DEFENCE INSTITUTE OF ADVANCED TECHNOLOGY
DEEMED UNIVERSITY
शास्त्रेण
रक्षाम्
शस्त्रं प्रकरोति

Machine Learning for Cyber Security (CS-602)

L#04

Introduction – Bias-Variance Tradeoff – part1

By

Dr Sunita Dhavale

Syllabus

- Data Analytics Foundations: R programming, Python Basics -Expressions and Variables, String Operations, Lists and Tuples, Sets, Dictionaries Conditions and Branching, Loops, Functions, Objects and Classes, Reading/Writing files, Hand ling data with Pandas, Scikit Library, Numpy Library, Matplotlib, scikit programming for data analysis, setting up lab environment, study of standard datasets. Introduction to Machine Learning- Applications of Machine Learning, Supervised, unsupervised classification and regression analysis
- Python libraries suitable for Machine Learning Feature Extraction. Data pre-processing, feature analysis etc., Dimensionality Reduction & Feature Selection Methods, Linear Discriminant Analysis and Principal Component Analysis, tackle data class imbalance problem

Syllabus

- Supervised and regression analysis, Regression, Linear Regression, Non-linear Regression, Model evaluation methods, Classification, K-Nearest Neighbor, Naïve Bayes, Decision Trees, Logistic Regression, Support Vector Machines, Artificial Neural Networks, Model Evaluation. Ensemble Learning, Convolutional Neural Networks, Spectral Embedding, Manifold detection and Anomaly Detection
- Unsupervised classification K-Means Clustering, Hierarchical Clustering, Density-Based Clustering, Recommender Systems-Content-based recommender systems, Collaborative Filtering, machine learning techniques for standard dataset, ML applications, Case studies on Cyber Security problems that can be solved using Machine learning like Malware Analysis, Intrusion Detection, Spam detection, Phishing detection, Financial Fraud detection, Denial of Service Detection.

Text/Reference Books

1. Building Machine Learning Systems with Python – Willi Richert, Luis Pedro Coelho

2. Alessandro Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies Publication date :Aug 2, 2019, Packt, ISBN-13, 9781789804027

3. Machine Learning: An Algorithmic Perspective – Stephen Marsland

4. Sunita Vikrant Dhavale, "Advanced Image-based Spam Detection and Filtering Techniques", IGI Global, 2017

5. Soma Halder , Sinan Ozdemir, Hands-On Machine Learning for Cybersecurity: Safeguard your system by making your machines intelligent using the Python ecosystem, By Publication date : Dec 31, 2018, Packt, ISBN-13 :9781788992282

1. Stuart Russell, Peter Norvig (2009), "Artificial Intelligence – A Modern Approach", Pearson Elaine Rich & Kevin Knight (1999), "Artificial Intelligence", TMH, 2nd Edition

2. NP Padhy (2010), "Artificial Intelligence & Intelligent System", Oxford

3. ZM Zurada (1992), "Introduction to Artificial Neural Systems", West Publishing Company

4. Research paper for study (if any) – White papers on multimedia from IEEE/ACM/Elsevier/Spinger/ Nvidia sources.

Lab assignments

1	**Python Programming part-1**
2	Python Programming part-2
3	Study and Implement Linear Regression Algorithm for any standard dataset like in cyber security domain
4	Study and Implement KMeans Algorithm for any standard dataset in cyber security domain
5	Study and Implement KNN for any standard dataset in cyber security domain
6	Study and Implement ANN for any standard dataset in cyber security domain
7	Study and Implement PCA for any standard dataset in cyber security domain
8	Case Study: Use of ML along with Fuzzy Logic/GA to solve real world Problem in cyber security domain
9	Mini assignment: Apply ML along with PSO/ACO to solve any real world problem in cyber security domain
10	ML Practice Test – 1 Quiz

Defence Institute of Advanced Technology

School of Computer Engineering & Mathematical Sciences

SEMESTER-I TIME TABLE (AUTUMN 2024)$

PROGRAMMES: (I) CS [M.TECH IN CYBER SECURITY] (II) AI [M.TECH CSE (ARTIFICIAL INTELLIGENCE)] BATCH: 2024-2026

Lecture Day	L1 0900-1000	L2 1000-1100	L3 1100-1200	L4 1200-1300		L4 1400-1500	L4 1500-1600	L4 1600-1700	L4 1700-1800
Monday	CE-602 (AI) CS-602 (CS)	CE-604 (AI) CS-603 (CS)	CE-601 (AI) CS-604 (CS)	CE-601 (AI) LAB CS-603 (CS)	Lunch Break 1300-1400	LAB CE-601 (AI) LAB CS-602 (CS)		AM607	
Tuesday	CE-603 (AI) LAB CS-603 (CS)	CE-602 (AI) CS-602 (CS)	CE-601 (AI) CS-605 (CS)	CE-604 (AI) CS-604 (CS)		PGC 601		AM607	
Wednesday	CS-605 (CS)	CE-603 (AI) CS-602 (CS)	CE-602 (AI) CS-603 (CS)	CE-604 (AI) CS-604 (CS)		CE-605(AI) LAB CS-605 (CS)	LAB CS-605 (CS)	AM607	
Thursday	LAB CE-604 (AI) CS-603 (CS)	LAB CE-604 (AI) CS-605 (CS)	LAB CE-602 (AI) CS-601 (CS)	CE-603 (AI) CS-601 (CS)		PGC 601		AM607	
Friday	LAB CE-603 (AI) LAB CS-601 (CS)		LAB CE-602 (AI) CS-601 (CS)	LAB CS-604 (CS)		CE-605(AI) LAB CS-604 (CS)	CE-605(AI)	LAB CE-605(AI)	

COURSE CODE & COURSE NAME		FACULTY
Programme: CS [M.Tech in Cyber Security] Classroom: Arjun	Programme: AI [M.Tech CSE (Artificial Intelligence)] Classroom: Kaveri	
CS-601 Data Security & Privacy	CE-601 Responsible Artificial Intelligence;	MJN: Dr. Manisha J. Nene
CS-602 ML for Cyber Security	CE-604 Practical Machine Learning;	SVD: Dr. Sunita V. Dhavale
CS-605 Network and Cloud Security	CE-602 Intelligent Algorithms	CRS: Prof. CRS Kumar
CS-604 Advanced System Security	---------	DVV: Dr. Deepti V. Vidyarthi
CS-603 Applied Cryptography	---------	AM: Dr. Arun Mishra
---------	CE-603 Deep Neural Network;	US: Dr. Upasna Singh
---------	CE-605 Mathematics for ML;	Unit-2: Dr Upasna, Unit 4: Dr Sunita, Unit3:MJN, Unit 1: Faculty To be Nominated
AM-607 Mathematics for Engineers	AM-607 Mathematics for Engineers	OO/DS/DP: Dr Odellu O., Dr Dasari S., Dr. Debasis P.
PGC-601 Research Methodology	PGC-601 Research Methodology	Common Subject for All

$ TENTATIVE T.T. SUBJECT TO CHANGE

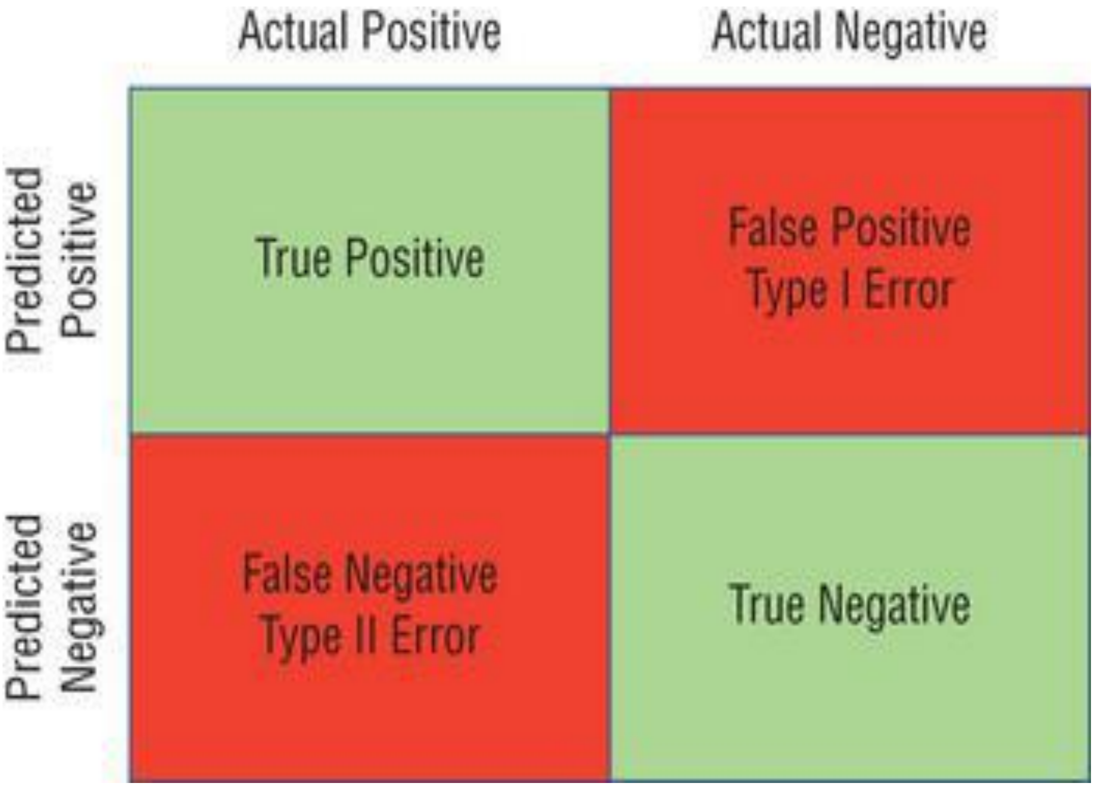
Program Coordinator,
M.Tech (CS & AI), Batch 2024-26

Director, SoCE&MS

Different types of error that may occur in a classification problem.

- When the predicted value for an observation differs from the actual value for that same observation, an *error* occurs.
- *False positive/ Type I errors* occur when the model labels an observation as predicted positive when it is, in reality, an actual negative.
- If the model identifies someone as likely lactose intolerant while they are, in reality, lactose tolerant.
- *False negative/Type II errors* occur when the model labels an observation as predicted negative when it is, in reality, an actual positive.
- If the model predicts someone as lactose tolerant when they are, in reality, lactose intolerant.

Evaluation-Types of errors



	Actual Positive	Actual Negative
Predicted Positive	True Positive	False Positive Type I Error
Predicted Negative	False Negative Type II Error	True Negative

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

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

Terms

True Positive Rate (TPR) or sensitivity or recall or hit rate is a measure of how many true positives were identified out of all the positives identified. TPR/Recall/Sensitivity, on the other hand, tells how many relevant items we selected.

True Negative Rate (TNR) or specificity is the ratio of true negatives and total number of negatives we have predicted.

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \qquad \mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Terms

Precision is defined as how many selected items are relevant. That is, how many of the predicted ones are actually correctly predicted.

precision= TP/(TP+FP)

Accuracy is the ratio of correct predictions and all the total predictions

If precision is closer to one, we are more accurate in our predictions.

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{P+N} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$

F-score

- F-score, or F1-score, is another measure of accuracy. Technically, it is the harmonic mean of precision and recall.
- The range for F1 Score is **[0, 1]**.
- It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances)

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} = ?$$

$$\text{Specificity} = \frac{TN}{TN+FP} = ?$$

		Prediction	
		1	0
Ground Truth	1	60	30
	0	80	20

60+30 = 90 cases in the dataset were class 1 (edge)

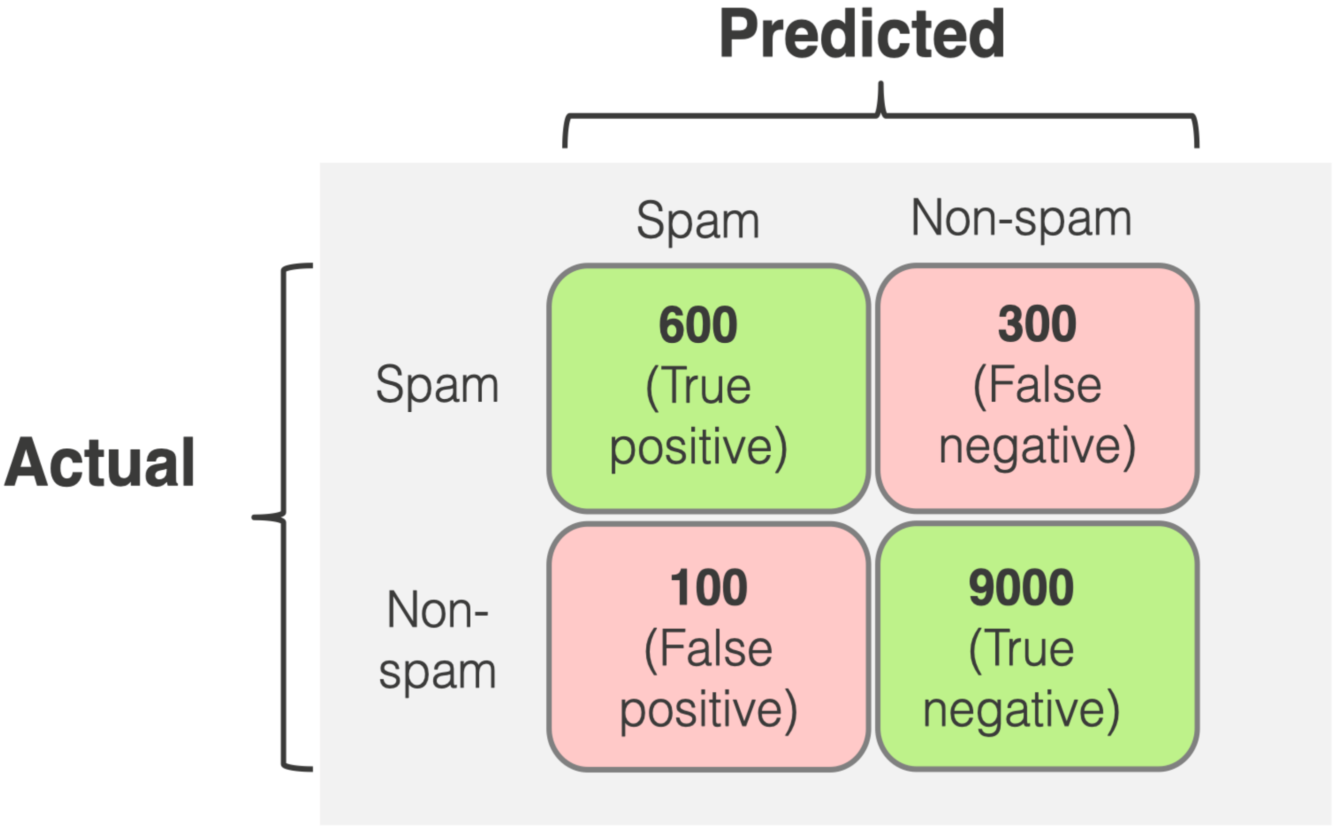
80+20 = 100 cases in the dataset were class 0 (non-edge)

90+100 = 190 examples (pixels) in the data overall

Exercise

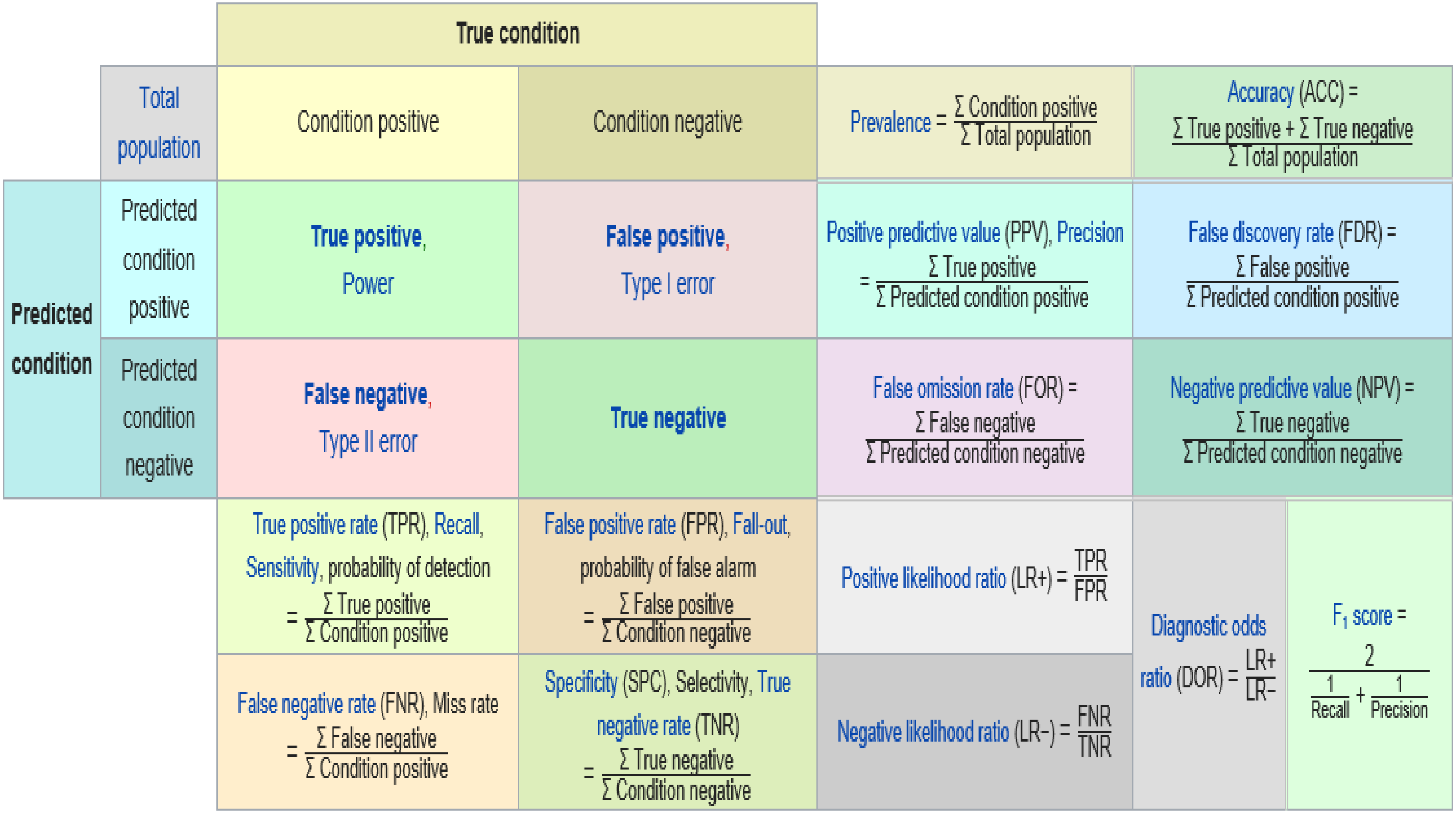
- You developed a machine learning algorithm that assesses a patient's risk of heart attack (a positive event) based on a number of diagnostic criteria. How would you describe each of the following events?
- Your model identifies a patient as likely to suffer a heart attack, and the patient does suffer a heart attack.
- Your model identifies a patient as likely to suffer a heart attack, and the patient does not suffer a heart attack.
- Your model identifies a patient as not likely to suffer a heart attack, and the patient does not suffer a heart attack.
- Your model identifies a patient as not likely to suffer a heart attack, and the patient does suffer a heart attack.

Calculate TPR(recall), FPR, Accuracy



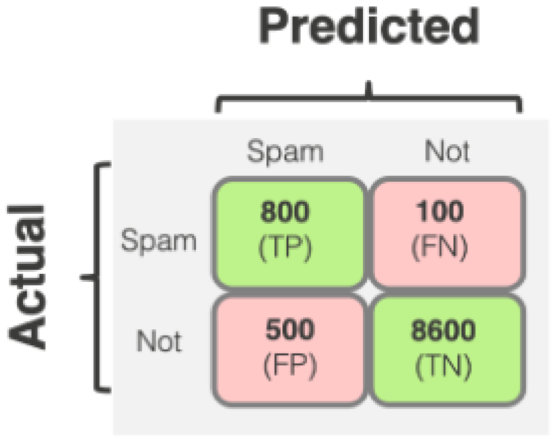
		Predicted	
		Spam	Non-spam
Actual	Spam	**600** (True positive)	**300** (False negative)
	Non-spam	**100** (False positive)	**9000** (True negative)

Confusion Matrix Parameters



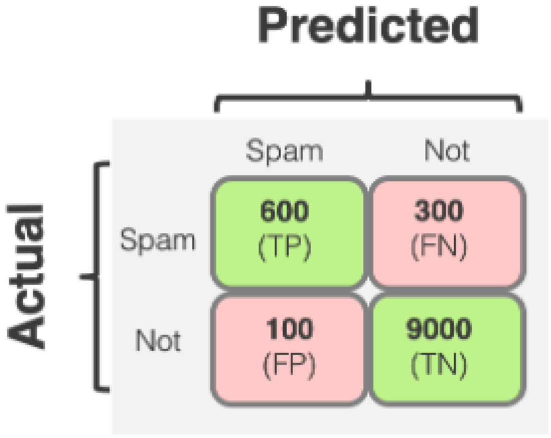
		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$	F_1 score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$		

Effect of change the threshold ->errors/ confusion matrices



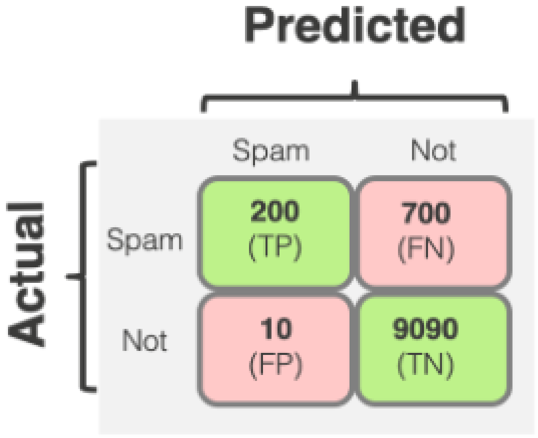
		Predicted	
		Spam	Not
Actual	Spam	800 (TP)	100 (FN)
	Not	500 (FP)	8600 (TN)

Threshold: 0.5



		Predicted	
		Spam	Not
Actual	Spam	600 (TP)	300 (FN)
	Not	100 (FP)	9000 (TN)

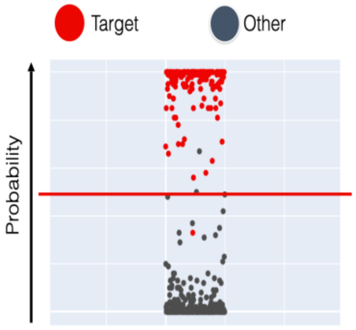
Threshold: 0.8



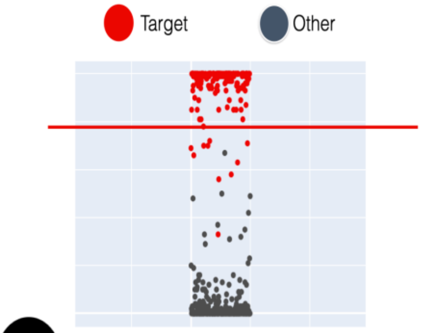
		Predicted	
		Spam	Not
Actual	Spam	200 (TP)	700 (FN)
	Not	10 (FP)	9090 (TN)

Threshold: 0.95

Probability > 50%



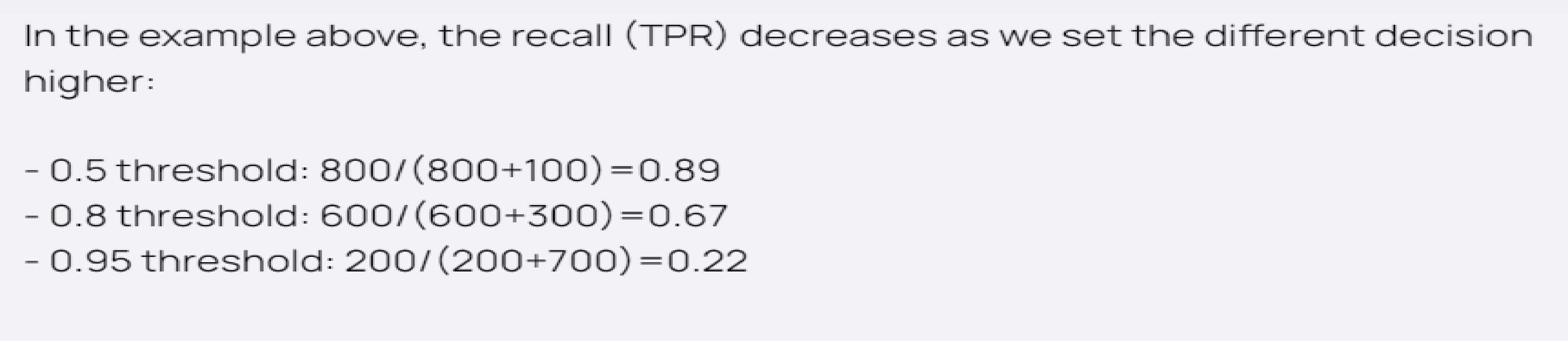
Probability > 80%



You can vary the decision threshold that defines how to convert the model predictions into labels. This, in turn, can change the number of errors the model makes.

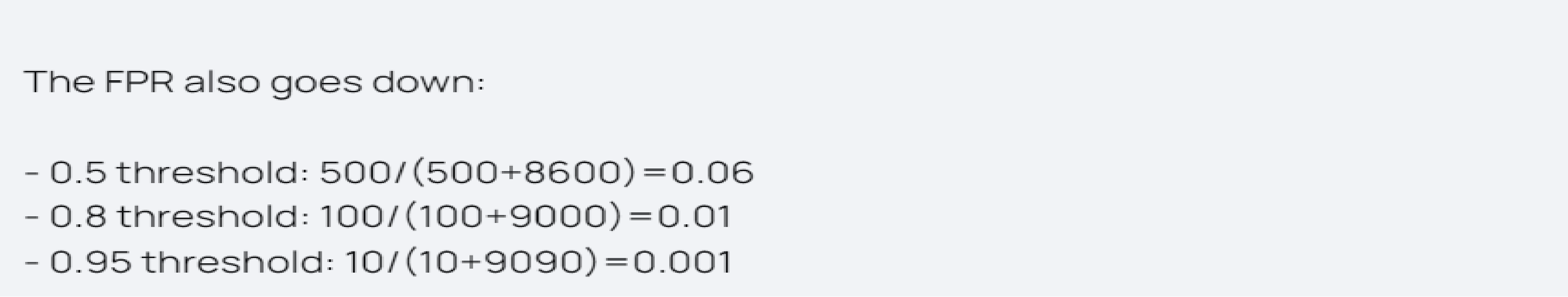
Calculate TPR and FPR for above case.

Effect of change the threshold ->errors



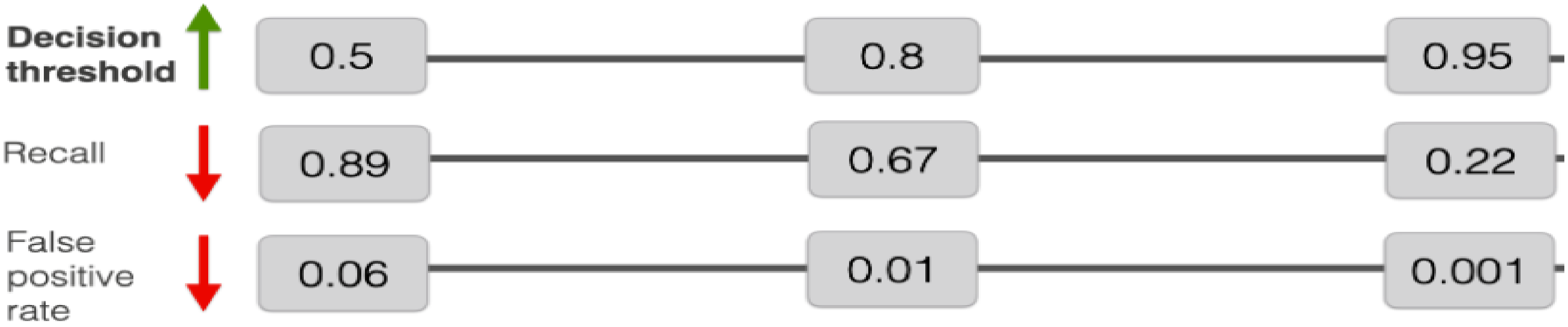
In the example above, the recall (TPR) decreases as we set the different decision higher:

- 0.5 threshold: 800/(800+100)=0.89
- 0.8 threshold: 600/(600+300)=0.67
- 0.95 threshold: 200/(200+700)=0.22



The FPR also goes down:

- 0.5 threshold: 500/(500+8600)=0.06
- 0.8 threshold: 100/(100+9000)=0.01
- 0.95 threshold: 10/(10+9090)=0.001

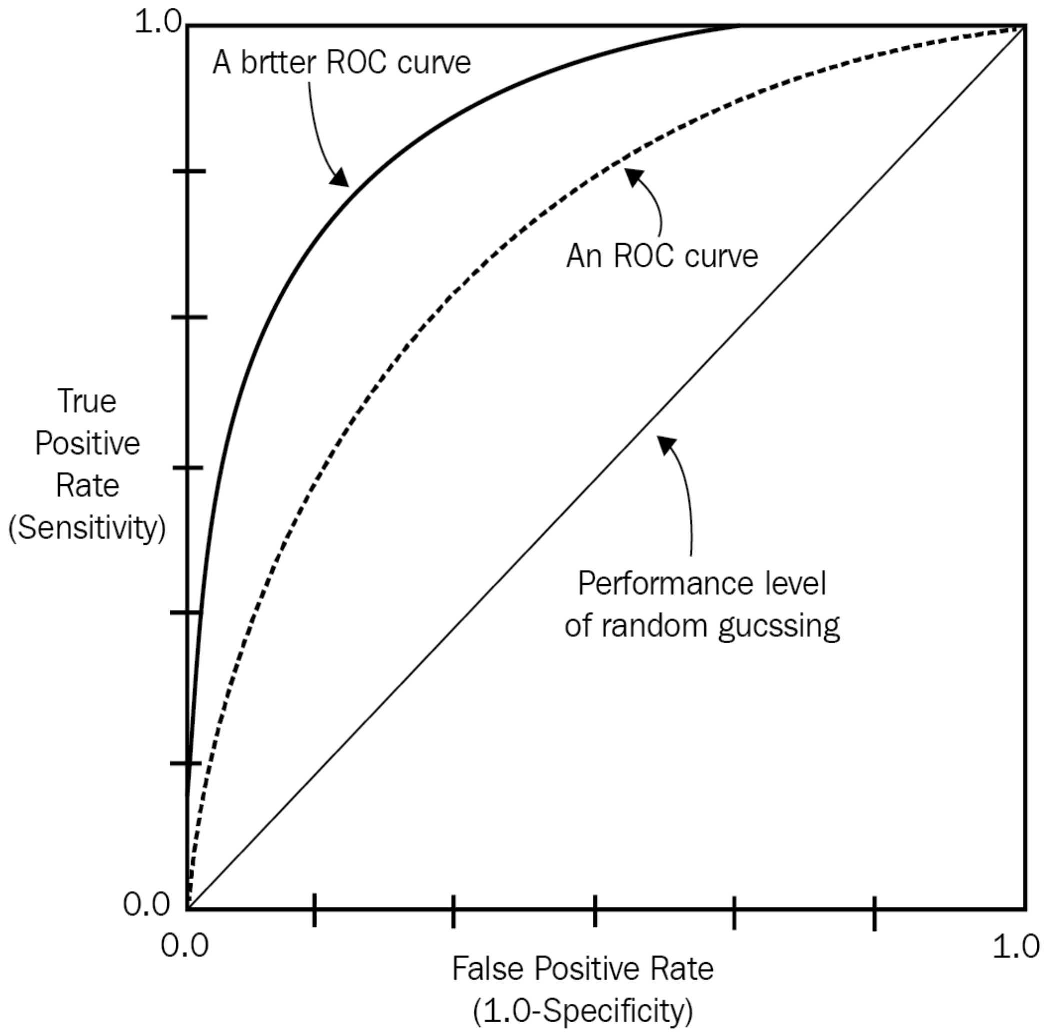


Decision threshold	↑	0.5	0.8	0.95
Recall	↓	0.89	0.67	0.22
False positive rate	↓	0.06	0.01	0.001

Effect of change the threshold

- When you set the threshold higher, you make the model "more conservative." It assigns the True label when it is "more confident." But as a consequence, you typically lower recall/TPR: you detect fewer examples of the target class overall.
- When you set the threshold lower, you make the model "less strict." It assigns the True label more often, even when "less confident." Consequently, you increase recall/TPR: you will detect more examples of the target class. However, this may also lead to lower precision, as the model may make more False Positive predictions.
- TPR and FPR change in the same direction. The higher the recall (TPR), the higher the rate of false positive errors (FPR).
- The lower the recall, the fewer false alarms the model gives.

ROC-Receiver Operating Characteristic Curve

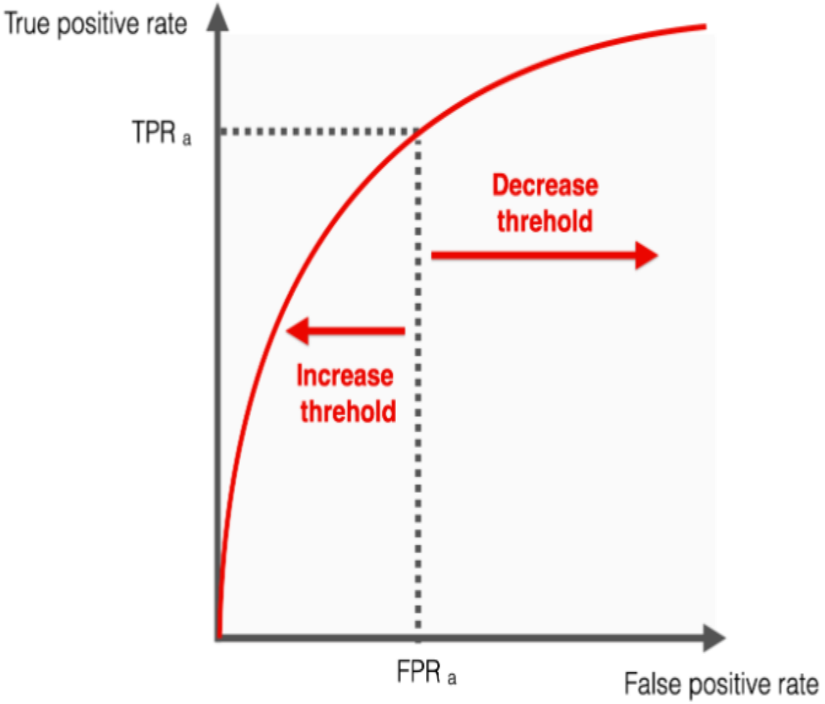


ROC plots classifier performance over the entire range of possible decision thresholds.
Each point on the curve represents a specific decision threshold with a corresponding True Positive rate and False Positive rate.

A discrete classifier returns only the predicted class and gives a single point on the ROC space. But for probabilistic classifiers, which give a probability or score that reflects the degree to which an instance belongs to one class rather than another, we can create a curve by changing the threshold for the score.

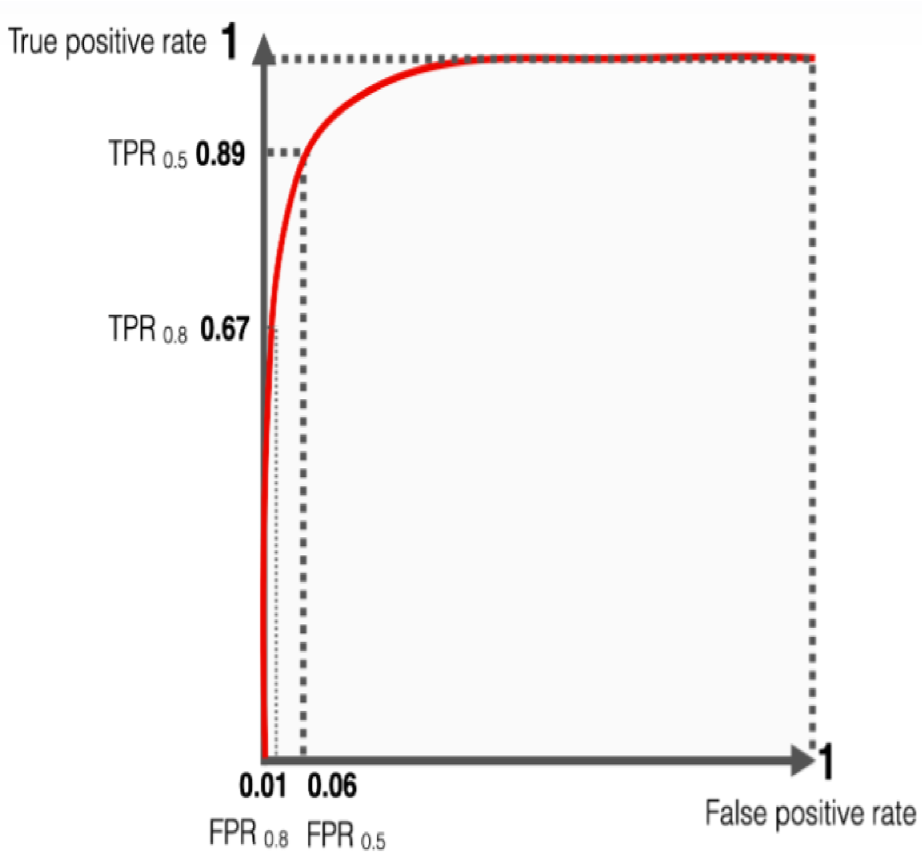
ROC-Receiver Operating Characteristic Curve

- The left side of the curve corresponds to the more "confident" thresholds: a higher threshold leads to lower recall and fewer false positive errors. The extreme point is when both recall and FPR are 0. In this case, there are no correct detections but also no false ones.
- The right side of the curve represents the "less strict" scenarios when the threshold is low. Both recall and False Positive rates are higher, ultimately reaching 100%. If you put the threshold at 0, the model will always predict a positive class: both recall, and the FPR will be 1.



ROC-Receiver Operating Characteristics

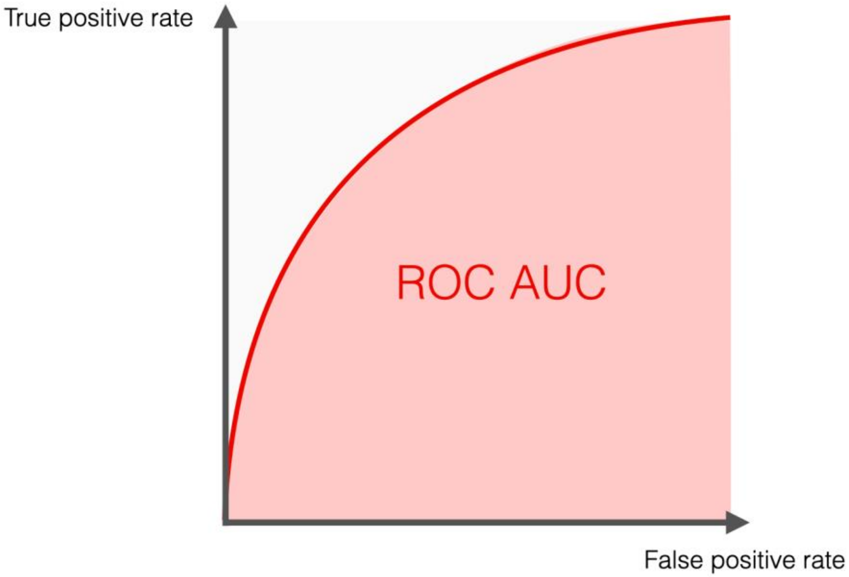
- A plot - FPR on the X axis and TPR on the Y axis.
- examines the tradeoff between the ability of a classifier to correctly identify positive cases and the number of negative cases that are incorrectly classified
- The point (0,1) is the perfect classifier: it classifies all positive cases and negative cases correctly.
- Area under the ROC curve of 1.
- The point (0,0) represents a classifier that predicts all cases to be negative, while the point (1,1) corresponds to a classifier that predicts every case to be positive.
- Point (1,0) is the classifier that is incorrect for all classifications.



In many cases, a classifier has a parameters that can be adjusted to get (FP, TP) pair and a series of such pairs can be used to plot an ROC curve.

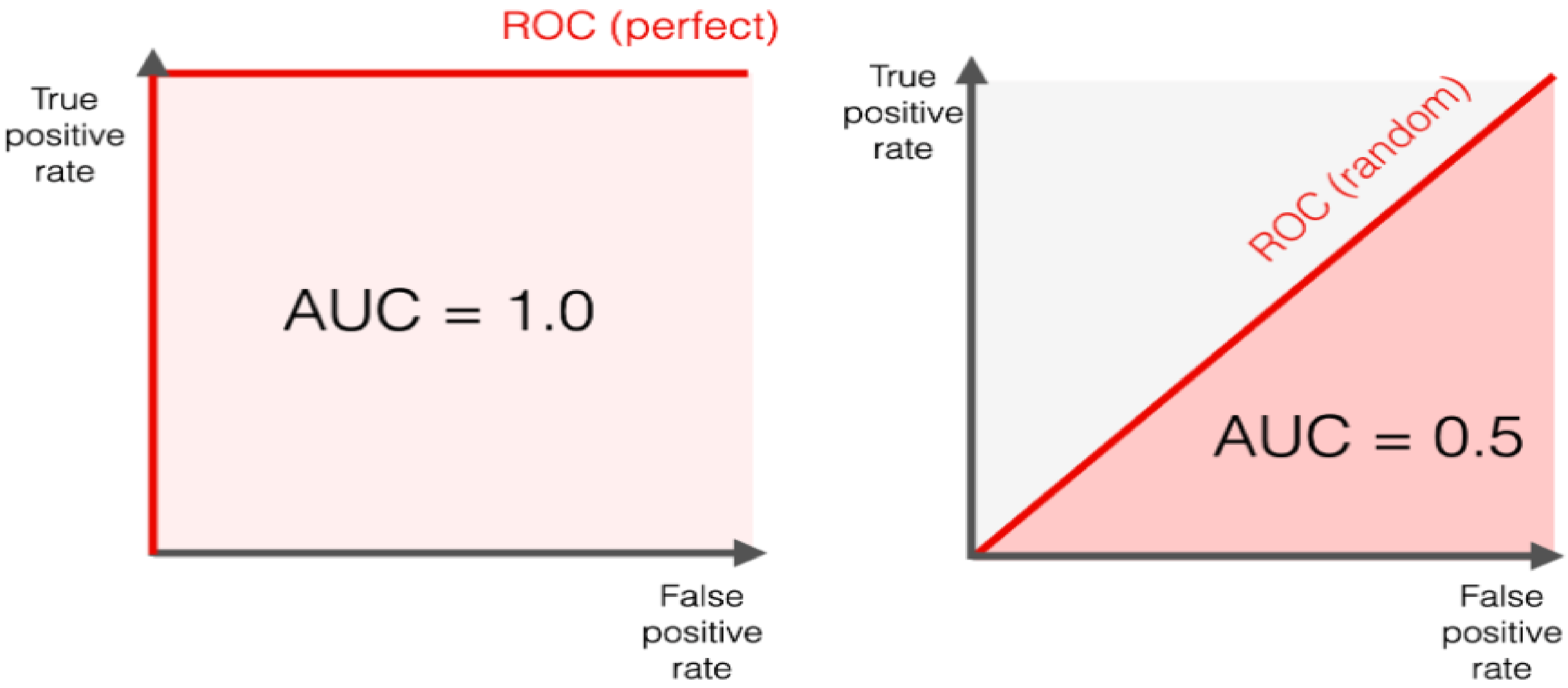
An ROC curve or point is independent of class distribution

Area Under Curve (AUC)



- is one of the most widely used metrics for model evaluation for binary classification problems.
- measures the entire two-dimensional area present underneath the entire ROC curve.
- The Area Under the Curve provides the ability for a classifier to distinguish between classes.
- ROC AUC score is a single number that summarizes the classifier's performance across all possible classification thresholds
- The higher the AUC - better the performance of the model
- AUC near to the 1 -> good separability.
- AUC near 0 -> poor model -> worst separability i.e. predicting 0s as 1s and 1s as 0s.
- If AUC is 0.5 -> model has no class separation capacity.

Area Under Curve (AUC)



ROC AUC reflects the model quality in one number. It is convenient to use a single metric, especially when comparing multiple models.
ROC AUC is less useful when you care about **different costs of error** and want to find the optimal threshold to optimize for the cost of a specific error.

Thank You

- **???**