# Overview

|  |
| --- |
|  |
|  |
| **This week we will look at “bag of words” model. Next session will be on NLP with deep learning techniques.** |
|  |
|  |
|  |
|  |
|  |
| **For every page, it computes the sentiment, ranks by median sentiment. Each circle is a paragraph on that page.** |
|  |
|  |
|  |
| **From scikit, we create a count vectorizer** |
| **Create a pipeline (classification)**  **Use a prelabelled dataset to train the classifier.** |
|  |
|  |
| **pdftools for extracting text, OCR on PDFs**  **magick – for transforming images and working with image data** |
|  |
|  |

# Code

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Number of pages in the PDF** |
| **Now we talk about images for the shiny app**    **Reads the first page and converts it into a high res image (2550).**  **Down converts to 600** |
|  |
| **Saves image as PNG for later use in the Shiny app** |
| **Now we will talk about Data extraction.**  **Single line of code. Every page is a character vector.** |
|  |
| **Converted to tibble** |
| **Now I want to break every page into paragraphs** |
| **Unnest into individual paragraphs** |
| **Add paragraph number** |
| **Only a few paragraphs on page 6, because there are tables in there. You may want to clean this up later if making it production worthy.** |
|  |
| **This is the labelled data for the training. It is an external corpus.** |
| **Unicode characters, etc.** |
| **Remove those unicode characters** |
| **Now we convert to Pandas** |
| **Access R variables using “r.paragraph\_1”** |
|  |
| **You can also create a sentence tokenizer. We don’t use it here, but this is available.** |
|  |
| **Output of lemmatization** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **We can swap out the logistic regression with a linear regression to find out the magnitude of the sentiment.** |
|  |
| **Using py$, we can access python variables in R.** |
|  |
|  |
|  |
|  |
|  |