

# Supervised Learning

## POS6933: Computational Social Science

Jake S. Truscott, Ph.D

University of Florida  
Spring 2026



## Overview

- Supervised Classification
  - Naive Bayes

## Supervised Classification

- **Supervised Learning:** A statistical (or ML) approach in which a model is trained on a set of texts that have been labeled **in advance**, so that it can learn the relationship between textual features and known outcomes and then predict those outcomes for new, unseen texts.

## Supervised Classification

- **Supervised Learning:** A statistical (or ML) approach in which a model is trained on a set of texts that have been labeled **in advance**, so that it can learn the relationship between textual features and known outcomes and then predict those outcomes for new, unseen texts.
  - **Goal:** Prediction of a known label, not general language representation or word scoring

## Supervised Classification

- **Supervised Learning:** A statistical (or ML) approach in which a model is trained on a set of texts that have been labeled **in advance**, so that it can learn the relationship between textual features and known outcomes and then predict those outcomes for new, unseen texts.
  - **Goal:** Prediction of a known label, not general language representation or word scoring
  - Relationship b/w text features and outcomes (classes/labels) is **learned from labeled data**, rather than fixed invariably by dictionaries.

## Supervised Classification (Compared to Earlier Stuff)

- **Dictionaries:** I'm going to create a dictionary with classes assigned to words, where the scores I derive are the result of locating and identifying which features from the different classes are found in a document.

Ex: A string can best be labeled as *positive* because more words from that class appear than words in the *negative* class.

## Supervised Classification (Compared to Earlier Stuff)

- **Dictionaries:** I'm going to create a dictionary with classes assigned to words, where the scores I derive are the result of locating and identifying which features from the different classes are found in a document.  
Ex: A string can best be labeled as *positive* because more words from that class appear than words in the *negative* class.
- **MLM and VSM:** I'm able to make probabilistic assessments based on the features of a document  
Ex: Madison most likely wrote *Federalist 51* because of the lexical variance in that document versus Madison's broader profile

## Supervised Classification (Compared to Earlier Stuff – Cont.)

- **Supervised Learning:** I'm able to use (perhaps) a small portion of the available data, assign labels (classes), then use that data to train a model for a task with unseen (testing) data (i.e., the larger volume of remaining data)

Ex: If I want to understand how members of the Senate Judiciary Committee communicate with nominees to the US Supreme Court or whether social media posts are relaying positive sentiments with respect to certain policy areas, I can label a small portion of that data to train a model/structure an algorithm for assessing the larger (remaining) set of data – rather than having to label it by hand.

## Supervised Classification (Compared to Earlier Stuff – Cont.)

- **Supervised Learning:** I'm able to use (perhaps) a small portion of the available data, assign labels (classes), then use that data to train a model for a task with unseen (testing) data (i.e., the larger volume of remaining data)  
Ex: If I want to understand how members of the Senate Judiciary Committee communicate with nominees to the US Supreme Court or whether social media posts are relaying positive sentiments with respect to certain policy areas, I can label a small portion of that data to train a model/structure an algorithm for assessing the larger (remaining) set of data – rather than having to label it by hand.
- Trade-off: Incredibly flexible, theory-driven, and training based (rather than rules based), **but** reliable models require sufficient training.

## Supervised Classification – Building a Training Set

- Imagine you have a dataset with 10,000 observations.
  - From that 10,000 – partition an 80/20 split, such that a random subset of 2,000 are removed to develop a training set, while those remaining will constitute the unseen (*testing*) set.
  - With that 2,000 – assign a binary or multiclass label that you contend best represents its features
  - Use that labeled data to train a classifier – we will discuss **Naive Bayes** in a moment.

## Supervised Classification – Building a Training Set (Cont.)

- To make sure training set is representative, draw from a (stratified) random sample. Generally speaking, a 70/30 or 80/20 split seems to be the most common – though validation will be necessary always!

## Supervised Classification – Building a Training Set (Cont.)

- To make sure training set is representative, draw from a (stratified) random sample. Generally speaking, a 70/30 or 80/20 split seems to be the most common – though validation will be necessary always!
- In the realm of validation, it's best practice to do so with more than one perspective – i.e., incorporate one (or more) individuals to validate both your labels and the model results from that training

Ex: Parrots Paper

## Supervised Classification – Building a Training Set (Cont.)

- To make sure training set is representative, draw from a (stratified) random sample. Generally speaking, a 70/30 or 80/20 split seems to be the most common – though validation will be necessary always!
- In the realm of validation, it's best practice to do so with more than one perspective – i.e., incorporate one (or more) individuals to validate both your labels and the model results from that training
  - Ex: Parrots Paper
- Training sets need to retain: objective-intersubjectivity, an a priori design, reliability, validity, and replicability.
  - Ex: If you say term (n-gram, string, document, etc.) is  $x$ , it must be  $x$  – not  $x'$ !

## Bayes' Rule of Conditional Probability

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad \text{Bayes Rule}$$

- The probability of  $A$  given  $B$  equals the probability of observing  $B$  if  $A$  were true, multiplied by the prior probability of  $A$ , and divided by the overall probability of observing  $B$ .

# Naive Bayes – Canonical Supervised Learning Model

- **Naive Bayes**: leverages Bayes' rule of conditional probability to recover probability that an unseen document ( $D_i$ ) containing words ( $w_{i1}, w_{ij}$ ) belongs to a certain classification ( $\pi_{ik}$ ), where  $\pi_{ik}$  is represented as 0 or 1 given the membership of  $D_i$  in class  $k$ .

## Naive Bayes – Canonical Supervised Learning Model

- **Naive Bayes:** leverages Bayes' rule of conditional probability to recover probability that an unseen document ( $D_i$ ) containing words ( $w_{i1}, w_{ij}$ ) belongs to a certain classification ( $\pi_{ik}$ ), where  $\pi_{ik}$  is represented as 0 or 1 given the membership of  $D_i$  in class  $k$ .
  - **Conditional Independence (Naive) Assumption:** Conditional on class, words are independent of each other – i.e., once you know the class of any document, learning that one appears in it tells you nothing additional about whether another word appears. This is almost certainly wrong (*More Later...*) – but assumption makes this tractable.

# Naive Bayes – Canonical Supervised Learning Model

- **Naive Bayes:** leverages Bayes' rule of conditional probability to recover probability that an unseen document ( $D_i$ ) containing words ( $w_{i1}, w_{ij}$ ) belongs to a certain classification ( $\pi_{ik}$ ), where  $\pi_{ik}$  is represented as 0 or 1 given the membership of  $D_i$  in class  $k$ .
  - **Conditional Independence (Naive) Assumption:** Conditional on class, words are independent of each other – i.e., once you know the class of any document, learning that one appears in it tells you nothing additional about whether another word appears. This is almost certainty wrong (*More Later...*) – but assumption makes this tractable.
  - *In Short...:* The probability that a document (sentence, string, etc.) is a certain classification is conditioned on Bayes' Rule of conditional probability, where prior probabilities are set by the training data

# Naive Bayes – Canonical Supervised Learning Model

- **Naive Bayes:** leverages Bayes' rule of conditional probability to recover probability that an unseen document ( $D_i$ ) containing words ( $w_{i1}, w_{ij}$ ) belongs to a certain classification ( $\pi_{ik}$ ), where  $\pi_{ik}$  is represented as 0 or 1 given the membership of  $D_i$  in class  $k$ .
  - **Conditional Independence (Naive) Assumption:** Conditional on class, words are independent of each other – i.e., once you know the class of any document, learning that one appears in it tells you nothing additional about whether another word appears. This is almost certainty wrong (*More Later...*) – but assumption makes this tractable.
  - *In Short...:* The probability that a document (sentence, string, etc.) is a certain classification is conditioned on Bayes' Rule of conditional probability, where prior probabilities are set by the training data

## Naive Bayes – Walkthrough

$$p(\pi_{ik} \mid D_i) = \frac{p(\pi_{ik} = 1)p(D_i \mid \pi_{ik} = 1)}{p(D_i)} \quad \text{Bayes' Rule – Conditional Probability}$$

$p(\pi_{ik} = 1)$  Baseline Probability that  $D_i$  Belongs to Class  $k$  Before Reading It

$p(D_i \mid \pi_{ik} = 1)$  Probability of Observing  $w_{i1} \dots w_{ij}$  if  $D_i$  in Class  $k$

$p(D_i)$  Probability of Observing  $D_i$  in Any Class – Will Use to Normalize At End

## Naive Bayes – Walkthrough (Cont.)

$$W_i \mid \pi_{ik} = 1 \sim \text{Multinomial}\left(\sum_j W_{ij}, \mu_k\right)$$

Conditional on class  $k$  ( $=1$ ), the words in document  $i$  are drawn from a multinomial distribution with the total word count equal to the document length and word probabilities  $\mu_k$  estimated from the training data.

$$p(W_i \mid \pi_{ik} = 1) \propto \prod_{j=1}^j \mu_{kj}^{W_{ij}}$$

Probability of document  $i$  being class  $k$  is thus proportional to product of probabilities estimated from the training data for each word and raised to their actual appearance in document  $i$

## Naive Bayes – Walkthrough (Cont.)

$$\hat{\mu}_{kj} = \frac{c + \sum_i^N \pi_{ik} W_{ij}}{Jc + \sum_i \sum_j \pi_{ik} W_{ij}}$$

Word probabilities for class  $k$  are estimated as the count of word  $j$  across all documents in class  $k$  divided by the total word count in class  $k$ , with  $c$  as a smoothing constant (e.g., *Laplace smoothing*) to avoid zero probabilities

**Basically:** The probability that a word belongs to a class  $k$  (versus  $k'$ ) is determined by how often it appears in training documents labeled as  $k$  (versus  $k'$ ).

## Naive Bayes – Putting it All Together

$$p(\pi_{ik} = 1 \mid W_i) \propto \frac{\sum_i^N I(y_i = k)}{N} \prod_{j=1}^j \mu_{kj}^{ij}$$

- *Simplified*: The probability that document  $i$  belongs in class  $k$  is proportional to the class's prior frequency in the training data multiplied by the product of the probabilities of each word in the document given that class.
  - *Or...* The probability that a document belongs to class  $k$  is proportional to the class's prior probability and the likelihood of the document's observed features given that class.

## Naive Bayes – Example (News Coverage)

Training Set:

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

Class ( $k$ ) Priors:  $\frac{\sum_i^N I(y_i=k)}{N} =$  The total number of documents with class  $k$  over the total number of documents in the training set.

## Naive Bayes – Example (News Coverage)

Training Set:

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

Class ( $k$ ) Priors:  $\frac{\sum_i^N I(y_i=k)}{N} =$  The total number of documents with class  $k$  over the total number of documents in the training set.

- $\frac{1}{3}$  Politics
- $\frac{2}{3}$  Sports

## Naive Bayes Example – Calculate $\mu_{kj}$

- Like MLM, our next step is to calculate  $\mu$  for each word but add a layer for each class label ( $\mu_{kj}$ ).

Naive Bayes Example – Calculate  $\mu_{kj}$ 

- Like MLM, our next step is to calculate  $\mu$  for each word but add a layer for each class label ( $\mu_{kj}$ ).
- Because we have a small vocabulary (Game, Team, Vote, J = 3), there are instances where some words don't appear for a class (e.g., no Game or Team in *Politics*) – So we're going to add Laplace smoothing ( $c$ ) to avoid zero probabilities.

$$\hat{\mu}_{kj} = \frac{c + \sum_i^N \pi_{ik} W_{ij}}{Jc + \sum_i \sum_j \pi_{ik} W_{ij}}$$

Naive Bayes Example – Calculate  $\mu_{kj}$  (Cont.)

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

- Ex:  $\hat{\mu}_{kj} = \frac{c + \sum_i^N \pi_{ik} W_{ij}}{Jc + \sum_i \sum_j \pi_{ik} W_{ij}}$        $\mu_{Sports, Game} = \frac{1_{(c)} + Game_{(3)}}{3_{(J)} \cdot 1_{(c)} + Game_{(3)} + Team_{(3)} + Vote_{(0)}} = \frac{4}{9} \approx 0.444$

Naive Bayes Example – Calculate  $\mu_{kj}$  (Cont.)

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

- Your Turn:  $\mu_{Sports, Team}$

Naive Bayes Example – Calculate  $\mu_{kj}$  (Cont.)

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

- **Your Turn:**  $\mu_{Sports, Team}$
- $\mu_{Sports, Team} = \frac{1+3}{3+6} \approx 0.444$

Naive Bayes Example – Calculate  $\mu_{kj}$  (Cont.)

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

- **Your Turn:**  $\mu_{Sports, Team}$
- $\mu_{Sports, Team} = \frac{1+3}{3+6} \approx 0.444$
- $\mu_{Sports, Vote} = \frac{1+0}{3+6} \approx 0.111$

Naive Bayes Example – Calculate  $\mu_{kj}$  (Cont.)

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

- **Your Turn:**  $\mu_{Politics, Game}$

Naive Bayes Example – Calculate  $\mu_{kj}$  (Cont.)

Document	Class	Words (Counts)
1	Sports	Game (2), Team (1), Vote (0)
2	Sports	Game (1), Team (2), Vote (0)
3	Politics	Game (0), Team (0), Vote (3)

- **Your Turn:**  $\mu_{Politics, Game}$
- $\mu_{Politics, Game} = \frac{1+0}{3+3} \approx 0.167$
- $\mu_{Politics, Team} = \frac{1+0}{3+3} \approx 0.167$
- $\mu_{Politics, Vote} = \frac{1+3}{3+3} \approx 0.667$

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document

- Now that we've trained our classifier, let's test it on a new (unseen) document:  
 $D_{New} = (Game, Team, Vote)$

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document

- Now that we've trained our classifier, let's test it on a new (unseen) document:  
 $D_{New} = (\text{Game}, \text{Team}, \text{Vote})$
- Using Naive Bayes, what is the probability that it belongs to either class (*Sports*, *Politics*)?

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document

- Now that we've trained our classifier, let's test it on a new (unseen) document:  
 $D_{New} = (\text{Game}, \text{Team}, \text{Vote})$
- Using Naive Bayes, what is the probability that it belongs to either class (*Sports*, *Politics*)?

$$p(\text{Sports} | D_{New}) \propto p(\text{Sports}) \times \mu_{\text{Sports}, \text{Game}} \times \mu_{\text{Sports}, \text{Team}} \times \mu_{\text{Sports}, \text{Vote}}$$

$$\frac{2}{3} \times 0.444 \times 0.444 \times 0.111 \approx 0.0145$$

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document

- Now that we've trained our classifier, let's test it on a new (unseen) document:  
 $D_{New} = (\text{Game}, \text{Team}, \text{Vote})$
- Using Naive Bayes, what is the probability that it belongs to either class (*Sports*, *Politics*)?

$$p(\text{Sports} | D_{New}) \propto p(\text{Sports}) \times \mu_{\text{Sports}, \text{Game}} \times \mu_{\text{Sports}, \text{Team}} \times \mu_{\text{Sports}, \text{Vote}}$$

$$\frac{2}{3} \times 0.444 \times 0.444 \times 0.111 \approx 0.0145$$

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document (Cont.)

- **Your Turn:** What about  $p(\text{Politics} \mid D_{\text{new}})$ ?

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document (Cont.)

- **Your Turn:** What about  $p(Politics | D_{new})$ ?

$$p(Politics | D_{New}) \propto p(Politics) \times \mu_{Politics, Game} \times \mu_{Politics, Team} \times \mu_{Politics, Vote}$$

$$\frac{1}{3} \times 0.167 \times 0.167 \times 0.667 \approx 0.0062$$

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document (Cont.)

- After normalizing, we get:

$$p(Sport|D_{New}) = \frac{0.0145}{0.0145 + 0.0062} \approx 0.70$$

$$p(Politics|D_{New}) = \frac{0.0062}{0.0145 + 0.0062} \approx 0.299$$

## Naive Bayes Example – Putting Together: Testing on New (Unseen) Document (Cont.)

- After normalizing, we get:

$$p(\text{Sport} | D_{\text{New}}) = \frac{0.0145}{0.0145 + 0.0062} \approx 0.70$$

$$p(\text{Politics} | D_{\text{New}}) = \frac{0.0062}{0.0145 + 0.0062} \approx 0.299$$

Although the non-normalized probabilities are (very) small, we can improve our chances with more training!

Naive Bayes Example – Another Example  $D_{New} = (Vote, Team, Team)$

Naive Bayes Example – Another Example  $D_{New} = (Vote, Team, Team)$

$$p(Sports | D_{New}) \propto p(Politics) \times \mu_{Sports, Vote} \times \mu_{Sports, Team} \times \mu_{Sports, Team}$$

$$p(Sports | D_{New}) = \frac{2}{3} \times 0.111 \times 0.444 \times 0.444 \approx 0.014$$

Naive Bayes Example – Another Example  $D_{New} = (Vote, Team, Team)$

$$p(Sports | D_{New}) \propto p(Politics) \times \mu_{Sports, Vote} \times \mu_{Sports, Team} \times \mu_{Sports, Team}$$

$$p(Sports | D_{New}) = \frac{2}{3} \times 0.111 \times 0.444 \times 0.444 \approx 0.014$$

$$p(Politics | D_{New}) \propto p(Politics) \times \mu_{Politics, Vote} \times \mu_{Politics, Team} \times \mu_{Politics, Team}$$

$$p(Sports | D_{New}) = \frac{1}{3} \times 0.667 \times 0.167 \times 0.167 \approx 0.0061$$

Naive Bayes Example – Another Example  $D_{New} = (Vote, Team, Team)$ 

- Normalized:

$$p(Sports \mid D_{New}) = \frac{0.014}{0.014 + 0.0061} \approx 69.7$$

$$p(Politics \mid D_{New}) = \frac{0.0061}{0.0061 + 0.014} \approx 0.33$$

## Naive Bayes Example – Larger Train & Test Set (IMDB)

- Maneuver to R File

## Naive Bayes Example – Larger Train & Test Set (IMDB)

- Maneuver to R File
- Adjust the size of the training set – Does that improve predictive accuracy?

## Looking Forward

- Next Class: Topic Selection & Clustering – Another dense series of material
- Class 6 **and** Class 7 Problem Sets due Sunday
- **Reminder:** Keep working on papers!