Do It Skills
do it. enjoy it.

NAÏVE BAYES

MACHINE LEARNING CLASSIFICATION ALGORITHM

CONDITIONAL PROBABILITY

1. In probability theory, conditional probability is a measure of the probability of an event given that another event has already occurred.

2. If the event of interest is A and the event B is assumed to have occurred, "the conditional probability of A given B", or "the probability of A under the condition B", is usually written as P(A|B), or sometimes PB(A).

Probability of event A occured and event B occured

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

Probability of event A given B has occured

Probability of event B

Do It Skills
do it. enjoy it.

BAYES THEOREM

What is Bayes Theorem?

1. In probability theory and statistics, Bayes' theorem (alternatively Bayes' law or Bayes' rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

Statement of theorem [edit]

Bayes' theorem is stated mathematically as the following equation:[2]

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

where A and B are events and $P(B) \neq 0$.

- $P(A)$ and $P(B)$ are the probabilities of observing A and B without regard to each other.
- $P(A \mid B)$, a conditional probability, is the probability of observing event A given that B is true.
- $P(B \mid A)$ is the probability of observing event B given that A is true.

Do It Skills
do it. enjoy it.

Posterior Probability Likelihood Class Prior Probability

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

Predictor Prior Probability

where A and B are events and $P(B) \neq 0$.

- $P(A)$ and $P(B)$ are the probabilities of observing A and B without regard to each other.
- $P(A \mid B)$, a conditional probability, is the probability of observing event A given that B is true.
- $P(B \mid A)$ is the probability of observing event B given that A is true.

NAIVE BAYES CLASSIFIER

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.

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

where A and B are events and $P(B) \neq 0$.

- $P(A)$ and $P(B)$ are the probabilities of observing A and B without regard to each other.
- $P(A \mid B)$, a conditional probability, is the probability of observing event A given that B is true.
- $P(B \mid A)$ is the probability of observing event B given that A is true.

Posterior Probability Likelihood Class Prior Probability

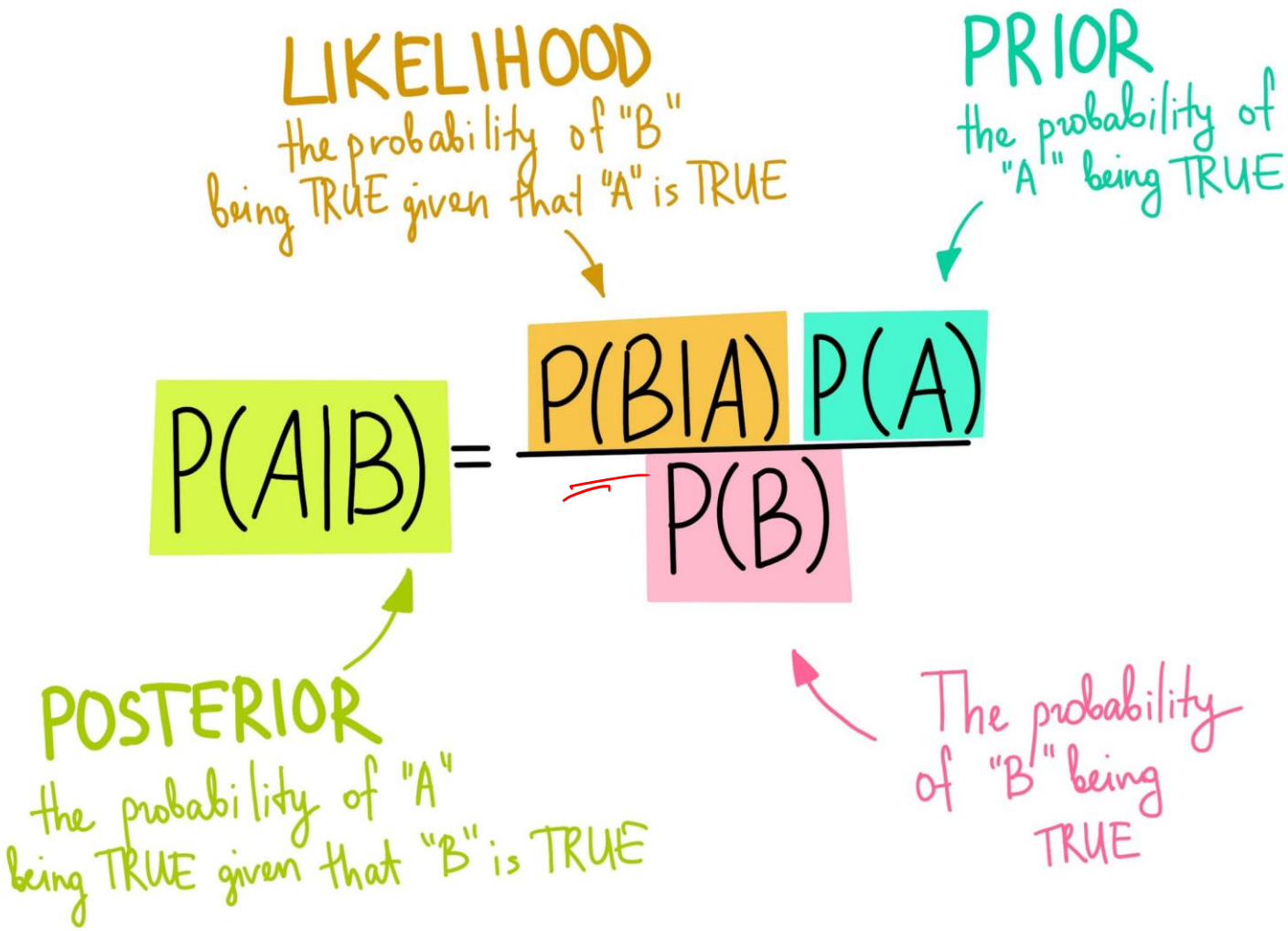
$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

Predictor Prior Probability

Likelihood: This is the "likelihood" of seeing that evidence if your hypothesis is correct.

Prior: This is the "prior". what you believed before you saw the evidence.

Predictor: This is the normalizing constant. The likelihood of that evidence under any circumstances


LIKELIHOOD
the probability of "B" being TRUE given that "A" is TRUE
PRIOR
the probability of "A" being TRUE
P(A|B) = P(B|A) P(A) / P(B)
POSTERIOR
the probability of "A" being TRUE given that "B" is TRUE
The probability of "B" being TRUE

WHY NAÏVE?

Naive Bayes (NB) is ‘naive’ because it makes the assumption that features of a measurement are independent of each other.

Assumptions:

- All the events (i.e. the features) of the dataset are not at all dependent on one another.
- Each event contributes equally to classifying the outcome.

NAÏVE BAYES FORMULA

Bayes Theorem : $$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Naïve bayes classifier with “n” features :

$$P(y|x_1, \ldots, x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

(Likelihood) (Class Prior Probability)

Posterior Probability $P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$ Predictor Prior Probability

Likelihood: This is the "likelihood" of seeing that evidence if your hypothesis is correct.

Prior: This is the "prior". what you believed before you saw the evidence.

Predictor: This is the normalizing constant. The likelihood of that evidence under any circumstances

(Likelihood) (Class Prior Probability)

Posterior Probability $P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$ Predictor Prior Probability

Probability of Yes given Today

A= Yes
B=Today

Likelihood: This is the "likelihood" of seeing that evidence if your hypothesis is correct.

Prior: This is the "prior". what you believed before you saw the evidence.

Predictor: This is the normalizing constant. The likelihood of that evidence under any circumstances

NAÏVE BAYES FOR SINGLE VARIABLE

Outlook	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

NAIVE BAYES CLASSIFIER

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.

NAÏVE BAYES FOR MULTIPLE VARIABLE

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

01 CALCULATE PRIOR PROBABILITY FOR GIVEN CLASS LABELS

02 CALCULATE CONDITIONAL PROBABILITY WITH EACH ATTRIBUTE FOR EACH CLASS

03 MULTIPLY SAME CLASS CONDITIONAL PROBABILITY.

04 MULTIPLY PRIOR PROBABILITY WITH STEP 3 PROBABILITY.

05 SEE WHICH CLASS HAS HIGHER PROBABILITY, HIGHER PROBABILITY CLASS BELONGS TO GIVEN INPUT SET STEP.

MARGINAL LIKELIHOOD?

By Bayes Theorem, for P(Yes|today) and P(No|today) if we multiply by P(today) and eliminate the term then we don't need to find P(today).

$$P(\text{Yes}|\text{today}) = \frac{P(\text{Outlook(Overcast)}|\text{Yes}) * P(\text{Temp(Mild)}|\text{Yes}) * P(\text{Yes})}{P(\text{today})} \quad X \; P(\text{today})$$

$$P(\text{No}|\text{today}) = \frac{P(\text{Outlook(Overcast)}|\text{No}) * P(\text{Temp(Mild)}|\text{No}) * P(\text{No})}{P(\text{today})} \quad X \; P(\text{today})$$

MARGINAL LIKELIHOOD?

By Bayes Theorem, for P(Yes|today) and P(No|today) if we multiply by P(today) and eliminate the term then we don't need to find P(today).

$$P(\text{Yes}|\text{today}) = \frac{P(\text{Outlook(Overcast)}|\text{Yes}) * P(\text{Temp(Mild)}|\text{Yes}) * P(\text{Yes})}{P(\text{today})} \quad X \; P(\text{today})$$

$$P(\text{No}|\text{today}) = \frac{P(\text{Outlook(Overcast)}|\text{No}) * P(\text{Temp(Mild)}|\text{No}) * P(\text{No})}{P(\text{today})} \quad X \; P(\text{today})$$

(Likelihood) (Class Prior Probability)

Posterior Probability $P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$ Predictor Prior Probability

Probability of Yes given Today

A= Yes
B=Today

Do It Skills
do it. enjoy it.

PLAY (YES/NO)?

$$P(\text{Yes}\mid\text{today}) = P(\text{Outlook(Overcast)}\mid\text{Yes})\ P(\text{Temp(Mild)}\mid\text{Yes})\ *P(\text{Yes})$$

$$= \frac{4}{9} * \frac{4}{9} * \frac{9}{14}$$

$$= 0.124$$

$$P(\text{No}\mid\text{today}) = P(\text{Outlook(Overcast)}\mid\text{No})\ P(\text{Temp(Mild)}\mid\text{No})\ *P(\text{No})$$

$$= \frac{0}{5} * \frac{2}{5} * \frac{5}{14}$$

$$= 0$$

These numbers can be converted into probability by normalization.

PREDICTION BY MODEL

We can play on that day or not?

Now, when we normalize the value, we get:

$$P(Yes) = \frac{P(Yes|today)}{P(Yes|today) + P(No \mid today)}$$

P(Yes)= .13/.13+0=1

$$P(No) = \frac{P(No|today)}{P(Yes|today) + P(No \mid today)}$$

P(No)= 0/.13+0=0

Do It Skills
do it. enjoy it.

TYPES OF NAIVE BAYES CLASSIFIER:

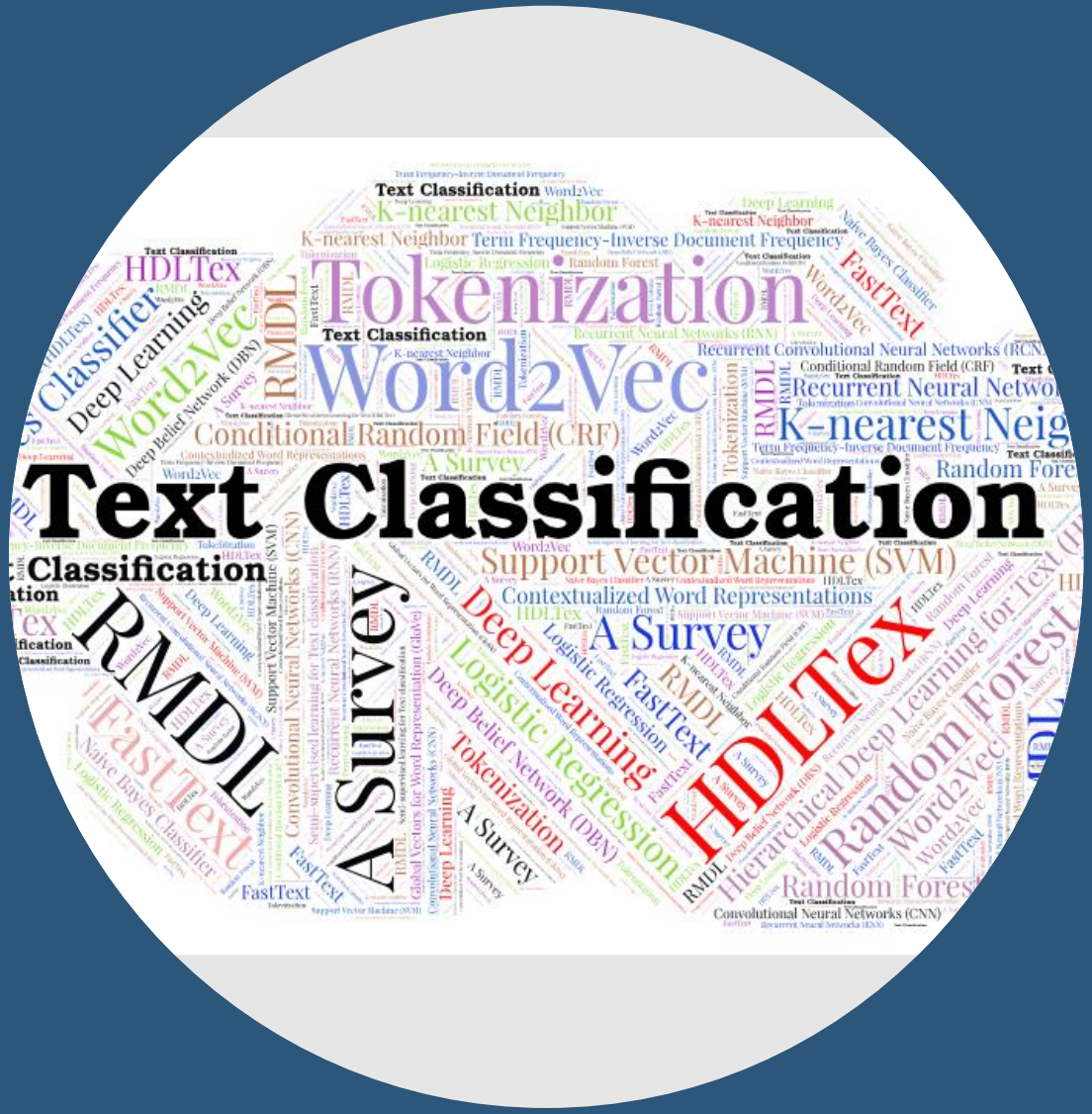
- For continuous features: Gaussian (Normal) distribution
- For discrete features (i.e. word counts): Multinomial distribution
- For binary features (Y/N, True/False, 0/1): Bernoulli distribution

Gaussian Naive Bayes:
When the Features take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution

Multinomial Naive Bayes:
This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

Bernoulli Naive Bayes:
This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

Do It Skills
do it. enjoy it.



TEXT CLASSIFICATION

USING NAIVE BAYES

MULTINOMIAL VS GAUSSIAN NAÏVE BAYES

- For **continuous** data we use Gaussian naïve bayes, e.g: from the Golf example we see that we find likelihood for no of rainy observations given no of yes belongs to rainy.
- For **Discrete** data we use multinomial naïve bayes, e.g.: for text classification we use multinomial as we need to find likelihood probability for each word separately given all word from the class

APPLYING MULTINOMIAL NAÏVE BAYES

If a word from the new sentence does not occur in the class within the training set, the equation becomes zero. To solve this problem, we use Laplace Smoothing.

In order to avoid the problem of *zero* probabilities, an additional smoothing term can be added to the *multinomial Bayes* model. The most common variants of additive smoothing are the so-called *Lidstone smoothing* ($\alpha < 1$) and *Laplace smoothing* ($\alpha = 1$).

$$\hat{P}(x_i \mid \omega_j) = \frac{N_{x_i,\omega_j} + \alpha}{N_{\omega_j} + \alpha\, d} \quad (i = (1, \ldots, d))$$

where

- N_{x_i,ω_j}: Number of times feature x_i appears in samples from class ω_j.
- N_{ω_j}: Total count of all features in class ω_j.
- α: Parameter for additive smoothing.
- d: Dimensionality of the feature vector $\mathbf{x} = [x_1, \ldots, x_d]$.

TEXT DATA

sent	class
This is my book	stmt
They are novels	stmt
have you read this book	question
who is the author	question
what are the characters	question
This is how I bought the book	stmt
I like fictions	stmt
what is your favorite book	question

Let us consider sentence classification to classify a sentence to either ‘question’ or ‘statement’. In this case, there are two classes (“question” and “statement”).

We need to find out if a new sentence, say, ‘what is the price of the book’ is a question or not.

APPLY BAYES THEOREM

We need to find out what is the Probability of class 'Question' given the new sentence and the Probability of class 'Statement' given the new sentence

$$P(\text{Question} \mid \text{What is the price of the book}) = \frac{P(\text{What is the price of the book} \mid \text{Question}) * P(\text{Question})}{P(\text{What is the price of the book})}$$

$$P(\text{Stmt} \mid \text{What is the price of the book}) = \frac{P(\text{What is the price of the book} \mid \text{Stmt}) * P(\text{Stmt})}{P(\text{What is the price of the book})}$$

Now we have to find required likelihoods and prior probabilities

LIKELIHOOD AND PRIOR PROBABILITIES

sent	class
This is my book	stmt
They are novels	stmt
have you read this book	question
who is the author	question
what are the characters	question
This is how I bought the book	stmt
I like fictions	stmt
what is your favorite book	question

$$P(\text{Stmt}) = \frac{\text{Number of sentences in Stmt Class}}{\text{Total number of sentences in the training set}} = \frac{4}{8} = 0.5$$

$$P(\text{Question}) = \frac{\text{Number of sentences in Question Class}}{\text{Total number of sentences in the training set}} = \frac{4}{8} = 0.5$$

Do It Skills
do it. enjoy it.

LIKELIHOOD

P(What is the price of the book | Question) = P(What | Question) x P(is | Question) x P(the | Question) x P(price | Question)
x P(of | Question) x P(the | Question) x P(book | Question)

P(What is the price of the book | Stmt) = P(What | Stmt) x P(is | Stmt) x P(the | Stmt) x P(price | Stmt)
x P(of | Stmt) x P(the | Stmt) x P(book | Stmt)

To find the total number of times a word appears in a class,
we can use CountVectorizer from sklearn.

COUNT OF WORDS

Code to find Frequency of words in each class:
Frequency of words for Statement Class:

```
word_list_s = vec_s.get_feature_names();
count_list_s = X_s.toarray().sum(axis=0)
freq_s = dict(zip(word_list_s,count_list_s))
freq_s
```

```
{'are': 1,
 'book': 2,
 'bought': 1,
 'fictions': 1,
 'how': 1,
 'is': 2,
 'like': 1,
 'my': 1,
 'novels': 1,
 'the': 1,
 'they': 1,
 'this': 2}
```

```
word_list_q = vec_q.get_feature_names();
count_list_q = X_q.toarray().sum(axis=0)
freq_q = dict(zip(word_list_q,count_list_q))
freq_q
```

```
{'are': 1,
 'author': 1,
 'book': 2,
 'characters': 1,
 'favorite': 1,
 'have': 1,
 'is': 2,
 'read': 1,
 'the': 2,
 'this': 1,
 'what': 2,
 'who': 1,
 'you': 1,
 'your': 1}
```

PROBABILITIES OF WORDS IN STATEMENT CLASS

```
prob_s=[]
for word,count in zip(word_list_s,count_list_s):
    prob_s.append(count/len(word_list_s))
dict(zip(word_list_s,prob_s))
```

```
{'are': 0.08333333333333333,
 'book': 0.16666666666666666,
 'bought': 0.08333333333333333,
 'fictions': 0.08333333333333333,
 'how': 0.08333333333333333,
 'is': 0.16666666666666666,
 'like': 0.08333333333333333,
 'my': 0.08333333333333333,
 'novels': 0.08333333333333333,
 'the': 0.08333333333333333,
 'they': 0.08333333333333333,
 'this': 0.16666666666666666}
```

PROBABILITIES OF WORDS IN QUESTION CLASS

```
prob_q=[]
for word,count in zip(word_list_q,count_list_q):
    prob_q.append(count/len(word_list_q))
dict(zip(word_list_q,prob_q))
```

```
{'are': 0.07142857142857142,
 'author': 0.07142857142857142,
 'book': 0.14285714285714285,
 'characters': 0.07142857142857142,
 'favorite': 0.07142857142857142,
 'have': 0.07142857142857142,
 'is': 0.14285714285714285,
 'read': 0.07142857142857142,
 'the': 0.14285714285714285,
 'this': 0.07142857142857142,
 'what': 0.14285714285714285,
 'who': 0.07142857142857142,
 'you': 0.07142857142857142,
 'your': 0.07142857142857142}
```

‘what is the price of the book’

The word ‘price’ did not occur in our training set in either of the classes. So P(price|Stmt) = 0 and P(price|Question) = 0 which will nullify the equation. In short, for the above equation to work , we would need all the words of each new sentence appear in our training set which is not possible.

APPLYING MULTINOMIAL NAÏVE BAYES

If a word from the new sentence does not occur in the class within the training set, the equation becomes zero. To solve this problem, we use Laplace Smoothing.

In order to avoid the problem of *zero* probabilities, an additional smoothing term can be added to the *multinomial Bayes* model. The most common variants of additive smoothing are the so-called *Lidstone smoothing* ($\alpha < 1$) and *Laplace smoothing* ($\alpha = 1$).

$$\hat{P}(x_i \mid \omega_j) = \frac{N_{x_i,\omega_j} + \alpha}{N_{\omega_j} + \alpha\, d} \quad (i = (1, \ldots, d))$$

where

- N_{x_i,ω_j}: Number of times feature x_i appears in samples from class ω_j.
- N_{ω_j}: Total count of all features in class ω_j.
- α: Parameter for additive smoothing.
- d: Dimensionality of the feature vector $\mathbf{x} = [x_1, \ldots, x_d]$.

PROBABILITIES OF WORDS FROM NEW SENTENCE

New Sentence :‘what is the price of the book’

$$P(What\ is\ the\ price\ of\ the\ book|Stmt) = P(What|Stmt) \times P(is|Stmt) \times P(the|Stmt) \times P(price|Stmt) \times P(of|Stmt)\ P(the|Stmt) \times P(book|Stmt)$$

$$P(What\ is\ the\ price\ of\ the\ book|Question) = P(What|Question) \times P(is|Question) \times P(the|Question) \times P(price|Question) \times P(of|Question)\ P(the|Question) \times P(book|Question)$$

Do It Skills
do it. enjoy it.

PROBABILITIES OF WORDS FROM NEW SENTENCE

$$P(What\ is\ the\ price\ of\ the\ book|Stmt) = P(What|Stmt) \times P(is|Stmt) \times P(the|Stmt) \times P(price|Stmt) \times P(of|Stmt)\ P(the|Stmt) \times P(book|Stmt)$$

```
prob_s_with_ls = []
for word in new_word_list:
    if word in freq_s.keys():
        count = freq_s[word]
    else:
        count = 0
    prob_s_with_ls.append((count + 1)/(total_cnts_features_s + total_features))
dict(zip(new_word_list,prob_s_with_ls))
```

```
{'what': 0.02777777777777776,
 'is': 0.08333333333333333,
 'the': 0.05555555555555555,
 'price': 0.02777777777777776,
 'of': 0.02777777777777776,
 'book': 0.08333333333333333}
```

Laplace smoothing ($\alpha = 1$).

$$\hat{P}(x_i \mid \omega_j) = \frac{N_{x_i,\omega_j} + \alpha}{N_{\omega_j} + \alpha\, d} \quad (i = (1, \ldots, d))$$

Do It Skills
do it. enjoy it.

PROBABILITIES OF WORDS FROM NEW SENTENCE

$$P(What\ is\ the\ price\ of\ the\ book|Question) = P(What|Question) \times P(is|Question) \times P(the|Question) \times P(price|Question) \times P(of|Question)\ P(the|Question) \times P(book|Question)$$

```
prob_q_with_ls = []
for word in new_word_list:
    if word in freq_q.keys():
        count = freq_q[word]
    else:
        count = 0
    prob_q_with_ls.append((count + 1)/(total_cnts_features_q + total_features))
dict(zip(new_word_list,prob_q_with_ls))
```

Laplace smoothing ($\alpha = 1$).

$$\hat{P}(x_i \mid \omega_j) = \frac{N_{x_i,\omega_j} + \alpha}{N_{\omega_j} + \alpha\, d} \quad (i = (1, \ldots, d))$$

```
{'what': 0.07692307692307693,
 'is': 0.07692307692307693,
 'the': 0.07692307692307693,
 'price': 0.02564102564102564,
 'of': 0.02564102564102564,
 'book': 0.07692307692307693}
```

Do It Skills
do it. enjoy it.

$$P(What\ is\ the\ price\ of\ the\ book|Stmt) = P(What|Stmt) \times P(is|Stmt) \times P(the|Stmt) \times P(price|Stmt) \times P(of|Stmt)\ P(the|Stmt) \times P(book|Stmt)$$

P(what is the price of the book|stmt) = 0.0277 * 0.0833 * 0.0555 * 0.0277 * 0.0277 * 0.0277 * 0.0555*0.0833
=1.2583314328337572e-11

$$P(What\ is\ the\ price\ of\ the\ book|Question) = P(What|Question) \times P(is|Question) \times P(the|Question) \times P(price|Question) \times P(of|Question)\ P(the|Question) \times P(book|Question)$$

= 0.0769 * 0.0769 * 0.0769*0.0256*0.0256*0.0769*0.0769
=1.7624289971722582e-09

POSTERIOR PROBABILITY

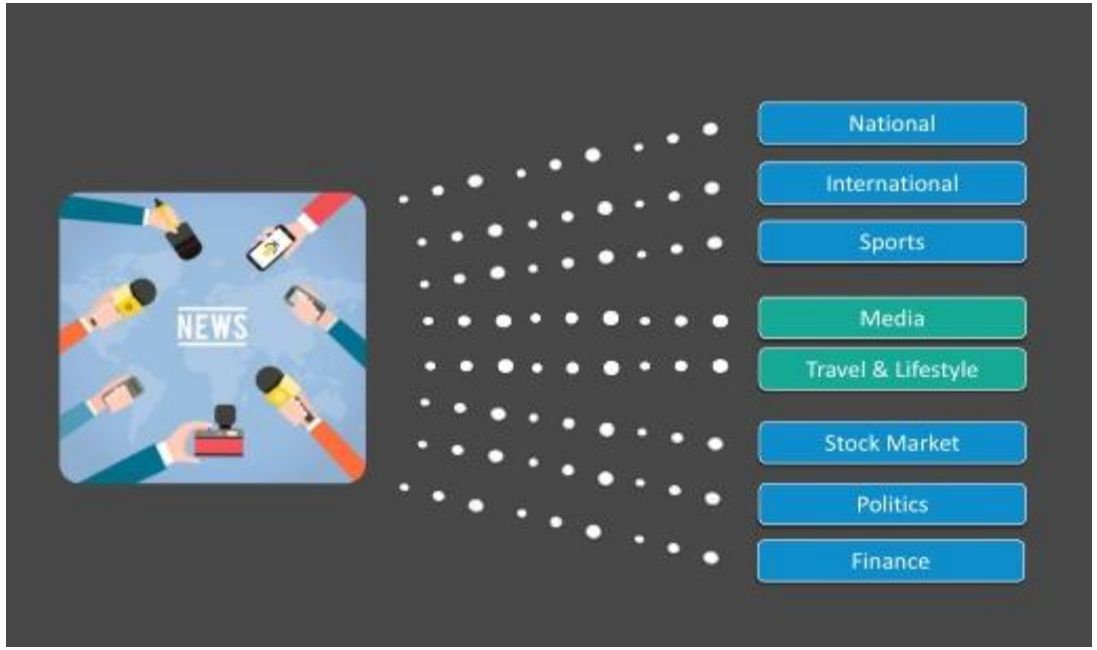
P(Stmt |What is the price of the book) = P(What is the price of the book | Stmt) * P(Stmt)
= 1.2583314328337572e-11 * 0.5
=6.2916571641687 86e-12

P(Question |What is the price of the book) = P(What is the price of the book | Question) * P(Question)
=1.7624289971722582e-09 * 0.5
=8.812144985861291e-10

8.812144985861291e-10 > 6.2916571641687 86e-12
Therefore the new sentence 'What is the price of the book' will be classified as 'Question'

INDUSTRIAL USE OF NAÏVE BAYES ALGORITHM

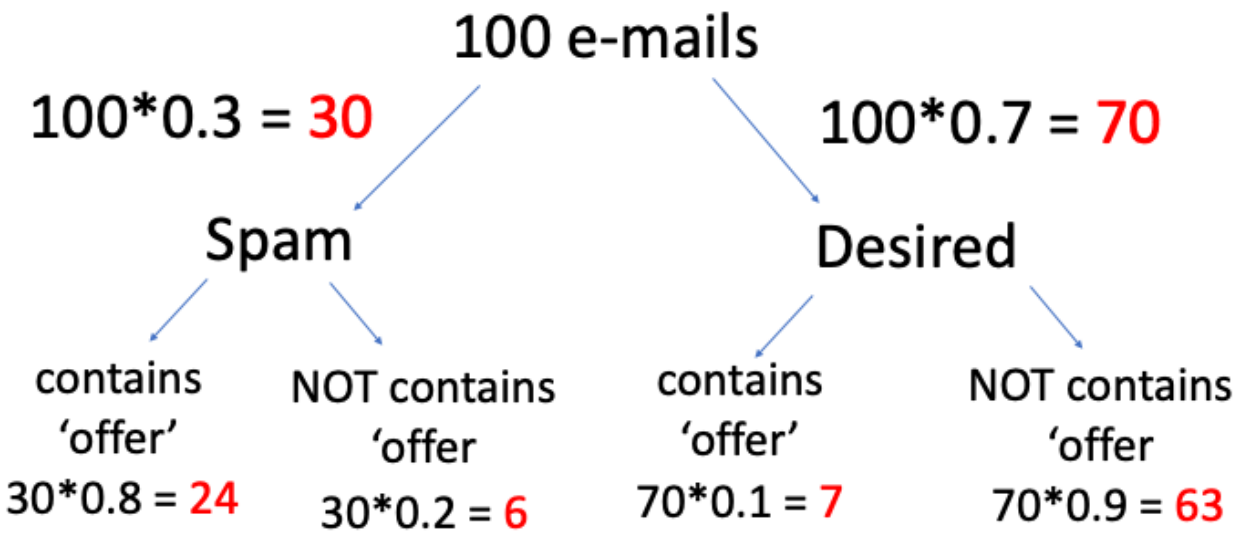
1] NEWS categorization



2] Spam filtering

BAYES THEOREM EXAMPLE 1

The question is that what is the probability of spam where the mail contains the word 'offer':



$P(spam|contains\ offer) =$?

P(contains offer | spam) = 0.8 (given in the question)

P(spam) = 0.3 (given in the question)

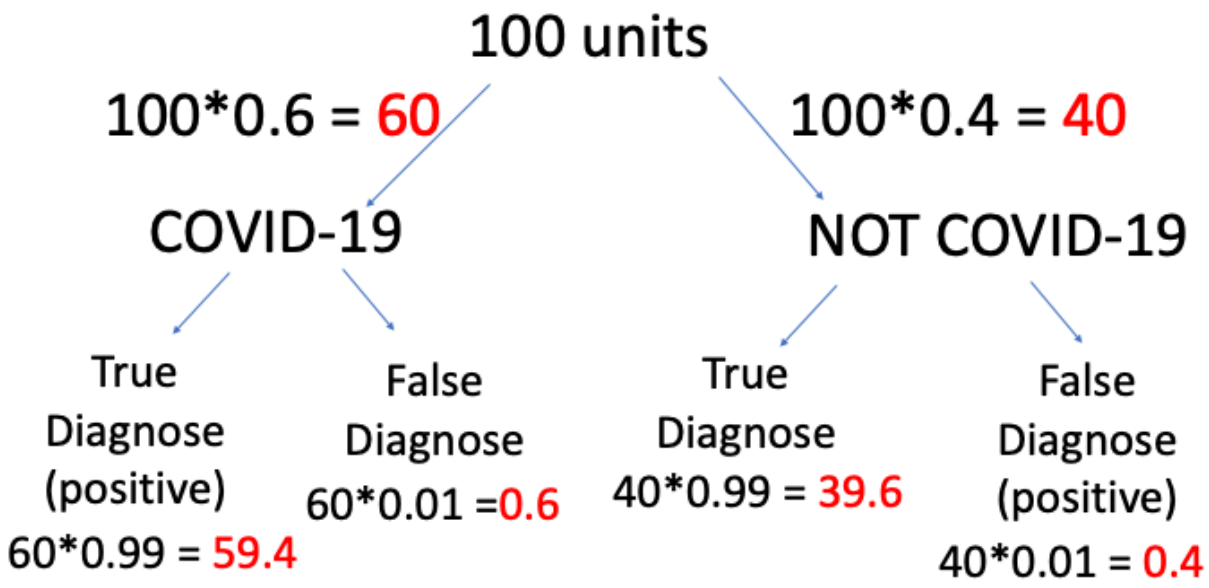
P(contains offer) = 0.3*0.8 + 0.7*0.1 = 0.31
Image for post

$$P(spam|contains\ offer) = \frac{P(contains\ offer|spam) * P(spam)}{P(contains\ offer)}$$

$$P(spam|contains\ offer) = \frac{0.8 * 0.3}{0.31} = 0.774$$

Do It Skills
do it. enjoy it.

BAYES THEOREM EXAMPLE 2



$P(covid19|positive) =$?

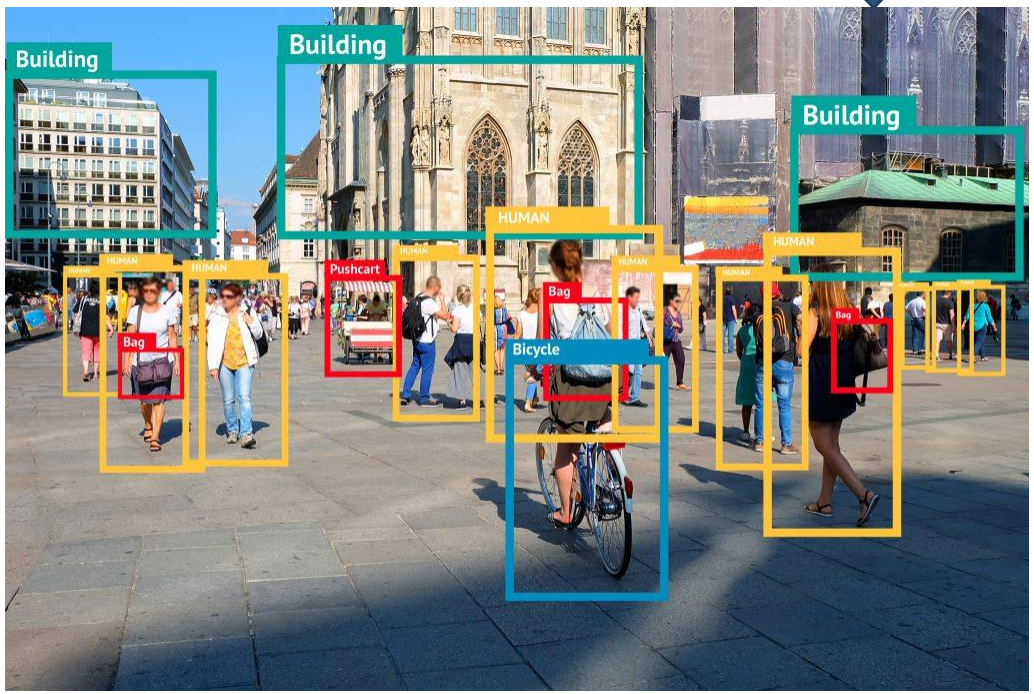
P(positive|covid19) = 0.99
P(covid19) = 0.6
P(positive) = 0.6*0.99+0.4*0.01=0.598

$$P(covid19|positive) = \frac{P(positive|covid19) * P(covid19)}{P(positive)}$$

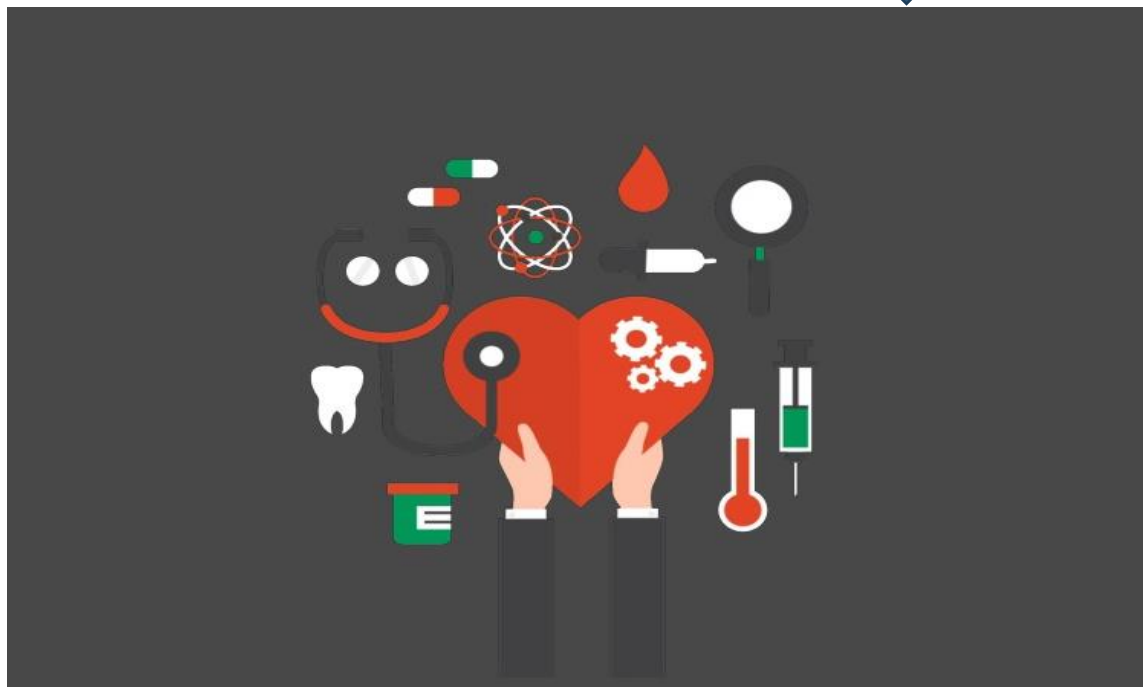
$$P(covid19|positive) = \frac{0.99 * 0.6}{0.598} = 0.993$$

Do It Skills
do it. enjoy it.

3]Object and face detection



4] Medical Diagnosis



SUMMARY: NAIVE BAYES IS NOT SO NAIVE

- Very Fast, low storage requirements
- Robust to Irrelevant Features
- Irrelevant Features cancel each other without affecting results
- Very good in domains with many equally important features
- Decision Trees suffer from *fragmentation* in such cases – especially if little data

- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification

Do It Skills
do it. enjoy it.

THANK YOU

ARUNKG99@GMAIL.COM

WWW.DOITSKILLS.COM