# LAB 2 – Basic Information Retrieval Models

## Objective

In this Lab, you are requested to implement and compare different basic Information Retrieval models, namely:

1. Classic Boolean Model
2. Fuzzy Boolean Model
3. Extended Boolean Model
4. Vector Space Model (VSM)

The goal is to understand how each model represents documents, interprets user queries, and computes relevance between them.

## 1. Prerequisites

Before starting this Lab, ensure you have successfully completed Lab 1, where you:

- Indexed the document collection $D_1$ to $D_6$.
- Tokenized the text using this specified regular expression

```
tokenizer = RegexpTokenizer(
    r'(?:[A-Za-z]\.)+'                 # Abbreviations like D.Z.A
    r'|[A-Za-z]+[\-@]\d+(?:\.\d+)?'  # Words combined with numbers, e.g., data-1
    r'|\d+(?:[\.\,\-]\d+)*%?'        # Numbers with decimals, separators, or percentages
    r'|[A-Za-z]+'                      # Simple words (alphabetic)
)
```

- Removed stop words and applied Porter stemming (only PorterStemmer will be used for this lab).
- Calculated Term Frequency (TF) and TF–IDF weights according to the formulas defined in Lab 1.
- Built both **Document–Term** and **Inverted Index** structures in the following formats

  **Document-Term File**
  <Document number> <Term> <Frequency> <Weight>
  **Inverted Index File**
  <Term> <Document number> <Frequency> <Weight>

The document collection ($D_1$.txt – $D_6$.txt) and their corresponding Document-Term and Inverted Index files generated in Lab 1 will be reused in this lab.

## 2. Implementation

### Task 1 – Boolean Models

Implement the following Boolean-based models using the mathematical formulations presented in the course:

1) **Classic Boolean Model** 2) **Fuzzy Boolean Model** 3) **Extended Boolean Model**

**Test Query**

**q = (query AND reformulation) OR (Language AND model)**

***Parentheses define precedence***

**Expected Steps**

1. Preprocess the query using the same pipeline as for documents (tokenization, stop word removal, and stemming).
2. Parse the Boolean expression into logical operations.
3. For the **Classic Boolean Model**: retrieve only the documents that strictly satisfy the Boolean condition.
4. For the **Fuzzy** and **Extended Boolean Models**:
   - Compute partial degrees of relevance for each document according to the model's equations given in lecture notes.
   - Rank documents by their computed degree of match with the query.

## Task 2 – Vector Space Model

Implement the Vector Space Model (VSM) using the TF–IDF weighted representation of documents and queries. Each document and query should be represented as a term-weight vector.

Compute document–query similarity using three different similarity measures: **Inner Product Similarity, Cosine Similarity and Jaccard Similarity (formulas provided in lecture notes).**

**Test Queries**

Use the following queries to test your implementation:

**q1: large language models for information retrieval and ranking**
**q2: LLM for information retrieval and Ranking**
**q3: query Reformulation in information retrieval**
**q4: ranking Documents**
**q5: Optimizing recommendation systems with LLMs by leveraging item metadata**

## Expected Steps

1. Load the Document–Term and Inverted Index files generated in Lab 1.
2. Preprocess each query (tokenization, stop word removal, stemming).
3. For each similarity measure (Inner Product, Cosine, Jaccard):
   - Compute the similarity score between the query and every document.
     **For queries, use a binary weighting scheme: assign a weight of 1 to each term that appears in the query and 0 to terms that do not appear.**
   - Rank the documents in descending order of similarity.
   - Display the ranked documents for each query.