Tufts

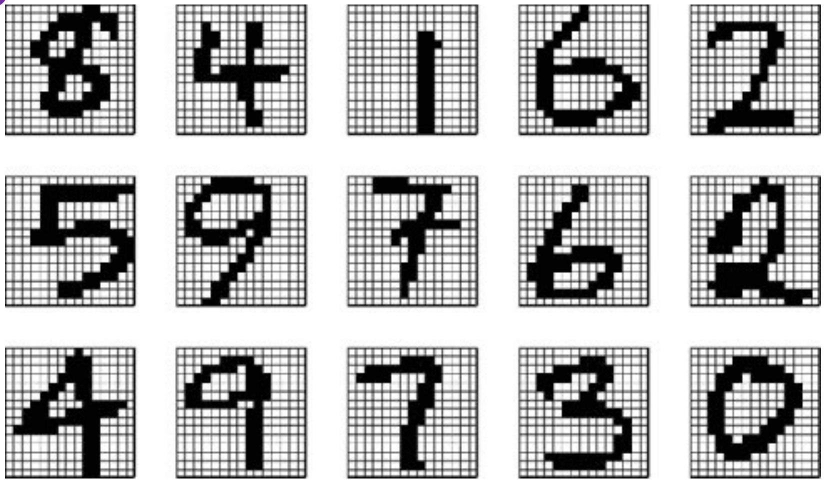
CS135
Introduction to Machine Learning

Lecture 8: Metrics; Evaluating Performance

How do we acquire knowledge?

- Explicitly (Rule-based)
  - When you learn a second language you receive a grammar book with lots of rules and have vocabulary lessons.

- By learning



Types of learning

- Supervised – Making predictions from **labeled** examples

  Classification: Finite labels (predict gender, predict color)

  Regression: Ordered, continuous labels (predict height)

- Unsupervised – Detecting patterns from **unlabeled** examples

- Semi-supervised – Learning from labeled and unlabeled

Memorization and generalization

- Memorization:
  - Are we specifically only learning an algorithm on our data?
- Generalization
  - Refers to the capability of applying learned knowledge to previously unseen data
  - Without generalization there is no learning, just memorizing!

Features and labels

- **Features** (a.k.a covariates) are the variables that describe the data.

- **Label** (target variable/target covariate): The feature we want to predict in supervised learning.
  - So, a label is kind of like a feature

Supervised vs unsupervised examples

- Supervised - Predict housing prices using:
  - Features: number of bathrooms, floor, number of windows, kitchen size
  - Label: House prices

    *** We can switch around the label and the features

- Unsupervised - Find patterns in books using:
  - # of pages, # of printed copies, language, mean words per page
  - Label: there is none
    - Maybe we will ‘cluster’ the books into genres? Maybe into authors? Etc.

More examples of supervised vs unsupervised

Supervised:

1. Regression: Predict housing prices given the prices of 1,000 previously sold houses, with various **features** describing the houses (size, # of windows, …)

2. Classification: Predict if a gift is a book, a toy, or something else (three finite options only)

Unsupervised

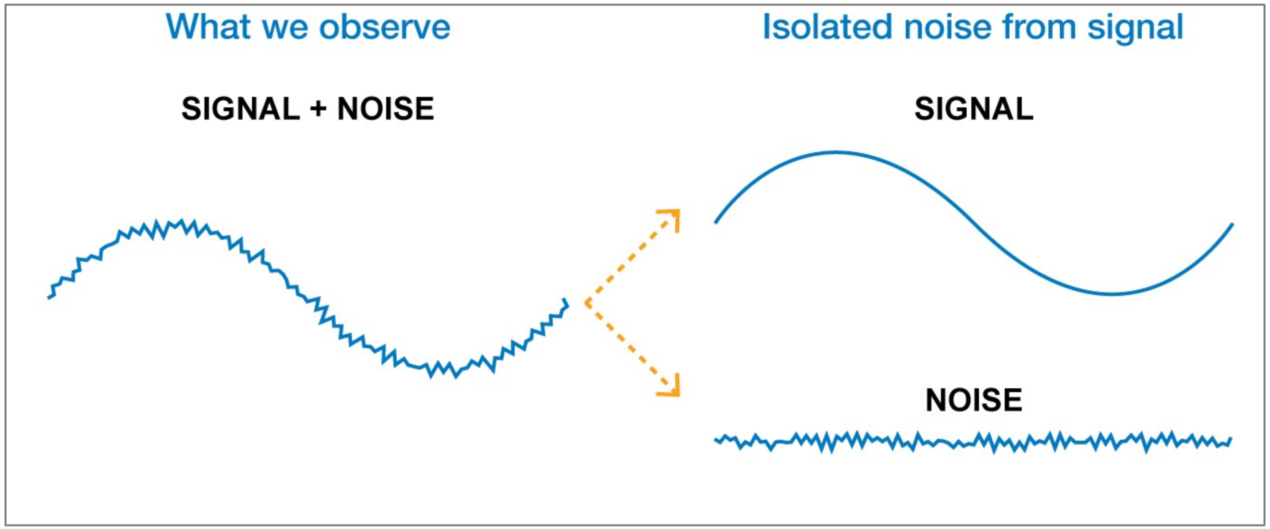
3. Get data on sports players and see if there are specific patterns.
   - Maybe the data has 3 'clusters' that mean something– basketball players, soccer players, and sumo wrestlers?
   - Maybe it clusters into categories nobody has thought about?

What's in the data we learn from

The data usually has two components:

- Signal: information that is relevant to pattern detection or prediction
- Noise: information that is irrelevant to the future

Finding which is which is the challenge



Why do we evaluate a model’s performance?

1. To find the best preforming models

2. Part of parameter tuning

3. To report/publish results

4. To make research/business decisions
   - Can the model be used as is?
   - Do we need to try to improve?

Performance metrics

- Supervised
  - **Classification:**
    - Accuracy
    - Precision & Recall
    - ROC curves & AUC
  - **Regression:**
    - Mean square error (MSE) + Root MSE
    - Percent error
    - Mean absolute percent error
  - **Ranking (ordinal/discrete regression):**
    - Precision at N

- Unsupervised

Classification: Precision and recall

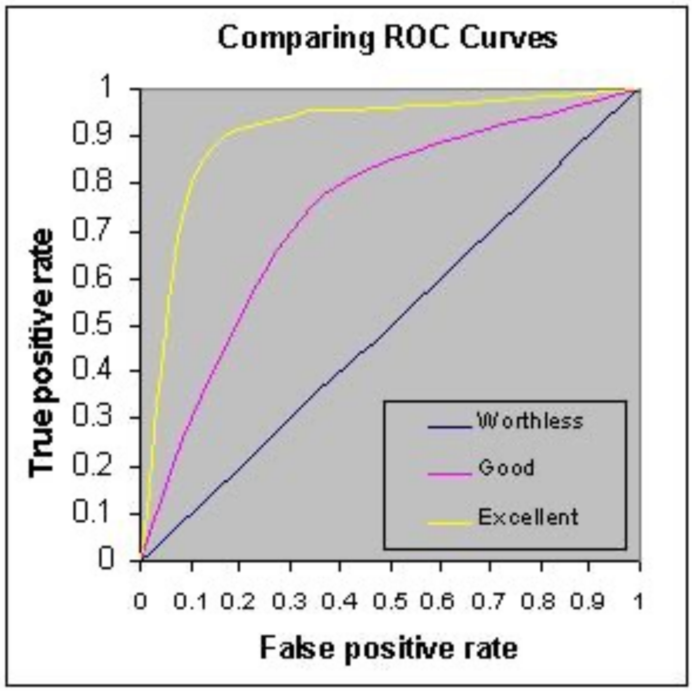
- Precision (a.k.a PPV): What percent of our predictions are correct?
- Recall (a.k.a sensitivity): What percent of the accurate predictions did we capture?
- F1 score: A single number that combines the two values above. Good for ranking/sorting, and imbalanced classes

$$\text{F1} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

- Accuracy: What percent of all our predictions (positive and negative) are correct?

Tufts

Classification: ROC curve

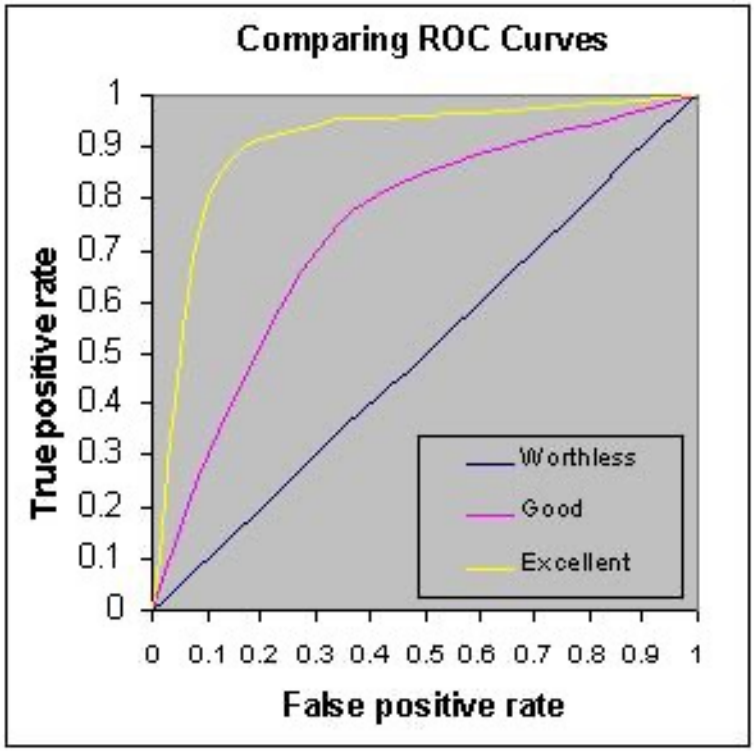


Area under curve (the ROC curve)

A Receiver Characteristic Curve (ROC) plots the True positive rate (TPR) vs. the False positive rate (FPR). The maximum area under the curve (AUC) is 1. Completely random predictions have an AUC of 0.5. The advantage of this metric is that it is continuous.

AUC interpretation

- The AUC is equal to the probability that a random positive example will be ranked above a random negative example.

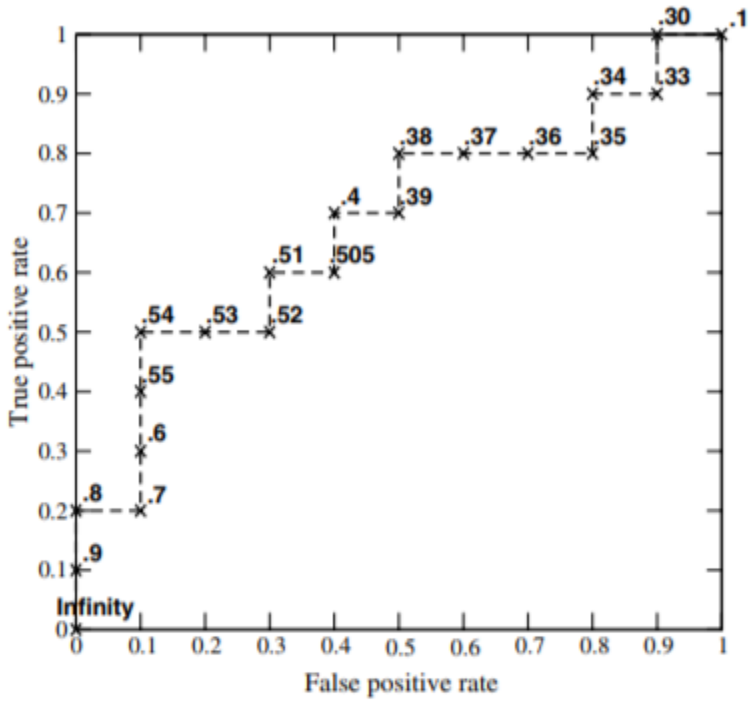


Constructing a ROC curve

Sort predictions

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

Plot



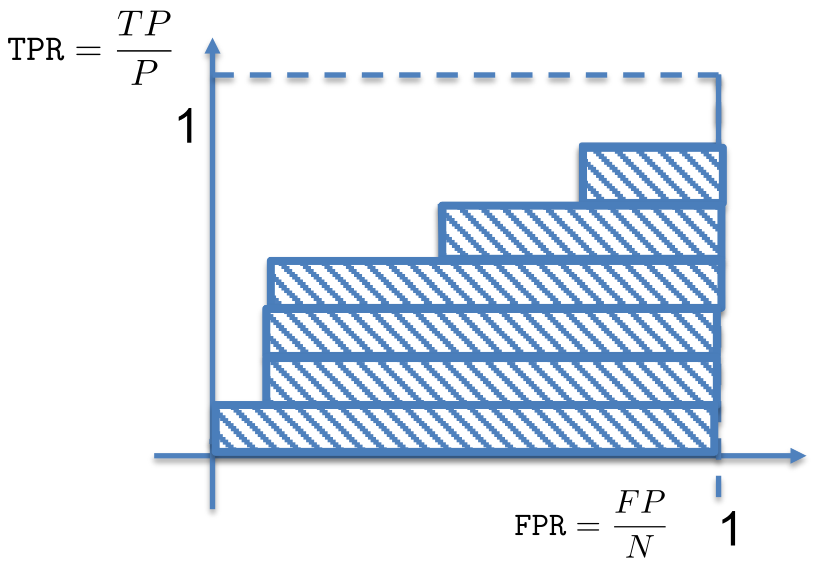
Tufts

Computing the AUC

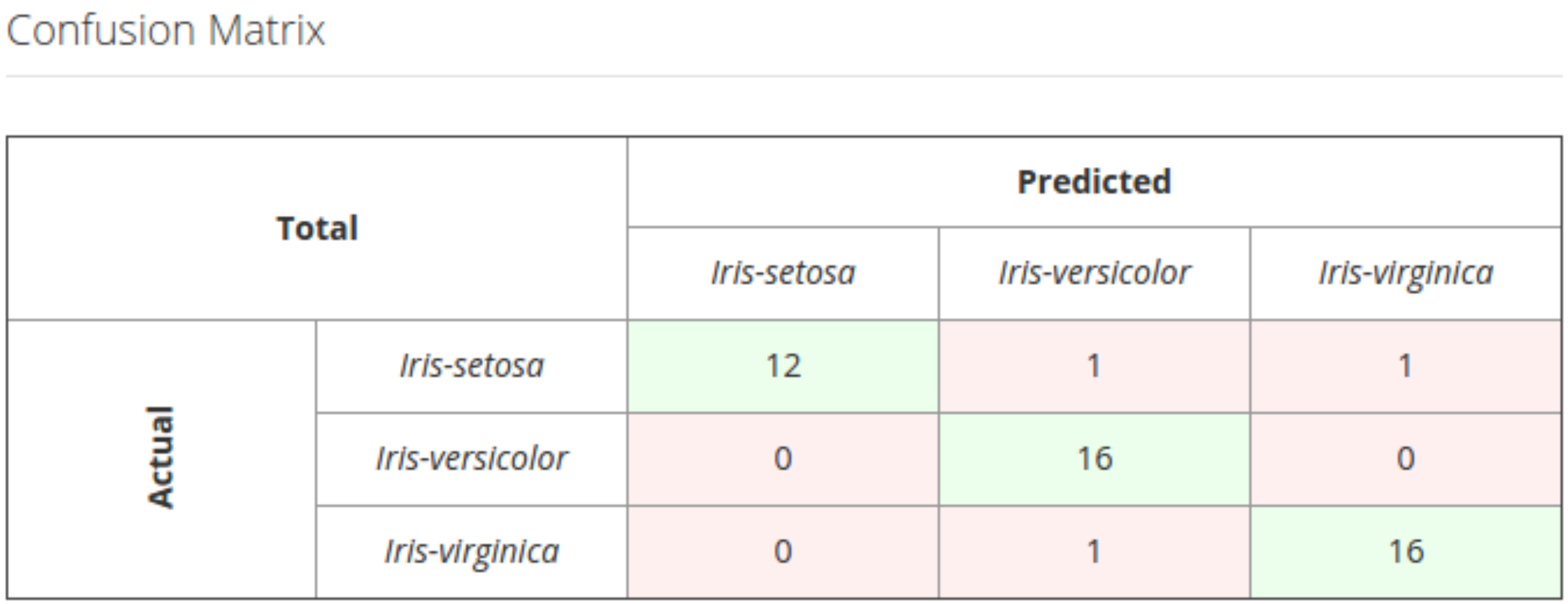
```
def AUC(scores,labels):
P = len( [x for x in labels if x >0  ])
     N = len( [x for x in labels if x <=0  ])
     zipped = sorted(zip(scores,labels))
     auc = 0.0
     count = 0.0
     for score,label in zipped:
          if label < 0:
                count += 1.0/N
          else:
                auc += count/P
     return auc

if __name__=="__main__":
    scores = [1,3,2,5,4]
    labels = [-1,-1,+1,+1,+1]
    print AUC(scores,labels)

0.833333333333
```



Classification: Confusion matrix



Confusion Matrix

Total		**Predicted**		
		Iris-setosa	*Iris-versicolor*	*Iris-virginica*
Actual	*Iris-setosa*	12	1	1
	Iris-versicolor	0	16	0
	Iris-virginica	0	1	16

- Used in classification

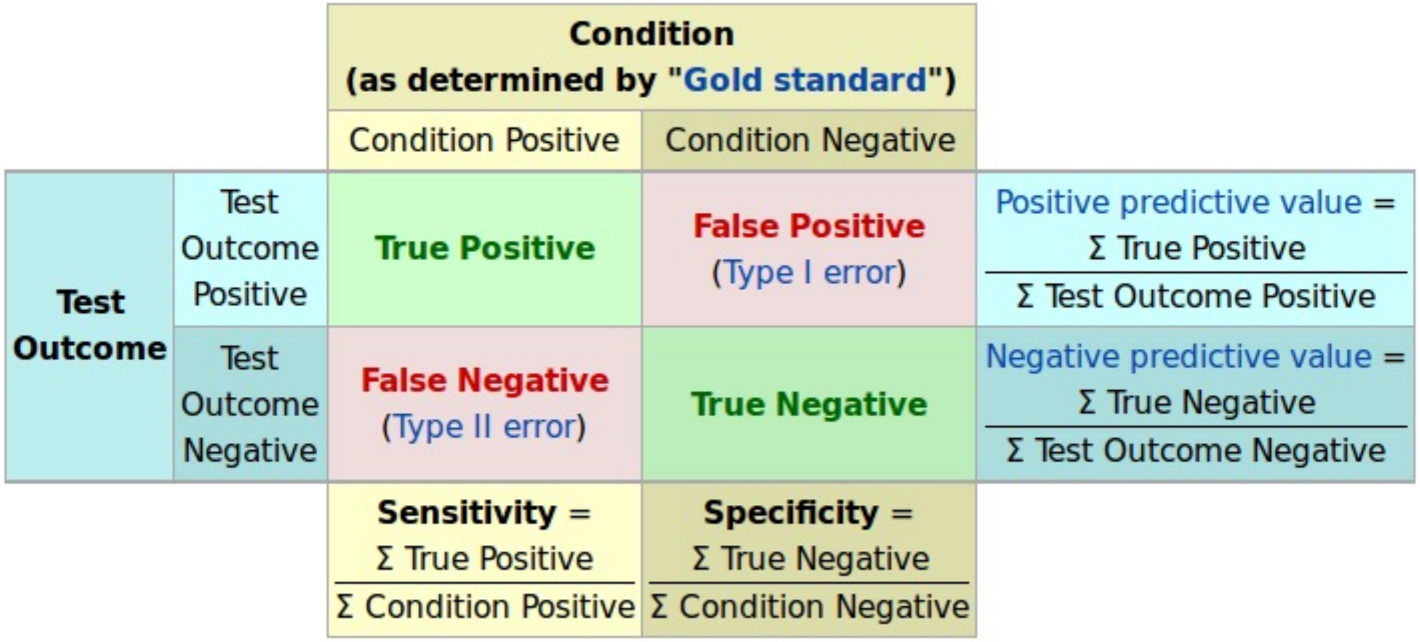
- Show Actual vs predicted results

- Enables visualizing performance and calculating performance metrics

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

The confusion matrix – list of statistical metrics that can be calculated

- Many performance terms: Precision, recall, sensitivity, specific, true negative rate, true positive rate, PPV, NPV, type 1 error, type 2 error



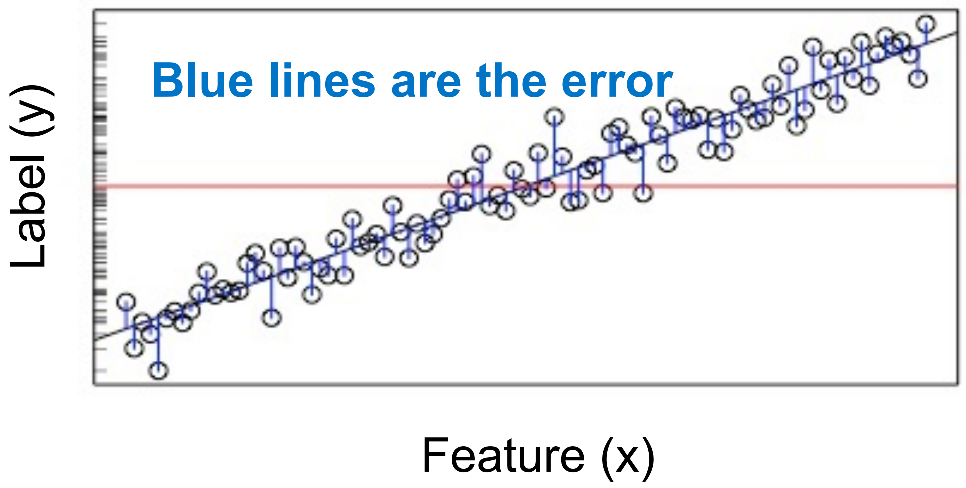
		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

Ranking – precision at N

- Precision at N
  - How many accurate examples did we capture in our top N ranked examples. This is often used in information (document) retrieval

Regression – performance metrics

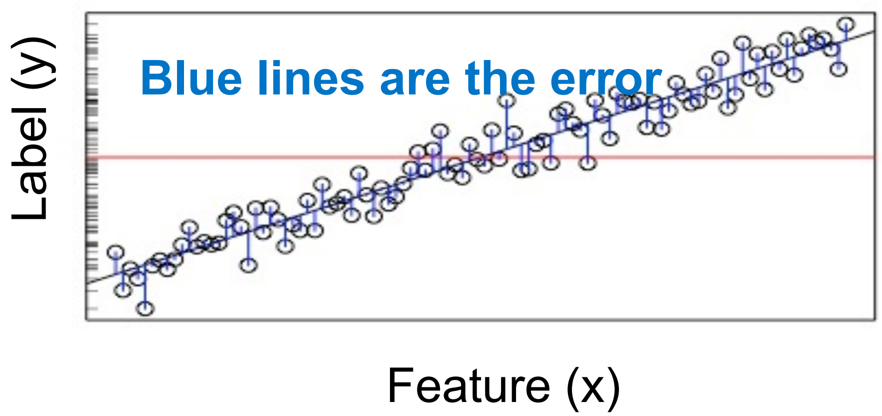


For every example, we have:

- Error $=(y_{predicted} - y_{true})$
- Percent error $=(y_{predicted} - y_{true})/y_{true}$

Now lets aggregate the errors and get a single value (next slide)

Regression – performance metrics



- Mean square error (MSE) and Root MSE (RMSE):
  - MSE = $\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$
  - Gives high weight to outliers, most common metric

- MAPE: Mean absolute percent error
  - Percentages are easy to understand. Simply the mean of the lPercent errorl

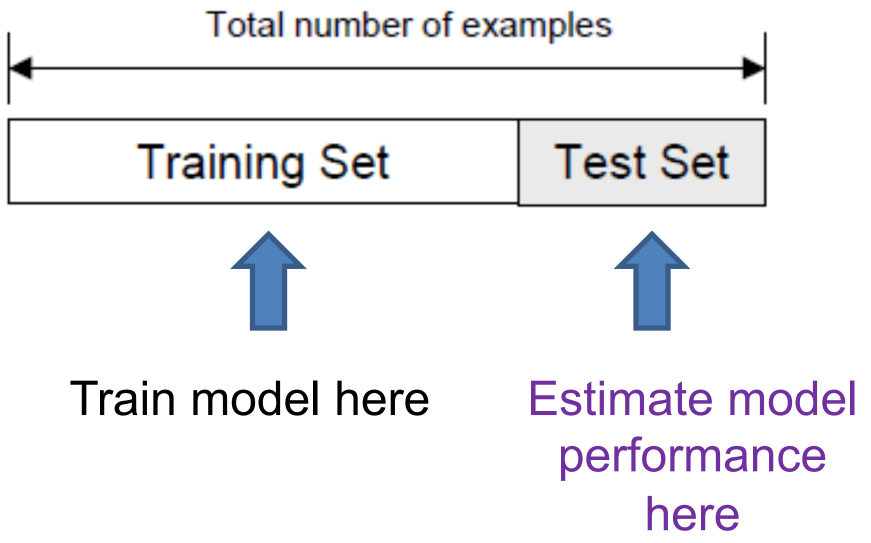
Tufts

The basic rule in model validation

Don’t test your model on data it’s been trained with!

Holdout test set: The naïve approach

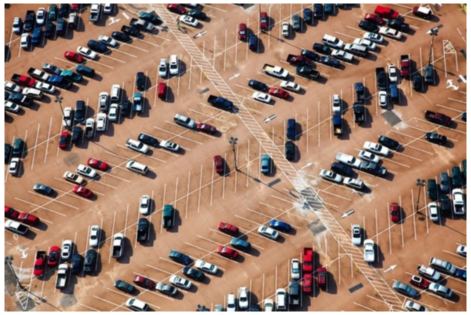
- Randomly split the entire dataset into:
  - Training set: A dataset used for training the model
  - Test set (a.k.a validation set): Data only used for testing the model



Holdout test set – considerations

- Can the data be split randomly without bias?
  - Test set and training set should be interchangeable, except for data size. Otherwise test results are not a true reflection of performance.
  - In other words, test and train data should be from the same distribution
- Do we have enough data?
  - Example: Predict if a car is American or Japanese based on 50 examples?

    Will the test set be big enough to accurately report our model performance?



What is biased/unbiased data?

- Unbiased data sampling:
  - Data that represents the diversity in our target distribution of data on which we want to make predictions.
  - I.e., a random, representative sample of the entire distribution.

- Biased data:
  - Data that does not accurately reflect the target distribution.
  - Examples:
    - **Elections**: we only asked people with glasses who they will vote for, instead of asking a random sample of the population.
    - **Simulated data**: We cannot get real data, so we create simulated/synthetic data
  - May be addressed using transfer learning/domain adaptation

The three-way split

- **Training set**

  A set of examples used for learning

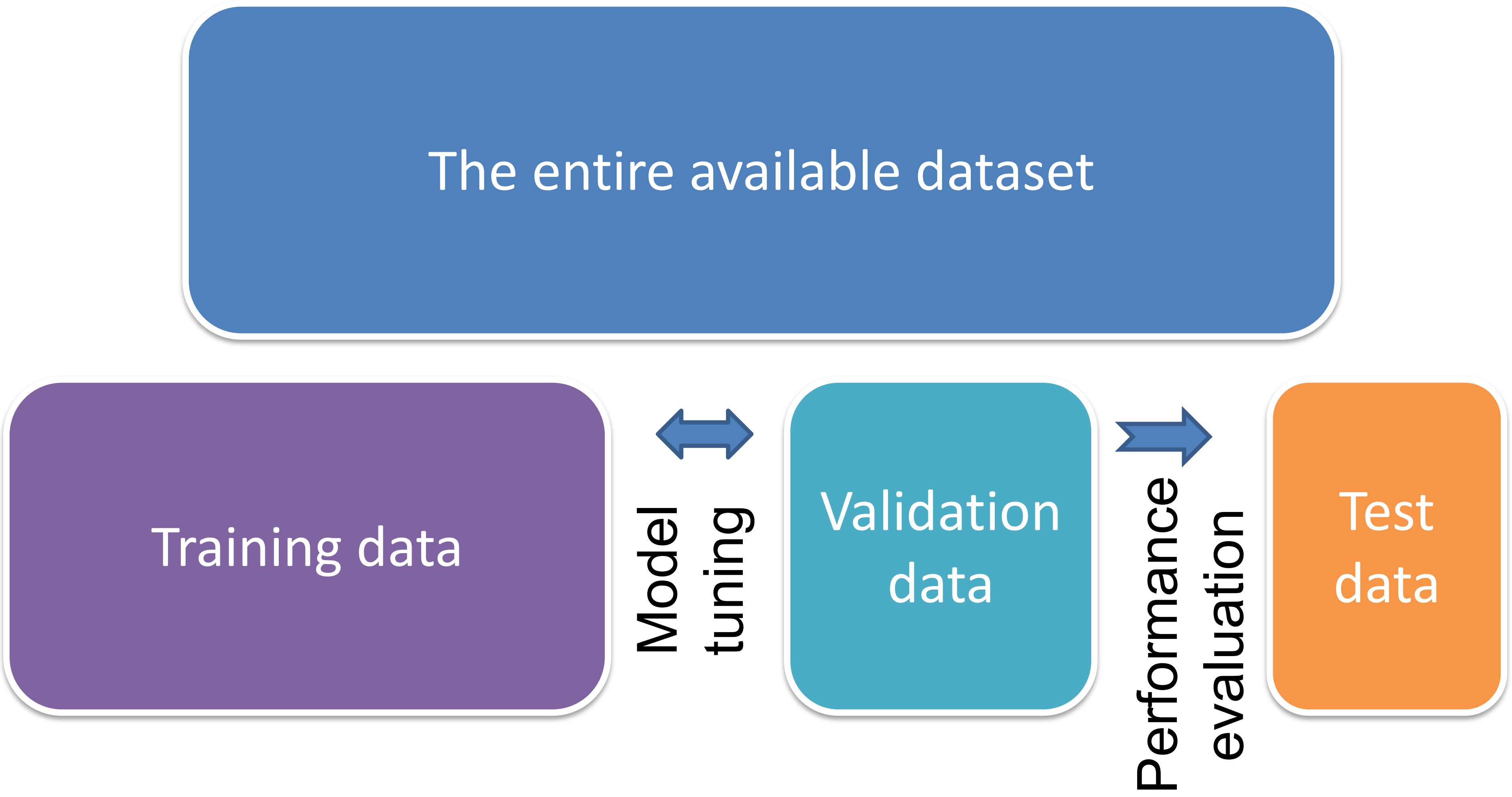
- **Validation (a.k.a development) set**

  A set of examples used to tune the parameters of a classifier

- **Test set**

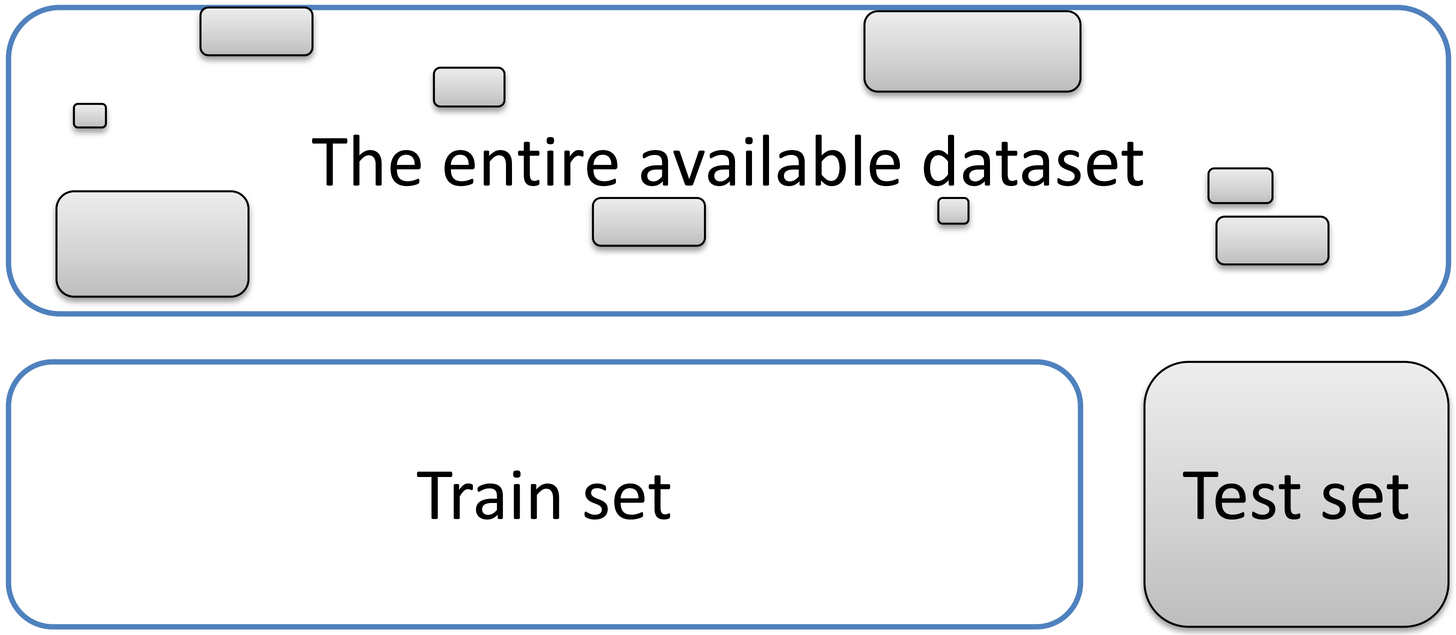
  A set of examples used only to assess the performance of fully-trained classifier. After assessing the model with the test set, YOU *MUST NOT* further tune your model (that's the theory anyway… - in order to prevent 'learning the test set' and 'overfitting')

The three-way split



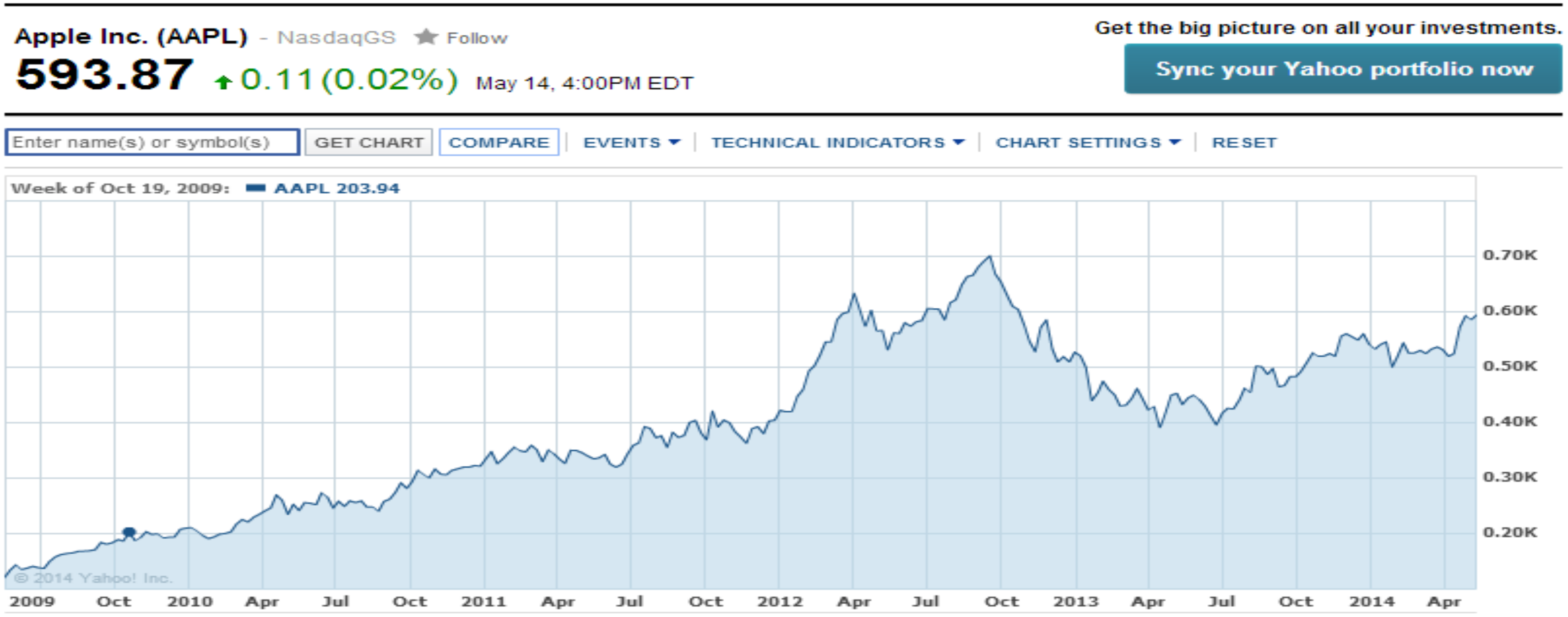
How to perform the split?

- How many examples in each data set?
  - Training: Typically 60-80% of data
  - Test set: Typically 20-30% of your trained set
  - Validation set: Often 20% of data
- Examples
  - 3 way: Training: 60%, CV: 20%, Test: 20%
  - 2 ways: Training 70%, Test: 30%



How to perform the split?

- Time based – If time is important



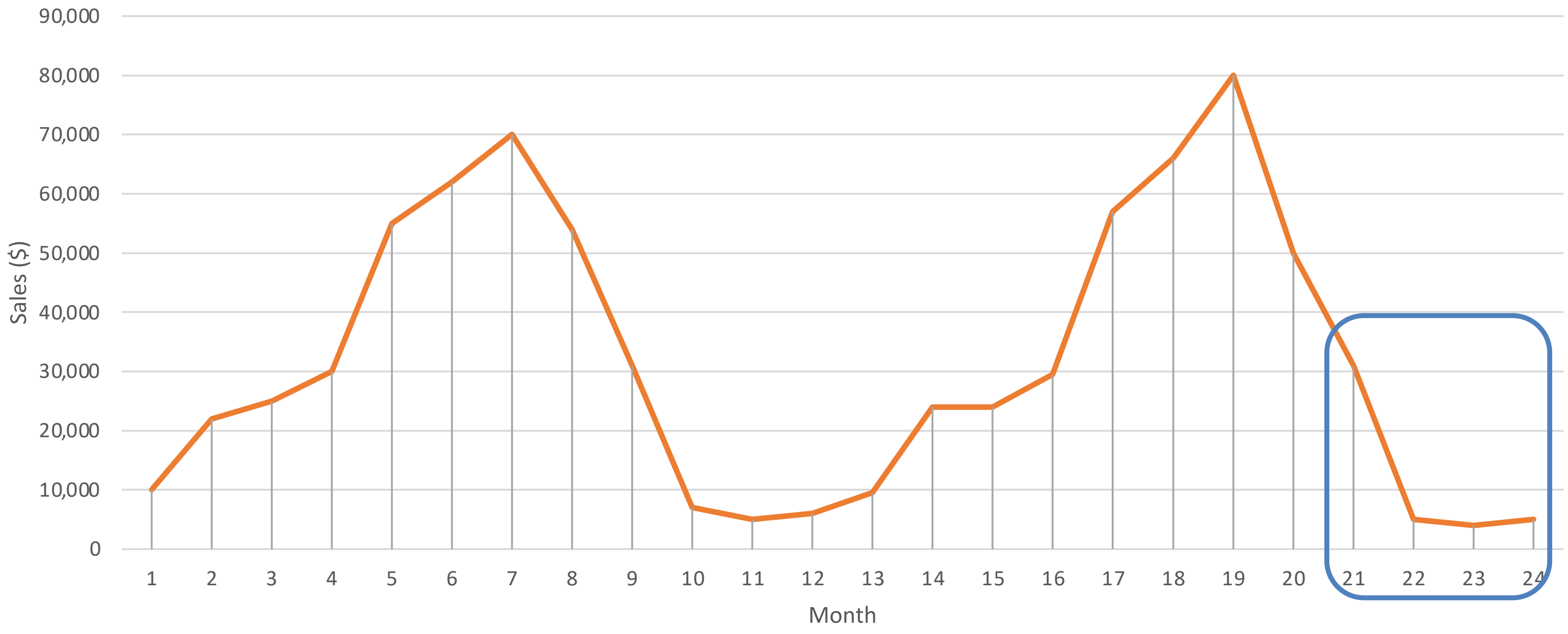
Apple Inc. (AAPL) - NasdaqGS
Follow
593.87 0.11 (0.02%) May 14, 4:00PM EDT
Get the big picture on all your investments.
Sync your Yahoo portfolio now
Enter name(s) or symbol(s)
GET CHART
COMPARE
EVENTS
TECHNICAL INDICATORS
CHART SETTINGS
RESET
Week of Oct 19, 2009: AAPL 203.94
0.70K
0.60K
0.50K
0.40K
0.30K
0.20K
© 2014 Yahoo! Inc.
2009 Oct 2010 Apr Jul Oct 2011 Apr Jul Oct 2012 Apr Jul Oct 2013 Apr Jul Oct 2014 Apr

Can you use Jul 2011 stock price to try and predict Apr 2011 stock price?

Tufts

How to perform the split?

- Some datasets are effected by seasonality



Is it a good test set?

Tufts

Holdout summary

- Good news:
  - Intuitive
  - Usually easy to perform
  - Considered the ideal method for evaluation

- Drawbacks:
  - In small datasets you don’t have the luxury of setting aside a portion of your data
  - The performance will be misleading if we had unfortunate split
  - Often requires planning ahead of time to prevent training on test set or getting insights from it)
  - External evaluation is ideal

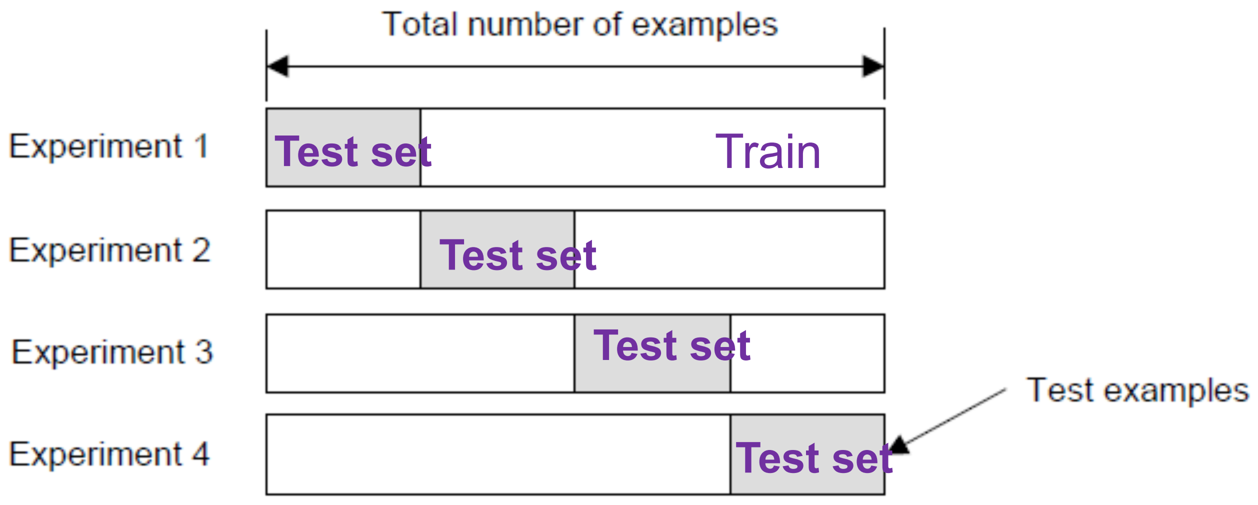
Cross validation

- Used when:
  - a hold-out plan is difficult to design or was not designed at the beginning
  - the dataset is small, and you want to use all the data for good model training

Cross - Validation

- Create a K-fold partition of the dataset
  - For each of the K experiments, use K-1 folds for training and the remaining one for testing

4-fold cross validation example



Tufts

Cross - Validation

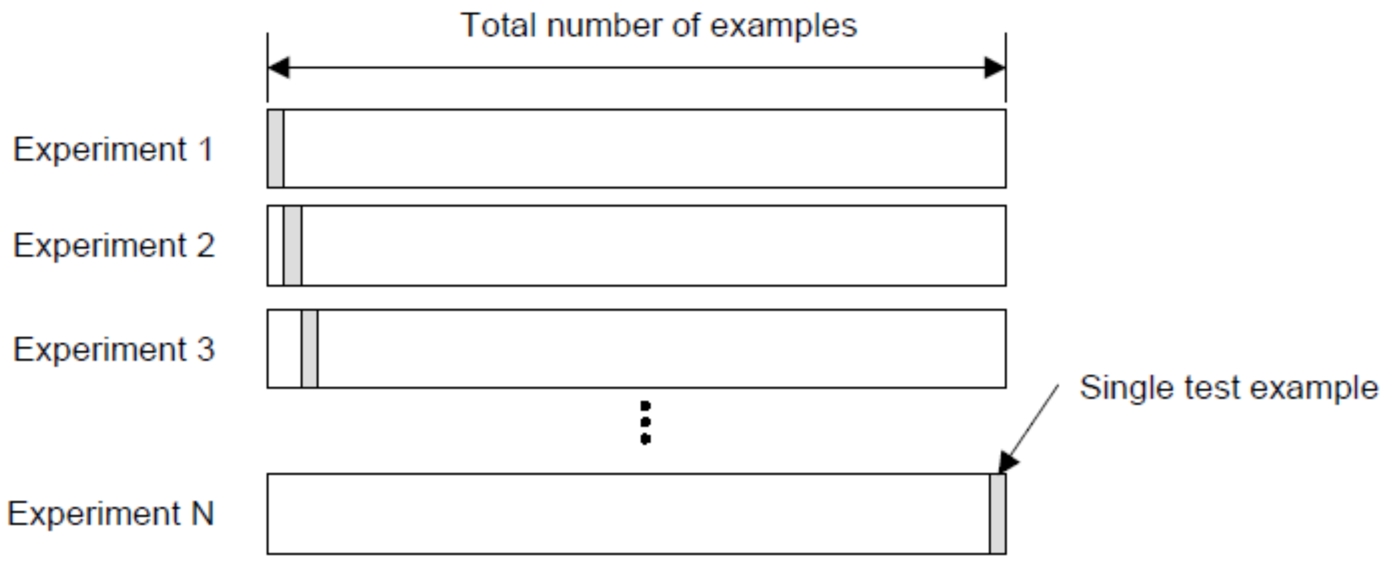
- How do you summarize the performance?
  - **Average:** Usually average of performance between experiments

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

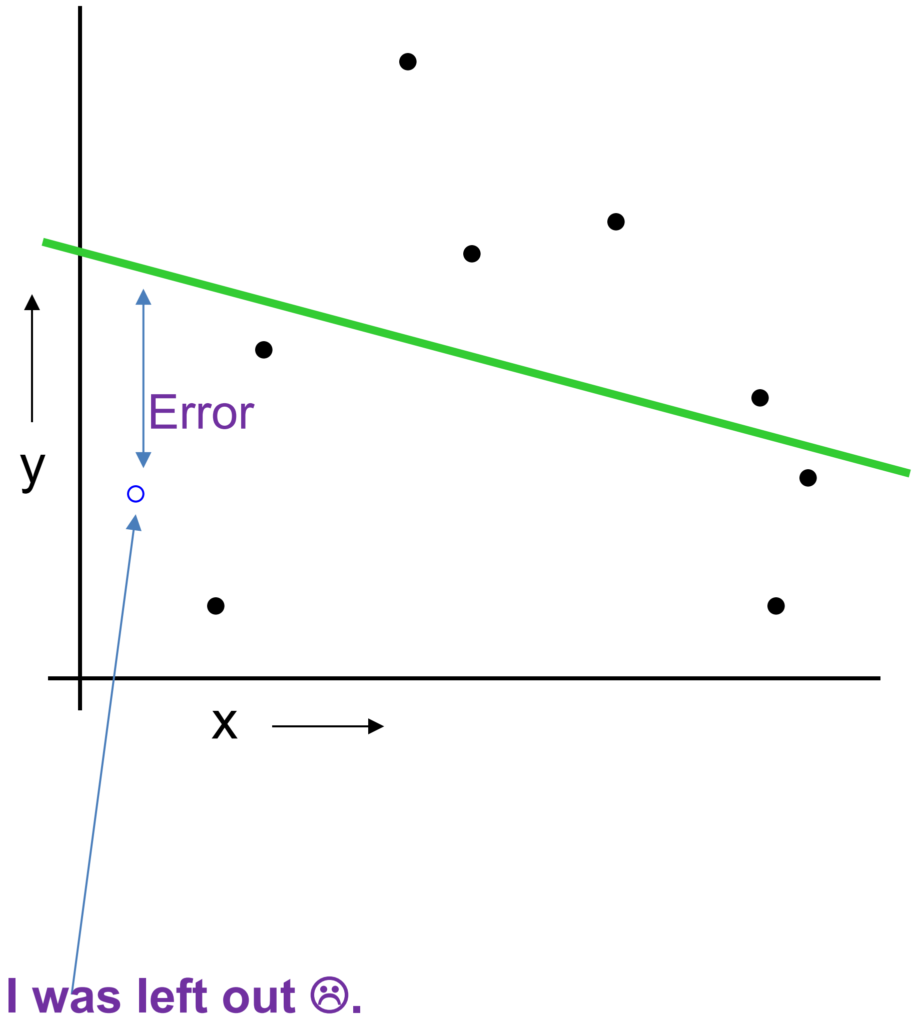
- How many folds are needed?
  - Common choice: 5-fold or 10-fold cross validation (probably since these numbers sound nice)
  - Large datasets → even a 3-Fold cross validation will be quite accurate
  - Smaller datasets → we will prefer bigger K
  - Leave-One-Out approach (K=n). In this case the number of folds K is equal to the number of examples n. Used for smaller datasets

Leave-One-Out Cross Validation (LOOCV)

- Leave-One-Out is the degenerate case of K-Fold cross validation, where K is chosen as the total number of examples



LOOCV in action

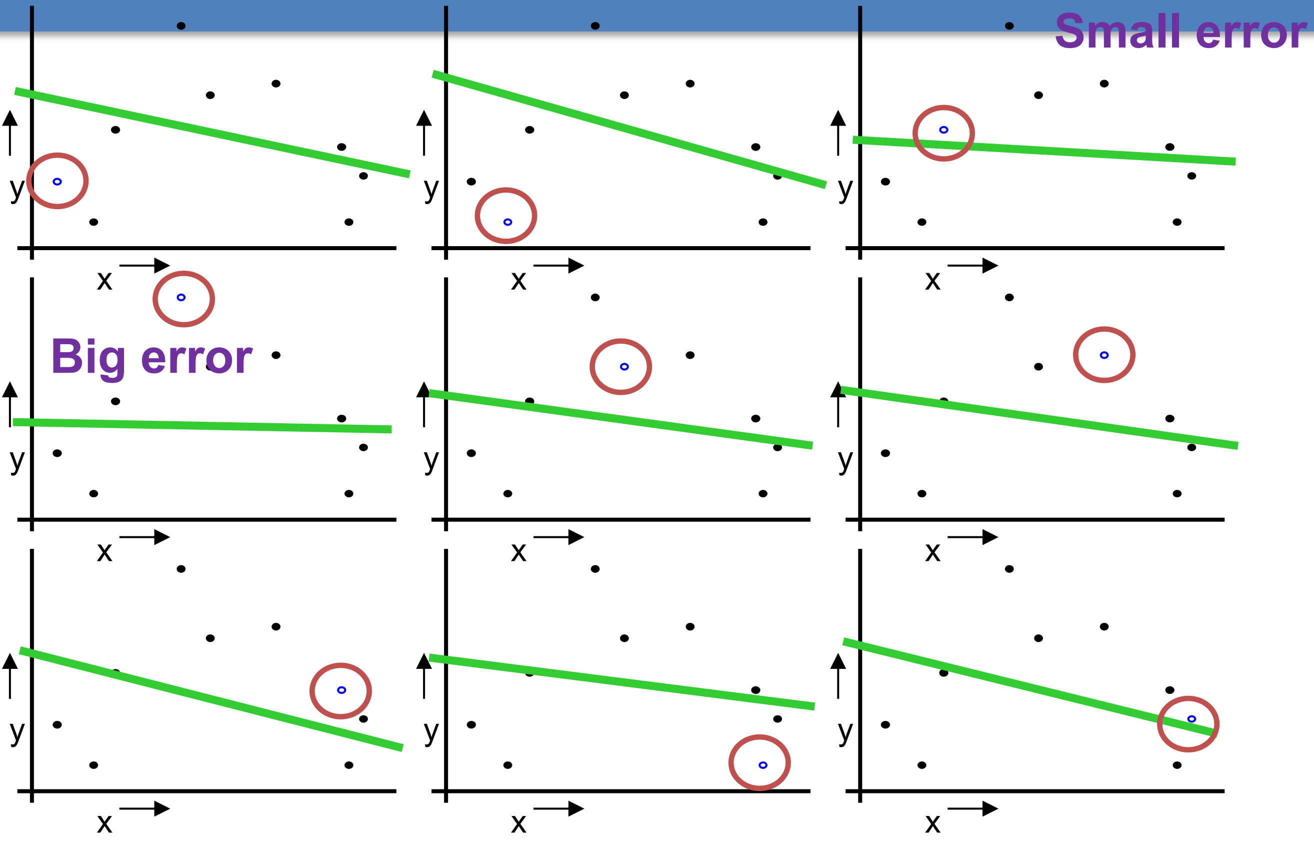


For k=1 to R, do:

1. *Let (x_k,y_k) be the k'th record*
2. *Remove (x_k,y_k) from the dataset*
3. *Train on the remaining R-1 datapoints*
4. *Measure your error (x_k,y_k)*

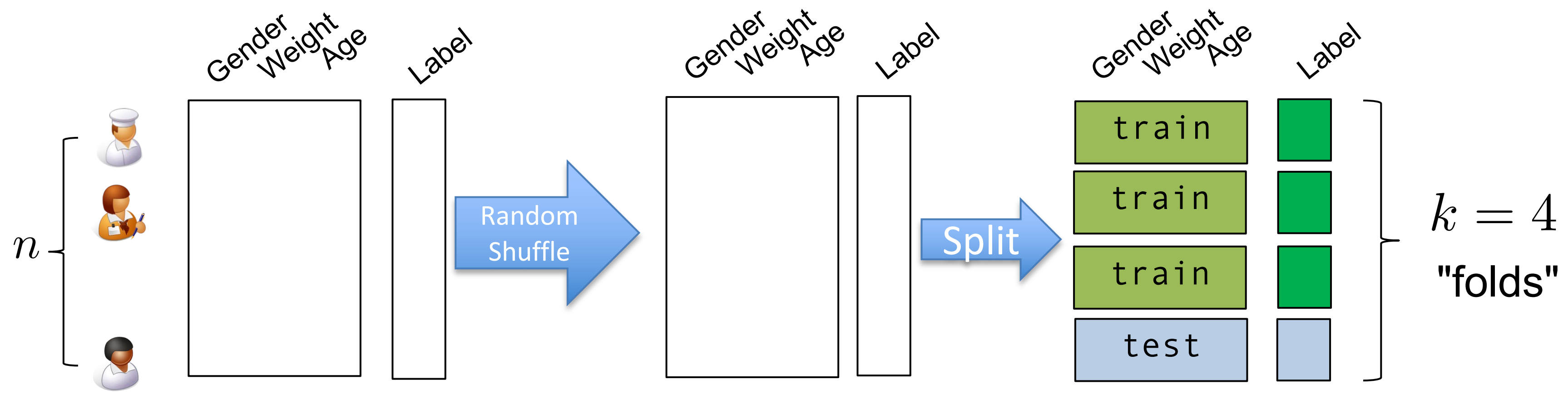
When you've done all points, report the mean error.

LOOCV in action



$MSE_{LOOCV} = 2.12$

Prediction Quality

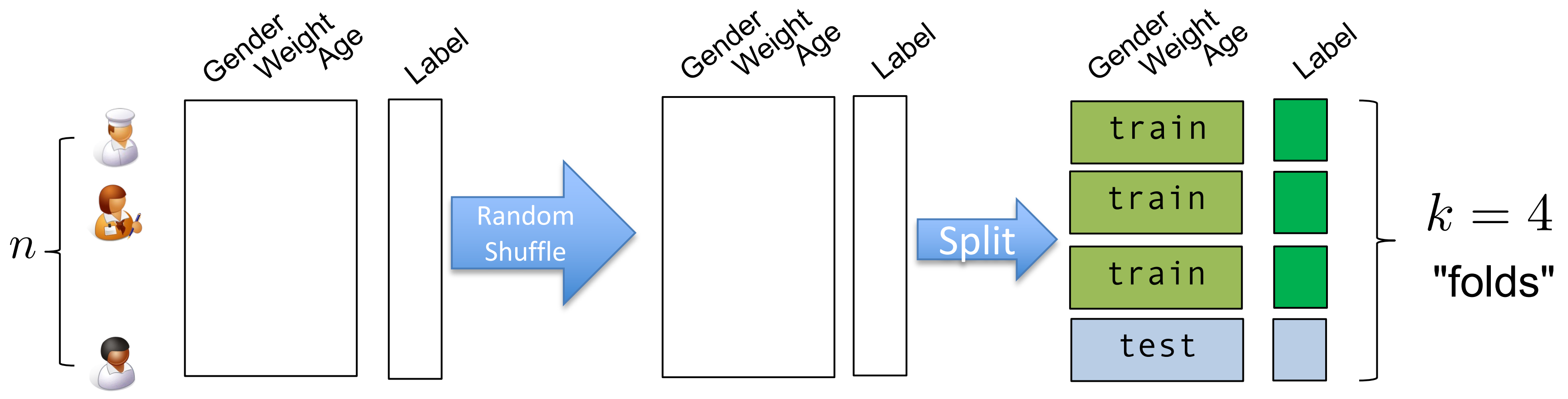


- ❑ How to test quality of trained model $\hat{\beta}$?

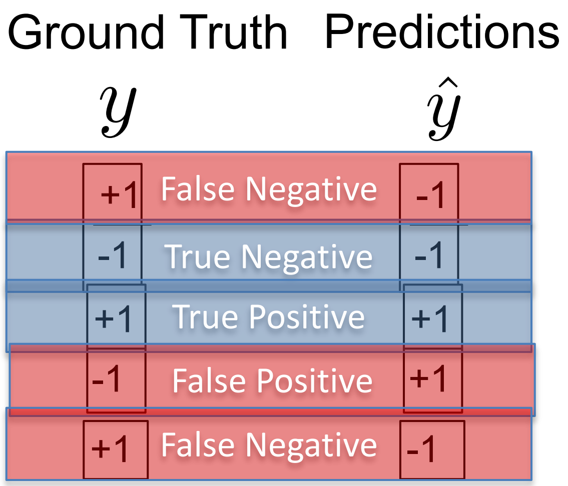
- ❑ Loss: $$L_{\text{test}}(\hat{\beta}) = \sum_{i=1}^{n} \ell(\hat{\beta}; y_i, x_i)$$

non-intuitive

Accuracy



- How to test quality of trained model $\hat{\beta}$?
- Accuracy: $\mathrm{ACC} = \dfrac{TP + TN}{P + N}$

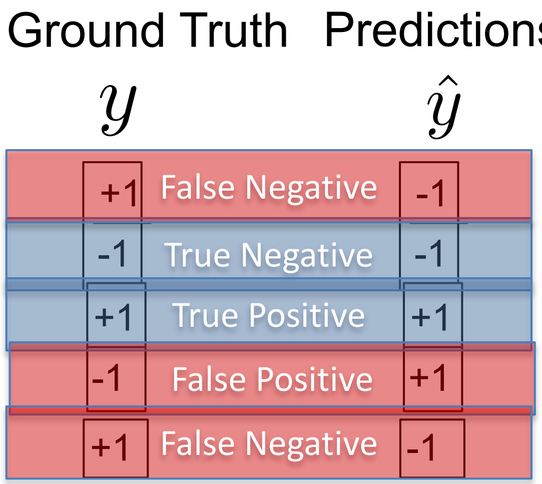


Ground Truth
Predictions

y		$\hat{y}$
+1	False Negative	-1
-1	True Negative	-1
+1	True Positive	+1
-1	False Positive	+1
+1	False Negative	-1

Accuracy

- Accuracy: $\mathrm{ACC} = \dfrac{TP + TN}{P + N}$



Q: Is a classifier with 0.99999 accuracy good?

Depends on how imbalanced dataset is (e.g., -1 is 0.00000001%)

Depends on relative value/cost of TP,TN,FP,FN

Additional Metrics

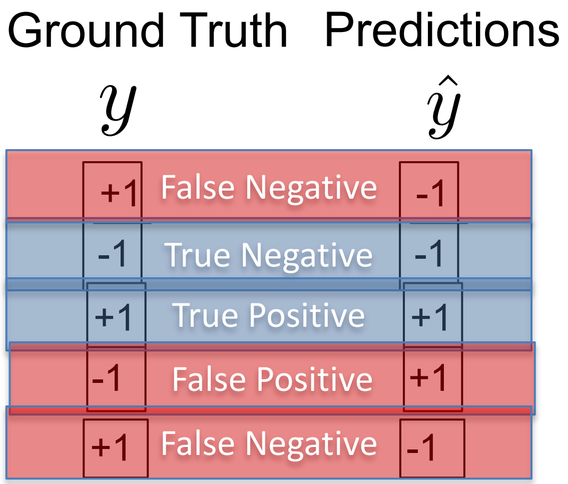
- ❑ Accuracy: $\mathtt{ACC} = \dfrac{TP + TN}{P + N}$
- ❑ True Positive Rate (Sensitivity):

$$\mathtt{TPR} = \frac{TP}{P} \quad \text{(close to 1 is good)}$$

- ❑ False Positive Rate (Fallout):

$$\mathtt{FPR} = \frac{FP}{N} \quad \text{(close to 0 is good)}$$

- ❑ True Negative Rate (Specificity): $\mathtt{TNR} = \dfrac{TN}{N} = 1 - \mathtt{FPR}$
- ❑ False Negative Rate (Miss Rate): $\mathtt{FNR} = \dfrac{FN}{P} = 1 - \mathtt{TPR}$



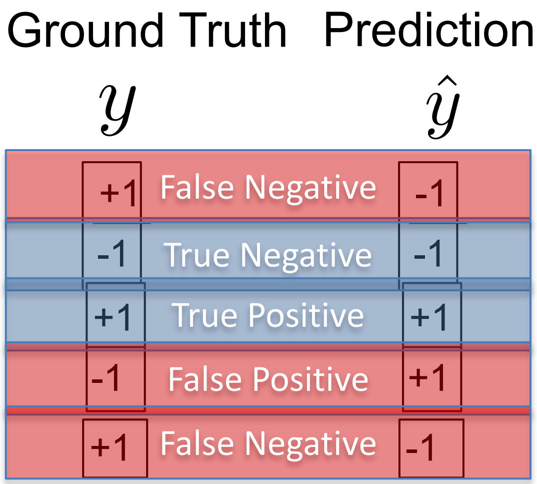
Tradeoff Between TPR and FPR

- True Positive Rate (Sensitivity):

$$\text{TPR} = \frac{TP}{P} \quad \text{(close to 1 is good)}$$

- False Positive Rate (Fallout):

$$\text{FPR} = \frac{FP}{N} \quad \text{(close to 0 is good)}$$



Ground Truth y		Predictions $\hat{y}$
+1	False Negative	-1
-1	True Negative	-1
+1	True Positive	+1
-1	False Positive	+1
+1	False Negative	-1

Predict **all** $i \in \texttt{test}$ to be **positive**: TPR =1 (good) FPR= 1 (bad)

Predict **all** $i \in \texttt{test}$ to be **negative**: TPR =0 (bad) FPR= 0 (good)

True irrespective of label imbalance in test set

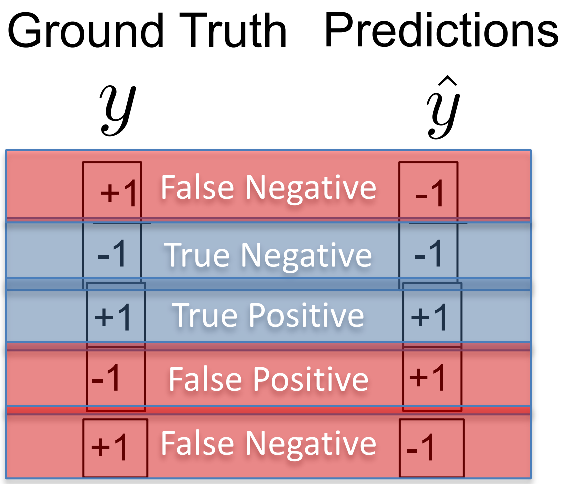
Tradeoff Between TPR and FPR

- True Positive Rate (Sensitivity):

$$\text{TPR} = \frac{TP}{P} \quad \text{(close to 1 is good)}$$

- False Positive Rate (Fallout):

$$\text{FPR} = \frac{FP}{N} \quad \text{(close to 0 is good)}$$



Predict $i \in \texttt{test}$ to be **positive** if $s_i \equiv f(\beta^\top x_i) > 0.5$

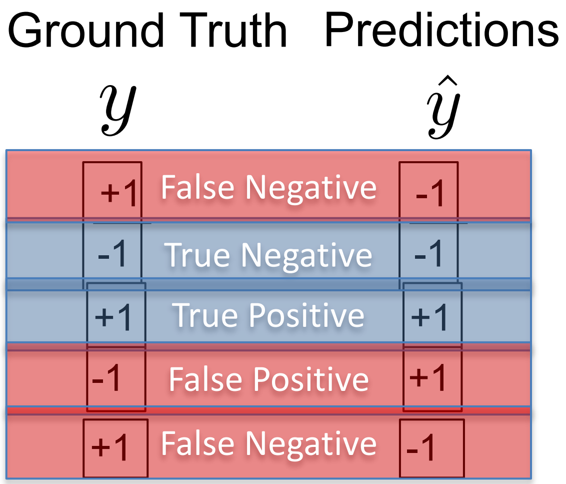
Tradeoff Between TPR and FPR

- True Positive Rate (Sensitivity):

$$\mathrm{TPR} = \frac{TP}{P}$$ (close to 1 is good)

- False Positive Rate (Fallout):

$$\mathrm{FPR} = \frac{FP}{N}$$ (close to 0 is good)



Predict $i \in \mathtt{test}$ to be **positive** if $s_i \equiv f(\beta^\top x_i) > \tau$

- $\tau = 0.5$: standard classifier
- $\tau = 0.0$: label all positive (good TPR, bad FPR)
- $\tau = 1.0$: label all negative (bad TPR, good FPR)

Tufts

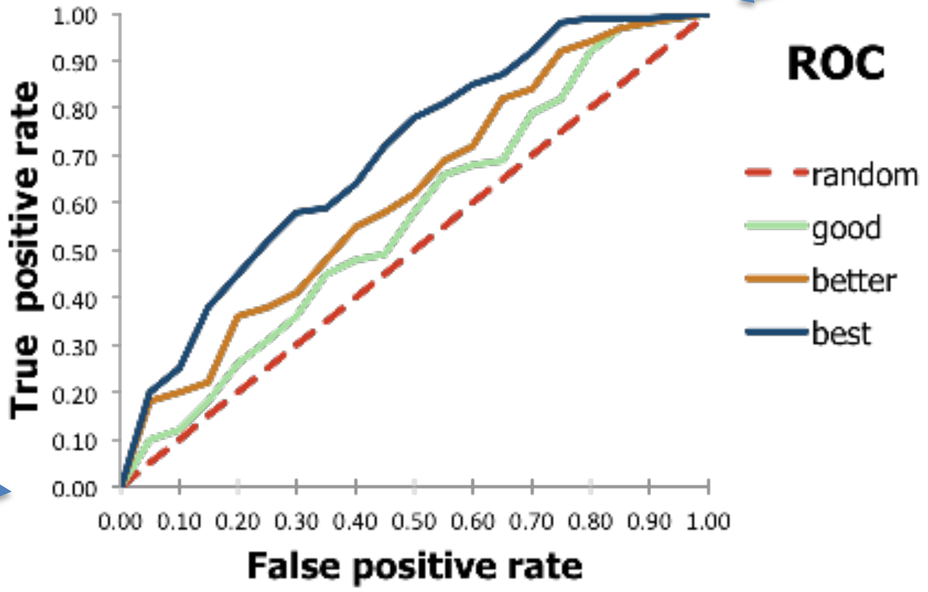
Receiver Operating Characteristic (ROC) Curve

- ❑ True Positive Rate (Sensitivity): $\mathtt{TPR} = \frac{TP}{P}$ (close to 1 is good)
- ❑ False Positive Rate (Fallout): $\mathtt{FPR} = \frac{FP}{N}$ (close to 0 is good)

Predict $i \in \mathtt{test}$ to be **positive** if $s_i \equiv f(\beta^\top x_i) > \tau$

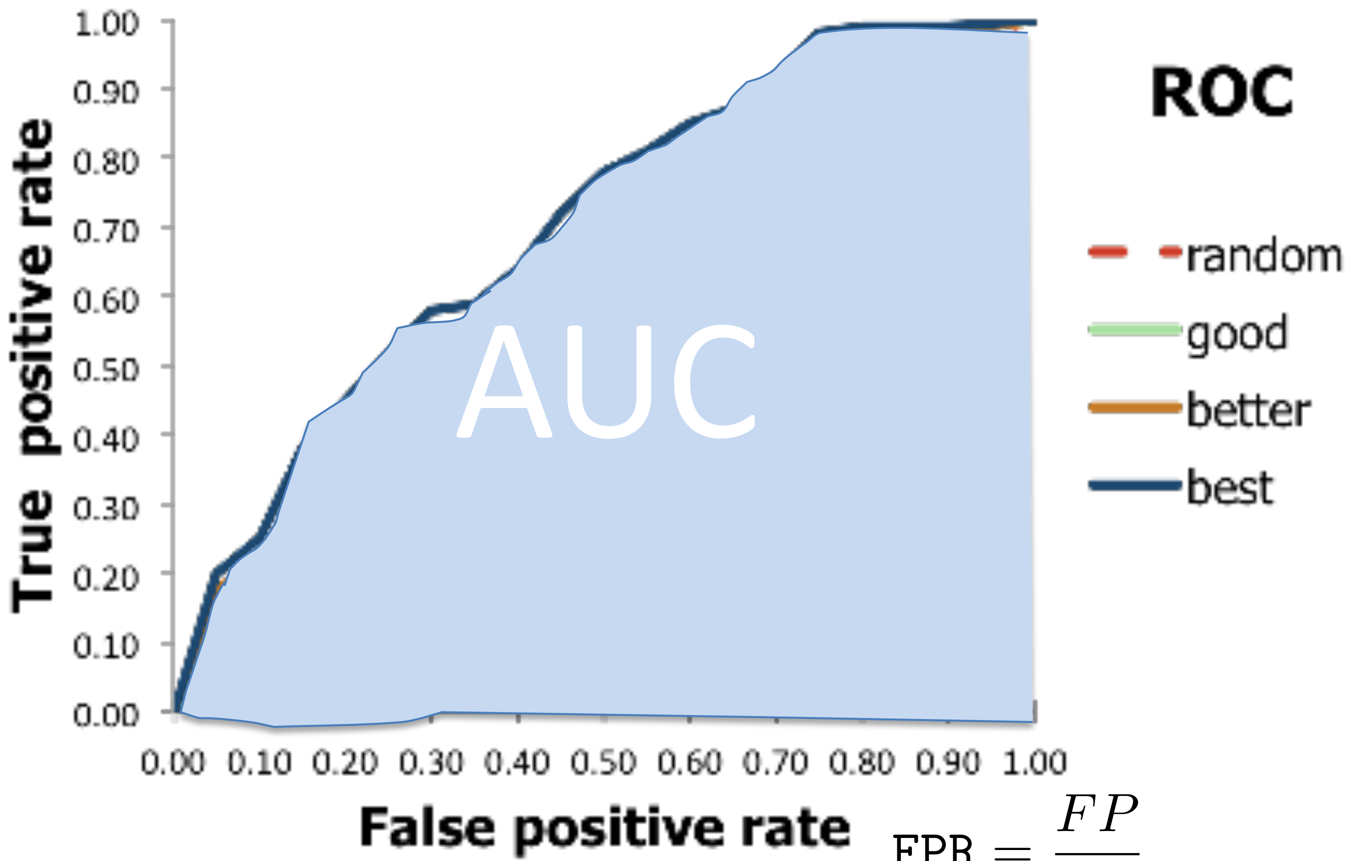
$\tau = 0.0$

$\tau = 1.0$



Area Under the Curve (AUC)

$$\text{TPR} = \frac{TP}{P}$$



$$\text{FPR} = \frac{FP}{N}$$

Tufts

Area Under the Curve (AUC): Random Prediction

$$\text{TPR} = \frac{TP}{P} \qquad \text{FPR} = \frac{FP}{N}$$

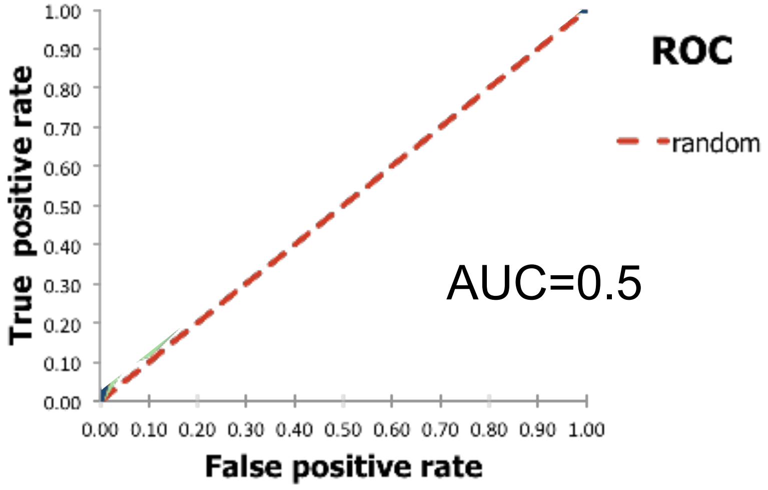
Suppose **a fraction x** of test set is **positive** and (1-x) is negative.

Consider **random** predictor: with **probability p predicts positive**, with (1-p) predicts negative

Then:

$$\text{TPR} = \frac{TP}{P} \approx \frac{x \cdot p}{x} = p$$

$$\text{FPR} = \frac{FP}{N} \approx \frac{(1-x) \cdot p}{1-x} = p$$



AUC=0.5

Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

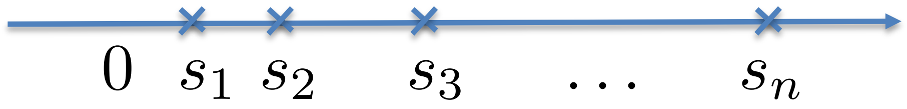
Suppose we sort test set so that

$$0 \leq s_1 \leq s_2 \leq s_3 \leq \ldots \leq s_n$$

Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

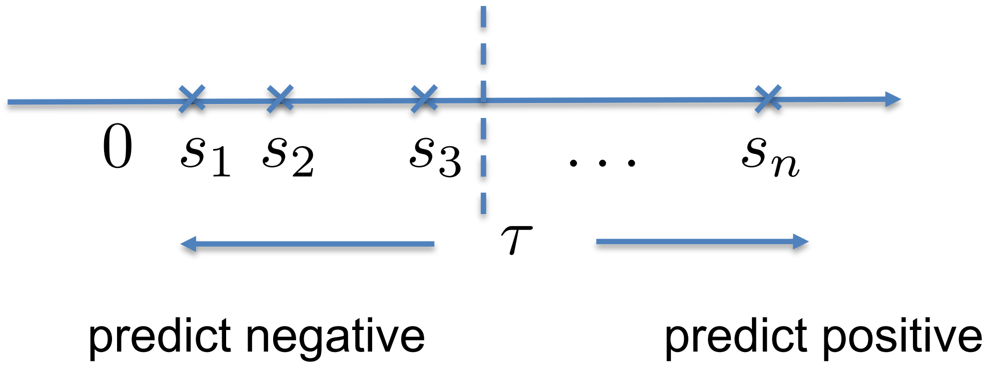
Suppose we sort test set so that



Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

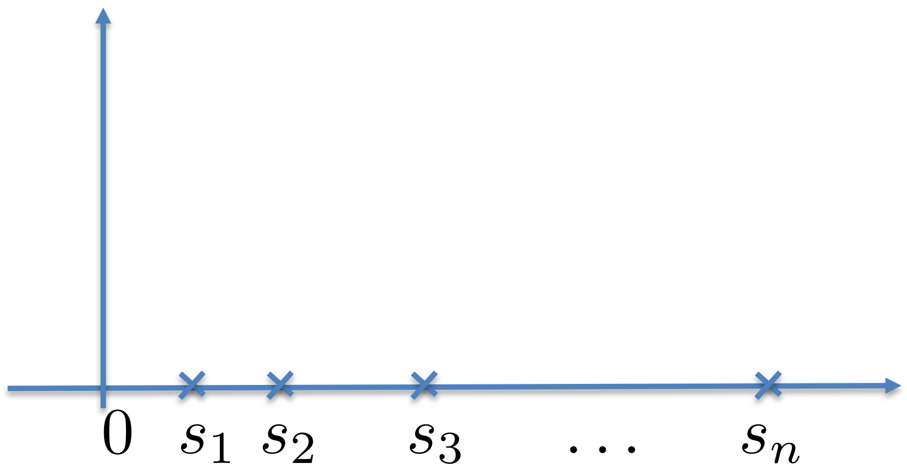


Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

$$\text{TPR} = \frac{TP}{P}$$



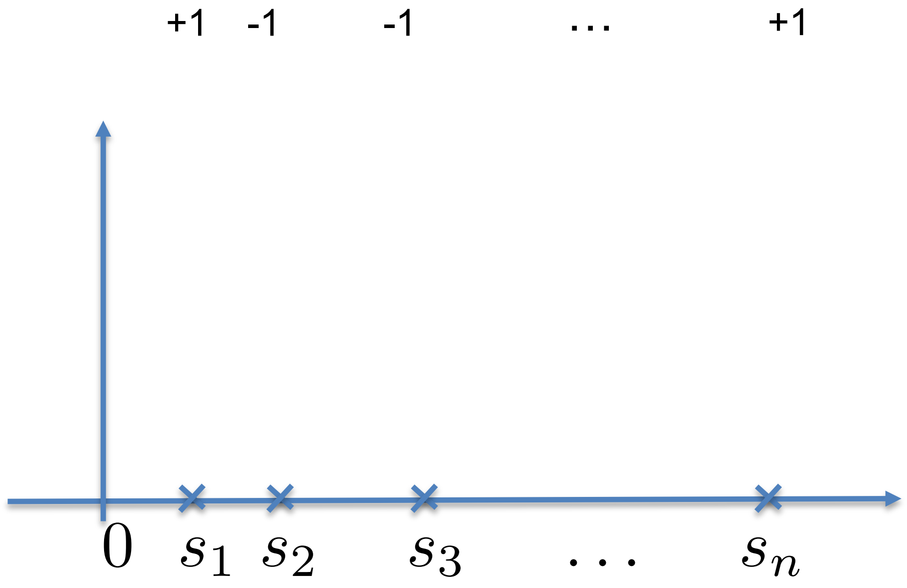
Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

Ground Truth:

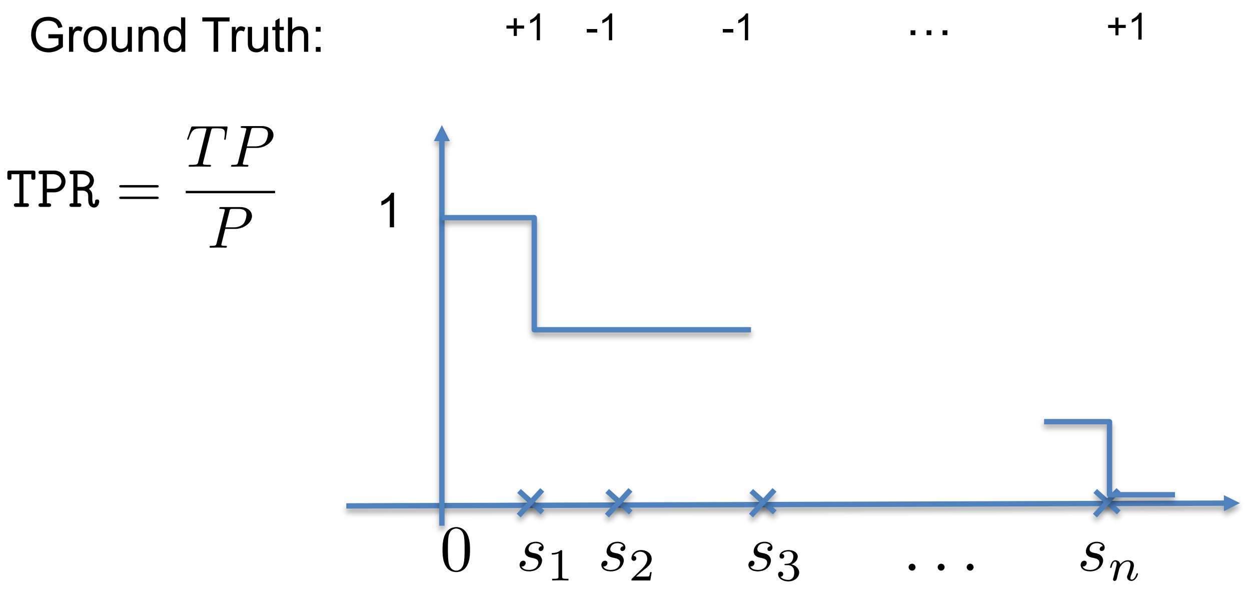
$$\text{TPR} = \frac{TP}{P}$$



Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that



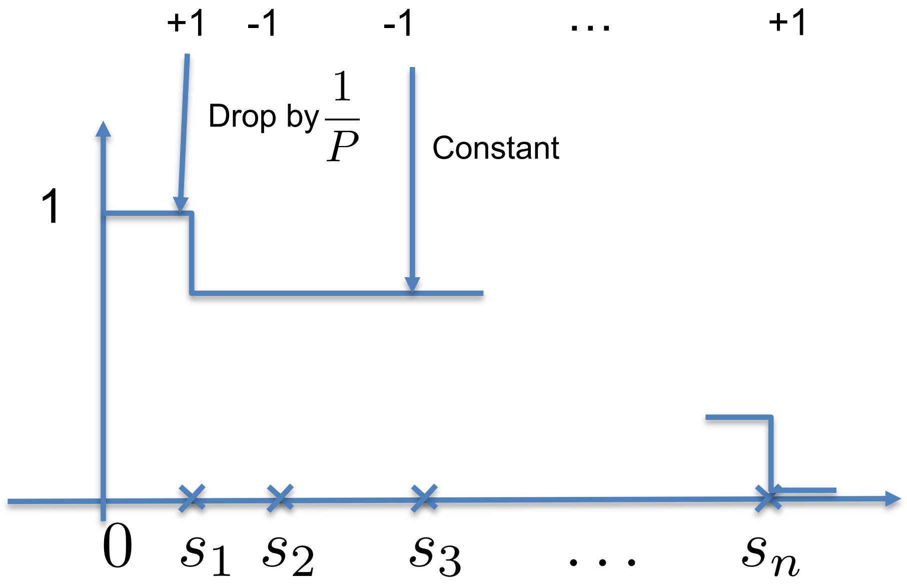
Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

Ground Truth:

$$\text{TPR} = \frac{TP}{P}$$



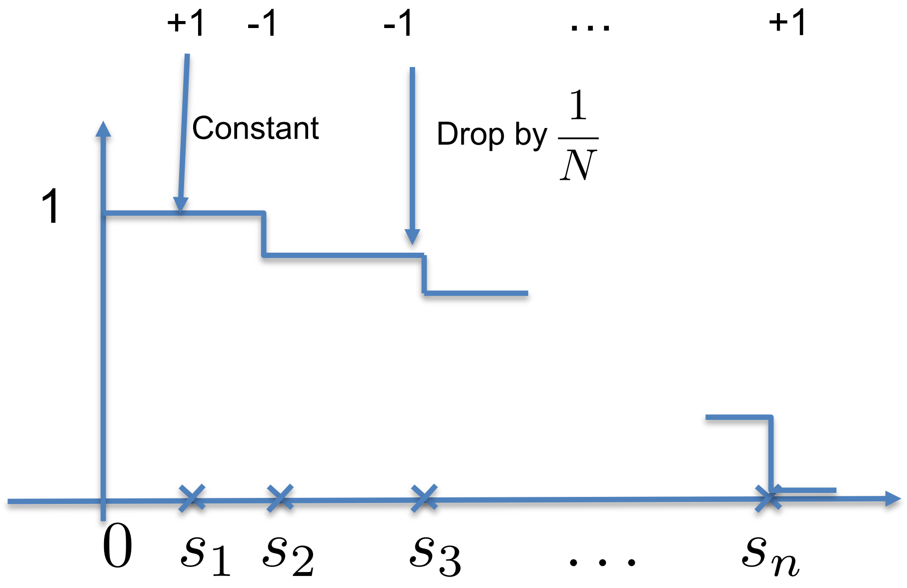
Computing the AUC

Let $s_i = f(\beta^\top x_i)$ be the score of an $i \in \texttt{test}$

Suppose we sort test set so that

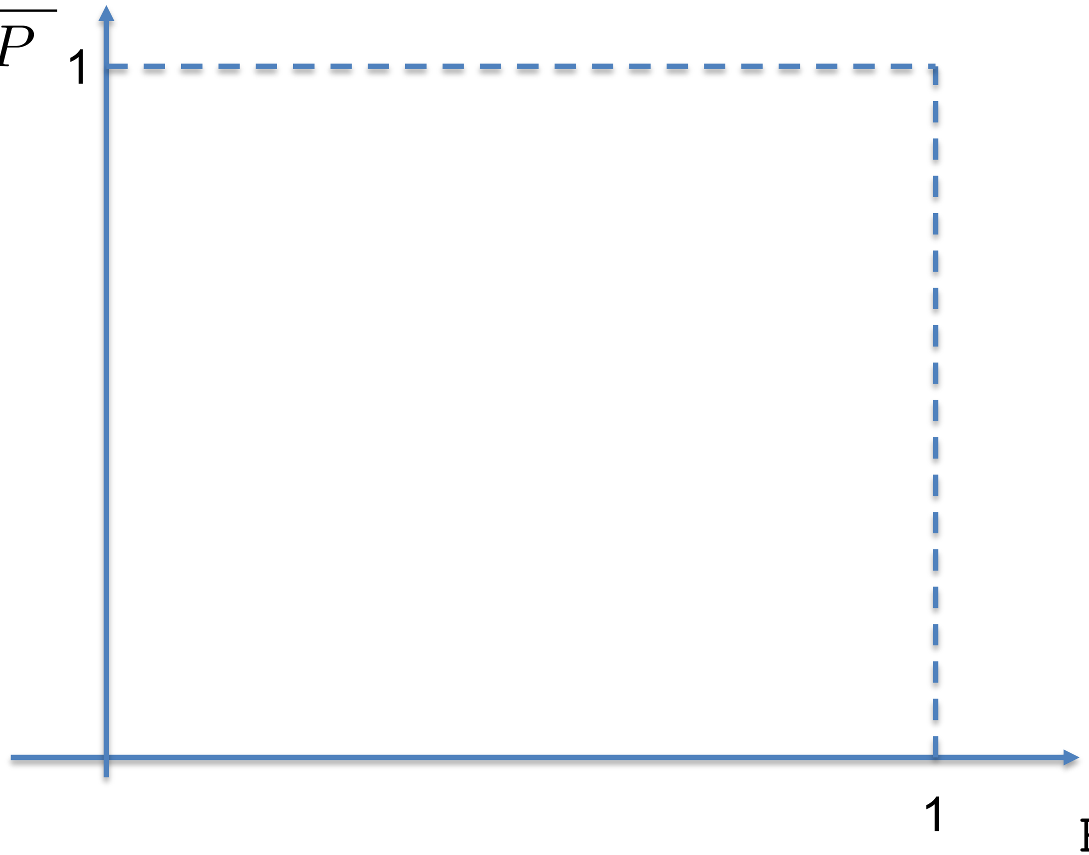
Ground Truth:

$$\text{FPR} = \frac{FP}{N}$$



Computing the AUC

$$\text{TPR} = \frac{TP}{P}$$



$$\text{FPR} = \frac{FP}{N}$$

Sorted Ground Truth

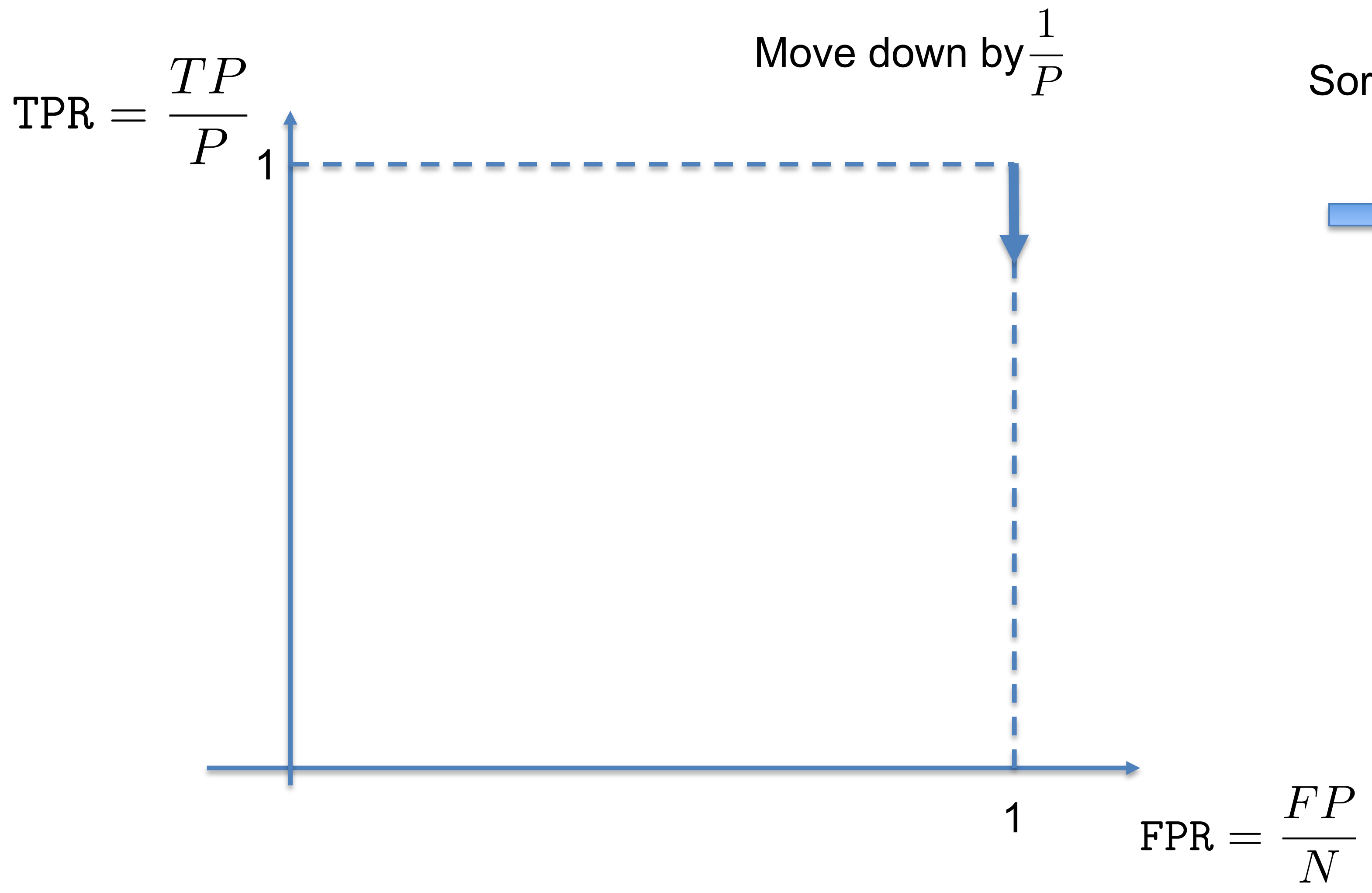
+1

-1

-1

+1

Computing the AUC


$\text{TPR} = \frac{TP}{P}$
1
Move down by $\frac{1}{P}$
1
$\text{FPR} = \frac{FP}{N}$

Sorted Ground Truth

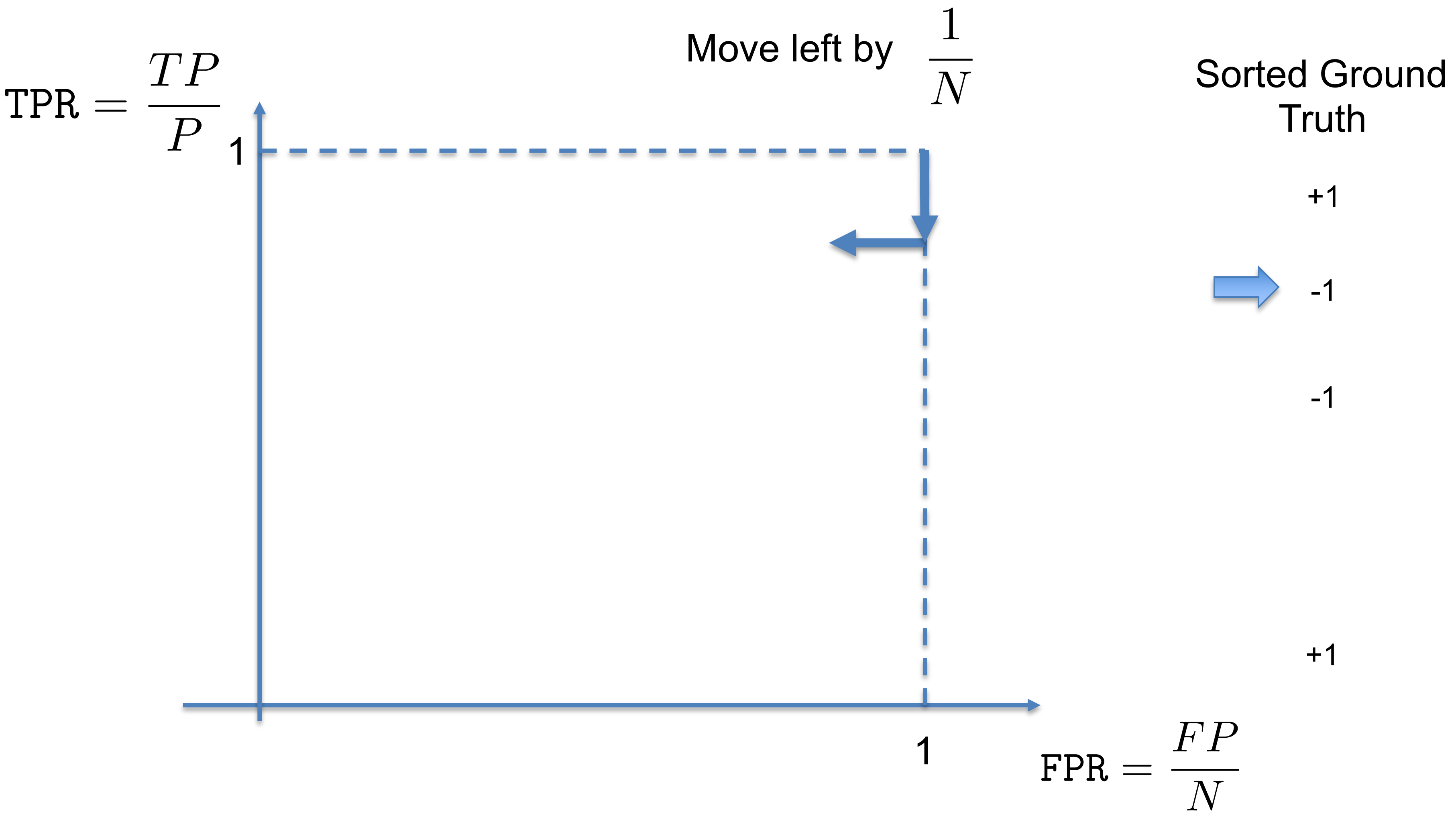
+1

-1

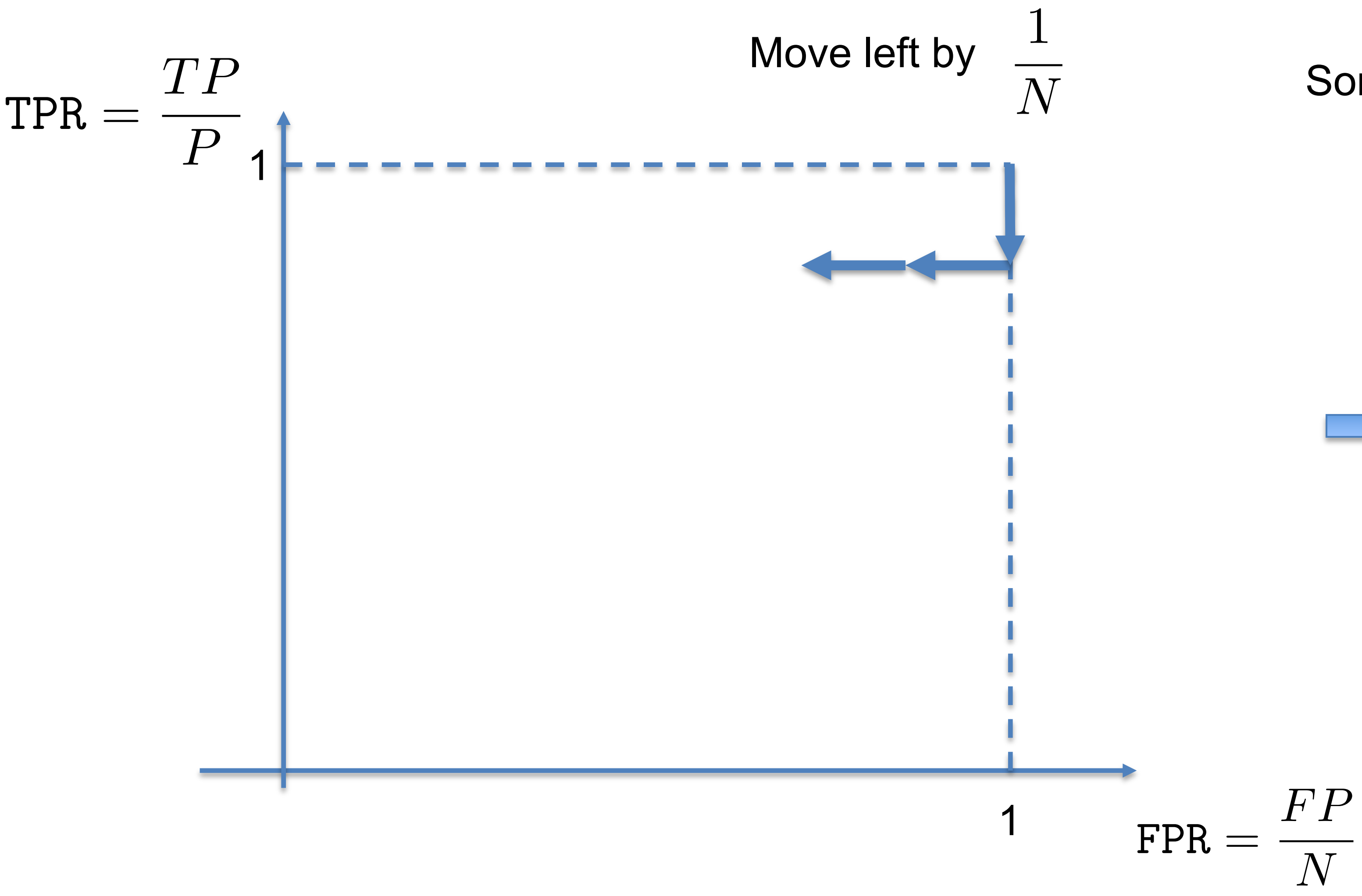
-1

+1

Computing the AUC


Move left by $\frac{1}{N}$
Sorted Ground Truth
$\mathrm{TPR} = \frac{TP}{P}$
1
1
$\mathrm{FPR} = \frac{FP}{N}$
+1
-1
-1
+1

Computing the AUC



Move left by $\frac{1}{N}$
$\mathrm{TPR} = \frac{TP}{P}$
1
1
$\mathrm{FPR} = \frac{FP}{N}$

Sorted Ground Truth

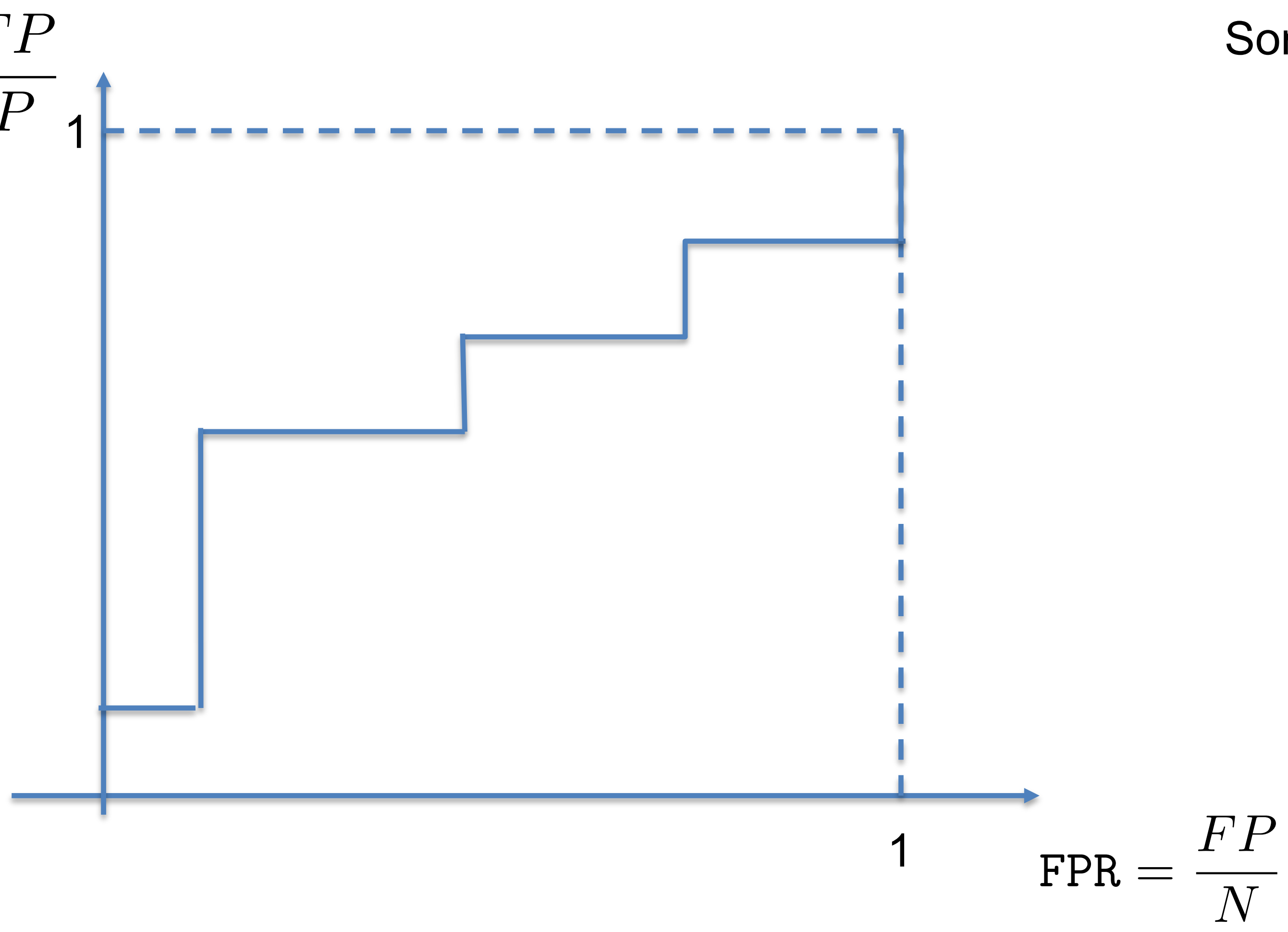
+1

-1

-1

+1

Computing the AUC



Sorted Ground Truth

+1

-1

-1

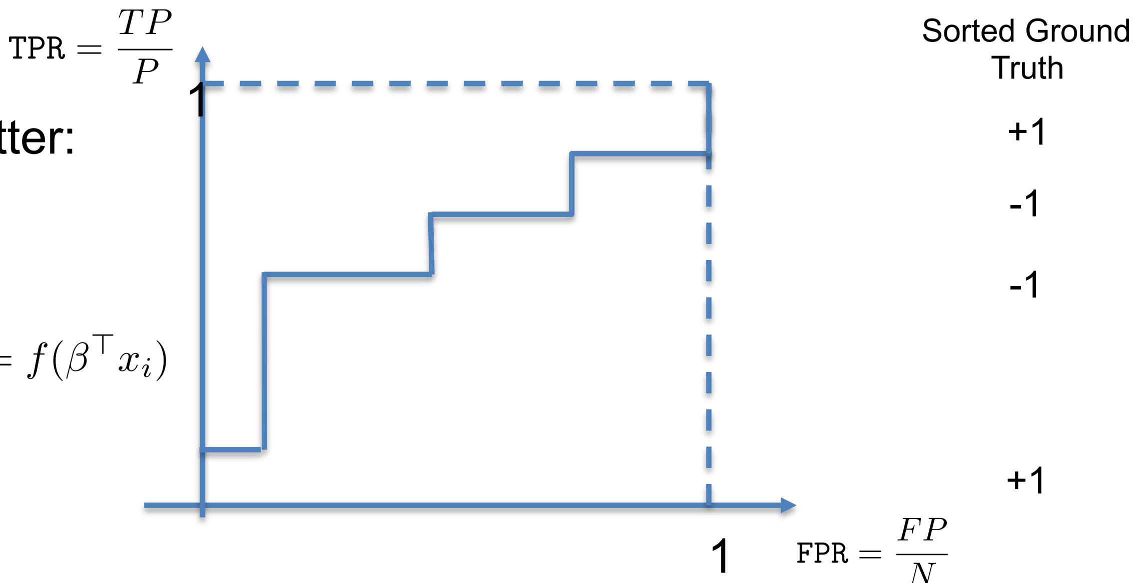
+1

Tufts

Computing the AUC

- ❑ Values of scores do not matter: only ordering does.

e.g. $s_i = \beta^\top x_i$ instead of $s_i = f(\beta^\top x_i)$



- ❑ Imbalance of P vs N only effects granularity

- ❑ You do not need to "integrate" over τ