Naïve Bayes

Problem: Email classification

Suppose we have this stats from a simple training data of 500 emails:

	# Emails	"*Discount*" appears	"*Free*" appears
Spam	100	80	70
Not Spam	400	20	40
Total	500		

If you are given an email that contains the words ***"Discount" and "free"*** , then what is the probability that given email is **spam** ?

i.e. Compute P(Spam | *Discount, Free*)

Naïve Bayes

Naïve Bayes falls under **Supervised Machine Learning.**

It is used for **classification task**, **not for regression task**

Cont…

Naïve Bayes

Bayes Theorem:

$$P(Y \mid X) = \frac{P(X \mid Y)\, P(Y)}{P(X)}$$

x_1	x_2	x_3	...	x_n	Y
			..		
			..		
			..		

- Y = class label (e.g., spam or not spam)
- X = feature vector (e.g., words in an email)
- P(Y|X) = probability of class Y given the features
- P(X|Y) = probability of observing those features given Y
- P(Y) = prior probability of the class
- P(X) = probability of the features

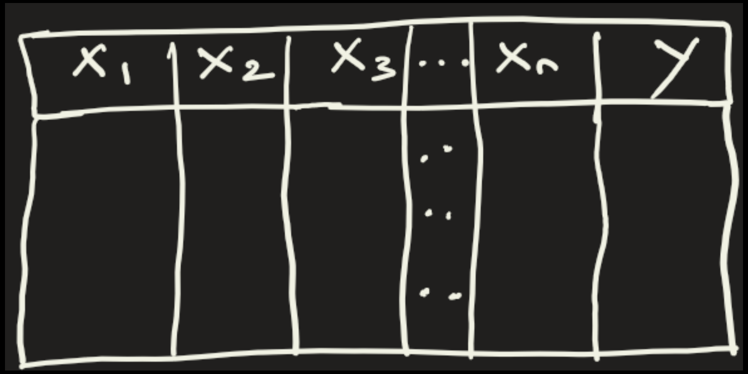
The classifier predicts the class with the **highest posterior probability**:

$$\hat{Y} = \arg\max_{Y} P(Y \mid X)$$

$$\hat{y} = \operatorname{argmax}[0.2, 0.8] = 1$$

Naïve Bayes

Why "Naïve"? (The Big Assumption)



Naïve Bayes assumes that **all features are independent of each other given the class**.
For example, we assume that in Spam classification, the words "discount" and "free" are Independent of each other (i.e. uncorrelated)

Mathematically:

$$P(X_1, X_2, \ldots, X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$$

This assumption is almost never true in real life — features are usually correlated.
But surprisingly, Naïve Bayes works extremely well in **many** applications (especially text classification).

This Naïve assumption makes math easy.

Naïve Bayes

The final Naïve Bayes Formula:

$$P(Y \mid x_1, x_2, \ldots, x_n)$$

$$= \frac{P(x_1, x_2, \ldots x_n \mid Y) \cdot P(Y)}{P(x_1, x_2, \ldots, x_n)}$$

(apply Naive assumption)

$$= \frac{P(x_1 \mid Y) \cdot P(x_2 \mid Y) \cdots P(x_n \mid Y) \cdot P(Y)}{P(x_1, x_2, \ldots, x_n)}$$

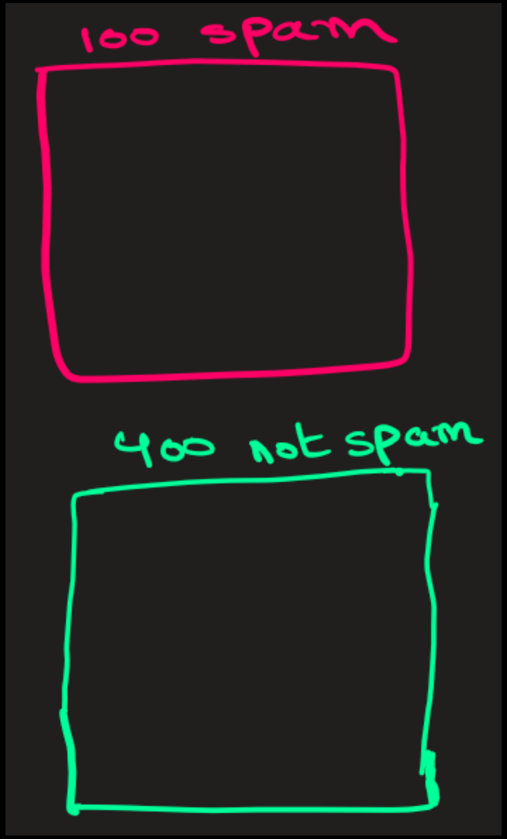
$$\propto P(x_1 \mid Y) \cdot P(x_2 \mid Y) \cdots P(x_n \mid Y) \cdot P(Y)$$

x_1	x_2	x_3	...	x_n	Y
			..		
			..		
			..		

The denominator is just a constant in all classes

Naïve Bayes

Problem 1: Email classification
We have training data of 500 emails: 100 of them are **spam** and 400 are **not spam.**



What is the probability that given random email is a spam ?
i.e. Compute P(Spam)

Ans:

$$P(\text{Spam}) = \frac{100}{500} = 0.2 = 20\%$$

Naïve Bayes

Problem2: Email classification

Now I give you additional information about this training data. I give you following stats:

	# Emails	"Discount" appears	"Free" appears
Spam	100	80	70
Not Spam	400	20	40
Total	500		

What is the probability that given email is spam **given that it contains words "Discount" and "free"** ?

i.e. Compute P(Spam | *Discount, Free*)

Naïve Bayes

$$P(Y|X_1,X_2) = \frac{P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

Here we have to calculate:

$$P(\underbrace{Y}_{\text{Spam}}|\underbrace{X_1}_{\text{"Discount"}},\underbrace{X_2}_{\text{"Free"}}) = \frac{P(X_1,X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

Here we assume that X_1, X_2 are independant:

$$P(X_1,X_2|Y) = P(X_1|Y)\cdot P(X_2|Y)$$

$$P(Y|X_1,X_2) = \frac{P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

$$\propto P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)$$

100 spam

"Discount"	"Free"
$\downarrow$	$\downarrow$
80	70

400 not spam

"Discount"	"Free"
$\downarrow$	$\downarrow$
20	40

Now we calculate each term separately

Naïve Bayes

$$P(Y|X_1,X_2) = \frac{P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

Terminology:

$$\overbrace{P(Y|X_1,X_2)}^{\text{Posterior}} \propto \overbrace{P(X_1|Y)\cdot P(X_2|Y)}^{\text{Likelihood}} \cdot \overset{\overset{\text{Prior}}{\uparrow}}{P(Y)}$$

100 spam

"Discount" $\downarrow$ 80

"Free" $\downarrow$ 70

400 not spam

"Discount" $\downarrow$ 20

"Free" $\downarrow$ 40

Naïve Bayes

$$P(Y|X_1,X_2) = \frac{P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

1. Compute the priors

$$\mathrm{P(Spam)} = \frac{100}{500} = 0.2$$

$$\mathrm{P(not\text{-}Spam)} = \frac{400}{500} = 0.8$$

2. Compute likelihoods

$$\mathrm{P}(\mathit{Discount}|\ \mathrm{Spam}) = \frac{80}{100} = 0.8$$

$$\mathrm{P}(\mathit{Free}|\ \mathrm{Spam}) = \frac{70}{100} = 0.7$$

$$\mathrm{P}(\mathit{Discount}|\mathrm{not\text{-}Spam}) = \frac{20}{400} = 0.05$$

$$\mathrm{P}(\mathit{Free}|\mathrm{not\text{-}Spam}) = \frac{40}{400} = 0.1$$

100 spam

"Discount" → 80

"free" → 70

400 not spam

"Discount" → 20

"free" → 40

Naïve Bayes

$$P(Y|X_1,X_2) = \frac{P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

3. The posterior

For each class:

$$\mathrm{P}(\mathrm{Spam}|\,Discount, Free) \propto \mathrm{P}(Discount \mid \mathrm{Spam}) \times \mathrm{P}(Free \mid \mathrm{Spam}) \times \mathrm{P}(\mathrm{Spam})$$

$$= 0.8 \times 0.7 \times 0.2 = 0.112$$

$$\mathrm{P}(\text{not-Spam}|\,Discount, Free) \propto \mathrm{P}(Discount \mid \text{not-Spam}) \times \mathrm{P}(Free \mid \text{not-Spam}) \times \mathrm{P}(\text{not-Spam})$$

$$= 0.05 \times 0.1 \times 0.8 = 0.004$$

Since 0.112 >> 0.004, therefore, **the email that contains words Discount and Free is spam.**

100 spam

"Discount" → 80

"Free" → 70

400 not spam

"Discount" → 20

"Free" → 40

NOTE: The above is not normalized, so let's normalize:

$$\mathrm{P}(\mathrm{Spam}|\,Discount, Free) = \frac{0.112}{0.112 + 0.004} = 0.97$$

$$\mathrm{P}(\text{not-Spam}|\,Discount, Free) = \frac{0.004}{0.112 + 0.004} = 0.03$$

Conclusion: If an email contains words "*Discount*" and "*Free*", then it is 97% chance that it is Spam

Naïve Bayes: Types

Naive Bayes classifiers are categorized based on the type of data they handle.

Bernoulli Naive Bayes: It is designed for **binary or Boolean features**. It's effective in scenarios where data is represented as **yes/no or true/false or 0/1.**
This classifier is frequently employed in **spam detection and sentiment analysis.**

Multinomial Naive Bayes: It excels with **discrete data.** This classifier is adept at handling features that represent counts, like word frequencies in documents.
It's commonly used in **text classification tasks and document categorization.**

Gaussian Naive Bayes: It is suited for **continuous data.** It posits that the features adhere to a Gaussian distribution.
This classifier is particularly useful for numerical data, such as **measurements or sensor readings.**

Naïve Bayes: Types

Which Naïve Bayes to use when you have **mixed data types** in feature space X ?

Ans: Use **Label encoding** and then use **Gaussian NB**.

x_1 age	x_2 income	x_3 gender	x_4 depart.
40	30K	m	HR
31	20K	f	IT
23	40K	m	Sales
22	30k	m	IT
⋮	⋮	⋮	⋮

Label encoding →

x_1 age	x_2 income	x_3 gender	x_4 depart.
40	30K	0	0
31	20K	1	1
23	40K	0	2
22	30k	0	1
⋮	⋮		⋮

Naïve Bayes: Advantages

1. Simple and Easy to Implement

- Based on basic probability rules -> Very easy to understand and code.
- No need for scaling or standardization
- Requires very little parameter tuning.

2. Works Extremely Well for Large Datasets

- Training is fast because it only calculates probabilities.
- Works efficiently even with millions of records.
- Ideal for **real-time prediction systems**.

3. Performs Well in High-Dimensional Data

- Works well even when the number of features is very large.
- Very effective in:
    - Text classification
    - Spam detection
    - Document categorization

Naïve Bayes: Disadvantages

Naïve Assumption of Feature Independence

- Assumes that all features are independent.
- In real-world data, features are usually correlated: “hot” and “sunny” in weather prediction.
- This can reduce accuracy.

STOP

EXTRA

Naïve Bayes

Bayes Theory works on coming to a hypothesis (h) from a given set of data D. It relates to two things: the probability of the hypothesis before the evidence P(h) and the probability after the data D is given P(h|D). The Bayes Theory is explained by the following equation:

$$\mathbf{P(h|D) = (P(D|h) * P(h))/P(D)}$$

- P(h): the probability of hypothesis h being true (regardless of the data). This is known as the **prior** probability of h.
- P(D): the probability of the data (regardless of the hypothesis). This is known as the prior probability.
- P(h|D): the probability of hypothesis h given the data D. This is known as **posterior** probability.
- P(D|h): the probability of data d given that the hypothesis h was true. This is known as **posterior** probability.

Assumption of naïve bayes:

- One feature does not affect other features.(Aka feature independence). This is called “Naive” assumption. This is what drives the following formula:

If any two events **X1 and X2 are independent**, then, $\mathbf{P(X1,X2) = P(X1) * P(X2)}$

If any two events **X1 and X2 are conditionally independent given Y**, then, $\mathbf{P((X1,X2) \mid Y) = P(X1 \mid Y) * P(X2 \mid Y)}$

- All features contribute equally to the outcome
- Continuous features are normally distributed:
- Discrete features have multinomial distributions

Naïve Bayes Theorem application:

$$P(Y|X_1,X_2) = \frac{P(X_1,X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

Now apply Naive assumption: X_1 & X_2 are conditionally indep. given Y

i.e. $P(X_1,X_2|Y) = P(X_1|Y)\cdot P(X_2|Y)$

$$= \frac{P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)}{P(X_1,X_2)}$$

$$\propto P(X_1|Y)\cdot P(X_2|Y)\cdot P(Y)$$

Example1 : Medical Diagnosis

Scenario

Y: A patient has a disease.

X1: The patient's blood test is positive for disease.

X2: The patient's X-ray is positive for disease.

Based on empirical observation of past data, we observed that

Prior: P(Y) = 0.01 (1% of people have disease)

Likelihoods:

P(X1 | Y) = 0.95 (Blood test is positive if disease present)

P(X2 | Y) = 0.90 (X-ray is positive if disease present)

False positives:

P(X1 |¬Y) = 0.05 (Blood test is positive if no disease)

P(X2 |¬Y) = 0.10 (X-ray is positive if no disease)

Assume tests are conditionally independent given Y or ¬Y.

Question: Given that both tests are positive, what's the chance the patient actually has the disease ? i.e. calculate P(Y | X1,X2).

Answer

First, compute:

P(X1,X2|Y) = P(X1|Y) * P(X2|Y) = 0.95 * 0.90 = 0.855

P(X1,X2 |¬Y) = P(X1|¬Y) * P(X2|¬Y) = 0.05 * 0.10 = 0.005

Next, use Law of Total Probability:

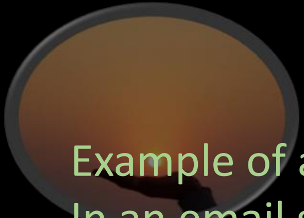
P(X1, X2) = P(X1, X2 |Y)*P(Y) + P(X1,X2 |¬Y)*P(¬Y)

= 0.855×0.01+0.005×0.99=0.0135

Finally, apply Bayes:

P(Y| X1,X2) = (0.855*0.01)/0.0135 = 0.63

Given both tests are positive, there's a 63% chance the patient has the disease, even though the prior was only 1%!



Example of application of Bayes Theorem:
In an email spam filter (like Naive Bayes classifier), Bayes Theorem is used to calculate P(Spam | Words) - the probability that an email is spam given the words it contains.

Question: Suppose from past historical data we have following:
1) 40% of all emails are spam: P(Spam) = 0.4
2) The word “Discount” appears in 70% of spam emails: P(“Discount” | Spam) = 0.7
3) The word “Discount” appears in 1% of non-spam emails: P(“Discount” | Not Spam) = 0.01
What is the probability that an email is spam given that word “discount” appears in it?

Solution: When a new email has the word “Discount”, the filter calculates:
STEP 1) calculate P(“Discount”)
P(“Discount”) , which is the total probability that any email contains “Discount”, whether it’s spam or not can be calculated using the Law of Total Probability:
P(“Discount”)
$= P(\text{“Discount”} \mid \text{Spam}) \times P(\text{Spam}) + P(\text{“Discount”} \mid \text{Not Spam}) \times P(\text{Not Spam})$
$= (0.7 \times 0.4) + (0.01 \times 0.6) = 0.28 + 0.006 = 0.286.$

STEP2) calculate P(Spam | “Discount”):
$P(\text{Spam} \mid \text{“Discount”}) = P(\text{“Discount”} \mid \text{Spam}) \times P(\text{Spam}) / P(\text{“Discount”}) = (0.7 * 0.4) / 0.286 = 0.979$
This is Bayes’ Theorem in action.

Email classification (style2)

Suppose you have **10 emails**, labeled as *Spam* or *Not Spam*, and also a column indicating whether the email contain the words **“Discount”** and **“Free”**.

Email	**Spam/Not Spam**	**Contains “Discount”?**	**Contains “Free”?**
E1	Spam	Yes	Yes
E2	Spam	Yes	Yes
E3	Spam	Yes	No
E4	Spam	No	Yes
E5	Spam	Yes	No
E6	Not Spam	Yes	No
E7	Not Spam	No	Yes
E8	Not Spam	No	No
E9	Not Spam	No	No
E10	Not Spam	Yes	No

Step 1: Calculate the Priors

Total emails = 10

Spam emails = 5

Not Spam emails = 5

So:

P(Spam)= 5/10 = 0.5

P(not Spam)= 5/10 = 0.5

Email classification (style2)

Step 2: Calculate the Likelihoods
For each word and each class, compute:

A) P(Discount | Spam)
Spam emails: 5
Of these, 4 contain “Discount” (E1, E2, E3, E5)
P(Discount | Spam)= 4/5 =0.8

B) P(Free | Spam)
Spam emails: 5
Of these, 3 contain “Free” (E1, E2, E4)
P(Free|Spam)= 3/5 =0.6

C) P(Discount | Not Spam)
Not Spam emails: 5
Of these, 2 contain “Discount” (E6, E10)
P(Discount | Not Spam) = 2/5 =0.4

D) P(Free | Not Spam)
Not Spam emails: 5
Of these, 1 contains “Free” (E7)
P(Free | Not Spam)= 1/5 =0.2

Step 3: Now classify a new email
Suppose you receive an email that contains both “Discount” and “Free”.
You want to know P(Spam|Discount, Free)?

📌 **By Naive Bayes:**

$$P(\text{Spam}|\text{Discount, Free}) \propto P(\text{Spam}) \times P(\text{Discount}|\text{Spam}) \times P(\text{Free}|\text{Spam})$$

$$P(\text{Not Spam}|\text{Discount, Free}) \propto P(\text{Not Spam}) \times P(\text{Discount}|\text{Not Spam}) \times P(\text{Free}|\text{Not Spam})$$

So:
For Spam: P(spam | discount, free) α 0.5×0.8×0.6 = 0.24
For Not Spam: P(not-spam | discount, free) α 0.5×0.4×0.2 = 0.04
So, the non-normalized probability is

Class	Probability
Spam	0.24
Not Spam	0.04

To get the normalized probability that it’s Spam:
P(Spam | Discount, Free) = 0.24 / (0.24+0.04) = 0.857
So ~86% chance the email is Spam!

1. First Approach (In case of a 1 feature)(p1)

Naive Bayes classifier calculates the probability of an event in the following steps:

Step 1: Calculate the prior probability for given class labels

Step 2: Find Likelihood probability with each attribute for each class

Step 3: Put these value in Bayes Formula and calculate posterior probability.

Step 4: See which class has a higher probability, given the input belongs to the higher probability class.

1. First Approach (In case of a 1 feature)(p2)

Whether	Play
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Whether	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9

Likelihood Table 1

Whether	No	Yes		
Overcast		4	=4/14	0.29
Sunny	2	3	=5/14	0.36
Rainy	3	2	=5/14	0.36
Total	5	9		
	=5/14	=9/14		
	0.36	0.64		

Likelihood Table 2

Whether	No	Yes	Posterior Probability for No	Posterior Probability for Yes
Overcast		4	0/5=0	4/9=0.44
Sunny	2	3	2/5=0.4	3/9=0.33
Rainy	3	2	3/5=0.6	2/9=0.22
Total	5	9		

1. First Approach (In case of a 1 feature)(p3)

Now suppose you want to calculate the probability of playing when the weather is overcast. Here **h**=Yes, **D**=overcast.

(A) Probability of playing:

P(Yes | Overcast) = P(Overcast | Yes) P(Yes) / P (Overcast) ->(1)

1.Calculate Prior Probabilities:
P(Overcast) = 4/14 = 0.29
P(Yes)= 9/14 = 0.64

2. Calculate Posterior Probabilities:
P(Overcast |Yes) = 4/9 = 0.44

3. Put Prior and Posterior probabilities in equation (1)
P (Yes | Overcast) = 0.44 * 0.64 / 0.29 = 1(Higher)

(B) Probability of not playing:

P(No | Overcast) = P(Overcast | No) P(No) / P (Overcast) -> (2)

1.Calculate Prior Probabilities:
P(Overcast) = 4/14 = 0.29
P(No)= 5/14 = 0.36

2. Calculate Posterior Probabilities:
P(Overcast |No) = 0/9 = 0

3. Put Prior and Posterior probabilities in equation (2)
P (No | Overcast) = 0 * 0.36 / 0.29 = 0 (or use 1-1 = 0)

The probability of a 'Yes' class is higher.
So, based on historical data we can conclude that, if the weather is overcast then players will play the sport.

2. Second Approach (In case of multiple features)(p1)

Now suppose you want to calculate the probability of playing when the weather is overcast, and the temperature is mild.

Here h is Play=yes and

D is (weather = overcast, temperature = mild)

Whether	Temperature	Play
Sunny	Hot	No
Sunny	Hot	No
Overcast	Hot	Yes
Rainy	Mild	Yes
Rainy	Cool	Yes
Rainy	Cool	No
Overcast	Cool	Yes
Sunny	Mild	No
Sunny	Cool	Yes
Rainy	Mild	Yes
Sunny	Mild	Yes
Overcast	Mild	Yes
Overcast	Hot	Yes
Rainy	Mild	No

2. Second Approach (In case of multiple features)(p2)

Probability of playing:

P(Play= Yes | Weather=Overcast, Temp=Mild)
= P(Weather=Overcast, Temp=Mild | Play= Yes) * P(Play=Yes) -> (1)

P(Weather=Overcast, Temp=Mild | Play= Yes)
= P(Overcast |Yes) P(Mild |Yes) ->(2)

1.Calculate **Prior** Probabilities: P(Yes)= 9/14 = 0.64
2.Calculate **Posterior** Probabilities:
P(Overcast |Yes) = 4/9 = 0.44
P(Mild |Yes) = 4/9 = 0.44

3. Put Posterior probabilities in equation (2)
P(Weather=Overcast, Temp=Mild | Play= Yes) = 0.44 * 0.44 = 0.1936(Higher)

4. Put Prior and Posterior probabilities in equation (1)
P(Play= Yes | Weather=Overcast, Temp=Mild) = 0.1936*0.64 = 0.124

Whether	Temperature	Play
Sunny	Hot	No
Sunny	Hot	No
Overcast	Hot	Yes
Rainy	Mild	Yes
Rainy	Cool	Yes
Rainy	Cool	No
Overcast	Cool	Yes
Sunny	Mild	No
Sunny	Cool	Yes
Rainy	Mild	Yes
Sunny	Mild	Yes
Overcast	Mild	Yes
Overcast	Hot	Yes
Rainy	Mild	No

2. Second Approach (In case of multiple features)(p2)

Probability of not playing:

P(Play= No | Weather=Overcast, Temp=Mild)
*= P(Weather=Overcast, Temp=Mild | Play= No)*P(Play=No) -> (3)

P(Weather=Overcast, Temp=Mild | Play= No)
= P(Weather=Overcast |Play=No) P(Temp=Mild | Play=No) -> (4)

1.Calculate Prior Probabilities:
P(No)= 5/14 = 0.36

2. Calculate Posterior Probabilities:
P(Weather=Overcast |Play=No) = 0/5 = 0
P(Temp=Mild | Play=No)=2/5=0.4

3. Put posterior probabilities in equation (4)
P(Weather=Overcast, Temp=Mild | Play= No) = 0 * 0.4= 0

4. Put prior and posterior probabilities in equation (3)
P(Play= No | Weather=Overcast, Temp=Mild) = 0*0.36=0

The probability of a 'Yes' class is higher. So, if the weather is overcast and temp. is mild, then players will play the sport.

Whether	Temperature	Play
Sunny	Hot	No
Sunny	Hot	No
Overcast	Hot	Yes
Rainy	Mild	Yes
Rainy	Cool	Yes
Rainy	Cool	No
Overcast	Cool	Yes
Sunny	Mild	No
Sunny	Cool	Yes
Rainy	Mild	Yes
Sunny	Mild	Yes
Overcast	Mild	Yes
Overcast	Hot	Yes
Rainy	Mild	No

TODO: Gaussian NB, Binomial NB

Naïve Bayes

$$P(Y \mid X_1, X_2) = \frac{P(X_1 \mid Y) \cdot P(X_2 \mid Y) \cdot P(Y)}{P(X_1, X_2)}$$

When does this fail?
If the words aren't independent (e.g., "Discount" and "Free" often appear together for a reason), then this is only an approximation.

But despite the "Naive" part, it works well for text classification because real text data has lots of words, so the independence assumption is surprisingly robust in practice!