**TECHNIQUES USED TO CONVERT WORDS TO NUMBERS**

**ONE-HOT ENCODING:**

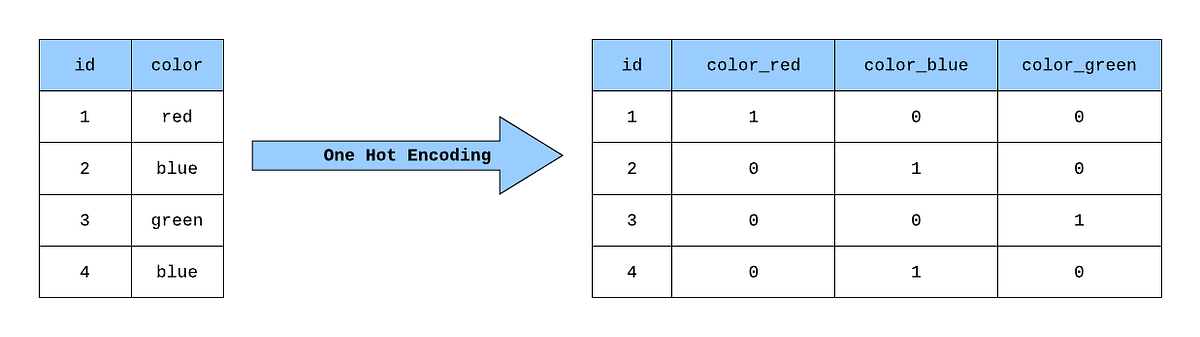
* One-Hot Encoding is the simplest way to convert words into numbers.
* The main rule here is that the size of the vector = size of the vocabulary(all unique elements in the dataset).
* One hot Encoding gives every word a unique identity.
* Here only one element is 1 and rest is 0.

**ADVANTAGES:**

* One Hot Encoding represents categorical data numerically.
* One Hot Encoding can help to improve the performance of machine learning models by treating categorical variables as single entity.

**DISADVANTAGES:**

* It can lead to increased dimensionality as a separate column is created for each category in the variable. This can make the model more complex and slow to train.
* If the sample size is small and there are many categories, the model memorizes the data and not understand it.



**BAG OF WORDS(BoW):**

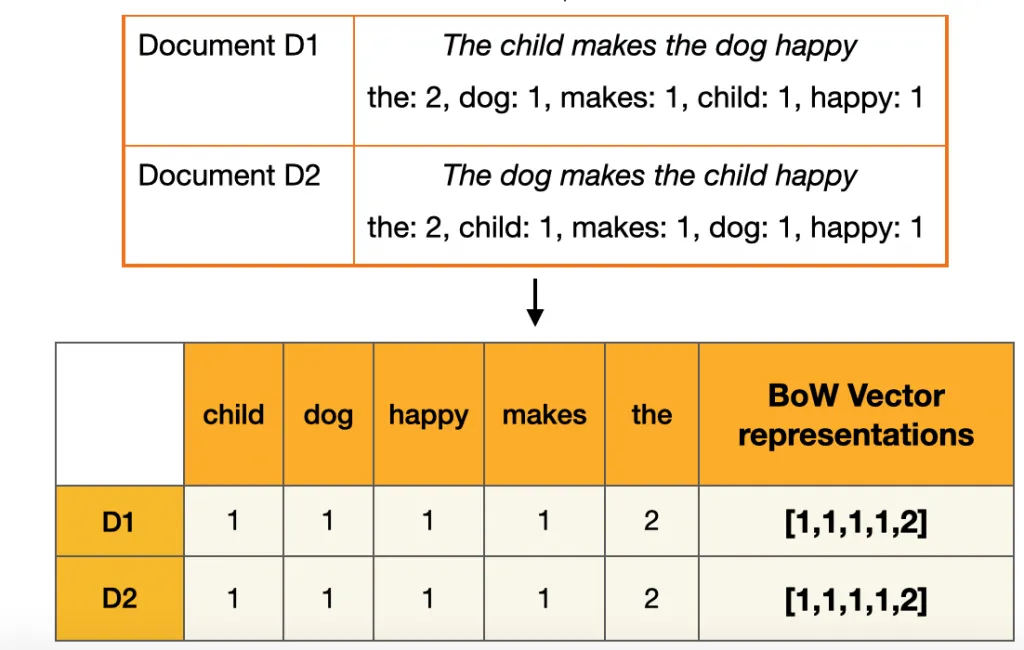
* The bag of words model is a way to represent as numerical feature vector by counting the occurrence of the words.
* Here bag means all words of the dataset irrespective of order.
* In this model only the frequency of word matters.
* The model ignores grammar , word order and case sensitivity
* Working: This model is represented as a binary matrix where each row corresponds to a sentence and each column represents one of the top N frequent words. A 1 in the matrix shows that the word is present in the sentence and a 0 shows its absence.

**ADVANTAGES:**

* Simplicity: It is easy to implement and computationally efficient.
* Versatility: It can be used for various NLP tasks such as text classification, sentiment analysis and document clustering.

**DISADVANTAGE:**

* Loss of Context: It ignores word order and context which means it might miss important relationships between words.



**Term frequency-inverse document frequency (TF-IDF):**

* Term Frequency (TF): Measures how often a term (word) appears in a document.

TF(T,D) = frequency of T in document d/total number of terms in D.

* Inverse Document Frequency (IDF): Measures the importance of a term across a collection of documents.

IDF(T,d) = log(Total number of documents(d)/Number of documents with T).

* The higher the TF-IDF score for a term in a document, the more important that term is to that document within the context of the entire corpus.
* The TF-IDF(t,d,D) = TF(T,D)\*IDF(T,d).
* TF-IDF gives us a numerical weight for each word in each document that reflects:
* Importance inside the document (TF).
* Uniqueness across the corpus (IDF).
* High TF + High IDF → very important word for that document.
* High TF + Low IDF → common word, not very useful (e.g., “the”, “is”).
* Low TF + High IDF → rare word, but not frequent enough in this doc to matter.

**Applications:**

* It ranks words by importance making it possible to automatically highlight key terms, generate document tags.
* By converting documents into numerical vectors TF-IDF enables comparison and grouping of related texts.

**WORD EMBEDDINGS:**

* A word embedding is a way of representing words as dense numerical vectors (usually 50–300 dimensions), where:
* Relationships between words are captured mathematically**,** unlike one-hot vectors (sparse, huge, no meaning).
* It allows words with similar meanings to have a similar representation. Thus, Similarity can be assessed based on Similar vector representations.