```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

Problem 1: Build a machine Learning model to predict location from an address

```
df10=pd.read_excel("Problem_1_2_4_Dataset.xlsx")
```

```
df10.isnull().sum()

     FORMATTED_ADDRESS    0
     LOCALITY             0
     dtype: int64
```

```
df10["LOCALITY"].value_counts()

     Mumbai       70120
     Bengaluru    17288
     Hyderabad    11667
     Delhi         9639
     Chennai       9174
     Ahmedabad     7312
     Kolkata       7277
     Pune          6291
     pune             1
     Name: LOCALITY, dtype: int64
```

```
df10["LOCALITY"]=df10["LOCALITY"].replace("pune","Pune")
df10
```

Saved successfully! ×

	FORMATTED_ADDRESS	LOCALITY
0	Sumangal-project by Heritage group, Samarth Na...	Mumbai
1	#1, Shirke Layout, Kengeri Satellite Town, Ben...	Bengaluru
2	#2, Magadi Main Rd, Mariyappanapalya, Kempapur...	Bengaluru
3	#27/110, Govindapura Main Rd, Govindapura, Nag...	Bengaluru
4	#68, 2 Cross, Sri Venkateshwara Nagar Layout R...	Bengaluru
...	...	...
138764	Zelam, 2-A,, 2-A, Nagri Niwara Cooperative Hou...	Mumbai
138765	Zohra Colony, Shalibanda, Hyderabad, Telangana...	Hyderabad
138766	ZP Rd, Sriramana Colony, Hastinapuram, Hyderab...	Hyderabad
138767	Zuzart's 7, Deccan Paper Mill Rd, Magarpatta C...	Pune
138768	परमेश्वर धाम, परमेश्वर धाम opp सन्यास आश्र...	Mumbai

138769 rows × 2 columns

Libraries for text preproccesing

```
import nltk
from nltk.tokenize import word_tokenize
nltk.download("punkt")
from nltk.corpus import stopwords
nltk.download("stopwords")
from nltk.stem import PorterStemmer,WordNetLemmatizer
nltk.download("wordnet")
ps=PorterStemmer()
lemma=WordNetLemmatizer()

     [nltk_data] Downloading package punkt to /root/nltk_data...
     [nltk_data]   Package punkt is already up-to-date!
     [nltk_data] Downloading package stopwords to /root/nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
     [nltk_data] Downloading package wordnet to /root/nltk_data...
     [nltk_data]   Package wordnet is already up-to-date!
```

Converting Marathi to English

```
#regex to find marathi char in df
pattern=r"[\u0900-\u097F]"            #UNICODE FOR MARATHI CHARACTERS
marathi_address=df10[df10["FORMATTED_ADDRESS"].str.contains(pattern)]
marathi_address
```

	FORMATTED_ADDRESS	LOCALITY
1355	1, RC Marg, Ashok Nagar, कलेक्टर कॉलोनी, Chemb...	Mumbai
1503	1, Senetorium Ln, भट्टवाडी, Kapol wadi, Pant N...	Mumbai
2794	101, Collectors Colony Rd, कलेक्टर कॉलोनी, Che...	Mumbai
3292	102-B, Collectors Colony Rd, कलेक्टर कॉलोनी, C...	Mumbai
3576	104, कलेक्टर कॉलोनी, Chembur East, Mumbai, Mah...	Mumbai
...	...	...
137045	Ranjit Niwas, Chembur East, कलेक्टर कॉलोनी, Ku...	Mumbai
137163	Rukim Villa, कलेक्टर कॉलोनी, Chembur East, Mum...	Mumbai
138714	X-47 B, विक्रोळी व्हिलेज रोड, Godrej Colony, P...	Mumbai
138715	X-48, विक्रोळी व्हिलेज रोड, Godrej Colony, Pir...	Mumbai
138768	परमेश्वर धाम, परमेश्वर धाम opp सन्यास आश्र...	Mumbai

182 rows × 2 columns

```
from google.cloud import translate_v2
translate_client = translate_v2.Client()
def translate_text(text):
    result = translate_client.translate(text, target_language='en')
    return result['input'], result['translatedText']
```

Saved successfully!

```
RMATTED_ADDRESS"].apply(translate_text)
```

Double-click (or enter) to edit

```
# marathi_address["FORMATTED_ADDRESS"]=marathi_address["FORMATTED_ADDRESS"].apply(translate_text)
```

```
def clean_sent(text):
  #tokenization and case conversion
  token=word_tokenize(text.lower())
  #token--->list of tokens
  #removing non alpha char
  ftoken=[i for i in token if i.isalpha()]
  #ftoken-----> list
  sw=stopwords.words("english")
  stokens=[i for i in ftoken if i not in sw]
  #stokens--->list
  #lemmatization
  lemma=WordNetLemmatizer()
  ltoken=[lemma.lemmatize(i) for i in stokens]
  #ltoken--->list
  #joining all tokens
  return " ".join(ltoken)
  text+=1
```

```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
```

Separating target and features

```
x=df10["FORMATTED_ADDRESS"]
y10=df10["LOCALITY"]
```

```
x10=[clean_sent(i) for i in x]
x10
from sklearn.model_selection import train_test_split
xtrain10,xtest10,ytrain10,ytest10=train_test_split(x10,y10,test_size=0.3,random_state=1)
```

Random Forest Classifier (works well on imbalanced data)

```
pipe=Pipeline([

    ("vec",TfidfVectorizer()),
    ("rf",RandomForestClassifier())
])

pipe.fit(xtrain10, ytrain10)
ypred10_rf = pipe.predict(xtest10)
from sklearn.metrics import classification_report
print(classification_report(ytest10, ypred10_rf))
```

```
              precision    recall  f1-score   support

   Ahmedabad       1.00      1.00      1.00      2174
   Bengaluru       1.00      1.00      1.00      5260
     Chennai       1.00      1.00      1.00      2727
       Delhi       1.00      1.00      1.00      2901
   Hyderabad       1.00      1.00      1.00      3526
     Kolkata       1.00      1.00      1.00      2160
      Mumbai       1.00      1.00      1.00     20975
        Pune       1.00      1.00      1.00      1908

    accuracy                           1.00     41631
   macro avg       1.00      1.00      1.00     41631
weighted avg       1.00      1.00      1.00     41631
```

Prediction using Random Forest Classifier

```
def predict(text,model):
  pipe=Pipeline([
```

Saved successfully!

```
  pipe.fit(xtrain10, ytrain10)
  ypred10 = pipe.predict([text])[0]
  return ypred10
```

```
address="A-1/403, wing,Shraddha Saburi Tower, Vitawa ,Thane near mumbai"
adress2="4 th floor jaitu Apt. Pimpri-Chinchwad ,pune"
print(f"Result of address 1:\n{address}\nCity:{predict(address,RandomForestClassifier())}")
print(f"Result of address 2:\n{adress2}\nCity:{predict(adress2,RandomForestClassifier())}")
address3="4 th floor jaitu Apt.Pune wadi Pimpri-Chinchwad ,mumbai metropolitan"
print(f"Result of address 2:\n{address3}\nCity:{predict(address3,RandomForestClassifier())}")
```

```
Result of address 1:
A-1/403, wing,Shraddha Saburi Tower, Vitawa ,Thane near mumbai
City:Mumbai
Result of address 2:
4 th floor jaitu Apt. Pimpri-Chinchwad ,pune
City:Pune
Result of address 2:
4 th floor jaitu Apt.Pune wadi Pimpri-Chinchwad ,mumbai metropolitan
City:Mumbai
```

Naive Bayes Algorithm (as it works on conditional Probability)

```
from sklearn.naive_bayes import MultinomialNB
pipe_nb=Pipeline([
    ("vec",TfidfVectorizer()),
    ("nb",MultinomialNB())
])
pipe_nb.fit(xtrain10,ytrain10)
ypred_nb=pipe_nb.predict(xtest10)
print(classification_report(ytest10, ypred_nb))
```

```
              precision    recall  f1-score   support

   Ahmedabad       1.00      1.00      1.00      2174
   Bengaluru       1.00      1.00      1.00      5260
     Chennai       1.00      1.00      1.00      2727
       Delhi       1.00      0.99      0.99      2901
   Hyderabad       1.00      1.00      1.00      3526
     Kolkata       1.00      1.00      1.00      2160
      Mumbai       0.99      1.00      1.00     20975
        Pune       1.00      0.95      0.97      1908

    accuracy                           1.00     41631
   macro avg       1.00      0.99      0.99     41631
weighted avg       1.00      1.00      1.00     41631
```

```
address="A-1/403, wing,Shraddha Saburi Tower, Vitawa ,Thane near mumbai"
adress2="4 th floor jaitu Apt. Pimpri-Chinchwad ,pune"
print(f"Result of address 1:\n{address}\nCity:{predict(address,MultinomialNB())}")
print(f"Result of address 2:\n{adress2}\nCity:{predict(adress2,MultinomialNB())}")
address3="4 th floor jaitu Apt.Pune wadi Pimpri-Chinchwad ,mumbai metropolitan"
print(f"Result of address 2:\n{address3}\nCity:{predict(address3,MultinomialNB())}")
```

```
Result of address 1:
A-1/403, wing,Shraddha Saburi Tower, Vitawa ,Thane near mumbai
City:Mumbai
Result of address 2:
4 th floor jaitu Apt. Pimpri-Chinchwad ,pune
City:Pune
Result of address 2:
4 th floor jaitu Apt.Pune wadi Pimpri-Chinchwad ,mumbai metropolitan
City:Pune
```

Naive Bayes does not work that accurate but Random Forest does.

To improve accuracy we need our data to be balanced eg Pune has a very less data so the model is not getting learned on Pune which shows that its accuracy of Predicting Pune is low , so we need to have balanced dataset.

Problem 2: Build search engine with autosuggestion

We can build this using the concept of n grams

Saved successfully!

Problem 3:Optimisation for faster execution

```
df1=pd.read_excel("Problem_3_Dataset.xlsx",sheet_name="SOURCE")
df2=pd.read_excel("Problem_3_Dataset.xlsx",sheet_name="DESTINATION")
df1["SOURCE_ID"]=df1.index + 1 #will start from 1
df1.columns=["Source_Latitude",'Source_Longitude',"Source_id"]     #renaming columns
df2["Destination_id"]=df2.index + 1
df2.columns=["Destination_Latitude",'Destination_Longitude',"Destination_id"]
df1["id"]=df1.index +1
df2["id"]=df2.index +1
```

The data is too large and my PC's disk is not supporting . I will work with sample

```
dfnew=df1.iloc[17:23,:]
df=pd.merge(dfnew,df2.iloc[12:42,:],how="cross")
```

There are 6 records in 1st table and 30 in 2nd so total records as per condition must be 6*20=180 combinations

```
df
```

	Source_Latitude	Source_Longitude	Source_id	id_x	Destination_Latitude	Destination_Longitude	Destination_id	id_y

```
df=df[["Source_id",'Source_Latitude', 'Source_Longitude','Destination_id','Destination_Latitude', 'Destination_Longitude']]
df
```

	Source_id	Source_Latitude	Source_Longitude	Destination_id	Destination_Latitude	Destination_Longitude
0	18	12.902804	77.470458	13	28.534694	76.908986
1	18	12.902804	77.470458	14	28.539775	77.052487
2	18	12.902804	77.470458	15	28.540232	76.886004
3	18	12.902804	77.470458	16	28.541810	76.951002
4	18	12.902804	77.470458	17	28.541891	76.886162
...	...	...	...	...	...	...
175	23	13.036399	77.667213	38	28.561498	77.002024
176	23	13.036399	77.667213	39	28.563472	76.932670
177	23	13.036399	77.667213	40	28.566170	76.888658
178	23	13.036399	77.667213	41	28.567451	76.984274
179	23	13.036399	77.667213	42	28.567647	77.049304

180 rows × 6 columns

The haversine library in Python is to calculate the Euclidean distance between two points given their latitude and longitude coordinates

```
# !pip install haversine
from haversine import haversine
```

Saved successfully!

```
titude"],df["Source_Longitude"],df["Destination_Latitude"],df["Destination_Longitude"]

  dist=haversine((lat1,lon1),(lat2,lon2), unit="km")
  return dist

df["DISTANCE_KM"]=df.apply(distance,axis=1)
```

```
df
```

	Source_id	Source_Latitude	Source_Longitude	Destination_id	Destination_Latitude	Destination_Longitude	DISTANCE_KM
0	18	12.902804	77.470458	13	28.534694	76.908986	1739.161200
1	18	12.902804	77.470458	14	28.539775	77.052487	1739.292715
2	18	12.902804	77.470458	15	28.540232	76.886004	1739.857749
3	18	12.902804	77.470458	16	28.541810	76.951002	1739.812002
4	18	12.902804	77.470458	17	28.541891	76.886162	1740.041482
...	...	...	...	...	...	...	...
175	23	13.036399	77.667213	38	28.561498	77.002024	1727.686673
176	23	13.036399	77.667213	39	28.563472	76.932670	1728.206799
177	23	13.036399	77.667213	40	28.566170	76.888658	1728.712753
178	23	13.036399	77.667213	41	28.567451	76.984274	1728.422177
179	23	13.036399	77.667213	42	28.567647	77.049304	1728.181780

180 rows × 7 columns

PROBLEM 4 :TF & IDF value for each word

```
data=pd.read_excel("Problem_1_2_4_Dataset.xlsx")
data
```

	FORMATTED_ADDRESS	LOCALITY
0	Sumangal-project by Heritage group, Samarth Na...	Mumbai
1	#1, Shirke Layout, Kengeri Satellite Town, Ben...	Bengaluru
2	#2, Magadi Main Rd, Mariyappanapalya, Kempapur...	Bengaluru
3	#27/110, Govindapura Main Rd, Govindapura, Nag...	Bengaluru
4	#68, 2 Cross, Sri Venkateshwara Nagar Layout R...	Bengaluru
...	...	...
138764	Zelam, 2-A,, 2-A, Nagri Niwara Cooperative Hou...	Mumbai

```
from sklearn.feature_extraction.text import TfidfVectorizer
import nltk
from nltk.tokenize import word_tokenize
nltk.download("punkt")
from nltk.corpus import stopwords
nltk.download("stopwords")
from nltk.stem import PorterStemmer,WordNetLemmatizer
nltk.download("wordnet")
ps=PorterStemmer()
lemma=WordNetLemmatizer()
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
x=data["FORMATTED_ADDRESS"]
```

Saved successfully!

```
stpw="""a
aadi
aaj
aap
aapne
aata
aati
aaya
aaye
ab
abbe
abbey
abe
abhi
able
about
above
accha
according
accordingly
acha
achcha
across
actually
after
afterwards
again
against
agar
ain
aint
ain't
aisa
aise
aisi
alag
all
allow
allows
almost
alone
along
already
also
although
```

```
always
am
among
amongst
an
and
andar
another
any
anybody
anyhow
anyone
anything
anyway
anyways
anywhere
ap
apan
apart
apna
apnaa
apne
apni
appear
are
aren
arent
aren't
around
arre
as
aside
ask
asking
```

Saved successfully! ×

```
aya
aye
baad
baar
bad
bahut
bana
banae
banai
banao
banaya
banaye
banayi
banda
bande
bandi
bane
bani
bas
bata
batao
bc
be
became
because
become
becomes
becoming
been
before
beforehand
behind
being
below
beside
besides
best
better
between
beyond
bhai
bheetar
bhi
bhitar
bht
```

```
bilkul
bohot
bol
bola
bole
boli
bolo
bolta
bolte
bolti
both
brief
bro
btw
but
by
came
can
cannot
cant
can't
cause
causes
certain
certainly
chahiye
chaiye
chal
chalega
chhaiye
clearly
c'mon
com
come
```

Saved successfully!

```
couldnt
couldn't
d
de
dede
dega
degi
dekh
dekha
dekhe
dekhi
dekho
denge
dhang
di
did
didn
didnt
didn't
dijiye
diya
diyaa
diye
diyo
do
does
doesn
doesnt
doesn't
doing
done
dono
dont
don't
doosra
doosre
down
downwards
dude
dunga
dungi
during
dusra
dusre
dusri
```

```
dvaara
dvara
dwaara
dwara
each
edu
eg
eight
either
ek
else
elsewhere
enough
etc
even
ever
every
everybody
everyone
everything
everywhere
ex
exactly
example
except
far
few
fifth
fir
first
five
followed
following
follows
```

Saved successfully! ×

```
from
further
furthermore
gaya
gaye
gayi
get
gets
getting
ghar
given
gives
go
goes
going
gone
good
got
gotten
greetings
haan
had
hadd
hadn
hadnt
hadn't
hai
hain
hamara
hamare
hamari
hamne
han
happens
har
hardly
has
hasn
hasnt
hasn't
have
haven
havent
haven't
having
```

```
he
hello
help
hence
her
here
hereafter
hereby
herein
here's
hereupon
hers
herself
he's
hi
him
himself
his
hither
hm
hmm
ho
hoga
hoge
hogi
hona
honaa
hone
honge
hongi
honi
hopefully
hota
hotaa
```

Saved successfully! ×

```
howbeit
however
hoyenge
hoyengi
hu
hua
hue
huh
hui
hum
humein
humne
hun
huye
huyi
i
i'd
idk
ie
if
i'll
i'm
imo
in
inasmuch
inc
inhe
inhi
inho
inka
inkaa
inke
inki
inn
inner
inse
insofar
into
inward
is
ise
isi
iska
iskaa
iske
```

```
iski
isme
isn
isne
isnt
isn't
iss
isse
issi
isski
it
it'd
it'll
itna
itne
itni
itno
its
it's
itself
ityaadi
ityadi
i've
ja
jaa
jab
jabh
jaha
jahaan
jahan
jaisa
jaise
jaisi
jata
```

Saved successfully! ×

```
jinhe
jinhi
jinho
jinhone
jinka
jinke
jinki
jinn
jis
jise
jiska
jiske
jiski
jisme
jiss
jisse
jitna
jitne
jitni
jo
just
jyaada
jyada
k
ka
kaafi
kab
kabhi
kafi
kaha
kahaa
kahaan
kahan
kahi
kahin
kahte
kaisa
kaise
kaisi
kal
kam
kar
kara
kare
karega
```

```
karegi
karen
karenge
kari
karke
karna
karne
karni
karo
karta
karte
karti
karu
karun
karunga
karungi
kaun
kaunsa
kayi
kch
ke
keep
keeps
keh
kehte
kept
khud
ki
kin
kine
kinhe
kinho
kinka
kinke
kinhi
```

Saved successfully! ×

```
kino
kis
kise
kisi
kiska
kiske
kiski
kisko
kisliye
kisne
kitna
kitne
kitni
kitno
kiya
kiye
know
known
knows
ko
koi
kon
konsa
koyi
krna
krne
kuch
kuchch
kuchh
kul
kull
kya
kyaa
kyu
kyuki
kyun
kyunki
lagta
lagte
lagti
last
lately
later
le
least
```

```
lekar
lekin
less
lest
let
let's
li
like
liked
likely
little
liya
liye
ll
lo
log
logon
lol
look
looking
looks
ltd
lunga
m
maan
maana
maane
maani
maano
magar
mai
main
maine
mainly
```

Saved successfully! ×

```
mano
many
mat
may
maybe
me
mean
meanwhile
mein
mera
mere
merely
meri
might
mightn
mightnt
mightn't
mil
mjhe
more
moreover
most
mostly
much
mujhe
must
mustn
mustnt
mustn't
my
myself
na
naa
naah
nahi
nahin
nai
name
namely
nd
ne
near
nearly
necessary
neeche
```

```
need
needn
neednt
needn't
needs
neither
never
nevertheless
new
next
nhi
nine
no
nobody
non
none
noone
nope
nor
normally
not
nothing
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
```

Saved successfully! ✕

```
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
par
pata
pe
pehla
pehle
pehli
people
per
perhaps
phla
phle
phli
placed
please
plus
poora
poori
provides
pura
puri
q
que
quite
raha
rahaa
rahe
rahi
rakh
rakha
```

```
rakhe
rakhen
rakhi
rakho
rather
re
really
reasonably
regarding
regardless
regards
rehte
rha
rhaa
rhe
rhi
ri
right
s
sa
saara
saare
saath
sab
sabhi
sabse
sahi
said
sakta
saktaa
sakte
sakti
same
sang
```

```
say
saying
says
se
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
shan
shant
shan't
she
she's
should
shouldn
shouldnt
shouldn't
should've
si
since
six
so
soch
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
```

```
somewhere
soon
still
sub
such
sup
sure
t
tab
tabh
tak
take
taken
tarah
teen
teeno
teesra
teesre
teesri
tell
tends
tera
tere
teri
th
tha
than
thank
thanks
thanx
that
that'll
thats
that's
```

Saved successfully! ×

```
theirs
them
themselves
then
thence
there
thereafter
thereby
therefore
therein
theres
there's
thereupon
these
they
they'd
they'll
they're
they've
thi
thik
thing
think
thinking
third
this
tho
thoda
thodi
thorough
thoroughly
those
though
thought
three
through
throughout
thru
thus
tjhe
to
together
toh
too
took
```

```
toward
towards
tried
tries
true
truly
try
trying
tu
tujhe
tum
tumhara
tumhare
tumhari
tune
twice
two
um
umm
un
under
unhe
unhi
unho
unhone
unka
unkaa
unke
unki
unko
unless
unlikely
unn
unse
until
```

Saved successfully! ×

```
upar
upon
us
use
used
useful
uses
usi
using
uska
uske
usne
uss
usse
ussi
usually
vaala
vaale
vaali
vahaan
vahan
vahi
vahin
vaisa
vaise
vaisi
vala
vale
vali
various
ve
very
via
viz
vo
waala
waale
waali
wagaira
wagairah
wagerah
waha
wahaan
wahan
wahi
```

```
    wahin
    waisa
    waise
    waisi
    wala
    wale
    wali
    want
    wants
    was
    wasn
    wasnt
    wasn't
    way
    we
    we'd
    well
    we'll
    went
    were
    we're
    weren
    werent
    weren't
    we've
    what
    whatever
    what's
    when
    whence
    whenever
    where
    whereafter
    whereas
    [illegible]
```

Saved successfully! ✕

```
    whereupon
    wherever
    whether
    which
    while
    who
    whoever
    whole
    whom
    who's
    whose
    why
    will
    willing
    with
    within
    without
    wo
    woh
    wohi
    won
    wont
    won't
    would
    wouldn
    wouldnt
    wouldn't
    y
    ya
    yadi
    yah
    yaha
    yahaan
    yahan
    yahi
    yahin
    ye
    yeah
    yeh
    yehi
    yes
    yet
    you
    you'd
    you'll
```

```
your
you're
yours
yourself
yourselves
you've
yup"""
```

```
stpw=stpw.split()
```

```
stpw
```

```
     'waha',
     'wahaan',
     'wahan',
     'wahi',
     'wahin',
     'waisa',
     'waise',
     'waisi',
     'wala',
     'wale',
     'wali',
     'want',
     'wants',
     'was',
     'wasn',
     'wasnt',
     "wasn't",
     'way',
     'we',
     "we'd",
     'well',
     "we'll",
     'went',
     'were',
```

Saved successfully!

```
     'werent',
     "weren't",
     "we've",
     'what',
     'whatever',
     "what's",
     'when',
     'whence',
     'whenever',
     'where',
     'whereafter',
     'whereas',
     'whereby',
     'wherein',
     "where's",
     'whereupon',
     'wherever',
     'whether',
     'which',
     'while',
     'who',
     'whoever',
     'whole',
     'whom',
     "who's",
     'whose',
     'why',
     'will',
     'willing',
     'with',
     'within',
     ...]
```

```
def clean_sent(text):
  #tokenization and case conversion
  token=word_tokenize(text.lower())
  #token--->list of tokens
  #removing non alpha char
  ftoken=[i for i in token if i.isalpha()]
  #ftoken-----> list
  sw=stopwords.words("english")

  stokens=[i for i in ftoken if i not in sw or stpw]
  #stokens--->list
  #lemmatization
  lemma=WordNetLemmatizer()
  ltoken=[lemma.lemmatize(i) for i in stokens]
  #ltoken--->list
  #joining all tokens
  return " ".join(ltoken)
  text+=1


x=[clean_sent(i) for i in x]
x
```

```
     'mia mohammad chhotani rd mahim mumbai maharashtra india',
     'kushal nagar sewri mumbai maharashtra india',
     'lokmanya tilak rd gorai borivali west mumbai maharashtra india',
     'kalina kurla rd kolivery village kunchi kurve nagar kalina santacruz east mumbai maharashtra india',
     'kokan nagar rd shyam nagar jogeshwari east mumbai maharashtra india',
     'liberty garden rd number navy colony somwari bazar malad west mumbai maharashtra india',
     'jawaharlal nehru rd padmanabha nagar choolaimedu chennai tamil nadu india',
     'mahadev nagar tekra ahmedabad gujarat india',
     'rr thakur rd gupha tekdi jogeshwari east mumbai maharashtra india',
     'moti nagar mulund colony mulund west mumbai maharashtra india',
     'lion juhu rd indira nagar vile parle west mumbai maharashtra india',
     'jb temkar marg adarsh nagar worli mumbai maharashtra india',
     'mitha ghar rd navghar mulund east mumbai maharashtra india',
     'malpa dongri malpa dongri andheri east mumbai maharashtra india',
     'sahar rd tarun bharat andheri east mumbai maharashtra india',
     'l bhandari marg shimpoli borivali west mumbai maharashtra india',
     'mustafa bazar mazgaon mumbai maharashtra india',
     'lake rd sadan wadi bhandup west mumbai maharashtra india',
     'mantanpada rd mahavir nagar borivali west mumbai maharashtra india',
     'kanjur marg village indira nagar karve nagar kanjurmarg east mumbai maharashtra india',
     'lal bahadur shastri marg cgs colony pant nagar ghatkopar west mumbai maharashtra india',
     'marve rd bmc colony rathodi malad west mumbai maharashtra india',
     'new mandala anushakti nagar mumbai maharashtra india',
     'kalyan society maharashtra society ellisbridge ahmedabad gujarat india',
     'layout rd shri punit nagar borivali west mumbai maharashtra india',
     'mohan gokhale rd umershetpada gokuldham colony goregaon east mumbai maharashtra india',
     'mahakali cave rd shanti nagar andheri east mumbai maharashtra india',
     'meghwadi indira nagar jogeshwari east mumbai maharashtra india',
     'new nagardas rd sai baba wadi natwar nagar jogeshwari east mumbai maharashtra india',
                                 a india',
                                 bai maharashtra india',
     'lal bahadur shastri marg kismat nagar kurla west mumbai maharashtra india',
     'mantanpada road mahavir nagar borivali west mumbai maharashtra india',
     'mg ramachandran marg cheeta camp sector g trombay mumbai maharashtra india',
     'old hanuman ln lohar chawl kalbadevi mumbai maharashtra india',
     'marol maroshi rd bhavani nagar marol andheri east mumbai maharashtra india',
     'marol maroshi rd christian wadi marol village andheri east mumbai maharashtra india',
     'naroda ahmedabad gujarat india',
     'marol maroshi rd bori colony marol village andheri east mumbai maharashtra india',
     'new link rd mhada colony satya nagar borivali west mumbai maharashtra india',
     'premier colony kurla west mumbai maharashtra india',
     'saki vihar rd savarkar nagar chandivali andheri east mumbai maharashtra india',
     'residency rd miyapur hyderabad telangana india',
     'marol maroshi rd vijay nagar midc marol andheri east mumbai maharashtra india',
     'mv shinde marg dina bama estate bhandup west mumbai maharashtra india',
     'new prabhadevi marg kamgar nagar prabhadevi mumbai maharashtra india',
     'masjid st yellagondanpalya victoria layout bengaluru karnataka india',
     'narayan gajanan acharya marg chembur gaothan chembur east mumbai maharashtra india',
     'lokmanya tilak rd hanuman chowk mulund east mumbai maharashtra india',
     'old n nagar munshi nagar andheri west mumbai maharashtra india',
     'prernatirth derasar rd prernatirth part jodhpur ahmedabad gujarat india',
     'sangeetkar n dutta marg gharkul society indira nagar four bungalow andheri west mumbai maharashtra india',
     'rd jaspark society rajeswari society isanpur ahmedabad gujarat india',
     'matunga railway colony matunga mumbai maharashtra india',
     'pune university ganeshkhind pune maharashtra india',
     'lakshami nappu rd matunga railway colony matunga mumbai maharashtra india',
     'mg road rajawadi colony ghatkopar east mumbai maharashtra india',
     ...]
```

Saved successfully! ×

```
vec=TfidfVectorizer()
tf_idf_matrix=vec.fit_transform(x)


feature=vec.get_feature_names_out()
idf=vec.idf_
idf
```

```
    array([10.27562381, 12.14742598, 12.14742598, ..., 11.4542788 ,
           12.14742598, 12.14742598])
```

```
for i,text in enumerate(x):
  tf_idf=tf_idf_matrix[i].toarray()[0]
  d={}
  d["Location"]=data["LOCALITY"]
  locations = list(set(data['LOCALITY']))
  loc_id_map = {loc: i+1 for i, loc in enumerate(locations)}
  data['Location_id'] = data['LOCALITY'].map(loc_id_map)
  for j,k in enumerate(tf_idf):
    if k>0:
      word=feature[j]
      d["word"]=word
      d["term frequency (TF)"]= k
      d["Inverse document frequency(IDF)"] = idf[j]

Freq_data=pd.DataFrame(d)
```

```
Freq_data
```

	Location	word	term frequency (TF)	Inverse document frequency(IDF)
0	Mumbai	valley	0.49481	7.636566
1	Bengaluru	valley	0.49481	7.636566
2	Bengaluru	valley	0.49481	7.636566
3	Bengaluru	valley	0.49481	7.636566
4	Bengaluru	valley	0.49481	7.636566
...	...	...	...	...
138764	Mumbai	valley	0.49481	7.636566
138765	Hyderabad	valley	0.49481	7.636566
138766	Hyderabad	valley	0.49481	7.636566
138767	Pune	valley	0.49481	7.636566
138768	Mumbai	valley	0.49481	7.636566

138769 rows × 4 columns

Saved successfully!

	FORMATTED_ADDRESS	LOCALITY	Location_id
0	Sumangal-project by Heritage group, Samarth Na...	Mumbai	9
1	#1, Shirke Layout, Kengeri Satellite Town, Ben...	Bengaluru	4
2	#2, Magadi Main Rd, Mariyappanapalya, Kempapur...	Bengaluru	4
3	#27/110, Govindapura Main Rd, Govindapura, Nag...	Bengaluru	4
4	#68, 2 Cross, Sri Venkateshwara Nagar Layout R...	Bengaluru	4
...	...	...	...
138764	Zelam, 2-A,, 2-A, Nagri Niwara Cooperative Hou...	Mumbai	9
138765	Zohra Colony, Shalibanda, Hyderabad, Telangana...	Hyderabad	5
138766	ZP Rd, Sriramana Colony, Hastinapuram, Hyderab...	Hyderabad	5
138767	Zuzart's 7, Deccan Paper Mill Rd, Magarpatta C...	Pune	2
138768	परमेश्र्वर धाम, परमेश्र्वर धाम opp सन्यास आश्र...	Mumbai	9

138769 rows × 3 columns

```
Freq_data["Location_id"]=data["Location_id"]
Freq_data
```

```
          Location   word  term frequency   Inverse document  Location_id
                                     (TF)    frequency(IDF)

     1   Bengaluru  valley        0.49481          7.636566            4
     2   Bengaluru  valley        0.49481          7.636566            4
     3   Bengaluru  valley        0.49481          7.636566            4
     4   Bengaluru  valley        0.49481          7.636566            4
   ...         ...     ...            ...               ...          ...
138764      Mumbai  valley        0.49481          7.636566            9
138765   Hyderabad  valley        0.49481          7.636566            5
138766   Hyderabad  valley        0.49481          7.636566            5
```

! 0s completed at 11:47 AM ● ×

Saved successfully! ×