**MILESTONE 1 :UNDERSTANDING THE PROBLEM STATEMENT**

We as a team referred to certain articles and youtube videosto get some information about our problem statement and acquire an idea to proceed with the statement. We found the articles and the youtube resources useful and here is what we inferred from the them.

**SENTIMENT ANALYSIS:**

Sentiment analysis is basically analyzing people’s emotions ( positive or maybe negative) towards a matter, be it a movie or an online product by analysing their feedback or reviews. It mainly uses NLP (Natural language processing) . It is text analysis method based on texts/documents/paragraph and helps the product developer to develop the product meeting the customer needs. It is based on polarity or feelings

**TRADITIONAL STEPS INVOLVED ARE** :

* Break a sentence into various components.
* Neglecting the punctuationa and other stop words.
* Identify the word and assign a sentiment score.
* Calculate the overall score to determine the sentiment of the statement.

**BENEFITS OF SENTIMENT ANALYSIS**:

* It is efficient and cost effective.
* Identify situation and takes action.
* Improve aacuracy and gain better insights.

**ALGORITHMS**:

**LOW-LEVEL**: it includes simple techniques such grouping similar words, removing unnecessary parts and assigning scores. It includes techniques such as Tokenization, Lmammetization, Data Cleaning.

**AUTOMATIC**: Low -level analytics may become uneffective at times and that is where machine learning comes into picture..Large amount of text can be used to train the model using supervised or unsupervised techniques. it contains training and prediction process which converts text inputs and fed to machine learning and generates predicted tags.

**HYBRID**: It is an efficient method which combines both low-level text analytics and machine learning algorithms to enhance the classification.

Some machine learning techniques involve Naïve bytes(baeyes’s theorem), Linear regression, Support vector machines and Deep learning.

**CHALLENGES**:

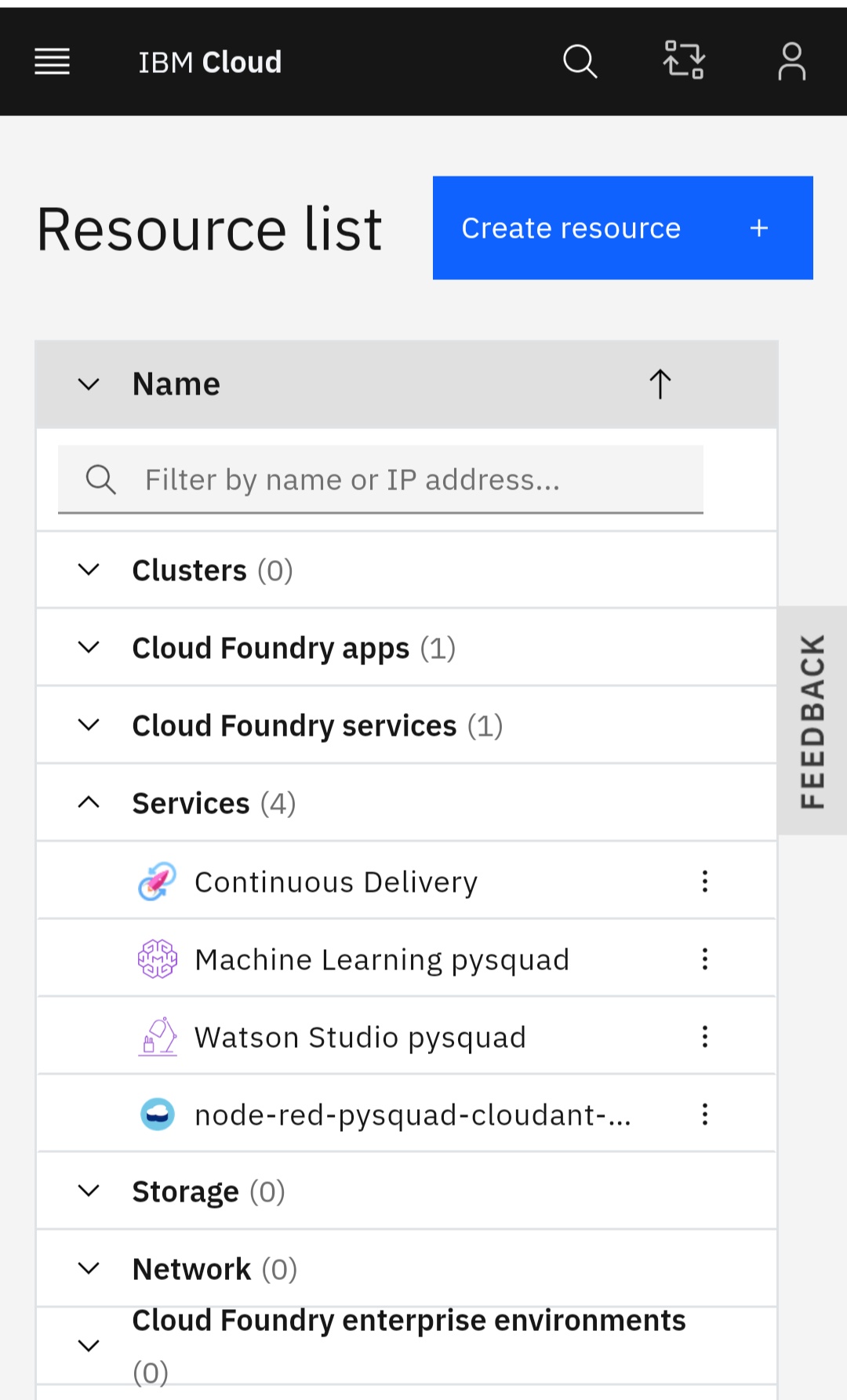
* Subjectivity and tone: based on subject and tone
* Context and polarity:deals with preprocessing and postprocessing
* Iron and sarcasm: it is not detected based on textal cue
* Comparision: done by comparing with other products
* Emojis : contains western and eastern emojis and need preprocessing to convert both to tokens

**REAL-TIME APPICATIONS**

* Social meadia monitoring: such as twitter,facebook etc. and the benoifits are prioritize actions,track trends over time
* Brand monitoring: benefits are as be tuned,brand reputation evolves
* Voice of customer(VOC): uses NPS(Net promoter Score) to gain feedback and benefits are as use results to design better, understand customer needs
* Customer service: benefits are responding to needs,increase efficiency
* Market research: get new sources of informations,informations based on real time.

**MILESTONE 2 :CREATING IBM CLOUD SERVICES:**

* A Cloud account has been created with my IBM ID: kavi17117.ei@rmkec.ac.in.
* The following cloud services are created in my account
* Node red
* Watson Studio
* machine learning



**MILESTONE 3 :BUILDING THE PYTHON CODE** :  
**Creating the Notebook:**

We created a new project named "Sentiment analysis of Covid-19 tweets". We created a notebook by clicking on the assets tab and selecting notebook.

**Importing the necessary packages:**

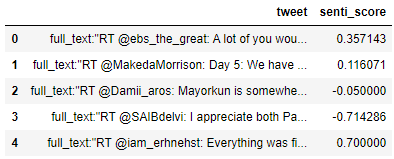
We imported all the necessary packages inclduing pandas, numpy, nltk and scikit-learn.

1. import pandas as pd
2. import numpy as np
3. import re
4. import string
5. import nltk
6. import matplotlib.pyplot as plt
7. from nltk.corpus import stopwords
8. from nltk.tokenize import word\_tokenize
9. from sklearn.feature\_extraction.text import TfidfVectorizer
10. from sklearn.model\_selection import train\_test\_split
11. from nltk.stem import PorterStemmer
12. from nltk.stem import WordNetLemmatizer
13. from sklearn.metrics import accuracy\_score
14. from sklearn.naive\_bayes import MultinomialNB
15. from sklearn.linear\_model import LogisticRegression
16. from sklearn.svm import SVC

**Importing the dataset:**

We gathered the dataset from :<https://ieee-dataport.org/open-access/coronavirus-covid-19-tweets-dataset>

Due to twitter's redistribution policy the dataset consists of only twitter IDs along with pre-determined sentiment scores using textblob. We hydrated (imported fresh tweets from twitter using the IDs) using the hydrator app.We neglected the other unwanted columns . We created a new data frame and saved the required columns.It looks like this.



**Pre-Processing the data:**

In order to remove the unnecessary data from the tweets, we make use of various nltk functions.

First step is to clean noise such as punctuation, special characters, numbers,twitter handles,http links etc.

Tokenisation : This method I used to split strings into smaller parts called tokens. The splitting is done based on whitespace and punctuation.

Removal of stopwords: removal of words that wont add weightage to the analysis, nltk has a predefined set of sop words.

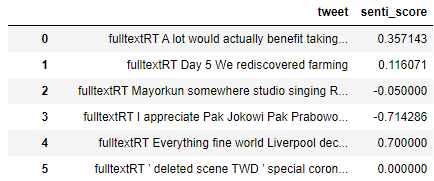
Normalization : This method group together the same meaning but different forms . Thus it includes 2 popular techniques called Lemmatization and Stemming

Lemmatization : This process is used to normalize words with morphological analysis of words in text.

Stemming : This process is to remove affixes from a word and it workswith verb forms.

1. nltk.download('stopwords')
2. nltk.download('punkt')
3. stop\_words = set(stopwords.words('english'))
4. def preprocess(tweet):
5. tweet.lower()
6. tweet = re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+#]|[!\*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+','', tweet)
7. tweet = re.sub("(@[A-Za-z0-9\_]+)","", tweet)
8. tweet = tweet.translate(str.maketrans('', '', string.punctuation)
9. tweet\_tokens = word\_tokenize(tweet)
10. filtered\_words = [w for w in tweet\_tokens if not w in stop\_words]
11. ps = PorterStemmer()
12. stemmed\_words = [ps.stem(w) for w in filtered\_words]
13. return " ".join(filtered\_words)
14. tweets\_new['tweet']=tweets\_new['tweet'].apply(preprocess)

Now our tweets looked like this.



We added a column Result to the dataset, in order to group the tweets into three categories based on the sentiment score.

1. def result(score):
2. if score < 0:
3. return 'negative'
4. elif score == 0:
5. return 'neutral'
6. else:
7. return 'positive'
8. tweets\_new['Result']=tweets\_new['senti\_score'].apply(result)
9. tweets\_new

**Training the model:**

Feature Extraction: This method is used to covert texts to numbers as Machine learning deals with numbers.

TF-IDF : Term frequency – Inverse document frequency which defines the importance of words.

Term Frequency : This method depend on words that appears in document frequently.

Inverse Document frequency : This method depends on words that appears less frequently in the document because they are more important and informative. This basically reduces the common words usage

**Spliting data into train and test**:

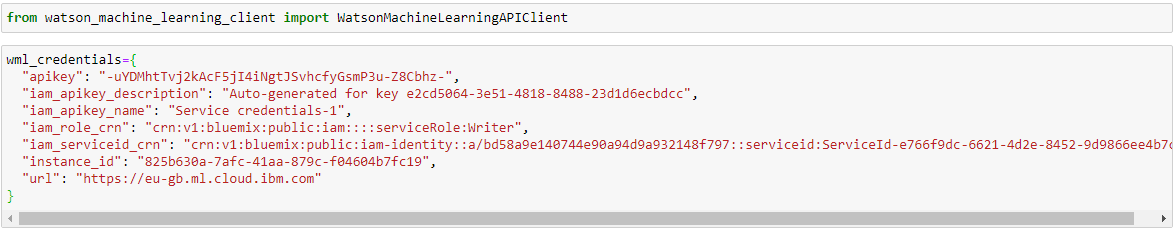
The data has been split into train and test data, we use naive bayes model in this project. **Naive Bayes Model :**

* It is a classifier to classify text which is applied based on Bayes theorem.
* Mainly uses Machine learning and data mining techniques to filter Irrelevant informations.
* this Model performs better than other models and need less training data.
* It can handle both continous and discrete datas.

1. vectorizer = TfidfVectorizer(sublinear\_tf=True)
2. x = vectorizer.fit\_transform(tweets\_new.tweet)
3. y= tweets\_new.Result
4. x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.20,random\_state=0)
5. NB\_model = MultinomialNB()
6. NB\_model.fit(x\_train, y\_train)
7. y\_predict= NB\_model.predict(x\_test)

**MILESTONE 4 :MACHINE LEARNING SERVICES:**

Machine learning credentials were created and given into the notebook.

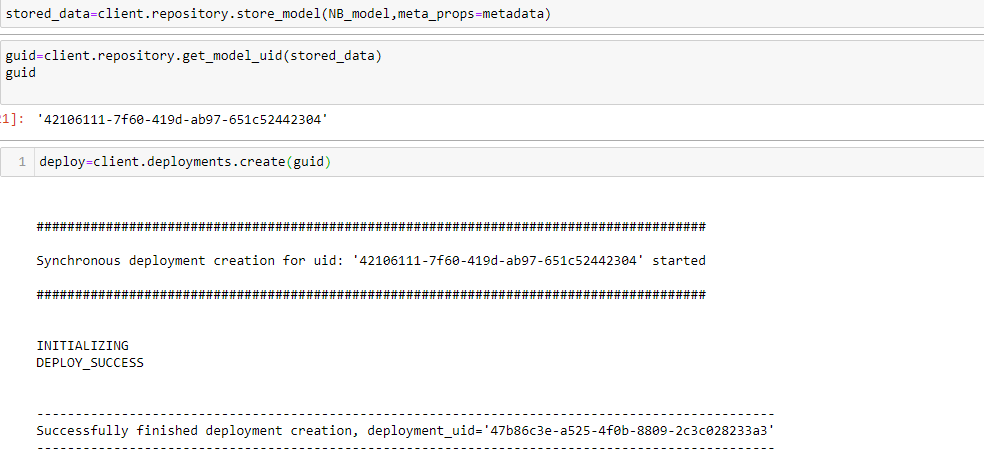


Client was created using the machine learning credentails

The metaData was created.



The model was stored in the repository and deployed sucessfully.



**MILESTONE 5:CREATING THE UI (DASHBOARD)**:  
We used the IBM CONOS DASHBOARD for feeding and visualising our data.

