Vectorization

- **Vectorization** is the process of converting **raw data (text, images, signals, or structured data)** into **numerical vectors** so that they can be processed by **machine learning and statistical models**
- Each vector represents **meaningful characteristics (features)** of the original data
- Enables **distance computation, similarity measurement, and model training**

- **Methods of Vectorization**
- **1. Text Vectorization**
- Bag of Words (BoW)
- TF-IDF
- N-grams
- Word Embeddings (Word2Vec, GloVe, FastText)
- Contextual Embeddings (BERT, GPT embeddings)

Bag of Words (**BoW**)

- **Bag of Words** is a **text vectorization technique** that represents documents using **word frequency**
- Ignores **grammar, word order, and context**
- Vocabulary is created from **unique words** across the corpus
- Each document is represented as a **fixed-length vector**
-
- **How BoW Works**
- Collect all unique words → **Vocabulary**
- Count word occurrences in each document
- Form a **document–term matrix**
-
- **Example**
- Sentence: *"I love data science"*
- Vocabulary: [I, love, data, science]
- Vector: **[1, 1, 1, 1]**
-

- **Advantages**
- Simple and easy to implement
- Works well for **basic text classification**
- Interpretable features
-

- **Limitations**
- High dimensional & sparse
- No semantic meaning
- Word order is lost

TF–IDF (Term Frequency–Inverse Document Frequency)

- **TF–IDF** is a **text vectorization technique** that measures how **important a word is to a document** relative to a collection of documents
- Reduces the weight of **common words** and increases the weight of **rare but informative words**
- Widely used in **information retrieval and text classification**

Components

1. Term Frequency (TF)

- Measures how often a word appears in a document
- $\mathrm{TF}(t,d) = \frac{\text{Count of term } t \text{ in } d}{\text{Total terms in } d}$

2. Inverse Document Frequency (IDF)

- Measures how rare a word is across documents
- $\mathrm{IDF}(t) = \log \frac{N}{1+df(t)}$

3. TF–IDF Score

- $\mathrm{TF\text{–}IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t)$

Word2Vec

- **Word2Vec** is a **neural-network–based word embedding technique** that represents words as **dense, low-dimensional vectors**
- Captures **semantic and syntactic relationships** between words
- Words with similar meanings have **similar vector representations**
-

- **How Word2Vec Works**
- Trains on a large text corpus using a **sliding context window**
- Learns word vectors by predicting:
  - **CBOW (Continuous Bag of Words):** predicts target word from context
  - **Skip-Gram:** predicts surrounding context from target word
  -

- **Key Characteristics**
- Dense vectors (e.g., 100–300 dimensions)
- Preserves **semantic similarity**
  king − man + woman ≈ queen
- Context-based learning
.

- **Advantages**
- Captures meaning better than BoW & TF-IDF
- Low dimensional and efficient
- Works well for many NLP tasks