NLP Specialization

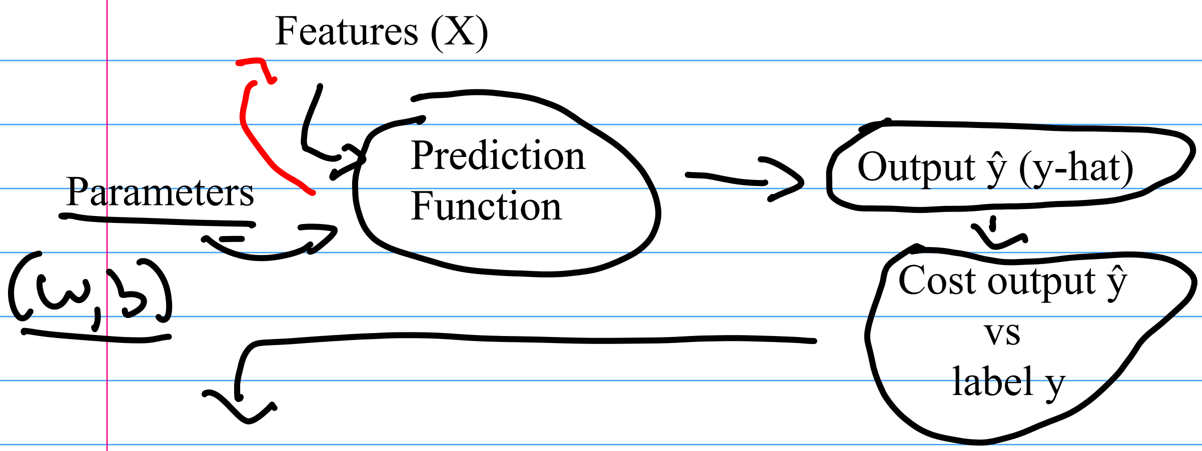
(Natural Language Processing)

Using Supervised Learning with logistic regression:

Features (X)

Labels (Y)

To make sure you're getting the most accurate prediction based on your data, the goal is to minimize the error rate as much as possible (with cost error function + gradient descent)



Features (X)

Parameters

(w,b)

Prediction Function

Output ŷ (y-hat)

Cost output ŷ vs label y

Update parameters and repeat the process until the cost function is minimized

Supervised machine learning of Sentiment Analysis

Tweet analysis:

"I am happy today because I slept well"

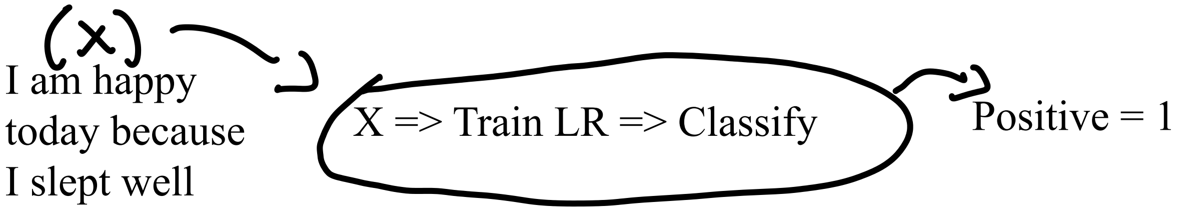
Goal: Predict whether a tweet has a positive or a negative sentiment

-> Create a training set with the following label:

Positive sentiment = 1
Negative sentiment = 0

Using a logistic regression classifier which assigns 2 distinct classes

Workflow:



(X)

I am happy
today because
I slept well

X => Train LR => Classify

Positive = 1

Process the raw tweets in the training set and extract useful features

Steps:
1 - Extract the Features
2 - Train your model
3 - Classify based on the trained model

Extracting features:

Represent a text as a vector

1) Build a vocabulary that allows to encode any text or any tweet as an array of numbers

Vocabulary:

Tweets:

[tweet_1, tweet_2, ..., tweet_m]

"I am happy because I slept well"
...
"I hated the movie"

[I, am, happy, because, slept, well, hated, the, movie]

Check if every word from the vocabulary appears in the tweet:
E.g. = "I am happy because I slept well"

If it does, assign a value of 1 to that feature.
If it doesn't, assign a value of 0

1 = I
1 = am
1 = happy
1 = because
1 = slept
1 = well
0 = hated
0 = the
0 = movie