Logistic Regression

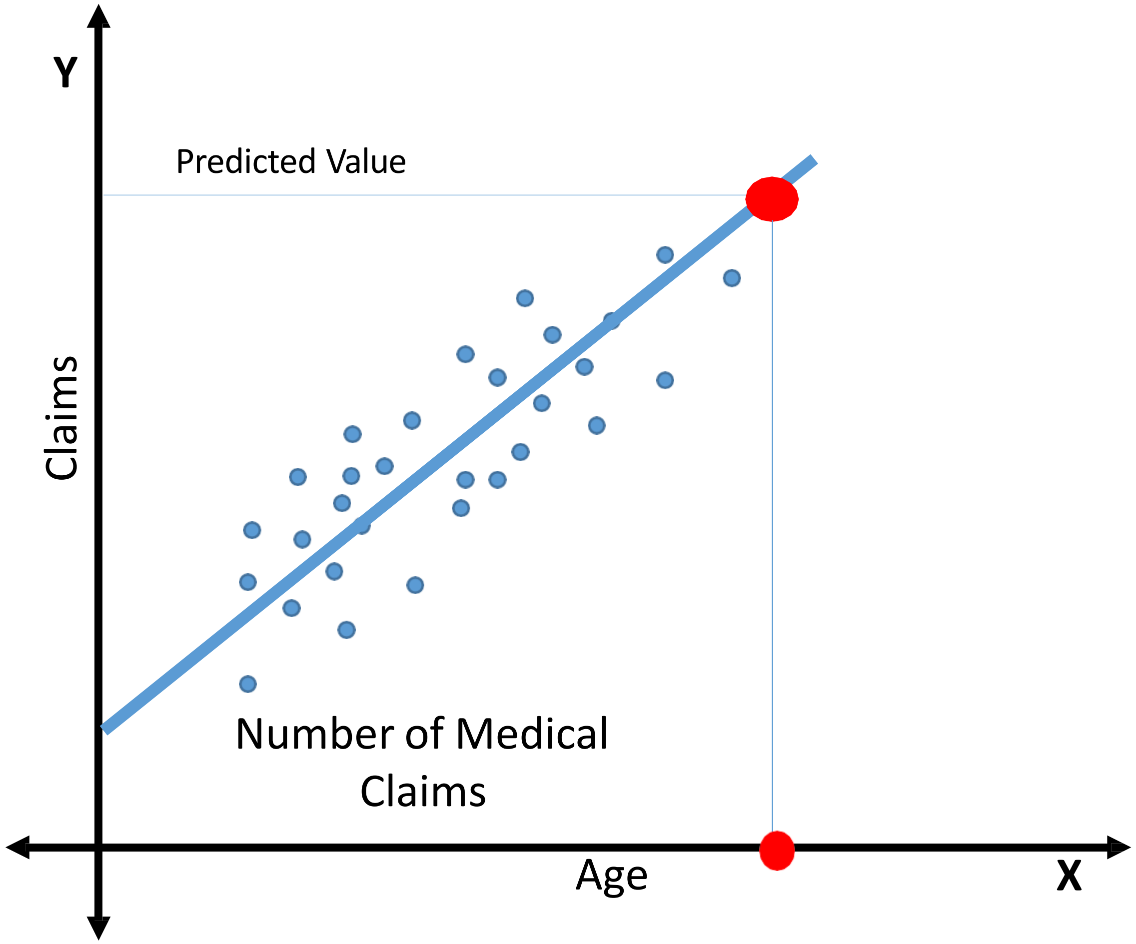
What is Logistics Regression?

- Used to predict the probability of an outcome
- Can be binary – Yes/No or Multiple
- Supervised learning method
- Must provide a dataset that already contains the outcomes to train the model.

Understanding the Logistic Regression

$$y = b_0 + \ b_1 x$$

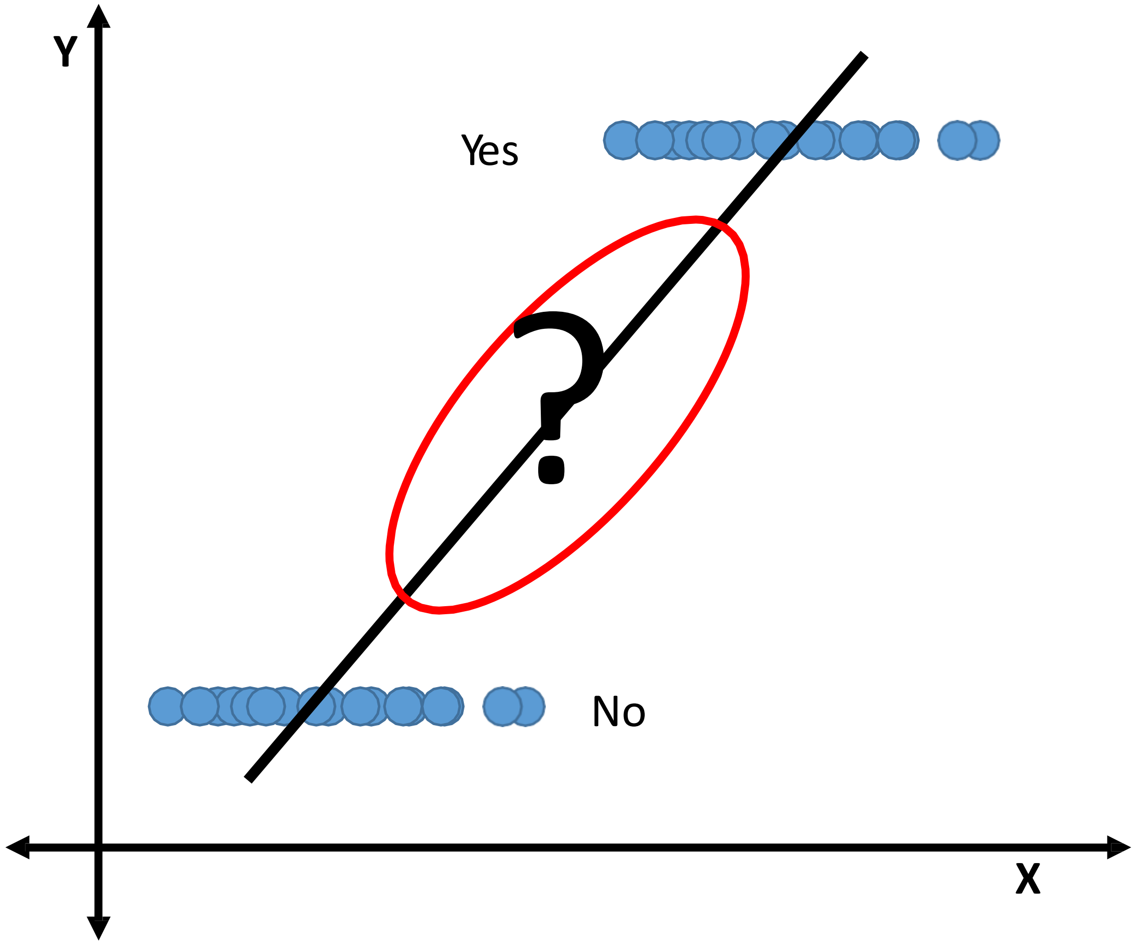
No of claims = 18 + b1(age)



Simple Linear Regression

Logistic Regression?

- Outcome is categorical
- Will this customer buy my product?
- What is the probability of this customer buying this product?



Logistic Regression?

- Probability needs to satisfy two basic conditions
  - Always positive i.e. > 0
  - Always less than or equal to 1

$$y = b_0 + b_1 x \xrightarrow{\text{Always Positive}} e^{y} \xrightarrow{\text{Make it less than 1}} \frac{e^{y}}{e^{y} + 1}$$

Logistic Regression?

$$y = b_0 + b_1 x \xrightarrow{\text{Always Positive}} e^{y} \xrightarrow{\text{Make it less than 1}} \frac{e^{y}}{e^{y}+1}$$

$$P = \frac{e^{y}}{e^{y}+1}$$

Logistic Regression?

$$P = \frac{e^{y}}{e^{y} + 1}$$

$$Q = 1 - P = 1 - \frac{e^{y}}{e^{y} + 1} = \frac{e^{y} + 1 - e^{y}}{e^{y} + 1} = \frac{1}{e^{y} + 1}$$

Probability of Failure

Logistic Regression?

$$P = \frac{e^{y}}{e^{y} + 1}$$

$$\text{Odds} = \frac{P(\text{Success})}{P(\text{Failure})}$$

$$1 - P = \frac{1}{e^{y} + 1}$$

Logistic Regression?

$$P = \frac{e^{y}}{e^{y} + 1}$$

$$1 - P = \frac{1}{e^{y} + 1}$$

$$\text{Odds} = \frac{P}{1 - P} = \frac{\dfrac{e^{y}}{\cancel{e^{y} + 1}}}{\dfrac{1}{\cancel{e^{y} + 1}}} = e^{y}$$

Logistic Regression?

$$P = \frac{e^{y}}{e^{y} + 1}$$

$$1 - P = \frac{1}{e^{y} + 1}$$

$$\frac{P}{1 - P} = e^{y}$$

Logistic Regression?

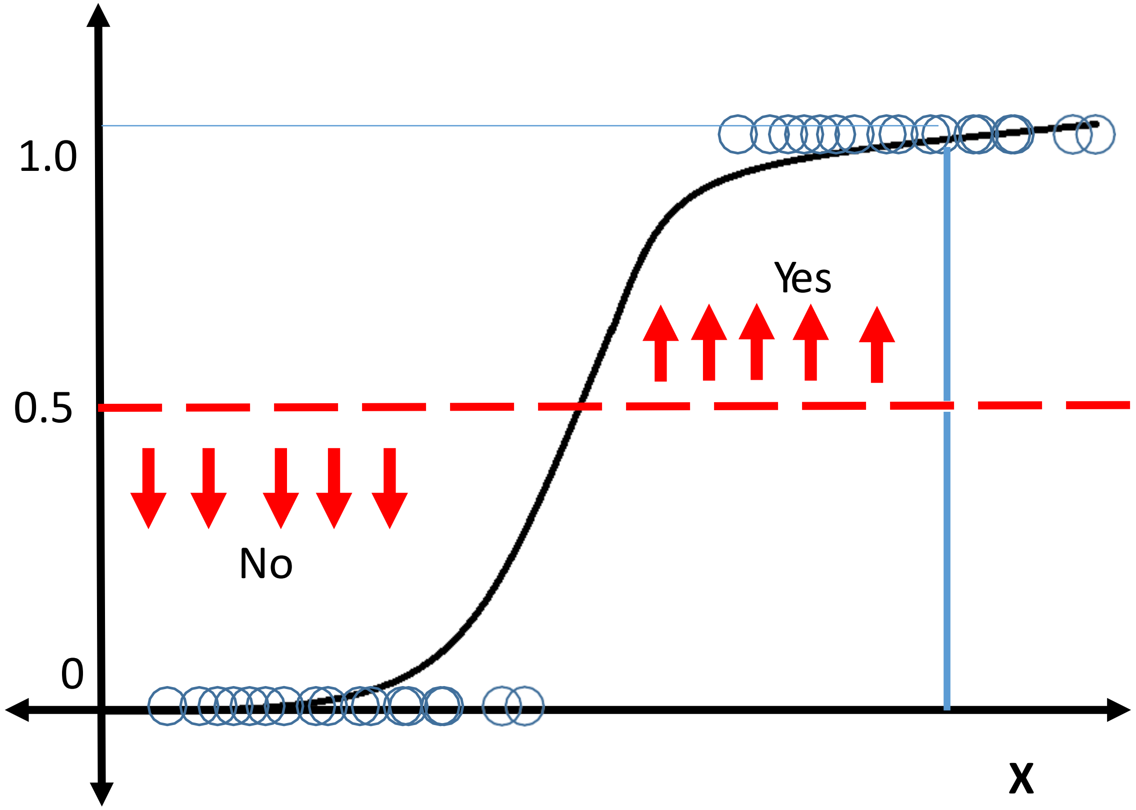
$$P = \frac{e^{y}}{e^{y} + 1}$$

$$1 - P = \frac{1}{e^{y} + 1}$$

$$\log\left(\frac{P}{1 - P}\right) = y = (b_0 + b_1 x)$$

Plotting Logistic Regression

$$\log\left(\frac{P}{1-P}\right) = (b_0 + b_1 x)$$



Threshold Value of 50% probability
> 0.5 Success
< 0.5 Failure

Stratification (stratify)

Split without Stratification

Index	Status
1	Yes
2	No
3	No
4	Yes
5	Yes
6	Yes
7	No
8	No
9	Yes
10	Yes

Yes = 6 No = 4

Train

Index	Status
1	Yes
4	Yes
5	Yes
6	Yes
9	Yes
10	Yes

Test

Index	Status
2	No
3	No
7	No
8	No

All Nos

Split without Stratification

Index	Status
1	Yes
2	No
3	No
4	Yes
5	Yes
6	Yes
7	No
8	No
9	Yes
10	Yes

Train

Index	Status
1	Yes
2	No
3	No
5	Yes
7	No
8	No

Test

Index	Status
4	Yes
6	Yes
9	Yes
10	Yes

All Yes

Yes = 6 No = 4

Split without Stratification

Index	Status
1	Yes
2	No
3	No
4	Yes
5	Yes
6	Yes
7	No
8	No
9	Yes
10	Yes

Yes = 6 No = 4

Train

Index	Status
1	Yes
2	No
3	No
5	Yes
7	No
9	Yes

Test

Index	Status
4	Yes
6	Yes
8	No
10	Yes

Imbalanced

Stratification (stratify) with split at 50%

Index	Status
1	Yes
2	No
3	No
4	Yes
5	Yes
6	Yes
7	No
8	No
9	Yes
10	Yes

Yes = 6 No = 4

Train

Index	Status
1	Yes
2	No
3	No
5	Yes
9	Yes

Yes = 6 * 0.5 = 3

No = 4 * 0.5 = 2

Test

Index	Status
4	Yes
6	Yes
7	No
8	No
10	Yes

Yes = 6 * 0.5 = 3

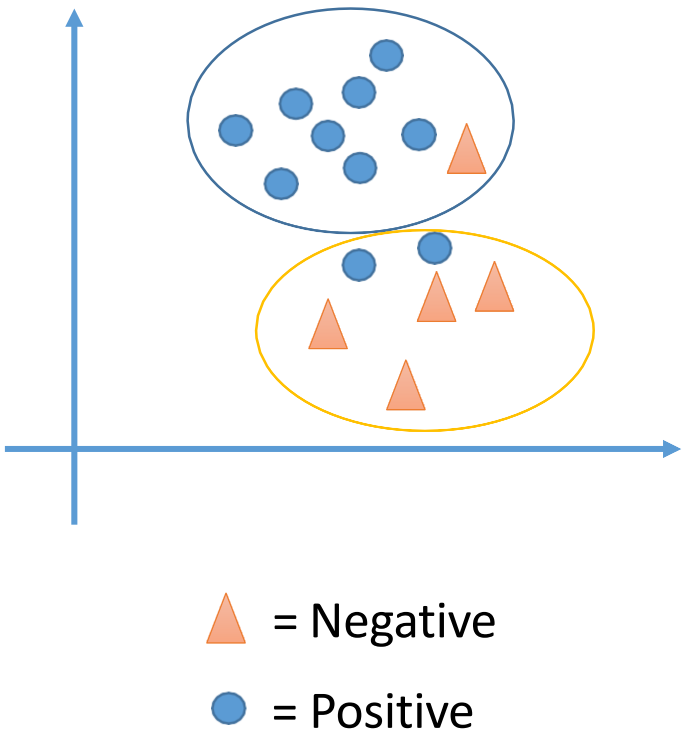
No = 4 * 0.5 = 2

Understanding the results

Prediction Outcome

	Predicted Negatives	Predicted Positives
Actual Negatives	True Negatives	False Positives
Actual Positives	False Negatives	True Positives

	Predicted Negatives	Predicted Positives	
Actual Negatives	4	1	5
Actual Positives	2	8	10
	6	9	



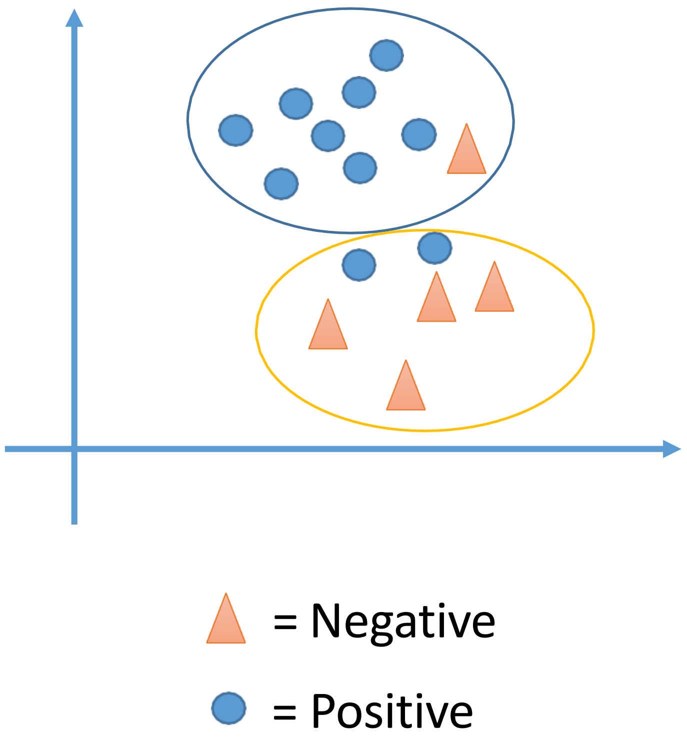
Accuracy – Proportions of total number of correct results

Accuracy = (8 + 4) / 15 = 0.8 or 80%

Prediction Outcome

	Predicted Negatives	Predicted Positives
Actual Negatives	True Negatives	False Positives
Actual Positives	False Negatives	True Positives

	Predicted Negatives	Predicted Positives	
Actual Negatives	4	1	5
Actual Positives	2	8	10
	6	9	



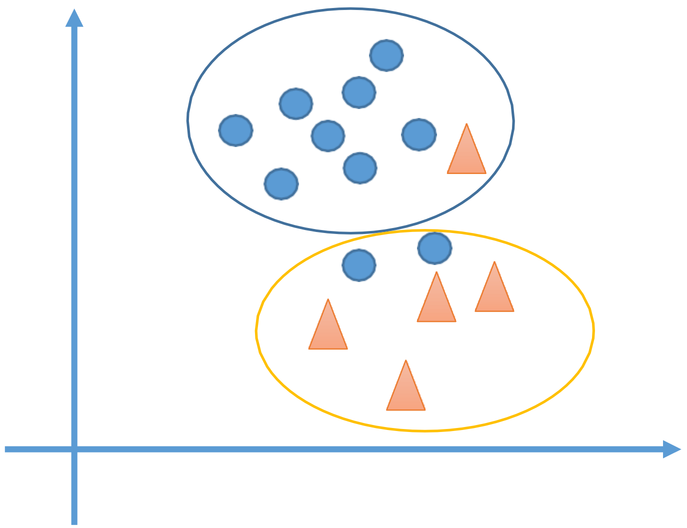
Precision – Proportion of correct positive results out of all predicted positive results

Precision = 8 / 9 = 0.889 or 88.9%

Prediction Outcome

	Predicted Negatives	Predicted Positives
Actual Negatives	True Negatives	False Positives
Actual Positives	False Negatives	True Positives

	Predicted Negatives	Predicted Positives	
Actual Negatives	4	1	5
Actual Positives	2	8	10
	6	9	



= Negative

= Positive

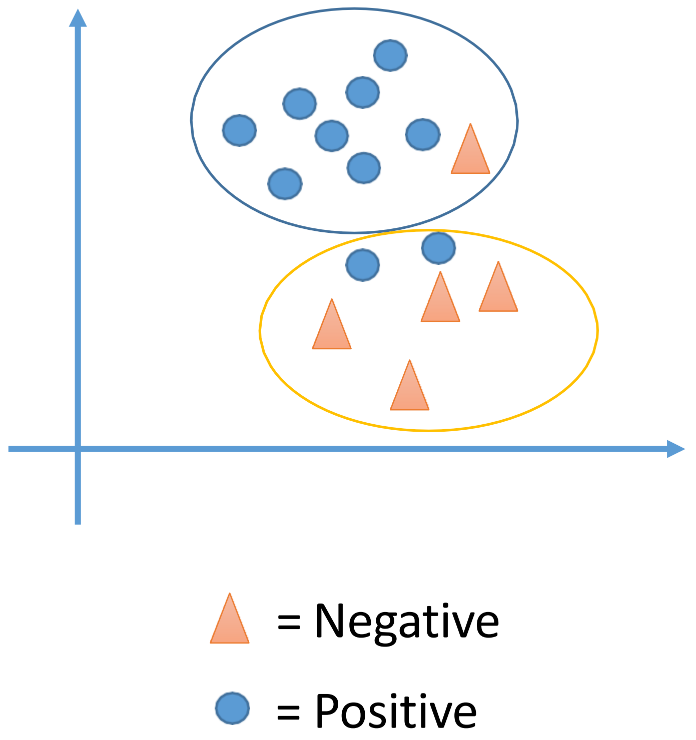
Recall – Proportion of actual positive cases

Recall = 8 / (8 + 2) = 0.8 or 80%

Prediction Outcome

	Predicted Negatives	Predicted Positives
Actual Negatives	True Negatives	False Positives
Actual Positives	False Negatives	True Positives

	Predicted Negatives	Predicted Positives	
Actual Negatives	4	1	5
Actual Positives	2	8	10
	6	9	



F1-Score – Weighted Average (Harmonic Mean) of Precision and Recall

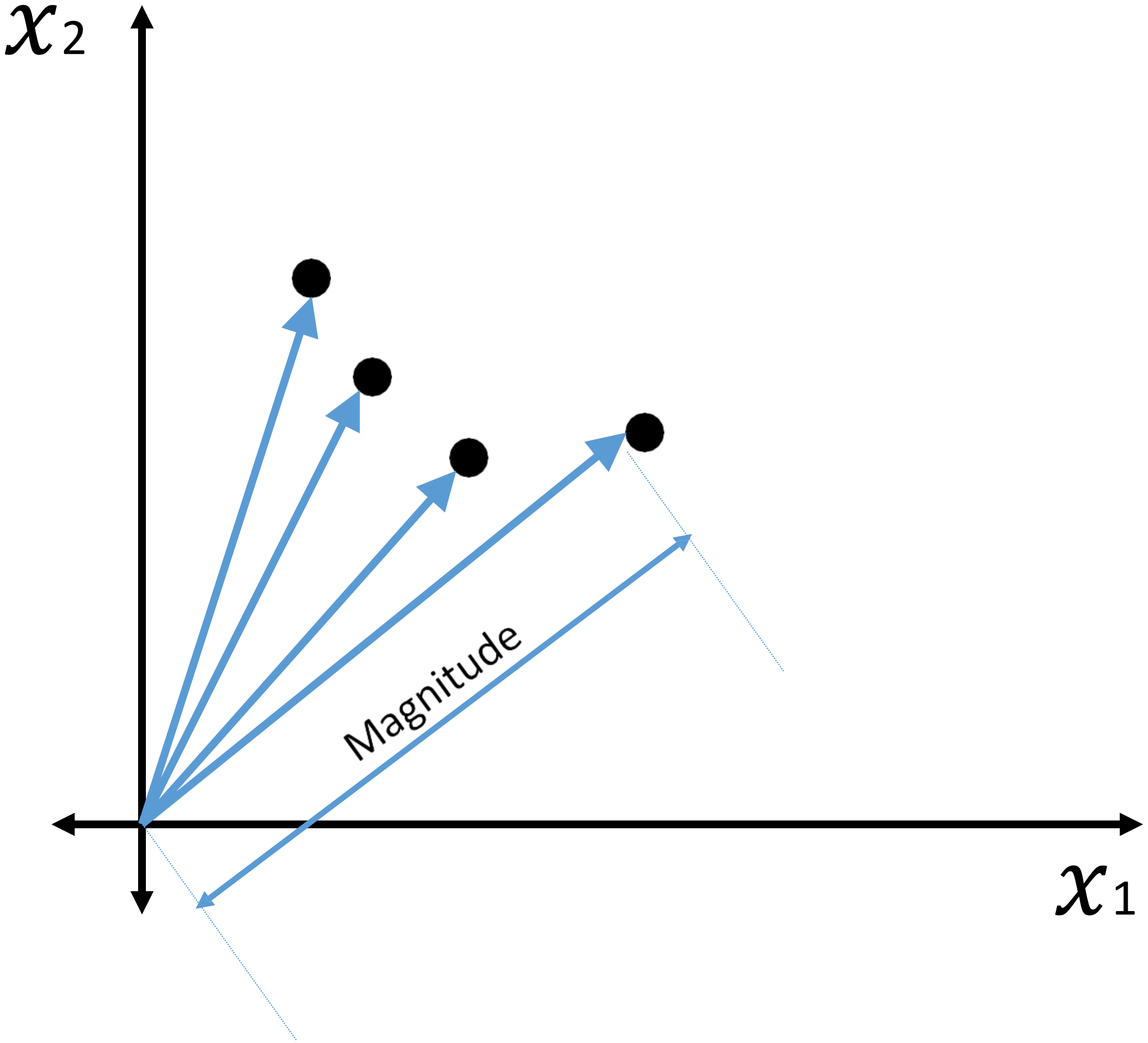
F1Score = 2 * Precision * Recall / (Precision + Recall) = 0.84

Support Vector Machine

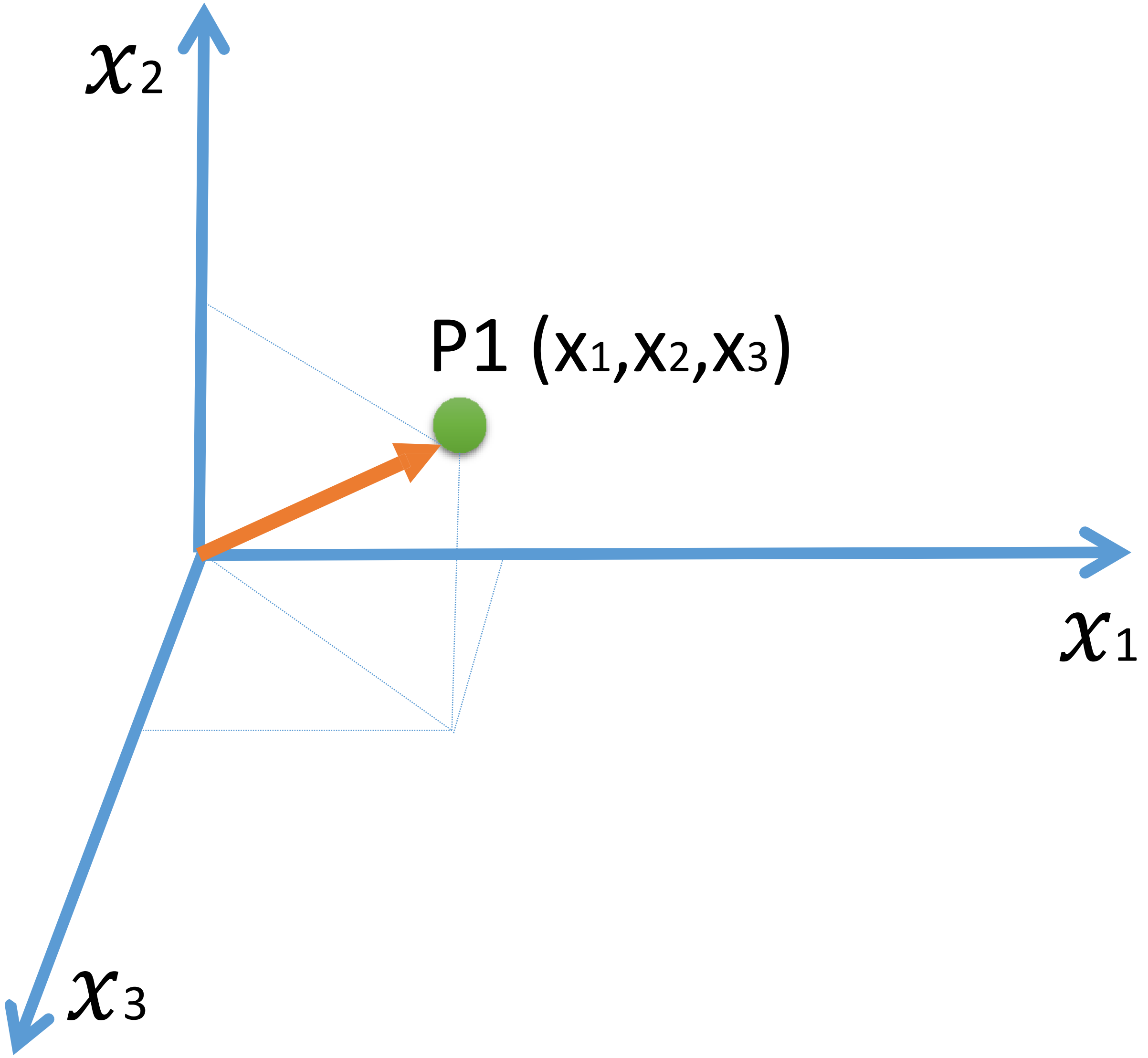
What is SVM?

- Supervised Learning Algorithm
- Can be used for both Regression as well as Classification
- The observations are separated by a hyperplane in the space

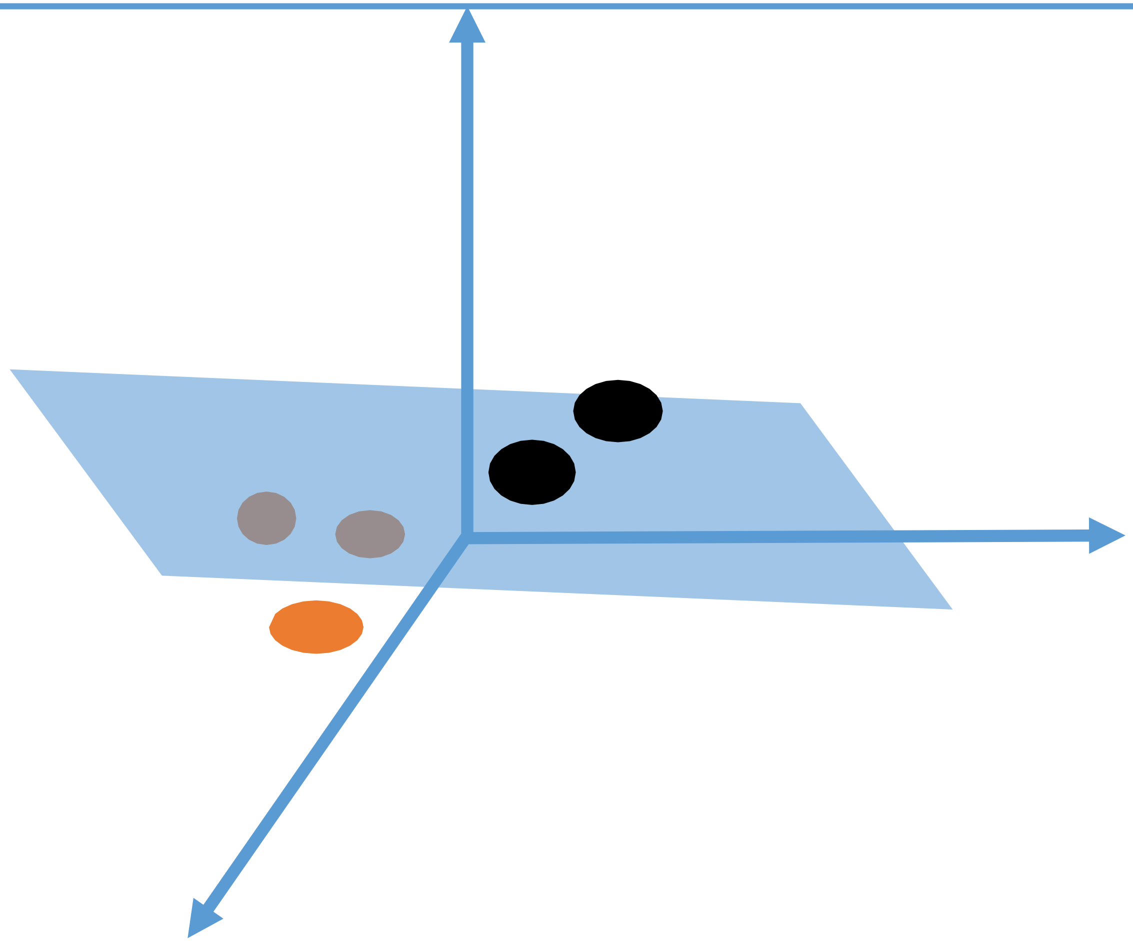
Vectors



Vectors



Hyperplane



1D – Point

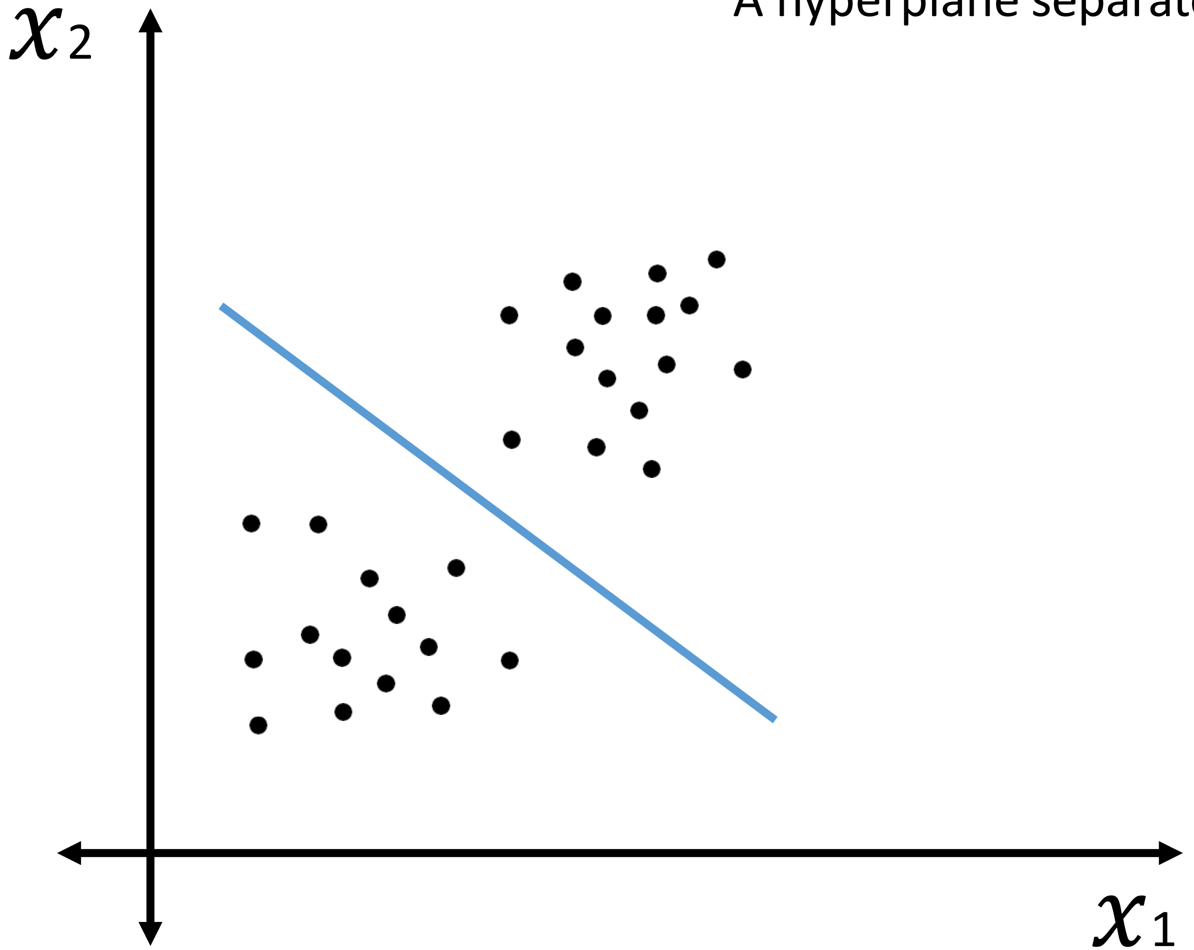
2D – Line

3D – Plane

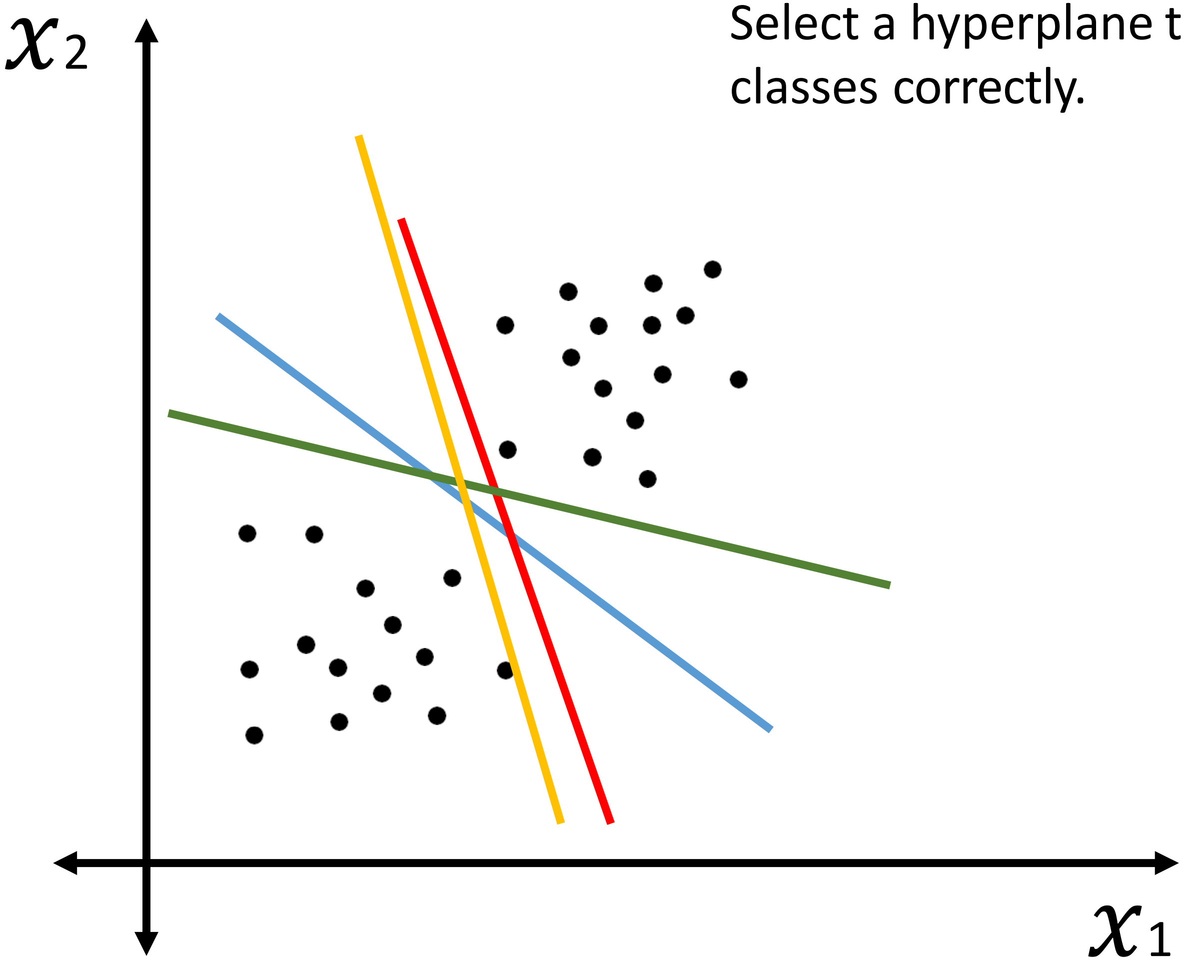
4D – Hyperplane

Hyperplane

A hyperplane separates the two classes.



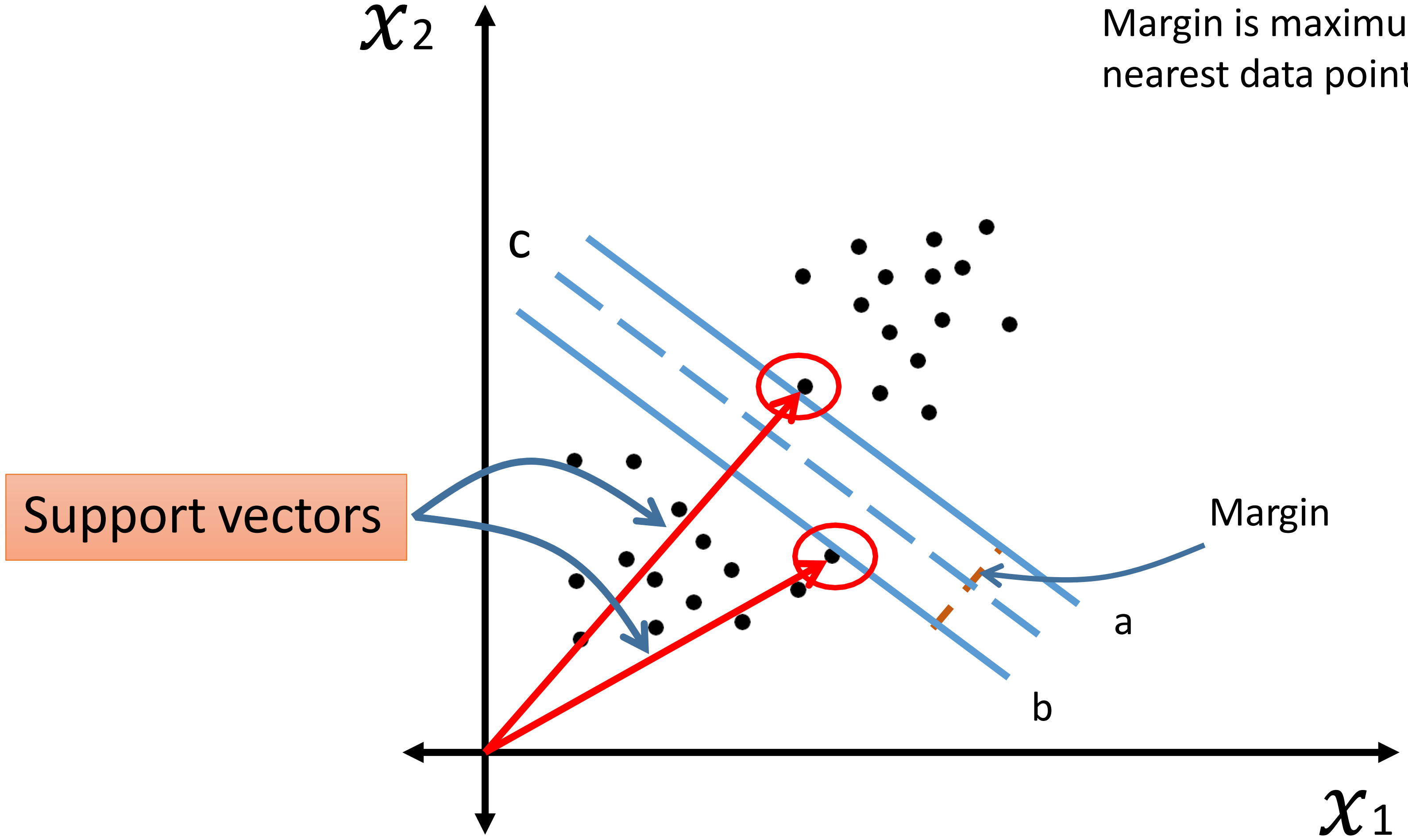
Choosing a Hyperplane



Select a hyperplane that separates the two classes correctly.

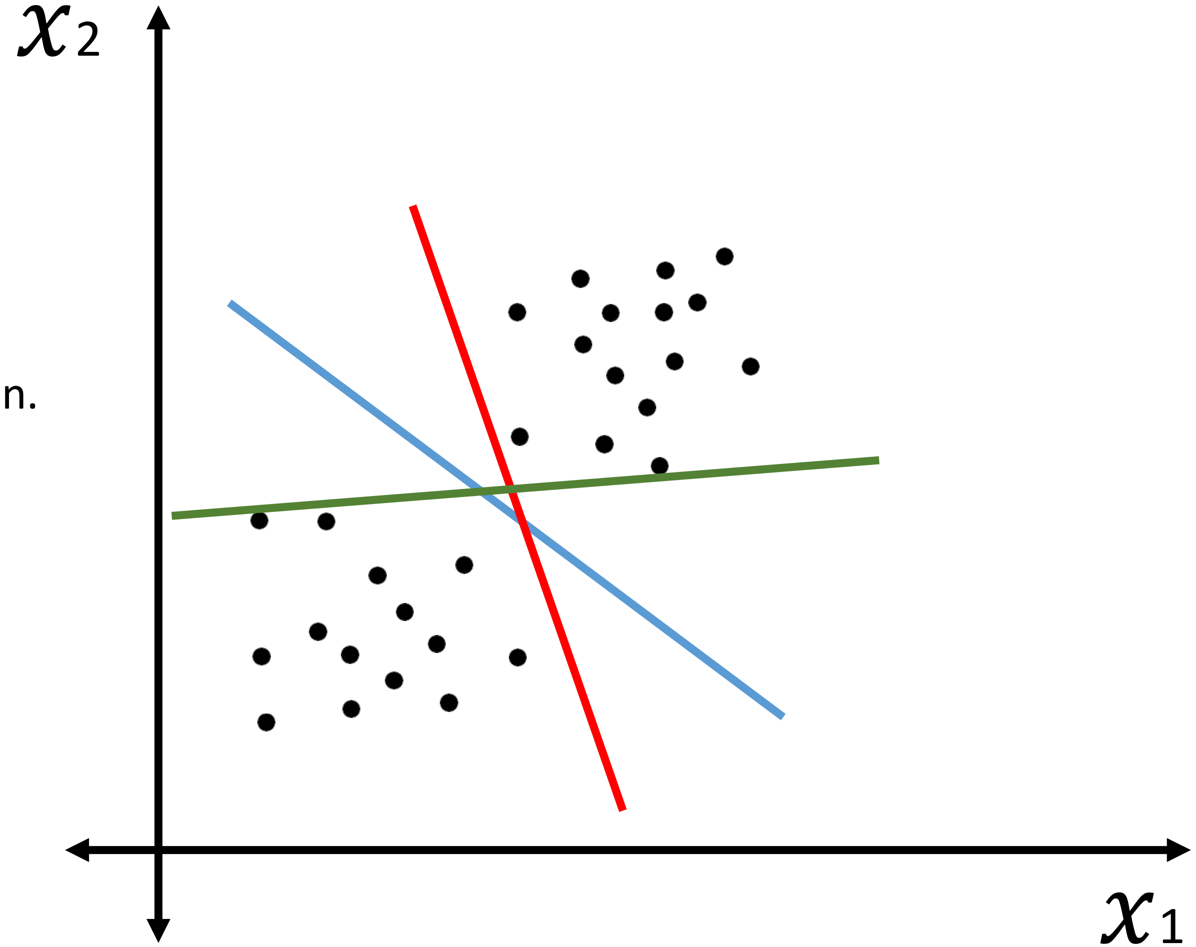
Select the right hyperplane

Margin is maximum distance between the nearest data points and the hyperplane.



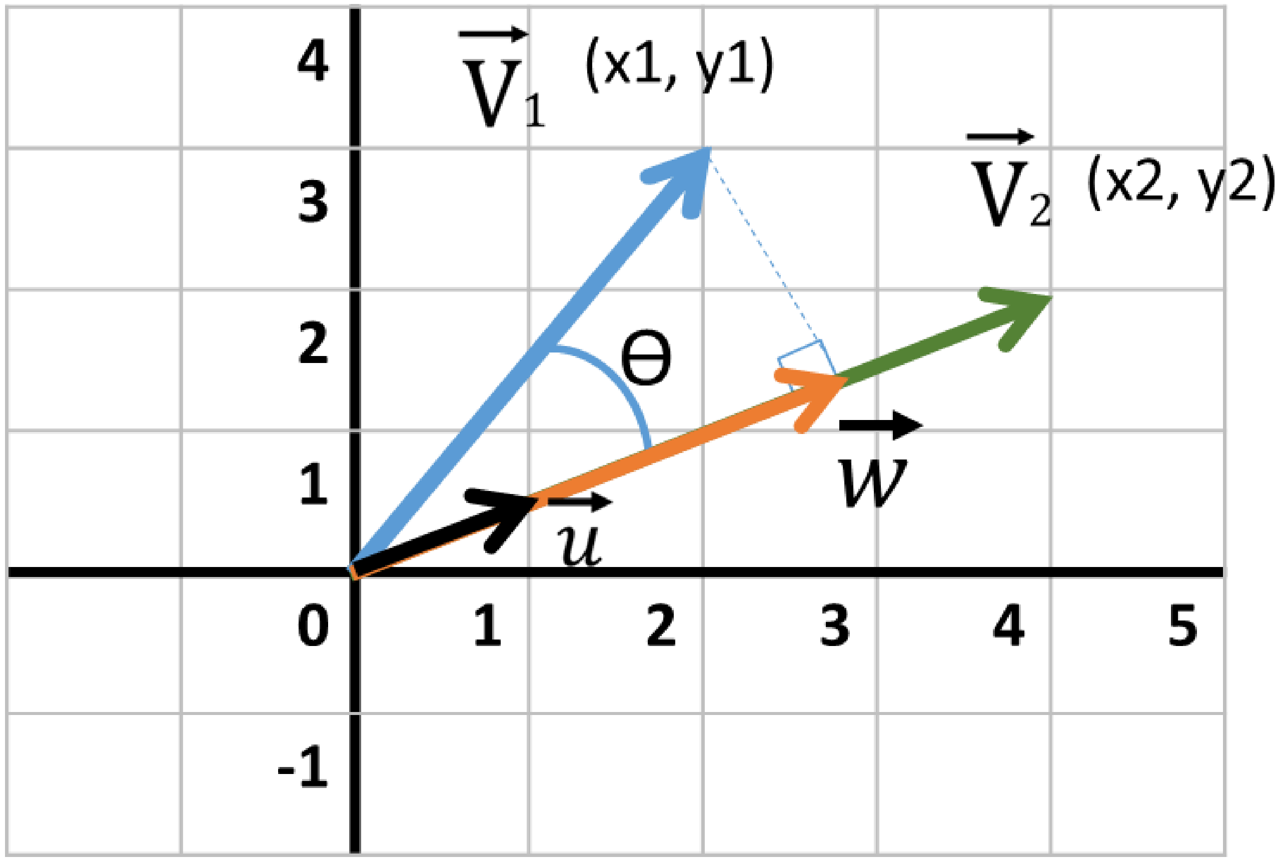
What comes first?

Identifying the accurate classes comes first before margin calculation.

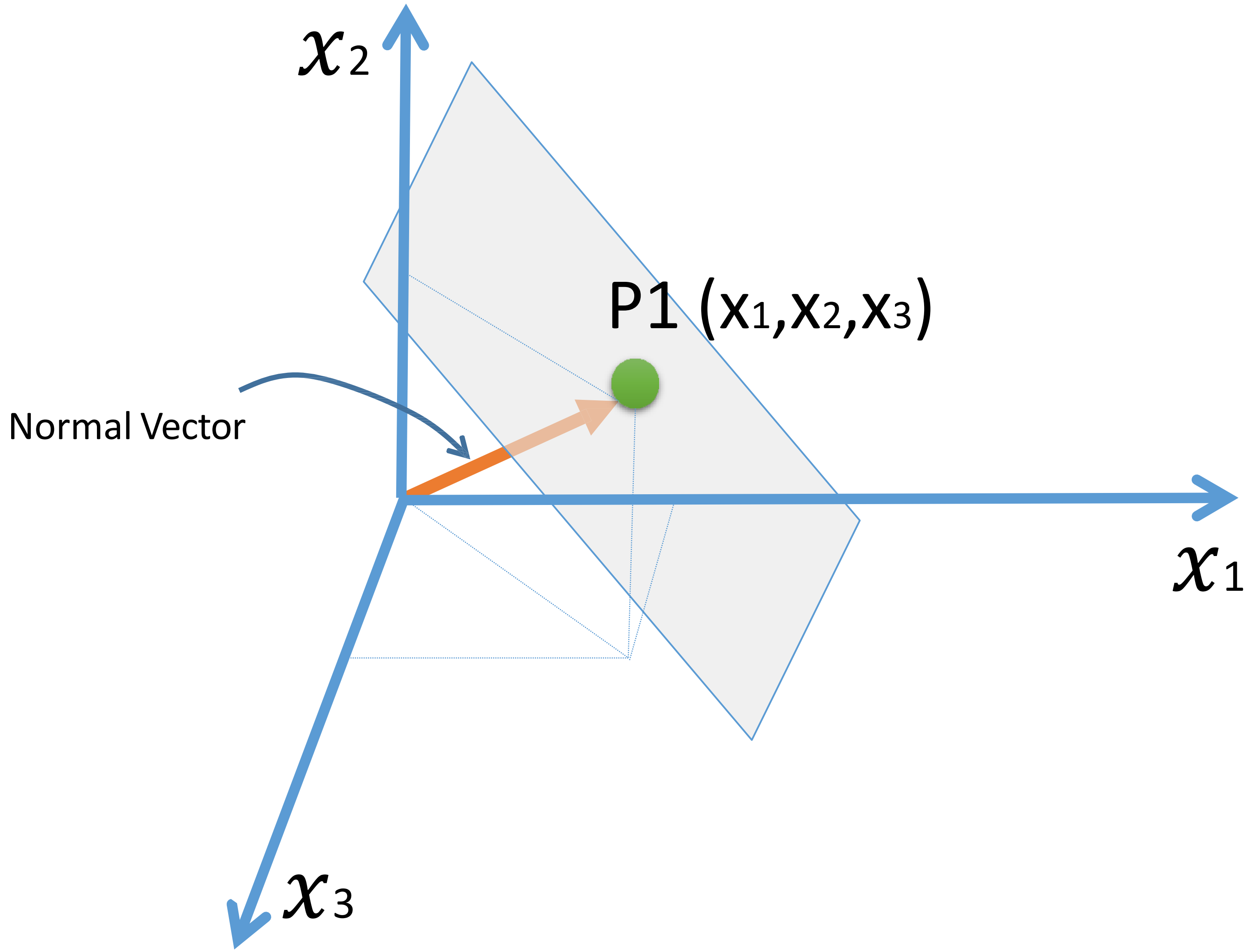


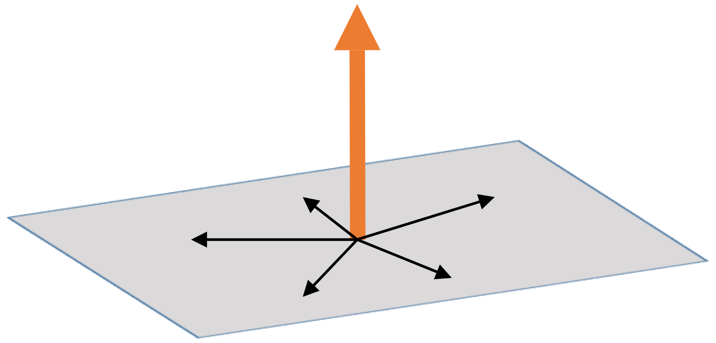
Orthogonal Projection of Vector

$$\|w\| = \overrightarrow{V_1} \cdot \overrightarrow{u}$$

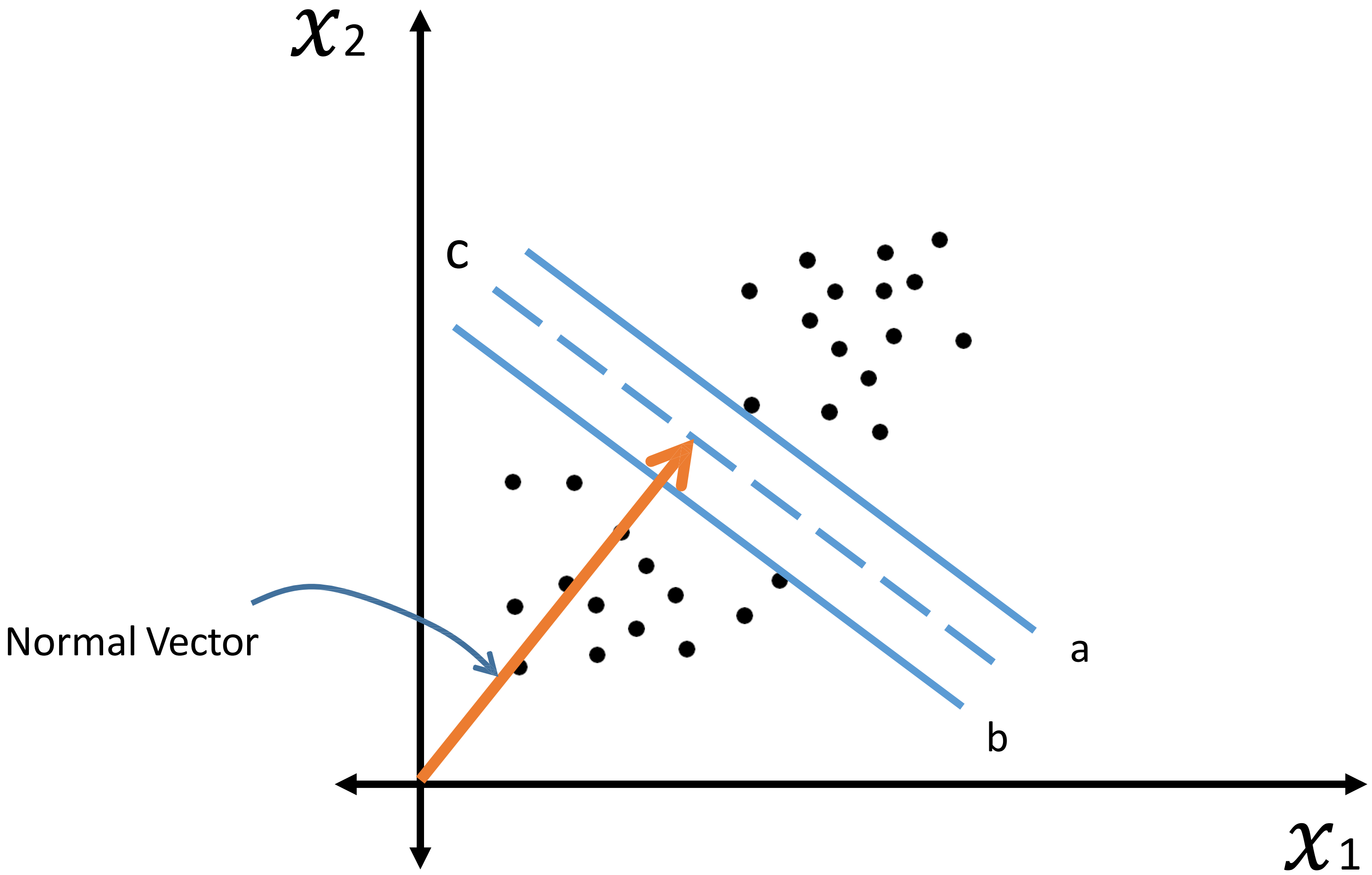


Normal Vector of a plane



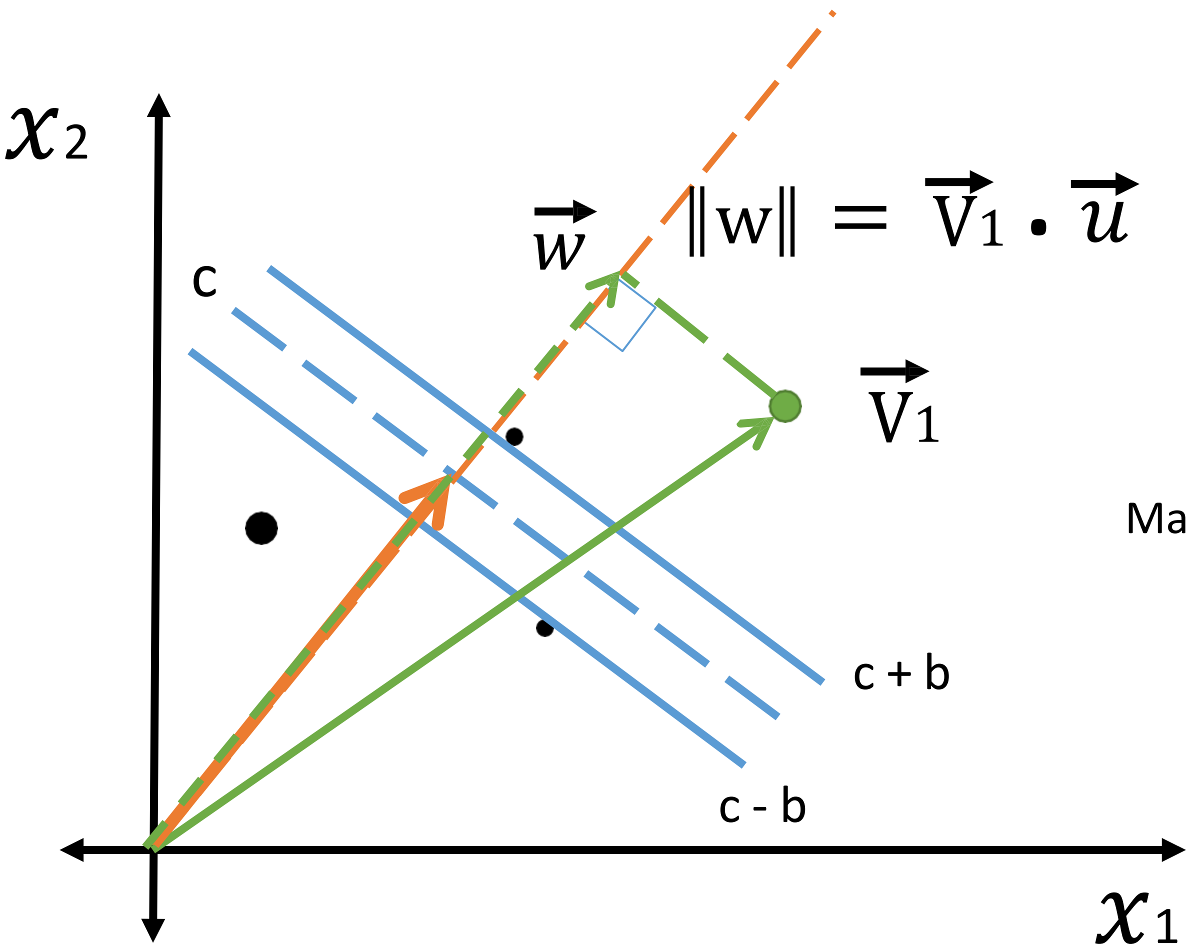


Select the right hyperplane



Determine Class

C = Magnitude of Normal

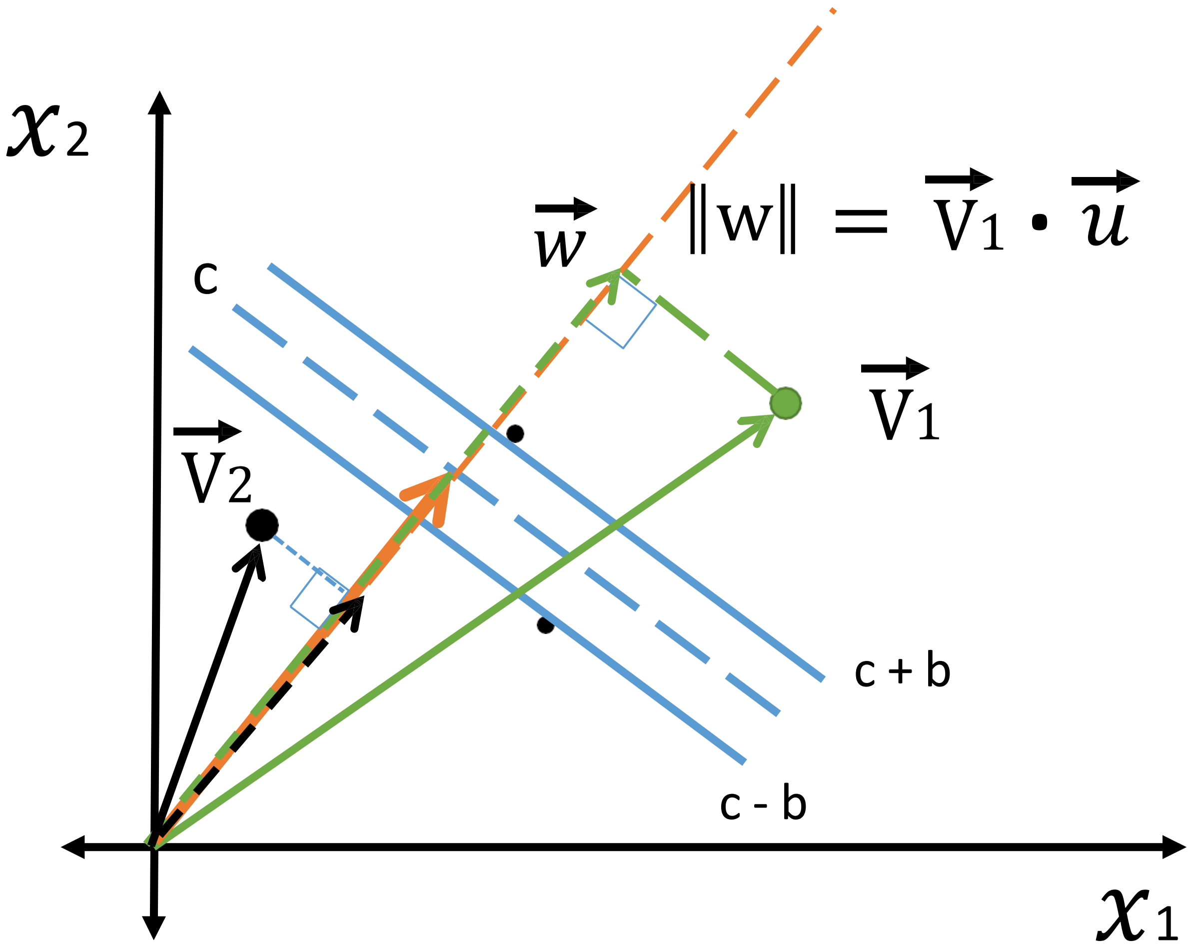


Margin = 2*b

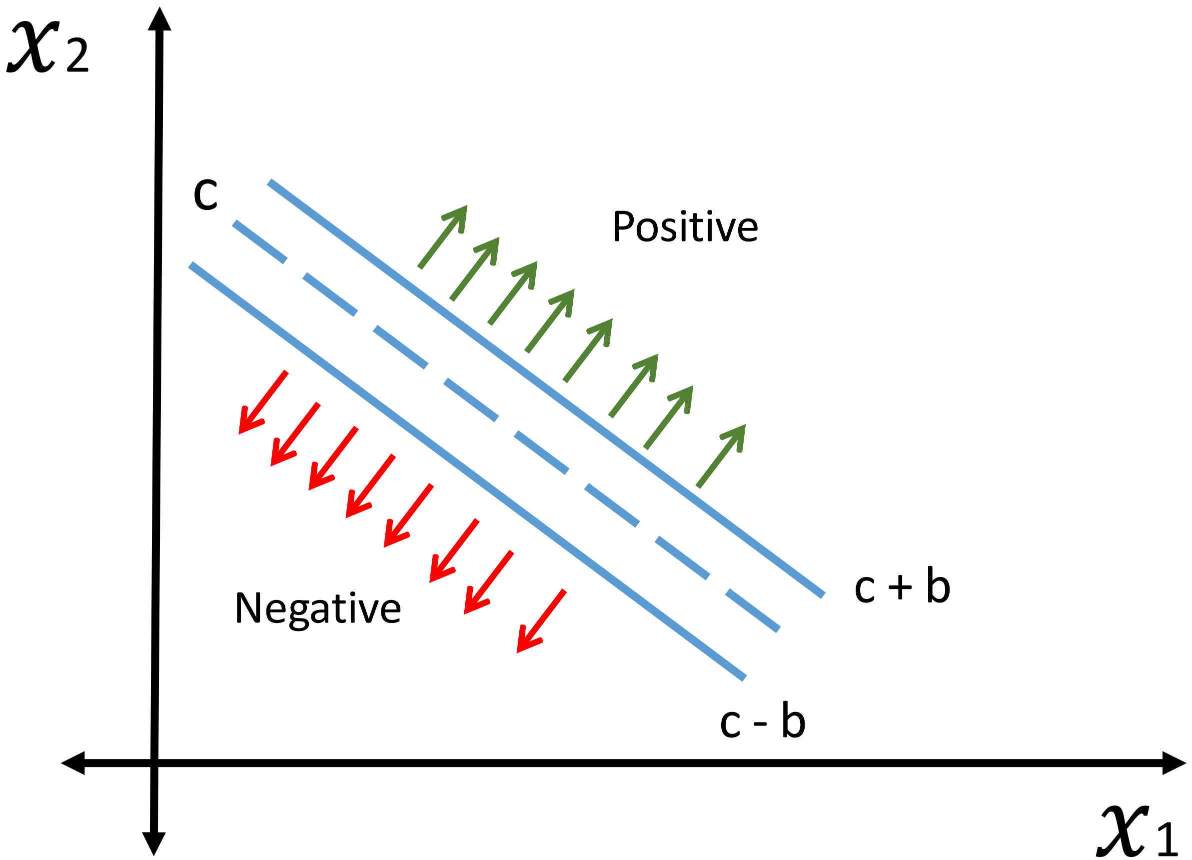
Determine Class

$\|w\| \geq c + b$ ➡ Positive

$\|w\| \leq c - b$ ➡ Negative

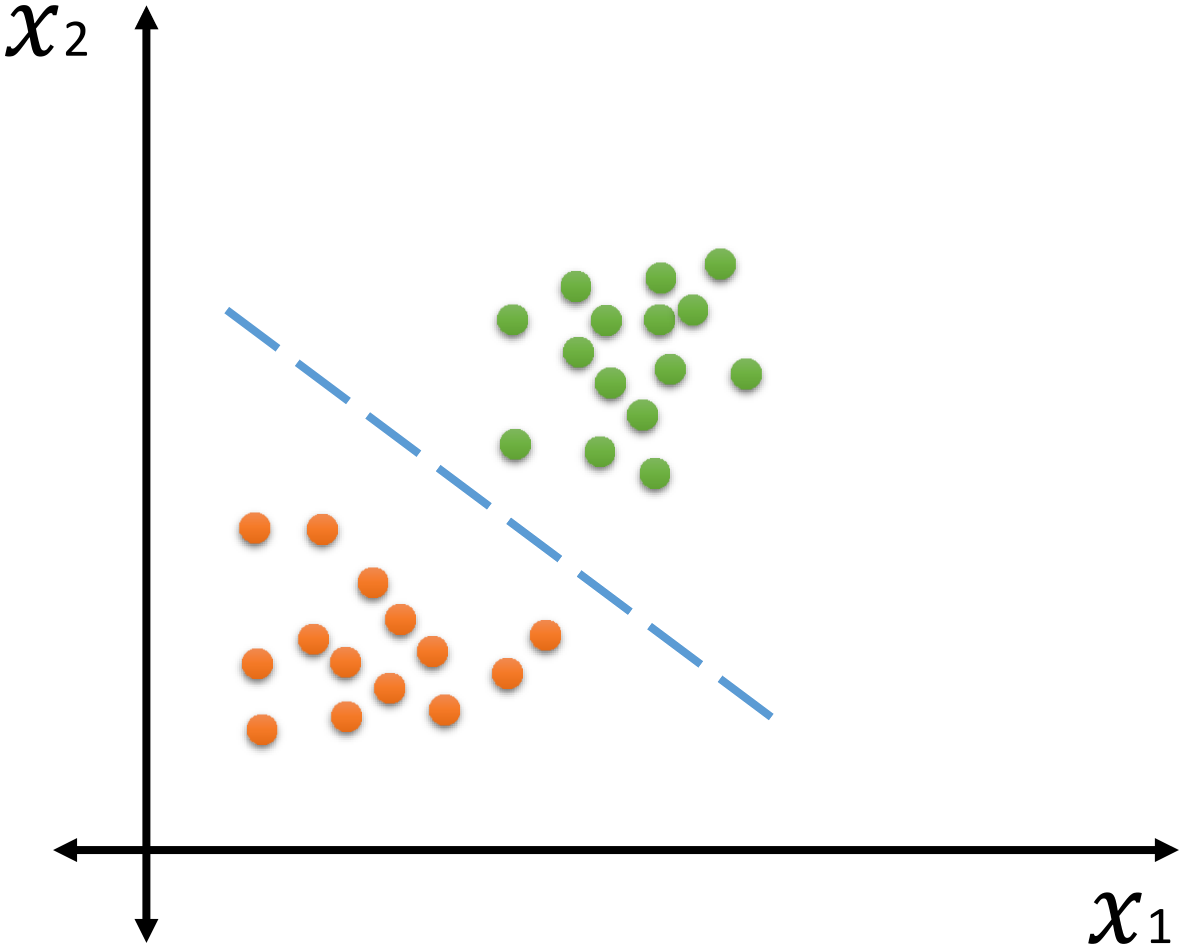


Determine Class

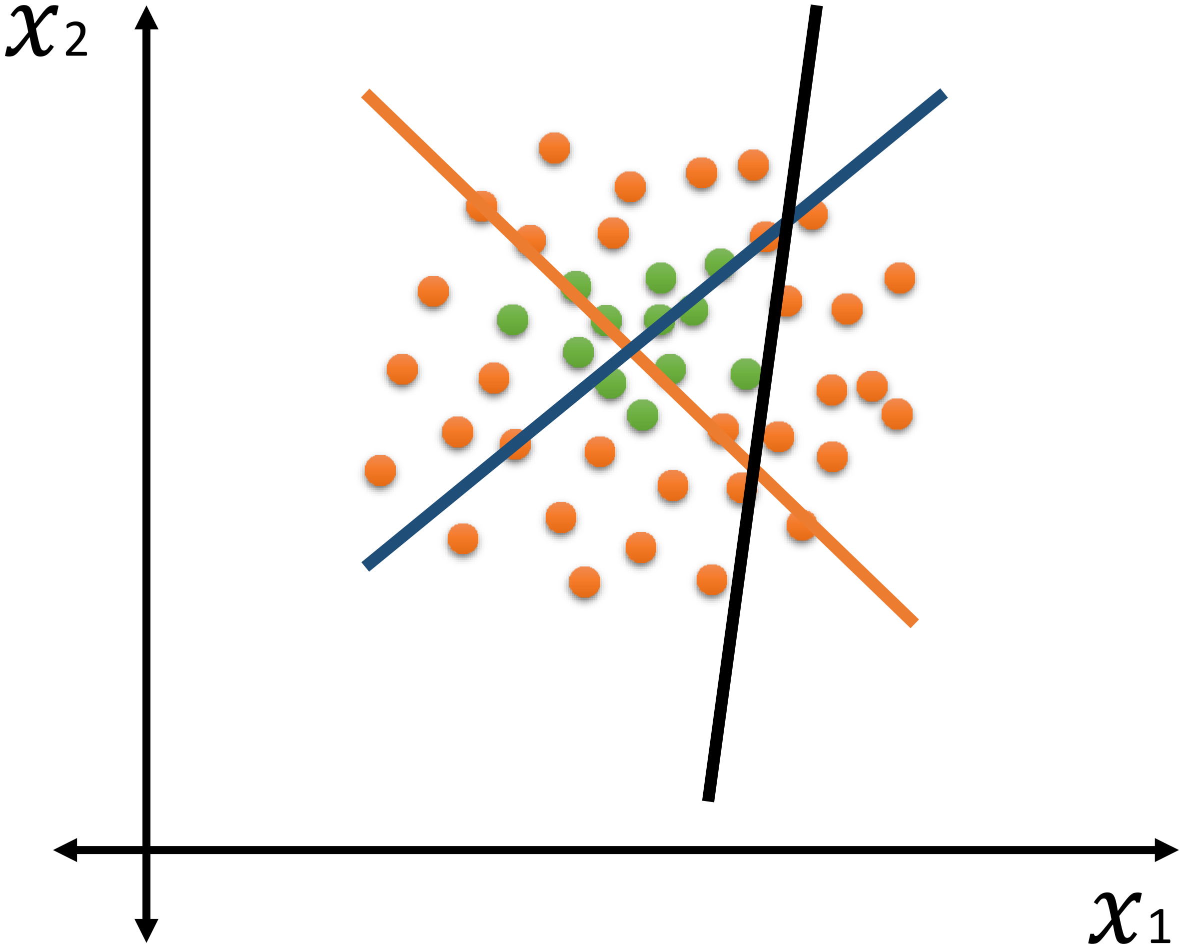


SVM Kernel

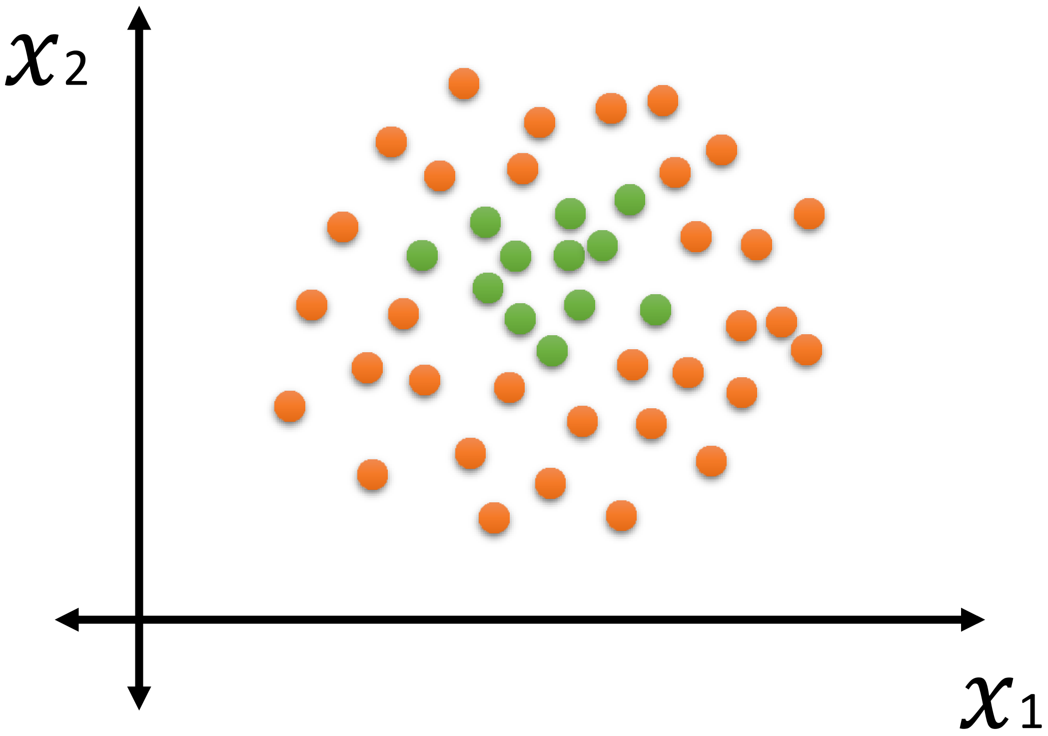
Linearly Separable



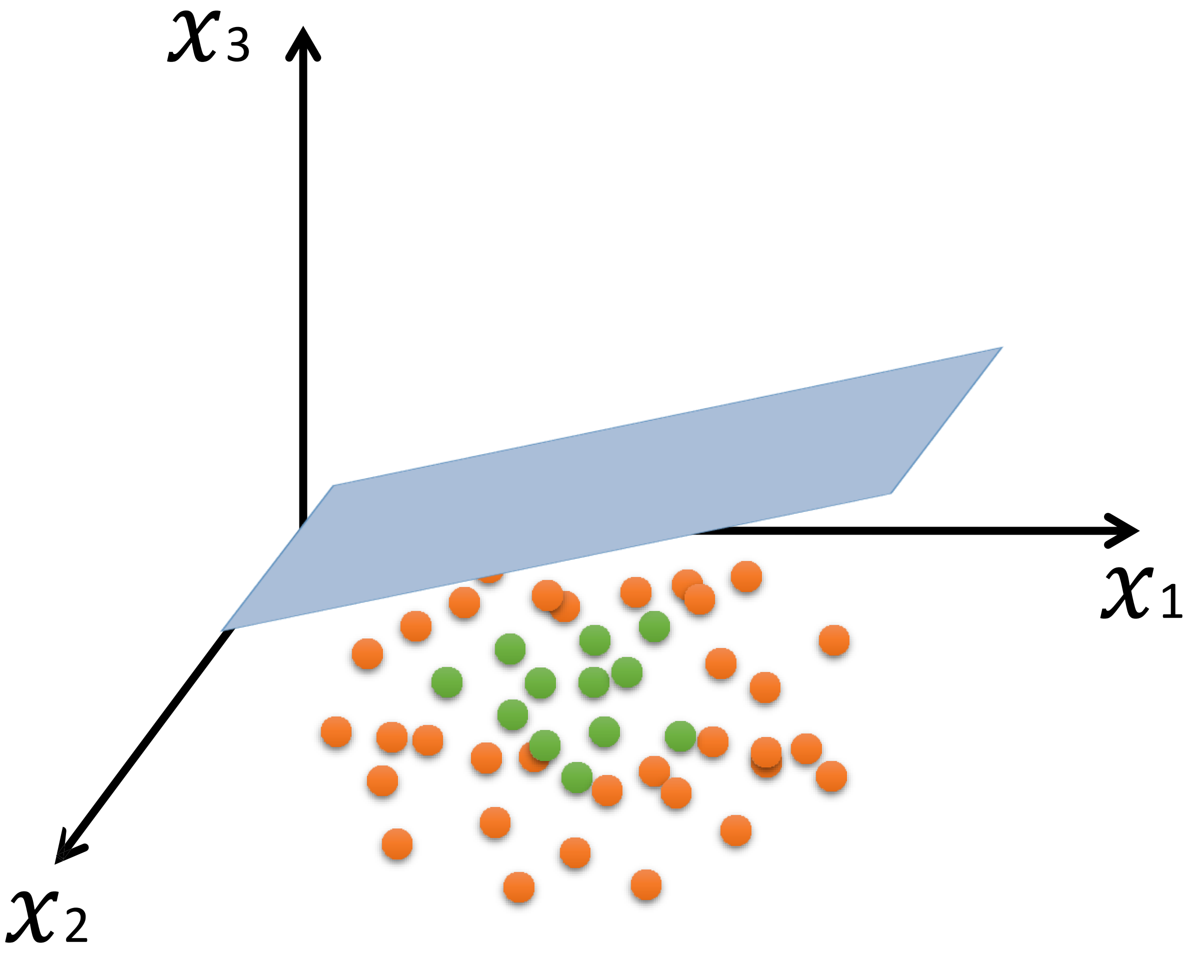
Linearly Separable



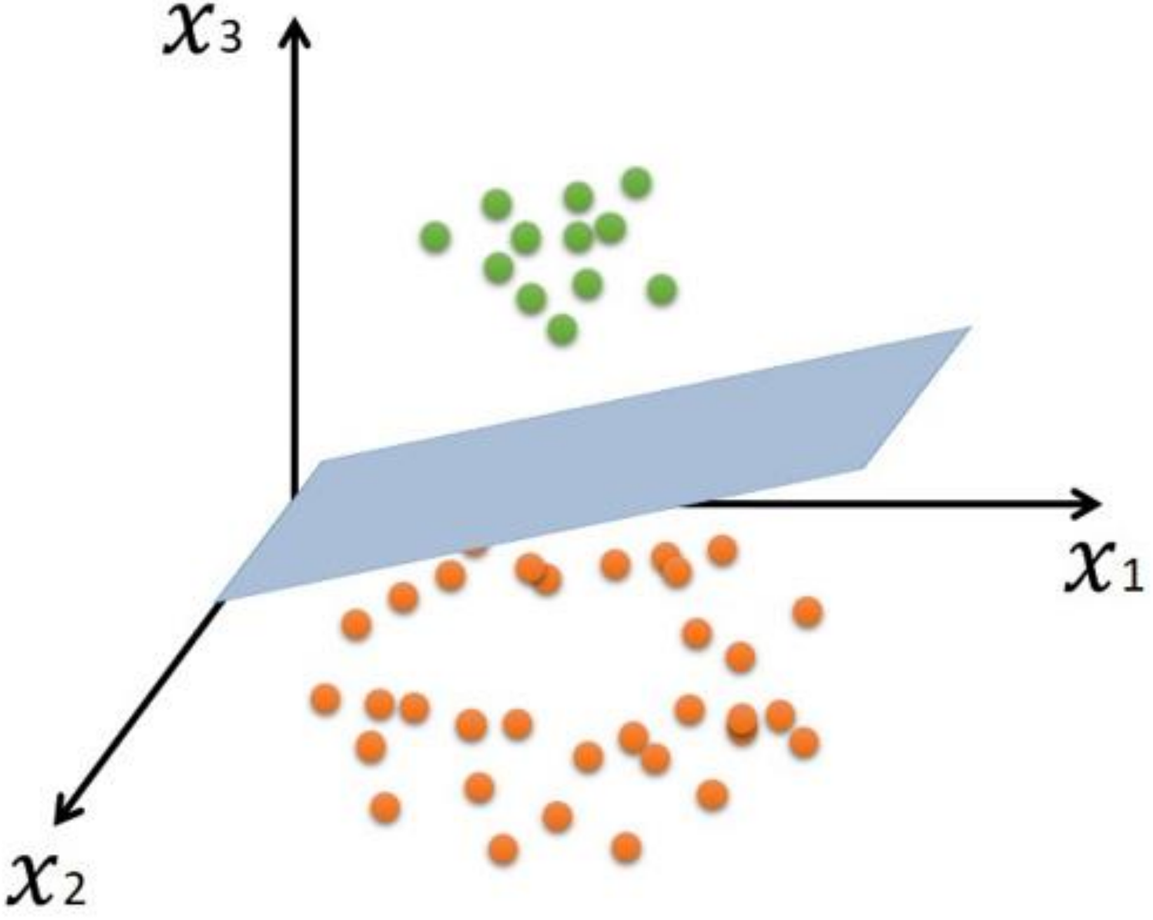
Change the dimensions



Change the dimensions

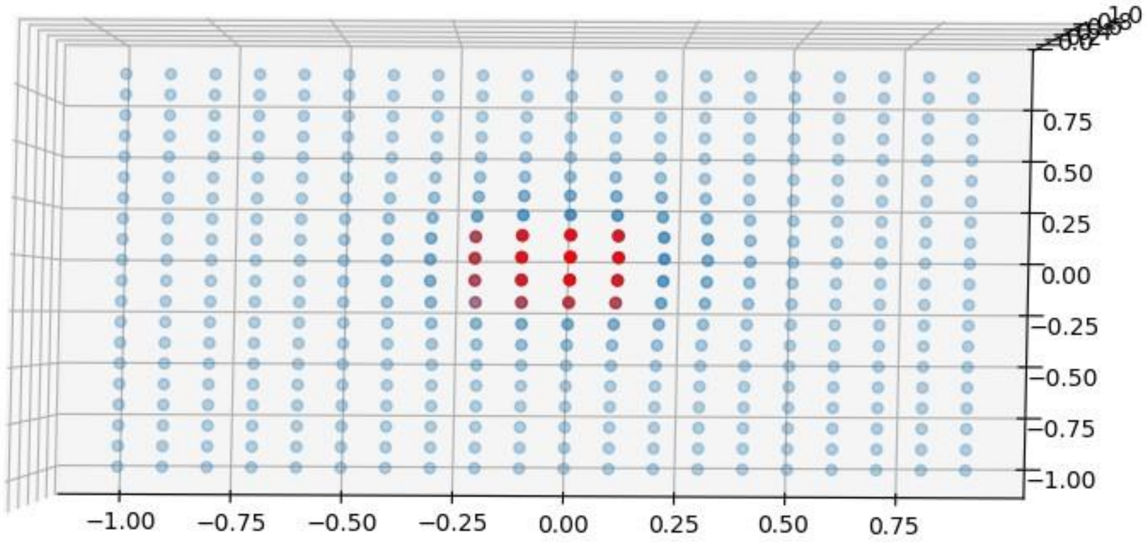


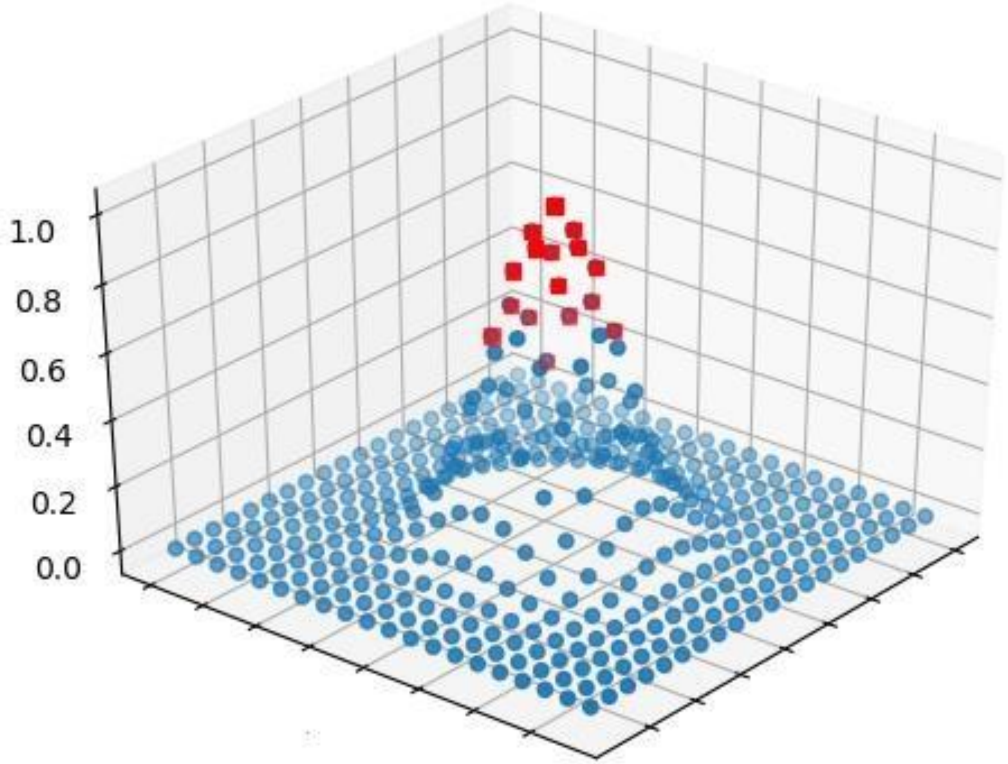
Change the dimensions



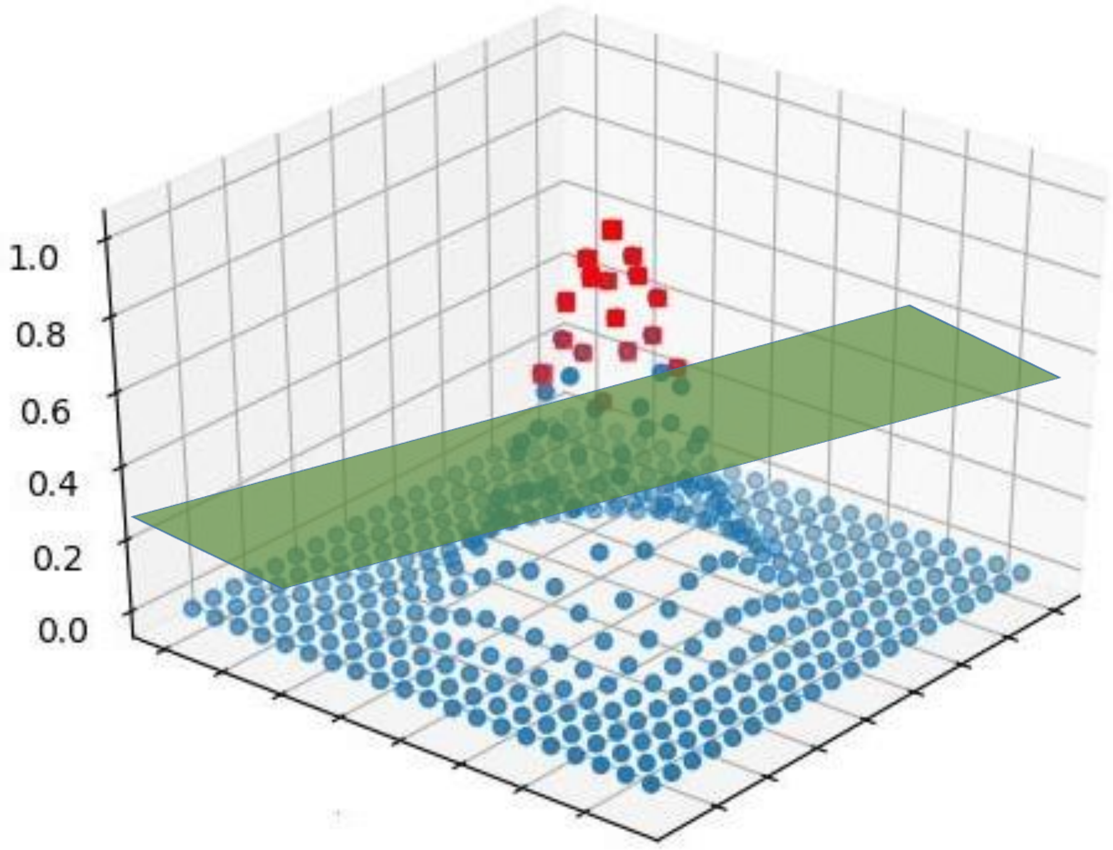
Let’s see it Graphically

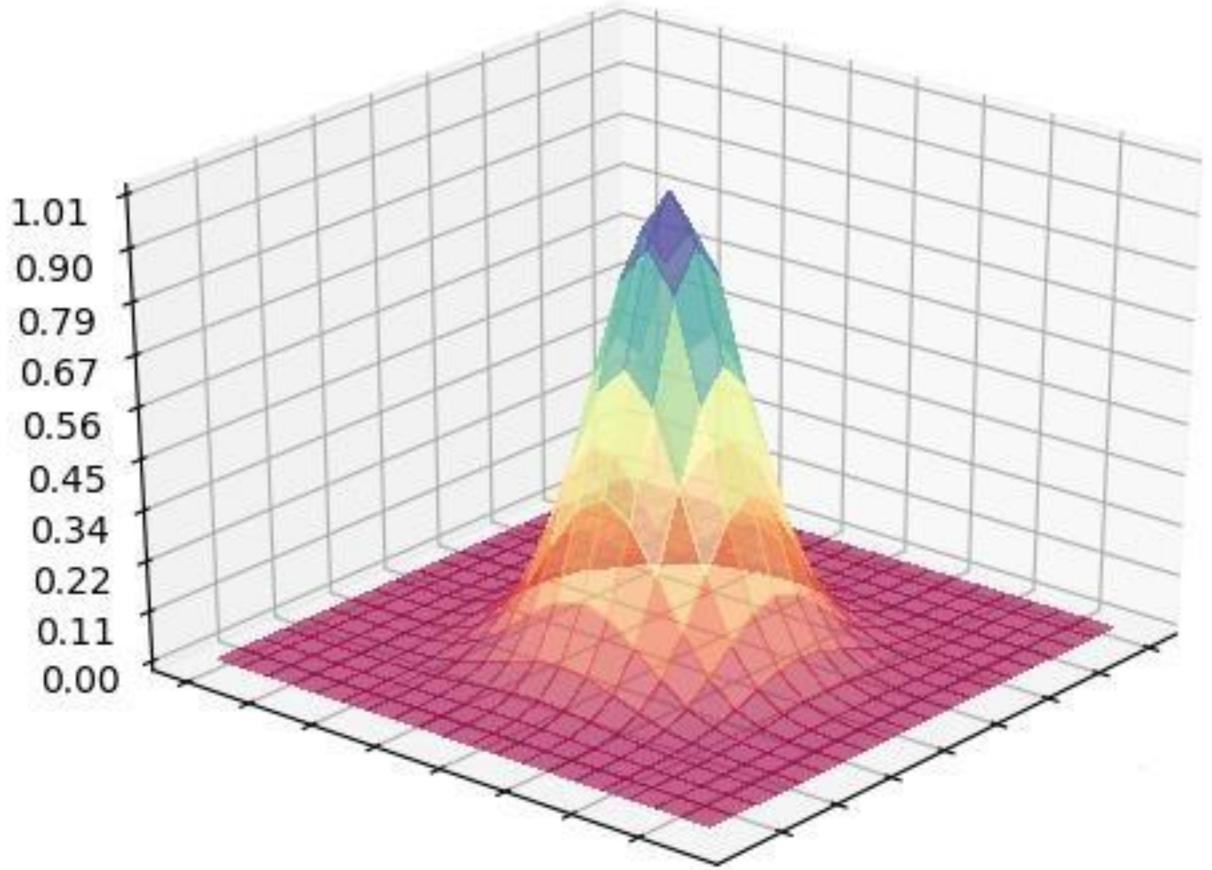
Change of perspective



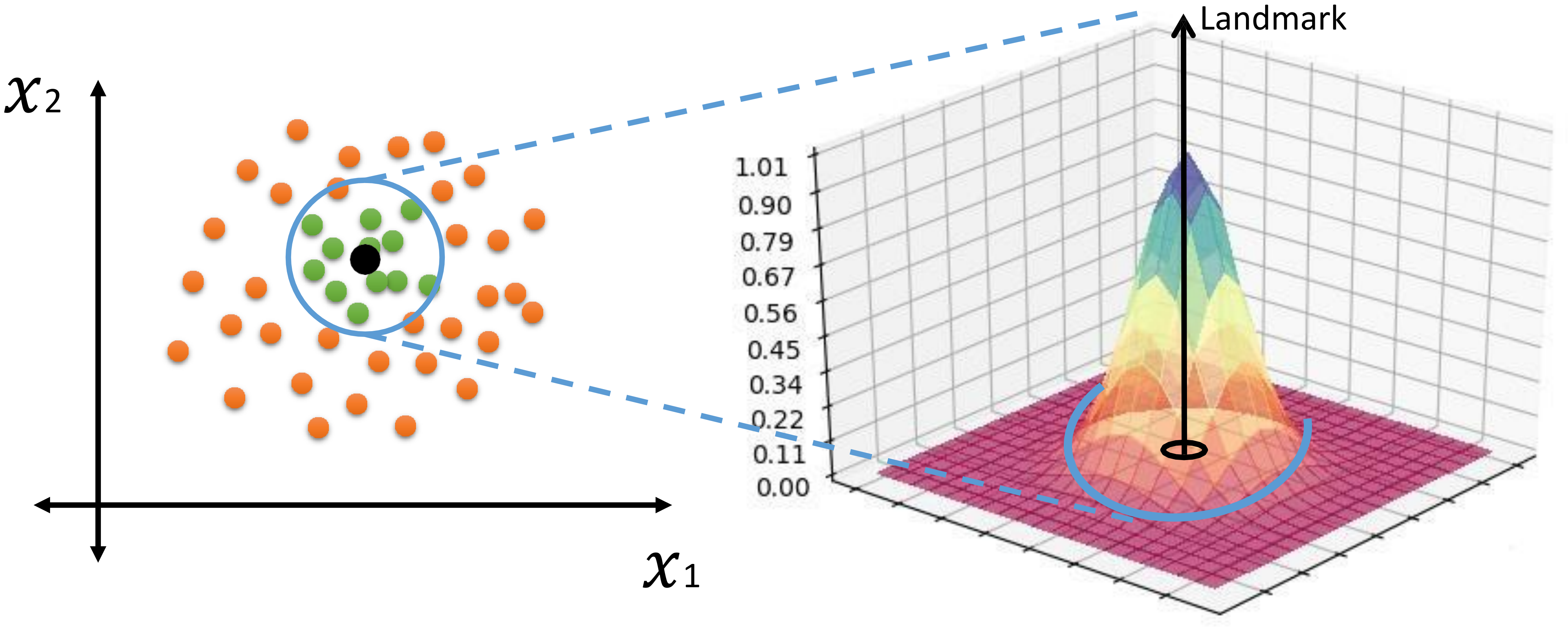


Gaussian Transformation



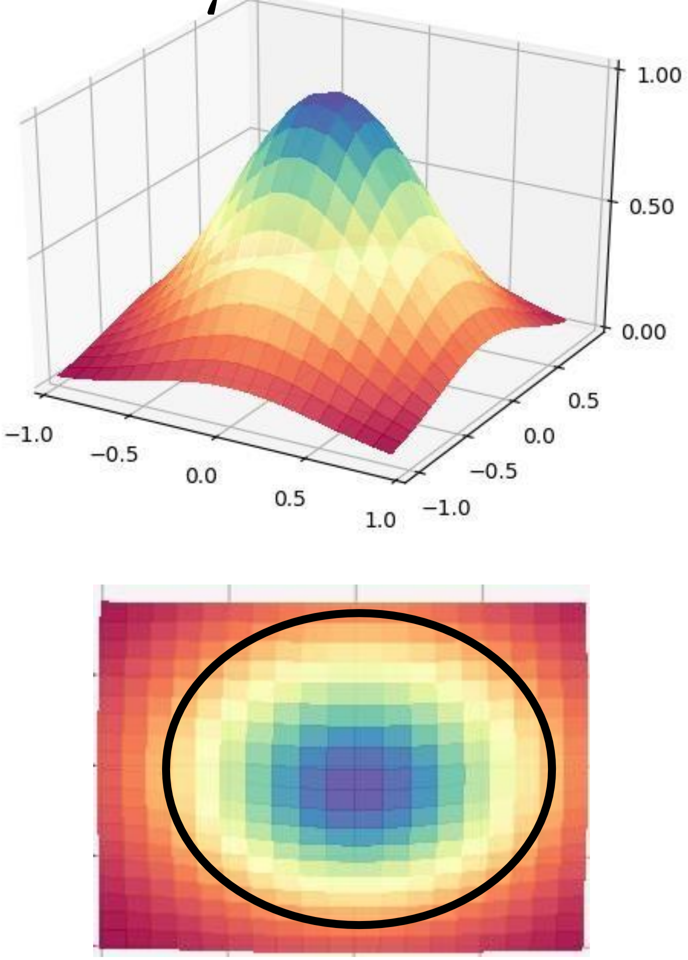


Radial Basis Function

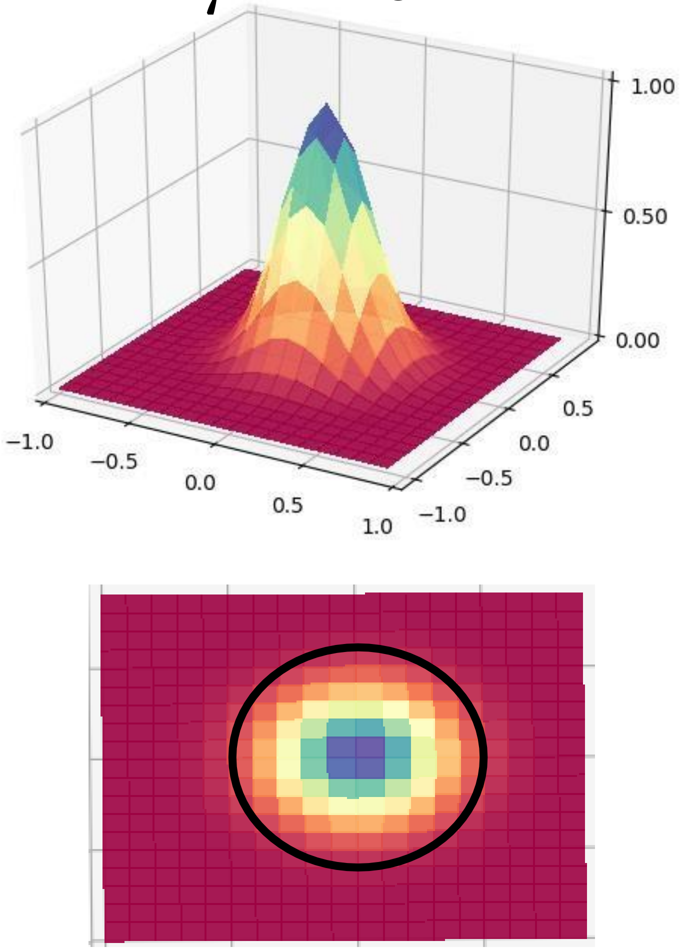


$$K(x,l) = e^{-\gamma \|\vec{x} - \vec{l}\|^2}$$

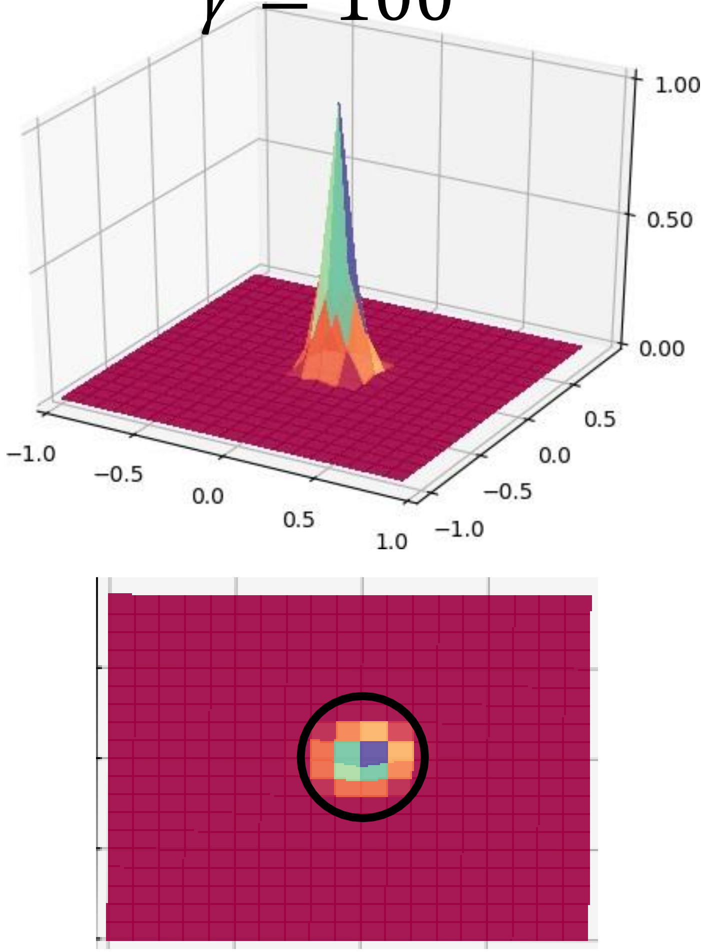
$\gamma = 2$



$\gamma = 10$

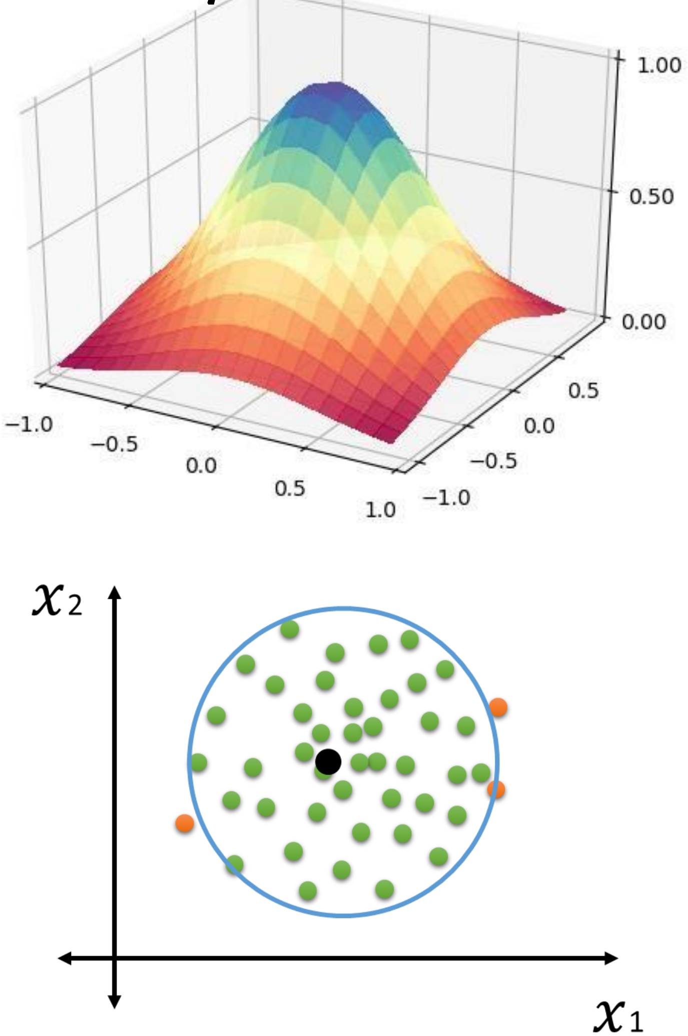


$\gamma = 100$

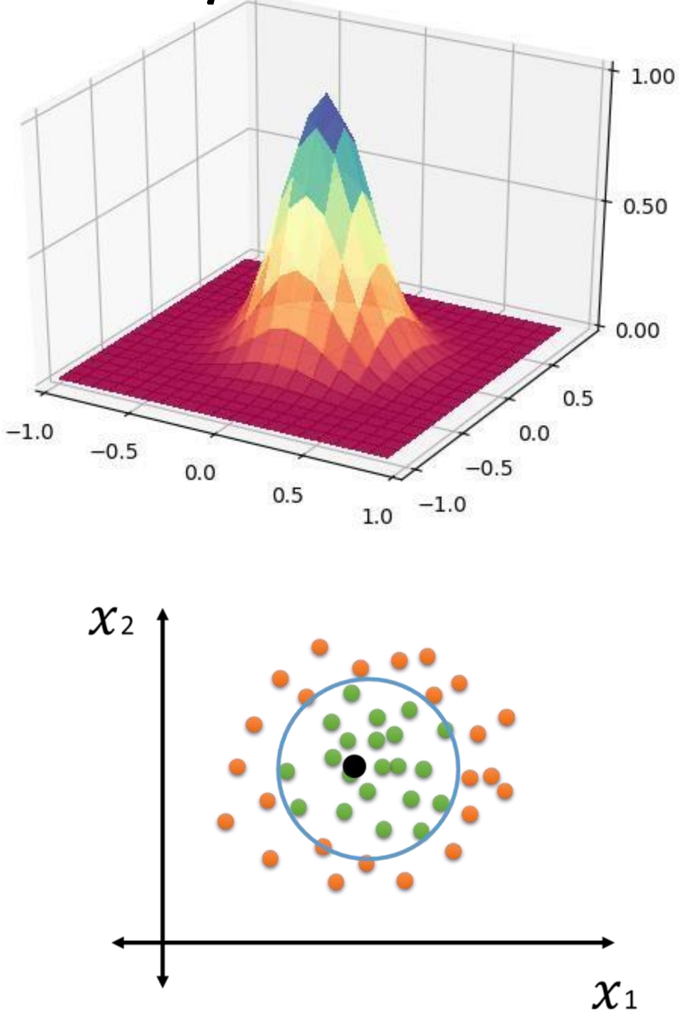


$$K(x,l) = e^{-\gamma \| \vec{x} - \vec{l} \|^2}$$

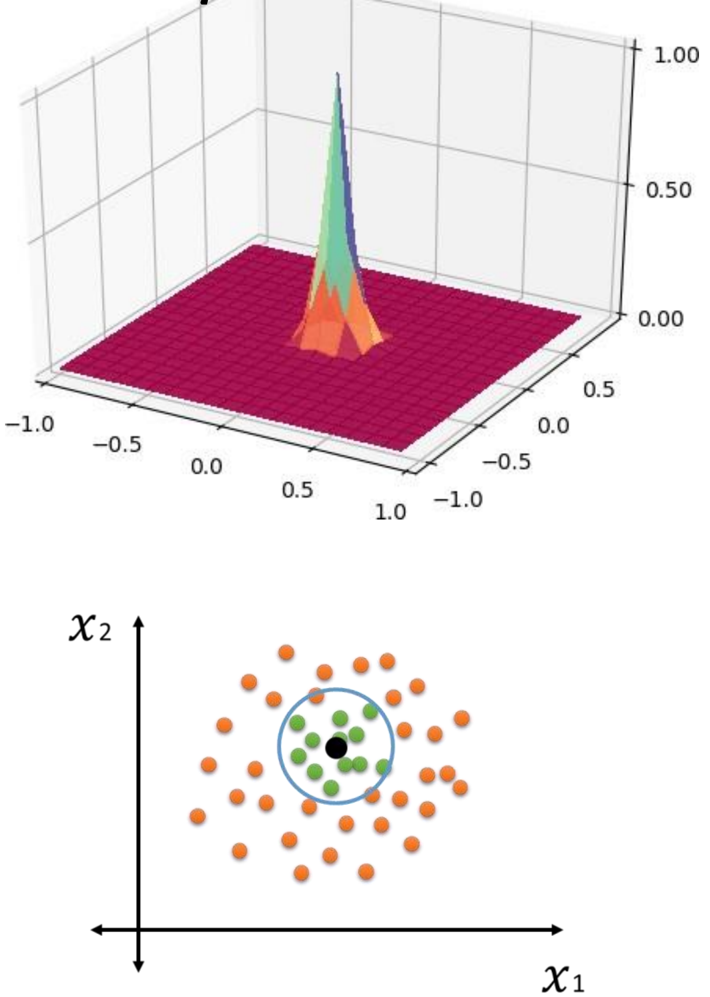
$\gamma = 2$



$\gamma = 10$

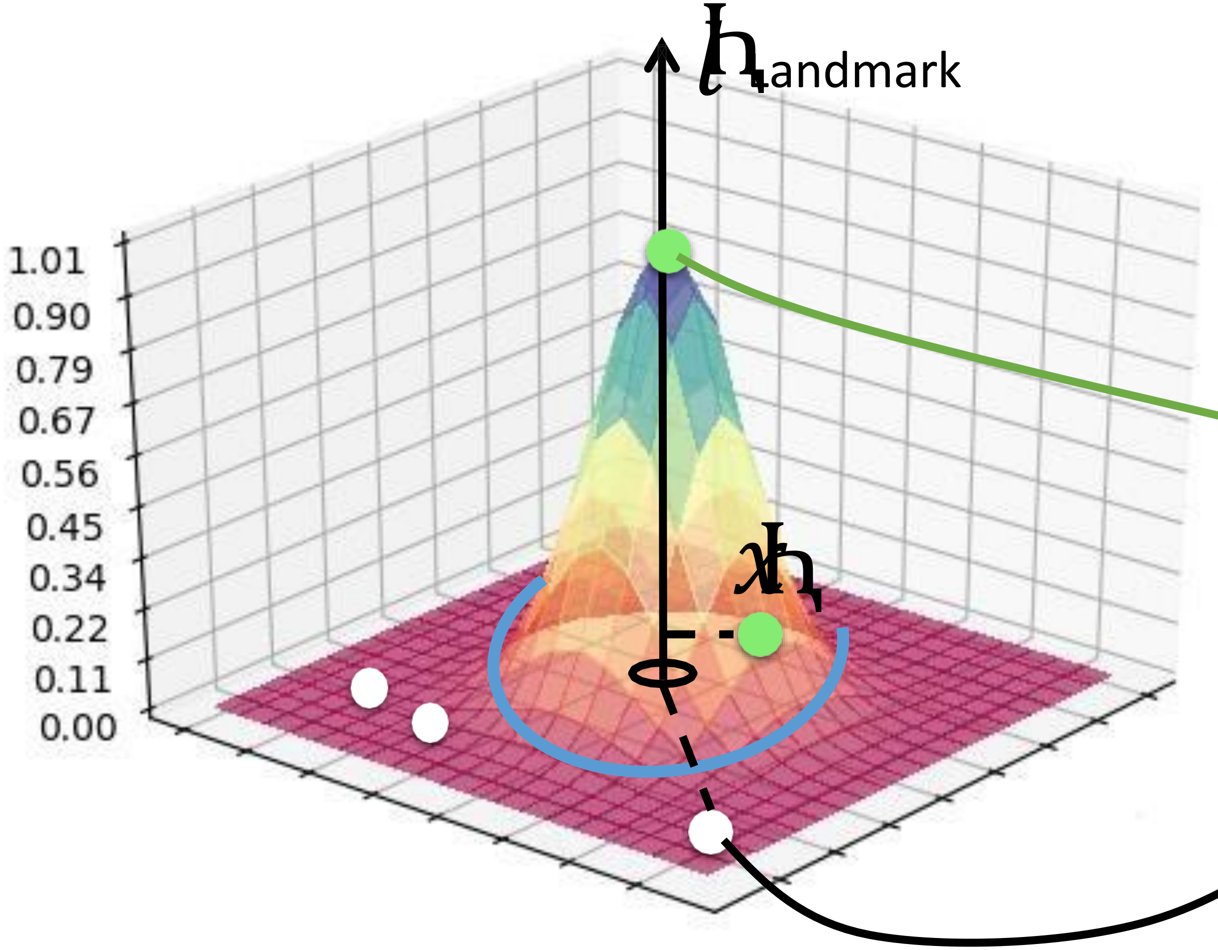


$\gamma = 100$



Radial Basis Function

$$K(x,l) = e^{-\frac{\|\vec{x} - \vec{l}\|^2}{2\sigma^2}}$$

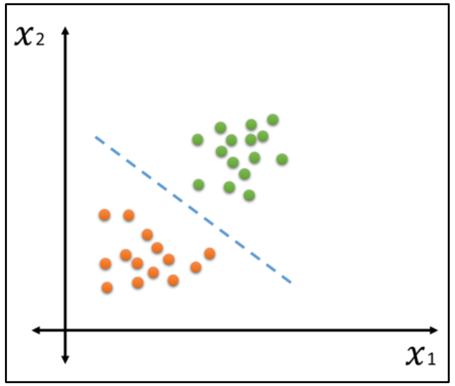


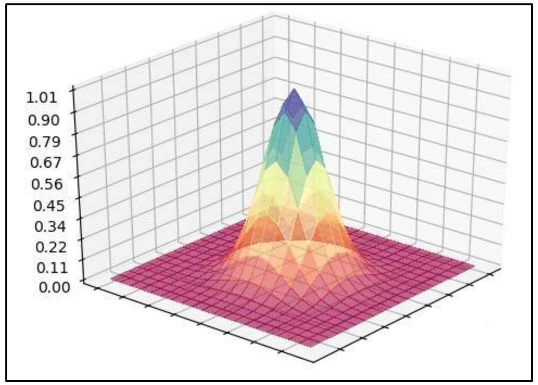
$K(x,l) \sim 1$

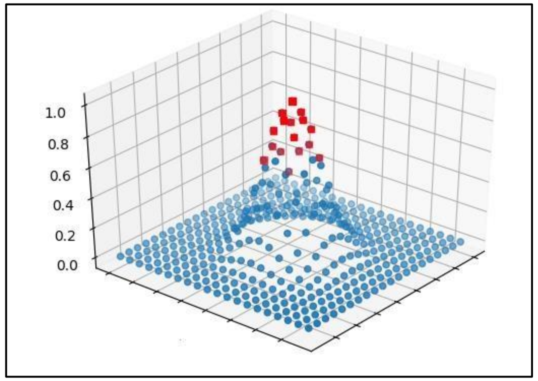
$K(x,l) \sim 0$

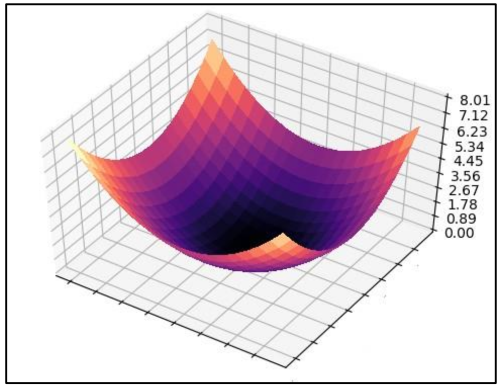
Types of Kernel Functions

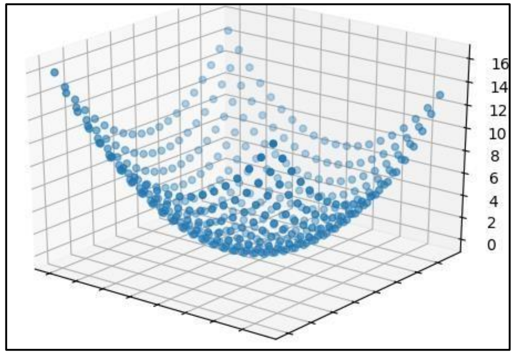
Linear	RBF	Polynomial	Sigmoid

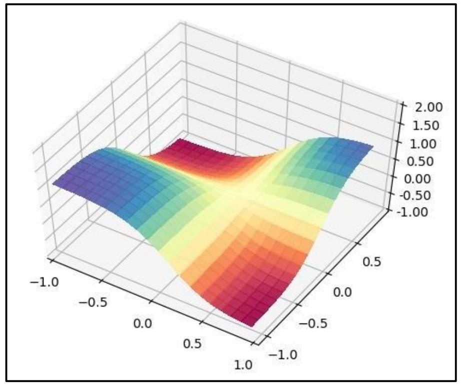


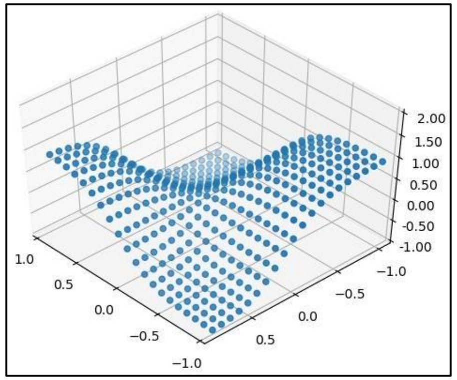










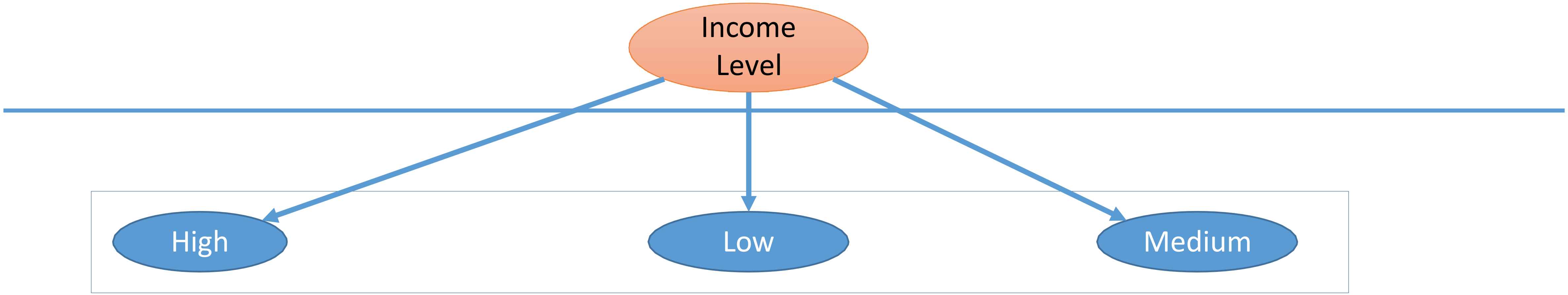


Decision Tree Classifier

What is Decision Tree?

- Supervised learning method
- Decision support tool that uses a tree-like graph or model of decisions and their possible consequences
- Various variations such as Boosted Decision Tree, Random Forest
- Can be used for categorical as well as continuous variables

Loan ID	Income Level	Credit Score	Employment	Approved?
L1	Medium	Low	Self-Employed	No
L2	High	Low	Self-Employed	Yes
L3	High	High	Salaried	Yes
L4	Medium	Low	Salaried	Yes
L5	Low	High	Salaried	No
L6	Low	Low	Self-Employed	No
L7	High	Low	Salaried	Yes
L8	Medium	Low	Self-Employed	No
L9	High	High	Self-Employed	Yes
L10	Medium	High	Self-Employed	Yes
L11	High	Low	Salaried	Yes
L12	Medium	High	Salaried	Yes
L13	Medium	High	Self-Employed	Yes
L14	Low	Low	Self-Employed	No
L15	Low	High	Self-Employed	No
L16	**Medium**	**High**	**Salaried**	**???**



LID	IL	CS	ET	Status
L2	High	Low	SE	Yes
L3	High	High	Salaried	Yes
L7	High	Low	Salaried	Yes
L9	High	High	SE	Yes
L11	High	Low	Salaried	Yes

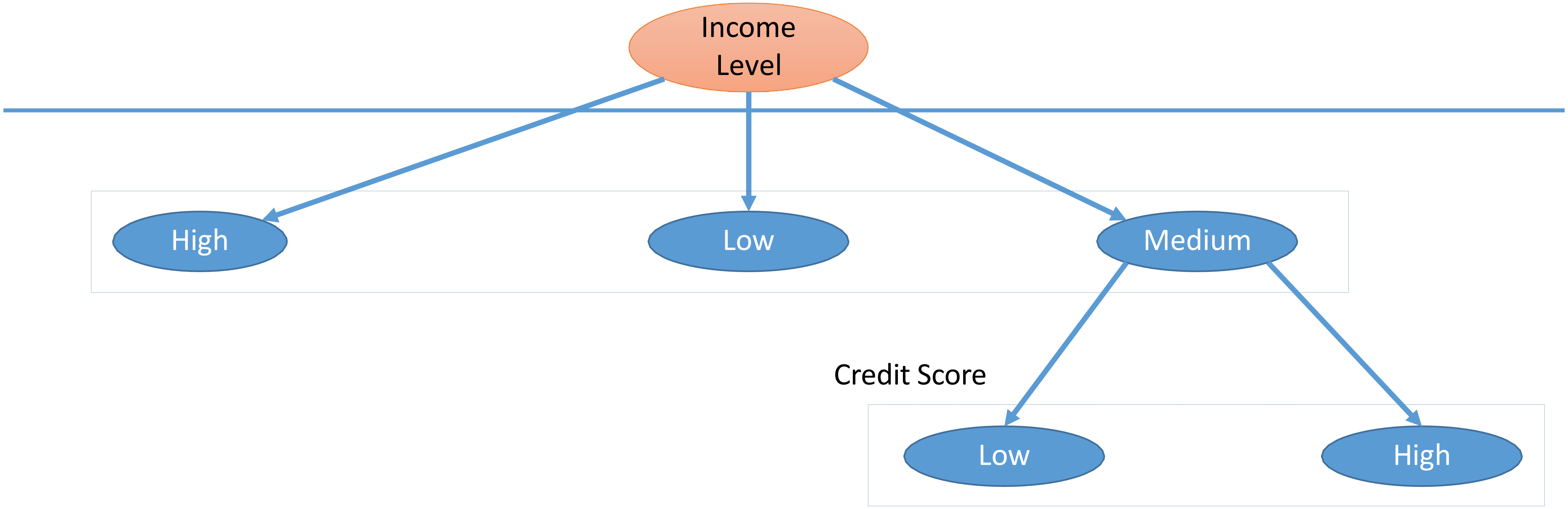
Pure Subset

LID	IL	CS	ET	Status
L5	Low	High	Salaried	No
L6	Low	Low	SE	No
L14	Low	Low	SE	No
L15	Low	High	SE	No

Pure Subset

LID	IL	CS	ET	Status
L1	Medium	Low	SE	No
L4	Medium	Low	Salaried	Yes
L8	Medium	Low	SE	No
L10	Medium	High	SE	Yes
L12	Medium	High	Salaried	Yes
L13	Medium	High	SE	Yes

Split Further

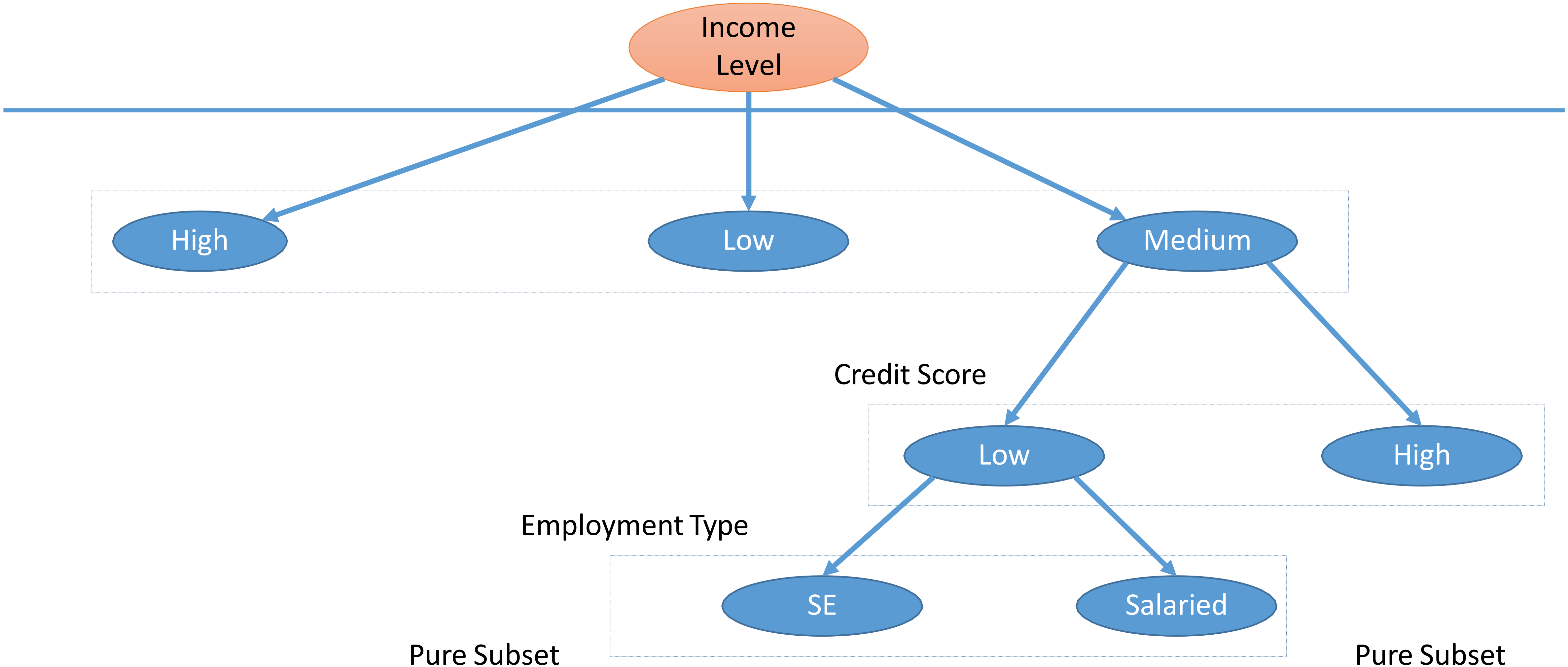


LID	IL	CS	ET	Status
L1	Medium	Low	SE	No
L4	Medium	Low	Salaried	Yes
L8	Medium	Low	SE	No

Split Further

LID	IL	CS	ET	Status
L10	Medium	High	SE	Yes
L12	Medium	High	Salaried	Yes
L13	Medium	High	SE	Yes

Pure Subset



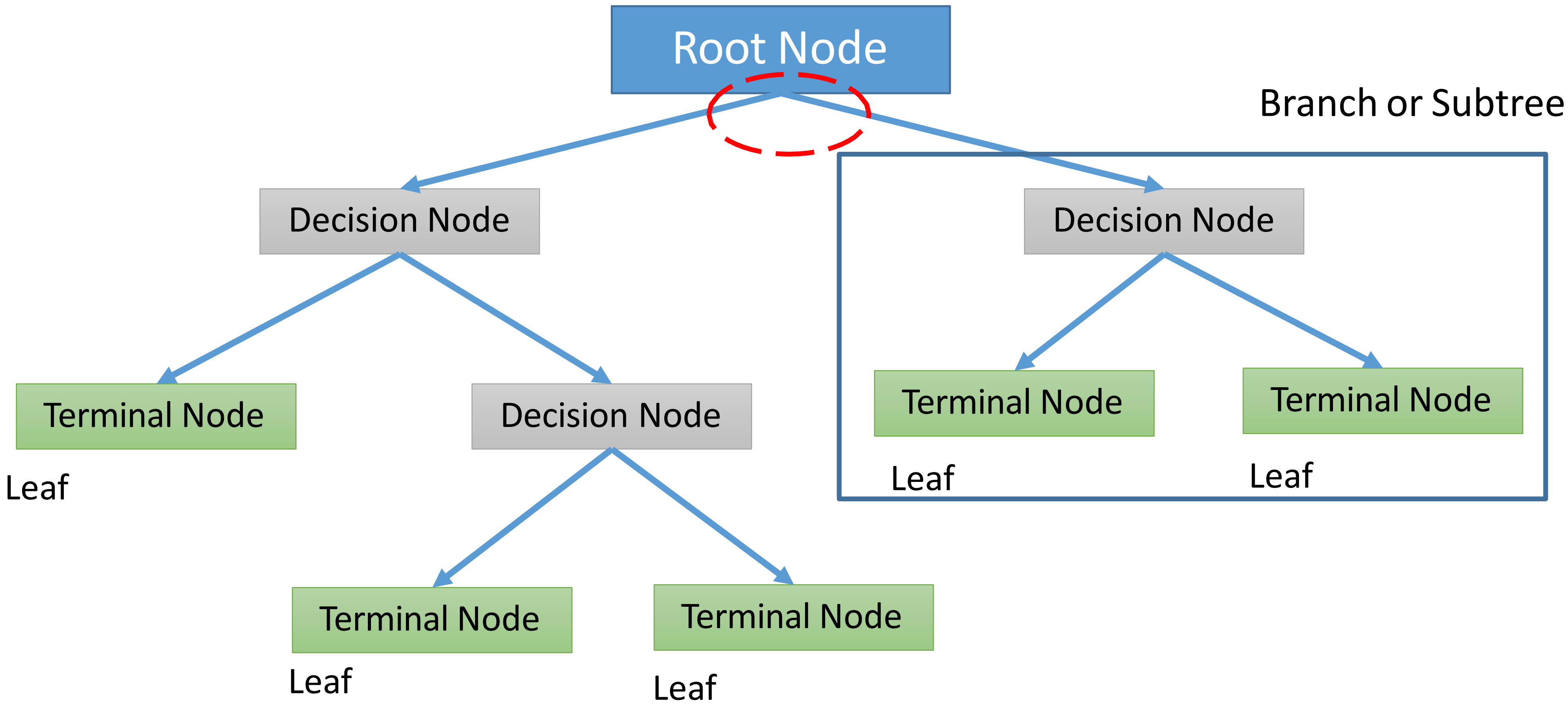
Pure Subset

LID	IL	CS	ET	Status
L1	Medium	Low	SE	No
L8	Medium	Low	SE	No

Pure Subset

LID	IL	CS	ET	Status
L4	Medium	Low	Salaried	Yes

Decision Tree Terms



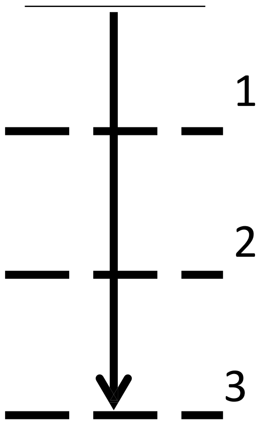
Parameter of Decision Tree Classifier

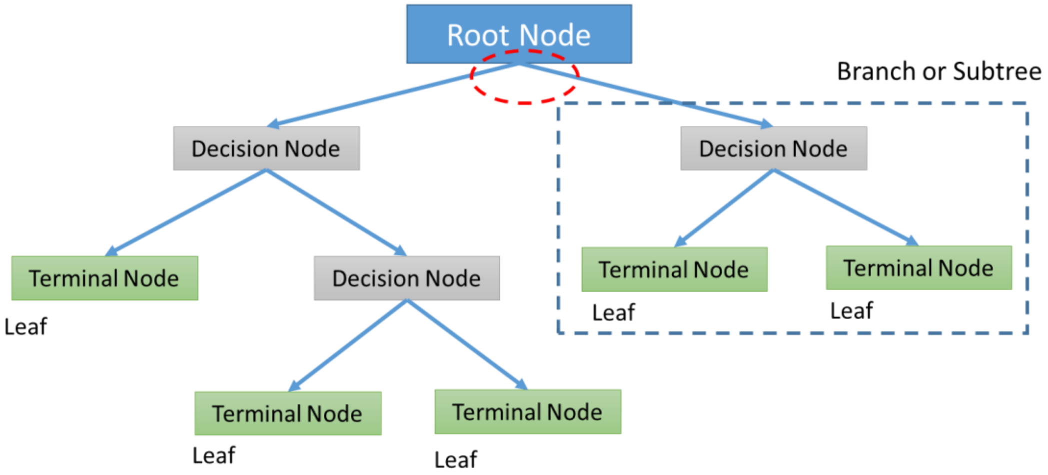
sklearn.tree.DecisionTreeClassifier – Parameters

- max_depth
- min_samples_split
- min_samples_leaf
- max_leaf_nodes
- splitter
- max_features
- presort
- criterion
- min_impurity_decrease

sklearn.tree.DecisionTreeClassifier – Parameters

- **max_depth – max depth of the tree**

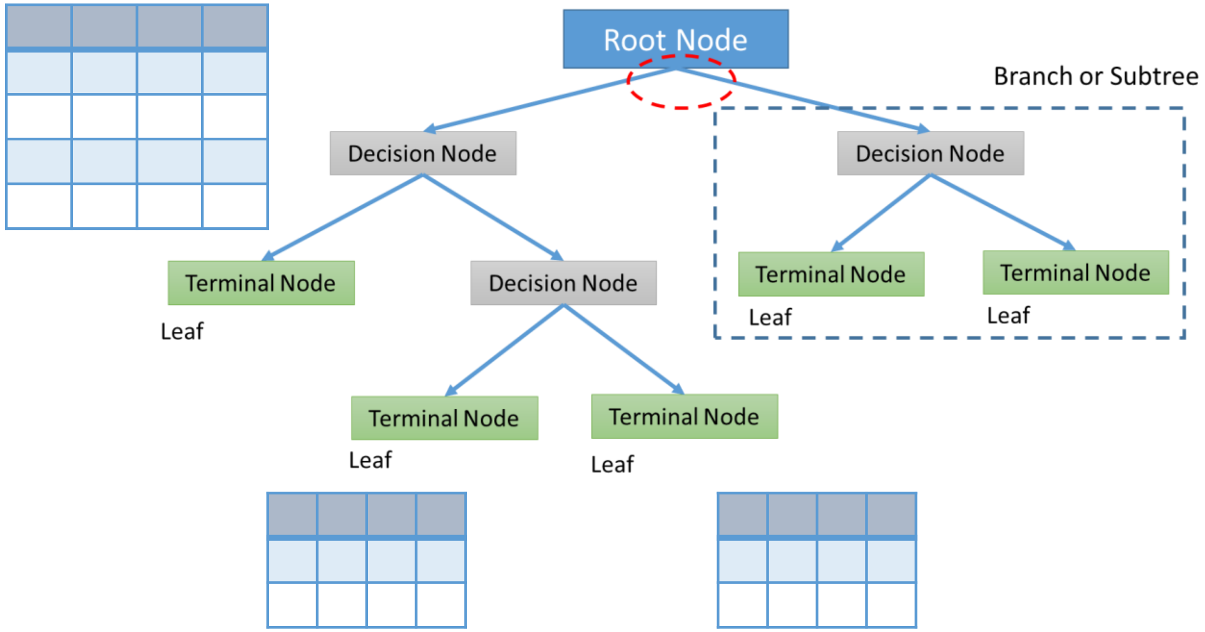




- min_samples_split
- min_samples_leaf
- max_leaf_nodes

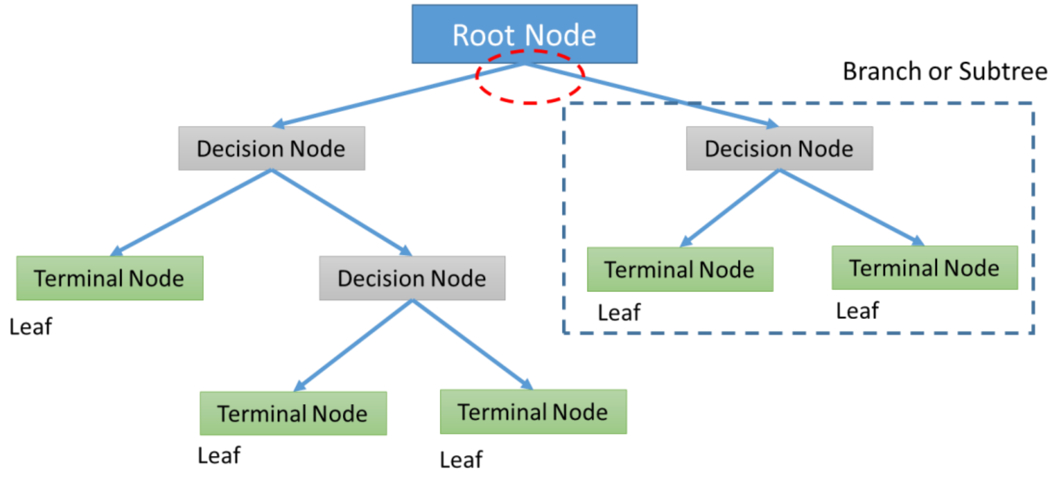
sklearn.tree.DecisionTreeClassifier – Parameters

- max_depth
- **min_samples_split – Min Samples required for the split**
- min_samples_leaf
- max_leaf_nodes



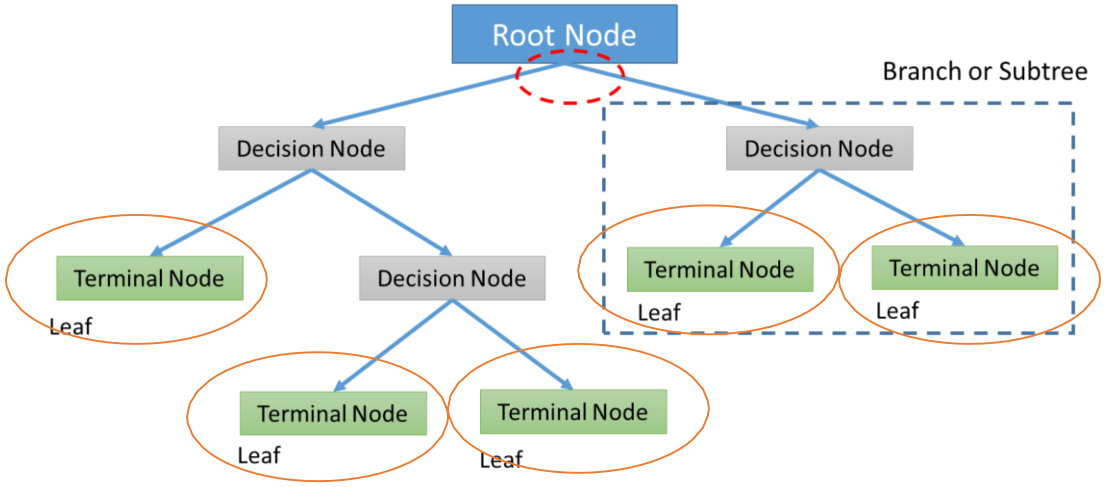
sklearn.tree.DecisionTreeClassifier – Parameters

- max_depth
- min_samples_split
- **min_samples_leaf - Min samples required at the leaf**
- max_leaf_nodes



sklearn.tree.DecisionTreeClassifier – Parameters

- max_depth
- min_samples_split
- min_samples_leaf
- **max_leaf_nodes - max number of leaf nodes**



sklearn.tree.DecisionTreeClassifier – Parameters

- max_depth
- min_samples_split
- min_samples_leaf
- max_leaf_nodes

- splitter
- max_features
- presort

- criterion
- min_impurity_decrease

Gender

Male: Yes = 4 No = 4

Gender	Approved?
Male	No
Male	Yes
Male	Yes
Male	No
Male	Yes
Male	No
Male	Yes
Male	No

50% 50%

Female: Yes = 4 No = 4

Gender	Approved?
Female	Yes
Female	No
Female	No
Female	Yes
Female	Yes
Female	Yes
Female	No
Female	No

50% 50%

Highly Impure

Credit History

Low: Yes = 1 No = 7

Credit Score	Approved?
Low	No
Low	Yes
Low	No
Low	No
Low	No
Low	No
Low	No
Low	No

12% 88%

High: Yes = 7 No = 1

Credit Score	Approved?
High	Yes
High	Yes
High	No
High	Yes
High	Yes
High	Yes
High	Yes
High	Yes

88% 12%

Less Impure

Best Split → Credit History

Information Gain – Ability to answer the question

Gender

Male

Yes = 4 No = 4

Gender	Approved?
Male	No
Male	Yes
Male	Ye
Male	N
Male	Ye
Male	No
Male	Yes
Male	No

50%	50%

Female

Yes = 4 No = 4

Gender	Approved?
Female	Yes
Female	No
[illegible]	[illegible]
[illegible]	[illegible]
Female	Yes
Female	Yes
Female	No
Female	No

50%	50%

Credit History

Low

Yes = 1 No = 7

Credit Score	Approved?
Low	No
Low	Yes
[illegible]	[illegible]
[illegible]	[illegible]
Low	No
Low	No
Low	No
Low	No

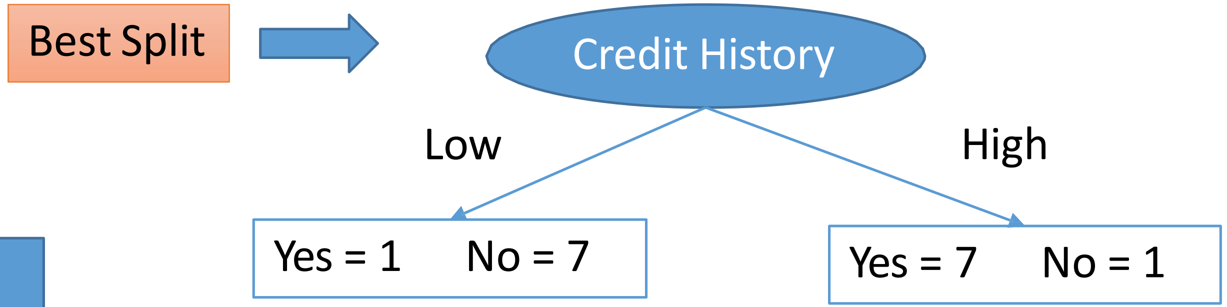
12%	88%

High

Yes = 7 No = 1

Credit Score	Approved?
High	Yes
High	Yes
[illegible]	[illegible]
[illegible]	[illegible]
High	Yes
High	Yes
High	Yes
High	Yes

88%	12%



Parameters of Decision Tree relation to splitting

- **splitter – Split strategy for Best feature or Random feature**
- max_features
- presort

Credit Score	Approved?
Low	No
Low	Yes
Low	No
Low	No
Low	No
Low	No
Low	No
Low	No

12%	88%

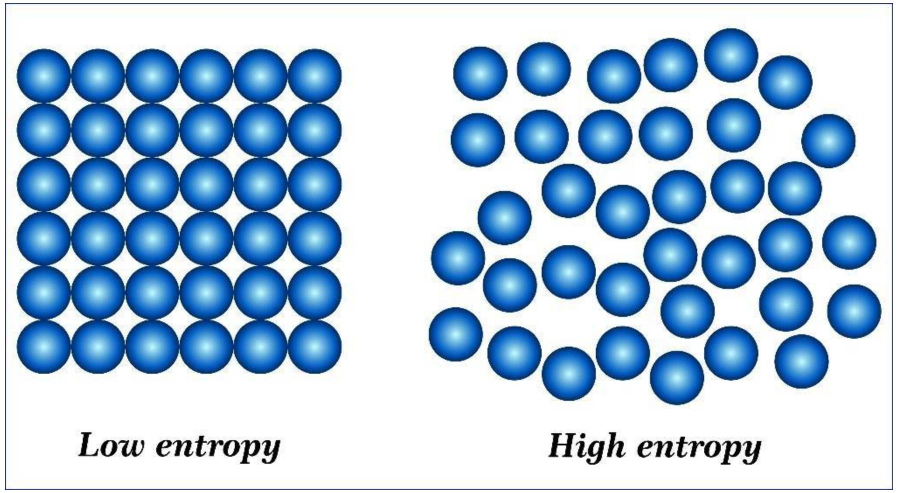
Credit Score	Approved?
High	Yes
High	Yes
High	No
High	Yes
High	Yes
High	Yes
High	Yes
High	Yes

88%	12%

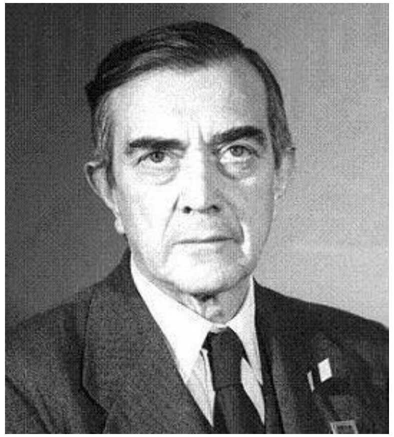
How to decide which Feature has the Best Split?

What should be the **criterion**?

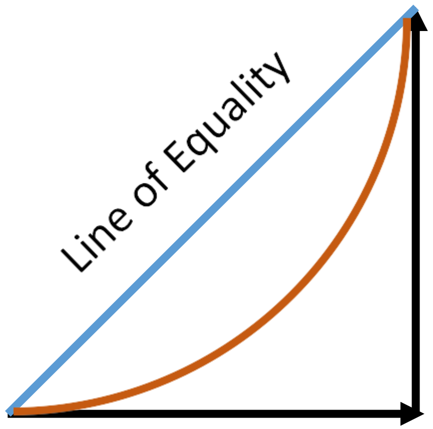
Entropy



Gini

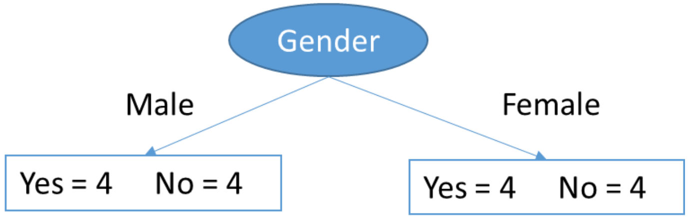


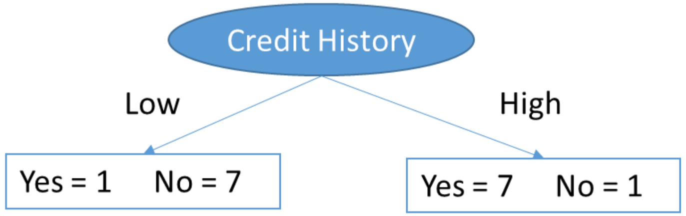
Corrado Gini



Entropy - Measure of Impurity

$$Entropy = -1 * \sum_{i=1}^{n} p_i \log_2 p_i$$





S = $-1 * (0.5*\log_2 0.5 + 0.5*\log_2 0.5)$

$= -1 * (0.5 * (-1) + 0.5 * (-1))$

$= 0.5 + 0.5$

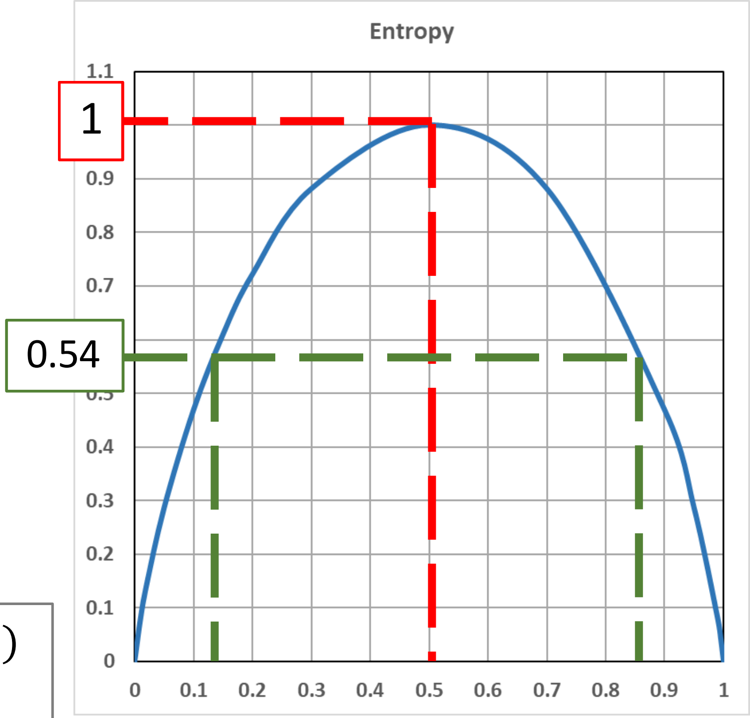
$= 1$

S = $-1 * (0.125*\log_2 0.125 + 0.875*\log_2 0.875)$

$= -1 * (0.125 * (-3) + 0.875 * (-0.1926))$

$= -1 * (-0.375 + (-0.1685))$

$= 0.5435$



Gini

$$Gini = 1 - \sum_{i=1} p^2$$

Gender

Male: Yes = 4, No = 4

Female: Yes = 4, No = 4

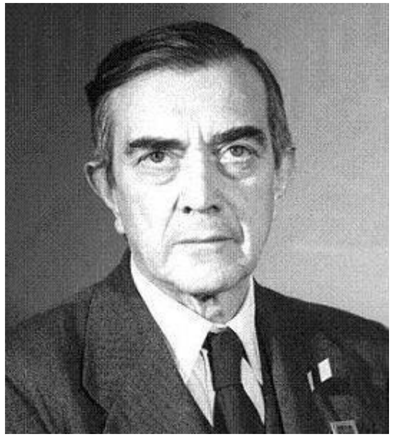
Credit History

Low: Yes = 1, No = 7

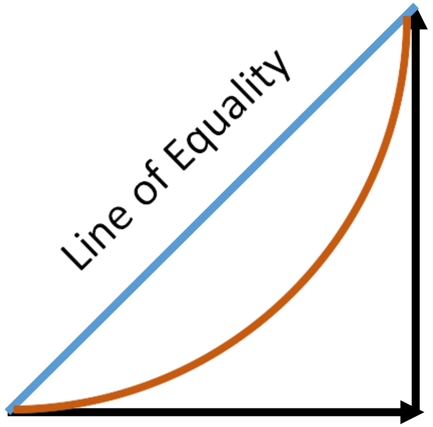
High: Yes = 7, No = 1

$$\begin{aligned} \mathrm{Gini} &= 1 - (0.5^2 + 0.5^2) \\ &= 1 - (0.25 + 0.25) \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} \mathrm{Gini} &= 1 - (0.125^2 + 0.875^2) \\ &= 1 - (0.25 + 0.25) \\ &= 0.22 \end{aligned}$$



Corrado Gini



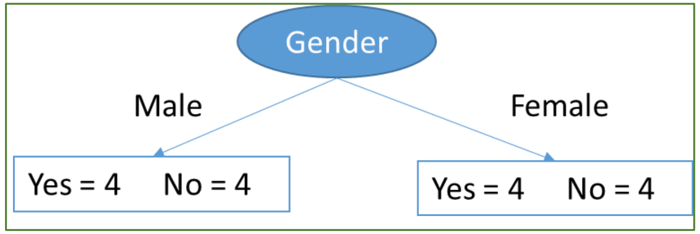
Information Gain

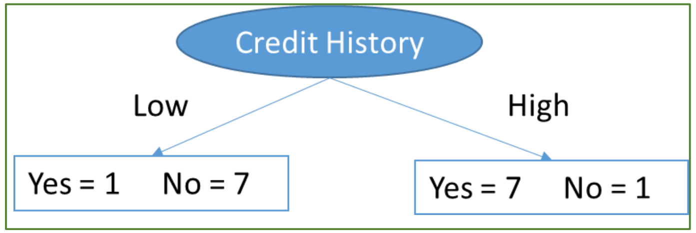
$$Information\ Gain = \boldsymbol{PM} - \frac{N\ Left\ Side}{N\ Before\ Split} * \mathrm{LSM} - \frac{N\ Right\ Side}{N\ Before\ Split} * RSM$$

Metric → Entropy or Gini Value
PM → Parent Metric value

N Left Side → Total number of observations on the left side of the split
N Right Side → Total number of observations on the right side of the split
N Before Split → Total number of observations before the split (Left + Right)

LSM → Left Side Metric Value
RSM → Right Side Metric Value





Ensemble Learning

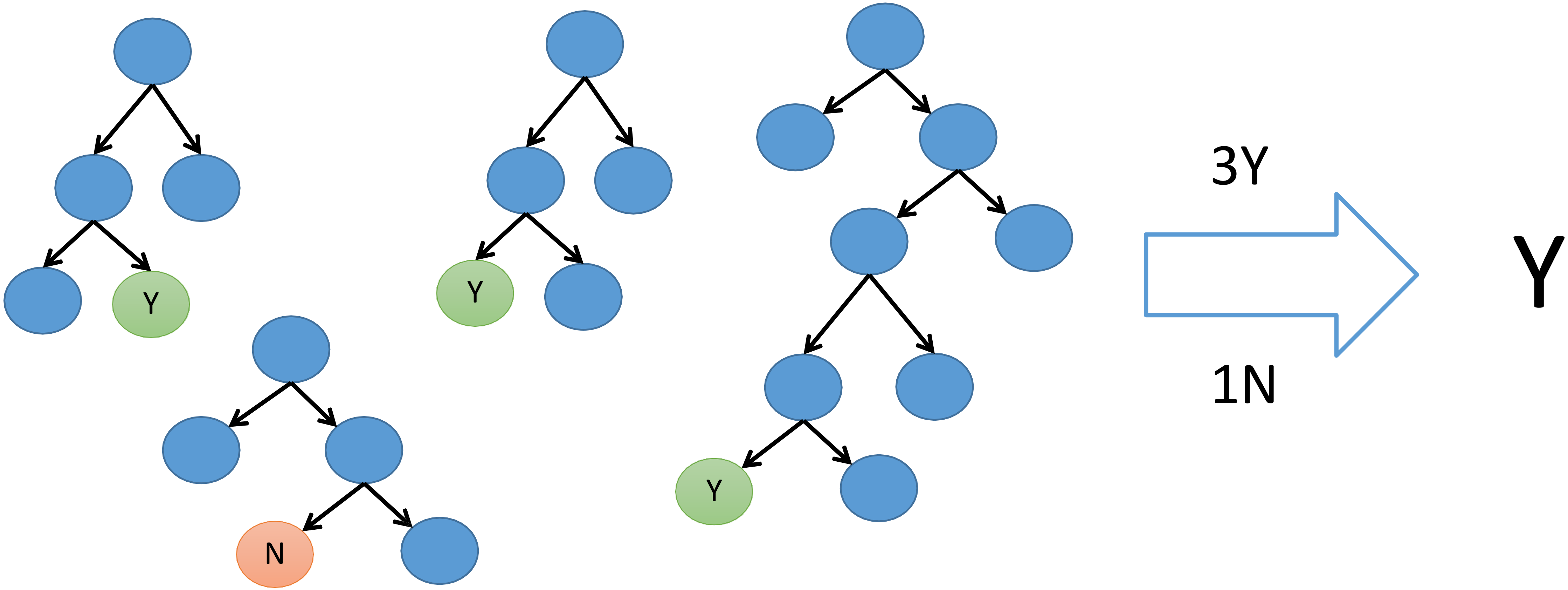
Ensemble Learning

- All algorithms have errors
- Collective wisdom is higher than the individual intelligence
- Generate a group of base learners and combined result gives higher accuracy

- Different base learners can use different,
  - Parameters
  - Sequence
  - Training sets etc
- Two major Ensemble Learning Methods
  - Bagging
  - Boosting

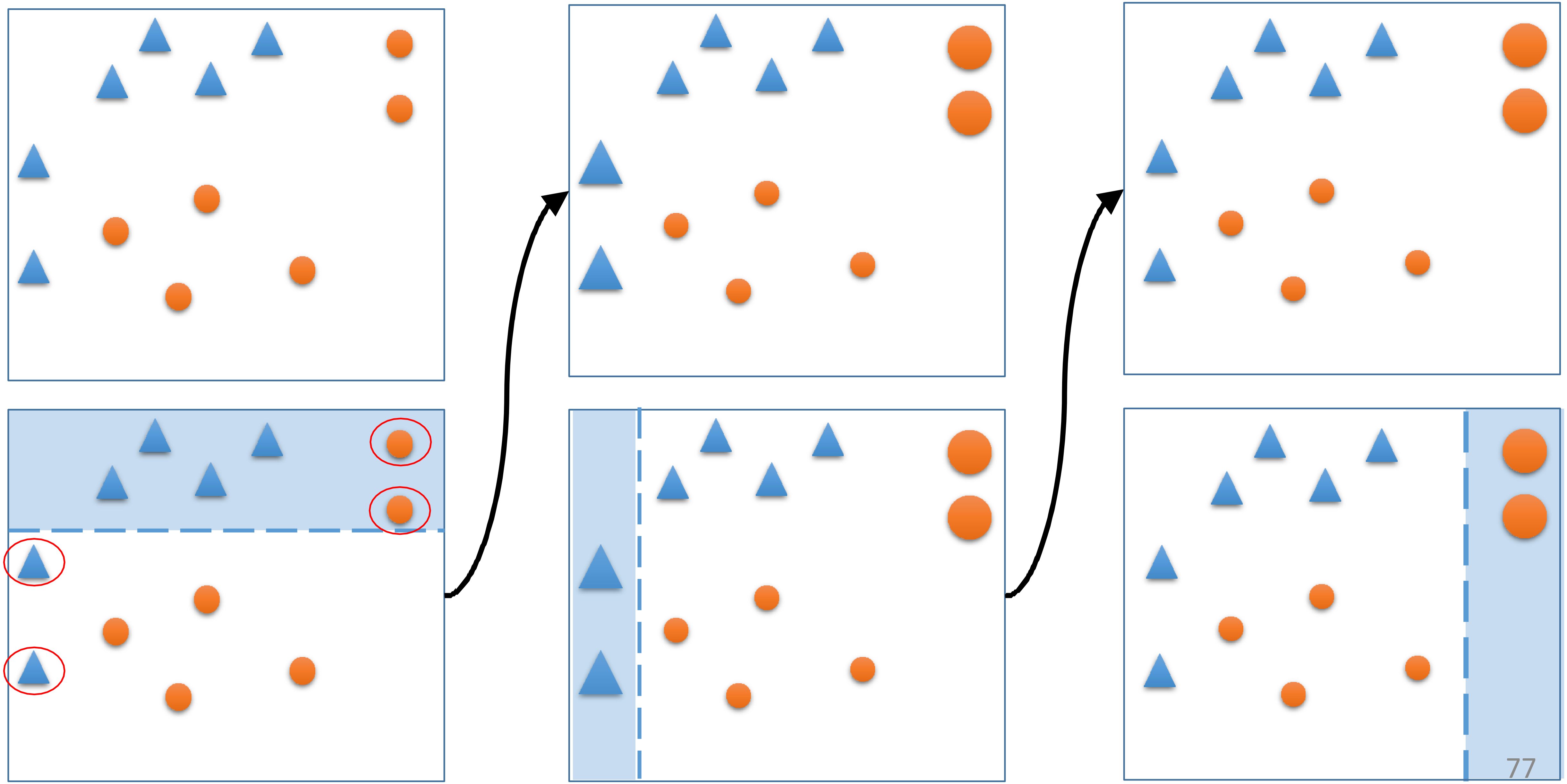
Bagging

- Various models are built in parallel
- All models vote to give the final prediction

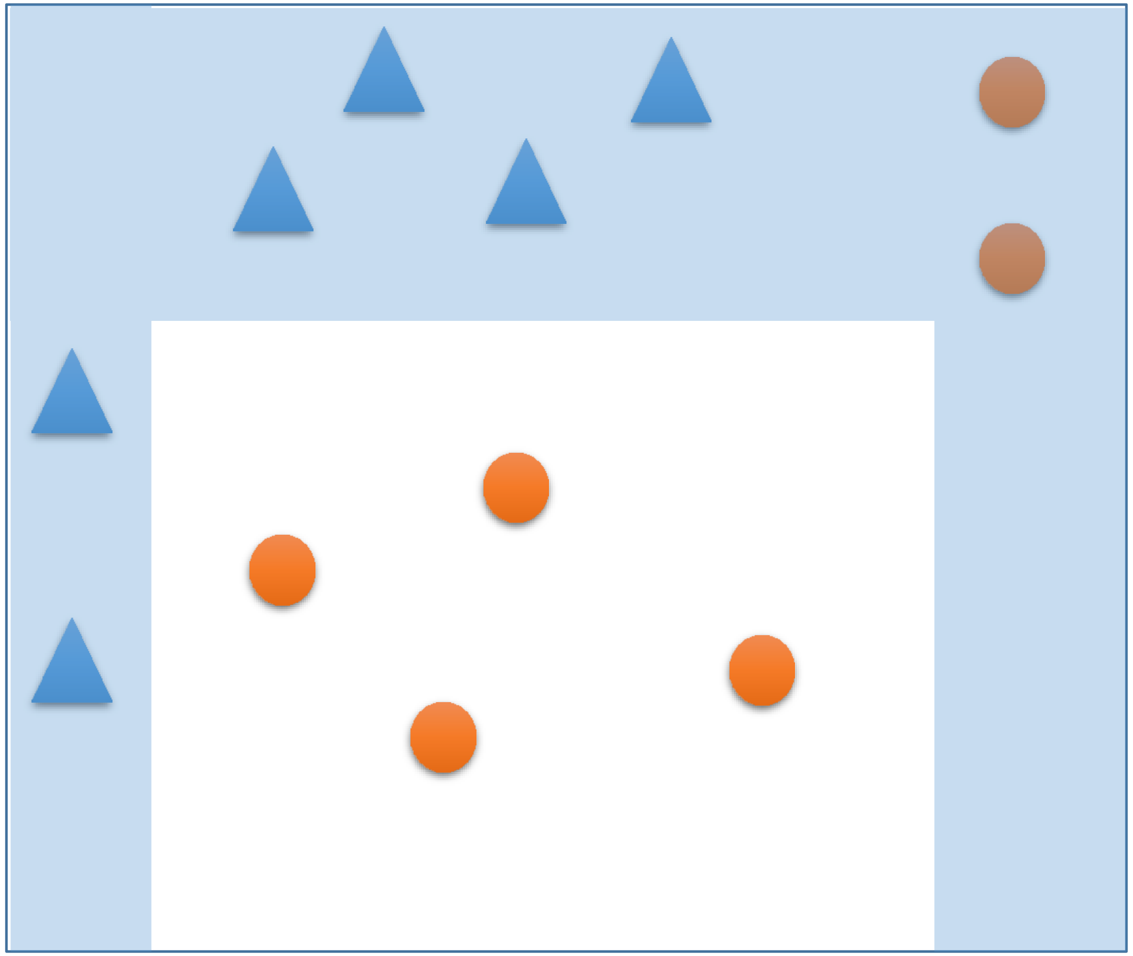


Boosting

- Train the Decision Tree in a sequence
- Learn from the previous tree by focussing on incorrect observations
- Build new model with higher weight for incorrect observations from previous sequence



Boosted Model



Evaluation

How good is the model in predicting different classes?

1. Loan Approval Prediction
2. Adult Income Class Prediction
3. Fraud Prediction (Hypothetical)

Loan Approval Prediction

Loan_ID	Gender	Married	Dependents	Self_Employed	Income	LoanAmt	Term	CreditHistory	Property_Area	Status
LP001002	Male	No	0	No	$5,849.00		60	1	Urban	Y
LP001003	Male	Yes	1	No	$4,583.00	$128.00	120	1	Rural	N
LP001005	Male	Yes	0	Yes	$3,000.00	$66.00	60	1	Urban	Y
LP001006	Male	Yes	2	No	$2,583.00	$120.00	60	1	Urban	Y

- Automate loan eligibility process
- Identify customers whose loan will be approved

gender	married	ch	income	loanamt	status
		1	5849		Y
		0	4583	128	N
		1	3000	66	Y
		1	2583	120	Y
		1	6000	141	Y
		1	5417	267	Y

Accuracy Score – Loan Approval Prediction

	Predicted Negatives	Predicted Positives
Actual Negatives	True Negatives	False Positives
Actual Positives	False Negatives	True Positives

	Predicted Negatives	Predicted Positives	
Actual Negatives	29	20	49
Actual Positives	2	108	110
	31	128	159

Logistic Regression – Loan Approval prediction

	0	1
0	29	20
1	2	108

$$Accuracy = \frac{TN + TP}{Total\ Observations}$$

$$= \frac{29 + 108}{159}$$

= 0.8616

Accuracy Score – Adult Income Class Prediction

	Predicted Negatives	Predicted Positives
Actual Negatives	True Negatives	False Positives
Actual Positives	False Negatives	True Positives

	0	1
0	3814	559
1	800	764

	Predicted Negatives	Predicted Positives	
Actual Negatives	TN = 3814	FP = 559	4373
Actual Positives	FN = 800	TP = 764	1564
	4614	1323	5937

$$Accuracy = \frac{TN + TP}{Total\ Observations}$$

$$= \frac{3814 + 764}{5937}$$

= 0.77

Accuracy Score – Fraud Prediction

	Predicted Negatives	Predicted Positives
Actual Negatives	True Negatives	False Positives
Actual Positives	False Negatives	True Positives

	Predicted Negatives	Predicted Positives	
Actual Negatives	TN = 9500	FP = 400	9900
Actual Positives	FN = 20	TP = 80	100
	9520	480	10,000

$$Accuracy = \frac{TN + TP}{Total\ Observations}$$

$$= \frac{9500 + 80}{10000}$$

= 0.9580

Loan Approval 0.86

		Predicted 0	Predicted 1
Actual	0	TN = 29	FP = 20
	1	FN = 2	TP = 108

Fraud Detection 0.958

		Predicted 0	Predicted 1
Actual	0	TN = 9500	FP = 400
	1	FN = 20	TP = 80

Adult IncomeClass 0.77

		Predicted 0	Predicted 1
Actual	0	TN = 3814	FP = 559
	1	FN = 800	TP = 764

Null Accuracy

		0	1
Actual	0	TN = 0	FP = 49
	1	FN = 0	TP = 110

$$Accuracy = 0.69$$

		0	1
Actual	0	TN = 9900	FP = 0
	1	FN = 100	TP = 0

$$Accuracy = 0.99$$

		0	1
Actual	0	TN = 4373	FP = 0
	1	FN = 1564	TP = 0

$$Accuracy = 0.73$$

Classification Measures
&
Reports

Classification Measures

$$Accuracy = \frac{TN + TP}{Total\ Observations}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Specificity\ or\ Selectivity = \frac{TN}{TN + FP}$$

True Negative Rate

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN}$$

True Positive Rate

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Which metric to use?

- High Accuracy is nice to have
- High Precision when its OK to have false negatives

Precision =1

Actual			
	0	TN = 9900	FP = 0
	1	FN = 30	TP = 70

Recommendation

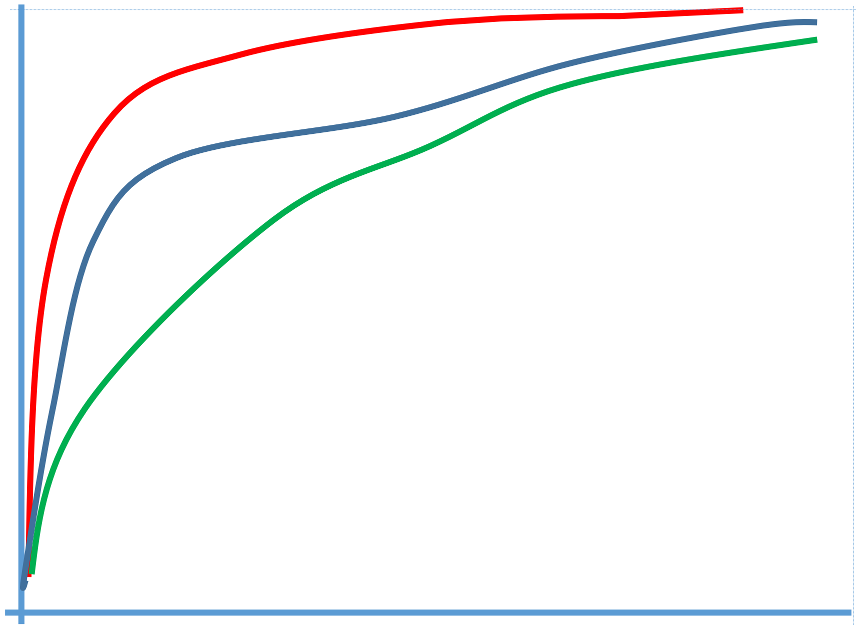
- High recall or sensitivity when cost of false negative is very high

Recall = 0.9

Actual			
	0	TN = 9878	FP = 22
	1	FN = 10	TP = 90

Loan Default

AUC ROC



AUC – Area Under the Curve

ROC – Receiver Operating Characteristics

Provides a single number that lets you compare models of different types.