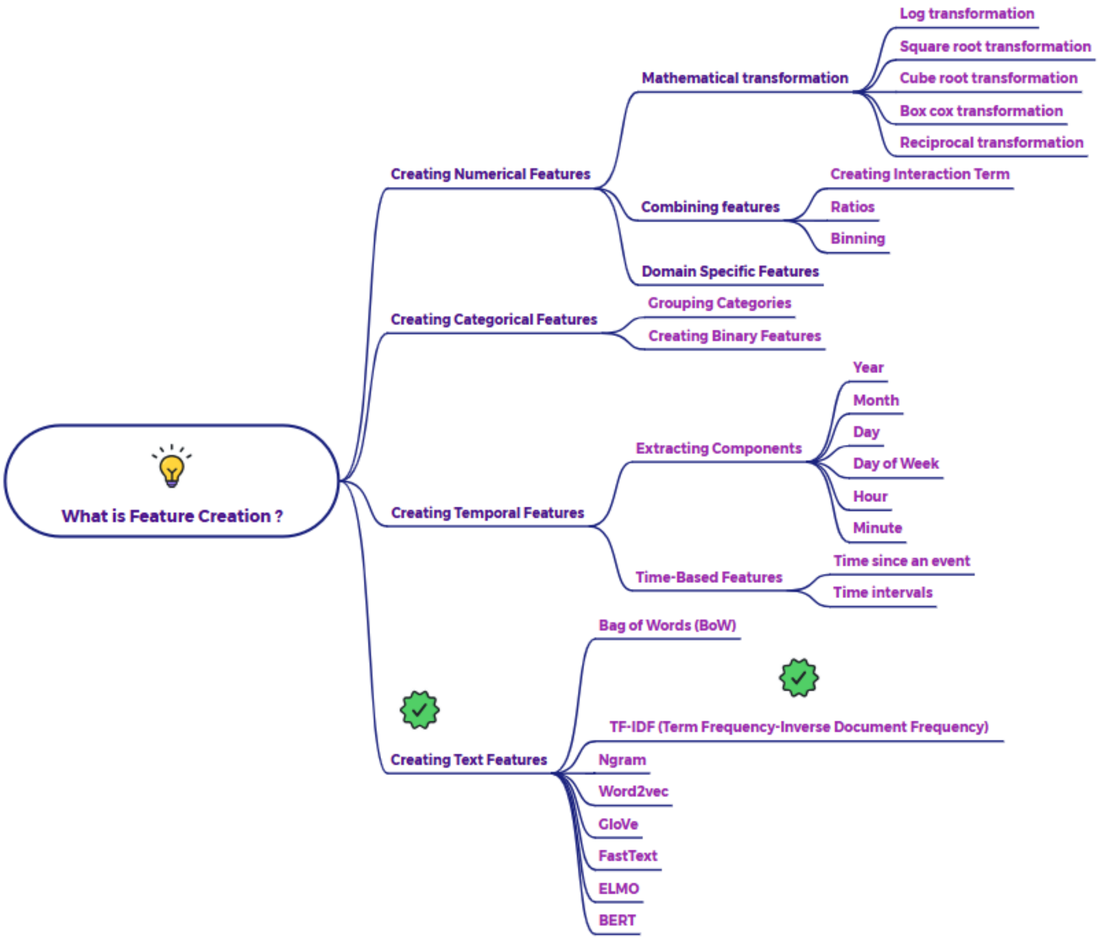
Explain TF-IDF with an example



TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is a technique used in natural language processing (NLP) to represent text data numerically, similar to Bag of Words (BoW). However, TF-IDF improves upon BoW by not only considering the frequency of words in a document but also their importance in the entire corpus (collection of documents). It's also a "frequency-based" method.

How it works:

TF-IDF assigns a weight to each word in a document. This weight reflects how important a word is to that document in the context of the entire corpus. Words that appear frequently in a document but rarely in other documents are considered more important.

TF-IDF consists of two parts:

1. **Term Frequency (TF):**
    - This measures how often a word appears in a document.
    - It's the same as the word frequency used in BoW.
    - TF(t, d) = (Number of times term t appears in document d) / (Total number of terms in document d)
2. **Inverse Document Frequency (IDF):**
    - This measures how rare a word is across the entire corpus.
    - It gives less weight to words that appear in many documents and more weight to words that appear in only a few.
    - IDF(t) = loge(Total number of documents / Number of documents containing term t)

The TF-IDF weight of a term t in document d is calculated by multiplying its TF and IDF values:

$$TF = \frac{Number\ of\ times\ a\ word\ "X"\ appears\ in\ a\ Document}{Number\ of\ words\ present\ in\ a\ Document}$$

$$IDF = log\left(\frac{Number\ of\ Documents\ present\ in\ a\ Corpus}{Number\ of\ Documents\ where\ word\ "X"\ has\ appeared}\right)$$

$$TF\ IDF\ =\ TF * IDF$$

Detailed Example:

Let's use the same three documents from the BoW example:

- Document 1: "The quick brown fox jumps over the lazy dog."
- Document 2: "The dog is happy."
- Document 3: "A quick brown fox."

1. **Preprocessing**: (Same as BoW example)
    - Document 1: "quick brown fox jumps lazy dog"

- Document 2: "dog happy"
- Document 3: "quick brown fox"

2. **Vocabulary**: (Same as BoW example)
   - {"quick", "brown", "fox", "jumps", "lazy", "dog", "happy"}
3. **Calculations**:
   - **Term Frequency (TF):**

     For example, for Document 1:

     TF("quick", Document 1) = 1/6 ("quick" appears once in a document of 6 words)

     TF("dog", Document 1) = 1/6

     For Document 2:

     TF("dog", Document 2) = 1/2

     TF("happy", Document 2) = 1/2

     For Document 3:

     TF("quick", Document 3) = 1/3

     TF("fox", Document 3) = 1/3
   - **Inverse Document Frequency (IDF):**

     IDF("quick") = loge(3/2)

     IDF("brown") = loge(3/2)

     IDF("fox") = loge(3/2)

     IDF("jumps") = loge(3/1) = loge(3)

     IDF("lazy") = loge(3/1) = loge(3)

     IDF("dog") = loge(3/2)

     IDF("happy") = loge(3/1) = loge(3)

- **TF-IDF:**

  TF-IDF("quick", Document 1) = TF("quick", Document 1) * IDF("quick") = (1/6) * loge(3/2)

  TF-IDF("dog", Document 1) = (1/6) * loge(3/2)

  TF-IDF("dog", Document 2) = (1/2) * loge(3/2)

  TF-IDF("happy", Document 2) = (1/2) * loge(3)

  TF-IDF("quick", Document 3) = (1/3) * loge(3/2)

  TF-IDF("fox", Document 3) = (1/3) * loge(3/2)

4. **Document Vectors**: The TF-IDF vectors would contain these calculated TF-IDF values for each word in each document. The vectors will have the same dimensions as in the BoW example, but the values will be different.

Key Differences from BoW:

- TF-IDF weighs words, giving more importance to rare words.
- BoW only considers word frequency.

TF-IDF is a very common technique for converting text to numerical data, and it's often used in information retrieval, text classification, and other NLP tasks.