Module 2

Lazy Learning

Module 2

- **Lazy Learning - Classification Using k-Nearest Neighbor algorithm. Measuring similarity. Choice of k.**
-
- **Probabilistic Learning - Naive Bays' classifier. Review of probability - Joint probability, Conditional probability and Bay's theorem, Naive Bayes algorithm.**

Lazy Learning

- Lazy learning methods simply store the data and generalizing beyond these data is postponed until an explicit request is made.

The k-NN algorithm

- The nearest neighbors approach to classification is exemplified by the K-Nearest Neighbors algorithm (k-NN).
- one of the simplest machine learning algorithms
- Widely used.

k-NN algorithm

- The k-NN algorithm gets its name from the fact that it uses information about an example's k-nearest neighbors to classify unlabeled examples.
- *k* → a variable implying that any number of nearest neighbors could be used.
- After choosing *k*, the algorithm requires a training dataset made up of examples that have been classified into several categories, as labeled by a nominal variable.

k-NN algorithm (cntd..)

- Then, for each unlabeled record in the test dataset, k-NN identifies *k* records in the training data that are the "nearest" in similarity.
- The unlabeled test instance is assigned the class of the majority of the k nearest neighbors.

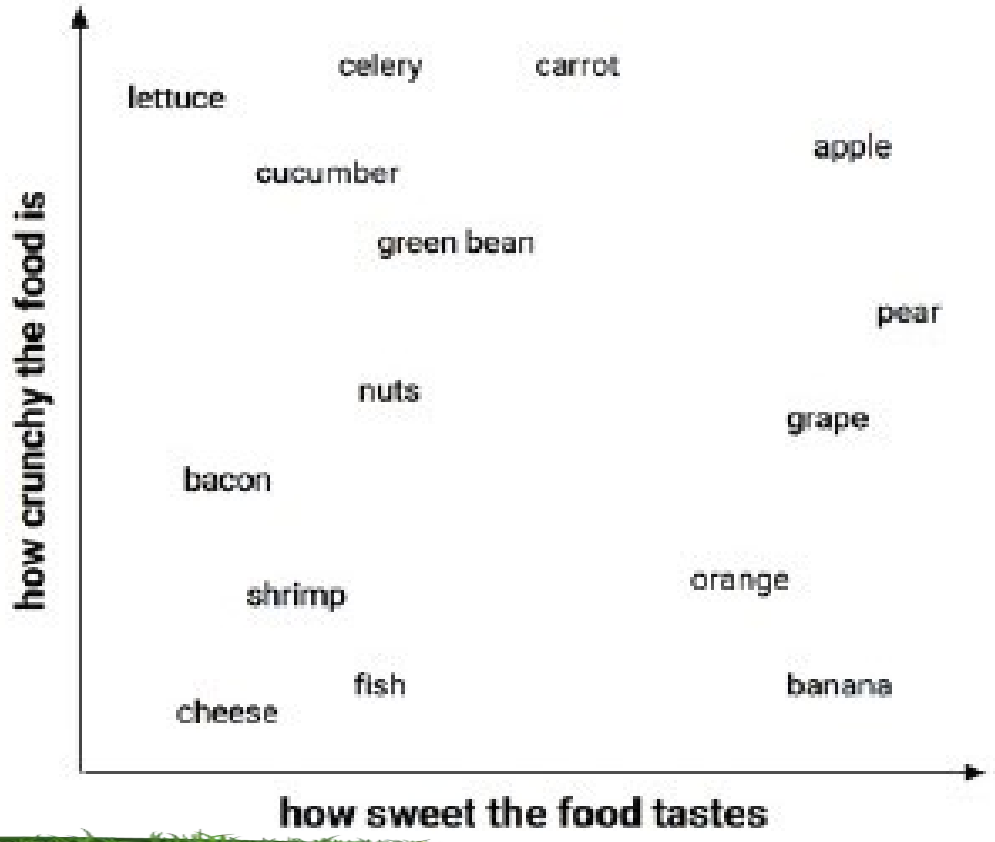
k-NN algorithm (cntd..)

- Eg;- Blind Tasting Experience.
- To keep things simple, we rated only two features of each ingredient.

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

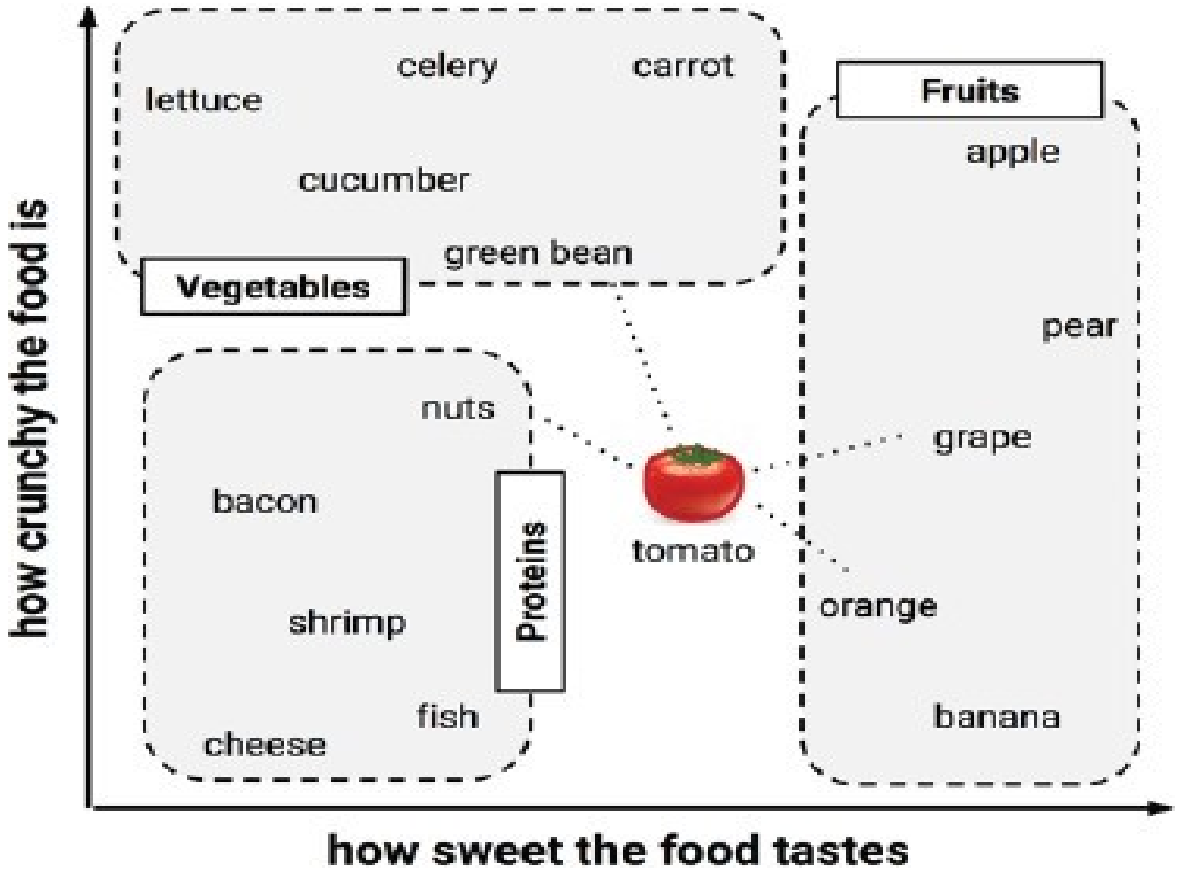
k-NN algorithm (cntd..)

- The k-NN algorithm treats the features as coordinates in a multidimensional feature space.
- As our dataset includes only two features, the feature space is two - dimensional.



k-NN algorithm (cntd..)

- After constructing this dataset, we decide to use it to settle the question:
  - Is tomato a fruit or vegetable?
- We can Use the Nearest Neighbor Approach to determine WHICH CLASS is a better fit, as shown in the following diagram:



Measuring similarity with distance

- Locating the tomato's nearest neighbors requires a distance function, or a formula that measures the similarity between the two instances.
- There are many different ways to calculate distance.
- k-NN algorithm uses Euclidean distance:
  - The distance one would measure if it were possible to use a ruler to connect two points.

Measuring similarity with distance (cntd.)

- **Euclidean Distance:**
- **p & q → examples to be compared, each having n - features.**
- **p1 → the value of the first feature of example p;**
- **q1 → the value of the first feature of example q:**

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

Euclidean Distance (cntd.):

- The distance formula involves comparing the values of each feature.
- Eg:- to calculate the distance between the tomato (sweetness = 6, crunchiness = 4), and the green bean (sweetness = 3, crunchiness = 7), we can use the formula as follows:

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

Measuring similarity with distance (cntd.)

- **calculate the distance between the tomato and several of its closest neighbors as follows:**

Ingredient	Sweetness	Crunchiness	Food type	Distance to the tomato
grape	8	5	fruit	*sqrt((6 - 8)^2 + (4 - 5)^2)* = 2.2
green bean	3	7	vegetable	*sqrt((6 - 3)^2 + (4 - 7)^2)* = 4.2
nuts	3	6	protein	*sqrt((6 - 3)^2 + (4 - 6)^2)* = 3.6
orange	7	3	fruit	*sqrt((6 - 7)^2 + (4 - 3)^2)* = 1.4

Measuring similarity with distance (cntd.)

1-NN classification:

- To classify the tomato as a vegetable, protein, or fruit, we'll begin by assigning the tomato, the food type of its single nearest neighbor.
- This is called 1-NN classification because k = 1.
- The orange is the nearest neighbor to the tomato, with a distance of 1.4.
- As orange is a fruit, the 1-NN algorithm would classify tomato as a fruit.

Measuring similarity with distance (cntd.)

- **3-NN Classification (k=3)**
- **If we use the k-NN algorithm with k = 3, it performs a vote among the three nearest neighbors:**
  - **orange, grape, and nuts.**
- **Since the majority Class among these neighbors is fruit (2 of the 3 votes), the tomato again is classified as a fruit.**

Choosing an appropriate k

- The decision of how many neighbors to use for k-NN determines how well the model will generalize to future data.
- ***Bias Variance Tradeoff:***
- The balance between overfitting and underfitting the training data is a problem known as bias variance tradeoff.
  - Overfitting → occurs when a statistical model or machinelearning algorithm captures the noise of the data.
  - Underfitting → refers to a model that can neither model the training data nor generalize to new data

Choosing an appropriate k (cntd..)

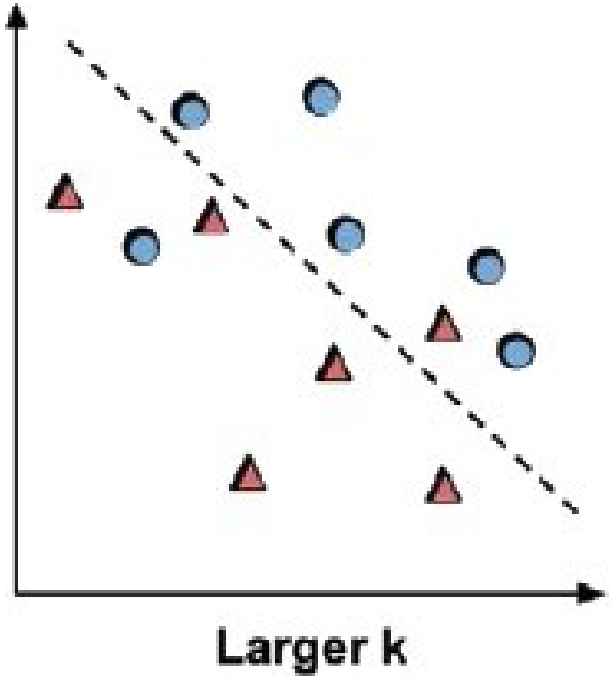
- **Choosing a large k:**
  - reduces the impact or variance caused by noisy data.
- **Very large k** (as large as the total number of observations in the training data).
  - The model would consequently always predict the majority class, regardless of the nearest neighbors.
- **using a single nearest neighbor:**
  - allows the noisy data or outliers to influence the classification of examples.

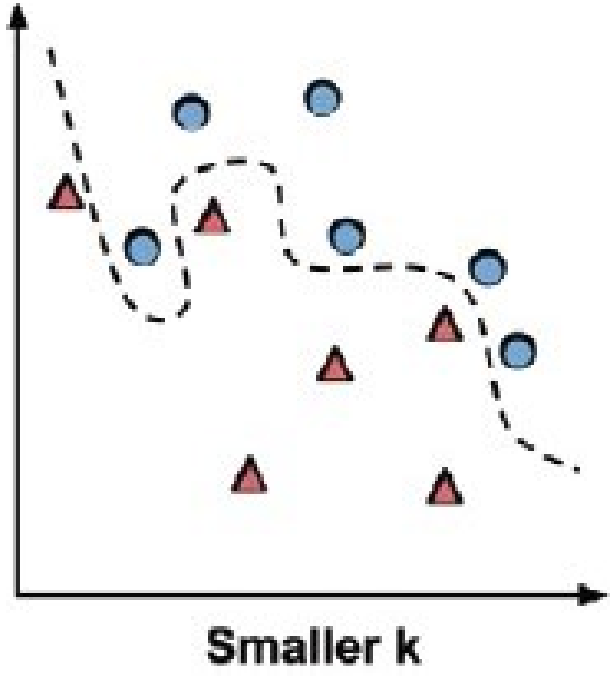
"these two are extremes

✓ The best k value → is somewhere between

Choosing an appropriate k (cntd..)

- **Smaller values allow more complex decision boundaries that more carefully fit the training data.**





Choosing an appropriate k (cntd..)

- **One common practice is:**
  - **To begin with k = square root of the number of training examples.**
  - **In the food classifier we developed previously, we might set k = 4 because there were 15 example ingredients in the training data and**
  - **square root of 15 = 3.87.**

Preparing data for use with k-NN

- **Features** are typically transformed to a **standard range** before applying the k-NN algorithm.
- **Distance formula** is highly **dependent** on how **features are measured**.
- Eg:- if certain features have a much **larger range of values than the others**, the distance measurements will be **strongly dominated** by the features with **larger ranges**.

Preparing data for use with k-NN

- The solution is to "rescale the features" by shrinking or expanding their range such that each one contributes relatively equally to the distance formula.
- For example, if sweetness and crunchiness are both measured on a scale from 1 to 10, we would also like spiciness to be measured on a scale from 1 to 10.

Preparing data for use with k-NN

- There are several methods for scaling.
  - min-max normalization.
  - z-score standardization.
  - dummy coding.

1. Min-max Normalization:

- **Traditional method of rescaling features for k-nn.**
- **This process transforms a feature such that all of its values fall in a range between 0 and 1.**
- **The formula for normalizing a feature is as follows:**

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

2. Z-score Standardization:

- **Another common transformation.**

$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

- **The resulting value is called a z-score.**
- **The z-scores fall in an unbound range of negative and positive numbers.**
- **they have no predefined minimum and maximum.**

3. Dummy Coding :

- **Euclidean distance formula is not defined for *nominal data*.**
- **Therefore, to calculate the distance between nominal features, we need to convert them into a numeric format.**
- **A typical solution utilizes dummy coding, where a value of 1 → one category, and 0 → the other category.**
- **For instance, dummy coding for a gender variable could be constructed as:**

$$\text{male} = \begin{cases} 1 & \text{if } x = \text{male} \\ 0 & \text{otherwise} \end{cases}$$

3. Dummy Coding : (cntd..)

- Dummy coding of the two-category (binary) gender variable results in a single new feature named male.
- There is no need to construct a separate feature for female;
- since the two genders are mutually exclusive, knowing one or the other is enough.

3. Dummy Coding : (cntd..)

- An *n*-category nominal feature can be dummy coded by creating the binary indicator variables for (*n - 1*) levels of the feature.
- For example, the dummy coding for a 3 - category temperature variable :
- Eg:- hot, medium, or cold could be set up as:
- *(3 - 1) = 2* features, as shown here:

$$\text{hot} = \begin{cases} 1 & \text{if x} = \text{hot} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{medium} = \begin{cases} 1 & \text{if x} = \text{medium} \\ 0 & \text{otherwise} \end{cases}$$

3. Dummy Coding : (cntd..)

- Knowing that hot and medium are both 0 is enough to know that the temperature is cold.
- We, therefore, do not need a third feature for the cold category.
- A convenient aspect of dummy coding is:
  - the distance between dummy coded features is always one or zero, and thus,
  - the values fall on the same scale as min-max normalized numeric data.
  - No additional transformation is necessary.

Why is the k-NN algorithm lazy?

- **No abstraction occurs.**
- **The abstraction and generalization processes are skipped altogether, and this undermines the definition of learning.**
- **process of making predictions tends to be relatively slow.**
- **lazy learning is also known as "*Instance-based Learning"* or '*Rote Learning'*.**

Strengths & Weaknesses Of KNN - Algorithm

Strengths	Weaknesses
• Simple and effective • Makes no assumptions about the underlying data distribution • Fast training phase	• Does not 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 and missing data require additional processing

Understanding Nearest Neighbor Classification.

- Nearest neighbor classifiers are defined by their characteristic of classifying unlabeled examples by assigning them the class of similar labeled examples.
- simple
- powerful.

Understanding Nearest Neighbor Classification.(cntd.)

used for:

- Computer vision applications (optical character recognition and facial recognition).
- Predicting whether a person will enjoy a movie or music .
- Identifying patterns in genetic data.

- **Probabilistic Learning**
  - **Naive Bays' classifier**

Probabilistic Learning

- EG:- Meteorologist – weather forecast- “70 % chance of rain “.
- Use data on past events to extrapolate future events.
- A 70% chance of rain → 7 out of the 10 past cases with similar conditions, precipitation(weather condition) occurred somewhere in the area.

Probabilistic Classifier

- A classifier that is able to predict, given a sample input, a probability distribution over a set of classes, (rather than only outputting the most likely class that the sample should belong to.)

Naive Bayes

- **Uses probabilities**
- **mathematician Thomas Bayes**
  - **developed foundational principles to describe the probability of events, &**
  - **how probabilities should be revised in the light of additional information.**
- **These principles formed the foundation for what are now known as *Bayesian methods.***

Naive Bayes (cntd..)

- probability → number between 0 and 1 (that is, between 0 % and 100 %)
  - which captures the chance that an event will occur in the light of the available evidence.
- lower probability → less likely the event is to occur.
- probability of 0 → event will definitely not occur
- probability of 1 → that the event will occur with 100 percent certainty.

Naive Bayes

- Classifiers based on Bayesian methods utilize training data to calculate an observed probability of each outcome based on the evidence provided by feature values.
- When the classifier is later applied to unlabeled data, it uses the observed probabilities to predict the most likely class for the new features.

Bayesian classifiers

used for:

- Text classification, such as junk e-mail (spam) filtering
- Intrusion/anomaly detection in computer networks
- Diagnosing medical conditions given a set of observed symptoms

Bayesian classifiers

- Best applied to problems in which the information from numerous attributes should be considered simultaneously in order to estimate the overall probability of an outcome.

❖ Many machine learning algorithms ignore features that have weak effects, Bayesian methods utilize all the available evidence to subtly change the predictions.

- If large number of features have relatively minor effects, taken together, their combined impact could be quite large.

Bayesian classifiers

- Bayesian probability theory is rooted in the idea that the " estimated likelihood of an event, or a potential outcome, should be based on the evidence across multiple trials, or opportunities for the event to occur. "

- **Events and trials for several real-world outcomes:**

Event	Trial
Heads result	Coin flip
Rainy weather	A single day
Message is spam	Incoming e-mail message
Candidate becomes president	Presidential election
Win the lottery	Lottery ticket

Understanding probability

- **The probability of an event is estimated from the observed data by dividing the number of trials in which the event occurred by the total number of trials.**
- **Probability = (Number of trials in which the event occurred / Total number of trials.)**

Understanding probability (cntd..)

- **Eg:- if it rained 3 out of 10 days with similar conditions as today, the probability of rain today = *3 / 10* = *0.30 %* or 30 percent.**
- **if 10 out of 50 prior email messages were spam, then the probability of any incoming message being spam = *10 / 50* = *0.20%* or 20 percent.**

Understanding probability (cntd..)

- *P(A)* → **probability of event *A*.**
- **Eg:-** ***P(rain) = 0.30*** **&**
  - ***P(spam) = 0.20.***

Understanding probability (cntd..)

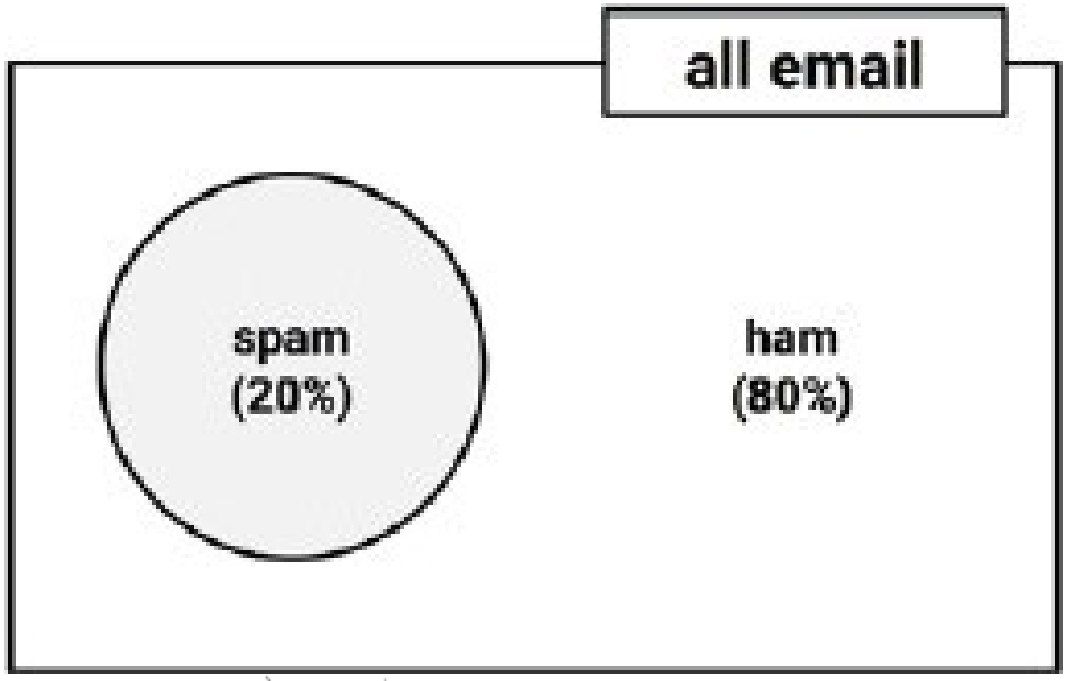
- Probability of all the possible outcomes of a trial must always sum to 1;
  - because a trial always results in some outcome happening.
- an event is always mutually exclusive and exhaustive with its complement.

Understanding probability (cntd..)

- The complement of event *A* is typically denoted A^c or A'.
- $P(\neg A)$ → probability of event *A* not occurring,
- Eg: - $P(spam) = 0.20$.
  - $P(\neg spam) = 0.80$.
- This notation is equivalent to $P(A^c)$.

Understanding probability (cntd..)

- **Eg:- events and their complements:**

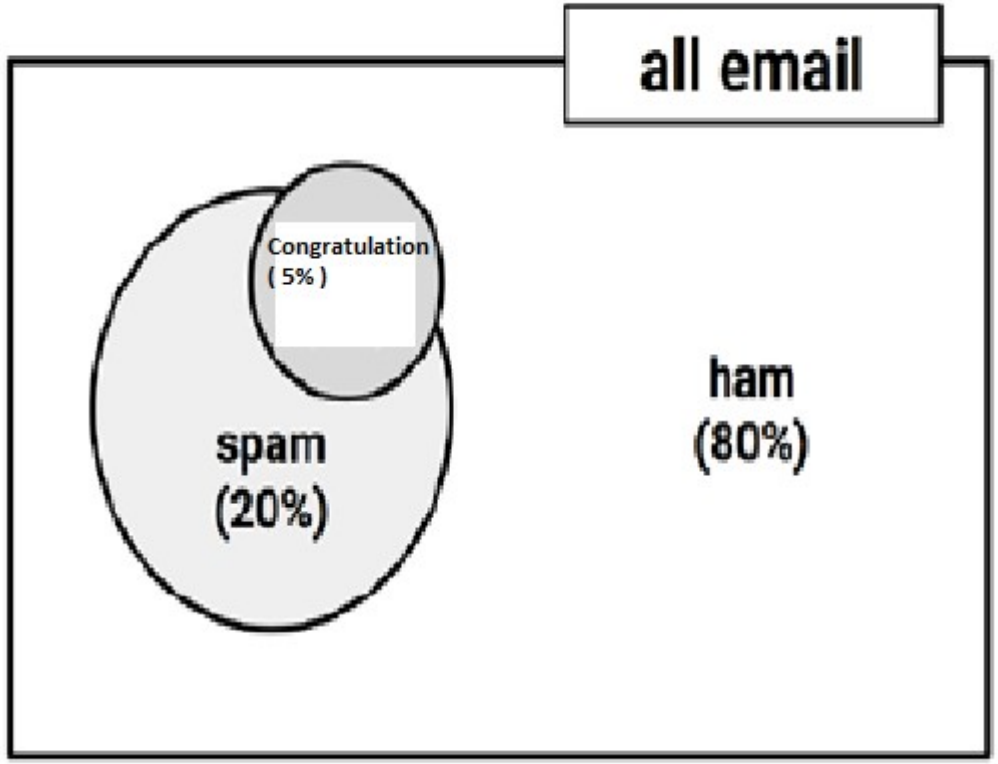


Joint Probability

- If certain events occur with the event of interest, we may be able to use them to make predictions.
- Eg:- consider a second event based on the outcome that an e-mail message contains the word "Congratulations".

– its presence in an incoming e-mail is therefore a very strong evidence that the message is spam.

Joint Probability (cntd..)



- **NOT ALL** spam messages contain the word " **Congratulations** " and **not every e-mail** with the word "**Congratulations** " is spam.

Joint Probability (cntd..)

Venn diagram.

- By John Venn
- diagram uses circles to illustrate the overlap between sets of items.
- it is used as a reminder to allocate probability to all possible combinations of events:

Joint Probability (cntd..)

Joint probability :

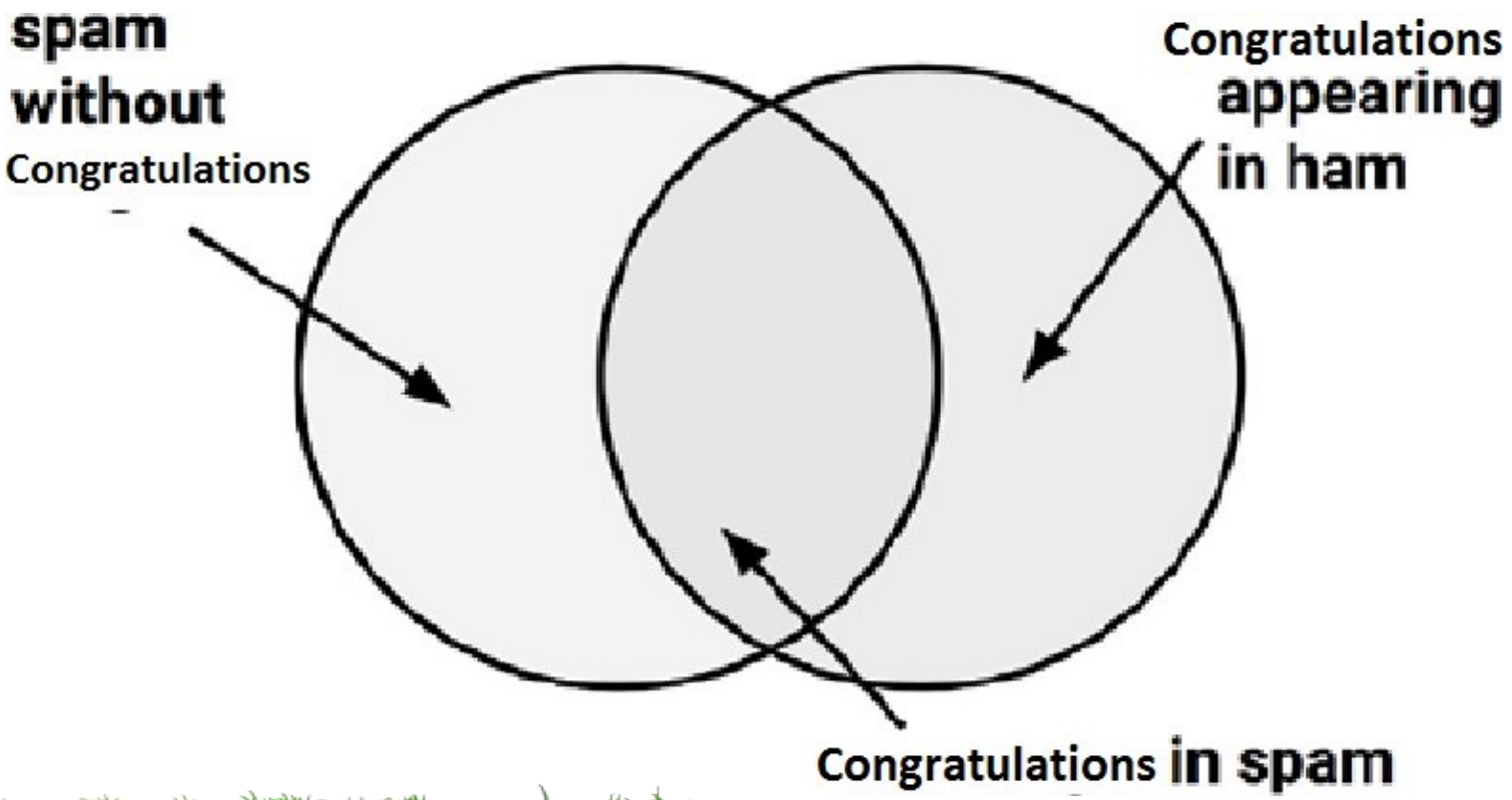
- **Probability of event Y occurring at the same time event X occurs.**
- **Notation for joint probability :**

$$P(X \cap Y) \text{ or } P(X, Y)$$

Which reads : the joint probability of X and Y.

Joint Probability (cntd..)

- **Venn Diagram**



Joint Probability (cntd..)

- To estimate the probability that both P(spam) and P(Congratulations) occur:

→P(spam ∩ Congratulations).

→Intersection of the two events;

→ Calculating P(spam ∩ Congratulations) depends on the joint probability of the two events or

→how the probability of one event is related to the probability of the other.

Joint Probability (cntd..)

Independent Events.

- **If the two events are totally unrelated, they are called independent events.**
  - **Outcome of one event does not provide any information about the outcome of the other.**
- **For independent events A and B, the probability of both happening can be expressed as:**
- **$P(A \cap B) = P(A) * P(B)$;**
  - **$P(A) = 20\% = 0.2$; $P(B) = 5\% = 0.05$;**
  - **$P(A \cap B) = 0.05 * 0.20 = 0.01$;**

Joint Probability (cntd..)

Dependent Events:

- Basis of predictive modeling.
- Eg:- presence of clouds is predictive of a rainy day.
- Appearance of the word Congratulations is predictive of a spam e-mail.

Joint Probability (cntd..)

- P(spam) and P(Congratulations) are likely to be highly dependent, which means that this calculation is incorrect.
- To obtain a reasonable estimate, we need to use a more careful formulation of the relationship between these two events, which is based on Advanced Bayesian Methods.

Conditional Probability - With Bayes' Theorem

- The relationships between dependent events can be described using Bayes' theorem.
- Estimateof probability of one event in light of the evidence provided by another event:

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

Conditional Probability - With Bayes' Theorem (cntd..)

Conditional probability:

- **Conditional probability of event A is the probability that the event will occur, given that event B already occurred.**
  - **P(A|B) → read as the "probability of event A, given that event B occurred".**
- **Since, probability of *A* is dependent (conditional) on what happened with event *B*.**

Conditional Probability - With Bayes' Theorem (cntd..)

- Bayes' theorem tells that estimate of $P(A|B)$ should be based on ;

– $P(A \cap B)$, a measure of how oftenA and B are observed to occur together, &

– $P(B)$, a measure of how often B is observed to occur in general.

Conditional Probability - With Bayes' Theorem (cntd..)

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

- ***P(A∩B) = P(A|B) * P(B),***
- **We know: *P(A∩B) = P(B∩A)***
- **So, P(A∩B) = P(B|A) * P(A), use this in Bayes theorem: we get;**

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

Conditional Probability - With Bayes' Theorem (cntd..)

Prior Probability:

- **The prior probability of an event is the probability of the event computed before the collection of new data.**
- **Eg:- Without knowledge of an incoming message's content, the best estimate of its spam status would be *P(spam)*, the probability that any prior message was spam, which we calculated previously to be 20 %.**

– ***P(spam)* = 20 %**

Conditional Probability - With Bayes' Theorem (cntd..)

Likelihood:

– You obtained additional evidence by looking more carefully at the set of previously received messages to examine the frequency that the term " Congratulations " appeared.

❖ The probability that the word Congratulations was used in previous spam messages, or

❖ P(Congratulations|spam), is called the likelihood.

- *Marginal Likelihood*:

– The probability that Congratulations appeared in any message at all, or

Conditional Probability - With Bayes' Theorem (cntd..)

Posterior Probability :

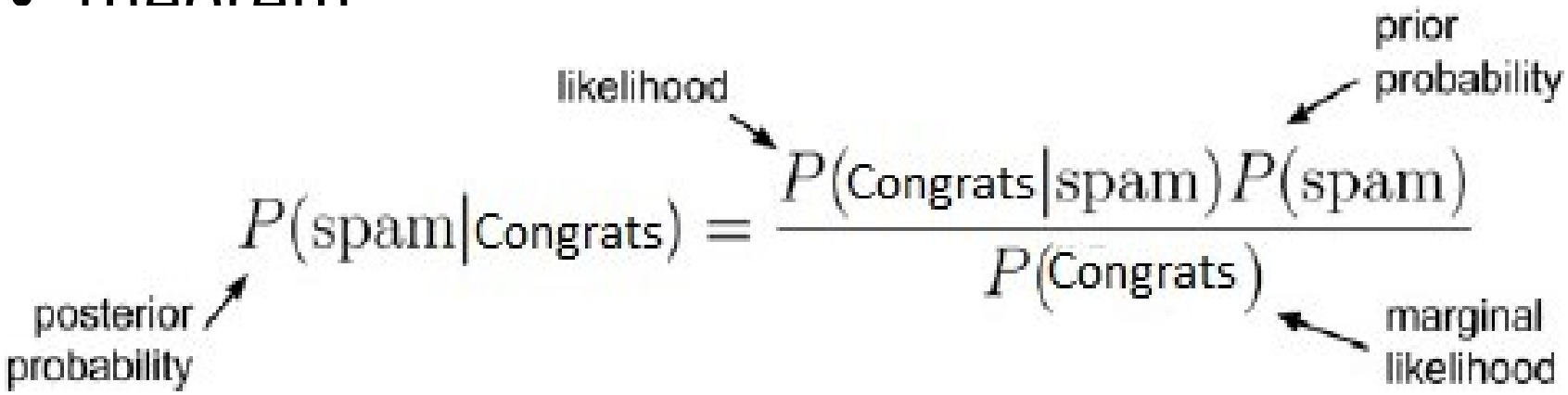
- Probability of event A occurring given that event B has occurred.
- Calculated by updating the prior probability by using Bayes' theorem.

- *Eg:-* we can compute a posterior probability that measures how likely the message is to be spam.
- *Eg:* *P(spam|congratulations)* → how likely the message is to be spam.

Conditional Probability - With Bayes' Theorem (cntd..)

Bayes' Theorem :

- theorem



- **Frequency Table;** Number of times **"congratulations"** appeared in spam and ham messages.

	Congratulations		
Frequency	**Yes**	**No**	**Total**
spam	4	16	20
ham	1	79	80
Total	5	95	100

- **Likelihood Table:**

	Congratulations		
Likelihood	**Yes**	**No**	**Total**
spam	4 / 20	16 / 20	20
ham	1 / 80	79 / 80	80
Total	5 / 100	95 / 100	100

Bayes' Theorem : (cntd.)

- likelihood table reveals that:
- P(congratulations=Yes|spam) = 4/20 = 0.20;
  - →probability is 20 percent that a message contains the term congratulations, given that the message is spam.
- We know, $P(A \cap B) = P(B|A)*P(A)$;
- So, $P(spam \cap congratulations)$

= P(congratulations|spam) * P(spam)

= (4/20) * (20/100) = 0.04.

- 4 out of the 100 messageswere spam with the term "congratulations".

Bayes' Theorem : (cntd.)

?? To compute the posterior probability, P(spam| congratulations):

- P(spam| congratulations):
- =P(congratulations|spam)*P(spam)/P(congratul ations)
- =(4/20)*(20/100) / (5/100) = 0.80.

– probability is 80 % that a message is spam, given that it contains the word congratulations.

- In light of this result, any message containing this term should

The Naive Bayes algorithm

- Describes a simple method to apply Bayes' theorem to classification problems.
- Most common machine learning method that utilizes Bayesian methods.
- Particularly true text classification

Strengths and Weaknesses:

Strengths	Weaknesses
• Simple, fast, and very effective • Does well with noisy and missing data • Requires relatively few examples for training, but also works well with very large numbers of examples • Easy to obtain the estimated probability for a prediction	• Relies on an often-faulty assumption of equally important and independent features • Not ideal for datasets with many numeric features • Estimated probabilities are less reliable than the predicted classes

The Naive Bayes algorithm (cntd..)

- Naive Bayes algorithm is named as such because it makes some "naive“ assumptions about the data.
- In particular, Naive Bayes assumes that all of the features in the dataset are equally important and independent.
- These assumptions are rarely true in most real-world applications.

The Naive Bayes algorithm (cntd..)

- To identify spam by monitoring e-mail messages:
  - It is almost certainly true that some features will be more important than others.
  - Eg:-, the e-mail sender may be a more important indicator of spam than the message text.
  - Additionally, the words in the message body are not independent from one another, since the appearance of some words is a very good indication that other words are also likely to appear.
  - A message with the word "congratulations" will probably also contain the words "earn" or "subscribe".

The Naive Bayes algorithm (cntd..)

- Due to the algorithm's versatility and accuracy across many types of conditions, Naive Bayes is often a strong first candidate for classification learning tasks.

Classification with Naive Bayes

- **Spam filter - by adding a few additional terms to be monitored in addition to the term congratulations(w1): earn(w2), subscribe(w3) and money(w4).**
- ***Frequency Table:***

	congrats (W_1)		earn (W_2)		subscribe (W_3)		Money (W_4)		
Likelihood	Yes	No	Yes	No	Yes	No	Yes	No	Total
spam	4 / 20	16 / 20	10 / 20	10 / 20	0 / 20	20 / 20	12 / 20	8 / 20	20
ham	1 / 80	79 / 80	14 / 80	66 / 80	8 / 80	71 / 80	23 / 80	57 / 80	80
Total	5 / 100	95 / 100	24 / 100	76 / 100	8 / 100	91 / 100	35 / 100	65 / 100	100

Classification with Naive Bayes(cntd..)

- Eg: - suppose that a message contains the terms congratulations and Money, but does not contain either earn or subscribe.

Classification with Naive Bayes(cntd..)

- **Using Bayes' theorem, we can define the problem as shown in the following formula.**

$$P(\text{spam}|W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4) = \frac{P(W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4|\text{spam})P(\text{spam})}{P(W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4)}$$

- **It captures the probability that a message is spam, given that congratulations = *Yes*, *earn = No*, *subscribe = No*, and Money= *Yes*:**
- **formula → difficult to solve.**

Classification with Naive Bayes(cntd..)

- **The work becomes much easier if we can exploit the fact that Naive Bayes assumes independence among events.**
- **Specifically, it assumes class-conditional independence;**
- **Class-conditional Independence:**

– **means that events are independent so long as they are conditioned on the same class value.**

Classification with Naive Bayes(cntd..)

- Assuming conditional independence allows us to simplify the formula using the probability rule for independent events, which states that $P(A \cap B) = P(A) * P(B)$.

Classification with Naive Bayes(cntd..)

- **So, the conditional probability of spam can be expressed as:**

$$P(\text{spam}|W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4) \propto P(W_1|\text{spam})P(\neg W_2|\text{spam})P(\neg W_3|\text{spam})P(W_4|\text{spam})P(\text{spam})$$

- **Probability that the message is ham can be expressed as:**

$$P(\text{ham}|W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4) \propto P(W_1|\text{ham})P(\neg W_2|\text{ham})P(\neg W_3|\text{ham})P(W_4|\text{ham})P(\text{ham})$$

Classification with Naive Bayes(cntd..)

- ****** Equals symbol(=) has been replaced by the "proportional-to" symbol to indicate the fact that the denominator has been omitted.**

Classification with Naive Bayes(cntd..)

- Overall likelihood of spam is :

  = (4/20) * (10/20) * (20/20) * (12/20) * (20/100)

  = 0.012;

- Likelihood of ham is:

  = (1/80) * (66/80) * (71/80) * (23/80) * (80/100)

  = 0.002

Classification with Naive Bayes(cntd..)

- Because, 0.012/0.002 = 6, we can say that this message is six times more likely to be spam than ham.
- However, to convert these numbers into probabilities:

– we need to perform one last step to reintroduce the denominator that had been excluded.

Classification with Naive Bayes(cntd..)

- So, rescale the likelihood of each outcome by dividing it by the total likelihood across all possible outcomes.

Classification with Naive Bayes(cntd..)

- **Probability of spam**

= likelihood that the message is spam / likelihood that the message is either spam or ham:

- **Probability(spam)= likelihood(spam)/likelihood(spam+ham);**
  - **0.012/(0.012 + 0.002) = 0.857;**
  - **Probability(spam)= 85.7 %**
- **→ ie, we expect that the message is spam with 85.7 % probability**

- **Probability(ham) = likelihood that the message is ham / likelihood that the message is either spam or ham:**
- **0.002/(0.012 + 0.002) = 0.143;**
- **Probability(ham)= 14.3%.**
- **Since, these are mutually exclusive and exhaustive events, the sum of probabilities = 1.**

- **Naive Bayes classification algorithm can be summarized by the following formula:**

$$P(C_L|F_1, \ldots, F_n) = \frac{1}{Z} p(C_L) \prod_{i=1}^{n} p(F_i|C_L)$$

- **C_L → Level for class C,**
- **F_1,..F_n → features F1 through Fn,**
- **1/Z → scaling factor(which converts the likelihood values into probabilities):**

The Laplace estimator

- Before we employ Naive Bayes in more complex problems, there are some nuances (special cases) to consider.
- Suppose that we received another message, this time containing all 4- terms:
  - congratulations(w1), earn(w2), subscribe(w3) and money(w4).

The Laplace estimator(cntd..)

- Using the Naive Bayes algorithm as before, we can compute:
- Likelihood of spam :

= (4/20) * (10/20) * (0/20) * (12/20) * (20/100)

= 0

- Likelihood of ham :

= (1/80) * (14/80) * (8/80) * (23/80) * (80/100)

= 0.00005

➢ Therefore, the probability of spam :

= 0/(0 + 0.00005)

= 0

The Laplace estimator(cntd..)

- **Probability of ham :**

= 0.00005/(0 + 0. 0.00005)

= 1

- **These results suggest that the message is spam with 0 % probability &**
- **ham with 100 % probability.**

The Laplace estimator(cntd..)

- *P(spam|subscribe) = 0%*.
- Because probabilities in the Naive Bayes formula are multiplied in a chain, this 0 % value causes the posterior probability of spam to be zero, giving the word *subscribe* the ability to effectively nullify and reject all of the other evidence.
- Even if the e-mail was otherwise expected to be spam, the absence of the word *subscribe* in spam will always affect the other evidence , and
- result in the probability of spam = 0.

The Laplace estimator(cntd..)

- A solution to this problem involves using Laplace Estimator:

Laplace Estimator:

– Named after the French mathematician Pierre-Simon Laplace.

- The Laplace estimator essentially adds a small number to each of the counts in the frequency table, which ensures that each feature has a nonzero probability of occurring with each class.
- Typically, the Laplace estimator is set to 1, which ensures that each class-feature combination is found in the data at least once.

The Laplace estimator(cntd..)

- Using a Laplace value = 1;
  - We add one to each numerator in the likelihood function.
  - The total number of 1 values must also be added to each conditional probability denominator.
- So , Likelihood of spam:
- = (5/24) * (11/24) * (1/24) * (13/24) * (20/100)
- = 0.0004 . &
- Likelihood of ham :

= (2/84) * (15/84) * (9/84) * (24/84) * (80/100)

The Laplace estimator(cntd..)

- Ie,probability of spam = 80%; and
- probability of ham = 20 %;
  - which is a more acceptable result than the one obtained when the term *subscribe* alone determined the result.

Using numeric features with Naive Bayes.

- Using numeric features with Naive Bayes Because Naive Bayes uses frequency tables to learn the data, each feature must be categorical in order to create the combinations of class and feature values comprising of the matrix.
- Since, numeric features do not have categories of values, the preceding algorithm does not work directly with numeric data.

Using numeric features with Naive Bayes(cntd..)

- One easy and effective solution is to discretize numeric features.

Discretization / Binning.

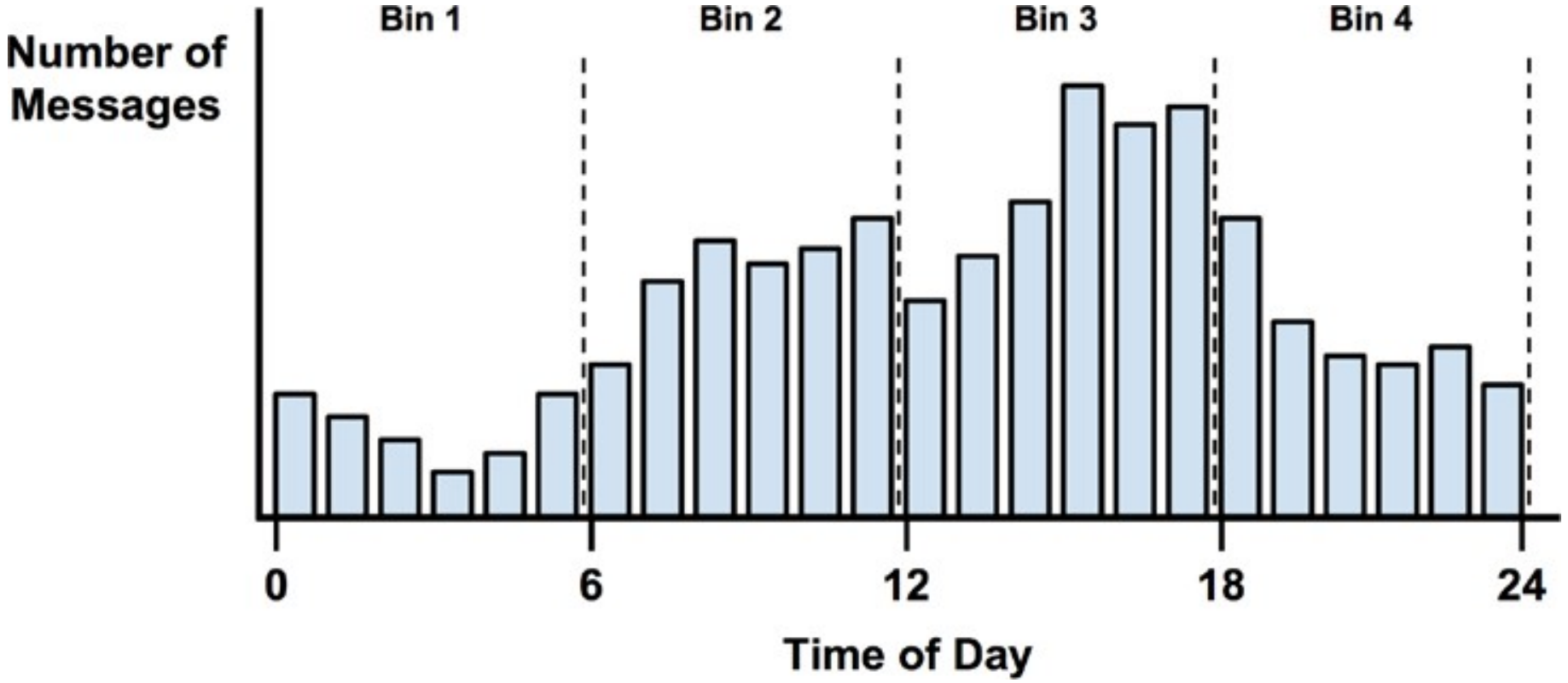
- numbers are put into categories known as bins.
- discretization is also called binning.
- This method is ideal when there are large amounts of training data, a common condition while working with Naive Bayes.

Using numeric features with Naive Bayes(cntd..)

- Several different ways to discretize a numeric feature.
- most common is to explore the data for natural categories or cut points in the distribution of data.
- Eg:- suppose that you added a feature to the spam dataset that recorded the time of night or day the e-mail was sent, from 0 to 24 hours.

Using numeric features with Naive Bayes(cntd..)

- **Time data Depicted using a histogram like the following diagram.**



Using numeric features with Naive Bayes(cntd..)

- In the early hours of morning →message frequency is low.
- The activity picks up → during business hours and tapers off → evening.
- It create 4 natural bins of activity, as partitioned by the dashed lines indicating places where the numeric data are divided into levels of a new nominal feature, which could then be used with Naive Bayes:

Choice of four bins:

- arbitrary
- based on the natural distribution of data and a
- Assumption about how the proportion of spam might change throughout the day.
  - We might expect that spammers operate in the late hours of the night or they may operate during the day, when people are likely to check their e-mail.
  - This said, to capture these trends, we could have just as easily used 3- bins or twelve.

❖ NOTE:

- Too few bins → can result in important trends being obscured.
- Too many bins → can result in small counts in the Naive Bayes frequency table;
  - Which can increase the algorithm's sensitivity to noisy data.