Machine Learning to identify rats with higher risk of Alcohol use Disorder

Dr. Kshitij Jadhav,
Post doctoral fellow,
Lab on the Neurobiology of Addictive and Eating Disorders,
CHUV-Department of Psychiatry,
Universite de Lausanne,
Switzerland

Introduction

Introduction

1] Lever pressing for ethanol in the absence of ethanol

2] Progressive ratio paradigm

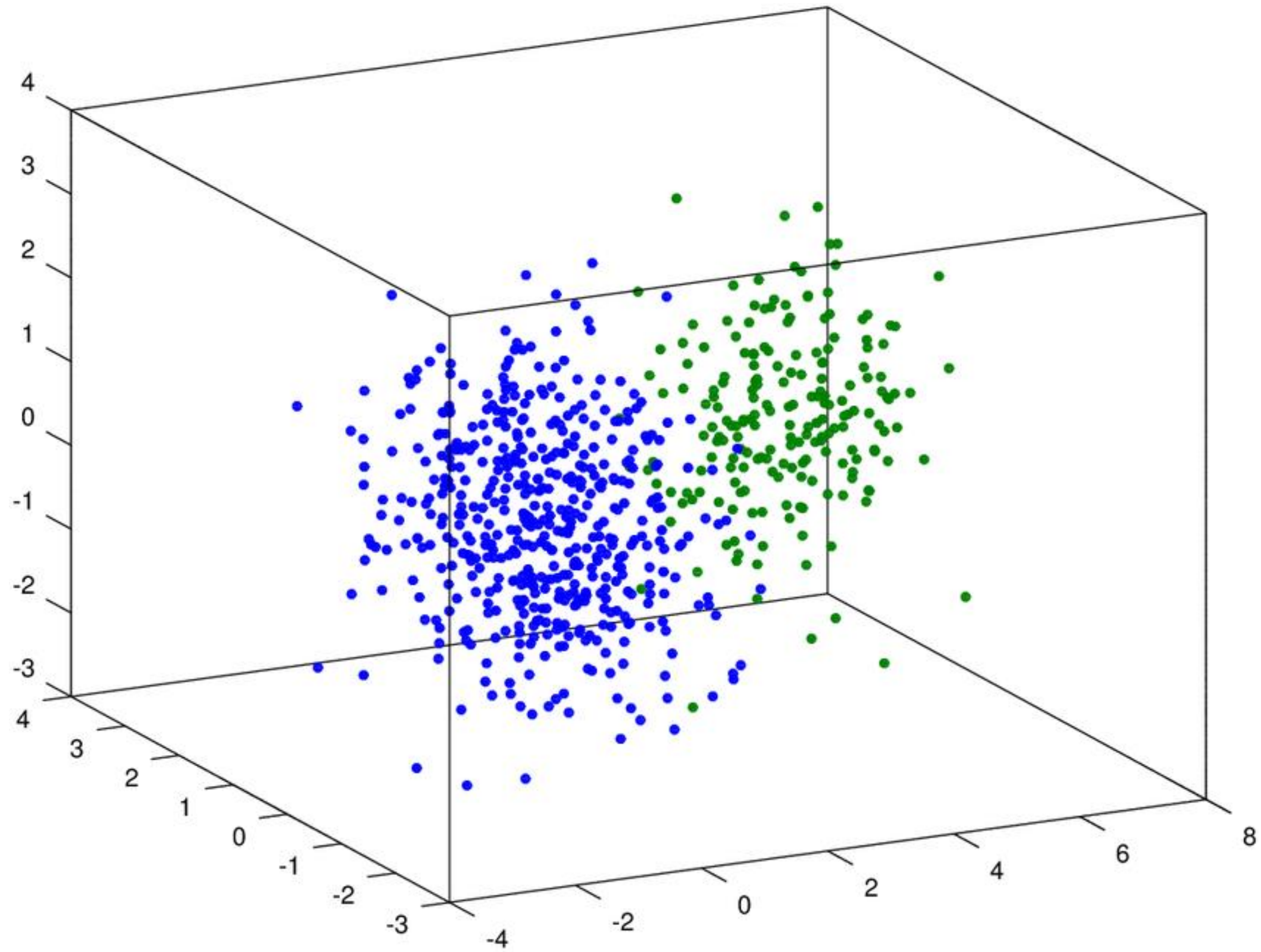
3] Compulsive alcohol taking in presence of shock

Introduction

RAT ID	Lever pressing in absence of ethanol	Progressive ratio	Compulsivity in presence of shock
1	5.67	27.33	10
2	15.33	54	41.67
3	9.33	40.33	7.67
4	11.67	37.67	3.67
5	4.33	28.67	8
6	5.67	21	5.33
7	3.33	28.33	6
8	9.67	68	44.33
9	14	49.33	81.67
10	14.33	46	13
11	13.33	55	15.67
12	15.67	56.67	9.33
13	5.33	35	3
14	15.33	56	28.67
15	8.33	28.33	11
16	4.67	39	1.67
17	9.33	28	9
18	2.33	23.67	3.33

Introduction

- But we also know….
- Not all individuals show similar vulnerability to develop addiction
- So, we have to cluster these rats based on their similarities



What I usually do?

RAT ID	Lever pressing in absence of ethanol	Progressive ratio	Compulsivity in presence of shock
1	5.67	27.33	10
2	15.33	54	41.67
3	9.33	40.33	7.67
4	11.67	37.67	3.67
5	4.33	28.67	8
6	5.67	21	5.33
7	3.33	28.33	6
8	9.67	68	44.33
9	14	49.33	81.67
10	14.33	46	13
11	13.33	55	15.67
12	15.67	56.67	9.33
13	5.33	35	3
14	15.33	56	28.67
15	8.33	28.33	11
16	4.67	39	1.67
17	9.33	28	9
18	2.33	23.67	3.33

Rats in top 33% are positive for that behaiour

What I usually do?

0 Criteria

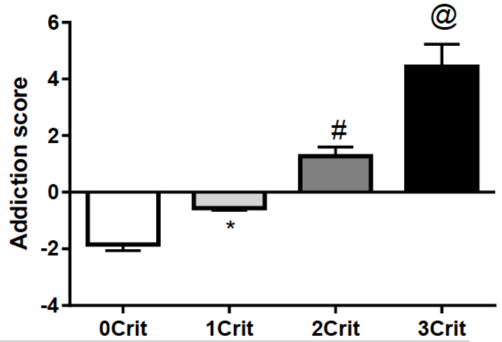
1 Criteria

Resilient

2 Criteria

3 Criteria

Vulnerable

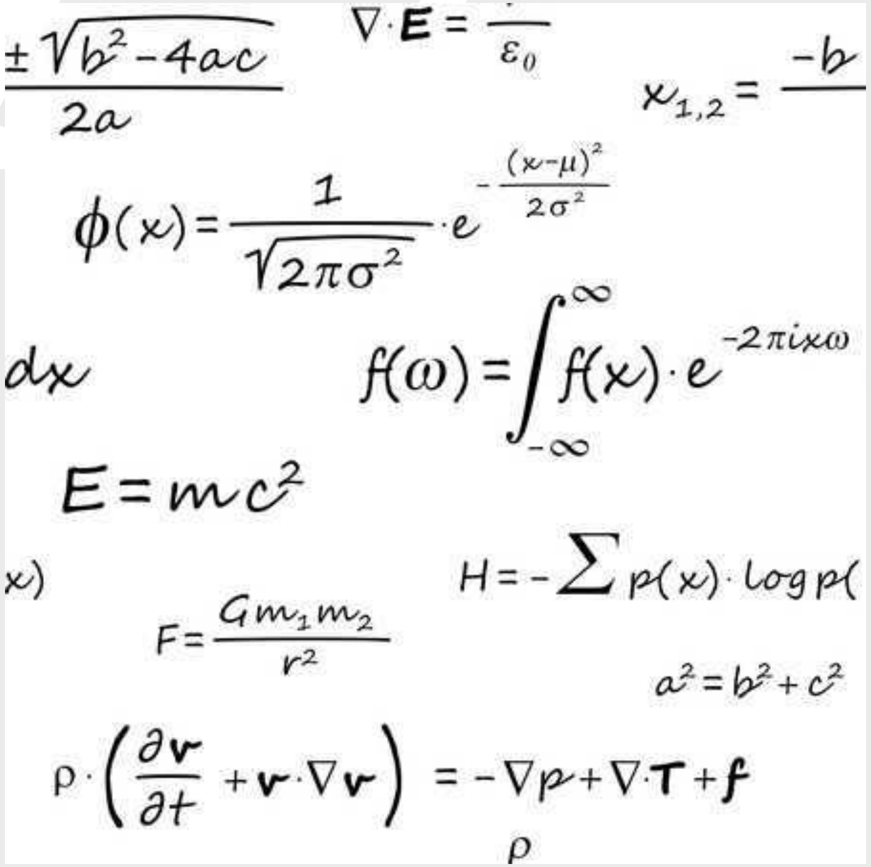


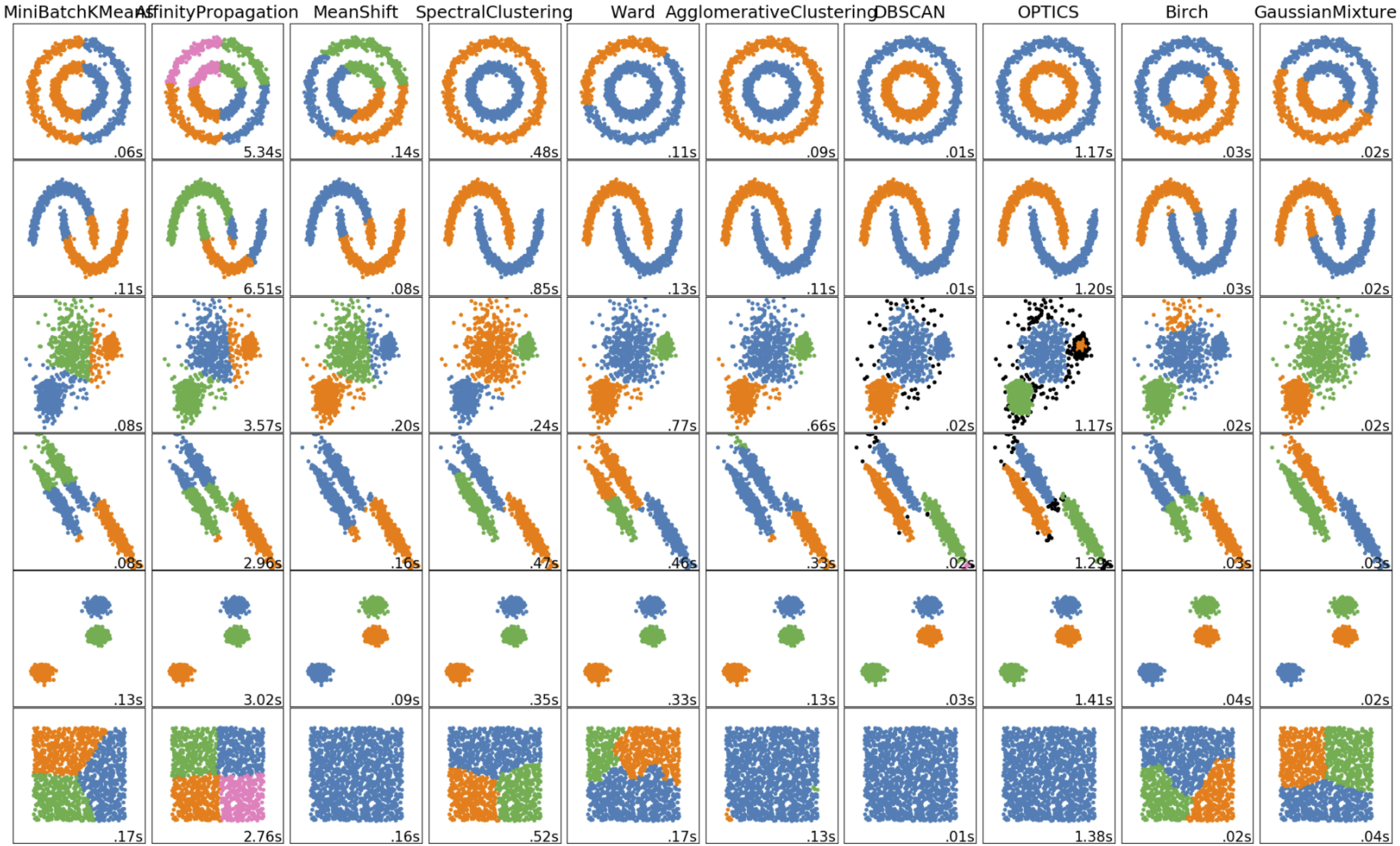
MANUAL method

Could there be a different way of clustering the animals?

Unsupervised Machine Learning

The Algorithms are trying to identify some segments or clusters in your data.


$\pm\sqrt{b^2-4ac}$ / $2a$
$\nabla\cdot\mathbf{E} = \frac{}{\varepsilon_0}$
$x_{1,2} = -b$
$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
dx
$f(\omega) = \int_{-\infty}^{\infty} f(x)\cdot e^{-2\pi i x\omega}$
$E = mc^2$
$x)$
$H = -\sum p(x)\cdot \log p($
$F = \frac{Gm_1m_2}{r^2}$
$a^2 = b^2 + c^2$
$\rho\cdot\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \nabla\cdot\mathbf{T} + \mathbf{f}$
ρ



A comparison of the clustering algorithms in scikit-learn

Let’s take a breather here

What do we know about ML till now?

- If we have a bunch of data, and we suspect that there are subgroups within the dataset, then without defining the subgroups, we can use Unsupervised ML algorithms to find those subgroups for us.

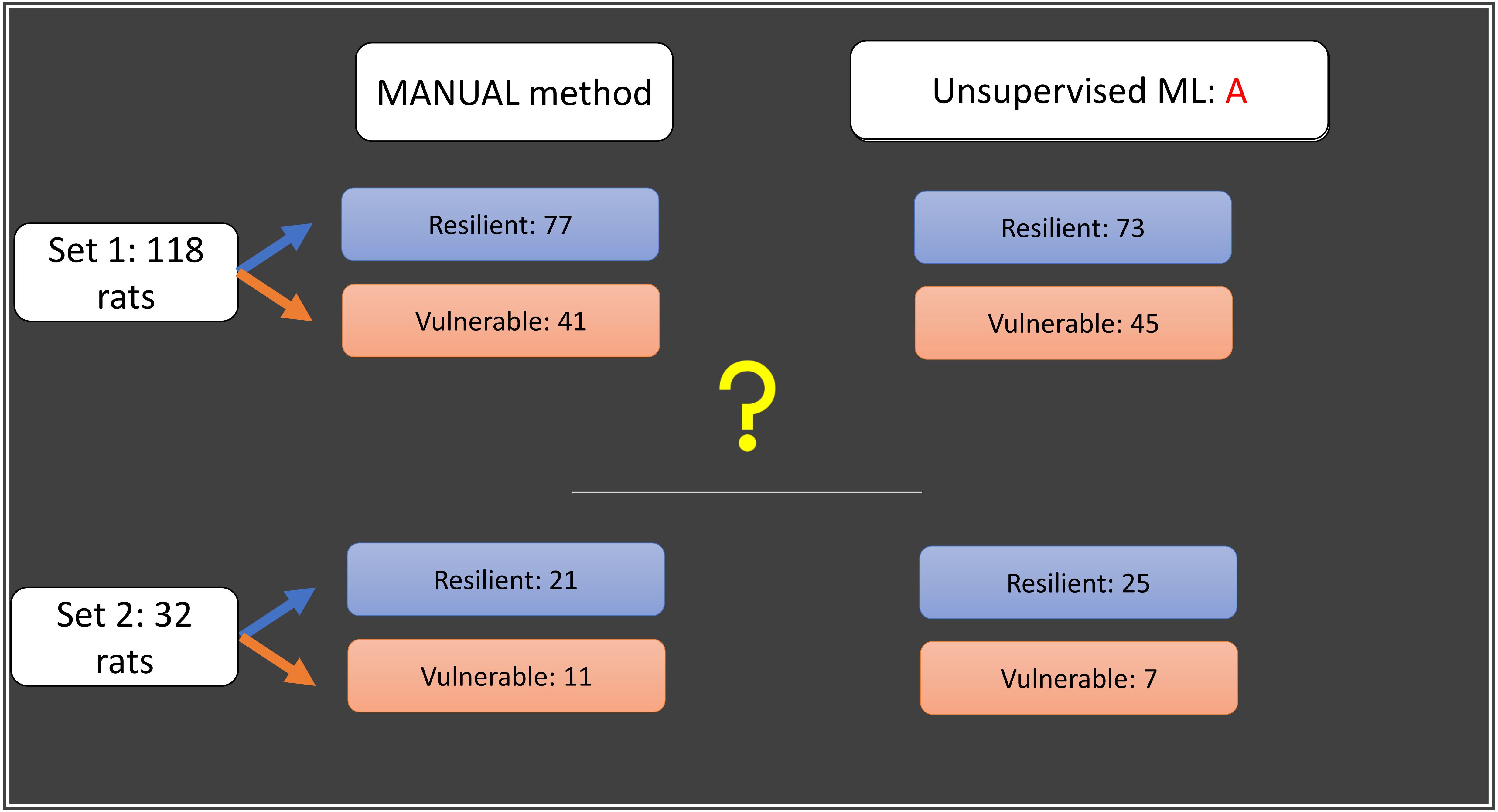
IMPORTANT: We only feed the raw data to the UNSUPERVISED ML ALGORITHM.

THE UNSUPERVISED ML ALGORITHM TELLS US THE POSSIBLE GROUPS

The real question is, which UNSUPERVISED ML algorithm to chose?

Or, far more important question is how do we know which is better?

Let’s take a look at this scenario



MANUAL method

Unsupervised ML: A

Set 1: 118 rats

Resilient: 77

Vulnerable: 41

Resilient: 73

Vulnerable: 45

?

Set 2: 32 rats

Resilient: 21

Vulnerable: 11

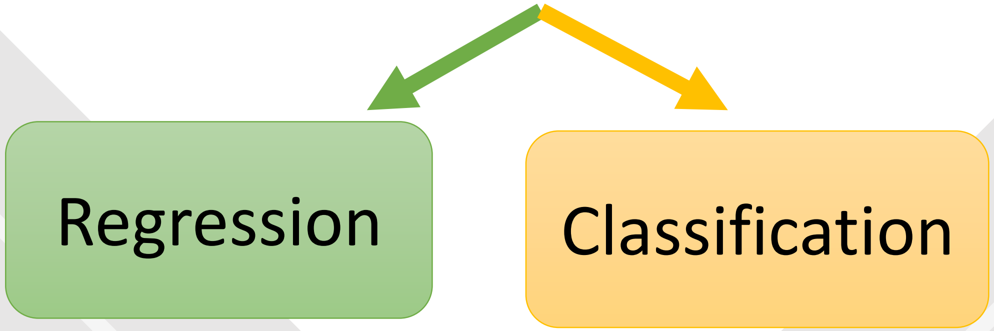
Resilient: 25

Vulnerable: 7

Strength of a ML model is its consistent predictive ability

This is where we introduce another form of Machine Learning

Supervised Machine Learning

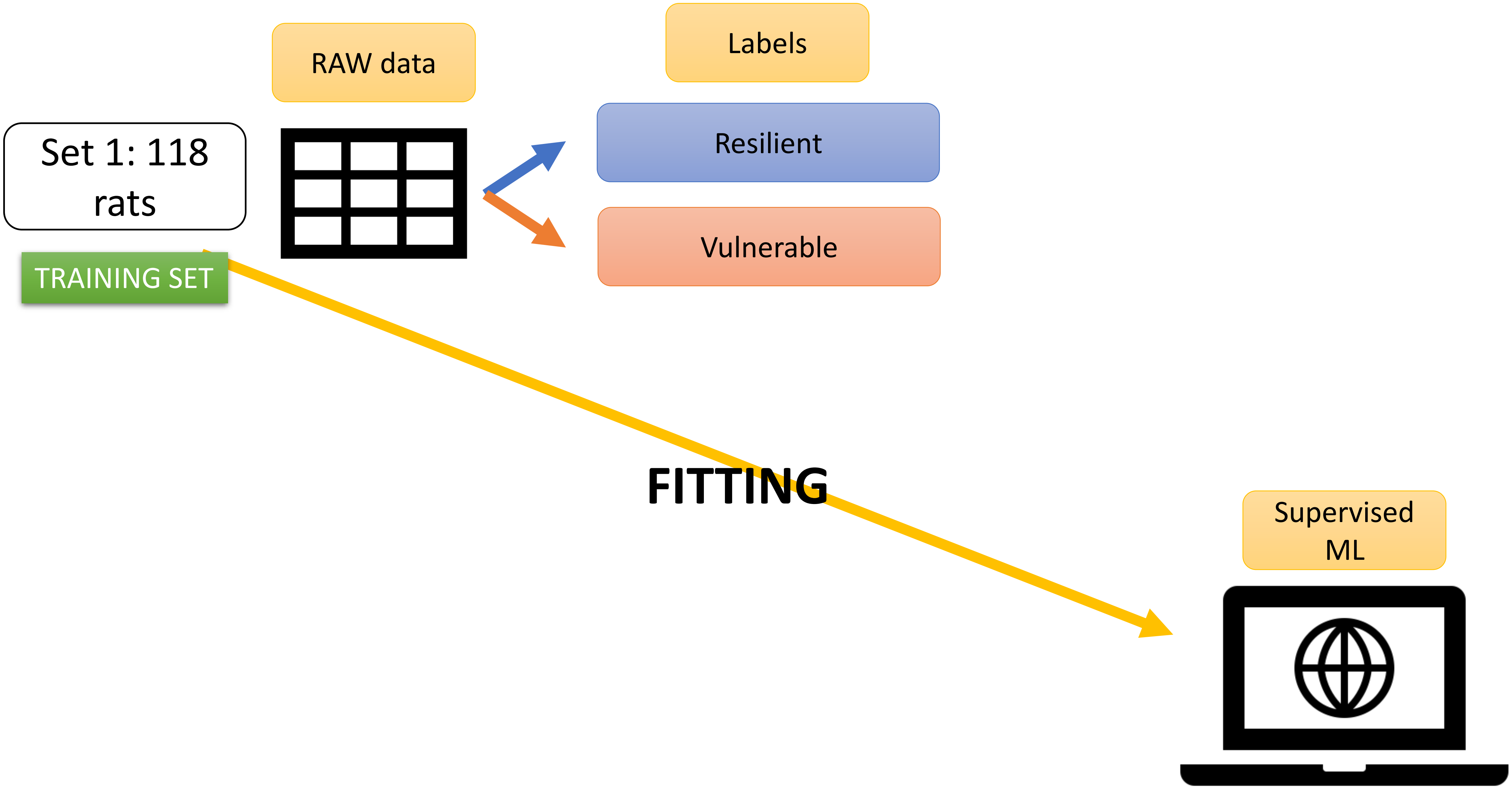


Regression

Classification

HOW DOES SUPERVISED ML WORK?

CLUSTERS GIVEN BY UNSUPERVISED ML OR MANUAL METHOD



Set 1: 118 rats

TRAINING SET

RAW data

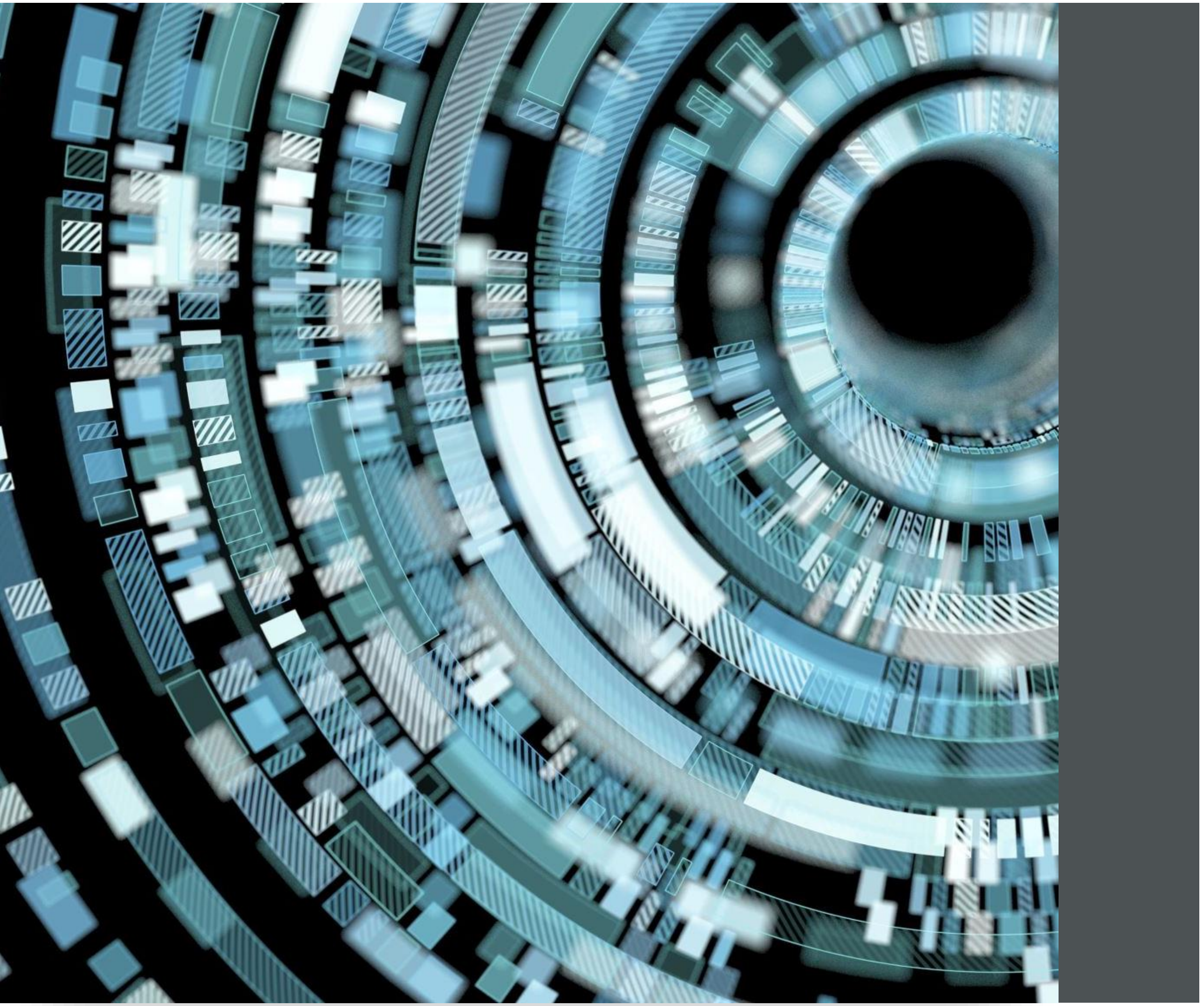
Labels

Resilient

Vulnerable

FITTING

Supervised ML

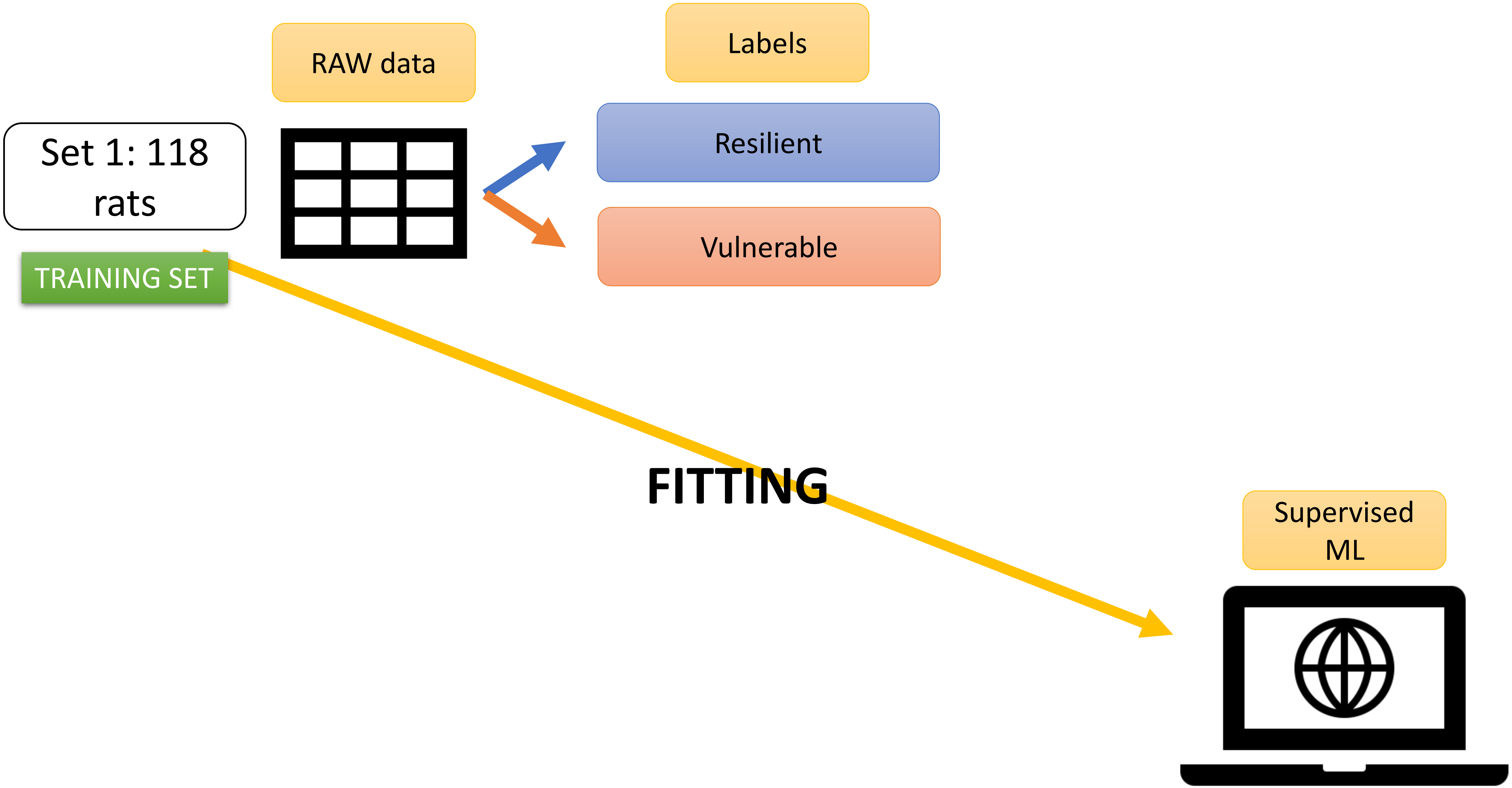


What is Fitting the data to Supervised ML algorithm?

Fitting means finding the best possible curve that explains the data

Grumpy old Statistician will say: ML is nothing but Glorified Curve Fitting

CLUSTERS GIVEN BY UNSUPERVISED ML OR MANUAL METHOD



RAW data

Labels

Set 1: 118 rats

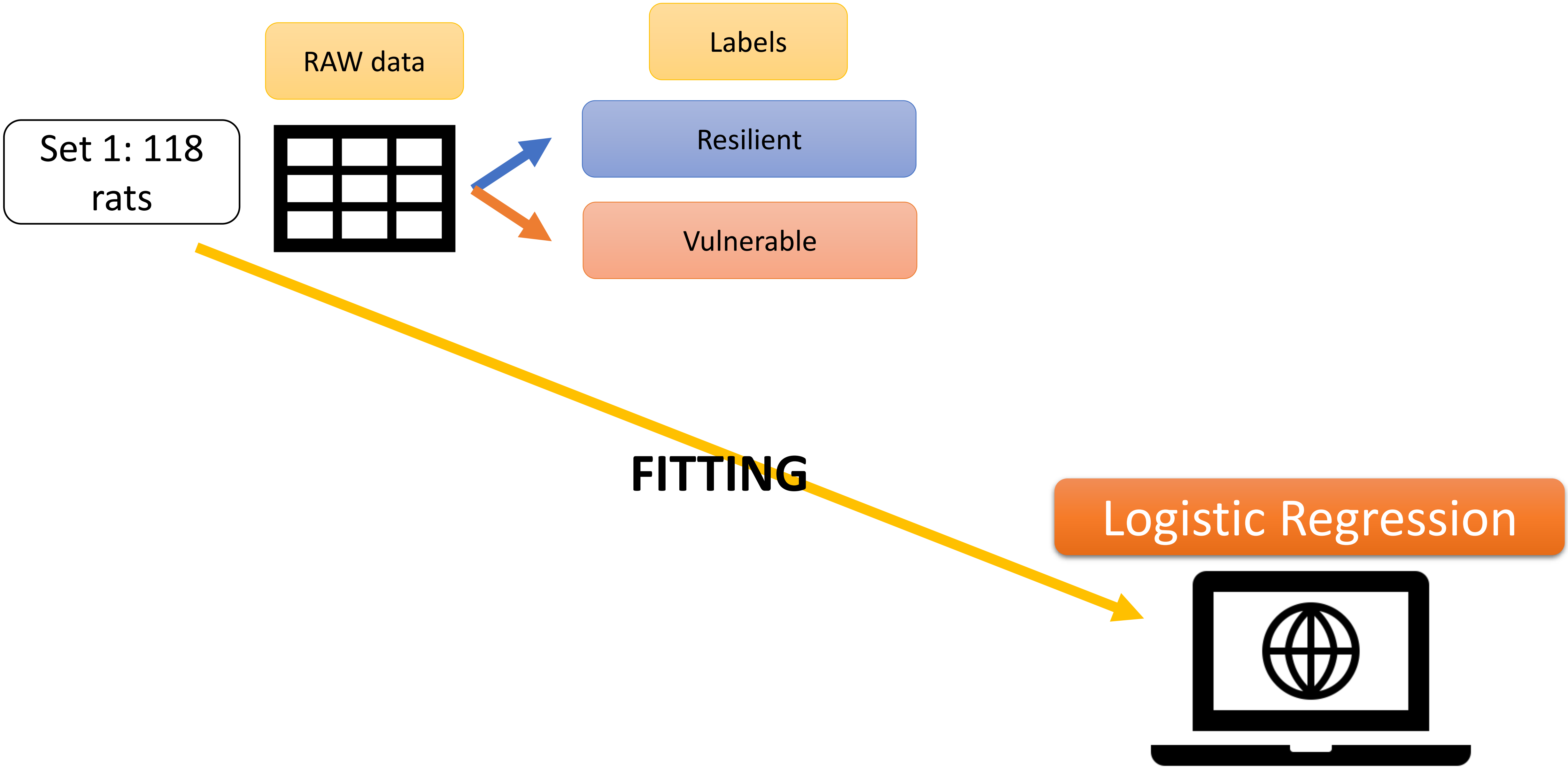
Resilient

Vulnerable

TRAINING SET

FITTING

Supervised ML



Set 1: 118 rats

RAW data

Labels

Resilient

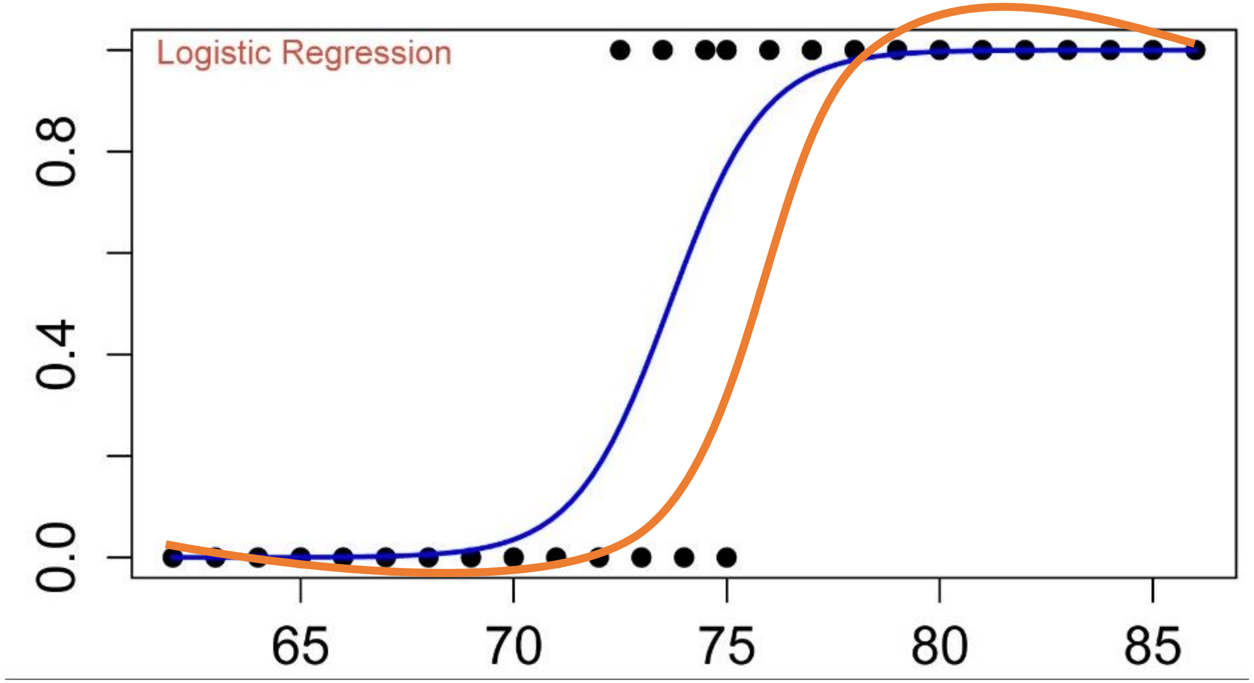
Vulnerable

FITTING

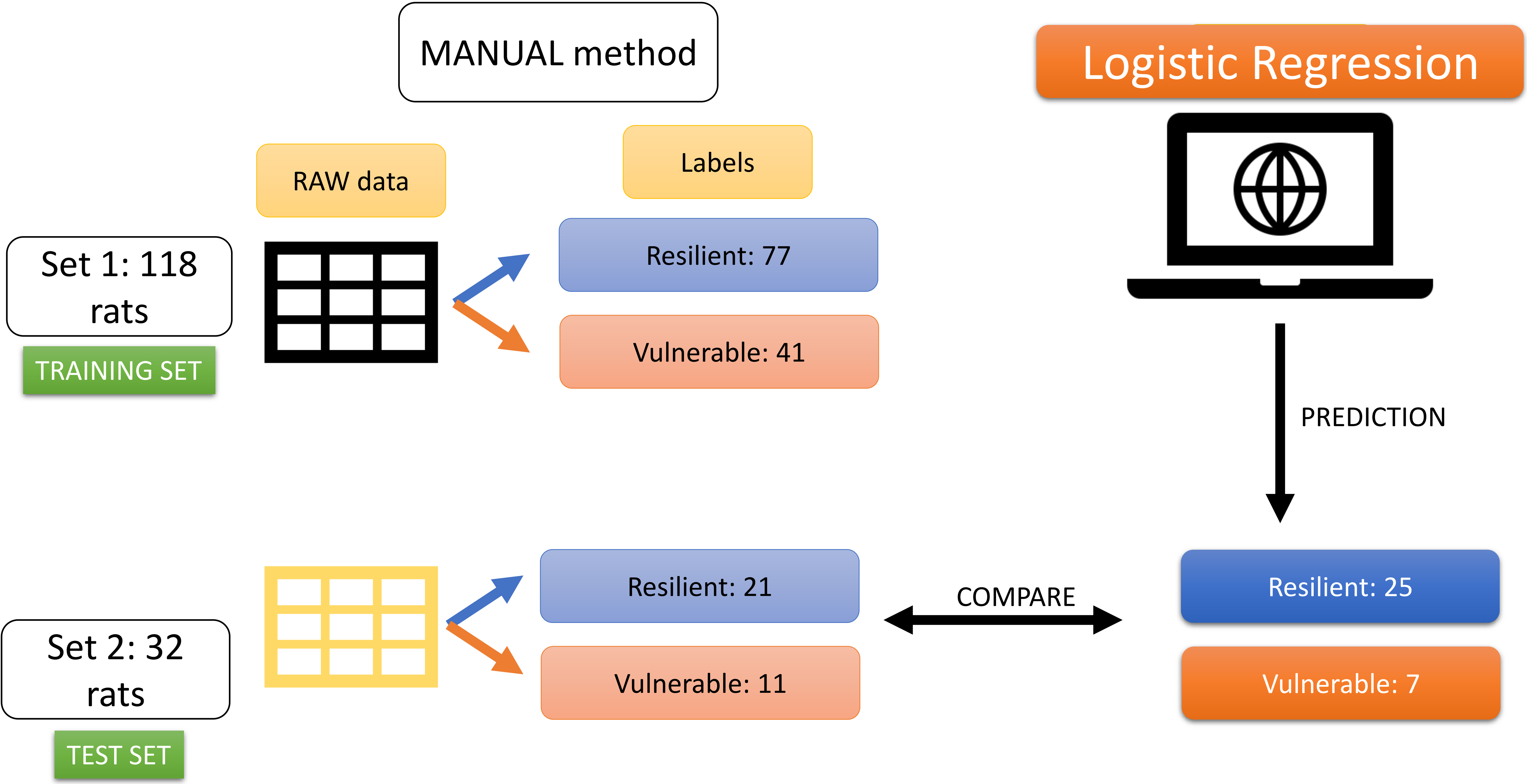
Logistic Regression

Logistic regression

Vulnerable



Resilient



MANUAL method

Logistic Regression

RAW data

Labels

Set 1: 118 rats

TRAINING SET

Resilient: 77

Vulnerable: 41

PREDICTION

Set 2: 32 rats

TEST SET

Resilient: 21

Vulnerable: 11

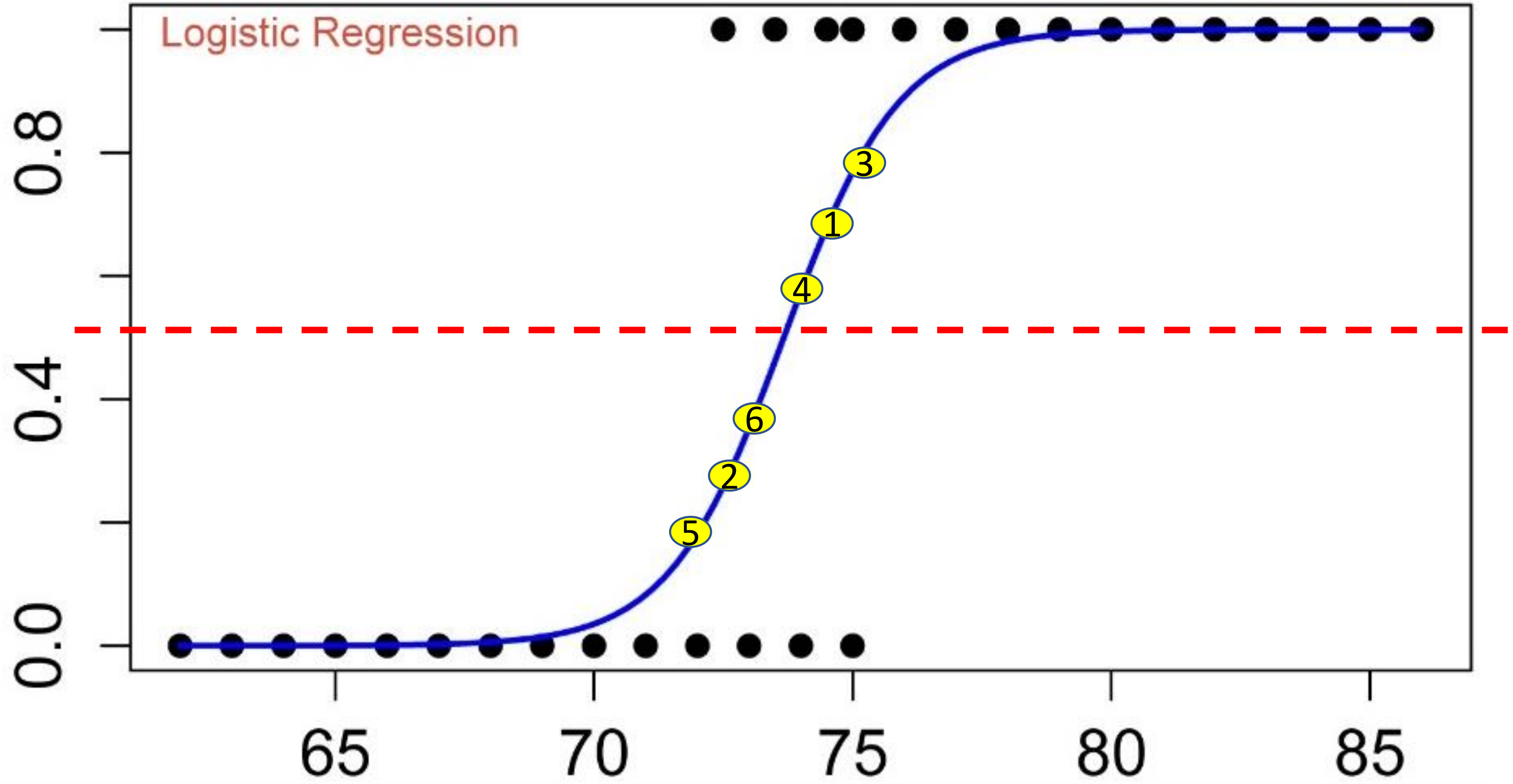
COMPARE

Resilient: 25

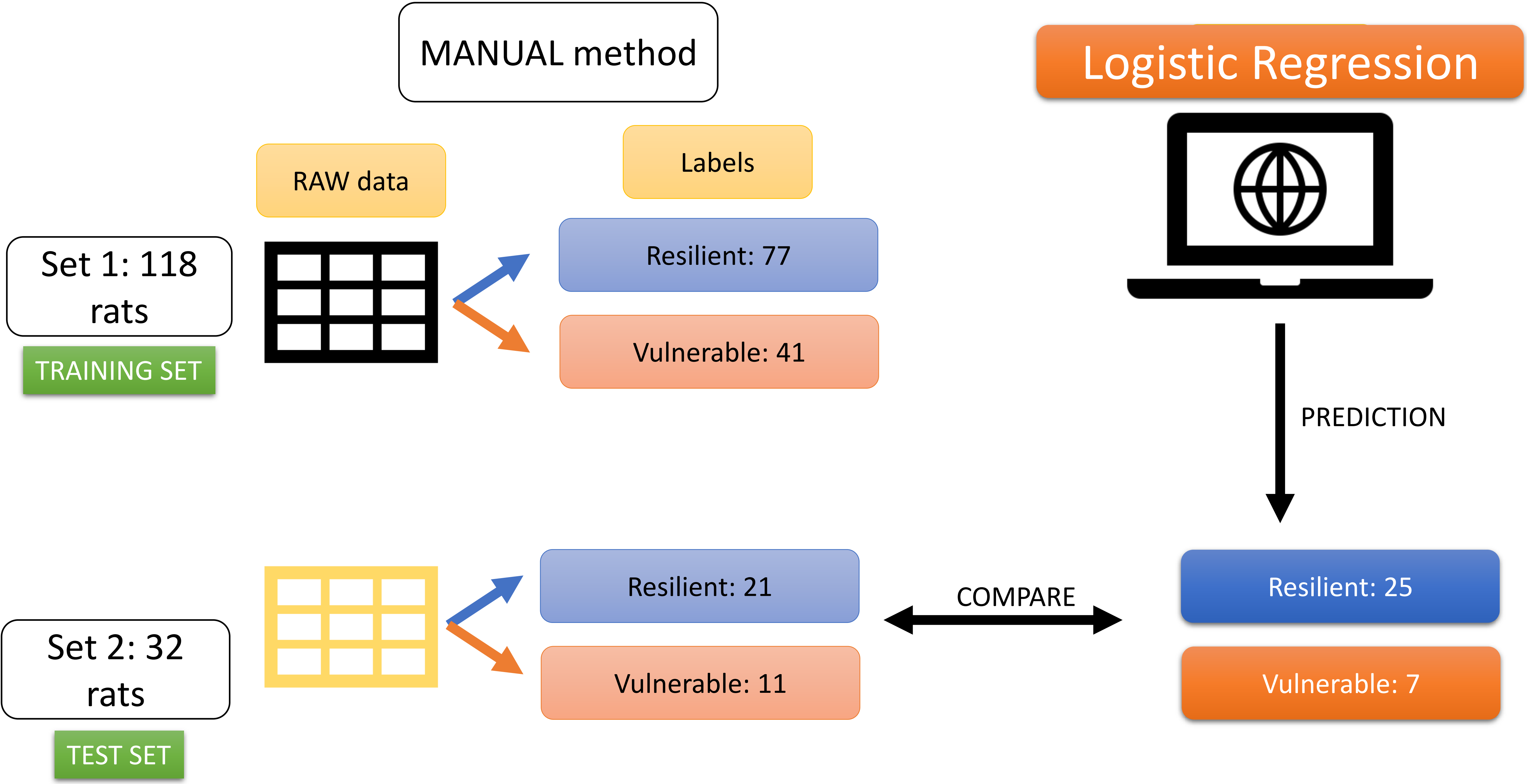
Vulnerable: 7

But how does Logistic regression predict?

Vulnerable



Resilient



MANUAL method

Logistic Regression

RAW data

Labels

Set 1: 118 rats

TRAINING SET

Resilient: 77

Vulnerable: 41

PREDICTION

Set 2: 32 rats

TEST SET

Resilient: 21

Vulnerable: 11

COMPARE

Resilient: 25

Vulnerable: 7

CONFUSION MATRIX: MANUAL METHOD followed by LOGISTIC REGRESSION

	Resilient (Logistic Regression- Test dataset)	**Vulnerable (Logistic Regression- Test dataset)**	**Total**
Resilient (Manual method: test data)	21 (True Resilient)	0 (False Vulnerable)	**21**
Vulnerable (Manual method: test data)	4 (False Resilient)	7 (True Vulnerable)	**11**
Total	**25**	**7**	**32**

Prediction Accuracy: 87.5%

Remember what we had set out to do

- Finding a better way to find Resilient and Vulnerable rats.

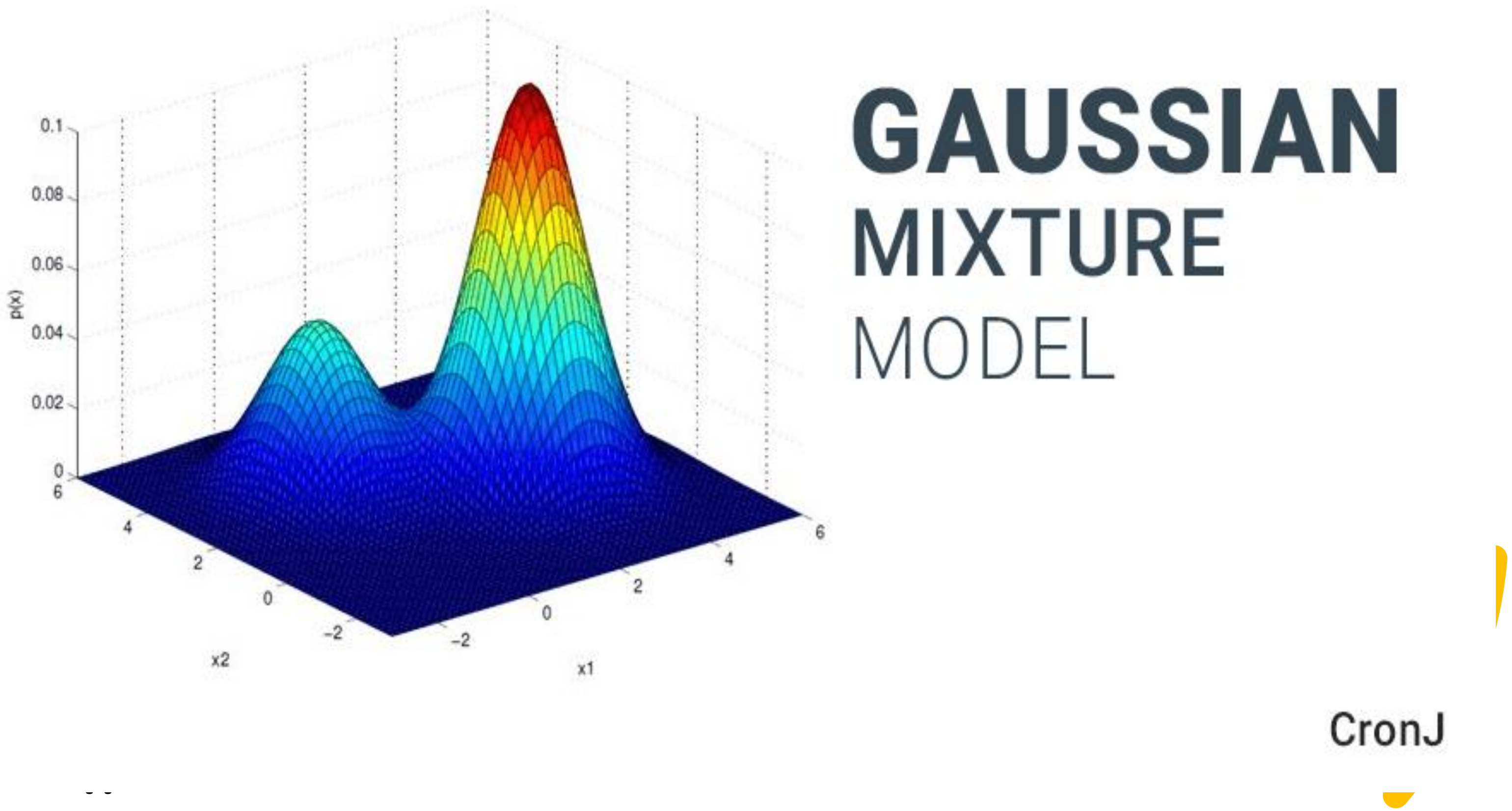
- We tested MANUAL Method.

- Its good, but not perfect.

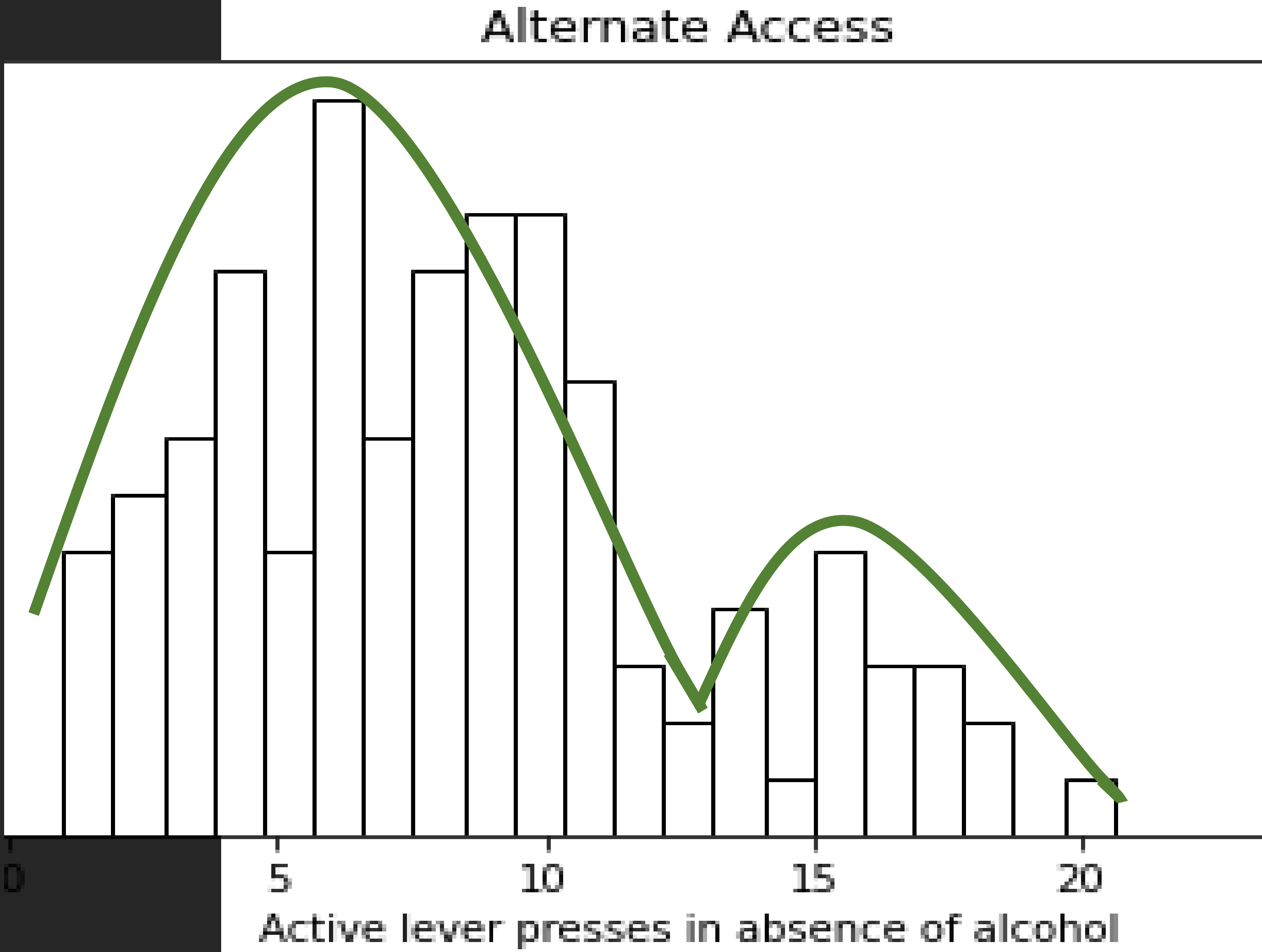
- So, we need to find a different clustering model

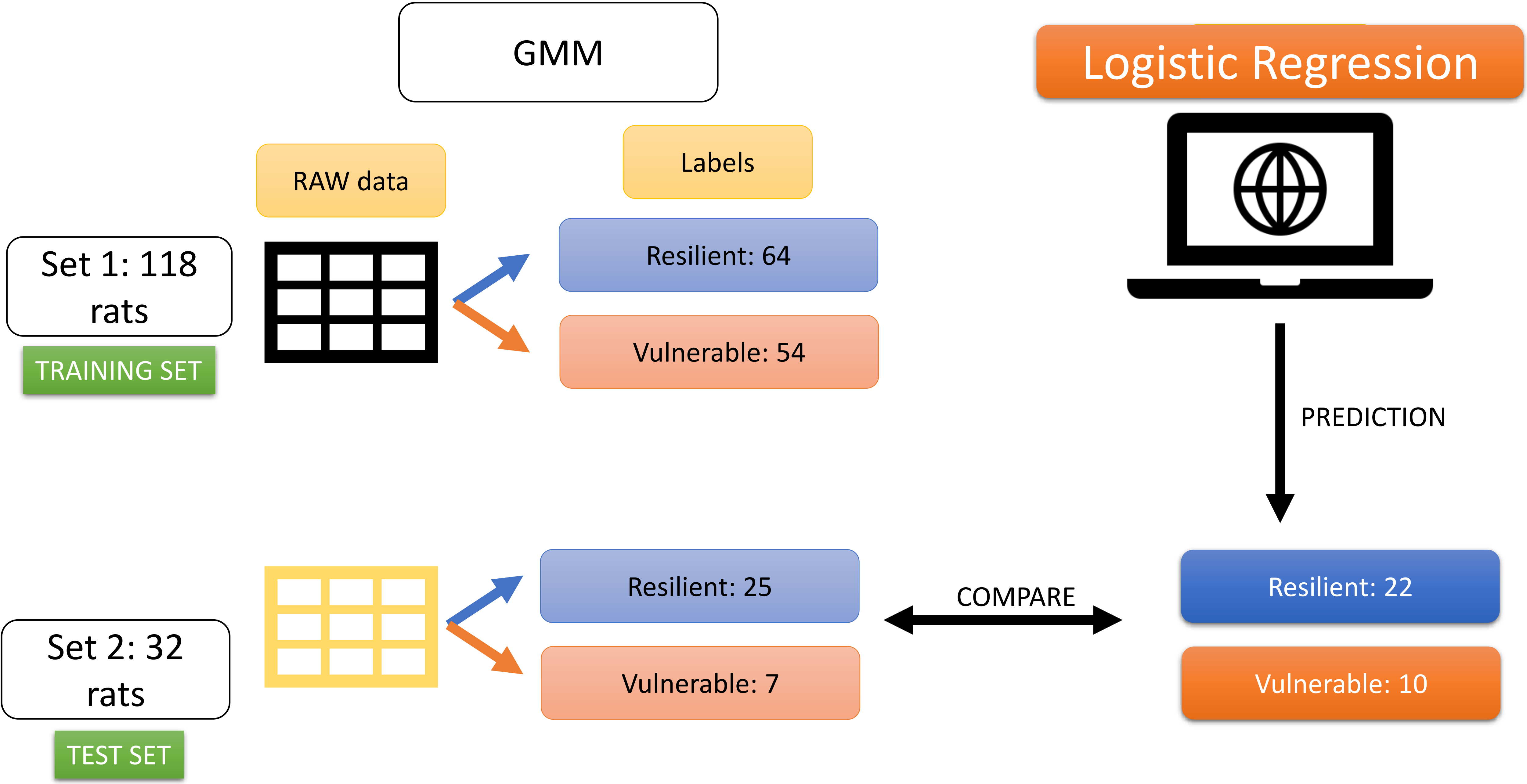
Examples of Unsupervised Clustering ML algorithms

- Gaussian Mixture Method



Why the GMM?


Alternate Access
0
5
10
15
20
Active lever presses in absence of alcohol



GMM

Logistic Regression

RAW data

Labels

Set 1: 118 rats

TRAINING SET

Resilient: 64

Vulnerable: 54

PREDICTION

Set 2: 32 rats

TEST SET

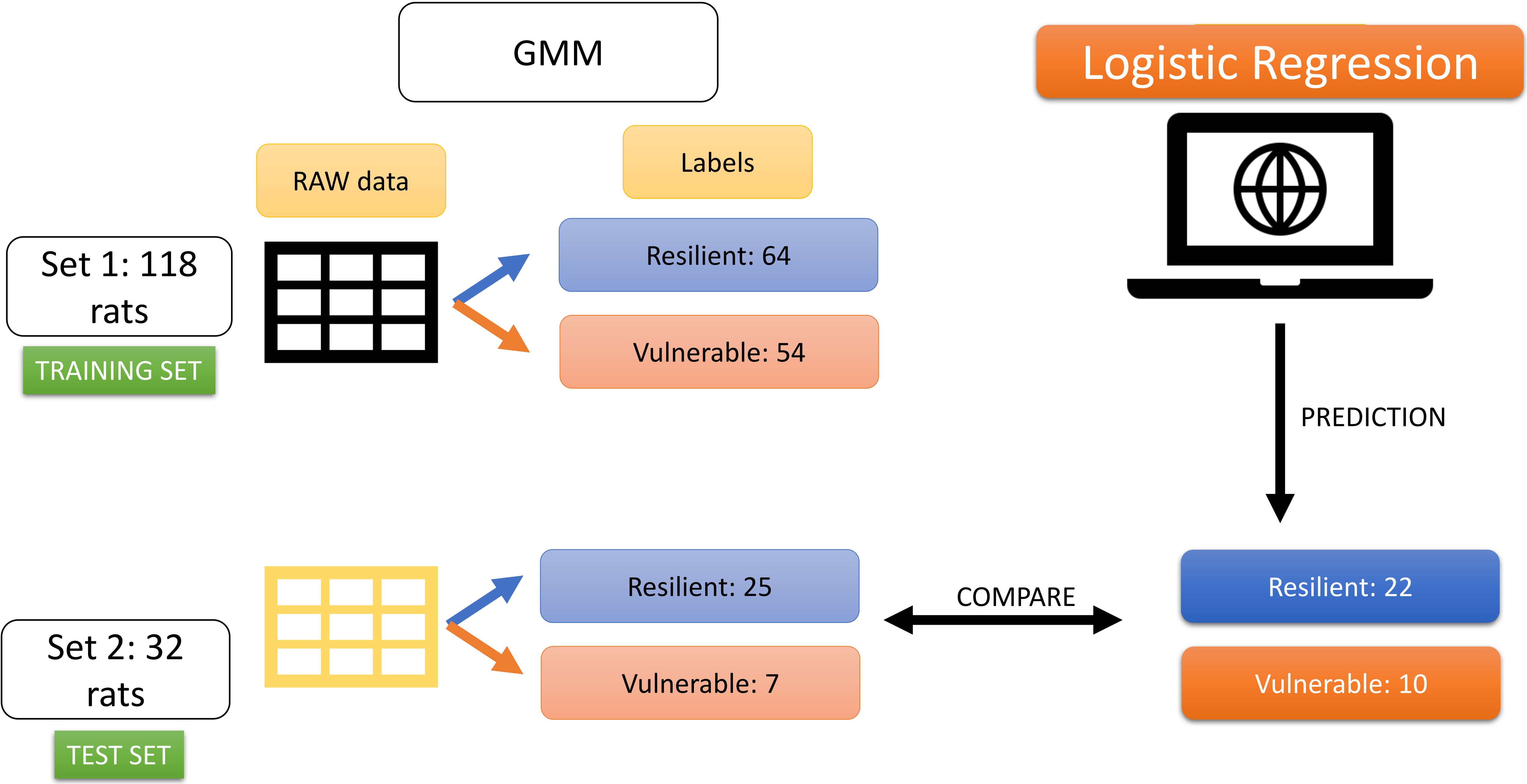
Resilient: 25

Vulnerable: 7

COMPARE

Resilient: 22

Vulnerable: 10



GMM

Logistic Regression

RAW data

Labels

Set 1: 118 rats

TRAINING SET

Resilient: 64

Vulnerable: 54

PREDICTION

Set 2: 32 rats

TEST SET

Resilient: 25

Vulnerable: 7

COMPARE

Resilient: 22

Vulnerable: 10

CONFUSION MATRIX: GMM followed by LOGISTIC REGRESSION

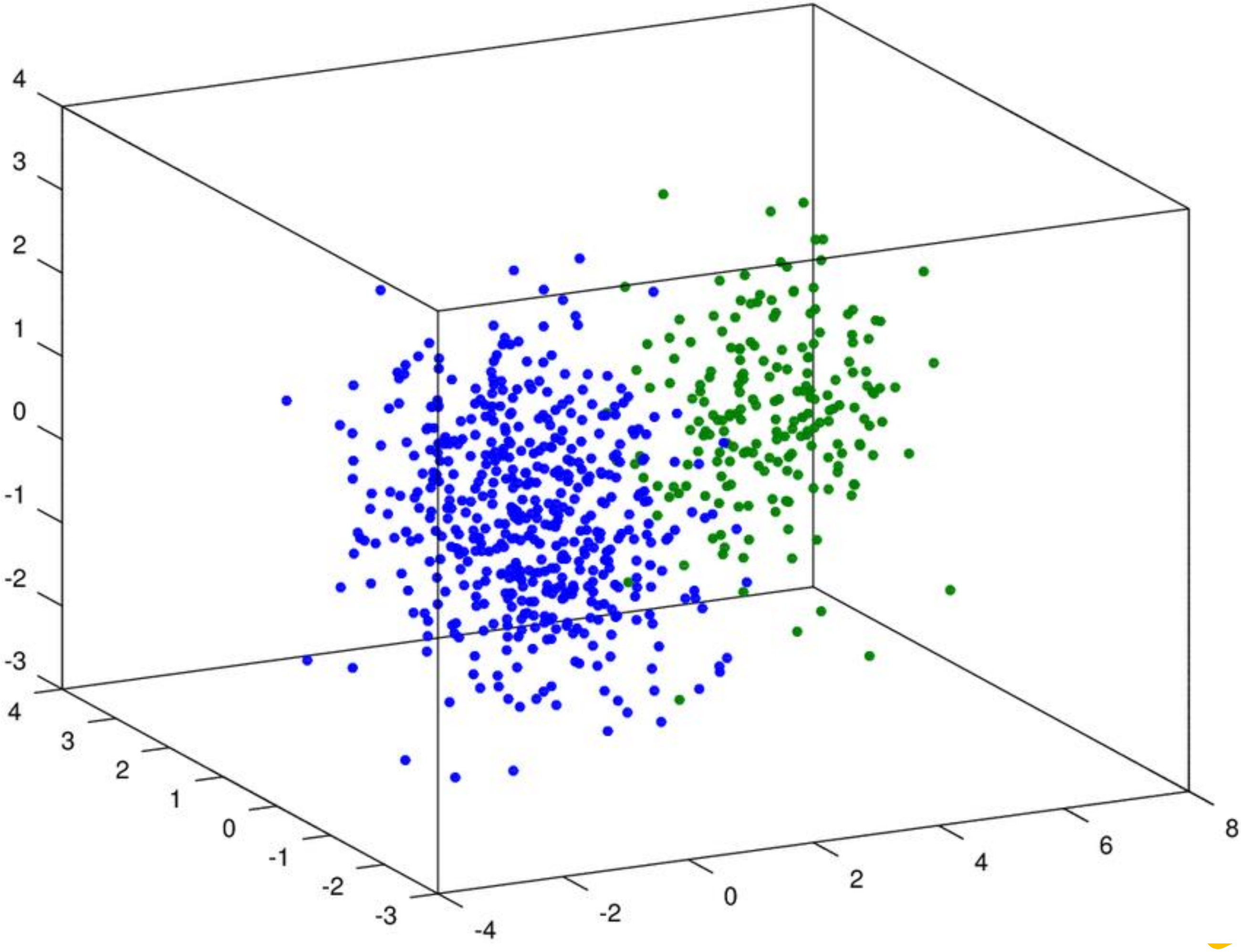
	Resilient (Logistic Regression- Test dataset)	**Vulnerable (Logistic Regression- Test dataset)**	**Total**
Resilient (GMM)	22 (True Resilient)	3 (False Vulnerable)	**25**
Vulnerable (GMM)	0 (False Resilient)	7 (True Vulnerable)	**7**
Total	**22**	**10**	**32**

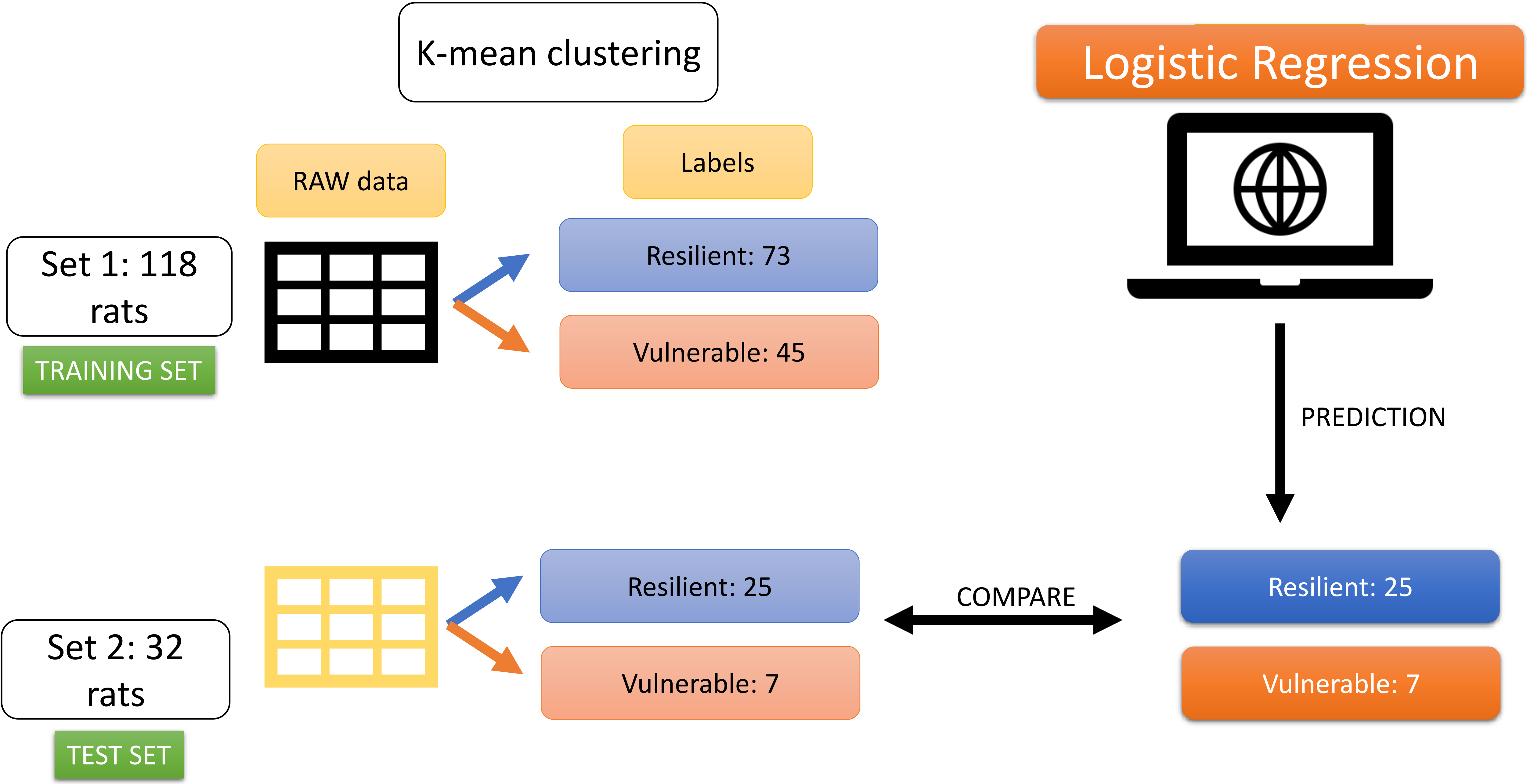
Prediction Accuracy: 90.62 %

Better than MANUAL method but not perfect

Examples of Unsupervised Clustering ML algorithms

- K-mean Clustering





K-mean clustering

Logistic Regression

RAW data

Labels

Set 1: 118 rats

TRAINING SET

Resilient: 73

Vulnerable: 45

PREDICTION

Set 2: 32 rats

TEST SET

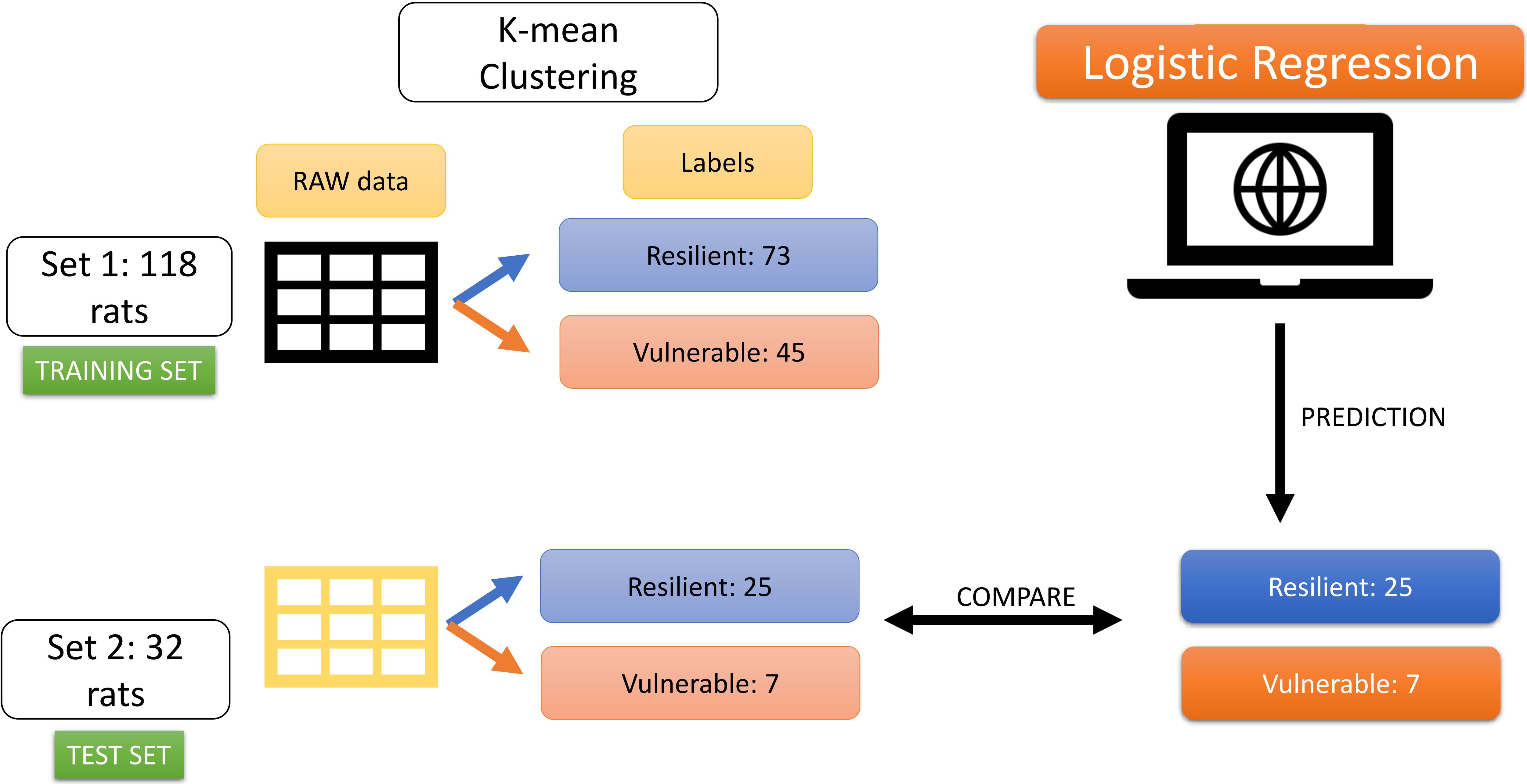
Resilient: 25

Vulnerable: 7

COMPARE

Resilient: 25

Vulnerable: 7



K-mean Clustering

Logistic Regression

RAW data

Labels

Set 1: 118 rats

TRAINING SET

Resilient: 73

Vulnerable: 45

PREDICTION

Set 2: 32 rats

TEST SET

Resilient: 25

Vulnerable: 7

COMPARE

Resilient: 25

Vulnerable: 7

CONFUSION MATRIX: K-mean clustering followed by LOGISTIC REGRESSION

	Resilient (Logistic Regression-Test dataset)	**Vulnerable (Logistic Regression-Test dataset)**	**Total**
Resilient (K-mean clustering)	25 (True Resilient)	0 (False Vulnerable)	**25**
Vulnerable (K-mean Clustering)	0 (False Resilient)	7 (True Vulnerable)	**7**
Total	**25**	**0**	**32**

Prediction Accuracy: 100 %

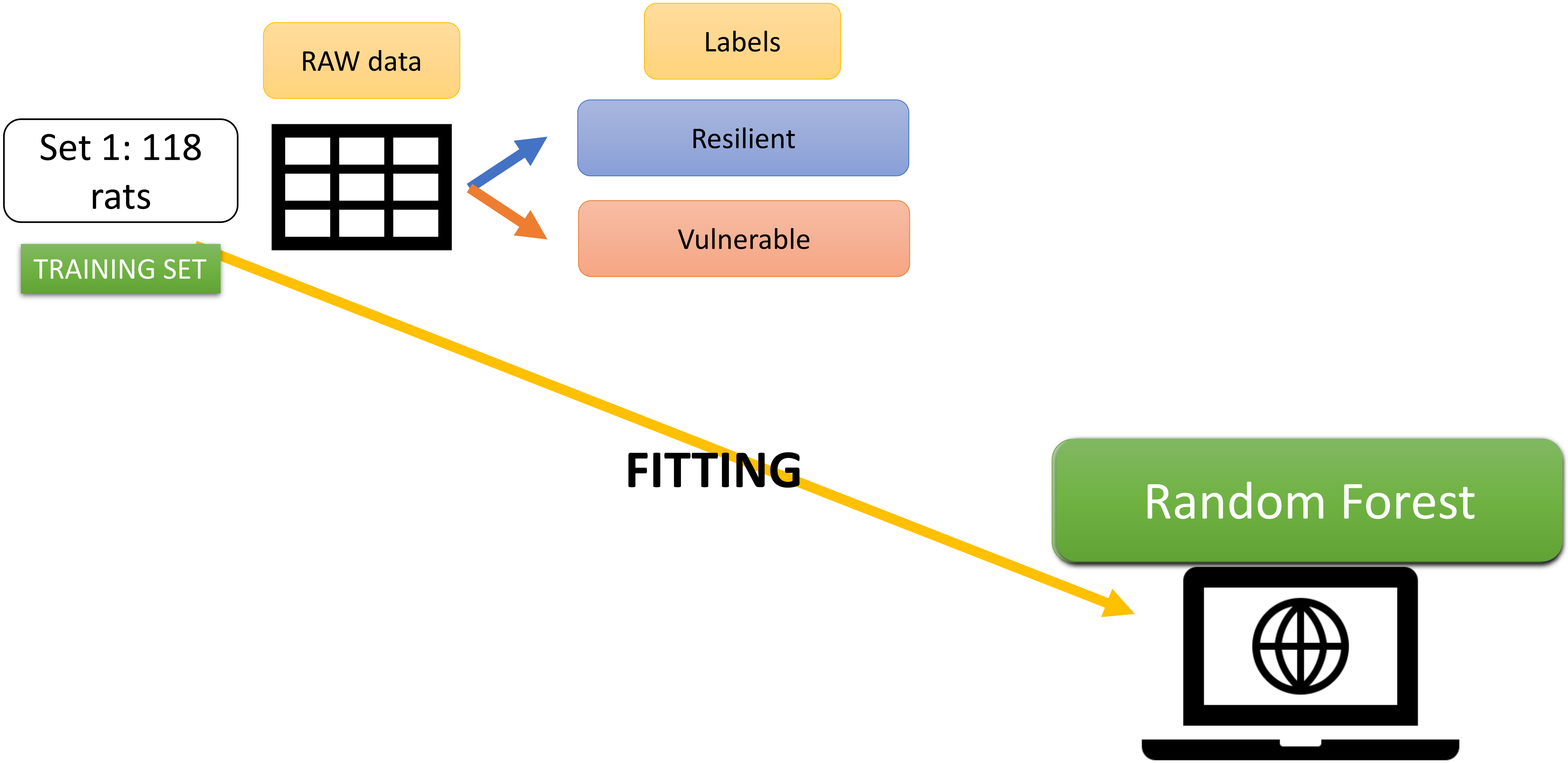
PERFECT SCORE

What does this tell us?

- K-mean Clustering is definitely superior for grouping the rats to determine their addiction vulnerabilities.

- However, MANUAL method isn’t off target by a huge margin.

Quick Pointers



RAW data

Labels

Set 1: 118 rats

Resilient

Vulnerable

TRAINING SET

FITTING

Random Forest

Summary of using other Supervised ML models

Predictive accuracy	Logistic Regression	Support Vector Machines	K-Nearest Neighbour
Manual Method	87.5%	84.37%	87.5%
GMM	90.62%	93.75%	93.75%
K-mean Clustering	100%	100%	100%

K mean clustering wins!!!

Finally, Deep Learning

- It’s the latest gizmo in Machine Learning world.
- It tries to replicate something like our brain.
- But it is resource intensive so, I haven’t used it here

Resources used for this analysis

- Google Colab environment
- Python 3.8.1
- Python Libraries
  - SKLEARN
  - MATPLOTLIB
  - PANDAS
  - NUMPY

Thank you,
Dr. Benjamin Boutrel
and lab.

Thank you, Google.

Thank you everyone
for Listening.