Formula

1. Least Square

Population Y intercept

Population Slope Coefficient

Independent Variable

Random Error term

Dependent Variable

$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

Linear Regression

- Gradient Descendant
- Least Square
- Linear Equation
- SVD
- QR Decomposition
- Monte Carlo Method

Y = mx +c

Least Square

$$\beta = \frac{\mathrm{cov}(x,y)}{\mathrm{var}(x)}$$

$$\mathrm{var}(x) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

$$\beta = \frac{22.65}{23.2} = 0.976293103448275 8$$

$$\mathrm{cov}(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

$$\alpha = \bar{y} - \beta\bar{x}$$

x bar is the mean of x, xi is the value of x for the ith training instance, and n is the number of training instances.

Linear Equation

$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Table 1

x	y
1	1
2	3
3	3
4	2
5	5

2. Logistic Regression

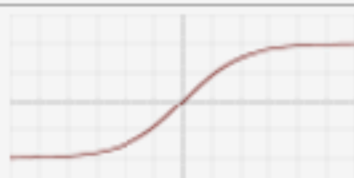
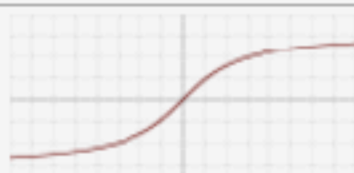
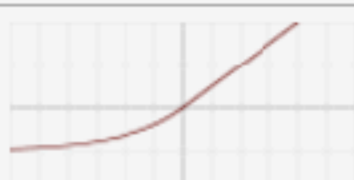
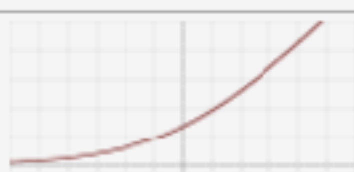
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x \neq 0 \\ ? & \text{for} \quad x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Table 1

X1	X2	Y
2.7810836	2.550537003	0
1.465489372	2.362125076	0
3.396561688	4.400293529	0
1.38807019	1.850220317	0
3.06407232	3.005305973	0
7.627531214	2.759262235	1
5.332441248	2.088626775	1
6.922596716	1.77106367	1
8.675418651	-0.242068655	1
7.673756466	3.508563011	1

3. KNN

Euclidean Distance = $\sqrt{(X_H - H_1)^2 +} (X_W - W_1)^2$

Where

X_H: Observation value of variable Height

H_1: Centroid value of Cluster 1 for variable Height

X_W: Observation Value of variable Weight

W_1: Centroid value of cluster 1 for variable Weight

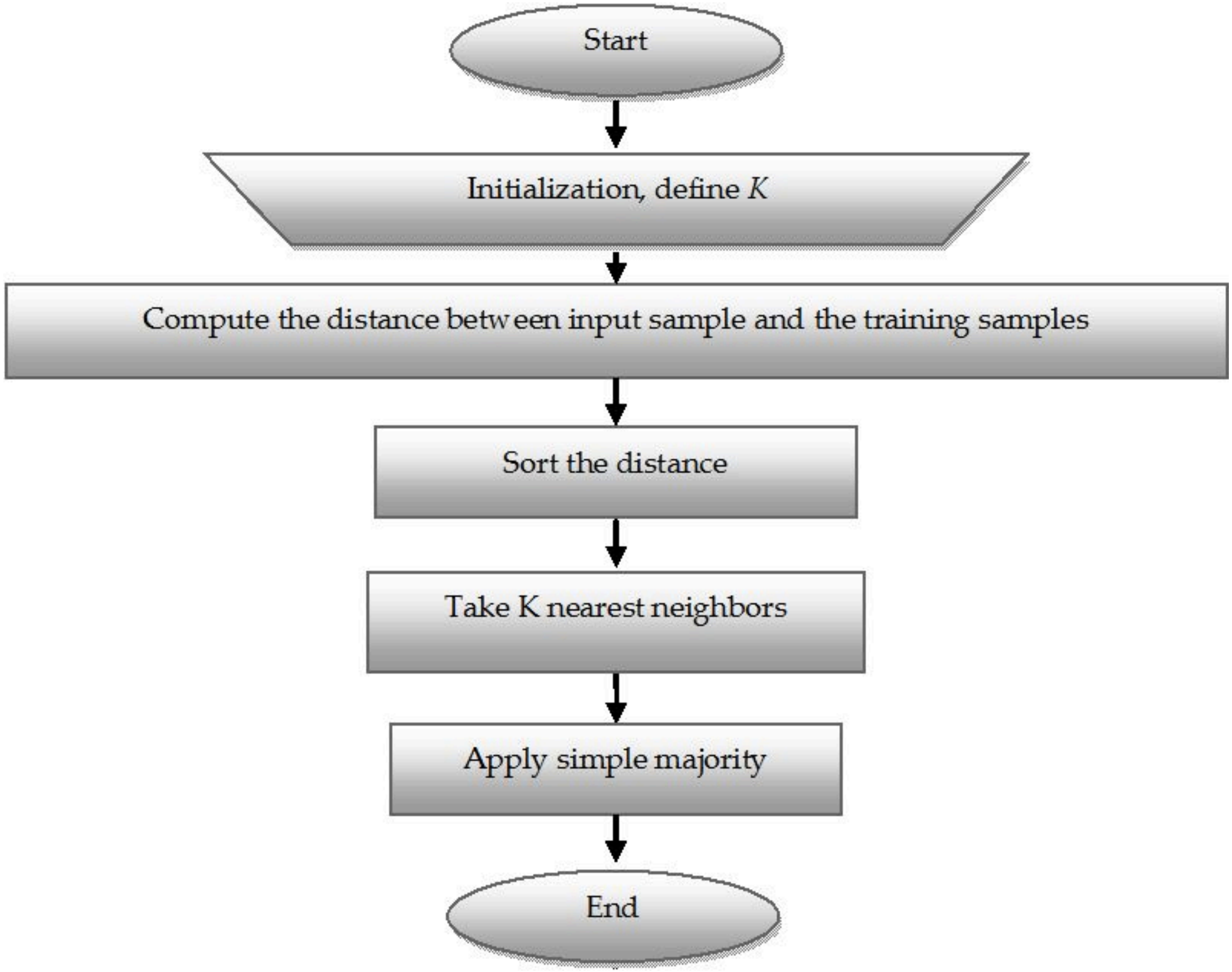

Start
Initialization, define K
Compute the distance between input sample and the training samples
Sort the distance
Take K nearest neighbors
Apply simple majority
End

Table 1

X1	X2	Y
3.393533211	2.331273381	0
3.110073483	1.781539638	0
1.343808831	3.368360954	0
3.582294042	4.67917911	0
2.280362439	2.866990263	0
7.423436942	4.696522875	1
5.745051997	3.533989803	1
9.172168622	2.511101045	1
7.792783481	3.424088941	1
7.939820817	0.791637231	1

4. Naive Bayes

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}$$

Chills	Runny Nose	Headache	Fever	Flu?
Yes	No	Moderate	Yes	No
Yes	Yes	No	No	Yes
Yes	No	Strong	Yes	Yes
No	Yes	Moderate	Yes	Yes
No	No	No	No	No
No	Yes	Strong	Yes	Yes
No	Yes	Strong	No	No
Yes	Yes	Moderate	Yes	Yes

What about the next patient? Symptoms:

Chills	Runny Nose	Headache	Fever	Flu?
Yes	No	Moderate	No	?

Condition	Probability	Condition	Probability
P(Flu=Yes)	0.625	P(Flu=No)	0.375
P(Chills=Yes|Flu=Yes)	0,6	P(Chills=Yes|Flu=No)	0.333
P(Chills=No|Flu=Yes)	0,4	P(Chills=No|Flu=No)	0.666
P(Runny Nose=Yes|Flu=Yes)	0,8	P(Runny Nose=Yes|Flu=No)	0.333
P(Runny Nose=No|Flu=Yes)	0,2	P(Runny Nose=No|Flu=No)	0.666
P(Headache=Moderate|Flu=Yes)	0,4	P(Headache=Moderate|Flu=No)	0.333
P(Headache=No|Flu=Yes)	0,2	P(Headache=No|Flu=No)	0.333
P(Headache=Strong|Flu=Yes)	0,4	P(Headache=Strong|Flu=No)	0.333
P(Temperature=Yes|Flu=Yes)	0,8	P(Temperature=Yes|Flu=No)	0.333
P(Temperature=No|Flu=Yes)	0,2	P(Temperature=No|Flu=No)	0.666

Chills	Runny Nose	Headache	Fever	Flu?
Yes	No	Moderate	No	?

P(Flu=Yes)P(Chills=Yes|Flu=Yes)P(Runny Nose=No|Flu=Yes)P(Headache=Moderate|Flu=Yes)P(Temperature =No|Flu=Yes) = 0.006

P(Flu=No)P(Chills=Yes|Flu=No)P(Runny Nose=No|Flu=No)P(Headache=Moderate|Flu=No)P(Temperature= No|Flu=No) = 0.0185

Table 1

Weather	Car	Class
sunny	working	go-out
rainy	broken	go-out
sunny	working	go-out
sunny	working	go-out
sunny	working	go-out
rainy	broken	stay-home
rainy	broken	stay-home
sunny	working	stay-home
sunny	broken	stay-home
rainy	broken	stay-home

Table 1

Chills	Runny Nose	Headache	Fever	Flu?
Yes	No	Moderate	Yes	No
Yes	Yes	No	No	Yes
Yes	No	Strong	Yes	Yes
No	Yes	Moderate	Yes	Yes
No	No	No	No	No
No	Yes	Strong	Yes	Yes
No	Yes	Strong	No	No
Yes	Yes	Moderate	Yes	Yes

5. K Means

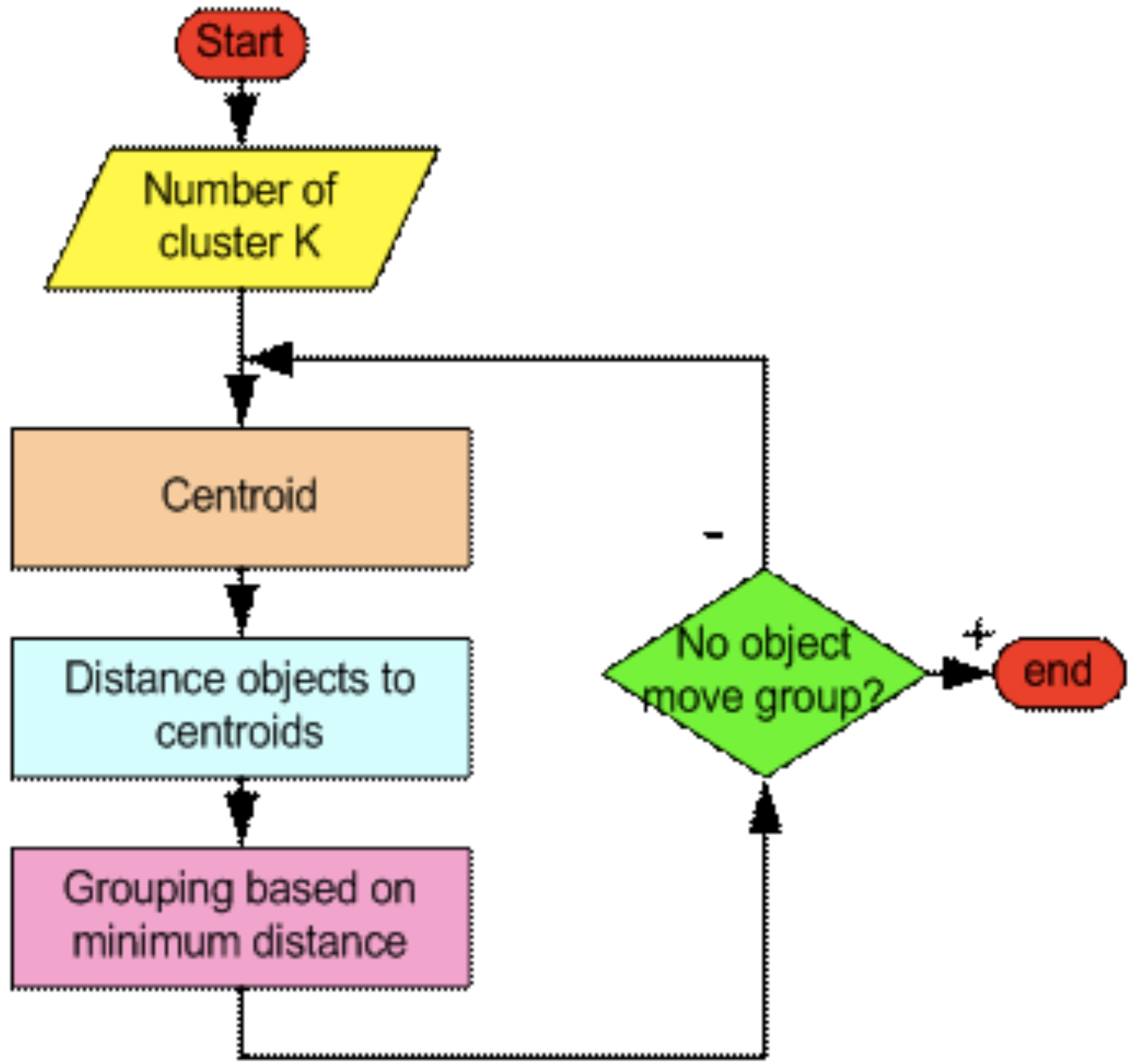
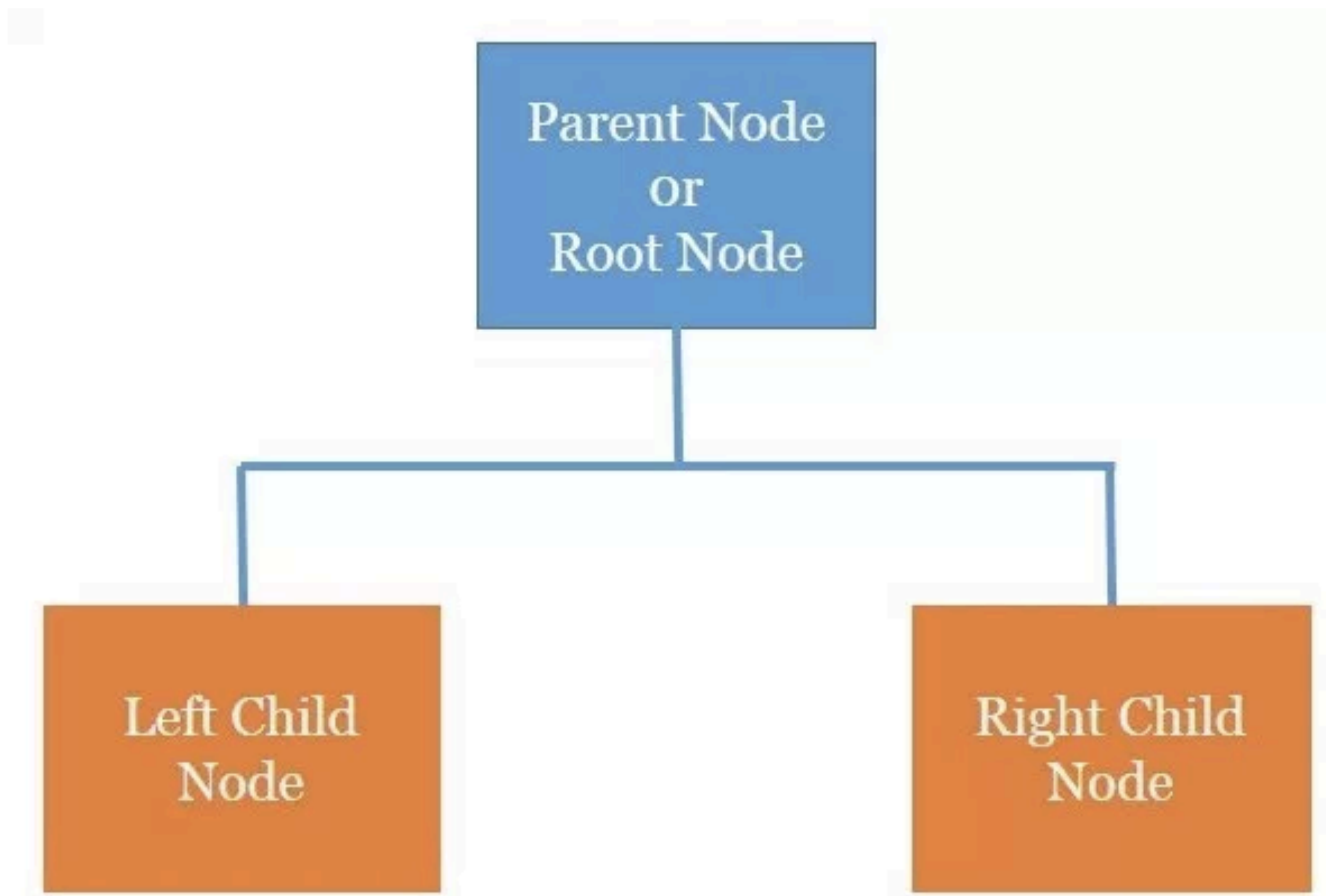

Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

Table 1

Height	Weight
185	72
170	56
168	60
179	68
182	72
188	77
180	71
180	70
183	84
180	88
180	67
177	76

6. Decision Tree

Decision Tree



L : Left child node for the root node

R : Right child node for the root node

$$P_R = \frac{\text{Number of records in right child node}}{\text{Total number of records}}$$

$$P_R = \frac{\text{Number of records in right child node}}{\text{Total number of records}}$$

$$P(k|\mathrm{L}) = \frac{\text{Number of class k records in left child node}}{\text{Number of records in left child node}}$$

$$P(k|\mathrm{R}) = \frac{\text{Number of class k records in right child node}}{\text{Number of records in right child node}}$$

Activity in the Last Quarter	Number of Solicited Customers	Campaign Results		Success Rate
		Responded (*r*)	Not Responded (*nr*)	
low	40000	720	39280	1.8%
medium	30000	1380	28620	4.6%
high	30000	2100	27900	7.0%

As you know, CART decision tree algorithm splits the root node into just two child nodes. Hence for this data, CART can form three combinations of binary trees as shown in the table below.

Left Node	Right Node	P_L	P_R	P(k|L) = a	P(k|R) = b	Ψ(Large Piece) 2$P_L P_R$	Ψ(Pick Cherries) Σ(a-b)	Goodness of Split
Low	**Medium+High**	0.4	0.6	r: 0.018	r: 0.058	0.48	0.080	**0.0384**
				nr: 0.982	nr: 0.942			
Low+Medium	**High**	0.7	0.3	r: 0.030	r: 0.070	0.42	0.080	**0.0336**
				nr: 0.970	nr: 0.930			
Low+high	**Medium**	0.7	0.3	r: 0.040	r: 0.046	0.42	0.011	**0.0048**
				nr: 0.960	nr: 0.954			

$$P_L = \frac{\text{\# customers in Low}}{\text{All the customers}} = \frac{40000}{100000} = 0.4$$

$$P_R = \frac{\text{\# customers in Medium + High}}{\text{All the customers}} = \frac{60000}{100000} = 0.6$$

$$\Psi(\text{Large Piece}) = 2P_LP_R = 2 \times 0.4 \times 0.6 = 0.48$$

Remember, r represents responded and nr represents not-responded customers for our campaign's example.

$$\text{r: } P(k|L) = \frac{\text{\# customers responded in Low}}{\text{Total number of customers in Low}} = \frac{720}{40000} = 0.018$$

$$\text{nr: } P(k|L) = \frac{\text{\# customers not responded in Low}}{\text{Total number of customers in Low}} = \frac{39280}{40000} = 0.982$$

$$\Psi(\text{Pick Cherries}) = |P(r|L) - P(r|R)| + |P(nr|L) - P(nr|R)|$$

$$\therefore \Psi(\text{Pick Cherries}) = |0.018 - 0.058| + |0.982 - 0.942| = 0.080$$

$$\text{Goodness of split} = \Psi(\text{Large Piece}) \times \Psi(\text{Pick Cherries}) = 0.48 \times 0.080$$

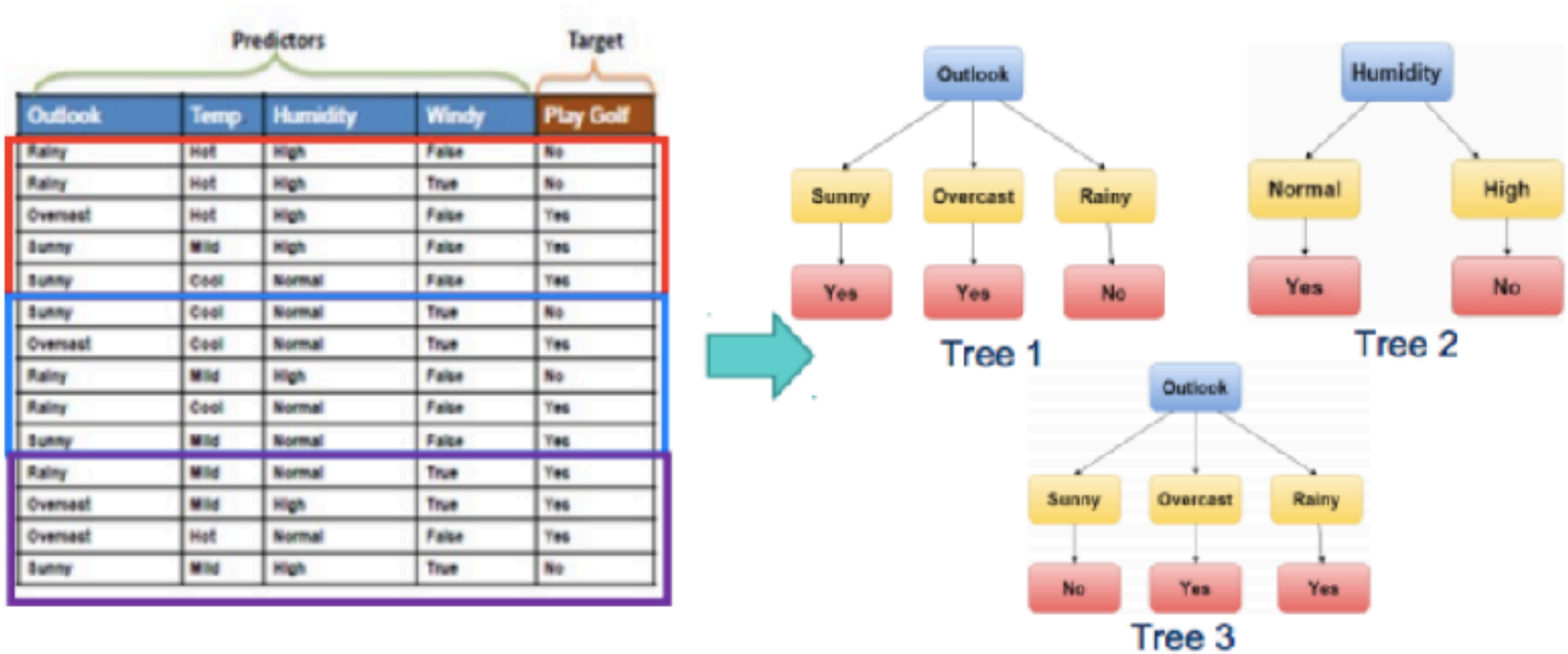
$$\therefore \text{Goodness of split} = 0.0384$$

Table 1

X1	X2	Y
2.771244718	1.784783929	0
1.728571309	1.169761413	0
3.678319846	2.81281357	0
3.961043357	2.61995032	0
2.999208922	2.209014212	0
7.497545867	3.162953546	1
9.00220326	3.339047188	1
7.444542326	0.476683375	1
10.12493903	3.234550982	1
6.642287351	3.319983761	1

7. Random Forest

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Mild	High	False	?



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Tree 1 : No
Tree 2 : No
Tree 3 : Yes

Yes : 1
No : 2

Result : No

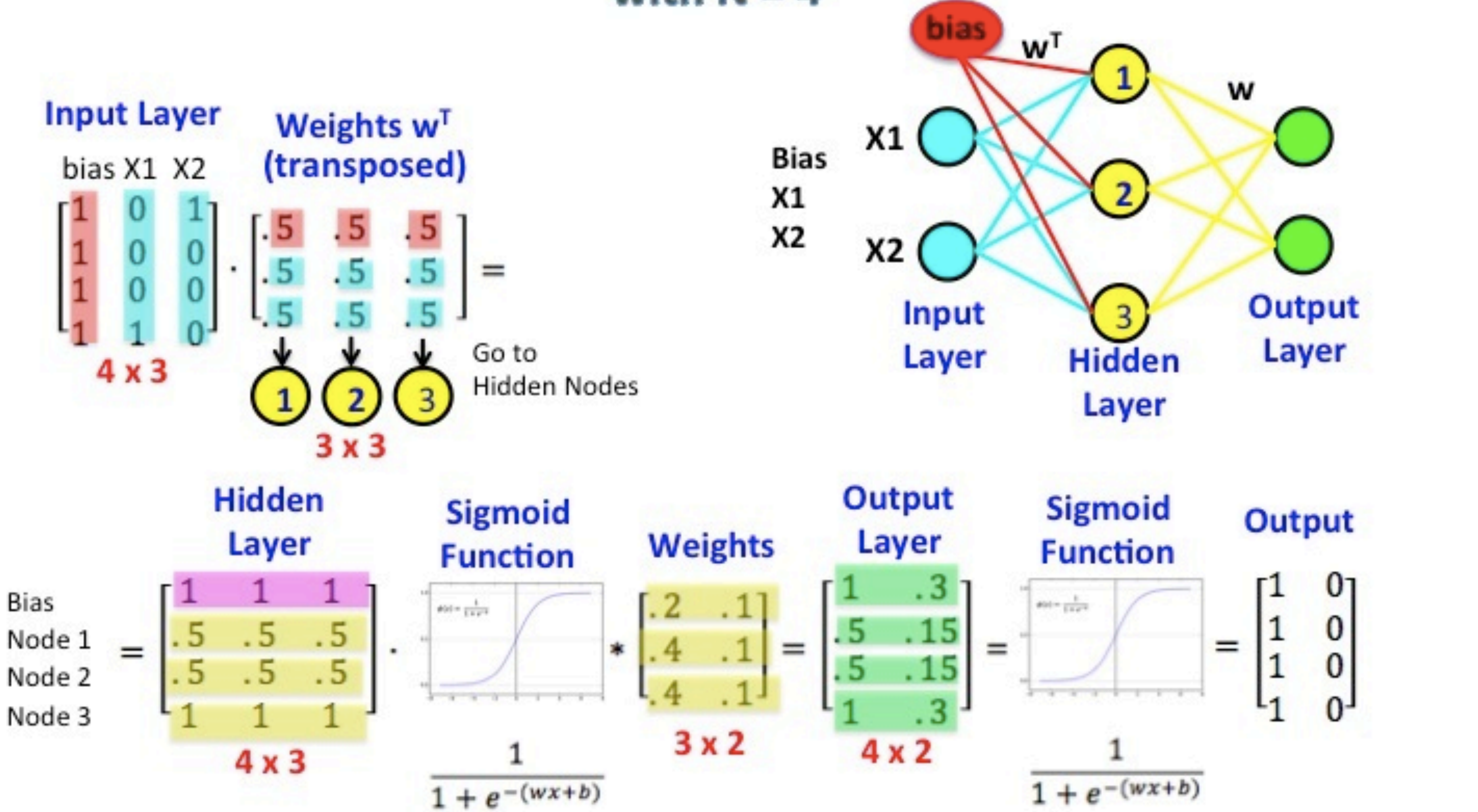
Table 1

X1	X2	Y
2.771244718	1.784783929	0
1.728571309	1.169761413	0
3.678319846	2.81281357	0
3.961043357	2.61995032	0
2.999208922	2.209014212	0
7.497545867	3.162953546	1
9.00220326	3.339047188	1
7.444542326	0.476683375	1
10.12493903	3.234550982	1
6.642287351	3.319983761	1

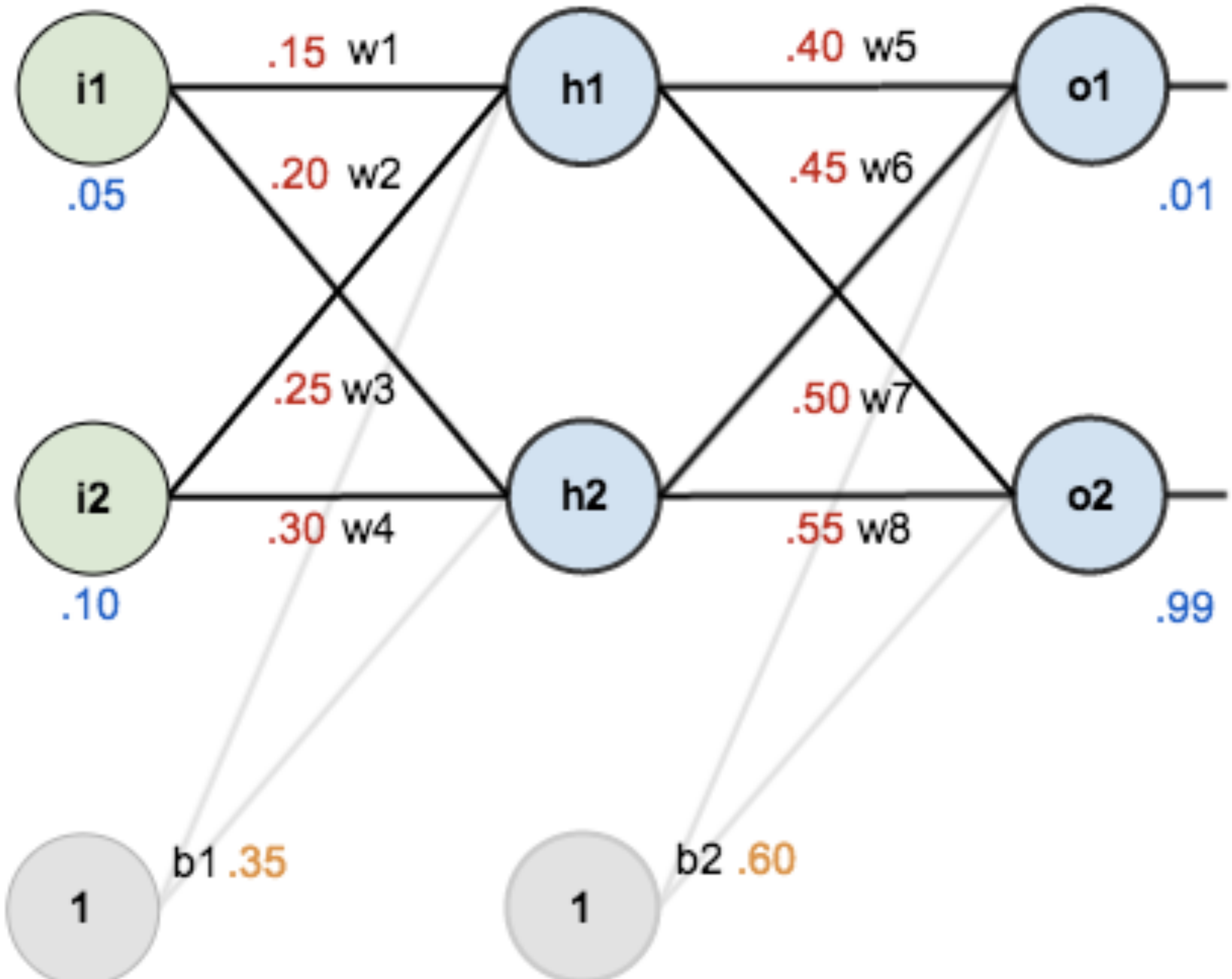
8. NN

Neural Networks

Color Guided Matrix Multiplication for a Binary Classification Task with N = 4



Rubens Zimbres



i1
.05
i2
.10
.15 w1
.20 w2
.25 w3
.30 w4
h1
h2
.40 w5
.45 w6
.50 w7
.55 w8
o1
.01
o2
.99
1
b1 .35
1
b2 .60

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}} = 0.593269992$$

$$out_{h2} = 0.596884378$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$net_{o1} = 0.4 * 0.593269992 + 0.45 * 0.596884378 + 0.6 * 1 = 1.105905967$$

$$out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$$

$$out_{o2} = 0.772928465$$

$\frac{1}{2}$ is included so that exponent is cancelled when we differentiate later on.

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2 = \frac{1}{2}(0.01 - 0.75136507)^2 = 0.274811083$$

$$E_{o2} = 0.023560026$$

$$E_{total} = E_{o1} + E_{o2} = 0.274811083 + 0.023560026 = 0.298371109$$

$$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = 2 * \frac{1}{2}(target_{o1} - out_{o1})^{2-1} * -1 + 0$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = -(target_{o1} - out_{o1}) = -(0.01 - 0.75136507) = 0.74136507$$

$$out_{o1} = \frac{1}{1+e^{-net_{o1}}}$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1}) = 0.75136507(1 - 0.75136507) = 0.186815602$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0 = out_{h1} = 0.593269992$$

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

9. SVM

$$B0 + (B1 \times X1) + (B2 \times X2) = 0$$

If the output value is greater than 1 it suggests that the training pattern was not a support vector.

$$b = (1 - \frac{1}{t}) \times b$$

If the output is less than 1 then it is assumed that the training instance is a support vector and must be updated to better explain the data.

$$b = (1 - \frac{1}{t}) \times b + \frac{1}{lambda \times t} \times (y \times x)$$

B1 = 0.0

B2 = 0.0

X1 = 2.327868056, X2 = 2.458016525, Y = 1

output=Y *(B1*X1)+(B2*X2)

output = -1*(0.0 * 2.327868056) + (0.0 * 2.458016525)

output = 0.0   //The output is less than 1.0

$$b = (1 - \frac{1}{t}) \times b + \frac{1}{lambda \times t} \times (y \times x)$$

B1 = (1-1/1) * 0.0 + 1/(0.45 * 1) * (1 ⇥ 2.327868056)

B1 = 5.173040124

B1 = (1-1/1) * 0.0 + 1/(0.45 * 1) * (1 ⇥ 2.458016525)

B2 = 5.462258944

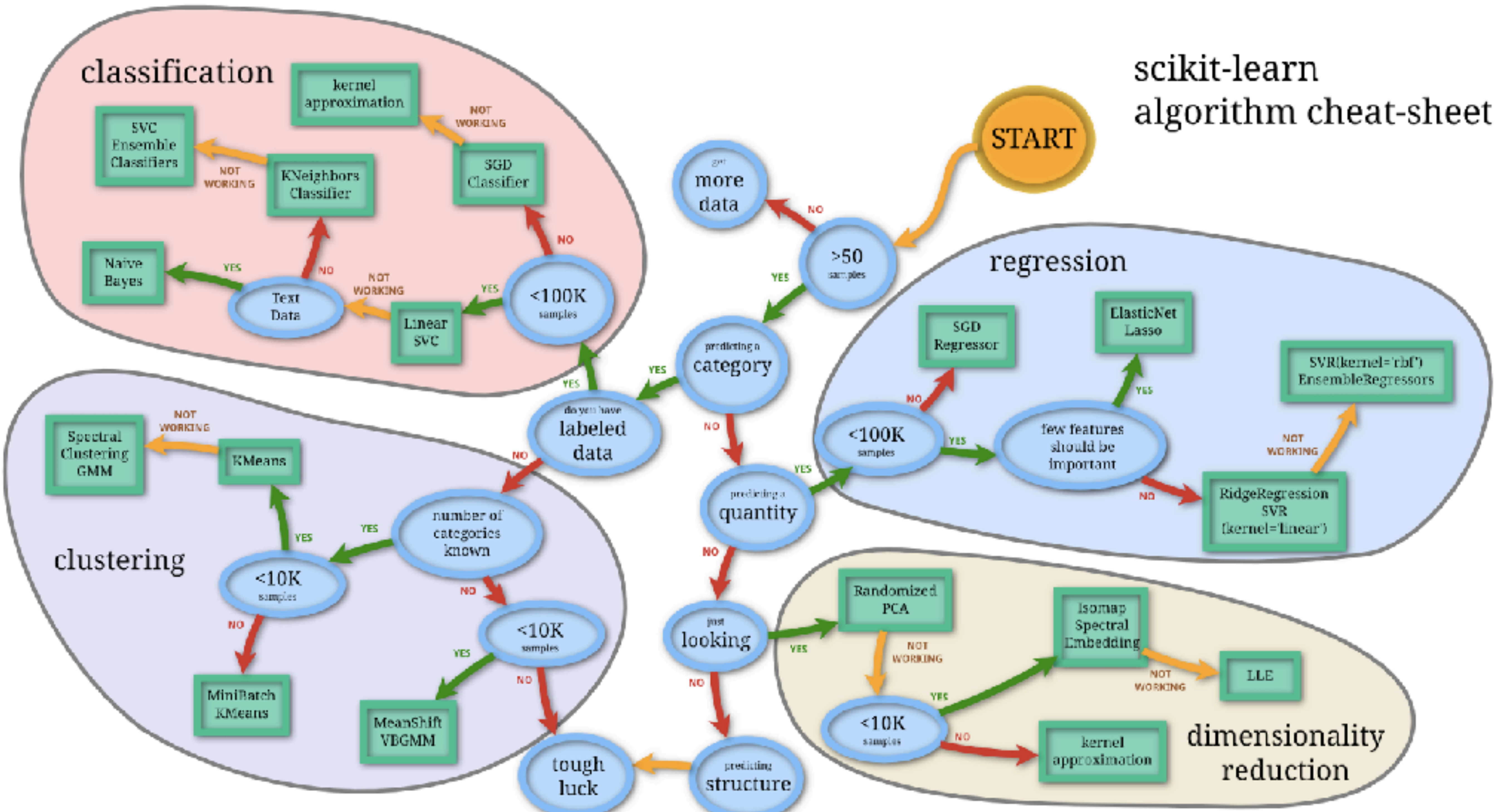
$B1 = -5.173040124$

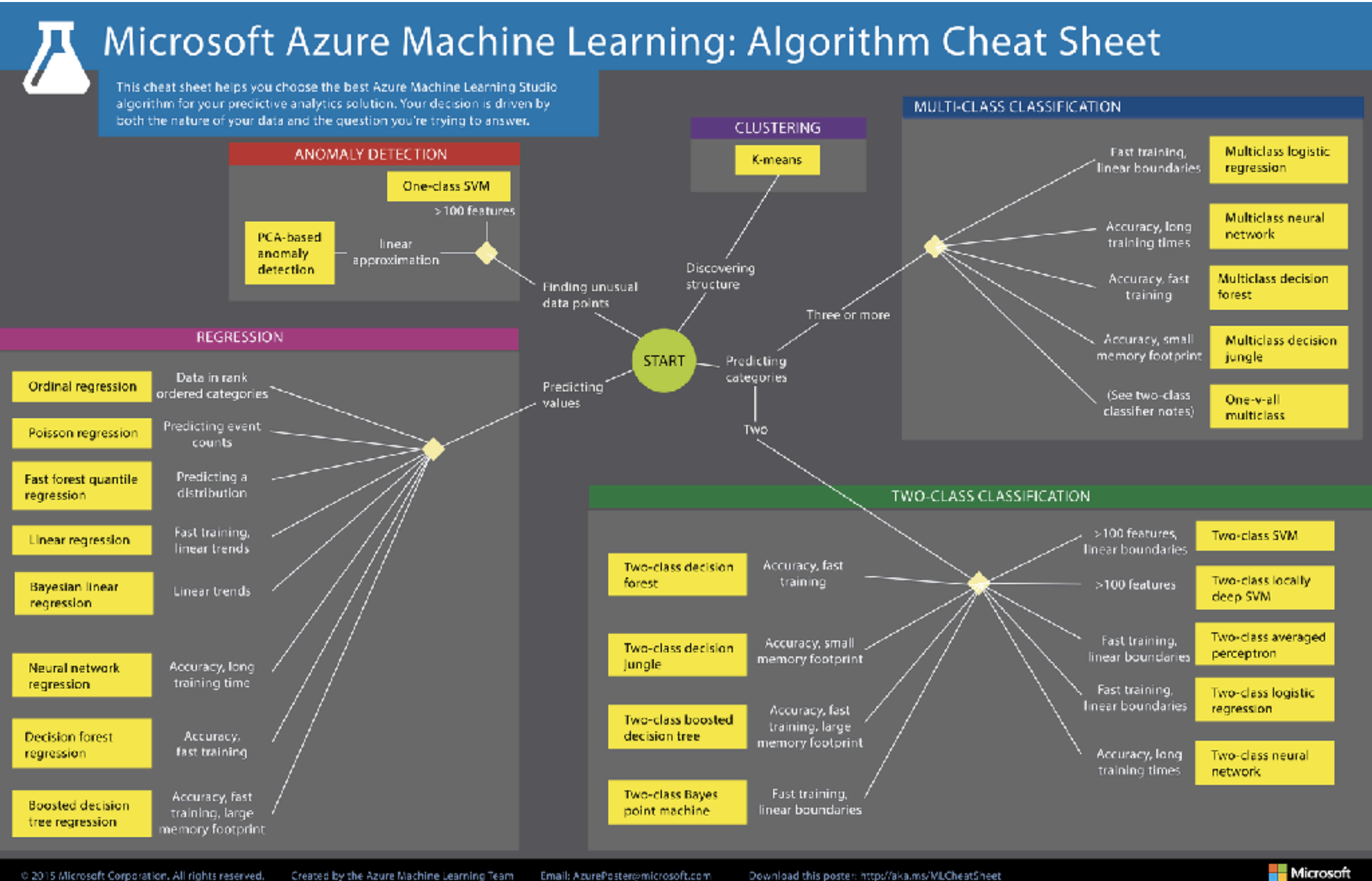
$B2 = -5.462258944$

$X1 = 2.327868056$, $X2 = 2.458016525$, $Y = 1$

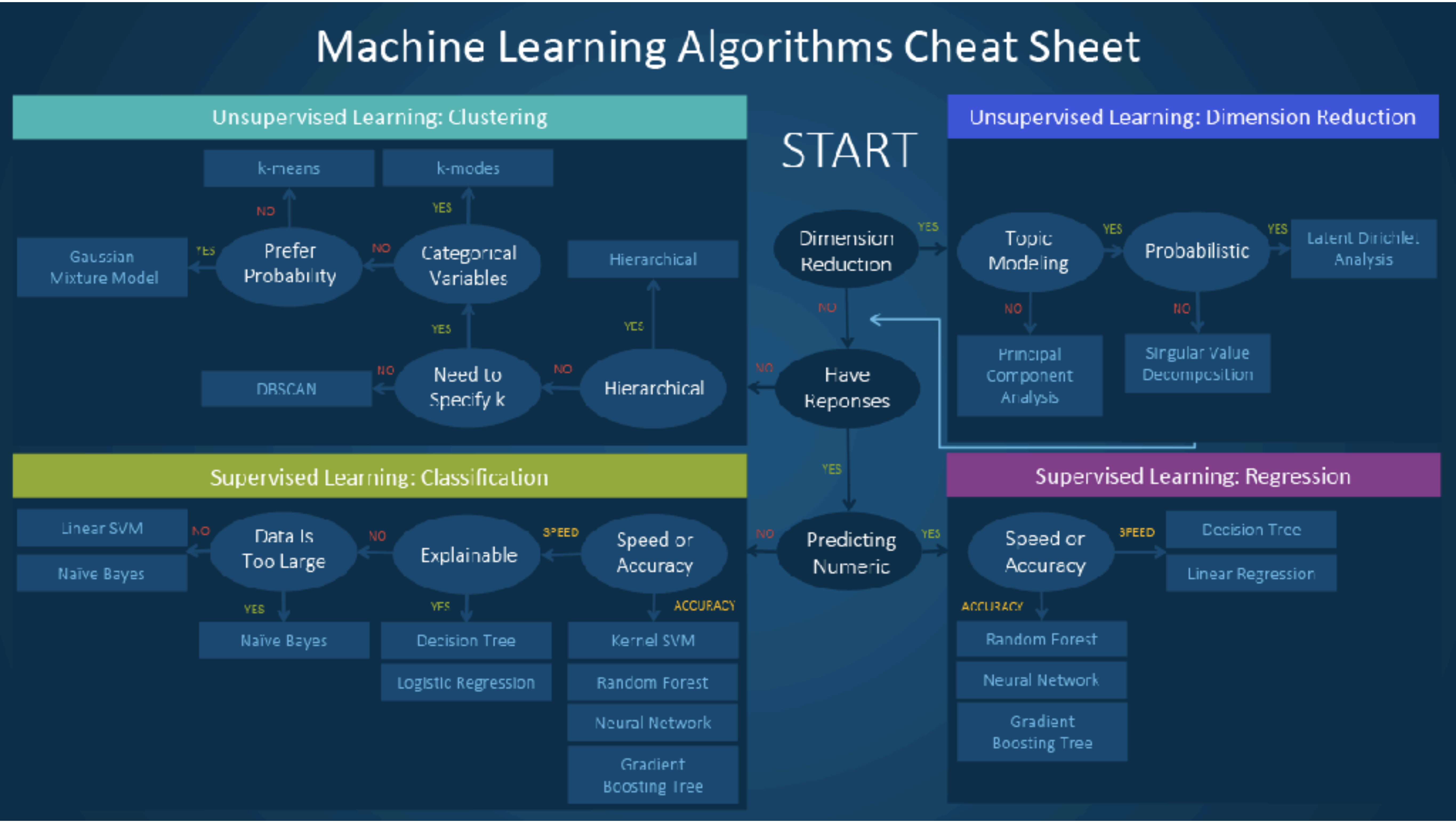
Table 1

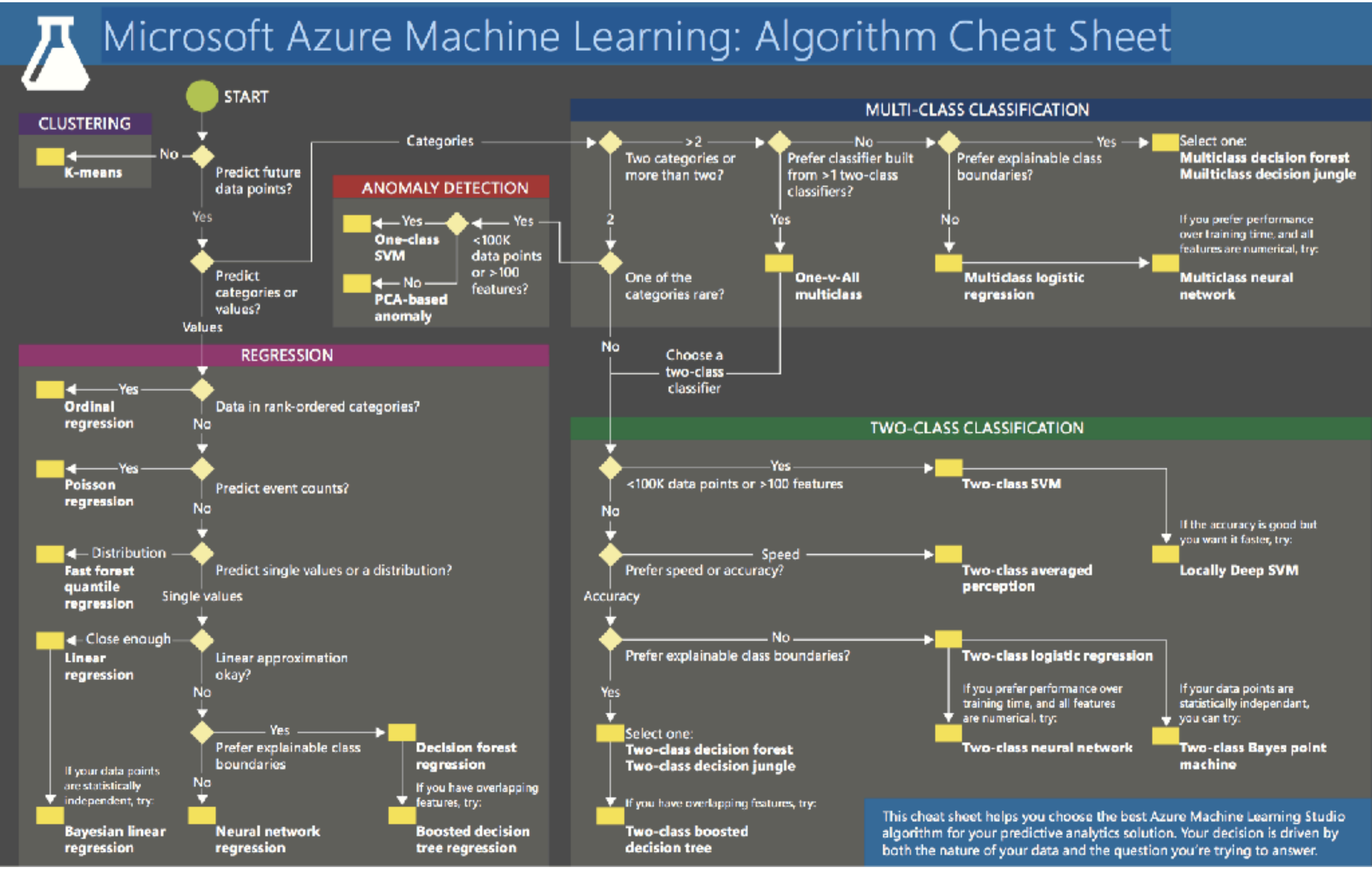
X1	X2	Y
2.327868056	2.458016525	-1
3.032830419	3.170770366	-1
4.485465382	3.696728111	-1
3.684815246	3.846846973	-1
2.283558563	1.853215997	-1
7.807521179	3.290132136	1
6.132998136	2.140563087	1
7.514829366	2.107056961	1
5.502385039	1.404002608	1
7.432932365	4.236232628	1


scikit-learn
algorithm cheat-sheet
START
>50 samples
get more data
predicting a category
do you have labeled data
<100K samples
Text Data
Naive Bayes
KNeighbors Classifier
SVC Ensemble Classifiers
Linear SVC
SGD Classifier
kernel approximation
classification
clustering
number of categories known
<10K samples
KMeans
Spectral Clustering GMM
MiniBatch KMeans
MeanShift VBGMM
tough luck
predicting a quantity
regression
SGD Regressor
ElasticNet Lasso
few features should be important
RidgeRegression SVR(kernel='linear')
SVR(kernel='rbf') EnsembleRegressors
just looking
predicting structure
Randomized PCA
Isomap Spectral Embedding
LLE
kernel approximation
dimensionality reduction
YES
NO
NOT WORKING



Microsoft Azure Machine Learning: Algorithm Cheat Sheet
This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.
START
ANOMALY DETECTION
Finding unusual data points
One-class SVM
>100 features
linear approximation
PCA-based anomaly detection
CLUSTERING
Discovering structure
K-means
MULTI-CLASS CLASSIFICATION
Predicting categories
Three or more
Fast training, linear boundaries
Multiclass logistic regression
Accuracy, long training times
Multiclass neural network
Accuracy, fast training
Multiclass decision forest
Accuracy, small memory footprint
Multiclass decision jungle
(See two-class classifier notes)
One-v-all multiclass
REGRESSION
Predicting values
Data in rank ordered categories
Ordinal regression
Predicting event counts
Poisson regression
Predicting a distribution
Fast forest quantile regression
Fast training, linear trends
Linear regression
Linear trends
Bayesian linear regression
Accuracy, long training time
Neural network regression
Accuracy, fast training
Decision forest regression
Accuracy, fast training, large memory footprint
Boosted decision tree regression
TWO-CLASS CLASSIFICATION
Two
>100 features, linear boundaries
Two-class SVM
>100 features
Two-class locally deep SVM
Fast training, linear boundaries
Two-class averaged perceptron
Fast training, linear boundaries
Two-class logistic regression
Accuracy, long training times
Two-class neural network
Accuracy, fast training
Two-class decision forest
Accuracy, small memory footprint
Two-class decision jungle
Accuracy, fast training, large memory footprint
Two-class boosted decision tree
Fast training, linear boundaries
Two-class Bayes point machine
© 2015 Microsoft Corporation. All rights reserved.
Created by the Azure Machine Learning Team
Email: AzurePoster@microsoft.com
Download this poster: http://aka.ms/MLCheatSheet
Microsoft


Machine Learning Algorithms Cheat Sheet
START
Unsupervised Learning: Clustering
k-means
k-modes
NO
YES
Gaussian Mixture Model
YES
Prefer Probability
NO
Categorical Variables
Hierarchical
YES
YES
DBSCAN
NO
Need to Specify k
NO
Hierarchical
Dimension Reduction
YES
NO
NO
Have Reponses
YES
Unsupervised Learning: Dimension Reduction
Topic Modeling
YES
Probabilistic
YES
Latent Dirichlet Analysis
NO
NO
Principal Component Analysis
Singular Value Decomposition
Supervised Learning: Classification
Linear SVM
Naïve Bayes
NO
Data Is Too Large
NO
Explainable
SPEED
Speed or Accuracy
YES
YES
ACCURACY
Naïve Bayes
Decision Tree
Logistic Regression
Kernel SVM
Random Forest
Neural Network
Gradient Boosting Tree
NO
Predicting Numeric
YES
Supervised Learning: Regression
Speed or Accuracy
SPEED
Decision Tree
Linear Regression
ACCURACY
Random Forest
Neural Network
Gradient Boosting Tree


Microsoft Azure Machine Learning: Algorithm Cheat Sheet
START
CLUSTERING
No
K-means
Predict future data points?
Yes
Predict categories or values?
Categories
Values
ANOMALY DETECTION
Yes
One-class SVM
Yes
<100K data points or >100 features?
No
PCA-based anomaly
MULTI-CLASS CLASSIFICATION
Two categories or more than two?
>2
Prefer classifier built from >1 two-class classifiers?
No
Prefer explainable class boundaries?
Yes
Select one:
Multiclass decision forest
Muilticlass decision jungle
2
Yes
No
If you prefer performance over training time, and all features are numerical, try:
One of the categories rare?
One-v-All multiclass
Multiclass logistic regression
Multiclass neural network
No
Choose a two-class classifier
REGRESSION
Yes
Ordinal regression
Data in rank-ordered categories?
No
Yes
Poisson regression
Predict event counts?
No
Distribution
Fast forest quantile regression
Predict single values or a distribution?
Single values
Close enough
Linear regression
Linear approximation okay?
No
Yes
Prefer explainable class boundaries
Decision forest regression
If your data points are statistically independent, try:
No
If you have overlapping features, try:
Bayesian linear regression
Neural network regression
Boosted decision tree regression
TWO-CLASS CLASSIFICATION
Yes
<100K data points or >100 features
Two-class SVM
If the accuracy is good but you want it faster, try:
No
Speed
Prefer speed or accuracy?
Two-class averaged perception
Locally Deep SVM
Accuracy
No
Prefer explainable class boundaries?
Two-class logistic regression
If you prefer performance over training time, and all features are numerical, try:
If your data points are statistically independant, you can try:
Yes
Select one:
Two-class decision forest
Two-class decision jungle
Two-class neural network
Two-class Bayes point machine
If you have overlapping features, try:
Two-class boosted decision tree
This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.