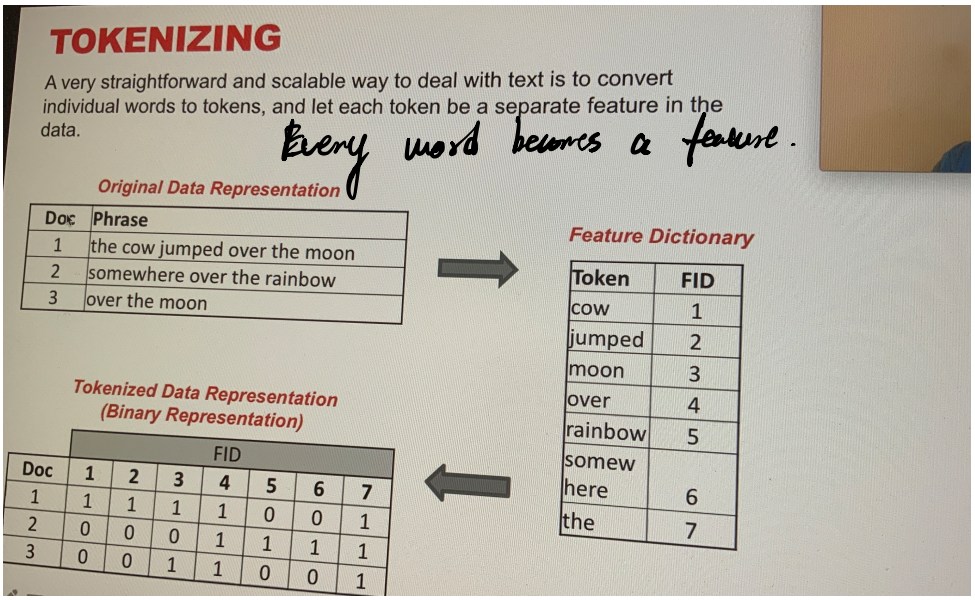
Text Processing:

Tokenization:



TF-IDF = Term Frequency × Inverse Document Frequency.

TF measures how many times a particular token appears in a particular document, and gives an indication of how important that token is to that document.

$$TF(term, Doc) = \sum_{word \in Doc} I(term == word)$$

IDF measures how often a word appears across all documents. It penalizes those words that appear frequently

$$IDF(term, Corpus) = \log \frac{|corpus|}{\sum_{doc \in Corpus} I(term \in Doc)}$$

Word Embeddings:

Represent a word by a dense vector such that words with similar meaning or semantic are close to each other in the vector space. (measured by cosine similarity)

Naive Bayes:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

$$P(y|x_1 x_2 \cdots x_m) = \frac{P(y)\,P(x_1 x_2 \cdots x_m|y)}{P(x_1 x_2 \cdots x_m)}$$

$$= \frac{P(y)\cdot P(x_1|y)P(x_2|y)\cdots P(x_m|y)}{P(x_1 x_2 \cdots x_m)}$$

$$= \frac{P(y)\cdot \prod_{i=1}^{m} P(x_i|y)}{P(x_1 \cdots x_m)}$$

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}}\; P(y)\cdot \prod_{i=1}^{m} P(x_i|y)$$

if binary, this will need $2n$ parameters.

$+2$ prior parameters.

Spam Detection:

Deliberately put some spam emails to inbox to avoid the negative feedback loops caused by selection bias.

ex of selection bias → negative feedback loop:

- If 100% of African princess related email go to the spam email. we will never learn this feature. when we train our model because we are using user's feedback on Inbox email.