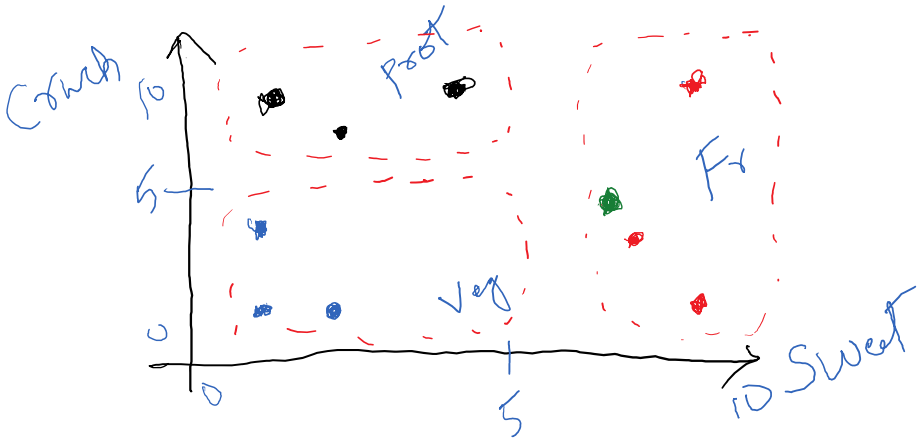
KNN → K- nearest Neighbour.

Features | Label

Ingredient	Sweet	Crunch	Food type
Apple	10	9	Fr
bacon	1	4	Pr
banana	10	1	Fr
carrot	7	10	Veg
Celery	3	10	Veg
Cheese	1	1	Pr



→ Simplest algorithm → Classification tasks → similar class will have similar characters.

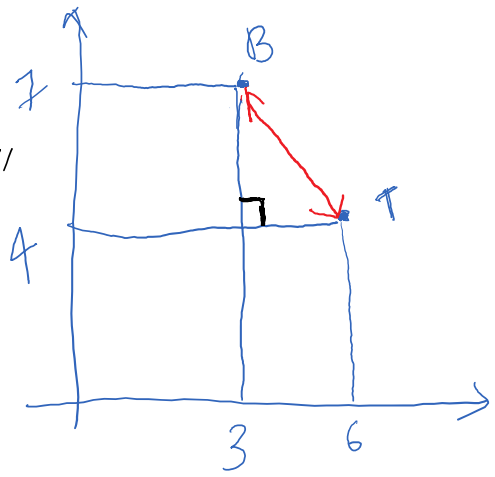
→ Process → Select → K. (K-NN)

→ Training dataset → classifications & labels. (Features)

Each → Unnamed variable → Category is Identified

Euclediandist → "How a crow flies"

Manhattan Dist → "Man walks in a block"



	S	C	L	Dist
grape	8	5	Fr	2.2 ✓
bean	3	7	Veg	4.2
nuts	3	6	Pr	3.6 ✓
Orange	7	3	Pr	1.4 ✓
Tomato	6	4		

$$dist(T,B) = \sqrt{(6-3)^2 + (4-7)^2}$$

$$= 4.2$$

K	Conclusion
1	Fr
2	Fr + Fr
3	Fr + Fr + Pr

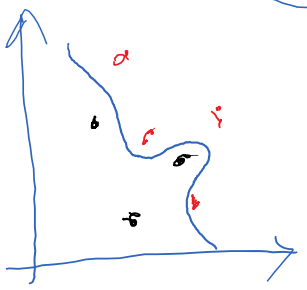
$K \simeq \sqrt{n}$

K = Most → Odd (Tie)

→ Avoid ↑ & ↓ values of K

→ Ignore minor / insignificant factors.

Higher K.



optimum K

Lower K

→ Different scales → Computation becomes very difficult

(Min-Max) Normalisation

$x - \min(x)$

(Z) Standardisation

$x = \frac{x - \mu}{}$

Sweetness → 0 - 1

Crunch → 0 - 1

Scoville → 0 - 1 Million

$X_{new} = \frac{X - Min(X)}{Range}$

$X_{new} = \frac{X - \mu}{\sigma}$ → Z-Standardization

Scoville → 0 - 1 Million

Dummy Coding → Nominal data (M/F) (T/F) (Y/N)

Feature → (Y/N) → 1 = Yes
0 = Otherwise

2 → features
1 → DummyCode

→ Cold, Med, Hot

Feature — C → 1 → Cold
0 → otherwise

M → 1 → Medium
0 → otherwise

H < ∞

3 → features
2 → DummyCodes

n → features
n-1 → DummyCodes

Building Model

S_1 → Data Collection

S_2 → Data Exploration

→ Transform → Normalization ⊛ → Min-Max

→ Z-standardize ⊛

→ Create a Train & Test data.

S → Training the model.

S_3 → Training the model.

p ← knn(train = ... , test = ... ; cl = ... , k = ...)

cl is a vector with class of each variable

S_4 → Evaluate performace → Cross Table

S_5 → Improve performe → change Normalisation etc → min-max, z

→ Test Values

S_6 → Summary → K = ? , Normalization (Min/max, z) , % succes

Decide on Trials