Big Data Computing

Master's Degree in Computer Science

2019-2020

Gabriele Tolomei

Department of Computer Science

Sapienza Università di Roma

tolomei@di.uniroma1.it

Sapienza Università di Roma

Recap from Last Lectures

- We discussed 2 main methods to approach classification tasks:
  - Logistic Regression
  - Decision Trees (and ensemble of those)

Recap from Last Lectures

- We discussed 2 main methods to approach classification tasks:
  - Logistic Regression
  - Decision Trees (and ensemble of those)
- No clear "winner" between the two
  - It depends on the actual task we are trying to achieve
  - Typically, we need to compare several approaches against each other

Recap from Last Lectures

- We discussed 2 main methods to approach classification tasks:
  - Logistic Regression
  - Decision Trees (and ensemble of those)
- No clear "winner" between the two
  - It depends on the actual task we are trying to achieve
  - Typically, we need to compare several approaches against each other

We need a robust evaluation framework to assess models performance

Evaluation Metrics for Machine Learning

- Tied and specific to a machine learning task

Evaluation Metrics for Machine Learning

- Tied and specific to a machine learning task
- There are different metrics for clustering, regression, classification, etc.

Evaluation Metrics for Machine Learning

- Tied and specific to a machine learning task
- There are different metrics for clustering, regression, classification, etc.
- Some metrics, such as precision-recall, may be useful for multiple tasks

Evaluation Metrics for Machine Learning

- Tied and specific to a machine learning task
- There are different metrics for clustering, regression, classification, etc.
- Some metrics, such as precision-recall, may be useful for multiple tasks
- We have already talked about quality metrics for clustering and regression

Evaluation Metrics for Machine Learning

- Tied and specific to a machine learning task
- There are different metrics for clustering, regression, classification, etc.
- Some metrics, such as precision-recall, may be useful for multiple tasks
- We have already talked about quality metrics for clustering and regression
- We now discuss performance metrics for **classification**

Confusion Matrix

- It is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model

Confusion Matrix

- It is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model
- It is used when the output can be of two or more types of classes

Confusion Matrix

- It is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model
- It is used when the output can be of two or more types of classes
- It is not a performance measure as such, but almost all of the performance metrics are based on it

Confusion Matrix

- It is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model
- It is used when the output can be of two or more types of classes
- It is not a performance measure as such, but almost all of the performance metrics are based on it
- **Example:** binary classification task to predict whether a patient has cancer (class label=1) or not (class label=0)

Confusion Matrix: Example

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)		
	NEGATIVES (0)		

Confusion Matrix: True Positives (TP)

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	
	NEGATIVES (0)		

True Positives (TP)

The actual class of the data point is 1 (True) and the predicted is also 1 (True)

A patient actually has cancer (1) and the model predicts he has cancer (1)

Confusion Matrix: True Negatives (TN)

True Negatives (TN)

The actual class of the data point is 0 (False) and the predicted is also 0 (False)

A patient has NO cancer (0) and the model predicts he has NO cancer (0)

Confusion Matrix: False Positives (FP)

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)		
	NEGATIVES (0)	FALSE POSITIVES **(FP)**	

False Positives (FP)

The actual class of the data point is 0 (False) and the predicted is 1 (True)

A patient has NO cancer (0) and yet the model diagnosed him with cancer (1)

Confusion Matrix: False Negatives (FN)

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)		FALSE NEGATIVES **(FN)**
	NEGATIVES (0)		

False Negatives (FN)

The actual class of the data point is 1 (True) and the predicted is 0 (False)

A patient actually has cancer (1) and yet the model predicts he has NO cancer (0)

Confusion Matrix: Example

		PREDICTED	
		POSITIVES (1)	**NEGATIVES (0)**
ACTUAL	**POSITIVES (1)**	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

What Error Should We Minimize?

- Both FP and FN are considered as **errors** as they indicate the model classifies things incorrectly as compared to the actual classes

What Error Should We Minimize?

- Both FP and FN are considered as **errors** as they indicate the model classifies things incorrectly as compared to the actual classes
- Ideally, we would like the model to have both FP and FN equal to 0

What Error Should We Minimize?

- Both FP and FN are considered as **errors** as they indicate the model classifies things incorrectly as compared to the actual classes
- Ideally, we would like the model to have both FP and FN equal to 0
- In practice, this is not the case as any model will not be 100% accurate

What Error Should We Minimize?

- Both FP and FN are considered as **errors** as they indicate the model classifies things incorrectly as compared to the actual classes
- Ideally, we would like the model to have both FP and FN equal to 0
- In practice, this is not the case as any model will not be 100% accurate
- There is no strict rule on what should be minimized in all the situations as this depends on the business needs and the context of the problem

What Error Should We Minimize?

- Both FP and FN are considered as **errors** as they indicate the model classifies things incorrectly as compared to the actual classes
- Ideally, we would like the model to have both FP and FN equal to 0
- In practice, this is not the case as any model will not be 100% accurate
- There is no strict rule on what should be minimized in all the situations as this depends on the business needs and the context of the problem
- We might want to minimize either FP or FN

Minimize False Negatives

- Suppose in our cancer detection task only 5 out of 100 people have cancer

Minimize False Negatives

- Suppose in our cancer detection task only 5 out of 100 people have cancer
- We may want to correctly classify all the cancerous patients (minority class)

Minimize False Negatives

- Suppose in our cancer detection task only 5 out of 100 people have cancer
- We may want to correctly classify all the cancerous patients (minority class)
- Even a very bad model which always predicts every patient as non-cancerous will get to a 95% accuracy (more on this later…)

Minimize False Negatives

- Suppose in our cancer detection task only 5 out of 100 people have cancer
- We may want to correctly classify all the cancerous patients (minority class)
- Even a very bad model which always predicts every patient as non-cancerous will get to a 95% accuracy (more on this later…)
- In order to capture **all** cancer cases, we might end up classifying as cancerous patients who are actually **not** having cancer

Minimize False Negatives

- Suppose in our cancer detection task only 5 out of 100 people have cancer
- We may want to correctly classify all the cancerous patients (minority class)
- Even a very bad model which always predicts every patient as non-cancerous will get to a 95% accuracy (more on this later…)
- In order to capture **all** cancer cases, we might end up classifying as cancerous patients who are actually **not** having cancer

Misclassifying an actual cancerous patient will be a much more severe and harmful mistake than the other way around!

Minimize False Positives

- Let's use a different example: a model that classifies whether an email is spam (1) or non-spam (0)

Minimize False Positives

- Let's use a different example: a model that classifies whether an email is spam (1) or non-spam (0)
- Suppose the model classifies an important non-spam email as spam

Minimize False Positives

- Let's use a different example: a model that classifies whether an email is spam (1) or non-spam (0)
- Suppose the model classifies an important non-spam email as spam
- This is pretty worst than classifying a spam email as non-spam since in that case, we can still go ahead and manually delete it

Minimize False Positives

- Let's use a different example: a model that classifies whether an email is spam (1) or non-spam (0)
- Suppose the model classifies an important non-spam email as spam
- This is pretty worst than classifying a spam email as non-spam since in that case, we can still go ahead and manually delete it
- So, in this case minimizing FP is more important than minimizing FN

Accuracy

The number of **correct** predictions **over all** the predictions made

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

Accuracy

The number of **correct** predictions **over all** the predictions made

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

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

When to Use Accuracy?

- Accuracy is a good measure when the target variable classes in the data are **nearly balanced**

When to Use Accuracy?

- Accuracy is a good measure when the target variable classes in the data are **nearly balanced**
- **Example:**
  - An image classifier trained over a quite balanced dataset of 60% pictures of dogs and 40% pictures of cats

When to Use Accuracy?

- Accuracy is a good measure when the target variable classes in the data are **nearly balanced**
- **Example:**
  - An image classifier trained over a quite balanced dataset of 60% pictures of dogs and 40% pictures of cats
  - 97% accuracy would indicate a very good performing model

When **not** to Use Accuracy?

- Accuracy should **never** be used as a measure when the distribution of target variable classes is **skewed**

When **not** to Use Accuracy?

- Accuracy should **never** be used as a measure when the distribution of target variable classes is **skewed**
- **Example:**
  - In our cancer detection setting, only 5 out of 100 people have cancer

When **not** to Use Accuracy?

- Accuracy should **never** be used as a measure when the distribution of target variable classes is **skewed**
- **Example:**
  - In our cancer detection setting, only 5 out of 100 people have cancer
  - A trivial model which always predict the majority class (i.e., no cancer) will still classify 95 patients correctly

When **not** to Use Accuracy?

- Accuracy should **never** be used as a measure when the distribution of target variable classes is **skewed**
- **Example:**
  - In our cancer detection setting, only 5 out of 100 people have cancer
  - A trivial model which always predict the majority class (i.e., no cancer) will still classify 95 patients correctly
  - Even though the model is terrible at predicting cancer, its accuracy is 95%

Precision or Positive Predicted Value (PPV)

The number of correctly predicted positive instances over all the positive predictions made

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

Precision or Positive Predicted Value (PPV)

The number of correctly predicted positive instances over all the positive predictions made

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

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

Precision or Positive Predicted Value (PPV)

The number of correctly predicted positive instances over all the positive predictions made

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

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

Total positive predictions

Precision: Example

- Indicates the proportion of patients that the model diagnosed with cancer and actually have it

Precision: Example

- Indicates the proportion of patients that the model diagnosed with cancer and actually have it
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)

Precision: Example

- Indicates the proportion of patients that the model diagnosed with cancer and actually have it
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- However, the same bad classifier will have a very **poor precision**

Precision: Example

- Indicates the proportion of patients that the model diagnosed with cancer and actually have it
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- However, the same bad classifier will have a **very low precision**
  - TP + FP = 100 (as the model only predicts the positive class label)

Precision: Example

- Indicates the proportion of patients that the model diagnosed with cancer and actually have it
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- However, the same bad classifier will have a **very low precision**
  - TP + FP = 100 (as the model only predicts the positive class label)
  - TP = 5 → Precision = 5/100 = **5%**

Recall or Sensitivity or True Positive Rate (TPR)

The number of correctly predicted positive instances over all the actually positive existing instaces

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

Recall or Sensitivity or True Positive Rate (TPR)

The number of correctly predicted positive instances over all the actually positive existing instaces

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (TP)	FALSE NEGATIVES (FN)
	NEGATIVES (0)	FALSE POSITIVES (FP)	TRUE NEGATIVES (TN)

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

Recall or Sensitivity or True Positive Rate (TPR)

The number of correctly predicted positive instances over all the actually positive existing instaces

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (TP)	FALSE NEGATIVES (FN)
	NEGATIVES (0)	FALSE POSITIVES (FP)	TRUE NEGATIVES (TN)

$$\text{Recall} = \frac{TP}{\boxed{TP + FN \quad \text{Total actually positives}}}$$

Recall: Example

- Indicates the proportion of patients that actually have cancer that are correctly diagnosed by the model as cancerous

Recall: Example

- Indicates the proportion of patients that actually have cancer that are correctly diagnosed by the model as cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)

Recall: Example

- Indicates the proportion of patients that actually have cancer that are correctly diagnosed by the model as cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- This classifier will trivially have a **very high recall**

Recall: Example

- Indicates the proportion of patients that actually have cancer that are correctly diagnosed by the model as cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- This classifier will trivially have a **very high recall**
  - TP + FN = 5 + 0 = 5 (as the model only predicts the positive class label)

Recall: Example

- Indicates the proportion of patients that actually have cancer that are correctly diagnosed by the model as cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- This classifier will trivially have a **very high recall**
  - TP + FN = 5 + 0 = 5 (as the model only predicts the positive class label)
  - TP = 5 → Recall = 5/5 = **100%**

When to Use Precision and When Recall?

Precision gives us information about model performance with respect to FP (how many did we caught)

When to Use Precision and When Recall?

Precision gives us information about model performance with respect to FP (how many did we caught)

Recall gives us information about model performance with respect to FN (how many did we miss)

When to Use Precision and When Recall?

Precision gives us information about model performance with respect to FP (how many did we caught)

Recall gives us information about model performance with respect to FN (how many did we miss)

If the goal is to **minimize FN**, we want to **maximize Recall** (close to 1) without hurting Precision too much

When to Use Precision and When Recall?

Precision gives us information about model performance with respect to FP (how many did we caught)

Recall gives us information about model performance with respect to FN (how many did we miss)

If the goal is to **minimize FN**, we want to **maximize Recall** (close to 1) without hurting Precision too much

If the goal is to **minimize FP**, we want to **maximize Precision** (close to 1) without hurting Recall too much

Precision vs. Recall Trade-Off

- To fully evaluate the quality of a model we need to have a look at **both** Precision and Recall

Precision vs. Recall Trade-Off

- To fully evaluate the quality of a model we need to have a look at **both** Precision and Recall
- One way of doing this is through **Precision-Recall curve**
  - Plot of Recall (x) vs. Precision (y)

Precision vs. Recall Trade-Off

- To fully evaluate the quality of a model we need to have a look at **both** Precision and Recall
- One way of doing this is through **Precision-Recall curve**
  - Plot of Recall (x) vs. Precision (y)
- Compute precision-recall pairs for different probability thresholds
  - Figure out the desired trade-off threshold from the plot

Specificity or True Negative Rate (TNR)

The number of correctly predicted negative instances over all the actually negative existing instaces

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

Specificity or True Negative Rate (TNR)

The number of correctly predicted negative instances over all the actually negative existing instaces

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (**TP**)	FALSE NEGATIVES (**FN**)
	NEGATIVES (0)	FALSE POSITIVES (**FP**)	TRUE NEGATIVES (**TN**)

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

Specificity or True Negative Rate (TNR)

The number of correctly predicted negative instances over all the actually negative existing instaces

		PREDICTED	
		POSITIVES (1)	NEGATIVES (0)
ACTUAL	POSITIVES (1)	TRUE POSITIVES (TP)	FALSE NEGATIVES (FN)
	NEGATIVES (0)	FALSE POSITIVES (FP)	TRUE NEGATIVES (TN)

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

Total actually negatives

Specificity: Example

- Indicates the proportion of patients that actually **don't** have cancer that are correctly diagnosed by the model as non-cancerous

Specificity: Example

- Indicates the proportion of patients that actually **don't** have cancer that are correctly diagnosed by the model as non-cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)

Specificity: Example

- Indicates the proportion of patients that actually **don't** have cancer that are correctly diagnosed by the model as non-cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- This classifier will trivially have a **very low specificity**

Specificity: Example

- Indicates the proportion of patients that actually **don't** have cancer that are correctly diagnosed by the model as non-cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- This classifier will trivially have a **very low specificity**
  - TN + FP = 0 + 95 = 95 (as the model only predicts the positive class label)

Specificity: Example

- Indicates the proportion of patients that actually **don't** have cancer that are correctly diagnosed by the model as non-cancerous
- A bad classifier which always predicts the majority class gets 95% accuracy (assuming the same 95÷5 ratio of patients w/o and w cancer)
- This classifier will trivially have a **very low specificity**
  - TN + FP = 0 + 95 = 95 (as the model only predicts the positive class label)
  - TN = 0 → Specificity = 0/95 = 0%

Combining Precision and Recall

- We would love to have an aggregated score which combines both Precision (P) and Recall (R)

Combining Precision and Recall

- We would love to have an aggregated score which combines both Precision (P) and Recall (R)
- One way to do it could be simply taking the **mean**: (P + R)/2

Combining Precision and Recall

- We would love to have an aggregated score which combines both Precision (P) and Recall (R)
- One way to do it could be simply taking the **mean**: (P + R)/2
- However, this might be bad in some extreme situations

Combining Precision and Recall

- We would love to have an aggregated score which combines both Precision (P) and Recall (R)
- One way to do it could be simply taking the **mean**: (P + R)/2
- However, this might be bad in some extreme situations
- **Example:** 100 credit card transactions of which 97 are legitimate and 3 fraudulent, and a classifier predicting everything as fraudulent

Combining Precision and Recall

		PREDICTED	
		FRAUD (1)	LEGIT (0)
ACTUAL	FRAUD (1)	3	0
	LEGIT (0)	97	0

Combining Precision and Recall

		PREDICTED	
		FRAUD (1)	LEGIT (0)
ACTUAL	FRAUD (1)	3	0
	LEGIT (0)	97	0

$$\text{Precision} = \frac{3}{100} = 3\%$$

Combining Precision and Recall

		PREDICTED	
		FRAUD (1)	LEGIT (0)
ACTUAL	FRAUD (1)	$\boxed{3}$	0
	LEGIT (0)	97	0

$$\text{Precision} = \frac{3}{100} = 3\%$$

$$\text{Recall} = \frac{3}{3} = 100\%$$

Combining Precision and Recall

		PREDICTED	
		FRAUD (1)	LEGIT (0)
ACTUAL	FRAUD (1)	3	0
	LEGIT (0)	97	0

$$\text{Precision} = \frac{3}{100} = 3\%$$

$$\text{Recall} = \frac{3}{3} = 100\%$$

$$\text{Avg} = \frac{\text{Precision} + \text{Recall}}{2} \approx 52\%$$

Combining Precision and Recall

		PREDICTED	
		FRAUD (1)	LEGIT (0)
ACTUAL	FRAUD (1)	3	0
	LEGIT (0)	97	0

$$\text{Precision} = \frac{3}{100} = 3\%$$

$$\text{Recall} = \frac{3}{3} = 100\%$$

$$\text{Avg} = \frac{\text{Precision} + \text{Recall}}{2} \boxed{\approx 52\%}$$

Too "good" for such a bad model!

F1 Score

- We need something more "balanced" than the arithmetic mean, which is known to be sensible to extreme values

F1 Score

- We need something more "balanced" than the arithmetic mean, which is known to be sensible to extreme values
- **Harmonic Mean** between x and y

$$\boxed{\mathrm{H}_{\mathrm{mean}}(x, y) = \frac{2xy}{x + y}}$$

F1 Score

- We need something more "balanced" than the arithmetic mean, which is known to be sensible to extreme values
- **Harmonic Mean** between x and y $$\mathrm{H}_{\text{mean}}(x, y) = \frac{2xy}{x + y}$$
  - Similar to the average when x and y are close to each other
  - Closer to the smaller number as compared to the larger number
  - Mitigate the impact of large outliers and aggravate the impact of small ones

F1 Score

- We need something more "balanced" than the arithmetic mean, which is known to be sensible to extreme values
- **Harmonic Mean** between x and y

$$\mathrm{H}_{\mathrm{mean}}(x, y) = \frac{2xy}{x + y}$$

  - Similar to the average when x and y are close to each other
  - Closer to the smaller number as compared to the larger number
  - Mitigate the impact of large outliers and aggravate the impact of small ones

$$\boxed{\text{F1-score}(P, R) = \frac{2PR}{P + R}}$$

F1 Score

- In the example before:

$$\text{F1-score}(P, R) = \frac{2 * 3 * 100}{103} \approx 5.8\%$$

F1 Score

- In the example before:

$$\text{F1-score}(P, R) = \frac{2 * 3 * 100}{103} \approx 5.8\%$$

- F1 Score is an **effective** evaluation metric in the following scenarios:
  - When both FP and FN errors are equally harmful
  - When TN is high

Matthews Correlation Coefficient (MCC)

- Note that F1 Score does not consider TN at all

Matthews Correlation Coefficient (MCC)

- Note that F1 Score does not consider TN at all
- This may be acceptable whenever TN are sort of innumerable (e.g., in information retrieval)

Matthews Correlation Coefficient (MCC)

- Note that F1 Score does not consider TN at all
- This may be acceptable whenever TN are sort of innumerable (e.g., in information retrieval)
- MCC is much **more informative** than F1 Score because it considers and balances all the 4 categories of the confusion matrix

Matthews Correlation Coefficient (MCC)

- Note that F1 Score does not consider TN at all
- This may be acceptable whenever TN are sort of innumerable (e.g., in information retrieval)
- MCC is much **more informative** than F1 Score because it considers and balances all the 4 categories of the confusion matrix

$$\boxed{\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}}$$

Receiver Operating Characteristic (ROC)

- ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier as its **discrimination threshold** is varied

Receiver Operating Characteristic (ROC)

- ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier as its **discrimination threshold** is varied
- It is created by plotting the **true positive rate** (TPR) against the **false positive rate** (FPR) at various threshold settings

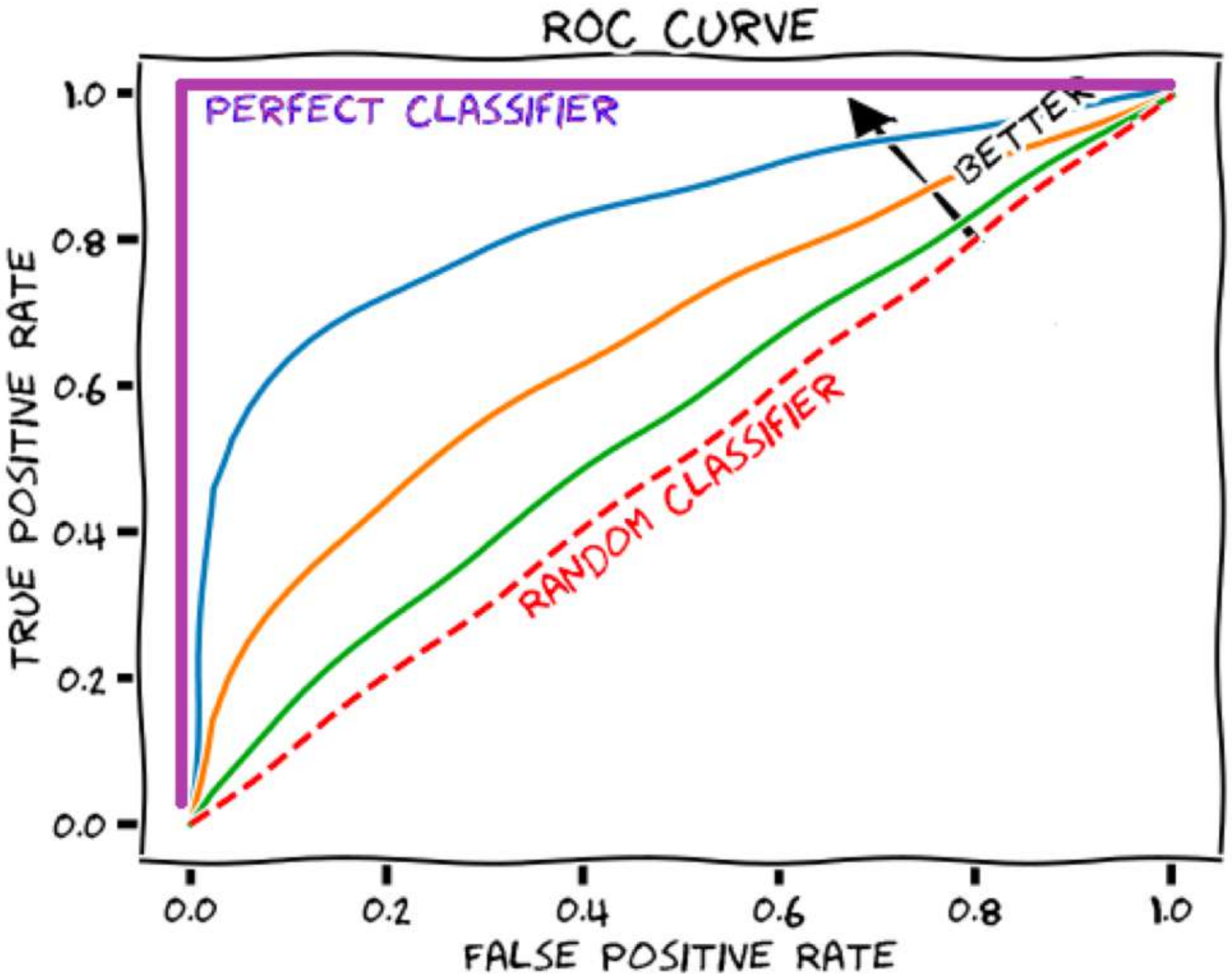
Receiver Operating Characteristic (ROC)

- ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier as its **discrimination threshold** is varied
- It is created by plotting the **true positive rate** (TPR) against the **false positive rate** (FPR) at various threshold settings
- TPR is equivalent to Recall (or Sensitivity)

Receiver Operating Characteristic (ROC)

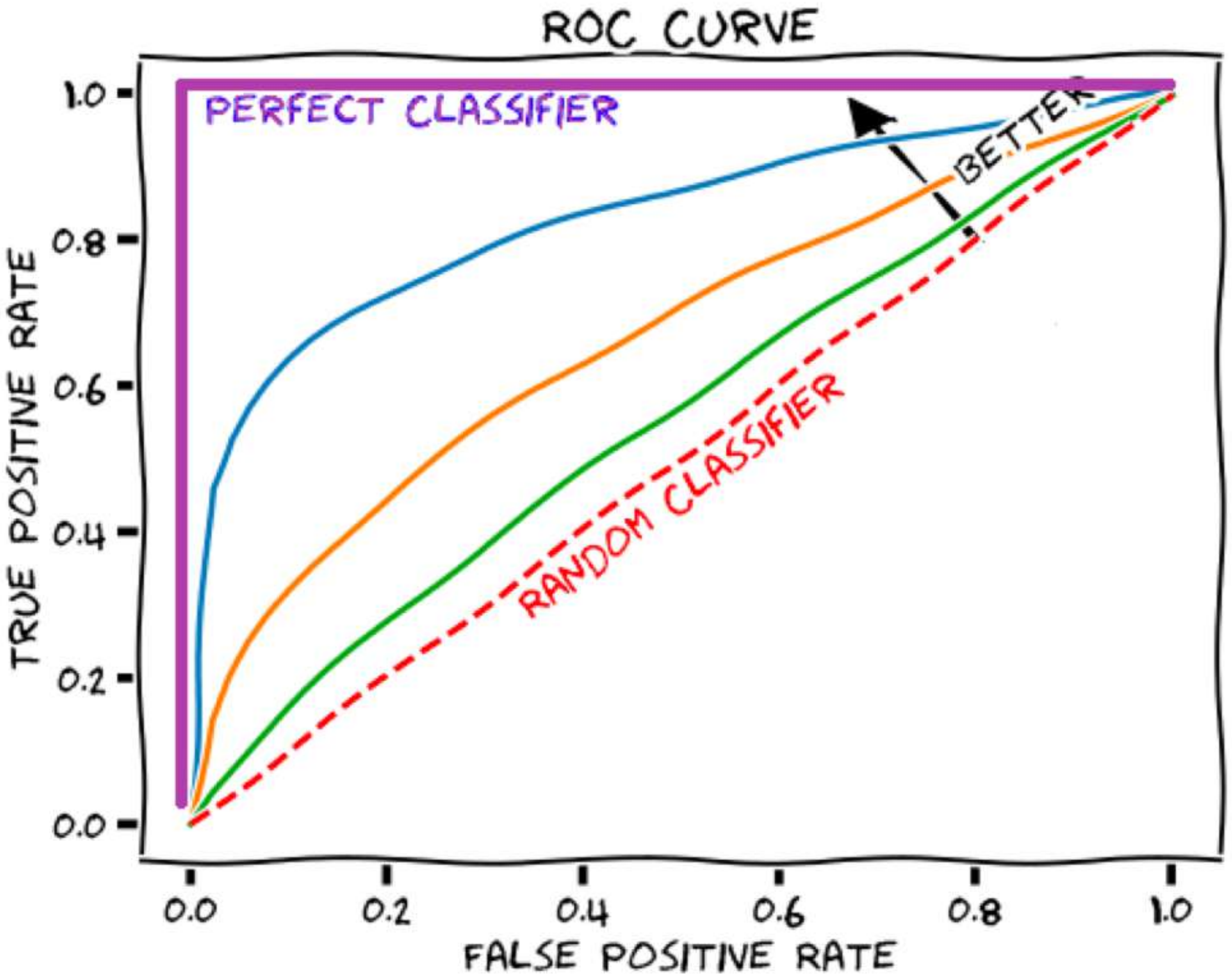
- ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier as its **discrimination threshold** is varied
- It is created by plotting the **true positive rate** (TPR) against the **false positive rate** (FPR) at various threshold settings
- TPR is equivalent to Recall (or Sensitivity)
- FPR is also known as Fall-Out (or 1-Specificity)

ROC Curve: Example



(0, 0) represents a classifier which **never** predicts the **positive** class

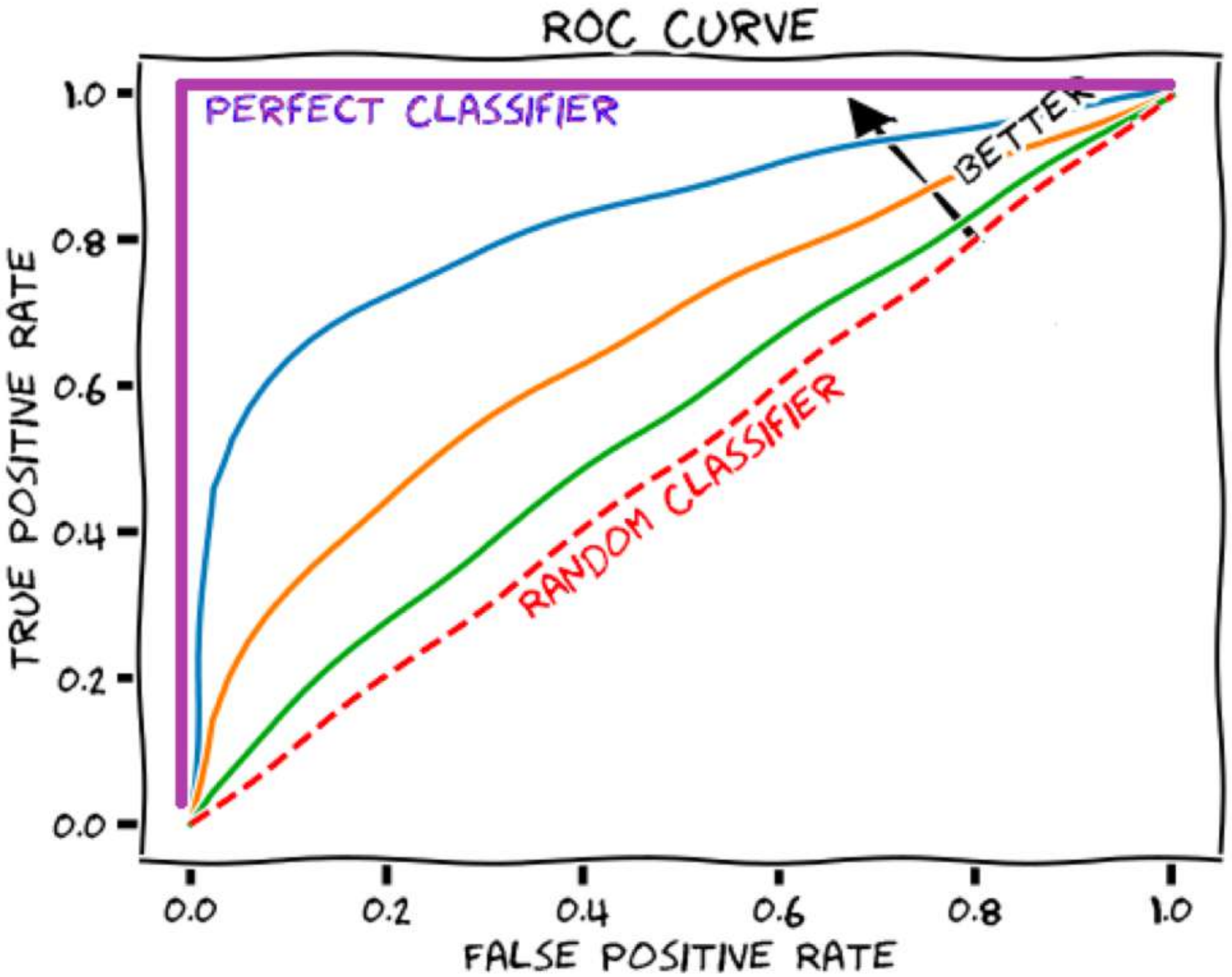
ROC Curve: Example



(0, 0) represents a classifier which **never** predicts the **positive** class

No False Positives at all

ROC Curve: Example

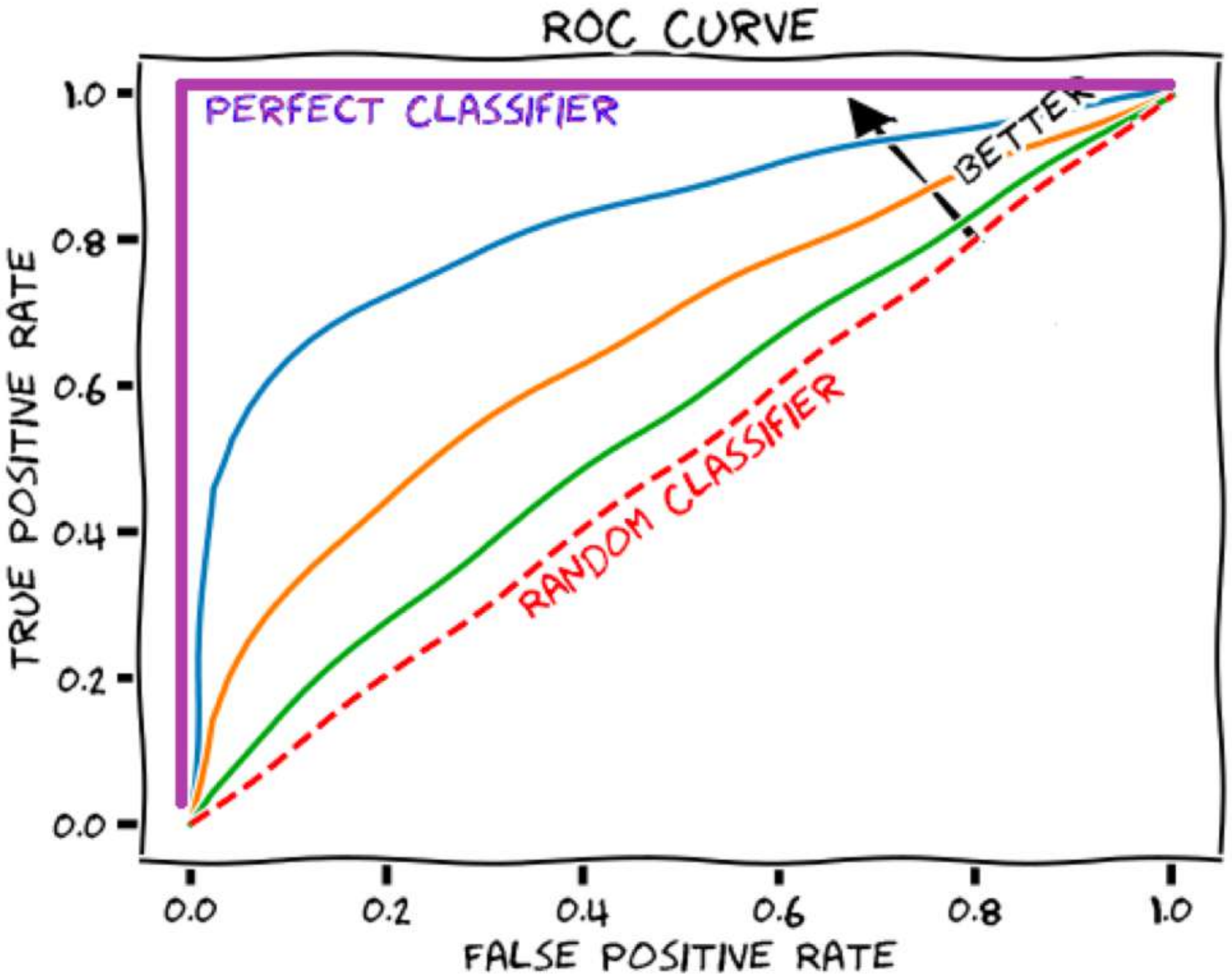


(0, 0) represents a classifier which **never** predicts the **positive** class

No False Positives at all

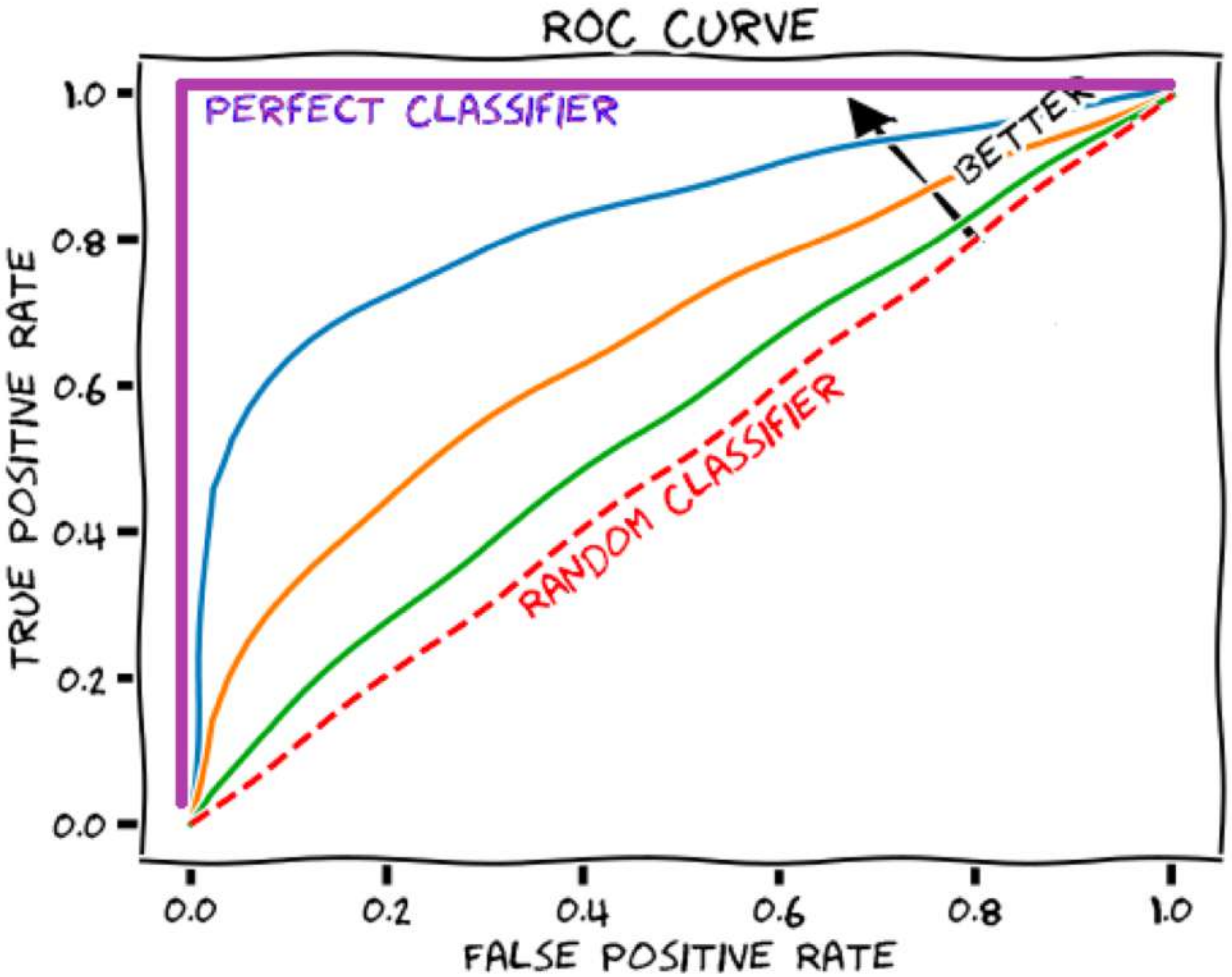
No True Positives either

ROC Curve: Example



(1, 1) represents a classifier which **always** predicts the **positive** class

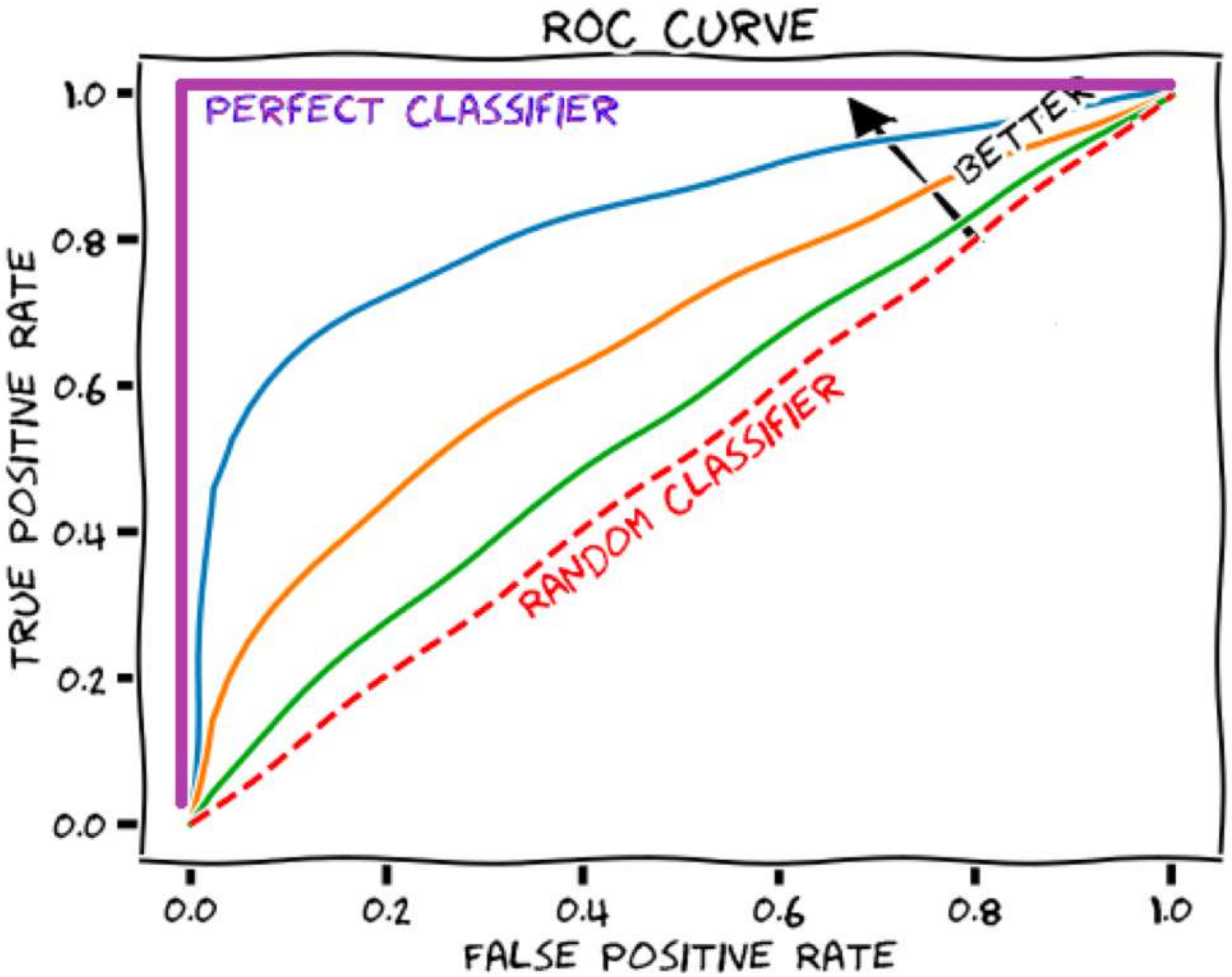
ROC Curve: Example



(1, 1) represents a classifier which **always** predicts the **positive** class

No False Negatives at all

ROC Curve: Example

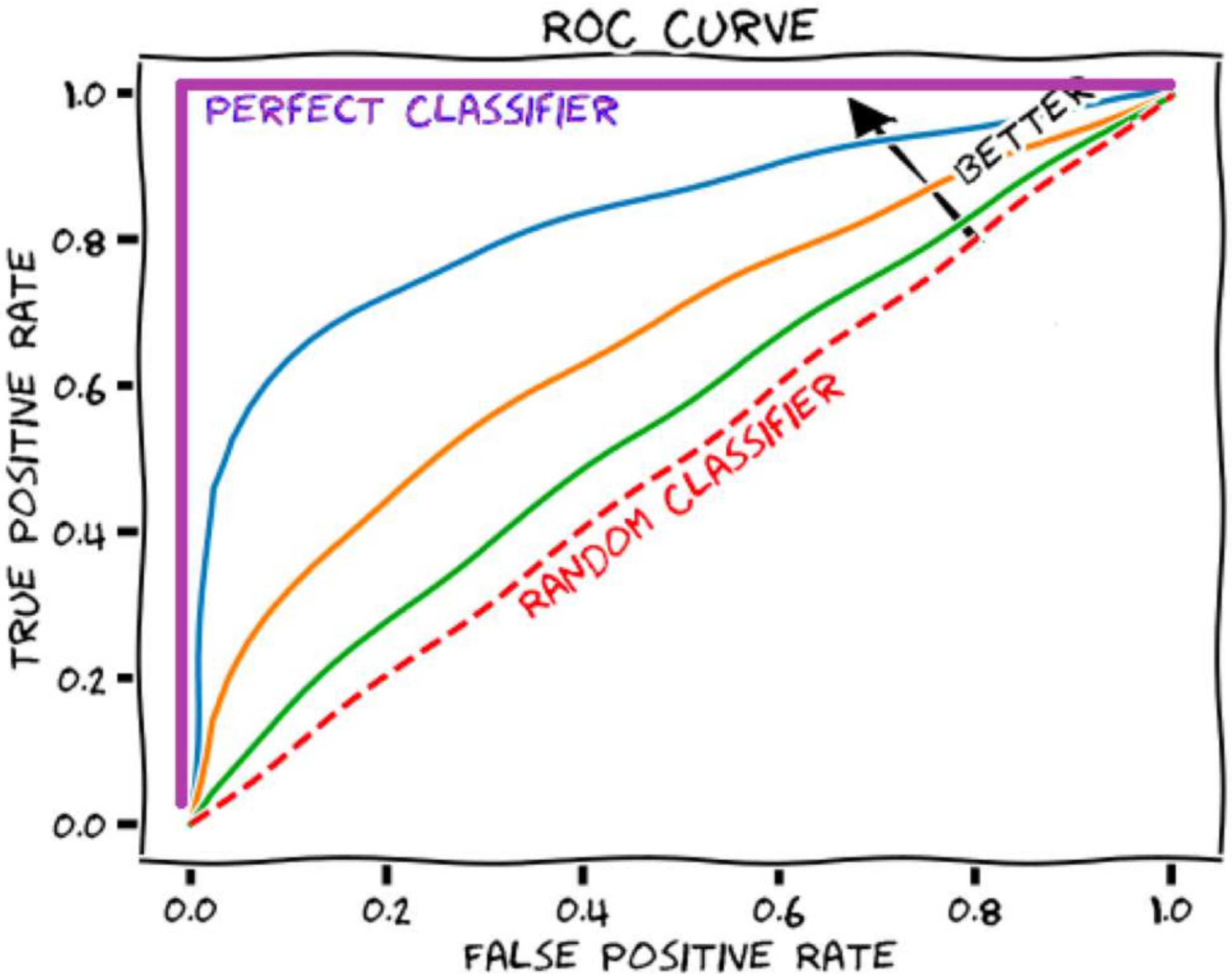


(1, 1) represents a classifier which **always** predicts the **positive** class

No False Negatives at all

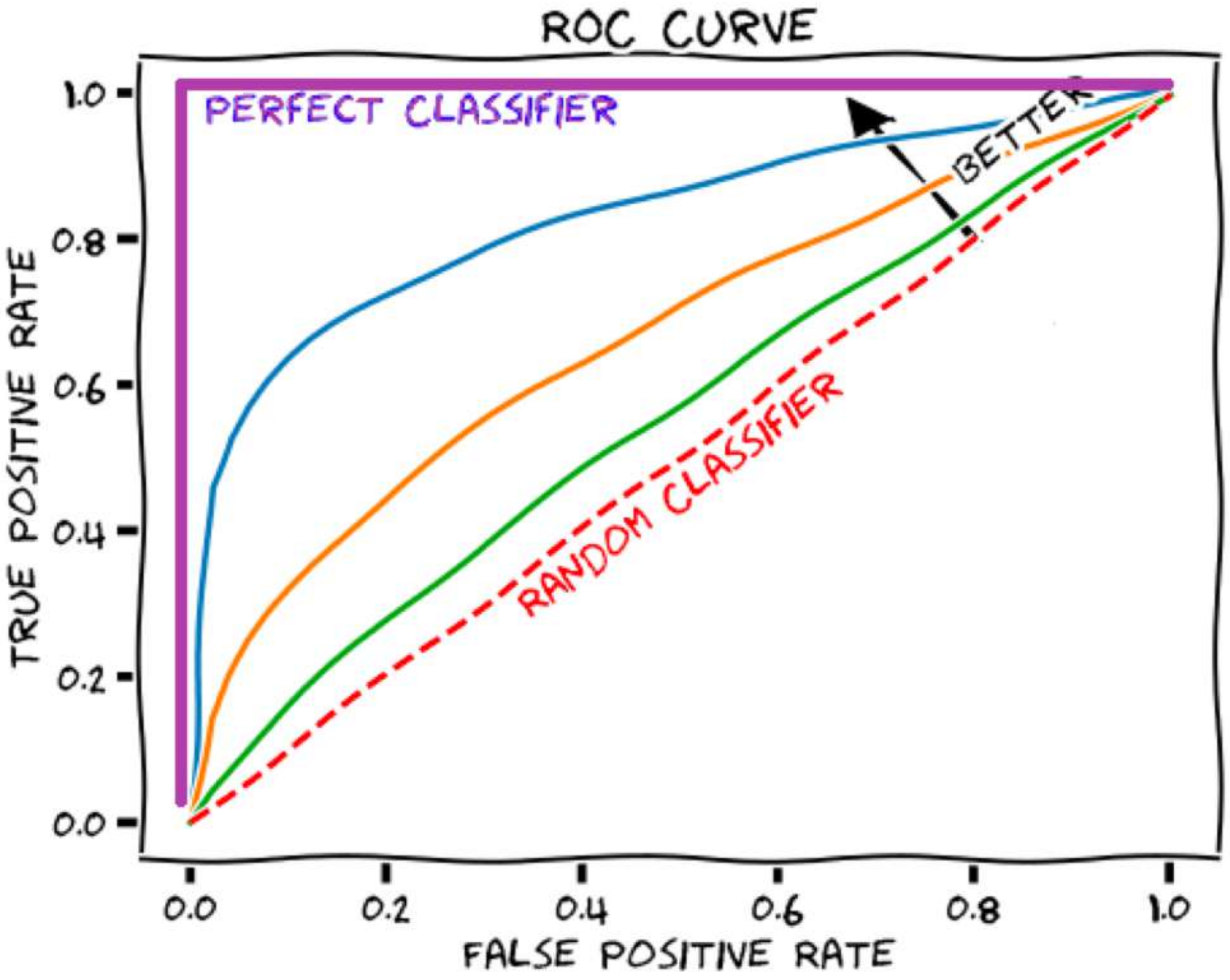
High False Positives

ROC Curve: Example



(0, 1) represents the **best** classifier possible

ROC Curve: Example

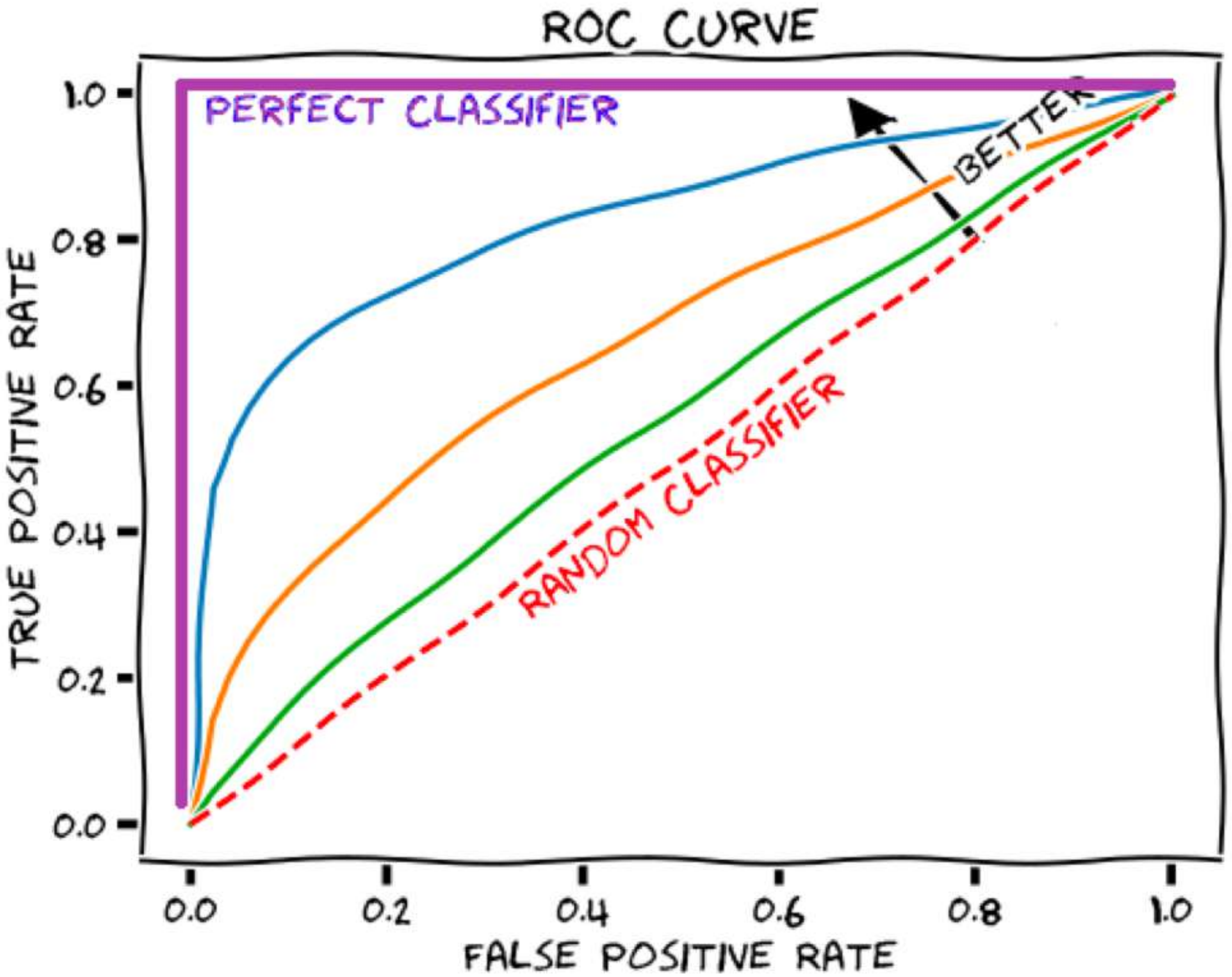


(0, 1) represents the **best** classifier possible

No False Negatives at all

100% Sensitivity

ROC Curve: Example



(0, 1) represents the **best** classifier possible

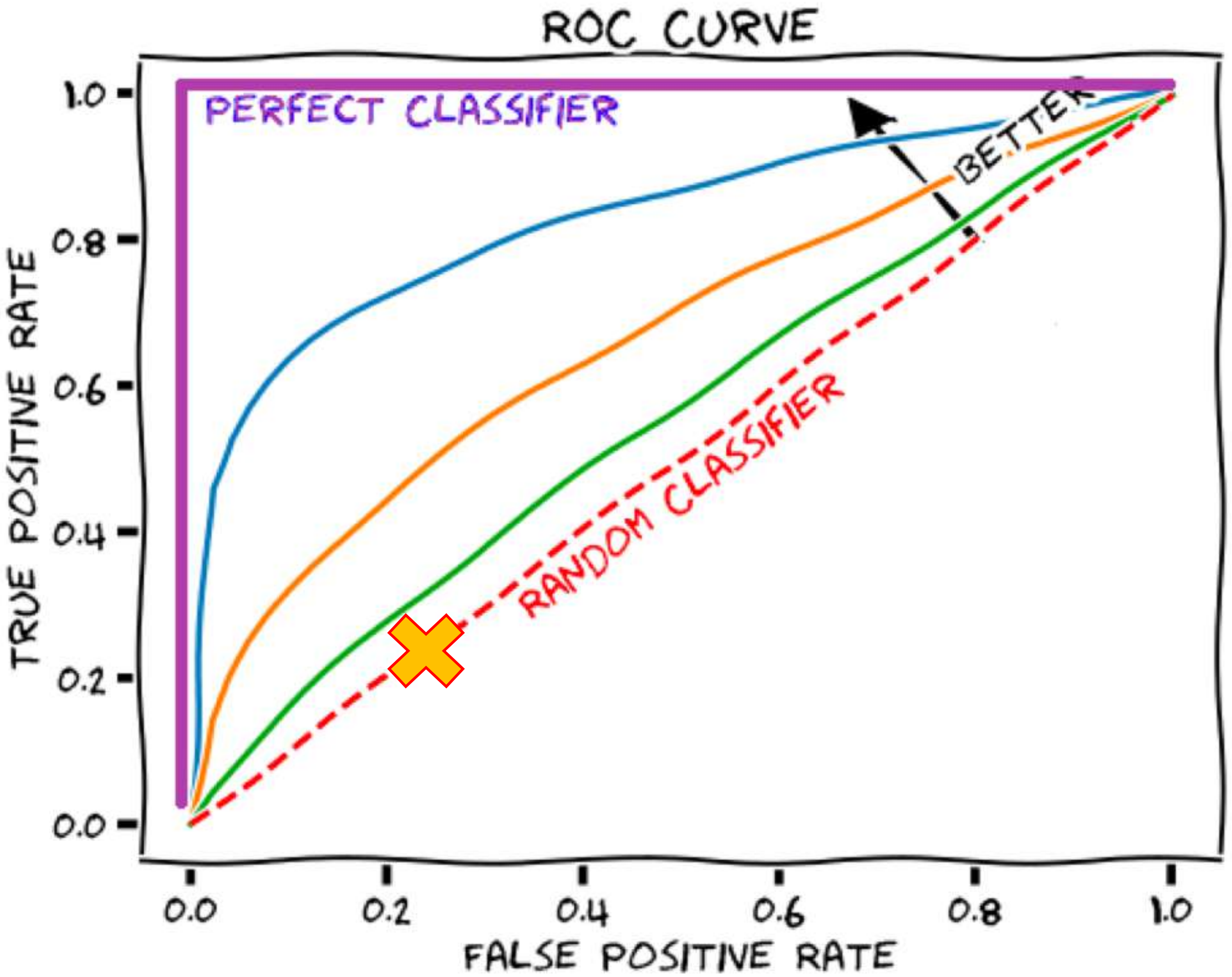
No False Negatives at all

100% Sensitivity

No False Positives at all

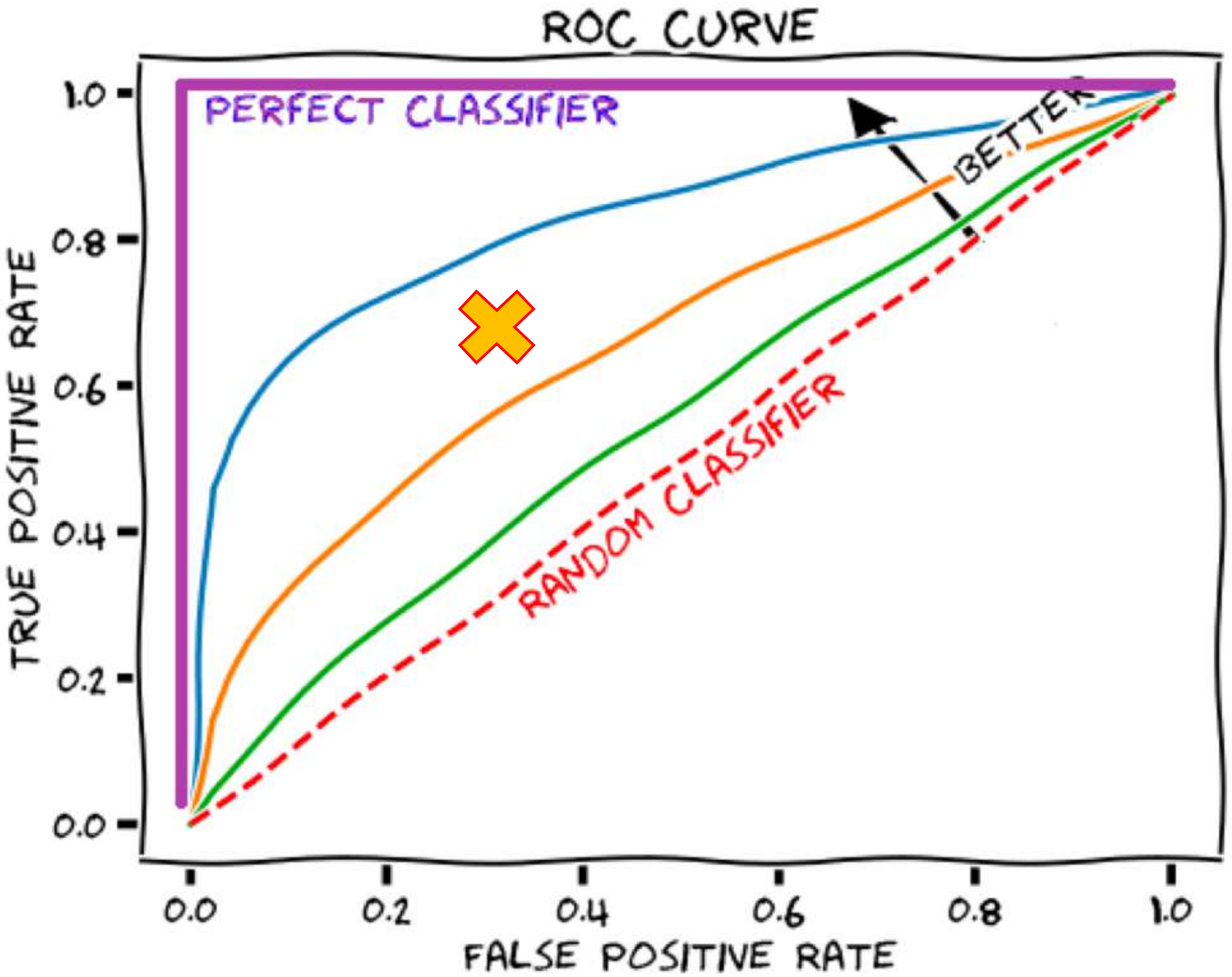
100% Specificity

ROC Curve: Example



Any random guess will result into a point along the diagonal line (**no-discrimination**)

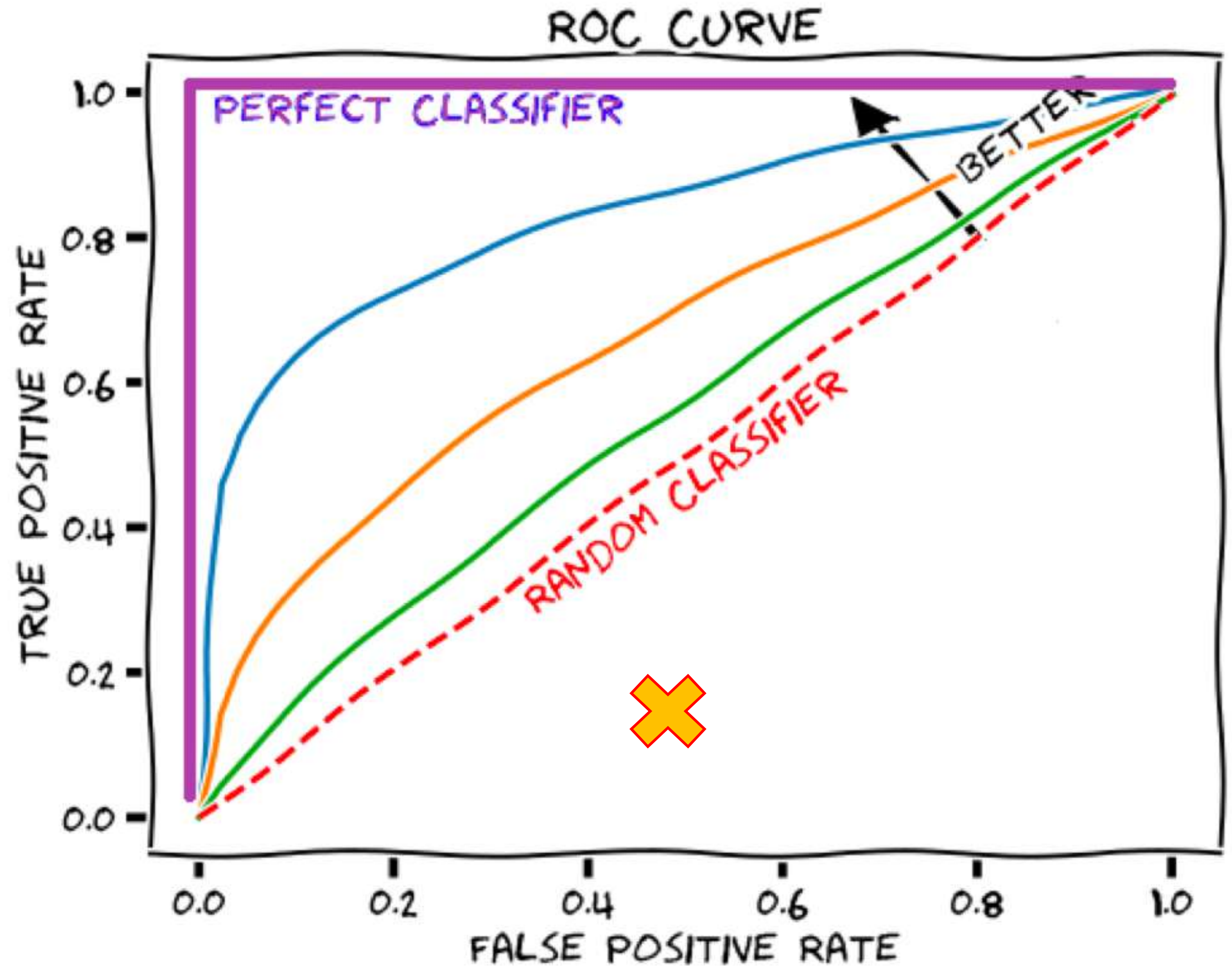
ROC Curve: Example



Any random guess will result into a point along the diagonal line (**no-discrimination**)

Points above the diagonal are good classification results (**better than random**)

ROC Curve: Example



Any random guess will result into a point along the diagonal line (**no-discrimination**)

Points above the diagonal are good classification results (**better than random**)

points below the diagonal are bad classification results (**worse than random**)

ROC Curve: Properties

- Insensitive to changes in class distribution

ROC Curve: Properties

- Insensitive to changes in class distribution
- If the proportion of positive to negative instances changes in a test set, the ROC curves won't change (class skew independece)

ROC Curve: Properties

- Insensitive to changes in class distribution
- If the proportion of positive to negative instances changes in a test set, the ROC curves won't change (class skew independedece)
- This is because the metrics TPR and FPR used for ROC are independent of the class distribution (as opposed to, for instance, accuracy)

ROC Curve: How to Compute Data Points

- Suppose we have a logistic regression classifier

ROC Curve: How to Compute Data Points

- Suppose we have a logistic regression classifier
- We can evaluate its performance using several different values of classification thresholds (e.g., p=0.1, p=0.2, …, p=0.9)

ROC Curve: How to Compute Data Points

- Suppose we have a logistic regression classifier
- We can evaluate its performance using several different values of classification thresholds (e.g., p=0.1, p=0.2, …, p=0.9)
- For each threshold value, we can compute the corresponding (FPR, TPR) coordinate in the ROC space

ROC Curve: How to Compute Data Points

- Suppose we have a logistic regression classifier
- We can evaluate its performance using several different values of classification thresholds (e.g., p=0.1, p=0.2, …, p=0.9)
- For each threshold value, we can compute the corresponding (FPR, TPR) coordinate in the ROC space
- In practice, though, we typically use a single, aggregated score from the ROC curve, i.e., its **Area Under the Curve** (**AUC**)

ROC AUC: Area Under the ROC Curve

- The AUC of a ROC curve is a portion of the unit-square surface

ROC AUC: Area Under the ROC Curve

- The AUC of a ROC curve is a portion of the unit-square surface
- As such, it always ranges between 0 and 1

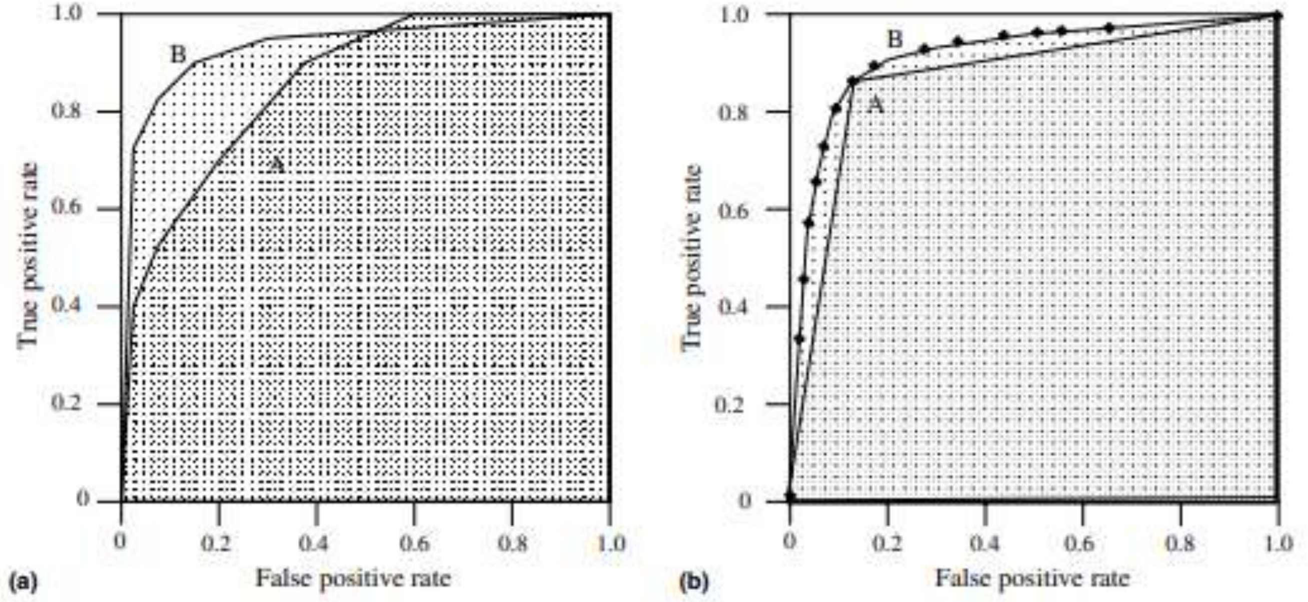
ROC AUC: Area Under the ROC Curve

- The AUC of a ROC curve is a portion of the unit-square surface
- As such, it always ranges between 0 and 1
- The random classifier lies along the diagonal line and has ROC AUC = 0.5

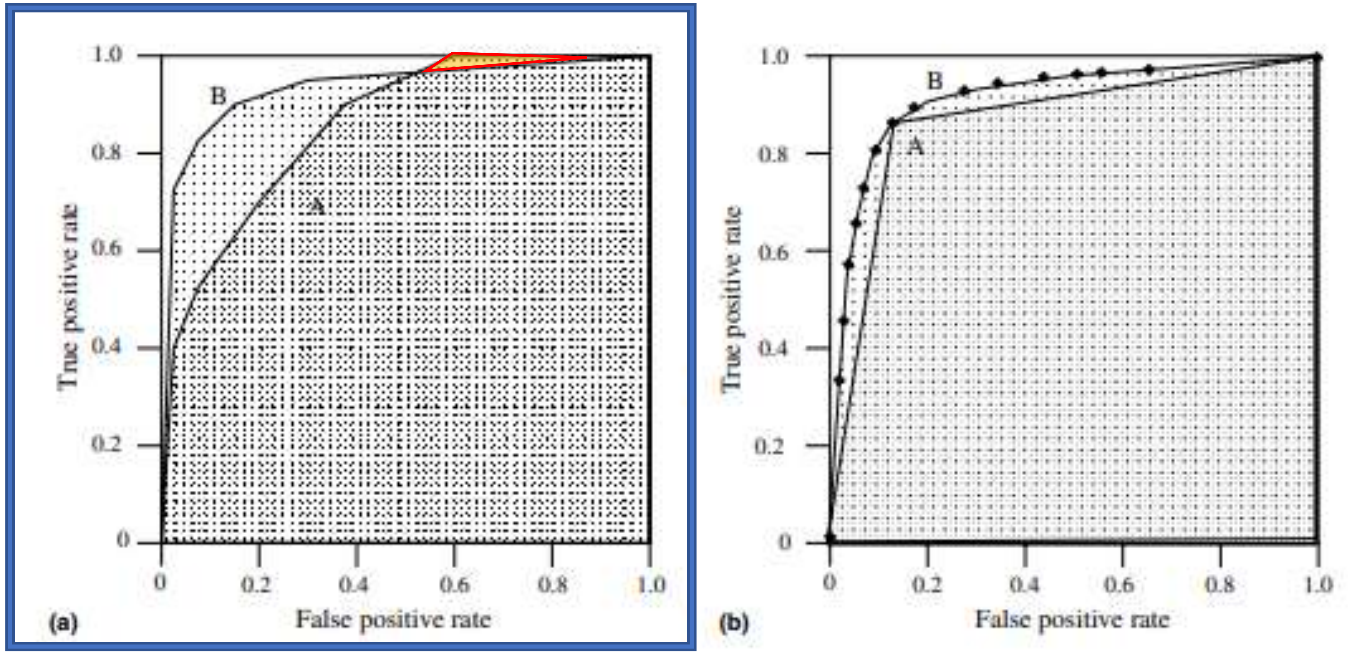
ROC AUC: Area Under the ROC Curve

- The AUC of a ROC curve is a portion of the unit-square surface
- As such, it always ranges between 0 and 1
- The random classifier lies along the diagonal line and has ROC AUC = 0.5
- Any realistic and useful classifier should have ROC AUC > 0.5

ROC AUC: Area Under the ROC Curve



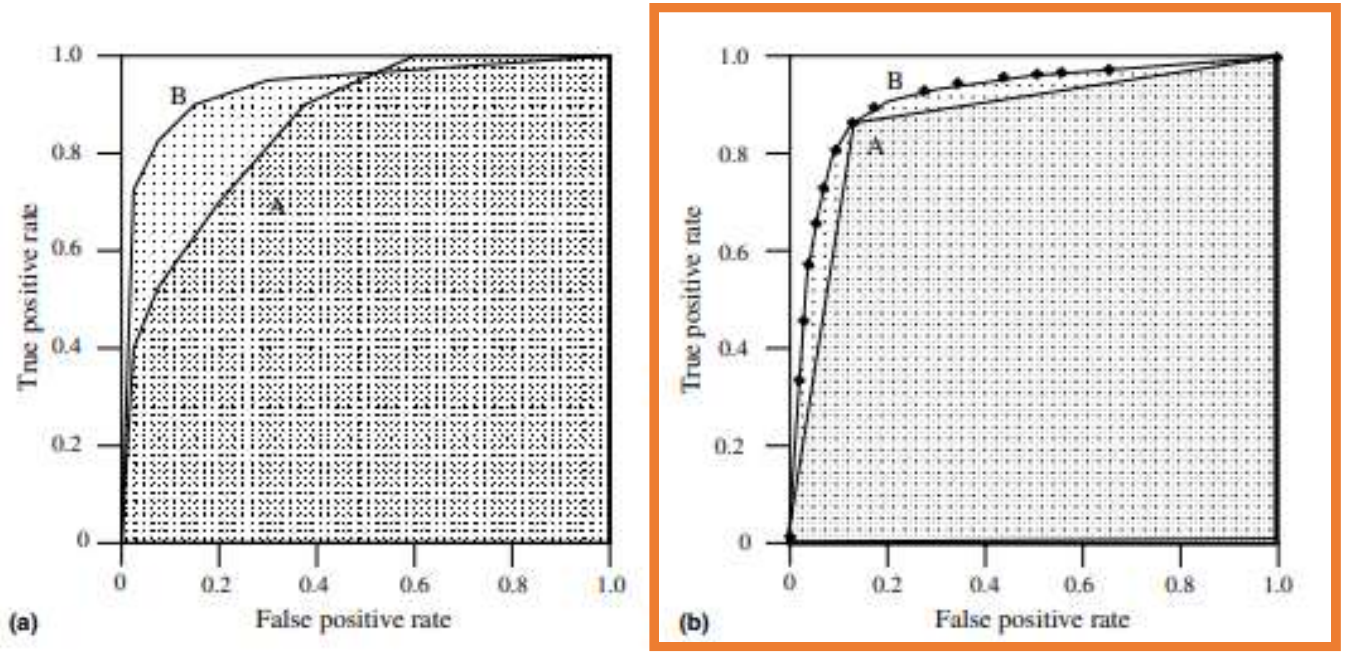
ROC AUC: Area Under the ROC Curve



Classifier B has a greater ROC AUC than classifier A, although the latter may outperform the former at some specific threshold

(e.g., at FPR = 0.6 A is performing better than B)

ROC AUC: Area Under the ROC Curve



B is a scoring classifier (e.g., logistic regression predicting class probabilities)
A is a binary classifier which directly predicts the class label

ROC AUC: Advantages

2 main reasons why ROC AUC is a desirable evaluation metric

ROC AUC: Advantages

$\boxed{\textbf{2}\text{ main reasons why ROC AUC is a desirable evaluation metric}}$

scale-invariant

measures how well predictions are ranked, rather than their absolute values

ROC AUC: Advantages

2 main reasons why ROC AUC is a desirable evaluation metric

scale-invariant

measures how well predictions are ranked, rather than their absolute values

classification-threshold-invariant

measures the quality of the model's predictions irrespective of what classification threshold is chosen

Evaluation Metrics: Last Remarks

- Of course there exist many other evaluation metrics out there

Evaluation Metrics: Last Remarks

- Of course there exist many other evaluation metrics out there
- The metrics we discussed are **offline** metrics as opposed to **online** metrics which are what we ultimately want to measure

Evaluation Metrics: Last Remarks

- Of course there exist many other evaluation metrics out there
- The metrics we discussed are **offline** metrics as opposed to **online** metrics which are what we ultimately want to measure
  - e.g., In online advertising, we want to show ads that get clicked to increase the **revenue** (**online** metric): we measure the "accuracy" of our click model in predicting the **click probability** first (**offline** metric)

Evaluation Metrics: Last Remarks

- Of course there exist many other evaluation metrics out there
- The metrics we discussed are **offline** metrics as opposed to **online** metrics which are what we ultimately want to measure
  - e.g., In online advertising, we want to show ads that get clicked to increase the **revenue** (**online** metric): we measure the "accuracy" of our click model in predicting the **click probability** first (**offline** metric)
- Offline metrics should represent a **good proxy** of the online metric(s) we are ultimately interested in

Take-Home Message of Today

- Several well-known **offline** evaluation metrics for classification models

Take-Home Message of Today

- Several well-known **offline** evaluation metrics for classification models
- Some of them make sense only under specific circumstances (e.g., when class labels are uniform and balanced)

Take-Home Message of Today

- Several well-known **offline** evaluation metrics for classification models
- Some of them make sense only under specific circumstances (e.g., when class labels are uniform and balanced)
- Evaluation metrics can be extended to the case of multi-class although things get more complex

Take-Home Message of Today

- Several well-known **offline** evaluation metrics for classification models
- Some of them make sense only under specific circumstances (e.g., when class labels are uniform and balanced)
- Evaluation metrics can be extended to the case of multi-class although things get more complex
- Offline metrics usually do not coincide with the online metrics we aim to optimize but they must be good **proxies** of those