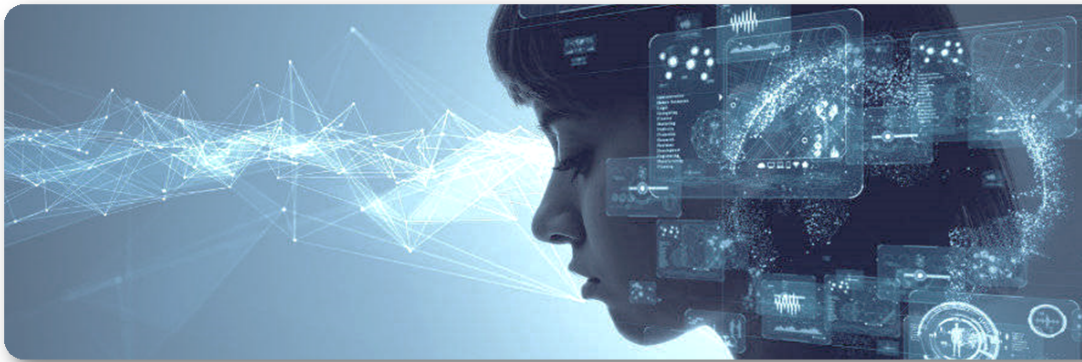


COMP 2211 Exploring Artificial Intelligence

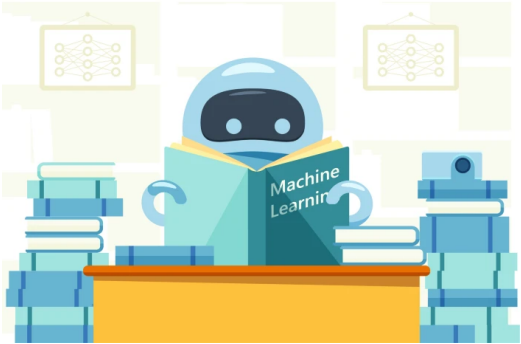
Naïve Bayes Classifier

Dr. Desmond Tsoi

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology, Hong Kong SAR, China

Part I

Fundamentals of Machine Learning

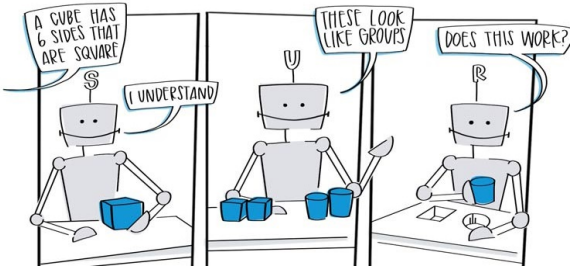


What is Machine Learning?

Definition

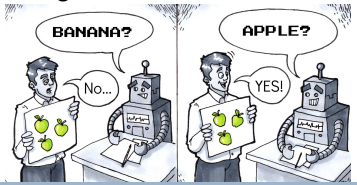
Machine Learning is the science and engineering of getting computers to act without being explicitly programmed.

- Types:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning (More details will be covered in class later)



Supervised Learning

- We form a training dataset consisting of examples with inputs and correct outputs (we called these labels).
  – This training dataset is called a labelled dataset
- The labelled dataset is then used to train a computer/machine to yield the desired output, i.e., allowing the machine to learn the relationship/mapping between the inputs and outputs.
- Supervised learning can be further divided into two types:
  1. Classification: Classify the data into categories (also called classes – represented using different labels).
     For example, yes or no, male or female, true or false, etc.
  2. Regression: Give a real or continuous value as output.
     For example, salary based on work experience, weight based on height, etc.



An Example Demonstrating the Idea of Supervised Learning

- Suppose we want to train our computer/machine to do maths.
- Instead of explicitly programming it, we pass a set of input data and the corresponding answers to the computer/machine and then ask it to learn/figure out the relationship between the data and the answers.

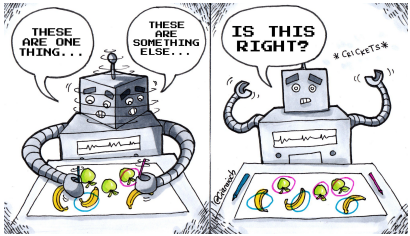
1.	2	4	5	=	3	5.	6	2	2	=	10
2.	5	2	8	=	2	6.	3	1	1	=	2
3.	2	2	1	=	3	7.	5	3	4	=	11
4.	4	2	2	=	6	8.	1	8	1	=	7

Real-Life Applications of Supervised Learning

- Image classification
  - E.g., Facebook can recognize your friend in a picture from an album of tagged photos.
- Visual recognition
  - E.g., The ability of a machine to identify objects, places, people, actions, and images.
- Signature recognition
  - E.g., Automatic signature recognition for automating bank cheques processing.
- Spam detection
  - E.g., Google has now advanced to the point that it can detect and filter out spam and phishing emails with about 99.9% accuracy.
- Weather forecasting
  - E.g., HK Observatory incorporates machine learning in strengthening the techniques in monitoring and forecasting.

Unsupervised Learning

- We use an unlabelled dataset (i.e., without giving the correct outputs) and train the machines to learn on themselves.
- The computer/machine tries to find a pattern in the unlabeled data and respond.
- Unsupervised learning can be further divided into two groups:
  1. Clustering: Group data into clusters that are similar between them and are dissimilar to the objects belonging to another cluster. For example, finding out which customers made similar product purchases.
  2. Association: A rule-based machine learning to discover the probability of the co-occurrence of items in a collection. For example, finding out which products were purchased together.



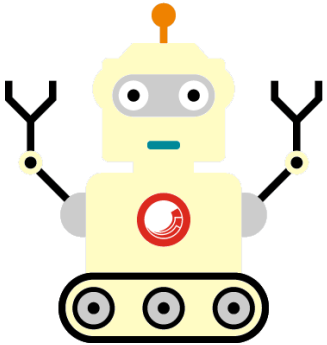
An Example Demonstrating the Idea of Unsupervised Learning

- Once again, suppose we want to train our computer/machine to do maths.
- Instead of explicitly programming it, we pass a set of input data, but NOT the corresponding answers, to the computer/machine and then ask it to learn/figure out the patterns.

1.	2	4	5	=	5.	6	2	2	=
2.	5	2	8	=	6.	3	1	1	=
3.	2	2	1	=	7.	5	3	4	=
4.	4	2	2	=	8.	1	8	1	=

Real-Life Applications of Unsupervised Learning

- Semantic Clustering
  - Semantically similar words share a similar context
  - Grouping images/videos based on similarities
  - Grouping audios based on acoustic similarity
- Identifying Accident Prone Areas
  - Used to identify accident-prone areas and introduce safety measures based on the intensity of those accidents.



Comparison between Supervised and Unsupervised Learning

	Supervised Learning	Unsupervised Learning
Training data	Labeled data Need domain expert to label data	Unlabeled data
Categories	Classification and regression	Clustering and association
Preference	Calculate outcomes	Discover underlying patterns
Training time	Shorter	Longer
Accuracy	High	Low
Optimal strategy	Depend on the data and learning algorithm	Depend on the data
Algorithms	Naïve Bayes, K-Nearest Neighbour, Decision Tree, Linear Regression, Logistics Regression, Support Vector Machine	K-Means Clustering, Hierarchical Clustering, Apriori Algorithm

Problem

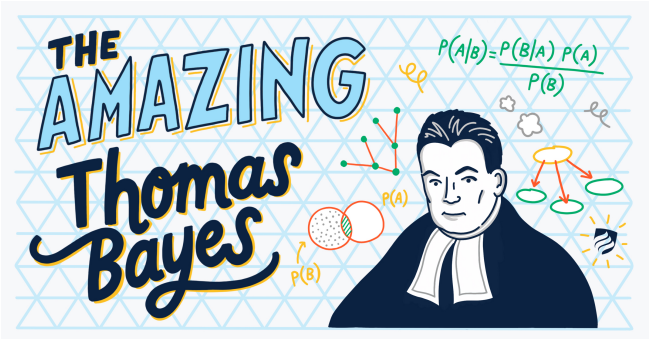
Suppose we have a dataset consisting of symptoms and the results showing the patients suffering from disease 'Z' or not.

Data	Blood Pressure	Fever	Diabetes	Vomit	Suffering from disease Z
1	High	High	Yes	No	No
2	High	High	Yes	Yes	No
3	Low	High	Yes	No	Yes
4	Normal	Mild	Yes	No	Yes
5	Normal	No fever	No	No	Yes
6	Normal	No fever	No	Yes	No
7	Low	No fever	No	Yes	Yes
8	High	Mild	Yes	No	No
9	High	No fever	No	No	Yes
10	Normal	Mild	No	No	Yes
11	High	Mild	No	Yes	Yes
12	Low	Mild	Yes	Yes	Yes
13	Low	High	No	No	Yes
14	Normal	Mild	Yes	Yes	No

Can we estimate whether a person suffering from disease 'Z' based on his symptoms (Blood Pressure = high, Fever = no, Diabetes = yes, and Vomit = yes) from the above table? Yes, use Bayes' rule.

Part II

Bayes' Rule



Conditional Probability

- Conditional probability denoted by $P(B|A)$ means what the probability that an event B occurs, given that event A has already occurred.
- Example:

Transport	Car	Foot	Bicycle	Total
Boys	200	330	120	650
Girls	250	320	80	650
Total	450	650	200	1300

  - What is the probability $P(Car|Boys)$?
    Answer: $200/650 = 0.31$
  - What is the probability $P(Bicycle|Girls)$
    Answer: $80/650 = 0.12$

Bayes' Rule

Bayes' rule is named after 18th-century British mathematician Thomas Bayes. It is the most important rule in data science. It is the mathematical rule that describes how to update a belief, given some evidence.

$$P(B|E) = P(B) \times \frac{P(E|B)}{P(E)}$$

- E: Evidence (also called Feature)
- B: Belief (also called Class)
- $P(B|E)$: Posterior probability (How often belief happens given evidence happens)
- $P(B)$: Prior probability (How likely belief is on its own)
- $P(E|B)$: Likelihood (How often evidence happens given that belief happens)
- $P(E)$: Marginal probability (How likely evidence is on its own)

Example

- E: Smoke, B: Fire
- P(B): how often there is fire
- P(E): how often there is smoke
- P(B|E): how often there is fire when we can see smoke
- P(E|B): how often there is smoke when there is fire
- So, Bayes' rule is a kind of formula telling us P(Fire|Smoke) when we know P(Smoke|Fire)

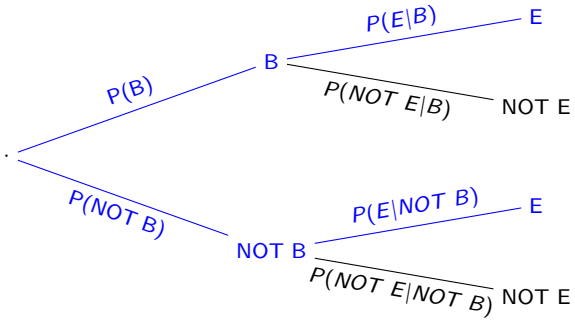
- Dangerous fires are rare (1%): $P(B) = P(\text{Fire}) = 0.01$
- But smoke is fairly common (10%) due to BBQ: $P(E) = P(\text{Smoke}) = 0.1$
- And 90% of dangerous fires make smoke: $P(E|B) = P(\text{Smoke}|\text{Fire}) = 0.9$

Probability of dangerous Fire when there is smoke

$$P(B|E) = P(Fire|Smoke) = \frac{P(Fire)P(Smoke|Fire)}{P(Smoke)} = \frac{(0.01)(0.9)}{0.1} = 0.09$$

How to Understand Bayes' Rule?

- We can represent the relationship between E and B as a tree.



$$P(E) = P(E|B)P(B) + P(E|Not\ B)P(Not\ B)$$

- It is okay if we know $P(E|B)$, $P(B)$, $P(E|Not\ B)$, $P(Not\ B)$. So, we have

$$P(B|E) = \frac{P(B)P(E|B)}{P(E|B)P(B) + P(E|Not\ B)P(Not\ B)} = \frac{P(B)P(E|B)}{P(E)}$$

Another Example: Virus Detection

- A doctor gives a patient a test for a particular virus.
- It is believed that 0.1% of the population has this virus.
- Based on experience, the doctor knows that, in 99% of the cases in which the virus is present, the test is positive; and in 95% of the cases in which the virus is not present, the test is negative.
- Question: If the test turns out to be positive, what is the chance that the virus is present?

Can we apply Bayes' rule to solve this problem? Let's try.

- Let V be the part of the population carrying the virus
- Let H be the remaining part

$$P(V) = 0.001, P(H) = 1 - 0.001 = 0.999$$
$$P(+|V) = 0.99, P(-|H) = 0.95$$
$$P(V|+) = ?$$

Another Example: Virus Detection

- According to Bayes' rule

$$P(V|+) = \frac{P(V)P(+|V)}{P(+)}$$

What is $P(+)$?

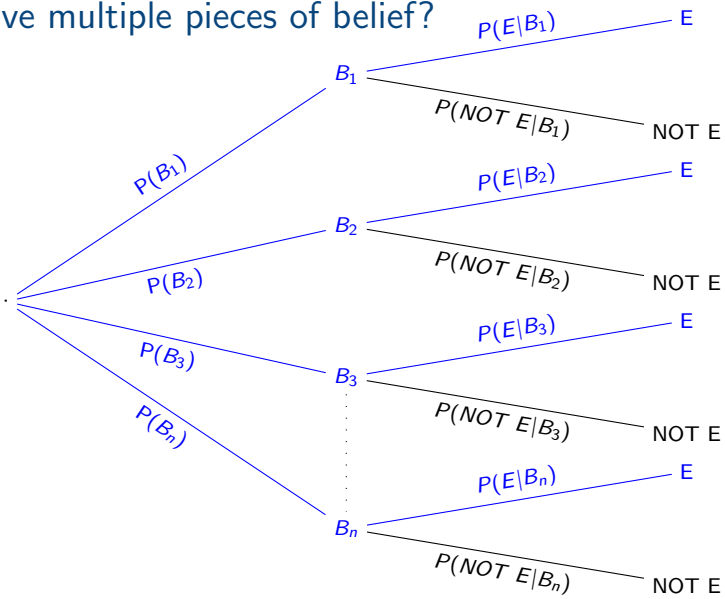
- In fact,

$$\begin{aligned} P(+) =& P(+|V)P(V) + P(+|H)P(H) \\ =& (0.99)(0.001) + (1 - 0.95)(0.999) \\ =& (0.99)(0.001) + (0.05)(0.999) \\ =& 0.05094 \end{aligned}$$

- So,

$$\begin{aligned} P(V|+) &= \frac{(0.001)(0.99)}{0.05094} \\ &\approx 0.02 \end{aligned}$$

What if we have multiple pieces of belief?



General Form of Bayes' Rule

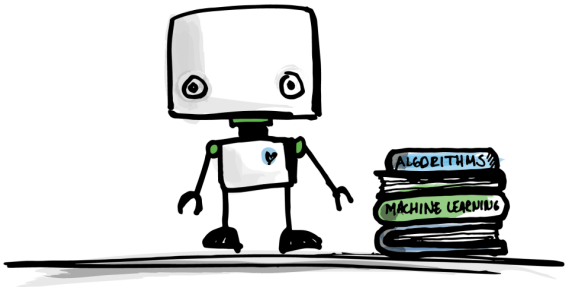
$$P(B_i|E) = \frac{P(B_i)P(E|B_i)}{P(E|B_1)P(B_1) + P(E|B_2)P(B_2) + P(E|B_3)P(B_3) + \cdots + P(E|B_n)P(B_n)}$$
$$= \frac{P(B_i)P(E|B_i)}{\sum_{j=1}^{n} P(E|B_j)P(B_j)}$$

- Each belief, B_i, is also called a class. For instance, V (population carrying the virus) and H (the population that does not carry the virus) are two different classes in the virus detection example.

What if we have multiple pieces of evidence also?

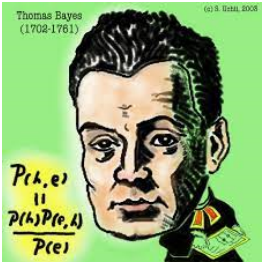
- Say, we have d pieces of evidence $E = \{e_1, e_2, e_3, \ldots, e_d\}$

$$P(B_i|E) = \frac{P(B_i)P(E|B_i)}{P(E|B_1)P(B_1) + P(E|B_2)P(B_2) + P(E|B_3)P(B_3) + \cdots + P(E|B_n)P(B_n)}$$
$$= \frac{P(B_i)P((e_1, e_2, e_3, \ldots, e_d)|B_i)}{\sum_{j=1}^{n} P((e_1, e_2, e_3, \ldots, e_d)|B_j)P(B_j)}$$



Part III

Naïve Bayes



Naïve Bayes

- The Bayes' rule is called "Naïve Bayes" if we assume each evidence makes an independent and equal contribution to the belief.
  - Independent:
    For example, the temperature being "Hot" has nothing to do with the humidity being "high" or outlook being "Rainy" does not affect the winds. Hence the features are assumed to be independent.
  - Equal contribution:
    For example, knowing only temperature and humidity alone cannot predict the outcome accurately. Each evidence contributes equally to the outcome.

The assumption made by Naïve Bayes are NOT generally correct in real-world situations. In fact, the independence assumption is never correct but often works well in practice.

Naïve Bayes

- With the independent and equal contribution to the belief, Bayes' rule can be transformed to the following.

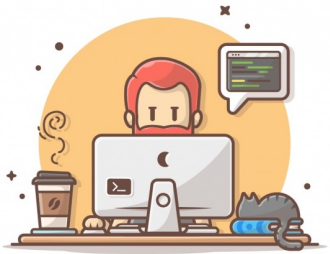
$$
\begin{aligned}
P(B_i|E) &= \frac{P(B_i)P((e_1, e_2, e_3, \ldots, e_d)|B_i)}{\sum_{j=1}^{n} P((e_1, e_2, e_3, \ldots, e_d)|B_j)P(B_j)} \\
&= \frac{P(B_i)P(e_1|B_i)P(e_2|B_i)P(e_3|B_i)\ldots P(e_d|B_i)}{\sum_{j=1}^{n} P(e_1|B_j)P(e_2|B_j)P(e_3|B_j)\ldots P(e_d|B_j)P(B_j)}
\end{aligned}
$$

Naïve Bayes

- Suppose we further assume that we are only interested in knowing which brief has the highest probability value rather than the actual probability value. In that case, we then we can further simplify the equation by removing the denominator as follows:

$$B_{NB} = argmax_{B_i} P(B_i)(P(e_1|B_i)P(e_2|B_i)P(e_3|B_i) \cdots P(e_d|B_i))$$

- It is because the denominator remains constant for a given input.



Predicting Suffering from Disease 'Z' using Naïve Bayes Classifier

Data	Blood Pressure	Fever	Diabetes	Vomit	Suffering from disease Z
1	High	High	Yes	No	No
2	High	High	Yes	Yes	No
3	Low	High	Yes	No	Yes
4	Normal	Mild	Yes	No	Yes
5	Normal	No fever	No	No	Yes
6	Normal	No fever	No	Yes	No
7	Low	No fever	No	Yes	Yes
8	High	Mild	Yes	No	No
9	High	No fever	No	No	Yes
10	Normal	Mild	No	No	Yes
11	High	Mild	No	Yes	Yes
12	Low	Mild	Yes	Yes	Yes
13	Low	High	No	No	Yes
14	Normal	Mild	Yes	Yes	No

For a patient with symptoms (High, No, Yes, Yes), what is the probability he suffers from disease 'Z'?

Predicting Suffering from Disease 'Z' using Naïve Bayes Classifier

Counts:

Blood Pressure			Fever			Diabetes			Vomit			Disease 'Z'	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
High	2	3	High	2	2	Yes	3	4	Yes	3	3	9	5
Normal	3	2	Mild	4	2	No	6	1	No	6	2		
Low	4	0	No Fever	3	1								

Relative Frequency:

Blood Pressure			Fever			Diabetes			Vomit			Disease 'Z'	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
High	2/9	3/5	High	2/9	2/5	Yes	3/9	4/5	Yes	3/9	3/5	9/14	5/14
Normal	3/9	2/5	Mild	4/9	2/5	No	6/9	1/5	No	6/9	2/5		
Low	4/9	0/5	No Fever	3/9	1/5								

Predicting Suffering from Disease 'Z' using Naïve Bayes Classifier

- Given the naïve bayes' formula

$$B_{NB} = argmax_{B_i} P(B_i)(P(e_1|B_i)P(e_2|B_i)P(e_3|B_i)\cdots P(e_d|B_i))$$

we have

$$P(DiseaseZ = Yes)(P(BloodPressure|Yes)P(Fever|Yes)P(Diabetes|Yes)P(Vomit|Yes))$$
$$P(DiseaseZ = No)(P(BloodPressure|No)P(Fever|No)P(Diabetes|No)P(Vomit|No))$$

- Now, (Blood Pressure, Fever, Diabetes, Vomit) = (High, No, Yes, Yes)

$$P(DiseaseZ = Yes)(P(BP = High|Yes)P(Fever = No|Yes)P(Diabetes = Yes|Yes)P(Vomit = Yes|Yes))$$
$$= \left(\frac{9}{14}\right)\left(\frac{2}{9}\right)\left(\frac{3}{9}\right)\left(\frac{3}{9}\right)\left(\frac{3}{9}\right) = 0.005291$$
$$P(DiseaseZ = No)(P(BP = High|No)P(Fever = No|No)P(Diabetes = Yes|No)P(Vomit = Yes|No))$$
$$= \left(\frac{5}{14}\right)\left(\frac{3}{5}\right)\left(\frac{1}{5}\right)\left(\frac{4}{5}\right)\left(\frac{3}{5}\right) = 0.020571$$

- Result: The patient is not suffering from Disease 'Z'. Since $0.020571 > 0.005291$.
- After normalization, we have $0.020571/(0.020571 + 0.005291) = 0.795414121$, $0.005291/(0.020571 + 0.005291) = 0.204585879$.

Naïve Bayes Implementation using Scikit-Learn

```
import numpy as np
# CategoricalNB is for doing Naive Bayes classifier for categorical features
from sklearn.naive_bayes import CategoricalNB

# Forming training data
# Features: Blood Pressure (0=High,1=Normal,2=Low), Fever (0=High,1=Mild,2=No Fever),
#           Diabetes (0=Yes,1=No), Vomit (0=Yes,1=No)
training = np.array([[0, 0, 0, 1], [0, 0, 0, 0], [2, 0, 0, 1], [1, 1, 0, 1], [1, 2, 1, 1],
                     [1, 2, 1, 0], [2, 2, 1, 0], [0, 1, 0, 1], [0, 2, 1, 1], [1, 1, 1, 1],
                     [0, 1, 1, 0], [2, 1, 0, 0], [2, 0, 1, 1], [1, 1, 0, 0]])
# Forming the label set
# Labels: 0=Yes, 1=No
outcome = np.array([1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1])

# This is the new data to be evaluated using the trained AI (model)
new_sample = np.array([[0, 2, 0, 0]])

# Train Naive Bayes classifier according to training, outcome
clf = CategoricalNB(alpha=1.0e-10).fit(training, outcome)
# Perform classification on new_sample, and get the probability estimates
pred_class = clf.predict(new_sample); prob = clf.predict_proba(new_sample)
# Print the results
print(pred_class, prob)
```

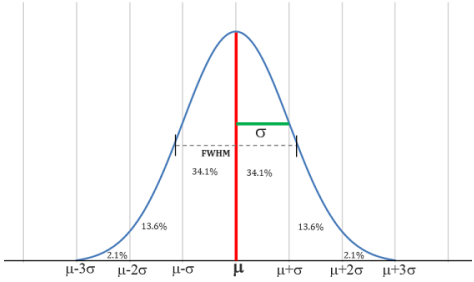
Calculation for Continuous Variables

- In the last example, the symptoms – Blood Pressure and Fever are discrete values. What if they are continuous?
- If the Blood Pressure and Fever were continuous, the calculation would have been different.
- Assume the probability of likelihood follows Gaussian distribution, i.e.,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

where

- μ: Mean
- σ: Standard deviation



Calculation for Continuous Variables

- We need to calculate the mean and standard deviation of Blood Pressure and Fever for each belief (i.e., suffering from disease 'Z' and not suffering from disease 'Z').
- Suppose,
  - For Blood Pressure
    - Suffering from disease 'Z' is Yes: Mean (μ) = 73, Standard deviation (σ) = 6.2
    - Suffering from disease 'Z' is No: Mean (μ) = 75, Standard deviation (σ) = 7.9
  - For Fever
    - Suffering from disease 'Z' is Yes: Mean (μ) = 79, Standard deviation (σ) = 10.2
    - Suffering from disease 'Z' is No: Mean (μ) = 86, Standard deviation (σ) = 9.7

Any potential problem of this approach?

Calculation for Continuous Variables

- For Blood Pressure = 66, what will be $P(Blood\ Pressure = 66|Disease\ Z = Yes)$?

$$P(Blood\ Pressure = 66|Disease\ Z = Yes) = \frac{1}{\sqrt{2\pi}(6.2)} exp\left(-\frac{(66-73)^2}{2*(6.2)^2}\right) = 0.0340187$$

- For Fever = 97, what will be $P(Fever = 97|Disease\ Z = Yes)$?

$$P(Fever = 97|Disease\ Z = Yes) = \frac{1}{\sqrt{2\pi}(10.2)} exp\left(-\frac{(97-79)^2}{2*(10.2)^2}\right) = 0.00824276$$

- Posterior

$$P(DiseaseZ = Yes)(P(BP = 66|Yes)P(Fever = 97|Yes)P(Diabetes = Yes|Yes)P(Vomit = Yes|Yes))$$
$$= \left(\frac{9}{14}\right)(0.0340187)(0.00824276)\left(\frac{3}{9}\right)\left(\frac{3}{9}\right) = 0.00002$$

Calculation for Continuous Variables

- For Blood Pressure = 66, what will be $P(Blood\ Pressure = 66|Disease\ Z = No)$?

$$P(Blood\ Pressure = 66|Disease\ Z = No) = \frac{1}{\sqrt{2\pi}(7.9)} exp\left(-\frac{(66-75)^2}{2*(7.9)^2}\right) = 0.0263909$$

- For Fever = 97, what will be $P(Fever = 97|Disease\ Z = No)$?

$$P(Fever = 97|Disease\ Z = No) = \frac{1}{\sqrt{2\pi}(9.7)} exp\left(-\frac{(97-86)^2}{2*(9.7)^2}\right) = 0.0216215$$

- Posterior

$$P(DiseaseZ = No)(P(BP = 66|No)P(Fever = 97|No)P(Diabetes = Yes|No)P(Vomit = Yes|No))$$
$$= \left(\frac{5}{14}\right)(0.0263909)(0.0216215)\left(\frac{4}{5}\right)\left(\frac{3}{5}\right) = 0.0000978$$

- Result: The Patient is not suffering from Disease 'Z'. Since $0.0000978 > 0.00002$.
- After normalization, we have $0.0000978/(0.000978 + 0.00002) = 0.8302207$, $0.00002/(0.000978 + 0.00002) = 0.1697793$.

Underflow Prevention

- Multiplying lots of probabilities between 0 and 1 by definition, can result in floating-point underflow.
- $log(xy) = log(x) + log(y)$, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Belief with the highest final log probability score is still the most probable.

$$B_{NB} = argmax_{B_i} \left(logP(B_i) + \sum_{n=1}^{d} logP(e_n|B_i) \right)$$

Applications of Naïve Bayes Algorithm

- Real-time prediction: Naïve Bayes Algorithm is fast and always ready to learn hence best suited for real-time predictions.
- Multi-class prediction: The probability of multi-classes of any target variable can be predicted using a Naïve Bayes algorithm.
- Text classification where Naïve Bayes is mostly used is Spam Filtering in emails (Naïve Bayes is widely used for text classification). Due to its better performance with multi-class problems and its independence rule, the Naïve Bayes algorithm performs better or have a higher success rate in text classification. Therefore, it is used in sentiment analysis and spam filtering.
- Recommendation system: Naïve Bayes classifier and collaborative filtering together build a recommendation system that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

Advantages and Disadvantages of Naïve Bayes

- Advantages
  - It is very easy to implement and obtain very good results on large datasets.
  - It has computational efficiency. The assumption that all features are independent makes a Naïve Bayes algorithm very fast compared to complicated algorithms. In some cases, speed is preferred over higher accuracy.
  - It works well on a large size dataset.
  - It can also be used to predict multiple classes.
  - It can also use in the Natural Language Processing text classification.
- Disadvantages
  - The strong assumption about the features to be independent which is hardly true in real-life applications.
  - If the dataset size is small, it will decrease the precision value.
  - When the goal is to predicting probability instead of classification, then the method provides very biased results.
  - If the categorical variable has a category in the test data set, which was not observed in the training data set, then the model will assign a 0 probability and will be unable to make a prediction. This is often known as Zero Frequency.

Practice Problem

	Animals	Size	Color	Can we pet them?
1	Dog	Medium	Black	Yes
2	Dog	Big	White	No
3	Rat	Small	White	Yes
4	Cow	Big	White	Yes
5	Cow	Small	Brown	No
6	Cow	Big	Black	Yes
7	Rat	Big	Brown	No
8	Dog	Small	Brown	Yes
9	Dog	Medium	Brown	Yes
10	Cow	Medium	White	No
11	Dog	Small	Black	Yes
12	Rat	Medium	Black	No
13	Rat	Small	Brown	No
14	Cow	Big	White	Yes

For the animal (Cow, Medium, Black), can we pet it?

Practice Problem

- Given the naïve bayes' formula

$$P(B_i|E) = P(B_i)(P(e_1|B_i)P(e_2|B_i)P(e_3|B_i)\cdots P(e_n|B_i))$$

  we have

$$P(Pet = Yes|(Animals, Size, Color))$$
$$= P(Pet = Yes)(P(Animals|Yes)P(Size|Yes)P(Color|Yes))$$

- Now, (Animals, Size, Color) = (Cow, Medium, Black)

$$P(Pet = Yes)(P(Cow|Yes)P(Medium|Yes)P(Black|Yes))$$
$$= \left(\frac{8}{14}\right)\left(\frac{3}{8}\right)\left(\frac{2}{8}\right)\left(\frac{3}{8}\right) = 0.02$$
$$P(Pet = No)(P(Cow|No)P(Medium|No)P(Black|No))$$
$$= \left(\frac{6}{14}\right)\left(\frac{2}{6}\right)\left(\frac{2}{6}\right)\left(\frac{1}{6}\right) = 0.0079$$

- Result: We can pet it. Since $0.02 > 0.0079$.
- After normalization, we have $0.02/(0.02 + 0.0079) = 0.7168459$, $0.0079/(0.02 + 0.0079) = 0.2831541$.

Naïve Bayes Implementation using Scikit-Learn

```
import numpy as np
# CategoricalNB is for doing Naive Bayes classifier for categorical features
from sklearn.naive_bayes import CategoricalNB

# Forming training data
# Features: Animals (0=Dog,1=Rat,2=Cow), Size (0=Big,1=Medium,2=Small),
#           Color (0=Black, 1=White, 2=Brown)
training = np.array([[0, 1, 0], [0, 0, 1], [1, 2, 1], [2, 0, 1], [2, 2, 2],
                     [2, 0, 0], [1, 0, 2], [0, 2, 2], [0, 1, 2], [2, 1, 1],
                     [0, 2, 0], [1, 1, 0], [1, 2, 2], [2, 0, 1]])
# Forming the label set
# Labels: 0=Yes, 1=No
outcome = np.array([0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0])

# This is the new data to be evaluated using the trained AI (model)
new_sample = np.array([[2, 1, 0]])

# Train Naive Bayes classifier according to training, outcome
clf = CategoricalNB(alpha=1.0e-10).fit(training, outcome)
# Perform classification on new_sample and get the probability estimates
pred_class = clf.predict(new_sample); prob = clf.predict_proba(new_sample)
# Print the results
print(pred_class, prob)
```

That's all!

Any questions?