CSE/STAT 416

Classification

Tanmay Shah
University of Washington
April 10, 2024

❓ **Questions?** Raise hand or **sli.do #cs416**
💬 **Before Class:** Does a straw have two holes or one?
🎵 **Listening to:** nothing, enjoy the calm

Administrivia

We have now finished the “Regression” component of the course!

Next two weeks (4 lectures): Classification

HW1 due tomorrow 11:59PM

- Up to Sat 4/13 11:59PM if you use late days

HW2 released Fri

Roadmap So Far

1. Housing Prices - Regression
   - Regression Model
   - Assessing Performance
   - Ridge Regression
   - LASSO
2. Sentiment Analysis – Classification
   - Classification Overview
   - Logistic Regression

Regression vs. Classification

Regression problems involve predicting **continuous values**.

- E.g., house price, student grade, population growth, etc.

Classification problems involve predicting **discrete labels**

- e.g., spam detection, object detection, loan approval, etc.

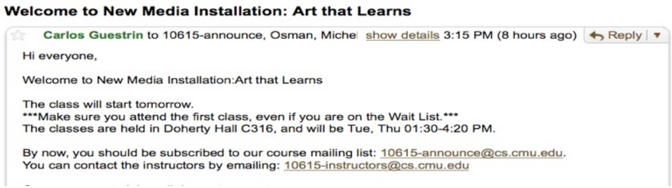
Spam Filtering


Osman Khan to Carlos show details Jan 7 (6 days ago) Reply

sounds good
+ok

Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos


Welcome to New Media Installation: Art that Learns

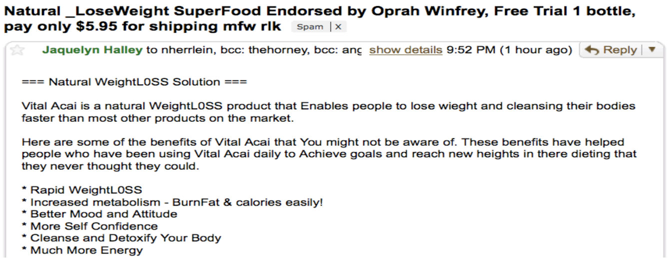
Carlos Guestrin to 10615-announce, Osman, Michel show details 3:15 PM (8 hours ago) Reply

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu


Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only $5.95 for shipping mfw rlk Spam | x

Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: ang show details 9:52 PM (1 hour ago) Reply

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

* Rapid WeightL0SS
* Increased metabolism - BurnFat & calories easily!
* Better Mood and Attitude
* More Self Confidence
* Cleanse and Detoxify Your Body
* Much More Energy

Output: y

Spam

Not Spam
(ham)

Input: x

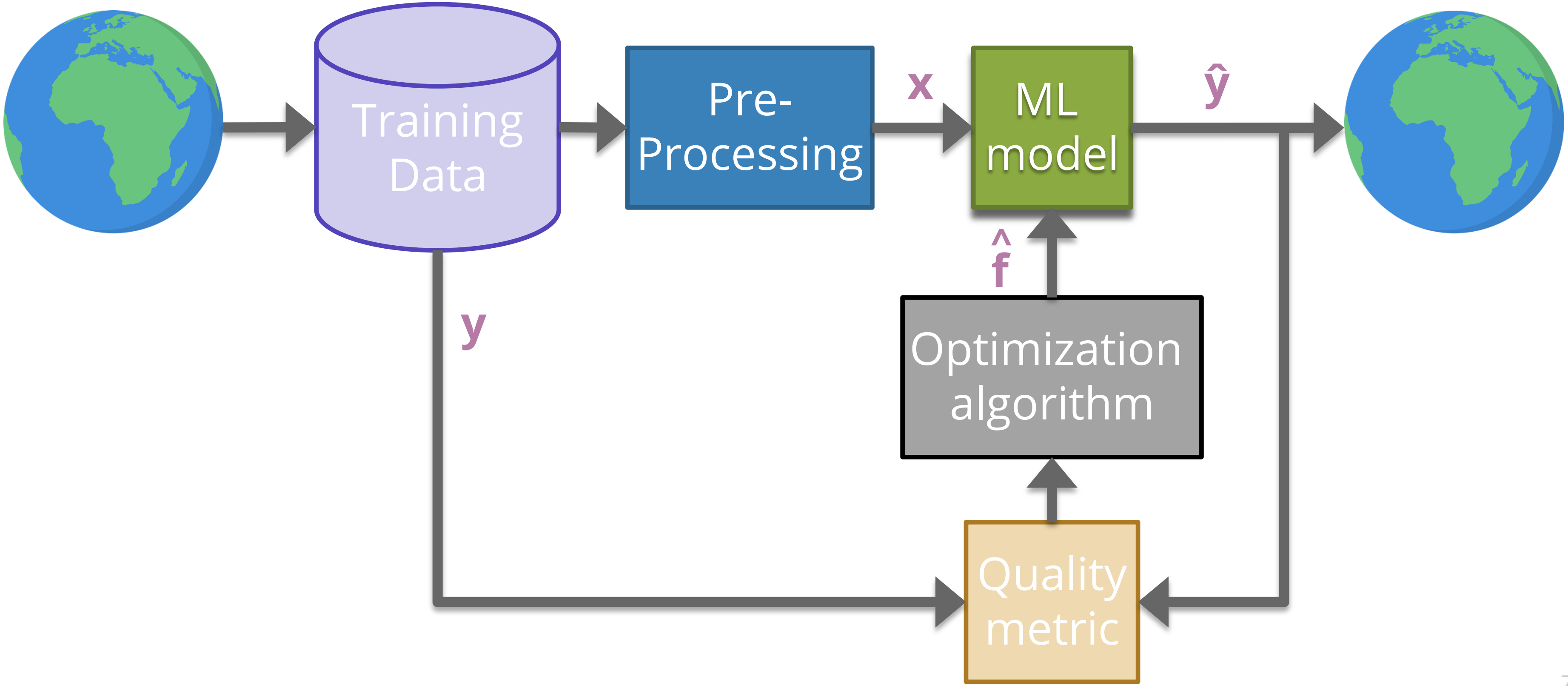
Text of email
Sender
Subject
…

Object Detection

Input: x
Pixels

Output: y
Class
(+ Probability)

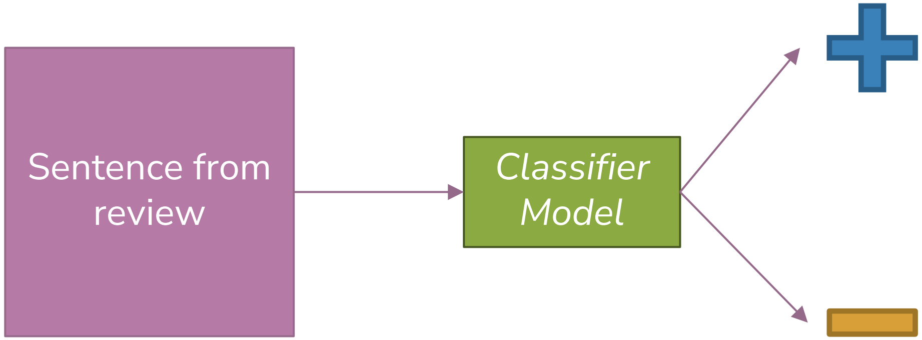
ML Pipeline



Training Data
Pre-Processing
x
ML model
ŷ
f̂
y
Optimization algorithm
Quality metric

Sentiment Classifier

In our example, we want to classify a restaurant review as positive or negative.



Input: x

Output: y
Predicted class

Converting Text to Numbers (Vectorizing):

Bag of Words

Idea: One feature per word!

Example: "Sushi was great, the food was awesome, but the service was terrible"

sushi	was	great	the	food	awesome	but	service	terrible

This **has** to be too simple, right?

Stay tuned (today and Wed) for issues that arise and how to address them ☺

Pre-Processing: Sample Dataset

Review	Sentiment
"Sushi was great, the food was awesome, but the service was terrible"	+1
…	…
"Terrible food; the sushi was rancid."	-1

Vectorizer

Sushi	was	great	the	food	awesome	but	service	terrible	rancid	Sentiment
1	3	1	2	1	1	1	1	1	0	+1
…	…	…	…	…	…	…	…	…	…	…
1	1	0	1	1	0	0	0	1	1	-1

How to Implement Sentiment Analysis?

Attempt 1: Simple Threshold Analysis

Attempt 2: Linear Classifier

Attempt 3 (Wed): Logistic Regression

Attempt 1: Simple Threshold Classifier

Idea: Use a list of good words and bad words, classify review by the most frequent type of word

Word	**Good?**
sushi	None
was	None
great	Good
the	None
food	None
but	None
awesome	Good
service	None
terrible	Bad
rancid	Bad

Simple Threshold Classifier

Input x: Sentence from review

Count the number of positive and negative words, in x

If $\text{num\_positive} > \text{num\_negative}$:

- $\hat{y} = +1$

Else:

- $\hat{y} = -1$

Example: "Sushi was great, the food was awesome, but the service was terrible"

Limitations of Attempt 1 (Simple Threshold Classifier)

Words have different degrees of sentiment.

- Awesome > Great
- How can we weigh them differently?

Single words are not enough sometimes...

- “Good” → Positive
- “Not Good” → Negative

How do we get list of positive/negative words?

Words Have Different Degrees of Sentiments

What if we generalize good/bad to a numeric weighting per word?

Word	Good?
sushi	None
was	None
great	Good
the	None
food	None
but	None
awesome	Good
service	None
terrible	Bad
rancid	Bad

Word	Weight
sushi	0
was	0
great	1
the	0
food	0
but	0
awesome	2
service	0
terrible	-1
rancid	-2

How do we get the word weights?

What if we learn them from the data?

$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$	$h_5(x)$	$h_6(x)$	$h_7(x)$	$h_8(x)$	$h_9(x)$
sushi	**was**	**great**	**the**	**food**	**awesome**	**but**	**service**	**terrible**
1	3	1	2	1	1	1	1	1

Word	**Weight**
sushi	w_1
was	w_2
great	w_3
the	w_4
food	w_5
awesome	w_6
but	w_7
service	w_8
terrible	w_9

In linear regression we learnt the weights for each feature. Can we do something similar here?

Attempt 2: Linear Classifier

Idea: Use labelled training data to learn a weight for each word. Use weights to score a sentence.

Model:

$$\hat{y}_i = sign\big(Score(x_i)\big) = sign(s_i)$$

$$= sign\left(\sum_{j=0}^{D} w_j h_j(x_i)\right) = sign(w^T h(x_i))$$

Word	Weight
sushi	0
was	0
great	1
the	0
food	0
awesome	2
but	0
service	0
terrible	-1

$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$	$h_5(x)$	$h_6(x)$	$h_7(x)$	$h_8(x)$	$h_9(x)$
sushi	**was**	**great**	**the**	**food**	**awesome**	**but**	**service**	**terrible**
1	3	1	2	1	1	1	1	1

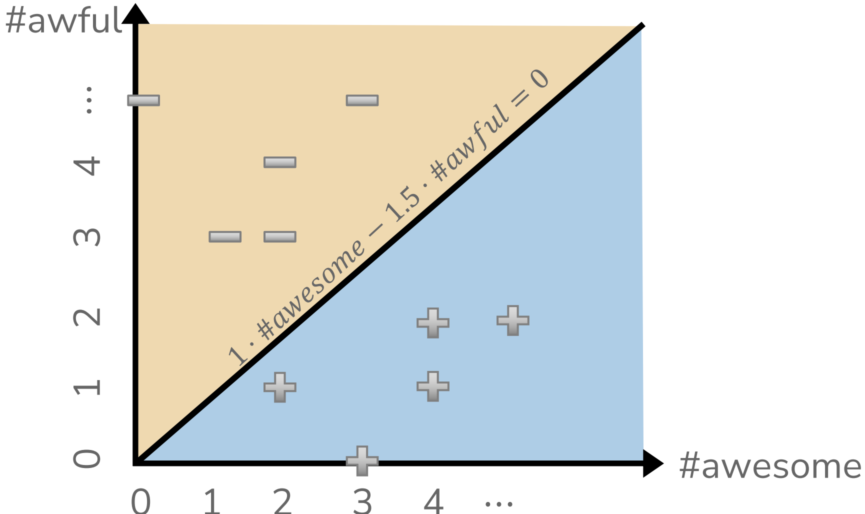
”Sushi was great, the food was awesome, but the service was terrible”

Decision Boundary

Consider if only two words had non-zero coefficients

Word	Coefficient	Weight
	w_0	0.0
awesome	w_1	1.0
awful	w_2	-1.5

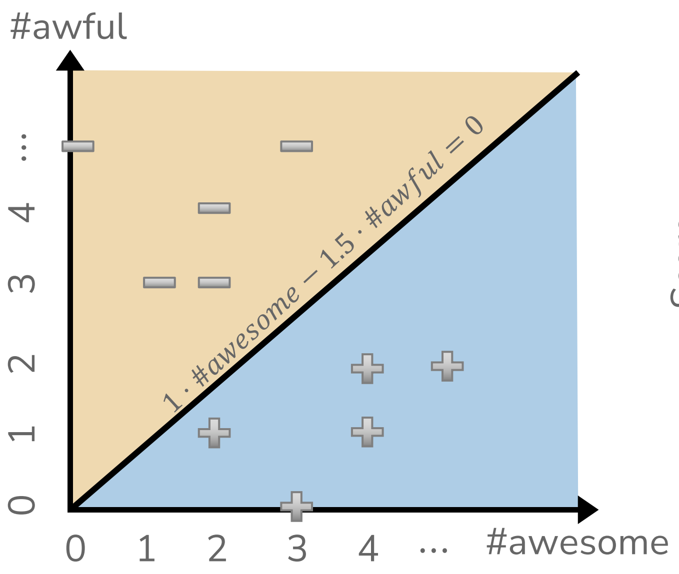
$$\hat{s} = 1 \cdot \#awesome - 1.5 \cdot \#awful$$



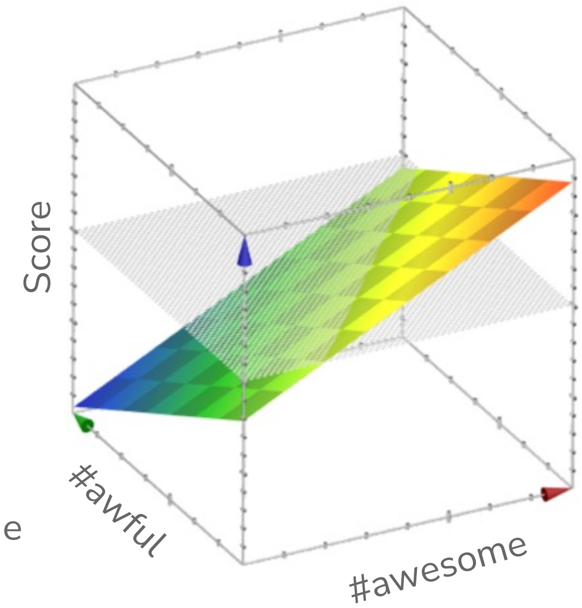
Decision Boundary

$$Score(x) = 1 \cdot \#awesome - 1.5 \cdot \#awful$$

2-dimensional view



3-dimensional view

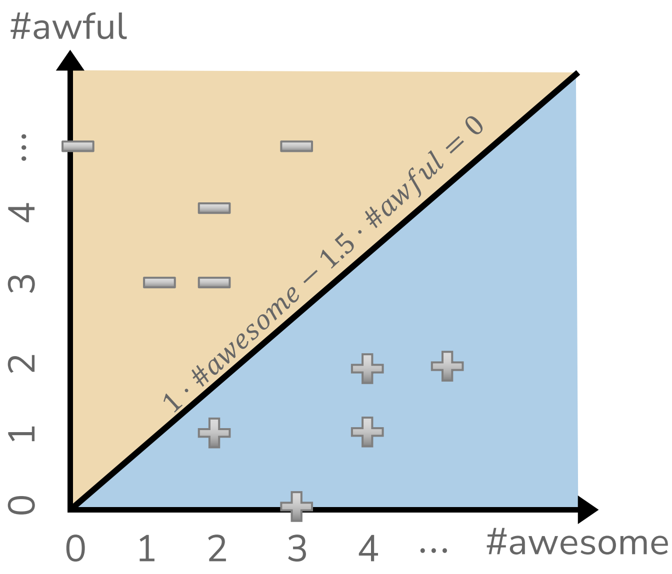


Generally, with classification we don't us a plot like the 3d view since it's hard to visualize, instead use 2d plot with decision boundary

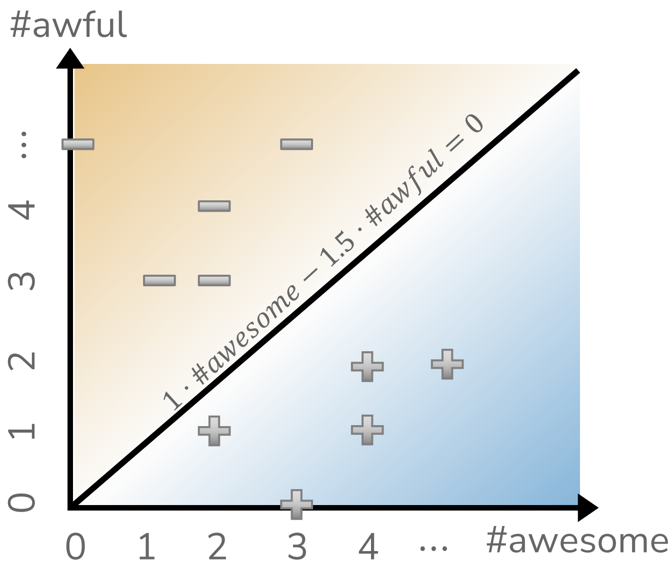
Decision Boundary with Score

$$Score(x) = 1 \cdot \#awesome - 1.5 \cdot \#awful$$

2-dimensional view



2-d view with score

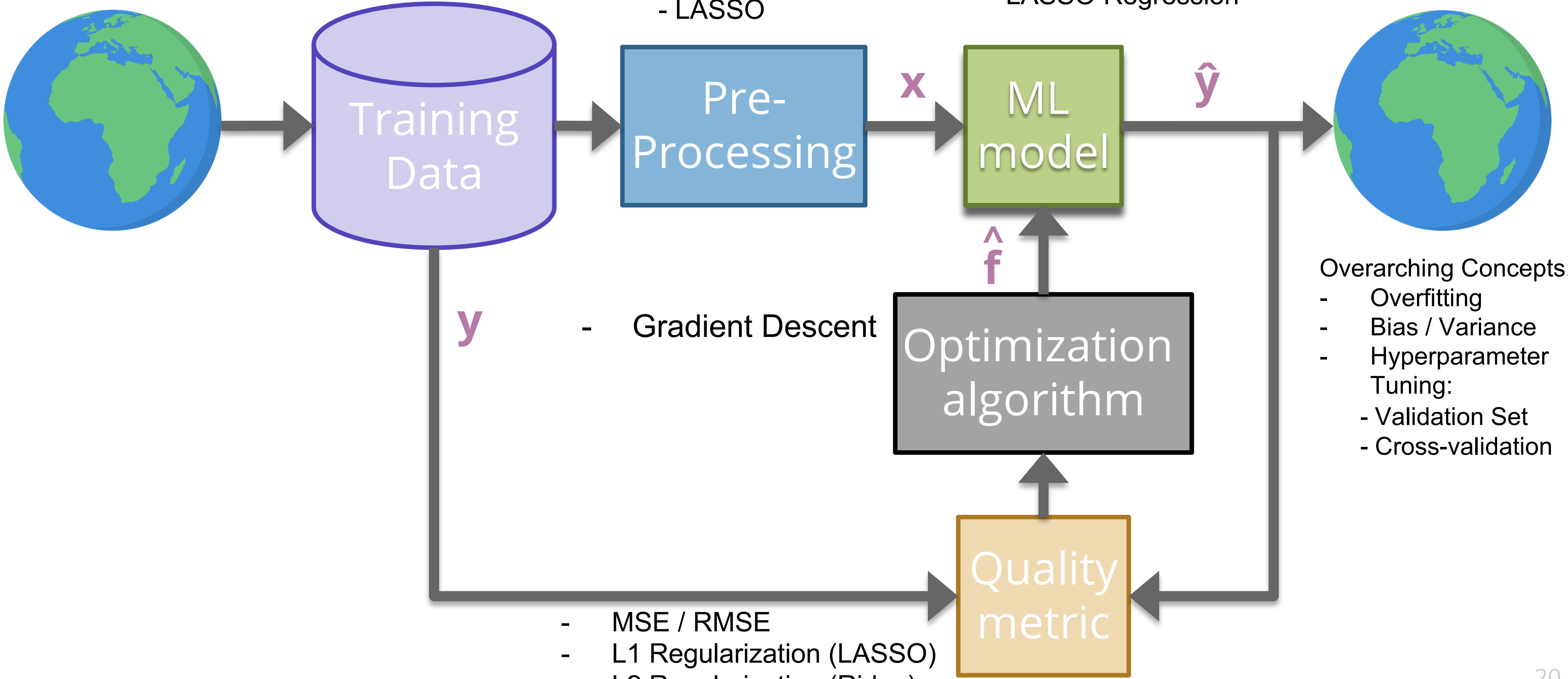


ML Pipeline

- Train/val/test -split

- Scaling / Normalization
- Feature Selection
  - All Subsets
  - Greedy
  - LASSO

- Linear Regression
- Polynomial Regression
- Ridge Regression
- LASSO Regression



- Gradient Descent

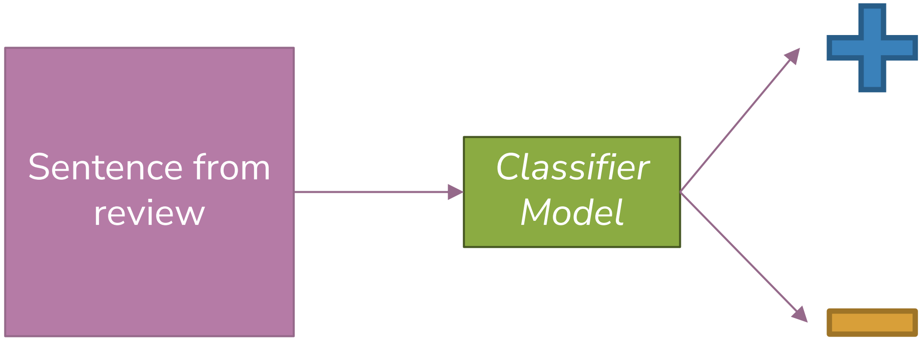
- MSE / RMSE
- L1 Regularization (LASSO)
- L2 Regularization (Ridge)

Overarching Concepts:
- Overfitting
- Bias / Variance
- Hyperparameter Tuning:
  - Validation Set
  - Cross-validation

Classification

Sentiment Classifier

In our example, we want to classify a restaurant review as positive or negative.



Input: x

Output: y
Predicted class

Attempt 1: Simple Threshold Classifier

Idea: Use a list of good words and bad words, classify review by the most frequent type of word

Word	Good?
sushi	None
was	None
great	Good
the	None
food	None
but	None
awesome	Good
service	None
terrible	Bad
rancid	Bad

Simple Threshold Classifier

Input x: Sentence from review

Count the number of positive and negative words, in x

If $\mathrm{num\_positive} > \mathrm{num\_negative}$:

- $\hat{y} = +1$

Else:

- $\hat{y} = -1$

Example: "Sushi was great, the food was awesome, but the service was terrible"

Attempt 2: Linear Classifier

Idea: Use labelled training data to learn a weight for each word. Use weights to score a sentence.

Model:

$$\hat{y}_i = sign(Score(x_i)) = sign(s_i)$$

$$= sign\left(\sum_{j=0}^{D} w_j h_j(x_i)\right) = sign(w^T h(x_i))$$

$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$	$h_5(x)$	$h_6(x)$	$h_7(x)$	$h_8(x)$	$h_9(x)$
sushi	**was**	**great**	**the**	**food**	**awesome**	**but**	**service**	**terrible**
1	3	1	2	1	1	1	1	1

Word	Weight
sushi	0
was	0
great	1
the	0
food	0
awesome	2
but	0
service	0
terrible	-1

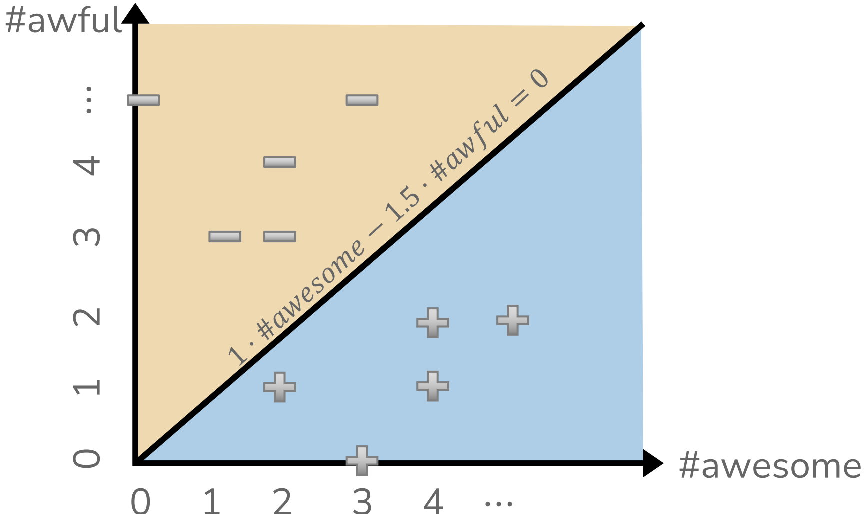
"Sushi was great, the food was awesome, but the service was terrible"

Decision Boundary

Consider if only two words had non-zero coefficients

Word	Coefficient	Weight
	w_0	0.0
awesome	w_1	1.0
awful	w_2	-1.5

$$\hat{s} = 1 \cdot \#awesome - 1.5 \cdot \#awful$$



slido

Think

1 min

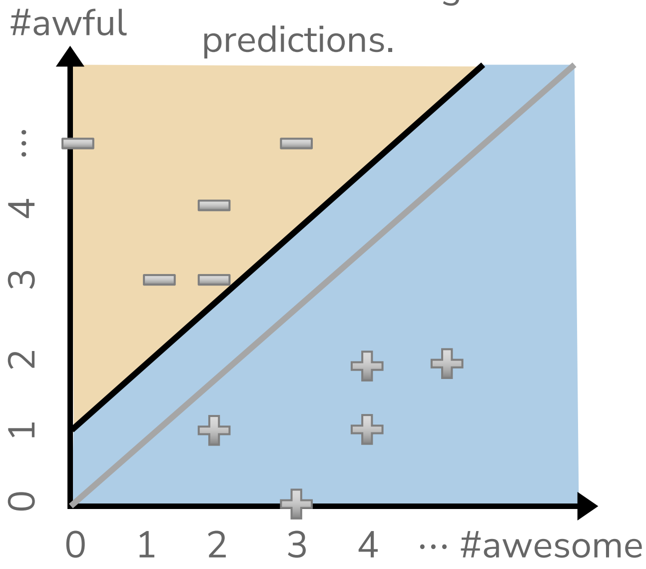
sli.do #cs416

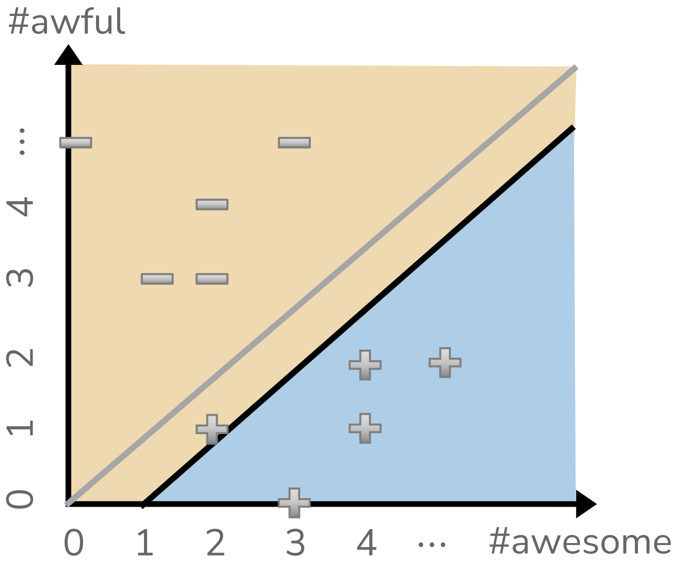
What happens to the decision boundary if we add an intercept?

$$Score(x) = 1.0 + 1 \cdot \#awesome - 1.5 \cdot \#awful$$

Which graph shows the new decision boundary (black)?

Describe in English what we are doing in terms of the model's predictions.





slido

Group

2 min

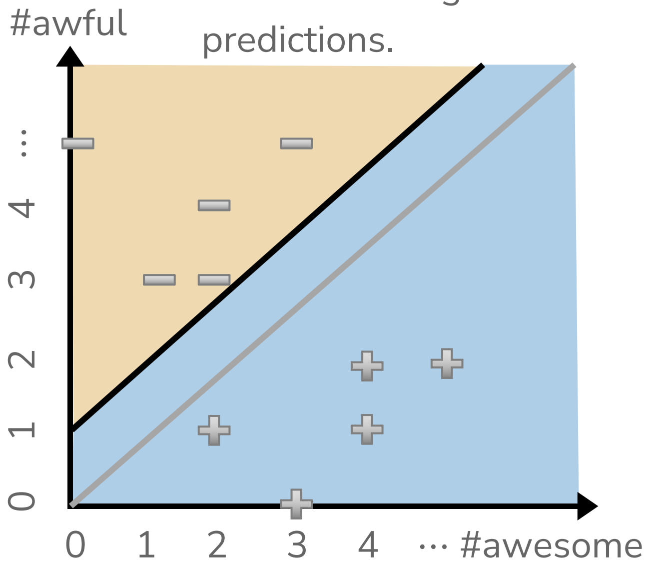
sli.do #cs416

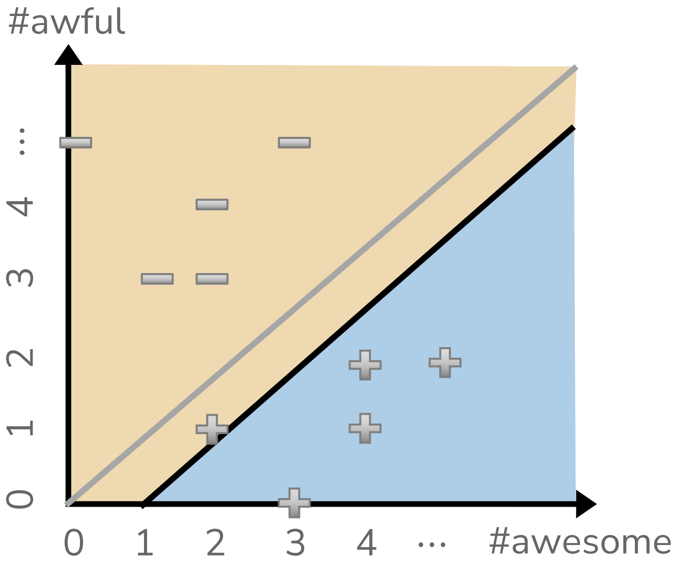
What happens to the decision boundary if we add an intercept?

$$Score(x) = 1.0 + 1 \cdot \#awesome - 1.5 \cdot \#awful$$

Which graph shows the new decision boundary (black)?

Describe in English what we are doing in terms of the model's predictions.

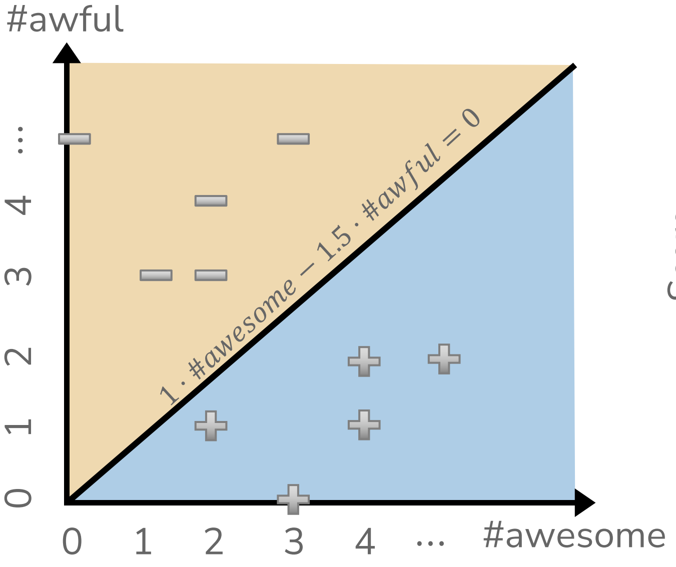




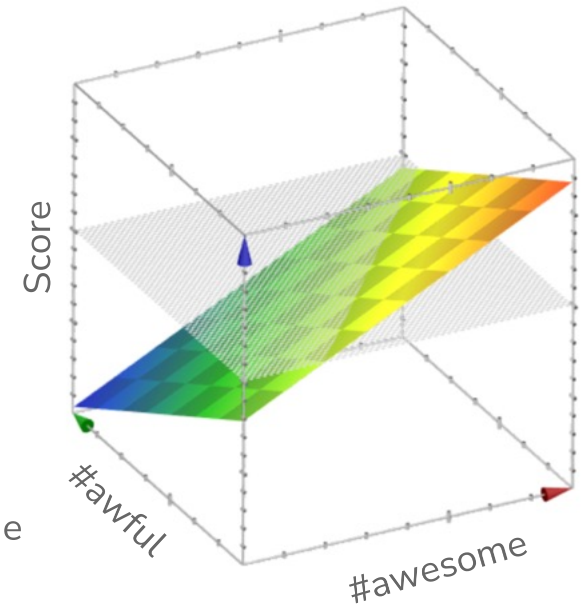
Decision Boundary

$$Score(x) = 1 \cdot \#awesome - 1.5 \cdot \#awful$$

2-dimensional view



3-dimensional view

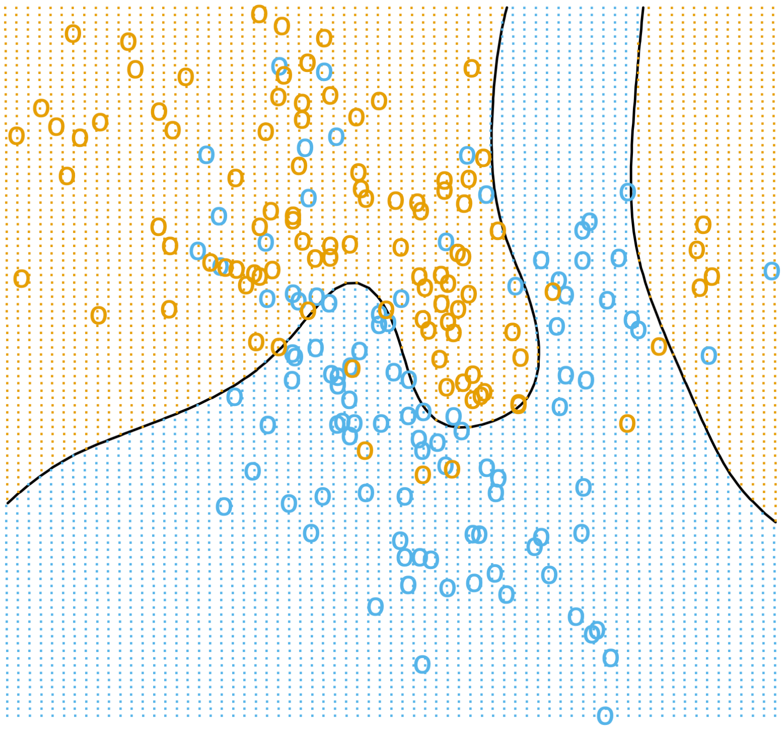


Generally, with classification we don't us a plot like the 3d view since it's hard to visualize, instead use 2d plot with decision boundary

Complex Decision Boundaries?

What if we want to use a more complex decision boundary?

- Need more complex model/features! (Come back Wed)



Single Words Are Sometimes Not Enough!

What if instead of making each feature one word, we made it two?

- **Unigram**: a sequence of one word
- **Bigram**: a sequence of two words
- **N-gram**: a sequence of n-words

”Sushi was good, the food was good, the service was not good”

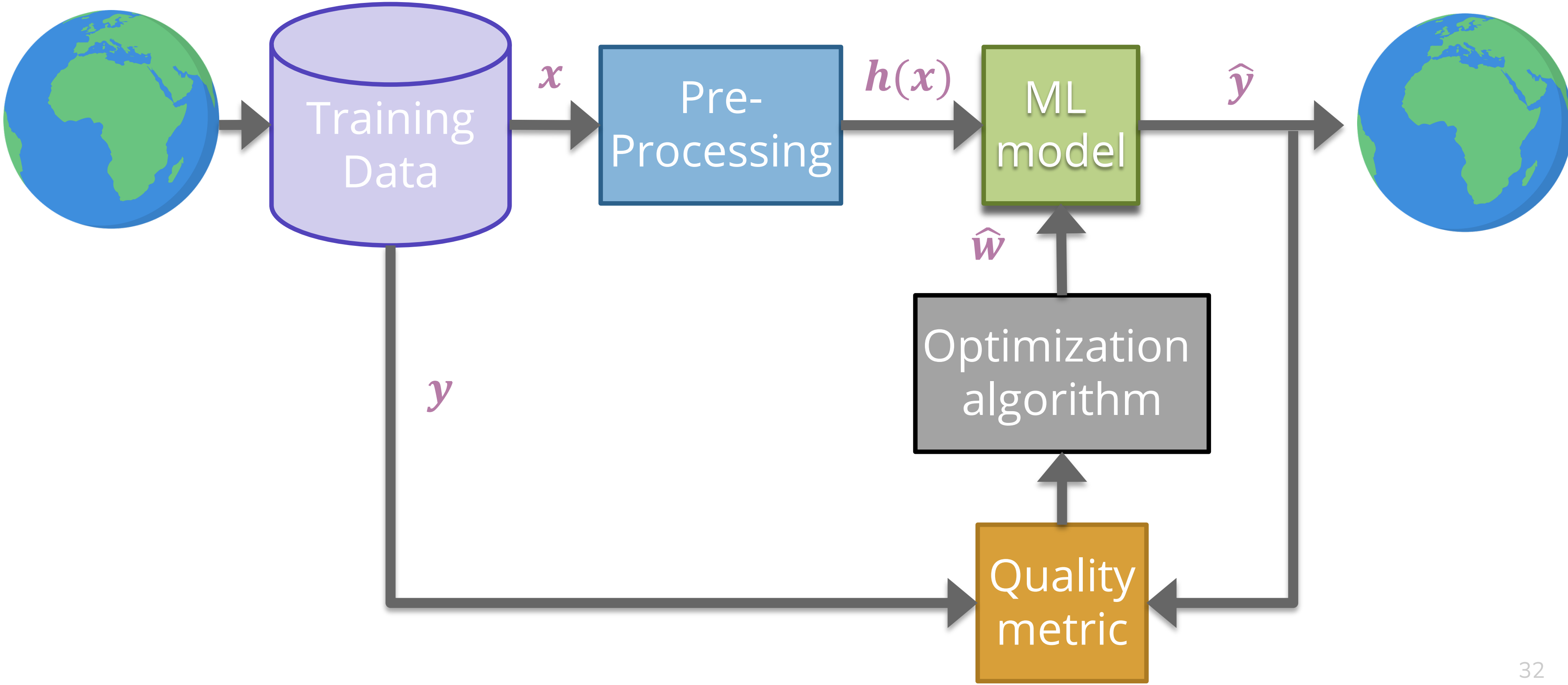
sushi	was	good	the	food	service	not
1	3	3	2	1	1	1

sushi was	was good	good the	the food	food was	the service	service was	was not	not good
1	2	2	1	1	1	1	1	1

Longer sequences of words results in more context, more features, and a greater chance of overfitting.

Evaluating Classifiers

ML Pipeline



Training Data
x
Pre-Processing
$h(x)$
ML model
$\hat{y}$
$\hat{w}$
Optimization algorithm
y
Quality metric

Classification Error

Ratio of examples where there was a mistaken prediction

What's a mistake?

If the true label was positive ($y = +1$),
but we predicted negative ($\hat{y} = -1$)

If the true label was negative ($y = -1$),
but we predicted positive ($\hat{y} = +1$)

Classification Error

Classification Accuracy

What's a good accuracy?

For binary classification:

Should at least beat random guessing...

Accuracy should be at least 0.5

For multi-class classification (k classes):

Should still beat random guessing

Accuracy should be at least: 1 / k

- 3-class: 0.33
- 4-class: 0.25
- ...

Besides that, higher accuracy means better, right?

Detecting Spam

Imagine I made a “Dummy Classifier” for detecting spam

The classifier ignores the input, and always predicts spam.

This actually results in 90% accuracy! Why?

- Most emails are spam…

This is called the **majority class classifier.**

A classifier as simple as the majority class classifier can have a high accuracy if there is a **class imbalance**.

A class imbalance is when one class appears much more frequently than another in the dataset

This might suggest that accuracy isn't enough to tell us if a model is a good model.

Assessing Accuracy

Always digging in and ask critical questions of your accuracy.

Is there a **class imbalance**?

How does it compare to a baseline approach?

- Random guessing
- Majority class
- ...

Most important: **What does my application need?**

- What's good enough for user experience?
- What is the impact of a mistake we make?

Brain Break

Confusion Matrix

For binary classification, there are only two types of mistakes

$$\hat{y} = +1, \quad y = -1$$

$$\hat{y} = -1, \quad y = +1$$

Generally we make a **confusion matrix** to understand mistakes.

	Predicted Label	
True Label	+	−
+	True Positive (TP)	False Negative (FN)
−	False Positive (FP)	True Negative (TN)

Tip on remembering: complete the sentence "My prediction was a …"

Confusion Matrix Example

		Predicted Label	
		+	−
True Label	+	True Positive (TP)	False Negative (FN)
	−	False Positive (FP)	True Negative (TN)

Which is Worse?

What's worse, a false negative or a false positive?

It entirely depends on your application!

Detecting Spam

False Negative: Annoying

False Positive: Email lost

Medical Diagnosis

False Negative: Disease not treated

False Positive: Wasteful treatment

In almost every case, how treat errors depends on your context.

Errors and Fairness

We mentioned on the first day how ML is being used in many contexts that impact crucial aspects of our lives.

Models making errors is a given, what we do about that is a choice:

- Are the errors consequential enough that we shouldn't use a model in the first place?
- Do different demographic groups experience errors at different rates?
  - If so, we would hopefully want to avoid that model!

Will talk more about how to define whether or a not a model is fair / discriminatory next week. Will use these notions of error as a starting point!

Binary Classification Measures

Notation

$$C_{TP} = \#\mathrm{TP}, \quad \mathrm{C}_{\mathrm{FP}} = \#\mathrm{FP}, \quad \mathrm{C}_{\mathrm{TN}} = \#\mathrm{TN}, \quad \mathrm{C}_{\mathrm{FN}} = \#\mathrm{FN}$$

$$N = C_{TP} + C_{FP} + C_{TN} + C_{FN}$$

$$N_P = C_{TP} + C_{FN}, \quad N_N = C_{FP} + C_{TN}$$

Error Rate

$$\frac{C_{FP} + C_{FN}}{N}$$

Accuracy Rate

$$\frac{C_{TP} + C_{TN}}{N}$$

False Positive rate (FPR)

$$\frac{C_{FP}}{N_N}$$

False Negative Rate (FNR)

$$\frac{C_{FN}}{N_P}$$

True Positive Rate or Recall

$$\frac{C_{TP}}{N_P}$$

Precision

$$\frac{C_{TP}}{C_{TP} + C_{FP}}$$

F1-Score

$$2\,\frac{Precision \cdot Recall}{Precison + Recall}$$

[See more!]

Multiclass Confusion Matrix

Consider predicting (*Healthy, Cold, Flu*)

Predicted Label

True Label	Healthy	Cold	Flu
Healthy	60	8	2
Cold	4	12	4
Flu	0	2	8

slido

Think

1 min

sli.do #cs416

Suppose we trained a classifier and computed its confusion matrix on the training dataset. **Is there a class imbalance in the dataset and if so, which class has the highest representation?**

Predicted Label

True Label	Pupper	Doggo	Woofer
Pupper	2	27	4
Doggo	4	25	4
Woofer	1	30	2

slido

Group

2 min

sli.do #cs416

Suppose we trained a classifier and computed its confusion matrix on the training dataset. **Is there a class imbalance in the dataset and if so, which class has the highest representation?**

Predicted Label

True Label	Pupper	Doggo	Woofer
Pupper	2	27	4
Doggo	4	25	4
Woofer	1	30	2

Learning Theory

How much data?

The more the merrier

- But data quality is also an extremely important factor

Theoretical techniques can bound how much data is needed

- Typically too loose for practical applications
- But does provide some theoretical guarantee

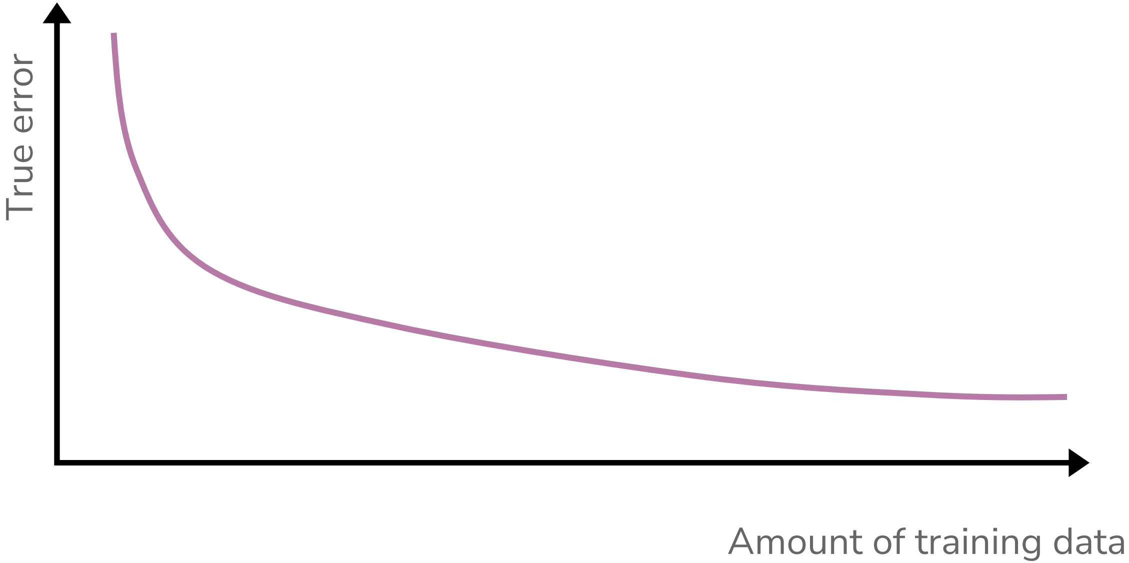
In practice

- More complex models need more data

Learning Curve

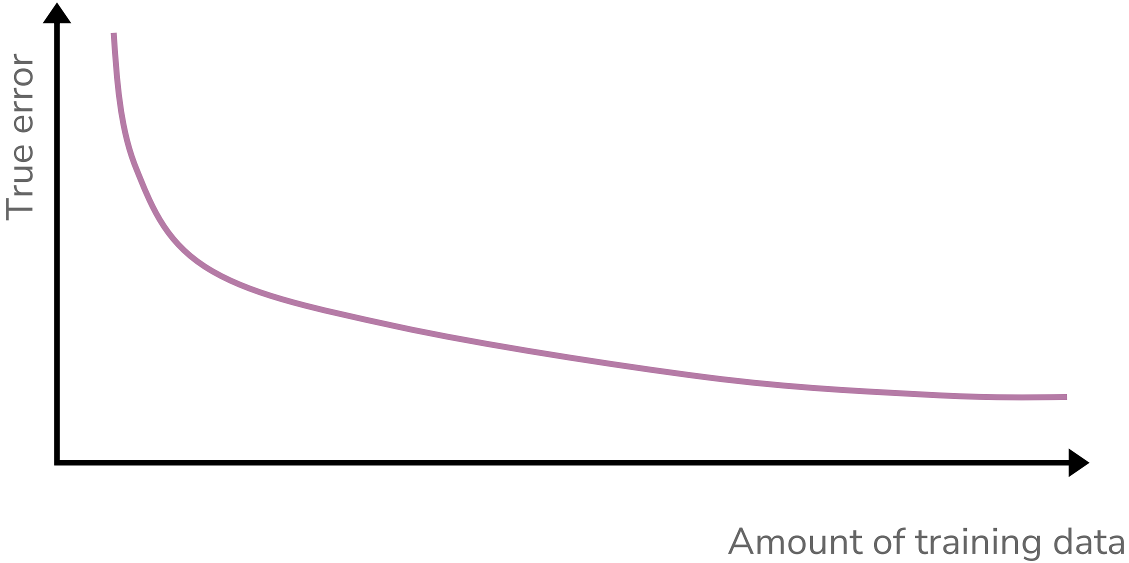
How does the true error of a model relate to the amount of training data we give it?

Hint: We've seen this picture before



Learning Curve

What if we use a more complex model?



Next Time

We will address the issues highlighted with the Linear Classifier approach from today by predicting the probability of a sentiment, rather than the sentiment itself.

$$P(y|x)$$

Normally assume some structure on the probability (e.g., linear)

$$P(y|x,w) \approx w^T x$$

Use machine learning algorithm to learn approximate $\hat{w}$ such that $\hat{P}(y|x)$ is close to $P(y|x)$, where:

$$\hat{P}(y|x) = P(y|x,\hat{w})$$

Recap

Theme: Describe high level idea and metrics for classification

Ideas:

- Applications of classification
- Linear classifier
- Decision boundaries
- Classification error / Classification accuracy
- Class imbalance
- Confusion matrix
- Learning theory