Metrics of performance in machine learning

Andreea Ioana Sburlea

08.12.2022

Agenda

- Regression and classification metrics
- When to use what?
- Relation between metrics

Good practice

- Apply the evaluation metrics on the same test dataset!

Regression metrics

- Regression models have continuous output. So, we need a metric based on calculating some sort of distance between *predicted* and *ground truth.*
- *Some metrics:*
  - Mean Absolute Error (MAE),
  - Mean Squared Error (MSE),
  - Root Mean Squared Error (RMSE),
  - R^2 (R-Squared).

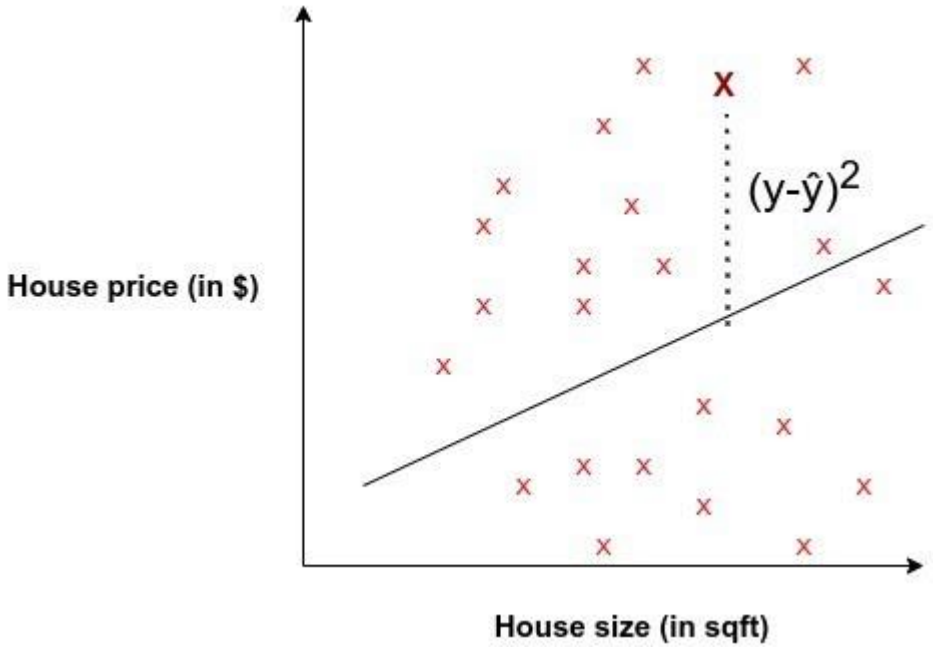
Mean Squared Error (MSE)

- Mean squared error is perhaps the most popular metric used for regression problems. It essentially finds the average of the squared difference between the target value and the value predicted by the regression model.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \check{y}_j)^2$$

where:

- y_j: ground-truth value
- y_hat: predicted value from the regression model
- N: number of datums



Few key points related to MSE:

- It penalizes even small errors by squaring them, which essentially leads to an overestimation of how bad the model is.
- Error interpretation has to be done with squaring factor(scale) in mind.
- Due to the squaring factor, it's fundamentally more prone to outliers than other metrics.

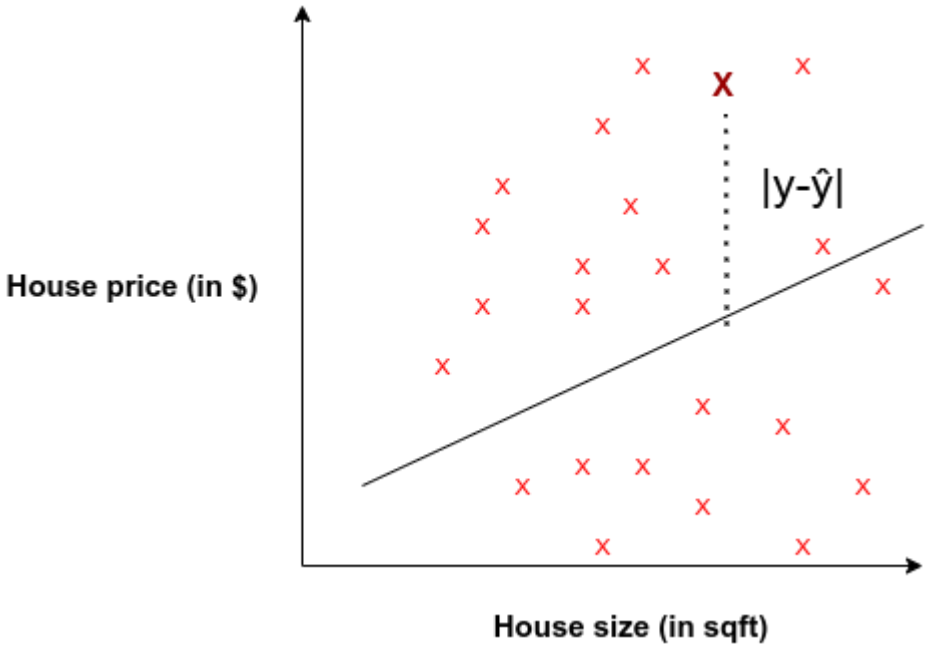
Mean Absolute Error (MAE)

- Mean Absolute Error is the average of the difference between the ground truth and the predicted values. Mathematically, its represented as :

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \check{y}_j|$$

where:

- y_j: ground-truth value
- y_hat: predicted value from the regression model
- N: number of datums



Few key points for MAE

•It’s more robust towards outliers than MSE, since it doesn’t exaggerate errors.

•It gives us a measure of how far the predictions were from the actual output. However, since MAE uses absolute value of the residual, it doesn’t give us an idea of the direction of the error, i.e. whether we’re under-predicting or over-predicting the data.

•Error interpretation needs no second thoughts, as it perfectly aligns with the original degree of the variable.

•Recovers the median of the solution.

Root Mean Squared Error (RMSE)

- Root Mean Squared Error corresponds to the square root of the average of the squared difference between the target value and the value predicted by the regression model. Basically, sqrt(MSE). Mathematically it can be represented as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \breve{y}_j)^2}$$

It addresses a few downsides in MSE.

Few key points related to RMSE:

•It retains the differentiable property of MSE.

•It handles the penalization of smaller errors done by MSE by square rooting it.

•Error interpretation can be done smoothly, since the scale is now the same as the random variable.

•Since scale factors are essentially normalized, it’s less prone to struggle in the case of outliers.

R^2 Coefficient of determination

- R^2 Coefficient of determination actually works as a post metric, meaning it's a metric that's calculated using other metrics.
- The point of even calculating this coefficient is to answer the question **"How much (what %) of the total variation in Y(target) is explained by the variation in X(regression line)"**
- This is calculated using the sum of squared errors. Let's go through the formulation to understand it better.
- Total variation in Y (Variance of Y):

$$SE(\bar{Y}) = \sum_{i=1}^{N} (y_i - \bar{y})^2$$

R^2 Coefficient of determination

- the percentage of variation described the regression line:

$$1 - \frac{SE(line)}{SE(\bar{Y})}$$

- Finally, we have our formula for the coefficient of determination, which can tell us how good or bad the fit of the regression line is:

$$coeff(R^2) = 1 - \frac{SE(line)}{SE(\bar{Y})}$$

R^2 Coefficient of determination

Few intuitions related to R^2 results:

- If the sum of Squared Error of the regression line is small => R^2 will be close to 1 (Ideal), meaning the regression was able to capture 100% of the variance in the target variable.
- Conversely, if the sum of squared error of the regression line is high => R^2 will be close to 0, meaning the regression wasn't able to capture any variance in the target variable.
- You might think that the range of R^2 is (0,1) but it's actually $(-\infty,1)$ because the ratio of squared errors of the regression line and mean can surpass the value 1 if the squared error of regression line is too high (>squared error of the mean).

Adjusted R^2

- The Vanilla R^2 method can mislead the researcher into believing that the model is improving when the score is increasing but in reality, the learning is not happening. This occurs when a model overfits the data, in that case the variance explained will be 100% but the learning hasn't happened. To rectify this, R^2 is adjusted with the number of independent variables.
- Adjusted R^2 is always lower than R^2, as it adjusts for the increasing predictors and only shows improvement if there is a real improvement.

$$R_a^2 = 1 - [(\frac{n-1}{n-k-1}).(1-R^2)]$$

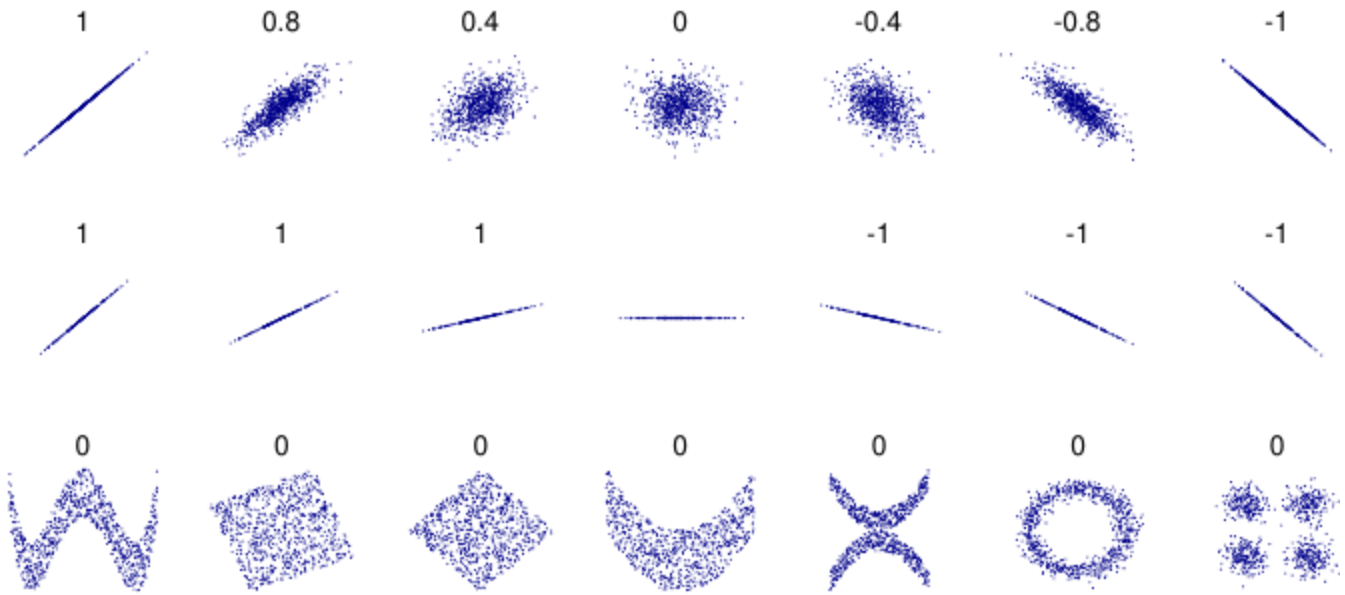
Where:

- n = number of observations
- k = number of independent variables (D=dimensionality of the features)
- Ra^2 = adjusted R^2

(Pearson) Correlation Coefficient

This metric measures how the variables are linearly related. It ranges between $-1 \leq R \leq 1$.

$$R(y, \hat{y}) = \frac{\sum_i (y_i - \mu_{y_i})(\hat{y}_i - \mu_{\hat{y}_i})}{\sqrt{n \sum_i y_i^2 - (\sum y_i)^2} \sqrt{n \sum_i \hat{y}_i^2 - (\sum \hat{y}_i)^2}}$$



Note that the correlation reflects the strength and direction of a linear relationship (top row), but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom).

Other metrics

Kullback-Leibler Divergence

Distance measure between probability distributions p and q. The Cross-Entropy is a simplified version of this loss (with some assumptions).

$$L(p,q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)} dx\right) \qquad L(y,\hat{y}) = \sum_i y_i \log\left(\frac{y_i}{\hat{y}_i}\right)$$

Cosine Similarity

Used to compare vectors.

$$L(y,\hat{y}) = \sum_i \frac{\text{dot}(\hat{y}_i, y_i)}{||\hat{y}_i|| ||y_i||}$$

When to use what?

- MAE – when you have few or no outliers in the data, or when you want to ignore the outliers while fitting your model to the data
- MSE/RMSE – when you have a large number of outliers that you want to emphasize (large penalty) while fitting your model
- The lower value of MAE, MSE, and RMSE implies higher performance of a regression model. However, a higher value of R square is considered desirable.
- R2 shows how well the data fits a curve or line. R2 increases with every predictor added to a model in the training set. As R2 always increases and never decreases, it can appear to be a better fit with the more terms you add to the model. This can be completely misleading.
- Adjusted R2 also indicates how well the data fits a curve or line, but adjusts for the number of variables in a model. If you add more and more useless variables to a model, adjusted r-squared will decrease. If you add more useful variables, adjusted r-squared will increase.

Classification metrics

- Classification models have discrete output, so we need a metric that compares discrete classes in some form. *Classification Metrics* evaluate a model's performance and tell us how good or bad the classification is, but each of them evaluates it in a different way.
- Some metrics:
  - Accuracy
  - Confusion Matrix (not a metric but fundamental to others)
  - Precision and Recall
  - F1-score
  - AUROC

Accuracy

Accuracy

A common **classification only** metric that is easily interpretable by humans. Its range is $[0, 1]$, with the best value being 1. If multiplied by 100 it can be interpreted as a percentage.

$$\text{Acc}(y, \hat{y}) = n^{-1} \sum_i 1[y_i = \hat{y}_i]$$

Where $1[x]$ is the indicator function, returning 1 if x is true, and 0 if x is false.

Top-k Accuracy

Accuracy computed as considering any of the top k class predictions as correct. Typically used with classifiers that can rank their class predictions, and for tasks with a large number of classes.

Binary Accuracy

- It measures how many observations, both positive and negative, were correctly classified.

$$ACC = \frac{tp + tn}{tp + fp + tn + fn}$$

- You **shouldn’t use accuracy on imbalanced problems**. Then, it is easy to get a high accuracy score by simply classifying all observations as the majority class.

Be careful with “Accuracy”

The simplest measure of performance would be the fraction of items that are correctly classified, or the “accuracy” which is:

$$\frac{tp + tn}{tp + tn + fp + fn}$$

But this measure is dominated by the larger set (of positives or negatives) and favors trivial classifiers.

e.g. if 5% of items are truly positive, then a classifier that always says “negative” is 95% accurate.

Balanced Accuracy

When a dataset has class imbalance, the standard accuracy can be misleading. This problem can be fixed with the balanced accuracy metric:

$$\text{BalancedAcc}(y, \hat{y}) = C^{-1} \sum_{c} \text{Acc}_c(y, \hat{y}) \tag{9}$$

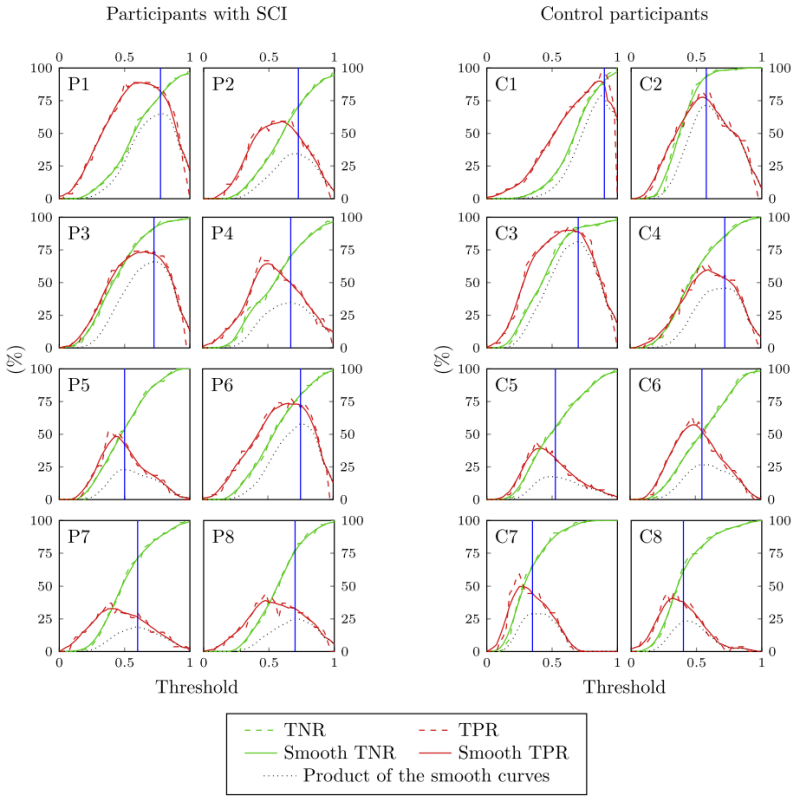
Where $\text{Acc}_c(y, \hat{y})$ is the accuracy computed only for class c. The balanced accuracy is computed as the mean of per-class accuracies.

Can also be computed from the binary confusion matrix:

$$\text{BalancedAcc} = \frac{\text{TPR} + \text{TNR}}{2} \tag{10}$$

Threshold selection

- Since the accuracy score is calculated on the predicted classes (not prediction scores) we **need to apply a certain threshold** before computing it. The obvious choice is the threshold of 0.5 but it can be suboptimal.



The blue line represents the threshold that maximizes the product of the smooth curves, which is represented by a black dotted line.

Figure from Lopes-Dias, C., Sburlea, A. I., Breitegger, K., Wyss, D., Drescher, H., Wildburger, R., & Müller-Putz, G. R. (2021). Online asynchronous detection of error-related potentials in participants with a spinal cord injury using a generic classifier. *Journal of Neural Engineering*, *18*(4), 046022.

When is accuracy a good metric?

- When your **problem is balanced** using accuracy is usually a good start. An additional benefit is that it is really easy to explain it to other people in your project.
- When **every class is equally important** to you.

- Reviewing both precision and recall is useful in cases where there is an imbalance in the observations between the two classes. Specifically, there are many examples of no event (class 0) and only a few examples of an event (class 1).

Confusion matrix

		Predicted condition			
	Total population $= P + N$	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP),** **hit**	**False negative (FN),** **type II error, miss,** **underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP),** **type I error, false alarm,** **overestimation**	**True negative (TN),** **correct rejection**	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Confusion matrix

Confusion Matrix

It is a metric that produces a matrix value, indicating how classes are predicted and confused with another class. It is made with matrix M initialized with zeros, and then:

1. Take integer index class predictions $\hat{y}_i$ and their correct labels y_i.
2. For each item i in the dataset, set:

$$M[y_i, \hat{y}_i] = M[y_i, \hat{y}_i] + 1$$

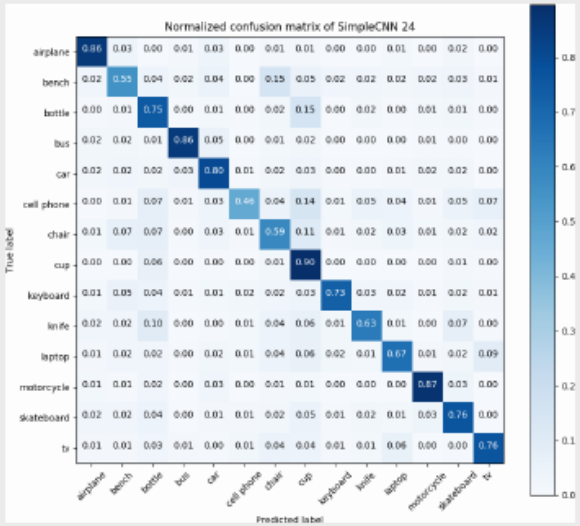
3. Optionally the matrix M can be normalized across rows by dividing each element by its sum.

$$M_{ij} = \frac{M_{ij}}{\sum_i M_{ij}}$$

Confusion matrix

Confusion Matrix

The interpretation of a confusion matrix is that row i represents the true labels, while columns j are the predictions. Elements in the diagonal represent correct predictions, while off diagonal elements are incorrect predictions.



Confusion matrix

Confusion Matrix in Binary Classification

For binary classification, the confusion matrix is always 2×2 and its elements have a special meaning:

$$M = \begin{pmatrix} \text{True Positive} & \text{False Negative} \\ \text{False Positive} & \text{True Negative} \end{pmatrix} \tag{3}$$

These elements have varying importance depending on the application, for example, medical systems usually require a low number of false positives. Other metrics are derived:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{Recall, Sensitivity, or TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

Confusion matrix

Confusion Matrix in Binary Classification

$$\text{Specificity, or TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{6}$$

A single score can be provided for a classifier, usually the F1 score, computed as the Harmonic mean between Precision (P) and Recall (R):

$$\text{F1} = 2\frac{PR}{P + R} \tag{7}$$

Precision

		Predicted	
		Has Cancer	Doesn't Have Cancer
Ground Truth	Has Cancer	TP	FN
	Doesn't Have Cancer	FP	TN

- Precision is the ratio of true positives and total positives predicted:
- The precision metric focuses on **Type-I errors**(FP). A **Type-I error occurs when we reject a true null Hypothesis (H^0 : The individual has cancer)**. So, in this case, Type-I error is incorrectly labeling non-cancer patients

$$P = \frac{TP}{TP+FP} = \frac{\text{Cancer patients correctly identified}}{\text{Cancer patients correctly identified+incorrectly labelled non-cancerous patients as cancerous}}$$

- A precision score towards 1 will signify that your model didn't miss any true positives, and is able to classify well between correct and incorrect labeling of cancer patients. *What it cannot measure is the existence of Type-II error, which is false negatives – cases when a cancerous patient is identified as non-cancerous.*
- A low precision score (<0.5) means your classifier has a high number of false positives which can be an outcome of imbalanced class or untuned model hyperparameters. In an imbalanced class problem, you have to prepare your data beforehand with over/under-sampling or focal loss in order to curb FP/FN.

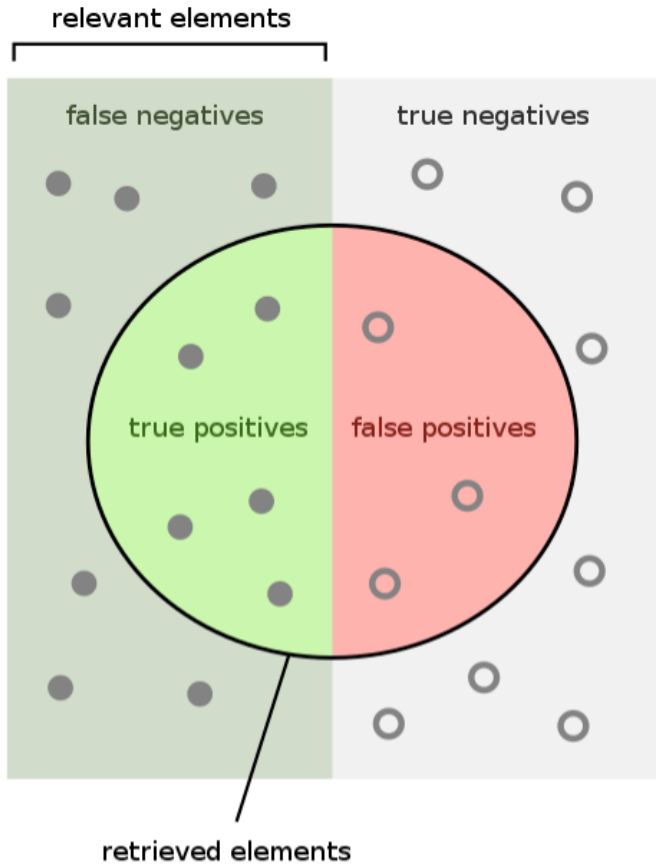
Recall/Sensitivity/ Hit-Rate

		Predicted	
		Has Cancer	Doesn't Have Cancer
Ground Truth	Has Cancer	TP	FN
	Doesn't Have Cancer	FP	TN

- A **Recall** is essentially the ratio of true positives to all the positives in ground truth.

$$R = \frac{TP}{TP+FN} = \frac{\text{Cancer patients correctly identified}}{\text{Cancer patients correctly identified+incorrectly labelled cancerous patients as non-cancerous}}$$

- 0<R<1
- The recall metric focuses on **type-II errors**(FN). A type-II error occurs when we **accept a false null hypothesis(H^0)**. So, in this case, type-II error is incorrectly labeling non-cancerous patients as cancerous.
- Recall towards 1 will signify that your model didn't miss any true positives, and is able to classify well between correctly and incorrectly labeling of cancer patients.
- *What it cannot measure is the existence of type-I error which is false positives i.e the cases when a cancerous patient is identified as non-cancerous.*
- A low recall score (<0.5) means your classifier has a high number of false negatives which can be an outcome of imbalanced class or untuned model hyperparameters. In an imbalanced class problem, you have to prepare your data beforehand with over/under-sampling or focal loss in order to curb FP/FN



How many retrieved items are relevant?

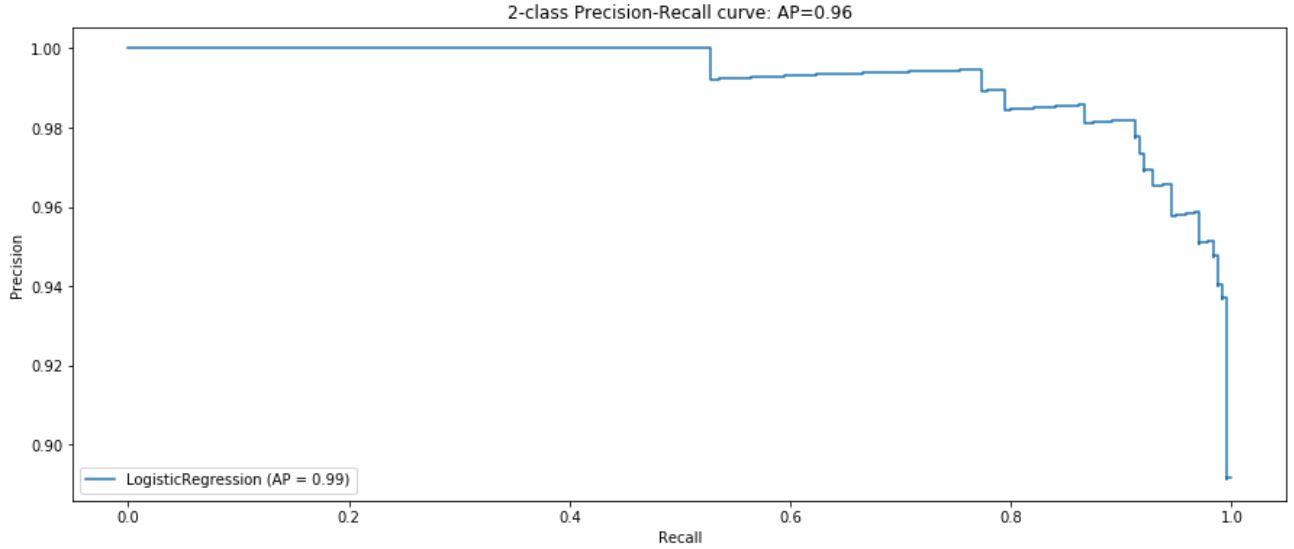
$$\text{Precision} = \frac{\text{green}}{\text{green} + \text{red}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{green}}{\text{green} + \text{green}}$$

Precision-Recall tradeoff (PR curve)

- To improve your model, you can either improve precision or recall – **but not both!** If you try to reduce cases of cancerous patients being labeled as non-cancerous (FN/type-II), no direct effect will take place on non-cancerous patients being labeled as cancerous.



- This tradeoff highly impacts real-world scenarios, so we can deduce that precision and recall alone aren't very good metrics to rely on and work with. That's the reason why many online competitions urge the submission metric to be a combination of precision and recall.

Average precision and mAP

- The area under the PR curve can be computed as the average precision (AP). For a multi-label settings, an AP value an be computed for each class (as the binary class vs no-class problem), and then the mean average precision (mAP) metric can be used to characterize the classifiers' predictions.
- Multi-label means multiple correct classes can be predicted for the same input

F1 score

- it combines precision and recall into one metric by calculating the harmonic mean between those two. It is actually a **special case of** the more general function **F beta**

$$F_{beta} = (1+\beta^2)\frac{precision * recall}{\beta^2 * precision + recall} \qquad F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

- When choosing beta in your F-beta score **the more you care about recall** over precision **the higher beta** you should choose. For example, with F1 score we care equally about recall and precision with F2 score, recall is twice as important.

- It is important to remember that F1 score is calculated from Precision and Recall which, in turn, are calculated on the predicted classes (not prediction scores).

When should you use it?

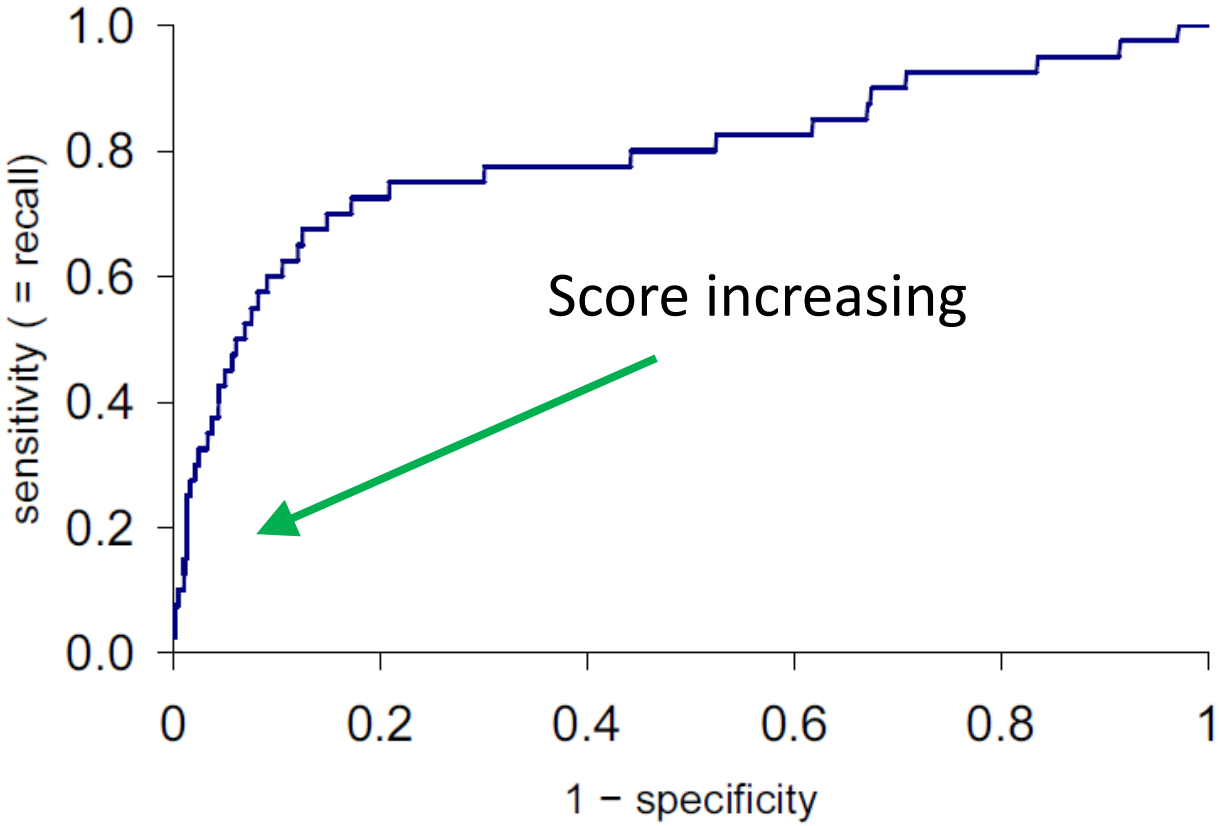
- Pretty much in every binary classification problem where you care more about the **positive class**.
- It **can be easily explained** to other members in a project.

ROC plots

ROC is Receiver-Operating Characteristic. ROC plots

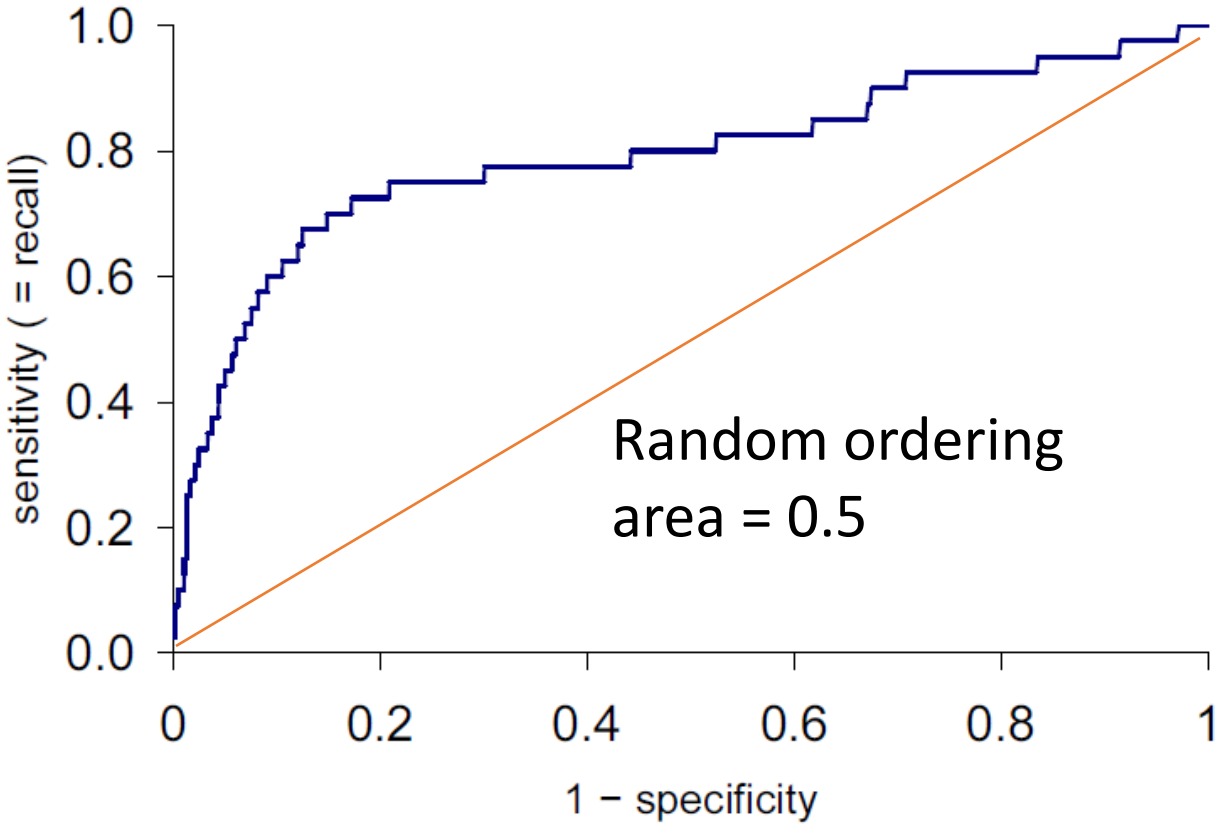
Y-axis: true positive rate = tp/(tp + fn), same as recall

X-axis: false positive rate = fp/(fp + tn) = 1 - specificity



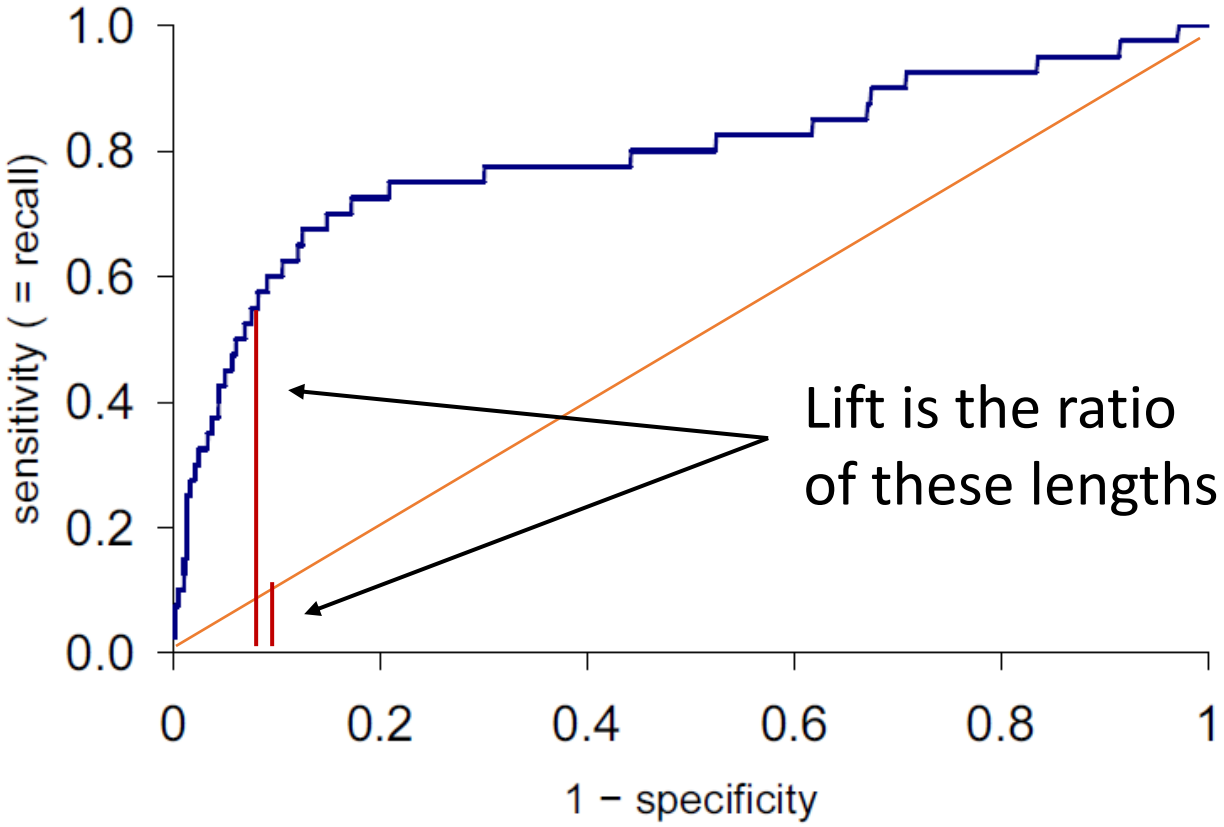
ROC AUC

ROC AUC is the “Area Under the Curve” – a single number that captures the overall quality of the classifier. It should be between 0.5 (random classifier) and 1.0 (perfect).



Lift Plot

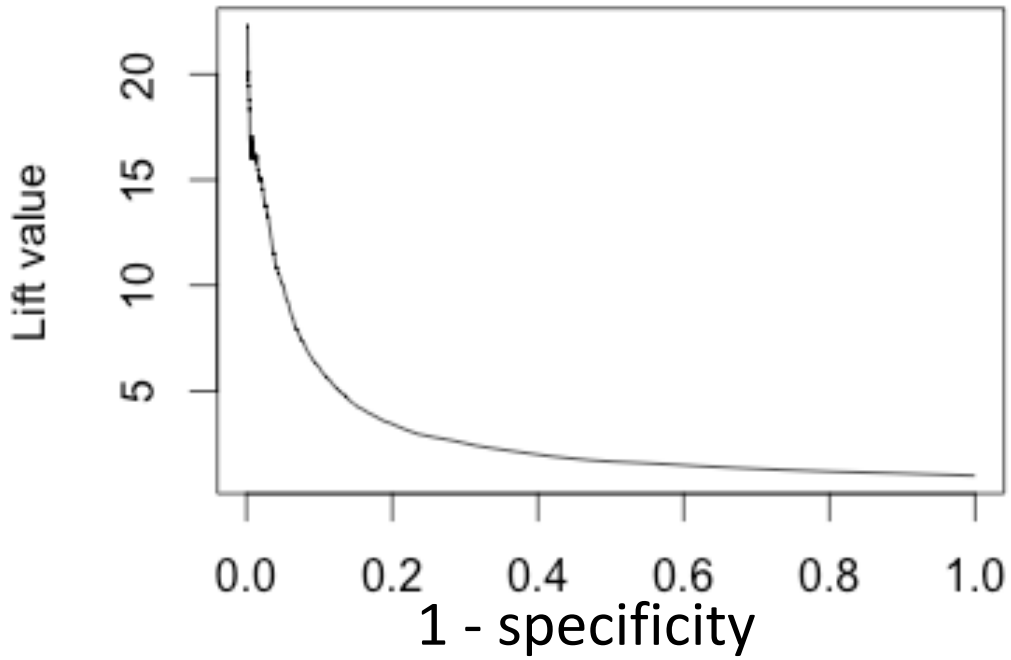
A derivative of the ROC plot is the lift plot, which compares the performance of the actual classifier/search engine against random ordering, or sometimes against another classifier.



Lift Plot

Lift plots emphasize initial precision (typically what you care about), and performance in a problem-independent way.

Note: The lift plot points should be computed at regular spacing, e.g. 1/100 or 1/1000. Otherwise the initial lift value can be excessively high, and unstable.



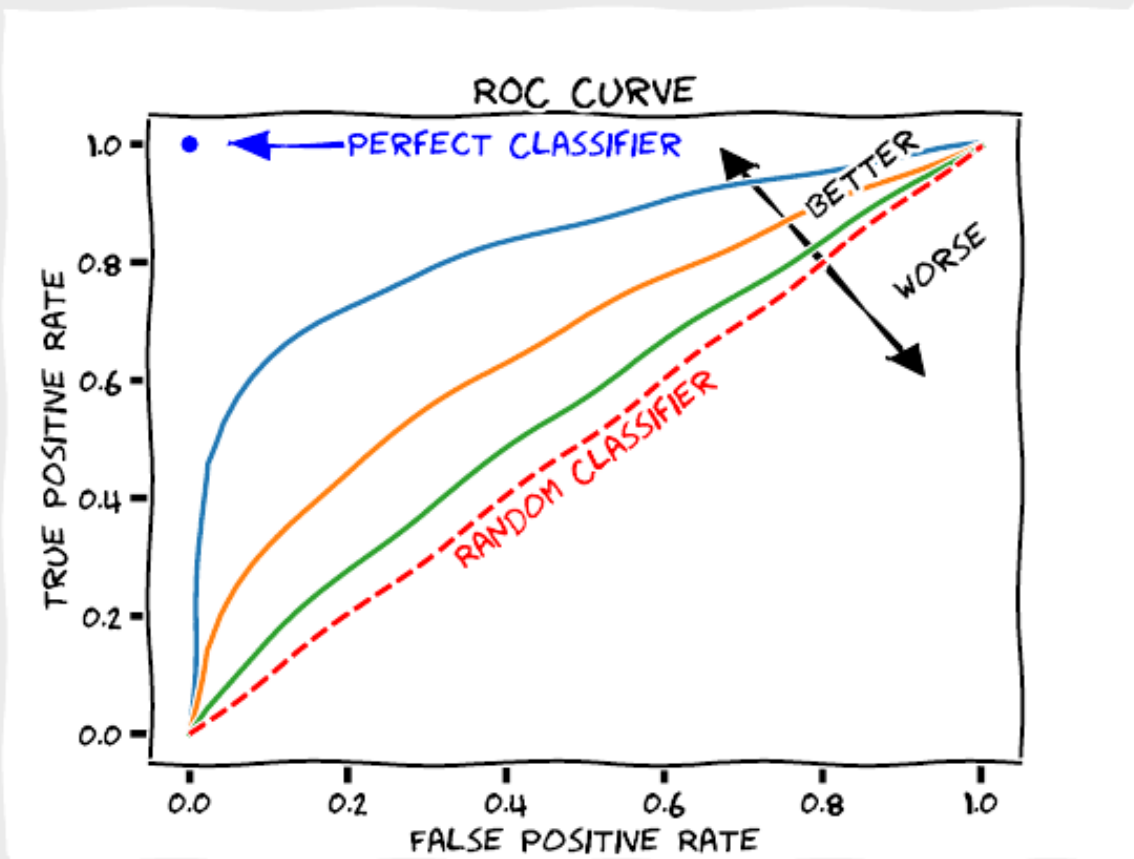
Receiver Operating Characteristic (ROC) Curve

When a binary classifier produces a probability for the positive class, then there is a need to select a threshold value T to discriminate between negative or positive classes:

$$\text{Decision}(p) = \begin{cases} \text{Positive Class} & \text{if } p \geq T \\ \text{Negative Class} & \text{if } p < T \end{cases} \tag{8}$$

By varying the value of T, the values of all binary metrics change.
This can be applied to any score that needs thresholding.
Then by varying the threshold, an operating space is produced.
Typically the TPR (Sensitivity) and FPR (1– Specificity) metrics are plotted.
The curve in operating space is a characterization of the classifier's quality.

Receiver Operating Characteristic (ROC) Curve



Receiver Operating characteristic (ROC) curve

- Example: Let's say we have a binary classification problem (e.g., left/right hand MI). Here we look at the probability output of our classifier for one of the classes in the first 5 observations: [0.7, 0.4, 0.25, 0.6, 0.9]

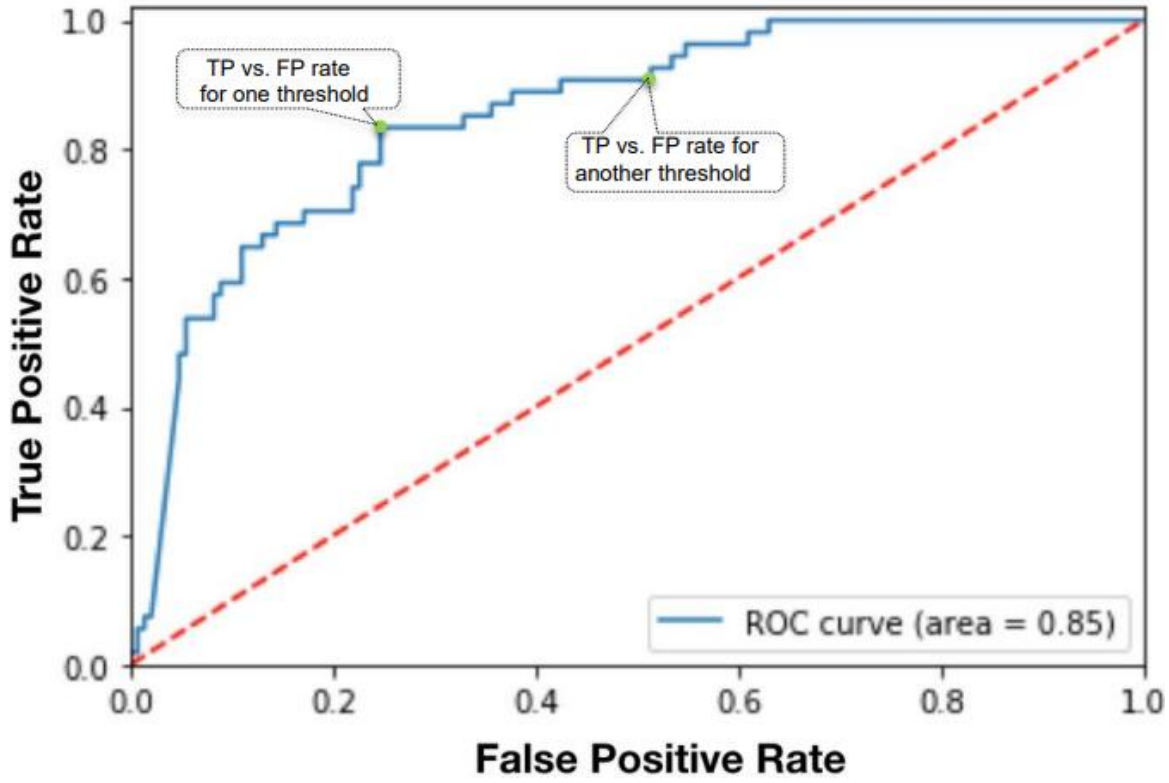
To assess the prediction of the labels we need to choose a threshold:

- Cut-off threshold = 0.5 => predicted labels [1 0 0 1 1]
- Cut-off threshold = 0.2 => predicted labels [1 1 1 1 1]
- Cut-off threshold = 0.8 => predicted labels [0 0 0 0 1]

For different thresholds we obtain different predictions.

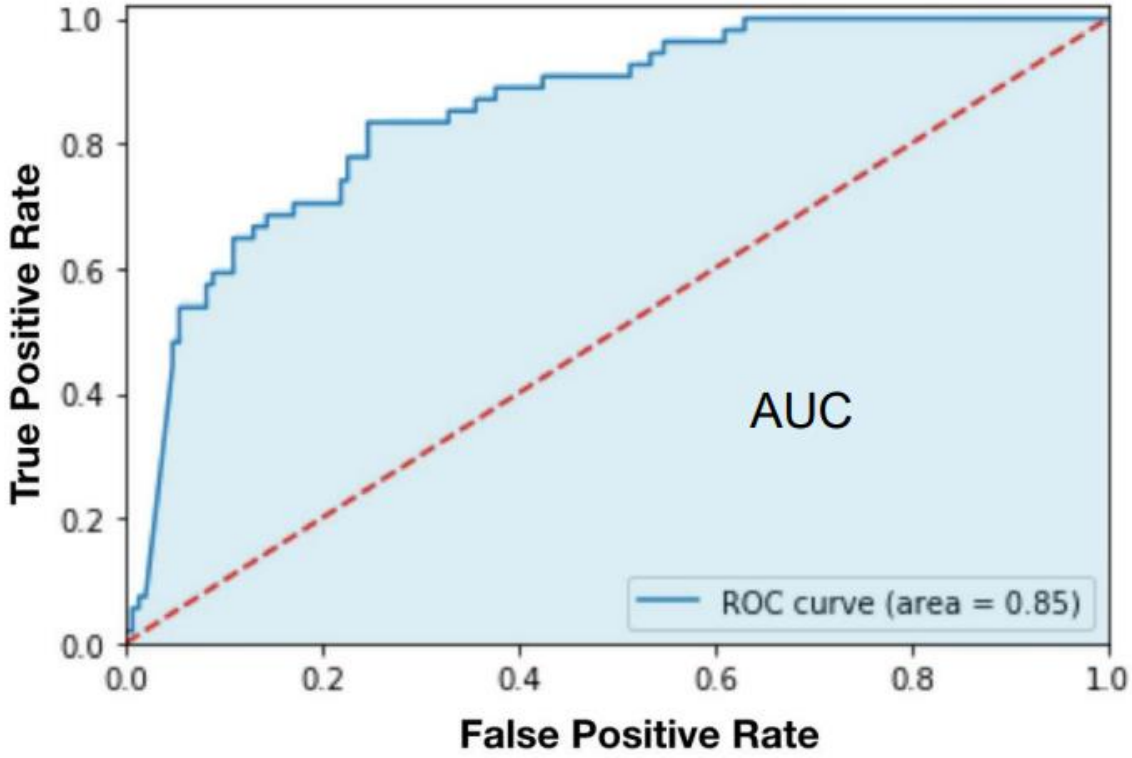
Receiver Operating characteristic (ROC) curve

- shows the distribution of probabilities of the true positive rate (TPR) against the false positive rate (FPR) for various threshold values



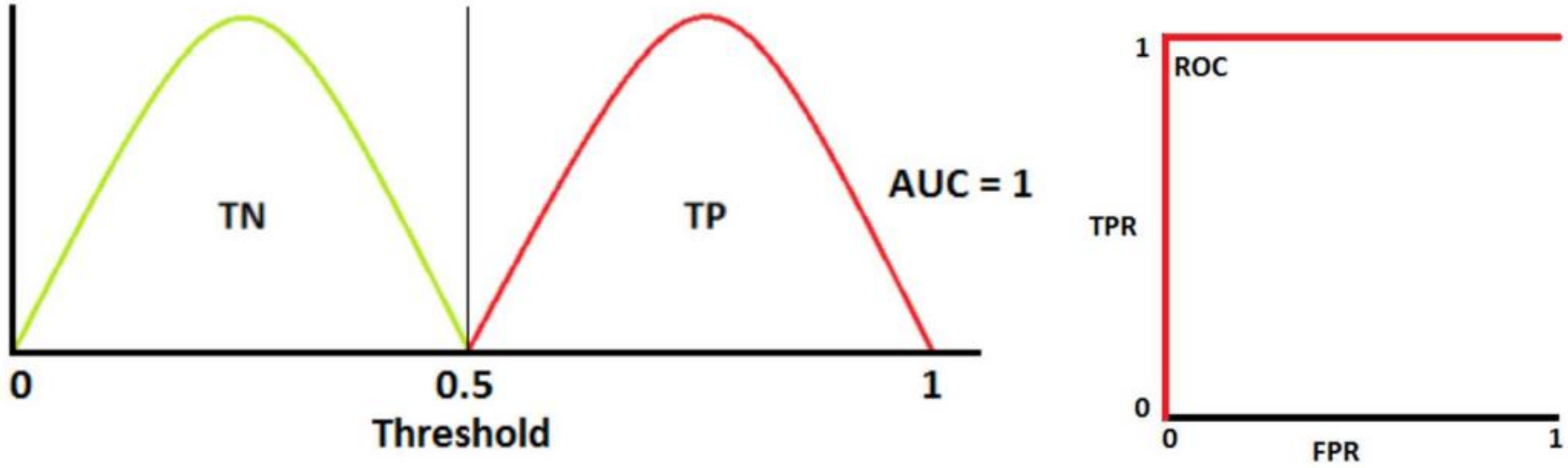
Area under the curve (AUC)

- tells us how well a model can distinguish between classes
- it’s a scalar between 0 and 1



How do these metrics relate?

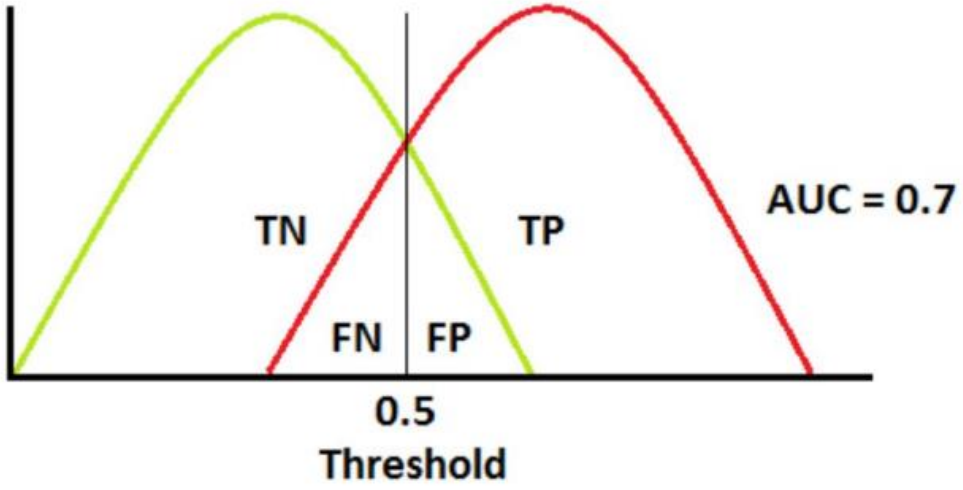
- Let’s get back to our example of a binary classification (left vs. right hand MI). Below are the distributions of the probabilities of the 2 classes.

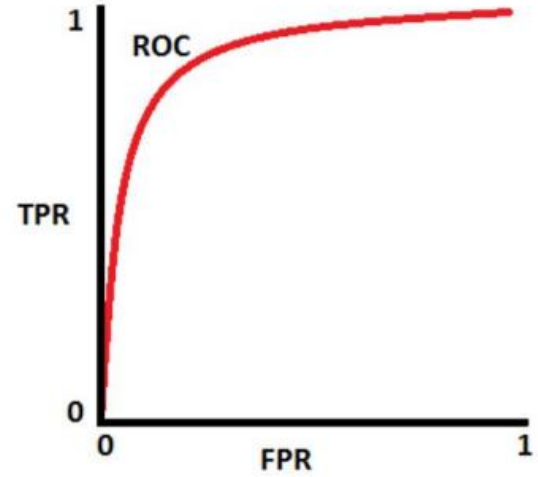


- This is an ideal situation. The two curves don’t overlap, so the classifier can perfectly distinguish between the two classes.

Figures from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

How do these metrics relate?

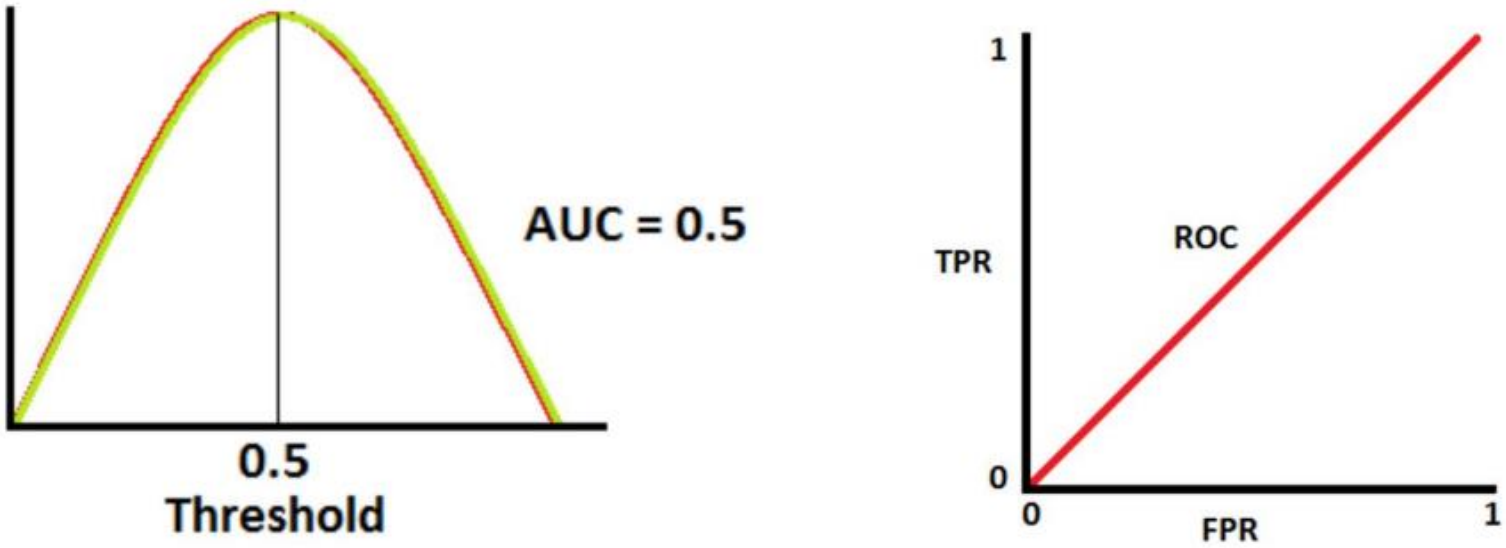




- When two distributions overlap, we introduce type 1 (FP) and type 2 (FN) error. Depending upon the threshold, we can find a trade-off between them. When AUC is 0.7, it means there is 70% chance that the classifier will be able to distinguish between the two classes.

Figures from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

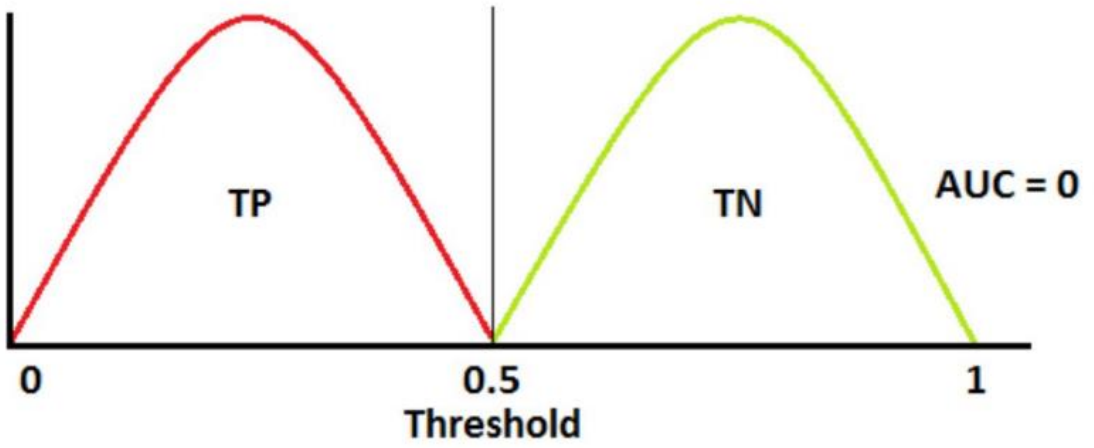
How do these metrics relate?

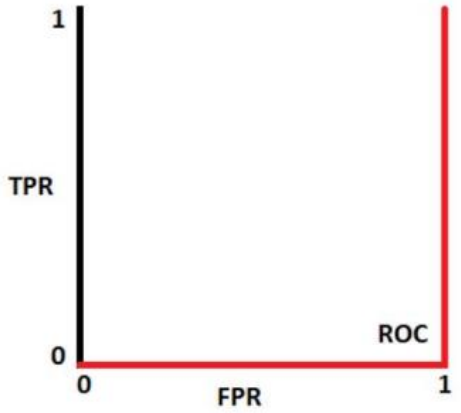


- This is the worst situation. When AUC is 0.5, our classifier has no discrimination capacity between the two classes.

Figures from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

How do these metrics relate?





- When AUC is 0 (or very close to it), our classifier is actually predicting the opposite class, so class 1 gets predicted as class 2 and vice versa.

Figures from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Take home messages

- Regression and classification metrics measure different properties of the model with respect to the data. Make sure you know what type of problem you are solving before choosing a metric.
- One metric can hint at some information about other metrics.
- It is useful to compute more than one metric in order to evaluate your choice.

Questions to think about

- Why is accuracy a worse metric for classification than F1?
- Can accuracy be used to evaluate a regression model?
- Is precision more important than recall?