# University of Greenwich Coursework Cover Sheet

| | | | |
|---|---|---|---|
| **Student Number** | **001091609** | **Tutor/Supervisor** | **Konstantin Kapinchev** |
| **Course** | **COMP-1682 – Final Year Project** | | |
| **Coursework Title** | **The use of topic modelling to classify research documents.** | | |
| **Due Date (Deadline)** | **24/04/2023** | **Submission Date (Today's Date)** | **24/04/2023** |

**The University subscribes to an electronic plagiarism detection service and reserves the right to submit the work submitted by any student to that service for analysis.**

By submitting this assignment, you are agreeing and confirming the following:

*"I confirm that the submitted coursework is my own work and that all material attributed to others (whether published or unpublished) has been clearly identified and fully acknowledged and referred to original sources. I agree that UGIC has the right to submit my work to the plagiarism detective service TurnitinUK® for originality checks."*

If you are in any doubt about the appropriate procedures for acknowledging and referencing the work of others, you should seek advice from your Personal Tutor/Head of English.

# The use of topic modelling to classify research documents.

## Table of Contents


Table of Figures. ..... 3

List of Tables ..... 3

1. Abstract ..... 4

2. Preface ..... 4

3. Glossary ..... 5

4. Problem Domain ..... 7

5. Acknowledgements ..... 7

Literature Review ..... 8

6. Introduction ..... 8

6.1 Topic Modelling ..... 8

6.2 Performing Dimensionality Reduction ..... 10

7. Machine Learning (Unsupervised & Supervised Learning) ..... 17

8. Natural Language Processing (NLP) ..... 21

9. Ethical issue in Natrural Language Processing ..... 24

10. Data Visualization and Exploration ..... 25

11. Methodology ..... 33

11.1 Front-End Implementation ..... 33

11.2 Environment setup ..... 35

11.3 Data selection process. ..... 35
11.4 Pre-processing of the data ..... 36
12. Model tuning and setting the hyperparameters. ..... 38
12.1 Unsupervised-learning models. ..... 38
12.2 Supervised Learning Models ..... 40
13. Results ..... 41
14. Restrictions and limitations ..... 44
15. Conclusion ..... 45
16. Reference list ..... 46

# Table of Figures.

(Figure 1; the concept behind topic models, and how documents are associated with topics and words.) ..... 10
(Figure 2: demonstrates the words and the respected probability of each word) ..... 16
(Figure 3: housing prices based on their size in $feet2$ the straight line representing the learning curve, and the curved line represents the quadratic formula) ..... 18
Figure 4, the number line graph of benign and malignant tumours) ..... 19
(Figure 5, Before and after clustering) ..... 20
(Figure 6, interactive dashboard using pyLDAvis) ..... 27
(Figure 7, topic 1 selected and it's terms) ..... 28
(Figure 8, list of recurring topics in our data set, with their ID and index) ..... 29
(Figure 9, certain terms can be found in other topics example) ..... 29
(Figure 10, word cloud of the top 200 most recurring words) ..... 30
(Figure 11, the most common words in Computer Science) ..... 31
Figure 12, Word cloud from Stahl, Timmermans and Mittelstadt (2016) used to demonstrate the difference between research interest before to research interest nowadays) ..... 32
(Figure 13, web application front-end interface and results) ..... 34
(Figure 14, the labelling of the input's topic and recommended articles based on the topic label.) ..... 35
(Figure 15, the process of data collection, processing, and exporting to the machine learning model.) ... 37

# List of Tables

(Table 1: Representation of topics and their probability of belonging to a topic) ..... 16

(Table 2: BoW process example) ..... 23
Table 3: The results of testing the models. ..... 42
Table 4: The results of the MultinomialNB model to the topics in the dataset. ..... 42
Table 5: The results of the Random Forest Classifier (RFC) model to the topics in the dataset. ..... 43
Table 6: The results of the C-Support Vector Classification (SVC) model to the topics in the dataset. ... 44

# 1. Abstract

In recent years, topic modelling has become an important area of research in the field of data science and its uses can be found in many other different industrial sectors. In this report we will explore the different ways to implement a topic model and where to use it. Furthermore, we will discuss the optimal setup for each model that has been used in this project and what were the restrictions and limitations that we faced while building and training the models. In this project, we explored the use of Latent Dirichlet Allocation (**LDA**), Latent Semantic Analysis (**LSA**), Non-negative Matrix Factorization (**NMF**), and Term Frequency-Inverse Document Frequency (**TF-IDF**) models to label a dataset of over 20,000 abstracts to their respective topics. The main objective of this project was to identify the most effective approach to topic modelling in terms of accuracy and efficiency. We also aimed to correctly label each abstract in the dataset to the corresponding fields and take the input of a user and determine the recurring topic in that abstract.

# 2. Preface

As we move towards a more digitalized environment, we will be required to move important documents to a more secure location that will last the test of time. However, that comes with a challenge, moving data that big manually will take a significant and incomprehensible amount of time, thus, the use of Machine Learning and AI techniques is required. One such method is known as Topic Modelling.

Topic modelling is a Natural Language Processing tool for identifying hidden patterns and themes within data that contains textual content and can be applied to a wide range of domains including New Articles, documentations, and academic research.

The findings of our project investigating the use of various topic modelling techniques for labelling a dataset of more than 20,000 abstracts are presented in this report. We hope that this report will serve as a valuable resource for researchers and practitioners interested in the use of topic modelling for data analysis.

# 3. Glossary

In this project there are different terminology that is being used. In this section, you'll find the terminology and their meaning.

| Term | Meaning |
|---|---|
| LDA. | Latent Dirichlet Allocation. |
| NMF. | Non-negative Matrix Factorization. |
| LSA. | Latent Semantic Analysis. |
| TF-IDF. | Term Frequency-Inverse Document Frequency. |
| Topic Modelling. | division of machine learning that handles the grouping of documents based on their contextual themes. |
| Model. | A Machine learning algorithm. |
| Supervised and Unsupervised. | Supervised is a model that has the right answer and labels, and unsupervised is a model that isn't given labels or the right answer (or path). |
| Clustering | The process of grouping together similar points of data |
| Natural Language Processing (NLP) | A machine learning division tasked with the processing of the human language. |
| Features | Attributes of a data point. |
| Dimensionality Reduction | is to transform the original high-dimensional dataset into a lower-dimensional representation that still contains most of the important information |
| PCA | is a linear dimensionality reduction technique that works by finding the principal components of the data |
| Visualization | to explore the data, represent it in a meaningful way, and interpret the results in a better way. |
| Count Vectorization | refers to the process of breaking down a sentence or any text into words such as converting all words to lowercase and thereby removing special characters. |

| | |
|---|---|
| Word Cloud | A representation of the most common words in the form of a cluster that resembles a cloud |
| Data Cleaning | Means to remove the stop words, and then tokenize and lemmatize the text data. |
| Stop words | Such as “a” and “the”. |
| Tokenization | splitting the input data into a sequence of meaningful parts |
| Lemmatization | to break a word down to its root meaning to identify similarities. |

## 4. Problem Domain

Research studies conducted on existing topic models have found restrictions and limitations that can affect the accuracy and precision of the results. There has been an overall improvement to the algorithms used for topic modelling, however, it is still difficult to interpret the end results. This can be due to having features that are not well selected, or not enough features available and having too many topics or far too few can also affect the result and accuracy of the model.

In this project, we are to develop and test different topic models with the purpose of finding the recurring topics in our dataset of abstracts, and then test the accuracy of the models based on a test data set. Both the data sets must be cleaned, have their stop words removed (more on this in the methodology section), and pre-process the data before having them be digested by the models.

## 5. Acknowledgements

We would like to express our gratitude to our supervisor and mentor for their guidance and support throughout this project.

# Literature Review

## 6. Introduction

Massive growth in data can be attributed to a rapid shift towards the Digital Age which we've witnessed over recent decades. The excessiveness of data generated because of this change is making its management and analysis increasingly difficult. To tackle this problem, researchers have developed topic modelling algorithms as one viable method and technique for categorizing and facilitating analysis of such voluminous amounts of data (Churchill and Singh, 2022). More sophisticated techniques for information analysis and management are required because of the developments in digital transformation and the accessibility of vast amounts of data. Topic modelling can be of help to a plethora of industrial sectors such as finance, social media, marketing, healthcare, and many more that require the need to handle natural language processing of documentations.

In a large corpus of unstructured textual data, topic modelling is a method for locating latent themes and patterns. It involves the use of statistical models to identify the underlying topics that are present in the text and to label each document in accordance with its main topic.

In this project, we investigated the application of several well-known topic modelling techniques, such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Non-negative Matrix Factorization (NMF), and Term Frequency-Inverse Document Frequency (TF-IDF). Our objective was to determine the recurring topics in the dataset that we have obtained online. The dataset contains more than 20000 samples of research article abstracts. We then proceed to label the data points using the topic modelling techniques.

This is the code which will be used to aid students identify the recurring topics in research articles using the LDA (Latent Dirichlet Allocation) model. The model will read the abstracts of each article and classify it into the different categories that are keep recurring in the text.

### 6.1 Topic Modelling

Large amounts of unstructured textual data can be effectively analysed using topic modelling. The underlying topics that are present in the text are found using statistical models, and each document is then categorized according to its main topic. A wide range of uses for this strategy are possible, including text classification and scholarly research (Churchill and Singh, 2022).

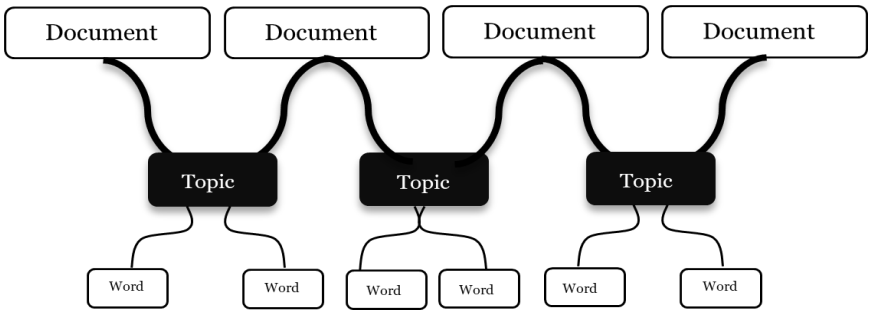
Topic modelling is a division of machine learning that handles the grouping of documents based on their contextual themes by identifying similar keywords that can be found in the clusters of documents. The goal of dimensionality reduction is to transform the original high-dimensional dataset into a lower-dimensional representation that still contains most of the important information, and then we would fit the transformed data set to the topic models. There are two main general expectations that we can make about topic models, firstly being that each document that the model handles will have different topics within, and this means that each topic will contain a collection of words (Crossno et al., 2011).

One of the challenges of topic modelling is that it can be computationally expensive and time-consuming, particularly when working with large data sets, to address this we would need to preform dimensionality reduction to reduce the total number of variables. A Dimensionality Reduction is performed to the set of words in the documents, and then split into smaller sets of topics that would be interpreted better and have more meaning. In addition, words can have associations with different topics (Hecking and Leydesdorff, 2018). There are several methods to implement dimensionality reduction such as Principal Components Analysis (PCA), Latent Semantic Analysis (LSA), Non-negative Matrix Factorization (NMF), and Latent Dirichlet Allocation (LDA) which are all models used in this project.

Research article's key themes and topics can be found using topic modelling, which enables researchers to spot trends and areas of study in their area of expertise (Xiong et al., 2019). The research field can benefit from topic modelling in this situation. In our project, topic modelling is applied to perform text classification. This entails applying topic modelling as a reprocessing step for text classification tasks, enabling the data to be labelled in accordance with the most pertinent topics before being fed into a machine learning model. There are a multitude of topic modelling models that we can use (Vayansky and Kumar, 2020), each model has its own use case. For example, LDA is flexible in its adaptation, however it isn’t always suitable for more complex data relationships. In our project we will be using **LSA**, **NMF, and LDA models**.

The term "topic models" refers to statistical techniques for identifying the latent semantic structures of a large text body (Kee et al., 2019). In the world we live in today, the volume of textual information and data

we encounter daily is essentially beyond our capacity to process it. We may be able to manage and comprehend the enormous collections of unstructured textual data and information with the help of topic models. Topic models were first developed as a text-mining tool, but they have since found use in many other disciplines (Neogi et al.,2019). This paper conducts a thorough comparison of LSA and the widely used TF-IDF approach for text classification and demonstrates that LSA produces higher classification accuracy. As stated previously, each topic modelling model has advantages in different aspects of text processing, however, all the models share a similar weakness. Topic modelling is widely known for optimization issues and instability (Vayansky and Kumar, 2020).




*(Figure 1; the concept behind topic models, and how documents are associated with topics and words.)*

For example, in this project to fine tune the hyperparameters it would take hours for the model to reprocess all the information, and it was also prone to throwing errors which would cause us to initiate the model pre-process process once again. In addition, a few models are not suitable for real-world data representation (Zhang et al., 2022). This can be due to the model assuming crucial parameters calculations to try and overcome uncertainty in the data. Thus, causing us to perform more time-consuming iterations to find the best tuning for the parameters. Optimization can be done by selecting optimal and meaningful features early on, although there has been an improvement to the topic modelling algorithms, however, optimization is still important and highly required to provide results that are reliable.

## 6.2 Performing Dimensionality Reduction

### 6.2.1 Non-negative Matrix Factorization (NMF)

NMF, also known as **Non-negative Matrix Factorization,** is a machine learning method where we restrict the matrix to non-negative values and is another method that is used to extract features and for dimensionality reduction. The goal of NMF is to find a set of basis vectors that can be combined to produce the original data matrix (Arora, Ge and Ankur Moitra, 2012). By reducing the dimensionality of the basis vectors, it is possible to reduce the dimensionality of the data.

For example, if we factorize matrix $A$ into two separate matrices $B$ and $C$

$$A \approx B \times C$$

However, this is just an approximation and not a guarantee that the matrices are equivalent to the original matrix.If we are to assume that:

$$(b_1, b_2, b_3, \dots, b_n) \; and \; (c_1, c_2, c_3, \dots, c_m)$$

$$A = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \end{bmatrix}, B = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_i \end{bmatrix}, C = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_i \end{bmatrix}$$

To have this equation make more sense is by considering the following formula:

$$x_i = \begin{bmatrix} n_{1i} & n_{2i} & \cdots & n_{3i} \end{bmatrix} \times \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_i \end{bmatrix} = \sum_{j=1}^{k} nij \times mi$$

Where $n_i$ are the weights of each component and the $m_i$ are the components. For the components to be meaningfully interpreted, we introduce several conditions. We restrict the weights and underlying components in the case of NMF to be non-negative. In essence, NMF would overlay a certain number of components onto each data point (Wang and Zhang, 2022).

The main restriction in NMF is that the B and C matrices must be non-negative, which will ensure that the factorized results are additive and interpretable (Zhu and Cunningham, 2022). This restriction is ideal for feature extraction tasks, where the basis vectors in B can have meaningful features that can be interpreted in the data, and the coefficients in C can indicate the strength and relation of each feature to each observation.

NMF is also suited to handling sparse data and is ideal for datasets with a great number of zero values. In addition, NMF can be used for clustering and can identify clumps of data that share similar features or topics.

### 6.2.2 NMF Use Case:

NMF is appropriate for tasks where the underlying factors can be interpreted as non-negative. Consider breaking down a term-document matrix where each column represented a document and each element in the document represented the weight of a particular word (the weight could be the raw count, the TF-IDF weighted count, or some other encoding scheme; these specifics are not relevant at this point) (Sherstinova et al., 2020).

If the documents were taken from news articles, what would happen when we split this up into two matrices? In articles about networks, the word "web" is likely to appear along with other words like "cyber" and "connection.". To create a "connection" component vector, these words would likely be grouped together, and each article would have a certain weight for the "connection" topic. Considering this, an NMF decomposition of the term-document matrix would produce elements that could be regarded as "topics" and decompose each document into a weighted sum of topics. Topic modelling is a crucial NMF application.

Note that using other decomposition techniques would prevent this interpretation from being possible. We are unable to define what it means to weigh food in a "negative" way. This is another instance where the underlying elements (topics) and their weights ought to be non-negative. The fact that NMF generates sparse representations naturally is another intriguing feature of this algorithm. This makes sense in the context of topic modelling because there are typically not many topics in documents.

We use the Frobenius Norm formula to minimize the weight costs.

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$

### 6.2.3 Latent Semantic Analysis (LSA)

Another fundamental method in topic modelling is Latent Semantic Analysis, or LSA. The fundamental concept is to take a matrix of our existing documents and terms and separate it into a document-topic matrix and a topic-term matrix (Mohammed and Al-Augby, 2020). One of the fundamental methods in topic modelling is latent semantic analysis, or LSA. The main concept is to take a matrix of our existing documents and terms and break it down into two separate topic-term and document-topic matrices.

The generation of our document-term matrix comes first. Given m documents and n words in our vocabulary, we can build an $m \times n$ matrix $A$ where each row denotes a document, and each column denotes a word. Each entry can simply be a raw count of how many times the $j^{th}$ word appeared in the $i^{th}$ document in the most basic form of LSA. However, since they do not consider the importance of each word in the document, raw counts are not particularly useful in practice. For instance, the word "nuclear" probably tells us more about the topic(s) of a given document than the word “test” (Kalepalli et al., 2020).

As a result, TF-IDF scores are frequently used in place of raw counts in document-term matrices in LSA models. TF-IDF, or term frequency-inverse document frequency, assigns the following weight to term $j$ in document $i$:

$$w_{ij} = tf_{ij} \times \log\frac{N}{df_i}$$

with $w_{ij}$ being the TF-IDF score, $tf_{ij}$ being the occurrences of the terms in the document, and $\log\frac{N}{df_i}$ being the total number of documents over the documents containing the term/word.

Truncated SVD can be used to perform dimensionality reduction. Singular value decomposition, or SVD, is a technique in linear algebra that factors any matrix M into the product of three different matrices: $M = U \times S \times V$, where $S$ is a diagonal matrix of the singular values of $M$ (Arora, Ge and Ankur Moitra, 2012).

LSA is useful when performing information retrieval as it can help improve search result accuracy. This is done by reducing the dimensionality of our text data. LSA can also be used in document classification and recommendation systems, these being an important factor in this project. We utilize the documentation classification feature to classify documentation based on their contents, as LSA can assist in identifying both similarities and differences between documents.

Importantly, truncated SVD reduces dimensionality by choosing only the $t$ largest singular values and keeping the first $t\ columns$ of $U$ and $V$. In this case, $t$ is a hyperparameter that we can choose and modify to reflect the quantity of topics we're looking for. AS such we can think of $t$ as the most important dimensions in the transformed space.

The formula can be written as such:

$$\boxed{A \approx U_t S_t {V_t}^t}$$

### 6.2.4 Principal Components Analysis (PCA)

PCA is a linear dimensionality reduction technique that works by finding the principal components of the data. These principal components are orthogonal directions in the feature space that capture the largest amount of variance in the data. By projecting the original data onto these principal components, it is possible to reduce the dimensionality of the data while still preserving much of the structure (Hecking and Leydesdorff, 2019). PCA is optimal to use in Image processing and natural language processing. Our main concern is how it can be used in NLP (natural language processing), it helps by finding the main components of the word vectors. Word vectors are the mathematical representation of words in a text corpus (Which consists of structured sets of texts that are large) which holds the meaning of the words and the relationships between them.

By analysing large amounts of text data and identifying patterns in how words are used in context, machine learning algorithms can learn to generate word vectors that capture these semantic relationships between words.

The mathematical formula consists of the following variables:

$$u = \frac{1}{\sqrt{\lambda}} \times Z^T \times N^{\frac{1}{2}} \times v$$

$$v = \frac{1}{\sqrt{\lambda}} \times N^{\frac{1}{2}} \times Zu$$

PCA consists of studying a data matrix Z, provided with a metric matrix $\mathbf{I}_p$ defined in $\mathbb{R}^p$ and another metric N defined in $\mathbb{R}^n \approx \boldsymbol{N} = (\frac{\mathbf{1}}{\boldsymbol{n}})\mathbf{I}_n$ . The Z matrix can be normalized or non-normalized using PCA. The normalized Z matrix is equivalent to $\boldsymbol{Z} = \boldsymbol{XS^{-1}}$ where S is the diagonal matrix of the standard deviations and non-normalized is denoted as $\boldsymbol{Z} = \boldsymbol{X}$ without the addition of the diagonal matrix.

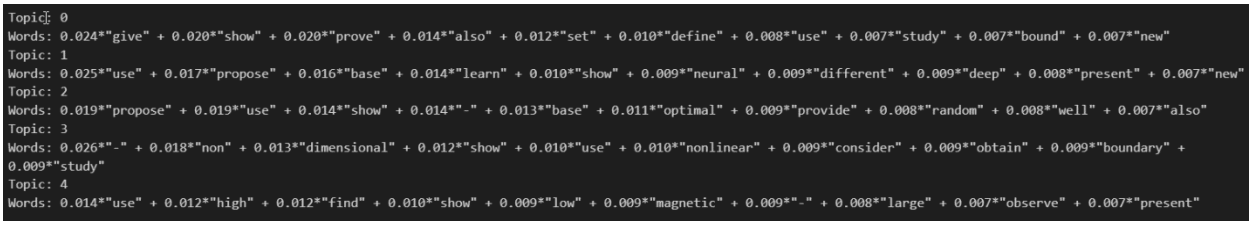
### 6.2.5 Latent Dirichlet Allocation (LDA)

Natural language processing (NLP) employs the probabilistic topic modelling method known as Latent Dirichlet Allocation (LDA). By identifying the topics that most accurately represent each theme, LDA aims to reveal the hidden thematic structure of a group of documents (Bergamaschi and Po, 2014). LDA can be used to find the topics that are most pertinent to computer science, mathematics, and physics in the case of abstracts for STEM subjects. There are set assumptions that we can make about LDA models, the first of which is that the documents are going to be a collection of words also known as Bag of Words, meaning the order and grammatical structure of the words are not considered (Kalepalli et al., 2020).

We then perform a pre-processing of the text data to get rid of stop words, punctuation, and other extraneous details to represent the frequency of each term in each document, we tokenize the text to separate it into individual words or phrases.

The topics that best explain the variation in the data can be found using LDA once we have the document-term matrix until the model converges on a stable solution, this entails repeatedly assigning each word in each document to a topic and adjusting the topic probabilities (Greene, O'Callaghan and Cunningham, 2014). The LDA model produces a list of topics, and each of which is represented by a distribution over the vocabulary words. The topics can then be understood by looking at the most frequently occurring words in each topic and using domain knowledge to assign them to pertinent STEM subject areas. LDA is suitable

to use for unseen documentation, as it has generative components that can assign topics to those unseen documents.




*(Figure 2: demonstrates the words and the respected probability of each word)*

As we can see here the from the output of our model, each topic contains a set of words, and each word then contains the probability of it belonging to that topic.

as shown in the table below. Each row in the table represents a different topic and each column a different word in the corpus. Each cell contains the probability that the word(column) belongs to the topic(row).

(Table 1: Representation of topics and their probability of belonging to a topic)

| Topics | Word 1 | Word 2 | Word 3 | Word 4 | Word 5 |
|---|---|---|---|---|---|
| **0** | 0.024 | 0.020 | 0.014 | 0.012 | 0.010 |
| **1** | 0.025 | 0.017 | 0.014 | 0.010 | 0.009 |
| **2** | 0.019 | 0.016 | 0.014 | 0.013 | 0.011 |
| **3** | 0.026 | 0.013 | 0.010 | 0.011 | 0.010 |
| **4** | 0.014 | 0.012 | 0.012 | 0.009 | 0.009 |

### 6.2.6 TF-IDF

TF-IDF also known as Term Frequency- Inverse Document Frequency is a topic modelling technique used to evaluate the importance of words in relation to the corpus of texts. This is done by comparing the relevancy of a word to a document or a set of documents, the process involves comparing the number of times a word is repeated in a document to the frequency of it appearing in the corpus (Zhao et al., 2018).

The higher the frequency of appearance for the word in a document and the lower appearance frequency in the corpus then the word would have higher importance to the document.

The frequency is calculated the number of times a word appears in document divided by the sum of words in the document. TF-IDF contains both the frequency and inverse frequency to calculate the score for a word. The inverse frequency is the amount of information a word holds across the total sum of documents in a corpus divided by the number of documents where the word appears.

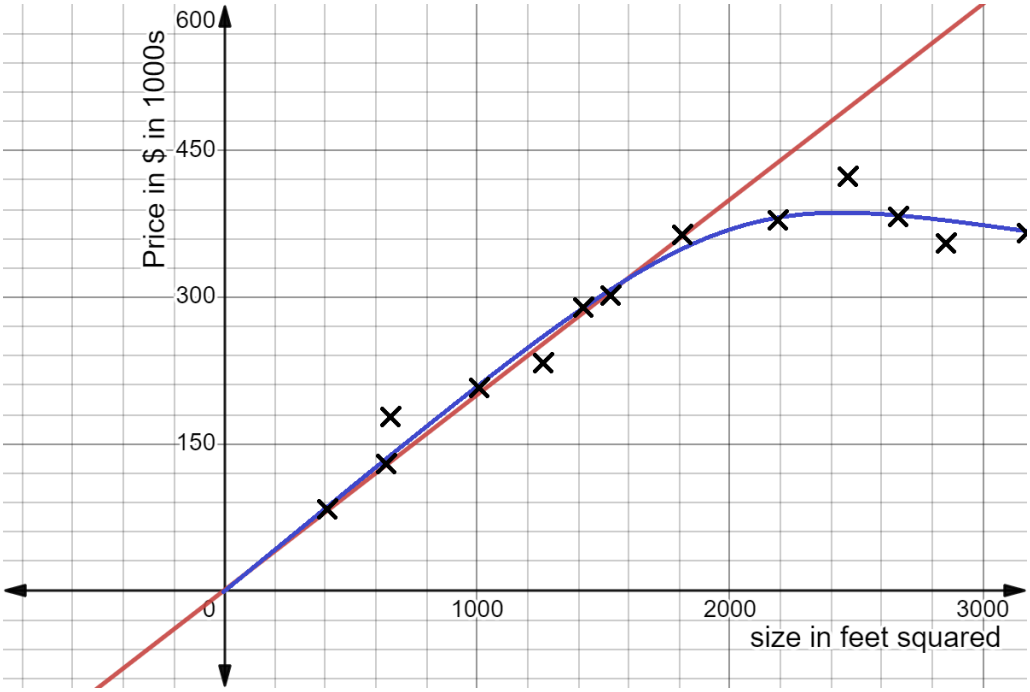
$$\text{tf} - \text{idf}(t, d) = tf(t, d) \times idf(t)$$

$$\therefore \text{Where } idf(t) = \log\left[\frac{n}{df(t)}\right] + 1$$

Where $n$ is the total number of documents in the document set and $df(t)$ is the document frequency of $t$. When used in topic modelling, TF-IDF can be of help in identifying the words of most importance that will aid in defining a topic or cluster of related documentation.

# 7. Machine Learning (Unsupervised & Supervised Learning)

There are multiple types of machine learning algorithms such as supervised learning, unsupervised learning, reinforcement learning, and recommender systems. The main two algorithms that are used are supervised and unsupervised learning. They are used to analyse and model data. In supervised learning, the algorithm is trained on a labelled dataset, meaning that the right answer is given, where the input data is associated with the output or target variable (Berry, Mohamed and Bee Wah Yap, 2019). An example of this would be the housing market prices.




(Figure 3: housing prices based on their size in $feet^2$ the straight line representing the learning curve, and the curved line represents the quadratic formula)

The goal is to learn a mapping function that can predict the target variable for new, unseen inputs accurately. To achieve this goal, we can use regression methods or classification methods (Sathya and Abraham, 2013). Regression means to predict a continuous valued output (for example the price value in the graph), while classification deals with more discrete valued outputs, generally being a 1 or 0, yes or no, and a false or true output. However, there are times where a classification problem can have multiple answers, for example in the case where a cancer tumour can either be benign or malignant, and if it were malignant, then which type would it be considered.

$$\{0_{benign}, 1_{malignant\ types(1\ ,2,or\ 3)}\}$$

We can chart this data using a number line graph as such:

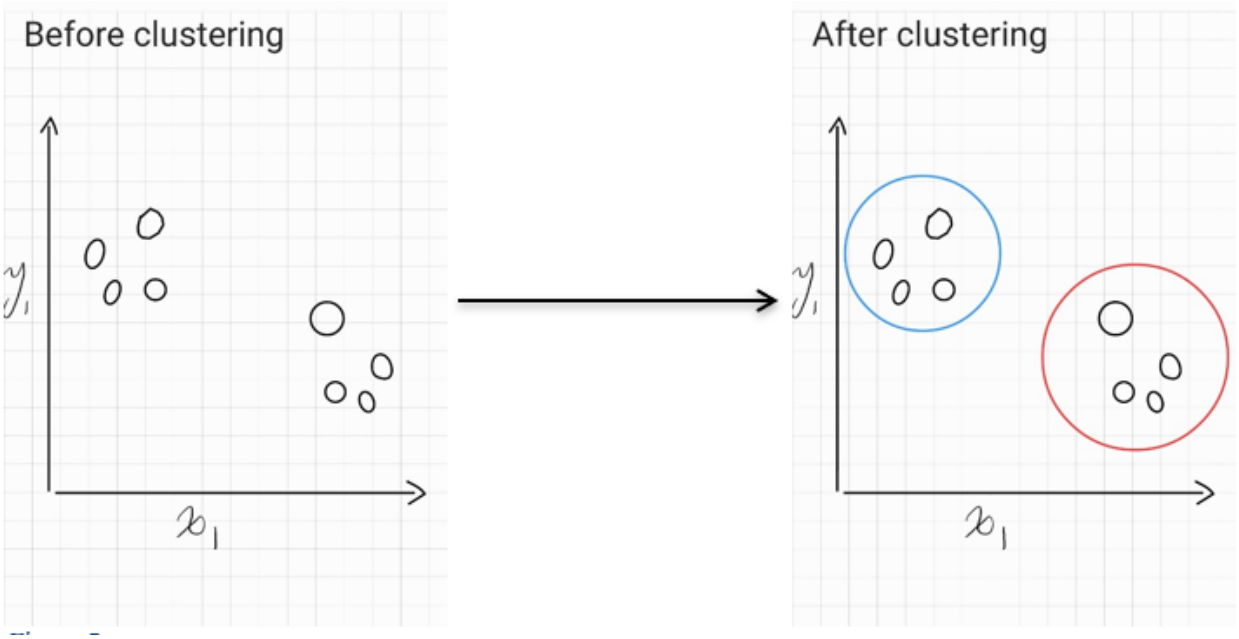
Figure 4, the number line graph of benign and malignant tumours)

Instead of using crosses for both options, we denote benign with a circle symbol, and malignant with a cross. In this case we only used 1 feature to denote the problem. However, in other machine learning problems, to have better results, we would need to have more features for the model to better diagnose the data. For example, to further improve this cancer diagnosis model we can add features for the thickness of the tumour clumps, the uniformity of the cell size and shape, etc.

In essence, the more features that we can add then the more the algorithm will learn, thus, providing us with better predictions.

In contrast, unsupervised learning algorithms do not have labelled data, meaning the model is not given any answers. Furthermore, we aren't given the data's information or purpose, and the goal is to find structure or patterns in the input data without knowing the output.

The algorithm would then decide what to do with the subsets of data, it might decide that each set or subset is a separate cluster or that a group of subsets form a cluster of their own, this process is called clustering and it is part of the cluster algorithm. Clustering and unsupervised learning is widely used in organizing large computer clusters and data such as social network analytical data, market segmentations, and astronomical data analysis (Schwenker and Trentin, 2014). This algorithm is also used in our project to classify and group each topic set to one another, and in the methodology a demonstration will be demonstrated.




*(Figure 5, Before and after clustering)*

In figure 5, we can see that groups of data with similarities tend to be nearer of each other, the algorithm would then classify those groups of data and label them. The algorithms runtimes would increase based on the squared value of $n$; thus, clustering algorithms have a complexity of $O(n^2)$. This is not ideal, as generally we are working with datasets that are very substantial in size from tens of thousands of data points to millions or even greater than that. This entails selecting a clustering algorithm that is less complex such as K-means algorithm. K-means algorithm has a complexity of $O(n)$, this means that the algorithm scales linearly with the data points.

In K-means algorithm we determine the best k centre points also known as centroids using an iterative process. The process would then assign the data points that are closest to similar centroids to the same cluster/group. The k-means algorithm picks centroid locations to minimize the cumulative *square* of the distances from each example to its closest centroid (Likas, Vlassis and J. Verbeek, 2003). For example, topics in our data set that have similarities, such as networking, programming languages, etc. would be grouped together under the label of Computer Science.

Topic modelling is a form of unsupervised learning that is used to identify the underlying topics or themes in a collection of documents. It is often applied to large text datasets, where the documents are unstructured

and difficult to analyse manually. In topic modelling, the algorithm is trained on the input data, and the goal is to discover the hidden structure of the data, i.e., the topics, and their associated words.

One common technique for clustering in topic modelling is called the Latent Dirichlet Allocation (LDA) algorithm. LDA is a probabilistic model that assumes each document is a mixture of topics, and each topic is a probability distribution over words. LDA starts by randomly assigning words in each document to one of the topics, and then iteratively updates the topic assignments based on the distribution of words in the corpus (Li, Kuo and Lin, 2011).

The algorithm works by iteratively optimizing two main components: the document-topic distribution and the topic-word distribution. The document-topic distribution describes the likelihood of each document containing each topic, while the topic-word distribution describes the probability of each word appearing in each topic. These distributions are learned by computing the probabilities of each word given the current topic assignments and then updating the assignments based on the probability distributions.

Once the algorithm has converged, each document is associated with a distribution over topics, and each topic is associated with a distribution over words. This allows us to perform clustering on the documents based on their topic distributions. Documents that have similar topic distributions are likely to be clustered together, indicating that they share common themes or topics. In conclusion, clustering is an essential part of topic modelling and allows us to identify the underlying structure of the data. By using unsupervised learning algorithms such as LDA, we can discover the hidden topics in a collection of documents and group them together based on their similarities. This can be useful in many applications, such as information retrieval, content recommendation, and data visualization.

# 8. Natural Language Processing (NLP)

NLP (Natural Language Processing) is a set of algorithms and models, such as deep learning algorithms, that allow machines to understand and interpret human language and emulate it with high accuracy. This is achieved through training computer models for the purpose of analysing and manipulating language data, such as text and speech, and can also be used to read an individual's handwriting (Lopez and Kalita, 2017).

The purpose of natural language processing (NLP) is to create algorithms and models that can handle different language-related tasks such as text categorization, sentiment analysis, machine translation, and text synthesis (Acero and Stern, 1990).

An example of this is the use of NLP in speech recognition, which can be seen in AI assistants. Language consists of different characteristics, and in NLP, we would categorize those different characteristics into different groups.

such as Morphology for the shape and structure of a word, Syntax for the rules of how words are used, Semantics for the meaning behind words, and other important characteristics that will allow models to learn how to contextualize words and phrases with meaning (Uday Kamath, Zhanjiang Liu, and Whitaker, 2019).

NLP is essential in the development of intelligent systems such as chatbots, virtual assistants, machine translation systems, and topic modelling models. The demand for NLP technologies has grown exponentially in response to the explosion of digital data, resulting in exciting new developments in the field.

Natural language processing methods are used to pre-process the text data in our documentation which will allow the analysis process to be easier to perform. This may include removing stopping words such as “the” and “a” as these don’t provide meaning to words. Stemming is another NLP process in which we reduce the words to the original form, for example, “played” will return to its origin root word “play”. We perform stemming to reduce the number of unique words that the algorithm will need to process, thus improving the performance of the model and increasing its efficiency (Jivani, 2011).

NLP is also used to represent the textual data numerically using TF-IDF and another technique known as Bag-of-Words (BoW). BoW is used to represent the text data in the form of a matrix of word count, each row corresponds to a document and each column corresponds to a word in the vocabulary. For example, the sentences “The bug in the code is here” and “The programmer is good” will first be turned into a vocabulary as such [‘The’, ‘bug’, ‘in’, ‘code’, ‘is’, ‘here’, ‘programmer’, ‘good’], as we can see the vocabulary consists of 8 unique words (Zhao et al., 2018). We can then make a vector of these sentences which will show the words and the number of occurrences in the sentences as such:

*(Table 2: BoW process example)*

| | 1<br>The | 2<br>Bug | 3<br>in | 4<br>code | 5<br>is | 6<br>here | 7<br>programmer | 8<br>good | Length of sentence (in words) |
|---|---|---|---|---|---|---|---|---|---|
| Sentence 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 6 |
| Sentence 2 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 4 |

By representing the data as a matrix of word counts, it will allow the use of common machine learning algorithms such as support vector machines (SVM) and random forest regression to predict the results based on our data. And in topic modelling it used to identify the recurring words that hint at a particular topic. However, BoW has multiple limitations such as not taking the order of words into consideration i.e. “There he fell” and “He fell there” would be represented in the same way. Furthermore, BoW might not capture the meaning of the text data, as it mainly focuses on the frequency of individual words rather than the relationship between them and other words in the documents.

NLP is additionally used to calculate the model’s perplexity and coherence. Both being intrinsic evaluation matrices. In Intrinsic Evaluation, the result is to be evaluated against the reference text provided in the by the user. In addition, intrinsic evaluation can be divided into two categories those being adequacy and fluency. Fluency deals with the grammatical structure and correctness of the results, while adequacy deals with the amount of meaning that is being expressed by the output.

Performing intrinsic evaluation is important as it allows us to critically assess the quality of the model’s performance based on criteria that is internal, rather than relying on external measures to gauge how well our model performs a task. In addition, intrinsic evaluation is important as it allows us to compare all the topic modelling algorithms that we have developed in our project, thus allowing us to determine which model performs best for certain datasets, and also provide us with ways to improve our parameters to enhance the performance of our models.
Intrinsic evaluation is then applied to a subset of the data as it is faster and simpler to evaluate and will allow us to understand the system used to create the word vectors.

Perplexity focuses on the likely hood of the model being able to predict the new unseen data. And coherence represents the semantic similarity between the top words in a topic and the number of re-occurrences that those top words have with each other.

# 9. Ethical issue in Natrural Language Processing

A negative aspect that was found whilst researching the topic modelling and NLP is that there is a likelihood of delving into gender stereotypes by reinforcing and exploiting those social biases that might be in the underlying data (Lucy and Bamman 2021). For example, in an AI generative model, when it is tasked with generating a piece of work that has feminine characters it would have a higher chance of including topics such as family, children, and other terminology and vocabulary that can be construed as “weaker” in nature by societal norms. However, the results would be different when using more muscular vocabulary.

In a talk conducted by Chang (2019) when an automatic filtering system was tasked with candidate selection based on their gender and race, it would often reinforce societal biases and possibly perpetuate the systematic unfairness onto the selection. In another study conducted by Bolukbasi et al. (2016) demonstrated that even machine learning models that have been trained on news articles displayed gender stereotypes. This behaviour is alarming, as continuous enforcements of these biases will only serve to amplify them. This is due to the direction in which the discussing or article is heading also known as the word embedding, and secondly, gender neutral terms are separated from gendered definitions in the word embedding as well.

Models use the text corpus as the methods in which they gather an understanding of word meanings, meaning, the text corpora of the word co-occurrences present a dictionary for the machine learning algorithm that it uses to understand the underlaying meaning of the data. Dictionaries aren't perfect, as words with similar semantic meanings tend to have vectors that are close together, and additionally, the vector differences between terms can at times show a relationship between words. For example, if we were to provide a machine learning model as question to fill out the missing value for the following sentence "Man is to king as woman is to $X$", then the model might find that $x$ is equivalent to queen and deems it as the best and most optimal answers (Bolukbasi et al., 2016).

# 10. Data Visualization and Exploration

We visualize the data to explore the data, represent it in a meaningful way, and interpret the results in a better way. Visualization is performed differently foe each topic model, this includes the type and complexity of the model, the size of the corpus and how diverse it is, and the purpose of the visualization. We must also understand the type of audience to whom we will be presenting our visuals to, individuals with different backgrounds will interpret data differently, this is why it's important to understand our target audience. Since the audience for this paper is for academics from different scientific backgrounds then we must include a diverse selection of visual models and methods.

There are common visualization methods for topic models such as:

Word clouds, bar charts, matrix plots, and interactive dashboards. Word clouds are a graphical representation of the most repeated and frequent words in a document or topic, the relevancy and importance are demonstrated by the size of font for each word and can also include different colours. Word clouds are useful to have a generalized overview of the recurring topics in the corpus, however, relationships between words or topics can't be seen, nor how the topics are distributed across the corpus.

Bar charts are another visualization method that is used to compare the presentation of the topics in the documents or the corpus. We use it to show how dominant or diverse each topic is across multiple different documents. Bar charts are also useful to see contrasts between different topics and compare the differences between topics. However, similarly to word clouds, the details of each topic aren't demonstratable, such as the exact words that compose those topics.

Interactive dashboards are the web-based applications that allows a user to better interact with the topic model by creating real-time interactions with the model using widgets, such as sliders, buttons, drop-down menus, and maps of the topics. It can allow users to view the data from multiple perspectives and from different levels of detail, such as showing the overall distribution of topics across different corpuses and documents. In addition, we can also view the composition of topics by their words and subtopics. Interactive dashboards are a flexible and engaging way to explore and communicate a topic model. In this project we used several libraries to visualize our topic model, such as Gensim, pyLDAvis, and matplotlib.py.

Gensim is a Python library that provides tools for natural language processing and topic modelling. It can create and manipulate various types of topic models, such as latent semantic analysis (LSA), latent Dirichlet allocation (LDA), or hierarchical Dirichlet process (HDP).

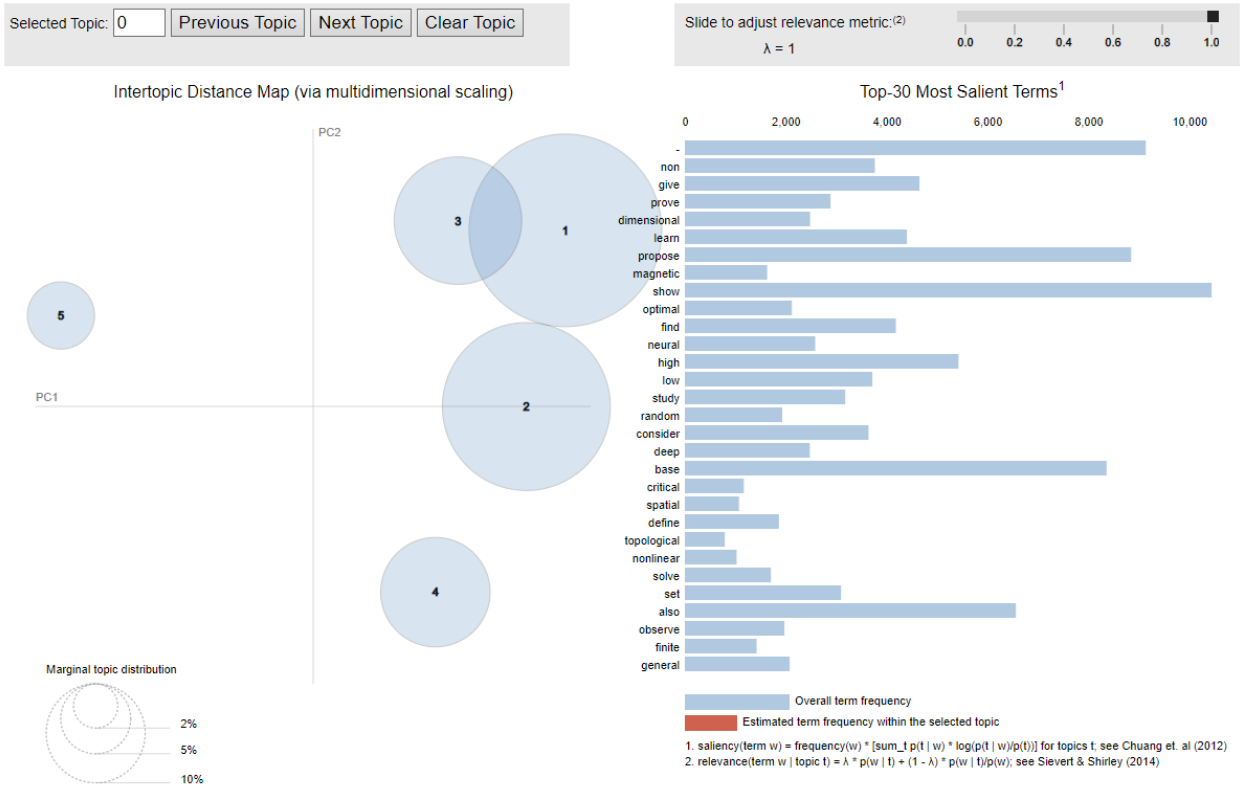
We used Gensim to create an LDA model from our corpus of documents. We chose LDA because it is one of the most popular and widely used methods for topic modelling. It assumes that each document is a mixture of topics, and each topic is a distribution of words. It uses a probabilistic approach to infer the latent topics and their proportions in each document from the observed words and their frequencies.

Matplotlib.py is a Python library that provides tools for creating static and interactive plots and charts. It can produce various types of visualizations, such as line plots, scatter plots, histograms, pie charts, etc.

We used matplotlib.py to create bar charts that show the distribution of topics across different groups of documents. We grouped our documents by some metadata variables, such as date, author, genre, etc. We used bar charts because they are simple and intuitive to compare different groups by their topic proportions.

**10.1 PyLDAvis**

PyLDAvis is a Python library that provides tools for creating interactive web-based dashboards for LDA models. It can generate interactive plots that show the global overview and local details of an LDA model. In addition, it uses matplotlib.py for creating word clouds and bar charts, and d3.js for creating network graphs and matrix plots.
visualizations of topic models. We demonstrate how to use these libraries to perform the following steps: (1) pre-process the text data and create a document-term matrix; (2) train a latent Dirichlet allocation (LDA) model using Gensim; (3) plot the topics and their word distributions using matplotlib.py; and (4) create an interactive visualization of the topics and their document distributions using pyLDAvis.



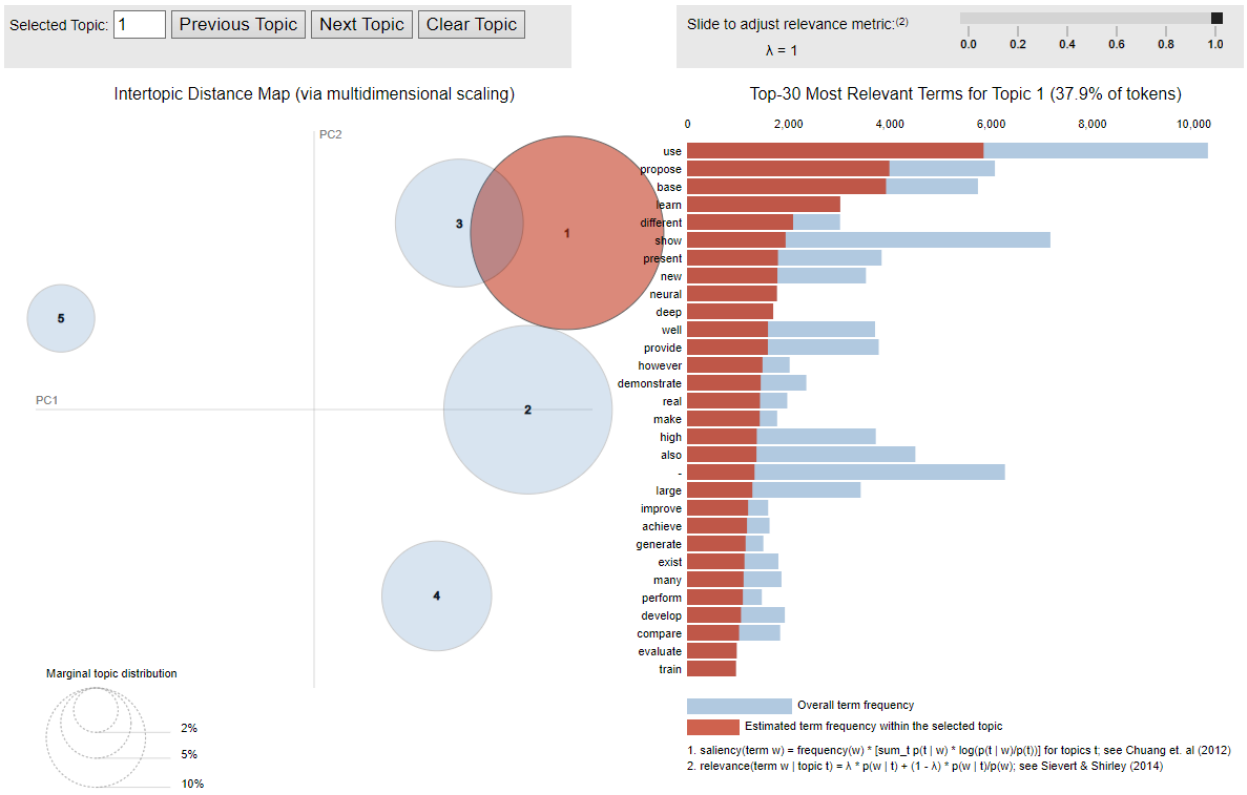

*(Figure 6, interactive dashboard using pyLDAvis)*

In figure 6, the interactive dashboard is created by the pyLDAvis and Gensim libraries. The topics are interpreted in an "Intertopic" distance map in the form of clusters, we can see that some topics have an interlaced relationship with other topics such as topic 1 and topic 3. And on the side, we have the top 30 repeating terms in the corpus. The blue line representing the term frequency and the red line (which will be demonstrated in the next figure) represents the estimated term frequency within the selected topic. In addition, the relevance metric can be adjusted based on the user's requirements, it's set to $\lambda = 1$. Relevance is calculated using the following formula:

$$relevance(term\ w\ |topic\ f) = \lambda \times p(w|f) + (1 - \lambda) \times \frac{p(w|f)}{p(w)}$$

And the term frequency (saliency) is calculated using the following formula:

$$saliency\ (term\ w) = frequency(w) \times \left[\sum_t p(t|w) \times \log\left(\frac{p(t \mid w)}{p(t)}\right)\right]$$

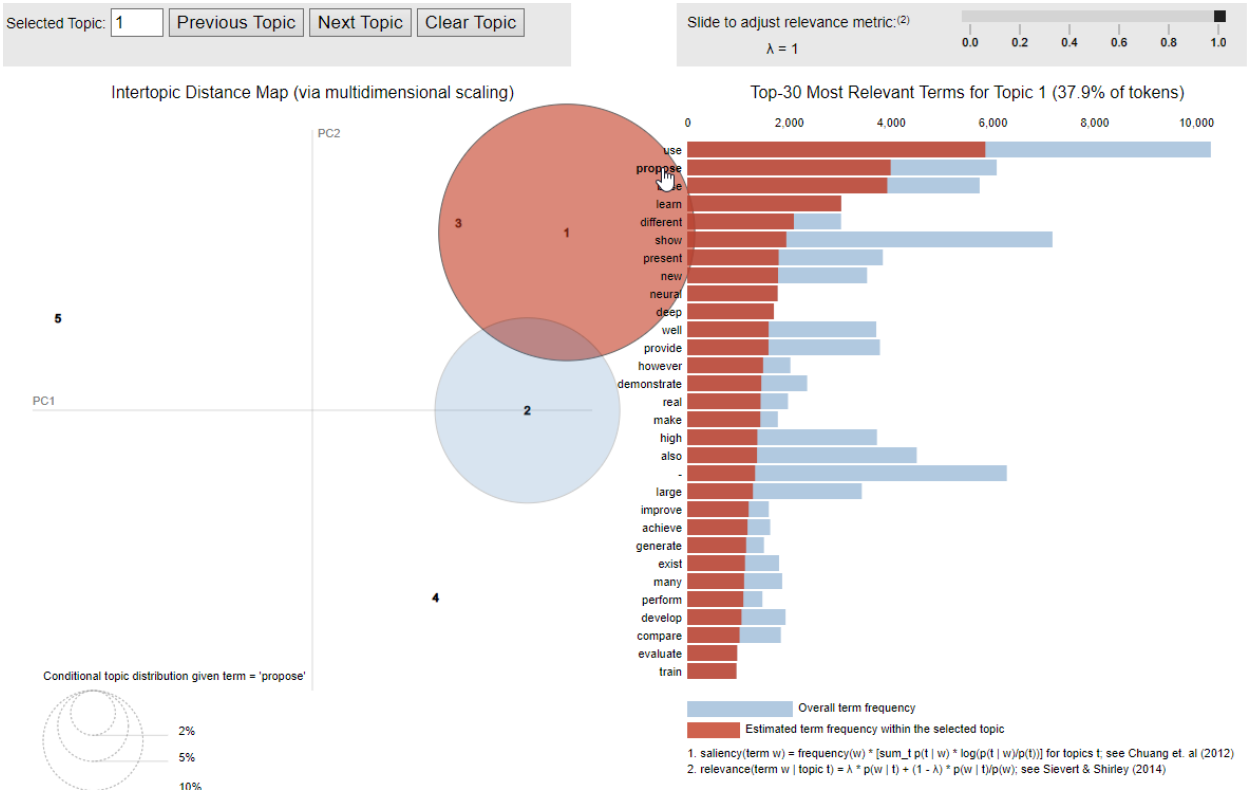



*(Figure 7, topic 1 selected and it's terms)*

In figure 7, when exploring the frequent terms that are being present in topic 1, we can see a terminology that relates to neural networks, models, deep learning, and other terms that hint to Computer Science. We know that this is accurate because when we refer to our table of recurring topics (as seen in figure 9) we see that computer science is the first topic in our data set. As this is an interactive dashboard we can interact with the different topics and see that certain terms have a relationship with other topics.

| | index | Computer Science | Physics | Mathematics | Statistics | Quantitative Biology | Quantitative Finance |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0.053478 | 0.000718 | 0.008020 | 0.000000 | 0.000000 | 0.011849 |
| 1 | 1 | 0.014648 | 0.004958 | 0.000000 | 0.060959 | 0.000000 | 0.000000 |
| 2 | 2 | 0.000000 | 0.033019 | 0.000000 | 0.001113 | 0.000000 | 0.006254 |
| 3 | 3 | 0.000000 | 0.048890 | 0.012750 | 0.000000 | 0.000000 | 0.027562 |
| 4 | 4 | 0.032414 | 0.000000 | 0.003643 | 0.000000 | 0.001994 | 0.011012 |

(Figure 8, list of recurring topics in our data set, with their ID and index)

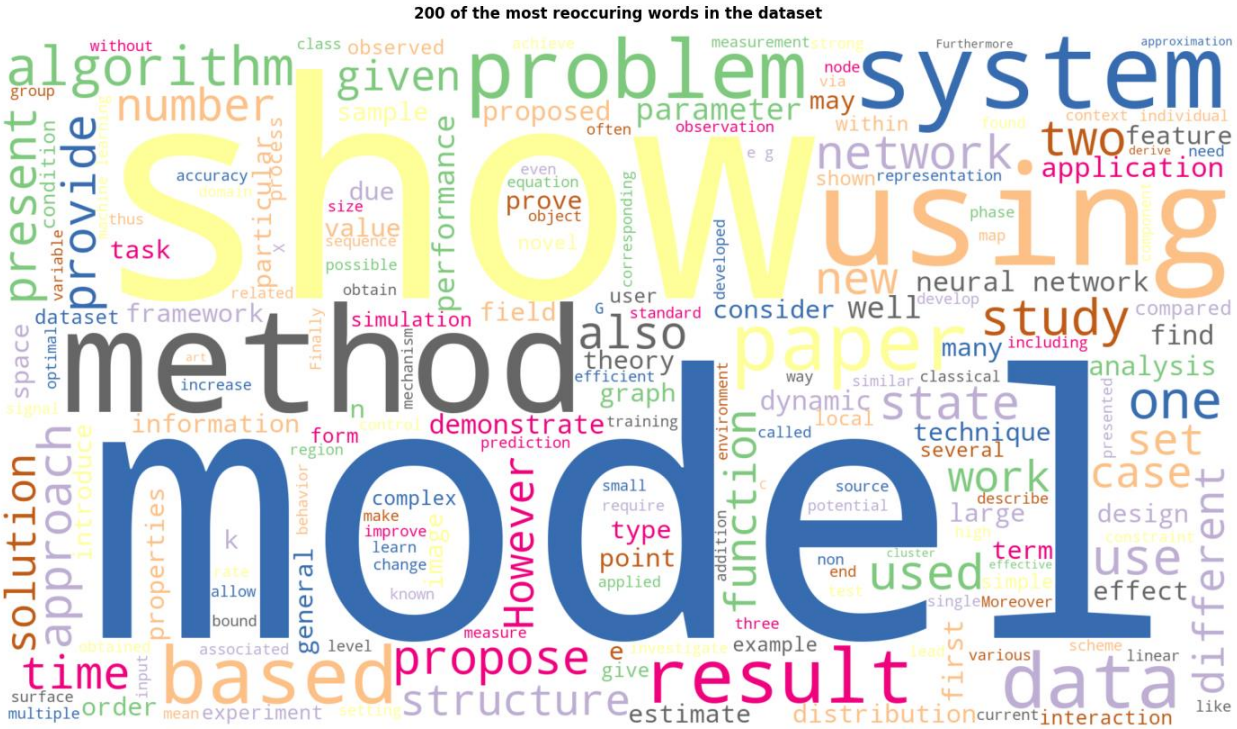



(Figure 9, certain terms can be found in other topics example)

For example, the word propose is mainly topic 1's recurring term, however, it can be seen used in topic 2 which is Physics in this data set. Using interactive dashboard is a great way to find how certain terms relate to other topics and can also provide us with a reason as to why the topic model can have difficulty at times to distinguish between different topics.

In general, STEM topics will have an overlapping nature, as for example, computer science uses aspects for Mathematics, Physics, and Statistics that can cause the model to be confused as to what it should categorise and label certain documents. In this case, having datasets with broader topics could yield better results and accuracy.

### *10.2 Word-clouds*

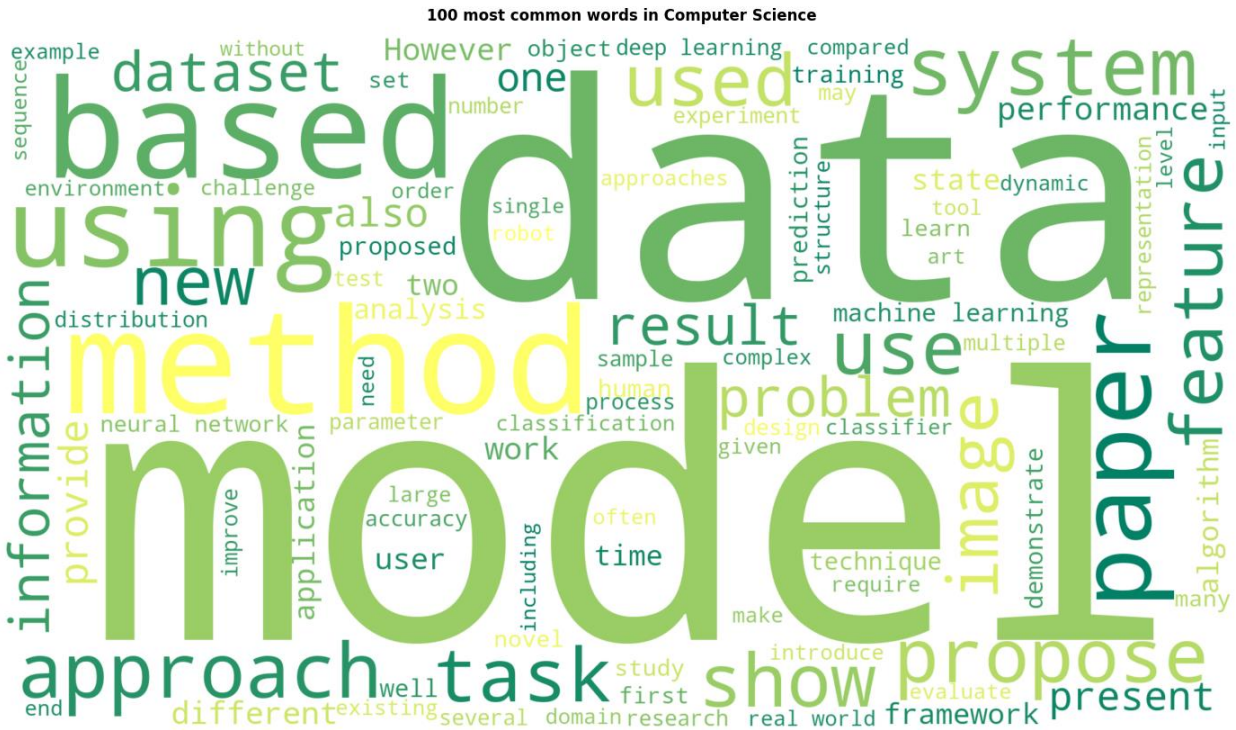



(Figure 10, word cloud of the top 200 most recurring words)

In figure 10, we showcase the top 200 most recurring words in our dataset, in our case the datasets contain research articles on STEM topics, visualizing word clouds can provide valuable insight into topics and topics that are frequently discussed in research. For example, if we first organize the text data and find the top 200 words, we can see that words like “model”, “method”, “system”, and “problem” are among the common recurring topics. Word clouds can also help us identify specific topics that are commonly discussed in research articles. For example, if we find that "result" or "data" are among the most common terms, it shows that these are important topics in the field. In addition, word clouds can be used to help us identify possible trends or changes in search that occur over time.

For example, if we compare the word cloud of a dataset of STEM research articles from the 1990s to the 2020s, we can see that the most common words have changed from "microprocessor" and "algorithm" to “artificial intelligence" and "big data". These changes in terminology can be due to a multitude of factors, such as advances in technology, changes of interest or a more defined interest in certain topics that cause a change in terminology to be more specific or broader, and economical change can also influence change.

Changes in research interest is also another major factor to reasons as to why terminology has changes over time. Previously, technological and computer system research had different terminology for different subjects, for example, storage in the 90s and early 2000s was mainly on floppy disks and CDs, thus, research at that time was more focused towards those topics and devices. However, in the 2020s, we have solid state drives, hard drives, NAND chips, and many more different ways to store information. Hence, in order to ensure that our word clouds and data in general are appropriate then we would need to perform a pre-processing of the data that is being analysed by the model.




(Figure 11, the most common words in Computer Science)

And in figure 11, we can see that the most common words in computer science relate to machine learning and artificial intelligence, that is due to our data set being compromised of more recent research papers, and a focus on machine learning. However, in an Ethic of Computing study by Stahl, Timmermans and

Mittelstadt (2016) where they utilize machine learning model to conduct a survey of computer science and technology academic literature using a data set from the early 2000s, and when exploring their dataset we can notice that their word cloud contains terminology that is older and out-of-date as seen in the figure below.

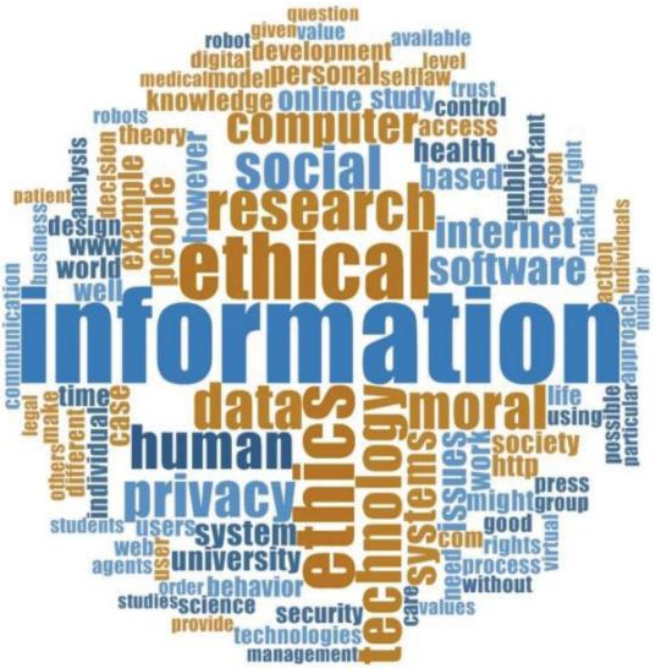



*Figure 12, Word cloud from Stahl, Timmermans and Mittelstadt (2016) used to demonstrate the difference between research interest before to research interest nowadays)*

We notice that the most repeated topics fall within ethics, information, and data. Furthermore, the rest of the topics are generally broader in their nature, this also demonstrates the change of research interest to either be more specific, or to tackle more relevant topics that are trending in more recent years.

For this reason, it is important to understand the topics that are discussed in any dataset, which is why we perform visualization to the data given. Visualizing the data provides us with an insight on how to tackle and configure our dataset. If we notice misalignment or an aspect or many aspects that are lacking in our data set, then we can perform a better pre-processing method that encompasses more filters and restrictions,

to ensure that the data that we deal with is accurate, concise, has meaning, and can be understood by the researchers and by the model.

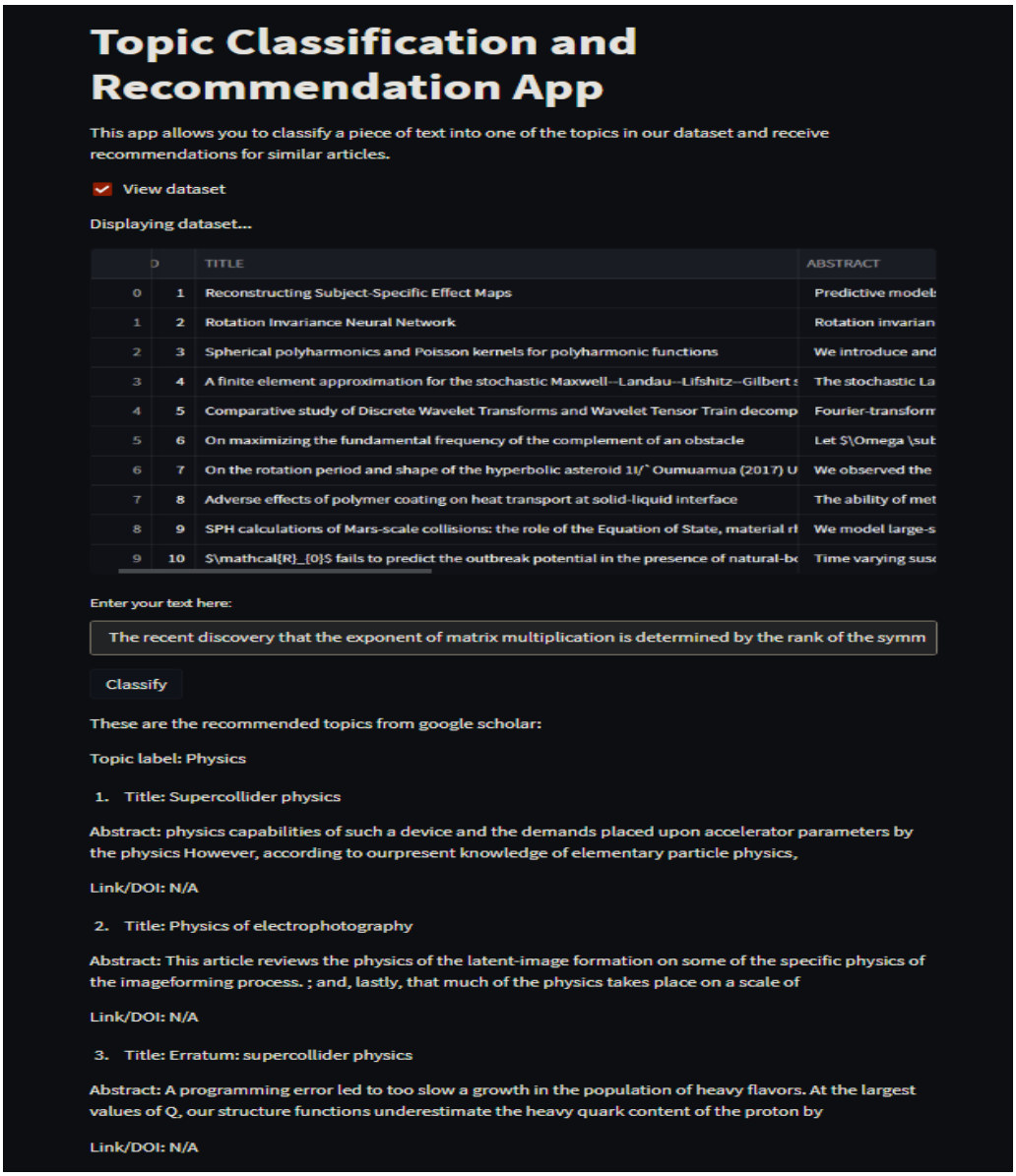
# 11. Methodology

## 11.1 Front-End Implementation

The open-source application framework Streamlit is used to construct the application's front-end. Using the tool Streamlit, users can create dynamic web applications based on data. Python scripts
for data modelling are the only scripts required to construct a user interface. Streamlit uses HTML, CSS, and JavaScript, to create web applications that are great to interact with and are simple to make. Users create the functions that they would like to run using python, and Streamlit processes the application and creates the interface per the user’s requirements.

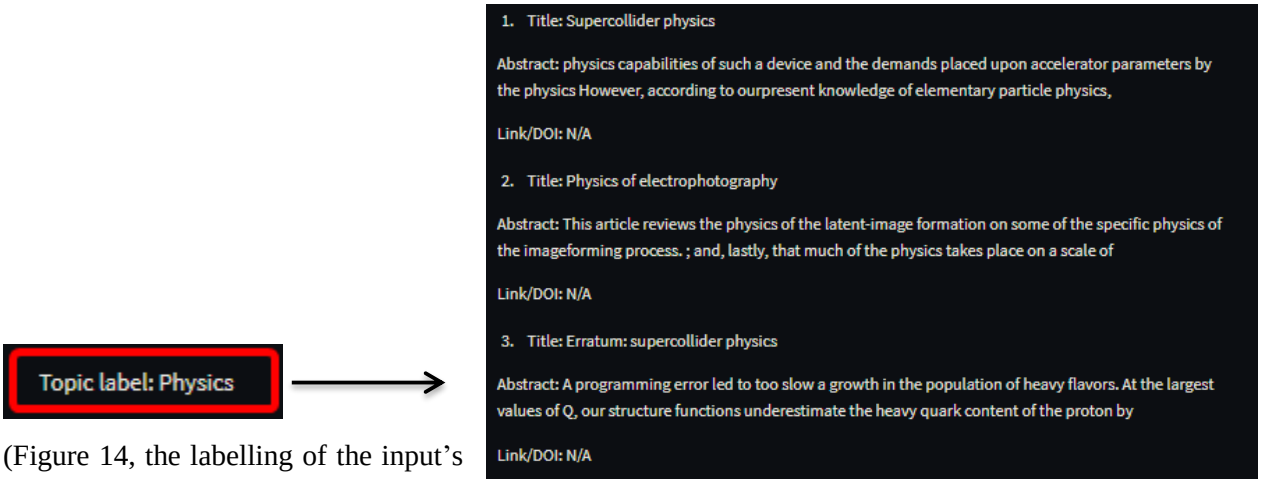
We used Streamlit to create an interface that will allow the user to input an abstract that they would want to get the general idea of what the topic of the research article is discussing. In addition, we can also recommend other research articles based on the topic. The web application is hosted locally, however, with further implementation we will be using a web server to host the website.

On the website we will have a brief preview of the data set that was used, and the diagrams that explain the topics, and in addition, we will display the result of the model, which will be the topic of the user’s input and the recommended research articles.




*(Figure 13, web application front-end interface and results)*

The figure above the layout of the website, we have a preview of the data set which is the top 10 items in the data set, and then we have an entry section for the user to input their abstract for the model to analyse. After the model performs the text cleaning and pre-processing, it would then generate the abstracts topic as seen below:




(Figure 14, the labelling of the input's topic and recommended articles based on the topic label.)

After acquiring the topic for the abstract, we will recommend similar abstracts based on the topic from search results that can be accessed from google scholar. This way we can help the user find research material quicker and more efficiently. We also provide a summary of the suggested abstract with a link or the DOIs of the article.

We have limited the recommendation to a minimum of 3 recommendations, however, the number can increase based on demand, for our demonstration three was sufficient to show the effectiveness of the model.

## 11.2 Environment setup

We have 3 main pages that we have set up, a clustering page using TF-IDF, NMF, PCA, and LSA, and an LDA and supervised model page, and lastly, the web application page.

## 11.3 Data selection process.

### Clustering using TF-IDF, NMF, PCA, and LSA models.

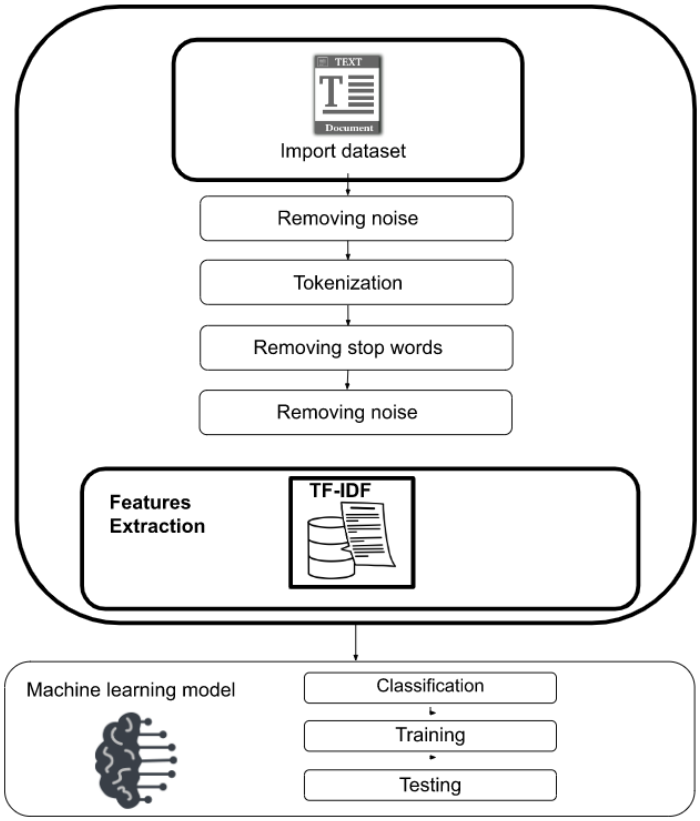
We have already explained how these models work in theory, and now we will discuss the process that we took to precure the models, results, and the data set modification and alteration that were made to fit our purposes.

We first begin by importing our data set and the libraries that will be used for processing, cleaning, and visualization. The data set that we used can be accessed through this link. It is comprised of five STEM topics being Computer Science, Physics, Statistics, Mathematics, Quantitative Biology, and Quantitative Finance. The data set can be further expanded to include more topics, however, the sheer amount of data would then be unable to run on our devices, thus, for this project it was decided to favour a more restricted data set in terms of topics than to have a broader and expansive one. For future adaptation, the data set would be merged with other data sets to create a "Super data set" that encompasses more topics and subjects and will require better feature management and a more complex pre-processing set up. Nevertheless, for this project, our data set is more than sufficient to perform the tasks that are required.

## 11.4 Pre-processing of the data

In the pre-processing section of this project, we will discuss the process that we performed to clean the data, vectorize it, and extract the relevant information from it, we will begin with the data cleaning as it is the first step in any NLP related machine learning project.

In the figure below, we are representing the data collection and feeding process. It begins by collecting the data from the data set, and then removing the noise, then performing the tokenization process, and afterwards we remove the stop words from the data set, and then we perform the stemming/lemmatizing process. The features are then extracted using the TF-IDF vectorizer and Count Vectorizer. Lastly, the data is fed to the models to train them and test their accuracy and results. In our project we also used supervised models to classify future data based on the results of our model training. Another important step that is required when dealing for example with translation text or text that isn't uniformed is Normalization. In a study case by Taef Alkhales et al. (2022) when working with a text that is of a different language than English, then certain steps should be taken to normalize the data, to make it uniform to the standards. In their example, they had to perform normalization to Arabic text, and in Arabic there is a characteristic of elongating words to express more meaning, this is called "Tat'weel" in Arabic and means to stretch and elongate. And for that, we need to perform a cleaning of the text to keep the uniformity, thus ensuring that our data is suitable and ready to be tokenized.




(Figure 15, the process of data collection, processing, and exporting to the machine learning model.)

### 11.4.1 Data cleaning

The process begins with cleaning the data by first transforming the text data into lowercase characters as it will help later when the data must be parsed, and then will remove any numbers or punctuation in the text. In addition, tokenization and lemmatizing processes are conducted to breakdown the sentences into $x$ number of words and restoring the words to their root word.

The lemmatizing process is performed to ensure that the model wouldn't give similar words such as store, stores, and stored different weights, by bringing the words to their origins (Stores → store).

Furthermore, stop words such as “a” and “the” will be removed to increase search performance, as stop words don’t provide any meaningfulness to the text data.

### 11.4.2 Count Vectorization process

Count Vectorizer is a term that refers to the process of breaking down a sentence or any text into words by doing pre-processing tasks such as converting all words to lowercase and thereby removing special

characters. That is due to NLP models not being able to understand textual data and the data must be transformed to numerical values as the models, as that is the only type of values that can be accepted by the model, thus, the textual data must be vectorized. In this project we will be performing two types of Count Vectorization.

The BOW (bag of words) model is a vectorized version of the array of words. It will convert the sentences into a "bag of words" with no meaning. For the first BOW model we will be running it using the cosine similarity function. This function will compute the similarity as the dot product (normalized) of the X and Y values. We will also be using the BOW model with TF-IDF (Term Frequency (TF) — Inverse Dense Frequency (IDF)) is a method to determine the meaning of sentences composed of words that overcomes the limitations of the Bag of Words approach, which is useful for text categorization or assisting a machine to interpret words in numbers.

We will be performing a dimensionality reduction process to the code which mean that the data is transformed from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some significant properties of the original data, ideally close to its intrinsic dimension.

For our model we used from the SKlearn the feature extraction functions CountVectorizer and TfidfVectorizer, these functions allow us to fit and transform the models to extract the text data from the data set and process it.

# 12. Model tuning and setting the hyperparameters.

## 12.1 Unsupervised-learning models.

Different model tuning is necessary since several models are of differing baseline assumptions, architectures and optimization targets which can affect their ability to perform on several tasks and datasets. We can improve their ability to perform the specific task and data at hand, while ensuring that we are using the best model available in this problem by tuning the hyperparameters of each model.

In addition, there are various data sets and tasks which may vary according to their size, robustness, complexity, or noise characteristics as well as the desired levels for interpretation, scalability, and accuracy of models. For this reason, it is important that the hyperparameters of each model are adjusted in such a way as to meet the specific requirements and limitations imposed by the situation while avoiding overcomplicating or undervaluing the data.

The models that were used for the unsupervised classification of the models were, LSA, NMF, LDA, and TF-IDF. We will be using the SKlearn library for the PCA function to perform a linear dimensionality reduction by projecting the data into a lower dimensional space using SVD (Singular Value Decomposition). We set the value of the components of the NMF model to 6 prior to the PCA analysis. we will now plot and fit the results of the topics. In the PCA analysis, we will be using the same number of components as the model and fit that to our model. In our testing, we found that the best fitting number of components is six for our NMF model using PCA. Furthermore, LSA demonstrated similar results using the same number of components, however, NMF demonstrated the better results overall.

As for the LDA model then we will begin similarly to the other methods, however, we will also be using a Natural Language Toolkit (NLTK) library to remove stop words from text data. Stop words are words that occur frequently in a language but do not carry much meaning, such as "a", "an", "the", "in", "of", etc. The first two lines of the code import the stop words module from NLTK and create a variable stop word that contains a list of English stop words.

Next, a function “remove_stopwords” is defined that takes a single argument text, which is a string containing text data. The function splits the input text into an array of words using the split method and then uses a list comprehension to remove any words that appear in the stop words list. The filtered words are then joined back together into a string using the join method and returned. Finally, the apply method is used to apply the “remove_stopwords” function to every row in the 'ABSTRACT' column of the “train_df” Data Frame. This removes the stop words from the text data in each row and updates the 'ABSTRACT' column in-place with the cleaned text.

We will use the SpaCy library to perform lemmatization on a list of input texts. Lemmatization is the process of reducing words to their base or dictionary form, which can be useful for standardizing text data and reducing noise in natural language processing tasks. However before beginning to use the Spacy library, you must first install the required tools to begin using the Spacy library. This will also be where we perform the tokenization process. After we finish with tokenizing, lemmatization, and removing the stop words, we

will then create a dictionary, also called a corpus, of all the text data that the model will use to train. The library that we use for the LDA model is called Gensim (discussed the library beforehand). We will create two LDA models to validate the accuracy of the model, and then proceed with whichever model we deem fit.

## 12.2 Supervised Learning Models

Lastly, we'll proceed to create the supervised models that will classify the data from our unsupervised models. Three supervisor machine learning models were implemented using Python, SKlearn, and Grid-SearchCV libraries to determine the classification models. The classifiers were as follows: SVC, Random Forest Classifier (RFC), and Multinomial Naïve Bayes (MNB) models.

### 12.1.1 C-Support Vector Classification (SVC)

SVC is a classifier that uses risk minimization theory to find the optimal separating hyperplane within the feature space. Simple SVC is used for linear regression and classification problems. To use the SVM classifier in a non-linear space, kernel functions must be used (e.g., RBF, Linear, Poly, and Sigmoid). The following equation demonstrates the RBF kernel: $K(X,Y) = \exp(-\frac{||X-Y||^2}{2\sigma^2})$ where "$\left|\left|X - Y\right|\right|^2$" is the Euclidean distance between the two points "$X$" and "$Y$," and "$\sigma$" is the hyperparameter and variance.

### 12.1.2 Random Forest Classifier (RFC)

Random Forest Classifier is a combination of decision trees predictors that predict the class label by randomly generating a forest. The forest is a collection of multiple decision trees, and each tree has the value of a random vector that is sampled independently from one another. Each tree is then distributed equally among all trees to share the results that it gathered. The final classification is based which forest had the best result and selected by the majority.

### 12.1.3 Multinomial Naive Bayes (MNB)

And lastly, the Multinomial Naive Bayes (MNB) method is a probabilistic classification model based on the Bayes theorem and the assumption of feature independence. It is a Naive Bayes variation created particularly for text classification issues, with feature vectors representing the frequency of occurrence of each phrase in the document. As this is a probabilistic classification model, the formula in which it operates in is based on probability, and is as such:

$$P(C\_k \mid x\_1, x\_2, \ldots, x\_n) = P(C\_k) * P(x\_1, x\_2, \ldots, x\_n \mid C\_k) / P(x\_1, x\_2, \ldots, x\_n)$$

Where $P(C_k)$ is the prior probability of class label $(C_k)$. which is estimated from the training set as the proportion of documents with label $(C_k)$. And $P(x\_1, x\_2, \ldots, x\_n \mid C\_k)$ is the likelihood of the feature vector x given class label $(C_k)$., which is estimated from the training set as the probability distribution of the frequency of each term in the documents with label $(C_k)$.

The evaluation of classifiers was measured with accuracy, precision, recall, and the use of an F1- score. We calculated the performance measurement for the classifiers with the following equations:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision \times Recall}$$

The variables in these equations have the following meaning, “TP” stands for true positive, while “TN” represents a true negative, and lastly, “FP” and “FN” stand for false positive and false negative respectively. The F1 score is calculated by taking the weighted average of recall and precision, and precision and recall are measured by the false positives and negatives of the classifying model.

# 13. Results

In this section we discuss the findings of the results of testing the classifiers. In table 3, we demonstrate the results of running a test on the accuracy, precision, recall, and F1 score of the models. The data set was split to an 8:2 ratio, where 80% of the dataset was used for training and 20% was for testing. Based on the results we concluded that the best model to use is the SVC with an average score of 90% accuracy. This is the model that was used in the web app to classify the data.

Table 3: The results of testing the models.

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| **Multinomial NB** | 0.76924 | 0.82675 | 0.5957 | 0.58333 |
| **RFC** | 0.60429 | 0.4598 | 0.4369 | 0.40535 |
| **SVC** | 0.90536 | 0.92288 | 0.87072 | 0.89209 |

And now we can also compare how each model faired in terms of each topic in the data set. This can be seen in the tables below:

Table 4: The results of the MultinomialNB model to the topics in the dataset.

| *Multinomial NB Model* | *Precision* | *Recall* | *F1-Score* | *Support* |
|---|---|---|---|---|
| *Computer Science* | 0.62 | 0.89 | 0.73 | 907 |
| *Mathematics* | 0.91 | 0.89 | 0.90 | 1093 |
| *Physics* | 0.86 | 0.86 | 0.90 | 922 |
| *Quantitative Biology* | 1 | 0.02 | 0.05 | 161 |
| *Quantitative Finance* | 0.72 | 0.78 | 0.75 | 768 |
| *Statistics* | 0.85 | 0.12 | 0.21 | 344 |
| | | | | |
| *Accuracy* | | | 0.77 | 4195 |
| *Macro average* | 0.83 | 0.60 | 0.58 | 4195 |

| | | | | |
|---|---|---|---|---|
| *Weighted Average* | 0.80 | 0.77 | 0.74 | 4195 |

Table 5: The results of the Random Forest Classifier (RFC) model to the topics in the dataset.

| *RFC* | *Precision* | *Recall* | *F1-Score* | *Support* |
|---|---|---|---|---|
| *Computer Science* | 0.56 | 0.67 | 0.61 | 907 |
| *Mathematics* | 0.53 | 0.98 | 0.69 | 1093 |
| *Physics* | 0.75 | 0.72 | 0.73 | 922 |
| *Quantitative Biology* | 0 | 0.00 | 0.00 | 161 |
| *Quantitative Finance* | 0.92 | 0.26 | 0.40 | 768 |
| *Statistics* | 0.00 | 0.00 | 0.00 | 344 |
| | | | | |
| *Accuracy* | | | 0.60 | 4195 |
| *Macro average* | 0.46 | 0.44 | 0.41 | 4195 |
| *Weighted Average* | 0.59 | 0.60 | 0.55 | 4195 |

Table 6: The results of the C-Support Vector Classification (SVC) model to the topics in the dataset.

| *SVC* | *Precision* | *Recall* | *F1-Score* | *Support* |
|---|---|---|---|---|
| *Computer Science* | 0.84 | 0.94 | 0.89 | 907 |
| *Mathematics* | 0.92 | 0.94 | 0.93 | 1093 |
| *Physics* | 0.93 | 0.91 | 0.92 | 922 |
| *Quantitative Biology* | 1.0 | 0.73 | 0.84 | 161 |
| *Quantitative Finance* | 0.92 | 0.88 | 0.90 | 768 |
| *Statistics* | 0.93 | 0.83 | 0.87 | 344 |
| | | | | |
| *Accuracy* | | | 0.91 | 4195 |
| *Macro average* | 0.92 | 0.87 | 0.89 | 4195 |
| *Weighted Average* | 0.91 | 0.91 | 0.91 | 4195 |

Based on these results we deem SVC to be the most optimal and accurate model that we have tested, and thus will be used from here on out. We conclude that it's the best based on the F1-Score, and from our test we can see that it has the highest score out of the models that we have tested.

# 14. Restrictions and limitations

NLP libraries and algorithms are very time consuming and resource exhaustive. On average on an 8 core CPU processor with 16 Gigabytes of ram, it would take from 5 minutes to hours to digest and process the data. In addition, these libraries have a knack of breaking or simply ceasing to work for no apparent reason. An example of this is the pyLDAvis library. In addition, setting up the environment is also tricky for beginners as I have found that there are certain pre-requisites to each of the libraries that I have used, thus I provided explanations and the steps necessary to set up the environment in the exact way that I had it to at least ensure that it would work.

Another issue that NLP is known to have optimisation problems, the algorithms are still being developed, there has been an improvement, however, overall, it is still poorly optimized. And lastly, the web application takes a while to load that is due to it running the classification process within then landing page and not off-site and loading the answer, and the fetching of recommendations takes a little while as it scours google scholar three related articles.

# 15. Conclusion

In conclusion, the use of supervised and unsupervised models for topic modelling has proven to be a valuable tool in the classification of user inputs into their relevant topics. The unsupervised models of LDA, LSA, NMF, and TF-IDF have demonstrated their effectiveness in discovering the underlying topics within our dataset without the need for prior knowledge or labelling, and with the use of clustering to create labels for the data points in the data set. On the other hand, the supervised models of SVC, MNB, and RFC have shown their ability to accurately classify user inputs into pre-defined categories with high precision and recall and displaying it over our web application.

Furthermore, the results of our study highlight the importance of selecting the appropriate model for the task at hand, based on the available data and research questions. The combination of both unsupervised and supervised models can offer a comprehensive approach to topic modelling that can be useful in a variety of fields, including education, data science, and research material analysis. Future research can explore the use of other topic modelling techniques and their applications in different domains.

# 16. Reference list

Alkhales, T., A. Almakki, R., E. Alnajim, S., K. Almarshad, S., S. Alhasaniah, R., S. Aljameel, S., A. Almuqhim, A. and A. Musleh, D. (2022). Twitter Arabic Sentiment Analysis to Detect Depression Using Machine Learning. Computers, Materials & Continua, 71(2), pp.3463–3477. doi:https://doi.org/10.32604/cmc.2022.022508.

Arora, S., Ge, R. and Ankur Moitra (2012). Learning Topic Models -- Going beyond SVD. Foundations of Computer Science. doi:https://doi.org/10.1109/focs.2012.49.

Bergamaschi, S. and Po, L. (2014). Comparing LDA and LSA Topic Models for Content-Based Movie Recommendation Systems. Lecture notes in business information processing. doi:https://doi.org/10.1007/978-3-319-27030-2_16.

Berry, M.W., Mohamed, A. and Bee Wah Yap (2019). Supervised and Unsupervised Learning for Data Science. Springer Nature.

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V. and Kalai, A. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. arXiv (Cornell University). doi:https://doi.org/10.48550/arxiv.1607.06520.

Chang, K.-W. (2019). Tutorial: Bias and Fairness in Natural Language Processing. [online] web.cs.ucla.edu. Available at: https://web.cs.ucla.edu/~kwchang/talks/emnlp19-fairnlp/ [Accessed 23 Apr. 2023].

Churchill, R. and Singh, L. (2022). The Evolution of Topic Modeling. ACM Computing Surveys. doi:https://doi.org/10.1145/3507900.

Crossno, P., Wilson, A., Shead, T.M. and Dunlavy, D.M. (2011). TopicView: Visually Comparing Topic Models of Text Collections. OSTI OAI (U.S. Department of Energy Office of Scientific and Technical Information). doi:https://doi.org/10.1109/ictai.2011.162.

Dahl, G.E., Yu, D., Deng, L. and Acero, A. (2012). Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20(1), pp.30–42. doi:https://doi.org/10.1109/tasl.2011.2134090.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North. Association for Computational Linguistics, pp.4171–4186. doi:https://doi.org/10.18653/v1/N19-1423.

Grace, K., Salvatier, J., Dafoe, A., Zhang, B. and Evans, O. (2018). Viewpoint: When will AI exceed human performance? Evidence from AI experts. Journal of Artificial Intelligence Research, 62, pp.729–754. doi:https://doi.org/10.1613/jair.1.11222.

Greene, D., O'Callaghan, D. and Cunningham, P. (2014). How Many Topics? Stability Analysis for Topic Models. Machine Learning and Knowledge Discovery in Databases, pp.498–513. doi:https://doi.org/10.1007/978-3-662-44848-9_32.

Hecking, T. and Leydesdorff, L. (2018). Topic Modelling of Empirical Text Corpora: Validity, Reliability, and Reproducibility in Comparison to Semantic Maps. arXiv (Cornell University). doi:https://doi.org/10.48550/arxiv.1806.01045.

Hecking, T. and Leydesdorff, L. (2019). Can topic models be used in research evaluations? Reproducibility, validity, and reliability when compared with semantic maps. Research Evaluation, 28(3), pp.263–272. doi:https://doi.org/10.1093/reseval/rvz015.

Jivani, A.G., 2011. A comparative study of stemming algorithms. Int. J. Comp. Tech. Appl, 2(6), pp.1930-1938.

Kalepalli, Y., Tasneem, S., Phani Teja, P.D. and Manne, S. (2020). Effective Comparison of LDA with LSA for Topic Modelling. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICICCS48265.2020.9120888.

Kee, Y.H., Li, C., Kong, L.C., Tang, C.J. and Chuang, K.-L. (2019). Scoping Review of Mindfulness Research: a Topic Modelling Approach. Mindfulness, 10(8), pp.1474–1488. doi:https://doi.org/10.1007/s12671-019-01136-4.

Kiersztyn, A. and Kiersztyn, K. (2023). The Impact of Data Preprocessing on Prediction Effectiveness. Lecture Notes in Computer Science, pp.353–362. doi:https://doi.org/10.1007/978-3-031-23492-7_30.

Landauer, T.K., McNamara, D.S., Dennis, S. and Kintsch, W. (2013). Handbook of Latent Semantic Analysis. Psychology Press.

Lauriola, I., Lavelli, A. and Aiolli, F. (2021). An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. Neurocomputing, Volume 470. doi:https://doi.org/10.1016/j.neucom.2021.05.103.

Li, C.-H., Kuo, B.-C. and Lin, C.-T. (2011). LDA-Based Clustering Algorithm and Its Application to an Unsupervised Feature Extraction. IEEE Transactions on Fuzzy Systems, 19(1), pp.152–163. doi:https://doi.org/10.1109/tfuzz.2010.2089631.

Likas, A., Vlassis, N. and J. Verbeek, J. (2003). The global k-means clustering algorithm. Pattern Recognition, 36(2), pp.451–461. doi:https://doi.org/10.1016/s0031-3203(02)00060-2.

Lopez, M.M. and Kalita, J. (2017). Deep Learning applied to NLP. arXiv:1703.03091 [cs]. [online] Available at: https://arxiv.org/abs/1703.03091.

Magnus Sahlgren (2020). Rethinking Topic Modelling: From Document-Space to Term-Space. Empirical Methods in Natural Language Processing. doi:https://doi.org/10.18653/v1/2020.findings-emnlp.204.

Mohammed, S. and Al-Augby, S. (2020). LSA & LDA Topic Modeling Classification: Comparison Study on E-books. Indonesian Journal of Electrical Engineering and Computer Science, 19(1). doi:https://doi.org/10.11591/ijeecs.v19.i1.

Neogi, P.P.G., Das, A.K., Goswami, S. and Mustafi, J. (2019). Topic Modeling for Text Classification. Advances in Intelligent Systems and Computing, pp.395–407. doi:https://doi.org/10.1007/978-981-13-7403-6_36.

Pittaras, N., Giannakopoulos, G., Papadakis, G. and Karkaletsis, V. (2020). Text classification with semantically enriched word embeddings. Natural Language Engineering, pp.1–35. doi:https://doi.org/10.1017/s1351324920000170.

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 952–961, Jeju Island, Korea, 12–14 July 2012. c 2012 Association for Computational Linguistics

Sathya, R. and Abraham, A., 2013. Comparison of supervised and unsupervised learning algorithms for pattern classification. International Journal of Advanced Research in Artificial Intelligence, 2(2), pp.34-38.

Schwenker, F. and Trentin, E. (2014). Pattern classification and clustering: A review of partially supervised learning approaches. Pattern Recognition Letters, 37, pp.4–14. doi:https://doi.org/10.1016/j.patrec.2013.10.017.

Setievi, F., Natalia, J., Theodore Raynard Tjhang, Ivan Sebastian Edbert and Derwin Suhartono (2022). A Comparative Study of Supervised Machine Learning Algorithms for Fake Review Detection. 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). doi:https://doi.org/10.1109/isriti56927.2022.10052860.

Sherstinova, T.Y., Mitrofanova, O., Skrebtsova, T., Ekaterina Zamiraylova and Kirina, M. (2020). Topic Modelling with NMF vs. Expert Topic Annotation: The Case Study of Russian Fiction. Lecture Notes in Computer Science. doi:https://doi.org/10.1007/978-3-030-60887-3_13.

Stahl, B.C., Timmermans, J. and Mittelstadt, B.D. (2016). The Ethics of Computing. ACM Computing Surveys, 48(4), pp.1–38. doi:https://doi.org/10.1145/2871196.

Uday Kamath, Zhanjiang Liu and Whitaker, J. (2019). Deep learning for NLP and speech recognition. Cham, Switzerland Springer.

Vayansky, I. and Kumar, S.A.P. (2020). A review of topic modeling methods. Information Systems, p.101582. doi:https://doi.org/10.1016/j.is.2020.101582.

Wang, D., Zhu, S., Li, T. and Gong, Y., 2009, August. Multi-document summarization using sentence-based topic models. In Proceedings of the ACL-IJCNLP 2009 conference short papers (pp. 297-300).

Wang, J. and Zhang, X.-L. (2022). Deep NMF topic modeling. Neurocomputing, 515, pp.157–173. doi:https://doi.org/10.1016/j.neucom.2022.10.002.

Xiong, H., Cheng, Y., Zhao, W. and Liu, J. (2019). Analyzing scientific research topics in manufacturing field using a topic model. Computers & Industrial Engineering, 135, pp.333–347. doi:https://doi.org/10.1016/j.cie.2019.06.010.

Zhang, L., Hu, X., Wang, B., Zhou, D., Zhang, Q.-W. and Cao, Y. (2022). Pre-training and Fine-tuning Neural Topic Model: A Simple yet Effective Approach to Incorporating External Knowledge. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). doi:https://doi.org/10.18653/v1/2022.acl-long.413.

Zhao, D. and Sun, J. (2017). Research on the Automatic Error Correction Model Combined with Artificial Intelligence for College English Essays. Advances in Intelligent Systems and Computing, 613, pp.41–51. doi:https://doi.org/10.1007/978-3-319-60744-3_5.

Zhao, G., Liu, Y., Zhang, W. and Wang, Y. (2018). TFIDF based Feature Words Extraction and Topic Modeling for Short Text. Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences - ICMSS 2018. doi:https://doi.org/10.1145/3180374.3181354.

Zhou, Z.-H. and Takashi Washio (2009). Advances in machine learning : first Asian conference on machine learning, ACML 2009, Nanjing, China, November 2-4, 2009 ; proceedings. Berlin ; New York: Springer.

Zhu, L. and Cunningham, S.W. (2022). Unveiling the knowledge structure of technological forecasting and social change (1969–2020) through an NMF-based hierarchical topic model. Technological Forecasting and Social Change, 174, p.121277. doi:https://doi.org/10.1016/j.techfore.2021.121277.