

# *Recipe Recommendation System*

NEW WAY TO COOK FOOD





# Introduction

Home cooks everywhere are always look for inspiration for new dishes to make. Searching through google helps, but google search queries return better results for just food categories (e.g. italian, japanese, baking, sauces, etc.). This recipe recommendation system intends to refine the recipe search process by allowing users to search recipes based on specific ingredients that they have or intend to use. The primary advantage for this search method is the ability to find recipes that incorporate disparate or unlikely ingredient pairings, allowing adventurous home cooks to find the cooking inspiration they otherwise couldn't find.



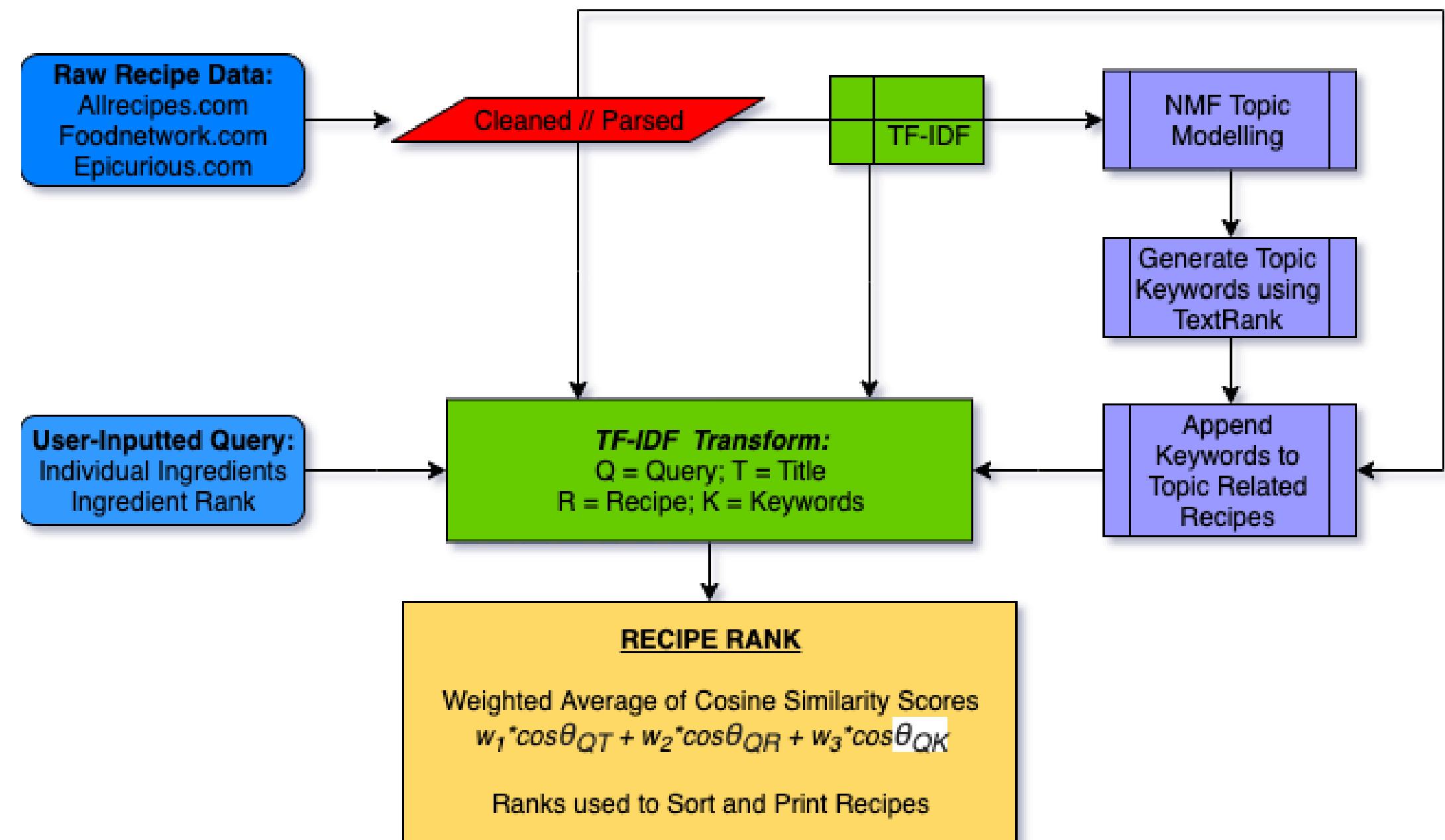
# Dataset

We constructed a dataset by web scraping from Allrecipes.com, Foodnetwork.com, and Epicurious.com using Python using Selenium

Data Features:

- Recipe Title
- Recipe Ingredients
- Recipe Instructions
- Recipe Score
- Recipe Keywords (Generated from Topic Modeling and Textrank)

# The *model* behind the system



# Model Tuning and Evaluation

Choosing a Topic  
Modeling Method:  
NMF vs. LDA

Choosing the Number  
of Documents per  
Topic from which to  
Pull Keywords

Choosing the number  
of documents to which  
Keywords are assigned

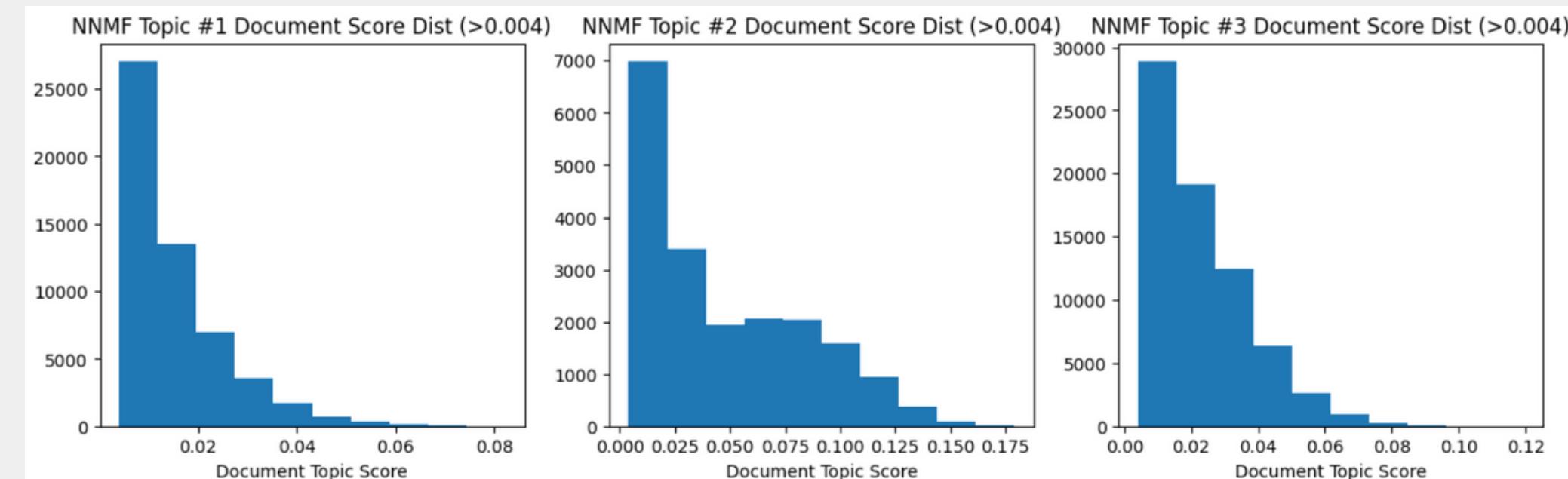
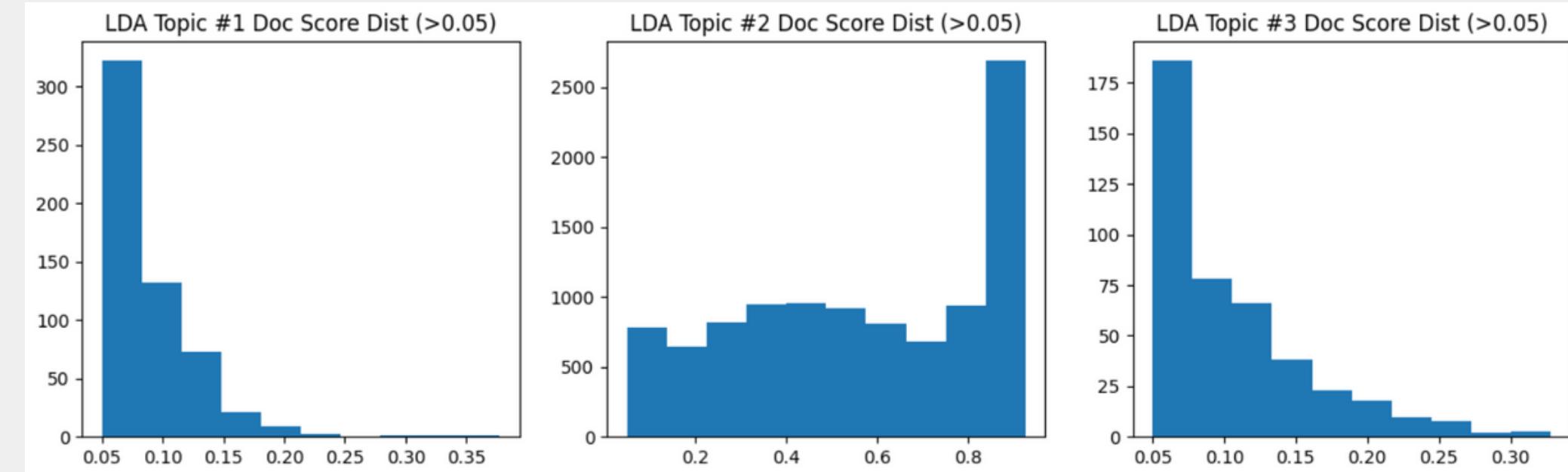
# Choosing A topic modelling method

The purpose is to visualize the distribution of document topic rankings and decide the cutoff for the documents that associate most with that topic.

IMPRESSIONS		By Documents	By Words
LDA	Topic 1	Salad	Salad
	Topic 2	Variety Beef/Pork	Pasta
	Topic 3	Cooked Fruit?	Stopwords?
NNMF	Topic 1	Spreads	Dressing
	Topic 2	Cake/Bread	Baking
	Topic 3	Chicken	Chicken

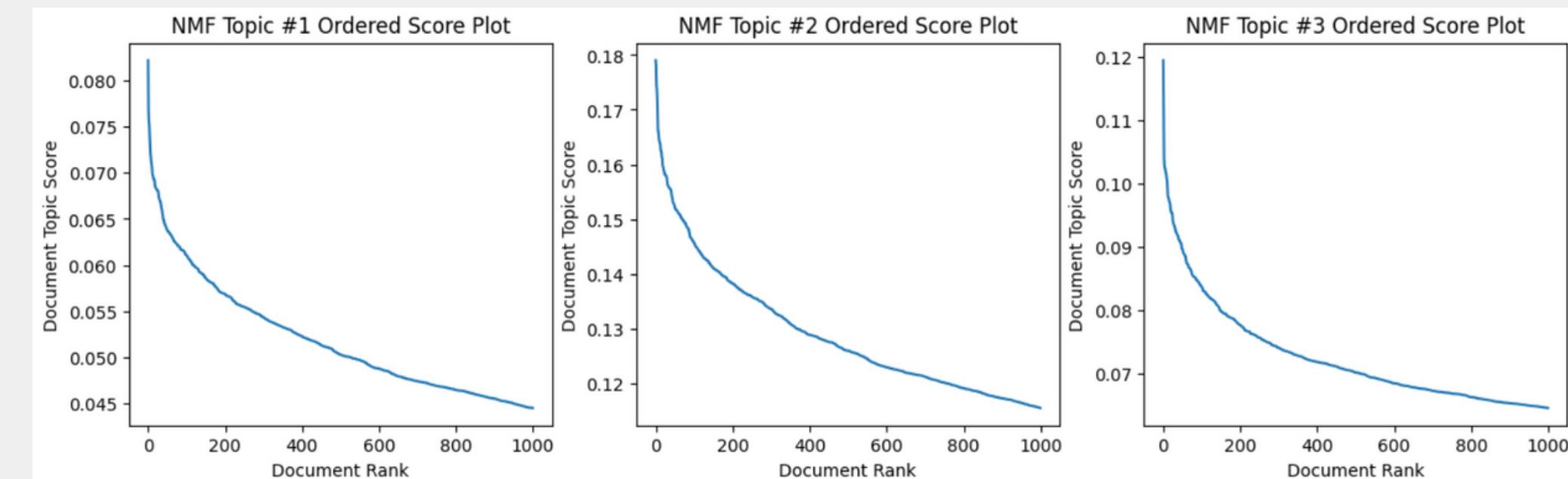
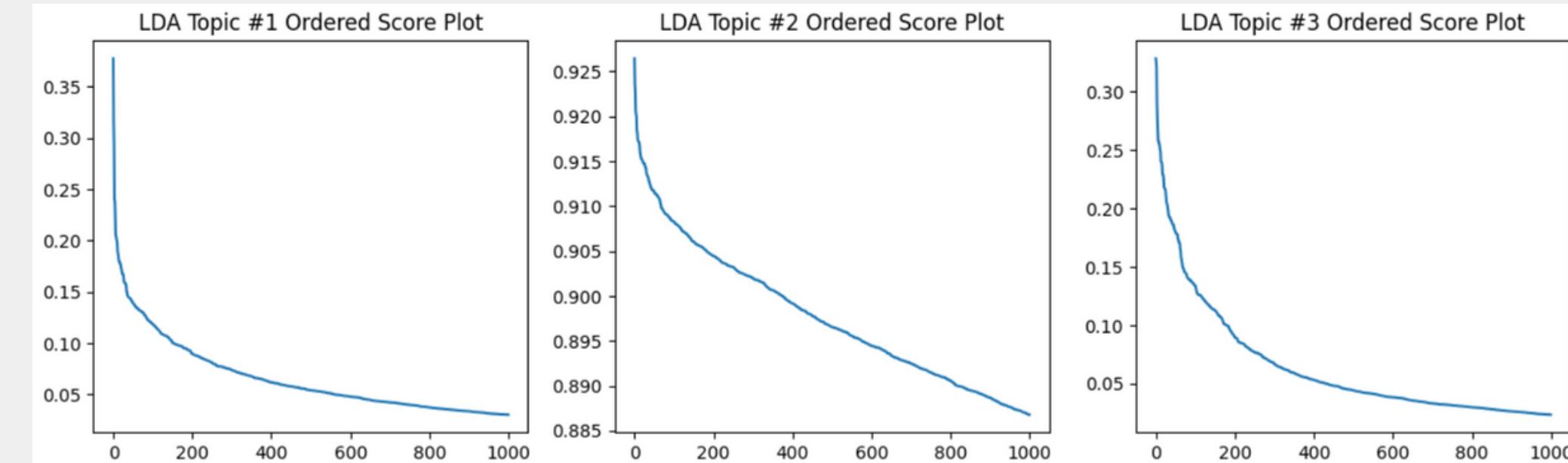
# Choosing the number of documents to which Keywords are assigned

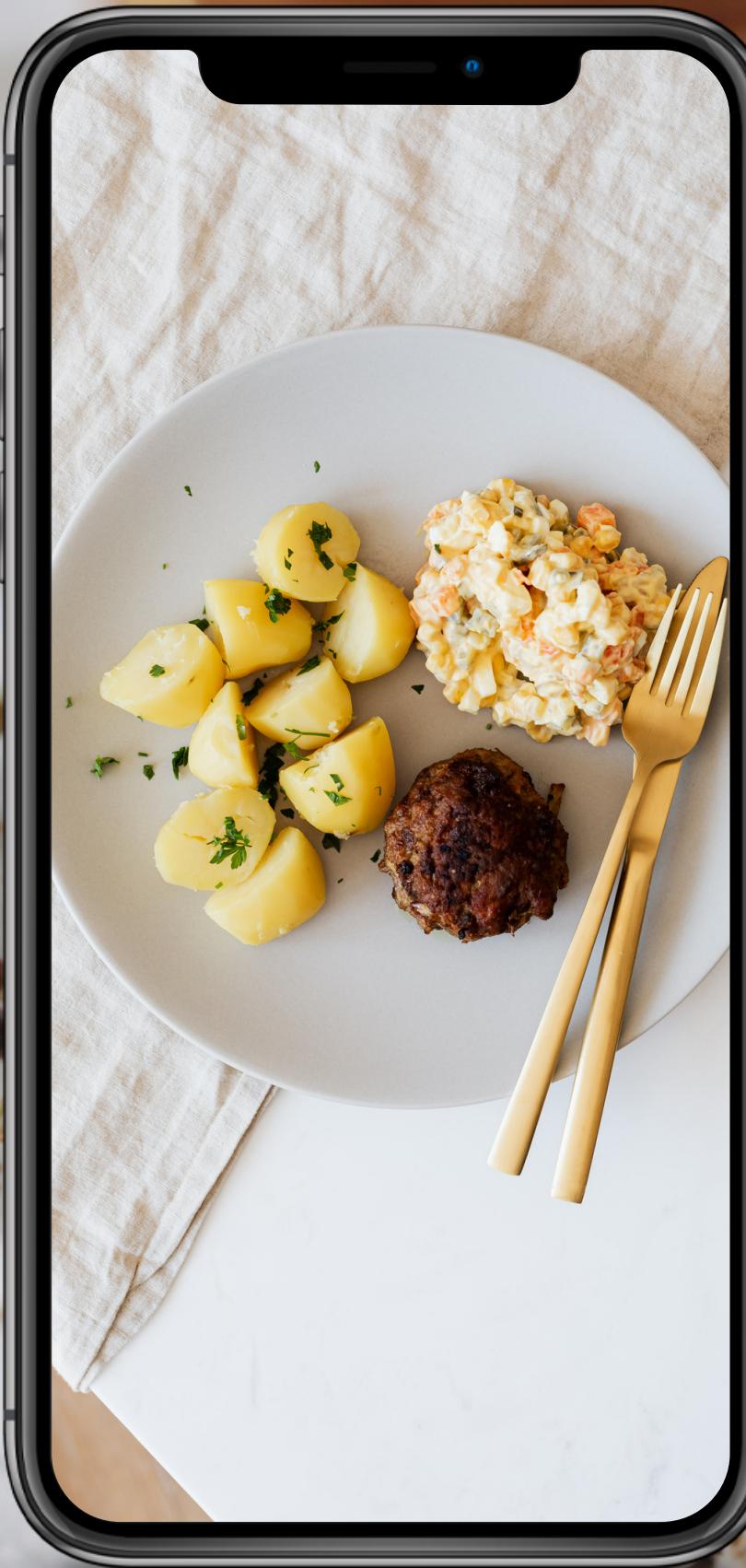
- Based on the first three topics for LDA and NMF, we will subjectly choose the top 1500-2000 documents. This number of documents seems to be where the loadings for each score distribution either level off or spike upward



# Choosing the Number of Documents per Topic from which to Pull Keywords

- The plots take after scree plots, and they plot the scores of documents as they relate to their respective topics in descending order. By plotting these elbow plots, it might shed light on the best number of documents to use to rank words.
- We will be using NNMF for the model because the topics seemed to have converged more distinctly on topic. According to the 'Scree Plots' corresponding to the NNMF topics, it seems that the scores begin leveling out around 200 documents. Therefore we will use TextRank on the top 200 documents for each topic.

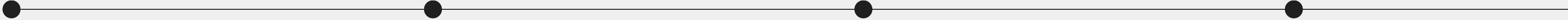




LETS TAKE A DEMO!

[HTTP://RECOR3.PYTHONANYWHERE.COM/](http://RECOR3.PYTHONANYWHERE.COM/)

# *Future Improvements*



## *1: Dense Word Embeddings*

Sophisticated Semantic Similarity Scoring  
Generate Abstractive Keyword Summarizations (Topic Names) for NMF produced topics

## *2 Database*

Updates using custom scrapers, Expand Recipe Dataset

## *3 Cloud Architecture*

Run Topic and Keyword generation on Spark in the cloud

## *4 Introduce ChatBot*

Introduc

# The People Behind the Business



*Sidhanth Jain*

MS Data Science



*Darshan Swami*

MS Data Science



*Parth Dhruv*

MS Computer Science



*Adithya Ramesh*

MS Electrical and Computer  
Engineering

# Thank you!

