# AIML428

- Presentations this Thursday
  - Names: Adam, Sam, Thomas, Jessie, Magnus, Nicholas
  - Submit PPT slides using online submission system
  - Make sure your slides work in the lecture room
  - 6 minutes talk +2 mins question times
- Peer review
  - Download the excel file from assignment page
  - 1-10, 1 is very poor, 10 is excellent
  - Submit the excel file for “peer review” in the week of presentation

## Main topic for today

- Text Representation
  - TF
  - IDF
  - TF.IDF implementation in Python

## An example

- Doc1: this is the first document
- Doc2: this document is the second document
- Doc3: and this is the third one
- Doc4: is this the first one

## Text representation: what features?

- Bag-of-words model
- Each unique word (stemmed) is a feature
- Build a vocabulary of all documents
- if the word is present, 1, otherwise, 0

- An example
    - Doc1: this is the first document
    - Doc2: this document is the second document
    - Doc3: and this is the third one
    - Doc4: is this the first one

## Term weights

Some words appear more than once

- The more often a word occurs in a document, the better that term is in describing what the document is about

- An example
  - Doc1: this is the first document
  - Doc2: this document is the second document
  - Doc3: and this is the third one
  - Doc4: is this the first one

## Term weights: count (Term Frequency)

If a word appear more than once, use count as the value instead of 1

An example

Doc1: this is the first document

Doc2: this document is the second document

Doc3: and this is the third one

Doc4: is this the first one

| this | is | the | first | document | second | and | third | one |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 2 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

## But some words are more important

- Term weights should reflect the estimated importance of each term

- The more often a word occurs in a document, the better that term is in describing what the document is about

But …

- Terms that appear in many documents in the collection are not very useful for distinguishing documents
  - Document frequency
  - inversed

## Term weights: TF*IDF

Many variants, but normally

- Term frequency TF = count
  - Number of times term t appears in a document
- the Inverse Document Frequency (IDF),
  - IDF = 1+ ln((N+1)/(DF+1))
  - N: Total number of documents
  - DF: Number of documents with term t in it
- Term weight W = TF * IDF

- Each document is a vector, the weights are normalised to [0,1]
  - Each value divided by the L2 norm of the vector
  - The L2 norm is calculated as the square root of the sum of the squared vector values.

## TF-IDF Vector representation

Doc1: this is the first document

Doc2: this document is the second document

Doc3: and this is the third one

Doc4: is this the first one

| Idf: 1 | 1 | 1 | 1.5 | 1.5 | 1.9 | 1.9 | 1.9 | 1.5 |
|---|---|---|---|---|---|---|---|---|
| this | is | the | first | document | second | and | third | one |
| 0.3 | 0.3 | 0.3 | 0.5 | 0.5 | 0 | 0 | 0 | 0 |
| 0.2 | 0.2 | 0.2 | 0 | 0.7 | 0.4 | 0 | 0 | 0 |
| 0.2 | 0.2 | 0.2 | 0 | 0 | 0 | 0.5 | 0.5 | 0.4 |
| 0.3 | 0.3 | 0.3 | 0.5 | 0 | 0 | 0 | 0 | 0.5 |

## Python on my laptop

- Installed with anaconda
  - https://www.anaconda.com/distribution/
  - Windows
- The package
  - Python 3.9.7
  - Python common libraries: numpy, pandas, matplotlib, scikit-learn
  - Jyputer Notebook: IDE, good for teaching and beginners
  - Spyder: IDE for more advanced
  - Anaconda prompt: Command line for install libraries
    - e.g.
    - pip install keras
    - pip install --user tensorflow==1.15
    - pip install nltk

## Use Python on Lab computers

- Python 3.10.13, Jupyter Notebook are installed
- Most libraries are installed, including keras, tensorflow 2.15.0

- To install a new library
  - Command line:  pip install --user libraryName
  - e.g. pip install --user nltk

## Python basics

- Interpreter, can run line by line,
- Similar to Java in many ways,
  - Many Python online courses with videos, exercises
    - Google’s Python course at
      - https://developers.google.com/edu/python/
      - Two days, good if you have programming experience
      - Core part of the language only
- Learn the libraries
  - Use online tutorials
  - Check documentation, run example code
- Ask questions, fix errors
  - Keep calm and Google

## Python for Count, TF.IDF vectorisation

- Import the libraries
- Read the data, load data into a DataFrame
- Prepare the data
  - Bag of words model

  - CountVectorizer
  - TfidfVectorizer

  - Fit the raw data
  - Transform the raw data to vectors

- Code is attached on lecture schedule page

## Simple classifier for book reviews

Short version with simple classifiers is attached at lecture page.

- Load data
- Split data into train, test
- Prepare the data:
  - X:
    - CountVectorizer
    - TfidfVectorizer
  - Y: LabelEncoder
- Create a learning model:
  - KNeighborsClassifier or naïve_bayes or LogisticRegression
  - fit
  - predict
  - accuracy_score

## Bag-of-words model

What are the limitations or disadvantages?