MINI PROJECT - II

(2021-22)

Tweet sentiment analysis by ML

FINAL REPORT

Institute of Engineering & Technology

Team Members

Deepak

(171500092)

Kumar Ashutosh Sharma

(171500169)

Supervised By

Mr. VAIBHAV DIWAN

Technical Trainer

Department of Computer Engineering & Applications

ACKNOWLEDGEMENT

We take this opportunity to thank all those who have helped us in completing the project successfully.

We would like to express our gratitude to **Mr. Vaibhav Diwan**, who as our guide/mentor provided us with every possible support and guidance throughout the development of project. This project would never have been completed without his encouragement and support.

Our heartiest thanks to Dr. (Prof). **Anand Singh Jalal**, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

DECLARATION

We hereby declare that the work which is being presented in the

Mini Project **"Tweet sentiment analysis by ML"**in partial fulfilment of the requirements for mini project viva voice, is an authentic record of my own work carried under the supervision of "GLA UNIVERSITY MATHURA".

Signature of Candidate:

Name of Candidate: Deepak, Kumar Ashutosh Sharma

Roll. No.: 171500092,171500169

Course: B.Tech. (Computer Science & Engineering) Year: 3rd

Semester: VI

ABSTRACT

This project addresses the problem of sentiment analysis in twitter; that is classifying tweets according to the sentiment expressed in them: positive, negative or neutral. Twitter is an online micro-blogging and social-networking platform which allows users to write short status updates of maximum length 140 characters. It is a rapidly expanding service with over 200 million registered users - out of which 100 million are active users and half of them log on twitter on a daily basis - generating nearly 250 million tweets per day [20]. Due to this large amount of usage we hope to achieve a reflection of public sentiment by analysing the sentiments expressed in the tweets. Analysing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.

Contents

Chapter 1.

Introduction

1. Motivation

We have chosen to work with twitter since we feel it is a better approximation of public sentiment as opposed to conventional internet articles and web blogs. The reason is that the amount of relevant data is much larger for twitter, as compared to traditional blogging sites. Moreover the response on twitter is more prompt and also more general (since the number of users who tweet is analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. This could be done by analysing overall public sentiment towards that firm with respect to time and using economics tools for finding the correlation between public sentiment and the firm's stock market value. Firms can also estimate how well their product is responding in the market, which areas of the market is it having a favourable response and in which a negative response (since twitter allows us to download stream of geo-tagged tweets for particular locations. If firms can get this information they can analyze the reasons behind geographically differentiated substantially more than those who write web blogs on a daily basis). Sentiment response, and so they can market their product in a more optimized manner by looking for appropriate solutions like creating suitable market segments. Predicting the results of popular political elections and polls is also an emerging application to sentiment analysis. One such study was conducted by Tumasjan et al. in Germany for predicting the outcome of federal elections in which concluded that twitter is a good reflection of offline sentiment

2. Overview

This project of analyzing sentiments of tweets comes under the domain of “Pattern Classification” and “Data Mining”. Both of these terms are very closely related and intertwined, and they can be formally defined as the process of discovering “useful” patterns in large set of data, either automatically (unsupervised) or semiautomatically (supervised). The project would heavily rely on techniques of “Natural Language Processing” in extracting significant patterns and features from the large data set of tweets and on “Machine Learning” techniques for accurately classifying individual unlabelled data samples (tweets) according to whichever pattern model best describes them.

3. Objective

To implement an algorithm for automaticclassification of text into positive, negative or neutral. Sentiment Analysis to determine the attitude of themass is positive, negative or neutral towards thesubject of interest. Graphical representation of the sentiment

Chapter 2. Software Requirement Analysis

2.1 Problem Statement

1) A major benefit of social media is that we can see the good or bad things people say about the particular brand or personality

2) The bigger your company gets difficult it becomes to keep a handle on how everyone feel about your band for large companies with thousands of daily mentions on social ,media and news sites, it's extremely difficult to do this manually

3) To combat this problem, sentimental analysis software are necessary. The soft wares can be used to evaluate the people's sentiment about particular or personality.

2.2 Modules

- **Data**

To gather the data many options are possible. In some previous paper researches, they built a program to collect automatically a corpus of tweets based on two classes, "positive" and "negative", by querying Twitter with two type of emoticons:

● Happy emoticons, such as ":)", ":P", ":)" etc.

● Sad emoticons, such as ":(", ":'(", "=(".

- **Data type Info**

  To gather what data types in data set

- **Remove Pattern**

  Remove the repeating pattern

- **Remove the special character**

- **Remove twitter handler name**

- **Individual words considered as tokens**

- **Stem the word**

- **Visualize the frequent words**
  - Frequent words visualization for positive word
  - Frequent words visualization for negative word

- **Extract the hashtag**
  - Non - racist tweets

- **Input**
  - Feature extraction

- **Model training**
  - Testing

2.3 Specific Requirements

Requirement Analysis

Hardware Requirements Specification:

Processor	: Intel dual Core ,i5
Main Memory (RAM)	: 4 GB
Cache Memory	: 2 MB
Monitor	: 14-inch Colour Monitor
Keyboard	: 108 Keys
Mouse	: Optical Mouse
Hard Disk	: 500 GB

Software Requirements Specification:

A) Google Colab,Project Jupyter

B) Windows 7,8,10

2.4 Technologies and Tools used

PYTHON:-

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects

- Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

- Open source general-purpose language.

- Easy to interface with C/ObjC/Java

- Easy-ish to interface with C++ (via SWIG)

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL.capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.

There is a fact behind choosing the name Python. Guido van Rossum was reading the script of a popular BBC comedy series "Monty Python's Flying Circus". It was late on-air 1970s.
Van Rossum wanted to select a name which unique, sort, and little-bit mysterious. So he decided to select naming Python after the "Monty Python's Flying Circus" for their newly created programming language.

Python 1.0 (Launched in 1994)
Python 2.0 (Launched in 2000)
Python 2.2 (Launched in 2001)
Python 2.5 (Launched in 2006)
Python 3.0(Launched in 2008)
Python 3.6 (Launched in 2016)
Python 3.8 (Launched in 2019)

Applications of Python

There are lot more things you can do with Python.

- **Web Development**

  Python web frameworks are known for their security, scalability, and flexibility. To add to that, Python's Package Index comes with useful libraries like Requests, BeautifulSoup, Paramiko, Feedparser, and Twisted Python.

- **Game Development**

  Python comes loaded with many useful extensions (libraries) that come in handy for the development of interactive games. For instance, libraries like PySoy (a 3D game engine that supports Python 3) and PyGame are two Python-based libraries used widely for game development.

- **Scientific and Numeric Application**

  Python has become a crucial tool in scientific and numeric computing. In fact, Python provides the skeleton for applications that deal with computation and scientific data processing. Apps like FreeCAD (3D modeling software) and Abaqus (finite element method software) are coded in Python.

- **Artificial Intelligence and Machine learning**

AI and ML models and projects are inherently different from traditional software models. When we talk about AI/ML projects, the tools and technologies used and the skillset required is totally different from those used in the development of conventional software projects. AI/ML applications require a language that is stable, secure, flexible, and is equipped with tools that can handle the various unique requirements of such projects. Python has all these

qualities, and hence, it has become one of the most favored languages of Data Science professionals.

- **Desktop GUI**

Python not only boasts of an English-like syntax, but it also features a modular architecture and the ability to work on multiple operating systems. These aspects, combined with its rich text processing tools, make Python an excellent choice for developing desktop-based GUI applications.

- **Software Development**

Python packages and applications aim to simplify the process of software development. From developing complex applications that involve scientific and numeric computing to developing desktop and web applications, Python can do it all. This is the reason why Software Developers use Python as a support language for build control, testing, and management.

- **Enterprise- level**

Enterprise-level software or business applications are strikingly different from standard applications, as in the former demands features like readability, extensibility, and scalability. Essentially, business applications are designed to fit the requirements of an organization rather than the needs of individual customers.

- **Education program and training course**

Python has an extremely straightforward syntax that’s similar to the English language. It has a short learning curve and hence, is an excellent choice for beginners. Python’s easy learning curve and simplicity are the two main reasons why it is one of the most used programming languages in educational programs, both at beginner and advanced levels.

- **Language Development**

Over the years, Python's design and module architecture has been the inspiration behind the development of many new programming languages such as Boo, Swift, CoffeeScript, Cobra, and OCaml. All of these languages share numerous similarities with Python on grounds like object model, syntax, and indentation.

- **Operating Development**

Python is the secret ingredient behind many operating systems as well, most popularly of Linux distributions. Linux-based Ubuntu's Ubiquity Installer and Fedora and Red Hat Enterprise's Anaconda Installer are coded in Python. Even Gentoo Linux leverages Python Portage (package management system). Usually, Python is combined with the C programming language to design and develop operating systems.

- **Web Scraping Application**

Python is a nifty tool for extracting voluminous amounts of data from websites and web pages. The pulled data is generally used in different real-world processes, including job listings, price comparison, R&D, etc.

BeautifulSoup, MechanicalSoup, Scrapy, LXML, Python Requests, Selenium, and Urllib are some of the best Python-based web scraping tools.

- **Image processing and Graphic Design Application Conclusion**

Python also finds a unique use case in image processing and graphic design applications. The programming language is used globally to design and build 2D imaging software like Inkscape, GIMP, Paint Shop Pro, and Scribus. Also, Python is used in several 3D animation packages such as Blender, Houdini, 3ds Max, Maya, Cinema 4D, and Lightwave, to name a few.

Advantages of HTML:

- Improved Productivity
- Interpreted Language
- Dynamically Typed
- Free and open Sources
- Vast Libraries Support

Disadvantages:

- Slow speed
- Not memory efficient
- Weak in mobile computing
- Database access
- Runtime Errors

Libraries:

i) **numPy**:-

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

ii) **Pandas**:-

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

iii) Matplotlib:-

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

iv) **Seaborn:-**

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

v) NLTK:-

NLTK is a leading platform for building Python programs to work with human language data.

Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure

vi) Worldcloud:-

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.

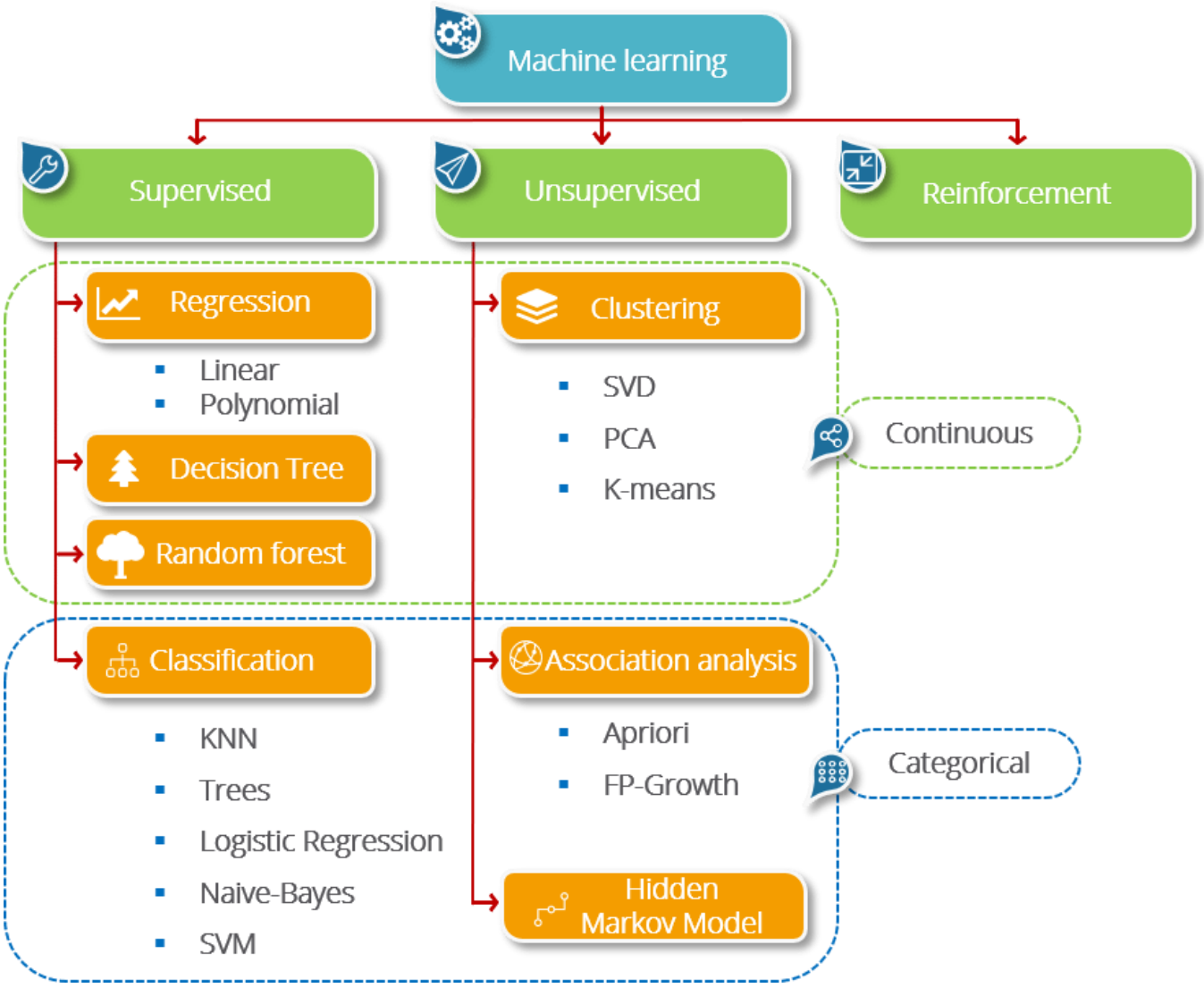
Machine Learning:-

Machine learning (**ML**) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning.In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

Machine Learning is broadly categorized under the following headings:



Supervised Learning:-

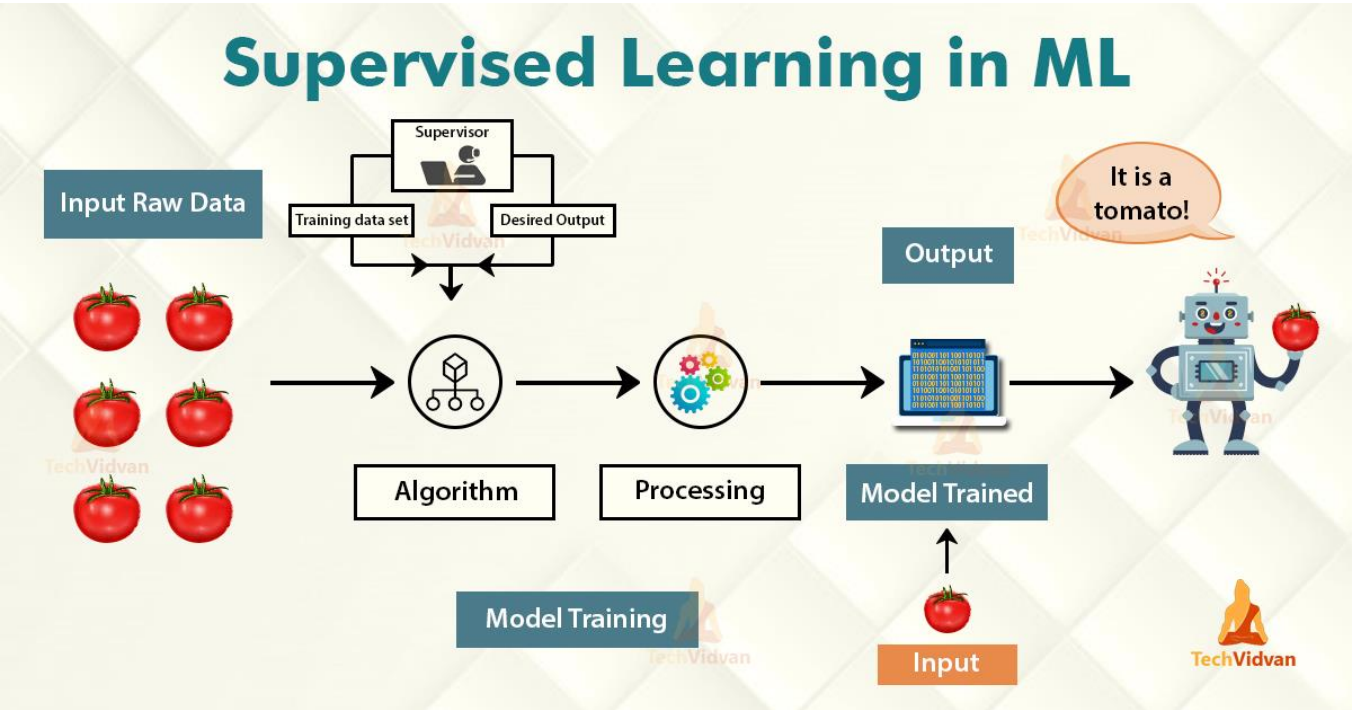
• Supervised Learning is the one, where you can consider the learning is guided by a teacher.

• We have a dataset which acts as a teacher and its role is to train the model or the machine.

• Once the model gets trained it can start making a prediction or decision when new data is given to it

• Supervised learning is the one where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output. Y = f(X)

• The goal is to approximate the mapping function so well that whenever you get some new input data (x), the machine can easily predict the output variables (Y) for that data.



Unsupervised Learning:-

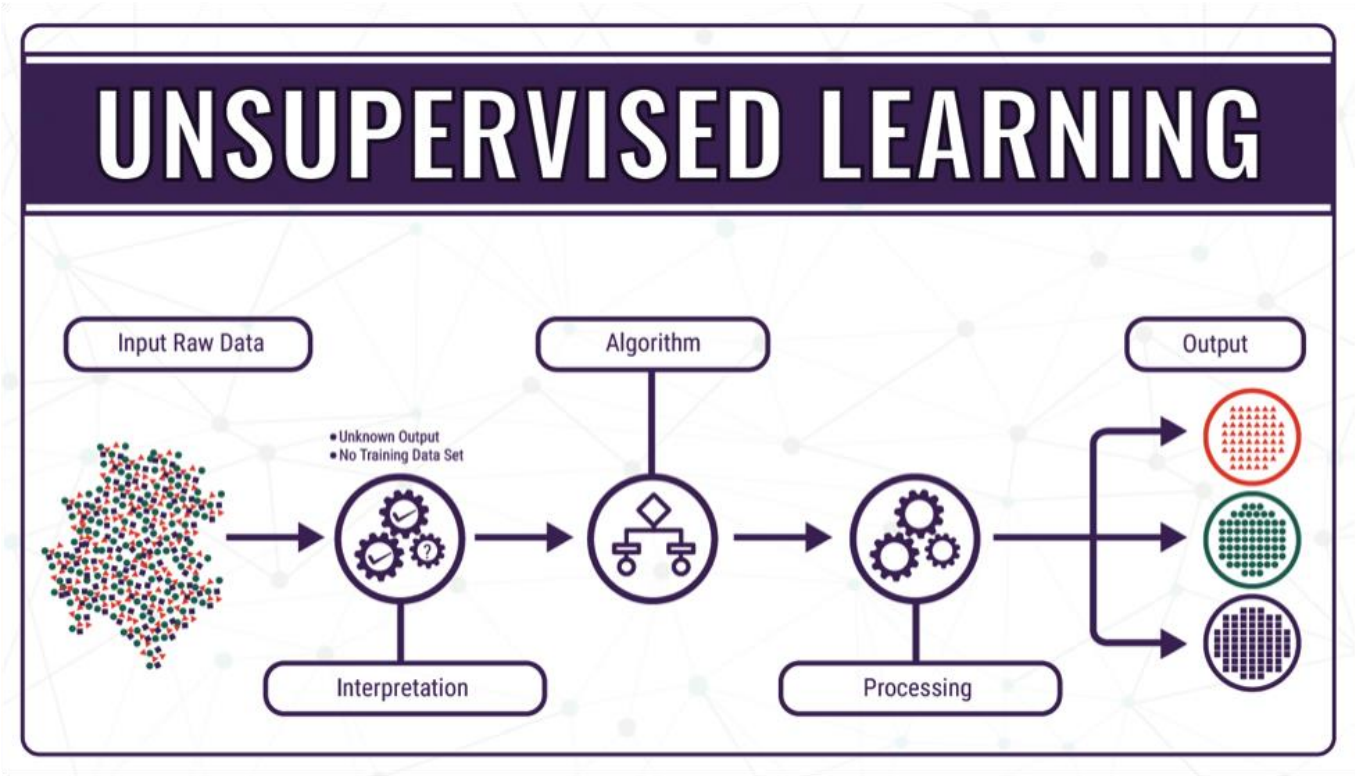
• The model learns through observation and finds structures in the data.

• Once the model is given a dataset, it automatically finds patterns and relationships in the dataset by creating clusters in it.

• What it cannot do is add labels to the cluster, like it cannot say this a group of apples or mangoes, but it will separate all the apples from mangoes

• Mathematically, Unsupervised learning is where you only have input data (X) and no corresponding output variables.

• The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data



Reinforcement Learning:-

• It is the ability of an agent to interact with the environment and find out what is the best outcome.

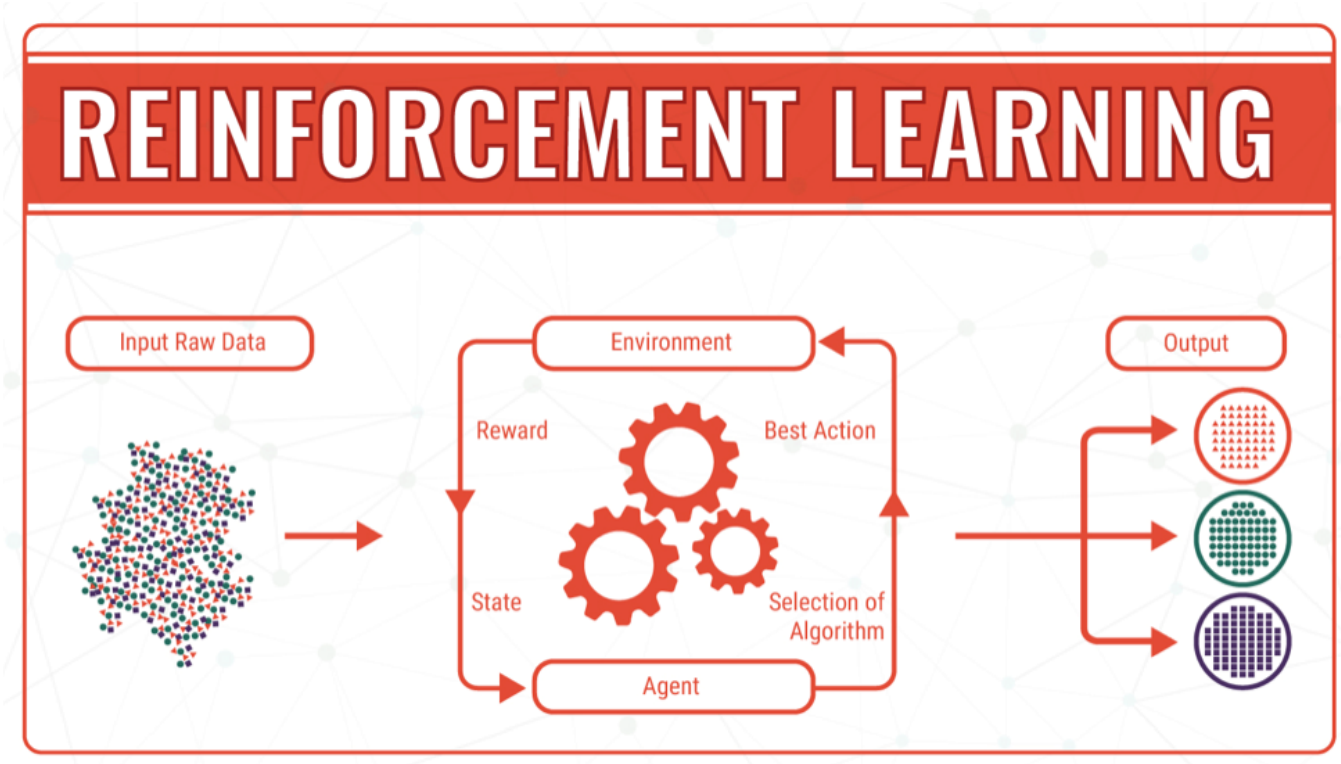
• It follows the concept of hit and trial method.

• The agent is rewarded or penalized with a point for a correct or a wrong answer, and on the basis of the positive reward points gained the model trains itself.

• And again once trained it gets ready to predict the new data presented to it.

• Reinforcement learning can be thought of as a hit and trial method of learning. The machine gets a Reward or Penalty point for each action it performs.

• If the option is correct, the machine gains the reward point or gets a penalty point in case of a wrong response.



Machine Learning Process:-

Step 1: **Define the objective** of the Problem Statement At this step, we must understand what exactly needs to be predicted.

Step 2: **Data Gathering** At this stage, you must be asking questions such as, What kind of data is needed to solve this problem?

Is the data available?

How can I get the data?

Step 3: Data Preparation

– The data you collected is almost never in the right format.

– You will encounter a lot of inconsistencies in the data set such as missing values, redundant variables, duplicate values, etc. Removing such inconsistencies is very essential because they might lead to wrongful computations and predictions.

– Therefore, at this stage, you scan the data set for any inconsistencies and you fix them then and there.

• Step 4: Exploratory Data Analysis

– Grab your detective glasses because this stage is all about diving deep into data and finding all the hidden data mysteries.

– EDA or Exploratory Data Analysis is the brainstorming stage of Machine Learning.

– Data Exploration involves understanding the patterns and trends in the data.

– At this stage, all the useful insights are drawn and correlations between the variables are understood.

• Step 5: Building a Machine Learning Model

– All the insights and patterns derived during Data Exploration are used to build the Machine Learning Model.

– This stage always begins by splitting the data set into two parts, training data, and testing data.

– The training data will be used to build and analyze the model.

– The logic of the model is based on the Machine Learning Algorithm that is being implemented.

• Step 6: Model Evaluation & Optimization

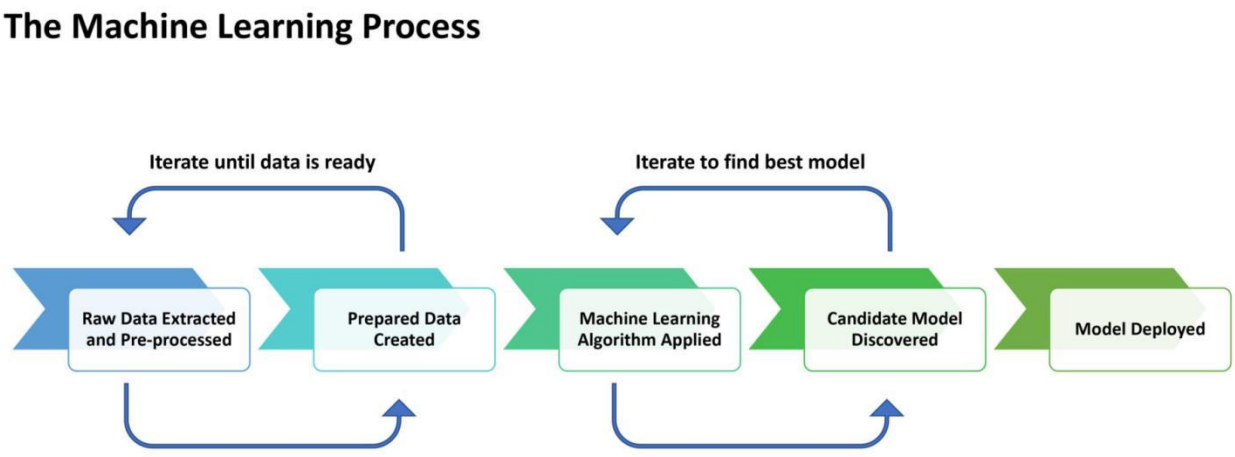
– After building a model by using the training data set, it is finally time to put the model to a test.

– The testing data set is used to check the efficiency of the model and how accurately it can predict the outcome.

– Once the accuracy is calculated, any further improvements in the model can be implemented at this stage.
– Methods like parameter tuning and cross-validation can be used to improve the performance of the model.

• Step 7: Predictions

– Once the model is evaluated and improved, it is finally used to make predictions. The final output can be a Categorical variable (eg. True or False) or it can be a Continuous Quantity (eg. the predicted value of a stock).



Linear Regression:-

• “Regression analysis is a form of predictive modelling technique which investigates relationship between a dependent and an independent variable”

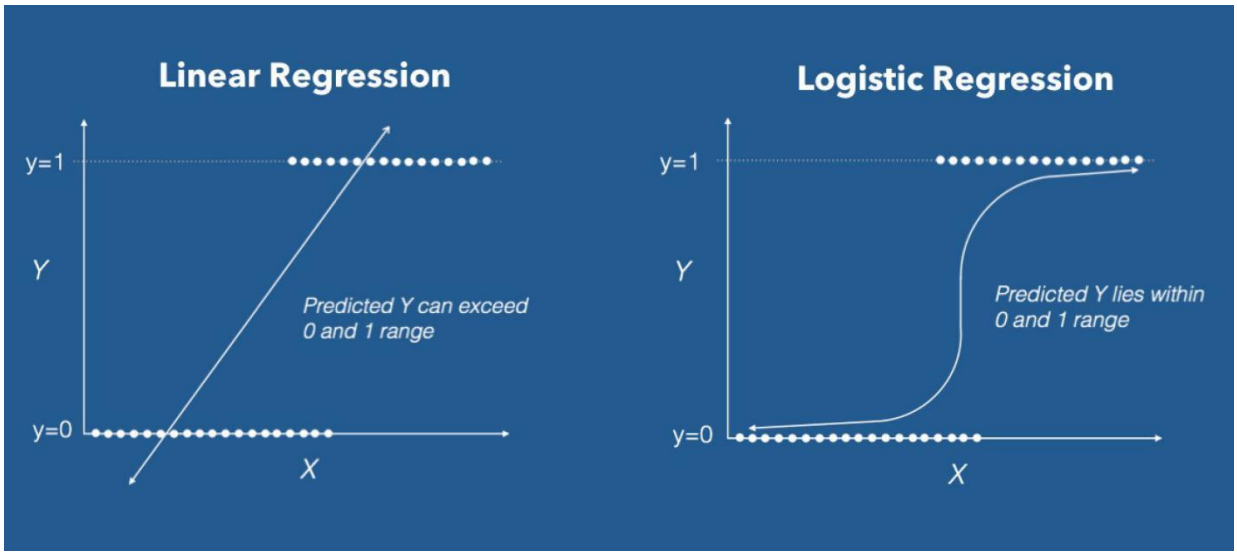
Uses:

• There are three main uses of regression analysis:

– Determining the strength of predictors.

– Forecasting an effect, and

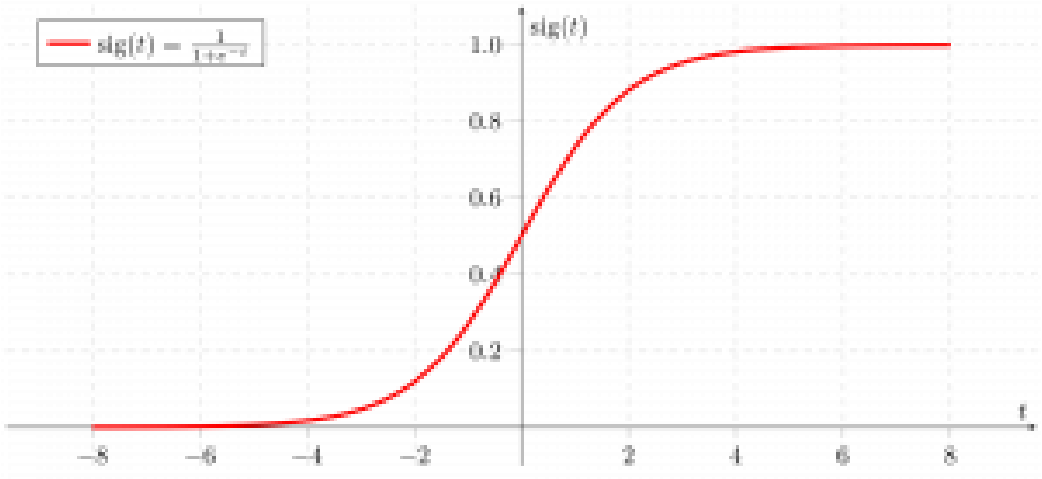
– Trend Forecastin



Logistic Regression:-

The basics of Logistic Regression and its implementation in Python. Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for given set of features(or inputs), X.

Contrary to popular belief, logistic regression IS a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.



The decision for the value of the threshold value is majorly affected by the values of precision and recall. Ideally, we want both precision and recall to be 1, but this seldom is the case. In case of a Precision-Recall tradeoff we use the following arguments to decide upon the thresold:-

1. **Low Precision/High Recall**:

In applications where we want to reduce the number of false negatives without necessarily reducing the number false positives, we choose a decision value which has a low value of Precision or high value of Recall. For example, in a cancer diagnosis application, we do not want any affected patient to be classified as not affected without giving much heed to if the patient is being wrongfully diagnosed with cancer. This is because, the absence of cancer can be detected by further medical diseases but the presence of the disease cannot be detected in an already rejected candidate.

2. **High Precision/Low Recall:**

In applications where we want to reduce the number of false positives without necessarily reducing the number false negatives, we choose a decision value which has a high value of Precision or low value of Recall. For example, if we are classifying customers whether they will react positively or negatively to a personalised advertisement, we want to be absolutely sure that the customer will react positively to the advertisemnt because otherwise, a negative reaction can cause a loss potential sales from the customer.

Based on the number of categories, Logistic regression can be classified as:

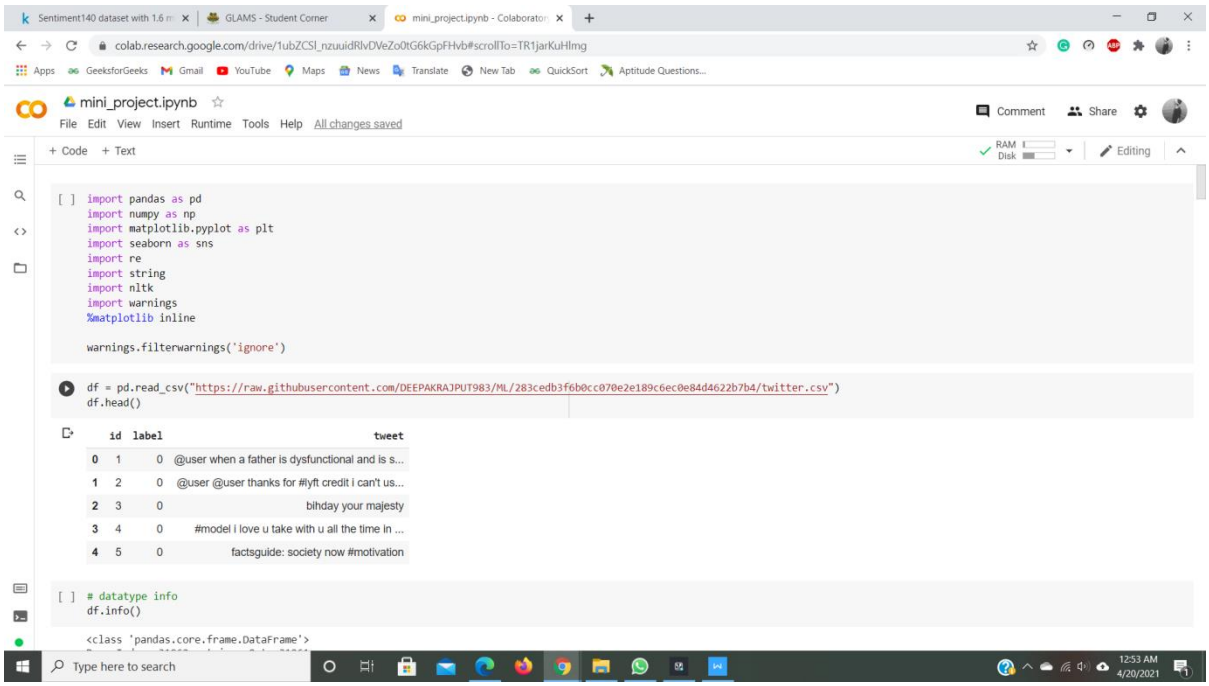
binomial: target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc.

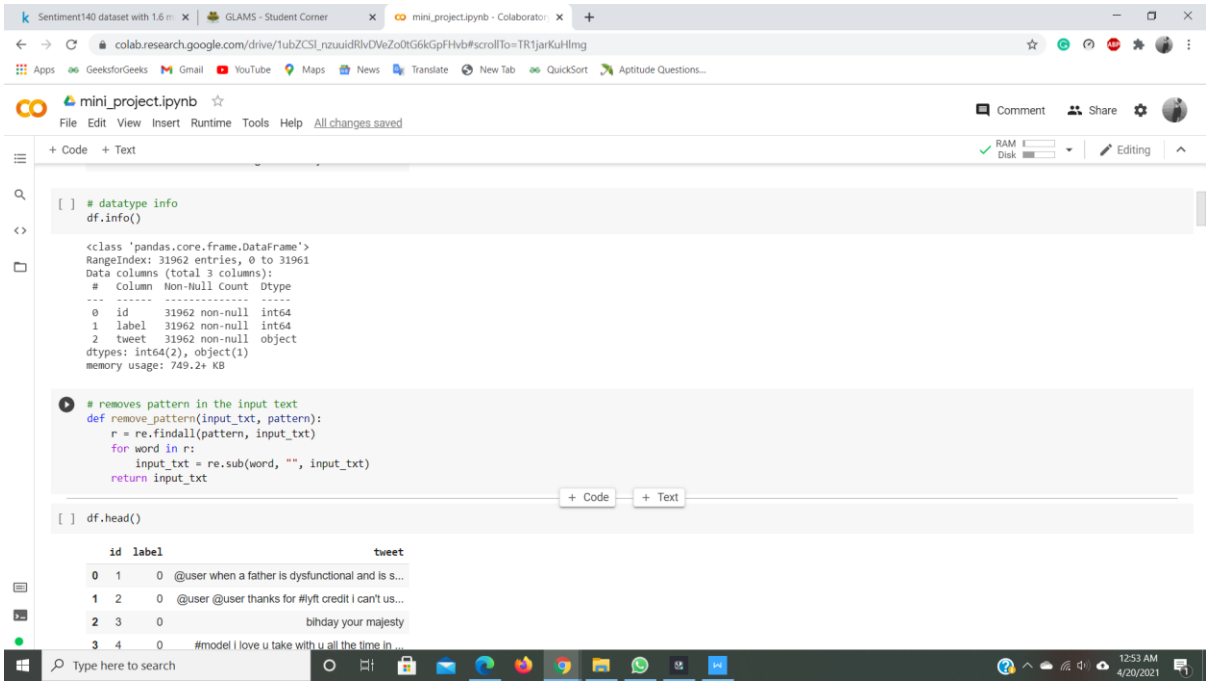
multinomial: target variable can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”.

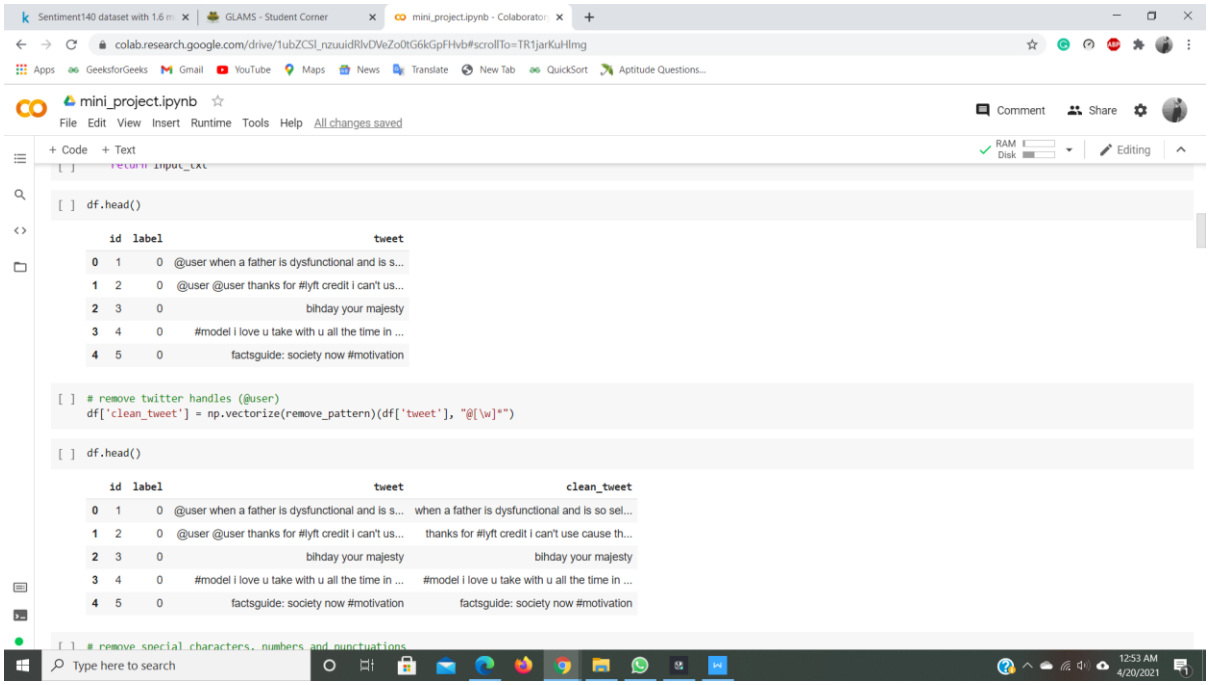
ordinal: it deals with target variables with ordered categories. For example, a test score can be categorized as:“very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3.

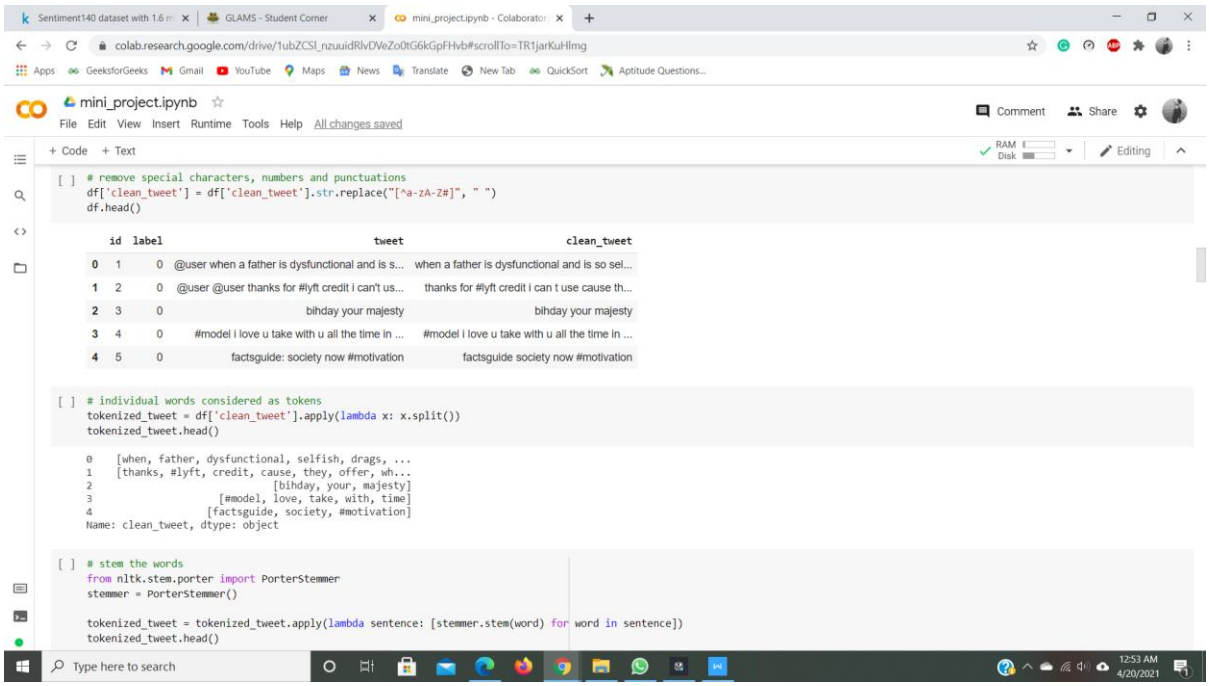
CHAPTER -3:

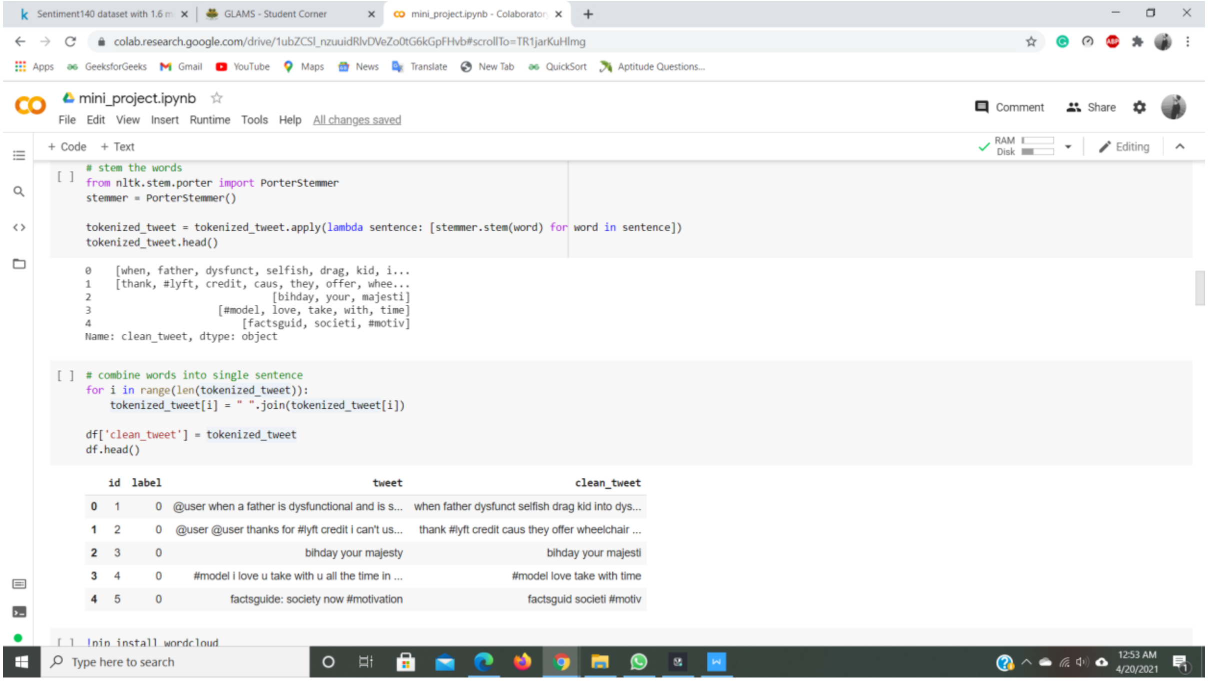
Screenshots

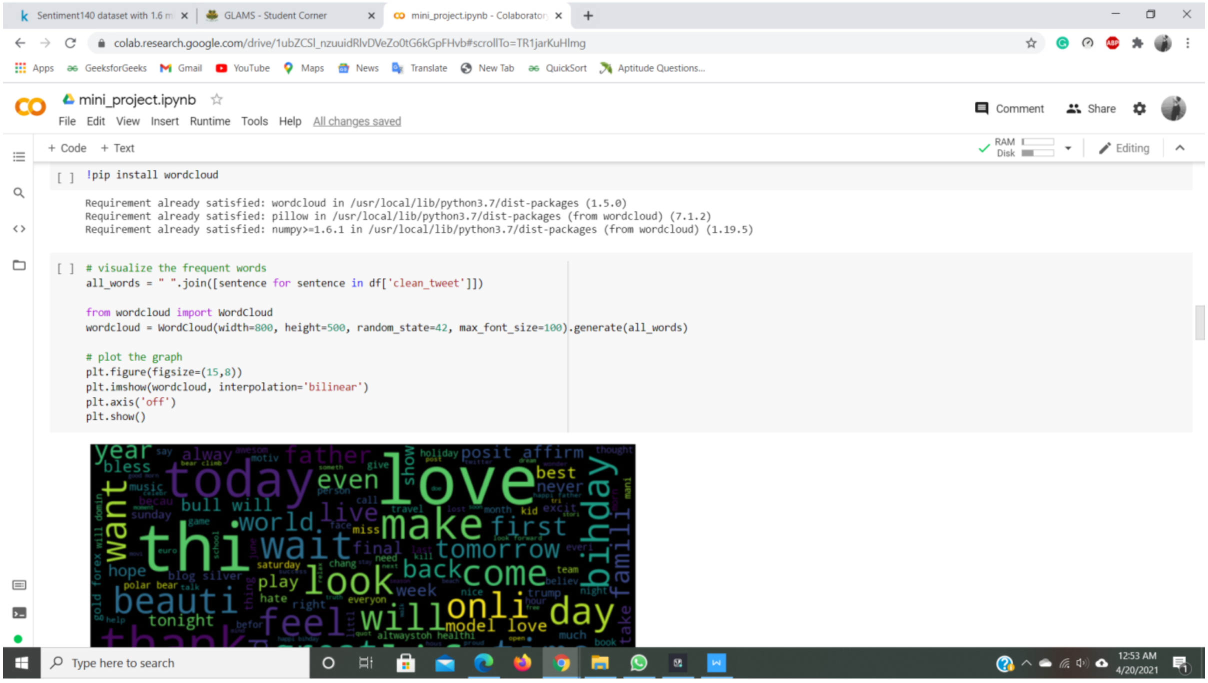


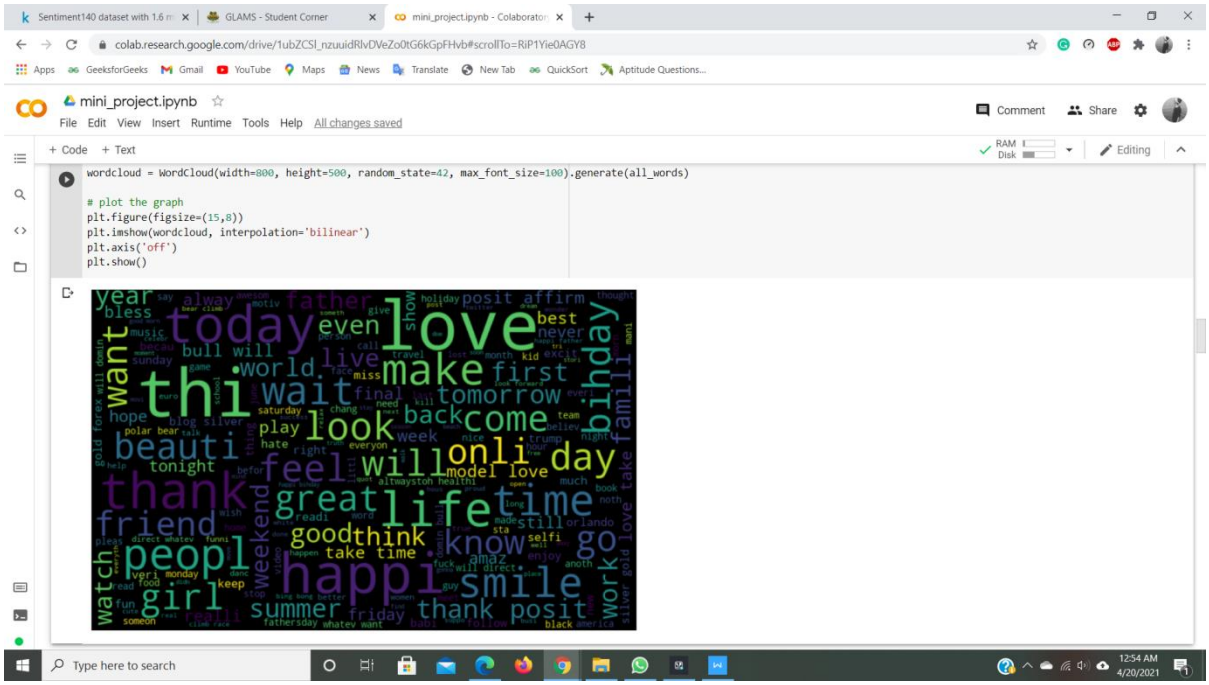


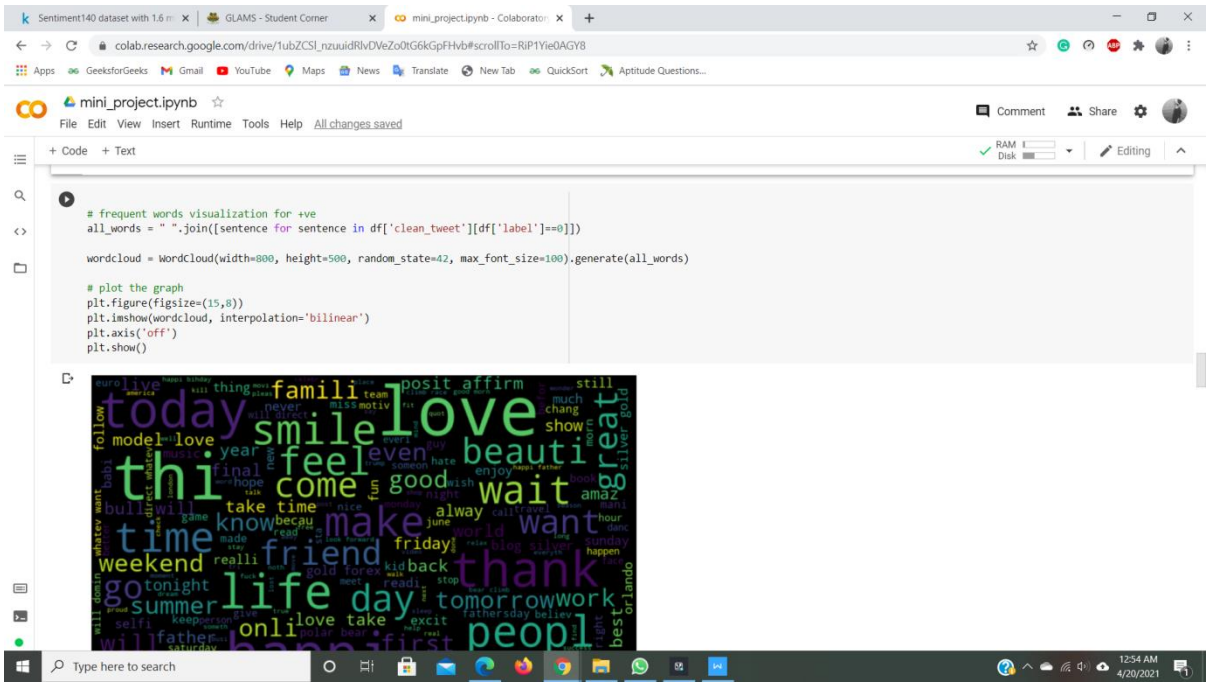

mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
[ ] df.head()
id label tweet
0 1 0 @user when a father is dysfunctional and is s...
1 2 0 @user @user thanks for #lyft credit i can't us...
2 3 0 bihday your majesty
3 4 0 #model i love u take with u all the time in ...
4 5 0 factsguide: society now #motivation
[ ] # remove twitter handles (@user)
df['clean_tweet'] = np.vectorize(remove_pattern)(df['tweet'], "@[\w]*")
[ ] df.head()
id label tweet clean_tweet
0 1 0 @user when a father is dysfunctional and is s... when a father is dysfunctional and is so sel...
1 2 0 @user @user thanks for #lyft credit i can't us... thanks for #lyft credit i can't use cause th...
2 3 0 bihday your majesty bihday your majesty
3 4 0 #model i love u take with u all the time in ... #model i love u take with u all the time in ...
4 5 0 factsguide: society now #motivation factsguide: society now #motivation
Type here to search
12:53 AM
4/20/2021

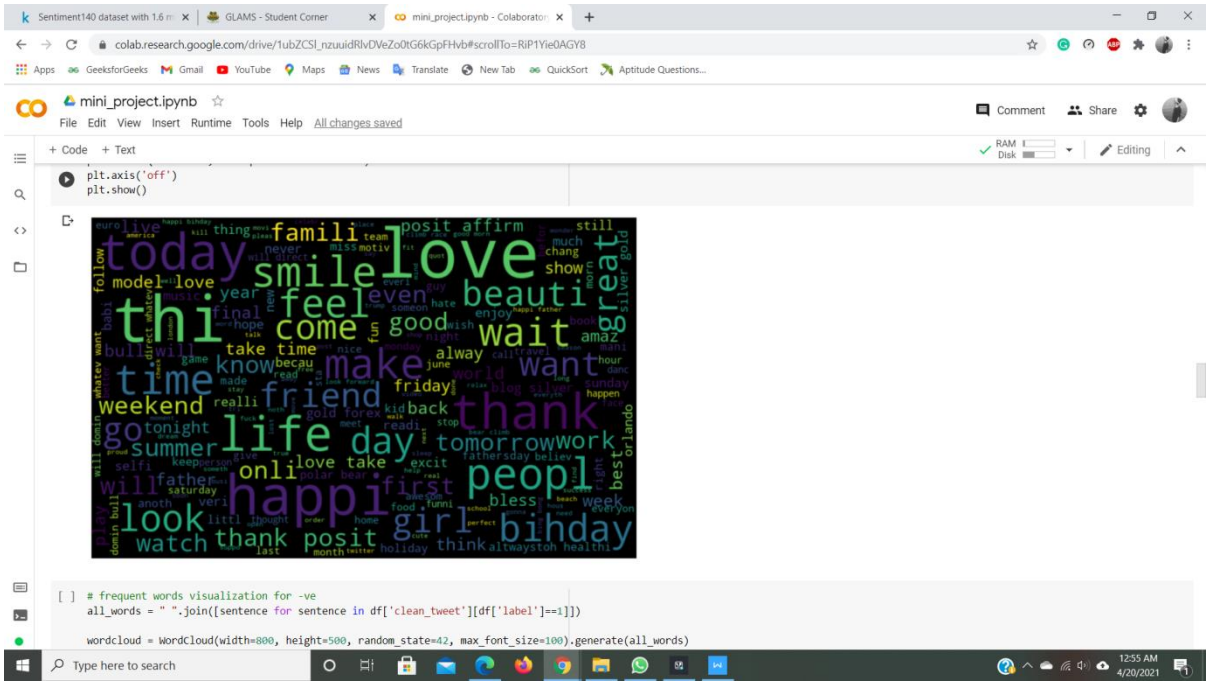

mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
[ ] # remove special characters, numbers and punctuations
df['clean_tweet'] = df['clean_tweet'].str.replace("[^a-zA-Z#]", " ")
df.head()
id label tweet clean_tweet
0 1 0 @user when a father is dysfunctional and is s... when a father is dysfunctional and is so sel...
1 2 0 @user @user thanks for #lyft credit i can't us... thanks for #lyft credit i can t use cause th...
2 3 0 bihday your majesty bihday your majesty
3 4 0 #model i love u take with u all the time in ... #model i love u take with u all the time in ...
4 5 0 factsguide: society now #motivation factsguide society now #motivation
[ ] # individual words considered as tokens
tokenized_tweet = df['clean_tweet'].apply(lambda x: x.split())
tokenized_tweet.head()
0 [when, father, dysfunctional, selfish, drags, ...
1 [thanks, #lyft, credit, cause, they, offer, wh...
2 [bihday, your, majesty]
3 [#model, love, take, with, time]
4 [factsguide, society, #motivation]
Name: clean_tweet, dtype: object
[ ] # stem the words
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()
tokenized_tweet = tokenized_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])
tokenized_tweet.head()
Type here to search
12:53 AM
4/20/2021

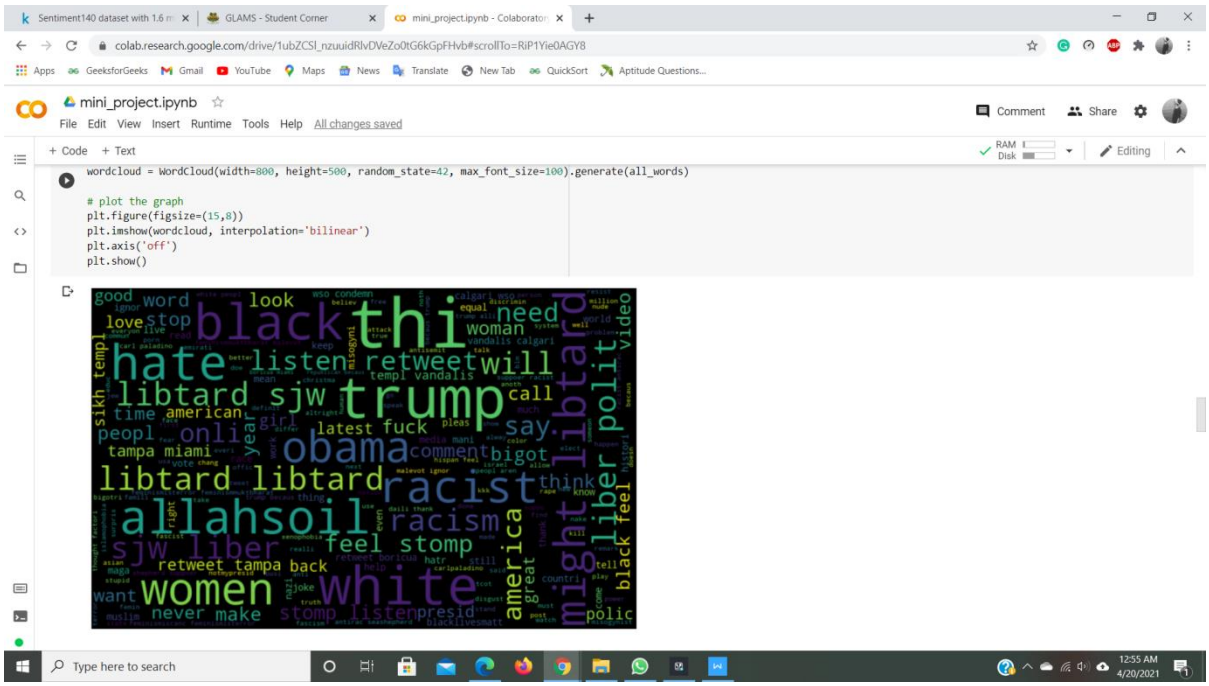

mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
# stem the words
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()
tokenized_tweet = tokenized_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])
tokenized_tweet.head()
0 [when, father, dysfunct, selfish, drag, kid, i...
1 [thank, #lyft, credit, caus, they, offer, whee...
2 [bihday, your, majesti]
3 [#model, love, take, with, time]
4 [factsguid, societi, #motiv]
Name: clean_tweet, dtype: object
# combine words into single sentence
for i in range(len(tokenized_tweet)):
tokenized_tweet[i] = " ".join(tokenized_tweet[i])
df['clean_tweet'] = tokenized_tweet
df.head()
id label tweet clean_tweet
0 1 0 @user when a father is dysfunctional and is s... when father dysfunct selfish drag kid into dys...
1 2 0 @user @user thanks for #lyft credit i can't us... thank #lyft credit caus they offer wheelchair ...
2 3 0 bihday your majesty bihday your majesti
3 4 0 #model i love u take with u all the time in ... #model love take with time
4 5 0 factsguide: society now #motivation factsguid societi #motiv

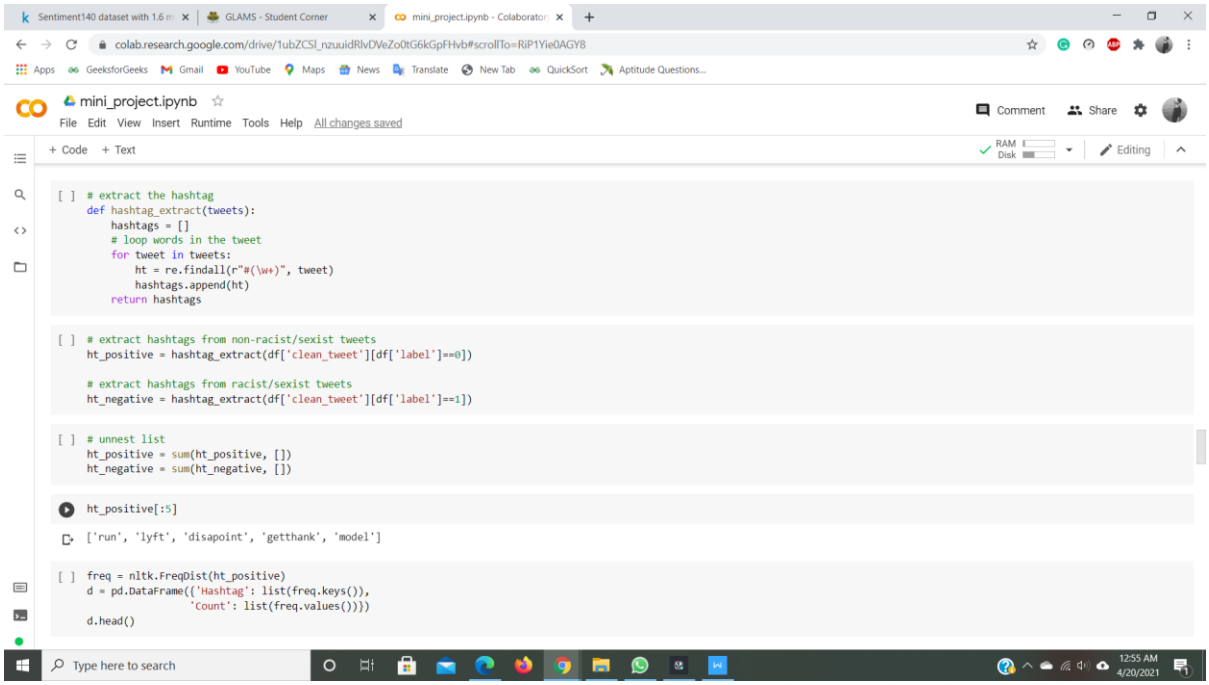

mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
!pip install wordcloud
Requirement already satisfied: wordcloud in /usr/local/lib/python3.7/dist-packages (1.5.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from wordcloud) (7.1.2)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.7/dist-packages (from wordcloud) (1.19.5)
# visualize the frequent words
all_words = " ".join([sentence for sentence in df['clean_tweet']])
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(all_words)
# plot the graph
plt.figure(figsize=(15,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(all_words)
# plot the graph
plt.figure(figsize=(15,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
Type here to search
12:54 AM
4/20/2021


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
# frequent words visualization for +ve
all_words = " ".join([sentence for sentence in df['clean_tweet'][df['label']==0]])
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(all_words)
# plot the graph
plt.figure(figsize=(15,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
Type here to search
12:54 AM
4/20/2021


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk Editing
plt.axis('off')
plt.show()
today
love
smile
thi
come
feel
beauti
wait
great
time
make
want
friend
thank
weekend
life
day
tomorrow
work
go
summer
peopl
happi
look
girl
bihday
watch
[ ] # frequent words visualization for -ve
all_words = " ".join([sentence for sentence in df['clean_tweet'][df['label']==1]])
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(all_words)
Type here to search
12:55 AM
4/20/2021


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk Editing
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(all_words)
# plot the graph
plt.figure(figsize=(15,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
black
thi
hate
trump
libtard
sjw
obama
racist
allahsoil
women
white
Type here to search
12:55 AM
4/20/2021



```
# extract the hashtag
def hashtag_extract(tweets):
    hashtags = []
    # loop words in the tweet
    for tweet in tweets:
        ht = re.findall(r"#(\w+)", tweet)
        hashtags.append(ht)
    return hashtags
```

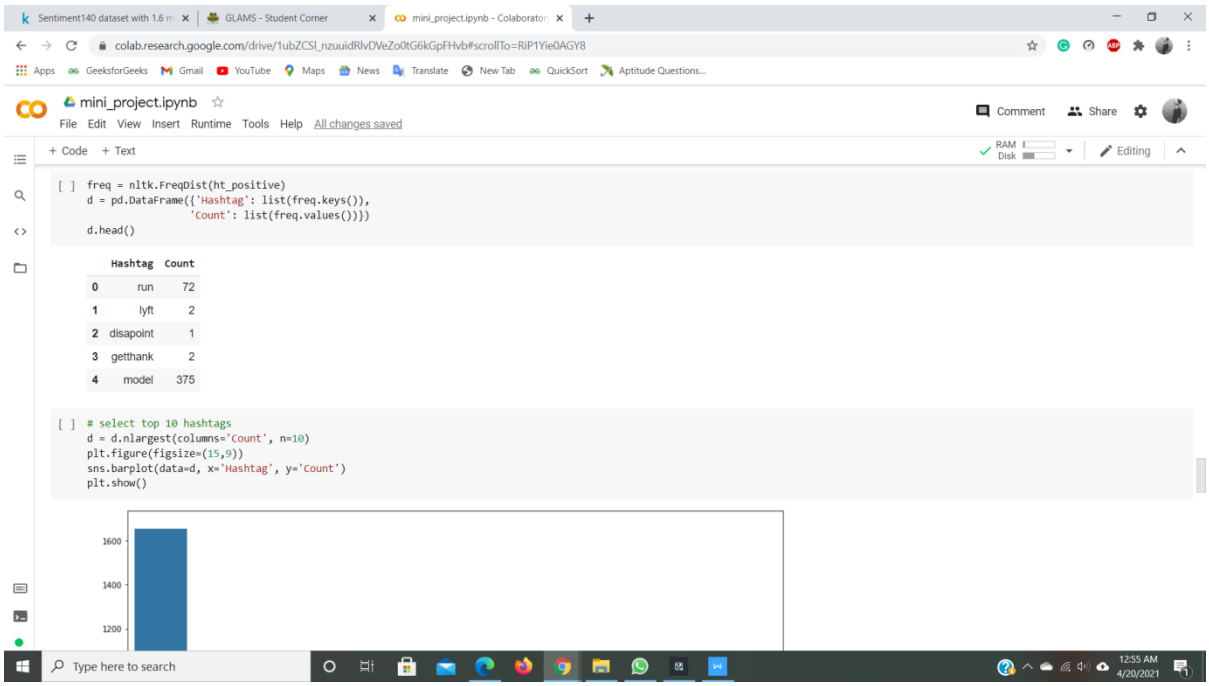
```
# extract hashtags from non-racist/sexist tweets
ht_positive = hashtag_extract(df['clean_tweet'][df['label']==0])

# extract hashtags from racist/sexist tweets
ht_negative = hashtag_extract(df['clean_tweet'][df['label']==1])
```

```
# unnest list
ht_positive = sum(ht_positive, [])
ht_negative = sum(ht_negative, [])
```

```
ht_positive[:5]
```

```
['run', 'lyft', 'disapoint', 'getthank', 'model']
```

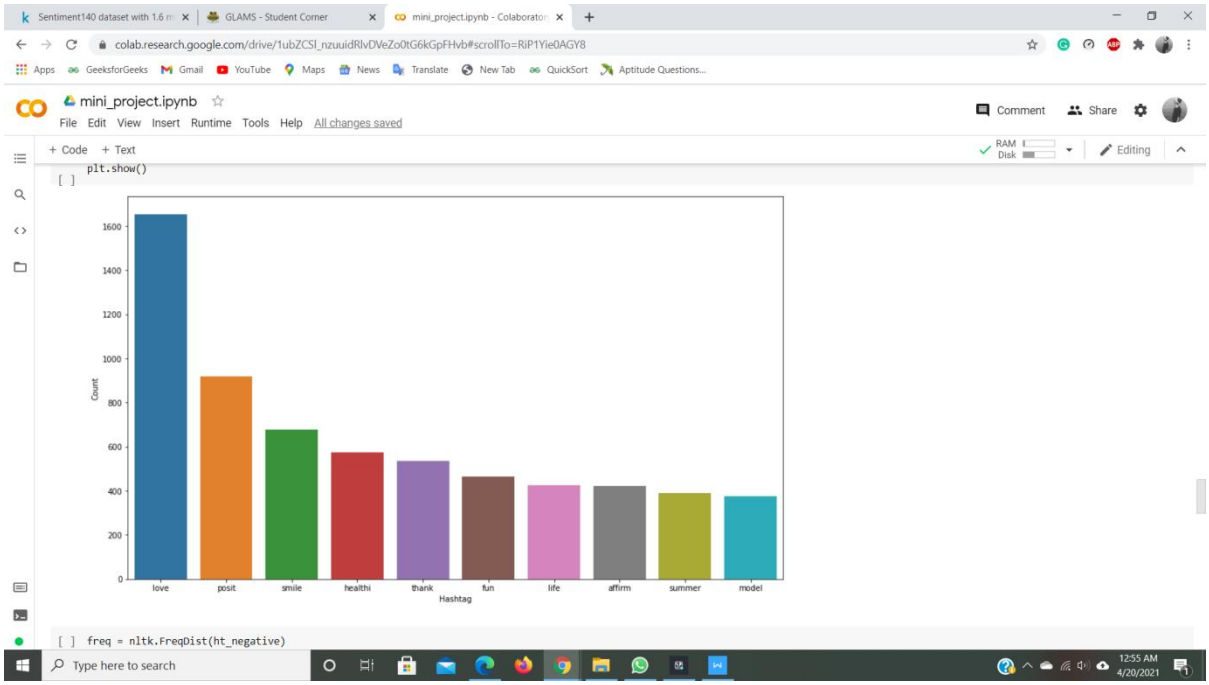
```
freq = nltk.FreqDist(ht_positive)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
                  'Count': list(freq.values())})
d.head()
```

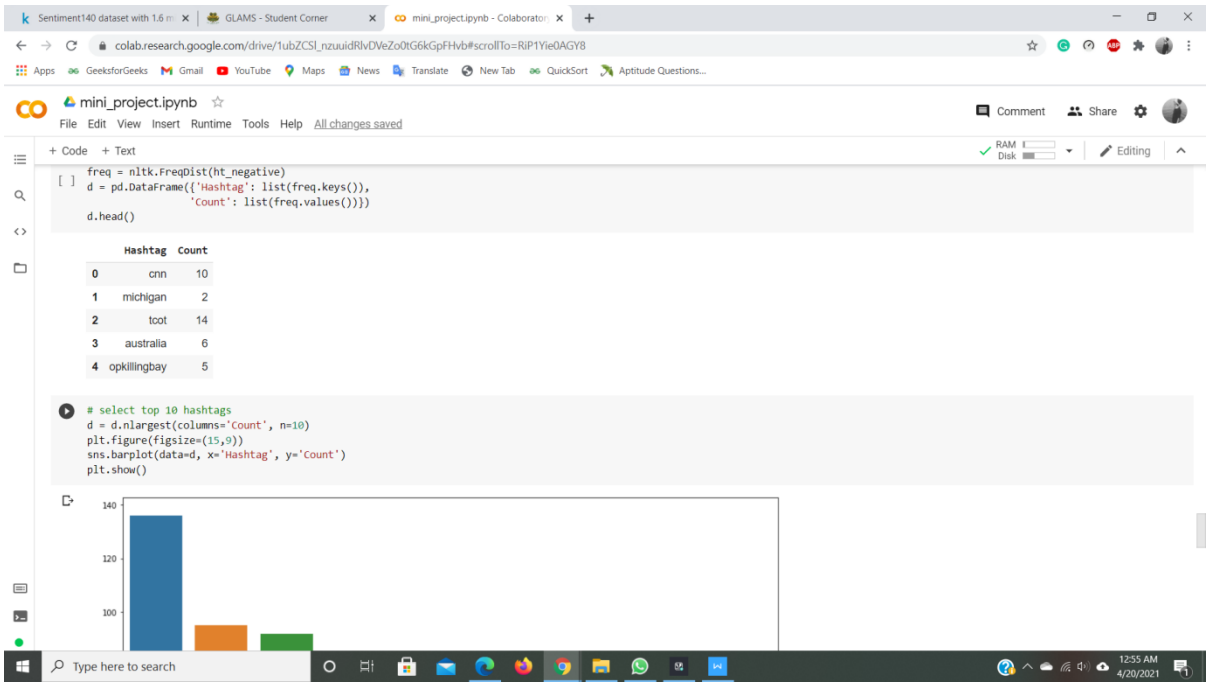


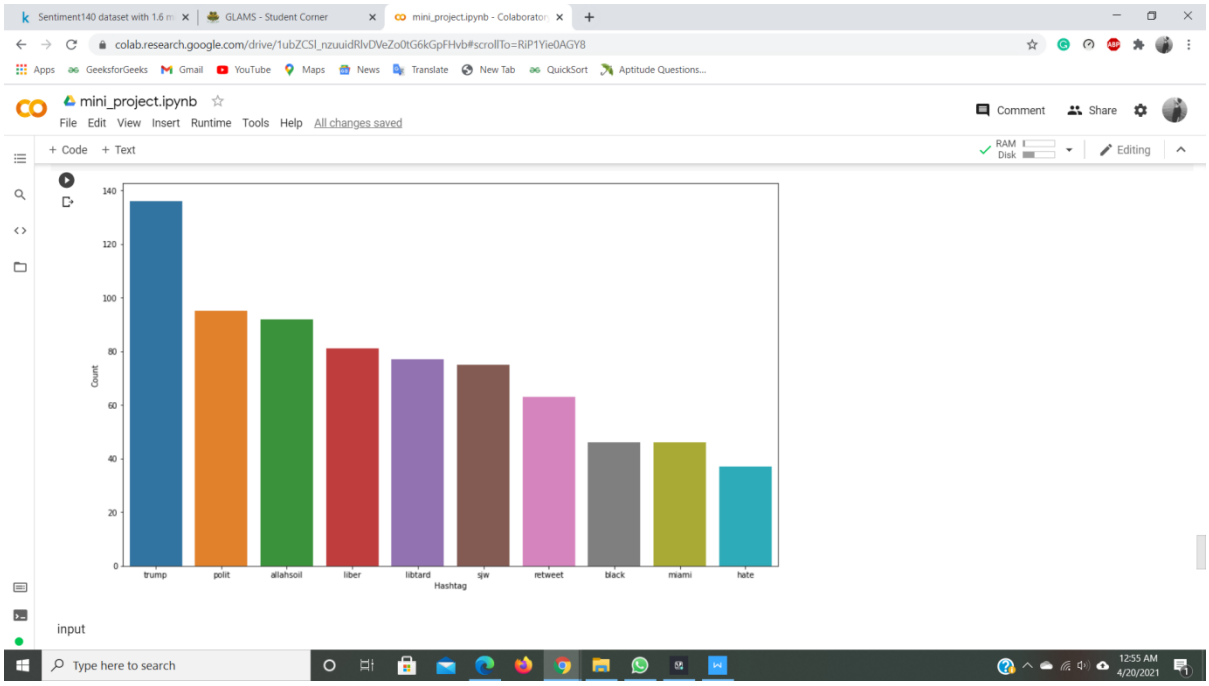
```
freq = nltk.FreqDist(ht_positive)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
                  'Count': list(freq.values())})
d.head()
```

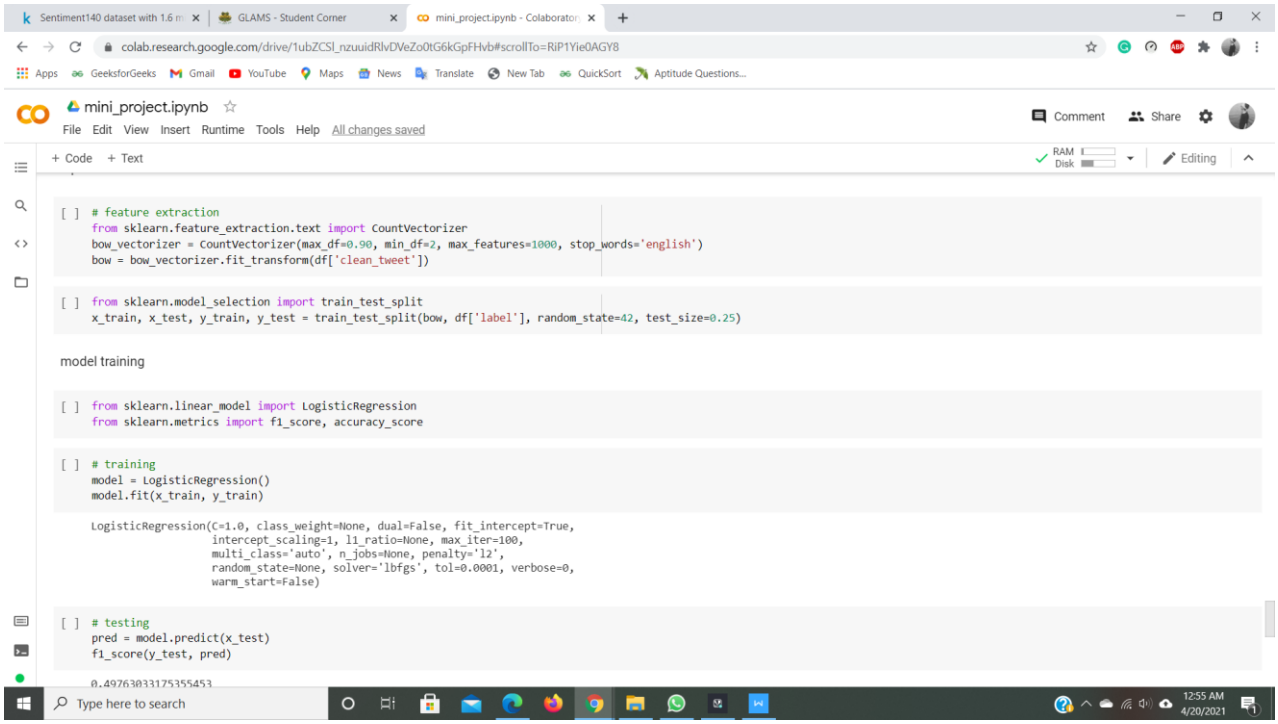
	Hashtag	Count
0	run	72
1	lyft	2
2	disapoint	1
3	getthank	2
4	model	375

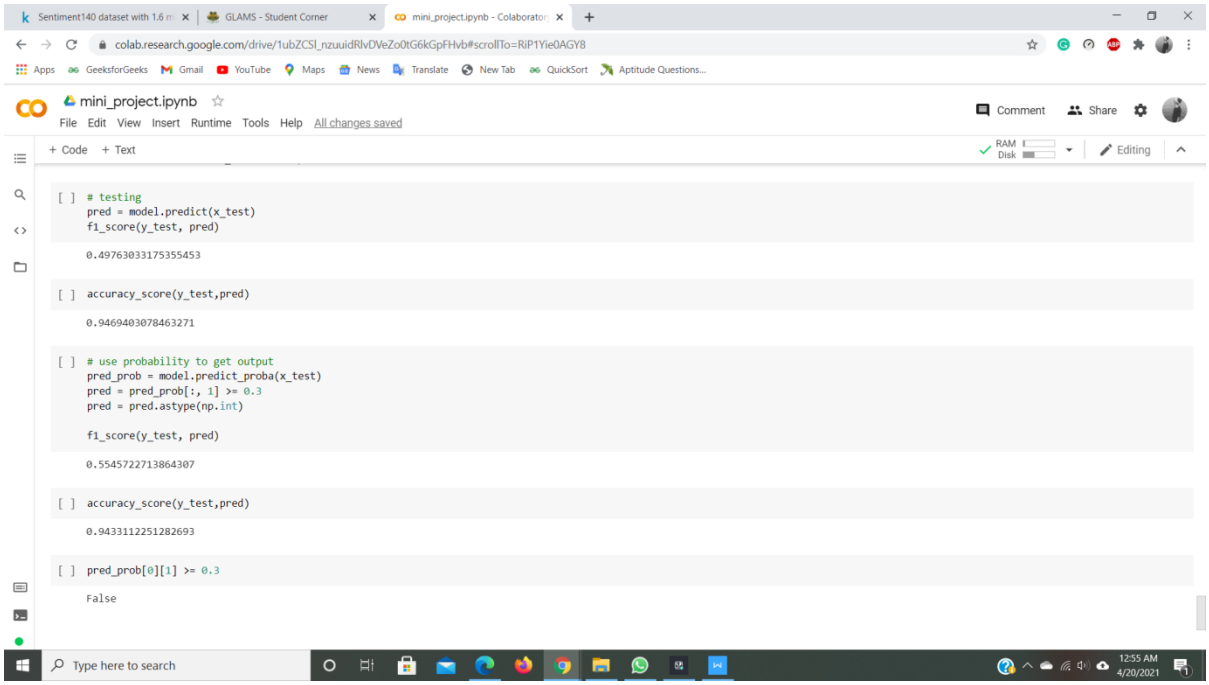
```
# select top 10 hashtags
d = d.nlargest(columns='Count', n=10)
plt.figure(figsize=(15,9))
sns.barplot(data=d, x='Hashtag', y='Count')
plt.show()
```


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
plt.show()
Count
Hashtag
love
posit
smile
healthi
thank
fun
life
affirm
summer
model
freq = nltk.FreqDist(ht_negative)
Type here to search
12:55 AM
4/20/2021


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
freq = nltk.FreqDist(ht_negative)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
'Count': list(freq.values())})
d.head()
Hashtag Count
0 cnn 10
1 michigan 2
2 tcot 14
3 australia 6
4 opkillingbay 5
# select top 10 hashtags
d = d.nlargest(columns='Count', n=10)
plt.figure(figsize=(15,9))
sns.barplot(data=d, x='Hashtag', y='Count')
plt.show()
Type here to search
12:55 AM
4/20/2021


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Count
Hashtag
trump
polit
allahsoil
liber
libtard
sjw
retweet
black
miami
hate
input
Type here to search
12:55 AM
4/20/2021


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
# feature extraction
from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer(max_df=0.90, min_df=2, max_features=1000, stop_words='english')
bow = bow_vectorizer.fit_transform(df['clean_tweet'])
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(bow, df['label'], random_state=42, test_size=0.25)
model training
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, accuracy_score
# training
model = LogisticRegression()
model.fit(x_train, y_train)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)
# testing
pred = model.predict(x_test)
f1_score(y_test, pred)
0.49763033175355453
Type here to search
12:55 AM
4/20/2021


mini_project.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[ ] # testing
pred = model.predict(x_test)
f1_score(y_test, pred)
0.49763033175355453
[ ] accuracy_score(y_test,pred)
0.9469403078463271
[ ] # use probability to get output
pred_prob = model.predict_proba(x_test)
pred = pred_prob[:, 1] >= 0.3
pred = pred.astype(np.int)
f1_score(y_test, pred)
0.5545722713864307
[ ] accuracy_score(y_test,pred)
0.9433112251282693
[ ] pred_prob[0][1] >= 0.3
False
Type here to search
12:55 AM
4/20/2021

FUTURE SCOPE

1) Data Pre - Processing using more parameter to get best sentimental

2) Updating Dictionary for new synonym and antonyms of already existing words.

3) Context Sentimental Analysis may be implemented in future for accuracy purposes

REFERENCES

1. **https://www.kaggle.com/kazanova/sentiment140**
2. **https://www.geeksforgeeks.org/understanding-logistic-regression/**
3. **https://www.geeksforgeeks.org/machine-learning/**
4. **https://www.w3schools.com/python/**