Solving Classification Problems

Swetha Kolalapudi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Understand the Naive Bayes algorithm

Solve Sentiment Analysis using the Naive Bayes algorithm

Understand the Support Vector Machines Algorithm

Solve Ad Detection using Support Vector Machines algorithm

Sentiment Analysis

Problem Statement

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

“Train a model” using the training data

Test

“Test the model” using test data

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

"Train a model" using the training data

Test

"Test the model" using test data

Features

Text 1	Positive
Text 2	Positive
Text 3	Negative
Text 4	Positive
Text 5	Negative
Text 6	Positive
.	.
.	.
.	.
.	.
Text 100	Positive

Training Data

A corpus i.e. large body of texts already labelled as positive/negative

Term Frequency Representation

Text 1	**Positive**
Text 2	**Positive**
Text 3	**Negative**
Text 4	**Positive**
Text 5	**Negative**
Text 6	**Positive**
. . . .	. . . .
Text 100	**Positive**

Represent a text using frequency of words in the text

Term Frequency Representation

(1,0,1,1..0)	Positive
(1,1,0,1..0)	Positive
(0,0,1,0..0)	Negative
(0,0,0,1..0)	Positive
(1,1,1,1..0)	Negative
(0,0,1,1..1)	Positive
. . . .	. . . .
(1,0,1,1..0)	Positive

Each element in a tuple represents the frequency of some **word**

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

"Train a model" using the training data

Test

"Test the model" using test data

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

“Train a model” using the training data

Test

“Test the model” using test data

Naive Bayes - Training Phase

(1,0,1,1..0)	Positive
(1,1,0,1..0)	Positive
(0,0,1,0..0)	Negative
(0,0,0,1..0)	Positive
(1,1,1,1..0)	Negative
(0,0,1,1..1)	Positive
. . . .	. . . .
(1,0,1,1..0)	Positive

If we pick a comment at random, the probability of

Positive P_0

Negative $1-P_0$

Naive Bayes - Training Phase

(1,0,1,1..0)	Positive
(1,1,0,1..0)	Positive
(0,0,1,0..0)	Negative
(0,0,0,1..0)	Positive
(1,1,1,1..0)	Negative
(0,0,1,1..1)	Positive
.	.
.	.
.	.
.	.
(1,0,1,1..0)	Positive

Comments/texts are made up of **words**

Naive Bayes - Training Phase

(1,0,1,1..0)	Positive
(1,1,0,1..0)	Positive
(0,0,1,0..0)	Negative
(0,0,0,1..0)	Positive
(1,1,1,1..0)	Negative
(0,0,1,1..1)	Positive
.	.
.	.
.	.
.	.
(1,0,1,1..0)	Positive

We can compute a **Positivity Score and Negativity Score** for every word

Naive Bayes - Training Phase

Ex: The word “Happy”

$$Pos_{Happy} = \frac{\text{Sum frequency of “Happy” in positive comments}}{\text{Sum frequency of “Happy” in the entire corpus}}$$

$$Neg_{Happy} = 1 - Pos_{Happy}$$

Naive Bayes - Training Phase

Positive P_0

Negative $1-P_0$

	Positive
Happy	90%
Love	95%
Food	50%
Hate	10%
Bad	30%
.	.
.	.
.	.

Use all this to classify any new comments

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

“Train a model” using the training data

Test

“Test the model” using test data

Typical Classification Setup

Problem Statement	Features	Training	Test
Define the problem statement	Represent the training data and test data using numerical attributes	“Train a model” using the training data	**“Test the model” using test data**

Naive Bayes - Test Phase

Given any new comment

“Love the food!”

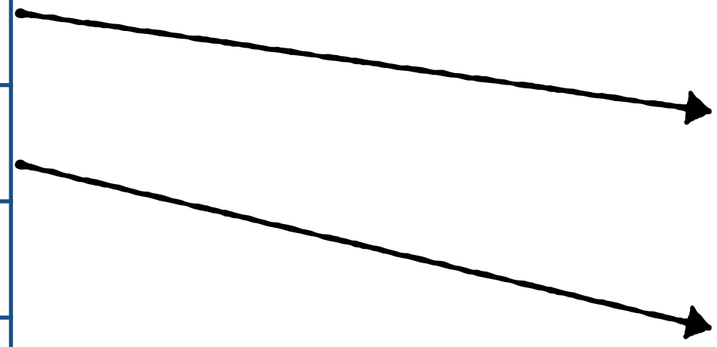
Positivity score

Negativity score

Compare the 2 scores

Naive Bayes - Test Phase

	Positive context
Happy	90%
Love	95%
Food	50%
Hate	10%
Bad	30%
.	.
.	.
.	.
.	.



Positivity score

$$Pos_{Love} * Pos_{food} * P_0$$

Naive Bayes - Test Phase

	Positive context
Happy	90%
Love	95%
Food	50%
Hate	10%
Bad	30%
.	.
.	.
.	.
.	.

Negativity score

$(1-Pos_{Love})$ *

$(1-Pos_{food})$ *

$(1-P_0)$

Naive Bayes - Test Phase

$$Pos_{Comment} = Pos_{Word1} * Pos_{Word2} * \ldots.$$

Similarly, we compute a Neg score and compare the 2

Why “Naive”?

$$Pos_{Comment} = Pos_{Word1} * Pos_{Word2} * \ldots.$$

Each word contributes
independently

Why “Naive”?

$$Pos_{Comment} = Pos_{Word1} * Pos_{Word2} * \ldots.$$

No term accounting for a
two or more words
appearing together

Why “Naive”?

The independence assumption
is the reason why

Naive Bayes algorithm is called
“Naive”

Independence Assumption

Naive Bayes is not any less powerful

Strengths of Naive Bayes

Excellent results for many classification problems

Robust models

Use Naive Bayes

When you have very little training data

When you have less information about the problem itself

Ad Detection

Problem Statement

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

"Train a model" using the training data

Test

"Test the model" using test data

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

"Train a model" using the training data

Test

"Test the model" using test data

Features

(200,1,0,1,...0)

Tuple of numbers

Image Features

Height, Width

Page URL

Image URL

Page text

Image Caption text

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

"Train a model" using the training data

Test

"Test the model" using test data

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

“Train a model” using the training data

Test

“Test the model” using test data

SVM - Training Phase

Training Data

A large dataset of images
which are already labelled as
Ad/Non-Ad

SVM - Training Phase

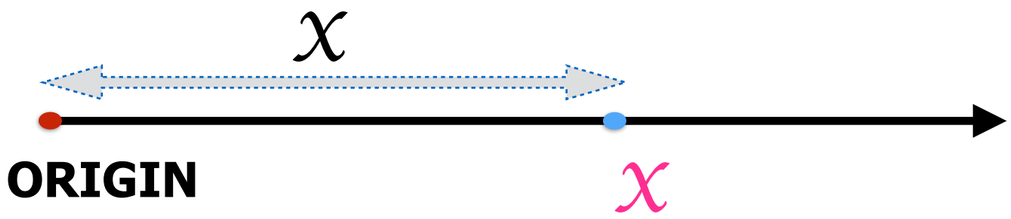
Represent all the images as points in an N-Dimensional Hypercube

N-Dimensional Hypercube

N-Dimensional Hypercube

A line is a 1-dimensional shape

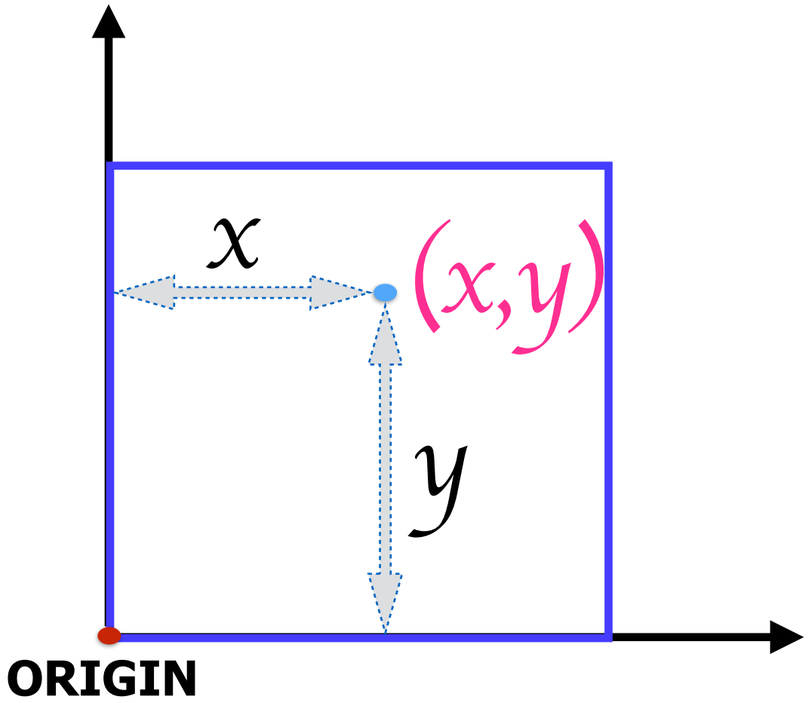
Any point on a line can be represented using 1 number



N-Dimensional Hypercube

A square is a 2-Dimensional shape

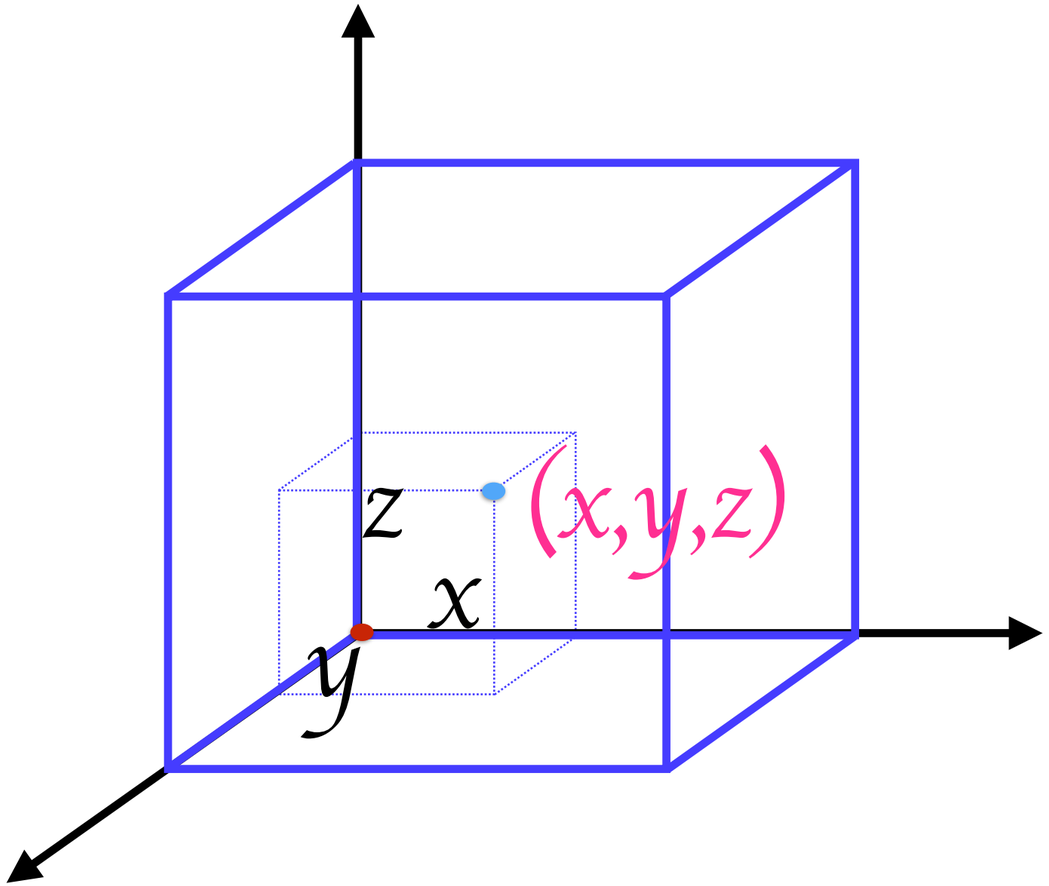
Any point in a square can be represented using 2 numbers



N-Dimensional Hypercube

A cube is a 3-dimensional shape

Any point in a cube can be represented with 3 numbers



N-Dimensional Hypercube

A set of N numbers represents a point in an N-Dimensional Hypercube

N-Dimensional Hypercube

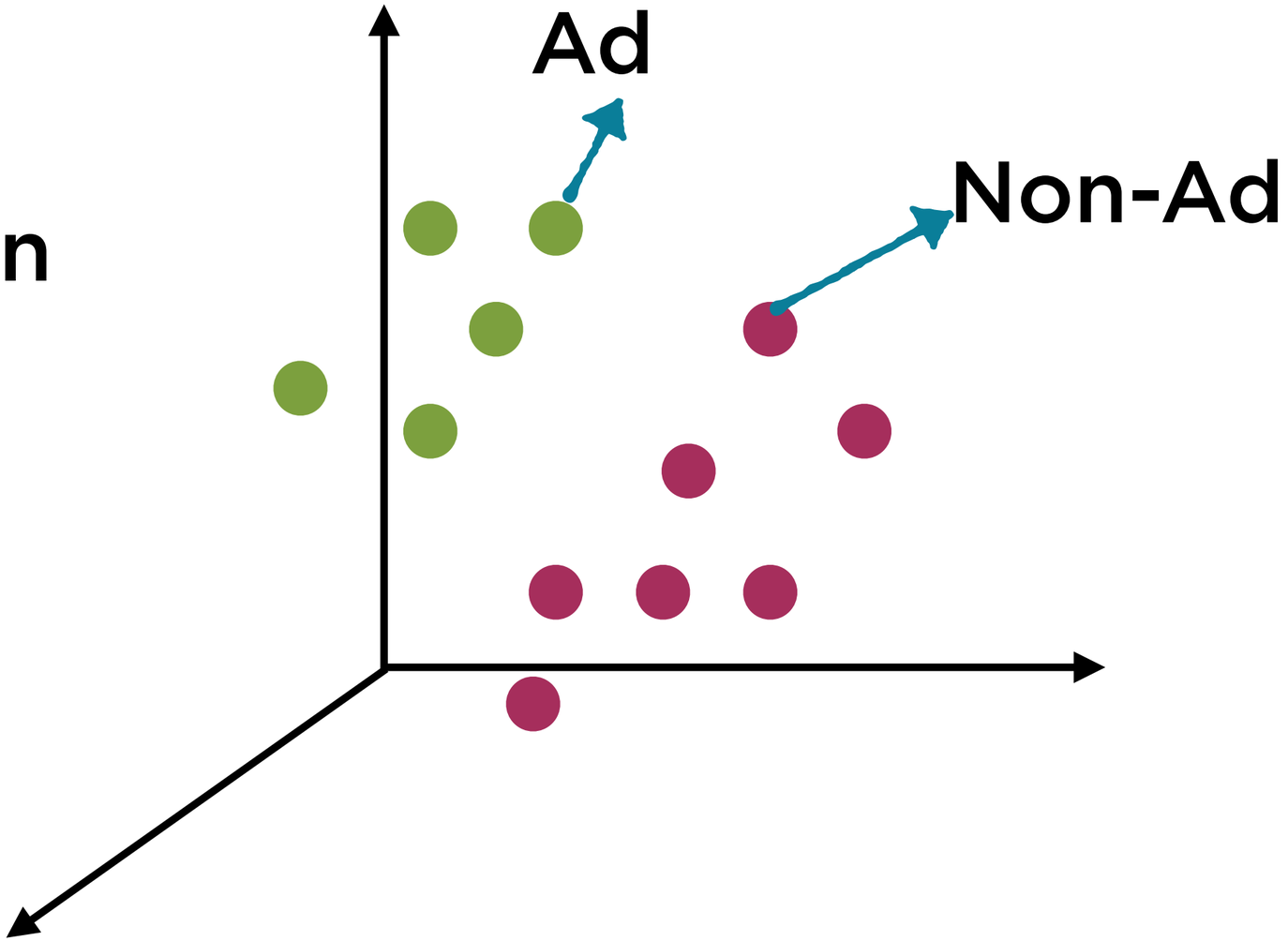
(200,1,0,1,...0)

Tuple of N numbers

A point in an N-Dimensional Hypercube

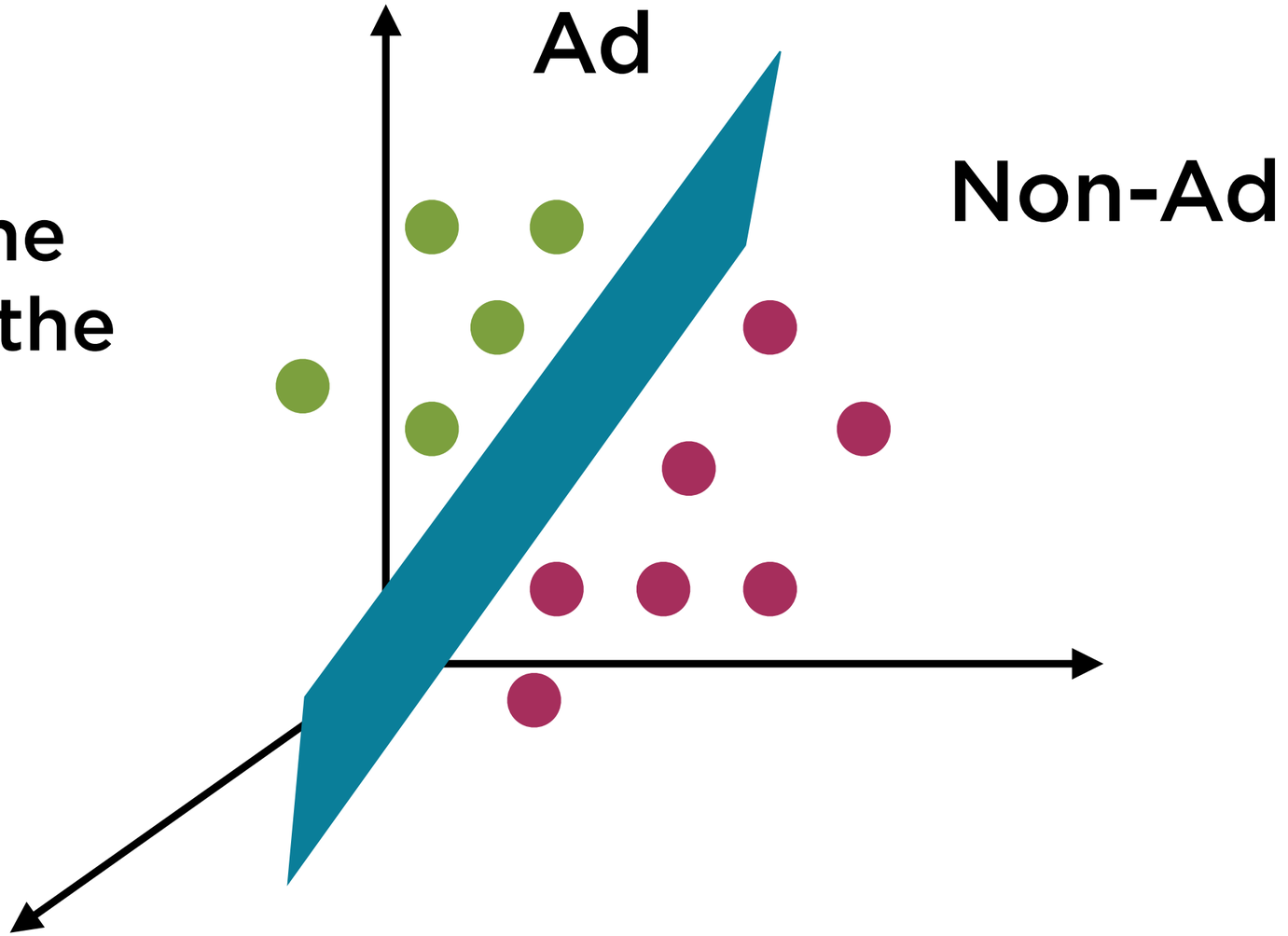
SVM - Training Phase

Represent all the images as points in an N-Dimensional Hypercube



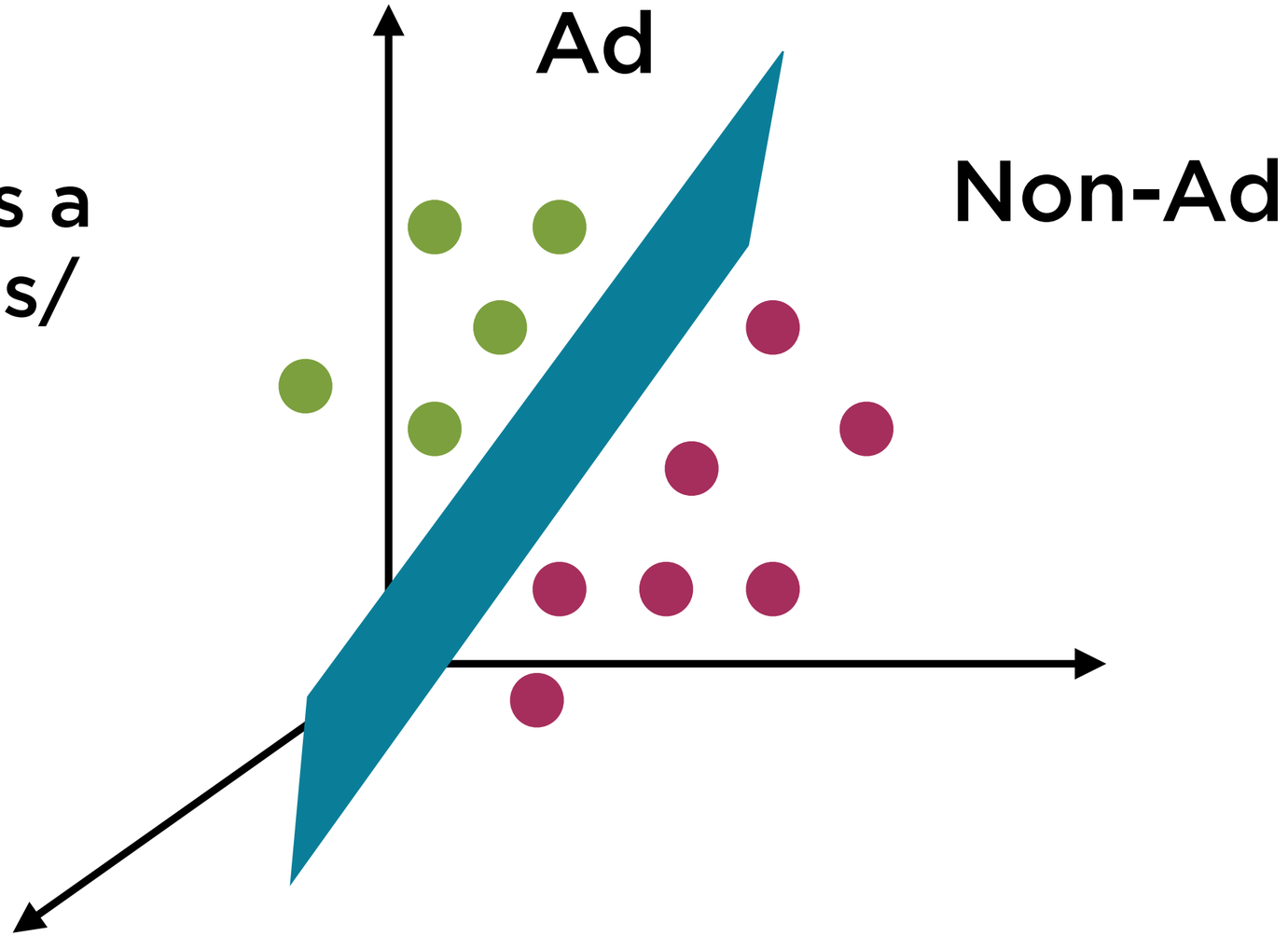
Training Phase

SVM finds a hyperplane that neatly separates the 2 sets of points



Training Phase

The hyperplane acts as a **boundary** between Ads/ Non-Ads



Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

Training

“Train a model” using the training data

Test

“Test the model” using test data

Typical Classification Setup

Problem Statement

Define the problem statement

Features

Represent the training data and test data using numerical attributes

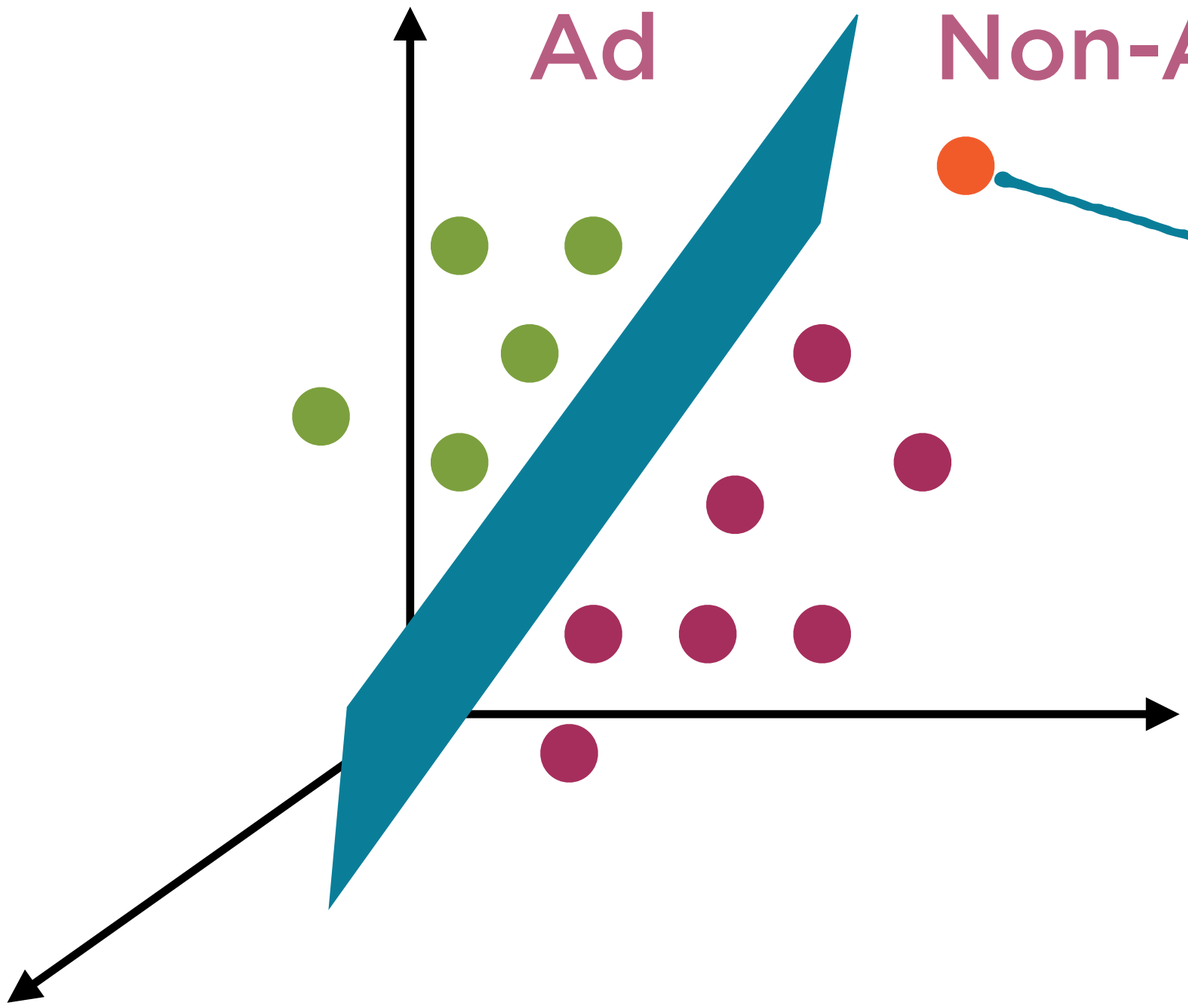
Training

"Train a model" using the training data

Test

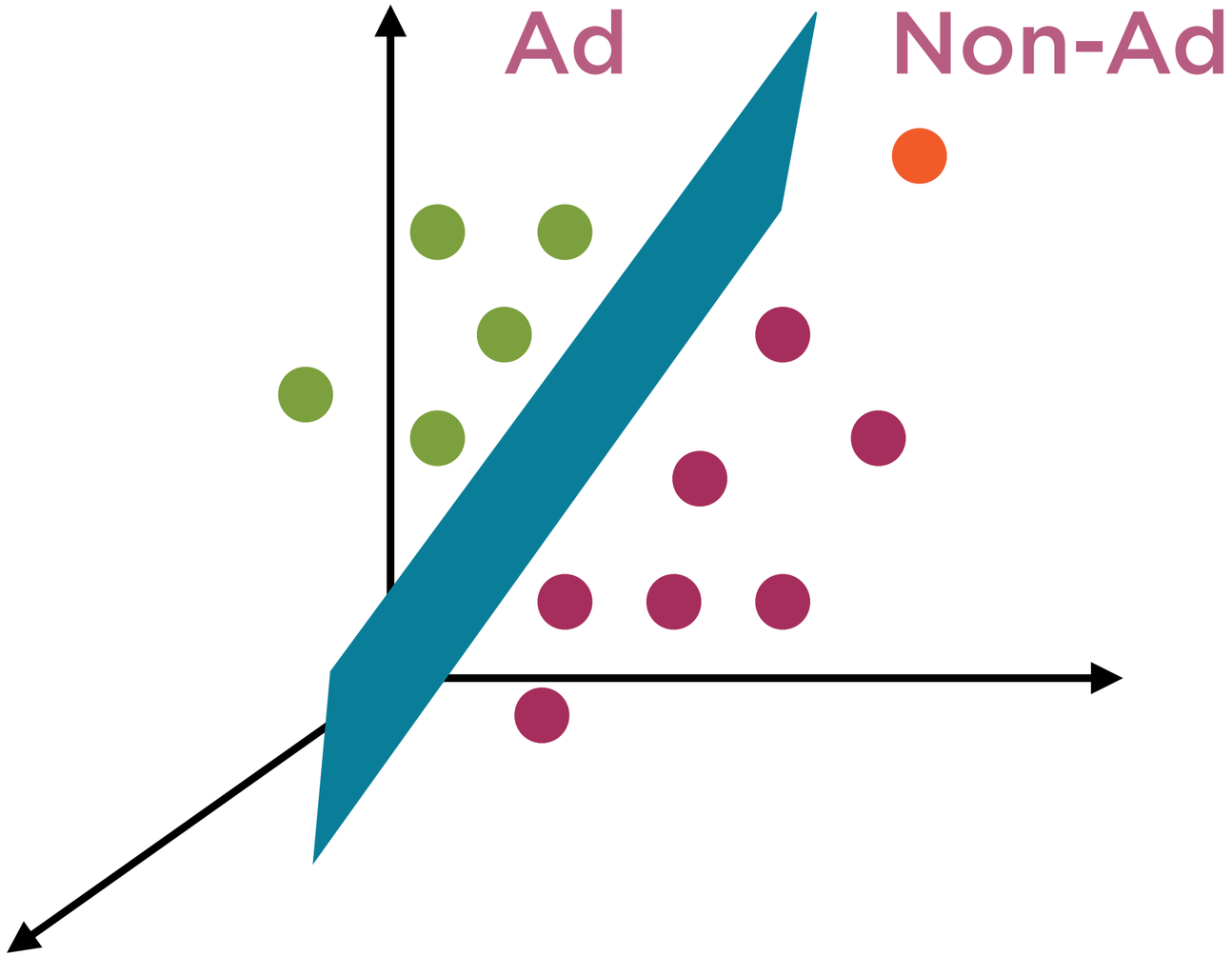
"Test the model" using test data

Test Phase



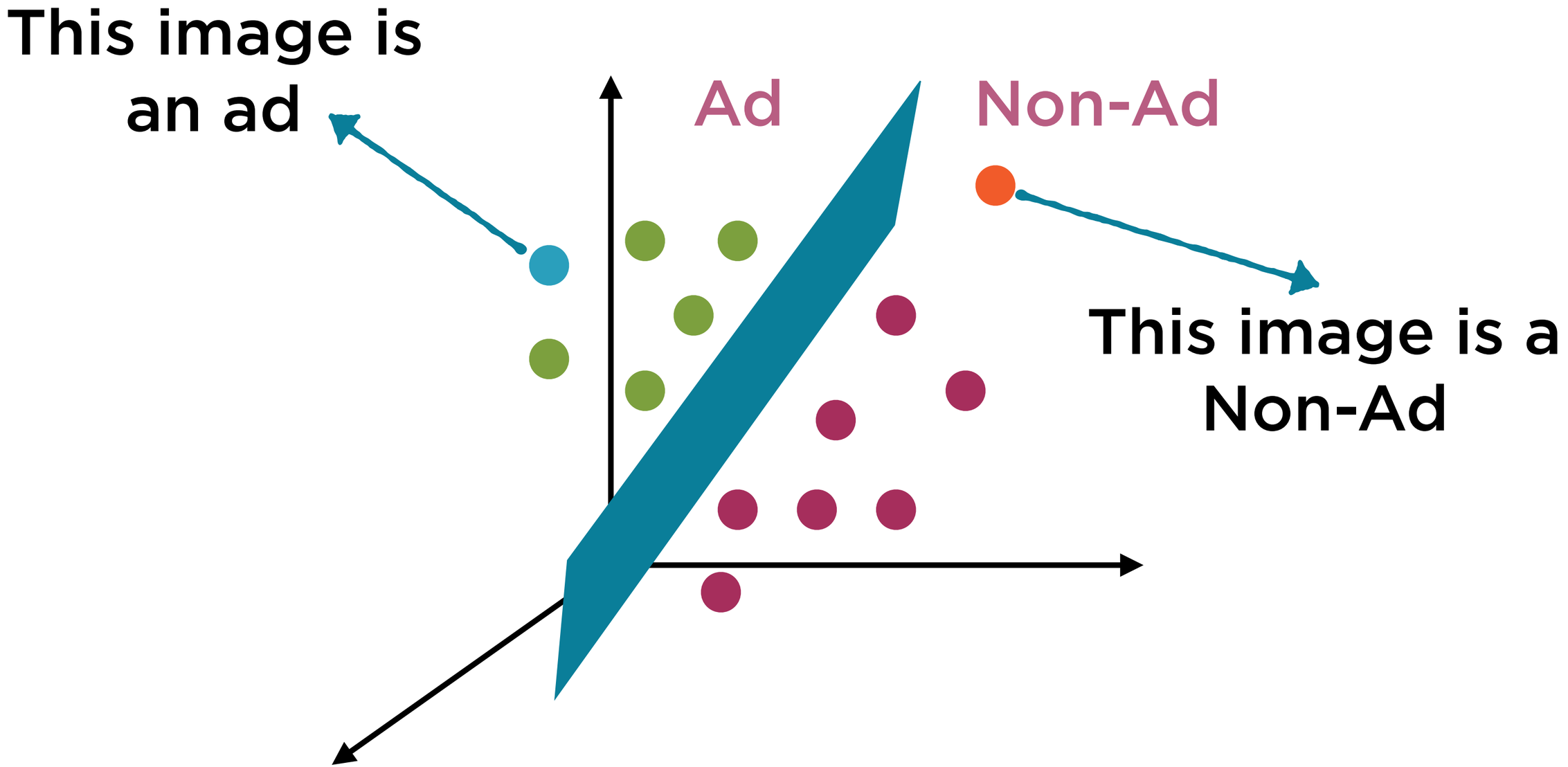
In the test phase, we get a new image

Test Phase

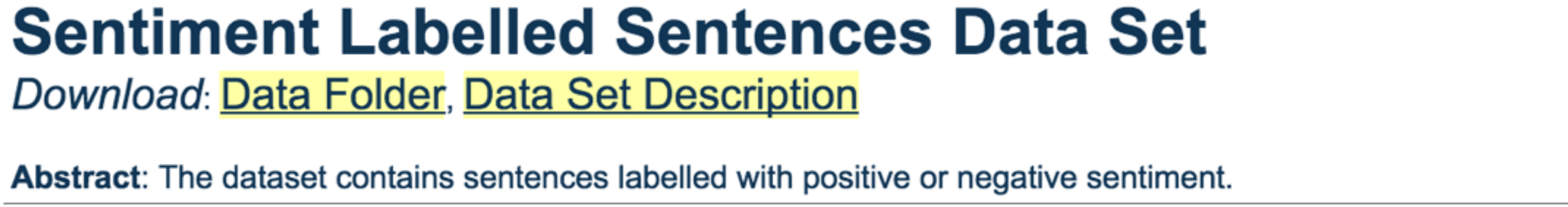


Check which side of the boundary the new image falls on

Test Phase



The dataset for this demo is from the UCI Machine Learning Repository



https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences

The data set has sentences from 3 sources

IMDB reviews

Yelp reviews

Amazon reviews

Each line in the data set

review	label
Wasted two hours.	0

Demo

Implement Naive Bayes algorithm on the Sentiment Labelled Sentences data set

Use the Scikit-Learn Module in Python

The dataset for this demo is from the UCI Machine Learning Repository

Internet Advertisements Data Set

Download: Data Folder, Data Set Description

Abstract: This dataset represents a set of possible advertisements on Internet pages.

https://archive.ics.uci.edu/ml/datasets/Internet+Advertisements

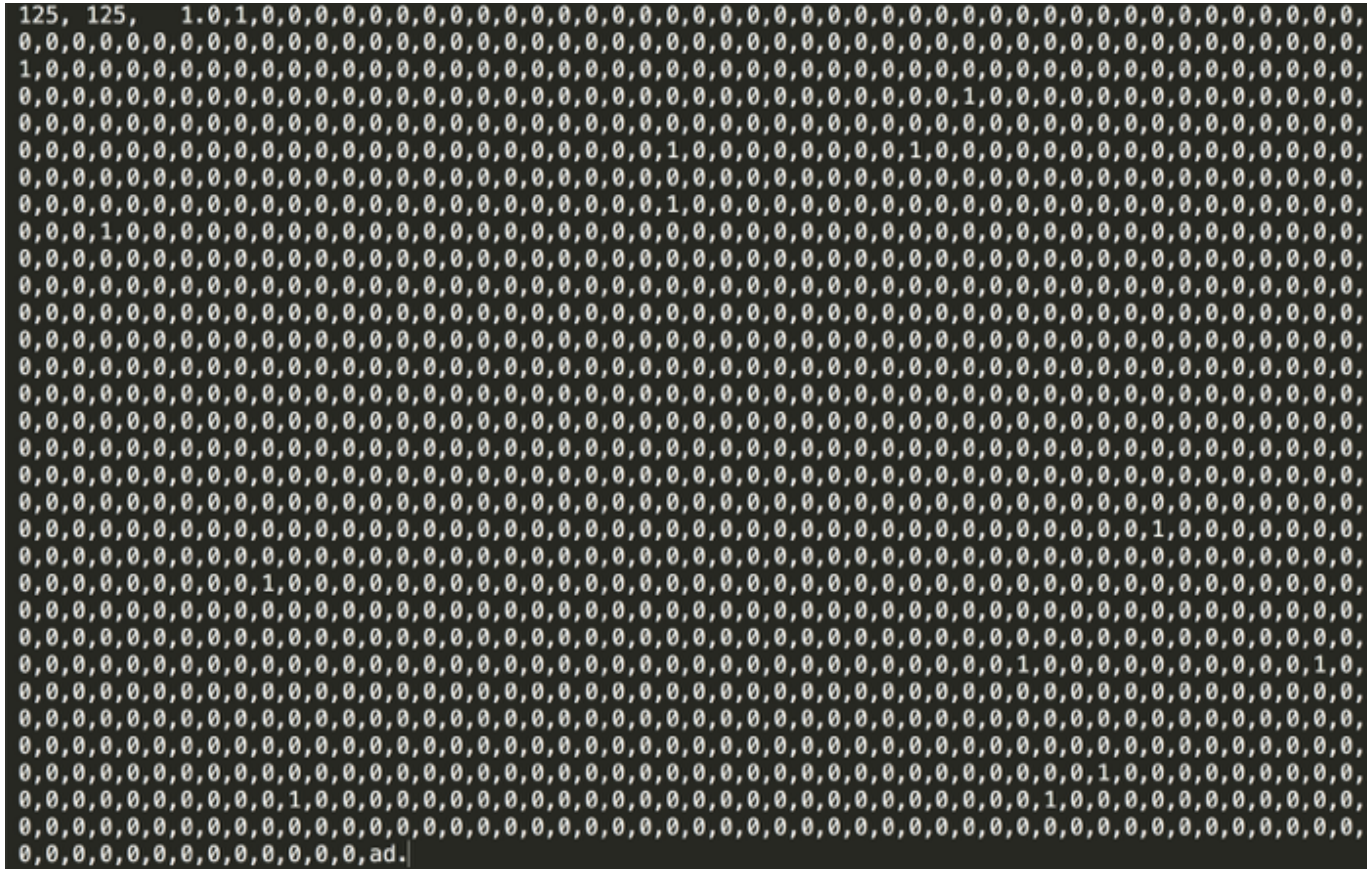
Each image is

- labelled as Ad/Non-Ad
- represented using ~1500 attributes

```
height: continuous. | possibly missing
width: continuous.  | possibly missing
aratio: continuous. | possibly missing
local: 0,1.
| 457 features from url terms, each of the form "url*term1+term2...";
| for example:
url*images+buttons: 0,1.
  ...
| 495 features from origurl terms, in same form; for example:
origurl*labyrinth: 0,1.
  ...
| 472 features from ancurl terms, in same form; for example:
ancurl*search+direct: 0,1.
  ...
| 111 features from alt terms, in same form; for example:
alt*your: 0,1.
  ...
| 19 features from caption terms
caption*and: 0,1.
  ...
```

1557 numerical attributes

ad. / nonad.



Demo

Implement SVM on the Internet Advertisements data set

Summary

Understand the Naive Bayes algorithm

Solve Sentiment Analysis using the Naive Bayes algorithm

Understand the Support Vector Machines Algorithm

Solve Ad Detection using Support Vector Machines algorithm