SML

U-3

(1) Naive Bayes Classification

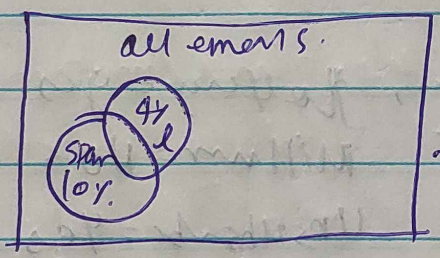
→ It is a classification algo based on bayes's theorm

→ It describes the prob. of an event based on prior knowledge

→ It calc. the prob. of an event based on prior prob. of the evnt ~~and~~ and cond. prob of the event.

→ Using Joint prob., we can predict the event outcome

ex→ The venn diag. showing the joint prob. of spam with lottery



l → lottery

→ not all spam msg contain word lottery and not every mail with lottery is spam

→ Using cond. prob., we can calc. the relationship b/w dependent evnts. for exmp, A & B are two evnts, calc. P(A|B), i.e the prob. of event A occured, given the fact that event B is already occured.

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

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

classification:

→ construct a likelihood table for three words (W1, W2, W3) for 100 emails

	Lottery (W1)		Million (W2)		Unsubscribe (W3)		
Likelihood	yes	no	yes	no	yes	no	total
spam	3/22	19/22	11/22	11/22	13/22	9/22	22
ham	2/78	76/78	15/78	63/78	21/78	57/78	78
total	5/100	95/100	26/100	74/100	34/100	66/100	100

Using Bayes Theorem, Lottery = yes
Million = No
Unsubscribe = Yes.

$$P(\text{spam} \mid w_1 \cap \neg w_2 \cap w_3) = \frac{P(w_1 \mid \text{spam}) * P(\neg w_2 \mid \text{spam}) * P(w_3 \mid \text{spam}) * P(\text{spam})}{P(w_1) * P(\neg w_2) * P(w_3)}$$

$$P(\text{Ham} \mid w_1 \cap \neg w_2 \cap w_3) = \frac{P(w_1 \mid \text{Ham}) * P(\neg w_2 \mid \text{ham}) * P(w_3 \mid \text{ham}) * P(\text{ham})}{P(w_1) * P(\neg w_2) * P(w_3)}$$

$P(Spam \mid w_1 \cap w_2 \cap w_3) =$

$= \frac{3}{22} \times \frac{11}{22} \times \frac{13}{22} \times \frac{22}{100} = 0.00886\,4$

$P(Ham \mid w_1 \cap w_2 \cap w_3) =$

$\frac{2}{78} \times \frac{15}{78} \times \frac{21}{78} \times \frac{78}{100} = 0.00434\,9$

$P(Spam) = \frac{0.00886\,4}{(0.00886\,4 + 0.00434\,9)} = 0.67$

$P(Ham) = \frac{0.00434\,9}{(0.00886\,4 + 0.00434\,9)} = 0.33$

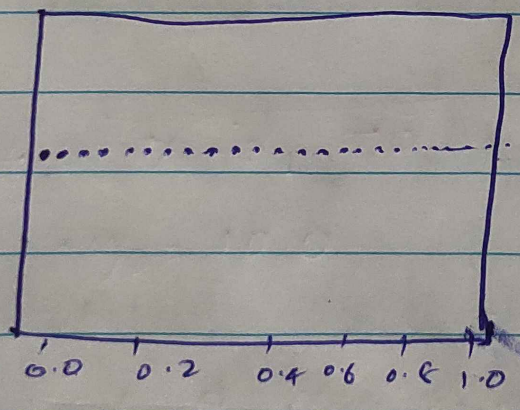
(2) Curse dimensionality with 1D, 2D, 3D example:

→ Curse of dimensionality depends on KNN
→ Let's select 60 Random points

1-D plot.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

one_d_data = np.random.rand(60, 1)
one_d_data_df = pd.DataFrame(one_d_data)
one_d_data_df.columns = ["1D_data"]
one_d_data_df["height"] = 1
```

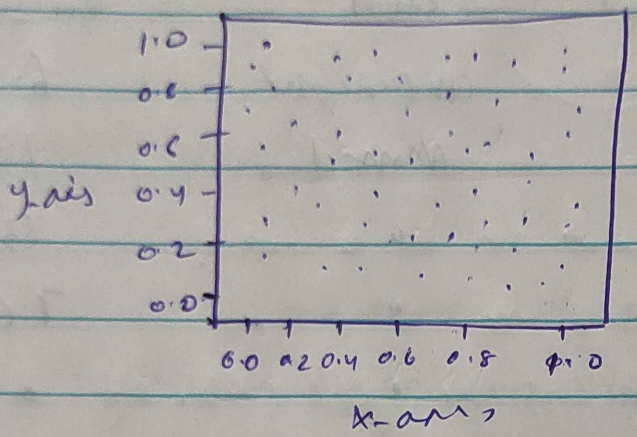


60 data points exported in 1-D

2-D plot.

using x and y, plot them

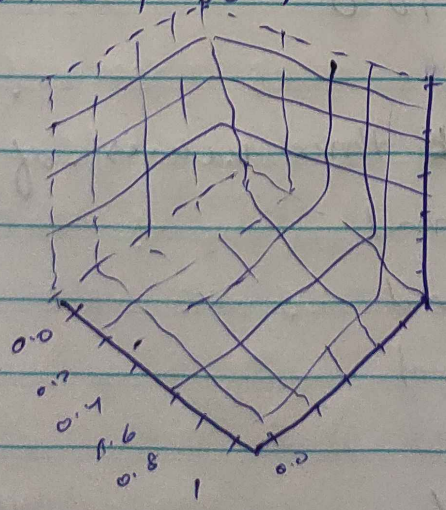
```
two_d_data = np.random.rand(60,2)
two_d_data_df = pd.DataFrame(two_d_data)
two_d_data.columns = ["x_axis", "y_axis"]
```



3-D plot

plotting 3D Space of 60 data pnts.

```
three_d_data = np.random.rand(60,3)
three_d_data_df = pd.DataFrame(three_d_data)
three_d_data_df.columns = ["x axis", "y axis", "z axis"]
```



60 random pnts on 3D

③ KNN classifies with breast cancer Wisconsin data

example:

//Std'D

④ Weight = 57 kgs height = 100 cm (using KNN)

Wt (x_1)	Ht (y_1)	class	Euclidean dist
51	167	underweight	6.7
62	182	Normal	13
69	176	"	13.41
64	173	"	7.615
65	172	"	8.24
56	174	underweight	4.12
58	169	Normal	(1.41) N_1
57	173	"	(3) N_3
55	170	"	(2) N_2

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

$x_2 = 57$, $y_2 = 100$

Select the least three values of K.

$N_1 = 1.41$ = normal

$N_2 = 2$ = normal

$N_3 = 3$ = normal

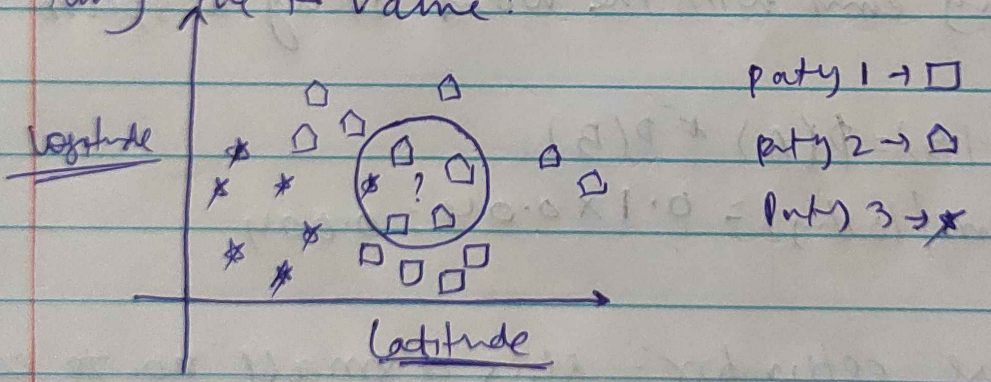
Person is normal.

⑤ KNN

→ Non-parametric ML model
→ Instance-based learning
→ known as lazy learning as it does not learn during the training phase, it stores nothing only during test/evaluation phase for comparison.

KNN voter example:

→ Objective is to predict the party
→ Voters will vote based on their neighbourhood
→ Voters will vote based on majority voters did for that particular party.
→ Tuning the K value:



Here, one voter will vote party 3

one ——— 1

3 ——— party 2

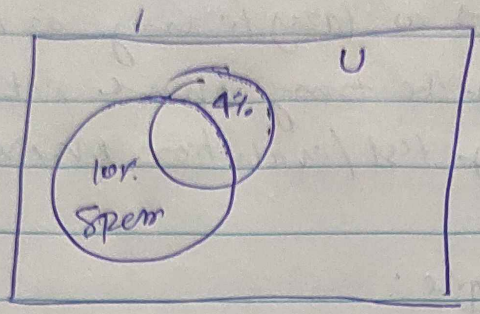
So, the ? can be vote for party 2 as they have majority

⑥ Joint probability.

↳ two or more events occurring together

↳ it is the prob. of intersection of two or more events

↳



4% → lottery

10% → spam

U → all emails

→ if email msg present in the world, which is likely to be lottery or spam.

→ This venn-diag. indicates the joint prob. of spam with lottery

→ It shows all spam msg contain the word lottery and not every email with the word lottery is spam

$$P(A \cap B) = P(A) * P(B)$$

$$P(\text{spam} \cap \text{lottery}) = 0.1 \times 0.04 = 0.004$$

⑦ Laplace estimator: adds a small no. to each of the counts in the freq. table

: this ensures that each feature has non-zero occurring prob.

: Usually Laplace estimator set to 1

eg $P(\text{Spam} \mid w_1 \cap \neg w_2 \cap \neg w_3) = \frac{3}{22} * \frac{11}{22} * \frac{0}{22} * \frac{22}{100} = 0$

using Laplace estimator for $w_3 = 1$

$$P(\text{Spam} \mid w_1 \cap \neg w_2 \cap \neg w_3) = \frac{3}{25} * \frac{11}{28} * \frac{1}{25} * \frac{22}{100} = 0$$

⑧ Curse of Dimensionality

↳ phenomenon where the performance of the ML algo reduces as no. of features in data increases

↳ As features ↑, amnt of data req. to maintain the same level grows exponentially.

↳ It is because, the no. of dimensions or increases, the volume of feature grows exponentially and data points become ~~increasingly~~ increasingly sparse.

↳ It affect many ML algo like clustering, classification and regression.

↳ It can lead to overfitting also.

~~↳ To resolve these problems~~

↳ To check these curse of dimensionality, we can vs 1D, 2D or 3D space also.