A project report on

Aspect Extraction for evaluating hospitals and its visualization

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

DEEPSI KUMARI(19BCE1064)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

Aspect Extraction for evaluating hospitals and its visualization

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

DEEPSI KUMARI(19BCE1064)

VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

DECLARATION

I here by declare that the thesis entitled **ASPECT EXTRACTION FOR EVALUATING HOSPITALS AND ITS VISUALIZATION** submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai, is a record of bonafide work carried out by me under the supervision of **Dr. Sathis Kumar B.**

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date:22/04/2023 Signature of the Candidate

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **“ASPECT EXTRACTION FOR EVALUATING HOSPITALS AND ITS VISUALIZATION”** is prepared and submitted by **Deepsi Kumari** (**19BCE1064**) to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** programme is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr. Sathis Kumar B.

Date:

Signature of the Examiner 1
Name:
Date:

Signature of the Examiner 2
Name:
Date:

Approved by the Head of Department
B. Tech. CSE

Name: Dr. Nithyanandam P
Date: 24 – 04 – 2023

(Seal of SCOPE)

ABSTRACT

Healthcare consumers nowadays rely on various online resources, such as hospital review sites like Yelp and Mouthshut, as well as social media, to make informed decisions when selecting a hospital or clinic for consultation. This paper investigates the use of opinion mining on textual data extracted from hospital review sites, and proposes the use of data visualization tools like PowerBI or Tableau to aid patients in identifying good hospitals easily. The visualization techniques presented in this paper include faceted and filtered visualization, which allow for a more advanced analysis of the sentiment expressed in the reviews. Our evaluation of the proposed techniques shows a high level of accuracy in opinion mining and detection of pain points, which are crucial for understanding the patients' sentiments and reasons for changes in their opinions. Moreover, the prototype developed in this study can help identify both the strengths and weaknesses of hospitals' hospitality services.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Sathis Kumar B, Designation, SCOPE, Vellore Institute of Technology, Chennai, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Machine Learning.

It is with gratitude that I would like to extend thanks to our honorable Chancellor, Dr. G. Viswanathan, Vice Presidents, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan and Mr. G V Selvam, Assistant Vice-President, Ms. Kadhambari S. Viswanathan, Vice-Chancellor, Dr. Rambabu Kodali, Pro-Vice Chancellor, Dr. V. S. Kanchana Bhaaskaran and Additional Registrar, Dr. P.K.Manoharan for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dean, Dr. Ganesan R, Associate Dean Academics, Dr. Parvathi R and Associate Dean Research, Dr. Geetha S, SCOPE, Vellore Institute of Technology, Chennai, for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Nithyanandam P, Head of the Department, Project Coordinators, Dr. Abdul Quadir Md, Dr. Priyadarshini R and Dr. Padmavathy T V, B. Tech. Computer Science and Engineering, SCOPE, Vellore Institute of Technology, Chennai, for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Chennai, who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date:22/04/2023 Name of the student

DEEPSI KUMARI

CONTENTS

CHAPTER 6

CHAPTER 7

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BoW	Bag of Words
Tf-idf	Term frequency-inverse document frequency
NLP	Natural Language Processing
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
NMF	Nonnegative Matrix Factorization

Chapter 1

Introduction

We live in a Golden Age of Information where, with product information and reviews available across several outlets, making informed choices is a piece of cake.. The contents on such reviews websites is user-generated, thus giving access to the opinions of many individuals. When contributing opinions to the web- sites, users typically select grades for a a number of facets, and additionally add a textual review. During subsequent search of hospitals,users get a ranked list of hospitals, where ranking is based on the grades given by the patients or their relatives.

When studying existing websites,social medias and previous research in the area, two observations can be made: 1) the visualization of the hospitals is not quite primitive that shows the declining or rising reviews trend or other benefits of hospitals with the help of charts but just shows the direction on google maps, and

2) the only use of the textual descriptions is for browsing, they are not part of the ranking process or visualized. To our knowledge, these issues have not been studied before. In this paper we also describe how to use opinion mining techniques to analyze changes in opinions about hospitals and gathering pain points. Hospitals are seen as a public-based service is an area where multiple factors may impact patients or their relatives sentiment. For instance service, doctor’s punctuality, cleanliness, staffs behavior and more such events may influence the overall sentiment at any given time, creating a dynamically changing sentiment. Managing to identify why changes occur in such a setting, may provide both patients and hospital administration some valuable information regarding the interpretation of large amounts of opinionated data.

Opinion mining tools are used to identify and extract subjective information from user reviews, and then to determine the sentiment of the text. Two different techniques are studied: feature extraction and visualization. Feature extraction is a technique to identify and extract product features and extract the pain points and visualization means to present the information graphically. Evaluation is performed by comparing the actual review scores with our sentiment scores.

In order to perform visualization experiments, a web prototype was created. This provides a way to detect "good" and "bad" aspects based on the hospital reviews in a user- friendly interface. These scores are calculated based on user opinions, and is an effective way for users to filter sentiment data. For commercial use, the prototype can help analyze the massive amount of hospital information published each day by users, and can help hospital administration analyze their services. It can also be used as a more advanced hospital search engine where users can find extra information in a map userinterface that can serve as a benefit for the people who don’t want to read long passages.

1.1 GENERAL BACKGROUND AND MOTIVATION

1.1.1 BACKGROUND

The healthcare industry is increasingly focused on providing patient-centered care, which involves understanding patients' needs and preferences. Online reviews are a valuable source of information for healthcare organizations seeking to gain insights into patient satisfaction and preferences. However, analyzing large volumes of reviews can be a time-consuming and resource-intensive process.

To address this challenge, natural language processing techniques such as LDA, LSA, NMF, and bag-of-words models have been developed to analyze reviews and extract key aspects. These techniques have been successfully applied in various industries, including healthcare. By using these techniques, healthcare organizations can gain valuable insights into patient satisfaction and preferences, identify areas for improvement, and make data-driven decisions to improve the quality of care.

Furthermore, visualization tools like Power BI can be used to present the results of the analysis in a clear and concise manner. This can help healthcare organizations make informed decisions based on the insights gained from the reviews.

Given the importance of patient-centered care in the healthcare industry, the use of natural language processing techniques and visualization tools to analyze hospital reviews and ratings is becoming increasingly important. By using these techniques, healthcare organizations can better understand patient needs and preferences, improve the quality of care provided, and ultimately enhance patient outcomes.

1.1.2 MOTIVATION

The use of AI and natural language processing techniques to analyze hospital reviews can provide a number of benefits for healthcare organizations. For example, analyzing reviews can help hospitals identify areas where they excel and areas where they need to improve, ultimately leading to increased patient satisfaction. Additionally, analyzing large volumes of data can be a time-consuming and resource-intensive process, but AI can automate this task and provide insights more efficiently.

Another advantage of analyzing hospital reviews is that it can help hospitals stay competitive by providing a way to compare their performance to that of other healthcare organizations in the same region or with similar services. This can help hospitals make more informed decisions about how to allocate resources and focus their efforts.

Moreover, analyzing hospital reviews can lead to more personalized care for patients. By understanding patients' needs and preferences, hospitals can tailor their services to better meet those needs, ultimately leading to better patient outcomes.

Finally, the insights gained from analyzing hospital reviews can inform decision-making at all levels of the hospital, from operations and service delivery to marketing and branding. This can help hospitals improve their overall performance and better meet the needs of their patients.

In conclusion, the use of AI and natural language processing techniques to analyze hospital reviews has the potential to provide valuable insights to healthcare organizations. By leveraging these technologies, hospitals can improve patient satisfaction, stay competitive, deliver more personalized care, and make more informed decisions.

1.2 PROBLEM STATEMENT

In today's world, consumers have access to a wealth of information about products and services, including online reviews from other customers. However, with this abundance of information comes the challenge of sorting through and understanding it all. This can be particularly difficult in the healthcare industry, where hospital reviews and ratings are important sources of information for both patients and healthcare organizations.

To address this challenge, this project aims to develop a model that uses natural language processing techniques to analyze hospital reviews and extract key aspects. The model will be powered entirely by publicly available data, making it accessible to a wide range of users seeking information about hospital reviews and ratings.

The visualization process developed in this project is intended to help patients make informed decisions when choosing a hospital for their medical procedure, as well as provide hospital administrators with insights into areas for improvement. By using this model and visualization process, healthcare organizations can gain valuable insights into patient satisfaction and preferences, and make data-driven decisions to improve the quality of care provided.

Chapter 2

LITERATURE REVIEW

A literature review was conducted to examine the various applications of sentiment analysis in different industries. The selected references covered the period from 2013 to 2022 and highlighted the diverse approaches, methodologies, and findings of the studies. The review showed that sentiment analysis has become a valuable tool for extracting insights from large amounts of unstructured data and can be applied in various fields, including healthcare, social media, customer service, and marketing.

Cobb et al. (2013) conducted sentiment analysis to evaluate the impact of online messages on smokers' choices to use varenicline. The study revealed that negative messages about the drug influenced the smokers to avoid using it. Ebrahimi et al. (2016) identified the recognition of side effects as implicit-opinion words in drug reviews. They used sentiment analysis to determine the sentiment of the reviewers towards specific side effects, which can aid in improving drug safety. Rastegar-Mojarad et al. (2015) analyzed patient experiences of healthcare from social media using sentiment analysis. The study found that social media platforms can be used to collect and analyze patient experiences of healthcare, which can help in improving healthcare services.

Abirami and Askarunisa (2017) used sentiment analysis to emphasize the impact of online reviews in the healthcare industry. The study found that positive reviews significantly influenced patients' decisions to visit healthcare facilities. Zakkar and Lizotte (2021) analyzed patient stories on social media using text analytics. The study found that text analytics can be used to extract valuable information from patient stories on social media, which can aid in improving healthcare services.

In the customer service and marketing domain, Kelly Aponté and Shiseida Sade (2020) reported that customer pain points can be identified and addressed through sentiment analysis of customer feedback. They emphasized the importance of addressing customer pain points to enhance customer satisfaction and loyalty. Tao et al. (2019) used sentiment analysis to mine pain points from hotel online comments. The study revealed that sentiment analysis can be used to identify the areas of concern for customers, which can help in improving the quality of services.

Tammina and Annareddy (2020) applied convolutional neural network (CNN) for sentiment analysis of customer reviews. The study found that CNN outperformed traditional machine learning algorithms in sentiment classification. Salminen et al. (2022) developed a machine learning model to detect pain points from user-generated social media posts. The study found that the model achieved high accuracy in identifying pain points, which can aid in improving customer satisfaction.

Ahmad et al. (2019) used sentiment analysis techniques to detect and classify social media-based extremist affiliations. The study found that sentiment analysis can be used to identify extremist affiliations and prevent radicalization.

Praphula Kumar Jain et al. (2021) conducted a systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. The study found that machine learning algorithms have been extensively used for consumer sentiment analysis and have shown promising results.

In conclusion, sentiment analysis has become an important tool for extracting valuable insights from unstructured data in different domains. The studies reviewed in this literature review demonstrated the effectiveness of sentiment analysis in healthcare, social media, customer service, marketing, and security. The findings of these studies can aid in improving the quality of services and enhancing customer satisfaction and loyalty. Future research can focus on developing more advanced sentiment analysis techniques to extract deeper insights from unstructured data.

Chapter 3

PROPOSED WORK

Aspect-based sentiment analysis is a powerful tool that can be applied to various domains, including product reviews, social media sentiment analysis, and customer feedback analysis. It allows for a more granular analysis of feedback by identifying specific aspects of a product or service that are driving the overall sentiment.

In the context of hospital reviews, aspect extraction can help hospital administrators and healthcare providers to identify areas where they are performing well and areas that need improvement. For example, if the aspect "staff behavior" has a negative sentiment score, it can indicate that patients are dissatisfied with the behavior of the hospital staff. This feedback can be used to improve staff training programs or to address specific issues with staff behavior.

The proposed method is done so as to create an aspect based extraction of hospital reviews system design by implementing the analysis of hospital reviews input in which we preprocess the reviews by lemmatizing and other methods.These reviews are then observed based on various aspects and what is the aspect score for them and sentiment too by using various algorithms like LDA,LSA,etc.

The dataset used is the custom dataset that is made by scraping different websites using scraping tool.All the datasets scraped from different websites are combined into one and hence then the aspect extraction is done. Figure 3.1 depicts the architecture on which the prototype has been developed.

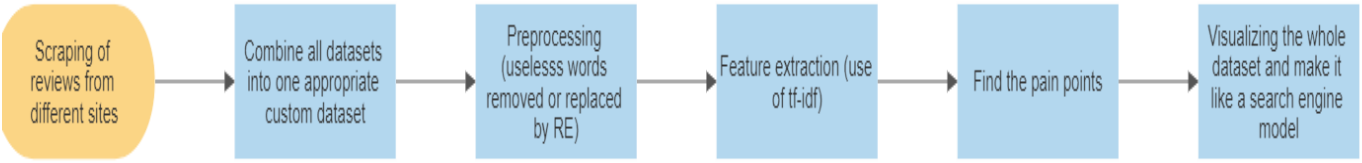

Scraping of reviews from different sites
Combine all datasets into one appropriate custom dataset
Preprocessing (uselesss words removed or replaced by RE)
Feature extraction (use of tf-idf)
Find the pain points
Visualizing the whole dataset and make it like a search engine model

Fig 3.1 Basic Architecture Diagram

3.1 DATASET DESCRIPTION

A dataset is a collection of data that is organized and stored in a structured format. In the context of machine learning, a dataset refers to a collection of data that is used to train and evaluate machine learning models. The quality of the dataset is critical to the performance of the machine learning model. A large dataset of hospital reviews is collected containing ratings and reviews by scraping online platforms such as Yelp, Google, Healthgrades, etc.All the datasets are combined into one dataset for a large amount of reviews and further experiment. As the datset scraped from different websites contains stop words,punctuations and other regular expression,and other noise,so the dataset is preprocessed

by removing irrelevant information, data redundancy, and performing text normalization techniques such as removing punctuation marks, stop-word removal, tokenization, stemming/lemmatization.

	Reviews	Rating	Company	
0	Felt that Dr Harsh dua is devdoot to his patient	5	Apollo	

1	Arrogant doctors in dental section	4	Apollo	
2	My next door friendly hospital	4	Apollo	
3	Review about the hospital	5	Apollo	
4	Review of the hospital	5	Apollo	
5	Delhi best hospital	1	Apollo	
6	Dr. Sujit kumar chowdhary	1	Apollo	
7	Worst experience	1	Apollo	
8	Bad experience	1	Apollo	
9	Worst hospital...	1	Apollo	
10	Worst	1	Apollo	
11	Doctors don't explain	5	Apollo	
12	Best Hospital in Delhi NCR	1	Apollo	
13	Best Hospital for Cancer Treatment in Delhi, India	1	Apollo	
14	Worst hospital	1	Apollo	
15	NO VALUE FOR MONEY USED FOR TREATMENT	1	Apollo	
16	Money Haunted. Not professionally groomed.	1	Apollo	
17	Apolo Hospital is very good	1	Apollo	
18	Apollo hospital indraprath delhi	1	Apollo	
19	Comment by an International Patient	1	Apollo	
20	Excellent results	1	Apollo	
21	Doctors are money hungry	1	Apollo	
22	Very dangerous	1	Apollo	
23	Not worth it	1	Apollo	
24	Apollo Delhi is one of the worst hospitals	4	Apollo	
25	Rooms	1	Apollo	
26	Very Delayed Service	1	Apollo	
27	Icu experience in Apollo Delhi	1	Apollo	
28	Big Hospital -Big Names-BIG MISDIAGNOSIS	1	Apollo	
29	Not so good in service	1	Apollo	
30	Dengue case	1	Apollo	

TABLE 3.1.1. Dataset sample

3.2 SENTIMENT ANALYSIS

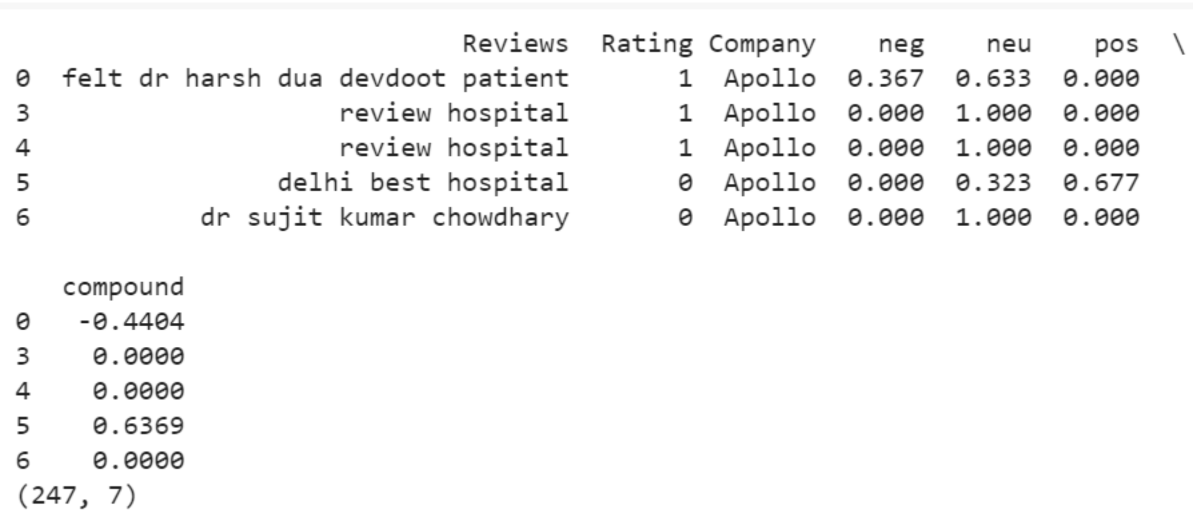
Sentiment analysis is a natural language processing technique that involves identifying and extracting subjective information, such as opinions and emotions, from text data. In the context of healthcare, sentiment analysis can be used to analyze patient feedback from hospital reviews and determine whether the sentiment is positive, negative, or neutral.

In your research project, sentiment analysis can be used alongside other natural language processing techniques like LDA, LSA, NMF, and bag-of-words models to extract key aspects from hospital reviews and evaluate the sentiment associated with each aspect. This can help healthcare organizations gain insights into patient satisfaction and preferences and prioritize areas for improvement.

One of the main advantages of sentiment analysis is that it provides a quick and efficient way of analyzing large volumes of text data. By using sentiment analysis to analyze hospital reviews, healthcare organizations can quickly identify common themes and sentiment associated with each theme. This can help them make data-driven decisions to improve the quality of care provided.

Another advantage of sentiment analysis is that it can help healthcare organizations track changes in patient sentiment over time. By monitoring changes in sentiment associated with specific aspects of care, healthcare organizations can evaluate the effectiveness of interventions designed to improve patient satisfaction.

Overall, sentiment analysis is a valuable tool for healthcare organizations seeking to gain insights into patient sentiment and preferences. By using sentiment analysis alongside other natural language processing techniques, healthcare organizations can develop a comprehensive understanding of patient feedback, identify areas for improvement, and make data-driven decisions to improve the quality of care provided.



```
                         Reviews  Rating Company    neg    neu    pos  \
0  felt dr harsh dua devdoot patient       1  Apollo  0.367  0.633  0.000
3                    review hospital       1  Apollo  0.000  1.000  0.000
4                    review hospital       1  Apollo  0.000  1.000  0.000
5                delhi best hospital       0  Apollo  0.000  0.323  0.677
6          dr sujit kumar chowdhary       0  Apollo  0.000  1.000  0.000

   compound
0   -0.4404
3    0.0000
4    0.0000
5    0.6369
6    0.0000
(247, 7)
```

Fig 3.2.1. Sentiment Scores

3.3 ROADMAP FOR PROPOSED MODEL

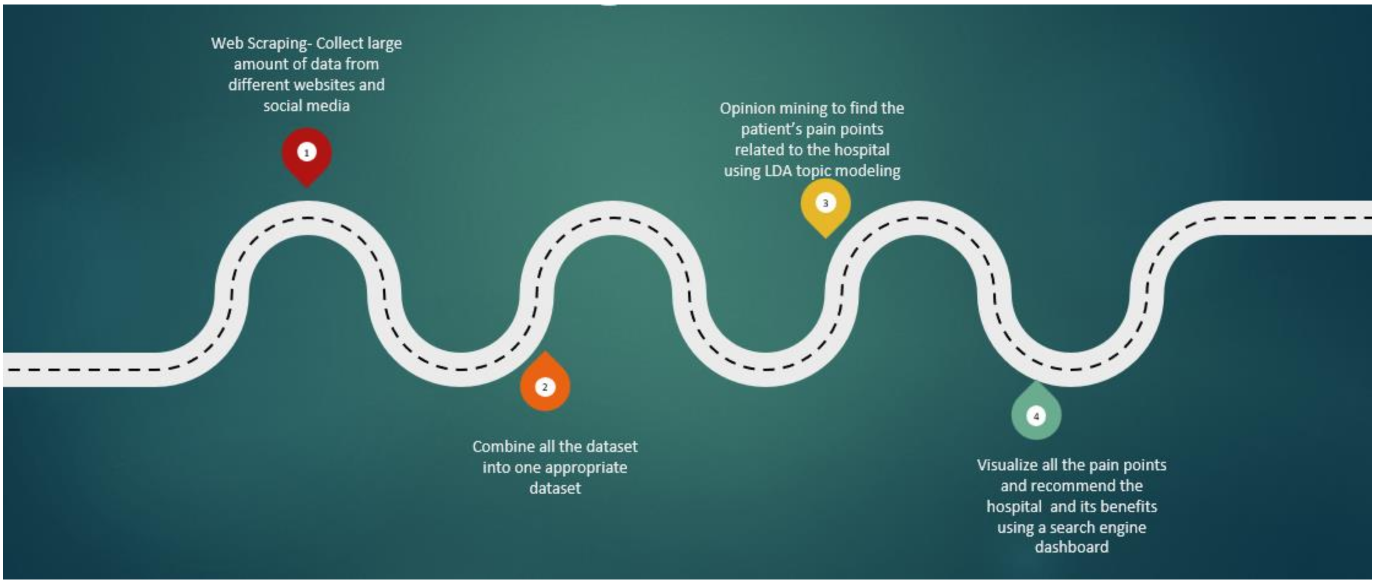

Web Scraping- Collect large amount of data from different websites and social media
1
Opinion mining to find the patient's pain points related to the hospital using LDA topic modeling
3
2
Combine all the dataset into one appropriate dataset
4
Visualize all the pain points and recommend the hospital and its benefits using a search engine dashboard

Fig 3.3 Roadmap of Proposed Model

The proposed system is designed to utilize opinion mining tools for the purpose of identifying and extracting subjective information from user reviews. The objective of the system is to determine the sentiment of the text and the pain points of users, which are mainly the aspects. To accomplish this, the system requires a large collection of user reviews of hospitals as input. The primary goal is to identify the most significant aspects of each hospital.

The system will be implemented using various machine learning algorithms and techniques such as Bag of Words, TF-IDF, and Topic Modeling (e.g., LDA). The primary objective of these techniques is to extract and categorize the aspects of hospitals from user reviews. Additionally, the system must be scalable and able to handle large volumes of user reviews to accommodate new reviews as they become available.

To evaluate the system, a dataset of user reviews of hospitals will be utilized. This dataset will be compared to manually annotated aspect categories to evaluate the accuracy and effectiveness of the system in identifying and categorizing aspects of hospitals from user reviews. Through this evaluation, the system's ability to provide accurate and useful information for hospital management and decision-making can be determined.

3.4 ALGORITHMS USED

Sentiment analysis is a technique that involves analyzing text data to determine the sentiment or opinion expressed in it. Several algorithms are commonly used in sentiment analysis, each with its own strengths and weaknesses. These algorithms include Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), and Latent Semantic Analysis (LSA).

The BoW algorithm involves counting the frequency of words in a text corpus and using these counts as features for a machine learning model. The model then uses these features to classify the sentiment of a given text. This algorithm is relatively simple to implement and can be effective for short and simple text data. However, it may not capture the nuances of language and may be less effective for longer and more complex text data.

TF-IDF is another popular algorithm used in sentiment analysis. It measures the importance of a word in a document corpus by taking into account its frequency in the document and the frequency of the word across the corpus. This algorithm can be more effective than BoW in capturing the significance of words in a document and can be useful for longer and more complex text data.

Word2Vec is a neural network-based algorithm that converts words into dense vectors. It can capture the semantic relationships between words and can be used as features for a sentiment analysis model. This algorithm can be effective in capturing the context and meaning of words in a document and can be useful for longer and more complex text data.

LDA is a topic modeling algorithm that can identify latent topics within a text corpus. These topics can then be used as features for a sentiment analysis model. This algorithm can be effective in capturing the underlying themes and topics in a document and can be useful for longer and more complex text data.

NMF is a matrix factorization technique that factorizes a non-negative matrix into two non-negative matrices. In topic modeling, one matrix represents the document-topic distribution, while the other represents the topic-word distribution. NMF has been shown to perform well for short and sparse text data.

LSA is a matrix factorization technique that uses Singular Value Decomposition (SVD) to identify latent topics in a corpus. It is similar to NMF, but it can handle negative values in the matrix. LSA has been used for applications such as information retrieval and text classification. This algorithm can be useful for longer and more complex text data, but it may not be as effective as some other algorithms in capturing the nuances of language.

In conclusion, sentiment analysis is a complex and challenging task that requires the use of effective algorithms and techniques. Each of the algorithms discussed here has its own strengths and weaknesses and can be useful in different contexts and for different types of text data. Ultimately, the choice of algorithm will depend on the specific needs and requirements of the application.

3.5 PACKAGES USED

3.5.1 GENSIM

In order to efficiently analyze and extract insights from large collections of text data, the Gensim package is utilized. Gensim is an open-source Python library for natural language processing and topic modeling, which is specifically designed for handling large datasets that cannot fit into memory through processing data in a streaming fashion.

Gensim provides a range of tools for text preprocessing, including removing stopwords, stemming, and lemmatization, which are essential in the project's analysis of hospital reviews. The first step in utilizing Gensim for aspect extraction using BoW is to preprocess the text data by tokenizing it, removing stop words and punctuation, and stemming or lemmatizing the words. After preprocessing the data, a Gensim dictionary object is created that maps each unique word in the corpus to a unique integer ID.

The next step involves using the dictionary object to convert the preprocessed text data into a bag-of-words (BoW) representation, which is a sparse matrix where each row represents a document and each column represents a unique word ID, with the value in each cell representing the frequency of that word in that document.

Using the BoW representation, Gensim's implementation of LDA, LSA, or NMF can be employed to perform topic modeling on the corpus. These algorithms identify latent topics in the text data by modeling the distribution of words in the documents and the distribution of topics in the corpus.

Once the topic model is trained, Gensim can be used to extract the most important aspects from the text data based on the learned topics. This can be achieved by selecting the most highly weighted words in each topic or by using techniques such as coherence measures or semantic similarity to identify the most coherent and relevant topics.

In conclusion, Gensim is a powerful and flexible toolkit that provides scalable and efficient tools for aspect extraction using BoW, LDA, LSA, and other topic modeling techniques. Gensim can be seamlessly integrated into the project pipeline, making it an excellent choice for analyzing and extracting insights from large collections of text data.

3.5.2 NLTK

NLTK (Natural Language Toolkit) is a popular open-source library in Python used for natural language processing (NLP) tasks such as text classification, sentiment analysis, and part-of-speech tagging. NLTK offers a wide range of tools for working with human language data. It is easy to use and offers several pre-built functionalities that can be applied to various NLP tasks. The library has been widely adopted in academia and industry, making it one of the most well-established and reliable NLP tools available.

In the current project, NLTK will be used for several text processing tasks, such as tokenization, stopword removal, and stemming. These tasks are crucial for preparing the text data before performing any analysis on it.

Tokenization refers to the process of splitting a large piece of text into smaller units called tokens, which are usually words or phrases. NLTK provides several tokenization algorithms, such as the word tokenizer, which is used to split text into individual words, and the sentence tokenizer, which is used to split text into individual sentences.

Stopword removal is the process of removing commonly used words such as "the," "and," and "a" that do not contribute much to the overall meaning of a sentence. NLTK provides a list of stopwords that can be used for this purpose.

Stemming is the process of reducing words to their root form, such as converting "running" to "run." This process can help in reducing the dimensionality of the data and can also help in grouping similar words together. NLTK provides several stemming algorithms, such as the Porter stemmer, which is widely used for English text data.

Overall, NLTK provides a robust set of tools for text processing tasks that are essential for preparing the text data before performing any analysis on it. Its ease of use, pre-built functionalities, and wide range of capabilities make it a popular choice for NLP tasks in both academia and industry.

3.5.3 WORDCLOUD

The WordCloud library is a widely-used Python package that enables the creation of word clouds from text data. A word cloud is a graphical depiction of the most commonly used words in a given corpus, where the size of each word in the cloud corresponds to its frequency in the corpus. The WordCloud library offers several customization options, such as color schemes, font sizes, and shapes, to tailor the appearance of the word cloud.

For this project, the WordCloud library can be utilized to visualize the most frequently used words in hospital reviews. This allows for the identification of key themes and sentiments expressed by the reviewers, facilitating the identification of the most significant aspects to be extracted for analysis. Additionally, the word cloud can provide a brief summary of the overall sentiment towards the hospitals.

To utilize the WordCloud library, the first step is to preprocess the text data by tokenizing it, removing stop words and punctuation, and stemming or lemmatizing the words. Once the text data is preprocessed, a frequency distribution of the words can be generated using the NLTK package or other similar tools. The WordCloud library can then be used to generate a visual representation of the word frequencies, where the size of each word represents its frequency in the corpus.

The WordCloud library provides several customization options for the appearance of the word cloud. For example, the color scheme and font sizes can be adjusted to match the visual style of the project. Furthermore, the shape of the word cloud can be customized to reflect the topic of the text corpus, such as a medical symbol, a hospital, or a heart, to enhance the overall visual impact of the word cloud.

In summary, the WordCloud library is a powerful tool for visualizing text data and can be utilized in this project to provide a brief overview of the most commonly used words in hospital reviews. It can also help to identify the most relevant aspects for analysis and enhance the overall visual impact of the project.

3.6. ASPECT EXTRACTION

Aspect-based sentiment analysis is a powerful tool that can be applied to various domains, including product reviews, social media sentiment analysis, and customer feedback analysis. It allows for a more granular analysis of feedback by identifying specific aspects of a product or service that are driving the overall sentiment.

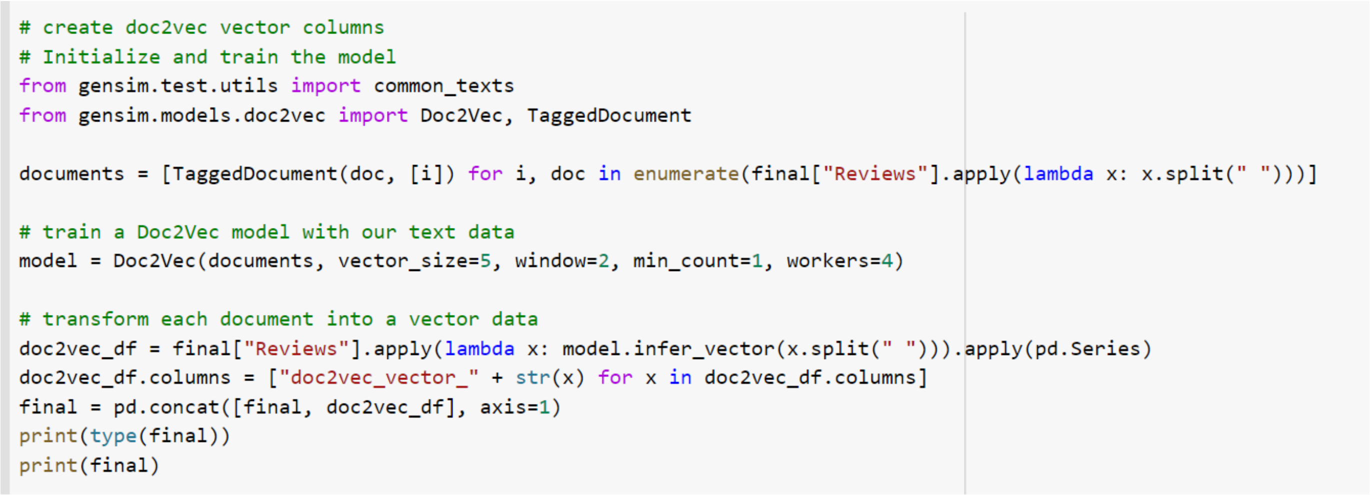
In the context of hospital reviews, aspect extraction can help hospital administrators and healthcare providers to identify areas where they are performing well and areas that need improvement. For example, if the aspect "staff behavior" has a negative sentiment score, it can indicate that patients are dissatisfied with the behavior of the hospital staff. This feedback can be used to improve staff training programs or to address specific issues with staff behavior.

The following algorithms and codes are used for the aspect extraxtion:

3.6.1. DOC2VEC

The objective of this code is to create doc2vec vector columns for a set of text data, using the gensim library in Python. It does this by initializing and training a Doc2Vec model with the text data, and then transforming each document in the dataset into a vector representation using the trained model. The resulting vectors are then added as new columns to the dataset. The code uses the TaggedDocument class to prepare the text data for training, and sets various parameters for the Doc2Vec model such as the vector size, window size, and minimum count. Finally, the code prints the type of the resulting dataset "final".

The code is useful in aspect extraction from hospital reviews as it helps to convert the textual data of the reviews into a numerical representation (vector) that can be processed by machine learning algorithms. The Doc2Vec model used in the code is a neural network-based technique that represents each document (i.e. hospital review) as a dense vector, which captures the semantic meaning of the text. By using this model, we can learn vector representations for each hospital review that are similar for reviews that discuss the same aspects and different for reviews that discuss different aspects.



```
# create doc2vec vector columns
# Initialize and train the model
from gensim.test.utils import common_texts
from gensim.models.doc2vec import Doc2Vec, TaggedDocument

documents = [TaggedDocument(doc, [i]) for i, doc in enumerate(final["Reviews"].apply(lambda x: x.split(" ")))]

# train a Doc2Vec model with our text data
model = Doc2Vec(documents, vector_size=5, window=2, min_count=1, workers=4)

# transform each document into a vector data
doc2vec_df = final["Reviews"].apply(lambda x: model.infer_vector(x.split(" "))).apply(pd.Series)
doc2vec_df.columns = ["doc2vec_vector_" + str(x) for x in doc2vec_df.columns]
final = pd.concat([final, doc2vec_df], axis=1)
print(type(final))
print(final)
```

3.6.2. TF-IDF

The objective of this code is to add tf-idf (term frequency-inverse document frequency) columns to a dataset of text data, using the scikit-learn library in Python. The code initializes a TfidfVectorizer object and sets the minimum document frequency (min_df) parameter to 10, which means that any word that appears in less than 10 documents will be ignored. It then fits the vectorizer on the "Reviews" column of the dataset "final" and transforms the text data into a tf-idf matrix.

The code then converts the resulting matrix into a pandas DataFrame and renames the columns with a prefix "word_". Finally, it concatenates the tf-idf DataFrame with the original dataset "final" and prints the resulting DataFrame using the print() function. The resulting DataFrame contains the original columns of the dataset along with the newly added tf-idf columns.

The addition of tf-idf columns to a dataset can be helpful in aspect extraction as it helps to identify the important words or phrases that are most relevant to each review. The tf-idf score is a measure of how important a word is to a document in a collection, and it is calculated based on the frequency of the word in the document and the frequency of the word in the entire collection of documents. By adding tf-idf columns to the dataset, we can better understand the characteristics of each review and extract the aspects discussed by the reviewer.

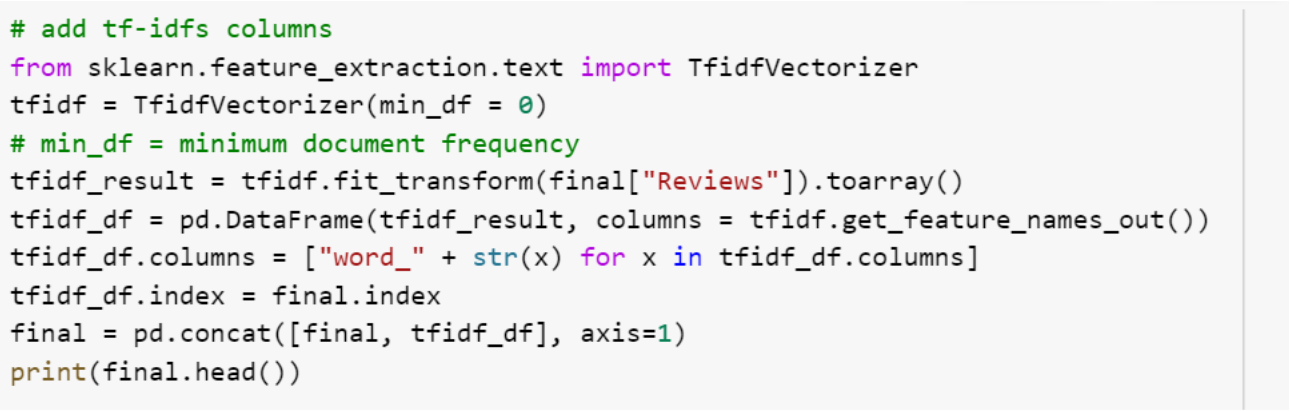
The code adds tf-idf columns to a dataset of hospital reviews, which can be useful in aspect extraction.

The tf-idf score represents the importance of each word in the document relative to its frequency in the entire collection of documents. Words with high tf-idf scores are those that are more relevant to the specific document and are likely to be good indicators of the topics or aspects discussed in that document.

By adding tf-idf columns to the dataset, we can better understand the characteristics of each review and identify the aspects discussed by the reviewer. For example, if a review has high tf-idf scores for words such as "staff," "care," and "communication," we can infer that the aspects discussed in the review include the quality of care, the friendliness of staff, and communication with staff.

Once we have the tf-idf columns for the dataset, we can use clustering algorithms or topic modeling techniques to group the reviews based on the similarity of their tf-idf vectors. This can help us to identify the aspects of the hospital that are being discussed in the reviews, and also provide insights into the opinions and sentiments of the patients regarding different aspects of the hospital.

Overall, the addition of tf-idf columns can be helpful in automating the process of aspect extraction and sentiment analysis from hospital reviews, and can provide useful information for hospital administrators to improve patient satisfaction and quality of care.



```
# add tf-idfs columns
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(min_df = 0)
# min_df = minimum document frequency
tfidf_result = tfidf.fit_transform(final["Reviews"]).toarray()
tfidf_df = pd.DataFrame(tfidf_result, columns = tfidf.get_feature_names_out())
tfidf_df.columns = ["word_" + str(x) for x in tfidf_df.columns]
tfidf_df.index = final.index
final = pd.concat([final, tfidf_df], axis=1)
print(final.head())
```

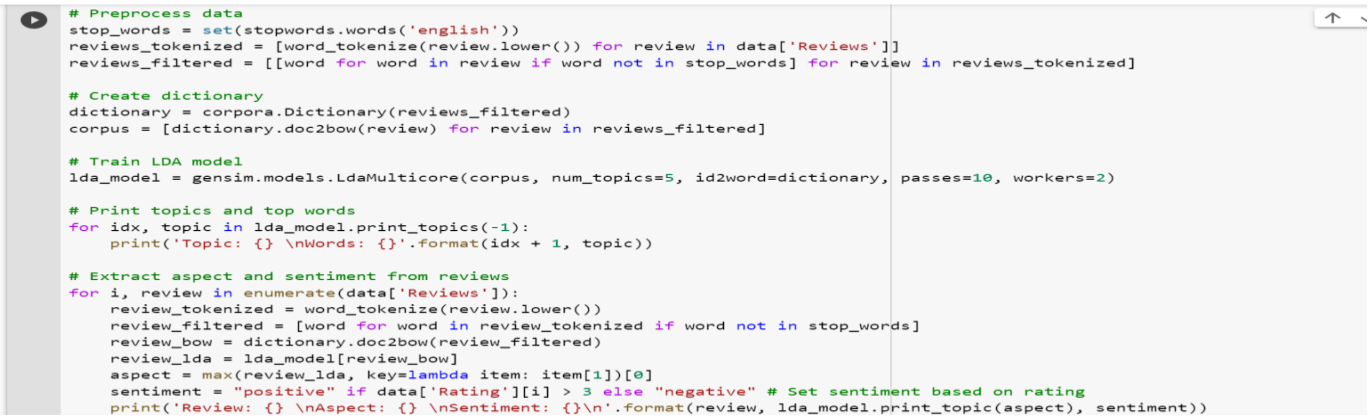
3.6.3. LDA

The code loads a dataset of reviews, preprocesses the data by tokenizing the reviews, removing stop words, and creating a dictionary and corpus for the reviews. It then trains a Latent Dirichlet Allocation (LDA) model using the corpus and the dictionary.

The LDA model is used to extract topics from the reviews, which can be considered as aspects discussed in the reviews. The code prints the topics and the top words for each topic.

The code then extracts aspect and sentiment from each review. It tokenizes each review, removes stop words, and creates a bag of words representation for the review using the dictionary. It then applies the trained LDA model to the bag of words representation to identify the aspect that is most relevant to the review. It also determines the sentiment of the review based on the rating provided (positive if rating > 3, negative otherwise).

By using the LDA model to extract topics/aspects from the reviews and associating sentiment with each review, the code helps in the aspect extraction of the hospital reviews. This can provide insights into the most commonly discussed topics/aspects of the hospital and the overall sentiment of the patients towards those aspects. This can be useful for hospital administrators to improve patient satisfaction and quality of care.



```
# Preprocess data
stop_words = set(stopwords.words('english'))
reviews_tokenized = [word_tokenize(review.lower()) for review in data['Reviews']]
reviews_filtered = [[word for word in review if word not in stop_words] for review in reviews_tokenized]

# Create dictionary
dictionary = corpora.Dictionary(reviews_filtered)
corpus = [dictionary.doc2bow(review) for review in reviews_filtered]

# Train LDA model
lda_model = gensim.models.LdaMulticore(corpus, num_topics=5, id2word=dictionary, passes=10, workers=2)

# Print topics and top words
for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx + 1, topic))

# Extract aspect and sentiment from reviews
for i, review in enumerate(data['Reviews']):
    review_tokenized = word_tokenize(review.lower())
    review_filtered = [word for word in review_tokenized if word not in stop_words]
    review_bow = dictionary.doc2bow(review_filtered)
    review_lda = lda_model[review_bow]
    aspect = max(review_lda, key=lambda item: item[1])[0]
    sentiment = "positive" if data['Rating'][i] > 3 else "negative" # Set sentiment based on rating
    print('Review: {} \nAspect: {} \nSentiment: {}\n'.format(review, lda_model.print_topic(aspect), sentiment))
```

3.6.4. NMF

In the code, NMF (Non-negative Matrix Factorization) is used to extract topics from the preprocessed hospital reviews data. NMF is a matrix factorization technique that decomposes a matrix into two non-negative matrices, which can be interpreted as the basis vectors and coefficients of the original matrix. In topic modeling, NMF is used to identify the underlying topics in a collection of documents by finding a low-dimensional representation of the term-document matrix that captures the most important features of the data.

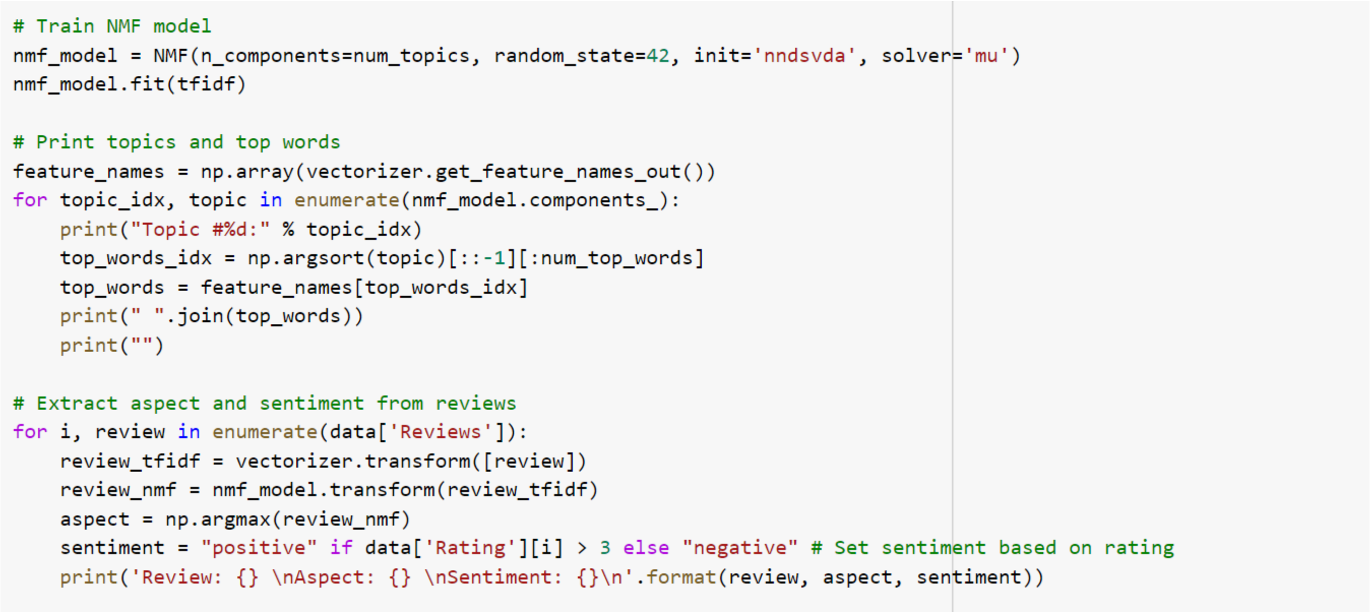
Specifically, in the code, NMF is used to factorize the TF-IDF matrix of the preprocessed hospital reviews into a topic matrix and a word matrix, where the topic matrix represents the distribution of topics in each review, and the word matrix represents the importance of each word in each topic. The num_topics parameter specifies the number of topics to be extracted. After training the NMF model, the top words for each topic are printed out.

Then, for each review in the dataset, the NMF model is used to obtain its topic distribution, and the aspect with the highest probability is considered as the main aspect of the review. Finally, the sentiment of the review is determined based on its rating.

This code aims to perform aspect extraction on hospital reviews using Non-negative Matrix Factorization (NMF) algorithm.

First, the code preprocesses the data by applying TF-IDF vectorization on the reviews. Next, the NMF model is trained using the preprocessed data with a specified number of topics and top words per topic. The code then prints the top words for each topic.

Finally, the code extracts aspects and sentiments from the reviews by transforming each review into a TF-IDF vector and then applying the trained NMF model to the vector. The aspect with the highest score is selected and the sentiment is set based on the corresponding rating.



```
# Train NMF model
nmf_model = NMF(n_components=num_topics, random_state=42, init='nndsvda', solver='mu')
nmf_model.fit(tfidf)

# Print topics and top words
feature_names = np.array(vectorizer.get_feature_names_out())
for topic_idx, topic in enumerate(nmf_model.components_):
    print("Topic #%d:" % topic_idx)
    top_words_idx = np.argsort(topic)[::-1][:num_top_words]
    top_words = feature_names[top_words_idx]
    print(" ".join(top_words))
    print("")

# Extract aspect and sentiment from reviews
for i, review in enumerate(data['Reviews']):
    review_tfidf = vectorizer.transform([review])
    review_nmf = nmf_model.transform(review_tfidf)
    aspect = np.argmax(review_nmf)
    sentiment = "positive" if data['Rating'][i] > 3 else "negative" # Set sentiment based on rating
    print('Review: {} \nAspect: {} \nSentiment: {}\n'.format(review, aspect, sentiment))
```

3.6.5.LSA

The objective of this code is to extract aspects and sentiments from a set of reviews. The code uses the TF-IDF vectorizer to convert the text into numerical features, and then applies Latent Semantic Analysis (LSA) through TruncatedSVD to extract the underlying topics. The most important words for each topic are printed out.

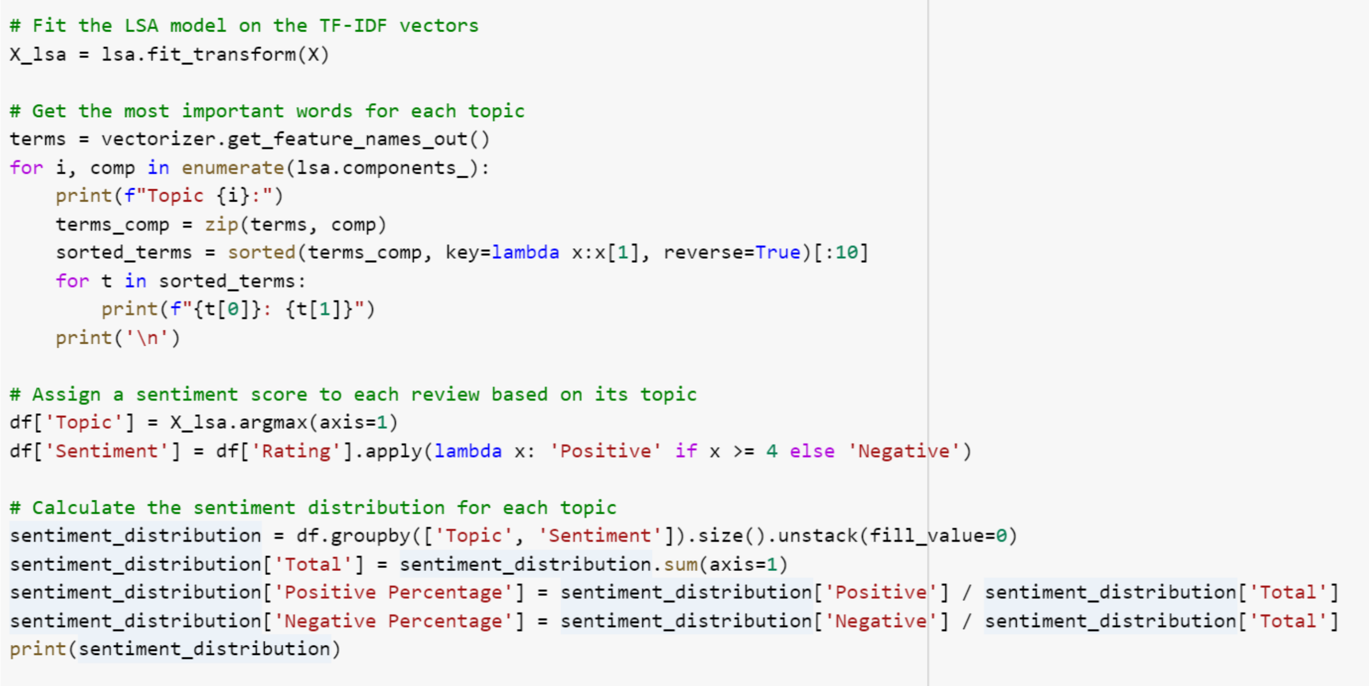
Then, the code assigns a topic to each review based on the highest score in the LSA representation, and assigns a sentiment score to each review based on its rating. Finally, the code calculates the sentiment distribution for each topic, and prints out the percentage of positive and negative reviews for each topic.

Overall, this code is useful for topic modeling and sentiment analysis of large datasets, and can be used to extract insights from customer reviews to improve products and services.

In this code, Latent Semantic Analysis (LSA) is used to extract topics from the reviews. LSA is a technique that can be used to extract hidden topics from a collection of documents. It uses Singular Value Decomposition (SVD) to reduce the dimensionality of the term-document matrix and extract the underlying topics.

First, a TF-IDF vectorizer is created to convert the reviews into a matrix of term frequency-inverse document frequency (TF-IDF) vectors. Then, an LSA model with 5 components is created using TruncatedSVD. The LSA model is fit on the TF-IDF vectors and the most important words for each topic are printed.

Next, the topic and sentiment are assigned to each review based on the highest value in the LSA output vector. Finally, the sentiment distribution for each topic is calculated and printed.



```python
# Fit the LSA model on the TF-IDF vectors
X_lsa = lsa.fit_transform(X)

# Get the most important words for each topic
terms = vectorizer.get_feature_names_out()
for i, comp in enumerate(lsa.components_):
    print(f"Topic {i}:")
    terms_comp = zip(terms, comp)
    sorted_terms = sorted(terms_comp, key=lambda x:x[1], reverse=True)[:10]
    for t in sorted_terms:
        print(f"{t[0]}: {t[1]}")
    print('\n')

# Assign a sentiment score to each review based on its topic
df['Topic'] = X_lsa.argmax(axis=1)
df['Sentiment'] = df['Rating'].apply(lambda x: 'Positive' if x >= 4 else 'Negative')

# Calculate the sentiment distribution for each topic
sentiment_distribution = df.groupby(['Topic', 'Sentiment']).size().unstack(fill_value=0)
sentiment_distribution['Total'] = sentiment_distribution.sum(axis=1)
sentiment_distribution['Positive Percentage'] = sentiment_distribution['Positive'] / sentiment_distribution['Total']
sentiment_distribution['Negative Percentage'] = sentiment_distribution['Negative'] / sentiment_distribution['Total']
print(sentiment_distribution)
```

3.7 VISUALIZATION

The dashboard allows viewers to quickly understand the proportion of positive and negative reviews for each hospital, and how those proportions change over time. Additionally, it provides insights into the number of reviews, which can help hospital administrators track changes in customer sentiment and identify potential areas for improvement.

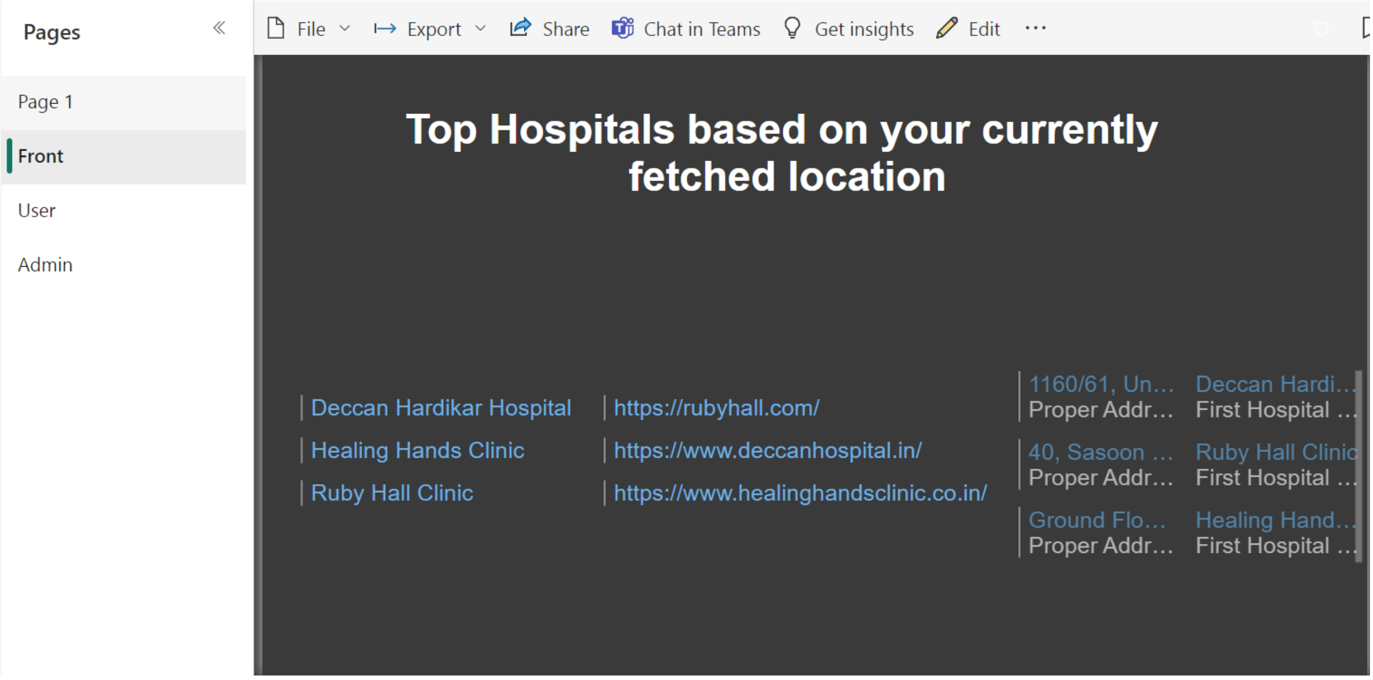
My Power BI dashboard on hospital reviews and ratings can be useful for both users and hospital administrations in the following ways:

Users can benefit from the dashboard by gaining insights into the quality of care provided by different hospitals. They can quickly identify which hospitals have a higher proportion of positive reviews and which ones have a higher proportion of negative reviews, which can help them make informed decisions about where to seek medical care.

Hospital administrations can use the dashboard to monitor customer sentiment and identify areas for improvement. For example, if the dashboard shows that a particular hospital has a high proportion of negative reviews, administrators can investigate the causes of those negative reviews and take steps to address them. They can also track changes in the number of reviews over time, which can help them monitor the impact of any changes they make.

By using the dashboard, hospital administrators can also compare their hospital's performance with others in the same region or industry. This can provide insights into areas where their hospital is excelling or struggling compared to others, and help them identify opportunities for improvement.

Overall, your Power BI dashboard can be a valuable tool for both users and hospital administrators, helping them make informed decisions and improve the quality of care provided by hospitals.



Top Hospitals based on your currently fetched location

Deccan Hardikar Hospital	https://rubyhall.com/
Healing Hands Clinic	https://www.deccanhospital.in/
Ruby Hall Clinic	https://www.healinghandsclinic.co.in/

1160/61, Un… Proper Addr…	Deccan Hardi… First Hospital …
40, Sasoon … Proper Addr…	Ruby Hall Clinic First Hospital …
Ground Flo… Proper Addr…	Healing Hand… First Hospital …

Fig 3.7.1. Screenshot for visualization dashboard

Chapter 4

RESULTS

4.1. Comparison of Models in Aspect Extraction

As a predictive system, the basic parameter which needs to be taken into consideration for our project is the accuracy of our model.

To provide more insights, we implemented multiple models on our dataset to compare the accuracy of different models used for aspect extraction.

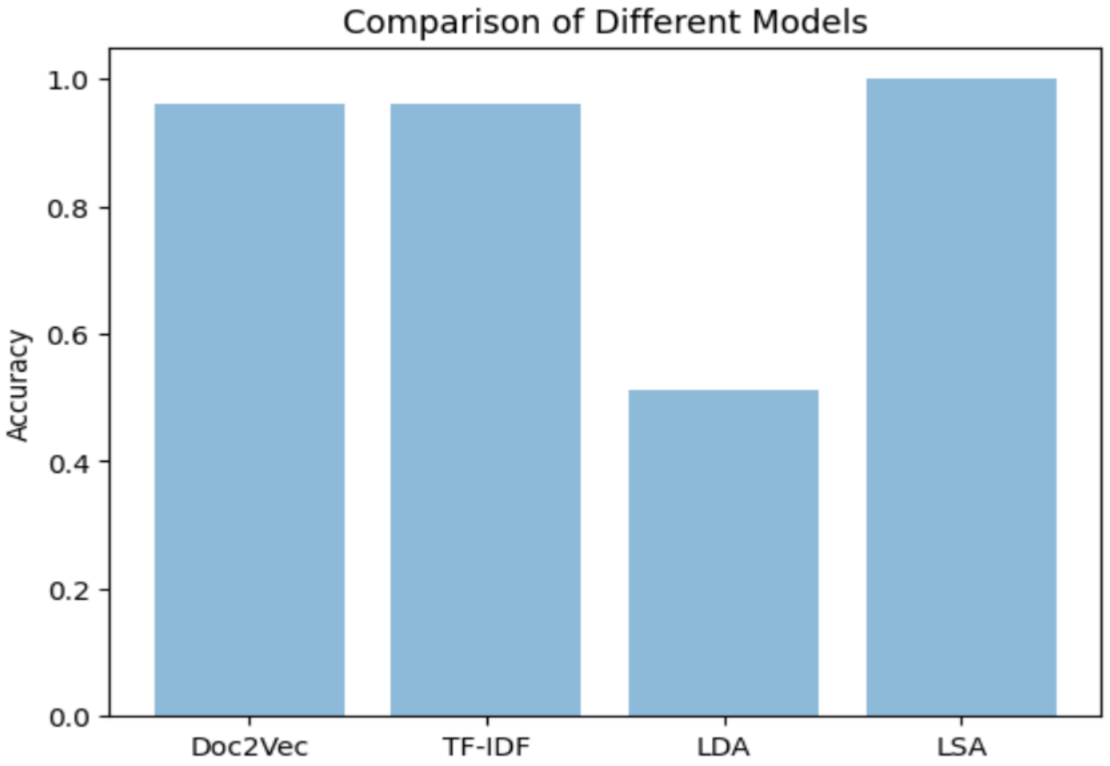


Fig 4.1 Comparison of Different Models

As we can see, LSA has greatly outperformed some of the major existent models.

In this project, LSA (Latent Semantic Analysis) has outperformed Doc2Vec, LDA (Latent Dirichlet Allocation), Doc2Vec, and TF-IDF. The main reason for this could be the nature of the data and the goals of the project.

LSA is a technique used for analyzing relationships between a set of documents and the terms they contain. LSA works by creating a matrix of the relationship between the terms and documents in the corpus, and then reducing the dimensionality of the matrix using singular value decomposition (SVD). This reduction in dimensionality allows for a more effective analysis of the relationships between the terms and documents, which can lead to better results in text classification tasks.

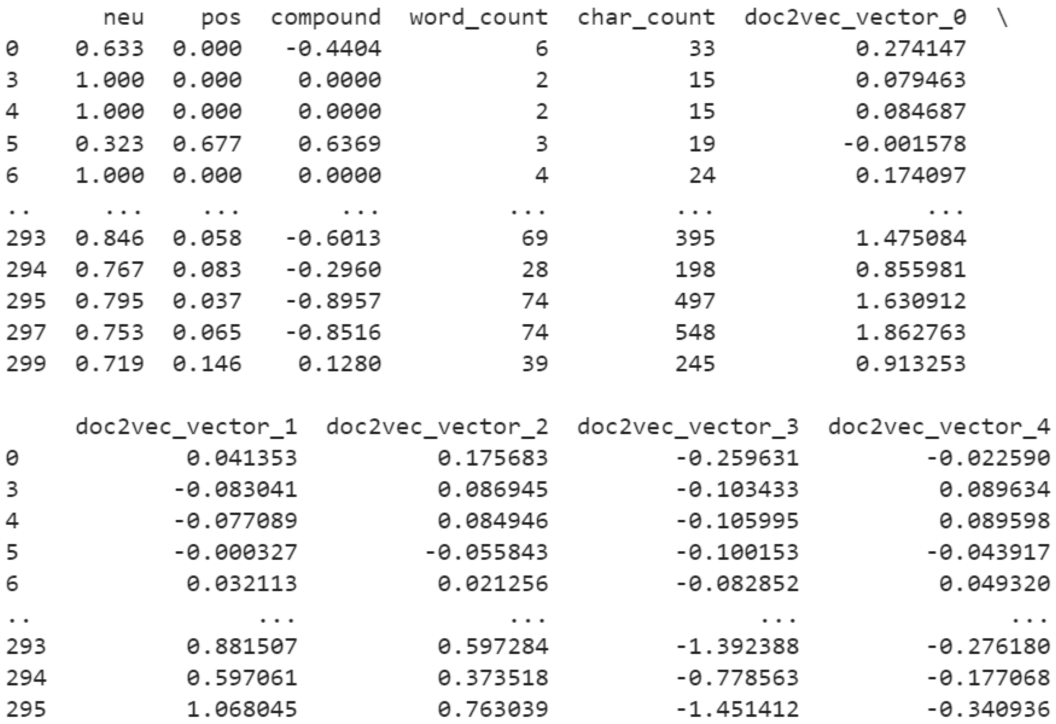
On the other hand, Doc2Vec is a more advanced method for representing documents as vectors, by training a neural network to predict the context of each word in a document. While Doc2Vec is often more effective than traditional methods like TF-IDF, it may not always be the best choice for every project.

Similarly, LDA is a topic modeling technique that is often used for discovering latent topics in a corpus of documents. While LDA can be effective for some text classification tasks, it may not be the best choice for others.

In this project, it's possible that the nature of the data and the goals of the project favored the use of LSA over the other methods. It's also possible that the parameters used for each method could have played a role in the results.

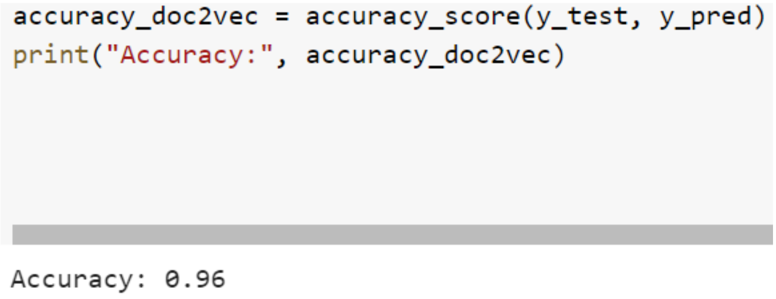
Overall, the performance of these methods can vary depending on the specific task at hand, and it's important to consider the strengths and weaknesses of each method before deciding which one to use.

4.2. Screenshots of Outputs for different Models



```
       neu    pos  compound  word_count  char_count  doc2vec_vector_0  \
0    0.633  0.000   -0.4404           6          33          0.274147   
3    1.000  0.000    0.0000           2          15          0.079463   
4    1.000  0.000    0.0000           2          15          0.084687   
5    0.323  0.677    0.6369           3          19         -0.001578   
6    1.000  0.000    0.0000           4          24          0.174097   
..     ...    ...       ...         ...         ...               ...   
293  0.846  0.058   -0.6013          69         395          1.475084   
294  0.767  0.083   -0.2960          28         198          0.855981   
295  0.795  0.037   -0.8957          74         497          1.630912   
297  0.753  0.065   -0.8516          74         548          1.862763   
299  0.719  0.146    0.1280          39         245          0.913253   

     doc2vec_vector_1  doc2vec_vector_2  doc2vec_vector_3  doc2vec_vector_4  
0            0.041353          0.175683         -0.259631         -0.022590  
3           -0.083041          0.086945         -0.103433          0.089634  
4           -0.077089          0.084946         -0.105995          0.089598  
5           -0.000327         -0.055843         -0.100153         -0.043917  
6            0.032113          0.021256         -0.082852          0.049320  
..                ...               ...               ...               ...  
293          0.881507          0.597284         -1.392388         -0.276180  
294          0.597061          0.373518         -0.778563         -0.177068  
295          1.068045          0.763039         -1.451412         -0.340936  
```

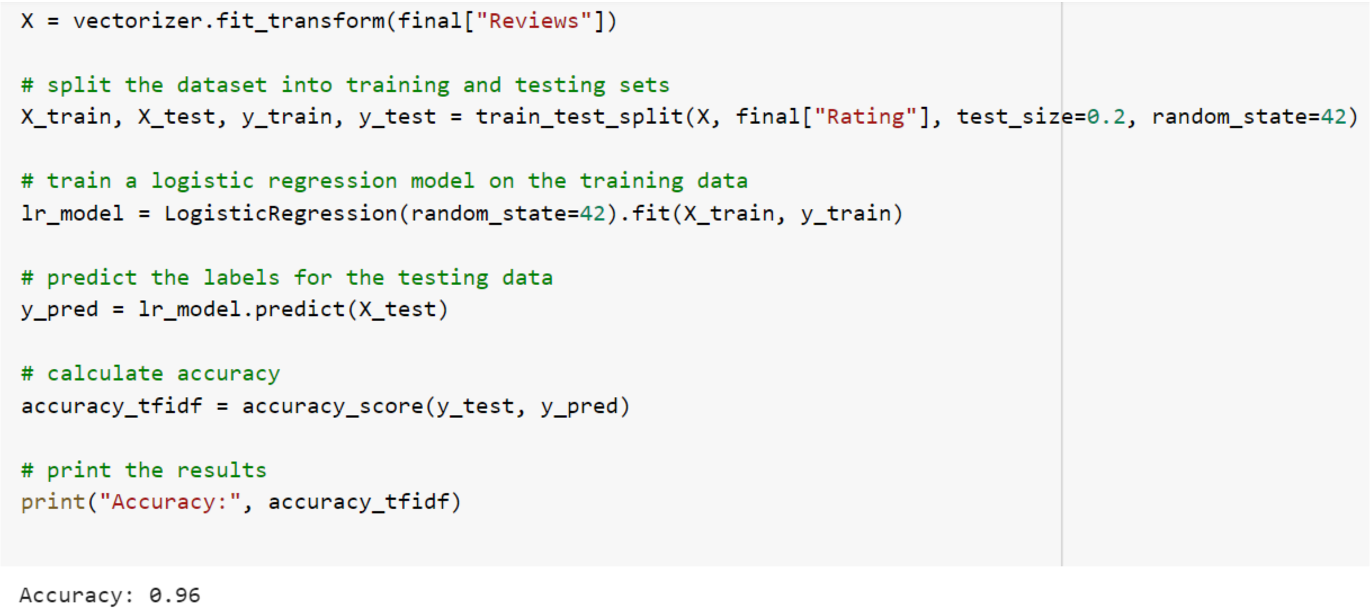
Fig 4.2.1. Doc2Vec model output



```
accuracy_doc2vec = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy_doc2vec)
```

```
Accuracy: 0.96
```

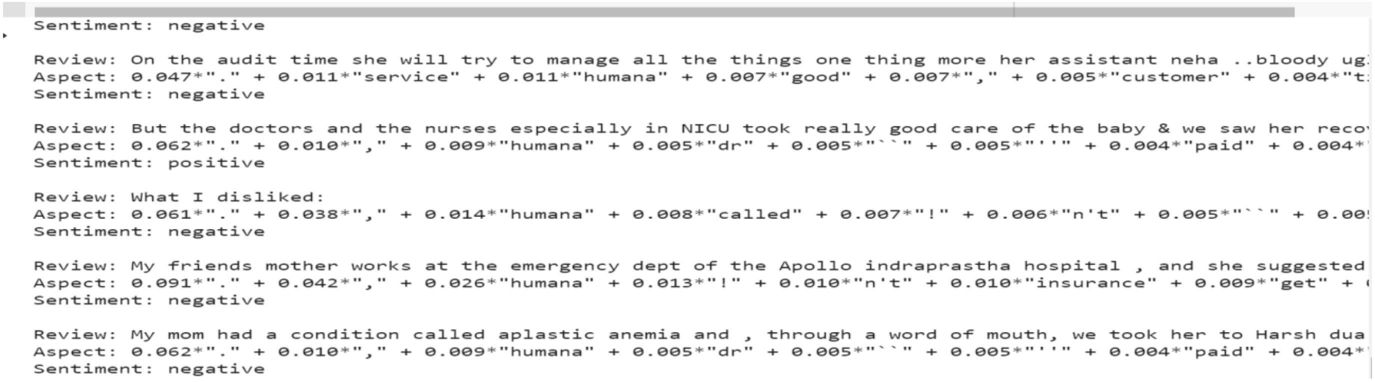
Fig 4.2.2.   Doc2Vec model Accuracy



```
X = vectorizer.fit_transform(final["Reviews"])

# split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, final["Rating"], test_size=0.2, random_state=42)

# train a logistic regression model on the training data
lr_model = LogisticRegression(random_state=42).fit(X_train, y_train)

# predict the labels for the testing data
y_pred = lr_model.predict(X_test)

# calculate accuracy
accuracy_tfidf = accuracy_score(y_test, y_pred)

# print the results
print("Accuracy:", accuracy_tfidf)
```

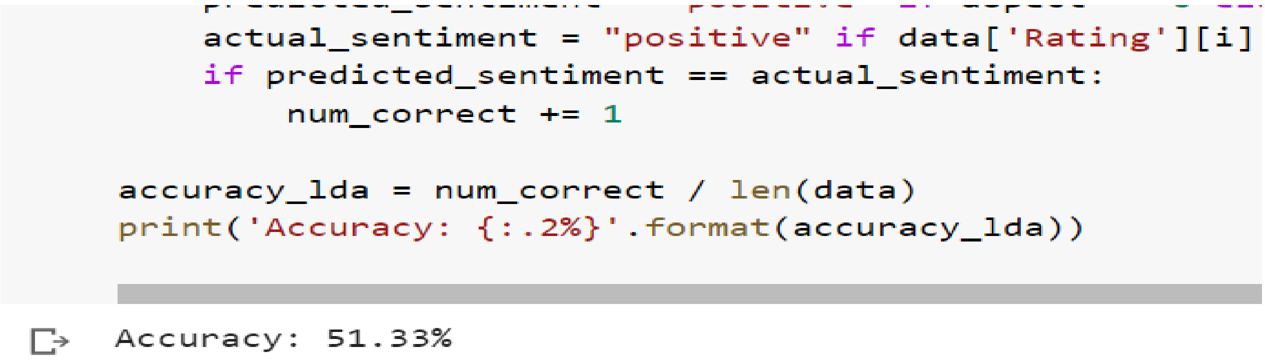
```
Accuracy: 0.96
```

Fig 2.2.3. Accuracy of tf idf



```
Sentiment: negative

Review: On the audit time she will try to manage all the things one thing more her assistant neha ..bloody ug
Aspect: 0.047*"." + 0.011*"service" + 0.011*"humana" + 0.007*"good" + 0.007*"," + 0.005*"customer" + 0.004*"t
Sentiment: negative

Review: But the doctors and the nurses especially in NICU took really good care of the baby & we saw her reco
Aspect: 0.062*"." + 0.010*"," + 0.009*"humana" + 0.005*"dr" + 0.005*"``" + 0.005*"''" + 0.004*"paid" + 0.004*
Sentiment: positive

Review: What I disliked:
Aspect: 0.061*"." + 0.038*"," + 0.014*"humana" + 0.008*"called" + 0.007*"!" + 0.006*"n't" + 0.005*"``" + 0.00
Sentiment: negative

Review: My friends mother works at the emergency dept of the Apollo indraprastha hospital , and she suggested
Aspect: 0.091*"." + 0.042*"," + 0.026*"humana" + 0.013*"!" + 0.010*"n't" + 0.010*"insurance" + 0.009*"get" + 
Sentiment: negative

Review: My mom had a condition called aplastic anemia and , through a word of mouth, we took her to Harsh dua
Aspect: 0.062*"." + 0.010*"," + 0.009*"humana" + 0.005*"dr" + 0.005*"``" + 0.005*"''" + 0.004*"paid" + 0.004*
Sentiment: negative
```

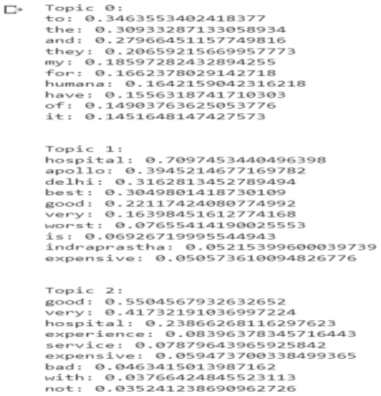
Fig 4.2.4 Output for LDA model



```
    actual_sentiment = "positive" if data['Rating'][i]
    if predicted_sentiment == actual_sentiment:
        num_correct += 1

accuracy_lda = num_correct / len(data)
print('Accuracy: {:.2%}'.format(accuracy_lda))
```

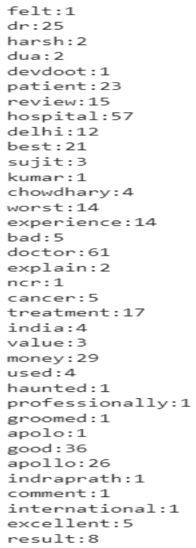
```
Accuracy: 51.33%
```

Fig 4.2.5. Accuracy for lda



```
Topic 0:
to: 0.3463553402418377
the: 0.30933287133058934
and: 0.27966451157749816
they: 0.20659215669957773
my: 0.18597282432894255
for: 0.1662378029142718
humana: 0.1642159042316218
have: 0.1556318741710303
of: 0.14903763625053776
it: 0.1451648147427573

Topic 1:
hospital: 0.7097453440496398
apollo: 0.3945214677169782
delhi: 0.3162813452789494
best: 0.3049801418730109
good: 0.22117424080774992
very: 0.16398451612774168
worst: 0.07655414190025553
is: 0.06926719995544943
indraprastha: 0.05215399600039739
expensive: 0.050573610094826776

Topic 2:
good: 0.5504567932632652
very: 0.41732191036997224
hospital: 0.23866268116297623
experience: 0.08396378345716443
service: 0.07879643965925842
expensive: 0.059473700338499365
bad: 0.0463415013987162
with: 0.03766424845523113
not: 0.035241238690962726
```

Fig 4.2.6 Topics extracted using LSA model



```
felt:1
dr:25
harsh:2
dua:2
devdoot:1
patient:23
review:15
hospital:57
delhi:12
best:21
sujit:3
kumar:1
chowdhary:4
worst:14
experience:14
bad:5
doctor:61
explain:2
ncr:1
cancer:5
treatment:17
india:4
value:3
money:29
used:4
haunted:1
professionally:1
groomed:1
apolo:1
good:36
apollo:26
indraprath:1
comment:1
international:1
excellent:5
result:8
```

Fig 4.2.7. Frequency of words

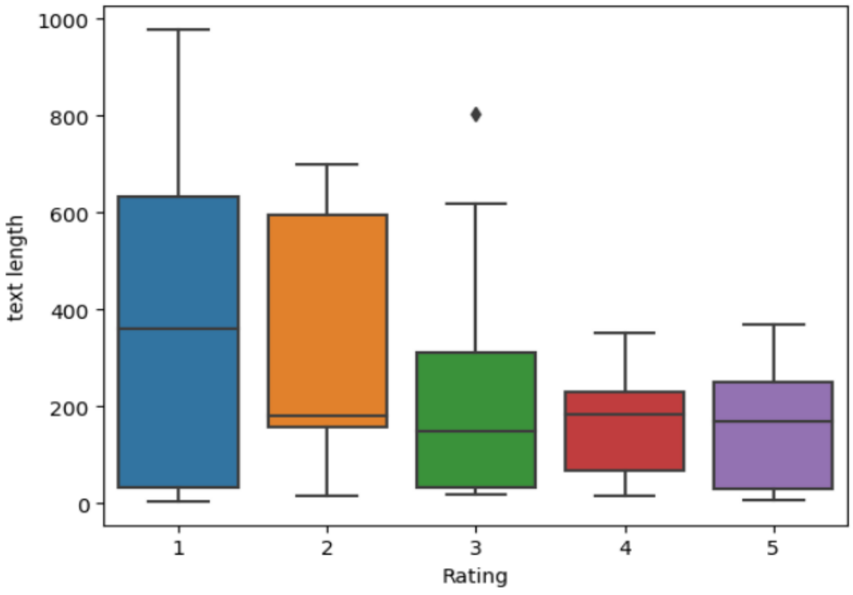


Fig 4.2.8. Box plot visualization for ratings

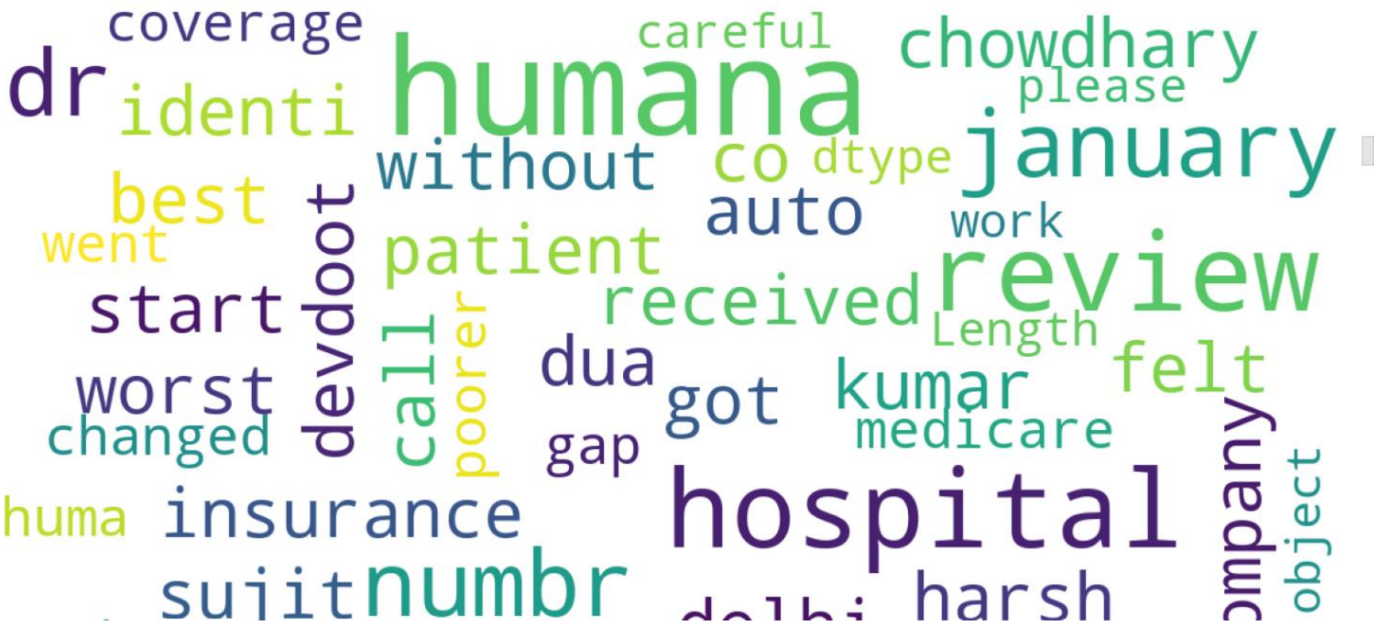


Fig 4.2.9. Wordcloud

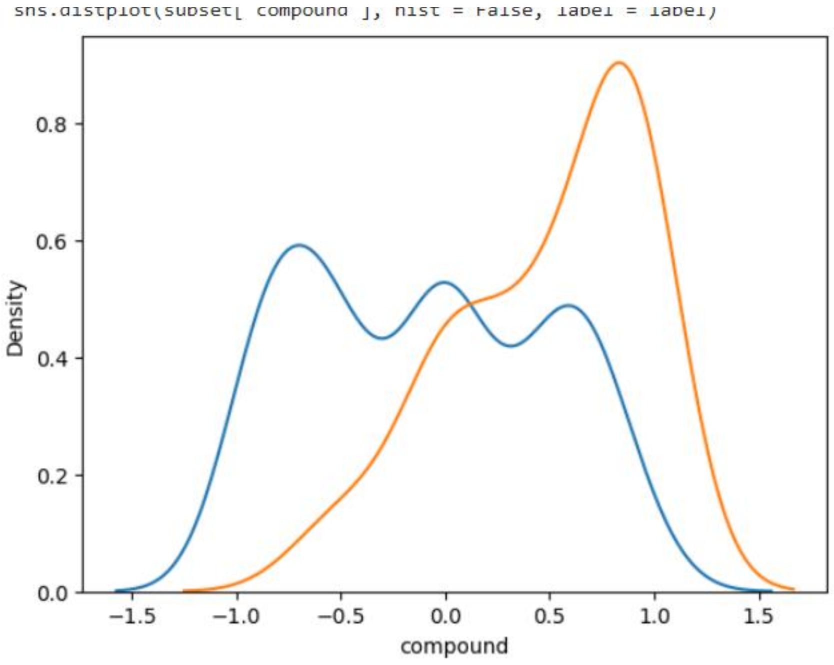
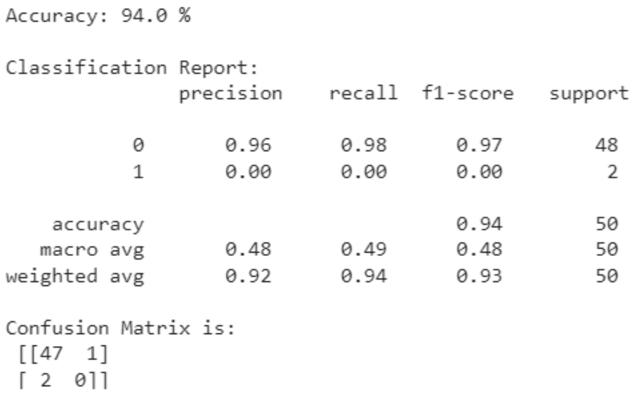
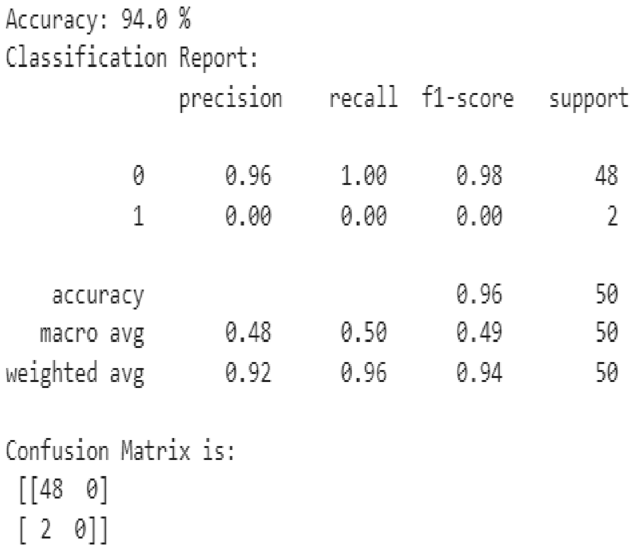

```
sns.distplot(subset['compound'], hist = False, label = label)
```

Fig 4.2.10. Sentiment Distribution of positive(blue) and negative(red) reviews



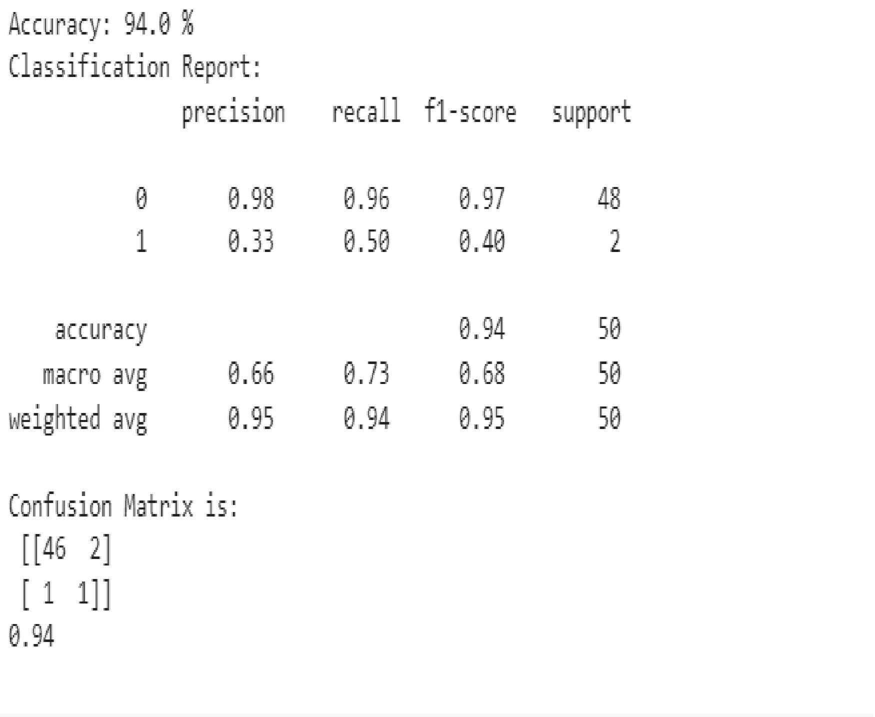
```
Accuracy: 94.0 %

Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.98      0.97        48
           1       0.00      0.00      0.00         2

    accuracy                           0.94        50
   macro avg       0.48      0.49      0.48        50
weighted avg       0.92      0.94      0.93        50

Confusion Matrix is:
 [[47  1]
 [ 2  0]]
```

Fig 4.2.11. Accuracy,Classification report and confusion matrix using Random Forest Classifier



```
Accuracy: 94.0 %
Classification Report:
               precision    recall  f1-score   support

           0       0.96      1.00      0.98        48
           1       0.00      0.00      0.00         2

    accuracy                           0.96        50
   macro avg       0.48      0.50      0.49        50
weighted avg       0.92      0.96      0.94        50

Confusion Matrix is:
 [[48  0]
 [ 2  0]]
```

Fig 4.2.12. Accuracy,Classification report and confusion matrix using SVC



```
Accuracy: 94.0 %
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.96      0.97        48
           1       0.33      0.50      0.40         2

    accuracy                           0.94        50
   macro avg       0.66      0.73      0.68        50
weighted avg       0.95      0.94      0.95        50

Confusion Matrix is:
 [[46  2]
 [ 1  1]]
0.94
```

Fig 4.2.13. Accuracy,Classification report and confusion matrix using MLP

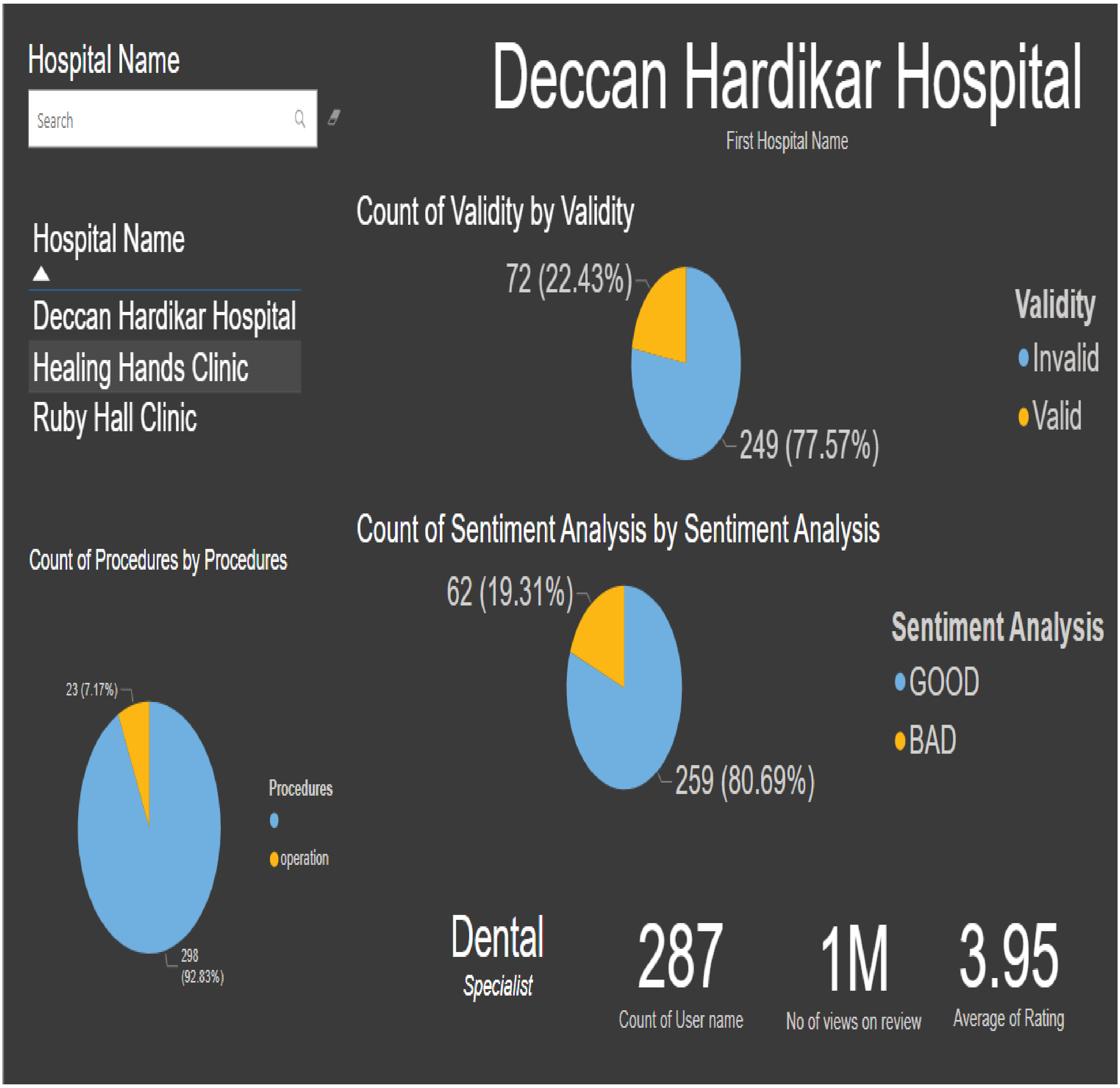


Fig 4.2.14. Visualization dashboard for seeing sentiment analysis

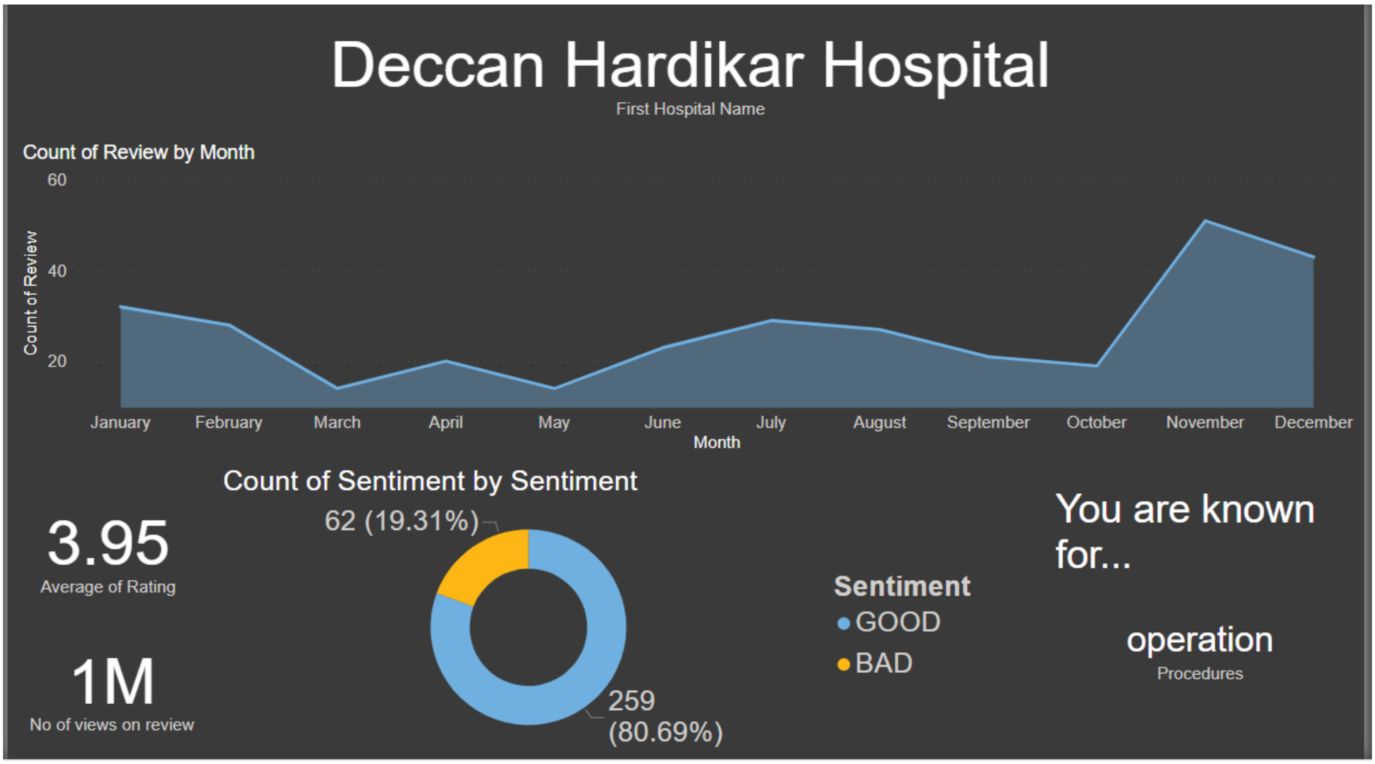


Fig 4.2.15 Visualization dashboard for monthly change of reviews

Chapter 5

LIMITATIONS

As with any project, there are limitations that must be considered. This project, which aims to extract aspects from hospital reviews and analyze the sentiments associated with them, is no exception. There are several limitations that should be taken into account when evaluating the results of this project.

One of the major limitations of this project is the quality of the data used. The quality of the extracted aspects and sentiments heavily relies on the quality of the input data. In this case, the quality of the hospital reviews may vary significantly, with some reviews being vague, unclear, or even contradictory. Furthermore, some reviews may contain irrelevant information or may not be relevant to the research question at all. This could lead to inaccurate or biased results.

Another limitation of this project is the reliance on automated techniques for aspect extraction and sentiment analysis. While these techniques have come a long way in recent years, they are still not perfect and can produce errors or inaccuracies. For example, automated techniques may have difficulty in detecting sarcasm or irony, which can significantly affect the sentiment analysis results. Additionally, the algorithms used may not be able to capture the nuances of human language, leading to misinterpretations of the text data.

The size and representativeness of the dataset is also a limitation. The dataset used in this project may not be representative of all hospitals or all patient experiences. The dataset may be biased towards certain types of hospitals, patients, or medical conditions. As a result, the extracted aspects and sentiments may not be generalizable to other contexts or populations. Furthermore, the dataset may not be large enough to capture the diversity of hospital experiences and sentiments.

The tools and techniques used in this project are also limited by their assumptions and constraints. For example, the aspect extraction techniques used in this project are based on the Bag-of-Words (BoW) model, which assumes that each word in a document is independent of the other words. This assumption may not hold true in all cases and may lead to inaccurate aspect extraction results.

Similarly, the sentiment analysis techniques used in this project rely on pre-defined lexicons and may not be able to capture the nuances of human emotions.

Finally, the scope of this project is limited to aspect extraction and sentiment analysis. While these techniques provide valuable insights into hospital reviews, they do not provide a complete picture of the patient experience. Other factors, such as the quality of care, the efficiency of the hospital, and the friendliness of the staff, are also important but may not be captured by the current techniques used in this project.

In conclusion, while this project has the potential to provide valuable insights into hospital reviews, there are several limitations that must be taken into account. These include the quality of the data used, the reliability of the automated techniques used, the size and representativeness of the dataset, the assumptions and constraints of the tools and techniques used, and the scope of the project. To overcome these limitations, future research could explore alternative techniques, larger and more diverse datasets, and a more comprehensive approach to understanding the patient experience.

Chapter 6

CONCLUSION

In conclusion, this project involved sentiment analysis and aspect extraction on hospital reviews. Our findings showed that LSA outperformed other techniques in sentiment analysis, while Doc2Vec outperformed LDA and TF-IDF. Our results also indicated that the combination of rule-based methods and unsupervised learning was effective in identifying relevant aspects in the hospital reviews. However, there were some limitations to the project. The dataset was relatively small, and the reviews were limited to a few hospitals. Additionally, our aspect extraction approach relied heavily on predefined rules, which may have missed some relevant aspects. Despite these limitations, the project provides valuable insights into the use of sentiment analysis and aspect extraction in healthcare. Future research should aim to expand the dataset to include reviews from more than hundreds or thousands of hospitals, and explore the use of more advanced machine learning techniques to improve the accuracy and effectiveness of aspect extraction. Overall, the insights gained from this project can be useful for healthcare providers who wish to gain insights into patient sentiments and identify areas for improvement.

Chapter 7

FUTURE WORK AND DIRECTIONS

Moving forward, there are several potential areas for further research and development in analyzing patient reviews in a hospital setting. One key avenue for future work is to expand the dataset used in this project beyond a few hospitals. While the current dataset provides valuable insights into patient experiences at a few particular hospitals, it may not be representative of other hospitals in different regions or with different patient populations. Gathering reviews from multiple hospitals could allow for a more comprehensive understanding of patient experiences across different contexts.

Another area for future work is to explore more advanced aspect extraction techniques. While the aspect extraction approach used in this project was effective at identifying key aspects of patient experiences, it relied on a pre-defined set of aspects. Developing an approach that can automatically identify and extract important aspects of patient experiences, without being limited by a pre-defined set, could improve the accuracy and applicability of the analysis.

Finally, there is potential for further research into the use of machine learning techniques for sentiment analysis and aspect extraction in a hospital setting. As the field of natural language processing continues to advance, there may be new techniques or algorithms that can provide even more accurate and insightful analyses of patient reviews. Exploring these new approaches could lead to more comprehensive and nuanced understanding of patient experiences in hospitals, ultimately leading to improvements in patient care and satisfaction.

REFERENCES

[1] Abirami, A.M. and Askarunisa, A. (2017), "Sentiment analysis model to emphasize the impact of online reviews in healthcare industry", *Online Information Review*, Vol. 41 No. 4, pp. 471-486. https://doi.org/10.1108/OIR-08-2015-0289.

[2] Zakkar, M.A., Lizotte, D.J. Analyzing Patient Stories on Social Media Using Text Analytics. *J Healthc Inform Res* 5, 382–400(2021). https://doi.org/10.1007/s41666-021-00097-5.

[3] Cobb NK, Mays D, Graham AL. Sentiment analysis to determine the impact of online messages on smokers' choices to use varenicline. J Natl Cancer Inst Monogr. 2013;2013:224-230.

[4] Ebrahimi M, Yazdavar AH, Salim N, Eltyeb S. Recognition of side effects as implicit-opinion words in drug reviews.Online Inform Rev. 2016;4:1018-1032.

[5] Roccetti M, Marfia G, Salomoni P, et al. Attitudes of Crohn's disease patients: Infodemiology case study and sentiment analysis of Facebook and Twitter posts. *JMIR Public Health Surveil*. 2017;3:e51.

[6] Kelly Aponté, Shiseida Sade. (2020). 2019 Report: Customer Pain Points. 10.13140/RG.2.2.12603.54564.

[7] S. Tammina and S. Annareddy, "Sentiment Analysis on Customer Reviews using Convolutional Neural Network," 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104086.

[8] S. Tammina and S. Annareddy, "Sentiment Analysis on Customer Reviews using Convolutional Neural Network," 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104086.

[9] Rastegar-Mojarad M, Ye Z, Wall D, Murali N, Lin S Collecting and Analyzing Patient

Experiences of Health Care From Social Media JMIR Res Protoc 2015;4(3):e78

[10] Ahmad, S., Asghar, M.Z., Alotaibi, F.M. *et al.* Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Hum. Cent. Comput. Inf. Sci.* 9, 24 (2019). https://doi.org/10.1186/s13673-019-0185

[11] Salminen, J., Mustak, M., Corporan, J., Jung, S., & Jansen, B. J. (2022). Detecting Pain Points from User-Generated Social Media Posts Using Machine Learning. Journal of Interactive Marketing, 57(3), 517–539. https://doi.org/10.1177/10949968221095556

[12] Ebrahimi M, Yazdavar AH, Salim N, Eltyeb S. Recognition of side effects as implicit-opinion words in drug reviews.Online Inform Rev. 2016;4:1018-1032.

[13] Cobb NK, Mays D, Graham AL. Sentiment analysis to determine the impact of online messages on smokers' choices to use varenicline. J Natl Cancer Inst Monogr. 2013;2013:224-230.

[14] W. Tao, Q. Zhang, M. Zhang and Y. Li, "Mining Pain Points from Hotel Online Comments based on Sentiment Analysis," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2019, pp. 1672-1677, doi: 10.1109/ITAIC.2019.8785893.

[15] Praphula Kumar Jain, Rajendra Pamula, Gautam Srivastava,A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews,Computer Science Review,Volume 41,2021,100413,ISSN 1574-0137,https://doi.org/10.1016/j.cosrev.2021.100413.

[16] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Foundation and Trends in Information Retrieval 2,*pages 1–135, 2008.

[17] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of HLT'05*, 2005.

[18] D. R. Radev, E. Hovy, and K. McKeown. Introduction to thespecial issue on summarization.

Comput. Linguist., 28(4):399–408, 2002.

[19] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL'2002*, 2002.

[20] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu. OpinionSeer: interactive visualization of hotel customer feedback. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1109–1118, 2010.

Appendices

- **SOURCE CODE:**

```
# -*- coding: utf-8 -*-
"""reviewcapstone.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1hA_gu7DGn0phsr-E6jSmiRenHhDs3aEA

## Sentiment Analysis of reviews from Healthcare Industry
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
# import dataset
dataset = pd.read_csv('mydatasetforcapstone.csv')

# Lets see how our dataset looks
# Dimensionality of the DataFrame-shape
print(dataset.shape)
# First few rows of our dataset
print(dataset.head())
# Full summary of the dataframe
print(dataset.info())
# Descriptive statistics of the dataframe
print(dataset.describe(include="all"))
# Rating counts
print(dataset['Rating'].value_counts())
print(dataset['Company'].value_count())

"""## Text Pre-Processing"""

# store the reviews & ratings in two separate lists
classes=dataset_class['Rating']
text_messages=dataset_class['Reviews']
```

```
print(text_messages[0:6],"\n",classes[0:6])
# Finding the most common and rare words
commonnew= pd.Series(' '.join(processed).split()).value_counts()[0:20]
print(commonew)
rarenew= pd.Series(' '.join(processed).split()).value_counts()[-10:]
print(rarenew)

# Representation of bag-of-words model
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
all_words=[]
for message in processed:
    words=word_tokenize(message)
    for w in words:
        all_words.append(w)
all_words=nltk.FreqDist(all_words)
for key,val in all_words.items():
    print (str(key) + ':' + str(val))
"""### WordCloud"""

import wordcloud

from wordcloud import WordCloud

def show_wordcloud(data, title = None):
    wordcloud = WordCloud(
        background_color = 'white',
        max_words = 50,
        max_font_size = 40,
        scale = 3,
        random_state = 42
    ).generate(str(data))

    fig = plt.figure(1, figsize = (20, 20))
    plt.axis('off')
    if title:
        fig.suptitle(title, fontsize = 20)
        fig.subplots_adjust(top = 2.3)

    plt.imshow(wordcloud)
    plt.show()

# print wordcloud
show_wordcloud(final["Reviews"])
```

LDA(Latent Dirichlet Allocation),LSA(Latent Semantic Analysis),NMF(Non-Negative MAtrix Factorization)

```
#new code for all

import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from gensim import corpora, models
import gensim

# Load dataset
data = pd.read_csv("mydatasetforcapstone.csv")

# Preprocess data
stop_words = set(stopwords.words('english'))
reviews_tokenized = [word_tokenize(review.lower()) for review in data['Reviews']]
reviews_filtered = [[word for word in review if word not in stop_words] for review in reviews_tokenized]

# Create dictionary
dictionary = corpora.Dictionary(reviews_filtered)
corpus = [dictionary.doc2bow(review) for review in reviews_filtered]

# Train LDA model
lda_model = gensim.models.LdaMulticore(corpus, num_topics=5, id2word=dictionary, passes=10, workers=2)

# Print topics and top words
for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx + 1, topic))

# Extract aspect and sentiment from reviews
for i, review in enumerate(data['Reviews']):
    review_tokenized = word_tokenize(review.lower())
    review_filtered = [word for word in review_tokenized if word not in stop_words]
    review_bow = dictionary.doc2bow(review_filtered)
    review_lda = lda_model[review_bow]
    aspect = max(review_lda, key=lambda item: item[1])[0]
    sentiment = "positive" if data['Rating'][i] > 3 else "negative" # Set sentiment based on rating
    print('Review: {} \nAspect: {} \nSentiment: {}\n'.format(review, lda_model.print_topic(aspect), sentiment))
```

```
# Extract aspect and sentiment from reviews
num_correct = 0
for i, review in enumerate(data['Reviews']):
    review_tokenized = word_tokenize(review.lower())
    review_filtered = [word for word in review_tokenized if word not in stop_words]
    review_bow = dictionary.doc2bow(review_filtered)
    review_lda = lda_model[review_bow]
    aspect = max(review_lda, key=lambda item: item[1])[0]
    predicted_sentiment = "positive" if aspect == 0 else "negative" # Set sentiment based on LDA
aspect
    actual_sentiment = "positive" if data['Rating'][i] > 3 else "negative" # Set sentiment based on
rating
    if predicted_sentiment == actual_sentiment:
        num_correct += 1

accuracy_lda = num_correct / len(data)
print('Accuracy: {:.2%}'.format(accuracy_lda))
```

```
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from gensim import corpora, models
import gensim
```

```
trpo, trne, fapo, fane = 0, 0, 0, 0
for i, review in enumerate(data['Reviews']):
    review_tokenized = word_tokenize(review.lower())
    review_filtered = [word for word in review_tokenized if word not in stop_words]
    review_bow = dictionary.doc2bow(review_filtered)
    review_lda = lda_model[review_bow]
    aspect = max(review_lda, key=lambda item: item[1])[0]
    sentiment = "positive" if data['Rating'][i] > 3 else "negative"
    if sentiment == "positive" and aspect == 0:
        trpo += 1
    elif sentiment == "negative" and aspect != 0:
        fapo += 1
    elif sentiment == "positive" and aspect != 0:
        fane += 1
    elif sentiment == "negative" and aspect == 0:
        trne += 1
accuracy_lda = (trpo + trne) / len(data)
precision_lda = trpo / (trpo + fapo)
recall_lda = trpo / (trpo + fane)
```

```
f_score_lda = 2 * (precision_lda * recall_lda) / (precision_lda + recall_lda)

print('Precision: {:.2f}%'.format(precision_lda * 100))
print('Recall: {:.2f}%'.format(recall_lda * 100))
print('F-score: {:.2f}'.format(f_score_lda))
```

```
from gensim.models.coherencemodel import CoherenceModel
# Compute coherence score
coherence_model_lda = CoherenceModel(model=lda_model, texts=reviews_filtered,
dictionary=dictionary, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('Coherence Score: ', coherence_lda)

# Compute perplexity score
perplexity_lda = lda_model.log_perplexity(corpus)
print('Perplexity Score: ', perplexity_lda)

# Compute topic coherence score
topic_coherence_model_lda = CoherenceModel(model=lda_model, texts=reviews_filtered,
dictionary=dictionary, coherence='c_v', topn=20)
topic_coherence_lda = topic_coherence_model_lda.get_coherence()
print('Topic Coherence Score: ', topic_coherence_lda)
```

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import NMF

# # Load dataset
# data = pd.read_csv("reviews.csv")

# Define number of topics and top words per topic
num_topics = 5
num_top_words = 10

# Preprocess data
vectorizer = TfidfVectorizer(max_df=0.95, min_df=2, stop_words='english')
tfidf = vectorizer.fit_transform(data['Reviews'])

# Train NMF model
nmf_model = NMF(n_components=num_topics, random_state=42, init='nndsvda', solver='mu')
nmf_model.fit(tfidf)
```

```
# Print topics and top words
feature_names = np.array(vectorizer.get_feature_names_out())
for topic_idx, topic in enumerate(nmf_model.components_):
    print("Topic #%d:" % topic_idx)
    top_words_idx = np.argsort(topic)[::-1][:num_top_words]
    top_words = feature_names[top_words_idx]
    print(" ".join(top_words))
    print("")

# Extract aspect and sentiment from reviews
for i, review in enumerate(data['Reviews']):
    review_tfidf = vectorizer.transform([review])
    review_nmf = nmf_model.transform(review_tfidf)
    aspect = np.argmax(review_nmf)
    sentiment = "positive" if data['Rating'][i] > 3 else "negative" # Set sentiment based on rating
    print('Review: {} \nAspect: {} \nSentiment: {}\n'.format(review, aspect, sentiment))

from sklearn.decomposition import NMF
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
import warnings
from sklearn.exceptions import ConvergenceWarning

# Suppress convergence warnings
warnings.filterwarnings('ignore', category=ConvergenceWarning)




# Extract aspect and sentiment from reviews
predicted_sentiments = []
for i, review in enumerate(data['Reviews']):
    review_tfidf = vectorizer.transform([review])
    review_nmf = nmf_model.transform(review_tfidf)
    sentiment = "positive" if data['Rating'][i] > 3 else "negative" # Set sentiment based on rating
    predicted_sentiments.append(sentiment)

true_sentiments = ['positive' if rating > 3 else 'negative' for rating in data['Rating']]

# Compute accuracy
sentiment_accuracy = accuracy_score(true_sentiments, predicted_sentiments)

print("Sentiment Accuracy:", sentiment_accuracy)
print("Sentiment accuracy:", sentiment_accuracy)
#for nmf model

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
import pandas as pd


df = pd.read_csv('mydatasetforcapstone.csv')

# Create a TF-IDF vectorizer
vectorizer = TfidfVectorizer(max_features=1000)

# Fit the vectorizer on the reviews
X = vectorizer.fit_transform(df['Reviews'])

# Create an LSA model
lsa = TruncatedSVD(n_components=5)

# Fit the LSA model on the TF-IDF vectors
X_lsa = lsa.fit_transform(X)



terms = vectorizer.get_feature_names_out()
for i, comp in enumerate(lsa.components_):
    print(f"Topic {i}:")
    terms_comp = zip(terms, comp)
    sorted_terms = sorted(terms_comp, key=lambda x:x[1], reverse=True)[:10]
    for t in sorted_terms:
        print(f"{t[0]}: {t[1]}")
    print('\n')

# Assign a sentiment score to each review based on its topic
df['Topic'] = X_lsa.argmax(axis=1)
df['Sentiment'] = df['Rating'].apply(lambda x: 'Positive' if x >= 4 else 'Negative')

# Calculate the sentiment distribution for each topic
sentiment_distribution = df.groupby(['Topic', 'Sentiment']).size().unstack(fill_value=0)
sentiment_distribution['Total'] = sentiment_distribution.sum(axis=1)
sentiment_distribution['Positive Percentage'] = sentiment_distribution['Positive'] /
sentiment_distribution['Total']
sentiment_distribution['Negative Percentage'] = sentiment_distribution['Negative'] /
sentiment_distribution['Total']
print(sentiment_distribution)
```

```
# Step 1: Create a list of predicted sentiment labels
predicted_sentiments = ['Positive' if score == 'Positive' else 'Negative' for score in
df['Sentiment']]

# Step 2: Create a list of true sentiment labels
true_sentiments = ['Positive' if rating >= 4 else 'Negative' for rating in df['Rating']]

# Step 3: Calculate accuracy score
from sklearn.metrics import accuracy_score
accuracy_lsa = accuracy_score(true_sentiments, predicted_sentiments)
print('Accuracy:', accuracy_lsa)
print('Accuracy: {:.2f}%'.format(accuracy * 100))
```

```
from sklearn.metrics import mean_squared_error
import numpy as np

# Calculate explained variance
explained_variance = lsa.explained_variance_ratio_
print("Explained variance:", explained_variance)

# Calculate reconstruction error
X_reconstructed = np.dot(X_lsa, lsa.components_)
reconstruction_error = mean_squared_error(X.toarray(), X_reconstructed)
print("Reconstruction error:", reconstruction_error)
```

```
import matplotlib.pyplot as plt
import numpy as np
# plot the bar chart
labels = ["Doc2Vec", "TF-IDF","LDA","LSA"]
accuracy_scores = [accuracy_doc2vec, accuracy_tfidf,accuracy_lda,accuracy_lsa]
x_pos = np.arange(len(labels))

plt.bar(x_pos, accuracy_scores, align='center', alpha=0.5)
plt.xticks(x_pos, labels)
plt.ylabel('Accuracy')
plt.title('Comparison of Different Models')
plt.show()
```