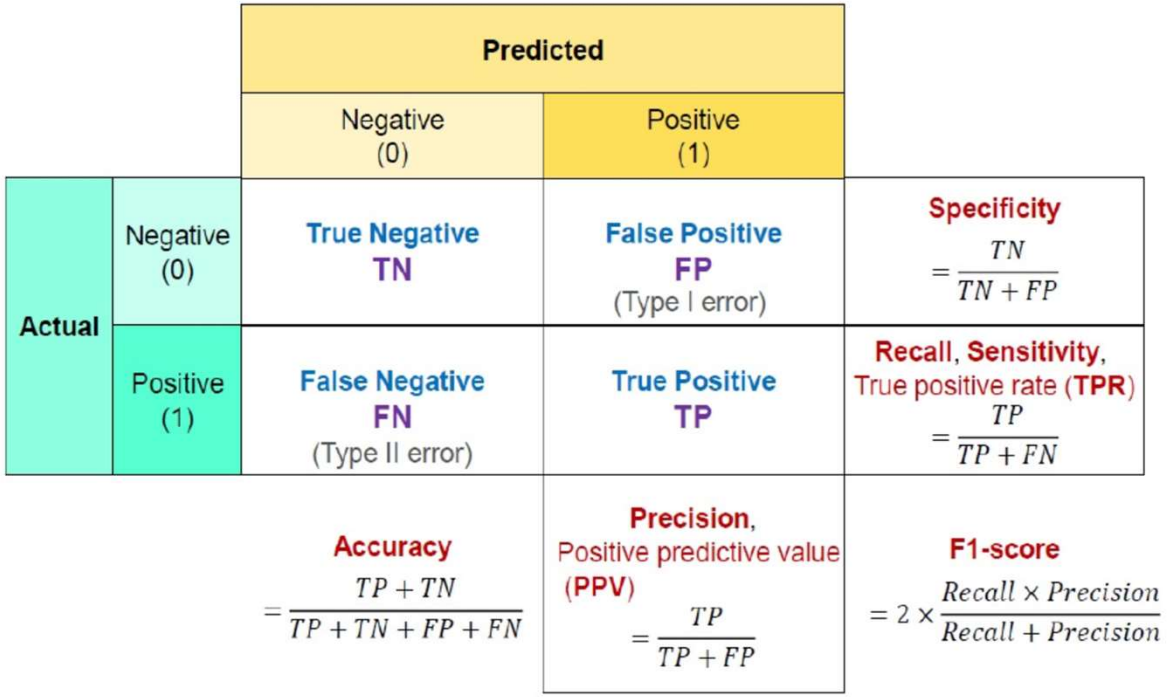
Metrics and Validation

Fatemeh Mansoori

Some images in these slides are selected from the slides of the Machine Learning course by Sharifi zarchi at Sharif University

Confusion Matrix



The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of different classes

TP and *TN* tell us when the classifier is getting things right, while *FP* and *FN* tell us when the classifier is getting things wrong

Class imbalance

- **class imbalance problem**
  - main class of interest is rare.
  - data set distribution reflects a significant majority of the negative class and a minority positive class
  - E.g. in fraud detection applications, the class of interest (or positive class) is *“fraud,”* which occursmuch less frequently than the negative *“nonfraudulant”* class

Class imbalance a scenario

- In medical data, there may be a rare class, such as *"cancer."* Suppose that you have trained a classifier to classify medical data tuples, where the class label attribute is *"cancer"* and the possible class values are *"yes"* and *"no."* An accuracy rate of, say, 97% may make the classifier seem quite accurate, but what if only, say, 3% of the training tuples are actually cancer? Clearly, an accuracy rate of 97% may not be acceptable—the classifier could be correctly labeling only the noncancer tuples, for instance, and misclassifying all the cancer tuples. Instead, we need other measures, which access how well the classifier can recognize the positive tuples (*cancer yes*) and how well it can recognize the negative tuples (*cancer no*).
- The **sensitivity** and **specificity** measures can be used, respectively, for this purpose

Class imbalance example

Classes	*yes*	*no*	*Total*	*Recognition (%)*
yes	**90**	**210**	300	30.00
no	**140**	**9560**	9700	98.56
Total	230	9770	10,000	96.40

How we can handle class-imbalanced data?

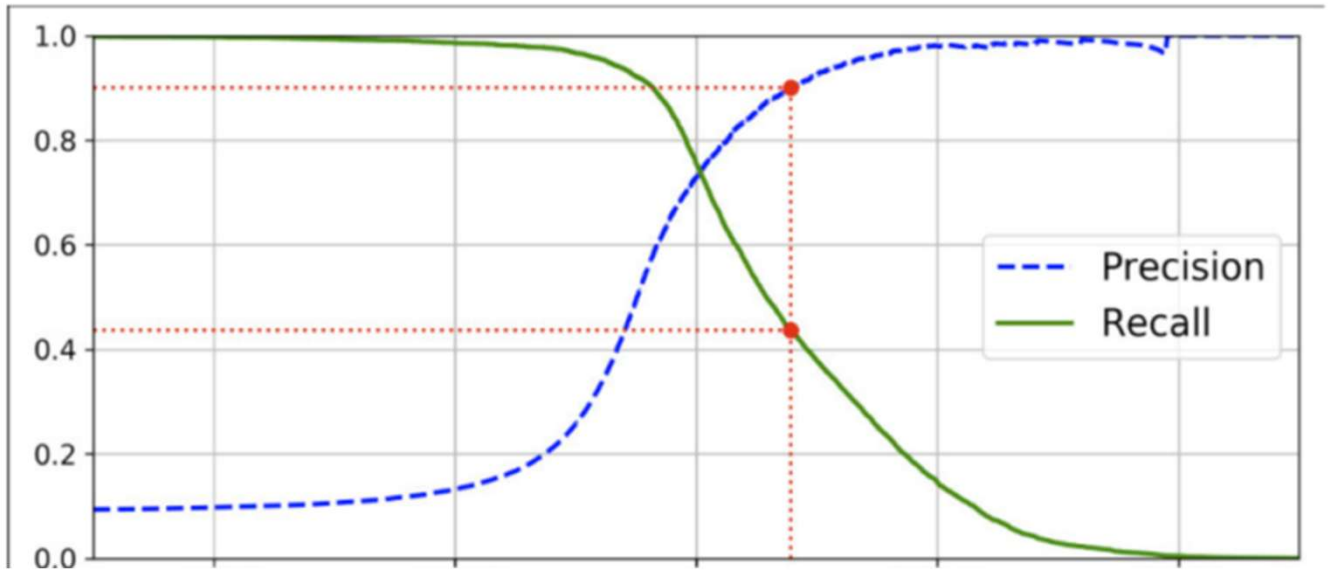
Precision and Recall

- The *precision* and *recall* measures are widely used in classification.
- **Precision**
  - a measure of *exactness* (i.e., what percentage of tuples labeled as positive are actually such)
- **Recall**
  - a measure of *completeness* (what percentage of positive tuples are labeled as such).
  - Same as sensitivity (or the *true positive rate*).

Precision Recall Tradeoff

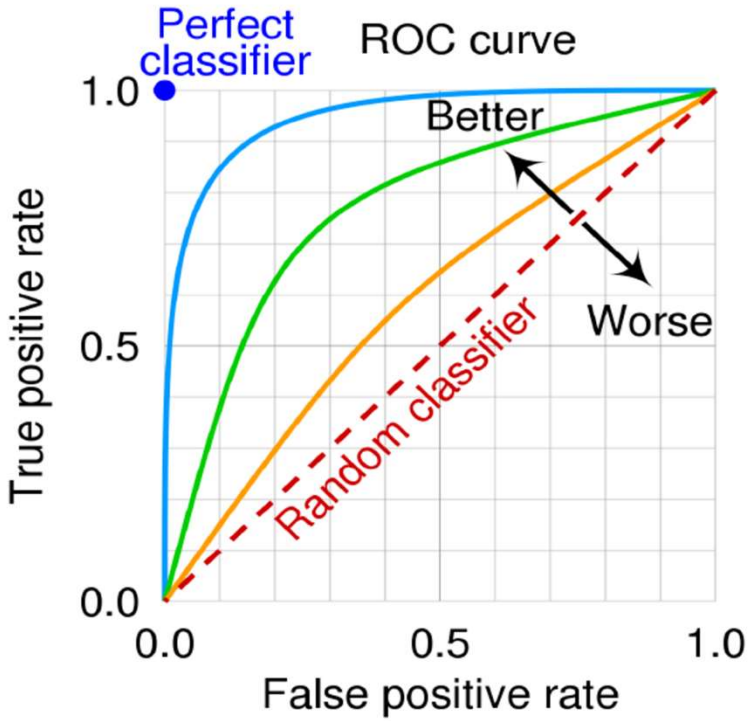
- What a perfect precision score of 1.0 means?
  - for a class *C* means that every tuple that the classifier labeled as belonging to class *C* does indeed belong to class *C* However
  - does not tell anything about the number of class *C* tuples that the classifier mislabeled.
- What a recall score of 1.0 for *C* means ?
  - that every item from class *C* was labeled as such
  - does not tell us how many other tuples were incorrectly labeled as belonging to class *C*.
- There is an inverse relationship between precision and recall
  - For example, medical classifier may achieve high precision by labeling all cancer tuples that present a certain way as *cancer*, but may have low recall if it mislabels many other instances of *cancer* tuples.
- Precision and recall scores are typically used together, where precision values are compared for a fixed value of recall

Precision Recall Trade-off



F measure is the *harmonic mean* of precision and recall

Precision Recall Trade-off

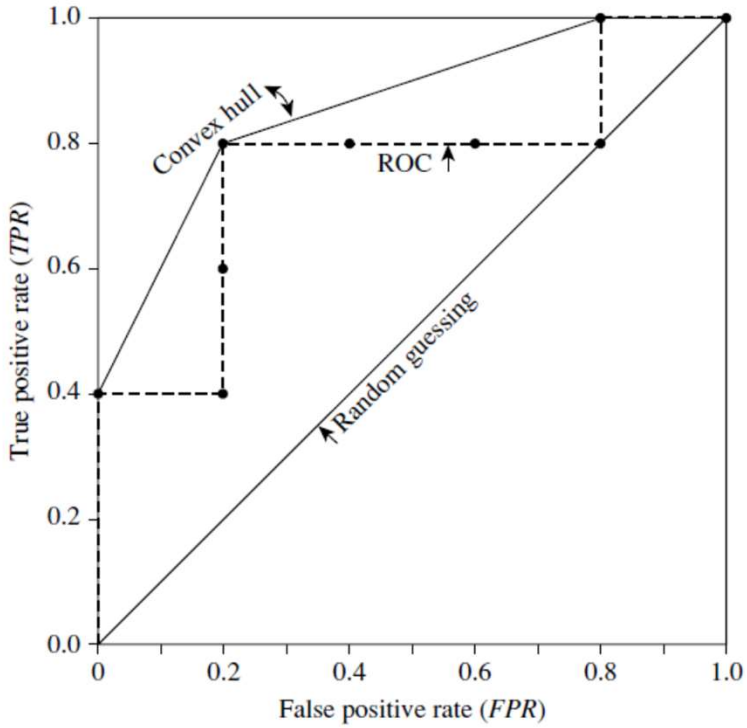


ROC curve

- ROC curve allows us to visualize the trade-off between TPR and FPR
- *TPR* = TP/P and FPR = FP/N
- To plot an ROC curve for a given classification model, *M*, the model must be able to return a probability of the predicted class for each test tuple

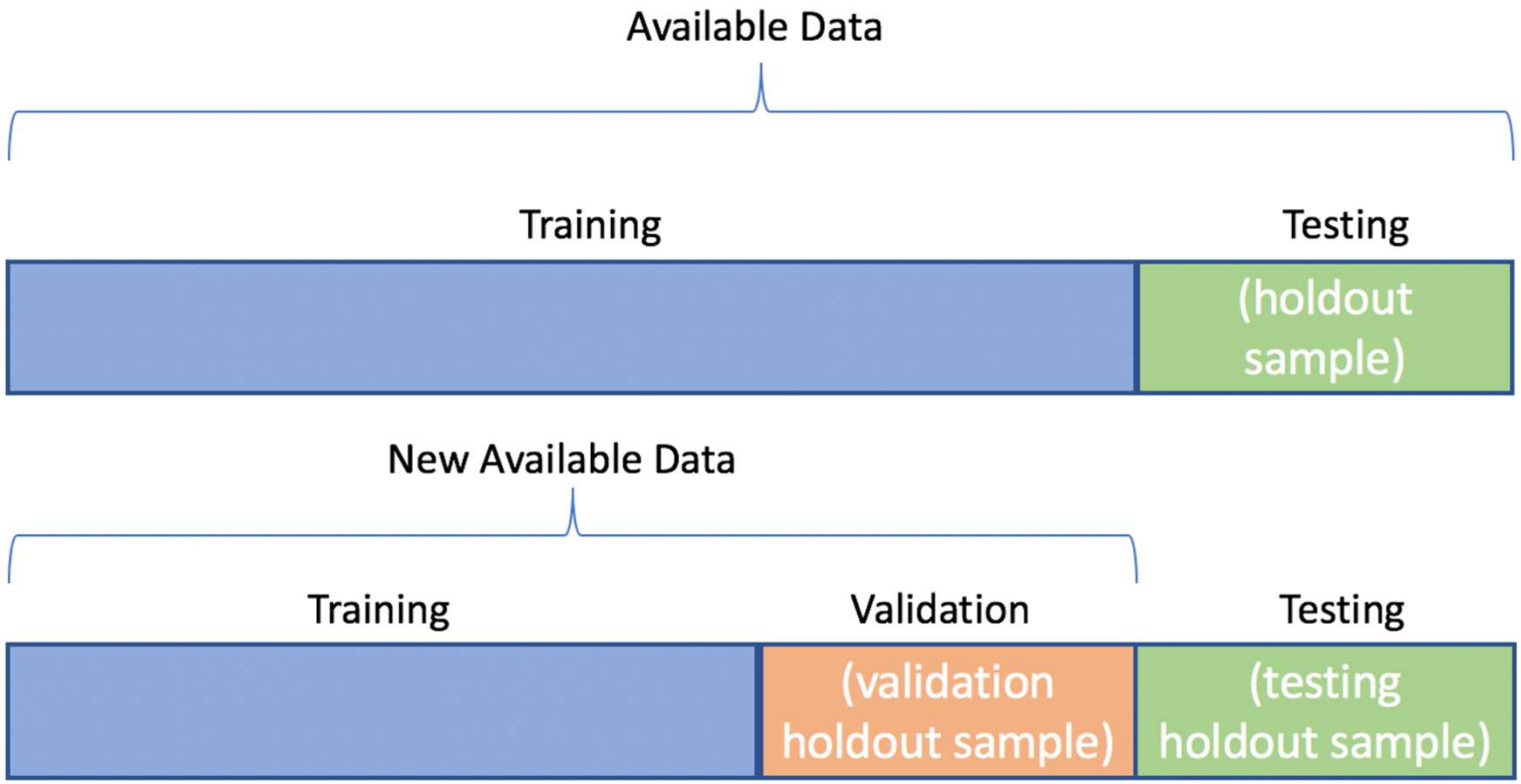
ROC Curve

Tuple #	Class	Prob.	TP	FP	TN	FN	TPR	FPR
1	*P*	0.90	1	0	5	4	0.2	0
2	*P*	0.80	2	0	5	3	0.4	0
3	*N*	0.70	2	1	4	3	0.4	0.2
4	*P*	0.60	3	1	4	2	0.6	0.2
5	*P*	0.55	4	1	4	1	0.8	0.2
6	*N*	0.54	4	2	3	1	0.8	0.4
7	*N*	0.53	4	3	2	1	0.8	0.6
8	*N*	0.51	4	4	1	1	0.8	0.8
9	*P*	0.50	5	4	0	1	1.0	0.8
10	*N*	0.40	5	5	0	0	1.0	1.0



Validation

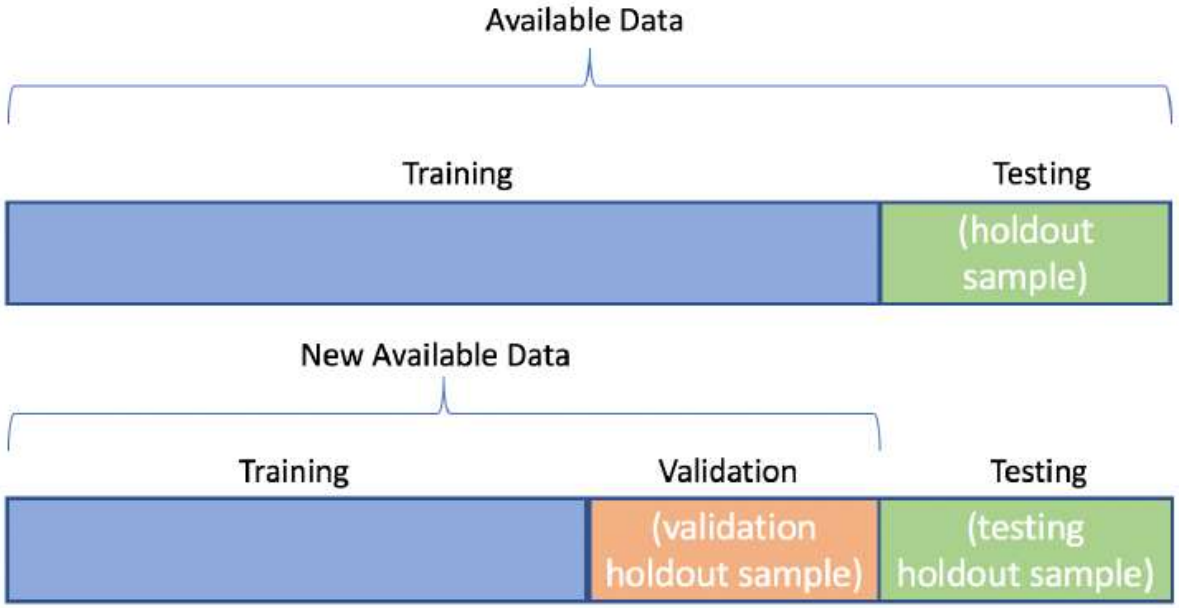
Validation


Available Data
Training
Testing
(holdout sample)
New Available Data
Training
Validation
Testing
(validation holdout sample)
(testing holdout sample)

Type of validation

- Holdout Validation
- K-Fold Cross Validation

Holdout Validation



Pros

- Fully independent of data
- Lower computational costs

Cons

- higher variance

K-Fold Cross Validation