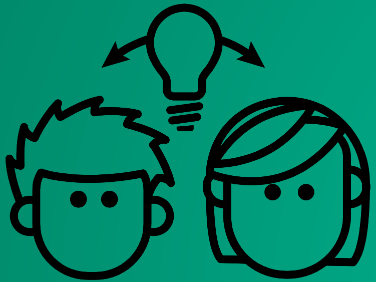
3. DISCOVER

Pattern extraction, segmentation, baselining, anomalies

Cybersecurity Data Science (CSDS)

TOPIC
1. FRAME
2. DATA
3. DISCOVER
4. DETECT
5. DEPLOY



Idea Exchange

What do you feel is the biggest challenge in deploying cybersecurity analytics?

Ponemon INSTITUTE

Challenges Preventing Successful Use of Cybersecurity Analytics*

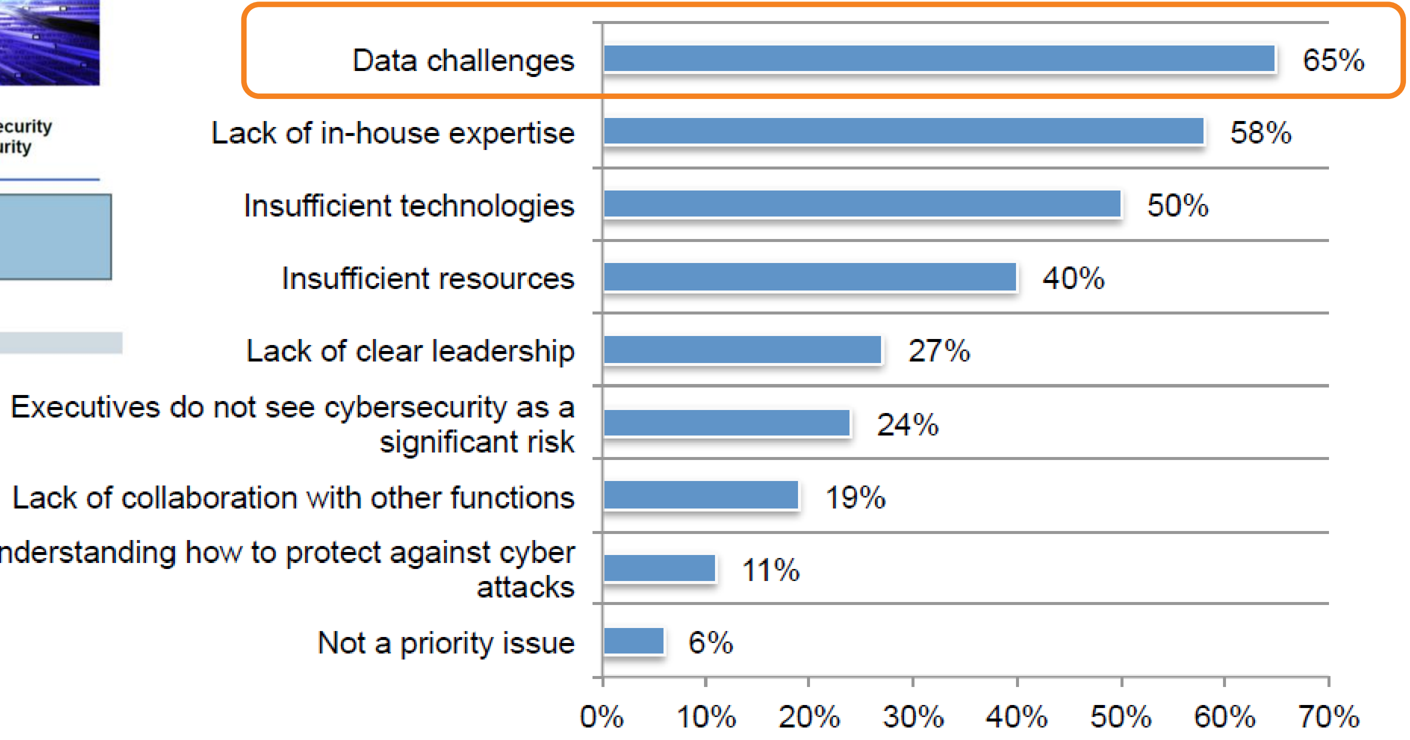
When Seconds Count: How Security Analytics Improves Cybersecurity Defenses

Sponsored by SAS Institute

Independently conducted by Ponemon Institute LLC

Publication Date: January 2017

Ponemon Institute© Research Report

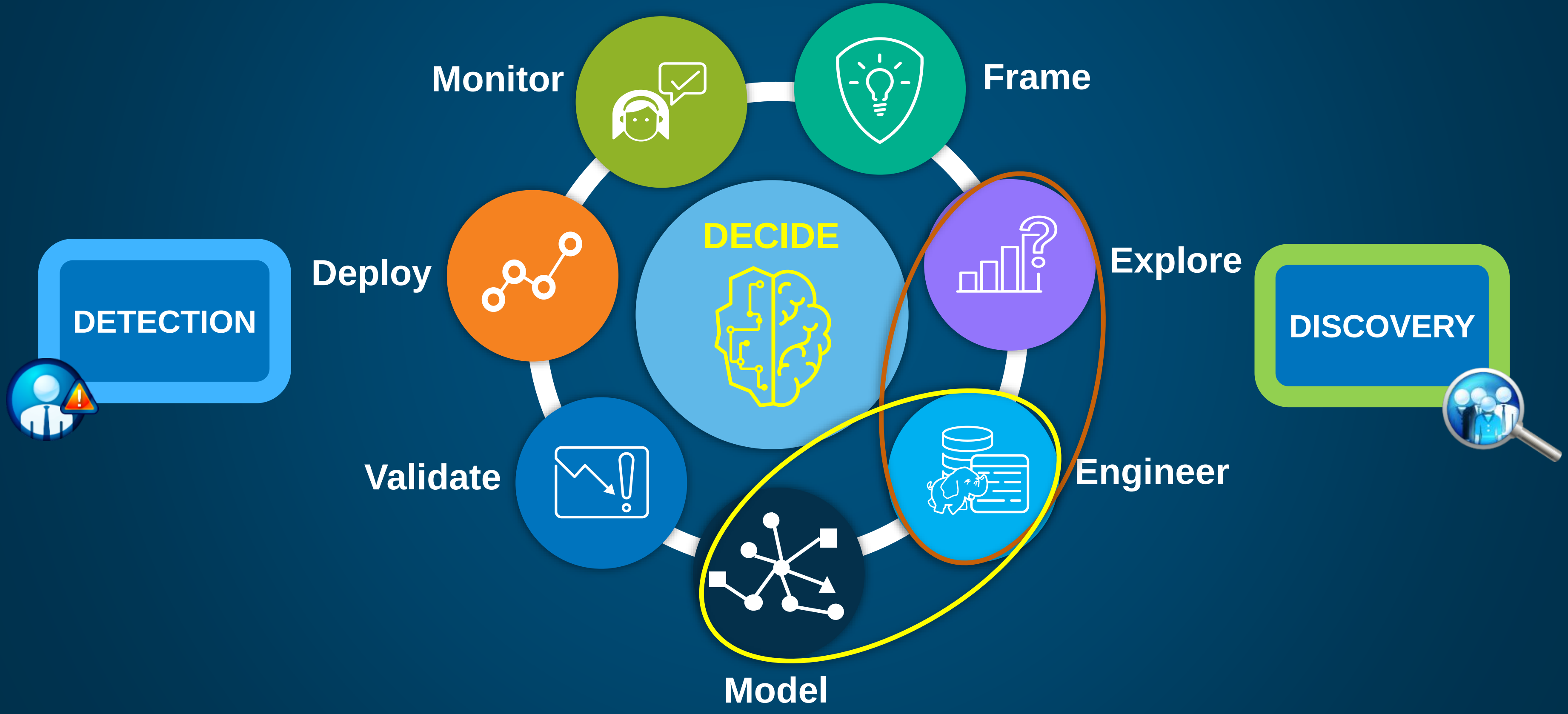


https://www.sas.com/en_us/whitepapers/ponemon-how-security-analytics-improves-cybersecurity-defenses-108679.html

** Survey of 621 global IT security practitioners*

Learning Objectives

Cybersecurity Data Science (CSDS) Lifecycle



Objectives of Cybersecurity Pattern Extraction

Baselining Using Unsupervised Machine Learning

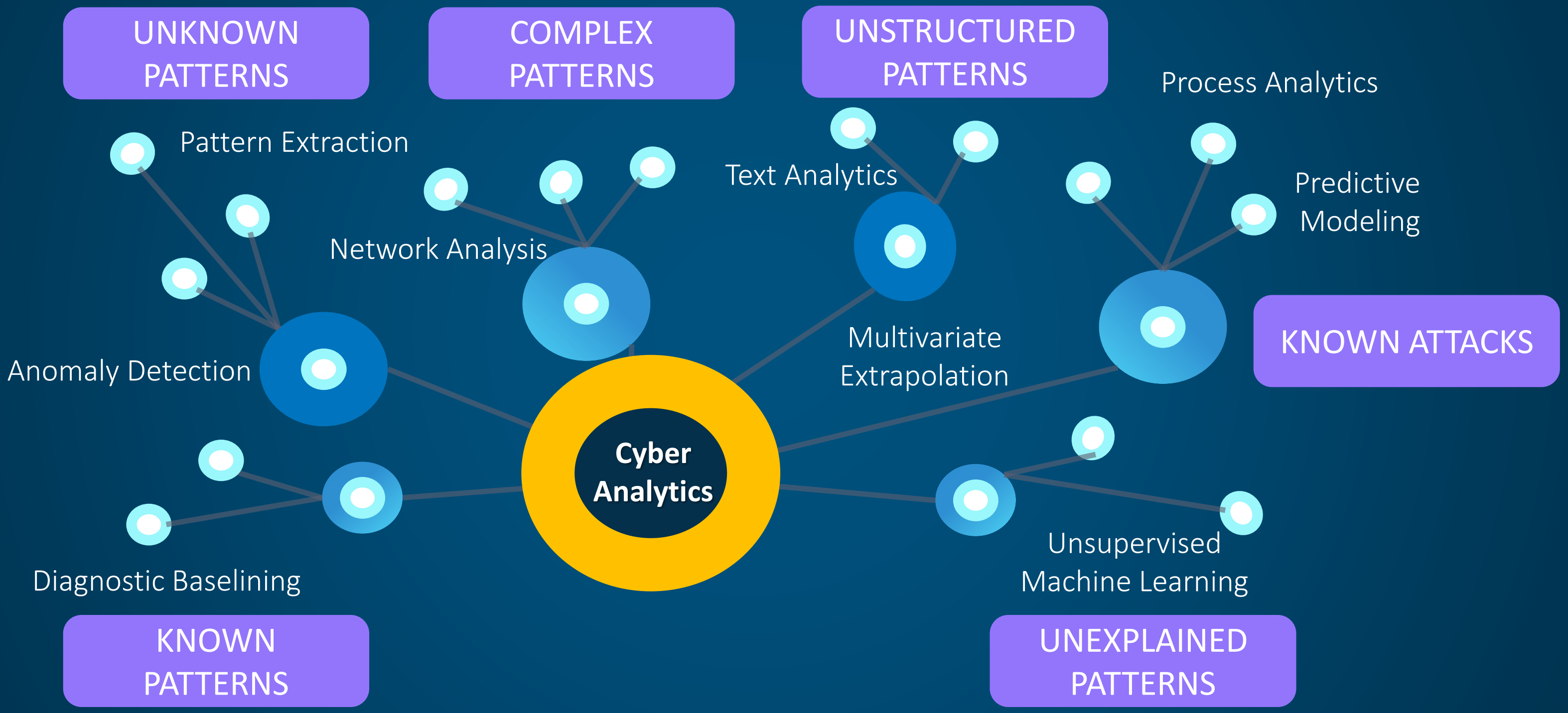
- Exploring statistical aspect and relations in data
  - Intuition versus testable hypothesizes and statistical patterns
- Hands on with data analytics tools
- Extracting groups from data as *statistical* categories
  - Apply unsupervised machine learning (cluster analysis) to extract statistical patterns / baselines
- Establishing a foundation for prediction

CSDS Process

Unified Orchestration

Machine Learning

CSDS: Diverse Analytics Toolkit



Machine Learning Model = Active Data Vehicle

Role of Algorithms

SUPERVISED LEARNING
Teaching by example.

SEMI-SUPERVISED LEARNING
A bit of both

UNSUPERVISED LEARNING
No answer key is provided.

REINFORCEMENT
AI becomes reality.

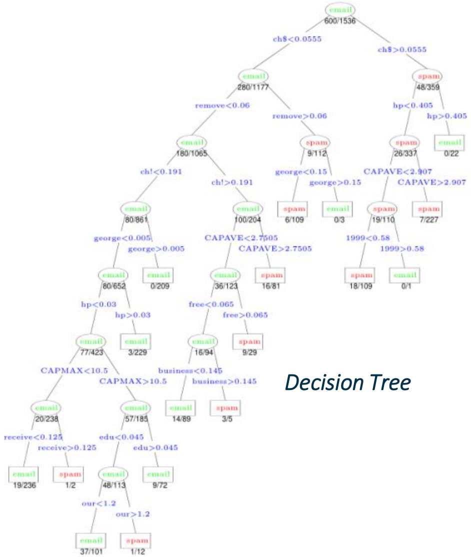
Machine Learning

Descriptive (Unsupervised)

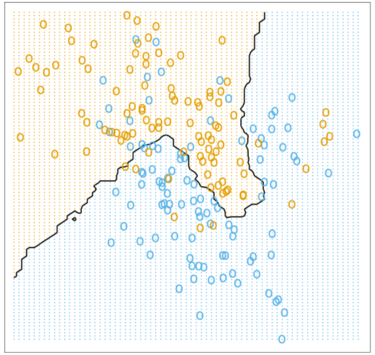
- Cluster analysis
- Factor analysis
- Self-Organizing Maps (SOMs)

Predictive (Supervised)

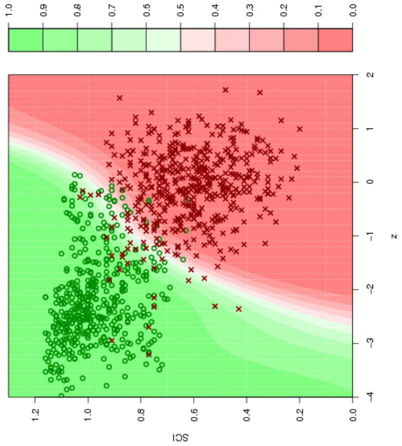
- K-Means
- Decision Trees (DT)
  (random forests, boosted trees)
- Naïve Bayes classifier
- Neural networks
- Support Vector Machine (SVM)
- Ensembles / Ensemble Learning



Decision Tree



k-nearest neighbors



Support Vector Machines

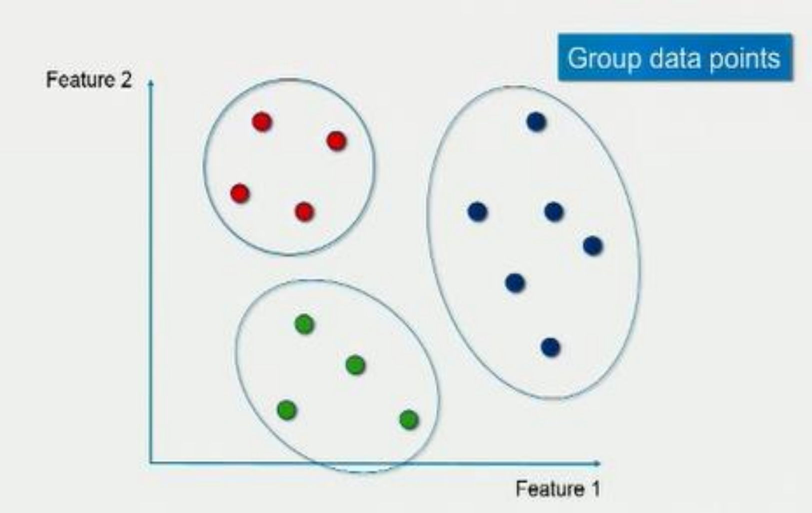
Unsupervised Machine Learning

CLUSTER ANALYSIS

Which group has the most blue marbles?

Descriptive analytics

Application of cluster analysis (one of a number of UNSUPERVISED machine learning techniques)



Once segmented into STATISTICAL categories, it becomes much easier to profile the groups and to detect in-group anomalies

Machine learning tasks

A priori rules

Clustering

Dimension Reduction

k-means clustering

Factorization

PCA

Network Analysis

Affinity Analysis

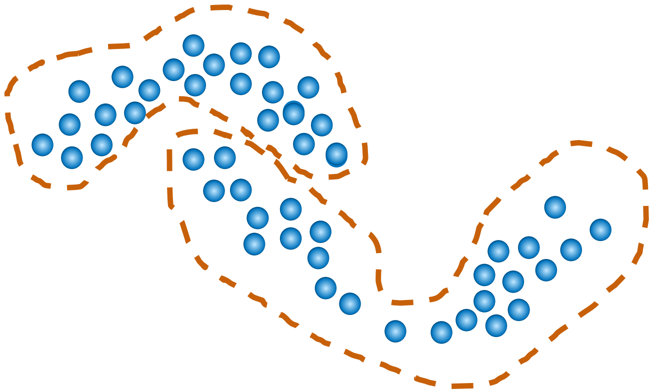
Markov Models

Unsupervised Learning

- No target is defined.
- Data is unlabeled. Draws inferences and conclusions based solely on analyzing input data.

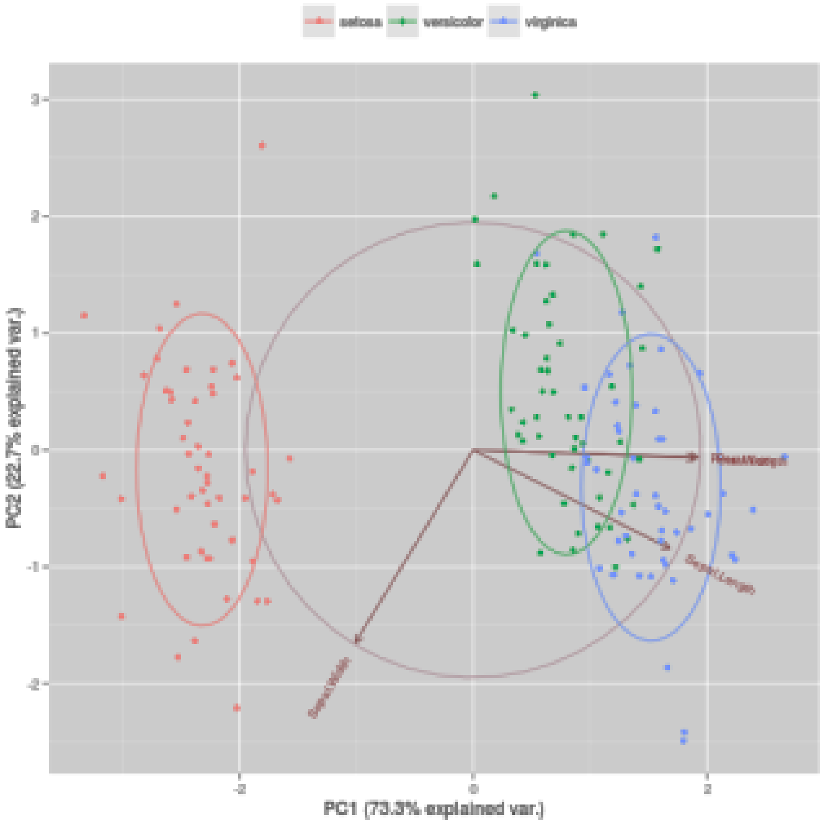
Considerations

- Overcomes ‘known patterns’ issue
- More complex to understand
- Patterns are everywhere



Unsupervised Machine Learning

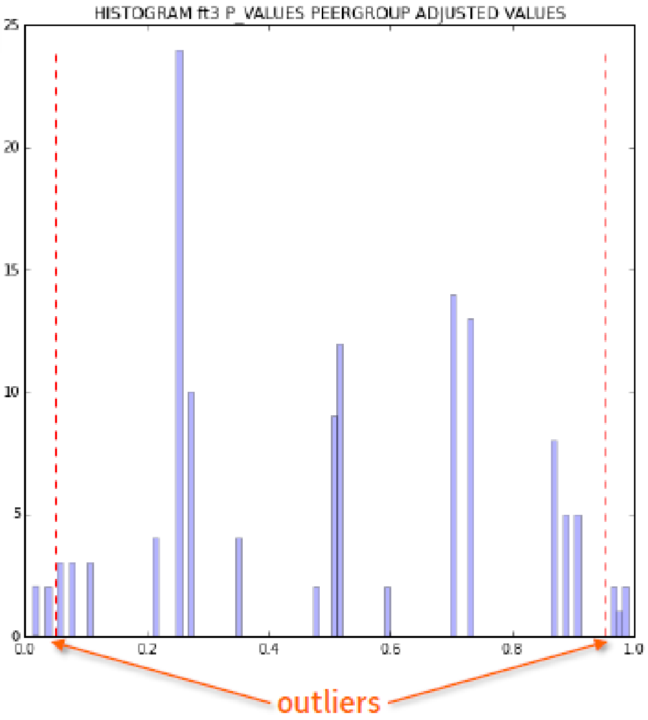
=> Pattern Extrapolation



- Unsupervised techniques
  - No prior categorization scheme
  - Goal is to extract ***statistically meaningful segments***

- Examples…
  - Multivariate analysis – e.g. PCA
  - Cluster Analysis
  - Neural networks

De Facto Approach: Organizational Groups



Deterministic (organizational groupings)

Nominal Logistic Fit for peerGroupCode

Whole Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	3305.967	440	6611.934	<.0001*
Full	24287.706			
Reduced	27593.673			

RSquare (U)	0.1198
AICc	49536.7
BIC	52845
Observations (or Sum Wgts)	10740

Little improvement over random groupings

Measure	Training	Definition
Entropy RSquare	0.1198	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.4624	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/...
Mean -Log p	2.2614	∑ -Log(ρ[j])/n
RMSE	0.8581	√ ∑(y[j]-ρ[j]
Mean Abs Dev	0.8507	∑ |y[j]-ρ[j]|
Misclassification Rate	0.7438	∑ (ρ[j]...
N	10740	n

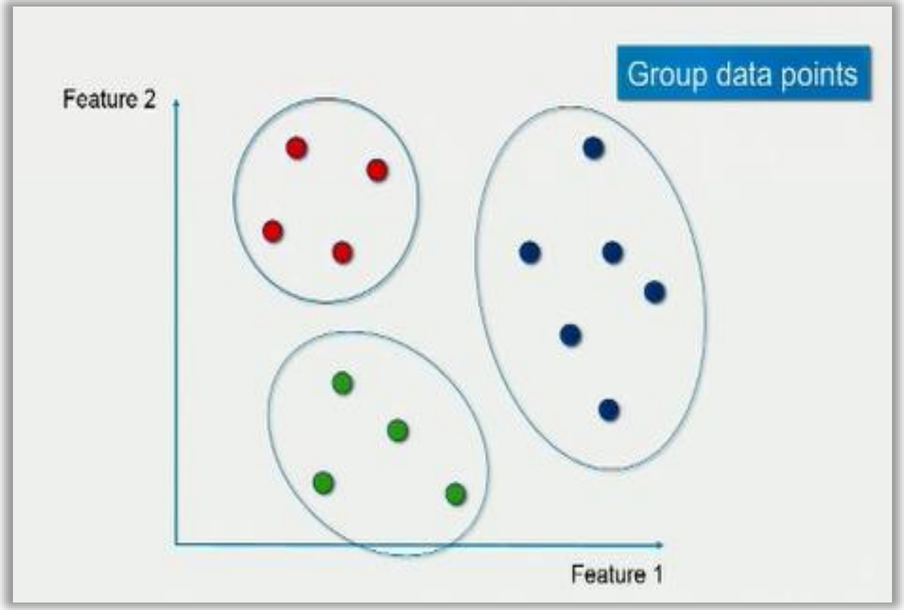
~25% correct classification rate

Lack Of Fit

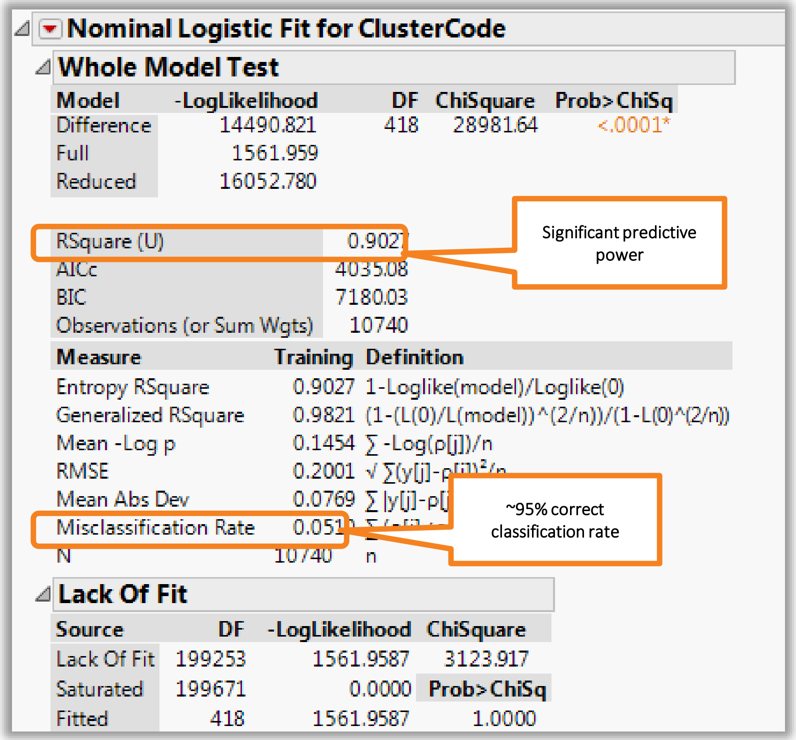
Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	209740	24210.109	48420.22
Saturated	210180	77.598	Prob>ChiSq
Fitted	440	24287.706	1.0000

Unsupervised Machine Learning

Cluster Analysis



Derived (computer-generated clusters)



Nominal Logistic Fit for ClusterCode

Whole Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	14490.821	418	28981.64	<.0001*
Full	1561.959			
Reduced	16052.780			

RSquare (U)	0.9027
AICc	4035.08
BIC	7180.03
Observations (or Sum Wgts)	10740

Measure	Training	Definition
Entropy RSquare	0.9027	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.9821	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.1454	∑ -Log(ρ[j])/n
RMSE	0.2001	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.0769	∑ |y[j]-ρ[j]
Misclassification Rate	0.051	∑
N	10740	n

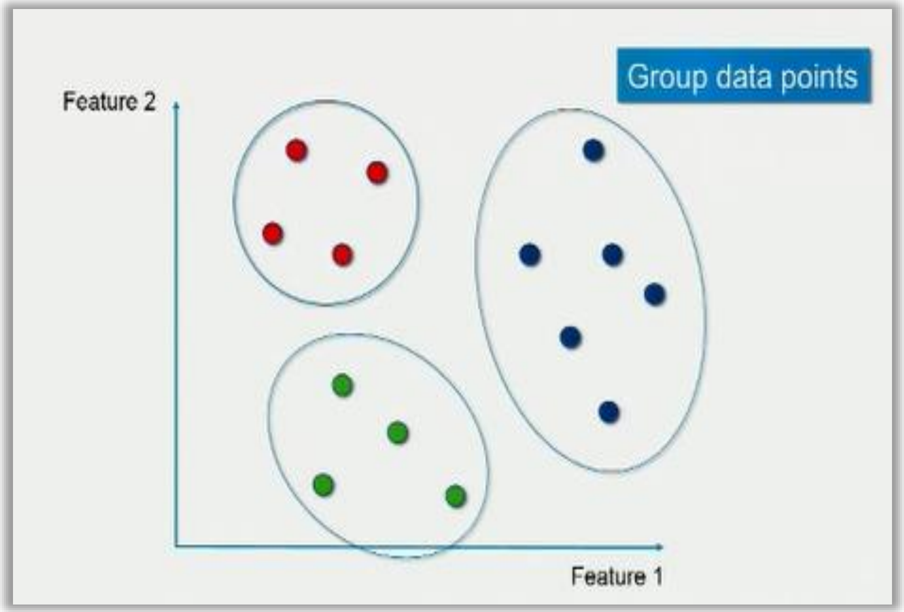
Lack Of Fit

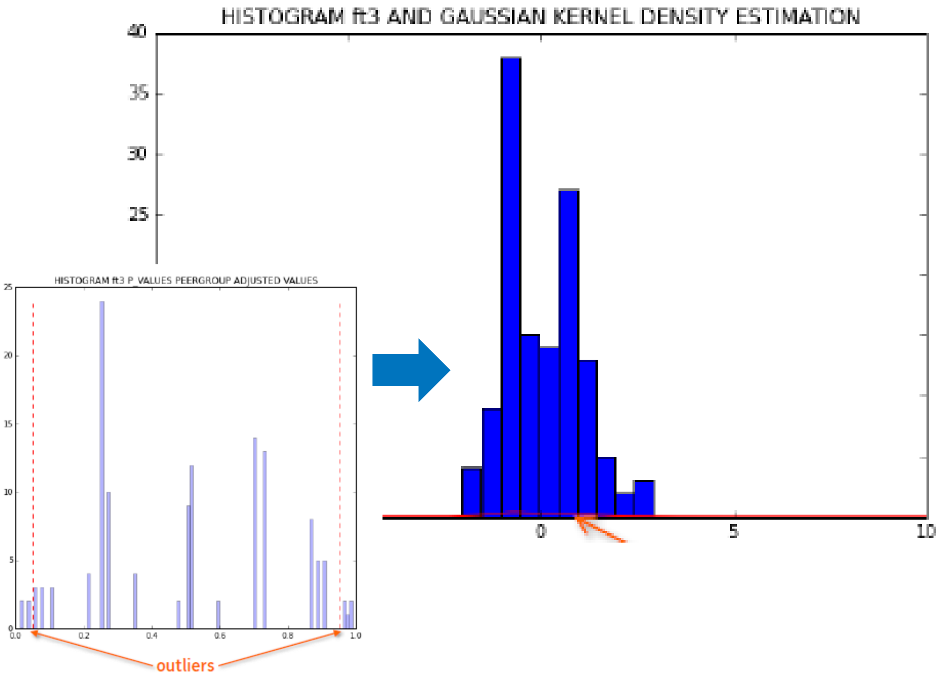
Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	199253	1561.9587	3123.917
Saturated	199671	0.0000	Prob>ChiSq
Fitted	418	1561.9587	1.0000

Unsupervised Machine Learning

Cluster Analysis

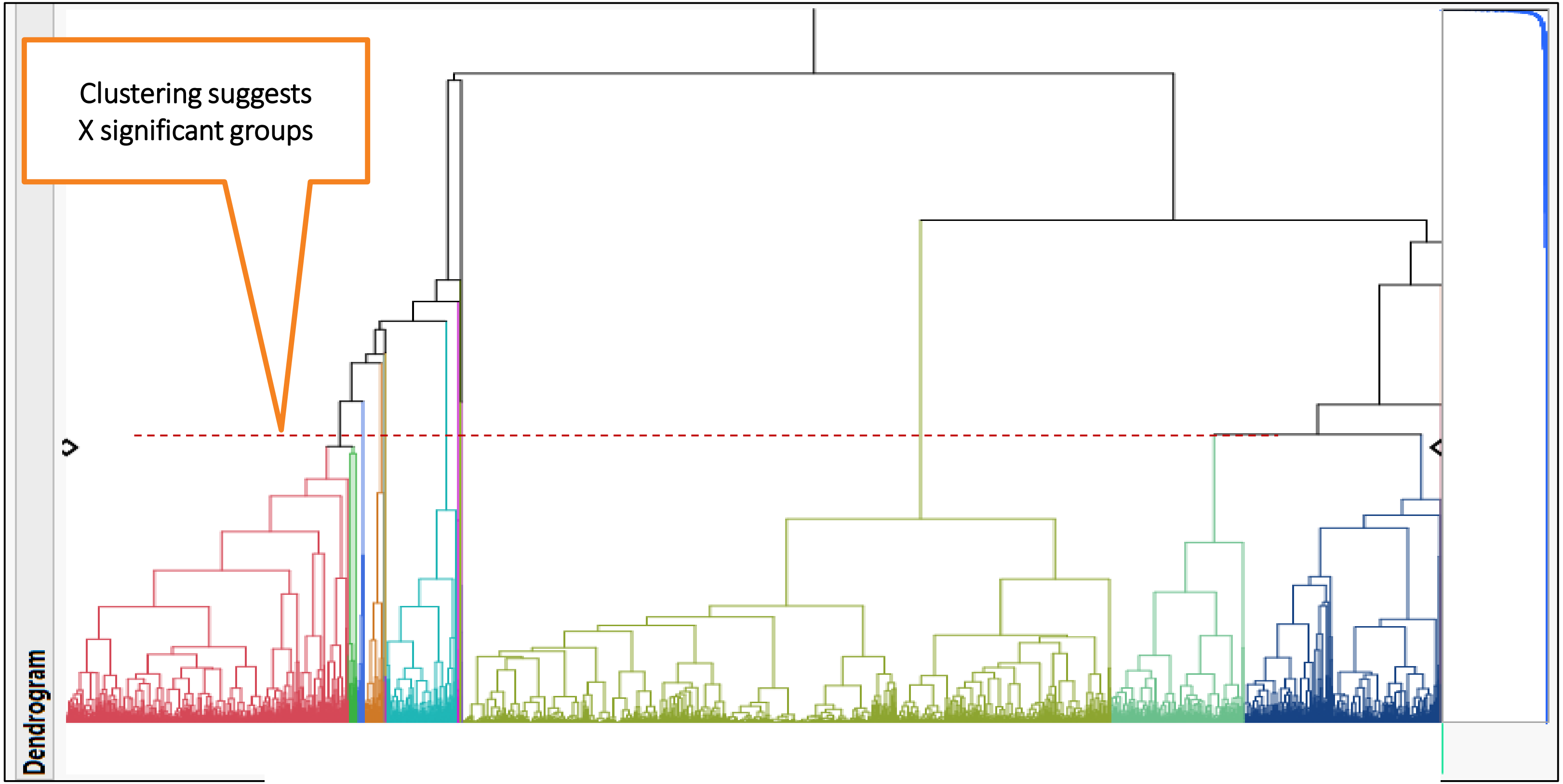
Derived (computer-generated clusters)





Cluster Analysis

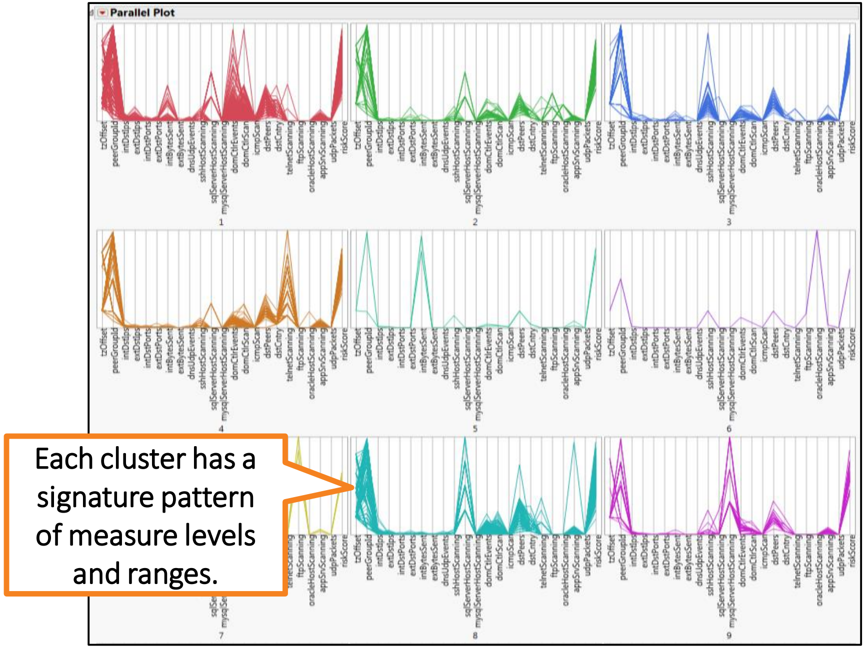
Extracting Statistically Self-Similar Groups

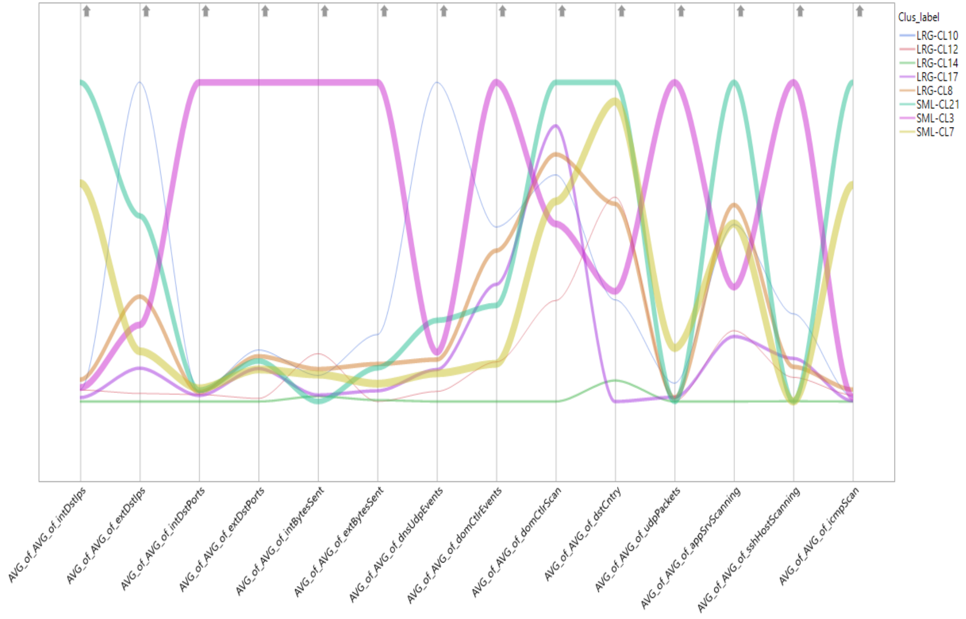


Clustering 'Peer Group' Labeling

Describing Behavioral Patterns

- Cluster analysis leads to insights into the nature of the patterns in each identified group.
- This will suggest descriptive labels, and should include a focused validation with SMEs.

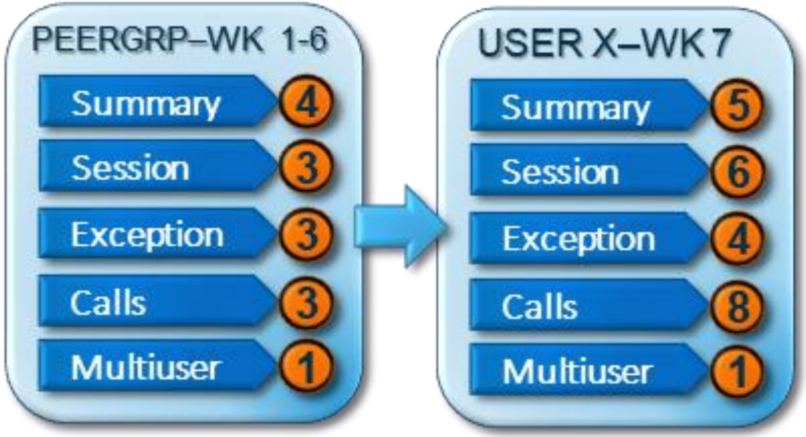




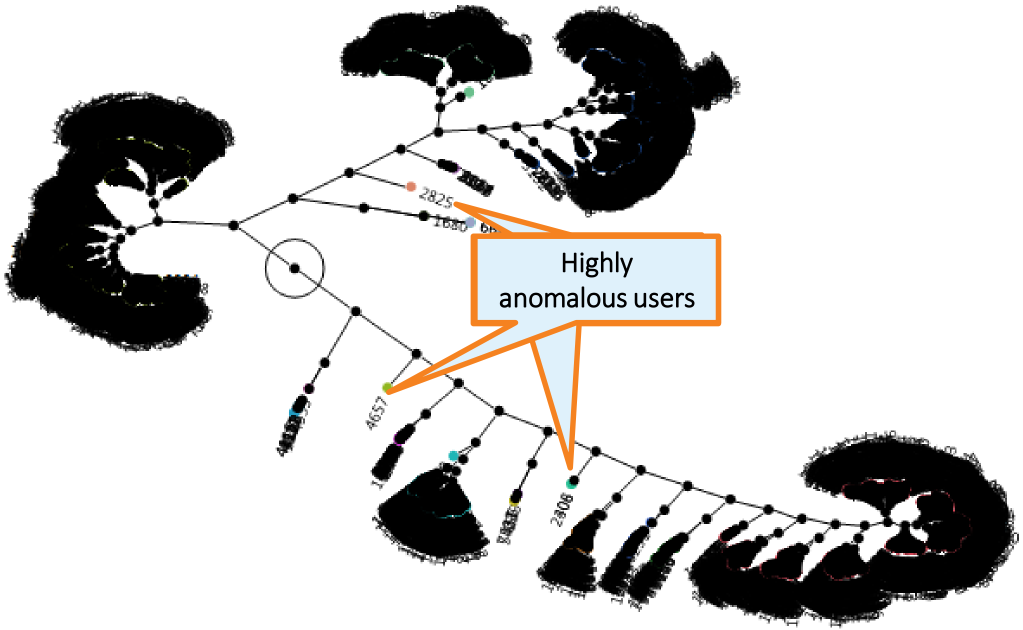
Uses of Cluster Analysis

Statistical Baselining for 'Normal' versus 'Abnormal'

USER DEVIATION FROM PEERGROUP

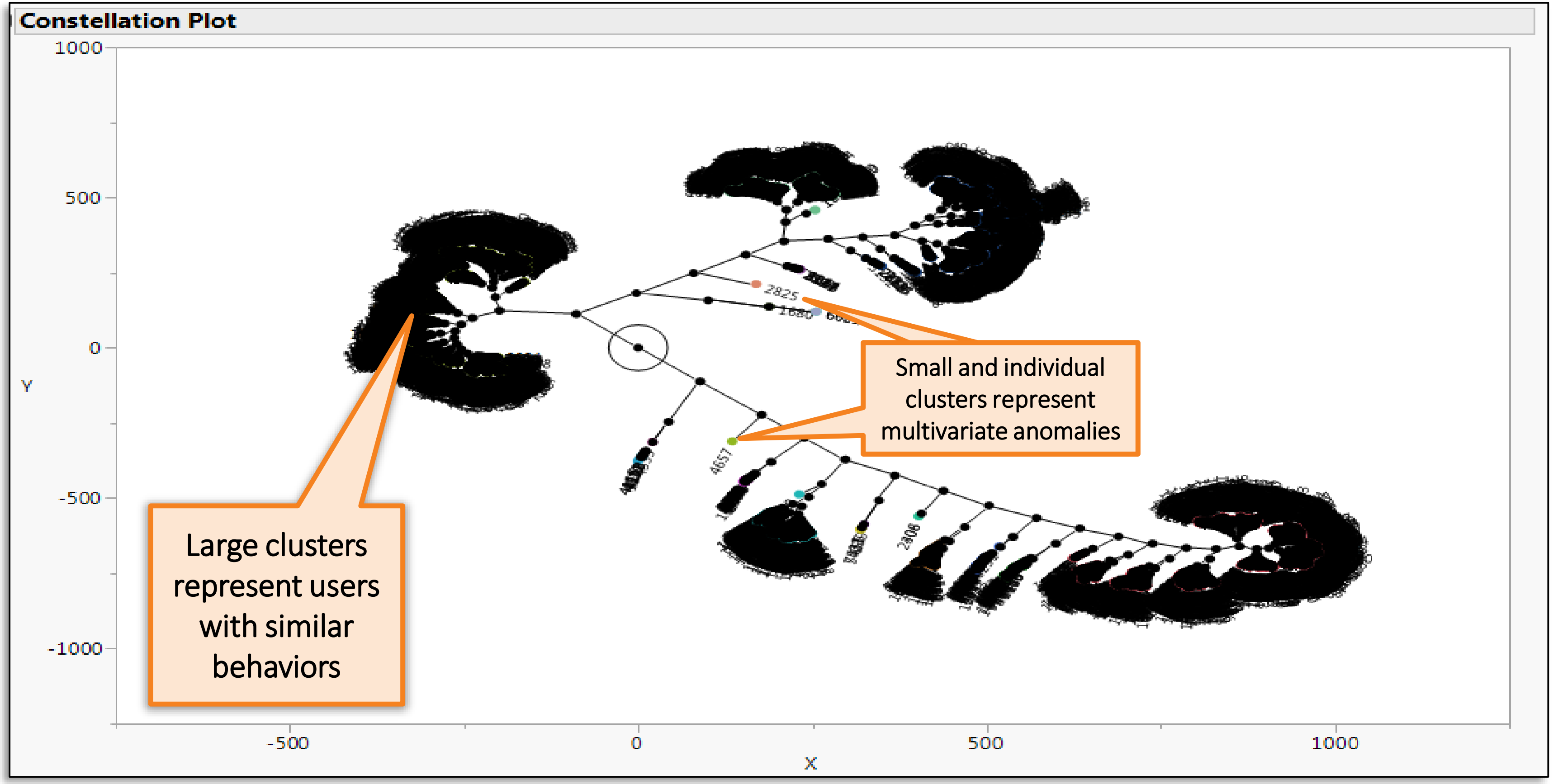


USER MULTIVARIATE ANOMALIES



Cluster-Based Outlier Detection

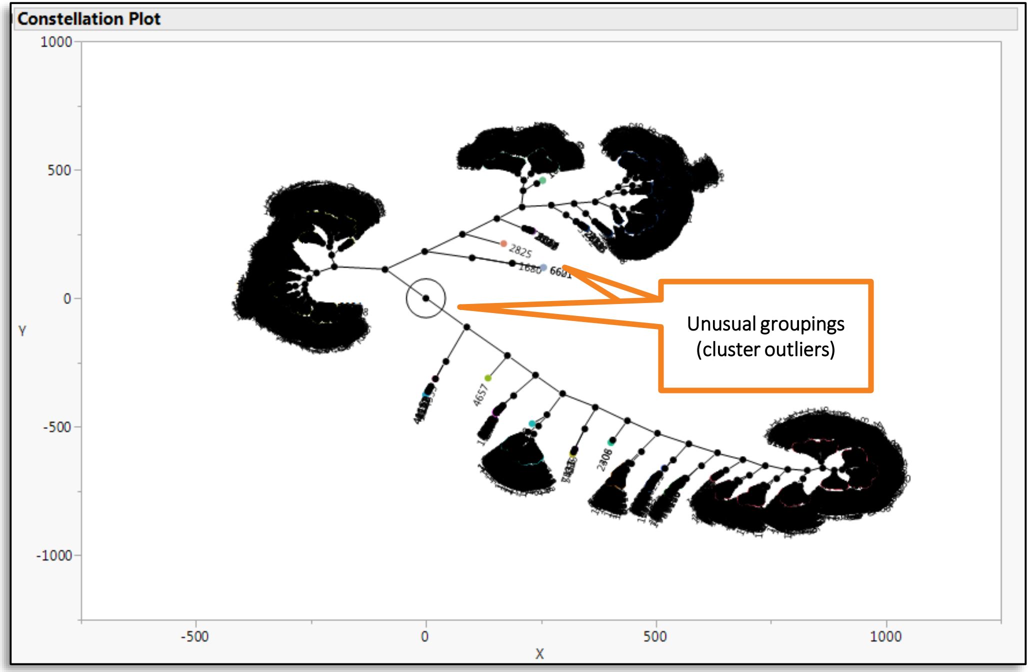
Identifying Anomlies in Environments of Uncertainty



Cluster-Based Outlier Detection

Cluster Analysis Surfaces Initial Outliers

- Analyzing the results of clustering provides insights to identify and flag behavioral outliers.
- The goal is to identify whether outliers should be put in special behavioral peer groups based on an approved exception, or whether a user has violated a policy or there is a security event occurring.
- After outliers are segmented, clustering can be re-run.



Column Summary

Column	RSquare
intBytesSent	0.6089
extBytesSent	0.8203
dnsUdpEvents	0.8561
sshHostScanning	0.5107
sqlServerHostScanning	0.7982
mysqlServerHostScanning	0.7863
domCtlrEvents	0.5181
domCtlrScan	0.4046
icmpScan	0.8791
dstPeers	0.4571
dstCntry	0.6208
telnetScanning	0.6000
ftpScanning	0.8734
oracleHostScanning	0.7479
appSrvScanning	0.4342
udpPackets	0.6871
riskScore	0.4062

Portion of total variation in each column absorbed by clustering

Cluster-Based Outlier Detection

First-Stage Cluster Analysis Surfaces Outliers

- First-stage cluster analysis quickly identifies outliers as small, unusual clusters.
- It is recommended that these be flagged and discussed with SMEs to determine whether they are special cases that deserve their own peer group, explainable temporary anomalies, or potential security events.

Hierarchical Clustering

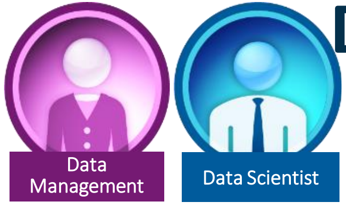
Cluster Summary

Cluster Means

Cluster	Count	numdevices	numhits	intDstIps	extDstIps	intDstPorts	extDstPorts	intBytesSent	extBytesSent	dnsUdpEvents	sshHostScanning
1	4672	1.295	116.687	4.285	2.992	1.259	1.007	19.107	8.396	1.983	3.74e-18
2	976	1.619	150.661	11.654	3.699	1.688	1.099	55.970	22.378	1.418	3.36e-18
3	107	1.366	124.364	34.645	8.804	3.486	1.579	44.391	23.709	5.150	-2.2e-19
4	832	1.445	104.457	5.893	10.069	1.123	1.132	35.027	70.841	2.490	3.25e-18
5	252	1.292	109.758	4.552	5.127	1.175	1.155	92.828	78.440	36.583	2.11e-18
6	1	1.125	18.000	4.000	2.000	1.000	1.000	161771.8	1429.877	2.000	5.42e-20
7	10	2.242	175.400	8.000	3.400	1.200	1.000	3889439	190.894	3.300	5.42e-20
8	58	1.486	137.034	3.879	2.293	1.345	0.948	76.185	24.444	1.810	-3.8e-19
9	131	1.525	72.870	6.870	106.527	1.511	1.962	35.184	70.266	4.924	4.88e-19
10	74	1.508	45.635	4.392	6.608	1.851	0.973	443.958	197.762	1.986	-3.8e-19
11	13318	1.319	120.575	2.945	2.415	1.211	0.942	25.277	4.856	1.126	-1.3e-16
12	10449	1.320	103.579	3.101	2.864	1.017	1.096	78.561	20.472	1.014	-1e-16
13	1768	1.309	97.141	2.801	3.197	1.059	0.700	20.377	18.017	9.584	3.52e-18
14	2987	2.687	252.104	2.851	1.727	1.174	0.796	17.848	2.534	0.857	3.63e-18
15	5	1.261	136.200	9.200	5.200	1.000	1.000	3178260	2556227	1.600	5.42e-20
16	107	2.450	267.000	4.084	0.607	53.897	0.542	7.010	0.143	1.336	-2.2e-19
17	6	1.336	108.667	3.167	1.667	1.333	0.833	4.639	2.636	-4.4e-16	0.00
18	25	1.390	133.160	6.480	7.400	1.280	1.280	378.347	107.726	2.080	1.00
19	1658	1.478	99.098	-4.1e-14	0.992	0.354	0.612	2.112	2.917	4.37e-14	3.52e-18
20	27885	1.233	51.077	1.122	0.000646	0.984	0.000143	2.683	0.015	1.343	1.76e-17

Exercise 1: Enterprise Guide

Cluster analysis and diagnostics

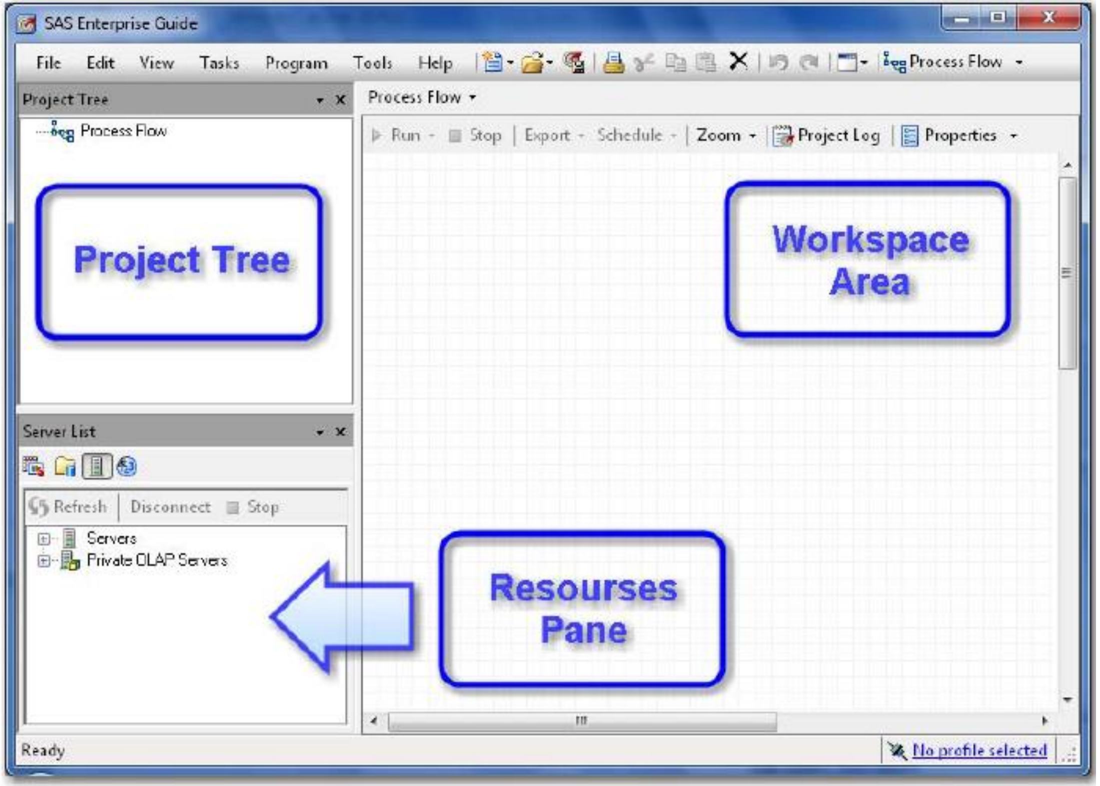


Data Analysis, Transformation, Analytics

SAS Enterprise Guide

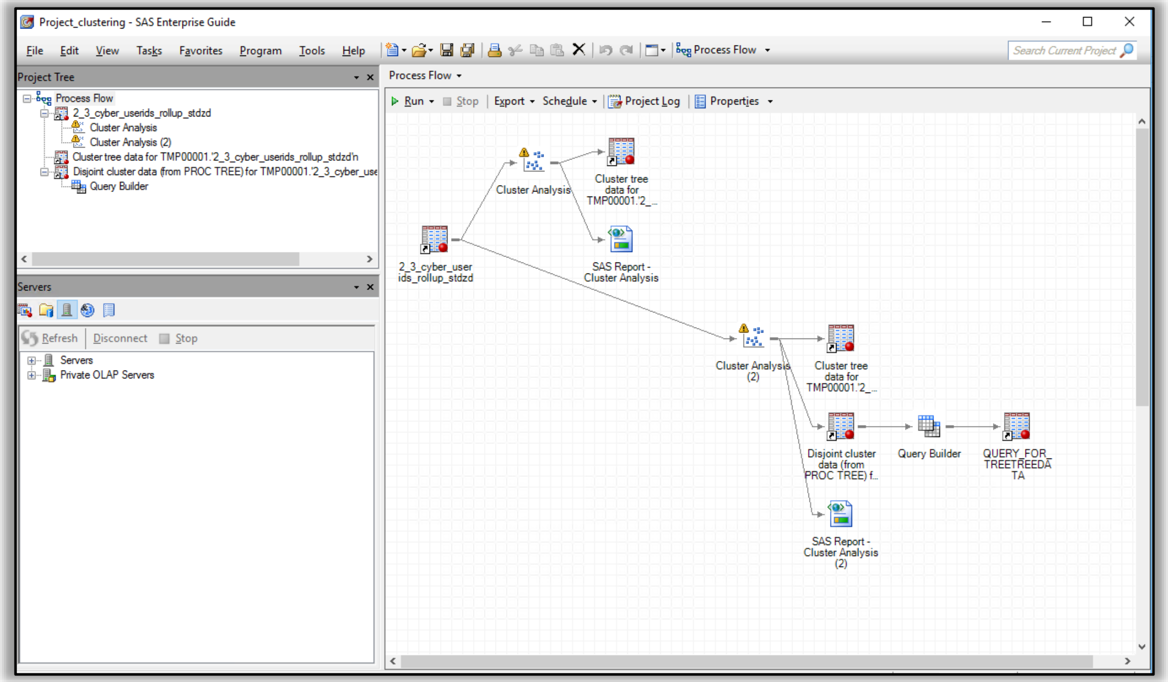
SAS Enterprise Guide – user-friendly interface to SAS Analytics:

- Preliminary data analysis
- Converting data into analytics-ready variables
- Creating workflows that structure and automate a complex set of procedures
- Performing statistical analysis, analytics, and machine learning
- Integrating SAS code



Example: Cluster Analysis

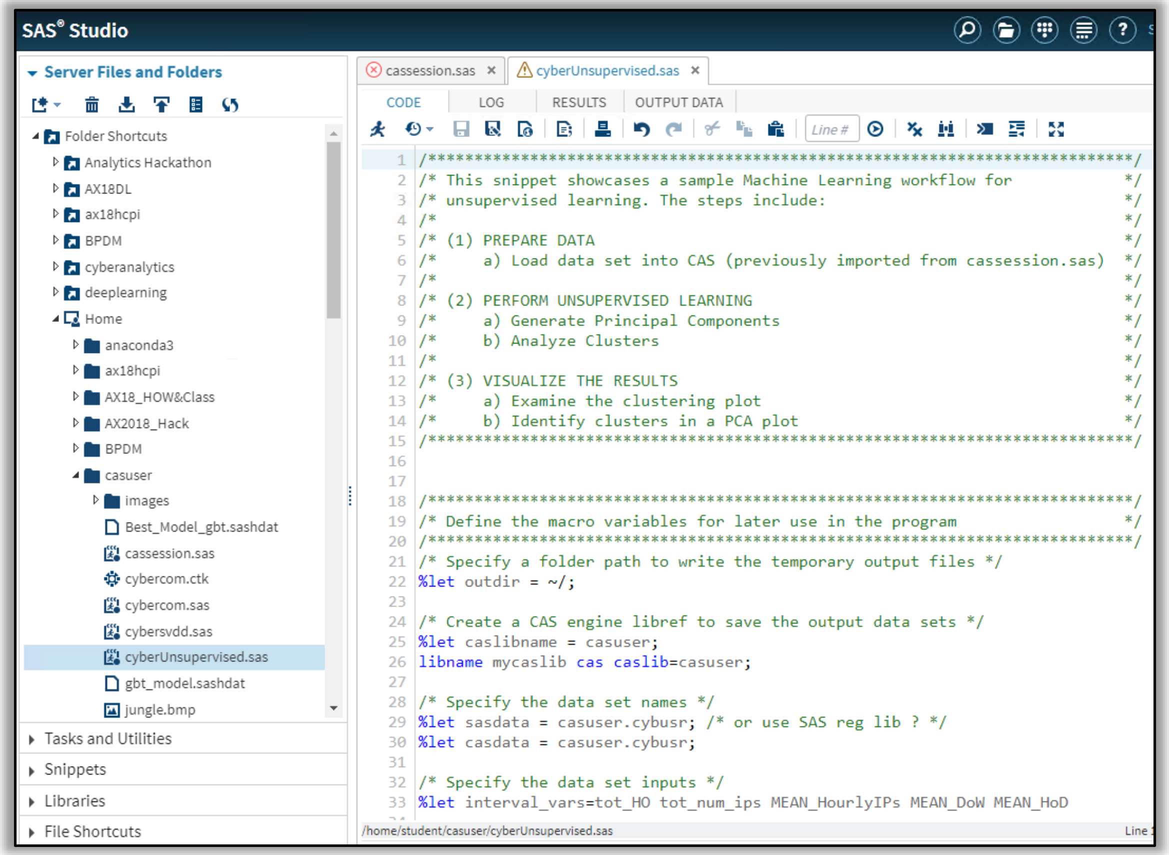
- 2_3_cyber_userids_rollup_stdsd (produced earlier, examined in JMP)
- 14,850 userids
- March 23rd – April 16th 2018



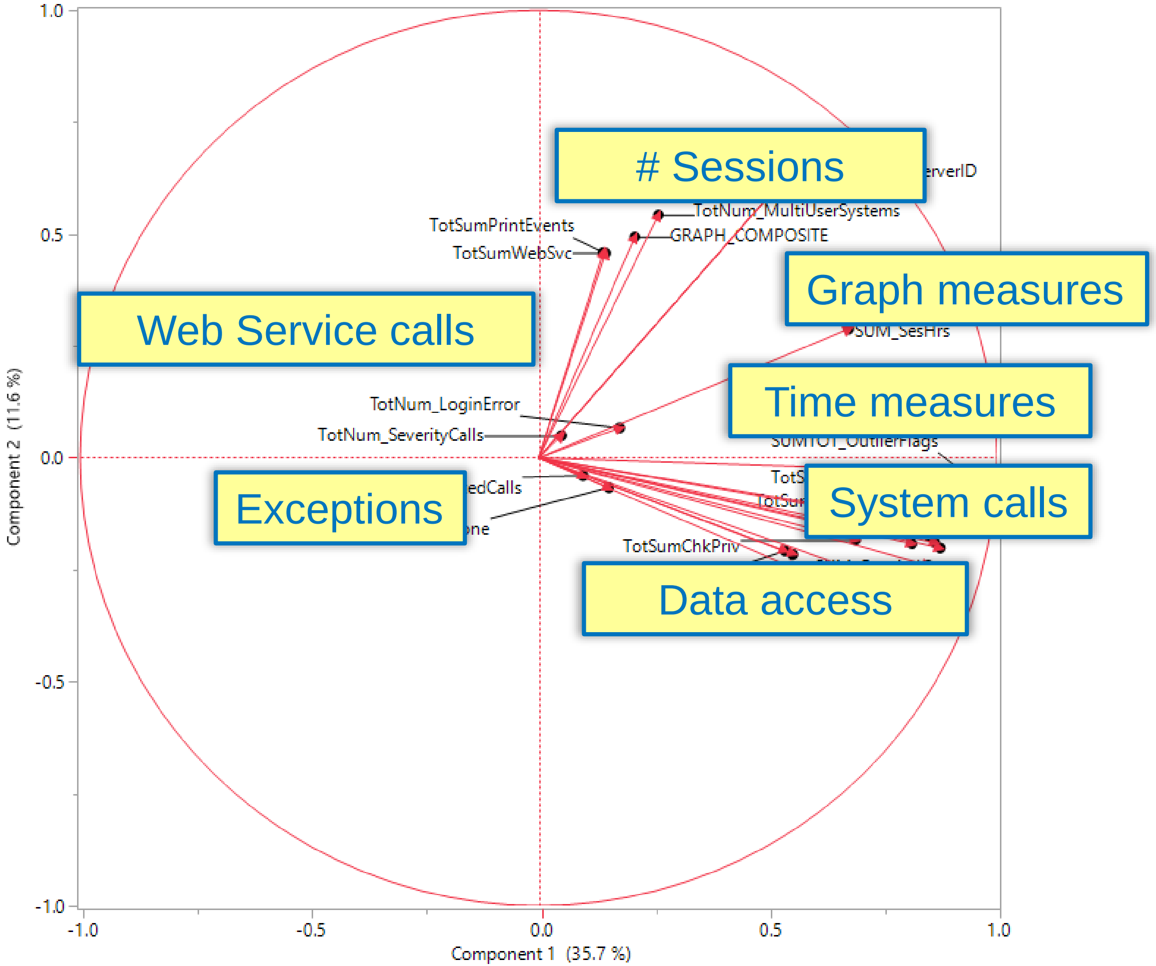
Exercise 2: Enterprise Guide

Cluster analysis + PCA

VDMML: Cluster Analysis + PCA visualization



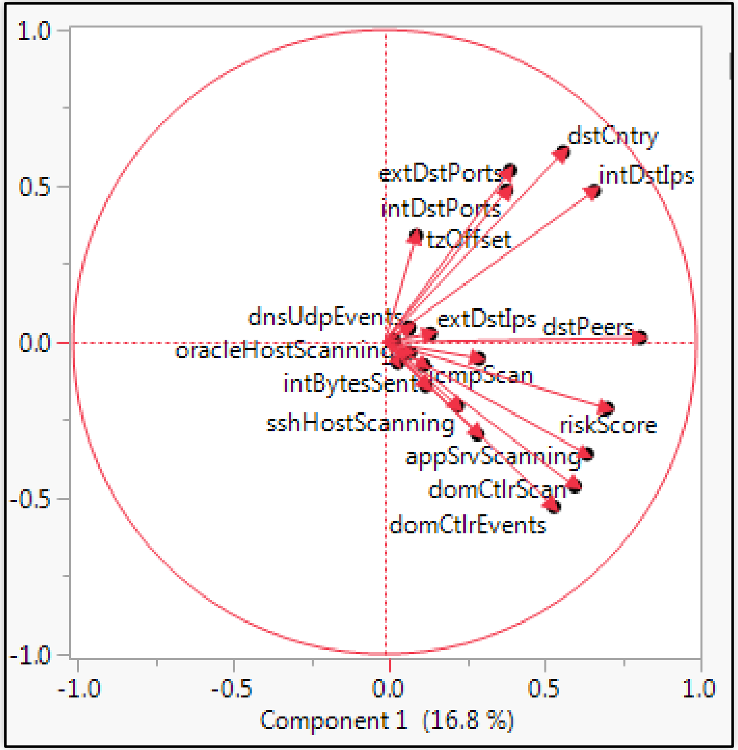
Review: Principal Component Analysis (PCA)

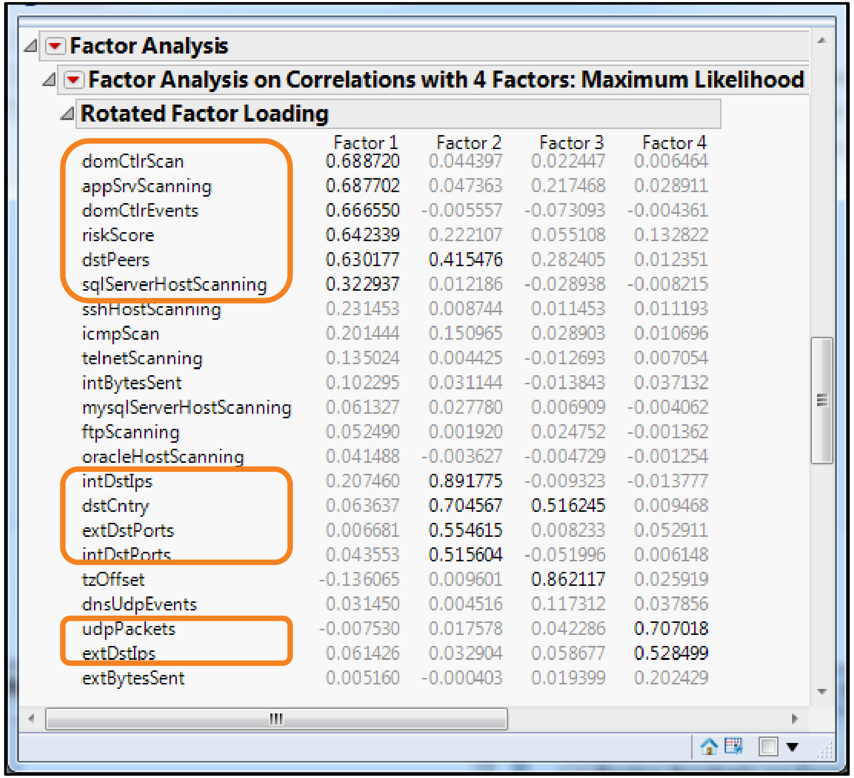


Review: Principal Component Analysis (PCA)

Seeking Connections Amongst Variables

- Examining relationships between variables -> Factors separate self-similar variables





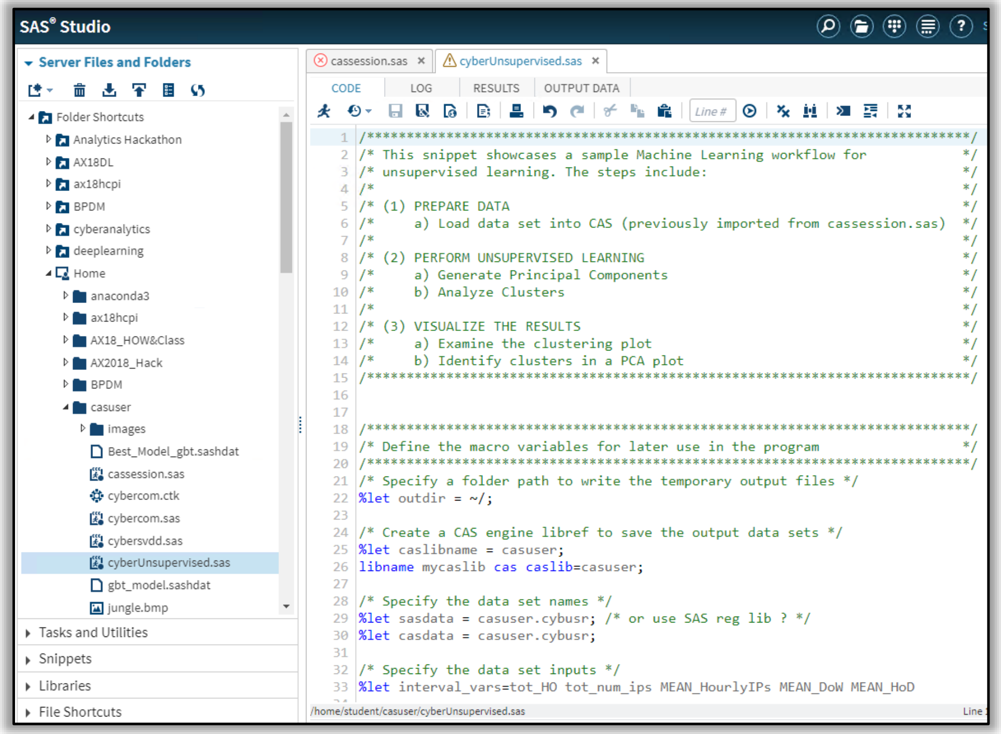
Factor Analysis

Factor Analysis on Correlations with 4 Factors: Maximum Likelihood

Rotated Factor Loading

	Factor 1	Factor 2	Factor 3	Factor 4
domCtlrScan	0.688720	0.044397	0.022447	0.006464
appSrvScanning	0.687702	0.047363	0.217468	0.028911
domCtlrEvents	0.666550	-0.005557	-0.073093	-0.004361
riskScore	0.642339	0.222107	0.055108	0.132822
dstPeers	0.630177	0.415476	0.282405	0.012351
sqlServerHostScanning	0.322937	0.012186	-0.028938	-0.008215
sshHostScanning	0.231453	0.008744	0.011453	0.011193
icmpScan	0.201444	0.150965	0.028903	0.010696
telnetScanning	0.135024	0.004425	-0.012693	0.007054
intBytesSent	0.102295	0.031144	-0.013843	0.037132
mysqlServerHostScanning	0.061327	0.027780	0.006909	-0.004062
ftpScanning	0.052490	0.001920	0.024752	-0.001362
oracleHostScanning	0.041488	-0.003627	-0.004729	-0.001254
intDstIps	0.207460	0.891775	-0.009323	-0.013777
dstCntry	0.063637	0.704567	0.516245	0.009468
extDstPorts	0.006681	0.554615	0.008233	0.052911
intDstPorts	0.043553	0.515604	-0.051996	0.006148
tzOffset	-0.136065	0.009601	0.862117	0.025919
dnsUdpEvents	0.031450	0.004516	0.117312	0.037856
udpPackets	-0.007530	0.017578	0.042286	0.707018
extDstIps	0.061426	0.032904	0.058677	0.528499
extBytesSent	0.005160	-0.000403	0.019399	0.202429

VDMML: Cluster Analysis + PCA visualization



- Visualizing the relationship between variables in multivariate statistics can be challenging
- It requires viewing the data in hyper-dimensions
  One option for visualizing the relationship between all variables is to examine principal components
- Principal component analysis (PCA) will create uncorrelated linear combinations of the variables
- First two principal components can be thought of as the two dimensions among variables that are the most un-related
- Plotting the data against the first two principal components will give the most un-correlated view of the data, thereby allowing the separation between observations to be best seen in two dimensions

For example
http://support.sas.com/resources/papers/proceedings13/447-2013.pdf

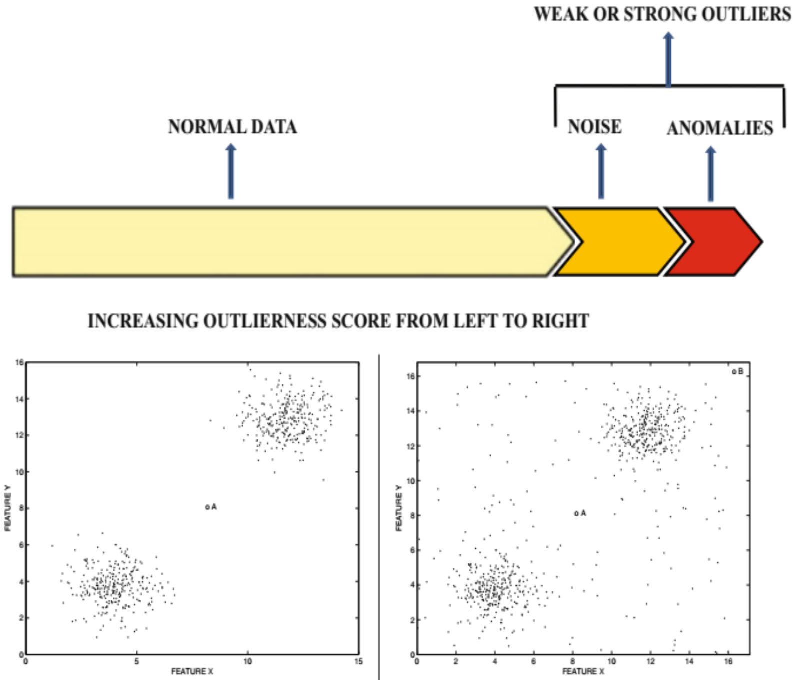
Anomaly Detection

Support Vector Data Description (SVDD)

Focused on outlier detection, SVDD is a machine learning technique where the model builds a minimum radius sphere around multidimensional training data and scores new observations by comparing to distance from sphere center from sphere radius

Simply Complex

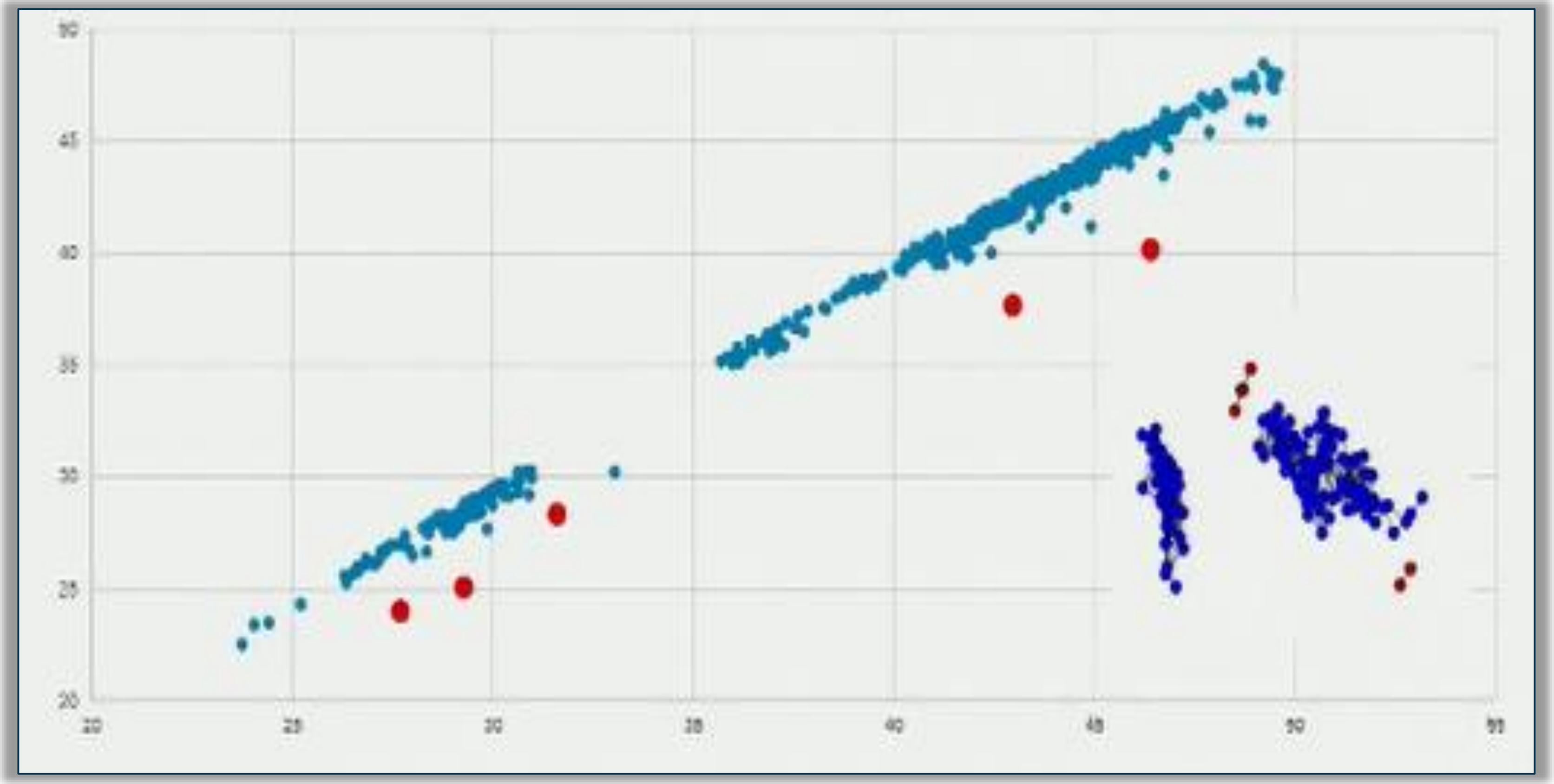
Identifying targeted anomalies amongst and ocean of noise…



(a) No noise (b) With noise

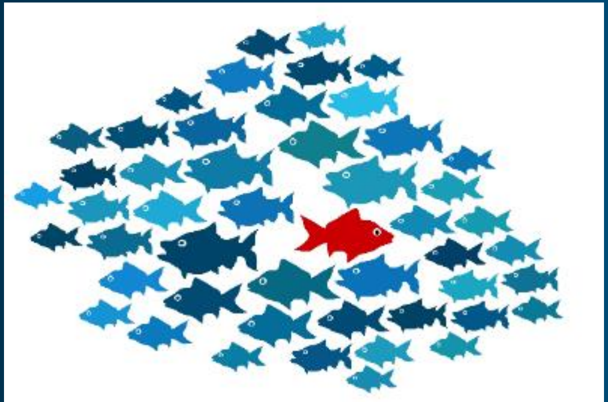
SOURCE
Aggarwal, Charu C. (2017). "Outlier Analysis: Second Edition". Springer International Publishing AG.

ANOMALY DETECTION



Example Methods for Anomaly Detection

Surfacing Rare Events



Support Vector Data Description

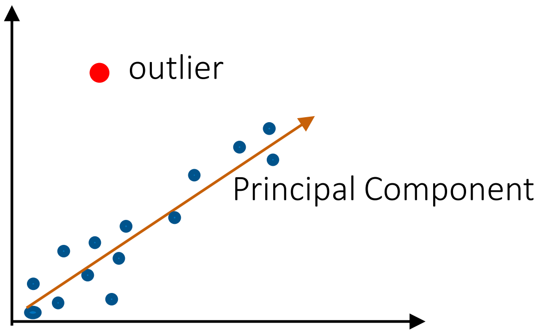
Robust PCA

Auto Encoders

Principal Component Analysis (PCA) Anomaly Detection

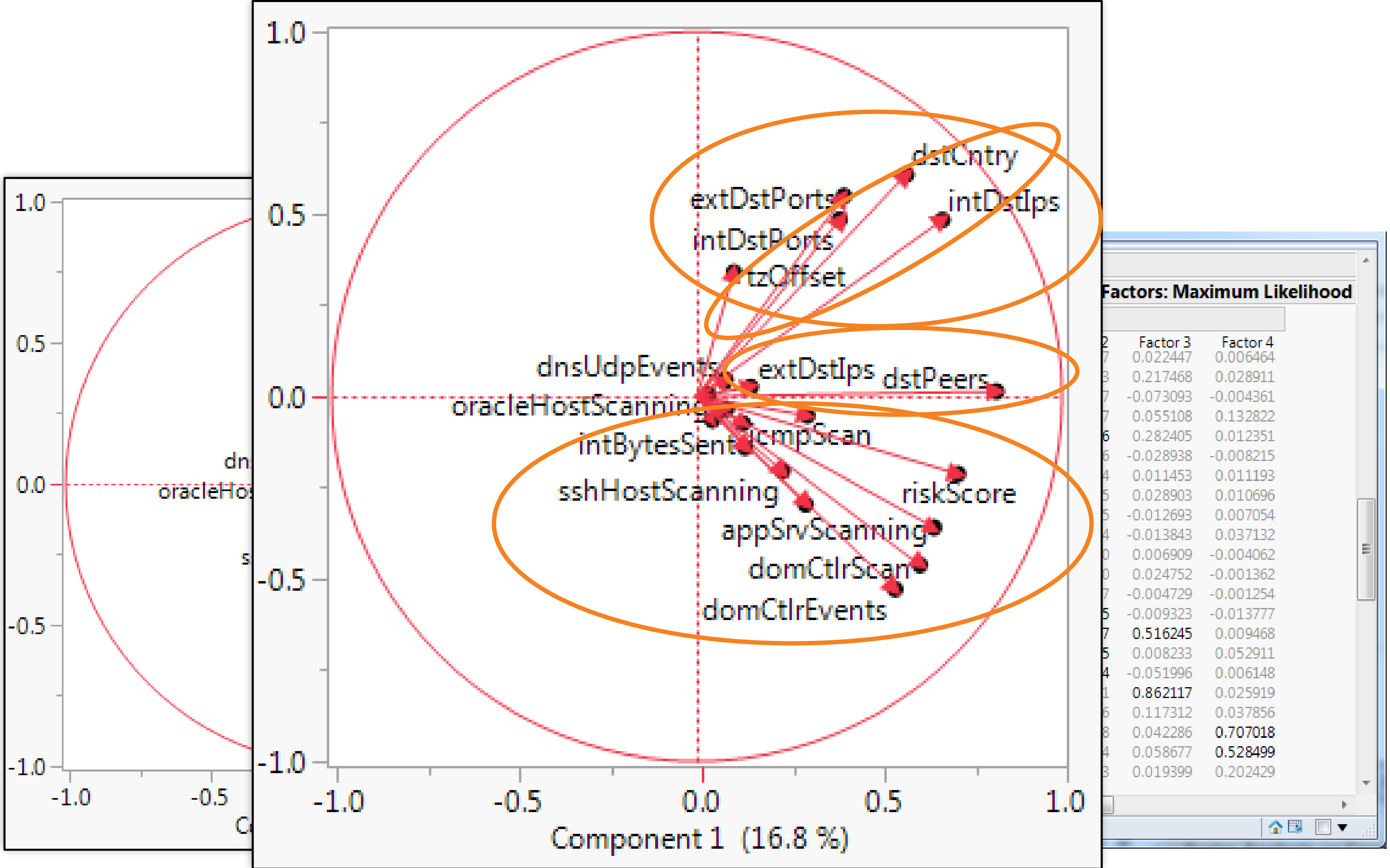
Analyze the input features like ”Internal Bytes, External Bytes etc. (27 comparison measurements) within the peer group, and look for relationship among those features, and determine *linear* combination of values that best capture the difference.

Source IP Address	Source IP Peer ID	PCA Anomalous Flag Count	PCA Score Sum
10.0.0.1	Switch	24	24.23402
10.20.19.36	Computer(PC)	24	23.79015
140.33.21.38	PLC	24	23.78423



https://www.jmp.com/support/help/14/principal-components.shtml

https://en.wikipedia.org/wiki/Principal_component_analysis



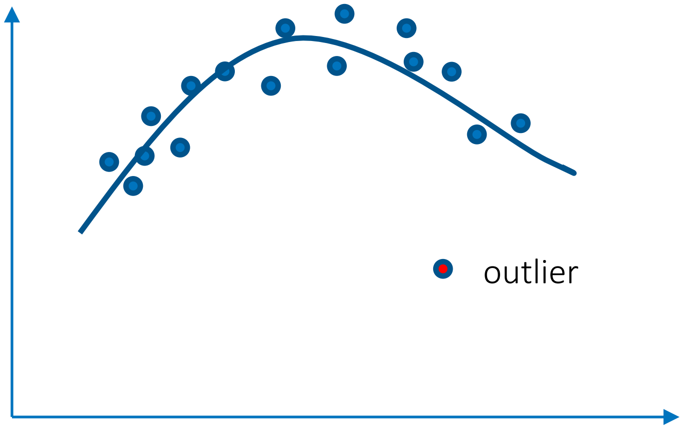
1.0
0.5
0.0
-0.5
-1.0
dstCntry
extDstPorts
intDstIps
intDstPorts
tzOffset
dnsUdpEvents
extDstIps
dstPeers
oracleHostScanning
intBytesSent
icmpScan
sshHostScanning
riskScore
appSrvScanning
domCtlrScan
domCtlrEvents
Component 1 (16.8 %)
Factors: Maximum Likelihood
Factor 3 Factor 4
0.022447 0.006464
0.217468 0.028911
-0.073093 -0.004361
0.055108 0.132822
0.282405 0.012351
-0.028938 -0.008215
0.011453 0.011193
0.028903 0.010696
-0.012693 0.007054
-0.013843 0.037132
0.006909 -0.004062
0.024752 -0.001362
-0.004729 -0.001254
-0.009323 -0.013777
0.516245 0.009468
0.008233 0.052911
-0.051996 0.006148
0.862117 0.025919
0.117312 0.037856
0.042286 0.707018
0.058677 0.528499
0.019399 0.202429

Autoencoder Anomaly Detection

Unsupervised ML: neural network-based (‘deep belief-learning’)

Extension of PCA except it accommodates *nonlinear* datasets

Source IP Address	Source IP Peer ID	Autoencoder Anomalous Flag Count	Autoencoder Score Sum
190.50.141.14	Switch	20	22.09677748
124.19.21.26	PLC	20	22.41426623



[CERN – outlier detection through autoencoders]

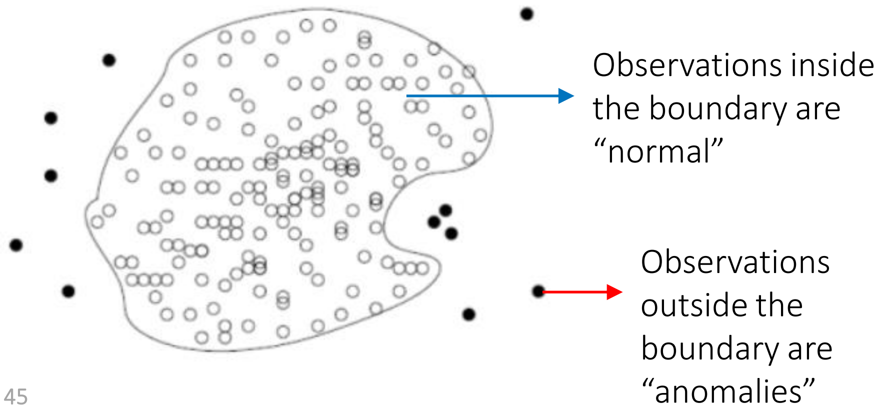
Fraud domain example:
https://shiring.github.io/machine_learning/2017/05/01/fraud

Support Vector Data Description (SVVD) Anomaly Detection

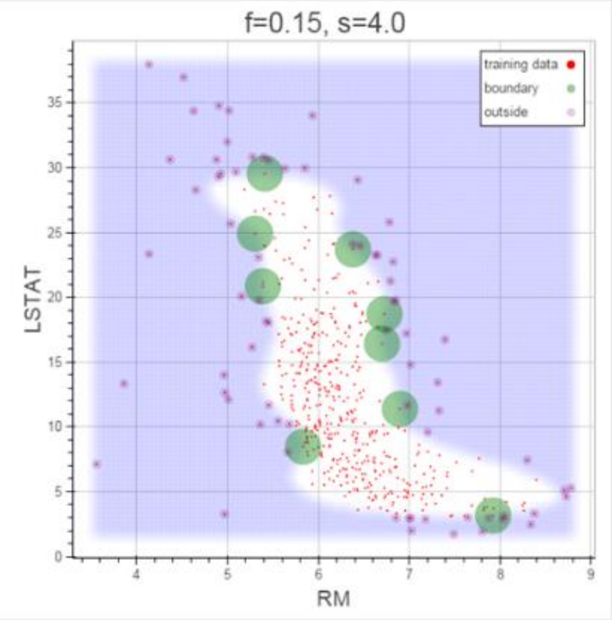
- SVDD is a variant of the support vector machine : supervised machine learning (classification)
- Avoids overfitting
- Uses peer group as a labels
- Centroid distance
- SVDD will identify a decision boundary that can distinguish “normal” data from anomalies

Source IP Address	Source IP Peer ID	SVDD Anomalous Flag Count	SVDD Distance Sum
120.19.42.13	PLC	24	28.65784311
190.31.21.50	Smarteye	24	44.47409719

SVDD academic paper



Support Vector Data Description



What is it?

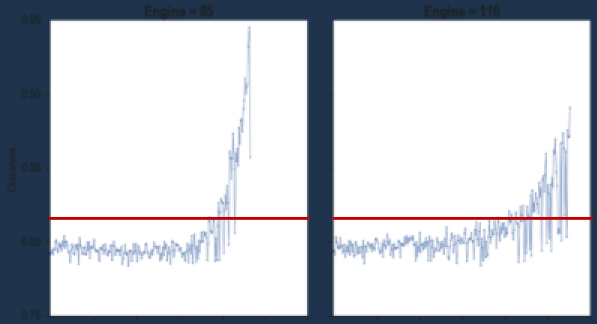
Single class classification technique

- Identifies minimum radius hypersphere around “normal” data
- Works on multivariate data
- Does not require assumption of normality
- Fits flexible surfaces using kernel function
- Minimizes the chance of accepting outliers

Use Case: Anomaly Detection

- Cyber-security intrusion detection
- Fraud detection
- Also, Identify process degradation (manufacturing, health care, capitally intensive assets)

Support Vector Data Description



How Does it Work?

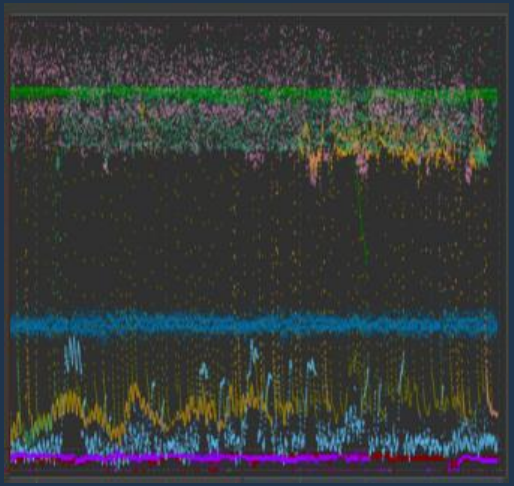
Unsupervised Machine Learning

- Creates minimum-radius hypersphere around the training dataset
  - Test multiple kernel function values
  - Identify first occurrence when second derivative of kernel output radius equals zero
  - Retune model with kernel function value associated with previous step
- Scores new observations by calculating distance to hypersphere center
  - Observations with distances greater than minimum-radius are flagged as anomalies

Support Vector Data Description

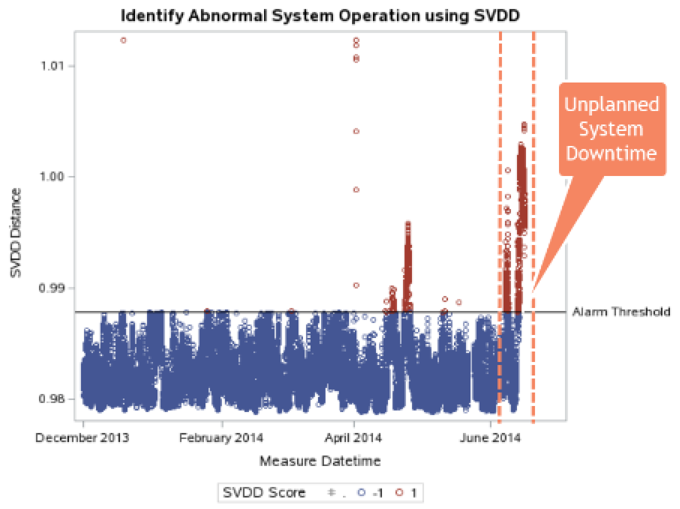
Example: Identify Abnormal System Operation

Multivariate Data



- ✓ Very simple approach
- ✓ No need to identify anomalous observations (single class classifier)
- ✓ Supports multivariate data
- ✓ Does not require assumption of normality
- ✓ Flexible data descriptions allowing multiple different regions of "normal" operating conditions

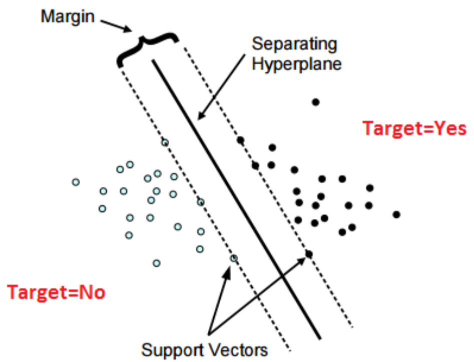
Support Vector Data Description



Support Vector Data Description

How is it different than Support Vector Machines?

Support Vector Machine

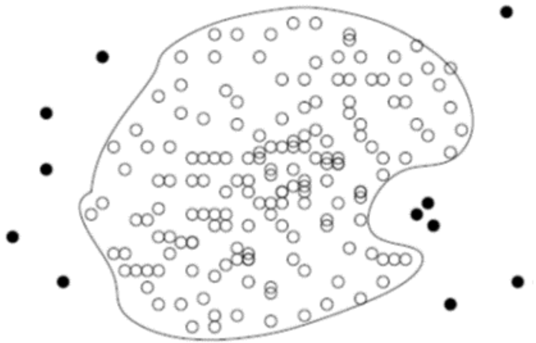


Two-class classifier

- Constructs hyperplane maximizing margin between two classes

Requires sufficient data representing both classifiers to obtain a good model

Support Vector Data Description



One-class classifier

- constructs a closed hypersphere around the "target" class where excluded observations are "anomalies"

Used in domains where the majority of the data belongs to one class

How to use SVDD

Programmatic Approach using

SAS Studio / Code Node

The SVDD Procedure

Model Information	
Optimization Method	Active set
Kernel Type	RBF
RBF Kernel Bandwidth	121.89038176
Bandwidth Relative Scale	1
Expected Outlier Fraction	0.0001
Optimization Tolerance	0.0001
Number of Interval Variables	25
Number of Nominal Variables	0

Number of Observations Read	1822
Number of Observations Used	1822

Optimization Summary	
Number of Iterations	1
Objective Value	0.1074216427
Infeasibility	8.39262E-12
Optimization Status	Optimal
Degenerate	False

Training Results	
Number of Support Vectors	24
Number of Support Vectors on Boundary	24
Number of Dropped Observations	0
Threshold R^2 Value	0.89258
Constant (C_r) Value	0.10742
Run Time (Seconds)	0.02347

PROC SVDD

Example:

```
proc svdd data=casuser.train outlier_fraction=0.0001 nthreads=4;
    input cycle X1-X24 / level=interval;
    kernel rbf / bw=mean;
    solver actset /;
    savestate rstore=casuser.svddmodel;
    id engine cycle;
run;
```

How to understand _SVDDSCORE_:

- _SVDDScore_ = 1 means anomaly
- _SVDDScore_ = -1 means normal

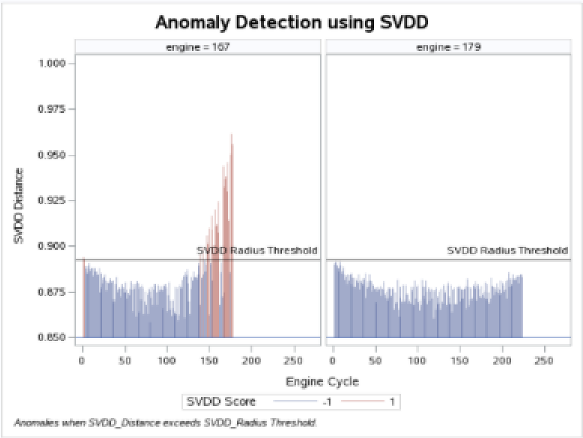
Best Practices:

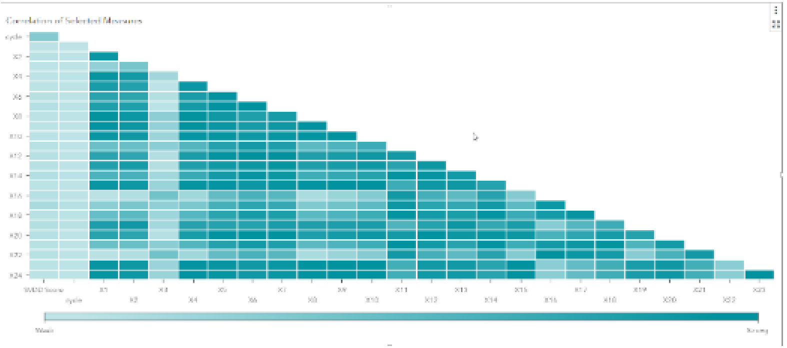
- Specify outlier fraction you believe is likely
- Use bw=mean to find optimal bandwidth value

  NOTE: bw=mean only works if all inputs are interval
- Use solver actset for small training datasets
- Use solver stochs for large training data sets
- Use id to specify non-input variables needed and available when scoring new observations
- Capture Threshold R^2 Value from PROC SVDD results for outlier/anomaly _SVDDISTANCE_ cutoff

PROC SVDD Example

```
/*******************************************************************************************
/* Run SVDD workflow */
/*******************************************************************************************
proc svdd data=casuser.train outlier_fraction=0.0001 nthreads=4;
    input cycle X1-X24 / level=interval;
    kernel rbf / bw=mean;
    solver actset /;
    savestate rstore=casuser.svddmodel;
    id engine cycle;
run;

/* Capture the Threshold R^2 Value from PROC SVDD results*/
%let threshold=0.89258;

/* Score SVDD on all data */
proc astore;
    score data=public.PHM08_MOD_SCORE
    out=casuser.svddscore
    rstore=casuser.svddmodel;
quit;

/* Plot SVDD Anomaly Detection results */
ods graphics / antialias=on antialiasmax=5200;
proc sgpanel data=casuser.svddscore;
panelby engine / spacing=5;
needle x=cycle y=_SVDDdistance_ / group=_SVDDScore_ baseline=0.85 transparency=0.5;
refline &threshold /label="SVDD Radius Threshold" lineattrs=(color=black) labelpos=max;
title H=14pt "Anomaly Detection using SVDD";
colaxis label="Engine Cycle";
rowaxis label="SVDD Distance";
footnote H=8pt j=l italic "Anomalies when SVDD_Distance exceeds SVDD_Radius Threshold.";
run;

/* distinct list of engines and cycles identified as 'anomalies' */
data casuser.svddanomalies;
    set casuser.svddscore;
    where _SVDDSCORE_ = 1;
    flag = "Anomaly";
run;

/* save state of svdd model */
proc astore;
    download rstore=casuser.svddmodel
    store="/home/dishaw/sasuser.viya/svddstate.sasast";
quit;
```



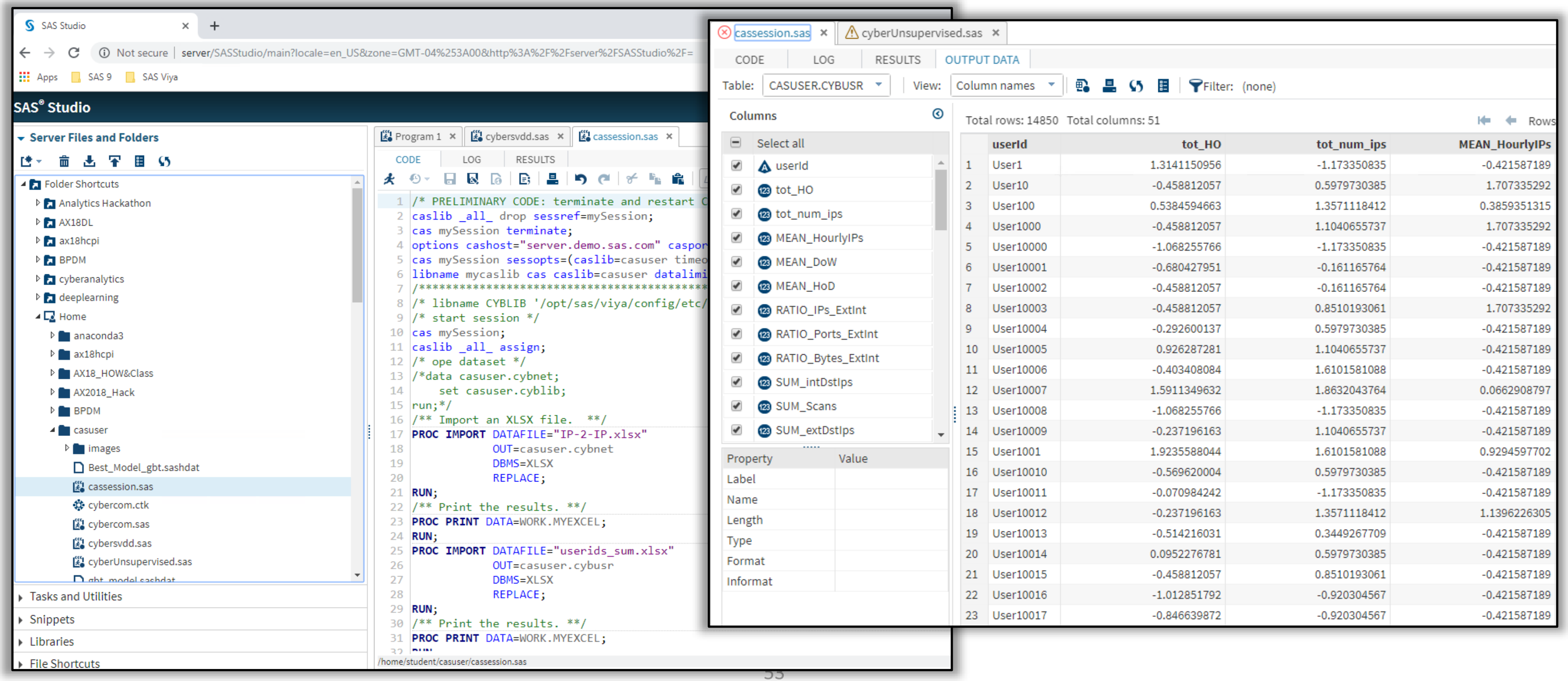


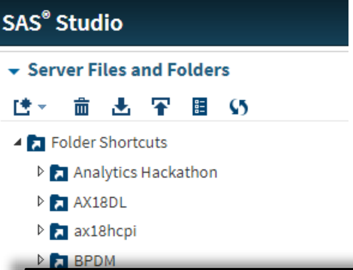
Exercise 3: Anomaly Detection (SVDD)

This practice reinforces the concepts discussed previously.

Training and Deploying a Real-Time SVDD Anomaly Detection Model

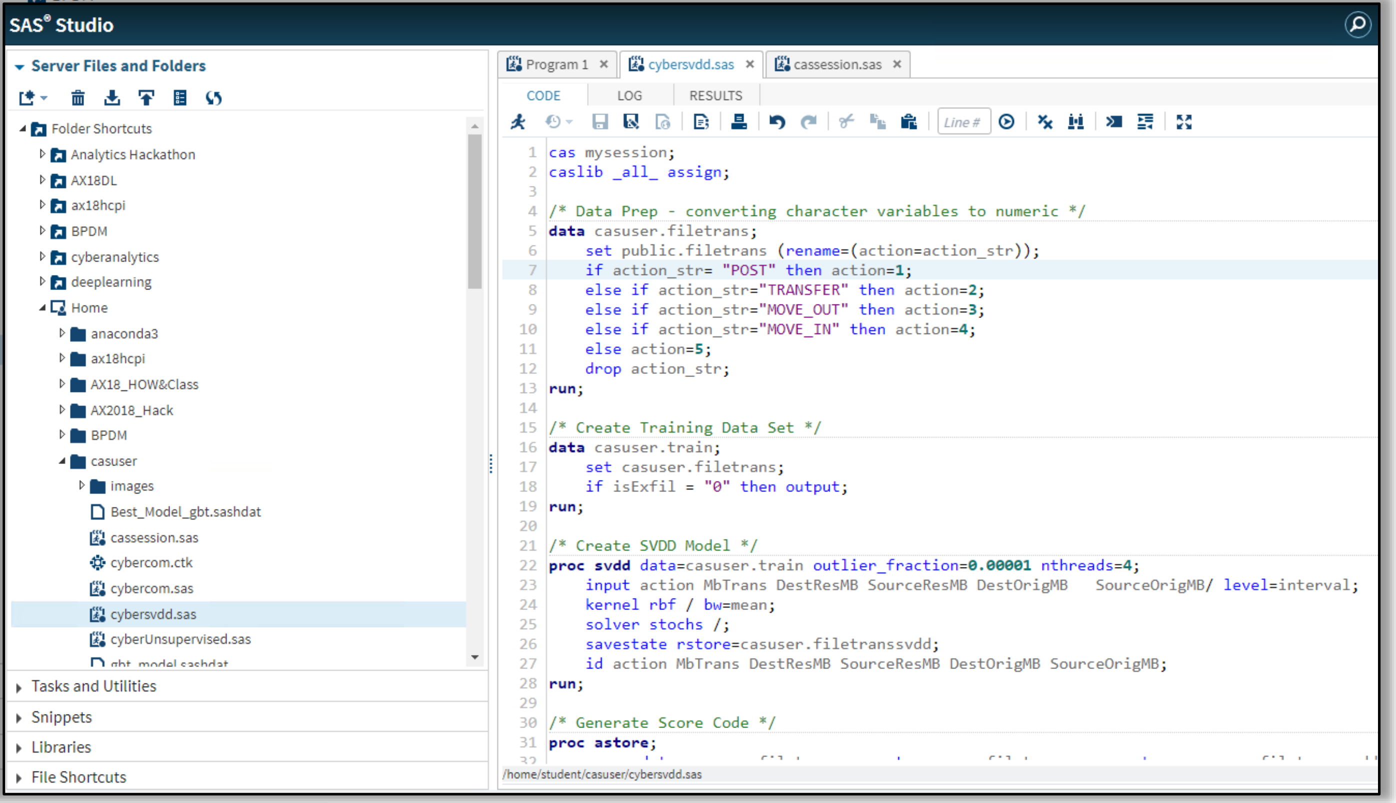
- Chrome => SAS Studio
- Folder Shortcuts => Home => casuser => cassession.sas
  - CASUSR.CYBUSR (same as earlier cyber_userids_rollup_stdzd.sas7bdat

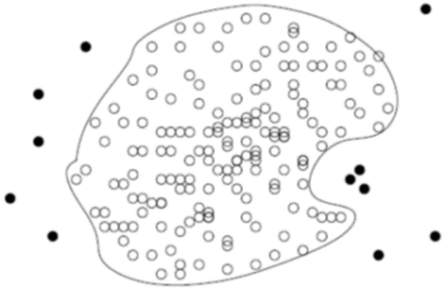




Training and Deploying a Real-Time SVDD Anomaly Detection Model

- Chrome => SAS Studio
- Folder Shortcuts => Home => casuser => cybersvdd.sas



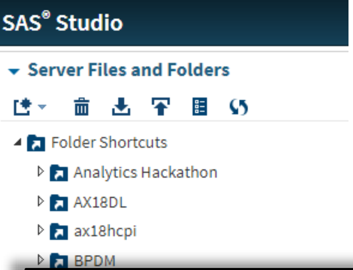


Scenario

- Highly secure document management repository
- Analysts move documents between secure folders for projects
- 1,048,576 records of file access / transfers (originally from Kagle competition)
  - Small minority of flagged exfiltration / improper access incidents (not used)
  - Majority are ‘cleared’ (not exfil) file access / transfers
- Desire to establish focused anomaly detection to warn of incidents
- Anomaly detection with SVDD Model in SAS Studio
  - Only ‘cleared’ transactions not used to train
  - SVDD only takes numeric data
  - Finds unusual cases - rejected from ‘sphere’

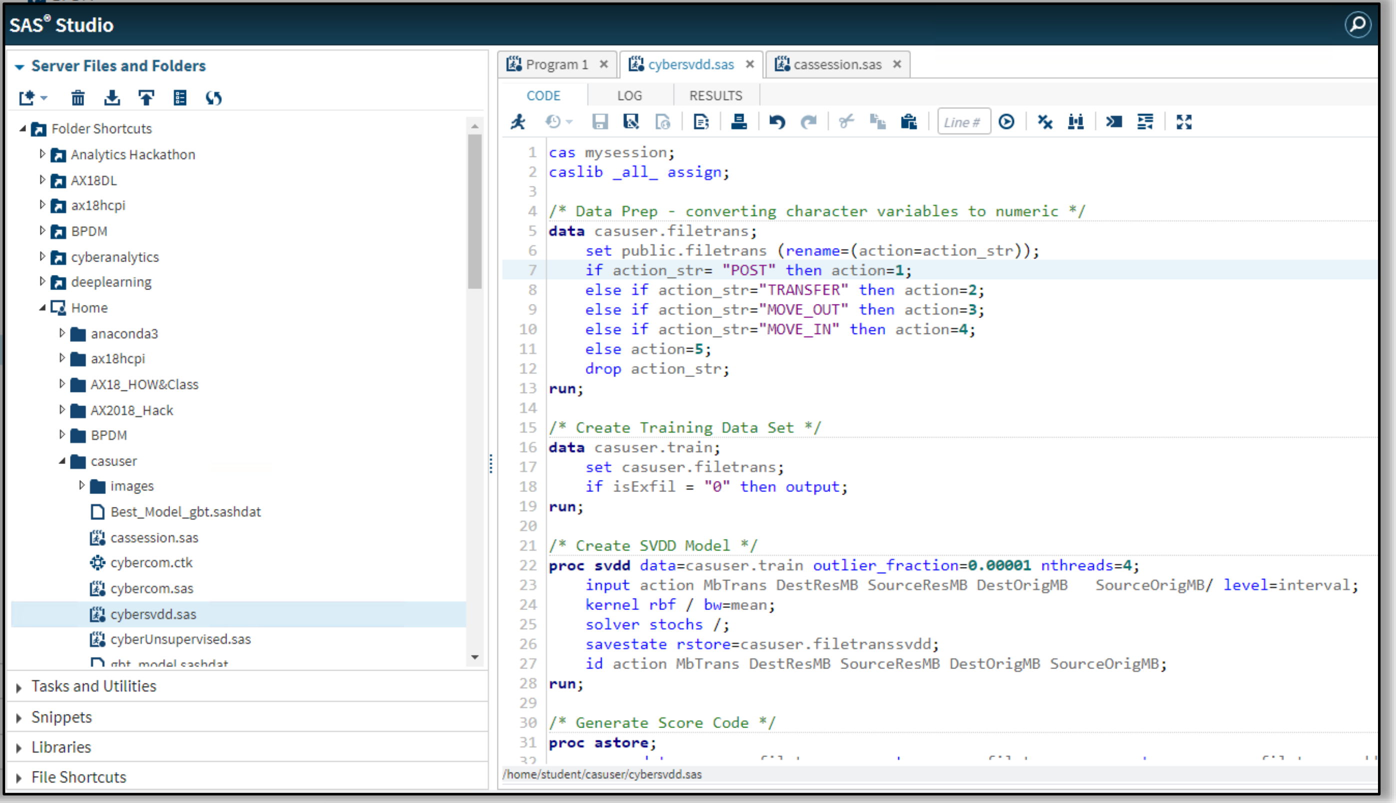
Data Available: Secure Document System

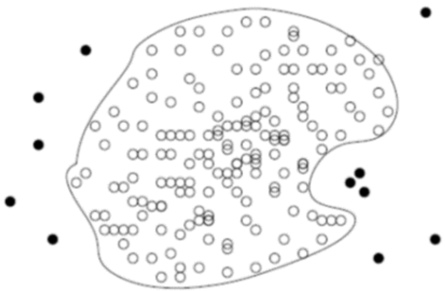
VARIABLE	DESCRIPTION
Action	Files are either posted, transferred, moved_out, extracted (utilized), or moved_in
MbTrans	Mbs transfered in action
FolderSource	Source folder name
SourceOrigMB	Number of MB in source folder before action
SourceResMB	Number of MB in source folder after action
FolderDest	Destination folder name
DestOrigMB	Number of MB in destination folder before action
DestResMB	Number of MB in destination folder after action
IsExfil	Marked as detected exfiltration / improper axtion event
IsExfilFlag	Flagged as exfil / improper access
UserId	UserId of agent performing action



Training and Deploying a Real-Time SVDD Anomaly Detection Model

- Chrome => SAS Studio
- Folder Shortcuts => Home => casuser => cybersvdd.sas





More Information

- Viya Jupyter Notebook SVDD code (GitHub)

https://github.com/sassoftware/sas-viya-programming/blob/master/high-frequency-analytics/Support%20Vector%20Data%20Description%20(SVDD)%20to%20identify%20Turbofan%20Engine%20Asset%20Degradation.ipynb

- Video demonstration https://www.youtube.com/watch?v=tGL5AUSzHLk
- Python Jupyter example: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html
- One class classification http://homepage.tudelft.nl/n9d04/thesis.pdf

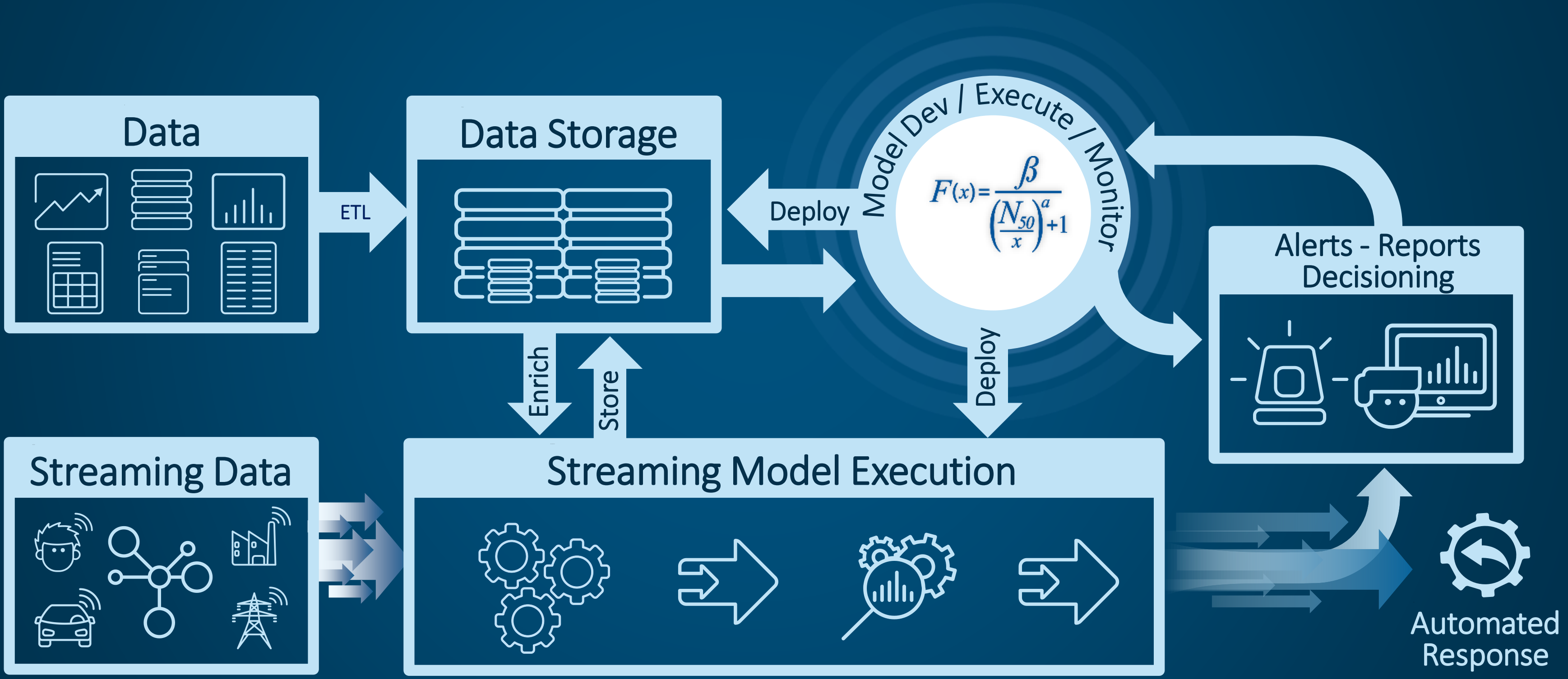
Exercise Review

SAS Event Stream Processing (ESP)

X

Analytics Lifecycle

IoT Analytics Lifecycle



SAS Event Stream Processing

Functional Architecture

Streaming Data

STUDIO

STREAMVIEWER

EVENT STREAM MANAGER

PUBLISHING INTERFACE

EVENT STREAM PROCESSING ENGINE

- Processes data **continuously**, on the **move**, in-**memory** with very high **speed** and **low latency**
- Apply rules and analysis using a **dataflow centric ESP model**

Filtering, aggregation, thresholding, pattern detection, calculations, correlations, machine learning, text mining, geofencing, image analytics and much more…

SUBSCRIBING INTERFACE

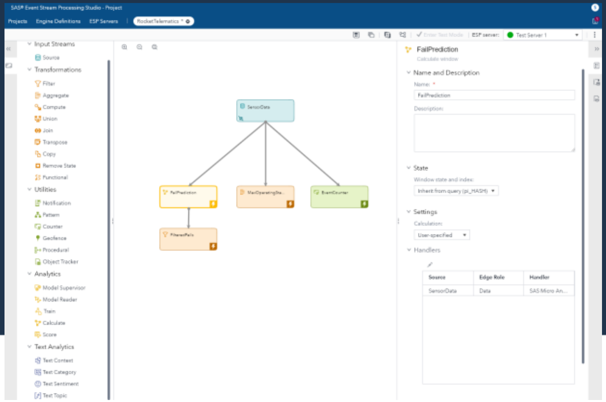
Streaming Data

SAS® Event Stream Processing

A Governed & Flexible, Design Environment

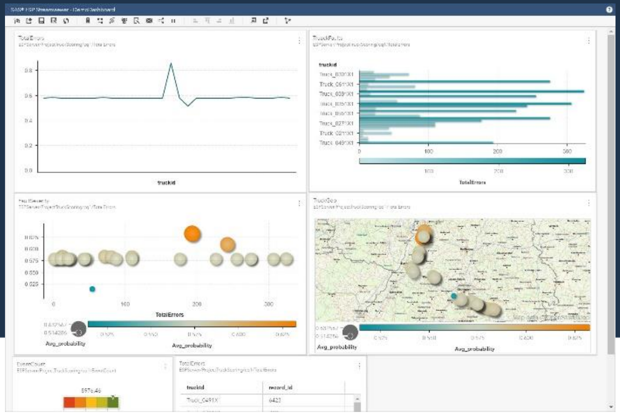
Studio

Visual Dataflow Modeling Interface
Model definition and maintenance simplified
Full set of components to build any type of process
Interactive model testing
Flexibility of Visual, XML, Python or C modeling



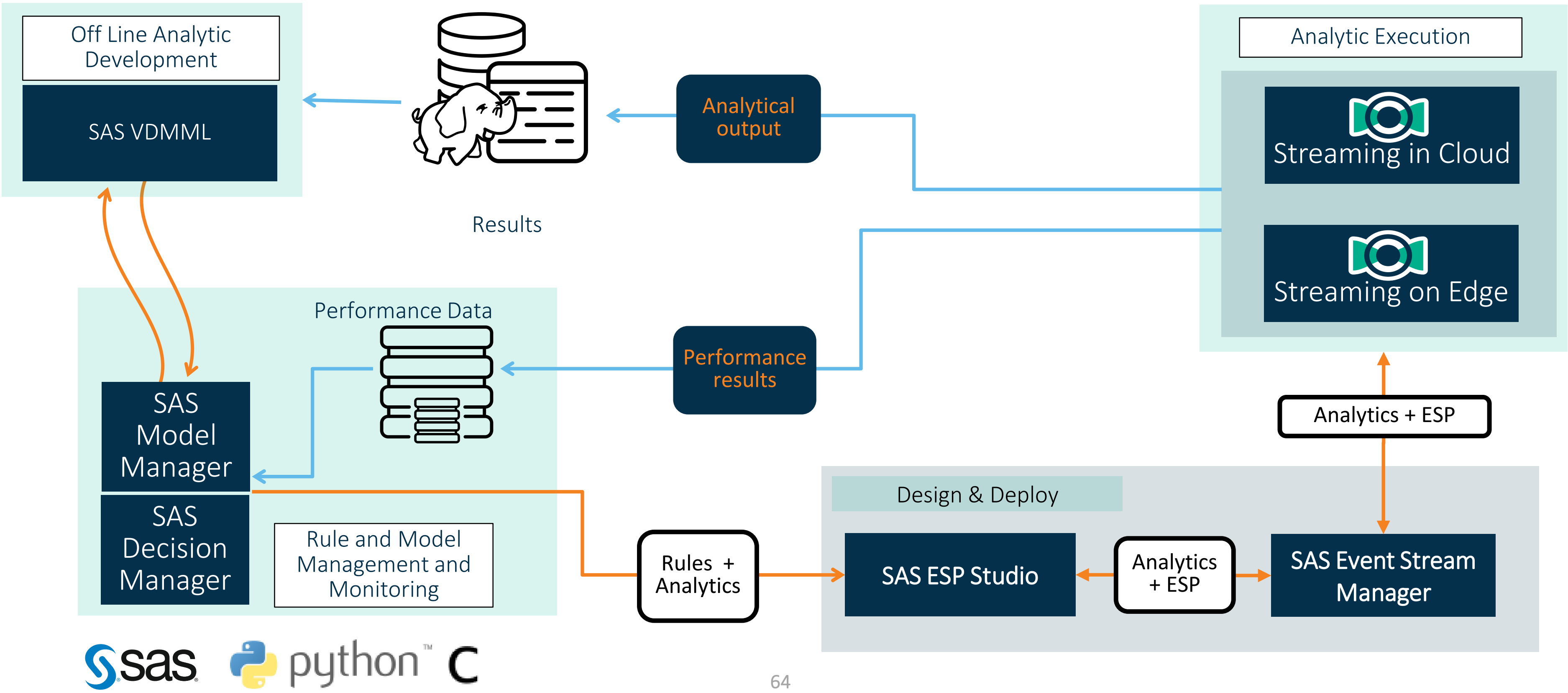
Streamviewer

Real-Time Dashboards for live event streams monitoring
Create, embed and share dashboards
HTML5 and SAS® Graphs visualization
View Multiple Models across different ESP Server



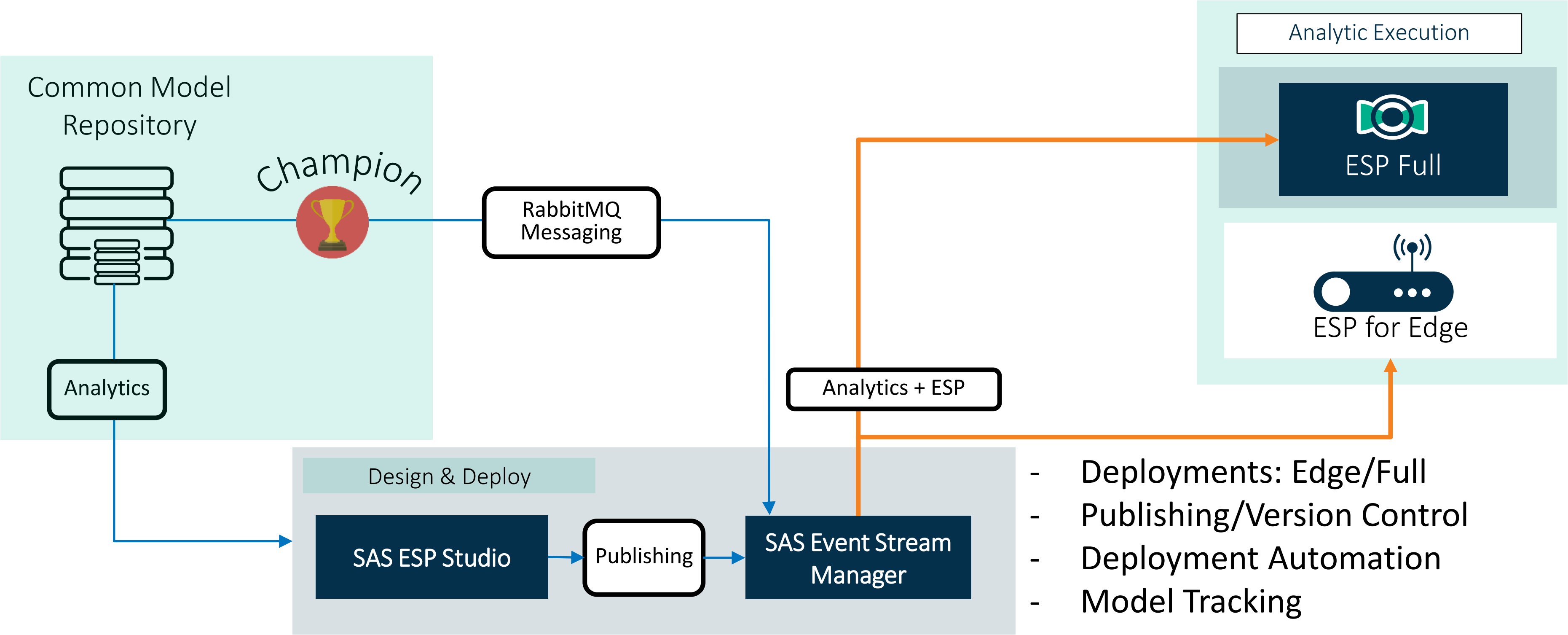
SAS Analytics Ecosystem

Dev – Test – Deploy – Monitor - Improve



Operationalization

Putting streaming analytics to work



- Deployments: Edge/Full
- Publishing/Version Control
- Deployment Automation
- Model Tracking

Wrap-Up

Section Review

Cybersecurity Analytics Maturity

Anomaly Detection

- Big data management
- Flags, rules, and alerts

- Multivariate statistics, inference & unsupervised machine learning
- Segments extracted as baselines

Data-aware Investigations

<u>Understanding</u>

- Feature engineering
- Labeling
- Diagnostics
- *Unsupervised ML*

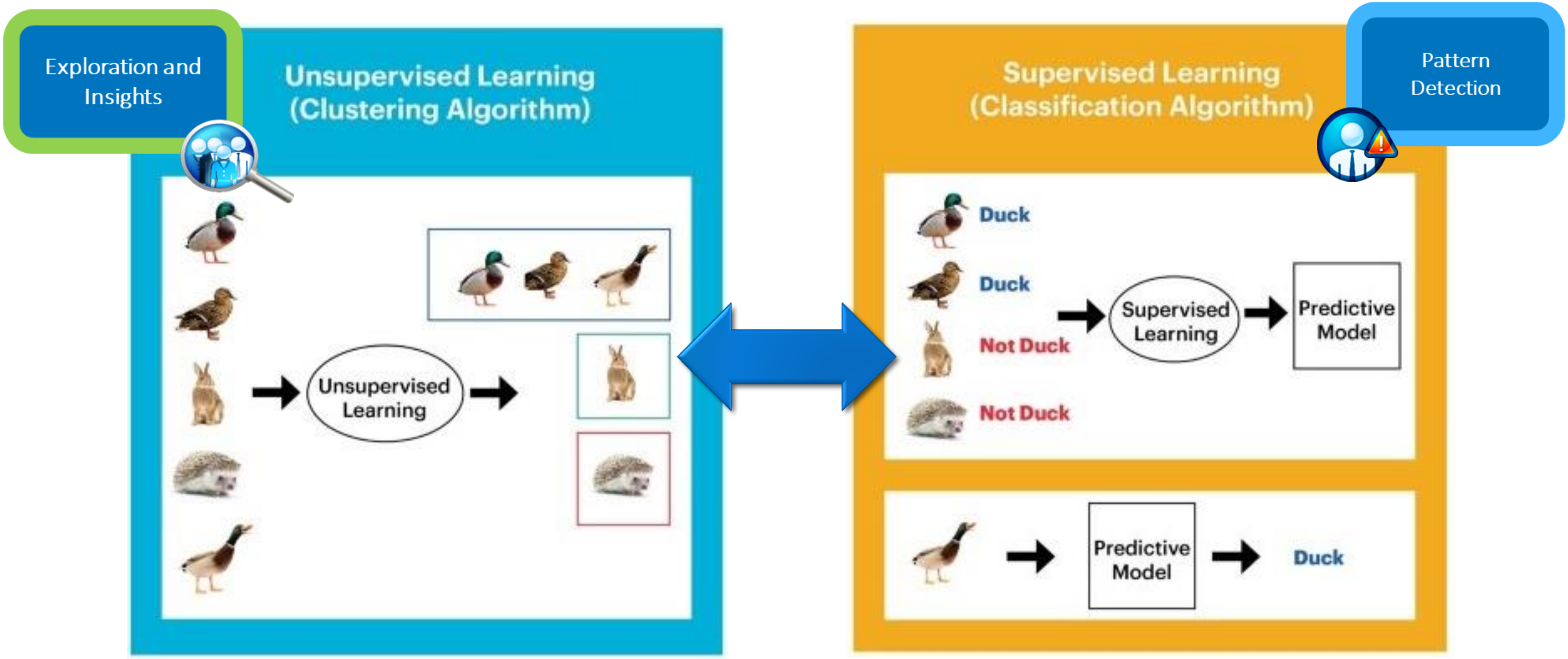
Predictive Detection

?

Risk Awareness / Resource Optimization

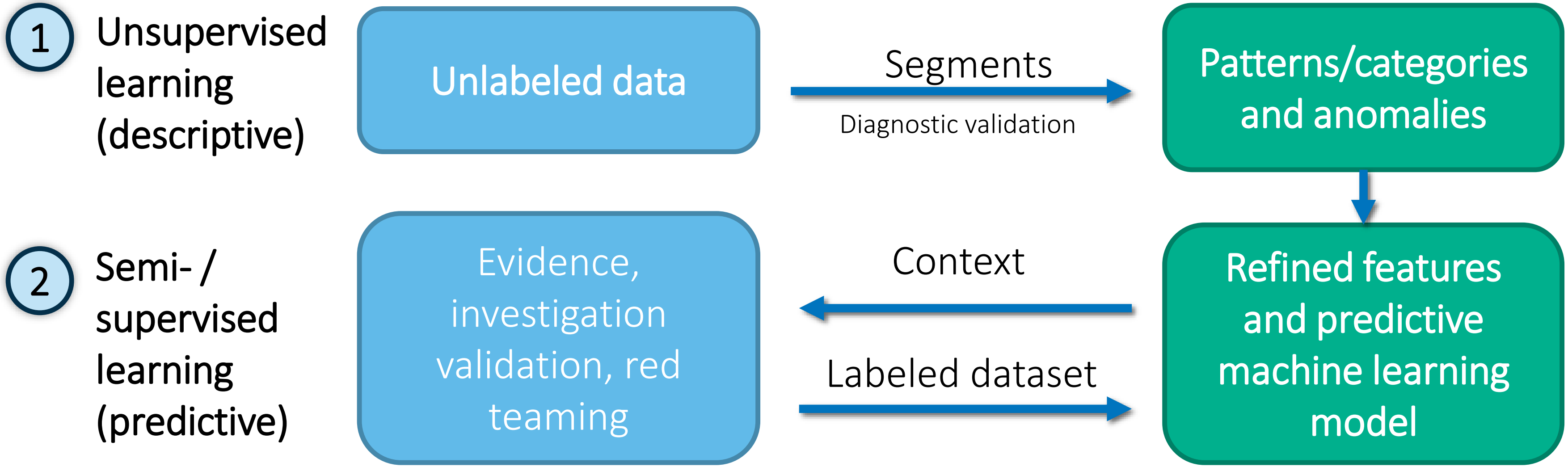
?

Machine Learning Segmentation and Classification

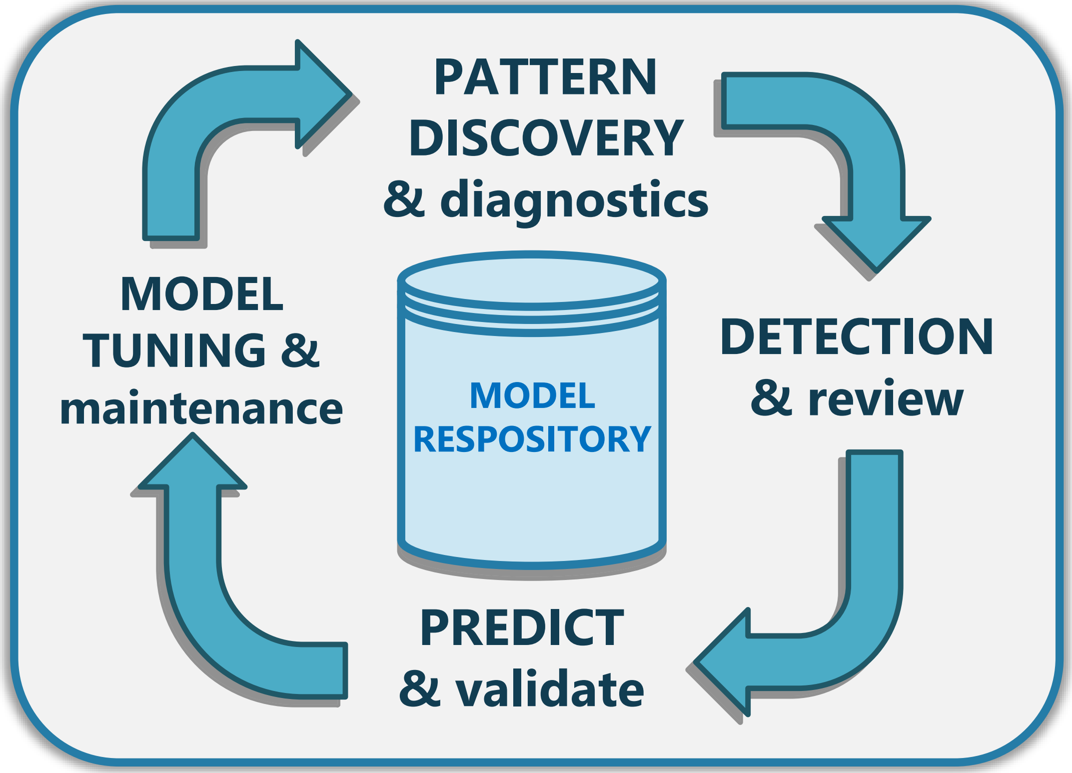


https://medium.com/datadriveninvestor/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6

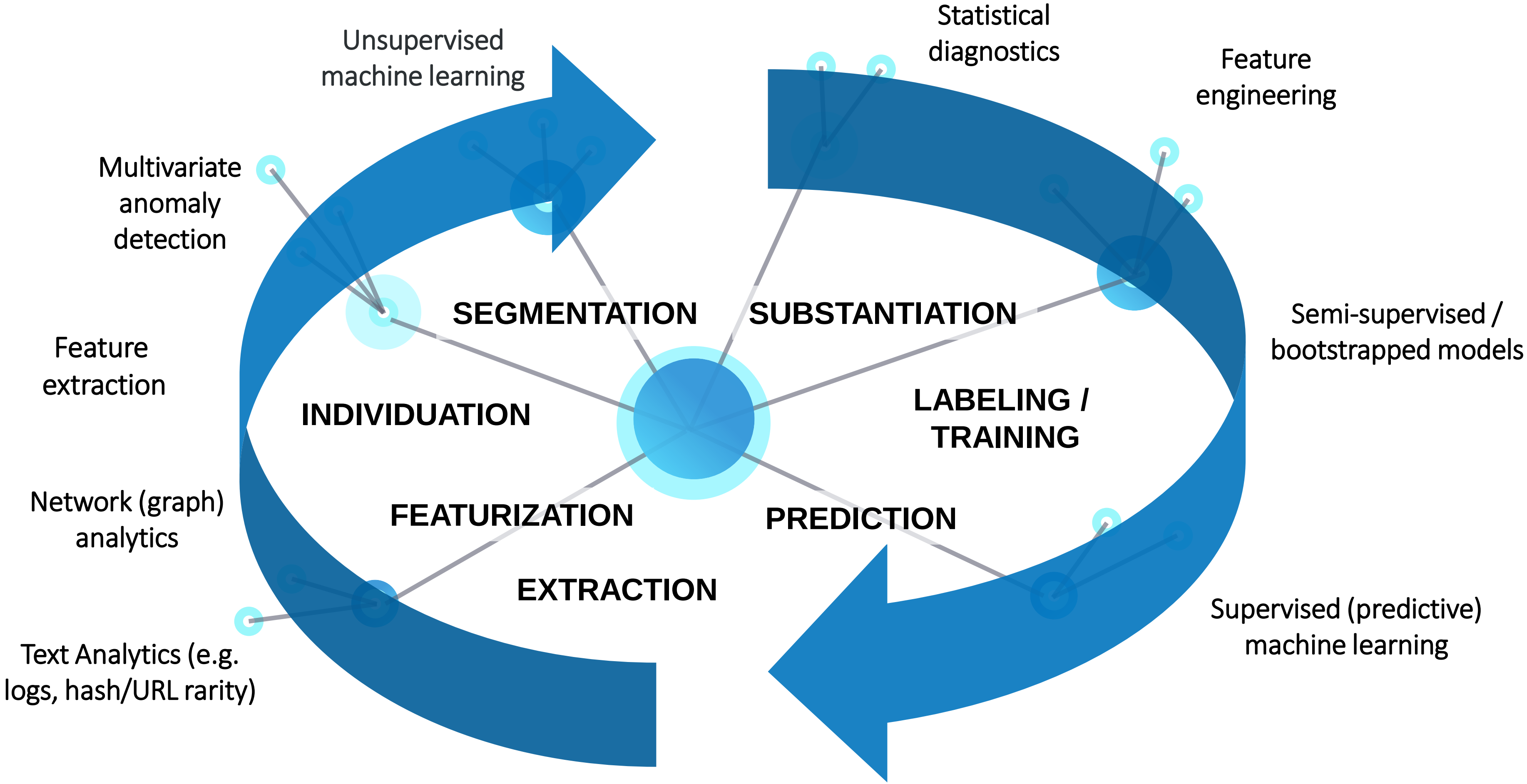
From Anomalies to Focused Incident Detection



Machine Learning as a Process



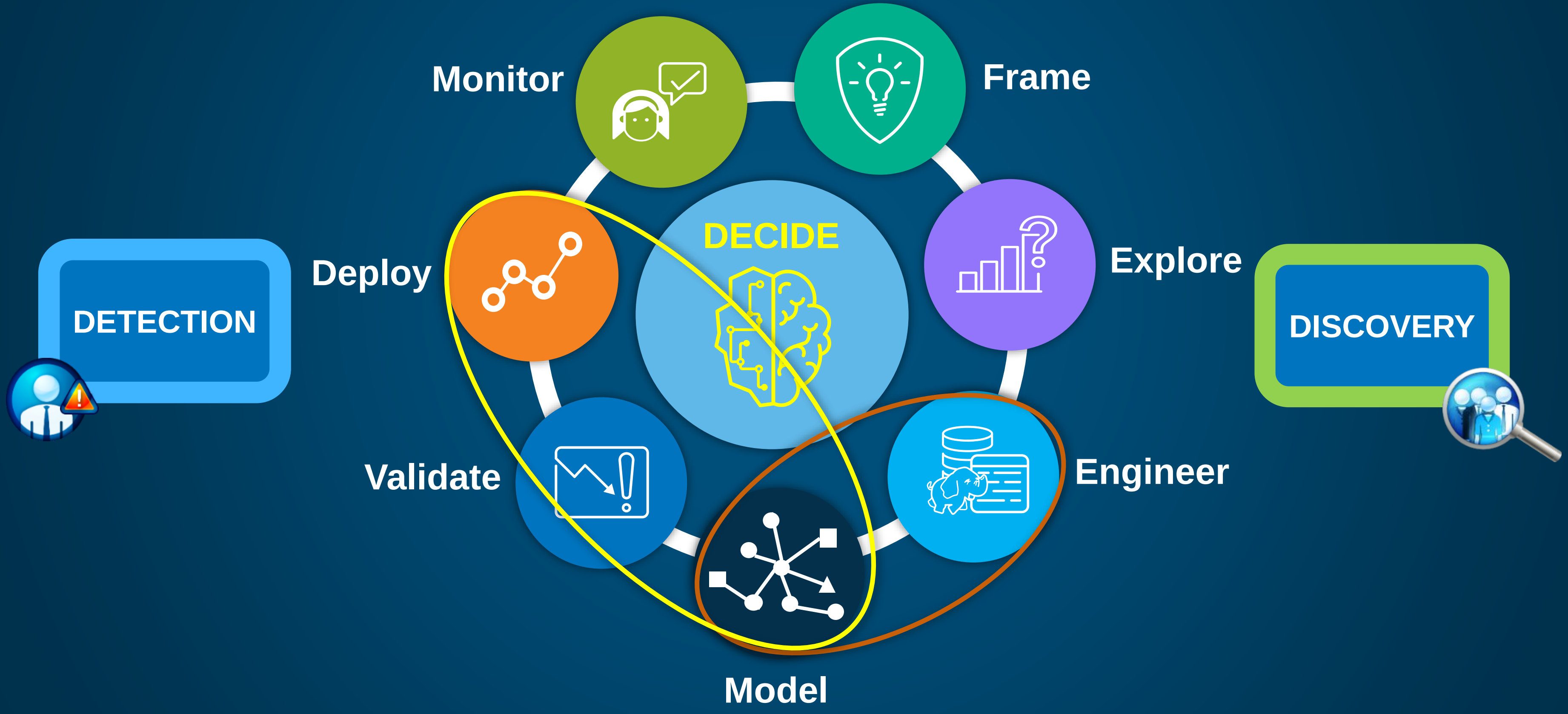
Applied Cybersecurity Analytics Process



SOURCES

http://www.oreilly.com/data/free/archive.html

Cybersecurity Data Science (CSDS) Lifecycle



REFERENCES

Anomaly Detection

REFERENCES: Cybersecurity Anomaly Detection

D. Barbara, Y. Li, J. Couto, J.-L. Lin, and S. Jajodia. Bootstrapping a Data Mining Intrusion Detection System. Symposium on Applied Computing , 2003.

D. Barbara, J. Couto, S. Jajodia, and N. Wu. Detecting Novel Network Intrusions using Bayes Estimators. SIAM Conference on Data Mining , 2001.

C. Chow, sand D. Yeung. Parzen-Window Network Intrusion Detectors. International Conference on Pattern Recognition , 4, 2002.

E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A Geometric Framework for Unsupervised Anomaly Detection, In Applications of Data Mining in Computer Security . Kluwer, 2002.

C.Kruegel,D.Mutz,W.Robertson,andF.Valeur. Bayesian Event Classification for Intrusion Detection. Computer Security Applications Conference , 2003.

C. Kruegel, T. Toth, and E. Kirda. Service Specific Anomaly Detection for Network Intrusion Detection. ACM symposium on Applied computing , 2002.

REFERENCES: Cybersecurity Anomaly Detection II

M.Mahoney,andP.Chan.Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks, ACM KDD Conference , 2002.

M. Mahoney, and P. Chan. Learning Rules for Anomaly Detection of Hostile Network Traffic, ICDM Conference , 2003.

K. Sequeira, and M. Zaki. ADMIT: Anomaly-based Data Mining for Intrusions, ACM KDD Conference , 2002.

M. Thottan, and C. Ji. Anomaly Detection in IP Networks. IEEE Transactions on Signal Processing , 51(8), pp. 2191–2204, 2003.

N. Ye, and Q. Chen. An Anomaly Detection Technique based on a Chi-square Statistic for Detecting Intrusions into Information Systems. Quality and Reliability Engineering International , 17, pp. 105–112, 2001.

A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. SIAM Conference on Data Mining , 2003.

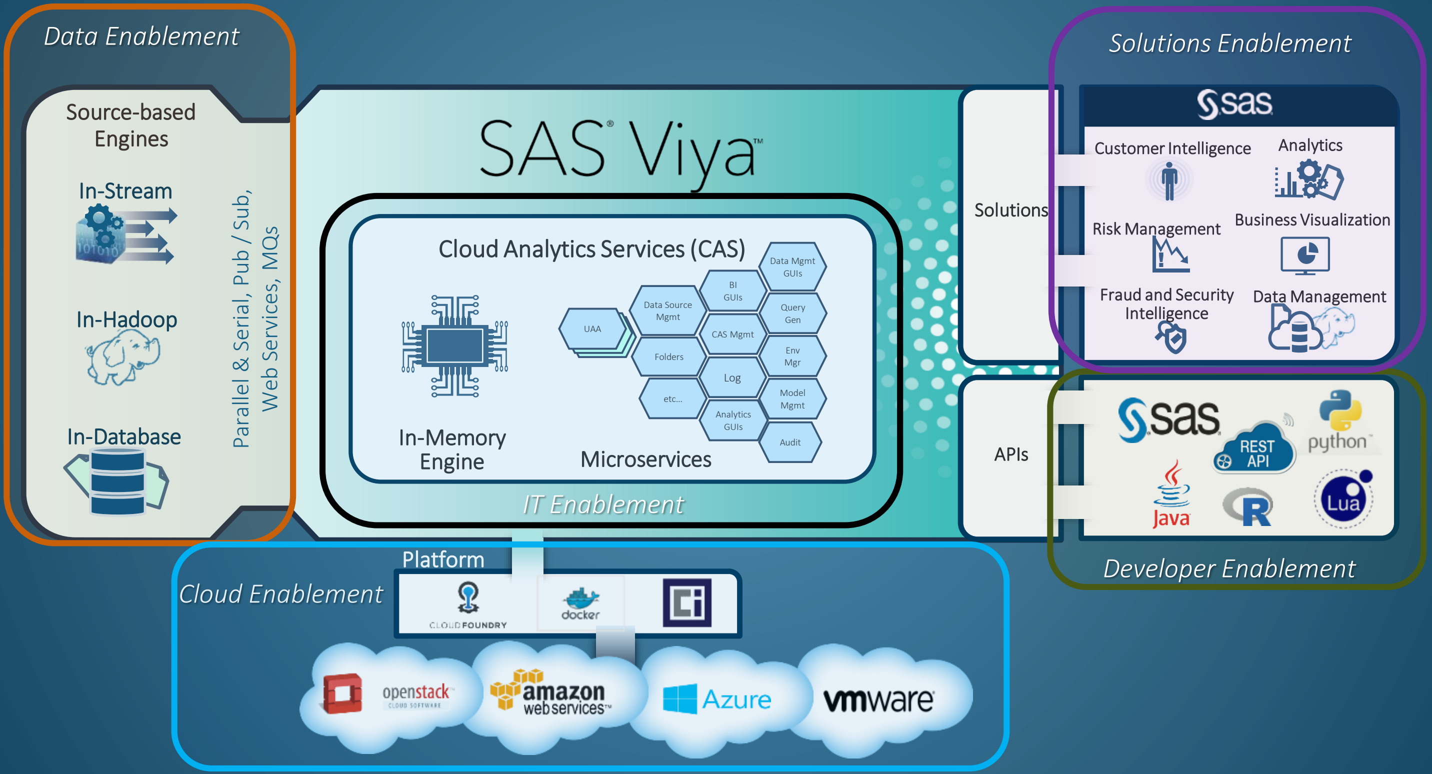
APPENDIX

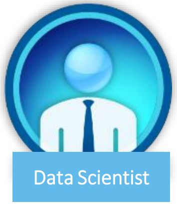
Viya VDDML & CAS

SAS Viya VDMML

Visual Data Mining and Machine Learning

SAS VIYA ANALYTICS ENGINE

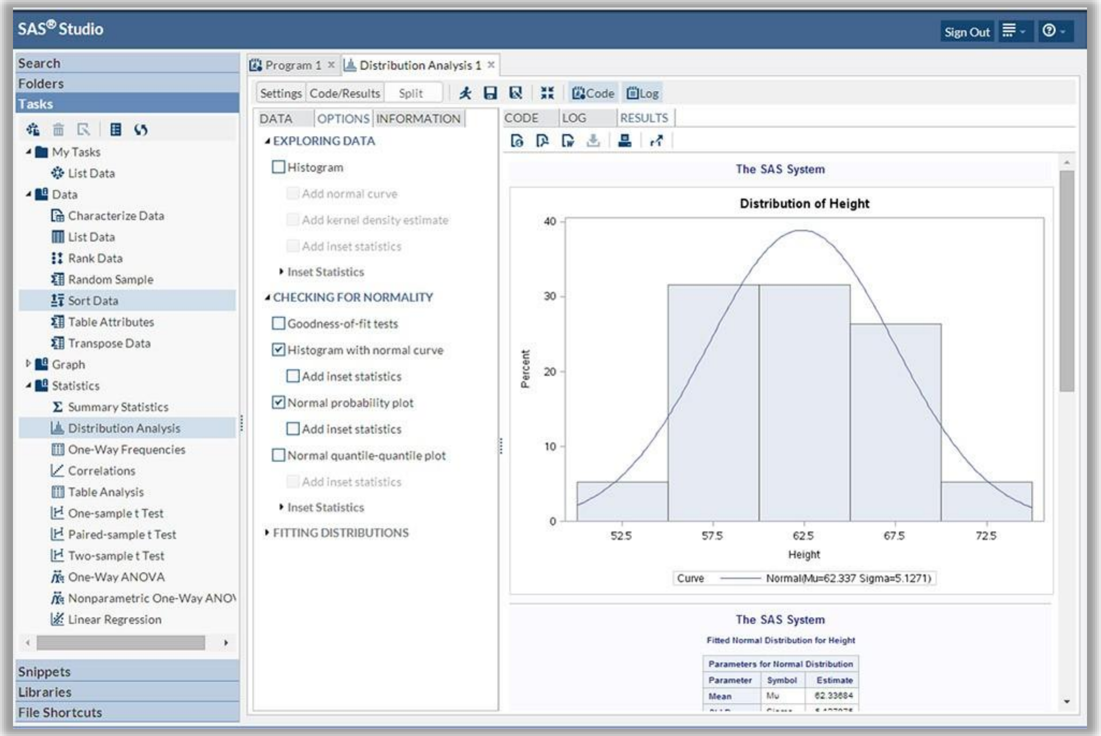




Advanced Analytics and Machine Learning

SAS VIYA VDMML – web-based interface to SAS anywhere, anytime

- Web browser-based interface
- Integrate R and Python code directly
- No client installation - zero client footprint
- Customizable environment
- Assistive programming tools
- Automatic code generation
- Integrate &/or run from cloud
- Seamlessly move between devices and maintain interactive experience
- Create and add code snippets to shared snippet library



https://www.sas.com/en_my/software/foundation/studio.html

SAS VIYA VDMML Algorithms

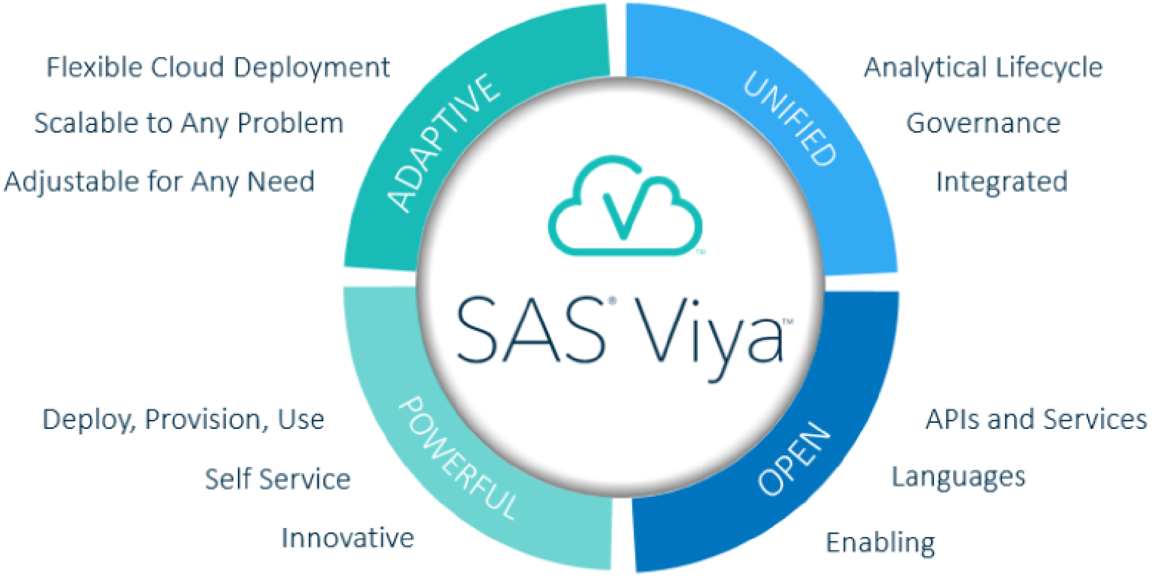
Data Wrangling	**Modeling**
Binning	Logistic Regression
Cardinality	Linear Regression
Imputation	Generalized Linear Models
Transformations	Nonlinear Regression
Transpose	Ordinary Least Squares Regression
SQL	Partial Least Squares Regression
Sampling	Quantile Regression
Variable Selection	Decision Trees
Principal Components Analysis (PCA)	Forest
K-Means Clustering	Gradient Boosting
Moving Window PCA	Neural Network
Robust PCA	Support Vector Machines
	Factorization Machines
	Network / Community Detection
	Text Mining
	Support Vector Data Description

https://support.sas.com/content/dam/SAS/support/en/books/free-books/discovering-sas-viya-special-collection.pdf

SAS Viya CAS

Cloud Analytics Server

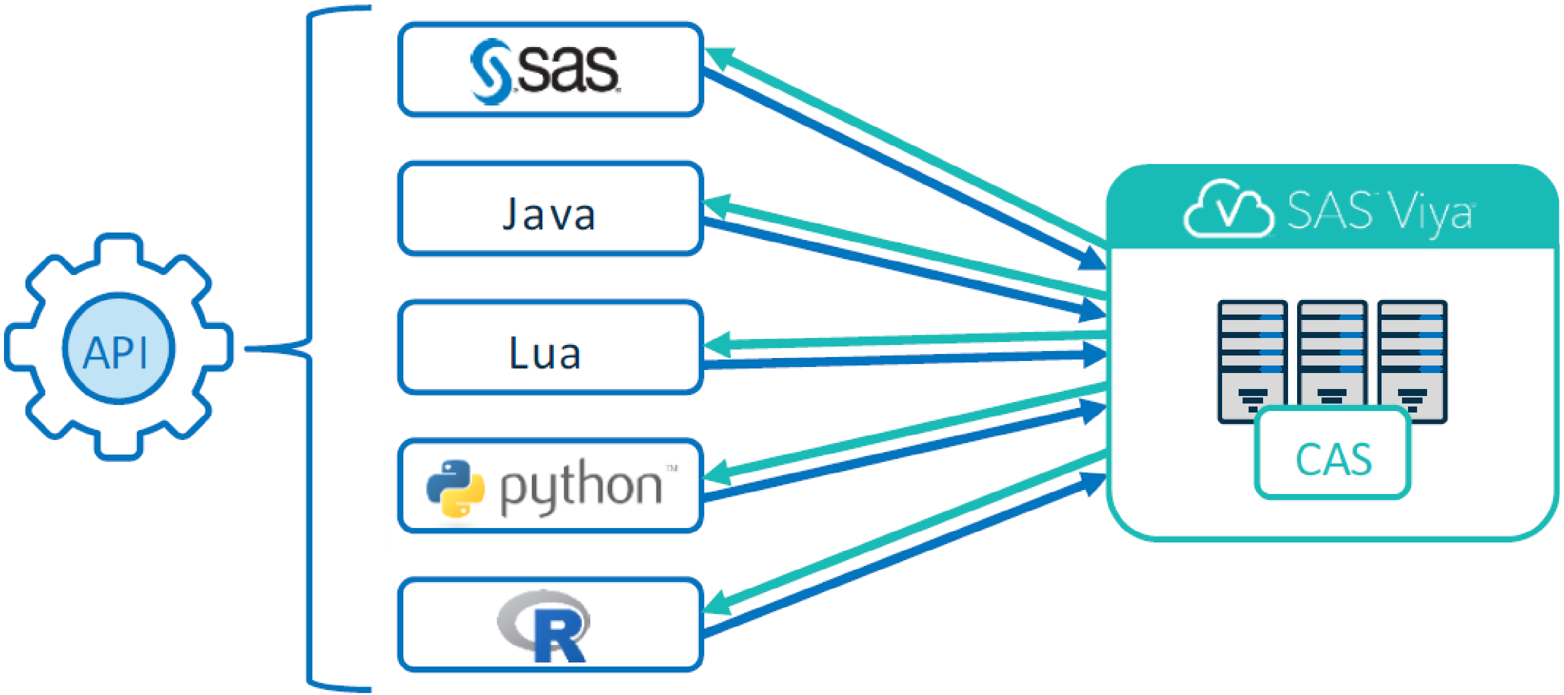
Cloud Analytic Services (CAS)



SAS Viya

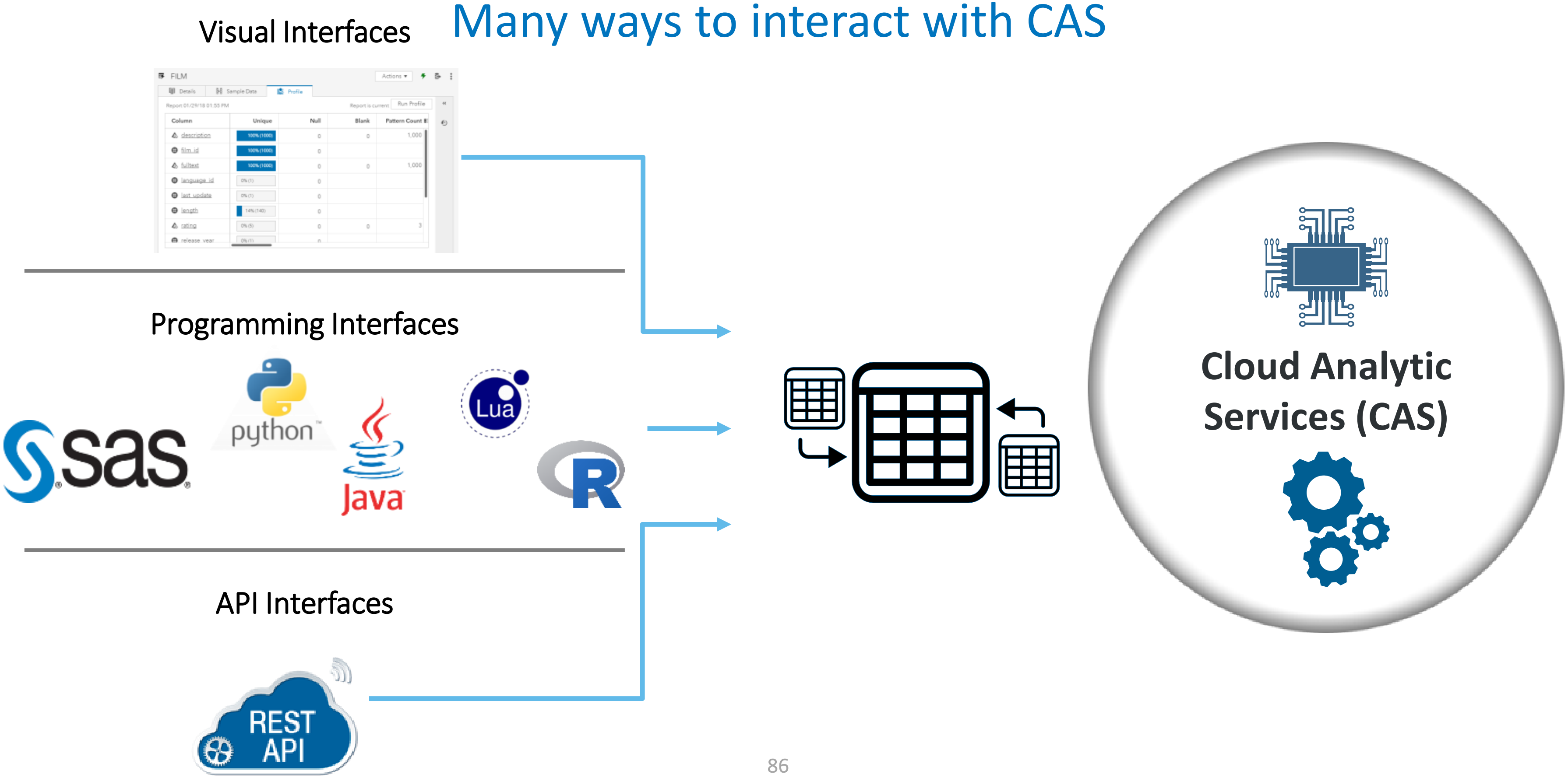
- Use open source software to take control of analytical tools.

SAS Viya is Open



How can we process data in CAS?

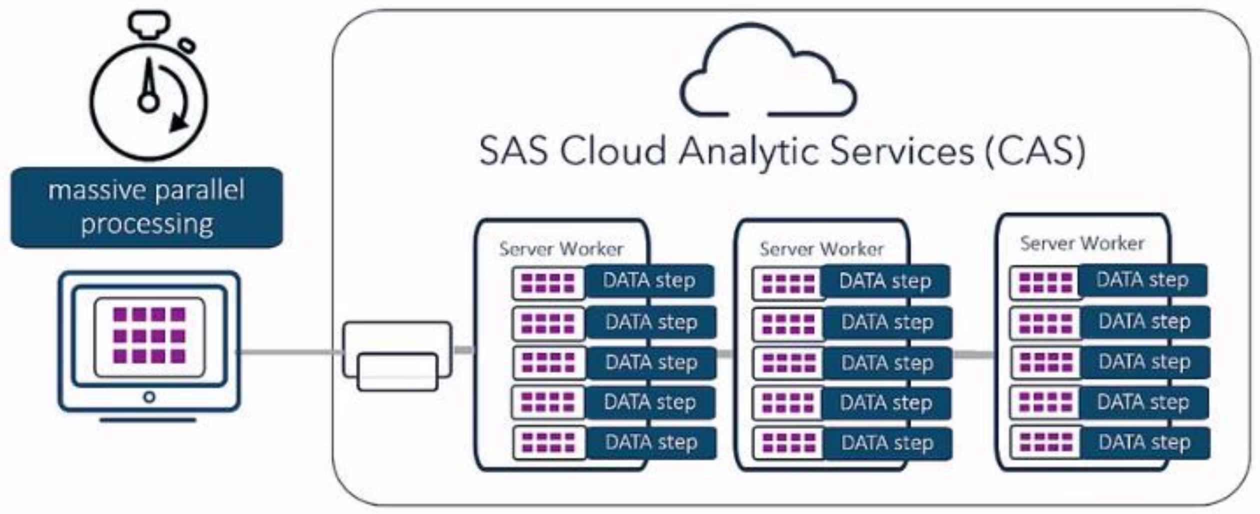
Many ways to interact with CAS



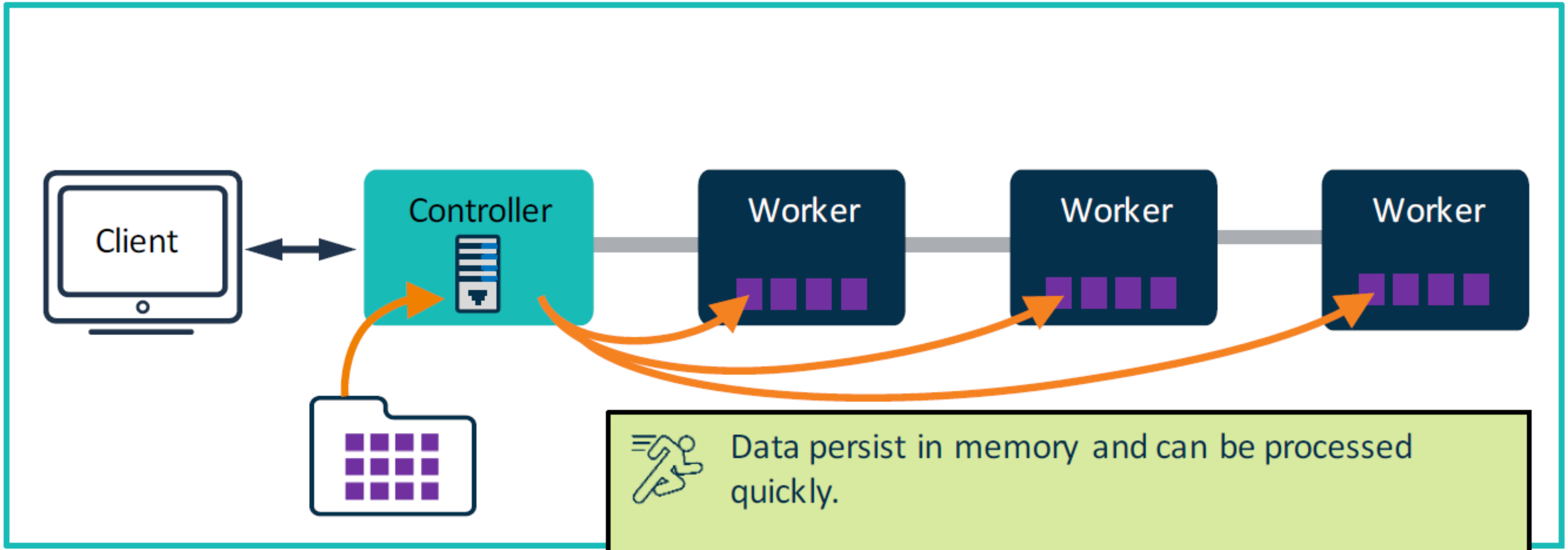
Data Processing in CAS

Massively parallel & In-memory

- DATA Step
  - Including Data Quality functions*
- DS2
- FedSQL
- Transpose
- …



CAS Distributed Environment



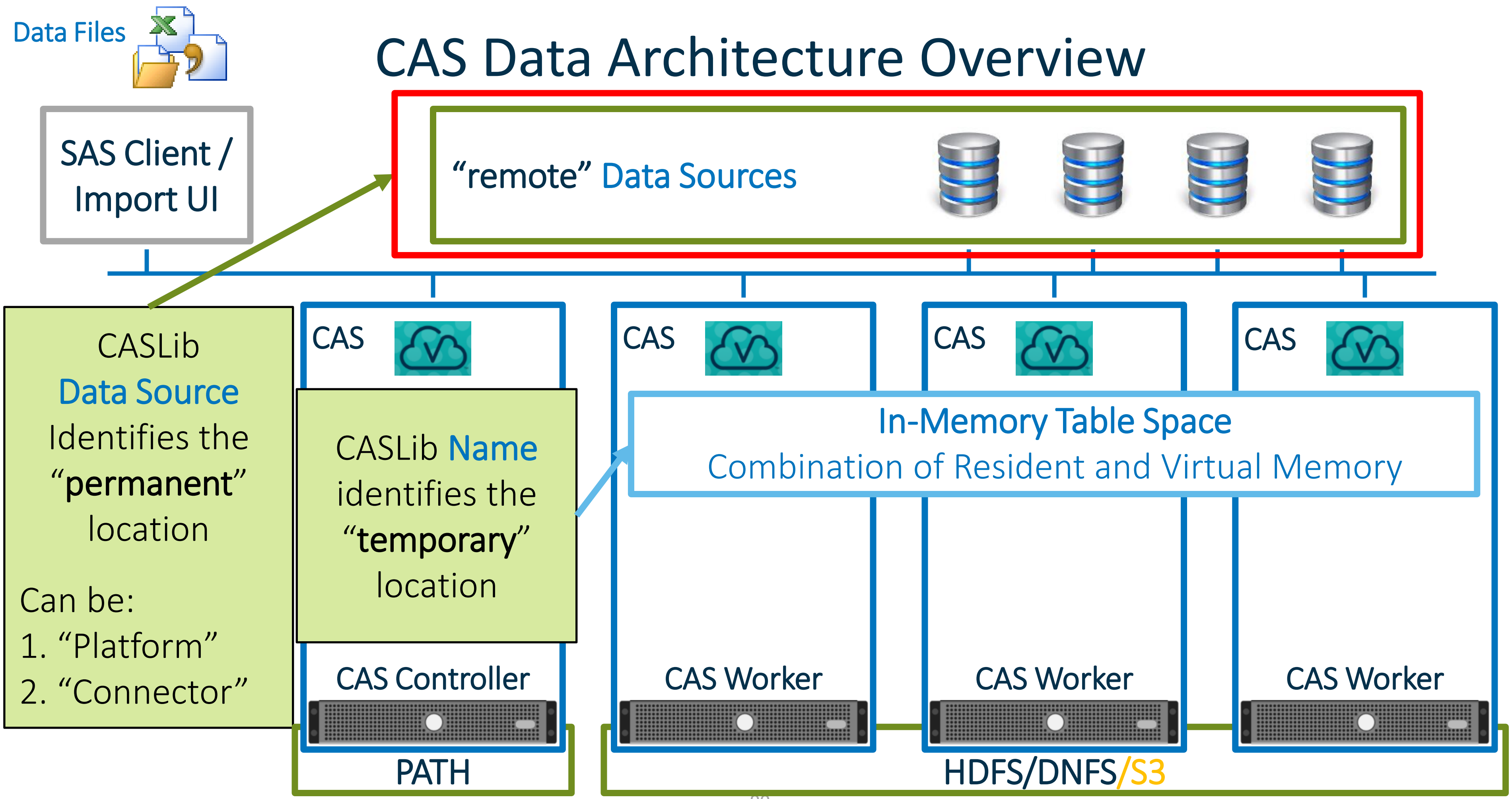
Data persist in memory and can be processed quickly.

SAS Viya algorithms are designed for distributed environments.

Supported Data Providers

[data platform]	SAS/ACCESS to	SAS Data Connector to
Amazon Redshift	●	●
DB2	●	●
Hadoop	●	●
Impala	●	●
Microsoft SQL Server	●	●
ODBC	●	●
Oracle	●	●
PostgreSQL	●	●
PC Files	●	●
SAP HANA	●	●
Teradata	●	●
JDBC	●	●
MySQL	●	●
Spark (LA)	●	●
Vertica	●	●

CAS Data Architecture Overview



Understanding CAS Libraries

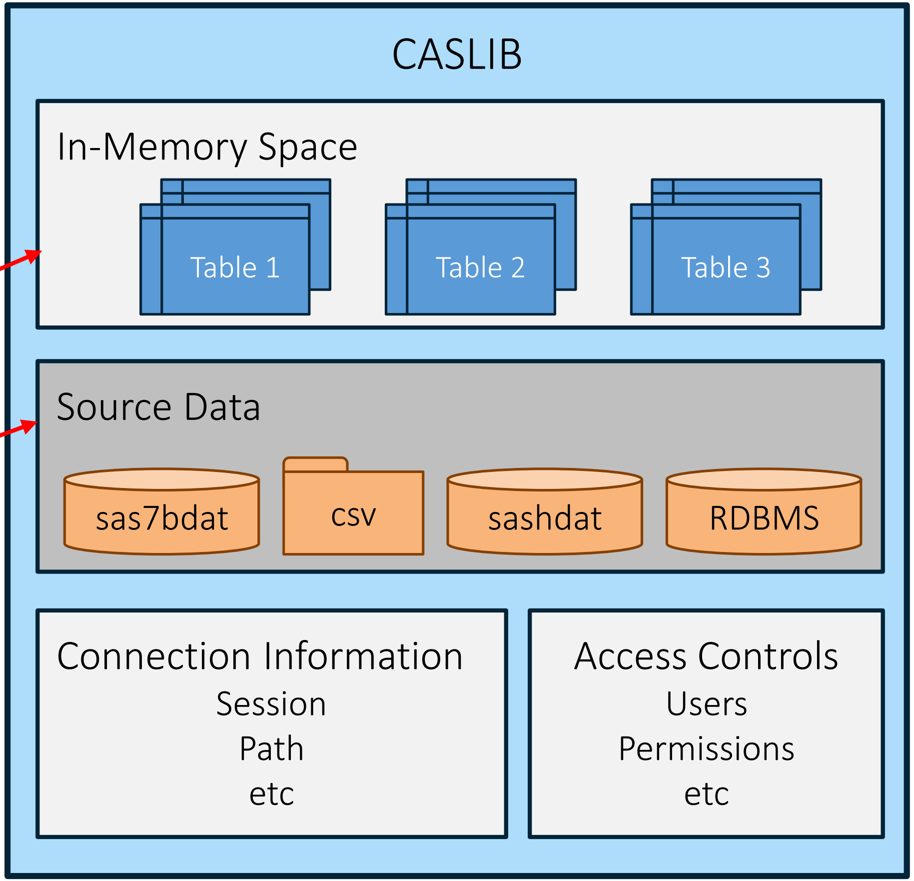
A CAS library – analogous to a SAS library

Provides access to:

- A CAS In-memory table space
- Files/Tables in a data source (DBMS Tables; Directory files)

- Examples:
- caslib **caspth** path="/data/cust/" type=path;
- caslib **caspgdvd** **dataSource**=(srcType="**postgres**", server="pg1.sas.com", database=d1, port=5432, …);

Session versus **Global** CAS Libraries



…