EEP 596: Adv Intro ML || Lecture 5

Dr. Karthik Mohan

Univ. of Washington, Seattle

January 17, 2023

Logistics

- **Conceptual 1** due Saturday, Jan 21

Logistics

- **Conceptual 1** due Saturday, Jan 21
- **Assignment 2** due Sunday, Jan 22

Logistics

- **Conceptual 1** due Saturday, Jan 21
- **Assignment 2** due Sunday, Jan 22
- **Kaggle competition:** Team by yourself for the Kaggle Contest. Basic NLP pre-processing code provided as part of the assignment. Additional feature engineering has been shared in quiz section + today as well.

Logistics

- **Conceptual 1** due Saturday, Jan 21
- **Assignment 2** due Sunday, Jan 22
- **Kaggle competition:** Team by yourself for the Kaggle Contest. Basic NLP pre-processing code provided as part of the assignment. Additional feature engineering has been shared in quiz section + today as well.
- **Lecture 6 next Monday (make-up lecture):** We won't have lecture this Thursday. Instead, we will have make up lecture (Lecture 6) next Monday at 4 pm pst.

Next week:-

online In-person

M, T, Th

4 PM

Logistics

- **Conceptual 1** due Saturday, Jan 21
- **Assignment 2** due Sunday, Jan 22
- **Kaggle competition:** Team by yourself for the Kaggle Contest. Basic NLP pre-processing code provided as part of the assignment. Additional feature engineering has been shared in quiz section + today as well.
- **Lecture 6 next Monday (make-up lecture):** We won't have lecture this Thursday. Instead, we will have make up lecture (Lecture 6) next Monday at 4 pm pst.
- **Manage Job Anxiety!** Impactful De-stress sessions

Last time

- More on regularization ℓ_1/ℓ_2
- All about GD, SGD, mini-batch SGD

1) Engine behind "training"/ optimization modern ML models

Mathematical

2) ML = optimization over training data

Today's class

- Logistic Regression
- Evaluation Metrics
- Feature engineering + NLP specific features

classification

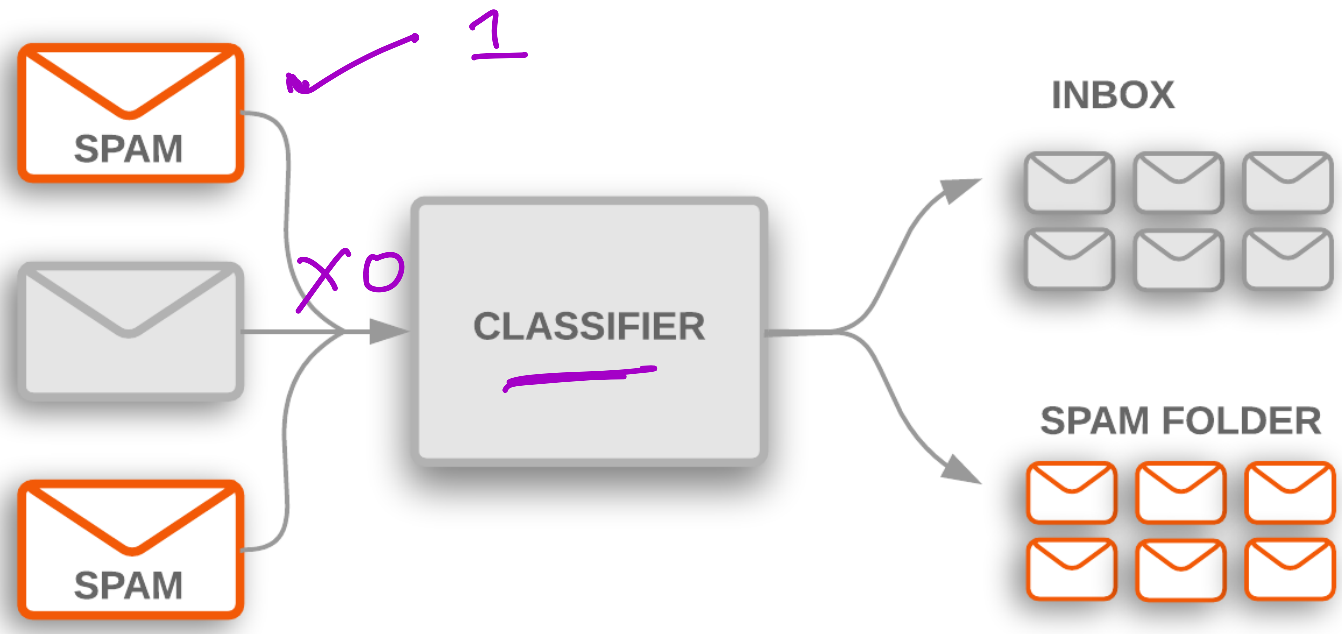
Course Outline

Week	Lecture Material	Assignment
1	Linear Regression	Housing Price Prediction
2	**Classification**	**Spam classification (Kaggle)**
3	Classification	Flower/Leaf classification
4	Clustering	MNIST digits clustering
5	Anomaly Detection	Crypto Prediction (Kaggle + P)
6	Data Visualization	Crypto Prediction (Kaggle + P)
7	Deep Learning	Visualizing 1000 images
8	Deep Learning (DL)	ECG Arrythmia Detection
9	DL in NLP	TwitterSentiment Analysis (Kaggle + P)
10	DLs in Vision	TwitterSentiment Analysis (Kaggle + P)

Classification in Machine Learning

$y_i = 1$

$y_i = 0$



"Phising"

Difference between Classification and Regression

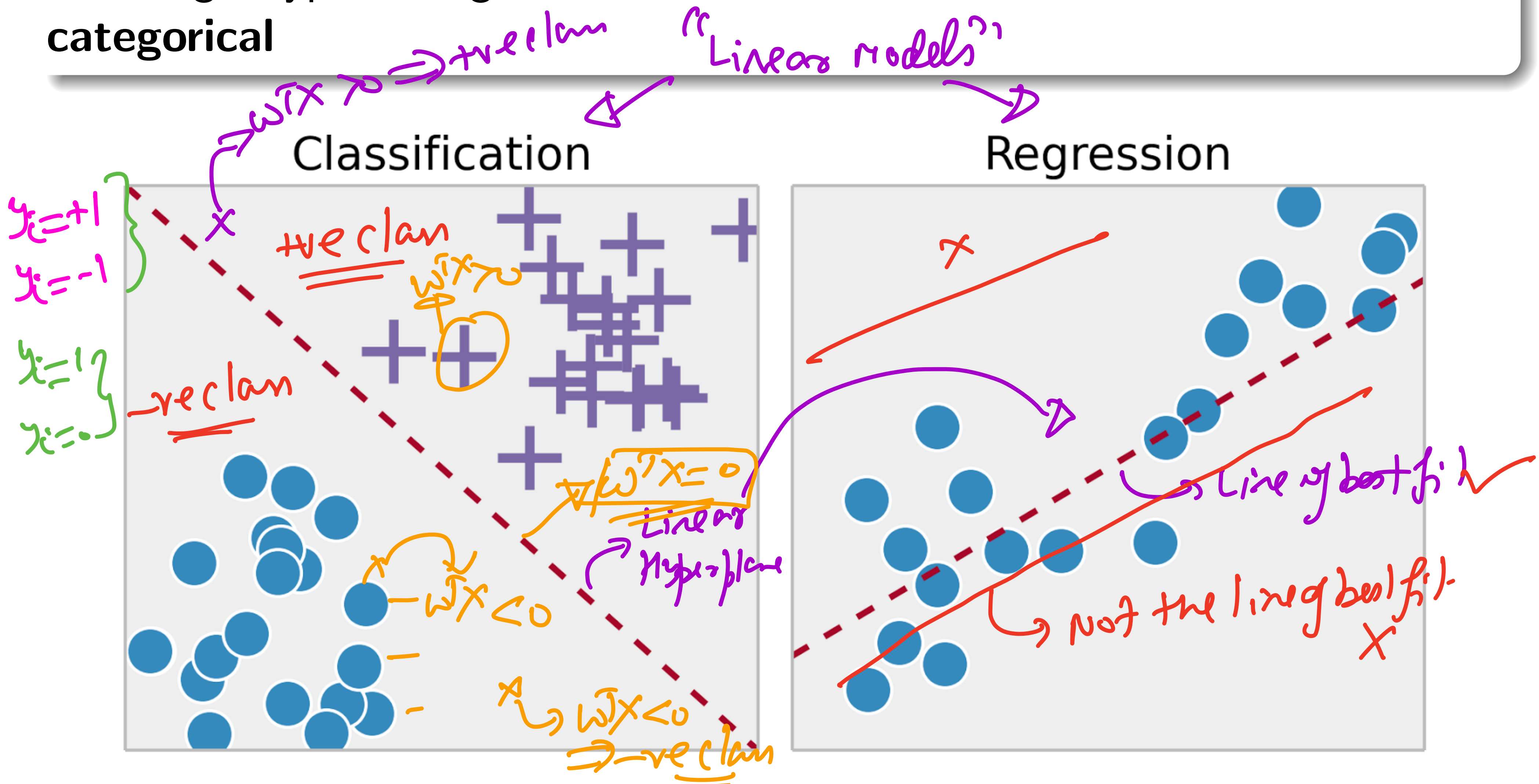
Simple difference

The target type in Regression is **numeric** whereas that in classification is **categorical** $\rightarrow y_i$

Difference between Classification and Regression

Simple difference

The target type in Regression is **numeric** whereas that in classification is **categorical**



Binary vs Multi-class classification

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

Binary → spam
Binary → not spam

Multi-class → Rose, Jasmine, Lily, Lotus

0, 1, 2, 3, ..., K

Image of a flower

	Prob.
Rose	0.6
Jasmine	0.3
Lotus	0.1

Types of Classification

Binary vs Multi-class classification

With binary categories, its a binary classification problem and with multiple categories, we have a multi-class classification.

Target is called Label

For binary classification, the convention is to label the target as positive or negative. Example: Positive for spam and negative for not-spam

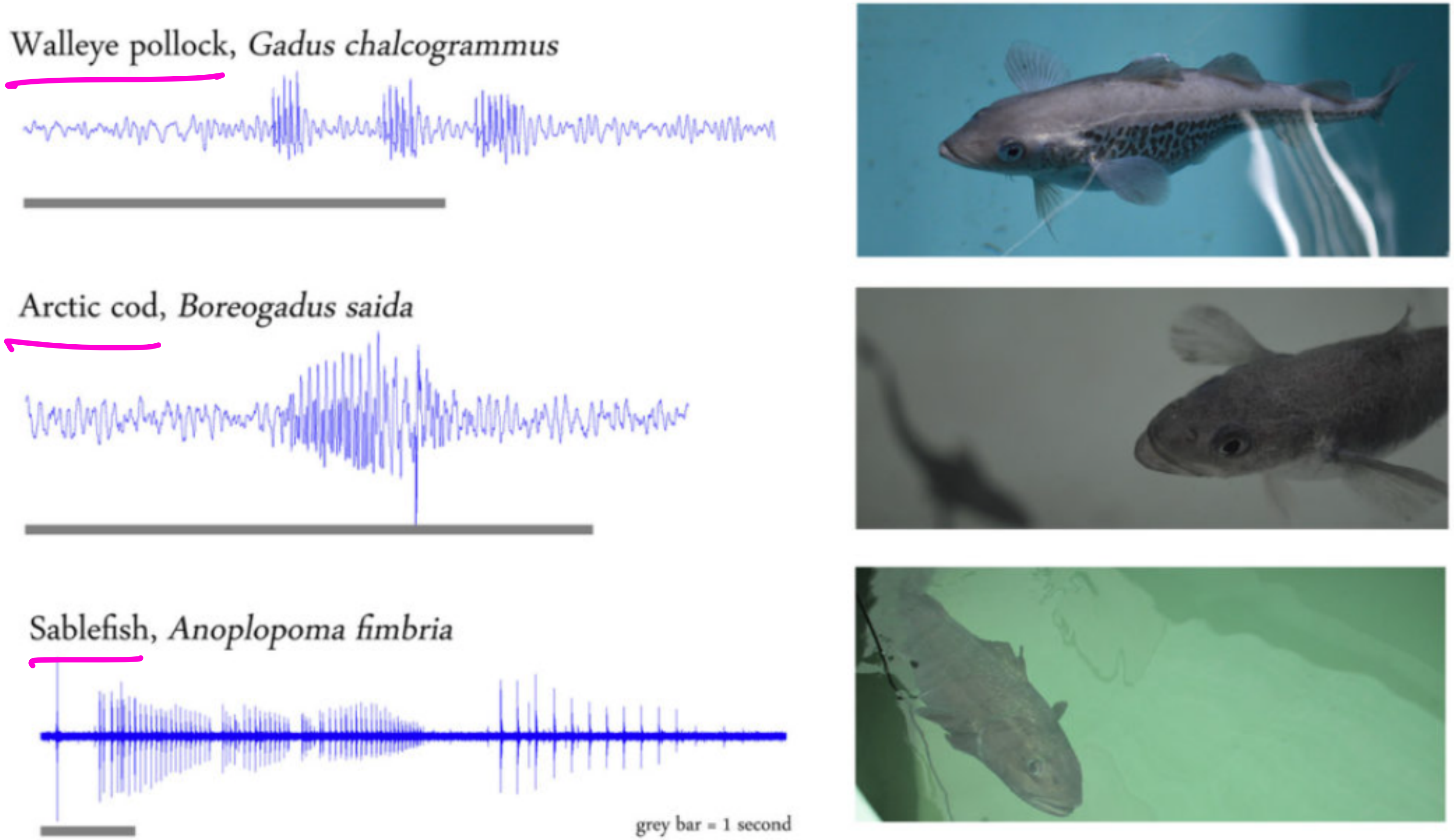
Spam Classification Example

Ground truth

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	+1
Looking forward to your presentation...	Not-spam	-1
...	...	...

Fishes Classification from hydrophone data

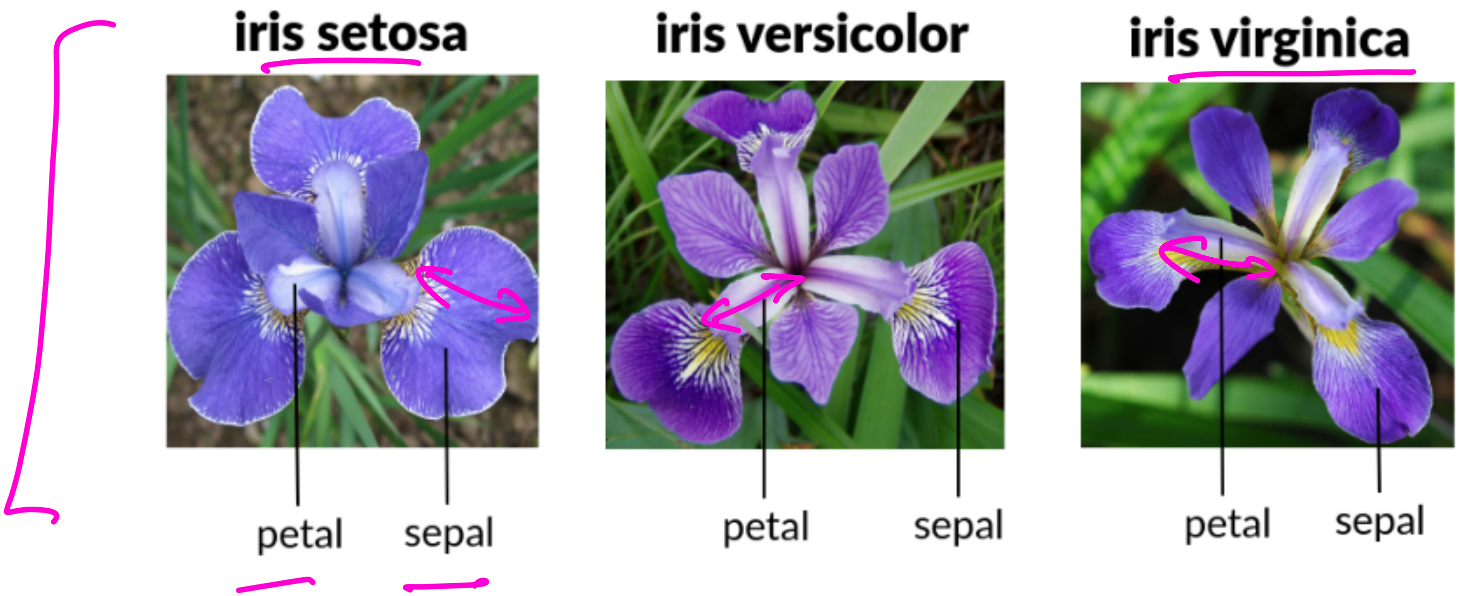
Time-Series Signals



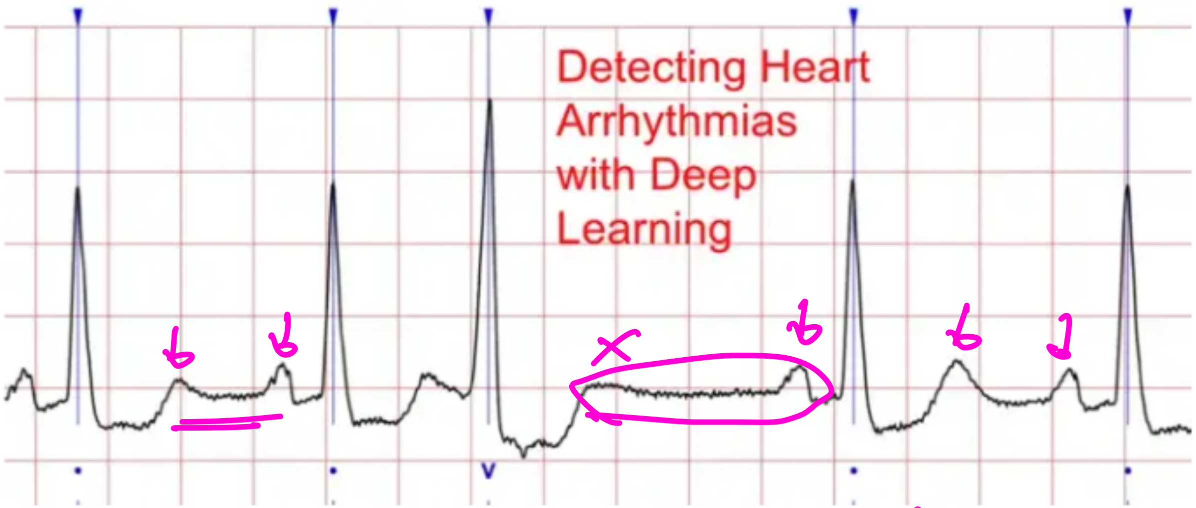
Waveforms representing characteristic sounds made by Walleye pollock, Arctic cod and Sablefish

Fish Detection Reference

Flower classification



Arryhthymia Detection



Time-Series Signal

↳ Pre-Emptive Diagnostics with IoT devices

Fraud Investigation

Shopping Site

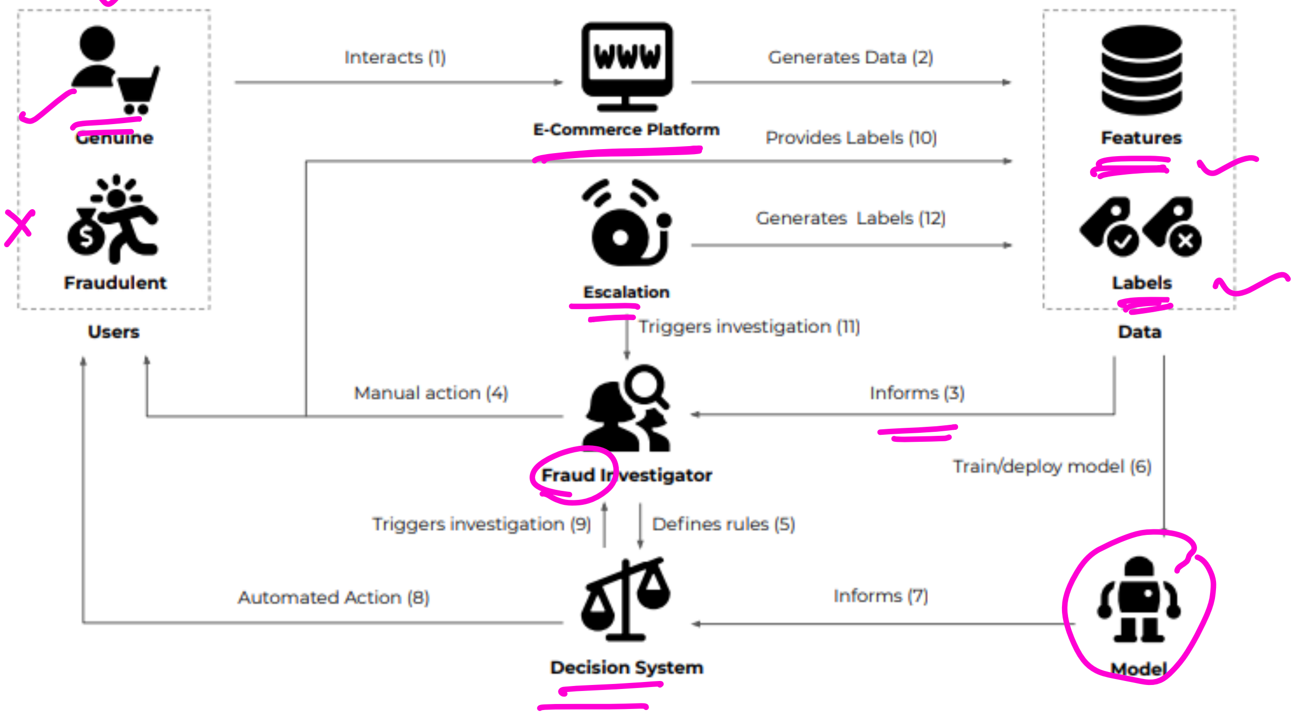


Fig. 1. A model of the daily operations of an anti-fraud department in an e-commerce organization.

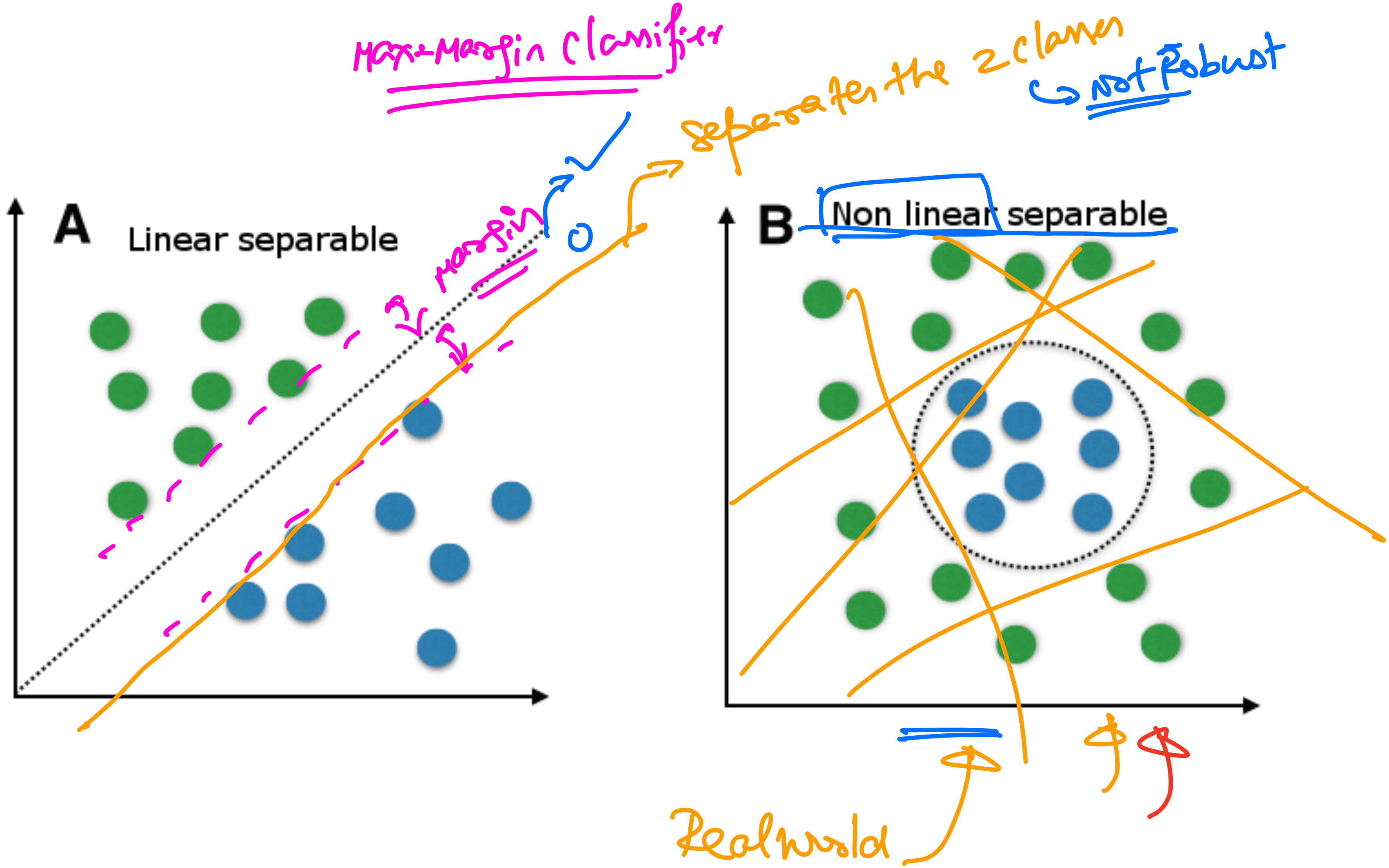
Machine Learning for Fraud Detection in E-Commerce: A research agenda

(Optional) Breakout #1

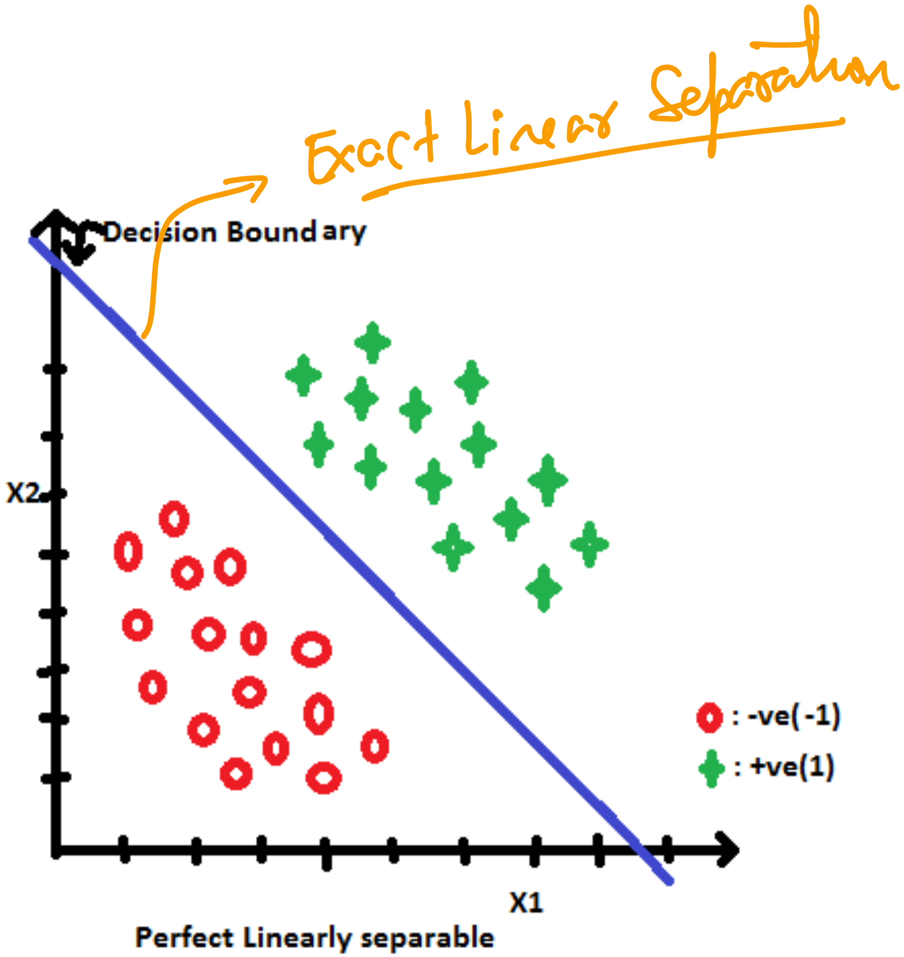
Fraud Detection Deep-Dive (5 mins)

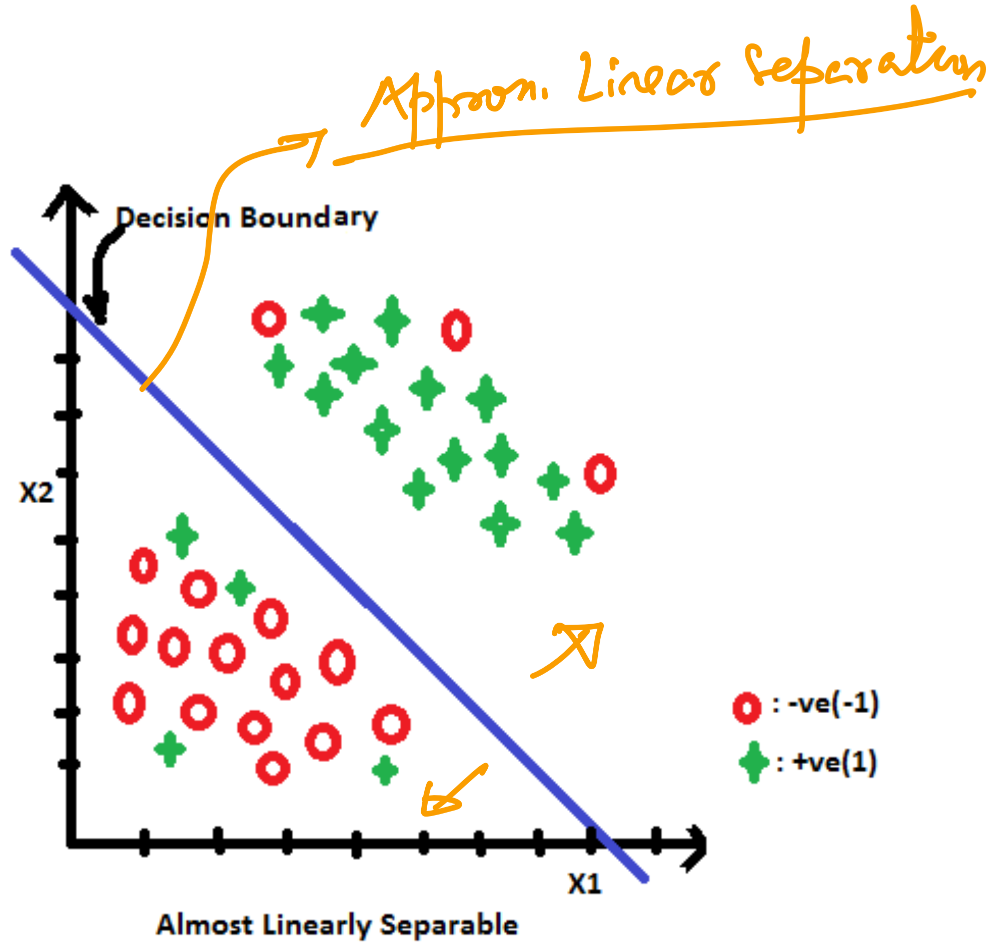
You go to a BullsEye store in Bothell and your credit card gets declined. The store manager shows up and says their system isn't accepting visa cards as of now. You suspect this might be related to some fraudulent transaction in the past. But you don't believe your credit card has been misused. What could be the cause of this? You try another visa card and decide to make a much smaller purchase and this time the transaction goes through. The store manager seems surprised but says he doesn't fully understand how it works and that many customers are having issues today. You visit another BullsEye store in Bellevue in the evening and make a big purchase with the first credit card and have no issues. If the accept/declines on credit card transactions were being triggered by an autaomated Machine Learning based models - What possible features in your transactions were triggering the accepts/declines?

Linear Separability

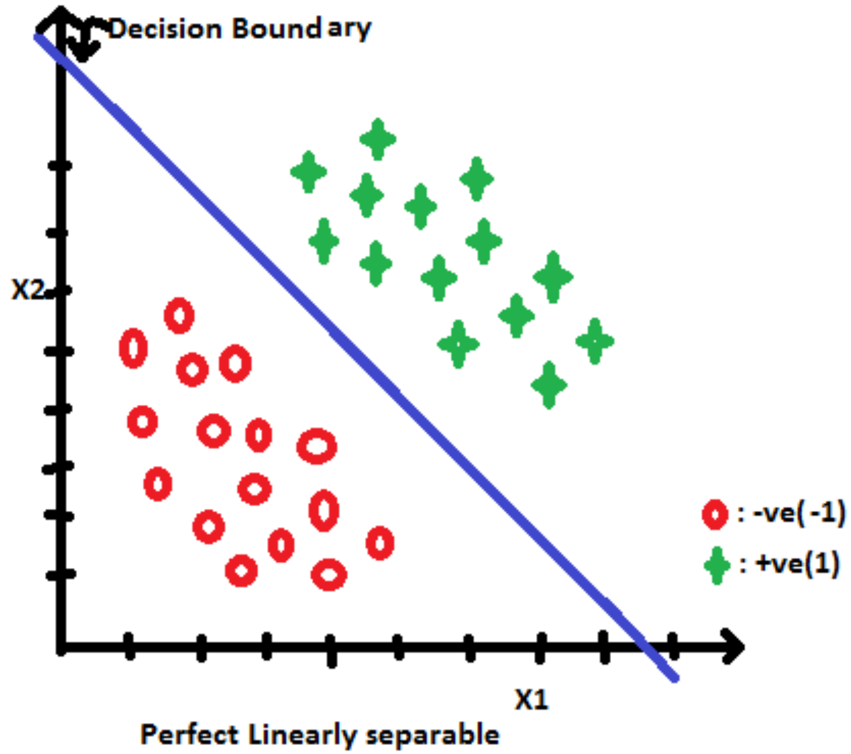


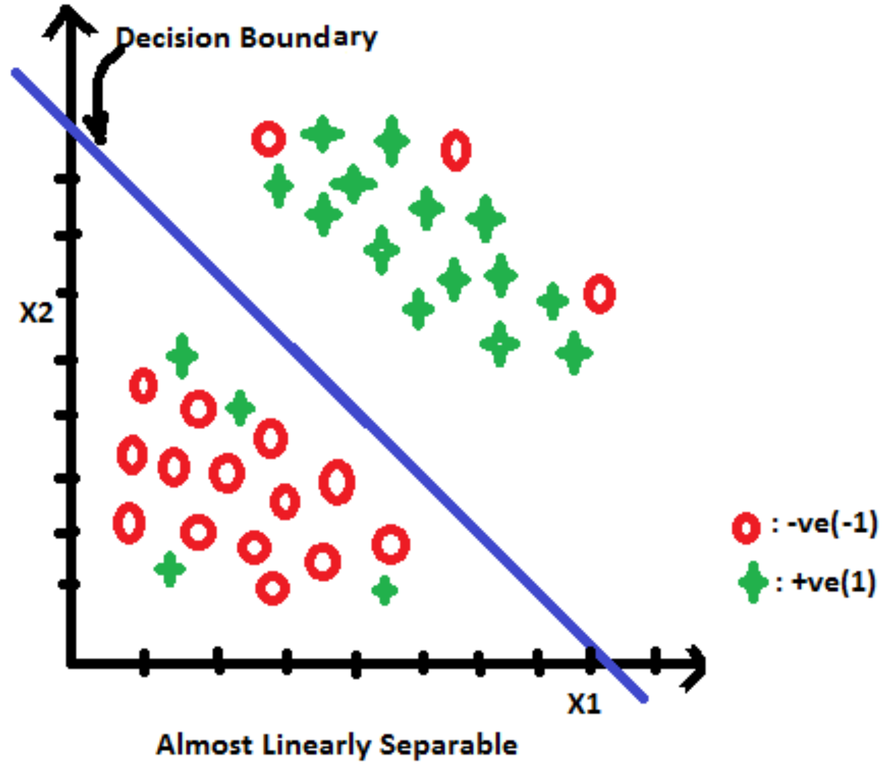
Approximate Linear Separability





Approximate Linear Separability





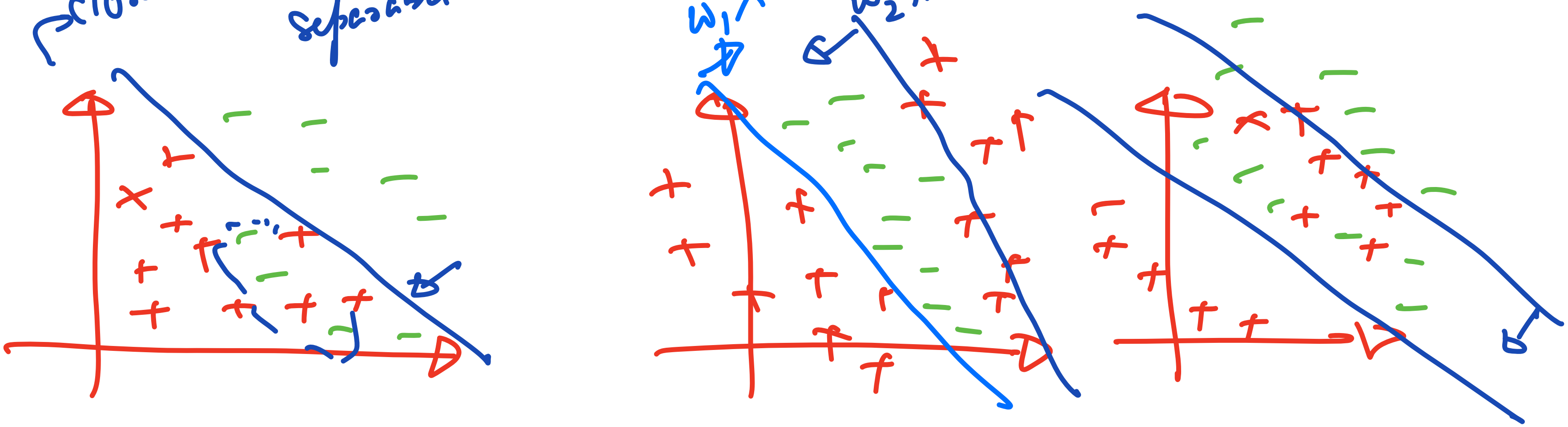
ICE #1

Decision trees

Which of the following data sets is the closest to being linearly separable?

closest to linearly separable

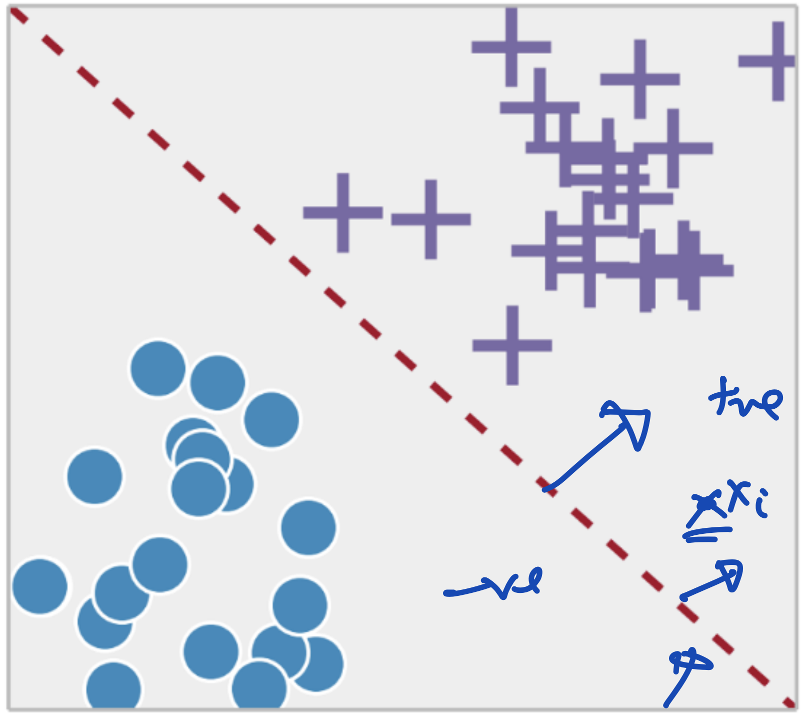
$w_1^T x$ $w_2^T x$



a) b) c)

↳ Multi-linear separable

Logistic Regression



$w^T x_i > 0 \Rightarrow \text{Score}(x_i) > 0$

Linear Hyperplane

LR fundamentals

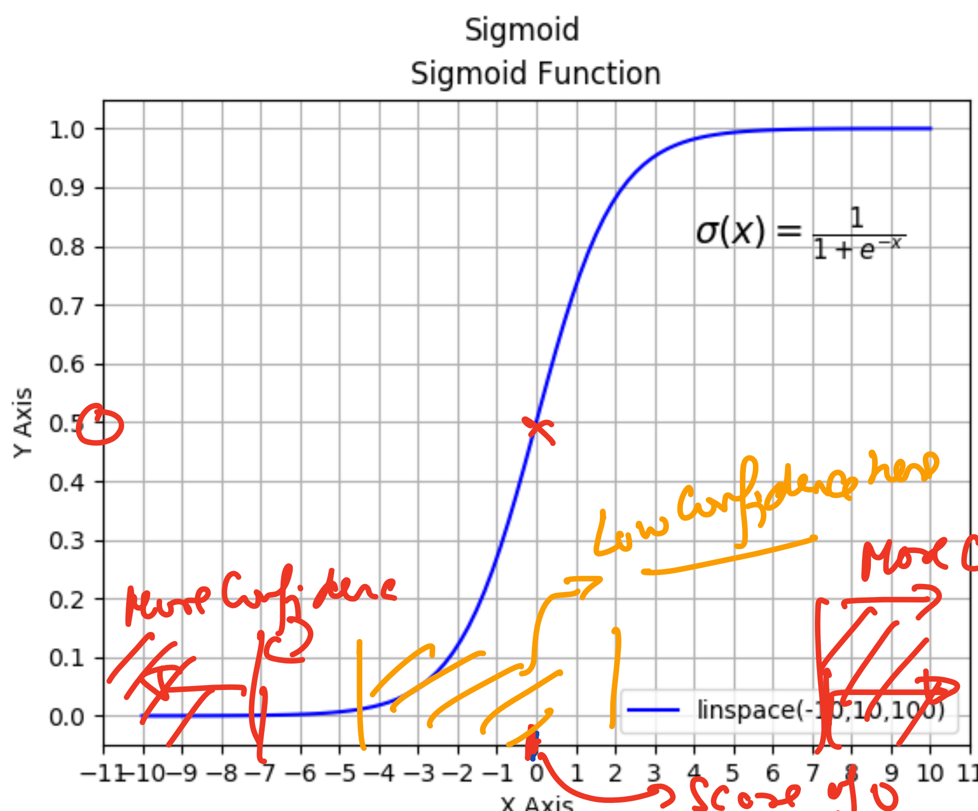
- Linear Model
- Want score $w^T x^i$ (Score) > 0 for $y_i = +1$ and $w^T x_i < 0$ for $y_i = -1$!
- If linearly separable data, above is feasible. Else, minimize error in separability!!

Logistic Regression

Probability for a class

In LR, the score, $w^T x$ is converted to a probability through the sigmoid function. So we can talk about $P(\hat{y}^i = +1)$ or $P(\hat{y}^i = -1)$

Sigmoid Function



S shaped fn.

Score to a Probability

Sigmoid / Prob. fn.
allows an infinite range of Scores (w^TX) to fall between 0 & 1!

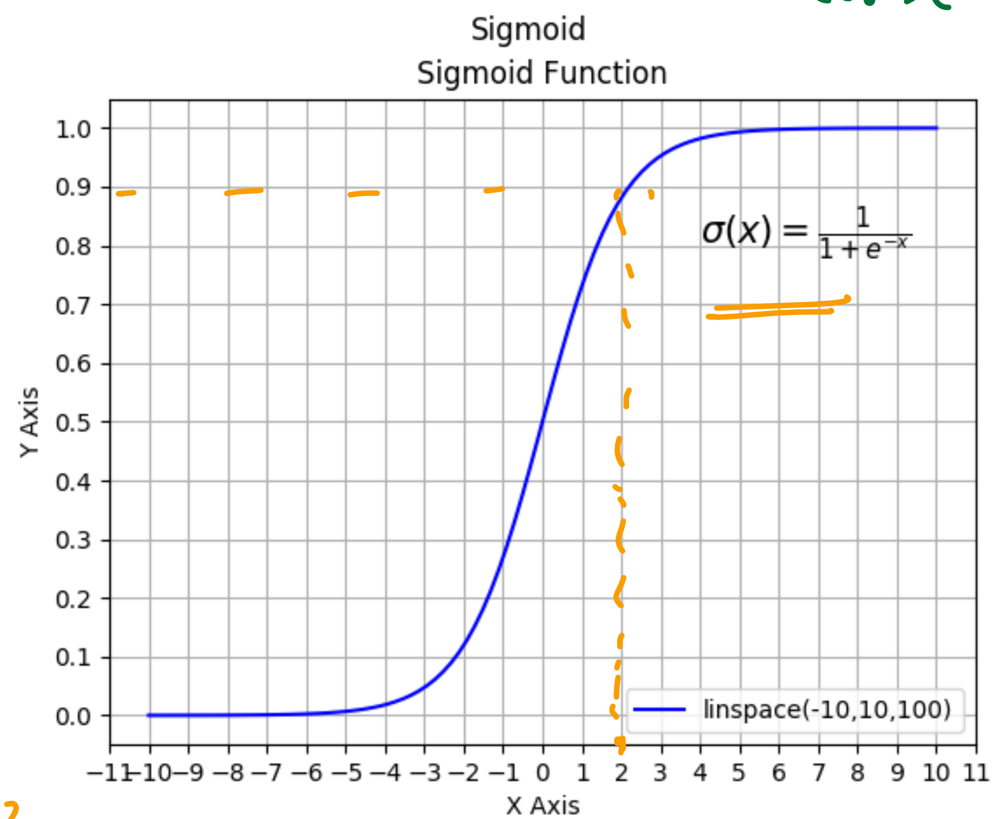
Sigmoid Function

$w^TX = 2$

$\sigma(w^TX) = \frac{1}{1+e^{-2}}$

$= 0.9$

$P(y_i=+1) = \frac{1}{1+e^{-2}} \leftarrow w^TX = 2$



$\sigma(x) = \frac{1}{1+e^{-x}}$

$\uparrow w^TX$

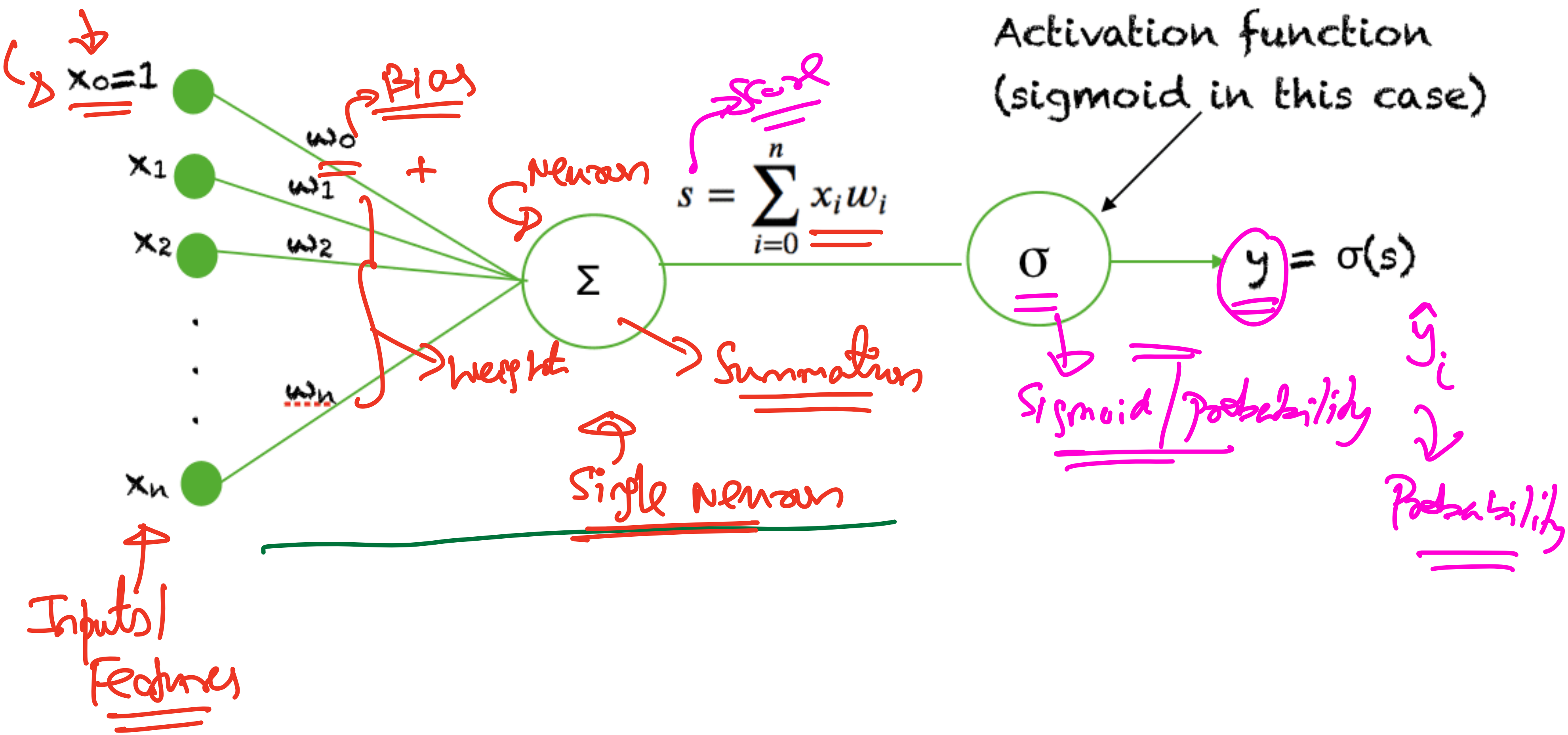
LR represented Graphically

$w^T x = -5$

$w^T x + w_0 \cdot 1 = 0$

$w_0 = 5$

Neuron in brain fn.



Logistic Regression

Probability is the prediction!

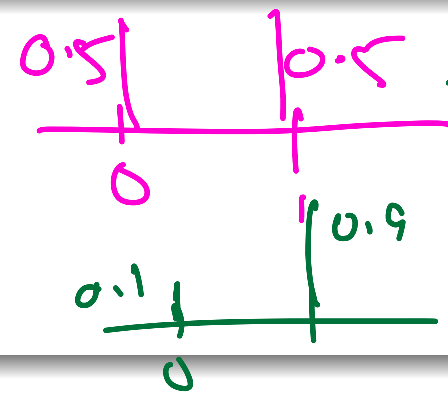
LR Prediction

$$\hat{y}_i = \frac{1}{1 + e^{-\hat{w}^T x^i}}$$

Entropy

$$H(p) = -\sum_i p_i \log(p_i)$$

uniform distribution max entropy



Cross-Entropy

$$H(p, q) = -\sum_i p_i \log(q_i)$$

y_i	$\hat{y}_i$
+1	0.9
−1	0.8
+1	0.2
−1	0.1

$q(y_i=+1)=0.1$

$q_2(y_i=-1)=0.9$

$p(y_i=+1)=1, \; p(y_i=-1)=0$

$q(\hat{y}_i=+1)=0.9 \quad q(\hat{y}_i=-1)=0.1$

$p \qquad p(y_i=+1)=p, \quad p(y_i=-1)=1-p$

$q \qquad q(y_i=+1)=q, \quad q(y_i=-1)=1-q$

Cross entropy $= -p\log q - (1-p)\log(1-q)$

$\min_q$ Cross entropy

$$= \min_q -p\log q - (1-p)\log(1-q)$$

$$\frac{\partial}{\partial q}(\ ) = 0$$

$$\Rightarrow \frac{-p}{q} + \frac{(1-p)}{(1-q)} = 0$$

$$\Rightarrow p(1-q) = q(1-p)$$

$$\Rightarrow p - \cancel{pq} = q - \cancel{qp}$$

$$\Rightarrow \underline{\underline{p=q}}\ !!$$

$\Rightarrow$ Minimising cross-entropy helps you fit to the true distribution better!

Quadratic Loss vs Cross-Entropy loss

Quadratic Loss

$$L(w) = \sum_i (y_i - \hat{y}_i)^2 = \|y - \hat{y}\|_2^2$$

Cross-Entropy Loss

Instead of a numeric prediction, we have a probability. What's a good way to say the prediction probability be close to true probability/class labels?

$$L(w) = -\sum_i y_i \log(\hat{y}_i) - (1 - y_i)\log(1 - \hat{y}_i)$$

Let $p = y_i, q = \hat{y}_i$. Then p, q are probability distributions!! So the loss-function is the sum of cross-entropies over the data points!

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.

For a good performance

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$

Line of Best Fit

Probability

weights

Separating Hyperplane

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!

Summary on Logistic Regression

1. Uses a linear model just like Linear Regression.
2. Assumes linear separability or approximate linear separability.
3. For linear regression, $\hat{y}_i = \hat{w}^T x^i$. For LR, $\hat{y}_i = \frac{1}{1+e^{-\hat{w}^T x^i}}$
4. Linear regression predicts numeric values that can range in $(-\infty, \infty)$. Logistic Regression predicts a probability of a class that ranges between $[0, 1]$.
5. Logistic Regression uses the Sigmoid or S-shaped function to go from a score to a probability!
6. Logistic Regression uses the log-loss or cross-entropy loss whereas Linear Regression uses the quadratic loss

Evaluating Classifiers!

ICE #2

Let's say you own an email server and want to provide a service to your email customers to help sort their emails into spam vs not-spam. So you go ahead and build a spam classifier on a training data set. Your data set has 100 spam emails and 900 non-spam emails. You notice that your classifier has 90% accuracy on the training data set and also your validation data set. Should you be happy with your classifier?

a) Yes
b) No
c) Maybe!
d) Something's fishy!

Evaluating classifiers

Class imbalance

The above data set is an example of class imbalance. What can go wrong here?

Just classify "everything" as non-spam!
$\Rightarrow$ 98% accuracy!!

Evaluating classifiers

Class imbalance

The above data set is an example of class imbalance. What can go wrong here?

Better metric than accuracy

Consider the **confusion matrix** for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives	0	100
Negatives	0	900

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives	0	100
Negatives	0	900

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives	0	100
Negatives	0	900

Better metric than accuracy

Accurcay is how many data points the classifier got right divided by the total data points. What's accuracy here?

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

Evaluating classifiers

Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

Accuracy, Precision, Recall and F1-score

	Predicted Positive	Predicted Negatives
Positives (P)	TP	FN
Negatives (N)	FP	TN

Evaluating classifiers

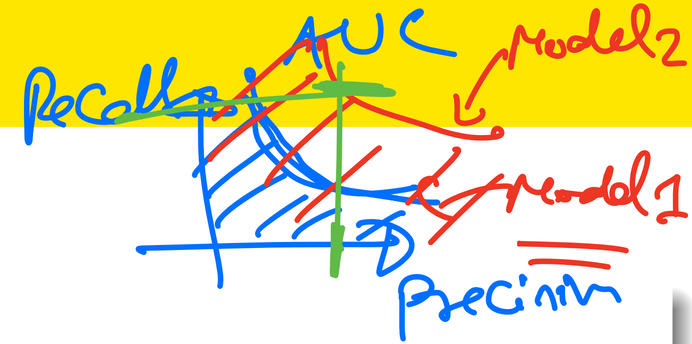
Better metric than accuracy

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

e

Evaluating classifiers



Better metric than accuracy

Precision

Consider the confusion matrix for above Spam classification example with the trivial classifier (predict everything as non-spam).

	Predicted Positive	Predicted Negatives
Positives (P)	0	100
Negatives (N)	0	900

e

Recall

Accuracy, Precision, Recall and F1-score

AUC ?

F1-score ?

AUC

Precision (Pr) = TP/(TP + FP) $= 0/0 = 1$

Recall (R) = TP/(TP + FN) = TP/P $= 0/100 = 0$

F1-score = $\frac{2\times Pr\times R}{Pr+R}$ = HM of Pr & R $= \frac{2\times1\times0}{1+0} = 0$

Accuracy (Acc) = $(TP + TN)/(P + N)$

ICE #3

More Confusion!

Let's say we computed a **Confusion Matrix** for another Spam Classifier on a different data set and we obtained:

	Predicted Positive	Predicted Negatives
Positives (P)	50	50
Negatives (R)	100	400

→ Recall

→ Precision

→ Accuracy

Metrics!

Accuracy, **Pr**, **R** and **F1** are as follows:

a) $75\%, 0.2, 0.5, 0.285$

b) $80\%, 0.3, 0.4, 0.285$

c) $80\%, 0.5, 0.3, 0.1875$

d) $75\%, 0.3, 0.5, 0.1875$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Strategies to handle class imbalance

+ve

-ve

}class Imbalance

1. Metrics:- use AUC & F1-Score instead of ACC.
2. Model Training:- Can upsample the weaker class & downsample the dominant class
3. Data Augmentation for the weaker class.

Features for Text Data

Email Excerpt

Congratulations! You have been selected for our special offer.

Features for Text Data

Email Excerpt

Congratulations! You have been selected for our special offer.

String to Features

How do we convert strings of text to features that we can use?

Features for Text Data

Email Excerpt

Congratulations! You have been selected for our special offer.

String to Features

How do we convert strings of text to features that we can use?

Bag of words Model

Represent every string in terms of a long vector of words (e.g. every possible word in vocabulary). However, only a few elements of those words are non-zero. So its a 'sparse' vector representation!

Features for Text Data

Bag of words Example

Sentence 1: This is a simple sentence
Sentence 2: How about this one
Sentence 3: One more

Features for spam classification

Data Pre-processing

The notebook for **Assignment 2** has some data pre-processing that can help you get started. E.g. removal of stop words like 'the' or 'a' that may not contribute to the prediction. Also removal of punctuation if it doesn't help, etc.

Features for spam classification

Data Pre-processing

The notebook for **Assignment 2** has some data pre-processing that can help you get started. E.g. removal of stop words like 'the' or 'a' that may not contribute to the prediction. Also removal of punctuation if it doesn't help, etc.

Example

Sentence: Congratulations!!! You have been selected for our special offer.
Sentence after pre-processing: Congratulations have been selected our special offer

Features for spam classification

Data Pre-processing

The notebook for **Assignment 2** has some data pre-processing that can help you get started. E.g. removal of stop words like 'the' or 'a' that may not contribute to the prediction. Also removal of punctuation if it doesn't help, etc.

Example

Sentence: Congratulations!!! You have been selected for our special offer.
Sentence after pre-processing: Congratulations have been selected our special offer

After pre-processing

After pre-processing of the sentence, bag of words can be used to vectorize each pre-processed sentence like on the previous slide!

Features for spam classification

Non-linear features

Some times combination of words can be more useful in making predictions. E.g. "Not bad" may indicate a positive sentiment while just "not" or "bad" indicates negative sentiment. A new feature, "not_bad" can capture this combination and is called a bi-gram.

Features for spam classification

Non-linear features

Some times combination of words can be more useful in making predictions. E.g. "Not bad" may indicate a positive sentiment while just "not" or "bad" indicates negative sentiment. A new feature, "not_bad" can capture this combination and is called a bi-gram.

N-grams

Combination of n-consecutive words can be a feature in the bag of words model. Usually N = 1,2,3 (uni, bi and tri-grams).

ICE #4

Bag of words

Let's say you have 1000 sentences in a document and you want to represent each sentence in the document with bag of words. There are a total of 6723 words in the document with 5000 unique words. What would be the dimension of the vector that represents each sentence using the bag of words model (assume uni-grams and no pre-processing of the sentence)?

a) 5000
b) 6723
c) 1000
d) 5723

ICE #4

Bag of words

Let's say you have 1000 sentences in a document and you want to represent each sentence in the document with bag of words. There are a total of 6723 words in the document with 5000 unique words. What would be the dimension of the vector that represents each sentence using the bag of words model (assume uni-grams and no pre-processing of the sentence)?

a) 5000
b) 6723
c) 1000
d) 5723

New words in the evaluation data set?

How do you deal with words you have not seen in training show up in test?

Scalable representations?

Learned feature representations

What if we have 1000's of documents, each with 1000 sentences and an average of 5 words per sentence! That gives us 5 MM words and let's say 300k unique words. Bag of words representation becomes memory intensive and also computationally cumbersome!
Enter: Low-dimensional 'learned' features. E.g. Glove Embedding/Word2Vec Embedding.

TF-IDF

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

TF-IDF for Spam Classification Example

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	$+1$
Looking forward to your presentation...	Not-spam	-1
$\ldots$	$\ldots$	$\ldots$

TF-IDF for Spam Classification Example

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	$+1$
Looking forward to your presentation...	Not-spam	-1
...	...	...

TF-IDF of you in Email 1

$\text{TF} = 1$

$\text{IDF} = \log(3/2) = \log(1.5)$

$\text{TF-IDF} = \log(1.5)$

TF-IDF for Spam Classification Example

Email excerpt	Type	Label
Could you please respond by tomorrow?	Not-spam	-1
Congratulations!!! You have been selected...	Spam	+1
Looking forward to your presentation...	Not-spam	-1
...	...	...

TF-IDF of `you` in Email 1

TF $= 1$
IDF $= \log(3/2) = \log(1.5)$
TF-IDF $= \log(1.5)$

TF-IDF of `presentation` in Email 3

TF $= 1$
IDF $= \log(3) = \log(3)$
TF-IDF $= \log(3)$

ICE #5

TF-IDF

Consider the following sentences:
S1: Are you in Seattle right now?
S2: What's the time right now?
S3: Right, that doesn't sound right
If we use a TF-IDF filter, which word is most likely to get eliminated from any of the sentences?

a. the
b. you
c. right
d. now

Summary

- Understanding Logistic Regression
- Evaluation metrics for classifiers
- Feature engineering for spam classification