Universität Basel

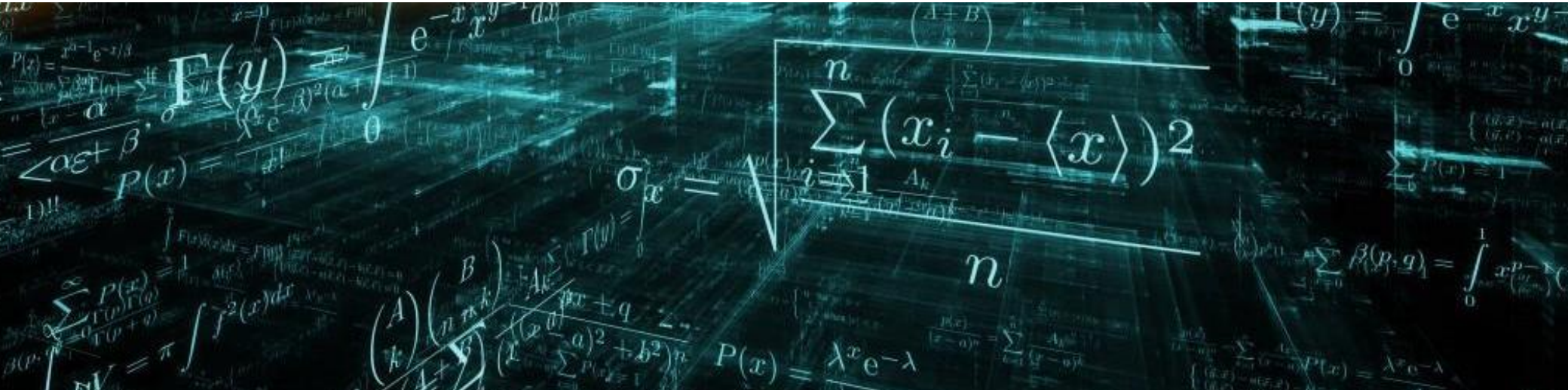
Computer Science / 15731-01 / 2024

Multimedia Retrieval

Chapter 2: Performance Evaluation

2.1 Introduction
2.2 Boolean Retrieval
2.3 Retrieval with Order
2.4 Retrieval with Graded Relevance
2.5 The Confusion Matrix
2.6 Optimizing Hyperparameters
2.7 Literature and Links

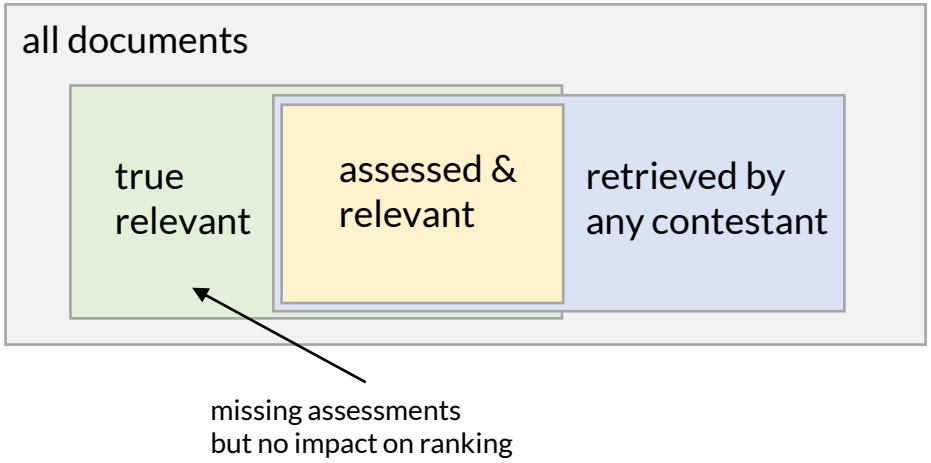
Dr. Roger Weber, roger.weber@gmail.com



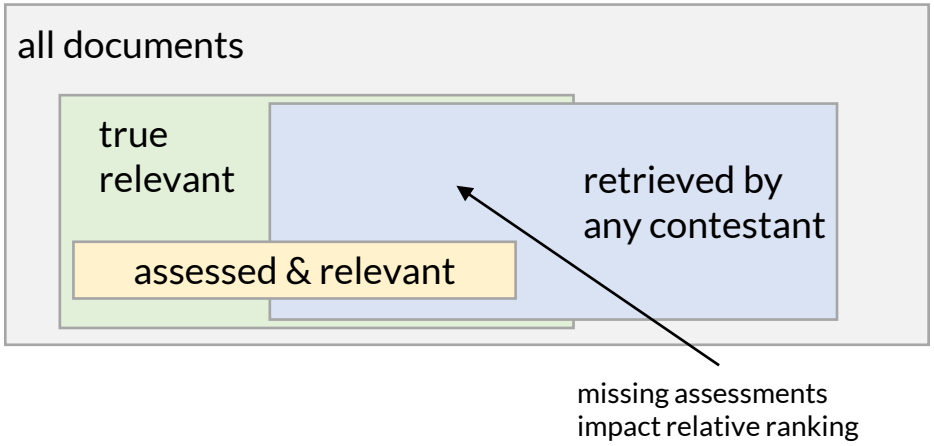
2.1 Introduction

- So far, we have not discussed the evaluation of classification and retrieval methods. However, both use similar metrics such as precision and recall. Precision measures the percentage of relevant or correctly labeled items among those retrieved or with the same label, while recall calculates the percentage of retrieved or correctly classified items out of the set of relevant items or items with the same label.
- Before we define these metrics, let's consider what we need for a retrieval benchmark. Firstly, we need a collection of documents that match the retrieval scenario. Secondly, we require multiple queries covering various aspects of the retrieval task, along with a relevance assessment of documents against these queries. Lastly, we need a performance goal that the algorithms should achieve.
  - **Collections and queries:** examples include MS MARCO (Microsoft Machine Reading Comprehension Dataset), which contains over 500 thousand queries from Bing against millions of retrieved documents and passages. Another example is the TREC (Text REtrieval Conference) data sets, featuring 50+ queries against several thousand documents.
  - **Assessment:** in the TREC data set, each query is assessed against the collection of retrieved documents (only the ones returned by competing systems). Even though we do not assess all documents for each query, we obtain a relatively dense assessment. On the other hand, assessments for MS MARCO are sparse, with only a few documents assessed against the 500 thousand queries to keep the assessment efforts low. The impact of these different assessment approaches will be visualized on the next page.
  - **Performance goal:** in web retrieval, users typically focus on the top result or the first 10 documents. The performance goal is to have a relevant document at the top of the ranking. On the other hand, a patent lawyer or a researcher aims to retrieve as many relevant documents as possible with only a few non-relevant items. They want more documents and at the same time reduce the fall-out, i.e., returned documents that turn out to be not relevant and thus incur overhead going through the results.

- In a typical text retrieval competition, each contestant evaluates all queries against the collection. The competition assess the combined set of documents retrieved by all contestants (often with the help of the contestants), leading to a dense assessment, as shown on the right side. Even though not all documents are assessed, this approach maintains the relative order of competing algorithms. To illustrate, if we have a relevant document that was not assessed (see arrow), it may lead to a slight overestimation of the algorithms' ability to retrieve relevant documents (recall). However, including assessments for these documents would not alter the relative ranking of the methods.



- Competitions like MS MARCO have a large number of queries, making dense assessments impractical. Instead, they only assess a few documents per query (sometimes just 2-5 documents), leading to a sparse assessment, as shown on the right side. This significantly differs from the dense assessment above. For instance, there could be assessed and relevant documents that none of the contestants found. However, the challenge arises from missing assessment for retrieved documents, which can negatively impact the performance evaluation. For example, consider the area highlighted by the arrow: it contains relevant documents for which assessments are missing. Consequently, even if a competing algorithm provides a good answer, due to missing relevancy assessment, it may not receive credit for it.



2.2 Boolean Retrieval

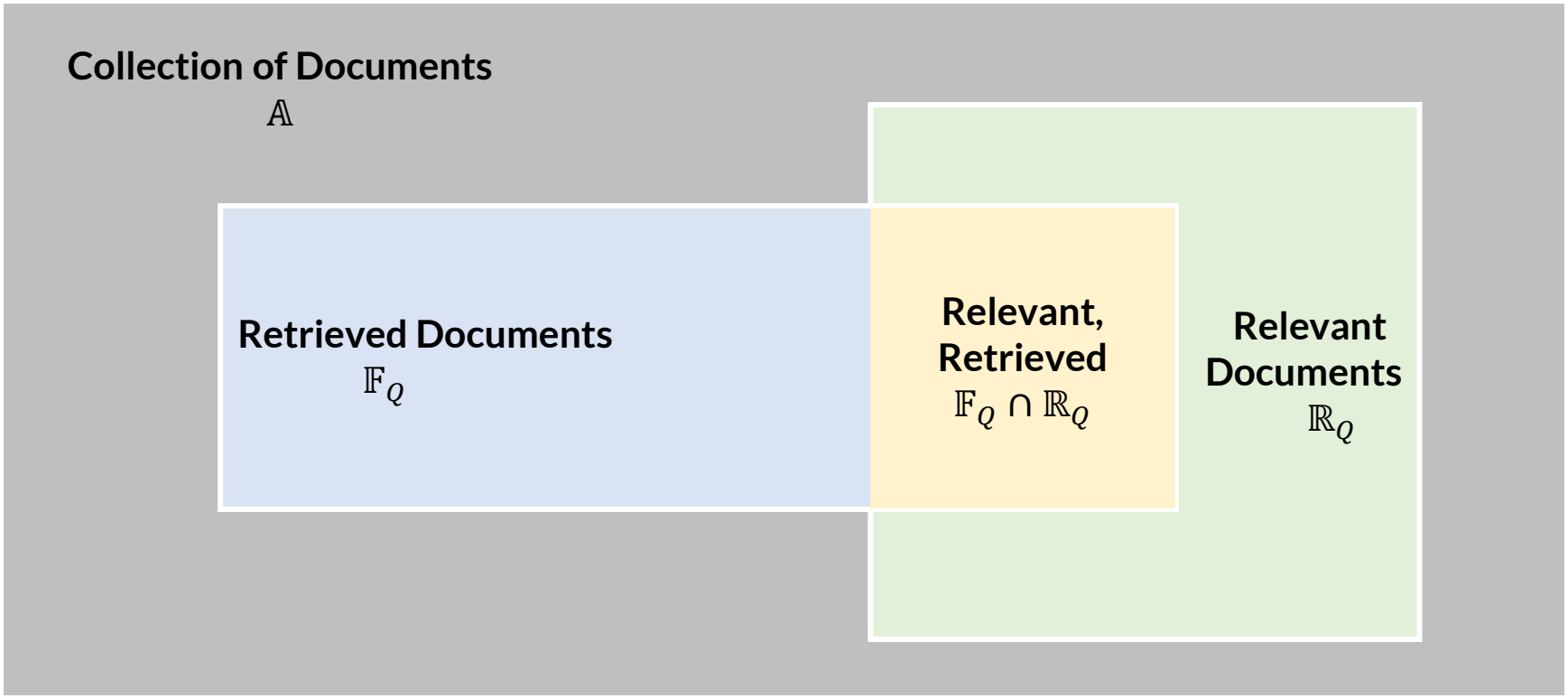
- First, let's examine the case of Boolean retrieval. In this basic scenario, documents are retrieved as sets without any specific order among them. In other words, the retrieval system provides a list of documents but we do not consider the order in which the documents are presented when assessing the system. Later, we will explore how to expand this approach and consider the order of documents in our evaluation.
- **Precision** and **recall** are the earliest and still most important measures used for the evaluation of search algorithms. Precision denotes how many of the answers are relevant from a user's perspective. Recall describes the percentage of retrieved and relevant answers over all relevant documents in the collection. They form the key dimensions that covers the user's interests:
  - Precision is the measure of having relevant documents easily accessible without the need to go through many documents in the result set. In many cases, we are not interested in all relevant documents, but only a few that give us the necessary information. A good search engine that offers only relevant documents is suitable for many knowledge-based or fact-checking queries (e.g., students, fact-checkers).
  - Recall is about exploring most of the relevant documents. In such cases, we want a comprehensive overview of the relevant documents and aim to minimize the number of non-relevant documents (fall-out, false positives). A good search engine should retrieve most or all of the relevant documents while avoiding too many non-relevant ones in the answer (e.g., patent lawyers, searches with vague criteria).
- Notations:

$\mathbb{A}$	Set of all documents
$\mathbb{R}_Q$	Set of relevant documents for a query Q in the collection $\mathbb{A}$
$\mathbb{F}_Q$	Set of documents retrieved by a system for query Q

- Precision p, recall r, and fallout f are then defined as (see visualization on next page):

$$p = \frac{|\mathbb{F}_Q \cap \mathbb{R}_Q|}{|\mathbb{F}_Q|} \qquad r = \frac{|\mathbb{F}_Q \cap \mathbb{R}_Q|}{|\mathbb{R}_Q|} \qquad f = \frac{|\mathbb{F}_Q \setminus \mathbb{R}_Q|}{|\mathbb{A} \setminus \mathbb{R}_Q|}$$

- Visualization or precision and recall



Precision: $p = \frac{|\mathbb{F}_Q \cap \mathbb{R}_Q|}{|\mathbb{F}_Q|}$ Recall: $r = \frac{|\mathbb{F}_Q \cap \mathbb{R}_Q|}{|\mathbb{R}_Q|}$ Fallout: $f = \frac{|\mathbb{F}_Q \setminus \mathbb{R}_Q|}{|\mathbb{A} \setminus \mathbb{R}_Q|}$

– The **F-Measure** combines precision and recall into a single value, simplifying the comparison of different retrieval methods. The parameter β determines the importance of recall over precision. When β=0, only precision is considered; when β=∞, only recall is considered.

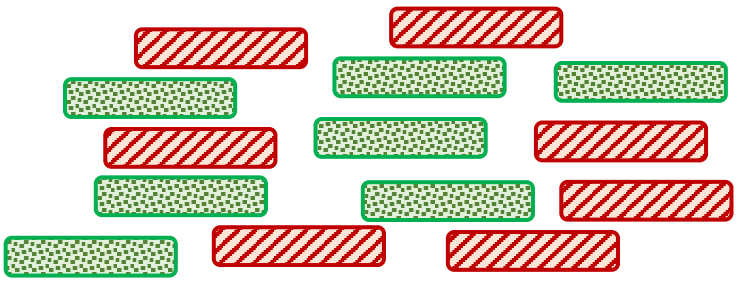
$$F_\beta = \frac{(\beta^2 + 1) \cdot p \cdot r}{\beta^2 \cdot p + r}$$

$$F_1 = \frac{2 \cdot p \cdot r}{p + r} = \text{F-score}$$

The $\boldsymbol{F_1}$**-score** is a common choice with β=1 and is also frequently used in machine learning tasks to optimize hyperparameters (see later in this chapter). Generally, β should be selected thoughtfully depending on the retrieval task's performance goal. For example, for fact-checking tasks, precision is prioritized over recall, making a smaller β=0.5 suitable. On the other hand, a patent lawyer may choose β=2 to emphasize the importance of retrieving many relevant documents while maintaining reasonable precision.

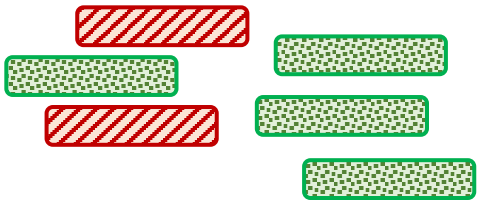
- Example: Comparing two methods (query has a total of 20 relevant documents)

🔍 Search Method A



$p = \frac{7}{14} = 50\%$ $r = \frac{7}{20} = 35\%$ $F_1 = \frac{2 \cdot \frac{7}{14} \cdot \frac{7}{20}}{\frac{7}{14} + \frac{7}{20}} = 0.41$

🔍 Search Method B



$p = \frac{4}{6} = 67\%$ $r = \frac{4}{20} = 20\%$ $F_1 = \frac{2 \cdot \frac{4}{6} \cdot \frac{4}{20}}{\frac{4}{6} + \frac{4}{20}} = 0.31$

relevant document non-relevant document

- Typically, we have multiple queries, and for each query, we calculate a precision-recall pair. To evaluate the overall retrieval performance, we need to compute average precision and recall. Let N represent the number of queries. For each query Q_i, we have a set $\mathbb{F}_i$ retrieved by the search method and a set $\mathbb{R}_i$ of relevant documents for that query. We use p_i and r_i to denote the precision and recall values, respectively, as explained earlier.
- With **macro evaluation**, we simply compute the average over all precision and recall values as follows:

$$p = \frac{1}{N}\sum_{i=1}^{N} p_i = \frac{1}{N}\sum_{i=1}^{N} \frac{|\mathbb{F}_i \cap \mathbb{R}_i|}{|\mathbb{F}_i|} \qquad r = \frac{1}{N}\sum_{i=1}^{N} r_i = \frac{1}{N}\sum_{i=1}^{N} \frac{|\mathbb{F}_i \cap \mathbb{R}_i|}{|\mathbb{R}_i|}$$

  While the macro evaluation method is generally effective, it can have limitations when dealing with varying sizes of relevant documents for queries. For example, consider a query Q_i that has only one relevant document in the entire collection while the other queries have hundreds of relevant documents. Not finding that relevant document for Q_i would result in a precision value $p_i = 0$. This can significantly lower the average precision, even if the method performs well and produces high precision values for the other queries.

- **Micro evaluation** overcomes this issue by summing the true positives and the retrieved/relevant documents before calculating the average precision and recall. This ensures a fair evaluation regardless of the set sizes of queries:

$$p = \frac{\sum_{i=1}^{N} |\mathbb{F}_i \cap \mathbb{R}_i|}{\sum_{i=1}^{N} |\mathbb{F}_i|} \qquad r = \frac{\sum_{i=1}^{N} |\mathbb{F}_i \cap \mathbb{R}_i|}{\sum_{i=1}^{N} |\mathbb{R}_i|}$$

  With the example from above, the impact on missing out on the relevant document Q_i is now much smaller and may better suit the retrieval benchmark's design

- The choice between micro and macro evaluation depends on the nature of the retrieval task and the importance given to different queries. Micro evaluation tends to emphasize the performance on queries with more relevant documents since they contribute more to the overall counts. Macro evaluation, on the other hand, gives equal weight to each query, regardless of its size or importance, providing a more balanced view of the overall performance across all queries.

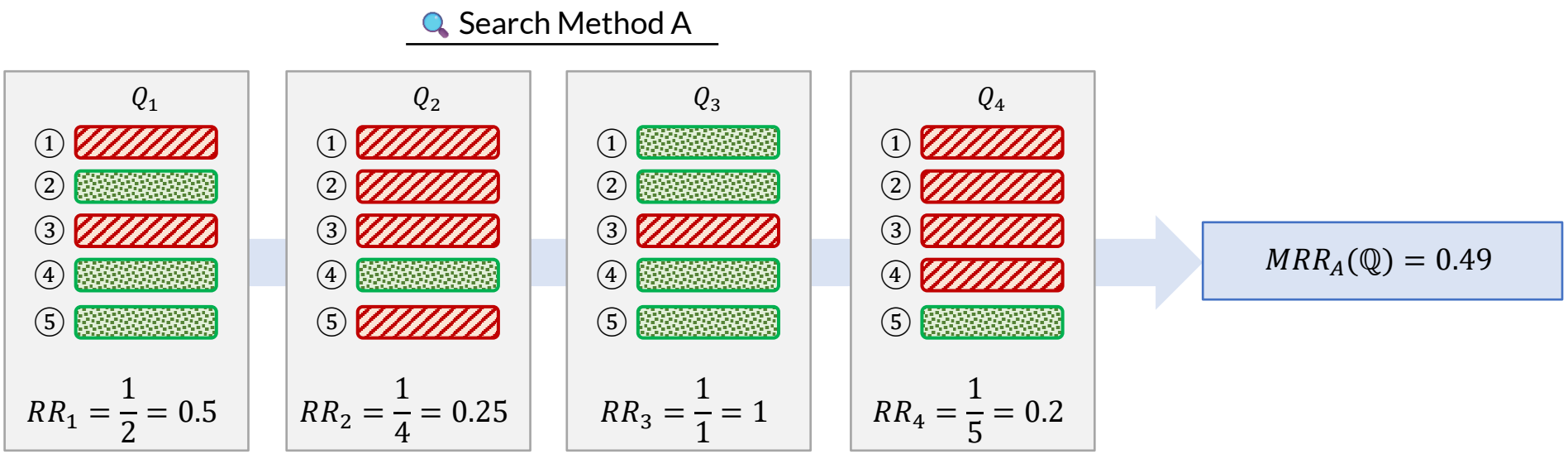
2.3 Retrieval with Order

- In many retrieval scenarios, the order of results is crucial. For example, for web search engines or fact-checking tasks, users expect the answer to appear among the top results. In such cases, the **Mean Reciprocal Rank (MRR)** is the preferred metric. MRR is especially useful when dealing with benchmarks that have sparse assessments as it prevents meaningful calculations of precision and recall values. The definition for queries $Q_i \in \mathbb{Q}$ is as follows:

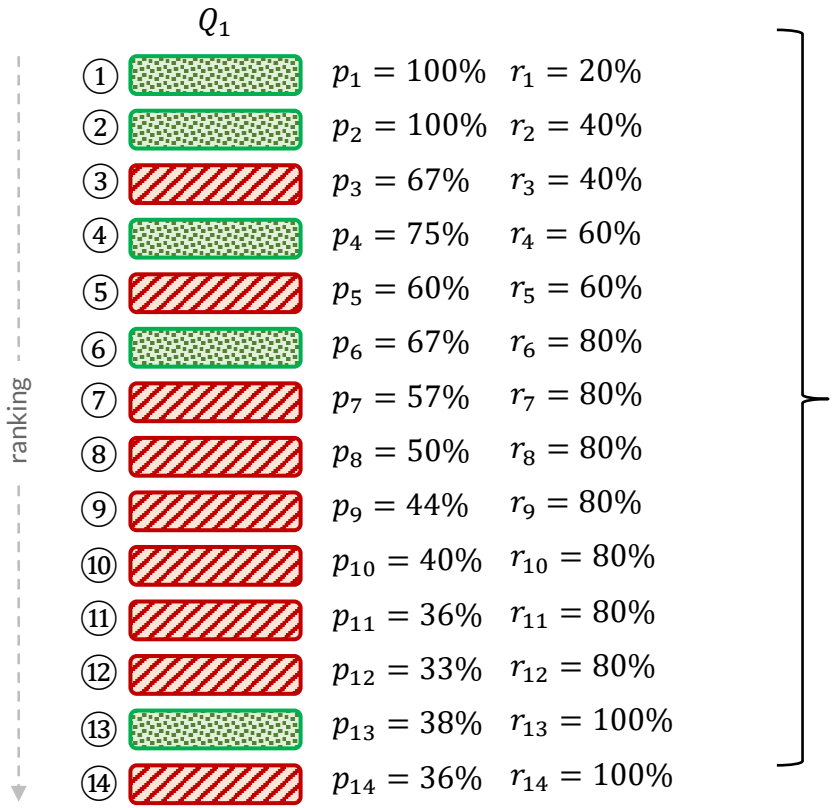
$$MRR(\mathbb{Q}) = \frac{1}{|\mathbb{Q}|} \sum_{Q_i \in \mathbb{Q}} \frac{1}{rank_i}$$

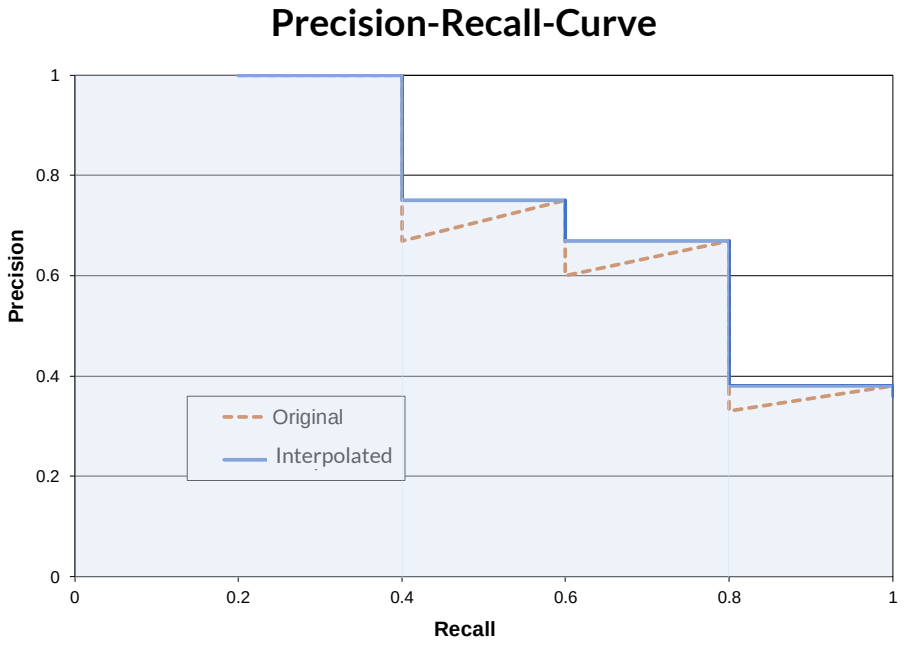
  with $rank_i$ being the rank of the first relevant document for query Q_i. Unlike precision and recall, MRR only focuses on the first relevant document and ignores the rest. It places a high importance on the top position in the ranking and assigns lower importance to later positions, converging quickly to 0 as the rank increases.

- Example: consider the search method A and its results for queries Q_1 to Q_4, as shown in the visualization below. The Mean Reciprocal Rank (MRR) is calculated as the average of the reciprocal ranks RR_i for all the queries Q_i.

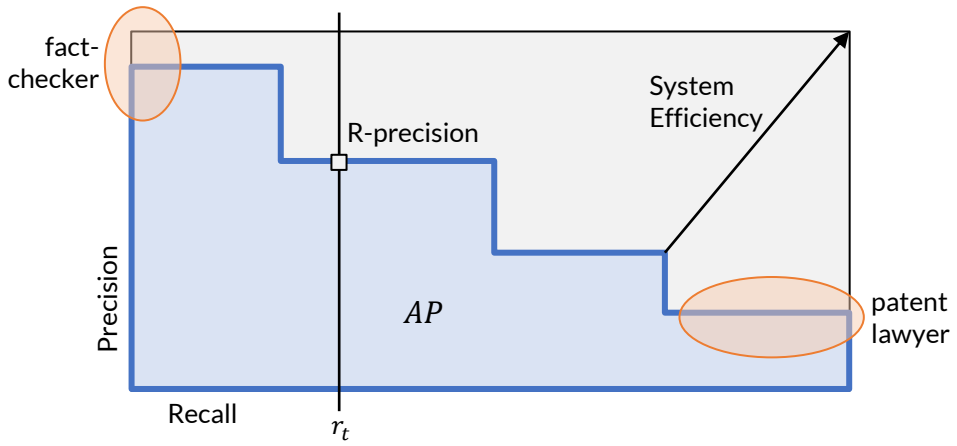


- We can extend the definition of precision and recall to include the ranking of retrieved documents. The **Precision-Recall Curve** only considers the top-k results in the ranking for $k = 1..n$ (as shown in the picture below on the left side). For each top-k result, it calculates the precision and recall values based on their relevance assessment. In the example below with $k = 3$, the precision is $p_3 = 2/3$, as 2 out of 3 documents in the top-3 are relevant. If we have 5 relevant documents overall in the collection, then the recall is $r_3 = 2/5$, as the top-3 contains 2 out of 5 relevant documents. We can calculate all the other precision p_i and recall r_i values, forming the precision-recall curve as depicted on the right side below.
  - As we increase k, the precision increases when the next document is relevant and decreases if it is not relevant. On the other hand, recall values only increase whenever we find a new relevant document. This results in a characteristic "sawtooth" plot, which is often interpolated to simplify subsequent calculations such as the area under the precision-recall curve (blue area in the picture on the right side)





- Note that we can always achieve a recall value of 1 if we keep enumerating documents in the list until all relevant ones have been returned. Once we have all the relevant documents, the PR-curve becomes complete, and we can use it directly for visual comparisons between two methods or calculate simpler metrics for the comparison.
  - Near the point where recall is 0 and precision is 1, the PR-curve shows how well a method can answer fact-checker type queries where high precision for the top-10 documents is expected but recall does not really matter.
  - Near recall values of 1, the PR-curve identifies methods that can find most of the relevant documents in the collection. High precision values are preferred as they indicate low overhead when going through the result list.
  - The ideal system would achieve a precision and recall value of 1. The **system efficiency** is measured by calculating the distance d of the PR-curve to this ideal point. The system efficiency E is then given by $E = 1 - d/\sqrt{2}$.
  - The **precision at k (P@k)** is a commonly used measure, calculated as the precision p_k of the top-k results. It is often used when we are not interested in all relevant documents and, thus, do not consider recall values.
  - Similarly, the **R-precision** measures the precision once a threshold recall value r_t is reached. For example, with $r_t = 20\%$, the metric evaluates the precision once 1/5th of the relevant documents were found. This method requires knowing the total number of relevant documents in the collection.
  - Finally, the **average precision (AP)** measures the area under the PR-curve (blue area, formula on the right side). High AP values indicate that a method maintains high precision as more and more relevant documents are found.
  - By iterating over a set of queries $\mathbb{Q}$, we can easily calculate the mean values for all the measures introduced above. The formula on the lower right side shows an example for the **mean average precision (MAP)**.



$$AP = \int_0^1 p(r)dr = \sum_{k=1}^{n} p_k(r_k - r_{k-1})$$

$$MAP(\mathbb{Q}) = \frac{1}{|\mathbb{Q}|} \sum_{Q_i \in \mathbb{Q}} AP(Q_i)$$

2.4 Retrieval with Graded Relevance

- Until now, we have only considered binary relevance assessments for documents. However, we can also use graded relevance assessment where the grade indicates varying degrees of match between the document and the query. For example, we can introduce relevance values between 0 and 3, where 0 means "not relevant," 1 means "somewhat relevant," 2 means "relevant," and 3 means "highly relevant." The grades now influence how we assess a search method: higher grades are preferred over lower grades at the top of the rankings.
- The **cumulative gain (CG-k)** is a measure of how valuable the top-k results are, similar to precision at k:

$$CG_k = \sum_{i=1}^{k} rel_i \qquad \widehat{CG}_k = \frac{\sum_{i=1}^{k} rel_i}{k \cdot rel_{max}} \qquad rel_i \in [0, rel_{max}]$$

  with rel_i denoting the graded relevance of the document at position i. To obtain a value between 0 and 1, CG_i needs to be normalized with $k \cdot rel_{max}$. In the case of binary relevance assessments, $\widehat{CG}_k$ is equal to precision at k.

- The **discounted cumulative gain (DCG-k)** incorporates the ranking by penalizing relevant documents in lower ranks:

$$DCG_k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} \qquad \text{variant:} \quad DCG'_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

  The variant on the right emphasizes relevant documents more strongly than the left formula. Other variants may use different logarithm bases and slight modifications of this approach.

- Search results for a given query may have varying lengths, making it challenging to interpret DCG-k values and compare them across queries. To address this, we compute a **normalized DCG (nDCG-k)** by first establishing an ideal ranking where documents are sorted by their graded relevance in descending order, and then calculating the DCG-k value for this ideal ranking. This yields the highest possible value, known as the ideal DCG-k (IDCG-k), which we use to normalize the DCG value of the actual result.

$$nDCG_k = \frac{DCG_k}{IDCG_k} \qquad \text{with } IDCG_k = \max DCG_k \quad \text{(over all possible rankings of documents)}$$

- Example: let's consider the first 10 documents of a search result with graded relevance values between 0 and 3

k	rel_k	CG_k	$\widehat{CG}_k$	$\frac{1}{\log_2(k+1)}$	DCG_k	ideal rel_k	$IDCG_k$	$nDCG_k$
1	0	0	0.00	1.00	0.00	3	3.00	0.00
2	2	2	0.33	0.63	1.26	3	4.89	0.26
3	1	3	0.33	0.50	1.76	3	6.39	0.28
4	3	6	0.50	0.43	3.05	3	7.68	0.40
5	0	6	0.40	0.38	3.05	3	8.85	0.35
6	2	8	0.44	0.36	3.77	2	9.56	0.39
7	0	8	0.38	0.33	3.77	2	10.22	0.37
8	3	11	0.46	0.32	4.71	2	10.86	0.43
9	1	12	0.44	0.30	5.01	2	11.46	0.44
10	3	15	0.50	0.29	5.88	2	12.04	0.49

  - The table above displays 10 results, with the 2nd column indicating the graded relevance rel_k for each document at position k. The cumulative gain CG_k is the sum of these values up to position k. To obtain the normalized version, we divide CG_k by $3 \cdot k$. Consequently, we obtain $\widehat{CG}_{10} = 0.5$. For comparison, if we consider any $rel_k > 0$ as relevant, then the "precision at 10" is 0.70 overrating the documents with low relevance grades.
  - The discounted cumulative gain DCG_k is shown in the next columns: first, we have the discount factors for each ranking position. By multiplying them with the graded relevance rel_k and summing them up to position k, we obtain the DCG_k values. Since we have not normalized the relevance values, they are difficult to interpret.
  - For the normalized DCG, we assume there are a total of 5 documents with a graded relevance of 3 and 10 documents with a graded relevance of 2 in the collection. The "ideal rel_k" column shows an ideal ranking for this scenario, allowing us to compute the ideal DCG ($IDCG_k$) values and use them to obtain the normalized DCG ($nDCG_k$) values in the rightmost column. An $nDCG_{10}$ value of 0.49 illustrates solid performance in this example.

2.5 The Confusion Matrix

- In the introduction, we covered performance measures like mean squared error and cross-entropy loss in machine learning. Now, we delve into the performance of classification methods, focusing on both binary and multi-class tasks with hard assignments.
- The confusion matrix is a common approach that presents correct and incorrect classifications in a tabular form, enabling the extraction of various metrics to assess the performance of a method. Rows represent the predicted conditions, also known as test results, while columns denote the observed actual conditions, also known as ground truth (the labels in the data sets). Let's examine the table below:
  - The term "condition" is a general description of the task's output value. For instance, it could be "it will rain," "it's a dog," "patient has the disease," "the object belongs to the class," or "the student passes the exam."
  - The "True" row contains all data items for which the method predicts that the item fulfills the condition. Let's compare these predictions with the actual values: first, we have the **True Positives (TP)** where the prediction is correct (matches the observation). Second, we have the **False Positives (FP)** where the prediction is wrong, and the method overestimates the condition.
  - The "False" row contains all data items for which the method predicts that the item does not fulfill the condition. If we compare these outcomes with the actual values, we observe the **True Negatives (TN)** for which the prediction is correct, and the **False Negatives (FN)** for which the prediction is wrong. In the latter case, the method is underestimating the condition.

		Actual Condition (as observed)	
	Population	Positive (P)	Negative (N)
Predicted Condition (as computed)	True (T)	**True Positive** (TP)	**False Positive** (FP)
	False (F)	**False Negative** (FN)	**True Negative** (TN)

There is also “confusion” regarding the placement of actual values in the confusion matrix. Earlier papers display the confusion matrix with actual values in the columns (like here on the left side), while more recent publications and popular software packages use the transformed notation, placing the actual values as rows. This does not change the interpretation of the confusion matrix but makes it difficult to read the table if it appears in the unfamiliar form.

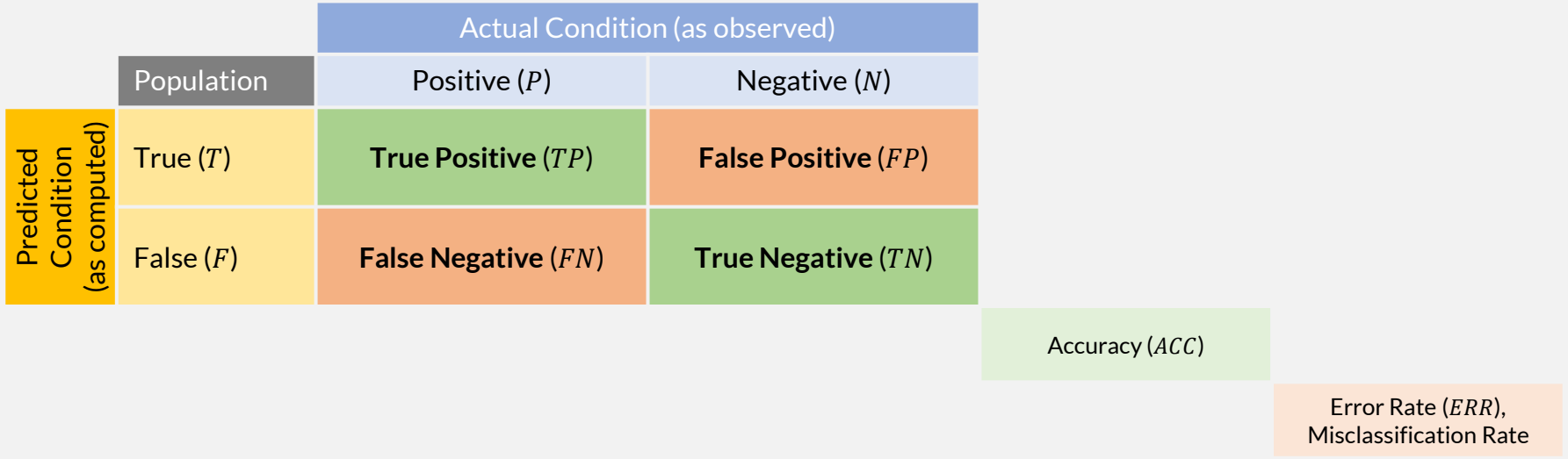
- Based on this basic table, we can derive further metrics. To this end, let us introduce the following notation:
  - Based on the ground truth, P and N represent the number of positive and negative items, respectively. $P + N$ is the total size of the dataset. In the 2x2 confusion matrix, the values in the columns sum up to P and N.
  - Based on the predicted values, T and F represent the number of "true" and "false" outputs, respectively. $T + F$ is the total size of the dataset. In the 2x2 confusion matrix, the values in the rows sum up to T and F.
  - The values in the cells correspond to the notion introduced on the previous page: True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN).
- Let's first consider the rows in more detail:
  - The **Positive Predictive Value (PPV)**, or **Precision**, is calculated as the ratio TP/T. It represents the proportion of correctly predicted positive items. In the context of a disease test, it indicates the percentage of people with positive ('true') test results who are actually sick. A low PPV value means that the method would wrongly diagnose a disease from which the patient does not actually suffer. The **False Discovery Rate (FDR)** is the complement of PPV and is computed as $\mathrm{FP}/T = 1 - PPV$.
  - The **Negative Predicative Value (NPV)** is calculated as the ratio TN/F. It represents the proportion of correctly predicted negative items. In the context of a disease test, it indicates how well a method can exclude a disease during diagnostics of symptoms. A low NPV value means that the method misses many sick people. The **False Omission Rate (FOR)** is the complement of NPV and is computed as $FN/F = 1 - NPV$.

		Actual Condition (as observed)			
	Population	Positive (P)	Negative (N)		
Predicted Condition (as computed)	True (T)	**True Positive** (TP)	**False Positive** (FP)	Positive Predictive Value (PPV), Precision	False Discovery Rate (FDR)
	False (F)	**False Negative** (FN)	**True Negative** (TN)	False Omission Rate (FOR)	Negative Predictive Value (NPV)

- Next, we look at the columns for further insights:
  - The **True Positive Rate (TPR)**, also known as **Recall**, is calculated as TP/P. It represents the proportion of correctly predicted items among all positive cases. This measure is often referred to as **Sensitivity**, for example in the context of a disease test, as it indicates the percentage of actual sick people the test can detect. A high sensitivity in a disease test effectively "rules out" the disease in negative predictions, as it rarely misdiagnoses those who have the disease. However, a high sensitivity does not necessarily indicate the ability to "rule in" the disease. For example, a fake test that always returns positive results will have a sensitivity of 100%, but it is not effective. The **False Negative Rate (FNR)** is the complement of TPR and is computed as $FN/P = 1 - TPR$.
  - The **True Negative Rate (TNR)**, also known as **Specificity**, is calculated as TN/N. It represents the proportion of correctly predicted items among all negative cases. This measure, for example in the context of a disease test, indicates the percentage of healthy people the test correctly classifies as "not sick" (false). A high specificity in a disease test effectively "rules in" the disease in positive predictions, as it rarely diagnoses the disease for healthy people. However, a high specificity does not necessarily indicate the ability to "rule out" the disease. For example, a fake test that always returns negative results will have a specificity of 100% but it is not effective. The **False Positive Rate (FPR)**, or **Fall-Out**, is the complement of TNR and is computed as $FP/N = 1 - TNR$.

		Actual Condition (as observed)	
	Population	Positive (P)	Negative (N)
Predicted Condition (as computed)	True (T)	**True Positive** (TP)	**False Positive** (FP)
	False (F)	**False Negative** (FN)	**True Negative** (TN)
		True Positive Rate (TPR), Sensitivity, Recall, Hit Rate	False Positive Rate (FPR), Fall-Out
		False Negative Rate (FNR), Miss Rate	True Negative Rate (TNR), Specificity

- Finally, we consider the diagonals of the confusion matrix:
  - **Accuracy (ACC)** is calculated as $(TP + TN)/(P + N)$. It represents the percentage of correctly predicted items and has become a standard measure for many classification tasks in machine learning. However, high accuracy alone is not always a good indicator, as we will discuss later with an example.
  - The **Error Rate (ERR)**, or **Misclassification Rate**, is the complement of the accuracy and measures the percentage of wrongly predicted items. It is calculated as $(FP + FN)/(P + N) = 1 - ACC$.
- The literature has produced many more measures around the confusion matrix. A few examples include:
  - The **Prevalence** is the ratio of P over the total population. In a balanced scenario where positive and negative cases are about equally frequent, P is close to 0.5. An extreme value for P (<0.1, >0.9) often suggests revisiting the applicability of some of the metrics, as we will see in examples later on.
  - The F_1-score, as we introduced earlier in this chapter, is a harmonic mean between precision and recall. It is computed as $F_1 = 2 \cdot P \cdot R/(P + R) = 2TP/(2TP + FP + FN)$. Note how the F_1-score does not take the true negative values TN into account. The F_1-score is widely used in the natural language processing literature for tasks such as word segmentation or entity recognition.



- Overview of popular metrics based on the confusion matrix for binary classifications:
  - To help remember the formulae below better, the core 2x2 matrix of the confusion matrix results from counting the predictions and comparing them with the actual values. P, N, T, and F are the sums of their respective column and row. The cells in the 2x2 matrix to the right are calculated by taking the ratio of the value of the core cell in the same position and the row's total T or F. The rows of this matrix sum up to 1. Similarly, the cells in the 2x2 matrix below are calculated by taking the ratio of the value of the core cell in the same position and the column's total P or N. The columns of this matrix sum up to 1. Accuracy and Error Rate are the ratios of the sum of the diagonals and the total number of cases ($P + N = T + F$).

		Actual Condition (as observed)			
	Population	Positive (P)	Negative (N)		
Predicted Condition (as computed)	True (T)	**True Positive** (TP)	**False Positive** (FP)	Positive Predictive Value (PPV), Precision	False Discovery Rate (FDR)
	False (F)	**False Negative** (FN)	**True Negative** (TN)	False Omission Rate (FOR)	Negative Predictive Value (NPV)
		True Positive Rate (TPR), Sensitivity, Recall, Hit Rate	False Positive Rate (FPR), Fall-Out	Accuracy (ACC)	
		False Negative Rate (FNR), Miss Rate	True Negative Rate (TNR), Specificity		Error Rate (ERR), Misclassification Rate

$$TPR = \frac{TP}{P} \qquad FPR = \frac{FP}{N}$$

$$FNR = \frac{FN}{P} \qquad TNR = \frac{TN}{N}$$

$$TPR + FNR = 1 \qquad FPR + TNR = 1$$

$$PPV = \frac{TP}{T} \qquad FDR = \frac{FP}{T} \qquad PPV + FDR = 1$$

$$FOR = \frac{FN}{F} \qquad NPV = \frac{TN}{F} \qquad FOR + NPV = 1$$

$$ACC = \frac{TP + TN}{P + N} \qquad ERR = \frac{FP + FN}{P + N} \qquad ACC + ERR = 1$$

Example 1: Is this a good cancer test?

		Actual Condition (as observed)		
	Population (2030)	Positive ($P = 30$)	Negative ($N = 2000$)	
Predicted Condition (as computed)	True (200)	**True Positive** ($TP = 20$)	**False Positive** ($FP = 180$)	$PPV = \frac{20}{200} = 10\%$
	False (1830)	**False Negative** ($FN = 10$)	**True Negative** ($TN = 1820$)	$NPV = \frac{1820}{1830} = 99\%$
		$TPR = \frac{20}{30} = 67\%$	$TNR = \frac{1820}{2000} = 91\%$	$ACC = \frac{1840}{2030} = 91\%$

- The prevalence value of P=30/2030=1.4% for this data set indicates a highly unbalanced distribution of positive and negative cases. This strongly suggests that we examine different values before drawing conclusions.
  - We observe a high accuracy value of 91%, which seems good at first. However, on closer examination, we find that the precision (PPV) is only 10%. Out of 200 positive test results, only 20 people actually had cancer. This means that 180 people would be wrongly classified as having cancer if we follow the test results. The high accuracy is primarily due to the large number of true negatives. Interestingly, we could achieve an even higher accuracy (99%) by using a test method that always returns false, as this leads to 2000 true negatives and 0 true positives in this case.
  - The low precision already indicates that we should not rely on a positive test outcome. However, when the test is negative, it is correct in 99% of the cases (NPV). Furthermore, we observe a specificity of 91% (TNR). In other words, during the diagnostic process, we can successfully rule out cancer for 91% of the patients. If this is an affordable test, it can be used as an initial step to eliminate the possibility of this cancer type.
  - We might be concerned about the low sensitivity value of 67% (TPR, recall). Using only this test would miss one-third of the positive cases. Therefore, a doctor should consider other factors like symptoms or additional test results before reaching a conclusion. However, we would not recommend this test for a widely applied preventive campaign, as it could result in many false positives and unnecessary alerts, while still missing many positive cases.

Example 2: Can this test prevent further spreading of a contagious virus in its early stages?

		Actual Condition (as observed)		
	Pop. (100,000)	Positive ($P = 700$)	Negative ($N = 99{,}300$)	
Predicted Condition (as computed)	True (5,560)	**True Positive** ($TP = 595$)	**False Positive** ($FP = 4{,}965$)	$PPV = \frac{20}{200} = 11\%$
	False (94,440)	**False Negative** ($FN = 105$)	**True Negative** ($TN = 94{,}335$)	$NPV = \frac{1820}{1830} = 99\%$
		$TPR = \frac{20}{30} = 85\%$	$TNR = \frac{1820}{2000} = 95\%$	$ACC = \frac{1840}{2030} = 95\%$

- The prevalence value of P=700/99,300=0.7% for this dataset shows a highly unbalanced distribution of positive and negative cases. As in the previous example, let's examine the details. Additionally, let's assume that contagious individuals need to isolate themselves, and only those with severe symptoms require further medical attention.
  - Once more, we see a high accuracy of 95% but a low precision of 11%. The test incorrectly indicates a positive result for many people who do not have the virus. Unlike the first example, a false positive in this case is not as impactful for the individual (unnecessary isolation compared to unnecessary cancer treatment).
  - With a high specificity of 95% (TNR), the test is effective in correctly identifying a large portion of people who do not carry the virus. However, as discussed earlier, about 5% of individuals may still be unnecessarily required to enter isolation.
  - The sensitivity value is crucial in this case. We observe a fairly good value of 85% (TPR), indicating that most people carrying the virus are correctly identified. However, whether the test is acceptable depends on other factors. If the virus is highly contagious, an 85% sensitivity may not be sufficient. As discussed later in this chapter, we may need to adjust certain parameters of the test to increase sensitivity at the cost of lower specificity. This means accepting more false positives in return for reducing false negatives.
  - To enhance the test's sensitivity, we can set a minimum threshold for specificity (e.g., it must be >90%), or we can create a weighted sum of specificity and sensitivity to optimize the test as we make adjustments.

- In many classification tasks, there are multiple classes, such as labeling images with recognized animals or objects. The generalized confusion matrix compares each pair of the actual class and recognized class, forming an $K \times K$ table where K is the number of classes. Each cell represents the number of items with the actual class in the column and the recognized (or predicted) class in the rows. Some newer literature and software packages may transpose the table, but it does not affect the interpretation of the result. Let's consider the example below with 3 classes: "Woman", "Man", "Child":
  - The table's diagonal represents the correctly recognized classes, while all other cells indicate the prediction errors. For instance, out of the 20 women, 13 were correctly recognized. On the other hand, 2 women were wrongly recognized as men, and 5 women were wrongly recognized as children. To visualize correct and wrong classifications, we use different colors, which make it easier to identify areas of "confusion," especially with a large number of classes. Rearranging columns and rows to create "clusters of confusion" further helps pinpoint issues in the applied method
  - Accuracy is calculated by taking the sum of the cells on the diagonal and dividing it by the total population. In this example, $ACC = (13 + 15 + 57)/100 = 85\%$. On the other hand, the error rate is the complement of ACC, giving us $ERR = 1 - ACC = 15\%$ for the example shown
  - But how do we calculate sensitivity, specificity, precision, and other metrics from the binary confusion matrix when dealing with multiple classes? We can use a simple trick: if we want to focus on a particular class C_k, we can collapse the multi-class view into a binary view with new conditions "$\in C_k$" and "$\notin C_k$" and then compute the measures as introduced before. Let's consider an example on the next page

		Actual Class		
	Population (100)	Woman (20)	Man (20)	*Child (60)*
Recognized Class	Woman (19)	13	4	2
	Man (18)	2	15	1
	Child (63)	5	1	57

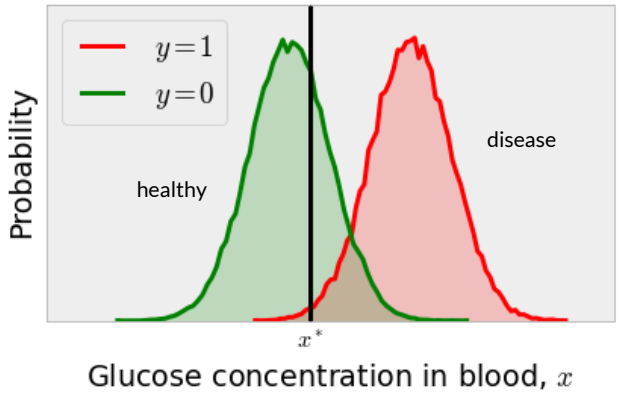
- By collapsing the classes "Woman" and "Child", we can delve deeper into the performance of the example:
  - For the class "Woman," we observe a high specificity (correctly dismissing the class) but low precision and sensitivity values. A closer examination of the prevalence also indicates that the high specificity and accuracy values are a consequence of the class imbalance ($20/100 = 20\%$)
  - For the class "Child," we observe high values for sensitivity, specificity, and precision, indicating that the method successfully recognizes children in the images. The balanced prevalence of 60/100=60% and the high accuracy of 91% suggest that the method is performing well for this class

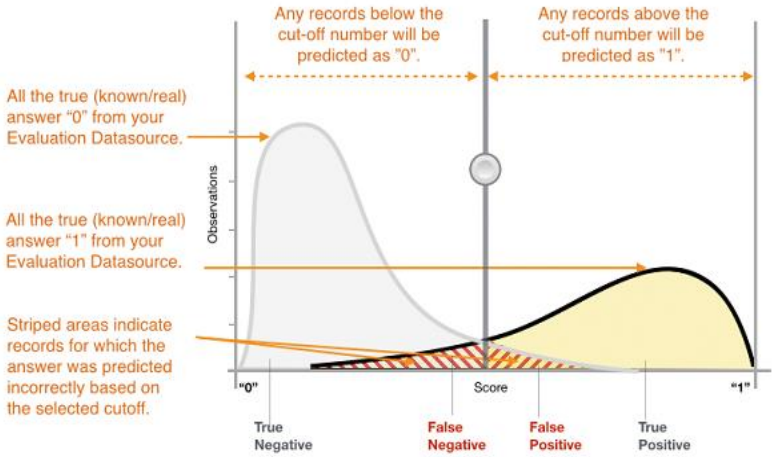
		Actual Class		
	Total Population	Woman (*P=20*)	Not a Woman (*N=80*)	
Recognized Class	Woman (*19*)	**True Positive** (*TP=13*)	**False Positive** (*FP=6*)	$PPV = \frac{13}{19} = 68\%$ (precision)
	Not a Woman (*81*)	**False Negative** (*FN=7*)	**True Negative** (*TN=74*)	$NPV = \frac{74}{81} = 91\%$
		$TPR = \frac{13}{20} = 65\%$ (sensitivity)	$TNR = \frac{74}{80} = 93\%$ (specificity)	$ACC = \frac{87}{100} = 87\%$

		Actual Class		
	Total Population	Child (*P=60*)	Not a Child (*N=40*)	
Recognized Class	Child (*63*)	**True Positive** (*TP=57*)	**False Positive** (*FP=6*)	$PPV = \frac{57}{63} = 90\%$ (precision)
	Not a Child (*37*)	**False Negative** (*FN=3*)	**True Negative** (*TN=34*)	$NPV = \frac{34}{37} = 92\%$
		$TPR = \frac{57}{60} = 95\%$ (sensitivity)	$TNR = \frac{34}{40} = 85\%$ (specificity)	$ACC = \frac{91}{100} = 91\%$

2.6 Optimizing Hyperparameters

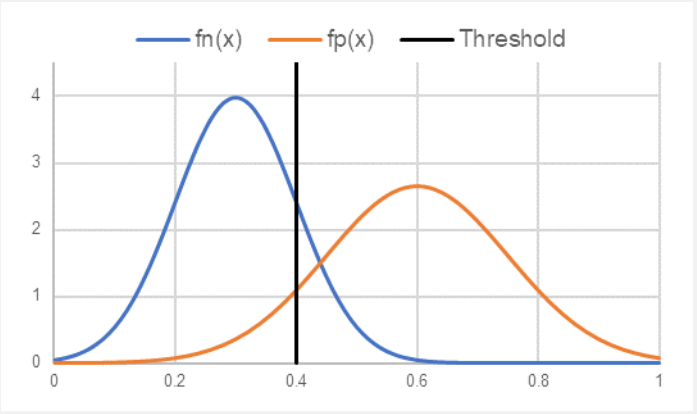
- In many binary classification algorithms, the output is not a direct assignment to a class, but rather a prediction score. Take the example on the top right, where there is a clear correlation between glucose concentration and the presence of diabetes:
  - Plotting the distribution of concentration results in two curves: green for healthy people and red for those with diabetes
  - The distributions overlap, causing uncertainty in determining whether a person is healthy or sick in this overlapping area
  - To make predictions, we need to choose a threshold, denoted as x^* in the figure. Scores (glucose concentration in this case) below x^* are classified as healthy while scores above x^* are classified as sick. x^* becomes a hyperparameter
- But how do we select x^*? Let's consider the bottom right figure:
  - Scores below the threshold are predicted as negative cases which we referred to as "false" in the confusion matrix
  - Scores above the threshold are predicted as positive cases which we referred to as "true" in the confusion matrix
  - In general, the actual distributions of scores in the positive and negative classes overlap, making it impossible to perfectly separate the two classes. This results in false negatives (scores below threshold but actually positive cases) and false positives (scores above threshold but actually negative cases)
  - In the example on the top right, we can choose the threshold x^* as shown to safely rule out diabetes with minimal false negatives (high sensitivity). Alternatively, we could set x^* further to the right to reduce false positives and accurately identify people with diabetes (high specificity)

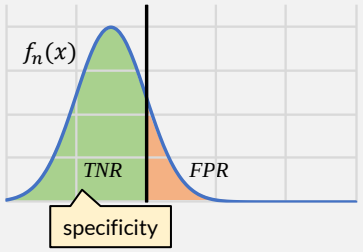




source: https://docs.aws.amazon.com/machine-learning/latest/dg/binary-classification.html

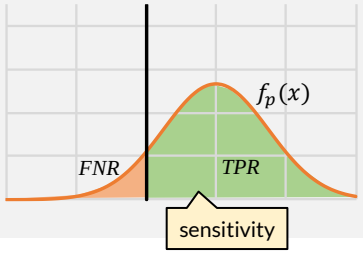
- Let's have a closer look at how the setting of the threshold impacts the results of the prediction:
  - The plot on the bottom left shows the actual distribution of scores between 0 and 1 for the negative class as $f_n(x)$ and for the positive class as $f_p(x)$. When we introduce a threshold T, we classify a score $x < T$ as negative
  - By considering only $f_n(x)$, we can determine the true negatives (green) and false positives (red) as shown in the upper right plot. The true negative rate (TNR), or specificity, is then the integral of $f_n(x)$ over scores from $-\infty$ to T (the green area under $f_n(x)$). Similarly, we obtain the false positive rate (FPR) as the integral of $f_n(x)$ over scores from T to $+\infty$ (the red area under $f_n(x)$)
  - By considering only $f_p(x)$, we can determine the true positives (green) and false negatives (red) as shown in the lower right plot. The false negative rate (FNR) is then the integral of $f_p(x)$ over scores from $-\infty$ to T (the red area under $f_p(x)$). Similarly, we obtain the true positive rate (TPR), or sensitivity, as the integral of $f_p(x)$ over scores from T to $+\infty$ (the green area under $f_p(x)$)
  - When the two distributions, $f_n(x)$ and $f_p(x)$, overlap, shifting the threshold T to the left will increase the true positive rate (TPR), or sensitivity, while decreasing the true negative rate (TNR), or specificity. Conversely, moving the threshold T to the right will decrease the TPR, or sensitivity, while increasing the TNR, or specificity
  - Prevalence can significantly impact the outcome. For example, if we have 10 times more negative cases, having equal values for TNR and TPR would result in 10 times more true negatives than true positives





$$TNR(T) = \int_{-\infty}^{T} f_n(x)\, dx$$

$$FPR(T) = \int_{T}^{\infty} f_n(x)\, dx$$

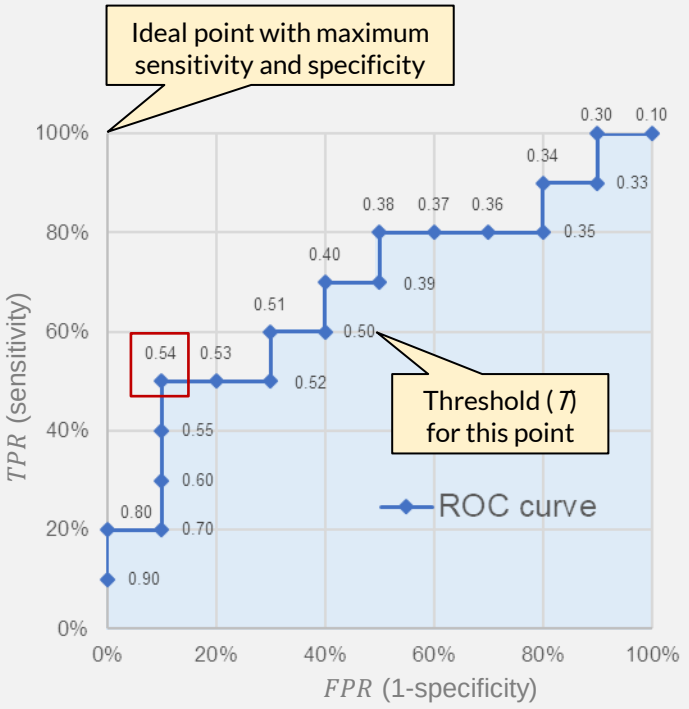


$$FNR(T) = \int_{-\infty}^{T} f_p(x)\, dx$$

$$TPR(T) = \int_{T}^{\infty} f_p(x)\, dx$$

- The **Receiver Operating Characteristic curve**, or **ROC curve**, is a graph that visualizes the performance of a binary classifier by varying the threshold. It plots true positive rate (TPR, sensitivity) on the y-axis, and false positive rate (FPR, 1-specificity) on the x-axis. We can generate the ROC curve by sorting the data items in the validation set based on their scores in descending order, as illustrated in the table below:
  - The first column, labeled "Class", represents the actual class of each item (ground truth). The second column, labeled "Score", contains the scores assigned by the binary classifier to each item
  - The threshold for each row is set to the score value in that row. All items at and above the row's threshold are classified as "positives" by the classifier, while all items below the threshold are classified as "negatives"
  - Let's focus on the highlighted row in the table: using a threshold of 0.54, we can determine the true positives (TP=5, by counting all P's above and including the row), false positives (FP=1, by counting all N's above and including the row), false negatives (FN=5, by counting all P's below the row), and true negatives (TN=9, by counting all N's below the row). With a total of 10 P's and N's, we can calculate the TPR and FPR and then plot the results, as shown in the graph on the right-hand side.

						sensitivity	1-specificity	
Class	Score	TP	FP	FN	TN	TPR	FPR	ACC
P	0.90	1	0	9	10	10%	0%	55%
P	0.80	2	0	8	10	20%	0%	60%
N	0.70	2	1	8	9	20%	10%	55%
P	0.60	3	1	7	9	30%	10%	60%
P	0.55	4	1	6	9	40%	10%	65%
P	0.54	5	1	5	9	50%	10%	70%
N	0.53	5	2	5	8	50%	20%	65%
N	0.52	5	3	5	7	50%	30%	60%
P	0.51	6	3	4	7	60%	30%	65%
N	0.50	6	4	4	6	60%	40%	60%
P	0.40	7	4	3	6	70%	40%	65%
N	0.39	7	5	3	5	70%	50%	60%
P	0.38	8	5	2	5	80%	50%	65%
N	0.37	8	6	2	4	80%	60%	60%
N	0.36	8	7	2	3	80%	70%	55%
N	0.35	8	8	2	2	80%	80%	50%
P	0.34	9	8	1	2	90%	80%	55%
N	0.33	9	9	1	1	90%	90%	50%
P	0.30	10	9	0	1	100%	90%	55%
N	0.10	10	10	0	0	100%	100%	50%



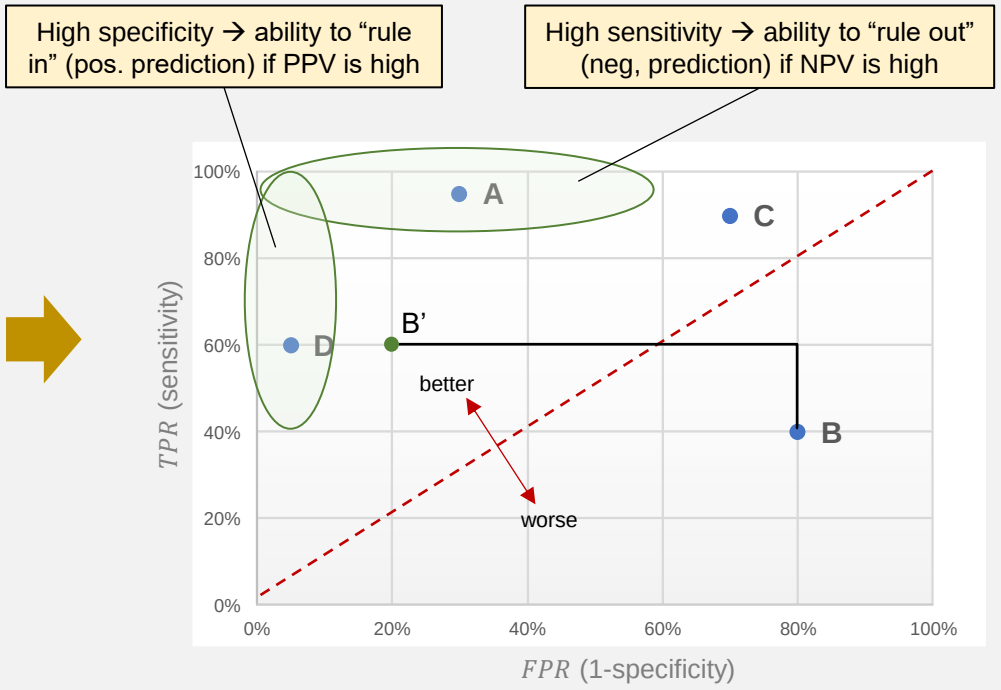
- Interpretation of the ROC-curve: consider the 4 methods **A, B, C, D** and their confusion matrices below:
  - **A** lies in an are with high sensitivity. If the NPV is also high (consider the prevalence), then **A** can effectively "rule out" (negative predictions). **A** would be a good candidate for the diagnostic process to rule out diseases
  - **B** lies in the area below the diagonal. In such cases, we can construct **B'** which negates the outcome of **B** to obtain a better method: TP and FN switch their values; FP and TN switch their values. This results in new values for the diagram as follows: $TPR' = 1 - TPR = 60\%$ and $FPR' = 1 - FPR = 20\%$
  - **C** has a high sensitivity but a lower NPV than **A** due to its lower specificity. This affects its ability to "rule out," and a negative prediction is incorrect in 25% of the cases, making it unsuitable for many scenarios
  - **D** lies in an area with high specificity. If the PPV is also high (consider the prevalence), then **D** can effectively "rule in" (positive predictions). **D** would be a good candidate to provide evidence for the presence of a disease
  - Finally, note that the points $(TPR = 0, FPR = 0)$ and $(TPR = 1, FPR = 1)$ are the results of extreme thresholds that always return negative or positive predictions, respectively. Methods close to these areas, including **C** below, may exhibit a too strong bias towards negative or positive predictions

A	
$TP = 95$	$FP = 30$
$FN = 5$	$TN = 70$
$TPR = 95\%$	$FPR = 30\%$
$PPV = 76\%$	$NPV = 93\%$
$ACC = 83\%$	

B	
$TP = 40$	$FP = 80$
$FN = 60$	$TN = 20$
$TPR = 40\%$	$FPR = 80\%$
$PPV = 33\%$	$NPV = 25\%$
$ACC = 30\%$	

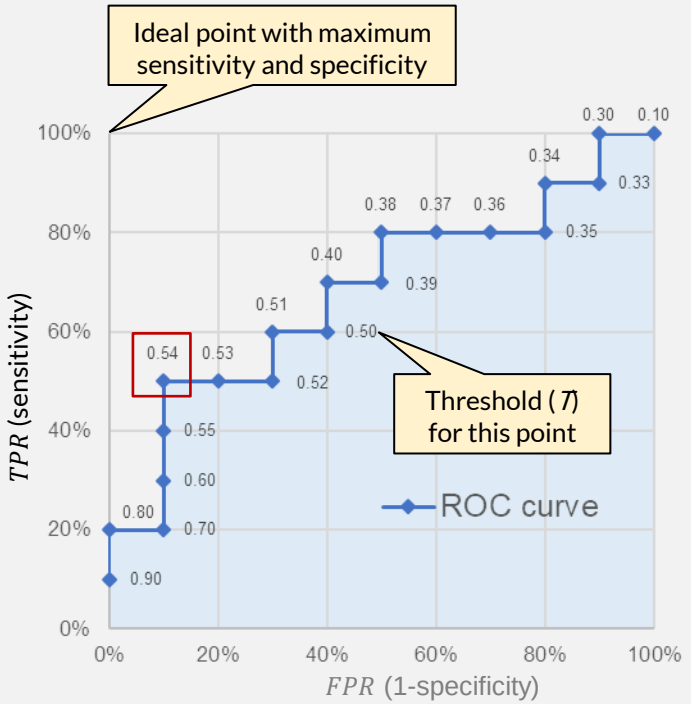
C	
$TP = 90$	$FP = 70$
$FN = 10$	$TN = 30$
$TPR = 90\%$	$FPR = 70\%$
$PPV = 56\%$	$NPV = 75\%$
$ACC = 60\%$	

D	
$TP = 60$	$FP = 5$
$FN = 40$	$TN = 95$
$TPR = 60\%$	$FPR = 5\%$
$PPV = 92\%$	$NPV = 70\%$
$ACC = 78\%$	



- Back to the example from before, depicted again the bottom of the page:
  - To find the optimal threshold, we must consider the specific performance goal for our task. If we aim to "rule out" or "rule in" the condition of the task, we choose thresholds with corresponding TPR and FPR values in or close to that area as highlighted on the previous page
  - In machine learning classification tasks, accuracy is a commonly used performance measure. In this case, we would select the threshold that maximizes the accuracy value. In the example shown below, the threshold of 0.54 gives the highest accuracy and is thus a good choice for this scenario
  - If we don't have a specific performance goal, we can choose the threshold that is closest to the ideal point in the upper left corner. Alternatively, we can optimize for the sum of sensitivity and specificity to make the decision
  - The **area under the ROC curve (AUC)** is a comprehensive performance measure across all thresholds. It reflects how well a method can distinguish between positive and negative predictions based on the computed scores. In the example below, if a method consistently assigns higher scores to the P's than the N's, the AUC would cover the entire space

Class	Score	TP	FP	FN	TN	TPR (sensitivity)	FPR (1-specificity)	ACC
P	0.90	1	0	9	10	10%	0%	55%
P	0.80	2	0	8	10	20%	0%	60%
N	0.70	2	1	8	9	20%	10%	55%
P	0.60	3	1	7	9	30%	10%	60%
P	0.55	4	1	6	9	40%	10%	65%
P	0.54	5	1	5	9	50%	10%	70%
N	0.53	5	2	5	8	50%	20%	65%
N	0.52	5	3	5	7	50%	30%	60%
P	0.51	6	3	4	7	60%	30%	65%
N	0.50	6	4	4	6	60%	40%	60%
P	0.40	7	4	3	6	70%	40%	65%
N	0.39	7	5	3	5	70%	50%	60%
P	0.38	8	5	2	5	80%	50%	65%
N	0.37	8	6	2	4	80%	60%	60%
N	0.36	8	7	2	3	80%	70%	55%
N	0.35	8	8	2	2	80%	80%	50%
P	0.34	9	8	1	2	90%	80%	55%
N	0.33	9	9	1	1	90%	90%	50%
P	0.30	10	9	0	1	100%	90%	55%
N	0.10	10	10	0	0	100%	100%	50%



2.7 Literature and Links

- **MS MARCO** (Microsoft Machine Reading Comprehension Dataset), https://microsoft.github.io/msmarco/
- **Text REtrieval Conference**, http://trec.nist.gov/
- **Kaggle Competitions**, https://www.kaggle.com/competitions
- Kent, Allen; Berry, Madeline M.; Luehrs, Jr., Fred U.; Perry, J.W. (1955). **Machine literature searching VIII. Operational criteria for designing information retrieval systems**. American Documentation. **6** (2): 93. doi:10.1002/asi.5090060209.
- Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier (1999). **Modern Information Retrieval**. New York, NY: ACM Press, Addison-Wesley. ISBN 0-201-39829-X
- David A. Grossman, Ophir Frieder, **Information Retrieval Algorithms and Heuristics**, Kluwer Academic Publishers, 1998
- Zou, Kelly H.; O'Malley, A. James; Mauri, Laura (2007); **Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models**, Circulation, 115(5):654–7, http://circ.ahajournals.org/content/115/5/654.full
- Hanley, James A.; McNeil, Barbara J. (1982). **The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve,** *Radiology*. **143** (1): 29-36. PMID 7063747. doi:10.1148/radiology.143.1.7063747.
- Fawcett, Tom (2006), **An Introduction to ROC Analysis**, *Pattern Recognition Letters*. **27** (8): 861–874. doi:10.1016/j.patrec.2005.10.010 or http://people.inf.elte.hu/kiss/11dwhdm/roc.pdf
- Amazon AWS, **Amazon Machine Learning – Developer Guide**, https://docs.aws.amazon.com/machine-learning/latest/dg/evaluating-model-accuracy.html
- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, **Introduction to Information Retrieval, Chapter 8: Evaluation in information retrieval,** Cambridge University Press. 2008. Online version: https://nlp.stanford.edu/IR-book/
- D. R. Radev; H. Qi; H. Wu; W. Fan (2002). **Evaluating web-based question answering systems,** Proceedings of LREC. http://www.lrec-conf.org/proceedings/lrec2002/pdf/301.pdf
- Related Wikipedia articles
  - **Evaluation measures**, https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)
  - **Discounted cumulative gain**, https://en.wikipedia.org/wiki/Discounted_cumulative_gain
  - **Confusion Matrix**, https://en.wikipedia.org/wiki/Confusion_matrix
    Note that the Wikipedia article uses the transposed confusion matrix. There is also some comments on that on the discussion page
  - **ROC curve**, https://en.wikipedia.org/wiki/Receiver_operating_characteristic