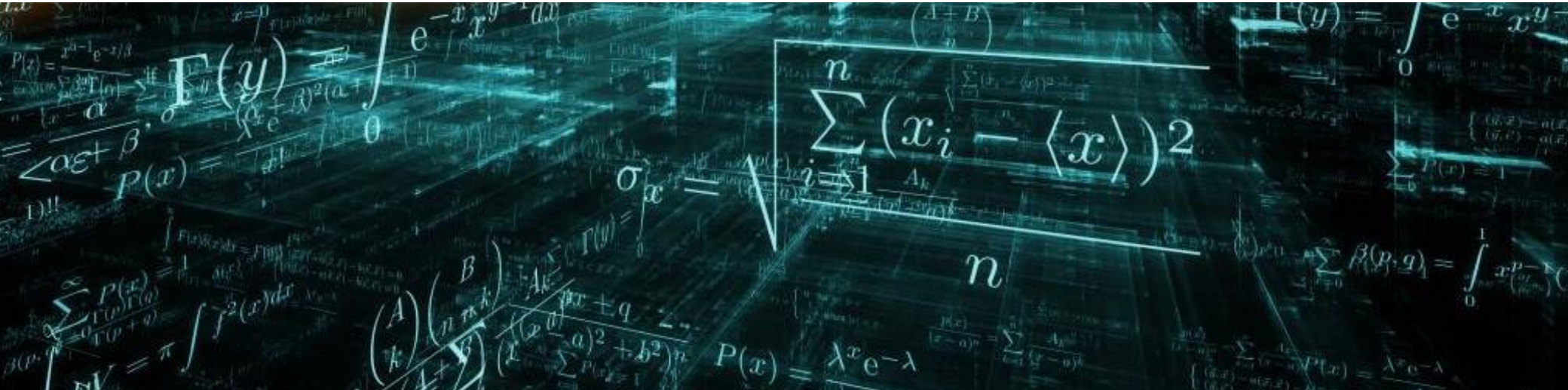
Universität Basel

Computer Science / 15731-01 / 2025

Multimedia Retrieval

Chapter 10: Multimodal Content Analysis

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



10.1 Introduction

- Multimodal content analysis supports modern information retrieval systems by interpreting and combining data from text, images, audio, and video. Its goal is to extract useful insights from varied media and to unify them into coherent, searchable knowledge representations. The field brings together computer vision, natural language processing, audio signal processing, and machine learning. It has become more important as digital media grows rapidly across platforms and devices.
- The explosive growth of multimedia content, especially on social and streaming platforms, has increased the need for systems that can understand and organize multimodal information. Unlike traditional unimodal systems that work in a single format such as text-only search, multimodal analysis recognizes that each medium provides complementary signals. Text gives explicit meaning, images add visual context, and audio supplies emotional or timing cues. Together these elements produce richer and more human interpretations of meaning.
- A key challenge in this process is the semantic gap, the disconnect between low-level computational features, such as pixel intensities or sound frequencies, and the higher-level concepts they represent. Closing this gap requires combining bottom-up methods that extract features directly from media with top-down techniques guided by semantic knowledge and context. Recent advances in neural models, especially transformer-based architectures and contrastive learning methods like CLIP, have created more integrated cross-modal representations, enabling more flexible retrieval that is better grounded in meaning.
- Different modalities experience the semantic gap in different ways, but the problem is common across them. For text, keyword-based search worked well for decades. Only recently, with semantic embeddings and neural networks, did we begin to capture meaning, context, and intent. For images, audio, and video the problem is harder. These data types have no simple and built in way to extract information that is directly useful to users like text keywords are. A photo contains thousands of pixels, but without semantic interpretation, a system cannot tell whether the patterns show a cat, a cloud, or abstract shapes. That disconnect requires several complementary strategies to connect low level data to high level meaning.
- In this chapter, we build the foundation for multimodal content analysis, present the main feature types, and take a closer look at metadata and its use in search. We then cover in subsequent chapters perceptual features for images, audio, and video, followed by conceptual and semantic features.

- Multimodal content analysis is essential for industries and platforms that rely on rich media:
  - **Social Media and Digital Content Platforms:** Users generate and consume billions of images, videos, and multimedia posts every day, creating major challenges for content discovery and recommendation. Traditional social platforms relied mainly on collaborative filtering, which suggests items based on what similar users engaged with, and on simple link-based propagation. These methods captured behavioral patterns but struggled to understand the actual content. For example, a user who likes cat videos might be shown more cat material, yet the system cannot tell if a clip shows playful kittens or a lion in a documentary without deeper content analysis.

    TikTok's algorithm suggests videos by analyzing both user behavior, such as likes, shares, and watch time, and video content, such as captions, hashtags, sounds, effects, and visual features. In 2025 it places stronger emphasis on original, TikTok-native content and on the depth of session engagement, and it penalizes cross-posted or recycled videos from other platforms. The recommendation pipeline runs in two phases. A candidate generation step selects videos that match a user's behavioral profile and expressed interests. Then a fine-ranking stage weighs video content features alongside engagement metrics to optimize suggestions for each user. As TikTok evolves, search and content-based discovery play a larger role, merging direct content analysis with feedback from user behavior. This creates personalized feeds based on what is shown inside the video, not just how users interact with it, and it boosts discovery and viral reach for creators who focus on originality and production quality.

TikTok

Year	Algorithm Characteristics
2020	Heavy emphasis on viral trends, dance challenges, and trending audio clips. The "For You Page" prioritized content using popular sounds regardless of creator size.
2022	Text-heavy looping videos dominated, with 7-8 second clips containing multiple text overlays. Algorithm rewarded artificial watch time inflation through forced re-watching.
2024	Shift to longer content with 60+ second requirement for monetization. Algorithm began measuring depth of engagement rather than surface-level metrics.
Early 2025	Complete pivot to search-based discovery. Algorithm analyzes search intent, keyword relevance, and authentic engagement over viral metrics.
Key Shifts	From entertainment-first to information-first platform, from viral moments to sustained engagement, from broad reach to targeted communities
Metric Evolution	2020: Views and shares dominated. 2022: Loop count mattered most. 2024: Total watch time priority. 2025: Search ranking and completion rates rule.
Future Direction	Increasing sophistication in content understanding, AI-driven personalization, potential voice/visual search integration, commerce-content fusion

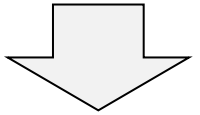
source: https://napolify.com/blogs/news/algorithm-tiktok

- **Media Archives and Newsrooms:** Large media organizations face special challenges when managing huge multimedia archives built up over decades. A typical broadcaster or news agency produces and licenses millions of items, including photos, video, audio, and text. These archives are valuable for everyday work, but they are useful only if staff can find the right material quickly. In a news story, for example, a newly elected president prompts newsrooms to gather biographies, historical photos, video clips of key moments in the candidate's career, and related context at once.

  Traditional archive management depended on manual metadata annotation by human catalogers, who described each asset with keywords, dates, locations, and subjects. That method produced high quality metadata but could not scale to modern production volumes and created major bottlenecks. Automated or semi automated tagging with multimodal analysis is now essential. Modern systems use facial recognition to identify people across thousands of hours of footage, scene detection to break video into meaningful segments, object detection to list items seen in images, and speech to text to make spoken content searchable.

  Recent advances in AI let media companies index video, audio, and images with embeddings that capture meaning rather than relying on manual tags or keywords. This enables semantic search, so a query like "earthquake" can find relevant clips, interviews, and photos even without exact text matches. Using multimodal models such as CLIP, broadcasters can generate embeddings from visuals, speech, and text, making archives searchable by concept and context and giving faster, richer access to decades of material and more efficient newsroom workflows.

The archive of the Danish Broadcasting Corporation
https://www.flickr.com/photos/kulturarvsprojektet/6498650083/



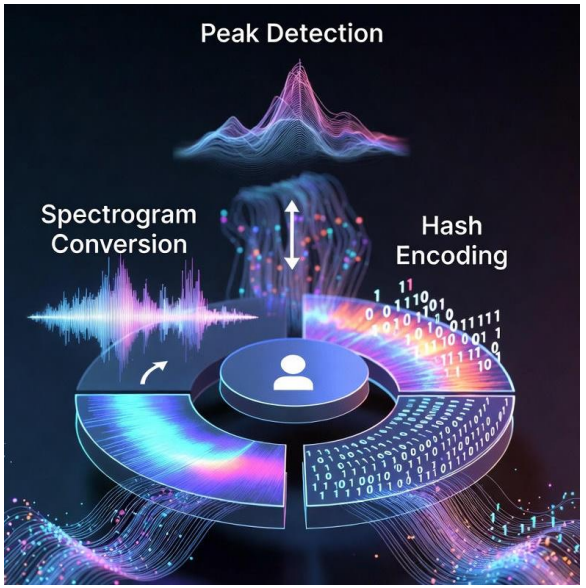
Breaking News!

- **Music and Audio Services:** Applications like Spotify and Shazam show multimodal retrieval in audio. Shazam uses audio fingerprints, compact perceptual signatures, to identify songs from short recordings. Spotify combines acoustic features, lyrics, and user behavior to recommend tracks that match a mood or theme. The challenge is aligning acoustic patterns with textual or emotional descriptions and letting users find music through diverse cues, such as songs that sound like this or music that fits a rainy afternoon.

  Shazam: The approach is to generate compact, distinctive audio fingerprints that remain identifiable even under challenging conditions such as background noise, compression, or low recording quality. Shazam converts audio into spectrograms, finds prominent frequency peaks over time, and encodes relationships among those peaks into hash values that act as fingerprints. These fingerprints must resist common distortions while still distinguishing millions of songs in the database.

  Spotify: Music recommendation requires understanding both perceptual similarity and musical traits. Spotify analyzes acoustic features such as tempo, key, timbre and energy. It also extracts higher level attributes like genre, mood and instrumentation, processes textual metadata such as artist information, user tags and lyrics, and combines collaborative filtering signals from listening behavior. The challenge is to capture objective musical properties and subjective perceptual qualities that shape listener preferences. A user who likes a song may do so for the vocalist's timbre, a rhythmic pattern, the emotional lyrics or cultural associations, so the system must model multiple dimensions of similarity at once.

Shazam

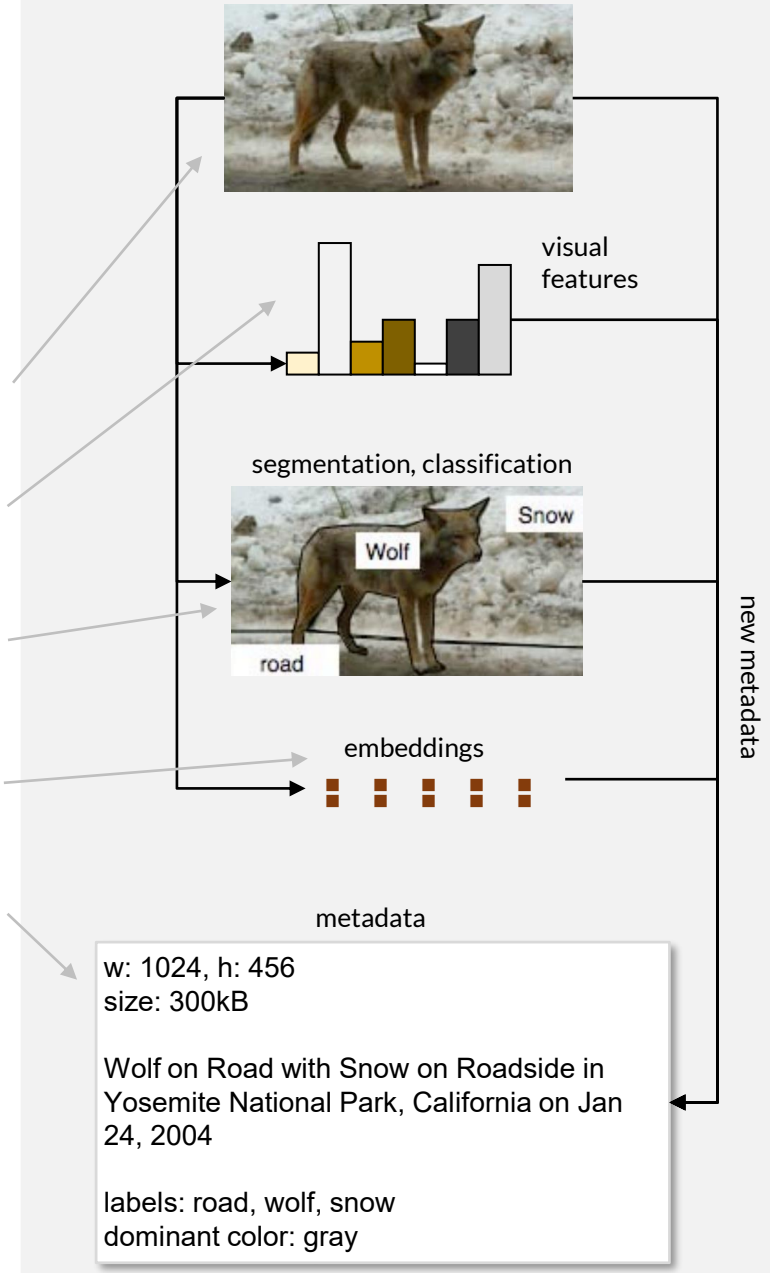


10.1.1 Addressing the Semantic Gap Across Modalities

- The semantic gap affects all parts of multimodal content analysis and appears differently across media types:
  - For **text**, keyword search acted as a practical bridge for information retrieval. Users could enter terms and systems could find documents that contained those terms. This approach missed semantic relationships such as synonyms, related concepts, and contextual meanings, and it struggled with ambiguity and with different ways people use language. The development of embeddings and neural networks changed text search by capturing semantic relationships in vector space representations.
  - For images, audio, and video, closing the semantic gap requires several complementary strategies at different levels of abstraction. At the most basic level, **metadata annotations** provide text descriptions that turn a multimodal search into a text search. These annotations can be manual, for example photographers writing captions or archivists creating detailed catalog entries; semi-automated, for example facial recognition suggesting identities for human verification; or fully automated, for example image captioning models generating descriptions.
  - **Perceptual and low-level features** work directly on raw media to extract signal-based descriptors that reflect human perception. In images, color histograms show hue and brightness distributions, which help queries such as find red images. Texture features use Gabor filters, co-occurrence matrices, or wavelet analysis to describe surface patterns, and edge detection finds contours and shapes. In audio, Mel-Frequency Cepstral Coefficients (MFCCs) capture spectral properties similar to human hearing, enabling speech and music analysis. These features record sensory qualities but do not provide semantic meaning. For example, a blue area might be sky, water, or fabric, and higher-level reasoning is needed to resolve that ambiguity.
  - **Classification and clustering methods** link perceptual features to semantic categories. Object detection finds entities such as cars and people. Scene classification labels environments, for example indoor or outdoor. Face recognition identifies people across different media. These systems moved from hand-crafted features and support vector machines to deep learning models such as convolutional neural networks and vision transformer models, which capture both local and global patterns and narrow the semantic gap.
  - **Cross-modal embeddings** close the semantic gap by placing text, images, and audio into a shared space. Models like CLIP learn to match text and images using contrastive learning. This lets users search with text or with images, for example finding pictures of a dog in a park. Other models map speech and music to text descriptions, making it possible to retrieve content across modalities.

- Feature extraction is the foundation of multimodal content analysis. It converts complex raw media into structured representations that can be compared, classified, and retrieved. It works in layers, from basic signal representation to abstract semantic meaning. Each layer transforms the previous one and gradually closes the semantic gap between low-level data and high-level understanding.

Layer	Examples	Purpose
Raw-Level (Informational)	JPG, MP3, WAV, MP4, TXT	Store original content; basic digital representation
Low-Level (Perceptual)	Color histograms, edge maps, MFCCs, spectral centroid, textures	Capture sensory properties such as color, sound, or motion
Mid-Level (Structural)	Clustering, SVM, CNN/ViT embeddings, object or genre classification	Represent patterns, structure, and relationships within data
High-Level (Semantic)	Captioning, keyword generation, OCR, speech-to-text, topic extraction	Produce interpretable, concept-level meaning
Contextual (Metadata)	Timestamps, geolocation, device data, user tags, social context	Provide external descriptive context for indexing and filtering



- Next, we will briefly review each level. Metadata is covered later in this chapter, and the perceptual, structural, and semantic levels are discussed in subsequent chapters. This chapter also explains how to measure how well structural-level classification works using the confusion matrix.

- **Raw-Level (Informational):**
  - At the foundation is the raw informational layer, which stores original data in digital form. Examples include image formats such as JPG or PNG, audio formats like MP3 or WAV, and video containers such as MP4 or AVI. These files hold raw sensory data, for example pixel intensities, color channels, or sound pressure amplitudes, but they are not directly meaningful to humans or computers without further interpretation.
  - For example, a video of a political speech is stored as millions of frames and sound samples. By themselves, these data only record color and amplitude values. The idea that the clip shows a president giving a speech is not present at this stage. Raw text, however, already carries meaning because language symbols encode ideas. In other modalities, meaningful representation must be extracted through analysis.
  - This level is for storage and access. It preserves the media's fidelity and reproducibility, but it helps little with search or retrieval because similarity in raw data rarely matches how things look or what they mean.
- **Low-Level (Perceptual)**
  - The perceptual layer converts raw data into features that match how people perceive the world. These features summarize visual, sound, and motion characteristics without interpreting them in detail.
  - In images, this includes descriptors such as color histograms, texture measures, edge maps, or local feature points (for example SIFT). They allow systems to find visually similar pictures, such as grouping sunset photos by dominant red tones, but they cannot distinguish between conceptually different scenes that share similar colors (for example a red car and a sunset).
  - In audio, perceptual analysis often uses features such as Mel-Frequency Cepstral Coefficients (MFCCs), spectral centroid, and chroma features. These capture the timbre and tonal qualities of sound that match human hearing. For example, Shazam finds songs by matching distinctive spectral patterns, showing how perceptual signatures can uniquely identify audio content.
  - In video, perceptual features extend into the time dimension. Examples include optical flow, motion vectors, and 3D convolutional filters, which describe how visual information changes over time. These features capture rhythm and motion style, helping to distinguish a slow-paced documentary from an action scene.
  - This layer connects raw data to perceptual understanding by summarizing how content looks or sounds. It allows efficient comparison and grouping but does not capture meaning or context.

- **Mid-Level (Structural)**
  - The structural layer arranges sensory data into patterns, relationships, or compact summaries. It shifts focus from describing what data look or sound like to recognizing how they are organized and grouped.
  - Historically, this level used methods such as clustering, principal component analysis (PCA), and support vector machines (SVMs) to group or classify features into categories, objects, genres, or events. Modern approaches use deep learning, with convolutional and transformer-based models that produce embeddings to represent content similarity in a learned vector space.
  - In image analysis, structural features include detected objects (for example, cat, person, car) or scene categories (beach, stadium). In audio, structural models identify speech segments, music genres, or mood clusters. In video, this layer captures shot boundaries, scene coherence, and action patterns. These structural abstractions allow systems to connect perceptually similar but not identical media, for example grouping different performances of the same song or identifying paintings with a similar style.
  - The main purpose of this level is organization and representation. By learning the underlying structure of content, systems can represent different types of data in common forms, enabling higher-level reasoning and retrieval.
- **High-Level (Semantic)**
  - At the semantic layer, systems convert structural patterns into clear concepts that people can understand. This is where media analysis reaches real understanding.
  - Semantic extraction includes tasks such as caption generation, keyword tagging, and topic recognition. An image might be labeled "man riding a horse," or an audio segment transcribed into text through speech-to-text. Video summarization systems can describe scenes, such as a soccer player scoring a goal, or detect broader narrative themes. In text, topic modeling or named-entity recognition produce similar outcomes, extracting meaning that corresponds to human categories.
  - Deep learning models, especially transformer based architectures, have greatly improved this layer. Multimodal models such as CLIP and BLIP match text and image meanings, so a user can search for a red car on a mountain road and find matching images or video clips without explicit tags.
  - The purpose here is interpretability: to create representations that align with linguistic or conceptual categories and connect data to human understanding.

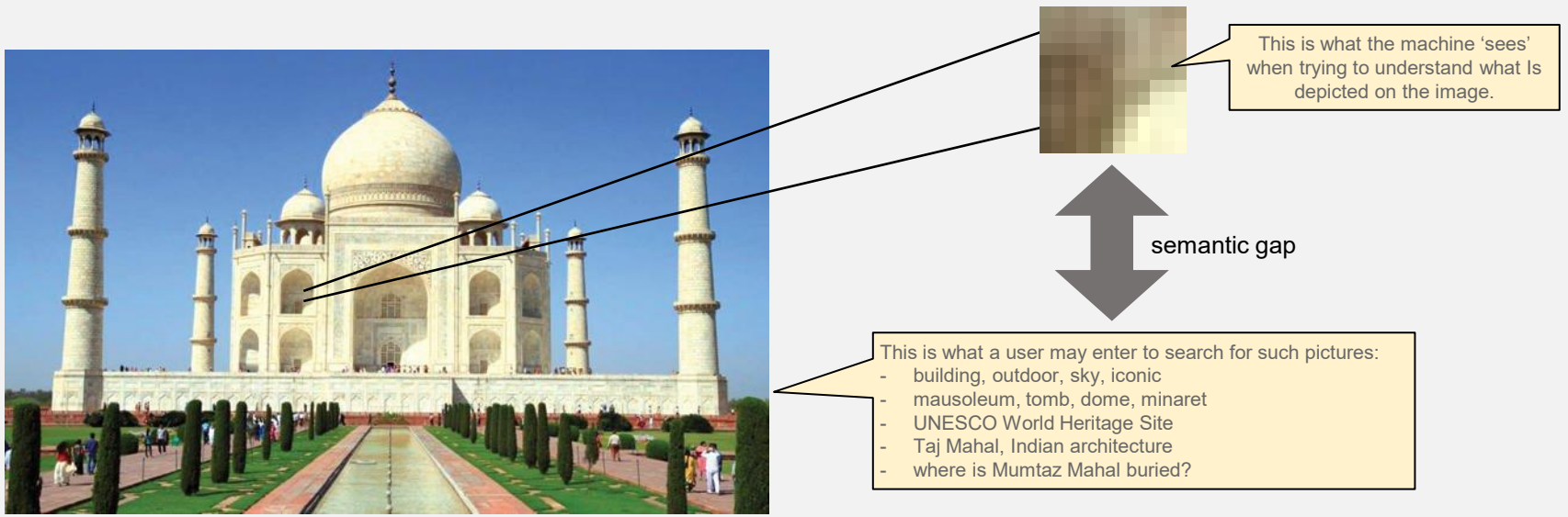
- **Contextual (Metadata)**
  - The contextual layer adds external information to content features, locating the media in time, place, and social context. Metadata can be manual (titles, descriptions, tags), automatic (timestamps, GPS coordinates, camera settings), or inferred (scene detection, face recognition, social network links).
  - For instance, a news image might include a timestamp, the photographer's name, and the location, which are valuable cues for retrieval even when the visual content is ambiguous. Music databases such as MusicBrainz, film repositories like IMDb, and book catalogues such as Goodreads depend on curated metadata to link works by author, genre, or release date. In contrast, personal photo collections often have little or inconsistent metadata, so automated enrichment is crucial for organization and search.
  - This layer is essential for indexing, filtering, and recommendation. Metadata lets users refine searches, for example photos taken in Paris in 2023, and supports hybrid retrieval that combines semantic and contextual information. It also enables provenance tracking and personalization, both vital for large-scale multimedia systems.
- Video and audio add time and space dimensions that make multimodal analysis harder. Unlike static images, video unfolds over time and has narrative structure, transitions, and changing content that must be segmented and indexed at the right levels of detail. Someone searching video content usually wants specific segments or scenes, not whole hours long recordings. This is like finding relevant passages in a long text rather than retrieving the entire document.
  - **Temporal segmentation** divides a video into hierarchical units of semantic coherence. At the finest level, shot boundary detection finds transitions between camera shots. These include abrupt cuts, where one frame immediately switches to a different view, and gradual transitions such as fades, dissolves, and wipes, where one shot blends into another. Shots are atomic units of continuous camera recording and provide the basis for higher level segmentation. Scene segmentation groups related shots into coherent narrative units. For example, a conversation may span multiple shots showing different speakers and angles, but all these shots belong to the same scene. Scene detection is harder because it requires semantic understanding rather than only visual discontinuity.
  - **Spatio-temporal features** capture motion and activity patterns that unfold across space and time. Optical flow measures pixel movement between consecutive frames, revealing object motion and camera motion. Three-dimensional convolutional neural networks extend spatial convolutions into the temporal dimension, learning features from short video volumes that represent actions. Temporal convolutional networks model sequential dependencies and temporal context. These spatio-temporal representations enable queries about dynamic content, such as finding video clips of running, identifying car chase scenes, or detecting unusual movement patterns in surveillance footage, which cannot be answered by analyzing individual frames alone.

10.2 Meta Data Management

- We have previously discussed the concept of the “semantic gap” and its challenges. Let’s look at an example in the area of image retrieval and consider the picture below of the Tay Mahal as a running example:
  - The context of the picture is as follows:

    The Taj Mahal, located in Agra, India, is a magnificent mausoleum built by Emperor Shah Jahan in memory of his beloved wife Mumtaz Mahal, who passed away in 1631. The Taj Mahal is one of the most iconic landmarks in the world and is recognized as a UNESCO World Heritage Site. Mumtaz Mahal's tomb is situated in the main chamber, alongside Shah Jahan's tomb.

  - When users search for pictures of the Taj Mahal, they may use various keywords, as depicted in the lower right box. The image retrieval system must then find matches to these search queries within its image database. However, it faces a challenge as it cannot directly compare the pixel information of images with the keywords provided by the users. Unlike text retrieval, this disparity requires other methods to bridge the semantic gap.
  - To address this gap, we need to map the distinct perspectives (pixels in images and user-provided keywords) into a comparable space where we can more effectively assess the relevance. In this introductory chapter, we focus on meta data to bridge the semantic gap.

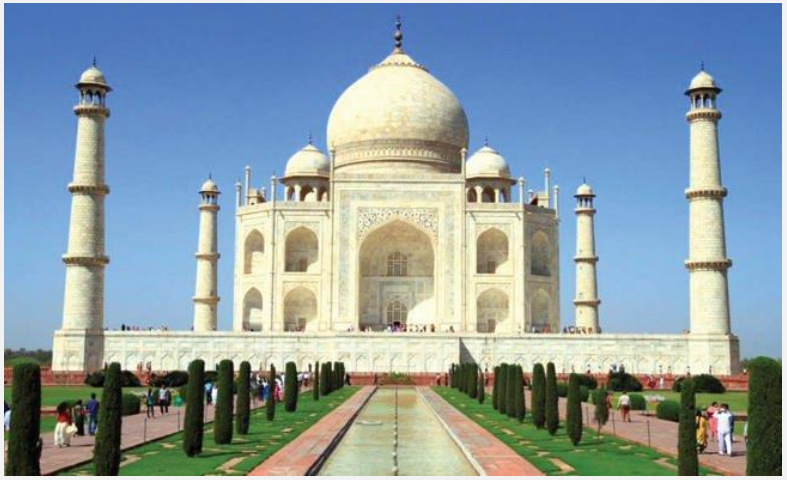


- Meta data refers to additional information associated with a source document, providing context, descriptions, or annotations. In our ongoing example, we can enrich the image of the Taj Mahal with various textual metadata elements, as presented on the following page. These textual pieces allow the retrieval system to find relevant images with text retrieval methods. For example, if the image's description metadata includes the keywords "Taj" and "Mahal", we can directly match it with user queries such as "Taj Mahal".
- However, it is not as easy as it seems. Firstly, we need to gather metadata for the images in the database. How?
  - Manual annotations: human workers inspect each image and provide context, descriptions, categories, tags and other metadata items for the image
  - Automated annotations: technical metadata, such as geo-location, can be captured at the time of taking the image. Additionally, AI workers can analyze an image and extract pre-learned annotations. If the image is embedded in a broader context, such as a web page, that context can yield more information about the image
  - Generative AI: the latest multi-modal transformer models can extract relevant information from a wide range of images, providing high-quality metadata and text descriptions.
- Secondly, annotations obtained by two different workers, whether human or AI, can semantically differ from each other disabling a direct matching approach. Let's consider an example:
  - Worker A adds the following keywords: Taj Mahal, India, iconic building, 17th century
  - Worker B adds the following keywords: religious building, great weather, few people outside, nice

  Both workers have provided accurate annotations, but they differ in semantic levels. When comparing the phrases "Taj Mahal" with "religious building", a retrieval system must consider the relationships between words. In this case, "Taj Mahal" is a more specific term used by workers familiar with the building, while workers who have never seen the Taj Mahal (or an AI not trained to recognize the building) or lack context may opt for the more generic phrase "religious building". Similar relationships appear almost everywhere in natural language: horse ↔ mammal, Matterhorn ↔ mountain, Italy ↔ Europe. As we learned previously, we can address these relationships with natural language processing and the more recent development of embeddings

  Worker B's use of the keyword "nice" expresses a subjective and abstract concept. Obtaining and normalizing such abstract concepts, where two individuals would agree upon them, can be challenging. However, if we can match these abstract concepts with user preferences, we can provide more relevant examples for the user's queries. For example, if a user plans a visit to India and searches for sites with "great" architecture, both "India" and "great" describe abstract concepts which are not present in the pixels alone, but are obtainable from metadata.

- Let's annotate our ongoing example, the picture of the Taj Mahal. In essence, we can consider three types of metadata: generic, specific, and abstract. Furthermore, we can annotate images across various facets as illustrated below. It's worth noting that we can extract certain metadata like "outside", "time/date taken", "sky", or "building" directly from the raw pixel information, regardless of whether a human or AI worker performs the task. However, other information such as "UNESCO", "Mumtaz", "1648" or even "Taj Mahal" requires a human or AI worker who possesses contextual awareness since these details cannot be derived from the pixels alone.



Object Facet	Value
Generic Object Instance	building, water, sky
Generic Object Class	mausoleum, tomb, dome, minaret
Specific Named Object Class	UNESCO World Heritage Site (since 1983)
Specific Named Object Instance	Taj Mahal

Spatial Facet	Value
Generic Location	outside
Specific Location Hierarchy	India, Uttar Pradesh, Agra

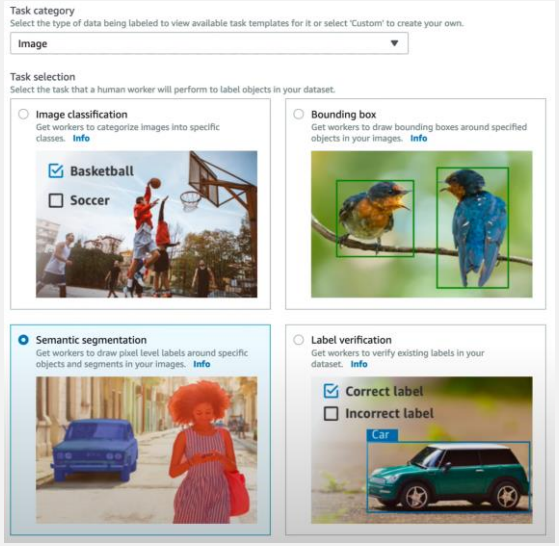
Event / Activity Facet	Value
Generic Event/Activity	tourism, attraction
Specific Event Instance	International World Heritage Expert Meeting on Visual Integrity in 2006

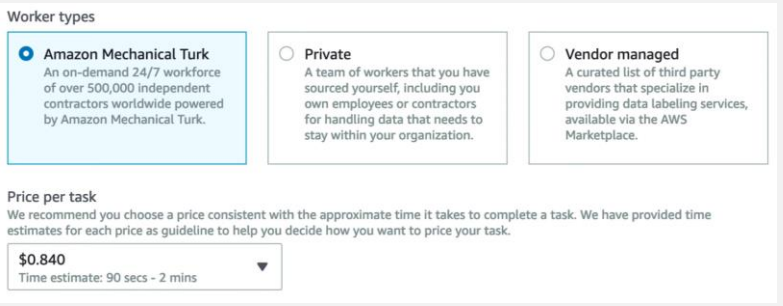
Temporal Facet	Value
Generic Time	summer, daytime
Specific Time	2006 (photo taken)

Contextual Facet	Value
Topic	Indian Architecture
Related Concepts / Objects	Shah Jehan, Mumtaz Mahal, Islam
Abstract Concept	love, death, devotion, remembrance
Context	built in memory of his favorite wife Mumtaz Mahal, by Shah Jehan; completed 1648

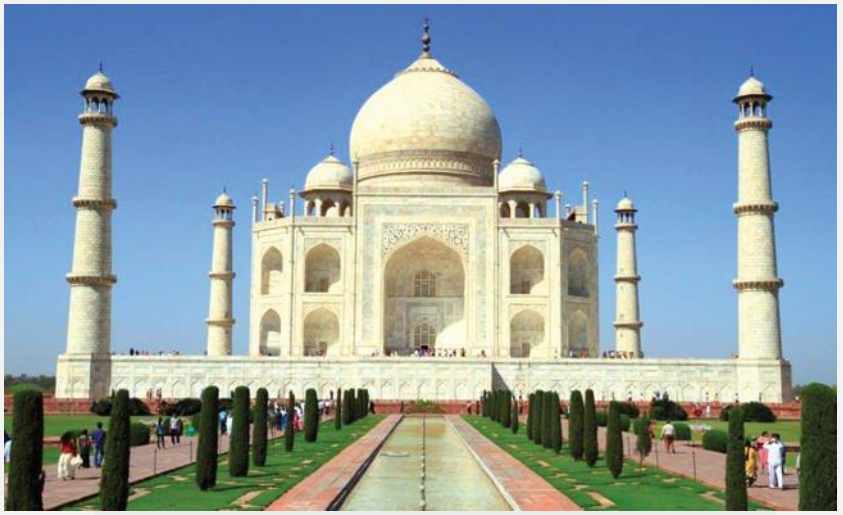
10.2.1 Manual Metadata Creation

- The process of creating metadata using human workers has gained popularity on various machine learning platforms. In supervised learning, labeled data is required to train models, and these labels at the same time generate metadata for images. For instance, Amazon Mechanical Turk offers access to over 500,000 independent contractors who can perform well-defined tasks at specified prices, as depicted in the example below. Similarly, ChatGPT was trained with the help of thousands of workers to assess the quality of answers generated by the AI.
- Annotation or labeling tasks typically cost around $1 and upwards, depending on the complexity of the task and the required domain knowledge. For basic tasks in machine learning, generic labels and descriptions often suffice. However, annotating stock images or categorizing items in a media archive demands more specific labels and extensive domain knowledge, leading to higher annotation costs. By leveraging a global workforce, annotation tasks can be scaled to millions of items at reasonable costs, yielding results within a reasonable time frame. We will see a few examples in one of the upcoming pages.
- In the case of machine learning, the initial investment in training a model can subsequently produce automated labels, as we will explore further in this course.





- The quality and substance of manually created labels can greatly vary depending on the domain expertise of the human workers. In the examples provided below, we can observe two distinct approaches to annotating a picture of the Taj Mahal. On the left side, we have the results of a more generic and concise labeling task, whereas the right side shows a comprehensive analysis and description that demonstrates deep domain knowledge.
- This serves as a good example for the challenges when dealing with manually created metadata such as variations in level of detail, choice of keywords, and depth of domain knowledge. The annotations on the left side may not provide sufficient information to boost the image for queries of the Taj Mahal, while the annotations on the right side are so detailed that they are less likely to align with typical queries for the Taj Mahal.

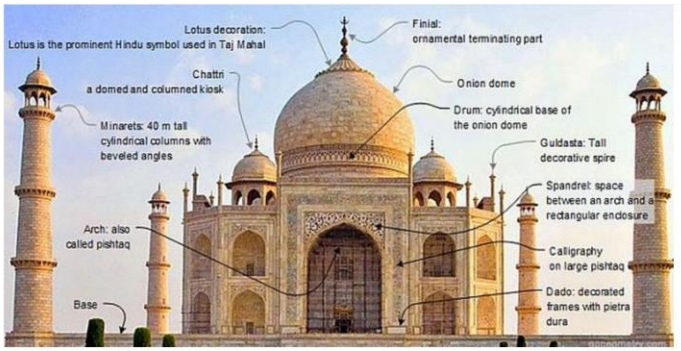


Keywords

Taj Mahal Photos | Agra Photos | Architecture Photos | City Photos | Color Image Photos | Culture of India Photos
Famous Place Photos | Horizontal Photos | India Photos | Indian Ethnicity Photos | International Landmark Photos
Mahal - Palace Photos | Marble - Rock Photos | Mausoleum Photos | Medium Group Of People Photos | Monument Photos
Outdoors Photos | People Photos | See all

Structural Details and Components of Taj Mahal

By: Haseeb Jamal / On: Aug 31, 2017

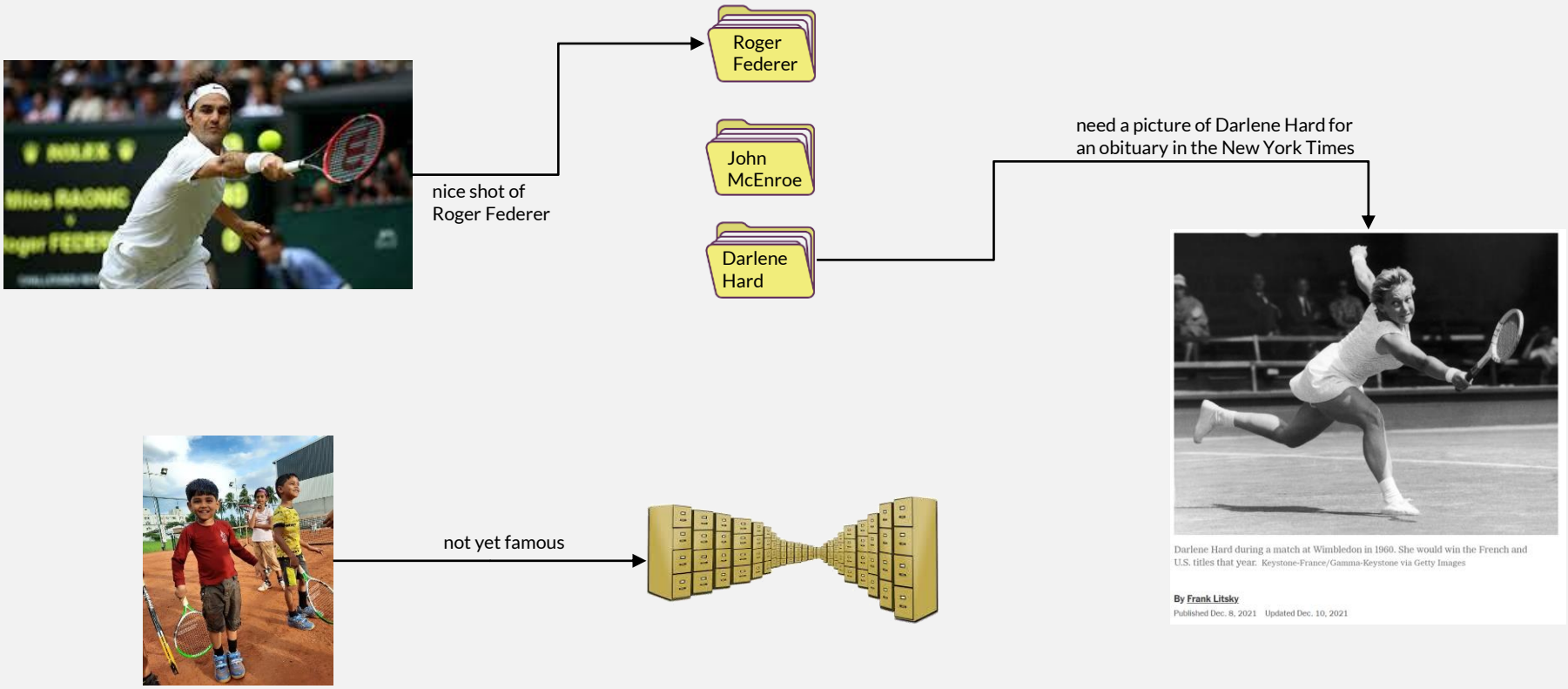


Structural Details of Taj Mahal

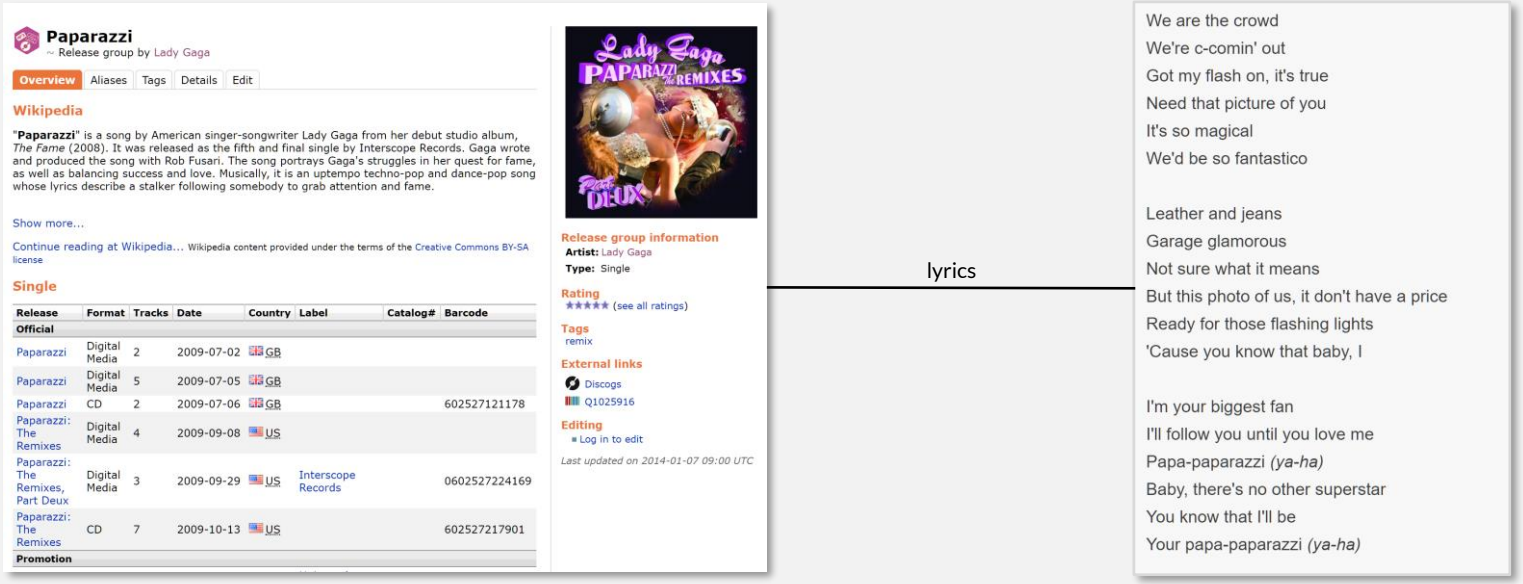
1. On a platform 22' high and 313' square. Each tower is 133 feet tall
   Building is 186 feet high and 70 wide.
2. Corner minarets are 137' tall. Main structure 186' on a side, dome to 187'.
3. The mausoleum is 57 m (190 ft) square in plan.
4. "The central inner dome is 24.5 m (81 ft) high and 17.7 m (58 ft) in diameter, but is surmounted by an outer shell nearly 61 m (200).
5. The Taj stands on a raised, square platform (186 x 186 feet) with its four corners truncated, forming an unequal octagon.

6. The architectural design uses the **interlocking arabesque** concept, in which each element stands on its own and perfectly integrates with the main structure. It uses the principles of self-replicating geometry and symmetry of architectural elements.
7. Its central dome is fifty-eight feet in diameter and rises to a height of 213 feet.
8. It is flanked by four subsidiary domed chambers.
9. The four graceful, slender minarets are 162.5 feet each.
10. The entire mausoleum (inside as well as outside) is decorated with inlaid design of flowers and calligraphy using precious gems such as agate and jasper.
11. The main archways, chiseled with passages from the Holy Qur'an and the bold scroll work of flowery pattern, give a captivating charm to its beauty.
12. The central domed chamber and four adjoining chambers include many walls and panels of Islamic decoration.

- Stock photo services and media company archives maintain concise keyword lists for each image. They also utilize "faceted navigation" which involves categorizing images based on various attributes with pre-defined values such as prominent individuals, locations, brands, or time periods like decades. For instance, sports event photos are examined to identify shots featuring known individuals. Only a limited number of selected shots from each event are annotated for faceted navigation to keep the overall number manageable. This allows users, like journalists, to easily browse through a curated list instead of scanning thousands of pictures when they need an image of a prominent person. However, one drawback of this approach is that acquiring pictures of individuals before they gain prominence is challenging and often relies on lucky discoveries or contributions from the individuals themselves or their entourage.

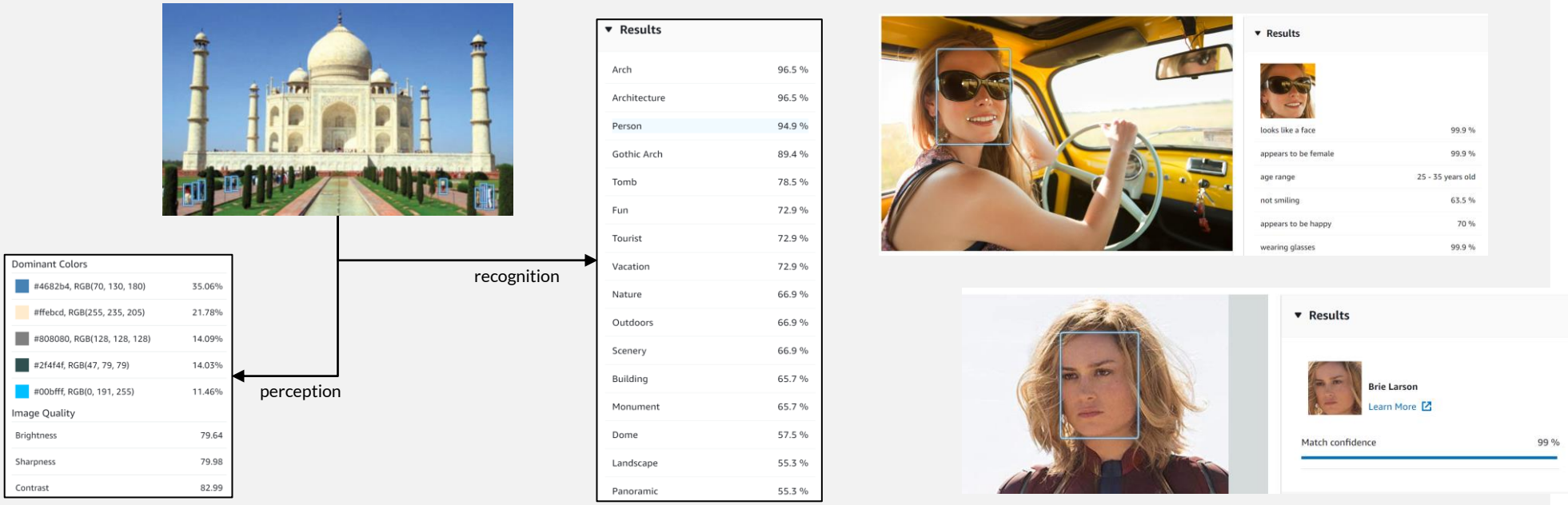


- In domains with a well curated set of items, such as songs or movies, metadata annotations with quality control and consistent structure are available. IMDb is an example of such a database that holds records for movies and episodes from various producers. Each item is annotated with predefined attributes and has relationships with other items. The database is curated by volunteers, actors, crews, and industry executives, and is accessible online in compiled formats. As such, this is an excellent illustration of “scaled-out” metadata gathering.
- MusicBrainz is another good example for a community-maintained open-source encyclopedia of music information. It provides details about artists, albums, songs, and releases. When combined with a lyrics database, music search benefits from a wide range of textual features and factual data that are not obtainable from the raw audio alone.
- With such curated databases, retrieval of information greatly benefits from high-quality annotations. Often, the metadata alone is sufficient to bridge the semantic gap, meaning that the audio data itself is only used in rare cases, such as with Shazam, to retrieve information about the currently playing song. Due to the commercial and community interest in these domains, the additional efforts involved in creating and maintaining the metadata are covered by increased revenues.

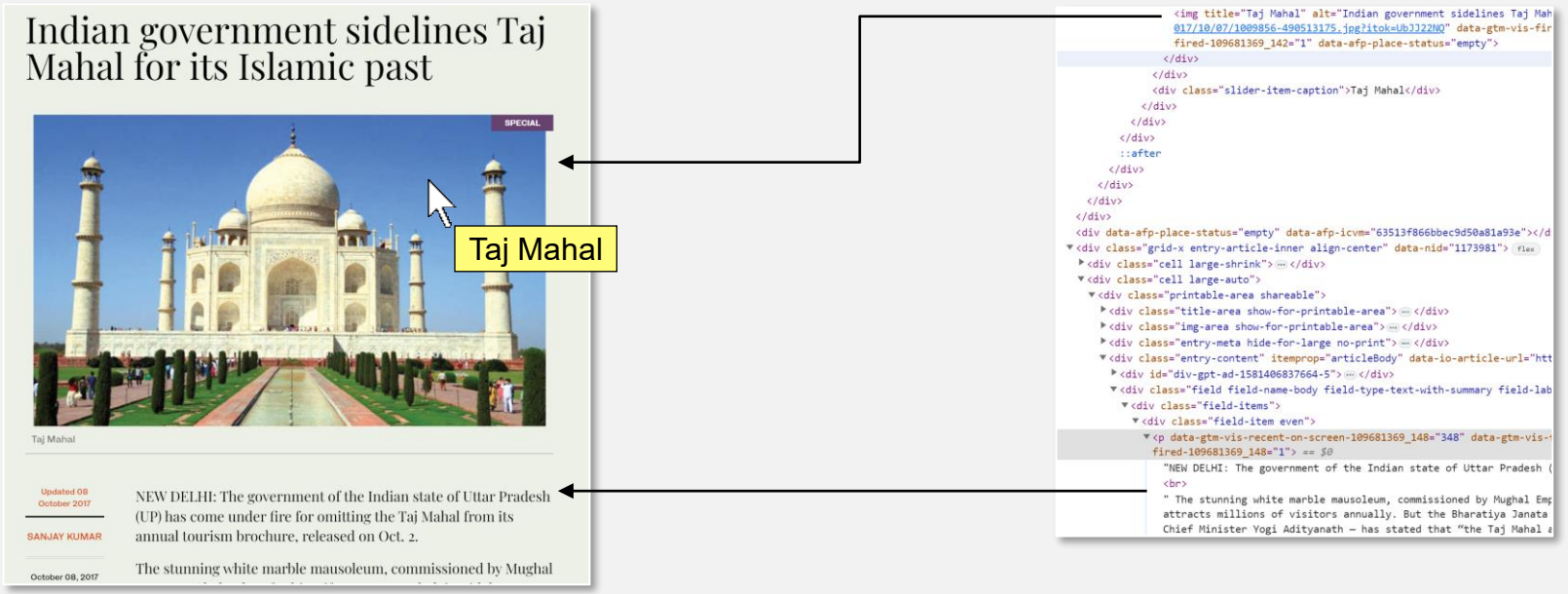


10.2.2 Automated Metadata Extraction

- Annotating arbitrary photos and videos raises challenges due to the absence of a curated reference database for readily obtaining metadata. The costs associated with annotating every single photo and video would far outweigh the added value of having metadata (unless you do it for your photos and videos as fun activity after vacations).
- In such cases, automated annotations and AI-based metadata extraction provide valuable support for retrieval systems. The following examples illustrate the extraction of metadata at different semantic levels:
  - Perception level (left side, lower part): The signal information is processed to capture key aspects that enable comparisons between items based on how humans interpret the signals. This course will provide extensive examples covering various types of multimedia items.
  - Structural level (middle to right side): Machine learning methods analyze the signal information and its context to extract pre-trained metadata items that can be generic, specific, or abstract. Examples include generic object recognition (architecture, person, female, outdoors), specific object recognition (Brie Larson), or abstract concepts that a typical human would recognize (fun, age, happy).



- When documents are embedded on the web, there is a simple yet powerful approach to extracting context, relationship information, and textual metadata:
  - HTML tags such as `<a>` or `<img>` include special attributes that provide descriptions or short annotations for the referenced objects. These attributes can be extracted and used as metadata.
  - The surrounding area on the web page, including title information, text blocks, and captions, can serve as another valuable source of keywords that are likely to overlap with the context of the embedded object. While not always a perfect match, this source often provides sufficiently valuable information and is easily obtainable.
- In the early days of the web, the "surrounding area" referred to the immediate vicinity within the HTML source code. By considering a window of a few dozen tokens before and after the embedded object, most of the relevant keywords could be captured. Additionally, the header sections (`<h1>`, `<h2>`) and title of the web page were useful sources. However, modern web applications utilize advanced scripts and CSS styles that dynamically change data and layout, making the direct neighborhood within the HTML less reliable for capturing relevant keywords. As shown in the illustration on the right-hand side, the distance between the image and the text paragraph can be large in terms of both text position and hierarchical position due to CSS instructions.



- Advanced web-based metadata extraction considers the visual proximity between embedded objects and text blocks, even though it comes with higher extraction costs. Here's how it works:
  - The web page is rendered in a browser and we identify all objects and text elements of interest
  - Each DOM element has a bounding box, accessible through the `getBoundingClientRect` method which provides on-screen distances between objects
  - We can scan for visual, CSS, or textual cues to eliminate or weigh down text blocks that are not directly relevant such as sidebars or other articles
  - Distances and cues provide proximity weights for the keywords in text blocks that we can use to describe the context of the embedded object

```
function getPositionAtCenter(element) {
    const {top, left, width, height} =
                element.getBoundingClientRect();
    return {
        x: left + width / 2,
        y: top + height / 2
    }
}

function getDistanceBetweenElements(a, b) {
    const apos = getPositionAtCenter(a);
    const bpos = getPositionAtCenter(b);

    return Math.hypot(apos.x - bpos.x,
                      apos.y - bpos.y)
}

getDistanceBetweenElements(image, text)
```

- Multi-modal transformer models can merge text and image data:
  - The encoder model can map text and images into the same conceptual space, making it possible to directly compare the two representations.
  - The decoder model can use encoded image data and a prompt to create text that matches the image.
  - By using various prompts, we can gather different information and adjust the meta-data we want to create and how we want it to look.
- Example: Picture of the iconic Matterhorn
  - We start by using a prompt to create a detailed description of what can be seen in the image.

```
Prompt: Please provide a detailed description of the
image you see. Include information about:
- The main subject or focus of the image
- The setting or background
- Colors and lighting
- ...
```

  - Then, we use a prompt to create a JSON document that includes various features extracted from the image content.

```
Prompt: Analyze the given image and provide a JSON
output with the following information:
{...definition omitted here...}
Please ensure that: 1) The "headline" is concise and
accurately represents the image's main subject, 2)
"keywords" include relevant terms describing the ...
```

- Note: You can create better prompts by using meta-prompts, which generate the prompt based on your basic specifications.

```
The main subject of this image is a majestic, snow-covered
mountain peak, which appears to be the iconic Matterhorn in the
Swiss Alps. The mountain dominates the frame, its distinctive
pyramid shape rising dramatically against a clear blue sky. The
setting is a high-altitude alpine environment, with the peak
surrounded by other snow-capped mountains and glaciers visible in
the lower portions of the image. The background is primarily
composed of a vivid blue sky with a few wispy clouds. The colors
in the image are striking, with the brilliant white of the snow
contrasting sharply against the deep blue of the sky. The
lighting appears to be natural sunlight, creating a play …
```

```
{
   "headline": "Majestic Matterhorn Pierces Blue Sky",
   "keywords": ["mountain", "peak", "snow", … ],
   "named_entities": ["Matterhorn"],
   "dominant_colors": ["blue", "white", "gray"],
   "dominant_forms_shapes": ["triangle", "pyramid"],
   "extracted_text": "",
   "people_present": false,
   "is_selfie": false,
   "is_outdoors": true,
   "is_text_image": false
}
```

- The trustworthiness of metadata is a subject of concern, as highlighted by Cory Doctorow's "seven insurmountable obstacles" to achieving a meta-utopia. These obstacles include:
  - **People lie**: Unscrupulous content creators may publish misleading or dishonest metadata to redirect traffic
  - **People are lazy**: Many content publishers lack the motivation to thoroughly annotate their published content
  - **People are stupid**: Not all content publishers possess the necessary intelligence to effectively catalog their produced content
  - **Mission impossible—know thyself**: Inadvertently misleading metadata can be published by content creators
  - **Schemas aren't neutral**: Classification schemes are subjective and can introduce biases
  - **Metrics influence results**: Competing metadata standards bodies may never reach an agreement
  - **More than one way to describe it**: Resource description is subjective, and different perspectives exist.

- With generative AI, we can add an 8th law that involves extracting keywords and meta-data.
  - **Models hallucinate**: Hallucinations happen when a language model predicts words that sound plausible but are not true. Because it selects the most likely next word from patterns rather than from facts, it can give incorrect information with confidence.
- However, we should not disregard metadata entirely. Instead, it is important to exercise caution and carefully evaluate the information it provides. High-quality metadata, as seen in platforms like IMDb and MusicBrainz, can be exceptionally valuable. Observational metadata obtained through web crawling can also be beneficial, especially when the system is designed to resist manipulation. For example, the Google web search engine gives higher importance to anchor texts provided by others linking to a page rather than relying solely on the keywords provided by the content owner. However, even these advanced approaches can potentially be manipulated as was successfully demonstrated with the so-called Google-bomb.

10.3 Performance Measure for Structural Features

- Previously, we covered performance measures such as precision and recall, which evaluate how well a system ranks documents for a given query. For tasks that use structural features, such as extracting labels from images, we want to evaluate how well a classifier performs. We consider both binary and multi-class tasks with assignments of labels.
- 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 transposed 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.

		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)

Accuracy (ACC)

Error Rate (ERR), Misclassification Rate

- 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.
  - The test shows 91% accuracy, which looks good at first. However, precision (positive predictive value) is only 10%. Of 200 positive test results, only 20 people actually had cancer. That means 180 people would be wrongly told they have cancer if we follow the test results. The high accuracy comes mainly from the large number of true negatives. In fact, a test that always returns negative would reach 99% accuracy here, producing 2000 true negatives and 0 true positives.
  - 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

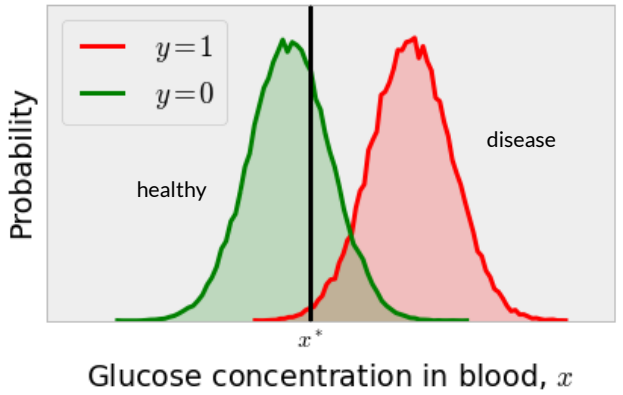
- By collapsing the classes "Woman" and "Child", we can delve deeper into the performance of the example:
  - For the class "Woman," we observe a high specificity (correctly dismissing the class) but low precision and sensitivity values. A closer examination of the prevalence also indicates that the 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

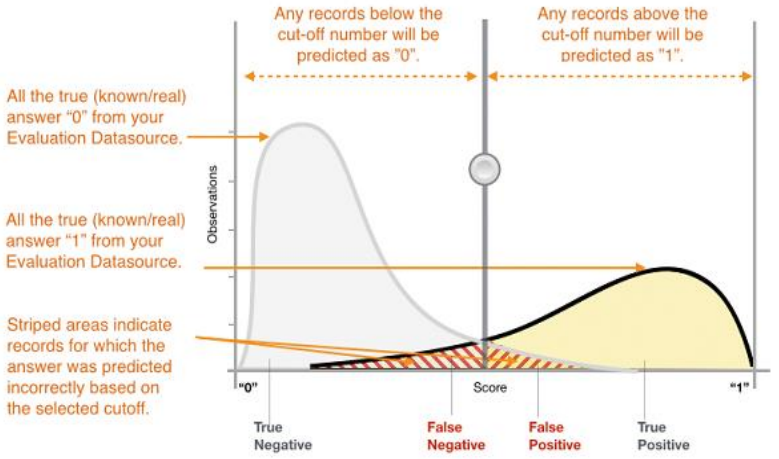
		Actual Class		
	Total Population	Woman (P=20)	Not a Woman (*N=80*)	
Recognized Class	Woman *(19)*	**True Positive** (*TP=13*)	**False Positive** (*FP=6*)	$PPV = \frac{13}{19} = 68\%$ (precision)
	Not a Woman (*81*)	**False Negative** (*FN=7*)	**True Negative** (*TN=74*)	$NPV = \frac{74}{81} = 91\%$
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

10.3.1 Optimizing Hyperparameters

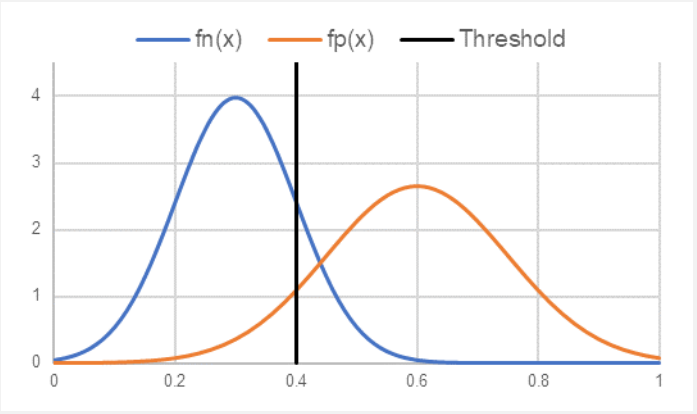
- In many binary classification algorithms, the output is not a direct assignment to a class, but rather a prediction score. Take the example on the top right, where there is a clear correlation between glucose concentration and the presence of diabetes:
  - Plotting the distribution of concentration 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)

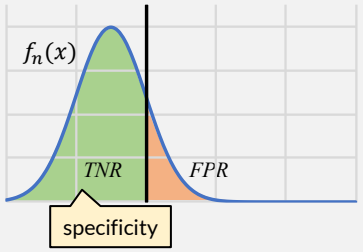




source: https://docs.aws.amazon.com/machine-learning/latest/dg/binary-classification.html

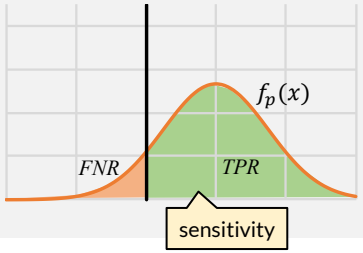
- Let's have a closer look at how the setting of the threshold impacts the results of the prediction:
  - The plot on the bottom left shows the actual distribution of scores between 0 and 1 for the negative class as $f_n(x)$ and for the positive class as $f_p(x)$. When we introduce a threshold T, we 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





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

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

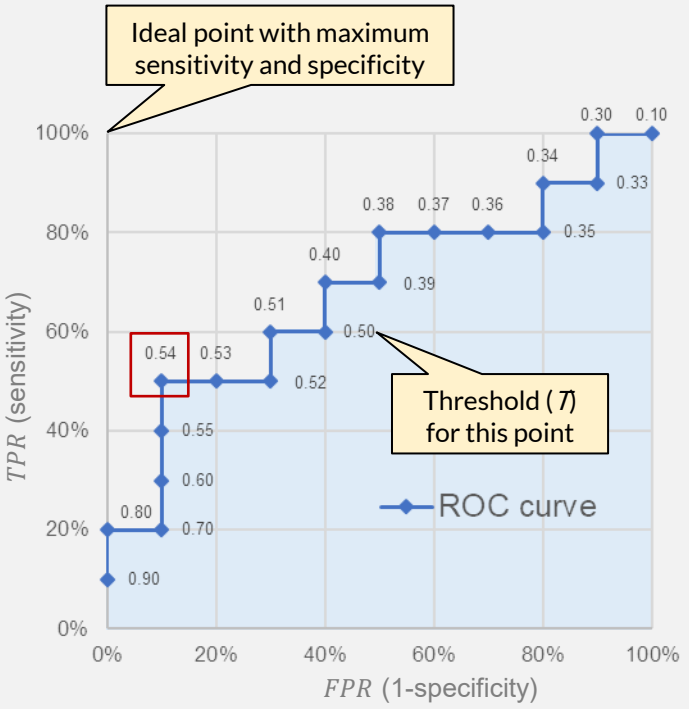


$$FNR(T) = \int_{-\infty}^{T} f_p(x)\,dx$$

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

- 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%



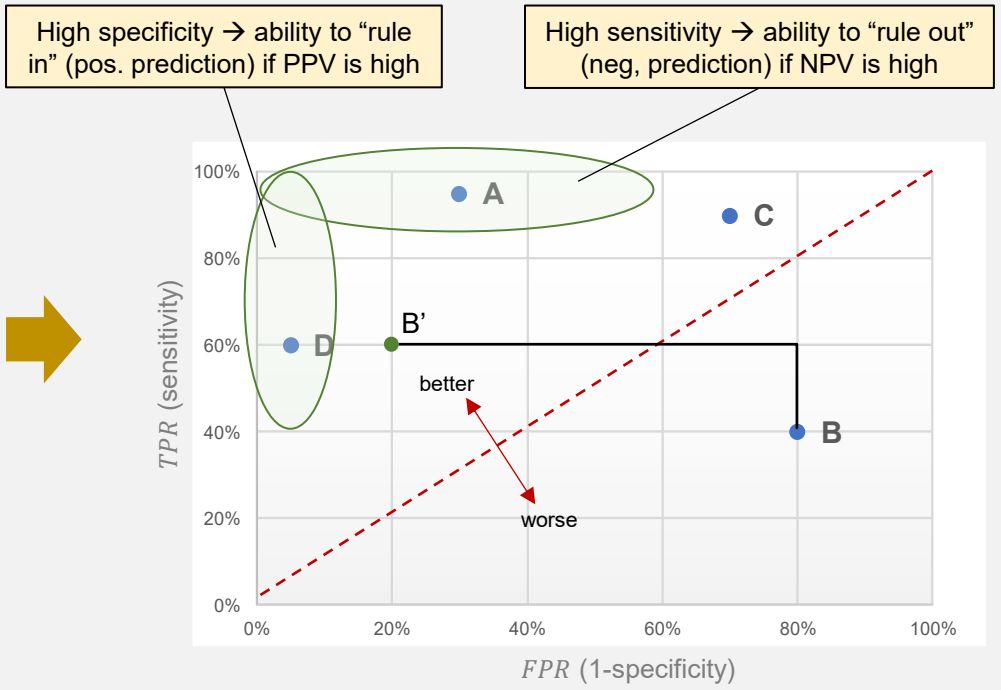
- Interpretation of the ROC-curve: consider the 4 methods **A, B, C, D** and their confusion matrices below:
  - **A** lies in an are with high sensitivity. If the NPV is also high (consider the prevalence), then **A** can effectively "rule out" (negative predictions). **A** would be a good candidate 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	
$TP = 40$	$FP = 80$
$FN = 60$	$TN = 20$
$TPR = 40\%$	$FPR = 80\%$
$PPV = 33\%$	$NPV = 25\%$
$ACC = 30\%$	

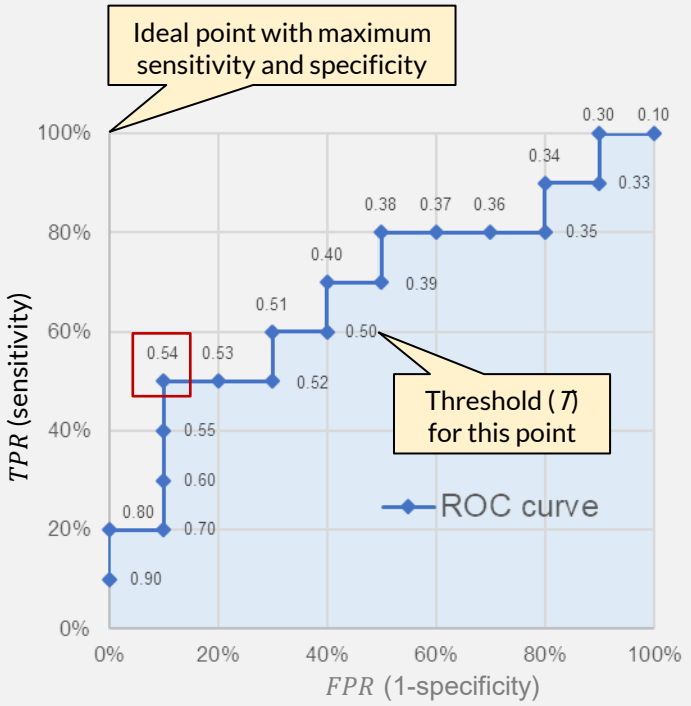
C	
$TP = 90$	$FP = 70$
$FN = 10$	$TN = 30$
$TPR = 90\%$	$FPR = 70\%$
$PPV = 56\%$	$NPV = 75\%$
$ACC = 60\%$	

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%



10.4 Literature and Links

- Napolify, **TikTok's algorithm: what are the latest changes?**, https://napolify.com/blogs/news/algorithm-tiktok
- enrichlabs.ai, **Tiktok Algorithm 2025**, https://www.enrichlabs.ai/blog/tiktok-algorithm-2025
- 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
- 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
  - **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