Probabilistic approach with Naive bays algorithm

Bayesian probability theory is rooted in the idea that the estimated likelihood of an event should be based on the evidence at hand.
Events are possible outcomes, such as sunny or rainy, head or tail
Trial is a single opportunity for the event to occur such as day's weather. a coin flip.
The probability of an event can be estimated from observed data by dividing the no of trials in which an event occured by the total no. of trials
For eg: if it rained 3 out of 10 days, the probability of rain can be estimated as 30 persent.
The total probability of all possible outcomes of a trial must always be 100%.
Mutually exclusive and exhaustive: If the trials has only two outcomes that cannot occur simultaneously.
Joint probability: The probability that two events will both occur. In other words, joint probability is the likelihood of two events occuring together for joint probability, the events must be independent.
Independent events: if the outcome of one event does not affect the outcome of the other event; for eg: the probability of getting heads on two coin tosses.
Dependent events: if the outcome of one event affect the outcome of other event; for eg: the probability of clouds in the sky has an impact on the probability of rain that day.

Joint probability $= P(A \cap B) = P(A) \times P(B)$

where;

$P(A \cap B)$: joint probability of event A & B

$P(A)$: probability of event A occuring

$P(B)$: probability of event B occuring

Conditional probability: probability of an event occuring given that another event has already occured. The relationship b/w depent events can be described using Baye's theorem,. The notation $P(A|B)$ can be read as the Probability of event A given the event B occured.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

Using Bayes theorem, we can find the probability of A happening, given that B has occured. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. ie presence of one particular feature does not affect the other. Hence it is called naive it is called Bayes because it depends on the principle of Baye's Theorem.

Working of Naive Baye's :

Eg:

S_1	Outlook	play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

step 1: Convert the dat set into a freequency table

weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	4

step 2: Create likelihood table by finding probability

weather	Yes	No	
Overcast	5	0	$5/14 = 0.35$
Rainy	2	2	
Sunny	3	2	$4/14 = 0.29$
			$5/14 = 0.35$
All	$10/14 = 0.71$	$4/14 = 0.29$	

step 3: Applying Baye's theorem

$$P(\text{Yes}/\text{sunny}) = \frac{P(\text{sunny}/\text{Yes}) \times P(\text{Yes})}{P(\text{sunny})}$$

$P(\text{sunny}/\text{yes}) = 3/10 = 0.3$

$P(\text{sunny}) = 0.35$

$P(\text{yes}) = 0.71$

So, $P(\text{Yes}/\text{sunny}) = \frac{0.3 \times 0.71}{0.35} = \underline{0.60}$

$$P(\text{No}/\text{sunny}) = \frac{P(\text{Sunny}/\text{No}) \times P(\text{No})}{P(\text{sunny})}$$

$p(\text{sunny}/\text{no}) = 2/4 = 0.5$

$p(\text{no}) = 0.29$

$p(\text{sunny}) = 0.35$

$$p(\text{no}/\text{sunny}) = \frac{0.5 \times 0.29}{0.35} = \underline{\underline{0.41}}$$

Hence, $p(\text{Yes}/\text{sunny}) > p(\text{no}/\text{sunny})$

Lazzy learning approach with KNN algorithm

In machine learning, lazy learning method is which generalization of the learning data. In theory, delayed until a query is made to the system as opposed to eager learning, where the system tries to generalize the training data before receiving queries.
Lazy learning simply store the data & generalizing beyond these data is postponed until a explicit & request is made. Here no abstraction occurs, abstraction & generalization process are skipped altogether.
Nearest neighbour classifiers are defined by their characteristic of classifying unlabled examples by assigning them the class of similar labeled examples. These methods are really powerfull. They have been used succesfully for:

- Computer vision app. including optical character recognition and facial recognition in both still images & video.
- predicting whether a person will enjoy a movie/music recommandation
- Identifying patterns in generic data, perhaps to use them in detecting specific proteins/diseases.

strengths :-

- simple & effective
- makes no assumptions about the underlying data distribution
- fast training phase

weakness :-

- Doesn't produce a model, limiting the ability to understand how the features are related to the class
- Requires selection of an appropriate k
- slow classification phase
- Nominal features & missing data require additional processing
- KNN is based on Eucledian distance

$$dis(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + \dots (p_n-q_n)^2}$$

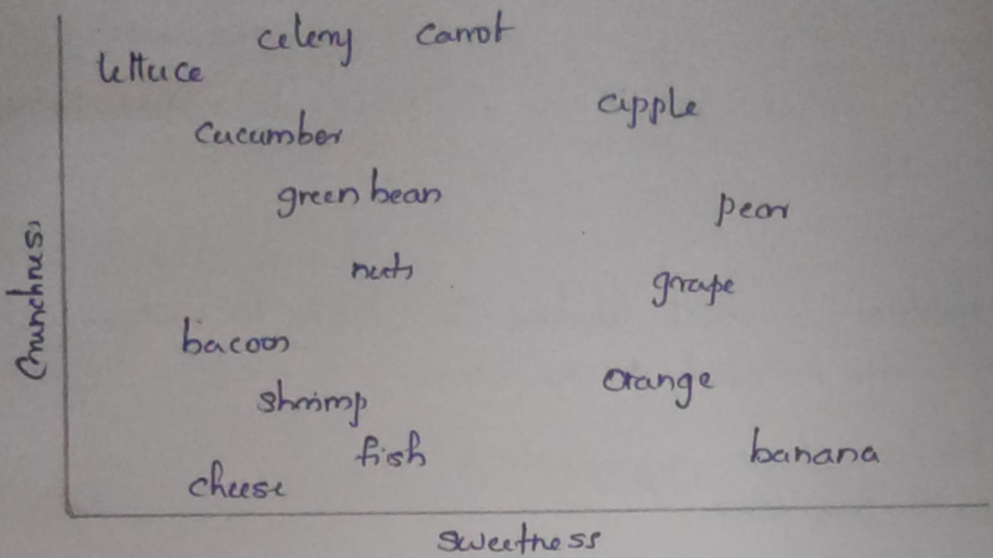
Eg:

Ingredient	Sweetness	Crunchiness	Food Type
Apple	10	9	Fruit
Bacon	1	4	protein
Banana	10	1	fruit
carrot	7	10	Vegetable
celery	3	10	Vegetable
cheese	1	1	Protein

→ Free Space

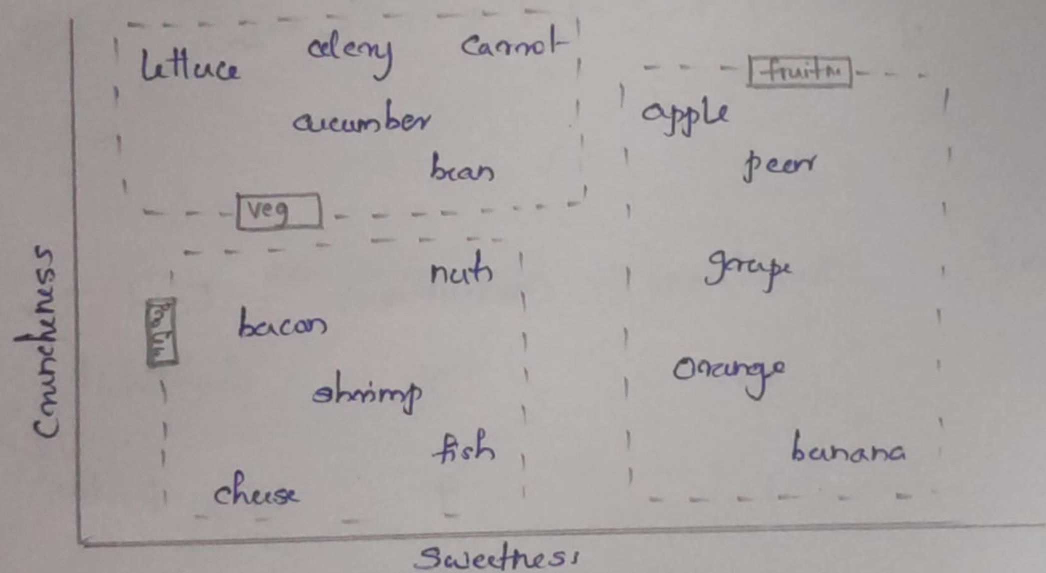
KNN treats the features as coordinates in a multidimensional feature space

Scatter plot

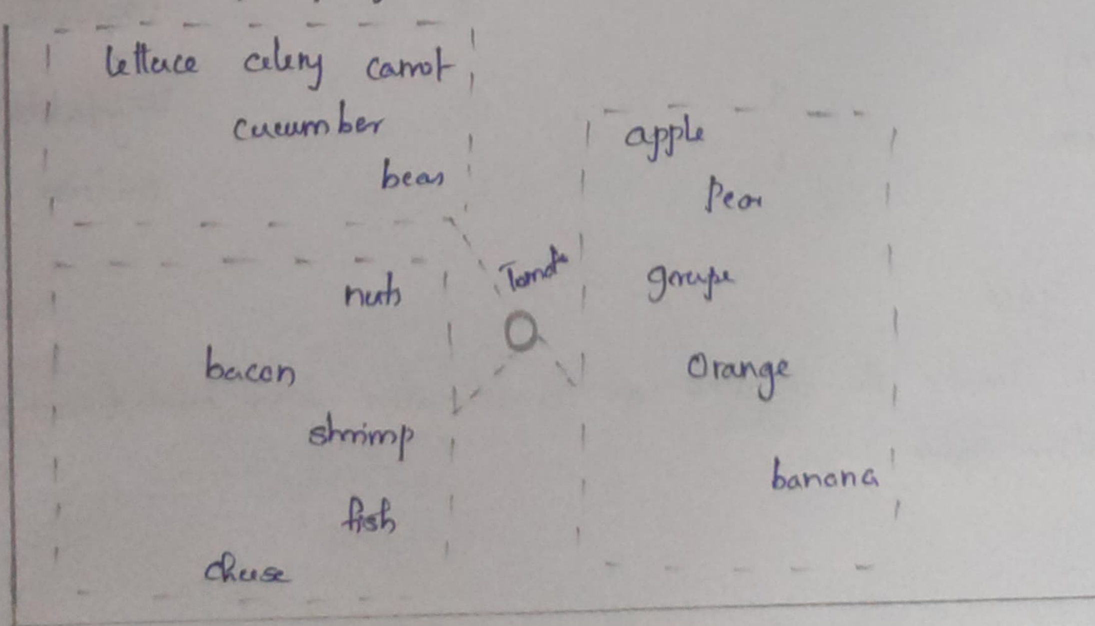


The pattern

Similar types of food tent to be grouped closely together



→ Is tomato a fruit/veg ?



distance b/w tomato (sweetness $\rightarrow$ 6, Crunchiness $\rightarrow$ 4) and green bean (sweetness $\rightarrow$ 3 Cruncheness $\rightarrow$ 7)

$$dis(\text{tomato}, \text{greenbean}) = \sqrt{(6-3)^2+(4-7)^2} = 4.2$$

Simillarly,

Ingredient	Sweetnes	crunchiness	fruittype	distance
Grape	8	5	Fruit	2.2
Green bean	3	7	Veg	4.2
Nuts	3	6	proteins	3.6
Orange	7	3	Fruit	1.4

Here, Orange has a distance of 1.4

$\therefore$ Tomato is a fruit