Topic Modelling and Sentiment Analysis of Covid Tweets

by

Thu Yein Oo

P-2706710

Supervisor

Waddah Saeed

A Dissertation

Submitted to Computing, Engineering and Mathematics

De Montfort University

In Partial Fulfillment of the Requirements

for the Degree MSc in Data Analytics

May 2024

# Acknowledgement

I would like to thank my parents and friends for supporting me on this journey. To my supervisor Waddah as well for his support and suggestions.

# Abstract

The unprecedented surge in online discourse surrounding the COVID-19 pandemic has generated a vast amount of textual data in the form of tweets. Latent Dirichlet Allocation and BERTopic are used for topic modelling to find some underlying themes and trends in the tweets. To predict positive, negative or neutral sentiments several classification algorithms are used for sentiment analysis. Furthermore, hyperparameter tuning is employed to achieve optimal performance of Multinomial Naive Bayes classifier. This project provides an opportunity to evaluate different approaches of topic modeling as well as sentiment analyzing algorithms within the context of COVID-19 discourse on Twitter which is often complex and ambiguous. Advantages as well as disadvantages of each method by means of empirical validation and comparative analysis is assessed to offer some insights for scholars engaged in studying public health communication or anyone interested in computational linguistics today.

Contents

[Acknowledgement 1](#_Toc166211901)

[Abstract 2](#_Toc166211902)

[1 Introduction 7](#_Toc166211903)

[1.1 Problem Statement 7](#_Toc166211904)

[1.2 Research Aim 8](#_Toc166211905)

[1.3 Research Questions 8](#_Toc166211906)

[1.2 Limitations 8](#_Toc166211907)

[1.3 Data Confidentiality 8](#_Toc166211908)

[2 Literature Review 9](#_Toc166211909)

[2.1 Natural Language Processing 9](#_Toc166211910)

[2.2 Topic Modelling 9](#_Toc166211911)

[2.2.1 Latent Dirichlet Allocation 10](#_Toc166211912)

[2.2.2 BERTopic 13](#_Toc166211913)

[2.2.3 Measuring Performance of Topic Models 16](#_Toc166211914)

[2.3 Sentiment Analysis 17](#_Toc166211915)

[2.3.1 TF=IDF 17](#_Toc166211916)

[2.3.2 Multinomial Naïve Bayes 18](#_Toc166211917)

[2.3.3 Multinomial Logistic Regression 18](#_Toc166211918)

[2.3.4 Decision Tree 18](#_Toc166211919)

[2.3.5 Random Forest 19](#_Toc166211920)

[2.3.6 Gradient Boosting 20](#_Toc166211921)

[2.3.7 Support Vector Machine 21](#_Toc166211922)

[2.3.8 Measuring Performance of Sentiment Analysis Classification 21](#_Toc166211923)

[3 Methodology 22](#_Toc166211924)

[3.1 Topic Modelling 22](#_Toc166211925)

[3.2 Sentiment Analysis 23](#_Toc166211926)

[3.3 Programming Language 24](#_Toc166211928)

[3.4 Data Description 24](#_Toc166211929)

[3.5 Model Building 26](#_Toc166211930)

[3.5.1 LDA 26](#_Toc166211931)

[3.5.2 BERTopic 26](#_Toc166211932)

[3.5.3 Sentiment Analysis Models 27](#_Toc166211933)

[3.6 Visualization and Results Generation 28](#_Toc166211934)

[3.6.1 LDA 28](#_Toc166211935)

[3.6.2 Bertopic 29](#_Toc166211936)

[3.6.3 Sentiment Analysis Models 29](#_Toc166211937)

[4 Results and Discussion 29](#_Toc166211938)

[4.1 Topic Models 30](#_Toc166211939)

[4.1.1 LDA 30](#_Toc166211940)

[4.1.2 BERTopic 32](#_Toc166211941)

[4.1.3 Comparing LDA and BERTopic 34](#_Toc166211942)

[4.2 Sentiment Classification 35](#_Toc166211943)

[4.2.1 Advantages and Limitations of Classification Algorithms in Sentiment Analysis 37](#_Toc166211944)

[5 Conclusion 39](#_Toc166211945)

[5.1 Research Questions 39](#_Toc166211946)

[5.2 Recommendations for Further Work 39](#_Toc166211947)

[6 References 39](#_Toc166211948)

[7 Appendix 43](#_Toc166211949)

[Figure 1 Graphical Representation of LDA. The outer square is the number of documents, and the inner square is the collection of words and topics within that document. From (Axelborn and Berggren, 2023) 11](#_Toc166212026)

[Figure 2 High Level of BERTopic Model. From (Grootendorst, 2022) 12](#_Toc166212027)

[Figure 3 Granular Illustration of BERTopic Model, from (Grootendorst, 2022) 13](#_Toc166212028)

[Figure 4 Decision Tree, from (Charbuty and Abdulazeez, 2021) 18](#_Toc166212029)

[Figure 5 Overview of Topic Modelling Methodology 21](#_Toc166212030)

[Figure 6 Overview of Sentiment Analysis Methodology 22](#_Toc166212031)

[Figure 7 Top 30 Most Frequent Words in the Tweets. 24](#_Toc166212032)

[Figure 8 Word Count Distribution for all the Tweets. 24](#_Toc166212033)

[Figure 9 Trend of the Sentiment of Tweets 25](#_Toc166212034)

[Figure 10 Intertopic Distribution of LDA Model 29](#_Toc166212035)

[Figure 11 Word Cloud for the Word Distributions in Topics 31](#_Toc166212036)

[Figure 13 Intertopic Distance of BERTopic Model 32](#_Toc166212037)

[Figure 14 Word Distribution in Topics of BERTopic Model 33](#_Toc166212038)

[Figure 15 Performance Comparison of Sentiment Analysis Classification Algorithms 35](#_Toc166212039)

[Table 1 Descriptive Statistics for the cleaned tweets dataset 23](#_Toc166212040)

[Table 2 Topic Distribution of LDA Model in the whole Tweet Data Set 30](#_Toc166212041)

[Table 3 Results of LDA Model 31](#_Toc166212042)

[Table 4 Topic Distribution of BERTopic Model in the whole Tweet Data Set 32](#_Toc166212043)

[Table 5 Comparison of LDA and BERTopic on multiple features 33](#_Toc166212044)

[Table 6 Result Comparison of Sentiment Analysis Classification Algorithms 35](#_Toc166212045)

# 1 Introduction

As the COVID-19 pandemic unfolds globally, social media platforms are responding to an influx of coronavirus-related content. From its origin and initial coronavirus outbreak in Wuhan, China, COVID-19 has crossed all continents causing a global pandemic. On March 11, 2020, the World Health Organization officially declared this public health crisis COVID-19 pandemic while two days later United States President Donald Trump also declared national pandemic emergency status because of COVID-19 outbreak in country. Once off 30th June 2020 an estimation of virus infection cases surpassed a mark of 10 million people with 503,000 dead worldwide (World Health Organization, 2020).

To curb its spread, different countries have used lockdowns, quarantines and travel bans to restrict people’s movements. COVID-19 has brought life to a standstill, with schools closed, jobs lost on a wide scale and many stuck indoors. Most of the communications shifted towards digital interaction and twitter has emerged as a pivotal platform for real-time communication where people can express their opinions, feelings and information of the pandemic.

Social media has developed as a critical way of distributing health information, where almost all persons in different nations use at least one form of social media (World Health Organization, 2020). The importance of social media as an important source for health-related information is underlined by surveys carried out by pew research in many nations (Silver et al., 2019). This is why it is not shocking to find out that in relation to the COVID-19 people choose to exchange sentiments and information through social media platforms.

This is where topic modelling comes into play. Topic modelling, a branch of natural language processing (NLP), is an unsupervised machine learning technique that can scan documents and automatically group similar words and expressions (Pascual, 2019). Traditional books, scientific articles and social media posts all have varying types of discernible patterns of words (Churchill and Singh, 2022).

Sentiment analysis is also a popular method in processing text. Twitter is a unique source of social text data where researchers can retrieve publicized information and process what people are feeling. There is a huge interest in areas such as commerce, health and disaster management (Mohammad, Kiritchenko and Zhu, 2013).

The application of these methods aids in finding new trends, changes in feelings, and what people are talking about when it comes to COVID-19. Analytical results help us to understand more about society in relation to the current pandemic’s conversations and actions on interventions, strategies for communication between different communities or countries among others who may have an interest in it.

## Problem Statement

Millions of tweets are written by users each day and among them, covid related topics hold a significant portion during the pandemic. There are a lot of papers that tackle the topic modelling and sentiment analysis classification. They use big data sets and high computationally consuming algorithms to deal with these cases (Kit and Mokji, 2022) (Konstantinas Korovkinas, Paulius Danėnas and Gintautas Garšva, 2018). For organizations or people who want to analyze these huge data sets, there will be financial constraints.

This master project will determine which algorithms are feasible in terms of high accuracy and computational efficiency.

## 1.2 Research Aim

This project will explore unsupervised topic modelling, Latent Dirichlet Allocation (LDA) and Bidirectional Encoder Representations from Transformers (BERTopic). By using LDA and BERTopic, the project will analyze the tweets and see if valuable insights can be drawn. With the supervised sentiment analysis part, popular algorithms such as naïve bayes, logistic regression, decision tree, random forest, light gradient boosting, extreme gradient boosting (XGBoost), cat boosting, and support vector machine will be used to perform classification of tweets with negative, positive and neutral states. These methods will be thoroughly analyzed to find which model has the best computational efficiency.

## 1.3 Research Questions

* Which unsupervised model, LDA or Bertopic, provides better accuracy in dealing with tweets (short text) data and gives greater interpretability?
* Which supervised model, Naive Bayes, Regression, Tree Models or Support Vector Machine, has better performance in classifying the positive, negative and neutral tweets?
* Among those, what model can be further optimized, and hyper parameter tuned to improve performance in terms of better efficiency (less computing power) and accuracy?

## 1.2 Limitations

Since the project has a limited time constraint, a cleaned data set from Kaggle is used (<https://www.kaggle.com/datasets/arunavakrchakraborty/covid19-twitter-dataset>). The data is pre-processed by first turning all the text into lower case. All extra white spaces, numbers, special characters, ASCII characters, URLs, punctuations & stop words are removed as well. All covid related words are converted into ‘covid19’ for consistency and the numbers are taken out. As the last step, the words are lemmatized to the base form. Since the project will focus on the natural language processing algorithms, a cleaned data set is used.

## 1.3 Data Confidentiality

All the tweets used in the data set are scrapped from public tweets and the usernames are removed so that there is anonymity. Therefore, there is no ethical issue from using this data.

# 2 Literature Review

## 2.1 Natural Language Processing

Natural Language Processing (NLP) is a subfield of artificial intelligence, and it is also an intersection of linguistic and computer science where it tries to connect human language and machine understandability (Nadkarni, Ohno-Machado and Chapman, 2011). NLP, at its core, tries to understand, interpret and generate text data or human language. Applications of NLP include machine translation, email spam detection, information extraction, summarization, question answering, etc. (Khurana et al., 2022).

The major problem that makes NLP a difficult problem to tackle is that human languages are very hard to work with because they are always changing as well as having many dialects which exist concurrently within them all over the world. Some words and phrases that seem obvious to humans may pose a challenge for a machine to understand. The same set of words may have different meanings depending on the context, situation, and culture. In certain cases, homonyms where words with similar sounds but different spelling can cause problems in speech to text applications. Sarcasm could be differently interpreted as well. Above mentioned cases could be improved by providing a huge set of training data, but it could only be useful for a specific domain (Khurana et al., 2022).

In order to solve these problems, NLP uses numerous methods and ways that originate from fields, for instance, deep learning, machine learning, computational linguistics among others. When it comes to executing tasks, NLP practitioners have at their disposal a selection of tools and algorithms from rule-based systems up to neural network designs; such tools can cater tasks ranging from simple language comprehension ones to those that are highly sophisticated pertaining cognitive functions.

## 2.2 Topic Modelling

In machine learning and natural language processing, topic models are generative models with probabilistic frameworks. These techniques usually serve the purpose of organizing, understanding, searching, summarization of huge volumes of textual information automatically. The term topics refers to latent variables that must be estimated through the relationship that exists between words in a vocabulary and their appearances within documents. These latent topics are mixed together in documents, while models identify hidden themes and tag documents for each topic. Each word is associated with a topic; thus, the word topic is considered as the origin of that word. Consequently, we get another way of looking at data with topics as we finally have a topic distribution which spans the entire document (Tong and Zhang, 2016).

The earliest form of topic modelling can be traced back to 1980s where latent semantic analysis (LSA) was a prototypical research project. LSA can extract relationships from words in a corpus of text. LSA captures simultaneous occurrences of words and identifies topics by using matrix factorization techniques such as singular value decomposition (Papadimitriou et al., 1998).

Thomas Hofmann introduced the idea of probabilistic latent semantic analysis (PLSA) in the late 1990s. PLSA is an automated document indexing model trained with an expectation-maximization algorithm which estimates the probability of topics for each document and words for each topic. The multiple layered factorization matrix of the distributions of words over topics and topics over documents plays a crucial role in PLSA. PLSA paved the way for unsupervised topic modelling (Hofmann, 2017).

### 2.2.1 Latent Dirichlet Allocation

LDA is one of the powerful probabilistic models and was invented by Blei, Ng, and Jordan in year 2003. LDA has since picked up traction in different fields as it efficiently conducts topic modeling and document clustering tasks.

LDA has been thoroughly studied and improved upon since its inception, leading to many advancements and expansions on the original model. One significant research area involves enhancing the scalability and efficiency of LDA inference algorithms for handling increasingly large and varied text corpora. Proposed solutions to this problem include variational inference, collapsed Gibbs sampling as well as optimization for parallel computing that allowed LDA to operate on large data sets. One such example is the Auto-encoded Variational Inference for Topic Mode (AVITM) where the researchers modified a single line of code in LDA to dramatically improve the performance (Srivastava and Sutton, 2017).

Additionally, scholars have come up with numerous extensions or modifications of LDA that are intended to meet specific needs within different application areas. A striking example is supervised LDA where document labels or metadata are used during modeling process so that discovered topics are easier to interpret or distinguish from each other in comparison with unsupervised mode. In a similar manner, dynamic LDA accounts for time series’ temporal dynamics thereby detecting how themes evolve or change with time. In one study, researchers preprocessed tweets by pooling them via hashtags without modifying the base LDA (Mehrotra et al., 2013). This also has improvements in performance.

Apart from methodical improvements, LDA employment is extensive. It cuts across various disciplines such as natural language processing, information retrieval, social media analysis, and computational social science. In this regard, analysts have used LDA to scrutinize vast amounts of text data obtained from sources like news reports, scientific papers, social network messages, and online forums.

By and large, the existing work on Latent Dirichlet Allocation underscores both its foundational nature within the domain of computational linguistics as well as its increasing importance for understanding and analyzing enormous text corpora. As for future, it will continue to be an important instrument of uncovering concealed designs or themes within written materials because researchers are always refining it with new modifications when they study different types of texts and examine how these behave when they are grouped together under different classifications of documents.

LDA uses words, documents and corpus when explaining abstract notions. They are formally defined in the following terms as in the original paper (Blei et al., 2003).

* A word is the basic unit of discrete data, defined to be an item from a vocabulary indexed by {1, ..., V}. We represent words using unit-basis vectors that have a single component equal to one and all other components equal to zero. Thus, using superscripts to denote components, the vth word in the vocabulary is represented by a V-vector w such that wv = 1 and wu = 0 for u ≠ v.
* A document is a sequence of N words denoted by w = (w1, w2, ..., wN), where wN is the nth word in the sequence.
* A corpus is a collection of M documents denoted by D = {w1, w2, ..., wM}.

In Latent Dirichlet Allocation (LDA), the main idea is that we can represent documents as mixtures of hidden topics, each of which comprises as a set of words. Usually, words in a topic manifesting the highest likelihood are those that are indicative of the content of that topic. LDA asserts that there is a distribution of topics involved in every document. Moreover, the overall distribution of topics over all the documents in corpus can be characterized by a Dirichlet prior which is shared by all the documents. Also, one can consider each hidden topic as a distribution over words, where the distributions of words within individual topics adhere to the same Dirichlet prior. In addition to constructing an accurate probabilistic model of a given corpus, LDA is geared towards assigning high probabilities to documents whose topic mixtures are similar (Blei et al., 2003).

The graphical model of LDA is as follows:

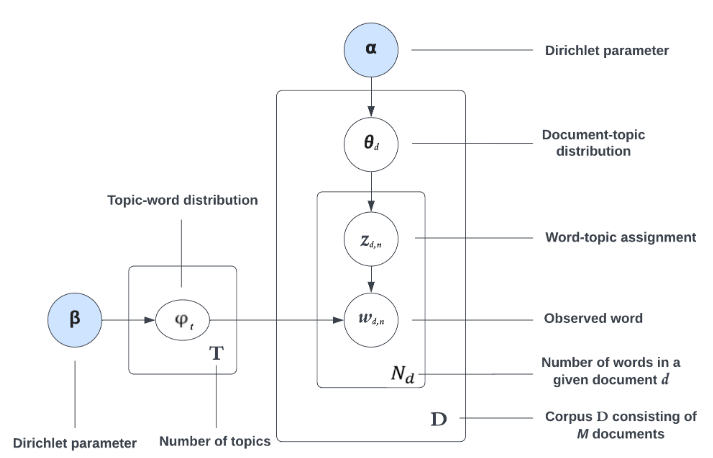


Figure 1 Graphical Representation of LDA. The outer square is the number of documents, and the inner square is the collection of words and topics within that document. From (Axelborn and Berggren, 2023)

The generative process of LDA in each document is as follows (Blei et al., 2003):

1. Φt which is a multinomial distribution for a topic is chosen from a Dirichlet distribution with parameter .
2. Θd which is a multinomial distribution for a document is chosen from a Dirichlet distribution with parameter α.
3. For a word wn in document d:
   1. Choose a topic zn from Θd.
   2. Choose a word wn from Φzn, on the topic zn.

The latent variables are Φ and Θ while the hyper parameters are α and . The probability of the observed data D is calculated as follows.

A black and white math symbol

Description automatically generated

α parameters come from the topic Dirichlet prior and parameters come from the word-topic Dirichlet distribution. M is the size of the documents; N is the size of the words. Θd is a document-level variable, zdn and wdn are word-level variables.

LDA has a hierarchical structure that includes three levels related to documents, topics, and words. These hierarchy levels have their own respective prior distributions, depending on the parameters that are on the previous level. This kind of arrangement enables this model to infer parameters using Bayesian statistics by relying on Bayes’ theorem for their estimation process (Kunsabo and Dobša, 2022). The Bayes’ theorem is further explained in 2.3.1 Multinomial Naïve Bayes.

### 2.2.2 BERTopic

Maarten Grootendorst introduced BERTopic (Grootendorst, 2022) which is an innovative technique used for topic modeling this method has gained attention due to its ability to utilize pre-trained BERT (Bidirectional Encoder Representations from Transformers) models for effective and efficient clustering based on topics. BERTtopic has since been explored and implemented in various areas of study, with significant outcomes that reveal obscured themes in text data.

BERTopic, in high level, comprises of five or six steps depending on fine tuning as shown Figure 2.

A diagram of a diagram

Description automatically generated

Figure 2 High Level of BERTopic Model. From (Grootendorst, 2022)

Research has shown that BERTopic outperforms models such as LDA in terms of topic coherence and interpretability; thus, making it applicable in different fields because it produces clear high-quality topics easily out of textural information. BERTopic, in its granularity, uses standard BERT for embedding, Uniform Manifold Approximation and Projection (UMAP) for reducing dimensions, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) for clustering, Count Vectorizer for tokenization, class based term frequency- document term frequency (c-TF-IDF) for weighting scheme and keyBERT for fine tuning as shown in Figure 3.

A chart of different colored blocks

Description automatically generated with medium confidence

Figure 3 Granular Illustration of BERTopic Model, from (Grootendorst, 2022)

#### 2.2.2.1 Embedding

In the BERTopic, the embeddings are words or phrases representations generated by the pre-trained BERT model that is contextualized. These embeddings communicate the meanings from the input text which are largely comprised of their semantics, and they identify where documents share similar meaning through analyzing or clustering them into topics. It could not be possible for BERTopic to achieve such results without using BERT embeddings given that other models in traditional topical patterning only remain directed by words and how frequently they occur together or linguistic features that are superficial.

For embedding, BERTopic uses sentence transformers as a default although there are many methods that can be chosen. The model "all-MiniLM-L6-v2" is a pre-trained sentence transformer widely used for many NLP applications.

#### 2.2.2.2 Dimensionality Reduction

After setting numerical representations for our texts, the next step is to work on reducing the dimensions of these representations. High-dimensionality data is problematic for clustering models because there is an issue associated with dimensionality which can result in long training times. Despite the efficiency of Principal Component Analysis (PCA) in dimension reduction, BERTopic favors using UMAP because it preserves local and global characteristics while reducing the dimensions of the dataset. What makes UMAP special is that it retains both local and global structures of the dataset while conducting dimensionality reduction. The maintenance of this form is important since it carries information that is necessary for determining clusters of texts with similar meanings (McInnes, Healy and Melville, 2020).

#### 2.2.2.3 Clustering

After reducing the size of the embeddings, clustering the data comes next. For this, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), a density-based clustering method, is used. This method is good at finding different shaped clusters and it has an advantage in that it can pick out any anomalies that might exist within them. In so doing, documents are refrained from being lumped into artificial categories. Therefore, this technique decreases the interference of the noise and gives a superior representation of topics.

DBSCAN, a clustering algorithm, groups data points with reference to their spatial density. It can recognize regions of high-density points or clusters, separated by regions of low density, which are normally considered as noise or outliers. DBSCAN eliminates the need to pre-define the number of clusters and can find arbitrarily shaped or sized clusters. It comprises three different types of data points: core points surrounded by other points; border points that lie on the edge of a cluster; noise points do not belong to any cluster. The algorithm builds clusters by connecting core points and assigning border points to belonging clusters through iterating over each point’s neighborhood (Schubert et al., 2017). The method is robust to noise and outliers but the proper parameter values, such as the maximum radius of neighborhood (epsilon) and the minimum number of points that need to lie within that range for a cluster to be created (minPts), must be chosen carefully.

#### 2.2.2.4 Tokenization

The BERTopic tokenization refers to breaking down text into single tokens or words. The text is separated into meaningful entities that include words, punctuation marks as well as special characters before they are transformed into numerical values that can be processed by the BERT model. It is an important part of preparing texts for analysis since it enables BERTopic to effectively extract semantic information from the input text. BERTopic uses CountVectorizer from Scikit-learn.

#### 2.2.2.5 Weighting Scheme

BERTopic assigns significance for individual words or tokens in text data through its weighting schemes. One such common weighting scheme is TF-IDF which gauges how critical a word is in a document as compared to other documents in the whole corpus based on the number of times it appears (Ramos, 2003). BERTopic uses a modified version of TF-IDF called class-based TF-IDF.

C-TF-IDF treats all documents in a cluster as a single document. If a word has high significance in a cluster, it will become one of the words representing the topic. C-TF-IDF is calculated as follows:

A math equations and formulas

Description automatically generated with medium confidence

Instead of numerous documents, all the clusters are combined into a single document where we can now calculate the frequency of words across cluster c. A class-oriented matrix, T.F (term frequency) is generated. Then the natural logarithm of one plus the average number of words in each cluster (A) divided by the occurrences of the word x in all the cluster is calculated. One is added into the logarithm so that any term occurring without the regularity gets a weight. This is called the class-specific frequency (IDF) formation, and it is used similarly to TF-IDF which multiplies together TF by IDF giving out the prominence weight. Fundamentally, instead of using the traditional way of ranking words in text bodies, an altered TF-IDF algorithm designed for more efficient word structure is utilized.

#### 2.2.2.6 Fine Tuning

New advancements in NLP has given us the opportunity to further fine tune the BERTopic model. Tools such as GPT, T5, KeyBERT, Spacy, etc. can be implemented. KeyBERT, created by Maarten Grootendorst, is a straightforward yet potent method enabling users to extract the most pertinent keywords and key phrases from a document by analyzing its content. It will be used in fine tuning the BERTopic.

### 2.2.3 Measuring Performance of Topic Models

#### 2.2.3.1 Coherence

Topic models are commonly evaluated with c\_v coherence. It calculates the semantic similarity between words in a topic model. It computes pairwise cosine similarity of all word vectors among all the pairs of words or terms which are found within those topics. This score shows how closely related the words are to each other in the topic. The higher the score, the better the words are semantically similar and related to each other in a topic (Röder, Both and Hinneburg, 2015).

#### 2.2.3.2 Perplexity

Perplexity is a way of indicating how well the model makes guesses concerning the frequent occurrence of words in a test set of text data. Lower perplexity means the model makes better guesses about the frequent occurrence of word in the test paragraph and thus has a good grasp of the data’s underlying structure. In mathematical terms, perplexity can be measured by taking the exponential function of the mean negative log probability for a test data divided by total number of words or symbols on it (Blei et al., 2003).

A black and white text

Description automatically generated

## 2.3 Sentiment Analysis

Sentiment analysis is the process of analyzing text to determine the sentiment expressed within it, also known as opinion mining. Involving the classification of text as positive, negative, or neutral depending on opinions, emotions or attitudes conveyed by the language used, sentiment analysis is popularly used in different applications like social media monitoring, customer feedback analysis and market research among others to discern public opinion, check out how people feel about different brands thus enabling data based decisions to be made (Medhat, Hassan and Korashy, 2014).

In this project, multiple algorithms will be explored and compared against each other on how well they can predict the results. The data will be supervised with a negative, positive and neutral feature.

### 2.3.1 TF=IDF

A widely used technique in natural language processing is TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF is used for ranking and weighting terms in a document corpus. TF-IDF works by comparing the frequency of particular words in one document with their commonness in the whole collection. This essentially tells how essential a term is in a given document. Words that appear in a single document or a limited set of documents typically exhibit higher TF-IDF values compared to commonly occurring words like articles and prepositions (Ramos, 2003).

The equation of TF-IDF have minor differences but the overall approach is as follows:



Where:

* D is a document collection
* d is an individual document within D
* w is a word
* *fw, d* is the frequency of w appearing in d
* *fw, D* is the frequency of w appearing in D

### 2.3.2 Multinomial Naïve Bayes

Multinomial Naive Bayes is a version of the Naive Bayes algorithm that is frequently deployed in sentiment analysis projects, particularly for text data that is represented in the form of word counts or term frequencies. Under this method, the algorithm assumes that the features (words) are derived from a multinomial distribution after which it computes the probability of observing every word across sentiment classes.

MNB is based on Bayes’s rule:



Where:

* p(y|x) is the probability event y occurring given the event of x.
* p(x|y) is the probability event x occurring given the event of y.
* p(x) and p(y) are observed probabilities of x and y independent of any conditions.

### 2.3.3 Multinomial Logistic Regression

Logistic regression predicts the probability that any text has a place in specific sentiment category by use of a logistic function. The logistic function yields a probability score ranging from 0 to 1 that shows how much a particular statement is likely to fall under a specific sentiment category (IBM, 2022). In this project, multinomial logistic regression is used since it will be predicting a non-binary class.

### 2.3.4 Decision Tree

A decision tree can be defined as “a tree-based technique in which any path beginning from the root is described by a data separating sequence until a Boolean outcome at the leaf node is achieved” (Charbuty and Abdulazeez, 2021). For each inner node in the tree, select a feature and a threshold value the best splits the data into two or more subsets as shown in Figure 4. The selection is based on a criterion such as Gini impurity for classification tasks or mean squared error for regression tasks measure the homogeneity or purity of the subsets.

A diagram of a tree

Description automatically generated

Figure 4 Decision Tree, from (Charbuty and Abdulazeez, 2021)

The decision tree splits the data repeatedly until stopping points occur, like when it reaches the highest limit or when there are too few samples left in a node or purity does not change significantly. If this process goes on forever, it will eventually produce a model with zero errors on the training set which is a classic case of overfitting.

### 2.3.5 Random Forest

Random forest is a versatile and powerful algorithm for ensemble learning that is applied to both classification and regression problems. It operates by training numerous decision trees and afterwards, it makes a prediction based on a majority vote for classifications and an average value in case of regression (Donges, 2021).

Random forests usually operate as follows:

* **Sampling:** Random forest starts by creating multiple samples that are later used to train a separate decision tree.
* **Feature Selection:** Selecting a subset of features randomly at each node of the decision tree is substitute for analyzing each feature during the random forest decision-making process. Doing so helps in decreasing inter-tree correlation and giving the collection various forms.
* **Decision Tree Training:** The features that were selected were used to train a decision tree. Based on a criterion like Gini impurity for classifications, or mean squared errors in regressions, the optimal split is chosen at each internal node of the tree.
* **Voting (Classification) or Averaging (Regression):** After training all the decision trees, the predictions for new data points are made by aggregating the individual trees’ predictions. For classification tasks, the prediction takes the mode of the predictions. On the other hand, for regression tasks, the average is chosen.

One of the reasons why it is advantageous to use random forest is that they are very good at avoiding overfitting, they can scale well on large data sets and can handle high-dimensional feature spaces. Another reason is that random forests are not as affected by noisy data as any single decision tree would be (Donges, 2021).

### 2.3.6 Gradient Boosting

Gradient boosting is a very strong machine learning technique used for supervised learning tasks like classification, regression and ranking, and it constructs a sequence of weak learners that are usually decision trees correcting errors of the previous ones by calculating residues.

#### 2.3.6.1 Light Gradient Boosting

Light gradient boosting is a combination of two algorithms: Gradient-based One-side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS therefore reduces the number of utilized instances while keeping the distribution of gradients in whole. As a result, the speed of training is improved, and memory usage is also reduced. During the construction of the tree, EFB bundles several unique features together, which in turn reduces the number splits hence better training speed and model performance (Ke et al., 2017).

#### 2.3.6.2 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a boosting algorithm and consists of several key parts:

**Exact Greedy Algorithm:** The computational intensity of generating all potential splits for continuous features while handling this efficiently requires initially sorting the data based on feature values by the algorithm. Next, it goes through the sorted data to accumulate gradient statistics required to evaluate the structure score (Chen and Guestrin, 2016).

**Approximate Algorithm and Weighted Quantile Sketch:** Firstly, the approximate algorithm proposes potential dividing points through taking percentiles of the feature distribution. Secondly, it partitions continuous features into buckets created by these possible divisions whereby in each bucket it combines the statistics and finally selects the best solution among the proposals based on combined statistics.

An important part of approximate algorithm is using weighted quantile sketching to manage weighted data It involves constructing a statistical graph of the data distribution determined by quantile values that accounts for the weights associated with each datapoint. This way, the data is summarized concisely (Chen and Guestrin, 2016).

**Sparsity-aware Split Finding:** It is meant for handling sparse data effectively while looking for the best splits in decision trees. It considers the data's sparsity to optimize its splitting process making it an efficient way of handling sparse features for enhanced performance of XGBoost algorithm on sparse datasets.

#### 2.3.6.3 Categorical Boosting

CatBoost is able to handle categorical features directly. This can be one-hot encoded, or label encoded without the need for preprocessing. It employs a technique known as ordered boosting that transforms categorical variables to numerical ones during training, maintaining their hierarchical position (Prokhorenkova et al., 2018).

### 2.3.7 Support Vector Machine

The Support Vector Machine (SVM) algorithm is a powerful supervised machine learning method that is used for both classification and regression tasks. Particularly effective in high-dimensional spaces, it is widely used in various applications like image classification, text classification, and bioinformatics.

SVM has two major components:

**Hyperplane Classifier:** Support Vector Machine (SVM) works on finding the hyperplane which separates the data into the best classes. A hyperplane is a boundary for making decisions amongst any number of regions in the feature space depending on attributes associated with them. The purpose here is to increase the margin between the hyperplane and the nearest data points from each class, a value which is known as support vectors (Hearst et al., 1998),

**Kernel Trick:** Instead of locating a linear hyperplane in the initial feature space, SVM repositions the information into a higher dimension through a method commonly referred to as the kernel trick, therefore; making linear separability of classes possible. This is how SVM can be able to establish a decision boundary which is nonlinear within the extended dimension (Hearst et al., 1998).

### 2.3.8 Measuring Performance of Sentiment Analysis Classification

For tasks in which more than two classes are present, precision, recall and F1 score can also be calculated. Here’s an extension of how precision, recall as well as f1 score get extended into multi-class classification (Singh, 2019)

**Precision:** It calculates how correctly the model predicts the positives. Precision is true positives divided by the sum of true positives and false positives.

**Recall:** It calculates the partition of all actual positives. Recall is true positives divided by the sum of true positives and false negatives.

**F1 Score:** The F1 score is the harmonic mean of the precision and recall. It gives an equivalent and balanced measure of the model's performance through considering both precision and recall. The formula for F1 score is:

****

To perform multi-classification, precision, recall and F1-score can be computed per each class separately or averaged across all classes to have a single measure summarizing overall model performance across all classes. Most popular averaging techniques include micro average, macro average and weighted average.

Micro-average computes metrics globally by counting the total number of true positives, false positives, and false negatives across all classes. Macro-average computes metrics independently for each class and then averages them. Weighted average computes metrics for each class and then averages them, weighted by the number of true instances for each class.

# 3 Methodology

In this section, the overview of the workflow of the topic modelling and sentiment analysis is shown.

## 3.1 Topic Modelling

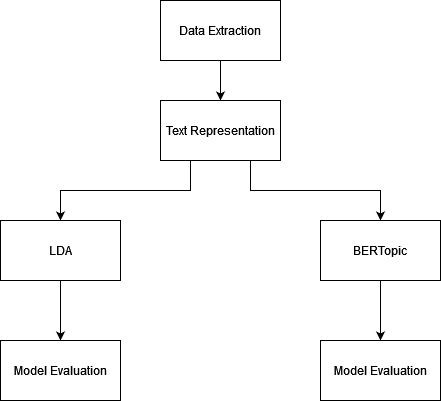


Figure 5 Overview of Topic Modelling Methodology

The dataset employed in this investigation has been directly taken from Kaggle dataset for ensuring its credibility and relevance to the research goals. Before analysis can be conducted on these, the tweets must first go through a meticulous cleaning that removes all unwanted data such as URLs, special character sets as well as stop words among others. Once these have been completed, a bag-of-words representation is used where each row represents a document (tweet) and each column represents a unique word present in the corpus.

After that, the preprocessed tweets are processed through two popular algorithms for topic modeling i.e., the Latent Dirichlet Allocation (LDA) and BERTopic algorithms that extract hidden topics and patterns existing within the text corpus. This way, the inclusion of such techniques ensures that we exhaustively investigate different themes and emotions that characterize discussions around the COVID-19 pandemic on various platforms especially social networking sites. It is possible through the incorporation of various topic modeling strategies to evaluate comparatively how well each one performs concerning retrieving valuable information from the entire tweet database.

After using topic modeling algorithms, these subjects are subjected to close examination to establish the semantic coherence and relevance of research questions. Through looking at the results obtained from LDA and BERTopic, crucial information is obtained on existing themes, moods, and issues discussing the COVID-19 outbreak reflected in the tweet dataset.

## 3.2 Sentiment Analysis

## 

Figure 6 Overview of Sentiment Analysis Methodology

Once the cleaning is done, the next step is vectorizing tweets which is where the Term Frequency-Inverse Document Frequency (TF-IDF) comes into play, a method commonly used in natural language processing (NLP) to transform text data into numeric counterparts. It assigns values to each word depending on how often it occurs in a document and how infrequent it is across a large swath of documents meaning that it tells us much more about individual tweet tweet’s relative importance compared to all other tweets than just its position in the data set.

Following this, the vectorized tweets are processed through various classification algorithms for sentiment analysis distinguishing them as positive, negative or neutral. Multinomial Naïve Bayes, logistic regression, decision tree models and support vector machines (SVMs) are some which have been used as classifiers. This allows for thorough exploration of the entire tweet collection from multiple viewpoints, given that each of these techniques comes with its own advantages and properties.

When using multinomial Naïve Bayes, more work goes into fine-tuning parameters to maximize model performance as well as prediction accuracy. Such tuning includes changes made by moving around values within hyperparameters as in the case with adjusting alpha for smoothing purpose so that both bias and variance are reduced leading to more stable learners capable of correctly predicting on new datasets.

After the modeling phase, we evaluate and contrast the performance of each classification algorithm using various evaluation metrics including accuracy, precision, recall, F1-score, and confusion matrices. This provides us with useful recommendations on how to effectively predict sentiment in tweets and classify them properly.

## 3.3 Programming Language

Throughout this thesis project, the primary programming language has been Python. Due to its efficiency and ease of use, Python has been the best choice for implementing NLP models for many years. Moreover, Python is compatible with many open-source libraries and packages. For computing power, the process runs on the author’s own personal laptop with a sufficient GPU to train models.

## 3.4 Data Description

The data set used in the project is the covid tweets data set ranging from April 2021 to June 2021. The main columns that are used are the “clean\_tweet” and the “sentiment”. Table 1 shows the basic description of the data set. There are more positive tweets compared to negative tweets while the majority are neutral. Figure 9 shows the count of positive, negative and neutral tweets across three months that the tweets are taken from.

|  |  |
| --- | --- |
| **Dataset Summary** | |
| Average word per tweet | 60 |
| Minimum word for a tweet | 2 |
| Maximum word for a tweet | 125 |
| Negative Tweets | 36,401 |
| Neutral Tweets | 65,369 |
| Positive Tweets | 44,756 |
| Total tweets | 146,526 |

Table 1 Descriptive Statistics for the cleaned tweets dataset

Figure 7 shows words that appears most commonly in the tweets. It is evident that the word covid and vaccine would appear the most. Figure 8 illustrates the distribution of words in the tweets and the words per tweets follow a gaussian distribution.

A graph of words and a number of words

Description automatically generated

Figure 7 Top 30 Most Frequent Words in the Tweets.

A graph of a number of blue bars

Description automatically generated

Figure 8 Word Count Distribution for all the Tweets.

Figure 9 Trend of the Sentiment of Tweets

## 3.5 Model Building

### 3.5.1 LDA

The corpus needs to be tokenized using the doc2bow function to turn them into the bag-of-words format, (token\_id, count) dictionary pairs, to feed into the LDA algorithm. In order to map individual token\_ids to their respective words, a dictionary was created that associated each token\_id with the word it stood for. This dictionary was generated using the corpora.dictionary function of Gensim library.

Radim Řehůřek created the Gensim library in 2008. It provides tools for creating an LDA model from a set of documents and estimating topic distribution in unseen texts. The Bag-of-Words representation depicted the text corpus in the LDA model implementation, whereby the LdaModel function was employed through importing models.ldamodel module from Gensim library. There are several parameters available in this function that one can use to customize the LDA model. We set the two Dirichlet hyperparameters alpha and beta to be the default 'auto' setting which controls how sparse the topic-document and word-topic distributions are respectively. Under this setting, an asymmetric prior is learned directly from data by the model. This means that α and β in each document were optimized by the dataset that implies this model would adapt over time to the appropriate value of α and β for a given document as seen through learning it.

### 3.5.2 BERTopic

BERTopic includes several steps in the creation of topics.

1. The first step is the model embedding where the sentensetransformer function is used via pre-trained model: “all-MinLM-L6-v2”.
2. After text representation, the dimension of the data is reduced by UMAP as shown in 2.2.2.2 Dimensionality Reduction via the UMAP function.
3. Then the data is clustered by HBDSCAN from the scikit-learn library via the HBDSCAN method.
4. In each cluster, the bag-of-words representation is generated via CountVectorizor function in scikit-learn library.
5. To differentiate between each cluster, the classTfidftransformer function from the bertopic library is used.
6. Finally, the fine tuning of the generated topic models is done by keyBERTinspired function from the bertopic library.

After choosing a model suite, it had to be trained specifically on the available data for which this was accomplished by applying the fit\_transform technique found under the BERTopic library. Then, the outlier topics are removed, and the number of topics is then reduced for better representation.

### 3.5.3 Sentiment Analysis Models

The sentiment analysis models follow a similar structure. First, the document is vectorized by tfidfVectorizer function from the scikit-learn library. In addition, label encoding is required by some classification algorithms to transform categorical sentiment labels (which are negative, neutral or positive) into numbers like zero, one, or two respectively. These algorithms use a label encoder function to implement this encoding process; this assists them in aligning their input requirements to those which they can accepted as an input for the function.

It is common practice to split the data after preprocessing, into two parts for training and testing using 80:20 ratio for a good balance between training and evaluation dataset. Later on this training set can be used to train your model on sentiment analysis applying .fit method which lets it capture the relationship between the features and the target sentiment variables.

After training, the model is tested on test data to generate sentiment labels of unknown tweets using the .predict method. Such an inference stage reveals unambiguously by how far the model can be regarded as a general or specific text classifier.

Take for example the training sequence of the Multinomial Naïve Bayes classifier. To train this model, the .fit method is used to pass vectorized training data as parameter which will enable algorithm estimate parameters of probability distribution for each outcome class. The next step involves using the obtained information from the training phase while fitting model for prediction purpose.

A screen shot of a computer code

Description automatically generated

#### 3.5.3.1 Hyper Parameter Tuning of Multinomial Naïve Bayes

The parameter tuning of the MNB is done through grid search method with cross validation. Multiple parameters are iterated through the grid and looked for the best combination. The reason why MNB is chosen to tune is that it is a relatively light algorithm and has low training time so many loops can be performed.

A screenshot of a computer code

Description automatically generated

Tfidf\_max\_features is to construct a vocabulary that only includes the top terms ordered by their frequency across the entire corpus.

Tfidf\_max\_df is when constructing the vocabulary, disregard terms that have a document frequency exceeding the specified threshold.

Tfidf\_min\_df is when creating the vocabulary, disregard terms that appear in documents less frequently than the provided threshold. This value is commonly referred to as the "cut-off" in literature.

Nb\_alpha is the additive Laplace soothing. The α parameter is smoothing parameter normally used in cases where the certain terms given in the training set with zero frequencies in a specific class when using Multinomial Naive Bayes. During classification, this is added to the count of each term to prevent zero probabilities. The smoothing strength relies on the alpha value with more strong smoothing having a larger alpha value and weaker smoothing having a smaller alpha value. By adjusting the alpha parameter, you can strike a balance between incorporating prior information especially when α is small or basing your decisions much on data which has been observed when α is high.

## 3.6 Visualization and Results Generation

### 3.6.1 LDA

PyLDAvis, a python module, is used to visualize topics interactively. In the bubble chart generated by this tool, each topic is represented by a bubble. In this case, higher bubble sizes imply that the relevant topic is more prevalent than smaller bubble sizes do on average. The reason is due to how many different tokens have been assigned to each respective topic. Large non touching bubbles spread all over the chart reflect good topic models contrary to those concentrated in one place. Small bubbles would be the characteristic of a model with too many topics, and they would keep on forming clusters by overlapping each other.

To calculate the coherence score and perplexity of LDA model, .get\_coherence() and .log\_perplexity() are used.

### 3.6.2 Bertopic

To visualize Bertopic model, several methods are used namely .visualize\_topics() and .visualize\_barchart().

In LDA, perplexity serves as a measurement for calculating model fitness and establishing the final model’s best number of topics to have. Nonetheless, BERTopic users don’t have to pre-specify the number of topics especially when using HDBSCAN. Thus, it is incorrect to consider perplexity in the context of evaluating the best model for BERTopic. For coherence, a custom function using the Gensim library is written.

### 3.6.3 Sentiment Analysis Models

The confusion matrices are the best tool which can be used for visualizing the performance of sentiment analysis models. In Python, Matplotlib is the commonly used plotting library that enables creation of intuitive and informative visualizations on these confusion matrices. In these matrices, true positives/negatives and false positives/negatives for each sentiment class are used thus allowing us to fully comprehend how well our model performs in classifying sentiments. True positives/negatives are those items correctly classified as belonging to the class while false positives/negatives include those items wrongly assigned to it.

Sentiment analysis models are performance evaluated using precision, recall, and F1 scores as well as confusion matrices. These metrics are computed using classification report method from the scikit-learn library.

Through the utilization of these metrics of evaluation and ways of visualizing data, the stakeholders can be able to comprehend more about how well sentiment analysis models are behaving and thus they will be able to determine which sections need to be revised as well as choosing wisely while considering deploying or making use of the models in relation to COVID-19 tweets that are meant for sentiment classification. This full assessment approach adds to the strength and trustworthiness of sentiment analysis models which then leads to their appropriateness in dealing with complex sentiment analysis challenges under social media conversation about the pandemic.

# 4 Results and Discussion

This section is divided into two sections where 4.1 Topic Models that shows the results of LDA and BERTopic and 4.2 Sentiment Classification that shows the results of classification of multiple classification models.

## 4.1 Topic Models

### 4.1.1 LDA

The LDA model is set to produce ten topics. Each bubble’s size matches the unique tokens’ proportion related to the topic. Evaluating the human interpretability about the topics of this model show that there are three distinct regions of topics. From Figure 10, topics 1 and 2 are overlapping quite a bit so they share some words, topic seven to ten are meshed up as well and share similar words. From Table 2, we can see those topics one, two, and three contain the majority of words.

A diagram of a diagram

Description automatically generated

Figure 10 Intertopic Distribution of LDA Model

|  |  |
| --- | --- |
| **Topic** | **Distribution(%)** |
| Topic 1 | 25.2 |
| Topic 2 | 24.4 |
| Topic 3 | 16 |
| Topic 4 | 8.5 |
| Topic 5 | 6 |
| Topic 6 | 5.3 |
| Topic 7 | 5.1 |
| Topic 8 | 4.7 |
| Topic 9 | 4.5 |
| Topic 10 | 2.4 |

Table 2 Topic Distribution of LDA Model in the whole Tweet Data Set

Figure 11 shows the word cloud of the most frequent words in each topic. Some coherence can be seen in the topics.

* Topic 1 is about general covid related about new reports, cases and death.
* Topic 2 is about a new delta variant and how the government is dealing with it.
* Topic 3 is about people need to go to work and get help.
* Topic 4 is about vaccination.
* Topic 5 is about global issues.
* Topic 6 is about clinics and women issues.
* Topic 7 is about hospitalization and losing a loved one.
* Topic 8 is about spreading knowledge.
* Topic 9 is about leadership in America.
* Topic 10 is about getting an appointment in Walgreen locations. This can be about getting oxygen supplies from there.

The evaluation scores are presented in Table 3.



Figure 11 Word Cloud for the Word Distributions in Topics

|  |  |  |
| --- | --- | --- |
| **Number of Topics** | **Perplexity** | **Coherence** |
| 10 | -7.68 | 0.34 |

Table 3 Results of LDA Model

### 4.1.2 BERTopic

In BERTopic model, the inter distance between the bubbles is evident as seen in Figure 13. There is a clear distinction between the topics. Looking at Figure 14 and Table 4, the topics diverge from LDA model in the way that the covid related topics are group into a single large bubble and the rest are different.

The **coherence score** of BERTopic model is 0.57.

A grid of circles and dots

Description automatically generated

Figure 13 Intertopic Distance of BERTopic Model

|  |  |
| --- | --- |
| **Topic** | **Count** |
| 0 | 131,895 |
| 1 | 3,314 |
| 2 | 2,663 |
| 3 | 2,542 |
| 4 | 1,939 |
| 5 | 1,903 |
| 6 | 1,138 |
| 7 | 890 |
| 8 | 211 |
| Outliers | 31 |

Table 4 Topic Distribution of BERTopic Model in the whole Tweet Data Set

90 percent of the words are assigned to topic 0 which is not desirable for human interpretability although the coherence score is much better than LDA. The topic representations are generally similar to that of LDA.

A group of colorful bars

Description automatically generated

Figure 14 Word Distribution in Topics of BERTopic Model

### 4.1.3 Comparing LDA and BERTopic

Table 4 shows the comparison between LDA and BERTopic on several features.

|  |  |  |
| --- | --- | --- |
| **Features** | **LDA** | **BERTopic** |
| Model Type | Generative | Embedding-based |
| Topic Interpretability | Interpretable | Interpretable (with word embeddings) |
| Preprocessing | Bag-of-words | Document Embeddings |
| Number of Topics | Must be predefined | Support topic reduction for better understanding |
| Scalability | Decreases with large corpus | Maintains scalability |
| Human Interpretability | Easy | Medium |
| Speed | Fast and inexpensive | Requires better CPU or GPU performance |
| Performance Evaluation | Perplexity, Coherence | Coherence, Visualization |

Table 5 Comparison of LDA and BERTopic on multiple features

#### 4.1.3.1 Advantages and Limitations of LDA

One of the reasons LDA is still important in the field of topic modeling includes its effectiveness, speed and flexibility in terms of input formats. Introduced in 2003, this model has continued to be used extensively. What is more, LDA can be applied to short or long documents, as well as small or big datasets. To illustrate, this project contains more than 140,000 tweets each having 60 words on average and its training process took less than three minutes on a typical laptop CPU; however, it took around 10 minutes when with BERTopic.

An issue with LDA is the lack of clear evaluation metrics when trying to assess its efficiency since they are data-driven soft clusters where it is hard to find an absolute measure for the optimal topic number. Users must determine the number of topics themselves while no direct way exists for determining what number is preferred over others. Coherence and perplexity scores are widely adopted when determining the right quantity of topics for an LDA model at the same time; however, their trustworthiness has been questioned as metrics. LDA deploys a representational model called bag-of-words, which allows the model to identify the themes of interest in each data set. Unfortunately, this method does not consider the semantic complexity that goes beyond that which is observable at surface level. The accuracy of the topic model results may be compromised if the meanings of words are not taken into consideration.

#### 4.1.3.2 Advantages and Limitations of LDA

BERTopic, a generatively enhanced model that is based on LDA and addresses the limitations of semantic understanding through inclusion of embedding part, can handle big data sets without constraints and has been shown to be competitive with other cutting-edge topic modeling algorithms such as LDA.

BERTopic incorporates embedding layers, which are a type of hidden layer in a neural network. These layers map input information from a high-dimensional to a lower-dimensional space, enhancing the network's ability to understand relationships between inputs and process data efficiently. By selecting the most suitable embeddings for the specific dataset and use case, users can customize the model to improve its performance. Unlike LDA, BERTopic addresses the challenge of determining the optimal number of topics. It supports hierarchical topic reduction, enabling a more nuanced understanding of relationships between topics.

BERTopic processes each tweet with one topic per tweet which is not true in practice since a tweet may cover different topics. Also, the process of implementing and adapting the BERTopic model is time-consuming and computationally demanding than it is with LDA. While BERTopic performs well on big data sets owing to its deep learning nature, this may not be the case when the data is small.

## 4.2 Sentiment Classification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Algorithm** | **Type** | **Precision (Accuracy)** | **Recall** | **F1 Score** |
| Multinomial Naïve Bayes (Default) | Average (Micro) | 0.86 | | |
| Macro Average | 0.87 | 0.93 | 0.84 |
| Weighted Average | 0.86 | 0.86 | 0.86 |
| Multinomial Naïve Bayes (parameter Tuned) | Average (Micro) | 0.9 | | |
| Macro Average | 0.89 | 0.88 | 0.88 |
| Weighted Average | 0.9 | 0.9 | 0.9 |
| Multinomial Logistic Regression | Average (Micro) | 0.89 | | |
| Macro Average | 0.91 | 0.87 | 0.89 |
| Weighted Average | 0.9 | 0.89 | 0.89 |
| Decision Tree | Average (Micro) | 0.95 | | |
| Macro Average | 0.94 | 0.94 | 0.94 |
| Weighted Average | 0.95 | 0.95 | 0.95 |
| Random Forest | Average (Micro) | 0.9 | | |
| Macro Average | 0.92 | 0.88 | 0.89 |
| Weighted Average | 0.91 | 0.9 | 0.9 |
| Light GBM | Average (Micro) |  |  | 0.9 |
| Macro Average | 0.92 | 0.87 | 0.89 |
| Weighted Average | 0.91 | 0.9 | 0.9 |
| XGBoost | Average (Micro) | 0.9 | | |
| Macro Average | 0.92 | 0.84 | 0.89 |
| Weighted Average | 0.9 | 0.9 | 0.89 |
| CatBoost | Average (Micro) | 0.79 | | |
| Macro Average | 0.86 | 0.74 | 0.77 |
| Weighted Average | 0.83 | 0.79 | 0.78 |
| Support Vector Machine | Average (Micro) | 0.89 | | |
| Macro Average | 0.91 | 0.87 | 0.88 |
| Weighted Average | 0.9 | 0.89 | 0.89 |

Table 6 Result Comparison of Sentiment Analysis Classification Algorithms

A graph of different colored lines

Description automatically generated with medium confidence

Figure 15 Performance Comparison of Sentiment Analysis Classification Algorithms

The sentiment analysis examination of numerous machine learning models unveils substantial performance metric differences. The default decision tree model performs better than other models like the hyperparameter-tuned multinomial naïve Bayes classifier. This highlights how imperative it is to carefully evaluate diverse algorithms since default settings might be good enough for some cases.

On the other hand, catboosting method shows worst performance among all the models tested. Its performance in sentiment analysis tasks does not compare well against other examined algorithms although it has potential advantages in dealing with categorical features and overfitting robustness. This shows that catboosting is not an ideal algorithm for sentiment analysis.

Equally, the other left models have almost similar levels of performance with only slight differences in evaluation measures. This supports the claim that when it comes to sentiment classification, these algorithms are on par although there might be underlying disparities posing various scenarios or datasets.

In addition, significant improvements have been noted by experimenting with various values in the multinomial naïve Bayes classifier. The model's accuracy is improved through fine-tuning from .86 to .9 which shows how well parameter optimization works when it comes to enhancing model performances. This underscores how important it is for fine-tuned algorithms to ensure comprehensive results that correspond best with task requirements as well as dataset properties. The best parameters are as follows:



However, one should note that out of all models, the support vector machine (SVM) has significantly higher training times. This increased length of time can be attributed to its built-in complexity and computational demands; it clearly shows that there should be a balance between how complicated a model is and how fast the training time will be.

The confusion matrices of all the classification algorithms can be found in section 7 Appendix.

### 4.2.1 Advantages and Limitations of Classification Algorithms in Sentiment Analysis

#### 4.2.1.1 Naïve Bayes

The advantage of multinomial Naive Bayes is that it is an easy method to implement and computationally efficient on big data sets. It works better on documents classification instances, especially if features (terms) are extracted before by a count or statistics technique of term occurrence. It allows for multiple class assignments which make it suitable for more than one purpose.

On the other hand, Multinomial Naive Bayes assumes features to be conditionally independent, but this is not always the case in real-world scenarios. It may fail to capture intricate relationships between features due to its simplicity. It will assign probabilities of zero to instances not found during training, such as rare occurrences or those with infrequent features.

#### 4.2.1.1 Multinomial Logistics Regression

Multinomial logistic regression advantageous because it generates predictions in probability whether a given text’s sentiment can be classified into any of the classes of interest, enabling users to make informed decisions based on the likelihood of each sentiment label. Regularization techniques can be used in logistic regression. These include L1 and L2 regularization, which prevents overfitting and improves generalization performance at the same time. Logistic regression can assess the importance of features, allowing you to understand which features are most valuable for prediction.

The drawbacks include the handling of complex data sets in which features and responses do not have a linear relationship. It is also prone to outliers. Its performance depends on how well the features are created for the sentiment analysis. If features are not selected or engineered properly, the model may perform poorly.

#### 4.2.1.3 Tree Models

Decision trees can be easily understood and interpreted unlike some other complex models. They provide insights into which features are most important for sentiment prediction. Nonlinear relationships between features and sentiment labels can be captured by using tree-based models. Random Forests and Gradient Boosting Machines (GBMs) are ensemble techniques that consolidate several decision trees to enhance prediction accuracy. This is achieved by minimizing overfitting while increasing generalization.

The limitations of tree models are that a single decision tree might generate dissimilar results on different data subsets because they have a higher variance. To overcome this issue, ensemble methods blend the predictions from many trees. Decision Trees are influenced negatively by noisy features in datasets or those that are irrelevant hence splits become suboptimal leading to low predictive accuracy. Although Decision Trees provide some insight into what is happening on a single tree, random forests and GBMs are nearly impossible to understand since they are considered as black boxes.

#### 4.2.1.4 Support Vector Machine

Having the ability to work with large numbers of attributes or intricate feature representations, SVMs are best suited for tasks such as sentiment analysis containing high amounts of features. They are also less prone to overfitting. To model intricate relationships between features and sentiment labels, SVMs exploit different kernel functions such as linear, polynomial, radial basis function to create better models.

The disadvantages of SVMs are when working with large datasets, SVMs can be expensive in terms of computation as well as memory due to the need to save support vectors as well as calculate kernel functions for each data point. SVMs are equipped with several hyperparameters (e.g., regularization parameter C or kernel parameters) which need a precise adjustment for the machine to yield the best possible results which adds additional time to the long training time. They are also black boxed in nature where the innerworkings are difficult to interpret.

# 5 Conclusion

## 5.1 Research Questions

**Which unsupervised model, LDA or Bertopic, provides better accuracy in dealing with tweets (short text) data and gives greater interpretability?**

Having gone through both quantitative evaluation metrics as well as human assessment, LDA performs better with topic interpretability where each covid related topic is separated out even though BERTopic’s coherence score is much better. BERTopic groups all the covid related topics into a single cluster which is not what the project desired. LDA has better computing time as well.

**Which supervised model, Naive Bayes, Regression, Tree Models or Support Vector Machine, has better performance in classifying the positive, negative and neutral tweets?**

The decision tree model has the highest performance although it may be prone to overfitting and have problems with unseen data. For this project, the decision tree model has the best accuracy with relatively fast computing time.

**Among those, what model can be further optimized, and hyper parameter tuned to improve performance in terms of better efficiency (less computing power) and accuracy?**

With careful consideration, multinomial naïve bayes is chosen to parameter tune because of its simplicity and computational efficiency. There is noticeable improvement in its accuracy after the tuning. Although there are other models that will outperform MNB when they are hyper parametrized, a personal laptop will not be able to handle the requirements.

## 5.2 Recommendations for Further Work

One way to improve the system might be to take advantage of some features provided by LDA model which include word search. When you select this feature, it would allow you as a user to provide what one wants from this field followed by giving potential topics related to that specific word. This can monitor particular terms from tweets for trend analysis purposes.

The tweets are from a three-month period and therefore, they are not generalized. Social media trends change rapidly, and it will be better to test the models, both topic and sentiment on a larger timeline with better computational power.

Better performing models such as tree models and support vector machine can be hyper parameter tuned to have a better prediction for classification if the necessary recourses are accessible.

# 6 References

Blei, D., Edu, B., Ng, A., Jordan, M. and Edu, J. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, [online] 3, pp.993–1022. Available at: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=http://githubhelp.com.

Charbuty, B. and Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, 2(01), pp.20–28.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, [online] pp.785–794. doi:https://doi.org/10.1145/2939672.2939785.

Churchill, R. and Singh, L. (2022). The Evolution of Topic Modeling. *ACM Computing Surveys*, 54(10s). doi:https://doi.org/10.1145/3507900.

Donges, N. (2021). *Random Forest: A Complete Guide for Machine Learning*. [online] Built in. Available at: https://builtin.com/data-science/random-forest-algorithm.

Grootendorst, M. (2022). *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*. [online] Available at: https://arxiv.org/pdf/2203.05794.

Grootendorst, M. (n.d.). *The Algorithm - BERTopic*. [online] maartengr.github.io. Available at: https://maartengr.github.io/BERTopic/algorithm/algorithm.html#visual-overview [Accessed 30 Apr. 2024].

Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J. and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, [online] 13(4), pp.18–28. doi: https://doi.org/10.1109/5254.708428.

Hofmann, T. (2017). Probabilistic Latent Semantic Indexing. *ACM SIGIR Forum*, 51(2), pp.211–218. doi: https://doi.org/10.1145/3130348.3130370.

IBM (2022). *What Is Logistic Regression?* [online] IBM. Available at: https://www.ibm.com/topics/logistic-regression.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In: I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds. [online] Curran Associates, Inc. Available at: https://proceedings.neurips.cc/paper\_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Khurana, D., Koli, A., Khatter, K. and Singh, S. (2022). Natural Language processing: State of the art, Current Trends and Challenges. *Multimedia Tools and Applications*, 82(3), pp.3713–3744. doi: https://doi.org/10.1007/s11042-022-13428-4.

Kibriya, A.M., Frank, E., Pfahringer, B. and Holmes, G. (2004). Multinomial Naive Bayes for Text Categorization Revisited. *Lecture Notes in Computer Science*, pp.488–499. doi: https://doi.org/10.1007/978-3-540-30549-1\_43.

Kit, Y. and Mokji, M.M. (2022). Sentiment Analysis Using PreTrained Language Model With No FineTuning and Less Resource. *IEEE Access*, 10, pp.107056–107065. doi: https://doi.org/10.1109/ACCESS.2022.3212367.

Konstantinas Korovkinas, Paulius Danėnas and Gintautas Garšva (2018). SVM Accuracy and Training Speed Trade-Off in Sentiment Analysis Tasks. *Communications in computer and information science*, pp.227–239. doi: https://doi.org/10.1007/978-3-319-99972-2\_18.

Kunsabo, J. and Dobša, J. (2022). A systematic literature review on topic modelling and sentiment analysis. In: *Faculty of Organization and Informatics Varazdin*. pp.371–380.

McInnes, L., Healy, J. and Melville, J. (2020). *UMAP: Uniform manifold approximation and projection for dimension reduction*.

Medhat, W., Hassan, A. and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), pp.1093–1113. doi: https://doi.org/10.1016/j.asej.2014.04.011.

Mehrotra, R., Sanner, S., Buntine, W. and Xie, L. (2013). Improving LDA topic models for microblogs via tweet pooling and automatic labeling. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’13*. doi: https://doi.org/10.1145/2484028.2484166.

Mohammad, S.M., Kiritchenko, S. and Zhu, X. (2013). *NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets*. [online] arXiv.org. doi: https://doi.org/10.48550/arXiv.1308.6242.

Nadkarni, P.M., Ohno-Machado, L. and Chapman, W.W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, [online] 18(5), pp.544–551. doi: https://doi.org/10.1136/amiajnl-2011-000464.

Papadimitriou, C.H., Tamaki, H., Raghavan, P. and Vempala, S. (1998). Latent semantic indexing. *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems - PODS ’98*, pp.159–168. doi: https://doi.org/10.1145/275487.275505.

Pascual, F. (2019). *Introduction to Topic Modeling*. [online] MonkeyLearn Blog. Available at: https://monkeylearn.com/blog/introduction-to-topic-modeling/.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. and Gulin, A. (2018). *CatBoost: unbiased boosting with categorical features*. [online] Available at: https://proceedings.neurips.cc/paper\_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

Ramos, J. (2003). *Using TF-IDF to Determine Word Relevance in Document Queries*. [online] Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b3bf6373ff41a115197cb5b30e57830c16130c2c.

Röder, M., Both, A. and Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM ’15*, [online] pp.399–408. doi: https://doi.org/10.1145/2684822.2685324.

Schubert, E., Sander, J., Ester, M., Kriegel, H.P. and Xu, X. (2017). DBSCAN Revisited, Revisited. *ACM Transactions on Database Systems*, [online] 42(3), pp.1–21. doi: https://doi.org/10.1145/3068335.

Silver, L., Huang, C., Taylor, K., Researcher, S., Cohn, S. and Cornibert, S. (2019). *In Emerging Economies, Smartphone and Social Media Users Have Broader Social Networks Digital technology users say they more regularly interact with people from diverse backgrounds FOR MEDIA OR OTHER INQUIRIES*. [online] Available at: https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2019/08/PI-PG\_2019-08-22\_social-networks-emerging-economies\_FINAL.pdf.

Singh, M. (2019). *Bounce SSO*. [online] www.servicenow.com. Available at: https://www.servicenow.com/community/performance-analytics-blog/how-to-measure-the-accuracy-of-your-sentiment-analysis-results/ba-p/2269161.

Srivastava, A. and Sutton, C. (2017). *Published as a conference paper at ICLR 2017 AUTOENCODING VARIATIONAL INFERENCE FOR TOPIC MODELS*. [online] Available at: https://arxiv.org/pdf/1703.01488 [Accessed 30 Apr. 2024].

Tong, Z. and Zhang, H. (2016). A Text Mining Research Based on LDA Topic Modelling. *Computer Science & Information Technology ( CS & IT )*, pp.201–210. doi: https://doi.org/10.5121/csit.2016.60616.

World Health Organization (2020). *Coronavirus Disease (COVID-19) Situation Report – 162*. [online] World Health Organization. Available at: https://www.who.int/docs/default-source/coronaviruse/20200630-covid-19-sitrep-162.pdf [Accessed 9 May 2024].

Axelborn, H. and Berggren, J. (2023). *Topic Modeling for Customer Insights A Comparative Analysis of LDA and BERTopic in Categorizing Customer Calls*. [online] Available at: https://umu.diva-portal.org/smash/get/diva2:1763637/FULLTEXT01.pdf [Accessed 3 Apr. 2024].

# 7 Appendix

This appendix is the collection of confusion matrices for the sentiment analysis classifications.

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a blue box

Description automatically generated

A screenshot of a blue box

Description automatically generated

A screenshot of a chart

Description automatically generated

A screenshot of a graph

Description automatically generated

A blue squares with numbers

Description automatically generated