Eighth Session

Alireza Moradi

Evaluation metrics for Classification

- We want some metrics that evaluate our models' performance.

- Example:

Observation	Midterm	Statistics score	Number of absence	Pass?
1	15	16	2	pass
2	13	11	7	Fail
3	11	15	0	pass
4	3	15	6	Fail
5	6	12	5	Fail
6	20	18	3	pass
7	19	15	2	pass
8	20	20	1	pass
…	…	…	…	…
100	14	18	0	pass

Train set or test set?

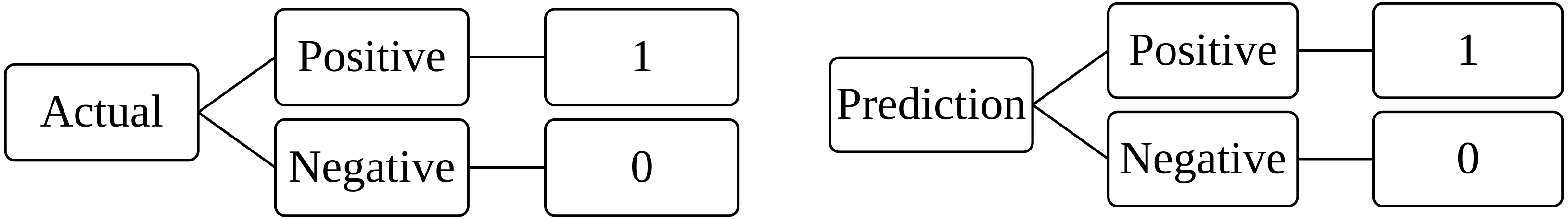
Predicted by model(classifier)

Observation	Actual	Predicted
1	Pass	Pass
2	Fail	Pass
3	Pass	Pass
4	Fail	Pass
5	Fail	Fail
6	Pass	Fail
7	Pass	Pass
8	Pass	Pass
9	Pass	Pass
10	Pass	Pass

Confusion matrix; Binary

- A **confusion matrix** represents the prediction summary in matrix form. It shows how many prediction are correct and incorrect per class. It helps in understanding the classes that are being confused by model as other class.

- In a binary classification problem, we have two class; positive(1) & negative(0).



Confusion matrix; Binary

- In a binary classification problem with classes named *Negative* and *Positive*, Confusion matrix will be:

		Predicted	
		Negative (N) **-**	**Positive (P)** **+**
Actual	**Negative** **-**	True Negative (**TN**)	False Positive (**FP**) **(Type 1 Error)**
	Positive **+**	False Negative (**FN**) **(Type 2 Error)**	True Positive (**TP**)

Confusion matrix; Binary

- Example: (Here our classes are named *Pass* and *Fail*)

Observation	Actual	Predicted
1	Pass	Pass
2	Fail	Pass
3	Pass	Pass
4	Fail	Pass
5	Fail	Fail
6	Pass	Fail
7	Pass	Pass
8	Pass	Pass
9	Pass	Pass
10	Pass	Pass

Confusion matrix

		Predicted	
		Pass	Fail
Actual	Pass	6	1
	Fail	2	1

Confusion matrix; Binary

- Example: (Here our classes are named *Cat* and *Dog*)

Observation	Actual	Predicted
1	Cat	Cat
2	Dog	Cat
3	Cat	Cat
4	Dog	Dog
5	Dog	Dog
6	Cat	Cat
7	Cat	Cat
8	Dog	Dog
9	Cat	Dog
10	Dog	Dog

Confusion matrix

		Predicted	
		Cat	Dog
Actual	Cat		
	Dog		

Confusion matrix; Multiclass

- Example: (Here our classes are named *Green*, *Blue*, and *Red*)

Observation	Actual	Predicted
1	Green	Green
2	Red	Red
3	Blue	Blue
4	Blue	Blue
5	Red	Red
6	Green	Red
7	Red	Red
8	Green	Green
9	Green	Blue
10	Green	Green
11	Red	Red
12	Red	Blue
13	Red	Red
14	Blue	Blue
15	Blue	Blue

Confusion matrix

		Predicted		
		Green	Blue	Red
Actual	Green			
	Blue			
	Red			

Confusion matrix; Binary

1. **TP = True Positive** = The number of samples that were correctly predicted as positive.

2. **TN = True Negative** = The number of samples that were correctly predicted as negative.

		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

3. **FP = False Positive** = The number of samples that were incorrectly predicted as positive.

4. **FN = False Negative** = The number of samples that were incorrectly predicted as negative.

Confusion matrix; metrics

- **Accuracy = Recognition** rate:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Accuracy} = \frac{1 + 6}{1 + 6 + 1 + 2} = 70\% \uparrow$$

		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

		Predicted	
		Pass	Fail
Actual	Pass	6	1
	Fail	2	1

Confusion matrix; metrics

- **Error rate = misclassification** rate:

$$\text{misclassification rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

$$\text{misclassification rate} = \frac{1 + 2}{1 + 6 + 1 + 2} = 30\% \downarrow$$

		Predicted	
		Pass	Fail
Actual	Pass	6	1
	Fail	2	1

Confusion matrix; metrics

- **Sensitivity = true positive rate = Recall:**

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

$$\text{Recall} = \frac{1}{1 + 2} = 33\% \downarrow$$

		Predicted	
		Pass	Fail
Actual	Pass	6	1
	Fail	2	1

Confusion matrix; metrics

- **Precision:**

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

$$\text{Precision} = \frac{1}{1 + 1} = 50\% \downarrow$$

		Predicted	
		Pass	Fail
Actual	Pass	6	1
	Fail	2	1

Confusion matrix; metrics

- **F= F_1 = F-score =** harmonic mean of precision and recall**:**

$$F = 2 \times \frac{\text{Precision . Recall}}{\text{Precision} + \text{Recall}}$$

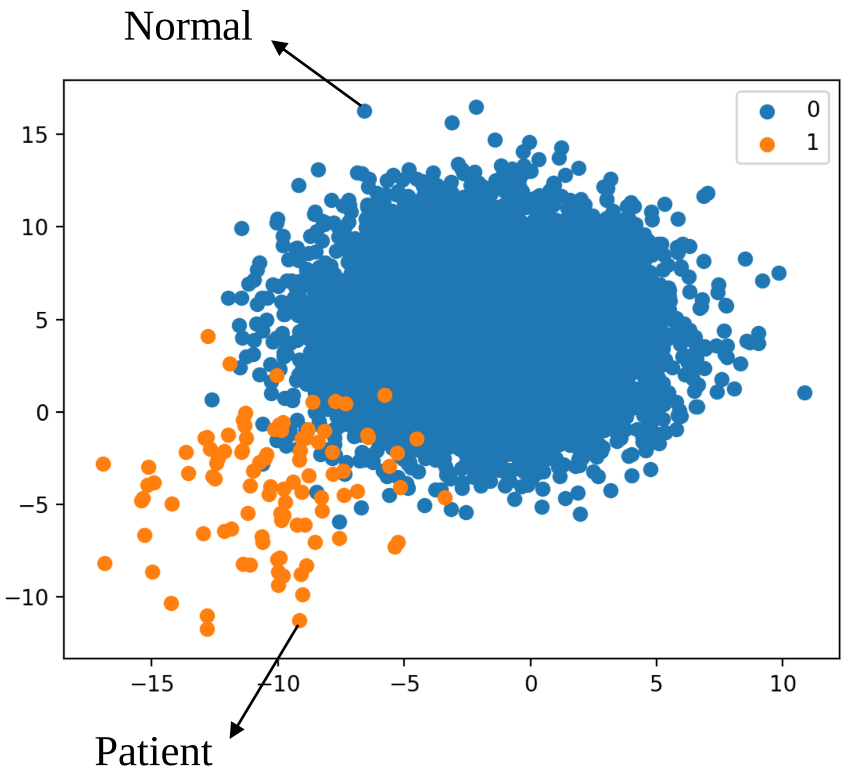
		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

$$F = 2 \times \frac{0.33 \times 0.5}{0.33 + 0.5} = 38\ \% \downarrow$$

		Predicted	
		Pass	Fail
Actual	Pass	6	1
	Fail	2	1

Imbalanced data

- A classification data set with skewed class proportions is called **imbalanced**.
- Classes that make up a large proportion of the data set are called **majority** classes. Those that make up a smaller proportion are **minority** classes.
- Minority class is usually more important to us.
- e.g., diabetes



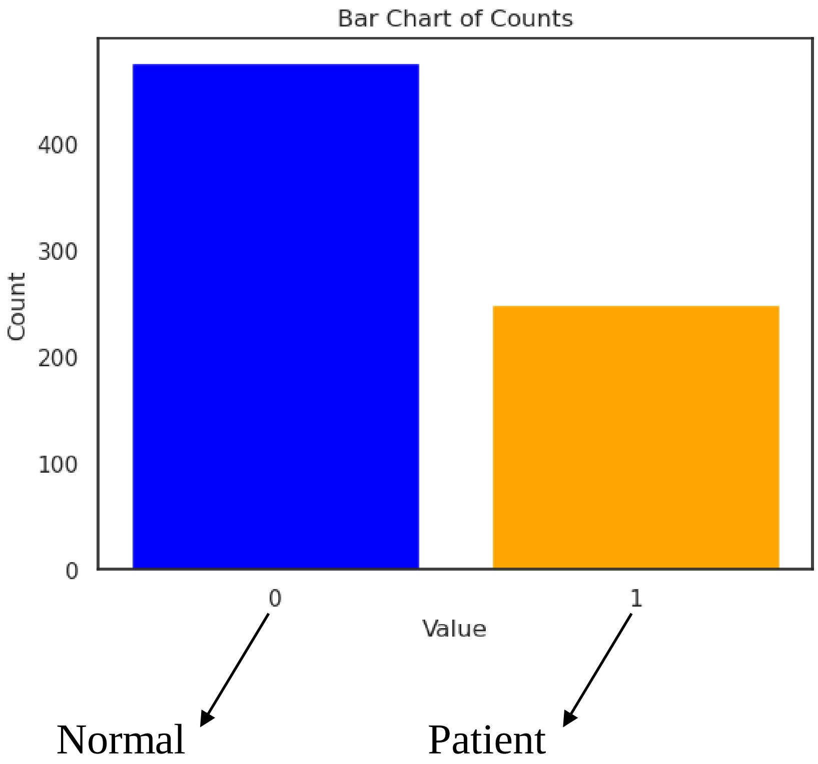
Imbalanced data

- The dataset is available on Kaggle here.

Patient

index	Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	DPF	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

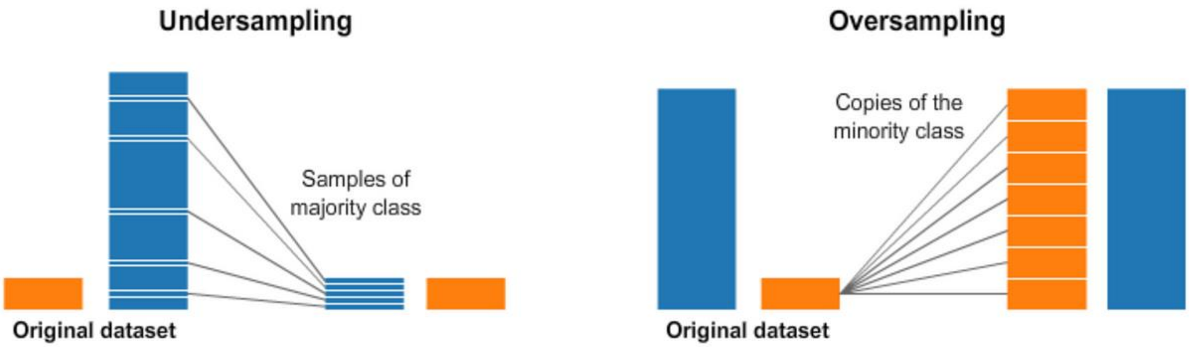
Imbalanced data



Imbalanced data

- On of the techniques for tackling this problem is **Resampling**:
  1. **Under sampling**: Reduce the size of the majority class by randomly removing data points to match the size of the minority class.

  2. **Oversampling**: Increase the size of the minority class by replicating existing data points or generating synthetic data points.



Imbalanced data

- **Precision**: Out of all the people we predicted to have diabetes, what proportion have it?

- **Recall**: Out of all the people who have diabetes, what proportion did we correctly identify as having it?

$$\begin{cases} \text{Precision} = \dfrac{\text{TP}}{\text{TP} + \text{FP}} = \\ \\ \text{Recall} = \dfrac{\text{TP}}{\text{TP} + \text{FN}} = \end{cases}$$

		Predicted	
		Normal	Patient
Actual	Normal	47	8
	Patient	15	20

Logistic Regression with imbalanced classes

Imbalanced data

- In binary classification, we typically assign the label "positive" to the class of interest. (the more important class)

- Compare two models:

		Predicted	
		Normal	Patient
Actual	Normal	47	8
	Patient	15	20

Logistic Regression with imbalanced classes

Accuracy = 74%

		Predicted	
		Normal	Patient
Actual	Normal	22	9
	Patient	7	21

Logistic Regression with balanced classes (Under sampled)

Accuracy = 72%

Imbalanced data

		Predicted	
		Normal	Patient
Actual	Normal	47	8
	Patient	15	20

(1) Logistic Regression with imbalanced classes

		Predicted	
		Normal	Patient
Actual	Normal	22	9
	Patient	7	21

(2) Logistic Regression with balanced classes
(Under sampled)

model	(1)	**(2)**
Accuracy	**74%**	72%
Precision	**71%**	70%
Recall	57%	**75%**
F1 score	63%	**72%**

ROC Curve

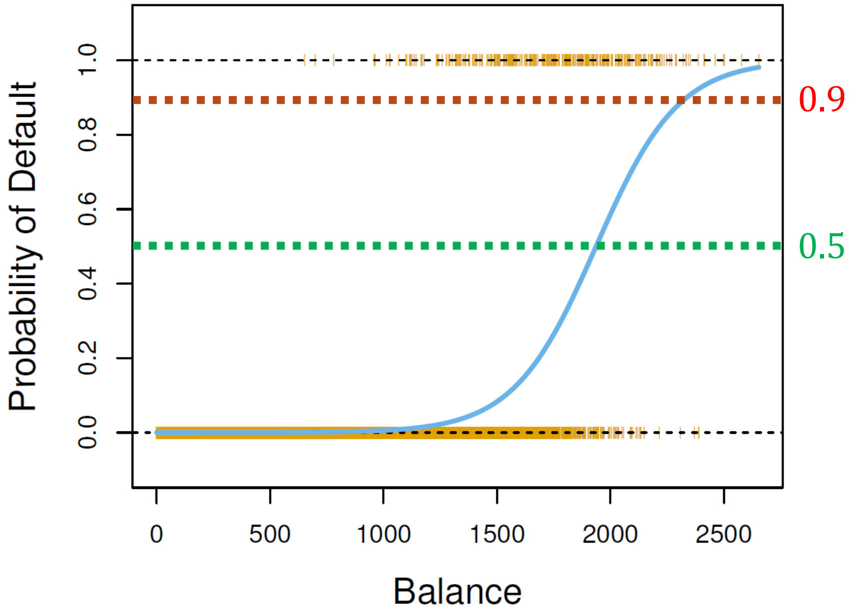
- An **ROC curve** (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.

- Logistic regression model:

$$P(X) = \frac{e^{\beta_0+\beta_1.X}}{1+e^{\beta_0+\beta_1.X}}$$

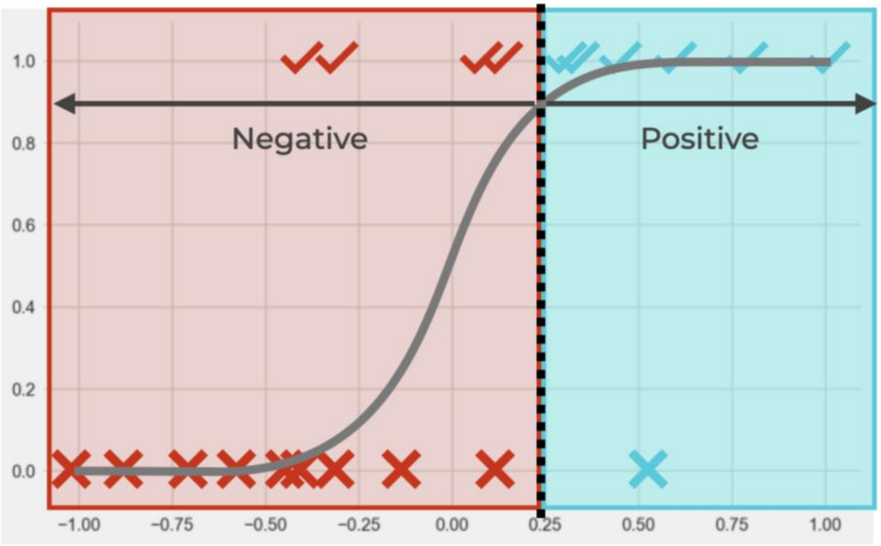
- How we classify observations?

$$if\ p(X) \geq Threshold \rightarrow \hat{y} = 1$$
$$if\ p(X) < Threshold \rightarrow \hat{y} = 0$$

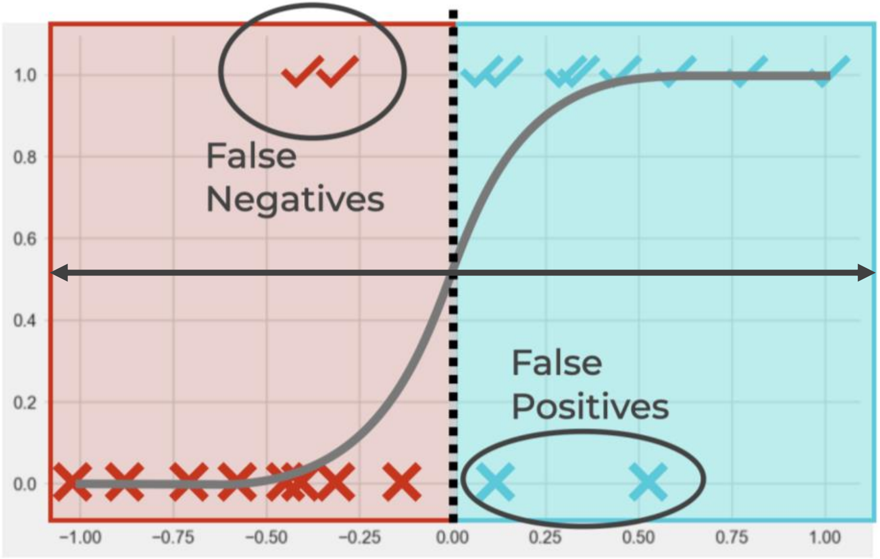


ROC Curve; Threshold

- Lowering the classification threshold classifies more items as positive.



0.9



0.5

ROC Curve

- ROC curve, plots two parameters for different thresholds:

  1. **True Positive Rate(Recall)**
  2. **False Positive Rate**

		Predicted	
		N	P
Actual	N	TN	FP
	P	FN	TP

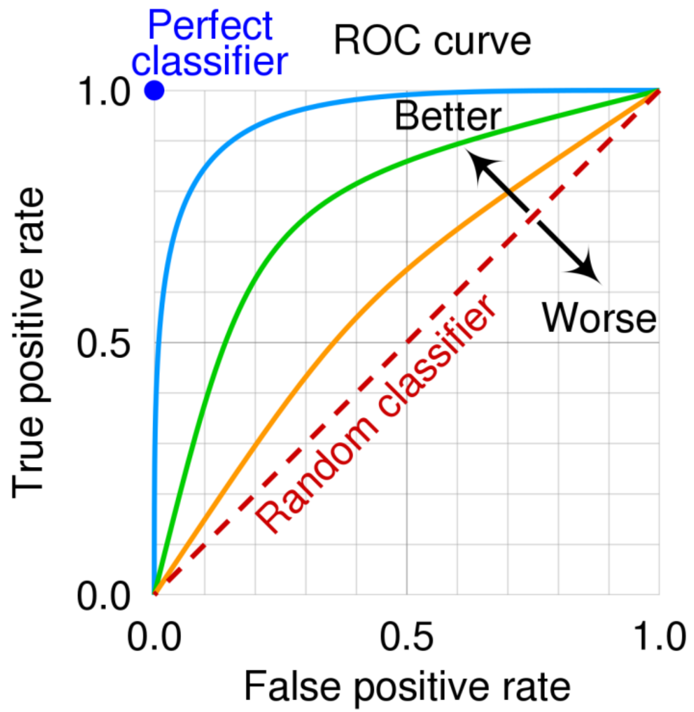
$$\text{True Positive Rate (Recall)} = \frac{\text{TP}}{\text{TP+FN}}$$

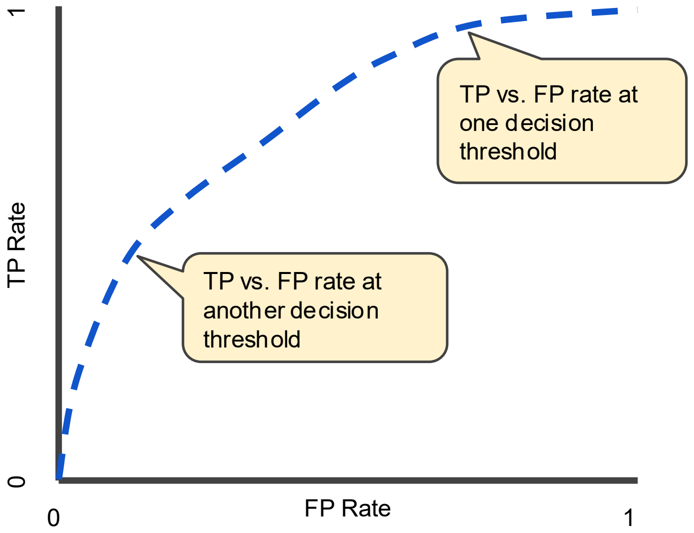
$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

- An ROC curve plots **TPR** vs. **FPR** at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

ROC Curve

- ROC curve helps us choose the best threshold.





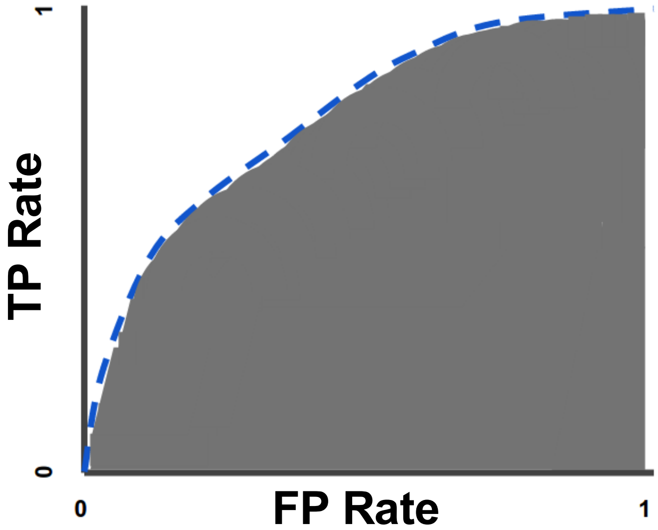
ROC AUC

- **AUC** stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve

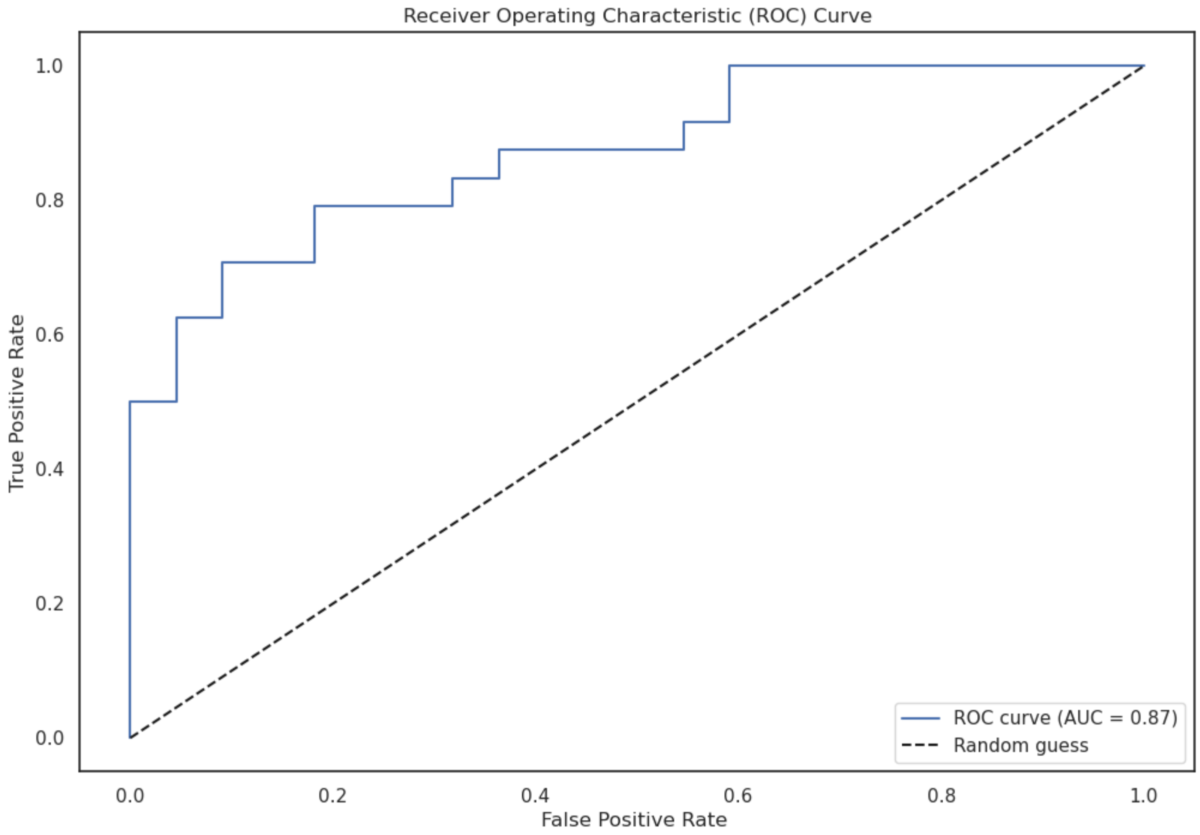
- **AUC** provides an aggregate measure of performance across all possible classification thresholds.

- 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.



ROC AUC



Receiver Operating Characteristic (ROC) Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (AUC = 0.87)
Random guess

Performance estimation

- Recall the distinction between the **test error** and the **training error**:

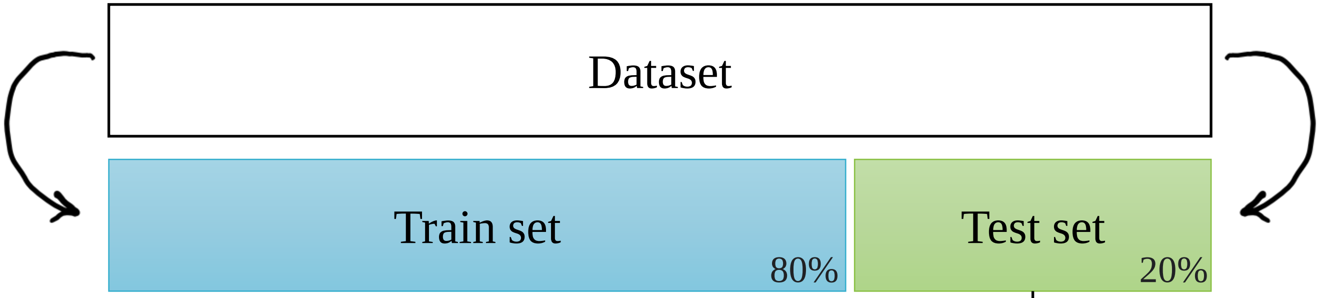
  "The test error is the average error that results from using a learning method to predict the response on a new observation, one that was not used in training the method."

- **Minimizing test error** (or **maximizing test performance**) is the primary objective in learning methods.

- However, we cannot directly measure the true test error since it involves unseen data. Therefore, we **estimate** the test error using various techniques like cross-validation or hold-out testing.

$$Test\ Error \Rightarrow \widehat{Test\ Error}$$

Hold-out



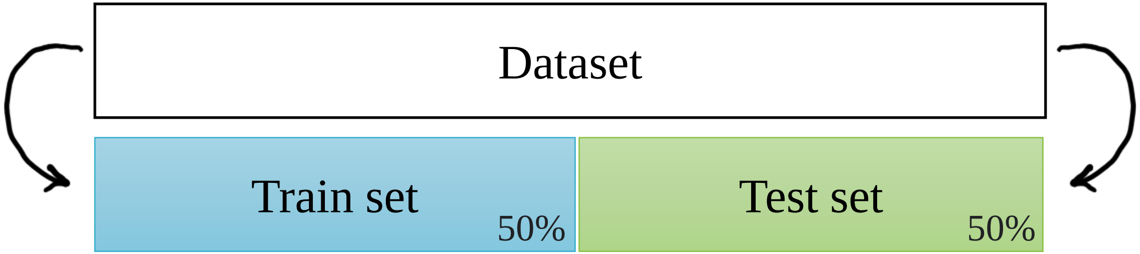
- Suppose we want to evaluate our models with accuracy

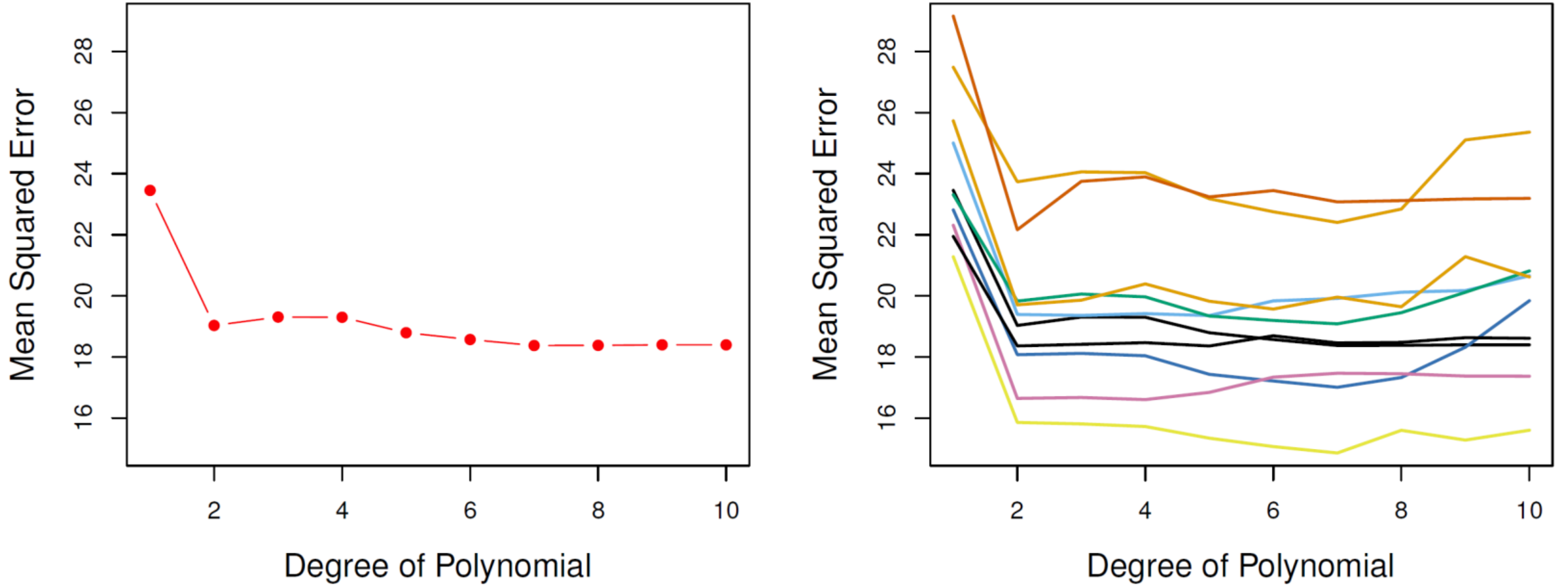
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Hold-out

- Simulation data:

✓ How to make this better?

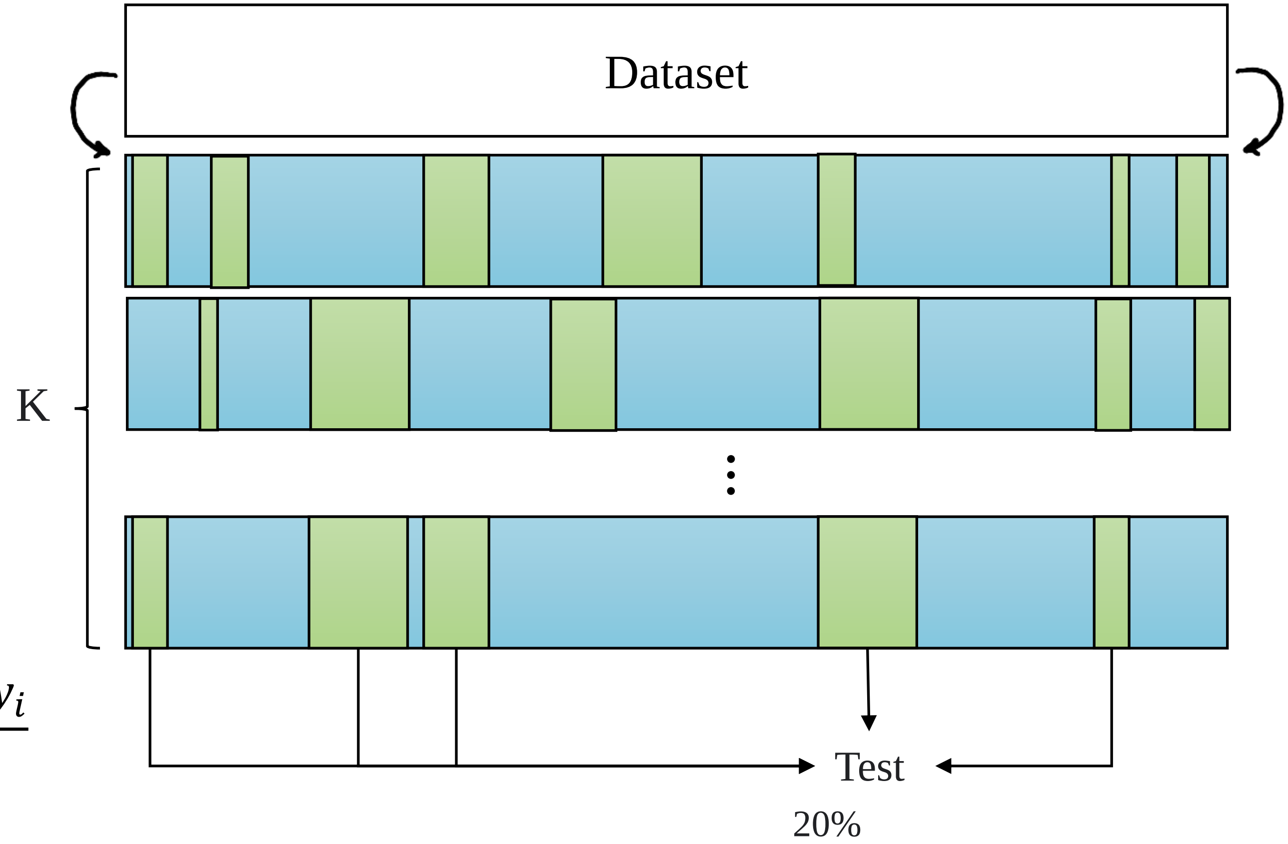




Random Sampling

- 20% test
- 80% train

- K times



$$\text{Accuracy} = \frac{\sum_{i=1}^{k} Accuracy_i}{k}$$

K-fold Cross validation

- **Cross-validation** is a resampling procedure used to evaluate machine learning models on a limited data sample.

- **K-fold cross-validation** is a specific implementation of cross-validation. K-fold Cross-validation is a widely used approach for estimating test error.

- Main idea is to randomly divide the data into K equal-sized parts. We leave out part k, fit the model to the other k-1 parts (combined), and then obtain predictions for the left-out kth part.

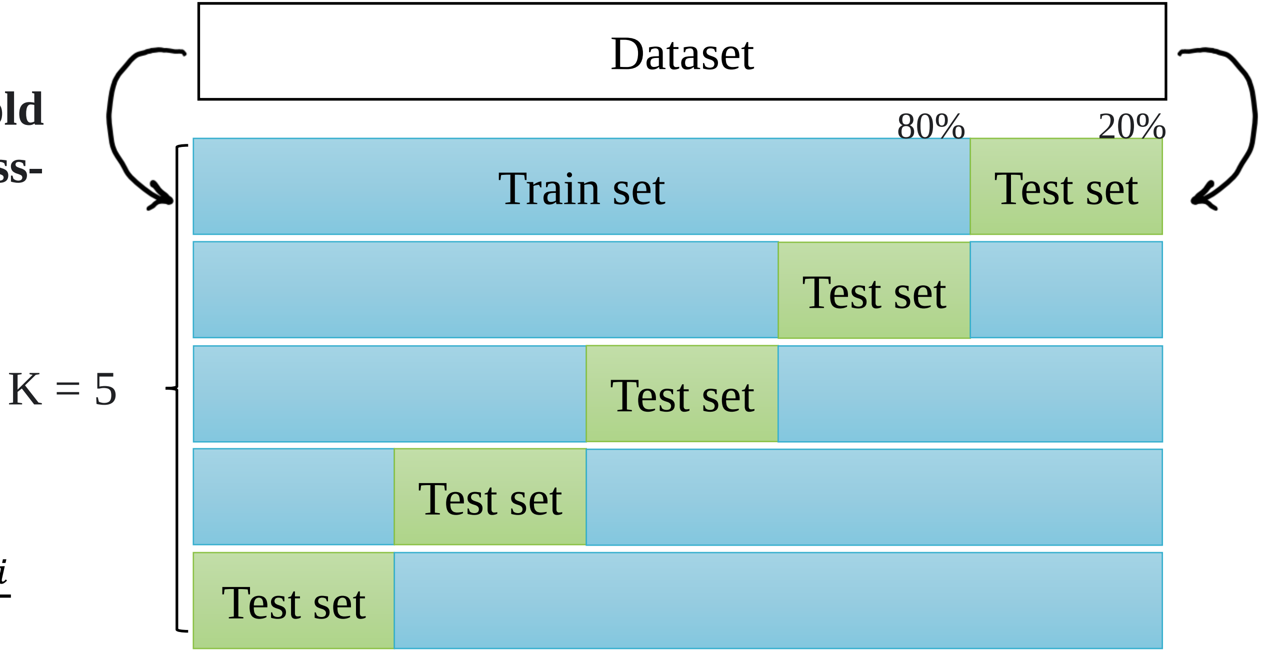
- This is done in turn for each part $k = 1, 2, \ldots, K$, and then the results are combined.

Cross validation

- ✓ K-fold cross-validation with K=5 is a popular technique.

- ✓ Setting K = n yields **n-fold** or **leave-one out cross-validation (LOOCV).**

$$\text{Accuracy} = \frac{\sum_{i=1}^{k} Accuracy_i}{k}$$



Cross validation