

SENTIMENTAL ANALYSIS ON KABITHA KITCHEN

In [1]:

```
#Here we are going to nltk package for sentimental anlaysis and also install the required packages
for preprocessing
#Then the kabita kitchen dataset is read using .csv

import nltk
import pandas as pd
import numpy as np
import string
import re
stop_words = nltk.corpus.stopwords.words('english')
ps = nltk.PorterStemmer()
dataset_kb = pd.read_csv("kabitakitchen.csv", encoding='latin-1')
dataset_kb.head()
```

Out[1]:

	id	commentText	Labels
0	Ugy_CBM-_CKA3YqrzcB4AaABAg	Pudina ptta nhi dalu to	7
1	Ugy9mx9nuTWJu4dRac14AaABAg	Chiken kacha tu ni rhy ga sis	7
2	Ugz8T2MKLYucL3dM9nh4AaABAg	Hello mam, I love your all recipes.... ðð...	4
3	Ugx_1cCjRbCaDgL0FLF4AaABAg	Its awesome recipe plzz make handi chicken in ...	2
4	UgzLhKVAJ6NN3nZXyjN4AaABAg	Yeh jo measurement hai.....kitne logon ke ...	7

CLEAN TEXT

In [2]:

```
# As we can see from the above displayed dataset ,it contains stop words punctations ,upper case
# letters such features slow the model so the dataset should go some preprocceasing
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
# we create a corpus which contains the clean text and we will be using for vectorization
corpus = []
for i in range(0,4900):
    review = re.sub('[^a-zA-Z]', ' ', dataset_kb['commentText'][i]) #removes all the numbers, symbols
    # it keeps only the alphabets
    review = review.lower() #it converts all the alphabets into lower case
    review = review.split()
    ps = PorterStemmer() #it is used for stemming
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)
```

CREATING COUNT VECTORIZER FOR KABITHA KITCHEN DATASET

In [3]:

```
#we now import count vectorizer from sklearn.feature extraction
from sklearn.feature_extraction.text import CountVectorizer
cv_K = CountVectorizer() # creating count vectorizer
Xk_c = cv_K.fit_transform(corpus).toarray() #fitting the count vectorizer to the dataset
yk_c = dataset_kb.iloc[:,2].values
```

In [4]:

```
print(Xk_c.shape)
```

(4900, 4092)

In [5]:

corpus

Out[5]:

```
['pudina ptta nhi dalu',  
'chiken kacha tu ni rhi ga si',  
'hello mam love recip ingredi easili avail way explain good',  
'awesom recip plzz make handi chicken handi',  
'yeh jo measur hai kitn logon ke liy hai',  
'kabita mam tri ur egg biryani everyon hous love thank much delici ur recip',  
'cooker kar sakt na',  
'mujh bhot ach lagi apki respi mene subscrib kardia bhot ach',  
'mam dahi k jgh p kuch use kr skte h kya',  
'woooooooo yummmmmmm love',  
'perfect biryani recip apko follow kar k banaya acchi bani biryani',  
'hi didi alway curiou biryani made thank much put detail video weekend tri serv famili realli mot  
ivati',  
'thanx respect madam appn tel nahi dala',  
'made tast awesom thank kabita ji',  
'amaz',  
'nice video',  
'aur kya recip ko cokar bana sakt hepleas zarur repli kijiyy ga',  
'aur mam ism aap chawal kitn k g use kiya',  
'rice oil dale chalega',  
'masha allah yummi main thing u like video way u show step one one way u cook realli look beauti'  
,  
'thank friend post video make chicken biryani like thank u much love u friend dubai',  
'curd ko skip kar sakt h maa sahi bariyani masala chicken masala ensabko skip kar sakt h maa',  
'u show make merin chicken',  
'hi kabita agar briyani ki quantiti jyada ho bhi kya cook time rahegi',  
'u r awesom kabita ji',  
'chicken marinett oil ya ghee ni daleng wo ji bartan hum bna rahe h usk nich chipkegi nahi kya pl  
bttaiy na mujh bnani h',  
'love recip',  
'kabita didi plz tell us brand basmati rice use plz',  
'measur rice water',  
'u r sweet even get mani view comment read everyon comment respond gestur',  
'alreadi tri egg biryani time chicken mam master',  
'delici',  
'aap mujh bahut pasand ho mam',  
'though unabl follow languag step step prepar clear easi simpl method prepar dum briyani definit  
tri',  
'mast',  
'ap boht achi hain apki recip bhi kamal ki hai',  
'wao nice recip',  
'abhi tak isko baar bana chuki hu unexpress par comment pahli baar kar rahi hu',  
'good',  
'made today ur biryani receip tasti nd nice',  
'nice recip yummi',  
'nice recip',  
'well explain hit home prepar mani time n everyon love superb mam keep',  
'superb mam',  
'nice',  
'mam wheh mix rose water',  
'kewda water bhi use krte h kya ism',  
'simplest way cook biryani tri thank video',  
'brand shahi biryani masala chicken masala differ brand differ flavour pl advic',  
'agr dhi pudina na ho',  
'skip mint leav',  
'hi didi mujh puchna tha ki jo dahi apna dala oo kesa tha mene last time dahi dala dahi khata tha  
toh chiken gravi khata ho geya',  
'thank',  
'oooooh',  
'kya tamatar nahi dalna hai',  
'tast',  
'chicken ko freez mein nehi rakhn se koi prblm ho gaya kya di',  
'tri ur recip dum biryani good n tasti chicken got burnt littl avoid pleas give suggest',  
'shahi chicken masala combin yahan pe shahi masala nahi milta pl repli mam',  
'big process',  
'tri today came delici ad kade masala ingredi time kg chicken kg gm basmathi rice follow everi st  
ep came realli well famili love recip thank u share recip',
```

'kitn log khayeng',
'ab ye javitri kya hota hai',
'mene ye tri kiya delici thnkuu love recip mam',
'nice kabita si madurai tamil nadu',
'nice present',
'wow nice',
'kg chicken main kitna dahi cahiy',
'agar rice chicken ka quantiti kam ho time rahega',
'mene app ki biryani tri ki bahot achi bani sab ko achi lagi lekin muje agar jada grevi chahi ky a kare thank',
'maam kya ghe ke jagah batter use kar sakt kya',
'much time cook stove make beef',
'u r best chef youtub mam',
'mam aapn kitn chawal kitna pani liya pakan ke liy kyuki jb bhi mai banati hu tb chawal sahj se c ook hote hi nhi plzz mam ek baar sahi se bata dijiy',
'super recip mam tri awesom thank lot super keep go',
'amaz recip best',
'mam chikn ke pi kacch nahi raheng pl mere qustion ka aanswer dena pl',
'hai dear kerala big fan hyderabad biriyani doubt non stick pan case type vessel use dum step',
'thank',
'tri testi',
'cook neat clean',
'world best briyani move canada crave mom biryani tast exactli like thank much must tryyyyi',
'hii maam question ask u taken g chicken di biryani time accord vari amount chicken time stil vl chang time pl help',
'bhauji agar mutton kg aur chawal kg ho time kya hoga',
'itna aata zayaah krna dum dene k lye himaqt nahi',
'pahl mai sochta tha doctor se hi sadi karunga iss video ko dekhne ke bad lagta hai hotel manag be st hogi mere liy husband realli lucki wish someday wife coock',
'tri yummi thank u kabita mam',
'perfect recip chicken dum biriyani simpli follow step biriyani awsum first tri thank share descr ib video',
'hello kabita ji good day dubai cook today biryani cook delici thank much',
'made exactli u knw insan awesom best ever restaur thank much ms kabita',
'nice recip',
'kukar ni bna skte',
'hyderabad khachi ghost ki biryani',
'hello mam tri biryani recip today turn amaz thank alot delici biryani india cook channel kabita kitchen',
'hi mam tri mani receip upma pea nut chutney phoa dosa biryani recip turn good appreci thank u mu ch clear explain ur recip god bless keep',
'madam ye rice kitn rs waal ha',
'oh wow simpl yet delici tri',
'bhai koy tri mat karna ne tri kiya ra pura chikan jal gaya',
'tri great job',
'hi kabita tri biriyani tasti lot water collect bottom vessel solv',
'madam chicken dahi daln se chicken khata nahi hota hai na',
'madam simpl wale chawal bna skte hai kya',
'madam chicken kaun sa legn leg piec wala ki jo dukan pr kaat kr dete hai',
'madam shahi biryani masala kaisey banay ya phir bahar karidley',
'measur pretti accur simpl step step explan best wish malaysia',
'agar kg chawal lege kitna dahi lena hai plz pti',
'sahi use aap jyda mere hai',
'nice kabitaji easi inform',
'raam chicken b j p walo se bach ke rahiyo mulaa ghosit kar deng sorri joke',
'way cook good recip fantast tri biryani saunday pleas upload recip chicken momo love u much kabita dii',
'nice awesom',
'atta k dough crack aa ja rha aur pressur releas ho ja rhe solut',
'kabita mam reason learn cook made cook hobbi bachelor roommat alway wait come offic make someth credit goe mam realli awesom',
'hello mam tri recip today awesom never tast biryani restaur earlier',
'nice',
'u r magician ju love everi recip',
'yummi recip nice tri tasti',
'bht achchi bani thi chicken dum biryani thanku much kabita di',
'sorri u pleas cook time mani minut',
'mam aapki recip bahut achha laga aur aapka bathan ka tarika bhi bahuti achha hai mam mera ek saw al hai aapn usko seal kark banaya hai agar mai iss beiyani ko pressur cooker banan se kaisa rahega pl mam aap bathayeng',
'go first cook ever life',
'delici kabita mam u realli great',
'nice recip',
'apn aata bataya hi nahi',
'low medium flame either wud b low medium',
'delici time consum',
'never cook biryani look video tri look ever nice',

'perfect recip mam thank lot keep good work',
'tri today hope come gud',
'oil bilkul daala hi nahi aap ne',
'mani peopl serv want tri tomorrow',
'bahut hi achhi recip hai mujh bahut pasand hai hmesa market hi khane jana padta tha lekin ab ye
problem bhi door ho gyi thank mam',
'nice cook style',
'biryani awesom made home major problem face make biryani bottom use get burn way u said noth got
burn one thing add tabl spoon oil marin chicken even give tast',
'wonder chef instructor keep good work',
'u ad cook oil wil work',
'pl learn telugu',
'jab ghee dalt hai uska tast ya smell aata hai kya kyu ki mujh ghee kewal roti k saath hi aacha l
agta hai',
'biryani pan ki jgah cookar bna skte h kya',
'want dum biriyani want normal one seal edg pan answer pleas',
'tq sister amaz tri superb sister',
'make kg chicken biryani time',
'hi thank u',
'super recip amaz',
'hello mam plzz want recip biryani masala powder plzzz',
'thank',
'nice recip',
'good mam nice recip',
'today tri recip awesom husband love thank alot make day',
'time explain kijiyy plzzzzz',
'biryani ka time nhi samjha',
'kitina bhi jayada biriyani ho time rahega na plz repli maam',
'agar tawa nahi hai direct heat kar sakt hai',
'hii mam mera biryani bht acha bna bt chckn hlka sa lg jata h aisa q hota h',
'bahot best',
'thanx mam ur biryani superb famili love much',
'pudina patta na ho kadi ptta use kr sakt h',
'man dahi ko replac kark kuch aur nhi daal sakt',
'awesom recip thank much',
'love content keep good work',
'hi mam peopl ke liy kitna amount rice chicken use krna chahiy plzzzzz repli mam',
'ye meri rd tym h aapki recip follow karti hnu ye toh restaur ki biriyani se v bahat bahat bahat
acha h juzz lv well done mam',
'see peopl posit comment definit make want tri recip give go weekend thank share recip',
'mam nest chef',
'kesar na ho madam',
'pleas app fri chicken biriyani recip ki video baaniy',
'pleas never forget write ingredi descript',
'explain well even cook hous make biryani well soooo yummm yummm one best biryani ever tast thank g
ood recip biryani home',
'itn der tak biryani banan se accha bahar se hi order kar lo',
'kitchen king acha sabji poudar masala bataiy na',
'nice biryani',
'maam one nicest youtub countri keep good work go',
'chicken masala lagakar fridg kitna samai tak rakh sakht hai',
'agar thora masal dar chay kya kre',
'u great mam cute',
'india live germani ur recip life saver thankyu',
'marin chicken cook put rice',
'hi kavita tire ur recip delici one problem rice dri pleas suggest make rice soft thank',
'mam mujh nai khana bana ke itna lamba prose kuch chhota aasan sa btaiy plzz',
'skip curd mam',
'brand basmati recommend recip recommend parboil basmati sela seila regular basmati',
'pudina ko skip nhi kar skte kya qki ye local market avail nhi hote',
'calori one go',
'mam agar aloo bokhar daln ho tho wo chicken ke andar daal dene h jo marin kart howe kya chicken
ko pehl cook nei kar sakt ese darr lagta h ke chicken pakk naa yaa jal na jae pleas repli',
'mam chicken ko ach saaf krlijiy gnda h jb chicken itna gnda h biryani ka test kais acha ho akta pl
z agli recipi chicken ko ach saaf kriyega safai pr dhiyan de',
'tri recip today chicken cook beauti well rice masala got burn n stuck bottom due littl steam pas
s dum dough two leakag spot',
'nice recip tri',
'hi kabita tri biryani home tast good use flour seal pot seal got leak middl biryani got burn bot
tom idea properli seal pot',
'hi dear made biryani yesterday per instruct lower chicken got burn ruin whole dish',
'kg chicken biryani much time requir cook chicken say pleas',
'thick pot ki jagah par kya use kar sakt h',
'meat look uncook',
'make shahi biryani masala',
'kabita ji apki recip bohota simpl hai ap jesh bol kar kart hai bhot acha hai',
'mam oil kab dalna hai',

'take time',
'tri sure thank',
'superb mam',
'much spiciy',
'also tri recip multipl time realli yummi recip anyon beginn u check cooker biryani recip easi quick delici',
'chicken pak jaayega ais',
'maam kia hum chicken ki layr laga skate hai',
'iss achchi meri mumma bnati h',
'aap master chef ho',
'oh wow',
'ye kachi dum biriyani',
'tri n realli soo yummi mam thank lot upload delici receip',
'galti sy mint low ker dia ab kia keron',
'chicken',
'love',
'superb',
'nice',
'hello mam super',
'maam ye jo apn banaya hai vo kitn lg ke liy hai tamatar nai use krna hota hai',
'kabitaji main aapki recip last year banayi thi bahut mast bani thi bacchonko ab tak yaad hai ab savan ka mahina chalu hone se pahal banani hai isliy vapa aapki ye haidrabadi biryani ki video dekh rahi hoon thank kabitaji aapka recip bataneka tarika itna acchha hai ki sab aasan ho jata hai',
'mam apn oil nehi dala chicken marin',
'today first time prepar biryani turn realli well thank',
'like recip recip use',
'steam toh nikla hi nehi end mein iski cbi jaanch honi chahiy',
'pehl khooD sikho phir sab ko sikha',
'love recip chicken dam biryani kabita mam realli awesom mashaa allah deffin tri recip whenever mad e biryani home everi time got confus rice quantiti biryani look like lott masala insiden less rice pleas give tip correct rice quantiti',
'love way explain nice fast tri excel',
'mam chicken rubber jaisa cook hota hai mere se make soft chicken',
'tri rspi realli nice cook biryani thnqq keep share ur rspi',
'agar shahi biryani masala na ho kya kru',
'yummi mam ham tomato dal sakt h',
'yummi',
'thank share cant wait make',
'rice ko kitn time ubal',
'easi tasti recip thank',
'agar refriger naho',
'thank share recipi',
'superbbb',
'mam ays pakan se kiya chicken acchi tarha pak jayga',
'curd mix karn pe chicken agar jyda khatta ho gya',
'franc video help thank much love food',
'love darl',
'nice tast',
'mam mein jb v bant hu ye jal jata h aap kuch tip doo taki n jale',
'wonder',
'hey dear kabita thank u make soooooo easi tast like tandoor biryani big fan love u lot',
'u r super sister',
'today make recip yummi thank u much',
'har chiz ko bhut hi simpl bna deti ho mam aap',
'superb',
'kabita mene apki recip se biryani bnai bahut hi tasti bni thank nicerecip',
'biryani ban jane ke baad turant dhakkan hatana zaroori hai kya agar hum ghant baad hatayeng kya tast mein fark padega',
'use cooker',
'good',
'ye ham pardesh walon k liy kafi qeemti hai thank maim',
'thank',
'nice briyani kabita',
'wow nice',
'wow best home made biryani ever thank lot kabita didi',
'mam ghee k jgha refin oil use kr skte hai',
'tri yesterday bhot delici bani dahi phat gya main aapki recip step step follow ki thi phir kya galti ho sakti hai',
'tumhari recip tast kais kareng',
'much time cook kg biryani',
'could cook meat first layer rice',
'sister aapka video dekhne ke baad mene life first time biryani banai aur jab biryani ban gayi toh gharwal ungliya chatat reh gay thank sister',
'agr chicken gram le aur rice b gram kya masalo aur sbhi ingredi ka measur itna hi rkh skte h',
'agar rice chicken ki quantiti aap jo dikhay hain uss doubl ho kya biriyani cook krne ki time bhi doubl hogi yaa phir jitna time apn bataya utn mee hi ho jayega plz an mee jaldi mujh yeh suggest k

i bohot jarurat hai',
'nice recip mam maja agya',
'pyaz barik kate hue nice recip',
'hello kabita aaj main apn bataya huy tarik se biryani banayi bahut acchi bani thi thank share r
ecep us',
'tast superb satisfi',
'awwsom recip',
'hiee di how uh hope fine naa bdw told uh yesterday made biriyani n truli work wonder lucki charm
thank di god bless luv u load',
'wahh mam apka koi jawab nhi bahut hi achh se har ek chij batati h ap',
'best',
'never tri biriyani except made tri n know turn well coz u lucki charm alway ur dish help grow lu
v uu di b bless',
'recip best',
'best',
'get ghee',
'maam log k liy bnane se cook time rahega ya tym lagega',
'soak rice',
'non stick cookwar still prep wud remain',
'mam ghee ki jgh refin oil use kr skte',
'mam mujh ek doubt hai kya chicken sach accha pak jayega qki usual chicken bnane jyada time lagta
hai aur chicken accha nahi hota',
'wow nice chicken biriyani recip mam tasti',
'instead ghee use oil',
'mam ghee ki jagah tel use kr sakt h kya',
'like u bcoz ur cook process clear finis first time cook video subscib ur site keep best luck',
'hello kabita made biryani came realli well thank share recip',
'hi mam tri make biryani bit dri may know would went wrong n also masala becam less tell solut th
ing thank mam',
'kya kesar dalna jaruri h usk jgh pr jldi ni use kr skte',
'hi mam mene aaj chicken dum biryani banay sab ko bhut pasand aay yummi yummi thank sooooo much',
'nice',
'mam mani piec come gram chicken cut cut big piec',
'woow',
'dahi mandatori',
'nice recip question much total time taken readi biryani',
'tasti dear mam',
'ur voic resembl like air hostess explain plane',
'reali tasti tri thank u show show fri chicken kabab',
'kabita mam chang add cashew nut almond past almost tblspn opinion',
'yr video help lot kitchen thanx give us love video',
'humko fri chiken biriyani batayiy',
'kabita ji suppppper',
'mam import question mam apn esmai water nahi dala aur lagabhag mint cook kiya nich jo chicken hai
wo jaal jayega aur baki chiz sahi se paak v nahi payega aur hamn ese pack v kar diya ho sakta hai
ye blast kar jay kyouunki mai jaha v apn samn chicken ya matan ki biryani pakt dekha hai usmai gravi
hoti hai apn nahi dala koi v gravi ya water eska kya trick hai plz mam give detail',
'also make like',
'easi n delici learn frm u',
'nice',
'mam pleas upload haleem recip',
'hello sister ingredi use gm chicken make instanc g chicken doubl ingredi base amount meat amount
ingredi remain',
'hello sorri watch video earlier love recip wonder cook hand contain magic tell',
'mam hie nivedita main kaun se brand ka basmati chawal use karun mere biryani chawal achh nae hot
e main india gate basmati use karti',
'good recip tri',
'didi app reshma ji math bolo jt reshma bolna',
'didi app aisach new rcpi post karo hum app ka har ek vdo hmara frd ko jarur share karagay',
'tqqq didi muja repli daynay ka khaliea',
'kabitha didi ek doubt appko katnay recip cook karsakthi',
'kabitha mam appki lookg cuteeeeeee',
'ye chicken nicha oli one layer dalna kya g',
'wow',
'delici recip cook cooker',
'need go latin indian korean groceri store collect ingredi still guarante find everyth',
'tri recepi soo delici thanku soo much',
'amaz mam',
'good',
'smell kaisi khatamkaron',
'kesar dalni sy smell q ati hai',
'pakan kitna time lagega',
'love cook style u know aapki cook bilkul wais hi hoti h jaisa mugh pasand hai',
'tri awesom tasti di love much',
'mam aupka languag bahut acha lagta hai',
'hi use yoghurt instead curd',
'much testi',

'tri biryani home see video super tasti',
'didi kayea may kesor ko pani na vigo saktay hu plz answar q may aj app e recip bannung plzzzzzz
answar',
'mam mere ghar dum biriyani banan pr usm halka sa meetha flavour rha tha yesa kyu mam pleas repli
,
'learnt cook video',
'mama mere adh kilo chicken ke pais upper se mera time mere ko waapi chahiy',
'ise bekar biryani ni dekhi pyaj jal gya upar adh chawal b colour ni aya',
'nice recip',
'cute voic killer',
'tri yesterday tasti thank u much',
'mam main kg chicken ki biriyani banana chahti hu kya main tarah bana sakti hu ki time',
'food colour nahi dene se vi hoga kya',
'long video cut describ whole step nice watch video anyon fall love mam thank glad subscrib',
'apko chicken acch se saf krna chahi',
'watch ur chicken biryani recip realli miss hostel sure tri go home',
'u knw unhealthi ur biriyani becom u add food colour store bought chicken nd biriyani masala u dr
ain ur rice plastic contain seen ur video u alway use non stick mostli everyon still extrem
danger',
'khaak coma chalijaa',
'kg chckn much rice need maam',
'tri follow ur step came delici thank u much recip mam',
'hi karnataka bhut hi easi recip hai ek bt hai kya chicken puri tarah se pak jayega kyunki aap ne
chicken ko kisi b level pakaya nhi marin kiya hai aur jab last aapn jo low flame rakha kya chicken
poori tarah se pakega',
'yeh main tri ki bhut hi laziz bani easi bhi lagi thankyou much mam',
'thank beauti recipe use maida instead wheat flour wrape around handi',
'didi boil rice ghee kyun dalt hain',
'cooker main bana sakt hain kya',
'time consum recip job wala cant give much time prepar',
'mam mettha attar nahi diya apn',
'yeh miytha ya khatta dahi tha mam',
'nice',
'thank maam',
'aur aapn chicken kaccha rkha',
'use instead biryani masala skip',
'toooo gud everyon like thank kabita wonder recip',
'today prepar let see tast',
'chawal kitna daleng',
'food colour dal sakt hai na kaser k bdle',
'kaun sa oil dalu',
'oil kauna sa daleng',
'kyon makhan shi nhi rhega',
'ghee k bdle makhan dal sakt h kya',
'mam main ne tri ki aap ke batay gay tarik se par masala jal gaya puri pate ko chipak gaya q aisa
man plz bataiy',
'wow',
'woooooow',
'wao',
'masla jal gya kya kru',
'mam apn pan mei bnai hair pl tell name utensil',
'mam use chicken curri msala instead chicken msala',
'kabita ji thoda bataiy chicken biriyani banan keliy agar hum aluminum pot use kareng kitna total
kitna minut ga k uper bithayeng',
'kevda essenc',
'dont like',
'prawn biriyani recip',
'nice',
'sooooooper',
'nice',
'hi kabita look yummi suppos use big crockpot make good advic thank share knowledg appreci video'
,
'show make biryani masala',
'biryani look good think put much garam masala chicken',
'thank u mam',
'mem apki smile',
'biryani khanek liy adha kilo ata fekana padega isiliy hamko pasand nehi aya',
'ad boneless chicken make differ tast cook method',
'apn alu nahi dala aur chicken ko sirf marin kar ke usa dum mea dal diya apn tu chicken ko marin
ke baad fri nahi kiya pleas bhut confus hai next video mea achha se explain karya ga agar koy sugg
est dena hai tu repli mea de digia ga',
'jal gaya tha',
'kg chicken ke liy kitta rice curd kitta gram lgega',
'kabita mam tri home tast super love delici recip ankita best chef ever',
'easi biryani recip',
'hungri nice biryani',
'mam aapki sari recip mst h',

'hi kabita cook cooker',
 'dont pot make layer n cook dt pressur cooker plz suggest utensil thn di pot',
 'chicken biryani thank mam',
 'tasti biryani',
 'jharkhand',
 'nyc',
 'tri menu piec much time cook',
 'plz put recip seeg biryani',
 'u make shahi biryani masala',
 'kabitaji kg biryani ka ingredi batayiy pleas',
 'thank share video',
 'ok mast welcom',
 'waw nice thank yammi realli',
 'mam ye dough toh wast ho jaega',
 'cow ghee gagan vanaspati ghee',
 'bohat acchi biryani lag rahi hai aaj hi banaungi',
 'nice mam',
 'use yellow food colour',
 'get shahi biriyani masala also chicken masala pleas let know',
 'nice video mam',
 'super',
 'best',
 'tste',
 'mam biryani ko aur jyada masaledar krne keliy kya krna chahiy',
 'like recip',
 'chicken gal hi nahi raha esa ku',
 'nice mam main banaya bhut testi bani thank',
 'nice',
 'yummi',
 'tasti thank kabita ji',
 'yummi',
 'mam agr itta tym tk chikn ko naa chlaau wo pan chipkega n',
 'thank u sekhani ka liy allah mngheban',
 'bhaut easi se mam aap ni bataya',
 'hi mam good morn mujh aaj chicken biryani banana hai peopl pl give proport rice curd chicken day
 made chicken biryani awesom tast love today make biryani guest pl repli',
 'aap recip acha banati ho',
 'madam briyani ke rice mai shahi jira hota hai',
 'hyderabadi biriyani ki koi jawab nehi koi uska samna nehi kar sakta',
 'like cook time heat precaut best',
 'record video phone camer',
 'hello kabita tri chiken biryane',
 'super',
 'mam ye jo ate ko doug se isko dhakan ke charo taraf lagaya achhes set rakha ne k lia ham kya pre
 ssur cooker use nehi kar sakt aur non stick pan k badal normal pan use kareng chicken nich lagega n
 ehi na plz an mam',
 'mam bina kesar k ni bn skta kya ghee ki jagah oil use ho skta h kya',
 'thank much',
 'supop',
 'love ur grt cook big fan ur',
 'chiken kacha e dal dey gye mai smell aye gi chicken ki mujhi pasnd nai ayi',
 'kabita ji chicken uncook hi reh jayega chicken fri place',
 'mam much cup gram basmati rice mani piec come gram chicken mention cut chicken piec larg piec me
 an mani piec come gram cut larg piec',
 'khatta dahi add krna h ya meetha dahi',
 'ossam mam',
 'briyani got popular inlaw credit goe dear thanku much share kepp rock like oli',
 'ism konsa flavour ka lagta hai colour',
 'tri yummm mam person k liy chaawl chicken dahi piyaz kitna lage ga kitna time cook krna pade ga',
 ',
 'maem ky hm ise pan k jaga cooker n bna skte',
 'pleas tell us use artificci color food recip',
 'hello aap k sab recip hm tri krte sab bahut tasti banti thank u much',
 'aap ka jawab nahi',
 'mumma ne tri kiya tha',
 'best recip mam',
 'behan pyaz ko fri karna pehl seek lo plz biryani ka mazak mat maro',
 'apn rose water screw pine nhi dala',
 'colour dalna complsari hai kya',
 'mam aalu dalna hua kais daleng',
 'kabita mam make chicken biryani follow recip alway came wonder question use proport curd almost
 gm kg chicken appear lot could suggest much curd rice consid kg chicken',
 'video got view red colour rice',
 'naaam bada darshan chota satisfi recip lot mistak',
 'nice video',
 'tri make bt bottom layer get burn tip avoid pleas help',
 'pleas advis cook pressur cooker lid cover instead pan air lock dough use heat time cook pressur

'pressur cooker instead pan',
 'mam aapki dish hameesha heh bahut aachchi rehti hai',
 'badiya nice',
 'biggest fan cook kabita sister',
 'kabi mayri post ka bhi answer day dia karo dear sister',
 'kabita ji wan tri recip chicken masala readi made shayad koi particular name ho haray han agar r
 ecip hy bata dain plz masala ki plz waitinf',
 'rice use',
 'tri hubbi love thank kabita',
 'nice',
 'quantiti jyada ho dam krne ka tymng rhega ya jyada',
 'nice',
 'wow tri delici thank',
 'kikar thi biryani mai banaya',
 'tarik mujh nahi lagta masala aur chicken pak gaya hoga',
 'thank much realli help',
 'rrt',
 'mam plzz make chicken chilli',
 'kabita mam nail shukriya ais hi receipi upload kart rahiy',
 'use pressur cooker cook rice',
 'tasti',
 'use cooker instead pan',
 'aap log keser use kart hain karo lakin koi aap logo jaisa itna amir nahi hota ki itni mehengi si
 j kharid sake aap logo ko paisa milta hain hame kiya milega meri gurajih hain ki simpl tarik se
 khna bani usm jyada atcha hain age se mehengi sij mat batay kyonki sab lob amir nahi hota',
 'tri chicken burn bad tri',
 'mam mene bhi banai thi ye meri biryani nich se jal gai thi',
 'shahi biryani masala',
 'kitn log mare ye biryani kha k hehe',
 'mam presur cooker',
 'wish u wife p',
 'sister food color z nt gd fa hlth',
 'deserv subscrib mam',
 'realli like recip bcz simpl easi quick tasti inde use mace powder instead whole mace',
 'nice',
 'superb superb chicken biyrin',
 'kiti chaan banvali biryani',
 'wow',
 'awesom easi prepar di',
 'kya jada pyaj daln se kuch problem hoga',
 'ag kesar nai dale',
 'one best outcom wht ive made watch video ju follow wht say done love kabita thank recip',
 'tri cook tomorrow',
 'nich oil nay daala chicken k toh nich jalega nay chicken',
 'wow',
 'eat',
 'u show u cook n record shoot time wo b itna perfectli',
 'way cook hyderabad',
 'voic good',
 'ook',
 'vari nice',
 'nice recip',
 'mam layer kartr waqt bina kuchh ghee laga k krne se nich nhi lag jayega kya chicken k piec n ham
 ek layer chicken nek layer rice fie ek layer chicken ek layer rice ais nhi kar sakt kya',
 'big fan keep post recip lot love',
 'thank u biryani liy',
 'keep overnight freez fir marin',
 'infreez',
 'good recip',
 'make without curd shahi biryani masala',
 'husband say jadu agaya h tumhr hatho bcoz ur receip thank u much mam lot love',
 'nice mam tri thanku itni acchi easi resip ko share kiya',
 'ek baat bataiy main suna hai ki non vej se dudh dahi ghi ko dur rakha jata hai aur khane ke se g
 hant baad hi dudh dahi ghi khana chahiy warna charm rog hota hai aap daal bhi rahi hain sath bat k
 aha tak sachchai hai',
 'mam saffron import',
 'mam pleas app ne jo colour use kia hae bo kaha milta hae uska price kya hae bataei',
 'kabita mam ur best',
 'kabita mam mai aap ki subscrib hu aur aap bahot achchi hai mujh aap ke video dekhna mujh bahot p
 asant hai like kabita kichan',
 'nice plzz telm chicken khucha nai hoga kyunki mein suna tha chicken fri karna hota hai plzz mer
 a confus dorr karo',
 'ap ki sab recip achi hi hote h',
 'mam ur fan',
 'love ur biryani made twice everyon love thank u ur wonder recip',
 'mam aap ne oil use nhi kya base bhi oil nhi liya plz rep mam',
 'time vummi u get'.

'cume yummie a goo',
 'layer step want ue microven dum toh mujha kya karni hogi cz ga layer ka band mujha dare lagthi j
 al na ka plz help microwav dum kais karu',
 'descript box lekhna nahi aata lekha kara nah',
 'tnx tasti n chicken poora pak gaya tnx love',
 'c result baba ji pl meri birayani ban jay thik thik',
 'accord good chef',
 'yummmmyyyyyyi',
 'chicken pakk gayaa hai puri tarah pl tell mam',
 'aap ko modi pakoda malum hai anti jii',
 'love chicken mam ap chiku ki aur v recip jarur bnay hame btay',
 'oil use nhi kyea apna',
 'use pressur cooker instead cover top dough',
 'niceeeeeee',
 'nice tri best hyderabadi biryani world famou',
 'thank mam',
 'easi mam l tri definitli',
 'mam biryani dahi skip kar sakt h mere bacch ko dahi se allergi h biryani bina dahi ke ban sakti
 h kya ya dahi ka koi option h',
 'hi tri way littl bit steam come ok done someth rong plz repli st time cook',
 'superb',
 'ok',
 'mam tamatu nibbu yuz nahi krte',
 'good',
 'hi use sour curd food curd normal sweet curd sorri know noth cook want surpris wife
 anniversari',
 'yummi',
 'madam mujh aapki biryani bnane ki trick achhi lagi par video ko kese download karuu',
 'tri process today delici bt rice becom dri wonder el cud make rice dri plz suggest',
 'main thing make recip look easi',
 'use ke ora water pleas repli',
 'wife follow recip last night came good ask make tonight',
 'kavita madam main toh exactli kiya jais aap dikhay likh mera biryani jaal gaya aisa kiu hua',
 'yummi',
 'mention ingredi descript box pleas',
 'today mom gave tri went unsucces rice cook well shown video chicken peic complet burnt chicken
 turn coal heat biryani exact time exact flame shown video spoil plan mayb day',
 'tri last month yummi chicken got burnt avoid',
 'excel present clean cook keep post n good luck',
 'delici kabita ji thank u soo much',
 'hi kabita today made chicken briyani per ur video came realli outstand yummi husband love much u
 made day ur delici yummi tasti briyani love much thnkuuu thnkuuu',
 'time enough make chicken tender al dent type',
 'marin main oil nahi add karna',
 'madam ji chicken mix karn ka tariqa wrong h',
 'hello kabita watch video video realli amaz way cook realli appreci keep best luck',
 'superb fan cook',
 'ghee k badl oil use kar sakt hain kya',
 'nice chicken biryani thank video ko bhejn ke liy kavita',
 'love u r cook style',
 'good biryani made today style',
 'gem much love',
 'mem ism kesar use krna zruri',
 'kabita mam follow u regularli n ur dish nvr fail get appreciaion mother',
 'thank share mam',
 'pz mam rice ko boil krni ka time batin persent kitna time main boil hoga mra rice kbi ziyada kbi
 kam soft hoga pz batin rice kitna time main boil hoga',
 'hungri kabita ji',
 'u make everi dish simpl make fall love cook u r bless',
 'amaz sure tri tawa cook without dn much longer increas flame non stick pan work',
 'agar kg n biryani banay toh time chang hoga ya rahega',
 'add instead biryani masala',
 'owowwo nice one mam madam parson chicken biryani dum qanti ketna chiya',
 'thank recip wonder want add potato step add',
 'awesom',
 'mam bataiy na ki biryani essenc rose water kewra water marinad karth waqt tsp aur rice ko banath
 waqt tsp dal sakhti hu pleas mam mera doubt bataiy',
 'wow mam nice video thank much',
 'mam freez rakhna jaruri kya',
 'basmati rice use rose water brand',
 'definit gonna tri one last time cook slightli differ recip ad kewra rose water someth went wrong
 smell overpow aroma biryani mayb misjudg quantiti rose water ad kewra water made tast like
 agarbatti use rose kewra water essenc',
 'kuch oil nehi lagana chahiy masala jal nehi jayega',
 'aapn kons brand ka basmati rice use kiya hai',
 'kabita u r sch amaz cook',
 'mam agar chicken ho toh kaisay bansakhtay hai aur agar zaida banai ho matlab ppl kay liy',
 'ye pak chuki hogi madam chicken nice'

'ye pak chuti hoga madam chicken piec',
'hey kabita aap kaunsa brnd ka pan aur vessel use krthi',
'kya hum biryani ko pressur cooker bana sakt hai',
'chicken cook well pretti sour tast due curd tast like shit',
'nice ji',
'kabita ji apki recip dekhni pehl like kr diji apki sari recip achhi hoti h',
'nice prepar tri nirmal',
'kabita mam went recip chicken dum biryani earlier ad biryani essenc kewra water rose water
biryani add put much add marinad cook rice',
'hii kabita didi mein aaj aapki yeh recip tri ki hai bohot sundar dikh rahi hai abhi todo ter mei
n readi ho jayegi fir tast kark aapko repli deti happy makar sankranti',
'skip black cardamom chang tastr',
'yammmy',
'sara blood lga tha dhekhni k bad hme chicken bireyani se nafarat ho gai',
'mam apni chicken dhoya ni',
'thanku di',
'wowwww super mam tri tomorrow',
'bhabhi mere liji ek plet parsal kardo n pleas',
'mam agar last mein rice mein thodi si kasar kani reh jay mean kach reh jayein wat wud solut',
'nice',
'aapki voic bohot achhi hai',
'mam agar nonstick pan na ho tho chicken jalega na',
'acha banaya apni madam',
'cute',
'nice kabita ji',
'zam zam biryani dikhay na',
'ais banti biryani',
'chicken biryani recip',
'mam plz chicken corma ki recip batji',
'nice',
'yummmmyy',
'mine aaj banai biryani',
'u r tip r realli good',
'link recip pleas',
'mam agar kg biryani banay kitna kitna dalna padega sab chiz',
'mam kya mein biryani essenc rose water kewra water sab tsp marin karth waqt aur rice baneth waqt
tsp dal sakthi hu aur dhakad phool marinad karth waqt aur rice baneth use kar sakthi hu aur agar u
se kar sakthi hu tho kina dhaakad phool use karu mam pleas repli',
'mam always see ur recip realli like ur recip',
'didi mera nich sab jal gya aaj tri ki di jal gya sab',
'mam ur chicken biryani outstandig like ur recipi',
'dislik comment',
'didi mujh biryani ki dukan kholni h har baar chicken jal jata h aachi tarah seal karn ka tareeka
batayein jiss bhaap bahar na nikl pleas bhut zaroori h',
'gram chicken se jada nahi bana shaktya key',
'k thank mam',
'welcom mam',
'ye pressur cooker ho skta h kya',
'hii mam ye biryani hum aluminim jarmal ke bhagon bana sakt hai kya plz repli',
'ye nich se chipkega nai',
'tri recip everyon like thank kabita',
'hello mam aap ki har recip bhut bdhiya hoti hai',
'mam ek question hei chicken ko cook nhi korna hei',
'infreez cant wr keep room temp',
'thank video biryani amaz expect tast guy plzzz watch video u r prepar',
'gaavran chicken use kar sakt kya',
'gaavran rice two time think need cook rice cook chicken cook yet know could somebody tell like mayb
basmati rice strong differ smell anyway love pakistani food',
'thank much video',
'omg best recip mom cook togeth follow recip turn good finger like good make sure share channel f
riend famili thank much recip pleas upload video like',
'kavita mam apni direct chicken marin kar k biriyani banayi ism chicken achha se boil hoga na aur
chawal jal ni jayega',
'thanku mam apki ai recip bohot help hai thanku',
'aami ki ektu oil use kort pari',
'ei biriyani tee kono oil use nei',
'ami ajk banacch ei ta',
'good',
'superb kabita',
'wow',
'wow mom home tri biryani',
'add black chilli',
'mam mu karithili je pita lagila',
'use aluminium foil instead flour',
'aaataa lagaan k badl agr biriyani coocker dam ko rakh tho kaisa hai',
'like cook',
'pleas mam bataiji',
'mam ism curd milana jiruri hai'

mam ism curd milana juit hai ,
'yammi',
'agar chicken pakayng nhi kacchha nhi lgega',
'mam kya mein tsp tandoori chicken masala dal sakhti hu tast',
'christma pakka nd sister make biryani',
'',
'nice video',
'kabita neat perfect recip love hope get recip mainli chicken mutton egg fish',
'kabita mem tank aj tri kiya',
'mam chicken nich lag jaati koi trick hain jiss wo nech lage nai',
'like style',
'thankyou mam prepar dish wife',
'thank aunti',
'nice recip',
'zaiqa e yasmin also good',
'kya ism rose water dal sakt h',
'nice',
'kavita main bhi tri kiya mere famili guest ko paasand ayi main ye recip sab kay sath share kee',
'nice thank much',
'good',
'aata kya vo wast ho jaega',
'nice biryani mam',
'mam shahi biryani masala ki jagah chicken biryani masala use kr sakti',
'mujh question puchna ha aap kitna litr ka pot ye biryani banayi ha mera pot cover karn ke liy gl
ass lid diya gaya ha kya biryani ko dum dene ke liy glass lid ko aata se cover kar sakta hu kya is
s biryani pudina use nhi karn se chalega mean use nahi karn se tast kharab nhi hoga na kindly repl
i',
'agar biriyani jal jayaga hum kaya kara',
'superb',
'fridg mean defridg',
'mere pa pudinapatta nehi hein koi kuch dal sakti hun bataiye kal morn mein banungi',
'thank u much give nice idea make biryani',
'nice way make dum biryani',
'awesom made dish today turn tasit thank lot',
'kabita mam kya mein black jeera tsp marinad kart waqt aur rice banat waqt tsp use kar sakhti hu'
,
'thank advis k liy age tri karungi',
'good',
'kesar kon wala h eag kesar use kr skte h kya',
'tank',
'confus',
'mam rice ki quatiti cup mai tell kari plz kitnay cup rice leay apnay cup size plz',
'sorri kavita ji bura mat maniy jo mera experi wo share kiya pura ghar pareshan ho gaya bhuk se a
bhi khayi pura ghar dantn laga k apn ye kya bana diya',
'pata ni kesi biryani itna time lag gaya biryani jal gayi nich se sari ki sari ism photo option n
i warna apko daal deta chicken pata hi ni kaha gaya sara jal gaya',
'wow',
'steam nikla',
'superrrr mam',
'hello aunti pleas adopt',
'look tasti make today',
'mam kya mein green chilli ka past use kar sakhti hu',
'nice one',
'mam thand k mosam bina freez rakh v marin ho skta h kya',
'like ur reciepi nd gud job di',
'nice',
'kabita ji ur small littl fan',
'superb',
'behan piyaj kiya hota hao',
'super mam',
'use steel pan ye stop chicken get burnt',
'mane banaya tha habbi huya tha',
'mam mene ek baar banaya color konsa use karna hoga',
'nice',
'hungri',
'hello aap ki recip follow ker ti hu ye briyani bani per meri briyani chicken jal gya nich se tim
e asa hua time jo aap ne batya hai wo hi kiya tha',
'aap bht piyari lagti hai awr bht meetha bolti hain',
'mam itni time main ey jalega toh nahi na nich se',
'good',
'v v v v good',
'nice thank',
'love u yar',
'didnt know put ghee moistur rice',
'hydrabadi bryani lemon juic b daltey hai briyani dum denay pahlay humloag',
'pl write ingredi',
'pagal ho kya aap kuch bhi bnati ho aap jb chicken kacha hai rice pakka huwa hai jb usko hum payen
~ th chawal dal davenge samiki aap meri behan di'

g to chawal gai jayega samjhi aap meri bahan ji',
'kavita mam kya ham khade masalo ko grinder mix kark daal sakt hai',
'marin chicken overnight soft tast would odd',
'tri',
'mujh nhi aata tha kuch v bana apk video se mai bahot ach se bana pai nice',
'food colour naaa',
'majedar biryani definit go make sure love u thank u much mam',
'nice recip like',
'yummi tri',
'mam kya mein hour ke liy chicken ko marinad kar sakthi hu',
'hello mam aaj humn pehel bar ehi recip banai thi realli mam awsm n yummi',
'meri maa ne kaha ki tel mat kharab kar gst ke baad mehga ho gaya hai',
'gajab ek bani',
'favorit perfect',
'cook use pressur cooker without whistl chicken cook',
'wow kabita mam tasti',
'leav note appreci biryani recip big hit home much stop order biryani look forward recip',
'hello kabita best job recip perfect good keep go best luck',
'nice',
'jab itni der tak paka rahe tab oil nahi pani kuch vi nahi dya chicken aur rice k nich masala aur chicken jal nahi jayega pleas repli',
'mam ye biryani jo apn banaya hain ye kitn logo keliy hain',
'made biryani yesterday turn like ur actual masala chicken lot co like gravi use less dahi rice p l tell',
'wow',
'mei hostel rhta hu waha fridg ni h aisa jruri h kya k chicken ko marin krne k bad usko fridg rkhn a jruri h plz let know',
'perfect',
'hiii mam meri biryani niches jal gayi thi kya kare mean kya krne ka',
'tri tonight websit link forgotten add salt chicken marin guess rectifi',
'mashaallah bhot umda',
'kabila ji great recip thank',
'hello kabita mam prepar shahi biryani masala powder get outsid',
'thaq kabitha ji',
'salaam kabeta ne apko subscrib kyia thank much',
'bht hi zda tast bani thi bryni mai bi ap jaisa dikhay waisa hi banayi thi mere hsbnd ko bht zda psnd ayi bht khush huw mere hsbnd thnk u much di',
'kabita mam kya mein chicken masala garam masala shahi biryani masala jeera powder coriand powder nutmeg powder ke saath black pepper powder add kar sakth hain kyunki mujh best hyderabadi style chicken dum biryani ka flavour chahiy pleas mam repli',
'meri biryani hamesa nich se jaljata hai kyu plz repli',
'chiken mai dahi daln se kya hota hai aunti',
'nice voic',
'best mam million view still count keep mam',
'chan',
'want tri di',
'ghee kon sa deshi',
'hi kabita nice recip pleas tell brand kadai use fri onion like design',
'kabita mam import keep tava keep stove top low flame plz repli',
'mam zruri ni hy na k shahi biryani masala dalna agar na ho',
'st time abl make yummi chicken dum biryani earlier made sever unsucccess attempt thank much',
'nice',
'thanxx mam mei tri krugi',
'mam kesar compulsori hai kya',
'made biryani awesom everyon home like realli much thank much keep post',
'nice',
'nice',
'horribl tri bt hmko pasnd nhi ayi chicken ws cook get cook oil dont like apki saro recip bshut ac chi hoti h bt yeh wali tri krne k bad feel k soch smjh k banay',
'plz repli',
'hmm biryani look yummi',
'fb par ka name ha',
'kavita mam agar rice n chicken ki quantiti zada le dum k lie time kitna dena hoga',
'mujh aap ki recip achi lagi hai fb par kya naam hai',
'kabita mam marinad karth waqt kitna spoon salt add karna hain',
'nice',
'hi',
'tri biryani recip kabita ji realli appreci hard work',
'nice',
'di ye chicken biyani bhagol bana sakt h plz give an',
'mam hyderabadi recip mein thoda sa lime juic dalth hain tho hyderabadi flavour mein thoda sa lime juic tho dal sakthi hu mam bataiy na pleas ki kitna spoon lime juic dal sakthi hu kindli repl i bahut import doubt hain kabita mam',
'thanku mam aap good ho',
'mam spoon ghee matlab tblspn tsp',
'kabita mam put bay leaf marinad',
'tho marinad karth waqt chicken ginger garlicpast pudina coriand leav black elichi green elichi c

innamon clove javitri curd green chilli salt naldi chilli powder garam masala coriand powder jeera powder shahi biryani masala tsp nutmeg powder ke saath spoon ghee aur onion dalthi hu',
'kabita mam mein kitna spoon ghee dalu marin karth waqt aur kitn onion dalu marin karth waqt',
'r u sure chicken paak jayega ussay rice ki tarha pehlay thoda pakkanay ki jarurat toh nahi hai',
'kabita mam superb recip clear video clearli explain origin hyderabadi biryani recip',
'mam mein kitna spoon nutmeg powder add karu pleas bataiy',
'nic',
'mam kya hum thoda sa nutmeg powder dal sakth hain matlab apka opinion kya hain matlab coriand powder garam masala powder aur jeera powder ke saath acha lagega ya tast spoil hoga apka opinion kya hain',
'biryani masala kais bnaat hai market kbhi mila nhi',
'hi mme kind water u use make dough cold warm',
'hello mme r u nice briani ask u someth u mind u put water cook stick pan without water',
'apn video bht hi detail se sab btaya hai',
'yummi',
'super mam',
'wow great recip thank video',
'kilo biryani ke liy masala kitna lage ga',
'mam tho mein coriand powder jeera powder garam masala powder sab tsp dalu pleas bataiy',
'mam aur jyada matlab tsp chalega mam apka opinion kya kitn spoon dalu pleas bataiy',
'mam apn ghee tblspn use kiya har layer mein tho mein tsp use karu matlab apn bola na ki aada qantiti use kijiyo tho tsp dalu',
'mam mein patanjali cow ghee use kar sakthi hu',
'mam kya hum cow ka ghee ghar per bana sakth hain kyunki bahar ka qualiti acha nahin hain',
'seal karn cooker blast hoga kon zimmedaar hoga',
'tri right wish luck',
'dewsi ghee mean cow ghee',
'doodh mein kesar full night tk bhigok rkhn hai yaa instant',
'make shahi biryaani masala',
'kouc test nai hai wat video',
'kya ghee import hain',
'perfect receip',
'waaaaw tasti biryani myney aaj banaya hey tast yummmmm',
'sorri comment',
'asi chicken kachi rhygi',
'apni chicken pakya e ni',
'chicken andar kaccha nehi raheha',
'good',
'kounsa chawal achha rahega matlab long grain rice ka name jo aap',
'awesom recip',
'mujh aap ki help chahiyo mam mujh yeh biriyani person k liy banana hai pl batayeng sare ingredi k a kitna kitna measur rahega iss dushera guest aayeng ghar pe plss bataiyeh k kais yeh jyada achha banega',
'kewra water aur rose water add karn ke baadh mein saffron milk aur orang colour add kar sakthi hu pleas bataiy kabita mam kyunki kewra water aur colour ka chemic reaction tho nahin hoga',
'aur hame kitna spoon coriand powder aur jeera powder aur garam masala powder add karna hain',
'kya hum normal rice use kar sakt mam pleas bataiya kyunki hame apk recip ko dekhkar sabs best ch ristma dish banana hain',
'kya mein coriand powder jeera powder ke saath garam masala bhi add kar sakthi hu marinad karth waqt pleas bataiy kabita mam',
'like cook',
'thank god subtitl look pretti easi thank',
'hi mam saw recip channel name dark moon entertain thought tell u',
'ooh ok',
'aap non veg dish mat banaya kro veg veg jayada tasti hota hai',
'kya mein dhania powder aur jeera powder marinad karn ke liya use kar sakt hun ya biryani ka tast spoil hoga',
'awesom biryani mam',
'great resipi',
'bakwash',
'nice',
'iam done enjoy',
'thank kavita share di yummi recip made today n made delici thank',
'nice',
'bt great video',
'mujh sirf khaan maza aata hai',
'agar pudina patta na ho uski jagah kya use karna hai kyuki humar yaha ye easili market may nai milta hai plz advic',
'first time ever cook biryani treat mom follow everyth said n cook best biryani ever tast thank perfect recepi instruct',
'competit tast ye biryani achi hai ya appki pressur cooker wali achi hai plz repli fast muje sund ay ko banani hai',
'kya badi elichi import hain kyunki yaha shop mein avail nahi hain',
'kabita love biriyani thank u idea',
'hello kavita mam tri recip follow ur step step instruct biriyani turn awesom thank wonder recip detail',
'woow maam need thank lot ye bahot asan lg tha ab definit tri',
'..'

'wow tasti',
'reali awesom testi tha biriyani thank lot mam',
'osmmmm',
'definit tri',
'wow nice n ur explan made easi maam need fri chicken directli put marin chicken',
'thank madam',
'nice',
'love cook find best chef youtub trust u r best chef among chef',
'super duper',
'thank ji',
'hn saari biryaani jala di dikh rha hai vo nikalt time',
'madam use plastic stainer plastic tool hot food',
'use doubl dose saffron skip orang color also stand fresh coriand leav make sick tast like stink bug use parsley',
'kabita di aapki bhut bdi fan hu aaps milna chahti hu kolkata v kolkata se hi hu di',
'nice',
'aap koun si chawal use kartehain mam',
'main tri kiya tha',
'thxz kavita mam im frm punjan india tri definetli tri thxz fr video',
'hi kavita today tri ur biryani recip realli yummi thank',
'nice yummi',
'normal flame chicken kais paka madam tumhara bet chicken paka hi nahi hogaa tumhara',
'ok ty mam',
'like dish',
'cook biryani sinc long see recepi tri unfortun chicken bottom layer get burnt stick step still a bl come issu could pleas suggest exactli may go wrong',
'nice recip mam tasti',
'compani basmati rice use',
'thank actual show recip dum style peopl mix rice chicken fulli cook call dum biryani',
'tri soon',
'ghee se zyada moistur hogya tha oil daln se kam hota h kya',
'freez chicken ni rkhe kya kr skte h usk jgh plz say',
'ok kavita ji thank much',
'kuch samajh nahi aya',
'khala ap ne aloo tu dalay hi ni',
'kavita ji main saudi main hu ye sari cheez main kaha se lao jo aap ne batayi hai',
'thank u mam',
'chhota bada chamach ka size hous hous differ karta h plz aap quantiti jab batat h tab spoon size ko dikha digiy video',
'mujh khana hai ye biryani par mujh banan nahi aata',
'profi thank inform',
'marin chicken night keep fridg',
'mam biriyani hi hoto iski serv kiyni hong',
'much quantiti want make one person',
'tri son ask everi sunday',
'made tast awsm gr easi recipi',
'color mandatori',
'hi kabita tri recip today turn awesom everyon famili like thank',
'vri nice',
'pleas make video zam zam pulao search tasti zam zam pulao recip mani day get suitabl recip wait zam zam pulao recip mam',
'hous veryyyyyyyyyyyyyyyyyyyyyyyyy beauti',
'kg ka bane se itna hi time ho ga',
'agr ism potato bhi add krna ho usko chicken ke saath marin krke daleng plz btay',
'last part dough cap od like liter u need part',
'chiii mene tri ki ghr pr biryani ki naam ki insult h ye plzz guy koi tri mt krna ghr pr',
'cook today chicken dum biryani cook fast time awesom think help wapsid',
'plz mem shahi paneer daal makhni bna sakt ho kya',
'mam ham kaleji deal sakt h',
'also tri recip nice',
'meri birayani achhi bani thank kabita nice recip',
'main abhi bana rahi hun',
'cook today',
'eg n yummi rcp mam thank',
'tasti like',
'mam want tri chicken kchcha ni rhega ya koi smell ni aaegi plz tell',
'thank madam ji',
'mashaallah',
'use coriand powder biryani requir',
'aapn jo rd stage main layer main kya mujh ghee ya oil dalna hain chicken daaln se pehl apn nahi daala im confus pleas help',
'hey tri super se upar thanx ton keep shine',
'good video sorri say hyderabadi chicken biryani',
'javtre',
'actual ek doubt yahan pe chicken aches pakta nhi hoga kyun ki marin k bad direct rice dal rahe h o app u pleas explain mam',
'nice real video',
..

```

'yu r good mam apki recepi ko explain krne k way bhot acha hai',
'nice tasti recepi',
'dish awsom',
'mene aaj banai awesom realli tasti',
'nice',
'sorri sweet',
'good',
'look yakhni biryani recip find',
'like biriyani',
'bina pani ka jal jayega',
'yummmmmi',
'ok biriyani ka jo atar mitha atar hota hai usko ap nahi diya jism se bgiriyani ka orgin cent ate
hai',
'agar fridg na ho kese banayeng',
'kukar ka istem kr skte hai kya',
'mam curd sour lena hai pleas reapli an mam',
'tooooo yummmmmmmmmmyyyyyyyyyy',
'awsom',
'aap ne oil tho dala hi nahi',
'lollllll yeh biryani hai ya sabji ese todhi na banti hai pagal biryani',
'watch middl night mcdonald option guess',
'kesar necessari food colour',
'ingredi use high compar briyani made good',
'mast h mam tri today',
'nice one',
'v nice',
'main aaj tri ki bohot tasti bani thi thank alot',
'main tri kiya ye last friday ko n delici n awsm',
'kabita ji aapn khosboo k liy kuch use nahi kiya',
'kabita ji want tri pleas aap mujh batayeng chawal long aur fluffi kais banega main jitna bhi hig
h qualiti ka basmati rice liya magar wo restaur jaisa long fluffi nahi banta ha pleas repli',
'thankyou kabita kitchen',
'kg chicken ke liy aur kitni quantiti masal add karna padega aur rice k bhi kitni quantiti badani
padegi',
'easi process mam thank',
'owsom',
'nice biryani',
'maza aagaya kasam se sanjita',
'surelli tri shraavan',
'mam boneless chicken use karna ho kitni der pakana hai kyun ki bahut jaldi gal basra hai',
'fridg vegetarian',
'hello kabita mam ghee vanaspati pure ghee pleas repli quickli',
'love kabita aunti kitchen',
'muze aisi biwi chahiy jo sirf khana achcha banan wali ho baki samhal lunga',
'mam pleas kg biryani ke liy video banao',
'mam chahwal ko bhogona adha ghntta ya sirf ubalna',
'hai',
'help thank recip cook today yummi',
'nice mam thank u',
'u make khichda plz',
'nice',
'mam ba hr marin krke nd fir ba dum pe chadha k chicken pakk jayega kachha reh gya',
'vo jo chicken k badhey piec h vo jalt ni h proper cook ho jate h',
'mam chicken biryani tast coriand biryani take risk recip hydrabadi',
'kabita ji kya dahi dalna zrori',
'javitri matlab',
'ohhhhhhhhhhhhhhh nice',
'aap cook bahut acchi karti hai aap apna naam batay kyuki mujh aaps biryi sikni hai',
'bahut bariki se aapn batya hai mam',
'hi kabita today tri chicken biryani recipi first time life cook recipi tell absolut fantast shoc
j wish could tast thank wonder recipi tc gd bless',
'osm',
'thank lot recip never knew first tri would success',
'first briyani trust awesom big thank kabita kitchen',
'jst cook biryani trust guy realli good jst follow process step step told u end super tasti lip s
mack biryani kabita didi ur best thank u much',
...]
```

APPLYING THE FOLLOWING PARAMETRIC AND NON PARAMETRIC MACHINE LEARNING MODELS

LIST OF MACHINE LEARNING MODELS APPLIED

1)GAUSSIAN NAIVE BAYES (PARAMETRIC)

2)BERNOULLI NAIVE BAYES (PARAMETRIC)

3)SUPPORT VECTOR MACHINE WITH LINEAR KERNAL (PARAMETRIC)

4)SUPPORT VECTOR MACHINE WITH POLYNOMIAL KERNEL (PARAMETRIC)

5)SUPPORT VECTOR MACHINE WITH GAUSSIAN KERNEL (PARAMETRIC)

6)RANDOM FOREST (NON PARAMETRIC)

7)DECISION TREE (NON PARAMETRIC)

8)LOGESTIC REGRESSION (PARAMETRIC)

GAUSSIAN NAIVE BAYES MODEL USING COUNT VECTORIZER (KB)

In [114]:

```
# Splitting the dataset into the Training Set and Testing Set
from sklearn.model_selection import train_test_split
X_train_cv1, X_test_cv1, y_train_cv1, y_test_cv1 = train_test_split(Xk_c,yk_c,test_size = 0.2,
                                                                    random_state = 0)
```

In [115]:

```
# Training the Naive Bayes model on the Training set
from sklearn.naive_bayes import GaussianNB
nbk = GaussianNB()
nbk.fit(X_train_cv1, y_train_cv1)
```

Out[115]:

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [116]:

```
#cross validation
from sklearn.model_selection import cross_val_score
cross_nb_cv = cross_val_score(estimator=nbk, X=X_train_cv1, y=y_train_cv1, cv=10)
cross_nb_cv
```

Out[116]:

```
array([0.43367347, 0.46938776, 0.52806122, 0.44897959, 0.46173469,
       0.42091837, 0.44897959, 0.43367347, 0.4744898 , 0.48214286])
```

In [117]:

```
#prediction test
nb1_pred_cv = nbk.predict(X_test_cv1)
nb1_pred_cv
```

Out[117]:

```
array([3, 7, 3, 4, 4, 3, 2, 2, 2, 4, 7, 4, 4, 3, 1, 3, 2, 4, 1, 6, 1, 7,
       6, 5, 2, 3, 1, 3, 1, 1, 1, 5, 4, 2, 7, 3, 3, 6, 3, 1, 5, 7, 5, 6,
       1, 1, 2, 4, 7, 1, 7, 2, 5, 3, 2, 1, 3, 1, 3, 4, 1, 5, 1, 7, 6, 2,
       2, 1, 4, 2, 2, 2, 3, 2, 2, 7, 6, 3, 2, 1, 1, 3, 3, 2, 4, 3, 3, 6,
       6, 3, 4, 3, 4, 2, 4, 7, 4, 1, 4, 5, 6, 7, 2, 5, 2, 2, 5, 5, 6, 2,
       2, 2, 7, 5, 1, 4, 3, 4, 1, 4, 5, 3, 7, 1, 2, 4, 3, 5, 2, 7, 7, 1,
       5, 7, 6, 5, 2, 6, 2, 3, 4, 3, 3, 1, 1, 5, 4, 5, 6, 1, 1, 2, 1, 2,
       6, 7, 2, 5, 5, 5, 1, 2, 1, 2, 2, 5, 1, 3, 1, 6, 3, 2, 7, 1, 6, 4,
```

```

2, 3, 6, 3, 2, 6, 2, 7, 1, 6, 5, 4, 7, 6, 7, 2, 3, 2, 1, 6, 1, 1,
2, 1, 1, 7, 7, 7, 1, 1, 2, 7, 2, 2, 1, 7, 5, 7, 4, 1, 3, 2, 7, 5,
3, 3, 1, 2, 1, 6, 1, 7, 2, 1, 2, 2, 7, 3, 5, 7, 1, 1, 3, 1, 3, 1,
4, 1, 3, 6, 1, 1, 3, 2, 2, 2, 3, 4, 2, 2, 2, 2, 1, 3, 6, 3, 3, 3,
1, 3, 7, 2, 6, 1, 1, 2, 4, 1, 1, 5, 7, 2, 5, 2, 1, 2, 1, 2, 3, 2,
3, 6, 4, 5, 3, 2, 3, 6, 4, 5, 3, 4, 6, 4, 4, 1, 4, 2, 1, 2, 6, 3,
6, 3, 7, 2, 3, 2, 5, 3, 2, 5, 2, 2, 7, 3, 1, 5, 5, 4, 1, 2, 7,
4, 2, 3, 5, 6, 7, 3, 1, 5, 3, 6, 2, 3, 7, 1, 1, 5, 1, 7, 5, 2, 5,
1, 1, 3, 2, 5, 2, 3, 2, 7, 2, 1, 3, 7, 2, 4, 7, 2, 4, 6, 2, 2, 6,
7, 1, 6, 2, 3, 2, 3, 1, 7, 5, 7, 6, 4, 5, 3, 3, 3, 4, 2, 3, 1, 1,
1, 6, 3, 7, 5, 7, 3, 6, 5, 3, 2, 2, 5, 2, 7, 5, 1, 2, 6, 1, 7, 7,
4, 3, 3, 1, 1, 4, 1, 2, 7, 5, 4, 3, 2, 3, 1, 6, 2, 5, 5, 2, 7, 4,
5, 3, 2, 2, 3, 2, 3, 2, 6, 4, 3, 3, 3, 2, 7, 2, 3, 7, 3, 1, 3, 7,
3, 1, 3, 2, 4, 3, 1, 7, 3, 1, 2, 1, 7, 4, 5, 2, 2, 2, 3, 4, 2, 6,
4, 1, 1, 7, 3, 1, 5, 5, 2, 1, 4, 2, 2, 5, 3, 2, 3, 5, 1, 2, 7, 1,
2, 1, 5, 3, 2, 2, 1, 3, 7, 4, 6, 7, 5, 2, 5, 2, 7, 3, 7, 4, 7, 3,
3, 1, 1, 1, 5, 7, 5, 6, 3, 3, 1, 3, 7, 6, 1, 2, 1, 2, 2, 6, 3, 3,
7, 2, 7, 3, 2, 3, 5, 3, 2, 2, 1, 1, 3, 2, 3, 1, 7, 7, 5, 2, 3, 6,
3, 3, 3, 3, 3, 3, 3, 1, 2, 2, 1, 6, 4, 2, 2, 3, 7, 2, 7, 7, 4, 3,
2, 1, 2, 3, 5, 7, 6, 2, 2, 1, 3, 2, 2, 7, 2, 1, 2, 4, 3, 5, 4, 7,
1, 4, 3, 7, 3, 6, 3, 7, 3, 2, 2, 1, 3, 1, 2, 4, 1, 1, 1, 3, 5, 3,
2, 4, 2, 3, 1, 1, 3, 1, 5, 3, 7, 3, 7, 6, 3, 7, 3, 2, 2, 5, 1, 2,
3, 5, 3, 5, 1, 3, 3, 1, 3, 5, 1, 2, 1, 6, 7, 5, 2, 6, 7, 2, 1, 3,
4, 2, 2, 1, 5, 2, 7, 1, 3, 3, 1, 2, 5, 5, 7, 5, 4, 3, 1, 3, 3, 1,
3, 7, 3, 1, 5, 7, 3, 5, 2, 3, 3, 1, 1, 5, 7, 1, 3, 4, 1, 5, 2, 2,
2, 2, 2, 1, 2, 1, 1, 3, 5, 5, 2, 7, 4, 2, 3, 5, 3, 4, 3, 4, 6, 1,
2, 7, 7, 4, 1, 5, 7, 4, 1, 1, 3, 1, 2, 3, 4, 7, 3, 3, 3, 4, 2, 5,
2, 4, 2, 2, 3, 5, 1, 2, 4, 3, 3, 5, 2, 5, 5, 3, 4, 4, 2, 4, 5, 7,
2, 3, 7, 2, 1, 6, 3, 1, 4, 3, 5, 2, 2, 1, 5, 5, 2, 1, 1, 3, 6, 7,
3, 3, 2, 4, 6, 3, 3, 2, 3, 1, 1, 2, 3, 3, 7, 6, 2, 4, 5, 1, 2, 5,
5, 3, 3, 3, 2, 5, 3, 3, 1, 3, 5, 3, 2, 3, 2, 5, 6, 5, 5, 1, 2, 1,
1, 1, 1, 5, 1, 3, 2, 3, 1, 4, 4, 3, 1, 3, 7, 3, 3, 1, 1, 5, 2, 2,
7, 4, 4, 2, 5, 3, 5, 3, 3, 6, 7, 6, 5, 2, 1, 2, 1, 4, 5, 7, 1,
7, 2, 1, 2, 2, 2, 1, 3, 7, 3, 3, 3, 1, 2, 2, 7, 5, 2, 7, 1, 3, 1,
2, 5, 4, 2, 2, 7, 6, 3, 1, 4, 3, 4, 5, 1, 3, 2, 2, 2, 6, 2, 5, 6,
4, 3, 1, 1, 7, 3, 3, 5, 3, 5, 3, 3, 6, 5, 1, 3, 4, 1, 7, 2, 2, 4,
3, 1, 3, 3, 1, 3, 6, 5, 4, 3, 3, 2])

```

VALIDATION OF NAIVE BAYES MODEL

In [118]:

```

from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import f1_score
from sklearn import metrics
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
from sklearn.metrics import matthews_corrcoef

```

In [119]:

```

print("Naive bayes using count vectorizer Accuracy is:",metrics.accuracy_score(y_test_cv1,
nb1_pred_cv))
print("Naive bayes using count vectorizer f1 score is :",f1_score(y_test_cv1, nb1_pred_cv, average=
'macro'))
print("Naive bayes using count vectorizer recall score is :",recall_score(y_test_cv1, nb1_pred_cv,
average='macro'))
print("Naive bayes using count vectorizer precison score is :",precision_score(y_test_cv1,
nb1_pred_cv, average='macro'))
print("Naive bayes using count vectorizer precison score is :",precision_score(y_test_cv1,
nb1_pred_cv, average='macro'))
print("Naive bayes using count vectorizer matthews_corrcoef score is
:",matthews_corrcoef(y_test_cv1, nb1_pred_cv))

```

```

Naive bayes using count vectorizer Accuracy is: 0.4306122448979592
Naive bayes using count vectorizer f1 score is : 0.41940505171491654
Naive bayes using count vectorizer recall score is : 0.4369165427039314
Naive bayes using count vectorizer precison score is : 0.4347140190873368
Naive bayes using count vectorizer precison score is : 0.4347140190873368
Naive bayes using count vectorizer matthews_corrcoef score is : 0.33879634536365666

```

GAUSSIAN NAIVE BAYES MODEL USING TFIDF VECTORIZER (KB)

In [120]:

```
#creating tfidf vectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
tf_k = TfidfVectorizer()
```

```
Xk_tf = tf_k.fit_transform(corpus).toarray()
yk_tf = dataset_kb.iloc[:,2].values
print(Xk_tf.shape)
```

(4900, 4483)

In [9]:

```
#creating TFIDF vectorizer features
from sklearn.model_selection import train_test_split
X_train_tf1, X_test_tf1, y_train_tf1, y_test_tf1 = train_test_split(Xk_tf,yk_tf,test_size = 0.2,
                                                                    random_state = 0)
```

In [121]:

```
nbk.fit(X_train_tf1, y_train_tf1)
```

Out[121]:

GaussianNB(priors=None, var_smoothing=1e-09)

In [52]:

```
#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_nb_tf = cross_val_score(estimator=nbk, X=X_train_tf1, y=y_train_tf1, cv=10)
cross_nb_tf
```

Out[52]:

```
array([0.46938776, 0.51020408, 0.5127551 , 0.48214286, 0.52295918,
       0.43622449, 0.49489796, 0.46683673, 0.48979592, 0.5       ])
```

In [122]:

```
#prediction test
nb1_pred_tf = nbk.predict(X_test_tf1)
nb1_pred_tf
```

Out[122]:

```
array([3, 7, 3, 4, 4, 3, 2, 2, 2, 4, 7, 4, 4, 2, 1, 3, 2, 4, 1, 6, 1, 7,
       7, 5, 2, 2, 1, 3, 1, 1, 1, 5, 4, 2, 7, 3, 3, 6, 3, 1, 5, 7, 5, 6,
       1, 1, 2, 4, 7, 1, 7, 2, 6, 4, 2, 1, 3, 1, 3, 4, 1, 5, 7, 7, 6, 2,
       2, 1, 4, 2, 2, 2, 3, 2, 2, 7, 6, 3, 2, 1, 1, 3, 3, 2, 4, 3, 3, 7,
       7, 3, 4, 2, 4, 2, 4, 7, 4, 1, 4, 5, 6, 7, 2, 5, 2, 2, 5, 5, 7, 2,
       2, 2, 7, 5, 1, 4, 3, 4, 1, 4, 5, 3, 7, 4, 2, 4, 3, 5, 2, 7, 7, 1,
       6, 6, 7, 5, 2, 6, 2, 3, 4, 3, 3, 1, 1, 5, 4, 5, 6, 1, 1, 2, 4, 2,
       6, 7, 2, 5, 5, 5, 7, 2, 1, 2, 2, 5, 1, 3, 1, 6, 3, 2, 7, 1, 6, 4,
       2, 3, 6, 3, 2, 6, 2, 7, 1, 6, 5, 4, 7, 6, 7, 2, 3, 2, 6, 7, 4, 1,
       2, 5, 1, 7, 7, 7, 1, 1, 2, 7, 2, 2, 1, 7, 6, 7, 6, 1, 3, 2, 7, 5,
       3, 2, 1, 2, 1, 6, 1, 7, 2, 1, 2, 2, 7, 3, 5, 7, 1, 1, 3, 1, 3, 1,
       4, 1, 3, 6, 1, 1, 3, 2, 2, 2, 3, 4, 2, 2, 2, 2, 1, 3, 7, 3, 3, 3,
       1, 3, 7, 2, 7, 1, 1, 2, 6, 5, 1, 5, 7, 2, 5, 2, 1, 2, 1, 2, 3, 2,
       3, 6, 4, 5, 3, 2, 3, 6, 4, 5, 2, 4, 6, 4, 4, 1, 4, 2, 4, 2, 6, 3,
       6, 3, 7, 2, 3, 2, 5, 3, 2, 5, 2, 2, 2, 7, 3, 1, 5, 5, 4, 1, 2, 7,
       4, 2, 3, 5, 6, 7, 3, 1, 5, 3, 6, 2, 3, 7, 1, 1, 5, 1, 7, 5, 2, 5,
       1, 1, 3, 2, 5, 2, 3, 2, 7, 2, 1, 3, 7, 2, 4, 5, 2, 4, 6, 2, 2, 6,
       7, 1, 6, 2, 3, 2, 2, 1, 7, 5, 7, 6, 4, 5, 3, 2, 1, 4, 2, 3, 4, 1,
       4, 6, 3, 6, 5, 7, 3, 6, 5, 3, 2, 2, 5, 2, 7, 5, 1, 2, 6, 1, 7, 7,
       4, 2, 3, 1, 1, 4, 1, 4, 7, 5, 4, 2, 2, 3, 6, 6, 2, 5, 5, 2, 7, 4,
```

```

5, 2, 2, 2, 2, 6, 3, 2, 6, 4, 3, 3, 3, 2, 7, 2, 3, 7, 3, 5, 3, 7,
3, 1, 3, 2, 4, 3, 1, 7, 3, 1, 2, 1, 7, 4, 5, 2, 2, 2, 2, 4, 2, 6,
4, 1, 1, 7, 3, 1, 5, 5, 2, 5, 4, 2, 2, 5, 3, 2, 2, 6, 6, 2, 7, 1,
2, 1, 5, 3, 2, 2, 1, 3, 7, 4, 6, 6, 5, 2, 5, 2, 7, 3, 7, 4, 7, 3,
3, 1, 1, 1, 5, 7, 5, 6, 3, 3, 1, 2, 7, 6, 1, 2, 1, 2, 2, 7, 3, 3,
7, 2, 7, 3, 2, 3, 5, 3, 2, 2, 1, 1, 2, 2, 3, 1, 7, 7, 5, 2, 3, 7,
3, 3, 3, 3, 3, 3, 3, 1, 2, 2, 1, 7, 4, 2, 2, 3, 7, 2, 7, 7, 4, 2,
2, 1, 2, 3, 5, 7, 7, 2, 2, 1, 3, 2, 2, 7, 2, 1, 2, 4, 3, 5, 5, 7,
1, 4, 3, 7, 3, 6, 2, 7, 3, 2, 2, 1, 3, 1, 2, 7, 6, 6, 1, 3, 5, 3,
2, 4, 2, 2, 1, 1, 3, 1, 5, 3, 7, 3, 7, 6, 3, 7, 3, 2, 2, 5, 1, 2,
3, 5, 3, 5, 1, 3, 3, 4, 3, 5, 1, 2, 1, 6, 7, 5, 2, 6, 7, 2, 1, 2,
6, 2, 2, 1, 6, 2, 7, 5, 3, 3, 1, 2, 5, 5, 7, 5, 5, 2, 6, 3, 3, 1,
3, 7, 3, 4, 5, 7, 3, 5, 2, 3, 2, 1, 7, 5, 7, 1, 3, 4, 1, 5, 2, 2,
2, 2, 2, 1, 2, 1, 1, 3, 4, 5, 2, 7, 4, 2, 3, 5, 3, 4, 3, 4, 6, 1,
2, 7, 7, 4, 1, 5, 7, 4, 1, 1, 2, 1, 2, 3, 4, 7, 3, 3, 2, 4, 2, 5,
2, 4, 2, 2, 3, 5, 1, 2, 4, 3, 3, 5, 2, 6, 5, 3, 4, 4, 2, 4, 6, 6,
2, 3, 7, 2, 1, 6, 3, 4, 4, 3, 5, 2, 2, 1, 5, 5, 2, 4, 1, 3, 6, 7,
3, 2, 2, 4, 6, 1, 3, 2, 3, 1, 1, 2, 3, 2, 7, 6, 2, 4, 5, 1, 2, 5,
5, 3, 3, 2, 2, 6, 3, 3, 1, 3, 7, 3, 2, 3, 1, 5, 7, 5, 5, 1, 2, 4,
1, 1, 1, 5, 1, 3, 2, 3, 1, 4, 4, 3, 1, 3, 7, 3, 3, 1, 1, 5, 2, 7,
7, 4, 5, 2, 5, 3, 5, 3, 3, 6, 7, 6, 5, 2, 1, 2, 1, 4, 5, 7, 1,
7, 2, 1, 2, 2, 2, 1, 3, 7, 3, 3, 3, 1, 2, 2, 7, 5, 2, 7, 1, 3, 1,
6, 5, 4, 2, 2, 7, 6, 3, 1, 6, 3, 4, 5, 1, 3, 2, 2, 2, 6, 2, 5, 6,
4, 3, 1, 1, 7, 3, 3, 5, 3, 5, 3, 3, 6, 5, 1, 3, 7, 1, 7, 2, 2, 4,
3, 1, 3, 2, 1, 3, 6, 4, 4, 3, 3, 2])

```

In [54]:

```

#validation test

print("Naive bayes using tfidf vectorizer Accuracy is:",metrics.accuracy_score(y_test_tfl,
nbl_pred_tf))
print("Naive bayes using tfidf vectorizer f1 score is :",f1_score(y_test_tfl, nbl_pred_tf, average=
'macro'))
print("Naive bayes recall score is :",recall_score(y_test_tfl, nbl_pred_tf, average='macro'))
print("Naive bayes precision score is :",precision_score(y_test_tfl, nbl_pred_tf, average='macro'))
print("Naive bayes precision score is :",precision_score(y_test_tfl, nbl_pred_tf, average='macro'))
print("Naive bayes matthews_corrcoef score is :",matthews_corrcoef(y_test_tfl, nbl_pred_tf))

```

```

Naive bayes Accuracy is: 0.4561224489795918
Naive bayes f1 score is : 0.4441483505989759
Naive bayes recall score is : 0.4615437682735867
Naive bayes precision score is : 0.44924378051305486
Naive bayes precision score is : 0.44924378051305486
Naive bayes matthews_corrcoef score is : 0.3676340252405466

```

Bernoulli NAIVE BAYES USING COUNT VECTORIZER (KB)

In [7]:

```

#fitting bernoulli naive bayes algorithm with count vectorizer
from sklearn.naive_bayes import BernoulliNB
bnk=BernoulliNB()
bnk.fit(X_train_cv1, y_train_cv1)

```

Out[7]:

```

BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)

```

In [66]:

```

#cross validation test on training set
cross_bnk_cv = cross_val_score(estimator=bnk, X=X_train_cv1, y=y_train_cv1, cv=10)
cross_bnk_cv

```

Out[66]:

```

array([0.64540816, 0.66581633, 0.61989796, 0.65306122, 0.66581633,
       0.68622449, 0.65816327, 0.64795918, 0.64030612, 0.62244898])

```

In [57]:

```
#prediction test
bnk1_pred_cv = bnk.predict(X_test_cv1)
bnk1_pred_cv
```

Out[57]:

```
array([3, 7, 3, 5, 3, 2, 2, 1, 3, 4, 7, 4, 3, 2, 2, 3, 5, 3, 2, 4, 3, 7,
       7, 5, 1, 2, 1, 3, 4, 1, 1, 2, 4, 2, 7, 3, 3, 3, 2, 1, 5, 7, 6, 3,
       1, 2, 4, 4, 7, 3, 3, 1, 6, 3, 5, 7, 2, 1, 3, 4, 4, 4, 6, 1, 4, 1,
       2, 2, 4, 3, 2, 2, 3, 2, 2, 7, 2, 2, 2, 7, 2, 3, 3, 2, 4, 3, 3, 3,
       4, 3, 1, 2, 6, 2, 2, 6, 5, 5, 4, 7, 3, 7, 3, 7, 2, 3, 4, 5, 6, 2,
       5, 4, 7, 4, 2, 2, 3, 4, 5, 7, 5, 3, 5, 1, 2, 4, 2, 7, 5, 7, 5, 2,
       1, 7, 6, 2, 1, 2, 3, 3, 2, 3, 3, 1, 1, 4, 4, 6, 2, 2, 5, 1, 4, 2,
       3, 7, 2, 3, 5, 6, 4, 2, 3, 5, 3, 7, 1, 3, 6, 6, 3, 5, 7, 1, 6, 4,
       2, 3, 2, 3, 2, 3, 1, 7, 4, 6, 2, 4, 2, 4, 3, 1, 3, 2, 1, 7, 3, 2,
       2, 2, 5, 1, 7, 7, 4, 3, 3, 7, 2, 2, 1, 7, 1, 7, 4, 3, 3, 2, 2, 4,
       3, 2, 5, 2, 1, 6, 1, 7, 2, 1, 2, 1, 2, 3, 1, 7, 3, 4, 3, 3, 3, 1,
       3, 2, 2, 7, 5, 3, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
       1, 3, 7, 1, 6, 3, 2, 1, 4, 5, 5, 5, 7, 3, 3, 2, 6, 2, 2, 4, 3, 3,
       3, 6, 2, 7, 3, 1, 3, 4, 1, 4, 2, 5, 1, 1, 4, 3, 5, 5, 4, 1, 6, 3,
       6, 3, 7, 5, 3, 1, 4, 3, 2, 5, 2, 2, 2, 3, 2, 1, 7, 3, 4, 4, 1, 7,
       4, 1, 3, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 4, 1, 2, 2, 4, 3, 2, 1, 5,
       5, 5, 3, 5, 5, 1, 3, 1, 3, 2, 1, 3, 5, 1, 3, 7, 2, 1, 6, 3, 1, 6,
       6, 1, 6, 1, 2, 4, 2, 1, 7, 4, 7, 2, 4, 7, 3, 2, 3, 4, 3, 3, 4, 5,
       4, 6, 3, 6, 5, 3, 3, 6, 5, 3, 5, 2, 6, 3, 2, 3, 5, 5, 6, 4, 7, 6,
       3, 2, 3, 6, 1, 4, 6, 3, 7, 5, 4, 2, 1, 3, 1, 1, 2, 6, 3, 3, 7, 4,
       3, 2, 5, 2, 2, 1, 3, 2, 6, 7, 3, 2, 3, 2, 7, 1, 3, 7, 3, 3, 3, 7,
       2, 1, 3, 3, 4, 3, 1, 7, 3, 1, 5, 5, 7, 4, 5, 1, 2, 1, 2, 5, 2, 2,
       5, 1, 2, 4, 3, 4, 4, 3, 2, 7, 5, 2, 4, 4, 2, 1, 2, 5, 5, 4, 7, 5,
       1, 2, 3, 3, 1, 2, 2, 3, 7, 3, 2, 1, 2, 1, 4, 1, 7, 3, 3, 4, 7, 3,
       3, 3, 3, 5, 2, 6, 5, 6, 3, 3, 5, 2, 7, 7, 4, 2, 2, 1, 1, 6, 2, 3,
       7, 2, 3, 3, 2, 3, 1, 3, 7, 5, 2, 1, 2, 1, 3, 7, 7, 7, 3, 4, 3, 5,
       3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 3, 4, 1, 2, 3, 3, 2, 7, 7, 4, 2,
       5, 6, 1, 3, 5, 6, 2, 2, 1, 6, 3, 1, 5, 7, 3, 1, 5, 4, 3, 3, 4, 7,
       1, 6, 3, 7, 3, 7, 2, 6, 3, 2, 2, 3, 3, 1, 3, 1, 4, 1, 3, 5, 3,
       3, 4, 1, 2, 1, 5, 3, 1, 5, 3, 6, 3, 7, 7, 3, 7, 3, 2, 1, 7, 3, 1,
       2, 6, 3, 5, 3, 3, 3, 1, 3, 5, 1, 2, 2, 6, 7, 4, 5, 4, 7, 1, 6, 2,
       1, 1, 1, 3, 4, 1, 7, 1, 3, 3, 3, 5, 5, 5, 7, 2, 4, 2, 5, 3, 3, 2,
       3, 5, 3, 4, 1, 7, 3, 5, 2, 3, 2, 6, 4, 5, 7, 1, 3, 3, 3, 4, 5, 1,
       5, 3, 1, 1, 2, 1, 6, 3, 6, 5, 1, 4, 4, 2, 3, 5, 3, 4, 3, 4, 6, 1,
       1, 7, 7, 3, 5, 6, 7, 4, 3, 1, 2, 3, 5, 3, 4, 7, 3, 3, 2, 3, 1, 5,
       2, 4, 2, 5, 3, 4, 3, 5, 2, 3, 3, 5, 2, 2, 7, 3, 4, 4, 4, 4, 5, 7,
       2, 2, 7, 1, 2, 3, 3, 3, 4, 3, 6, 4, 2, 4, 2, 7, 2, 5, 1, 2, 6, 7,
       3, 2, 5, 3, 4, 4, 3, 2, 3, 1, 3, 2, 3, 2, 7, 6, 2, 3, 3, 5, 2, 4,
       5, 3, 3, 2, 5, 5, 2, 3, 3, 3, 7, 3, 1, 3, 5, 7, 2, 5, 5, 6, 1, 1,
       4, 5, 1, 7, 7, 2, 1, 3, 2, 6, 5, 3, 1, 3, 7, 3, 3, 7, 4, 5, 2, 2,
       3, 7, 4, 1, 5, 3, 7, 3, 2, 3, 6, 7, 6, 4, 3, 4, 2, 1, 5, 3, 7, 2,
       7, 1, 3, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 3, 3, 7, 7, 1, 2, 3, 3, 6,
       2, 2, 4, 2, 2, 3, 3, 2, 6, 6, 3, 5, 5, 5, 2, 2, 2, 2, 4, 5, 2, 4,
       4, 3, 5, 1, 7, 3, 3, 5, 3, 5, 3, 3, 6, 6, 1, 3, 3, 3, 2, 2, 3, 4,
       3, 3, 3, 2, 6, 3, 4, 5, 6, 3, 3, 2]])
```

In [59]:

```
#validation test
print("bernouli Naive bayes Accuracy is:",metrics.accuracy_score(y_test_cv1, bnk1_pred_cv))
print("bernouli Naive bayes f1 score is :",f1_score(y_test_cv1, bnk1_pred_cv, average='macro'))
print("bernouli Naive bayes recall score is :",recall_score(y_test_cv1, bnk1_pred_cv,
average='macro'))
print("bernouli Naive bayes precison score is :",precision_score(y_test_cv1, bnk1_pred_cv, average
='macro'))
print("bernouli Naive bayes precison score is :",precision_score(y_test_cv1, bnk1_pred_cv, average
='macro'))
print("bernouli Naive bayes matthews_corrcoef score is :",matthews_corrcoef(y_test_cv1,
bnk1_pred_cv))
```

```
bernouli Naive bayes Accuracy is: 0.6561224489795918
bernouli Naive bayes f1 score is : 0.6454996531177891
bernouli Naive bayes recall score is : 0.6577895009825048
bernouli Naive bayes precision score is : 0.6855335181343837
bernouli Naive bayes precision score is : 0.6855335181343837
bernouli Naive bayes matthews_corrcoef score is : 0.6091506068953487
```

Bernoulli NAIVE BAYES USING TFIDF VECTORIZER

In [71]:

```
#fitting bernouli naive bayes model using TFidf
bnk.fit(X_train_tfl, y_train_tfl)
```

Out[71]:

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

In [72]:

```
#cross validation
cross_bnk_tf = cross_val_score(estimator=bnn, X=X_train_tfl, y=y_train_tfl, cv=10)
cross_bnk_tf
```

Out[72]:

```
array([0.64540816, 0.66581633, 0.61989796, 0.65306122, 0.66581633,
       0.68622449, 0.65816327, 0.64795918, 0.64030612, 0.62244898])
```

In [73]:

```
#prediction test
bnk1_pred_tf = bnk.predict(X_test_tfl)
bnk1_pred_tf
```

Out[73]:

```
array([3, 7, 3, 5, 3, 2, 2, 1, 3, 4, 7, 4, 3, 2, 2, 3, 5, 3, 2, 4, 3, 7,
       7, 5, 1, 2, 1, 3, 4, 1, 1, 2, 4, 2, 7, 3, 3, 3, 2, 1, 5, 7, 6, 3,
       1, 2, 4, 4, 7, 3, 3, 1, 6, 3, 5, 7, 2, 1, 3, 4, 4, 4, 6, 1, 4, 1,
       2, 2, 4, 3, 2, 2, 3, 2, 2, 7, 2, 2, 2, 7, 2, 3, 3, 2, 4, 3, 3, 3,
       4, 3, 1, 2, 6, 2, 2, 6, 5, 5, 4, 7, 3, 7, 3, 7, 2, 3, 4, 5, 6, 2,
       5, 4, 7, 4, 2, 2, 3, 4, 5, 7, 5, 3, 5, 1, 2, 4, 2, 7, 5, 7, 5, 2,
       1, 7, 6, 2, 1, 2, 3, 3, 2, 3, 3, 1, 1, 4, 4, 6, 2, 2, 5, 1, 4, 2,
       3, 7, 2, 3, 5, 6, 4, 2, 3, 5, 3, 7, 1, 3, 6, 6, 3, 5, 7, 1, 6, 4,
       2, 3, 2, 3, 2, 3, 1, 7, 4, 6, 2, 4, 2, 4, 3, 1, 3, 2, 1, 7, 3, 2,
       2, 2, 5, 1, 7, 7, 4, 3, 3, 7, 2, 2, 1, 7, 1, 7, 4, 3, 3, 2, 2, 4,
       3, 2, 5, 2, 1, 6, 1, 7, 2, 1, 2, 1, 2, 3, 1, 7, 3, 4, 3, 3, 3, 1,
       3, 2, 2, 7, 5, 3, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
       1, 3, 7, 1, 6, 3, 2, 1, 4, 5, 5, 5, 7, 3, 3, 2, 6, 2, 2, 4, 3, 3,
       3, 6, 2, 7, 3, 1, 3, 4, 1, 4, 2, 5, 1, 1, 4, 3, 5, 5, 4, 1, 6, 3,
       6, 3, 7, 5, 3, 1, 4, 3, 2, 5, 2, 2, 3, 2, 1, 7, 3, 4, 4, 1, 7,
       4, 1, 3, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 4, 1, 2, 2, 4, 3, 2, 1, 5,
       5, 5, 3, 5, 5, 1, 3, 1, 3, 2, 1, 3, 5, 1, 3, 7, 2, 1, 6, 3, 1, 6,
       6, 1, 6, 1, 2, 4, 2, 1, 7, 4, 7, 2, 4, 7, 3, 2, 3, 4, 3, 3, 4, 5,
       4, 6, 3, 6, 5, 3, 3, 6, 5, 3, 5, 2, 6, 3, 2, 3, 5, 5, 6, 4, 7, 6,
       3, 2, 3, 6, 1, 4, 6, 3, 7, 5, 4, 2, 1, 3, 1, 1, 2, 6, 3, 3, 7, 4,
       3, 2, 5, 2, 2, 1, 3, 2, 6, 7, 3, 2, 3, 2, 7, 1, 3, 7, 3, 3, 3, 7,
       2, 1, 3, 3, 4, 3, 1, 7, 3, 1, 5, 5, 7, 4, 5, 1, 2, 1, 2, 5, 2, 2,
       5, 1, 2, 4, 3, 4, 4, 3, 2, 7, 5, 2, 4, 4, 2, 1, 2, 5, 5, 4, 7, 5,
       1, 2, 3, 3, 1, 2, 2, 3, 7, 3, 2, 1, 2, 1, 4, 1, 7, 3, 3, 4, 7, 3,
       3, 3, 3, 5, 2, 6, 5, 6, 3, 3, 5, 2, 7, 7, 4, 2, 2, 1, 1, 6, 2, 3,
       7, 2, 3, 3, 2, 3, 1, 3, 7, 5, 2, 1, 2, 1, 3, 7, 7, 7, 3, 4, 3, 5,
       3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 3, 4, 1, 2, 3, 3, 2, 7, 7, 4, 2,
       5, 6, 1, 3, 5, 6, 2, 2, 1, 6, 3, 1, 5, 7, 3, 1, 5, 4, 3, 3, 4, 7,
       1, 6, 3, 7, 3, 7, 2, 6, 3, 2, 2, 3, 3, 3, 1, 3, 1, 4, 1, 3, 5, 3,
       3, 4, 1, 2, 1, 5, 3, 1, 5, 3, 6, 3, 7, 7, 3, 7, 3, 2, 1, 7, 3, 1,
       2, 6, 3, 5, 3, 3, 3, 1, 3, 5, 1, 2, 2, 6, 7, 4, 5, 4, 7, 1, 6, 2,
       1, 1, 1, 3, 4, 1, 7, 1, 3, 3, 3, 5, 5, 5, 7, 2, 4, 2, 5, 3, 3, 2,
       3, 5, 3, 4, 1, 7, 3, 5, 2, 3, 2, 6, 4, 5, 7, 1, 3, 3, 3, 4, 5, 1,
       5, 3, 1, 1, 2, 1, 6, 3, 6, 5, 1, 4, 4, 2, 3, 5, 3, 4, 3, 4, 6, 1,
       1, 7, 7, 3, 5, 6, 7, 4, 3, 1, 2, 3, 5, 3, 4, 7, 3, 3, 2, 3, 1, 5,
       2, 4, 2, 5, 3, 4, 3, 5, 2, 3, 3, 5, 2, 2, 7, 3, 4, 4, 4, 4, 5, 7,
       2, 2, 7, 1, 2, 3, 3, 3, 4, 3, 6, 4, 2, 4, 2, 7, 2, 5, 1, 2, 6, 7,
       3, 2, 5, 3, 4, 4, 3, 2, 3, 1, 3, 2, 3, 2, 7, 6, 2, 3, 3, 5, 2, 4,
       5, 3, 3, 2, 5, 5, 2, 3, 3, 3, 7, 3, 1, 3, 5, 7, 2, 5, 5, 6, 1, 1,
       4, 5, 1, 7, 7, 2, 1, 3, 2, 6, 5, 3, 1, 3, 7, 3, 3, 7, 4, 5, 2, 2,
       3, 7, 4, 1, 5, 3, 7, 3, 2, 3, 6, 7, 6, 4, 3, 4, 2, 1, 5, 3, 7, 2,
```

```
7, 1, 3, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 3, 3, 7, 7, 1, 2, 3, 3, 6,
2, 2, 4, 2, 2, 3, 3, 2, 6, 6, 3, 5, 5, 5, 2, 2, 2, 2, 4, 5, 2, 4,
4, 3, 5, 1, 7, 3, 3, 5, 3, 5, 3, 3, 6, 6, 1, 3, 3, 3, 2, 2, 3, 4,
3, 3, 3, 2, 6, 3, 4, 5, 6, 3, 3, 2])
```

In [68]:

```
#validation test
print("Naive bayes Accuracy is:",metrics.accuracy_score(y_test_tf1, bnk1_pred_tf))
print("Naive bayes f1 score is :",f1_score(y_test_tf1, bnk1_pred_tf, average='macro'))
print("Naive bayes recall score is :",recall_score(y_test_tf1, bnk1_pred_tf, average='macro'))
print("Naive bayes precison score is :",precision_score(y_test_tf1, bnk1_pred_tf, average='macro')
)
print("Naive bayes precison score is :",precision_score(y_test_tf1, bnk1_pred_tf, average='macro')
)
print("Naive bayes matthews_corrcoef score is :",matthews_corrcoef(y_test_tf1, bnk1_pred_tf))
```

```
Naive bayes Accuracy is: 0.6561224489795918
Naive bayes f1 score is : 0.6454996531177891
Naive bayes recall score is : 0.6577895009825048
Naive bayes precision score is : 0.6855335181343837
Naive bayes precision score is : 0.6855335181343837
Naive bayes matthews_corrcoef score is : 0.6091506068953487
```

Support vector machine with linear kernel using Count vectorizer

In [12]:

```
#fitting svm using count vectorizer
from sklearn.svm import SVC
svm_k = SVC(kernel = 'linear', random_state=0)
svm_k.fit(X_train_cv1,y_train_cv1)
```

Out[12]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

In []:

```
#cross validation test on training set
cross_svm_cv = cross_val_score(estimator=svm_k, X=X_train_cv1, y=y_train_cv1, cv=10)
cross_bnk_cv
```

In [34]:

```
svm_pred_cv = svm_k.predict(X_test_cv1)
svm_pred_cv
```

Out[34]:

```
array([4, 7, 3, 5, 6, 2, 2, 1, 5, 4, 7, 4, 4, 2, 1, 4, 5, 4, 6, 4, 7, 7,
       7, 5, 1, 2, 1, 4, 2, 1, 1, 7, 2, 2, 7, 3, 2, 6, 2, 1, 4, 7, 5, 6,
       1, 1, 4, 4, 7, 6, 7, 1, 1, 3, 5, 7, 2, 1, 3, 4, 4, 4, 5, 1, 4, 1,
       2, 2, 4, 6, 5, 2, 3, 6, 6, 7, 2, 2, 2, 7, 6, 3, 3, 2, 4, 3, 3, 6,
       4, 3, 1, 7, 7, 4, 6, 6, 5, 1, 4, 7, 3, 7, 6, 7, 2, 4, 4, 1, 7, 2,
       5, 4, 7, 4, 6, 5, 3, 4, 5, 7, 5, 3, 2, 1, 2, 4, 2, 7, 5, 7, 1, 2,
       1, 1, 2, 5, 1, 2, 2, 3, 6, 3, 3, 1, 1, 5, 4, 6, 2, 5, 1, 1, 4, 2,
       7, 7, 2, 6, 5, 7, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 7,
       2, 3, 2, 3, 2, 3, 1, 7, 4, 6, 5, 4, 6, 4, 7, 1, 3, 2, 1, 7, 4, 2,
       2, 2, 5, 1, 7, 7, 2, 3, 2, 7, 6, 2, 5, 7, 5, 7, 4, 1, 3, 2, 7, 4,
       3, 2, 5, 2, 1, 6, 5, 7, 2, 6, 2, 1, 6, 4, 1, 7, 3, 4, 3, 1, 3, 1,
       3, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
       5, 3, 7, 5, 5, 3, 1, 1, 4, 5, 5, 1, 7, 6, 6, 6, 7, 2, 7, 4, 6, 3,
       3, 5, 2, 7, 3, 3, 1, 6, 4, 1, 4, 2, 5, 1, 1, 4, 6, 5, 5, 4, 1, 6, 3,
       6, 3, 7, 5, 3, 1, 4, 3, 2, 5, 7, 2, 2, 7, 2, 1, 7, 5, 4, 4, 1, 7,
       4, 1, 4, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 4, 1, 2, 2, 4, 7, 5, 1, 5,
```

```

5, 5, 3, 5, 1, 1, 6, 1, 7, 2, 1, 3, 5, 1, 6, 7, 2, 1, 7, 2, 1, 6,
5, 1, 7, 1, 2, 4, 2, 1, 7, 4, 7, 6, 4, 7, 3, 2, 4, 4, 3, 3, 4, 5,
4, 6, 3, 6, 5, 7, 3, 7, 5, 3, 5, 2, 5, 4, 2, 7, 5, 5, 1, 4, 7, 7,
4, 2, 6, 6, 1, 4, 6, 4, 7, 5, 4, 2, 1, 3, 1, 1, 2, 6, 4, 5, 7, 4,
6, 2, 5, 2, 2, 1, 3, 2, 6, 7, 3, 6, 2, 6, 7, 1, 3, 4, 6, 7, 2, 7,
2, 1, 4, 2, 4, 3, 4, 7, 3, 1, 5, 5, 7, 4, 5, 1, 2, 1, 2, 5, 2, 2,
2, 5, 2, 4, 4, 2, 1, 2, 2, 7, 5, 7, 4, 4, 2, 1, 2, 5, 5, 5, 7, 5,
1, 2, 6, 2, 1, 5, 1, 4, 7, 3, 7, 1, 2, 5, 4, 1, 7, 4, 3, 4, 7, 3,
3, 3, 3, 5, 2, 6, 5, 6, 3, 3, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
7, 2, 3, 3, 2, 3, 6, 3, 7, 5, 5, 1, 2, 1, 3, 7, 5, 7, 7, 4, 3, 5,
3, 3, 3, 3, 3, 3, 6, 7, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 7, 1, 2,
5, 7, 1, 4, 1, 1, 7, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 2, 3, 6, 7, 7,
1, 6, 3, 7, 3, 7, 4, 7, 3, 2, 2, 1, 4, 6, 1, 6, 1, 4, 1, 3, 1, 3,
6, 4, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 7, 3, 7, 3, 5, 1, 7, 6, 1,
2, 5, 6, 5, 6, 3, 3, 1, 3, 5, 1, 2, 5, 6, 7, 7, 5, 4, 7, 1, 6, 2,
1, 1, 1, 3, 4, 1, 7, 1, 3, 3, 6, 5, 5, 5, 7, 2, 6, 2, 5, 3, 3, 5,
3, 2, 4, 2, 1, 7, 3, 5, 2, 3, 2, 7, 5, 5, 7, 5, 4, 1, 7, 4, 5, 1,
5, 4, 1, 1, 5, 1, 6, 3, 6, 5, 1, 7, 4, 2, 3, 5, 6, 4, 6, 7, 6, 1,
1, 7, 6, 4, 5, 7, 7, 4, 3, 1, 2, 4, 1, 3, 4, 7, 4, 3, 2, 3, 1, 1,
2, 5, 4, 5, 7, 4, 6, 5, 2, 3, 3, 1, 2, 2, 7, 1, 4, 4, 5, 4, 1, 7,
5, 2, 7, 1, 5, 3, 3, 3, 4, 3, 1, 4, 2, 6, 2, 7, 2, 5, 1, 2, 7, 7,
3, 2, 5, 3, 4, 2, 3, 2, 3, 1, 3, 2, 3, 2, 7, 7, 2, 4, 6, 5, 2, 4,
5, 3, 3, 2, 5, 5, 2, 3, 1, 3, 7, 3, 5, 3, 1, 7, 7, 5, 5, 7, 1, 5,
4, 5, 1, 7, 7, 4, 1, 3, 5, 2, 1, 3, 6, 3, 7, 3, 3, 7, 4, 5, 2, 2,
7, 7, 7, 1, 5, 3, 5, 3, 7, 3, 5, 7, 6, 4, 2, 4, 3, 1, 5, 6, 7, 5,
7, 1, 5, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 4, 2, 2, 7, 1, 7, 5, 3, 7,
5, 2, 5, 2, 2, 6, 3, 2, 5, 4, 2, 1, 5, 5, 2, 2, 2, 2, 5, 6, 4,
4, 3, 5, 1, 7, 3, 3, 5, 3, 5, 3, 6, 7, 4, 1, 3, 7, 7, 6, 5, 3, 5,
3, 6, 4, 2, 6, 3, 6, 5, 5, 3, 3, 2])

```

In [35]:

```

#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_cv1, svm_pred_cv))
print("SVM(LINEAR) f1 score is :",f1_score(y_test_cv1, svm_pred_cv, average='macro'))
print("SVM(LINEAR) recall score is :",recall_score(y_test_cv1, svm_pred_cv, average='macro'))
print("SVM(LINEAR) precison score is :",precision_score(y_test_cv1, svm_pred_cv, average='macro'))
print("SVM(LINEAR) precison score is :",precision_score(y_test_cv1, svm_pred_cv, average='macro'))
print("SVM(LINEAR) matthews_corrcoef score is :",matthews_corrcoef(y_test_cv1, svm_pred_cv))

```

```

SVM Accuracy is: 0.7612244897959184
SVM(LINEAR) f1 score is : 0.7552869553425852
SVM(LINEAR) recall score is : 0.7613637364325591
SVM(LINEAR) precison score is : 0.758896633595664
SVM(LINEAR) precison score is : 0.758896633595664
SVM(LINEAR) matthews_corrcoef score is : 0.7226690162359307

```

SUPPORT VECTOR MACHINE USING TFIDF VECTORIZER

In [13]:

```

#fitting svm model using tfidf vectorizer
svm_k.fit(X_train_tf1,y_train_tf1)

```

Out[13]:

```

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)

```

In []:

```

from sklearn.model_selection import cross_val_score
cross_svm_tf = cross_val_score(estimator=svm_k, X=X_train_tf1, y=y_train_tf1, cv=10)
cross_svm_tf

```

In [14]:

```

svm_pred_tf = svm_k.predict(X_test_tf1)

```



```
svm_pred_tf
```

```
Out[14]:
```

```
array([4, 7, 3, 5, 6, 2, 2, 1, 5, 4, 7, 4, 4, 2, 1, 4, 5, 6, 6, 4, 7, 6,
       7, 5, 1, 2, 1, 4, 4, 1, 1, 6, 4, 2, 7, 3, 2, 6, 4, 1, 4, 7, 6, 6,
       1, 6, 4, 4, 7, 6, 7, 1, 1, 3, 5, 7, 2, 1, 3, 4, 4, 4, 6, 1, 4, 1,
       2, 2, 4, 6, 5, 2, 3, 6, 6, 6, 2, 2, 7, 6, 3, 3, 2, 4, 3, 3, 6,
       6, 3, 6, 7, 7, 4, 6, 6, 5, 1, 6, 7, 3, 7, 6, 7, 2, 4, 6, 1, 6, 2,
       5, 4, 7, 4, 6, 2, 3, 4, 5, 7, 5, 3, 2, 1, 2, 4, 2, 7, 5, 7, 7, 5,
       1, 6, 2, 6, 1, 2, 2, 3, 6, 3, 3, 1, 1, 4, 4, 6, 2, 5, 1, 1, 4, 2,
       6, 6, 2, 6, 5, 6, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 6,
       2, 3, 6, 3, 2, 6, 1, 7, 4, 6, 2, 4, 6, 4, 7, 1, 3, 2, 1, 7, 4, 2,
       2, 2, 5, 1, 7, 7, 4, 3, 2, 7, 6, 2, 1, 7, 5, 7, 6, 1, 3, 2, 7, 4,
       3, 4, 2, 2, 1, 6, 1, 7, 2, 6, 2, 1, 6, 4, 1, 7, 1, 4, 3, 1, 3, 5,
       3, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
       1, 3, 7, 5, 6, 3, 6, 1, 4, 5, 5, 1, 7, 6, 6, 6, 6, 2, 7, 7, 6, 3,
       3, 6, 2, 7, 3, 1, 6, 6, 1, 4, 2, 4, 1, 1, 4, 6, 5, 5, 4, 1, 6, 3,
       6, 3, 7, 5, 3, 1, 4, 3, 6, 7, 7, 2, 2, 7, 2, 1, 7, 6, 4, 3, 1, 7,
       4, 1, 4, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 6, 1, 2, 6, 4, 6, 6, 1, 5,
       5, 5, 3, 5, 1, 1, 6, 1, 7, 2, 1, 3, 6, 6, 6, 7, 2, 1, 6, 2, 1, 6,
       5, 1, 7, 1, 2, 4, 2, 1, 7, 4, 7, 6, 4, 7, 3, 2, 6, 4, 3, 3, 4, 5,
       4, 6, 3, 6, 1, 7, 3, 7, 5, 3, 5, 2, 6, 4, 2, 7, 5, 5, 6, 4, 7, 7,
       4, 2, 6, 6, 1, 4, 6, 4, 7, 5, 4, 2, 1, 3, 1, 4, 2, 6, 6, 5, 7, 4,
       6, 4, 5, 2, 2, 1, 3, 2, 6, 7, 3, 6, 2, 6, 7, 1, 3, 4, 6, 7, 2, 7,
       2, 1, 4, 2, 4, 3, 6, 7, 3, 1, 5, 5, 7, 4, 7, 1, 2, 1, 4, 5, 2, 2,
       4, 1, 2, 4, 4, 4, 6, 2, 2, 7, 5, 7, 4, 4, 2, 1, 2, 5, 1, 4, 7, 5,
       1, 2, 6, 2, 1, 2, 1, 6, 7, 3, 7, 2, 5, 4, 1, 7, 4, 7, 4, 7, 3,
       3, 3, 3, 4, 2, 6, 5, 6, 3, 3, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
       7, 6, 3, 3, 2, 3, 6, 3, 7, 5, 5, 1, 2, 1, 3, 6, 7, 7, 7, 4, 3, 7,
       3, 3, 3, 3, 3, 3, 6, 6, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 7, 6, 2,
       5, 6, 1, 4, 1, 6, 7, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 6, 3, 6, 7, 7,
       1, 6, 3, 7, 3, 7, 4, 7, 3, 2, 2, 1, 6, 6, 1, 6, 1, 4, 1, 3, 1, 3,
       6, 4, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 6, 3, 7, 3, 5, 1, 7, 6, 1,
       2, 6, 6, 7, 6, 3, 3, 1, 3, 5, 1, 2, 6, 6, 7, 7, 5, 4, 7, 1, 6, 4,
       1, 1, 1, 4, 6, 1, 7, 1, 3, 3, 6, 5, 4, 5, 7, 2, 6, 2, 5, 3, 3, 5,
       3, 2, 4, 4, 1, 7, 3, 5, 2, 3, 4, 7, 6, 5, 7, 1, 4, 4, 7, 4, 5, 1,
       5, 4, 1, 1, 5, 1, 6, 3, 6, 5, 1, 7, 4, 2, 3, 5, 6, 4, 6, 7, 6, 1,
       1, 7, 6, 4, 5, 6, 7, 4, 3, 4, 2, 4, 1, 3, 4, 7, 4, 3, 2, 3, 1, 1,
       2, 4, 2, 5, 7, 4, 6, 5, 2, 3, 3, 1, 2, 2, 7, 1, 4, 4, 4, 4, 1, 7,
       5, 2, 7, 1, 1, 6, 3, 3, 4, 3, 6, 4, 2, 6, 2, 7, 2, 5, 1, 2, 7, 7,
       3, 2, 5, 4, 4, 2, 3, 2, 3, 1, 4, 2, 3, 2, 7, 7, 2, 4, 6, 5, 2, 4,
       5, 3, 3, 2, 5, 5, 2, 3, 1, 3, 7, 3, 5, 3, 5, 7, 7, 5, 5, 6, 1, 5,
       4, 5, 6, 7, 7, 6, 1, 3, 5, 6, 1, 3, 6, 3, 7, 3, 3, 7, 4, 5, 2, 2,
       7, 7, 6, 1, 2, 3, 7, 3, 6, 3, 6, 7, 6, 4, 2, 4, 2, 1, 5, 6, 7, 1,
       7, 1, 5, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 6, 2, 6, 7, 1, 7, 1, 3, 7,
       5, 2, 4, 2, 2, 6, 3, 2, 6, 6, 2, 5, 5, 5, 2, 2, 6, 2, 4, 5, 6, 6,
       4, 3, 5, 1, 7, 3, 3, 6, 3, 5, 3, 6, 7, 4, 1, 3, 7, 7, 6, 2, 2, 6,
       3, 6, 4, 4, 6, 3, 6, 5, 6, 3, 3, 2]])
```

```
In [16]:
```

```
#validation test
print("bernouli Naive bayes Accuracy is:",metrics.accuracy_score(y_test_tf1, svm_pred_tf))
print("bernouli Naive bayes f1 score is :",f1_score(y_test_tf1, svm_pred_tf, average='macro'))
print("bernouli Naive bayes recall score is :",recall_score(y_test_tf1, svm_pred_tf,
average='macro'))
print("bernouli Naive bayes precison score is :",precision_score(y_test_tf1, svm_pred_tf, average=
'macro'))
print("bernouli Naive bayes precison score is :",precision_score(y_test_tf1, svm_pred_tf, average=
'macro'))
print("bernouli Naive bayes matthews_corrcoef score is :",matthews_corrcoef(y_test_tf1,
svm_pred_tf))
```

```
bernouli Naive bayes Accuracy is: 0.7714285714285715
bernouli Naive bayes f1 score is : 0.7703153649580219
bernouli Naive bayes recall score is : 0.7727650163003199
bernouli Naive bayes precison score is : 0.7788233091349863
bernouli Naive bayes precison score is : 0.7788233091349863
bernouli Naive bayes matthews_corrcoef score is : 0.7349163892338874
```

SUPPORT VECTOR MACHINE MODEL POLYNOMIAL KERNAL

USING COUNT VECTORIZER

In [39]:

```
#fitting svm using count vectorizer
from sklearn.svm import SVC
Psvm_k = SVC(kernel = 'poly', random_state=0)
Psvm_k.fit(X_train_cv1,y_train_cv1)
```

Out[39]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='poly',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

In []:

```
#cross validation test
cross_psvm_cv = cross_val_score(estimator=psvm_k, X=X_train_cv1, y=y_train_cv1, cv=10)
cross_psvm_cv
```

In [40]:

```
psvm_pred_cv = Psvm_k.predict(X_test_cv1)
psvm_pred_cv
```

Out[40]:

```
array([3, 7, 3, 5, 3, 3, 3, 1, 3, 3, 3, 4, 3, 2, 3, 3, 5, 3, 3, 4, 3, 3,
       7, 5, 1, 2, 1, 3, 3, 1, 1, 3, 3, 3, 7, 3, 3, 3, 3, 1, 4, 3, 2, 3,
       1, 3, 4, 4, 7, 3, 3, 1, 3, 3, 5, 3, 3, 3, 3, 4, 3, 4, 4, 1, 3, 1,
       3, 3, 3, 3, 5, 2, 3, 3, 3, 3, 2, 2, 3, 3, 3, 3, 3, 2, 4, 3, 3, 3,
       4, 3, 6, 3, 3, 3, 3, 6, 5, 1, 3, 7, 3, 3, 3, 3, 3, 3, 7, 3, 3,
       5, 4, 3, 4, 3, 3, 3, 3, 5, 3, 1, 3, 2, 3, 3, 3, 3, 3, 1, 3, 1, 3,
       6, 1, 2, 3, 1, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 1, 1, 4, 3,
       3, 3, 3, 3, 1, 3, 4, 3, 3, 5, 3, 7, 1, 3, 6, 3, 3, 1, 3, 1, 3, 3,
       3, 3, 3, 3, 3, 3, 1, 7, 3, 3, 3, 3, 3, 3, 3, 1, 3, 2, 6, 7, 3, 3,
       3, 2, 1, 1, 3, 6, 3, 3, 3, 3, 3, 2, 3, 7, 5, 3, 3, 3, 3, 2, 3, 4,
       3, 2, 3, 3, 1, 6, 2, 7, 3, 3, 3, 1, 3, 3, 1, 7, 3, 3, 3, 3, 3, 1,
       3, 3, 3, 7, 1, 3, 3, 2, 5, 1, 3, 3, 3, 1, 2, 1, 2, 3, 3, 3, 3, 3,
       5, 3, 3, 1, 7, 3, 3, 1, 3, 5, 5, 4, 7, 3, 3, 3, 3, 2, 3, 3, 3, 3,
       3, 7, 3, 3, 3, 1, 3, 3, 3, 1, 4, 2, 5, 1, 1, 3, 3, 5, 5, 3, 1, 3, 3,
       3, 3, 3, 1, 3, 1, 3, 3, 3, 7, 3, 3, 3, 3, 3, 3, 3, 3, 4, 3, 1, 7,
       4, 1, 3, 1, 3, 1, 3, 3, 4, 3, 6, 1, 3, 3, 1, 2, 3, 3, 3, 3, 1, 5,
       1, 5, 3, 1, 1, 1, 3, 1, 3, 3, 3, 3, 4, 3, 3, 7, 3, 1, 6, 3, 1, 3,
       7, 1, 3, 1, 3, 3, 2, 3, 3, 3, 7, 3, 4, 3, 3, 2, 3, 4, 3, 3, 4, 5,
       4, 3, 3, 3, 3, 3, 3, 3, 4, 3, 2, 3, 2, 3, 2, 3, 2, 5, 5, 3, 7, 3,
       3, 2, 3, 3, 1, 3, 6, 4, 3, 6, 4, 2, 1, 3, 3, 3, 3, 3, 3, 5, 3, 3,
       3, 2, 5, 3, 2, 1, 3, 3, 3, 3, 3, 3, 3, 7, 1, 3, 4, 3, 3, 3, 7,
       3, 1, 3, 3, 3, 3, 3, 7, 3, 1, 1, 3, 7, 4, 3, 1, 3, 1, 3, 6, 3, 2,
       5, 5, 2, 4, 3, 2, 3, 3, 3, 7, 5, 3, 3, 4, 3, 1, 2, 5, 3, 4, 3, 1,
       1, 3, 3, 3, 1, 3, 3, 3, 7, 3, 3, 3, 3, 1, 4, 1, 3, 3, 3, 3, 7, 3,
       3, 3, 3, 3, 3, 5, 3, 3, 3, 5, 2, 7, 3, 3, 2, 3, 1, 1, 3, 3, 3,
       7, 3, 3, 3, 2, 3, 3, 3, 2, 5, 2, 3, 2, 1, 3, 3, 6, 3, 3, 3, 3, 7,
       3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 4, 1, 2, 3, 3, 3, 7, 7, 3, 2,
       5, 7, 1, 3, 5, 3, 3, 3, 1, 7, 3, 1, 1, 3, 3, 1, 2, 5, 3, 3, 3, 3,
       1, 3, 3, 7, 3, 7, 2, 3, 3, 3, 3, 3, 3, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       3, 4, 1, 2, 1, 5, 3, 2, 5, 3, 6, 3, 7, 3, 3, 3, 3, 1, 1, 7, 3, 1,
       3, 3, 3, 7, 3, 3, 3, 1, 3, 1, 5, 3, 6, 6, 3, 3, 5, 4, 3, 1, 6, 3,
       1, 1, 1, 3, 6, 1, 7, 3, 3, 3, 3, 5, 5, 5, 7, 3, 3, 3, 5, 3, 3, 3,
       3, 2, 3, 2, 1, 7, 3, 5, 2, 3, 3, 3, 3, 5, 7, 4, 3, 3, 3, 4, 5, 1,
       5, 3, 1, 1, 1, 1, 3, 3, 3, 3, 1, 4, 3, 3, 3, 3, 3, 3, 3, 7, 3, 3,
       1, 4, 3, 3, 1, 3, 7, 3, 3, 3, 2, 3, 1, 3, 3, 7, 3, 3, 2, 3, 1, 5,
       3, 4, 3, 2, 3, 6, 3, 1, 3, 3, 3, 1, 3, 2, 3, 3, 4, 3, 5, 3, 1, 7,
       5, 3, 3, 1, 2, 3, 3, 3, 3, 3, 6, 3, 3, 3, 2, 3, 2, 5, 1, 3, 7, 3,
       3, 2, 5, 3, 4, 2, 3, 2, 3, 3, 3, 3, 3, 2, 3, 6, 3, 3, 3, 1, 3, 3,
       2, 3, 3, 2, 1, 5, 3, 3, 3, 3, 7, 3, 1, 3, 1, 2, 3, 5, 5, 3, 1, 5,
       4, 3, 3, 3, 7, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 7, 3, 1, 2, 3,
       3, 6, 3, 1, 5, 3, 7, 3, 3, 3, 3, 6, 3, 6, 3, 4, 3, 3, 1, 3, 7, 2,
       6, 1, 3, 1, 1, 3, 1, 3, 3, 2, 3, 3, 1, 3, 3, 6, 6, 1, 3, 3, 3, 3,
       5, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 1, 2, 3, 3, 2, 3, 2, 3, 5, 3, 3,
       3, 3, 5, 3, 3, 3, 3, 4, 3, 4, 3, 3, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 4, 6, 3, 3, 5, 6, 3, 3, 2])
```

In [41]:

```
#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_cv1, psvm_pred_cv))
print("SVM(POLY) f1 score is :",f1_score(y_test_cv1, psvm_pred_cv, average='macro'))
print("SVM(POLY) recall score is :",recall_score(y_test_cv1, psvm_pred_cv, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_cv1, psvm_pred_cv, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_cv1, psvm_pred_cv, average='macro'))
print("SVM(POLY) matthews_corrcoef score is :",matthews_corrcoef(y_test_cv1, psvm_pred_cv))
```

```
SVM Accuracy is: 0.4275510204081633
SVM(POLY) f1 score is : 0.4252104279672686
SVM(POLY) recall score is : 0.432174215827787
SVM(POLY) precison score is : 0.6195674993728799
SVM(POLY) precison score is : 0.6195674993728799
SVM(POLY) matthews_corrcoef score is : 0.39666228548651233
```

SVM MODEL POLY KERNAL USING TFIDF VECTORIZER

In [43]:

```
#fitting the svm model
Psvm_k.fit(X_train_tfl,y_train_tfl)
```

Out[43]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='poly',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

In []:

```
#cross validation test
cross_psvm_tf = cross_val_score(estimator=psvm_k, X=X_train_tfl, y=y_train_tfl, cv=10)
cross_psvm_tf
```

In [44]:

```
psvm_pred_tf = Psvm_k.predict(X_test_tfl)
psvm_pred_tf
```

Out[44]:

```
array([4, 7, 3, 5, 6, 2, 5, 1, 5, 4, 7, 4, 4, 2, 5, 1, 5, 6, 1, 4, 6, 6,
       7, 5, 6, 2, 1, 6, 4, 5, 1, 6, 6, 6, 7, 3, 2, 6, 4, 6, 4, 7, 6, 6,
       1, 6, 4, 4, 7, 1, 6, 1, 6, 6, 5, 7, 2, 1, 3, 4, 4, 6, 6, 6, 6, 5,
       2, 6, 4, 6, 5, 2, 3, 2, 6, 6, 6, 2, 6, 7, 6, 3, 3, 6, 6, 3, 3, 6,
       6, 3, 6, 6, 7, 4, 6, 6, 6, 1, 6, 7, 6, 7, 6, 7, 2, 4, 6, 6, 6, 2,
       5, 4, 7, 4, 6, 5, 3, 4, 5, 7, 5, 3, 6, 6, 2, 4, 2, 7, 5, 7, 7, 1,
       6, 6, 6, 6, 1, 6, 2, 3, 4, 3, 3, 6, 1, 4, 4, 6, 6, 5, 5, 6, 4, 2,
       6, 6, 4, 6, 4, 6, 4, 5, 1, 6, 6, 7, 6, 3, 6, 6, 6, 5, 7, 1, 6, 7,
       2, 3, 6, 3, 2, 6, 1, 7, 4, 6, 6, 6, 6, 6, 7, 1, 3, 2, 6, 7, 4, 6,
       2, 6, 5, 1, 7, 7, 6, 1, 2, 7, 6, 2, 1, 7, 7, 7, 6, 1, 3, 2, 7, 6,
       3, 4, 6, 2, 1, 6, 6, 7, 2, 6, 4, 1, 7, 6, 7, 7, 1, 4, 3, 1, 3, 5,
       4, 6, 2, 7, 6, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 6, 2, 7, 2, 3, 2,
       6, 3, 7, 5, 6, 6, 1, 6, 6, 5, 6, 6, 6, 6, 2, 6, 7, 6, 6,
       3, 6, 6, 7, 3, 6, 6, 6, 7, 6, 2, 6, 6, 4, 5, 6, 5, 5, 4, 1, 6, 4,
       6, 3, 7, 5, 3, 1, 6, 3, 6, 6, 2, 2, 6, 7, 2, 6, 7, 6, 4, 6, 1, 7,
       4, 1, 4, 6, 7, 6, 3, 6, 6, 3, 6, 1, 3, 6, 6, 6, 6, 6, 6, 1, 6,
       5, 5, 3, 5, 6, 1, 6, 1, 7, 2, 1, 3, 6, 6, 6, 7, 2, 5, 6, 2, 1, 6,
       6, 6, 6, 1, 2, 4, 2, 6, 7, 4, 7, 6, 6, 7, 3, 2, 4, 4, 2, 3, 4, 5,
       4, 6, 3, 6, 6, 7, 3, 7, 5, 3, 5, 2, 6, 4, 7, 7, 6, 5, 6, 5, 7, 7,
       4, 2, 6, 6, 1, 4, 6, 4, 7, 7, 4, 2, 1, 3, 6, 6, 2, 6, 6, 1, 7, 4,
       6, 4, 5, 2, 2, 6, 3, 5, 6, 7, 3, 6, 2, 6, 7, 1, 3, 7, 6, 7, 2, 7,
       2, 6, 3, 2, 4, 3, 6, 7, 3, 1, 5, 5, 7, 4, 7, 1, 2, 1, 4, 6, 2, 6,
       6, 7, 5, 4, 3, 4, 6, 5, 6, 7, 5, 7, 5, 4, 2, 5, 2, 5, 6, 4, 7, 1,
       5, 5, 5, 2, 1, 2, 5, 6, 7, 6, 7, 7, 2, 5, 4, 1, 7, 6, 7, 4, 7, 3,
       3, 5, 6, 6, 6, 5, 6, 3, 4, 4, 2, 7, 6, 4, 2, 6, 1, 1, 7, 2, 3,
```

```

6, 6, 7, 3, 6, 3, 6, 3, 7, 5, 5, 1, 2, 1, 3, 6, 6, 7, 7, 4, 3, 7,
3, 3, 3, 3, 3, 3, 6, 7, 1, 1, 6, 7, 4, 1, 2, 3, 7, 6, 7, 7, 6, 2,
5, 7, 1, 4, 6, 6, 7, 6, 1, 7, 6, 1, 5, 6, 4, 1, 5, 6, 3, 6, 7, 7,
1, 6, 3, 7, 6, 7, 4, 7, 6, 2, 2, 1, 6, 6, 1, 7, 4, 6, 1, 3, 6, 3,
6, 6, 1, 2, 1, 5, 6, 1, 5, 3, 6, 3, 7, 6, 6, 7, 3, 5, 5, 7, 6, 1,
2, 6, 6, 6, 6, 3, 3, 6, 3, 5, 1, 2, 6, 6, 7, 7, 5, 4, 7, 1, 6, 4,
6, 1, 5, 4, 6, 1, 7, 6, 3, 3, 6, 5, 4, 5, 7, 5, 6, 2, 4, 3, 3, 5,
3, 7, 3, 4, 7, 7, 4, 6, 2, 3, 4, 7, 6, 6, 7, 6, 6, 6, 7, 4, 5, 6,
5, 4, 1, 1, 5, 7, 6, 3, 6, 6, 1, 7, 6, 2, 3, 6, 6, 4, 6, 7, 6, 1,
1, 6, 6, 4, 5, 6, 7, 6, 4, 6, 2, 1, 5, 3, 4, 7, 6, 3, 2, 6, 1, 6,
2, 4, 2, 6, 7, 6, 6, 5, 6, 6, 6, 6, 2, 6, 7, 6, 6, 4, 4, 4, 6, 7,
5, 2, 7, 1, 1, 6, 3, 4, 4, 3, 6, 6, 6, 6, 6, 7, 6, 5, 1, 2, 7, 7,
3, 2, 5, 4, 6, 4, 3, 2, 3, 1, 4, 6, 3, 2, 7, 6, 6, 4, 6, 5, 6, 6,
6, 3, 3, 2, 5, 6, 2, 3, 1, 3, 7, 3, 5, 3, 6, 7, 7, 5, 5, 6, 1, 6,
4, 5, 6, 7, 7, 6, 1, 3, 5, 6, 5, 3, 6, 3, 7, 3, 3, 7, 4, 6, 4, 7,
7, 7, 6, 1, 5, 3, 7, 3, 6, 3, 6, 7, 6, 6, 2, 4, 2, 6, 4, 6, 7, 1,
7, 1, 5, 1, 1, 2, 1, 3, 6, 6, 3, 3, 1, 6, 2, 6, 7, 6, 6, 5, 3, 6,
7, 6, 4, 2, 2, 6, 6, 2, 6, 6, 2, 4, 5, 5, 2, 2, 6, 2, 6, 5, 6, 6,
4, 3, 5, 6, 7, 3, 3, 6, 3, 5, 3, 6, 6, 1, 2, 7, 6, 6, 5, 4, 6,
3, 6, 4, 4, 6, 3, 6, 5, 6, 3, 3, 2])

```

In [45]:

```

#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_tf1, psvm_pred_tf))
print("SVM(POLY) f1 score is :",f1_score(y_test_tf1, psvm_pred_tf, average='macro'))
print("SVM(POLY) recall score is :",recall_score(y_test_tf1, psvm_pred_tf, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_tf1, psvm_pred_tf, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_tf1, psvm_pred_tf, average='macro'))
print("SVM(POLY) matthews_corrcoef score is :",matthews_corrcoef(y_test_tf1, psvm_pred_tf))

```

```

SVM Accuracy is: 0.6459183673469387
SVM(POLY) f1 score is : 0.6641451304427157
SVM(POLY) recall score is : 0.6497461716398538
SVM(POLY) precision score is : 0.7300070535209428
SVM(POLY) precision score is : 0.7300070535209428
SVM(POLY) matthews_corrcoef score is : 0.6015740722166161

```

SVM GAUSSIAN KERNAL USING COUNT VECTORIZER

In [53]:

```

#fitting svm using count vectorizer
from sklearn.svm import SVC
rsvm_k = SVC(kernel = 'rbf', random_state=0)
rsvm_k.fit(X_train_cv1,y_train_cv1)

```

Out[53]:

```

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)

```

In []:

```

#cross validation test
cross_rsvm_cv = cross_val_score(estimator=rsvm_k, X=X_train_cv1, y=y_train_cv1, cv=10)
cross_rsvm_cv

```

In [54]:

```

#prediction test
rsvm_pred_cv = rsvm_k.predict(X_test_cv1)
rsvm_pred_cv

```

Out[54]:

```

array([4, 7, 3, 5, 6, 2, 2, 1, 5, 4, 7, 4, 6, 2, 2, 6, 5, 7, 6, 4, 6, 7,
       7, 7, 1, 2, 1, 4, 2, 1, 1, 6, 4, 2, 7, 3, 2, 6, 2, 1, 5, 6, 6, 6,

```

```

1, 6, 4, 4, 7, 6, 7, 1, 6, 6, 5, 7, 2, 1, 3, 4, 4, 4, 7, 5, 4, 1,
2, 6, 4, 6, 5, 2, 3, 6, 6, 7, 2, 2, 2, 7, 6, 3, 3, 2, 4, 3, 3, 6,
7, 3, 7, 6, 7, 4, 6, 7, 5, 5, 6, 7, 3, 7, 6, 7, 2, 4, 4, 7, 6, 2,
5, 4, 7, 4, 6, 2, 3, 4, 5, 7, 5, 3, 5, 1, 2, 4, 2, 7, 5, 7, 7, 2,
1, 7, 7, 6, 1, 2, 2, 3, 6, 3, 3, 5, 1, 4, 4, 6, 2, 2, 1, 1, 4, 5,
7, 7, 2, 6, 5, 7, 4, 2, 1, 5, 6, 7, 5, 3, 6, 6, 6, 5, 7, 5, 6, 6,
2, 3, 6, 3, 2, 6, 1, 7, 4, 7, 5, 4, 6, 4, 7, 1, 3, 2, 7, 7, 4, 2,
6, 2, 5, 1, 7, 7, 4, 3, 2, 7, 6, 2, 5, 7, 5, 7, 6, 1, 3, 2, 6, 4,
3, 4, 4, 2, 1, 7, 5, 7, 2, 6, 2, 1, 6, 6, 1, 7, 3, 4, 3, 6, 3, 5,
4, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
5, 3, 7, 5, 7, 3, 6, 1, 4, 5, 5, 7, 7, 6, 6, 6, 6, 2, 7, 4, 6, 3,
3, 7, 2, 7, 3, 1, 6, 6, 1, 4, 2, 5, 1, 1, 4, 6, 5, 5, 4, 1, 6, 3,
6, 3, 7, 5, 3, 1, 4, 3, 2, 7, 7, 2, 2, 6, 2, 7, 7, 6, 7, 4, 1, 7,
4, 1, 4, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 7, 1, 2, 6, 4, 7, 6, 1, 5,
5, 5, 3, 5, 5, 1, 6, 1, 7, 2, 1, 3, 7, 7, 6, 7, 2, 1, 7, 2, 1, 6,
7, 1, 7, 1, 2, 4, 2, 1, 7, 4, 7, 6, 6, 7, 3, 2, 6, 4, 3, 3, 4, 5,
4, 6, 3, 6, 5, 6, 3, 7, 7, 3, 5, 2, 6, 6, 5, 6, 2, 5, 6, 4, 7, 7,
4, 2, 6, 6, 1, 4, 6, 4, 7, 7, 4, 2, 1, 3, 1, 4, 2, 6, 6, 5, 7, 4,
6, 4, 5, 2, 2, 1, 3, 2, 7, 6, 3, 6, 2, 6, 7, 1, 3, 7, 6, 7, 2, 7,
2, 1, 3, 6, 4, 3, 7, 7, 3, 1, 5, 5, 7, 7, 7, 1, 2, 1, 2, 6, 6, 2,
7, 5, 2, 4, 3, 4, 6, 2, 2, 7, 5, 7, 4, 4, 2, 1, 2, 5, 1, 4, 7, 5,
1, 2, 6, 2, 1, 2, 2, 6, 7, 4, 7, 7, 2, 5, 4, 1, 7, 6, 7, 4, 7, 3,
3, 3, 3, 5, 2, 6, 5, 6, 3, 5, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
7, 4, 3, 3, 2, 3, 6, 3, 7, 5, 2, 1, 2, 1, 3, 6, 7, 7, 6, 4, 3, 7,
3, 3, 3, 3, 3, 3, 6, 6, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 7, 7, 2,
5, 7, 1, 6, 5, 7, 7, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 5, 3, 6, 7, 7,
1, 6, 3, 7, 3, 7, 4, 7, 4, 2, 2, 1, 6, 6, 1, 6, 5, 6, 1, 3, 1, 3,
6, 6, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 7, 3, 7, 3, 5, 1, 7, 6, 1,
2, 6, 6, 7, 6, 3, 3, 1, 3, 5, 5, 2, 6, 6, 7, 7, 5, 7, 7, 1, 7, 4,
1, 1, 1, 3, 6, 1, 7, 5, 3, 3, 6, 5, 7, 5, 7, 2, 6, 2, 7, 3, 3, 5,
3, 5, 3, 4, 1, 7, 3, 7, 2, 4, 2, 7, 7, 5, 7, 7, 6, 6, 7, 4, 5, 1,
5, 4, 1, 1, 5, 1, 6, 3, 6, 5, 1, 7, 4, 2, 3, 5, 6, 4, 6, 7, 6, 1,
1, 7, 6, 4, 5, 6, 7, 4, 3, 4, 2, 4, 5, 3, 4, 7, 4, 3, 2, 3, 1, 5,
2, 4, 4, 5, 7, 7, 6, 5, 2, 3, 3, 7, 2, 2, 7, 1, 4, 4, 5, 4, 7, 7,
5, 2, 7, 1, 2, 4, 3, 3, 4, 3, 7, 4, 2, 6, 2, 7, 2, 5, 1, 2, 7, 7,
3, 2, 5, 4, 4, 2, 3, 2, 3, 6, 3, 2, 3, 2, 7, 7, 2, 4, 6, 5, 2, 6,
5, 3, 3, 2, 5, 5, 2, 3, 1, 7, 7, 3, 5, 3, 5, 7, 7, 5, 5, 7, 1, 5,
4, 5, 7, 7, 7, 6, 1, 3, 5, 6, 5, 3, 6, 3, 7, 3, 3, 7, 4, 1, 2, 2,
6, 7, 7, 1, 7, 3, 7, 3, 7, 3, 7, 6, 4, 2, 4, 2, 7, 5, 6, 7, 2,
7, 1, 1, 1, 1, 2, 1, 3, 5, 2, 3, 3, 1, 4, 6, 7, 7, 1, 7, 6, 3, 7,
5, 2, 5, 2, 2, 6, 3, 2, 6, 6, 2, 1, 5, 5, 2, 2, 6, 2, 4, 5, 6, 4,
4, 3, 5, 1, 7, 3, 3, 7, 3, 5, 3, 6, 7, 6, 1, 6, 7, 6, 6, 6, 3, 7,
3, 6, 4, 4, 6, 3, 6, 7, 7, 3, 3, 2])

```

In [55]:

```

#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_cv1, rsvm_pred_cv))
print("SVM(GAUSSIAN) f1 score is :",f1_score(y_test_cv1, rsvm_pred_cv, average='macro'))
print("SVM(GAUSSIAN) recall score is :",recall_score(y_test_cv1, rsvm_pred_cv, average='macro'))
print("SVM(GAUSSIAN) precision score is :",precision_score(y_test_cv1, rsvm_pred_cv,
average='macro'))
print("SVM(GAUSSIAN) precision score is :",precision_score(y_test_cv1, rsvm_pred_cv,
average='macro'))
print("SVM(GAUSSIAN) matthews_corrcoef score is :",matthews_corrcoef(y_test_cv1, rsvm_pred_cv))

```

```

SVM Accuracy is: 0.7285714285714285
SVM(GAUSSIAN) f1 score is : 0.7291178064515587
SVM(GAUSSIAN) recall score is : 0.7297118202418095
SVM(GAUSSIAN) precision score is : 0.743496271423443
SVM(GAUSSIAN) precision score is : 0.743496271423443
SVM(GAUSSIAN) matthews_corrcoef score is : 0.6859557574584636

```

SVM GAUSSAIAN USING TFIDF VECTORIZER

In [56]:

```

#fitting the svm model
rsvm_k.fit(X_train_tfl,y_train_tfl)

```

Out[56]:

```

SVM(GAUSSIAN) matthews_corrcoef score is : 0.6859557574584636

```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

In []:

```
#cross validation test
cross_psvm_tf = cross_val_score(estimator=psvm_k, X=X_train_tf1, y=y_train_tf1, cv=10)
cross_psvm_tf
```

In [58]:

```
#prediction test
rsvm_pred_tf = rsvm_k.predict(X_test_tf1)
rsvm_pred_tf
```

Out[58]:

```
array([4, 7, 3, 5, 6, 2, 2, 1, 5, 4, 7, 4, 4, 2, 1, 4, 5, 6, 6, 4, 7, 6,
       7, 5, 1, 2, 1, 4, 4, 1, 1, 6, 4, 2, 7, 3, 2, 6, 4, 1, 4, 7, 6, 6,
       1, 6, 4, 4, 7, 6, 6, 1, 6, 6, 5, 7, 2, 1, 3, 4, 4, 4, 6, 1, 4, 5,
       2, 6, 4, 6, 5, 2, 3, 2, 6, 6, 2, 2, 2, 7, 6, 3, 3, 2, 4, 3, 3, 6,
       6, 3, 6, 6, 7, 4, 6, 3, 6, 5, 1, 6, 7, 3, 7, 6, 7, 2, 4, 6, 1, 6, 2,
       5, 4, 7, 4, 6, 5, 3, 4, 5, 7, 5, 3, 2, 1, 2, 4, 2, 7, 5, 7, 7, 2,
       6, 6, 2, 6, 1, 6, 2, 3, 6, 3, 3, 1, 1, 4, 4, 6, 6, 5, 1, 1, 4, 2,
       6, 6, 2, 6, 5, 6, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 6,
       2, 3, 6, 3, 2, 6, 1, 7, 4, 6, 6, 4, 6, 4, 7, 1, 3, 2, 1, 7, 4, 2,
       2, 2, 5, 1, 7, 7, 4, 5, 2, 7, 6, 2, 1, 7, 5, 7, 6, 1, 3, 2, 7, 4,
       3, 4, 4, 2, 1, 6, 1, 7, 2, 6, 2, 1, 6, 6, 1, 7, 1, 4, 3, 1, 3, 5,
       4, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
       1, 3, 7, 5, 6, 6, 6, 1, 4, 5, 5, 4, 7, 6, 6, 6, 6, 2, 7, 7, 6, 3,
       3, 6, 2, 7, 3, 1, 6, 6, 1, 4, 2, 5, 1, 1, 4, 6, 5, 5, 4, 1, 6, 3,
       6, 3, 7, 5, 3, 1, 4, 3, 6, 7, 7, 2, 2, 7, 2, 6, 7, 6, 4, 6, 1, 7,
       4, 1, 4, 1, 7, 1, 3, 6, 4, 3, 6, 1, 3, 6, 1, 2, 6, 4, 6, 6, 1, 5,
       5, 5, 3, 5, 1, 1, 6, 1, 7, 2, 1, 3, 6, 6, 6, 7, 2, 1, 6, 2, 1, 6,
       6, 1, 6, 1, 2, 4, 2, 1, 7, 4, 7, 6, 4, 7, 3, 2, 6, 4, 2, 3, 4, 5,
       4, 6, 3, 6, 5, 6, 3, 7, 5, 3, 5, 2, 6, 4, 2, 7, 6, 5, 6, 4, 7, 6,
       4, 2, 6, 6, 1, 4, 6, 4, 7, 7, 4, 2, 1, 3, 1, 6, 2, 6, 6, 5, 7, 4,
       6, 4, 5, 2, 2, 6, 3, 2, 6, 7, 3, 6, 2, 6, 7, 1, 3, 4, 6, 7, 2, 7,
       2, 1, 3, 2, 4, 3, 6, 7, 3, 1, 5, 5, 7, 4, 7, 1, 2, 1, 4, 6, 2, 2,
       6, 1, 2, 4, 3, 4, 6, 2, 2, 7, 5, 7, 4, 4, 2, 5, 2, 5, 1, 4, 7, 1,
       1, 2, 6, 2, 1, 2, 1, 6, 7, 4, 7, 7, 2, 5, 4, 1, 7, 6, 7, 4, 7, 3,
       3, 3, 2, 4, 2, 6, 5, 6, 3, 3, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
       7, 6, 3, 3, 2, 3, 6, 3, 7, 5, 5, 1, 2, 1, 3, 6, 6, 7, 7, 4, 3, 7,
       3, 3, 3, 3, 3, 3, 6, 6, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 7, 6, 2,
       5, 6, 1, 4, 1, 6, 7, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 6, 3, 6, 7, 7,
       1, 6, 3, 7, 2, 7, 4, 7, 4, 2, 2, 1, 6, 6, 1, 6, 1, 6, 1, 3, 1, 3,
       6, 4, 1, 2, 1, 5, 6, 1, 5, 3, 6, 3, 7, 6, 2, 7, 3, 5, 1, 7, 6, 1,
       2, 6, 6, 7, 6, 3, 3, 1, 3, 5, 1, 2, 6, 6, 7, 7, 5, 4, 7, 1, 6, 4,
       1, 1, 5, 4, 6, 1, 7, 4, 3, 3, 6, 5, 4, 5, 7, 2, 6, 2, 5, 3, 3, 5,
       3, 2, 3, 4, 1, 7, 4, 5, 2, 3, 4, 7, 6, 5, 7, 4, 6, 6, 7, 4, 5, 1,
       5, 4, 1, 1, 5, 1, 6, 3, 6, 5, 1, 7, 6, 2, 3, 5, 6, 4, 6, 7, 6, 1,
       1, 7, 6, 4, 5, 6, 7, 4, 3, 6, 2, 4, 5, 3, 6, 7, 4, 3, 2, 3, 1, 5,
       2, 4, 4, 5, 7, 6, 6, 5, 2, 2, 2, 1, 2, 2, 7, 6, 4, 4, 4, 1, 7,
       5, 2, 7, 1, 5, 6, 3, 4, 4, 3, 6, 4, 2, 6, 6, 7, 2, 5, 1, 2, 7, 7,
       3, 2, 5, 4, 4, 4, 3, 2, 3, 6, 4, 2, 3, 2, 7, 6, 2, 4, 6, 5, 6, 6,
       5, 3, 3, 2, 5, 5, 2, 3, 1, 3, 7, 3, 5, 3, 1, 7, 7, 5, 5, 6, 1, 1,
       4, 5, 6, 7, 7, 6, 1, 3, 5, 6, 1, 3, 6, 3, 7, 3, 3, 7, 4, 5, 2, 2,
       7, 7, 6, 1, 2, 3, 7, 3, 6, 3, 6, 7, 6, 6, 2, 4, 2, 1, 5, 6, 7, 1,
       7, 1, 5, 1, 1, 2, 1, 3, 6, 2, 3, 3, 1, 6, 2, 6, 7, 1, 7, 5, 3, 6,
       5, 2, 4, 2, 2, 6, 3, 2, 6, 6, 2, 5, 5, 5, 2, 2, 6, 2, 6, 5, 6, 6,
       4, 3, 5, 1, 7, 3, 3, 6, 3, 5, 3, 6, 6, 1, 3, 7, 7, 6, 6, 4, 6,
       3, 6, 4, 4, 6, 3, 6, 5, 6, 3, 3, 2]])
```

In [59]:

```
#validation test
print("SVM Accuracy is:", metrics.accuracy_score(y_test_tf1, rsvm_pred_tf))
print("SVM(GAUSSIAN) f1 score is :", f1_score(y_test_tf1, rsvm_pred_tf, average='macro'))
print("SVM(GAUSSIAN) recall score is :", recall_score(y_test_tf1, rsvm_pred_tf, average='macro'))
print("SVM(GAUSSIAN) precision score is :", precision_score(y_test_tf1, rsvm_pred_tf,
    average='macro'))
print("SVM(GAUSSIAN) precision score is :", precision_score(y_test_tf1, rsvm_pred_tf,
```

```
average='macro'))
print("SVM(GAUSSIAN) matthews_corrcoef score is :",matthews_corrcoef(y_test_tf1, rsvm_pred_tf))
```

```
SVM Accuracy is: 0.7561224489795918
SVM(GAUSSIAN) f1 score is : 0.7604806966362749
SVM(GAUSSIAN) recall score is : 0.7571660022153222
SVM(GAUSSIAN) precision score is : 0.7789350219354324
SVM(GAUSSIAN) precision score is : 0.7789350219354324
SVM(GAUSSIAN) matthews_corrcoef score is : 0.7186124945138185
```

RANDOM FOREST CLASSIFIER USING COUNT VECTORIZER

In [68]:

```
#fitting random forest classifier
from sklearn.ensemble import RandomForestClassifier
rf_k = RandomForestClassifier(n_estimators=200, random_state=0)
rf_k.fit(X_train_cv1, y_train_cv1)
```

Out [68]:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                        criterion='gini', max_depth=None, max_features='auto',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=200,
                        n_jobs=None, oob_score=False, random_state=0, verbose=0,
                        warm_start=False)
```

In []:

```
#cross validation test on training set
cross_rf_cv = cross_val_score(estimator=rf_k, X=X_train_cv1, y=y_train_cv1, cv=10)
cross_bnk_cv
```

In [69]:

```
#prediction test
rf_pred_cv = rf_k.predict(X_test_cv1)
rf_pred_cv
```

Out [69]:

```
array([4, 7, 3, 1, 6, 2, 3, 1, 3, 4, 7, 4, 6, 2, 2, 6, 5, 1, 6, 4, 6, 7,
       7, 5, 1, 2, 1, 4, 2, 1, 1, 6, 2, 2, 7, 3, 2, 4, 2, 1, 4, 6, 4, 6,
       1, 1, 4, 4, 7, 6, 6, 1, 6, 3, 5, 7, 2, 1, 3, 4, 6, 4, 4, 1, 4, 1,
       3, 6, 6, 6, 5, 2, 3, 2, 6, 6, 2, 2, 7, 6, 3, 3, 4, 4, 3, 3, 6,
       4, 3, 1, 6, 6, 1, 6, 6, 5, 1, 6, 7, 3, 6, 6, 7, 2, 3, 1, 1, 6, 2,
       5, 4, 6, 4, 6, 2, 3, 6, 5, 7, 5, 3, 5, 1, 2, 4, 2, 7, 5, 7, 1, 2,
       1, 1, 2, 6, 1, 2, 3, 3, 6, 3, 3, 1, 1, 4, 4, 6, 2, 2, 1, 1, 4, 2,
       6, 6, 2, 6, 5, 7, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 1, 7, 1, 6, 6,
       2, 3, 6, 3, 2, 3, 1, 7, 4, 4, 2, 4, 6, 4, 6, 1, 3, 2, 1, 7, 4, 2,
       6, 2, 5, 1, 7, 7, 4, 3, 2, 7, 6, 2, 5, 7, 5, 7, 6, 1, 3, 2, 6, 4,
       3, 4, 4, 2, 1, 6, 5, 7, 2, 6, 2, 1, 2, 6, 1, 7, 3, 7, 3, 1, 3, 5,
       3, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 2, 2, 7, 2, 3, 2,
       5, 3, 7, 5, 7, 3, 6, 1, 4, 5, 5, 4, 7, 6, 6, 6, 6, 2, 6, 7, 6, 3,
       3, 7, 2, 7, 3, 1, 6, 6, 1, 4, 2, 1, 1, 1, 5, 6, 5, 5, 4, 1, 6, 3,
       6, 3, 6, 5, 3, 1, 4, 3, 2, 5, 6, 2, 2, 6, 2, 1, 7, 1, 4, 6, 1, 7,
       4, 1, 4, 1, 7, 1, 3, 3, 1, 3, 6, 1, 3, 6, 1, 2, 6, 6, 6, 6, 1, 5,
       5, 5, 3, 5, 1, 1, 6, 1, 6, 2, 1, 3, 4, 6, 6, 7, 2, 1, 7, 2, 1, 6,
       5, 1, 7, 1, 2, 4, 2, 1, 7, 4, 7, 6, 4, 7, 3, 2, 3, 4, 3, 3, 4, 1,
       4, 6, 3, 6, 2, 6, 3, 7, 1, 3, 2, 2, 6, 6, 2, 6, 2, 5, 6, 4, 6, 1,
       4, 2, 6, 6, 1, 3, 6, 4, 7, 7, 4, 2, 1, 3, 6, 1, 2, 6, 1, 5, 7, 6,
       6, 4, 5, 2, 2, 1, 3, 2, 6, 7, 3, 6, 2, 6, 7, 1, 3, 6, 6, 7, 2, 7,
       2, 1, 3, 2, 4, 3, 6, 7, 3, 1, 5, 2, 7, 4, 7, 1, 2, 1, 4, 6, 2, 2,
       2, 5, 2, 4, 3, 2, 1, 5, 2, 7, 5, 7, 2, 4, 2, 1, 2, 5, 1, 2, 7, 1,
       1, 2, 6, 2, 1, 2, 2, 1, 7, 3, 7, 1, 2, 5, 4, 1, 7, 6, 3, 4, 7, 3,
       3, 3, 3, 4, 2, 6, 5, 6, 3, 6, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
       7 2 3 3 2 3 6 3 7 5 2 1 2 1 3 6 6 7 7 4 3 5
```

```

1, 2, 3, 3, 2, 3, 3, 3, 1, 3, 2, 1, 2, 1, 3, 3, 3, 1, 3, 3,
3, 3, 3, 3, 3, 3, 6, 6, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 7, 6, 2,
5, 7, 1, 3, 1, 1, 1, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 4, 3, 6, 7, 7,
1, 6, 3, 7, 3, 7, 4, 6, 3, 2, 2, 1, 6, 6, 1, 6, 1, 6, 1, 3, 1, 3,
6, 4, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 7, 3, 7, 3, 5, 1, 7, 6, 1,
2, 6, 6, 2, 6, 3, 3, 1, 3, 5, 1, 2, 6, 6, 7, 7, 5, 4, 7, 1, 6, 4,
1, 1, 5, 4, 4, 1, 7, 1, 3, 3, 6, 5, 5, 5, 7, 2, 6, 2, 5, 3, 3, 2,
3, 2, 3, 4, 1, 7, 4, 5, 2, 3, 4, 7, 7, 5, 7, 4, 6, 1, 7, 4, 5, 1,
5, 6, 1, 1, 5, 1, 6, 3, 6, 4, 1, 7, 4, 2, 3, 5, 6, 4, 6, 7, 6, 1,
1, 6, 6, 4, 5, 6, 7, 3, 3, 2, 4, 1, 3, 4, 7, 4, 3, 2, 3, 1, 1,
2, 4, 6, 5, 7, 4, 6, 1, 2, 3, 3, 1, 2, 2, 7, 1, 4, 4, 5, 4, 1, 7,
5, 2, 7, 1, 2, 3, 3, 3, 4, 3, 6, 4, 2, 6, 2, 7, 2, 5, 1, 4, 6, 7,
3, 2, 5, 4, 4, 4, 3, 2, 3, 1, 3, 2, 3, 2, 7, 6, 2, 6, 6, 5, 2, 4,
4, 3, 3, 2, 5, 5, 2, 3, 6, 3, 7, 3, 5, 3, 5, 7, 7, 5, 5, 6, 1, 1,
2, 3, 1, 7, 7, 6, 1, 3, 5, 2, 5, 3, 6, 3, 7, 3, 3, 7, 4, 1, 4, 2,
6, 6, 6, 1, 5, 3, 7, 3, 6, 3, 6, 7, 6, 6, 2, 4, 3, 7, 4, 6, 7, 5,
7, 1, 4, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 3, 2, 6, 7, 1, 6, 1, 3, 6,
5, 2, 4, 2, 2, 6, 3, 2, 6, 2, 1, 5, 5, 2, 2, 6, 2, 2, 5, 6, 6,
4, 3, 5, 1, 7, 3, 3, 4, 3, 2, 3, 6, 7, 6, 1, 3, 7, 1, 6, 2, 3, 3,
3, 6, 4, 4, 6, 3, 6, 5, 6, 3, 3, 2])

```

In [70]:

```

#validation test
print("random forest Accuracy is:",metrics.accuracy_score(y_test_cv1, rf_pred_cv))
print("random forest f1 score is :",f1_score(y_test_cv1, rf_pred_cv, average='macro'))
print("random forest recall score is :",recall_score(y_test_cv1, rf_pred_cv, average='macro'))
print("random forest precision score is :",precision_score(y_test_cv1, rf_pred_cv, average='macro'))
)
print("random forest precision score is :",precision_score(y_test_cv1, rf_pred_cv, average='macro'))
)
print("random forest matthews_corrcoef score is :",matthews_corrcoef(y_test_cv1, rf_pred_cv))

```

```

random forest Accuracy is: 0.7214285714285714
random forest f1 score is : 0.7173592139089454
random forest recall score is : 0.721082846105061
random forest precision score is : 0.7334969674222925
random forest precision score is : 0.7334969674222925
random forest matthews_corrcoef score is : 0.6775581528698955

```

RANDOM FOREST CLASSIFIER USING TFIDF VECTORIZER

In [71]:

```

#fitting random forest using tfidf
rf_k.fit(X_train_tfl, y_train_tfl)

```

Out[71]:

```

RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                        criterion='gini', max_depth=None, max_features='auto',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=200,
                        n_jobs=None, oob_score=False, random_state=0, verbose=0,
                        warm_start=False)

```

In []:

```

#cross validation test on training set
cross_rf_tf = cross_val_score(estimator=rf_k, X=X_train_tfl, y=y_train_tfl, cv=10)
cross_bnk_tf

```

In [72]:

```

#prediction test
rf_pred_tf = rf_k.predict(X_test_tfl)
rf_pred_tf

```


Out[72]:

```
array([4, 7, 3, 5, 6, 2, 2, 1, 5, 4, 7, 4, 6, 2, 5, 6, 5, 6, 6, 4, 6, 7,
       7, 5, 1, 2, 1, 4, 5, 5, 1, 6, 2, 2, 7, 3, 2, 4, 2, 1, 4, 6, 6, 6,
       1, 6, 4, 4, 7, 6, 6, 1, 6, 3, 5, 7, 2, 1, 3, 4, 4, 4, 7, 1, 4, 1,
       2, 6, 6, 6, 5, 2, 3, 2, 6, 6, 2, 2, 2, 7, 6, 3, 3, 4, 4, 3, 3, 6,
       4, 3, 1, 6, 6, 1, 6, 6, 5, 1, 6, 7, 2, 7, 6, 7, 2, 4, 4, 1, 6, 2,
       5, 4, 7, 4, 6, 5, 3, 4, 5, 7, 5, 3, 5, 1, 2, 4, 2, 7, 5, 7, 1, 2,
       1, 1, 6, 6, 1, 2, 3, 3, 6, 3, 3, 5, 1, 4, 4, 6, 2, 2, 5, 1, 4, 5,
       6, 6, 4, 6, 4, 7, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 6,
       2, 3, 6, 3, 2, 6, 1, 7, 4, 6, 2, 4, 6, 4, 7, 1, 3, 2, 1, 7, 4, 2,
       6, 2, 5, 1, 7, 7, 4, 6, 2, 7, 6, 2, 1, 7, 5, 7, 6, 1, 3, 2, 6, 4,
       3, 4, 4, 2, 1, 6, 5, 7, 2, 6, 2, 1, 6, 6, 1, 7, 1, 4, 3, 1, 3, 5,
       4, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 2, 2, 7, 2, 3, 2,
       1, 3, 7, 5, 7, 3, 6, 1, 4, 1, 5, 4, 7, 6, 6, 6, 6, 2, 7, 7, 6, 3,
       3, 7, 2, 7, 3, 1, 6, 6, 1, 4, 2, 5, 1, 1, 5, 6, 5, 5, 4, 1, 6, 3,
       6, 3, 6, 5, 3, 1, 4, 3, 6, 7, 6, 2, 2, 6, 2, 1, 7, 6, 4, 6, 1, 7,
       4, 1, 4, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 6, 1, 2, 6, 4, 6, 6, 1, 5,
       5, 5, 3, 5, 1, 1, 6, 1, 6, 2, 1, 3, 6, 7, 6, 7, 2, 1, 7, 2, 1, 6,
       1, 1, 6, 1, 2, 4, 2, 1, 7, 4, 7, 6, 4, 7, 3, 2, 3, 4, 3, 3, 4, 5,
       4, 6, 3, 6, 5, 6, 3, 7, 6, 3, 2, 2, 7, 6, 7, 6, 2, 5, 6, 4, 7, 7,
       4, 2, 6, 6, 1, 4, 6, 4, 7, 7, 4, 2, 1, 3, 1, 1, 2, 6, 1, 5, 7, 6,
       6, 4, 5, 2, 2, 1, 3, 2, 6, 7, 3, 6, 2, 6, 7, 1, 3, 4, 6, 7, 2, 7,
       2, 1, 3, 2, 4, 3, 6, 7, 3, 1, 5, 2, 7, 4, 7, 1, 2, 1, 4, 5, 2, 2,
       6, 1, 2, 4, 3, 4, 6, 5, 2, 7, 5, 7, 2, 4, 2, 1, 2, 5, 1, 4, 7, 1,
       1, 2, 6, 2, 1, 2, 5, 1, 7, 4, 7, 7, 2, 5, 4, 1, 7, 6, 7, 4, 7, 3,
       3, 3, 3, 4, 2, 7, 5, 6, 3, 7, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
       7, 6, 3, 3, 2, 3, 6, 3, 7, 5, 4, 1, 2, 1, 3, 6, 7, 7, 7, 4, 3, 7,
       3, 3, 3, 3, 3, 3, 6, 6, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 7, 6, 2,
       5, 7, 1, 3, 1, 6, 6, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 2, 3, 6, 7, 7,
       1, 6, 3, 7, 3, 7, 4, 7, 4, 2, 2, 1, 6, 6, 1, 6, 1, 6, 1, 3, 1, 3,
       6, 4, 1, 2, 1, 5, 6, 6, 5, 3, 6, 3, 7, 7, 3, 7, 3, 5, 1, 7, 6, 1,
       2, 6, 6, 4, 6, 3, 3, 1, 3, 5, 1, 2, 6, 6, 7, 7, 5, 4, 7, 1, 6, 4,
       1, 1, 5, 4, 6, 1, 7, 1, 3, 3, 6, 5, 5, 5, 7, 2, 4, 2, 5, 3, 3, 2,
       3, 5, 3, 4, 1, 7, 4, 5, 2, 4, 4, 7, 7, 5, 7, 4, 6, 1, 7, 4, 1, 1,
       5, 6, 1, 1, 5, 1, 6, 3, 6, 4, 1, 4, 6, 2, 3, 5, 6, 4, 6, 7, 6, 1,
       1, 7, 6, 4, 5, 6, 7, 4, 3, 6, 2, 4, 1, 3, 6, 7, 4, 3, 2, 3, 1, 1,
       2, 4, 4, 5, 7, 6, 6, 5, 2, 3, 3, 1, 2, 6, 7, 1, 4, 4, 4, 4, 5, 7,
       5, 2, 7, 1, 2, 1, 3, 4, 6, 3, 6, 4, 2, 6, 6, 7, 2, 5, 1, 2, 7, 7,
       3, 2, 5, 4, 4, 7, 3, 2, 3, 1, 3, 2, 3, 2, 7, 7, 2, 6, 6, 5, 2, 4,
       4, 3, 3, 2, 5, 5, 2, 3, 1, 3, 7, 4, 5, 3, 5, 7, 7, 5, 5, 7, 1, 1,
       4, 5, 1, 7, 7, 6, 1, 3, 5, 2, 5, 3, 6, 3, 7, 3, 3, 7, 4, 1, 4, 2,
       6, 7, 7, 1, 5, 3, 7, 3, 6, 3, 6, 7, 6, 6, 2, 4, 2, 7, 5, 6, 7, 2,
       7, 1, 4, 1, 1, 2, 1, 3, 6, 2, 3, 3, 1, 6, 2, 6, 7, 1, 7, 1, 3, 6,
       5, 2, 4, 2, 2, 6, 3, 2, 6, 6, 2, 5, 5, 5, 2, 2, 6, 2, 6, 5, 6, 4,
       4, 3, 5, 1, 7, 3, 3, 6, 3, 5, 3, 6, 7, 6, 1, 3, 7, 6, 6, 2, 3, 6,
       3, 6, 4, 4, 6, 3, 4, 5, 6, 3, 3, 2])
```

In [73]:

```
#validation test
print("random forest Accuracy is:",metrics.accuracy_score(y_test_tf1, rf_pred_tf))
print("random forest f1 score is :",f1_score(y_test_tf1, rf_pred_tf, average='macro'))
print("random forest recall score is :",recall_score(y_test_tf1, rf_pred_tf, average='macro'))
print("random forest precision score is :",precision_score(y_test_tf1, rf_pred_tf, average='macro')
)
print("random forest precision score is :",precision_score(y_test_tf1, rf_pred_tf, average='macro')
)
print("random forest matthews_corrcoef score is :",matthews_corrcoef(y_test_tf1, rf_pred_tf))
```

```
random forest Accuracy is: 0.7418367346938776
random forest f1 score is : 0.7407928848451814
random forest recall score is : 0.742711403973978
random forest precision score is : 0.7510759877074175
random forest precision score is : 0.7510759877074175
random forest matthews_corrcoef score is : 0.7007738302494091
```

DECESSION TREE CLASSIFIER USING COUNT VECTORIZER

In [81]:

```
#fitting decession tree using
from sklearn.tree import DecisionTreeClassifier
```

```
dt_k = DecisionTreeClassifier()  
dt_k.fit(X_train_cv1,y_train_cv1)
```

Out[81]:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',  
                        max_depth=None, max_features=None, max_leaf_nodes=None,  
                        min_impurity_decrease=0.0, min_impurity_split=None,  
                        min_samples_leaf=1, min_samples_split=2,  
                        min_weight_fraction_leaf=0.0, presort='deprecated',  
                        random_state=None, splitter='best')
```

In []:

```
#cross validation test on training set  
cross_dt_cv = cross_val_score(estimator=dt_k, X=X_train_cv1, y=y_train_cv1, cv=10)  
cross_bnk_cv
```

In [82]:

```
#prediction test  
dt_pred_cv = dt_k.predict(X_test_cv1)  
dt_pred_cv
```

Out[82]:

```
array([4, 7, 3, 5, 6, 2, 5, 1, 5, 4, 7, 4, 4, 2, 2, 1, 5, 1, 6, 4, 6, 7,  
       7, 2, 1, 2, 1, 6, 2, 1, 1, 6, 2, 2, 7, 3, 2, 6, 2, 1, 3, 6, 4, 3,  
       1, 6, 4, 4, 6, 6, 6, 1, 1, 3, 5, 7, 2, 1, 3, 2, 6, 4, 7, 1, 4, 5,  
       3, 2, 6, 6, 5, 2, 3, 2, 6, 6, 2, 5, 2, 7, 6, 3, 3, 2, 4, 3, 3, 1,  
       4, 3, 1, 6, 6, 1, 6, 6, 5, 5, 4, 7, 3, 7, 6, 7, 2, 3, 6, 1, 6, 2,  
       5, 4, 7, 4, 6, 4, 3, 6, 5, 7, 5, 3, 5, 1, 2, 4, 2, 7, 5, 7, 1, 2,  
       1, 1, 6, 7, 1, 2, 3, 3, 6, 3, 3, 5, 1, 4, 4, 6, 7, 2, 1, 1, 4, 2,  
       6, 6, 2, 6, 5, 7, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 1,  
       2, 3, 7, 3, 2, 3, 1, 7, 4, 4, 2, 4, 6, 4, 6, 1, 3, 2, 6, 7, 3, 2,  
       6, 2, 5, 1, 7, 7, 4, 3, 2, 7, 6, 2, 5, 7, 5, 7, 4, 1, 3, 4, 6, 4,  
       3, 4, 4, 2, 1, 4, 2, 7, 2, 6, 3, 1, 6, 6, 1, 7, 3, 4, 3, 1, 3, 5,  
       3, 2, 2, 6, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,  
       5, 3, 6, 5, 7, 3, 6, 1, 4, 5, 5, 7, 6, 6, 6, 6, 6, 2, 6, 7, 6, 3,  
       3, 6, 2, 7, 3, 1, 6, 4, 7, 4, 2, 1, 1, 5, 3, 6, 5, 5, 4, 1, 6, 3,  
       6, 3, 6, 5, 3, 1, 2, 3, 2, 7, 6, 2, 2, 6, 2, 1, 7, 6, 4, 6, 1, 7,  
       6, 1, 4, 1, 7, 1, 3, 3, 6, 3, 6, 1, 3, 4, 1, 2, 2, 6, 7, 6, 1, 5,  
       5, 5, 3, 5, 5, 1, 6, 1, 6, 2, 1, 3, 4, 7, 6, 7, 2, 1, 7, 2, 1, 6,  
       5, 1, 7, 1, 2, 4, 2, 2, 7, 4, 7, 6, 7, 7, 3, 2, 3, 4, 3, 3, 4, 5,  
       4, 6, 3, 6, 2, 6, 3, 7, 1, 3, 2, 2, 7, 6, 5, 6, 2, 5, 6, 4, 6, 1,  
       4, 2, 6, 6, 1, 3, 6, 4, 7, 5, 4, 2, 1, 3, 1, 6, 2, 6, 6, 5, 7, 6,  
       6, 4, 5, 2, 2, 1, 3, 2, 6, 6, 3, 6, 2, 6, 7, 1, 3, 6, 6, 7, 2, 7,  
       2, 1, 3, 2, 4, 3, 6, 6, 3, 1, 5, 2, 7, 4, 2, 1, 2, 1, 4, 2, 2, 2,  
       5, 5, 2, 2, 3, 4, 1, 5, 2, 7, 5, 7, 3, 7, 2, 5, 2, 2, 7, 6, 7, 5,  
       1, 2, 6, 2, 1, 2, 2, 6, 7, 3, 7, 7, 2, 5, 4, 1, 7, 6, 3, 3, 7, 3,  
       3, 3, 3, 4, 2, 6, 5, 4, 3, 6, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 5, 3,  
       6, 1, 3, 3, 2, 3, 6, 3, 7, 5, 2, 1, 2, 1, 3, 6, 6, 7, 6, 4, 3, 5,  
       3, 3, 3, 3, 3, 6, 6, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 6, 7, 6, 5,  
       5, 7, 1, 3, 1, 6, 6, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 4, 3, 6, 6, 7,  
       1, 6, 3, 7, 3, 7, 4, 6, 3, 2, 2, 1, 6, 6, 1, 6, 1, 6, 1, 3, 5, 3,  
       6, 4, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 7, 3, 7, 3, 5, 1, 7, 6, 1,  
       2, 6, 6, 5, 6, 3, 3, 1, 3, 5, 1, 2, 3, 6, 7, 7, 5, 7, 7, 1, 7, 4,  
       1, 1, 1, 4, 4, 1, 7, 1, 3, 3, 6, 1, 5, 5, 7, 2, 6, 2, 5, 3, 3, 5,  
       3, 2, 3, 4, 5, 7, 4, 1, 4, 3, 4, 6, 7, 5, 7, 4, 6, 1, 7, 4, 5, 1,  
       5, 6, 1, 1, 5, 1, 7, 3, 6, 3, 1, 7, 4, 2, 3, 5, 6, 4, 6, 7, 6, 1,  
       1, 6, 6, 4, 5, 6, 5, 3, 3, 6, 2, 4, 1, 3, 4, 7, 4, 3, 2, 3, 1, 5,  
       3, 4, 3, 4, 7, 4, 6, 1, 2, 3, 3, 1, 2, 2, 7, 1, 4, 4, 5, 4, 1, 7,  
       5, 2, 7, 1, 5, 3, 3, 3, 4, 3, 4, 4, 2, 6, 2, 7, 2, 5, 1, 2, 6, 7,  
       3, 2, 5, 4, 4, 7, 3, 2, 3, 1, 3, 2, 3, 2, 7, 7, 3, 6, 6, 5, 2, 4,  
       4, 3, 3, 2, 5, 5, 2, 3, 1, 3, 6, 3, 5, 3, 5, 6, 7, 5, 5, 6, 1, 5,  
       2, 5, 2, 7, 7, 6, 1, 3, 5, 2, 5, 3, 6, 3, 7, 3, 3, 7, 4, 5, 4, 2,  
       7, 7, 7, 1, 5, 3, 6, 3, 6, 3, 7, 7, 6, 6, 2, 6, 2, 7, 5, 7, 6, 5,  
       7, 1, 1, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 3, 2, 2, 7, 1, 6, 6, 3, 6,  
       5, 2, 5, 2, 2, 6, 3, 2, 6, 6, 2, 1, 2, 5, 2, 2, 6, 2, 2, 5, 6, 6,  
       4, 3, 5, 1, 7, 3, 3, 7, 3, 2, 3, 6, 6, 4, 1, 3, 7, 6, 6, 2, 3, 3,  
       3, 6, 4, 4, 6, 3, 4, 5, 6, 3, 3, 5])
```

In [83]:

```
#validation test
print("decision tree Accuracy is:",metrics.accuracy_score(y_test_cv1, dt_pred_cv))
print("decision tree f1 score is :",f1_score(y_test_cv1, dt_pred_cv, average='macro'))
print("decision tree recall score is :",recall_score(y_test_cv1, dt_pred_cv, average='macro'))
print("decision tree precision score is :",precision_score(y_test_cv1, dt_pred_cv, average='macro'))
print("decision tree precision score is :",precision_score(y_test_cv1, dt_pred_cv, average='macro'))
print("decision tree matthews_corrcoef score is :",matthews_corrcoef(y_test_cv1, dt_pred_cv))
```

```
decision tree Accuracy is: 0.6989795918367347
decision tree f1 score is : 0.6972805364193234
decision tree recall score is : 0.6982727990814358
decision tree precision score is : 0.7087808464344663
decision tree precision score is : 0.7087808464344663
decision tree matthews_corrcoef score is : 0.650516121899878
```

DECESSION TREE CLASSIFIER USING TFIDF

In [84]:

```
#fitting decession tree using
from sklearn.tree import DecisionTreeClassifier
dt_k = DecisionTreeClassifier()
dt_k.fit(X_train_tfl,y_train_tfl)
```

Out[84]:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                        max_depth=None, max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort='deprecated',
                        random_state=None, splitter='best')
```

In []:

```
#cross validation test on training set
cross_dt_tf = cross_val_score(estimator=dt, X=X_train_tfl, y=y_train_tfl, cv=10)
cross_dt_tf
```

In [85]:

```
#prediction test
dt_pred_tf = dt_k.predict(X_test_tfl)
dt_pred_tf
```

Out[85]:

```
array([4, 7, 3, 5, 6, 2, 3, 1, 3, 4, 7, 4, 4, 2, 2, 2, 5, 6, 6, 4, 6, 6,
        7, 5, 1, 2, 1, 2, 4, 1, 1, 6, 5, 2, 7, 3, 2, 4, 2, 1, 4, 6, 6, 2,
        1, 7, 2, 4, 7, 6, 6, 1, 1, 2, 5, 7, 2, 1, 3, 4, 4, 5, 4, 1, 6, 1,
        4, 2, 6, 6, 5, 2, 3, 2, 6, 6, 2, 2, 2, 7, 6, 3, 3, 2, 4, 3, 3, 2,
        5, 3, 5, 6, 7, 4, 6, 6, 1, 5, 4, 6, 3, 7, 6, 7, 2, 4, 7, 1, 6, 2,
        5, 4, 7, 4, 6, 5, 3, 6, 5, 7, 5, 3, 6, 1, 2, 4, 2, 7, 5, 7, 1, 2,
        1, 1, 6, 1, 1, 2, 3, 3, 6, 3, 3, 5, 1, 4, 4, 6, 2, 2, 1, 1, 4, 2,
        7, 6, 2, 6, 5, 7, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 2,
        2, 3, 7, 3, 2, 2, 1, 7, 4, 6, 2, 4, 6, 4, 7, 1, 3, 2, 1, 7, 2, 2,
        6, 4, 5, 1, 6, 7, 4, 5, 2, 7, 6, 2, 5, 7, 5, 7, 6, 1, 3, 5, 7, 4,
        3, 4, 4, 2, 1, 7, 6, 7, 2, 6, 4, 1, 6, 6, 1, 7, 1, 4, 3, 1, 3, 5,
        4, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 1, 2, 7, 2, 3, 2,
        5, 3, 7, 5, 7, 5, 7, 1, 4, 5, 5, 4, 7, 6, 6, 6, 6, 2, 7, 7, 6, 2,
        3, 4, 2, 7, 3, 1, 6, 6, 1, 5, 2, 1, 1, 1, 4, 6, 5, 1, 4, 1, 6, 4,
        6, 3, 6, 5, 3, 1, 4, 3, 2, 7, 6, 2, 3, 6, 2, 2, 7, 7, 4, 6, 1, 7,
        7, 1, 4, 1, 7, 1, 3, 2, 6, 3, 6, 1, 3, 5, 1, 2, 6, 7, 6, 6, 1, 5,
        5, 1, 3, 5, 1, 1, 6, 1, 6, 2, 1, 3, 6, 6, 6, 6, 2, 1, 7, 2, 1, 6,
        1, 1, 1, 1, 2, 4, 2, 1, 7, 4, 7, 6, 6, 7, 3, 2, 3, 4, 3, 3, 2, 1,
        4, 6, 3, 6, 2, 1, 3, 7, 1, 3, 4, 2, 6, 6, 5, 6, 6, 5, 4, 4, 7, 1,
        4, 2, 6, 6, 1, 4, 6, 4, 7, 6, 4, 2, 1, 3, 5, 6, 2, 6, 1, 5, 7, 6,
        6, 4, 5, 2, 2, 1, 3, 2, 6, 6, 3, 6, 2, 6, 7, 1, 3, 4, 6, 6, 2, 7,
        2, 1, 3, 2, 4, 3, 5, 7, 3, 1, 5, 2, 7, 4, 5, 1, 2, 1, 4, 2, 2, 5,
        5, 5, 5, 4, 3, 4, 1, 5, 2, 7, 5, 7, 4, 4, 2, 5, 2, 5, 7, 7, 6, 5,
        5, 2, 6, 2, 1, 2, 2, 4, 7, 4, 7, 1, 2, 5, 4, 1, 7, 6, 4, 4, 7, 3])
```

```

3, 2, 0, 2, 1, 2, 2, 4, 1, 4, 1, 1, 2, 3, 4, 1, 1, 0, 4, 4, 1, 3,
3, 3, 4, 4, 2, 7, 5, 4, 3, 7, 5, 2, 6, 7, 4, 2, 6, 1, 1, 7, 2, 3,
6, 2, 4, 3, 2, 3, 6, 3, 7, 5, 4, 1, 2, 1, 3, 6, 7, 7, 7, 4, 3, 7,
3, 3, 3, 3, 3, 6, 6, 1, 1, 7, 6, 5, 1, 2, 3, 6, 6, 7, 7, 7, 2,
5, 5, 1, 3, 5, 6, 7, 6, 1, 7, 6, 1, 5, 6, 4, 1, 5, 2, 3, 6, 7, 7,
1, 6, 3, 7, 3, 6, 4, 7, 5, 2, 2, 1, 6, 6, 1, 6, 1, 2, 1, 2, 5, 3,
6, 4, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 7, 3, 7, 3, 5, 1, 7, 6, 1,
2, 7, 6, 7, 6, 3, 3, 1, 3, 5, 1, 2, 4, 6, 7, 4, 5, 4, 7, 1, 7, 4,
1, 1, 1, 3, 6, 1, 7, 1, 3, 3, 6, 1, 5, 5, 6, 2, 6, 2, 5, 3, 3, 5,
3, 7, 3, 4, 5, 7, 4, 5, 4, 5, 4, 7, 7, 5, 7, 1, 6, 1, 7, 4, 5, 1,
5, 4, 1, 1, 5, 1, 6, 3, 6, 5, 1, 7, 7, 2, 3, 5, 6, 4, 6, 6, 6, 1,
1, 6, 6, 4, 5, 6, 1, 4, 3, 7, 2, 4, 1, 3, 6, 7, 5, 3, 2, 3, 1, 5,
2, 6, 2, 4, 7, 5, 6, 1, 2, 3, 3, 1, 2, 5, 7, 1, 4, 4, 5, 4, 1, 7,
5, 2, 7, 1, 2, 4, 3, 5, 6, 3, 6, 4, 2, 6, 5, 7, 2, 5, 1, 2, 7, 7,
3, 2, 5, 4, 4, 2, 3, 2, 3, 1, 3, 6, 3, 2, 7, 7, 4, 6, 6, 5, 2, 4,
5, 3, 3, 2, 5, 5, 2, 3, 1, 3, 7, 4, 1, 3, 5, 7, 7, 5, 5, 7, 1, 5,
2, 5, 6, 7, 7, 6, 1, 3, 5, 6, 5, 3, 6, 3, 7, 3, 3, 7, 4, 1, 2, 2,
7, 6, 7, 1, 5, 3, 7, 3, 7, 3, 6, 7, 6, 6, 2, 4, 5, 1, 5, 6, 7, 2,
6, 1, 1, 1, 1, 2, 1, 3, 2, 4, 3, 3, 1, 4, 2, 6, 7, 1, 7, 1, 3, 7,
5, 2, 2, 2, 2, 7, 3, 2, 4, 6, 2, 5, 1, 5, 2, 2, 6, 2, 5, 5, 6, 4,
4, 3, 5, 1, 7, 3, 3, 7, 3, 2, 3, 6, 6, 6, 1, 3, 6, 6, 7, 2, 2, 5,
3, 4, 4, 4, 4, 3, 4, 5, 6, 3, 3, 2])

```

In [86]:

```

#validation test
print("decision tree Accuracy is:",metrics.accuracy_score(y_test_tfl, dt_pred_tf))
print("decision tree f1 score is :",f1_score(y_test_tfl, dt_pred_tf, average='macro'))
print("decision tree recall score is :",recall_score(y_test_tfl, dt_pred_tf, average='macro'))
print("decision tree precision score is :",precision_score(y_test_tfl, dt_pred_tf, average='macro'))
print("decision tree matthews_corrcoef score is :",matthews_corrcoef(y_test_tfl, dt_pred_tf))

```

```

decision tree Accuracy is: 0.7051020408163265
decision tree f1 score is : 0.7042438345228407
decision tree recall score is : 0.7046043951147852
decision tree precision score is : 0.7070546263274432
decision tree precision score is : 0.7070546263274432
decision tree matthews_corrcoef score is : 0.6561828341460413

```

LOGISTIC REGRESSION USING COUNT VECTORIZER

In [98]:

```

from sklearn.linear_model import LogisticRegression
lr_k = LogisticRegression(random_state=0,solver='lbfgs')
lr_k.fit(X_train_cv1,y_train_cv1)

```

Out[98]:

```

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=0, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)

```

In []:

```

#cross validation test on training set
cross_lr_cv = cross_val_score(estimator=lr_k, X=X_train_cv1, y=y_train_cv1, cv=10)
cross_lr_cv

```

In [99]:

```

#prediction test
lr_pred_cv = lr_k.predict(X_test_cv1)
lr_pred_cv

```

Out[99]:

```

array([4, 7, 3, 5, 6, 2, 3, 1, 5, 4, 7, 4, 6, 2, 1, 6, 5, 4, 6, 4, 7, 6,

```

```
array([[1, 1, 5, 5, 0, 2, 5, 1, 5, 4, 1, 4, 0, 2, 1, 0, 4, 0, 4, 1, 0,
       7, 5, 1, 2, 1, 4, 2, 1, 1, 7, 4, 2, 6, 3, 2, 6, 4, 1, 4, 7, 6, 6,
       1, 6, 4, 4, 7, 6, 7, 1, 7, 3, 5, 7, 2, 1, 3, 4, 4, 4, 6, 1, 4, 1,
       2, 6, 4, 6, 5, 2, 3, 6, 6, 6, 2, 2, 7, 6, 3, 3, 2, 4, 3, 3, 6,
       6, 3, 1, 6, 7, 4, 6, 6, 5, 5, 4, 7, 3, 7, 6, 7, 2, 4, 4, 5, 6, 2,
       5, 4, 7, 4, 6, 2, 3, 4, 5, 7, 5, 3, 2, 1, 2, 4, 2, 7, 5, 7, 1, 2,
       1, 5, 2, 2, 1, 2, 2, 3, 6, 3, 3, 5, 1, 4, 4, 6, 2, 2, 1, 1, 4, 2,
       6, 6, 2, 6, 5, 6, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 6,
       2, 3, 2, 3, 2, 6, 1, 7, 4, 6, 2, 4, 6, 4, 7, 1, 3, 2, 1, 7, 4, 2,
       2, 2, 5, 1, 7, 7, 4, 3, 2, 7, 6, 2, 5, 7, 5, 7, 4, 1, 3, 2, 7, 4,
       3, 4, 4, 2, 1, 6, 5, 7, 2, 6, 2, 1, 6, 4, 1, 7, 3, 4, 3, 6, 3, 5,
       3, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
       5, 3, 7, 5, 6, 3, 6, 1, 4, 5, 5, 5, 7, 6, 6, 6, 6, 2, 7, 7, 6, 3,
       3, 6, 2, 7, 3, 1, 6, 4, 1, 4, 2, 5, 1, 1, 4, 6, 5, 5, 4, 1, 6, 3,
       6, 3, 7, 5, 3, 1, 4, 3, 2, 7, 7, 2, 2, 6, 2, 1, 7, 6, 4, 3, 1, 7,
       4, 1, 4, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 4, 1, 2, 6, 4, 6, 6, 1, 5,
       5, 5, 3, 5, 1, 1, 6, 1, 7, 2, 1, 3, 4, 6, 6, 7, 2, 5, 6, 2, 1, 6,
       5, 1, 7, 1, 2, 4, 2, 1, 7, 4, 7, 6, 4, 7, 3, 2, 3, 4, 3, 3, 4, 5,
       4, 6, 3, 6, 5, 6, 3, 7, 5, 3, 2, 2, 6, 2, 2, 6, 2, 5, 6, 4, 7, 7,
       4, 2, 6, 6, 1, 4, 6, 4, 7, 5, 4, 2, 1, 3, 1, 4, 2, 6, 6, 5, 7, 4,
       6, 4, 5, 2, 2, 1, 3, 2, 6, 7, 3, 6, 2, 6, 7, 1, 3, 4, 6, 7, 2, 7,
       2, 1, 4, 2, 4, 3, 4, 7, 3, 1, 5, 5, 7, 4, 7, 1, 2, 1, 2, 6, 2, 2,
       5, 5, 2, 4, 4, 4, 6, 2, 2, 7, 5, 7, 4, 4, 2, 1, 2, 5, 1, 4, 7, 5,
       1, 2, 6, 2, 1, 2, 1, 6, 7, 3, 7, 2, 5, 4, 1, 7, 4, 3, 4, 7, 3,
       3, 3, 3, 4, 2, 6, 5, 6, 3, 3, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
       7, 2, 3, 3, 2, 3, 6, 3, 7, 5, 5, 1, 2, 1, 3, 6, 6, 7, 6, 4, 3, 7,
       3, 3, 3, 3, 3, 3, 6, 6, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 6, 1, 2,
       5, 6, 1, 4, 1, 6, 7, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 2, 3, 6, 4, 7,
       1, 6, 3, 7, 3, 7, 4, 7, 3, 2, 2, 1, 4, 6, 1, 6, 1, 6, 1, 3, 1, 3,
       6, 4, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 6, 3, 7, 3, 5, 1, 7, 6, 1,
       2, 6, 6, 5, 6, 3, 3, 1, 3, 5, 1, 2, 6, 6, 7, 7, 5, 4, 7, 1, 6, 4,
       1, 1, 1, 3, 6, 1, 7, 1, 3, 3, 6, 5, 5, 5, 7, 2, 6, 2, 5, 3, 3, 5,
       3, 2, 4, 4, 1, 7, 3, 5, 2, 4, 2, 7, 6, 5, 7, 4, 4, 6, 7, 4, 5, 1,
       5, 4, 1, 1, 5, 1, 6, 3, 6, 5, 1, 4, 4, 2, 3, 5, 6, 4, 6, 7, 6, 1,
       1, 7, 6, 4, 5, 6, 7, 4, 3, 3, 2, 4, 5, 3, 6, 7, 4, 3, 2, 3, 1, 5,
       2, 4, 4, 5, 6, 4, 6, 5, 2, 3, 3, 1, 2, 2, 7, 1, 4, 4, 5, 4, 5, 7,
       5, 2, 7, 1, 2, 3, 3, 3, 4, 3, 6, 4, 2, 6, 2, 7, 2, 5, 1, 2, 7, 7,
       3, 2, 5, 4, 4, 2, 3, 2, 3, 1, 3, 2, 3, 2, 7, 6, 2, 4, 6, 5, 2, 4,
       5, 3, 3, 2, 5, 5, 2, 3, 1, 3, 7, 3, 5, 3, 1, 7, 7, 5, 5, 6, 1, 5,
       4, 5, 6, 7, 7, 4, 1, 3, 5, 2, 5, 3, 6, 3, 7, 3, 3, 7, 4, 5, 2, 2,
       6, 7, 7, 1, 5, 3, 7, 3, 7, 3, 6, 7, 6, 4, 2, 4, 3, 7, 5, 6, 7, 1,
       7, 1, 5, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 3, 2, 6, 6, 1, 7, 1, 3, 6,
       5, 2, 5, 2, 2, 6, 3, 2, 6, 6, 2, 5, 5, 5, 2, 2, 6, 2, 2, 5, 6, 6,
       4, 3, 5, 1, 7, 3, 3, 4, 3, 5, 3, 6, 6, 6, 1, 3, 7, 7, 6, 6, 3, 6,
       3, 6, 4, 4, 6, 3, 6, 5, 5, 3, 3, 2]])
```

In [101]:

```
#validation test
print("logistic regression Accuracy is:",metrics.accuracy_score(y_test_cv1, lr_pred_cv))
print("logistic regreesion score is :",f1_score(y_test_cv1, lr_pred_cv, average='macro'))
print("logistic regreesion score is :",recall_score(y_test_cv1, lr_pred_cv, average='macro'))
print("logistic regreesion score is :",precision_score(y_test_cv1, lr_pred_cv, average='macro'))
print("logistic regreesion score is :",precision_score(y_test_cv1, lr_pred_cv, average='macro'))
print("logistic regreesion matthews_corrcoef score is :",matthews_corrcoef(y_test_cv1, lr_pred_cv))
```

```
logistic regression Accuracy is: 0.7744897959183673
logistic regreesion score is : 0.7746629350602935
logistic regreesion score is : 0.7747817147737331
logistic regreesion score is : 0.7812500491293566
logistic regreesion score is : 0.7812500491293566
logistic regreesion matthews_corrcoef score is : 0.7378630186890988
```

LOGESTIC REGRESSION USING TFIDF

In [102]:

```
from sklearn.linear_model import LogisticRegression
lr_k = LogisticRegression(random_state=0,solver='lbfgs')
lr_k.fit(X_train_tfl,y_train_tfl)
```

Out[102]:

Out[102]:

```
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=0, solver='lbfgs', tol=0.0001, verbose=0,
                    warm_start=False)
```

In []:

```
#cross validation test on training set
cross_lr_cv = cross_val_score(estimator=lr_k, X=X_train_tfl, y=y_train_tfl, cv=10)
cross_lr_cv
```

In [103]:

```
#prediction test
lr_pred_tf = lr_k.predict(X_test_tfl)
lr_pred_tf
```

Out[103]:

```
array([4, 7, 3, 5, 6, 2, 5, 1, 5, 4, 7, 4, 6, 2, 1, 1, 5, 6, 6, 4, 7, 6,
        7, 5, 1, 2, 1, 4, 4, 1, 1, 7, 4, 2, 7, 3, 2, 6, 4, 1, 4, 7, 6, 6,
        1, 6, 4, 4, 7, 6, 6, 1, 6, 3, 5, 7, 2, 1, 3, 4, 4, 4, 6, 1, 4, 1,
        2, 6, 4, 6, 5, 2, 3, 6, 6, 6, 2, 2, 2, 7, 6, 3, 3, 2, 4, 3, 3, 6,
        6, 3, 6, 7, 7, 4, 6, 6, 5, 1, 6, 7, 3, 7, 6, 7, 2, 4, 4, 5, 6, 2,
        5, 4, 7, 4, 6, 2, 3, 4, 5, 7, 5, 3, 2, 1, 2, 4, 2, 7, 5, 7, 7, 2,
        1, 6, 2, 6, 1, 2, 2, 3, 6, 3, 3, 5, 1, 4, 4, 6, 2, 2, 1, 1, 4, 2,
        6, 6, 2, 6, 4, 6, 4, 2, 1, 5, 6, 7, 1, 3, 6, 6, 6, 5, 7, 1, 6, 6,
        2, 3, 6, 3, 2, 6, 1, 7, 4, 6, 2, 4, 6, 4, 7, 1, 3, 2, 1, 7, 4, 2,
        2, 2, 5, 1, 7, 7, 4, 3, 2, 7, 6, 2, 1, 7, 1, 7, 6, 1, 3, 2, 7, 4,
        3, 4, 4, 2, 1, 6, 1, 7, 2, 6, 2, 1, 6, 4, 1, 7, 3, 4, 3, 1, 3, 1,
        3, 2, 2, 7, 5, 6, 3, 2, 5, 1, 3, 4, 2, 1, 2, 1, 5, 2, 7, 2, 3, 2,
        1, 3, 7, 5, 6, 3, 6, 1, 4, 5, 5, 4, 7, 6, 6, 6, 6, 2, 7, 7, 6, 3,
        3, 6, 2, 7, 3, 1, 6, 6, 1, 4, 2, 5, 1, 1, 5, 6, 5, 5, 4, 1, 6, 3,
        6, 3, 7, 5, 3, 1, 4, 3, 2, 7, 7, 2, 2, 7, 2, 1, 7, 6, 4, 3, 1, 7,
        4, 1, 4, 1, 7, 1, 3, 3, 4, 3, 6, 1, 3, 6, 1, 2, 6, 4, 6, 6, 1, 5,
        5, 5, 3, 5, 1, 1, 6, 1, 7, 2, 1, 3, 6, 6, 6, 7, 2, 5, 6, 2, 1, 6,
        6, 1, 7, 1, 2, 4, 2, 1, 7, 4, 7, 6, 4, 7, 3, 2, 6, 4, 3, 3, 4, 5,
        4, 6, 3, 6, 5, 6, 3, 7, 5, 3, 2, 2, 6, 6, 2, 7, 2, 5, 6, 4, 7, 7,
        4, 2, 6, 6, 1, 4, 6, 4, 7, 7, 4, 2, 1, 3, 1, 4, 2, 6, 6, 5, 7, 4,
        6, 4, 5, 2, 2, 1, 3, 2, 6, 7, 3, 6, 2, 6, 7, 1, 3, 4, 6, 7, 2, 7,
        2, 1, 4, 2, 4, 3, 6, 7, 3, 1, 5, 5, 7, 4, 7, 1, 2, 1, 2, 6, 2, 2,
        2, 1, 2, 4, 4, 4, 6, 2, 2, 7, 5, 7, 4, 4, 2, 1, 2, 5, 1, 4, 7, 5,
        1, 2, 6, 2, 1, 5, 1, 6, 7, 4, 7, 7, 2, 5, 4, 1, 7, 4, 7, 4, 7, 3,
        3, 3, 3, 4, 2, 6, 5, 6, 3, 3, 5, 2, 7, 7, 4, 2, 6, 1, 1, 7, 2, 3,
        7, 6, 3, 3, 2, 3, 6, 3, 7, 5, 5, 1, 2, 1, 3, 6, 7, 7, 7, 4, 3, 7,
        3, 3, 3, 3, 3, 6, 7, 1, 1, 6, 6, 4, 1, 2, 3, 6, 6, 7, 7, 6, 2,
        5, 6, 1, 4, 1, 6, 7, 6, 1, 7, 6, 1, 5, 7, 4, 1, 5, 6, 3, 6, 7, 7,
        1, 6, 3, 7, 3, 7, 4, 7, 3, 2, 2, 1, 6, 6, 1, 6, 1, 6, 5, 3, 1, 3,
        6, 4, 1, 2, 1, 5, 6, 2, 5, 3, 6, 3, 7, 6, 3, 7, 3, 5, 1, 7, 6, 1,
        2, 6, 6, 7, 6, 3, 3, 1, 3, 5, 1, 2, 6, 6, 7, 7, 5, 4, 7, 1, 6, 4,
        1, 1, 5, 4, 6, 1, 7, 1, 3, 3, 6, 5, 5, 7, 2, 6, 2, 5, 3, 3, 5,
        3, 2, 4, 4, 1, 7, 3, 5, 2, 4, 2, 7, 6, 5, 7, 4, 4, 6, 7, 4, 5, 1,
        5, 4, 1, 1, 5, 1, 6, 3, 6, 5, 1, 4, 4, 2, 3, 5, 6, 4, 6, 7, 6, 1,
        1, 7, 6, 4, 5, 6, 7, 4, 3, 4, 2, 4, 5, 3, 6, 7, 4, 3, 2, 3, 1, 5,
        2, 4, 4, 5, 7, 4, 6, 5, 2, 3, 3, 1, 2, 2, 7, 1, 4, 4, 5, 4, 5, 7,
        5, 2, 7, 1, 2, 6, 3, 3, 4, 3, 6, 4, 2, 6, 2, 7, 2, 5, 1, 2, 7, 7,
        3, 2, 5, 4, 4, 2, 3, 2, 3, 1, 3, 2, 3, 2, 7, 6, 2, 4, 6, 5, 2, 4,
        5, 3, 3, 2, 5, 5, 2, 3, 1, 3, 7, 3, 5, 3, 1, 7, 7, 5, 5, 6, 1, 1,
        4, 5, 6, 7, 7, 6, 1, 3, 5, 6, 1, 3, 6, 3, 7, 3, 3, 7, 4, 5, 2, 2,
        6, 7, 7, 1, 5, 3, 7, 3, 7, 3, 6, 7, 6, 4, 2, 4, 3, 1, 5, 6, 7, 1,
        7, 1, 5, 1, 1, 2, 1, 3, 3, 2, 3, 3, 1, 6, 2, 6, 7, 1, 7, 5, 3, 6,
        5, 2, 4, 2, 2, 6, 3, 2, 6, 6, 2, 5, 5, 5, 2, 2, 6, 2, 4, 5, 6, 6,
        4, 3, 5, 1, 7, 3, 3, 6, 3, 5, 3, 6, 6, 1, 6, 7, 7, 6, 6, 3, 6,
        3, 6, 4, 4, 6, 3, 6, 5, 6, 3, 3, 2]])
```

In [104]:

```
#validation test
print("logistic regression Accuracy is:",metrics.accuracy_score(y_test_tfl, lr_pred_tf))
print("logistic regreesion score is :",f1_score(y_test_tfl, lr_pred_tf, average='macro'))
print("logistic regreesion score is :",recall_score(y_test_tfl, lr_pred_tf, average='macro'))
print("logistic regression score is :",precision_score(y_test_tfl, lr_pred_tf, average='macro'))
```

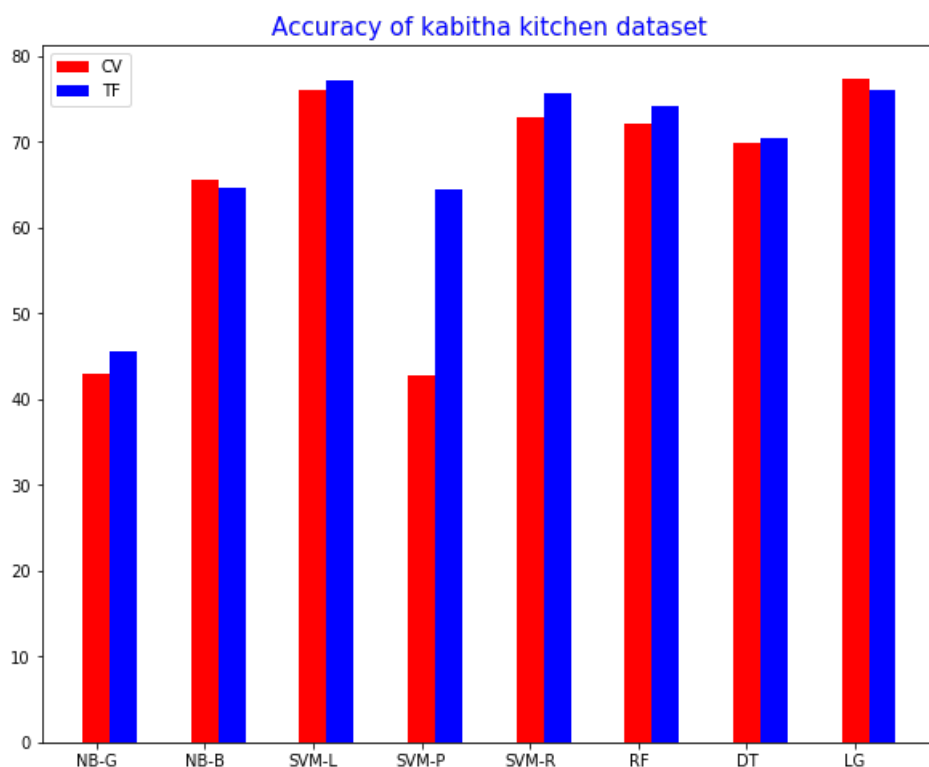
```
print("logistic regreesion score is :",precision_score(y_test_tf1, lr_pred_tf, average='macro'))
print("logistic regreesion score is :",precision_score(y_test_tf1, lr_pred_tf, average='macro'))
print("logistic regreesion matthews_corrcoef score is :",matthews_corrcoef(y_test_tf1, lr_pred_tf)
)
```

```
logistic regression Accuracy is: 0.7602040816326531
logistic regression score is : 0.7602343401282932
logistic regression score is : 0.7606392388277207
logistic regression score is : 0.7700553136566202
logistic regression score is : 0.7700553136566202
logistic regression matthews_corrcoef score is : 0.7218554248770348
```

Visualizing the accuracies of kabitha kitchen data set

In [17]:

```
# We impletement bar graph using pyplot from matplotlib library for visualization
# Here the visualization shows the accuracies of the machine learning model applied with both coun
t vectorizer and tfidf vectorizer
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(10,8))
X= ['CV','TF']
X= np.arange(8)
ACCURACY = [[43.0,65.6,76.1,42.7,72.8,72.1,69.8,77.4],[45.6,64.6,77.2,64.5,75.6,74.1,70.5,76.0]]
plt.bar(X+0.00,ACCURACY[0], color='r',width=0.25,label="CV")
plt.bar(X+0.25,ACCURACY[1], color='b',width=0.25,label="TF")
plt.xticks(X,["NB-G","NB-B","SVM-L","SVM-P","SVM-R","RF","DT","LG"])
plt.legend(loc="upper left")
plt.title("Accuracy of kabitha kitchen dataset",{ "size":15,"color":"blue"})
plt.show()
```



In []:

SENTIMENTAL ANALYSIS ON NISHA KITCHEN

In [17]:

```
dataset_n = pd.read_csv("Nishafin.csv", encoding='latin-1') #reading nisha kitchen dataset
dataset_n.head()
```

Out[17]:

	id	commentText	Labels
0	Ugz7LQV0ZWpuNnjBaxJ4AaABAg	i love potato ke prathe	2
1	UgxvWJ_kJL2nePTkg2N4AaABAg	Wwaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaww	3
2	UgxxeUmiH9f8XOZZWr14AaABAg	Nice	3
3	Ugz-UPvwHMx4nSKVoCB4AaABAg	Wow Amazing	3
4	UgzfXhbesBABQzi4-m94AaABAg	Thank you madam	1

CLEANING ON NISHA KITCHEN DATASET

In [18]:

```
# As we can see from the above display even in nisha dataset punctuations,upper cases etc are present
corpus = [] #creating corpus for nisha kitchen dataset
for i in range(0,4900):
    review = re.sub('[^a-zA-Z]', ' ', dataset_n['commentText'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)
```

In [19]:

```
#creating the count vectorizer
from sklearn.feature_extraction.text import CountVectorizer
cv_n = CountVectorizer()
Xn_c = cv_n.fit_transform(corpus).toarray()
yn_c = dataset_n.iloc[:,2].values
```

In [20]:

```
print(Xn_c.shape)
```

(4900, 4483)

In [21]:

```
corpus
```

Out[21]:

```
['love potato ke prath',
 'wwaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaww',
 'nice',
 'wow amaz',
 'thank madam',
 'nice',
 'nice',
 'exocl women',
 'soo kind',
 'believ million sub',
 'nice',
 'thank',
 'atta flour mean wheat flour',
 'paratha like kitchen design',
 'tasti aloo paratha',
 'thank aunti',
 'superb',
 'aunti aapn bahut hi acha aur bqhut simplic ke saath explain kiya ais hamar desh teacher ho jay p
 hir kisi bacch ko tution ki need ni ho u inspir paratha also good healthi',
 'bestwest world west']
```


'instruct realli use',
'ur way explain best aunti ji',
'love word thodi se chotti thodi se badi bana sakti hyn',
'nisha aunti sooooooooo humbl sweet keep',
'wao',
'thank',
'awesom',
'app mera bhahi hote toh bariya hota roj kuch na kuch naya khane ko milta',
'nice',
'hello nisha anti mai new hun bht mazi k pharahti bane muje se ap ki video dhek kr thank god muje
repli kiji ga nd main kuch bana hu kia likhun',
'beshk u r great bless frm pakistan',
'madam aap thoda nail cut karo suggest hai mera cook kart time warna glove wear karo',
'wow',
'love kerala',
'bhot asha aalo vala prantha',
'vedio bohot accha laga mam',
'nice recip easi cook',
'yummi yummi super yummi aaloo prath',
'mam mere potato ko mne boil krke cooker mein chod diy hour mne potato ko cheel ke masala banaya
masala kafi loos ho gya usk baad mne unh choka bhi laga diya phir bhi parath nhi bn rhe h pl batay
kya kre',
'yummi',
'still young due cook',
'nice madam ji',
'mam u highli talent ur cook skill extraordinari',
'good aunti',
'voic nice',
'nice',
'explain clearli sweetli thank',
'mam jab paratha belt hai tao sara masala corner pe aa jata h center kuch nhi rahta',
'also paratha receip nice nice',
'nisha mam tri n good make paratha di way kid week n dey enjoy u show awesom recip god bless u n
ur love one',
'thank mam easi yummi',
'wow wow wow mouth water',
'mom want make aloo ka paratha vid help thank',
'madam oil ka place ghee use kar sakth kya',
'delici',
'good nice recip mam',
'thank aap itna achha paratha banai hai aur aap bahut acchi hai thank',
'itna oil',
'osm pratha aunti ji',
'tri pleas one like',
'wow apk parath kitt kitt mast h wow mugh abi khane ka mn kr rhe h mae bi aj hi bnaungi',
'thank',
'realli recip testi ur big fan paratha recip also nice',
'majedar bana hai aloo paratha',
'newli mrd dono anythng cook hubbi lyk alu paratha tqsm mam fr di recip',
'nyc',
'women good',
'good pratha ji',
'thanku mam',
'nice holiday cook grandson im also cook expert wish',
'thank mam bht testi hai',
'nice bhabi ji',
'mam aap bahut ach tarik se sikhati h thank',
'thanku much',
'tri earli morn',
'nice aunti ji',
'thank tri tasti',
'thank mummi ji',
'aunti jokerfrom kashmir want ask use gram flour besan instead wheat flour maida',
'nice parath',
'super mam',
'koi new recip dijiy nisha ji',
'thank',
'yummmmmmmmi abi tri krni lgii hu',
'nice recip aap ka andaz itna pyara samjhan ka ke kia hi baat thank recip',
'yammyyyyyyyyyyyyyyyyyy',
'realli nice paratha',
'tri',
'marshallah beauti excel parathay',
'mam koi bhi stuf paratha bnati hu sari stuf kinaro p aa jati h beech sirf aata reh jata h kya kr
u ki stuf pure parath evenli lg jay',
'aaj bhi ese tri karugi mujh aloo ke parath banana aat hi nhi thank much mam',
'nice thank share',
'

'superb aunti ji aaps bahut sikhn ko milta hai ab main bhi aapka wajah se achi paranthi banani legi hoon thankyou aunti ji',
'great thank lot yummi paratha',
'v nice',
'thank tri paratha today succeed repli post',
'tasti also sweet aunti',
'superb mam',
'badhia mam apka express karn ka tarike kafi boor nahi hai ap bahut achi ho mam pyar se patience ke saath samajhati',
'wow thank u',
'good tasti bhi healthy bhi',
'mai kal banaungi aaloo ke parath',
'muje ba aap pasand ho',
'really tasti',
'really love',
'apko dekh k ki yad aa jati hai',
'nice recip mam love',
'nice',
'happy new year aunti',
'madam shall pour oil first pan put paranthi oil splash',
'wahhi yammi yammi',
'tasti parathaparath',
'nice yummi',
'thank much',
'wow',
'amaz tri tasti',
'yummi',
'nice',
'bot ganda',
'kya partha',
'nice',
'apki awaj jaan nikal leti bahut khub samajhati hai cute',
'nisha madhulika top women',
'wowow',
'auntii good talent',
'nice',
'thank mr singh show make kolkata really help',
'easy recip tri home thank',
'nice',
'love paratha love woman love pakistan',
'today eat paratha friend india they make love',
'aapki hr recip must rehti hai',
'auntii yummi aloo paratha',
'mam love way u narrate voice sweet n way talk calm',
'tasti',
'thank mam',
'nice way make aunti',
'nice mam',
'garlic dal orr bhi tasti banta',
'tasti',
'nisha jee r best chef really like u from pakistan',
'tasti',
'mam hamare parath karte hai kya karte ki fate nahi',
'nice',
'wow amaz learn paratha recipe',
'aloo ke parath mujh bahut bahut pasand hai aap mujh bhej na',
'hi nisha mam please make paratha curri recip early possible',
'aloo fry krke b parath banate hai plz uski recip bhi share kariy',
'auntii ji khane ka man kya',
'thank u mam',
'nice paratha sooo yummi explain karn ka tharika hohat achi superb',
'nice paratha',
'welcome mam',
'kuch dikh nahi raha',
'wow mam kya aapni parth banay hai',
'nice',
'awww seem like sweet ladi subscribe',
'mam intellig video good',
'bohot bohot shukriya auntii jee aapka pyar se batane ka tarika bhi bohot achi hai',
'hi nishaji love ur video n way u explain load love keep help us cook variety dish tri aloo parath n turn yummi thank lot',
'pata nahi kyo lekin jab bhi aapki video dekhta hu eisa feel hota hai meri maa mujh samjha rahi hai thank lot voice attract',
'thank auntii',
'brilliant yummi',
'nisha ji aap bhut acchi tarhan explain karte ho nice recipe',
'really like ur video inspire u ill try ur recip love u mam respect',

'tri ur recip came super thank u much learnt mani dish u pleas finger papad ingredi make post ur come video pleas',
'nice paratha',
'wahhhh aunti kya banaya paratha tasti lajawaab muh pani aa gaya kya bolu kash ye paratha bahar a a jata banaungi aaj hi nahi raha jayega',
'like ur recip n also tri home love u maam shahina frm new zealand',
'tasti aunti',
'talent thank alot mam',
'tadka lga kr bhi hum stuf bna sact h',
'nice mam bahut hi acch bataya h aapn thank',
'mam like way u cook easi n great tast',
'mam aap kon sa tail use krte hai pranth banan ke liy',
'kuch new recip batoo na nisha ji big fan ap ke raspic tri kart bhoot accha banta samosa aloo ke taki allmost cook nisha ji imran khan hasband citi nagpur',
'alu hre hai',
'wao mam teasti jarur tri krungi thanku mam',
'thnx favourit recip',
'nice',
'super madam watch smusic vaibhav puja marathi',
'super mam',
'nisha ji ke raspic bhoot pasand aat hai love cook nisha ji name mr nusrath khan nagpur citi',
'sweet',
'done difficult receip easi way awesom',
'mam usual vvv nici recip love alu paratha',
'ur explain clear love ur video mam',
'would say thank mam good cook',
'whenev want cook someth search youtube channel first reason voic nice dnt irrit like other youtube channel second way teach cook make feel easi learn',
'aaloo pratha',
'tasti',
'theek h esh hi banaugi',
'good method thank mam',
'nice paratha',
'pleas mam share diet food receip',
'bahut achh bne mere prath',
'hello mam love much',
'wow testi yummi yummi thanku mam tri',
'thank mam u sweet love',
'yummi paratha',
'nice',
'super',
'want ask question alway thought appropri resist today want ask whether use dishwash clean utensil coz look new',
'aap k samjhan ka tarika bhut acha h',
'wah nisha aunti love frm pakistan',
'testi',
'love u nd ur recipi aunti',
'nice aunti',
'mam recip good cook way easi aapki help se b kitchen bahut kuchh bnane lgi hu thank much mam biggest fan',
'look yummi',
'super super',
'excel mam thank much',
'thank mam nice recip',
'ak tumhari neck itni hil rahi hoti hai k mera dheyan tumhari neck p hota hai',
'kya aat sabhi aalo masalo koo mix kr skte h kya plz repli plzzzzzzzzzzzzzzzzzzzzzz',
'bohat tasti dikh rahah mean fantast',
'sweet',
'thnx mam aap repli karn bhi bahut fast ho',
'kya mam garam masala use kar skte h',
'hi kya hum purpos flour use kar sakt hai',
'wow soo tast',
'nice',
'nice',
'madam u explain softli sweetli like ur paratha',
'hello mam aapka har recip bahut testi hota h apka har recip follow krti hu mere husband toh bahut khush ho jaat h mere se thank mam',
'super aunti',
'nice',
'nice',
'well explain wish cook like',
'aapk hr video dekhti hu aap bhut ach ho',
'tasti',
'one like talent anti',
'nice made anti',
'buhat acha sikhati hayn',
'bahut tasti bahut mast tha ghar sbko bahut bahut pasand aaya bahut tasti tha',

'thank aunti recepi',
'bahot bhadiya bana h mam',
'thank',
'waooo yummi mouth water',
'aap mujh bhot achi lagti hain bhot meetha bolti hain aap aur apki recip jabs apka channel bana t
abs dekhrahi hun itna kuch seekha aaps god bless',
'maam sweet',
'thank mam',
'lv ur smile nd voic mam lv u alot',
'mama mia',
'super tq',
'aap etn acch kais bol lete hain aap ki present se hamara dil paratha se vi bhar gay hain unti th
ank u',
'tnx nisha madulika',
'ne kal ye parata banaya ta buht hi yummi bana ta',
'best chef',
'yummmmmmmmmmmmmmmmmmmmmmm',
'last modifi method',
'thank one english subtitl look nice',
'hmmmm look yummi',
'nice',
'grate potato mam',
'humain baji nisha madhulika',
'tasti',
'thank lot mam',
'thank u mam',
'aapki btae recip simpal aur bohat jada testi hoti hai',
'wow mam amaz yummi love kashmir',
'nice mam',
'app mam ho',
'realli amaz awesom superrbbb mne avi ye recip follow krke bnaya realli tasti amaz thank lot maam
,
'thank',
'dislik',
'nice mam',
'nice',
'u r awesom mam jab b kch bnana hota h sbse phle apki recip hi search krti hu',
'mehr bani aba',
'nice',
'inh bhee chalaate k beech bhark banaa sakt hai madam ji',
'nice one mam',
'respect everi mother',
'aapki har recepi mujh aacha lagti h nisha g aur main baat ye h ki aap bhot bariki tarika se samj
hati wo aur jyada aacha lagta r cute',
'nice tasti',
'thank mam',
'bhai kon lucki h wo jiski mummi h',
'aapn bahut acch se smjhaya thank mam',
'thank mam',
'nice',
'final made yaayyyyyi thank mam love yummi',
'thank much',
'yummi',
'wow nice mam',
'kya stafe magi masala dal sakt',
'goa aunti love way u explain make paroatha love u aunti',
'hii aunti kya ham ginger aur mango powder ke bina nahi banasakt kya',
'aanti aap bhut aacha resapi bant ho',
'love u mam',
'beauti kitchen wowwwwwwwwwwwwwwwww never seen beauti kitchen',
'thank made ate yummi',
'ap bahut sweet bolt ho ji',
'yor recip great',
'super aunti maja aa gya easi aacha partaha',
'nice anti ji',
'thank mam parath ekdm zabardust',
'god bless',
'anti bahot door hn nhn mn ap k pa zaroor khanay ko',
'mera aalu paratha fat kyu jata masal bhar nikl jata',
'love way teach',
'yummi',
'yummi',
'nice',
'best chef',
'r great',
'mere aloo ke parath full jaat hain aisa kya kru ji se paratha na pate',
'use recip make paratha mother birthday love thank wonder video',

'nice',
'nyc mem',
'mam aap bahut acchii tarah se samjhaat hoo har recepi ko acch se samjh aati h sabhi recepi thank u much mam itni acchi acchi recepi ki video banan k liy',
'ism aloo k sath pyaz ka use nahi kar skte hai kua',
'thanq nisha madhulika ji tri n mere parath bahut ach ayy ghee r yummyy',
'nice video',
'awesom recip also made feel hungri lazi',
'damn girl famou',
'thank put subtitl video definit tri recip glad share amaz recip us',
'nice mam ji',
'nice',
'much nice mam yummi',
'look tasti',
'mam mai jb bi alo paratha banati hu paratha fat jata h na alo bahar nikln lgta h kbi kbi kbi ach h se ban jata h',
'nyc anti iov voic',
'awesom mamm',
'mam mere pranth mein aaloo evenli nhi aata husband say kai kai jagah pranth mein aaloo nhi hota',
,
'nice mam',
'nice',
'tasti',
'experi amaz littl brother sister happi eat',
'tasti recip mam delici',
'recip easi tri make today first time thank share recip',
'nice explain ador passion say word hat nisha',
'soo nice mam',
'nice recip',
'meri mamiy ne banay h bahut tasti bane h',
'nice janab',
'aapn bahut achha banaya hai super mame',
'mam ap mom jese ho bilkul',
'tast prantha mam',
'nice parantha eat',
'osm recip aalu pratha favourit',
'madam kya aap pakistani ho',
'super maaaji',
'talent women n cute nice explain',
'fantast',
'nice paratha mam',
'tri stuf get make paratha',
'osummmmm yummi recip',
'kashala keli',
'realli appreci love turkey',
'nice',
'nice',
'look tasti',
'everi time make stuf bleed thru tri th time',
'nice',
'thank aunti cute way explain',
'nice pratha mem',
'nice mam explain good',
'hi pleas one day also publish prepar chicken chilli mutton korma pleas',
'mera aloo ka paratha fat jata hai stuf puri bahar nikal jati hai kya karna chahiy',
'waww wasom',
'nice',
'iam big fan mam lot love dubai',
'nisha mam recip great recip world tasti recip',
'tooo good n nice explain',
'tasti aap aloo ka paratha banati ho',
'wow nisha apki recip ghar ki yaad dila dete h nice recip',
'nice wow',
'love allo ka partha love pakistan',
'aunti u r sweet nd love nd love ur recip',
'sooooo yummi',
'thank u soo much love bangladesh',
'nisha ji aap ki cook mjhi bht pasand hy',
'explain well love mam soooo cute',
'aloo paratha',
'mam ap aur ap ki recip duno lajawab h',
'dayum gotta tri look delish',
'love aunti',
'wooooooooooooooooooooo',
'nani nice',
'nice one question pratha low ya high kais flame pr sekna chahiy',
'thakuu soo mch mam u r talent',

'thank u mam',
'aap bahut ach se samghaya thank',
'nice',
'nice recip',
'good',
'mei bhi hr baaar ais hi prath bnati hu jo ki mujh aapn sikha unti g',
'thank mem',
'nice mam',
'madam well keep go new recip thank kind inform',
'nice pratha',
'yeh bohot aasan reciepi hai main bohot dino se banay nahi hai aj apki recip dekh ke banaugi ise
kal thank mam',
'tri trust soooo yummi',
'thanku aunti mujh aap ki rashpi achi lagh ti h',
'wowwwwwwww nisha jii',
'nice',
'wow tri make today husband love',
'nice guider',
'thank much love',
'thank love',
'nice man apki recip bhi supper h',
'nishji ki recip alway best',
'aapko mai apni maa manti hu main aap se hi pura khana banana sikha hai thank mch n god bless',
'nisha ji aap ki recip suopr het',
'mam like ur explain style',
'nice paratha',
'love',
'tasti',
'wow nice recip',
'samjhat bohot atch ho aap receipi ko',
'thenku mem',
'nice',
'aamchur powder nahi ho kya dal sakat hai',
'mam haman aj hi banak dekha bahut hi achaa tha realli mam nice paratha love mam',
'thanku mam glad saw video highli appreci nd well explain u make day hv tri ur style amaz',
'aap ki har recip achi hoti hai aur easili bana de sakt hain',
'nice mam love',
'thank u mam main pahli bar banaya h bhut acha bna thank u soooooo much',
'thank mam behalf mum bottom heart',
'mam aapki aloo ke parath ki recip bahut achhi thank share recip',
'nyyccc',
'ok thank mam',
'nice aaloo parath nisha madhulika',
'look delici go make today',
'nisha ji aap ko bohot bohot dhanyavaad aap se seekhi hui har ek recip se banai huyi khana
hamesha super hit hoti hai thank much',
'mugh issm carrier banana h',
'nice pratha mam yummi',
'mugh khana bana baut jada accha lagta h',
'mam mugh apki help chiy',
'yammi',
'good mam',
'nice',
'yummi',
'ara best chef',
'nice',
'like ur everi recepi made u',
'thank mam',
'hum dekht hi bhook lag gayi man kar raha hai abhi hi toot pado thank much nishamadhulikji thank
thank thank much love',
'nice',
'thank simpl video mine came fair fine tast great ty',
'bhtut acha samjhati hai aap mam',
'explain super thank much mam',
'aapn aach se explain kiya thank u',
'tq nisha jii',
'wish could eat paratha onlin',
'good',
'mem agr alu pani chod jaye ptle ho jaye kya kre',
'aloo paratha super',
'aaj aalu k parath bnay aapk recip k accord ymmmi schi',
'nice',
'nice',
'wow mam',
'skip dri mango powder green chilli',
'im tri today st time',
'v nyc pratha',

'nice',
'nisha ji kya hum aloo ki stuf ko fri kark bhi bana sakt hain',
'love',
'amchur powder use lemon instead',
'beauti real mother hand',
'nice mam bhot testi bne path',
'nice tasti',
'hiiii',
'mam mere paratha line aa jati h',
'nisha ji u pleas make stuf egg paratha delhi style',
'alway love see feel like u r famili member u r sweet',
'kitchen amaz',
'good job aunti',
'shabash aagay bado',
'thank aunti help project',
'salut talent chef present entir indian home perfect round paratha easili made',
'ooo wao tasti paratha',
'thank',
'madam mera har bar paratha bele ne se pura fata jati issk liy kya karen',
'delici wud tri raini weather',
'thank nisha tayi',
'ap samjat bhut bhadiya h nice',
'nice mam',
'nice',
'mam nice paratha mene aaj tri kiya',
'good afternoon nice recip tri hope kid would like',
'nice g',
'yummi',
'nisha ji yami',
'use onion recip',
'thanx mam love',
'nice mam thank share recip aloo paratha',
'jitni achchi aapki recip banti hai utni hi pyari aap hai aap muze apni didi si pyari lagti hai a
ap itn pyar se batati hai ki har recip achchi banti hai hams love u',
'nisha ji whenev make paratha stuf goe side n middl part get empti hv tri mani way problem occur'
,
'best',
'thank mom',
'nice tasti',
'mem muje apka har recepi bhohat axa lgta hai mene aapka recepi banai hai shobko bohat axi lge th
ank mem',
'waaooo yammmi mam love',
'thank nishaji share follow best aloo paratha life today',
'thank give everi detail',
'aap bahut achi baatein kart hai thank aunti',
'nice',
'thanx',
'best recip',
'tri make aloo paratha mani time never got right way easi understand prepar yummi thank much god
bless',
'ooh yaaamee',
'aunti maja aa gaya banak khake aaya comment likn videsh se dhanyawaad',
'awsom madam',
'mam pleas mirchi ka paratha',
'mam love u',
'mam explain receip beauti bodi cook tri recip potato boil yet well thank',
'easi turori lot tip look delici',
'naic masa alla allo parata thank u racepi sere love u',
'thank antiji',
'nice parath mam',
'nice pratha',
'hi made allu paratha recip today turn great',
'kashturi methi aata gutht samay daal sakt hai kya mam',
'thank',
'nyc mam',
'thx mam',
'resipi best hi di yai',
'tasti',
'give coriand chilli',
'miss aap bahut ach se samaj ho g',
'great',
'wow mam apn boht ach se samjhaya aj hi bnaugi ap boht sweet ho',
'best love cook madam',
'nice video',
'thank lot shear favorit paratha',
'use besan gram flour',
'good madm',

'love receip made turn amaz',
'nisha u r sweet u explain nice love recip',
'thank mam ek aur request thi mam baccho ke tiffin ke liy koi asaan si resip btaiy jo asaani se j
hat pat ban jay aur tast healthi ho',
'u r mentor nisha happi mother day u husband love paratha learnt u thank u much love see ur video
,
'hum aaj hi tri kareng',
'fan hogaya assa haa',
'superb',
'thank much',
'one favorit thank',
'nice mam main tri kiya bahut acha bana thank',
'thanku',
'tast year',
'aap baut cute ho ek dum advic life de di apnay',
'nice',
'bhook lag gaya',
'love voic dd',
'aunti g kabhi khila bhi dijiy u r great',
'good',
'nyc',
'nice mam',
'nice inform',
'tasti mam thank share method',
'super aunti',
'wow',
'thank unti',
'mam dahi ke saath paratha accha lagega ya dahi bhi kuch milana hai',
'nisha ji plz malai paratha banana bataiy',
'lvu maaji',
'nice mam',
'thank mam favourit dish',
'masha allah today tri recip yammi tasti thank allot share us new recip thank much',
'mam u r best love soo much',
'nice',
'thank share good recip',
'hello must much love children speek sweetli children lucki mother like lot respect raja',
'nice',
'like potato paratha',
'wow',
'nice',
'love favorit food',
'pleas tell recip pyaj ke oarath',
'good',
'yummi parantha',
'dear extra sweet',
'mam mere aloo toh andar se sukh mere yaha pasand ni aay mem kya kr sakt h kya aloo spici ho jay'
,
'nice video',
'super sister',
'hey thank share mam made aloo paratha first time famili love thank',
'nice mam',
'use loov live india',
'super mam',
'nice',
'serious love way u repres food u allway smill',
'aapka bht bada fan hu reciep alway best',
'nice',
'nice recip go tri soon',
'nisha ji ki recipi dekhni valo se anurodh plg unh laik na kre dislaik bhi na kre',
'good morn mem apki har recipi mujh bhut acchi lagti apki har recipi ko laik karti hu mujh barha
accha lgta qyoki ap sabkuch hindi btati mujh jyada english nhi ati',
'love way talk explain realli want learn aloo paratha ill follow recip finger cross',
'thank madam',
'realli tasti aaloo fraitha',
'thank teach aalu paratha',
'nice',
'bhuka marengy aap thank',
'awesom muh mai paani aa gya',
'superb parota yummi yummi',
'use maida instead gehu ka aatha',
'pehli baar aloo ka paratha bana raha hoon bcoz ghar pr koi nahi hai bahut mushkil samajhta tha m
am aapna itn ach tareek se bataya hai thnk much mam',
'thank lot maam delici recip',
'aunti aap bohut sweet ho axh se samjhat ho',
'thank much',
'kya hm aloo ke stuf ko rat readi kr ke rakh sakt h subha ke nasta ke lye',

'yek dum accha hai',
'mam muh mein paani aagaya sure say thait naani make aloo parantha like',
'thank u much nishaji today made paratha n realli tasti thank u ssooo much',
'nice video voic also nice like',
'garma ke dino ke liy kuch recip',
'thank',
'look yummmmmmyyi',
'best aunti thank much',
'thank replay nishaji',
'superrrr',
'amchur powder ki jagah kuch aur use kar sakt hai kya',
'nice aunti',
'u r best aunti',
'fev aaloo paratha',
'wah anti kmal kr diya',
'tq parata supper',
'well done nissan ji',
'u r awesom cook agar mhe aapki ghar ke paa ek ghar rent lelethihoon aap video keliy jo kuch bi recip banath ho aap befhijar mujh tast karn keliy dhe dhena mhe kabhi bhi nahi kahoongi',
'nice mam love recip',
'thank easyy',
'good',
'thnk',
'test paratha',
'nice parata thank u',
'bht shukria tep denay ka indian logo ka urdu',
'uuuuu yummi paratha',
'mouth fill salaiva',
'bohot khub h aap aap ka method bhi bohot acha h like u mam much',
'thank much mam like video also tri recip turn realli nice sweet cute mam feel like mother explain teach cook best whenever want cook someth always see video detail explain provide logic reason technique ingredi ga power impress anyone learn cook video plu motherli love person care feel like mother stand long live nishaji mam god bless',
'nice',
'yummi thank aloo paratha',
'tasti paratha',
'tasti',
'thank great recip love way explain talk like mom',
'thank much yeh recipe main khud bhi bana sakta hun year old bahut hi aasan kar diya aapn thank much',
'delici',
'nice',
'speak sorry cute bye way',
'sleek like mom',
'cook wonder recip',
'veryyyyyyyyy nice',
'kya main iss stuff ko store kar sakti hu',
'amaz mam u r inspir cook teacher',
'thank provide info',
'thank main aap ke style mein parath banay aur bohot tasti bane',
'thank upload help much',
'lyk voic',
'wow amaz easy recip thank much mam',
'thank much mam',
'nice yummmmmmyyi an b like',
'tasti recipe',
'thank',
'thanku soo much u r gem thanku',
'tasti',
'kabli chane sukh kase banain nisha madhulika je',
'lyk ur recip',
'hungry',
'yummi',
'thnx lot mam',
'nice mam',
'ju luv nisha madhulika video explain everything fine',
'thank mam love aloo k paraath',
'wonder parata make today',
'great recip thank share',
'mam nice yummi',
'nice',
'mam u r sweet thank mam',
'love pataro parata recip tri',
'nice',
'thanku much repli nisha aunti tri recip',
'nice parata',
'use chaat garam masalain aloo paratha',

[illegible]

'awesom thankyou much god bless',
'superb',
'thank aunti jiii love u alot',
'wow',
'nice',
'good',
'ty mam',
'mam tikki banana sikh pleas',
'nice',
'thank anuti tast recip love',
'thank much realli appreci',
'best',
'isko dekh kar mujh bhook lag rahe hain',
'thank allot share us mam',
'wow video se mai ne sahi stuf method sikha thank much mam',
'cook aloo stuf wrong method use stuf without cook good health',
'favorit food',
'good',
'wow mam apki baat hi alg h khane k sath sath apk boln bhi test h love mam',
'delicoi',
'thank aunti',
'nice recip mam',
'tri tast good everyon hous like alott tank recipi',
'super unti',
'nice recip nisha g',
'nisha ji ur g love u',
'thank u',
'nice techniqu mam ur recip come tasti first trail',
'nice man bohut acch prath bant h agar kuch banat hu aapki hi recip dekht hu thank much mam',
'ill tri mor morn mam',
'nice yam',
'wow nice mom',
'khaiy mote ho jay',
'mam bless home alon bachelor thank lot',
'misti dokan je poreta hoy tar recip din',
'nisha mam aap ne coriand powder ko fri kara hai kyaa',
'hlo mam luv ur method nd ur smile also',
'mam mash ki dal wali parathi ki recip shar kari',
'kk mam sure let u know made sure yum',
'yum recip make tomorrow',
'thank lot mam tri made nd realli tasti thanku mch',
'good cook',
'awesom awesom awesom awesom awesom awesom',
'help',
'thank u',
'mam best one like',
'thank mam',
'awesom madam',
'nice',
'favourit favourit favourit',
'thank',
'add chees potato stuf',
'thank',
'love',
'nice mam',
'mam jab main tey karti hun allo pranda se bahar aa jate hai kya karo',
'love way explain remind mother',
'yummi yummi dish nisha jiii thank upload video paratha video thank nisha jiii',
'aunti thank share nice receipt big fan recip make aloo parata said get stuf come side roll kindl
i help',
'like process',
'aaj hi banaya main aaloo ke parath yammi thank mam love alway',
'mam aaj mai banaungi kash mai dikhapati',
'wow',
'yummi',
'thank much made today yummi easi love',
'bhookh lg gyi',
'nice',
'made aloo paneer paratha per ur recip turn tasti daughter love thank nishaji',
'thank lot share video',
'aunti ap k ghr aa rhi hu mjhe app k hath se khana h',
'love way explain maam smili face',
'tri easi delici thank mam share',
'aw great tast',
'mam maida se paratha bana sakt hai kya',
'best cook mam love voic',
'ap bohut achh se sikhati hn',

'aunti aap jitni sweet ho utni hi sweet tarik se apn paratha bana na sikhaya hai thank aunti',
'wowww',
'wow',
'l mam tn timer kind repli',
'nice',
'wow',
'yummi make potato arent dri becam soggi wat shud fr',
'like u aloo parath',
'mam without dri mango powder',
'show pictur doubl side paranta',
'sweet nice way ur cook madam',
'nice recip',
'superb',
'super parat',
'awesom',
'yummmmmmmmmmm',
'cute aunti',
'talk much',
'thank ji main bana ys achha bana',
'nice',
'nisha mam u cook well thank explain wonder',
'awesom maam',
'thank aunti',
'thank mam realli help',
'nice mam',
'bad',
'wow mem',
'nice paratha',
'mujh stuf varn hi problem hota',
'yummy',
'mam aapk parath dekh kr bhuk lg gyi look yummi',
'aap bahot sweet ho mam',
'wow tq show stuff paratha',
'nice paratha thank',
'nice paratha anti thank much bohota achha ye paratha yaammiiiiiii',
'thank madam',
'yummi mam hi baut dher bad dkae apk epi',
'stuf ko jeer se fri nahi kar shakte hai',
'nisha mandulia video excit',
'nisha ji watch video lekin mere se roti aur stuf paratha gol nahi hote',
'beauti woman good recip thank much',
'thank appa ka raspic bhoth follow karthe hum',
'super dadi',
'mam u r realli geniu',
'testi',
'mam main pehl parath ko bina tel ke sekti hu phir tel lagati hu ab aapk method se tri karungi',
'nice',
'luke testi',
'love aloo paratha mam wll tri ur process thankyou',
'nyc maa ji bahut tasti hai',
'tnx lot mem todi order aloo k paratha tho main apk video dekh kr tri karungi thank much',
'awesom recip soo good',
'niceeee',
'nice',
'thank u ji tri',
'best onlin chef whole world simpli amaz love hindi',
'tri recip come delici',
'thank u nisha mam u r recip nice',
'use record video',
'nice',
'bahut acchi hai ap ke resepi bahut acchi tarha samjate hai',
'porotta style also sweet',
'yummi',
'woman talent much respect pakistan',
'nice video mam thank u much n love u',
'hello fantast teacher love recip thank patient explain everi detail recip god bless',
'nisha great',
'mem aaj hamn aloo ke parant bany te bahut testi bane te thank mem',
'get recip aloo paratha',
'nice pratha',
'like recip nice thank much',
'wow nice mam',
'fabrit aloo paratha app banat hue dekh mu pani agaya',
'nice video like',
'nice way teach love style',
'awesom',
'thank'.

'namasta nisha aunti thank lot help make aalo ke paratha sister tri first time',
'yummi',
'yeeee mene bnay apn hatho se tn timer aunti',
'look good wish speak inglish understand make copi',
'nice nisha mam',
'hi mam aloo paratha kea sath kon si sabzi serv karoo',
'hello dear nice easi thank',
'yummi paratha',
'mam great see u make awesom dish n learn u',
'tri maam thnxxx',
'thank lot',
'nice easi receip bahot achh se shikhat ho aap love thank much',
'nice like recip easi',
'super thank lot mam',
'want eat paratha wild way mannnn',
'sorri mam mujh ye pasand nahi aaya',
'superb',
'like recip',
'nice mossi',
'good',
'tasti',
'thank mam',
'thank ji',
'like recip mam',
'thank share',
'great',
'today mom go make',
'aunti talk look like mom',
'thank',
'thank aunti way speak realli love sweet',
'thnx mam main aaj aloo parantha banaya nd bhoot yummi bane pehl mujhs ye stuf parantha k bahar a
a jati thi aaj main aapko follow kiya result mummi ki help k bina main parantha banaya thnx mam',
'thank nisha mam',
'thank anti ji',
'mam best',
'thank mam like',
'use instead amchur powder mam',
'thankyou',
'pratha',
'nyc',
'best way could aloo parantha first time yeyyyi came well husband like lot thank much video',
'kk thaqu',
'man realli good jese minut kya ban sakta',
'final got success thank lot made day',
'marvel recip mam thank whenever make paratha alway remain undercook place pleas guid',
'nice one',
'mam ism jira aur ajvain b dal sakt hai',
'yppppiiiiii',
'omg supeer tasti mam',
'yummi',
'nice aunti ji yummyyi',
'aap ki recip mint ka aap ki aat mint kasa hoa olo banr ho kya',
'nice',
'mam good trythi recip love',
'main aj banaya hai bht achi boni ha mam thanx',
'love ur recipi maam ur video make easi cook someth',
'nice know cook receip make easili husband children thank',
'asalam alikum aunti love ur recip',
'nishaji aapki recip bahut acchi rahti hai aajkl mai kuvh bhi banan se pahil aapki recip jarur
dekhti hu thank nisha ji',
'nice',
'nice',
'thank',
'amaz',
'nice',
'thank mam',
'nice',
'superbbbb',
'tri ur recip came realli good tasti',
'delici mam',
'thank lot nisha ji',
'thank',
'thank way u talk love',
'like cook style best want tast u r food',
'super',
'wow yeh mare mummi bhi banati thi par yeh usa bhi tasti lagta ha',
'nice'.

```

...],
'mujh aapk sikhan ka tarika bahut acha lagta hai',
'thank alot nice god bless u',
'love way explain us recip exactli way mom explain',
'apn maida ya gehu ka atta use kia hain',
'aunti ji aloo paratha awsom',
'aloor paratha famou indian food',
'superb',
'made squar paratha tasti',
'thank tasti interest present',
'mam lot go make right',
'happi subtitl yey',
'nice receipi sure tri',
'hi mam without dri dri mango masala kat prepar',
'like video simpl explan',
'mam mere paranth sahi se bilt nhi h fat jate h bataiy kya kru',
'nice present madam u use much oil caus risk health person',
'nice',
'nice mam',
'itna acha bana hai ghar par bhej de',
'thank',
'thank made famili love thank',
'nice recip',
'tasti yummi',
'hello mam thank u soo much video',
'tnx tri',
'brilliantli explain',
'good method g',
'im right away dinner thank share realli easi yummi recip greet germani',
'realli nice',
...]

```

GAUSSIAN NAIVE BAYES WITH COUNT VECTORIZER (NK)

In [22]:

```

# Splitting the dataset into the Training Set and Testing Set
from sklearn.model_selection import train_test_split
X_train_cv2, X_test_cv2, y_train_cv2, y_test_cv2 = train_test_split(Xn_c, yn_c, test_size = 0.2,
                                                                    random_state = 0)

```

In [12]:

```

#Fitting Naive bayes model algorithm with count vectorizer
from sklearn.naive_bayes import GaussianNB
nbk2 = GaussianNB()
nbk2.fit(X_train_cv2, y_train_cv2)

```

Out[12]:

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [13]:

```

#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_nb_cv2 = cross_val_score(estimator=nbk2, X=X_train_cv2, y=y_train_cv2, cv=10)
cross_nb_cv2

```

Out[13]:

```
array([0.54081633, 0.4744898 , 0.44897959, 0.48979592, 0.47193878,
       0.46428571, 0.40816327, 0.50510204, 0.48469388, 0.52295918])
```

In [14]:

```

#prediction test
nb2_pred_cv = nbk2.predict(X_test_cv2)
nb2_pred_cv

```

Out[14]:

```
array([[3, 3, 2, 1, 5, 3, 3, 4, 1, 4, 2, 2, 2, 1, 1, 6, 2, 5, 2, 3, 5, 4,
```

```
2, 2, 7, 1, 3, 1, 7, 5, 4, 1, 7, 4, 2, 3, 2, 7, 3, 4, 4, 6, 3, 3,
2, 3, 2, 6, 1, 1, 6, 3, 3, 6, 7, 3, 2, 2, 7, 2, 2, 3, 1, 4, 2, 5,
2, 2, 1, 2, 2, 3, 4, 1, 5, 2, 2, 1, 2, 3, 3, 5, 2, 7, 3, 7, 2, 2,
2, 4, 4, 2, 2, 4, 2, 7, 2, 5, 3, 1, 3, 3, 4, 3, 2, 4, 6, 4, 1, 1,
4, 5, 1, 4, 5, 1, 5, 2, 5, 3, 2, 3, 2, 4, 2, 2, 1, 2, 5, 3, 3, 1,
5, 2, 3, 2, 4, 6, 7, 7, 3, 4, 7, 5, 5, 2, 7, 3, 2, 5, 7, 2, 1, 5,
2, 3, 2, 7, 2, 2, 2, 1, 2, 5, 3, 1, 1, 7, 2, 2, 2, 1, 2, 7, 2, 6,
4, 4, 3, 2, 2, 2, 7, 3, 2, 7, 3, 7, 4, 2, 1, 7, 3, 7, 5, 5, 3, 4,
7, 4, 1, 2, 3, 1, 2, 2, 3, 4, 5, 3, 7, 3, 7, 3, 5, 3, 1, 2, 4, 6,
1, 7, 6, 3, 2, 1, 3, 4, 2, 1, 2, 7, 2, 7, 4, 3, 3, 7, 2, 7, 1, 7,
2, 1, 6, 3, 2, 6, 4, 2, 1, 3, 2, 1, 2, 4, 2, 6, 3, 7, 3, 5, 4, 3,
7, 5, 5, 2, 2, 3, 4, 1, 5, 1, 3, 2, 5, 5, 5, 6, 3, 3, 3, 3, 6, 4,
7, 2, 3, 3, 1, 7, 3, 4, 7, 2, 3, 2, 6, 5, 1, 6, 2, 6, 5, 4, 3, 2,
2, 2, 2, 2, 4, 1, 3, 4, 3, 3, 4, 5, 7, 1, 1, 7, 2, 1, 3, 2, 3, 2,
2, 7, 7, 3, 6, 2, 4, 3, 3, 3, 2, 2, 3, 2, 3, 2, 3, 3, 4, 5, 7, 2,
3, 1, 7, 1, 3, 7, 1, 4, 7, 1, 2, 2, 3, 6, 4, 2, 7, 1, 2, 5, 3, 2,
1, 4, 3, 7, 1, 3, 3, 3, 2, 1, 7, 3, 7, 1, 2, 2, 3, 2, 3, 5, 3, 2,
6, 2, 7, 5, 5, 1, 3, 5, 1, 5, 2, 7, 1, 3, 5, 7, 1, 2, 2, 2, 3, 7,
2, 4, 7, 3, 3, 3, 4, 4, 2, 5, 3, 5, 5, 5, 1, 3, 3, 2, 3, 3, 3, 7,
2, 7, 1, 3, 1, 2, 2, 3, 2, 3, 4, 1, 4, 5, 3, 6, 1, 6, 6, 2, 7, 1,
3, 4, 3, 7, 2, 5, 3, 3, 5, 4, 5, 4, 3, 3, 7, 3, 7, 1, 4, 2, 4, 4,
2, 3, 2, 4, 7, 3, 4, 6, 4, 4, 7, 1, 5, 4, 1, 7, 7, 7, 6, 4, 3, 2,
6, 5, 2, 1, 3, 2, 5, 1, 3, 3, 4, 2, 4, 3, 7, 7, 3, 1, 3, 5, 3, 1,
5, 2, 1, 3, 7, 7, 1, 2, 2, 3, 3, 3, 1, 4, 3, 2, 1, 2, 2, 3, 6, 1,
2, 3, 4, 5, 1, 3, 3, 4, 6, 5, 3, 5, 3, 3, 5, 7, 1, 4, 7, 1, 3,
2, 4, 2, 1, 2, 7, 5, 4, 2, 2, 3, 2, 5, 1, 2, 5, 2, 3, 6, 3, 2, 3,
3, 2, 7, 3, 4, 5, 6, 4, 1, 5, 2, 3, 1, 3, 2, 4, 3, 7, 3, 5, 4, 4,
7, 4, 2, 4, 2, 3, 2, 7, 6, 5, 3, 1, 3, 6, 2, 1, 3, 1, 3, 1, 4, 4,
3, 4, 4, 1, 3, 3, 1, 1, 6, 2, 4, 1, 6, 2, 7, 1, 4, 1, 5, 2, 2, 3,
1, 4, 2, 3, 7, 4, 1, 4, 7, 6, 5, 7, 2, 3, 2, 3, 2, 3, 7, 1, 2, 3,
7, 3, 3, 2, 7, 2, 2, 2, 7, 3, 4, 4, 3, 5, 7, 6, 5, 7, 4, 5, 7,
2, 3, 7, 2, 7, 4, 2, 3, 6, 4, 1, 2, 6, 1, 5, 2, 7, 6, 1, 3, 7, 5,
6, 2, 7, 1, 2, 7, 2, 4, 2, 3, 3, 6, 1, 3, 3, 5, 4, 7, 7, 7, 3, 2,
1, 6, 6, 2, 2, 7, 7, 2, 7, 3, 2, 6, 6, 1, 3, 6, 3, 3, 4, 5, 7, 7,
2, 1, 4, 5, 1, 5, 3, 2, 3, 4, 2, 7, 3, 3, 5, 1, 5, 2, 7, 5, 2, 2,
2, 6, 3, 3, 4, 7, 7, 3, 2, 1, 5, 2, 3, 7, 2, 3, 5, 4, 2, 1, 6, 7,
3, 2, 7, 6, 7, 7, 3, 4, 2, 1, 2, 5, 2, 4, 5, 2, 2, 5, 2, 5, 3, 4,
4, 3, 6, 7, 1, 3, 2, 7, 7, 2, 2, 2, 3, 1, 7, 1, 7, 2, 6, 7, 5,
2, 2, 4, 6, 2, 6, 7, 1, 3, 7, 1, 3, 6, 4, 6, 6, 2, 1, 3, 7, 4, 3,
2, 3, 2, 2, 2, 3, 2, 3, 2, 7, 7, 3, 3, 4, 7, 5, 2, 3, 1, 4, 2, 7,
1, 2, 2, 4, 7, 1, 2, 2, 4, 2, 2, 2, 3, 3, 1, 3, 4, 3, 3, 2, 7, 1,
4, 1, 3, 6, 5, 7, 7, 7, 3, 6, 1, 1, 2, 1, 7, 1, 7, 3, 4, 4, 2, 7,
2, 7, 6, 3, 2, 1, 3, 1, 3, 1, 4, 2, 1, 6, 3, 1, 4, 7, 1, 6, 1, 7,
2, 7, 7, 4, 7, 4, 4, 1, 7, 4, 3, 1]])
```

VALIDATION TEST

In [17]:

```
print("Naive bayes Accuracy is:",metrics.accuracy_score(y_test_cv2, nb2_pred_cv))
print("Naive bayes f1 score is :",f1_score(y_test_cv2, nb2_pred_cv, average='macro'))
print("Naive bayes recall score is :",recall_score(y_test_cv2, nb2_pred_cv, average='macro'))
print("Naive bayes precision score is :",precision_score(y_test_cv2, nb2_pred_cv, average='macro'))
print("Naive bayes precision score is :",precision_score(y_test_cv2, nb2_pred_cv, average='macro'))
print("Naive bayes matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, nb2_pred_cv))
```

```
Naive bayes Accuracy is: 0.4642857142857143
Naive bayes f1 score is : 0.4393439232618494
Naive bayes recall score is : 0.46851164840101156
Naive bayes precision score is : 0.44563137221068494
Naive bayes precision score is : 0.44563137221068494
Naive bayes matthews_corrcoef score is : 0.3815154405567936
```

GAUSSIAN NAIVE BAYES USING TFIDF VECTORIZER (NK)

In [23]:

```
#creating TFIDF VECTORIZER
from sklearn.feature_extraction.text import TfidfVectorizer
tf n = TfidfVectorizer()
```

```
Xn_tf = tf_n.fit_transform(corpus).toarray()
yn_tf = dataset_n.iloc[:,2].values
print(Xn_tf.shape)
```

(4900, 4483)

In [24]:

```
#Creating TFIDF features
from sklearn.model_selection import train_test_split
X_train_tf2, X_test_tf2, y_train_tf2, y_test_tf2 = train_test_split(Xn_tf, yn_tf, test_size = 0.2,
                                                                    random_state = 0)
```

In [21]:

```
#fitting naive bayes algorithm with tfidf vectorizer
nbk2.fit(X_train_tf2, y_train_tf2)
```

Out[21]:

GaussianNB(priors=None, var_smoothing=1e-09)

In [22]:

```
#cross validation test on training data
from sklearn.model_selection import cross_val_score
cross_nb_tf2 = cross_val_score(estimator=nbk2, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_nb_tf2
```

Out[22]:

```
array([0.54336735, 0.49234694, 0.44642857, 0.49234694, 0.49489796,
       0.46683673, 0.42857143, 0.51020408, 0.48214286, 0.51530612])
```

In [23]:

```
#prediction test
nb2_pred_tf = nbk2.predict(X_test_tf2)
nb2_pred_tf
```

Out[23]:

```
array([3, 3, 2, 1, 5, 3, 3, 4, 1, 4, 2, 2, 2, 1, 1, 6, 2, 5, 2, 3, 5, 5,
       2, 2, 7, 1, 3, 1, 7, 5, 4, 1, 7, 4, 2, 3, 4, 7, 3, 4, 4, 6, 3, 3,
       2, 3, 2, 6, 1, 4, 6, 3, 3, 6, 7, 3, 2, 2, 7, 2, 2, 3, 1, 4, 2, 5,
       2, 2, 1, 2, 2, 3, 4, 1, 5, 2, 2, 1, 2, 3, 3, 5, 2, 7, 3, 7, 2, 2,
       2, 4, 4, 6, 2, 4, 2, 7, 2, 5, 3, 1, 3, 3, 4, 3, 2, 6, 6, 4, 1, 1,
       5, 5, 1, 4, 5, 1, 5, 2, 5, 3, 1, 3, 2, 4, 2, 2, 1, 2, 5, 3, 3, 1,
       6, 3, 2, 7, 4, 6, 7, 6, 3, 4, 7, 5, 5, 2, 7, 3, 2, 6, 7, 2, 1, 5,
       2, 2, 3, 2, 7, 6, 2, 1, 2, 5, 3, 4, 1, 7, 2, 2, 3, 2, 1, 2, 6, 4, 6,
       4, 4, 3, 2, 2, 2, 7, 3, 2, 7, 3, 7, 4, 2, 1, 7, 3, 7, 5, 5, 3, 4,
       7, 4, 1, 2, 3, 1, 2, 2, 3, 4, 5, 3, 7, 3, 6, 3, 7, 3, 1, 2, 4, 6,
       1, 7, 6, 3, 2, 5, 3, 4, 2, 1, 2, 7, 2, 7, 4, 3, 3, 7, 2, 7, 1, 7,
       2, 1, 6, 3, 2, 6, 4, 2, 1, 3, 2, 1, 2, 4, 2, 6, 3, 7, 3, 5, 4, 3,
       7, 5, 5, 2, 2, 4, 1, 5, 1, 3, 2, 5, 5, 5, 6, 3, 3, 3, 3, 3, 6, 4,
       7, 2, 3, 3, 1, 7, 3, 7, 7, 2, 3, 2, 6, 5, 1, 6, 2, 6, 5, 6, 3, 2,
       2, 2, 2, 2, 4, 1, 3, 4, 3, 2, 4, 5, 7, 1, 7, 7, 2, 1, 3, 2, 3, 2,
       2, 7, 7, 3, 6, 2, 4, 3, 3, 3, 2, 2, 3, 2, 3, 2, 3, 3, 5, 6, 7, 2,
       3, 1, 7, 4, 3, 7, 1, 5, 7, 1, 2, 2, 3, 6, 4, 2, 7, 1, 2, 5, 3, 2,
       1, 4, 3, 7, 1, 3, 3, 3, 2, 4, 7, 3, 6, 1, 2, 2, 3, 2, 3, 5, 3, 2,
       6, 2, 7, 5, 5, 1, 3, 5, 1, 5, 2, 7, 1, 3, 5, 7, 1, 2, 2, 2, 3, 7,
       2, 4, 7, 3, 3, 3, 5, 4, 2, 5, 3, 5, 7, 5, 1, 3, 3, 2, 3, 3, 3, 7,
       2, 7, 1, 3, 1, 7, 2, 3, 2, 3, 4, 1, 4, 7, 3, 6, 1, 6, 6, 2, 6, 1,
       3, 4, 3, 7, 2, 7, 3, 3, 5, 4, 5, 4, 3, 3, 7, 3, 7, 1, 4, 2, 4, 4,
       2, 3, 2, 4, 7, 3, 4, 6, 5, 4, 7, 1, 5, 4, 1, 7, 7, 7, 6, 6, 3, 2,
       6, 5, 2, 1, 3, 2, 5, 6, 3, 3, 4, 2, 4, 3, 7, 7, 3, 4, 3, 5, 3, 1,
       5, 2, 1, 3, 7, 7, 1, 2, 2, 3, 3, 3, 1, 4, 3, 2, 1, 2, 2, 3, 6, 1,
       2, 3, 4, 5, 1, 3, 3, 4, 6, 7, 3, 5, 3, 3, 3, 5, 7, 1, 4, 7, 1, 3,
       2, 4, 2, 1, 2, 7, 5, 4, 2, 2, 3, 2, 5, 1, 2, 5, 2, 3, 6, 3, 2, 3,
       3, 2, 7, 3, 4, 5, 6, 4, 5, 5, 2, 3, 1, 3, 2, 4, 3, 6, 3, 5, 4, 4,
       7, 4, 2, 4, 2, 2, 2, 7, 6, 5, 2, 1, 2, 6, 2, 1, 2, 1, 2, 1, 4, 4,
```



```
array([2, 3, 1, 1, 7, 3, 3, 5, 4, 1, 2, 3, 2, 3, 5, 6, 1, 5, 1, 3, 7, 5,
       4, 3, 3, 5, 3, 4, 7, 4, 2, 3, 7, 4, 2, 3, 4, 7, 3, 5, 3, 6, 3, 2,
       2, 3, 2, 1, 1, 5, 6, 3, 3, 6, 6, 3, 1, 5, 7, 1, 1, 3, 3, 4, 4, 5])
```

```

3, 1, 4, 2, 5, 3, 3, 2, 6, 2, 1, 2, 1, 3, 3, 2, 1, 7, 3, 7, 1, 1,
1, 4, 4, 1, 2, 4, 2, 7, 1, 5, 3, 1, 3, 2, 4, 2, 3, 4, 7, 3, 5, 1,
5, 4, 1, 4, 5, 2, 5, 5, 3, 3, 5, 3, 2, 4, 3, 2, 6, 2, 5, 2, 3, 1,
7, 3, 3, 2, 2, 5, 4, 7, 2, 4, 7, 6, 1, 2, 7, 3, 1, 5, 7, 2, 6, 7,
3, 2, 1, 7, 1, 1, 2, 5, 2, 7, 2, 5, 3, 7, 3, 1, 1, 6, 2, 2, 4, 4,
4, 3, 3, 2, 2, 2, 7, 3, 1, 7, 3, 7, 4, 1, 5, 7, 3, 7, 5, 2, 3, 2,
7, 4, 1, 1, 3, 1, 2, 2, 3, 4, 7, 3, 3, 3, 7, 3, 3, 4, 4, 1, 5, 4,
2, 7, 7, 3, 3, 5, 2, 2, 2, 3, 2, 7, 1, 7, 5, 3, 3, 3, 2, 7, 6, 7,
5, 6, 6, 3, 1, 1, 4, 2, 5, 2, 2, 2, 3, 1, 1, 4, 7, 3, 5, 4, 3,
7, 5, 4, 5, 2, 1, 5, 7, 5, 4, 2, 7, 7, 3, 4, 3, 3, 3, 3, 2, 1,
7, 3, 2, 3, 1, 3, 3, 1, 7, 1, 3, 3, 7, 7, 5, 5, 2, 6, 4, 5, 3, 3,
2, 2, 2, 2, 4, 5, 3, 3, 3, 2, 4, 2, 3, 1, 5, 7, 1, 1, 3, 1, 3, 2,
1, 7, 7, 3, 7, 3, 4, 3, 3, 3, 2, 2, 3, 1, 3, 5, 3, 3, 5, 5, 7, 2,
3, 2, 7, 4, 3, 7, 2, 1, 7, 3, 3, 2, 3, 6, 2, 2, 7, 3, 1, 5, 3, 1,
3, 4, 3, 7, 1, 3, 3, 4, 5, 6, 2, 3, 6, 5, 1, 1, 3, 2, 3, 5, 3, 3,
1, 1, 7, 4, 7, 1, 3, 6, 3, 1, 2, 6, 1, 2, 2, 7, 6, 5, 1, 2, 3, 7,
1, 4, 5, 3, 3, 2, 6, 3, 1, 2, 3, 6, 7, 5, 3, 3, 3, 1, 3, 2, 3, 7,
5, 7, 1, 3, 3, 1, 1, 2, 1, 3, 4, 1, 3, 4, 3, 5, 2, 7, 2, 2, 7, 1,
3, 1, 3, 7, 4, 7, 3, 3, 4, 4, 2, 4, 3, 3, 7, 3, 7, 4, 4, 1, 4, 6,
3, 3, 1, 5, 7, 3, 1, 2, 6, 4, 7, 1, 5, 4, 4, 7, 7, 7, 3, 4, 3, 1,
1, 7, 3, 1, 2, 2, 7, 1, 3, 3, 3, 1, 5, 3, 2, 7, 3, 3, 3, 3, 3, 3,
6, 2, 3, 3, 7, 7, 1, 5, 2, 3, 3, 3, 2, 4, 3, 5, 1, 2, 1, 2, 4, 2,
5, 3, 3, 2, 1, 3, 3, 2, 6, 7, 3, 2, 3, 3, 3, 3, 6, 2, 4, 3, 1, 3,
2, 4, 2, 5, 2, 6, 5, 4, 1, 1, 3, 2, 5, 4, 2, 3, 2, 3, 5, 3, 2, 2,
3, 2, 7, 3, 4, 5, 7, 5, 2, 2, 1, 3, 1, 3, 2, 1, 3, 2, 3, 5, 5, 5,
6, 3, 5, 5, 2, 3, 3, 6, 2, 2, 3, 2, 3, 1, 1, 6, 3, 1, 2, 6, 2, 5,
3, 3, 4, 3, 3, 3, 2, 5, 6, 2, 2, 4, 4, 2, 7, 6, 4, 3, 5, 2, 2, 3,
3, 5, 2, 3, 7, 4, 5, 5, 7, 4, 5, 6, 2, 2, 3, 3, 2, 3, 2, 4, 3, 3,
7, 4, 2, 5, 3, 1, 2, 2, 3, 2, 4, 7, 4, 3, 5, 7, 5, 5, 1, 2, 6, 7,
1, 2, 7, 1, 4, 4, 1, 4, 3, 4, 4, 1, 6, 1, 5, 5, 7, 6, 2, 3, 7, 1,
7, 1, 7, 2, 2, 7, 2, 4, 3, 3, 3, 6, 1, 3, 2, 5, 5, 7, 7, 6, 4, 1,
1, 6, 6, 2, 2, 7, 7, 1, 7, 2, 1, 5, 3, 1, 3, 2, 3, 2, 3, 1, 3, 7,
2, 4, 4, 3, 1, 4, 3, 1, 3, 4, 2, 7, 2, 3, 3, 2, 5, 2, 7, 1, 3, 2,
5, 5, 2, 3, 4, 7, 7, 3, 1, 3, 2, 2, 3, 7, 1, 3, 7, 5, 1, 3, 5, 2,
4, 5, 7, 3, 4, 7, 3, 2, 5, 6, 2, 3, 2, 2, 4, 2, 3, 2, 2, 5, 3, 4,
4, 3, 6, 7, 4, 3, 2, 2, 7, 7, 2, 1, 3, 3, 3, 7, 4, 7, 3, 6, 6, 5,
2, 1, 4, 7, 1, 3, 7, 3, 3, 7, 6, 3, 2, 3, 3, 3, 1, 6, 3, 7, 4, 3,
2, 3, 1, 2, 2, 3, 3, 3, 3, 7, 7, 3, 3, 4, 7, 2, 2, 3, 5, 3, 3, 2,
2, 5, 2, 4, 7, 2, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 2, 3, 2, 6, 5,
1, 1, 3, 7, 7, 7, 7, 7, 3, 6, 2, 2, 1, 3, 7, 1, 4, 3, 7, 4, 2, 7,
1, 7, 6, 3, 2, 1, 3, 3, 3, 1, 1, 4, 1, 3, 1, 2, 3, 1, 6, 3, 7,
2, 6, 7, 2, 2, 7, 3, 1, 7, 4, 3, 4])

```

In [30]:

```

print("Naive bayes Accuracy is:",metrics.accuracy_score(y_test_cv2, bnn_pred_cv))
print("Naive bayes f1 score is :",f1_score(y_test_cv2, bnn_pred_cv, average='macro'))
print("Naive bayes recall score is :",recall_score(y_test_cv2, bnn_pred_cv, average='macro'))
print("Naive bayes precison score is :",precision_score(y_test_cv2, bnn_pred_cv, average='macro'))
print("Naive bayes precison score is :",precision_score(y_test_cv2, bnn_pred_cv, average='macro'))
print("Naive bayes matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, bnn_pred_cv))

```

```

Naive bayes Accuracy is: 0.6438775510204081
Naive bayes f1 score is : 0.6253907937313617
Naive bayes recall score is : 0.6480951278695215
Naive bayes precison score is : 0.6654036032342047
Naive bayes precison score is : 0.6654036032342047
Naive bayes matthews_corrcoef score is : 0.5960886963598336

```

Bernoulli NAIVE BAYES USING TFIDF VECTORIZER (NK)

In [32]:

```

#fitting bernouli naive bayes algorithm with count vectorizer
bnn2= BernoulliNB()
bnn2.fit(X_train_tf2, y_train_tf2)

```

Out[32]:

```

BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)

```

In [33]:

```
#cross validation test on training set
cross_bnk_tf2 = cross_val_score(estimator=bnn2, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_bnk_tf2
```

Out[33]:

```
array([0.65306122, 0.64030612, 0.62244898, 0.67346939, 0.64540816,
       0.61989796, 0.64285714, 0.65306122, 0.6377551 , 0.6377551 ])
```

In [149]:

```
bnk2_pred_tf = bnn.predict(X_test_tf2)
bnk2_pred_tf
```

Out[149]:

```
array([2, 3, 1, 1, 7, 3, 3, 5, 4, 1, 2, 3, 2, 3, 5, 6, 1, 5, 1, 3, 7, 5,
       4, 3, 3, 5, 3, 4, 7, 4, 2, 3, 7, 4, 2, 3, 4, 7, 3, 5, 3, 6, 3, 2,
       2, 3, 2, 1, 1, 5, 6, 3, 3, 6, 6, 3, 1, 5, 7, 1, 1, 3, 3, 4, 4, 5,
       3, 1, 4, 2, 5, 3, 3, 2, 6, 2, 1, 2, 1, 3, 3, 2, 1, 7, 3, 7, 1, 1,
       1, 4, 4, 1, 2, 4, 2, 7, 1, 5, 3, 1, 3, 2, 4, 2, 3, 4, 7, 3, 5, 1,
       5, 4, 1, 4, 5, 2, 5, 5, 3, 3, 5, 3, 2, 4, 3, 2, 6, 2, 5, 2, 3, 1,
       7, 3, 3, 2, 2, 5, 4, 7, 2, 4, 7, 6, 1, 2, 7, 3, 1, 5, 7, 2, 6, 7,
       3, 2, 1, 7, 1, 1, 2, 5, 2, 7, 2, 5, 3, 7, 3, 1, 1, 6, 2, 2, 4, 4,
       4, 3, 3, 2, 2, 2, 7, 3, 1, 7, 3, 7, 4, 1, 5, 7, 3, 7, 5, 2, 3, 2,
       7, 4, 1, 1, 3, 1, 2, 2, 3, 4, 7, 3, 3, 3, 7, 3, 3, 4, 4, 1, 5, 4,
       2, 7, 7, 3, 3, 5, 2, 2, 3, 2, 7, 1, 7, 5, 3, 3, 3, 2, 7, 6, 7,
       5, 6, 6, 3, 1, 1, 4, 2, 5, 2, 2, 2, 3, 1, 1, 4, 7, 3, 5, 4, 3,
       7, 5, 4, 5, 2, 1, 5, 7, 5, 4, 2, 7, 7, 3, 4, 3, 3, 3, 3, 2, 1,
       7, 3, 2, 3, 1, 3, 3, 1, 7, 1, 3, 3, 7, 7, 5, 5, 2, 6, 4, 5, 3, 3,
       2, 2, 2, 2, 4, 5, 3, 3, 3, 2, 4, 2, 3, 1, 5, 7, 1, 1, 3, 1, 3, 2,
       1, 7, 7, 3, 7, 3, 4, 3, 3, 3, 2, 2, 3, 1, 3, 5, 3, 3, 5, 5, 7, 2,
       3, 2, 7, 4, 3, 7, 2, 1, 7, 3, 3, 2, 3, 6, 2, 2, 7, 3, 1, 5, 3, 1,
       3, 4, 3, 7, 1, 3, 3, 4, 5, 6, 2, 3, 6, 5, 1, 1, 3, 2, 3, 5, 3, 3,
       1, 1, 7, 4, 7, 1, 3, 6, 3, 1, 2, 6, 1, 2, 2, 7, 6, 5, 1, 2, 3, 7,
       1, 4, 5, 3, 3, 2, 6, 3, 1, 2, 3, 6, 7, 5, 3, 3, 3, 1, 3, 2, 3, 7,
       5, 7, 1, 3, 3, 1, 1, 2, 1, 3, 4, 1, 3, 4, 3, 5, 2, 7, 2, 2, 7, 1,
       3, 1, 3, 7, 4, 7, 3, 3, 4, 4, 2, 4, 3, 3, 7, 3, 7, 4, 4, 1, 4, 6,
       3, 3, 1, 5, 7, 3, 1, 2, 6, 4, 7, 1, 5, 4, 4, 7, 7, 7, 3, 4, 3, 1,
       1, 7, 3, 1, 2, 2, 7, 1, 3, 3, 3, 1, 5, 3, 2, 7, 3, 3, 3, 3, 3,
       6, 2, 3, 3, 7, 7, 1, 5, 2, 3, 3, 3, 2, 4, 3, 5, 1, 2, 1, 2, 4, 2,
       5, 3, 3, 2, 1, 3, 3, 2, 6, 7, 3, 2, 3, 3, 3, 3, 6, 2, 4, 3, 1, 3,
       2, 4, 2, 5, 2, 6, 5, 4, 1, 1, 3, 2, 5, 4, 2, 3, 2, 3, 5, 3, 2, 2,
       3, 2, 7, 3, 4, 5, 7, 5, 2, 2, 1, 3, 1, 3, 2, 1, 3, 2, 3, 5, 5, 5,
       6, 3, 5, 5, 2, 3, 3, 6, 2, 2, 3, 2, 3, 1, 1, 6, 3, 1, 2, 6, 2, 5,
       3, 3, 4, 3, 3, 3, 2, 5, 6, 2, 2, 4, 4, 2, 7, 6, 4, 3, 5, 2, 2, 3,
       3, 5, 2, 3, 7, 4, 5, 5, 7, 4, 5, 6, 2, 2, 3, 3, 2, 3, 2, 4, 3, 3,
       7, 4, 2, 5, 3, 1, 2, 2, 3, 2, 4, 7, 4, 3, 5, 7, 5, 5, 1, 2, 6, 7,
       1, 2, 7, 1, 4, 4, 1, 4, 3, 4, 4, 1, 6, 1, 5, 5, 7, 6, 2, 3, 7, 1,
       7, 1, 7, 2, 2, 7, 2, 4, 3, 3, 3, 6, 1, 3, 2, 5, 5, 7, 7, 6, 4, 1,
       1, 6, 6, 2, 2, 7, 7, 1, 7, 2, 1, 5, 3, 1, 3, 2, 3, 2, 3, 1, 3, 7,
       2, 4, 4, 3, 1, 4, 7, 3, 1, 3, 4, 2, 2, 7, 2, 3, 3, 2, 5, 2, 7, 1, 3, 2,
       5, 5, 2, 3, 4, 7, 7, 3, 1, 3, 2, 2, 3, 7, 1, 3, 7, 5, 1, 3, 5, 2,
       4, 5, 7, 3, 4, 7, 3, 2, 5, 6, 2, 3, 2, 2, 4, 2, 3, 2, 2, 5, 3, 4,
       4, 3, 6, 7, 4, 3, 2, 2, 7, 7, 2, 1, 3, 3, 3, 7, 4, 7, 3, 6, 6, 5,
       2, 1, 4, 7, 1, 3, 7, 3, 3, 7, 6, 3, 2, 3, 3, 3, 1, 6, 3, 7, 4, 3,
       2, 3, 1, 2, 2, 3, 3, 3, 3, 7, 7, 3, 3, 4, 7, 2, 2, 3, 5, 3, 3, 2,
       2, 5, 2, 4, 7, 2, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 2, 3, 2, 6, 5,
       1, 1, 3, 7, 7, 7, 7, 7, 3, 6, 2, 2, 1, 3, 7, 1, 4, 3, 7, 4, 2, 7,
       1, 7, 6, 3, 2, 1, 3, 3, 3, 1, 1, 4, 1, 3, 1, 2, 3, 1, 6, 3, 7,
       2, 6, 7, 2, 2, 7, 3, 1, 7, 4, 3, 4])
```

In [133]:

```
print("Naive bayes Accuracy is:",metrics.accuracy_score(y_test_tf2, bnn_pred_tf))
print("Naive bayes f1 score is :",f1_score(y_test_tf2, bnk2_pred_tf, average='macro'))
print("Naive bayes recall score is :",recall_score(y_test_tf2, bnk2_pred_tf, average='macro'))
print("Naive bayes precison score is :",precision_score(y_test_tf2, bnk2_pred_tf, average='macro'))
print("Naive bayes precison score is :",precision_score(y_test_tf2, bnk2_pred_tf, average='macro'))
print("Naive bayes matthews_corrcoef score is :",matthews_corrcoef(y_test_tf2, bnk2_pred_tf))
```

```
Naive bayes Accuracy is: 0.6438775510204081
Naive bayes f1 score is : 0.6253907937313617
Naive bayes recall score is : 0.6480951278695215
Naive bayes precison score is : 0.6654036032342047
Naive bayes precison score is : 0.6654036032342047
Naive bayes matthews_corrcoef score is : 0.5960886963598336
```

SUPPORT VECTOR(LINEAR) MACHINE MODEL USING COUNT VECTORIZER

In [25]:

```
#fitting svm model using count vectorizer
svm_n = SVC(kernel = 'linear', random_state=0)
svm_n.fit(X_train_cv2,y_train_cv2)
```

Out[25]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

In []:

```
#cross validation test
cross_svm_cv2 = cross_val_score(estimator=svm_n, X=X_train_cv2, y=y_train_cv2, cv=10)
cross_svm_cv2
```

In [27]:

```
svm_pred_cv2 = svm_n.predict(X_test_cv2)
svm_pred_cv2
```

Out[27]:

```
array([2, 6, 1, 1, 7, 3, 3, 4, 4, 1, 3, 3, 5, 6, 5, 2, 1, 5, 1, 4, 7, 2,
       2, 5, 6, 5, 3, 4, 7, 4, 2, 6, 6, 4, 2, 3, 4, 7, 3, 5, 4, 6, 3, 6,
       3, 6, 6, 1, 1, 5, 4, 3, 3, 6, 6, 3, 1, 1, 7, 1, 1, 3, 6, 4, 4, 5,
       3, 1, 4, 6, 5, 3, 6, 6, 7, 6, 1, 5, 1, 4, 3, 2, 1, 7, 3, 7, 1, 1,
       5, 7, 4, 5, 2, 4, 2, 7, 1, 5, 4, 6, 6, 2, 4, 2, 5, 4, 7, 4, 5, 1,
       1, 4, 1, 4, 5, 6, 5, 5, 6, 3, 5, 3, 2, 4, 2, 5, 7, 2, 5, 2, 3, 3,
       7, 4, 3, 2, 2, 5, 4, 7, 2, 4, 6, 4, 1, 2, 7, 3, 5, 5, 7, 2, 6, 7,
       2, 2, 1, 7, 1, 1, 6, 5, 6, 7, 2, 5, 4, 7, 2, 1, 1, 6, 2, 6, 4, 5,
       2, 3, 3, 2, 2, 2, 7, 3, 1, 4, 4, 7, 4, 1, 5, 7, 4, 7, 5, 2, 3, 2,
       6, 4, 1, 1, 3, 5, 2, 2, 3, 4, 7, 3, 6, 3, 7, 6, 6, 4, 4, 1, 5, 4,
       5, 7, 7, 3, 3, 6, 2, 6, 6, 1, 4, 7, 5, 7, 5, 3, 3, 6, 2, 7, 6, 7,
       5, 6, 1, 3, 1, 1, 4, 2, 2, 2, 6, 6, 6, 1, 1, 4, 6, 4, 5, 4, 3,
       7, 5, 4, 5, 2, 1, 5, 7, 5, 4, 2, 7, 7, 6, 4, 3, 3, 3, 3, 6, 5,
       7, 6, 4, 6, 5, 7, 3, 1, 7, 1, 3, 3, 7, 6, 5, 5, 2, 2, 4, 5, 3, 6,
       2, 2, 2, 2, 5, 1, 3, 4, 6, 2, 5, 6, 6, 1, 5, 7, 5, 1, 3, 1, 3, 6,
       1, 7, 7, 3, 6, 2, 4, 3, 3, 6, 6, 2, 3, 1, 6, 5, 3, 4, 1, 5, 7, 2,
       3, 6, 7, 4, 4, 7, 5, 5, 7, 6, 5, 2, 6, 6, 6, 2, 7, 3, 1, 7, 3, 1,
       4, 4, 3, 6, 1, 3, 3, 4, 5, 4, 6, 3, 6, 5, 5, 1, 3, 2, 3, 1, 3, 6,
       5, 5, 7, 4, 7, 1, 6, 6, 1, 1, 2, 6, 5, 2, 2, 6, 6, 5, 5, 4, 3, 7,
       1, 4, 5, 3, 4, 2, 6, 6, 1, 4, 3, 6, 7, 2, 1, 3, 3, 5, 3, 4, 3, 7,
       5, 7, 1, 3, 6, 1, 1, 2, 1, 3, 4, 1, 4, 4, 3, 5, 2, 6, 2, 2, 7, 1,
       3, 1, 3, 7, 4, 7, 4, 3, 4, 5, 2, 4, 6, 4, 6, 3, 7, 4, 4, 1, 5, 6,
       2, 3, 1, 5, 7, 3, 1, 4, 6, 5, 6, 1, 5, 4, 4, 7, 7, 6, 7, 4, 3, 5,
       1, 7, 5, 5, 2, 2, 7, 1, 3, 3, 3, 1, 5, 6, 2, 7, 3, 6, 3, 3, 3,
       6, 5, 3, 3, 7, 7, 1, 5, 2, 3, 4, 3, 6, 4, 3, 5, 1, 3, 1, 2, 4, 5,
       5, 4, 3, 4, 1, 3, 3, 2, 6, 7, 4, 6, 3, 3, 3, 6, 6, 2, 1, 6, 5, 3,
       2, 4, 2, 1, 2, 6, 1, 4, 5, 1, 3, 2, 5, 4, 2, 4, 2, 3, 5, 6, 2, 2,
       3, 6, 7, 3, 4, 4, 7, 5, 2, 2, 1, 3, 5, 3, 2, 1, 3, 2, 3, 5, 5, 5,
       6, 6, 1, 5, 2, 3, 3, 7, 6, 6, 3, 2, 3, 1, 1, 6, 4, 1, 2, 7, 6, 5,
       3, 4, 4, 5, 3, 4, 2, 5, 7, 6, 2, 6, 6, 6, 7, 6, 6, 3, 1, 2, 2, 3,
       1, 5, 2, 3, 7, 4, 5, 5, 7, 4, 1, 6, 2, 2, 2, 3, 2, 3, 6, 4, 3, 3,
       7, 4, 2, 5, 3, 1, 4, 6, 6, 6, 4, 7, 4, 3, 1, 7, 4, 1, 1, 6, 7, 7,
       1, 6, 7, 1, 2, 4, 1, 4, 6, 4, 6, 1, 3, 1, 5, 5, 7, 5, 5, 3, 7, 1,
       7, 1, 6, 2, 2, 7, 2, 4, 6, 3, 6, 6, 1, 3, 2, 4, 1, 7, 7, 6, 6, 1,
```

```

5, 6, 6, 2, 2, 7, 7, 5, 7, 2, 1, 5, 3, 1, 3, 6, 3, 2, 4, 1, 7, 7,
2, 4, 4, 6, 1, 4, 3, 1, 3, 4, 2, 6, 4, 6, 6, 2, 1, 2, 7, 1, 2, 6,
5, 5, 2, 3, 4, 7, 6, 3, 1, 6, 4, 4, 6, 7, 1, 3, 7, 1, 1, 5, 5, 2,
4, 5, 7, 1, 6, 7, 6, 2, 1, 6, 2, 3, 2, 6, 5, 2, 3, 7, 2, 5, 3, 4,
4, 3, 6, 7, 4, 3, 2, 2, 7, 7, 2, 1, 3, 3, 6, 7, 4, 7, 2, 6, 7, 5,
2, 5, 4, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 1, 3, 6, 1, 6, 3, 7, 4, 3,
6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 5, 3, 3, 4, 7, 2, 2, 3, 5, 6, 2, 2,
5, 5, 2, 4, 6, 2, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 6, 3, 2, 6, 5,
5, 1, 3, 7, 7, 7, 7, 3, 6, 5, 2, 1, 4, 7, 1, 6, 6, 6, 4, 2, 7,
1, 7, 6, 3, 4, 1, 6, 4, 6, 1, 1, 1, 4, 6, 3, 1, 6, 6, 4, 6, 3, 7,
2, 6, 7, 2, 6, 6, 3, 1, 7, 4, 3, 4])

```

In [32]:

```

print("SVM(LINEAR) Accuracy is:",metrics.accuracy_score(y_test_cv2, svm_pred_cv2))
print("SVM(LINEAR) f1 score is :",f1_score(y_test_cv2, svm_pred_cv2, average='macro'))
print("SVM(LINEAR) recall score is :",recall_score(y_test_cv2, svm_pred_cv2, average='macro'))
print("SVM(LINEAR) precison score is :",precision_score(y_test_cv2, svm_pred_cv2, average='macro')
)
print("SVM(LINEAR) precison score is :",precision_score(y_test_cv2, svm_pred_cv2, average='macro')
)
print("SVM(LINEAR) matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, svm_pred_cv2))

```

```

SVM(LINEAR) Accuracy is: 0.7642857142857142
SVM(LINEAR) f1 score is : 0.7644287843397322
SVM(LINEAR) recall score is : 0.7652796310615485
SVM(LINEAR) precison score is : 0.7675495709587922
SVM(LINEAR) precison score is : 0.7675495709587922
SVM(LINEAR) matthews_corrcoef score is : 0.7255882045867541

```

SVM MODEL(LINEAR) USING TFIDF VECTORIZER

In [29]:

```

#fitting svm model using tfidf vectorizer
svm_n.fit(X_train_tf2,y_train_tf2)

```

Out[29]:

```

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)

```

In []:

```

cross_svm_tf2 = cross_val_score(estimator=svm_n, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_svm_tf2

```

In [30]:

```

svm_pred_tf2 = svm_n.predict(X_test_tf2)
svm_pred_tf2

```

Out[30]:

```

array([2, 6, 1, 1, 7, 3, 3, 5, 4, 4, 3, 3, 5, 6, 5, 6, 1, 5, 1, 4, 7, 5,
       4, 5, 6, 5, 3, 4, 7, 4, 6, 6, 7, 4, 2, 3, 4, 7, 3, 5, 4, 6, 3, 6,
       2, 6, 6, 1, 1, 5, 6, 3, 3, 6, 6, 3, 1, 1, 7, 1, 1, 3, 6, 4, 4, 5,
       3, 1, 4, 6, 5, 3, 7, 6, 7, 6, 1, 5, 1, 4, 3, 2, 1, 7, 3, 7, 1, 1,
       5, 7, 4, 2, 2, 4, 2, 7, 1, 5, 4, 6, 6, 2, 4, 2, 5, 4, 7, 4, 5, 1,
       1, 4, 1, 4, 6, 6, 5, 5, 7, 3, 5, 3, 2, 4, 2, 2, 7, 2, 5, 2, 3, 6,
       7, 4, 3, 2, 2, 5, 4, 7, 2, 4, 7, 4, 1, 2, 7, 3, 5, 4, 7, 2, 6, 7,
       2, 2, 1, 7, 1, 1, 6, 5, 6, 7, 2, 5, 4, 7, 2, 1, 1, 6, 2, 6, 4, 5,
       4, 6, 3, 2, 2, 2, 7, 3, 1, 4, 4, 7, 4, 1, 5, 7, 4, 7, 5, 6, 3, 2,
       6, 4, 1, 1, 3, 1, 2, 2, 3, 4, 7, 3, 7, 3, 7, 6, 6, 4, 4, 1, 5, 4,
       2, 7, 7, 3, 3, 6, 2, 6, 7, 1, 4, 7, 5, 7, 5, 3, 3, 6, 2, 7, 6, 7,
       5, 6, 6, 3, 1, 1, 4, 2, 2, 2, 2, 6, 6, 6, 1, 1, 4, 6, 4, 2, 4, 3,
       7, 5, 6, 5, 2, 6, 5, 7, 4, 4, 2, 7, 7, 6, 6, 3, 3, 3, 3, 3, 6, 5,
       7, 6, 4, 6, 5, 7, 3, 1, 7, 1, 3, 4, 7, 6, 5, 5, 2, 6, 4, 5, 3, 6,

```

```

2, 2, 2, 4, 5, 6, 3, 4, 6, 2, 4, 6, 6, 1, 5, 7, 5, 1, 3, 1, 3, 2,
1, 7, 7, 3, 6, 4, 4, 3, 3, 6, 2, 2, 3, 1, 6, 5, 3, 4, 1, 5, 7, 6,
3, 6, 7, 4, 4, 7, 5, 1, 7, 6, 5, 2, 6, 6, 6, 2, 7, 4, 1, 7, 3, 1,
4, 4, 3, 7, 1, 3, 3, 4, 5, 6, 6, 3, 6, 4, 5, 1, 3, 2, 3, 5, 3, 6,
5, 5, 7, 4, 7, 1, 6, 6, 1, 5, 2, 6, 5, 2, 2, 6, 6, 5, 5, 4, 3, 7,
1, 4, 5, 3, 4, 2, 7, 6, 1, 4, 3, 6, 7, 2, 1, 3, 3, 5, 3, 4, 3, 7,
5, 7, 5, 3, 6, 5, 1, 2, 1, 3, 4, 1, 4, 4, 3, 5, 2, 6, 2, 2, 7, 1,
3, 1, 4, 7, 4, 7, 4, 3, 4, 2, 2, 4, 3, 4, 6, 3, 7, 4, 4, 1, 4, 6,
4, 3, 1, 5, 7, 3, 1, 6, 6, 4, 6, 5, 5, 4, 4, 7, 7, 6, 7, 4, 3, 5,
1, 7, 5, 5, 2, 2, 7, 1, 3, 3, 4, 1, 5, 6, 7, 7, 3, 6, 3, 6, 3, 3,
6, 3, 3, 3, 7, 7, 1, 5, 2, 3, 4, 3, 6, 6, 3, 5, 1, 2, 1, 2, 6, 1,
5, 4, 4, 5, 1, 3, 3, 6, 6, 7, 4, 7, 3, 3, 3, 6, 6, 2, 6, 6, 5, 3,
2, 4, 2, 1, 2, 6, 5, 4, 5, 1, 3, 2, 5, 4, 2, 4, 2, 3, 5, 6, 2, 2,
3, 6, 7, 3, 4, 4, 7, 5, 2, 2, 1, 3, 1, 3, 2, 1, 3, 6, 3, 5, 5, 2,
6, 6, 1, 5, 2, 3, 3, 6, 6, 6, 3, 2, 3, 1, 1, 6, 3, 1, 2, 6, 7, 5,
3, 4, 4, 5, 3, 4, 2, 5, 7, 6, 2, 5, 6, 6, 7, 6, 4, 4, 5, 2, 2, 3,
1, 5, 2, 3, 7, 4, 5, 5, 7, 4, 1, 6, 2, 2, 2, 3, 2, 3, 6, 4, 3, 3,
7, 4, 2, 5, 6, 1, 2, 6, 6, 6, 4, 7, 4, 3, 5, 7, 6, 5, 1, 6, 7, 7,
1, 6, 7, 1, 6, 4, 1, 4, 6, 4, 4, 1, 6, 1, 5, 5, 7, 6, 5, 3, 7, 1,
7, 1, 6, 6, 2, 7, 2, 4, 6, 3, 6, 6, 1, 3, 2, 4, 1, 7, 7, 6, 3, 1,
1, 6, 6, 2, 2, 7, 7, 5, 7, 2, 1, 5, 6, 1, 3, 6, 3, 2, 4, 5, 7, 7,
2, 4, 4, 6, 1, 4, 3, 1, 3, 4, 2, 6, 4, 6, 6, 7, 5, 6, 7, 1, 2, 6,
5, 5, 2, 3, 4, 7, 7, 3, 1, 6, 4, 4, 6, 7, 1, 3, 7, 1, 1, 5, 5, 2,
4, 5, 7, 1, 6, 7, 6, 2, 5, 6, 2, 6, 2, 6, 1, 2, 3, 7, 2, 5, 3, 4,
4, 3, 6, 7, 4, 3, 2, 2, 7, 7, 6, 1, 3, 3, 6, 7, 4, 7, 2, 6, 7, 5,
2, 5, 4, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 5, 6, 6, 1, 6, 3, 7, 4, 3,
6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 7, 3, 3, 4, 7, 2, 2, 2, 5, 6, 2, 6,
5, 5, 2, 4, 6, 6, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 4, 6, 3, 2, 6, 2,
7, 1, 3, 7, 7, 7, 7, 3, 6, 5, 2, 1, 4, 7, 1, 6, 6, 7, 4, 2, 7,
1, 7, 6, 3, 4, 1, 6, 4, 6, 1, 1, 1, 4, 6, 6, 1, 6, 6, 6, 6, 7,
4, 6, 7, 4, 6, 6, 3, 4, 7, 4, 3, 4])

```

In [31]:

```

print(" SVM(LINEAR)  Accuracy is:",metrics.accuracy_score(y_test_tf2, svm_pred_tf2))
print(" SVM(LINEAR) f1 score is :",f1_score(y_test_tf2, svm_pred_tf2, average='macro'))
print("SVM(LINEAR) recall score is :",recall_score(y_test_tf2, svm_pred_tf2, average='macro'))
print("SVM(LINEAR) precison score is :",precision_score(y_test_tf2, svm_pred_tf2, average='macro')
)
print("SVM(LINEAR) precison score is :",precision_score(y_test_tf2, svm_pred_tf2, average='macro')
)
print("SVM(LINEAR) matthews_corrcoef score is :",matthews_corrcoef(y_test_tf2, svm_pred_tf2))

```

```

SVM(LINEAR)  Accuracy is: 0.7724489795918368
SVM(LINEAR) f1 score is : 0.7732338012552898
SVM(LINEAR) recall score is : 0.7736984716524092
SVM(LINEAR) precison score is : 0.7794806376066427
SVM(LINEAR) precison score is : 0.7794806376066427
SVM(LINEAR) matthews_corrcoef score is : 0.7357219868364955

```

SVM POLY KERNAL USING COUNT VECTORIZER

In [50]:

```

#fitting svm using count vectorizer
from sklearn.svm import SVC
Psvm_n = SVC(kernel = 'poly', random_state=0)
Psvm_n.fit(X_train_cv2,y_train_cv2)

```

Out[50]:

```

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='poly',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)

```

In []:

```

#cross validation test
cross_psvm_tf2 = cross_val_score(estimator=svm_n, X=X_train_cv2, y=y_train_cv2, cv=10)
cross_psvm_tf2

```

In [51]:

```
psvm_pred_cv2 = Psvm_n.predict(X_test_cv2)
psvm_pred_cv2
```

Out[51]:

```
array([3, 3, 1, 1, 3, 3, 3, 1, 3, 1, 3, 3, 5, 3, 5, 3, 1, 5, 1, 3, 3, 2,
       2, 3, 3, 1, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 7, 3, 1, 3, 6, 3, 3,
       3, 3, 3, 3, 1, 5, 2, 3, 3, 3, 3, 3, 1, 1, 7, 1, 1, 3, 3, 3, 3, 5,
       3, 1, 3, 3, 5, 3, 3, 3, 3, 2, 1, 2, 1, 3, 3, 2, 1, 7, 3, 7, 1, 1,
       1, 3, 3, 5, 2, 2, 3, 3, 1, 2, 3, 3, 3, 2, 2, 2, 3, 3, 3, 5, 1,
       5, 3, 1, 3, 5, 3, 5, 1, 3, 3, 5, 3, 3, 3, 3, 3, 3, 1, 2, 3, 3,
       7, 3, 3, 3, 3, 1, 3, 7, 2, 4, 3, 2, 3, 3, 3, 3, 1, 5, 7, 3, 3, 3,
       3, 2, 1, 3, 1, 1, 3, 5, 3, 3, 2, 5, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3,
       2, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 4, 1, 1, 7, 3, 7, 1, 3, 3, 3,
       3, 4, 1, 1, 3, 1, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 4,
       2, 7, 7, 3, 3, 2, 2, 3, 3, 3, 3, 3, 2, 2, 5, 3, 3, 3, 3, 3, 3, 7,
       1, 3, 7, 3, 1, 1, 3, 3, 2, 2, 2, 3, 3, 3, 1, 1, 4, 3, 3, 2, 3, 3,
       3, 1, 3, 5, 3, 5, 5, 3, 3, 3, 7, 2, 3, 3, 3, 3, 3, 3, 3, 3, 5,
       3, 3, 2, 3, 1, 3, 3, 1, 3, 1, 3, 3, 7, 3, 5, 5, 3, 2, 4, 5, 3, 3,
       3, 3, 3, 3, 1, 5, 3, 3, 3, 3, 5, 3, 3, 1, 5, 3, 5, 5, 3, 1, 3, 3,
       1, 7, 3, 3, 2, 3, 4, 3, 3, 3, 2, 3, 3, 1, 3, 5, 3, 3, 1, 5, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 1, 2, 3, 5, 3, 3, 3, 3, 2, 3, 3, 1, 7, 3, 1,
       3, 4, 3, 3, 3, 3, 3, 3, 5, 7, 3, 3, 3, 5, 1, 1, 3, 2, 3, 5, 3, 3,
       1, 3, 7, 4, 7, 2, 3, 3, 3, 1, 2, 2, 1, 2, 3, 3, 3, 1, 5, 3, 3, 7,
       1, 4, 1, 3, 3, 2, 3, 3, 1, 3, 3, 2, 3, 5, 3, 3, 3, 1, 3, 2, 3, 7,
       1, 7, 3, 3, 3, 1, 1, 3, 1, 3, 3, 3, 3, 3, 1, 2, 3, 2, 3, 3, 1,
       3, 2, 3, 3, 3, 7, 3, 3, 5, 5, 3, 4, 3, 3, 3, 3, 3, 3, 1, 5, 3,
       3, 3, 1, 1, 3, 3, 1, 3, 3, 3, 3, 1, 5, 3, 3, 3, 7, 3, 3, 3, 3, 1,
       1, 7, 3, 5, 2, 3, 3, 2, 3, 3, 3, 1, 5, 3, 3, 7, 3, 3, 3, 3, 3, 3,
       3, 2, 3, 3, 3, 7, 1, 1, 3, 3, 3, 3, 3, 4, 3, 1, 1, 3, 1, 2, 2, 2,
       1, 3, 3, 2, 1, 3, 3, 3, 3, 7, 3, 3, 3, 3, 3, 3, 6, 2, 3, 3, 3, 3,
       3, 2, 3, 5, 3, 3, 3, 4, 1, 1, 3, 2, 5, 4, 3, 3, 3, 3, 5, 3, 3, 2,
       3, 3, 3, 3, 4, 3, 3, 5, 3, 2, 1, 3, 3, 3, 3, 1, 3, 2, 3, 5, 1, 2,
       6, 3, 1, 1, 3, 3, 3, 2, 3, 3, 3, 3, 3, 1, 1, 3, 3, 1, 2, 3, 3, 5,
       3, 3, 3, 3, 3, 3, 3, 1, 7, 3, 2, 2, 3, 3, 7, 3, 3, 3, 1, 2, 3, 3,
       3, 1, 2, 3, 7, 4, 1, 3, 7, 4, 1, 3, 2, 2, 3, 3, 2, 3, 3, 3, 3, 3,
       7, 4, 2, 5, 3, 1, 2, 3, 3, 3, 3, 2, 3, 1, 7, 4, 6, 5, 3, 3, 3,
       1, 3, 3, 1, 2, 3, 1, 3, 3, 3, 3, 1, 3, 1, 5, 5, 7, 5, 2, 3, 3, 1,
       7, 1, 3, 5, 3, 7, 3, 4, 3, 3, 3, 3, 1, 3, 2, 4, 5, 7, 3, 4, 3, 1,
       1, 2, 3, 2, 3, 3, 3, 1, 3, 2, 1, 4, 3, 1, 3, 2, 3, 3, 3, 1, 3, 7,
       3, 2, 3, 3, 1, 4, 3, 1, 3, 4, 3, 3, 2, 3, 3, 2, 5, 3, 7, 1, 3, 3,
       1, 5, 2, 3, 4, 3, 3, 3, 1, 3, 3, 3, 3, 3, 1, 3, 7, 5, 1, 3, 5, 3,
       3, 1, 7, 3, 3, 7, 3, 3, 1, 3, 3, 3, 3, 3, 4, 3, 3, 3, 3, 2, 3, 3,
       3, 3, 2, 7, 3, 3, 3, 2, 7, 2, 3, 1, 3, 3, 3, 1, 3, 7, 3, 3, 3, 5,
       3, 1, 3, 7, 1, 3, 7, 3, 3, 6, 3, 3, 3, 3, 3, 3, 1, 3, 3, 7, 3, 3,
       3, 3, 1, 3, 3, 3, 3, 3, 7, 3, 3, 3, 4, 7, 3, 3, 3, 5, 3, 3, 3,
       3, 5, 2, 3, 3, 2, 2, 2, 2, 1, 1, 3, 3, 3, 3, 3, 1, 3, 3, 3, 2, 5,
       1, 1, 3, 7, 3, 7, 7, 3, 3, 3, 3, 2, 1, 3, 3, 1, 3, 3, 3, 3, 3, 3,
       1, 7, 4, 3, 3, 1, 3, 3, 3, 1, 1, 1, 3, 3, 3, 1, 3, 3, 3, 3, 3, 7,
       3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3])
```

In [52]:

```
#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_cv2, psvm_pred_cv2))
print("SVM(POLY) f1 score is :",f1_score(y_test_cv2, psvm_pred_cv2, average='macro'))
print("SVM(POLY) recall score is :",recall_score(y_test_cv2, psvm_pred_cv2, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_cv2, psvm_pred_cv2, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_cv2, psvm_pred_cv2, average='macro'))
print("SVM(POLY) matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, psvm_pred_cv2))
```

```
SVM Accuracy is: 0.4030612244897959
SVM(POLY) f1 score is : 0.37179213825551
SVM(POLY) recall score is : 0.40135821954625717
SVM(POLY) precison score is : 0.5783709467155164
SVM(POLY) precison score is : 0.5783709467155164
SVM(POLY) matthews_corrcoef score is : 0.35812398181937827
```

SVN POLY KERNEL USING TFIDF

In [47]:

```
#fitting svm using tfidf vectorizer
from sklearn.svm import SVC
Psvm_n = SVC(kernel = 'poly', random_state=0)
Psvm_n.fit(X_train_tf2,y_train_tf2)
```

Out[47]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='poly',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

In []:

```
#cross validation test
cross_psvm_tf2 = cross_val_score(estimator=rsvm_n, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_psvm_tf2
```

In [48]:

```
psvm_pred_tf2 = Psvm_n.predict(X_test_tf2)
psvm_pred_tf2
```

Out[48]:

```
array([6, 6, 1, 4, 7, 3, 4, 5, 6, 6, 3, 6, 5, 6, 5, 6, 1, 5, 1, 4, 7, 6,
       6, 5, 6, 5, 3, 4, 7, 6, 6, 6, 7, 4, 2, 3, 4, 7, 3, 6, 4, 6, 3, 6,
       6, 6, 6, 6, 1, 5, 6, 3, 3, 6, 6, 3, 6, 5, 6, 1, 1, 3, 6, 4, 6, 4,
       2, 1, 5, 2, 5, 3, 7, 6, 7, 6, 5, 5, 1, 4, 3, 6, 1, 7, 3, 7, 1, 1,
       4, 6, 6, 6, 2, 4, 6, 7, 1, 6, 4, 6, 6, 2, 4, 2, 5, 6, 6, 4, 4, 6,
       6, 6, 1, 6, 6, 6, 6, 5, 7, 3, 6, 3, 5, 4, 2, 6, 6, 2, 6, 2, 3, 6,
       7, 4, 3, 2, 4, 5, 6, 7, 2, 4, 6, 4, 6, 2, 7, 6, 5, 4, 7, 2, 6, 6,
       2, 2, 1, 7, 1, 6, 7, 5, 6, 7, 2, 5, 4, 7, 2, 6, 5, 6, 6, 6, 4, 6,
       4, 6, 3, 2, 5, 6, 7, 6, 1, 6, 4, 7, 4, 1, 5, 7, 4, 7, 5, 6, 3, 4,
       6, 4, 6, 1, 3, 6, 6, 2, 3, 4, 6, 3, 7, 3, 6, 6, 7, 5, 6, 1, 5, 6,
       5, 7, 7, 3, 3, 6, 2, 6, 6, 1, 5, 7, 2, 6, 5, 3, 3, 6, 2, 6, 6, 7,
       2, 6, 6, 3, 1, 6, 4, 2, 6, 2, 2, 6, 6, 6, 1, 6, 4, 6, 3, 6, 4, 3,
       7, 5, 6, 5, 5, 6, 5, 6, 4, 4, 6, 7, 7, 6, 6, 3, 3, 2, 3, 3, 6, 5,
       7, 6, 4, 6, 6, 7, 3, 6, 7, 6, 3, 4, 6, 7, 5, 6, 6, 6, 6, 6, 3, 6,
       2, 2, 2, 2, 4, 6, 3, 4, 6, 2, 6, 6, 7, 1, 6, 7, 5, 6, 3, 1, 3, 2,
       1, 7, 7, 3, 6, 4, 4, 3, 3, 6, 2, 2, 6, 6, 6, 5, 3, 4, 6, 6, 7, 6,
       3, 5, 7, 6, 4, 6, 6, 5, 6, 6, 5, 2, 6, 6, 6, 2, 7, 6, 1, 6, 3, 1,
       4, 6, 3, 6, 1, 3, 3, 4, 5, 6, 6, 6, 6, 4, 5, 6, 3, 2, 6, 5, 6, 6,
       6, 3, 7, 4, 7, 1, 6, 6, 1, 6, 2, 6, 5, 2, 5, 6, 6, 2, 5, 6, 3, 7,
       1, 4, 6, 3, 4, 2, 7, 6, 1, 6, 4, 6, 7, 6, 1, 3, 3, 1, 3, 4, 3, 7,
       5, 7, 1, 3, 6, 6, 1, 3, 1, 3, 4, 1, 4, 6, 3, 6, 5, 6, 6, 6, 6, 1,
       3, 6, 6, 7, 4, 7, 4, 3, 4, 5, 6, 6, 3, 4, 6, 3, 6, 4, 6, 1, 5, 6,
       4, 3, 1, 6, 6, 3, 6, 6, 4, 6, 5, 6, 4, 4, 7, 7, 6, 7, 6, 3, 5,
       6, 7, 6, 5, 2, 2, 7, 6, 3, 3, 4, 1, 6, 6, 7, 7, 3, 6, 3, 6, 3, 6,
       6, 4, 6, 3, 7, 6, 1, 6, 2, 3, 4, 3, 6, 6, 3, 5, 6, 6, 6, 2, 6, 5,
       5, 4, 6, 5, 6, 3, 3, 6, 6, 7, 4, 7, 4, 3, 3, 6, 6, 2, 6, 6, 5, 3,
       6, 4, 2, 5, 2, 6, 6, 4, 1, 1, 3, 2, 6, 4, 2, 6, 2, 3, 6, 6, 2, 2,
       3, 2, 7, 3, 4, 6, 6, 6, 5, 6, 1, 3, 1, 3, 6, 4, 3, 6, 3, 6, 5, 4,
       6, 6, 5, 5, 2, 3, 2, 6, 6, 6, 3, 6, 3, 6, 1, 6, 2, 6, 3, 6, 7, 5,
       3, 4, 4, 5, 3, 4, 6, 6, 5, 5, 6, 6, 6, 7, 6, 6, 4, 5, 2, 4, 3,
       1, 5, 5, 3, 7, 6, 5, 5, 7, 6, 5, 6, 2, 2, 2, 3, 2, 6, 7, 4, 2, 3,
       7, 4, 6, 1, 6, 1, 6, 6, 6, 6, 4, 7, 6, 3, 5, 7, 6, 6, 6, 6, 6, 7,
       6, 6, 7, 1, 6, 4, 1, 4, 6, 4, 6, 1, 6, 6, 5, 5, 7, 6, 2, 3, 7, 6,
       7, 1, 7, 6, 2, 7, 2, 4, 2, 3, 6, 6, 6, 3, 2, 6, 6, 7, 7, 6, 6, 1,
       6, 6, 6, 3, 2, 7, 7, 2, 7, 2, 1, 6, 6, 6, 3, 6, 6, 2, 4, 6, 7, 6,
       2, 5, 6, 6, 6, 4, 3, 1, 3, 4, 2, 7, 4, 6, 6, 7, 5, 6, 7, 5, 2, 1,
       2, 5, 2, 6, 4, 7, 6, 3, 1, 6, 6, 4, 6, 7, 1, 3, 7, 6, 1, 6, 6, 6,
       4, 5, 7, 6, 6, 7, 6, 6, 5, 6, 2, 6, 2, 6, 4, 2, 6, 7, 2, 5, 3, 6,
       6, 6, 6, 7, 6, 3, 2, 6, 7, 7, 6, 6, 6, 3, 6, 7, 4, 7, 2, 6, 7, 6,
       2, 5, 6, 7, 1, 6, 7, 1, 6, 6, 6, 3, 6, 5, 6, 6, 1, 6, 3, 7, 4, 3,
       6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 7, 6, 3, 4, 7, 6, 2, 2, 5, 6, 2, 7,
       5, 5, 6, 4, 6, 6, 5, 2, 4, 1, 1, 2, 3, 3, 6, 3, 6, 6, 3, 2, 6, 6,
       7, 1, 3, 6, 6, 7, 7, 3, 6, 5, 5, 1, 4, 7, 5, 6, 6, 7, 4, 2, 7,
       1, 7, 6, 3, 2, 1, 6, 4, 6, 6, 4, 1, 4, 6, 6, 6, 6, 7, 6, 6, 6, 7,
       2, 6, 6, 4, 6, 6, 5, 6, 7, 6, 3, 4])
```


In [49]:

```
#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_tf2, psvm_pred_tf2))
print("SVM(POLY) f1 score is :",f1_score(y_test_tf2, psvm_pred_tf2, average='macro'))
print("SVM(POLY) recall score is :",recall_score(y_test_tf2, psvm_pred_tf2, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_tf2, psvm_pred_tf2, average='macro'))
print("SVM(POLY) precison score is :",precision_score(y_test_tf2, psvm_pred_tf2, average='macro'))
print("SVM(POLY) matthews_corrcoef score is :",matthews_corrcoef(y_test_tf2, psvm_pred_tf2))
```

```
SVM Accuracy is: 0.6224489795918368
SVM(POLY) f1 score is : 0.6387060762385384
SVM(POLY) recall score is : 0.6221467343752831
SVM(POLY) precison score is : 0.7228488296816372
SVM(POLY) precison score is : 0.7228488296816372
SVM(POLY) matthews_corrcoef score is : 0.5802738368341996
```

SVM GAUSSIAN USING COUNT VECTORIZER

In [60]:

```
#fitting svm using count vectorizer
from sklearn.svm import SVC
rsvm_n = SVC(kernel = 'rbf', random_state=0)
rsvm_n.fit(X_train_cv2,y_train_cv2)
```

Out[60]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

In []:

```
#cross validation test
cross_rsvm_tf2 = cross_val_score(estimator=rsvm_n, X=X_train_cv2, y=y_train_cv2, cv=10)
cross_rsvm_tf2
```

In [62]:

```
#prediction test
rsvm_pred_cv2 = rsvm_n.predict(X_test_cv2)
rsvm_pred_cv2
```

Out[62]:

```
array([2, 6, 1, 1, 7, 3, 6, 5, 4, 7, 5, 2, 5, 6, 5, 6, 1, 7, 1, 4, 7, 2,
       2, 5, 6, 5, 3, 4, 7, 4, 2, 6, 7, 4, 2, 3, 4, 7, 3, 5, 3, 6, 3, 6,
       2, 6, 6, 6, 1, 5, 4, 3, 3, 6, 6, 3, 1, 1, 7, 1, 1, 3, 6, 4, 4, 5,
       3, 1, 4, 6, 5, 3, 7, 6, 7, 2, 1, 5, 1, 4, 3, 2, 1, 7, 3, 7, 1, 1,
       5, 7, 4, 5, 2, 4, 2, 7, 1, 5, 4, 6, 6, 2, 4, 2, 5, 4, 7, 4, 5, 5,
       5, 6, 1, 4, 7, 6, 5, 5, 6, 3, 5, 3, 2, 4, 6, 2, 6, 2, 7, 2, 3, 6,
       7, 4, 3, 6, 2, 1, 4, 7, 2, 4, 7, 7, 6, 2, 7, 3, 5, 5, 7, 2, 7, 7,
       2, 2, 1, 7, 1, 1, 6, 5, 6, 7, 2, 5, 4, 7, 2, 1, 1, 6, 2, 6, 4, 4,
       4, 6, 3, 2, 4, 2, 7, 3, 1, 7, 4, 7, 4, 1, 5, 7, 4, 7, 5, 6, 3, 2,
       6, 4, 5, 1, 3, 1, 2, 2, 3, 4, 7, 3, 6, 3, 7, 6, 6, 4, 4, 1, 5, 7,
       5, 7, 7, 3, 3, 7, 2, 6, 2, 1, 4, 7, 5, 7, 5, 3, 3, 6, 2, 7, 6, 7,
       5, 6, 7, 3, 1, 7, 4, 2, 2, 2, 2, 6, 6, 6, 1, 7, 4, 7, 4, 7, 4, 3,
       7, 5, 6, 5, 2, 5, 5, 7, 1, 4, 2, 7, 7, 6, 4, 3, 3, 6, 3, 3, 6, 5,
       7, 6, 4, 6, 5, 6, 3, 7, 7, 1, 3, 6, 7, 7, 5, 7, 2, 7, 4, 7, 3, 6,
       2, 2, 2, 4, 5, 7, 3, 6, 6, 2, 7, 6, 6, 1, 5, 7, 5, 7, 3, 1, 3, 6,
       1, 7, 7, 3, 7, 3, 4, 3, 3, 6, 2, 2, 3, 1, 6, 5, 3, 4, 1, 5, 7, 6,
       3, 6, 7, 4, 4, 7, 5, 1, 7, 6, 5, 2, 6, 7, 6, 2, 7, 4, 1, 7, 3, 1,
       4, 4, 3, 6, 1, 3, 3, 4, 5, 7, 6, 3, 6, 7, 5, 1, 3, 2, 3, 5, 3, 6,
       5, 1, 7, 4, 7, 1, 6, 6, 1, 1, 2, 6, 5, 2, 2, 7, 6, 5, 5, 4, 3, 7,
       1, 4, 5, 3, 4, 2, 6, 6, 1, 4, 3, 6, 7, 7, 6, 3, 3, 5, 3, 4, 3, 7,
       5, 7, 1, 3, 6, 1, 1, 4, 1, 3, 4, 1, 4, 4, 3, 5, 2, 6, 2, 2, 7, 1,
       3, 1, 3, 7, 4, 7, 4, 3, 7, 5, 2, 7, 6, 4, 7, 3, 7, 4, 4, 1, 7, 7,
       3, 3, 1, 5, 7, 3, 5, 6, 6, 4, 7, 5, 5, 4, 4, 7, 7, 7, 3, 4, 3, 5,
```

```

1, 7, 5, 5, 2, 2, 7, 2, 3, 3, 7, 1, 7, 6, 7, 7, 3, 6, 3, 6, 3, 3,
6, 5, 3, 3, 7, 7, 5, 5, 2, 3, 4, 3, 6, 7, 3, 5, 1, 6, 1, 2, 6, 5,
5, 4, 6, 4, 5, 3, 3, 6, 6, 7, 4, 6, 3, 3, 3, 6, 7, 2, 7, 6, 5, 3,
2, 4, 2, 5, 2, 7, 5, 4, 5, 1, 3, 2, 5, 7, 2, 4, 2, 6, 5, 6, 2, 2,
3, 6, 7, 3, 4, 4, 7, 7, 2, 5, 1, 3, 5, 3, 2, 1, 3, 2, 3, 7, 5, 5,
7, 6, 1, 5, 2, 3, 3, 7, 6, 6, 3, 2, 3, 1, 1, 6, 4, 7, 2, 6, 6, 5,
3, 4, 4, 1, 3, 4, 6, 5, 7, 6, 2, 6, 4, 6, 7, 6, 7, 4, 1, 2, 2, 3,
6, 5, 2, 3, 7, 7, 5, 5, 7, 6, 1, 6, 2, 2, 2, 3, 2, 3, 2, 4, 3, 3,
7, 4, 2, 5, 6, 7, 1, 5, 6, 6, 6, 4, 7, 4, 3, 5, 7, 5, 7, 5, 6, 7, 7,
1, 6, 7, 1, 6, 4, 1, 4, 6, 4, 6, 1, 6, 1, 5, 5, 7, 7, 5, 3, 7, 1,
7, 1, 7, 7, 2, 7, 6, 4, 6, 3, 6, 6, 1, 3, 2, 5, 7, 7, 7, 6, 6, 1,
5, 6, 6, 2, 2, 7, 7, 5, 7, 2, 1, 5, 6, 1, 3, 6, 3, 2, 3, 1, 7, 7,
2, 4, 6, 6, 1, 4, 3, 1, 3, 4, 2, 6, 4, 6, 6, 6, 5, 6, 7, 1, 3, 6,
5, 5, 2, 3, 4, 7, 7, 3, 1, 6, 4, 6, 6, 7, 1, 3, 7, 7, 1, 5, 7, 7,
4, 5, 7, 1, 6, 7, 6, 6, 5, 6, 2, 2, 2, 6, 7, 2, 3, 7, 2, 4, 3, 4,
4, 3, 7, 7, 4, 3, 2, 2, 7, 7, 6, 1, 3, 3, 6, 7, 4, 7, 3, 6, 7, 7,
2, 5, 4, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 3, 6, 6, 1, 6, 3, 7, 4, 3,
6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 7, 3, 3, 4, 7, 2, 2, 3, 5, 6, 2, 2,
2, 5, 2, 4, 7, 2, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 6, 3, 2, 6, 5,
1, 5, 3, 7, 7, 7, 7, 7, 3, 6, 2, 2, 1, 2, 7, 1, 6, 6, 6, 4, 2, 7,
1, 7, 6, 3, 4, 1, 6, 4, 6, 1, 1, 1, 4, 6, 6, 1, 6, 6, 6, 7, 6, 7,
4, 7, 7, 2, 6, 6, 3, 1, 7, 4, 3, 4])

```

In [63]:

```

#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_cv2, rsvm_pred_cv2))
print("SVM(GAUSSIAN) f1 score is :",f1_score(y_test_cv2, rsvm_pred_cv2, average='macro'))
print("SVM(GAUSSIAN) recall score is :",recall_score(y_test_cv2, rsvm_pred_cv2, average='macro'))
print("SVM(GAUSSIAN) precision score is :",precision_score(y_test_cv2, rsvm_pred_cv2,
average='macro'))
print("SVM(GAUSSIAN) precision score is :",precision_score(y_test_cv2, rsvm_pred_cv2,
average='macro'))
print("SVM(GAUSSIAN) matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, rsvm_pred_cv2))

```

```

SVM Accuracy is: 0.7183673469387755
SVM(GAUSSIAN) f1 score is : 0.7196516068431353
SVM(GAUSSIAN) recall score is : 0.719417879175691
SVM(GAUSSIAN) precision score is : 0.735169632246185
SVM(GAUSSIAN) precision score is : 0.735169632246185
SVM(GAUSSIAN) matthews_corrcoef score is : 0.6740480772028755

```

SVM GAUSSIAN USING TFIDF

In [65]:

```

#fitting svm using tfidf vectorizer
from sklearn.svm import SVC
rsvm_n = SVC(kernel = 'rbf', random_state=0)
rsvm_n.fit(X_train_tf2,y_train_tf2)

```

Out[65]:

```

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)

```

In []:

```

#cross validation test
cross_rsvm_tf2 = cross_val_score(estimator=rsvm_n, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_rsvm_tf2

```

In [66]:

```

rsvm_pred_tf2 = rsvm_n.predict(X_test_tf2)
rsvm_pred_tf2

```

Out[66]:

Out[66]:

```
array([[6, 6, 1, 1, 7, 3, 4, 5, 4, 4, 3, 6, 5, 6, 5, 6, 1, 5, 1, 4, 7, 5,
        4, 5, 6, 5, 3, 4, 7, 4, 6, 6, 7, 4, 2, 3, 4, 7, 3, 5, 4, 6, 3, 6,
        6, 6, 6, 6, 1, 5, 6, 3, 3, 6, 6, 3, 1, 5, 7, 1, 1, 3, 6, 4, 4, 5,
        2, 1, 4, 2, 5, 3, 7, 6, 6, 2, 1, 5, 1, 4, 3, 2, 1, 7, 3, 7, 1, 1,
        5, 7, 4, 2, 2, 4, 2, 7, 1, 2, 4, 6, 6, 2, 4, 2, 5, 4, 6, 4, 5, 4,
        5, 4, 1, 4, 5, 6, 5, 5, 7, 3, 5, 3, 5, 4, 2, 2, 7, 2, 6, 2, 3, 6,
        7, 4, 3, 2, 2, 5, 4, 7, 2, 4, 7, 4, 6, 2, 7, 2, 5, 4, 7, 2, 6, 7,
        2, 2, 1, 7, 1, 6, 6, 5, 6, 7, 2, 5, 4, 7, 2, 1, 1, 6, 2, 6, 4, 6,
        4, 6, 3, 2, 2, 2, 7, 6, 1, 6, 4, 7, 4, 1, 5, 7, 4, 7, 5, 6, 3, 2,
        6, 4, 1, 1, 3, 1, 2, 2, 3, 4, 7, 3, 7, 3, 7, 6, 6, 4, 4, 1, 5, 4,
        5, 7, 7, 3, 3, 6, 2, 6, 7, 1, 4, 7, 5, 7, 5, 3, 3, 6, 2, 7, 6, 7,
        5, 6, 6, 3, 1, 1, 4, 2, 2, 2, 2, 6, 6, 6, 1, 1, 4, 6, 4, 6, 4, 3,
        7, 5, 6, 5, 2, 6, 5, 6, 4, 4, 2, 7, 7, 6, 6, 3, 3, 2, 3, 3, 6, 5,
        7, 6, 4, 6, 5, 7, 3, 6, 7, 1, 3, 4, 7, 7, 5, 5, 6, 6, 4, 5, 3, 6,
        2, 2, 2, 4, 6, 3, 4, 6, 2, 4, 6, 6, 1, 5, 7, 1, 1, 3, 1, 3, 2,
        1, 7, 7, 3, 6, 4, 4, 3, 3, 6, 2, 2, 2, 1, 6, 5, 3, 4, 6, 5, 7, 6,
        3, 6, 7, 6, 4, 7, 5, 1, 7, 6, 5, 2, 6, 6, 6, 2, 7, 4, 1, 7, 3, 1,
        4, 4, 3, 6, 1, 3, 3, 4, 5, 6, 6, 3, 6, 4, 5, 1, 3, 2, 3, 5, 3, 6,
        5, 5, 7, 4, 7, 1, 6, 6, 1, 1, 2, 6, 5, 2, 4, 6, 6, 5, 5, 4, 3, 7,
        1, 4, 5, 3, 4, 2, 7, 6, 1, 6, 3, 6, 7, 2, 1, 3, 3, 1, 3, 4, 3, 7,
        5, 7, 1, 3, 6, 1, 1, 3, 1, 3, 4, 1, 4, 4, 3, 5, 5, 6, 2, 2, 7, 1,
        3, 1, 4, 7, 4, 7, 4, 3, 4, 5, 2, 4, 3, 4, 6, 3, 7, 4, 4, 1, 4, 6,
        4, 3, 1, 5, 6, 3, 1, 6, 6, 4, 6, 5, 5, 4, 4, 7, 7, 6, 7, 4, 3, 5,
        1, 7, 5, 5, 2, 2, 7, 1, 3, 3, 4, 1, 5, 6, 7, 7, 3, 6, 3, 6, 3, 6,
        6, 3, 4, 3, 7, 6, 1, 5, 2, 3, 4, 3, 6, 6, 3, 5, 1, 2, 1, 2, 6, 5,
        5, 4, 6, 4, 1, 3, 3, 6, 6, 7, 4, 7, 3, 3, 3, 6, 6, 2, 6, 6, 5, 3,
        2, 4, 2, 5, 2, 6, 5, 4, 5, 1, 3, 2, 5, 4, 2, 4, 2, 3, 5, 6, 2, 2,
        3, 2, 7, 3, 4, 4, 7, 5, 2, 2, 1, 3, 1, 3, 2, 5, 3, 6, 3, 5, 5, 5,
        6, 6, 1, 5, 2, 3, 2, 6, 6, 6, 3, 6, 3, 1, 1, 6, 4, 1, 2, 6, 7, 5,
        3, 4, 4, 5, 3, 4, 2, 5, 6, 6, 2, 5, 6, 6, 7, 6, 6, 4, 5, 2, 2, 3,
        1, 5, 2, 3, 7, 4, 5, 5, 7, 4, 5, 6, 2, 2, 2, 3, 2, 4, 6, 4, 2, 3,
        7, 4, 2, 5, 6, 1, 2, 6, 6, 6, 4, 7, 4, 3, 5, 7, 6, 6, 5, 6, 6, 7,
        1, 6, 7, 1, 6, 4, 1, 4, 6, 4, 4, 1, 6, 1, 5, 5, 7, 6, 2, 3, 7, 1,
        7, 1, 7, 6, 2, 7, 2, 4, 2, 3, 6, 6, 1, 3, 2, 4, 5, 7, 7, 6, 4, 1,
        1, 6, 6, 2, 2, 7, 7, 5, 7, 2, 1, 5, 6, 1, 3, 6, 3, 2, 4, 5, 7, 7,
        2, 4, 6, 6, 1, 4, 3, 1, 3, 4, 2, 7, 4, 6, 6, 7, 5, 6, 7, 1, 2, 6,
        5, 5, 2, 2, 4, 7, 7, 3, 1, 6, 6, 4, 6, 7, 1, 3, 7, 1, 1, 5, 6, 6,
        4, 5, 7, 6, 6, 7, 6, 4, 5, 6, 2, 6, 2, 6, 4, 2, 3, 7, 2, 5, 3, 4,
        4, 4, 6, 7, 4, 3, 2, 2, 7, 7, 6, 1, 2, 3, 6, 7, 4, 7, 2, 6, 6, 5,
        2, 5, 4, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 5, 6, 6, 1, 6, 3, 7, 4, 3,
        6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 7, 2, 3, 4, 7, 2, 2, 2, 5, 6, 2, 6,
        2, 5, 2, 4, 6, 6, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 4, 6, 3, 2, 6, 2,
        7, 1, 3, 7, 7, 7, 7, 7, 3, 6, 5, 2, 1, 4, 7, 1, 6, 6, 7, 4, 2, 7,
        1, 7, 6, 3, 4, 1, 6, 4, 6, 1, 1, 1, 4, 6, 6, 1, 6, 6, 6, 6, 7,
        2, 6, 6, 4, 6, 6, 6, 6, 7, 4, 3, 4]])
```

In [67]:

```
#validation test
print("SVM Accuracy is:",metrics.accuracy_score(y_test_tf2, rsvm_pred_tf2))
print("SVM(GAUSSIAN) f1 score is :",f1_score(y_test_tf2, rsvm_pred_tf2, average='macro'))
print("SVM(GAUSSIAN) recall score is :",recall_score(y_test_tf2, rsvm_pred_tf2, average='macro'))
print("SVM(GAUSSIAN) precision score is :",precision_score(y_test_tf2, rsvm_pred_tf2,
average='macro'))
print("SVM(GAUSSIAN) precision score is :",precision_score(y_test_tf2, rsvm_pred_tf2,
average='macro'))
print("SVM(GAUSSIAN) matthews_corrcoef score is :",matthews_corrcoef(y_test_tf2, rsvm_pred_tf2))
```

```
SVM Accuracy is: 0.7408163265306122
SVM(GAUSSIAN) f1 score is : 0.7445808160133142
SVM(GAUSSIAN) recall score is : 0.7420163406446986
SVM(GAUSSIAN) precision score is : 0.7604537712931209
SVM(GAUSSIAN) precision score is : 0.7604537712931209
SVM(GAUSSIAN) matthews_corrcoef score is : 0.7003934922986773
```

Random forest classifier using count vectorizer

In [74]:

```
#fitting random forest classifier using count vectorizer
rf_n = RandomForestClassifier(n_estimators=200, random_state=0)
```

```
rf_n.fit(X_train_cv2, y_train_cv2)
```

Out[74]:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                        criterion='gini', max_depth=None, max_features='auto',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=200,
                        n_jobs=None, oob_score=False, random_state=0, verbose=0,
                        warm_start=False)
```

In []:

```
#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_rf_cv2 = cross_val_score(estimator=rf_n, X=X_train_cv2, y=y_train_cv2, cv=10)
cross_rf_cv2
```

In [75]:

```
#prediction test
rf_pred_cv2 = rf_n.predict(X_test_cv2)
rf_pred_cv2
```

Out[75]:

```
array([2, 3, 1, 1, 7, 3, 3, 5, 4, 1, 3, 3, 5, 6, 5, 2, 1, 1, 1, 4, 7, 6,
       2, 5, 6, 5, 3, 4, 7, 4, 2, 6, 7, 4, 2, 3, 4, 7, 3, 5, 3, 6, 3, 6,
       3, 3, 6, 6, 1, 5, 2, 3, 3, 7, 6, 3, 1, 1, 7, 1, 1, 3, 6, 4, 4, 5,
       3, 1, 4, 2, 4, 3, 7, 6, 6, 2, 1, 2, 1, 4, 3, 2, 1, 7, 3, 7, 1, 1,
       4, 7, 6, 5, 2, 2, 7, 1, 2, 4, 6, 6, 2, 4, 2, 5, 4, 6, 4, 5, 5,
       1, 4, 1, 4, 5, 6, 5, 5, 7, 3, 5, 3, 2, 5, 2, 2, 6, 2, 6, 2, 3, 3,
       7, 5, 3, 2, 3, 1, 4, 7, 2, 4, 6, 6, 6, 3, 7, 3, 5, 5, 7, 2, 6, 7,
       2, 2, 1, 7, 1, 5, 6, 5, 6, 7, 5, 5, 3, 7, 3, 1, 5, 6, 2, 6, 6, 3,
       4, 6, 3, 2, 4, 5, 7, 3, 1, 4, 3, 7, 4, 1, 5, 7, 4, 7, 1, 6, 3, 2,
       6, 4, 1, 1, 3, 1, 2, 2, 3, 4, 7, 3, 7, 3, 7, 3, 6, 4, 6, 1, 5, 4,
       5, 7, 7, 3, 3, 7, 2, 6, 2, 1, 4, 7, 5, 7, 5, 3, 3, 1, 3, 7, 6, 7,
       5, 6, 6, 3, 1, 1, 4, 2, 2, 2, 6, 6, 6, 1, 1, 4, 7, 4, 2, 6, 3,
       7, 1, 6, 5, 5, 5, 5, 6, 4, 4, 2, 7, 7, 3, 6, 3, 3, 3, 3, 2, 5,
       7, 6, 4, 3, 5, 7, 3, 5, 7, 1, 3, 4, 7, 7, 5, 5, 2, 7, 4, 5, 3, 6,
       2, 2, 3, 2, 5, 5, 3, 6, 3, 2, 1, 6, 6, 5, 5, 7, 5, 5, 3, 1, 3, 6,
       1, 7, 7, 3, 7, 3, 4, 3, 3, 3, 2, 2, 3, 1, 3, 5, 3, 4, 1, 5, 7, 2,
       3, 3, 7, 4, 4, 6, 2, 1, 7, 3, 5, 2, 3, 6, 6, 2, 7, 3, 1, 7, 3, 1,
       4, 4, 3, 2, 1, 3, 3, 4, 5, 7, 6, 3, 6, 5, 5, 1, 3, 2, 3, 1, 3, 2,
       5, 1, 7, 4, 7, 1, 3, 6, 1, 1, 2, 6, 1, 2, 2, 6, 6, 5, 5, 4, 3, 7,
       1, 4, 1, 3, 4, 5, 7, 6, 1, 4, 4, 6, 7, 2, 1, 3, 3, 5, 3, 4, 3, 7,
       5, 7, 1, 3, 6, 1, 1, 2, 1, 4, 4, 1, 6, 4, 3, 5, 2, 6, 2, 2, 7, 1,
       3, 5, 3, 7, 4, 7, 4, 3, 4, 5, 2, 4, 3, 4, 7, 3, 6, 4, 4, 1, 4, 6,
       3, 3, 1, 5, 7, 3, 5, 6, 6, 3, 6, 5, 5, 4, 4, 7, 7, 6, 7, 4, 3, 4,
       1, 7, 5, 5, 2, 2, 7, 2, 3, 3, 7, 1, 4, 6, 7, 7, 3, 6, 3, 6, 3, 3,
       6, 5, 3, 3, 7, 6, 1, 5, 2, 3, 4, 3, 6, 4, 3, 5, 1, 3, 1, 2, 6, 2,
       5, 4, 6, 6, 1, 3, 3, 2, 6, 7, 4, 7, 3, 3, 3, 6, 6, 2, 4, 6, 5, 3,
       2, 2, 2, 5, 2, 6, 1, 4, 5, 1, 3, 2, 5, 4, 2, 6, 2, 3, 5, 3, 2, 2,
       3, 6, 7, 3, 4, 4, 7, 5, 2, 2, 1, 3, 5, 3, 2, 1, 3, 2, 3, 6, 1, 5,
       6, 6, 5, 5, 2, 3, 3, 6, 6, 6, 3, 2, 3, 1, 1, 6, 2, 1, 2, 6, 6, 5,
       3, 4, 4, 5, 3, 4, 2, 5, 7, 6, 2, 6, 4, 6, 7, 3, 6, 4, 1, 2, 2, 3,
       1, 1, 2, 3, 7, 4, 5, 2, 7, 4, 1, 6, 2, 2, 2, 3, 2, 3, 2, 4, 3, 3,
       7, 4, 2, 1, 6, 1, 2, 6, 6, 6, 2, 7, 2, 3, 5, 7, 5, 6, 5, 6, 7, 7,
       1, 6, 7, 1, 2, 4, 1, 4, 6, 4, 1, 1, 6, 1, 5, 5, 7, 6, 2, 3, 7, 1,
       7, 1, 7, 6, 2, 7, 6, 4, 3, 3, 6, 6, 1, 3, 2, 4, 5, 7, 2, 6, 3, 1,
       1, 2, 6, 2, 2, 7, 7, 5, 7, 2, 1, 4, 4, 1, 3, 2, 3, 2, 3, 1, 6, 7,
       2, 5, 6, 3, 1, 4, 3, 1, 3, 4, 2, 7, 4, 3, 6, 7, 5, 6, 7, 1, 2, 6,
       5, 5, 2, 3, 4, 7, 7, 3, 1, 6, 4, 3, 3, 7, 1, 3, 7, 5, 1, 5, 5, 2,
       4, 5, 7, 1, 4, 7, 3, 6, 1, 6, 2, 3, 3, 6, 4, 2, 3, 2, 2, 2, 3, 4,
       7, 3, 6, 7, 4, 3, 2, 2, 7, 7, 2, 1, 3, 3, 6, 1, 4, 7, 3, 3, 6, 5,
       2, 1, 4, 7, 1, 6, 7, 1, 3, 7, 6, 3, 2, 3, 6, 6, 1, 6, 3, 7, 4, 3,
       6, 3, 1, 2, 2, 4, 3, 3, 6, 7, 7, 3, 3, 4, 7, 2, 2, 2, 5, 3, 2, 2,
       2, 5, 2, 4, 6, 2, 5, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 6, 3, 2, 6, 5,
       1, 1, 3, 7, 7, 7, 7, 6, 3, 6, 2, 2, 1, 4, 7, 1, 6, 3, 7, 4, 2, 7,
       1, 7, 6, 3, 2, 1, 6, 4, 3, 1, 1, 4, 6, 3, 1, 6, 3, 6, 6, 3, 7,
       2, 1, 6, 2, 6, 6, 3, 1, 7, 6, 3, 4])
```

```
#validation test
print("random forest Accuracy is:",metrics.accuracy_score(y_test_cv2, rf_pred_cv2))
print("random forest f1 score is :",f1_score(y_test_cv2, rf_pred_cv2, average='macro'))
print("random forest recall score is :",recall_score(y_test_cv2, rf_pred_cv2, average='macro'))
print("random forest precison score is :",precision_score(y_test_cv2, rf_pred_cv2, average='macro'
))
print("random forest precison score is :",precision_score(y_test_cv2, rf_pred_cv2, average='macro'
))
print("random forest matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, rf_pred_cv2))
```

Random forest classifier using TFIDF vectorizer

```
#fitting random forest classifier using count vectorizer
rf_n = RandomForestClassifier(n_estimators=200, random_state=0)
rf_n.fit(X_train_tf2, y_train_tf2)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=0, verbose=0,
                       warm_start=False)
```

```
#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_rf_df2 = cross_val_score(estimator=rf_n, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_rf_df2
```

```
#prediction test
rf_pred_tf2 = rf_n.predict(X_test_cv2)
rf_pred_tf2
```

```
array([2, 6, 1, 1, 7, 3, 3, 1, 4, 1, 3, 3, 3, 6, 1, 2, 1, 1, 1, 4, 6, 2,
       2, 3, 6, 5, 3, 4, 7, 4, 2, 3, 7, 4, 2, 3, 4, 7, 3, 5, 3, 6, 3, 6,
       3, 6, 6, 6, 1, 1, 4, 3, 3, 6, 6, 3, 1, 1, 7, 1, 1, 3, 6, 4, 4, 5,
       3, 1, 4, 6, 5, 3, 7, 6, 6, 2, 1, 3, 1, 3, 3, 2, 1, 7, 3, 7, 1, 1,
       4, 7, 6, 5, 2, 2, 2, 7, 1, 2, 4, 6, 6, 2, 4, 3, 3, 4, 6, 6, 1, 5,
       1, 4, 1, 6, 6, 2, 5, 1, 7, 3, 1, 3, 2, 3, 2, 2, 6, 2, 5, 3, 3, 3,
       7, 3, 3, 2, 3, 1, 2, 7, 3, 4, 6, 6, 1, 3, 7, 3, 1, 5, 7, 2, 6, 6,
       3, 2, 1, 7, 1, 5, 6, 1, 6, 7, 3, 5, 3, 7, 3, 1, 1, 6, 3, 2, 6, 3,
       2, 3, 3, 2, 4, 3, 6, 3, 1, 4, 3, 7, 4, 1, 5, 7, 4, 7, 1, 6, 3, 2,
       6, 3, 1, 1, 3, 1, 2, 2, 3, 4, 7, 3, 7, 3, 7, 6, 6, 4, 6, 1, 3, 3,
       2, 7, 6, 3, 3, 6, 3, 6, 2, 1, 6, 7, 2, 7, 1, 3, 3, 1, 3, 6, 6, 7,
       5, 6, 6, 3, 1, 1, 4, 2, 2, 3, 2, 6, 6, 6, 1, 1, 4, 6, 4, 2, 6, 3,
       7, 1, 6, 1, 3, 1, 3, 6, 4, 4, 3, 7, 7, 6, 6, 3, 3, 3, 3, 3, 2, 1,
       7, 6, 4, 6, 5, 7, 3, 3, 7, 1, 3, 4, 3, 7, 3, 5, 2, 2, 4, 5, 3, 6,
       2, 3, 3, 2, 1, 5, 3, 6, 6, 3, 1, 6, 6, 1, 1, 7, 5, 5, 3, 1, 3, 2,
       1, 7, 6, 3, 6, 3, 6, 3, 3, 6, 2, 2, 3, 1, 6, 5, 3, 4, 1, 5, 7, 2,
       3, 3, 7, 4, 4, 6, 2, 1, 6, 6, 3, 2, 6, 6, 2, 3, 7, 3, 1, 7, 3, 1,
       4, 4, 3, 2, 6, 3, 3, 4, 3, 3, 6, 3, 6, 3, 1, 1, 3, 2, 3, 1, 3, 6])
```

```

5, 1, 7, 4, 7, 1, 6, 6, 1, 1, 2, 6, 1, 2, 2, 6, 6, 1, 1, 4, 3, 7,
1, 4, 1, 3, 4, 3, 7, 6, 1, 4, 6, 6, 6, 2, 6, 3, 3, 3, 4, 3, 7,
3, 7, 1, 3, 6, 1, 1, 2, 1, 6, 4, 1, 6, 3, 3, 5, 2, 6, 2, 2, 7, 1,
3, 6, 3, 7, 3, 7, 4, 3, 6, 5, 2, 4, 6, 4, 6, 3, 6, 4, 4, 1, 4, 6,
3, 3, 1, 1, 7, 3, 1, 6, 6, 3, 6, 4, 5, 3, 4, 7, 7, 6, 7, 4, 3, 4,
1, 7, 3, 1, 3, 2, 7, 2, 3, 3, 7, 1, 6, 6, 6, 7, 3, 6, 3, 6, 3, 3,
6, 3, 3, 3, 7, 6, 1, 1, 2, 3, 4, 3, 6, 6, 3, 5, 1, 2, 1, 3, 6, 3,
5, 4, 6, 6, 1, 3, 3, 2, 6, 7, 4, 7, 3, 3, 3, 6, 6, 2, 4, 6, 1, 3,
2, 2, 2, 5, 2, 6, 1, 4, 3, 1, 3, 2, 5, 4, 2, 6, 2, 3, 1, 6, 2, 2,
3, 2, 7, 3, 4, 4, 6, 1, 2, 2, 1, 3, 3, 3, 2, 1, 3, 2, 3, 6, 1, 3,
6, 6, 1, 1, 2, 3, 3, 6, 6, 6, 3, 2, 3, 1, 1, 6, 3, 1, 3, 6, 6, 1,
3, 3, 4, 3, 3, 4, 2, 5, 7, 3, 3, 2, 4, 6, 7, 3, 6, 3, 1, 2, 2, 3,
6, 1, 3, 3, 7, 4, 1, 2, 7, 4, 1, 6, 3, 2, 3, 3, 2, 3, 2, 4, 3, 3,
7, 4, 3, 3, 6, 1, 2, 6, 6, 2, 2, 7, 2, 3, 1, 7, 5, 6, 1, 6, 7, 7,
1, 6, 7, 1, 2, 4, 1, 4, 6, 4, 1, 1, 6, 1, 1, 3, 7, 6, 2, 3, 7, 5,
7, 1, 6, 6, 2, 7, 6, 4, 6, 3, 6, 6, 1, 3, 3, 4, 1, 7, 3, 6, 3, 1,
1, 2, 6, 2, 2, 7, 7, 1, 7, 3, 1, 6, 4, 1, 3, 2, 3, 3, 3, 1, 6, 7,
2, 3, 4, 6, 1, 4, 3, 1, 3, 4, 2, 6, 4, 6, 6, 6, 1, 6, 7, 1, 3, 6,
1, 5, 3, 3, 4, 7, 6, 3, 1, 6, 3, 6, 6, 7, 1, 3, 7, 1, 1, 3, 5, 2,
4, 1, 7, 1, 4, 7, 6, 4, 1, 6, 2, 3, 3, 6, 4, 2, 3, 2, 2, 2, 3, 4,
7, 3, 6, 7, 4, 3, 2, 3, 7, 7, 2, 1, 3, 3, 6, 1, 4, 7, 3, 3, 6, 5,
2, 1, 4, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 3, 3, 6, 1, 6, 3, 7, 4, 3,
2, 3, 1, 2, 2, 4, 6, 3, 6, 7, 6, 3, 3, 4, 7, 3, 2, 3, 1, 3, 3, 2,
2, 1, 3, 4, 6, 2, 2, 2, 6, 1, 1, 2, 3, 3, 6, 3, 5, 6, 3, 2, 6, 5,
1, 1, 3, 7, 7, 7, 7, 6, 3, 6, 2, 2, 1, 3, 7, 1, 6, 6, 6, 4, 2, 7,
1, 7, 6, 3, 6, 1, 6, 3, 6, 1, 1, 1, 4, 6, 3, 1, 2, 6, 6, 6, 6, 7,
6, 6, 7, 2, 6, 6, 3, 3, 7, 6, 3, 4])

```

In [79]:

```

#validation test
print("random forest Accuracy is:",metrics.accuracy_score(y_test_tf2, rf_pred_tf2))
print("random forest f1 score is :",f1_score(y_test_tf2, rf_pred_tf2, average='macro'))
print("random forest recall score is :",recall_score(y_test_tf2, rf_pred_tf2, average='macro'))
print("random forest precision score is :",precision_score(y_test_tf2, rf_pred_tf2, average='macro'
))
print("random forest precision score is :",precision_score(y_test_tf2, rf_pred_tf2, average='macro'
))
print("random forest matthews_corrcoef score is :",matthews_corrcoef(y_test_tf2, rf_pred_tf2))

```

```

random forest Accuracy is: 0.6122448979591837
random forest f1 score is : 0.5942109823467926
random forest recall score is : 0.612229324798556
random forest precision score is : 0.6461609688454512
random forest precision score is : 0.6461609688454512
random forest matthews_corrcoef score is : 0.5570815362776147

```

DECESSION TREE CLASSIFIER USING COUNT VECTORIZER

In [87]:

```

#fitting decession tree using count vectorizer
from sklearn.tree import DecisionTreeClassifier
dt_n = DecisionTreeClassifier()
dt_n.fit(X_train_cv2,y_train_cv2)

```

Out[87]:

```

DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                        max_depth=None, max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort='deprecated',
                        random_state=None, splitter='best')

```

In []:

```

#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_dt_cv2 = cross_val_score(estimator=dt_n, X=X_train_cv2, y=y_train_cv2, cv=10)
cross_dt_cv2

```

In [91]:

```
#prediction test
dt_pred_cv2 = dt_n.predict(X_test_cv2)
dt_pred_cv2
```

Out[91]:

```
array([2, 6, 1, 1, 5, 3, 3, 5, 4, 1, 3, 3, 5, 6, 1, 2, 1, 1, 1, 4, 7, 2,
       2, 5, 6, 1, 3, 6, 7, 4, 2, 6, 6, 4, 2, 3, 3, 7, 3, 6, 4, 7, 3, 6,
       3, 6, 6, 1, 1, 1, 4, 3, 3, 3, 6, 3, 1, 1, 7, 1, 1, 3, 6, 4, 4, 1,
       3, 1, 4, 2, 5, 3, 7, 6, 7, 2, 1, 5, 1, 4, 3, 4, 1, 7, 3, 6, 1, 1,
       4, 7, 3, 5, 2, 2, 2, 7, 1, 2, 4, 6, 6, 2, 4, 2, 5, 4, 7, 4, 1, 1,
       1, 4, 1, 4, 5, 6, 1, 5, 7, 3, 5, 3, 2, 5, 2, 6, 6, 2, 5, 2, 3, 3,
       7, 5, 3, 2, 2, 1, 5, 7, 2, 4, 6, 4, 1, 3, 7, 3, 1, 5, 7, 2, 6, 7,
       2, 2, 1, 7, 1, 5, 6, 1, 6, 7, 5, 5, 3, 7, 3, 1, 5, 6, 2, 6, 5, 2,
       4, 6, 3, 2, 4, 5, 7, 3, 1, 4, 3, 6, 4, 1, 5, 7, 4, 7, 1, 5, 3, 2,
       6, 4, 5, 1, 3, 1, 2, 2, 3, 4, 7, 3, 4, 3, 7, 6, 6, 4, 4, 1, 5, 2,
       5, 7, 6, 3, 3, 7, 4, 6, 2, 1, 4, 7, 5, 7, 4, 3, 3, 6, 3, 7, 6, 7,
       5, 6, 1, 3, 1, 5, 4, 2, 2, 2, 6, 6, 6, 5, 5, 4, 7, 4, 2, 5, 3,
       7, 1, 4, 5, 5, 1, 5, 6, 4, 4, 2, 7, 7, 6, 5, 3, 3, 3, 3, 2, 5,
       7, 6, 4, 6, 5, 6, 3, 1, 7, 5, 3, 2, 7, 7, 5, 5, 2, 2, 5, 1, 3, 6,
       2, 2, 3, 2, 5, 1, 3, 4, 6, 2, 5, 6, 6, 5, 1, 7, 5, 5, 3, 1, 3, 6,
       1, 7, 6, 3, 6, 3, 6, 3, 3, 6, 6, 2, 3, 1, 6, 5, 3, 4, 5, 5, 7, 2,
       3, 3, 7, 4, 4, 6, 2, 1, 7, 6, 1, 2, 6, 4, 6, 2, 7, 3, 1, 2, 3, 1,
       4, 4, 3, 7, 1, 3, 3, 4, 5, 5, 6, 3, 4, 5, 5, 1, 3, 2, 3, 5, 3, 2,
       5, 5, 7, 4, 7, 1, 6, 6, 1, 1, 2, 6, 5, 2, 2, 7, 7, 5, 1, 4, 3, 7,
       1, 4, 1, 3, 4, 5, 7, 6, 1, 4, 4, 6, 7, 4, 1, 3, 3, 1, 3, 4, 3, 7,
       5, 7, 4, 3, 6, 1, 1, 2, 1, 3, 4, 1, 6, 3, 3, 5, 5, 6, 2, 2, 7, 1,
       3, 1, 4, 2, 4, 7, 4, 3, 6, 5, 2, 4, 6, 4, 7, 3, 7, 4, 6, 1, 4, 6,
       3, 3, 1, 5, 7, 3, 5, 6, 6, 4, 6, 5, 5, 4, 4, 7, 7, 6, 7, 4, 3, 4,
       1, 7, 5, 1, 2, 2, 7, 2, 3, 3, 7, 1, 7, 6, 2, 7, 3, 6, 3, 3, 3, 3,
       6, 5, 3, 3, 7, 7, 1, 5, 2, 3, 4, 3, 6, 4, 3, 5, 1, 6, 1, 2, 3, 5,
       5, 4, 6, 6, 1, 3, 3, 2, 6, 7, 4, 7, 3, 3, 3, 6, 2, 2, 4, 6, 5, 3,
       2, 2, 2, 5, 2, 7, 1, 4, 5, 1, 3, 2, 1, 4, 2, 2, 2, 3, 4, 6, 2, 2,
       3, 6, 7, 3, 4, 4, 6, 5, 2, 2, 1, 3, 1, 3, 2, 4, 3, 2, 3, 3, 1, 2,
       7, 6, 5, 5, 2, 3, 3, 7, 6, 4, 3, 2, 3, 1, 1, 4, 3, 5, 2, 7, 6, 5,
       3, 4, 4, 5, 3, 4, 2, 5, 7, 3, 2, 6, 4, 6, 7, 3, 6, 4, 1, 2, 6, 3,
       1, 5, 2, 3, 6, 4, 5, 2, 7, 4, 1, 4, 2, 2, 2, 3, 5, 3, 2, 4, 3, 3,
       7, 4, 2, 1, 7, 1, 2, 6, 6, 6, 7, 2, 3, 5, 7, 4, 1, 1, 6, 7, 7,
       1, 4, 7, 1, 2, 4, 1, 4, 6, 4, 6, 1, 7, 1, 1, 5, 7, 5, 5, 3, 7, 1,
       7, 1, 6, 6, 2, 7, 2, 4, 6, 3, 6, 7, 1, 3, 2, 4, 4, 7, 2, 6, 4, 1,
       1, 2, 6, 2, 2, 6, 2, 5, 7, 2, 1, 4, 6, 5, 3, 2, 3, 2, 3, 1, 7, 7,
       2, 5, 4, 6, 1, 4, 3, 1, 3, 4, 2, 6, 4, 6, 6, 7, 5, 3, 7, 1, 2, 6,
       1, 5, 2, 3, 4, 7, 6, 3, 1, 6, 4, 4, 6, 7, 1, 3, 7, 5, 1, 5, 1, 2,
       4, 5, 7, 1, 6, 7, 6, 6, 1, 6, 2, 3, 6, 6, 4, 2, 3, 2, 2, 2, 3, 4,
       7, 3, 6, 7, 4, 3, 2, 2, 7, 7, 2, 1, 3, 3, 4, 5, 4, 7, 3, 4, 7, 5,
       2, 5, 4, 7, 1, 6, 7, 1, 6, 7, 7, 3, 2, 3, 3, 7, 1, 6, 3, 7, 4, 3,
       6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 5, 3, 3, 4, 5, 2, 2, 2, 5, 3, 2, 2,
       2, 5, 2, 4, 6, 2, 5, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 6, 3, 2, 6, 5,
       6, 1, 3, 6, 7, 7, 7, 6, 3, 6, 2, 2, 1, 4, 7, 1, 6, 6, 7, 4, 2, 7,
       1, 7, 4, 3, 2, 1, 6, 4, 6, 1, 1, 1, 4, 6, 3, 1, 6, 6, 6, 7, 6, 7,
       2, 6, 3, 2, 6, 6, 3, 4, 7, 6, 3, 4])
```

In [95]:

```
#validation test
print("decision tree Accuracy is:",metrics.accuracy_score(y_test_cv2, dt_pred_cv2))
print("decision tree f1 score is :",f1_score(y_test_cv2, dt_pred_cv2, average='macro'))
print("decision tree recall score is :",recall_score(y_test_cv2, dt_pred_cv2, average='macro'))
print("decision tree precision score is :",precision_score(y_test_cv2, dt_pred_cv2, average='macro')
)
print("decision tree precision score is :",precision_score(y_test_cv2, dt_pred_cv2, average='macro')
)
print("decision tree matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, dt_pred_cv2))
```

```
decision tree Accuracy is: 0.673469387755102
decision tree f1 score is : 0.67023866344529
decision tree recall score is : 0.6747894484346284
decision tree precision score is : 0.6717283515891455
decision tree precision score is : 0.6717283515891455
decision tree matthews_corrcoef score is : 0.6199743803072449
```

DECESSION TREE CLASSIFIER USING TFIDF

In [93]:

```
#fitting decession tree using tfidf vectorizer
from sklearn.tree import DecisionTreeClassifier
dt_n = DecisionTreeClassifier()
dt_n.fit(X_train_tf2,y_train_tf2)
```

Out[93]:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                        max_depth=None, max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort='deprecated',
                        random_state=None, splitter='best')
```

In []:

```
#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_dt_cv2 = cross_val_score(estimator=dt_n, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_st_cv2
```

In [94]:

```
#prediction test
dt_pred_tf2 = dt_n.predict(X_test_tf2)
dt_pred_tf2
```

Out[94]:

```
array([6, 6, 1, 1, 7, 3, 3, 1, 4, 1, 3, 2, 5, 6, 5, 2, 1, 5, 1, 4, 6, 4,
        4, 5, 6, 5, 3, 4, 7, 4, 4, 6, 7, 4, 2, 3, 4, 7, 3, 1, 2, 6, 3, 6,
        6, 6, 2, 1, 1, 5, 4, 3, 3, 6, 4, 3, 1, 5, 7, 1, 1, 3, 6, 4, 4, 5,
        6, 1, 4, 2, 2, 2, 7, 6, 7, 4, 1, 2, 1, 3, 3, 2, 1, 7, 3, 7, 1, 1,
        5, 7, 4, 5, 2, 4, 2, 7, 1, 2, 4, 6, 3, 2, 4, 2, 5, 4, 7, 4, 1, 1,
        1, 5, 5, 4, 6, 6, 1, 5, 7, 3, 5, 3, 2, 4, 2, 2, 6, 2, 5, 2, 3, 3,
        7, 4, 3, 2, 4, 1, 4, 7, 2, 4, 6, 6, 6, 2, 7, 3, 5, 5, 7, 2, 6, 7,
        2, 2, 1, 7, 1, 1, 6, 5, 6, 7, 5, 5, 2, 7, 3, 1, 5, 6, 2, 6, 4, 7,
        6, 6, 3, 2, 5, 2, 7, 3, 1, 6, 2, 6, 4, 1, 5, 7, 4, 7, 5, 4, 3, 2,
        6, 2, 1, 1, 3, 5, 2, 2, 3, 4, 7, 3, 7, 3, 7, 6, 6, 4, 4, 1, 1, 6,
        5, 7, 6, 3, 2, 6, 2, 6, 2, 1, 4, 7, 5, 7, 5, 3, 3, 6, 2, 7, 6, 7,
        5, 6, 1, 3, 1, 5, 4, 2, 2, 2, 2, 6, 2, 1, 1, 1, 5, 7, 4, 4, 4, 3,
        7, 5, 4, 5, 5, 1, 5, 5, 5, 4, 2, 7, 7, 6, 4, 3, 3, 3, 3, 3, 6, 1,
        7, 6, 4, 6, 5, 6, 3, 1, 7, 1, 2, 4, 4, 7, 1, 5, 2, 7, 4, 1, 3, 6,
        2, 3, 2, 2, 5, 5, 3, 1, 6, 2, 1, 4, 6, 1, 1, 7, 5, 1, 3, 1, 3, 6,
        1, 7, 6, 3, 4, 2, 4, 3, 3, 6, 2, 2, 3, 1, 6, 5, 3, 4, 5, 5, 7, 4,
        3, 4, 7, 4, 4, 6, 2, 1, 7, 6, 5, 2, 6, 4, 6, 1, 7, 2, 1, 4, 3, 1,
        4, 4, 3, 2, 1, 3, 3, 4, 5, 6, 2, 3, 4, 4, 1, 1, 3, 2, 4, 5, 3, 6,
        1, 1, 7, 4, 7, 5, 6, 6, 1, 1, 2, 6, 5, 2, 4, 7, 6, 5, 1, 4, 3, 7,
        1, 4, 5, 3, 4, 5, 6, 6, 1, 4, 4, 6, 7, 4, 1, 3, 3, 1, 3, 4, 3, 7,
        5, 7, 4, 3, 6, 1, 1, 3, 1, 3, 4, 1, 6, 4, 3, 5, 2, 6, 2, 2, 7, 1,
        3, 5, 3, 6, 4, 6, 4, 3, 5, 5, 2, 1, 6, 4, 6, 3, 7, 4, 5, 1, 6, 6,
        2, 3, 1, 5, 7, 3, 1, 6, 6, 4, 6, 5, 5, 4, 4, 7, 7, 6, 7, 4, 3, 5,
        1, 7, 5, 5, 2, 2, 7, 6, 3, 3, 7, 1, 1, 7, 7, 7, 3, 6, 3, 4, 3, 4,
        6, 2, 2, 3, 7, 7, 1, 5, 2, 3, 4, 3, 6, 6, 3, 5, 5, 2, 1, 2, 4, 1,
        5, 4, 6, 1, 1, 3, 3, 2, 6, 7, 4, 7, 2, 3, 3, 6, 6, 2, 4, 6, 5, 3,
        2, 6, 2, 1, 2, 7, 1, 4, 5, 1, 3, 2, 5, 4, 2, 4, 2, 3, 1, 6, 2, 2,
        3, 6, 7, 3, 5, 6, 6, 5, 3, 2, 1, 3, 1, 3, 2, 4, 3, 6, 3, 7, 5, 2,
        7, 4, 1, 5, 2, 3, 6, 6, 6, 2, 3, 1, 3, 5, 1, 2, 3, 5, 2, 6, 2, 5,
        3, 3, 4, 5, 3, 4, 2, 5, 6, 3, 2, 6, 6, 6, 7, 6, 2, 2, 5, 5, 6, 3,
        1, 5, 5, 3, 7, 4, 5, 5, 7, 5, 5, 4, 2, 2, 2, 3, 5, 2, 6, 4, 2, 3,
        7, 4, 3, 1, 7, 1, 6, 6, 6, 7, 6, 7, 4, 3, 5, 7, 6, 1, 5, 6, 7, 7,
        1, 4, 7, 1, 2, 4, 1, 4, 6, 4, 6, 1, 6, 1, 5, 5, 7, 3, 2, 3, 7, 5,
        7, 1, 6, 6, 2, 7, 6, 4, 6, 3, 6, 6, 1, 3, 2, 4, 4, 7, 4, 4, 3, 1,
        5, 7, 6, 2, 2, 6, 7, 5, 6, 2, 1, 5, 6, 6, 3, 6, 2, 2, 2, 1, 7, 7,
        2, 5, 4, 6, 1, 4, 3, 1, 3, 4, 2, 6, 4, 6, 7, 7, 5, 6, 7, 1, 2, 6,
        5, 5, 2, 3, 4, 7, 6, 3, 1, 6, 4, 2, 6, 7, 1, 3, 7, 1, 1, 5, 5, 6,
        4, 5, 7, 1, 6, 7, 6, 6, 5, 6, 2, 2, 4, 7, 4, 2, 3, 2, 2, 4, 3, 6,
        7. 5. 6. 7. 4. 3. 2. 2. 7. 7. 4. 1. 4. 3. 4. 1. 4. 7. 3. 6. 7. 5.]
```



```

2, 5, 6, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 5, 4, 7, 1, 6, 3, 7, 4, 3,
6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 5, 3, 3, 4, 7, 2, 2, 2, 5, 6, 2, 7,
2, 5, 5, 4, 6, 1, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 6, 3, 2, 6, 5,
5, 5, 3, 4, 7, 7, 7, 6, 3, 6, 2, 2, 1, 2, 7, 1, 6, 6, 7, 4, 2, 7,
1, 7, 5, 3, 2, 1, 6, 3, 6, 1, 1, 1, 4, 6, 6, 1, 6, 6, 4, 7, 6, 7,
2, 7, 7, 2, 6, 6, 3, 1, 7, 4, 3, 4])

```

In [96]:

```

#validation test
print("decision tree Accuracy is:",metrics.accuracy_score(y_test_tf2, dt_pred_tf2))
print("decision tree f1 score is :",f1_score(y_test_tf2, dt_pred_tf2, average='macro'))
print("decision tree recall score is :",recall_score(y_test_tf2, dt_pred_tf2, average='macro'))
print("decision tree precison score is :",precision_score(y_test_tf2, dt_pred_tf2, average='macro')
)
print("decision tree precison score is :",precision_score(y_test_tf2, dt_pred_tf2, average='macro')
)
print("decision tree matthews_corrcoef score is :",matthews_corrcoef(y_test_tf2, dt_pred_tf2))

```

```

decision tree Accuracy is: 0.6663265306122449
decision tree f1 score is : 0.6671628641684618
decision tree recall score is : 0.6674119260489599
decision tree precison score is : 0.6696961263232828
decision tree precison score is : 0.6696961263232828
decision tree matthews_corrcoef score is : 0.6111665287077429

```

LOGESTIC REGRESSION USING COUNT VECTORIZER

In [106]:

```

from sklearn.linear_model import LogisticRegression
lr_n = LogisticRegression(random_state=0,solver='lbfgs')
lr_n.fit(X_train_cv2,y_train_cv2)

```

Out[106]:

```

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=0, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)

```

In []:

```

#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_lr_cv2 = cross_val_score(estimator=lr_n, X=X_train_tf2, y=y_train_tf2, cv=10)
cross_lr_cv2

```

In [107]:

```

#prediction test
lr_pred_cv2 = lr_n.predict(X_test_cv2)
lr_pred_cv2

```

Out[107]:

```

array([2, 6, 1, 1, 7, 3, 3, 4, 4, 1, 3, 3, 5, 6, 5, 2, 1, 5, 1, 4, 7, 2,
        6, 5, 6, 5, 3, 4, 7, 4, 2, 6, 7, 4, 2, 3, 4, 7, 3, 5, 4, 6, 3, 6,
        3, 6, 6, 1, 1, 5, 4, 3, 3, 6, 6, 3, 1, 5, 7, 1, 1, 3, 6, 4, 4, 5,
        3, 1, 4, 6, 5, 3, 7, 2, 6, 2, 1, 5, 1, 4, 3, 2, 1, 7, 3, 7, 1, 1,
        5, 7, 4, 5, 2, 4, 2, 7, 1, 5, 4, 6, 6, 2, 4, 2, 5, 4, 7, 4, 5, 1,
        1, 4, 1, 4, 5, 6, 5, 5, 6, 3, 5, 3, 2, 4, 2, 2, 7, 2, 5, 2, 3, 6,
        7, 4, 3, 2, 2, 1, 4, 7, 2, 4, 6, 4, 1, 2, 7, 3, 5, 4, 7, 6, 6, 7,
        2, 2, 1, 7, 1, 1, 6, 5, 6, 7, 2, 5, 4, 7, 2, 1, 1, 6, 2, 6, 4, 3,
        2, 4, 3, 2, 2, 2, 7, 3, 1, 4, 4, 7, 4, 1, 5, 7, 4, 7, 5, 6, 3, 2,
        6, 4, 5, 1, 3, 1, 2, 2, 3, 4, 7, 4, 7, 3, 7, 6, 6, 4, 4, 1, 5, 4,
        5, 7, 7, 3, 3, 6, 2, 6, 6, 1, 4, 7, 5, 7, 5, 3, 3, 6, 2, 7, 6, 7,
        5, 6, 6, 3, 1, 1, 4, 2, 2, 2, 2, 6, 6, 6, 1, 1, 4, 6, 4, 2, 4, 3,
        7, 5, 4, 5, 2, 1, 5, 7, 4, 4, 2, 7, 7, 6, 4, 3, 3, 3, 3, 3, 6, 5,

```

```

7, 6, 4, 6, 5, 7, 3, 1, 7, 1, 3, 6, 7, 6, 5, 5, 2, 2, 4, 5, 3, 6,
2, 2, 2, 2, 5, 6, 3, 4, 6, 2, 1, 6, 6, 1, 5, 7, 5, 1, 3, 1, 3, 2,
1, 7, 7, 3, 7, 3, 4, 3, 3, 6, 6, 2, 3, 1, 6, 5, 3, 4, 1, 5, 7, 2,
3, 4, 7, 4, 4, 6, 5, 5, 7, 6, 5, 2, 6, 6, 6, 2, 7, 3, 1, 7, 3, 1,
4, 4, 3, 6, 1, 3, 3, 4, 5, 5, 6, 3, 6, 5, 5, 1, 3, 2, 3, 5, 3, 6,
5, 5, 7, 4, 7, 1, 6, 6, 1, 1, 2, 6, 5, 2, 4, 6, 6, 5, 5, 4, 3, 7,
1, 4, 5, 3, 4, 2, 7, 6, 1, 4, 4, 6, 7, 5, 1, 3, 3, 5, 3, 4, 3, 7,
5, 7, 1, 3, 6, 1, 1, 2, 1, 3, 4, 1, 4, 4, 3, 5, 2, 6, 2, 2, 7, 1,
3, 1, 3, 7, 4, 7, 4, 3, 4, 5, 2, 4, 6, 4, 7, 3, 7, 4, 4, 1, 4, 6,
3, 3, 1, 5, 7, 3, 1, 4, 6, 4, 6, 5, 5, 4, 4, 7, 7, 7, 7, 4, 3, 5,
1, 7, 5, 5, 2, 2, 7, 1, 3, 3, 7, 1, 5, 6, 7, 7, 3, 6, 3, 6, 3, 3,
6, 5, 3, 3, 2, 7, 1, 5, 2, 3, 4, 3, 6, 6, 3, 5, 1, 3, 1, 2, 4, 2,
5, 4, 6, 4, 1, 3, 3, 2, 6, 7, 4, 7, 3, 3, 3, 6, 6, 2, 6, 6, 5, 3,
2, 4, 2, 1, 2, 6, 1, 4, 5, 1, 3, 2, 5, 4, 2, 4, 2, 3, 5, 6, 2, 2,
3, 2, 7, 3, 4, 4, 7, 5, 2, 2, 1, 3, 5, 3, 2, 1, 3, 2, 3, 5, 5, 5,
6, 6, 1, 5, 2, 3, 3, 6, 6, 6, 3, 6, 3, 1, 1, 6, 3, 1, 2, 6, 6, 5,
3, 4, 4, 3, 3, 4, 6, 5, 7, 6, 2, 6, 4, 6, 7, 6, 6, 3, 1, 2, 2, 3,
1, 5, 2, 3, 7, 4, 5, 5, 7, 4, 1, 6, 2, 2, 2, 3, 2, 3, 2, 4, 3, 3,
7, 4, 2, 5, 7, 1, 2, 6, 7, 6, 4, 7, 4, 3, 5, 7, 6, 6, 5, 6, 7, 7,
1, 6, 7, 1, 6, 4, 1, 4, 6, 4, 4, 1, 6, 1, 5, 5, 7, 6, 5, 3, 7, 1,
7, 1, 6, 2, 2, 7, 2, 4, 6, 3, 6, 6, 1, 3, 2, 4, 5, 7, 7, 6, 6, 1,
1, 6, 6, 2, 2, 7, 7, 1, 7, 2, 1, 5, 6, 1, 3, 6, 3, 2, 3, 1, 7, 7,
2, 4, 4, 6, 1, 4, 3, 1, 3, 4, 2, 6, 4, 6, 6, 7, 5, 2, 7, 1, 3, 6,
5, 5, 2, 3, 4, 7, 6, 3, 1, 6, 4, 4, 6, 7, 1, 3, 7, 1, 1, 5, 5, 2,
4, 5, 7, 1, 6, 7, 6, 2, 5, 6, 2, 3, 2, 6, 4, 2, 3, 7, 2, 5, 3, 4,
4, 3, 6, 7, 4, 4, 2, 2, 7, 7, 2, 1, 3, 3, 6, 7, 4, 7, 3, 6, 7, 5,
2, 5, 4, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 3, 6, 6, 1, 6, 3, 7, 4, 3,
6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 5, 3, 3, 4, 7, 2, 2, 3, 5, 3, 2, 2,
5, 5, 2, 4, 7, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 6, 3, 2, 6, 5,
1, 5, 3, 7, 7, 7, 7, 7, 3, 6, 5, 2, 1, 3, 7, 1, 6, 6, 6, 4, 2, 7,
1, 7, 6, 3, 4, 1, 6, 4, 6, 1, 1, 1, 4, 6, 6, 1, 2, 6, 4, 6, 6, 7,
4, 6, 7, 4, 6, 7, 3, 1, 7, 4, 3, 4])

```

In [108]:

```

#validation test
print("logistic regression Accuracy is:",metrics.accuracy_score(y_test_cv2, lr_pred_cv2))
print("logistic regreesion score is :",f1_score(y_test_cv2, lr_pred_cv2, average='macro'))
print("logistic regreesion score is :",recall_score(y_test_cv2, lr_pred_cv2, average='macro'))
print("logistic regreesion score is :",precision_score(y_test_cv2, lr_pred_cv2, average='macro'))
print("logistic regreesion score is :",precision_score(y_test_cv2, lr_pred_cv2, average='macro'))
print("logistic regreesion matthews_corrcoef score is :",matthews_corrcoef(y_test_cv2, lr_pred_cv2
))

```

```

logistic regression Accuracy is: 0.7673469387755102
logistic regression score is : 0.7681950034201481
logistic regreesion score is : 0.7684522812741478
logistic regreesion score is : 0.7726398997345302
logistic regreesion score is : 0.7726398997345302
logistic regreesion matthews_corrcoef score is : 0.7293009854701492

```

LOGISTIC VECTORIZER USING TFIDF

In [109]:

```

#fitting the model
lr_n.fit(X_train_tf2,y_train_tf2)

```

Out[109]:

```

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=0, solver='lbfgs', tol=0.0001, verbose=0,
                    warm_start=False)

```

In []:

```

#cross validation on training set
from sklearn.model_selection import cross_val_score
cross_lr_tf2 = cross_val_score(estimator=lr_n, X=X_train_tf2, y=y_train_tf2, cv=10)

```

```
cross_lr_tf2
```

```
In [110]:
```

```
#prediction test
lr_pred_tf2 = lr_n.predict(X_test_tf2)
lr_pred_tf2
```

```
Out[110]:
```

```
array([2, 6, 1, 1, 7, 3, 4, 4, 4, 4, 3, 3, 5, 6, 5, 6, 1, 5, 1, 4, 7, 5,
       4, 5, 6, 5, 3, 4, 7, 4, 6, 6, 7, 4, 2, 3, 4, 7, 3, 5, 4, 6, 3, 6,
       2, 6, 6, 6, 1, 5, 6, 3, 3, 6, 6, 3, 1, 5, 7, 1, 1, 3, 6, 4, 4, 5,
       3, 1, 4, 6, 5, 3, 7, 6, 6, 2, 1, 5, 1, 4, 3, 2, 1, 7, 3, 7, 1, 1,
       5, 7, 4, 5, 2, 4, 2, 7, 1, 2, 4, 6, 6, 2, 4, 2, 5, 4, 6, 4, 5, 4,
       5, 4, 1, 4, 6, 6, 5, 5, 7, 3, 5, 3, 2, 4, 6, 2, 7, 2, 5, 2, 3, 6,
       7, 4, 3, 2, 2, 5, 4, 7, 2, 4, 7, 4, 1, 2, 7, 3, 5, 4, 7, 2, 6, 7,
       2, 2, 1, 7, 1, 6, 6, 5, 6, 7, 2, 5, 4, 7, 2, 1, 1, 6, 2, 6, 4, 3,
       4, 6, 3, 2, 2, 2, 7, 3, 1, 4, 4, 7, 4, 1, 5, 7, 4, 7, 5, 6, 3, 5,
       6, 4, 1, 1, 3, 1, 2, 2, 3, 4, 7, 4, 7, 3, 7, 6, 6, 4, 4, 1, 5, 4,
       2, 7, 7, 3, 3, 6, 2, 6, 7, 1, 4, 7, 5, 7, 5, 3, 3, 6, 2, 7, 6, 7,
       5, 6, 6, 3, 1, 1, 4, 2, 2, 2, 6, 6, 6, 1, 1, 4, 6, 4, 2, 4, 3,
       7, 5, 4, 5, 2, 1, 5, 6, 4, 4, 2, 7, 7, 6, 6, 3, 3, 6, 3, 3, 6, 5,
       7, 6, 4, 6, 5, 7, 3, 1, 7, 1, 3, 6, 7, 7, 5, 5, 2, 6, 4, 5, 3, 6,
       2, 2, 2, 2, 5, 6, 3, 4, 6, 2, 5, 6, 6, 1, 5, 7, 1, 1, 3, 1, 3, 2,
       1, 7, 7, 3, 6, 4, 4, 3, 3, 6, 2, 2, 3, 1, 6, 5, 3, 4, 1, 5, 7, 6,
       3, 6, 7, 4, 4, 6, 5, 1, 7, 6, 5, 2, 6, 6, 6, 2, 7, 4, 1, 7, 3, 1,
       4, 4, 3, 7, 1, 3, 3, 4, 5, 6, 6, 3, 6, 4, 5, 1, 3, 2, 3, 5, 3, 6,
       1, 1, 7, 4, 7, 1, 6, 6, 1, 1, 2, 6, 1, 2, 4, 6, 6, 5, 5, 4, 3, 7,
       1, 4, 5, 3, 4, 2, 7, 6, 1, 4, 3, 6, 7, 2, 1, 3, 3, 5, 3, 4, 3, 7,
       5, 7, 1, 3, 6, 1, 1, 3, 1, 3, 4, 1, 4, 4, 3, 5, 5, 7, 2, 2, 7, 1,
       3, 1, 3, 7, 4, 7, 4, 3, 4, 2, 2, 4, 6, 4, 6, 3, 7, 4, 4, 1, 4, 6,
       4, 3, 1, 5, 7, 3, 1, 6, 6, 4, 6, 5, 5, 4, 4, 7, 7, 7, 7, 4, 3, 5,
       1, 7, 5, 5, 2, 2, 7, 1, 3, 3, 7, 1, 5, 6, 7, 7, 3, 6, 3, 6, 3, 3,
       6, 3, 3, 3, 7, 6, 1, 5, 5, 3, 4, 3, 6, 6, 3, 5, 1, 6, 1, 2, 6, 2,
       5, 4, 6, 5, 1, 3, 3, 2, 6, 7, 4, 7, 3, 3, 3, 6, 6, 2, 6, 6, 5, 3,
       2, 4, 2, 1, 2, 6, 5, 4, 5, 1, 3, 2, 5, 4, 2, 4, 2, 6, 5, 6, 2, 2,
       3, 2, 7, 3, 4, 4, 7, 1, 2, 2, 1, 3, 1, 3, 2, 5, 3, 2, 3, 5, 5, 5,
       6, 6, 5, 5, 2, 3, 3, 6, 6, 6, 3, 6, 4, 1, 1, 6, 3, 1, 2, 6, 7, 5,
       3, 4, 4, 5, 3, 4, 6, 5, 7, 6, 2, 5, 6, 6, 7, 6, 6, 3, 1, 2, 2, 3,
       1, 5, 2, 3, 7, 4, 5, 5, 7, 4, 5, 6, 2, 2, 2, 3, 2, 3, 6, 4, 3, 3,
       7, 4, 2, 5, 6, 1, 2, 6, 6, 6, 4, 7, 4, 3, 5, 7, 6, 5, 1, 6, 7, 7,
       1, 6, 7, 1, 6, 4, 1, 4, 6, 4, 4, 1, 6, 1, 5, 5, 7, 6, 2, 3, 7, 1,
       7, 1, 7, 6, 2, 7, 6, 4, 6, 3, 6, 6, 1, 3, 2, 4, 5, 7, 7, 6, 6, 1,
       1, 6, 6, 2, 2, 7, 7, 5, 7, 2, 1, 5, 6, 1, 3, 6, 3, 2, 4, 5, 7, 7,
       2, 4, 4, 6, 1, 4, 3, 1, 3, 4, 2, 7, 4, 6, 6, 7, 5, 6, 7, 1, 2, 6,
       5, 5, 2, 3, 4, 7, 7, 3, 1, 6, 4, 4, 6, 7, 1, 3, 7, 1, 1, 5, 6, 6,
       4, 5, 7, 1, 6, 7, 6, 4, 5, 6, 2, 6, 2, 6, 4, 2, 3, 7, 2, 5, 3, 4,
       4, 3, 6, 7, 4, 4, 2, 2, 7, 7, 2, 1, 3, 4, 6, 7, 4, 7, 3, 6, 6, 5,
       2, 5, 4, 7, 1, 6, 7, 1, 6, 7, 6, 3, 2, 5, 6, 6, 1, 6, 3, 7, 4, 3,
       6, 3, 1, 2, 2, 4, 6, 3, 6, 7, 7, 3, 3, 4, 7, 2, 2, 3, 5, 6, 2, 2,
       2, 5, 2, 4, 6, 2, 2, 2, 4, 1, 1, 2, 3, 3, 1, 3, 5, 6, 3, 2, 6, 2,
       1, 1, 3, 7, 7, 7, 7, 7, 3, 6, 2, 2, 1, 4, 7, 1, 6, 6, 7, 4, 2, 7,
       1, 7, 6, 3, 4, 1, 6, 4, 6, 1, 1, 4, 6, 6, 1, 6, 6, 6, 6, 7,
       4, 6, 7, 4, 6, 6, 3, 4, 7, 4, 3, 4])
```

```
In [111]:
```

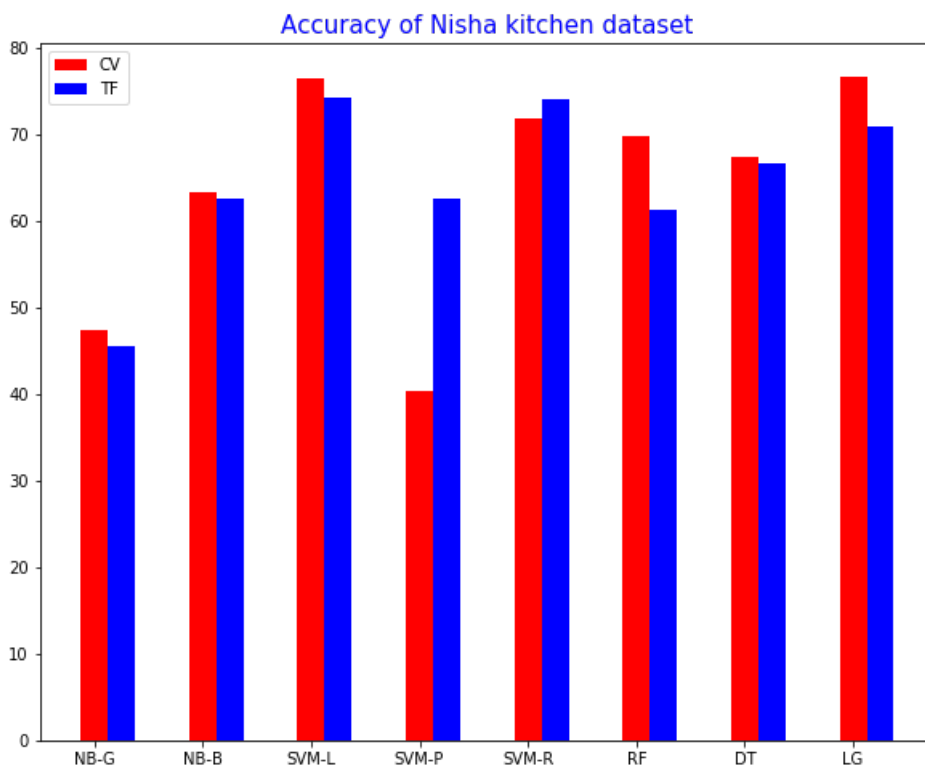
```
#validation test
print("logistic regression Accuracy is:",metrics.accuracy_score(y_test_tf2, lr_pred_cv2))
print("logistic regreesion score is :",f1_score(y_test_tf2, lr_pred_tf2, average='macro'))
print("logistic regreesion score is :",recall_score(y_test_tf2, lr_pred_tf2, average='macro'))
print("logistic regreesion score is :",precision_score(y_test_tf2, lr_pred_tf2, average='macro'))
print("logistic regreesion score is :",precision_score(y_test_tf2, lr_pred_tf2, average='macro'))
print("logistic regreesion matthews_corrcoef score is :",matthews_corrcoef(y_test_tf2, lr_pred_tf2))
```

```
logistic regression Accuracy is: 0.7673469387755102
logistic regreesion score is : 0.7495144687017662
logistic regreesion score is : 0.7499955792740627
logistic regreesion score is : 0.7570905725724024
logistic regreesion score is : 0.7570905725724024
logistic regreesion matthews_corrcoef score is : 0.7086511411686914
```

Visualizing the accuracies of Nisha kitchen data set

In [18]:

```
# We impletement bar graph using pypot from matplot library for visualization
# Here the visualization shows the accuracies of the machine learning model applied with both coun
t vectorizer and tfidf vectorizer
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(10,8))
X= ['CV','TF']
X= np.arange(8)
ACCURACY = [[47.4,63.3,76.4,40.3,71.8,69.8,67.3,76.7],[45.6,62.5,74.2,62.5,74.0,61.2,66.6,70.8]]
plt.bar(X+0.00,ACCURACY[0], color='r',width=0.25,label="CV")
plt.bar(X+0.25,ACCURACY[1], color='b',width=0.25,label="TF")
plt.xticks(X,['NB-G',"NB-B","SVM-L","SVM-P","SVM-R","RF","DT","LG"])
plt.legend(loc="upper left")
plt.title("Accuracy of Nisha kitchen dataset",{ "size":15,"color":"blue"})
plt.show()
```



In []: