How to automatically label text?

Sandeep Soni

09/26/2024

Question for the day

"How to predict a label for given text?"

Agenda

- Classification problem
- Naive Bayes
- Logistic regression
- Training/test setup
- Evaluation metrics

Text classification

- Input: document (e.g., email)
- Output: label (e.g., spam/ham)

Text classification problems

Task	x	y
Language ID	text	{english, mandarin, hindi, …}
spam classification	email	{spam, ham}
party affiliation	speech	{republican, democrat}
sentiment analysis	text	{positive, negative, mixed, neutral}
music genre	lyric	{rock, pop, jazz, rap,…}

Formal task

Formal task

- x is an instance (e.g., an email)
  - $x \in \mathcal{X}$ (e.g., set of emails)

Formal task

- x is an instance (e.g., an email)
  - $x \in \mathcal{X}$ (e.g., set of emails)
- y is the desired label (e.g., spam)
  - $y \in \mathcal{Y}$ (e.g., {spam, ham})

Formal task

- x is an instance (e.g., an email)
  - $x \in \mathcal{X}$ (e.g., set of emails)
- y is the desired label (e.g., spam)
  - $y \in \mathcal{Y}$ (e.g., {spam, ham})
- $y = h(x)$
  - h maps instances to labels

Classification

- True h is unknown so find $\hat{h}$ that's a closest approximation

Supervised learning

- Learn $\hat{h}$ from training data given in the form of several $<x, y>$ pairs

Naive Bayes

- One simple yet effective classification method is Naive Bayes
  - Similar to LDA, it's a generative model
  - We'll represent input text as a bag of words vector

Naive Bayes

- Naive conditional independence assumption

$$P(x_i \mid x_{i-1}, x_{i-2}, \ldots, y) = P(x_i \mid y)$$

- Given the category, the words (features) are independent of each other
- Under naive Bayes, P(“prince”|spam) = P(“prince”|spam, “kenyan”)

Now we can rewrite the posterior probability as:

$$\approx P(y)\ P(x_1|y)\ P(x_2|y)\ P(x_3|y)\ldots P(x_n|y)$$

$$= P(y) \prod_{i=1}^{n} P(x_i|y)$$

This is tractable!

Now we can rewrite the posterior probability as:

$$P(y \mid x_1, x_2, \ldots, x_n) \propto P(y)\ P(x_1 \mid y)\ P(x_2 \mid x_1, y)\ P(x_3 \mid x_2, x_1, y) \ldots P(x_n \mid x_{n-1}, \ldots, x_1, y)$$

$$\approx P(y)\ P(x_1 \mid y)\ P(x_2 \mid y)\ P(x_3 \mid y) \ldots P(x_n \mid y)$$

$$= P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

This is tractable!

We can estimate these probabilities by simply counting the instances in training data

$$P(y = spam) = \frac{\#\text{samples labeled spam}}{\#\text{ samples}}$$

$$P(\text{"kenyan"} \mid \text{spam}) = \frac{\#\text{samples labeled spam and contain "kenyan"}}{\#\text{samples labeled spam}}$$

Picking the label

- Once you estimate the probabilities from training data, you can pick the label that maximizes the posterior

$$P(y = \text{spam} \mid \text{text}) > P(y = \text{ham} \mid \text{text})$$ Spam

otherwise Ham

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^{N} \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^{N} \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto 1 \cdot \log P(y) + \sum_{i=1}^{N} 1 \cdot \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^{N} \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto 1 \cdot \log P(y) + \sum_{i=1}^{N} 1 \cdot \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto f_0 \cdot w_0 + \sum_{i=1}^{N} f_i \cdot w_i$$

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^{N} \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto 1 \cdot \log P(y) + \sum_{i=1}^{N} 1 \cdot \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto f_0 \cdot w_0 + \sum_{i=1}^{N} f_i \cdot w_i$$

$$\log P(y \mid x) \propto \sum_{i=0}^{N} f_i \cdot w_i = w^T f$$

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^{N} \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto 1 \cdot \log P(y) + \sum_{i=1}^{N} 1 \cdot \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto f_0 \cdot w_0 + \sum_{i=1}^{N} f_i \cdot w_i$$

$$\log P(y \mid x) \propto \sum_{i=0}^{N} f_i \cdot w_i = w^T f$$

Weights

Features

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^{N} \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto 1 \cdot \log P(y) + \sum_{i=1}^{N} 1 \cdot \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto f_0 \cdot w_0 + \sum_{i=1}^{N} f_i \cdot w_i$$

$$\log P(y \mid x) \propto \sum_{i=0}^{N} f_i \cdot w_i = w^T f$$

Weights

Features

Naive Bayes is a linear model

Logistic Regression

Logistic Regression

In logistic regression, we directly model the conditional probability of the label given the text

Logistic Regression

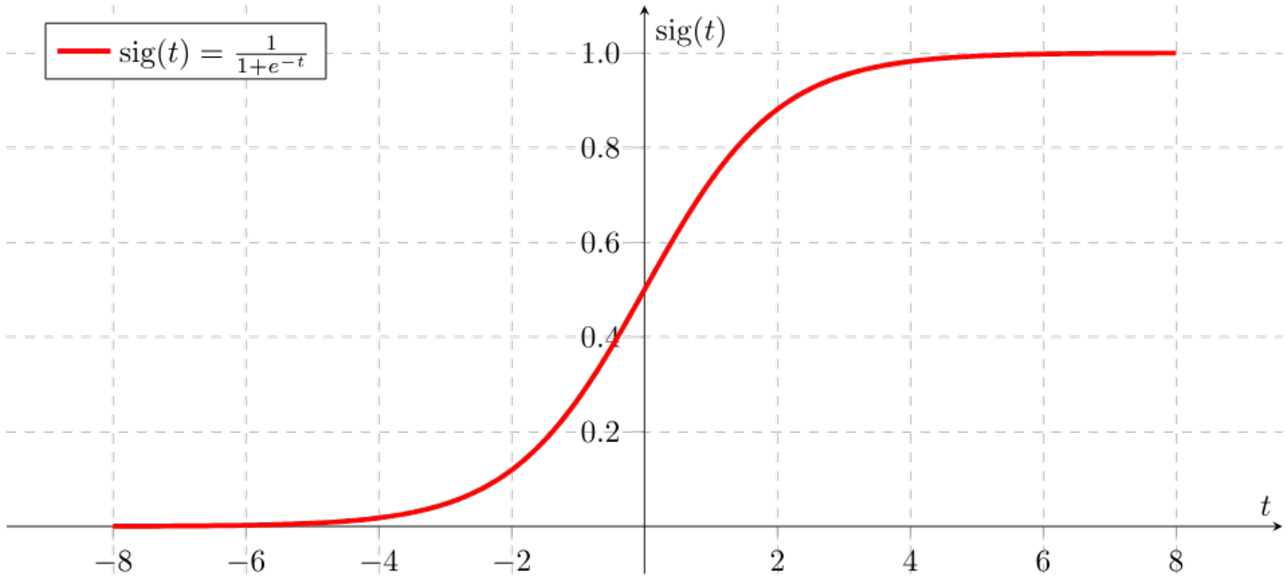
In logistic regression, we directly model the conditional probability of the label given the text

$$P(y \mid x) = \text{sig}(w^T f(x))$$

Logistic Regression

In logistic regression, we directly model the conditional probability of the label given the text

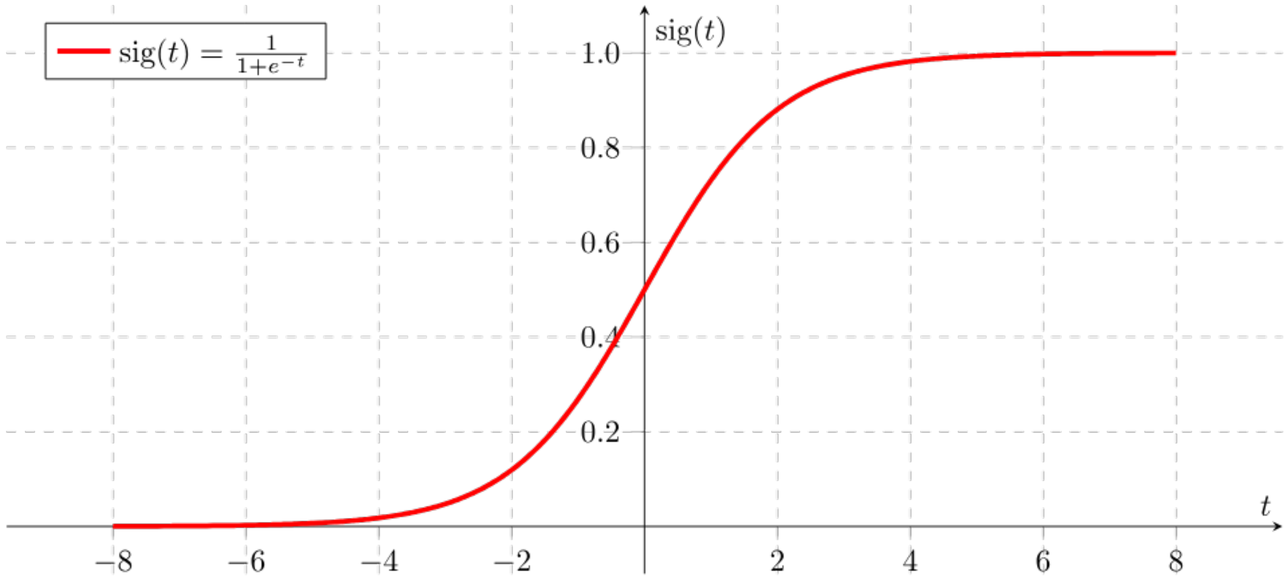
$$P(y \mid x) = \text{sig}(w^T f(x))$$



Logistic Regression

In logistic regression, we directly model the conditional probability of the label given the text

$$P(y \mid x) = \text{sig}(w^T f(x))$$



This is highly flexible because we can encode any type of features that we think could be useful

Learning

Learning

- The objective is to learn the vector of weights w

Learning

- The objective is to learn the vector of weights w
- Unlike Naive Bayes, no closed form solution

Learning

- The objective is to learn the vector of weights w
- Unlike Naive Bayes, no closed form solution
- Weights are learned by optimizing some criterion on training data

Generalized classification

Generalized classification

- Alternatives to linear models also exist.

Generalized classification

- Alternatives to linear models also exist.
- $P(y \mid x) = \mathtt{sig}(g(f(x)))$

Generalized classification

- Alternatives to linear models also exist.
- $P(y \mid x) = \text{sig}(g(f(x)))$
- For multi-class classification, $P(y \mid x) = \text{softmax}(g(f(x)))$

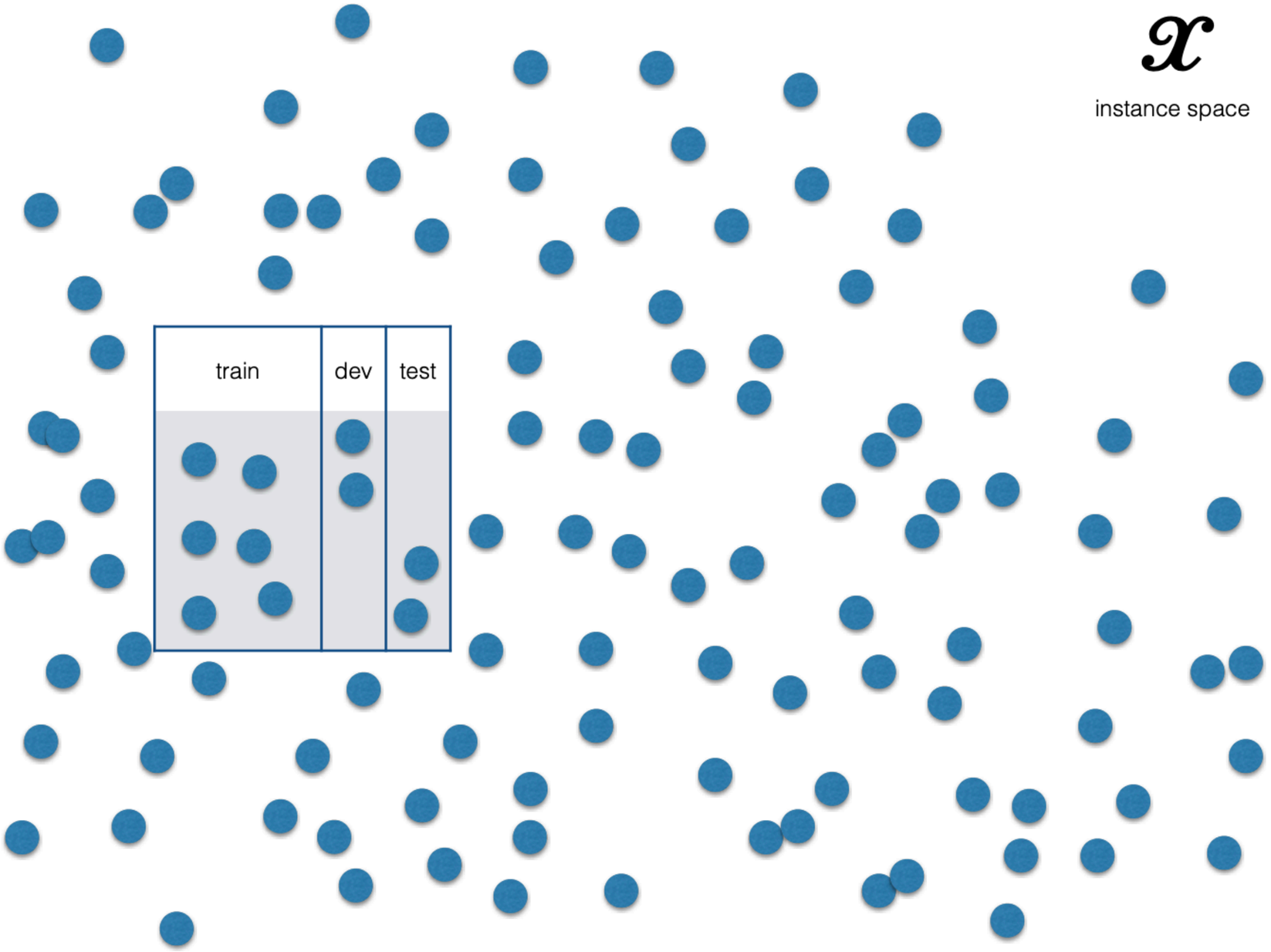
Generalized classification

- Alternatives to linear models also exist.
- $P(y|x) = \text{sig}(g(f(x)))$ g could be a neural network
- For multi-class classification, $P(y|x) = \text{softmax}(g(f(x)))$

Generalized forecasting

Generalized forecasting

- Same model can be used to predict continuous values
- $y = g(f(x))$
- If g is a linear function, this is linear regression



Slide from David Bamman

Experiment design

- Training set is to estimate parameters of the model
- Development set is to perform model selection
- Test set for evaluation

Experiment design

- Typically, we use 80% data for training, 10% for model selection and 10% for evaluation
- One should be careful never to use to development or test data to do estimation

Feature function

Feature function

- Feature function transforms text into a feature vector
  - e.g. text -> counts vector
  - e.g. text -> counts vector | topics proportion

Regularization

Regularization

- To enforce some structure or bake in some domain expertise, we use regularization by adding penalty terms to our optimization objective (e.g., if we don't want very high weights, we'll add a penalty to our loss that we're minimizing)

Regularization

- To enforce some structure or bake in some domain expertise, we use regularization by adding penalty terms to our optimization objective (e.g., if we don't want very high weights, we'll add a penalty to our loss that we're minimizing)

- The value of the penalty is controlled by a hyper parameter which should be tuned on the development set

How do we know if our learned classifier is good?

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

Confusion Matrix

Confusion Matrix

Predicted \ Observed	y=spam	y=ham
yhat = spam	4	3
yhat=ham	1	2

Confusion Matrix

Confusion Matrix

Predicted \ Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

Confusion Matrix

Predicted \ Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

- N = True positives + True negatives + False positives + False negatives

Accuracy

Accuracy

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

Accuracy

$$\frac{\text{True positives} + \text{True negatives}}{N}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

Accuracy

$$\frac{\text{True positives} + \text{True negatives}}{N}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

- We want the accuracy of the classifier to be high

Precision

Precision

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

Precision

$$\frac{\text{True positives}}{\text{True positives + False positives}}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

Precision

$$\frac{\text{True positives}}{\text{True positives + False positives}}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

- We want number of false positives to be low and precision to be high

Recall

Recall

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

Recall

$$\frac{\text{True positives}}{\text{True positives + False negatives}}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

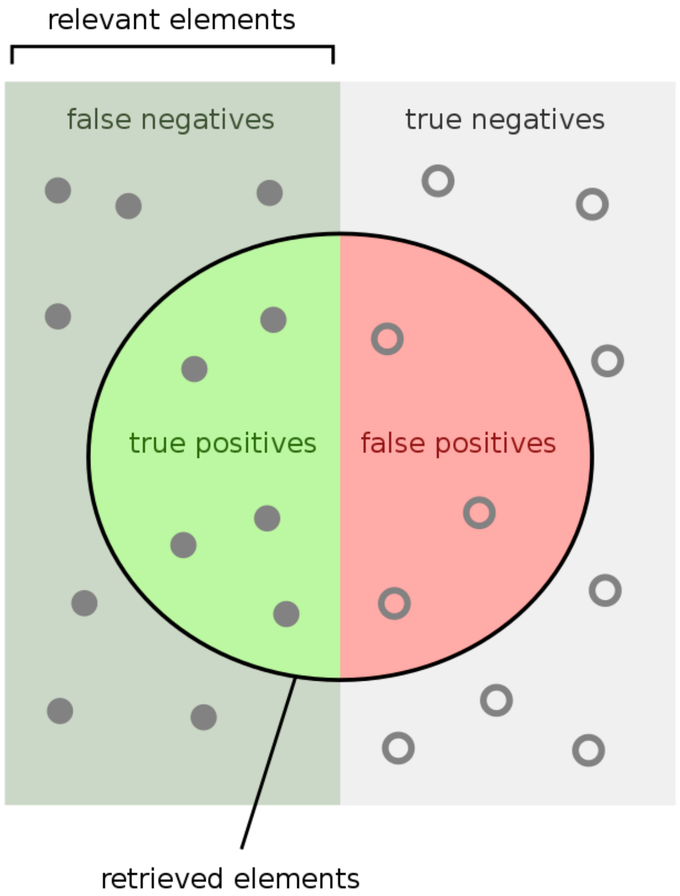
Recall

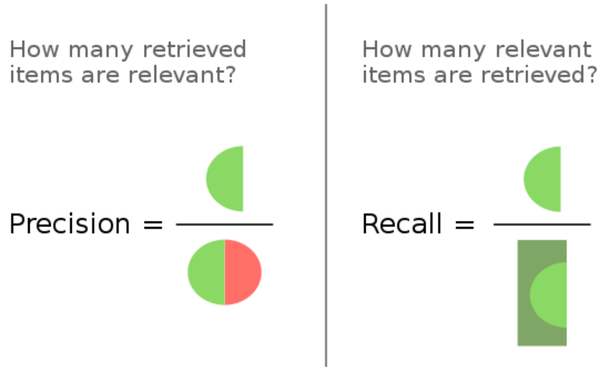
$$\frac{\text{True positives}}{\text{True positives + False negatives}}$$

Predicted\Observed	y=spam	y=ham
yhat = spam	True positives	False positives
yhat=ham	False negatives	True negatives

- We want number of false negatives to be low and recall to be high

Precision and Recall





Source: Wikipedia

F1

- We can combine precision and recall into a single metric by taking the harmonic mean of the two quantities.

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

In class

- Text classification demo