# Linear Models

- Linear Regression
- Logistic Regression
- Support Vector Machines (< clf Reg)

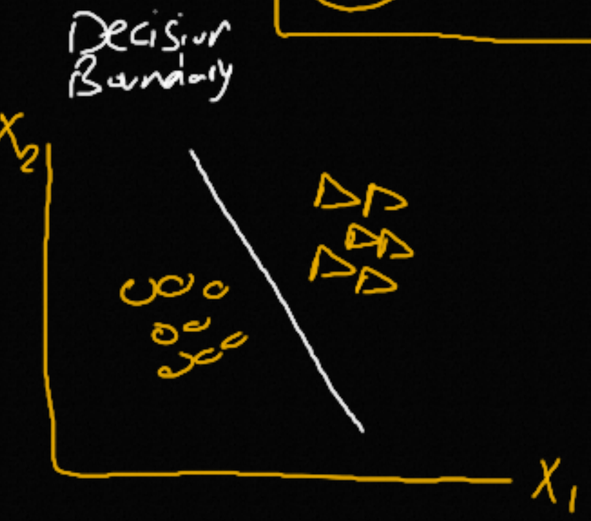
clf

clf

$w^T x \geq 0 \quad cls(1)$

$w^T x < 0 \quad cls(0)$

$y < \begin{matrix} 0 \\ 1 \end{matrix}$



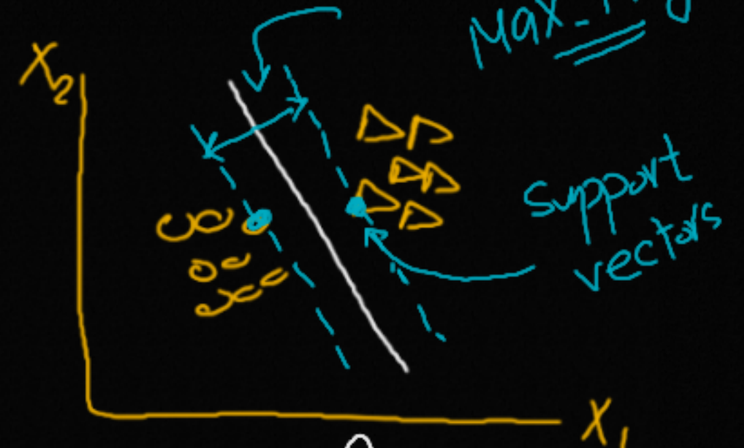

→ Cost fn (log loss)

→ Decision Boundary

$w^T X = 0$

objective

Max_Margin

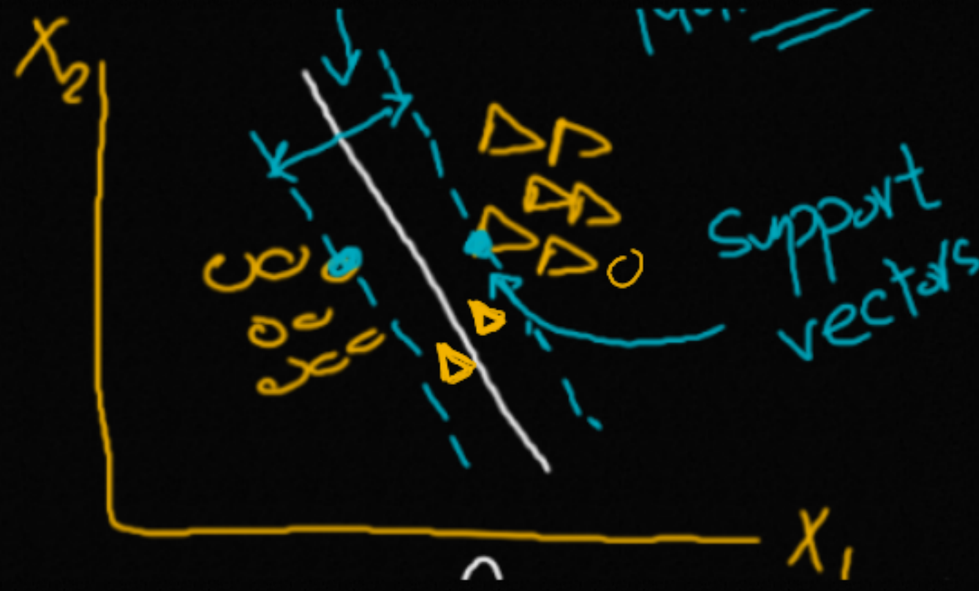



→ Cost fn (hinge loss)

→ $w^T X = 0$

→ $w^T X = +1$

→ $w^T X = -1$




May lead to overfitting

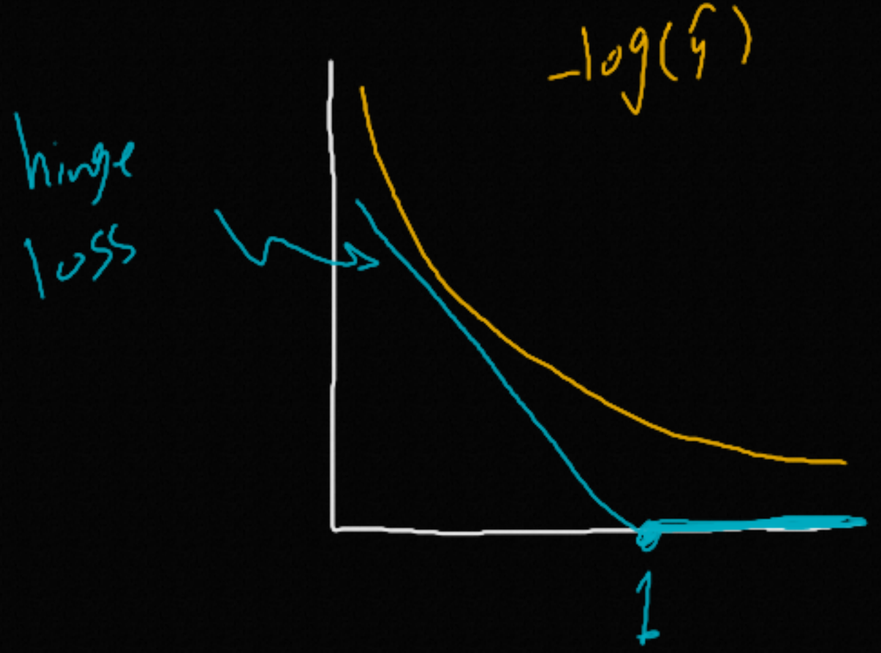
Max Margin → Hard Margin → No Misclassification
Max Margin → Soft Margin → Allow Misclassification

More General Model

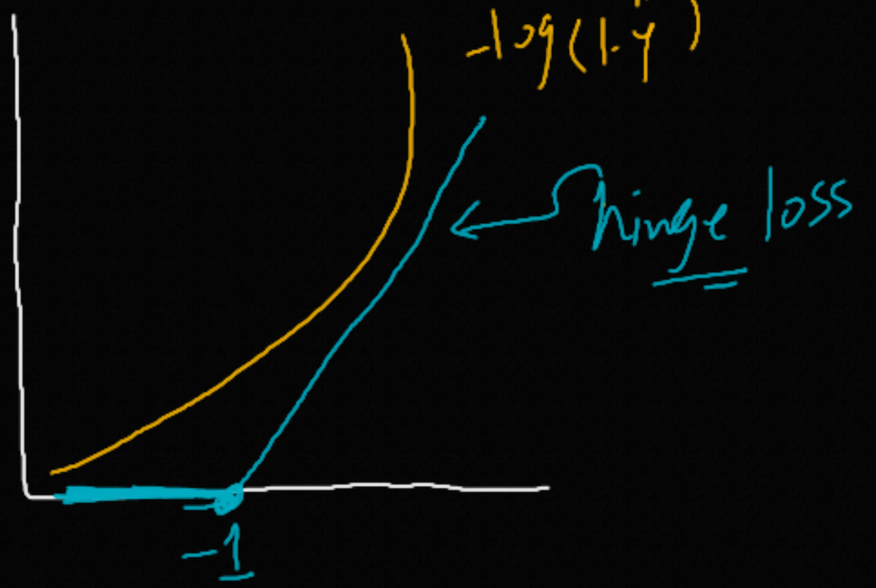
$y \begin{cases} 1 \\ -1 \end{cases}$

cls(1) $w^T x \geq +1$

cls(-1) $w^T x \leq -1$




$y = 1$




$y = 0$

- Cost fn: Hinge loss $f(w)$
- Gradient Descent: $\Rightarrow$ get weights

* Linear Separable
  - LinearSVC (Large datasets)
  - SVC ( kernel = 'linear' )

* Non linear
  - SVC ( kernel = { 'poly' / 'rbf' } )

- Non linear

Non linear Model

↳ ① Apply poly ② use linear Model
↳ Log Reg
↳ Linear SVC

for large datasets
~ "impractical"
↳ Computational Complexity

↳ Use SVC with Kernels

→ get the effect of transformation
without applying transformation

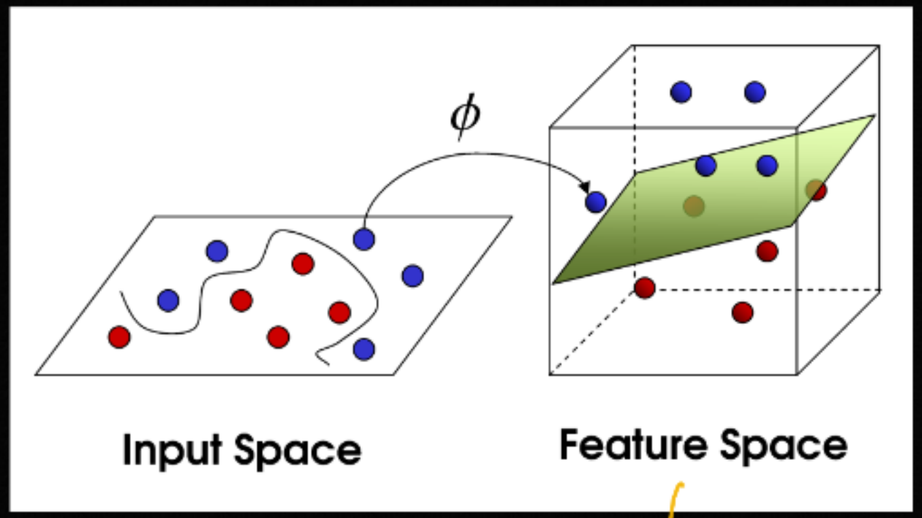
Kernel Trick


- Non linear
  - Kernelized SVC
    - polynomial → degree, coef0
    - rbf (gaussian) → gamma

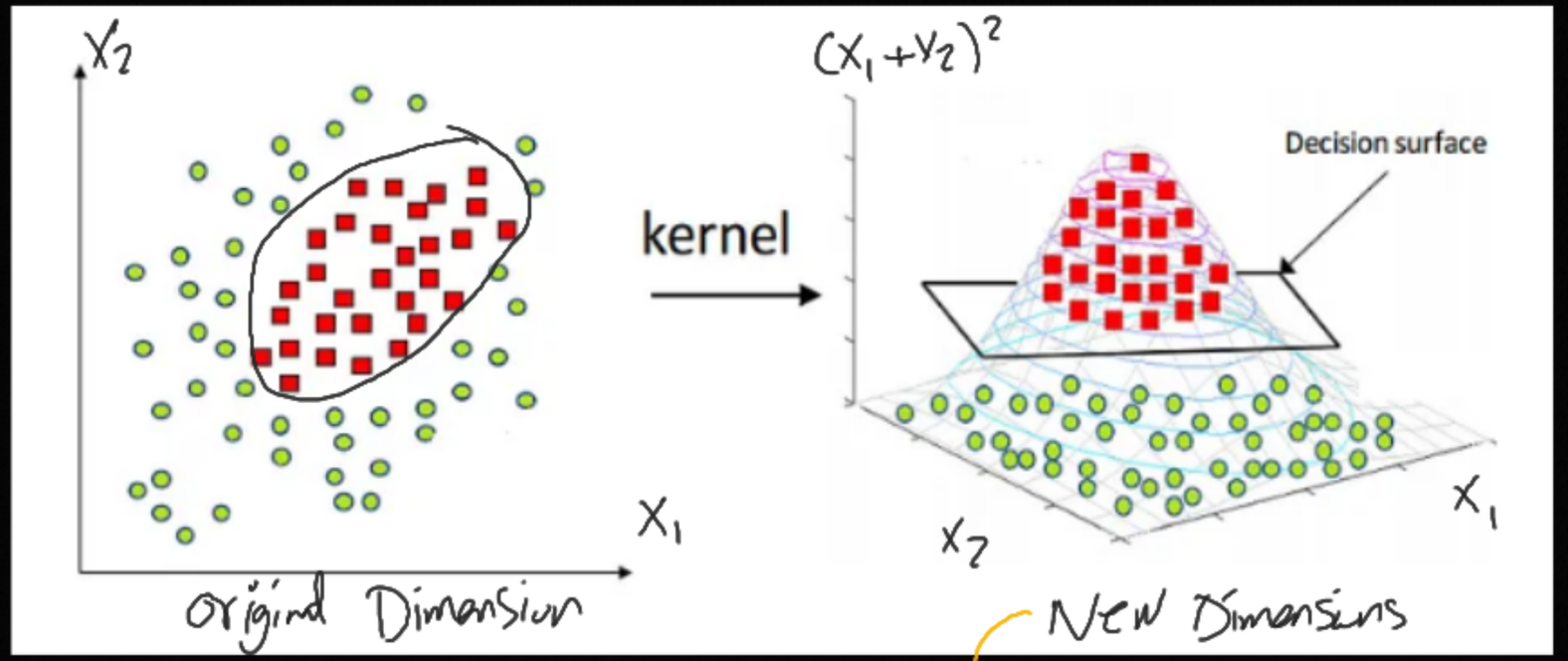
original Dimension → New Dimension

- Can't find Linear Model to fit the data
- We can find linear Model to fit the data

# Kernelized SVM



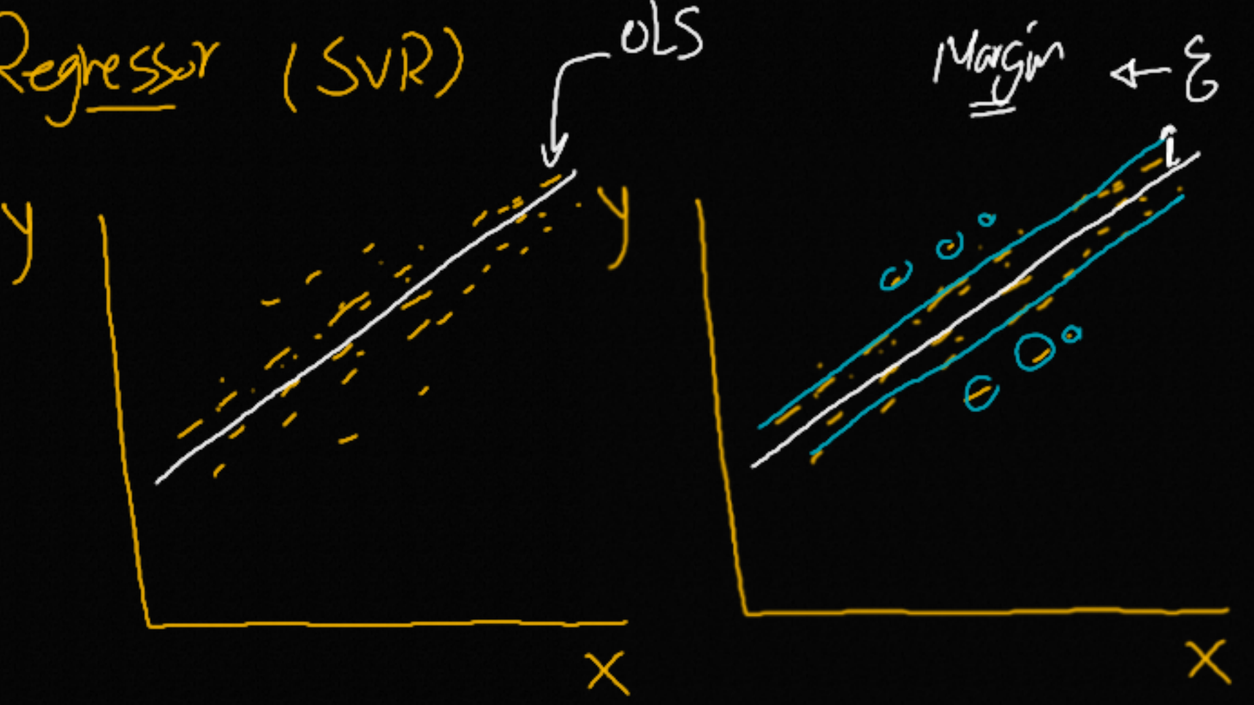

Linear Separation




Linear Separation

# Support vector Regressor (SVR)

- LinearSVR
- SVR ( kernel = 'linear' / 'poly' / 'rbf' , ... )




overfitting

↝ C → Regularization

Liner Regressor

SVR

cost fn

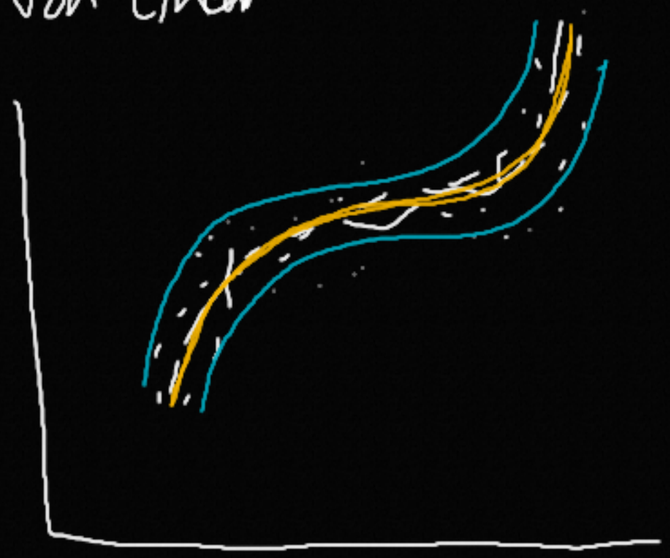
ε ↓↓ → underfitting

ε ↑↑ → overfitting

→ points within Margin
↳ Cost fn ≃ 0

→ points out of Margin
Cost fn ↑↑

* Non Linear



Kernel < Poly (degree)
rbf (gamma)

# ML Models

## Model Based

→ During Learning:
- Use training Data

→ After learning:
- Model (Eqn)
  - ↳ for prediction

Training Data have no effect

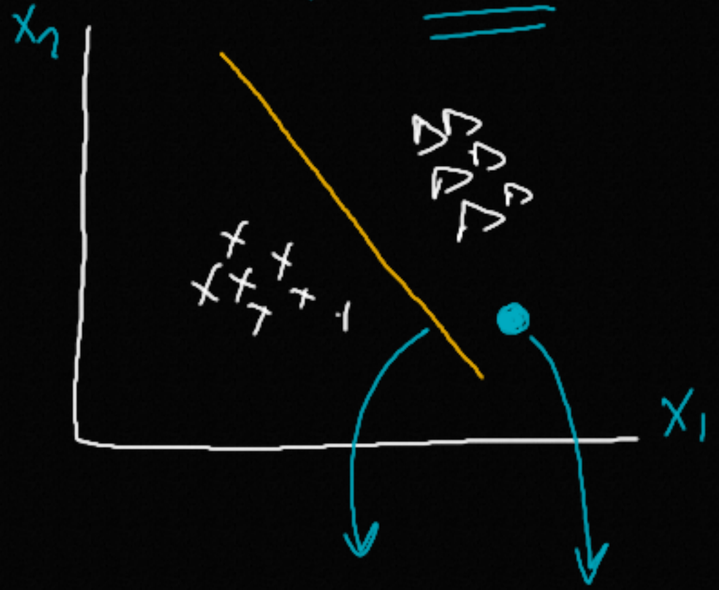
## Instance Based

→ During Learning:
- Memorize training Data

→ After Learning:
- No Model

→ prediction (Similarity)
- ↳ use training Data

# Model-Based



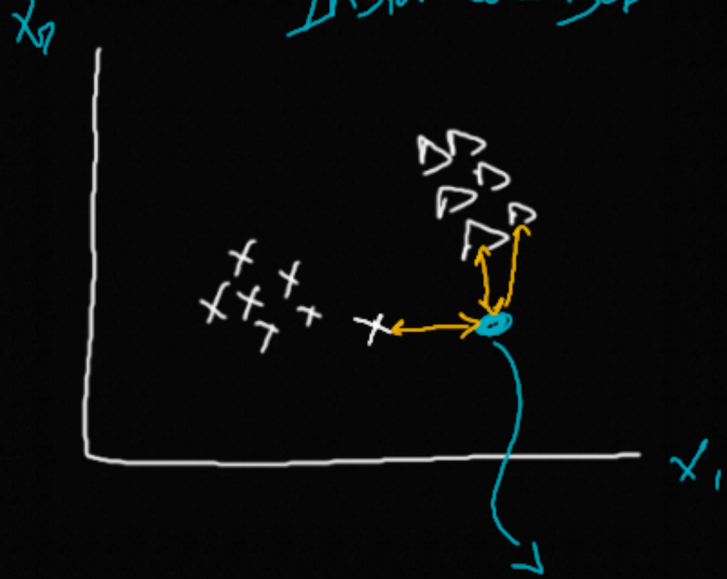

Model → new point

pred

cls ①

cls ⓪

# Instance-Based




Training
↳ Memorize

K = 3

Nearest K-points from train data ← new point

2 ✕ 1

Vote → Mode

↘ pred △

الحصول على شيء
نفع

# K-Nearest Neighbors

Lazy Algorithm

→ Training : very fast
→ prediction : slow

→ Similarity → Distance ??

→ n_neighbors (k) ??

① Distance
1- Euclidean
2- Manhattan
3- Cosine Similarity

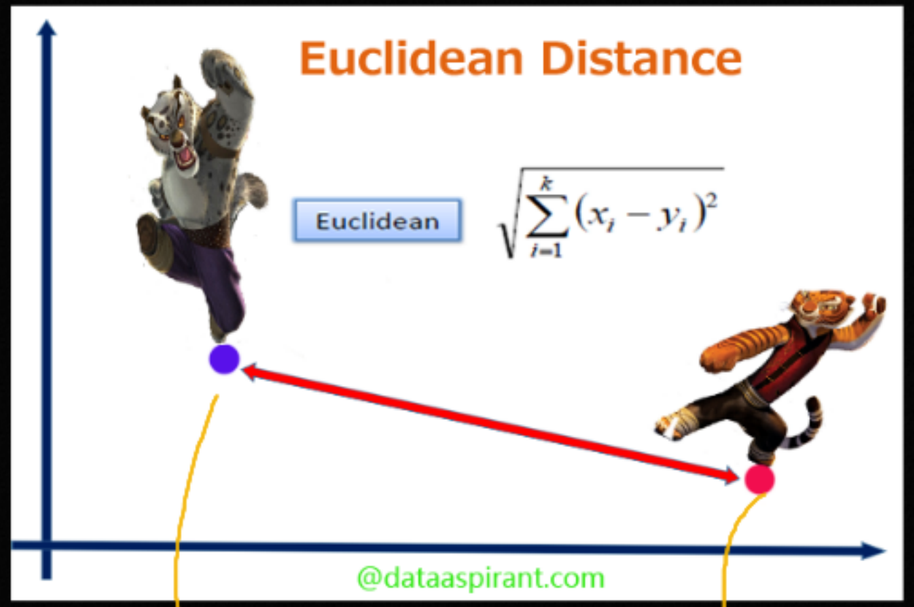
② n_neighbors

K=1 → overfitting

K=n → Constant pred (Mode)
underfitting

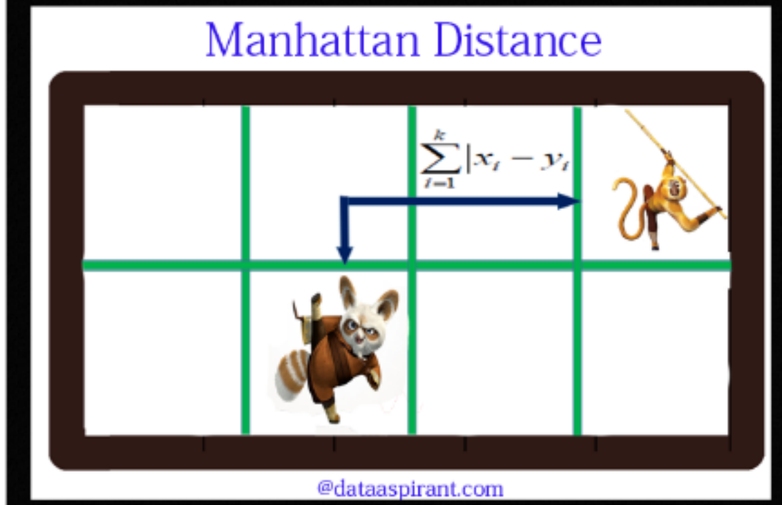
n = 5000 points
3000 (1) ⟷ 2000 (0)

# Distance Metrics

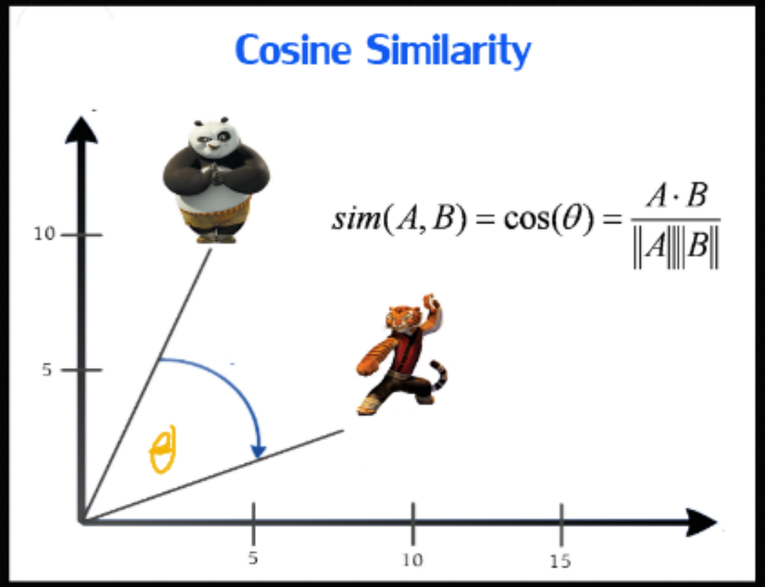



$(x_2, y_2)$ $(x_1, y_1)$

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

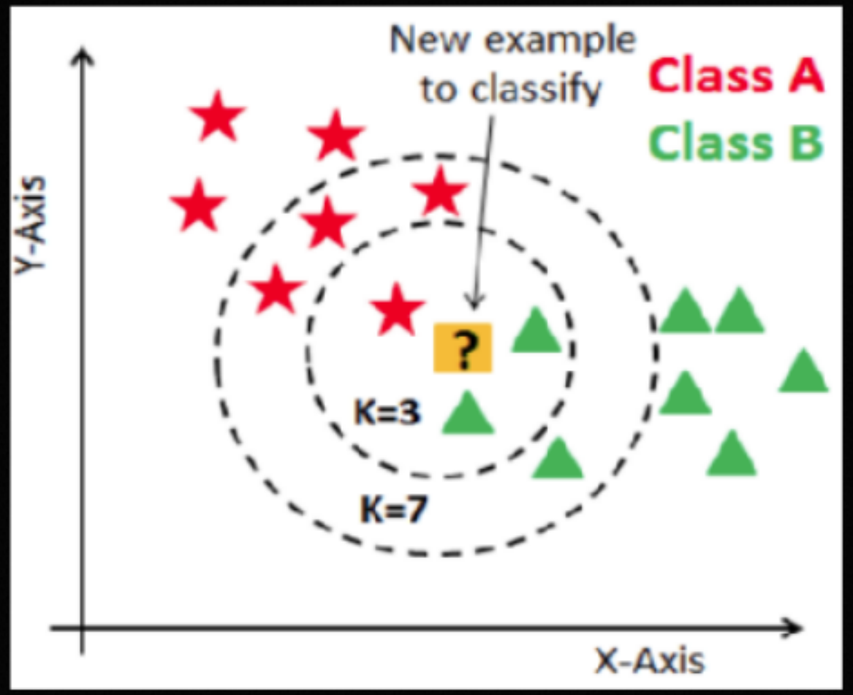


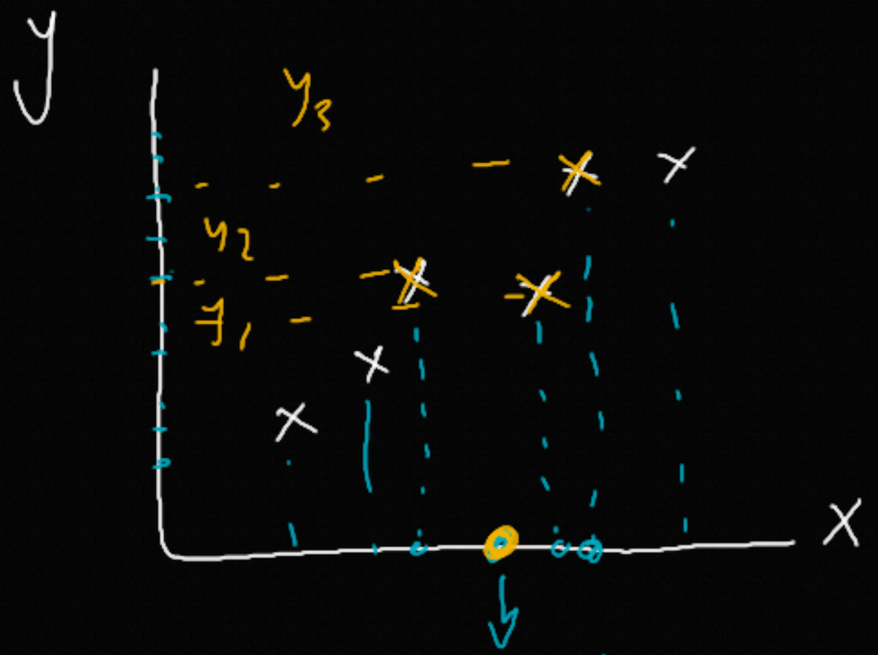

$$|x_2 - x_1| + |y_2 - y_1|$$




$\cos(\theta)$

# K (n_neighbors)






$$y_{pred} = \frac{y_1 + y_2 + y_3}{3}$$

# Knn Regressor

① During fitting:

→ Memorize train data

② During prediction

→ use train data (k_nearest)

Avg

Reg
Clf

Decision Tree

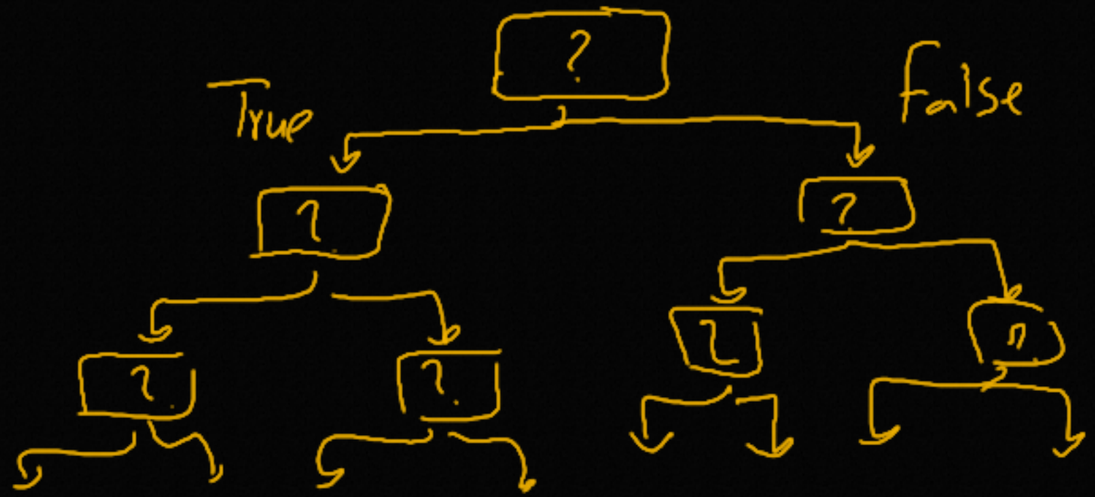
(Interpretable ML)
(White Box)

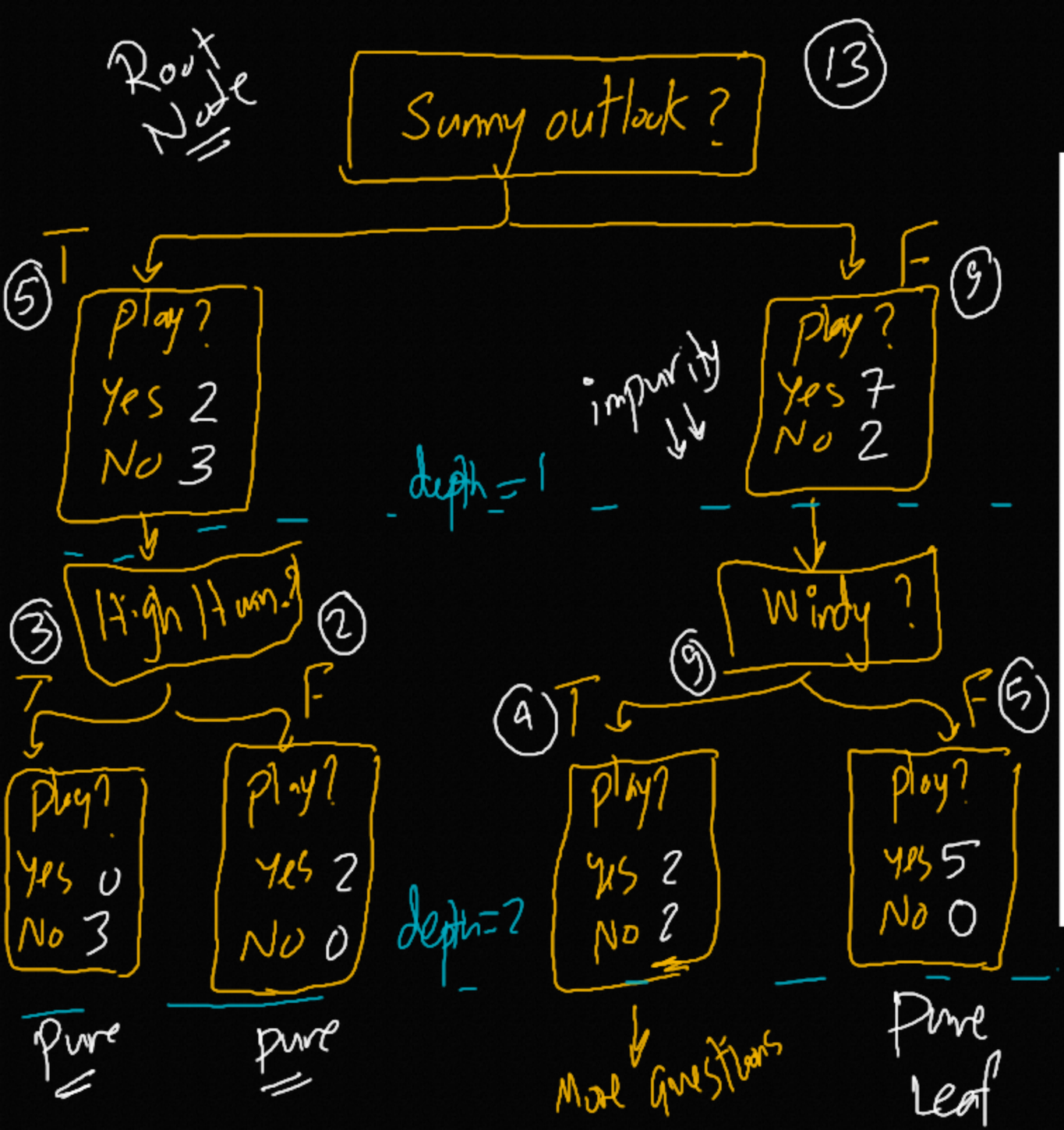
Linear

Non linear

Base Model => for Ensemble Models

- Business / Management
- Troubleshooting
- Coding (if/else)
- Guessing Game






features

Target

| Outlook | Temperature | Humidity | Windy | PlayTennis |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

5 No
9 Yes

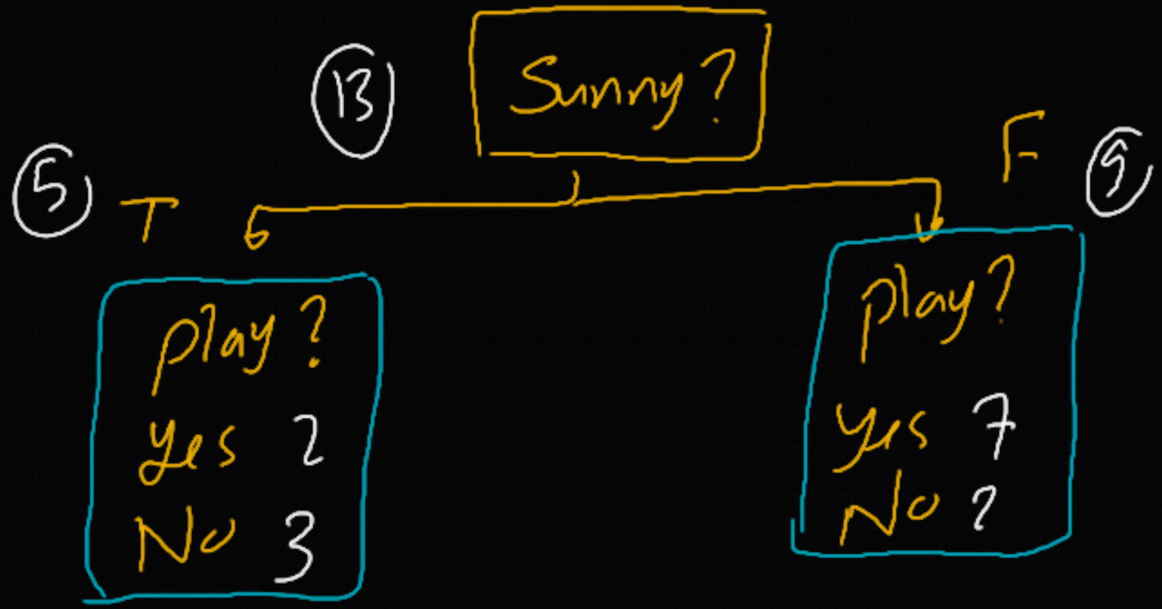
* Question Quality Metric → Information Gain ↑↑ (Impurity ↓↓)

* No. of Questions

Error

Gini Index

Entropy




$G_{left}$

$$1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$$

$G_{right}$

$$1 - \left(\frac{7}{9}\right)^2 - \left(\frac{2}{9}\right)^2 = 0.35$$

Gini-index

$$\text{Total Gini} = \frac{5}{13} \times 0.48 + \frac{9}{13} \times 0.35 = 0.4$$

* Gini

$$G = 1 - \sum p^2$$

$$J = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

Cost fn

* Entropy

$$H = -\sum p \log_2(p)$$

---

*Numerical feature