Lecture 20
Model Selection and Validation

DTS201TC Pattern Recognition

Outline

- Metrics for Evaluating Classifier Performance
- Bias-Variance Tradeoff
- Model selection and validation

Outline

- Metrics for Evaluating Classifier Performance
- Bias-Variance Tradeoff
- Model selection and validation

Metrics for Evaluating Classifier Performance (Review)

- **True positives** (*TP*): These refer to the positive tuples that were correctly labeled by the classifier. Let *TP* be the number of true positives.
- **True negatives** (*TN*): These are the negative tuples that were correctly labeled by the classifier. Let *TN* be the number of true negatives.
- **False positives** (*FP*): These are the negative tuples that were incorrectly labeled as positive (e.g., tuples of class *buys computer* = *no* for which the classifier predicted *buys computer* = *yes*). Let *FP* be the number of false positives.
- **False negatives** (*FN*): These are the positive tuples that were mislabeled as negative (e.g., tuples of class *buys computer* = *yes* for which the classifier predicted *buys computer* = *no*). Let *FN* be the number of false negatives.

Metrics for Evaluating Classifier Performance (Review)

- **Confusion Matrix**

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Metrics for Evaluating Classifier Performance

Measure	*Formula*
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
F, F_1, F-score, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β, where β is a non-negative real number	$\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

Evaluation Measures

- The **accuracy** of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. also referred to as the **recognition rate**
- Class Imbalance Problem**:** the data set distribution reflects a significant majority of the negative class and a minority positive class.
  - The **sensitivity** and **specificity** measures can be used

$$sensitivity = \frac{TP}{P}$$

$$specificity = \frac{TN}{N}.$$

Evaluation Measures

- 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

Sensitivity: 90 / 300 = 30.00%

Specification: 9560 / 9700 = 98.56%

Precision: 90/230 = 39.13%

ROC Curve

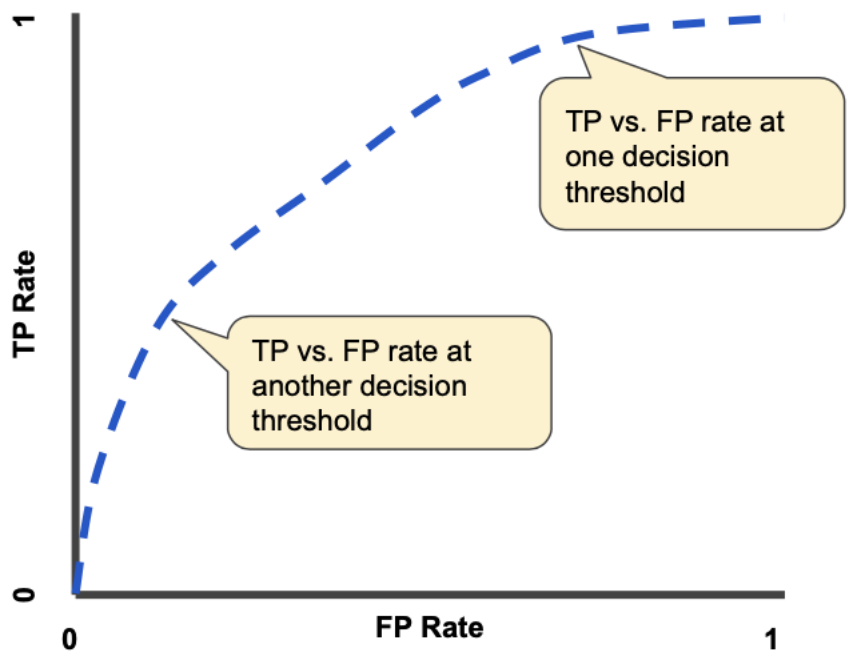
- An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds.

- This curve plots two parameters
  - True Positive Rate (TPR), where TP+FN = P

$$TPR = \frac{TP}{TP + FN}$$

  - False Positive Rate (FPR), where FP+TN = N

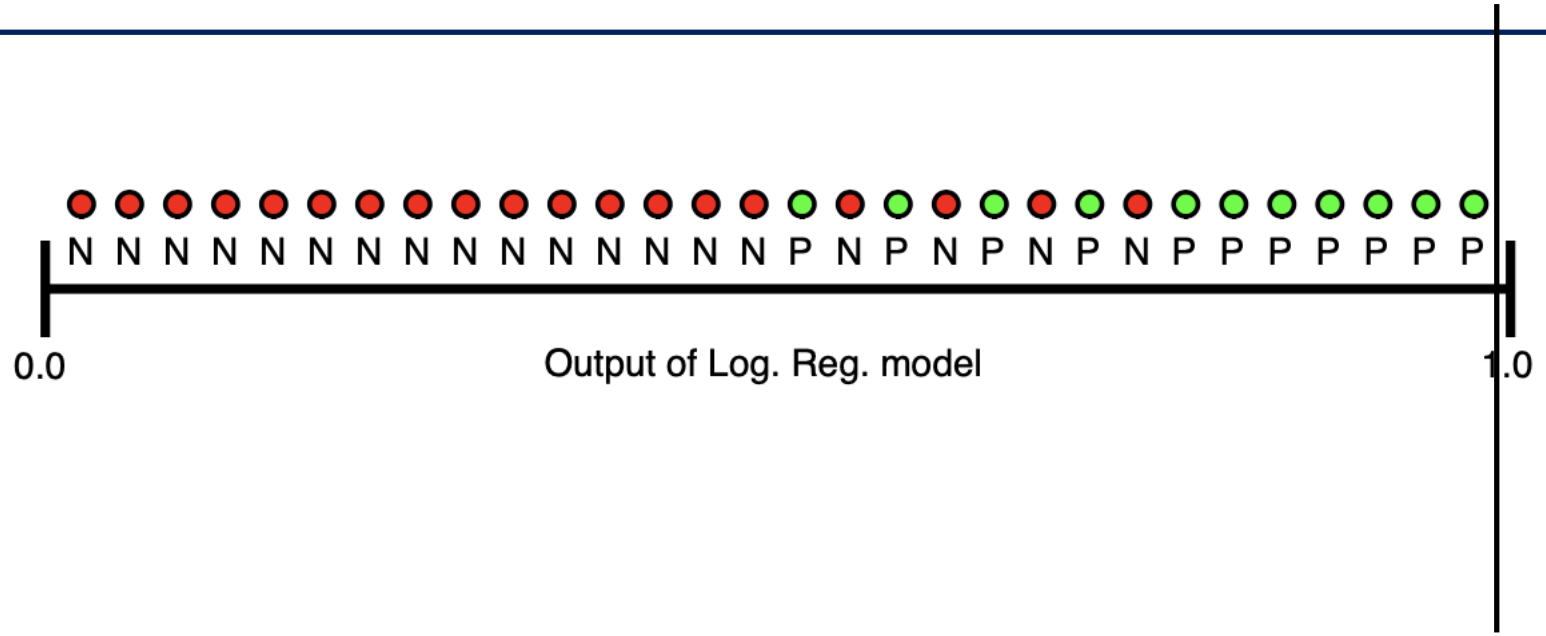
$$FPR = \frac{FP}{FP + TN}$$



ROC: Example

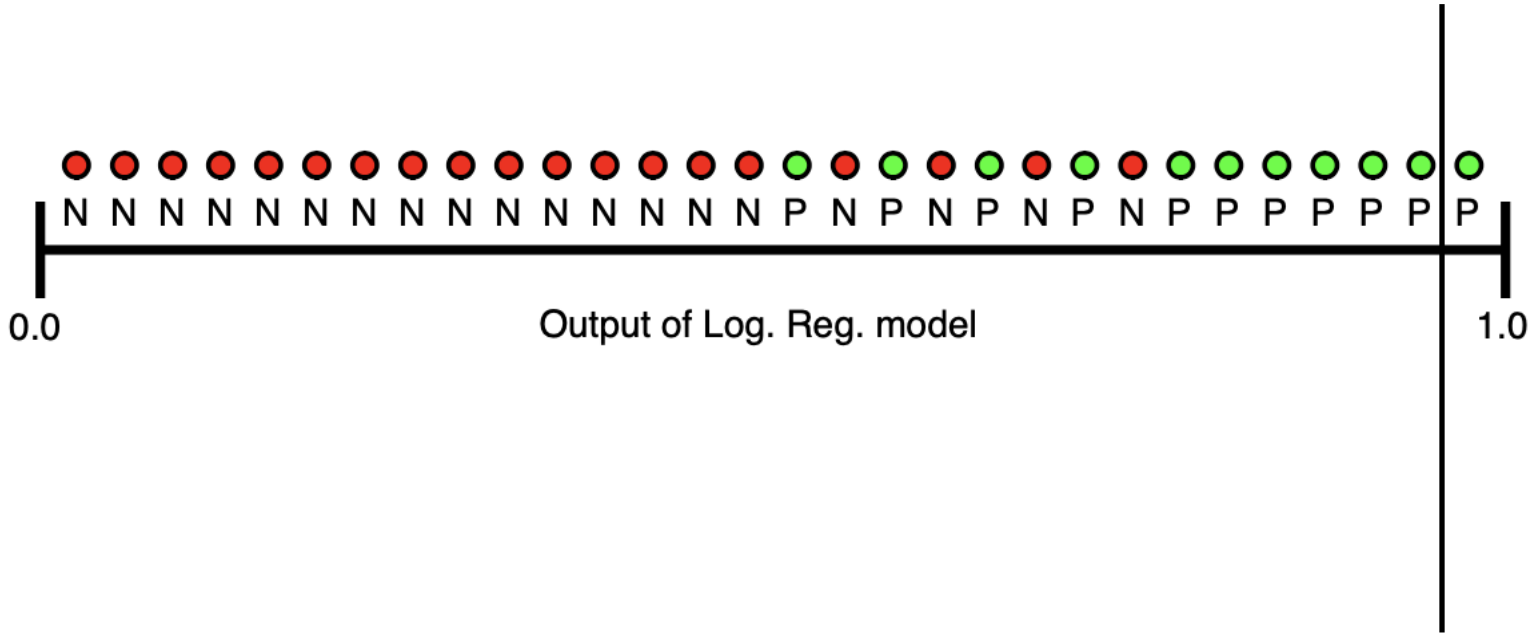
- Given the following examples, which are arranged from left to right in ascending order of logistic regression predictions: (30 examples)

ROC: Example

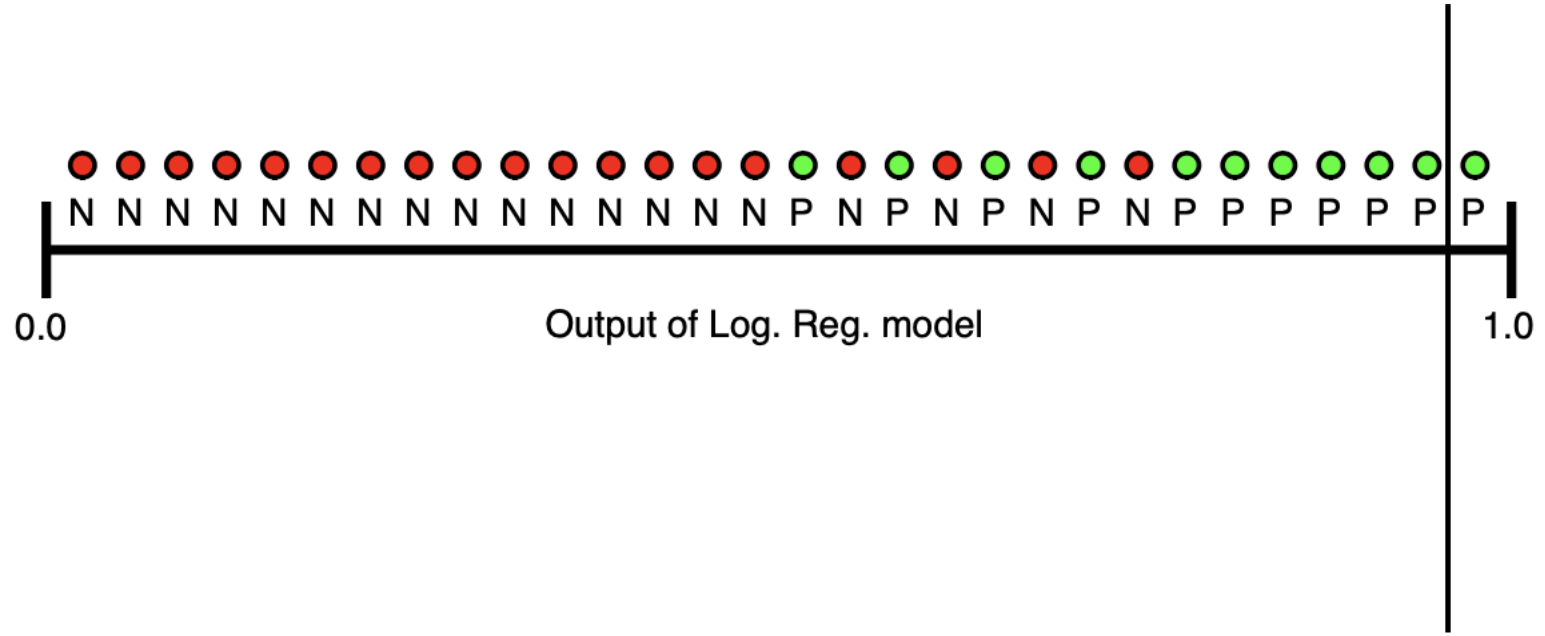


- TPR = 0
- FPR = 0
- (0,0)

ROC: Example

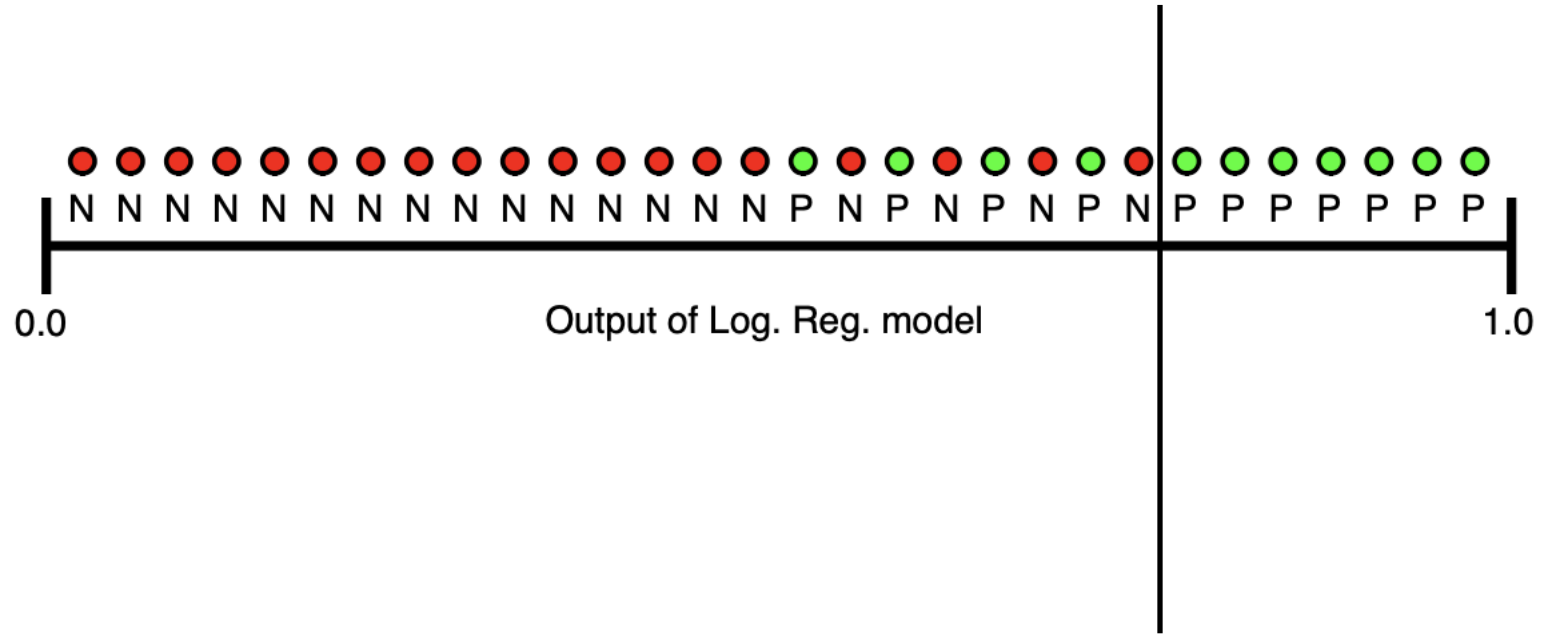

N N N N N N N N N N N N N N P N P N P N P N P P P P P P P
0.0
Output of Log. Reg. model
1.0

ROC: Example



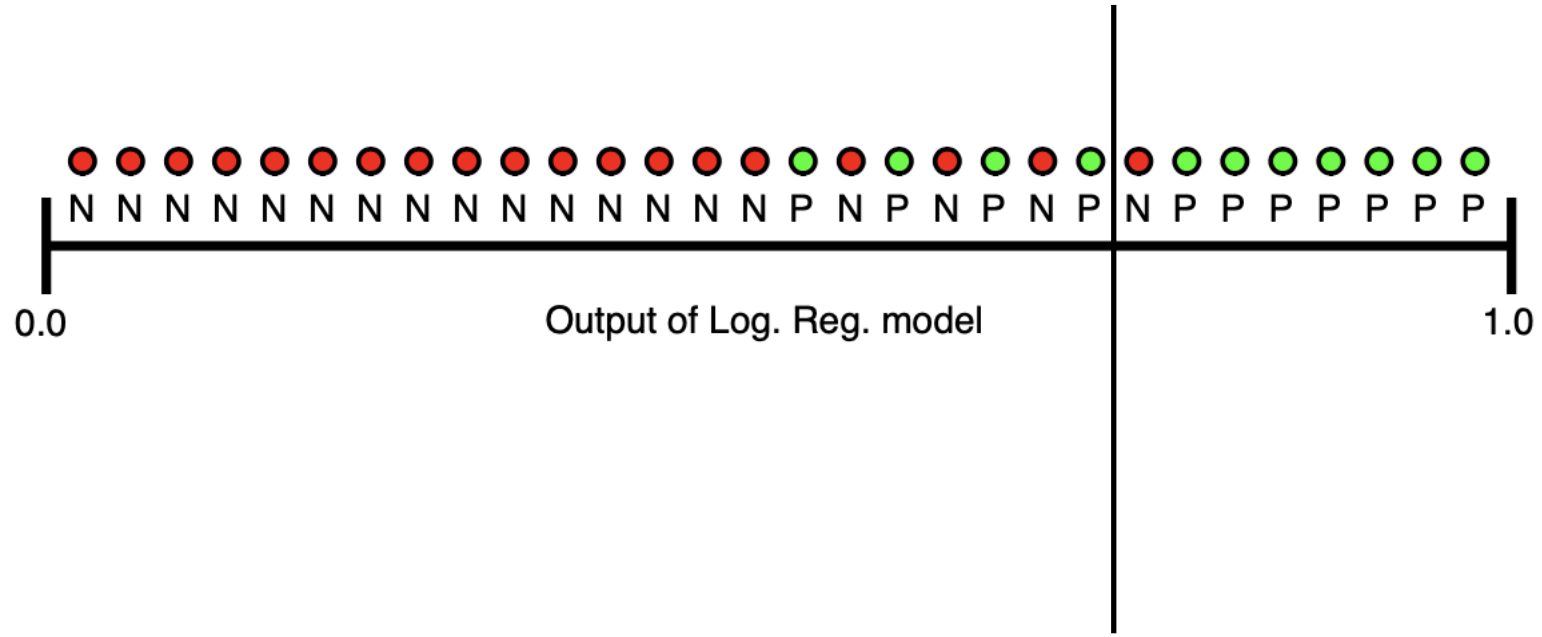
- TPR = 1/11 ≈ 0.09
- FPR = 0
- (0, 0.09)

ROC: Example



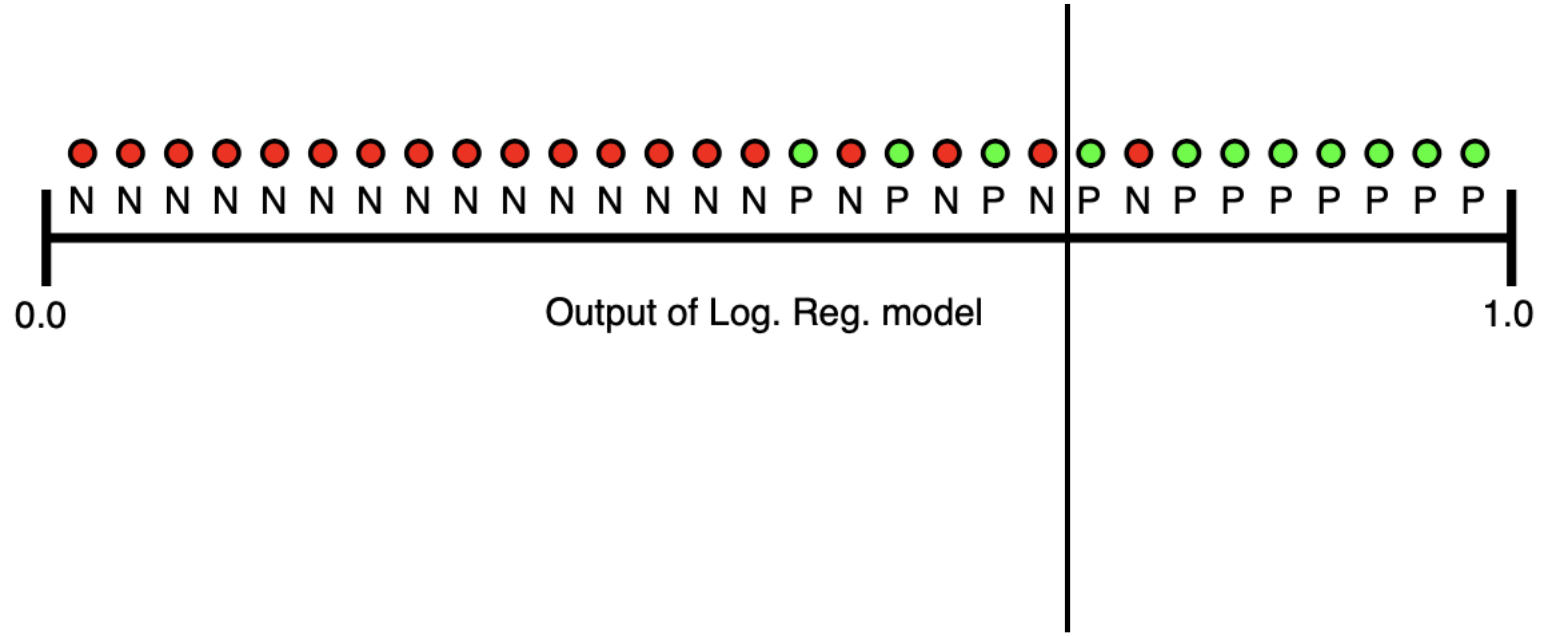
- TPR = 7/11 ≈ 0.64
- FPR = 0
- (0, 0.64)

ROC: Example



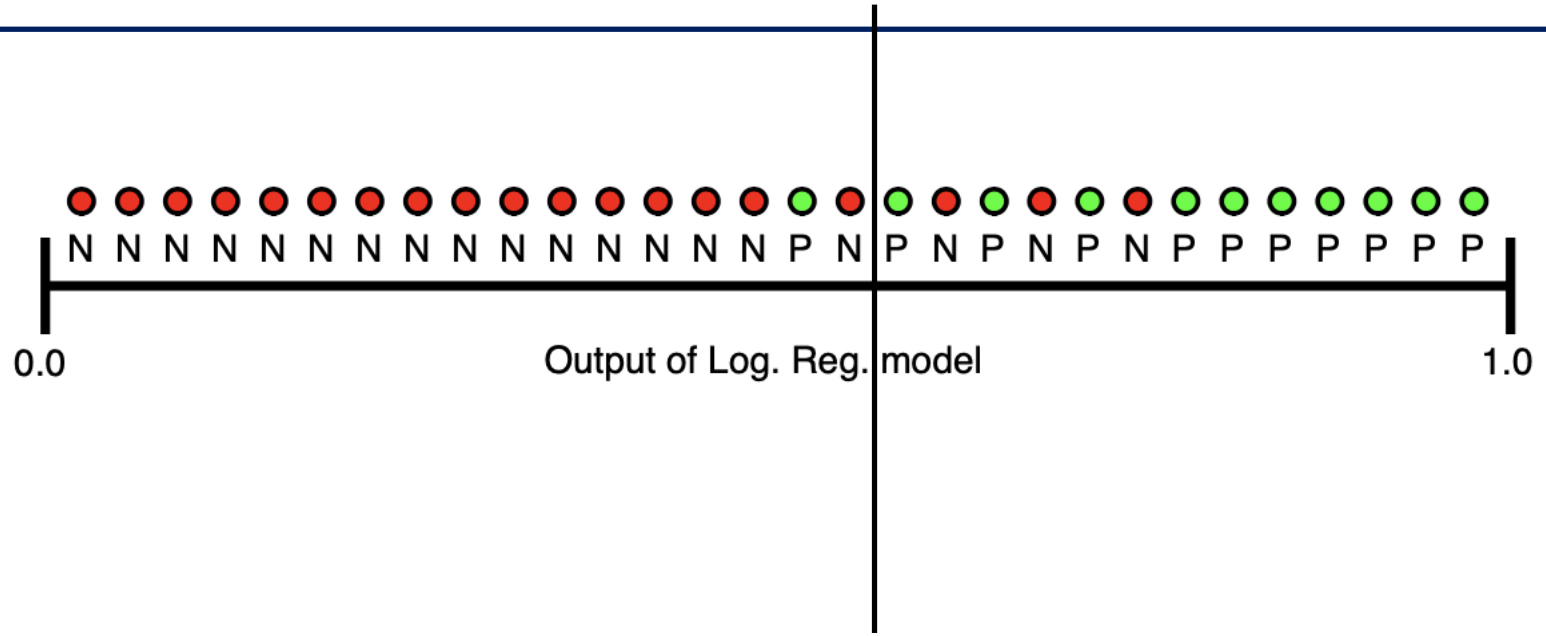
- TPR ≈ 0.64
- FPR = 1/19 ≈ 0.05
- (0.05, 0.64)

ROC: Example



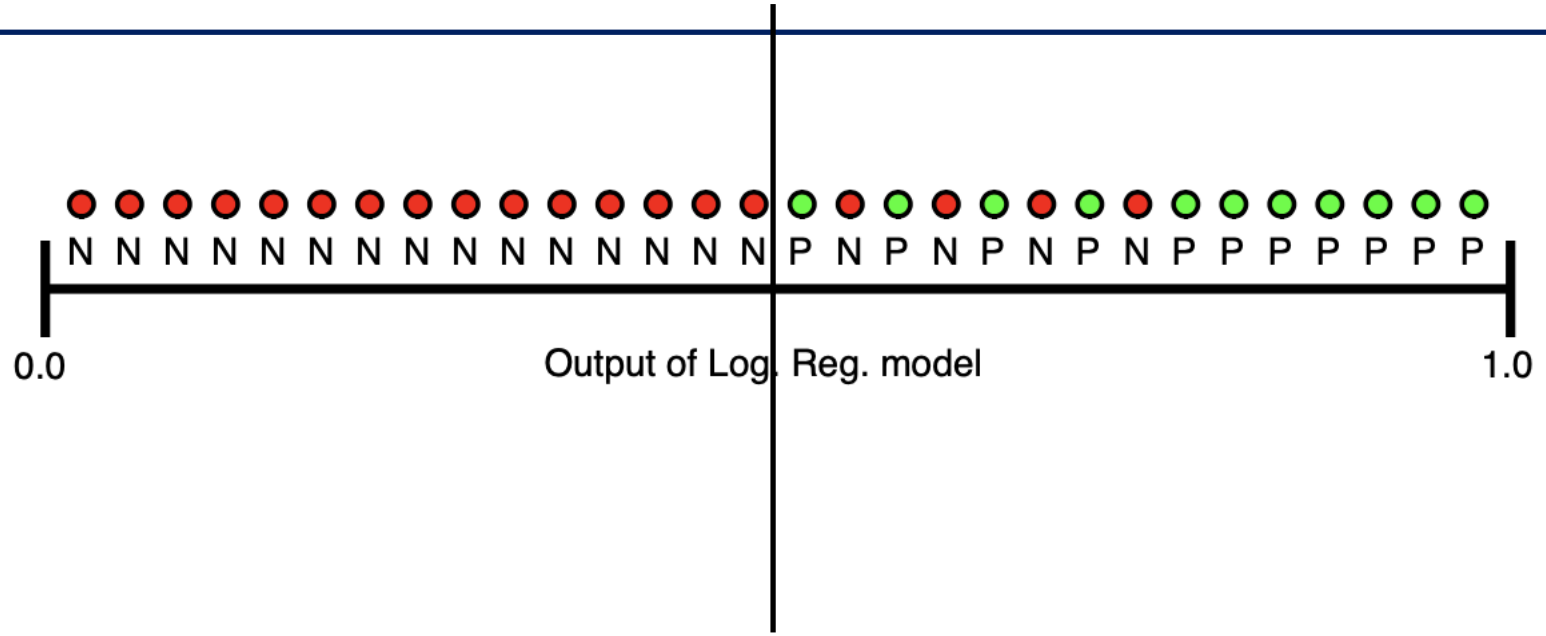
- TPR = 8/11 ≈ 0.73
- FPR = 1/19 ≈ 0.05
- (0.05, 0.73)

ROC: Example



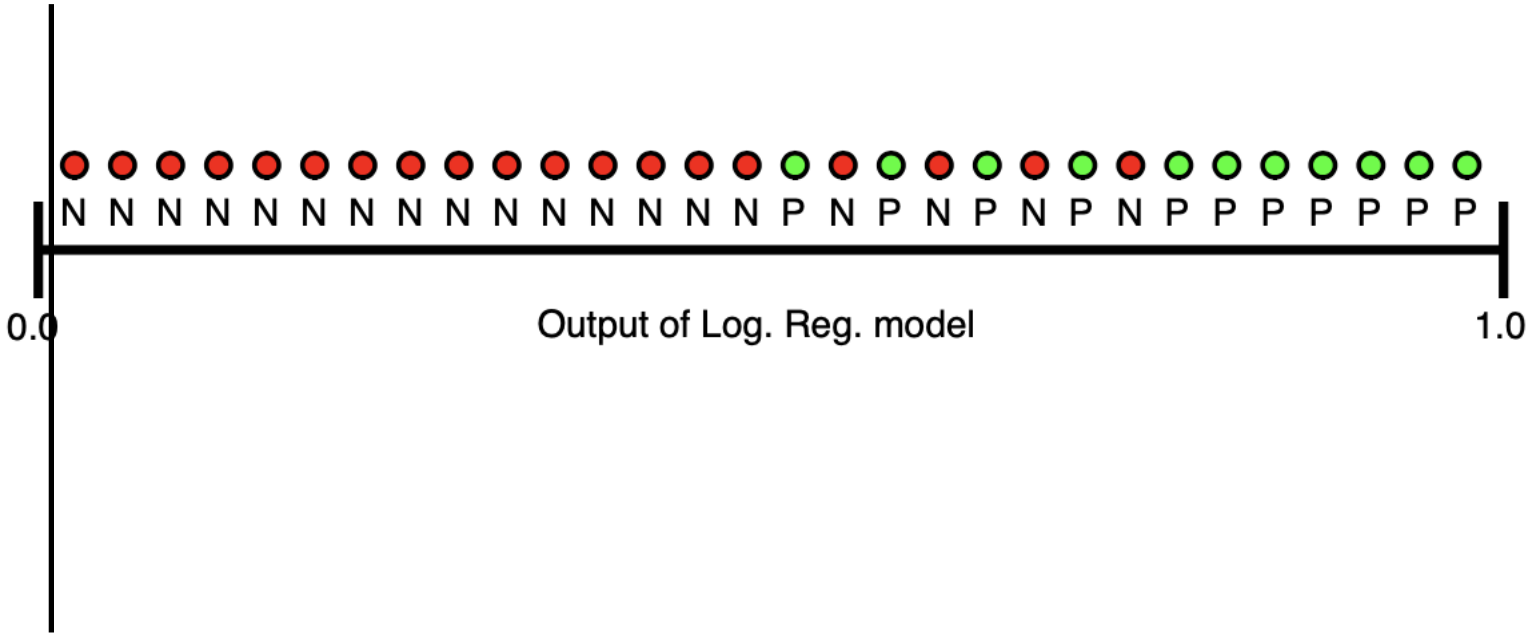
- TPR = 10/11 ≈ 0.9
- FPR = 3/19 ≈ 0.16
- (0.16, 0.9)

ROC: Example



- TPR = 1
- FPR = 4/19 ≈ 0.21
- (0.21, 1)

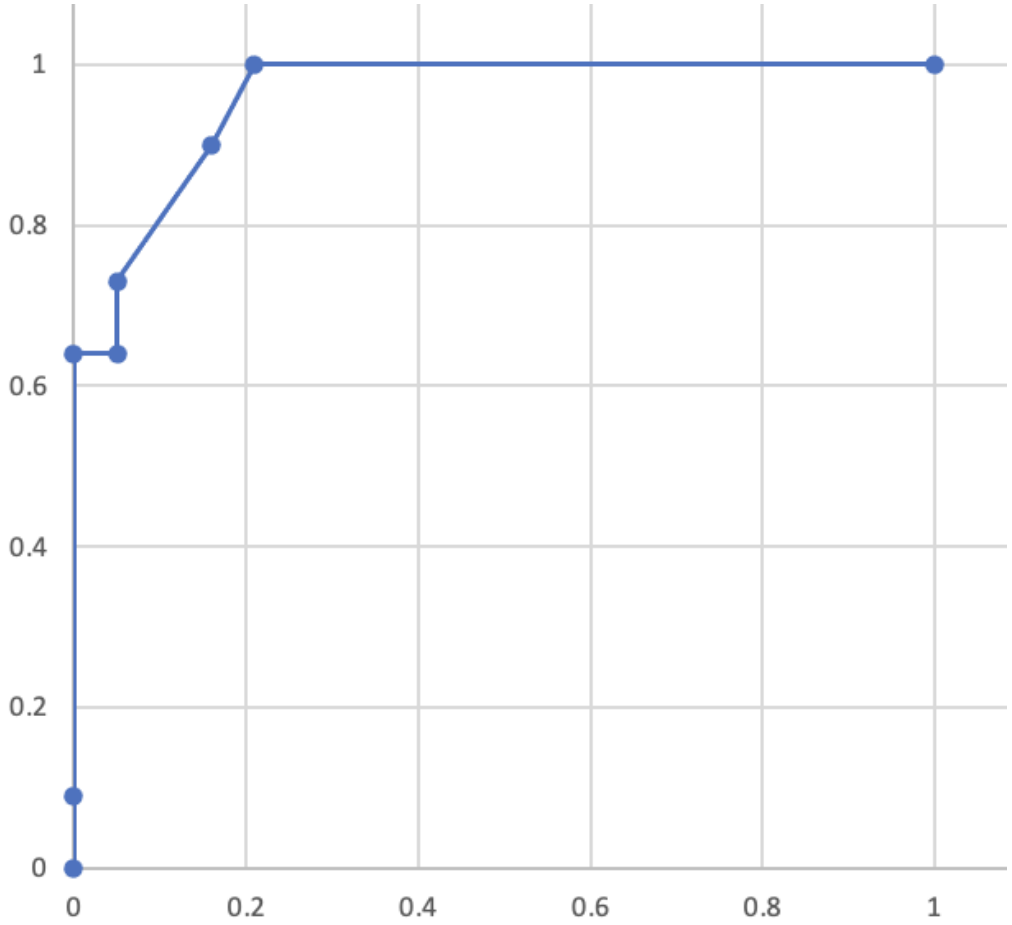
ROC: Example



- TPR = 1
- FPR = 1
- (1, 1)

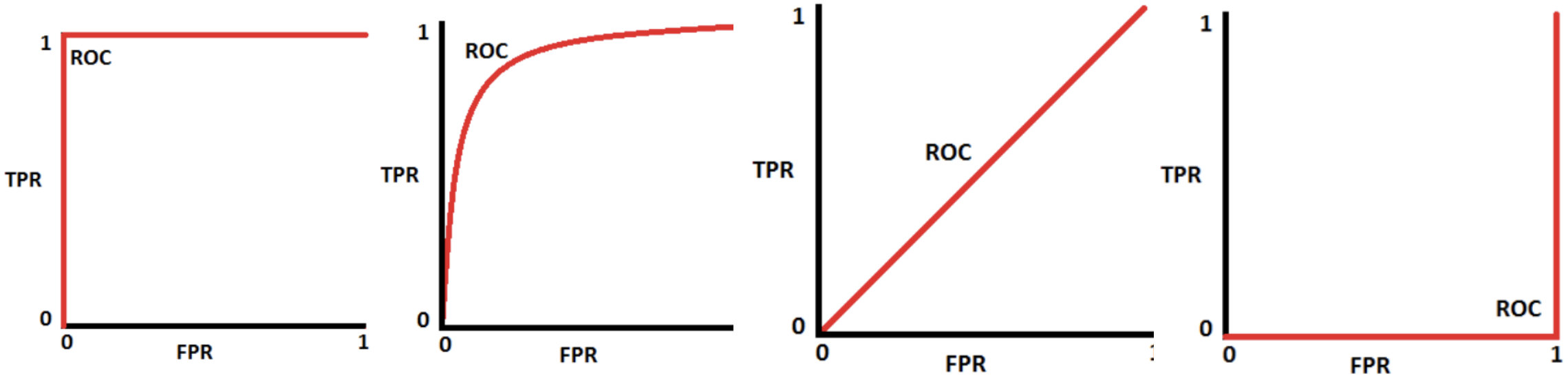
ROC Curve

- (0,0)
- (0, 0.09)
- (0, 0.64)
- (0.05, 0.64)
- (0.05, 0.73)
- (0.16, 0.9)
- (0.21, 1)
- (1,1)



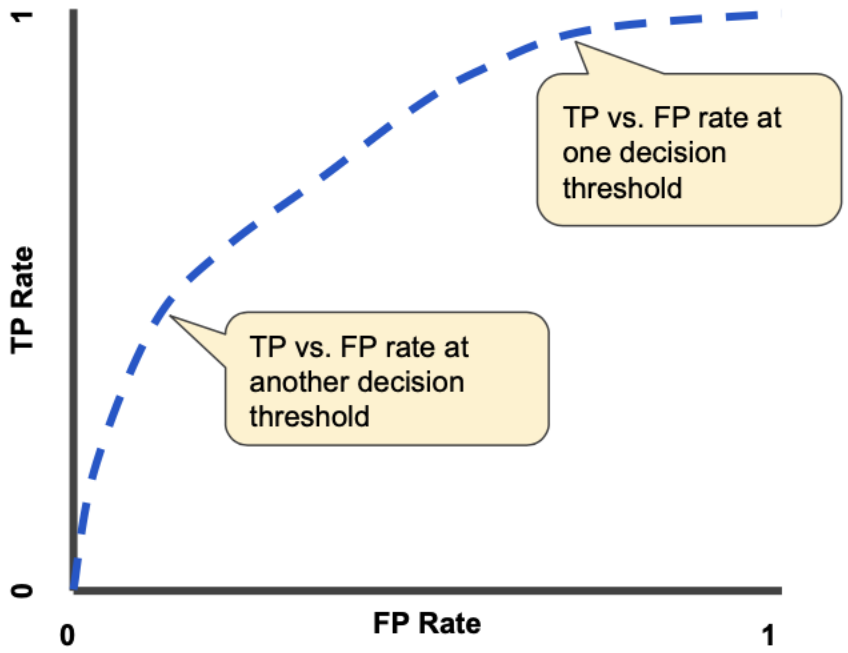
ROC Curve

- How can we get the performance by ROC curve



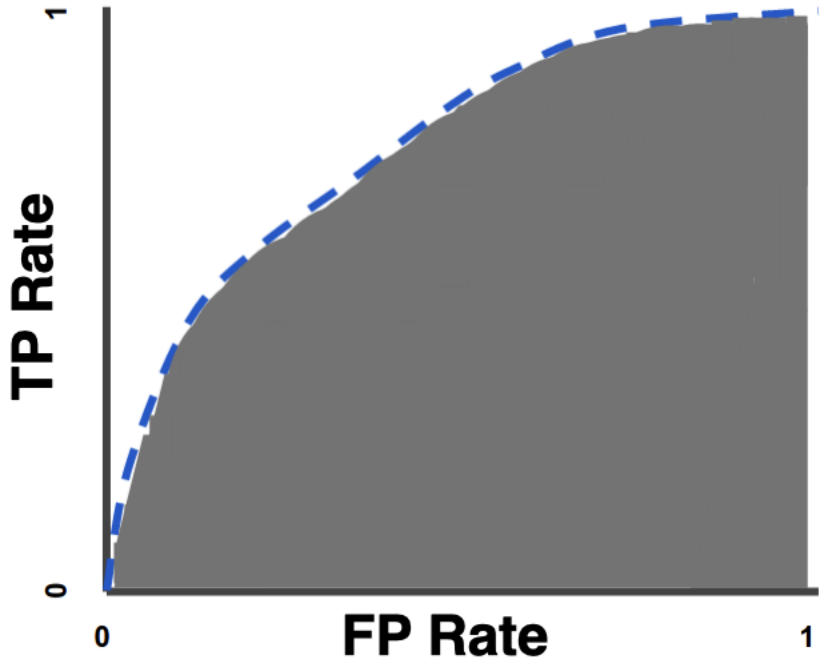
ROC Curve

- How can we get the performance by ROC curve: Area Under the Curve (AUC)



AUC

- **AUC** stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1)

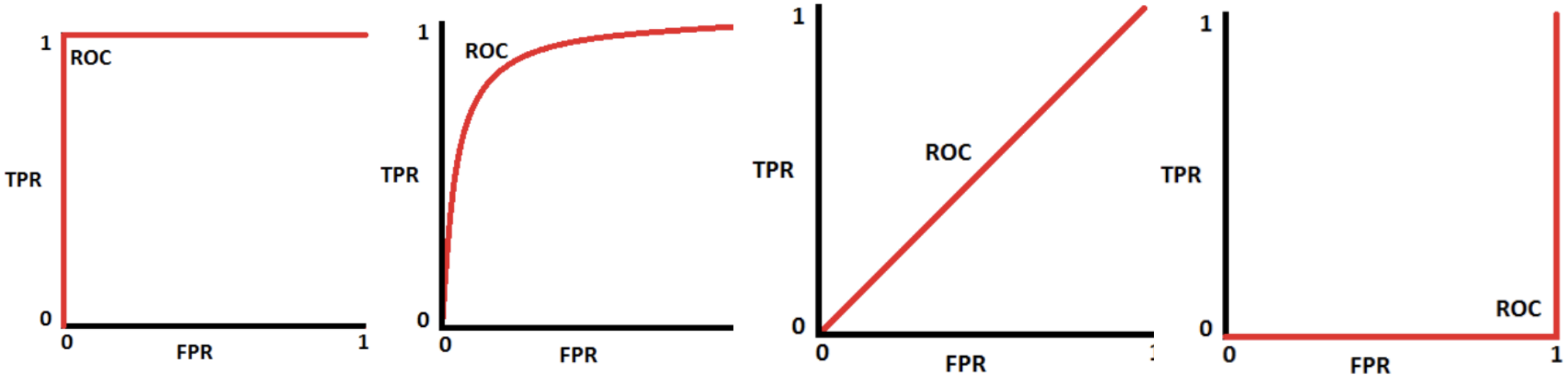


AUC

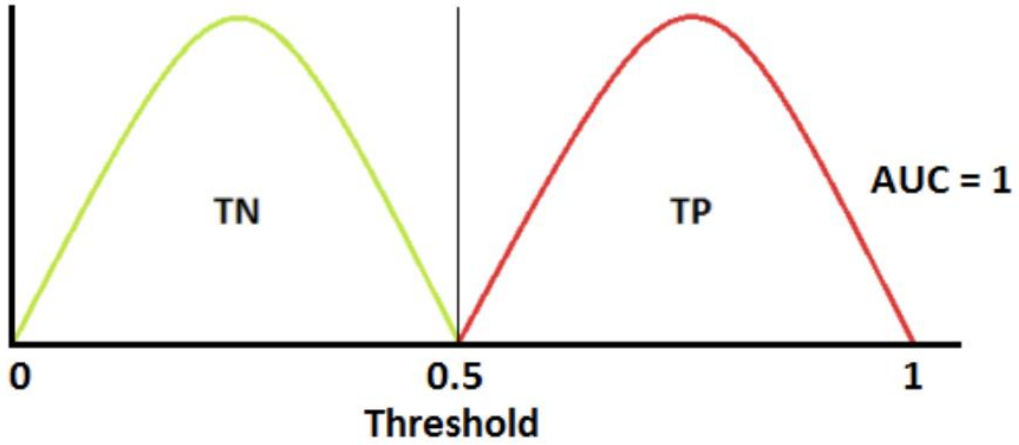
- AUC provides an aggregate measure of performance across all possible classification thresholds.
- One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example.
- AUC represents the probability that a random positive (green) example is positioned to the right of a random negative (red) example.
- AUC ranges in value from 0 to 1.
  - A model whose predictions are 100% wrong has an AUC of 0.0;
  - one whose predictions are 100% correct has an AUC of 1.0. (excellent model)

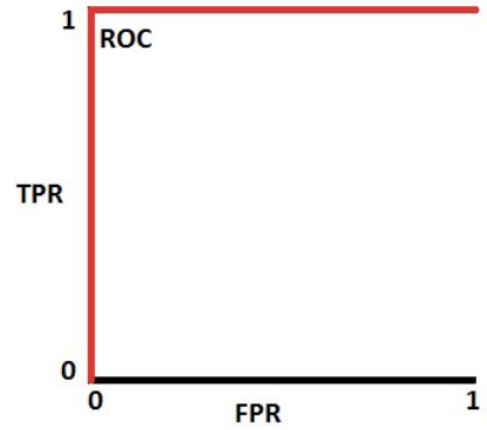
AUC

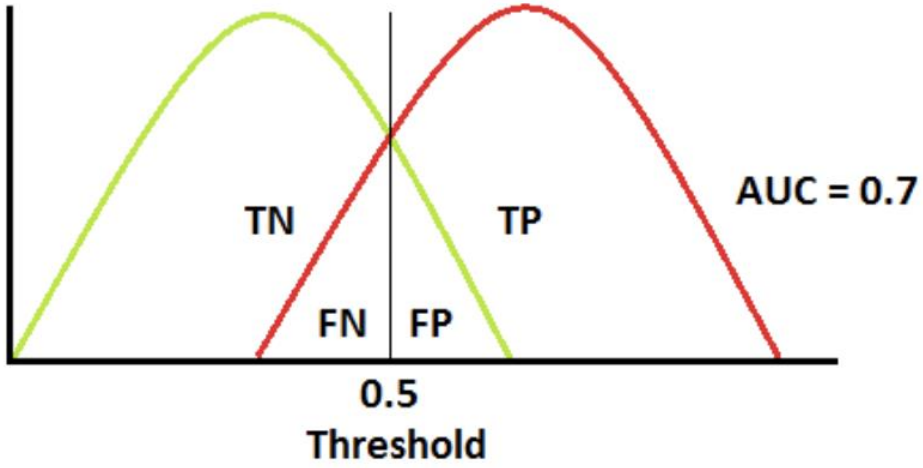
- Which one do you want ?

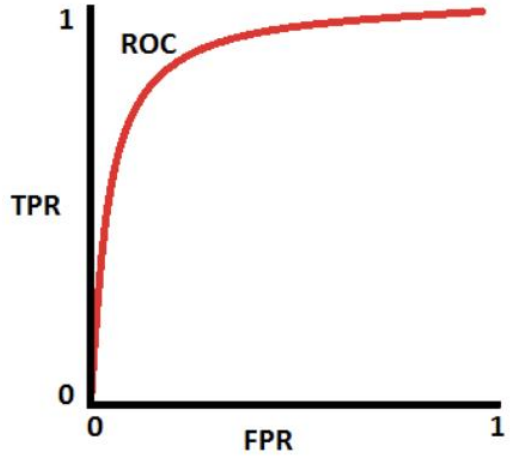


AUC

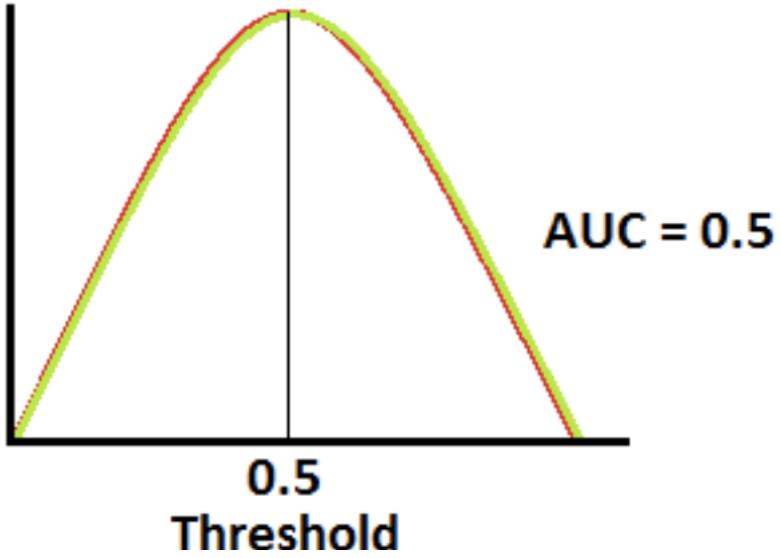

TN
TP
AUC = 1
0
0.5
1
Threshold

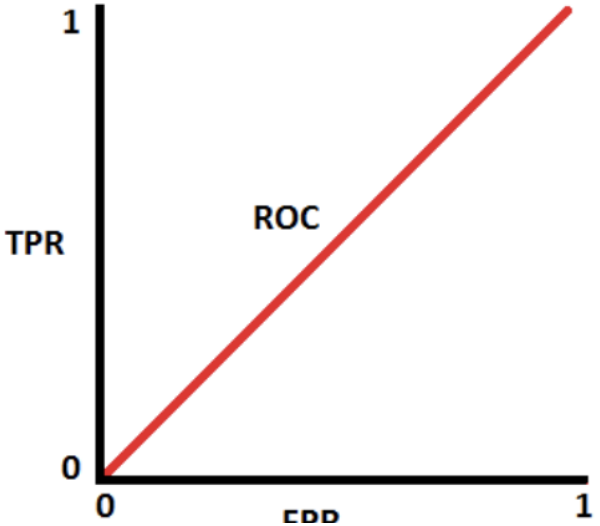

1
ROC
TPR
0
0
FPR
1

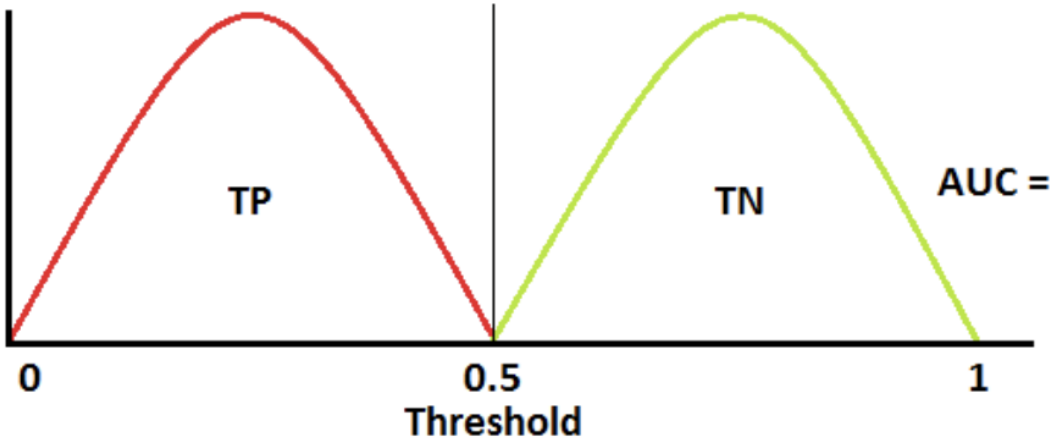

TN
TP
AUC = 0.7
FN
FP
0.5
Threshold

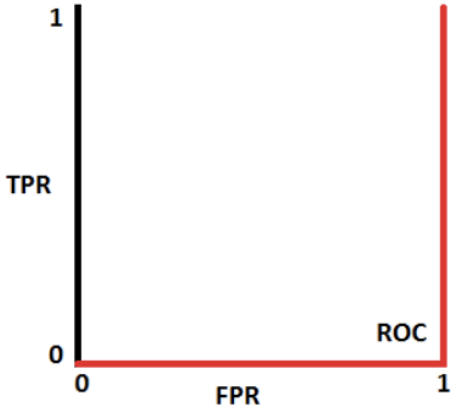

1
ROC
TPR
0
0
FPR
1

AUC


AUC = 0.5
0.5
Threshold


1
TPR
ROC
0
0
1


TP
TN
AUC = 0
0
0.5
1
Threshold


1
TPR
ROC
0
0
FPR
1

Additional Aspects

- In addition to accuracy-based measures, classifiers can also be compared with respect to the following additional aspects:
  - **Speed:** This refers to the computational costs involved in generating and using the given classifier.
  - **Robustness:** This is the ability of the classifier to make correct predictions given noisy data or data with missing values. Robustness is typically assessed with a series of synthetic data sets representing increasing degrees of noise and missing values.
  - **Scalability:** This refers to the ability to construct the classifier efficiently given large amounts of data. Scalability is typically assessed with a series of data sets of increasing size.
  - **Interpretability:** This refers to the level of understanding and insight that is provided by the classifier or predictor. Interpretability is subjective and therefore more difficult to assess.

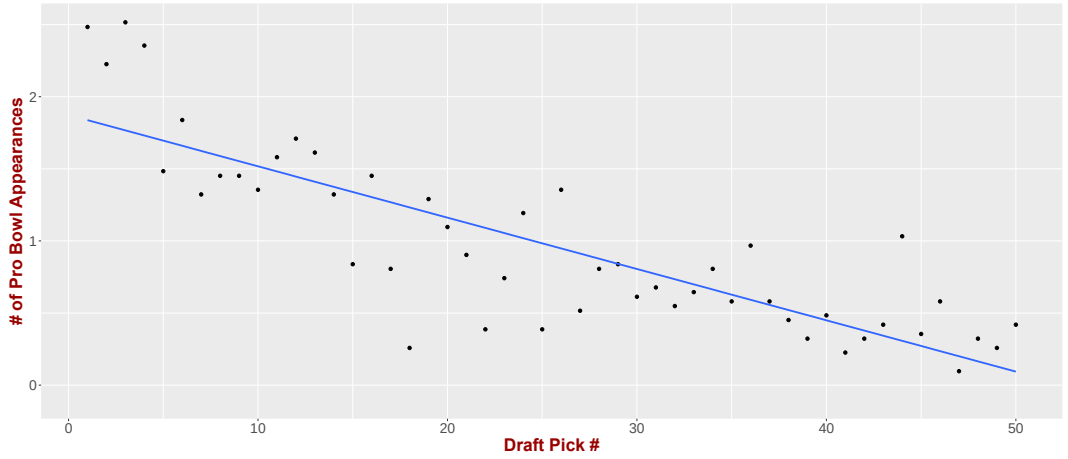
Outline

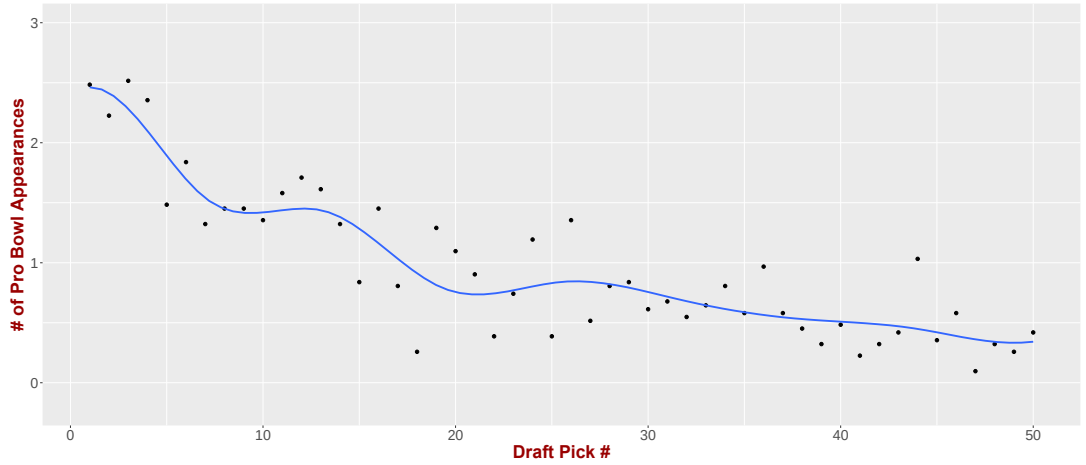
- Metrics for Evaluating Classifier Performance
- Bias-Variance Tradeoff
- Model selection and validation

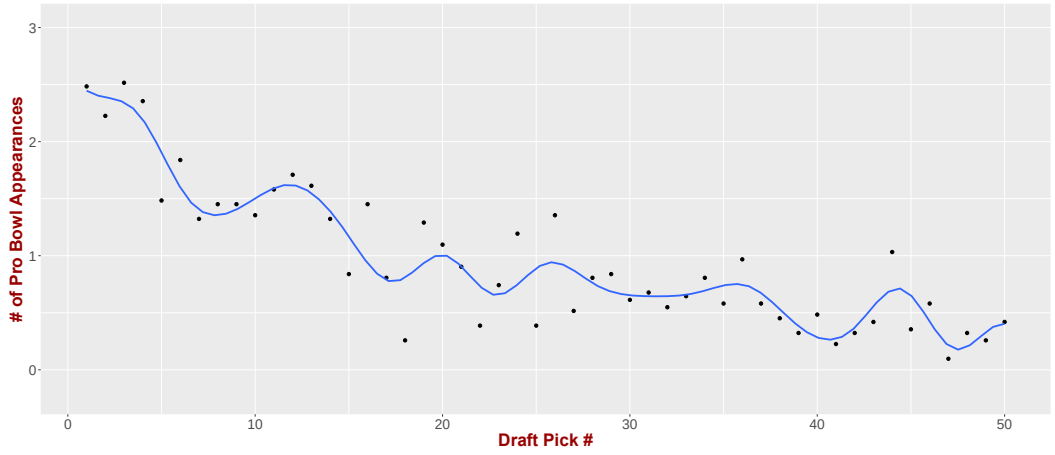
Underfitting and Overfitting

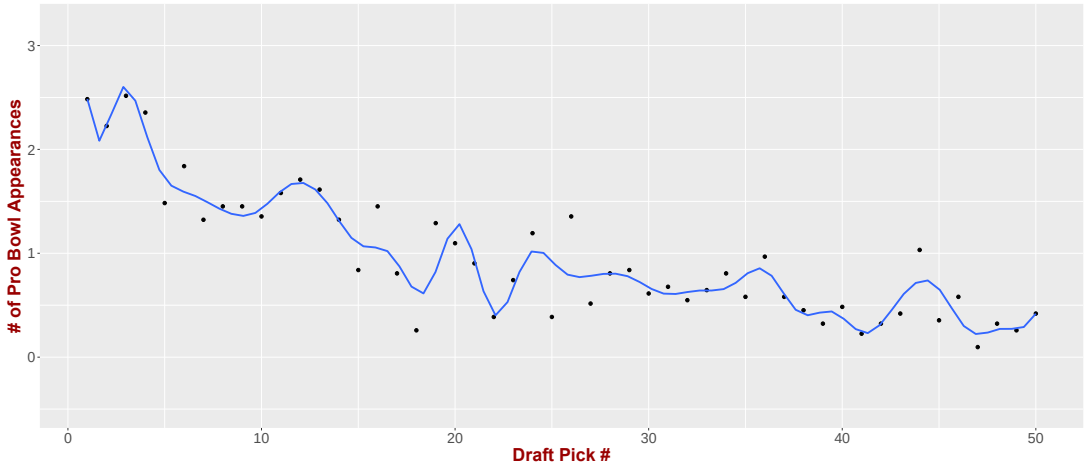
- **Underfitting** occurs when the model is unable to match the input data to the target data. This happens when the model is not complex enough to match all the available data and performs poorly with the training dataset.
- **Overfitting** relates to instances where the model tries to match non-existent data. This occurs when dealing with highly complex models where the model will match almost all the given data points and perform well in training datasets. However, the model would not be able to generalize the data point in the test data set to predict the outcome accurately.

Underfitting and Overfitting









Bias

- We can define **bias** as the error between average model prediction and the ground truth.

- It describes how well the model matches the training data set:
  - A model with a higher bias would not match the data set closely.
  - A low bias model will closely match the training data set.

Bias

- Characteristics of a high bias model include:
  - Failure to capture proper data trends
  - Potential towards underfitting
  - More generalized/overly simplified
  - High error rate

Variance

- **Variance** is the variability in the model prediction—how much the ML function can adjust depending on the given data set.
- Characteristics of a high variance model include:
  - Noise in the data set
  - Potential towards overfitting
  - Complex models
  - Trying to put all data points as close as possible

Bias-Variance Tradeoff

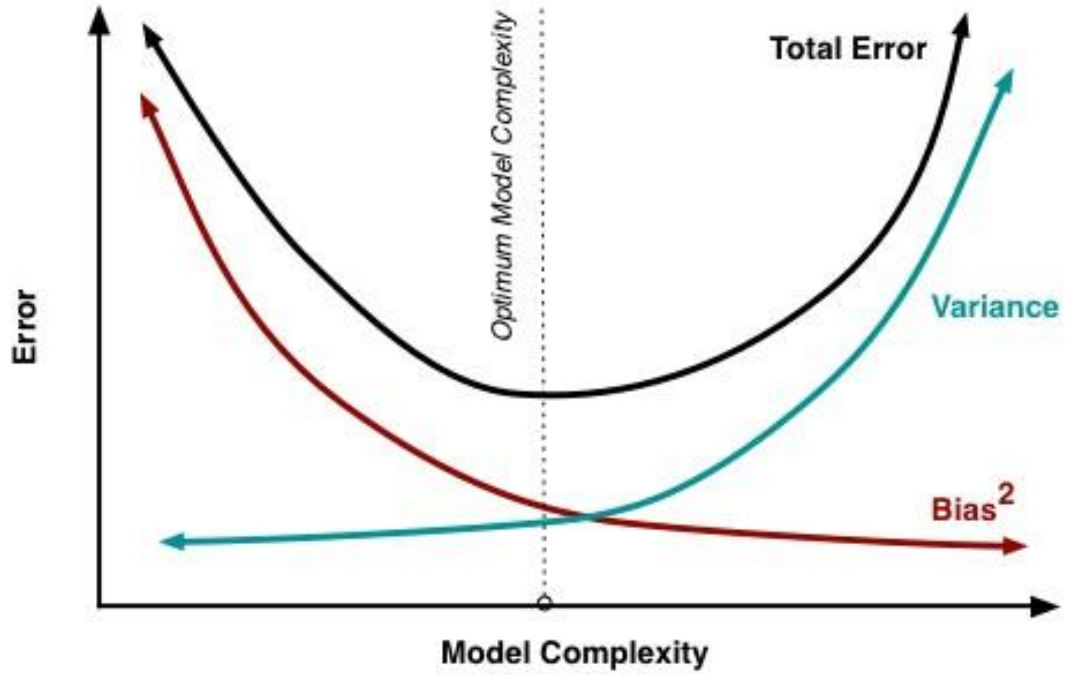
- Let’s consider a regression model and its mean squared error MSE (epsilon is noise):

$$MSE = bias^2 + variance + \varepsilon$$

What do we want ?

Bias-Variance Tradeoff

- If we want to minimize MSE, we need to minimize both bias and variance
  - However, when bias gets smaller, variance increases
  - vice versa



Outline

- Metrics for Evaluating Classifier Performance
- Bias-Variance Tradeoff
- Model selection and Validation

Introduction

- Two issues in pattern recognition techniques
  - Model Selection: How do we select the “optimal” parameter(s) for a given classification problem?
  - Validation: Once we have chosen a model, how do we estimate its true error rate?

Introduction

- If we had access to an unlimited number of examples, these questions would have a straightforward answer.
  - Choose the model that provides the lowest error rate on the entire population.

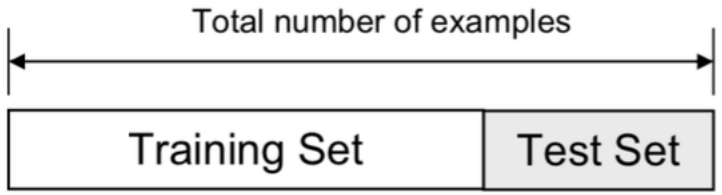
- However, in real applications only a finite set of examples is available
  - This number is usually smaller than we would hope for!
  - Why? Data collection is a very expensive process

Introduction

- The techniques to make the best use of your (limited) data for
  - Training
  - Model selection
  - Performance estimation

The holdout method

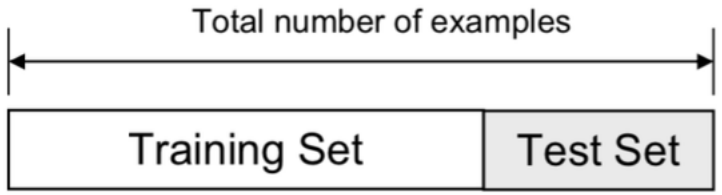
- Split dataset into two groups
  - **Training set:** used to train the classifier
  - **Test set:** used to estimate the error rate of the trained classifer



- The holdout method has two basic drawbacks
  - In problems where we have a sparse dataset we may not be able to afford the “luxury” of setting aside a portion of the dataset for testing
  - Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an “unfortunate” split

The holdout method

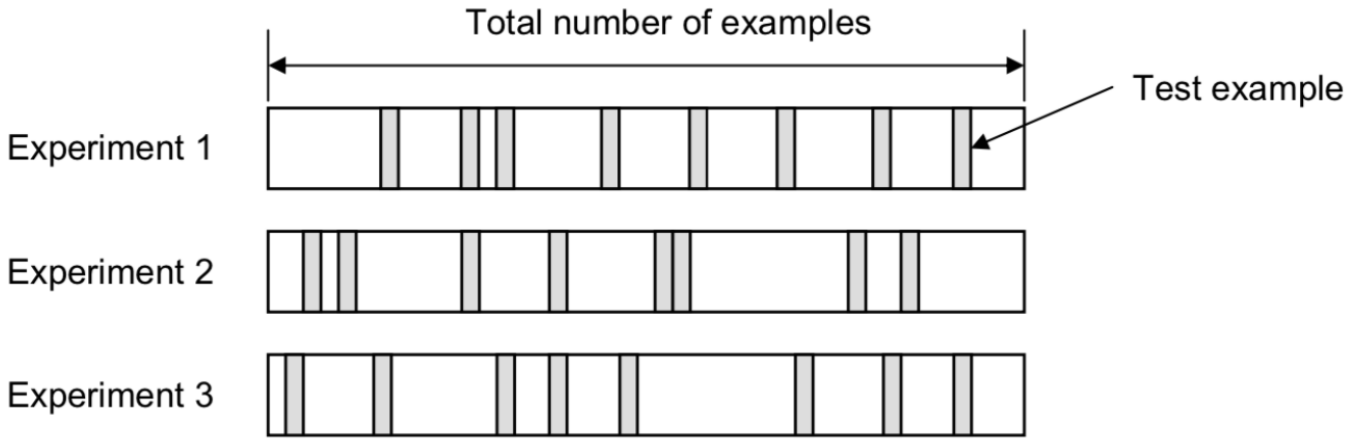
- Split dataset into two groups
  - **Training set:** used to train the classifier
  - **Test set:** used to estimate the error rate of the trained classifer



- The limitations of the holdout can be overcome with a family of re-sampling methods at the expense of higher computational cost
  - **Cross Validation**
    - Random Subsampling; K-Fold Cross-Validation; Leave-one-out Cross-Validation
  - Bootstrap

Random Subsampling

- Random Subsampling performs K data splits of the entire dataset.
  - Each data split randomly selects a (fixed) number of examples without replacement
  - For each data split we retrain the classifier from scratch with the training examples and then estimate e_i with the test examples



- The true error estimate is obtained as the average of the separate estimates e_i

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

K-fold Cross-validation

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

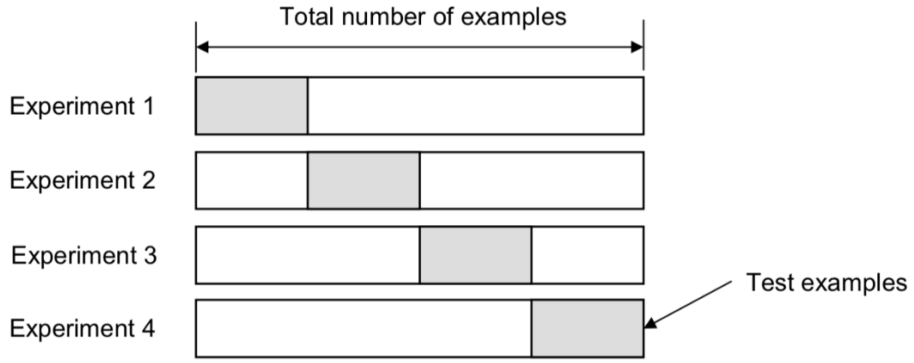


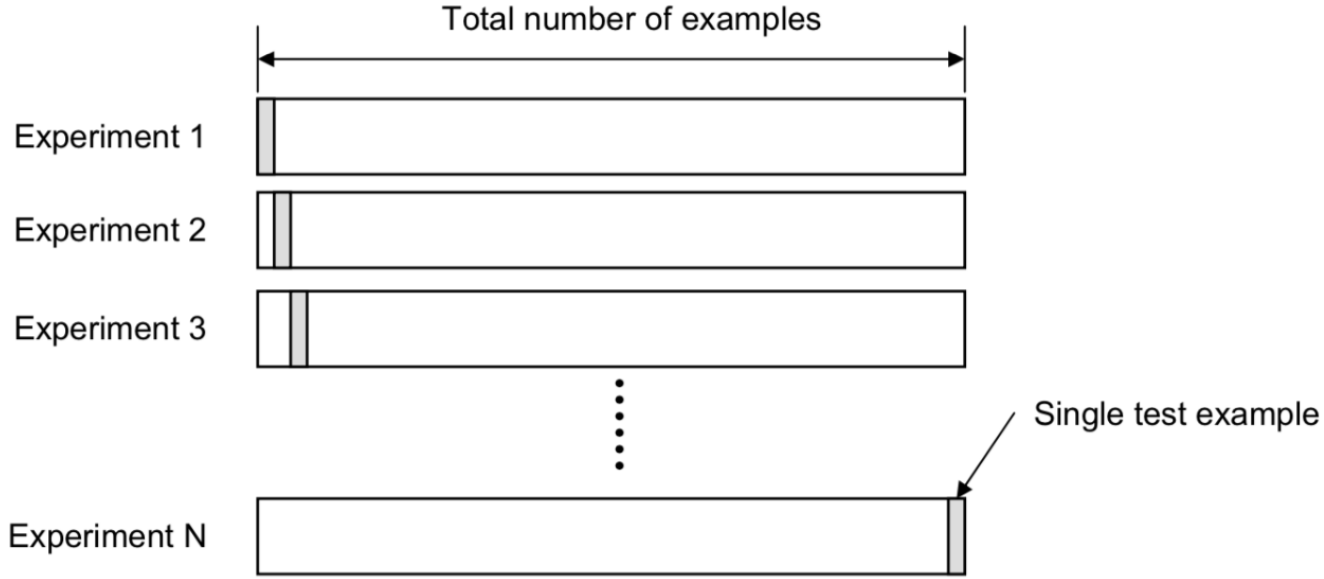
Figure: When $K = 4$

- K-Fold Cross validation is similar to Random Subsampling
  The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- As before, the true error is estimated as the average error rate on test examples

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

Leave-one-out Cross Validation

- Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples
  - For a dataset with N examples, perform N experiments
  - For each experiment use N-1 examples for training and the remaining example for testing



- As before, the true error is estimated as the average error rate on test examples

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

Cross-Validation

- What is the key problem of K-fold cross validation?

Cross-Validation

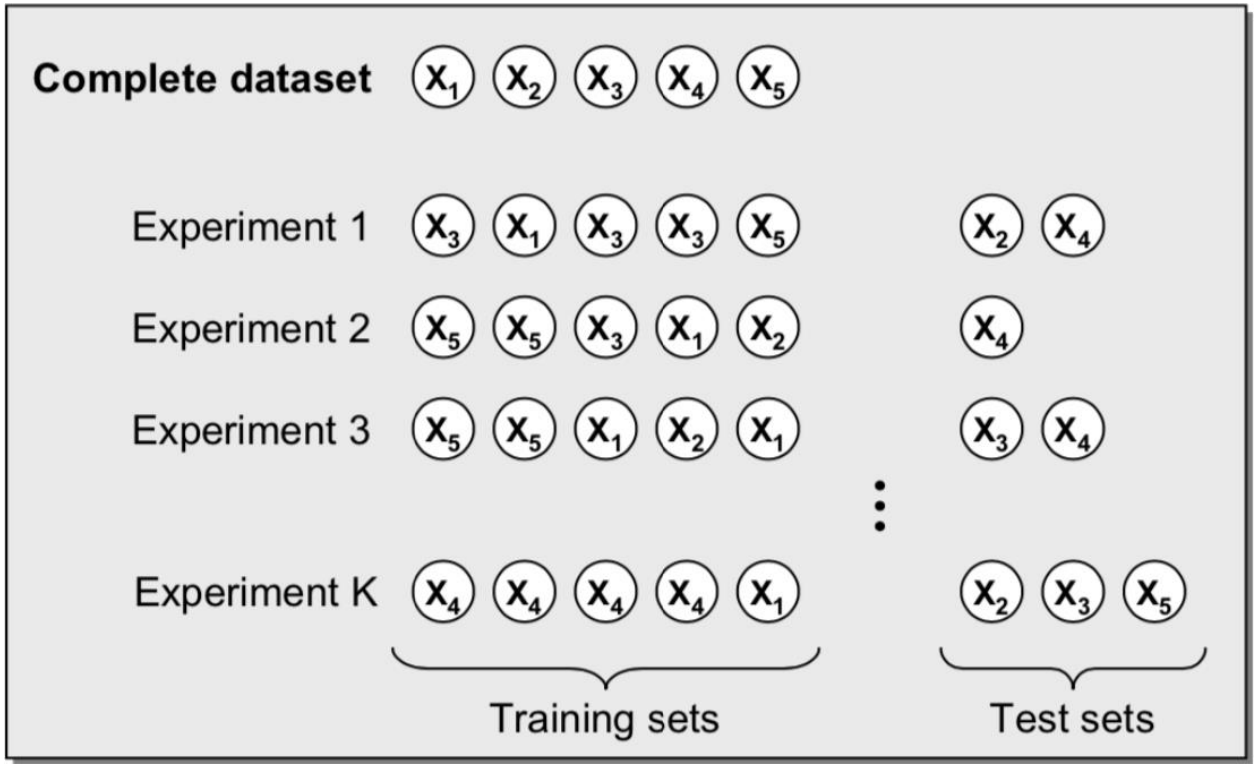
- What is the key problem of K-fold cross validation?
  - How to choose K?

How to choose K

- With a large number of folds
  - The bias of the true error rate estimator will be small (the estimator will be very accurate)
  - The variance of the true error rate estimator will be large
  - The computational time will be very large as well (many experiments)
- With a small number of folds
  - The number of experiments and, therefore, computation time are reduced
  - The variance of the estimator will be small
  - The bias of the estimator will be large (conservative or smaller than the true error rate)
- In practice, the choice of the number of folds depends on the size of the dataset
  - For large datasets, even 3-Fold Cross Validation will be quite accurate
  - For very sparse datasets, we may have to use leave-one-out in order to train on as many examples as possible
- A common choice for K-Fold Cross Validation is *K=10*

The Bootstrap

- The bootstrap is a resampling technique with replacement
  - From a dataset with N examples
    - Randomly select (with replacement) N examples and use this set for training
      The remaining examples that were not selected for training are used for testing
    - *This value is likely to change from fold to fold*
- Repeat this process for a specified number of folds (K)
- As before, the true error is estimated as the average error rate on test examples



Three-way data splits

- If model selection and true error estimates are to be computed simultaneously, the data needs to be divided into three disjoint sets [Ripley, 1996]
  - Training set: a set of examples used for learning: to fit the parameters of the classifier
  - Validation set: a set of examples used to tune the parameters of a classifier (Hyper parameter: epoch, batch-size, KNN-K, depth of the tree)
  - Test set: a set of examples used only to assess the performance of a fully-trained classifier

- Why separate test and validation sets?
  - The error rate estimate of the final model on validation data will be biased (smaller than the true error rate) since the validation set is used to select the final model
  - After assessing the final model on the test set, YOU MUST NOT tune the model any further!

Three-way data splits

1. Divide the available data into training, validation and test set
2. Select architecture and training parameters
3. Train the model using the training set
4. Evaluate the model using the validation set
5. Repeat steps 2 through 4 using different architectures and training parameters
6. Select the best model and train it using data from the training and validation sets
7. Assess this final model using the test set

Summary

- Metrics for Evaluating Classifier Performance
  - Confusion matrix
  - Accuracy
  - Sensitive and specification
  - ROC and AUC
- Bias-Variance Tradeoff
  - Underfitting and Overfitting
  - Bias and variance
  - Bias-variance tradeoff
- Model selection and validation
  - Holdout method
  - Ramdom Subsampling
  - K-fold Cross validation
  - Leave-one-out Cross Validation
  - Bootstrap