Model Evaluation

Boston University CS 506 - Lance Galletti

Confusion Matrix

	Predicted Class		
Actual Class		Class = Yes	Class = No
	Class = Yes	a (TP)	b (FN)
	Class = No	c (FP)	d (TN)

Accuracy = (a + d) / (a + b + c + d)

Accuracy can be misleading

Binary classification problem where:

Number of Class 0 examples: 9990

Number of Class 1 examples: 10

A model that predicts everything to be class 0 will have an accuracy of 99.9%

Cost Matrix

	Predicted Class		
Actual Class		Class = Yes	Class = No
	Class = Yes	C(Yes|Yes)	C(No|Yes)
	Class = No	C(Yes|No)	C(No|No)

Cost of Classification

COST	Predicted Class		
Actual Class		Yes	No
	Yes	-1	100
	No	1	0

Model 1	Predicted Class		
Actual Class		Yes	No
	Yes	150	40
	No	60	250

Accuracy = 80%
Cost = 3910

Model 2	Predicted Class		
Actual Class		Yes	No
	Yes	250	45
	No	5	200

Accuracy = 90%
Cost = 4255

Other metrics

COST	Predicted Class		
Actual Class		Yes	No
	Yes	a	b
	No	c	d

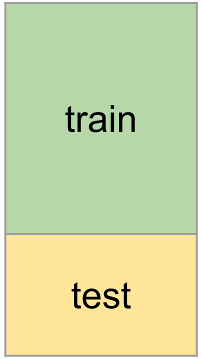
- Precision = a / (a + c)
- Recall = a / (a + b)
- F-measure = 2RP / (R + P)

Methods of Estimation

Goal: get a reliable estimate of the performance of the model on unseen data

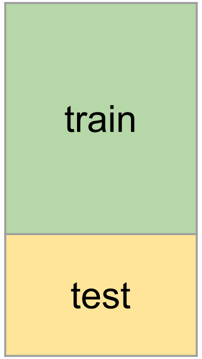
Methods of Estimation

- Holdout:
  - Ex: reserve ¼ of the dataset for testing and use ¾ for training

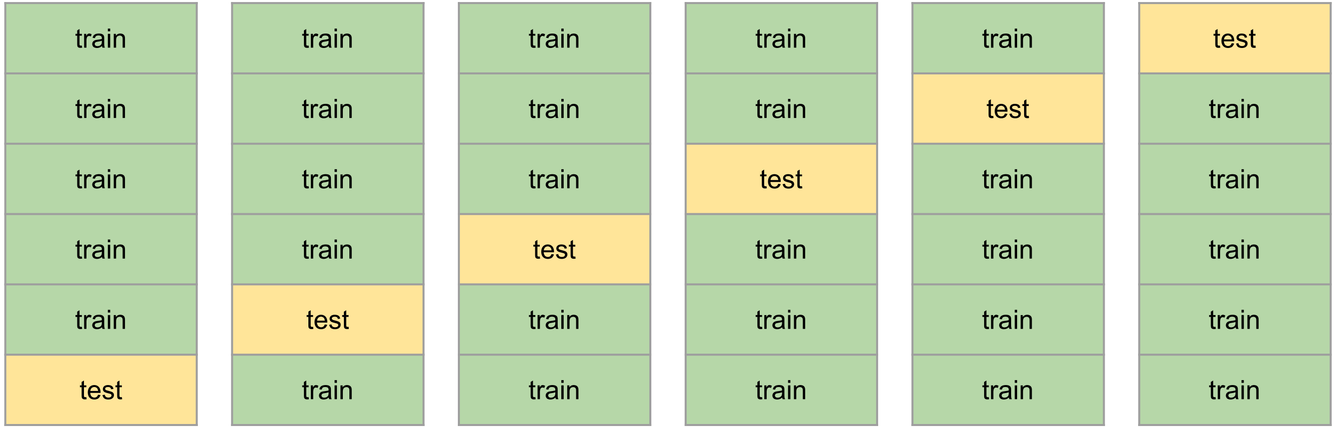


Methods of Estimation

- Holdout:
  - Ex: reserve ¼ of the dataset for testing and use ¾ for training

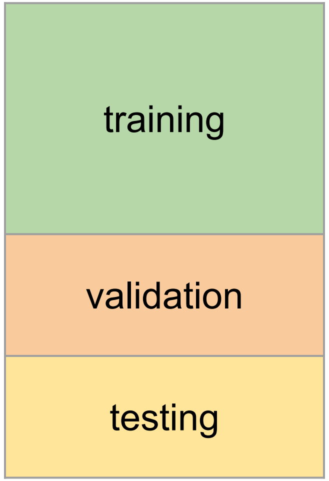


- Cross Validation:
  - Partition into K disjoint subsets
  - K-fold: train on K-1 partitions, test on the remaining one
  - K = n : leave-one-out



Validation Set

For tuning parameters



Discuss

You build a model on ¾ of a dataset and it performs well on the testing set. You're ready to ship to production. Before shipping the model live, do you retrain the model on the entire dataset in order to increase performance?

(yes)

https://datascience.stackexchange.com/questions/33008/is-it-always-better-to-use-the-whole-dataset-to-train-the-final-model

Ensemble Methods

Boston University CS 506 - Lance Galletti

The idea

Suppose you have trained 17 different classifiers on a dataset.

The idea

Suppose you have trained 17 different classifiers on a dataset.

- Every classifier has error rate **ε** = .20
- Assume all classifiers are independent

The idea

Suppose you have trained 17 different classifiers on a dataset.

- Every classifier has error rate $\boldsymbol{\varepsilon}$ = .20
- Assume all classifiers are independent

In order to classify a new record we poll all 17 classifiers and take the class that the majority agrees on.

What is the probability that this ensemble classifier makes a wrong prediction?

The idea

The majority needs to make a mistake (i.e. at least 9 out of 17 make mistakes)

$$P(X \geq 9) = \sum_{k=9}^{17} \binom{17}{k} (.2)^k (1 - .2)^{17-k} = 0.002581463$$

How to generate independent classifiers?

By generating samples of the data to train on

- Bagging
- Boosting

Bagging

Original data	1	2	3	4	5	6	7	8	9	10
Bootstrap sample 1	7	8	10	10	3	6	1	1	4	5
Bootstrap sample 2	6	2	7	9	3	5	7	7	1	8
Bootstrap sample 3	2	5	6	1	4	1	8	9	4	3

Build a classifier on each bootstrapped sample

Boosting

An adaptive sampling process to change the sampling distribution based on difficult-to-classify examples.

Start with all samples having equal probability of being selected. Next boosting round, increase the weights of those samples that were misclassified, decrease the weights of those samples that were correctly classified.

Boosting

Original data	1	2	3	4	5	6	7	8	9	10
Bootstrap sample 1	7	8	10	10	3	6	1	1	4	5
Bootstrap sample 2	6	4	7	9	3	5	7	4	1	8
Bootstrap sample 3	2	5	4	1	4	1	4	4	2	3

Here, sample 4 is hard to classify.

Boosting

Original data	1	2	3	4	5	6	7	8	9	10
Bootstrap sample 1	7	8	10	10	3	6	1	1	4	5
Bootstrap sample 2	6	4	7	9	3	5	7	4	1	8
Bootstrap sample 3	2	5	4	1	4	1	4	4	2	3

Classifiers trained on each sample dataset are given a weight that is a function of their error rate.

Boosting

Example: 5 classifiers

C_1 predicts 1
C_2 predicts 0
C_3 predicts 0
C_4 predicts 1
C_5 predicts 0

Boosting

Example: 5 classifiers

C_1 predicts 1 has a weight of .2
C_2 predicts 0 has a weight of .1
C_3 predicts 0 has a weight of .5
C_4 predicts 1 has a weight of .9
C_5 predicts 0 has a weight of .2