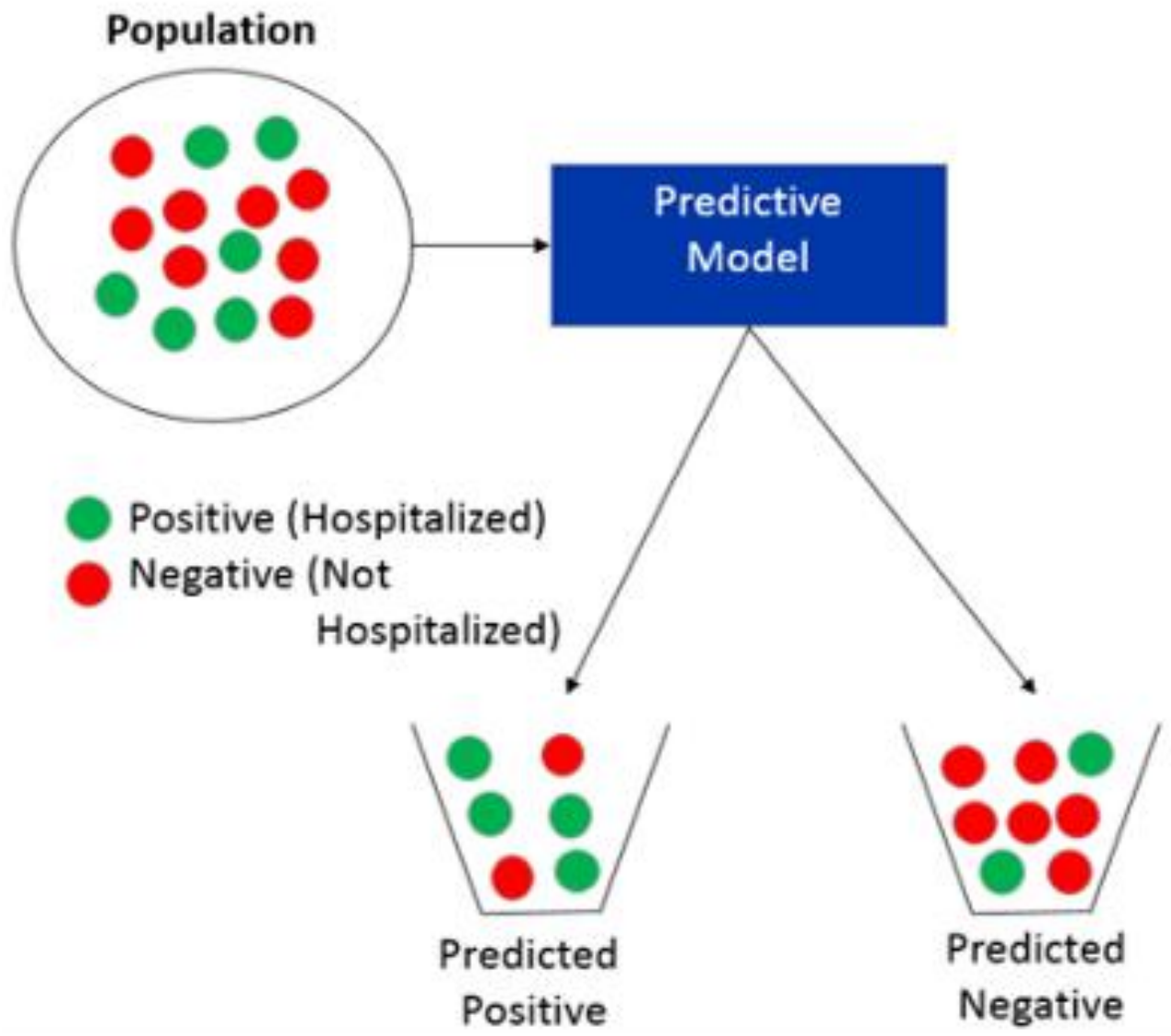
Evaluation Metrics

Classification Metrics


Population
Predictive Model
Positive (Hospitalized)
Negative (Not Hospitalized)
Predicted Positive
Predicted Negative

Confusion Matrix

Given an actual label and a predicted label, the first thing we can do is divide our samples in 4 buckets:

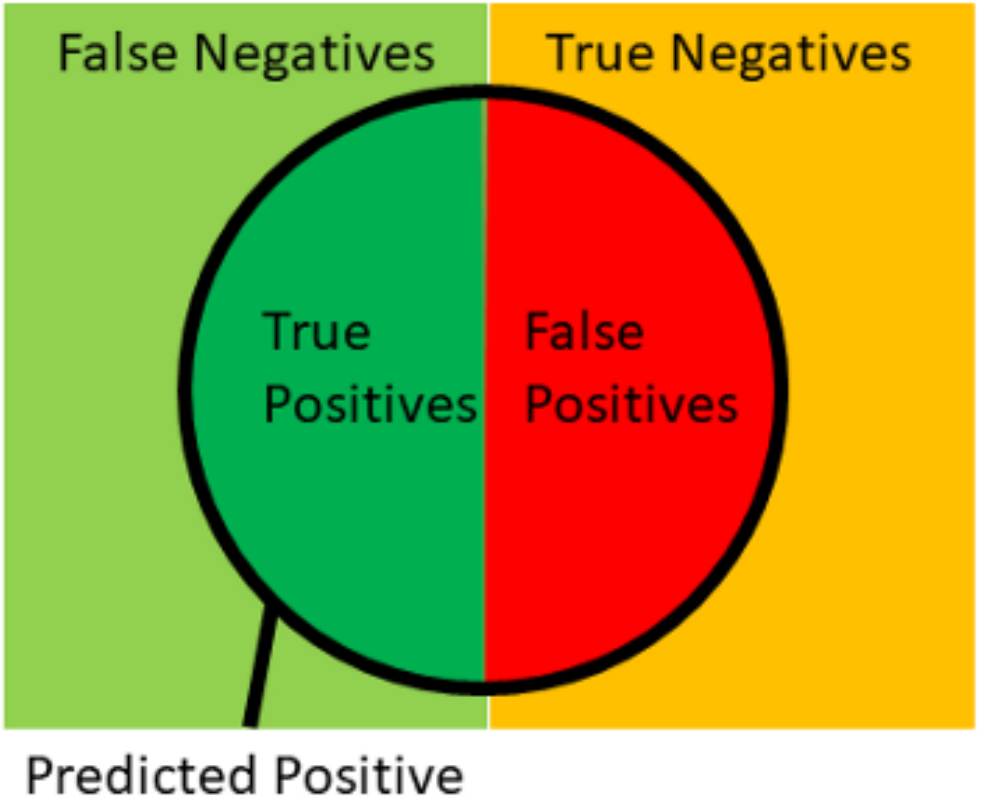
True positive — actual = 1, predicted = 1

False positive — actual = 0, predicted = 1

False negative — actual = 1, predicted = 0

True negative — actual = 0, predicted = 0

Confusion Matrix



Confusion Matrix		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Accuracy Score

- The most common metric for classification is accuracy, which is the fraction of samples predicted correctly as shown below:

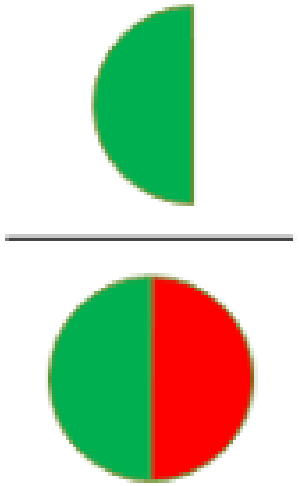
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} =$$

Fraction predicted correctly

Precision Score

- Precision is the fraction of predicted positives events that are actually positive as shown below:

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



Fraction of predicted positives that are actually positive

Recall Score

- Recall (also known as sensitivity) is the fraction of positives events that you predicted correctly as shown below:

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} = \frac{\text{[illegible]}}{\text{[illegible]}}$$

Fraction of positives predicted correctly

Confusion Matrix for Binary Classification

		True/Actual	
		Positive ()	Negative
Predicted	Positive ()	5	1
	Negative	2	2

Confusion Matrix for Multiclass

		True/Actual		
		Cat ( )	Fish ( )	Hen ( )
Predicted	Cat ( )	4	6	3
	Fish ( )	1	2	0
	Hen ( )	1	2	6

Precision vs Recall

- For example, a hospital has 200 patients where 100 have disease
- A Machine Learning model is retrieving 80 patients and saying that those 80 have disease and rest of 120 don’t have disease
- Imagine that 40 people out of the 80 patients retrieved by the model have disease
- Calculate precision and Recall
- Now which one is important?

Precision vs Recall

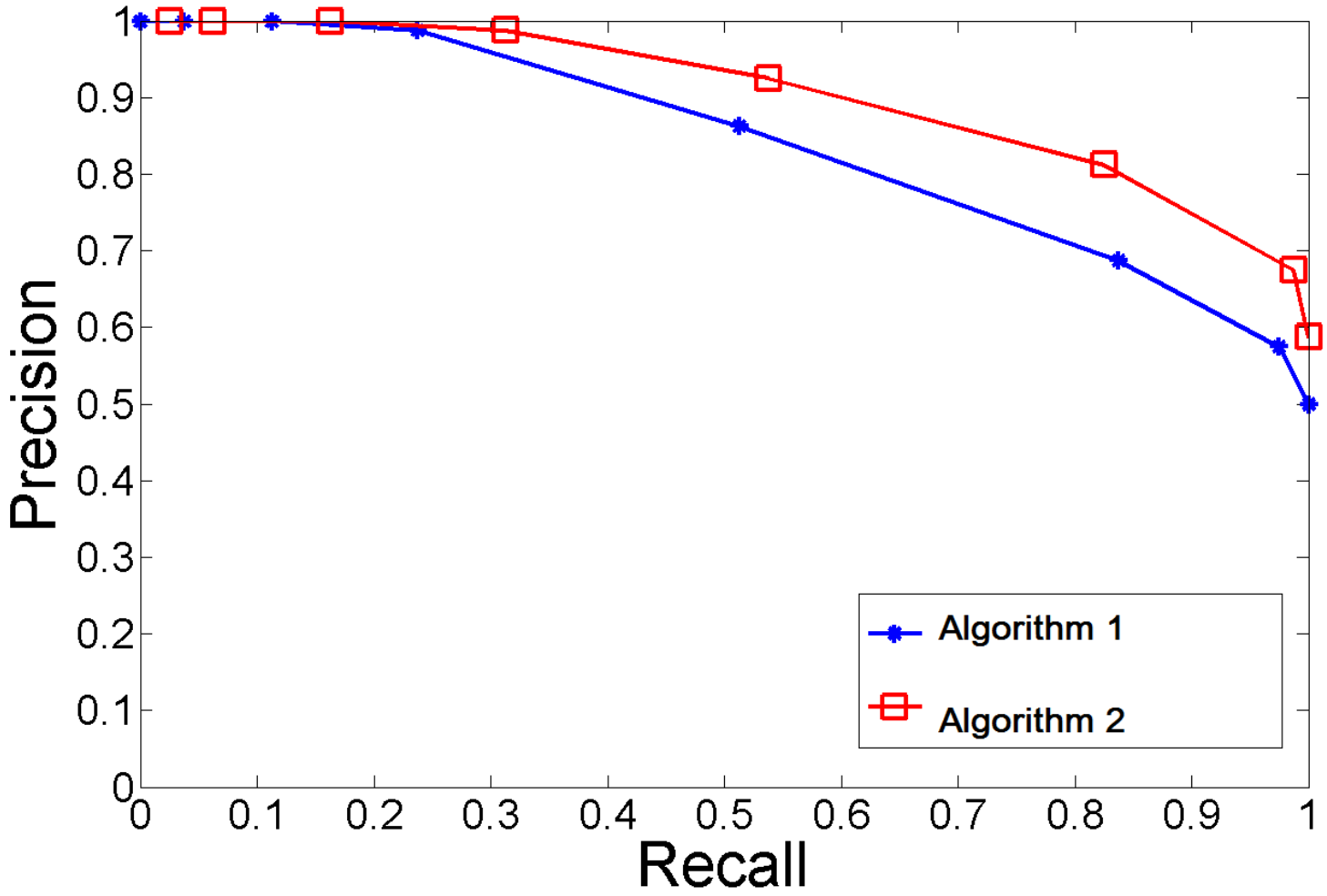
- Precision is 40/80 = 0.5, Recall is 40/100 = 0.4

F1 Score

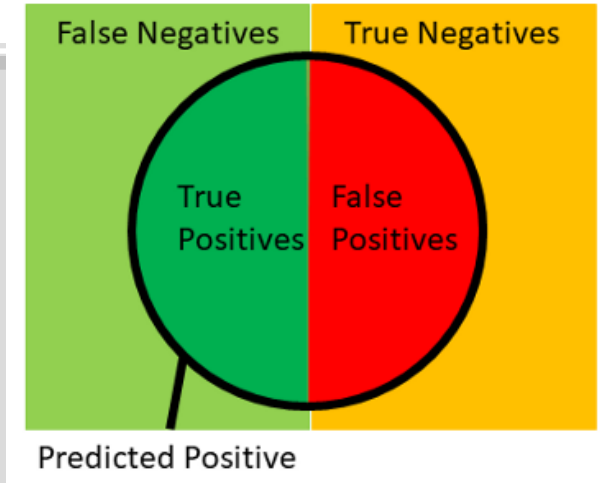
- The f1 score is the harmonic mean of recall and precision, with a higher score as a better model. The f1 score is calculated using the following formula:

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * (precision * recall)}{precision + recall}$$

Precision Recall Curve

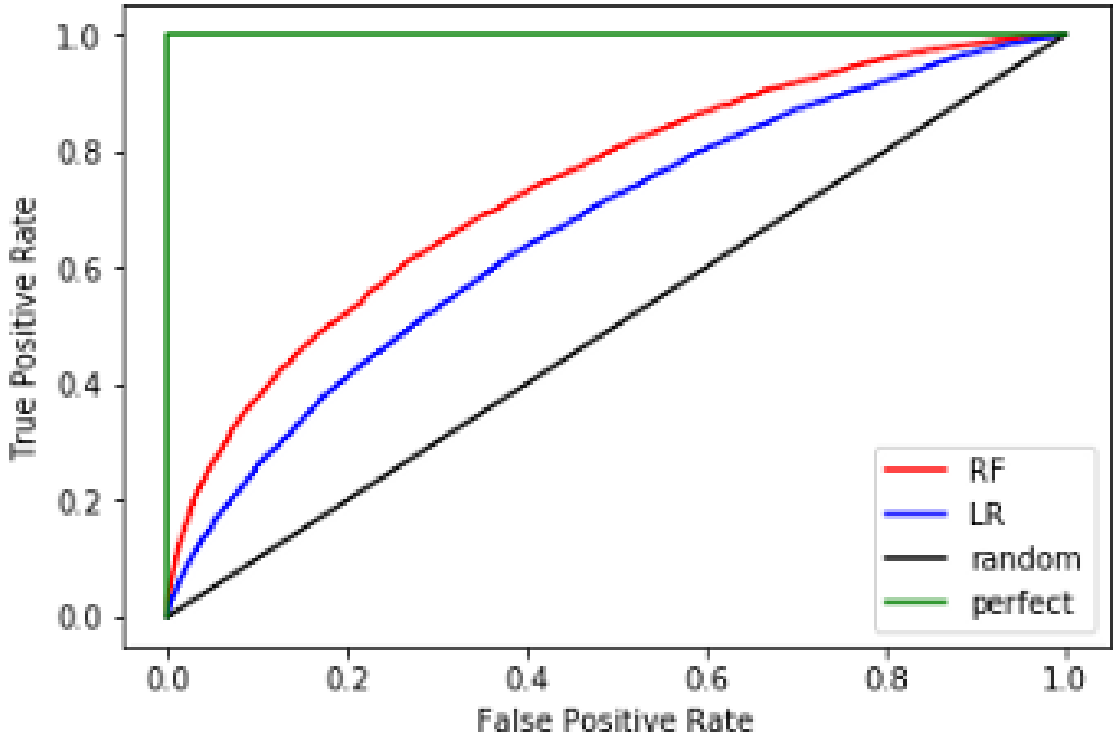


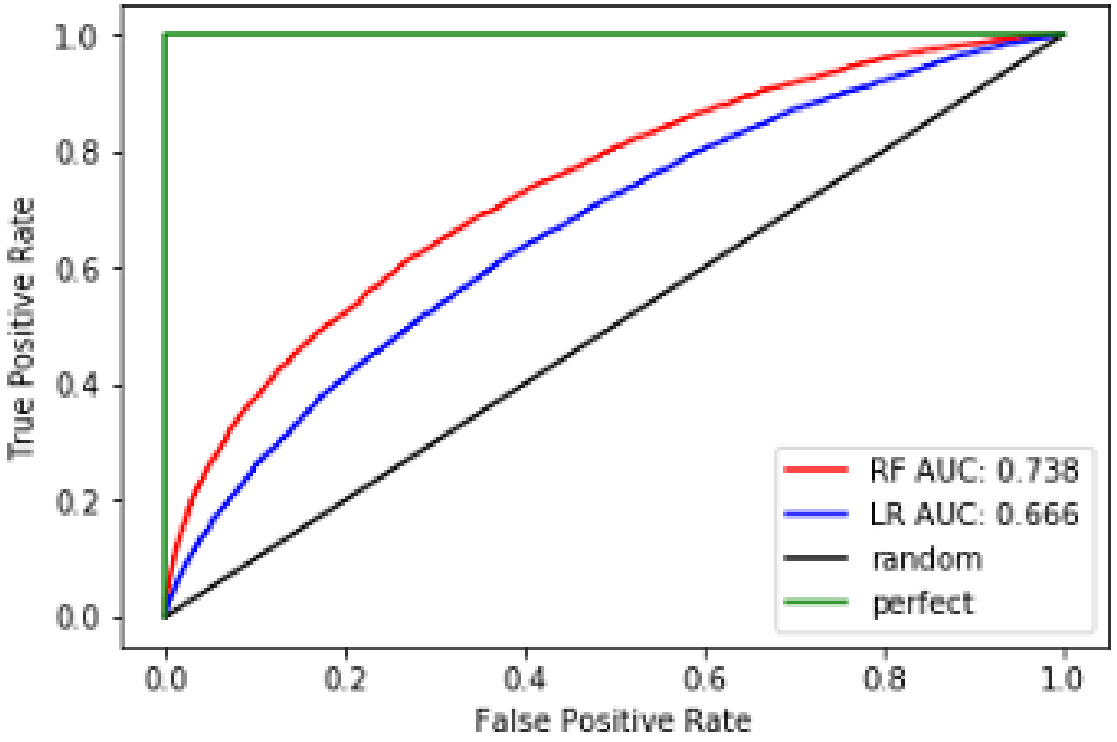
ROC Curve and ROC AUC Score



- ROC - **Receiver Operating Characteristic**
- ROC curves are VERY help with understanding the balance between true-positive rate and false positive rates.
- thresholds = all unique prediction probabilities in descending order
- tpr = the true positive rate (TP / (TP + FN)) for each threshold
- fpr = the false positive rate (FP / (FP + TN)) for each threshold

- https://en.wikipedia.org/wiki/Receiver_operating_characteristic





ROC Curve and ROC AUC Score

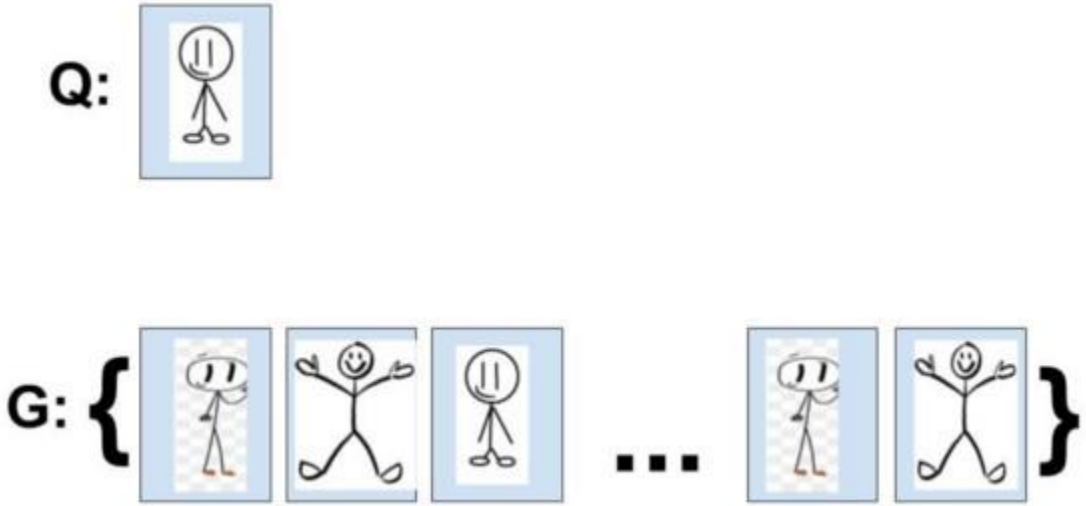
Precision and Recall for Information Retrieval

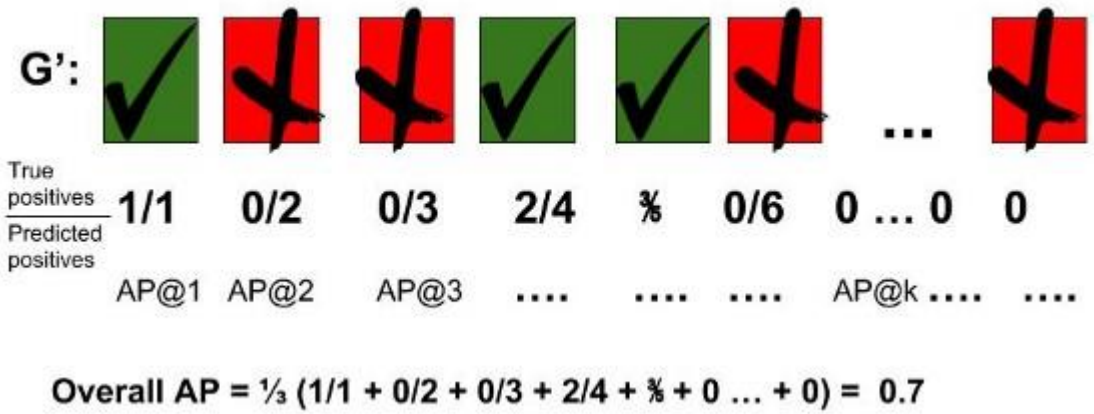
- Google Search
- Image Retrieval
- Video Retrieval
- Content based image/video Retrieval

Average Precision

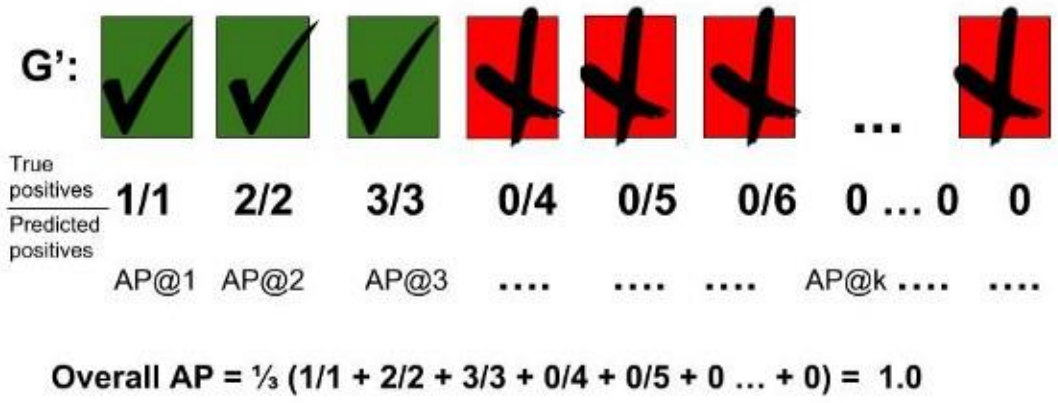
$$\text{AP@k} = \frac{1}{\text{GTP}} \sum_{i=1}^{k} \frac{\text{TP seen}}{i}$$

AP@k formula for information retrieval tasks





Calculation of AP for a given query, Q, with a GTP=3



Calculation of a perfect AP for a given query, Q, with a GTP=3

Mean Average Precision (mAP)

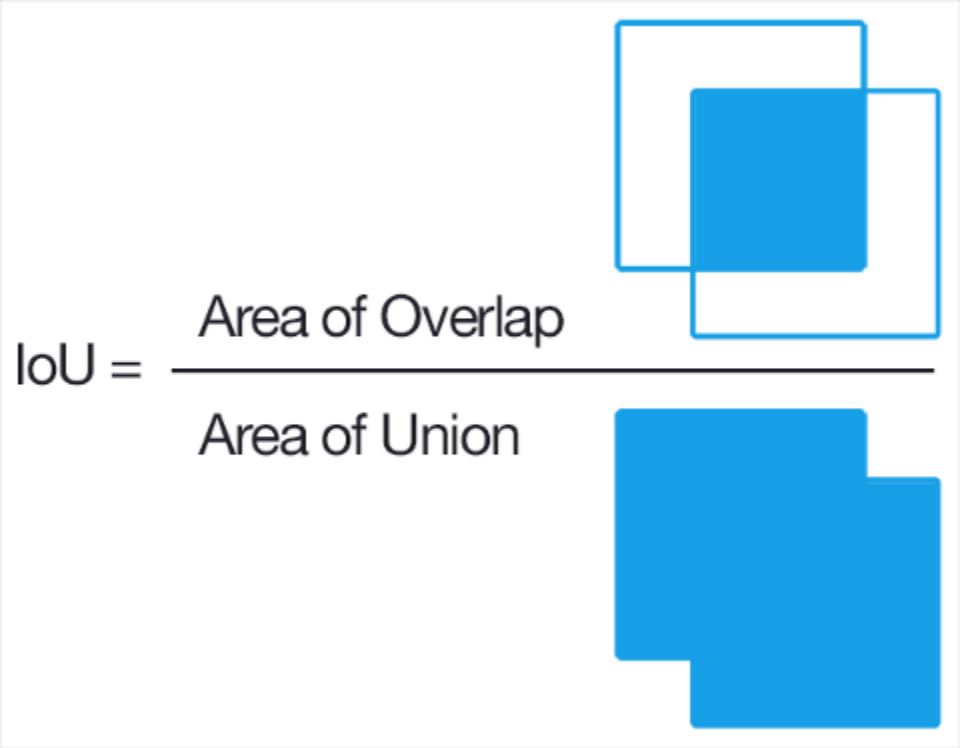
- For each query, Q, we can calculate a corresponding AP. A user can have as much queries as he/she likes against any labeled database. The mAP is simply the mean of all the queries that the use made.

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i$$

mAP formula for information retrieval

Intersection over Union (IoU)

- Useful in object detection and image segmentation problems



Regression Metrics

3.3.4.3. Mean absolute error

The `mean_absolute_error` function computes mean absolute error, a risk metric corresponding to the expected value of the absolute error loss or $l1$-norm loss.

If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean absolute error (MAE) estimated over n_{samples} is defined as

$$\text{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \left| y_i - \hat{y}_i \right| .$$

Treats all the errors equally

3.3.4.6. Median absolute error

The `median_absolute_error` is particularly interesting because it is robust to outliers. The loss is calculated by taking the median of all absolute differences between the target and the prediction.

If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, then the median absolute error (MedAE) estimated over n_{samples} is defined as

$$\text{MedAE}(y, \hat{y}) = \text{median}(\mid y_1 - \hat{y}_1 \mid, \ldots, \mid y_n - \hat{y}_n \mid).$$

The `median_absolute_error` does not support multioutput.

Prevents outliers contributing more error to the model evaluation

3.3.4.4. Mean squared error

The `mean_squared_error` function computes mean square error, a risk metric corresponding to the expected value of the squared (quadratic) error or loss.

If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over n_{samples} is defined as

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

Squaring always gives a positive value, so the sum will not be zero.

Squaring emphasizes larger differences—a feature that turns out to be both good and bad (think of the effect outliers have).

3.3.4.5. Mean squared logarithmic error

The `mean_squared_log_error` function computes a risk metric corresponding to the expected value of the squared logarithmic (quadratic) error or loss.

If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared logarithmic error (MSLE) estimated over n_{samples} is defined as

$$\text{MSLE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (\log_e(1+y_i) - \log_e(1+\hat{y}_i))^2.$$

Where $\log_e(x)$ means the natural logarithm of x. This metric is best to use when targets having exponential growth, such as population counts, average sales of a commodity over a span of years etc. Note that this metric penalizes an under-predicted estimate greater than an over-predicted estimate.

Useful when scale of prediction is too high, where log scales it down

3.3.4.2. Max error

The `max_error` function computes the maximum residual error , a metric that captures the worst case error between the predicted value and the true value. In a perfectly fitted single output regression model, `max_error` would be `0` on the training set and though this would be highly unlikely in the real world, this metric shows the extent of error that the model had when it was fitted.

If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the max error is defined as

$$\text{Max Error}(y, \hat{y}) = max(|y_i - \hat{y}_i|)$$

Aims to find what is the maximum error made by the model for a single sample... Finding a best model that covers all the samples

3.3.4.1. Explained variance score

$$s^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - \hat{\mu})^2$$

The `explained_variance_score` computes the explained variance regression score.

If $\hat{y}$ is the estimated target output, y the corresponding (correct) target output, and Var is Variance, the square of the standard deviation, then the explained variance is estimated as follow:

$$explained\_variance(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}$$

The best possible score is 1.0, lower values are worse.

Estimates the deviation between prediction and actual values

3.3.4.7. R^2 score, the coefficient of determination

The `r2_score` function computes the coefficient of determination, usually denoted as R^2.

It represents the proportion of variance (of y) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance.

As such variance is dataset dependent, R^2 may not be meaningfully comparable across different datasets. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value for total n samples, the estimated R^2 is defined as:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Very Popular one!

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2$.

Note that `r2_score` calculates unadjusted R^2 without correcting for bias in sample variance of y.