A7690682FF@sprintpcs.com	50103595	0.307	0.000
24	00A0D5099E37@sprintpcs.com	49777003	0.314	0.000
25	5001BB8EFA8E@sprintpcs.com	45321458	0.321	0.000

Example Modeler stream



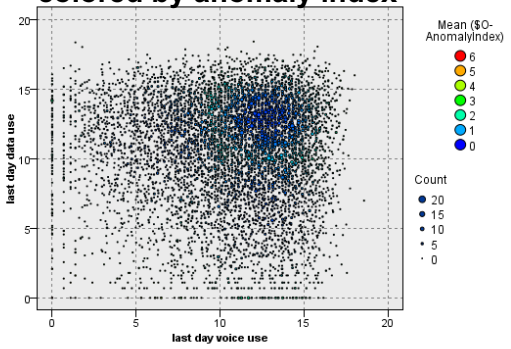
Cluster based anomaly detection

- **The (type of) data**: both continuous and categorical features
- **The method**
  - Specify the anomaly index cutoff or a % anomalies and run the Modeler anomaly detection node. The model will automatically find the number of clusters and assign each case a distance to its nearest cluster. Those far away from any cluster are anomalies.
- **Advantages**
  - Very easy to use
  - Easy to understand/explain
- **Disadvantages**
  - Doesn't account for correlated variables (models clusters as round balls, not cigar shaped)
- **Examples**
  - Last day voice vs data usage. Looking for very infrequently occurring combinations

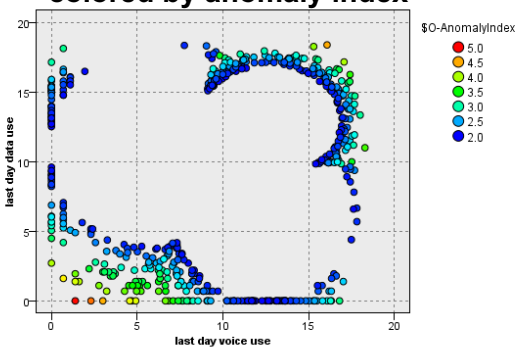
Model summary (found 3 clusters)

- Peer group-1: 4638 records
  - Anomalies: found 212 records from an estimated total of 4,638 records
  - Peer group profile
- Peer group-2: 2525 records
  - Anomalies: found 162 records from an estimated total of 2,525 records
  - Peer group profile
- Peer group-3: 2837 records
  - Anomalies: found 154 records from an estimated total of 2,837 records
  - Peer group profile

All data points colored by anomaly index



Anomalous points colored by anomaly index



Listing of anomalous points

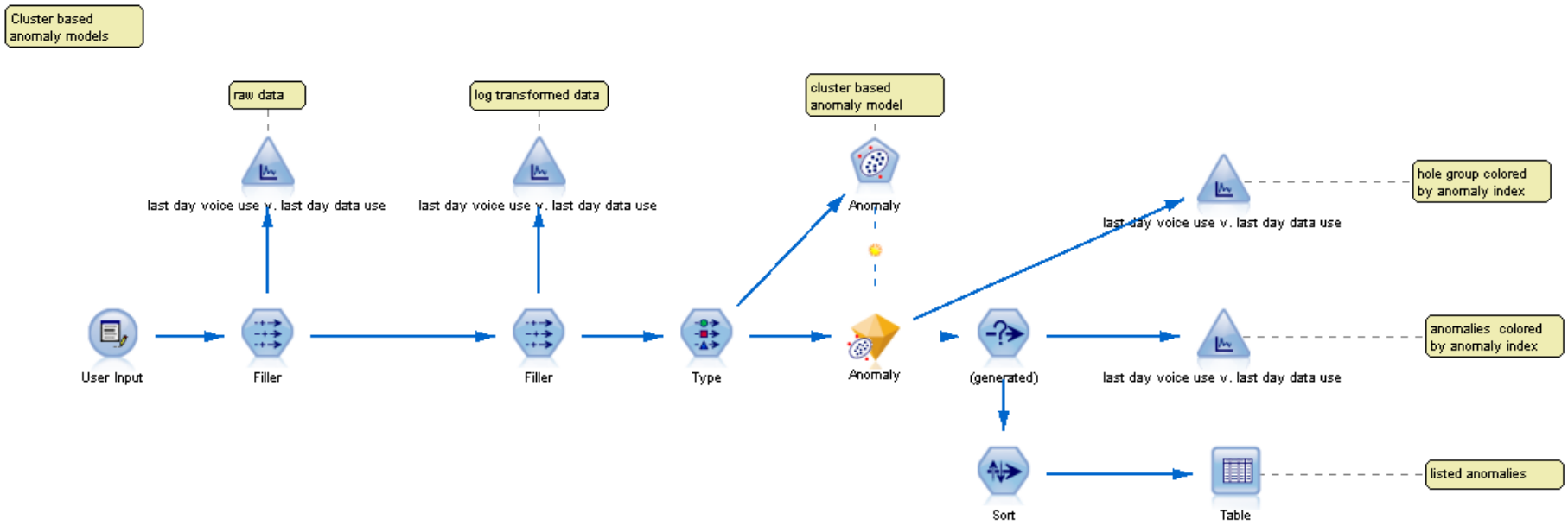
	last day voice use	last day data use	$O-Anomaly	$O-FieldImpact-1	$O-PeerGroup	$O-Field-1	$O-AnomalyIndex
1	0.000	0.000	T	0.700	3	last day data use	5.554
2	1.099	0.000	T	0.764	3	last day data use	5.084
3	17.478	17.872	T	0.532	1	last day data use	5.056
4	1.386	0.000	T	0.781	3	last day data use	4.975
5	0.000	1.386	T		3	last day data use	4.711
6	2.197	0.000	T		3	last day data use	4.705
7	2.197	0.000	T		3	last day data use	
8	5.979	0.000	T	0.572	2	last day voice use	4.271

Anomaly index (larger->more)

Nearest cluster

Field that's being anomalous

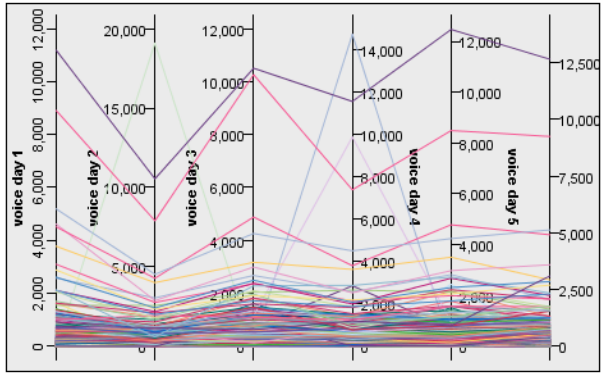
Example Modeler stream



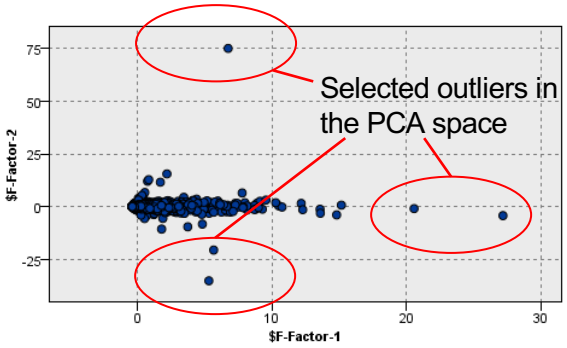
PCA based anomaly detection

- **The (type of) data**: multiple/many continuous features, preferable measuring the same quantity.
- **The method**
  - Execute a Principal Component Analysis (PCA) on the features, creating 2 (or more) factors. In case of 2 factors, display the data in a scatterplot, the axes being the factors. Observe and select outliers. In case of more factors were used, use one of the other described anomaly detection methods to automatically select those.
- **Advantages**
  - Well suited for larger number of variables
  - Not a lot of tuning required
  - Visual inspection of outliers
  - Can be input to other methods
- **Disadvantages**
  - Manual selection of outliers or need other methods for auto-selection
- **Examples**
  - Six days of voice usage: looking for patterns where something unexpected happens (greatly increasing minutes)

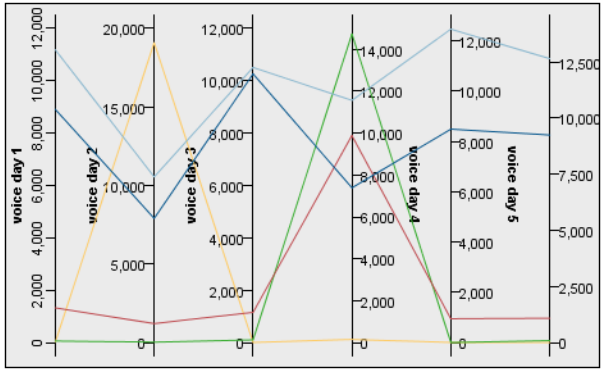
Sample of the data



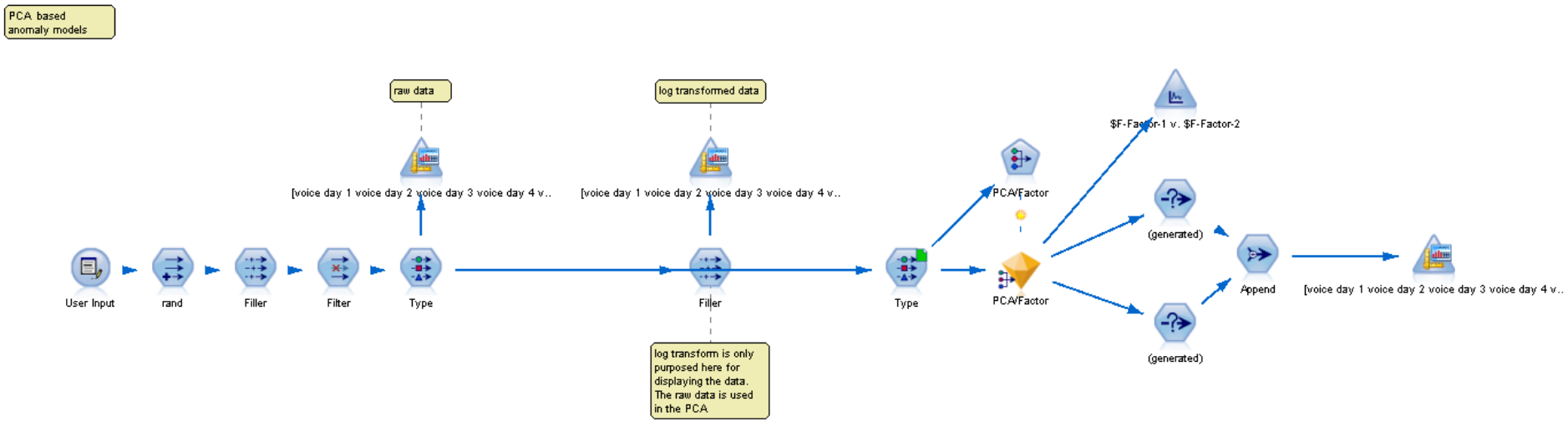
PCA view on the data



Selected anomalies



Example Modeler stream



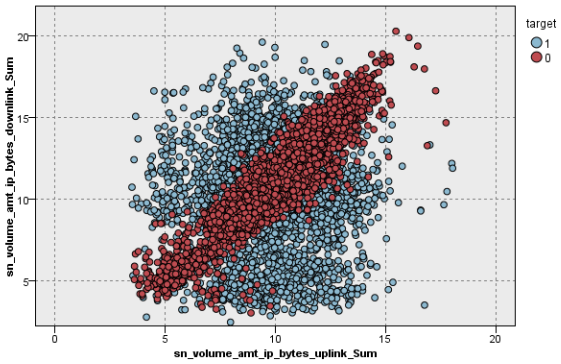
Permutation based anomaly detection

- **The (type of) data**: multiple/many continuous features, preferable measuring the same quantity.
- **The method**
  - Take the observed data and call it non-anomalous. Copy the dataset and permute the rows such that all relations between the variables are destroyed. Call this the anomalous data. Build a binary classifier to detect the difference between the non-anomalous and the anomalous data.
- **Advantages**
  - Results in a probability to be anomalous
  - Working based on relations between variables
- **Disadvantages**
  - Permuting larger number of variables can be tedious
- **Examples**
  - Up vs. download volumes.
  - Looking for patterns outside of normal use that indicate
    - Running a commercial webserver
    - Bot/malware activity

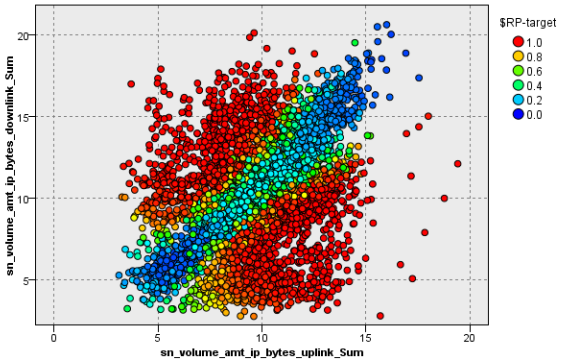
Listing anomalous cases

	radius_user_name	sn_volume...	sn_volume_...	$RP-target
1	00A0D52D9C38@sprintpcs.com	12.831	8.825	1.000
2	F8DB7F30A19E@sprintpcs.com	13.572	9.740	1.000
3	8C71F8586652@sprintpcs.com	12.277	8.944	0.999
4	D487D8D836EB@sprintpcs.com	12.947	9.651	0.999
5	64A769017C7D@sprintpcs.com	11.593	8.198	0.999
6	5001BB8074CF@sprintpcs.com	13.424	10.124	0.999
7	8C71F85D02F3@sprintpcs.com	11.631	8.472	0.999
8	64A76939036C@sprintpcs.com	13.815	10.616	0.999
9	64A76905344B@sprintpcs.com	11.567	8.612	0.999
10	D0176AA87E98@sprintpcs.com	13.433	10.333	0.999
11	845181148D45@sprintpcs.com	13.459	10.357	0.999
12	D487D8D69001@sprintpcs.com	13.955	10.745	0.999
13	00A0D52CE9BB@sprintpcs.com	10.996	7.694	0.999
14	64A769339D75@sprintpcs.com	13.103	10.087	0.999
15	38ECE4905C6E@sprintpcs.com	13.614	10.500	0.999
16	38ECE496869C@sprintpcs.com	11.886	9.052	0.999
17	38ECE4981C10@sprintpcs.com	11.617	8.897	0.998
18	00A0D5261A06@sprintpcs.com	10.651	7.434	0.998
19	5001BB8568C7@sprintpcs.com	14.164	11.068	0.998
20	00A0D50D19C1@sprintpcs.com	16.281	12.611	0.998

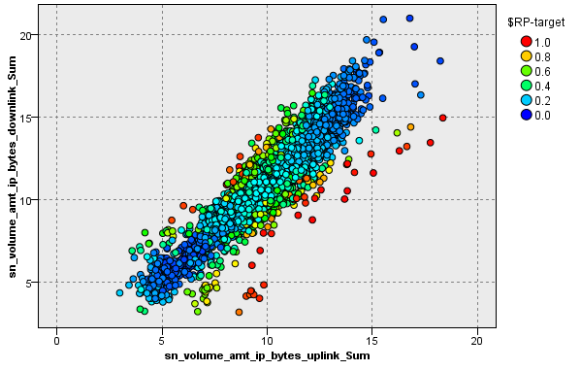
Training data:
Red: observed data
Blue: permuted data



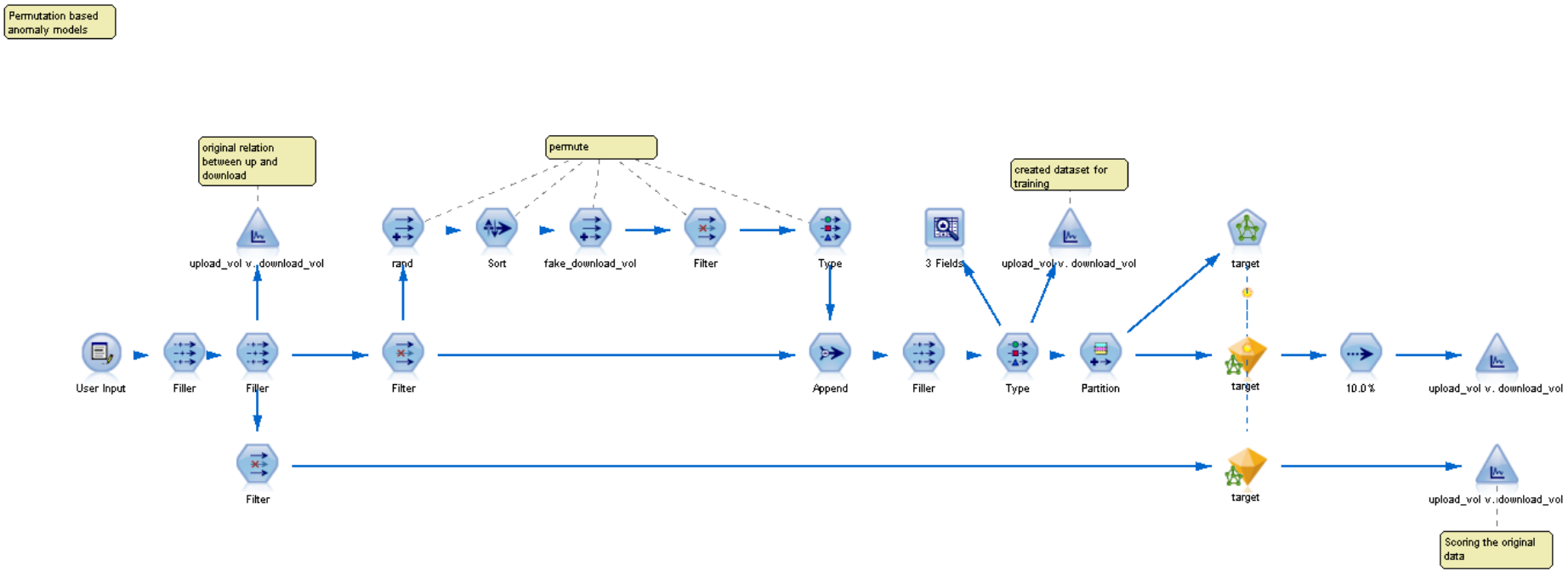
Training data:
Data colored by propensity to be anomalous (red is high)



Scoring new data:
Data colored by propensity to be anomalous (red is high)



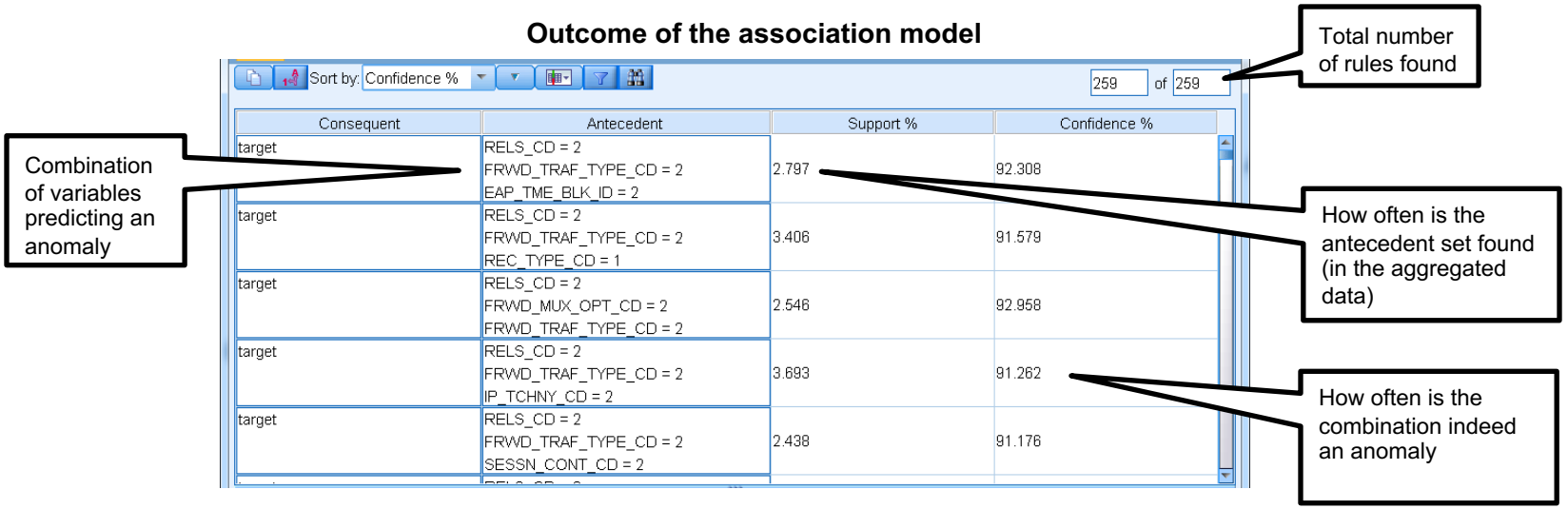
Example Modeler stream



Association model based anomaly detection

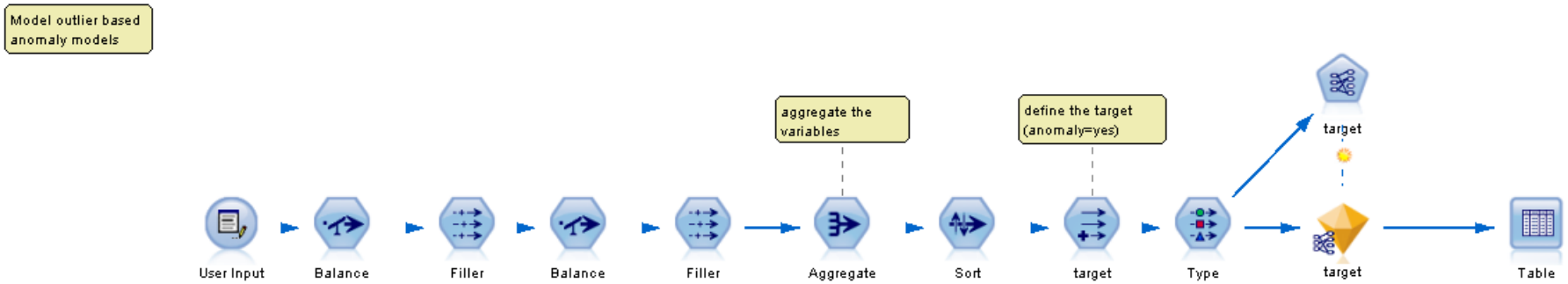
- **The (type of) data**: multiple/many categorical features with low (<10) number of categories
- **The method**
  - Aggregate (group by) all the features simultaneously, retain record count. Define outliers as counts below desired threshold. Create indicator for yes/no outlier and fit an association model with consequent set to the outlier indicator and antecedents set to the categorical features.
- **Advantages**
  - The association model creates rules that can be used in a rule engine.
  - The association model will find combinations of variables that occur in the defined outlier set (compare this to the multivariate aggregation method, where it was unclear what variables needed to be taken into the aggregation)
- **Disadvantages**
  - Model result in overlapping rules
- **Examples**
  - A set of low level connection variables are chosen to demonstrate the principle. The variables represent the administrative details of data packets flowing over routers. Variables included are ‘encrypted yes/no’, ‘packet type start/middle/end’, ‘frame size 1/2/3’ etc.

Outcome of the association model

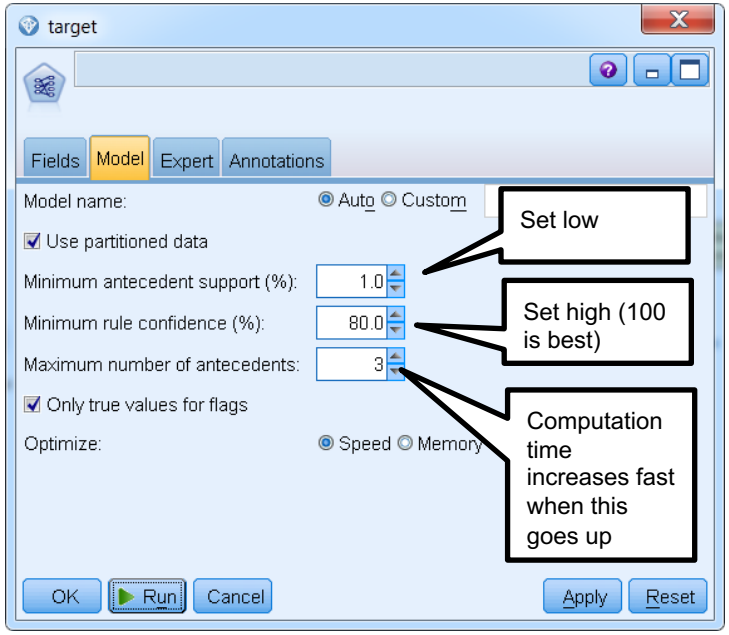


Consequent	Antecedent	Support %	Confidence %
target	RELS_CD = 2 FRWD_TRAF_TYPE_CD = 2 EAP_TME_BLK_ID = 2	2.797	92.308
target	RELS_CD = 2 FRWD_TRAF_TYPE_CD = 2 REC_TYPE_CD = 1	3.406	91.579
target	RELS_CD = 2 FRWD_MUX_OPT_CD = 2 FRWD_TRAF_TYPE_CD = 2	2.546	92.958
target	RELS_CD = 2 FRWD_TRAF_TYPE_CD = 2 IP_TCHNY_CD = 2	3.693	91.262
target	RELS_CD = 2 FRWD_TRAF_TYPE_CD = 2 SESSN_CONT_CD = 2	2.438	91.176

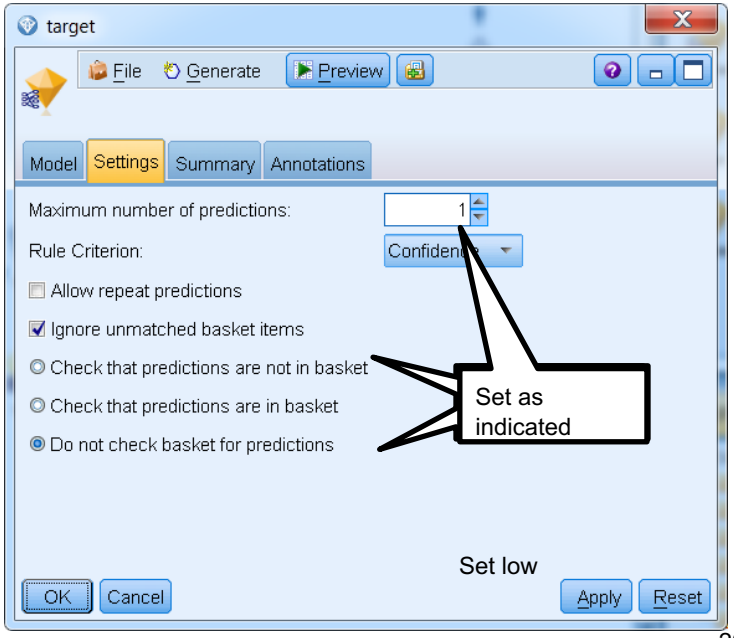
Example Modeler stream



Training settings of the association model

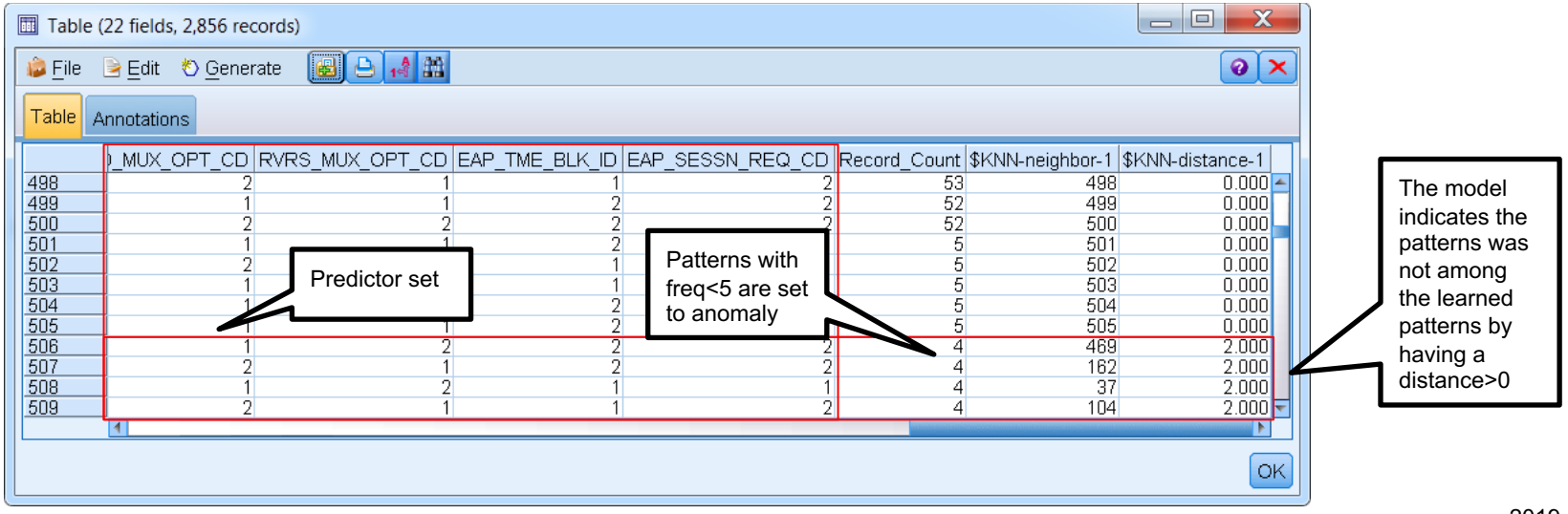


Scoring settings of the association model

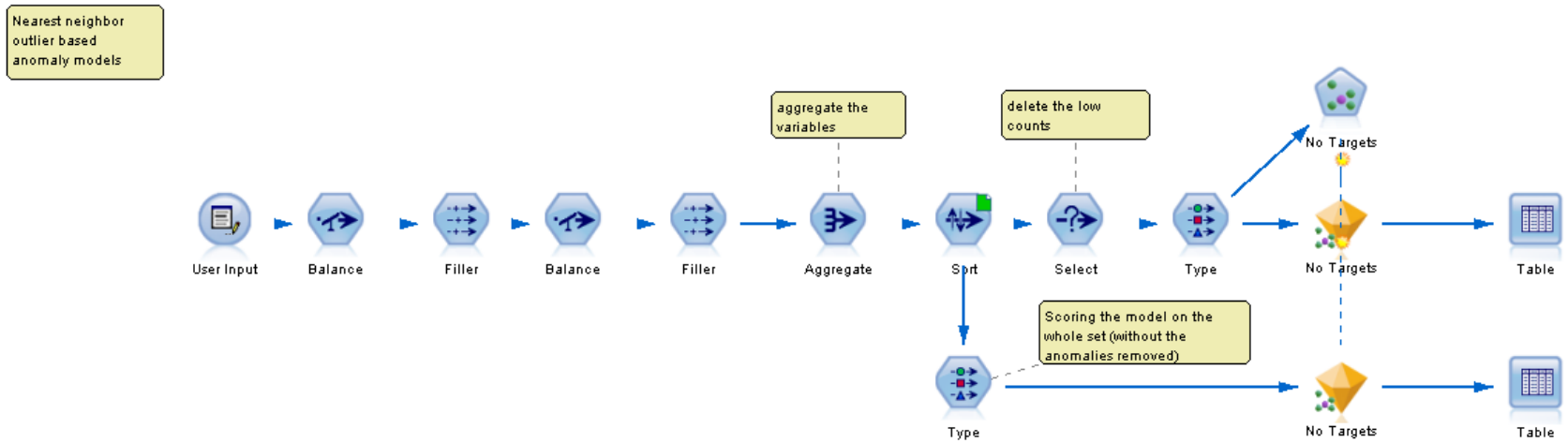


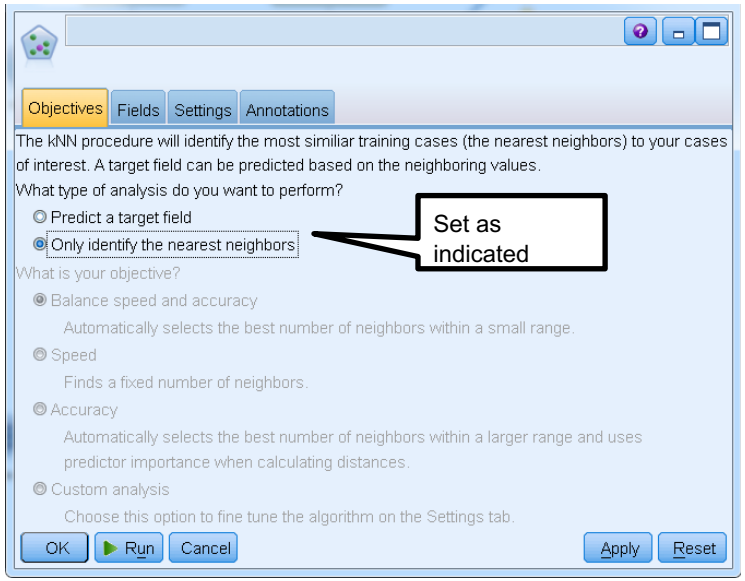
Nearest neighbor based anomaly detection

- **The (type of) data**: multiple/many categorical features with low (<10) number of categories
- **The method**
  - Aggregate (group by) the all features simultaneously, retain record count. Define outliers as counts below desired threshold and delete those cases. Run a 1 KNN model on the remaining set. If scored on a new set, a distance larger than 0 will indicate that the pattern was not in the aggregated set.
- **Advantages**
  - The knn model will find patterns that have never been observed
- **Disadvantages**
  - The data needs to collapse to a low (<1000) number of patterns in order for the model to be efficient
- **Examples**
  - A set of low level connection variables are chosen to demonstrate the principle. The variables represent the administrative details of data packets flowing over routers. Variables included are 'encrypted yes/no', 'packet type start/middle/end', 'frame size 1/2/3' etc.



Example Modeler stream





Other settings:

- No normalization
- K = 1
- City block distance

Setting up the data to find the ‘right’ anomalies

- It will be rare that the raw data contains the features for finding anomalies that you are interested in.
- Often some aggregation is needed
  - Usages over a period of time is compared among users
- Setting up reference data can be very useful
  - Compare today’s volume against a monthly average
- Anomalies can be found within peer groups
  - Building a model that contains both categorical features as well as continuous features is more difficult
  - An approach is to define peer groups from categorical features and within each cluster define a continuous data anomaly model

Thank You

Yann Gouedo
Data Scientist Leader – Machine Learning / Artificial Intelligence
Marketing / Risk / Fraud / Maintenance
IBM Certified Senior Data Scientist & IBM Certified Senior Architect