Automated pharmaceutical study selection in systematic review using machine learning algorithms

Shubham Uniyal

A Master Thesis

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Intelligent Systems)

Supervisor: Professor Arthur White

August 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Shubham Uniyal

August 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Shubham Uniyal

August 19, 2022

Automated pharmaceutical study selection in systematic review using machine learning algorithms

Shubham Uniyal, Master of Science in Computer Science

University of Dublin, Trinity College, 2022

Supervisor: Professor Arthur White

In this thesis, an automated systematic review was conducted on two datasets using text mining and machine learning strategies on the abstracts in the pharmaceutical domain. Manual Systematic Literature Review had already been performed on the two datasets. The datasets were preprocessed and tokenised to ensure the availability of the best set of features for the process of feature generations. Both the datasets were split into training and test data proportionately to their structure, ensuring an appropriate amount of training and testing information and a similar split between inclusions and exclusions. Term Frequency - Inverse Document Frequency(TFIDF) is used to generate textual features from the particular corpus of documents. Features having maximum weights are used for one dataset, and features having weights closer to the variance were used for the other. 1:2 downsampling was also performed for one dataset to balance the extremely unbalanced dataset. Four Classification models namely: Logistic Regression, Support Vector Machine (SVM), Naive Bayes (Gaussian) and Bagged Classification and Regression Trees (Bagged CART) are used to make predictions on test segments of the datasets. Naive Bayes outperforms all models with a Sensitivity of 1, predicting 4/4 and 5/5 relevant documents in both the datasets. The results are also compared with similar works in the automated Systematic Literature Review domain. The analysis shows that the strategy to use machine learning algotihms for Systematic Literature Review looks promising and should be further explored.

Acknowledgments

I want to express my gratitude to my supervisor Professor Arthur White for his continuous guidance and valuable insights throughout this dissertation. I would also like to thank my second reader, Professor Mimi Zhang, for her beneficial feedback on further improving this thesis during the presentation. I am extremely grateful to the School of Computer Science and Statistics, Trinity College for providing continuous education support over the course of the year.

Shubham Uniyal

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Systematic Literature Review(SLR) is the process of analysing and reviewing existing and recorded work for a particular topic in a systematic manner. Referenced from Fink (2019) a Systematic Literature Review is defined as ”systematic, explicit, comprehensive, and reproducible method for identifying, evaluating, and synthesizing the existing body of completed and recorded work produced by researchers, scholars, and practitioners”.

The process is extremely prominent in a lot of domains for articulating and assessing information in an organised manner. In the pharmaceutical domain, the methodology usually involves searching publicly available medical databases and collating a list of documents relevant to it. The documents are then screened manually using abstract to further filter out the documents. The final set of documents are then manually reviewed through a full text analysis. The process is usually conducted by multiple people so as to reduce any inaccuracies.

Hence, the procedure takes a significant amount of time to be conducted since a lot of information has to be carefully analysed and documented. This is also because if any relevant information is missed by the reviewers it can be extremely consequential for the organization conducting the review. Hence, multiple reviewers are often utilised to represent as fact checkers for the process. Also, since a limited set of documents are preferred for the full text analysis, a lot of documents are filtered out at the abstract screening stage. Hence, abstract filtering is a very critical stage in the methodology.

Therefore, our research applies and analyses the usage of text mining techniques and machine learning models to perform Systematic Literature Review on a corpus of documents. The research focuses on generating text features from the abstract present in the corpus of documents and use machine learning models to predict the relevant set of documents Also, the corpus should have had a manual Systematic Literature Review conducted on it so that we can compare the accuracy of our methodology.

1.2 Research Question

The research question of this dissertation is to implement an automated identification of relevant studies in systematic review using text mining and machine learning strategies, focussing on a pharmaceutical context.
The research objectives put forth to evaluate this question are:

- To identify a corpus of documents relevant to Systematic Literature Review in the pharmaceutical context. A Manual SLR should also have been manually performed on the dataset.
- Extract and preprocess relevant data to make it feasible for text mining and machine learning techniques.
- Analyse and implement different machine learning techniques pertaining to text mining for features generations.
- Analyse and compare the different classification models to predict relevant documents for a certain set of keywords with maximum accuracy.

1.3 Motivation

The systematic literature review is considered to be a very effective methodology for collecting and analysing existing scientific knowledge of a specific scientific area in a systematic manner, and is therefore becoming increasingly common Shojania et al. (2007). This methodology is also recommended as a superior collection of evidence regarding the state of current knowledge in the required field, or to corroborate the existence or otherwise of a given relationship (Shojania et al. (2007), Stros and Lee (2015), Mulrow et al. (1997)).

In the pharmaceutical context, the process of information gathering is very comprehensive. Even a general process of finding definitions such as Personalised medicine (PM) is done using a process which starts with searching for relevant information on PubMed using relevant keywords (Schleidgen et al. (2013)). A data extraction process from abstract and full text is developed and then the articles are filtered down based on the relevance of the information present in the set of articles. To reduce complexity of the resulting list, summary categories are also developed from the data.

In a much more complex scenario such as in Nam et al. (2014) where the evidence of the efficacy of biological disease-modifying antirheumatic drugs (bDMARD) needed to be updated for European League Against Rheumatism (EULAR) recommendations, the process gets further more comprehensive. In this process research papers were screened for the period of January 2009 and February 2013 using databases such as Medline, Embase and Cochrane. The initial search results yielded 10265 articles out of which 134 were selected for detailed review. After further analysis of the filtered articles, a total of 51 full papers and 57 abstracts ended up meeting the inclusion criteria.

The complexity of the process is also not just limited to the amount of the data that has to be filtered and processed but the quality and diversity in the human resource that is required for it. This can be illustrated in a scenario such as in Ash et al. (2012), where the available evidence for the efficacy and safety of Nonsteroidal Anti-Inflammator drugs (NSAIDs) was reviewed. During the process, a total of 30 rheumatologists, several of whom were also epidemiologists, one dermatologist, one infectious disease specialist and one one infectious disease specialist worked together to gather the relevant evidence.

Thus, we can safely conclude that the process of Systematic Literature Review in the pharmaceutical industry is both time and resource intensive. Intuitively, one can infer that the initial screening where thousands of articles are reviewed is the most time consuming aspect of the entire process.

Therefore, an automated way to expedite and improve this process of filtering down the set of relevant articles for review can definitely be helpful to the research community. Machine Learning and Text Analytics strategies are expected to be instrumental in solving such scenarios where the filtering process is based on certain keywords.

1.4 Theseis Overview

Text mining and classification techniques are prominently used to analyse a corpus of documents in various domains. Hence, this dissertation attempts to perform an automated systematic review using text mining techniques on the abstract sections of two sets of data. Both the datasets had a manual SLR performed on it and different sets of files pertaining to the abstract screening and full text screening stage. Since, the methodology extracts textual features from a given corpus of documents, this dissertation firstly ensures that a correct set of prospecting features(words) are available for it.

Since the number of documents found relevant in manual SLR for the final stage(full text analysis) was extremely less, the dissertation focused on the abstract analysis stage. Both sets of data are first pre-processed by labelling the relevant documents that made it through the abstract screening phase. The corpus information was then filtered so that each document had only information that was relevant to our analysis present. Tokenization was then performed on the abstract of each document removing all common words and text irregularities and a tokenized version of the abstract was created. The two datasets were then split into training and test data ensuring almost the same proportions for splitting of included documents.

Term Frequency-Inverse Document Frequency(TF-IDF) and Latent Semantic Analysis are both explored for feature extraction from the two datasets and TF-IDF was subsequently evaluated more suited for the analysis. Features are extracted using TF-IDF for both the training datasets. Due to extremely imbalanced data present in one of the dataset, 1:2 Downsampling is performed to balance the number of inclusions to the number of exclusions. Four different classification models namely Logistic Regression, Support Vector Machine (SVM), Naive Bayes and Bagged Classification and Regression Trees (CART), are then trained and used for making predictions on the test data version of both datasets.

This dissertation is structured as follows: Chapter 2 lists out the relevant works done in each aspect of our analysis, Chapter 3 then displays the methodology followed for our analysis. Chapter 4 presents the results and experiments pertaining to our analysis and the final chapter concludes our thesis and lists out possible future work relevant to it.

Chapter 2

Literature Review

A similar objective to achieve Systematic Literature Review through the usage of machine learning model was achieved in Popoff et al. (2020). The analysis also predicted a reason for exclusion as per the PICOS (population, intervention, comparator, outcomes, and study design) reasoning framework (Maharaj et al. (2015)). A word frequency matrix was created and from that a document matrix was created for the feature generation. Rare words(words occurring less than 5,10,100) were also removed to increase the efficiency. Downsampling was performed to increase the efficiency of the models since the relevant documents tend to be 5-10% of the entire corpus.

The authors performed both abstract and full text analysis and found the accuracy of the abstract analysis almost at par with the full text analysis. SVM(with hyperparameters) was found to be the best model achieving Sensitivity of 1 in 3 out of 5 datasets. On average, the authors were able to achieve a recall of 75% of excluded documents and an accuracy of 83% in terms of successfully determining the reason for exclusion.

2.1 Systematic Literature Review

The process that is used for the systematic review is usually a multi step process where the relevant literature is first identified using a relevant keyword search in databases such as PubMed, CINAHL and COCHRANE (Popoff et al. (2020)). The document is then usually screened using abstract and titles to further narrow down the filtered set. This filtered set of documents then underwent a complete and comprehensive full text review to generate the final set of relevant documents.

This process is usually (but not limited to) conducted by 2 independent reviewers and their results are then collated to reduce the amount of errors. The reviewers need to have a clear understanding of the eligibility criteria for the set of documents for both the initial screening and the full text review (Higgins et al. (2019)). Figure 2.1 illustrates a similar process of filtering of the documents to examine the impact of rapid response teams on hospital mortality (Maharaj et al. (2015)).

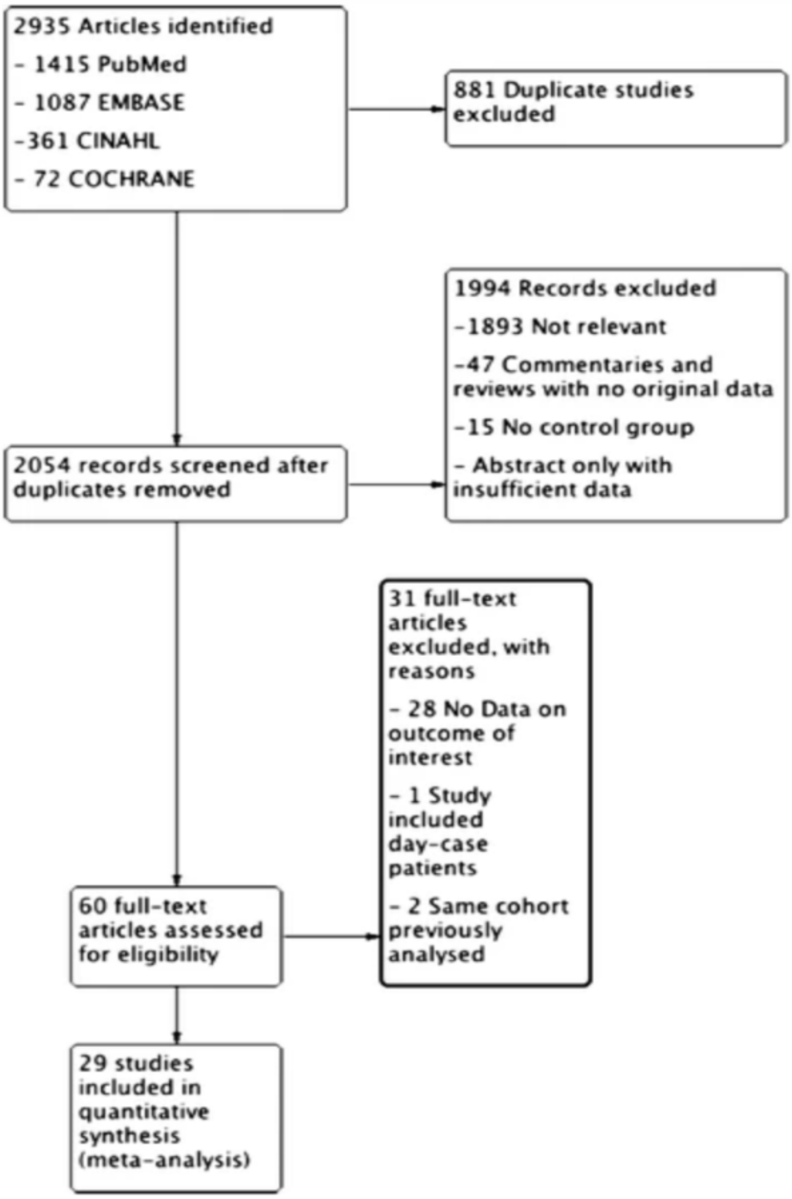
2935 Articles identified
- 1415 PubMed
- 1087 EMBASE
-361 CINAHL
- 72 COCHRANE
881 Duplicate studies excluded
1994 Records excluded
-1893 Not relevant
-47 Commentaries and reviews with no original data
-15 No control group
- Abstract only with insufficient data
2054 records screened after duplicates removed
31 full-text articles excluded, with reasons
- 28 No Data on outcome of interest
- 1 Study included day-case patients
- 2 Same cohort previously analysed
60 full-text articles assessed for eligibility
29 studies included in quantitative synthesis (meta-analysis)

Figure 2.1: Literature Search Flow diagram demonstrated in Maharaj et al. (2015)

There have been various studies indicating and encouraging the usage of machine learning and text mining techniques to improve the process of Systematic Literature Review in different domains (Adeva et al. (2014), Frunza et al. (2010), O'Mara-Eves et al. (2015)).

These algorithms are trained on sample data to learn the patterns or information that is relevant to the classification. The relevant information about the classification of data can be evaluated based on the training data that is exposed to the model. In the concept of SLR, the prevalent practice is to use abstract and title to generate training data (Adeva et al. (2014), Frunza et al. (2010), O'Mara-Eves et al. (2015)). There has also been evidence that the classification algorithm could be used as a fact checker or second screener for the purpose of reducing human error (Adeva et al. (2014)). The authors were able to reach a value of 84% in terms of overall precision and recall which is very promising.

This is also supported by Frunza et al. (2010) where a system-human performance matrix was generated for an automated systematic review classification. This was done to evaluate and update the performance of the generated model and render inferences based on the same. Figure 2.2 illustrates the architecture of the process used to build automatic text classification.

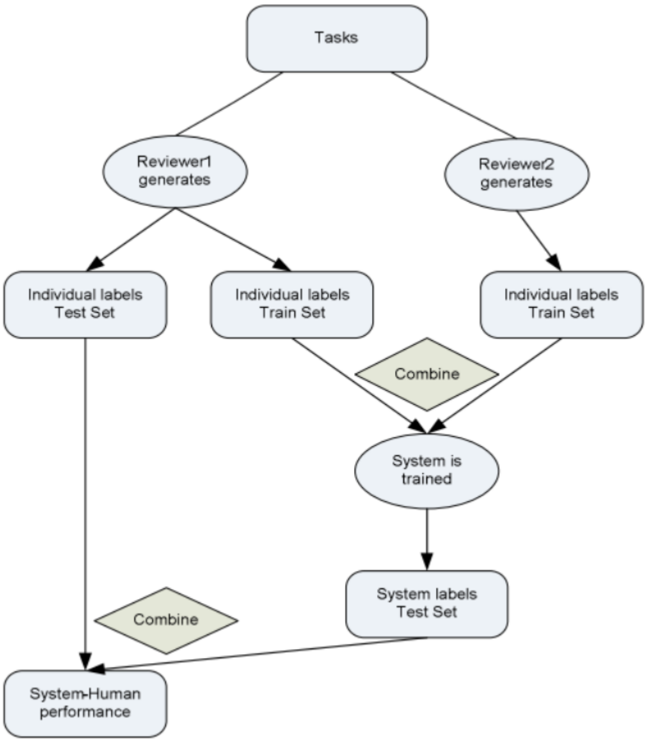

Figure 2.2: Embedding automatic text classification in the process of building a systematic review depicted in Frunza et al. (2010)

2.2 Feature Generation techniques

2.2.1 Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency Inverse Document Frequency (TF-IDF) is a very prominent approach in Natural Language Processing techniques for feature extraction. It is also expected to outperform other similar techniques such as Latent Semantic Analysis (LSA) and Linear Discriminant Analysis (LDA) when it comes to large datasets (Dzisevič and Šešok (2019)). It is also expected to perform better than standard BM25 feature extraction technique (Kadhim (2019)).

TF-IDF strategy is also effective in combination with other deep learning and NLP tools such as Word2Vec and Doc2Vec models where the score can be improved by considering the weights provided by TF-IDF vectorizer (Liu et al. (2018)). There are also instances where the tokenized documents are vectorised using the "TF-IDFVectorizer" for use as input to the SVM baked machine learning classifier (Kumar and Subba (2020)) as illustrated in Figure 2.3.

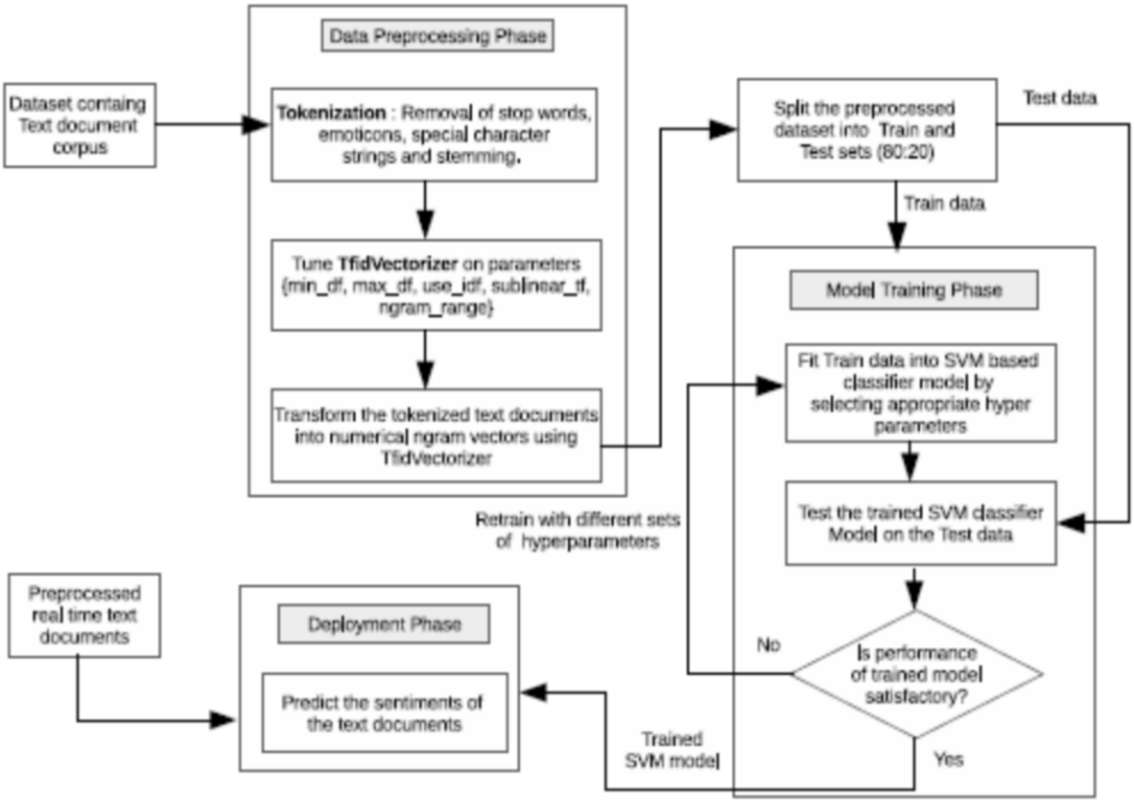
Data Preprocessing Phase
Dataset containg Text document corpus
Tokenization : Removal of stop words, emoticons, special character strings and stemming.
Tune TfidVectorizer on parameters {min_df, max_df, use_idf, sublinear_tf, ngram_range}
Transform the tokenized text documents into numerical ngram vectors using TfidVectorizer
Split the preprocessed dataset into Train and Test sets (80:20)
Test data
Train data
Model Training Phase
Fit Train data into SVM based classifier model by selecting appropriate hyper parameters
Test the trained SVM classifier Model on the Test data
Retrain with different sets of hyperparameters
No
Is performance of trained model satisfactory?
Yes
Trained SVM model
Preprocessed real time text documents
Deployment Phase
Predict the sentiments of the text documents

Figure 2.3: Architecture of the proposed TF-IDFvectorizer and SVM based sentiment analysis framework depicted in Kumar and Subba (2020)

In most cases TF-IDF is expected to be simple and effective, but this approach can have certain drawbacks. If the document corpus is extensive, this strategy can generate feature vectors with a large number of dimensions, which potentially could increase the chances to overfit the classification model Dzisevič and Šešok (2019). It is also expected that the TF-IDF model doesn't take similarities of the different features in account when generating the feature matrix which can also cause a decrease in the classification accuracy.

In Nafis and Awang (2021), a recursive filtering strategy of TF-IDF was used to obtain the optimal accuracy and enhance the feature extraction process. The authors first computed the term document matrix of the entire corpus and computed the variance of the weights assigned in it. The Features having a lower weighted score than the total variance were discarded and a new TF-IDF matrix was created. This process was done iteratively and the efficiency of the used model was tabulated. The most optimal set of features with the maximum accuracy were then finally used for evaluating the results.

2.2.2 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is also another technique that is widely promoted in the usage for feature extraction in an NLP framework. It is a modification of the TF-IDF strategy applying an algorithm for reducing dimensionality in the process of feature extraction (Dzisevič and Šešok (2019)). In Dong et al. (2006), the authors used LSA primarily because it is an efficient methodology for implementing summary based classification which was prevalent for the semantic analysis being executed.

In Nafis and Awang (2021), LSA was also used to extract semantic knowledge from the corpus of documents through feature extraction. The term frequency vector was then used to calculate the similarity of two similar blocks and to identify the pattern of a question sentence.

2.2.3 Bag of Words (BOW)

In Gabrilovich and Markovitch (2005), Bag of Words strategy was also implemented to check the performance of a text classification model with and without feature extraction and it was found that bag of words definitely improved the overall categorization. A similar strategy was also used in Alahmadi et al. (2013), where vector space modelling was implemented along with a bag of model strategy using a BOW scheme generating an 85% and above accuracy for the same.

2.3 Machine Learning Models

2.3.1 Logistic Regression

In Shah et al. (2020), a comparative analysis of different models was performed on documents pertaining to different categories using TF-IDF. As we can observe from Figure 2.4, the Logistic Regression model was able to outperform the rest of the models in most of the categories. A similar result was also achieved in Joshi and Abdelfattah (2021), where the authors used different machine learning models, including Logistic Regression to classify online reviews related to drugs. Logistic Regression along with TF-IDF classification was able to generate an accuracy score of more than 80%.

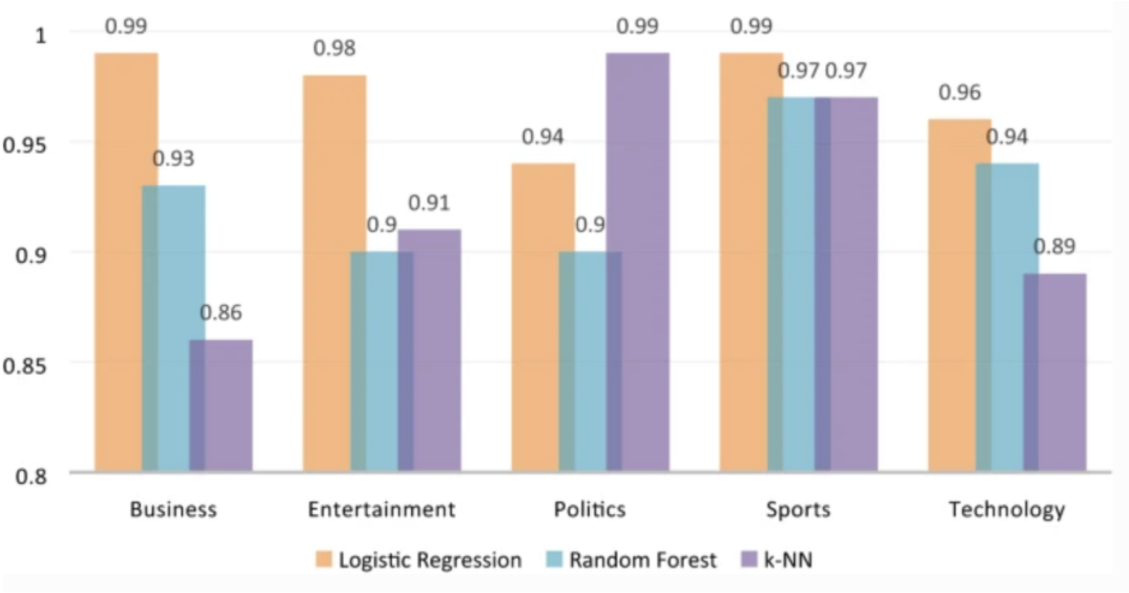

Figure 2.4: Categories versus precision showing variations of data set classes with respect to change in precision depicted in Shah et al. (2020)

2.3.2 Support Vector Machine (SVM)

In the text analytics and NLP domain, Support Vector Machine(SVM) modelling techniques are very prominently used for classifying the data in different classes Adeva et al. (2014). This modelling technique is also very widely used in the medicine domain as the penalty parameter can be altered easily to make changes for underfitting and overfitting the data(Dong et al. (2006)). As per Adeva et al. (2014), where automatic text classification has been implemented for systematic reviews, it is also expected to outperform other models in terms of accuracy and recall.

In Liu et al. (2010), where classification was performed for text categorization in fields such as sport, political, art, environment, SVM performed much better than the rest of the models. Figure 2.5 shows the accuracy comparison of SVM achieving more than 85% in all the topics.

However in Colas and Brazdil (2006), where the classification accuracy of SVM was compared with some older classification models such as Naive Bayes, it was found that the accuracy was almost at the same level. Despite varying the penalty parameter and using various kernels, the accuracy was more or less at par with the rest of the models. It was also concluded that while the strength of SVM lies in classifying non linear classification data, Naive Bayes and others are still worth considering for their simple implementation and much faster in use as compared to SVM.

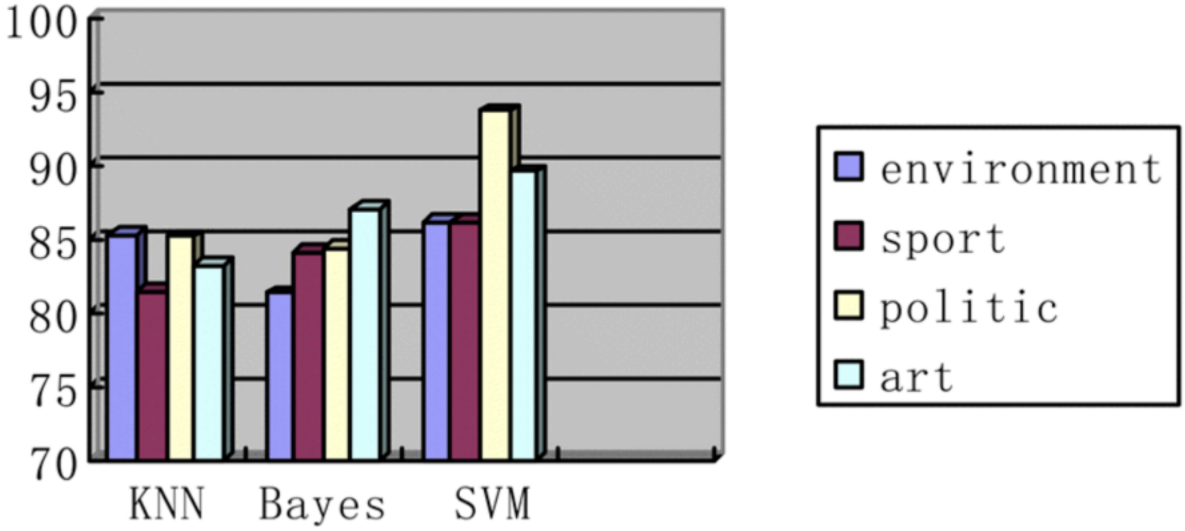

Figure 2.5: The comparison of Comprehensive index F1 value from Liu et al. (2010)

2.3.3 Naive Bayes

Some literature such as Frunza et al. (2010) also suggested that Naive Bayes performs better in the scenario where class imbalance is prominent. The authors performed automated selection of systematic reviews by using a per question classification method using an ensemble of classifiers including Naive Bayes and were able to get a recall of 70%-90% in different types of categorization of questions.

Naive Bayesian classification was also compared with SVM in Hassan et al. (2011) and it was concluded that naive bayes was a better choice if any external knowledge base corpus was used. It increased the accuracy by 29% over the baseline model that was implemented.

In Hassan et al. (2011), Naive Bayes performance was also compared with SVM on text classification and sentiment analysis of chapters in early American poetry novels. Both the models depicted high accuracy but Naive Bayes classifier was found to be the more optimal one as it displayed higher accuracy across all types of sections. Figure 2.6 displays the accuracy plot of both the models as the training set fraction is increased.

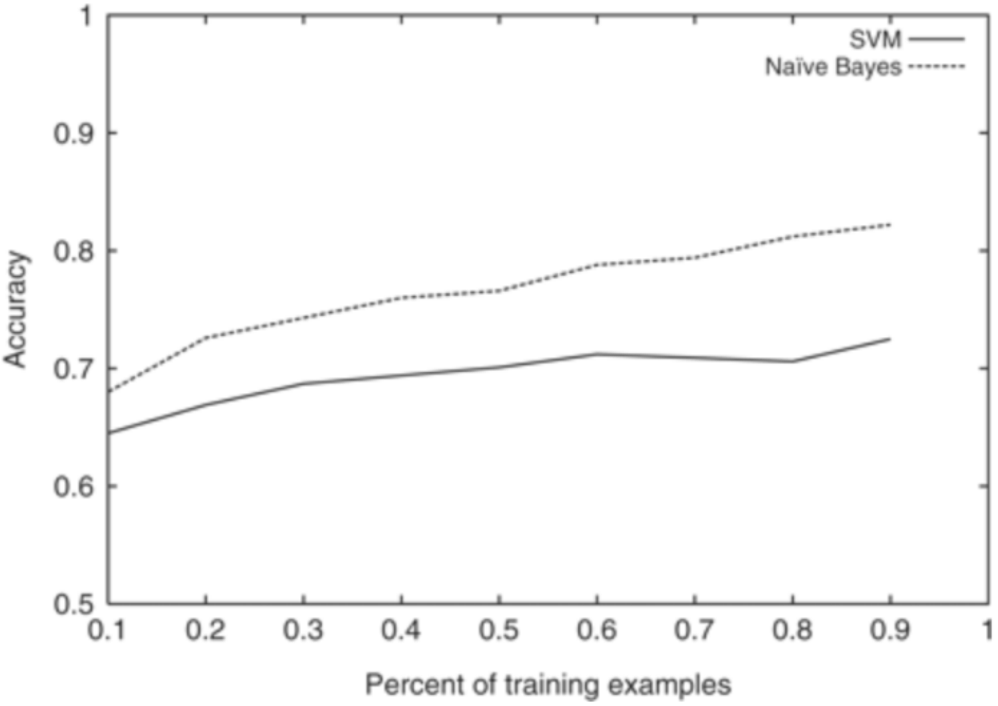

Figure 2.6: SVM and Naive Bayes learning curves depicted in Hassan et al. (2011)

2.3.4 Bagged Classification And Regression Trees (Bagged CART)

In Ibrahim et al. (2021) where a comparison study was drawn based between different types of models using Youtube Spam Collection Dataset based on 5-fold and 10-fold cross validation, the BAGGED CART model performed almost at par with the SVM in terms of accuracy. Figure 2.7 depicts the performance box analysis of all the models with CART and SVM both having almost similar accuracy.

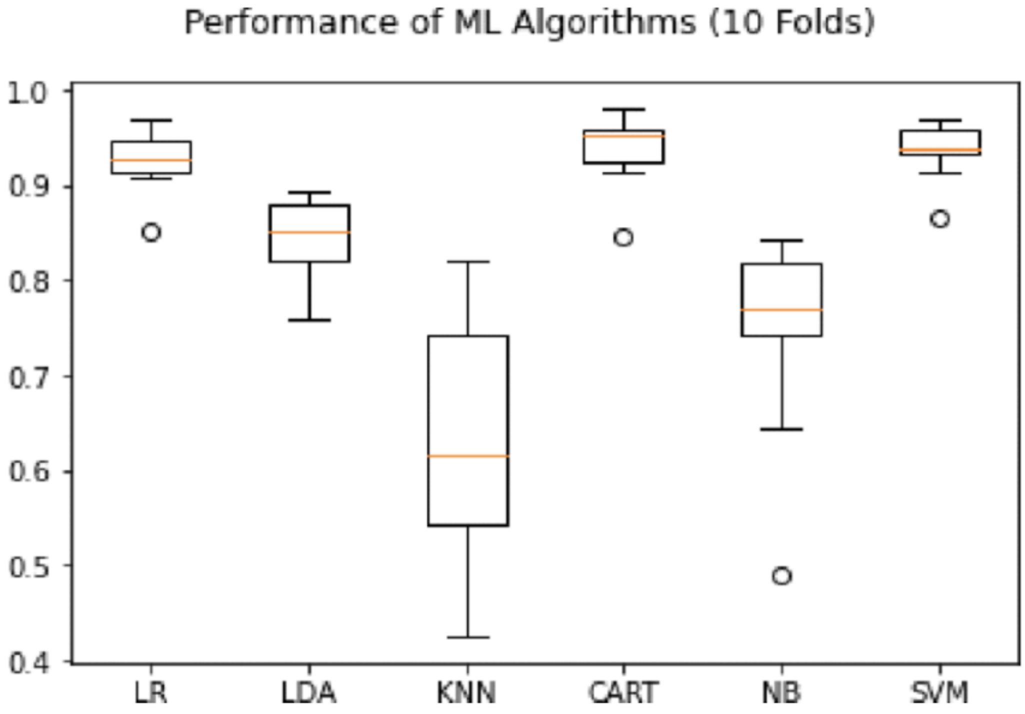

Figure 2.7: Performance of base machine learning algorithms using 10-fold cross validation presented in Ibrahim et al. (2021)

In Popoff et al. (2020), the Bagged CART classification model was also implemented for classifying the documents for inclusion and exclusion of the Systematic Literature Review of Medical data. It was found to be comparative in performance with SVM having an almost similar accuracy in both abstract and full text analysis.

Chapter 3

Methodology

3.1 Data Preparation

3.1.1 Data Description

The dataset used for analysis was obtained using electronic databases such as EMBASE, MEDLINE (via EBSCO), and CENTRAL (via the Cochrane Library). The data for the relevant studies searched from 01 January 2000 to 21 November 2020. Proceedings from the American Society of Hematology (ASH) and European Hematology Association (EHA) Annual Conferences were hand searched for the years 2014 to 2020. Terms used in searching of conference proceedings included: ‘tisagenlecleucel’, ‘ELIANA’, ‘ENSIGN’, ‘tisa-cel’, ‘blinatumomab’, ‘FLA-IDA’, ‘FLAG-IDA’, ‘acute lymphoblastic leukaemia’, and ‘paediatric’. EMA EPARs of tisagenlecleucel (7) and blinatumomab (8), and clinical trial reports from ClinicalTrials.gov (`www.clinicaltrials.gov`) were also searched. The filtering of articles were restricted to those published in English.

There were two datasets that were collated based on the Clinical and High Related Quality of Life (HRQOL) standards (Ashing-Giwa (2005)) divisions. Both dataset had a manual Systematic Literature Review conducted on it based on the standard PICOS (population, intervention, comparator, outcomes, and study design) reasoning framework (Maharaj et al. (2015)). Table 1 and Table 2 in the Appendix section depicts the inclusion and exclusion strategy used for Clinical and HRQOL divisions in the process, respectively.

The manual Systematic Literature Review had yielded two folders pertaining to both the divisions, respectively. In each of the folders, there were 3 documents present. Firstly, one with the list of entire citations generated from the keywords search from different electronic databases. Secondly, the ones that were selected for the full text analysis and finally, the ones that were found relevant at the end of the Systematic Literature Review. Table 3.1 and Table 3.2 depicts the data along with the document count in each stage for Clinical and HRQOL fields respectively.

Table 3.1: Manual Systematic Literature Review datasets for Clinical standards

File Name	File Format	No of Entries
All Clinical SLR Citations.txt	TY, AB, KW, ID, LA, M3, N1, PY, SN, SP ,ST ,T2, TI, UR, VL, ER	2132
All Clinical SLR Full text.xlsx	Title, Journal, Authors	51
All Clinical SLR included.xlsx	Title, Journal, Authors	2

Table 3.2: Manual Systematic Literature Review datasets for HRQOL standards

File Name	File Format	No of Entries
All HRQOL SLR Citations.txt	TY, AB, KW, ID, LA, M3, N1, PY, SN, SP ,ST ,T2, TI, UR, VL, ER	257
All HRQOL SLR Full text.xlsx	Title, Journal, Authors	21
All HRQOL SLR included.xlsx	Title, Journal, Authors	1

As it can be observed from Table 3.1 and Table 3.2, since the dataset in the final stage of inclusion has only 1 or 2 documents. The complexity for a machine learning model to predict 1 or 2 documents on such a limited dataset seemed excessive and hence we decided to model our analysis to screen the first stage of the systematic literature review. For example, for the Clinical dataset we would be focussing on generating predictions using All Clinical SLR Citations.txt and All Clinical SLR Full text.xlsx (51 documents included from 2132 documents).

Clinical Citations Dataset

Table 3.3 is a sample of a citation in Clinical citations dataset. The important attributes pertaining to our analysis are ID(Document Id), TI(Title), AB(Abstract) and KW (Keywords).

HRQOL Citations Dataset

The important attributes pertaining to our analysis are ID(Document Id), TI(Title), AB(Abstract) and KW (Keywords). The structure of the file was similar to that of the Clinical Dataset File (Table 3.3)

Table 3.3: Sample of Clinical Dataset

Keyword	**Value**
TY	JOUR
AB	Clofarabine [Clofrex™] is a purine nucleoside in development with...

C1	clofarex(Bioenvision)
	clofarex(Ilex Oncology)
	clofarex(Southern Research,United States)

C2	Bioenvision Ilex Oncology Southern Research(United States)
DB	Embase Medline
DO	10.2165/00126839-200405040-00005
IS	4
KW	antineoplastic agent clofarabine cytarabine DNA directed DNA polymerase beta orphan drug oxidoreductase purine purine nucleoside derivative ribose...
LA	English
M3	Article
N1	L39099110 2004-08-31
PY	2004
SN	1174-5886
SP	213-217
ST	Clofarabine
T2	Drugs in R and D
TI	Clofarabine
VL	5
ID	1476

For both the dataset it can be inferred that each document consists of a lot of attributes which are not relevant to our methodology and can be filtered out. Also, since in most of the research we found abstract analysis to be almost at par with the full text analysis (Popoff et al. (2020),Adeva et al. (2014),Frunza et al. (2010),O'Mara-Eves et al. (2015)), we focus most on the abstract text for data processing.

3.1.2 Data Preprocessing

As mentioned before, the dataset of each document had a lot of attributes that were not relevant to our experimentation. Also, our dataset was not cohesively present in one document. We went through different procedures to process the data for it to be ready for analysis discussed below.

Generate Citations data

Firstly, we went through the Full text files to recognise the files that were selected for full text review for both Clinical and HRQOL dataset. For these files we manually mapped the relevant file in the citations data with the label "IN" as True. Table 3.4 gives a reference to the the mapped document displayed in Table 3.3 (Clinical Dataset)

Table 3.4: Preprocessed version of Sample data

Keyword	**Value**
TY	JOUR
AB	Clofarabine [Clofrex™] is a purine nucleoside in development with...
TI	Clofarabine
KW	antineoplastic agent clofarabine cytarabine DNA directed DNA polymerase beta orphan drug oxidoreductase purine purine nucleoside derivative ribose...
ID	1476
IN	True
Others..	

Generate Processed File

We needed to generate a processed file which contained only relevant information for our analysis. Since, this was not an HTML or XML file, mark up language based adaptations couldn't be used. Hence, we implemented a strategy which worked on the below mentioned steps to generate the processed file.

1. **Filter out relevant data**
   We needed to filter out only the relevant attributes for our analysis. The relevant attributes included attributes such as ID(Document ID), AB (Abstract), IN(Included-Only present for included files) and KW(Keywords).

2. **Make KW one line text**
   KW attribute values were also present in separate lines. So we ensured that the processed file had KW value present in one line for faster analysis.

3. **IN attribute for each document**
   Also, since in the previous stage, the "IN" attribute tag was only present in the included documents. The processed files ensured that each document had an "IN" attribute with its value being True for those included in the full review file and False otherwise.

Table 3.5 displays the processed version of the preprocessed sample.

Table 3.5: Processed version of Sample data

Keyword	**Value**
ID	1476
AB	Clofarabine [Clofrex™] is a purine nucleoside in development with...
KW	antineoplastic agent clofarabine cytarabine DNA..
IN	True

Tokenizing the Abstract

Since we needed to generate the text features for our analysis, we decided to generate a tokenized file for both the Clinical and HRQOL processed file which contained a "TK" attribute having the value as the tokensied version of the abstract. For this purpose we implemented a tokenizer which followed the steps mentioned below for each document's abstract present in citations text.

1. **Remove unwanted text**
   This process removed all unwanted text such as single quotes, unwanted lines with special characters, digits and words containing digits, punctuation and non breaking new line characters. It also replaced all extra spaces with single space.

2. **Generate Tokens**
   In this process we first tokenized the abstract and generated tokens. These tokens were then shifted to lowercase. Finally, stopwords were filtered from tokens using the spacy library. A tokenized string was then generated using the tokens with a single space in between.

3. **Generate tokenized file**
   Firstly, we copied all the attributes from the processed file to the tokenized file. A new attribute "TK" was added to each document which consisted of the tokenized string from the spacy tokenizer.

Table 3.6 displays the tokenised version of the processed sample.

Table 3.6: Tokenised version of the sample data

Keyword	**Value**
ID	1476
AB	Clofarabine [Clofrex™] is a purine nucleoside in development with...
TK	clofarabine clofrex purine nucleoside development bioenvision . . .
KW	antineoplastic agent clofarabine cytarabine DNA directed DNA ...
IN	True

Training and Test Data

In order to create a balanced training data, we wanted to ensure that enough inclusions and exclusions are present for training the model. Since the document count for Clinical and HRQOL dataset was 2132 (51 inclusions) and 257 respectively (21 inclusions), we could observe that the data was unbalanced, especially for Clinical Dataset. Hence we decided to split the data in two different ways for Clinical and HRQOL datasets.

For the Clinical dataset, we went for a 90% (Training Data) and 10% Test data, since we wanted to ensure enough training data was available for the model to perform efficiently.

For the HRQOL dataset, we went for the standard 75% (Training Data) and 25% Test data, since we wanted to ensure enough positives for the test data to accurately measure model accuracy.

Train test split logic was performed in a manner that ensured that almost the same percentage of inclusions are available for both training and test data. Table 3.7 and Table 3.8 gives an overview of the training and test data for both the dataset.

Table 3.7: Training data information for Clinical and HRQOL Dataset

Dataset (Total Count)	**Training Data**	**Inclusions**	**Exclusions**
Clinical (2132)	1918	47	1871
HRQOL (257)	192	16	176

Table 3.8: Test data information for Clinical and HRQOL Dataset

Dataset (Total Count)	**Test Data**	**Inclusions**	**Exclusions**
Clinical (2132)	214	4	210
HRQOL (257)	65	5	60

3.2 Feature Generation

The next step in our process was feature generation using the abstract text from the corpus of the document. Since the HRQOL dataset is significantly less extensive than clinical dataset and the abstract length pertaining to a document is more or less same (same amount of words), the number of features required for analysis on HRQOL dataset are anticipated to be ideally less as compared to Clinical Dataset.

As mentioned in Chapter 2, we found that both Term Frequency - Inverse Document Frequency (TF-IDF) and Latent Semantic Analysis (LSA) were prominent for generating the features pertaining to a text corpus. Hence we considered them for the purpose of feature generation from the tokenized abstract.

3.2.1 Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF is a prominent methodology used in text mining for generating features (Dzisevič and Šešok (2019),Kadhim (2019),Kumar and Subba (2020)). As mentioned in Chapter 2, this methodology's performance is still comparable to novel adaptations(Kadhim (2019), Liu et al. (2018)). The documents in the corpus are interpreted as key factors for term weighting. TF-IDF is used to calculate text feature weights using the text corpus which can be used for model training. As explained in Liu et al. (2018), it is mainly composed of two parts, namely Term Frequency (frequency of words) and Inverse Document Frequency (frequency of inverse texts).

1. **Term Frequency(TF):** Term frequency refers to the number of occurrences of a given word in the file. Referenced from Liu et al. (2018), the formula for word frequency (TF) is as follows:

   $$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3.1}$$

   where total count of words in the corpus is denoted by k
   the count of occurrences of the word t_i in file d_j is denoted by n and,
   the summation of the occurrences of all words in the file d_j is denoted by $\sum_k n_{k,j}$.

2. **Inverse Document Frequency (IDF):** The inverse document frequency represents a measure of the general importance of a word. The entire number of documents is divided by the number of documents containing the text or word, which is then divided by the frequency of the inverse of the word. The quotient logarithm is then calculated. As mentioned in Liu et al. (2018), The formula for IDF is given as follows:

   $$idf_i = \log \frac{|D|}{|j : t_i \in d_j|} \tag{3.2}$$

   Where the total count of files in the corpus is denoted by $|D|$ and,
   the count of documents containing the word ti is denoted by $|j : t_i \in d_j|$"

Consequently, The weight for term i in document document ($w_{i,j}$) is determined as follows, if $tf_{i,j}$ is the term frequency of term i in document j, df_i is the document frequency of term i in the corpus and $|D|$ is the number of documents in the corpus:

$$w_{i,j} = tf_{i,j} \log \frac{|D|}{df_i} \tag{3.3}$$

This generates a term document matrix for the entire set of documents with the corresponding weights (as mentioned in the equation) for all the text features of the document. As we can observe from the corpus and also from Chapter 2, this will lead to a significant amount of feature generations that can cause overfitting of our model. Hence, we needed to identify the correct subset of features that should be used for our model training.

3.2.2 Latent Semantic Analysis (LSA)

Another well-known method is Latent Semantic Analysis (LSA), which was originally created to cut down on the amount of dimensions (features) that were produced in the document matrix. LSA is a variation of the TF-IDF methodology that interprets that phrases with similar meanings will frequently appear in the same locations throughout the text, which is a critical assumption of the algorithm (Dzisevič and Šešok (2019)).

By producing a collection of concepts connected to the documents, the LSA approach is used to examine relationships between a group of documents and the words they include. An extensive text corpus is utilized to create a matrix with word counts or weights per document (TF-IDF weights matrix), and each document is represented as a column vector in a word-document matrix. The Singular Value Decomposition (SVD) technique is then used to reduce the number of features(dimensions) while ensuring the structure similarity of the data.

As mentioned in Dzisevič and Šešok (2019), W (the Word document matrix) can be broken down into three matrices if K is the total ranks of W.:

$$W = USV^T \tag{3.4}$$

Where the diagonal matrix of singular values diagonal matrix with a size of (K X K) is denoted by S,
the left singular matrix with dimensions (M X K) is denoted by U and,
the right singular matrix with dimensions (N X K) is denoted by V .

The row column vectors of the singular matrix U are used to construct an orthonormal basis for the SVD transformation of the matrix's column vectors. The columns of SVT provide the vectors' coordinates. This basically indicates that the position of the document d_j in the R dimensions space is characterised by the column vectors S_vT_j or, equivalently, the row vector V_jS. Each row vector is linked to a document vector that is specifically connected to each document in the training set.

For every document that is not part of the training data, it is required to add the new document to the original training data and Latent Semantic Analysis would again have to be reevaluated. However, as SVD requires a lot of processing power, it should not be used ideally for every new test document.(Dzisevič and Šešok (2019)).

3.2.3 Comparing LSA and TF-IDF

We evaluated both the strategies for the purpose of feature generation and came to the conclusion that TF-IDF was more aligned with the analysis that we intended to perform. The key things considered are mentioned below.

1. As mentioned above, LSA is computationally very expensive as compared to TF-IDF and any new document would have to be added to the original training data, with Latent Semantic Analysis being performed again. Since, the data in both the dataset is extremely imbalanced, there is an expectation that we would have to split the dataset in a slightly different manner to maintain balance. Adding computational complexity would hinder the process significantly.

2. Another factor was the assumption in the LSA algorithm (mentioned above) that the terms that are similar in meaning have tendencies to occur in similar places of the sentence. While looking at the abstract present in the document corpus, we found that this assumption was not entirely true when it comes to analysing abstracts in a pharmaceutical context.

3.3 Evaluation Metrics

Since our analysis was going to be classification based to predict inclusion and exclusion of documents from the corpus for Systematic Literature review, we selected the below mentioned metrics for evaluation purposes. Table 3.9 defines the key terms used in our metrics and their meaning as per our scenario.

Table 3.9: Model metric term interpretation in our analysis

Term	Scenario Interpretation
True Positive	Correctly identified inclusions
True Negative	Correctly identified exclusions
False Positive	Incorrectly identified inclusions
False Negative	Incorrectly identified exclusions

3.3.1 Sensitivity and Specificity

Sensitivity (formulated for clinical terms in Lalkhen and McCluskey (2008)) is defined as the probability of a positive test, conditioned on truly being positive. It is also called True Positive Rate (TPR) or Recall.

Hence, Sensitivity is calculated as:

$$Sensitivity = \frac{number\ of\ true\ positives}{number\ of\ true\ positives + number\ of\ false\ negatives} \tag{3.5}$$

In our scenario it will be translate to:

$$Sensitivity = \frac{number\ of\ correct\ inclusions}{number\ of\ correct\ inclusions + number\ of\ incorrect\ exclusions} \tag{3.6}$$

Specificity (formulated for clinical terms in Lalkhen and McCluskey (2008)) is defined as the probability of a negative test, conditioned on truly being negative. It is also called True Negative Rate (TNR).

Hence, Specificity is calculated as:

$$Specificity = \frac{number\ of\ true\ negatives}{number\ of\ true\ negatives + number\ of\ false\ positives} \tag{3.7}$$

In our scenario it will be translate to:

$$Specificity = \frac{number\ of\ correct\ exclusions}{number\ of\ correct\ exclusions + number\ of\ incorrect\ inclusions} \tag{3.8}$$

3.3.2 Area Under Curve(AUC) of Receiver Operating Curve (ROC)

Receiver Operating Characteristic (ROC) Curve depicts the effectiveness of a binary classifier graphically as the classification criteria is altered. In ROC, the True Positive Rate (Sensitivity) is plotted on the y axis against the False Positive Rate on x axis.

The area under the ROC varies from 0 to 1 and the perfect classifier has the AUC of 1. Figure 3.1 displays some interpretations of using Aread under Curve of ROC curve. We chose this metric to graphically analyse the change in Sensitivity in different scenarios

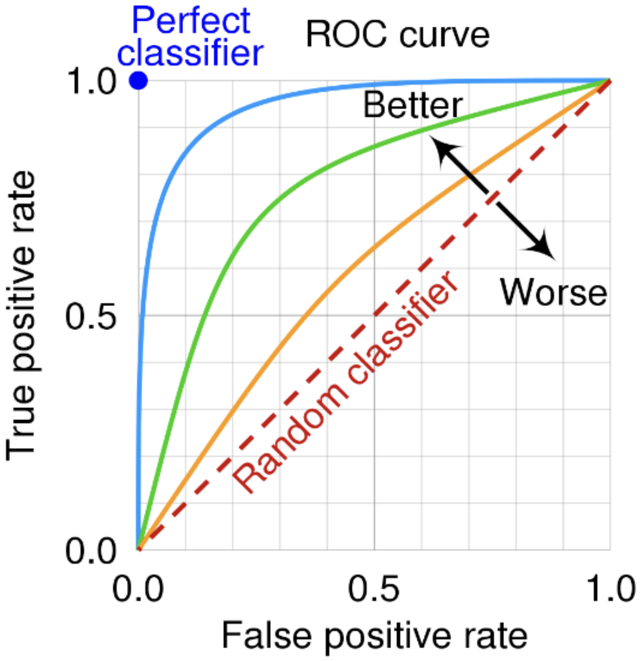

Figure 3.1: Area under curve from `https://commons.wikimedia.org/wiki/File:Roc_curve.svg`

throughout our analysis.

3.3.3 F1 Score

F1 score is also another prominent metric to measure a model's classification accuracy. The F1 score is calculated as the harmonic mean of precision and recall (Sensitivity).

Here, precision, sometimes referred to as positive predictive value, is the proportion of true positives to the total number of positives identified by the model.

In our scenario, precision will be defined as:

$$Precision = \frac{number\ of\ correct\ inclusions}{number\ of\ correct\ inclusions + number\ of\ incorrect\ inclusions} \tag{3.9}$$

Hence, F1 score is will be defined as:

$$F1 = \frac{2}{Sensitivity^{-1} + Precision^{-1}} \tag{3.10}$$

The F1 score varies from 0(if either precision or recall is 0) to 1 (both precision and recall are 1). While we do expect precision to be low, since the number of inclusions in a systematic literature review is very less as compared to the total documents, we wanted to ensure that precision was not zero and hence this metric was included in the analysis.

3.3.4 Prime metric for model performance (Sensitivity)

In a Systematic Literature review and especially in a medical or pharmaceutical context, an incorrect inclusion(False Positive) is a significantly less impactful error than an incorrect exclusion (False Negative).

In a scenario where a drug's impact on human health is being considered, false negatives can be pivotal and misleading, leading to serious implications. A model predicting high numbers of false negatives will never be considered efficient in the domain as key information could be present in the documents that were not predicted by the model. False Positives, on the can be manually screened off by the reviewer and thus will have little impact on the efficiency.

Therefore, in our analysis, the prime metric to determine the efficiency of the different models will be Sensitivity(Recall) so as to ensure a minimum amount of False Negatives.

3.4 Model Selection

For the purpose of evaluation, we chose four different types of classification models to utilise and analyse different strategies to improve the classification accuracy. Below are the models we used for classification of included or excluded documents for Systematic Literature Review.

3.4.1 Logistic Regression (Baseline)

Logistic regression is one of the most basic and extensively used classification algorithms. This algorithm uses a logistic function (Shah et al. (2020)) for the purpose of classification. A sigmoid function is primarily used as the logistic function in the Logistic regression model for the analysis. Figure 3.2 gives an example of the sigmoid curve that is used in the model.

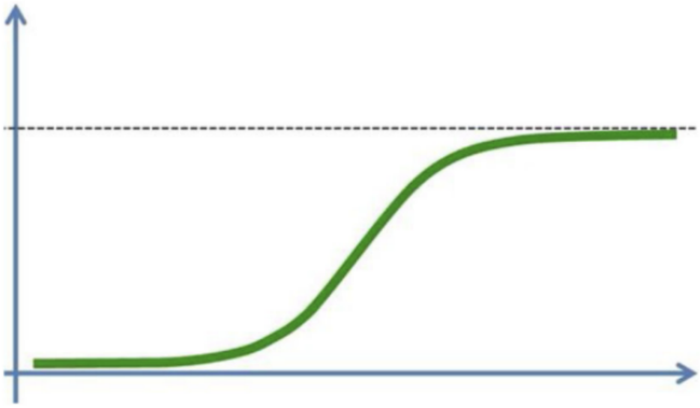

Figure 3.2: A standard Sigmoid Function as depicted in Shah et al. (2020)

As mentioned in Shah et al. (2020), the mathematical version of the logistic version using sigmoid function is defined as:

$$logit(S) = b0 + b1V1 + b2M2....bkVk \tag{3.11}$$

Where S is the likelihood that an interesting feature will be present, V1,V2...Vk are the predictor values and b0, b1....bk are intercepts of models

The Logistic Regression model predicts the text class in the form of a word vector by evaluating the vector of variables, coefficients for each input variable, and vector of variables(Shah et al. (2020)).

Hyperparameter tuning

The model has a penalty parameter which can be tuned to alter threshold for the analysis. This is primarily done to shrink the coefficients of less contributory features in cases where there are too many variables. Since our analysis will have an extensive feature set, adding penalties is also something that should be explored.

It also has a hyperparameter “C” which can be altered to better fit the training data. Higher values of C correspond to giving more weightage to training data.

Reason for selection

In the previous chapter, we found that logistic regression, whilst being a very simple implementation, still performed at par with other complex models in the text classification domain. Hence, we chose this model because we wanted to ensure we had a good baseline for our analysis.

Also, Logistic regression assumes that there are no linear relationships between independent and dependent variables in logistic regression and since our feature set is an independent word set, this is expected to hold true.

3.4.2 Support Vector Machine(SVM)

As mentioned in Chapter 2, Support Vector Machine Modelling is a very prominent modelling strategy used for the purpose of text classification. The SVM algorithm is trained by analyzing a hyperplane that serves as its decision surface and ensures the greatest margin of separation between the positive training data and the negative ones. Figure 3.3 depicts a linear separable SVM.

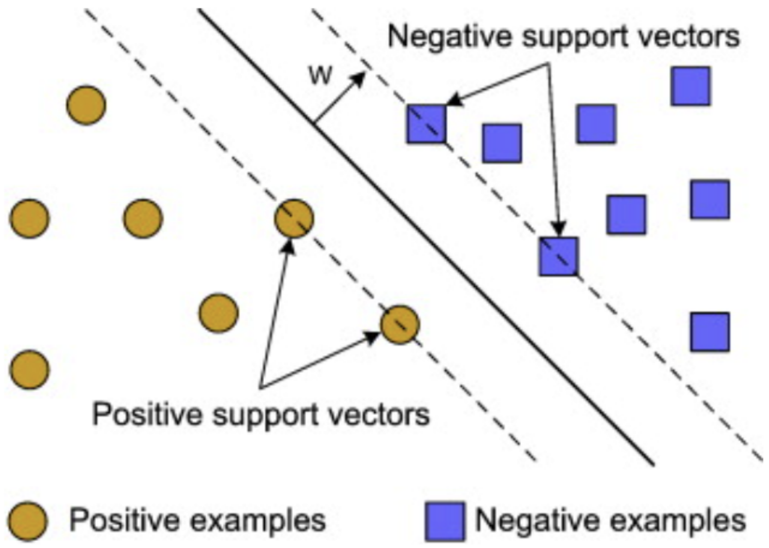

Figure 3.3: Linear Separable SVM as depicted in Shah et al. (2020)

As mentioned in Sun et al. (2009), the search for such a hyperplane can be described as a problem of minimization of w→, a vector perpendicular to the hyperplane that specifies its orientation.

After learning w→ and the location of the hyperplane, SVM uses a decision function to generate a score for an unlabeled text defined by its feature vector. The sign of the score is then used to predict the label. That instance, if the decision function predicts larger than 0, the document is labeled positive; otherwise, it is labeled negative. A portion of the positive and negative training instances, referred to as the positive and negative support vectors, respectively, define the learnt hyperplane.

Hyperparameter tuning

It has a hyperparameter “C” which can be altered to better fit the training data. Higher values of C will enable the optimiser to a smaller-margin hyperplane and conversely small values of C will enable the optimiser to generate a larger-margin separating hyperplane, even if that hyperplane misclassified some points.

Reason for selection

As mentioned in the previous chapter, we found that SVM is an extremely prominent model in text classification. It is regarded as an optimal strategy in classifying imbalance data, especially in the text mining domain. The model is expected to perform better than most due to the hyperplane estimation for a large number of feature sets that is generated in text classification and hence we wanted to explore the efficiency of the model.

3.4.3 Naive Bayes(Gaussian NB)

For text classification domains, Naive Bayes classifiers are frequently used. They are based on the conditional likelihoold of the features belonging to a particular class. These features are evaluated by feature selection methods. A major assumption in this modelling strategy is that the features should be independent of each other, which makes it an ideal model for text classification.

As mentioned in Zhang and Gao (2011), If we represent a set of documents as a vector of variables D(i), where d(i) corresponds to a letter, a word, or other properties of some text in practice, and C is a collection of predefined classes. The process of text classification includes assigning a class label c_j from C to a document. Bayes classifier is a hybrid parameter probability model defined as:

$$P(c_j|D) = \frac{P(D|c_j)P(c_j)}{P(D)} \tag{3.12}$$

Where the information from observations, which is the knowledge from the text itself to be classified is denoted by P(D), The distribution probability of document D in class space is denoted by $P(D|c_j)$ and The prior information on the occurring likelihood of the class is denoted by $P(c_j)$.

The posteriority of document D falling into each class is individually computed by the Bayes classifier after integrating these data. The document is then assigned into the class with the highest probability, determined by:

$$C^*(D) = \arg\max_j P(C_j|D) \tag{3.13}$$

Naive Bayes algorithm have different types of implementation based on the type of feature vectors used and since our features set will be a weighted set with continuous values and not discrete, we ended up using the Gaussian Naive Bayes model for our analysis.

Reason for selection

As mentioned in the previous chapter, we found that while most literature supported SVM as the prominent model in the text mining domain, there was significant evidence also to support that Naive Bayes performed at par if not better than SVM in many scenarios. The independent features assumption in the Naive Bayesian model could also be supported in our analysis. Also, we wanted to explore a more probabilistic approach (as is implemented in Naive Bayes) to analyse the classification. Hence, we decided to include the Naive Bayesian model also in our analysis.

3.4.4 Bagged CART

Bagging classifier is another ensemble technique which is designed to improve the stability and efficiency of the classification and regression process of machine learning models. A Bagged CART model is one variation of it where Bagging has been implemented over the Classification and Regression Trees (CART). As mentioned in Zhang and Gao (2011), its goal is to enhance the precision and consistency of machine learning algorithms used for classification and regression. In order to create the final forecast, it combines the classifications of training sets that are produced at random. With the addition of randomization to the construction process, these strategies are largely employed to reduce variance.

The Bagging process, also known as a ”smoothing operation,” is effective when attempting to raise the classification trees' forecast accuracy. The fundamental tenet is that a collection of ”weak trees” can be merged to create a stronger implementation. When a fresh set of data needs to be categorised, all the regression trees are exposed to it, and their analysis is assessed. Each tree offers a "vote," and the class with the most number of votes is predicted.

Reason for selection

As mentioned in Zareapoor et al. (2015), the prime reason for including a Bagged CART model in our analysis is that our data is extremely unbalanced and a regression tree analysis is known to be efficient in such a scenario. A Bagged Cart algorithm provides a good alternative where the model implements weighting of the results of the trees and reduces the variance of the dataset and overfitting.

3.5 Downsampling and Class Weights

In a Systematic Literature review process, the number of exclusions is often far more than the number of inclusions (Popoff et al. (2020)). This can also be observed by our clinical training dataset in Table 3.7 where there are only 47 inclusions as compared to 1918 exclusions. This is an extremely unbalanced dataset and can therefore affect the model accuracy considerably. HRQOL dataset is still far balanced and might not require other operations to be performed on it. But for the clinical dataset, we decided to explore the below mentioned strategies.

3.5.1 Class Weights

Adding class weights to the models is a strategy that is used to offset model predictions to a lower represented class. A similar methodology was used in Tan (2005), where a bigger weight was assigned for the smaller class nearby values in the dataset and a smaller weight was assigned for larger class nearby values in the dataset. Anand et al. (2010) also uses a weighted SVM to improve classification accuracy through misclassification of the two classes.

The process can be performed by adding a significantly higher class weight to the lower represented class (inclusions in our analysis) and a proportionately lower weight to the dominant class(exclusions in our analysis).

For example in our training clinical dataset (Table 3.7), since the ratio of the number of exclusions to the number of inclusions is almost 42:1 (1871: 47), the weight of inclusions that should be evaluated can be 10:1, 20:1 and so on. This can be used to improve the classification accuracy of our models for the extremely imbalanced training data for clinical dataset.

3.5.2 Downsampling

Downsampling is a process of random sampling from training data used to balance the imbalanced training data. The process is expected to work well to increase the sensitivity of the classifier for the low represented class (Prusa and Khoshgoftaar (2016)).

The process has to be performed carefully as it can cause a loss of information and hence a random downsampling pool is recommended (Prusa and Khoshgoftaar (2016), Cohen (2006)). Random downsampling refers to the process of choosing a random set of exclusion(dominating class) from the training data and using that to balance the dataset.

Downsampling can be performed in a proportionate way best suited to the analysis. 1:1 downsampling will have the same number of inclusions and exclusions. 1:2 Downsampling will have twice the number of exclusions than inclusions and so on. We decided to evaluate the different types of downsampling (1:3/1:2/1:1) and analyse which type was best suited to the clinical dataset.

3.6 Summary

As mentioned in the previous chapter, a similar analysis was performed in Popoff et al. (2020) and our entire methodology is consistent with the same. Feature generation was done using a document word matrix and rare words were also removed during the process. But we believed that rare words can have a significant impact when classifying documents in the pharmaceutical domain and hence we decided not to remove them. In Popoff et al. (2020), Downsampling was also performed with a 1:1 ratio, where random excludes were included as the same number as inclusions. Figure 3.4 displays the overall architecture of our analysis.

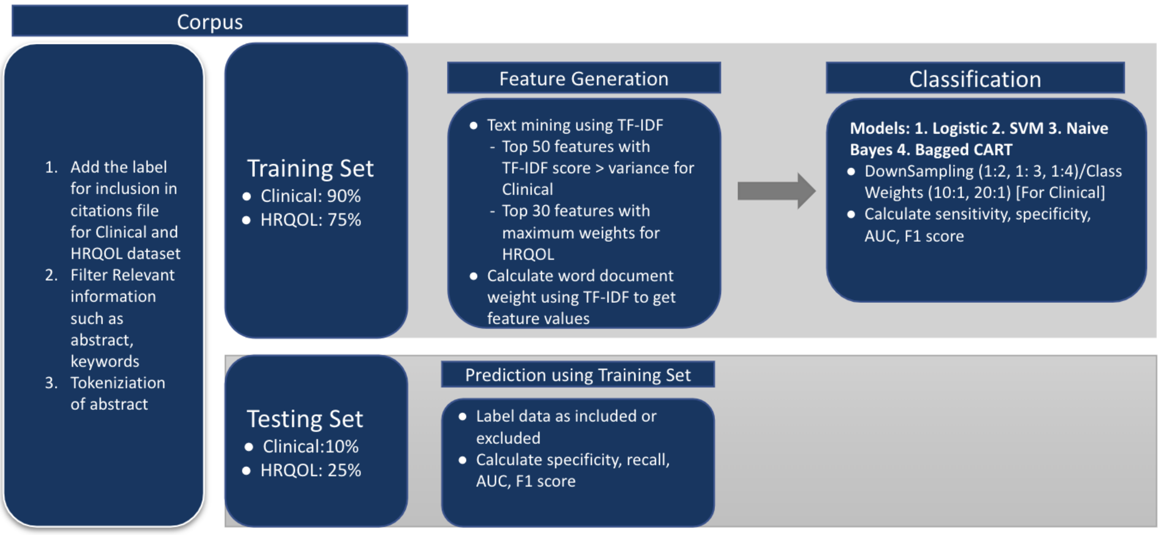
Corpus
1. Add the label for inclusion in citations file for Clinical and HRQOL dataset
2. Filter Relevant information such as abstract, keywords
3. Tokeniziation of abstract
Training Set
Clinical: 90%
HRQOL: 75%
Feature Generation
Text mining using TF-IDF
- Top 50 features with TF-IDF score > variance for Clinical
- Top 30 features with maximum weights for HRQOL
Calculate word document weight using TF-IDF to get feature values
Classification
Models: 1. Logistic 2. SVM 3. Naive Bayes 4. Bagged CART
DownSampling (1:2, 1: 3, 1:4)/Class Weights (10:1, 20:1) [For Clinical]
Calculate sensitivity, specificity, AUC, F1 score
Testing Set
Clinical:10%
HRQOL: 25%
Prediction using Training Set
Label data as included or excluded
Calculate specificity, recall, AUC, F1 score

Figure 3.4: System Architecture of our analysis

Chapter 4

Evaluation

4.1 Feature generation using TF-IDF

As mentioned in the previous chapter, we had anticipated that the number of features for training the Clinical and HRQOL dataset might be different. Since the number of words used in the abstract are almost the same throughout the dataset, the Clinical dataset generates a considerably higher feature count. The strategies used to analyse the features to be used for the two datasets are mentioned below.

We would also like to mention that in the below mentioned analysis, we plot the feature importance graph for the baseline model (Logistic Regression). As mentioned in Guyon and Elisseeff (2003), removal of features with a negative impact based on the score might not be the best strategy as the model might still be predicting correctly using other positive impact features. But, since our dataset is extremely unbalanced, we set a negative threshold value in our analysis based on the feature importance graph and ideally would want all features impact to be greater than that. This is done primarily because if a lot of features are having extremely high negative coefficients in our extremely imbalanced data, the probability of predicting an exclusion will increase significantly.

4.1.1 Clinical Dataset

After the document matrix was generated using TF-IDF vectorizer, a total of 10,108 textual features were extracted. As expected this number was huge and had to be reduced to a lower amount so as to not cause overfitting. Also, variance that was calculated using the by aggregating the individual column(features) weights and was evaluated to be 20.95.

We started analysing the top 50 features first to evaluate our model performance. These were evaluated by summing the column weight of the features across all documents and taking the top 50 features with highest values. We also analysed the feature correlation that was present. Figure 4.1 gives an overview of the initial top 50 features and the variance of the total dataset. Figure 4.2 gives the corresponding feature importance information using the Logistic Regression(baseline) model.

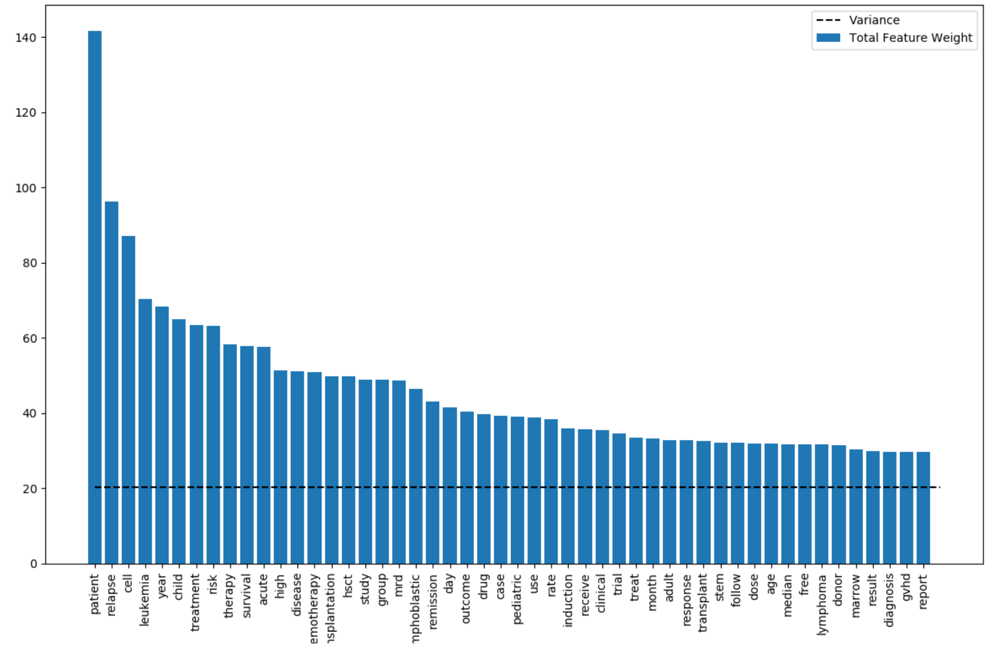

Figure 4.1: Top 50 features for clinical dataset

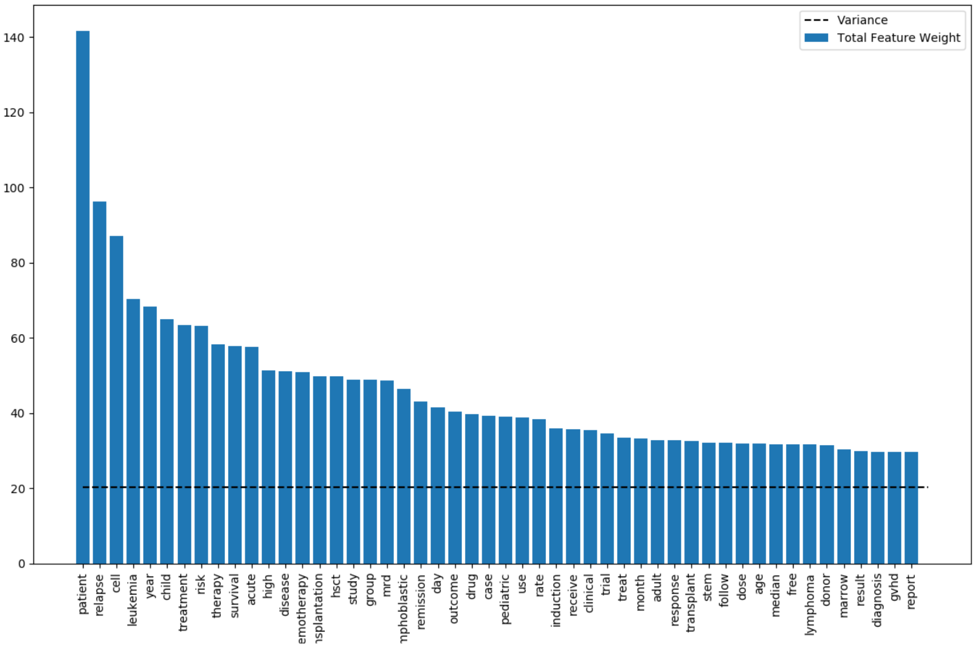

Figure 4.2: Features Importance graph for clinical dataset

There were 3 key inferences from Figure 4.1 and Figure 4.1:

1. There were many words such as "patients", "case" and others that, while being very prominent in the corpus, were having a very high negative importance value as compared to the rest of the features.

2. Most of the features were having negative importance were having values greater than -20. There are only 5 features having an importance value of less than -20. Hence, we decided to set our threshold at -20. This meant that, we would ideally want all the features to have an impact more than -20.

3. Most of the feature weights values were between 30 and 40.

As mentioned in the Chapter 2, we found evidence that textual feature extraction is found to be optimal when using those features with weights around variance. Hence, instead of using the top 50 features, we analysed the features within the range of (variance - 5 to variance +5). Figure 4.3 gives the list of final 50 features and their corresponding weights that we found in that range and Figure 4.4 gives the feature importance information using the SVM model.

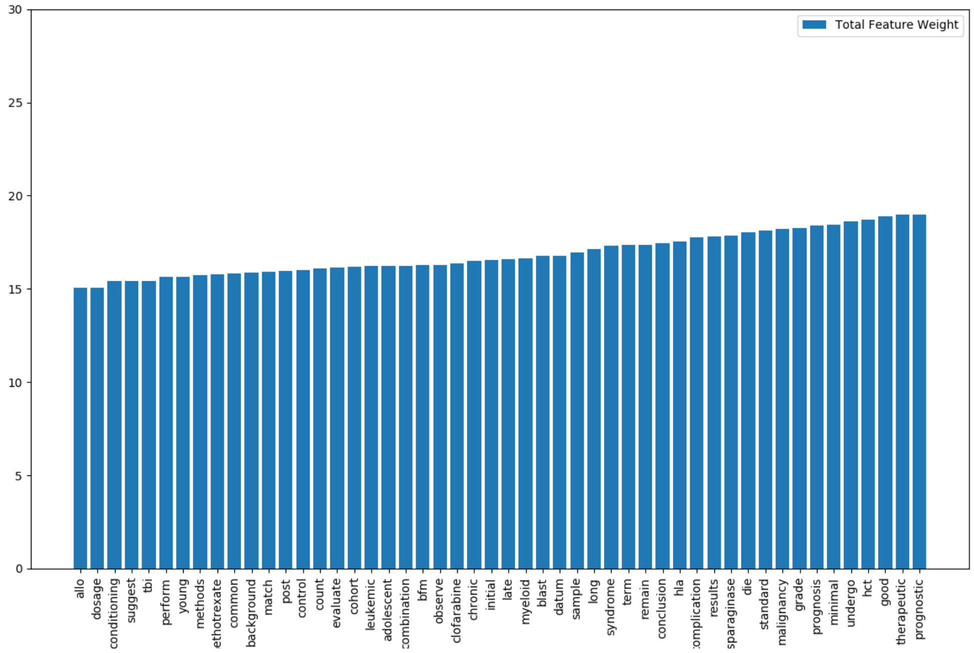

Figure 4.3: Final features for clinical dataset

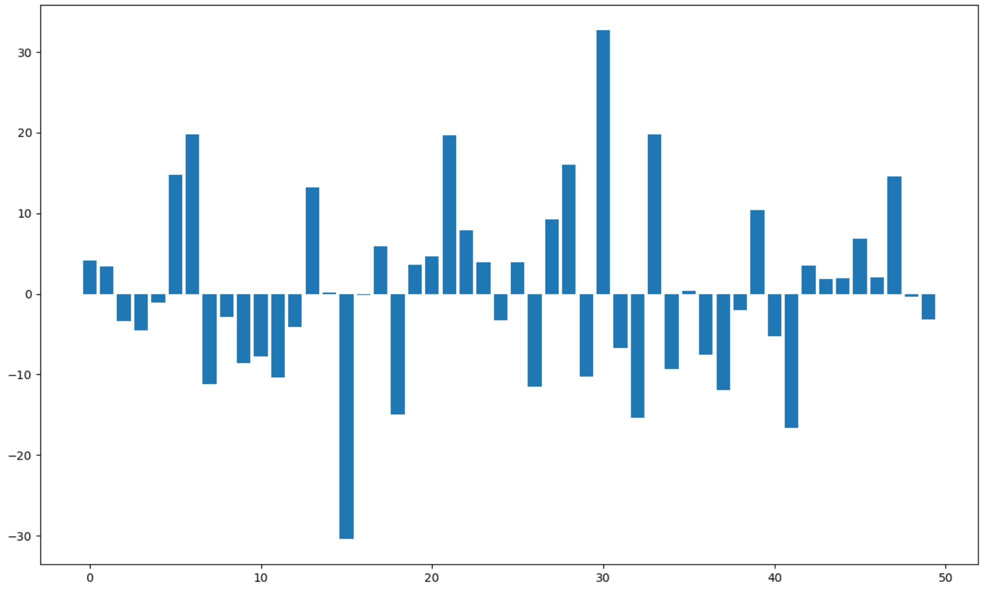

Figure 4.4: Final Features Importance graph for clinical dataset

As we can observe from Figure 4.4, the number of features having a value less than the threshold value(-20) is now reduced to 1 as compared to 5 earlier. Hence, we decided to use this as the final features list for our analysis pertaining to this dataset.

4.1.2 HRQOL Dataset

Similar to the clinical dataset, we first generated a document matrix using TF-IDF vectorizer and a total of 5122 textual features were extracted. As expected this number was huge and had to be reduced to a lower amount so as to not cause overfitting. Also, variance that was calculated using the by aggregating the individual column(features) weights and was evaluated to be 0.43.

In this scenario, since the corpus was substantially less as compared to the Clinical dataset, we started analysing the top 30 features first to evaluate our model performance. These were evaluated by summing the column weight of the features across all documents and taking the top 50 features with highest values. Since the variance in this dataset was extremely low with a lot of features just occurring once in the entire document set, we did not use variance as a feature filtering strategy in this dataset.

Similarly to the previous dataset, we also analysed the feature importance of our baseline model. Figure 4.5 gives an overview of the initial top 30 features and the variance of the total dataset. Figure 4.6 gives the corresponding feature importance information using the Logistic Regression(baseline) model.

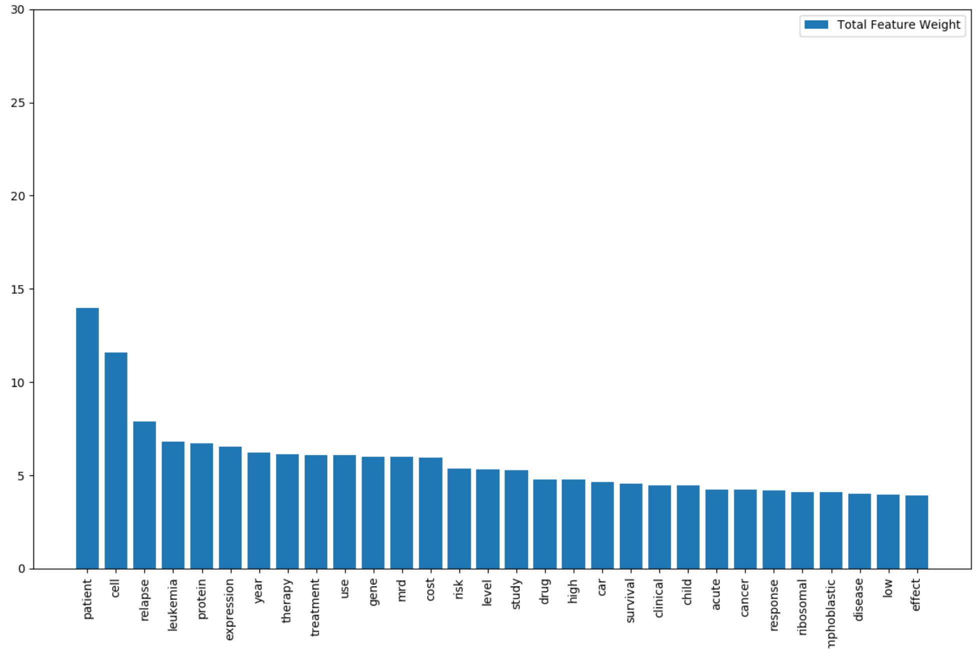

Figure 4.5: Top 30 features for HRQOL dataset

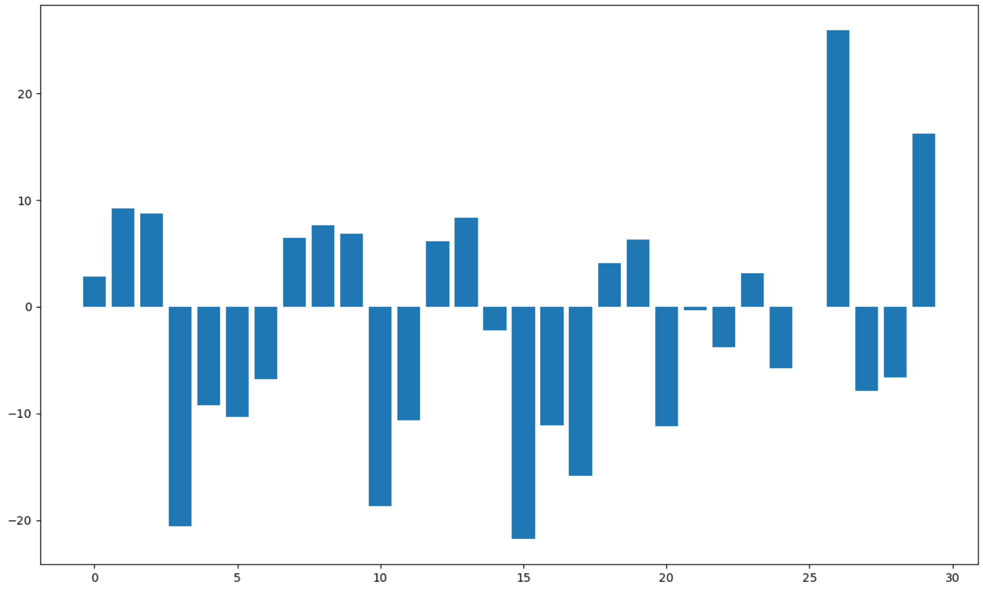

Figure 4.6: Features Importance graph for clinical dataset

As we can observe from Figure 4.5 and Figure 4.6, the weights of the top features are very less as compared to the clinical dataset. The threshold value can be inferred as

-20 for the analysis. While almost half of the features have a negative coefficient in the feature importance graph, only one value can be considered to have a negative impact (¡-20). Hence we decided to use the top 30 features only for our analysis in this particular feature set.

4.2 Class weights and Downsampling

As mentioned in Chapter 3, these strategies are used for balancing an unbalanced dataset to enhance the accuracy of the prediction model. Since in our analysis, the Clinical training dataset is extremely unbalanced as per Table 3.7 (1871 exclusions and 47 inclusions), we had anticipated the usage of these strategies for it. HRQOL training dataset (Table 3.7) was found to be much more balanced as compared to the Clinical dataset and these strategies were not used for the same. Table 4.1 gives an overview of the different models accuracy on the Clinical dataset before applying any of the class balancing strategies.

Table 4.1: Initial Accuracy Metric for Clinical Dataset

Model	Sensitivity	Specificity	F1 score	AUC
Logistic	0	1	0.98	0.4
SVM	0	1	0.98	0.5
Naive Bayes	0	0.85	0.83	0.28
Bagged CART	0	0.99	0.97	0.54

As we can infer from Table 4.1, sensitivity in no model exceeds 0 because of the highly imbalanced data.We can also conclude from Table 3.8 that since 98% of the clinical test data are exclusions, most of the models are only predicting exclusions and generating a high accuracy score as a result. Hence, we decided to use the class balancing strategies for the clinical training dataset and use our baseline model (Logistic Regression) as the rationale of this analysis.

4.2.1 Class weights

As mentioned in the previous chapter, since our clinical training dataset is extremely unbalanced, we decided to use extreme weights for inclusions as compared to inclusions. Since the summation of the weights of both the classes is expected to be 1, the exclusion class weight was taken to be the difference of 1 and inclusion class weight. Figure 4.7 gives an overview of how the sensitivity and specificity changes when the inclusion class weight is varied using the Logistic Regression model.

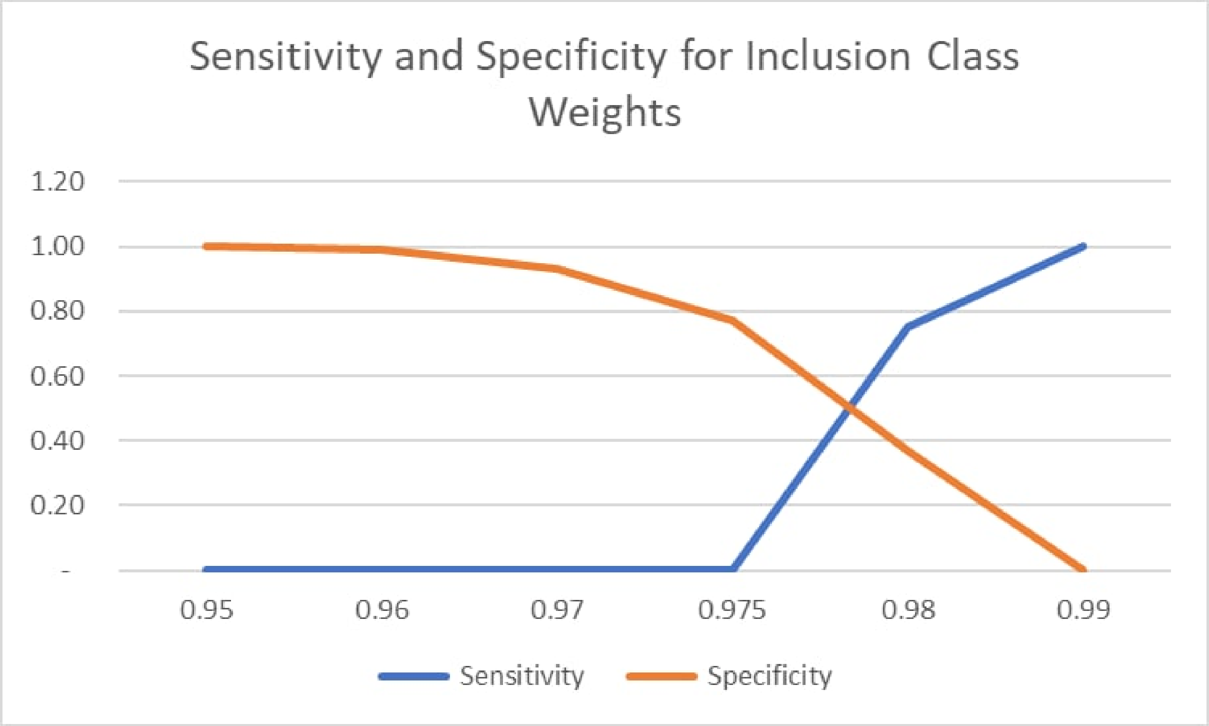

Figure 4.7: Variation in Sensitivty and Specificity with different inclusion class weights

As we can observe from Figure 4.7, sensitivity and specificity attain the most optimal value when inclusion class weight is around 0.98, before getting completely biased towards inclusions (Sensitivity=1, Specificity=0). Consequently, the exclusion class weight is 0.02 for the same. But when we evaluated the model predictions, we found that it was still predicting more than half the documents (135 documents out of 214) as inclusions.

Also, the strategy was extremely unscalable, with the value of sensitivity changing very quickly in a very short range of weights. For both of the above listed reasons, we decided to not use class weights in our final analysis.

4.2.2 Downsampling

As mentioned in the previous chapter, Downsampling is another strategy used for balancing an extremely unbalanced dataset. We took a proportionate sample of exclusions from our training data to balance the underwhelming class (inclusions) in our dataset. Possible examples include 1:1 downsampling (Same number of inclusions and exclusions), 1:2 downsampling (twice the number of exclusions to inclusions), 1:3 downsampling (thrice the number of exclusions to inclusions). We decided to evaluate all 3 of the above mentioned downsampling for our analysis. Figure 4.8 gives an overview of the sensitivity and specificity of the different types of downsampling using the Logistic Regression (Baseline) model.

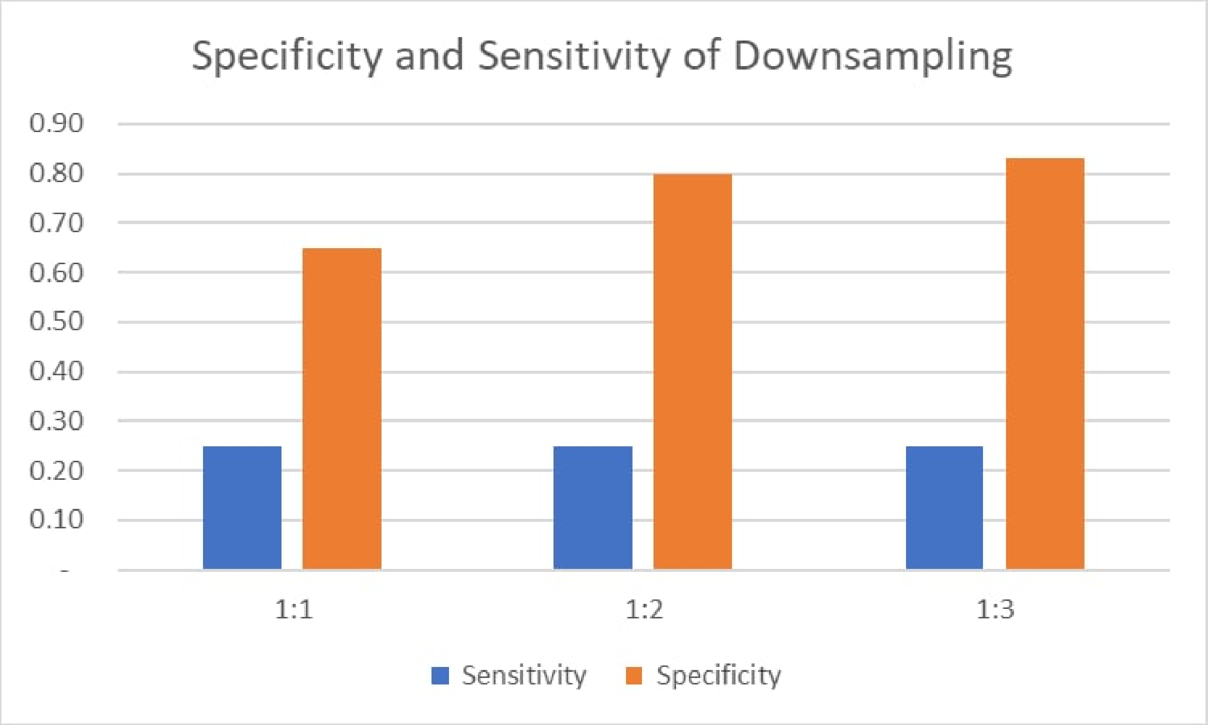

Figure 4.8: Variation in Sensitivity and Specificity with different types of downsampling

We can observe from Figure 4.8 that the results are much more promising than usage of class weights. The sensitivity remains constant for the different types of downsampling performed. Specificity increases as the exclusion count is increased, but the increase is very less between 1:2 and 1:3 downsampling (0.82 and 0.83 respectively). Hence, we decided to use 1:2 downsampling to balance the classification process of the clinical dataset.

4.3 Model Evaluation using metrics

As mentioned in Chapter 3, the metrics used for our analysis are Sensitivity, Specificity, Area under ROC curve and F1 score. Sensitivity will be the prime metric for our analysis as false exclusions (False Negatives) are detrimental in the process of Systematic Literature review.

4.3.1 Logistic Regression (Baseline)

As displayed in Figure 4.8, we were able to achieve a sensitivity of 0.25 with the Logistic Regression Model for clinical dataset. The sensitivity value was slightly on the lower side to be considered a baseline hence we decided to perform hyperparameter analysis.

Hyperparameter Tuning

As discussed in Chapter 3, we added the penalty "l2" in the methodology and varied the C parameter to increase sensitivity. Figure 4.9 displays the sensitivity recall along with the different values of C.

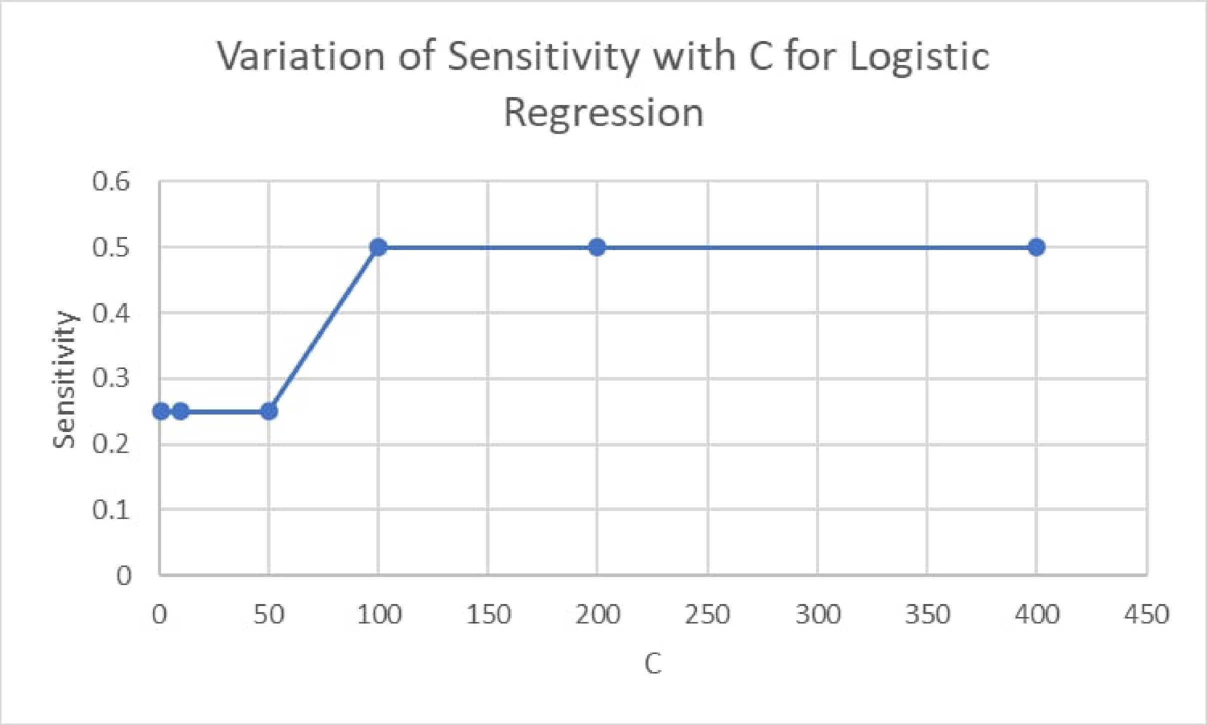

Figure 4.9: Variation of Sensitivity with Hyperparameter C for Logistic Regression model

We can infer from Figure 4.9 that the sensitivity attains a maximum value of 0.5 for C=100 and remains the same after it. Hence, we decided to use the minimum value of C (100) for our analysis to avoid overfitting. Table 4.2 gives the final logistic regression evaluation for both Clinical and HRQOL dataset and, Figure 4.10 and Figure 4.11 displays the ROC curves for the same.

Table 4.2: Logistic Accuracy Metrics for Clinical and HRQOL datasets

Dataset	Sensitivity	Specificity	AUC	F1 Score
Clinical	0.5	0.8	0.76	0.79
HRQOL	0.4	0.98	0.95	0.94

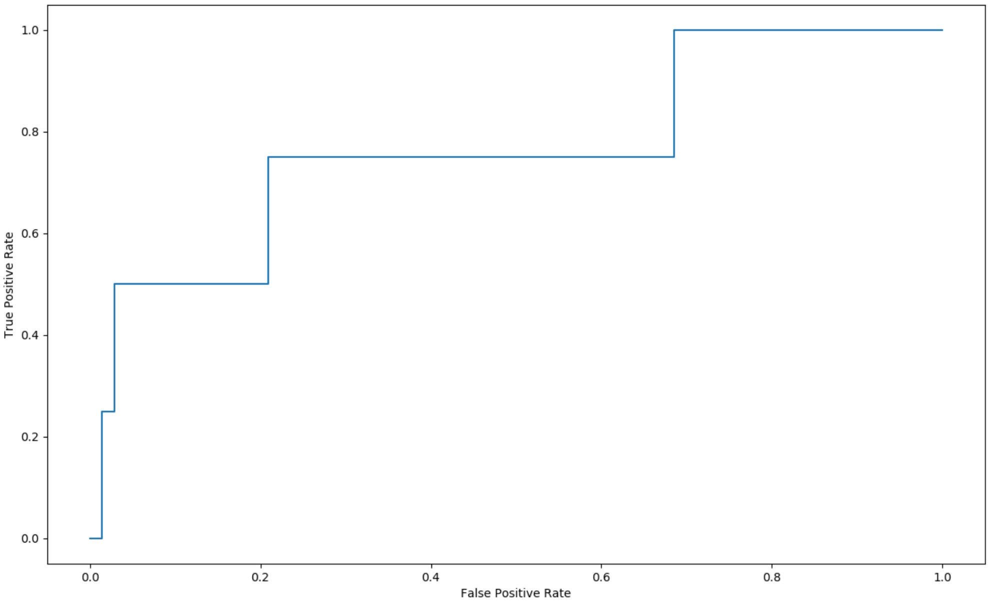

Figure 4.10: ROC curve of Logistic Regression model for Clinical Dataset

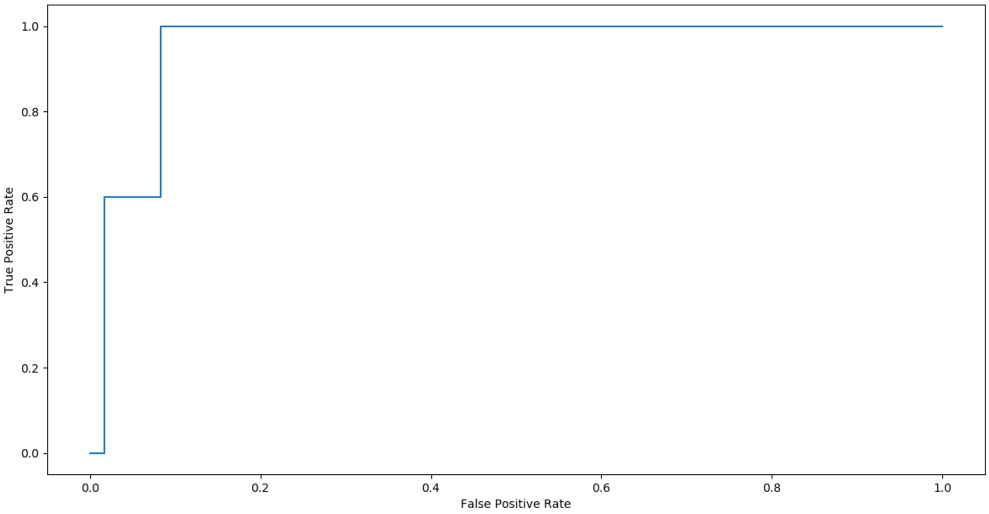

Figure 4.11: ROC curve of Logistic Regression model for HRQOL Dataset

As we can infer from Table 4.2, Figure 4.10 and Figure 4.11, the logistic regression model was able to predict 2 relevant documents out of 4 for the clinical dataset(2/4) and 2 relevant documents out of 5 for the HRQOL datasets (2/5).

4.3.2 Support Vector Machine (SVM)

Without no hypertuning, the sensitivity evaluated for the SVM Model was 0 for the clinical dataset. We then proceeded to perform hyperparameter tuning to increase the sensitivity

Hyperparameter Tuning

As discussed in Chapter 3, we varied the C parameter to increase sensitivity. Figure 4.12 displays the sensitivity variation along with the different values of C.

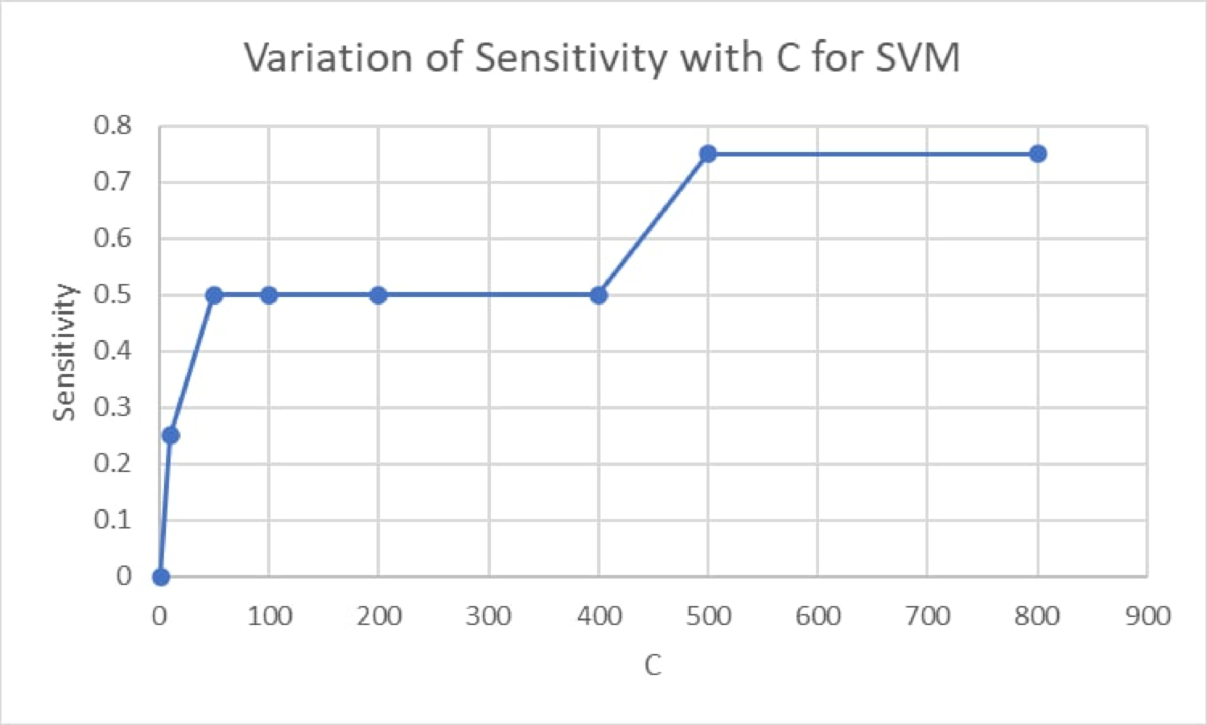

Figure 4.12: Variation of Sensitivity with Hyperparameter C for SVM model

We can infer from Figure 4.12, sensitivity attains a maximum value of 0.75 for C=500 and remains the same after it. Hence, we decided to use the minimum value of C (500) for our analysis on both the datasets to avoid overfitting. Table 4.3 gives the final SVM model evaluation for both Clinical and HRQOL dataset and, Figure 4.13 and Figure 4.14 displays the ROC curves for the same.

Table 4.3: SVM Accuracy Metrics for Clinical and HRQOL datasets

Dataset	Sensitivity	Specificity	AUC	F1 Score
Clinical	0.75	0.76	0.78	0.76
HRQOL	0.6	0.92	0.93	0.89

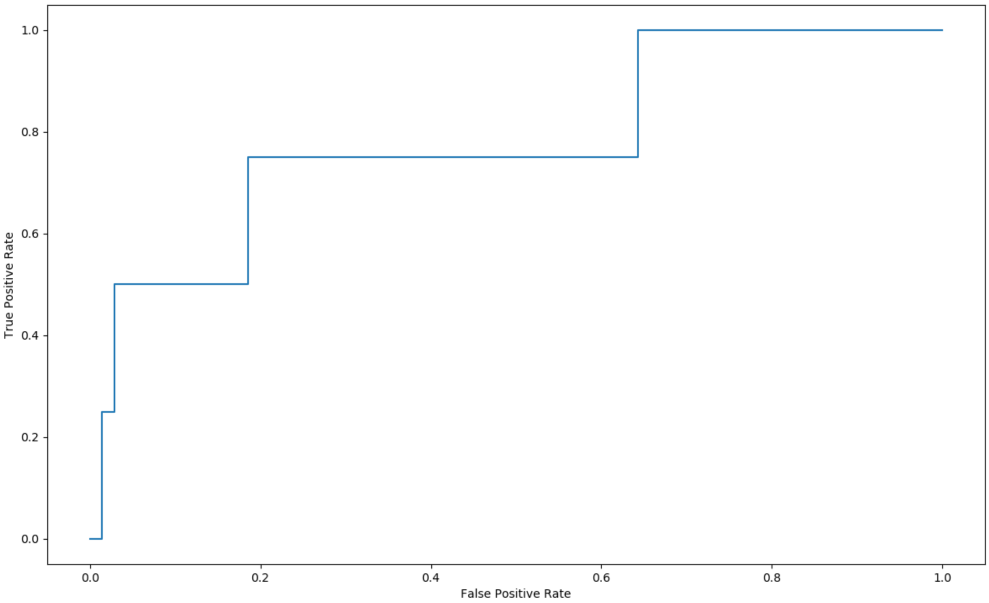

Figure 4.13: ROC curve of SVM model for Clinical Dataset

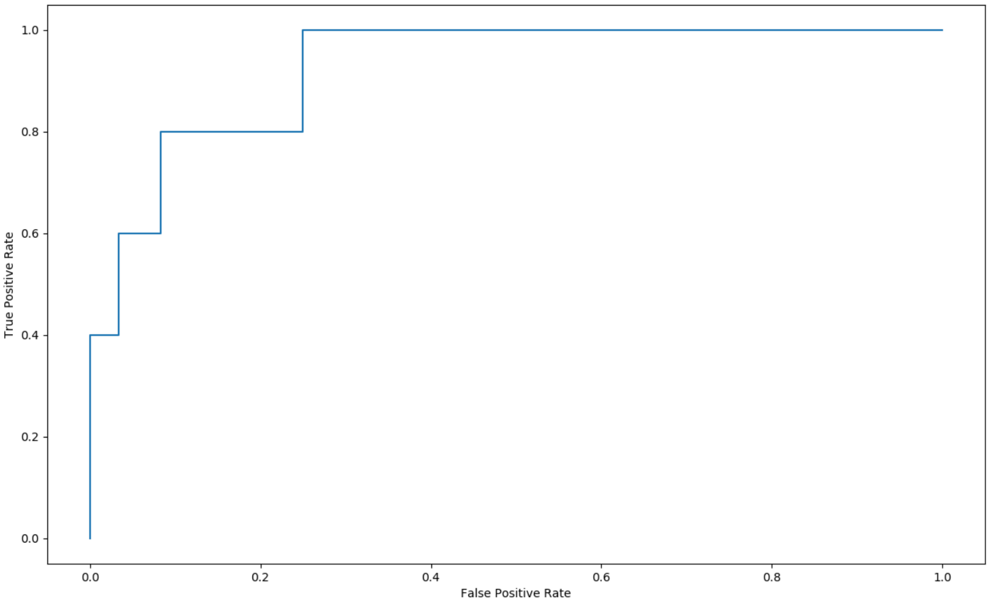

Figure 4.14: ROC curve of SVM model for HRQOL Dataset

On observing Table 4.3, Figure 4.10 and Figure 4.11 the SVM model was able to predict 3 out of 4 relevant documents for the clinical dataset (3/4) and 3 out of 5 relevant documents for the HRQOL dataset (3/5). Also, the model performed better than the baseline model (Logistic Regression) for both datasets.

4.3.3 Naive Bayes (Gaussian)

Table 4.4 gives the Gaussian Naive Bayes model evaluation for both Clinical and HRQOL dataset and, Figure 4.15 and Figure 4.16 displays the ROC curves for the same. As we can observe from the table, Gaussian Naive Bayes model was able to predict all relevant documents(4/4) for the clinical dataset and all relevant documents(5/5) for the HRQOL dataset. The model also performed better than our baseline model (Logistic Regression) for both datasets.

Table 4.4: Naive Bayes(Gaussian) Accuracy Metrics for Clinical and HRQOL datasets

Dataset	Sensitivity	Specificity	AUC	F1 Score
Clinical	1.0	0.64	0.85	0.65
HRQOL	1.0	0.7	0.9	0.72

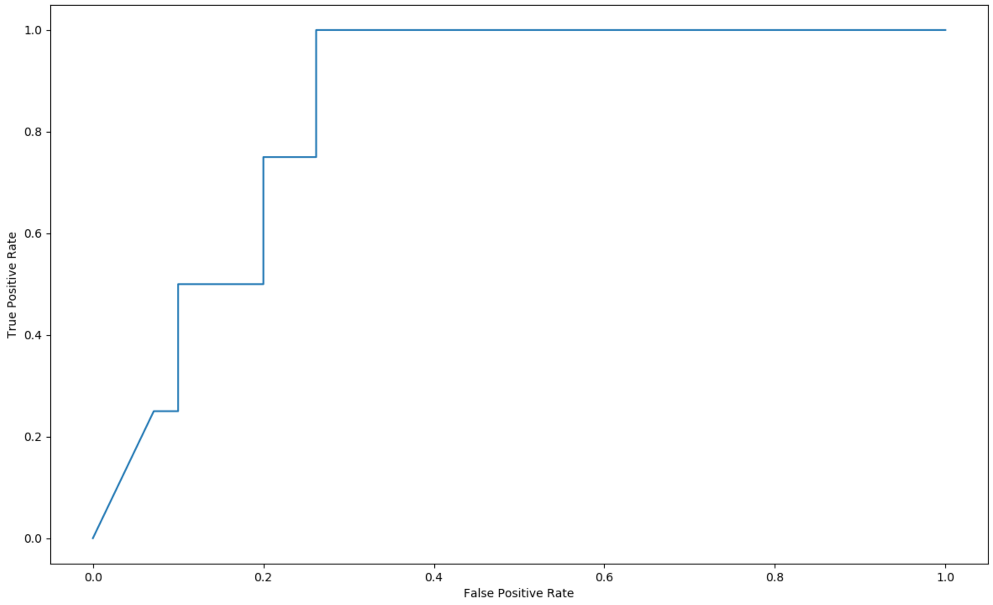

Figure 4.15: ROC curve of Naive Bayes(Gaussian) model for Clinical Dataset

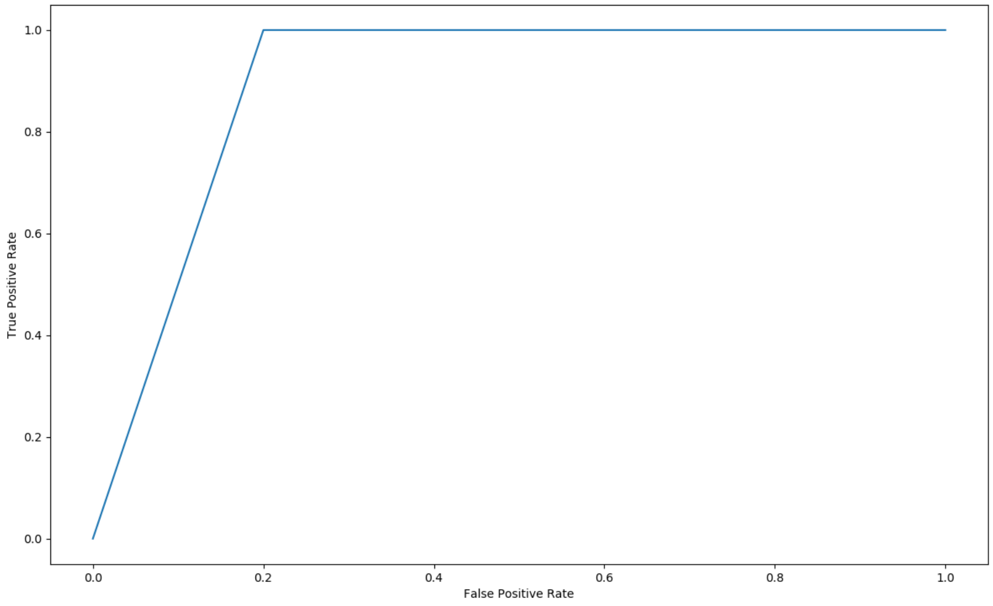

Figure 4.16: ROC curve of Naive Bayes(Gaussian) model for HRQOL Dataset

4.3.4 Bagged CART

Table 4.5 gives the Bagged CART evaluation for both Clinical and HRQOL dataset and, Figure 4.17 and Figure 4.18 displays the ROC curves for the same. Bagged CART model was able to predict 1 relevant document out of 4 relevant documents for the clinical dataset(1/4) and 2 out of 5 relevant documents for the HRQOL dataset (2/5). The model did not perform better than our baseline model (Logistic Regression) for the Clinical dataset and had the same sensitivity as the baseline model(Logistic Regression) for the HRQOL dataset.

Table 4.5: Bagged CART Accuracy Metrics for Clinical and HRQOL datasets

Dataset	Sensitivity	Specificity	AUC	F1 Score
Clinical	0.25	0.83	0.70	0.82
HRQOL	0.4	0.97	0.82	0.92

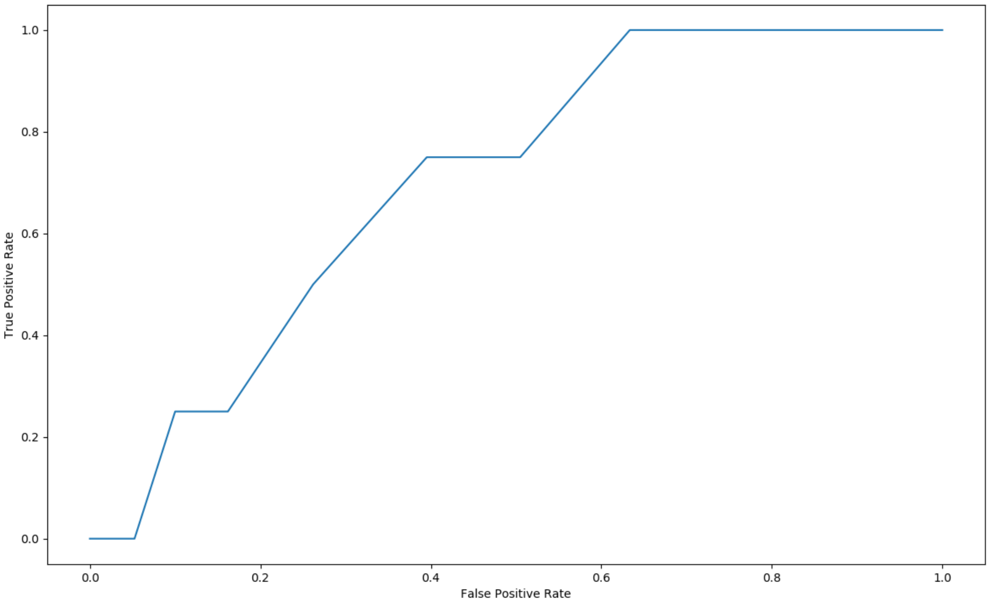

Figure 4.17: ROC curve of Bagged CART model for Clinical Dataset

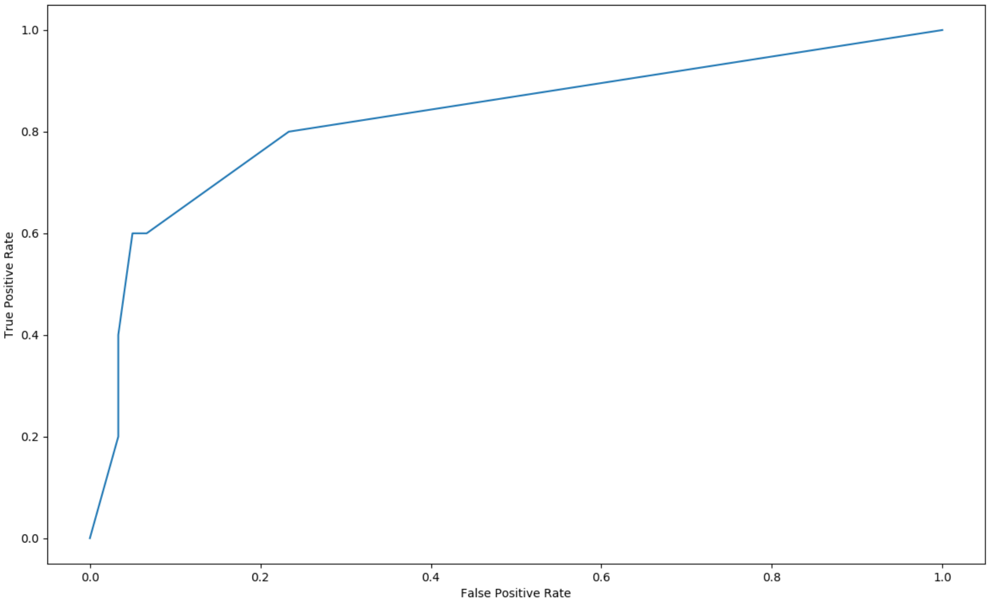

Figure 4.18: ROC curve of Bagged CART model for HRQOL Dataset

4.3.5 Comparison of the models

Figure 4.19 and Figure 4.20 gives the Sensitivity and Specificity comparison of our four models Clinical and HRQOL datasets respectively.

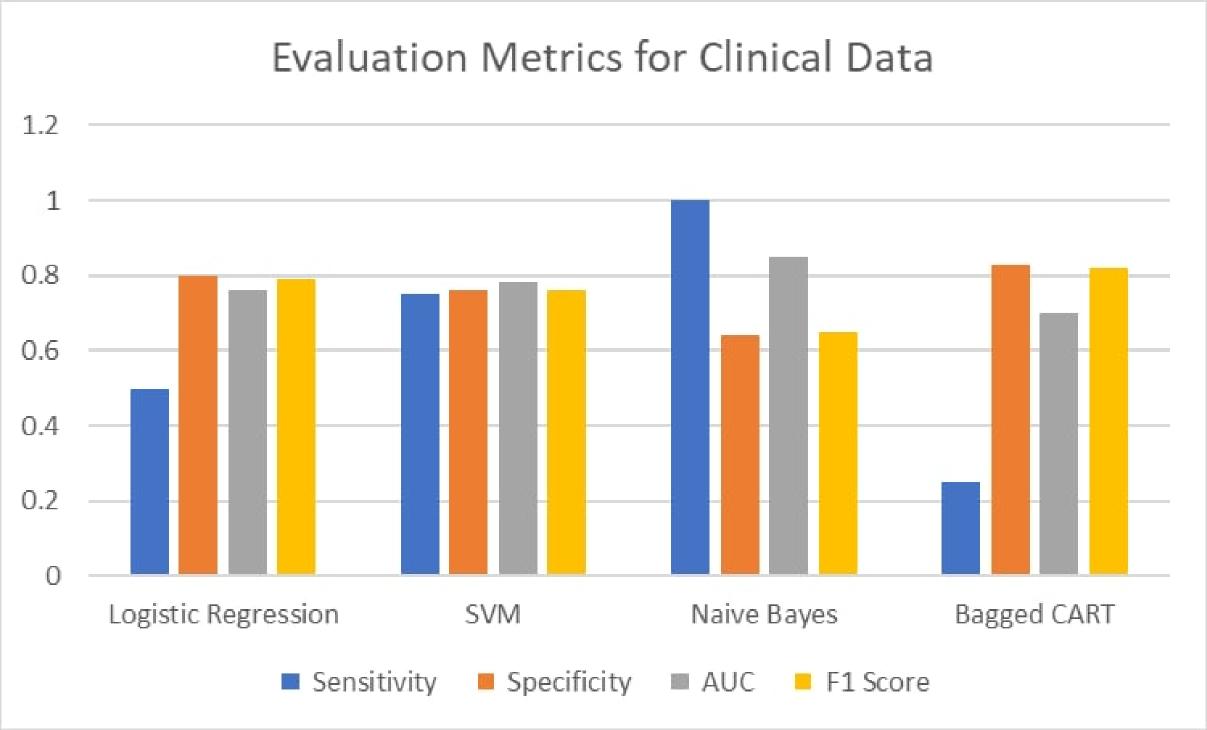

Figure 4.19: Metric Comparison for Clincal Dataset

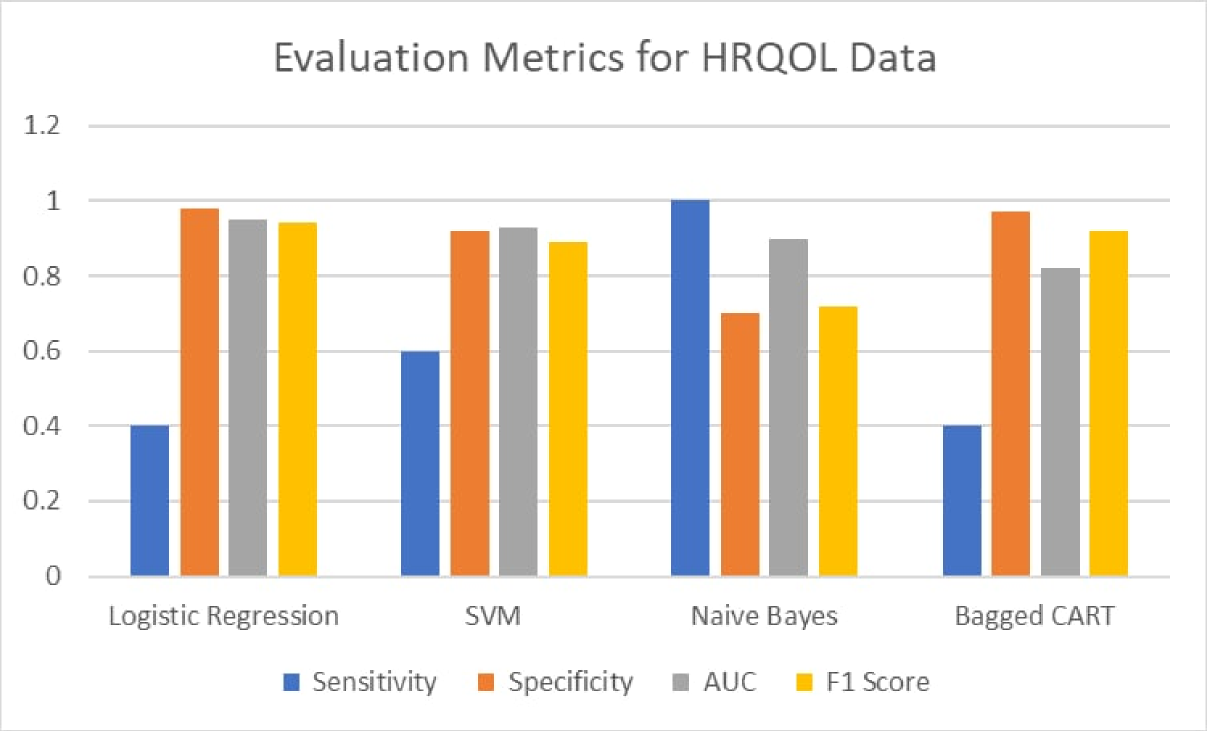

Figure 4.20: Metric Comparison for HRQOL Dataset

Overall, the best fitting model was found to be Naive Bayes (Gaussian) model with a sensitivity score of 1 for both clinical(4/4) and HRQOL (5/5) datasets. SVM model (with the best hyperparameter tuning) is second in performance while Bagged CART is found to be the least performing model.

4.4 Comparison with Similar work

As mentioned in the previous chapters, our analysis is similar to the one conducted in Popoff et al. (2020), where text mining techniques were also used to predict documents pertaining to the Systematic Literature review (in medical domain) using 4 datasets. In this section, we compare the results and observations between the 2 analyses.

Both Full text and abstract decisions were compared in Popoff et al. (2020), but our methodology was based on only abstract analysis. Exclusion and inclusion criteria were also predicted but this information was absent in our datasets.

As we can infer from Table 2 in Popoff et al. (2020), 5 different datasets were used, in which the rate of inclusions for Abstract ranged from 1.7% to 13.8%. The total number of abstracts in the datasets was as follows: Psoriasis(4422), Lung Cancer(12769), Liver Cancer(8507), Melanoma(3089) and Obesity(5187). In comparison, our datasets were limited with Clinical having 2132 documents (2.4% inclusion) and HRQOL having 257 documents (8.1% inclusion).

In Popoff et al. (2020), after 1:1 downsampling, the SVM model (with hypertuning) was found to be the best fitting model having a sensitivity of 100% in 3 of the 5 datasets. In comparison, Naive Bayes model performed better than SVM in our methodology having 100% sensitivity in both the datasets. Figure 4.21 is a snapshot from Popoff et al. (2020) (Figure 2, Page 6) displaying the statistics related to the methodology and metrics used.

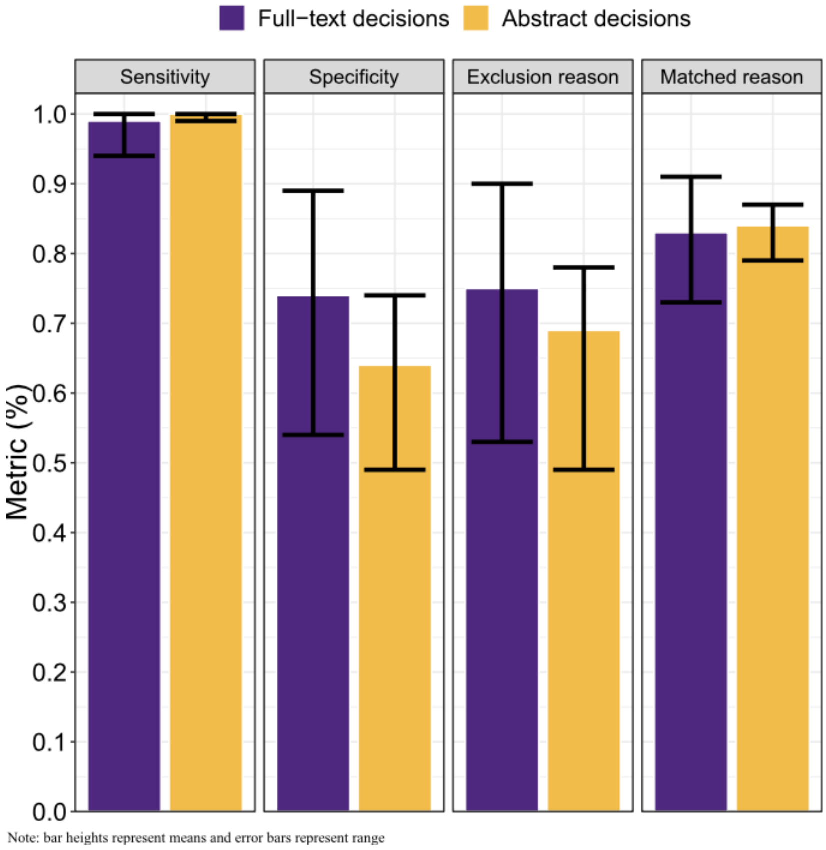

Figure 4.21: Automated SLR Performance analysis from Popoff et al. (2020)

In comparison with Figure 4.19 and Figure 4.20 of our analysis, we infer that even though the most optimal models were different, a similar level of sensitivity was achieved in both the analyses. Although, as can be inferred from Table 3 (Page 7) in Popoff et al. (2020) and Table 4.6 we can conclude that the precision levels are much better in Popoff et al. (2020) ranging from 10.84% to 20.76% across the 5 datasets. In comparison, precision in our analysis for the clinical dataset does not exceed 6%.

Table 4.6: Precision Metric of all models for Clinical Dataset

Model	Precision
Logistic Regression	0.045
SVM	0.057
Naive Bayes	0.051
Bagged CART	0.028

4.5 Limitations

As depicted in the Sections 4.4 and 4.5, while we were able to get high sensitivity in our analysis for the automated selection of documents for a systematic review process in the pharmaceutical domain, our methodology still has certain limitations. Some of these limitations are mentioned below.

4.5.1 Precision

While we were able to get high sensitivity, as expected precision did not exceed 6% (0.06) in our analysis. Table 4.6 gives an overview of the precision of the different models used in our analysis for the clinical dataset.

This was expected in our methodology since we are generating text features from a limited amount of documents in both our dataset as depicted in table 3.3 and 3.4. Ideally we would have wanted to achieve a precision of more than 30%, so as to ensure that the false inclusions (false positives) are also less.

4.5.2 Limited Dataset

The dataset that we used had a limited amount of documents for our training and testing data as displayed in Table 3.7 and Table 3.8. We believe a more comprehensive analysis could have been performed if we had a larger dataset pertaining to the Systematic Literature review that was performed.

As is expected in a systematic review process, the dataset was extremely unbalanced for inclusions. We were able to use downsampling to balance it, but it is possible that some relevant information might have been lost in the documents that were not included in the training set. This can also have impacted the classification accuracy of different models.

4.5.3 Reusability of trained models in different SLR

Reusability of our trained models is another factor in which our analysis has limitations. Since every new Systematic Review generates a new set of documents(using keywords search in databases such as PubMed), a new set of features would also need to be generated using the new corpus of documents.

Hence, a trained model on one systematic literature review is not expected to perform efficiently to predict documents pertaining to a different kind of systematic review. This is another limitation in just using text mining techniques for feature extraction for Systematic Reviews and might require certain alterations to make the strategy reusable across the pharmaceutical domain.

4.6 Security and Privacy Concerns

As mentioned in the previous chapter, since the datasets used were collated using publicly available databases such as PubMED, there are no security concerns related to our analysis. Also, the database used had no user's information and just publicly available pharmaceutical information, our analysis doesn't have any privacy implications as well.

Chapter 5

Conclusions & Future Work

5.1 Conclusion

This dissertation explored the concept of performing automated study selection for Systematic Literature Review using text mining techniques and machine learning models. While there are also similar works implemented for performing automated systematic literature review (Popoff et al. (2020)), we evaluated the feasibility of doing so on two datasets in a pharmaceutical context.

Abstract of each document was processed and tokenized prior to performing any text mining operations to ensure the best set of words to be considered as features. Since, both the datasets varied extremely as to corpus count, Training and Test data were split differently for Clinical and HRQOL (90%/10% for Clinical and 75% and 25% for HRQOL) to ensure that appropriate volume of data was present for training and generating features.

It is also important to observe that different strategies are used while generating the features primarily TF-IDF due to the difference in the corpus strength of the two datasets. Using features having weights closer to the variance helped improve the efficiency of our classification models for Clinical Dataset while features having the maximum weights worked much better for the subsequently less exhaustive HRQOL dataset. The notion of only using text features with maximum weight for classification did seem getting less significant as the corpus gets more exhaustive and the weights of the same set of words increase exponentially.

We also observed that downsampling was very effective in handling an extremely unbalanced dataset where the number of inclusions is very less as compared to the number of exclusions such as the Clinical Training Dataset (47:1871). While, there is an argument that important information might get lost due to significantly less training data, we can infer that the methodology using a random downsampling strategy should definitely be explored in the domain of text classification in a similar unbalanced scenario.

While most of our research suggested that SVM model is extremely prominent in the domain of text classification, we found Naive Bayes (Gaussian) to be the most accurate model in our analysis, achieving a sensitivity of 100% predicting 4/4 and 5/5 relevant documents in Clinical and HRQOL datasets respectively. As per our observations, Naive Bayes is a prominent modelling strategy and should definitely be considered for analysis in the text classification domain.

In comparison with similar work in the medical domain (Popoff et al. (2020)), the analysis was found to be similar but the best fitting model was different. SVM (with hyperparameters settings and 1:1 downsampling) was found to perform the best in their analysis achieving a sensitivity of 100% in 3 out of 5 datasets, whereas in our case Naive Bayes performed much better. We would also like to highlight that the datasets used in their analysis were much more exhaustive than the ones used in our methodology. Also, Precision evaluated in Popoff et al. (2020) is much better as compared to our analysis.

Considering both our analysis and the one depicted in Popoff et al. (2020), we can conclude that the usage of text mining and machine learning techniques for Systematic Literature Review does seem a promising strategy and should be explored. Even if a larger section of documents are recommended as compared to the manual Systematic Literature Review process, as long as all the relevant documents are present, the methodology can help expedite the overall process.

But one of the major hindrances as mentioned in Section 4.5.3 lies in the reusability of the trained model. Both our analysis and the one in Popoff et al. (2020) are using different sets of features as per different datasets and then classifying the test data documents. Reusing a set of features or models trained for a particular dataset in the systematic review of another dataset is not expected to be extremely efficient. To effectively use these strategies to expedite the Systematic review process, we would need an exhaustive methodology trained over a huge volume of corpus, that keeps track of the relevant set of documents for the keywords used to generate the same set of documents and use it to predict the documents for a new set of keywords.

5.2 Future Work

As mentioned in the previous section, while the strategy to use machine learning strategies in the Systematic Literature review does look promising, there is still a significant room for growth. There are two major areas which can be improved namely feature generation(including dataset related issues) and reusability. We list certain key aspects which can be used or improved upon below in the same domain.

5.2.1 Using Recursive Feature Elimination (RFE) with TF-IDF

As observed in our methodology, while we used TF-IDF to generate the features, the methodology differed in selection of features based on weights depending on the dataset. Ideally, a common feature generation strategy should be used to perform the analysis. Recursive Feature Elimination (Nafis and Awang (2021), Youn and Jeong (2009)) with TF-IDF provides a much more scalable alternative where an ideal document matrix is generated by iteratively first filtering features that have total weights less than the variance of the document matrix and then recomputing the document matrix again. This strategy could be used in any dataset to generate an optimal set of features with the maximum accuracy and make the analysis much more scalable.

5.2.2 Using Doc2Vec for feature generation

Another feature generation strategy that can be evaluated is the Doc2Vec Le and Mikolov (2014) modelling. In this strategy, neural networks are used to generate vector representations of documents and by extension for sentences included in them. Every document is represented by a dense vector which is used for predictions. An optimal set of features could be computed by using the particular dataset and hence a common process will be followed for the feature generation aspect of the analysis.

5.2.3 Exhaustive Dataset to improve precision

The Clinical dataset(document count: 2132) and HRQOL dataset(document count: 257) used in our analysis were fairly limited. Even though we were able to achieve an optimal Sensitivity level, on the precision front we could not exceed 6%. This meant a significant amount of documents were predicted as inclusions even though they were not. While it is not a major metric for our analysis as discussed in section 3.3.4, there is a definite room for improvement here which can be achieved by having a much larger feature set created from a much more exhaustive dataset.

5.2.4 Pool based Active Learning

Pool based Active learning (Tong and Koller (2001)) is an optimal unsupervised learning strategy that can be considered to improve reusability. The three key components of the active learner used in this strategy are classifier, labelled dataset and a query function. A classifier is first trained on a particular amount of data using the labelled dataset. The query function then evaluates the instances that are to be used for the next query. After each query, the active learner returns a classifier pertaining to the query. This process can help track the relevant documents based on the keywords(query) in our analysis and thus help in reusing the same model in different Systematic Literature Reviews.

Bibliography

Adeva, J. G., Atxa, J. P., Carrillo, M. U., and Zengotitabengoa, E. A. (2014). Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications*, 41(4):1498–1508.

Alahmadi, A., Joorabchi, A., and Mahdi, A. E. (2013). A new text representation scheme combining bag-of-words and bag-of-concepts approaches for automatic text classification. In *2013 7th IEEE GCC Conference and Exhibition (GCC)*, pages 108–113. IEEE.

Anand, A., Pugalenthi, G., Fogel, G. B., and Suganthan, P. (2010). An approach for classification of highly imbalanced data using weighting and undersampling. *Amino acids*, 39(5):1385–1391.

Ash, Z., Gaujoux-Viala, C., Gossec, L., Hensor, E. M., FitzGerald, O., Winthrop, K., Van Der Heijde, D., Emery, P., Smolen, J. S., and Marzo-Ortega, H. (2012). A systematic literature review of drug therapies for the treatment of psoriatic arthritis: current evidence and meta-analysis informing the eular recommendations for the management of psoriatic arthritis. *Annals of the rheumatic diseases*, 71(3):319–326.

Ashing-Giwa, K. T. (2005). The contextual model of hrqol: A paradigm for expanding the hrqol framework. *Quality of Life Research*, 14(2):297–307.

Cohen, A. M. (2006). An effective general purpose approach for automated biomedical document classification. In *AMIA annual symposium proceedings*, volume 2006, page 161. American Medical Informatics Association.

Colas, F. and Brazdil, P. (2006). Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Dong, Q.-W., Wang, X.-l., and Lin, L. (2006). Application of latent semantic analysis to protein remote homology detection. *Bioinformatics*, 22(3):285–290.

Dzisevič, R. and Šešok, D. (2019). Text classification using different feature extraction approaches. In *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pages 1–4. IEEE.

Fink, A. (2019). *Conducting research literature reviews: From the internet to paper*. Sage publications.

Frunza, O., Inkpen, D., and Matwin, S. (2010). Building systematic reviews using automatic text classification techniques. In *Coling 2010: Posters*, pages 303–311.

Gabrilovich, E. and Markovitch, S. (2005). Feature generation for text categorization using world knowledge. In *IJCAI*, volume 5, pages 1048–1053.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Hassan, S., Rafi, M., and Shaikh, M. S. (2011). Comparing svm and naïve bayes classifiers for text categorization with wikitology as knowledge enrichment. In *2011 IEEE 14th International Multitopic Conference*, pages 31–34. IEEE.

Higgins, J. P., Thomas, J., Chandler, J., Cumpston, M., Li, T., Page, M. J., and Welch, V. A. (2019). *Cochrane handbook for systematic reviews of interventions*. John Wiley & Sons.

Ibrahim, Y., Okafor, E., Yahaya, B., Yusuf, S. M., Abubakar, Z. M., and Bagaye, U. Y. (2021). Comparative study of ensemble learning techniques for text classification. In *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, pages 1–5. IEEE.

Joshi, S. and Abdelfattah, E. (2021). Multi-class text classification using machine learning models for online drug reviews. In *2021 IEEE World AI IoT Congress (AIIoT)*, pages 0262–0267. IEEE.

Kadhim, A. I. (2019). Term weighting for feature extraction on twitter: A comparison between bm25 and tf-idf. In *2019 international conference on advanced science and engineering (ICOASE)*, pages 124–128. IEEE.

Kumar, V. and Subba, B. (2020). A tfidfvectorizer and svm based sentiment analysis framework for text data corpus. In *2020 National Conference on Communications (NCC)*, pages 1–6. IEEE.

Lalkhen, A. G. and McCluskey, A. (2008). Clinical tests: sensitivity and specificity. *Continuing education in anaesthesia critical care & pain*, 8(6):221–223.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Liu, C.-z., Sheng, Y.-x., Wei, Z.-q., and Yang, Y.-Q. (2018). Research of text classification based on improved tf-idf algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, pages 218–222. IEEE.

Liu, Z., Lv, X., Liu, K., and Shi, S. (2010). Study on svm compared with the other text classification methods. In *2010 Second international workshop on education technology and computer science*, volume 1, pages 219–222. IEEE.

Maharaj, R., Raffaele, I., and Wendon, J. (2015). Rapid response systems: a systematic review and meta-analysis. *Critical Care*, 19(1):1–15.

Mulrow, C. D., Cook, D. J., and Davidoff, F. (1997). Systematic reviews: critical links in the great chain of evidence. *Annals of internal medicine*, 126(5):389–391.

Nafis, N. S. M. and Awang, S. (2021). An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification. *IEEE Access*, 9:52177–52192.

Nam, J. L., Ramiro, S., Gaujoux-Viala, C., Takase, K., Leon-Garcia, M., Emery, P., Gossec, L., Landewe, R., Smolen, J. S., and Buch, M. H. (2014). Efficacy of biological disease-modifying antirheumatic drugs: a systematic literature review informing the 2013 update of the eular recommendations for the management of rheumatoid arthritis. *Annals of the rheumatic diseases*, 73(3):516–528.

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., and Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews*, 4(1):1–22.

Popoff, E., Besada, M., Jansen, J., Cope, S., and Kanters, S. (2020). Aligning text mining and machine learning algorithms with best practices for study selection in systematic literature reviews. *Systematic reviews*, 9(1):1–12.

Prusa, J. D. and Khoshgoftaar, T. M. (2016). Designing a better data representation for deep neural networks and text classification. In *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, pages 411–416.

Schleidgen, S., Klingler, C., Bertram, T., Rogowski, W. H., and Marckmann, G. (2013). What is personalized medicine: sharpening a vague term based on a systematic literature review. *BMC medical ethics*, 14(1):1–12.

Shah, K., Patel, H., Sanghvi, D., and Shah, M. (2020). A comparative analysis of logistic regression, random forest and knn models for the text classification. *Augmented Human Research*, 5(1):1–16.

Shojania, K. G., Sampson, M., Ansari, M. T., Ji, J., Doucette, S., and Moher, D. (2007). How quickly do systematic reviews go out of date? a survival analysis. *Annals of internal medicine*, 147(4):224–233.

Stros, M. and Lee, N. (2015). Marketing dimensions in the prescription pharmaceutical industry: a systematic literature review. *Journal of Strategic Marketing*, 23(4):318–336.

Sun, A., Lim, E.-P., and Liu, Y. (2009). On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, 48(1):191–201.

Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4):667–671.

Tong, S. and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Youn, E. and Jeong, M. K. (2009). Class dependent feature scaling method using naive bayes classifier for text datamining. *Pattern Recognition Letters*, 30(5):477–485.

Zareapoor, M., Shamsolmoali, P., et al. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia computer science*, 48(2015):679–685.

Zhang, W. and Gao, F. (2011). An improvement to naive bayes for text classification. *Procedia Engineering*, 15:2160–2164.

Appendix

Table 1: SLR strategy for Clinical Dataset

Criteria	Inclusion	Exclusion
Population	Patients with relapsed or refractory B-cell ALL up to 25 years of age,	Patients aged >25 years of age
	any sex, any ethnicity	Treatment naïve patients
		Patients with T-cell ALL
tervention	Tisagenlecleucel used as monotherapy	CD19 CAR T-cell therapy used in combination therapy
		Clofarabine
		Inotuzumab
Comparator	Blinatumomab (with or without SCT)	Standard of care not otherwise defined
	FLA-IDA (fludarabine, cytarabine, idarubicin) (with or without SCT)	
	SCT (salvage regimen not specified)	
	Placebo	
	Any of the included interventions	
Outcome	Primary outcomes:	Response rates:
	Survival outcomes:	Objective response rate
	Overall survival	Duration of response
	Progression-free survival	Complete response
	Event-free survival	Partial response
	Leukaemia-free survival	
	Data extracted, but not an outcome if study only reported on these:	Pharmacokinetic/pharmacodynamic outcomes
	HRQOL	Social outcomes
	AEs	
Study type	Prospective randomised controlled trials	Single-centre trials
	Phase II non-randomised or uncontrolled trials	Retrospective studies
	Prospective observational studies	Reviews
	Patient registries	Letters
		Comments
		Editorials
		Case studies/reports
		Narrative publications
		Biomarker/prognostic studies
		Conference abstracts without full text
		Expanded access programmes
		Indirect treatment comparisons

Table 2: SLR strategy for HRQOL dataset

Criteria	**Inclusion**	**Exclusion**
Population	Patients with relapsed or refractory ALL up to 25 years of age,	Treatment naïve patients
	any sex, any ethnicity	
Intervention	Tisagenlecleucel	
Comparator	Blinatumomab (with or without SCT)	Not licensed in Europe
	FLA-IDA (fludarabine, cytarabine,	
	idarubicin) (with or without SCT)	
	Any licensed therapy in Europe (for R/R ALL)	
	Placebo	
	Any of the included interventions	
Outcome	Outcomes were required to be reported as a utility value; a format that allowed use as an input parameter in the bespoke cost-effectiveness model	
	1. Health-state utility values for event-free survival and progressed disease	
	2. Health-state utility values associated with long-term survival	
	3. Disutility values associated with treatment and associated administration or hospitalisation	
	4. Disutility values associated with short-term adverse events of treatment (less than eight weeks, as per the EPAR of tisagenlecleucel (20))	
	5. Disutility values associated with long-term (greater than eight weeks) adverse events of treatment	
	6. Disutility values associated with alloSCT (autoSCT also considered)	
Study type	Any study providing the required outcome was included	case studies
		studies providing data on a single patient