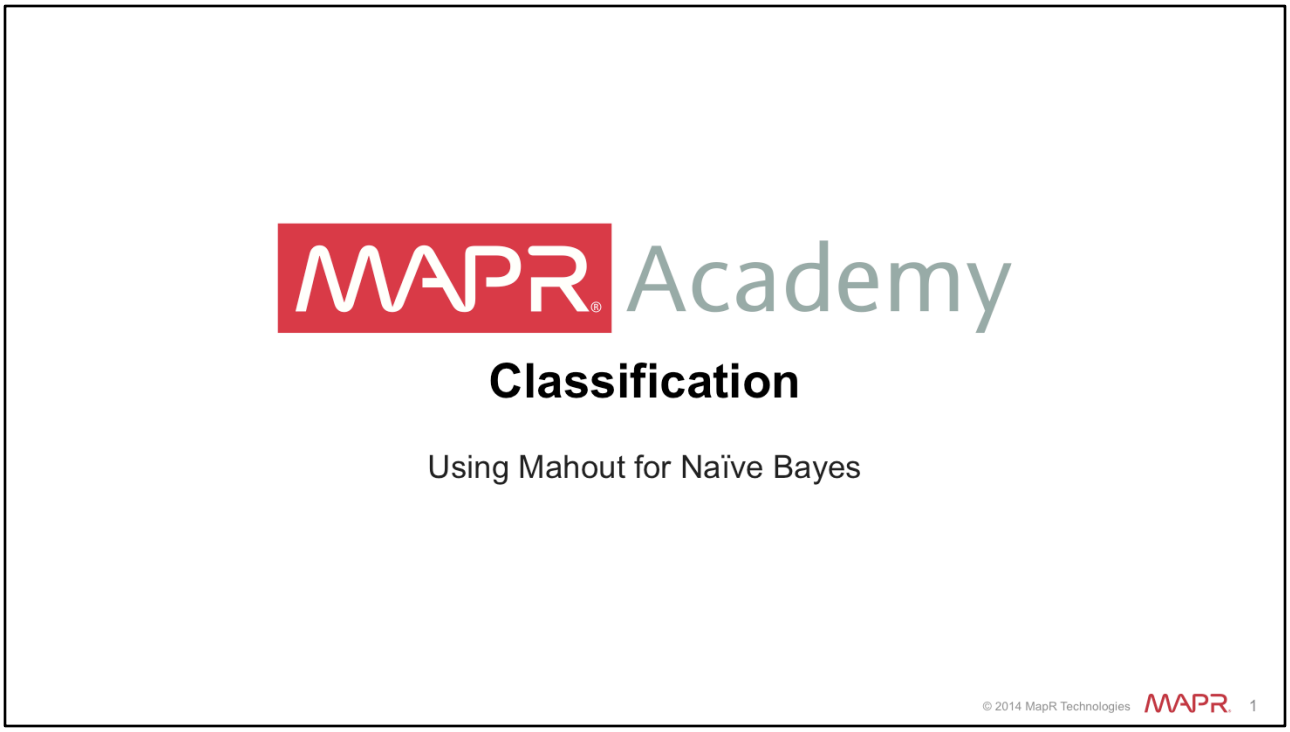
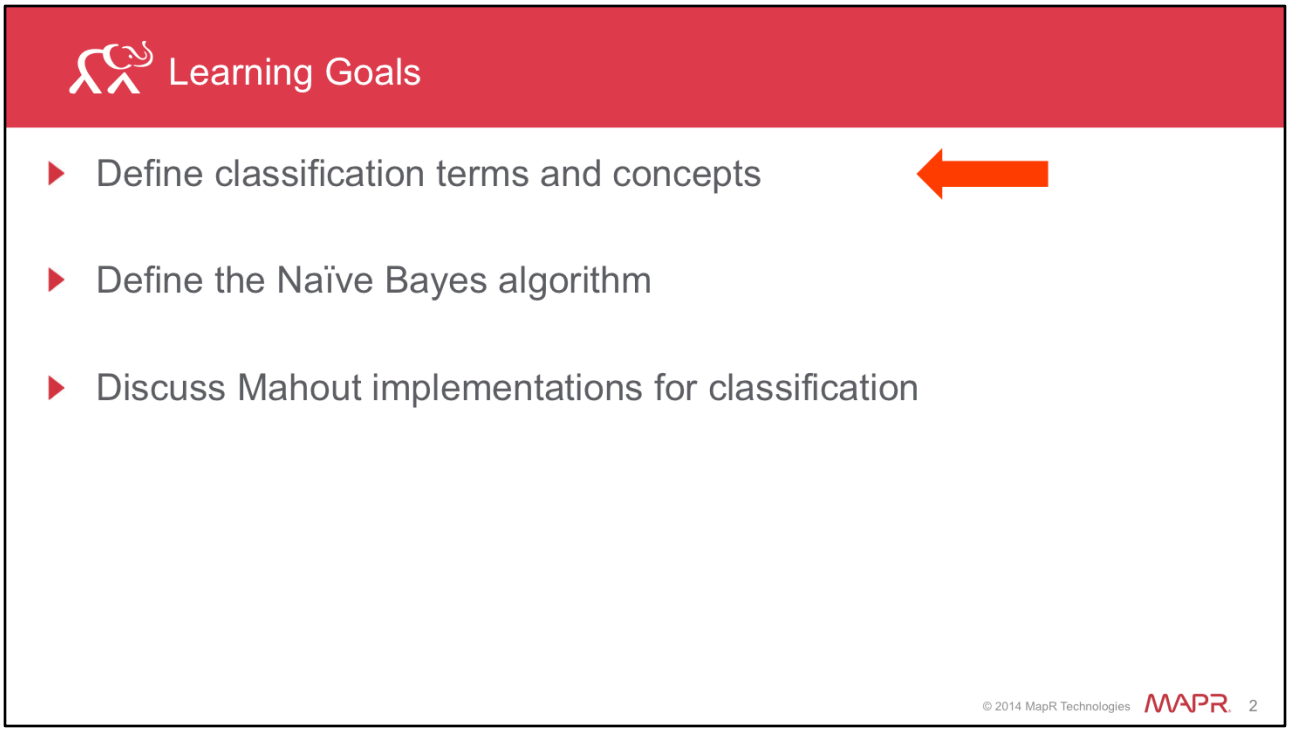
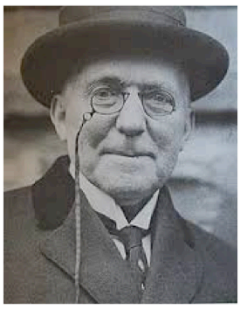


In this lesson, we discuss machine learning classification algorithms.



In this section, we discuss general classification terms and concepts.

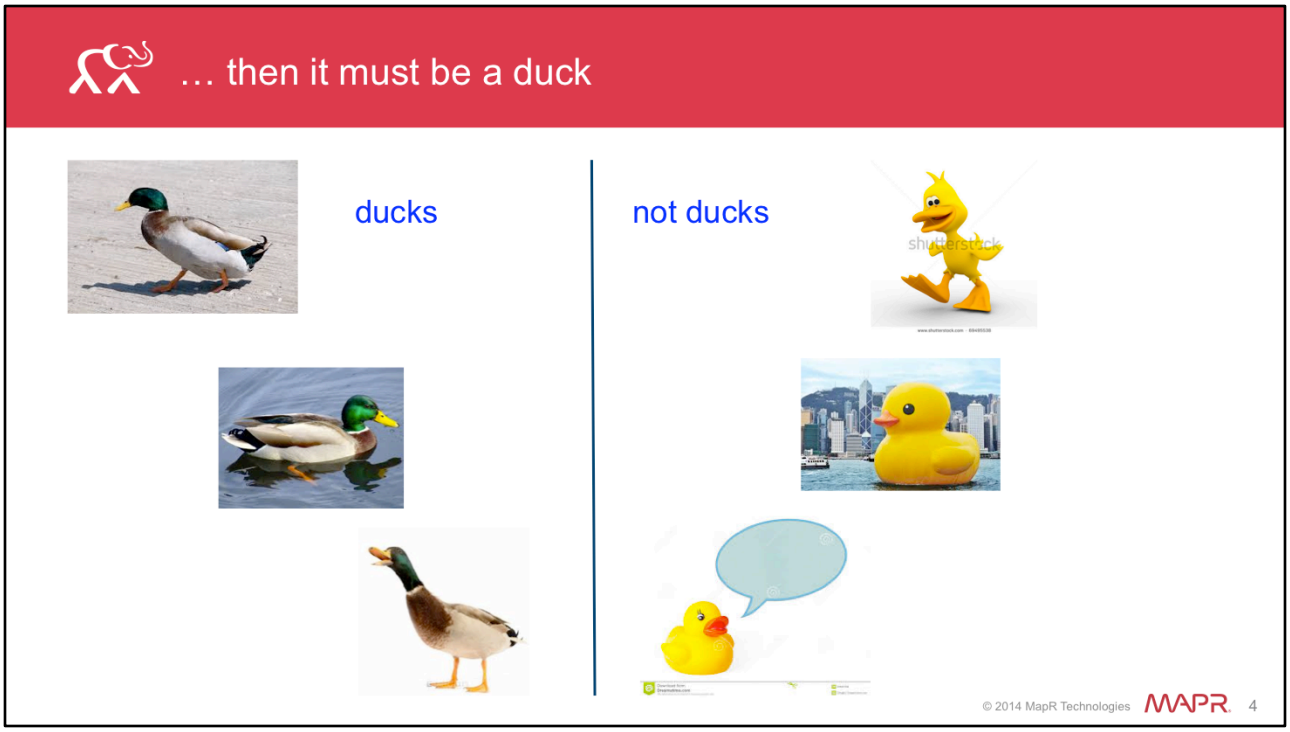
If it walks/swims/quacks like a duck …



"When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck."

Features:
- Walking
- Swimming
- Quacking

James Whitcomb Riley is famously remembered for the quote in the slide. People usually paraphrase this quote to "if it walks like a duck …" to mean you can classify something based on pre-determined "if" conditions.



Based on certain pre-defined criteria, you can classify something as belonging to one class or another. Obviously in this simplistic example, the classification is “binary” but classification can extend to n pre-defined classes.

What is classification

Form of ML that:

- Uses supervised learning algorithms
- Distinguishes one class of objects from another
- Emulates human decision-making

```
boolean isClassificationProblem(questions, answers) {
   if (questions.openEnded() || !answers.categorical())
      return false;
   else
      return maybe;
}
```

 5

Classification is a family of supervised machine learning algorithms that classify input as belonging to one of several pre-defined classes. If the questions you "ask" of a model are open-ended and the answers are not categorical, then you do NOT have a classification problem.

Examples of classification

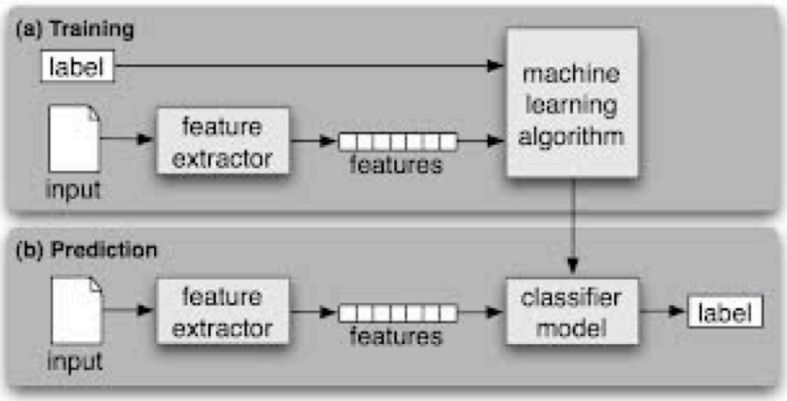
credit card fraud detection

credit approval

credit card profitability prediction

Some common use cases for classification include credit card fraud detection and credit approval (both of which are binary classification problems). The problem of determining how profitable a credit card model is does not qualify as a classification problem (it is moreover a regression problem).

Building and deploying a classifier model



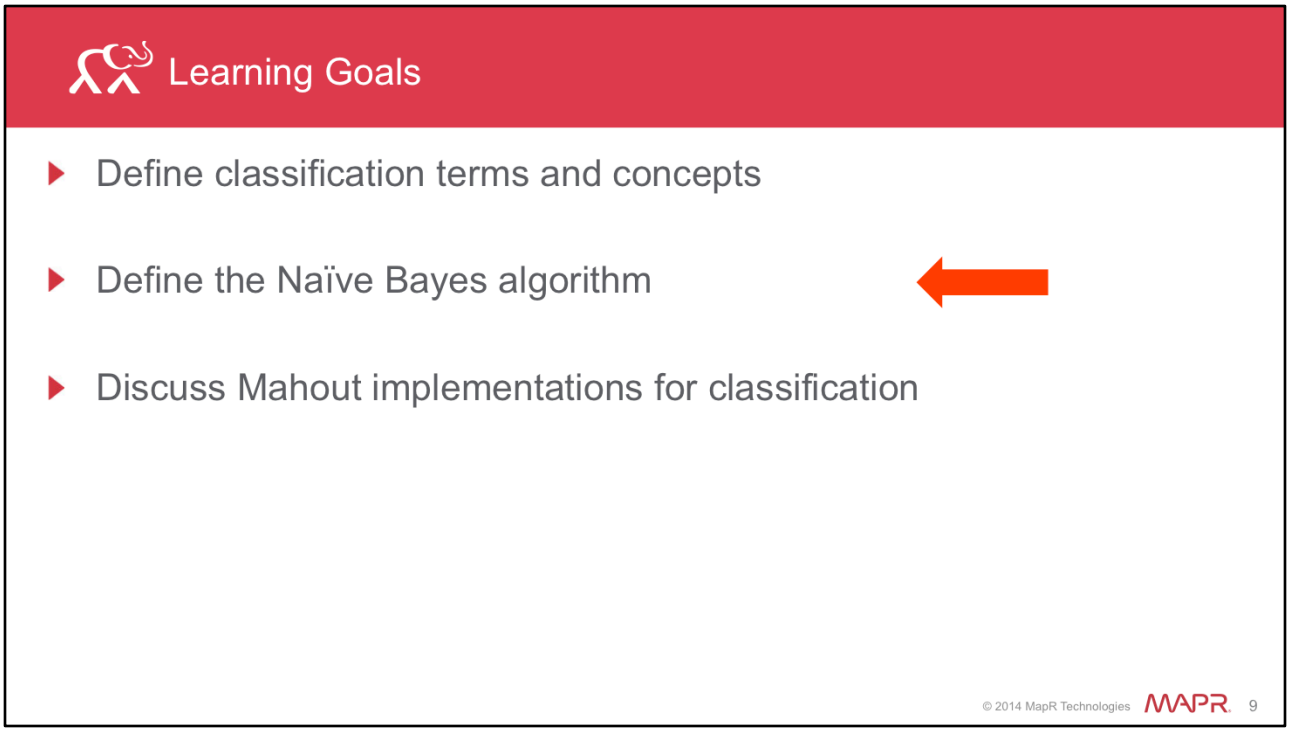
To build a classifier model, you first extract the features of interest that most contribute to the classification. You train your model by making associations between the input features and the labeled output associated with those features.

Then at runtime, you deploy your model by extracting the same set of features and "ask" the model to classify that set of features as one of the pre-defined set of labeled classes.

Predictor variable types

Term	Description	Example
Continuous	Decimal or floating point	0.1
Categorical	Small set of predefined values	{true, false}
Word-like	Large set of predefined values	English dictionary
Text-like	Sequence of word-like	Email message subject

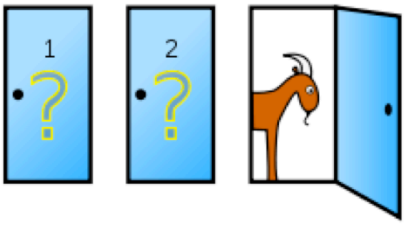
Predictor variables in a classification algorithm may be of different types, as described in the slide above.



In this section, we discuss the naïve bayes classification algorithm.

The Monty Hall Problem

Q: Should you change your door selection on “Let’s Make a Deal”?



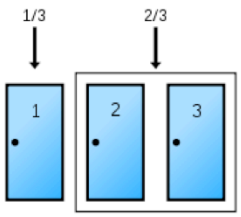
A: YES!!!

 10

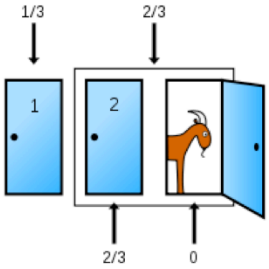
There was a game show called “Let’s Make a Deal” hosted by Monty Hall that aired on American television in the 1970’s. During the show, a contestant is shown 3 doors. Behind 2 of the doors is a goat, and behind one door is a new car. The contestant is asked to select a door, and the game show host (Monty Hall) then opens one of the 2 unselected doors to show a goat. Then Monty always asks the contestant, “do you want to keep your door selection or do you wish to change it?”.

Solution to Monty Hall Problem

visual proof



first choice



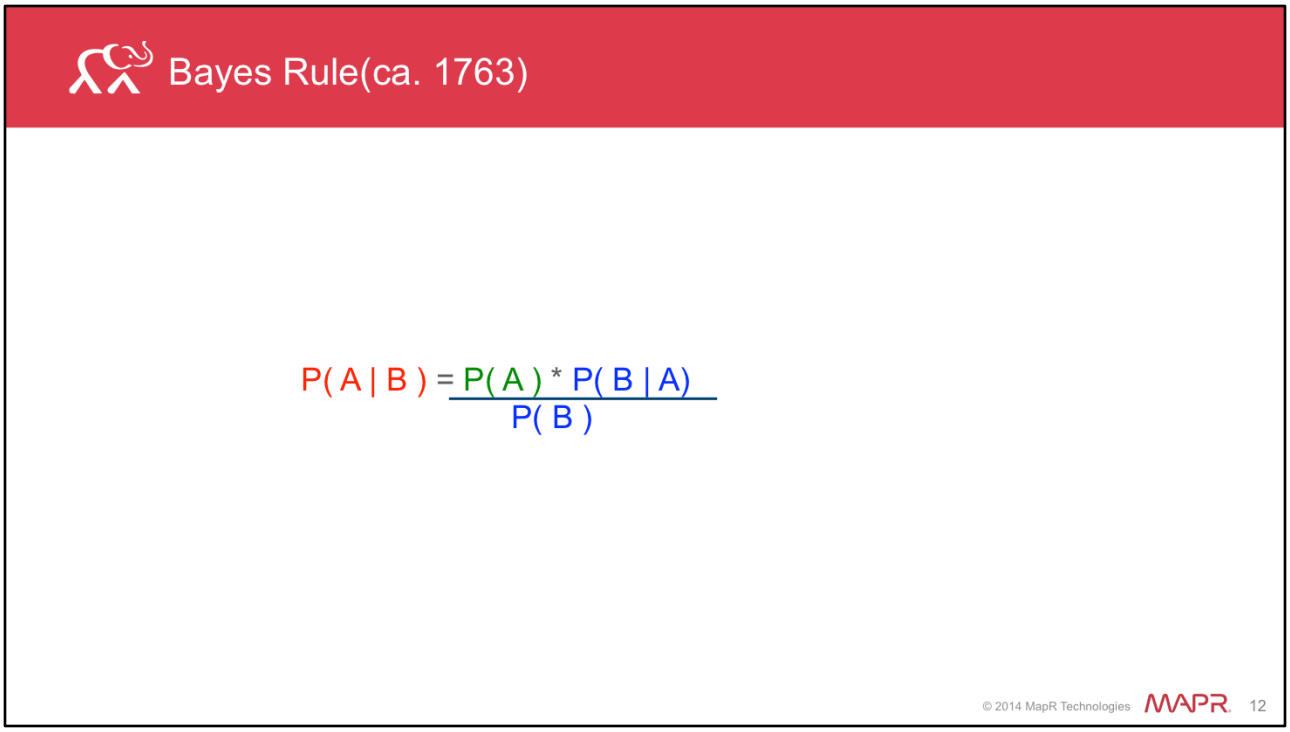
second choice

behind door 1	behind door 2	behind door 3	result if staying at door #1	result if switching to the door offered
Car	Goat	Goat	**Car**	Goat
Goat	**Car**	Goat	Goat	**Car**
Goat	Goat	**Car**	Goat	**Car**

proof by case exhaustion

 11

The answer to this problem is that the contestant should *always* change his/her selection. This is proven in 2 different ways in the slide above. You can also used Bayes Theorem, described in the next slide, to prove that changing your choice is statistically better.



The reverend Thomas Bayes showed statistically how new evidence may influence your belief system. The posterior is defined as a conditional probability (the probability of A given B). It is equal to the quotient of the prior (probability of event A) to the probability of the evidence (quotient of the probability of B given A and the probability of B).

What does Monty Hall have to do with Bayes theorem?

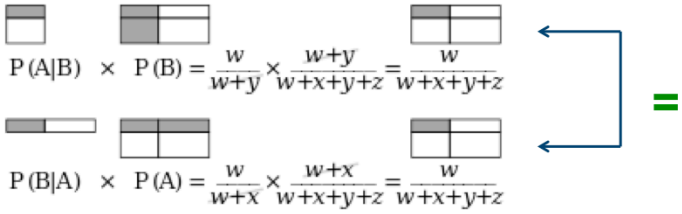
- The Monty Hall problem challenges your door selection based on seeing a goat after you selected your door
- Bayes Theorem challenges your prior belief based on new evidence
- The Monty Hall problem is a party trick
- Bayes Theorem can change your belief system

As an aside, you may wonder what the Monty Hall problem has to do with Bayes theorem. As previously mentioned, you can use Bayes theorem to prove the Monty Hall problem. While the Monty Hall problem is an inconsequential and historical anecdote, Bayes theorem is alive and well in machine learning.

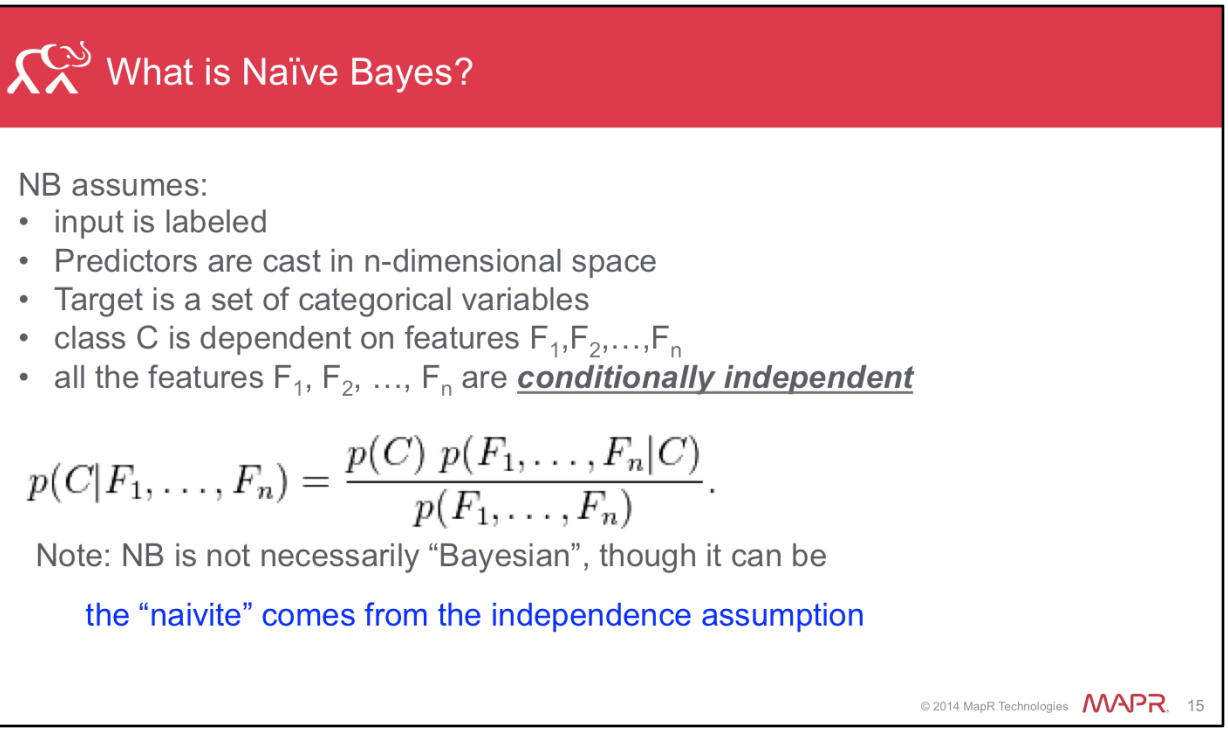
Bayes theorem visual proof

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

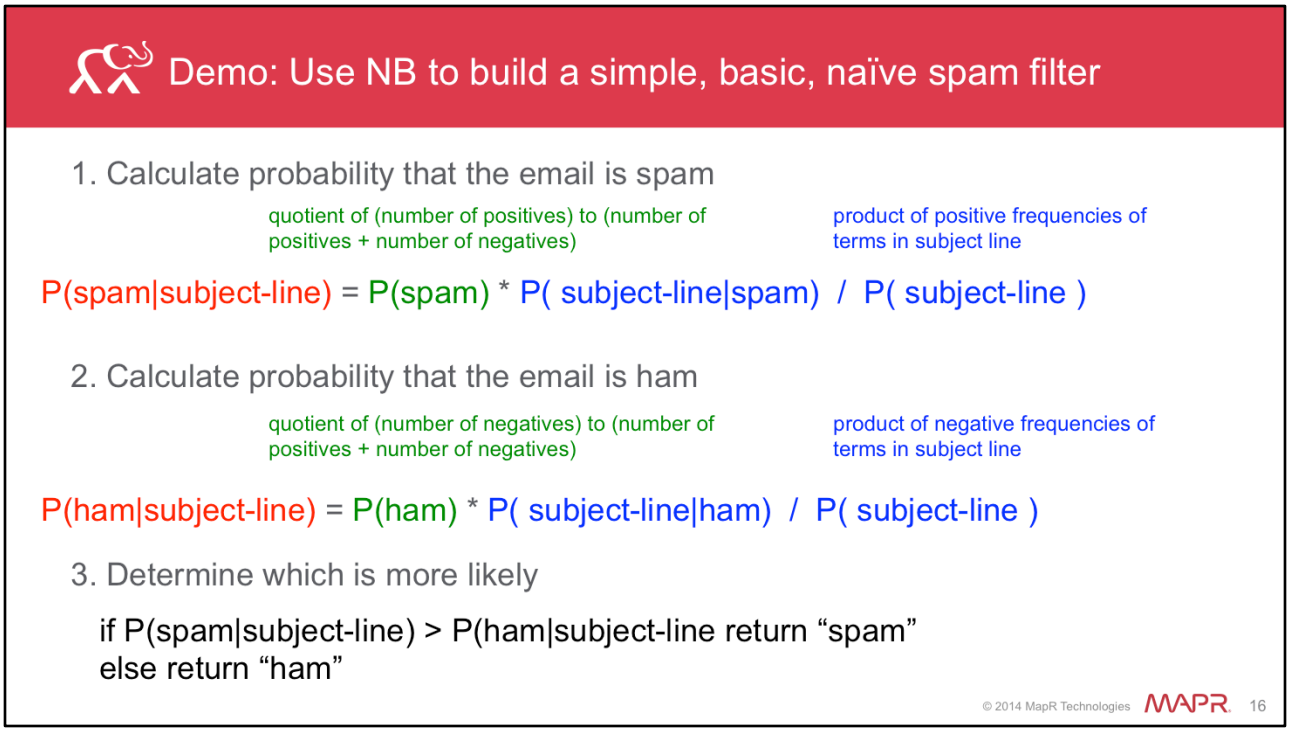
Relative size	Case B	Case $\bar{B}$	Total
Condition A	w	x	$w+x$
Condition $\bar{A}$	y	z	$y+z$
Total	$w+y$	$x+z$	$w+x+y+z$



A visual proof of Bayes theorem is given in the slide above. The variables w, x, y, and z represent the frequencies for A and B, A and not B, not A and B, and not A and not B, respectively. Using these quantities, you can prove the equality of Bayes Theorem.



Naïve Bayes is a machine learning algorithm that is based on Bayes theorem. The crux of Naïve Bayes (the "naïve'" part) is based on the assumption that all the features are independent of each other. Even though this may be too strong an assumption, Naïve Bayes has been shown to work well in several domains.



In this demo, we train and query a Naïve Bayes classifier for detecting spam email. This particular algorithm only considers the contents of the subject line in the model.

Demo: Use NB to build a simple, basic, naïve spam filter

- **Build model**
  - Download labeled data from http://spamassassin.apache.org
  - Extract features (pull out the subject lines from each email)
  - Create hold-out (10%)
  - Vectorize features (bag of words)
  - Calculate priors (probabilities of spam/ham *per term*)
- **Query model**
  - Vectorize input
  - Calculate P(spam|evidence) and P(ham|evidence)
  - Compare the two
  - Pick the bigger probability
- **Test model**

```
cat test-ham-100.txt | java MySpamClassifier DATA/train.txt | grep -c "model predicts not spam"
84

cat test-spam-100.txt | java MySpamClassifier DATA/train.txt | grep -c "model predicts spam"
75
```

 17

This particular model was built as described in the slide above. Again, the only data used to train and use the model is the subject lines of the emails. When testing this particular NB model, we find it gets 84% correct when testing non-spam emails and 75% correct when testing spam emails. That's pretty good considering such a simple algorithm, only using subject lines as input data, and training on such a small data set.

This particular NB program takes as input a line of text which represents the subject line of an email and outputs information regarding whether the model considers it spam or not. Interestingly, the string “vote now” is not considered spam while the same subject line with an exclamation point is considered spam. This is due to the fact that many of the spam emails in the training corpus contain the exclamation mark in their subject line.

Paper and pencil example

Positive movie reviews
"I loved the movie"
"A great movie. Good acting"
"great acting. Movie is great!"
"Great movie!"

Negative movie reviews
"I hated the movie"
"This movie is bad"
"Bad acting"

Unique terms (vocabulary)
[I, loved, the, movie, a, great, is, good, acting, hated, this, bad]

P(positive) = 4/7

P(I|positive) = 2/21
P(loved|positive) = 2/21
P(the|positive) = 2/21
P(movie|positive) = 5/21
P(a|positive) = 2/21
P(great|positive) = 5/21
P(good|positive) = 2/21
P(acting|positive)=3/21
P(is|positive)=2/21

P(negative) = 3/7

P(I|negative) = 2/20
P(hated|negative) = 2/20
P(the|negative) = 2/20
P(movie|negative) = 2/20
P(this|negative) = 2/20
P(bad|negative) = 3/20
P(acting|negative) = 2/20
P(is|negative) = 2/20

$$P(term|class) = \frac{1 + |term\ in\ class|}{(|total\ terms\ in\ class| + |vocabulary|}$$

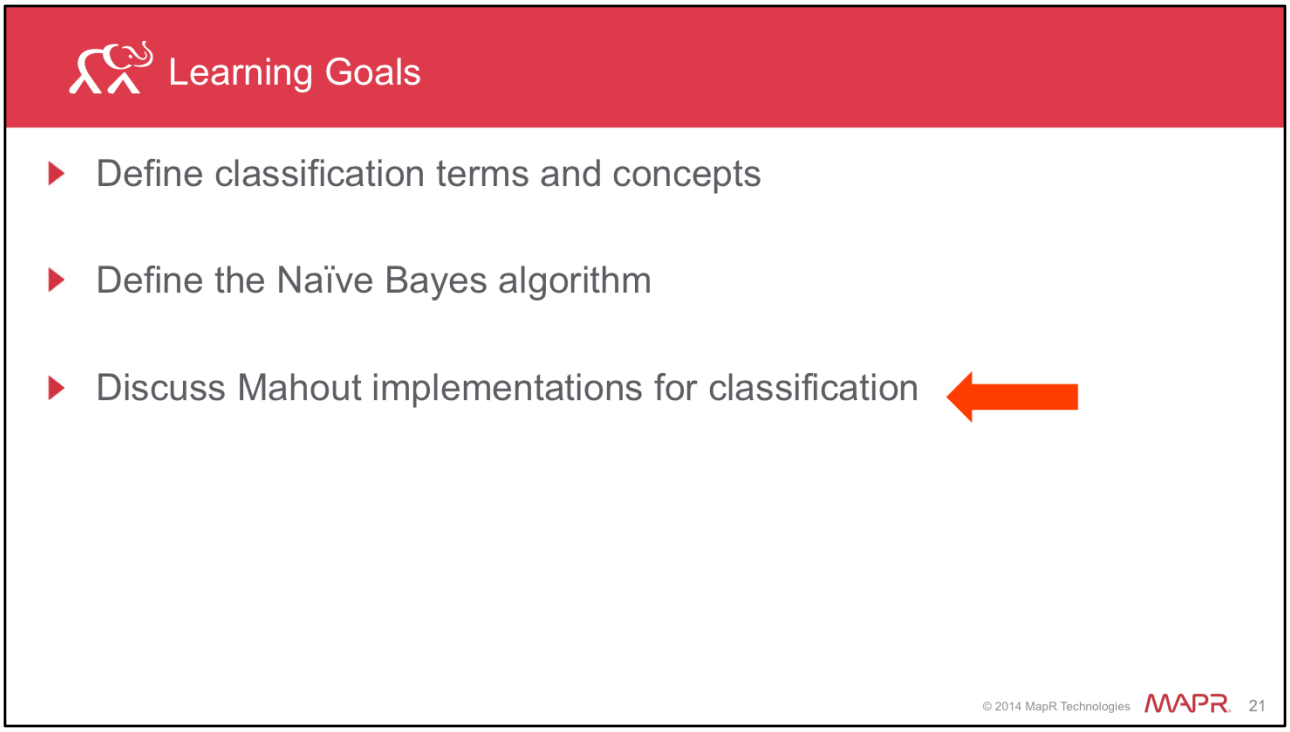
Paper and pencil example

Is “the acting is bad” a positive or negative review?

P(positive|”the acting is bad”) = P(positive) x P(the|positive) x P(acting|positive) x P(is|positive) x P(bad|positive)
= 4/7 x 2/21 x 3/21 x 2/21 x ***1/21***
= **48** / 7x21x21x21x21

P(negative|”the acting is bad”) = P(negative) x P(the|negative) x P(acting|negative) x P(is|negative) x P(bad|negative)
= 3/7 x 2/20 x 2/20 x 2/20 x 3/20
= **72** / 7x20x20x20x20

72/(7x20^4) > 48/(7x21^4)
so we conclude that the review “the acting is bad” is a negative review.



In this last section, we discuss the Mahout implementations for classification.

Using Mahout Naïve Bayes from CLI (demo)

```
mahout seqdirectory

mahout seq2sparse

mahout split

mahout trainnb

mahout testnb
```


Using Mahout for Naïve Bayes (discussion)

- Confusion matrix
- Accuracy
- Kappa