Getting Started

CS771: Introduction to Machine Learning

Purushottam Kar

Announcements

Course has reached capacity – no more adds unless drops take place

Please sign-up on Piazza – we will discontinue use of emails soon

All discussions/queries only on Piazza – none over email

Class polls on whether to hold extra discussions also only over Piazza

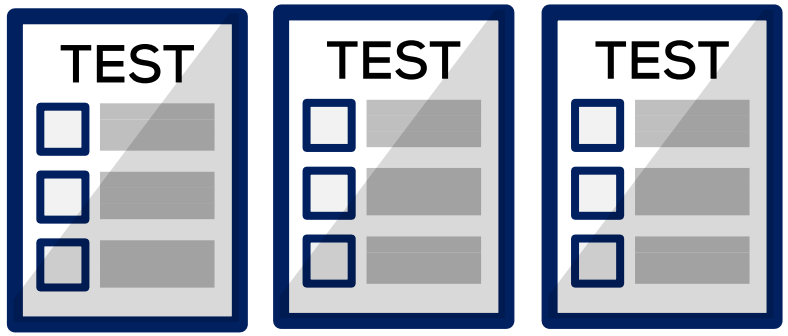
Do not worry about Gradescope – I will signup all registered students

Reference list updated – more organized, math for ML resources

An overview of ML

- Study similarity of ML with a student preparing for an exam
- Look at a toy ML problem
- Learn what is training data, test data?
- Learn what is a model?

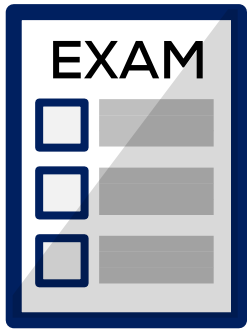
A typical study cycle (e.g. in a course)



Practice Tests/Study Material

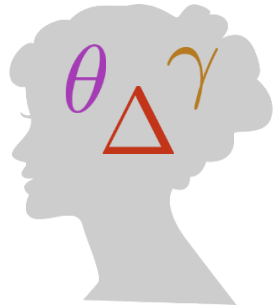
Preparation

Actual Exam

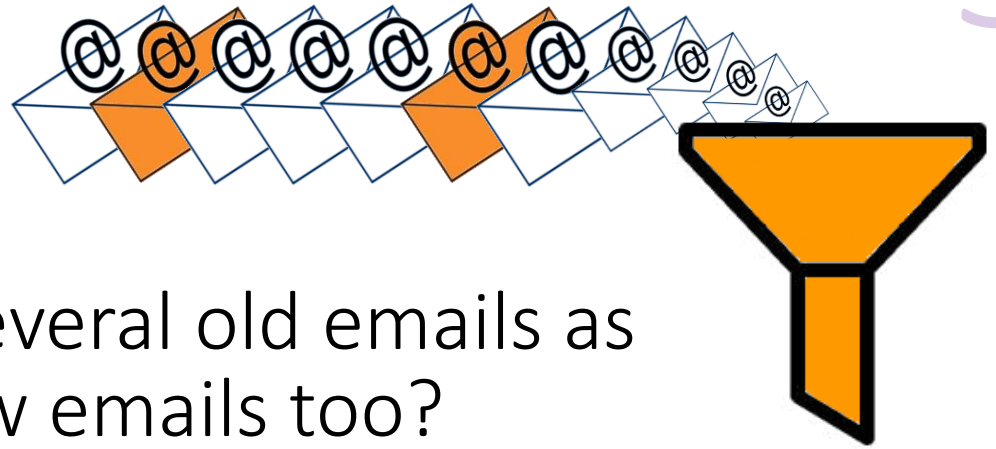


Pass / Fail

Subject Knowledge



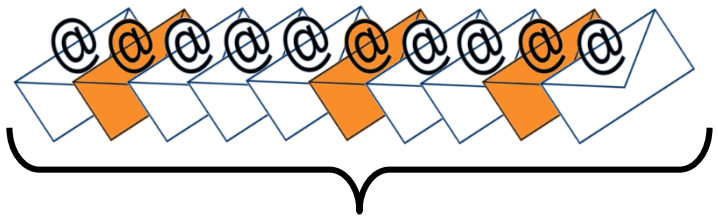
Spam Filtering



- Suppose Mary has already tagged several old emails as spam/non-spam, can we tag her new emails too?
- **Trick**: use the old tagged emails to try and understand what sort of emails does Mary think of as spam and which as non-spam!
- E.g. may find that emails about shopping always tagged as spam
- E.g. may find that emails from Jill are never tagged as spam
- These insights/patterns are what are stored in the spam filter
- Our spam filter helps us make predictions on new emails

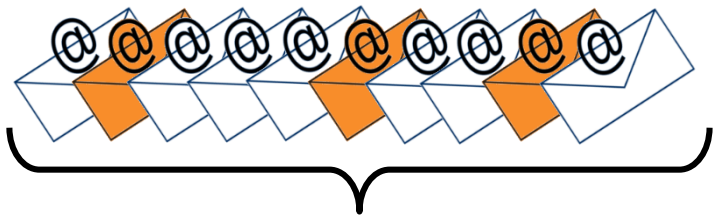
A typical ML workflow

A typical ML workflow

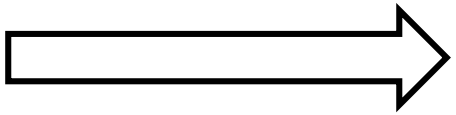


Old Emails

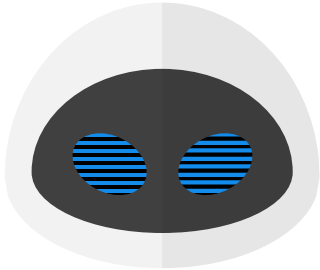
A typical ML workflow



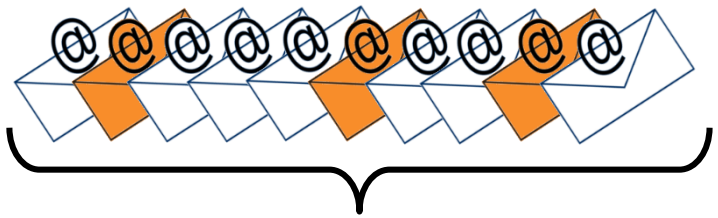
Old Emails



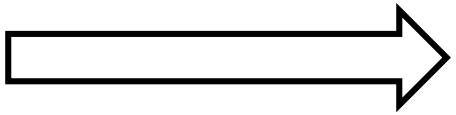
ML Algorithm



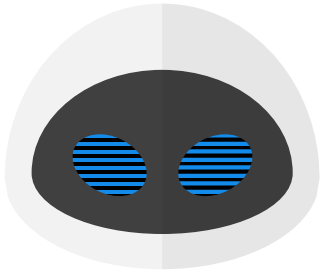
A typical ML workflow

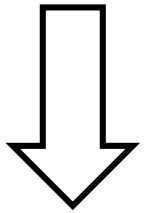


Old Emails

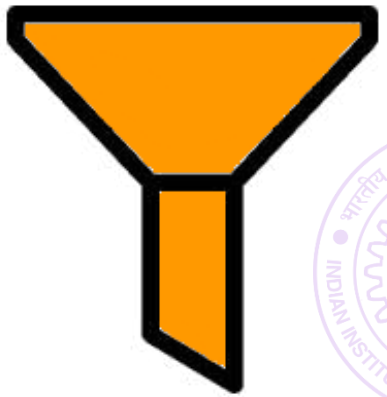


ML Algorithm

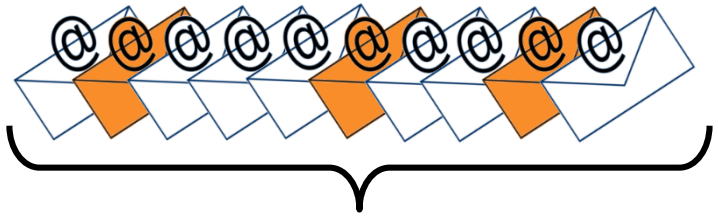




Spam Filter

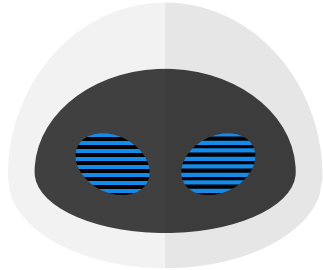


A typical ML workflow

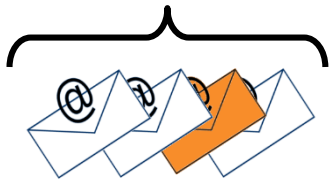


Old Emails

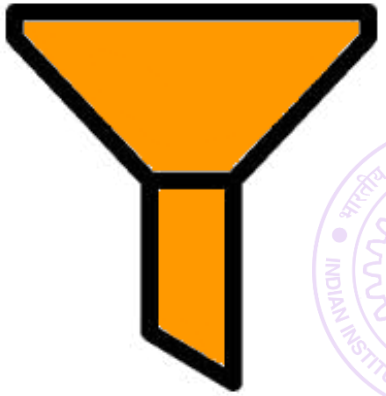
ML Algorithm



New Emails

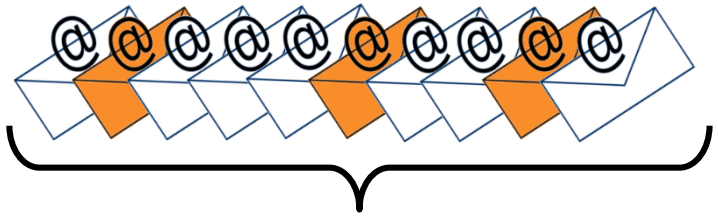


Spam Filter



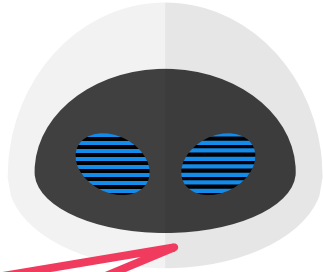
Spam/Non-spam

A typical ML workflow



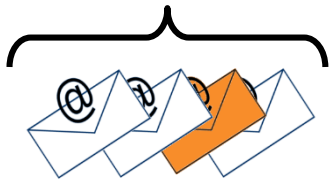
Old Emails

ML Algorithm

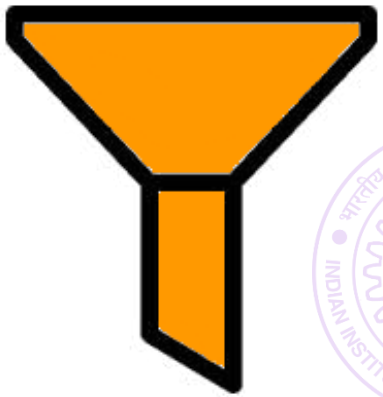


The spam filter stores information about your personal preferences about what looks spam to you

New Emails

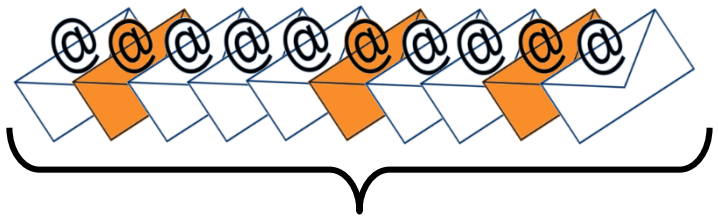


Spam Filter



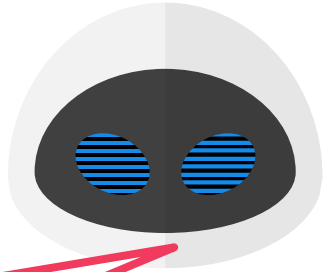
Spam/Non-spam

A typical ML workflow



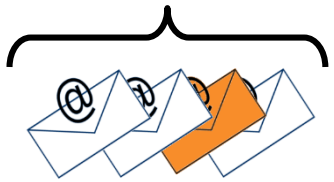
Old Emails

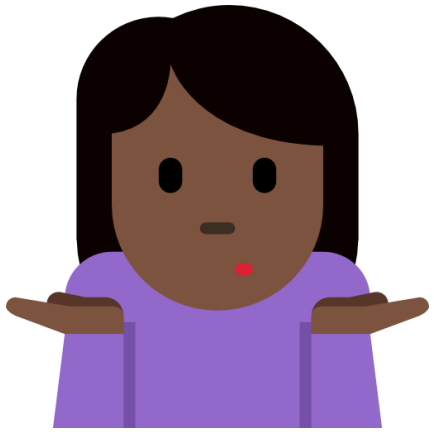
ML Algorithm



The spam filter stores information about your personal preferences about what looks spam to you

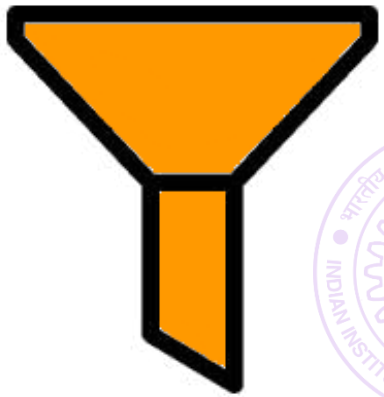
New Emails



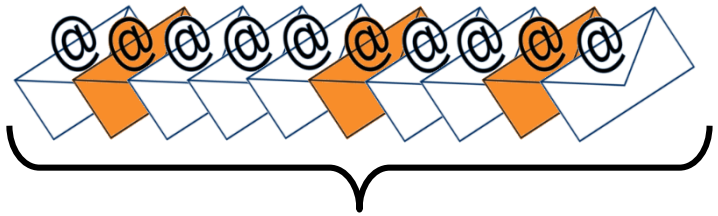


Spam/Non-spam

Spam Filter

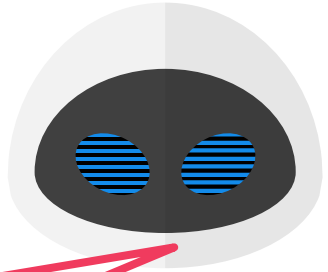


A typical ML workflow



Old Emails

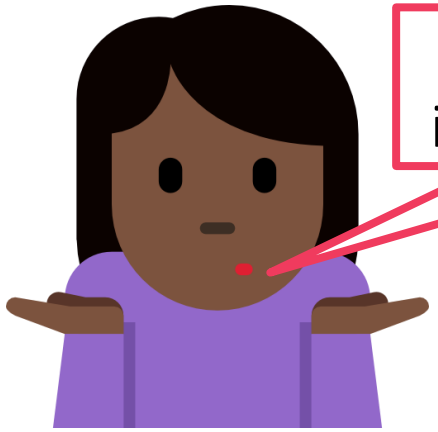
ML Algorithm



The spam filter stores information about your personal preferences about what looks spam to you

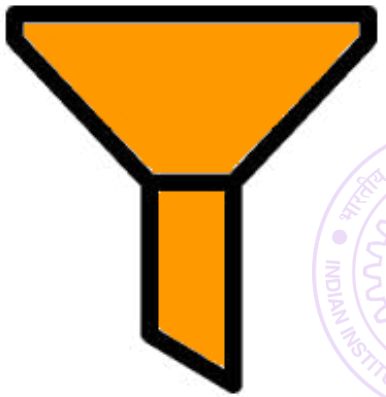
New Emails

How is this information stored?

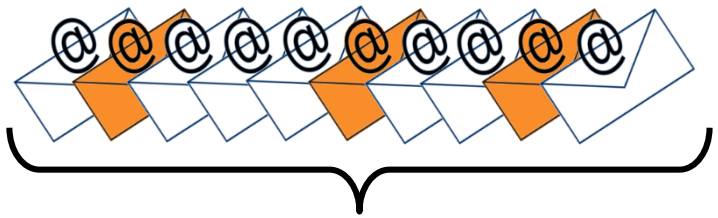


Spam/Non-spam

Spam Filter

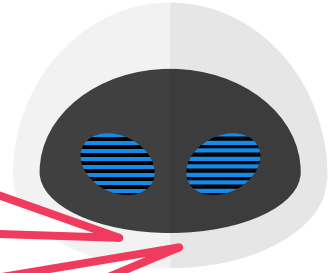


A typical ML workflow



Old Emails

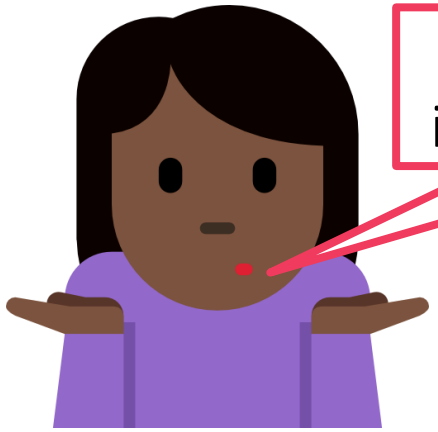
Ah! That is the fun (and artistic) bit about ML. We will learn tons of ways on how ML algos store patterns



The spam filter stores information about your personal preferences about what looks spam to you

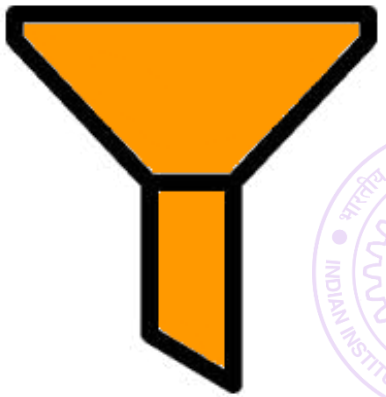
New Emails

How is this information stored?

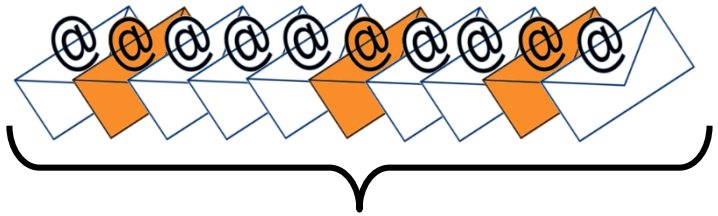


Spam/Non-spam

Spam Filter

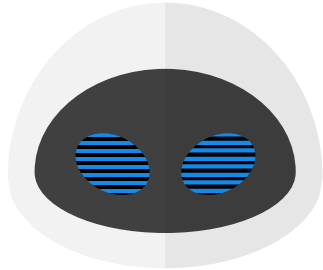


A typical ML workflow

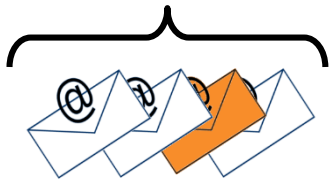


Old Emails

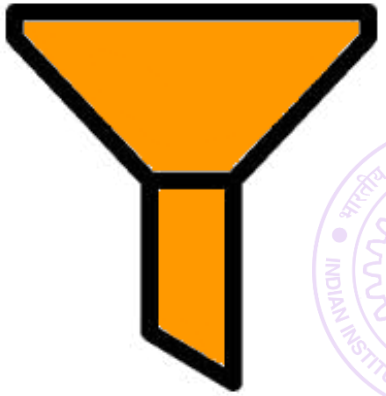
ML Algorithm



New Emails

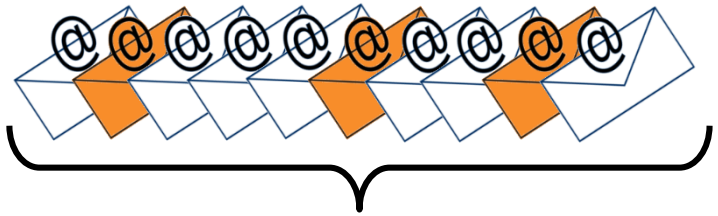


Spam Filter



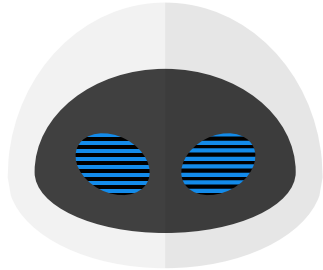
Spam/Non-spam

A typical ML workflow



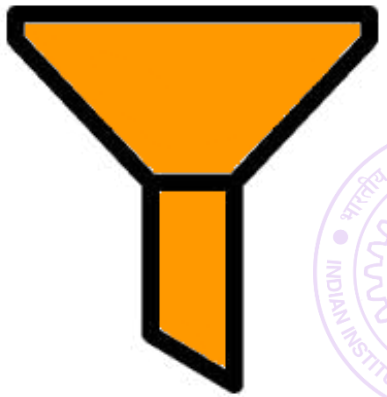
Training Data

ML Algorithm



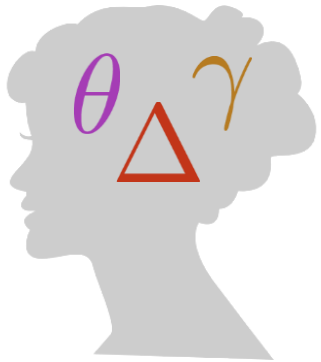
Test Data

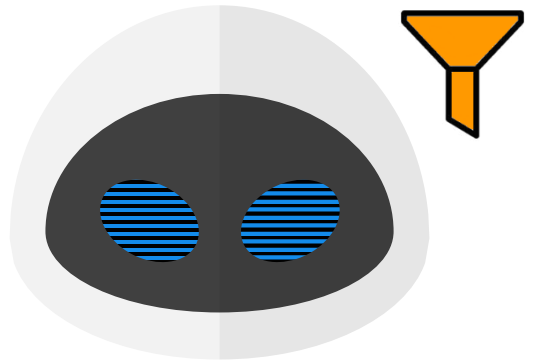
Model



Output

ML as an "examination"





Our brain stores subject matter	The model stores data patterns
Use subject matter to solve exam	Use model to predict on test data
Critical to do well on exam-day	Critical to do well on test data
Mock test results indicative	Training accuracies indicative
No out-of syllabus questions	Training/test data are similar
Should not leak exam paper before exam	Should not look at test data while training

ML can do lots of cool things with test data

Test Data

ML can do lots of cool things with test data

Test Data

Regression

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

ML can do lots of cool things with test data

Test Data

Regression

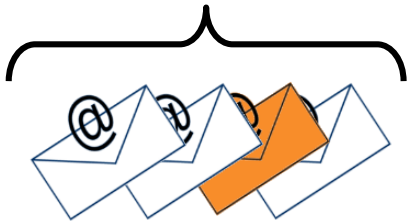
Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

Binary Classification

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Status: Yes

ML can do lots of cool things with test data

Test Data



Regression

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

Multi-classification

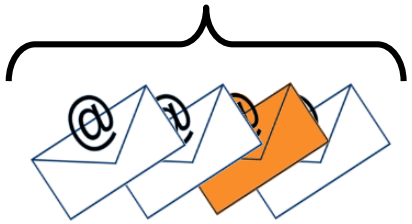
Primary Social Promotions
Updates Forums

Binary Classification

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Status: Yes

ML can do lots of cool things with test data

Test Data



Regression

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

Multi-classification

Primary Social Promotions
Updates Forums

Tagging

URGENT,
OFFICIAL,
TAX

Binary Classification

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Status: Yes

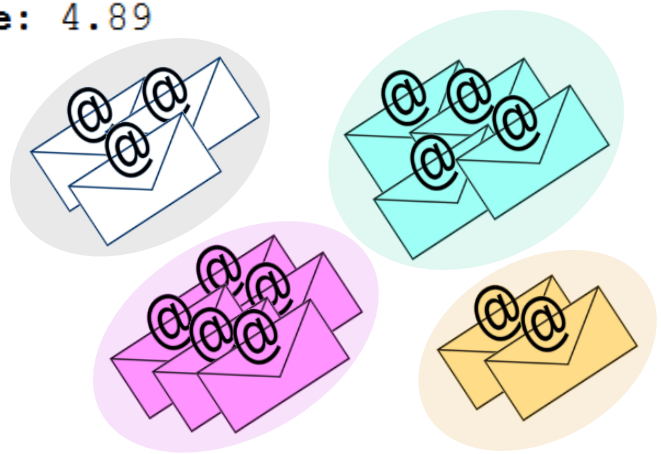
ML can do lots of cool things with test data

Test Data

Regression

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

Clustering



Multi-classification

Primary Social Promotions
Updates Forums

Tagging

URGENT, OFFICIAL, TAX

Binary Classification

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Status: Yes

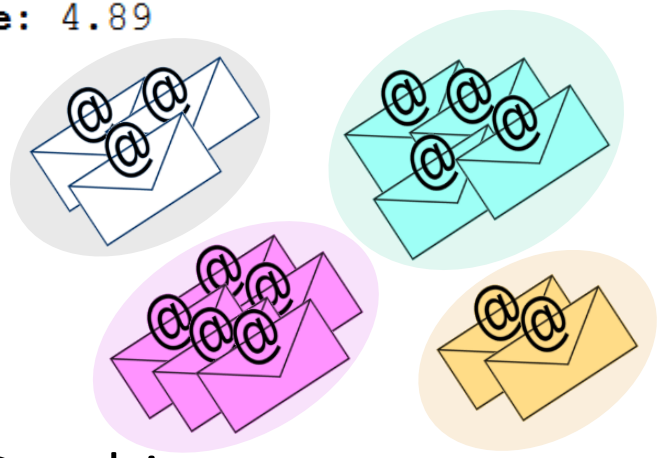
ML can do lots of cool things with test data

Test Data

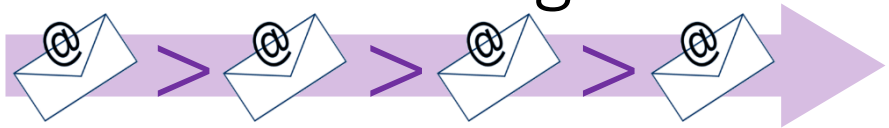
Regression

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

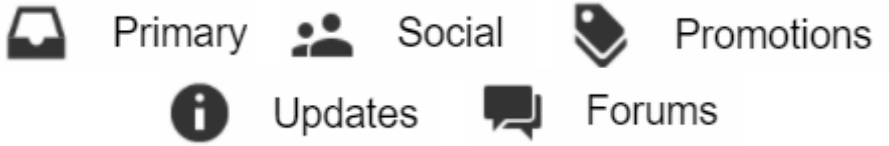
Clustering



Ranking



Multi-classification



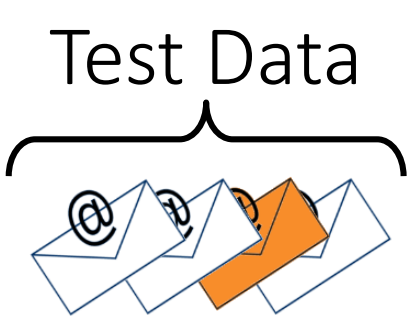
Tagging

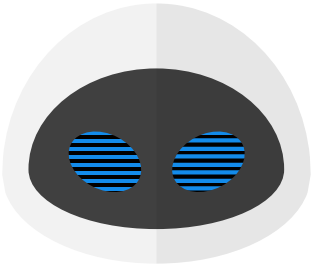
URGENT,
OFFICIAL,
TAX

Binary Classification

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Status: Yes

ML can do lots of cool things with test data

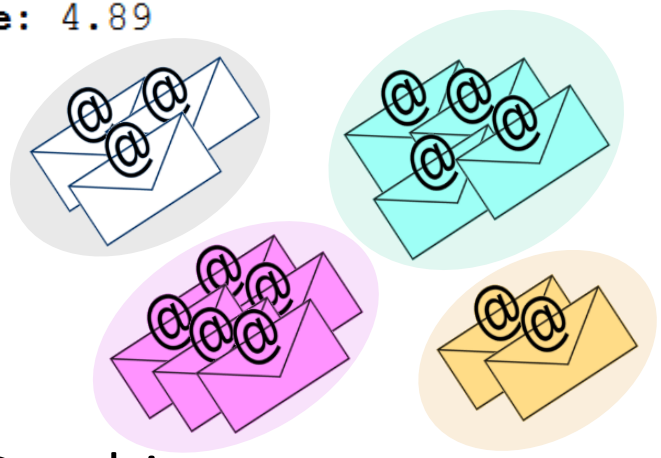




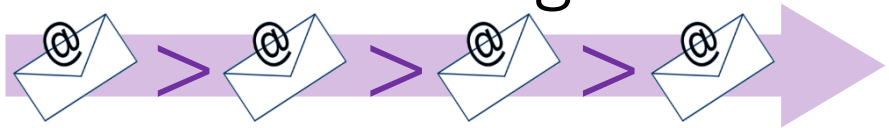
Regression

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

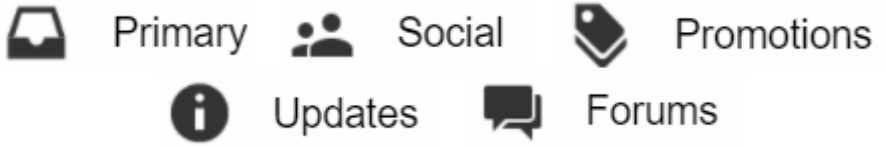
Clustering



Ranking



Multi-classification



Tagging

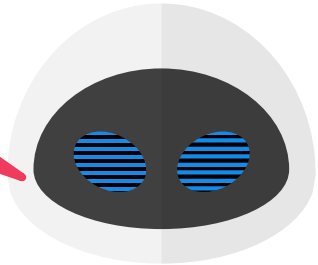
URGENT,
OFFICIAL,
TAX

Binary Classification

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Status: Yes

ML can do lots of cool things with test data

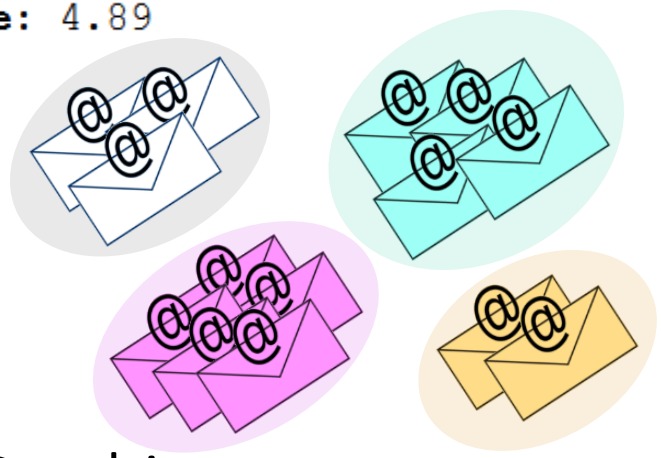
In this course, we will learn how to do most of these operations with test data



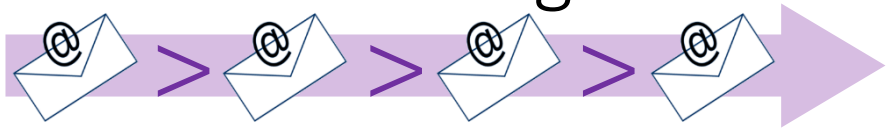
Regression

Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Score: 4.89

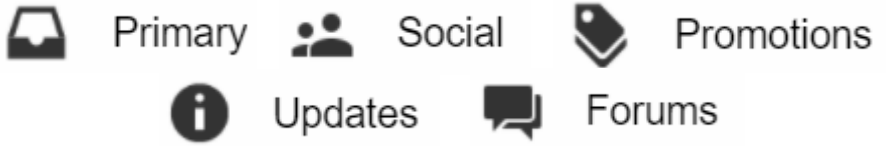
Clustering



Ranking



Multi-classification



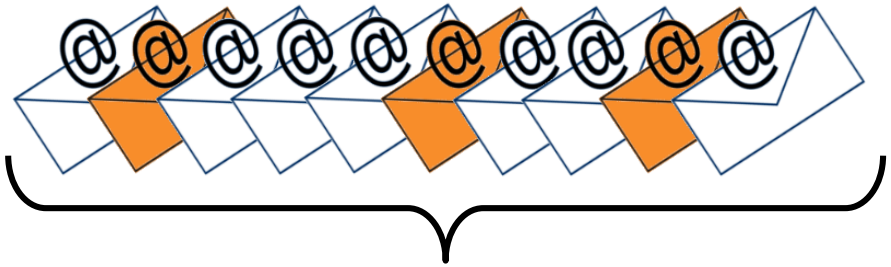
Tagging

URGENT, OFFICIAL, TAX

Binary Classification

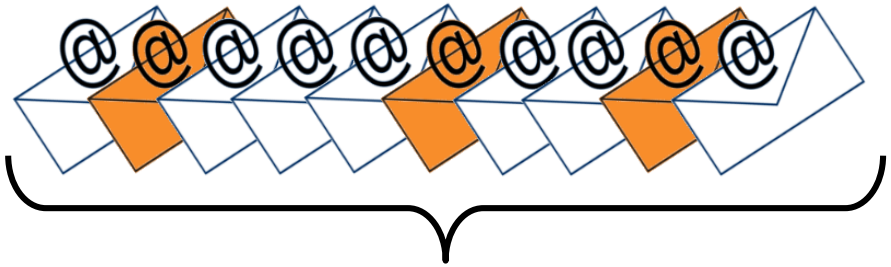
Subject: [**SPAM**] Free movie tickets every month
X-Barracuda-Spam-Status: Yes

ML can take in lots of kinds of training data



Training Data

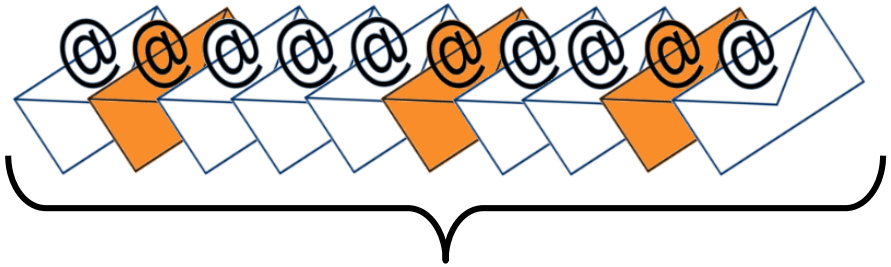
ML can take in lots of kinds of training data



Training Data

Batch Learning

ML can take in lots of kinds of training data

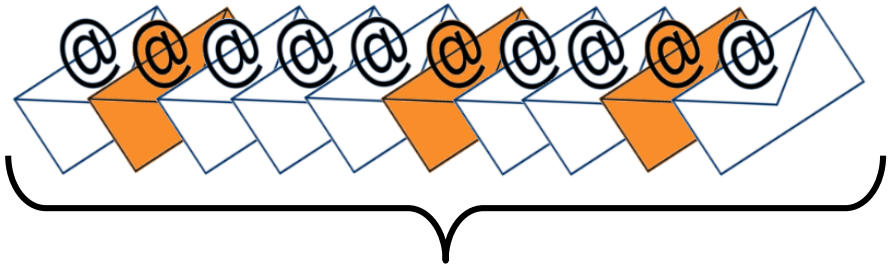


Training Data

Batch Learning

Online Learning

ML can take in lots of kinds of training data



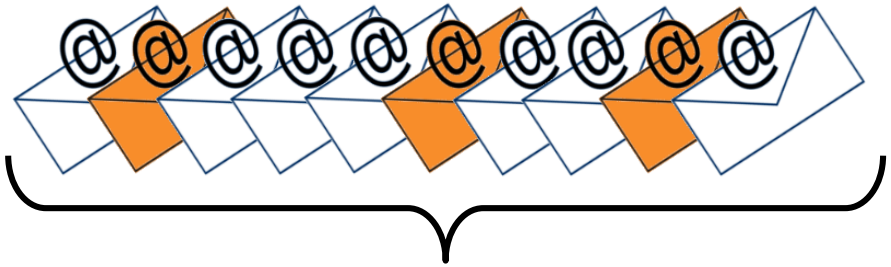
Training Data

Batch Learning

Supervised Learning

Online Learning

ML can take in lots of kinds of training data



Training Data

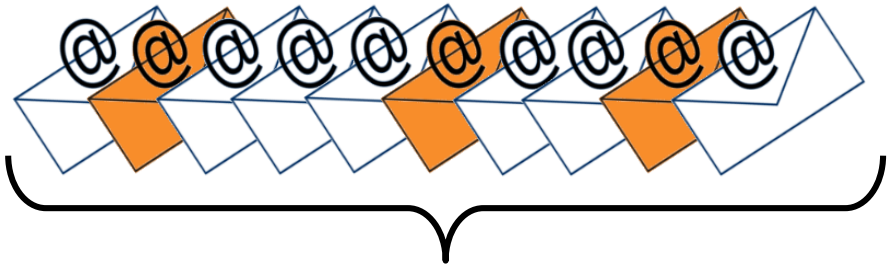
Batch Learning

Supervised Learning

Online Learning

Unsupervised Learning

ML can take in lots of kinds of training data



Training Data

Batch Learning

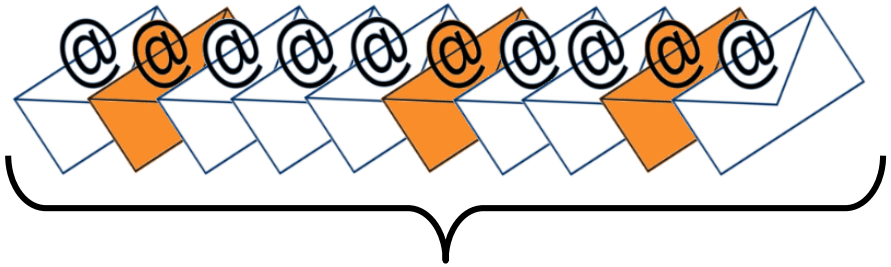
Semi-supervised Learning

Supervised Learning

Online Learning

Unsupervised Learning

ML can take in lots of kinds of training data



Training Data

Batch Learning

Active Learning

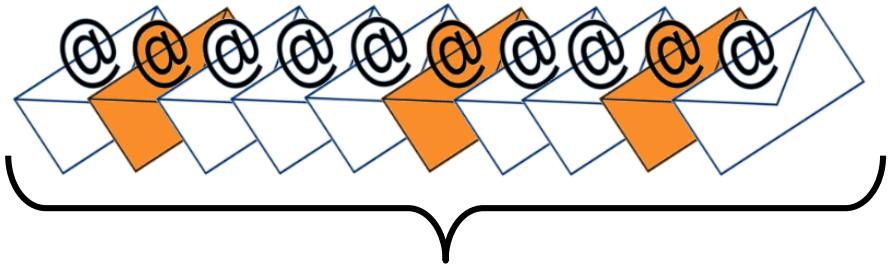
Semi-supervised Learning

Supervised Learning

Online Learning

Unsupervised Learning

ML can take in lots of kinds of training data



Training Data

Batch Learning

Active Learning

Robust Learning

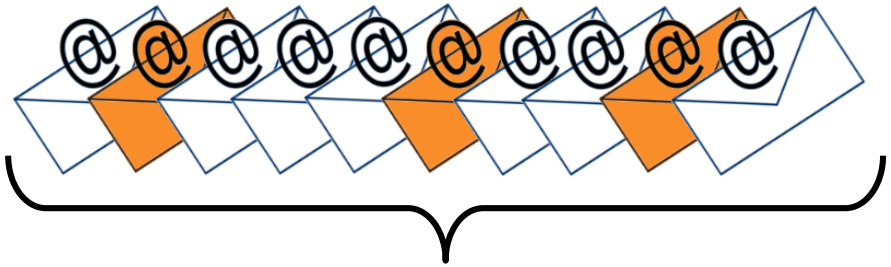
Semi-supervised Learning

Supervised Learning

Online Learning

Unsupervised Learning

ML can take in lots of kinds of training data



Training Data

Batch Learning

Active Learning

Robust Learning

Semi-supervised Learning

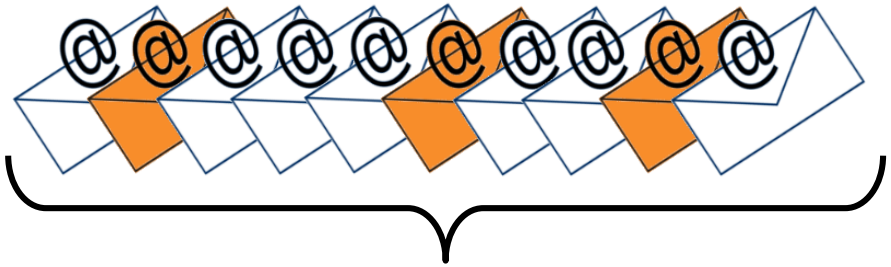
Supervised Learning

Online Learning

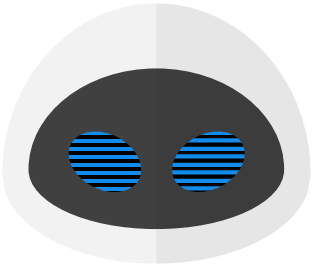
Unsupervised Learning

Reinforcement Learning

ML can take in lots of kinds of training data



Training Data



Batch Learning

Active Learning

Robust Learning

Semi-supervised Learning

Supervised Learning

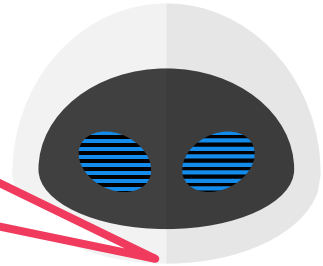
Online Learning

Unsupervised Learning

Reinforcement Learning

ML can take in lots of kinds of training data

We wont be able to cover all these training settings in this course – there are entire courses devoted to specific training settings e.g. CS773 (Online Learning)



Training Data

Batch Learning

Active Learning

Robust Learning

Semi-supervised Learning

Supervised Learning

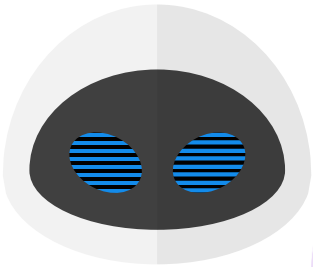
Online Learning

Unsupervised Learning

Reinforcement Learning

ML can store info in lots of innovative ways

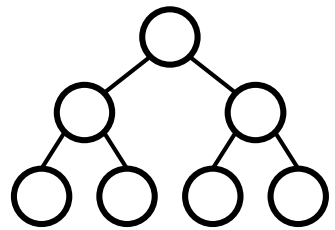
ML Models and Algorithms

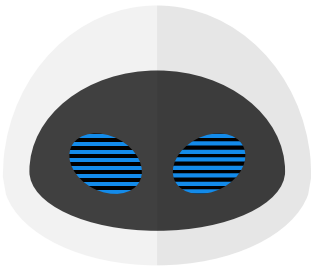


ML can store info in lots of innovative ways

ML Models and Algorithms

Local

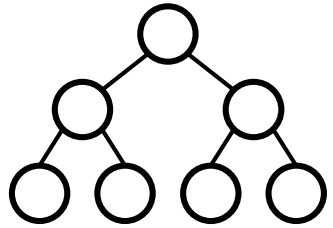




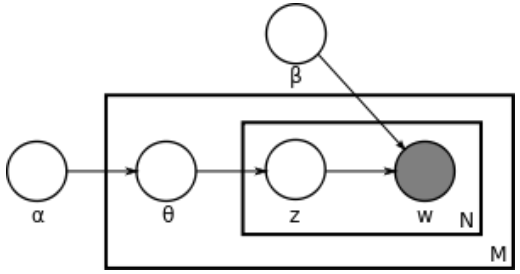
ML can store info in lots of innovative ways

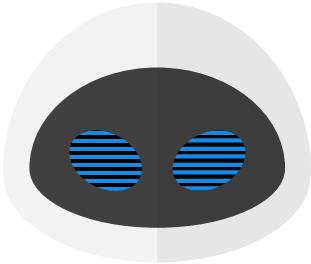
ML Models and Algorithms

Local



Probabilistic/Bayesian

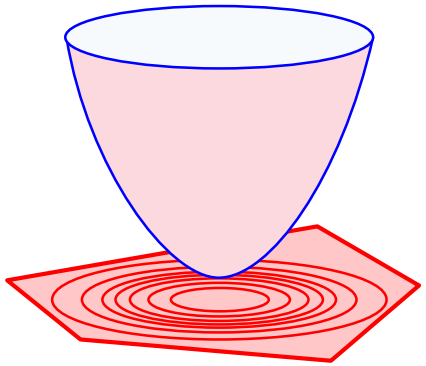




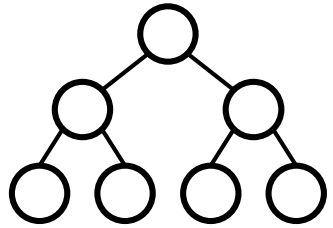
ML can store info in lots of innovative ways

ML Models and Algorithms

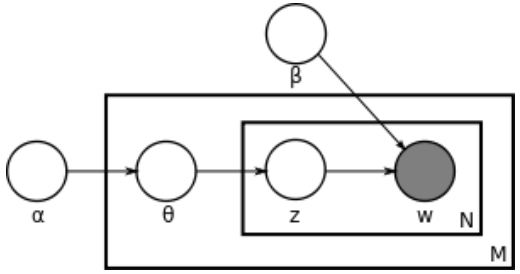
Linear/Opt

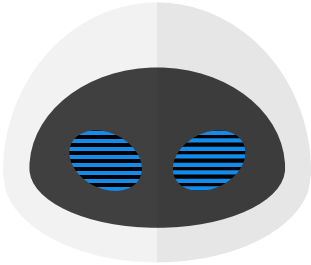


Local



Probabilistic/Bayesian

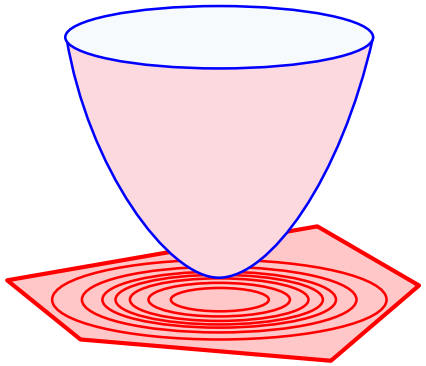




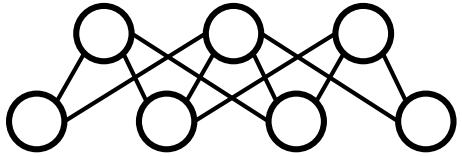
ML can store info in lots of innovative ways

ML Models and Algorithms

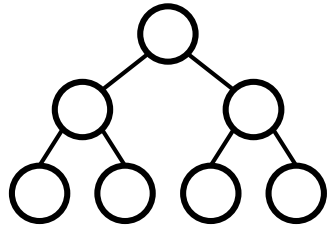
Linear/Opt



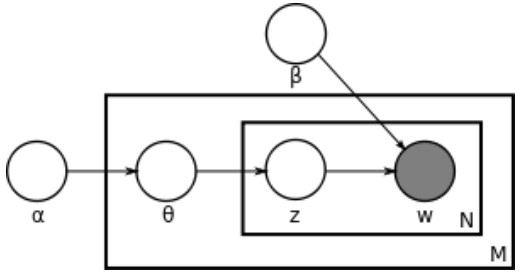
Neural/Deep

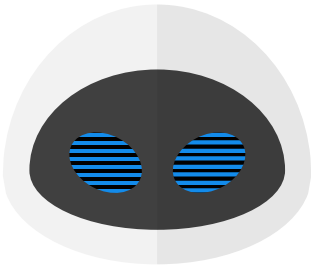


Local



Probabilistic/Bayesian

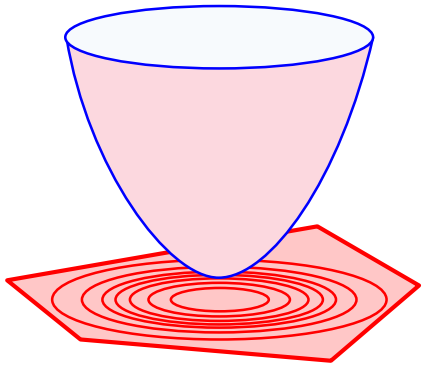




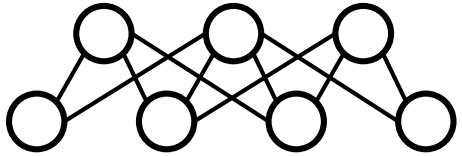
ML can store info in lots of innovative ways

ML Models and Algorithms

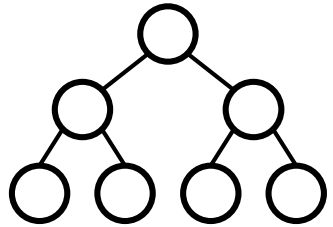
Linear/Opt



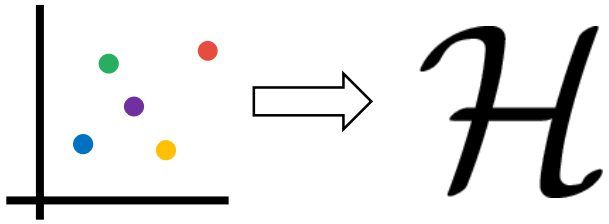
Neural/Deep



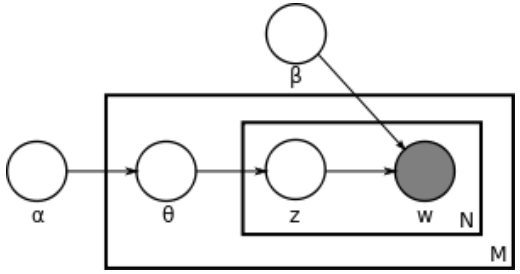
Local

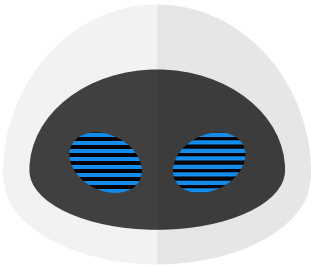


Kernel



Probabilistic/Bayesian

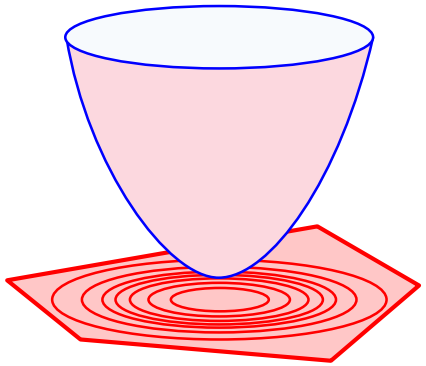




ML can store info in lots of innovative ways

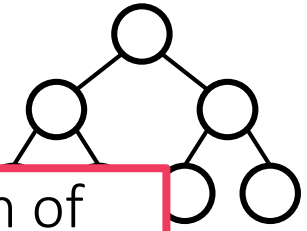
ML Models and Algorithms

Linear/Opt



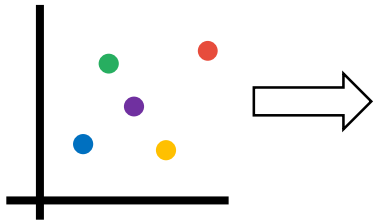
Neural/Deep

Local

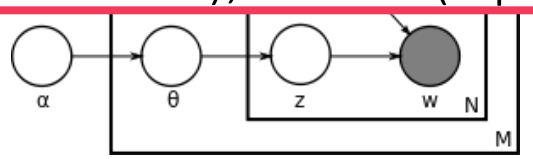


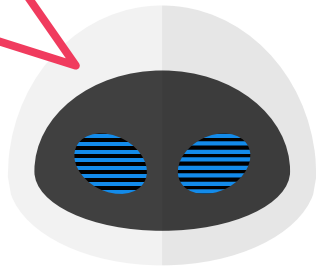
We will learn how to use each of these techniques in the course, but a bit briefly. As before, there are entire courses devoted to each technique e.g. CS772 (Prob ML), CS774 (Opt)

Kernel



$\mathcal{H}$

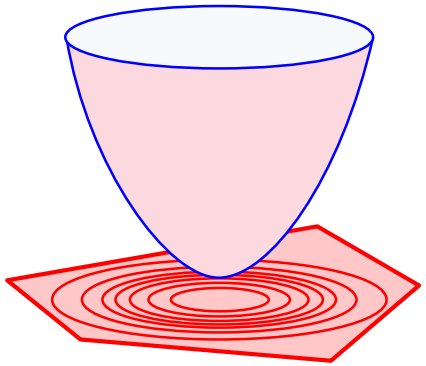




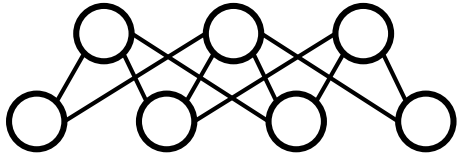
ML can store info in lots of innovative ways

ML Models and Algorithms

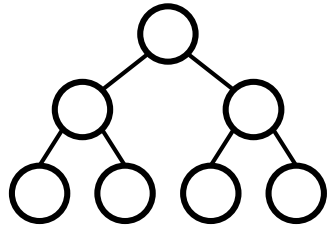
Linear/Opt



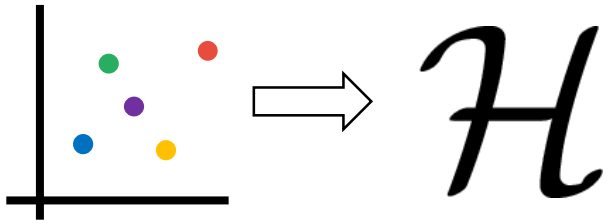
Neural/Deep



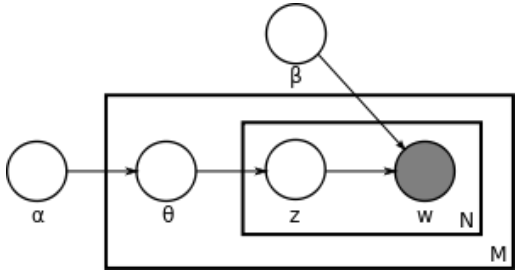
Local

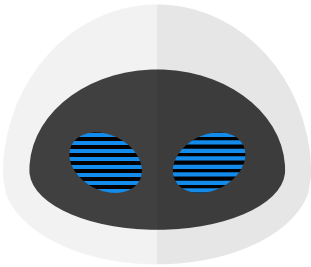


Kernel



Probabilistic/Bayesian

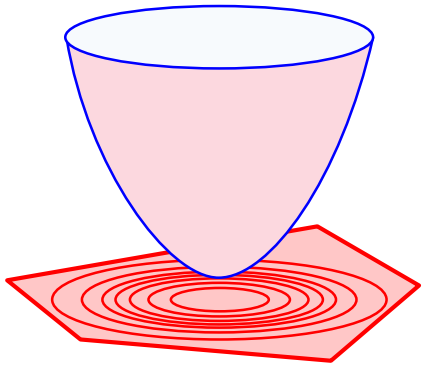




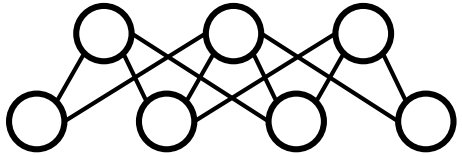
ML can store info in lots of innovative ways

ML Models and Algorithms

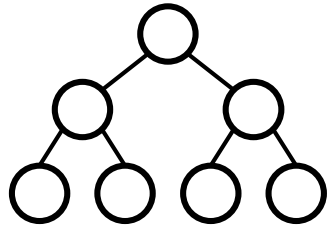
Linear/Opt



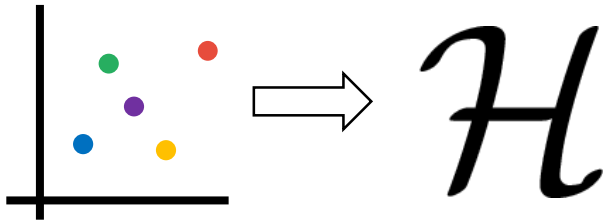
Neural/Deep



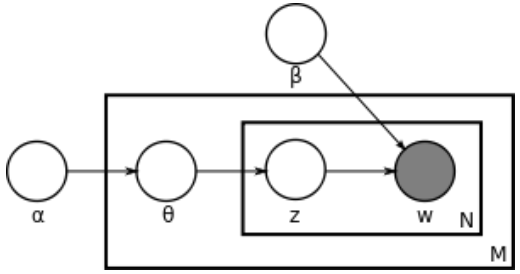
Local

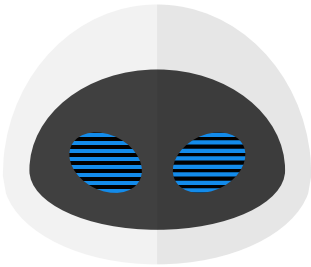


Kernel



Probabilistic/Bayesian





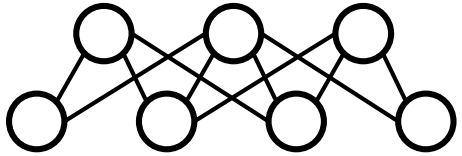
ML can store info in lots of innovative ways

We can mix-n-match these methods too e.g. Bayesian Deep Learning or Kernel Nearest Neighbours (Local)

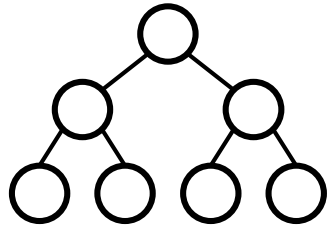
ML Models and Algorithms

Linear/Opt

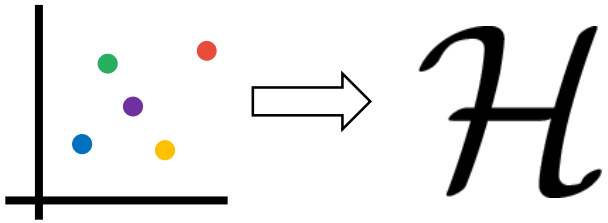
Neural/Deep



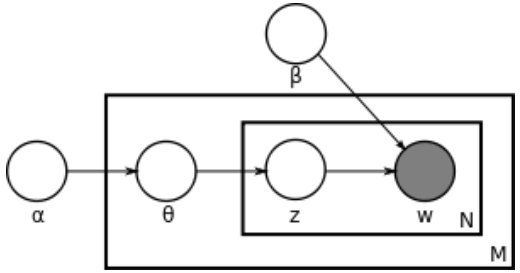
Local

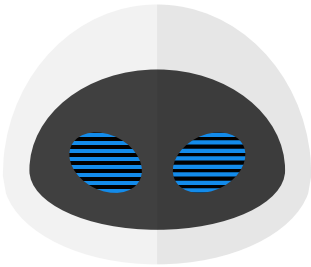


Kernel



Probabilistic/Bayesian



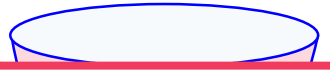


ML can store info in lots of innovative ways

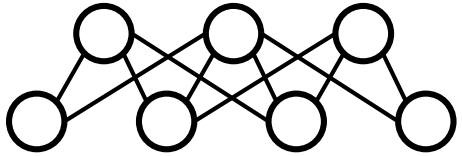
We can mix-n-match these methods too e.g. Bayesian Deep Learning or Kernel Nearest Neighbours (Local)

ML Models and Algorithms

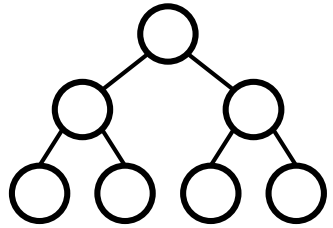
Linear/Opt



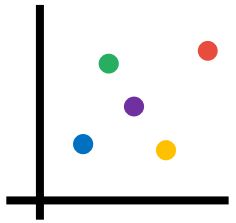
Neural/Deep



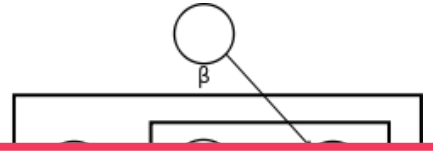
Local

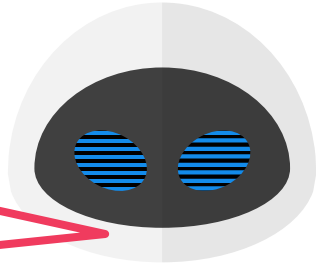


Kernel



Probabilistic/Bayesian





Correct! But we will not be able to cover such advanced methods either

Fantastic Features

- … and how to find them
- What are vectors
- How are vectors used in ML
- Useful operations on vectors

What are features

What are features

Features are a way for us to give input to ML algorithms

What are features

Features are a way for us to give input to ML algorithms

What are features

Features are a way for us to give input to ML algorithms

Since I am basically a computer program, I need you to convert your data into a nice set of numbers

What are features

Features are a way for us to give input to ML algorithms

Most ML algorithms use numerical vectors to represent features

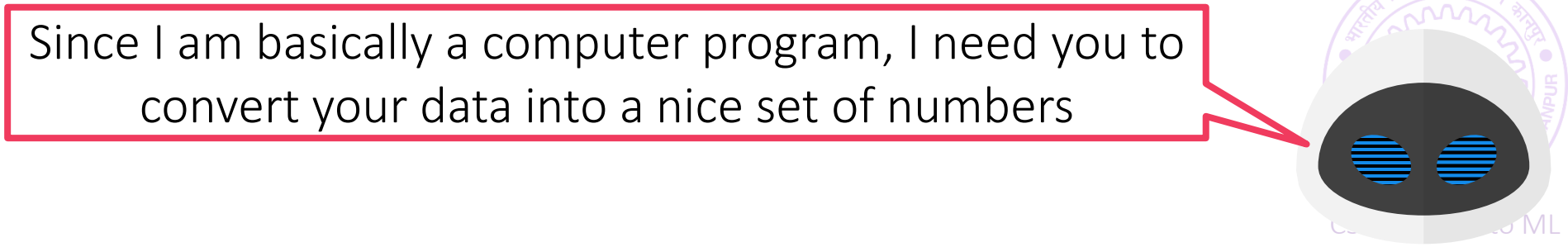
Since I am basically a computer program, I need you to convert your data into a nice set of numbers

What are features

Features are a way for us to give input to ML algorithms

Most ML algorithms use numerical vectors to represent features

For example, in spam classification, every old email (training data) as well as every new email (test data) must be converted into a vector



What are features

Features are a way for us to give input to ML algorithms

Most ML algorithms use numerical vectors to represent features

For example, in spam classification, every old email (training data) as well as every new email (test data) must be converted into a vector

Do you want to go for dinner?

Since I am basically a computer program, I need you to convert your data into a nice set of numbers

What are features

Features are a way for us to give input to ML algorithms

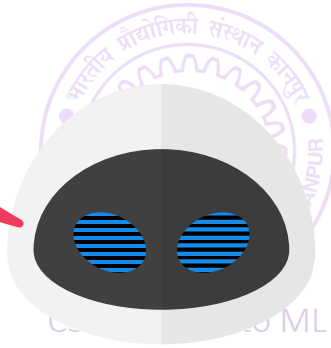
Most ML algorithms use numerical vectors to represent features

For example, in spam classification, every old email (training data) as well as every new email (test data) must be converted into a vector

Do you want to go for dinner?

Do you want to win a million dollars?

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features

Features are a way for us to give input to ML algorithms

Most ML algorithms use numerical vectors to represent features

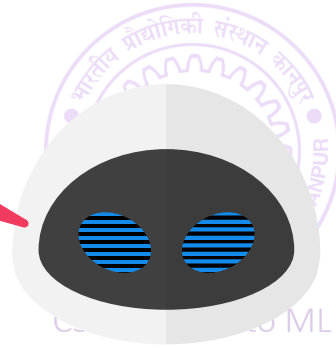
For example, in spam classification, every old email (training data) as well as every new email (test data) must be converted into a vector

Do you want to go for dinner?

Do you want to win a million dollars?

I have a million things to do today!

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features

Features are a way for us to give input to ML algorithms

Most ML algorithms use numerical vectors to represent features

For example, in spam classification, every old email (training data) as well as every new email (test data) must be converted into a vector

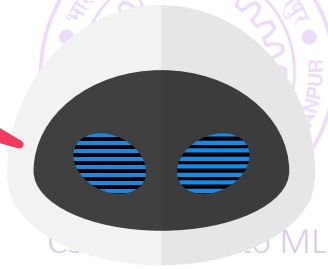
Do you want to go for dinner?

Do you want to win a million dollars?

I have a million things to do today!

Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features

Features are a way for us to give input to ML algorithms

Most ML algorithms use numerical vectors to represent features

For example, in spam classification, every old email (training data) as well as every new email (test data) must be converted into a vector

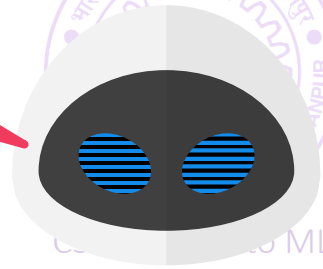
Do you want to go for dinner?

Do you want to win a million dollars?

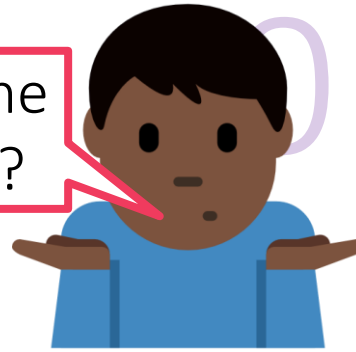
I have a million things to do today!

Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features



Features are a way for us to give input to ML algorithms

Most ML algorithms use numerical vectors to represent features

For example, in spam classification, every old email (training data) as well as every new email (test data) must be converted into a vector

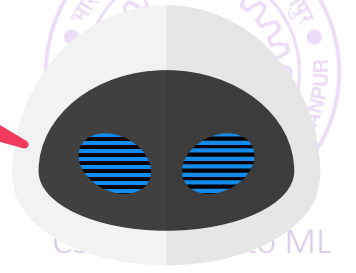
Do you want to go for dinner?

Do you want to win a million dollars?

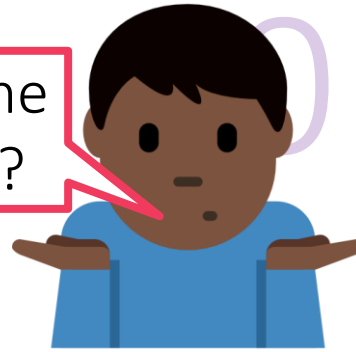
I have a million things to do today!

Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features



Features are a way for us to give input to ML algorithms

ML algorithms use numerical vectors to represent features

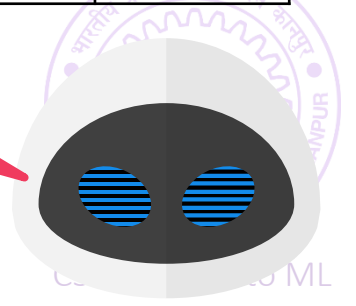
ample, in spam classification, every old email (training data) as every new email (test data) must be converted into a vector

Do you want to go for dinner?

Do you want to win a million dollars?

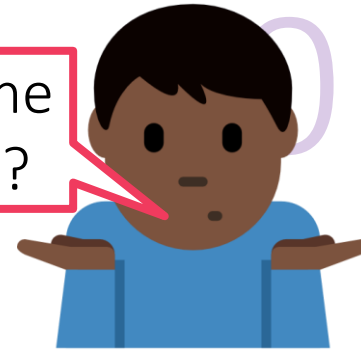
I have a million things to do today!

Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1



What are features

Why did we not keep the word “to” as a feature?



We could have – but it does not carry much information about spam/non-spam since it is such a common word!

Feature [illegible] to ML algorithms

ML al[illegible] use numerical vectors to represent features

ample, in spam classification, every old email (training data) as every new email (test data) must be converted into a vector

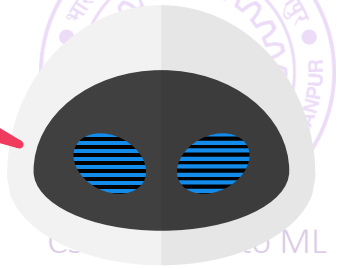
Do you want to go for dinner?

Do you want to win a million dollars?

I have a million things to do today!

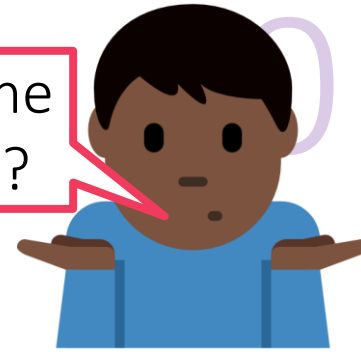
Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features

Why did we not keep the word “to” as a feature?



We could have – but it does not carry much information about spam/non-spam since it is such a common word!

Feature … t to ML algorithms

ML al… …s use numerical vectors to represent features

…ample, in spam classification, every old email (training dat…

every new email (test data) must be converted into a vec…

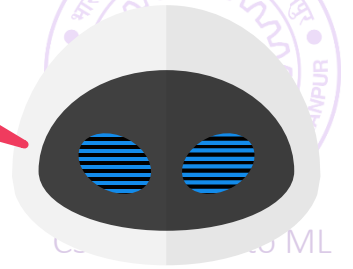
Do you want to go for dinner?

Do you want to win a million dollars?

I have a million things to do today!

Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features

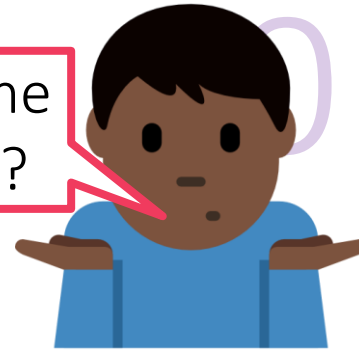
Feature … to ML algorithms

… ML al… …s use numerical vectors to represent features

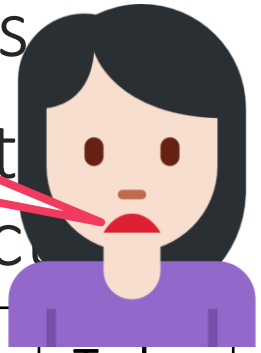
… ample, in spam c… dat…

… every new email (test data) must be converted into a vec…

We could have – but it does not carry much information about spam/non-spam since it is such a common word!



Why did we not keep the word “to” as a feature?



Guys, something is wrong with our feature. The word “do” means different things in two emails

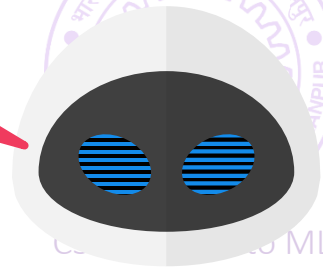
Do you want to go for dinner?

Do you want to win a million dollars?

I have a million things to do today!

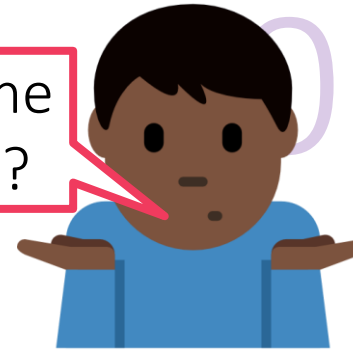
Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1

Since I am basically a computer program, I need you to convert your data into a nice set of numbers



What are features

Why did we not keep the word “to” as a feature?



We could have – but it does not carry much information about spam/non-spam since it is such a common word!

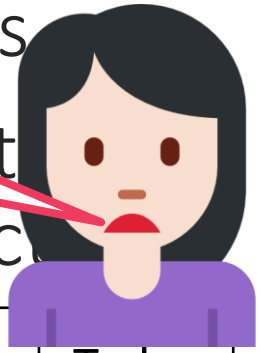
Feature … to ML algorithms

…ML al… …s use numerical vectors to represent features

…ample, in spam c… …dat…

every new email (test data) must be converted into a vec…

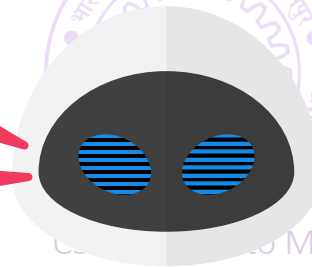
Guys, something is wrong with our feature. The word “do” means different things in two emails



	Do	You	Want	Go	Million	Dollars	Dinner	Today
Do you want to go for dinner?	1	1	1	1	0	0	1	0
Do you want to win a million dollars?	1	1	1	0	1	1	0	0
I have a million things to do today!	1	0	0	0	1	0	0	1

Since I am basically a computer program, I need you to convert your data into a nice set of numbers

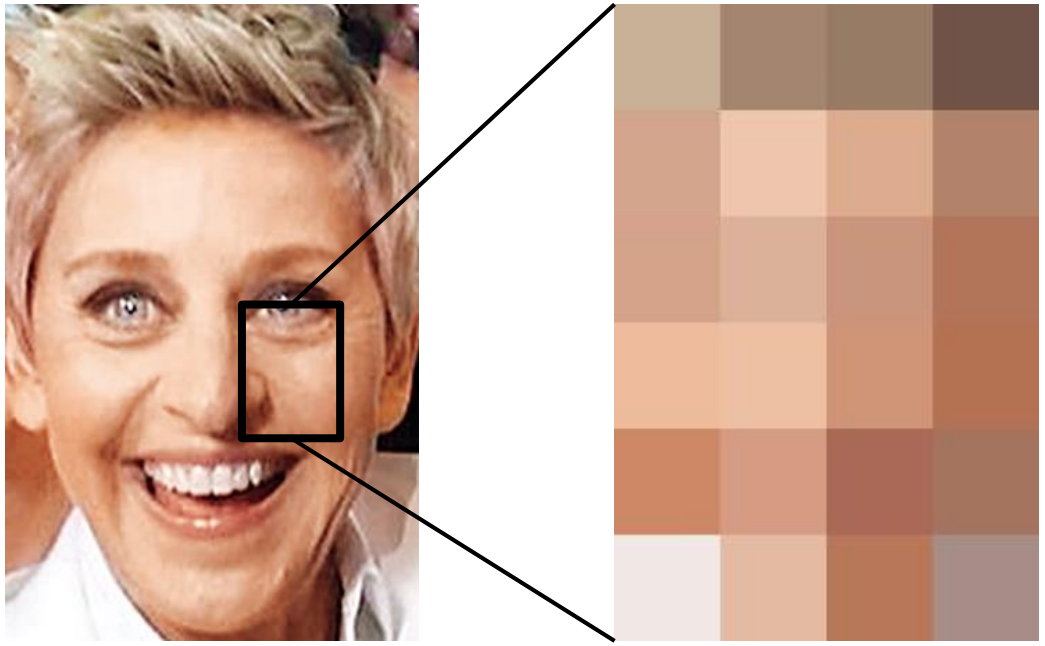
Good catch – this may not be the best feature representation!



What are features

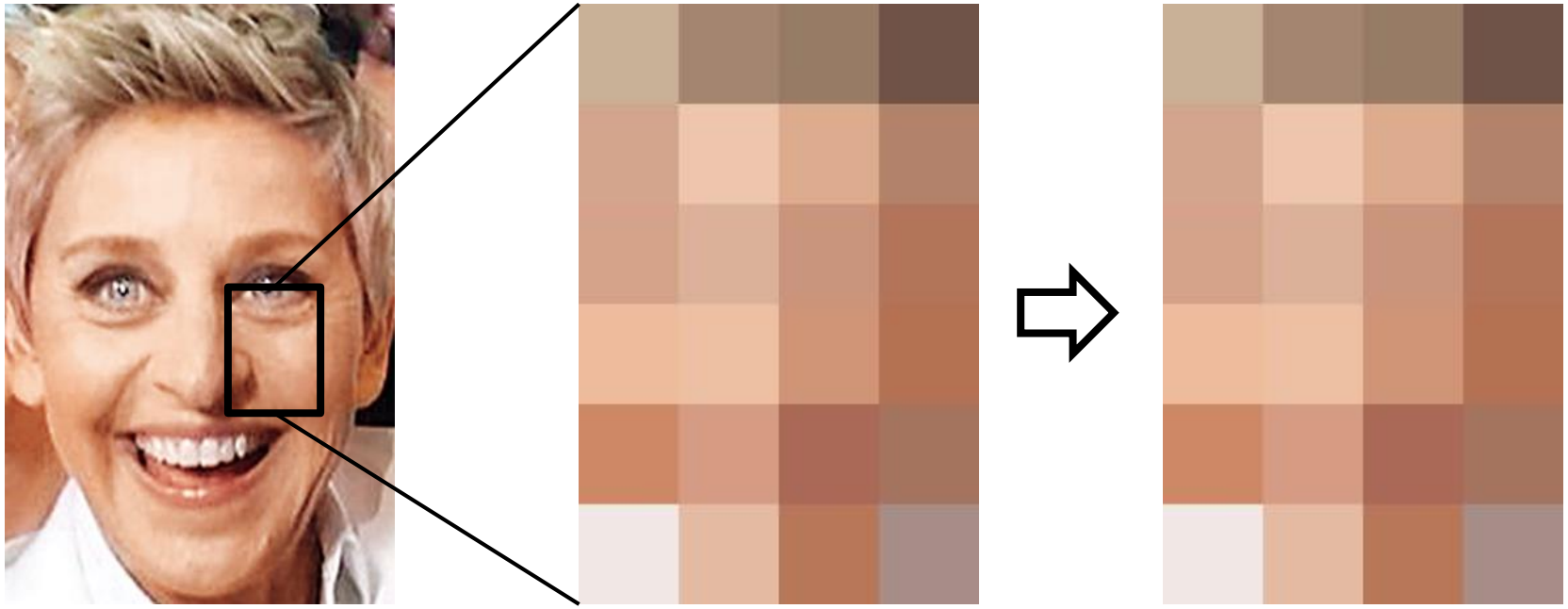
What are features

What are features



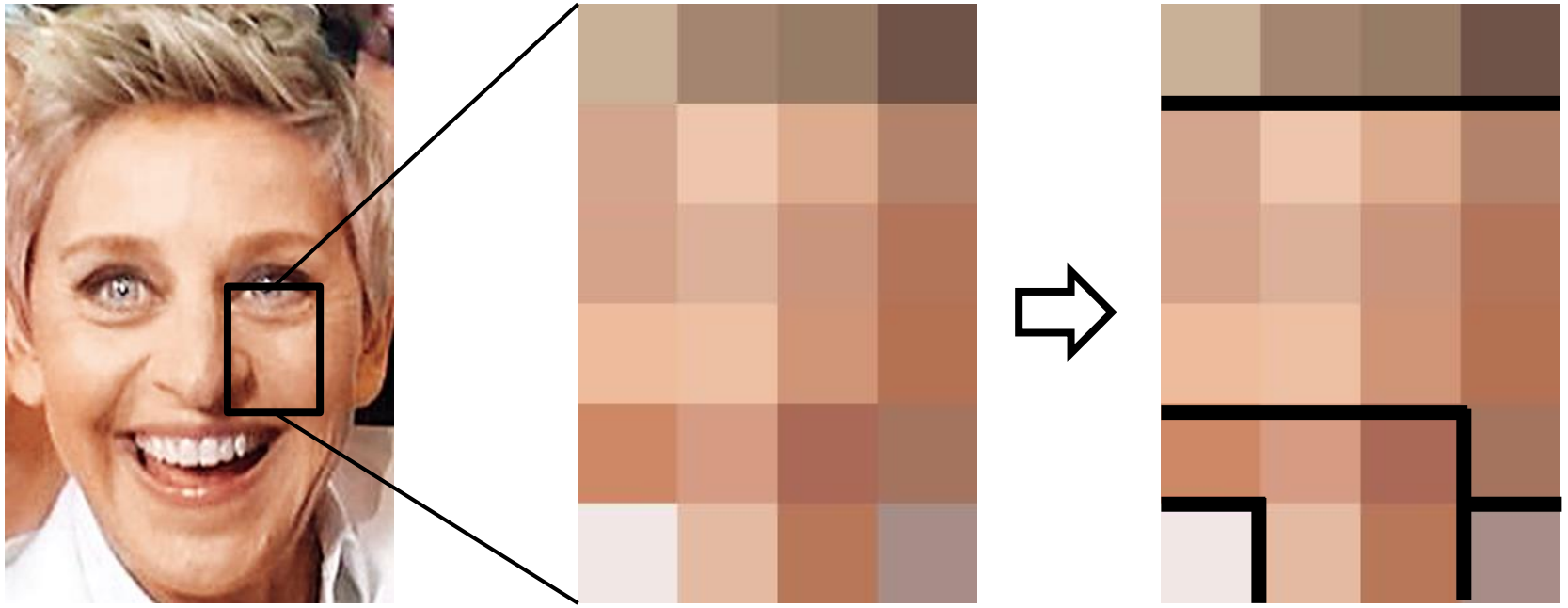
May represent images as a vector/matrix of pixel RGB values

What are features



May represent images as a vector/matrix of pixel RGB values

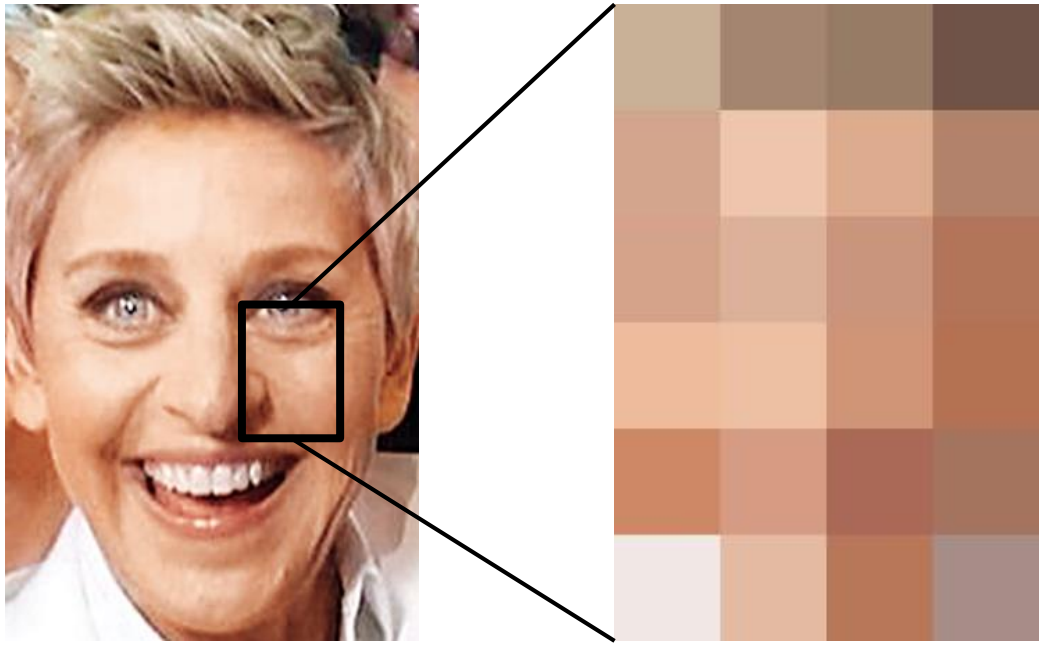
What are features



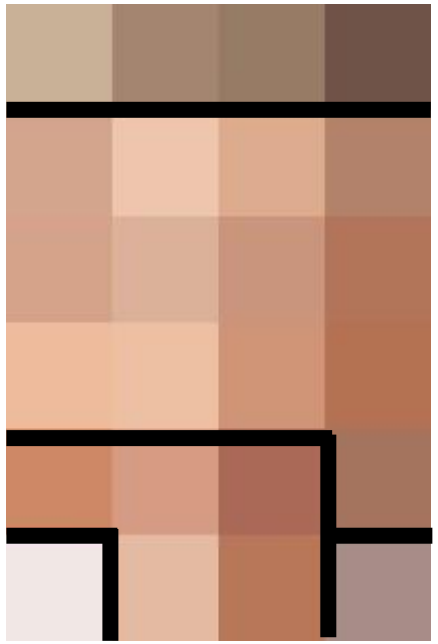
May represent images as a vector/matrix of pixel RGB values

May instead encode images using edges present in them

What are features



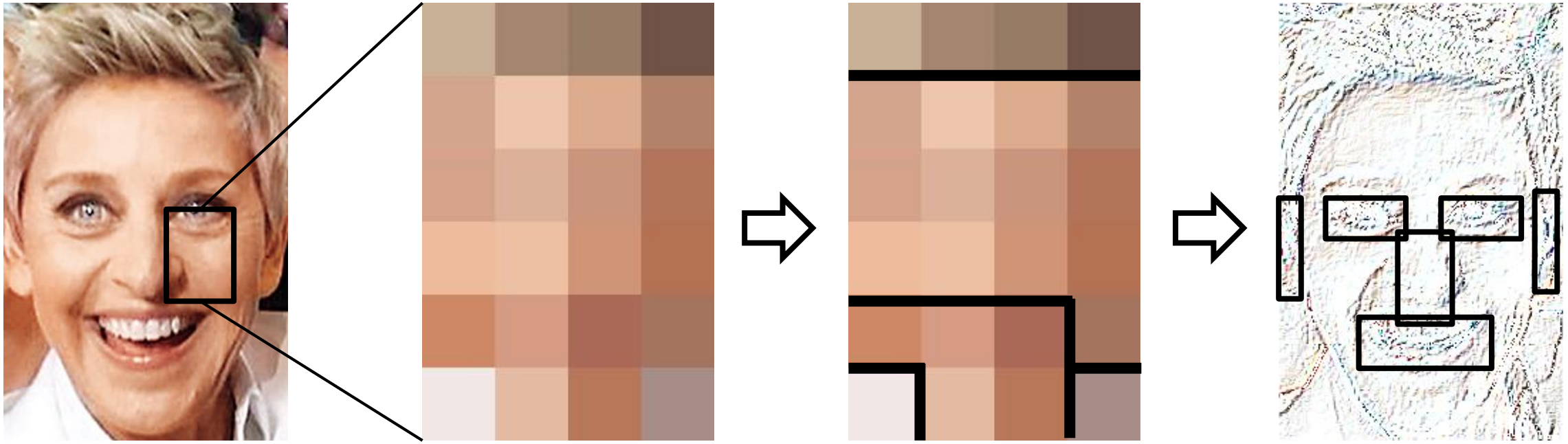
May represent images as a vector/matrix of pixel RGB values



May instead encode images using edges present in them

May instead encode images using presence of eyes/noses/ears

What are features

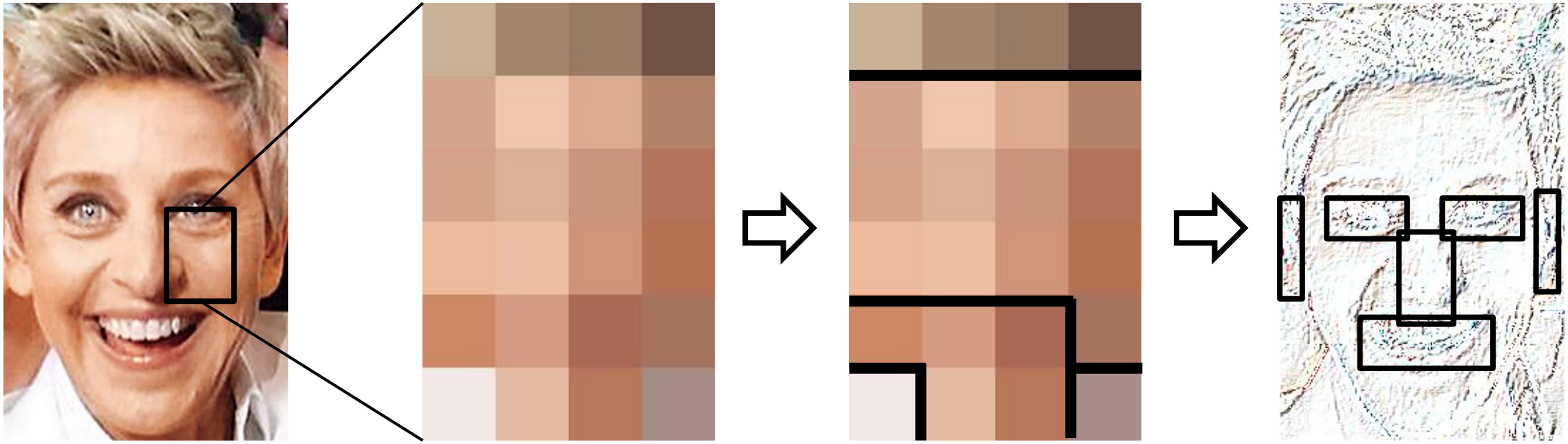


May represent images as a vector/matrix of pixel RGB values

May instead encode images using edges present in them

May instead encode images using presence of eyes/noses/ears

What are features



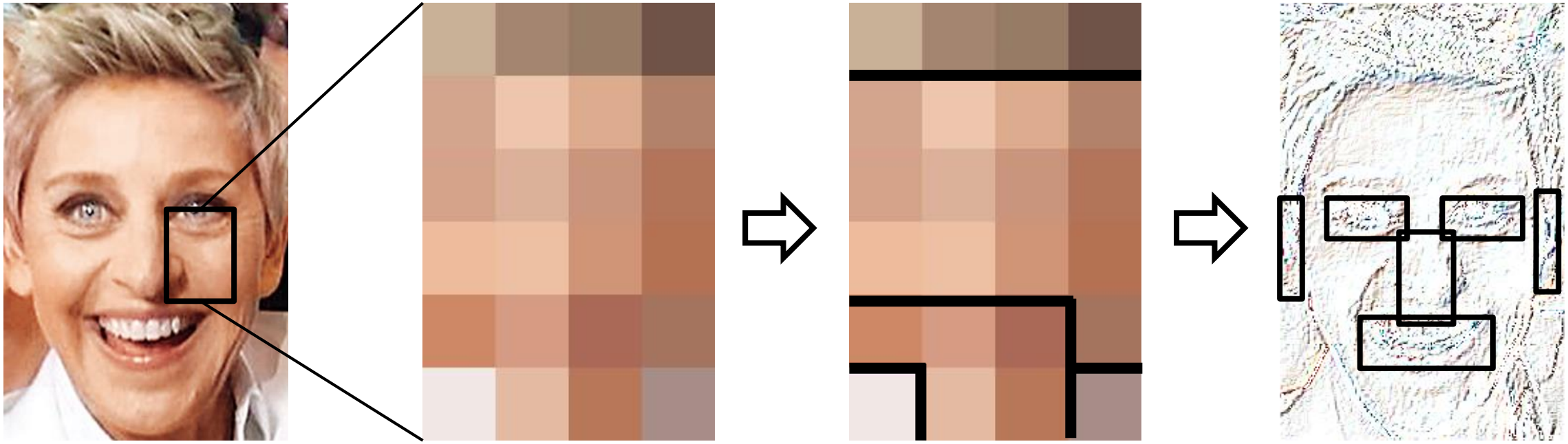
May represent images as a vector/matrix of pixel RGB values

May instead encode images using edges present in them

May instead encode images using presence of eyes/noses/ears

Raw/ Low-level features

What are features



May represent images as a vector/matrix of pixel RGB values

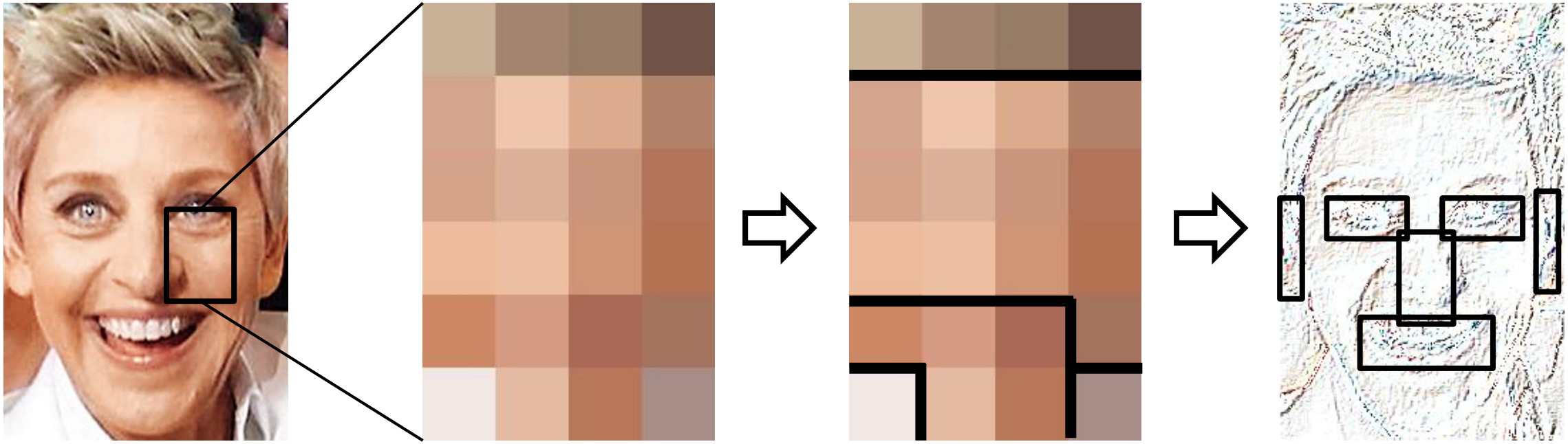
May instead encode images using edges present in them

May instead encode images using presence of eyes/noses/ears

Raw/ Low-level features

Derived/ High-level features

What are features



May represent images as a vector/matrix of pixel RGB values

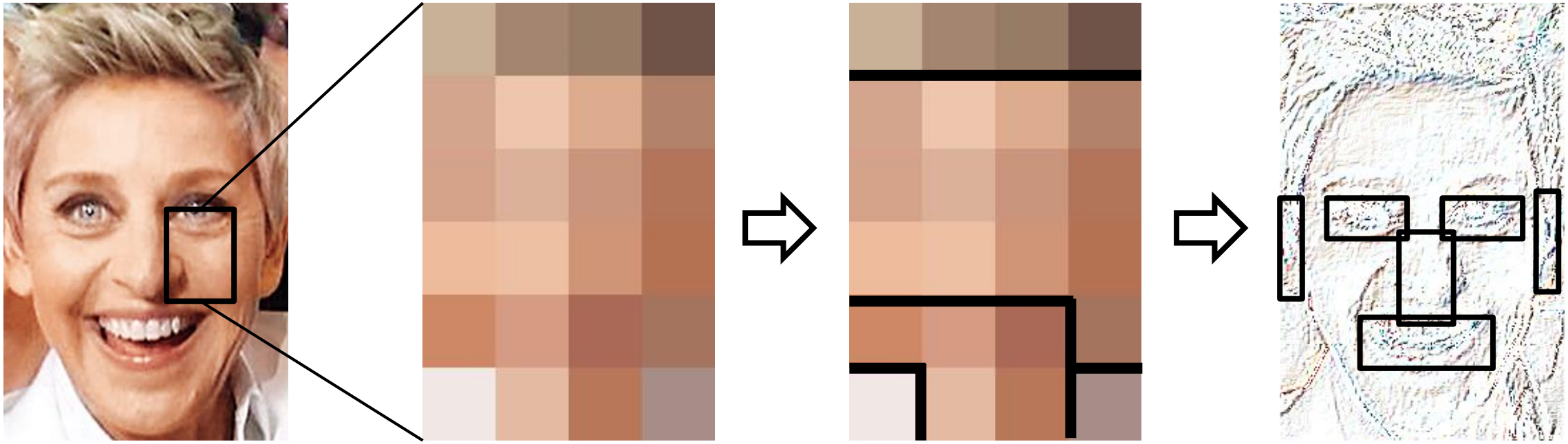
May instead encode images using edges present in them

May instead encode images using presence of eyes/noses/ears

Raw/ Low-level features

Derived/ High-level features

What are features



May represent images as a vector/matrix of pixel RGB values

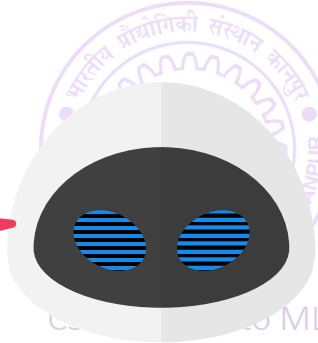
May instead encode images using edges present in them

May instead encode images using presence of eyes/noses/ears

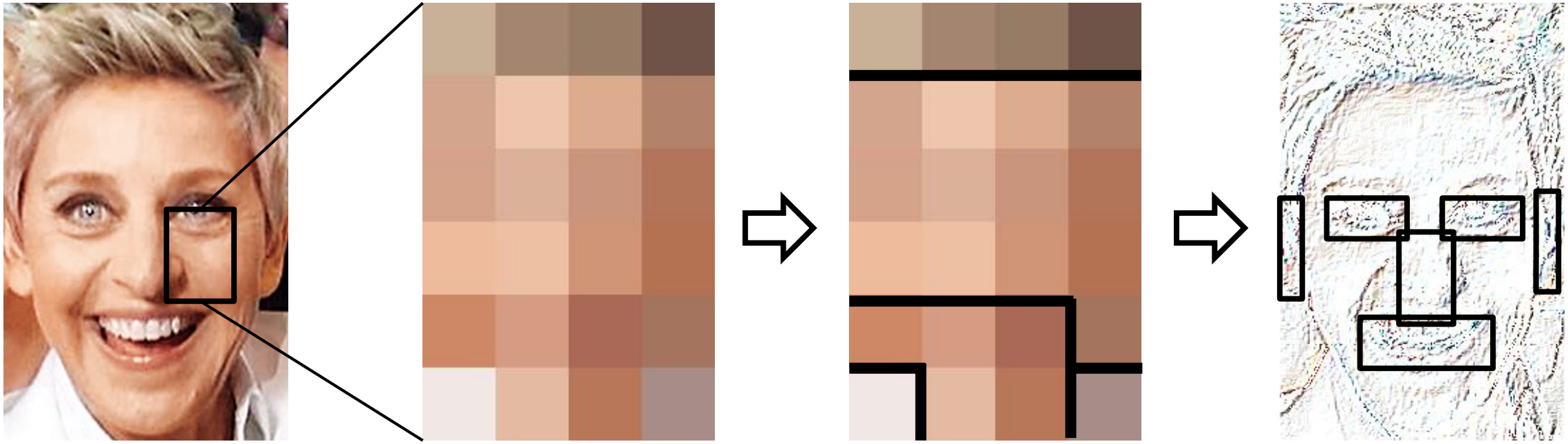
Raw/ Low-level features

Derived/ High-level features

If your features are good, my job becomes twice as easy.
If not, then doing any ML may become impossible!



What are features



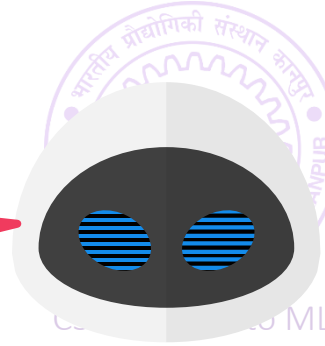
May represent images as a vector/matrix of pixel RGB values

May instead encode images using edges present in them

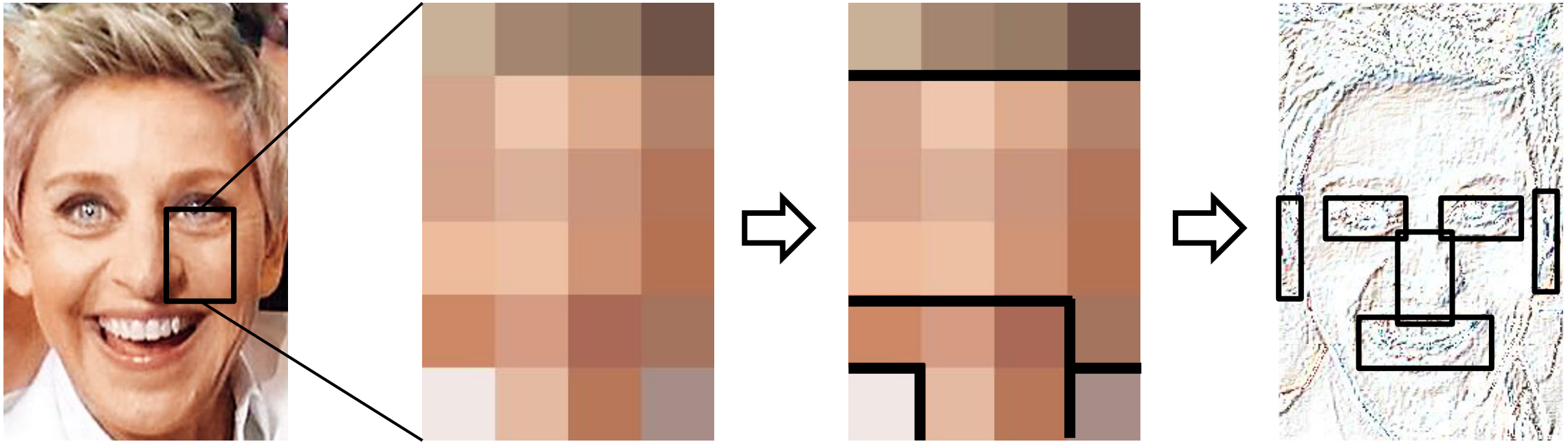
May instead encode images using presence of eyes/noses/ears

Raw/ Low-level features ⇨ Derived/ High-level features

If your features are good, my job becomes twice as easy.
If not, then doing any ML may become impossible!



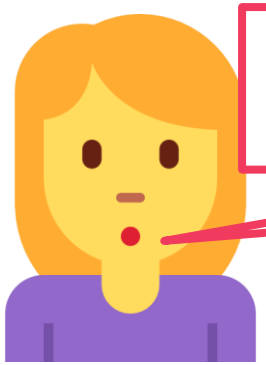
What are features



May represent images as a vector/matrix of pixel RGB values

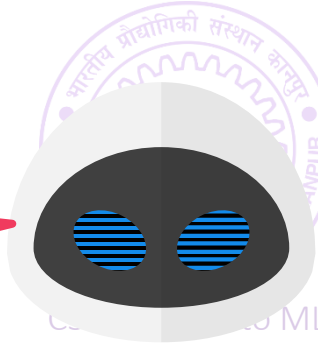
May instead encode images using edges present in them

May instead encode images using presence of eyes/noses/ears



Makes sense. Features are also a way for us to store what we know about data. If we throw away data, naturally doing anything becomes hard!

If your features are good, my job becomes twice as easy. If not, then doing any ML may become impossible!



Types of Features

Types of Features

Numerical Features

Types of Features

Numerical Features

Dangal ★★★★★

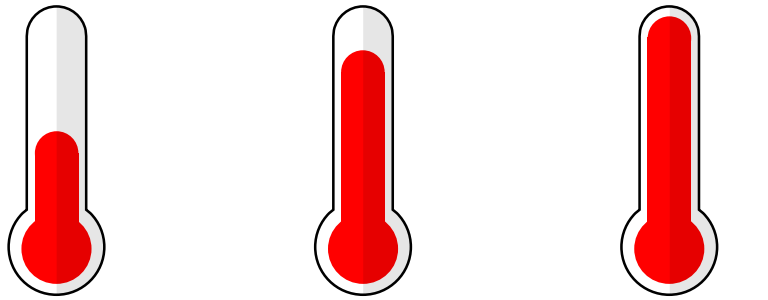
Dhoom 3 ★★★

Types of Features

Numerical Features

Dangal ★★★★★

Dhoom 3 ★★★



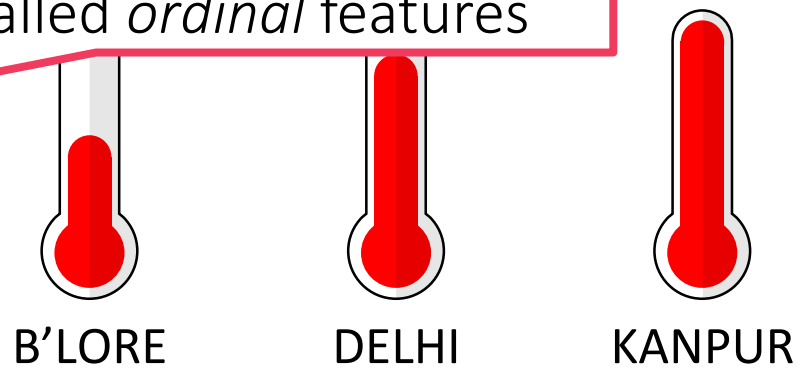
Types of Features

Discrete numerical features often called *ordinal* features

Numerical Features

Dangal ★★★★★

Dhoom 3 ★★★



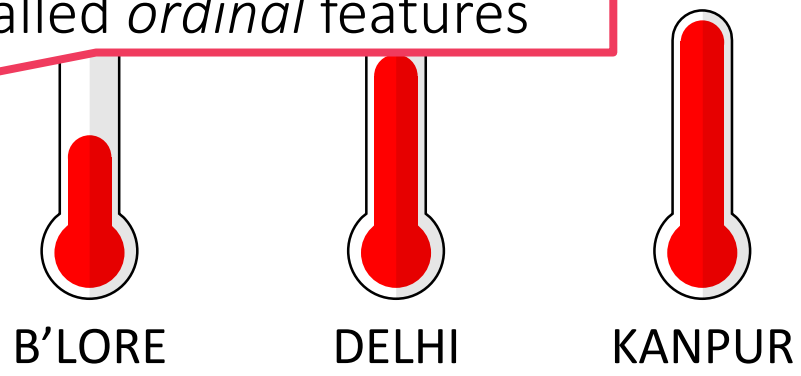
Types of Features

Numerical Features

Dangal ★★★★★

Dhoom 3 ★★★

Discrete numerical features
often called *ordinal* features



Categorical Features

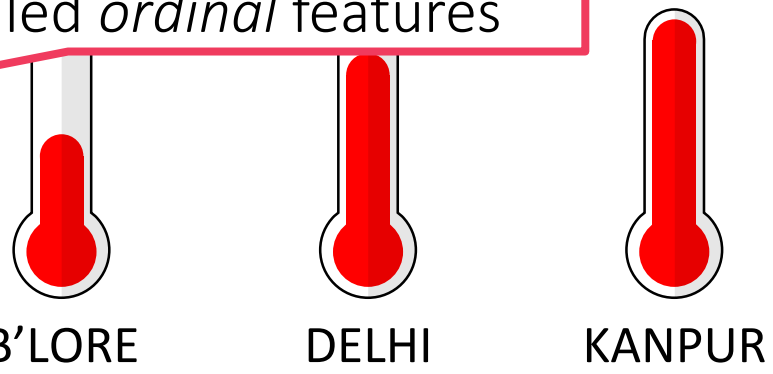
Types of Features

Discrete numerical features often called *ordinal* features

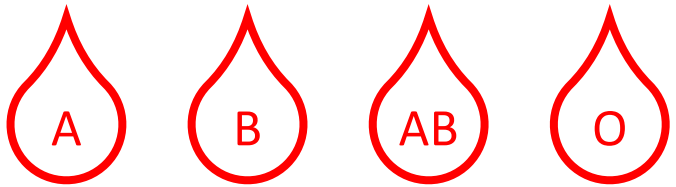
Numerical Features

Dangal ★★★★★

Dhoom 3 ★★★



Categorical Features



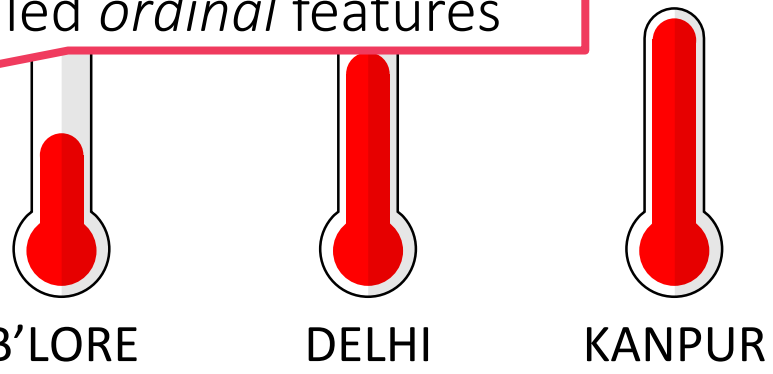
Types of Features

Discrete numerical features often called *ordinal* features

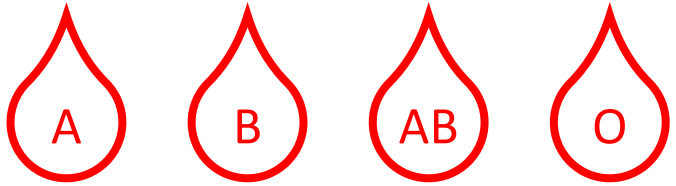
Numerical Features

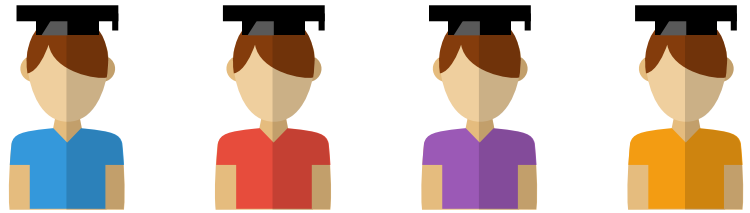
Dangal ★★★★★

Dhoom 3 ★★★



Categorical Features





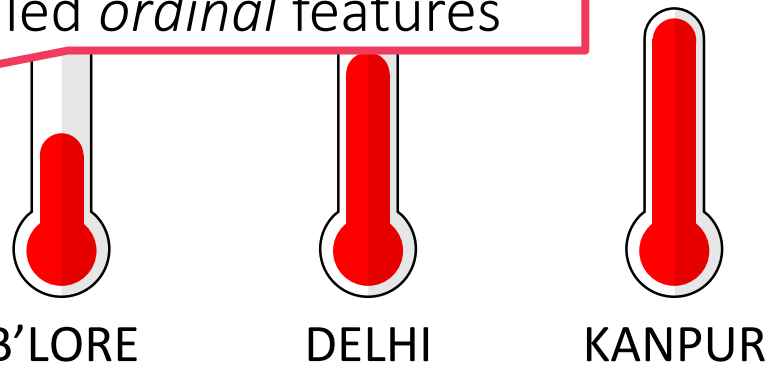
Types of Features

Discrete numerical features often called *ordinal* features

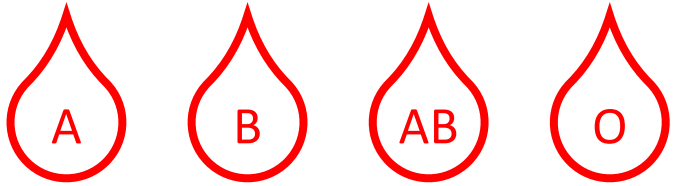
Numerical Features

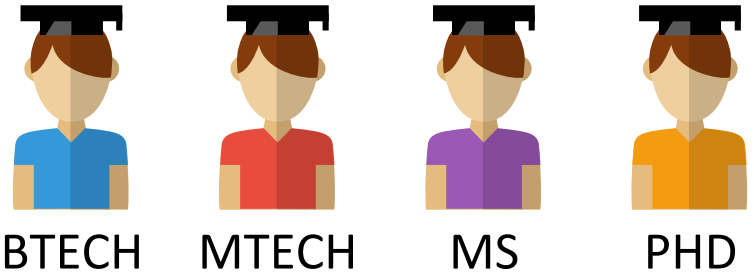
Dangal ★★★★★

Dhoom 3 ★★★



Categorical Features





Relational Features

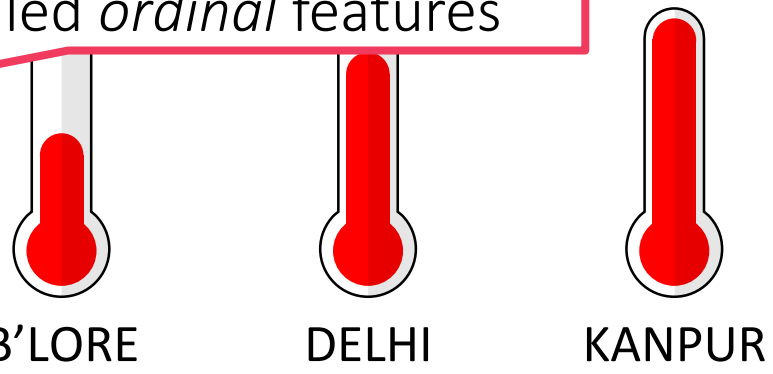
Types of Features

Discrete numerical features often called *ordinal* features

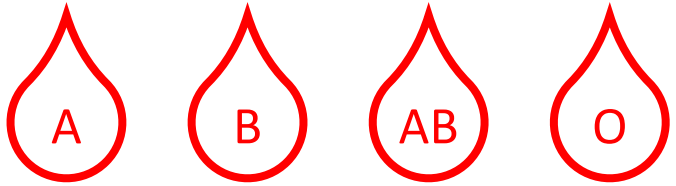
Numerical Features

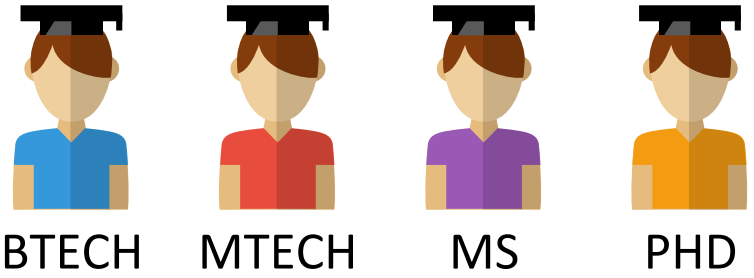
Dangal ★★★★★

Dhoom 3 ★★★



Categorical Features





Relational Features

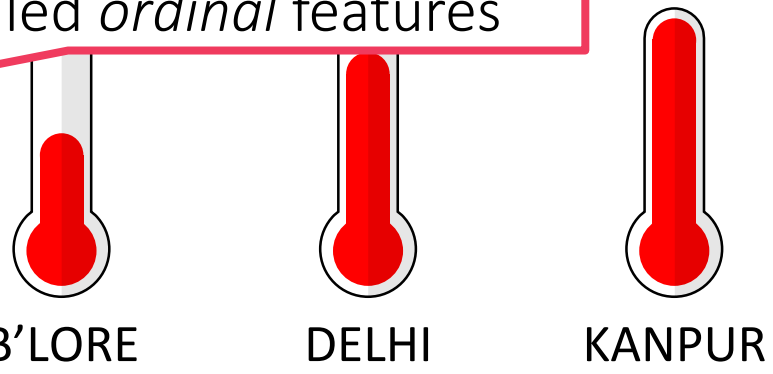
Types of Features

Discrete numerical features often called *ordinal* features

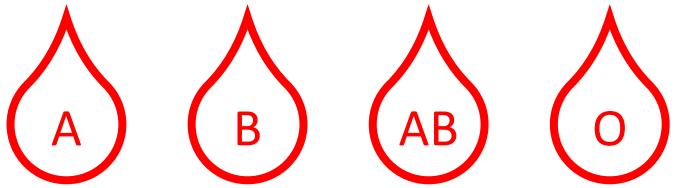
Numerical Features

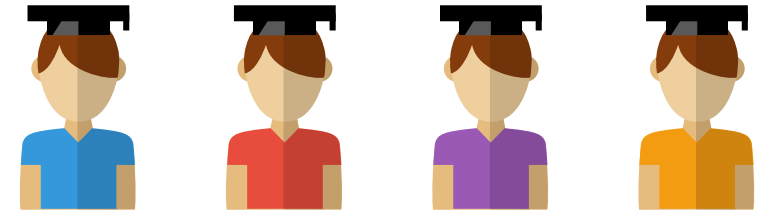
Dangal ★★★★★

Dhoom 3 ★★★

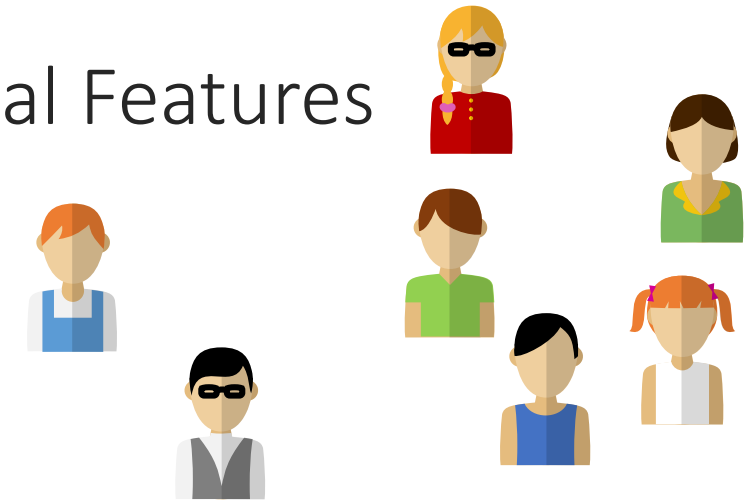


Categorical Features





Relational Features



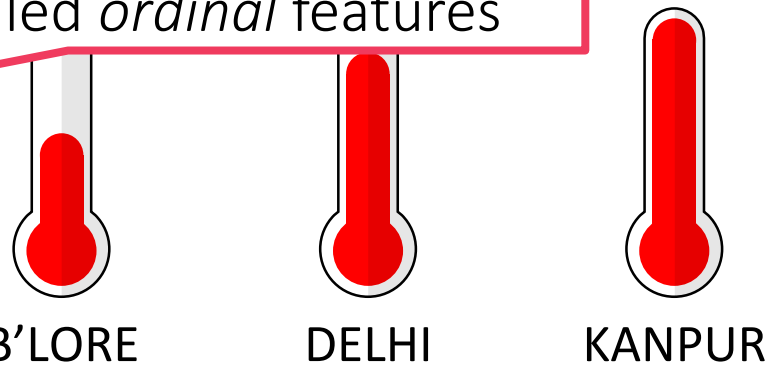
Types of Features

Discrete numerical features often called *ordinal* features

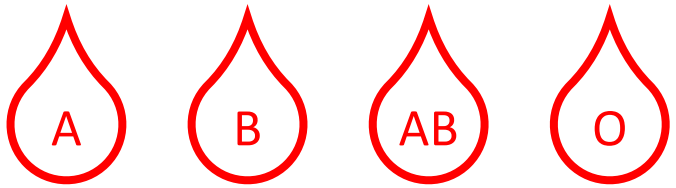
Numerical Features

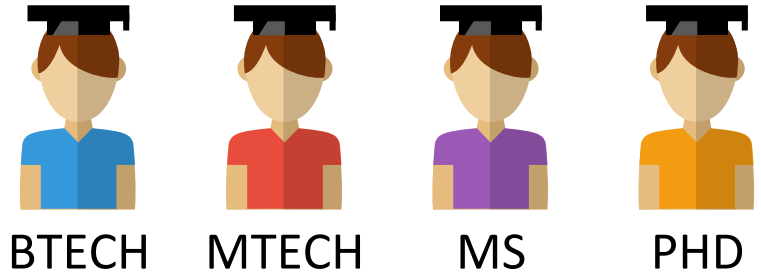
Dangal ★★★★★

Dhoom 3 ★★★

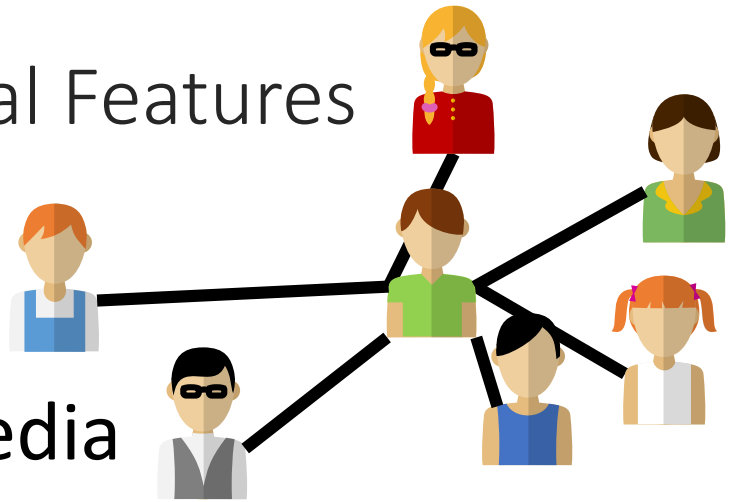


Categorical Features





Relational Features



Social Media

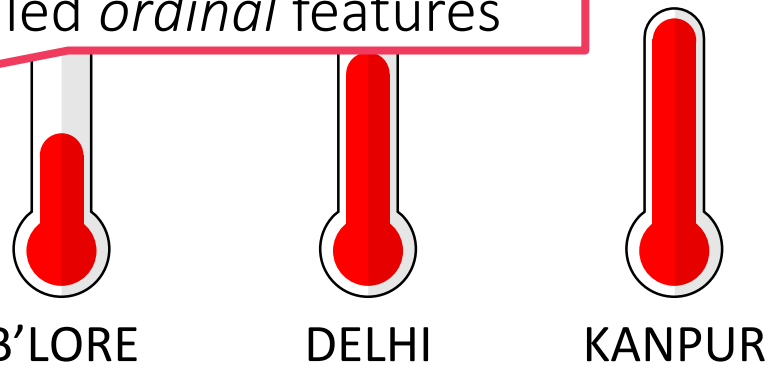
Types of Features

Discrete numerical features often called *ordinal* features

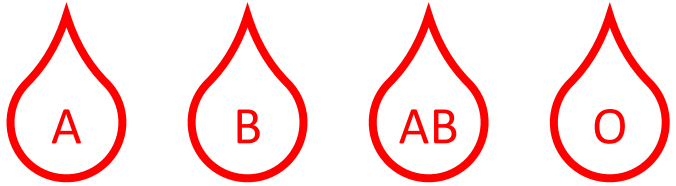
Numerical Features

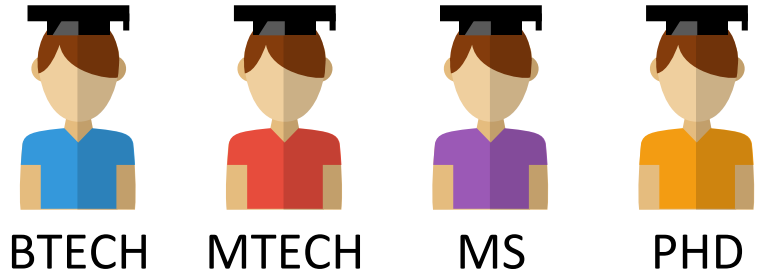
Dangal ★★★★★

Dhoom 3 ★★★

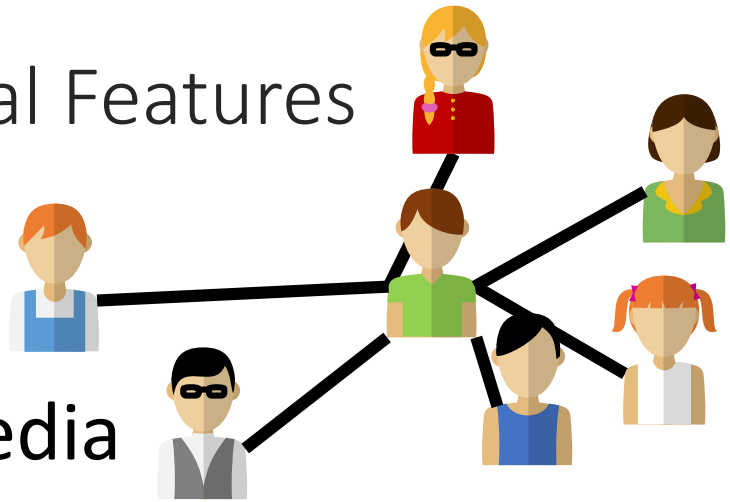


Categorical Features

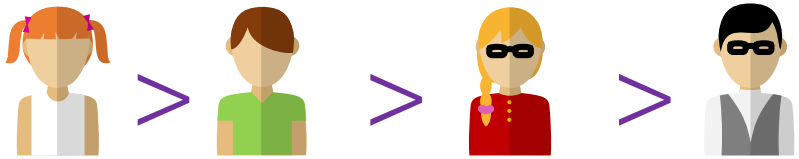




Relational Features



Social Media



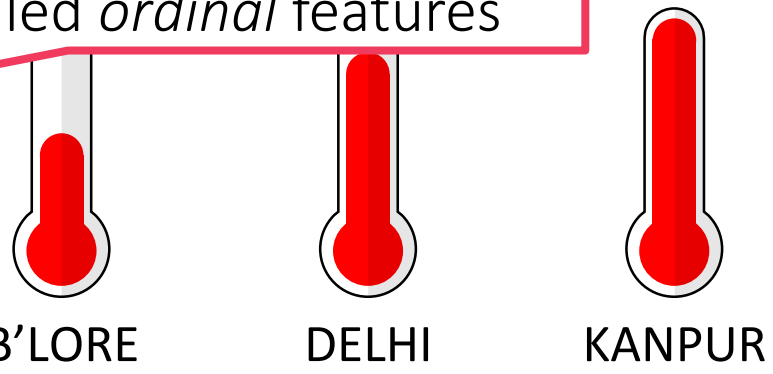
Types of Features

Discrete numerical features often called *ordinal* features

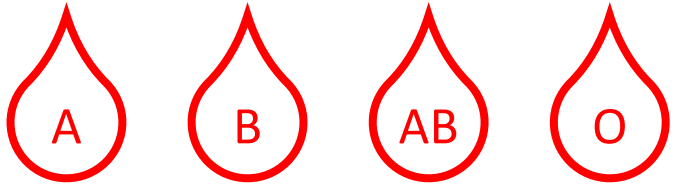
Numerical Features

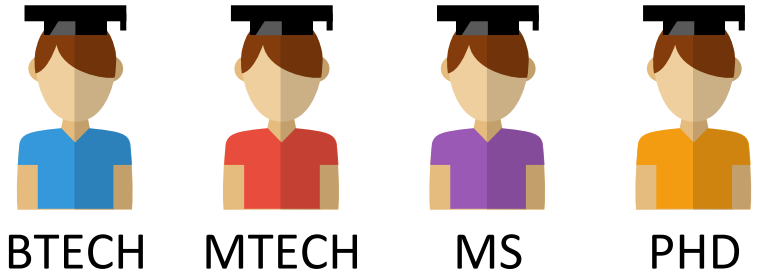
Dangal ★★★★★

Dhoom 3 ★★★

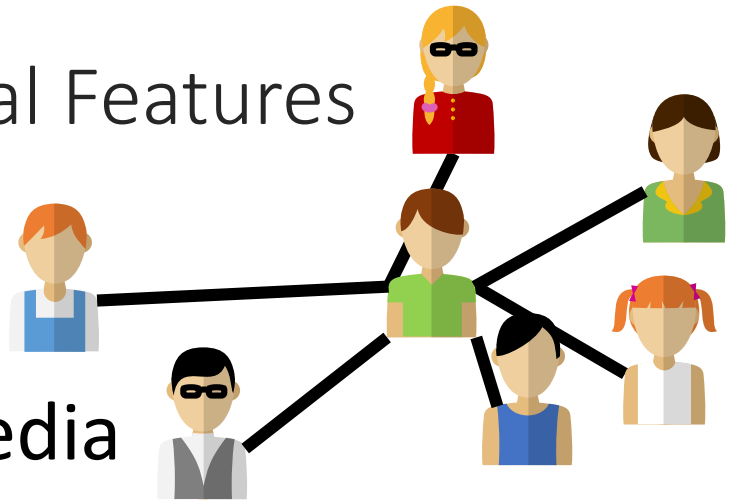


Categorical Features

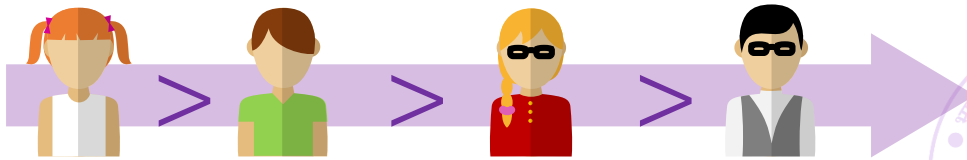




Relational Features



Social Media



Ranking in Class

Derived Features

Derived Features

Bagged/binned features

Derived Features

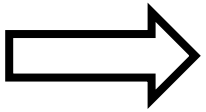
Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

Derived Features

Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



Derived Features

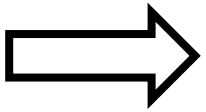
Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

Derived Features

Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

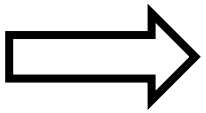


Pooled/aggregated features

Derived Features

Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



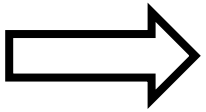
Pooled/aggregated features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

Derived Features

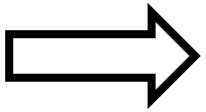
Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



Pooled/aggregated features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

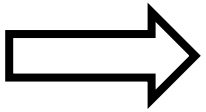


10 (max), 8.67 (avg)

Derived Features

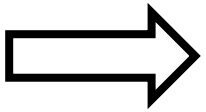
Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



Pooled/aggregated features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



10 (max), 8.67 (avg)

Do you want to go for dinner?

Do you want to win a million dollars?

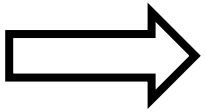
I have a million things to do today!

Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1

Derived Features

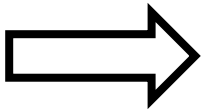
Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



Pooled/aggregated features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



10 (max), 8.67 (avg)

	Do	You	Want	Go	Million	Dollars	Dinner	Today
Do you want to go for dinner?	1	1	1	1	0	0	1	0
Do you want to win a million dollars?	1	1	1	0	1	1	0	0
I have a million things to do today!	1	0	0	0	1	0	0	1

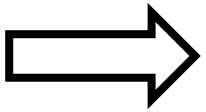
Derived Features

Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

Pooled/aggregated features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)



10 (max), 8.67 (avg)

The way we represented emails in the spam problem is called the *bag-of-words* feature

	Do	You	Want	Go	Million	Dollars	Dinner	Today
Do you want to go for dinner?	1	1	1	1	0	0	1	0
Do you want to win a million dollars?	1	1	1	0	1	1	0	0
I have a million things to do today!	1	0	0	0	1	0	0	1

Derived Features

Bagged/binned features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

Pooled/aggregated features

ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

10 (max), 8.67 (avg)

The way we represented emails in the spam problem is called the *bag-of-words* feature

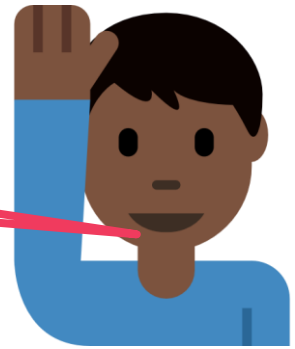
	Do	You	Want	Go	Million	Dollars	Dinner	Today
Do you want to go for dinner?	1	1	1	1	0	0	1	0
Do you want to win a million dollars?	1	1	1	0	1	1	0	0
I have a million things to do today!	1	0	0	0	1	0	0	1

Derived Features

Bagged/binned features

ESC101 (A), ESO207 (B), CS2
CS340 (C), MSO201 (A), CS7

As we saw, the bag-of-words (also called BoW) is not always a very good feature (it confuses polysemous words). However, it is still extremely popular due to its simplicity.



Pooled/aggregated features

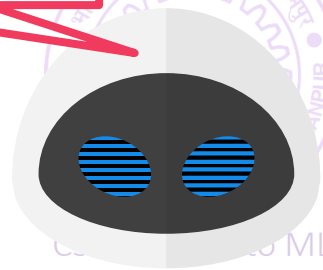
ESC101 (A), ESO207 (B), CS220 (B), CS340 (C), MSO201 (A), CS771 (A)

⇒ 10 (max), 8.67 (avg)

The way we represented emails in the spam problem is called the *bag-of-words* feature

Do you want to go for dinner?
Do you want to win a million dollars?
I have a million things to do today!

Do	You	Want	Go	Million	Dollars	Dinner	Today
1	1	1	1	0	0	1	0
1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1



Feature Selection??

Feature Selection??

Feature Selection??

(EXPENDITURE)

Feature Selection??

(EXPENDITURE)

Feature Selection??

DIET (VEG/NON-VEG)
EYE COLOR
TOTAL INCOME
EDUCATION LEVEL
FAMILY SIZE
HOUSE NO (ODD/EVEN)
NO OF CHILDREN
RENTED/SELF OWNED
SURNAME LENGTH

(EXPENDITURE)

Feature Selection??

- ✓ DIET (VEG/NON-VEG)
- ✗ EYE COLOR
- ✓ TOTAL INCOME
- ✓ EDUCATION LEVEL
- ✓ FAMILY SIZE
- ✗ HOUSE NO (ODD/EVEN)
- ✓ NO OF CHILDREN
- ✓ RENTED/SELF OWNED
- ✗ SURNAME LENGTH

(EXPENDITURE)

Feature Selection??

Useful for predicting expenditure

- ✓ DIET (VEG/NON-VEG)
- ✗ EYE COLOR
- ✓ TOTAL INCOME
- ✓ EDUCATION LEVEL
- ✓ FAMILY SIZE
- ✗ HOUSE NO (ODD/EVEN)
- ✓ NO OF CHILDREN
- ✓ RENTED/SELF OWNED
- ✗ SURNAME LENGTH

(EXPENDITURE)

Feature Selection??

Useful for predicting expenditure

Not useful for predicting expenditure

- DIET (VEG/NON-VEG)
- EYE COLOR
- TOTAL INCOME
- EDUCATION LEVEL
- FAMILY SIZE
- HOUSE NO (ODD/EVEN)
- NO OF CHILDREN
- RENTED/SELF OWNED
- SURNAME LENGTH

(EXPENDITURE)

Feature Selection??

Useful for predicting expenditure

Not useful for predicting expenditure

- DIET (VEG/NON-VEG)
- EYE COLOR
- TOTAL INCOME
- EDUCATION LEVEL
- FAMILY SIZE
- HOUSE NO (ODD/EVEN)
- NO OF CHILDREN
- RENTED/SELF OWNED
- SURNAME LENGTH

(EXPENDITURE)

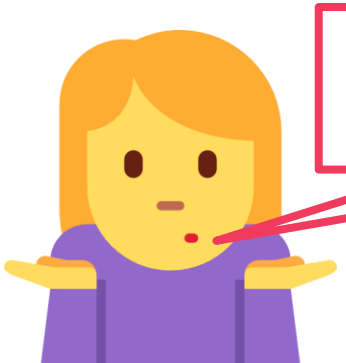
Feature Selection??

Useful for predicting expenditure

Not useful for predicting expenditure

- DIET (VEG/NON-VEG)
- EYE COLOR
- TOTAL INCOME
- EDUCATION LEVEL
- FAMILY SIZE
- HOUSE NO (ODD/EVEN)
- NO OF CHILDREN
- RENTED/SELF OWNED
- SURNAME LENGTH

(EXPENDITURE)



Can we perform automatic feature selection?

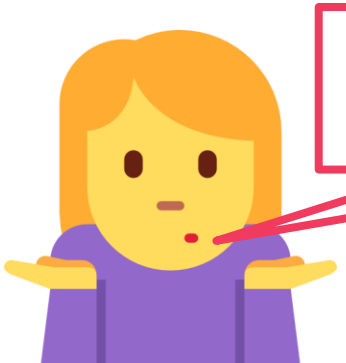
Feature Selection??

Useful for predicting expenditure

Not useful for predicting expenditure

- DIET (VEG/NON-VEG)
- EYE COLOR
- TOTAL INCOME
- EDUCATION LEVEL
- FAMILY SIZE
- HOUSE NO (ODD/EVEN)
- NO OF CHILDREN
- RENTED/SELF OWNED
- SURNAME LENGTH

(EXPENDITURE)



Can we perform automatic feature selection?

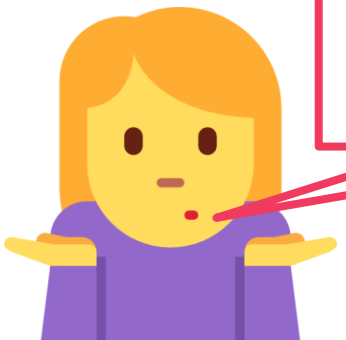
Feature Selection??

Useful for predicting expenditure

Not useful for predicting expenditure

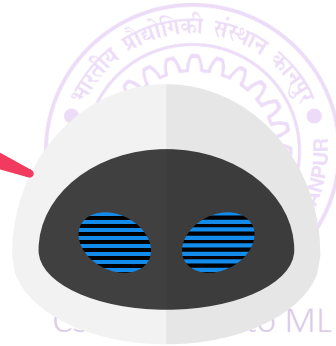
- DIET (VEG/NON-VEG)
- EYE COLOR
- TOTAL INCOME
- EDUCATION LEVEL
- FAMILY SIZE
- HOUSE NO (ODD/EVEN)
- NO OF CHILDREN
- RENTED/SELF OWNED
- SURNAME LENGTH

(EXPENDITURE)



Can we perform automatic feature selection?

Yes indeed, but those methods are a bit advanced for now!

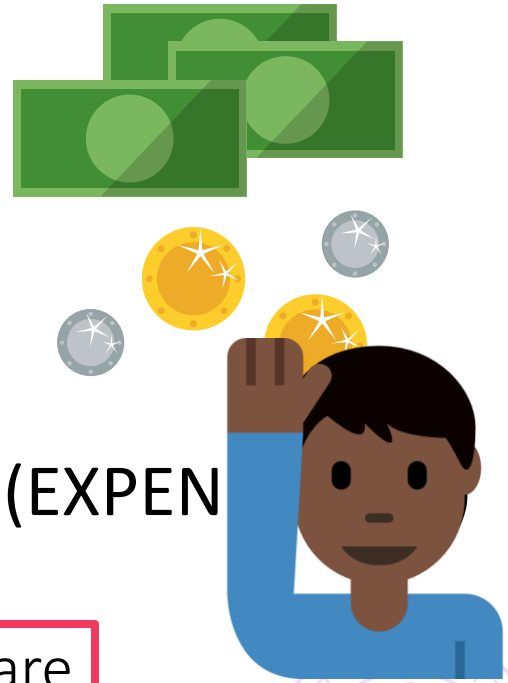


Feature Selection??

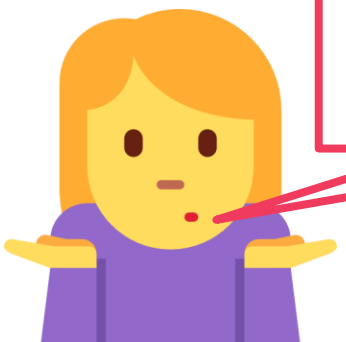
Useful for predicting expenditure

Not useful for predicting expenditure

- DIET (VEG/NON-VEG)
- EYE COLOR
- TOTAL INCOME
- EDUCATION LEVEL
- FAMILY SIZE
- HOUSE NO (ODD/EVEN)
- NO OF CHILDREN
- RENTED/SELF OWNED
- SURNAME LENGTH

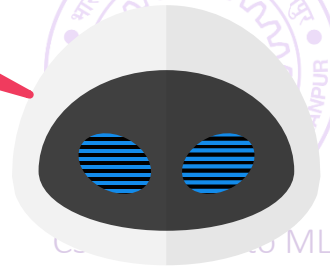


(EXPEN



Can we perform automatic feature selection?

Yes indeed, but those methods are a bit advanced for now!



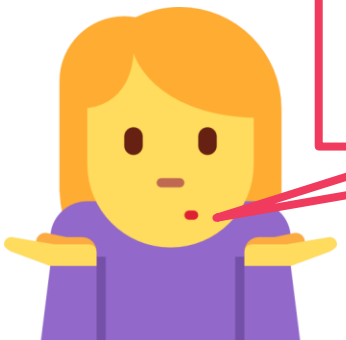
Feature Selection??

Useful for predicting expenditure

Not useful for predicting expenditure

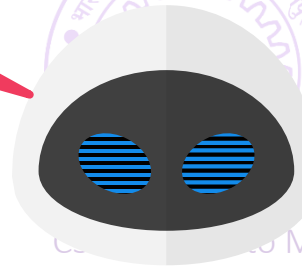
- DIET (VEG/NON-VEG)
- EYE COLOR
- TOTAL INCOME
- EDUCATION LEVEL
- FAMILY SIZE
- HOUSE NO (ODD/EVEN)
- NO OF CHI
- RENTED/SE
- SURNAME

In fact, one reason for the success of deep learning is that it learns good features themselves!



Can we perform automatic feature selection?

Yes indeed, but those methods are a bit advanced for now!



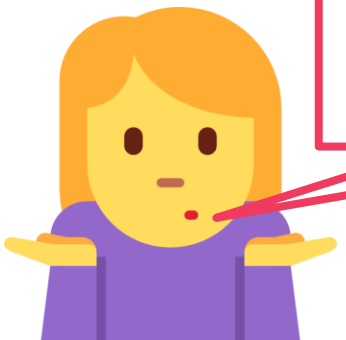
Feature Selection??

Useful for predicting expenditure

Not useful for predicting expenditure

- ✓ DIET (VEG/NON-VEG)
- ✗ EYE COLOR
- ✓ TOTAL INCOME
- ✓ EDUCATION LEVEL
- ✓ FAMILY SIZE
- ✗ HOUSE NO (ODD/EVEN)
- ✓ NO OF CHI
- ✓ RENTED/SE
- ✗ SURNAME

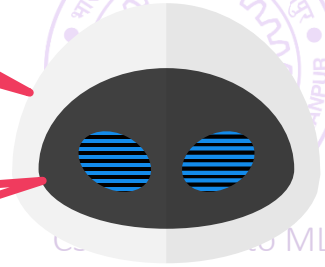
In fact, one reason for the success of deep learning is that it learns good features themselves!



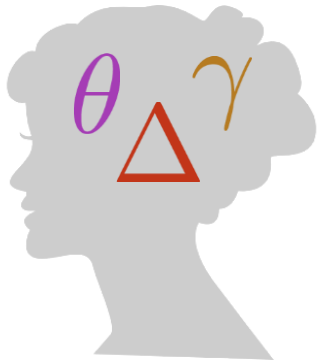
Can we perform automatic feature selection?

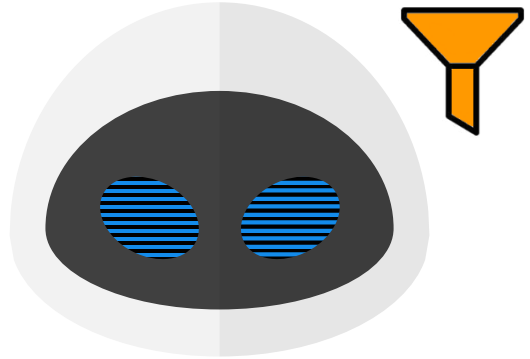
Yes indeed, but those methods are a bit advanced for now!

True, but more on deep learning later! For now, back to basics.

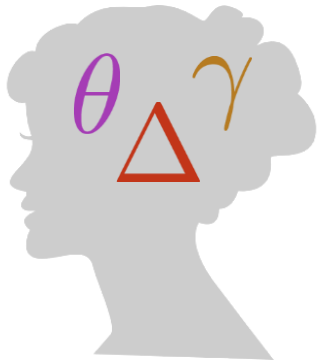


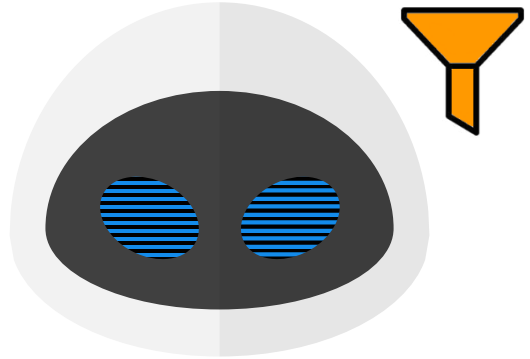
A bit of caution with features





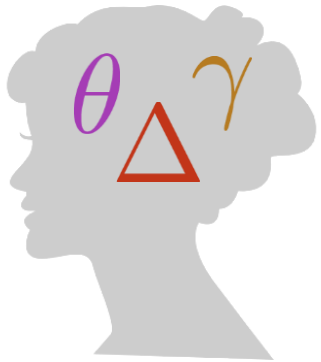
A bit of caution with features



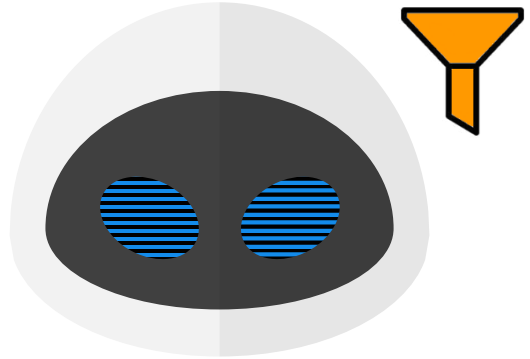


Tricks, mnemonics lessen cognitive load, increase speed

A bit of caution with features

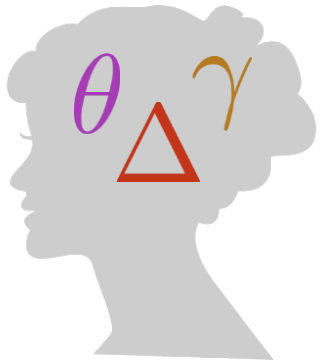


Tricks, mnemonics lessen cognitive load, increase speed



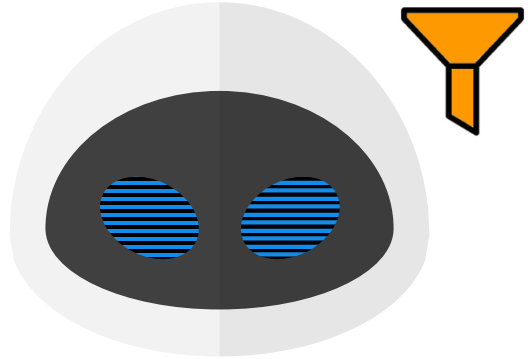
Derived features make learning easier, faster at test

A bit of caution with features



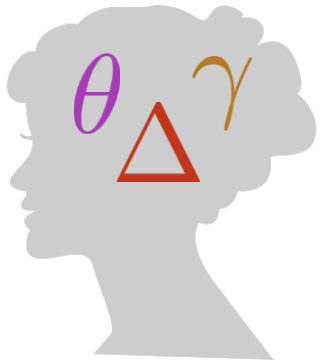
Tricks, mnemonics lessen cognitive load, increase speed

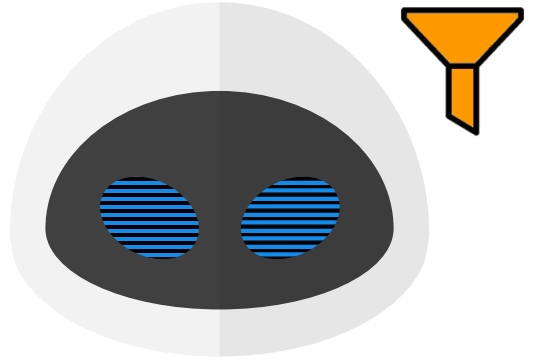
Easy questions can be solved in one step with a mnemonic!



Derived features make learning easier, faster at test

A bit of caution with features





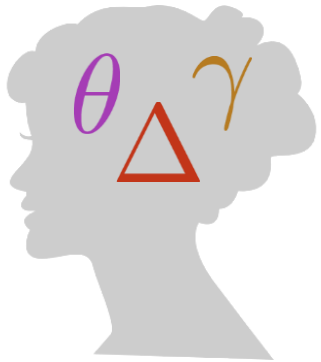
Tricks, mnemonics lessen cognitive load, increase speed

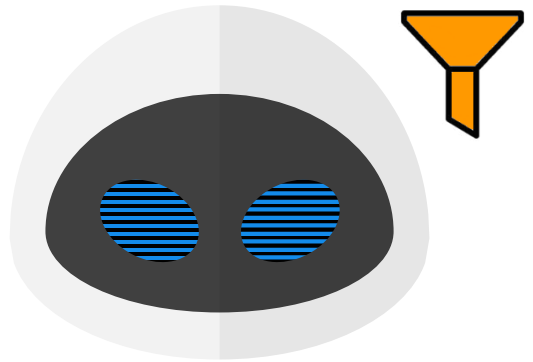
Easy questions can be solved in one step with a mnemonic!

Derived features make learning easier, faster at test

What you are trying to predict is just another feature of the data!

A bit of caution with features





Tricks, mnemonics lessen cognitive load, increase speed

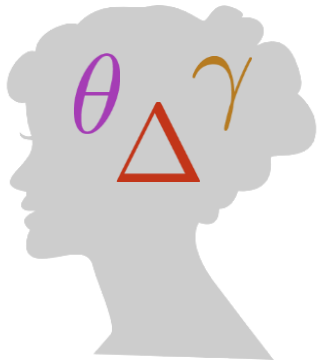
Easy questions can be solved in one step with a mnemonic!

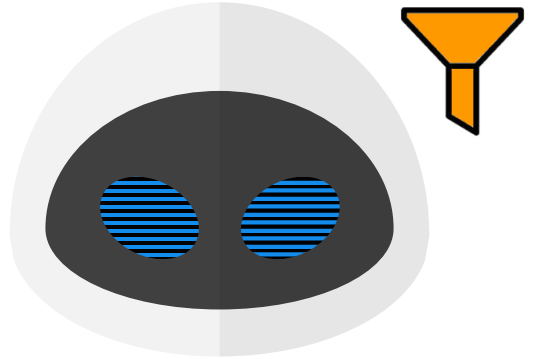
Too many mnemonics can confuse you at time of exam

Derived features make learning easier, faster at test

What you are trying to predict is just another feature of the data!

A bit of caution with features





Tricks, mnemonics lessen cognitive load, increase speed

Easy questions can be solved in one step with a mnemonic!

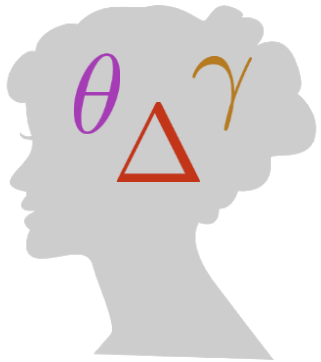
Too many mnemonics can confuse you at time of exam

Derived features make learning easier, faster at test

What you are trying to predict is just another feature of the data!

Too many useless features can confuse classifier

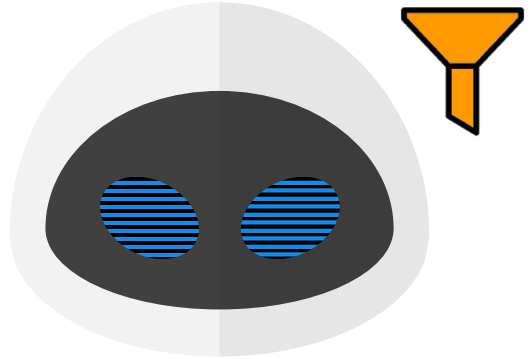
A bit of caution with features



Tricks, mnemonics lessen cognitive load, increase speed

Easy questions can be solved in one step with a mnemonic!

Too many mnemonics can confuse you at time of exam

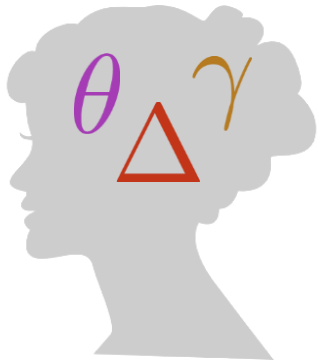


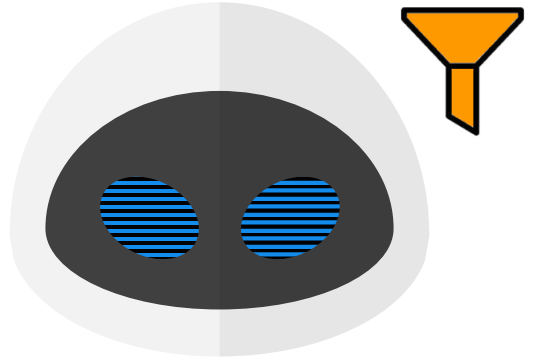
Derived features make learning easier, faster at test

What you are trying to predict is just another feature of the data!

Too many useless features can confuse classifier

A bit of caution with features





Tricks, mnemonics lessen cognitive load, increase speed

Easy questions can be solved in one step with a mnemonic!

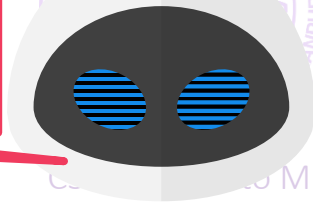
Too many mnemonics can confuse you at time of exam

Derived features make learning easier, faster at test

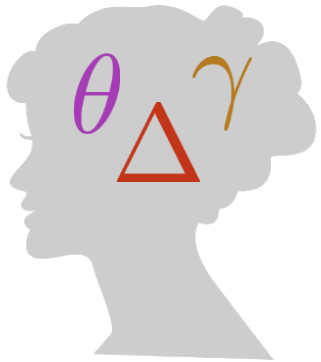
What you are trying to predict is just another feature of the data!

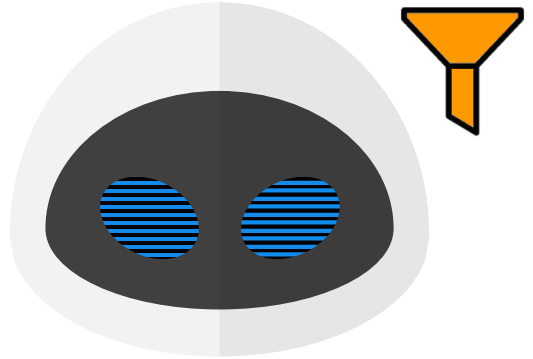
Too many useless features can confuse classifier

In fact, one of the main challenges in deep learning is that it learns way too many features



A bit of caution with features





Tricks, mnemonics lessen cognitive load, increase speed

Easy questions can be solved in one step with a mnemonic!

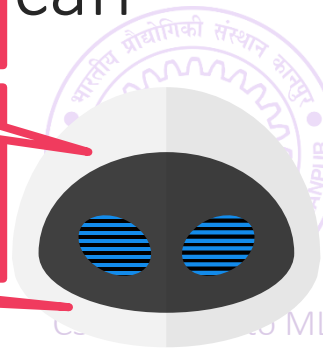
Too many mnemonics can confuse you at time of exam

Derived features make learning easier, faster at test

What you are trying to predict is just another feature of the data!

However, not to worry. For most of this course, we will give you pre-made feature vectors ☺

In fact, one of the main challenges in deep learning is that it learns way too many features



What are Vectors

Consider a d-dimensional real vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d], \mathbf{x}_i \in \mathbb{R}$

For a physicist, a vector is a way to encode a magnitude and direction

For a mathematician, a vector is an object in a vector space

For an ML person, a vector is simply a list of numbers, each number representing a useful piece of information about an object

Example: spam filter, a vector stored which words occurred in email

Example: image recognition, a vector stored the pixel RGB values

SEE REFERENCE [DFO] SECTION 1.1 FOR A NICE INTRODUCTION

What are Vectors

Consider a d-dimensional real vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d], \mathbf{x}_i \in \mathbb{R}$

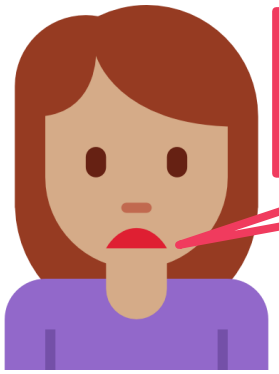
For a physicist, a vector is a way to encode a magnitude and direction

For a mathematician, a vector is an object in a vector space

For an ML person, a vector is simply a list of numbers, each number representing a useful piece of information about an object

Example: spam filter, a vector stored which words occurred in email

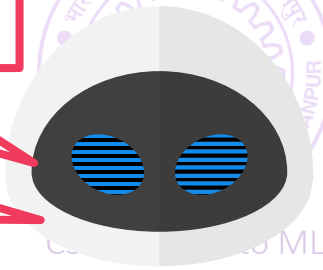
Example: image recognition, a vector stored the pixel RGB values



Does this mean I will have to learn math to do ML?

A bit of math will be required but 1) it will be simple and 2) it will be totally worth it!

True, for me, vectors are just like arrays of numbers. However, sometimes I do indeed do math with them



SEE REFERENCE [DFO] SECTION 1.1 FOR A NICE INTRO

Operations on Vectors

Given two d-dimensional vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we can do the following

Sum of two vectors (is also a vector of same dimension)

$$a = \mathbf{x} + \mathbf{y} \in \mathbb{R}^d$$

Difference of two vectors (is also a vector of same dimension)

$$b = \mathbf{x} - \mathbf{y} \in \mathbb{R}^d$$

Dot product of two vectors (is a real number, possibly negative)

$$p = \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{j=1}^{d} \mathbf{x}_j \mathbf{y}_j$$

Euclidean length of a vector (is a positive real number, possibly zero)

$$q = \|\mathbf{x}\|_2 = \sqrt{\sum_{j=1}^{d} \mathbf{x}_i^2} = \sqrt{\mathbf{x}^\top \mathbf{x}}$$

Operations on Vectors

Given two d-dimensional vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we can do the following

Sum of two vectors (is also a vector of same dimension)

$$a = \mathbf{x} + \mathbf{y} \in \mathbb{R}^d$$

Difference of two vectors (is also a vector of same dimension)

$$b = \mathbf{x} - \mathbf{y} \in \mathbb{R}^d$$

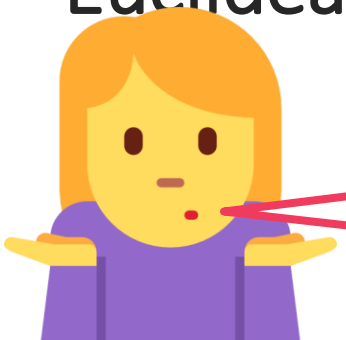
Dot product of two vectors (is a real number, possibly negative)

$$p = \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{d} \mathbf{x}_i \mathbf{y}_i$$

Euclidean length of

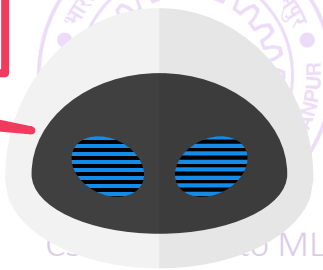
zero)

$$q = \|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$$



So I can take two feature vectors and add them. But what sense does it make to add two emails?

Good point – just wait a second for some intuition. However, to be honest, sometimes it is good in ML not to hunt too much for intuition and just let the math be!



ML Operations on Feature Vectors

Given n feature vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$

Average feature vector

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}^i$$

Euclidean distance between two feature vectors

$$d_2(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|_2 = \sqrt{\sum_{j=1}^{d} \left(\mathbf{x}_j^1 - \mathbf{x}_j^2\right)^2}$$

Distance of a feature vector from the average vector

$$d_2(\mathbf{x}^1, \boldsymbol{\mu}) = \|\mathbf{x}^1 - \boldsymbol{\mu}\|_2$$

ML Operations on Feature Vectors

Given n feature vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n$

fea

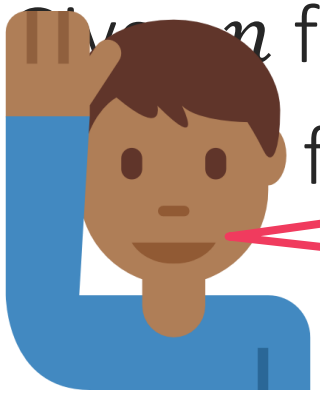
$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}^i$$

Euclidean distance between two fe

$$d_2(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|_2 = \sqrt{\sum_{j=1}(\mathbf{x}_j^1 - \mathbf{x}_j^2)}$$

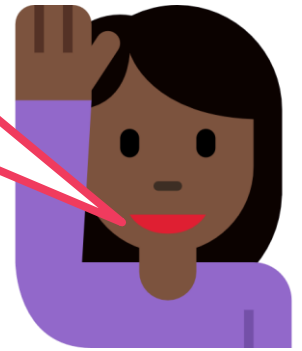
Distance of a feature vector from the average vector

$$d_2(\mathbf{x}^1, \boldsymbol{\mu}) = \|\mathbf{x}^1$$

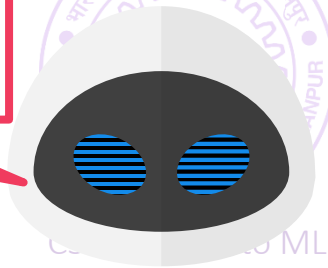


I can use this to find out if two emails are similar or very different from each other

I can use this to find out what an "average" email looks like ☺. So averaging vectors makes sense!



I can use this to find out if a new email is similar to an average spam email or an average regular email



Excellent! We are now ready for our first ML algorithm!!

Learning with Prototypes

- The basic mantra here is

> *If a new email looks similar to an average spam email, it maybe spam. On the other hand, if a new email looks similar to an average normal email, it should be normal*

- In ML, the word *prototype* is used to refer to something that is representative or captures the qualities of a class of objects?
- How can we do ML using prototypes?

Bird or not!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

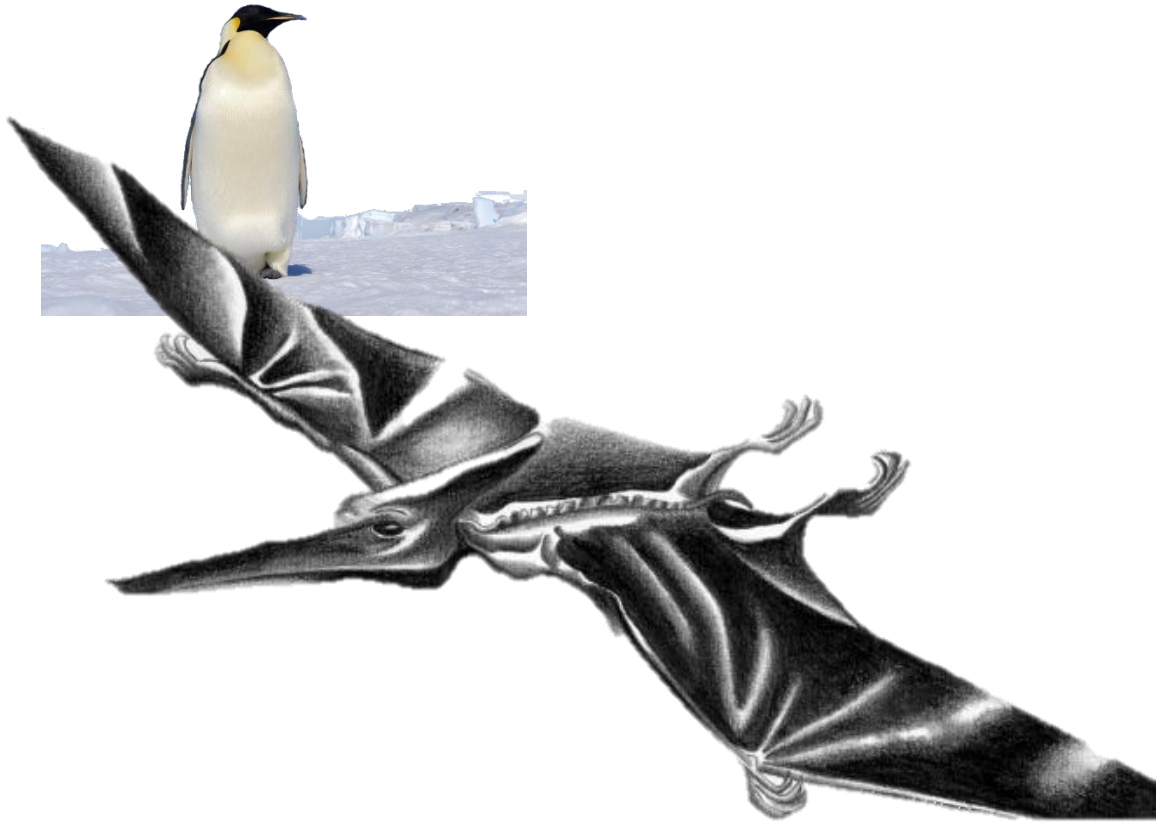
Bird or not!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

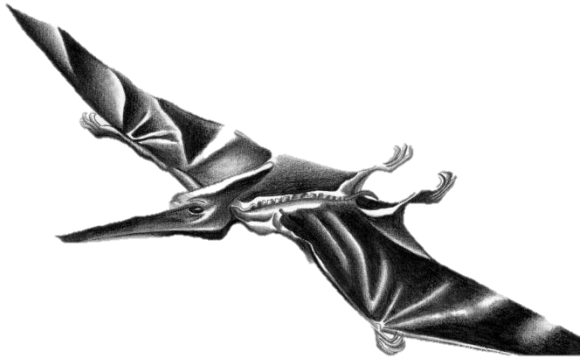
PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!



PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

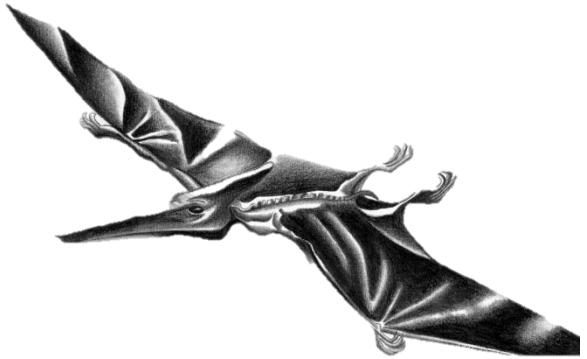


PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

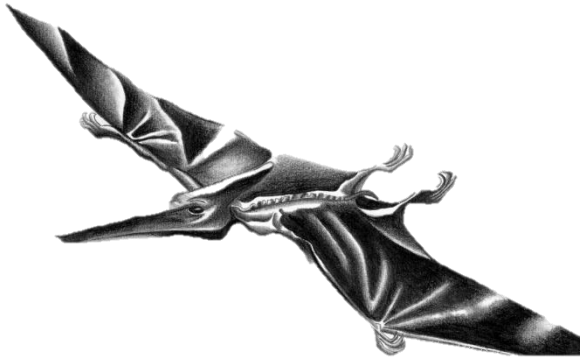


PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

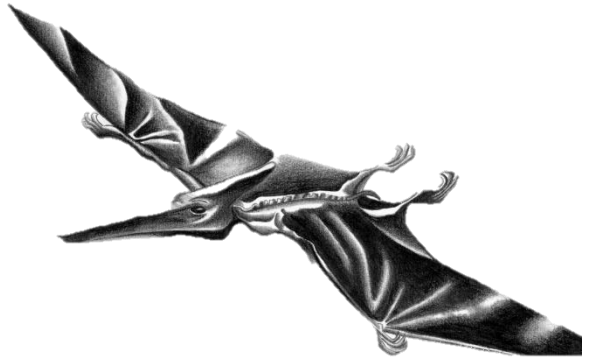


PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

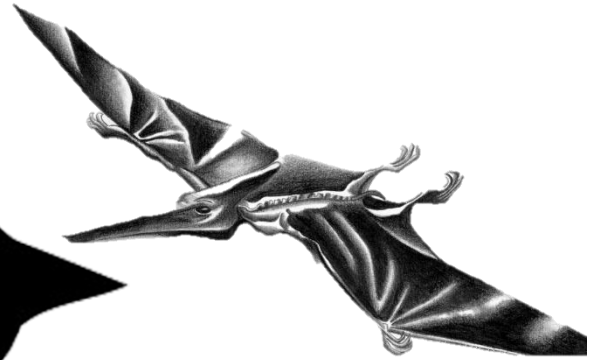
PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!



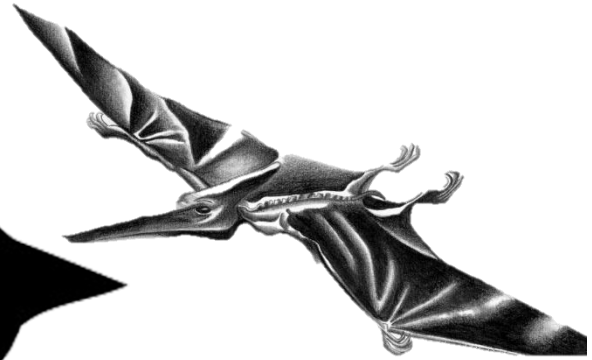
PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!



PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

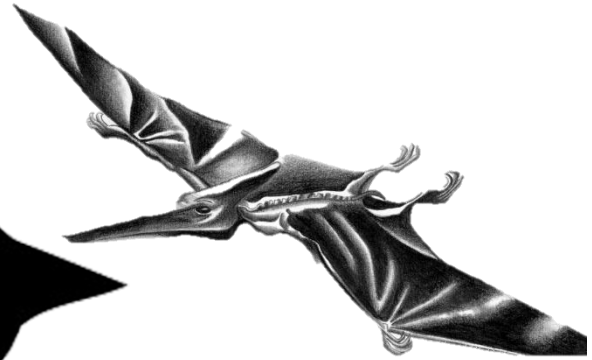
Bird or not!



A *prototypical* bird!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

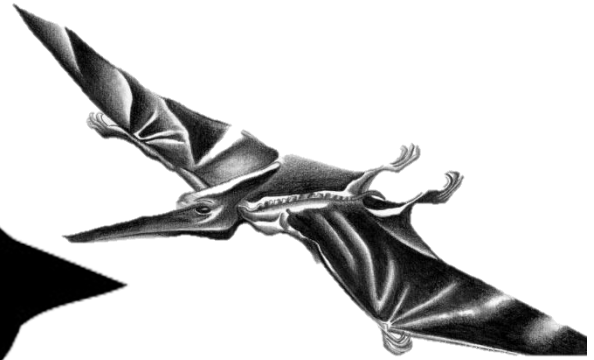
Bird or not!



A *prototypical* bird!

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

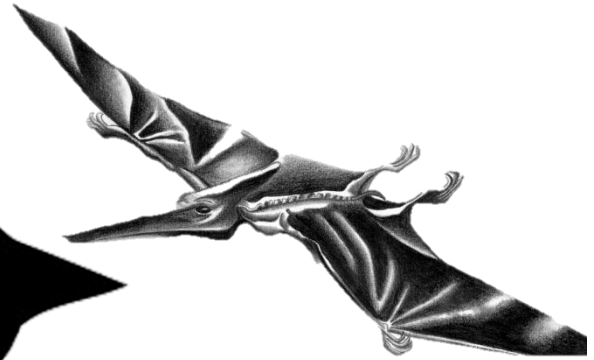


A *prototypical* bird!

What abstract qualities do we associate with “birdness”

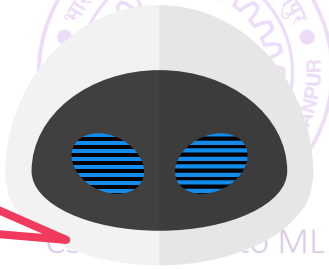
PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!

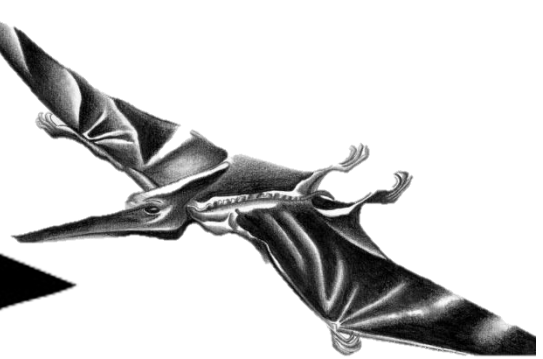


PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

What abstract qualities do we associate with “birdness”



Bird or not!



Has wings

Can fly

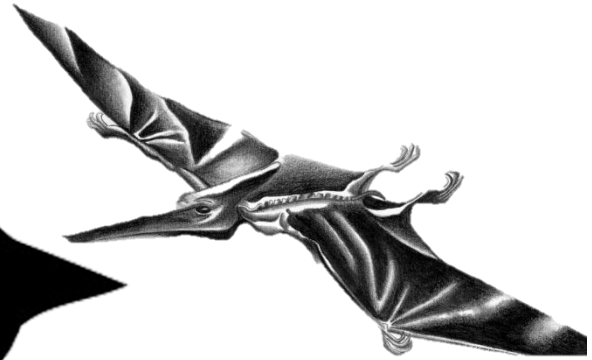
Can sing

Has a beak

What abstract qualities do we associate with “birdness”

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!



Has wings

Can fly

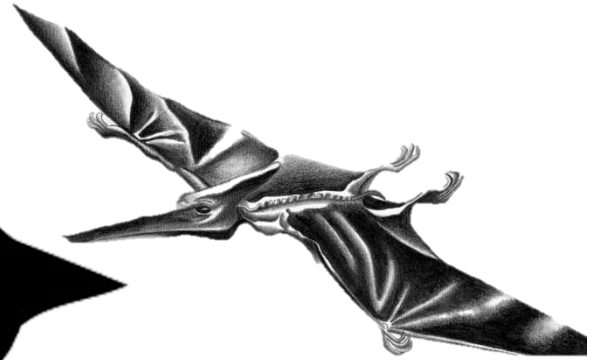
Can sing

Has a beak

What abstract qualities do we associate with “birdness”

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!



Has wings

Can fly

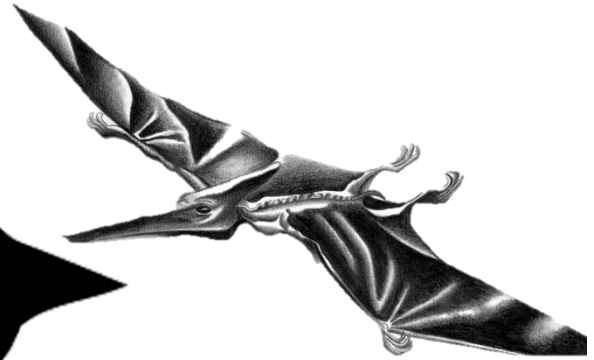
Can sing

Has a beak

What abstract qualities do we associate with “birdness”

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!



Important for deciding bird or not

Has wings

Can fly

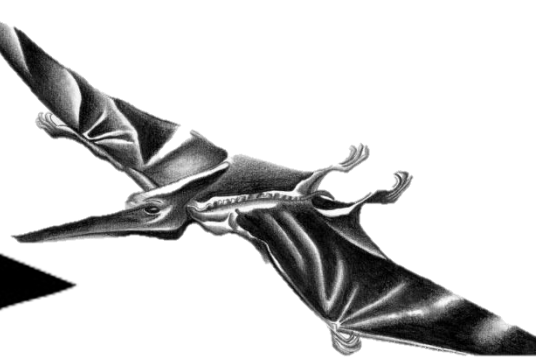
Can sing

Has a beak

What abstract qualities do we associate with “birdness”

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!



Important for deciding bird or not

Has wings

Can fly

Can sing

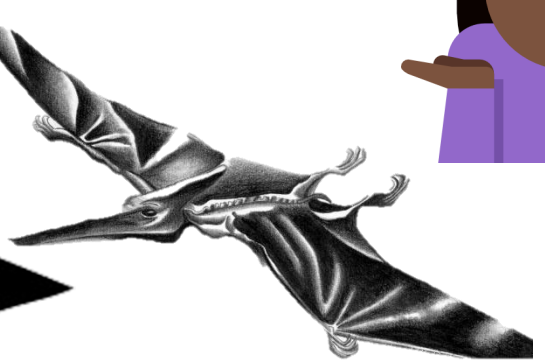
Not that important for deciding bird or not

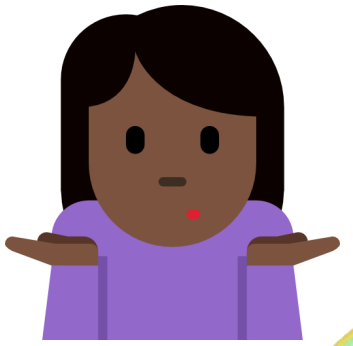
Has a beak

What abstract qualities do we associate with “birdness”

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not!





Important for deciding bird or not

Has wings

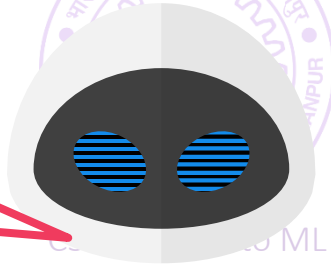
Can fly

Can sing

Not that important for deciding bird or not

Has a beak

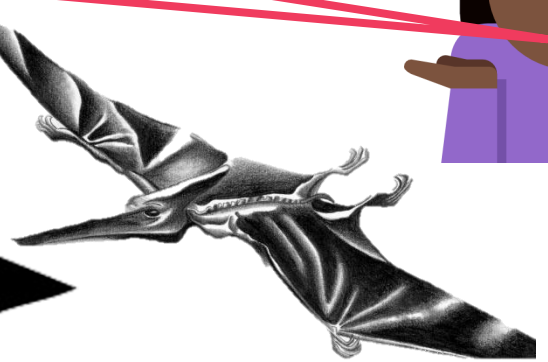
What abstract qualities do we associate with “birdness”

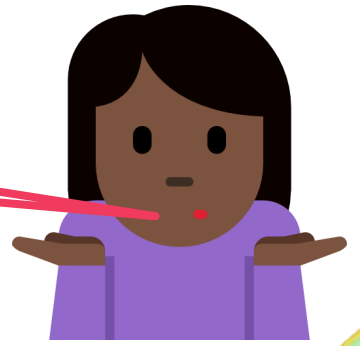


PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not

But the Pterosaur will still get classified as a bird – it has wings and a beak





Important for deciding bird or not

Has wings

Can fly

Can sing

Not that important for deciding bird or not

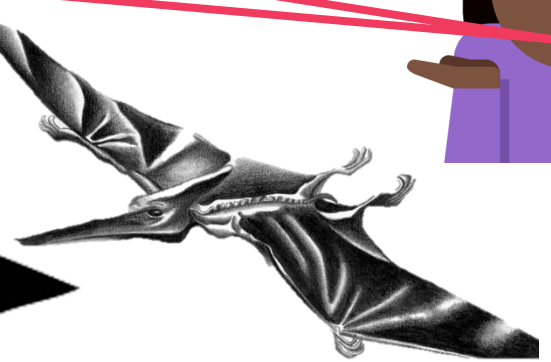
Has a beak

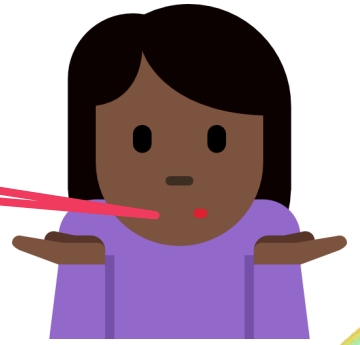
What abstract qualities do we associate with “birdness”

PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

Bird or not

But the Pterosaur will still get classified as a bird – it has wings and a beak





Important for deciding bird or not

Has wings

Can fly

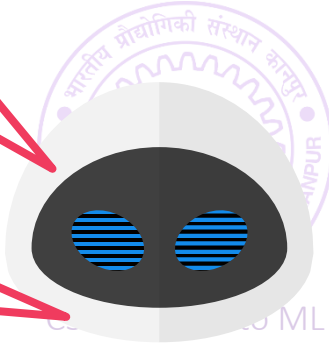
Can sing

Not that important for deciding bird or not

H

This means that our prototype is not good enough and we may need better features

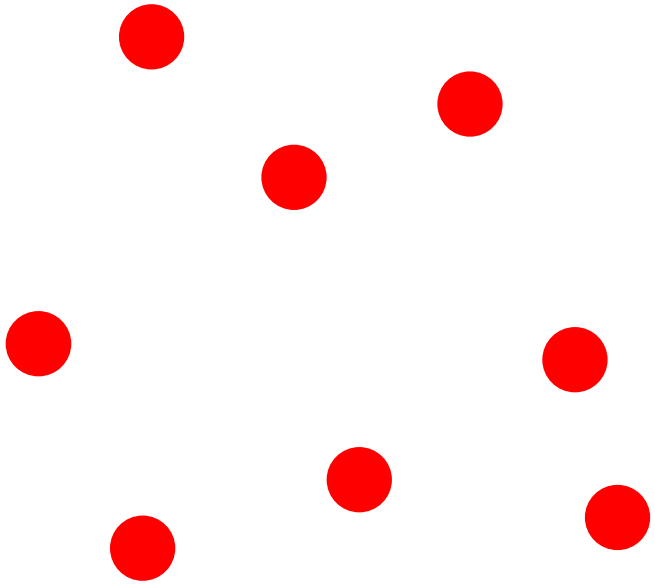
What abstract qualities do we associate with “birdness”

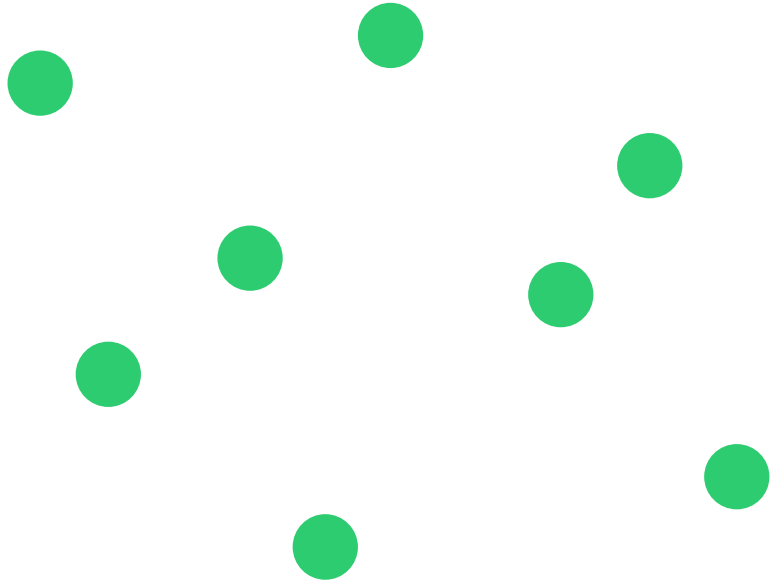


PIXABAY.COM, INDEPENDENT.CO.UK, YOUTUBE.COM, NEWDINOSAURS.COM, WIKIPEDIA.COM, CLIPARTQUEEN.COM

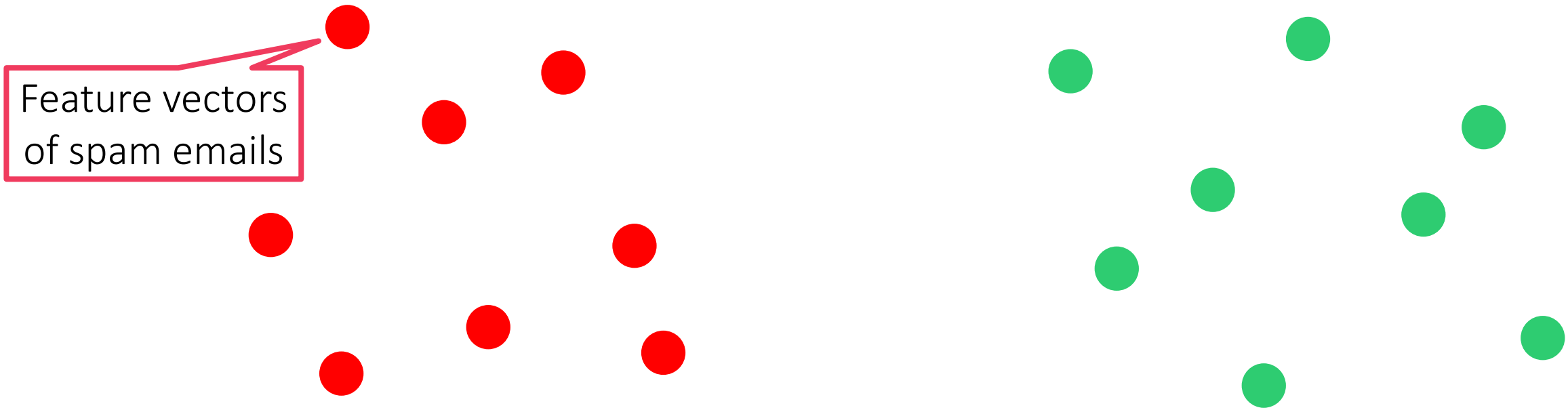
Learning with Prototypes

Learning with Prototypes

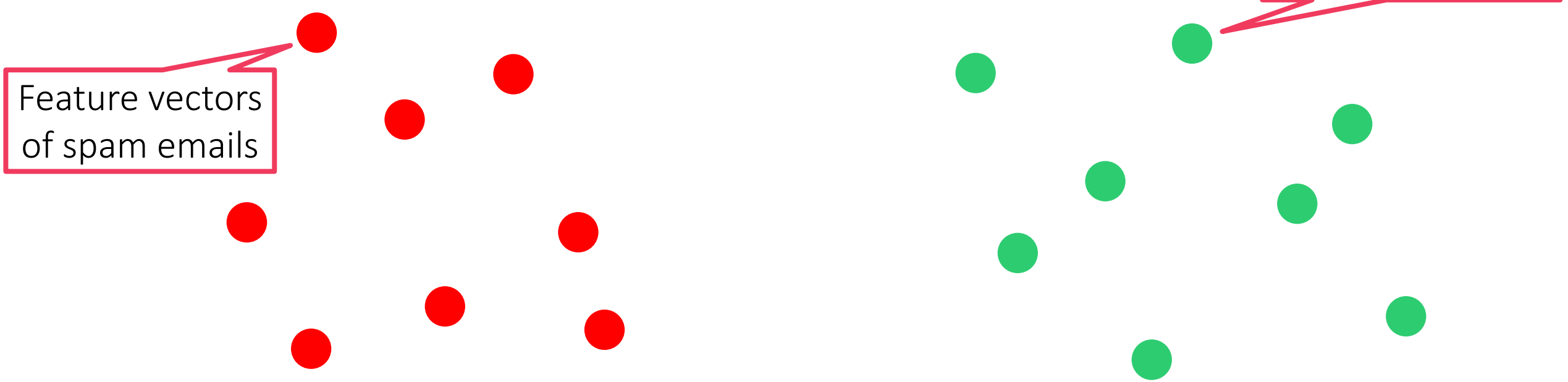




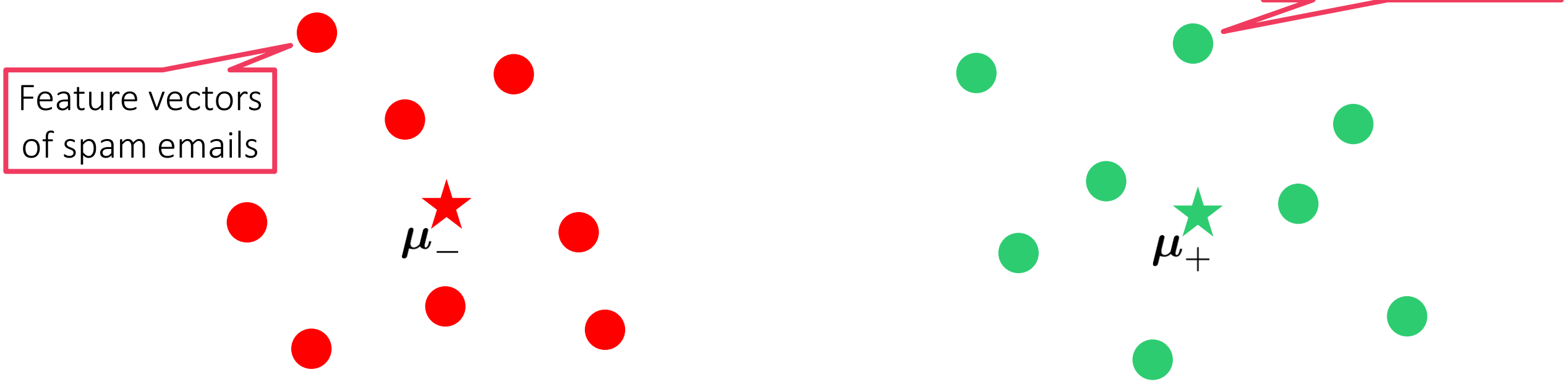
Learning with Prototypes



Learning with Prototypes

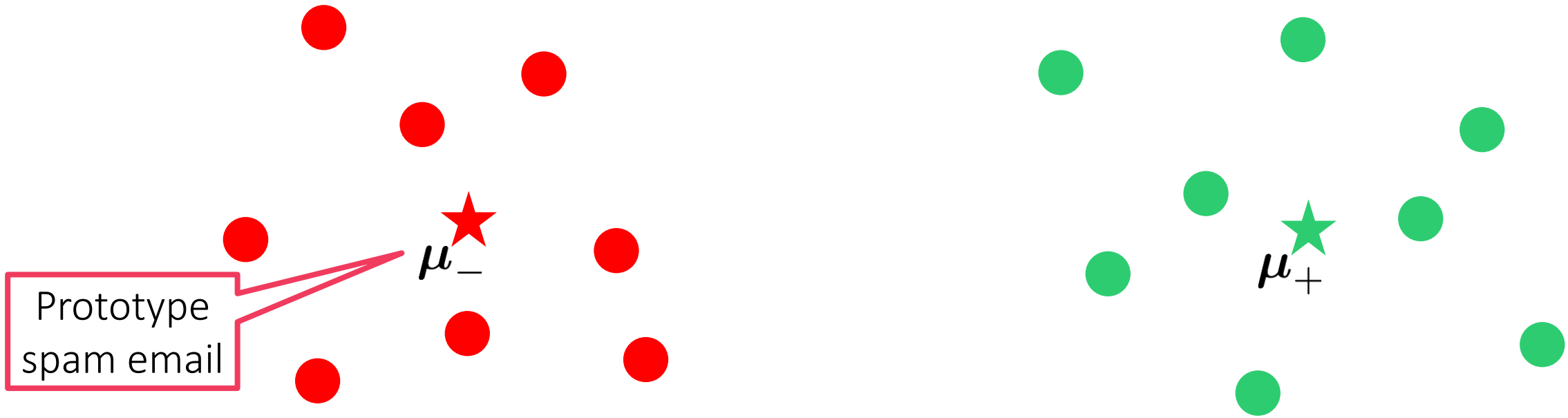


Learning with Prototypes



Learning with Prototypes

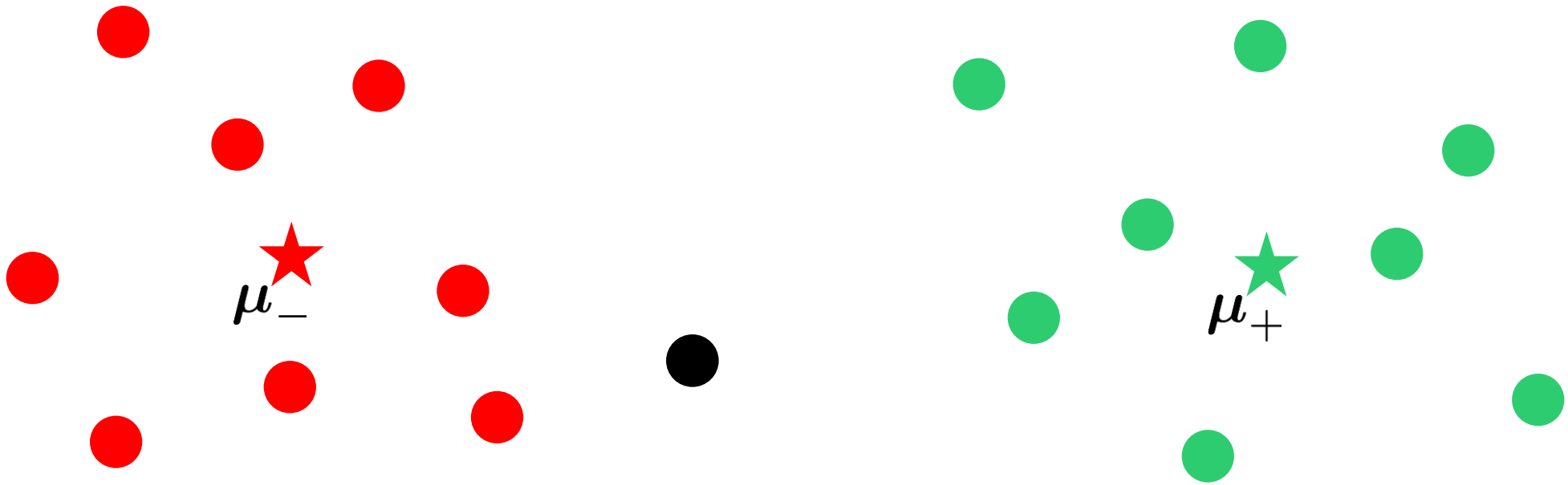
Learning with Prototypes



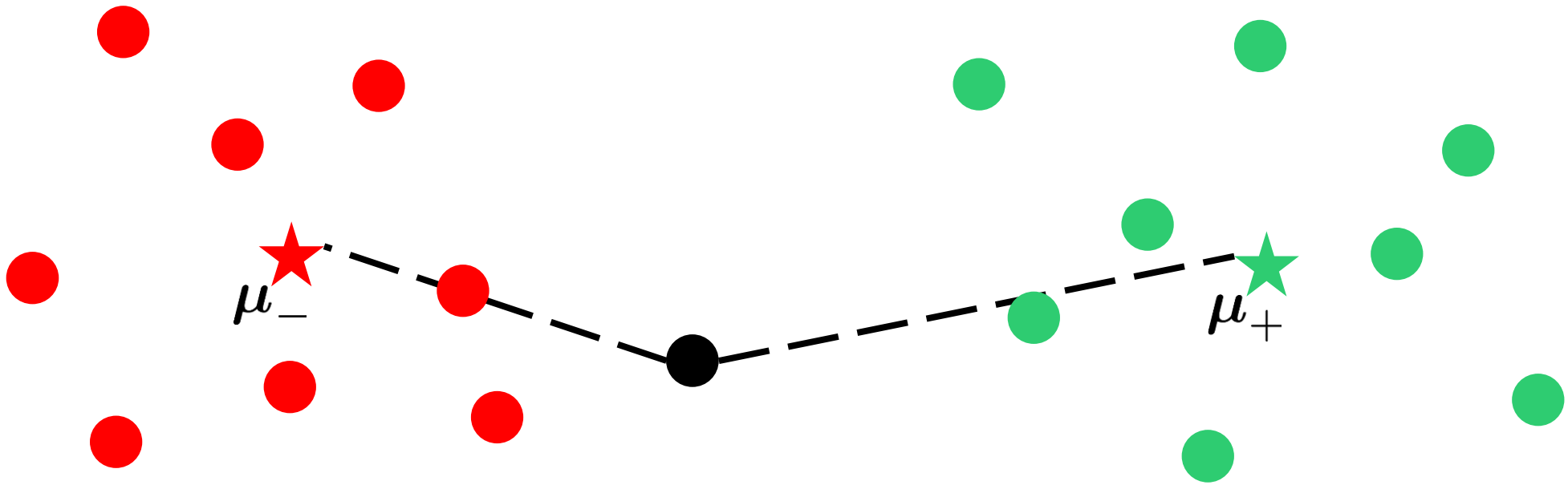
Learning with Prototypes

Learning with Prototypes

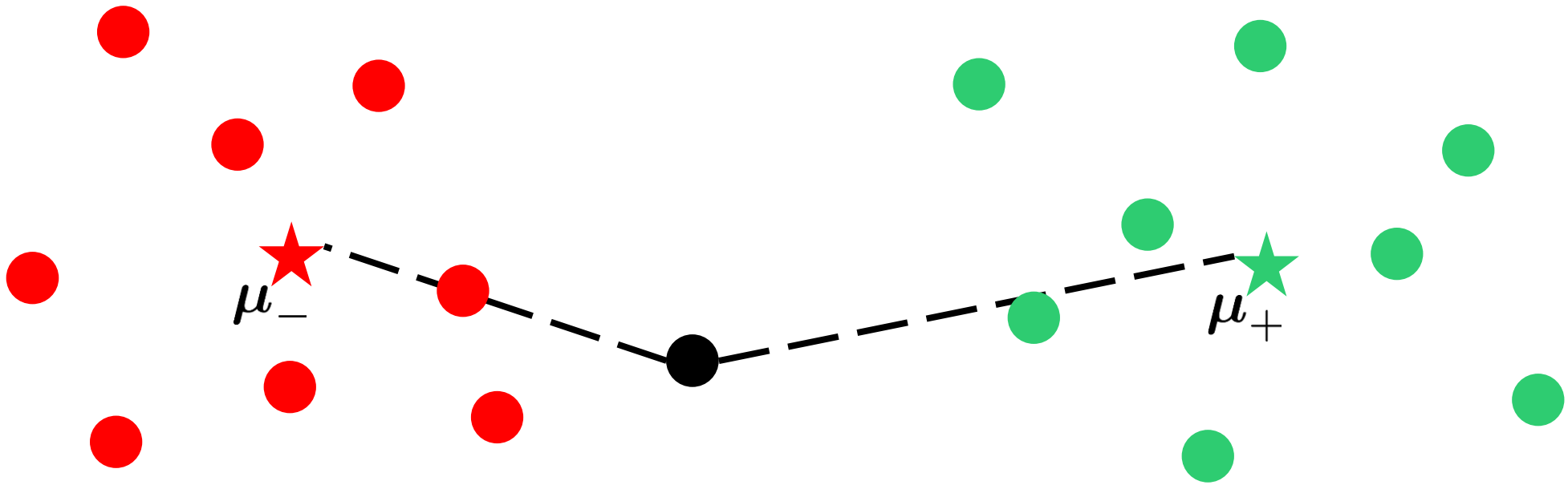
Learning with Prototypes



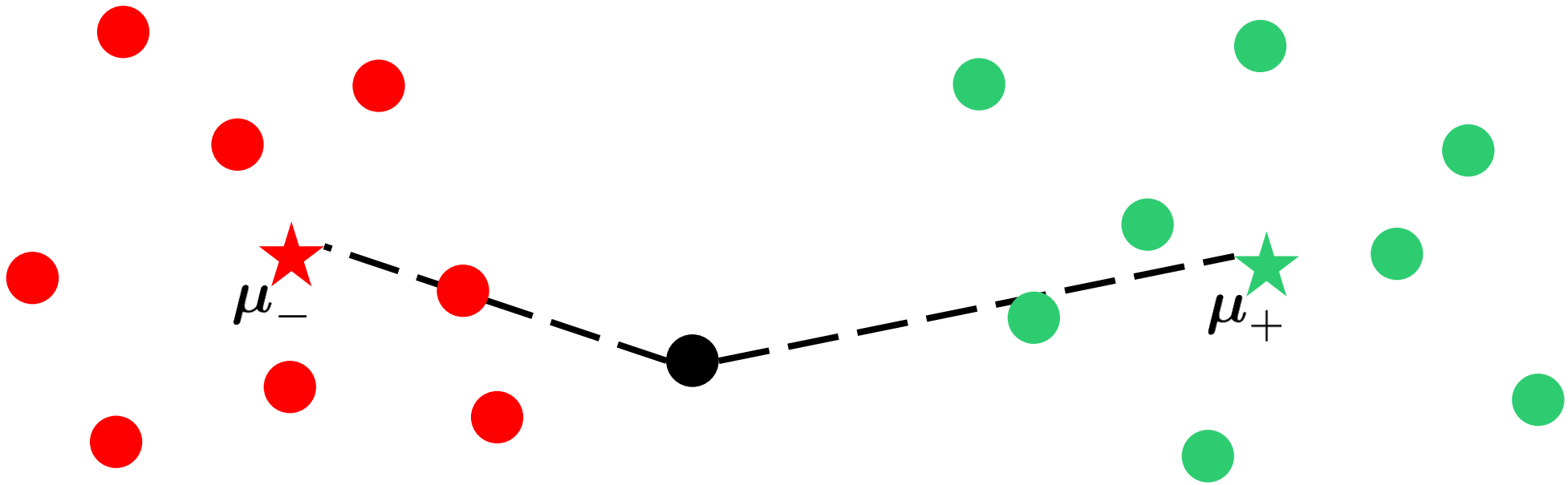
Learning with Prototypes



Learning with Prototypes

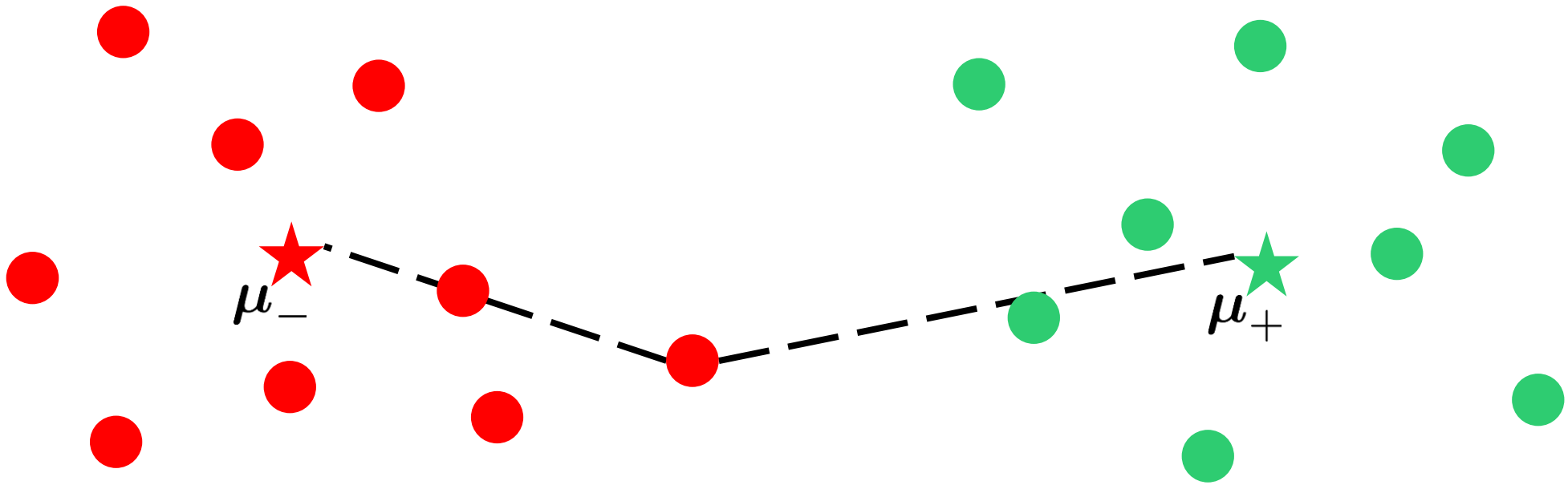


Learning with Prototypes



This email is closer to the prototype spam email than the prototype normal email so I will classify this new email as spam!

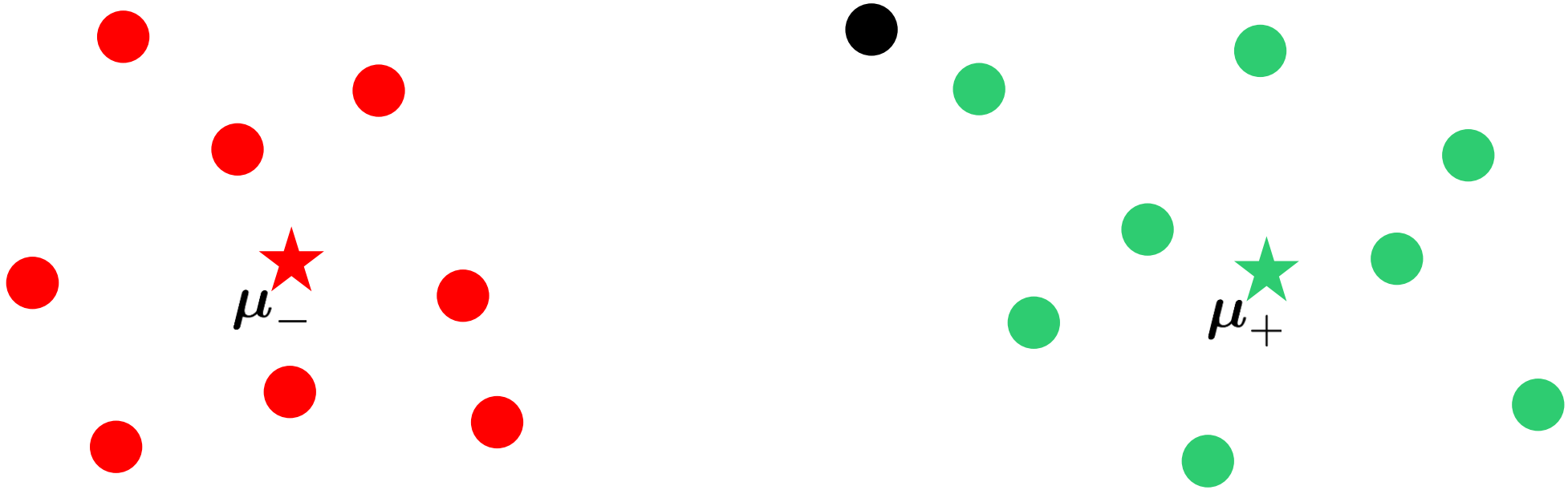
Learning with Prototypes



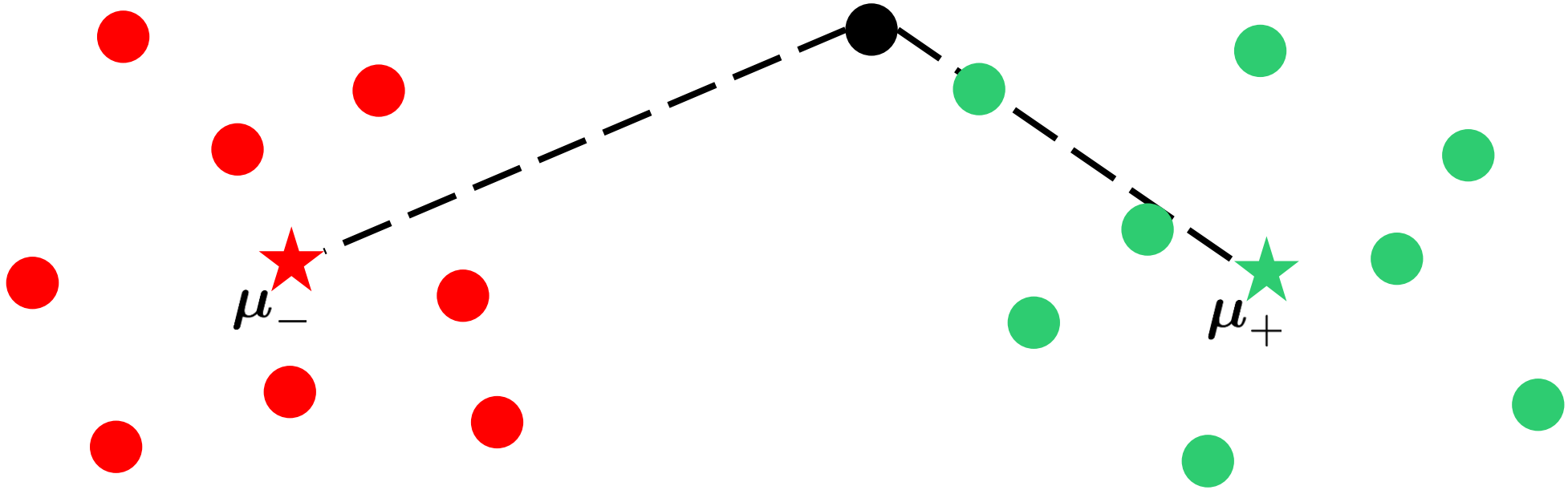
This email is closer to the prototype spam email than the prototype normal email so I will classify this new email as spam!

Learning with Prototypes

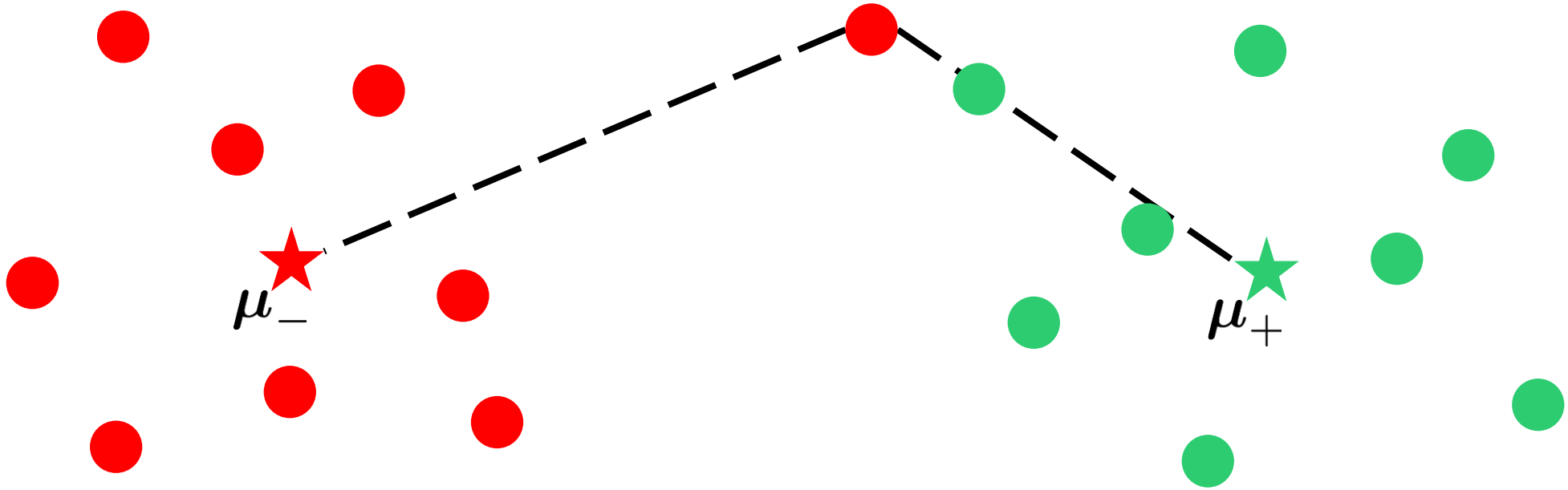
Learning with Prototypes



Learning with Prototypes

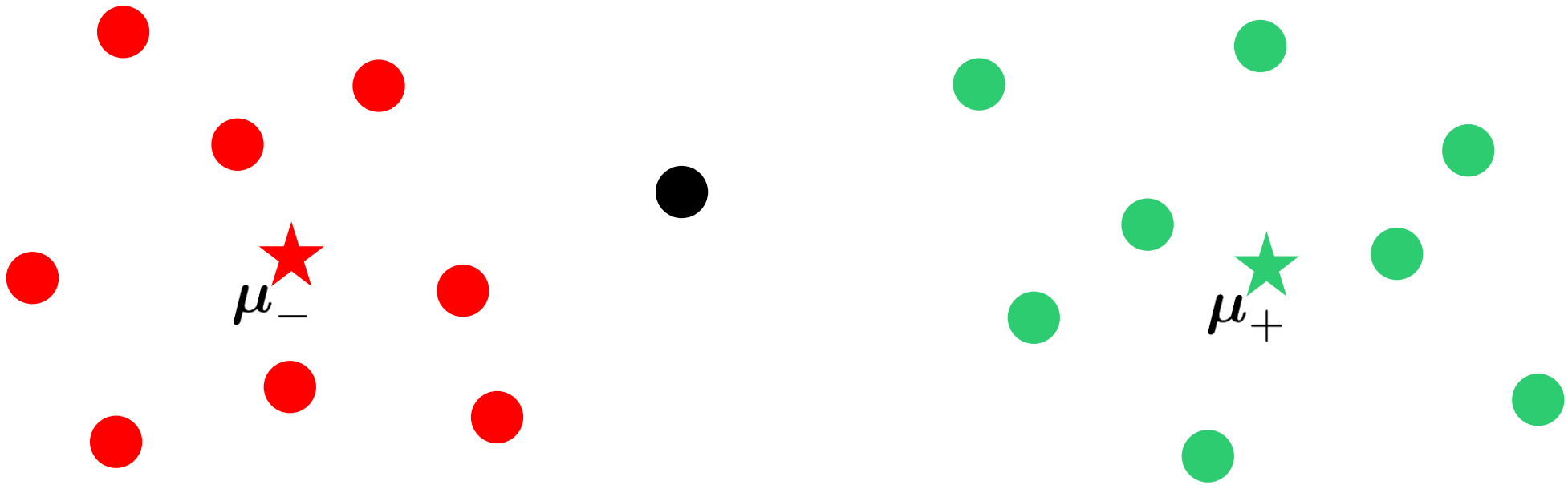


Learning with Prototypes

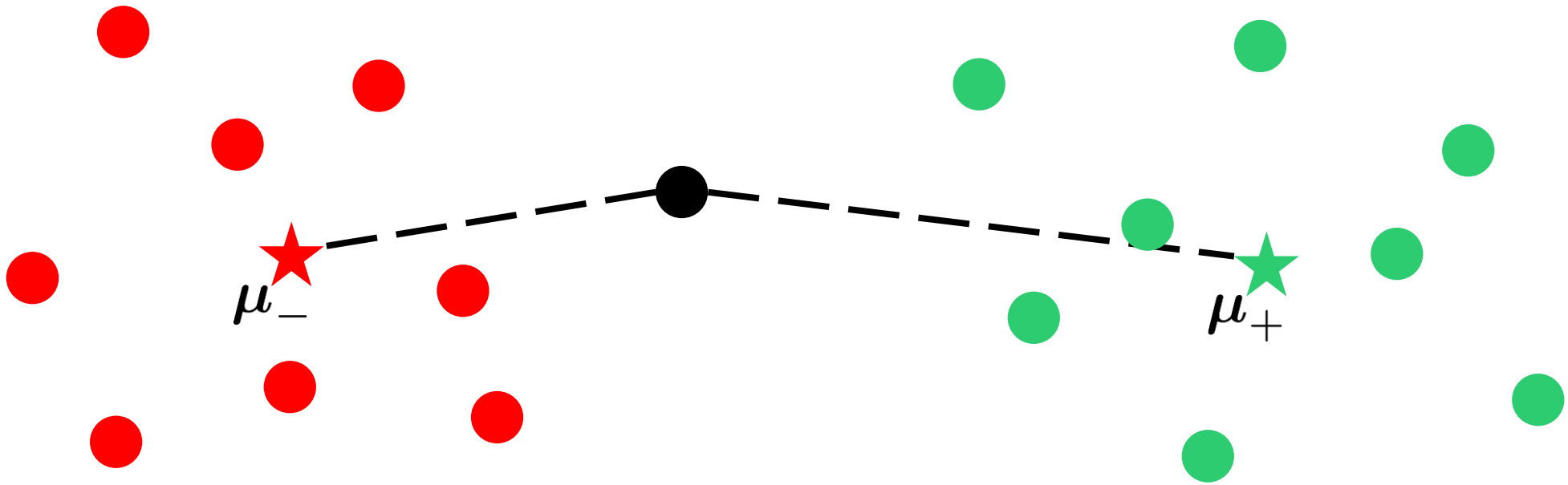


Learning with Prototypes

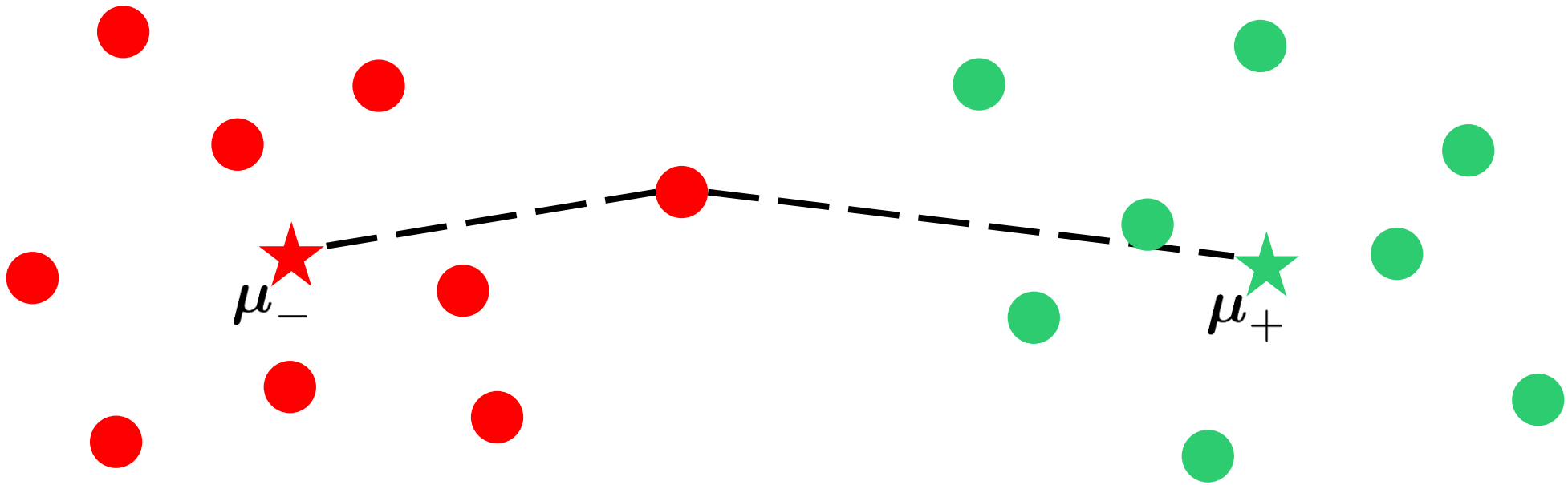
Learning with Prototypes



Learning with Prototypes



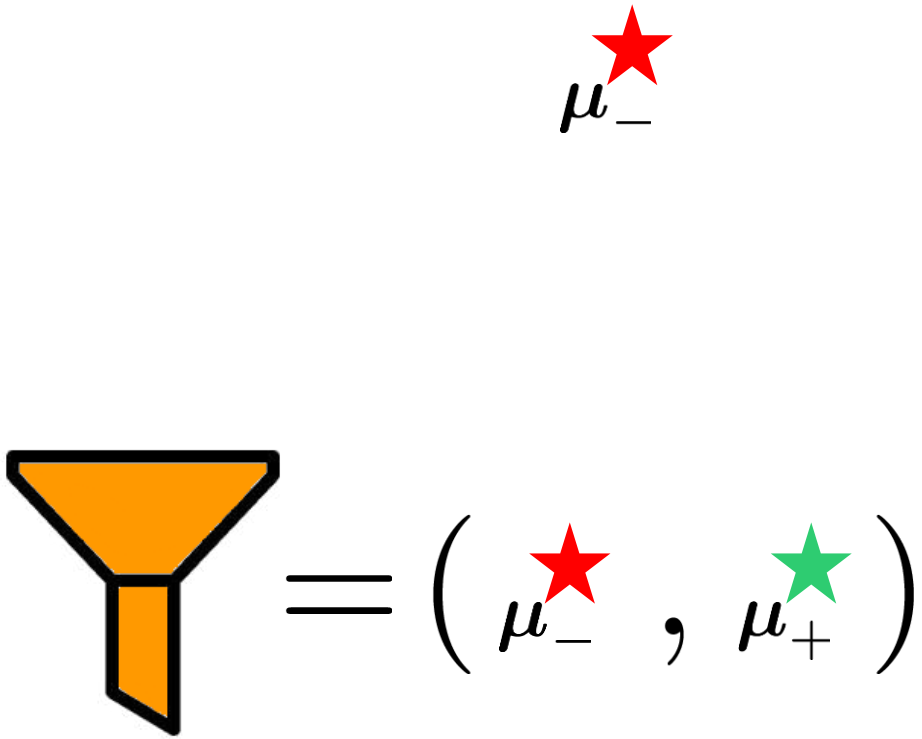
Learning with Prototypes



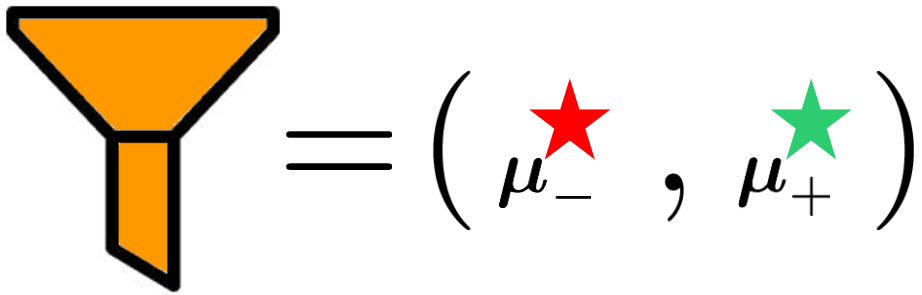
Learning with Prototypes

Learning with Prototypes

Learning with Prototypes

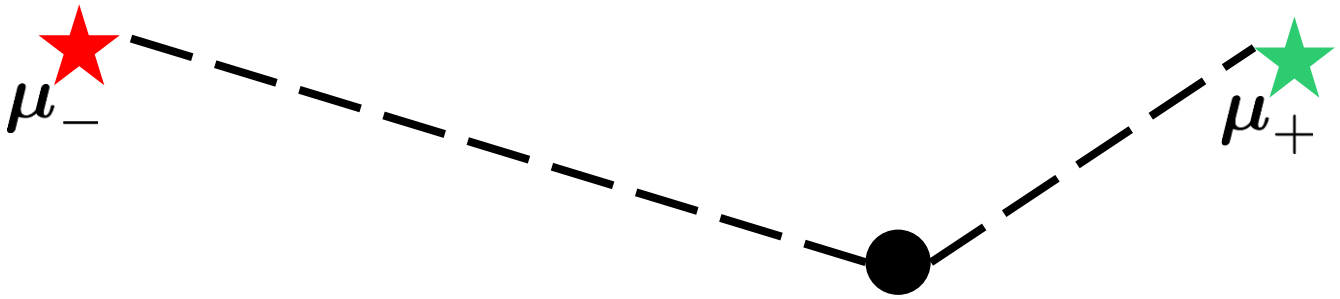


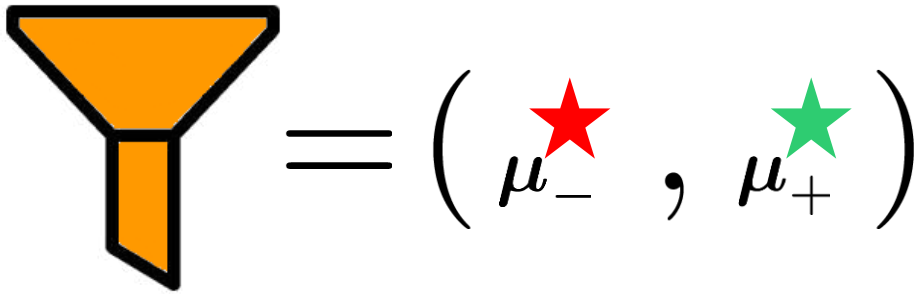
Learning with Prototypes



In order to classify a new email as spam/non-spam I only need the two prototypes, not the training emails

Learning with Prototypes

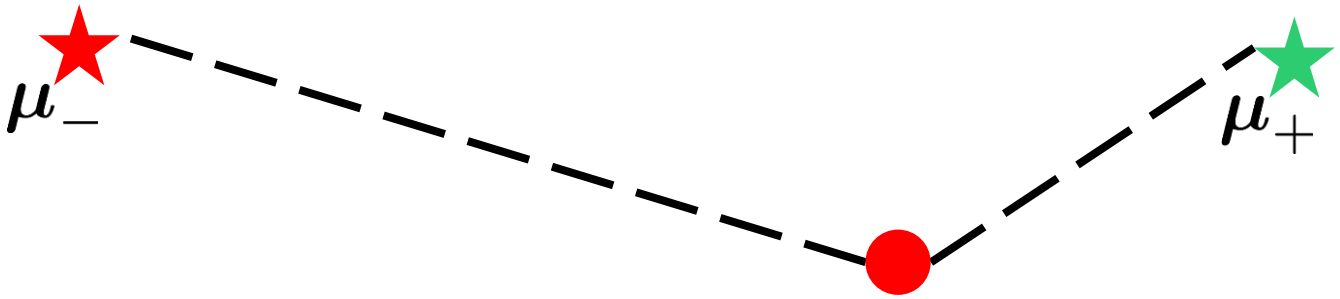


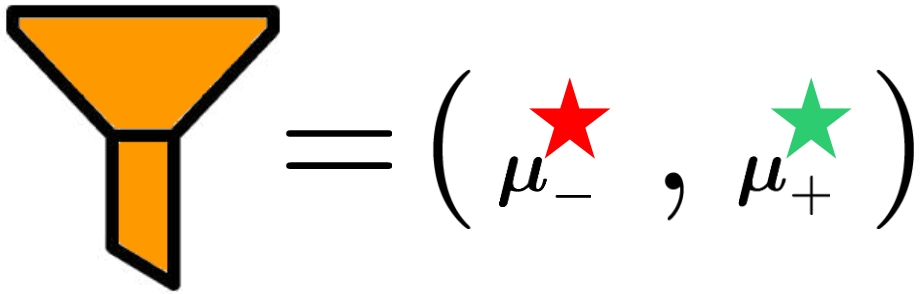


$$= \left(\mu_- \ , \ \mu_+ \right)$$

In order to classify a new email as spam/non-spam I only need the two prototypes, not the training emails

Learning with Prototypes

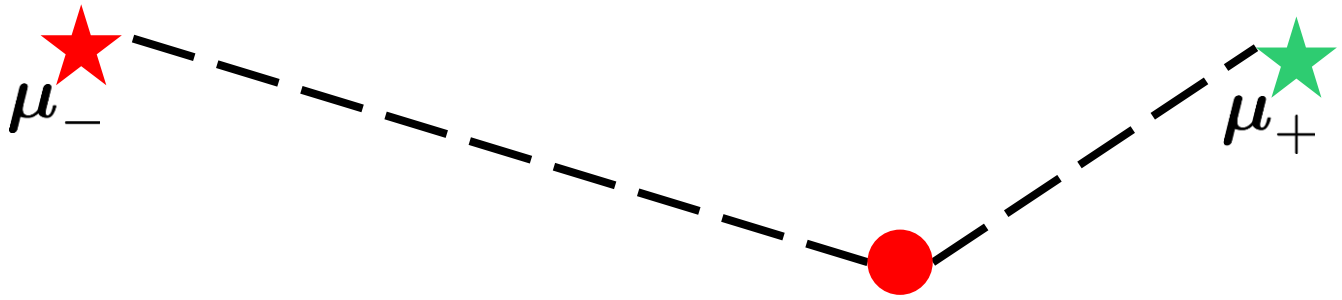


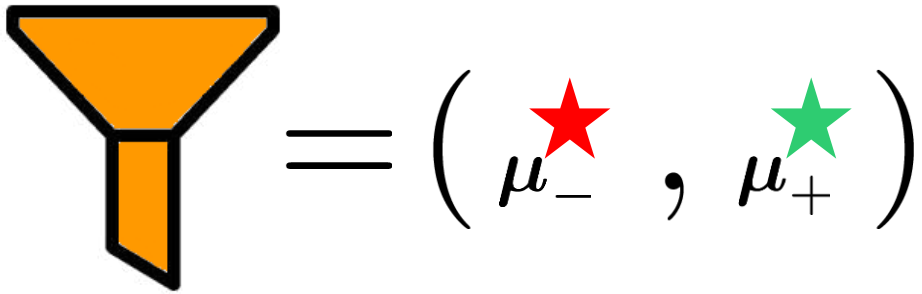


$$= \left(\mu_- \ , \ \mu_+ \right)$$

In order to classify a new email as spam/non-spam I only need the two prototypes, not the training emails

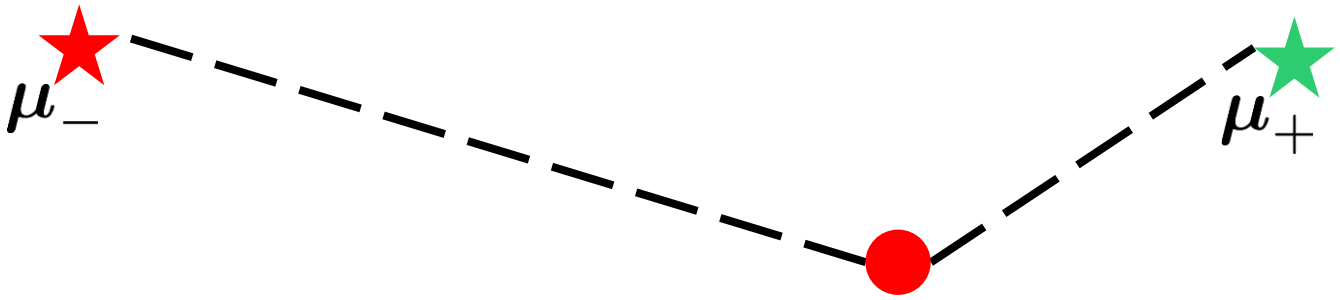
Learning with Prototypes





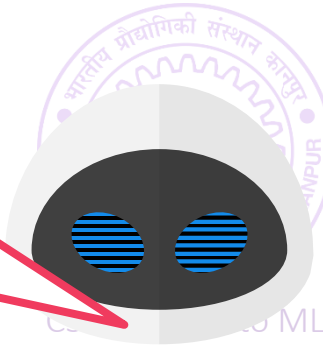
$$= \left(\mu_- \ , \ \mu_+ \right)$$

Learning with Prototypes

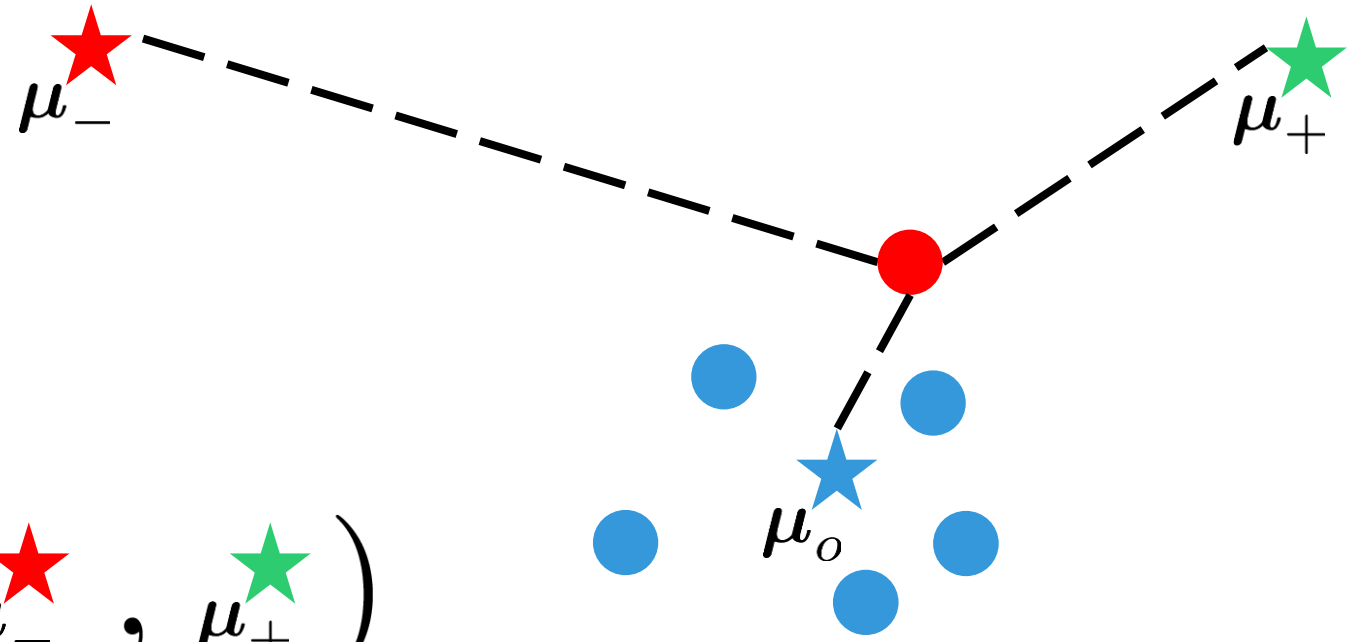


$$= \left(\mu_-^{\star} , \mu_+^{\star} \right)$$

In fact I can even handle cases where there are more than two classes e.g. instead of spam/non-spam, suppose the classes were Primary, Social, Updates …

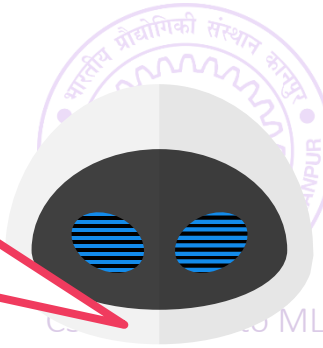


Learning with Prototypes

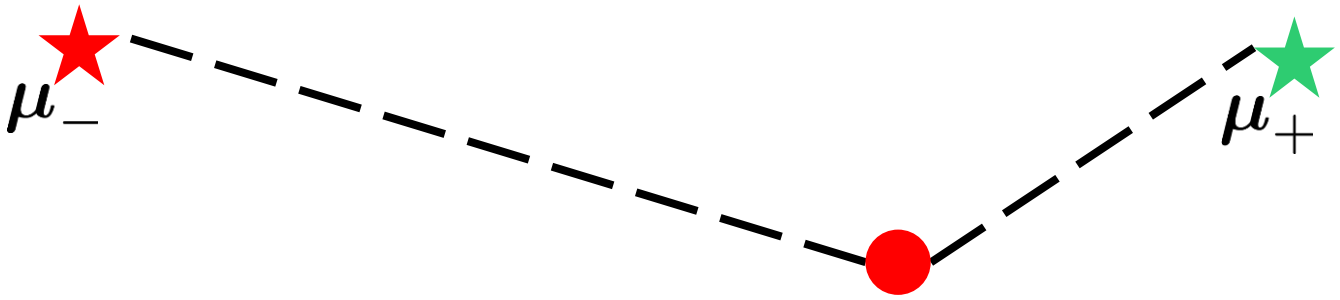


$= (\mu_-, \mu_+)$

In fact I can even handle cases where there are more than two classes e.g. instead of spam/non-spam, suppose the classes were Primary, Social, Updates …

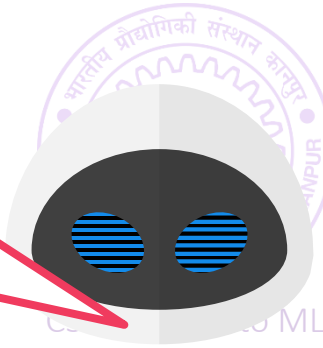


Learning with Prototypes

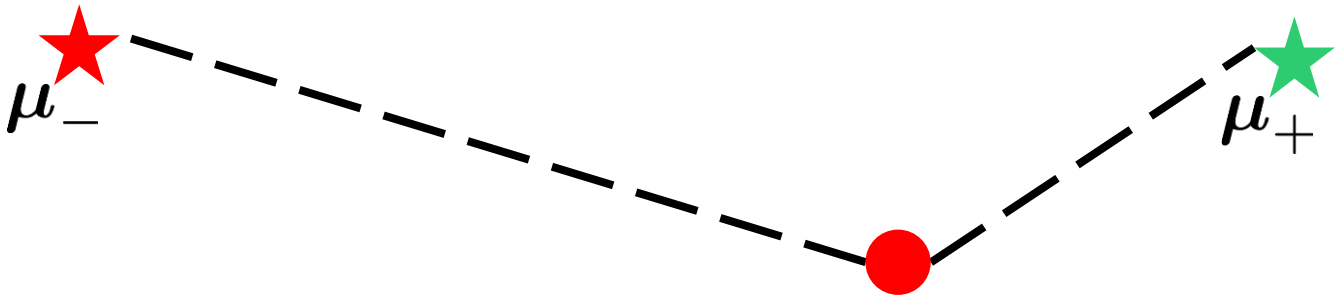


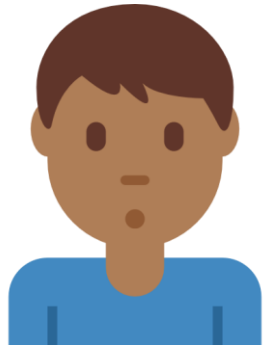
$$= \left(\mu_-^{\star} , \mu_+^{\star} \right)$$

In fact I can even handle cases where there are more than two classes e.g. instead of spam/non-spam, suppose the classes were Primary, Social, Updates …



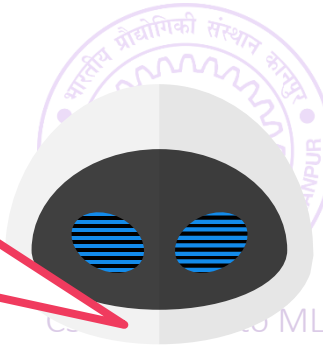
Learning with Prototypes



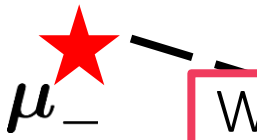


$$= \left(\mu_-^{\star} , \mu_+^{\star} \right)$$

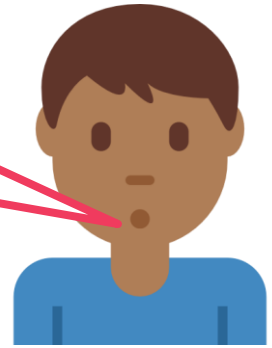
In fact I can even handle cases where there are more than two classes e.g. instead of spam/non-spam, suppose the classes were Primary, Social, Updates …



Learning with Prototypes

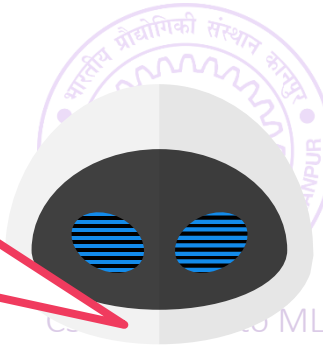


Wow! So so this is your model – just two vectors. You were able to distil all the information in the training emails into just two prototype vectors!!



$$= \left(\mu_-^{\star} \;,\; \mu_+^{\star} \right)$$

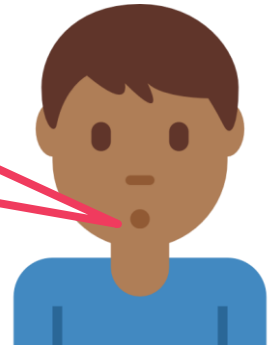
In fact I can even handle cases where there are more than two classes e.g. instead of spam/non-spam, suppose the classes were Primary, Social, Updates …



Learning with Prototypes

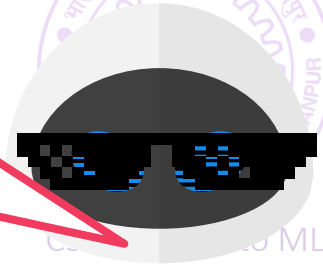
μ_-

Wow! So so this is your model – just two vectors. You were able to distil all the information in the training emails into just two prototype vectors!!



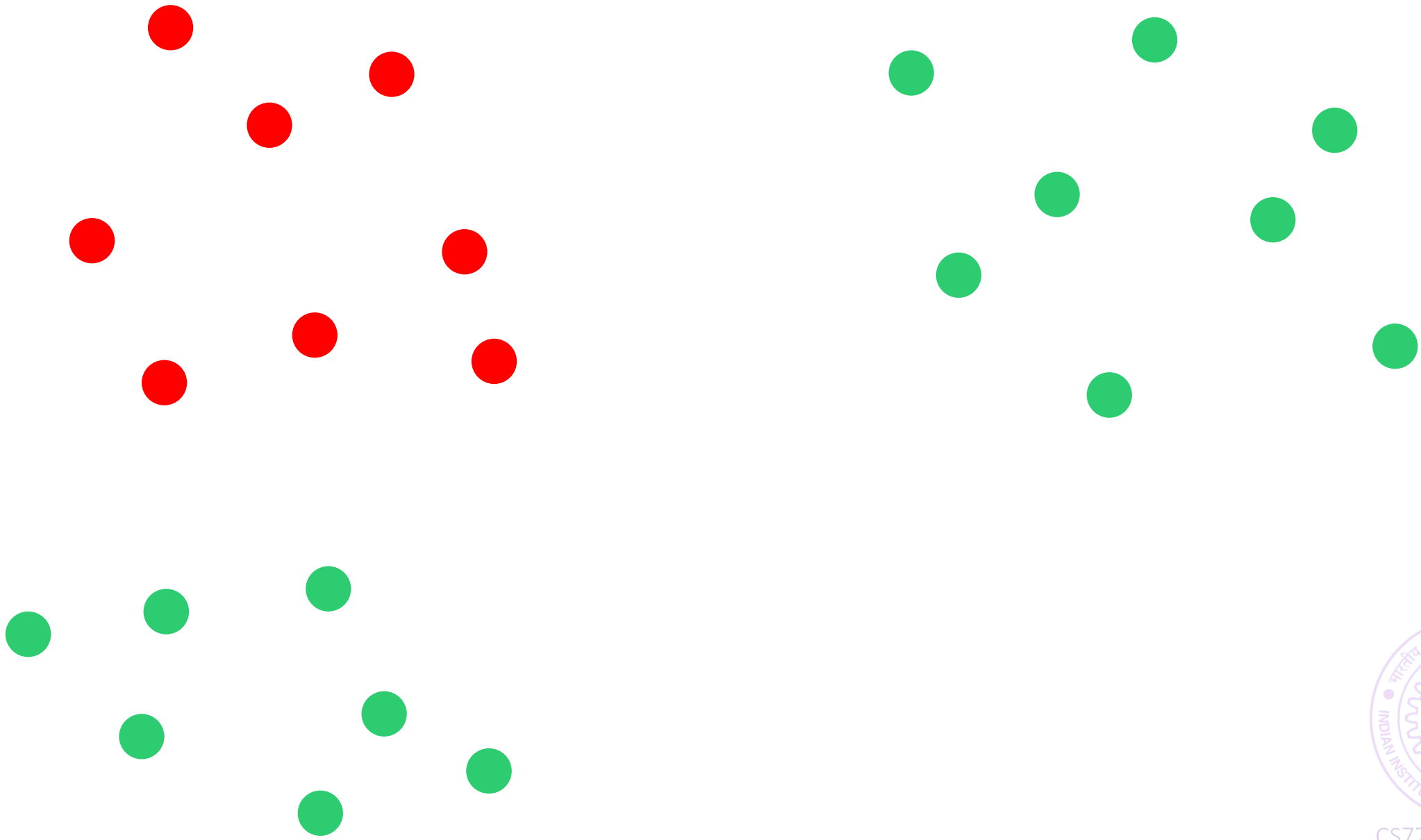
$= \left(\mu_- , \mu_+ \right)$

In fact I can even handle cases where there are more than two classes e.g. instead of spam/non-spam, suppose the classes were Primary, Social, Updates …

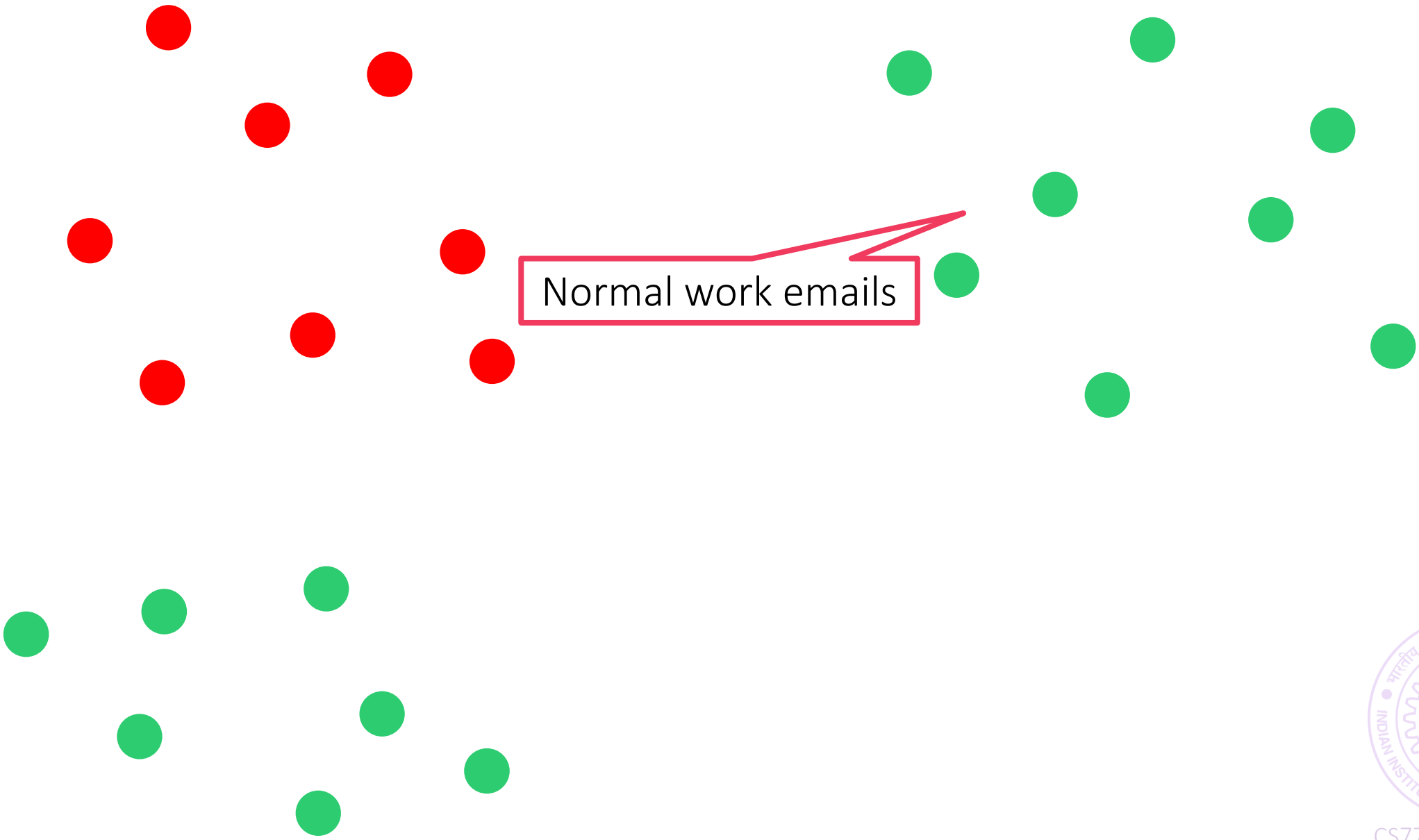


Learning with Prototypes - Issues

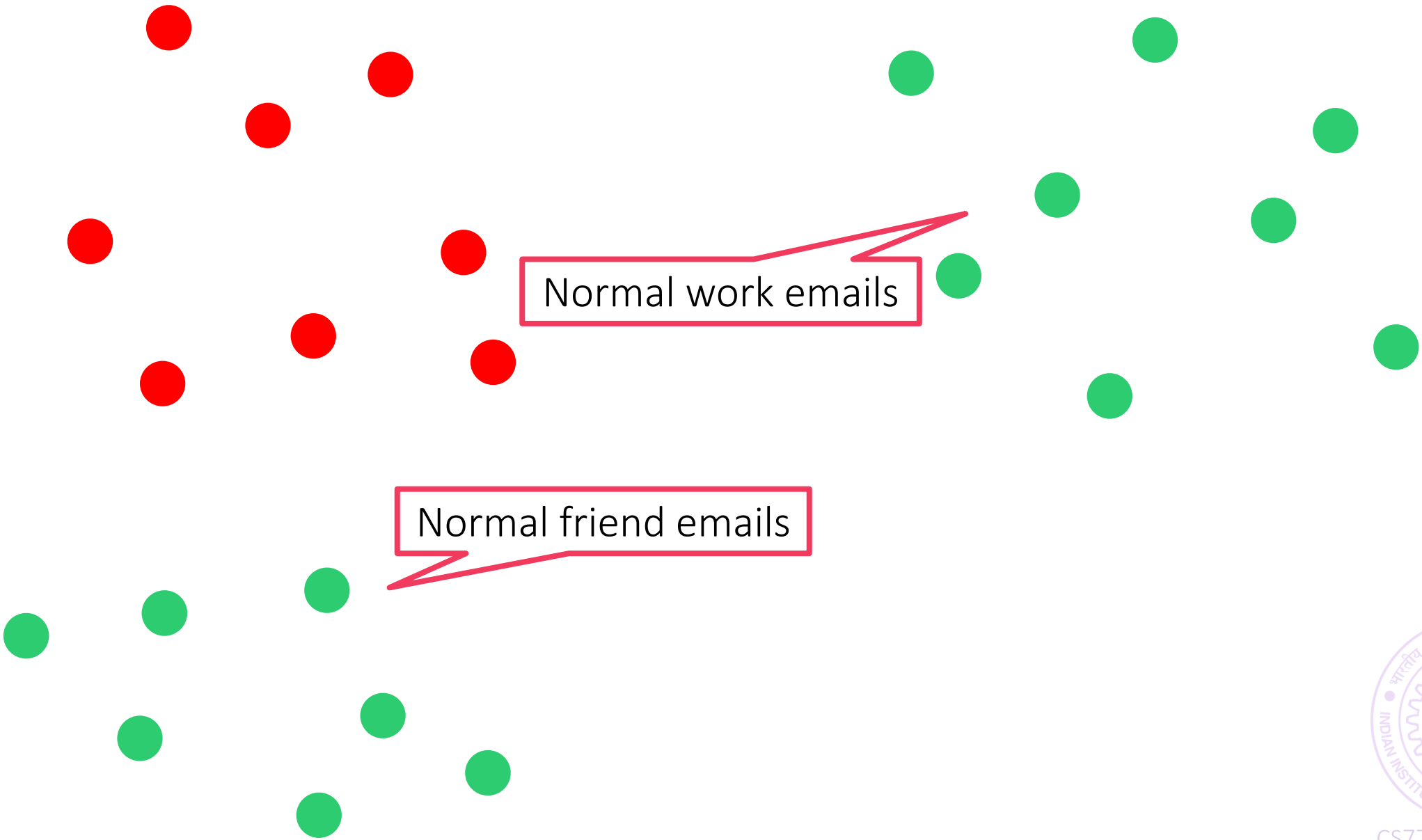
Learning with Prototypes - Issues



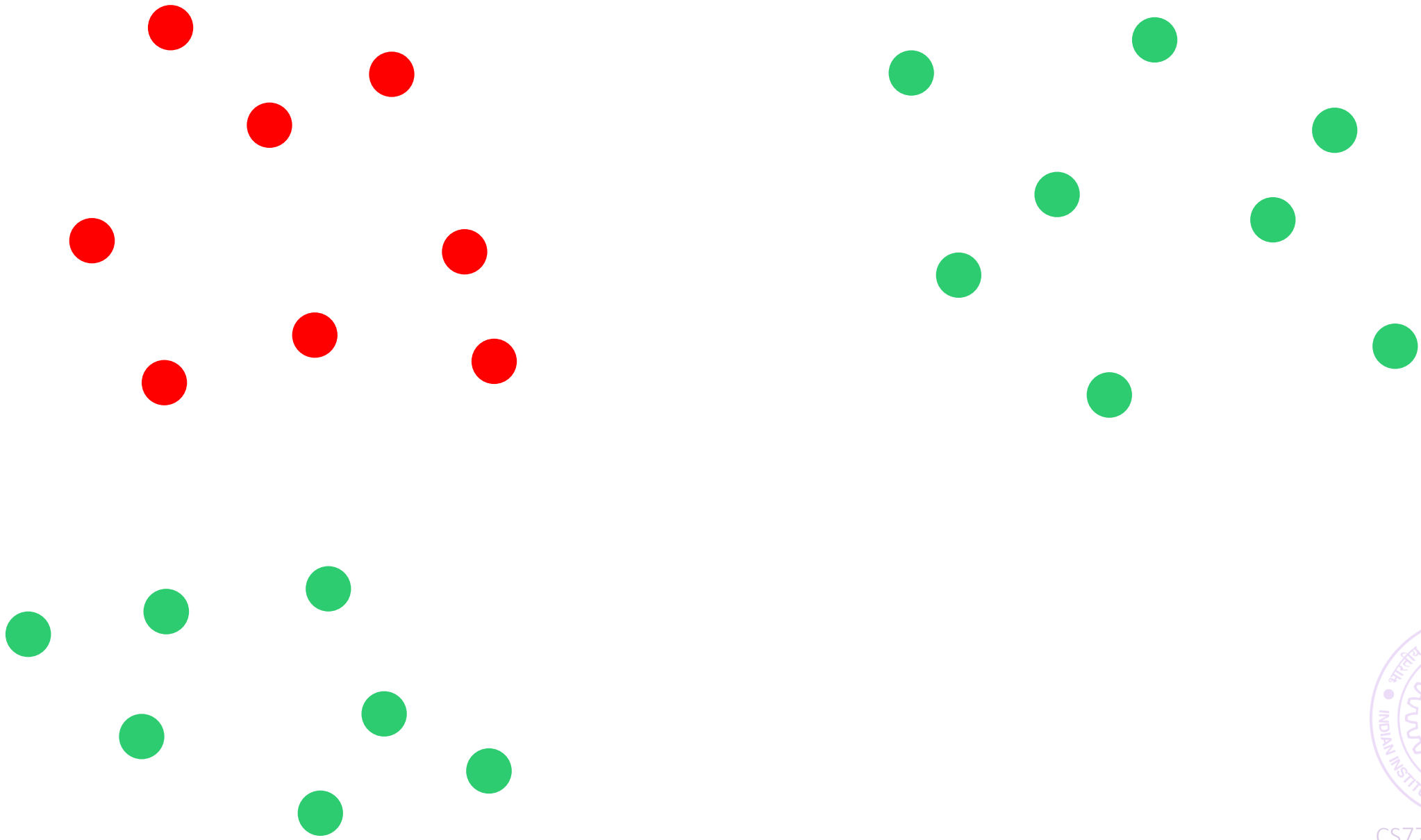
Learning with Prototypes - Issues



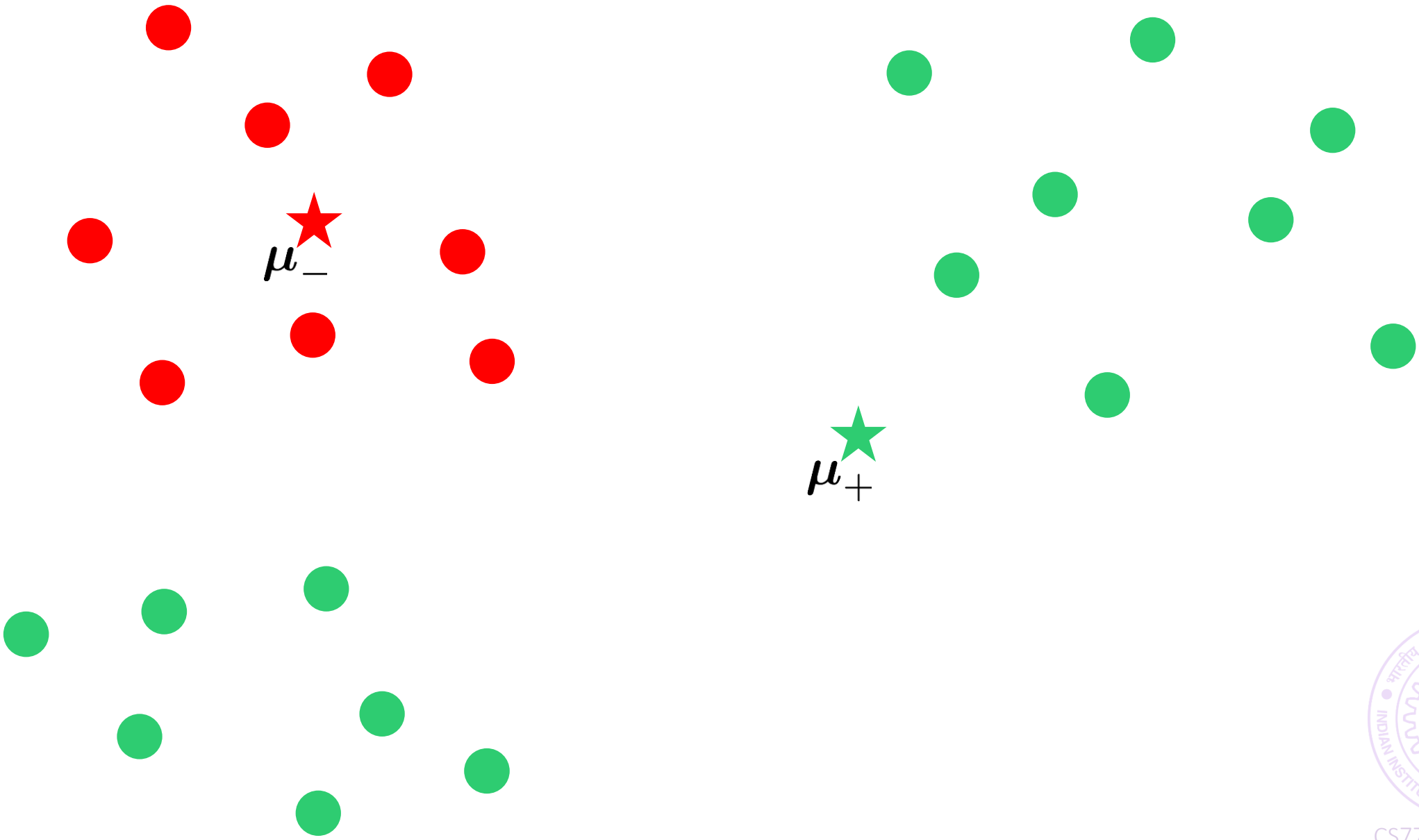
Learning with Prototypes - Issues



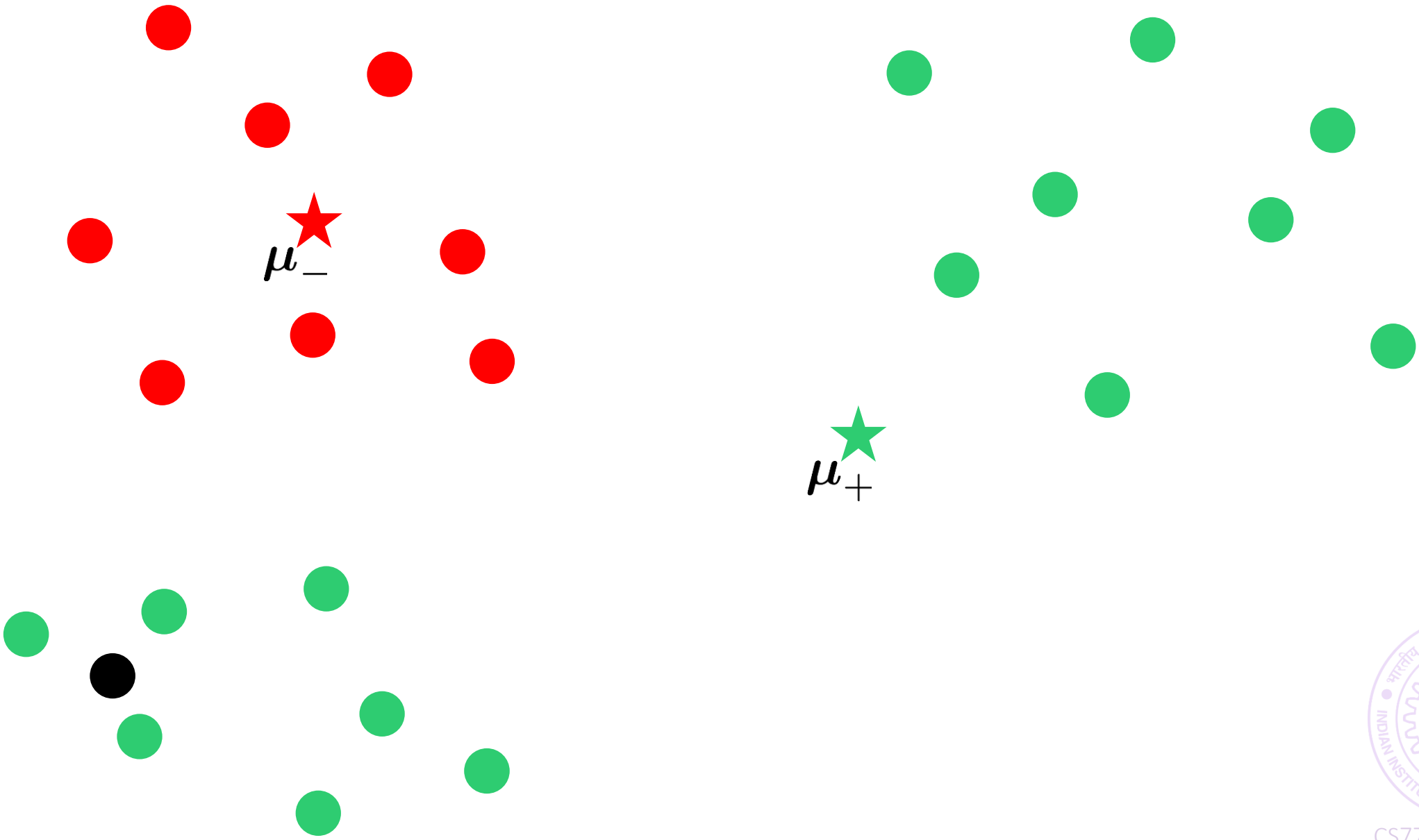
Learning with Prototypes - Issues



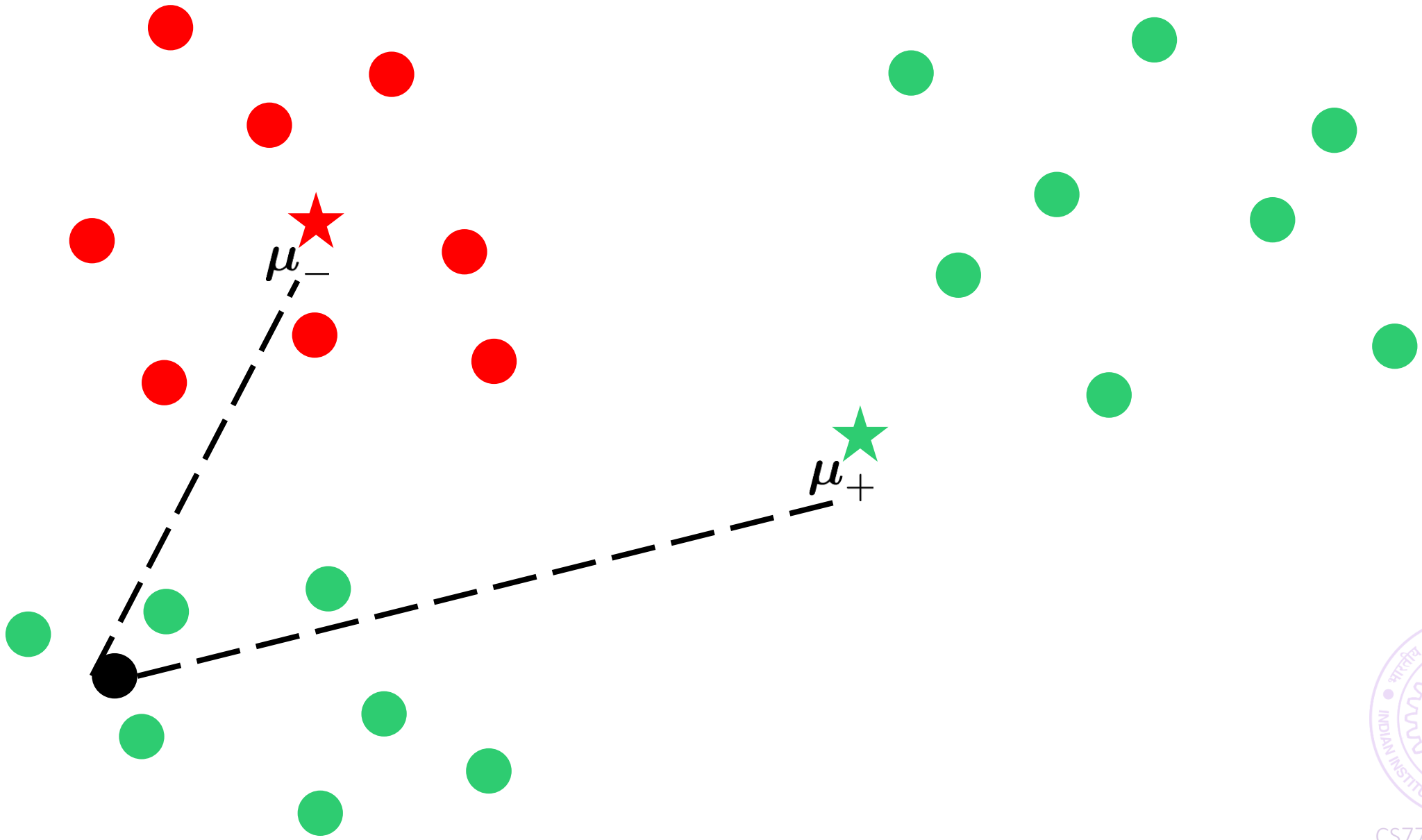
Learning with Prototypes - Issues



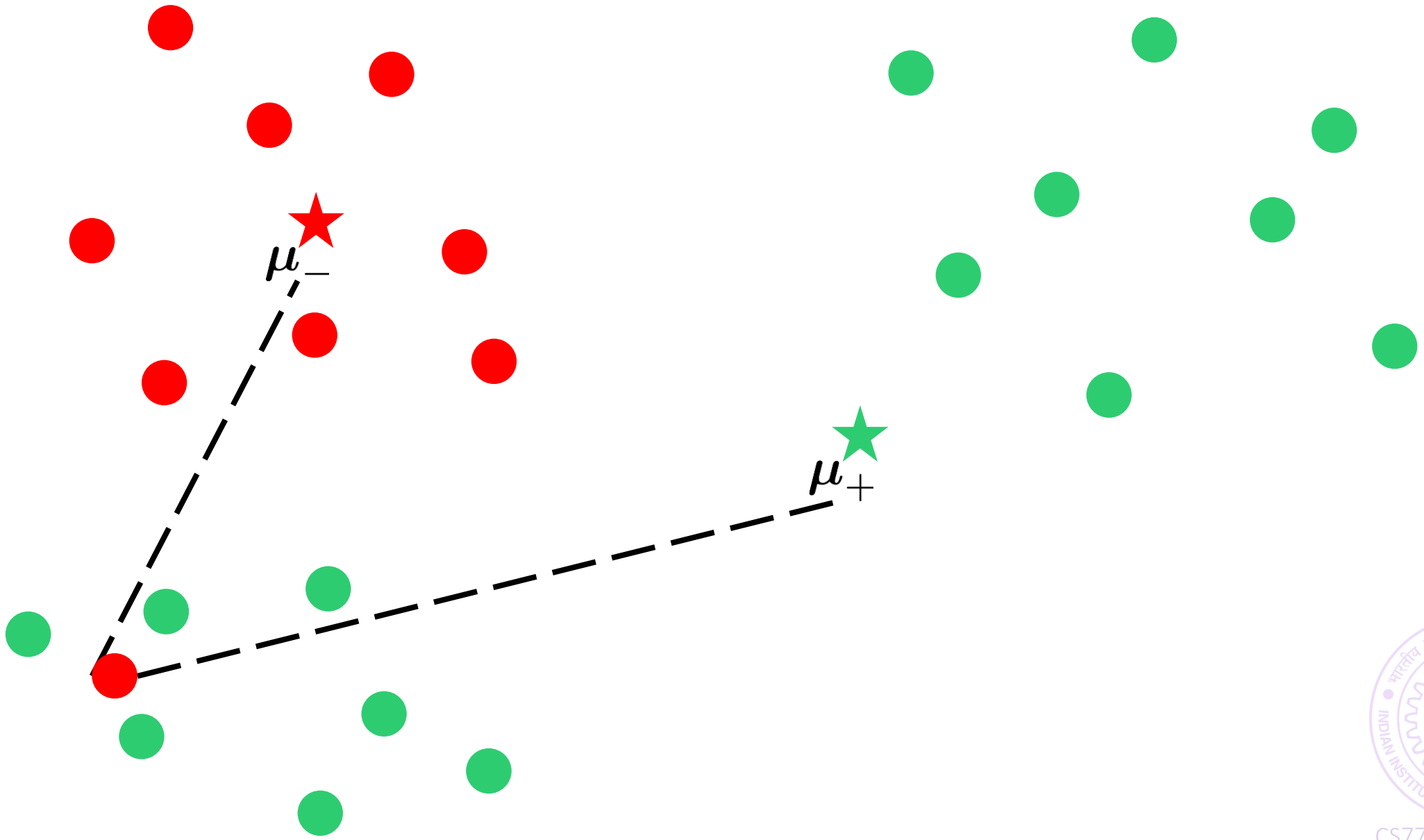
Learning with Prototypes - Issues



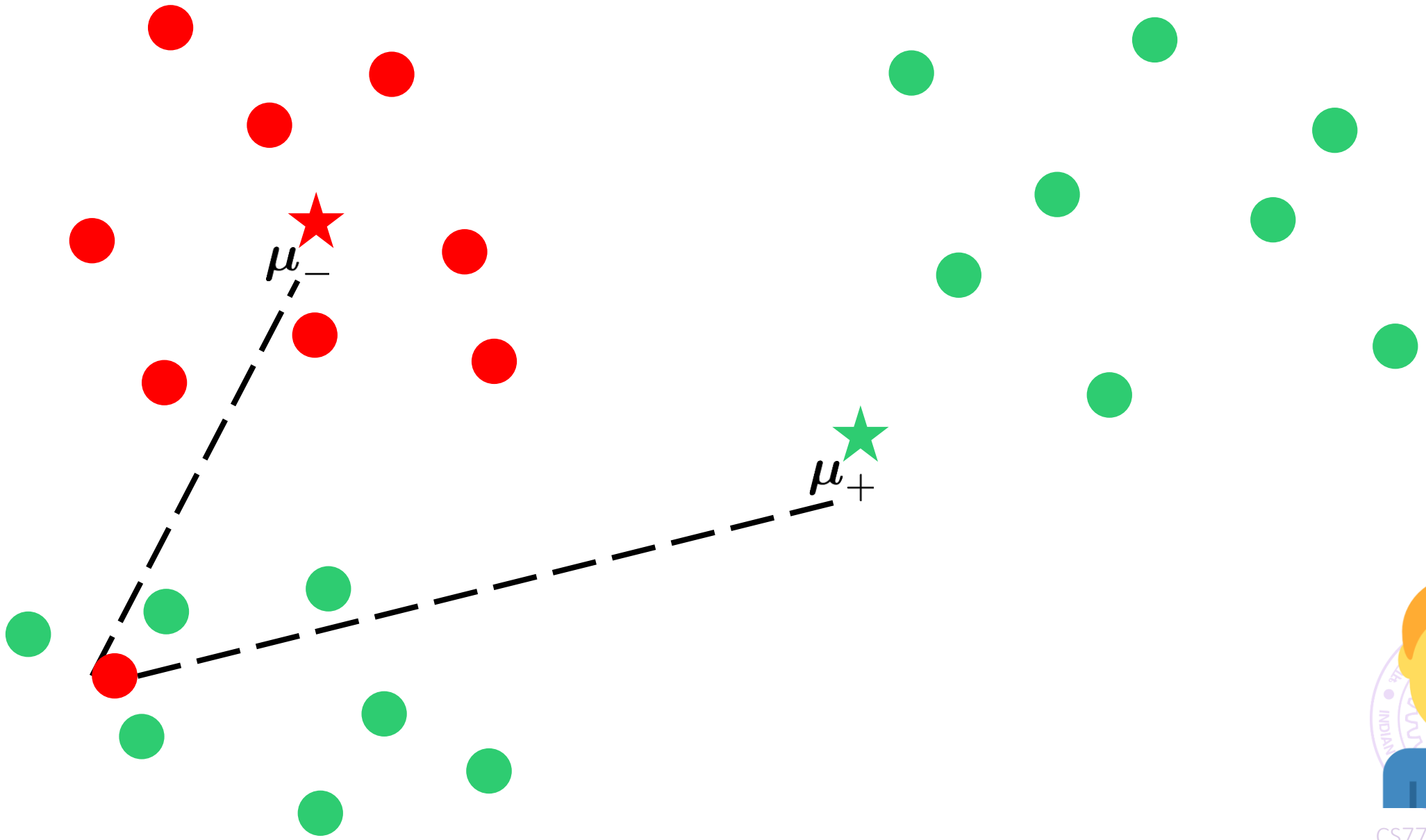
Learning with Prototypes - Issues



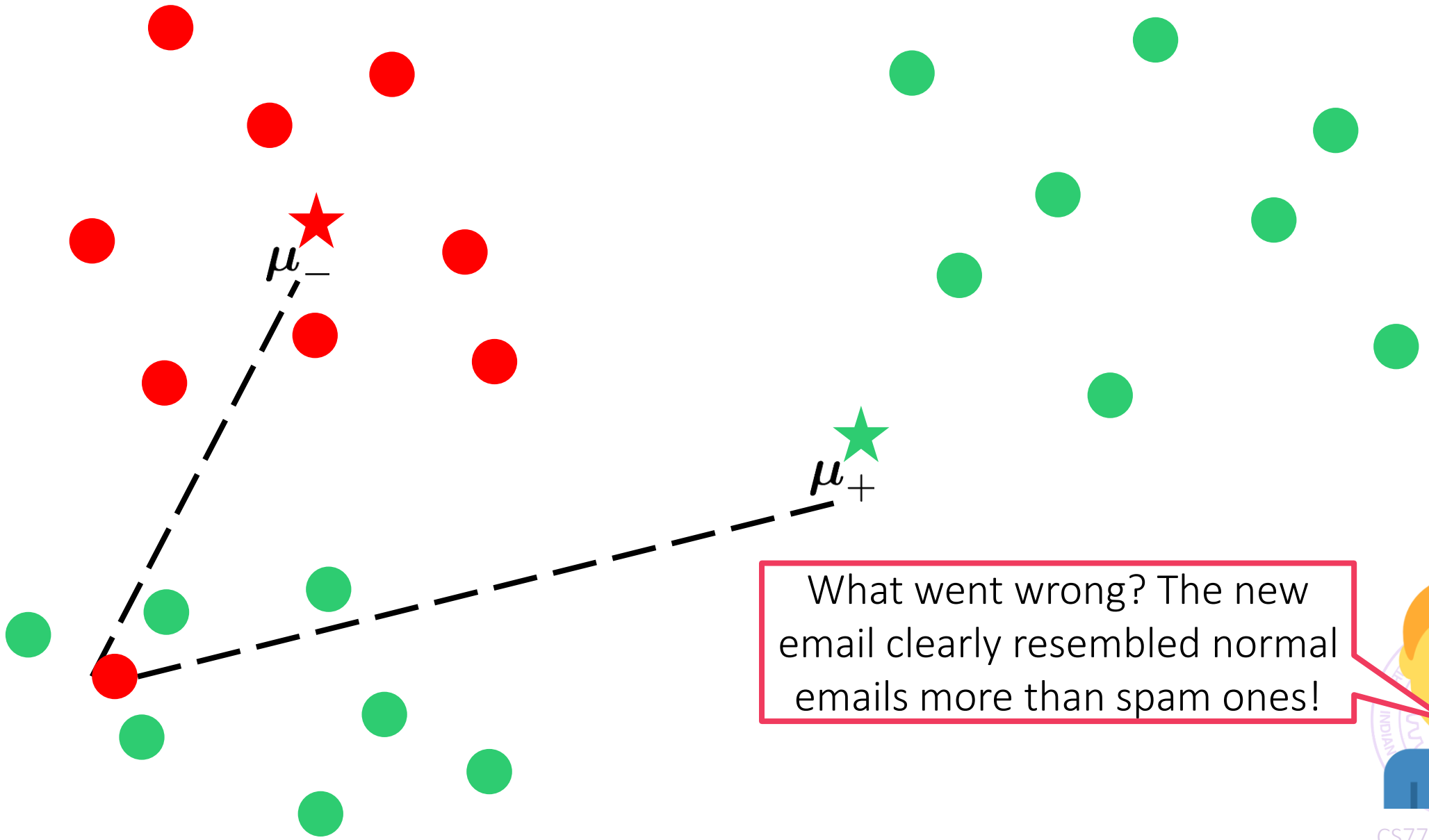
Learning with Prototypes - Issues



Learning with Prototypes - Issues

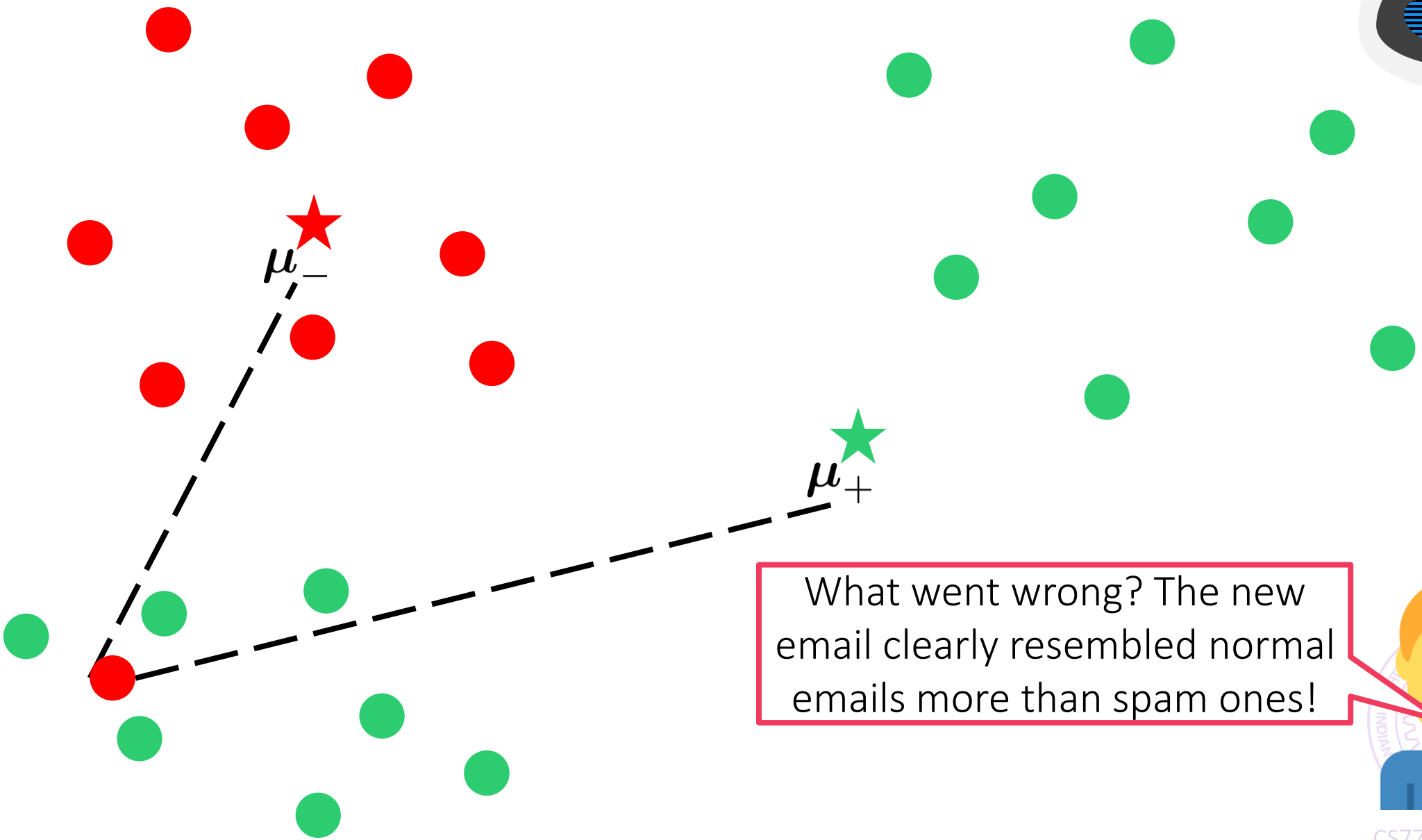


Learning with Prototypes - Issues



What went wrong? The new email clearly resembled normal emails more than spam ones!

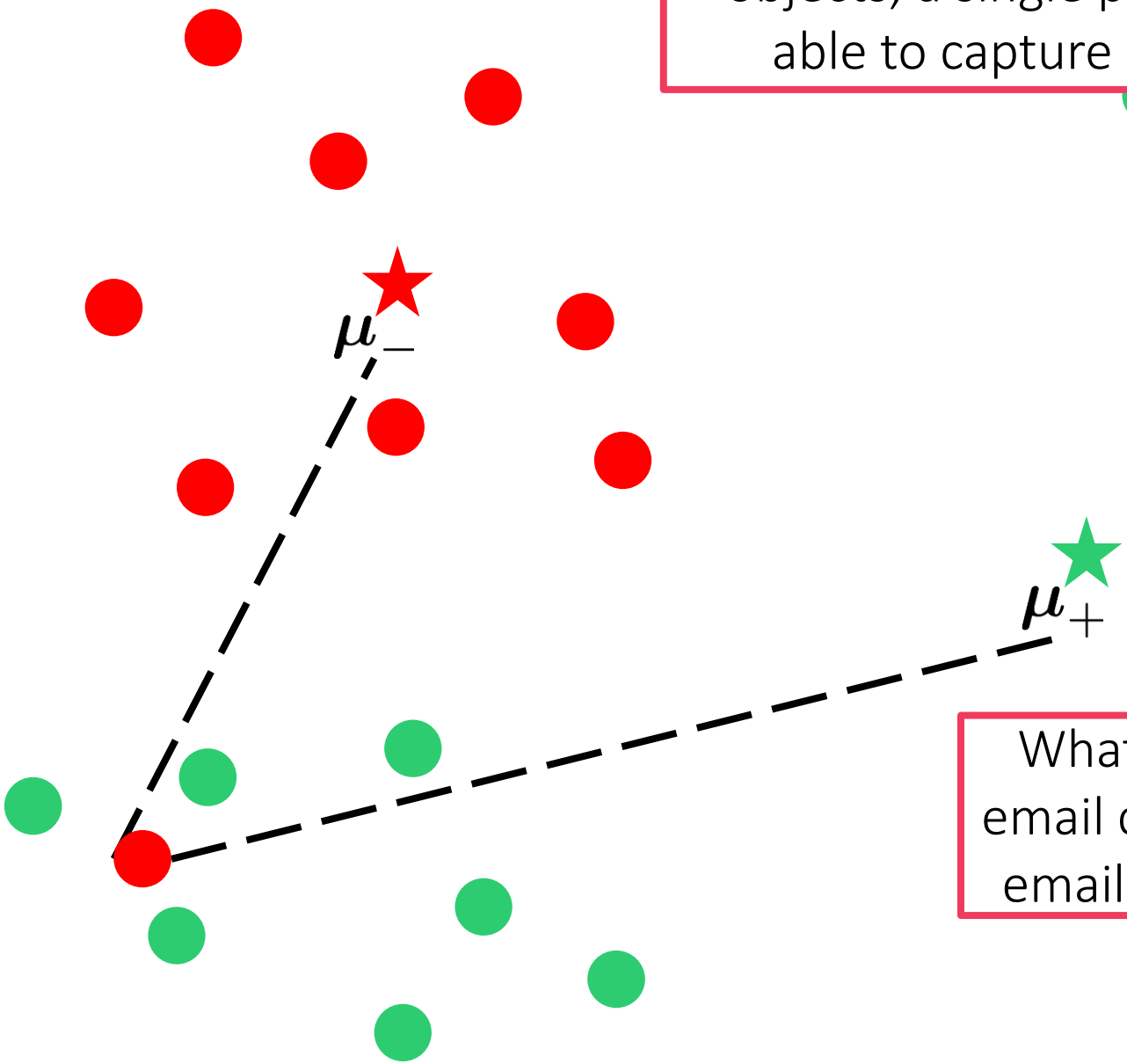
Learning with Prototypes - Issues



What went wrong? The new email clearly resembled normal emails more than spam ones!

Learning with Pro

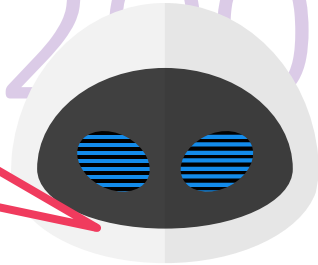
If there is too much diversity in a class of objects, a single prototype may not be able to capture all the information

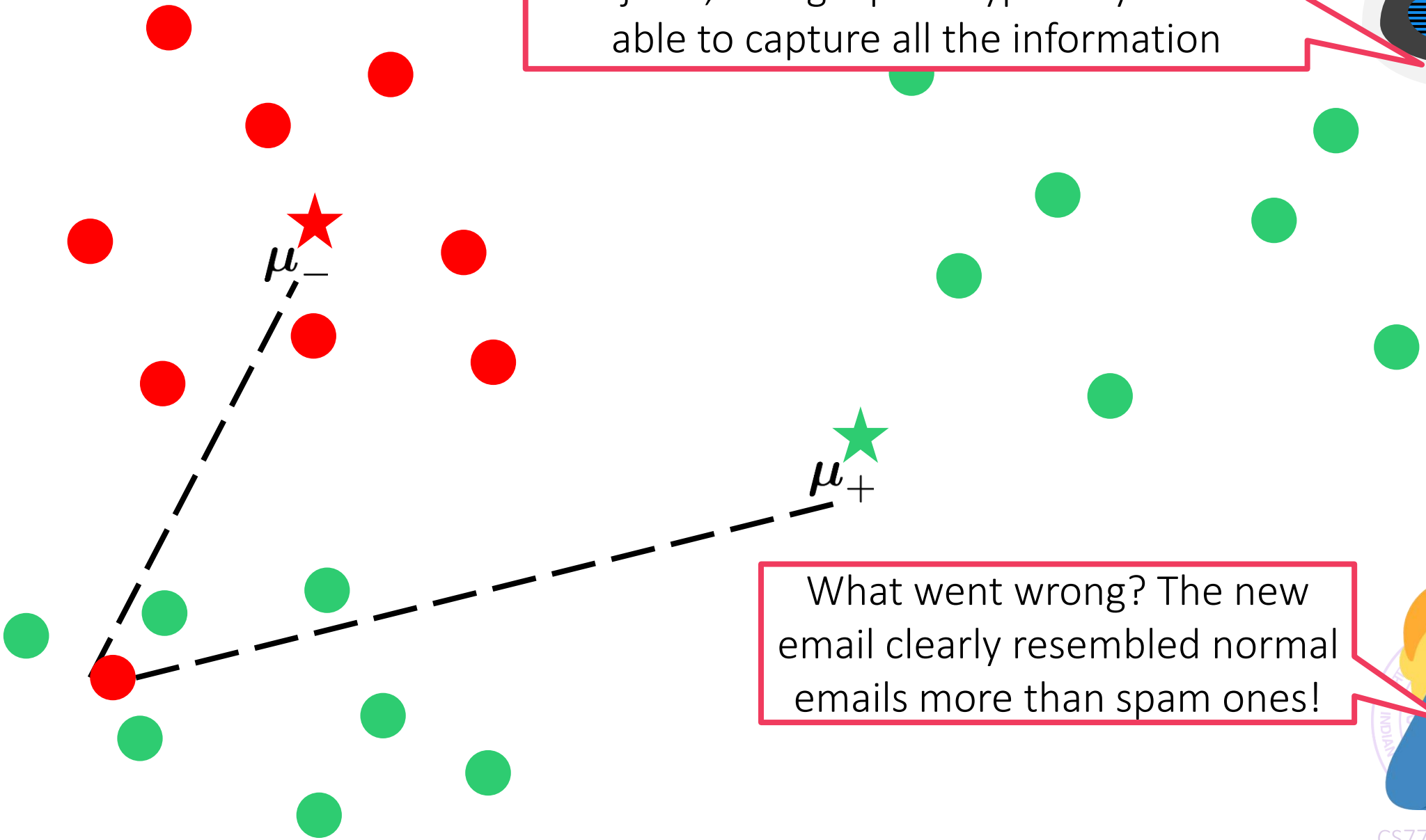


What went wrong? The new email clearly resembled normal emails more than spam ones!

Learning with Pro

If there is too much diversity in a class of objects, a single prototype may not be able to capture all the information

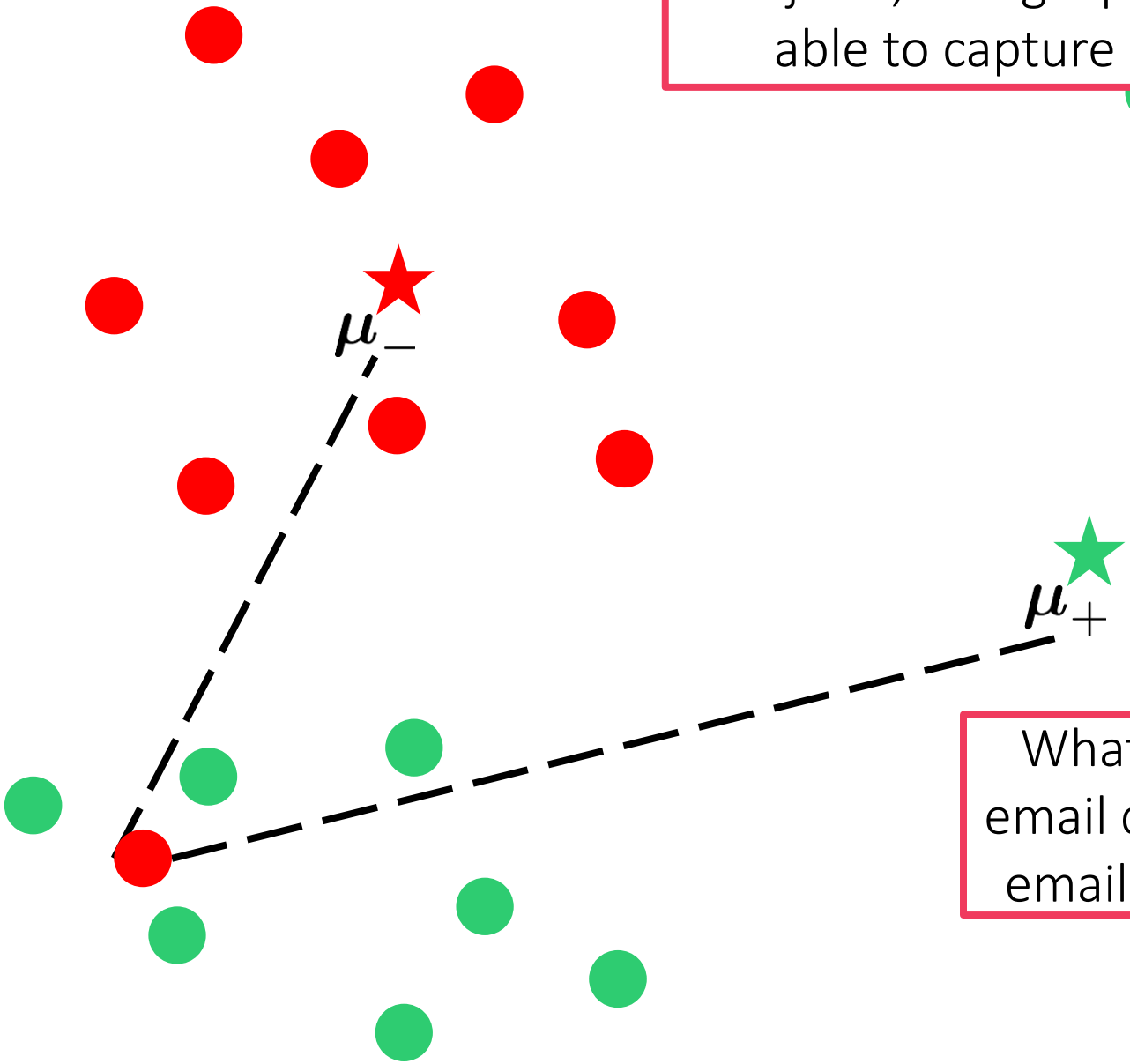




What went wrong? The new email clearly resembled normal emails more than spam ones!

Learning with Pro

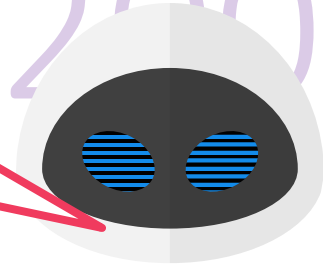
If there is too much diversity in a class of objects, a single prototype may not be able to capture all the information

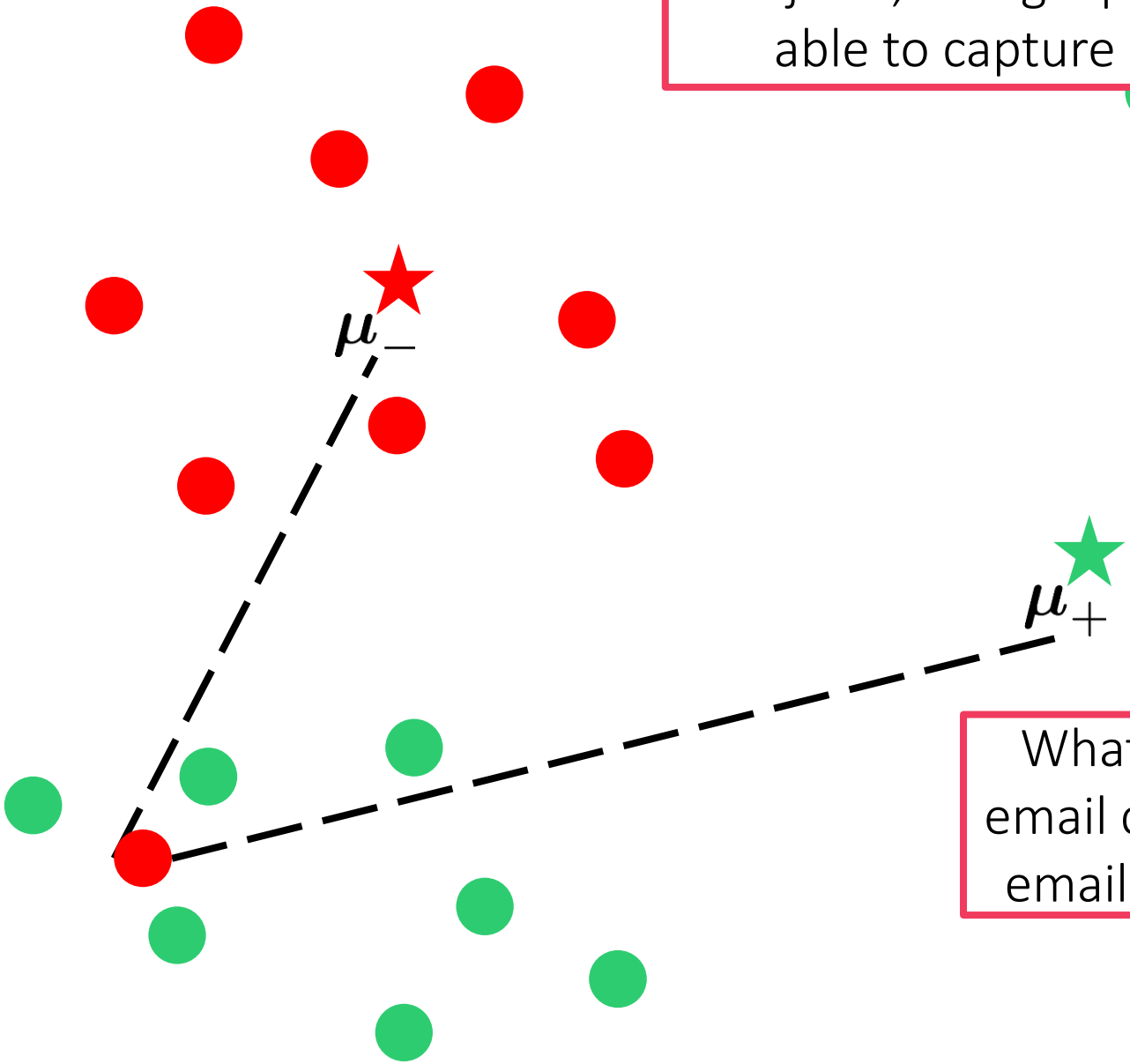


What went wrong? The new email clearly resembled normal emails more than spam ones!

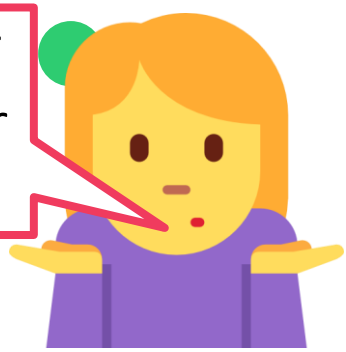
Learning with Pro

If there is too much diversity in a class of objects, a single prototype may not be able to capture all the information



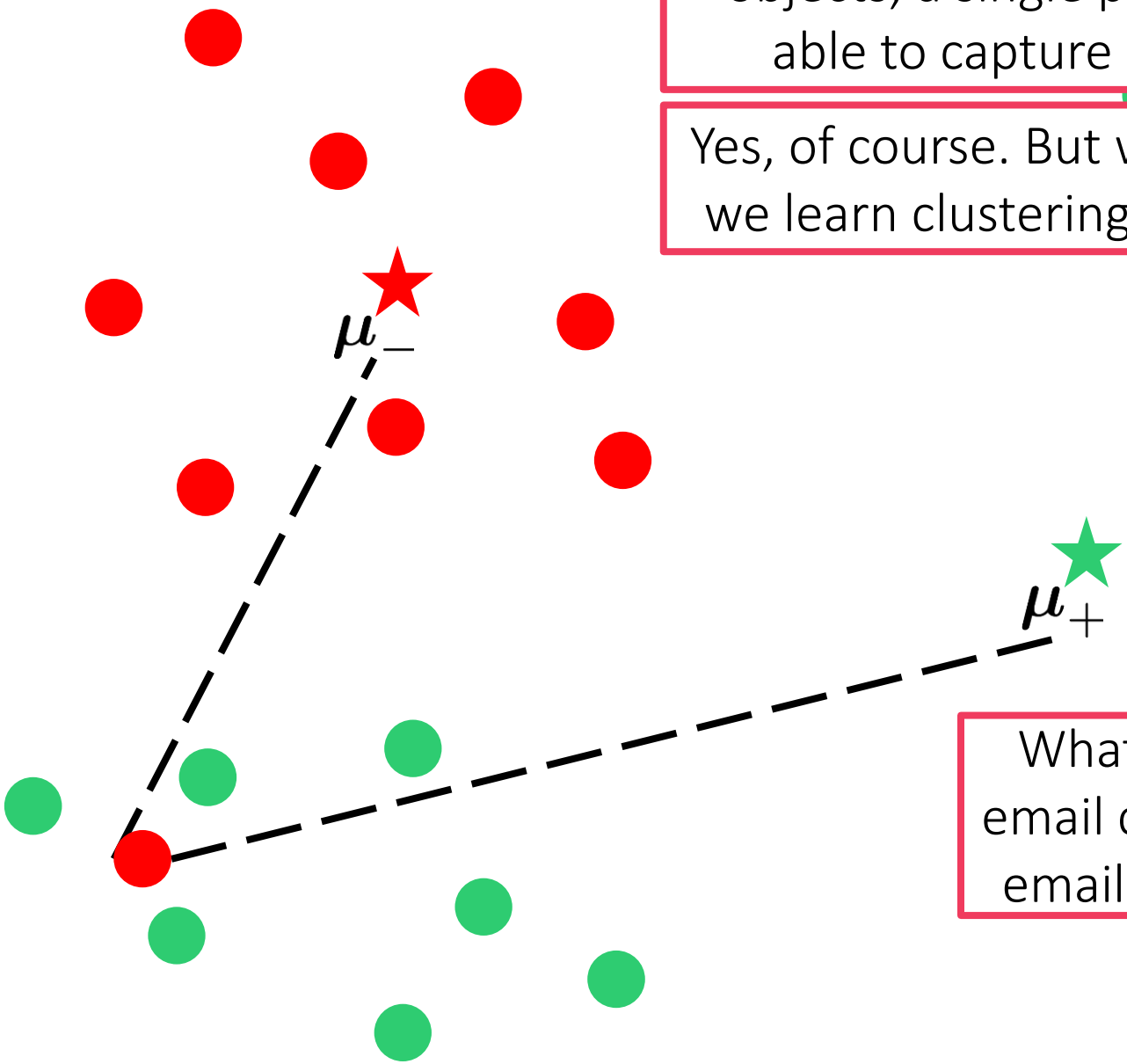


Can we have two or more prototypes for the normal class?

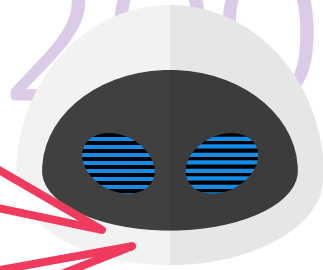


What went wrong? The new email clearly resembled normal emails more than spam ones!

Learning with Pro

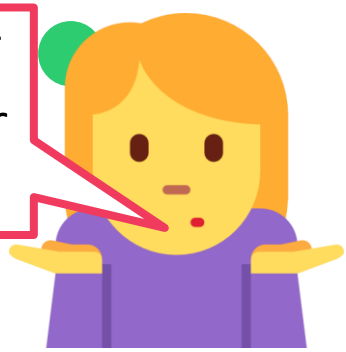


If there is too much diversity in a class of objects, a single prototype may not be able to capture all the information



Yes, of course. But we will have to wait till we learn clustering before we can do so.

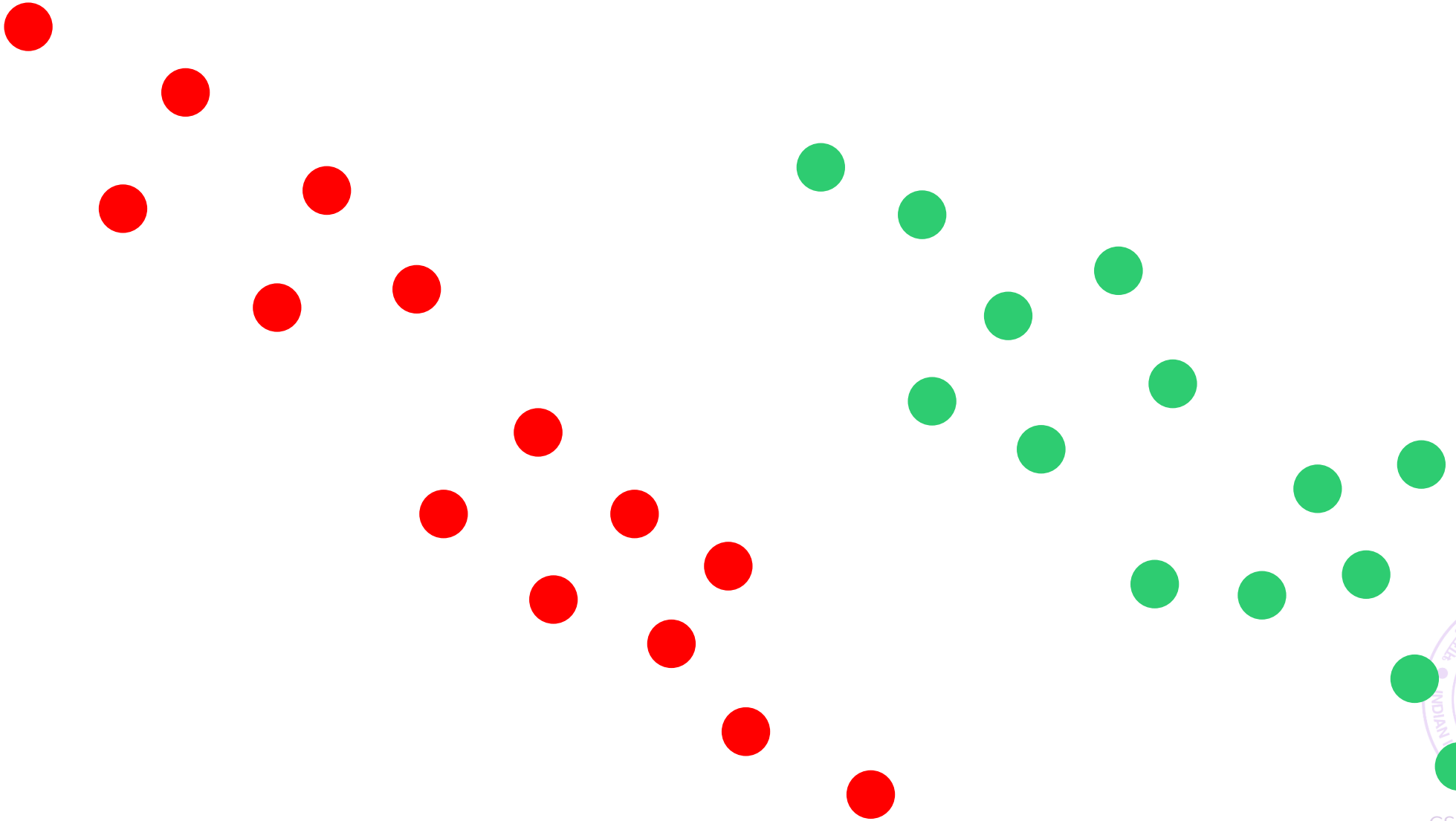
Can we have two or more prototypes for the normal class?



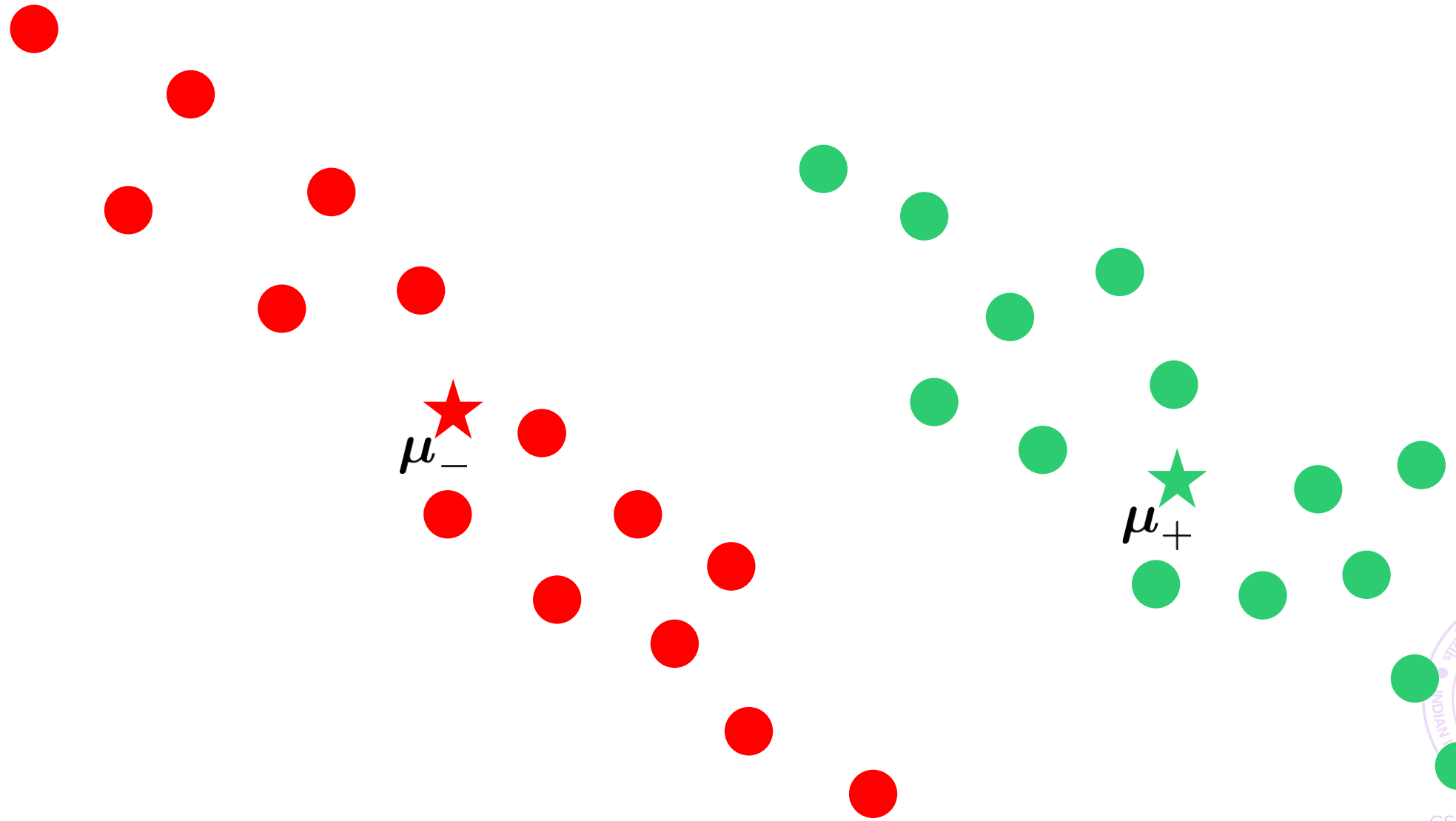
What went wrong? The new email clearly resembled normal emails more than spam ones!

Learning with Prototypes - Issues

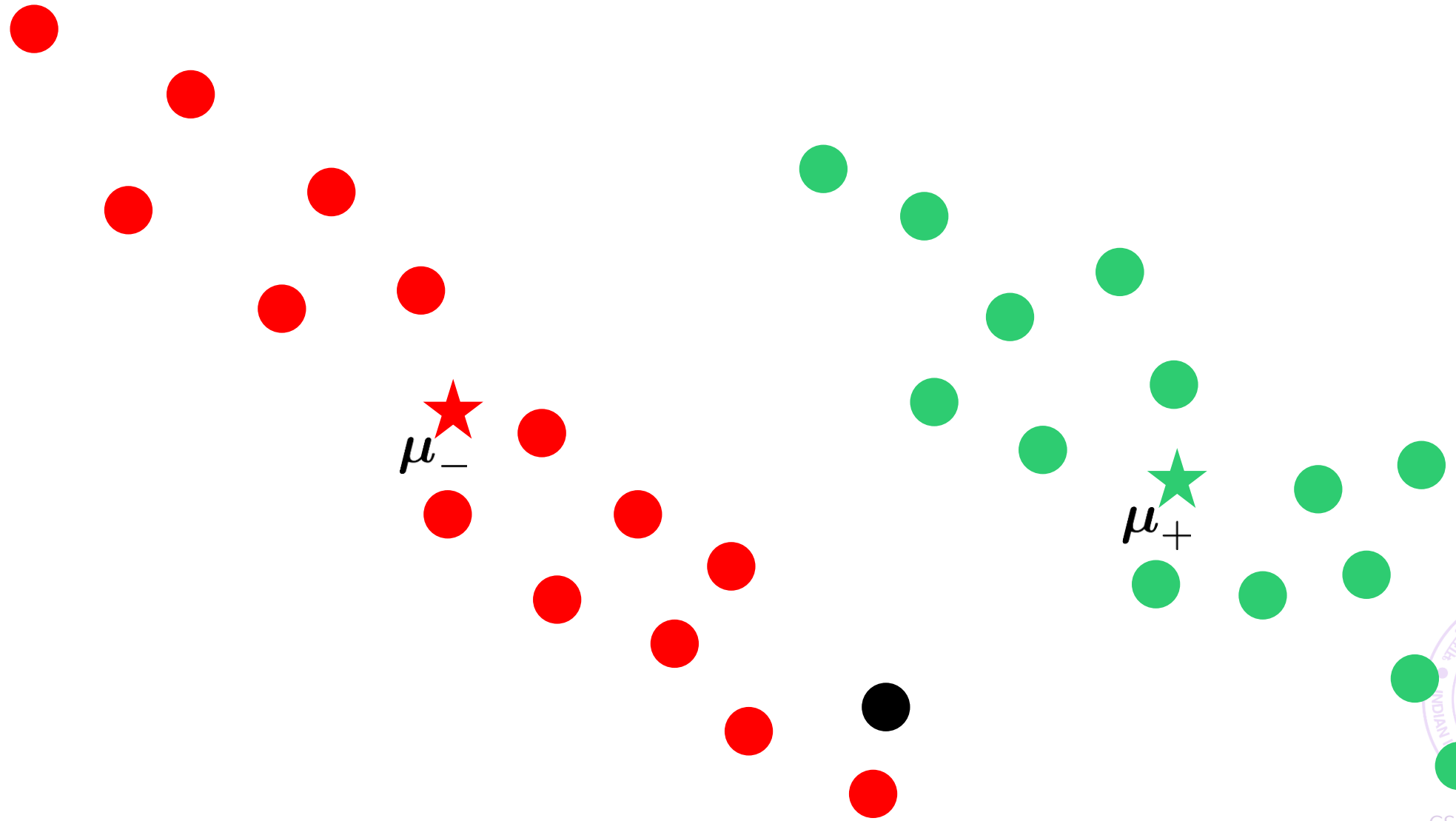
Learning with Prototypes - Issues



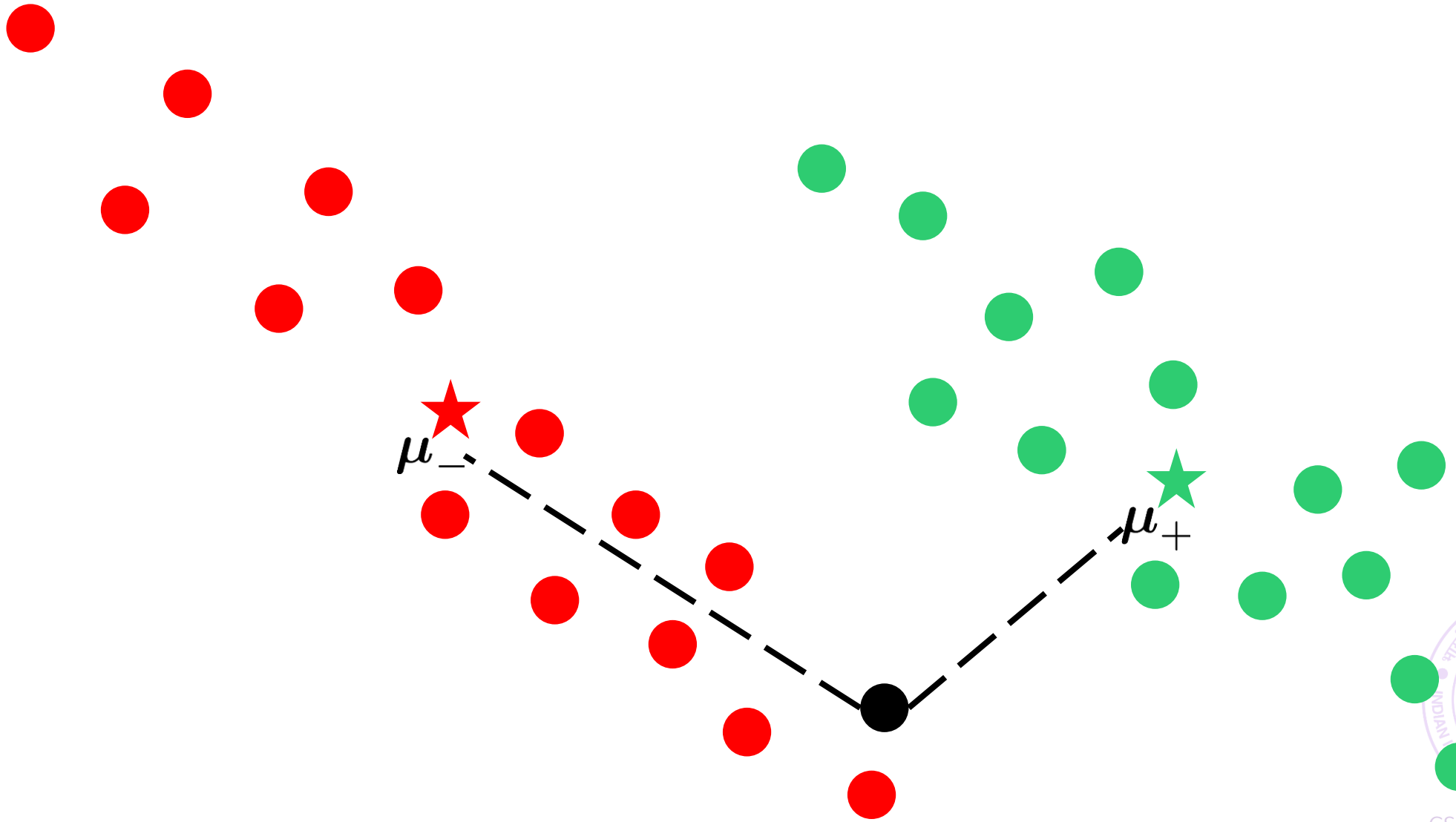
Learning with Prototypes - Issues



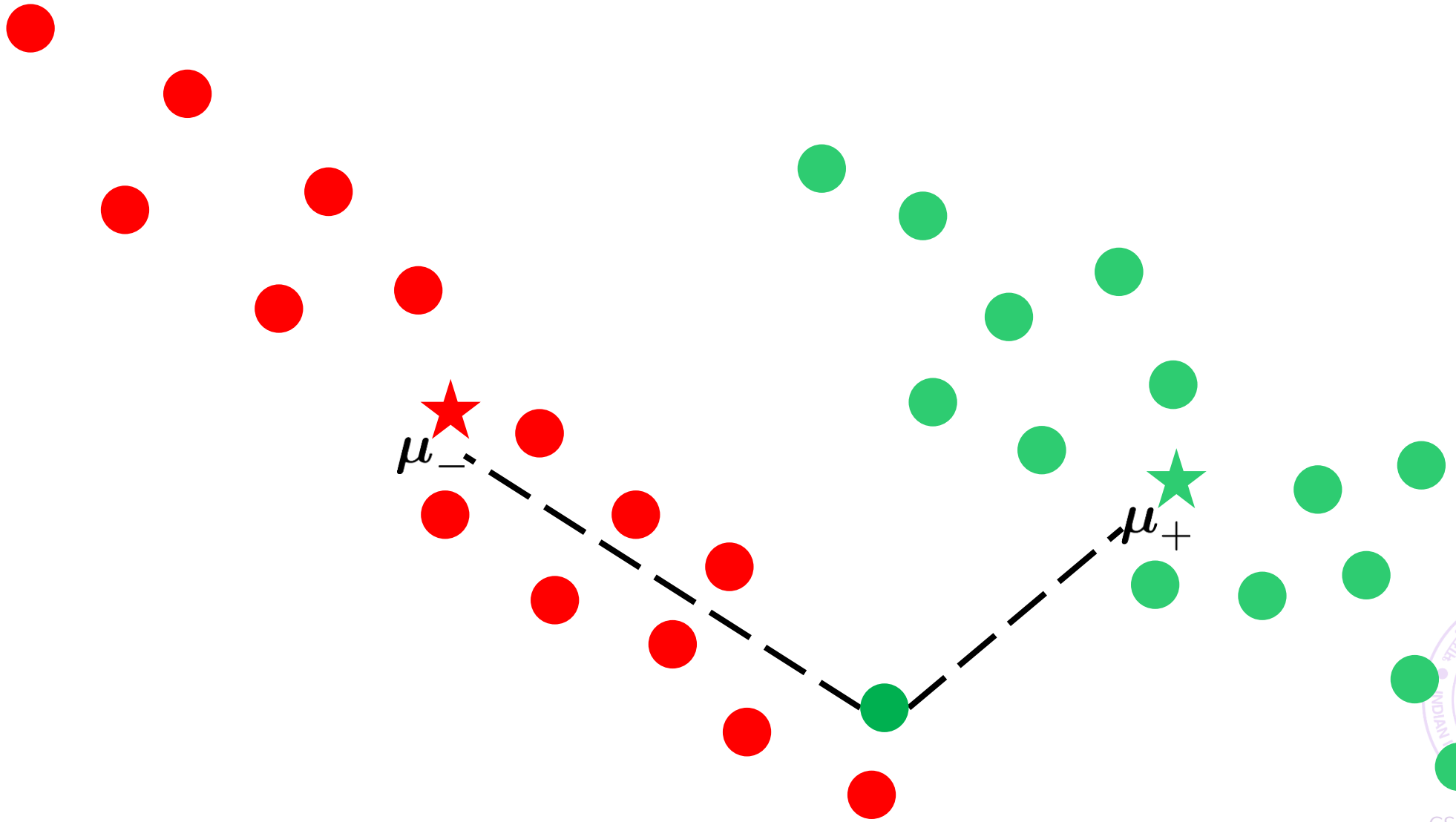
Learning with Prototypes - Issues



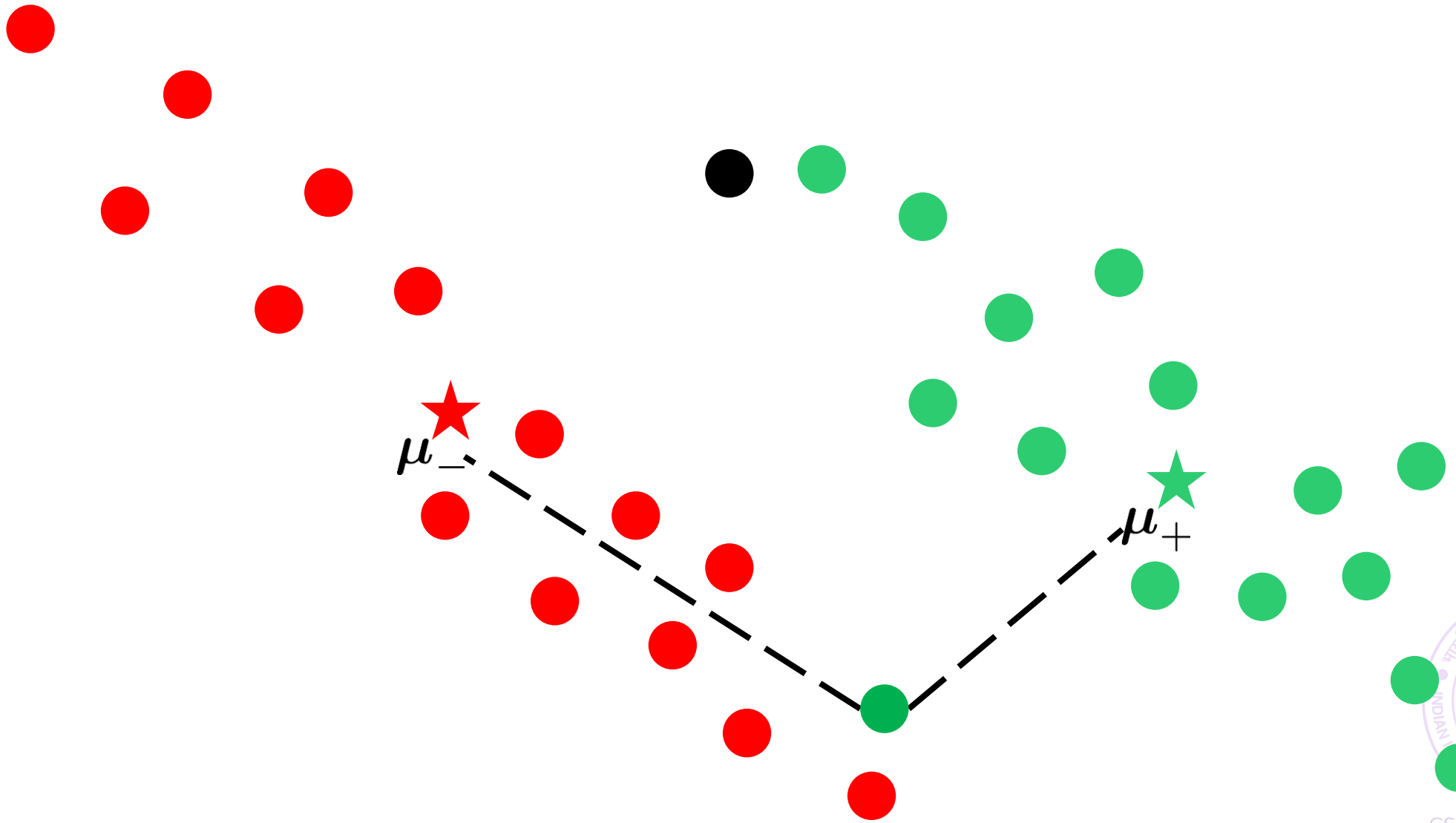
Learning with Prototypes - Issues



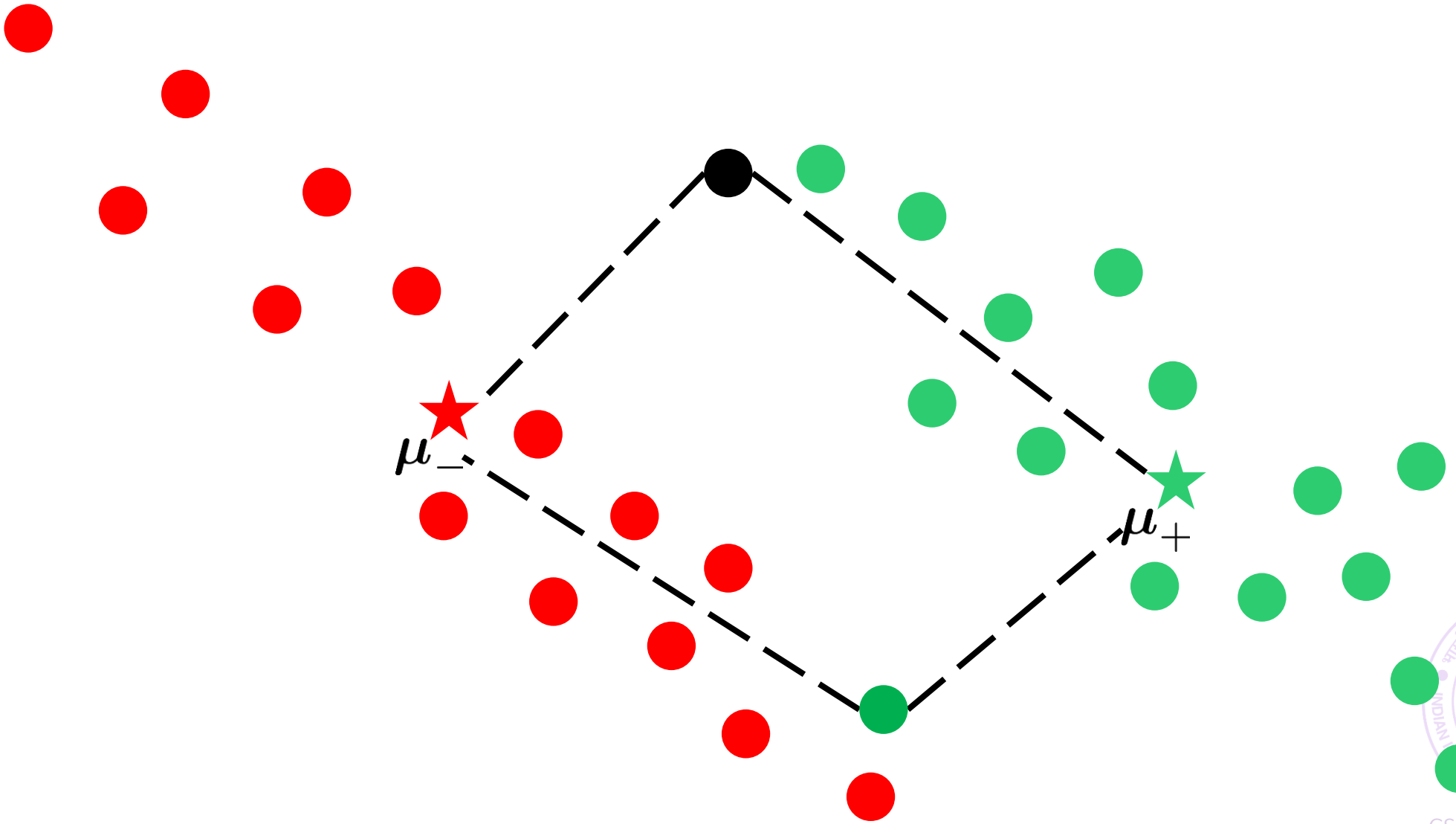
Learning with Prototypes - Issues



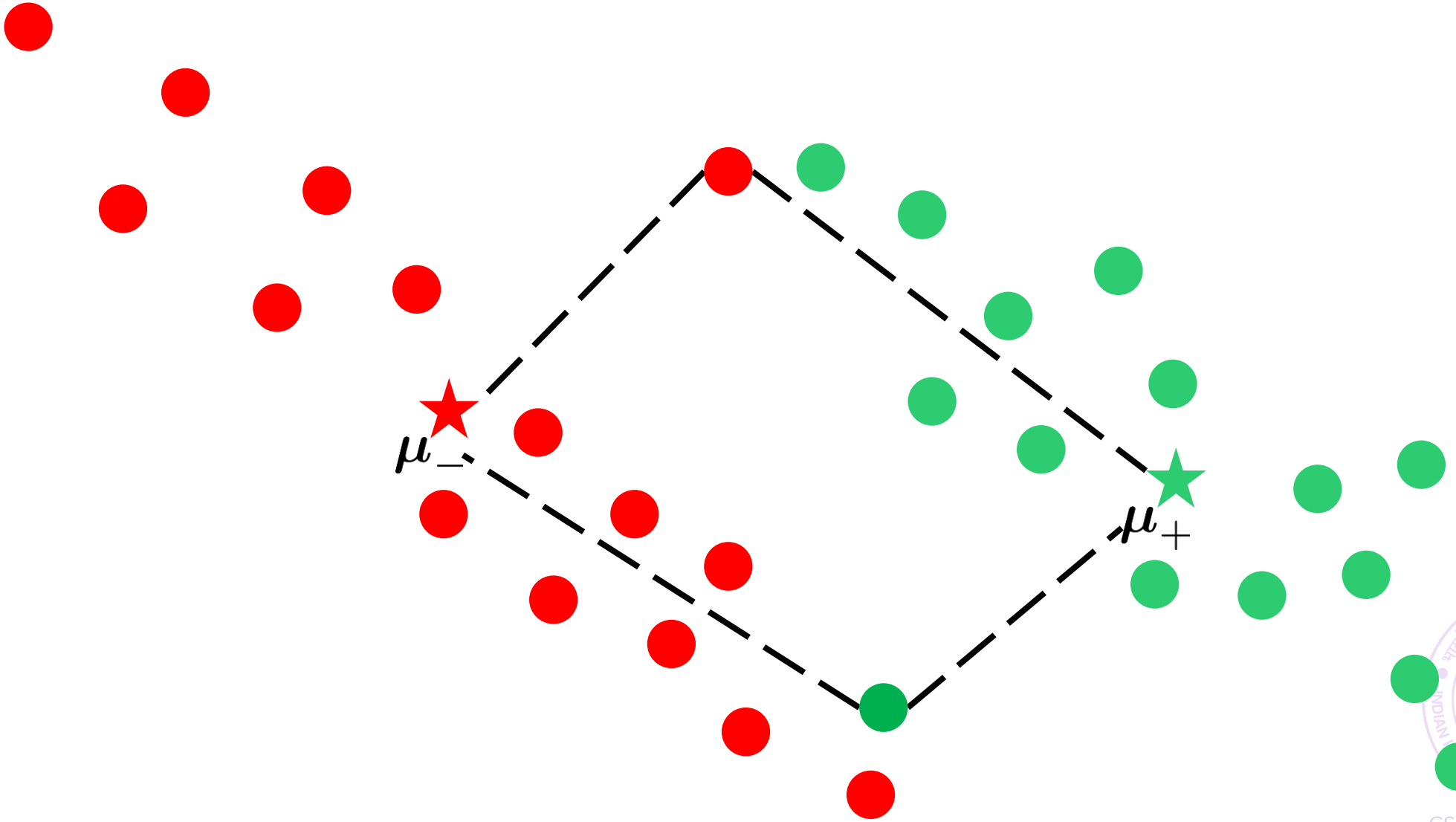
Learning with Prototypes - Issues



Learning with Prototypes - Issues



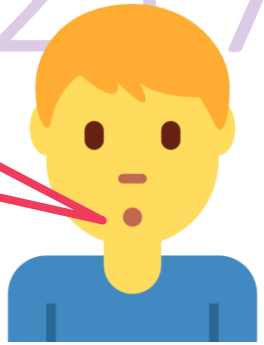
Learning with Prototypes - Issues

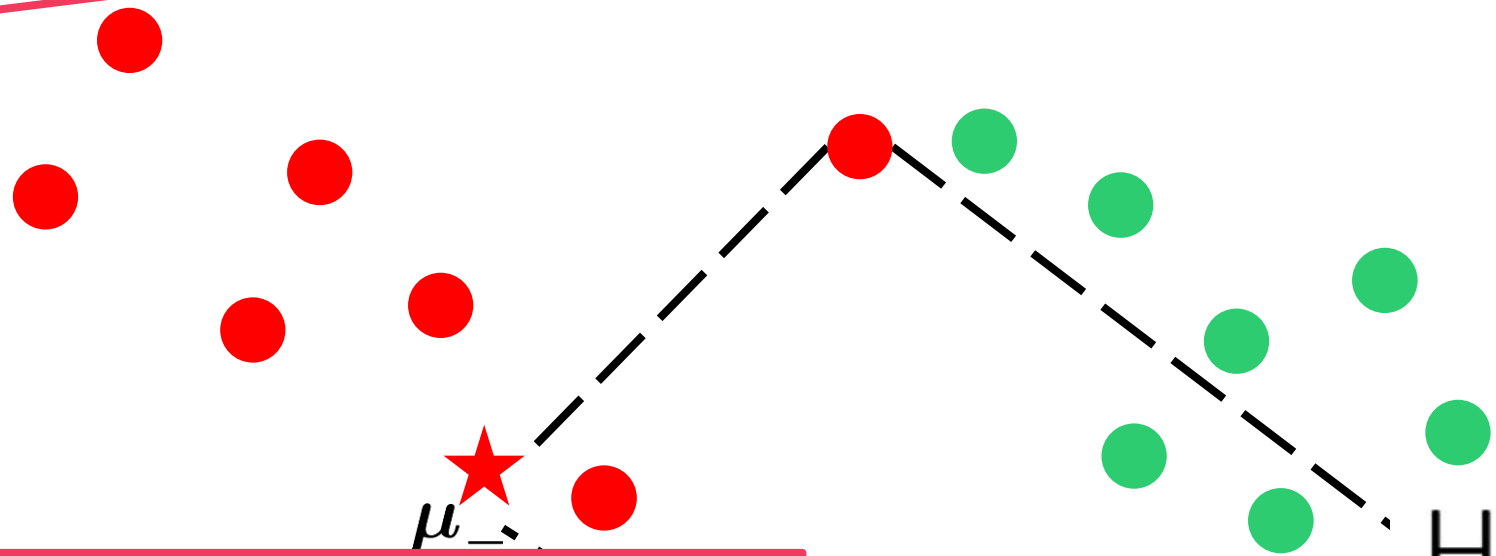


Learn...es - Issues

I think this time another issue is how we calculated distances. Can we use non-Euclidean distances?

What went wrong this time?





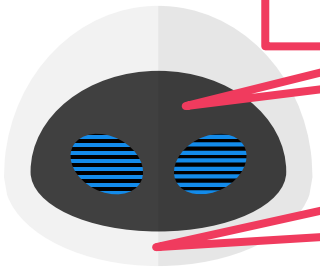
Well, notice that there is still a lot of intra-class diversity within the spam and the normal email classes

Has wings

Can fly

Can sing

Has a beak

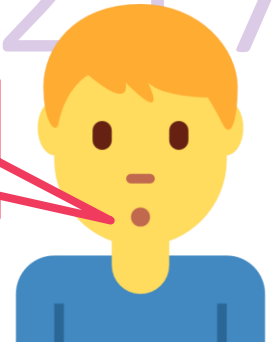


I think so too! E.g. we can give different weights to features while calculating distances

Learning with Prototypes - Issues

I think this time another issue is how we calculated distances. Can we use non-Euclidean distances?

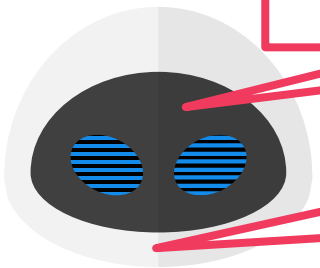
What went wrong this time?



$$d(\mathbf{x}^1, \mathbf{x}^2) = \sqrt{\sum_{j=1}^{d} w_j \cdot \left(\mathbf{x}_j^1 - \mathbf{x}_j^2\right)^2}$$

μ_-

Well, notice that there is still a lot of intra-class diversity within the spam and the normal email classes



I think so too! E.g. we can give different weights to features while calculating distances

Has wings

Can fly

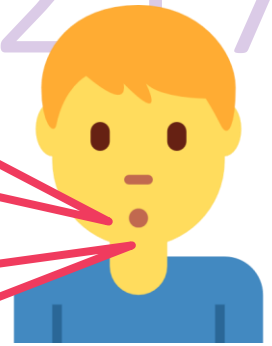
Can sing

Has a beak

Learning with Prototypes - Issues

I think this time another issue is how we calculated distances. Can we use non-Euclidean distances?

Yes, but let us revisit this issue later



What went wrong this time?

Can we learn these weights too?

$$d(\mathbf{x}^1, \mathbf{x}^2) = \sqrt{\sum_{j=1}^{d} w_j \cdot \left(\mathbf{x}_j^1 - \mathbf{x}_j^2\right)^2}$$

μ_-

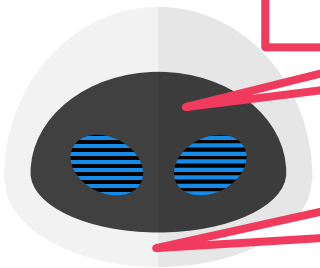
Has wings

Can fly

Can sing

Has a beak

Well, notice that there is still a lot of intra-class diversity within the spam and the normal email classes



I think so too! E.g. we can give different weights to features while calculating distances

Learning with Prototypes - Lessons

An extremely simple technique to classify data

Can do binary classification (2-classes) as well as multi-classification

Very compact, light-weight model (one prototype per class)

Actually used in industrial applications like extreme classification

Actually state of the art if we have very few data points for a class

For example, if we have very few spam emails in training data

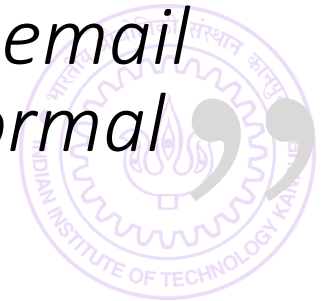
Works well when class data points are packed closely – less diversity

Improvements possible using multiple prototypes, metric learning (using a non-Euclidean distance function)

Nearest Neighbour Classifier

- A really extreme form of learning with prototypes
- Recall that we commented that having multiple (say 2-3) prototypes may help when there is lots of diversity in a class
- With NN, every single training point becomes a prototype ☺
- The basic mantra here seems to be

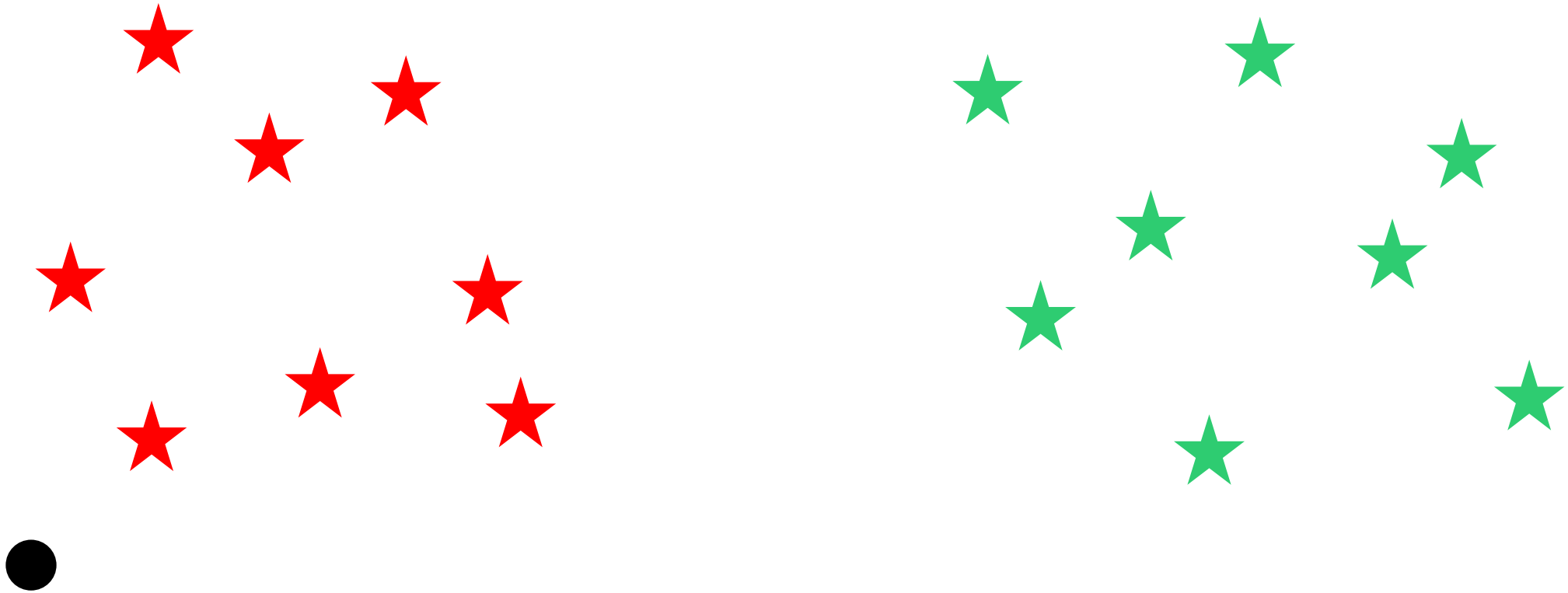
> *If the training email that most resembles a new email was spam, maybe the new email is spam too. Else, if a normal training email most resembles the new email, the new email should be normal*



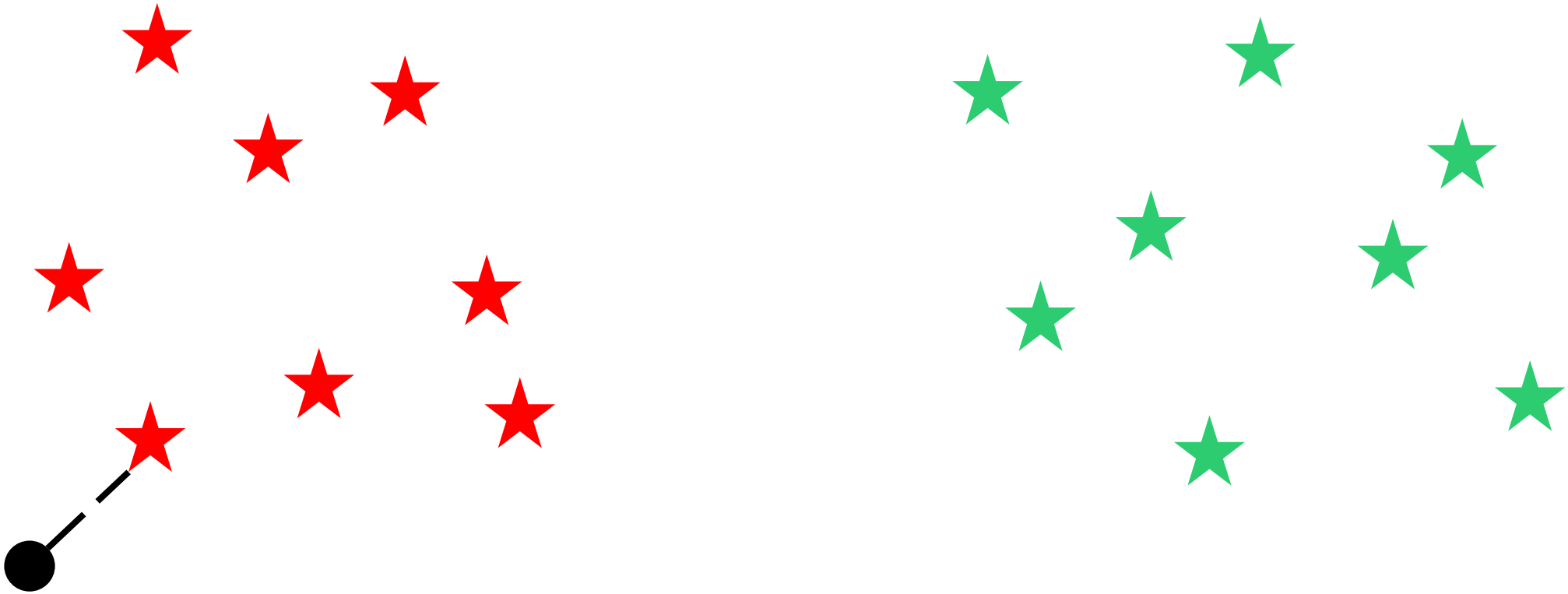
Learning with the 1-NN Classifier

Learning with the 1-NN Classifier

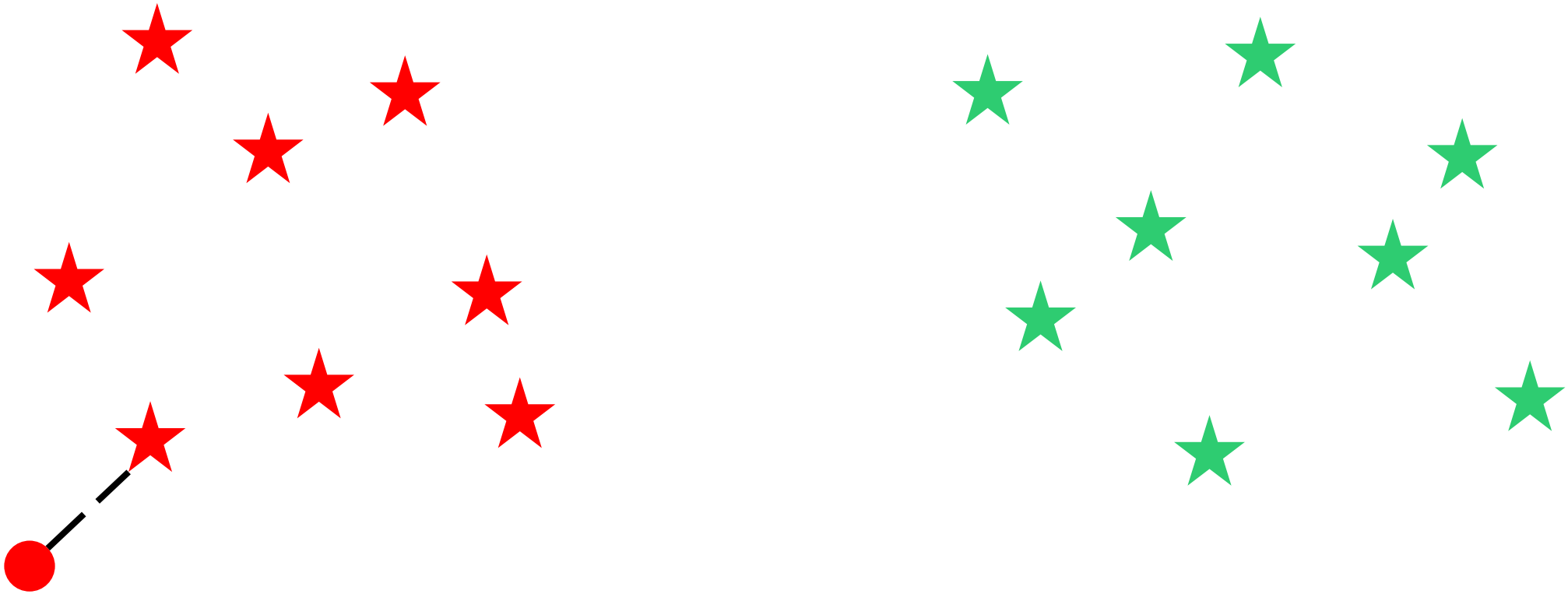
Learning with the 1-NN Classifier



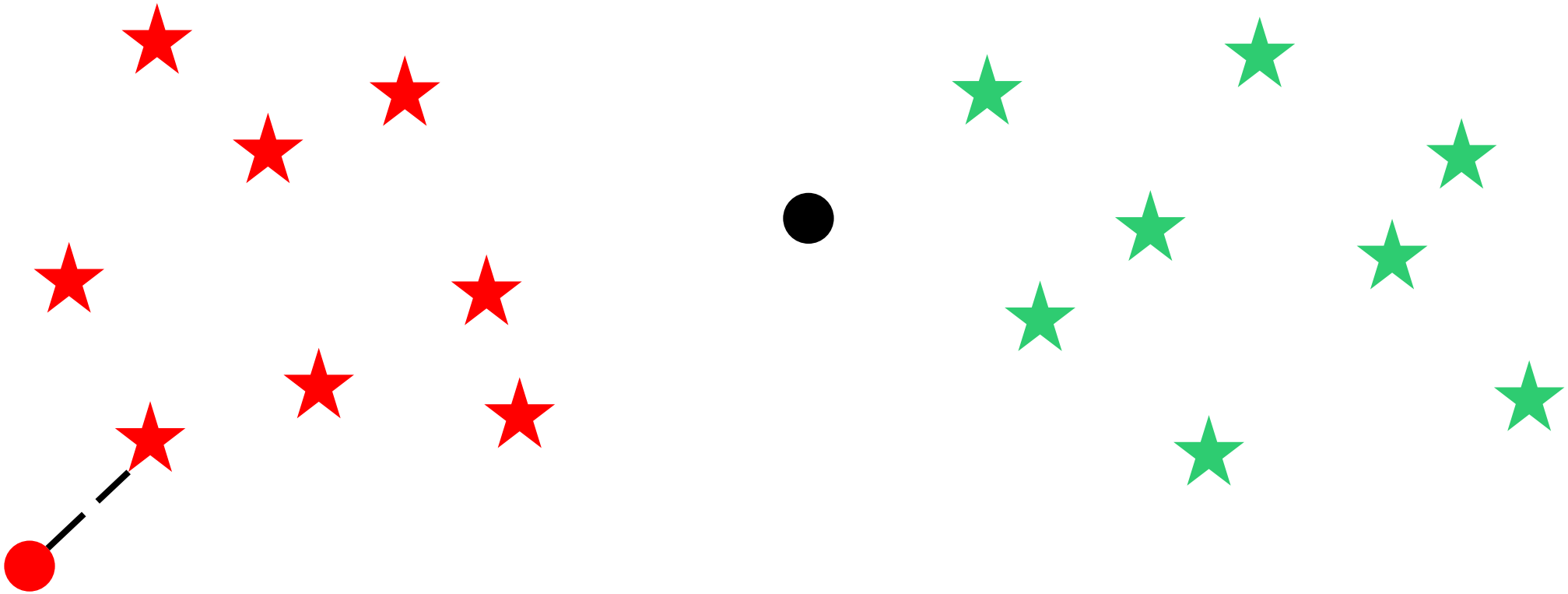
Learning with the 1-NN Classifier



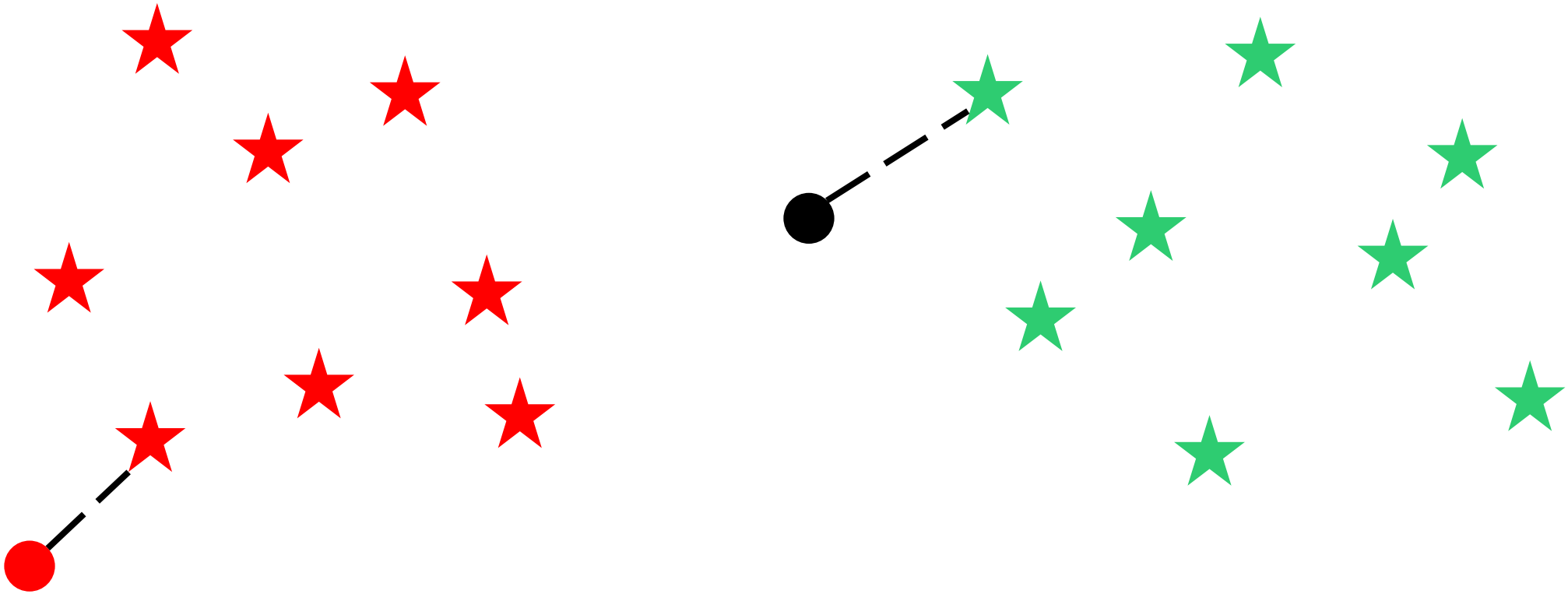
Learning with the 1-NN Classifier



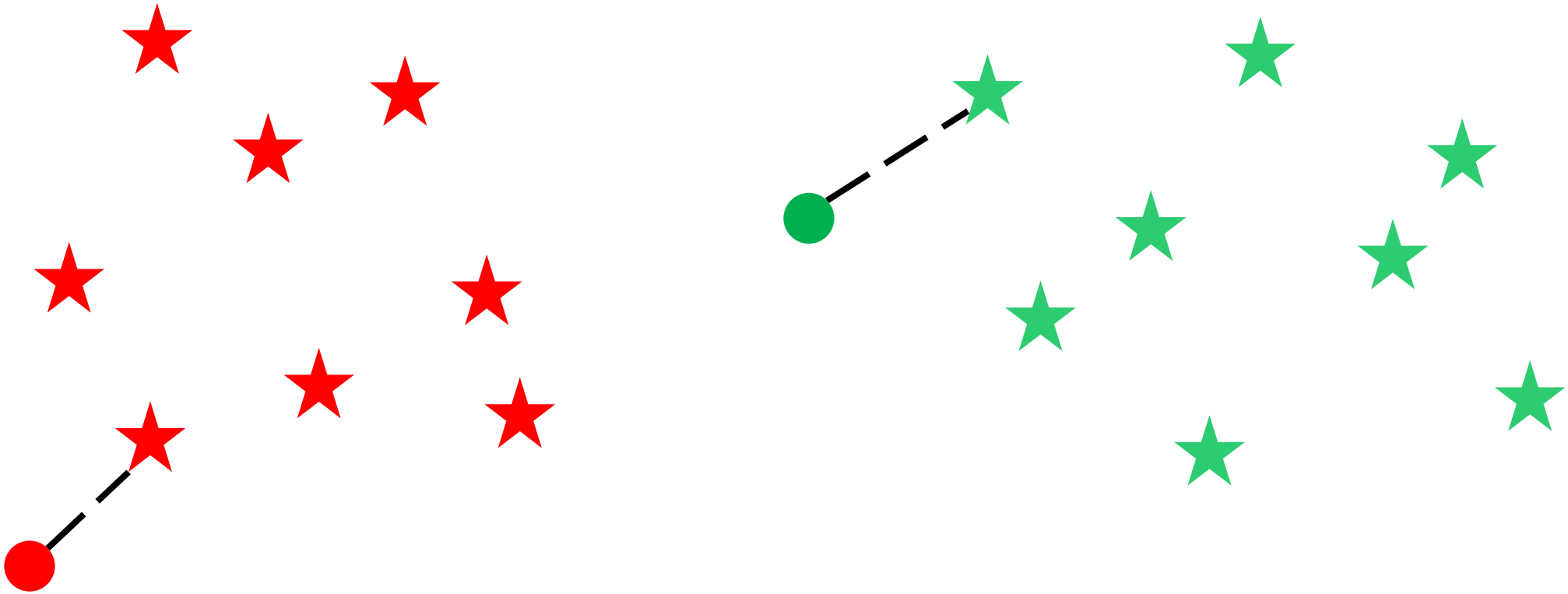
Learning with the 1-NN Classifier



Learning with the 1-NN Classifier

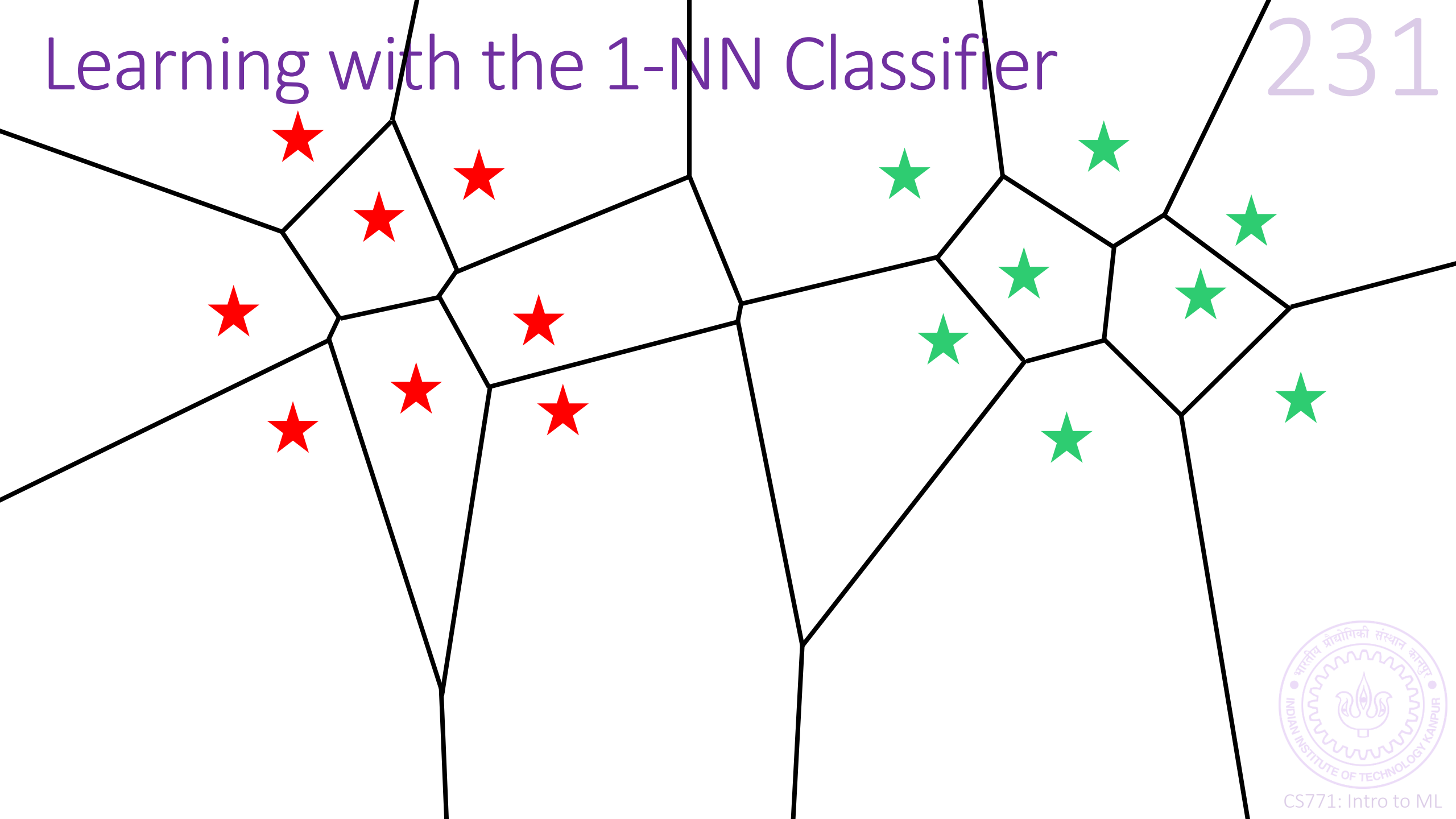


Learning with the 1-NN Classifier

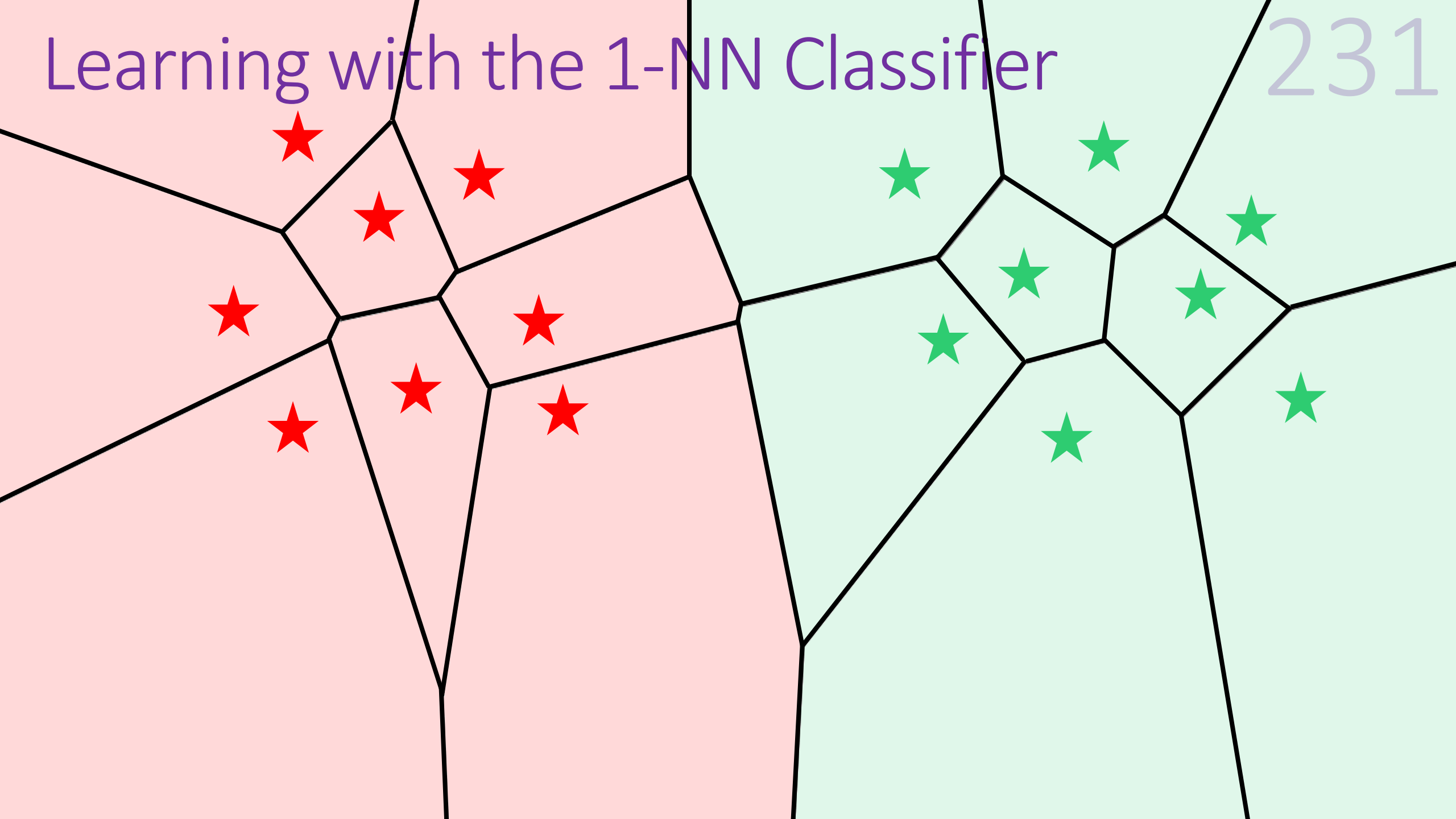


Learning with the 1-NN Classifier

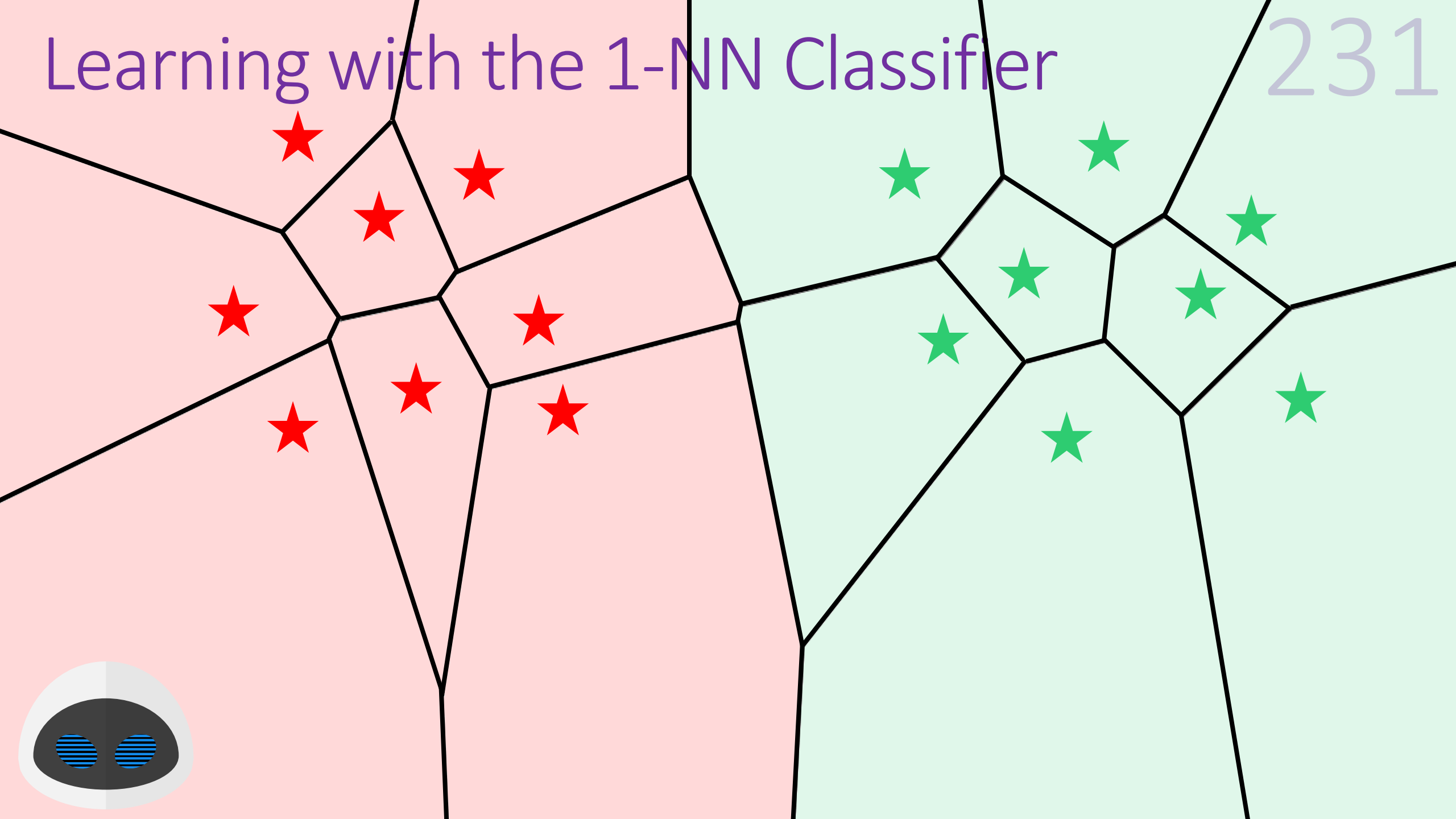
Learning with the 1-NN Classifier



Learning with the 1-NN Classifier



Learning with the 1-NN Classifier



Learning with the 1-NN Classifier