Topic Modelling Using LDA and Sentimental Analysis

A Main Project

Report

Submitted in the partial fulfillment of the requirements for

The award of the degree of

Bachelor of Technology

In

Department of Computer Science Engineering

By

Namburi Bhavana Lalitha Janaki 180030571

Repaka Surya Rasagna 180031143

Under the supervision of

DR.SASMITA PADHY

Department Of Computer Science Engineering

K L E F, Green Fields,

Vaddeswaram- 522502, Guntur (District), Andhra Pradesh, India.

November, 2021.

Declaration

The Project Report entitled " Topic Modelling Using LDA and Sentimental Analysis" is a record of bona fide work of Namburi Bhavana Lalitha Janaki , Repaka Surya Rasagna submitted in partial fulfillment for the award of B.Tech in Computer Science Engineering to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

Namburi Bhavana Lalitha Janaki 180030571

Repaka Surya Rasagna 180031143

Certificate

This is to certify that the Term Paper/Project Report entitled
"Topic Modelling Using LDA and Sentimental Analysis" is being submitted by Namburi Bhavana Lalitha Janaki, Repaka Surya Rasagna submitted in partial fulfillment for the award of B.Tech in Computer Science Engineering to the K L University is a record of bona fide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of Guide

ACKNOWLEDGEMENT

On the every aspect of this project report, We would like to sincerely thank Our guide and super visor and co super visor of our department for the support they have actually extended for us, we are very thankful for our department faculty and CSE dept. HOD prof.HARI KIRAN VEGE. I would like to extend my heartfelt obligation to all the persons who ever are with us during the entire project completion period. Without their guidance, help, encouragement which they have provided us during all the journey, we would not have actually made heading this way in the project.

We are very thankful and pay my gratitude to my faculty Dr.SASMITHA PADHY for her actual guidance and encouragement on the completion of my project in its presently.

We extend my sincere gratitude to K L UNIVERSITY for giving us such a great opportunity.

We would like to also acknowledge with deep sense of gratitude to our parents who have been with us during the entire journey supporting us morally and economically.

At last but not least I would like to thank my friends and some of the faculty helping me directly or indirectly for the completion of my project.

Any omission in this acknowledgement does not mean any lack of our gratitude.

Thanking You,

Namburi Bhavana Lalitha Janaki 180030571

Repaka Surya Rasagna 180031143.

ABSTRACT

Topic Modelling using LDA and Sentimental Analysis is the Project entitled as. Coming to the Project implementation the Project encloses of 3 Modules, module 1 deals with the topic modelling using LDA whereas module 2 and 3 deals with the sentimental analysis. The 2 main algorithms which are used in this project are Latent Dirichlet Allocation (LDA) and Naïve Bayes Algorithm.

Discussing about the module 1 about topic modelling using LDA algorithm, topic modelling is a process in Natural Language Processing (NLP) which is generally used to train the machine learning models to attain the actual required results for the targeted objective.

The process of selecting the words logically that actually belong to certain topic within a given document. This Topic Modelling provides a very great time and also effort saving benefits. Coming to LDA algorithm this is process of the topic model and which is used to classify text in a document to a particular topic it provides the information on the topic which is document is actually based on.

To explain briefly regarding the LDA algorithm, it builds a topic for each document model and the respective topics for the topic model, which is modeled as Dirichlet distributions. The main core concept will be actually replaced by the Dirichlet allocations where the distribution is mainly over a probability simplex.

We have taken a dataset comprising of different emails and concluded by identifying the various topics which is actual dataset is comprising of. With this we can easily identify what the mails are talking about. So like this we can work on large datasets to identify the topics hidden.

Module 2 and 3 is about sentimental Analysis It is automated method process of translating the large volumes of the data which is unstructured into the most qualitative data to uncover the hidden patterns and the emotion of the data this is mostly used to study the data of social media, The main role is identifying opinionative data, it is used for the computational study of text analysis to find the subject and emotion hidden.

CONTENTS OF THE REPORT

1. INTRODUCTION

Topic Modelling Using LDA and Sentimental Analysis is the project which is completed, Here by the 2 main pillars of this project is LDA Algorithm and Naïve Bayes Algorithm for the Topic Modelling and Sentimental Analysis.

These two are respectively implemented with the given algorithms, for topic modelling we do generally have three most common techniques used:

1. Latent Semantic Analysis (LSA)

2. Probabilistic Latent Semantic Analysis (PLSA)

3. Latent Dirichlet Allocation (LDA)

Being topic modelling a powerful technique for the most of the documents for unsupervised Analysis of the large document collections.

The large document collections are mostly unstructured data this is where the big data analytics comes into the picture.

It is very easy to use the large data sets to gain the insights by using topic modelling algorithm in further discussions it will be clearly pictured by the outputs and the procedure the algorithm is actually implemented.

So the topic models have a very wide range of the applications like recommendation of the tag, categorization of text, extraction of the keywords, and the most important one the similarity search in the broad fields of the areas of text mining, for the information retrieval and all.

The topic Modelling can conceive different Latent topics which are generally used in text using the different random variables and the structure of the entire procedure is with the posterior inference.

1.1 What is a Topic model?

It is the one that does automatically discovers the topics occurring in a collection of the given documents. A trained Model that which is used to discern with different topics in the new documents. The model actually picks out different portion of the topics which covers topics in the documents

When we generally consider the large data such as Wikipedia and different huge data they all are combination of several topics millions of documents covering some thousands of the topics

By using the topic modelling we can also discover some of the emerging topics as the documents can be written about them, considering news documents which keeps on being

in a constant change with several topics the topic modelling can be used to identify the topics which the newspaper is actually talking about.

1.2 Latent Dirichlet Allocation

The Algorithm LDA is the first pillar of this project introduction about this we will be able to get the actual advantages of applying this algorithm.

This LDA approach it involves building the different statistical models of the topics and the documents.

For example let us assume a topic to be modeled on the basis of the probability distribution over a given fixed set of the words.

This helps in actually formalizing the different set of words that come to the mind which are referred to the specific topic.

In this following way it helps the topics of the different document covers.

The main goal of the learning and understanding the LDA a methodology is that it helps in discovering from a corpus of documents, for the good number of distributions of the various topics, for the good number of the topic proportions in different kinds of documents.

1.2.1 Parameter for the LDA?

The most important parameter for the LDA is the number of topics.

Yes that's it this is one of the very important parameter that is to be passed along with the input that is the input dataset.

1.2.2 LDA Latent Dirichlet Allocation

LDA generally represents the documents as the different mixtures of the topics that do spit out words with the certain different probabilities.

Let us consider a desired number of topics with the set as a 'k' in the dimensional Dirichlet distribution.

We can define the LDA algorithm as the proven oone which actually delivers accurate results for the topic modelling use cases.

1.2.3 How does it actually work?

Step 1: Decide the number of words which are of N the document will have based on the Poisson distribution

Step 2: now you have to choose a topic mixture for the document (According to the Dirichlet distribution over a fixed set of already defined k topics should be taken)

Step 3: Generation

Here we have to generate each word Wi in the document by:

1. Picking up the a defined topic
2. Using the topic to actually generate the word itself (based on the topics multinomial distribution)

By the following ways assuming this generative model of different collection of the documents, LDA then concludes on trying to start backtracking the documents to find a good set of topics that are likely to be present in the generated collection.

1.3 Sentimental Analysis

Coming to the sentimental analysis it is one of the automated method process which is used for translating the large number of volumes of the mostly unstructured text into the qualitative data which is mostly used to discover the proper patterns, insights, trends of the data.

The sentimental Analysis is spoken out of computational study of text analysis to find actually and then extract the subjective study of the data or the big data.

It is used to analyze the different users review from the social media

For example let us consider the amazon reviews of the products, initially considering a product with specific reviews as the input data by performing different computational methods of sentimental analysis.

We would get an output of the emotions which are actually given in the form reviews on observing the result after performing computational study we will be able to understand the review of the product is it very good or good or bad or very bad,

Based on that the recommendations of the products is computed for the customers.

It is used to analyze the users review from the social media.

1.3.1 Sentimental Analysis for Social media

In the social media the role of the sentimental analysis is generally identifying the opinionative data, what is the opinion behind the data the, emotion, the opinion, the intention these are all the keywords to describe how the sentimental analysis actually works on the for the social media data.

For example let us consider a twitter data set, taking the recent pandemic into consideration covid-19 tweets there were millions of tweets which are tweeted by the citizens expressing different kind of emotions on the pandemic.

When these tweets are taken into consideration and if the sentimental analysis study is performed.

We will get different plots using different functions we can use different ways to understand the data the best way is the word cloud which gives the output in such a way that the most repeated topic in the word cloud is displayed bigger, by this we can conclude which topic is being more discussed on.

In the different ways usually we can also understand the real hidden meaning behind the tweet, if he is angry because of the pandemic, or feeling sad because of this kind of situation occurred, or he is happy for staying at home different emotions are seen in each tweet finding the emotion behind the data is exactly what is called the sentimental analysis.

They are definitely some drawbacks sure on the sentimental analysis for sure, more over using this analysis by keeping in mind in when and where to use provides the accurate output than anything.

1.3.2 Data processing for SA

The key concepts for processing a data is to:

1. Understand the data by studying it.
2. Get to know what kind of qualities does it possess
3. Advantages and disadvantages of the data usage method
4. Pick up the models which produces accurate results of the data

According to me processing a data with accuracy is a beautiful art only when you start understanding the data the way it is when you know where and when to use the data you have almost completed the project the balance is just the implementation.

This way we should put in more time and efforts to understand the data and the pick models to get accurate results of the data.

Sentimental Analysis is a problem which is based on the text analysis this entire whole project is based on the text analytics whether it be topic modelling or sentimental analysis.

The SA (Sentimental Analysis) is used in web classifying them according to their polarity which is used.

The main is that the problem is based on the text analysis.

It does helps in the hidden sentiments and the poly seamy of the data.

2. LITERATURE SURVEY

The literature survey for this project is conducted on the based on the topic modelling and Sentimental Analysis.

1. TOPIC MODELLING

2.1.1 Introduction

We have been seeing a lot continuous development of the IT (Information Technology) branch of science, based on information data as the internet is going on increasing in a rapid manner. The major news websites and channels have become the most important platform for the citizens to get the news or the information right away, however the data is being rapidly increasing in the most unstructured format day by day.

The data should always be preprocessed before using it, data being in an unstructured format will be very tough to understand using traditional data classification techniques, because they are unable to meet the requirements of the data when they are observed.

So the research area working on the area of the text mining, which would help news or data text classification to meet the requirements, which should be able to classify the text as fast as possible and also it should be in a position to handle the dataset, and try to make the accurate prediction of the data. So the automatic classification can help to complete the proper function which apt to predict the accurate results.

It should be highly effective with higher efficiency so that it helps the organization to save the expenses, in the modern era of the big data, the research on automatic text classification plays an increasingly important role.

Many of the classic text classification algorithms are mostly proposed and they being widely used all the way.

For example when we consider the naïve Bayes, k-nearest neighbor and decision tree these are different classifiers of the text.

According to me each classifier do have different strengths and weakness. They are dependent on the decision step they take towards. The way the decision is being made by the classifier defines its strength overall.

Due to the topic dimensionality of the different texts sometimes it is too high this the reason where we generally highlight using the topic modelling. This survey deals with the text classification usage very primarily using the topic modelling using LDA and also the sentimental analysis.

2.1.2Text categorization

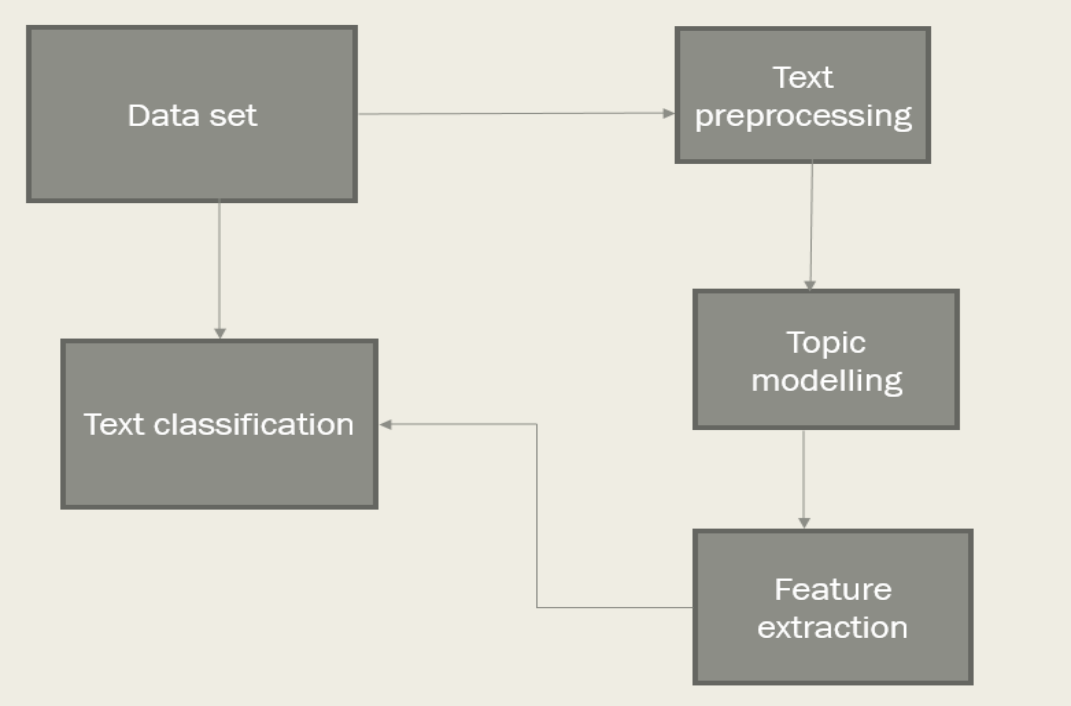


Fig 2.1

2.1.3 Latent Dirichlet Allocation (LDA)

LDA is a kind of topic model algorithm which is based on probability model, the algorithm deals with plurality of the topic mixture. It does uses the different ways which potential enough to find the hidden topic or the information of the topic in the very large scale of the document set.

The algorithm do assumes each and every word into the corpus of the given information through certain kind of probability to choose a particular topic from the chosen subject with a certain to select a word. The way it does actually work is that it does chooses a topic from the given topic distribution and chooses the specific word from the entire word in the distribution

The whole main idea of the algorithm is that is simply selects a topic vector and topics that determine the probability of the data that is being selected.

But the SLDA uses corresponding continuous response values using the algorithm of linear regression, which surely cannot be the text data which is of the multi class text data as the very data which is taken as input.

In simple words the topic modelling is the unsupervised learning method where as he text classification is the supervised learning method.

2. SENTIMENTAL ANALYSIS

2.2.1. Introduction

The sentimental analysis works on the text based analysis which is most important part of the entire project is Natural Language Processing (NLP). This NLP is entirely used for the detection most of the time. Analysis and the mining are the 2 subjective portions of the text which do contain the views and also the emotions, preferences and the intentions of the user. The field of the sentimental analysis is a very interesting field in the research area.

This sentimental analysis is a very big branch in the field of the natural language processing, The entire text analysis is based on how the text is categorized and that has great influence on the natural language processing, being most disciplinary field in the entire research it had been more concentrated in the research are by the most of the scholars everywhere.

This sentimental analysis field not only involves the NLP (Natural Language Processing) filed but also many more computational areas, machine learning and many more algorithms that are part of the artificial intelligence.

The most 3 main methods of this area are the analytical methods, the sentimental analysis as said will be based on the machine learning algorithm and the deep learning algorithm.

Coming to the point that the sentimental analysis which is also called as the opinion mining is the method in which it automatically finds the opinion or the intention behind the text, but this could be one of the really challenging area in the research area, mainly in the data mining field for the Social media.

We also have to look into the issue that how the social media will play the central part of this sentimental analysis, we know that whatever happens the social media reflects the society opinion for any kind of incident not only the news is given but also the entire people opinion on the incident can also be –observed, opinion on certain issues can be gathered, with this kind of expeditious development of the cycle web 2.O, people started increasing in showing their opinions on the people, this way day by day the handling odf this entire unstructured data does become complex.

2.2.2 Sentimental analysis Approaches

When generally this is formally stated by their task and is interpreted that how to mathematically inject the social media context and the topic context in the basic given prediction model. This shall be investigate accordingly using different kinds and ways of the correlations among the reserved topics and calculated to measure them as required.

The assumptions about the given entire social media data or the data context and topic a context were both ways incorporated by the hypothesis testing the results are extracted from the social media data.

1. LDA (Latent Dirichlet Allocation)

2. S-PLSA (Sentimental probabilistic approach)

3. ARSA (Auto regressive sentiment and quality aware model)

2.2.3 Methodology

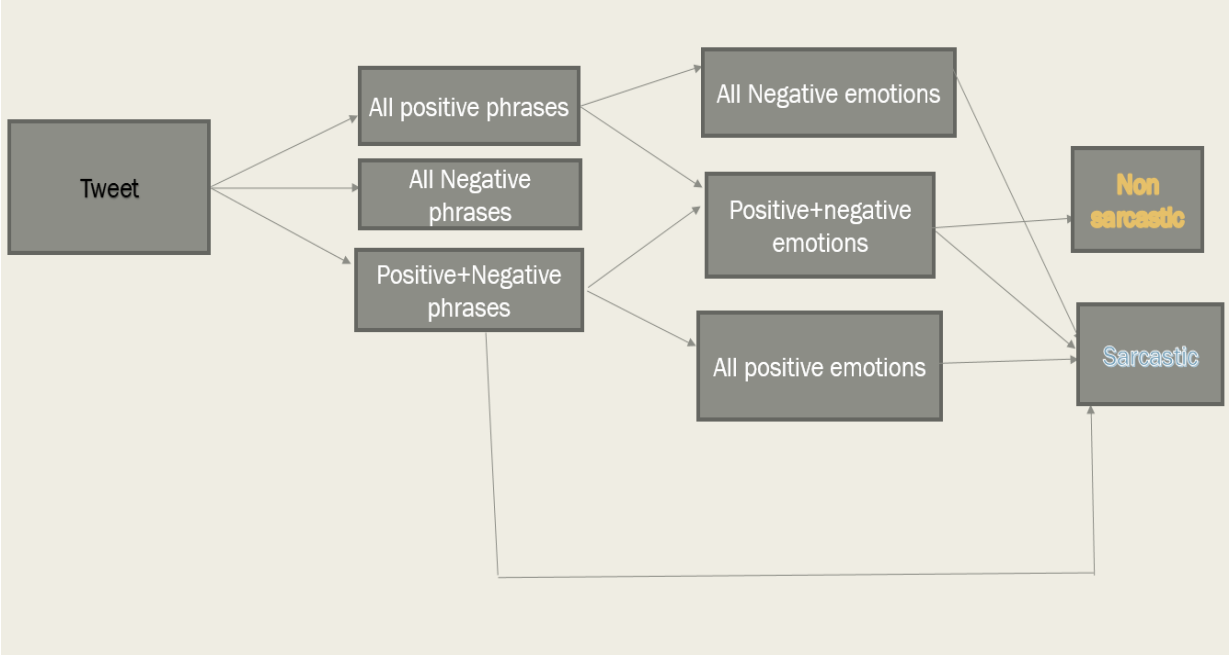


Fig 2.2

Whatever the tweet can be a positive phrase a negative phrase or there are also chance of It being a combination of both mixed emotions, This sentiment analysis is used to identify sarcasm which may exist actually which is contrast to the positive or the negative sentiments that are being given as the input.

There are different challenges that are associated with the sentimental analysis one of the most important one among the is the sarcasm detection.

The problem with this sarcasm is that it does not actually convey the meaning of the given phrase it does not show the pure or the actual intention. What does sarcasm actually mean is that a positive phrase may actually have the negative intention and vice versa this is where sarcasm is all about.

So in identifying the entire meaning of the phrase there may be a misunderstanding due to sarcasm it is actually complicated when the sentimental analysis faces this kind of challenges

3. THEORATICAL ANALYSIS

3.1 Topic Modelling

3.1.1 Short introduction

The Topic modelling which is a text mining technique generally deals with underlying and the hidden topics in the large documents, besides it also enable the one for the cluster documents on the thematic similarity. This topic identification is always achieved on the each document which ids formed by the generative process.

3.1.2 Different Topic models

We do have different kinds of topic models which are listed below

1. Mixture of multinomial

2. Gamma Poisson

3. Latent Dirichlet Allocation

4. Probabilistic Latent Semantic Analysis

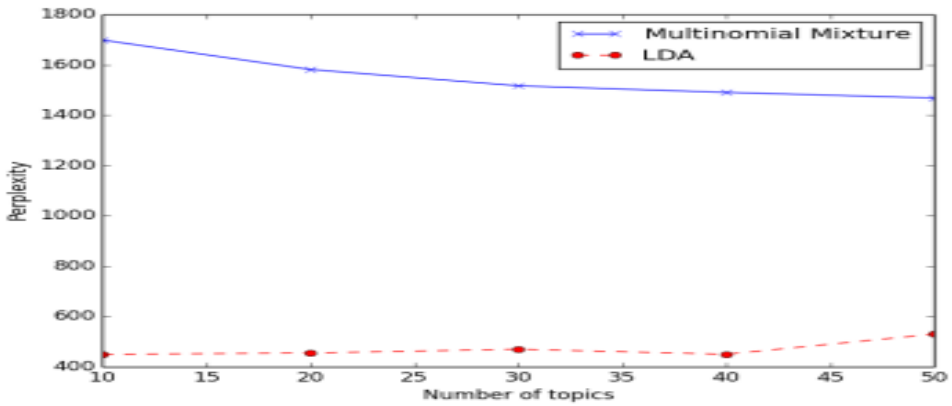
These are different approaches for the topic modelling there is such arise for the social media services which are twitter and Facebook etc. In comparison to long text short text are generally taken into the considerable for the problem when applying the different traditional topic models.

When short texts are considered as far as concerned the LDA model is not much applicable for this, the accuracy of the performance is too low.

Whereas the MM model will show accurate results when short text is considered it is a opposite to the performance of that of LDA.

In this way there many advantages and also disadvantages of the different topic models.

When a data with the corpus is considered the resultant output is:



3.2 Sentimental Analysis with Sarcasm Analysis

3.2.1 Introduction

This sentimental analysis can be defined as the classification task performed on text data with a significant preprocessing it to classify into mainly with different classification tasks they are binary or multi classification classes. The sentimental analysis is actually an operation that comprises of different computational tasks entirely that leads to statistical approaches in some directions.

It is one of the powerful machine learning application based on classifying the data or the text input into different classes, it is actually used mostly in analyzing the reviews such as customer reviews on a product, find the sentiment behind the review or the polarity of the information.

Exactly when come to the point of how does sentimental analysis in recent times the latest technologies under the NLP (AI AND ML). Sentimental analysis is not just used for the social media data accordingly it is used for different applications such as recommendation system and feedback analysis. In this theoretical analysis we will be able to evaluate the predictions of sentiment classifiers, attached with different cases. When it is briefly taken into the context where the complexity of the data emerges neural network classifiers provides us the very reasonable way of showing us the high accuracy with efficient results.

3.2.2 Motivation applied for the sentimental Analysis

In the modern era of the big data streaming in our daily lives, each and every person is connected with the social media in any way, so when we observe into this picture we could figure that there is lots of data that is being generated in any manner so, there should be something in a very strong position that can be able to handle such a big data.

The big data analytics plays a very crucial in our daily lives, more over when it is concerned social media data it is very complicated to analyze and calculate such data sentimental analysis as we discussed for sure shows us that a very form of tweets, posts, matter, photos, videos, is being streamed online All this raw text or unstructured data can be extracted and can be processed as far as the text is concerned there are specific algorithms and analysis that can be applied to understand the results.

We can analyze the data in different formats some of the aspects are mentioned below:

1. Brand monitoring of the product

2. Product analysis

3. Market and the research analysis

4. The recommendation system

5. Social media monitoring

3.2.3 General Work Flow of Sentimental Analysis

For every analytic task starts with the collection of the data. These days there are different social media platforms that are available such as twitter, Facebook, instagram and many more. These usually provide us very easy and wide open way to collect the data and use it for the preprocessing and working with the data. Real time twitter data the tweets can be extracted for the analysis and also with one twitter developer account and the tweepy which is one of the library in the python helps us to work accurately.

Amazon reviews of the product that can be extracted by different techniques.

We do have a specific flow for working with the sentimental analysis it is explained below:

Step 1: Data collection

Step 2: Preprocess the data

Step 3: Feature extraction and selection

Step 4: Sentimental classification

Step 5: Evaluating Results and Error analysis

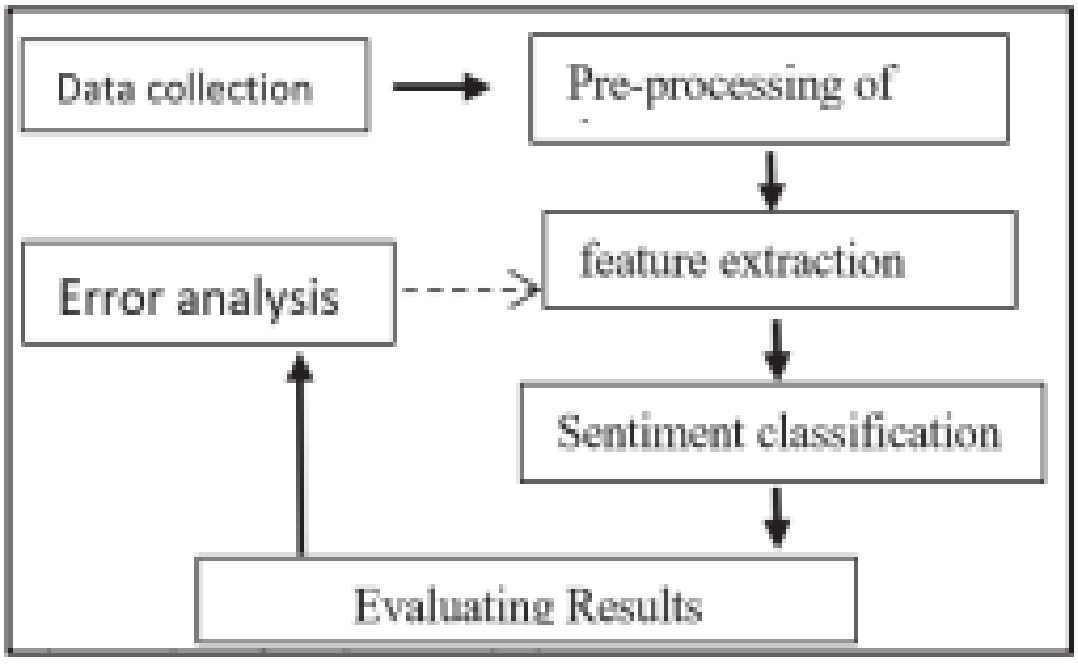


FIG 3.1

3.2.4 Sarcasm Analysis

Sarcasm is a nuanced form of the pronunciation or the way of communication where the actual intention behind the individual statement is opposite of what he/she have stated. What is actually implied is different from what actually is stated.

There are major challenges of this sarcasm nature because of its different behavior or the ambiguous nature. There is no prescribed definition for the word sarcasm actually it is just opposite to the individual statement.

They are different slang words that are being created these days on the social media which are used on these sites. There are thousands of words that are being created on the social media each word behind with different intentions.

Existing of the corpus with different negative and positive sentiments should also be considered because can actual mean something which is different from in there. They may not actually prove to be accurate in detecting the sarcasm of the text or the data.

There are different difficulties and the tricky nature associated with this sarcasm and generally ignored during the social network analysis.

The sarcasm detection that poses to be one of the most critical problems that should be considered as far as concerned which we usually need to overcome.

These NLP based systems are that which supports the text summarization and the sentimental analysis

Sarcasm can be simply defined as the

“Positive sentiment attached to the negative situation”

Steps present in this methodology:

1. Data extraction and cleaning
2. Seeding
3. Lexical classifier
4. Machine learning classifier
5. Emoticon extraction
6. Pos-neg recognition
7. Pragmatic classifier
8. Results

3.3 NLP Natural Language Processing of sentimental analysis based on project

The main aim of this sentimental analysis is used to detect the polarity of the given text or the input.

The misinterpreting of this sarcasm also poses a big challenge.

Natural language processing (NLP) is a subfield of the computer science to be defined precisely. Where the artificial intelligence is concerned with the interactions. These brand monitoring systems or the NLP systems have been built in such a way that they rely on the social media data such as the tweets posted on these social networks.

We will employ some of machine learning algorithms such as

1. Weighted Ensemble

2. Random Forest

3. Logistic Regression

4. Naive Bayes

Random Forest in an ensemble learning technique that operates by constructing multiple decision trees by splitting the training dataset into smaller ones and using each part for each tree. The output class is the mode or the mean average of each of the tree

Voted Ensemble Method: Voting Ensemble based learning comprises of different machine learning classifiers. The output class is predicted by the weighted ensemble classifier by taking the average mean or mode of each of the individual comprising classifier. The weightage is given to each of the comprising classifier according to their individual accuracy. It uses different classifiers: 1. Naive Bayes

2. Logistic Regression

3. Random Forest.

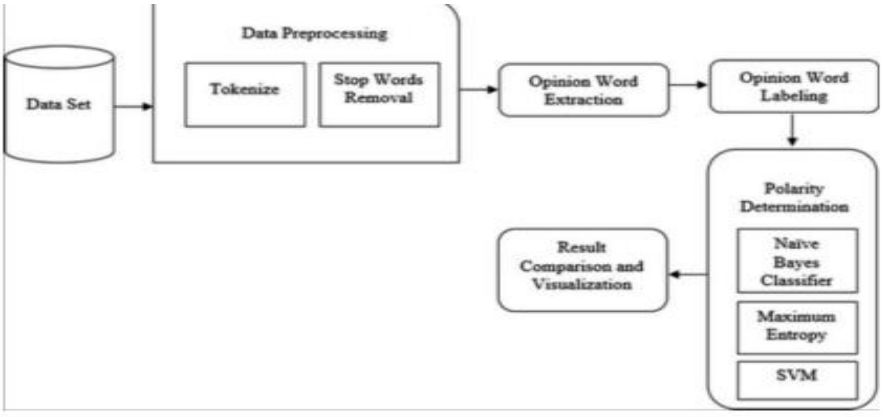


FIG 3.2

3.5 Multinomial Naïve Bayes

Multinomial Naïve Bayes it is one of the best approach in the context of the given text classification. This is generally fast, Reliable and that is better than the other classification algorithms with respective to the speed and also the accuracy. It just works on the basic simple concept of probability. Calculating Probability of each word in each document and the calculation part of the statistics.

Let us now discuss about the multinomial Naïve Bayes algorithm steps

Step 1: Training

Step 2: Read the preprocessed data

Step 3: Now we should create the empty dictionaries for both positive and negative

Step 4: In the document for each word in each document of the dataset to be reverted

Step 5: If the word that does not exist in the dictionary we should start adding it to dictionary

Step 6: if the word exist we should start incrementing the value by one

Step 7: Calculate the entire unique words in the given training dataset

Testing Phase:

Each word in testing document we should start calculating probability of that word with respect to both the given classes. Count is the count of the word with respect to class c and Σ count (w,c) count of all the words in the respective class. |V| is number that contains no of all the unique words in the dataset. Now we should start taking the product of all the obtained probabilities along with the all the previous calculated above for both the given respective classes.

Now we should classify the document with the class that has higher probability that is found when compared to all the other given respective classes.

Result:

Accuracy of the result:

Accuracy of naïve Bayes is nearly to 81.4%. This is one of the good accuracy for the given text classification problems due to the presence of the noise that is present in the datasets that cannot be removed easily. The Time taken for this MN Naïve Bayes algorithm is nearly to15 to 20 seconds. With the increase in dataset its accuracy increases.

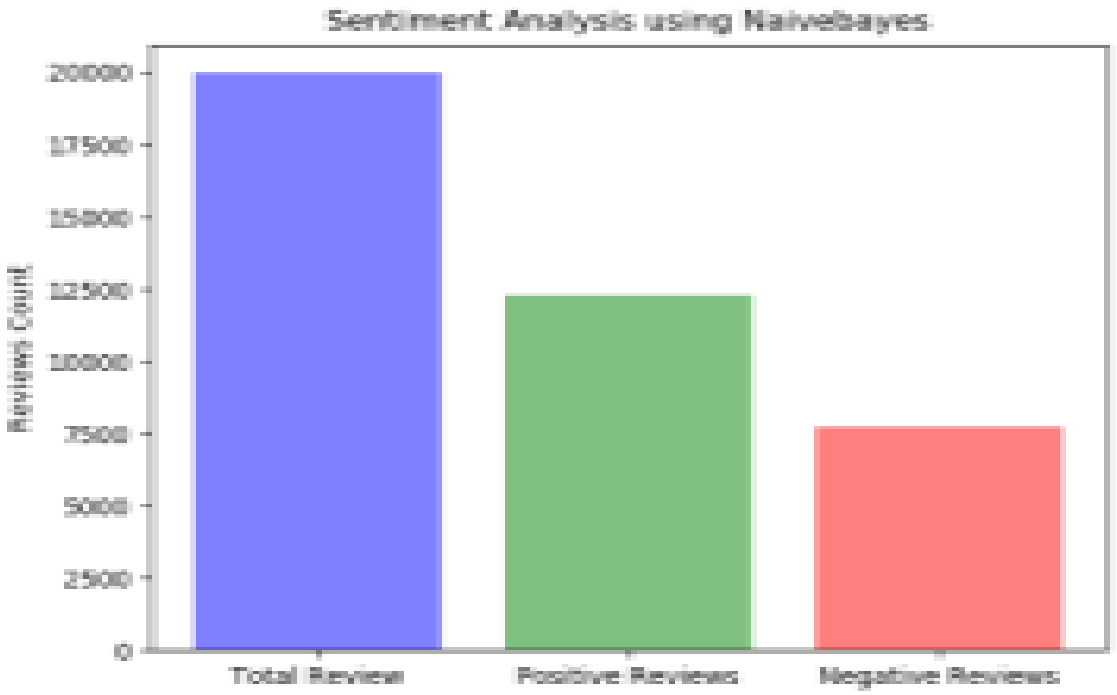


FIG 3.3

The above figure is of total 3 main contents

1. The positive reviews
2. The negative review
3. The total review

By using the naïve Bayes algorithm we have obtained the above result.

4. EXPERIMENTAL INVESTIGATION

REQUIREMENTS OF THE EXPERIMENT/ IMPLEMENTATION

1. The platform that is used to implement the code is KAGGLE

2. The language in which the experiment is carried out is PYTHON

3. The data sets that are used for the project code implementation are:

    1. https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json

    2. Santander customer transaction prediction

    3. Sentiment Analysis on Movie reviews

These are the 3 main data sets we have worked on

Module 1: Based on the topic modelling using LDA
Module 2: Prediction using Naïve Bayes algorithm
Module 3: Sentimental Analysis using Naïve Bayes and also different plot representations

4.1 KAGGLE ENVIRONMENT

Kaggle environment, is one of the best environments for a data analyst or data science student to work with.

It made me very easier to personally connect with data, and implement the data,

It is one of the friendly environment for the developer there are many datasets to work with-it is online community for the people who work with data science as well as machine learning practitioners.

Kaggle environment helps in offering the customizable environment for the developers, and also get chance to access Free GPU's and large storage of community to publish and code.

4.2 PYTHON

I have implemented the code in python language, because it helps in using different packages which it has such as NLTK (Natural Language Tool Kit) libraries and different libraries which are very important for us to implement this project code.

Different packages used in python language:

1. matplotlib.pyplot
2. tensorflow
3. unidecode
4. nltk
5. `genism`
6. `pyLDAvis`
7. `logging`
8. sklearn
9. word cloud

These are not all the packages that are used but these are very important packages used to complete this project.

4.2.1 Terminology

We would also like to discuss some of the important areas which we have implemented in our project:

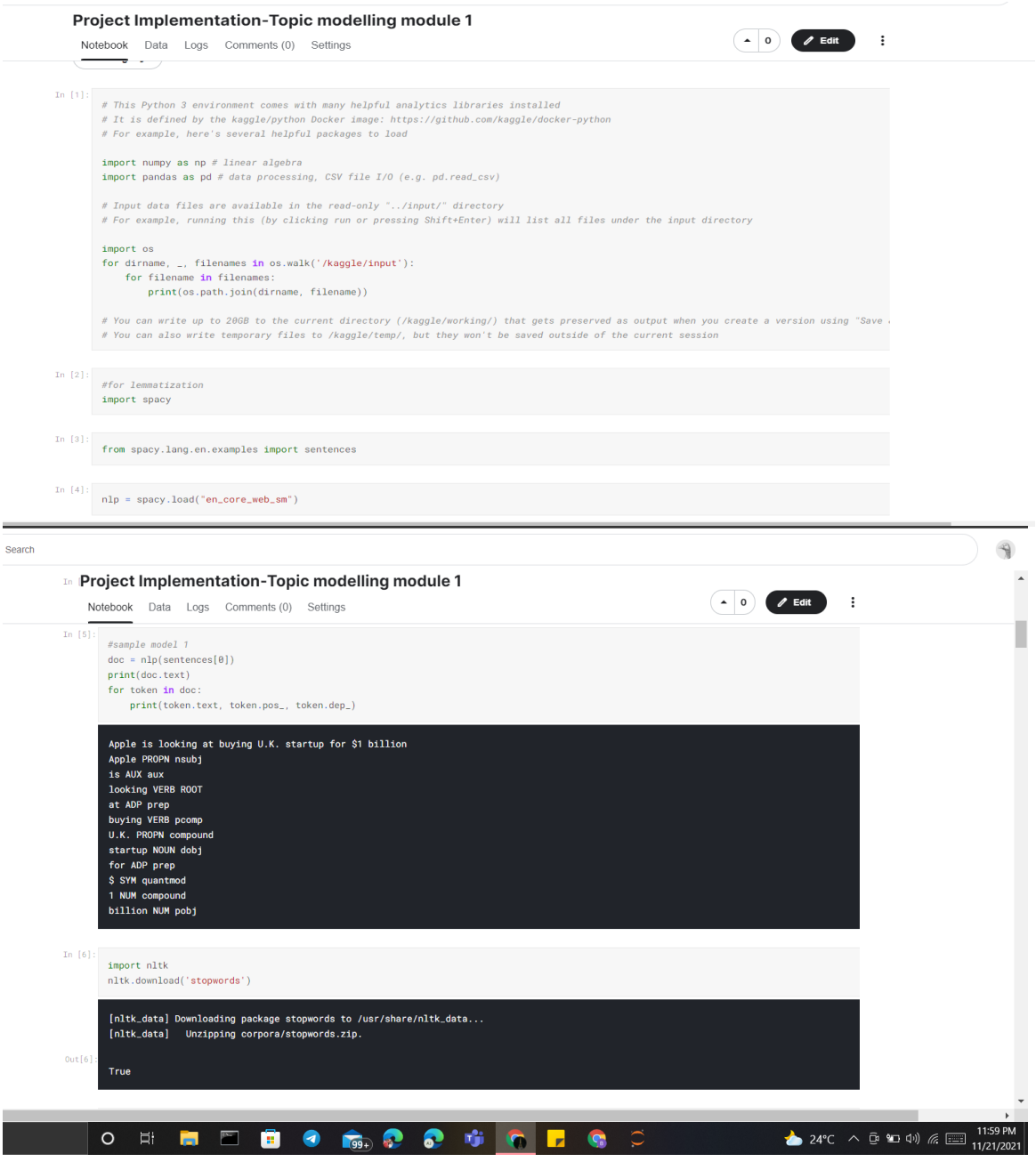
1. Word cloud is a data visualization technique this is very helpful for showing the results of the data based on its size here the size of the word is directly proportional to the frequency or most repeated theme of the data set.
2. K-nearest neighbor algorithm this is one of the very easier machine learning algorithm which is preferably based on the supervised learning technique. This usually assumes the similarity between the data and the available state and it puts it into the new state, the most similar are available in the same state.
3. KNN (Euclidean distance) this is used most widely and it is very popular which is most probably set to default in the sklearn KNN classifies library in the python language. It is generally used to calculate the distance between nearest neighbors.
4. Target Density and Target Probability:
   Target Density is one of the down sampling method which is used to discard the events based on their local density, the main goal is to ensure the ultimate density of the proportion which is considered as the sample should fall in the considered range of density.
   Target Probability it specifies the probabilities that actually determines the elimination of the trail in this the general reasonable values ranges from 0.5 to 0.6.
5. Probability describes how likely an event is to occur just simply defines probability, it is another word of the possibility of certain event to take place.

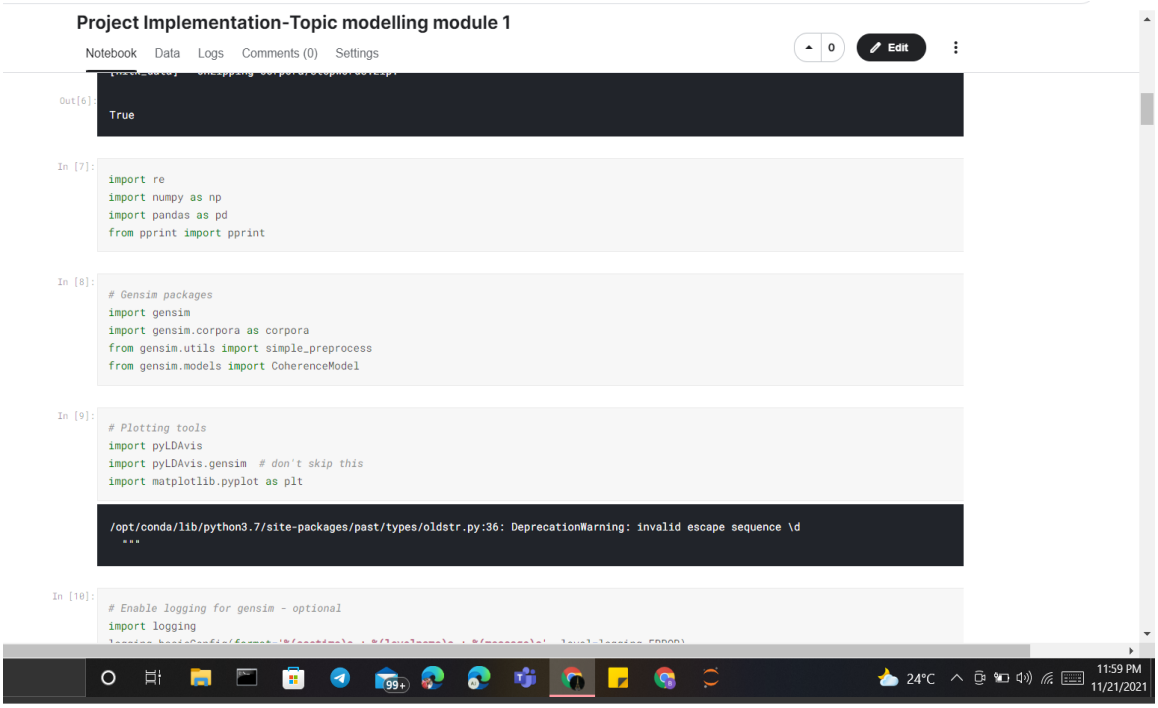
6. Z-score helps us to know better how far the mean is form the certain data point, it is simply used to describe the values relationship with the values of mean

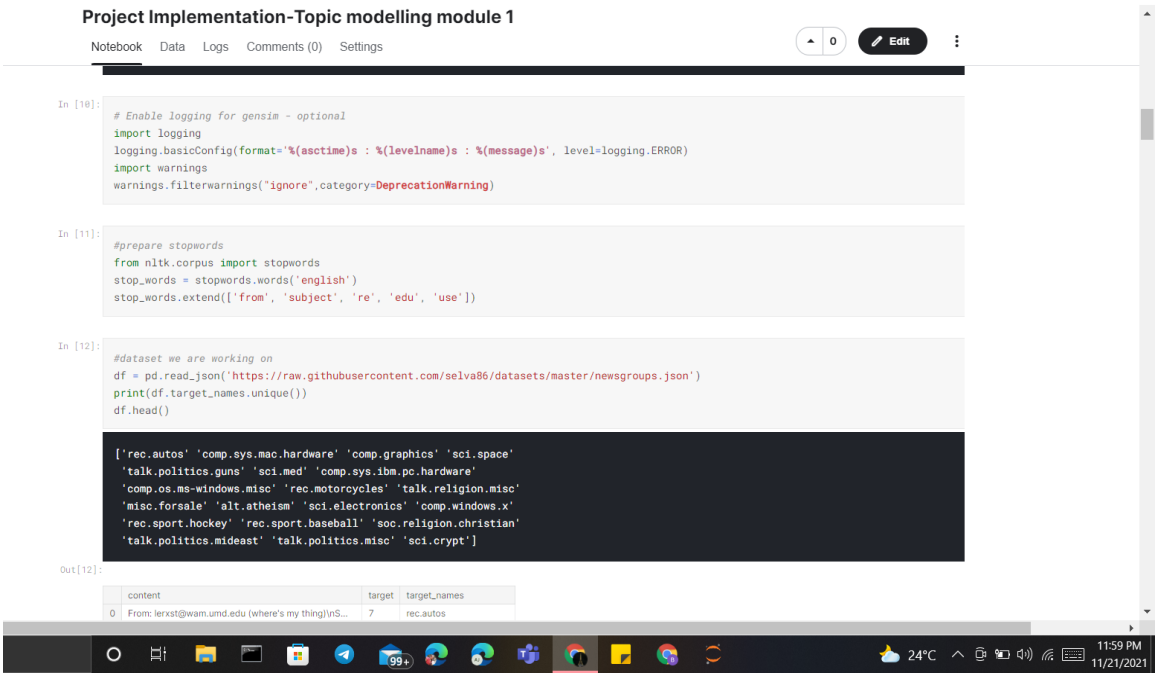
5. EXPERIMENTAL RESULTS

In this Chapter we would like to share the implementation details of our project:

5.1 MODULE 1 IMPLEMENTATION




Project Implementation-Topic modelling module 1
Notebook Data Logs Comments (0) Settings
0
Edit
Out[6]:
True
In [7]:
import re
import numpy as np
import pandas as pd
from pprint import pprint
In [8]:
# Gensim packages
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
In [9]:
# Plotting tools
import pyLDAvis
import pyLDAvis.gensim # don't skip this
import matplotlib.pyplot as plt
/opt/conda/lib/python3.7/site-packages/past/types/oldstr.py:36: DeprecationWarning: invalid escape sequence \d
"""
In [10]:
# Enable logging for gensim - optional
import logging
24°C
11:59 PM
11/21/2021


Project Implementation-Topic modelling module 1
Notebook Data Logs Comments (0) Settings
0
Edit
In [10]:
# Enable logging for gensim - optional
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.ERROR)
import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)
In [11]:
#prepare stopwords
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
In [12]:
#dataset we are working on
df = pd.read_json('https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json')
print(df.target_names.unique())
df.head()
['rec.autos' 'comp.sys.mac.hardware' 'comp.graphics' 'sci.space'
'talk.politics.guns' 'sci.med' 'comp.sys.ibm.pc.hardware'
'comp.os.ms-windows.misc' 'rec.motorcycles' 'talk.religion.misc'
'misc.forsale' 'alt.atheism' 'sci.electronics' 'comp.windows.x'
'rec.sport.hockey' 'rec.sport.baseball' 'soc.religion.christian'
'talk.politics.mideast' 'talk.politics.misc' 'sci.crypt']
Out[12]:
content
target
target_names
0
From: lerxst@wam.umd.edu (where's my thing)\nS...
7
rec.autos
24°C
11:59 PM
11/21/2021

Project Implementation-Topic modelling module 1

Notebook Data Logs Comments (0) Settings

0 Edit

In [14]:
```
#Tokenize words and Clean-up text
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True removes punctuations

data_words = list(sent_to_words(data))

print(data_words[:1])
```

```
[['from', 'wheres', 'my', 'thing', 'subject', 'what', 'car', 'is', 'this', 'nntp', 'posting', 'host', 'rac', 'wam', 'umd', 'edu', 'organ
y', 'of', 'maryland', 'college', 'park', 'lines', 'was', 'wondering', 'if', 'anyone', 'out', 'there', 'could', 'enlighten', 'me', 'on',
w', 'the', 'other', 'day', 'it', 'was', 'door', 'sports', 'car', 'looked', 'to', 'be', 'from', 'the', 'late', 'early', 'it', 'was', 'cal
he', 'doors', 'were', 'really', 'small', 'in', 'addition', 'the', 'front', 'bumper', 'was', 'separate', 'from', 'the', 'rest', 'of', 'th
'is', 'all', 'know', 'if', 'anyone', 'can', 'tellme', 'model', 'name', 'engine', 'specs', 'years', 'of', 'production', 'where', 'this',
'history', 'or', 'whatever', 'info', 'you', 'have', 'on', 'this', 'funky', 'looking', 'car', 'please', 'mail', 'thanks', 'il', 'brought'
'your', 'neighborhood', 'lerxst']]
```

n [15]:
```
#now to proceed with the project we need to actually create the bigram and trigram models
# Build the bigram and trigram models
bigram = gensim.models.Phrases(data_words, min_count=5, threshold=100) # higher threshold fewer phrases.
trigram = gensim.models.Phrases(bigram[data_words], threshold=100)
```

n [16]:
```
# Faster way to get a sentence clubbed as a trigram/bigram
bigram_mod = gensim.models.phrases.Phraser(bigram)
trigram_mod = gensim.models.phrases.Phraser(trigram)
```

Project Implementation-Topic modelling module 1

Notebook Data Logs Comments (0) Settings

0 Edit

In [17]:
```
# See trigram example
print(trigram_mod[bigram_mod[data_words[0]]])
```

```
['from', 'wheres', 'my', 'thing', 'subject', 'what', 'car', 'is', 'this', 'nntp_posting_host', 'rac_wam_umd_edu', 'organization', 'unive
and_college_park', 'lines', 'was', 'wondering', 'if', 'anyone', 'out', 'there', 'could', 'enlighten', 'me', 'on', 'this', 'car', 'saw',
y', 'it', 'was', 'door', 'sports', 'car', 'looked', 'to', 'be', 'from', 'the', 'late', 'early', 'it', 'was', 'called', 'bricklin', 'the'
'really', 'small', 'in', 'addition', 'the', 'front_bumper', 'was', 'separate', 'from', 'the', 'rest', 'of', 'the', 'body', 'this', 'is',
f', 'anyone', 'can', 'tellme', 'model', 'name', 'engine', 'specs', 'years', 'of', 'production', 'where', 'this', 'car', 'is', 'made', 'h
ever', 'info', 'you', 'have', 'on', 'this', 'funky', 'looking', 'car', 'please', 'mail', 'thanks', 'il', 'brought', 'to', 'you', 'by',
d', 'lerxst']
```

In [18]:
```
# Define functions for stopwords, bigrams, trigrams and lemmatization
def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc in texts]

def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

def make_trigrams(texts):
    return [trigram_mod[bigram_mod[doc]] for doc in texts]

def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    """https://spacy.io/api/annotation"""
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
    return texts_out
```

Project Implementation-Topic modelling module 1

Notebook Data Logs Comments (0) Settings

0 Edit

In [19]:
```
# Remove Stop Words
data_words_nostops = remove_stopwords(data_words)

# Form Bigrams
data_words_bigrams = make_bigrams(data_words_nostops)
```

In [20]:
```
nlp = spacy.load("en_core_web_sm",disable=['parser', 'ner'])
```

In [21]:
```
# Do lemmatization keeping only noun, adj, vb, adv
data_lemmatized = lemmatization(data_words_bigrams, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV'])

print(data_lemmatized[:1])
```

```
[['where', 's', 'thing', 'car', 'nntp_poste', 'host', 'park', 'line', 'wonder', 'enlighten', 'car', 'see', 'day', 'door', 'sport', 'car'
arly', 'call', 'door', 'really', 'small', 'addition', 'front_bumper', 'separate', 'rest', 'body', 'know', 'tellme', 'model', 'name', 'en
r', 'production', 'car', 'make', 'history', 'info', 'funky', 'look', 'car', 'mail', 'thank', 'bring', 'neighborhood', 'lerxst']]
```

In [22]:
```
#create dictionary for topic modelling
# Create Dictionary
id2word = corpora.Dictionary(data_lemmatized)

# Create Corpus
texts = data_lemmatized

# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]
```


Project Implementation-Topic modelling module 1

Notebook Data Logs Comments (0) Settings

0 Edit

In [22]:
```
#create dictionary for topic modelling
# Create Dictionary
id2word = corpora.Dictionary(data_lemmatized)

# Create Corpus
texts = data_lemmatized

# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]

# View
print(corpus[:1])
```

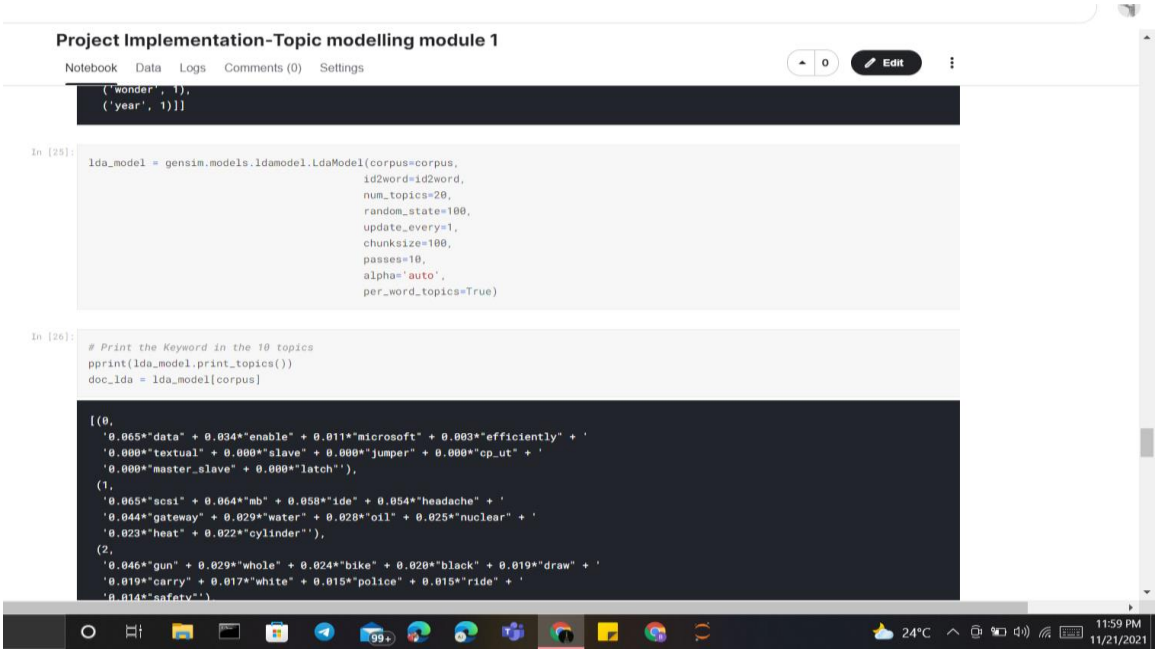

```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 5), (5, 1), (6, 2), (7, 1), (8, 1), (9, 1), (10, 1), (11, 1), (12, 1), (13, 1), (14, 1), (15, 1),
8, 1), (19, 2), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1), (29, 1), (30, 1), (31, 1), (32, 1), (33
1), (36, 1), (37, 1), (38, 1), (39, 1), (40, 1), (41, 1)]]
```

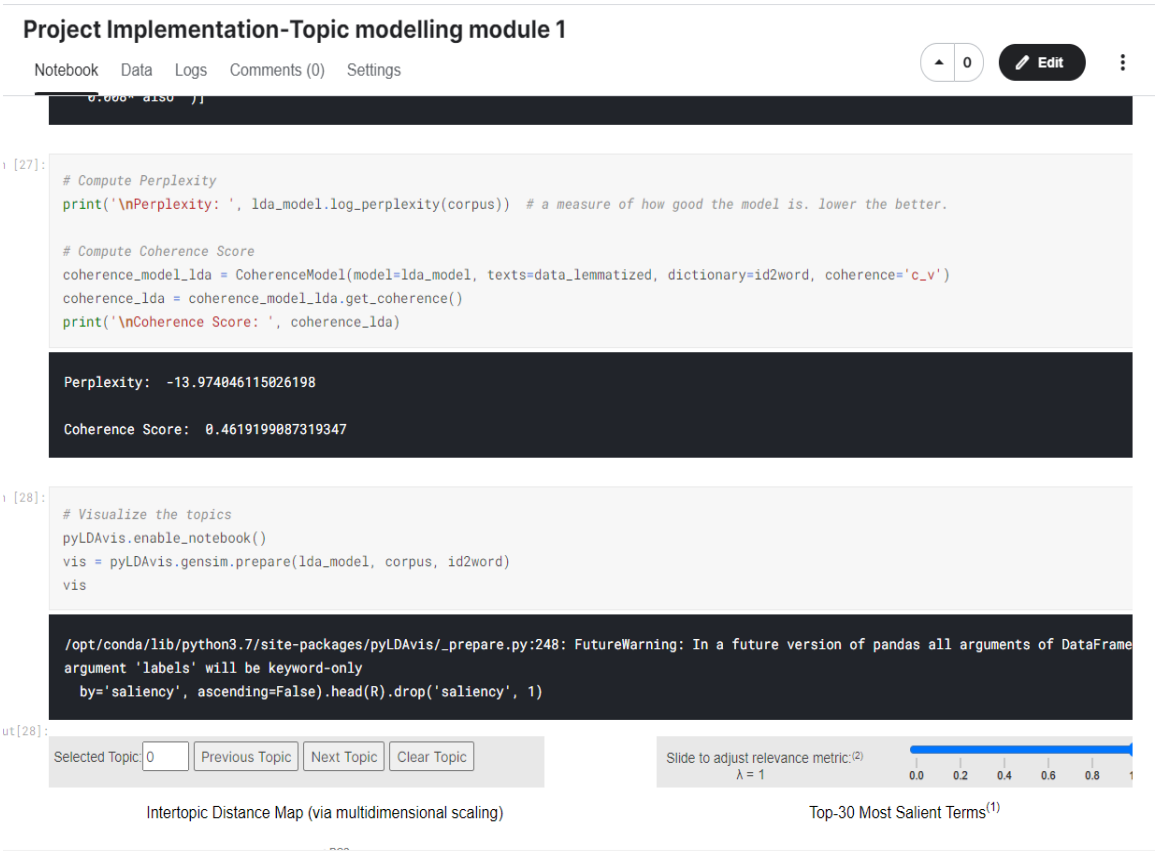
In [23]:
```
id2word[3]
```

Out[23]:
```
'call'
```

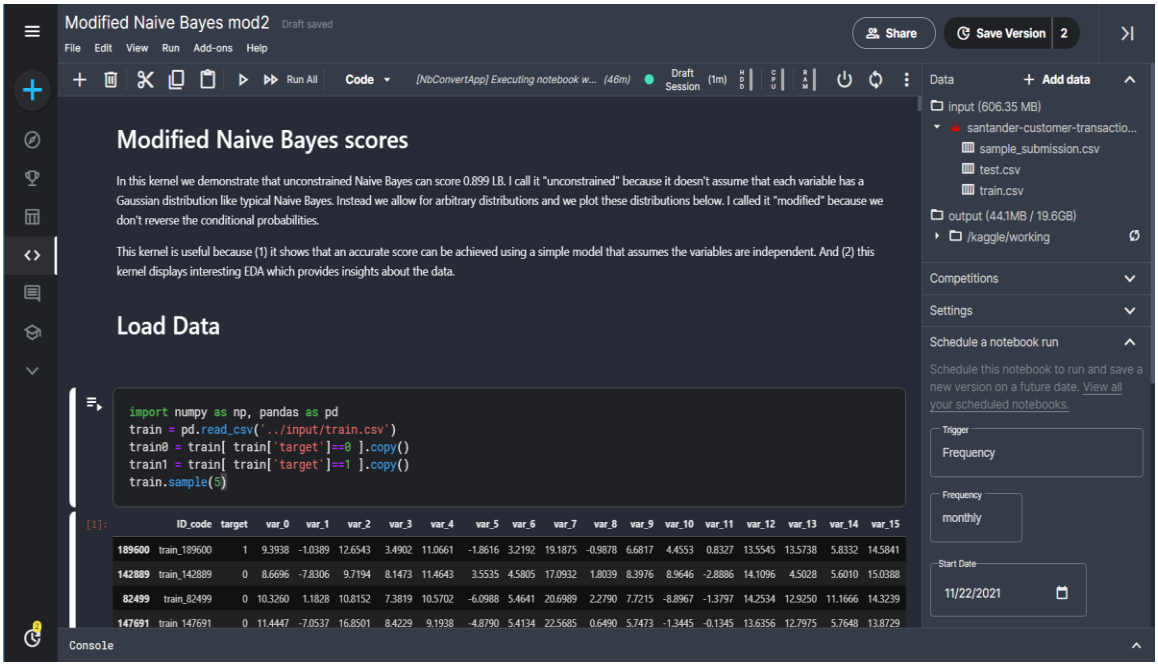
In [24]:
```
# Human readable format of corpus (term-frequency)
[[(id2word[id], freq) for id, freq in cp] for cp in corpus[:1]]
```

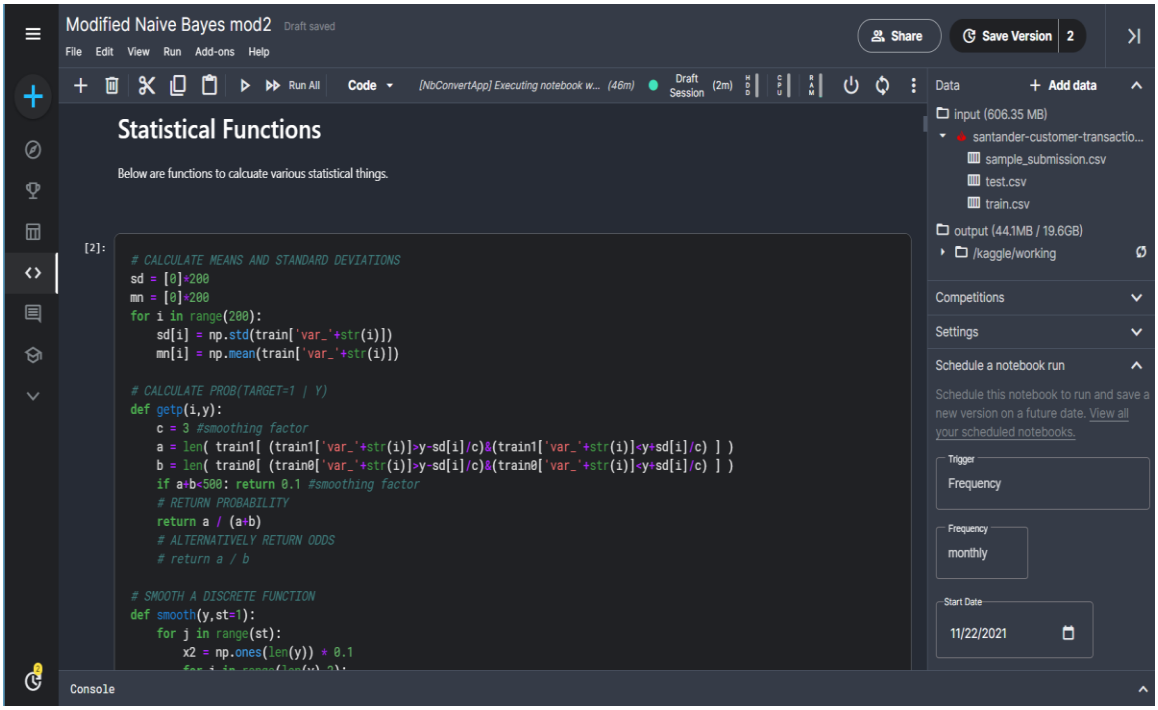
Out[24]:
```
[[('addition', 1),
  ('body', 1)
```

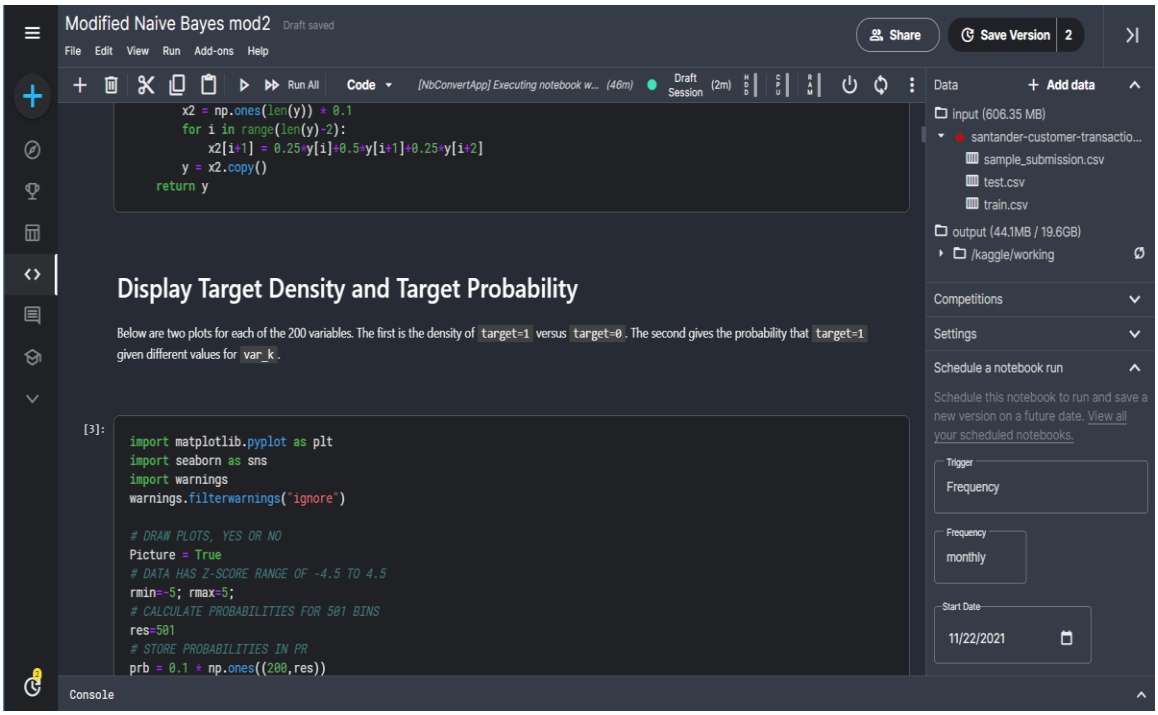

Project Implementation-Topic modelling module 1
Notebook Data Logs Comments (0) Settings
0
Edit
('wonder', 1),
('year', 1)]]
In [25]:
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
id2word=id2word,
num_topics=20,
random_state=100,
update_every=1,
chunksize=100,
passes=10,
alpha='auto',
per_word_topics=True)
In [26]:
# Print the Keyword in the 10 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
[(0,
'0.065*"data" + 0.034*"enable" + 0.011*"microsoft" + 0.003*"efficiently" + '
'0.000*"textual" + 0.000*"slave" + 0.000*"jumper" + 0.000*"cp_ut" + '
'0.000*"master_slave" + 0.000*"latch"'),
(1,
'0.065*"scsi" + 0.064*"mb" + 0.058*"ide" + 0.054*"headache" + '
'0.044*"gateway" + 0.029*"water" + 0.028*"oil" + 0.025*"nuclear" + '
'0.023*"heat" + 0.022*"cylinder"'),
(2,
'0.046*"gun" + 0.029*"whole" + 0.024*"bike" + 0.020*"black" + 0.019*"draw" + '
'0.019*"carry" + 0.017*"white" + 0.015*"police" + 0.015*"ride" + '
'0.014*"safety"'),
24°C
11:59 PM
11/21/2021

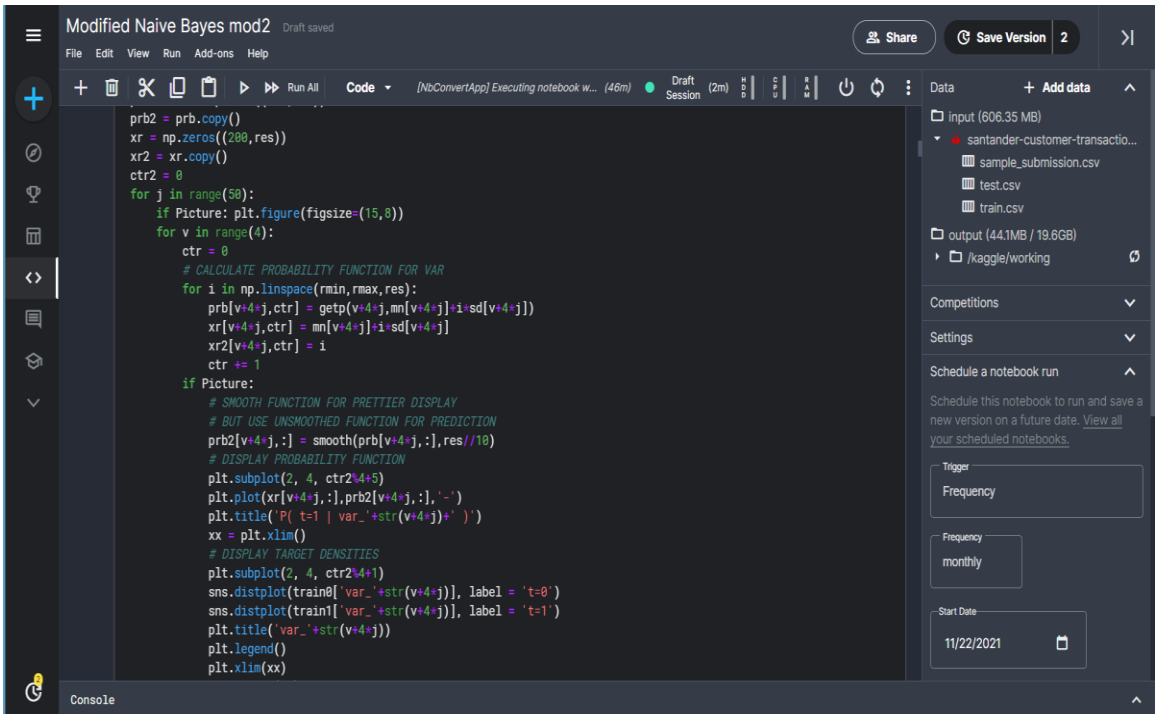

Project Implementation-Topic modelling module 1
Notebook Data Logs Comments (0) Settings
0
Edit
In [27]:
Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus)) # a measure of how good the model is. lower the better.
# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
Perplexity: -13.974046115026198
Coherence Score: 0.4619199087319347
In [28]:
# Visualize the topics
pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim.prepare(lda_model, corpus, id2word)
vis
/opt/conda/lib/python3.7/site-packages/pyLDAvis/_prepare.py:248: FutureWarning: In a future version of pandas all arguments of DataFrame
argument 'labels' will be keyword-only
by='saliency', ascending=False).head(R).drop('saliency', 1)
Out[28]:
Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
Intertopic Distance Map (via multidimensional scaling)
Top-30 Most Salient Terms(1)

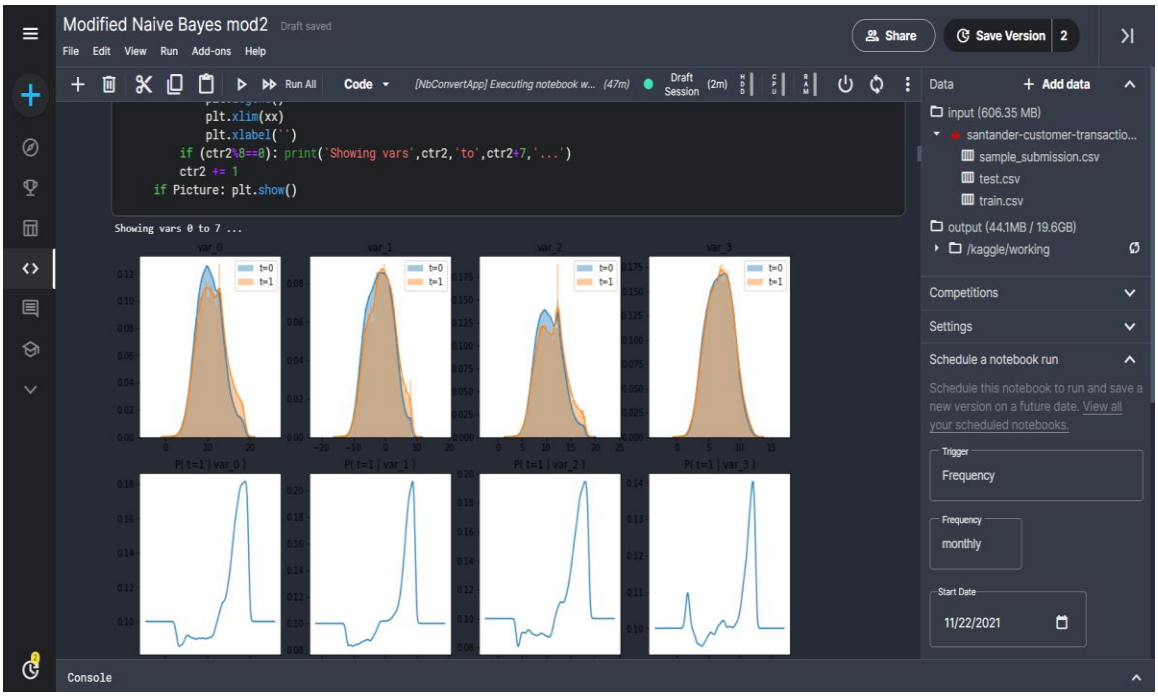
5.2 MODULE 2 IMPLEMENTATION

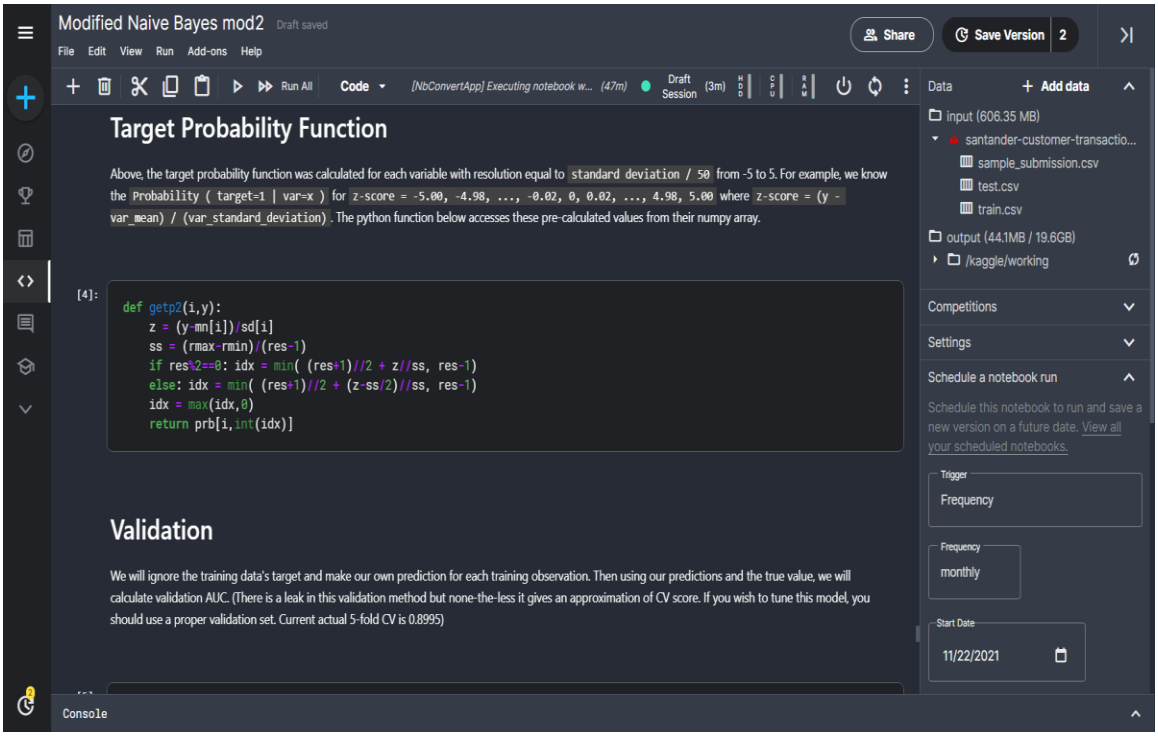





Modified Naive Bayes mod2
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 2
Run All
Code
[NbConvertApp] Executing notebook w... (46m)
Draft Session (2m)
x2 = np.ones(len(y)) * 0.1
for i in range(len(y)-2):
x2[i+1] = 0.25*y[i]+0.5*y[i+1]+0.25*y[i+2]
y = x2.copy()
return y
Display Target Density and Target Probability
Below are two plots for each of the 200 variables. The first is the density of target=1 versus target=0. The second gives the probability that target=1 given different values for var_k.
[3]:
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
# DRAW PLOTS, YES OR NO
Picture = True
# DATA HAS Z-SCORE RANGE OF -4.5 TO 4.5
rmin=-5; rmax=5;
# CALCULATE PROBABILITIES FOR 501 BINS
res=501
# STORE PROBABILITIES IN PR
prb = 0.1 * np.ones((200,res))
Data
Add data
input (606.35 MB)
santander-customer-transactio...
sample_submission.csv
test.csv
train.csv
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Console


Modified Naive Bayes mod2
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 2
Run All
Code
[NbConvertApp] Executing notebook w... (46m)
Draft Session (2m)
prb2 = prb.copy()
xr = np.zeros((200,res))
xr2 = xr.copy()
ctr2 = 0
for j in range(50):
if Picture: plt.figure(figsize=(15,8))
for v in range(4):
ctr = 0
# CALCULATE PROBABILITY FUNCTION FOR VAR
for i in np.linspace(rmin,rmax,res):
prb[v+4*j,ctr] = getp(v+4*j,mn[v+4*j]+i*sd[v+4*j])
xr[v+4*j,ctr] = mn[v+4*j]+i*sd[v+4*j]
xr2[v+4*j,ctr] = i
ctr += 1
if Picture:
# SMOOTH FUNCTION FOR PRETTIER DISPLAY
# BUT USE UNSMOOTHED FUNCTION FOR PREDICTION
prb2[v+4*j,:] = smooth(prb[v+4*j,:],res//10)
# DISPLAY PROBABILITY FUNCTION
plt.subplot(2, 4, ctr2%4+5)
plt.plot(xr[v+4*j,:],prb2[v+4*j,:],'-')
plt.title('P(t=1 | var_'+str(v+4*j)+')')
xx = plt.xlim()
# DISPLAY TARGET DENSITIES
plt.subplot(2, 4, ctr2%4+1)
sns.distplot(train0['var_'+str(v+4*j)], label = 't=0')
sns.distplot(train1['var_'+str(v+4*j)], label = 't=1')
plt.title('var_'+str(v+4*j))
plt.legend()
plt.xlim(xx)
Data
Add data
input (606.35 MB)
santander-customer-transactio...
sample_submission.csv
test.csv
train.csv
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Console


Modified Naive Bayes mod2
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 2
Run All
Code
[NbConvertApp] Executing notebook w... (47m)
Draft Session (2m)
plt.xlim(xx)
plt.xlabel('')
if (ctr2%8==0): print('Showing vars',ctr2,'to',ctr2+7,'...')
ctr2 += 1
if Picture: plt.show()
Showing vars 0 to 7 ...
var_0
var_1
var_2
var_3
t=0
t=1
P(t=1 | var_0)
P(t=1 | var_1)
P(t=1 | var_2)
P(t=1 | var_3)
Data
Add data
input (606.35 MB)
santander-customer-transactio...
sample_submission.csv
test.csv
train.csv
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Console


Modified Naive Bayes mod2
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 2
Run All
Code
[NbConvertApp] Executing notebook w... (47m)
Draft Session (3m)
Target Probability Function
Above, the target probability function was calculated for each variable with resolution equal to standard deviation / 50 from -5 to 5. For example, we know the Probability (target=1 | var=x) for z-score = -5.00, -4.98, ..., -0.02, 0, 0.02, ..., 4.98, 5.00 where z-score = (y - var_mean) / (var_standard_deviation) . The python function below accesses these pre-calculated values from their numpy array.
[4]:
def getp2(i,y):
z = (y-mn[i])/sd[i]
ss = (rmax-rmin)/(res-1)
if res%2==0: idx = min((res+1)//2 + z//ss, res-1)
else: idx = min((res+1)//2 + (z-ss/2)//ss, res-1)
idx = max(idx,0)
return prb[i,int(idx)]
Validation
We will ignore the training data's target and make our own prediction for each training observation. Then using our predictions and the true value, we will calculate validation AUC. (There is a leak in this validation method but none-the-less it gives an approximation of CV score. If you wish to tune this model, you should use a proper validation set. Current actual 5-fold CV is 0.8995)
Data
Add data
input (606.35 MB)
santander-customer-transactio...
sample_submission.csv
test.csv
train.csv
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Console

Modified Naive Bayes mod2 Draft saved

File Edit View Run Add-ons Help

Share Save Version 2

Run All Code [NbConvertApp] Executing notebook w... (47m) Draft Session (3m)

[5]:
```
from sklearn.metrics import roc_auc_score
print('Calculating 200000 predictions and displaying a few examples...')
pred = [0]*200000; ctr = 0
for r in train.index:
    q = 0.1
    for i in range(200):
        q *= 10*getp2(i,train.iloc[r,2+i])
    if ctr%25000==0: print('train',r,'has target =',train.iloc[r,1],'and prediction =',q)
    pred[ctr]=q; ctr += 1
print('##############')
print('Validation AUC =',roc_auc_score(train['target'], pred))
```

```
Calculating 200000 predictions and displaying a few examples...
train 0 has target = 0 and prediction = 0.020849097518485876
train 25000 has target = 0 and prediction = 0.047234280974034905
train 50000 has target = 0 and prediction = 0.10883774386764035
train 75000 has target = 0 and prediction = 0.2218646945701704
train 100000 has target = 0 and prediction = 0.12978893156190568
train 125000 has target = 0 and prediction = 0.019172676892370028
train 150000 has target = 0 and prediction = 0.062086314744358 1
train 175000 has target = 0 and prediction = 0.071607054743778 9
##############
Validation AUC = 0.9055702103328558
```

[6]:
```
#https://stackoverflow.com/questions/25009284/how-to-plot-roc-curve-in-python
from sklearn import metrics
fpr, tpr, threshold = metrics.roc_curve(train['target'], pred)
roc_auc = metrics.auc(fpr, tpr)

plt.figure(figsize=(6,6))
```

Data + Add data

input (606.35 MB)
- santander-customer-transactio...
  - sample_submission.csv
  - test.csv
  - train.csv

output (44.1MB / 19.6GB)
- /kaggle/working

Competitions

Settings

Schedule a notebook run

Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.

Trigger: Frequency

Frequency: monthly

Start Date: 11/22/2021

Console

Modified Naive Bayes mod2 Draft saved

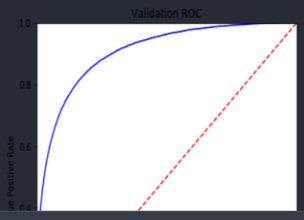
File Edit View Run Add-ons Help

Share Save Version 2

Run All Code [NbConvertApp] Executing notebook w... (48m) Draft Session (3m)

[6]:
```
#https://stackoverflow.com/questions/25009284/how-to-plot-roc-curve-in-python
from sklearn import metrics
fpr, tpr, threshold = metrics.roc_curve(train['target'], pred)
roc_auc = metrics.auc(fpr, tpr)

plt.figure(figsize=(6,6))
plt.title('Validation ROC')
plt.plot(fpr, tpr, 'b', label = 'Val AUC = %0.3f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```



Data + Add data

input (606.35 MB)
- santander-customer-transactio...
  - sample_submission.csv
  - test.csv
  - train.csv

output (44.1MB / 19.6GB)
- /kaggle/working

Competitions

Settings

Schedule a notebook run

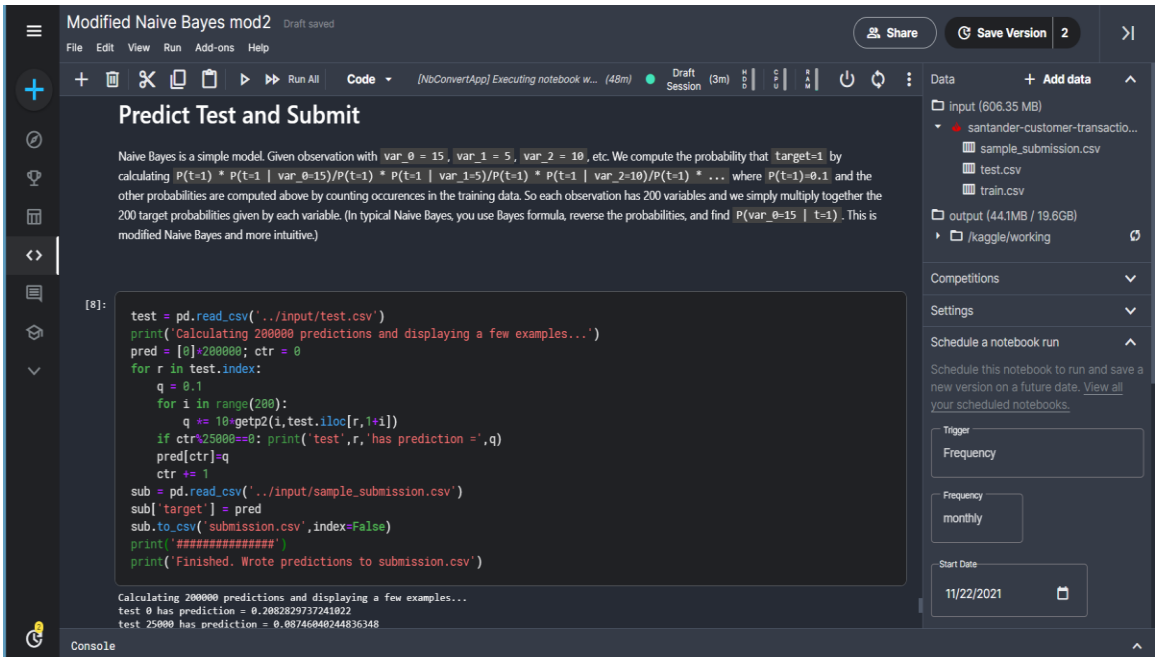
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.

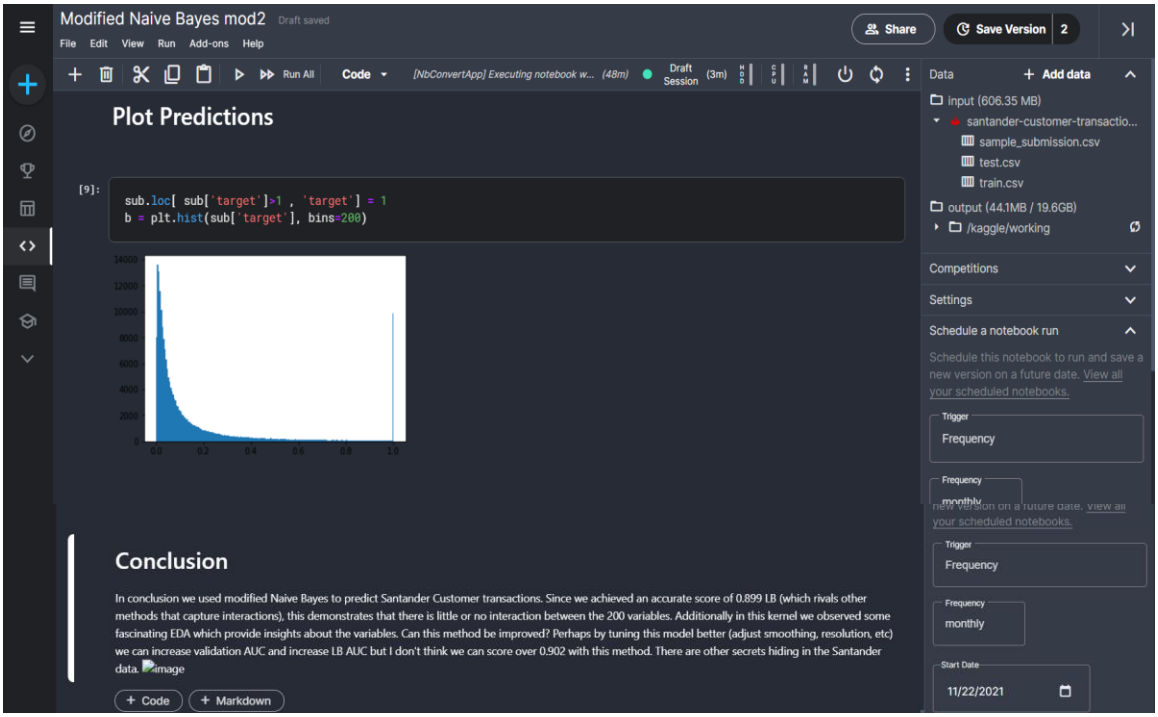
Trigger: Frequency

Frequency: monthly

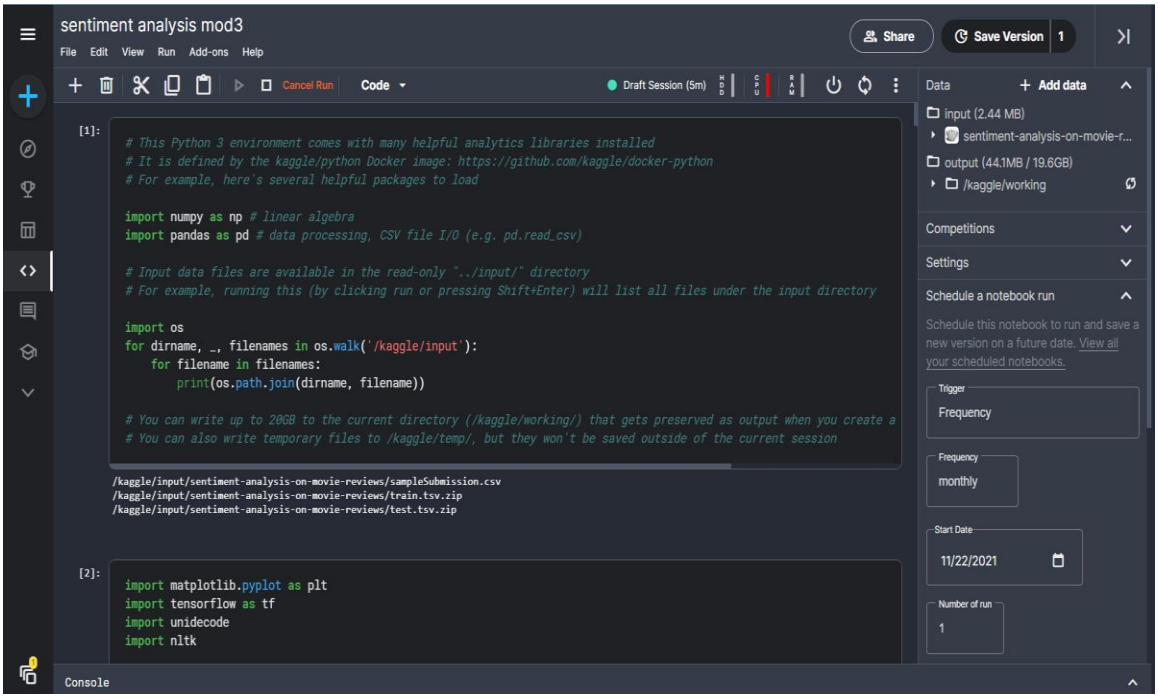
Start Date: 11/22/2021

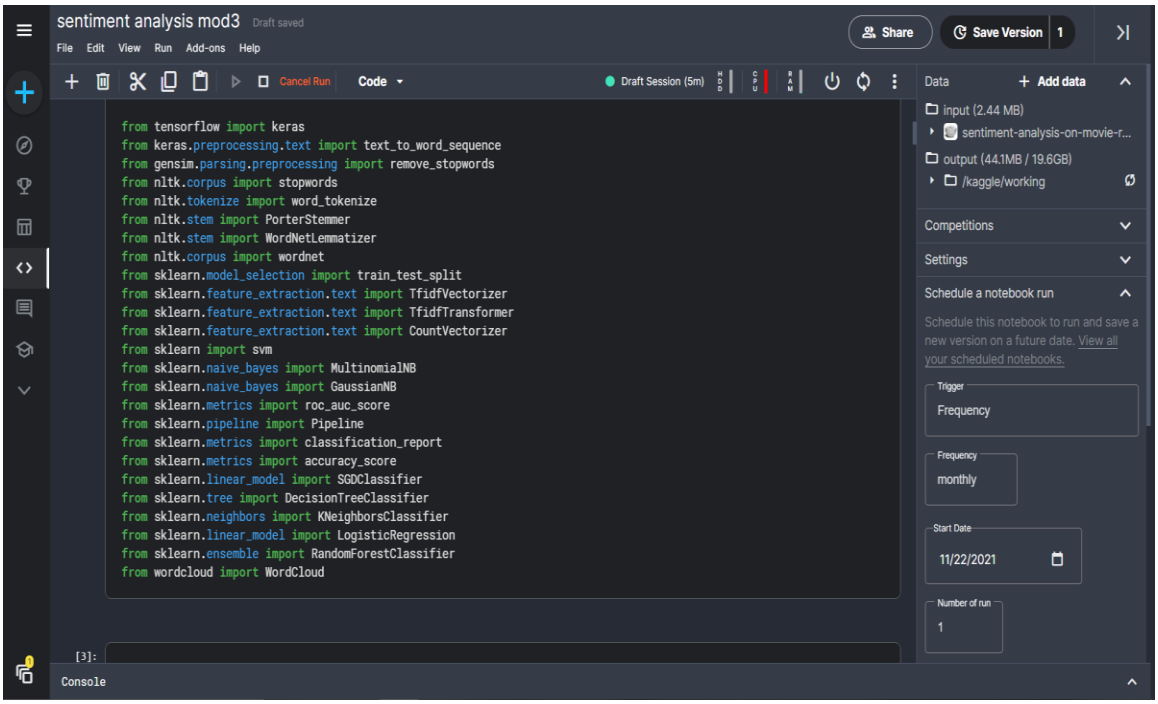
Console

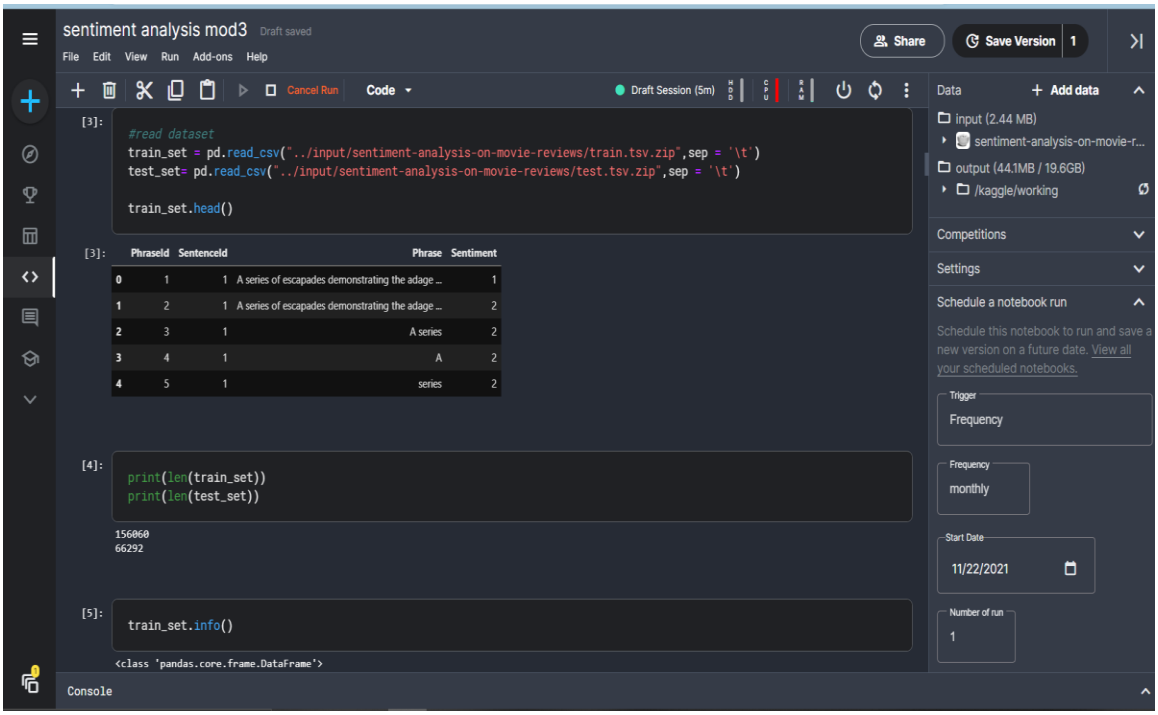

Modified Naive Bayes mod2
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 2
Run All
Code
[NbConvertApp] Executing notebook w... (48m)
Draft Session (3m)
Predict Test and Submit
Naive Bayes is a simple model. Given observation with var_0 = 15, var_1 = 5, var_2 = 10, etc. We compute the probability that target=1 by calculating P(t=1) * P(t=1 | var_0=15)/P(t=1) * P(t=1 | var_1=5)/P(t=1) * P(t=1 | var_2=10)/P(t=1) * ... where P(t=1)=0.1 and the other probabilities are computed above by counting occurences in the training data. So each observation has 200 variables and we simply multiply together the 200 target probabilities given by each variable. (In typical Naive Bayes, you use Bayes formula, reverse the probabilities, and find P(var_0=15 | t=1). This is modified Naive Bayes and more intuitive.)
[8]:
test = pd.read_csv('../input/test.csv')
print('Calculating 200000 predictions and displaying a few examples...')
pred = [0]*200000; ctr = 0
for r in test.index:
q = 0.1
for i in range(200):
q *= 10*getp2(i,test.iloc[r,1+i])
if ctr%25000==0: print('test',r,'has prediction =',q)
pred[ctr]=q
ctr += 1
sub = pd.read_csv('../input/sample_submission.csv')
sub['target'] = pred
sub.to_csv('submission.csv',index=False)
print('##############')
print('Finished. Wrote predictions to submission.csv')
Calculating 200000 predictions and displaying a few examples...
test 0 has prediction = 0.2082829737241022
test 25000 has prediction = 0.08746040244836348
Console
Data
Add data
input (606.35 MB)
santander-customer-transactio...
sample_submission.csv
test.csv
train.csv
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021

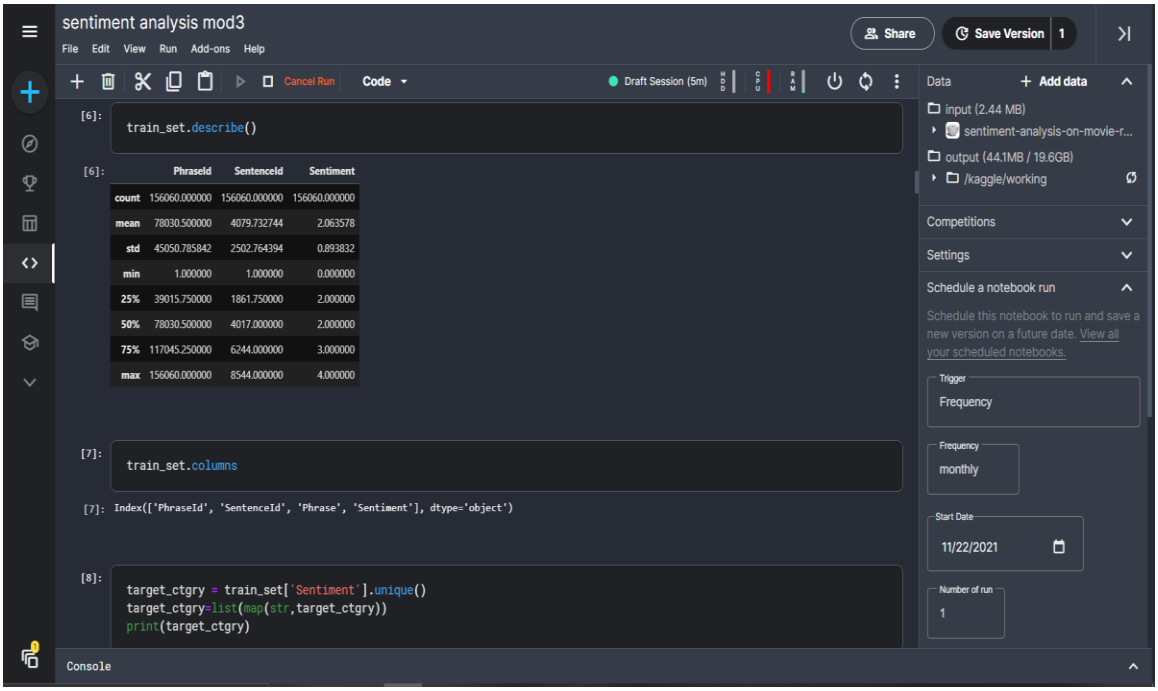

Modified Naive Bayes mod2
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 2
Run All
Code
[NbConvertApp] Executing notebook w... (48m)
Draft Session (3m)
Plot Predictions
[9]:
sub.loc[sub['target']>1 , 'target'] = 1
b = plt.hist(sub['target'], bins=200)
Conclusion
In conclusion we used modified Naive Bayes to predict Santander Customer transactions. Since we achieved an accurate score of 0.899 LB (which rivals other methods that capture interactions), this demonstrates that there is little or no interaction between the 200 variables. Additionally in this kernel we observed some fascinating EDA which provide insights about the variables. Can this method be improved? Perhaps by tuning this model better (adjust smoothing, resolution, etc) we can increase validation AUC and increase LB AUC but I don't think we can score over 0.902 with this method. There are other secrets hiding in the Santander data. image
+ Code
+ Markdown
Data
Add data
input (606.35 MB)
santander-customer-transactio...
sample_submission.csv
test.csv
train.csv
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021

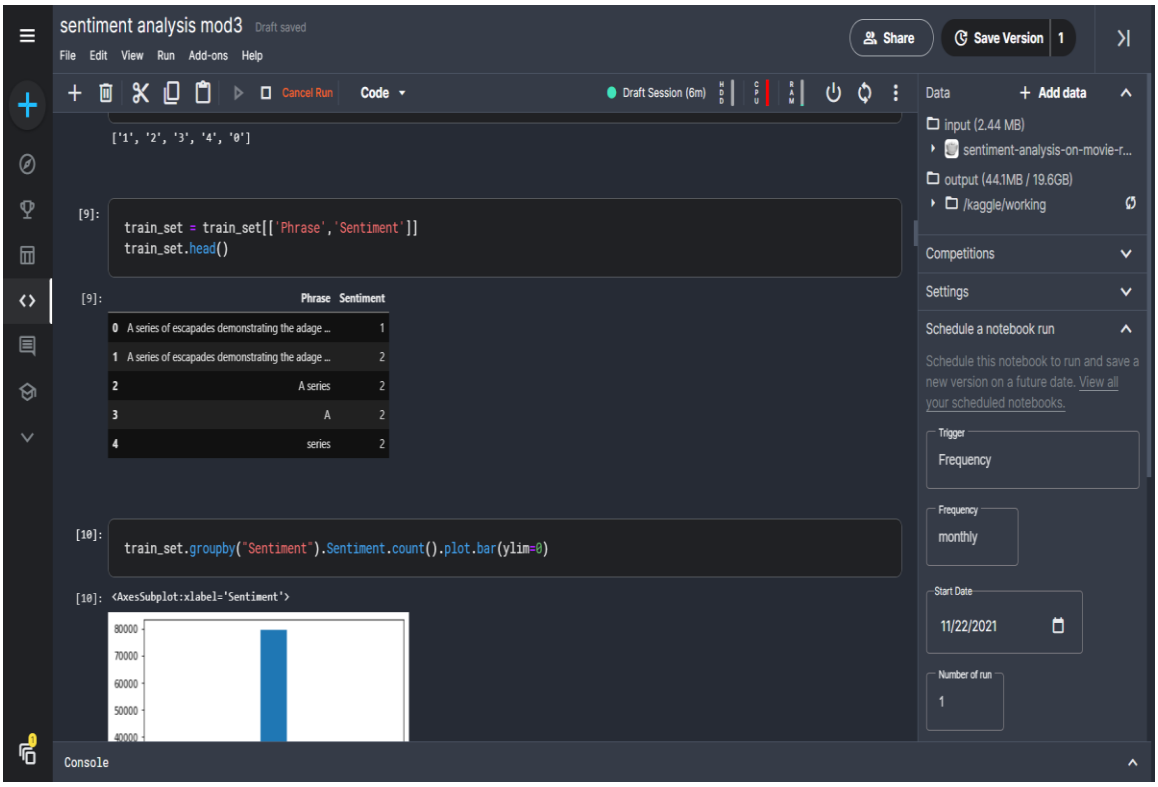
5.3 MODULE 3 IMPLEMENTATION

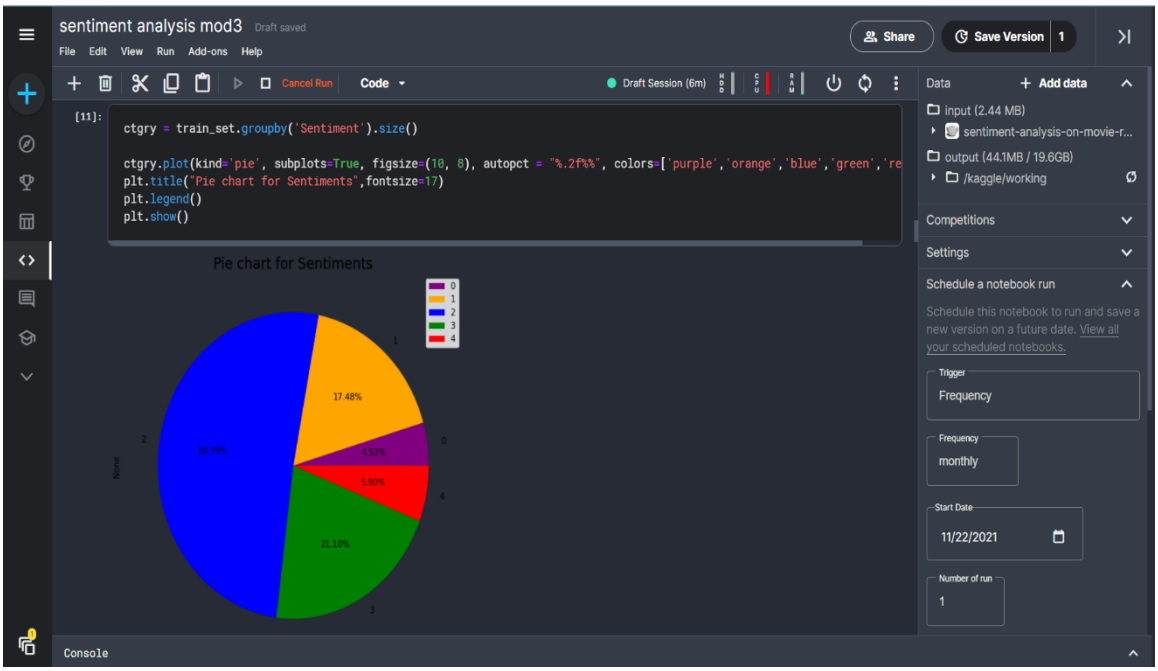


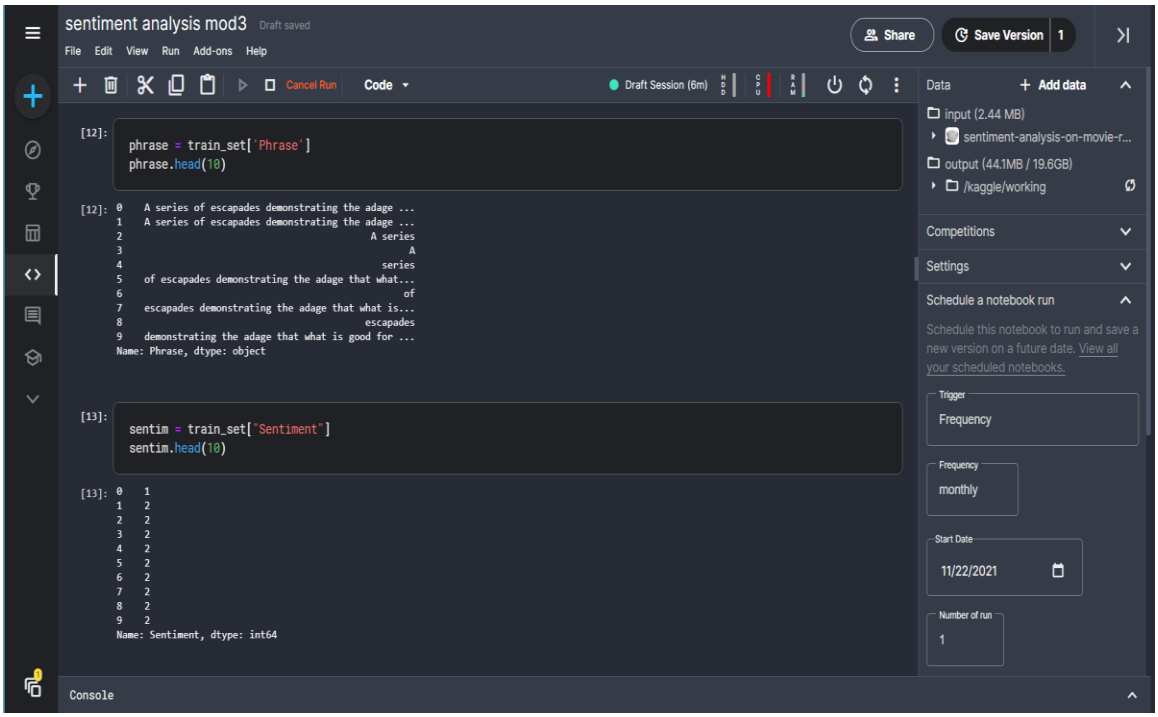


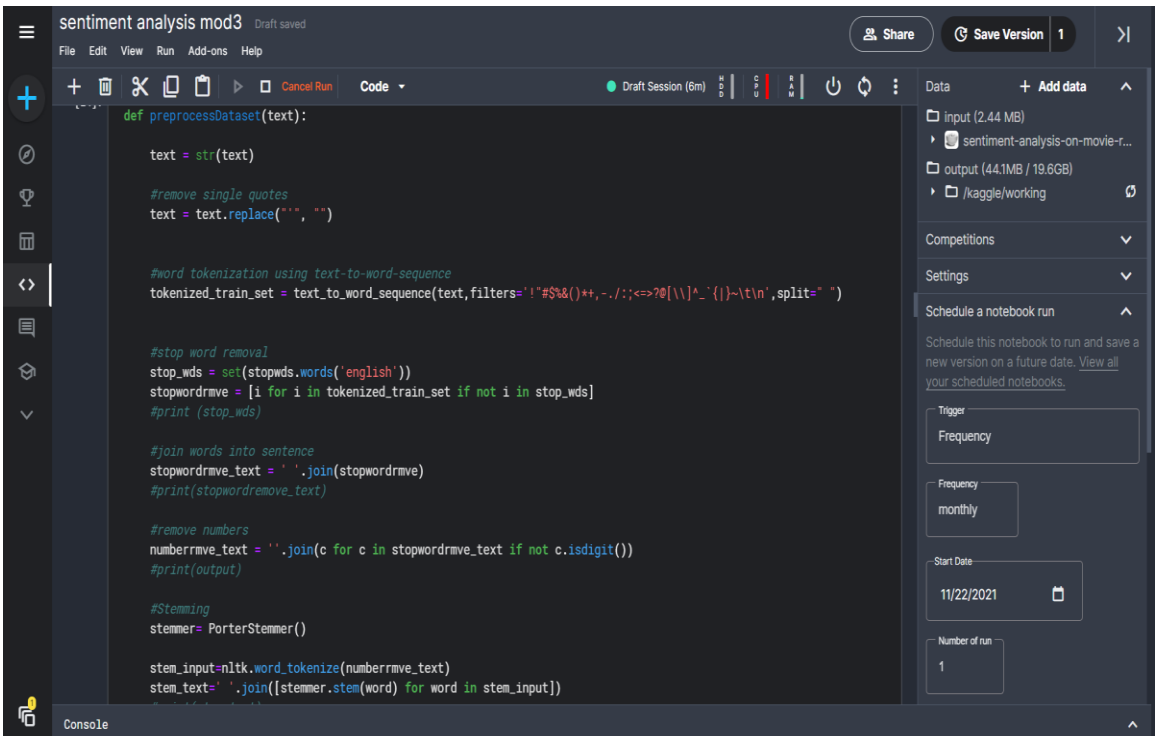

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (5m)
[3]:
#read dataset
train_set = pd.read_csv("../input/sentiment-analysis-on-movie-reviews/train.tsv.zip",sep = '\t')
test_set= pd.read_csv("../input/sentiment-analysis-on-movie-reviews/test.tsv.zip",sep = '\t')
train_set.head()
[3]:
PhraseId SentenceId Phrase Sentiment
0 1 1 A series of escapades demonstrating the adage ... 1
1 2 1 A series of escapades demonstrating the adage ... 2
2 3 1 A series 2
3 4 1 A 2
4 5 1 series 2
[4]:
print(len(train_set))
print(len(test_set))
156060
66292
[5]:
train_set.info()
<class 'pandas.core.frame.DataFrame'>
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1

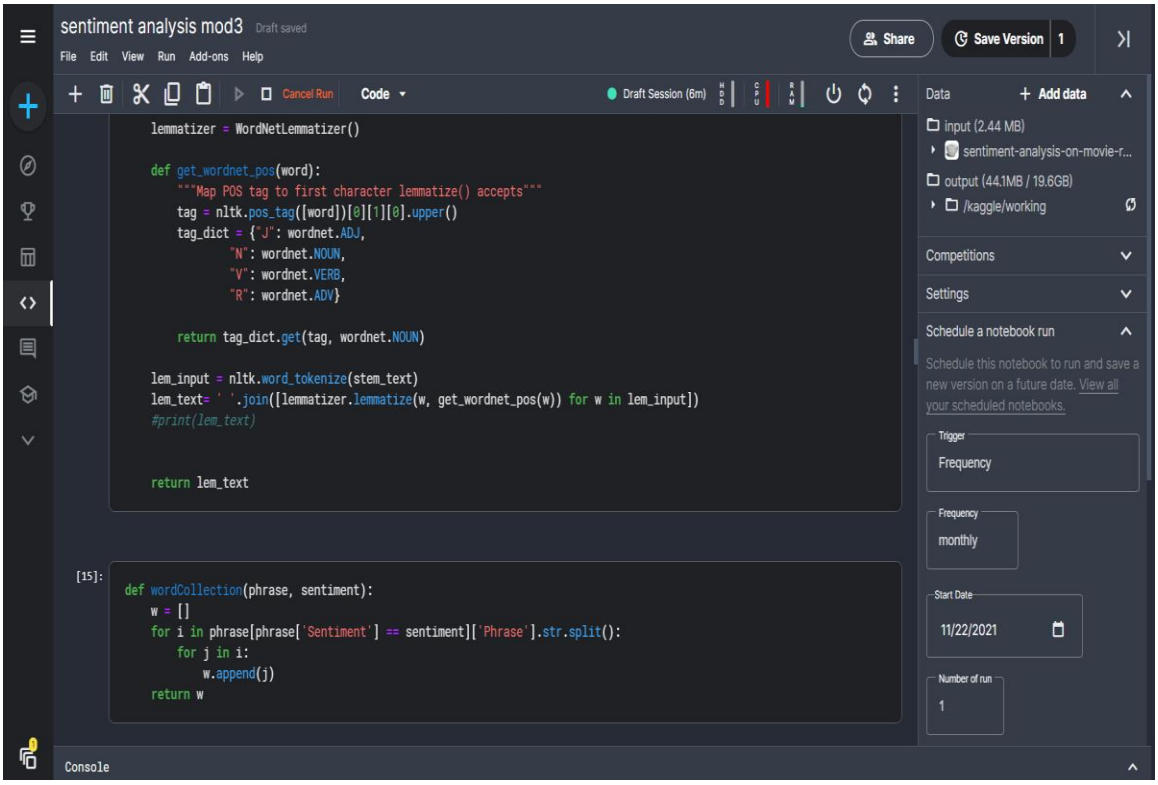

sentiment analysis mod3
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (5m)
[6]:
train_set.describe()
[6]:
PhraseId SentenceId Sentiment
count 156060.000000 156060.000000 156060.000000
mean 78030.500000 4079.732744 2.063578
std 45050.785842 2502.764394 0.893832
min 1.000000 1.000000 0.000000
25% 39015.750000 1861.750000 2.000000
50% 78030.500000 4017.000000 2.000000
75% 117045.250000 6244.000000 3.000000
max 156060.000000 8544.000000 4.000000
[7]:
train_set.columns
[7]: Index(['PhraseId', 'SentenceId', 'Phrase', 'Sentiment'], dtype='object')
[8]:
target_ctgry = train_set['Sentiment'].unique()
target_ctgry=list(map(str,target_ctgry))
print(target_ctgry)
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1

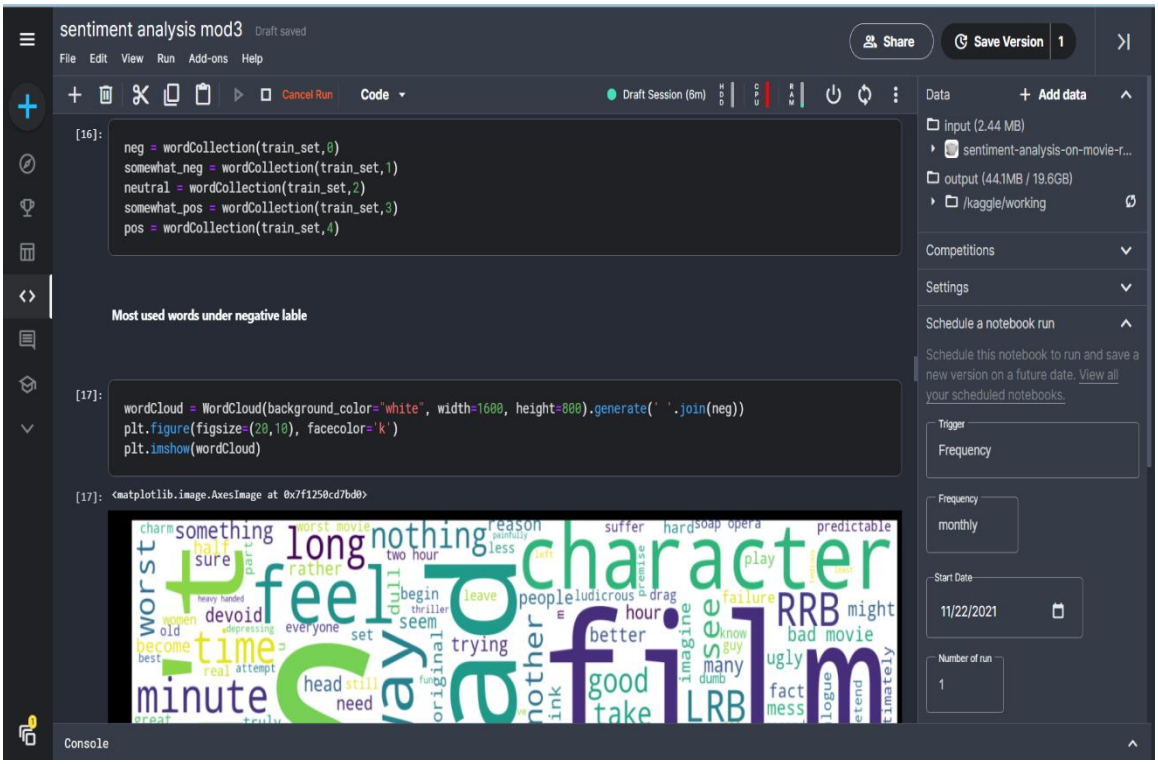

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (6m)
['1', '2', '3', '4', '0']
[9]:
train_set = train_set[['Phrase','Sentiment']]
train_set.head()
[9]:
Phrase Sentiment
0 A series of escapades demonstrating the adage ... 1
1 A series of escapades demonstrating the adage ... 2
2 A series 2
3 A 2
4 series 2
[10]:
train_set.groupby("Sentiment").Sentiment.count().plot.bar(ylim=0)
[10]:
<AxesSubplot:xlabel='Sentiment'>
80000
70000
60000
50000
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1

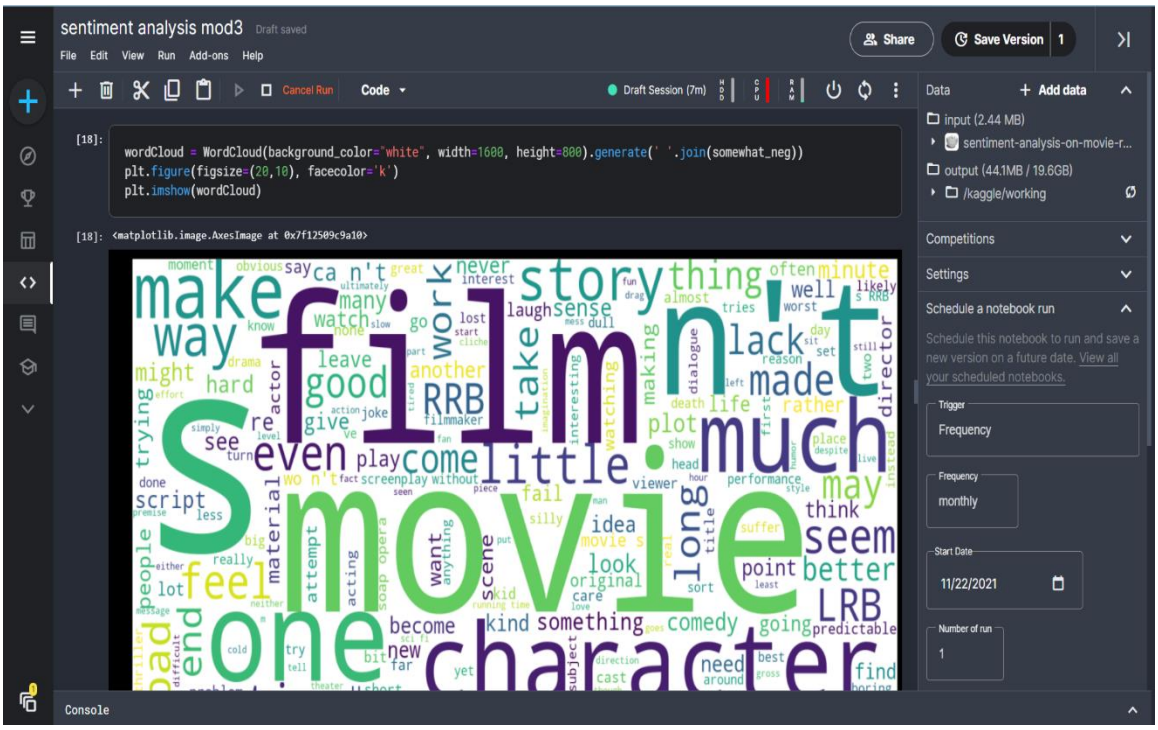

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (6m)
[11]:
ctgry = train_set.groupby('Sentiment').size()
ctgry.plot(kind='pie', subplots=True, figsize=(10, 8), autopct = "%.2f%%", colors=['purple','orange','blue','green','re
plt.title("Pie chart for Sentiments",fontsize=17)
plt.legend()
plt.show()
Pie chart for Sentiments
0
1
2
3
4
17.48%
4.53%
5.90%
21.10%
50.99%
None
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1

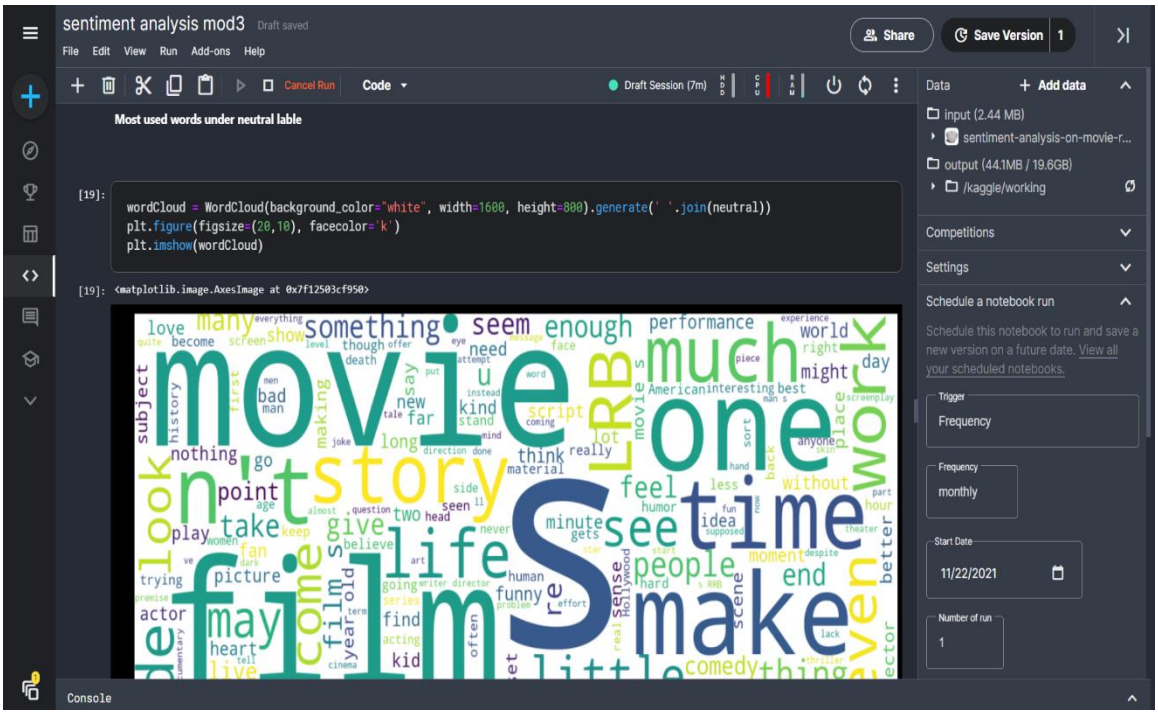

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (6m)
[12]:
phrase = train_set['Phrase']
phrase.head(10)
[12]: 0 A series of escapades demonstrating the adage ...
1 A series of escapades demonstrating the adage ...
2 A series
3 A
4 series
5 of escapades demonstrating the adage that what...
6 of
7 escapades demonstrating the adage that what is...
8 escapades
9 demonstrating the adage that what is good for ...
Name: Phrase, dtype: object
[13]:
sentim = train_set["Sentiment"]
sentim.head(10)
[13]: 0 1
1 2
2 2
3 2
4 2
5 2
6 2
7 2
8 2
9 2
Name: Sentiment, dtype: int64
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1

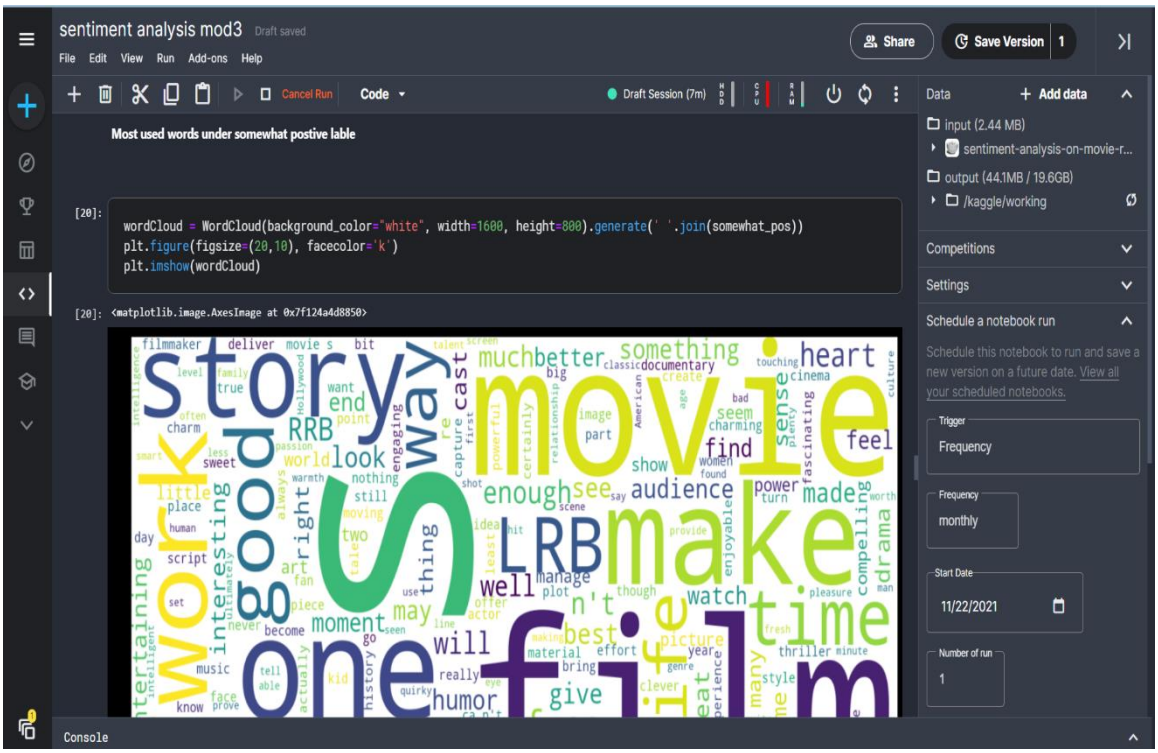

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (6m)
def preprocessDataset(text):
text = str(text)
#remove single quotes
text = text.replace("'", "")
#word tokenization using text-to-word-sequence
tokenized_train_set = text_to_word_sequence(text,filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n',split=" ")
#stop word removal
stop_wds = set(stopwds.words('english'))
stopwordrmve = [i for i in tokenized_train_set if not i in stop_wds]
#print (stop_wds)
#join words into sentence
stopwordrmve_text = ' '.join(stopwordrmve)
#print(stopwordremove_text)
#remove numbers
numberrmve_text = ''.join(c for c in stopwordrmve_text if not c.isdigit())
#print(output)
#Stemming
stemmer= PorterStemmer()
stem_input=nltk.word_tokenize(numberrmve_text)
stem_text=' '.join([stemmer.stem(word) for word in stem_input])
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1

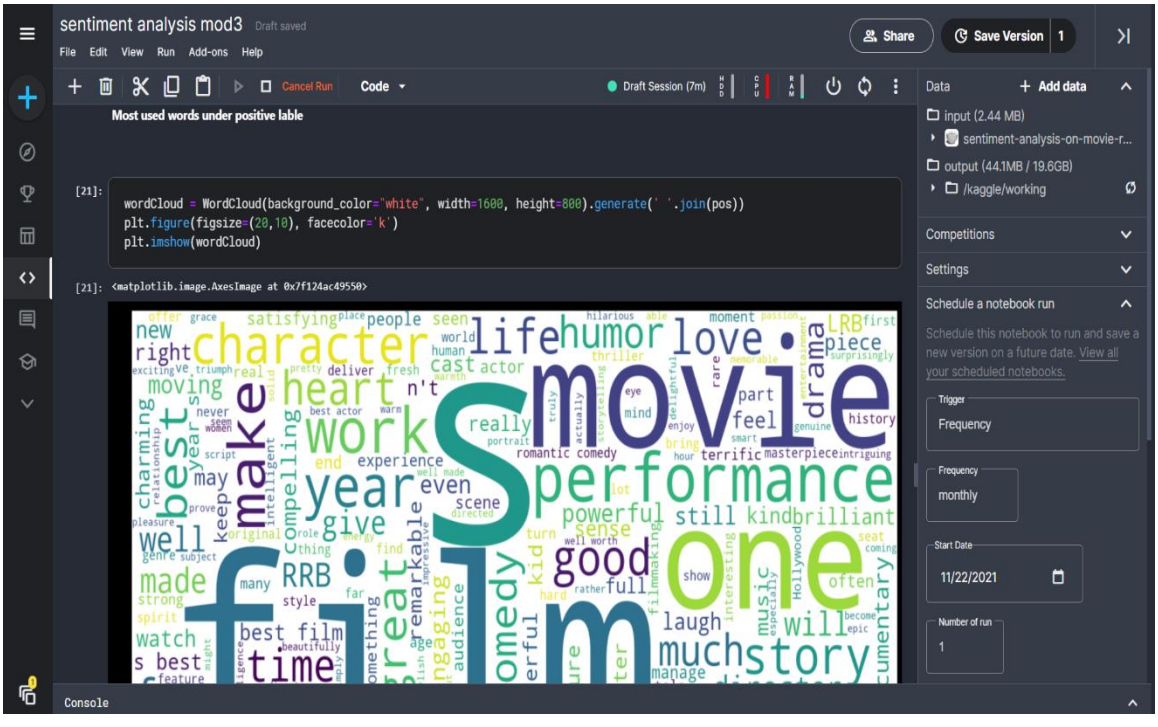

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (6m)
lemmatizer = WordNetLemmatizer()
def get_wordnet_pos(word):
"""Map POS tag to first character lemmatize() accepts"""
tag = nltk.pos_tag([word])[0][1][0].upper()
tag_dict = {"J": wordnet.ADJ,
"N": wordnet.NOUN,
"V": wordnet.VERB,
"R": wordnet.ADV}
return tag_dict.get(tag, wordnet.NOUN)
lem_input = nltk.word_tokenize(stem_text)
lem_text= ' '.join([lemmatizer.lemmatize(w, get_wordnet_pos(w)) for w in lem_input])
#print(lem_text)
return lem_text
[15]:
def wordCollection(phrase, sentiment):
w = []
for i in phrase[phrase['Sentiment'] == sentiment]['Phrase'].str.split():
for j in i:
w.append(j)
return w
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

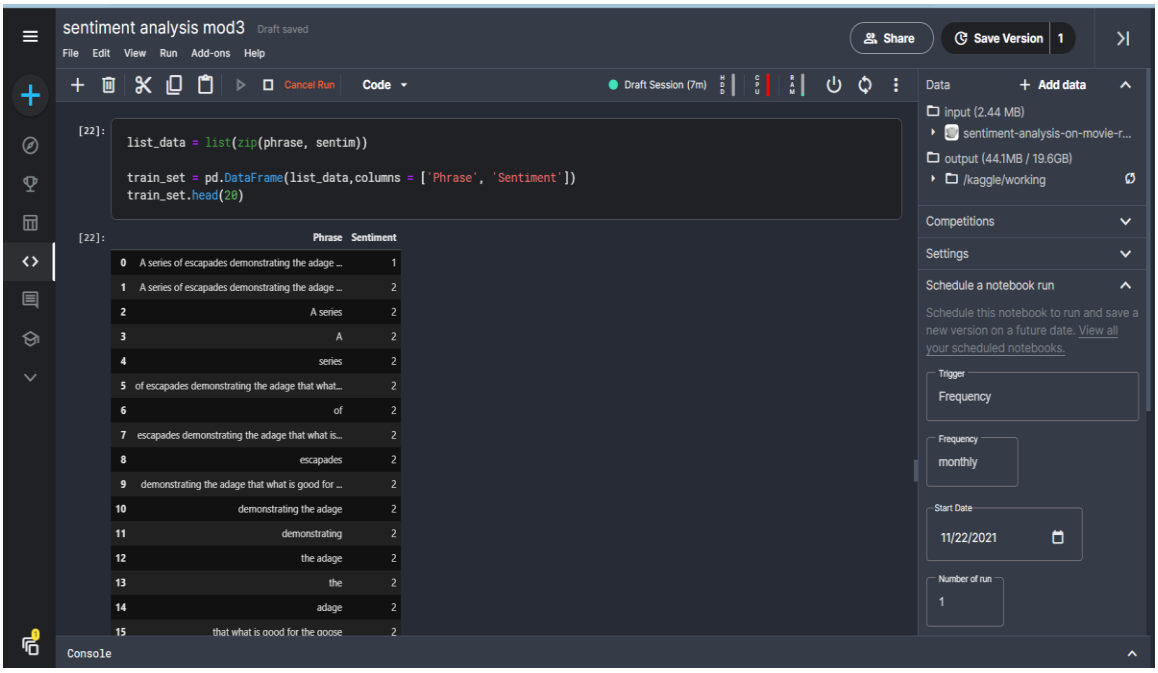

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (6m)
[16]:
neg = wordCollection(train_set,0)
somewhat_neg = wordCollection(train_set,1)
neutral = wordCollection(train_set,2)
somewhat_pos = wordCollection(train_set,3)
pos = wordCollection(train_set,4)
Most used words under negative lable
[17]:
wordCloud = WordCloud(background_color="white", width=1600, height=800).generate(' '.join(neg))
plt.figure(figsize=(20,10), facecolor='k')
plt.imshow(wordCloud)
[17]: <matplotlib.figure.AxesImage at 0x7f1250cd7bd0>
character
film
feel
bad
long
nothing
something
minute
time
RRB
good
bad movie
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

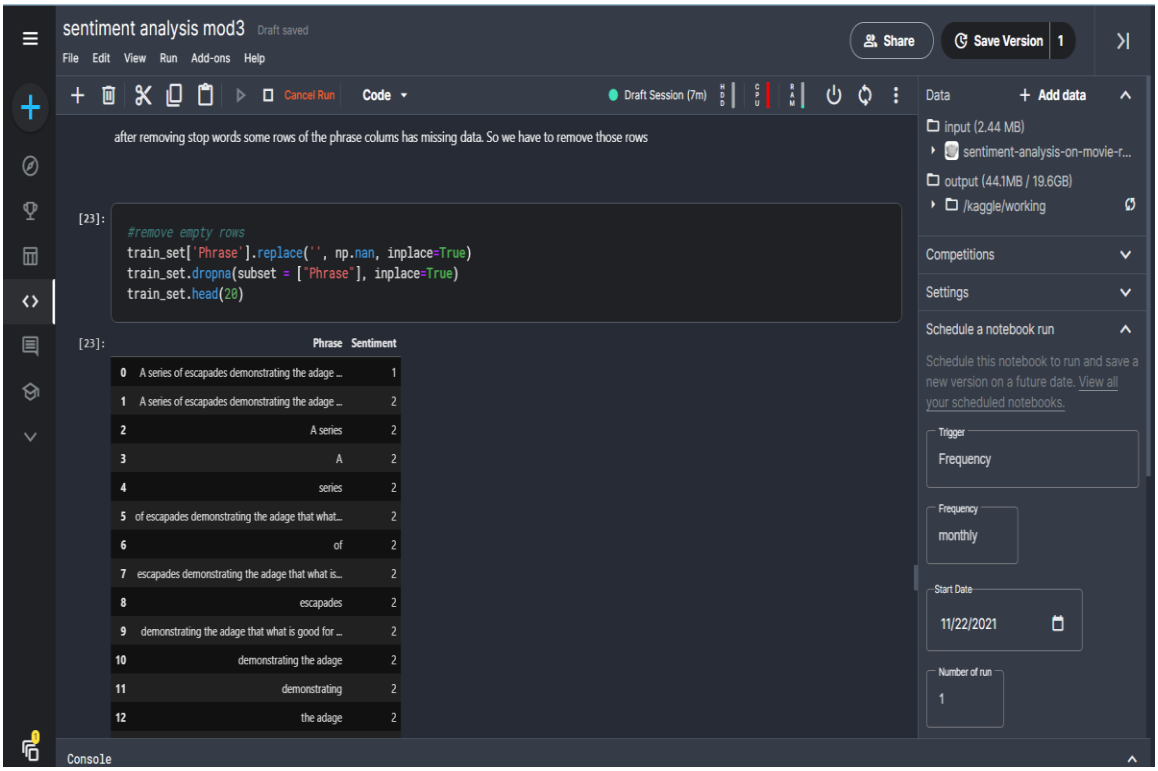

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
[18]:
wordCloud = WordCloud(background_color="white", width=1600, height=800).generate(' '.join(somewhat_neg))
plt.figure(figsize=(20,10), facecolor='k')
plt.imshow(wordCloud)
[18]: <matplotlib.image.AxesImage at 0x7f12509c9a10>
Data
+ Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

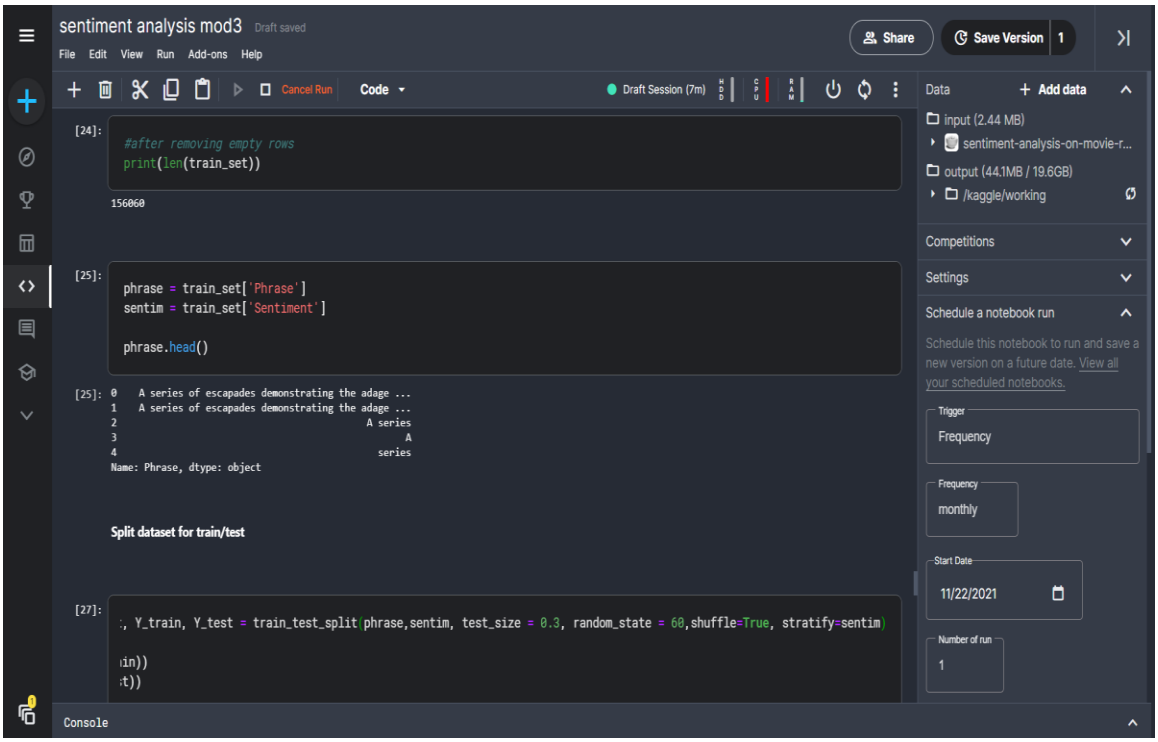

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
Most used words under neutral lable
[19]:
wordCloud = WordCloud(background_color="white", width=1600, height=800).generate(' '.join(neutral))
plt.figure(figsize=(20,10), facecolor='k')
plt.imshow(wordCloud)
[19]: <matplotlib.image.AxesImage at 0x7f12503cf950>
Data
+ Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

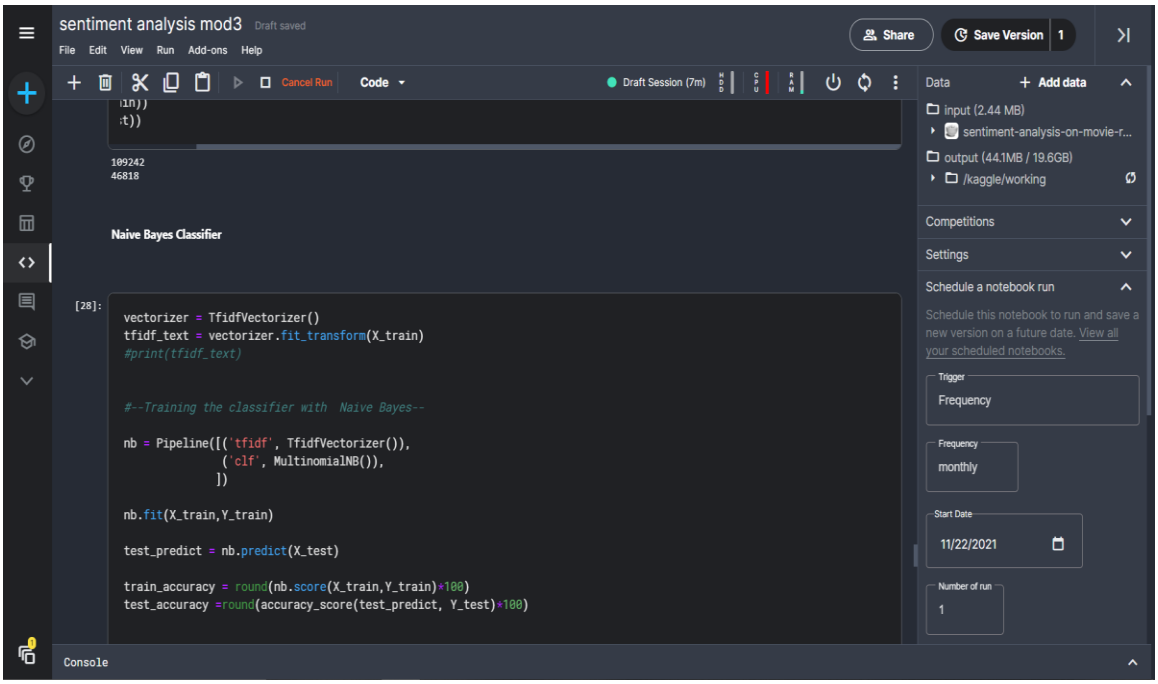

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
Most used words under somewhat postive lable
[20]:
wordCloud = WordCloud(background_color="white", width=1600, height=800).generate(' '.join(somewhat_pos))
plt.figure(figsize=(20,10), facecolor='k')
plt.imshow(wordCloud)
[20]: <matplotlib.image.AxesImage at 0x7f124a4d8850>
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

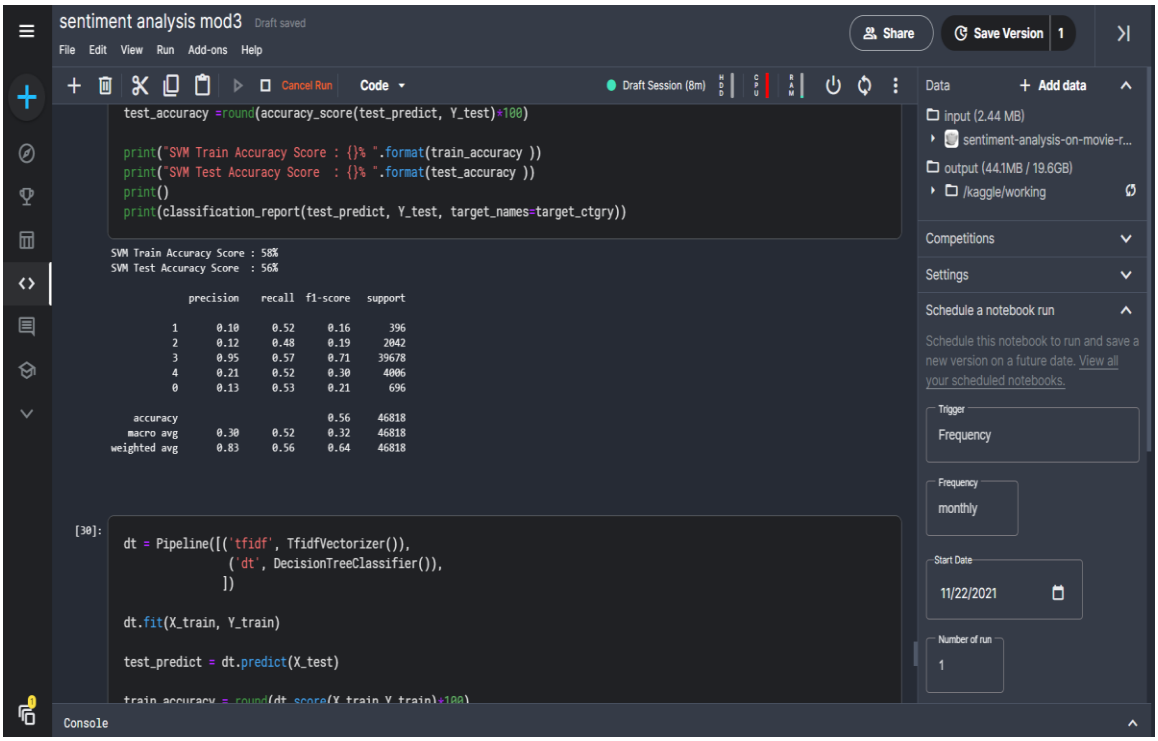

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
Most used words under positive lable
[21]:
wordCloud = WordCloud(background_color="white", width=1600, height=800).generate(' '.join(pos))
plt.figure(figsize=(20,10), facecolor='k')
plt.imshow(wordCloud)
[21]: <matplotlib.image.AxesImage at 0x7f124ac49550>
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

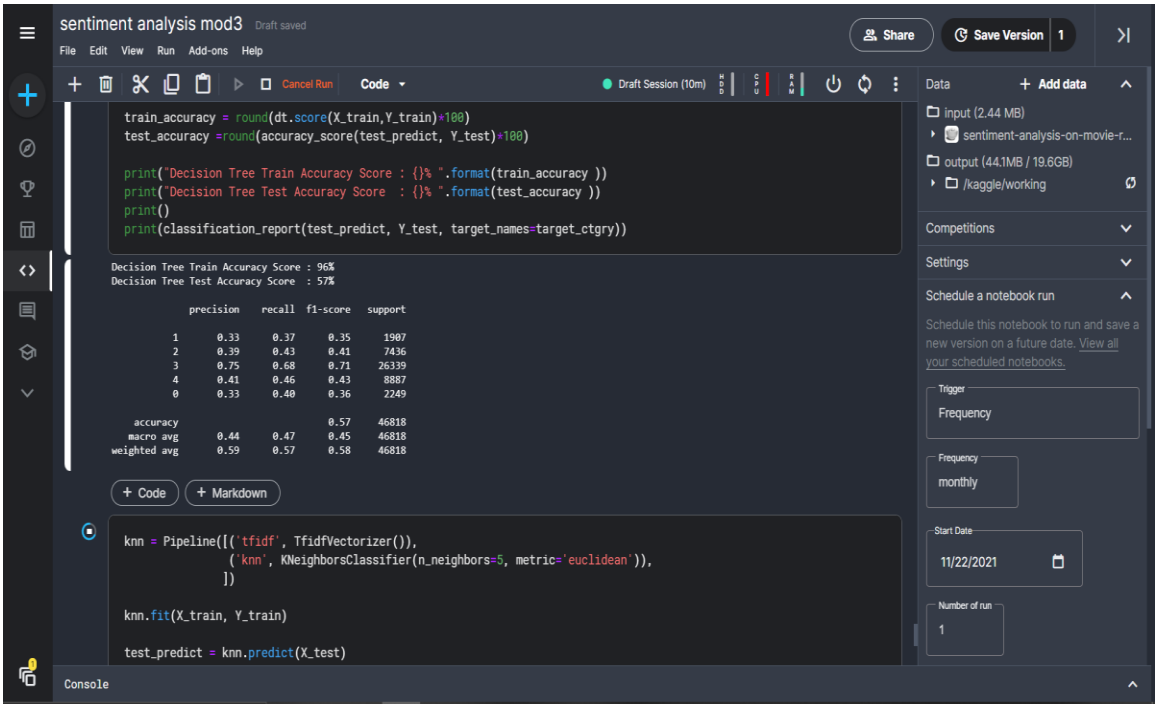

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
monthly
Start Date
11/22/2021
Number of run
1
[22]:
list_data = list(zip(phrase, sentim))
train_set = pd.DataFrame(list_data,columns = ['Phrase', 'Sentiment'])
train_set.head(20)
[22]:
Phrase Sentiment
0 A series of escapades demonstrating the adage ... 1
1 A series of escapades demonstrating the adage ... 2
2 A series 2
3 A 2
4 series 2
5 of escapades demonstrating the adage that what... 2
6 of 2
7 escapades demonstrating the adage that what is... 2
8 escapades 2
9 demonstrating the adage that what is good for ... 2
10 demonstrating the adage 2
11 demonstrating 2
12 the adage 2
13 the 2
14 adage 2
15 that what is good for the goose 2
Console

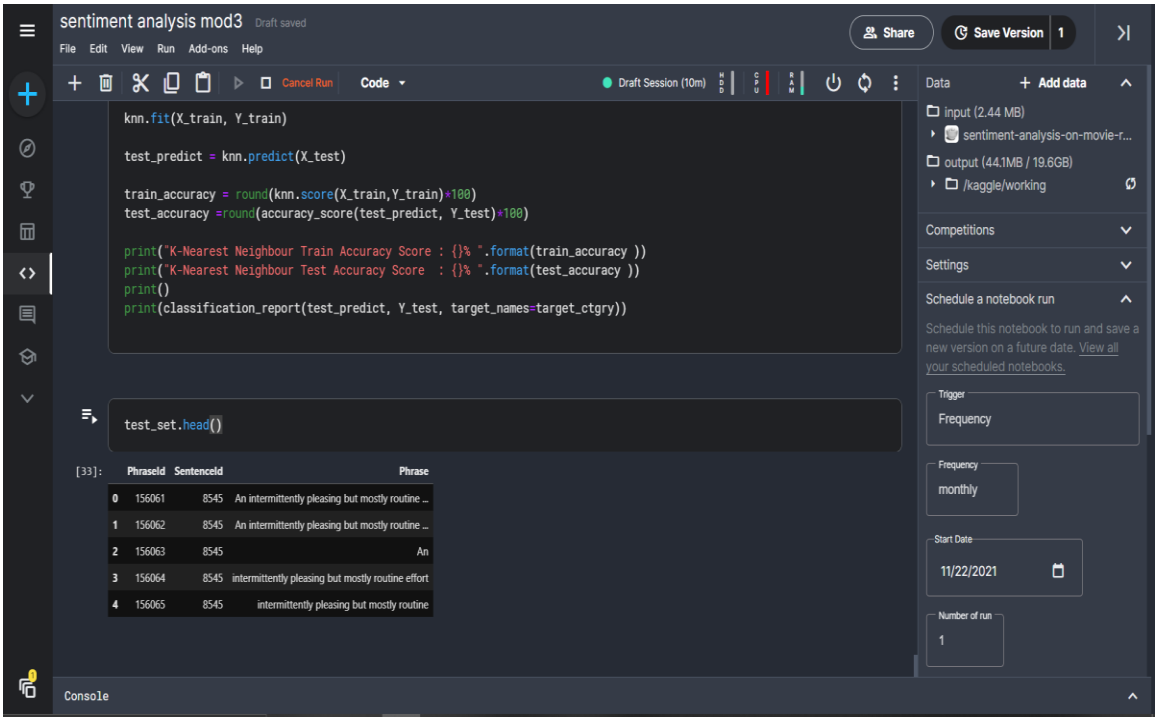

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
monthly
Start Date
11/22/2021
Number of run
1
after removing stop words some rows of the phrase colums has missing data. So we have to remove those rows
[23]:
#remove empty rows
train_set['Phrase'].replace('', np.nan, inplace=True)
train_set.dropna(subset = ["Phrase"], inplace=True)
train_set.head(20)
[23]:
Phrase Sentiment
0 A series of escapades demonstrating the adage ... 1
1 A series of escapades demonstrating the adage ... 2
2 A series 2
3 A 2
4 series 2
5 of escapades demonstrating the adage that what... 2
6 of 2
7 escapades demonstrating the adage that what is... 2
8 escapades 2
9 demonstrating the adage that what is good for ... 2
10 demonstrating the adage 2
11 demonstrating 2
12 the adage 2
Console


sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
[24]:
#after removing empty rows
print(len(train_set))
156060
[25]:
phrase = train_set['Phrase']
sentim = train_set['Sentiment']
phrase.head()
[25]: 0 A series of escapades demonstrating the adage ...
1 A series of escapades demonstrating the adage ...
2 A series
3 A
4 series
Name: Phrase, dtype: object
Split dataset for train/test
[27]:
, Y_train, Y_test = train_test_split(phrase,sentim, test_size = 0.3, random_state = 60,shuffle=True, stratify=sentim)
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1


sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (7m)
109242
46818
Naive Bayes Classifier
[28]:
vectorizer = TfidfVectorizer()
tfidf_text = vectorizer.fit_transform(X_train)
#print(tfidf_text)
#--Training the classifier with Naive Bayes--
nb = Pipeline([('tfidf', TfidfVectorizer()),
('clf', MultinomialNB()),
])
nb.fit(X_train,Y_train)
test_predict = nb.predict(X_test)
train_accuracy = round(nb.score(X_train,Y_train)*100)
test_accuracy =round(accuracy_score(test_predict, Y_test)*100)
Console
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1


sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (8m)
test_accuracy =round(accuracy_score(test_predict, Y_test)*100)
print("SVM Train Accuracy Score : {}% ".format(train_accuracy))
print("SVM Test Accuracy Score : {}% ".format(test_accuracy))
print()
print(classification_report(test_predict, Y_test, target_names=target_ctgry))
SVM Train Accuracy Score : 58%
SVM Test Accuracy Score : 56%
precision recall f1-score support
1 0.10 0.52 0.16 396
2 0.12 0.48 0.19 2042
3 0.95 0.57 0.71 39678
4 0.21 0.52 0.30 4006
0 0.13 0.53 0.21 696
accuracy 0.56 46818
macro avg 0.30 0.52 0.32 46818
weighted avg 0.83 0.56 0.64 46818
[30]:
dt = Pipeline([('tfidf', TfidfVectorizer()),
('dt', DecisionTreeClassifier()),
])
dt.fit(X_train, Y_train)
test_predict = dt.predict(X_test)
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

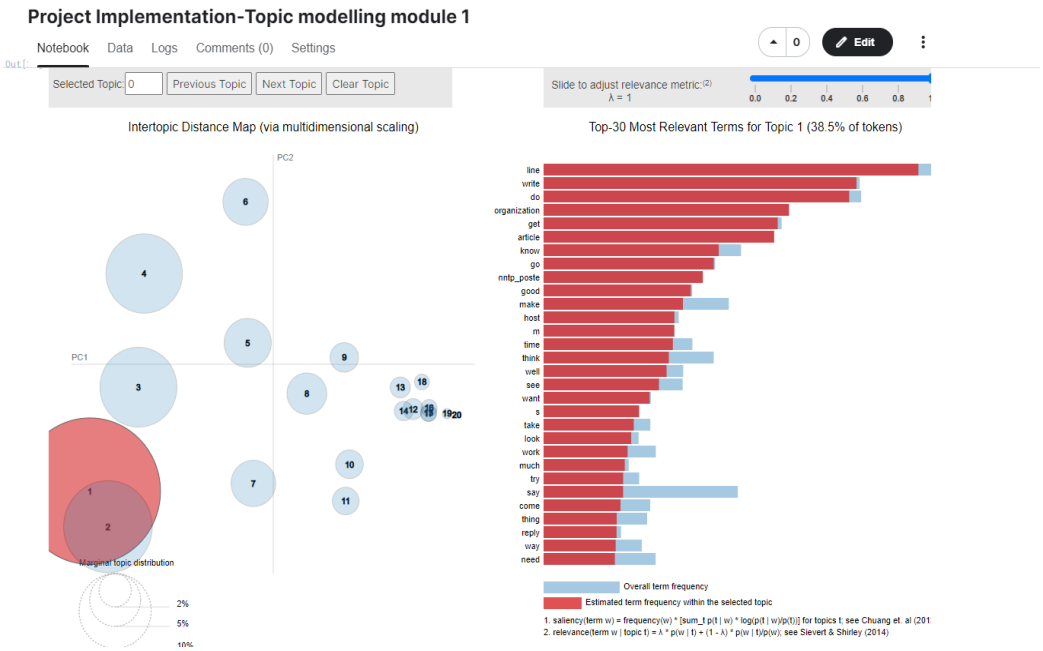

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (10m)
train_accuracy = round(dt.score(X_train,Y_train)*100)
test_accuracy =round(accuracy_score(test_predict, Y_test)*100)
print("Decision Tree Train Accuracy Score : {}% ".format(train_accuracy))
print("Decision Tree Test Accuracy Score : {}% ".format(test_accuracy))
print()
print(classification_report(test_predict, Y_test, target_names=target_ctgry))
Decision Tree Train Accuracy Score : 96%
Decision Tree Test Accuracy Score : 57%
precision recall f1-score support
1 0.33 0.37 0.35 1907
2 0.39 0.43 0.41 7436
3 0.75 0.68 0.71 26339
4 0.41 0.46 0.43 8887
0 0.33 0.40 0.36 2249
accuracy 0.57 46818
macro avg 0.44 0.47 0.45 46818
weighted avg 0.59 0.57 0.58 46818
+ Code
+ Markdown
knn = Pipeline([('tfidf', TfidfVectorizer()),
('knn', KNeighborsClassifier(n_neighbors=5, metric='euclidean')),
])
knn.fit(X_train, Y_train)
test_predict = knn.predict(X_test)
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

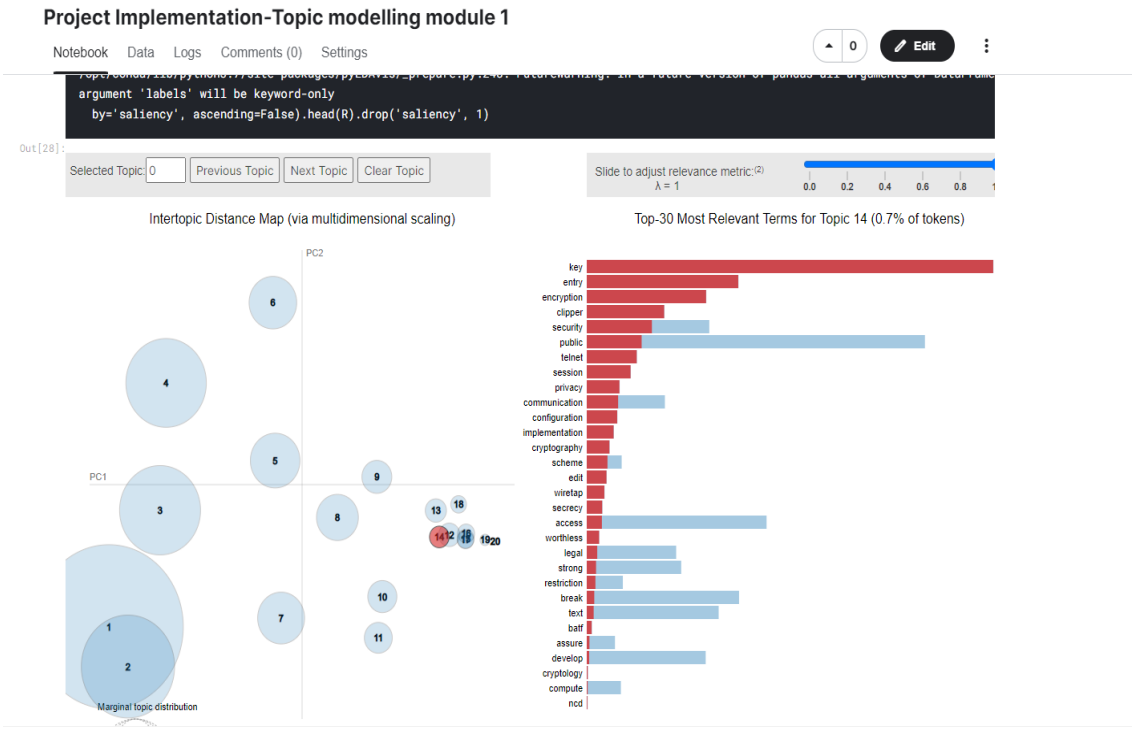

sentiment analysis mod3
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 1
Cancel Run
Code
Draft Session (10m)
knn.fit(X_train, Y_train)
test_predict = knn.predict(X_test)
train_accuracy = round(knn.score(X_train,Y_train)*100)
test_accuracy =round(accuracy_score(test_predict, Y_test)*100)
print("K-Nearest Neighbour Train Accuracy Score : {}% ".format(train_accuracy))
print("K-Nearest Neighbour Test Accuracy Score : {}% ".format(test_accuracy))
print()
print(classification_report(test_predict, Y_test, target_names=target_ctgry))
test_set.head()
[33]:
PhraseId SentenceId Phrase
0 156061 8545 An intermittently pleasing but mostly routine ...
1 156062 8545 An intermittently pleasing but mostly routine ...
2 156063 8545 An
3 156064 8545 intermittently pleasing but mostly routine effort
4 156065 8545 intermittently pleasing but mostly routine
Data
Add data
input (2.44 MB)
sentiment-analysis-on-movie-r...
output (44.1MB / 19.6GB)
/kaggle/working
Competitions
Settings
Schedule a notebook run
Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks.
Trigger
Frequency
Frequency
monthly
Start Date
11/22/2021
Number of run
1
Console

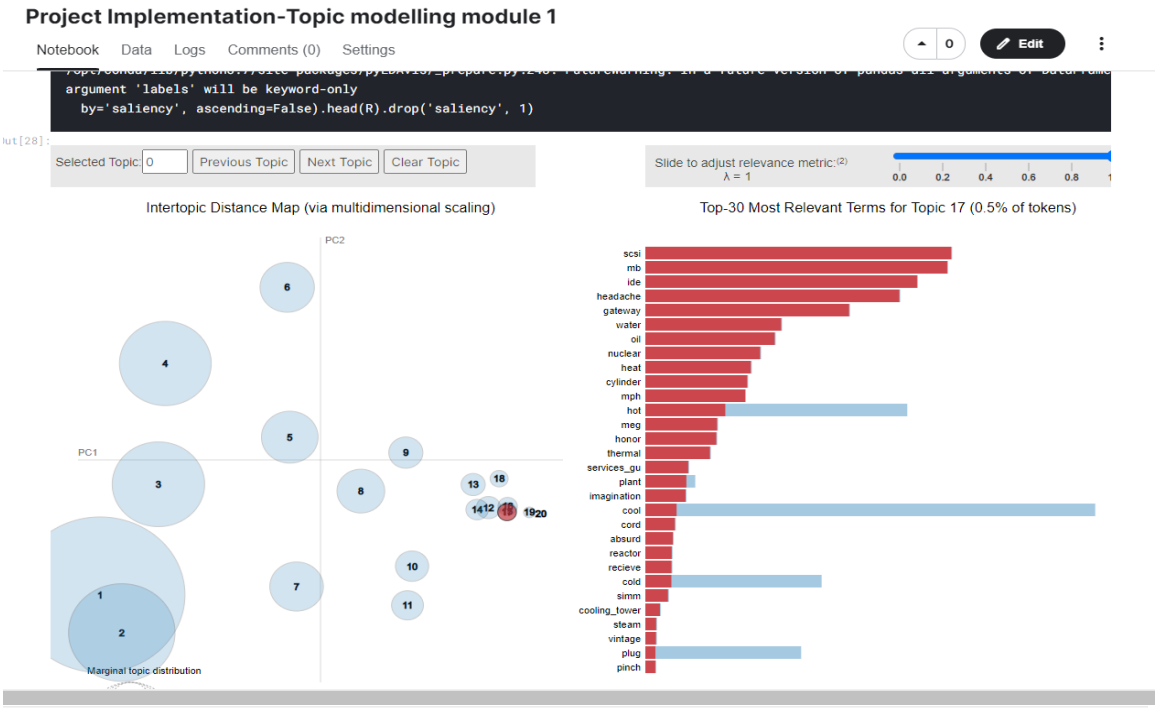
6. DISSCUSSION OF RESULTS

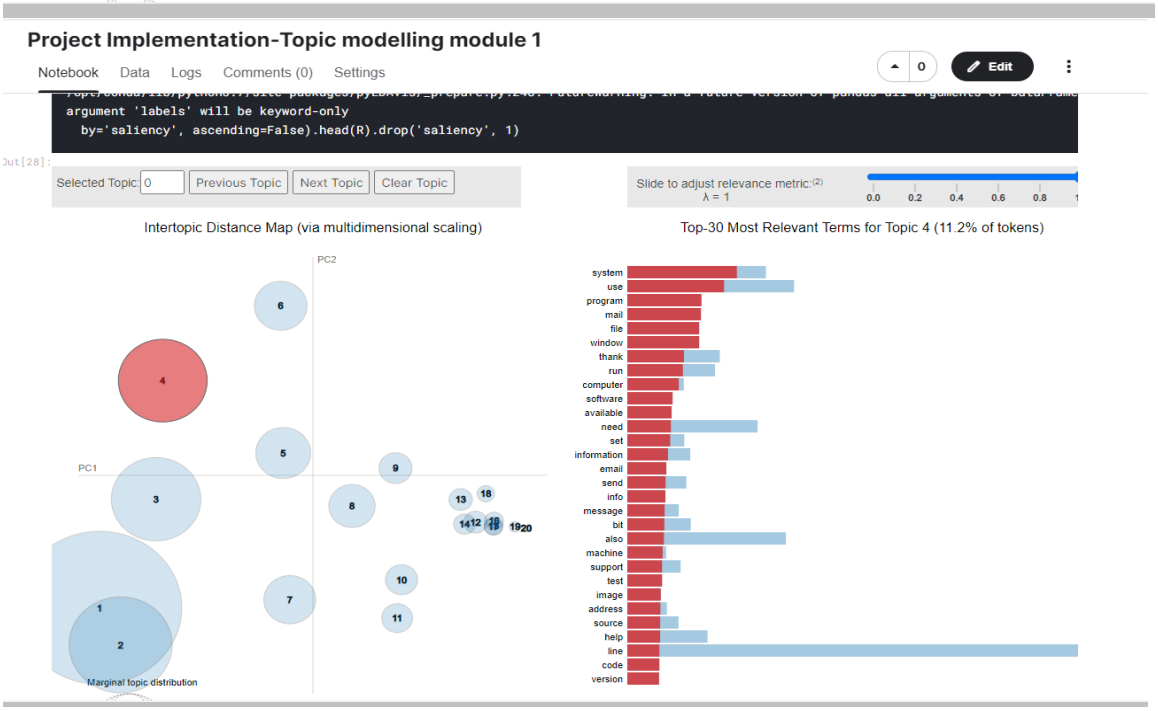
6.1 MODULE 1 OUTPUT

The output of module 1 is interactive I am attaching random 3 possibilities of output

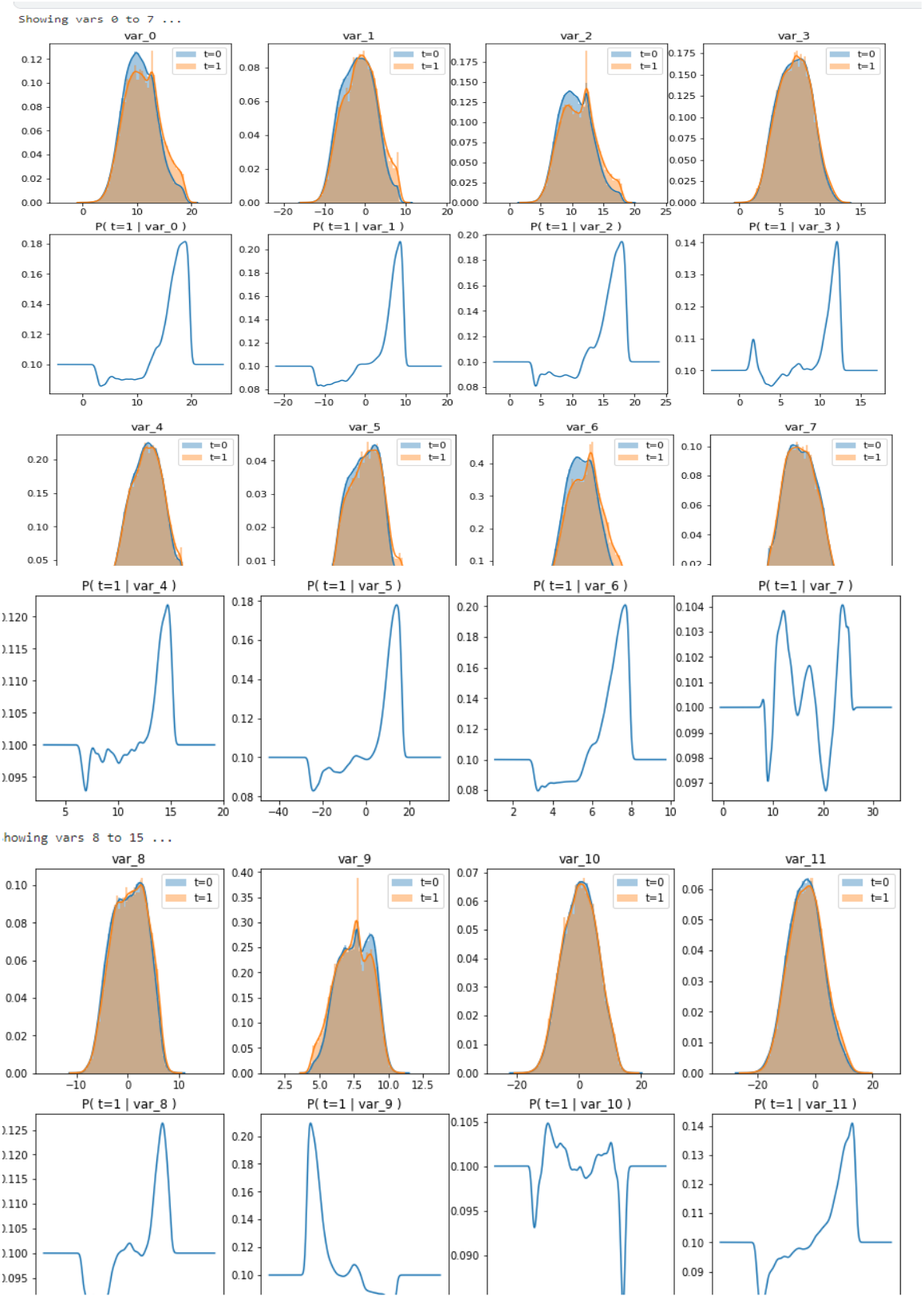


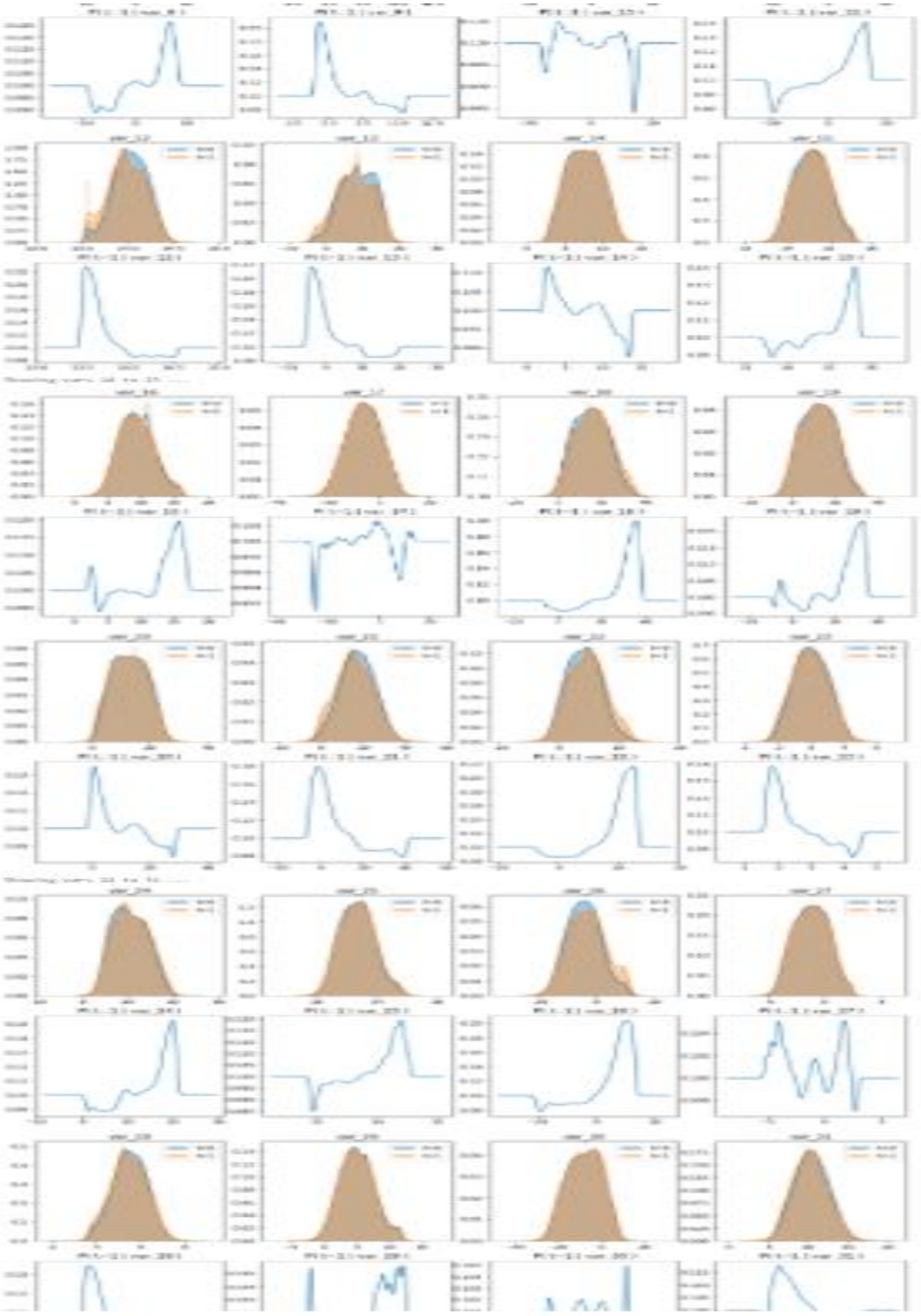



Project Implementation-Topic modelling module 1
Notebook
Data
Logs
Comments (0)
Settings
0
Edit
argument 'labels' will be keyword-only
by='saliency', ascending=False).head(R).drop('saliency', 1)
Out[28]:
Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
Intertopic Distance Map (via multidimensional scaling)
Top-30 Most Relevant Terms for Topic 17 (0.5% of tokens)
PC2
PC1
Marginal topic distribution
scsi
mb
ide
headache
gateway
water
oil
nuclear
heat
cylinder
mph
hot
meg
honor
thermal
services_gu
plant
imagination
cool
cord
absurd
reactor
recieve
cold
simm
cooling_tower
steam
vintage
plug
pinch

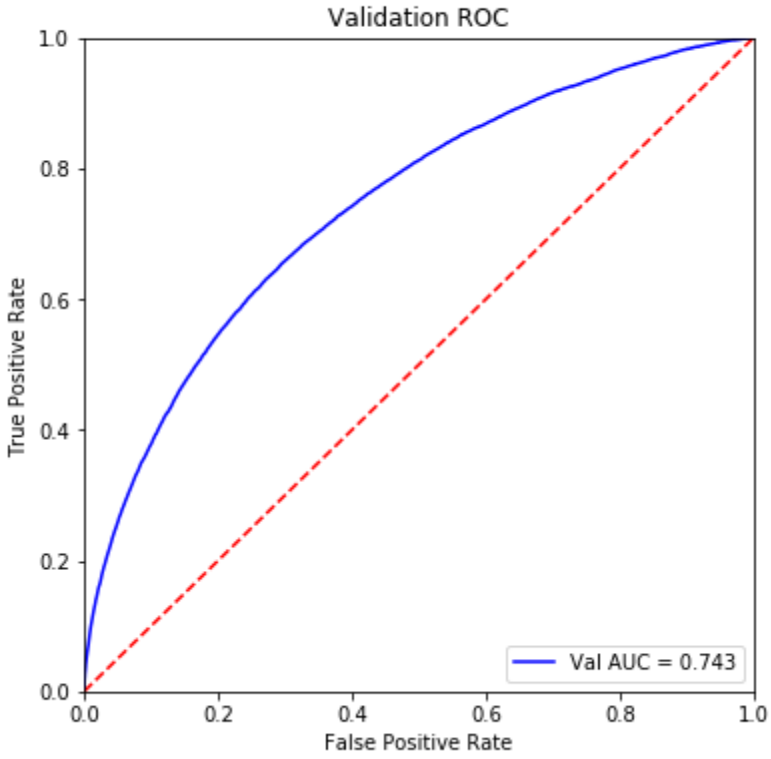

Project Implementation-Topic modelling module 1
Notebook
Data
Logs
Comments (0)
Settings
0
Edit
argument 'labels' will be keyword-only
by='saliency', ascending=False).head(R).drop('saliency', 1)
Out[28]:
Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
Intertopic Distance Map (via multidimensional scaling)
Top-30 Most Relevant Terms for Topic 4 (11.2% of tokens)
PC2
PC1
Marginal topic distribution
system
use
program
mail
file
window
thank
run
computer
software
available
need
set
information
email
send
info
message
bit
also
machine
support
test
image
address
source
help
line
code
version

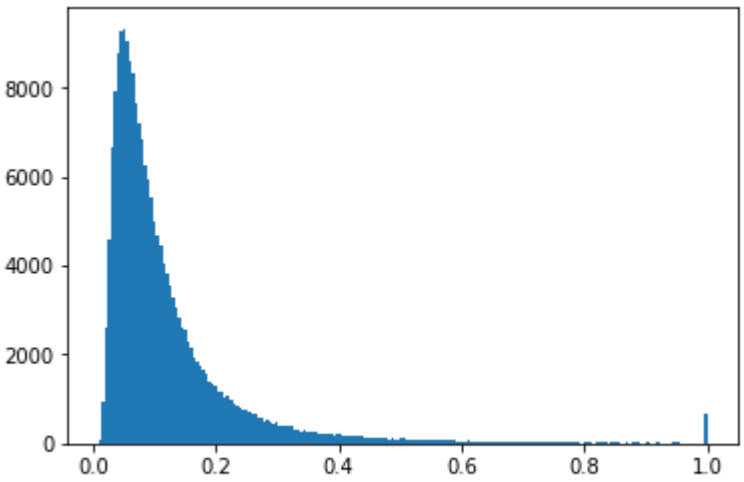
6.2 MODULE 2 OUTPUT





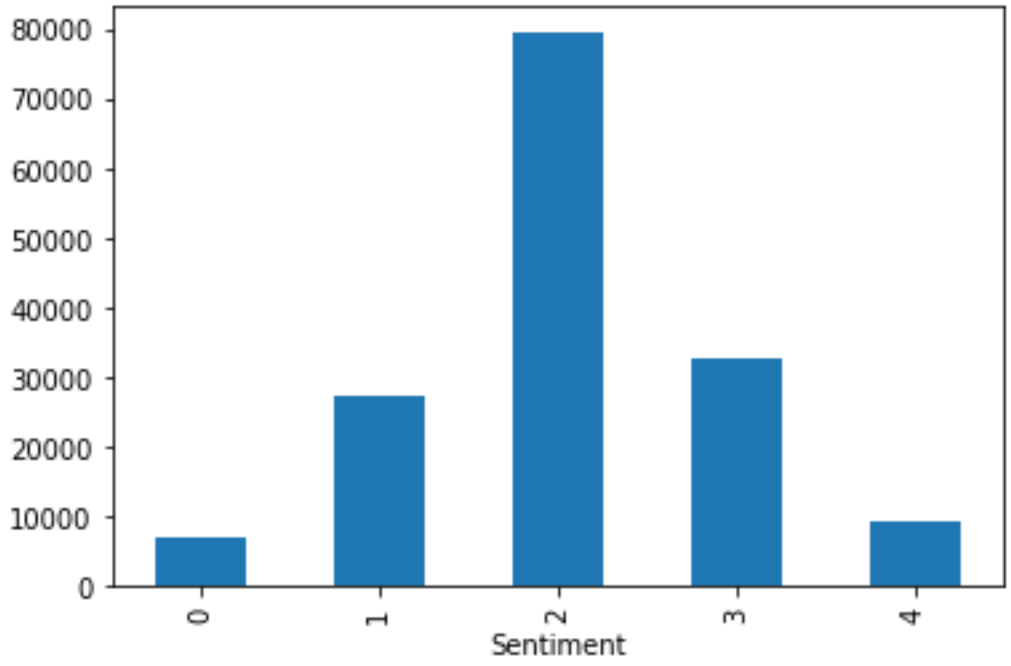
```
plt.show()
```

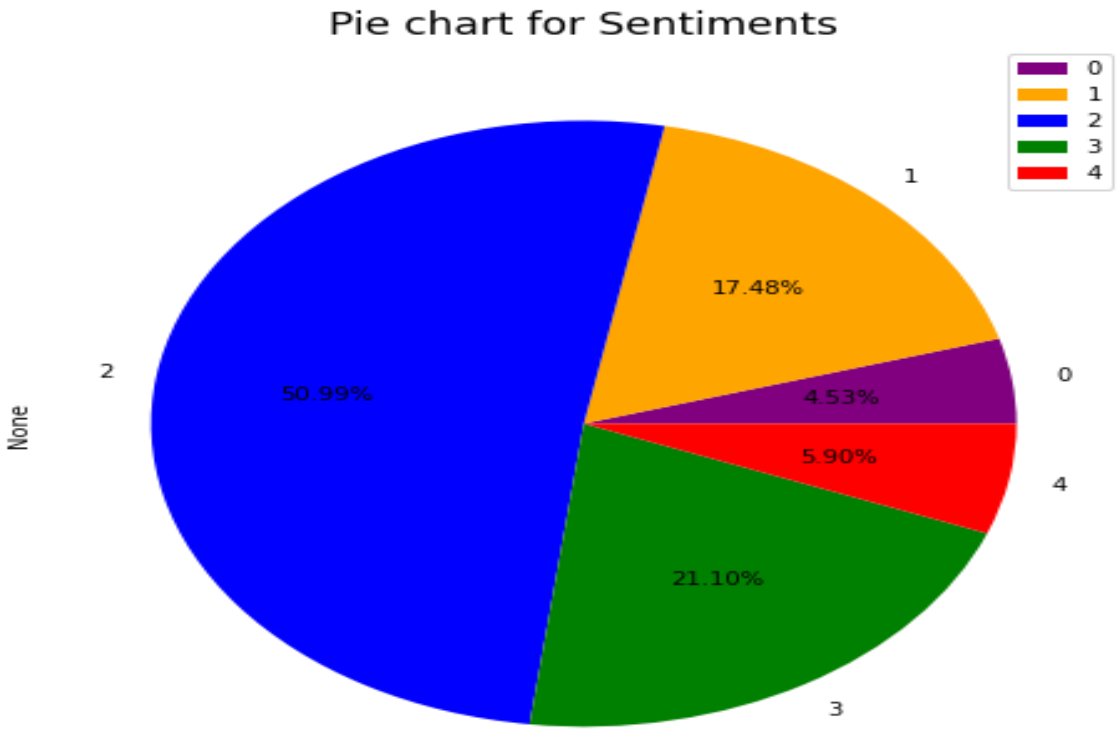


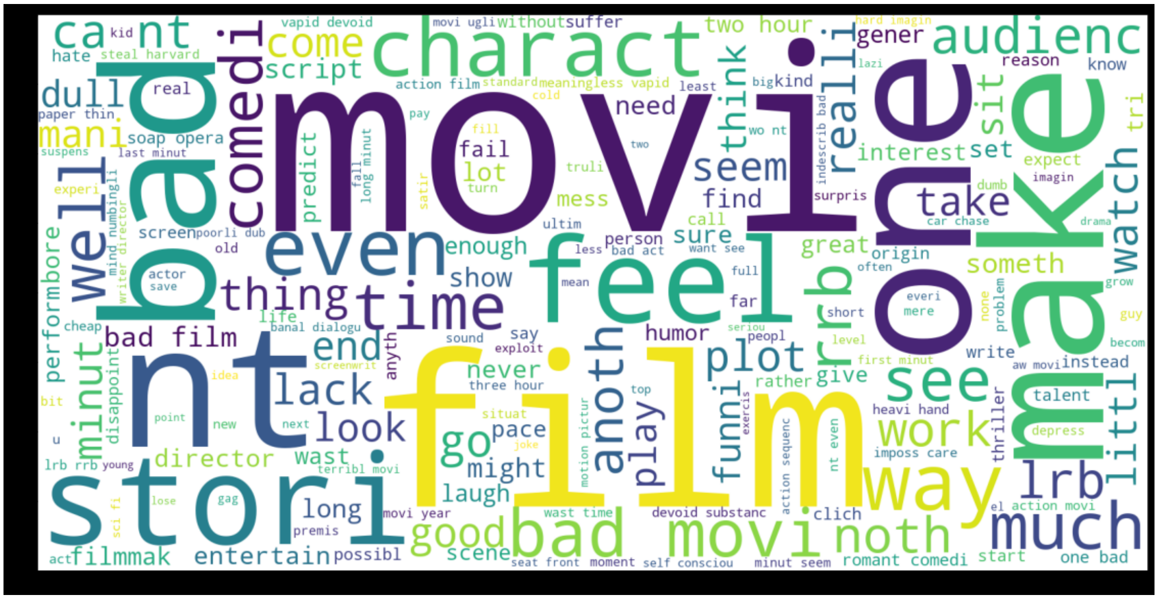


+ Code + Markdown

6.3 MODULE 3 OUTPUTS

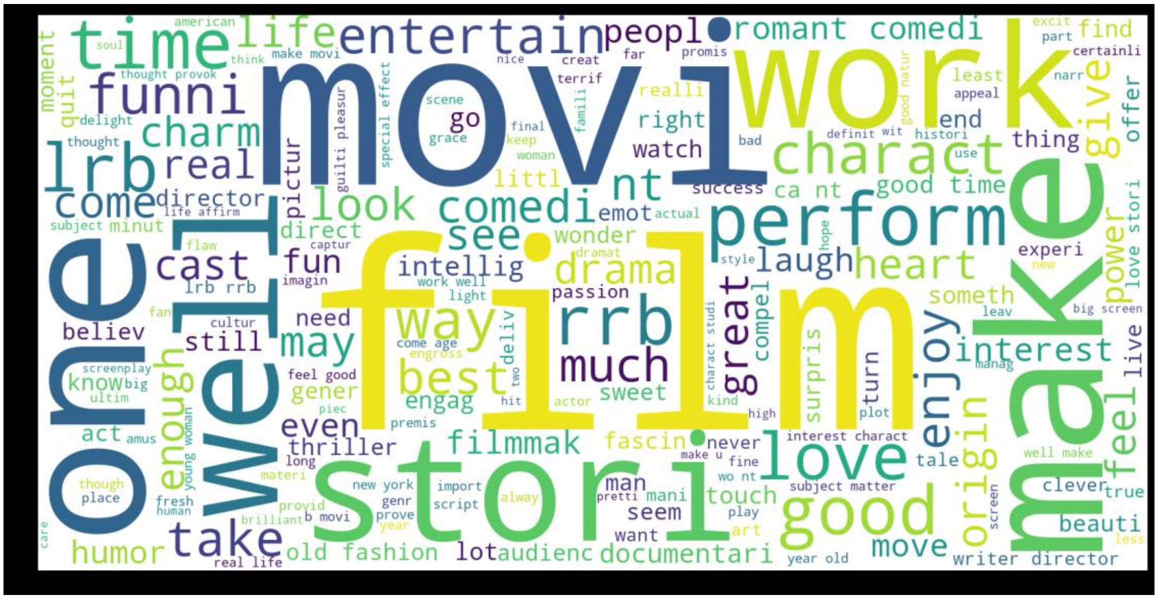


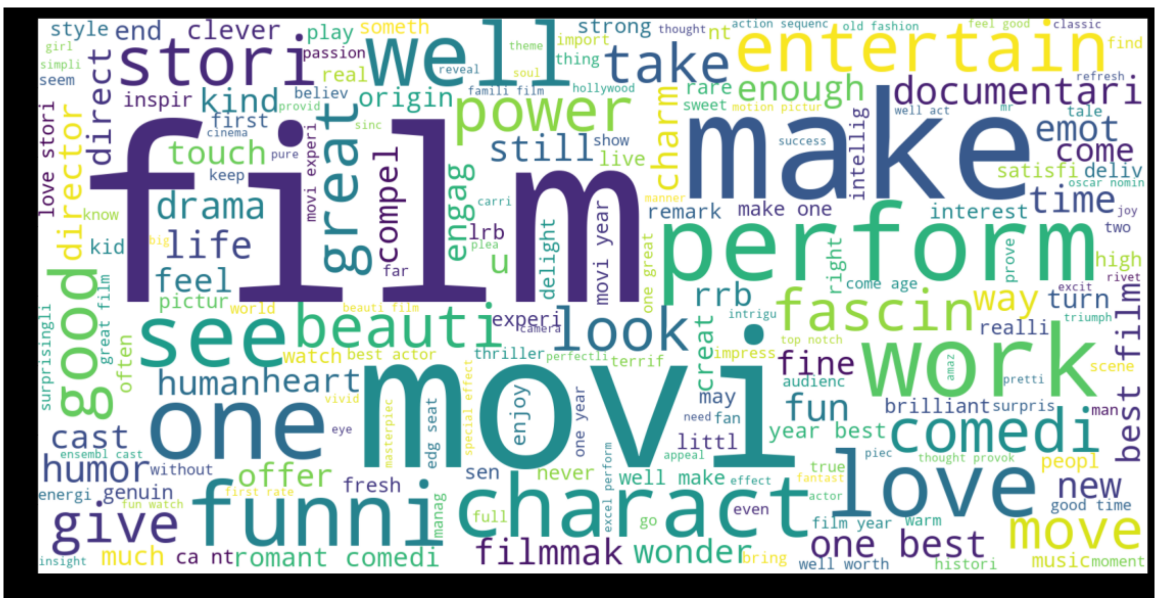



movi
film
one
make
bad
nt
stori
feel
charact
even
time
well
comedi
thing
bad movi
much
noth
way
audienc
see
plot
littl
work
lrb
anoth
lack
look
go
good
bad film
minut
director
entertain
filmmak
script
come
dull
mani
realli
think
take
watch

film
movi
one
make
feel
nt
much
charact
stori
way
may
ca nt
play
rrb
seem
even
bad
well
littl
time
look
work
come
good
thing
comedi
long
plot
go
give
lrb rrb

movi
film
one
make
nt
time
stori
charact
much
littl
life
even
rrb
work
well
seem
feel
look
see
may
way
take
director
perform
audienc
watch
lrb
give
romant comedi
writer director





These are the entire outputs of the 3 Modules of our project

6.1.1 MODULE 1 CODE RESULT:

The actual data set is json dataset which has different number of mails each email talks about different topics the main aim is to collect the data and preprocess the data by using different nltk libraries and then the ultimate result is that we get an interactive plot representing the first 20 email and the distribution frequency of each word in the email, form the plot we will be able to understand what is the email actually about just by looking at the output this is very interactive so that anyone can easily understand the result just by looking at the output.

Data visualization usually plays very important role to understand the data.

6.2.1 MODULE 2 OUTPUT:

From module 2 Naive Bayes to predict Santander Customer transactions. Since we achieved an accurate score of 0.899 (which rivals other methods that capture interactions), this helps us to demonstrate that there is little or I think no interaction between the 100 variables. Besides in this kernel we observed some fascinating EDA which provide insights about the variables. There are different other insights to discover.

6.3.1 MODULE 3 OUTPUT:

We have taken a dataset and performed the different sentimental analysis function to understand the data and tried focusing on what areas is the data actually focusing, being the movie data set we got to understand it is talking about the films and seeing the word cloud according to different queries we were able to identify the which is the result by observing the largest size of the word in the give word cloud.

7. CONCLUSION

The purpose of the entire project is to explore the different datasets, choose the accurate model and fit the dataset according to the model, predict the outputs as per the datasets you could get proper understanding of the dataset, in the modern era of the big unstructured data we need to understand the appropriate models and process which helps us accurately understand the data and use it efficiently.

On this particular topic models, we need to report the initial results understand the data by topic modelling we have understood that any millions of data can easily be preprocessed and get to know what the data is all about just by using this topic modelling and different probabilistic methods such LDA, PSLDA and many more.

By working on the sentimental analysis there actually many approaches that analyze the sentiments but they hardly work on the different grammatical errors, using sentimental analysis we can actually know the intention behind the phrase, we can understand if there any reviews it helps us a lot understand in the context of customers.

Though there are many advantages to work with sentimental analysis we do have different challenges on the sentimental analysis

1. Incremental approach
2. Parallel computing of the data
3. Sarcasm – major challenge
4. Behavior of the data
5. Grammatical errors
6. Review segmentation
7. Refinement or updating lexicons
8. Handling the noisy data

Massive users do share their opinions and their intentions through the social media, it being a very vital platform of the modern era, with the big data flowing each second, research areas trying to work on the new techniques which can accurately work on the data like sentimental analysis and semantic analysis work on the data providing efficient results of the data.

The traditional classification data does not actually provide or work that accurate of latest technologies and methods such as machine learning Natural language processing, deep learning, artificial intelligence and many more.

8. REFERENCES

1. Sentiment Analysis for Social Media: A Survey - Harshali P. Patil

2. Text Sentiment Analysis: A Review - Ronglei Hu, Lu rui

3. Sarcasm Detection of Tweets: A comparative Study - Tanya Jain, Nilesh Agarwal

4. L D A Based Topic Modeling of Journal Abstracts – P.Anupriya

5. Sentimental Analysis on Product Reviews B Ramya Sree1 , Potlapally Sai Vivek2 , Ram Gopal Adapa3 , Y Harika Devi4 , Manthena Prudhvi Raju5 , V Divyavani

6. Topic Modelling for Short Text Jocelyn Mazarura, Alta de Waal, Frans Kanfer and Sollie Millard Department of Statistics, University of Pretoria Centre for Artificial Intelligence Research (CSIR Meraka) South Africa

7. T. Iwata H. Sawada, Topic model for analyzing purchase data with price information. Data Mining and Knowledge Discovery, 26(3), pp.559-573, 2013

8. C. Lin H. Yulan, ”Joint sentiment/topic model for sentiment analysis.” In Proceedings of the 18th ACM conference on Information and knowledge management, pp. 375-384. ACM, 2009.

9. L. Hong D.D Brian, “Empirical study of topic modeling in twitter.” In Proceedings of the First Workshop on Social Media Analytics, pp. 80-88. ACM, 2010.

Topic Modelling Using LDA and Sentimental Analysis

By: NAVEEN N

As of: Nov 22, 2021 12:25:02 PM
6,699 words - 68 matches - 30 sources

Similarity Index

14%

Mode: Similarity Report

paper text:

Topic Modelling Using LDA and Sentimental Analysis A Main

Project Report Submitted in the partial fulfillment of the requirements for The award of the degree of 2
Bachelor of Technology In Department of Computer Science Engineering By

Namburi Bhavana Lalitha Janaki 180030571 Repaka Surya Rasagna 180031143

Under the supervision of DR .SASMITA PADHY **Department Of** Computer Science Engineering 2
K L E F, Green Fields

, Vaddeswaram- 522502, Guntur (District), Andhra Pradesh, India. November, 2021. ii iii Declaration The Project Report entitled " Topic Modelling Using LDA and Sentimental Analysis" is a record of bona fide work of Namburi Bhavana Lalitha Janaki , Repaka Surya Rasagna

submitted in partial fulfillment for the award of B.Tech in Computer Science Engineering to the K L 2
University. The results embodied in this report have not been copied from any other
departments/University/Institute

. Namburi Bhavana Lalitha Janaki 180030571 Repaka Surya Rasagna 180031143 iv

Certificate This is to certify that the Term Paper/Project Report entitled 9

"Topic Modelling Using LDA and Sentimental Analysis" is being submitted by Namburi Bhavana Lalitha Janaki, Repaka Surya Rasagna

submitted in partial fulfillment for the award of B.Tech in Computer Science Engineering to the K L 2
University is a record of bona fide **work carried out under our guidance and supervision. The**

results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of the Co-Supervisor Signature of

super visor

Signature of the HOD Signature of the External Examiner v **ACKNOWLEDGEMENT** On **the** 8 every aspect of **this project** report, We **would**

like to sincerely thank Our guide and super visor and co super visor of our department for the support they have actually extended for us, we are very thankful for our department faculty and CSE dept. HOD prof.HARI KIRAN VEGE. I would like to extend my heartfelt obligation to all the persons who ever are with us during the entire project completion period. Without their guidance, help, encouragement which they have provided us during all the journey, we would not have actually made heading this way in the project. We are very thankful and pay my gratitude to my faculty Dr.SASMITHA PADHY for her actual guidance and encouragement on the completion of my project in its presently. We extend my sincere gratitude to K L UNIVERSITY for giving us such a great opportunity. We would like to also acknowledge with deep sense of gratitude to our parents who have been with us during the entire journey supporting us morally and economically. At

last but not least I would like to thank my friends and some of **the** 21

faculty helping

me directly or indirectly for the **completion of** my **project** 8

. Any omission in this acknowledgement does not mean any lack of our gratitude. Thanking You, Namburi Bhavana Lalitha Janaki 180030571 Repaka Surya Rasagna 180031143. vi ABSTRACT Topic Modelling using LDA and Sentimental Analysis is the Project entitled as. Coming to the Project implementation the Project encloses of 3 Modules, module 1 deals with the topic modelling using LDA whereas module 2 and 3 deals with the sentimental analysis. The 2 main algorithms which are used in this project are Latent Dirichlet Allocation (LDA) and Naïve Bayes Algorithm. Discussing about the module 1 about topic modelling using LDA algorithm, topic modelling is a process in Natural Language Processing (NLP)

which is generally **used to train** the **machine learning models to** attain **the** 23

actual required results for the targeted objective. The process of selecting the words logically that actually belong to certain topic within a given document. This Topic Modelling provides a very great time and also effort saving benefits. Coming to LDA algorithm this is process of the

topic model and which **is used to classify text in a document to a particular topic it**	12

provides the information on the topic which is document is actually based on. To explain briefly regarding the LDA algorithm,

it builds a topic for each **document model and**	12

the respective topics for the

topic model, which is **modeled as Dirichlet** distributions. **The**	29

main core concept will be actually replaced by the Dirichlet allocations where the distribution is mainly over a probability simplex. We have taken a dataset comprising of different emails and concluded by identifying the various topics which is actual dataset is comprising of. With this we can easily identify what the mails are talking about. So like this we can work on large datasets to identify the topics hidden. Module 2 and 3 is about sentimental Analysis It is automated method process of translating the large volumes of the data which is unstructured into the most qualitative data to uncover the hidden patterns and the emotion of the data this is mostly used to study the data of social media, The main role is identifying opinionative data, it is used for the computational study of text analysis to find the subject and emotion hidden. vii CONTENTS OF THE REPORT S.NO CONTENT Page No 1. INTRODUCTION 8 2. LITERATURE SURVEY 12 3. THEORATICAL ANALYSIS 16 4. EXPERIMENTAL INVESTIGATIONS 23 5. EXPERIMENTAL RESULTS 25 6. DISCUSSIONS OF RESULTS 46 7. SUMMARY,CONCLUSIONS,RECOMMENDATIONS 55 8. REFERENCES 56 viii 1. INTRODUCTION Topic Modelling Using LDA and Sentimental Analysis is the project which is completed, Here by the 2 main pillars of this project is LDA Algorithm and Naïve Bayes Algorithm for the Topic Modelling and Sentimental Analysis. These two are respectively implemented with the given algorithms, for topic modelling we do generally have three most common techniques used:

1. Latent Semantic Analysis (LSA) 2. Probabilistic Latent Semantic Analysis (PLSA) 3. Latent Dirichlet Allocation (LDA) Being **topic modelling**	15

a powerful technique for the most of the documents for unsupervised Analysis of the large document collections. The large document collections are mostly unstructured data this is where the big data analytics comes into the picture. It is

very easy to use the large data sets to gain the insights by using topic modelling algorithm in further discussions it will be clearly pictured by the outputs and the procedure the algorithm is actually implemented. So the

topic models have a very **wide range of** the **applications like recommendation** of the tag, **categorization** of text, **extraction** of the keywords, **and** 10

the most important one the

similarity search in the broad fields of the areas of **text mining** , for the **information retrieval** 10

and all. The topic Modelling can conceive different Latent topics which are generally used

in text using the different **random variables and** the **structure** 10

of the entire procedure is with the posterior inference. 1.1 What is

a Topic model ? It **is** the **one that** does **automatically discovers** the **topics occurring in a collection of** the given **documents. A trained Model** that which is **used to discern** with different **topics in** the **new documents. The model** actually picks **out** different portion **of** the topics **which** covers **topics** 4

in the documents When we generally consider the large data such as Wikipedia and different huge data they all are combination of several topics millions of documents covering some thousands of the topics By using the topic modelling we can also discover some of the emerging topics as the documents can be written about them, considering news documents which keeps on being in a constant change with several topics the topic modelling can be used to identify the topics which the newspaper is actually talking about. 1.2 Latent Dirichlet Allocation The Algorithm LDA is the first pillar of this project introduction about this we will be able to get the actual advantages of applying this algorithm. This LDA approach it involves building the different statistical models of the topics and the documents. For example let us assume a topic to be modeled on the basis of the

probability distribution over a given **fixed set of** the **words. This** 4

helps in actually formalizing the different

set of words that come to the **mind** which are referred **to** the specific **topic** 4

. In this following way it helps the topics of the different document covers. The main goal of the learning and understanding the LDA a methodology is that it helps in discovering

from a corpus of documents , for the **good** number of **distributions of the various topics** , 4
for the **good** number of the **topic proportions in** different kinds of **documents**

. 1.2.1 Parameter for the LDA? The

most important parameter for the **LDA is** the **number of topics** 26

. Yes that's it this is one of the very important parameter that is to be passed along with the input that is the input dataset. 1.

2.2 LDA **Latent Dirichlet Allocation LDA** 30

generally represents the documents as the different

mixtures of the **topics that** do **spit out words with** the **certain** different **probabilities** . Let us consider **a** desired number **of topics** 20

with the set as a 'k' in the dimensional Dirichlet distribution. We can define the LDA algorithm as the proven oone which actually delivers accurate results for the topic modelling use cases. 1.2.3 How does it actually work? Step 1:

Decide the number of words which are of **N the document will have** based on the **Poisson distribution** 22

Step 2: now you have to

choose a topic mixture for the document (According to the **Dirichlet distribution over a fixed set of** already defined **k topics** 7

should be taken) Step 3: Generation Here we have to

generate each word Wi in the document by : 1. **Picking** up the **a** defined **topic** 11

2.

Using the topic to actually **generate the word itself** (based on **the** topics **multinomial distribution**) By **the following** 11

ways

assuming this generative model of different **collection of** the **documents, LDA then** concludes on trying **to** start backtracking **the documents to find a** good **set of topics that are likely to** be present in **the** generated **collection** 7

. 1.3 Sentimental Analysis Coming to the sentimental analysis it is one of the automated method process which is used for translating the large number of volumes of the mostly unstructured text into the qualitative data which is mostly used to discover the proper patterns, insights, trends of the data. The sentimental Analysis is spoken out of computational study of text analysis to find actually and then extract the subjective study of the data or the big data. It is used to analyze the different users review from the social media For example let us consider the amazon reviews of the products, initially considering a product with specific reviews as the input data by performing different computational methods of sentimental analysis. We would get an output of the emotions which are actually given in the form reviews on observing the result after performing computational study we will be able to understand the review of the product is it very good or good or bad or very bad, Based on that the recommendations of the products is computed for the customers. It is used to analyze the users review from the social media. 1.3.1 Sentimental Analysis for Social media In the social media the role of the sentimental analysis is generally identifying the opinionative data, what is the opinion behind the data the, emotion, the opinion, the intention these are all the keywords to describe how the sentimental analysis actually works on the for the social media data. For example let us consider a twitter data set, taking the recent pandemic into consideration covid-19 tweets there were millions of tweets which are tweeted by the citizens expressing different kind of emotions on the pandemic. When these tweets are taken into consideration and if the sentimental analysis study is performed. We will get different plots using different functions we can use different ways to understand the data the best way is the word cloud which gives the output in such a way that the most repeated topic in the word cloud is displayed bigger, by this we can conclude which topic is being more discussed on. In the different ways usually we can also understand the real hidden meaning behind the tweet, if he is angry because of the pandemic, or feeling sad because of this kind of situation occurred, or he is happy for staying at home different emotions are seen in each tweet finding the emotion behind the data is exactly what is called the sentimental analysis. They are definitely

some drawbacks sure on the sentimental analysis for sure, more over using this analysis by keeping in mind in when and where to use provides the accurate output than anything. 1.3.2 Data processing for SA The key concepts for processing a data is to: 1. Understand the data by studying it. 2. Get to know what kind of qualities does it possess 3. Advantages and disadvantages of the data usage method 4. Pick up the models which produces accurate results of the data According to me processing a data with accuracy is a beautiful art only when you start understanding the data the way it is when you know where and when to use the data you have almost completed the project the balance is just the implementation. This way we should put in more time and efforts to understand the data and the pick models to get accurate results of the data. Sentimental Analysis is a problem which is based on the text analysis this entire whole project is based on the text analytics whether it be topic modelling or sentimental analysis. The SA (Sentimental Analysis) is used in web classifying them according to their polarity which is used. The main is that the problem is based on the text analysis. It does helps in the hidden sentiments and the poly seamy of the data. 2. LITERATURE SURVEY The literature survey for this project is conducted on the based on the topic modelling and Sentimental Analysis. 1. TOPIC MODELLING 2.1.1 Introduction We have been seeing a lot

continuous development of the IT (**Information Technology**) branch **of** 5

science, based on information data as the internet is going on increasing in a rapid manner.

The major news websites and channels **have become the** most important **platform for** the 5 citizens **to get** the **news** or the **information** right away, **however the data is**

being rapidly increasing in the most unstructured format day by day. The data should always be preprocessed before using it, data being in an unstructured format will be very tough to understand using traditional data classification techniques, because they are unable to meet the requirements of the data when they are observed. So the research area working on the area of the text mining, which would help news or data text classification to meet the requirements, which should be able to classify the text as fast as possible and also it should be in a position to handle the dataset, and try to make the

accurate prediction of the data. **So** the **automatic classification can help to complete** the 3 proper **function**

which apt to predict the accurate results. It should be highly effective with higher efficiency so that it helps the organization

to save the **expenses, in the** modern **era of** the **big data, the research on automatic text classification plays an increasingly important role. Many of the classic text classification algorithms** 3

are mostly **proposed and** they being **widely used** all the way. **For example**

when we consider the naïve Bayes,

k-nearest neighbor and decision tree these are **different** classifiers **of** 18

the text. According to me each classifier do have different strengths and weakness. They are dependent on the decision step they take towards. The way the decision is being made by the classifier defines its strength overall. Due to the topic dimensionality of the different texts sometimes it is too high this the reason where we generally highlight using the topic modelling. This survey deals with the text classification usage very primarily using the topic modelling using LDA and also the sentimental analysis. 2.1.2Text categorization Fig 2.1 2.1.3

Latent Dirichlet Allocation (LDA) LDA **is a kind of topic model algorithm** which is **based on probability model, the algorithm** 3

deals with plurality of the topic mixture. It does uses the different ways which potential enough to find the hidden topic or the information of the topic in the very large scale of the

document set. The algorithm do **assumes each** and every **word** into **the corpus** 3

of the given information through certain kind of

probability to choose a particular **topic from** the chosen **subject with a certain to select a word** . The way **it** 3

does actually work is that it does

chooses a topic from the given **topic distribution and chooses** the specific **word from the** entire **word** 3

in the distribution The whole

main idea of the algorithm is 5

that is simply selects

a topic vector and topics that **determine the probability of**	5

the data that is being selected.

But the SLDA uses corresponding **continuous response values**	5

using the algorithm of linear regression, which surely cannot be the text data which is of the

multi class text data as the	5

very data which is taken as input. In simple words the topic modelling is the unsupervised learning method where as he text classification is the supervised learning method. 2. SENTIMENTAL ANALYSIS 2.2.1. Introduction The sentimental analysis works on the text based analysis which is most important part of the entire project is Natural Language Processing (NLP). This NLP is entirely used for the detection most of the time. Analysis and the mining are the 2 subjective portions of the text which do contain the views and also the emotions, preferences and the intentions of the user. The field of the sentimental analysis is a very interesting field in the research area. This sentimental

analysis is a very **big branch in the field of** the **natural language processing, The** entire **text analysis**	13

is based on how the text is categorized and that has great influence on the natural language processing, being most disciplinary field in the entire research it had been more concentrated in the research are by the most of the scholars everywhere. This sentimental analysis field not only involves the NLP (Natural Language Processing) filed but also many more computational areas, machine learning and many more algorithms that are part of the artificial intelligence. The most 3 main methods of this area are the analytical methods, the sentimental analysis as said will be based on the

machine learning algorithm **and** the **deep learning** algorithm. Coming **to the**	16

point that the

sentimental analysis which **is also called as** the **opinion mining is the**	16

method in which it automatically finds the opinion or the intention behind the text, but this could be one of the really challenging area in the research area, mainly in the data mining field for the Social media. We also have to look into the issue that how the social media will play the central part of this sentimental analysis, we know that whatever happens the social media reflects the society opinion for any kind of incident not only the news is given but also the entire people opinion on the incident can also be –observed, opinion on certain issues can be gathered, with this kind of expeditious development of the cycle web 2.O, people started increasing in showing their opinions on the people, this way day by day the handling odf this entire unstructured data does become complex. 2.2.2 Sentimental analysis Approaches When generally this is

formally stated by **their task and** is **interpreted** that **how to mathematically** inject the **social** media **context and** the topic **context** in **the basic** given **prediction model** 6

. This shall be investigate accordingly using different kinds and ways of the

correlations among the reserved **topics and calculated to measure them** as required. **The assumptions about** the given entire **social** 6

media data or the data context and topic a

context were both ways incorporated **by the hypothesis testing the** 6

results are extracted from the social media data. 1. LDA (Latent Dirichlet Allocation) 2. S-PLSA (Sentimental probabilistic approach) 3. ARSA (Auto regressive sentiment and quality aware model) 2.2.3 Methodology Fig 2.2 Whatever the

tweet can be a positive phrase a **negative phrase or** 1

there are also chance of It being a combination of both mixed emotions, This sentiment analysis is used to identify sarcasm which may exist actually which is contrast to the positive or the negative sentiments that are being given as the input. There are different challenges that are associated with the sentimental analysis one of the most important one among the is the sarcasm detection. The problem with this sarcasm is that it does not actually convey the meaning of the given phrase it does not show the pure or the actual intention. What does sarcasm actually mean is that a positive phrase may actually have the negative intention and vice versa this is where sarcasm is all about. So in identifying the entire meaning of the phrase there may be a misunderstanding due to sarcasm it is actually complicated when the

sentimental analysis faces this kind of challenges 3. THEORATICAL ANALYSIS 3.1 Topic Modelling 3.1.1 Short introduction The Topic modelling which is a text mining technique generally deals with underlying and the hidden topics in the large documents, besides it also enable the one for the

cluster documents on the **thematic similarity. This topic identification is** always **achieved** on the **each document** which ids **formed** by the **generative process** 19

. 3.1.2 Different Topic models We do have different kinds of topic models which are listed below 1. Mixture of multinomial 2. Gamma Poisson 3.

Latent Dirichlet Allocation 4. Probabilistic Latent Semantic Analysis 28

These are different approaches for the topic modelling there is such arise for the social media services which are twitter and Facebook etc. In comparison to long text short text are generally taken into the considerable for the problem when applying the different traditional topic models. When short texts are considered as far as concerned the LDA model is not much applicable for this, the accuracy of the performance is too low. Whereas the MM model will show accurate results when short text is considered it is a opposite to the performance of that of LDA. In this way there many advantages and also disadvantages of the different topic models. When a data with the corpus is considered the resultant output is: 3.2 Sentimental Analysis with Sarcasm Analysis 3.2.1 Introduction This sentimental analysis can be defined as the classification task performed on text data with a significant preprocessing it to classify into mainly with different classification tasks they are binary or multi classification classes. The sentimental analysis is actually an operation that comprises of different computational tasks entirely that leads to statistical approaches in some directions. It is one of the powerful machine learning application based on classifying the data or the text input into different classes, it is actually used mostly in analyzing the reviews such as customer reviews on a product, find the sentiment behind the review or the polarity of the information. Exactly when come to the point of how does sentimental analysis in recent times the latest technologies under the NLP (AI AND ML). Sentimental analysis is not just used for the social media data accordingly it is used for different applications such as recommendation system and feedback analysis. In this theoretical analysis we will be able to evaluate the predictions of sentiment classifiers, attached with different cases. When it is briefly taken into the context where the complexity of the data emerges neural network classifiers provides us the very reasonable way of showing us the high accuracy with efficient results. 3.2.2 Motivation applied for the sentimental Analysis

In the modern era of the big data 17

streaming in our daily lives, each and every person is connected with the social media in any way, so when we observe into this picture we could figure that there is lots of data that is being generated in any manner so, there should be something in a very strong position that can be able to handle such a big data. The big data analytics plays a very

crucial in our daily lives, more over when it is concerned social media data it is very complicated to analyze and calculate such data sentimental analysis as we discussed for sure shows us that a very form of tweets, posts, matter, photos, videos, is being streamed online All this raw text or unstructured data can be extracted and can be processed as far as the text is concerned there are specific algorithms and analysis that can be applied to understand the results. We can analyze the data in different formats some of the aspects are mentioned below: 1. Brand monitoring of the product 2. Product analysis 3. Market and the research analysis 4. The recommendation system 5. Social media monitoring 3.2.3 General Work Flow of Sentimental Analysis For every analytic task starts with the collection of the data. These days there are different social media platforms that are available such as twitter, Facebook, instagram and many more. These usually provide us very easy and wide open way to collect the data and use it for the preprocessing and working with the data. Real time twitter data the tweets can be extracted for the analysis and also with one twitter developer account and the tweepy which is one of the library in the python helps us to work accurately. Amazon reviews of the product that can be extracted by different techniques. We do have a specific flow for working with the sentimental analysis it is explained below: Step 1: Data collection Step 2: Preprocess the data Step 3: Feature extraction and selection Step 4: Sentimental classification Step 5: Evaluating Results and Error analysis FIG 3.1 3.2.4 Sarcasm Analysis

Sarcasm is a nuanced form of the pronunciation or **the** way **of** 14

communication where the actual intention behind the individual statement is opposite of what he/she have stated. What is actually implied is different from what actually is stated. There are major challenges of this sarcasm nature because of its different behavior or the

ambiguous nature. There is no prescribed definition for the word **sarcasm** actually it **is** 1

just opposite to the individual statement. They are different

slang words that **are being created** these days **on** 1

the social media which are used on these sites. There are thousands of words that are being created on the social media each word behind with different intentions. Existing of the corpus with different negative and positive sentiments should also be considered because can actual mean something which is different from in there. They

may not actually **prove to be accurate in detecting** the **sarcasm** of **the** 1

text or the data. There are different

difficulties and the tricky nature associated with this **sarcasm** 1

and

generally ignored during the **social network analysis. The** 1

sarcasm detection that **poses to be one of the most critical problems** 1

that should be considered as far as concerned which we usually need to overcome. These NLP based systems are that which supports the text summarization and the sentimental analysis Sarcasm can be simply defined as the “Positive sentiment attached to the negative situation” Steps present in this methodology: 1. Data extraction and cleaning 2. Seeding 3. Lexical classifier 4. Machine learning classifier 5. Emoticon extraction 6. Pos-neg recognition 7. Pragmatic classifier 8. Results 3.3 NLP Natural Language Processing of sentimental analysis based on project The

main aim of this sentimental **analysis is** used **to detect the polarity of the** given **text** 1

or the input. The misinterpreting of this sarcasm also poses a big challenge.

Natural language processing (NLP) is **a** subfield **of** the **computer science** 18

to be defined precisely. Where the artificial intelligence is concerned with the interactions. These brand monitoring systems or the NLP systems have been built in such a way that they rely on the social media data such as the

tweets posted on these social networks 1

. We will employ some of

machine learning algorithms such as 1. **Weighted Ensemble** 2. **Random Forest** 3. **Logistic Regression** 4. **Naive Bayes** 1

Random Forest in an ensemble learning technique that operates by constructing multiple decision trees by splitting the training dataset into smaller ones **and using each** part **for each tree. The output class is the mode or the mean average of each of the tree** Voted **Ensemble** Method: Voting Ensemble **based learning comprises of different machine learning classifiers. The output class is predicted by the weighted ensemble classifier by taking the average mean or mode of each of the individual comprising classifier. The weightage is given to each of the comprising classifier according to their individual accuracy** 1

. It uses different classifiers:

1. Naive Bayes 2. Logistic Regression 3. Random Forest 25

. FIG 3.2 3.5 Multinomial Naïve Bayes Multinomial Naïve Bayes it is one of the best approach in the context of the given text classification. This is generally fast, Reliable and that is better than the other classification algorithms with respective to the speed and also the accuracy. It just works on the basic simple concept of probability. Calculating

Probability of each **word in** each **document and the** calculation part **of** 27

the statistics. Let us now discuss about the multinomial Naïve Bayes algorithm steps Step 1: Training Step 2: Read the preprocessed data Step 3: Now we should create the empty dictionaries for both positive and negative Step 4: In the document for each word in each document of the dataset to be reverted Step 5: If the word that does not exist in the dictionary we should start adding it to dictionary Step 6: if the word exist we should start incrementing the value by one Step 7: Calculate the entire unique words in the given training dataset Testing Phase: Each word in testing document we should start calculating probability of that word with respect to both the given classes. Count is the count of the word with respect to class c and Σ count (w,c) count of all the words in the respective class. |V| is number that contains no of all the unique words in the dataset. Now we should start taking the product of all the obtained probabilities along with the all the previous calculated above for both the given respective classes. Now we should classify the document with the class that has higher probability that is found when compared to all the other given respective classes. Result: Accuracy of the result: Accuracy of naïve Bayes is nearly to 81.4%. This is one of the good accuracy for the given text classification problems due to the presence of the noise that is present in the datasets that cannot be removed easily. The Time taken for this MN Naïve Bayes algorithm is nearly to15 to 20 seconds. With the increase in dataset its accuracy increases. FIG 3.3 The above figure is of total 3 main contents 1. The positive reviews 2. The negative review 3. The total review By using the naïve Bayes algorithm we have obtained the above result. 4. EXPERIMENTAL INVESTIGATION REQUIREMENTS OF THE EXPERIMENT/ IMPLEMENTATION 1. The platform that is used to implement the code is KAGGLE 2. The language in which the experiment is carried out is PYTHON 3. The data sets that are used for the project code implementation are: 1. https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json 2. Santander customer transaction prediction 3. Sentiment Analysis on Movie reviews These are the 3 main data sets

we have worked on Module 1: Based on the topic modelling using LDA Module 2: Prediction using Naïve Bayes algorithm Module 3: Sentimental Analysis using Naïve Bayes and also different plot representations 4.1 KAGGLE ENVIRONMENT Kaggle environment, is one of the best environments for a data analyst or data science student to work with. It made me very easier to personally connect with data, and implement the data, It is one of the friendly environment for the developer there are many datasets to work with-it is online community for the people who work with data science as well as machine learning practitioners. Kaggle environment helps in offering the customizable environment for the developers, and also get chance to access Free GPU's and large storage of community to publish and code. 4.2 PYTHON I have implemented the code in python language, because it helps in using different packages which it has such as NLTK (Natural Language Tool Kit) libraries and different libraries which are very important for us to implement this project code. Different packages used in python language: 1. matplotlib.pyplot 2. tensorflow 3. unidecode 4. nltk 5. genism 6. pyLDAvis 7. logging 8. sklearn 9. word cloud These are not all the packages that are used but these are very important packages used to complete this project. 4.2.1 Terminology We would also like to discuss some of the important areas which we have implemented in our project: 1. Word cloud is a data visualization technique this is very helpful for showing the results of the data based on its size here the size of the word is directly proportional to the frequency or most repeated theme of the data set. 2. K-nearest neighbor algorithm this is one of the very easier machine learning algorithm which is preferably based on the supervised learning technique. This usually assumes the similarity between the data and the available state and it puts it into the new state, the most similar are available in the same state. 3. KNN (Euclidean distance) this is used most widely and it is very popular which is most probably set to default in the sklearn KNN classifies library in the python language. It is generally used to calculate the distance between nearest neighbors. 4. Target Density and Target Probability: Target Density is one of the down sampling method which is used to discard the events based on their local density, the main goal is to ensure the ultimate density of the proportion which is considered as the sample should fall in the considered range of density. Target Probability it specifies the probabilities that actually determines the elimination of the trail in this the general reasonable values ranges from 0.5 to 0.6. 5. Probability describes how likely an event is to occur just simply defines probability, it is another word of the possibility of certain event to take place. 6. Z-score helps us to know better how far the mean is form the certain data point, it is simply used to describe the values relationship with the values of mean 5. EXPERIMENTAL ANALYSIS In this Chapter we would like to share the implementation details of our project: 5.1 MODULE 1 IMPLEMENTATION 5.2 MODULE 2 IMPLEMENTATION 5.3 MODULE 3 IMPLEMENTATION 6. DISSCUSSION OF RESULTS 6.1 MODULE 1 OUTPUT The output of module 1 is interactive I am attaching random 3 possibilities of output 6.2 MODULE 2 OUTPUT 6.3 MODULE 3 OUTPUTS This are the entire outputs of the 3 Modules of our project 6.1.1 MODULE 1 CODE RESULT: The actual data set is json dataset which has different number of mails each email talks about different topics the main aim is to collect the data and preprocess the data by using different nltk libraries and then the ultimate result is that we get an interactive plot representing the first 20 email and the distribution frequency of each word in the email, form the plot we will be able to understand what is the email actually about just by looking at the output this is very interactive so that anyone can easily understand the result just by looking at the output. Data visualization usually plays very important role to understand the data. 6.2.1 MODULE 2 OUTPUT: From module 2 Naive Bayes to predict Santander Customer transactions. Since we achieved an accurate score of 0.899 (which rivals other methods that capture interactions), this helps us to demonstrate that there is little or I think no interaction between the 100 variables. Besides in this kernel we observed some fascinating EDA which provide insights about the variables. There are different other insights to discover. 6.3.1 MODULE 3 OUTPUT: We have taken a dataset and performed the

different sentimental analysis function to understand the data and tried focusing on what areas is the data actually focusing, being the movie data set we got to understand it is talking about the films and seeing the word cloud according to different queries we were able to identify the which is the result by observing the largest size of the word in the give word cloud. 7. CONCLUSION The purpose of the entire project is to explore the different datasets, choose the accurate model and fit the dataset according to the model, predict the outputs as per the datasets you could get proper understanding of the dataset,

in the modern era of the big unstructured **data we**	17

need to understand the appropriate models and process which helps us accurately understand the data and use it efficiently. On this particular topic models, we need to report the initial results understand the data by topic modelling we have understood that any millions of data can easily be preprocessed and get to know what the data is all about just by using this topic modelling and different probabilistic methods such LDA, PSLDA and many more. By working on the sentimental analysis there actually

many approaches that analyze the **sentiments but** they **hardly work on** the different **grammatical errors**	6

, using sentimental analysis we can actually know the intention behind the phrase, we can understand if there any reviews it helps us a lot understand in the context of customers. Though there are many advantages to work with sentimental analysis we do have different challenges on the sentimental analysis 1. Incremental approach 2. Parallel computing of the data 3. Sarcasm – major challenge 4. Behavior of the data 5. Grammatical errors 6. Review segmentation 7. Refinement or updating lexicons 8. Handling the noisy data Massive users do share their opinions and their intentions through the social media, it being a very vital platform of the modern era, with the big data flowing each second, research areas trying to work on the new techniques which can accurately work on the data like sentimental analysis and semantic analysis work on the data providing efficient results of the data. The traditional classification data does not actually provide or work that accurate of latest technologies and methods such as

machine learning Natural language processing, deep learning, artificial intelligence	24

and many more. 8. REFERENCES 1. Sentiment Analysis for Social Media: A Survey - Harshali P. Patil 2. Text Sentiment Analysis: A Review - Ronglei Hu, Lu rui 3. Sarcasm Detection of Tweets: A comparative Study - Tanya Jain, Nilesh Agarwal 4. L D A Based Topic Modeling of Journal Abstracts – P.Anupriya 5. Sentimental Analysis on Product Reviews B Ramya Sree1 , Potlapally Sai Vivek2 , Ram Gopal Adapa3 , Y Harika Devi4 , Manthena Prudhvi Raju5 , V Divyavani 6. Topic Modelling for Short Text Jocelyn Mazarura, Alta de Waal, Frans Kanfer and Sollie Millard Department of Statistics, University of Pretoria Centre for Artificial Intelligence Research (CSIR Meraka) South Africa 7. T. Iwata H. Sawada, Topic model for analyzing purchase data with price information. Data Mining and Knowledge Discovery, 26(3), pp.559-573,

2013 8. C. Lin H. Yulan, ”Joint sentiment/topic model for sentiment analysis.” In Proceedings of the 18th ACM conference on Information and knowledge management, pp. 375-384. ACM, 2009. 9. L. Hong D.D Brian, “Empirical study of topic modeling in twitter.” In Proceedings of the First Workshop on Social Media Analytics, pp. 80-88. ACM, 2010. 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56

sources:

1 179 words / 3% - Crossref
Tanya Jain, Nilesh Agrawal, Garima Goyal, Niyati Aggrawal. "Sarcasm detection of tweets: A comparative study", 2017 Tenth International Conference on Contemporary Computing (IC3), 2017

2 129 words / 2% - Internet from 13-Nov-2021 12:00AM
www.coursehero.com

3 98 words / 2% - Internet from 26-Nov-2018 12:00AM
ijesc.org

4 65 words / 1% - Internet from 16-Aug-2020 12:00AM
towardsdatascience.com

5 51 words / 1% - Crossref
Zhenzhong Li, Wenqian Shang, Menghan Yan. "News text classification model based on topic model", 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 2016

6 49 words / 1% - Crossref
Harshali P. Patil, Mohammad Atique. "Sentiment Analysis for Social Media: A Survey", 2015 2nd International Conference on Information Science and Security (ICISS), 2015

7 42 words / 1% - Internet from 25-Sep-2018 12:00AM
blog.quizzicol.com

8 22 words / < 1% match - Internet from 15-Jun-2020 12:00AM
www.coursehero.com

9 12 words / < 1% match - Internet from 09-Nov-2021 12:00AM
www.coursehero.com

10 31 words / < 1% match - Crossref
P. Anupriya, S. Karpagavalli. "LDA based topic modeling of journal abstracts", 2015 International Conference on Advanced Computing and Communication Systems, 2015

11 24 words / < 1% match - Crossref
Badr Hirchoua, Brahim Ouhbi, Bouchra Frikh. "A new knowledge capitalization framework in big data context", Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, 2017

12 23 words / < 1% match - Crossref
Mukhsimbayev Bobur, Kuralbayev Aibek, Bekbaganbetov Abay, Fuad Hajiyev. "Anomaly Detection Between Judicial Text-Based Documents", 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT), 2020

13 15 words / < 1% match - Internet from 01-Jun-2020 12:00AM
www.semanticscholar.org

14 8 words / < 1% match - Internet from 11-Aug-2020 12:00AM
www.semanticscholar.org

15 18 words / < 1% match - Internet from 05-Dec-2019 12:00AM
repositori.uji.es

16 17 words / < 1% match - Crossref
"ICDSMLA 2019", Springer Science and Business Media LLC, 2020

17 17 words / < 1% match - Crossref
"Knowledge Science, Engineering and Management", Springer Science and Business Media LLC, 2016

18 16 words / < 1% match - Crossref
"Advanced Intelligent Systems for Sustainable Development (AI2SD'2018)", Springer Science and Business Media LLC, 2019

19 15 words / < 1% match - Internet from 02-Dec-2015 12:00AM
www.prasa.org

20 13 words / < 1% match - Crossref
Daiva Goštautaitė, Jevgenij Kurilov. "Comparative Analysis of Exemplar-Based Approaches for Students' Learning Style Diagnosis Purposes", Applied Sciences, 2021

21 12 words / < 1% match - Internet from 05-Mar-2021 12:00AM
docplayer.net

22 12 words / < 1% match - Internet from 16-Apr-2021 12:00AM
www.aensiweb.net

23 10 words / < 1% match - Crossref
Anbazhagan Mahadevan, Michael Arock. "Review rating prediction using combined latent topics and associated sentiments: an empirical review", Service Oriented Computing and Applications, 2019

24 9 words / < 1% match - Internet from 26-Apr-2019 12:00AM
danielschristian.com

25 9 words / < 1% match - Internet from 17-Aug-2020 12:00AM
link.springer.com

26 9 words / < 1% match - Internet from 26-Oct-2020 12:00AM
www.machinelearningplus.com

27 8 words / < 1% match - Internet from 03-Jun-2020 12:00AM
asp-eurasipjournals.springeropen.com

28 8 words / < 1% match - Internet from 27-Mar-2016 12:00AM
www.bmva.org

29 7 words / < 1% match - Crossref
"International Conference on Communication, Computing and Electronics Systems", Springer Science and Business Media LLC, 2020

30 6 words / < 1% match - Crossref
"Text, Speech, and Dialogue", Springer Science and Business Media LLC, 2013