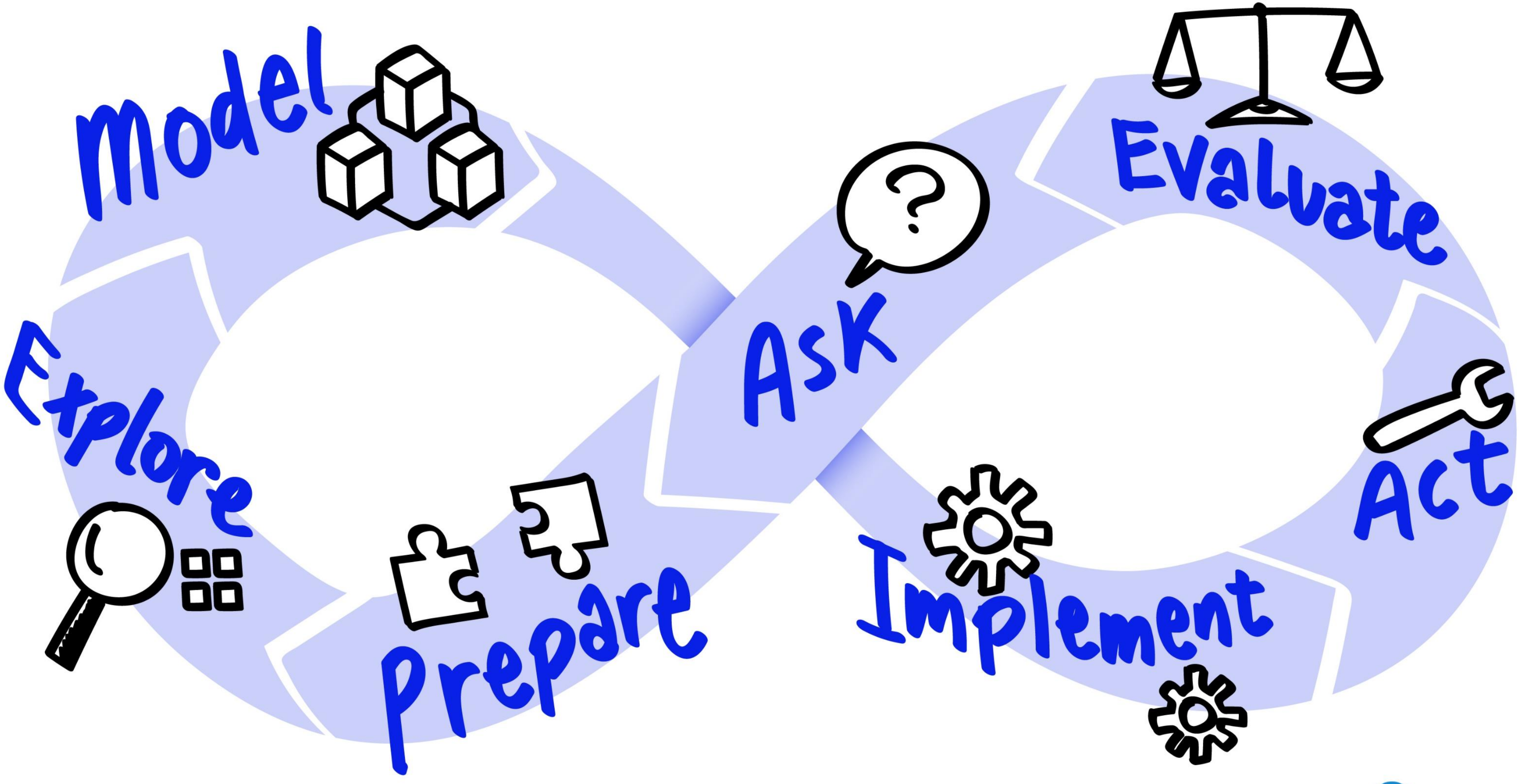
Data Science Jam Sessions by SAS

Talk of the GEEKS

sas THE POWER TO KNOW.

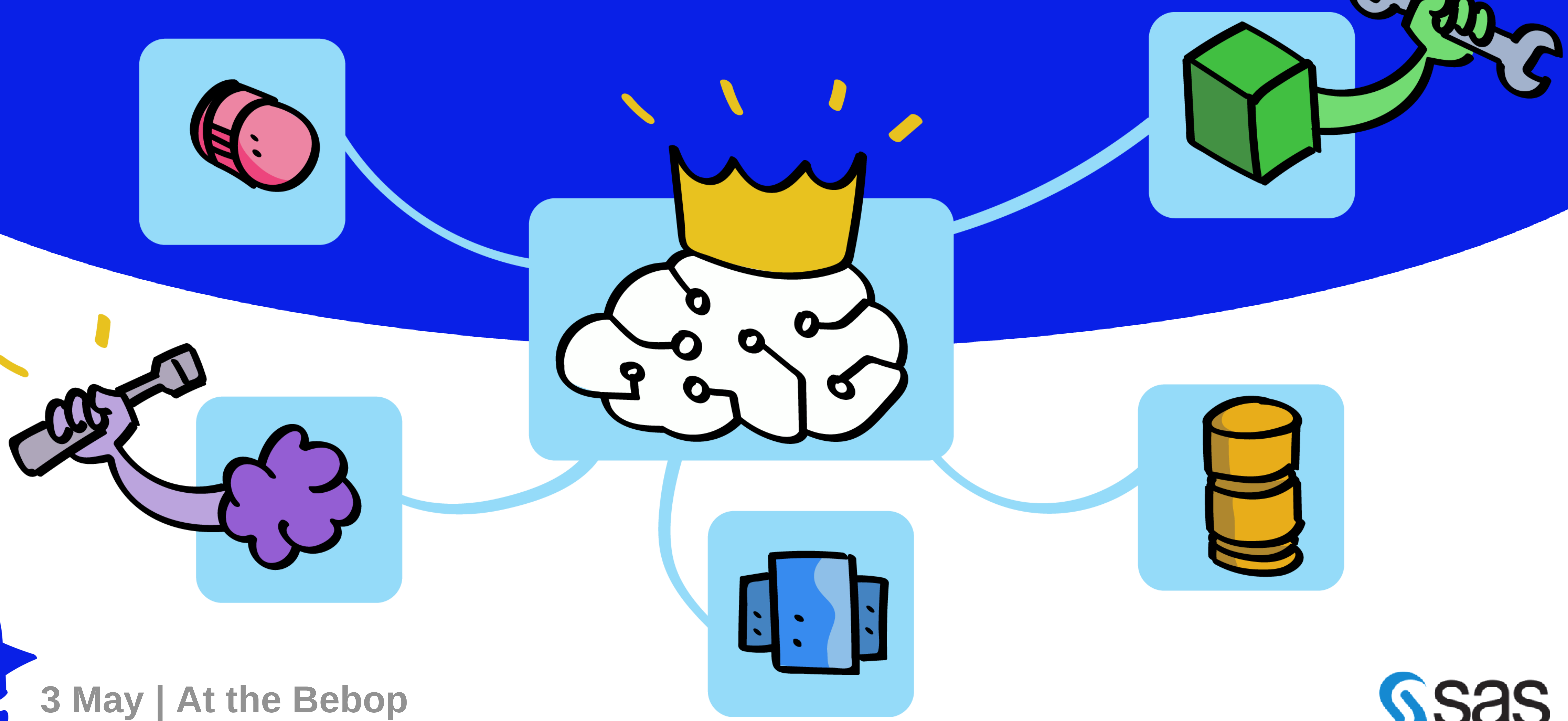
Analytical Lifecycle



sas THE POWER TO KNOW.

The DATA step in SAS VIYA

Speaker: Gwendoline Planchon



3 May | At the Bebop

The DATA step in SAS VIYA

What?

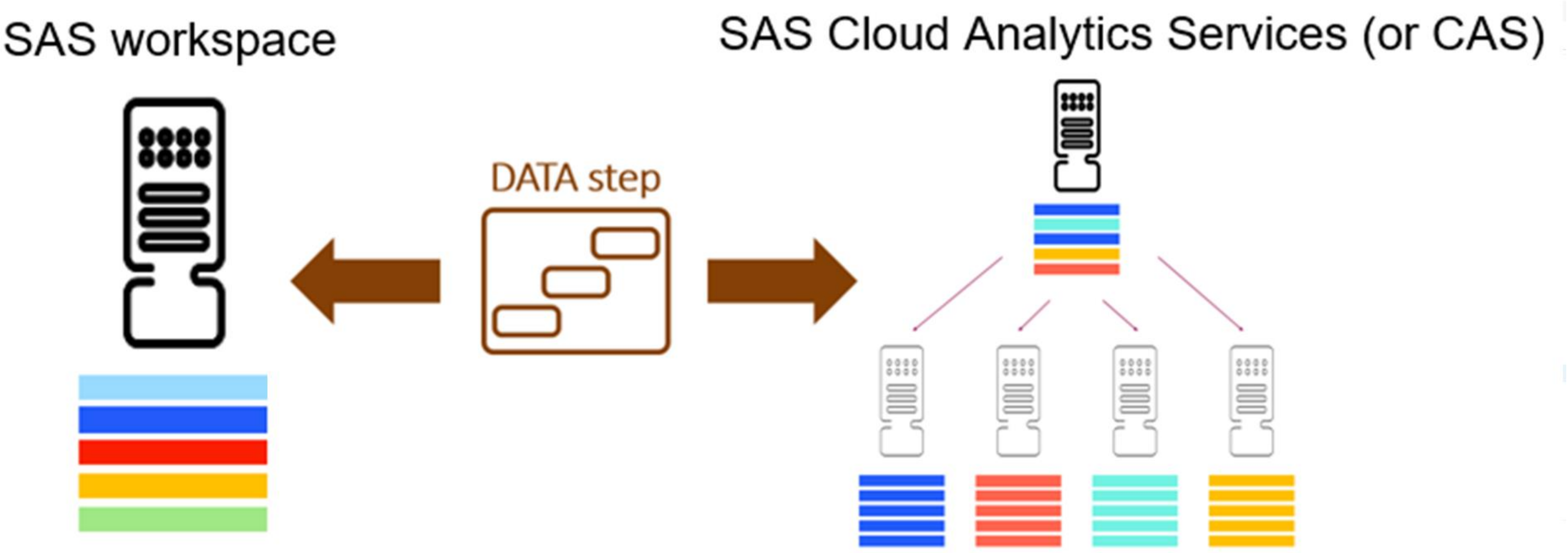
Cool thing about SAS Viya ?

Your data is processed **simpler** and **faster** !

```
data DataStep;
set sashelp.iris;
    length CatPetalLength $10;
    by Species;
    array Petals(*) Petal:;
    retain PowerSummation 0;

    AbbrevSpecies = substr(Species,1,3);

    if first.Species then SepalWidth = 0;
    if last.Species then SepalLength = 0;

    PowerSummation = Powersummation + _n_**2;
    TotalPetal = sum(of Petals{*});

    CatPetalLength = put(PetalLength,3.);
    output;
    drop PetalLength;
run;
```

The DATA step in SAS VIYA

Why?



- Single thread
- SAS V9 engine

- Multiple (or single) threads
- In-memory data

The DATA step in SAS VIYA

Method

1. Connect to a CAS session
2. Load data into CAS using PROC CASUTIL
3. Run your DATA step
4. Check the SAS log

```
/*****************************/
/**** SAS Viya DATA Step ****/
/*****************************/

cas mysess sessopts=(caslib=casuser);

libname mycas cas;

proc casutil;
    load data=sashelp.iris replace;
run;

data mycas.iris_proc;
    set mycas.iris;
run;
```

```
        data mycas.iris_proc;
        set mycas.iris;
        run;

NOTE: Running DATA step in Cloud Analytic Services.
NOTE: The DATA step will run in multiple threads.
NOTE: There were 150 observations read from the table IRIS in caslib CASUSER.
NOTE: The table iris_proc in caslib CASUSER has 150 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.28 seconds
      cpu time            0.01 seconds
```

The DATA step in SAS VIYA

Method

1. **Connect to a CAS session**
2. Load data into CAS using PROC CASUTIL
3. Run your DATA step
4. Check the SAS log

```
/***************************/
/**** SAS Viya DATA Step ****/
/***************************/

cas mysess sessopts=(caslib=casuser);

libname mycas cas;

proc casutil;
     load data=sashelp.iris replace;
run;

data mycas.iris_proc;
     set mycas.iris;
run;
```

```
85         data mycas.iris_proc;
86         set mycas.iris;
87         run;

NOTE: Running DATA step in Cloud Analytic Services.
NOTE: The DATA step will run in multiple threads.
NOTE: There were 150 observations read from the table IRIS in caslib CASUSER.
NOTE: The table iris_proc in caslib CASUSER has 150 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.28 seconds
      cpu time            0.01 seconds
```

The DATA step in SAS VIYA

Method

1. Connect to a CAS session

2. **Load data into CAS using PROC CASUTIL**

3. Run your DATA step

4. Check the SAS log

```
/****************************/
/**** SAS Viya DATA Step ****/
/****************************/

cas mysess sessopts=(caslib=casuser);

libname mycas cas;

proc casutil;
    load data=sashelp.iris replace;
run;

data mycas.iris_proc;
    set mycas.iris;
run;
```

```
85         data mycas.iris_proc;
86         set mycas.iris;
87         run;

NOTE: Running DATA step in Cloud Analytic Services.
NOTE: The DATA step will run in multiple threads.
NOTE: There were 150 observations read from the table IRIS in caslib CASUSER.
NOTE: The table iris_proc in caslib CASUSER has 150 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.28 seconds
      cpu time            0.01 seconds
```

The DATA step in SAS VIYA

Method

1. Connect to a CAS session
2. Load data into CAS using PROC CASUTIL
3. **Run your DATA step**
4. Check the SAS log

```
/***************************/
/**** SAS Viya DATA Step ****/
/***************************/

cas mysess sessopts=(caslib=casuser);

libname mycas cas;

proc casutil;
    load data=sashelp.iris replace;
run;

data mycas.iris_proc;
    set mycas.iris;
run;
```

```
85         data mycas.iris_proc;
86         set mycas.iris;
87         run;

NOTE: Running DATA step in Cloud Analytic Services.
NOTE: The DATA step will run in multiple threads.
NOTE: There were 150 observations read from the table IRIS in caslib CASUSER.
NOTE: The table iris_proc in caslib CASUSER has 150 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.28 seconds
      cpu time            0.01 seconds
```

The DATA step in SAS VIYA

Method

1. Connect to a CAS session

2. Load data into CAS using PROC CASUTIL

3. Run your DATA step

4. **Check the SAS log**

```
/***************************/
/**** SAS Viya DATA Step ****/
/***************************/

cas mysess sessopts=(caslib=casuser);

libname mycas cas;

proc casutil;
    load data=sashelp.iris replace;
run;

data mycas.iris_proc;
    set mycas.iris;
run;
```

```
85          data mycas.iris_proc;
86          set mycas.iris;
87          run;

NOTE: Running DATA step in Cloud Analytic Services.
NOTE: The DATA step will run in multiple threads.
NOTE: There were 150 observations read from the table IRIS in caslib CASUSER.
NOTE: The table iris_proc in caslib CASUSER has 150 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.28 seconds
      cpu time            0.01 seconds
```

The DATA step in SAS VIYA

Method – Attention points

Some operations requiring inter-row dependencies will return unexpected results:

- **RETAIN** statement
- **LAG** and **DIF** functions
- **Temporary arrays**

Alternatives to guarantiee the correctness of the results:

- Run on the Workspace server
- Run on CAS using the option `/ single=yes sessref=mysess`

```
/***************************/
/*** SAS Viya DATA step ***/
/***************************/

cas mysess sessopts=(caslib=casuser);
libname mycas cas;

data irisWithRowNr;
    set sashelp.iris;

    do i = 1 to 1000;
        RowNr = (_n_*1000) - 1000 + i;
        output;
    end;

    drop i;
run;

proc casutil;
    load data=irisWithRowNr replace;
run;

data mycas.iris_proc / single=yes sessref=mysess;
    set mycas.irisWithRowNr;
    retain PowerSummation 0;

    PowerSummation = PowerSummation + rowNr**2;
run;
```

The DATA step in SAS VIYA

Method – BY statement

CODE | LOG | RESULTS | OUTPUT DATA

```
/*************************/
/*** SAS Viya DATA step ***/
/*************************/

cas mysess sessopts=(caslib=casuser);
libname mycas cas;

proc casutil;
    load data=sashelp.iris replace;
run;

data mycas.iris_BY;
    set mycas.iris;
    by species sepalLength;

    if first.species then output;
    if last.species then output;
run;
```

- No sorting (PROC SORT) required
- How does it work?
  - Distribution based on the first BY-variable
  - Sorted on same worker using the other BY-variables
- Output sorting:

```
data mycars.iris_BY (partition=(species)
                     orderby=(sepalLength));
```

CODE | LOG | RESULTS | OUTPUT DATA

Table: MYCAS.IRIS_BY View: Column names Filter: (none)

Columns

- Select all
- Species
- SepalLength
- SepalWidth
- PetalLength
- PetalWidth

Total rows: 6 Total columns: 5

	Species	SepalLength	SepalWidth
1	Setosa	43	30
2	Setosa	58	40
3	Virginica	49	25
4	Virginica	79	38
5	Versicolor	49	24
6	Versicolor	70	32

The DATA step in SAS VIYA
Conclusion

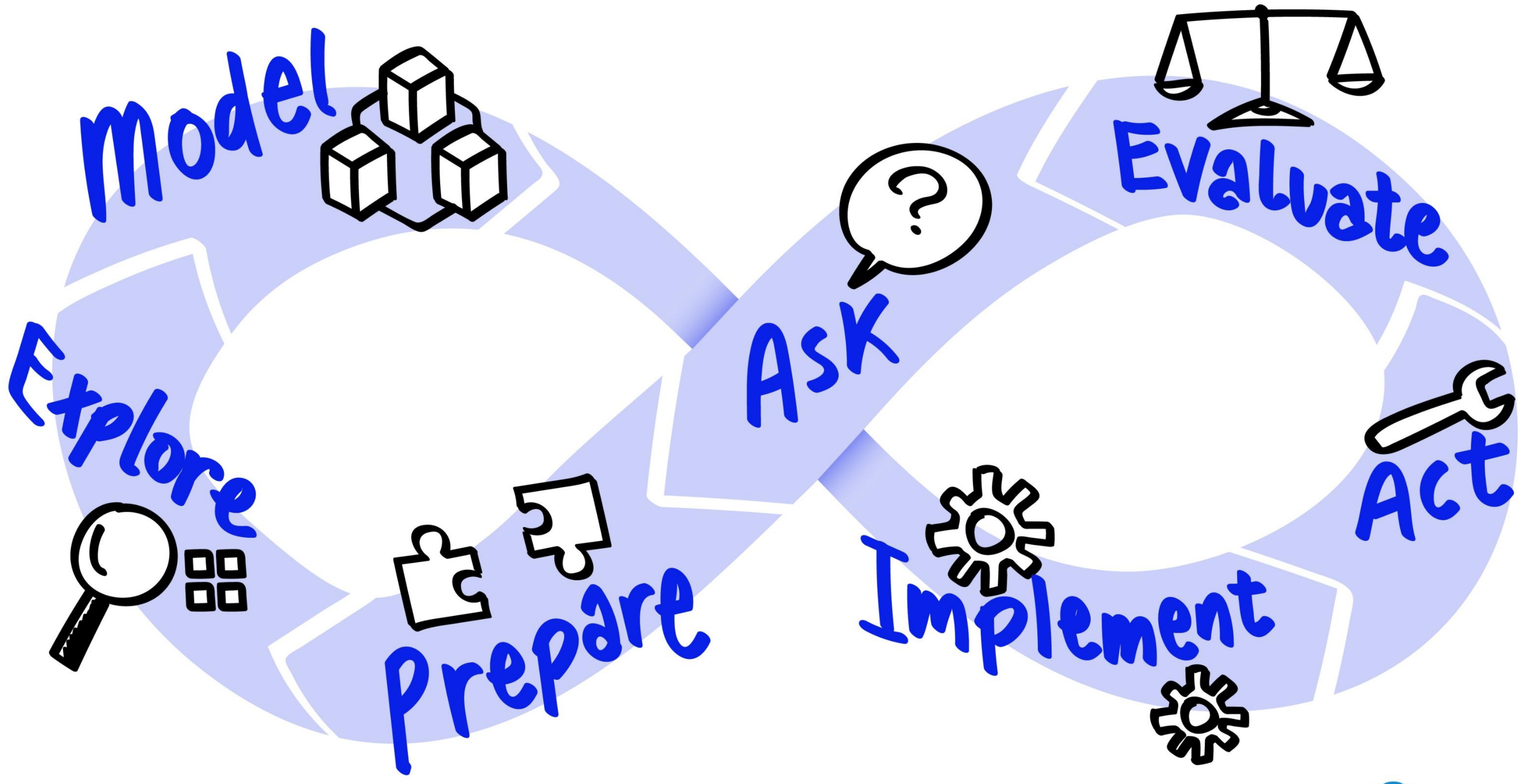
- Exactly the **same functionalities** as SAS 9.4, except for:
  - RETAIN statement
  - LAG and DIFF functions
  - Temporary arrays
- No need to **sort** the data anymore
- **In-memory**
- Much **simpler** and **faster**

Data Science Jam Sessions by SAS

Talk of the GEEKS

sas THE POWER TO KNOW.

Analytical Lifecycle

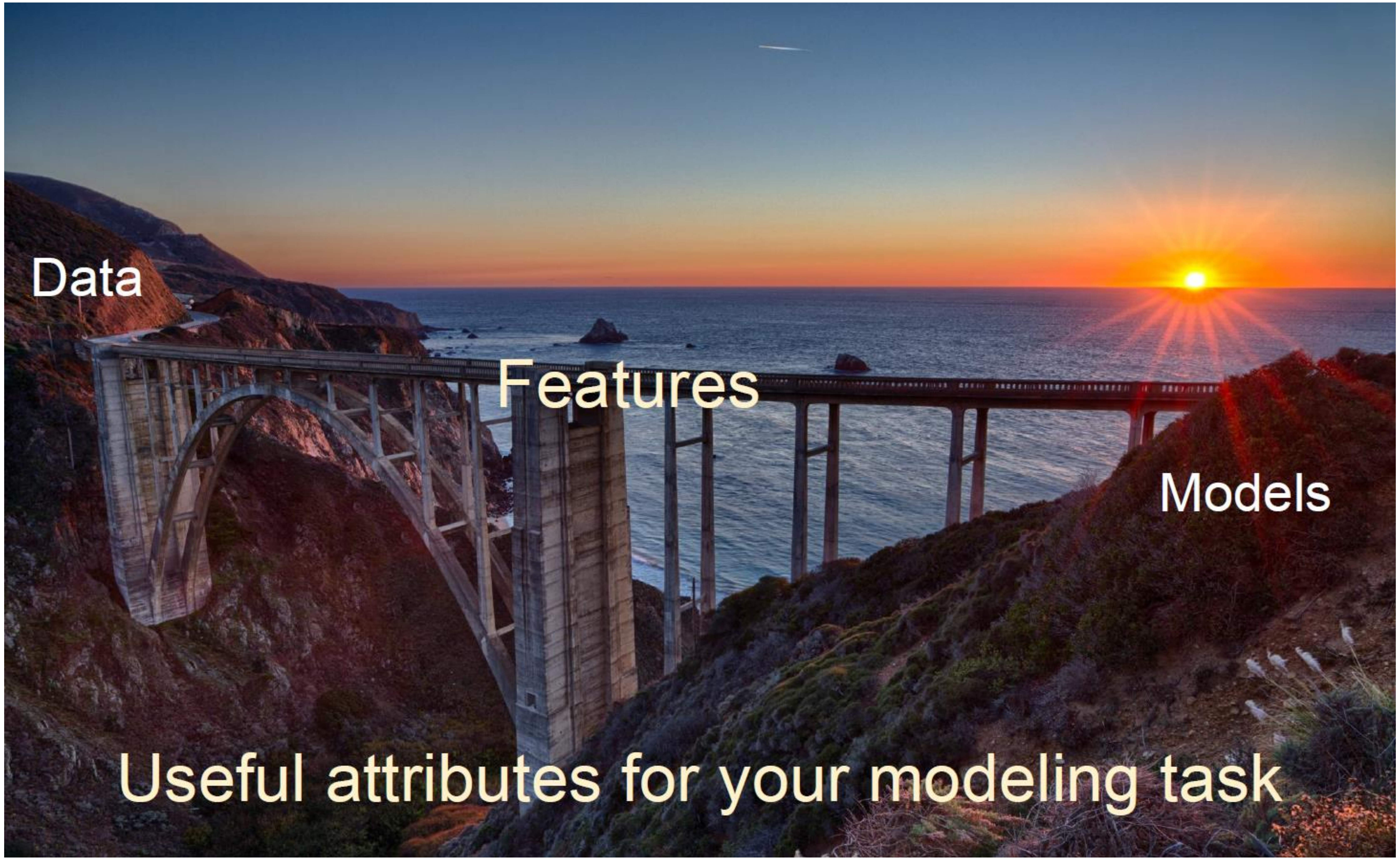


Variable binning in SAS VIYA: Increasing the predictive power of your white-box models while keeping their interpretability

Speaker: Frédéric Thys

3 May | At the Bebop

Within the field of Feature Engineering



Applied Machine Learning

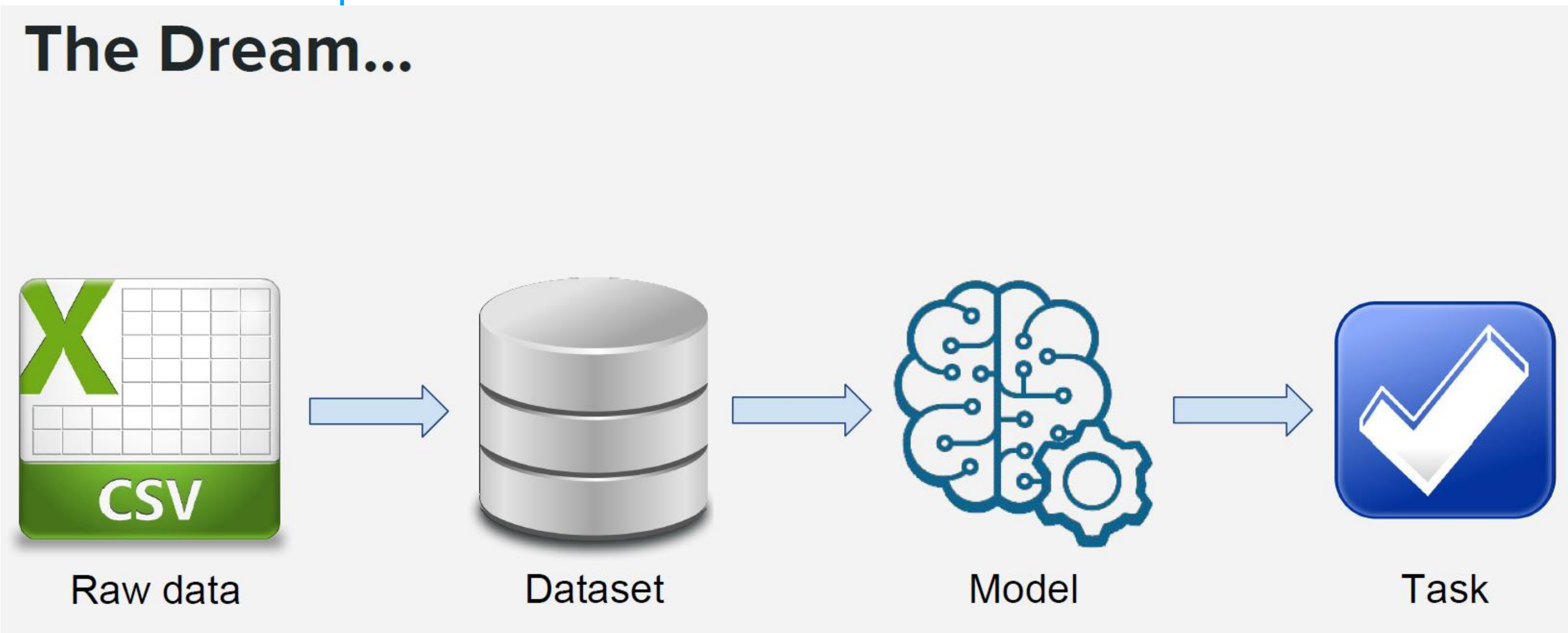
“Coming up with features is difficult,
time-consuming,
requires expert knowledge.
'Applied machine learning' is basically
feature engineering.”

– Andrew Ng

BINNING For Data Quality Control

Structure of real-world data rarely complete & straightforward

- Binning is a common step in data preparation
- You can use binning to
  - Handle **Predictors** with **Extreme Skewing**
  - Handle **Value Spikes** and **Distributions**
  - **Reduce** the **granularity** of interval variables
  - **Classify missing** variables
  - Reduce the impact of **outliers**
  - **Reveal Non Linear** behavior in relation to Target

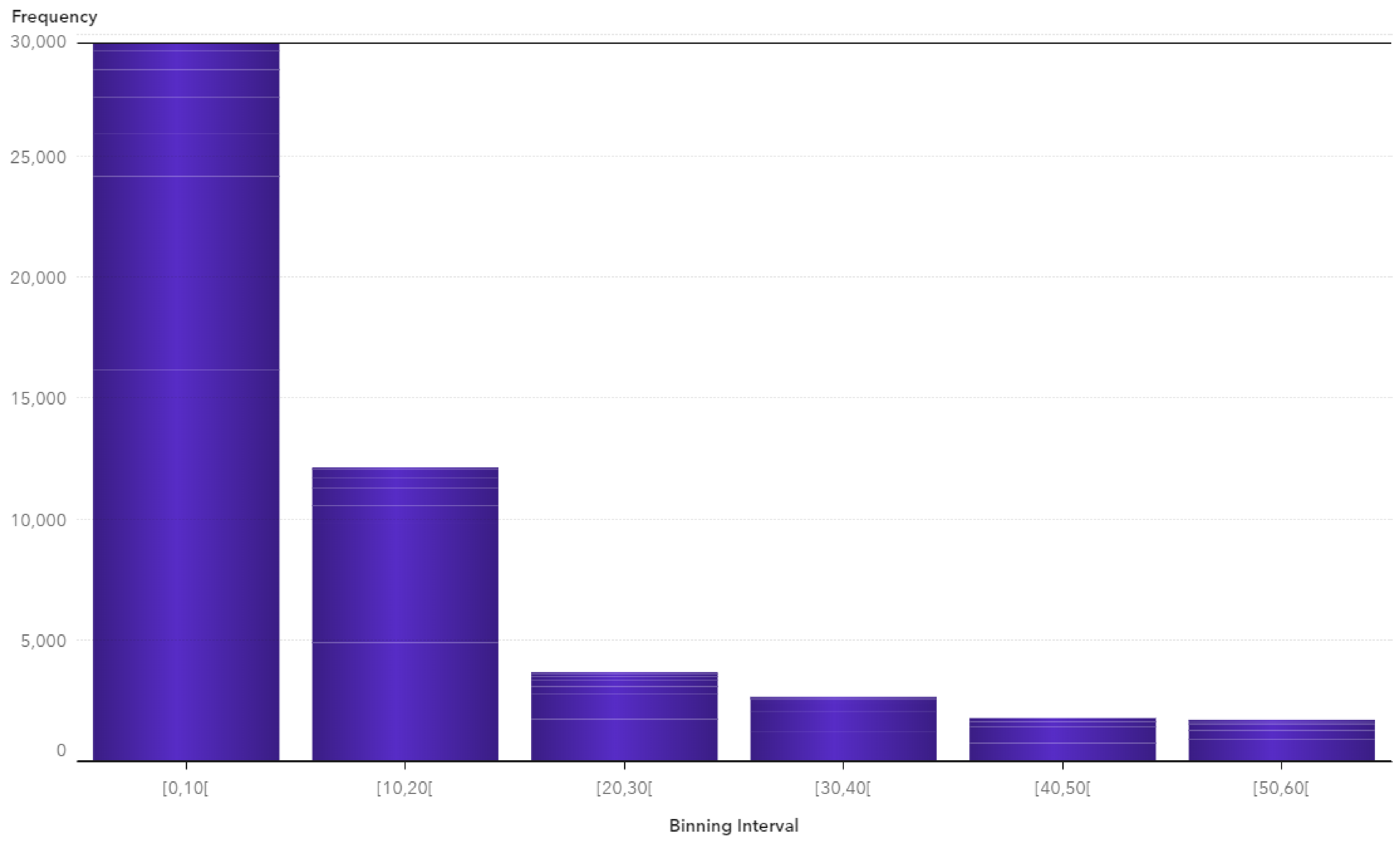


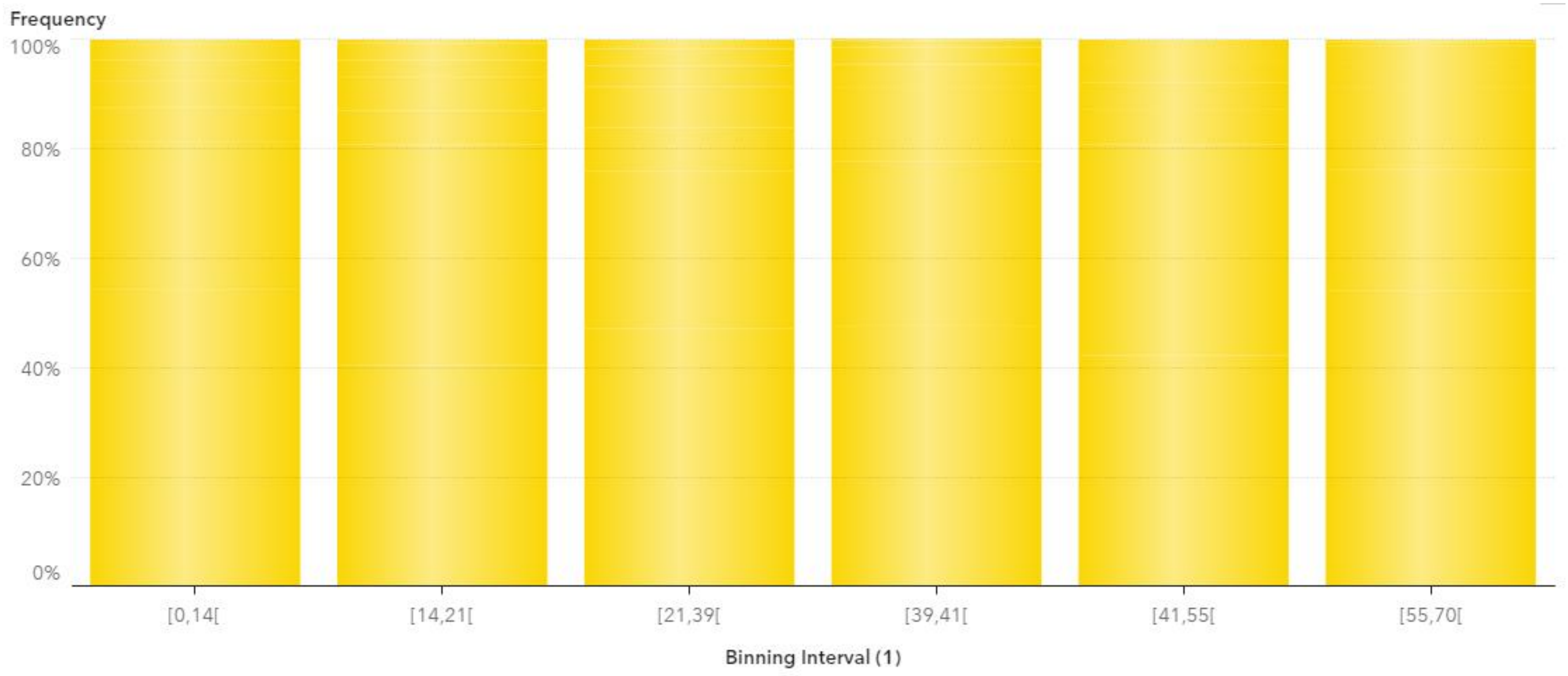
- Based on the binning results, **Weight of Evidence (WOE) and Information Value (IV)** also allow to **screen continuous, ordinal and categorical variables** based on their **predictive power**.

Unsupervised binning

Unsupervised binning methods **transform numerical** variables **into categorical** counterparts but do **not use the target** (class) information.

Equal Width and *Equal Frequency* are two unsupervised binning methods.





Unsupervised binning

Equal Width Binning

The algorithm divides the data into k intervals of equal size. The width of intervals is:

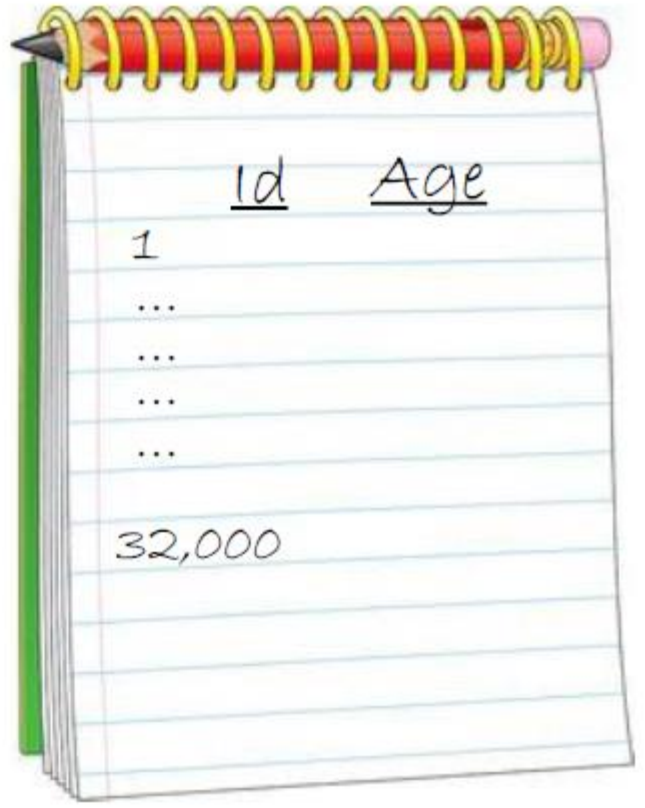
w = (max-min)/k

And the interval boundaries are:

min+w, min+2w, ... , min+(k-1)w

Equal Frequency Binning

The algorithm divides the data into k groups which each group contains the same number of values.

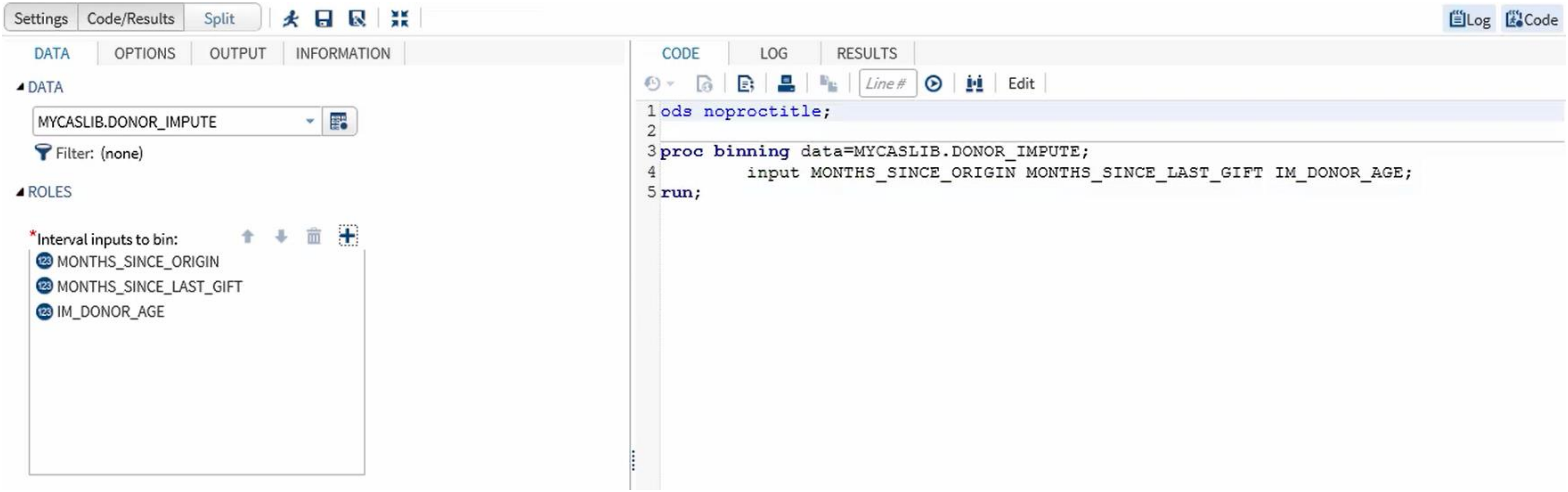


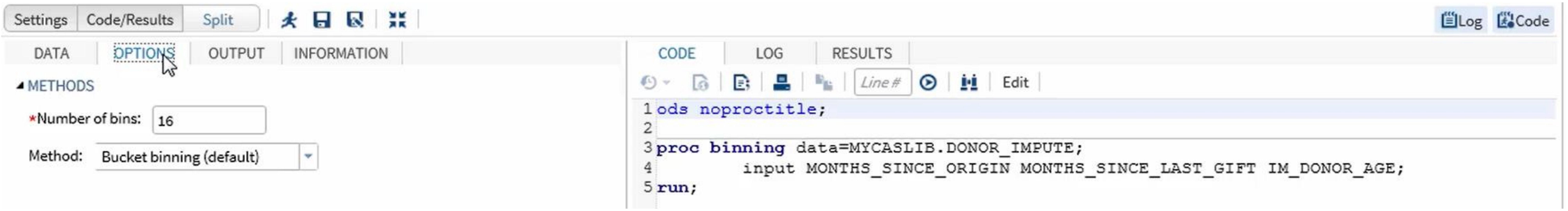
Mapping

Variable	Binned Variable	Range	Frequency	Proportion
Age	BIN_Age	Age < 31.6	3372	0.13015787
		31.6 <= Age < 47.2	9412	0.36329949
		47.2 <= Age < 62.8	9117	0.35191261
		62.8 <= Age < 78.4	3525	0.13606361
		78.4 <= Age	481	0.01856641

Binning Calculation

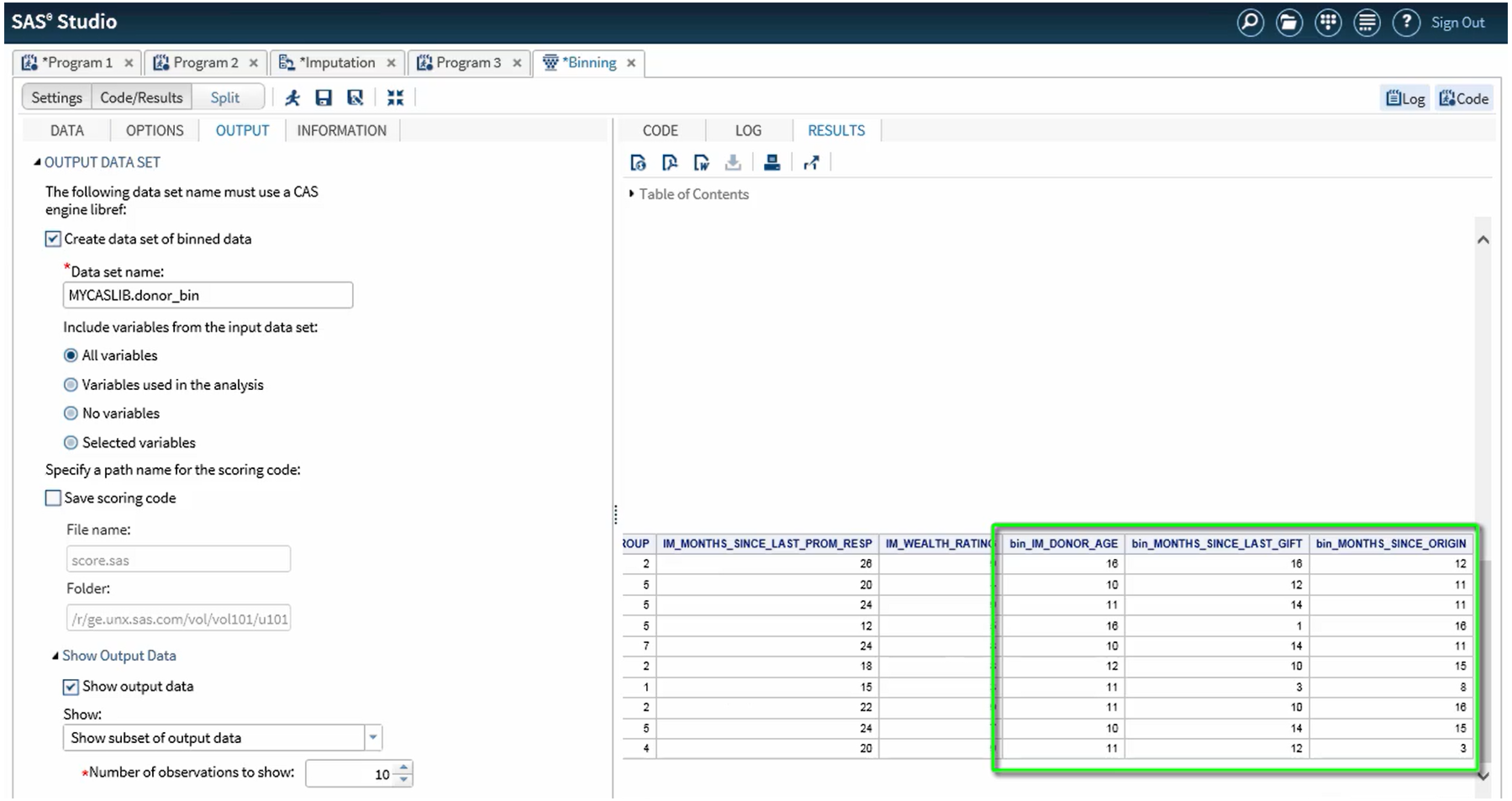
Code generator





Binning Calculation

Results



Variable Selection & Reduction via WOE & IV

Too many predictors & wide variability in values can result in significantly more chaotic information for models

Supervised Binning

Supervised binning methods transform **numerical variables into categorical** counterparts and **refer to the target** (class) information when selecting discretization cut points. WOE-based binning is an example of a supervised binning.

Information Value	Variable Predictiveness
< 0.02	Not useful for prediction
0.02 to 0.1	Weak
0.1 to 0.3	Medium
0.3 to 0.5	Strong
>0.5	Suspiciously good

Age	Income	Ins

Age
of customer in years

Income
in thousands of dollars

Ins
Did customer buy insurance product?
(Binary target variable)
1 = Yes 0 = No

WOE & IV Calculation

Results

*Program 1 | Program 2 | *Imputation | Program 3 | *Binning | Program 4

CODE | LOG | RESULTS

```
ods noproctitle;

proc binning data=MYCASLIB.DONOR_IMPUTE WOE;
        input MONTHS_SINCE_ORIGIN IM_DONOR_AGE / numbin=16;
        input MONTHS_SINCE_LAST_GIFT/ numbin=17;
        output out=MYCASLIB.donor_bin copyvars=(_all_);
        target TARGET_B / event='1';
run;

proc print data=MYCASLIB.donor_bin(obs=10);
        title "Subset of MYCASLIB.donor_bin";
run;
```

Variable Information Value	
Variable	Information Value
MONTHS_SINCE_ORIGIN	0.0360
IM_DONOR_AGE	0.0157
MONTHS_SINCE_LAST_GIFT	0.0500

Bin Details												
Variable	Bin ID	Lower Bound	Upper Bound	Bin Width	Number of Observations	Mean	Standard Deviation	Minimum	Maximum	Event Count	Weight of Evidence	Information Value
MONTHS_SINCE_ORIGIN	Missing				0					0		
	1	-Infty	13.250		7	5	0	5	5	6	-2.890	0.0034
	2	13.250	21.500	8.2500	1840	17.003	0.1042	17	21	313	0.4862	0.0197
	3	21.500	29.750	8.2500	3136	29	0	29	29	733	0.0887	0.0012
	4	29.750	38	8.2500	1	32	.	32	32	1	-2.197	0.0005
	5	38	46.250	8.2500	2423	41.000	0.0203	40	41	579	0.0598	0.0004
	6	46.250	54.500	8.2500	812	53.001	0.0351	53	54	216	-0.084	0.0003
	7	54.500	62.750	8.2500	1	57	.	57	57	1	-2.197	0.0005
	8	62.750	71	8.2500	1519	65.001	0.0513	65	67	361	0.0670	0.0003
	9	71	79.250	8.2500	1698	77	0	77	77	426	-0.005	194E-8
	10	79.250	87.500	8.2500	0					0	.	.
	11	87.500	95.750	8.2500	1612	89.002	0.0747	89	92	443	-0.128	0.0014
	12	95.750	104	8.2500	1087	101	0	101	101	283	-0.054	0.0002
	13	104	112.25	8.2500	0					0	.	.
	14	112.25	120.50	8.2500	1425	113.01	0.1675	113	119	391	-0.126	0.0012
	15	120.50	128.75	8.2500	708	125.00	0.1407	122	127	201	-0.173	0.0011
	16	128.75	Infty		3103	136.99	0.1982	129	137	889	-0.186	0.0058
IM_DONOR_AGE	Missing				0					0		
	1	-Infty	5.4375		6	1.6667	1.5055	0	4	1	0.5108	0.0001
	2	5.4375	10.875	5.4375	84	6.8810	0.3258	6	7	12	0.6931	0.0017
	3	10.875	16.313	5.4375	15	15.467	1.1255	12	16	6	-0.693	0.0004
	4	16.313	21.750	5.4375	122	17.549	1.2927	17	21	22	0.4155	0.0010
	5	21.750	27.188	5.4375	247	25.810	1.4980	22	27	49	0.2978	0.0010
	6	27.188	32.625	5.4375	347	30.254	1.2899	28	32	70	0.2769	0.0013
	7	32.625	38.063	5.4375	930	35.713	1.6101	33	38	194	0.2348	0.0025
	8	38.063	43.500	5.4375	1122	41.078	1.4694	39	43	272	0.0408	0.0001
	9	43.500	48.938	5.4375	1261	46.079	1.3277	44	48	312	0.0138	123E-7

SUMMARY PROC BINNING IN VIYA
Conclusion

"More data beats clever algorithms,
but **better data** beats more data."
– Peter Norvig

SUMMARY PROC BINNING IN VIYA
Conclusion

In recent years, **WOE and IV** have been receiving increasing attention from **various sectors** beyond scorecard development for credit risk.

Extremely useful in **reducing variables** and allowing **to boost the performances of interpretable analytical models** that are more likely to be **consumed and adopted** by the entire organization.

Powerful binning tool which saves the guesswork and adds options

Customized binning levels across variables

Weight of Evidence

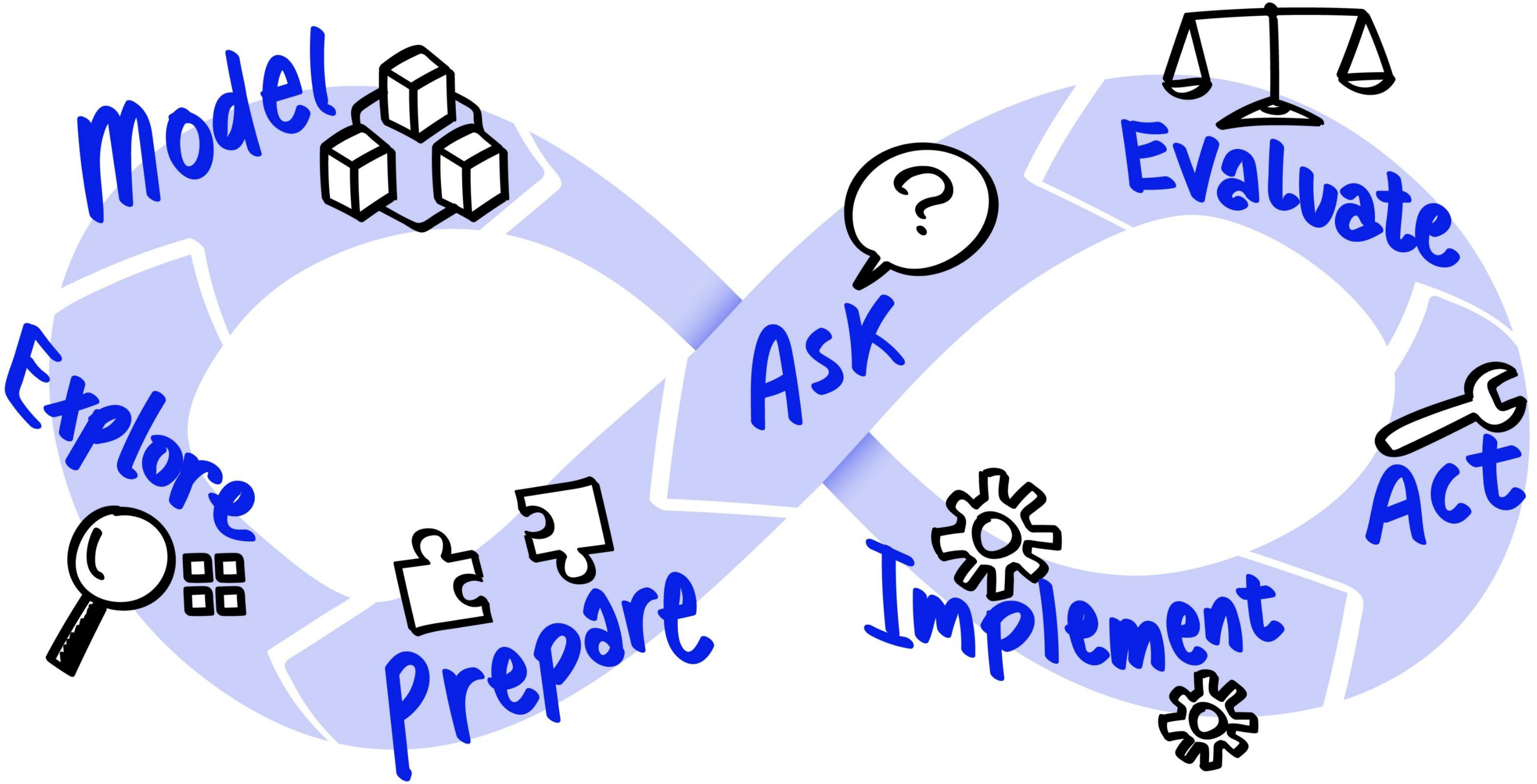
Information Values

Data Science Jam Sessions by SAS

Talk of the GEEKS

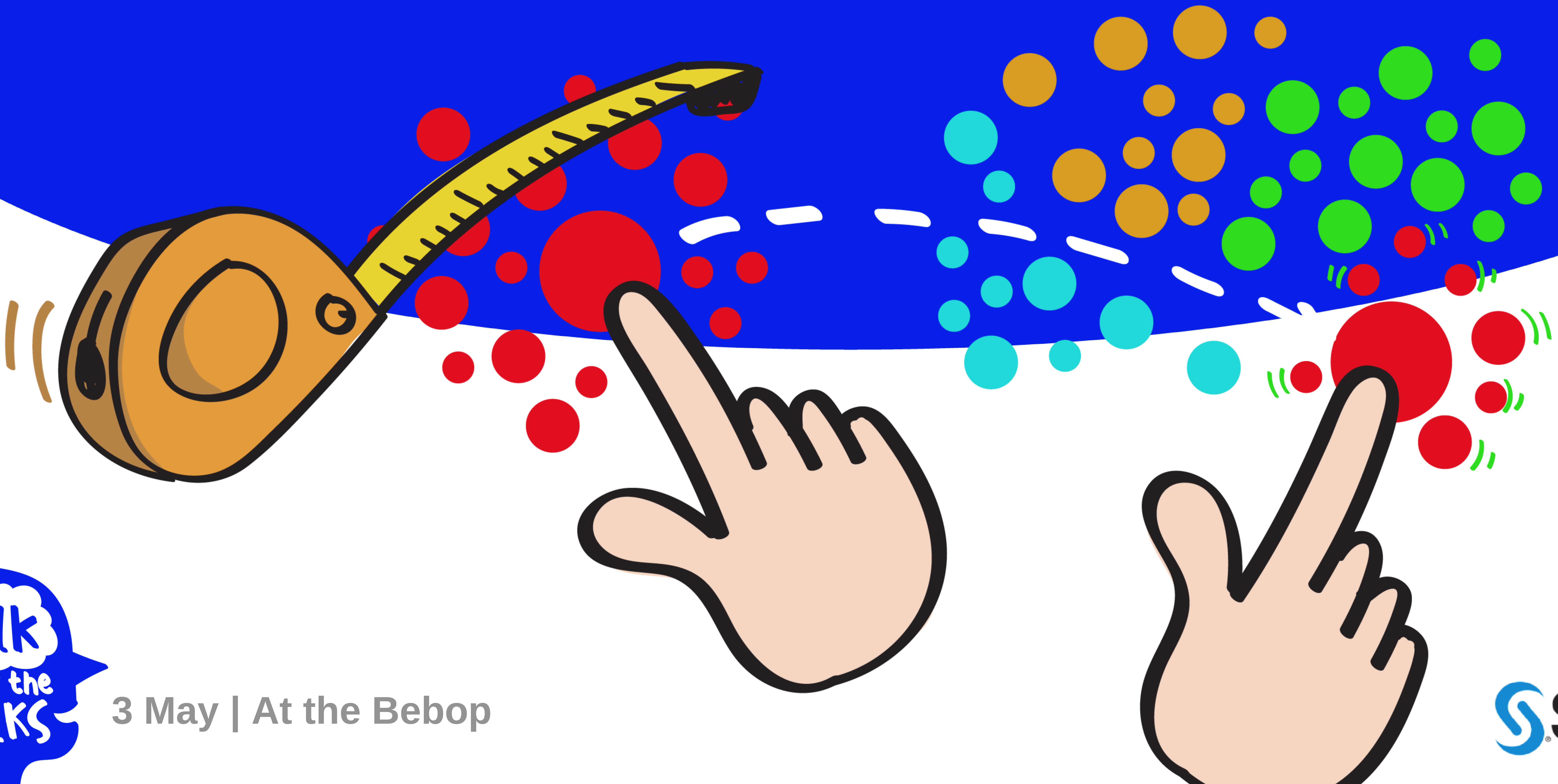
sas THE POWER TO KNOW®

Analytical Lifecycle



Clustering: same same and different

Speaker: Joline Jammaers



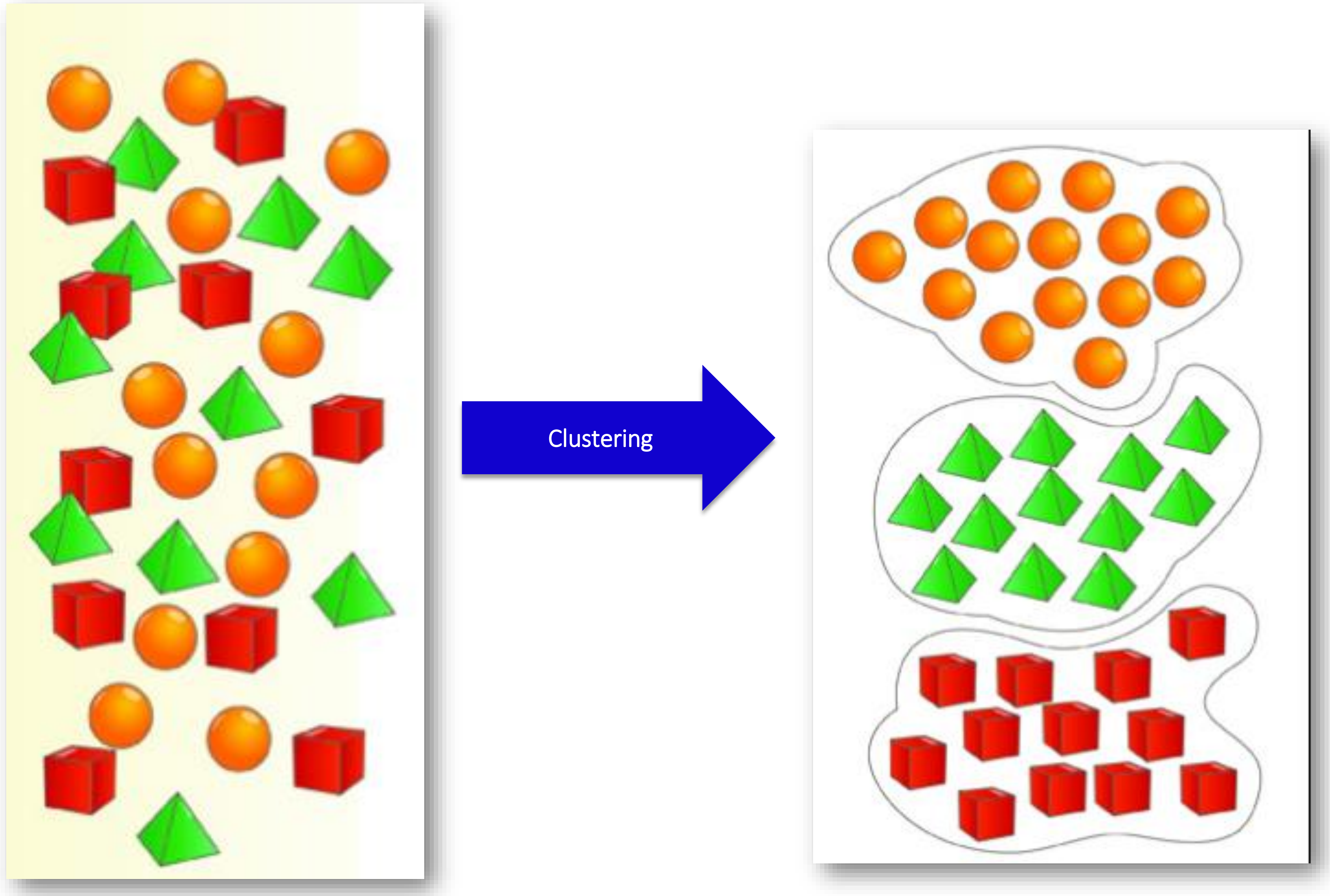
Talk of the GEEKS

3 May | At the Bebop

sas THE POWER TO KNOW.

Clustering: same same and different

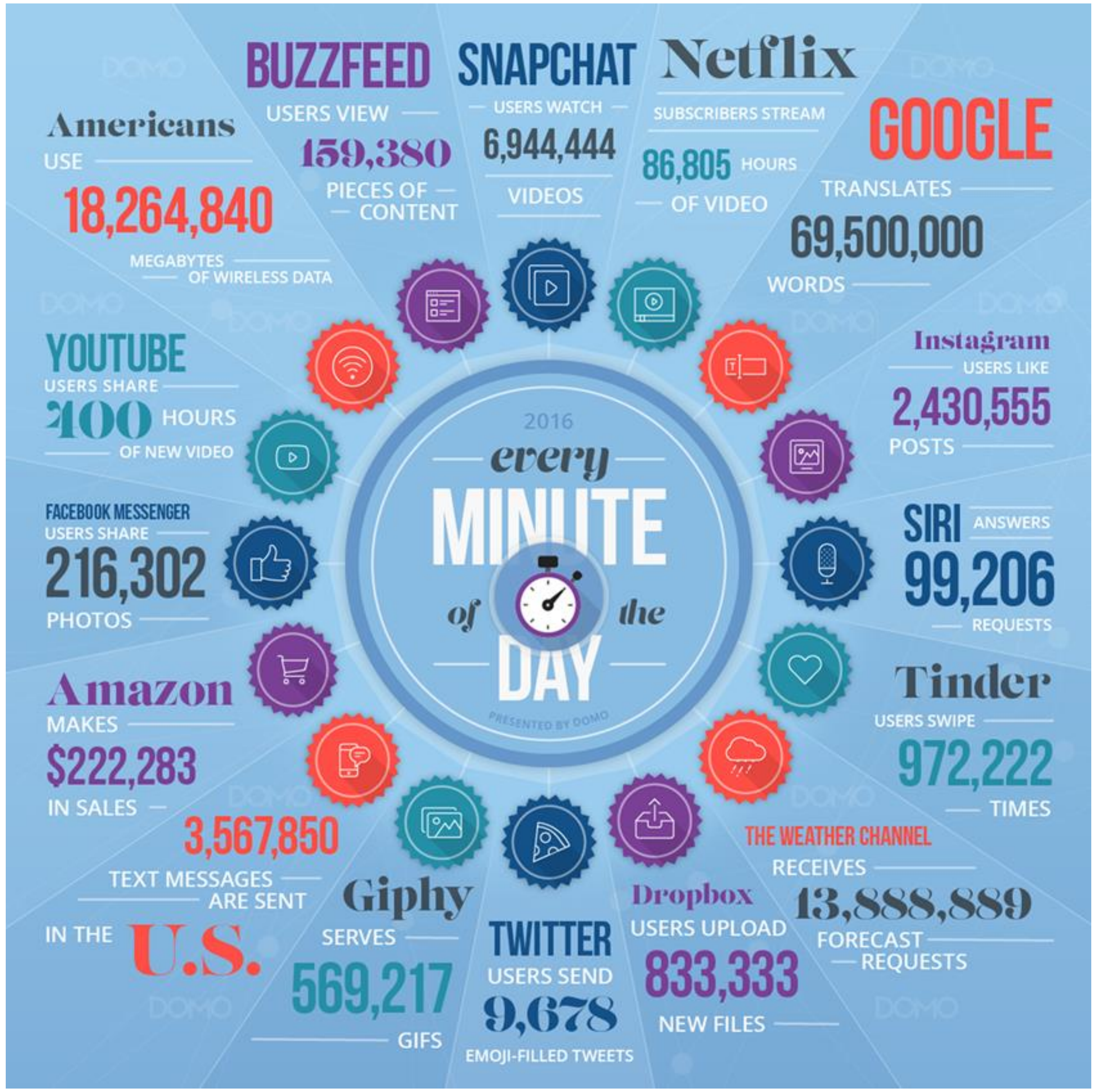
What?

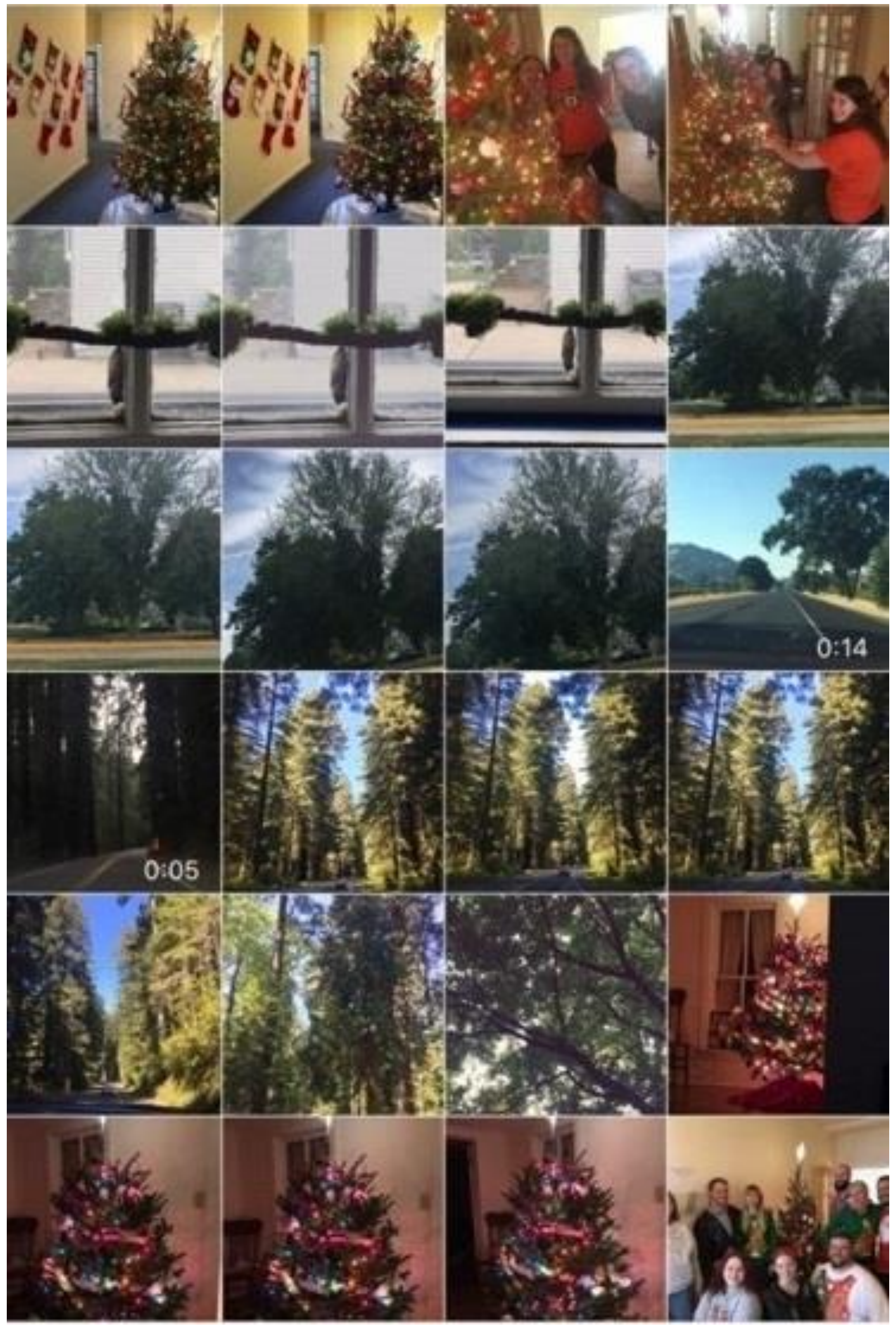


- Grouping of observations or variables
- Unsupervised learning technique
- Minimizing some metric of “distance” within the cluster and maximizing the distance between the clusters

Clustering: same same and different

Why?

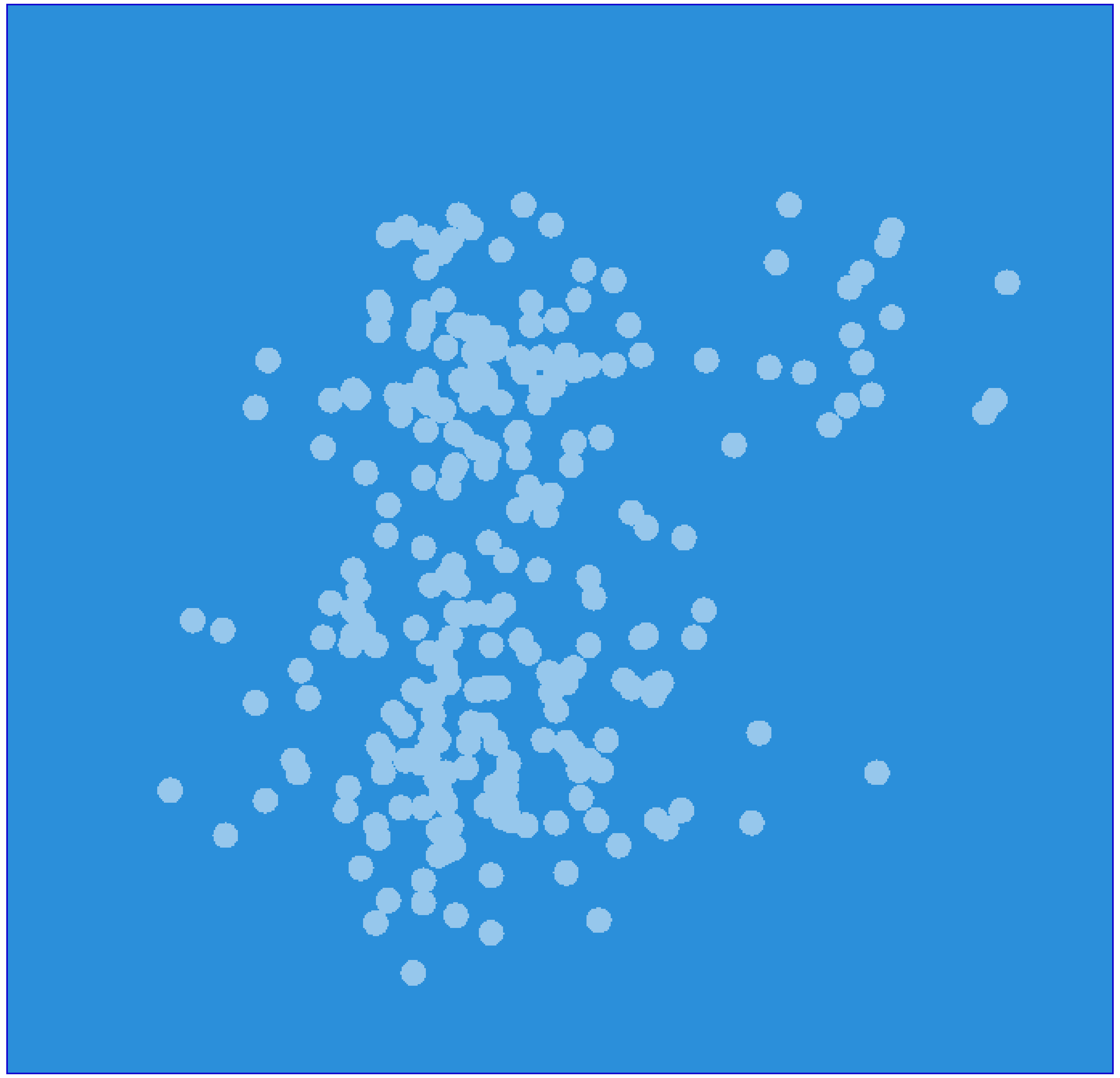




Clustering: same same and different

K-means Clustering

Training Data

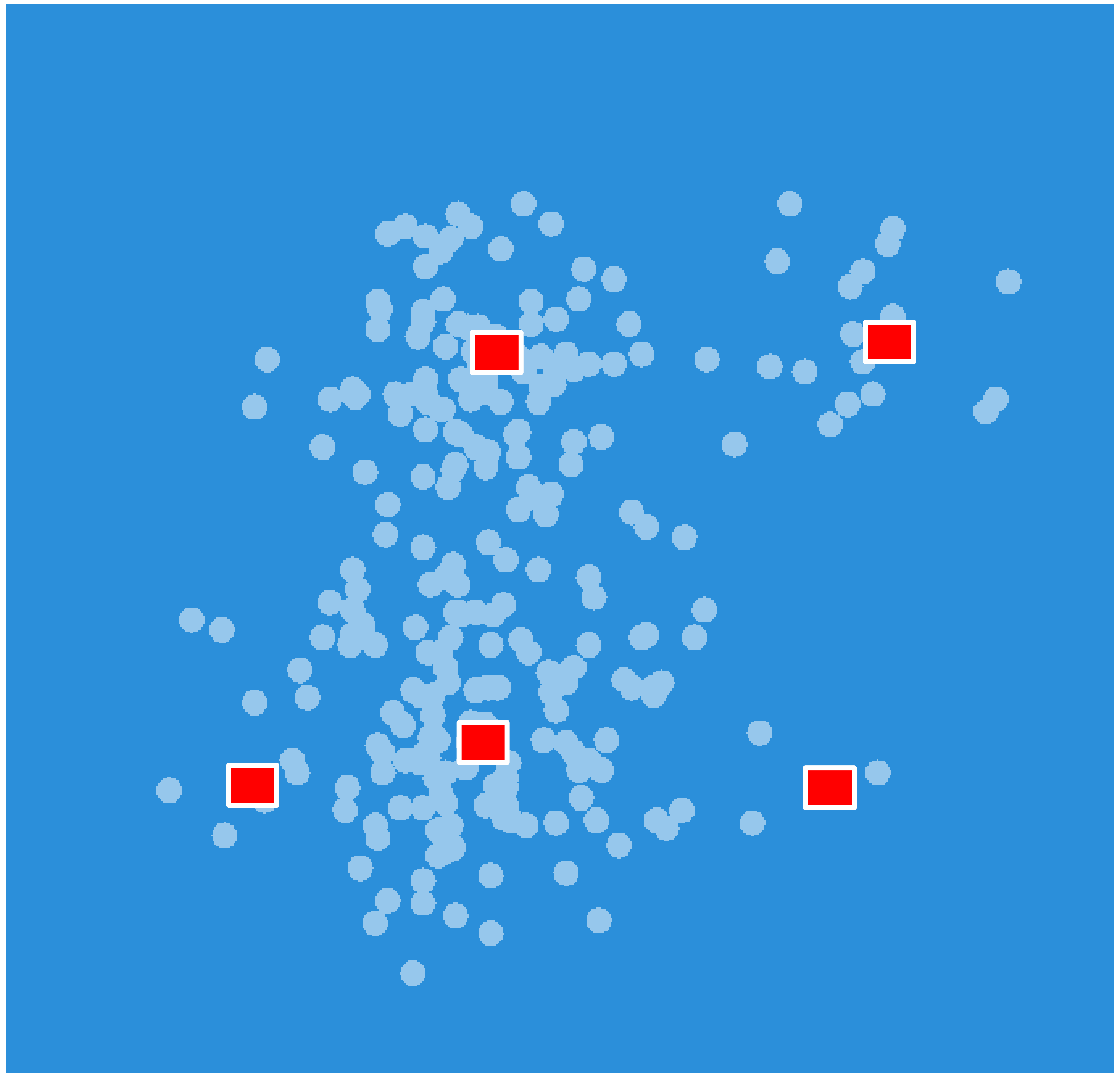


1. **Select inputs.**
2. **Select *k* cluster centers.**
3. **Assign cases to closest center.**
4. **Update cluster centers.**
5. **Re-assign cases.**
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

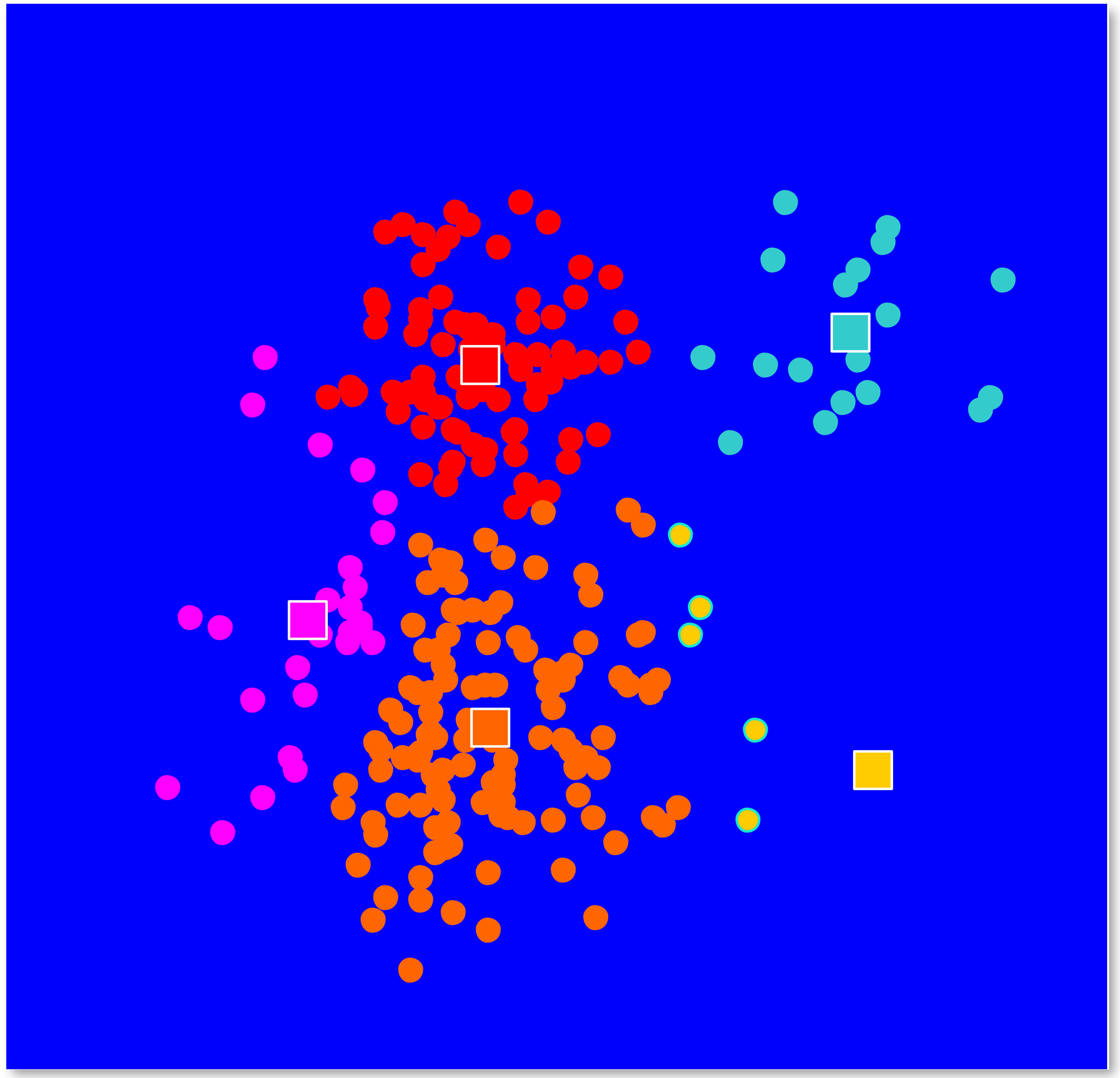


1. Select inputs.
2. **Select *k* cluster centers.**
3. Assign cases to closest center.
4. Update cluster centers.
5. Re-assign cases.
6. Repeat steps 4 and 5 until convergence.

Clustering: same same and different

K-means Clustering

Training Data

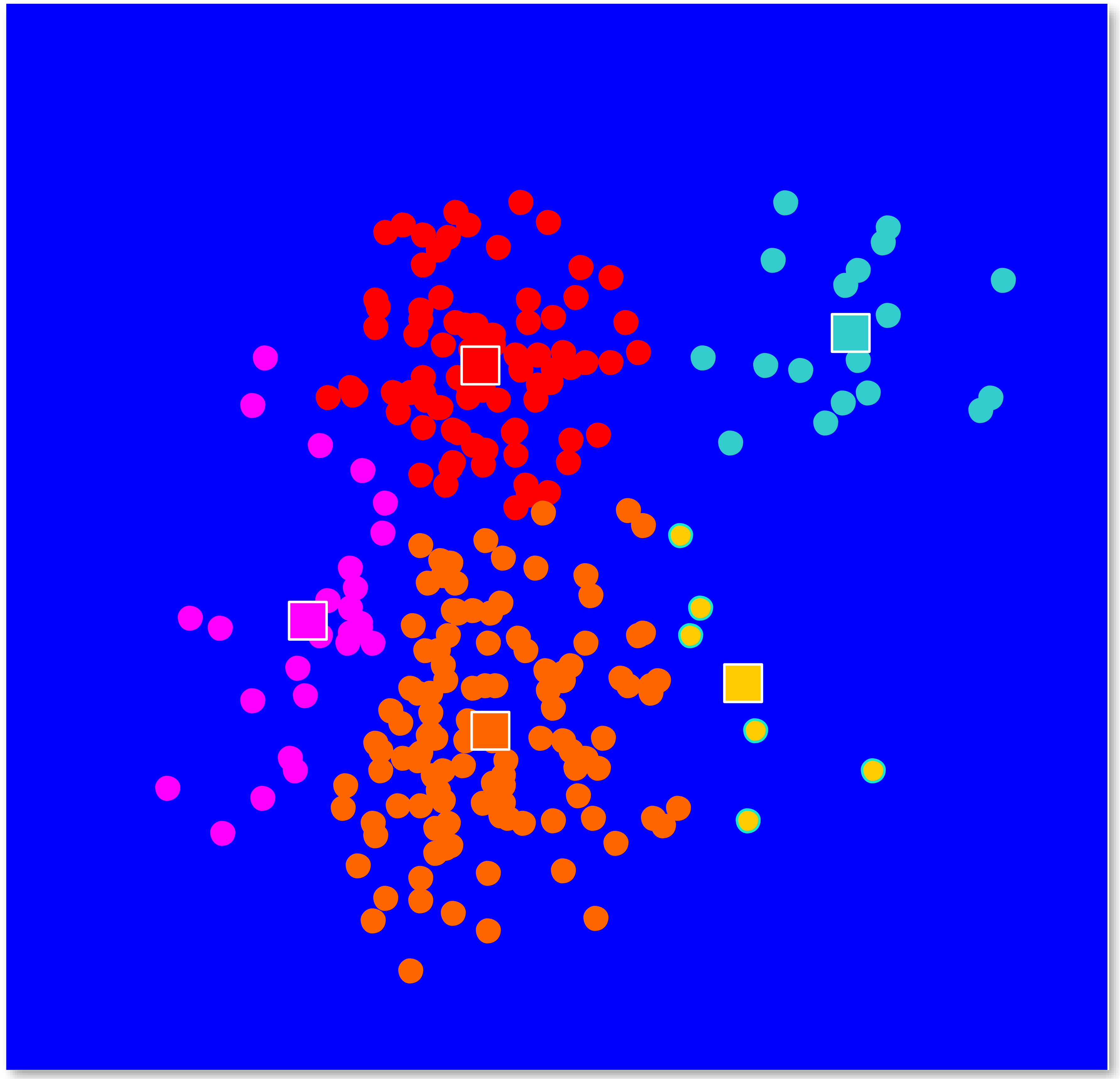


1. Select inputs.
2. Select *k* cluster centers.
3. **Assign cases to closest center.**
4. Update cluster centers.
5. Reassign cases.
6. Repeat steps 4 and 5 until convergence.

Clustering: same same and different

K-means Clustering

Training Data

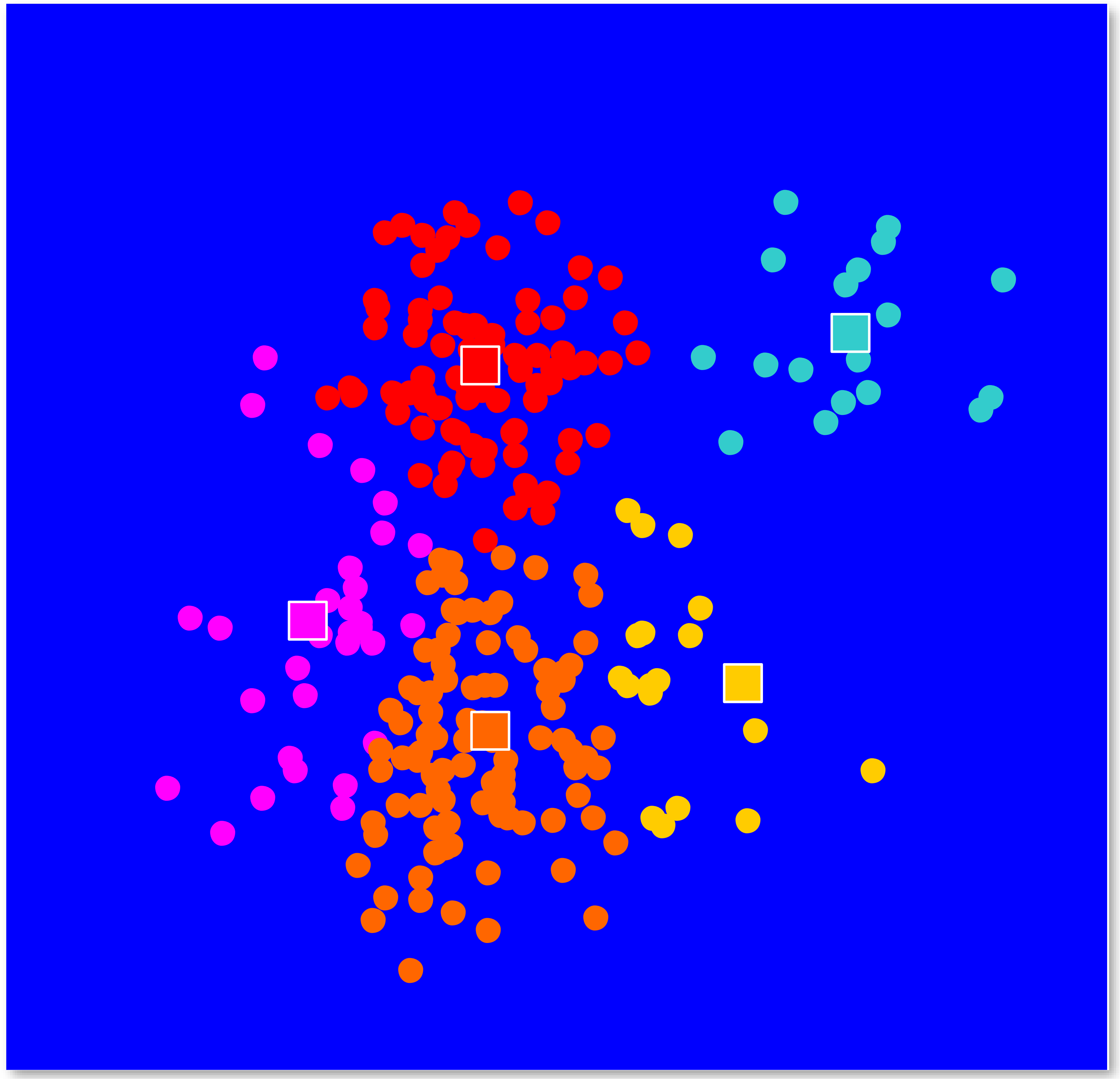


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. **Update cluster centers.**
5. Reassign cases.
6. Repeat steps 4 and 5 until convergence.

Clustering: same same and different

K-means Clustering

Training Data

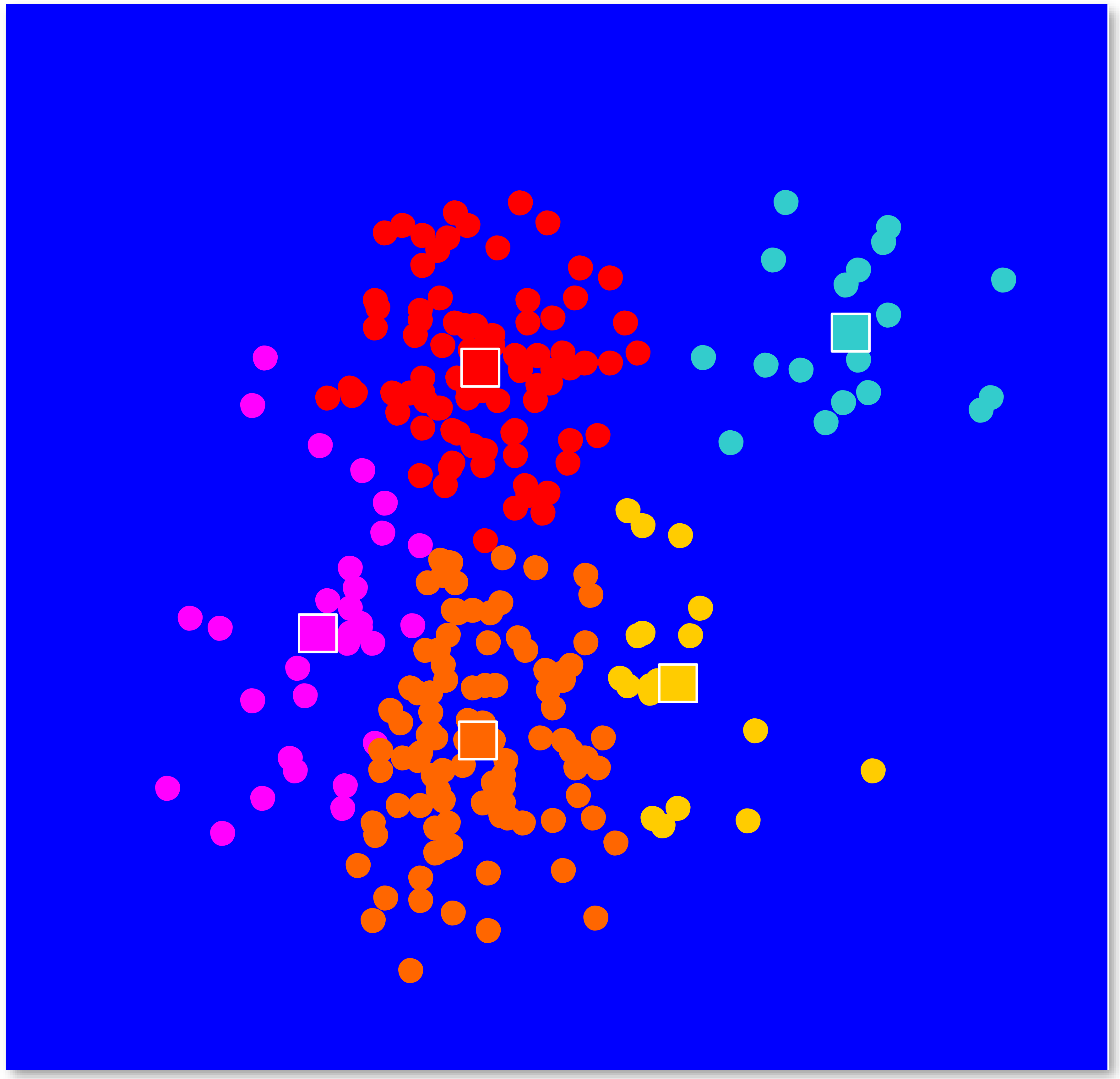


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. Update cluster centers.
5. **Reassign cases.**
6. Repeat steps 4 and 5 until convergence.

Clustering: same same and different

K-means Clustering

Training Data

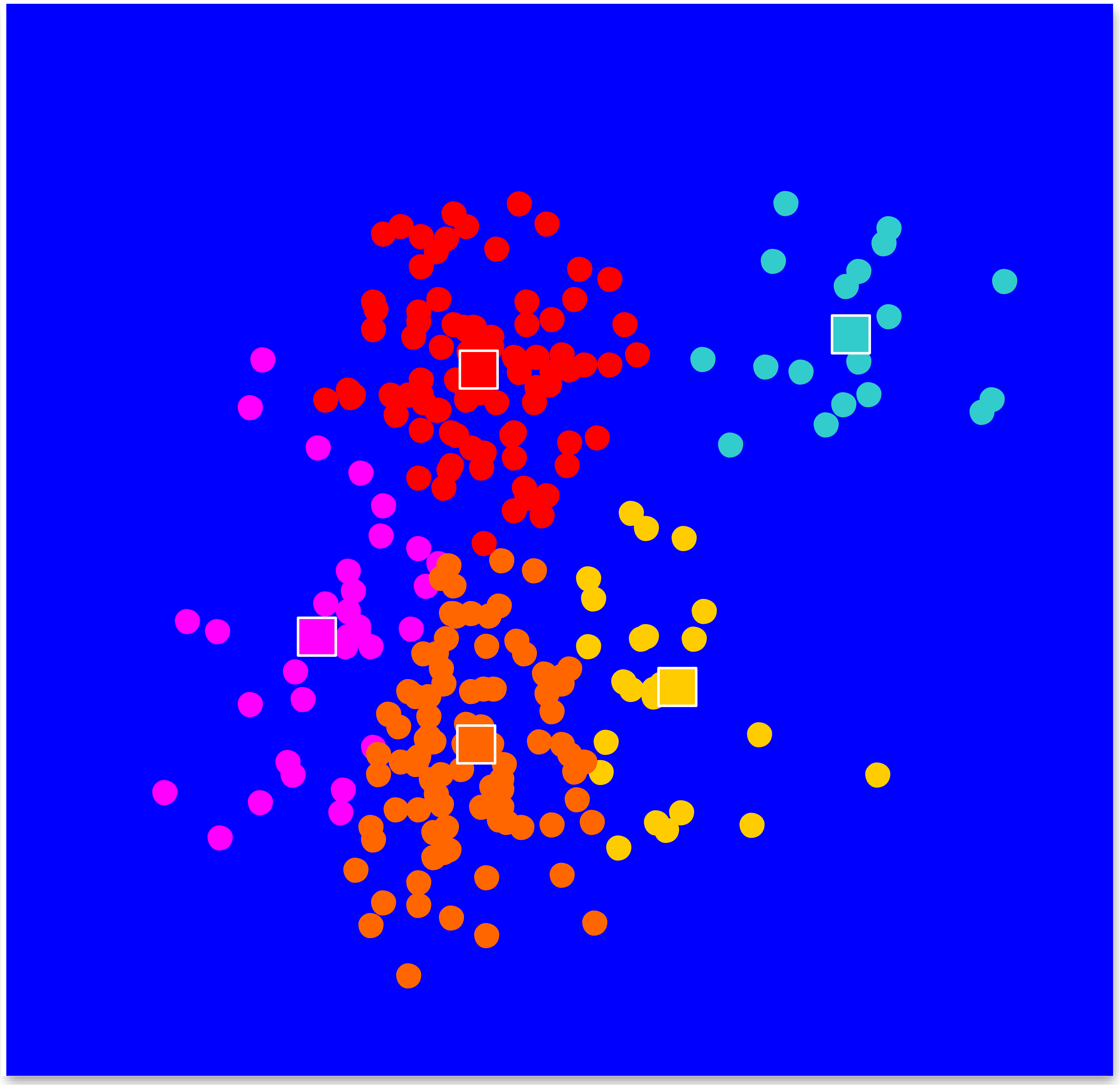


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. **Update cluster centers.**
5. Reassign cases.
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

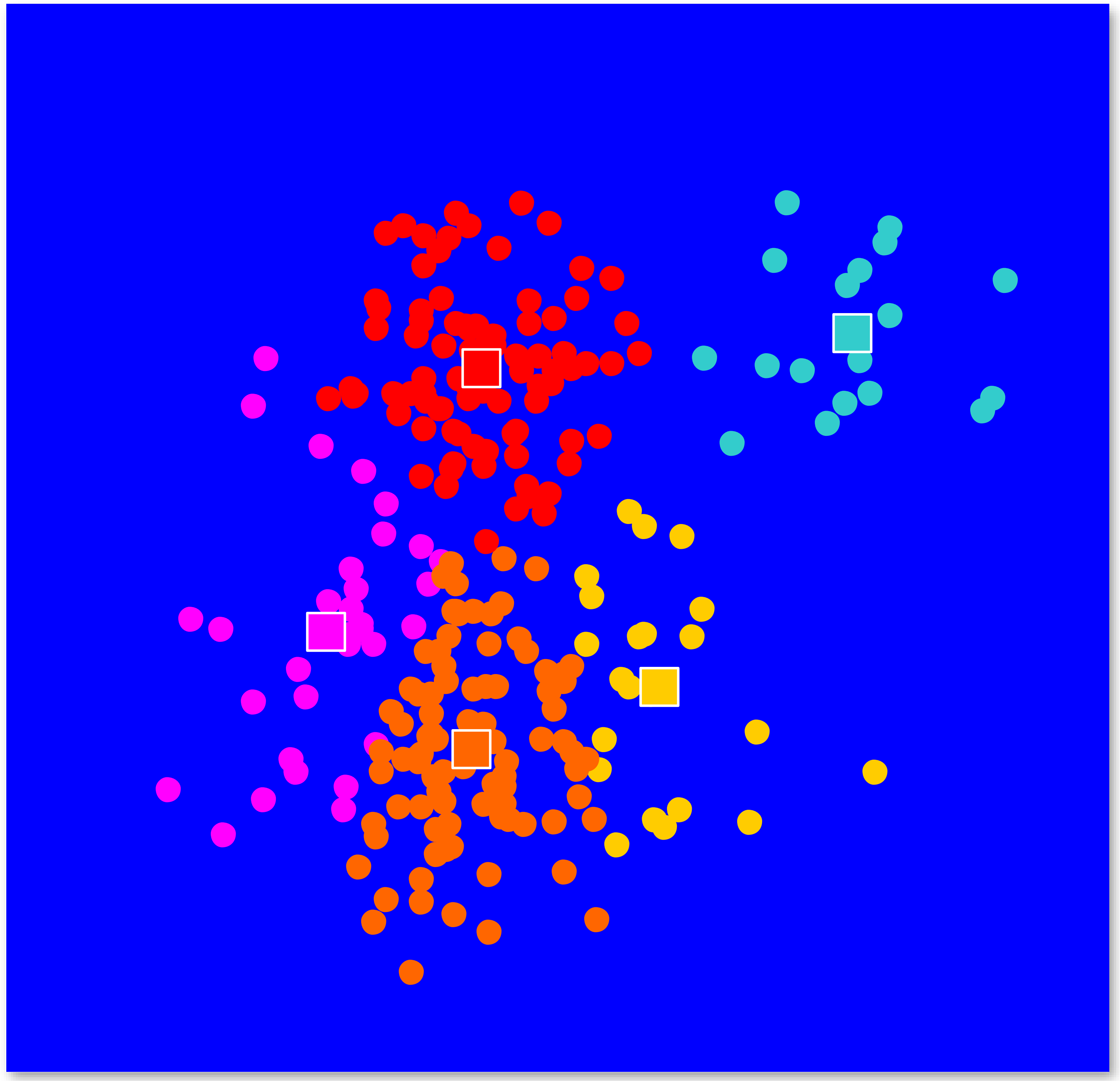


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. Update cluster centers.
5. **Reassign cases.**
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

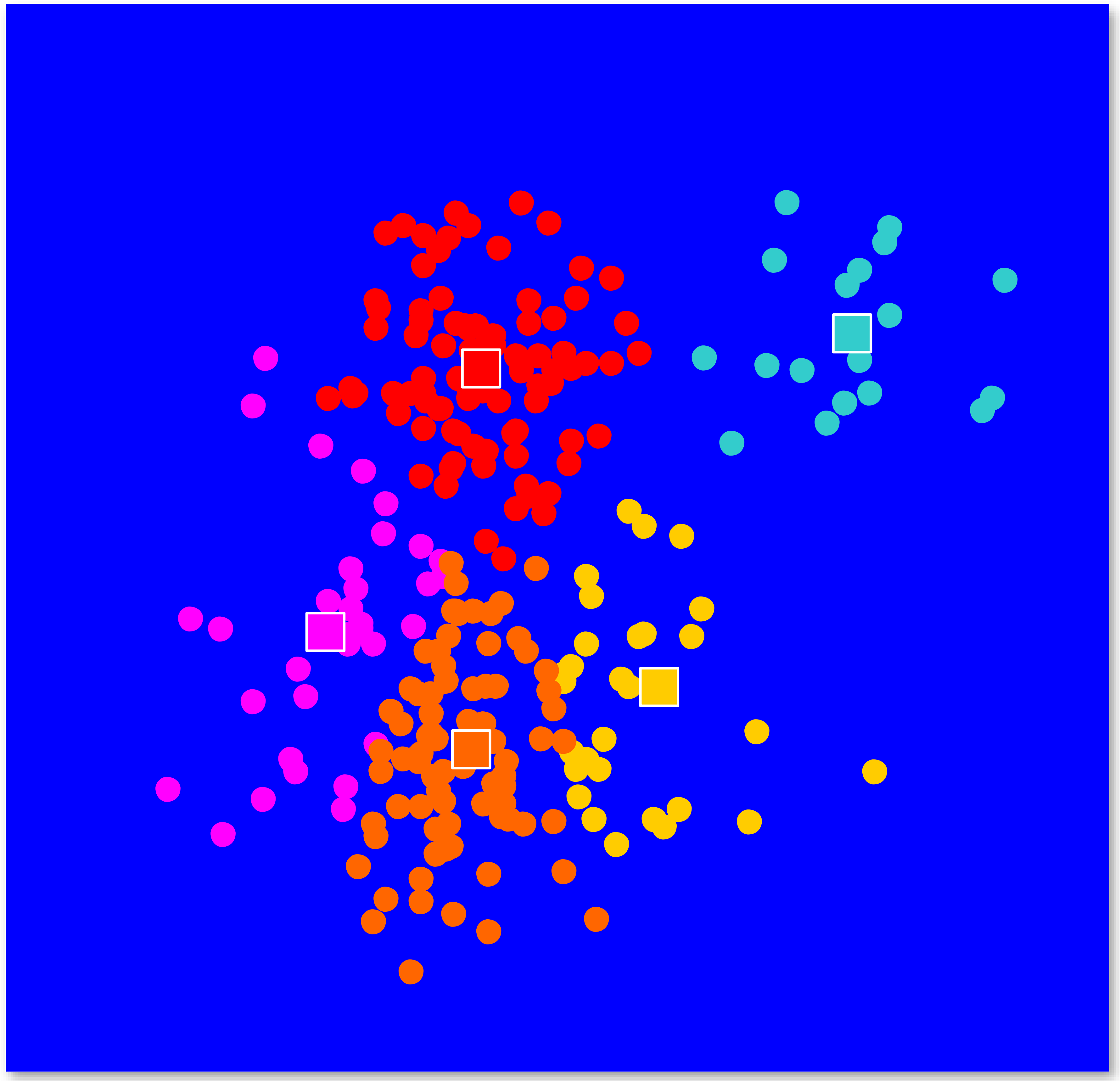


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. **Update cluster centers.**
5. Reassign cases.
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

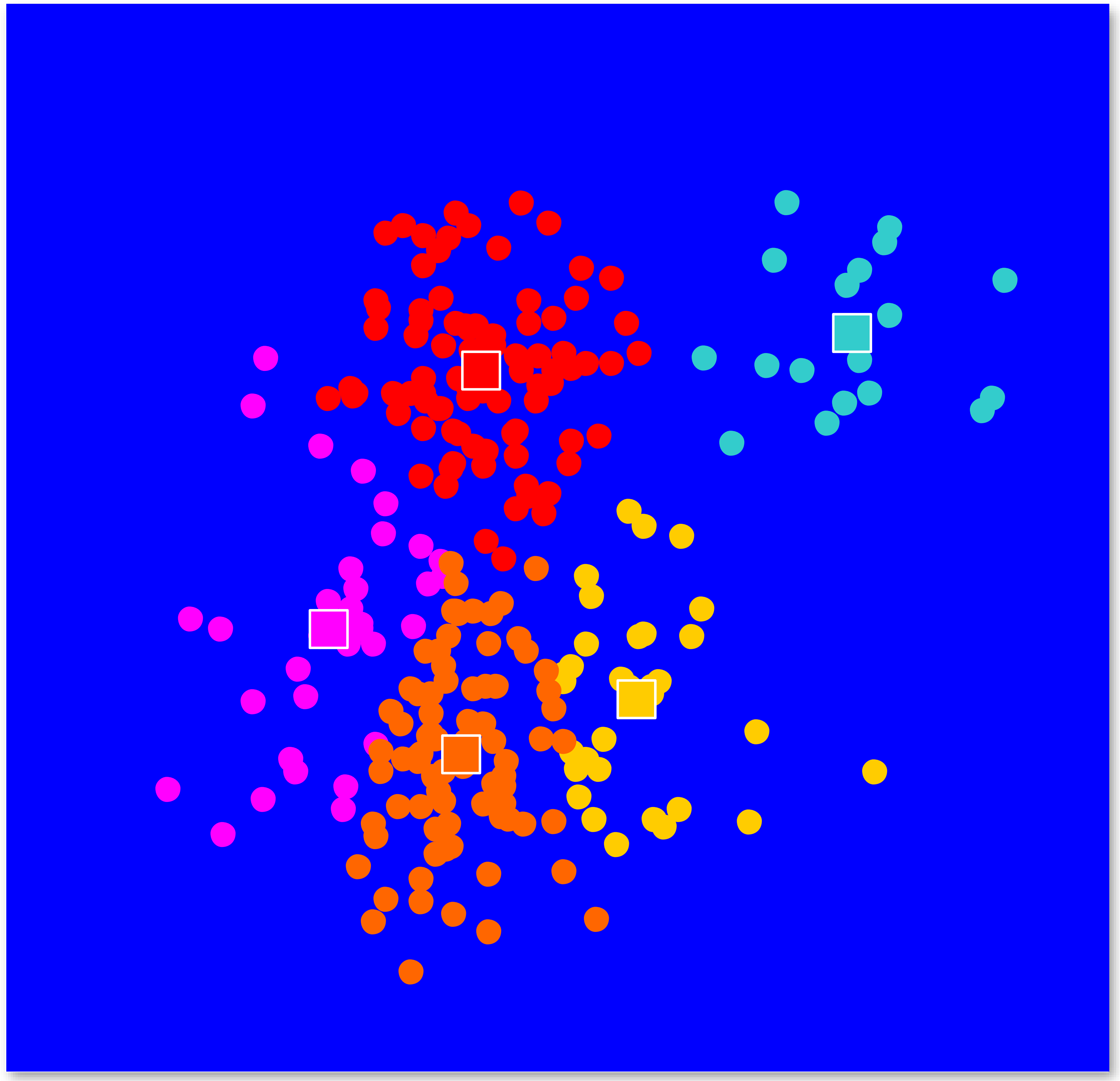


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. Update cluster centers.
5. **Reassign cases.**
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

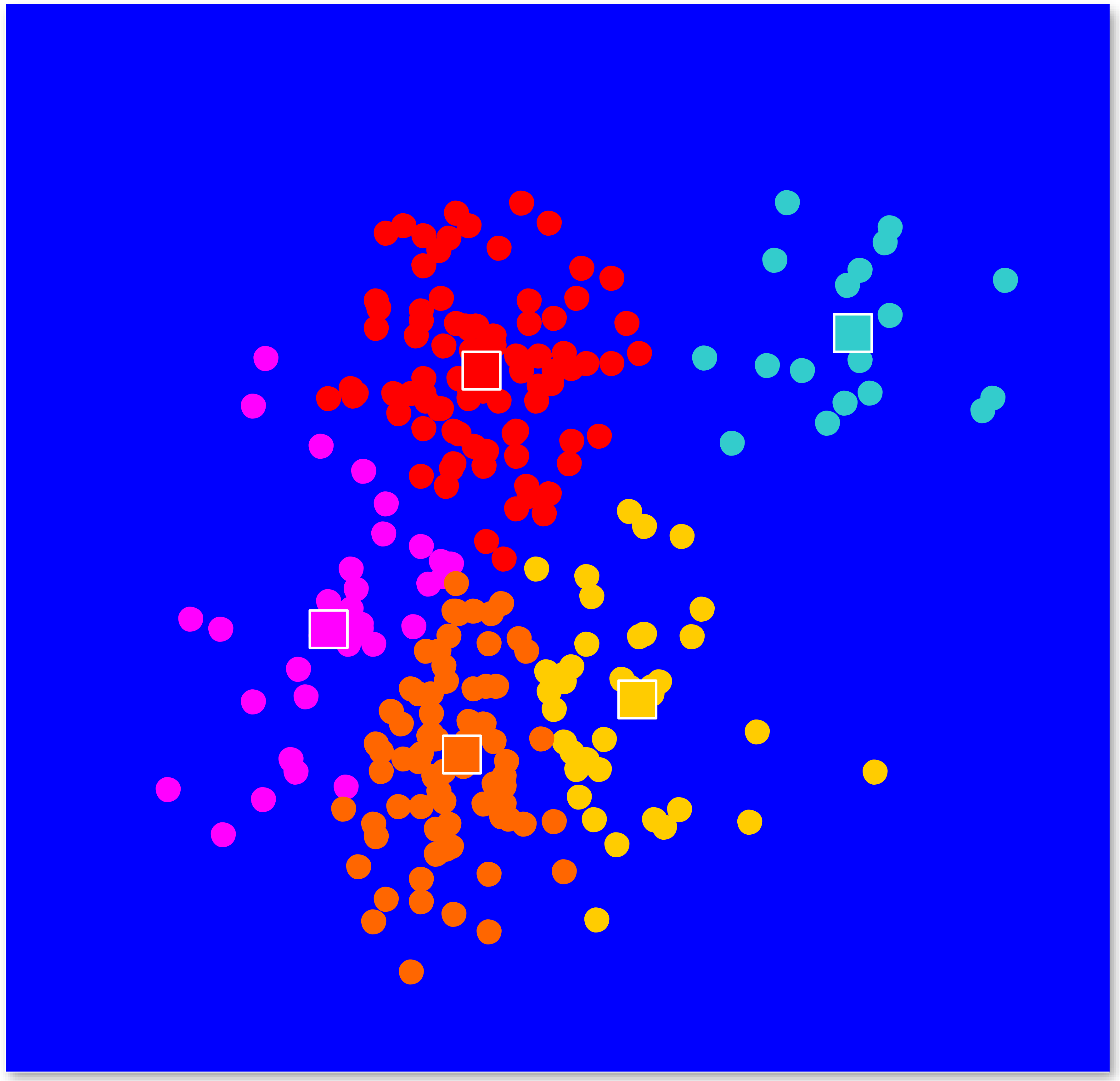


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. **Update cluster centers.**
5. Reassign cases.
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

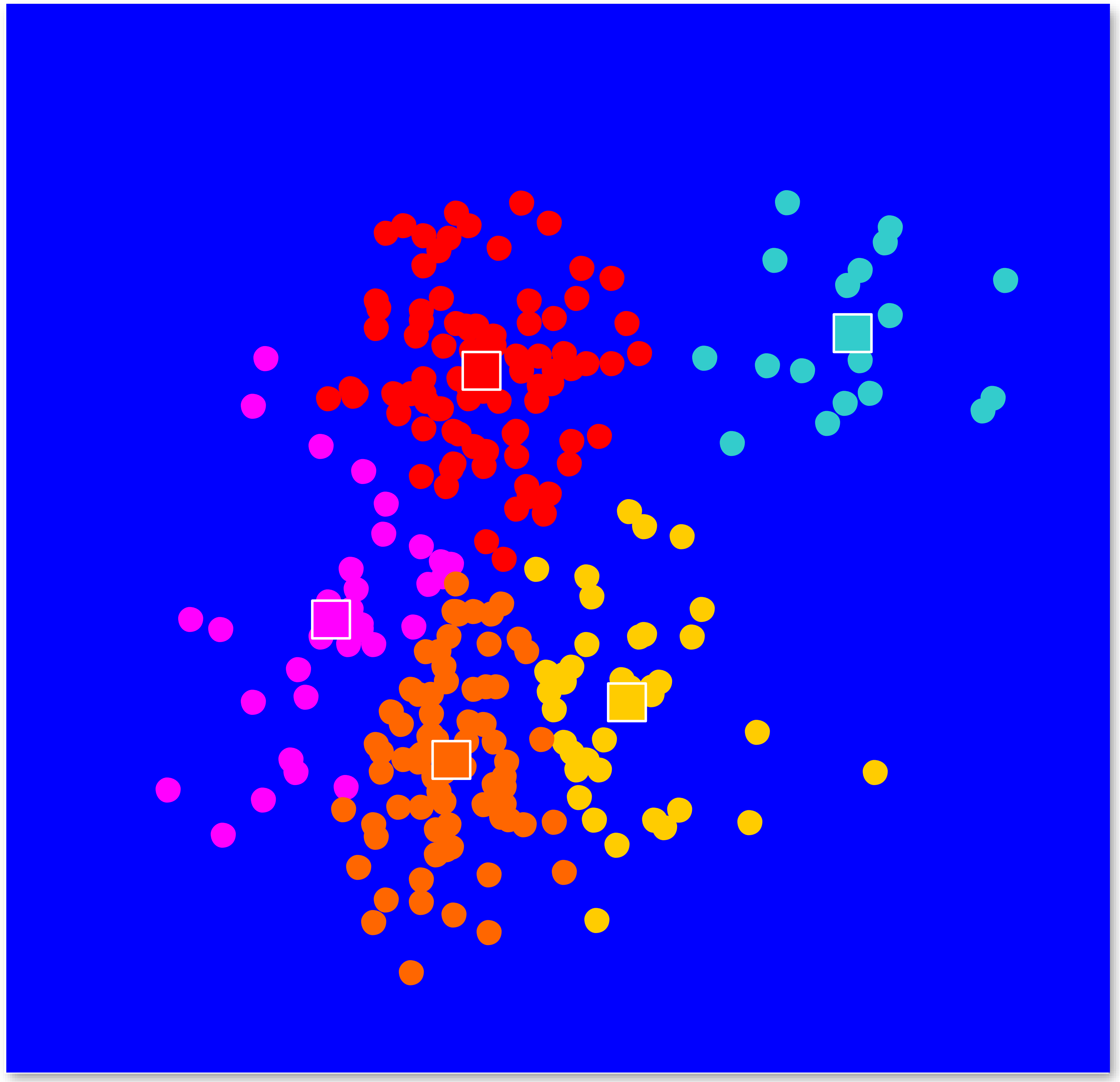


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. Update cluster centers.
5. **Reassign cases.**
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

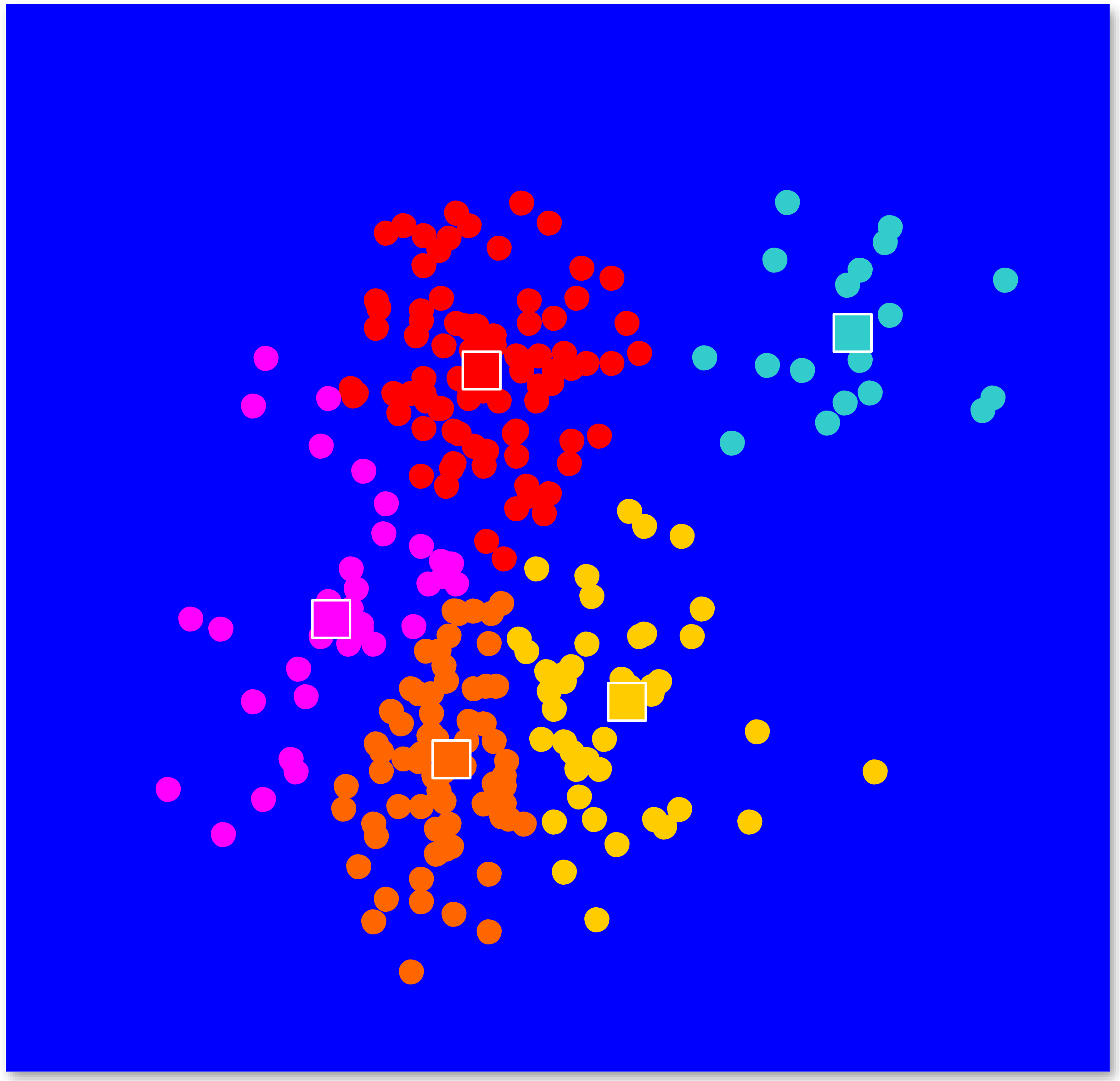


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. **Update cluster centers.**
5. Reassign cases.
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

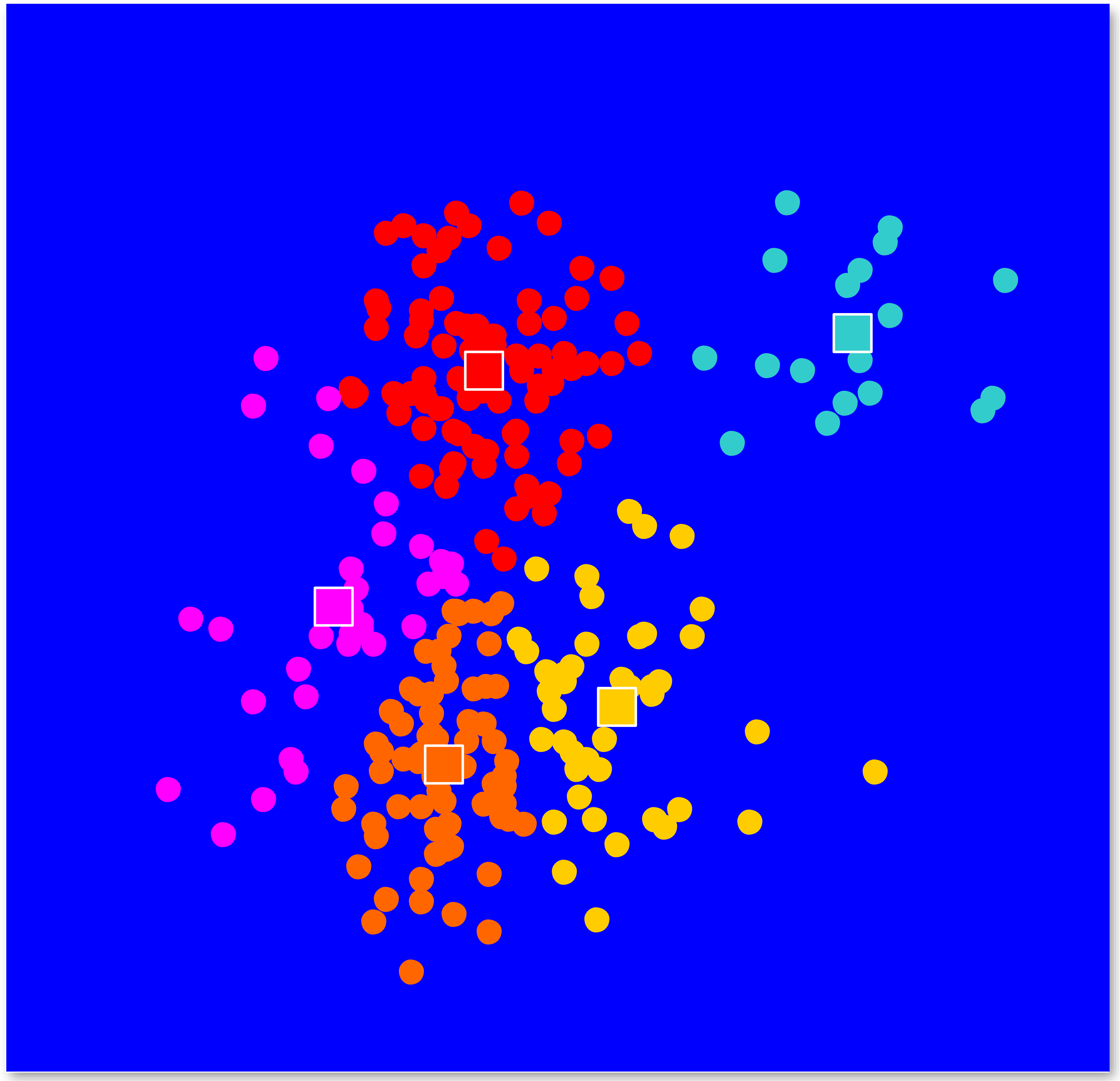


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. Update cluster centers.
5. **Reassign cases.**
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

K-means Clustering

Training Data

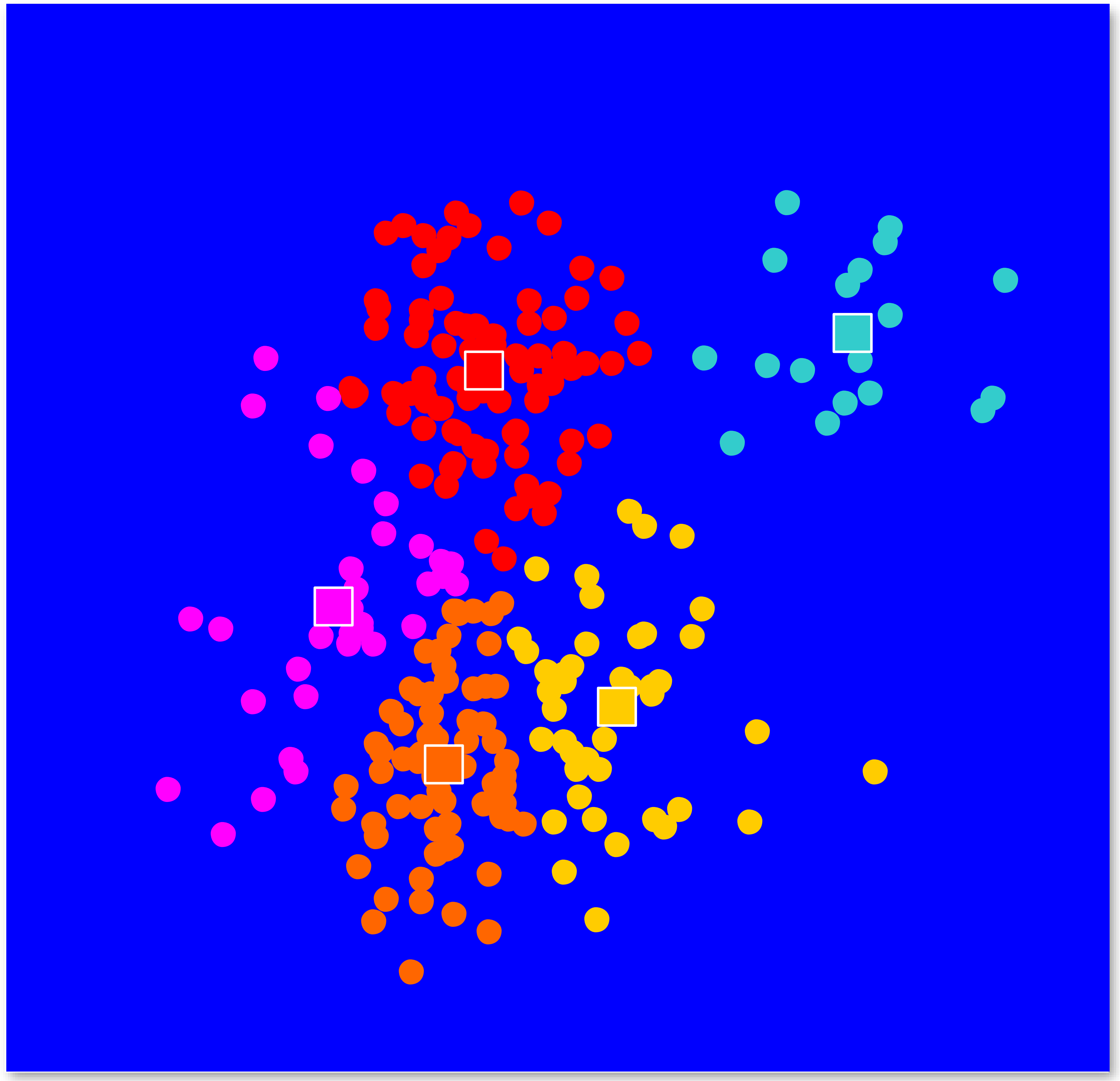


1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. Update cluster centers.
5. Reassign cases.
6. **Repeat steps 4 and 5 until convergence.**

Clustering: same same and different

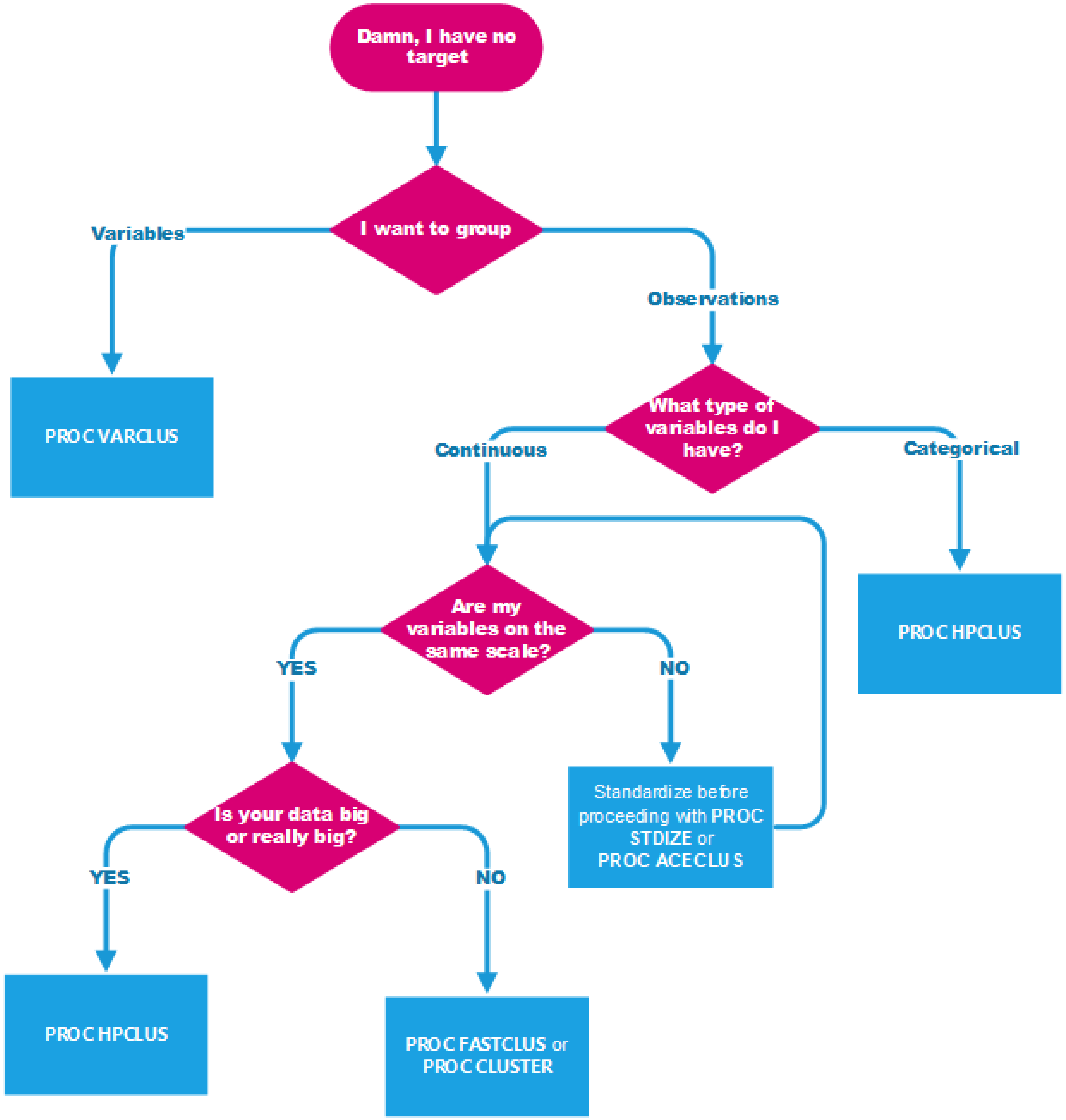
K-means Clustering

Training Data



1. Select inputs.
2. Select *k* cluster centers.
3. Assign cases to closest center.
4. Update cluster centers.
5. Reassign cases.
6. Repeat steps 4 and 5 until **convergence.**

Clustering: same same and different



Clustering: same same and different

Method

```
proc hpclus
     data= digits
     maxclusters= 8
     maxiter= 100
     seed= 54321
/* set seed for pseudo-random number generator */
     NOC= ABC(B= 1 minclusters= 3 align= PCA);
/* select best k between 3 and 8 using ABC */
     score out= OutScore;
     input pixel:;
/* input variables */
     ods output ABCStats= ABC;
/* save ABC criterion values for plotting */
run;
```

Clustering: same same and different

Conclusion

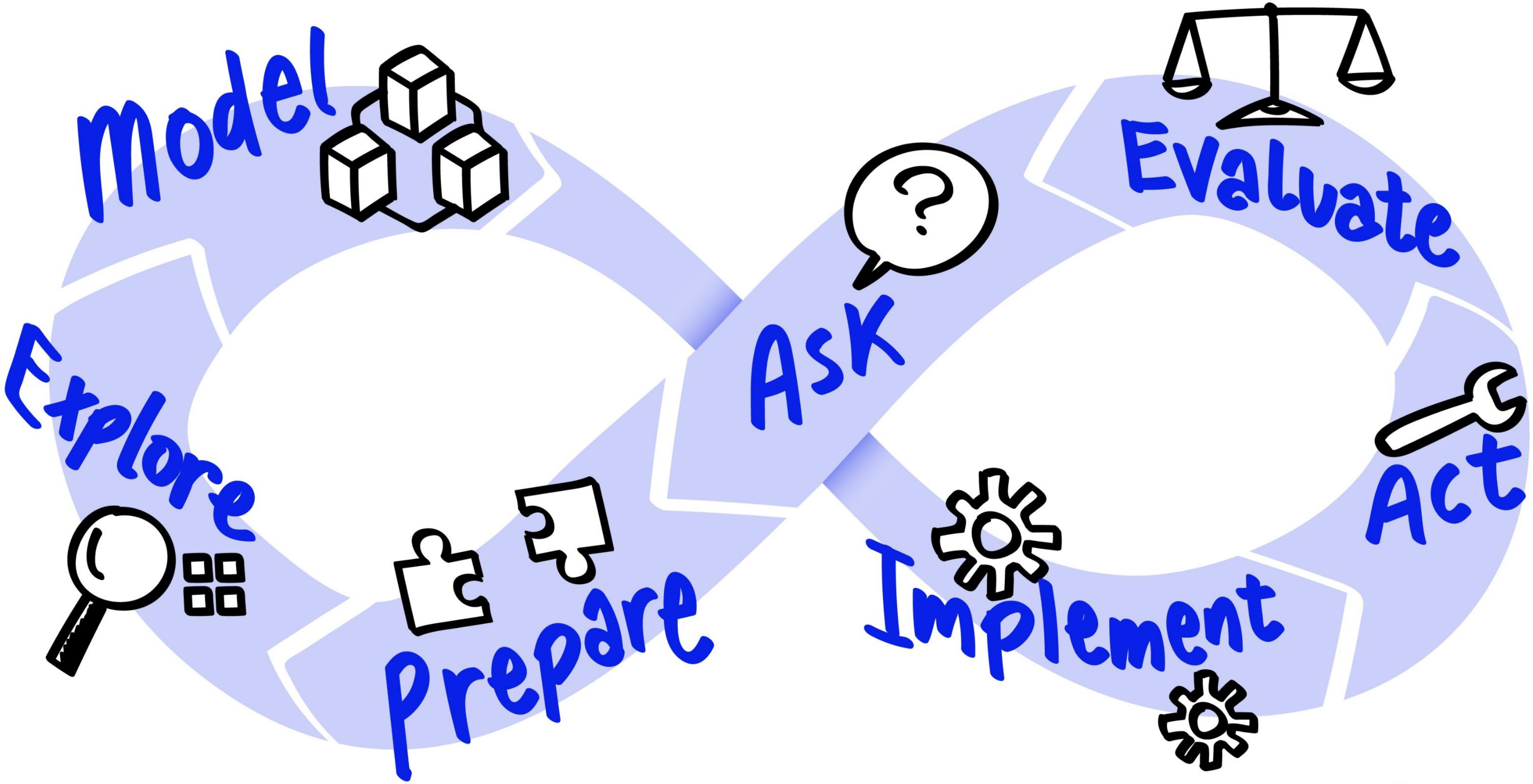
- Unsupervised learning – no target needed
- Excellent data dimension reduction technique
- Many possible ways to implement in SAS – Cheat Sheet will help you find the way

Data Science Jam Sessions by SAS

Talk of the GEEKS

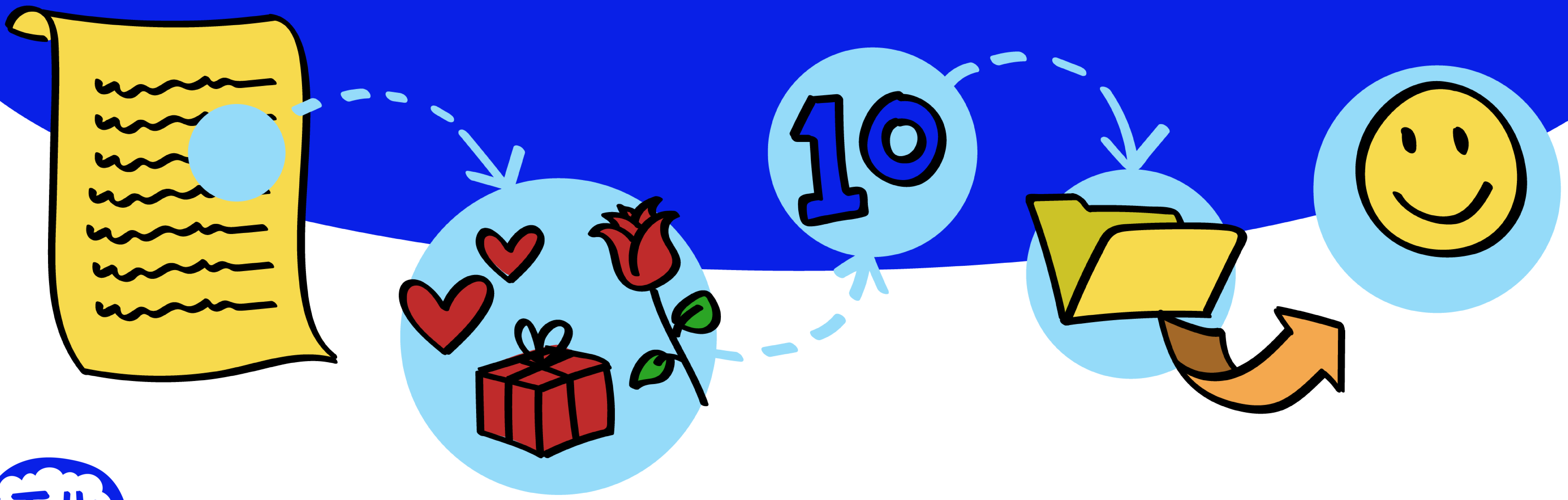
sas THE POWER TO KNOW

Analytical Lifecycle



Text analytics, using machines to mine an untouched source of information

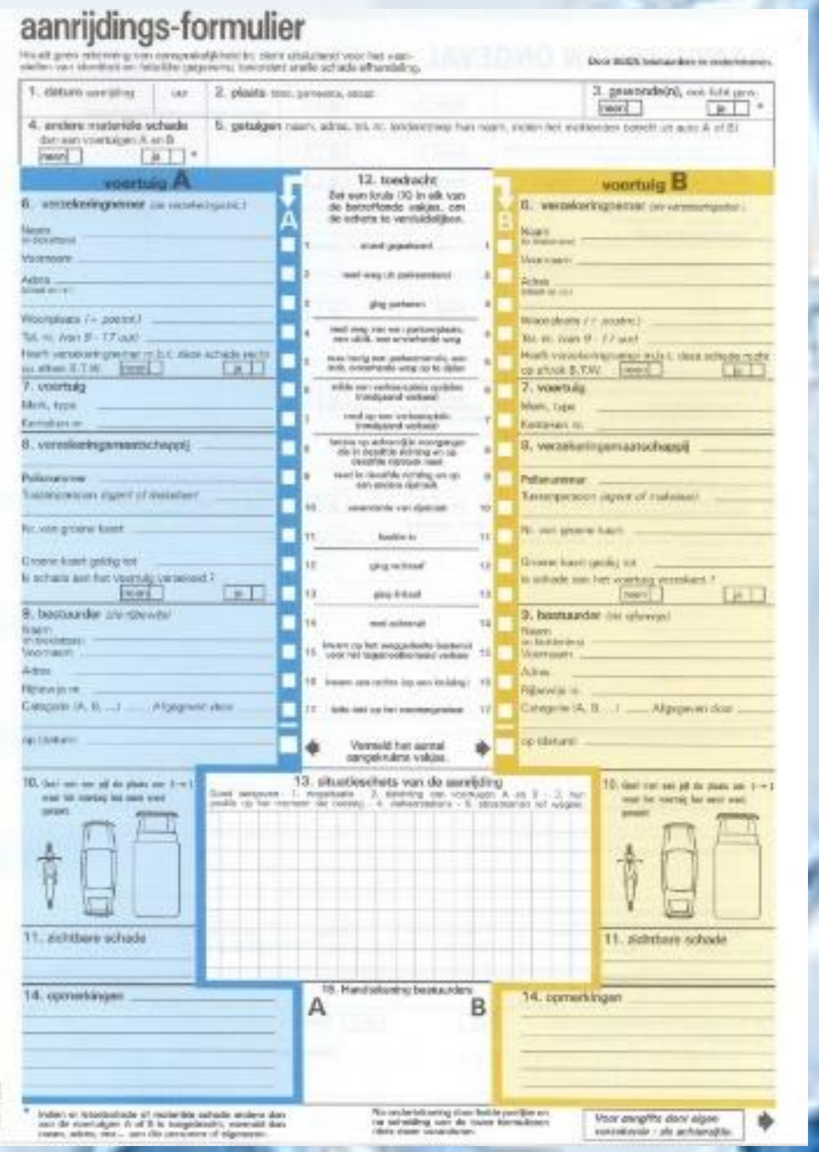
Speaker: Reinout Mensaert

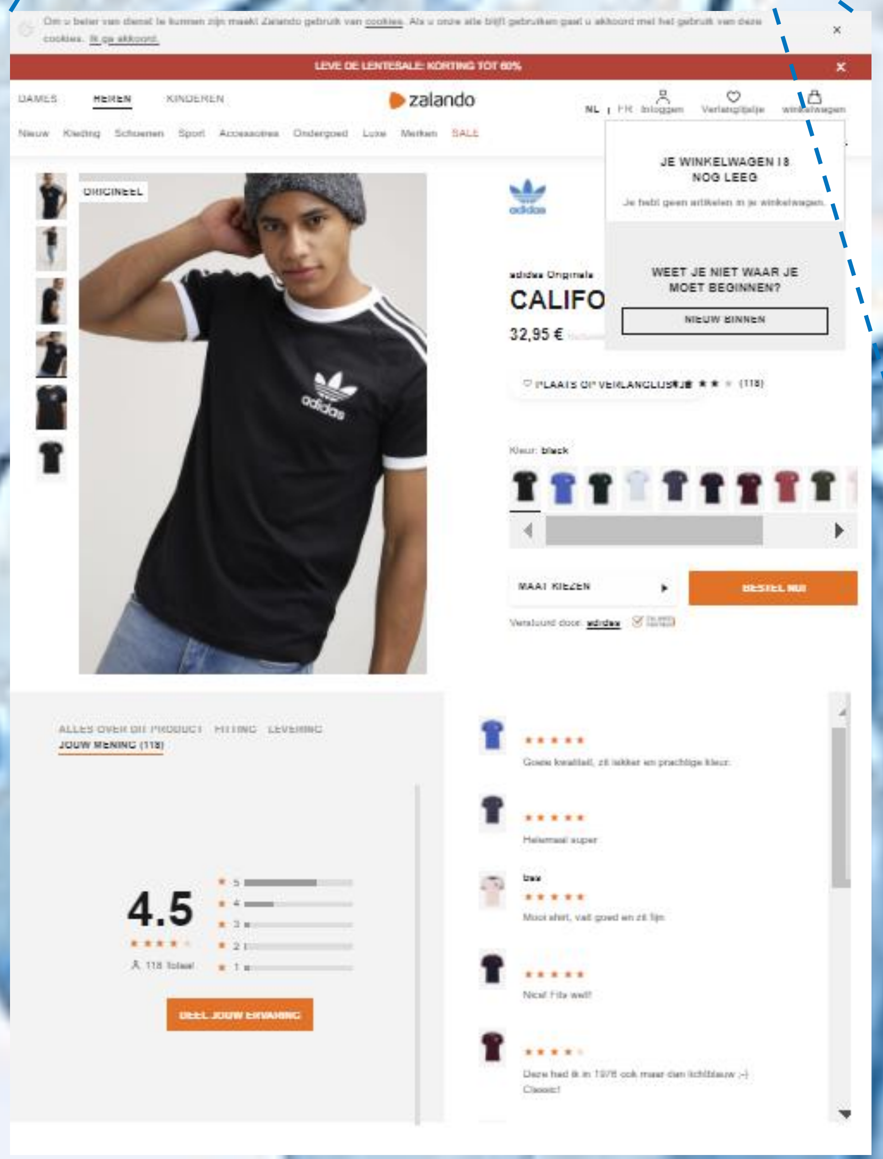


3 May | At the Bebop

Text analytics

Why?






Complaint Letter

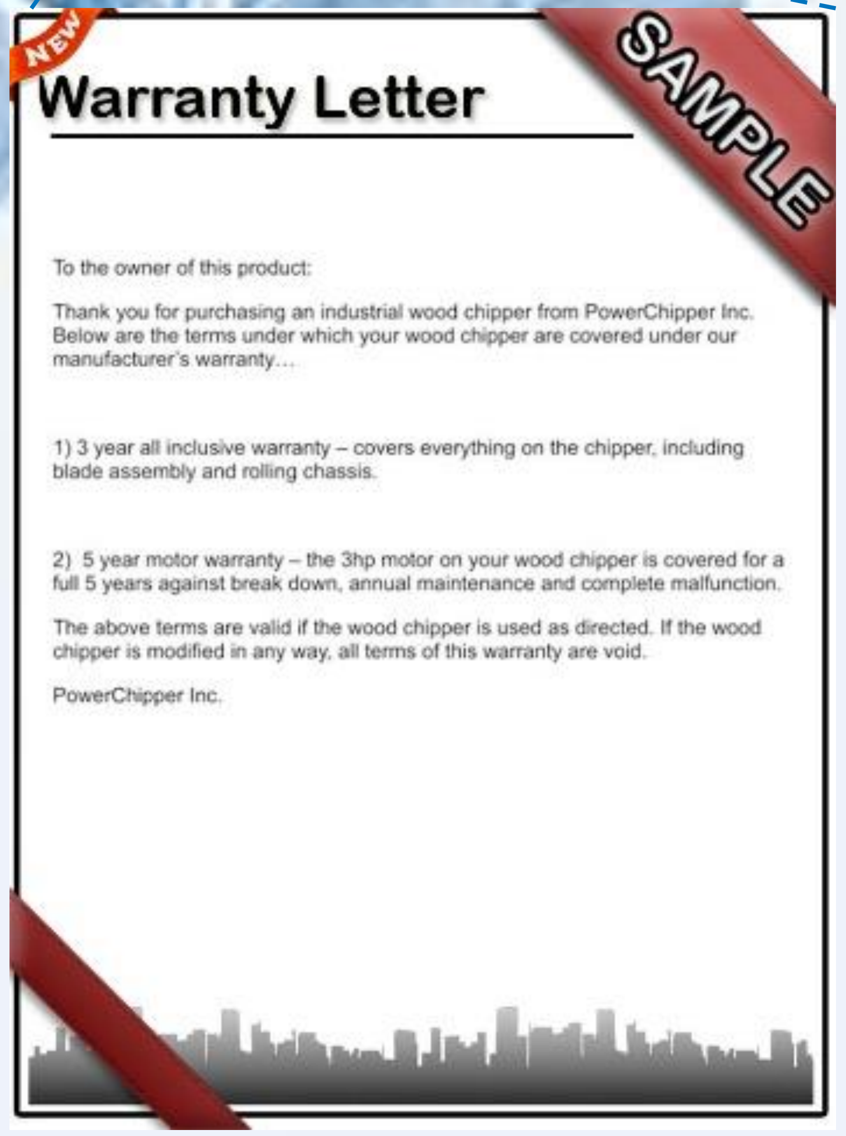
Sender Name
Sender's Title or Position
Sender's Organization Name
Sender Street Address
City, State, Zip Code

Recipient's Name
Recipient's Position or Title
Recipient's Organization Name
Recipient's Street Address
City, State, Zip Code

Dear Sir/ Madam,

Yours Sincerely

Write Your Name here


Warranty Letter

To the owner of this product:

Thank you for purchasing an industrial wood chipper from PowerChipper Inc. Below are the terms under which your wood chipper are covered under our manufacturer's warranty…

1) 3 year all inclusive warranty – covers everything on the chipper, including blade assembly and rolling chassis.

2) 5 year motor warranty – the 3hp motor on your wood chipper is covered for a full 5 years against break down, annual maintenance and complete malfunction.

The above terms are valid if the wood chipper is used as directed. If the wood chipper is modified in any way, all terms of this warranty are void.

PowerChipper Inc.

Automated Analysis of Text Documents

Sentiment Analysis | Topic Discovery | Text categorization

Text analytics

Method

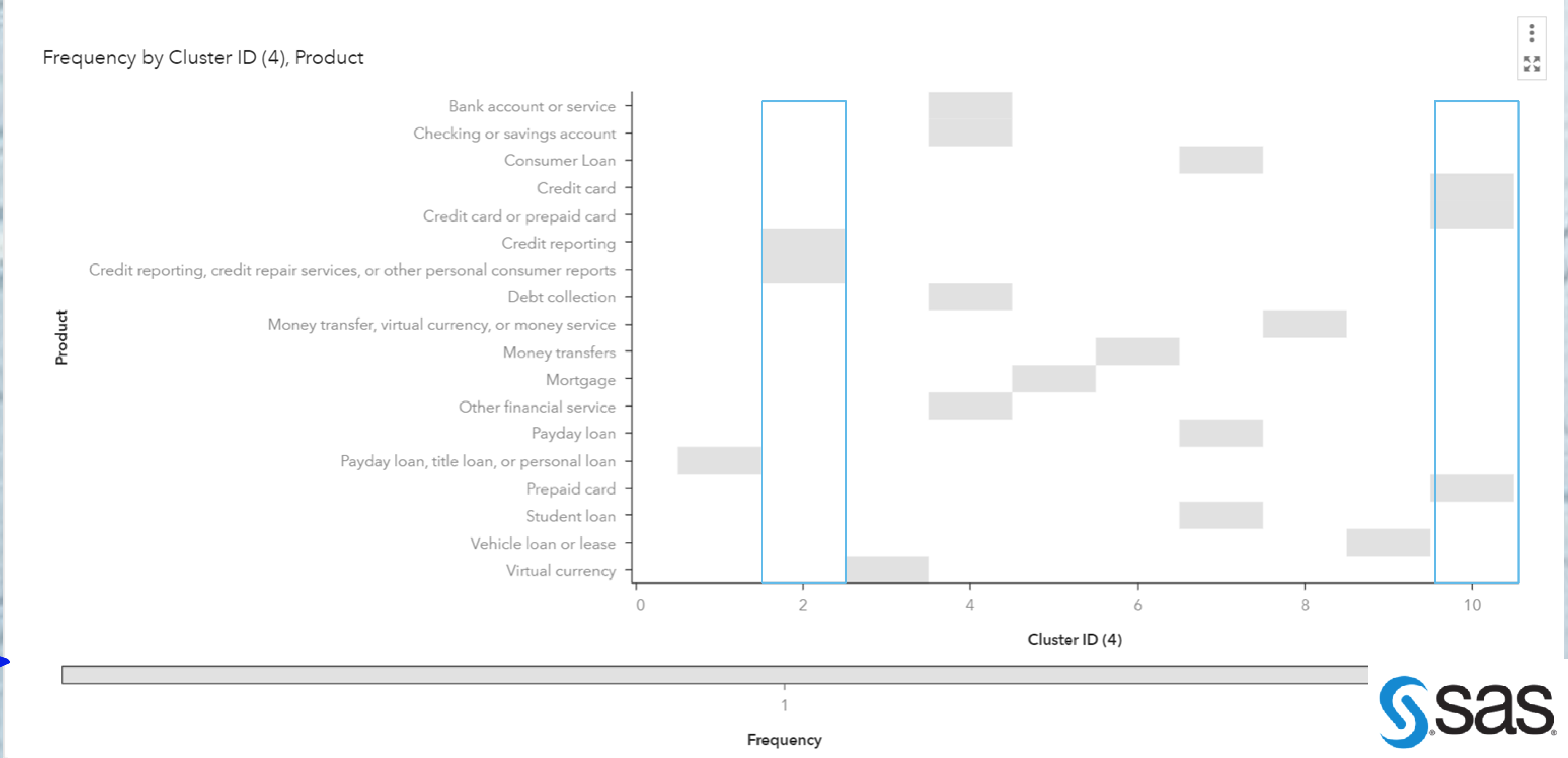
Doing mathematics on text by counting and clustering words

RowNr	Product
1	Bank account or service
2	Checking or savings account
3	Consumer Loan
4	Credit card
5	Credit card or prepaid card
6	Credit reporting
7	Credit reporting, credit repair services, or other personal consumer reports
8	Debt collection
9	Money transfer, virtual currency, or money service
10	Money transfers
11	Mortgage
12	Other financial service
13	Payday loan
14	Payday loan, title loan, or personal loan
15	Prepaid card
16	Student loan
17	Vehicle loan or lease
18	Virtual currency

Text analytics

Method

Doing mathematics on text by counting and clustering words



Text analytics

Method

Doing mathematics on text by counting and clustering words

RowNr	Product
1	Bank account or service
2	Checking or savings account
3	Consumer Loan
4	Credit card
5	Credit card or prepaid card
6	Credit reporting
7	Credit reporting, credit repair services, or other personal consumer reports
8	Debt collection
9	Money transfer, virtual currency, or money service
10	Money transfers
11	Mortgage
12	Other financial service
13	Payday loan
14	Payday loan, title loan, or personal loan
15	Prepaid card
16	Student loan
17	Vehicle loan or lease
18	Virtual currency

Concepts

Cities
Names
Telephone numbers
Account number

Text Parsing

Remove Stopwords
Stemming
Synonyms
Spelling mistakes
Slang
...

Topics

Text analytics

Method

Model Studio - Build Models

Search | sbxrem

Consumer Complaints Product Grouping › Topics

Close

Topics 11

	Topic	Documents
☑	card, +prepay, credit, loan, account	3
☐	loan, personal loan, title loan, title, payday	3
☐	+report, personal consumer, credit repair, repair, credit	2
☐	virtual currency, currency, virtual, transfer, money	2
☐	vehicle, lease, loan, student, payday	2
☐	money, +transfer, transfer, +service, +prepay	2
☐	account, bank, +service, transfer, money	1
☐	loan, consumer, personal consumer, credit repair, repair	

Terms 3 of 37

All | Matched

	Term	Relevancy	Role	Documents	Frequency
☐	card	0.758	N	3	4
☐	prepay	0.549	V	2	2
	prepaid	.	V	2	2
☐	credit	0.349	N	4	5

Topic Discovery

Documents 3

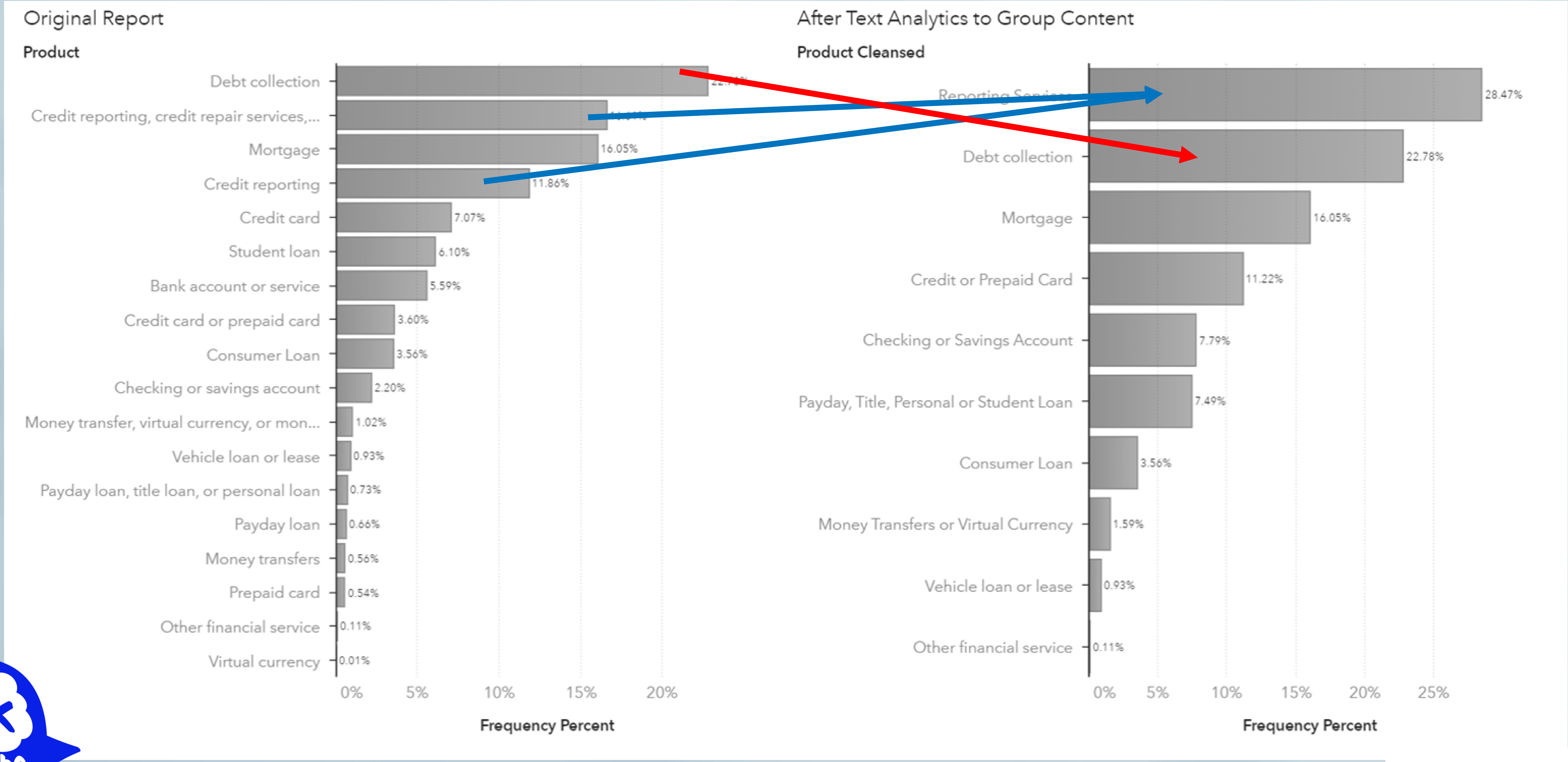
All | Matched

Product	Relevancy	Sentiment
Prepaid card	1.000	
Credit card or prepaid card	0.991	
Credit card	0.965	

Document 1 of 3

Text analytics

Conclusion



Text analytics

Conclusion

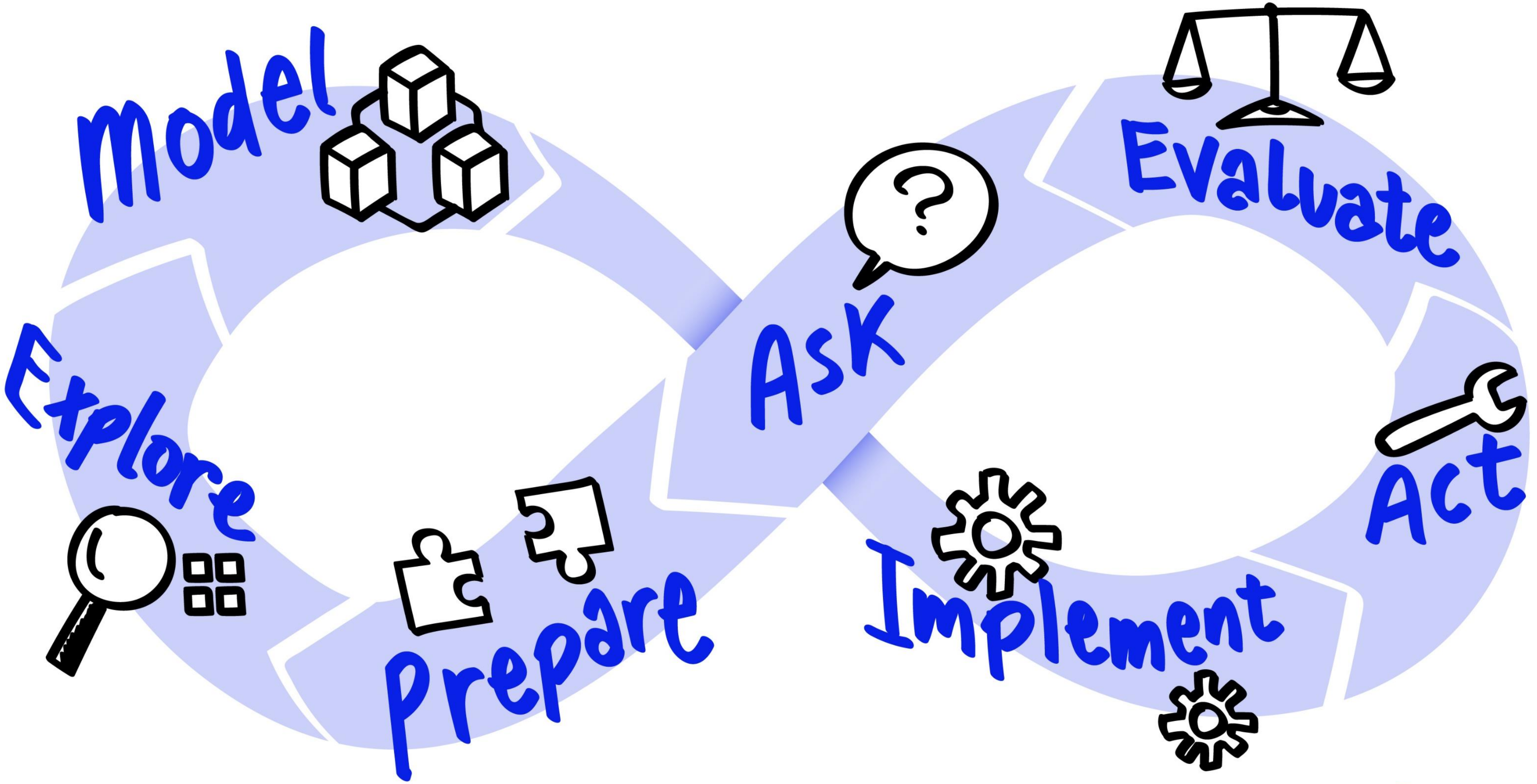
- Text analytics market size is expected to grow from USD 3.97 Billion in 2017 to USD 8.79 Billion by 2022 – MarketsAndMarkets Research
- Text analytics allows you to gain information out of your text by running advanced analytical techniques on the data
- Specific and elaborate data preparation work is necessary for good results
- SAS Visual Text Analytics does it all out-of-the-box

Data Science Jam Sessions by SAS

Talk of the GEEKS

sas THE POWER TO KNOW®

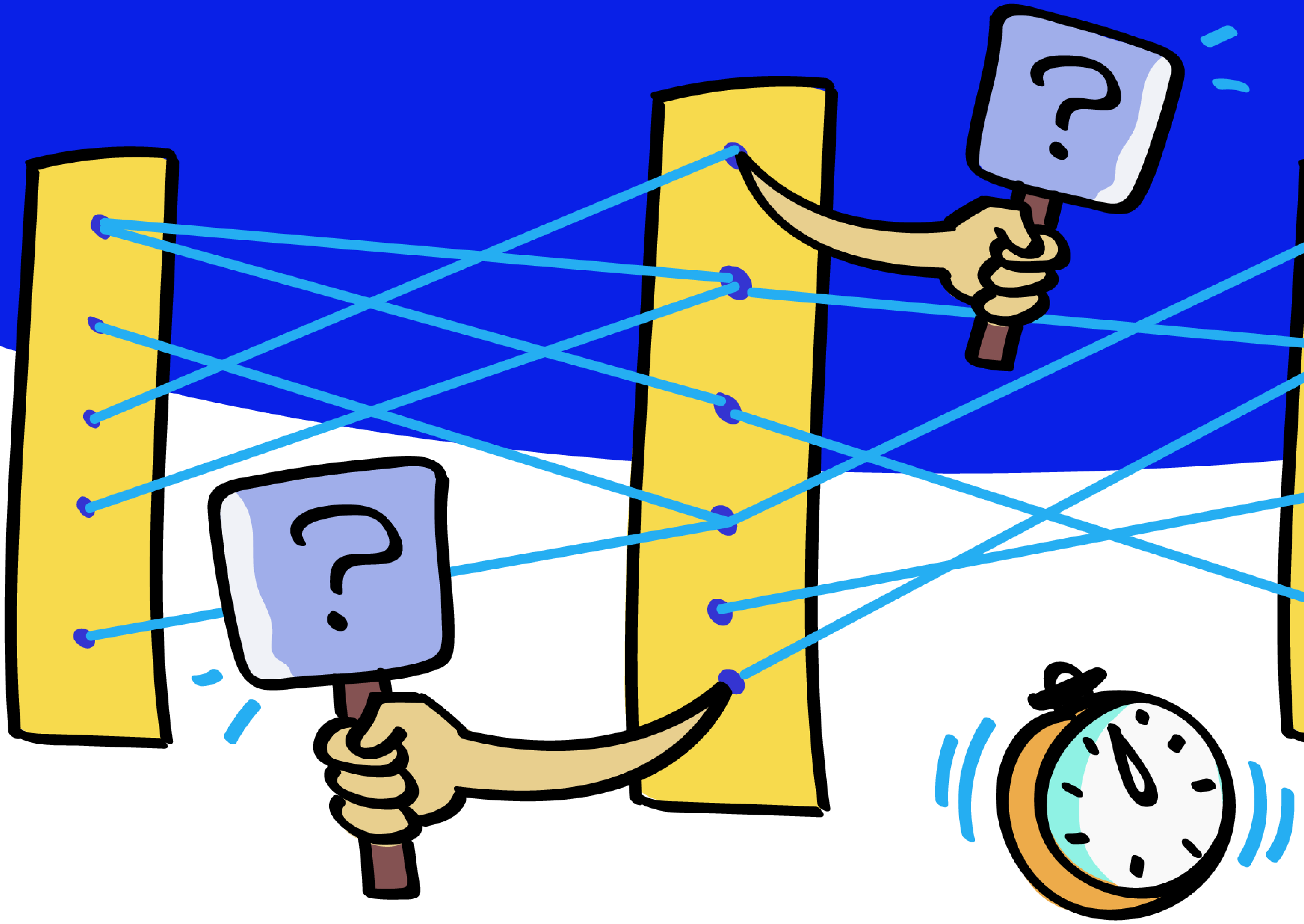
Analytical Lifecycle



Autotuning in SAS

Hey data scientist! Did you already optimize your model hyperparameters?

Speaker: Véronique Van Vlasselaer



Talk of the GEEKS

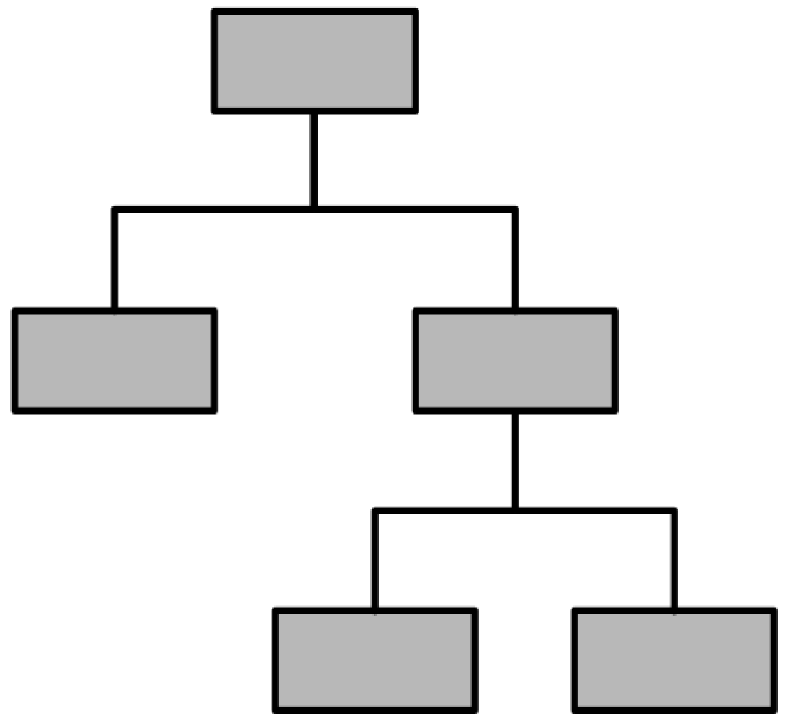
3 May | At the Bebop

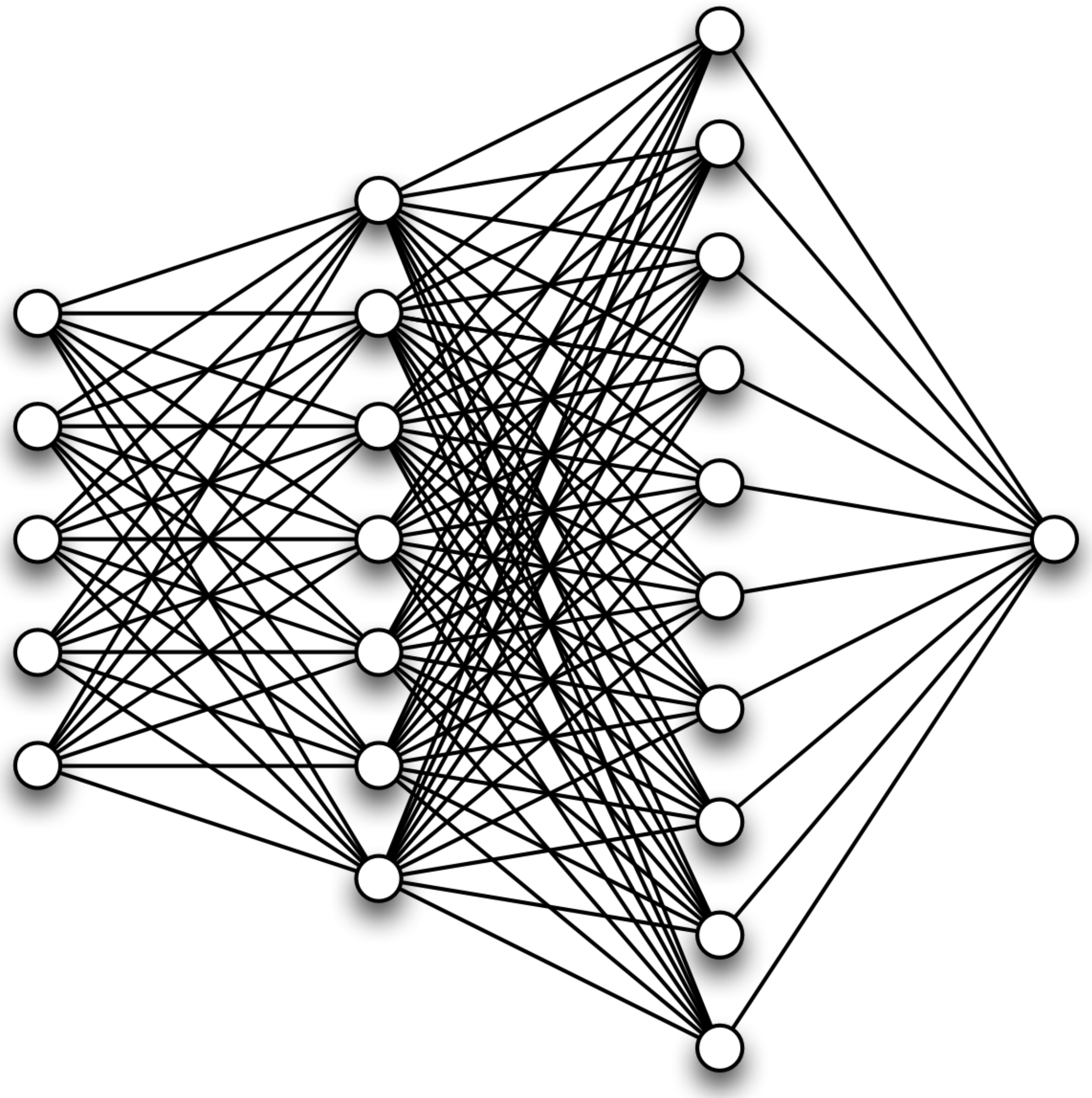
sas THE POWER TO KNOW.

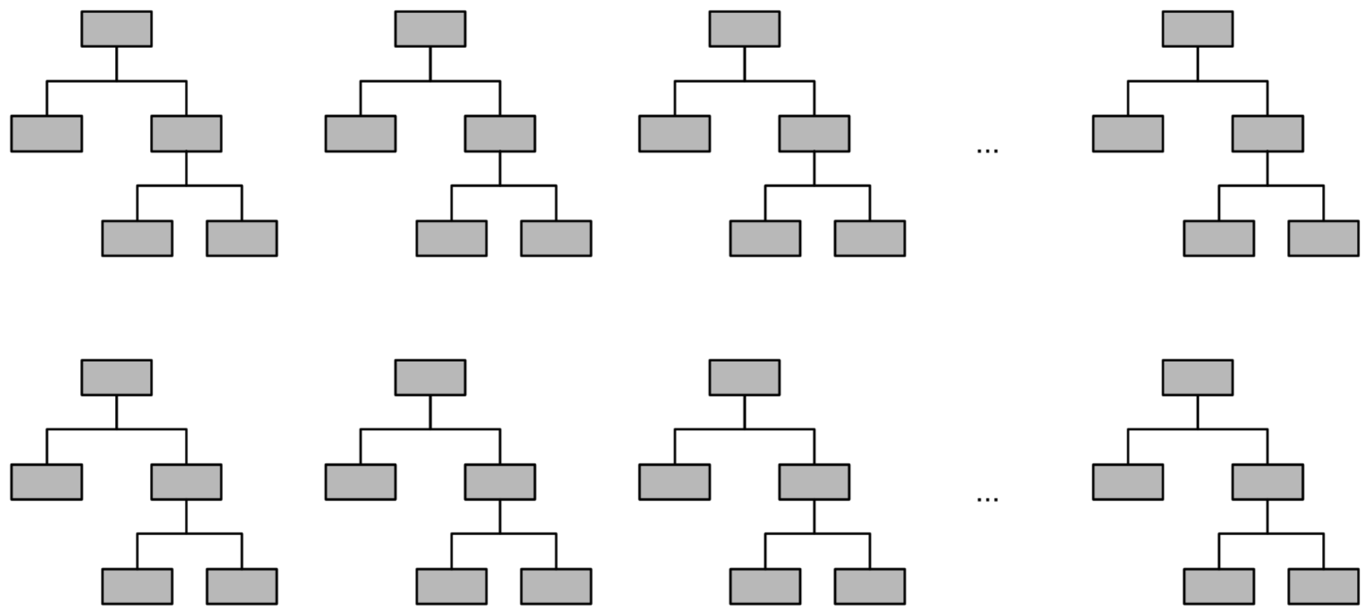
Autotuning in SAS

What?

- Training a model involves using an algorithm to determine **model parameters** or other logic to map inputs to a target.





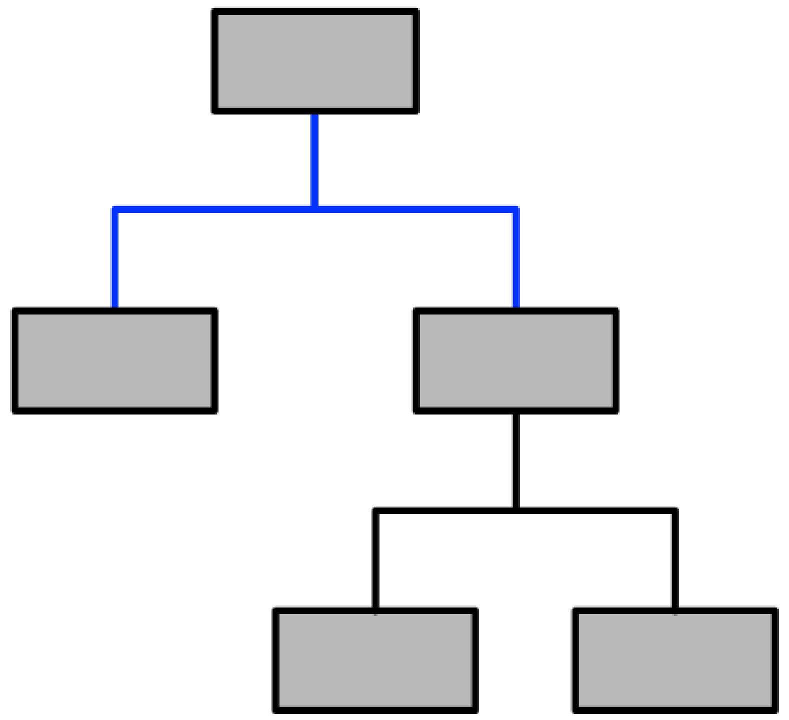


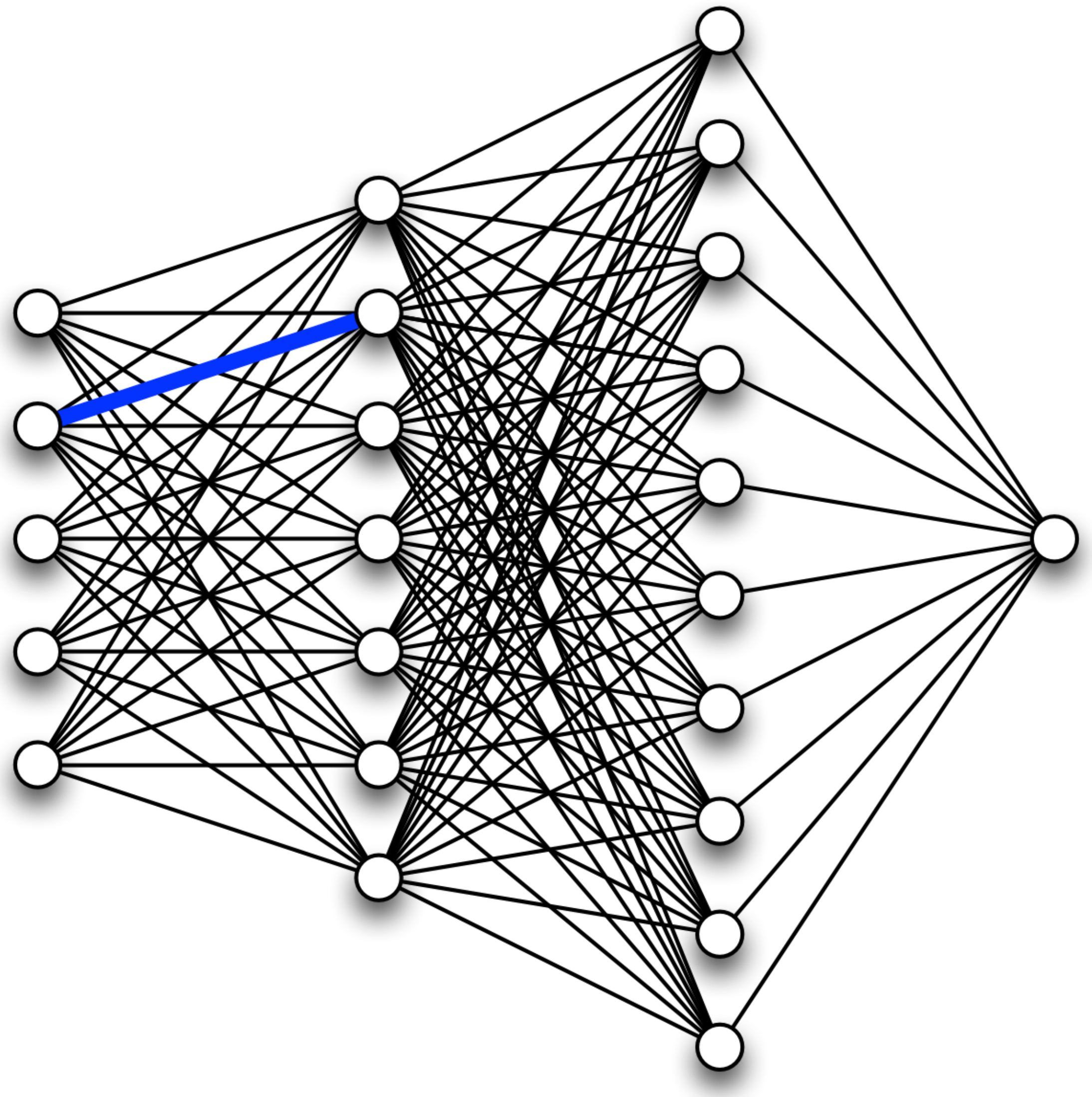
$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 \, x_2$$

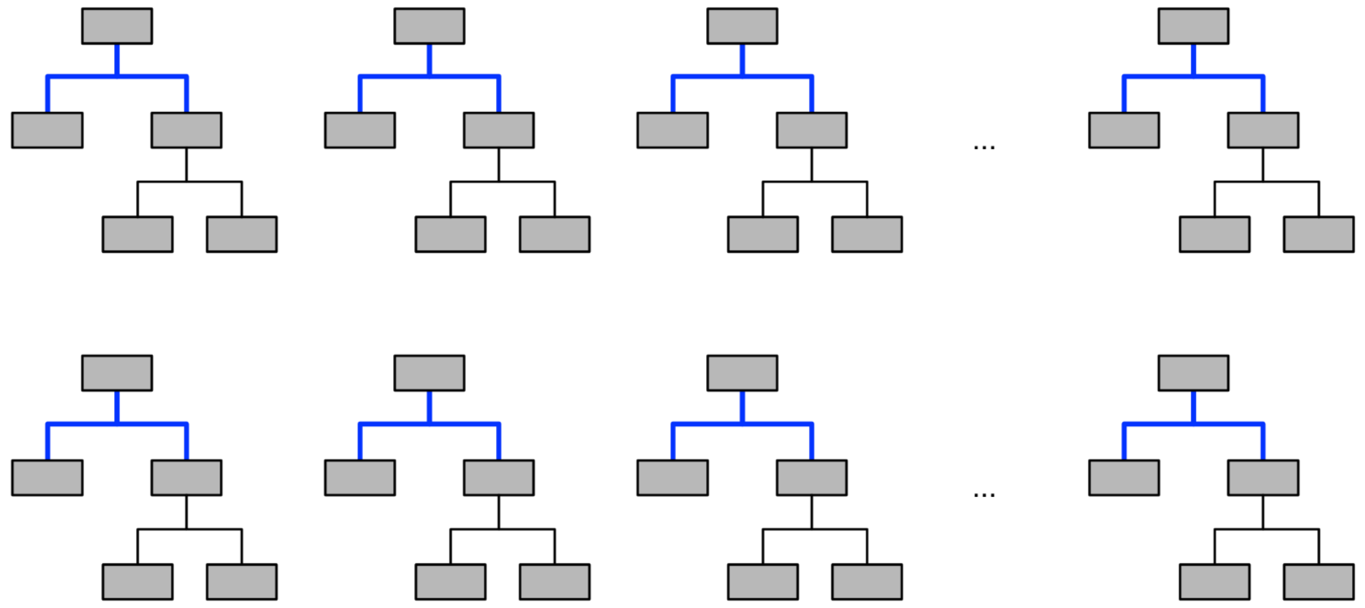
Autotuning in SAS

What?

- Training a model involves using an algorithm to determine **model parameters** or other logic to map inputs to a target.





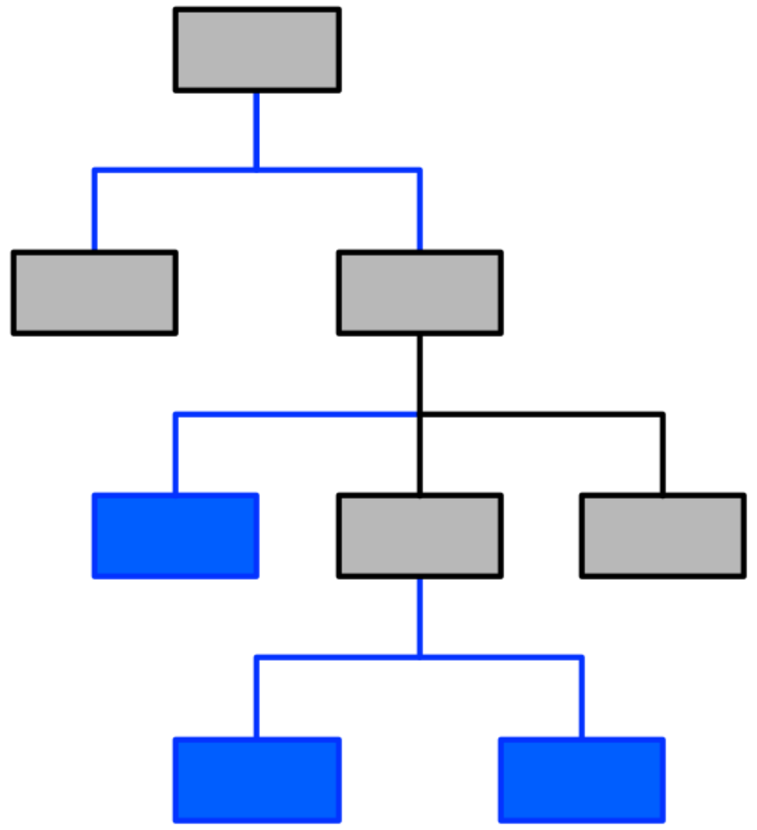


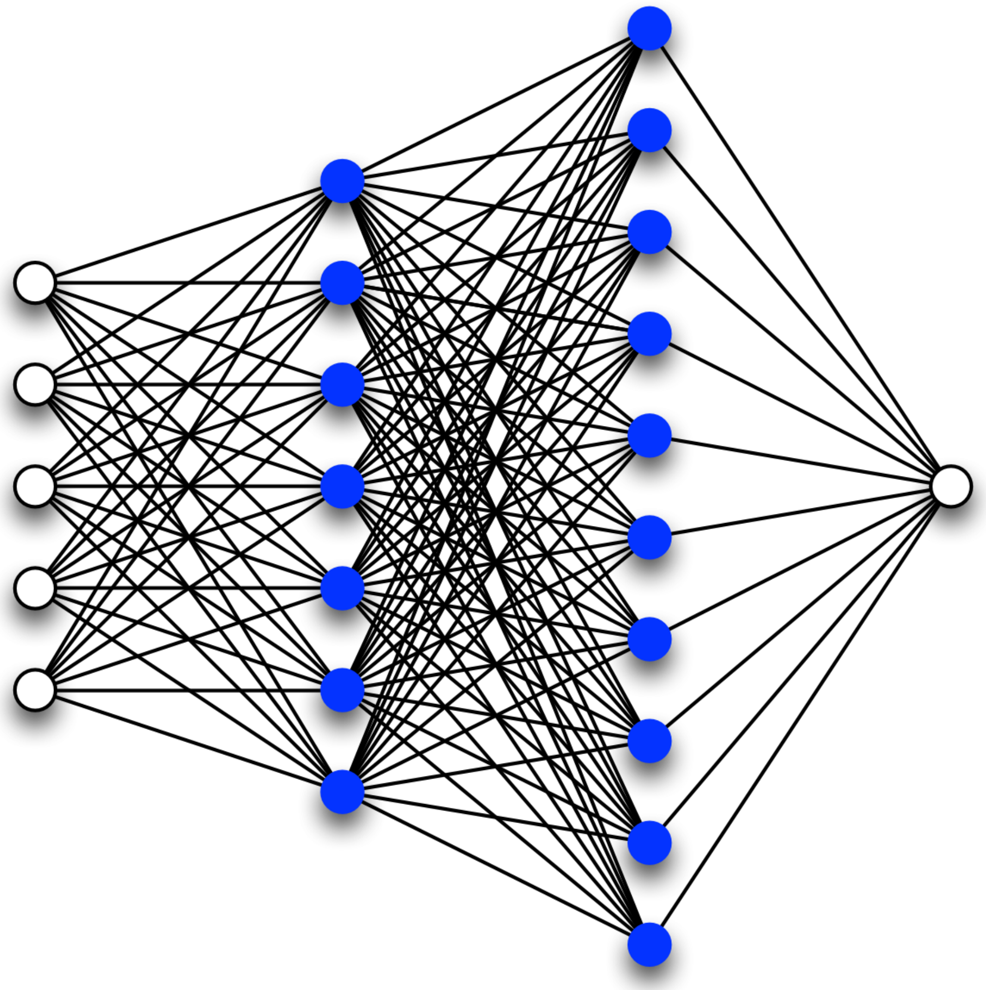
$$f(x) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1} x_1 + \boldsymbol{\beta_2} x_2 + \boldsymbol{\beta_3} x_1 \, x_2$$

Autotuning in SAS
What?

- Tuning a model involves determining the **algorithm hyperparameters** (tuning options) that result in the model which maximizes predictability on an independent data set.

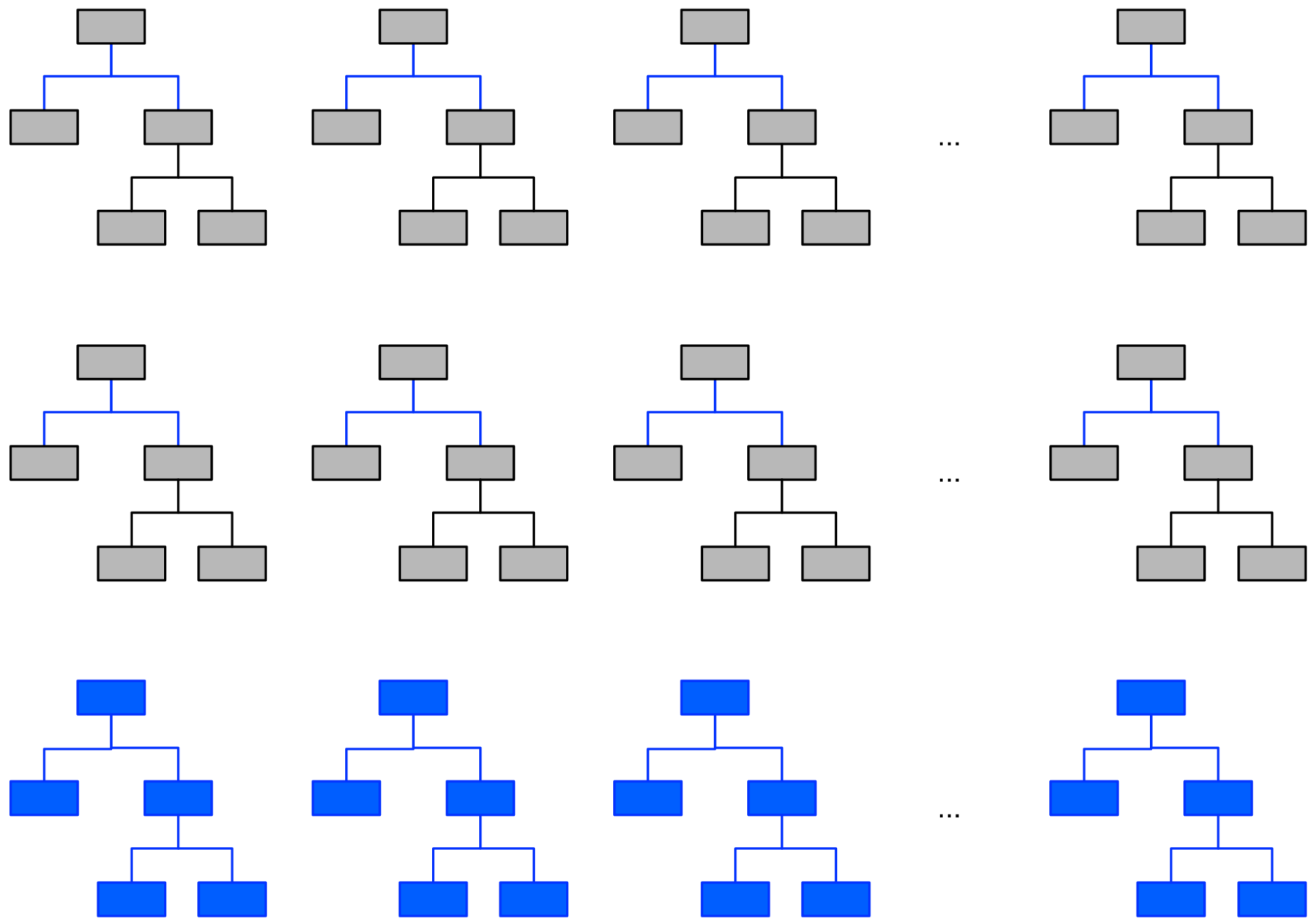
Splitting criterion? Width? Depth?





Number of layers? Number of neurons?

Number of trees? Variables? Observations?

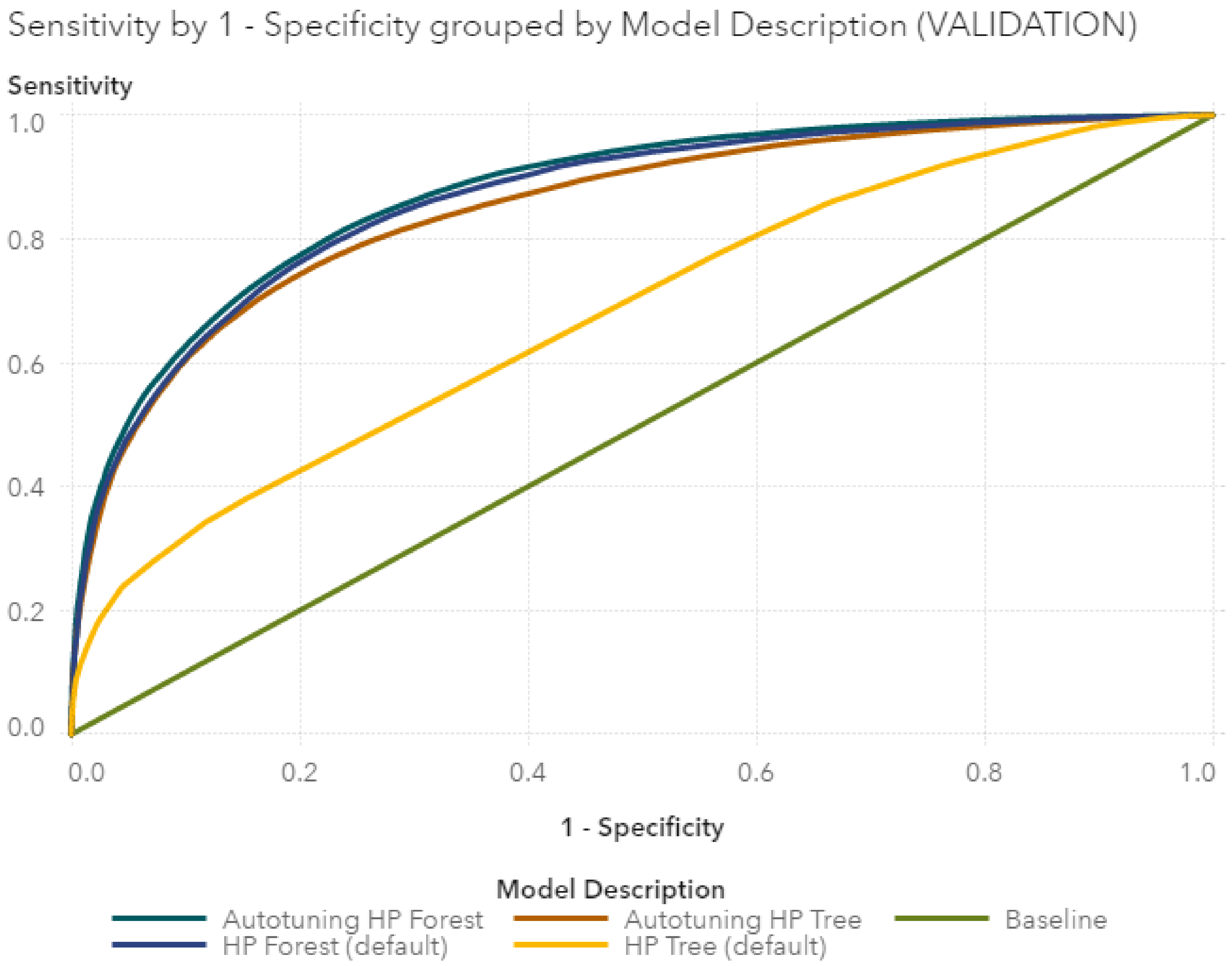


Autotuning in SAS

Why?

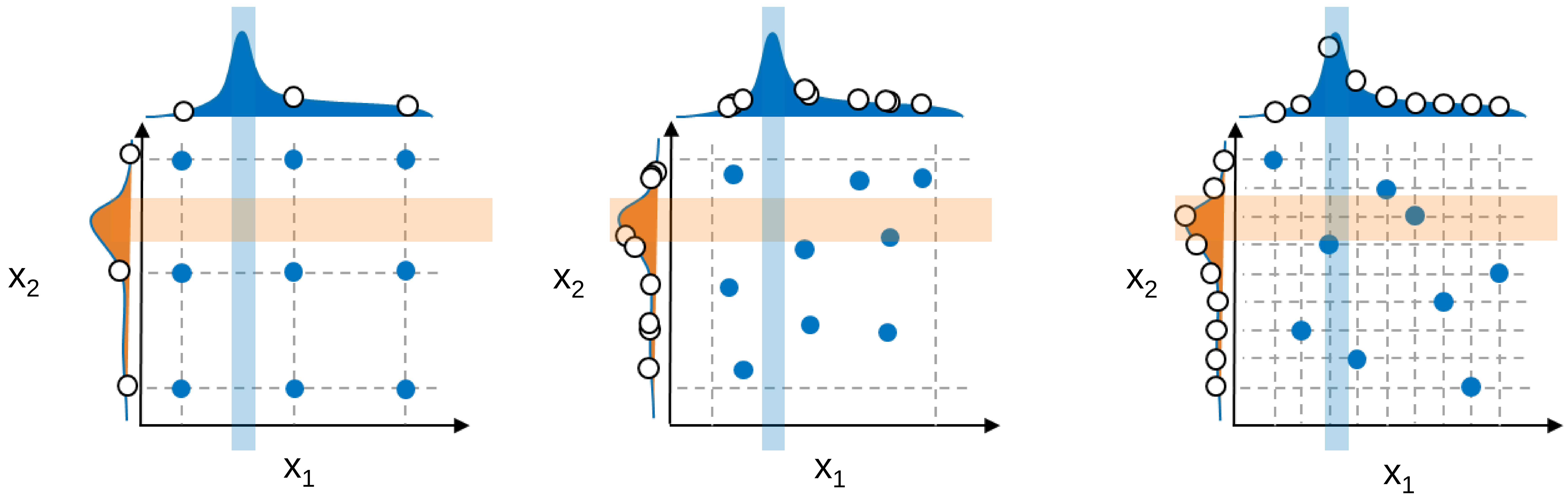
- Model performance might drastically improve just by adjusting the model settings…

- but manual search for optimal hyperparameters is often slow and inefficient.



Autotuning in SAS
Method

- A new functionality in Visual Data Mining and Machine Learning (VDMML)
- Exhaustive search versus heuristics



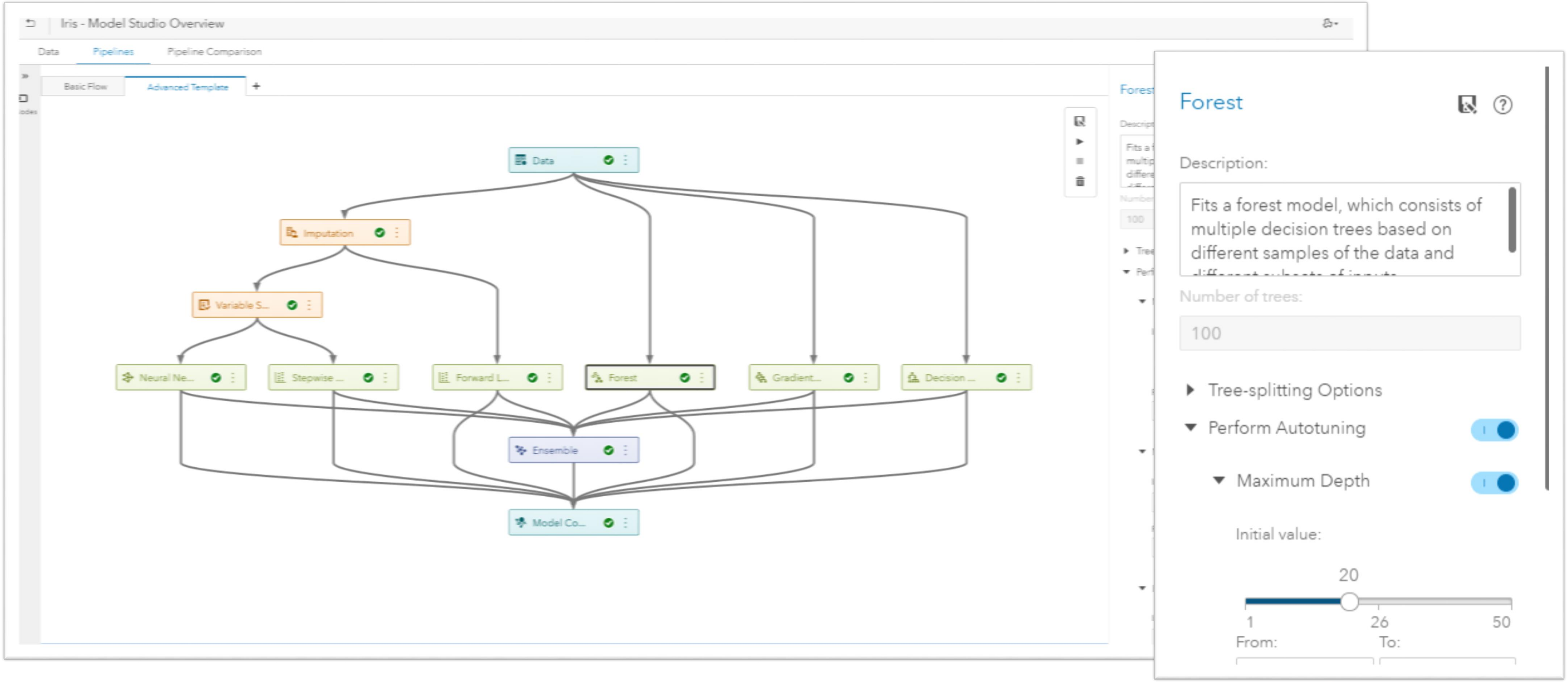
Standard Grid Search

Random Search

Latin Hypercube

Autotuning in SAS
Method

- SAS Model Studio - a brand new visual interface for the data scientist.



Autotuning in SAS

Method

- SAS Studio – the programming interface for the data scientist.
  - **`autotune`** statement with option **`tuningparameters=`**.

```
/* Random Forest */
proc forest data=MyDataSet;
    target myTarget / level=nominal;
    inp
    aut

run;
```

```
/* Neural Network */
proc nnet data=MyDataSet;
    target myTarget / level=nominal;
    input
    autot
        r
    train
run;
```

```
/* Gradient Boosting */
proc gradboost data=MyDataSet;
    target myTarget / level=nominal;
    inp
    aut

run;
```

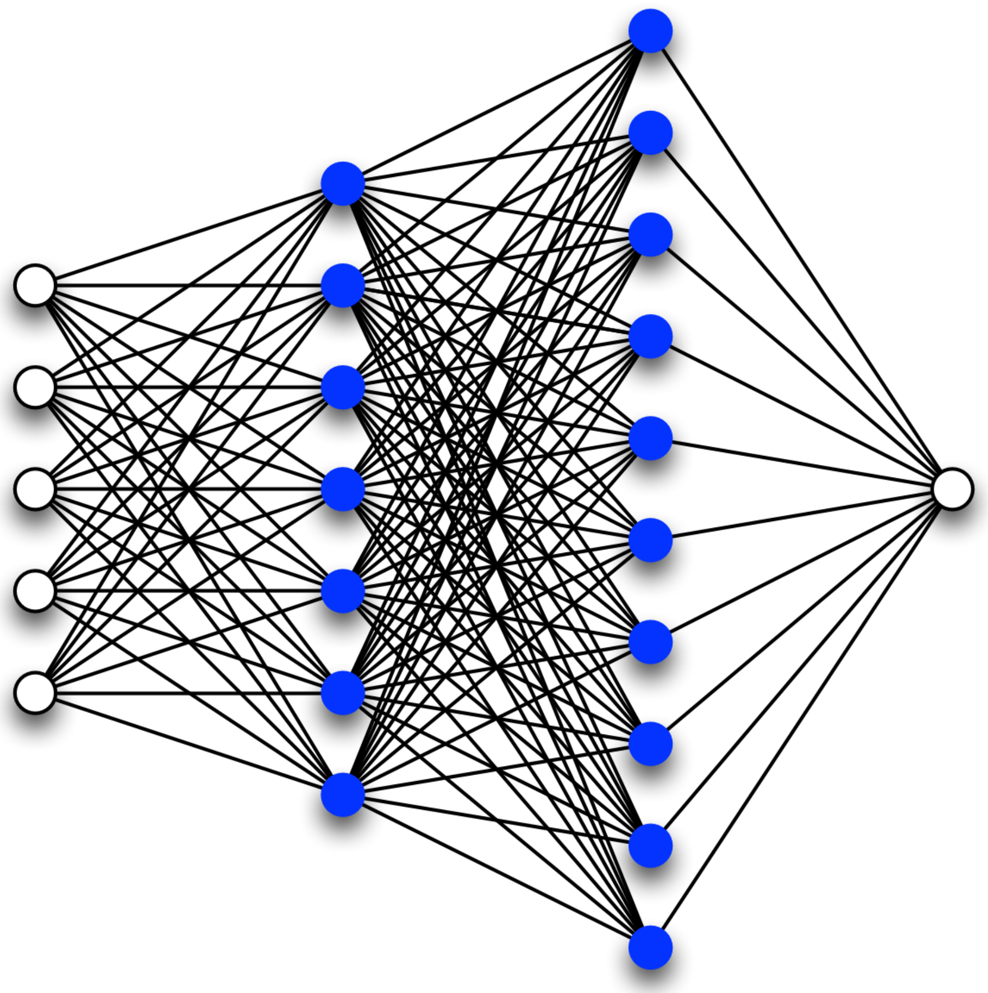
```
/* Decision Tree */
proc treesplit data=MyDataSet;
    target myTarget / level=nominal;
    input myInputs / level=interval;
    autotune tuningparameters=(maxdepth numbin criterion) objective=misc
        fraction=0.3;
run;
```

Documentation: https://support.sas.com/documentation/prod-p/vdmml/index.html

Autotuning in SAS
Conclusion

- Traditionally, focus on comparing various models (e.g., decision tree, logistic regression, neural network, SVMs, etc.) with their default settings.

- Autotuning a model tries to find the optimal **model hyperparameters** or model settings that results in the best performance. Example:
  - **Neural network**: number of hidden layers, neurons, L1/L2 regularization, etc.
  - **Decision tree**: maximum depth, splitting criterion, etc.
  - **Random Forest**: number of trees, number of variables for each tree, etc.

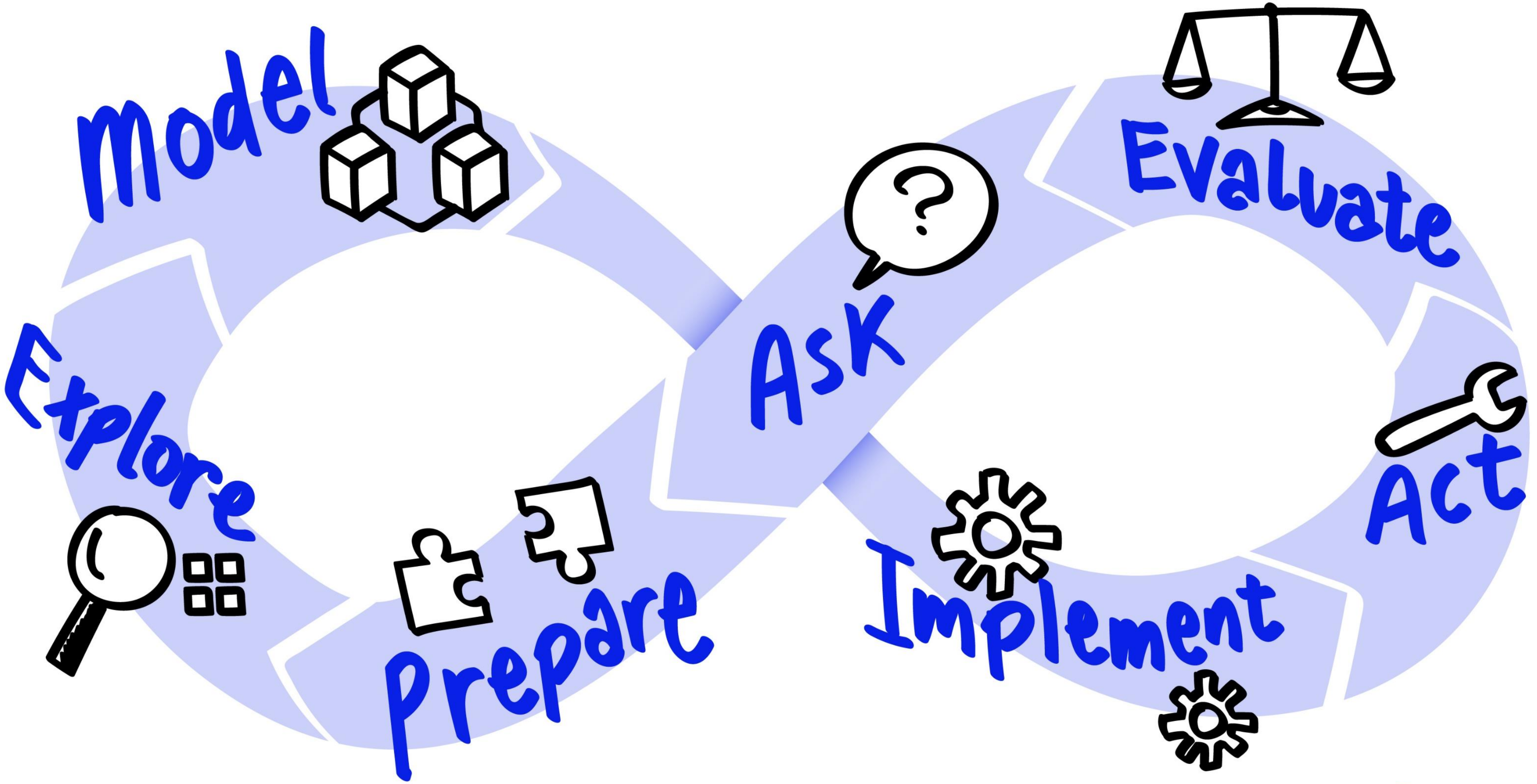


Data Science Jam Sessions by SAS

Talk of the GEEKS

sas. | THE POWER TO KNOW.

Analytical Lifecycle



Computer Vision in SAS: keeping AI (an eye) on the future

Speaker: Jaimy Van Dijk

Talk of the GEEKS

3 May | At the Bebop

sas THE POWER TO KNOW.

Computer Vision in SAS

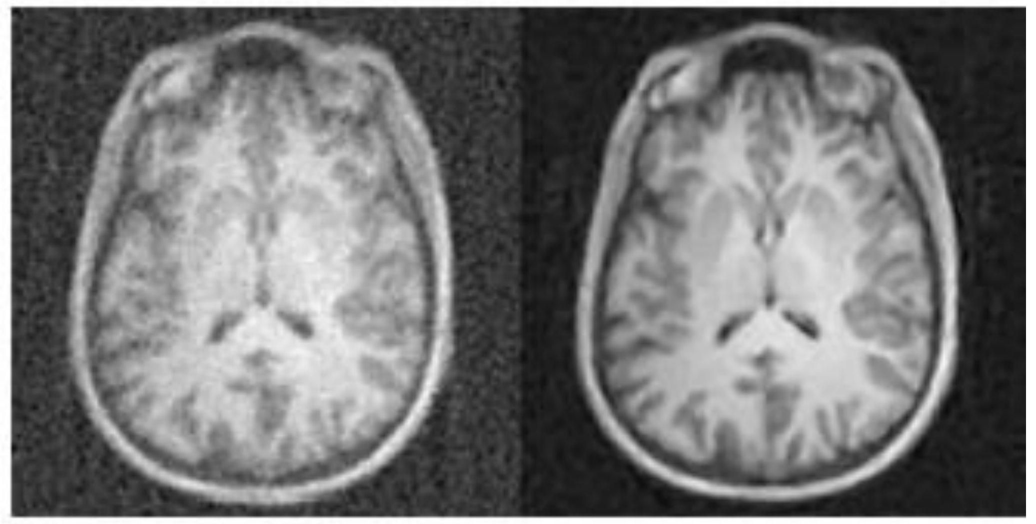
What?

- Everybody is talking about computer vision and image processing

PHYS.ORG

New artificial intelligence technique dramatically improves the quality of medical imaging

21 March 2018



A new artificial-intelligence-based approach to image reconstruction -- called AUTOMAP -- yields higher quality images from less data, reducing radiation doses for CT and PET and shortening scan times for MRI. Shown here are MR images reconstructed from the same data with conventional approaches (left) and AUTOMAP (right). Credit: Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital

INDEPENDENT

News › World › Asia

Chinese police are using facial-recognition glasses to scan travelers

The new accessories were unveiled ahead of the Chinese New Year rush and have already been used to arrest people

Tara Francis Chan Business Insider | Monday 12 February 2018 11:35 GMT | 1 comment

This photo taken on 5 February 2018 shows a police officer wearing a pair of smart glasses with a facial recognition system at Zhengzhou East Railway Station in Zhengzhou in China's central Henan province *AFP/Getty*

SAS GLOBAL FORUM

Demo: SciSports

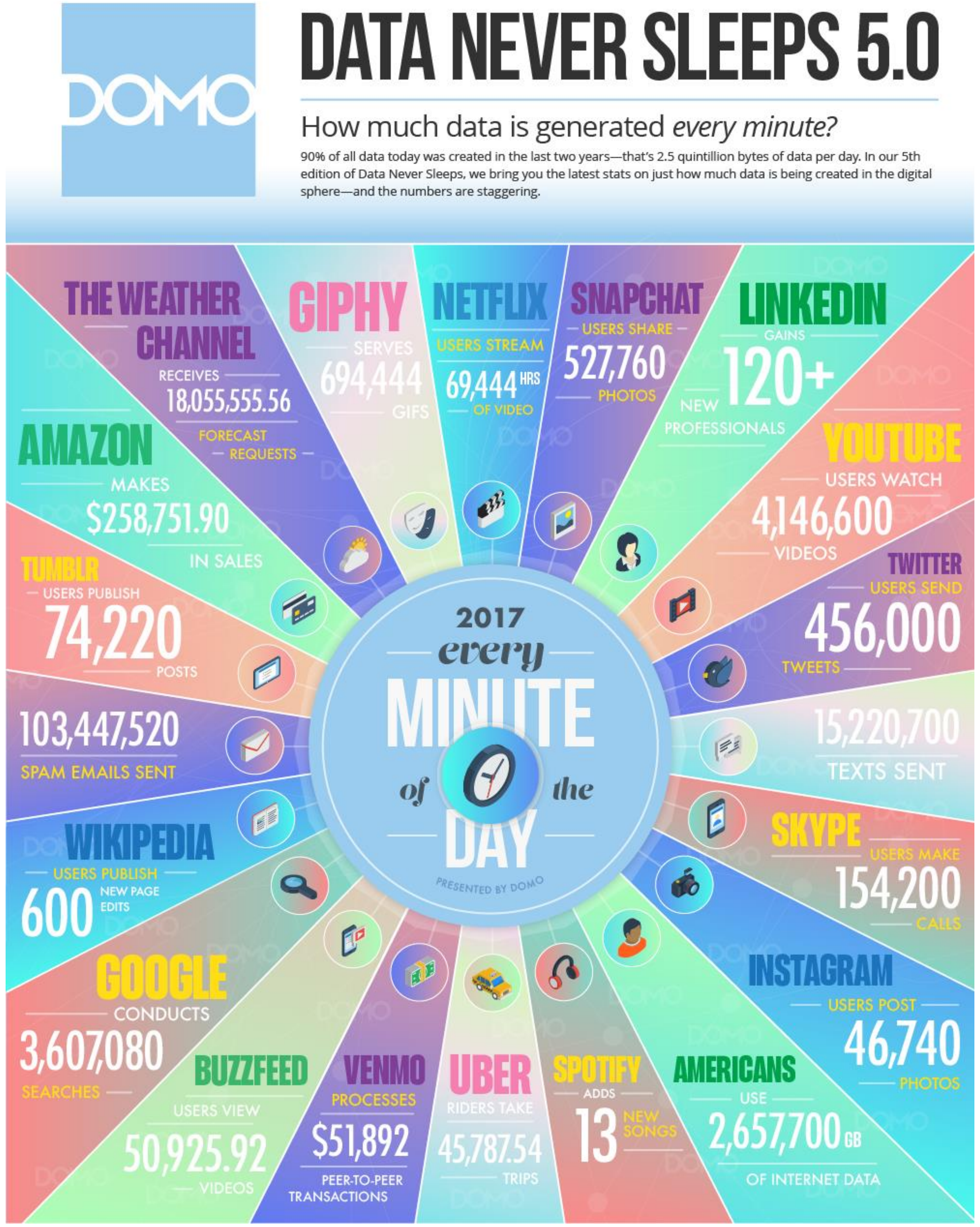
11:15

Executives from SciSports and SAS demonstrate the technology used to track soccer players in real-time.

Computer Vision in SAS

Why?

- Images are a relatively untapped data source
- We want computers/robots to interact with the world as we do



Computer Vision in SAS
What?

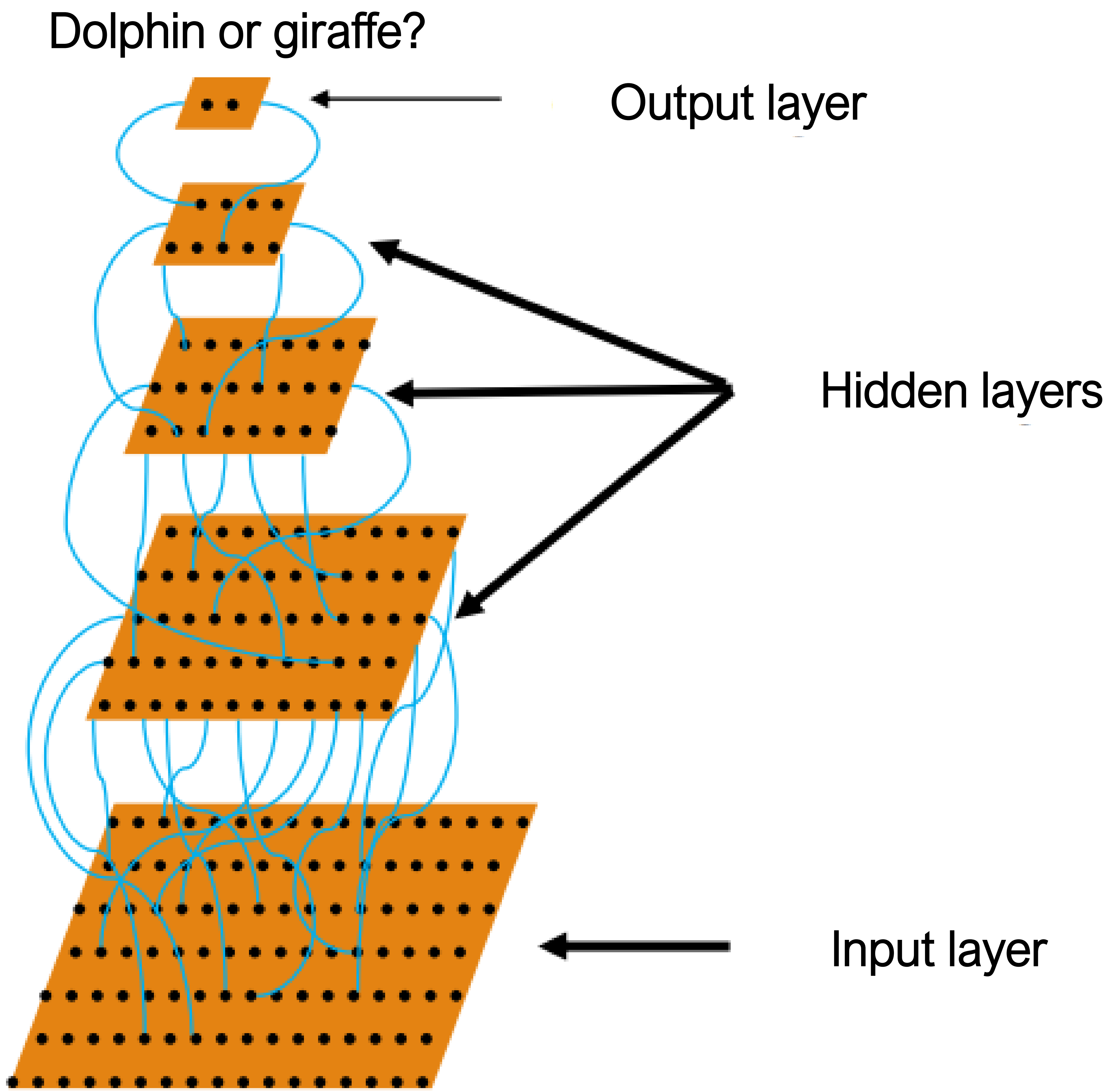
- What are image features?

- How are these features extracted?

Computer Vision in SAS

Convolutional Neural Networks

Is this a dolphin or a giraffe?

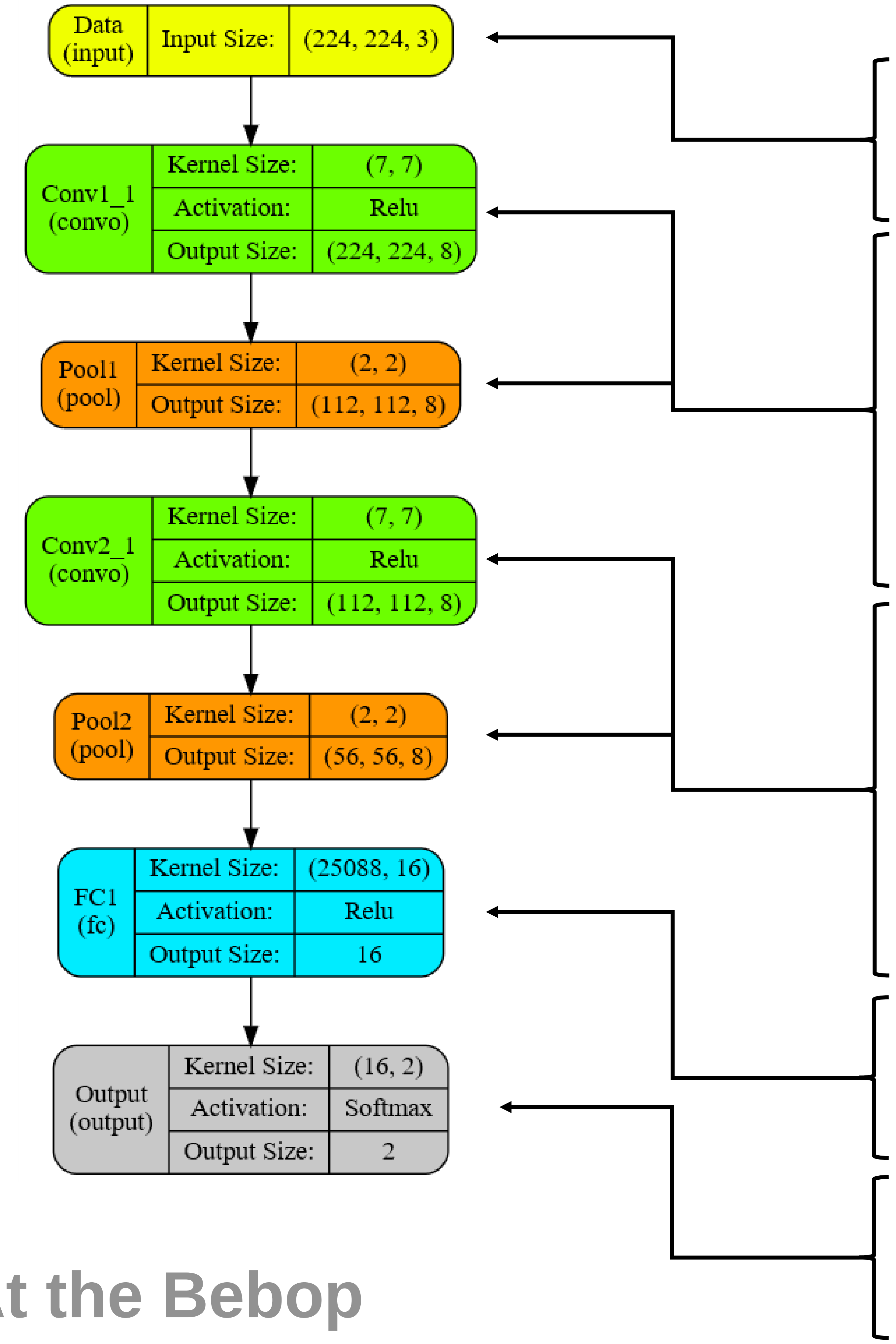


Computer Vision in SAS

Convolutional Neural Networks

1

DAG for Simple_CNN:



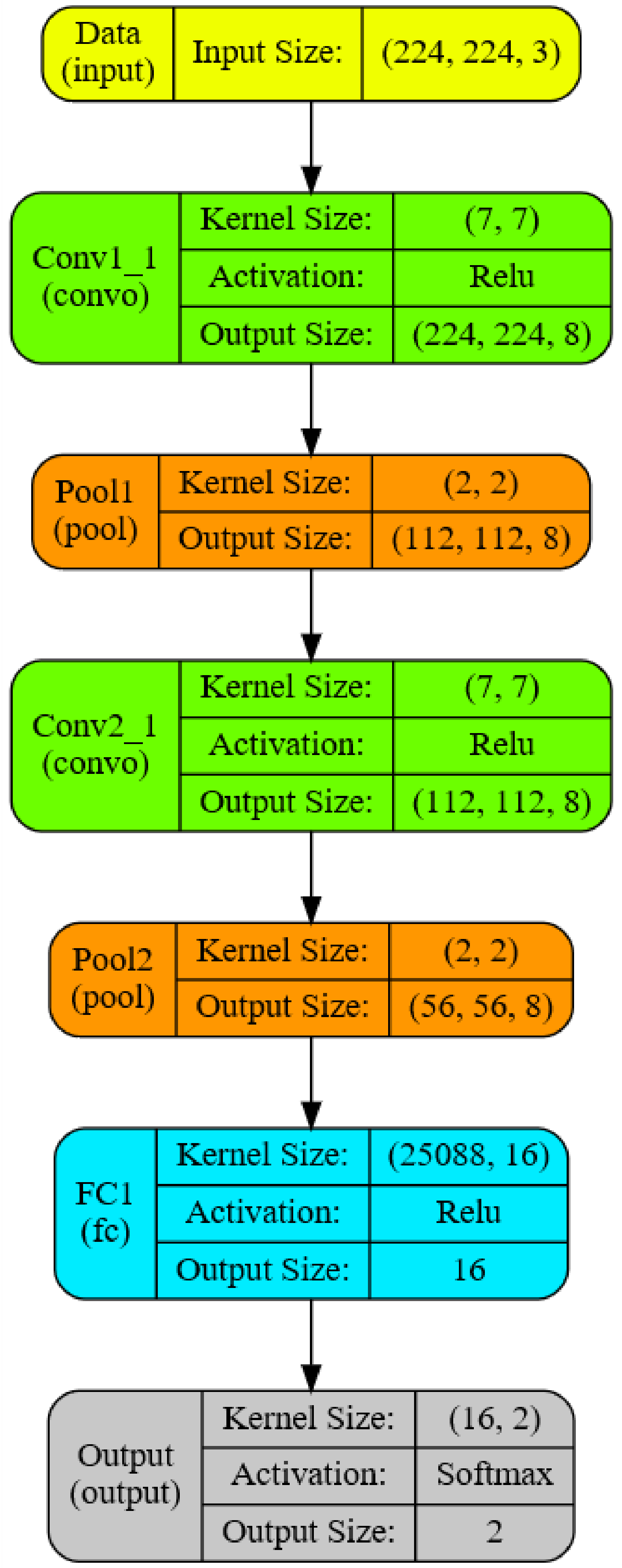

```
/* Build the network architecture. */
proc cas;
    session s;
    * Create a model and then add layers;
    deepLearn.buildModel /
        model = {'name'='SimpleCNN', 'replace'=True}
        type = 'CNN';

    * Add the layer that specifies the input dimensions;
    deepLearn.addLayer /
        model = 'SimpleCNN'
        name = 'data'
        layer = {'type'='input', 'nChannels'=3, 'width'=224, 'height'=224};

    * Add the first convolution layer;
    deepLearn.addLayer /
        model = 'SimpleCNN'
        name = 'conv1'
        layer = {'type'='convolution', 'nFilters'=8, 'width'=7, 'height'=7,
                 'stride'=1, 'act'='relu'}
        srcLayers = {'data'};
    * A pooling layer follows the convolution layer to reduce the dimensionality of the image by half;
    deepLearn.addLayer /
        model = 'SimpleCNN'
        name = 'pool1'
        layer = {'type'='pooling', 'width'=2, 'height'=2,
                 'stride'=2, 'pool'='max'}
        srcLayers = {'conv1'};

    * Add the second convolution layer;
    deepLearn.addLayer /
        model = 'SimpleCNN'
        name = 'conv2'
        layer = {'type'='convolution', 'nFilters'=8, 'width'=7, 'height'=7,
                 'stride'=1, 'act'='relu'}
        srcLayers = {'pool1'};
    * A pooling layer follows the convolution layer to reduce the dimensionality of the image by half;
    deepLearn.addLayer /
        model = 'SimpleCNN'
        name = 'pool2'
        layer = {'type'='pooling', 'width'=2, 'height'=2,
                 'stride'=2, 'pool'='max'}
        srcLayers = {'conv2'};

    * Add a fully connected layer to flatten the image into one dimension;
    deepLearn.addLayer /
        model = 'SimpleCNN'
        name = 'fc1'
        layer = {'type'='fullconnect', 'n'=16, 'act'='relu'}
        srcLayers = {'pool2'};

    * Finally, add the output layer;
    deepLearn.addLayer /
        model = 'SimpleCNN'
        name = 'prediction'
        layer = {'type'='output', 'act'='softmax'}
        srcLayers = {'fc1'};

run;
```

Computer Vision in SAS

Convolutional Neural Networks

①

DAG for Simple_CNN:



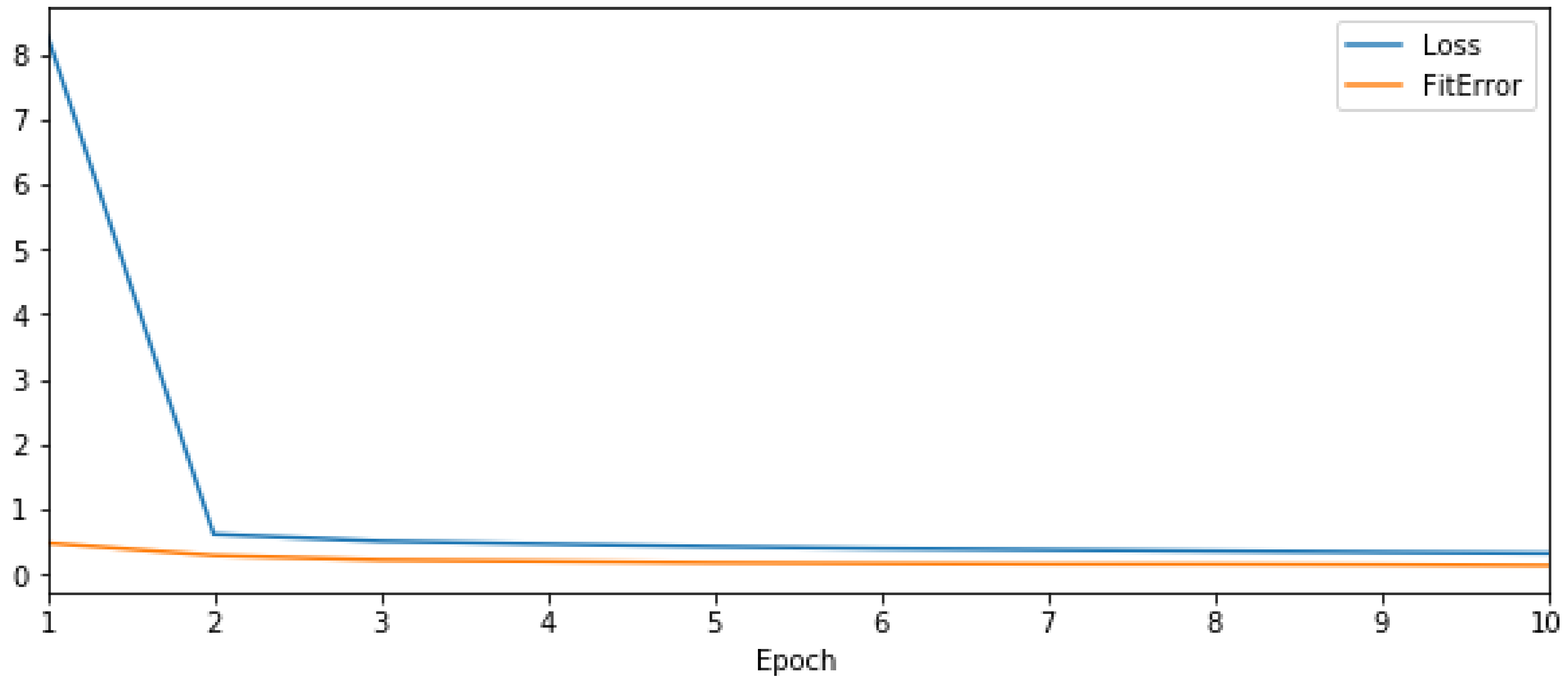

```
* Add the first convolution layer;
deepLearn.addLayer /
    model = 'SimpleCNN'
    name = 'conv1'
    layer = {'type'='convolution', 'nFilters'=8, 'width'=7, 'height'=7,
             'stride'=1, 'act'='relu'}
    srcLayers = {'data'};
* A pooling layer follows the convolution layer to reduce the dimensionality of the image by half;
deepLearn.addLayer /
    model = 'SimpleCNN'
    name = 'pool1'
    layer = {'type'='pooling', 'width'=2, 'height'=2,
             'stride'=2, 'pool'='max'}
    srcLayers = {'conv1'};
```

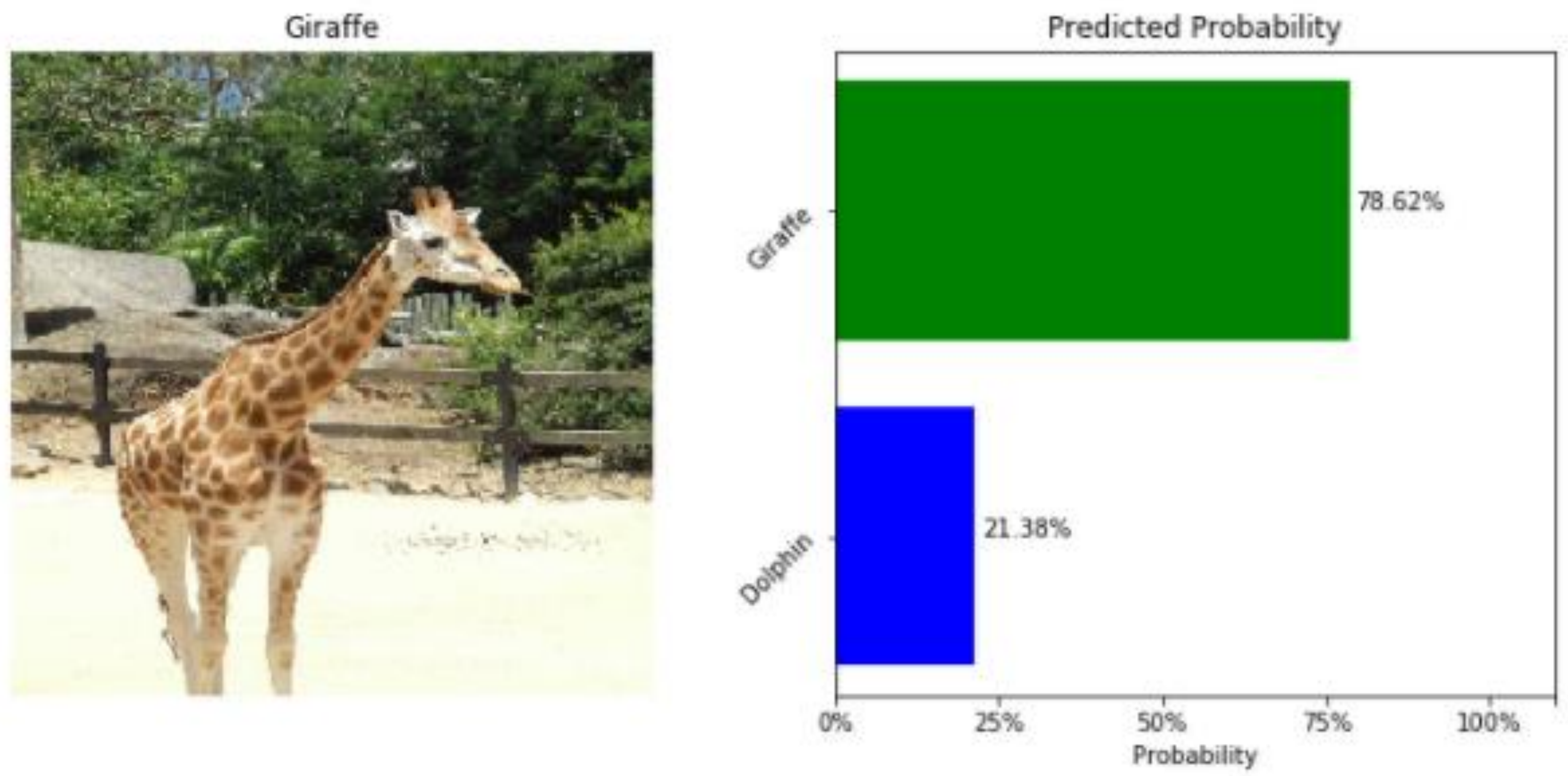
Computer Vision in SAS

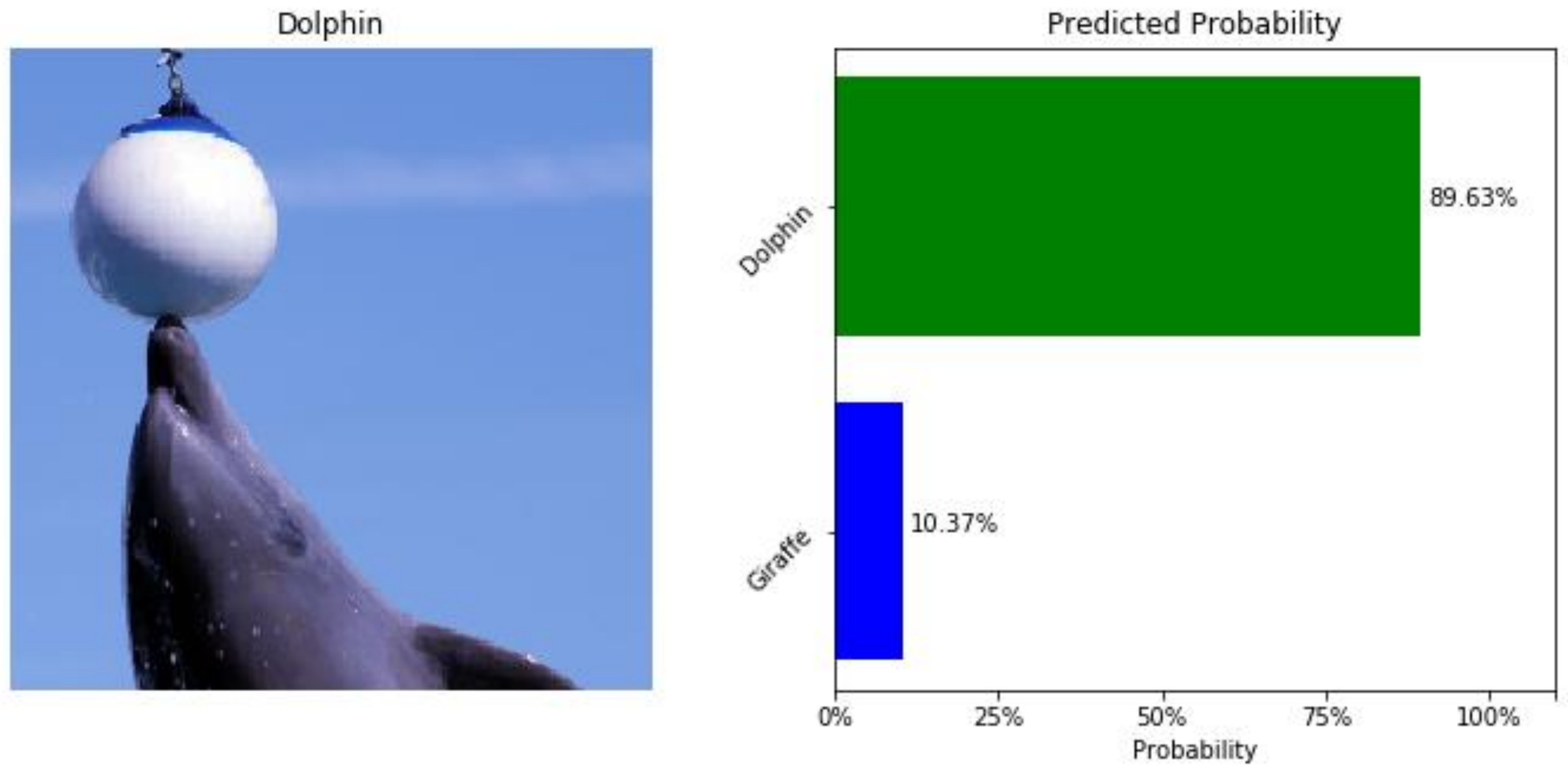
Convolutional Neural Networks

2

```
/* Train the model */
proc cas;
    session s;

    deepLearn.dlTrain /
      model = 'SimpleCNN'
      modelWeights = {name='SimpleCNN_weights', /* Model weights table */
                      replace=TRUE }
      table = {name='training_set'}            /* Training data */
      inputs = {'image'}                       /* Input variables */
      target = 'label';                        /* The target variable */
   run;
run;
```



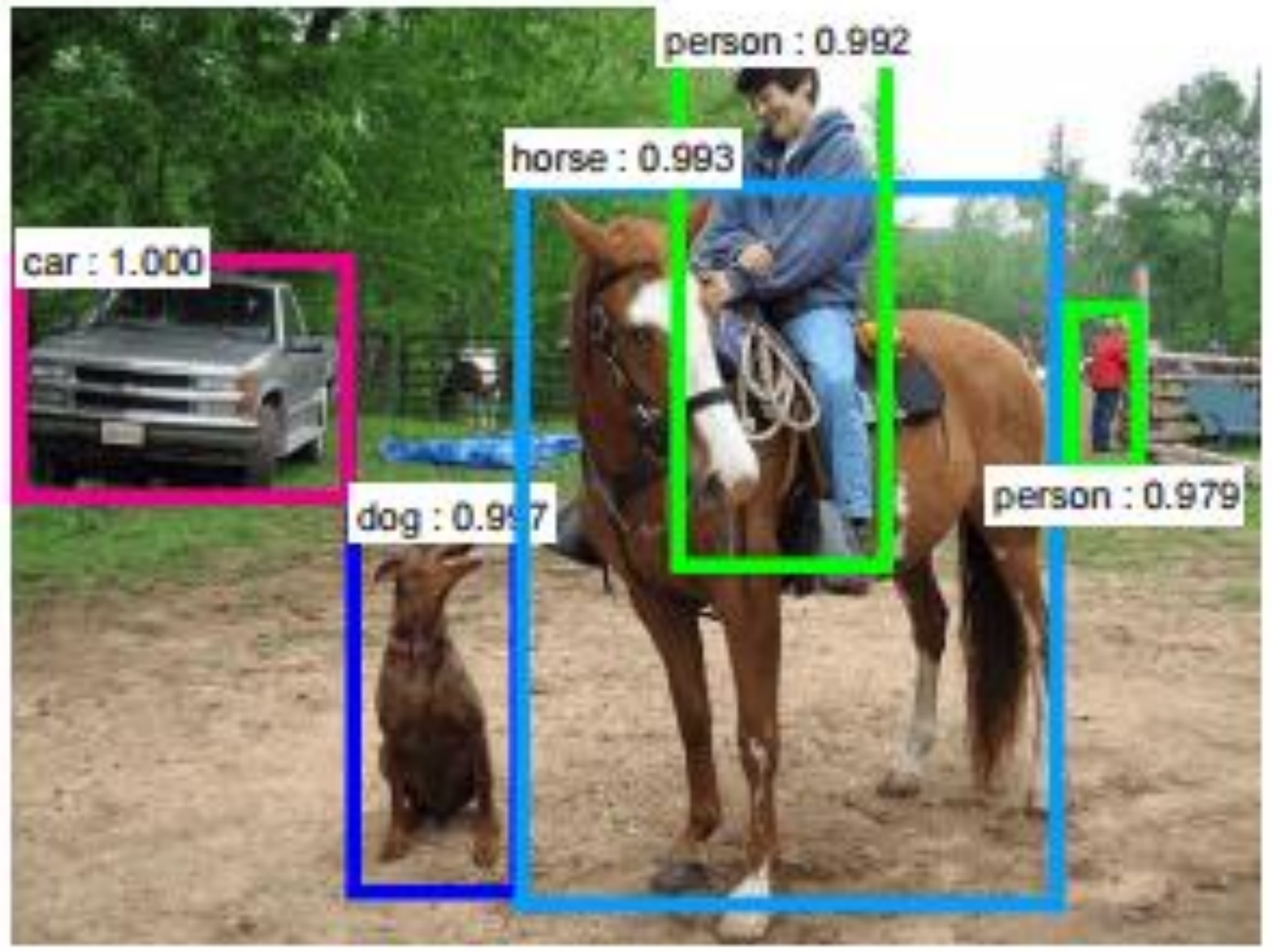


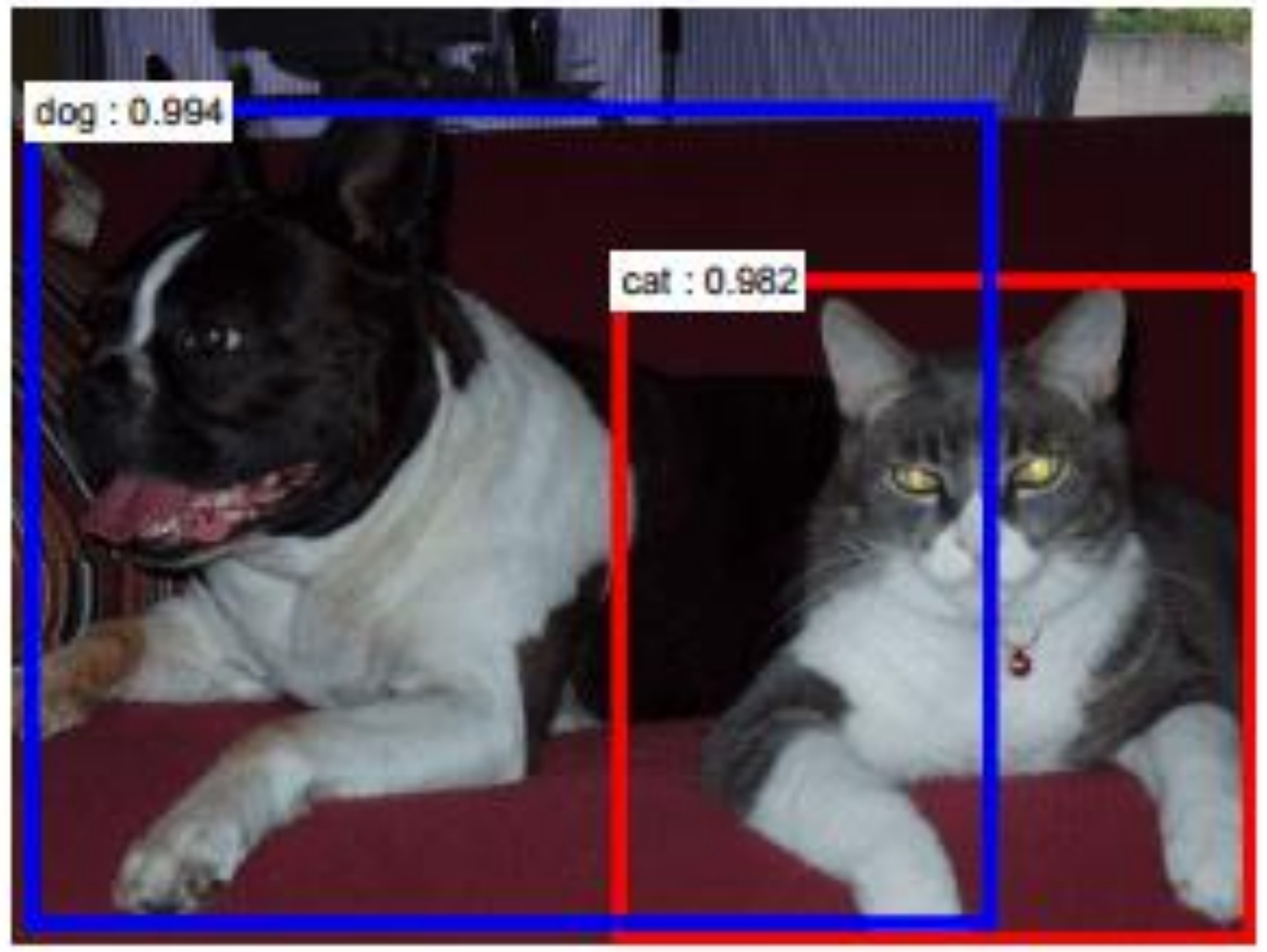


Computer Vision in SAS

Conclusion

- A new type of data is all ready to be processed in SAS
- SAS VDMML gives us access to the power and flexibility of Deep Learning



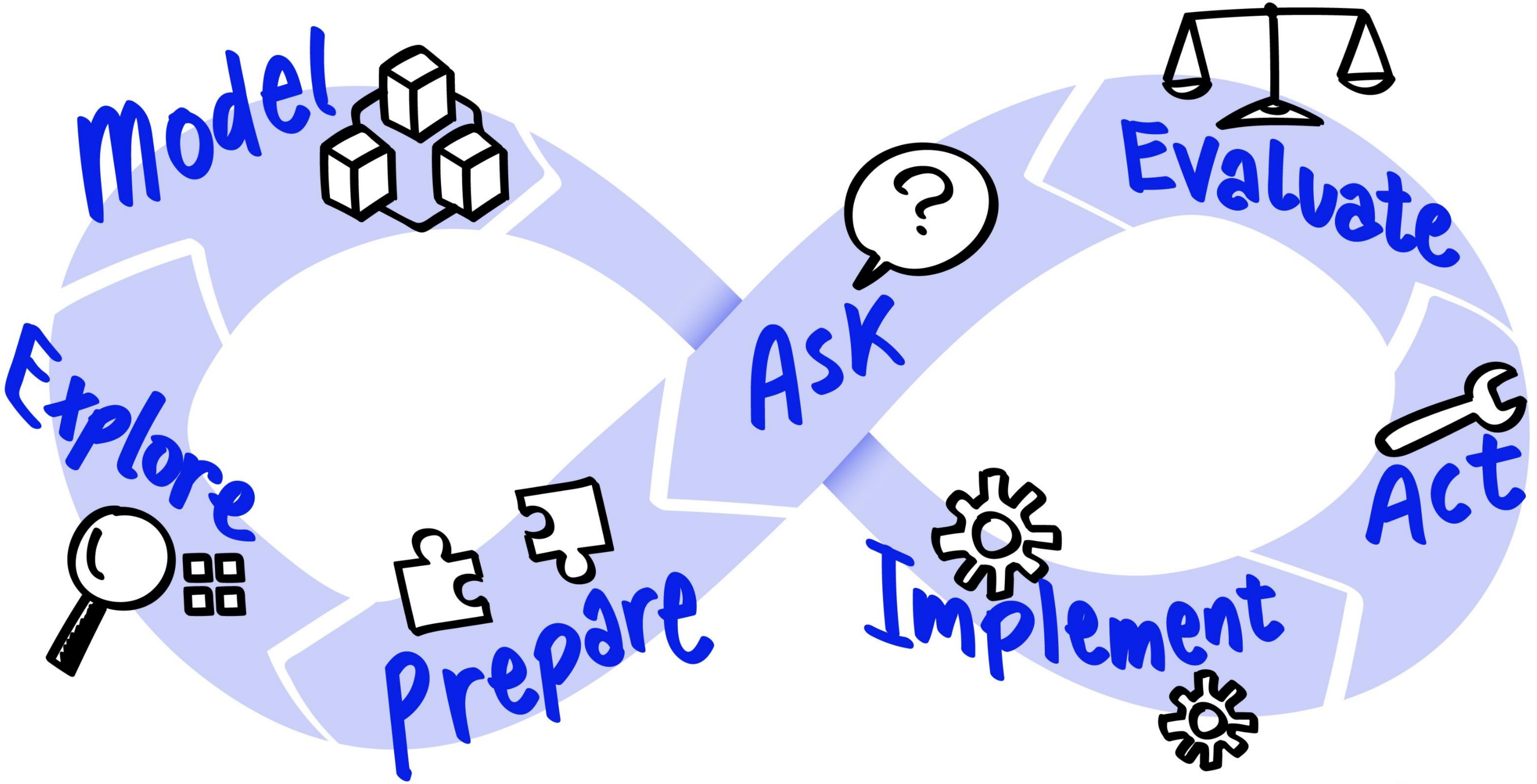


Data Science Jam Sessions by SAS

Talk of the GEEKS

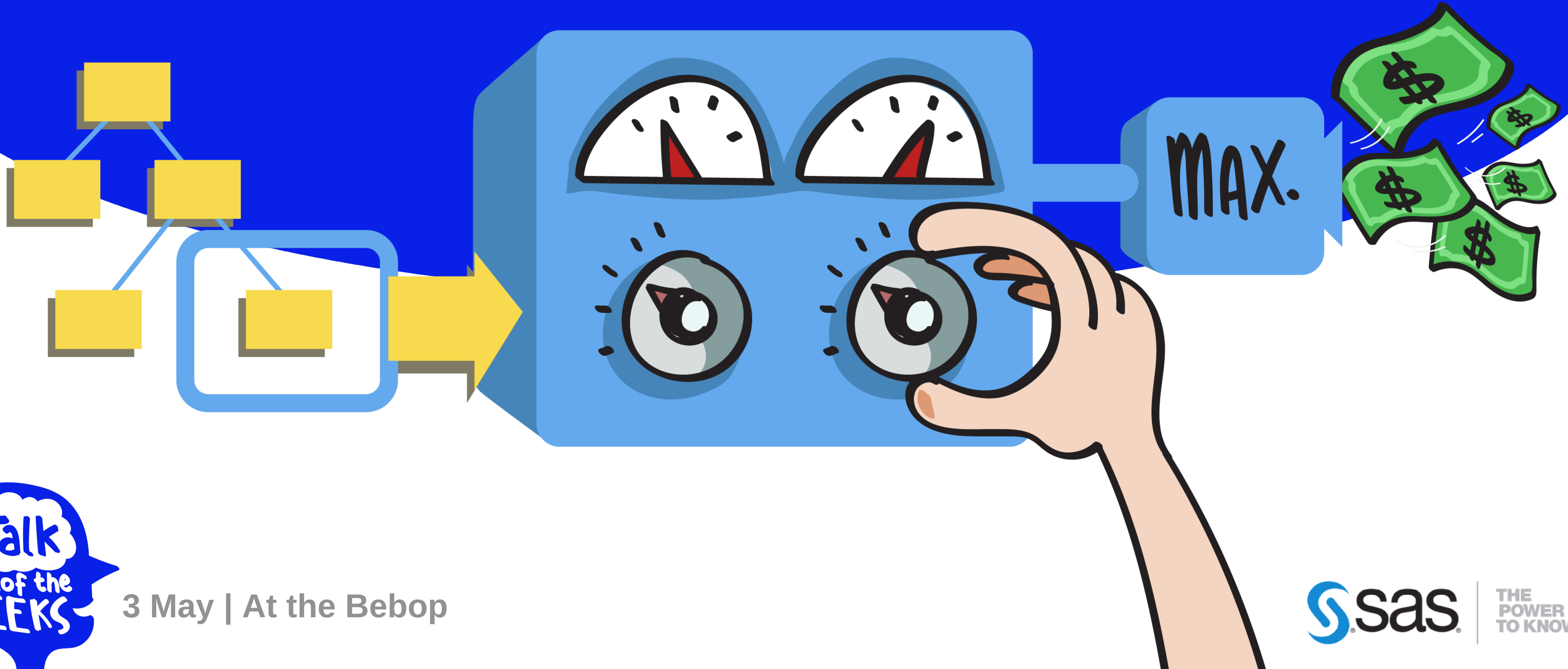
sas THE POWER TO KNOW.

Analytical Lifecycle



Make your analytics count! - from predictive to prescriptive analytics with SAS optimization

Speaker: Adriaan Van Horenbeek



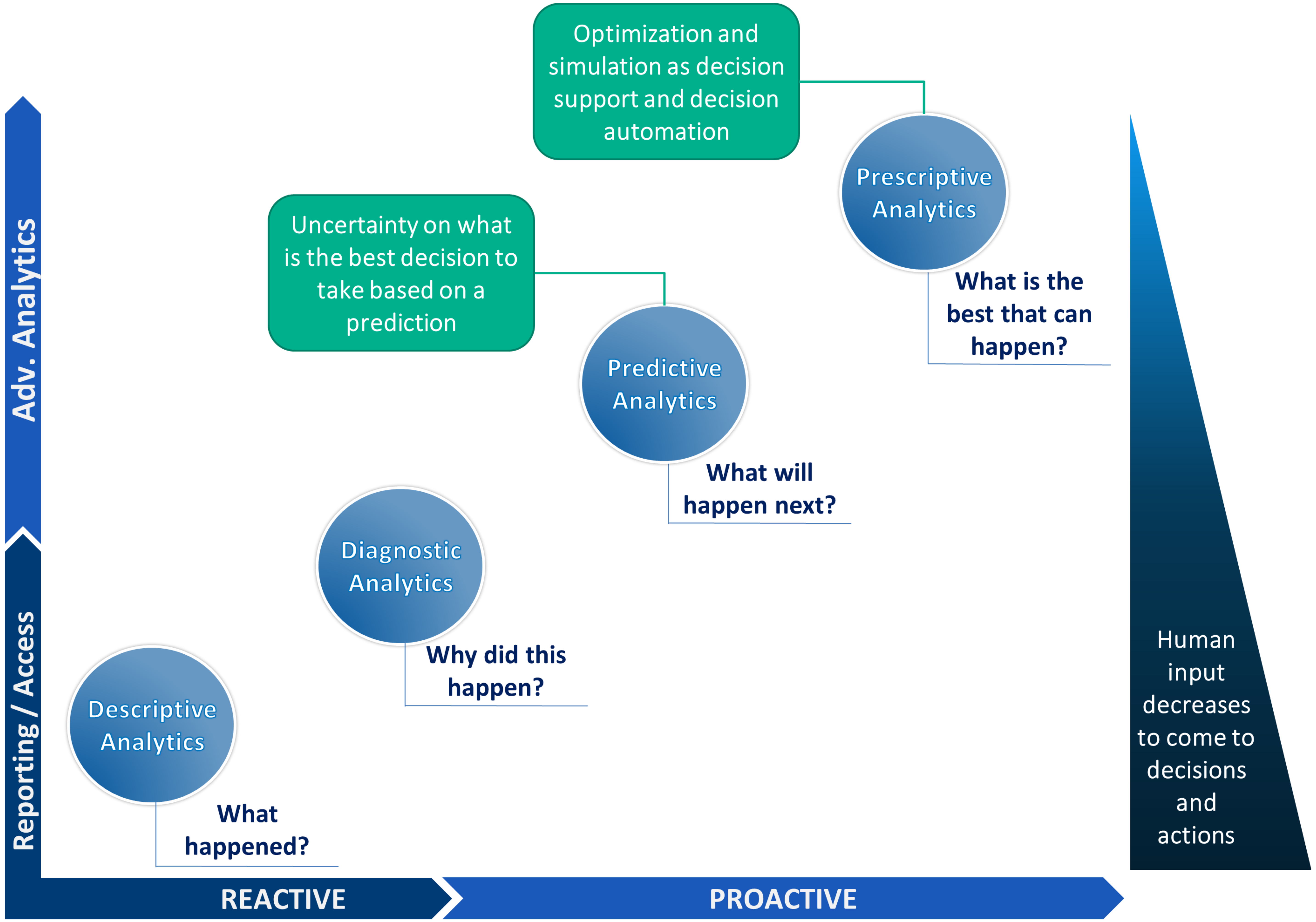
Talk of the GEEKS

3 May | At the Bebop

sas THE POWER TO KNOW.

From predictive to prescriptive analytics

The final frontier of analytic capabilities



From predictive to prescriptive analytics

The final frontier of analytic capabilities

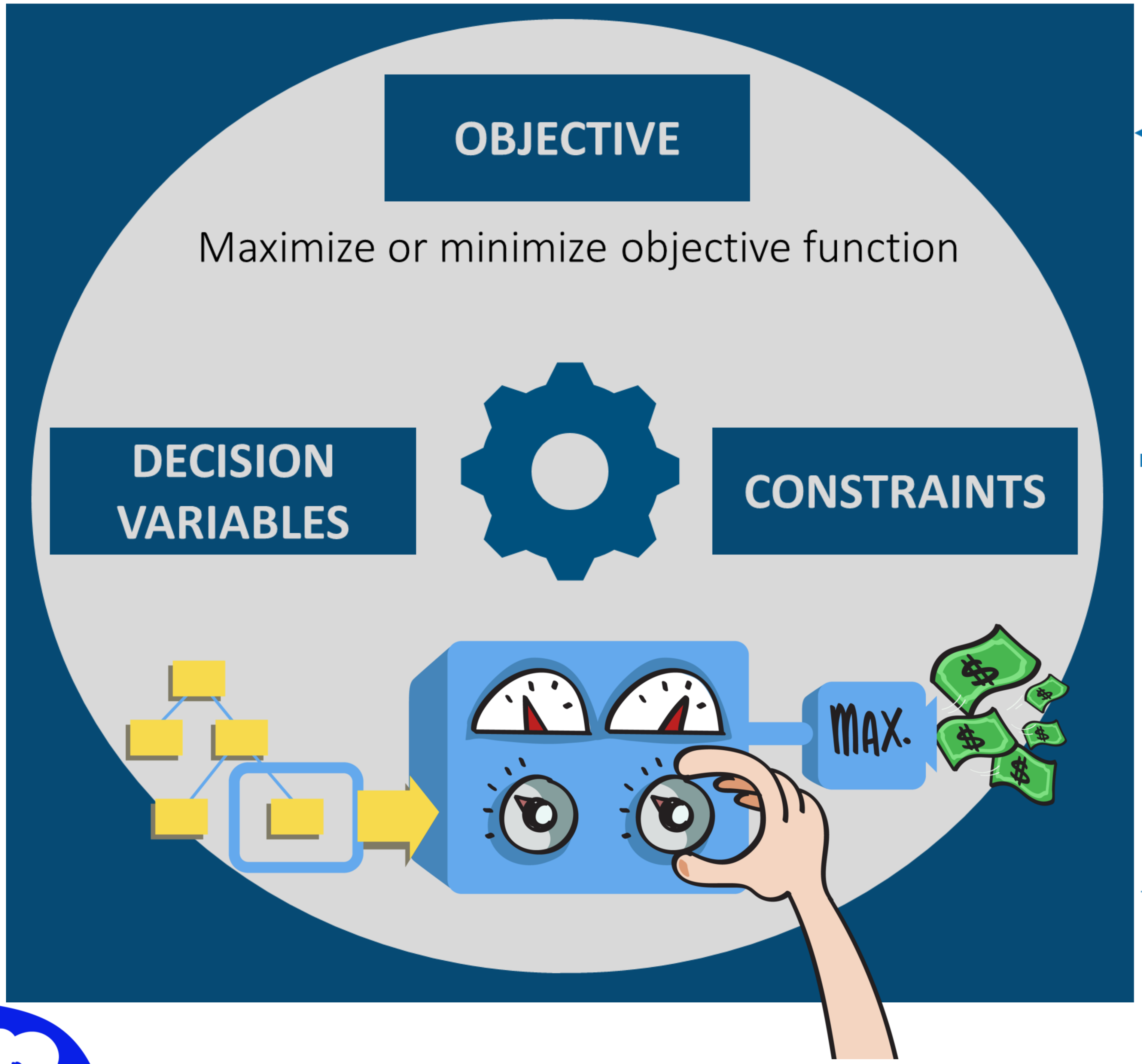
- **Prescriptive analytics** entails the application of mathematical and computational sciences and suggests decision options to *take advantage of the results of descriptive and predictive analytics*

  - It goes *beyond predictive analytics* by also suggesting actions to benefit from the predictions and showing the implications of each decision option

  - *Makes your predictive analytics actionable!*

From predictive to prescriptive analytics

The principle of optimization



DATA INPUTS
Variables, constraints, objective function…

RECOMMENDED ACTIONS
Decision variables to optimize objective

IMPLEMENTATION

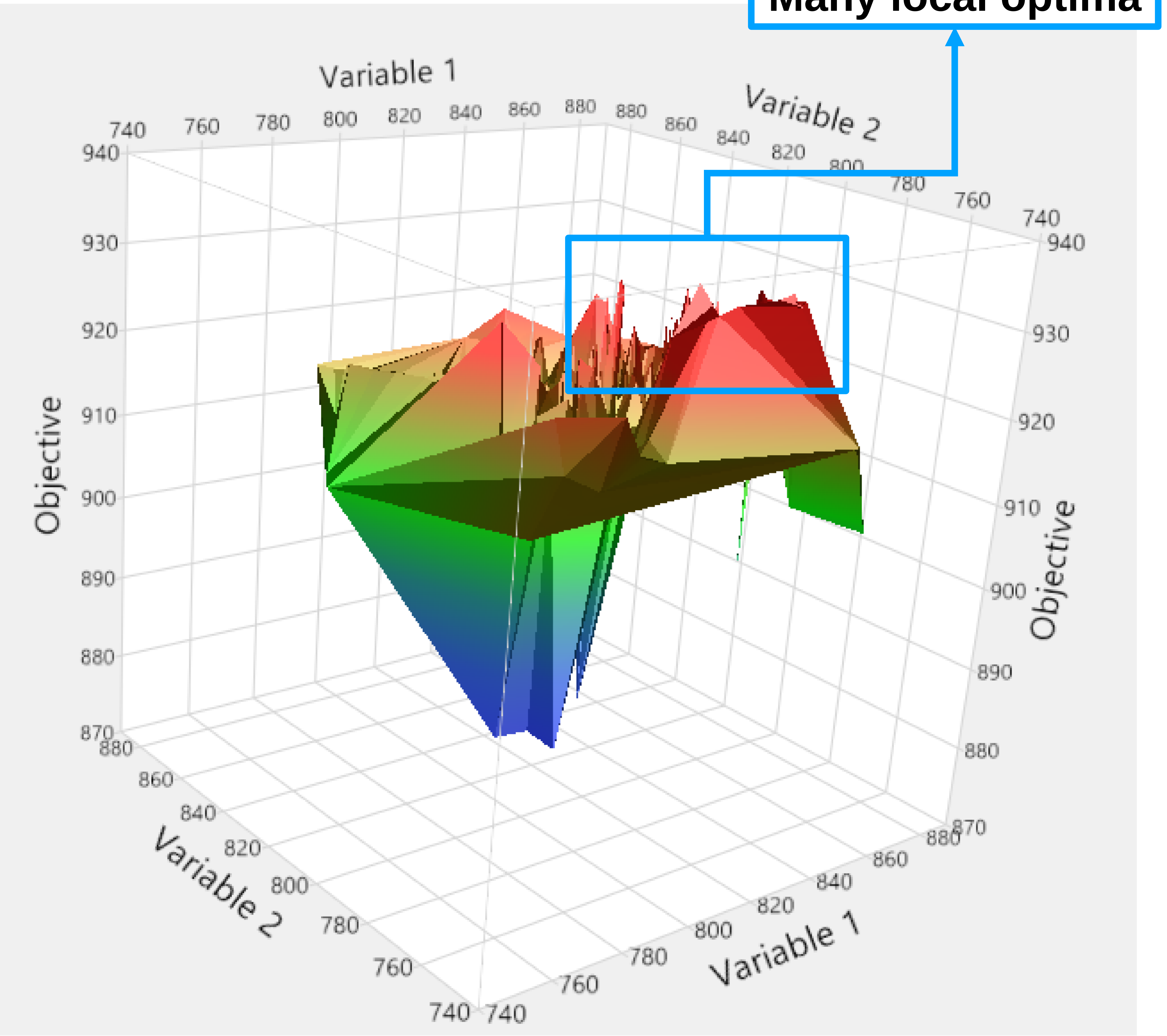
Measure results and adjust model

From predictive to prescriptive analytics

The challenge

- Using advanced predictive models as objective function or constraints in optimization is challenging:
  - Often these functions cannot be expressed in analytic closed form
  - Can be non-smooth, discontinuous and non-linear
  - Are computationally expensive to evaluate

Many local optima

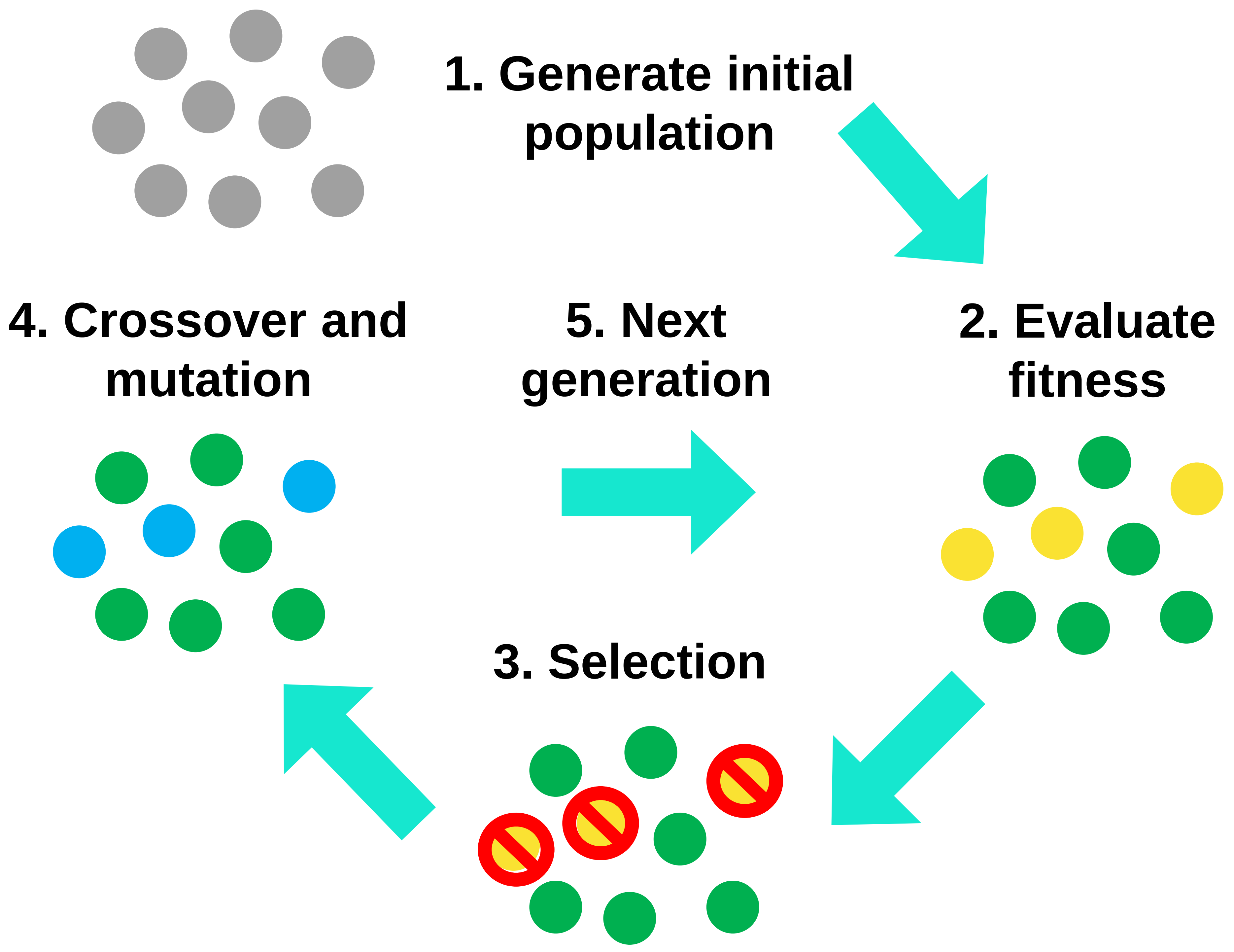


From predictive to prescriptive analytics
PROC OPTLSO to the rescue!

- The OPTLSO procedure performs optimization of general nonlinear functions that are defined by the FCMP procedure
  - In the FCMP function you can use the score code of an advanced predictive model
    - These functions do not need to be expressed in analytic closed form, can be non-smooth, discontinuous, and computationally expensive to evaluate
    - Uses global and local search algorithms in parallel and is based on a genetic algorithm (GA)

From predictive to prescriptive analytics
How does a GA work?

- GAs are a family of local search algorithms that seek optimal solutions to problems by applying the principles of natural selection and evolution
  - Can be applied to almost any optimization problem



From predictive to prescriptive analytics

PROC OPTLSO to the rescue!

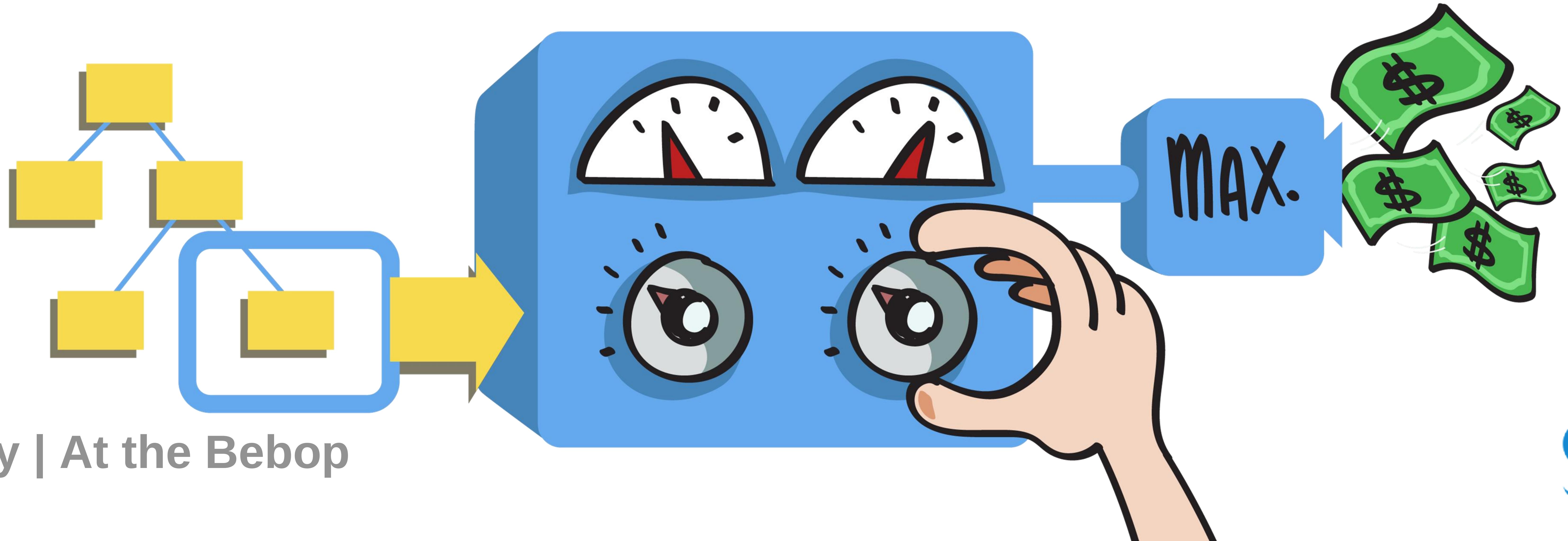
```
/* Define objective function in dataset */
data objdata;
length _function_ $ 15 _id_ $ 40;
input _id_ $ _function_ $ _sense_ $;
datalines; Predicted_Yield objective_function max;
run;

/* Define objective function by including score code from predictive model */
proc fcmp outlib=work.myfuncs.mypkg;
function objective_function(Decision_variable_1,Decision_variable_2,Decision_variable_3);
%include 'D:\Solvay Torrelavega POC\Trial scripts\score_ensemble_0304.sas';
return (Predicted_Yield);
endsub;

/* Use OPTLSO to perform optimization with
a machine learning model as objective function*/
proc optlso
primalout = solution                        Defines the output dataset with the best solution
variables = variable_limits                 Dataset that stores the decision variable names and bounds
objective = objdata                         Names the FCMP functions to be used as the objective
lincon = lincondata                         Describes the linear constraints
nlincon = nlincondata;                      Describes the nonlinear constraints
run;
```

Prescriptive analytics is the new predictive analytics

- SAS optimization possesses *advanced capabilities to use predictive models in optimization*

- Go *beyond predictive analytics* by also suggesting actions to benefit from the predictions

- Prescriptive analytics *makes your predictive analytics actionable!*

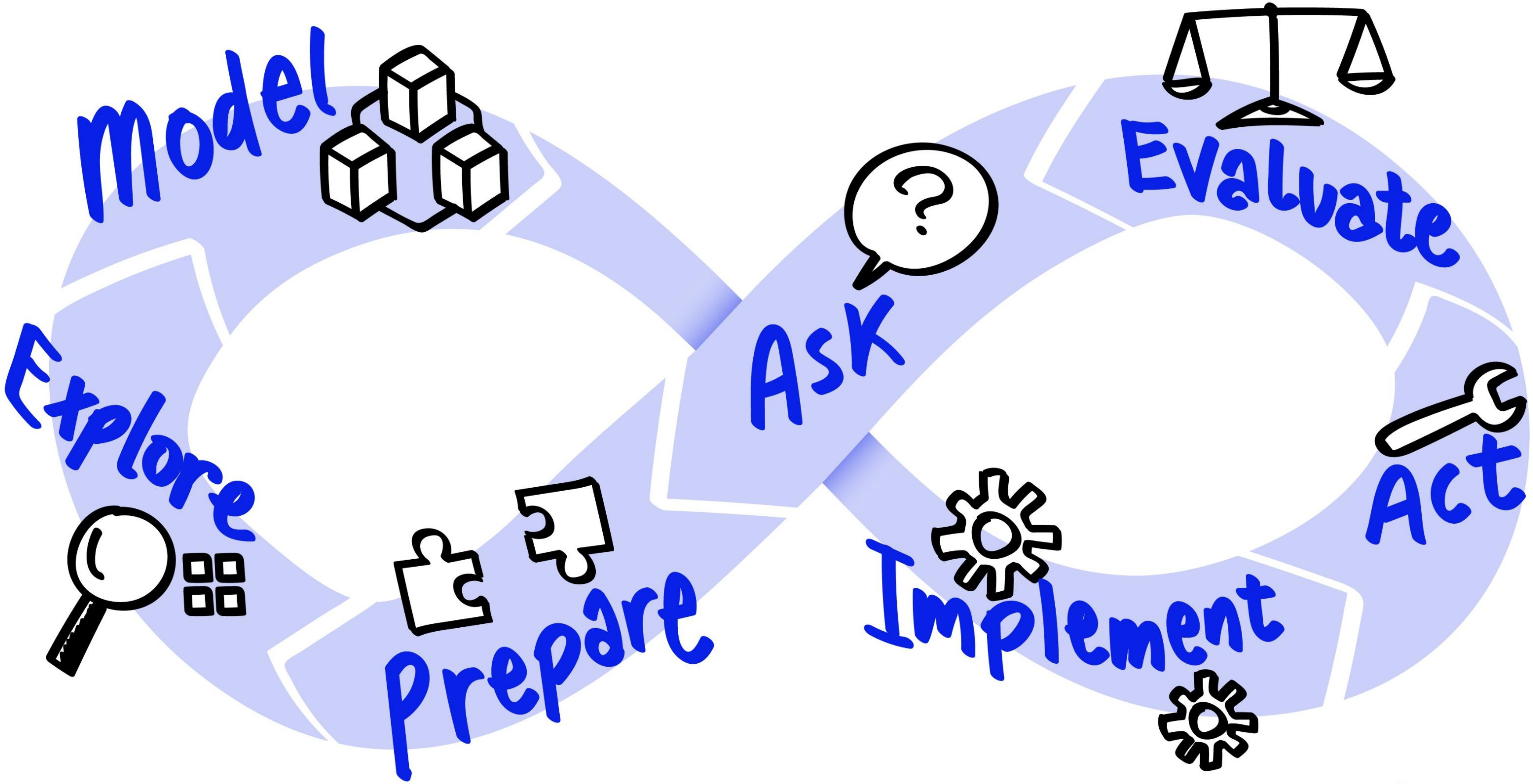


Data Science Jam Sessions by SAS

Talk of the GEEKS

sas THE POWER TO KNOW.

Analytical Lifecycle



Efficient scoring with PROC ASTORE

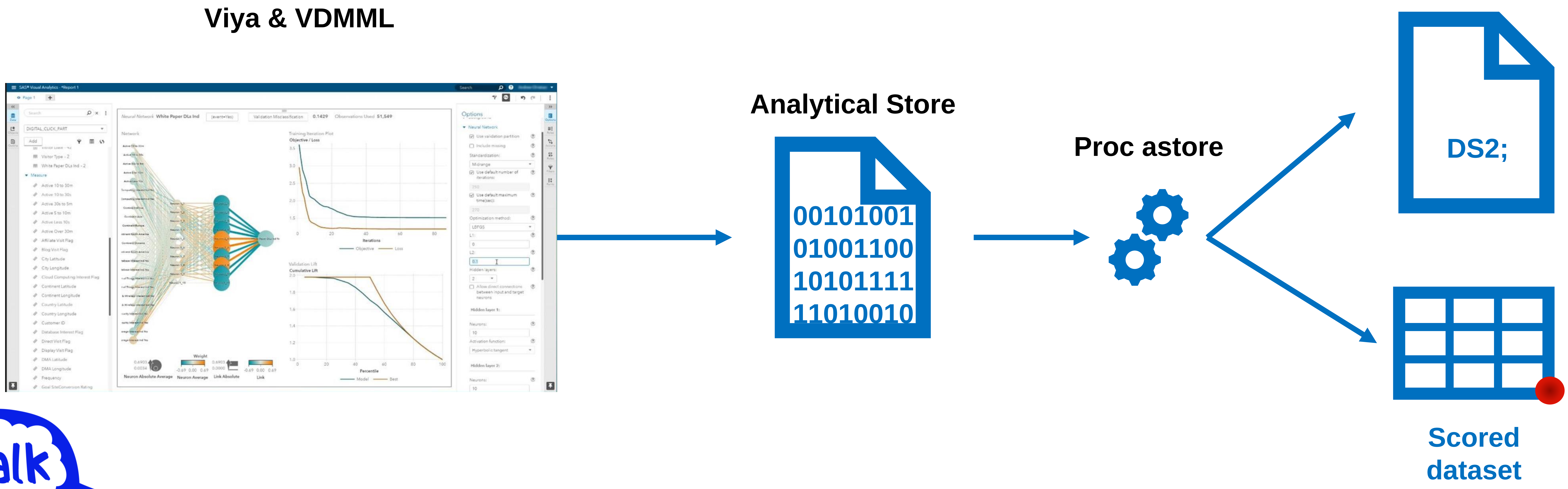
Speaker: Florian Bertrand

3 May | At the Bebop

Proc ASTORE

What?

Model deployment



Proc ASTORE
What?

- A procedure to describe and manage analytical stores and to use them to score new data

- Analytical store = a binary file which captures the state of a predictive model

- Analytical store can be created with a SAVESTATE statement from an analytic procedure

- Currently available with FACTMAC, FOREST, SVMACHINE, GRADBOOST, TEXTMINE, SVDD and STFT. Many more will follow soon

Proc ASTORE
Why?

- Some models generate huge data step code (up to >50M lines of code!)
  - Gradient Boosting, Random Forests, factorization machines etc.
  - Produces insufficient memory conditions during compiling
  - Ex : NETFLIX ($1M model never implemented)
- Transportable: use between databases/hadoop

Proc ASTORE
For what?

- Score new data in CAS
- Create DS2 scoring code out of the trained model
- Deploy models in ESP (Event Stream Processor)
- Move analytical store from local machine to/from CAS
  - Possibility for the use to add preprocessing (eg. input variable transformation…) and postprocessing code (eg. Decisions based on prediction…)

Proc ASTORE

Method

- First create a model with an anlytic procedure (here a support vector machine)
- Create the analytical store with a savestate statement
- Use proc astore with the analytical store to:
  - Score new data
  - Generate DS2 scoring code

```
proc svmachine data=mycas.myDataset;
input myVar1 myVar2 myVar3 /level=interval;
target myTarget;
id id;
savestate rstore=mycas.myAnalyticalStore;
run;
```

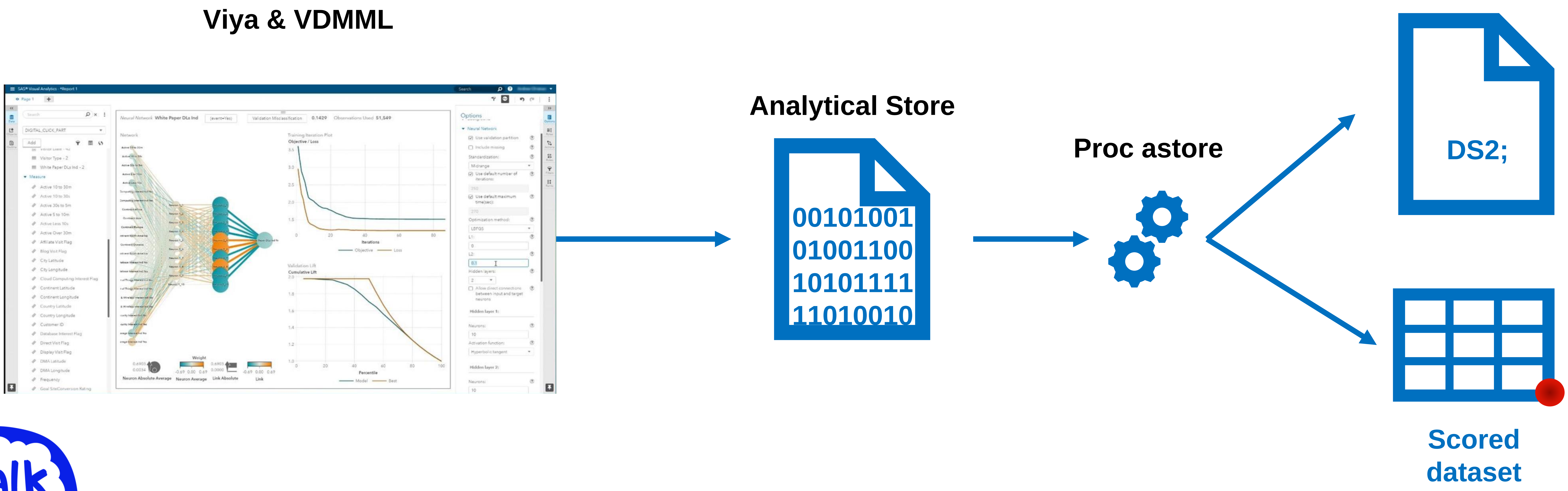
```
proc astore;
score data=mycas.toScore
out=mycas.scored
rstore=mycas.myAnalyticalStore ;
quit;
```

```
proc astore;
describe rstore=mycas.myAnalyticalStore
epcode='D:\data\myScoreCode.sas';
quit;
```

Proc ASTORE

Conclusion

- Proc astore offers a high flexibility way to efficiently score new data

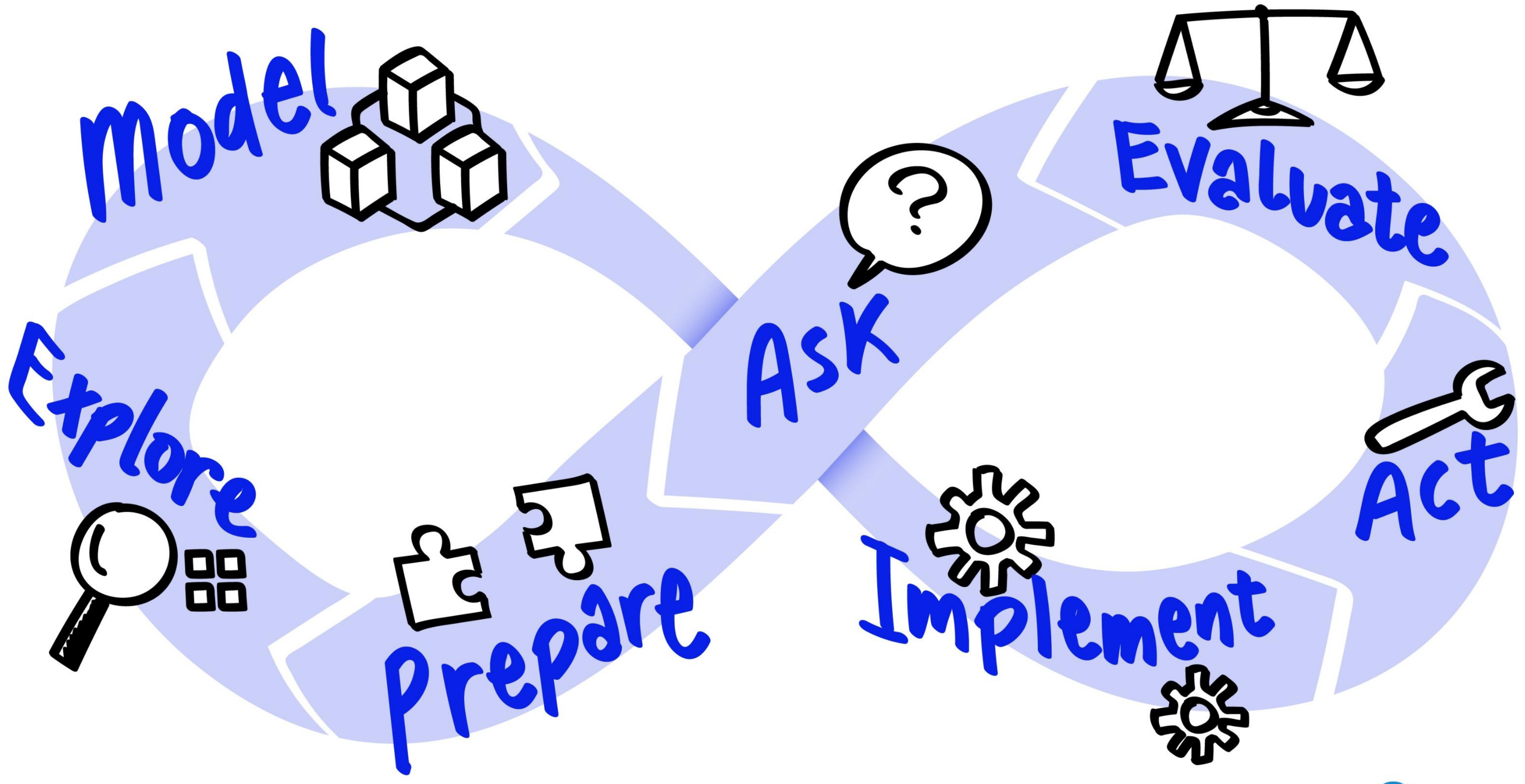


Data Science Jam Sessions by SAS

Talk of the GEEKS

sas | THE POWER TO KNOW.

Analytical Lifecycle



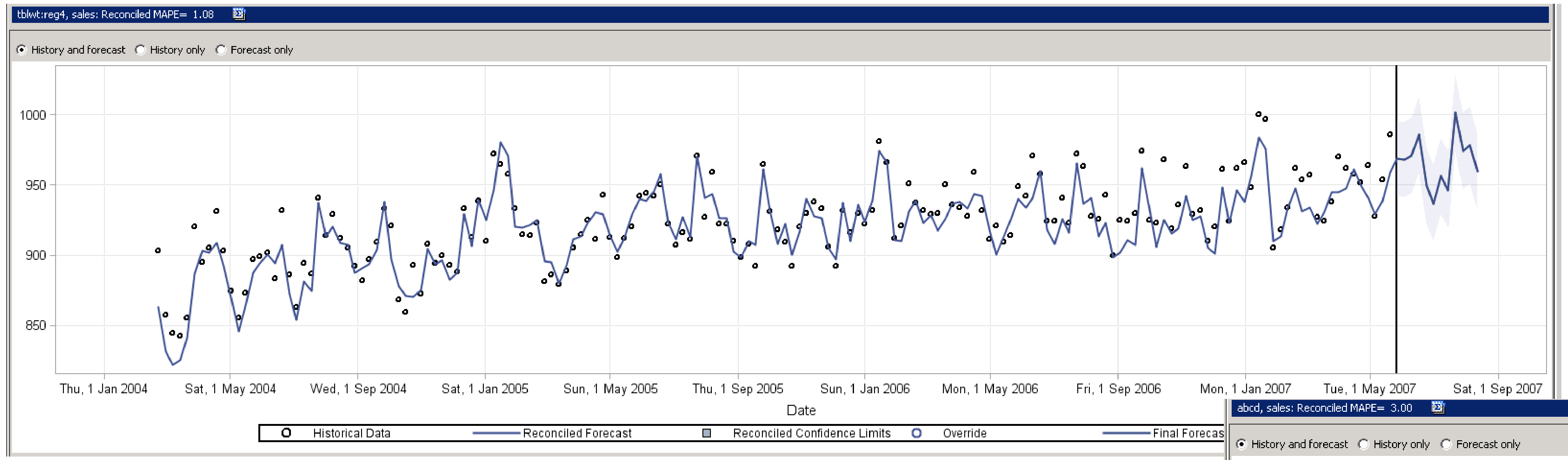
Forecast Exception Reporting

Speaker: Elke Potums

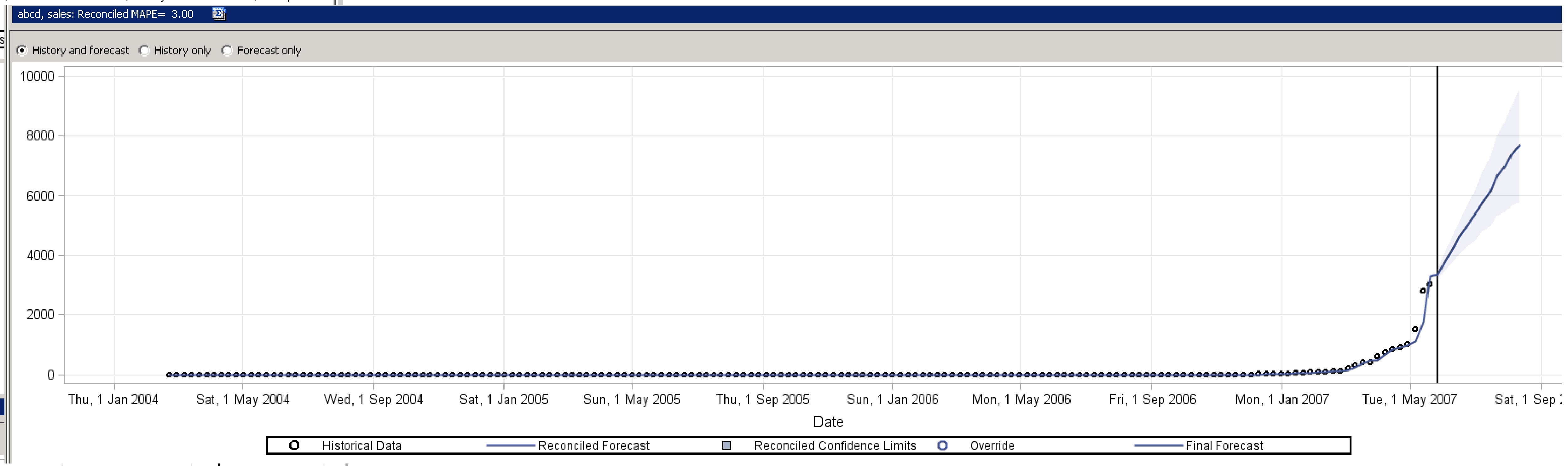
3 May | At the Bebop

Forecast Exception Reporting

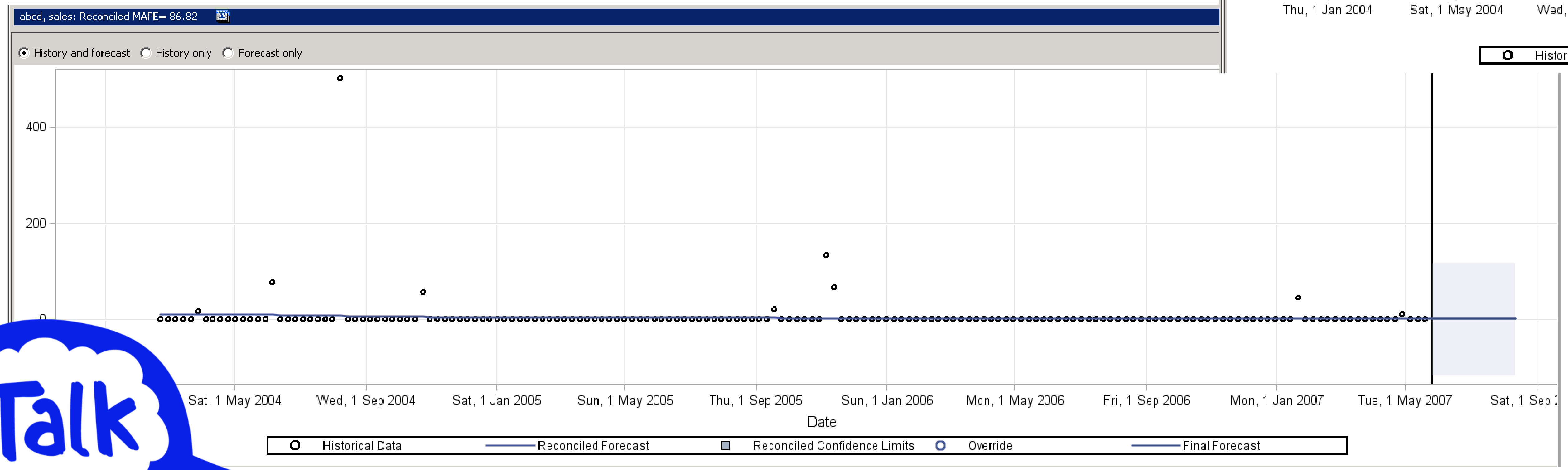
What?



80%



10%



10%

Forecast Exception Reporting
Why?

?Dilema: These Forecast Sever projects contain thousands upon thousands of forecasts – possibly over **100.000 forecasts**

- ➢ Even if only **10%** need attention … that's over **10.000 forecasts!**

➢Who among us has *time* to go through them one by one?

Forecast Exception Reporting
Method

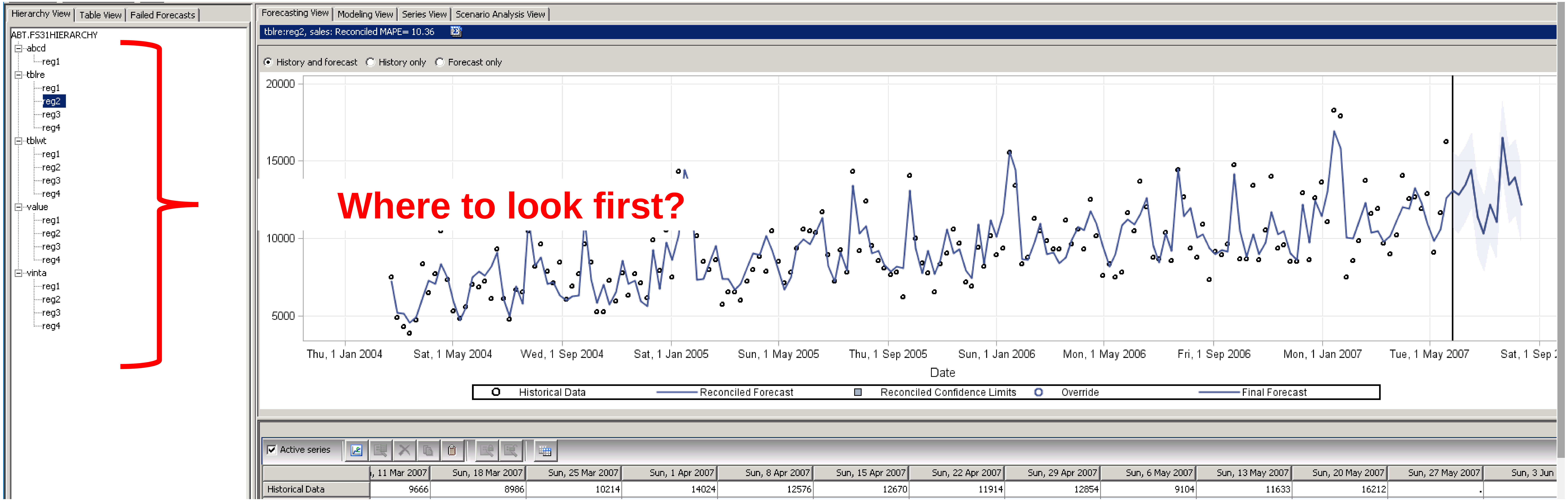
1. Define what constitutes “forecast exceptions” for the business user community
2. Develop business rules, methodology, and computations required to identify and flag those exceptions
3. Utilize information about the FS project to easily create these exception reports

→ %fsload macro

- &HPF_NUM_LEVELS
- &HPF_PROJECT_LOCATION
- &HPF_BYVARn
- &HPF_LEVEL_BYVARSm
- &HPF_RECONCILE_BYVAR

Forecast Exception Reporting

Method



Forecast Exception Reporting

Method

```
%macro fs_exception_reports(
                                userid =,
                                pwd =,
                                sasenv = default,
                                midtier = fsmain,
                                fsprojname =,
                                env = Default,
                                outlibname =,
                                high_exception_thold =,
                                outfilename =,
                                server = ,
                                port = 9621
                        );


   %fslogin(DESKTOP = NO,
       USER = &userid,
       PASSWORD = &pwd,
       SASENVIRONMENT = &sasenv,
       MIDTIER = &midtier
       );

   %fsload(
       PROJECTNAME = &fsprojname,
       ENVIRONMENT = &env,
       MIDTIER = &midtier
       );

   %fslogout(MIDTIER = &midtier);


       /* Check the number of levels in the forecast hierarchy and assign &start macro variable */

       %if &HPF_NUM_LEVELS > 1 %then %let start=2;
       %else %let start=1;
```

%FSLOAD

```
%do lev = &start %to &HPF_NUM_LEVELS;
    %if &HPF_NUM_LEVELS > 1 %then %let bynum = %eval(&lev -1);
    %else %let bynum = 1;
    %put bynum=&bynum;

    data _null_;
        call symput("sqlbyvar&lev",tranwrd("&&HPF_LEVEL_BYVARS&lev", ' ', ', '));
    run;

    %put sqlbyvar&lev=&&sqlbyvar&lev;
    %put HPF_BYVAR&bynum=&&HPF_BYVAR&bynum;
    %if &HPF_NUM_LEVELS > 1 %then libname fcst "&HPF_PROJECT_LOCATION.hierarchy/&&HPF_BYVAR&bynum";
    %else libname fcst "&HPF_PROJECT_LOCATION.hierarchy/leaf";
    ;

    /* Extract Forecacst MAX from FINALFOR data set */

    proc univariate data=fcst.finalfor noprint;
        by &&HPF_LEVEL_BYVARS&lev;
        var predict;
        output out=work.fcst_max_&lev max=fcst_max;
    run;

    /* If current level is the reconcile level, then use OUTFOR data set otherwise use RECFOR data set */

    %if "&&HPF_BYVAR&bynum" = "&HPF_RECONCILE_BYVAR" %then %do;

        /* Extract Historical MAX from OUTFOR data set */

        proc univariate data=fcst.outfor noprint;
            by &&HPF_LEVEL_BYVARS&lev;
            where actual ne .;
            var actual;
            output out=work.hist_max_&lev max=hist_max;
        run;

    %end;
```

Forecast Exception Reporting

Method

```
%else %do;

    /* Extract Historical MAX from RECFOR data set */

    proc univariate data=fcst.recfor noprint;
        by &&HPF_LEVEL_BYVARS&lev;
        where actual ne .;
        var actual;
        output out=work.hist_max_&lev max=hist_max;
    run;

%end;

/* Join Historical and Forecast MAX data sets, compute ratio of fcst_max to hist_max
    and set exception flag variable if ratio > 2 */

data work.exception_list_tmp_&lev;
    merge work.hist_max_&lev (in=one) work.fcst_max_&lev(in=two);
    by &&HPF_LEVEL_BYVARS&lev;
    if fcst_max <= 1e-4 then fcst_max = 0;
    if hist_max <= 1e-4 then hist_max = 0;
    if hist_max > 0 then ratio = fcst_max / hist_max;
    else ratio = .;
    length exception $35.;
    if ratio > &high_exception_thold then exception = 'Forecast to Historical Ratio High';
*   else if ratio < &low_exception_thold then exception = 'Forecast to Historical Ratio Low';  */
    else exception = '';
run;

/* Select only items flagged as exceptions */

data &outlibname..exception_list_&lev;
    set work.exception_list_tmp_&lev;
    by &&HPF_LEVEL_BYVARS&lev;
    where exception ne '';
run;
```

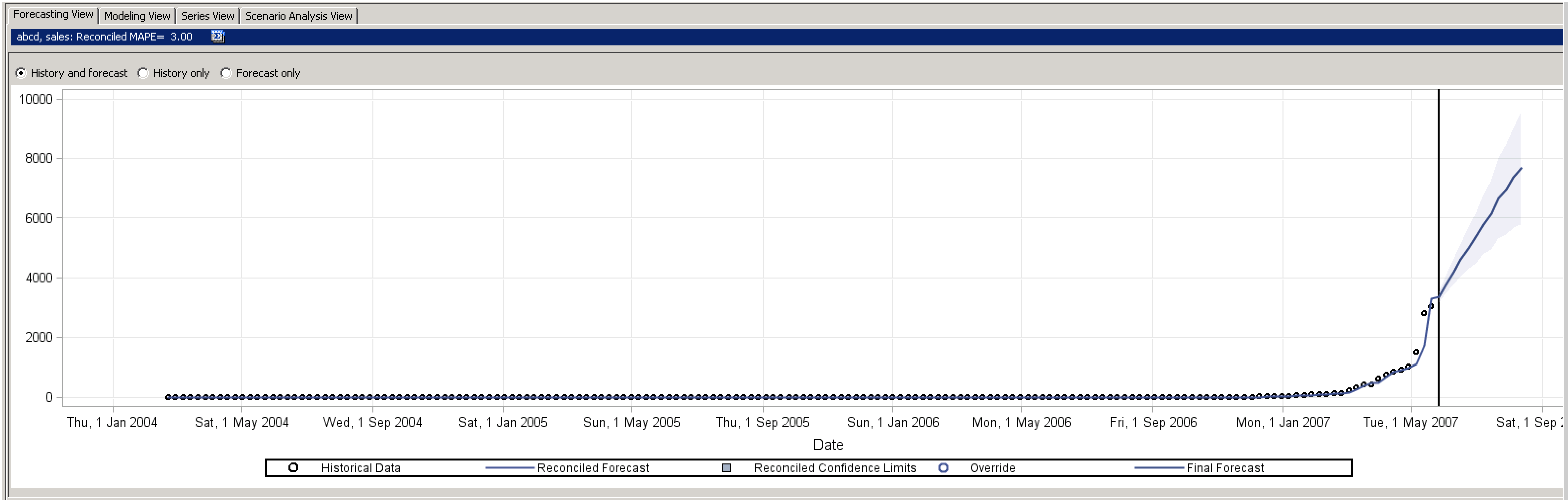
Define exception rules

```
    /* Sort by Exception type, descending ratio value, and &&HPF_LEVEL_BYVARS&lev */

    proc sort data=&outlibname..exception_list_&lev;
        by exception descending ratio &&HPF_LEVEL_BYVARS&lev;
    run;

    /* Delete temporary work files */

    proc datasets lib=&outlibname nolist;
        delete
            fskeydat
            fcst_max_&lev
            hist_max_&lev
            exception_list_tmp_&lev
        ;
    quit;

  %end;
%end;

%mend fs_exception_reports;
```

Forecast Exception Reporting

Method

	type	hist_max	fcst_max	ratio	exception
1	abcd	3015	7674.5893436	2.5454691024	Forecast to Historical Ratio High



Active series

	...n, 4 Mar 2007	Sun, 11 Mar 2007	Sun, 18 Mar 2007	Sun, 25 Mar 2007	Sun, 1 Apr 2007	Sun, 8 Apr 2007	Sun, 15 Apr 2007	Sun, 22 Apr 2007	Sun, 29 Apr 2007	Sun, 6 May 2007	Sun, 13 May 2007	Sun, 20 May 2007	Sun, 27 May 2007	Sun, 3 Jun...
Historical Data	210	315	410	430	600	731	832	900	1000	1511	2801	3015	.	
Forecast Model	132.61381672	247.32419271	374.02107773	480.58246598	484.45953069	691.31914024	835.11100236	935.15685227	991.9226447	1094.4678542	1738.5399702	3368.366615	**3470.5436966**	**3925.734...**
Reconciled Forecast	134.37448581	249.27763558	375.40971146	478.77755975	481.35172873	685.41688204	830.46602646	925.7023384	990.26670379	1096.314003	1729.349081	3294.7933867	**3357.5979507**	**3774.164...**
Override														
Lock Override														
Final Forecast	.	.	.	.	.	.	.	.	.	.	.	.	**3357.5979507**	**3774.164...**

View forecast model details...

Forecast Exception Reporting

Conclusion

- Built an *efficient* exception report structure →%fsload macro
- Exception reporting macro is *flexible* and *robust*
- Execution time is very *fast*

Data Science Jam Sessions by SAS

Talk of the GEEKS

sas THE POWER TO KNOW.

Analytical Lifecycle