In [1]:

```
# install and import Numpy and pandas dependecy

import numpy as np
import pandas as pd
```

In [2]:

```
# access ".csv" file(this file store on same project folder) and assin in "df" by using pandas

df = pd.read_csv('sms_spam.csv')
```

In [3]:

```
print(df.columns)
```

```
Index(['type', 'text'], dtype='object')
```

In [4]:

```
# try to test open same row and colums from the .csv file

df.sample(10)
```

Out[4]:

	type	text
3646	spam	wamma get laid?want real doggin locations sent...
3555	ham	am up to my eyes in philosophy
5439	ham	Am slow in using biola's fne
1304	ham	I cant pick the phone right now. Pls send a me...
1174	ham	Ü dun need to pick ur gf?
5503	spam	PRIVATE! Your 2003 Account Statement for 07808...
2511	ham	Yunny i'm walking in citylink now ü faster com...
362	ham	Oh ok no prob..
4931	spam	Hi, the SEXYCHAT girls are waiting for you to ...
4792	ham	Send me your resume:-)

In [5]:

```
# Rows and Colums find

df.shape
```

Out[5]:

```
(5574, 2)
```

In [6]:

```
###########==============================================#########
```

output insite the current .csv file [11] the 5574(Rows) and 2(Colums)

1. Data Cleaning

2. EDA (Exploratory data analysis)

3. Text Preprocessing

4. Model building

5. Evalution

6. Improvement

7. Deploy

In [7]:

```
###########===============================================#########
```

1. Data Cleaning

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5574 entries, 0 to 5573
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   type    5574 non-null   object
 1   text    5574 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

In [9]:

```
# rename the cloums

df.rename(columns={'type':'target'},inplace=True)
df.sample(10)
```

Out[9]:

	target	text
3969	ham	Did u turn on the heater? The heater was on an...
988	ham	Geeee ... I miss you already, you know ? Your ...
4362	ham	Don't Think About "What u Have Got" Think Abou...
5091	ham	What type of stuff do you sing?
2643	ham	They can try! They can get lost, in fact. Tee hee
104	ham	wow. You're right! I didn't mean to do that. I...
4033	ham	I'm very happy for you babe ! Woo hoo party on...
3481	ham	What was she looking for?
4484	ham	True lov n care wil nevr go unrecognized. thou...
233	ham	Sorry battery died, yeah I'm here

In [10]:

```
# cloumn target inside row name change name formate like ham->0 and spam-> 1
# for parpose of easy understaning

from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

In [11]:

```
# that error means not install "sklearn.preprocessing"

!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /home/cdac/.local/lib/python3.8/site-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /home/cdac/.local/lib/python3.8/site-packages (from sc
ikit-learn) (1.24.3)
Requirement already satisfied: scipy>=1.3.2 in /home/cdac/.local/lib/python3.8/site-packages (from sci
kit-learn) (1.10.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /home/cdac/.local/lib/python3.8/site-packages
(from scikit-learn) (3.1.0)
Requirement already satisfied: joblib>=1.1.1 in /home/cdac/.local/lib/python3.8/site-packages (from sc
ikit-learn) (1.2.0)
```

In [12]:

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

In [13]:

```
encoder.fit_transform(df['target'])
```

Out[13]:

```
array([0, 0, 1, ..., 0, 0, 0])
```

In [14]:

```
df.sample(10)
```

Out[14]:

	target	text
1873	ham	Oh ok i didnt know what you meant. Yep i am ba...
3838	ham	Early bird! Any purchases yet?
3791	ham	I love you !!! You know? Can you feel it? Does...
1784	ham	No dear i do have free messages without any re...
4881	ham	alright tyler's got a minor crisis and has to ...
5524	ham	Thats cool. I want to please you...
3416	ham	He remains a bro amongst bros
994	ham	The Xmas story is peace.. The Xmas msg is love...
5195	ham	It's wylie, you in tampa or sarasota?
893	ham	Nutter. Cutter. Ctter. Cttergg. Cttargg. Ctarg...

In [15]:

```
# ham-> 0
# spam-> 1

df['target'] = encoder.fit_transform(df['target'])
```

In [16]:

```
df.head()
```

Out[16]:

	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

In [17]:

```
# check "missing" value parsent or not
df.isnull().sum()
```

Out[17]:

```
target    0
text      0
dtype: int64
```

In [18]:

```
# check "duplicate" value parsent or not
df.duplicated().sum()
```

Out[18]:

```
414
```

In [19]:

```
# then "remove" duplicate value
df = df.drop_duplicates(keep='first')
```

In [20]:

```
# Now, recheck "duplicate" value parsent or not
df.duplicated().sum()
```

Out[20]:

```
0
```

In [21]:

```
# Review Rows and Colums parsent now
df.shape
```

Out[21]:

```
(5160, 2)
```

2. EDA (Exploratory data analysis)

In [22]:

```
print(df.columns)
```

```
Index(['target', 'text'], dtype='object')
```

In [23]:

```
# view parsent table

df.head()
```

Out[23]:

	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

In [24]:

```
# filter or count parsent total number of "ham -> 0" and "spam -> 1"

df['target'].value_counts()
```

Out[24]:

```
target
0    4518
1     642
Name: count, dtype: int64
```

In [25]:

```
# that errors means not install "matplotlib"

!pip install matplotlib
```

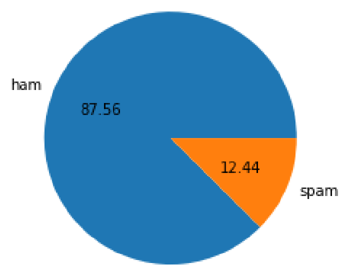
```
Requirement already satisfied: matplotlib in /home/cdac/.local/lib/python3.8/site-packages (3.7.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/cdac/.local/lib/python3.8/site-packages (fro
m matplotlib) (4.39.4)
Requirement already satisfied: contourpy>=1.0.1 in /home/cdac/.local/lib/python3.8/site-packages (from
matplotlib) (1.0.7)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.8/dist-packages (from ma
tplotlib) (2.8.2)
Requirement already satisfied: packaging>=20.0 in /home/cdac/.local/lib/python3.8/site-packages (from
matplotlib) (23.1)
Requirement already satisfied: pyparsing>=2.3.1 in /home/cdac/.local/lib/python3.8/site-packages (from
matplotlib) (3.0.9)
Requirement already satisfied: importlib-resources>=3.2.0; python_version < "3.10" in /home/cdac/.loca
l/lib/python3.8/site-packages (from matplotlib) (5.12.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/lib/python3/dist-packages (from matplotlib) (7.0.
0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/cdac/.local/lib/python3.8/site-packages (fro
m matplotlib) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /home/cdac/.local/lib/python3.8/site-packages (from mat
plotlib) (0.11.0)
Requirement already satisfied: numpy>=1.20 in /home/cdac/.local/lib/python3.8/site-packages (from matp
lotlib) (1.24.3)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.7->
matplotlib) (1.14.0)
Requirement already satisfied: zipp>=3.1.0; python_version < "3.10" in /home/cdac/.local/lib/python3.
8/site-packages (from importlib-resources>=3.2.0; python_version < "3.10"->matplotlib) (3.15.0)
```

In [26]:

```
# then that data show on "Pie Chart format"

import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
```

Out[26]:

```
([<matplotlib.patches.Wedge at 0x7fdc7fe4d0a0>,
  <matplotlib.patches.Wedge at 0x7fdc7fe37f70>],
 [Text(-1.0170346463201791, 0.4190948916228736, 'ham'),
  Text(1.0170346267009303, -0.4190949392337011, 'spam')],
 [Text(-0.5547461707200977, 0.22859721361247648, '87.56'),
  Text(0.5547461600186891, -0.22859723958201877, '12.44')])
```

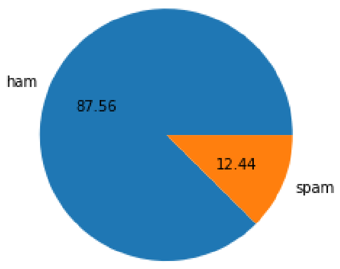


In [27]:

```
# remove extra code top of the "Pic Chart"
# by the using command this
# "plt.show()"
```

In [28]:

```
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
```



In [29]:

```
print(df.columns)
```

```
Index(['target', 'text'], dtype='object')
```

2.1 Data is imbalance So, Blance it

In [30]:

```
# get information from this Pic chart
#I see data "ham" and "spam" are not blanced
# sms ke ander kitne "No. of Alphabet, No. of Words, No of Santance" etc.
# use ho raha iska filtter karege
# iske liye "Three Cloumns" create karege

# so, i'm using "NLTK" Library
```

In [31]:

```
print(df.columns)
```

```
Index(['target', 'text'], dtype='object')
```

In [32]:

```
 import nltk
```

In [33]:

```
# that errors means not install "nltk"

!pip install nltk
```

```
Requirement already satisfied: nltk in /home/cdac/.local/lib/python3.8/site-packages (3.8.1)
Requirement already satisfied: regex>=2021.8.3 in /home/cdac/.local/lib/python3.8/site-packages (from
nltk) (2023.5.5)
Requirement already satisfied: joblib in /home/cdac/.local/lib/python3.8/site-packages (from nltk) (1.
2.0)
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from nltk) (7.0)
Requirement already satisfied: tqdm in /home/cdac/.local/lib/python3.8/site-packages (from nltk) (4.6
5.0)
```

In [34]:

```
# then after import nltk

import nltk
```

In [35]:

```
# then same importent "Dependency of NLTK download" so,

nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /home/cdac/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[35]:

```
True
```

In [36]:

```
# "text" cloumn ke ander "character" ka lenthg find out then...
# har massage ka text charactor length count kar de raha hai

df['text'].apply(len)
```

Out[36]:

```
0       111
1        29
2       155
3        49
4        61
       ... 
5569    160
5570     36
5571     57
5572    125
5573     26
Name: text, Length: 5160, dtype: int64
```

In [37]:

```
# ab "num_characters" cloums nam ke ander store kar dete hai "text length ko"
# create new cloumn of "num_characters"

df['num_characters'] = df['text'].apply(len)
```

In [38]:

```
# ab check karte hai parsent data table ko

df.head()
```

Out[38]:

	target	text	num_characters
0	0	Go until jurong point, crazy.. Available only ...	111
1	0	Ok lar... Joking wif u oni...	29
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	0	U dun say so early hor... U c already then say...	49
4	0	Nah I don't think he goes to usf, he lives aro...	61

In [39]:

```
# "num of words count" karte hai ki "text" ke "row" ke santance me kitne words hai
#iske liye lambda santance run karega or NLTK library ke "word_tokennize" word count karega

df['text'].apply(lambda x:nltk.word_tokenize(x))
```

Out[39]:

```
0       [Go, until, jurong, point, ,, crazy, .., Avail...
1                [Ok, lar, ..., Joking, wif, u, oni, ...]
2       [Free, entry, in, 2, a, wkly, comp, to, win, F...
3       [U, dun, say, so, early, hor, ..., U, c, alrea...
4       [Nah, I, do, n't, think, he, goes, to, usf, ,,...
                              ...                        
5569    [This, is, the, 2nd, time, we, have, tried, 2,...
5570      [Will, ü, b, going, to, esplanade, fr, home, ?]
5571    [Pity, ,, *, was, in, mood, for, that, ., So, ...
5572    [The, guy, did, some, bitching, but, I, acted,...
5573                  [Rofl, ., Its, true, to, its, name]
Name: text, Length: 5160, dtype: object
```

In [40]:

```
# "text" ke sabhi santance "words" me divide ho kar "Array list store" ho gaya
# ab Array me store words ka "length" count kar lenge
# So, use "len()"

df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
```

Out[40]:

```
0       24
1        8
2       37
3       13
4       15
        ..
5569    35
5570     9
5571    15
5572    27
5573     7
Name: text, Length: 5160, dtype: int64
```

In [41]:

```
# ab "num_words" cloums nam ke ander store kar dete hai "inside Arrry words length ko"
# create new cloumn of "num_words"

df ['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
```

In [42]:

```
# ab check karte hai parsent data table ko

df.head()
```

Out[42]:

	target	text	num_characters	num_words
0	0	Go until jurong point, crazy.. Available only ...	111	24
1	0	Ok lar... Joking wif u oni...	29	8
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37
3	0	U dun say so early hor... U c already then say...	49	13
4	0	Nah I don't think he goes to usf, he lives aro...	61	15

In [43]:

```
# "No. of Santance count" "text" ke ak "row" me kitne santance hai
#iske liye lambda santance run karega or NLTK library ke "sent_tokenize" word count karega

df['text'].apply(lambda x:nltk.sent_tokenize(x))
```

Out[43]:

```
0       [Go until jurong point, crazy.., Available onl...
1                         [Ok lar..., Joking wif u oni...]
2       [Free entry in 2 a wkly comp to win FA Cup fin...
3       [U dun say so early hor... U c already then sa...
4       [Nah I don't think he goes to usf, he lives ar...
                              ...
5569    [This is the 2nd time we have tried 2 contact ...
5570              [Will ü b going to esplanade fr home?]
5571    [Pity, * was in mood for that., So...any other...
5572    [The guy did some bitching but I acted like i'...
5573                         [Rofl., Its true to its name]
Name: text, Length: 5160, dtype: object
```

In [44]:

```
# "text" ke sabhi rows me "santance" me divide ho kar "Array list store" ho gaya
# ab Array me store santance ka "length" count kar lenge
# So, use "len()"

df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

Out[44]:

```
0       2
1       2
2       2
3       1
4       1
       ..
5569    4
5570    1
5571    2
5572    1
5573    2
Name: text, Length: 5160, dtype: int64
```

In [45]:

```
# ab "num_sentences" cloums nam ke ander store kar dete hai "inside Arrry sentance length ko"
# create new cloumn of "num_sentences"

df ['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

In [46]:

```
# ab check karte hai parsent data table ko

df.head()
```

Out[46]:

	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

In [47]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [48]:

```
# ab check karte hai ki pure table ka data analysis karte hai...
# like maximum or minmum length, total %, etc...
# So, use this funtion ".describe()"

df[['num_characters','num_words','num_sentences']].describe()
```

Out[48]:

	num_characters	num_words	num_sentences
count	5160.000000	5160.000000	5160.000000
mean	79.141085	18.588178	1.970543
std	58.289153	13.396252	1.455918
min	2.000000	1.000000	1.000000
25%	36.000000	9.000000	1.000000
50%	61.000000	15.000000	1.000000
75%	118.000000	26.000000	2.000000
max	910.000000	220.000000	38.000000

In [49]:

```
# yaha jese ki num_characters ke column ke ander
# maximum No of character use 910.000.. (describe both data ham and spam)
# so seprate data analysis for ham and spam
```

In [50]:

```
# for "ham ->0 message" data analysis

df[df['target'] == 0][['num_characters','num_words','num_sentences']].describe()
```

Out[50]:

	num_characters	num_words	num_sentences
count	4518.000000	4518.000000	4518.000000
mean	70.860558	17.289951	1.827579
std	56.584422	13.579652	1.394245
min	2.000000	1.000000	1.000000
25%	34.000000	8.000000	1.000000
50%	53.000000	13.000000	1.000000
75%	91.000000	22.000000	2.000000
max	910.000000	220.000000	38.000000

In [51]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [52]:

```
# for "spam ->1 message" data analysis

df[df['target'] == 1][['num_characters','num_words','num_sentences']].describe()
```

Out[52]:

	num_characters	num_words	num_sentences
count	642.000000	642.000000	642.000000
mean	137.414330	27.724299	2.976636
std	29.975596	7.028380	1.484527
min	13.000000	2.000000	1.000000
25%	131.000000	25.000000	2.000000
50%	148.000000	29.000000	3.000000
75%	157.000000	32.000000	4.000000
max	223.000000	46.000000	9.000000

In [53]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [ ]:

In [ ]:

In [54]:

```
# that errors means not install "seaborn"

!pip install seaborn
```

```
Requirement already satisfied: seaborn in /home/cdac/.local/lib/python3.8/site-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /home/cdac/.local/lib/python3.8/site-packages
(from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.8/dist-packages (from seaborn)
(2.0.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /home/cdac/.local/lib/python3.8/site-package
s (from seaborn) (3.7.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.8/dist-packages (from pandas>=0.
25->seaborn) (2023.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.8/dist-packages (from
pandas>=0.25->seaborn) (2.8.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.8/dist-packages (from pandas>=
0.25->seaborn) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in /home/cdac/.local/lib/python3.8/site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /home/cdac/.local/lib/python3.8/site-packages (from mat
plotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: packaging>=20.0 in /home/cdac/.local/lib/python3.8/site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: importlib-resources>=3.2.0; python_version < "3.10" in /home/cdac/.loca
l/lib/python3.8/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (5.12.0)
Requirement already satisfied: fonttools>=4.22.0 in /home/cdac/.local/lib/python3.8/site-packages (fro
m matplotlib!=3.6.1,>=3.1->seaborn) (4.39.4)
Requirement already satisfied: pyparsing>=2.3.1 in /home/cdac/.local/lib/python3.8/site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/cdac/.local/lib/python3.8/site-packages (fro
m matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in /usr/lib/python3/dist-packages (from matplotlib!=3.6.
1,>=3.1->seaborn) (7.0.0)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.2
->pandas>=0.25->seaborn) (1.14.0)
Requirement already satisfied: zipp>=3.1.0; python_version < "3.10" in /home/cdac/.local/lib/python3.
8/site-packages (from importlib-resources>=3.2.0; python_version < "3.10"->matplotlib!=3.6.1,>=3.1->se
aborn) (3.15.0)
```

In [55]:

```
# run/import again

import seaborn as sns
```

In [56]:

```
# traget column ke num_characters row me kitne charactor used ho rahe hai

df[df['target'] == 0]['num_characters']
```

Out[56]:

```
0       111
1        29
3        49
4        61
6        77
       ... 
5567     12
5570     36
5571     57
5572    125
5573     26
Name: num_characters, Length: 4518, dtype: int64
```

In [57]:

```
print(df.columns)
```

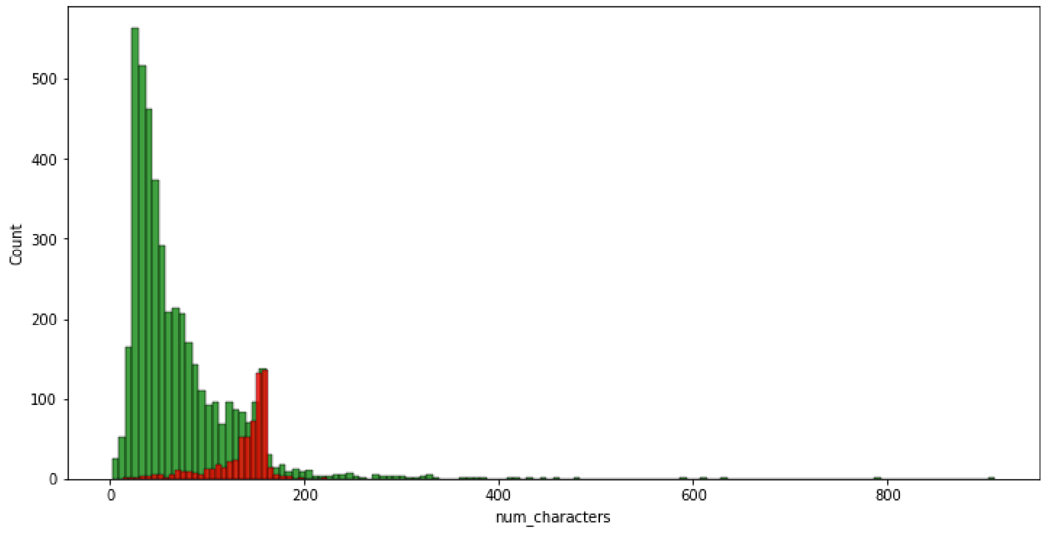
```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [58]:

```
# library seaborn ka "histplot" function used kar show karte hai
# ham-> 0 message ko color "green"
# spam-> 1 message ko color "red"
# or figure ka size bada kar dekhte hai

plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_characters'],color='green')
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

Out[58]:

```
<Axes: xlabel='num_characters', ylabel='Count'>
```



In [59]:

```
# owhi ab "num_words" or "num_santences" ke sath check karte hai
```

In [60]:

```
print(df.columns)
```

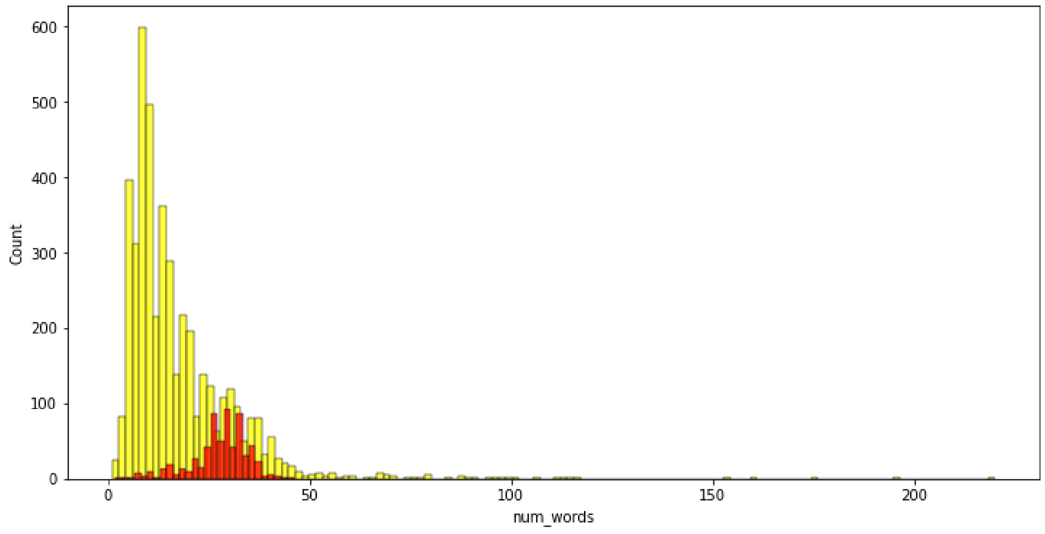
```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [61]:

```
# num_words
# ham-> 0 message ko color "yellow"
# spam-> 1 message ko color "red"

plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_words'],color='yellow')
sns.histplot(df[df['target'] == 1]['num_words'],color='red')
```

Out[61]:

```
<Axes: xlabel='num_words', ylabel='Count'>
```



In [62]:

```
print(df.columns)
```

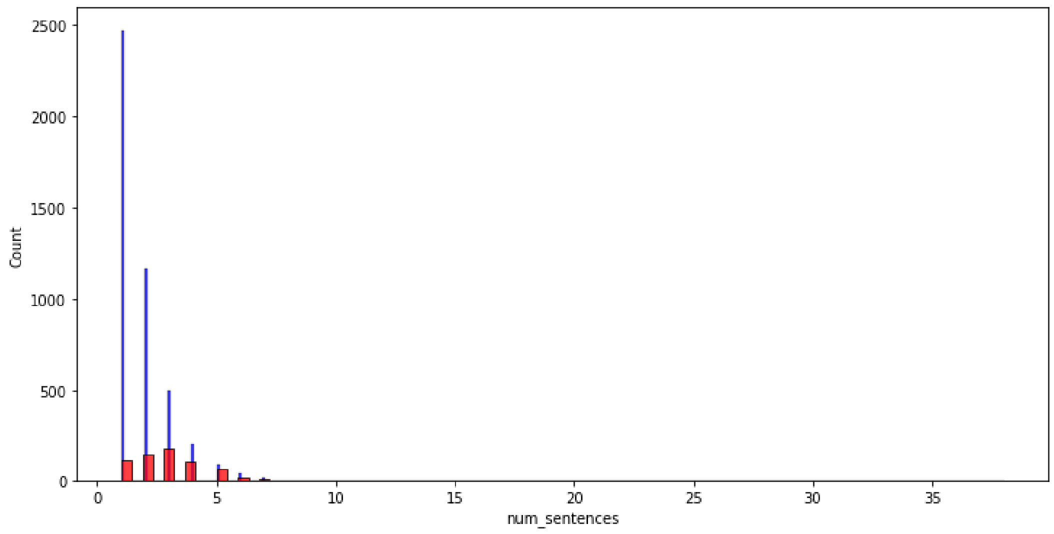
```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [63]:

```
# num_sentences
# ham-> 0 message ko color "blue"
# spam-> 1 message ko color "red"

plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_sentences'],color='blue')
sns.histplot(df[df['target'] == 1]['num_sentences'],color='red')
```

Out[63]:

```
<Axes: xlabel='num_sentences', ylabel='Count'>
```



In [64]:

```
print(df.columns)
```

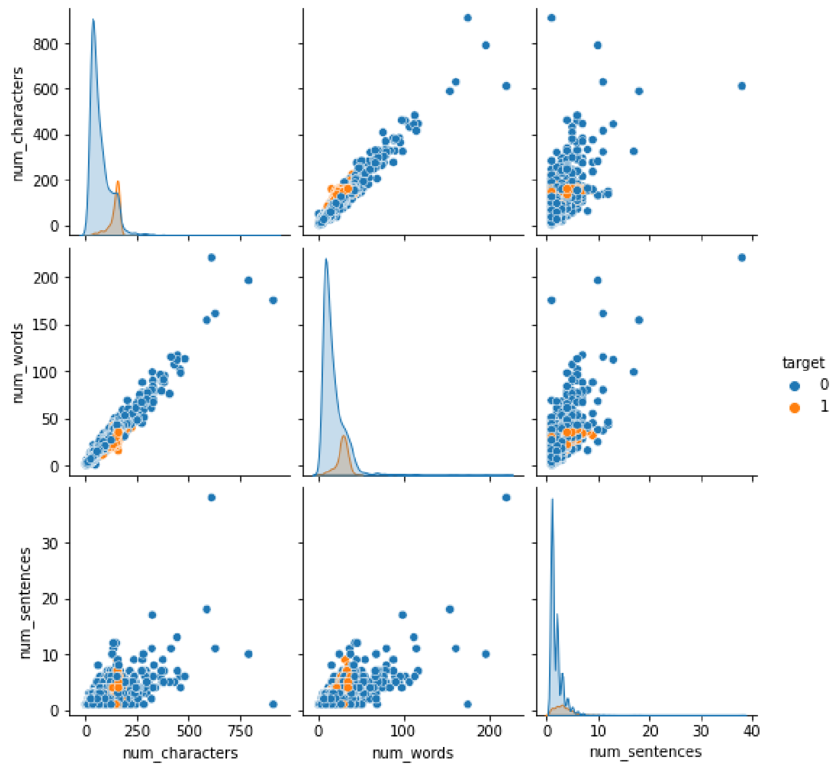
```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [65]:

```
#ab check karte hai ki "ham or spam ke message" ka tino
#Three (charactor, words, or sentance) apas me kaya relation hai

sns.pairplot(df,hue='target')
```

Out[65]:

```
<seaborn.axisgrid.PairGrid at 0x7fdc7a68cfa0>
```



In [66]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [67]:

```
# ab sabhi ko "heatmap" me chaeck karte hai apas me relation hai

sns.heatmap(df.corr(),annot=True)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-67-bfded30e3083> in <module>
      1 # ab sabhi ko "heatmap" me chaeck karte hai apas me relation hai
      2 
----> 3 sns.heatmap(df.corr(),annot=True)

/usr/local/lib/python3.8/dist-packages/pandas/core/frame.py in corr(self, method, min_periods, numeric
_only)
  10057         cols = data.columns
  10058         idx = cols.copy()
> 10059         mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  10060 
  10061         if method == "pearson":

/usr/local/lib/python3.8/dist-packages/pandas/core/frame.py in to_numpy(self, dtype, copy, na_value)
   1836         if dtype is not None:
   1837             dtype = np.dtype(dtype)
-> 1838         result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1839         if result.dtype is not dtype:
   1840             result = np.array(result, dtype=dtype, copy=False)

/usr/local/lib/python3.8/dist-packages/pandas/core/internals/managers.py in as_array(self, dtype, cop
y, na_value)
   1730                 arr.flags.writeable = False
   1731         else:
-> 1732             arr = self._interleave(dtype=dtype, na_value=na_value)
   1733             # The underlying data was copied within _interleave, so no need
   1734             # to further copy if copy=True or setting na_value

/usr/local/lib/python3.8/dist-packages/pandas/core/internals/managers.py in _interleave(self, dtype, n
a_value)
   1792             else:
   1793                 arr = blk.get_values(dtype)
-> 1794             result[rl.indexer] = arr
   1795             itemmask[rl.indexer] = 1
   1796 

ValueError: could not convert string to float: 'Go until jurong point, crazy.. Available only in bugis
n great world la e buffet... Cine there got amore wat...'
```

In [68]:

```
# this errors means "string se float" me convert "nahi" kar pa raha hai
# to "matplotlib.pyplot" Library import karte hai
```

In [69]:

```
import matplotlib.pyplot as plt
```

In [70]:

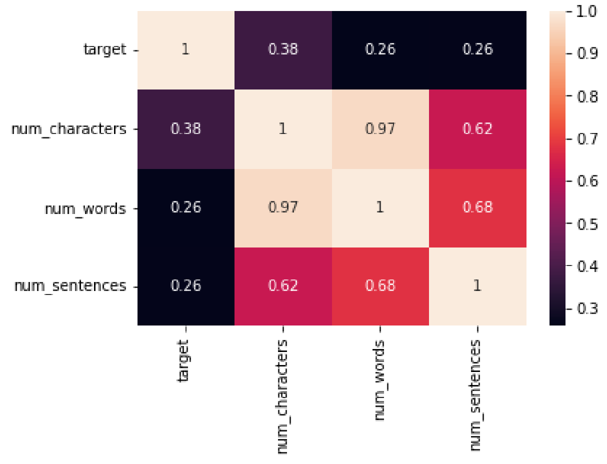
```
#=================WAR
#isse dik se run kare kunki kabhi kabhi isee run karne par "text" column remove ho jata hai

df = df.select_dtypes(include=[float, int])
```

In [71]:

```
sns.heatmap(df.corr(), annot=True)
```

Out[71]:

```
<Axes: >
```



In [70]:

```
#yaha appas me relation

# num_character <-> num_character => 1
# num_character <-> num_words => 0.97
# num_character <-> num_sentance => 0.62

# num_words <-> num_character => 0.97
# num_words <-> num_words => 1
# soon on.....

#strong relation
# num_character <-> num_character => 1
# num_words <-> num_words => 1
# num_sentance <-> num_sentance => 1

# "Model banane ke liye kisi ak ko lege jese "num_character" ko"
# ham tino (three) ko nahi lege kunki jayada strong ho jayega
```

3. Data Preprocessing

3.1 Lower case

3.2 Tokenization

3.3 Removing special characters

3.4 Removing stop words and punctuation

3.5 Stemming

In [71]:

```
# sabse pehale ham ye sabhi ka ak-ak "example" ke rup me dekh lete hai
# phir apne Project par apply karge
```

In [72]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [ ]:

```

```

In [73]:

```
#3.3 Example of ""Remove Special character"in the "sentence words""
# iske liye loop bana kar "isalnum()" call karte hai,
# "isalnum()" ye Alphabetic and Number ko select karega
# ".append()" ye yaha "y" me assin kar dega value ko

# def transform_text(text2):
#     text2 = text2.lower()
#     text2 = nltk.word_tokenize(text2)

#     y = []
#     for i in text2:
#         if i.isalnum():
#             y.append(i)
#     return y

# transform_text('Hi how Are You? e.g 20%')
```

In [ ]:

```

```

In [74]:

```
# 3.4.1 Example "StopWords"
# StopWords => yese "words" so sentence ke "meaning" me koi contribution "nahi" hota
# kewal iska kam sentance "formation" hota hai
# e.g..
```

In [75]:

```
# iske liye "NLTK Library" se "stopwords" find out karege
import nltk
```

In [76]:

```
# download karte hai "stopwords"

nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /home/cdac/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[76]:

```
True
```

In [77]:

```
# ab filter karte hai "stopwords" se "english words" ka list out karte hai

stopwords.words('english')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-77-b93a77273381> in <module>
      1 # ab filter karte hai "stopwords" se "english words" ka list out karte hai
      2
----> 3 stopwords.words('english')

NameError: name 'stopwords' is not defined
```

In [78]:

```
# ab aage future Direct use kar sakte hai iss command ke through

from nltk.corpus import stopwords
stopwords.words('english')
```

Out[78]:

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
```

In [79]:

```
# 3.4.1 Example "Punctuation" list sort list karte hai
# iske liye "import string library"

import string
string.punctuation
```

Out[79]:

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [80]:

```
# 3.5 Example "Stemming"
# ye "varb words" ko "original word" me la deta hai
# e.g.. Loving -> Love, Dancing -> Dance, Played -> Play etc..
# iske liye NLTK ka PorterStemmer module import karna hoga

from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('loving')
```

Out[80]:

```
'love'
```

In [81]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [82]:

```
# ab isse uppr wale transform_text par apply karte hai
# yaha "text = y[:]" ye "cloning" hai jo "y" ke value ko text me store kar raha hai

def transform_text(text):
    #Ex.3.1
    text = text.lower()

    #Ex.3.2
    text = nltk.word_tokenize(text)

    #Ex.3.3
    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    #Ex.3.4
    text = y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)
    #Ex.3.5
    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))

    return " ".join(y)

transform_text('I loved the CDAC leactures on Machine Learning. How about you? ')
```

Out[82]:

```
'love cdac leactur machin learn'
```

In [83]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences'], dtype='object')
```

In [84]:

```
df.head()
```

Out[84]:

	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

In [85]:

```
df['text'].apply(transform_text)
```

Out[85]:

```
0       go jurong point crazi avail bugi n great world...
1                                   ok lar joke wif u oni
2       free entri 2 wkli comp win fa cup final tkt 21...
3                     u dun say earli hor u c alreadi say
4                    nah think goe usf live around though
                              ...
5569    2nd time tri 2 contact u pound prize 2 claim e...
5570                              ü b go esplanad fr home
5571                                    piti mood suggest
5572    guy bitch act like interest buy someth els nex...
5573                                       rofl true name
Name: text, Length: 5160, dtype: object
```

In [86]:

```
# mere liye only "traget" and "tranformed_text" cloumns imported hai

df['transformed_text'] = df['text'].apply(transform_text)
```

In [87]:

```
df.head()
```

Out[87]:

	target	text	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

In [88]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences',
       'transformed_text'],
      dtype='object')
```

In [89]:

```
# ab "ham->0 spam->1" ye demo message "kya-kya words used" huwa hai
#usee a "Image me dekhe" ge, jo jyda use hoga owh sabse bada dikhega
# iske liye hame "WordCloud Library" ka use karna hoga
```

In [90]:

```
# from wordcloud import WordCloud
# wc = WordCloud(width=500,height=500,min_font_size=10,backgroud_color='white')
# spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].str.cat(sep=" "))
# plt.imshow(spam_wc)
```

In [ ]:

```

```

In [91]:

```
# ab most "top 30",50 etc. "common used" words in the "ham and spam mesaage"
#ko sortlist karte hai
```

In [92]:

```
#hame only "target" and "transformed_text" cloumns ke data par parfom karege
#sabse pehale spam->1 message ko table se alag karte hai

df[df['target'] == 1]
```

Out[92]:

	target	text	num_characters	num_words	num_sentences	transformed_text
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
5	1	FreeMsg Hey there darling it's been 3 week's n...	147	39	4	freemsg hey darl 3 week word back like fun sti...
8	1	WINNER!! As a valued network customer you have...	157	32	5	winner valu network custom select receivea pri...
9	1	Had your mobile 11 months or more? U R entitle...	154	31	3	mobil 11 month u r entitl updat latest colour ...
11	1	SIX chances to win CASH! From 100 to 20,000 po...	136	31	3	six chanc win cash 100 pound txt csh11 send co...
...	...	...	...	...	...	...
5539	1	Want explicit SEX in 30 secs? Ring 02073162414...	90	18	3	want explicit sex 30 sec ring 02073162414 cost...
5542	1	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...	158	38	6	ask 3mobil 0870 chatlin inclu free min india c...
5549	1	Had your contract mobile 11 Mnths? Latest Moto...	160	35	5	contract mobil 11 mnth latest motorola nokia e...
5568	1	REMINDER FROM O2: To get 2.50 pounds free call...	147	30	1	remind o2 get pound free call credit detail gr...
5569	1	This is the 2nd time we have tried 2 contact u...	160	35	4	2nd time tri 2 contact u pound prize 2 claim e...

642 rows × 6 columns

In [93]:

```
# ab isme se "transformed_text" me used words ko ak "List" me dal dete hai
# yaha har message ak "item hai"

df[df['target'] == 1]['transformed_text'].tolist()
```

Out[93]:

```
['free entri 2 wkli comp win fa cup final tkt 21st may text fa 87121 receiv entri question std txt r
ate c appli 08452810075over18',
 'freemsg hey darl 3 week word back like fun still tb ok xxx std chg send rcv',
 'winner valu network custom select receivea prize reward claim call claim code kl341 valid 12 hou
r',
 'mobil 11 month u r entitl updat latest colour mobil camera free call mobil updat co free 080029860
30',
 'six chanc win cash 100 pound txt csh11 send cost 6day tsandc appli repli hl 4 info',
 'urgent 1 week free membership prize jackpot txt word claim 81010 c lccltd pobox 4403ldnw1a7rw18',
 'xxxmobilemovieclub use credit click wap link next txt messag click http',
 'england v macedonia dont miss news txt ur nation team 87077 eg england 87077 tri wale scotland pob
oxox36504w45wq',
 'thank subscript rington uk mobil charg pleas confirm repli ye repli charg',
 '07732584351 rodger burn msg tri call repli sm free nokia mobil free camcord pleas call 08000930705
deliveri tomorrow',
 'sm ac sptv new jersey devil detroit red wing play ice hockey correct incorrect end repli end spt
v',
 'congrat 1 year special cinema pass 2 call 09061209465 c suprman v matrix3 starwars3 etc 4 free 150
```

In [94]:

```
# ab sabhi message ko ak-ak kar "print" karte hai by the help of "for loop"

for msg in df[df['target'] == 1]['transformed_text'].tolist():
    print(msg)
```

```
free entri 2 wkli comp win fa cup final tkt 21st may text fa 87121 receiv entri question std txt rat
e c appli 08452810075over18
freemsg hey darl 3 week word back like fun still tb ok xxx std chg send rcv
winner valu network custom select receivea prize reward claim call claim code kl341 valid 12 hour
mobil 11 month u r entitl updat latest colour mobil camera free call mobil updat co free 08002986030
six chanc win cash 100 pound txt csh11 send cost 6day tsandc appli repli hl 4 info
urgent 1 week free membership prize jackpot txt word claim 81010 c lccltd pobox 4403ldnw1a7rw18
xxxmobilemovieclub use credit click wap link next txt messag click http
england v macedonia dont miss news txt ur nation team 87077 eg england 87077 tri wale scotland pobox
ox36504w45wq
thank subscript rington uk mobil charg pleas confirm repli ye repli charg
07732584351 rodger burn msg tri call repli sm free nokia mobil free camcord pleas call 08000930705 d
eliveri tomorrow
sm ac sptv new jersey devil detroit red wing play ice hockey correct incorrect end repli end sptv
congrat 1 year special cinema pass 2 call 09061209465 c suprman v matrix3 starwars3 etc 4 free 150pm
dont miss
valu custom pleas advis follow recent review mob award bonu prize call 09066364589
urgent ur award complimentari trip eurodisinc trav aco entry41 claim txt di 87121 morefrmmob shracom
orsglsuplt 10 ls1 3aj
hear new divorc barbi come ken stuff
```

In [95]:

```
# ab isse "msg" se sabhi "words" ko ak-ak "alag" kar "list" me append(assin) kar dete hai

spam_corpus = []
for msg in df[df['target'] == 1]['transformed_text'].tolist():
    for word in msg.split():
        spam_corpus.append(word)
```

In [96]:

```
# check(view) list

spam_corpus
```

Out[96]:

```
['free',
 'entri',
 '2',
 'wkli',
 'comp',
 'win',
 'fa',
 'cup',
 'final',
 'tkt',
 '21st',
 'may',
 'text',
 'fa',
 '87121',
 'receiv',
 'entri',
 'question',
```

In [97]:

```
# count list length(spam_corpus)
# ki kitene words hai isse list me

len(spam_corpus)
```

Out[97]:

```
9808
```

In [98]:

```
# ab check karte hai ki isse list used words ka information nikalte hai
#like kitini par used huwa hai,"most_common" word, "least_common" word used....
# yaha "most_common" used nikal rahe hai "Top 30" words me se
# iske liye "Collections Library" ka used karte hai

from collections import Counter
Counter(spam_corpus).most_common(30)
```

Out[98]:

```
[('call', 313),
 ('free', 186),
 ('2', 154),
 ('txt', 139),
 ('text', 122),
 ('ur', 119),
 ('u', 118),
 ('mobil', 110),
 ('stop', 108),
 ('repli', 103),
 ('claim', 97),
 ('4', 95),
 ('prize', 79),
 ('get', 73),
 ('new', 64),
 ('servic', 64),
 ('send', 60),
 ('tone', 59),
 ('urgent', 57),
 ('award', 55),
 ('nokia', 54),
 ('contact', 54),
 ('phone', 52),
 ('cash', 50),
 ('pleas', 50),
 ('week', 49),
 ('win', 46),
 ('min', 45),
 ('c', 43),
 ('guarante', 42)]
```

In [99]:

```
# ab ak DataFrame me add kar dete hai sabhi ko

from collections import Counter
pd.DataFrame(Counter(spam_corpus).most_common(30))
```

Out[99]:

	0	1
0	call	313
1	free	186
2	2	154
3	txt	139
4	text	122
5	ur	119
6	u	118
7	mobil	110
8	stop	108
9	repli	103
10	claim	97
11	4	95
12	prize	79
13	get	73
14	new	64
15	servic	64
16	send	60
17	tone	59
18	urgent	57
19	award	55
20	nokia	54
21	contact	54
22	phone	52
23	cash	50
24	pleas	50
25	week	49
26	win	46
27	min	45
28	c	43
29	guarante	42

In [100]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences',
       'transformed_text'],
      dtype='object')
```

In [101]:

```
# ab isse "DataFrame" ko ak "Bar Chart" me "show" karte hai

# from collections import Counter
# sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam_corpus).most_common(3
# plt.xticks(rotation='vertical')
# plt.show()
```

4 Model Building

In [102]:

```
#ab hamare liye two cloumns importent hai
# 1st "target" (column jo ki 0-> ham, 1-> spam) ye hame liye "Output" ka kam karega
# 2nd "tranformed_text" (column jo sare filter karne ke bad mila hai) ye hamre liye "Input" ka kam karega

# lekin, dono cloumns ka "data(Row ka)" hame "interger(0 ya 1)" chahiye jo ki "vector" hoga
# lekin,, yaha hamare pass 'target' column ka data "interger" hai. But "transform_text" cloumn ka data "string" me h
# to isse("tranformed_text" ka sabhi text ko) "integer"(yani vector) bana hoga..

# iske liye "CountVectorizer Library" ka used karege
```

In [103]:

```
# iske liye "CountVectorizer Library" ka used karege

from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
```

In [104]:

```
# "tranform_text" columns ke data(text)) ko interger(0 ya 1)convert kar dete hai
#or usse array ke rup me "X" me assin(store) kar dete hai

X = cv.fit_transform(df['transformed_text']).toarray()
```

In [105]:

```
# yaha X hame mil gaya or X me sabhi 0 ke rup me assin(store) hoga jo ki sahi hai
X
```

Out[105]:

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

In [106]:

```
X.shape
```

Out[106]:

```
(5160, 6784)
```

In [107]:

```
# yaha sms => 5160 and word => 6784

print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences',
       'transformed_text'],
      dtype='object')
```

In [108]:

```
# ab hame Y bhi nikalna hoga to..

y = df['target'].values
```

In [109]:

```
y
```

Out[109]:

```
array([0, 0, 1, ..., 0, 0, 0])
```

4.1 Model Building (apply diff. Algo.)

check for best Accuracy

then select one Algo. after build Model

In [110]:

```
# iske liye ka "train_test_split" ko import karna hoga model select ke liye
#"sklearn.model_selection" ka "train_test_split" used kar rahe hai

from sklearn.model_selection import train_test_split
```

In [111]:

```
#abhi to only 20% data par hi apply karte hai

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [112]:

```
# abhi ham sklearn ka "naive_bayes" Library ka used kar rahe hai,
#jisme "naive_bayes" ka "GaussianNB,MultinomialNB,BernoulliNB" algorithom ka used
# "sklearn.metrics" ka "accuracy_score,confusion_matrix,precision_score"

from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [113]:

```
# sabhi algorithm ko ak "Object" me assin(store) kiya

gnb = GaussianNB()
mnd = MultinomialNB()
bnb = BernoulliNB()
```

In [114]:

```
# ab ak-ak kar sabhi algo. ko perform karte hai
#or chack karte hai kise hame best accuracy,confusion,precision milega
#jese yaha "gnb = GaussianNB()" apply kate hai to

gnb.fit(X_train,y_train)
y_pred1 = gnb.predict(X_test)

print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
```

```
0.875
[[803 113]
 [ 16 100]]
0.4694835680751174
```

In [115]:

```
#jese yaha "mnd = MultinomialNB()" apply kate hai to

mnd.fit(X_train,y_train)
y_pred2 = mnd.predict(X_test)

print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

```
0.9825581395348837
[[906  10]
 [  8 108]]
0.9152542372881356
```

In [116]:

```
#jese yaha "bnb = BernoulliNB()" apply kate hai to

bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)

print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
```

```
0.9748062015503876
[[913   3]
 [ 23  93]]
0.96875
```

In [117]:

```
# abhi tak to "precision_score"
# gnb => 0.46948
# mnb => 0.91525
# bnb => 0.96875

# best algo bnb best perform
```

4.1 for Model build try to diffrent Vectorization Module

-> then after check (compare) what is the best perform...

In [118]:

```
# try to another Library for convert "interger (yani vectorize)"
# to isse("tranformed_text" ka sabhi text ko) "integer"(yani vector) bana hoga..
# iske liye "TfidfVectorizer Library" ka used karege


# abhi tak to columns ke sabhi text ko "vectorize" kar rahe the "CountVectorizer" and "TfidfVectorizer" ke help se
# lakin yadi hame sabse jyda use hone wale wors ko hi yadi "vectorize" karna hai to asse me
# isseme "TfidfVectorizer" me ak achha features hai ki...
# (max_features=_______) kah ke isme valuse ko dal-dal kar check kar sakete hai. bhir utana hi text(words) ko "vect

# yaha ham 3000 sabse jayda use hon eword ko hi "verctorize" kar rahe hai


from sklearn.feature_extraction.text import TfidfVectorizer
## tfidf = TfidfVectorizer()
tfidf = TfidfVectorizer(max_features=3000)
```

In [119]:

```
X = tfidf.fit_transform(df['transformed_text']).toarray()
```

In [120]:

```
X
```

Out[120]:

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

In [154]:

```
X.shape
```

Out[154]:

```
(5160, 3000)
```

In [155]:

```
# ak addictional "Scalling" apply kar rahe hai
# iske liye "MinMaxScaler" library ko import karte hai
# and X me appent (assin) kar dete hai and bhir se sabhi code run karte hai

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X = scaler.fit_transform(X)
```

In [156]:

```
y = df['target'].values
```

In [157]:

```
y
```

Out[157]:

```
array([0, 0, 1, ..., 0, 0, 0])
```

In [158]:

```
# ab "bhir se" sabhi "same" algo perform karte hai model selection ke liye
# In[113 se 123 tak]

from sklearn.model_selection import train_test_split
```

In [159]:

```
# ab naive_bayes module ka ye threes "GaussianNB,MultinomialNB,BernoulliNB" Algorithm import karte hai
#or metrices madule se "accuracy_score,confusion_matrix,precision_score"

from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [160]:

```
# sabhi algorithm ko ak "Object" me assin(store) kiya

gnb = GaussianNB()
mnd = MultinomialNB()
bnb = BernoulliNB()
```

In [161]:

```
# ab ak-ak kar sabhi algo. ko perform karte hai
#or chack karte hai kise hame best accuracy,confusion,precision milega
#jese yaha "gnb = GaussianNB()" apply kate hai to

gnb.fit(X_train,y_train)
y_pred1 = gnb.predict(X_test)

print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
```

```
0.875
[[803 113]
 [ 16 100]]
0.4694835680751174
```

In [162]:

```
#jese yaha "mnd = MultinomialNB()" apply kate hai to

mnd.fit(X_train,y_train)
y_pred2 = mnd.predict(X_test)

print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

```
0.9825581395348837
[[906  10]
 [  8 108]]
0.9152542372881356
```

In [163]:

```
#jese yaha "bnb = BernoulliNB()" apply kate hai to

bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)

print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
```

```
0.9748062015503876
[[913   3]
 [ 23  93]]
0.96875
```

In [164]:

```
# abhi tak countVectorizer (cv) ->>mnb me change huwaha hai
# lekin vieo me tfidf--> MNB chona hai

# try to more model selection and those Algorithms
```

In [165]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences',
       'transformed_text'],
      dtype='object')
```

In [166]:

```
##########################MORE###################################
```

In [167]:

```
#requred library install now

!pip install xgboost
```

```
Requirement already satisfied: xgboost in /home/cdac/.local/lib/python3.8/site-packages (1.7.5)
Requirement already satisfied: numpy in /home/cdac/.local/lib/python3.8/site-packages (from xgboost)
(1.24.3)
Requirement already satisfied: scipy in /home/cdac/.local/lib/python3.8/site-packages (from xgboost)
(1.10.1)
```

In [168]:

```
# 1st. select diffrent selection_model
# (sabhi ak hi bar me result nikal kar compare karege
# or phir ak achha score dene wale algorithm ko chunege)

from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
```

In [169]:

```
# sabhi algorithm ko ak "Object" me assin(store) kiya

svc = SVC(kernel='sigmoid', gamma=1.0)
knc = KNeighborsClassifier()
mnb = MultinomialNB()
dtc = DecisionTreeClassifier(max_depth=5)
lrc = LogisticRegression(solver='liblinear', penalty='l1')
rfc = RandomForestClassifier(n_estimators=50, random_state=2)
abc = AdaBoostClassifier(n_estimators=50, random_state=2)
bc = BaggingClassifier(n_estimators=50, random_state=2)
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)
gbdt = GradientBoostingClassifier(n_estimators=50, random_state=2)
xgb = XGBClassifier(n_estimators=50, random_state=2)
```

In [170]:

```
# ak Disnary bana kar sabhi object ko key dediye

clfs = {
    'SVC' : svc,
    'KN' : knc,
    'NB' : mnb,
    'DT' : dtc,
    'LR' : lrc,
    'RF' : rfc,
    'AdaBoost' : abc,
    'BgC' : bc,
    'ETC' : etc,
    'GBDT' : gbdt,
    'xgb' : xgb
}
```

In [171]:

```
# ab funtion banaya "train_classifier" "clf.fit()" par apna "X_train,y_train" run
def train_classifier(clf,X_train,y_train,X_test,y_test):
    clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)
    precision = precision_score(y_test,y_pred)

    return accuracy,precision
```

In [172]:

```
# as a example for "SVC" perfrom on "train_classifier"

train_classifier(svc,X_train,y_train,X_test,y_test)
```

Out[172]:

```
(0.9331395348837209, 0.688)
```

In [173]:

```
# as a example for "MultinomialNB" perfrom on "train_classifier"

train_classifier(mnb,X_train,y_train,X_test,y_test)
```

Out[173]:

```
(0.9825581395348837, 0.9152542372881356)
```

In [174]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences',
       'transformed_text'],
      dtype='object')
```

In [175]:

```
#uus "Disnary" par apply karte hai
#by using "train_classifier" ke "clf" ke through
# or "X_train,y_train" ko apply karke "X_test,y_test" run karate hai
accuracy_scores = []
precision_scores = []
for name,clf in clfs.items():

    current_accuracy,current_precision = train_classifier(clf, X_train,y_train,X_test,y_test)

    print("For ",name)
    print("Accuracy - ",current_accuracy)
    print("Precision - ",current_precision)

    accuracy_scores.append(current_accuracy)
    precision_scores.append(current_precision)
```

```
For  SVC
Accuracy -  0.9331395348837209
Precision -  0.688
For  KN
Accuracy -  0.9253875968992248
Precision -  1.0
For  NB
Accuracy -  0.9825581395348837
Precision -  0.9152542372881356
For  DT
Accuracy -  0.9428294573643411
Precision -  0.9384615384615385
For  LR
Accuracy -  0.9718992248062015
Precision -  0.9306930693069307
For  RF
Accuracy -  0.9718992248062015
Precision -  1.0
For  AdaBoost
Accuracy -  0.9699612403100775
Precision -  0.9207920792079208
For  BgC
Accuracy -  0.9660852713178295
Precision -  0.9354838709677419
For  ETC
Accuracy -  0.9718992248062015
Precision -  0.978021978021978
For  GBDT
Accuracy -  0.9563953488372093
Precision -  0.9493670886075949
For  xgb
Accuracy -  0.9757751937984496
Precision -  0.9504950495049505
```

In [176]:

```
# "Pricision ke adhar par sabse "best score" wale ko acessending order(false) yai decending oder me bada se chota->

performance_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Precision':precision_scores}).so
```

In [177]:

```
# sortlist best sore of Pricision ko print kar rahe hai

performance_df
```

Out[177]:

	Algorithm	Accuracy	Precision
1	KN	0.925388	1.000000
5	RF	0.971899	1.000000
8	ETC	0.971899	0.978022
10	xgb	0.975775	0.950495
9	GBDT	0.956395	0.949367
3	DT	0.942829	0.938462
7	BgC	0.966085	0.935484
4	LR	0.971899	0.930693
6	AdaBoost	0.969961	0.920792
2	NB	0.982558	0.915254
0	SVC	0.933140	0.688000

In [178]:

```
#ab isse ak bar graph me show karte hai

performance_df1 = pd.melt(performance_df, id_vars = "Algorithm")
```

In [179]:

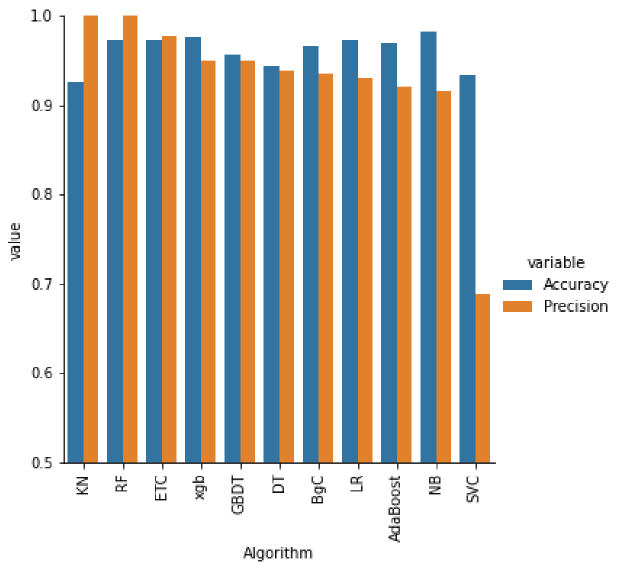
```
performance_df1
```

Out[179]:

	Algorithm	variable	value
0	KN	Accuracy	0.925388
1	RF	Accuracy	0.971899
2	ETC	Accuracy	0.971899
3	xgb	Accuracy	0.975775
4	GBDT	Accuracy	0.956395
5	DT	Accuracy	0.942829
6	BgC	Accuracy	0.966085
7	LR	Accuracy	0.971899
8	AdaBoost	Accuracy	0.969961
9	NB	Accuracy	0.982558
10	SVC	Accuracy	0.933140
11	KN	Precision	1.000000
12	RF	Precision	1.000000
13	ETC	Precision	0.978022
14	xgb	Precision	0.950495
15	GBDT	Precision	0.949367
16	DT	Precision	0.938462
17	BgC	Precision	0.935484
18	LR	Precision	0.930693
19	AdaBoost	Precision	0.920792
20	NB	Precision	0.915254
21	SVC	Precision	0.688000

In [180]:

```
sns.catplot(x = 'Algorithm', y='value',
            hue = 'variable',data=performance_df1, kind='bar',height=5)
plt.ylim(0.5,1.0)
plt.xticks(rotation='vertical')
plt.show()
```



In [181]:

```
# hame tfidf = TfidfVectorizer(max_features=3000) par most usegae words par only verctorize lagane par bhi koi chang
```

In [191]:

```
print(df.columns)
```

```
Index(['target', 'text', 'num_characters', 'num_words', 'num_sentences',
       'transformed_text'],
      dtype='object')
```

In [192]:

```
# phir se ak new "temp_df" ke nam se "Pricision ke adhar par sabse "best score" wale ko acessending order(false) yai


# "Pricision ke adhar par sabse "best score" wale ko acessending order(false) yai decending oder me "bada se chota"

#

performance_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Precision':precision_scores}).so
```

In [193]:

```
performance_df
```

Out[193]:

	Algorithm	Accuracy	Precision
1	KN	0.925388	1.000000
5	RF	0.971899	1.000000
8	ETC	0.971899	0.978022
10	xgb	0.975775	0.950495
9	GBDT	0.956395	0.949367
3	DT	0.942829	0.938462
7	BgC	0.966085	0.935484
4	LR	0.971899	0.930693
6	AdaBoost	0.969961	0.920792
2	NB	0.982558	0.915254
0	SVC	0.933140	0.688000

In [194]:

```
performance_df1 = pd.melt(performance_df, id_vars = "Algorithm")
```

In [195]:

```
performance_df1
```

Out[195]:

	Algorithm	variable	value
0	KN	Accuracy	0.925388
1	RF	Accuracy	0.971899
2	ETC	Accuracy	0.971899
3	xgb	Accuracy	0.975775
4	GBDT	Accuracy	0.956395
5	DT	Accuracy	0.942829
6	BgC	Accuracy	0.966085
7	LR	Accuracy	0.971899
8	AdaBoost	Accuracy	0.969961
9	NB	Accuracy	0.982558
10	SVC	Accuracy	0.933140
11	KN	Precision	1.000000
12	RF	Precision	1.000000
13	ETC	Precision	0.978022
14	xgb	Precision	0.950495
15	GBDT	Precision	0.949367
16	DT	Precision	0.938462
17	BgC	Precision	0.935484
18	LR	Precision	0.930693
19	AdaBoost	Precision	0.920792
20	NB	Precision	0.915254
21	SVC	Precision	0.688000

In [196]:

```
# isme jo hamne 3000 most used common ko jo victor kiya tha usee ""Precision" ke adhhar par "sort" karte hai and ak

temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_3000':accuracy_scores,'Precision_max_ft_3000':prec
```

In [197]:

```
temp_df
```

Out[197]:

	Algorithm	Accuracy_max_ft_3000	Precision_max_ft_3000
1	KN	0.925388	1.000000
5	RF	0.971899	1.000000
8	ETC	0.971899	0.978022
10	xgb	0.975775	0.950495
9	GBDT	0.956395	0.949367
3	DT	0.942829	0.938462
7	BgC	0.966085	0.935484
4	LR	0.971899	0.930693
6	AdaBoost	0.969961	0.920792
2	NB	0.982558	0.915254
0	SVC	0.933140	0.688000

In [198]:

```
# ab Scalling apply kar ke dekhate hai sayad achha precision ke sath Accurecay bhi mile
# isme hamene jo scaling ko apply kiya tha usee ""Precision" ke adhhar par "sort" karte hai and ak seprat colums ba

temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_scaling':accuracy_scores,'Precision_scaling':precision_sc
```

In [199]:

```
# "temp_df" ko table "performance_df" me "merge" karte hai or "new_df" variable me store karte hai

new_df = performance_df.merge(temp_df,on='Algorithm')
```

In [202]:

```
# ussi parkar "temp_df" ko "new_df" me "merge" karte hai and new_df_scaled nam ke variable assin (store) karte hai
# lakin abhi tak hamra table "performance_df" hi hai

new_df_scaled = new_df.merge(temp_df,on='Algorithm')
```

In [203]:

```
#ab phir privious table "performance_df" "text Sortlist" jo pure text ko "vectorize" kiya tha
# usme current table "temp_df" jo most common used hone wale ko hi "Vectorzie" kiya tha
#
# yaha privious table me hi current ke columns (yani current table) ko "Marge karte hai"


new_df_scaled
```

Out[203]:

	Algorithm	Accuracy	Precision	Accuracy_scaling_x	Precision_scaling_x	Accuracy_scaling_y	Precision_scaling_y
0	KN	0.925388	1.000000	0.925388	1.000000	0.925388	1.000000
1	RF	0.971899	1.000000	0.971899	1.000000	0.971899	1.000000
2	ETC	0.971899	0.978022	0.971899	0.978022	0.971899	0.978022
3	xgb	0.975775	0.950495	0.975775	0.950495	0.975775	0.950495
4	GBDT	0.956395	0.949367	0.956395	0.949367	0.956395	0.949367
5	DT	0.942829	0.938462	0.942829	0.938462	0.942829	0.938462
6	BgC	0.966085	0.935484	0.966085	0.935484	0.966085	0.935484
7	LR	0.971899	0.930693	0.971899	0.930693	0.971899	0.930693
8	AdaBoost	0.969961	0.920792	0.969961	0.920792	0.969961	0.920792
9	NB	0.982558	0.915254	0.982558	0.915254	0.982558	0.915254
10	SVC	0.933140	0.688000	0.933140	0.688000	0.933140	0.688000

In [ ]:

```

```

In [153]:

```
# yaha RF se hame best output mil raha hai kuki
# RF (Acc-> 0.971899 , Pre-> 1.000000)
# RF (Max_Acc-> 0.971899 , max_Pre-> 1.00000)
# RF (Scal_Acc-> 0.971899 , Scal_Pre-> 1.000000)

# yani dono ka "acc and pre" dono high score hai
```

In [204]:

```
# ab ak new "Voting Classifier"
# ko used karke dekhte hai ki multiple Algorithm ko campare karke
#vote karega ki kon sa algorithm best score de sakta hai
# example for "svm","mnb" and "etc" ye tino me voting karte hai

# svc = SVC(kernel='sigmoid', gamma=1.0)
# mnb = MultinomialNB()
# etc = ExtraTreesClassifier(n_estimators=50, random_state=2)

# from sklearn.ensemble import VotingClassifier
```

In [205]:

```
# voting = VotingClassifier(estimators=[('svc',svc), ('nb',mnb), ('et',etc)],voting='soft')
```

In [206]:

```
# voting.fit(X_train,y_train)
```

Out[206]:

```
▾                                   VotingClassifier
VotingClassifier(estimators=[('svc', SVC(gamma=1.0, kernel='sigmoid')),
                             ('nb', MultinomialNB()),
                             ('et',
                              ExtraTreesClassifier(n_estimators=50,
                                                   random_state=2))],
                 voting='soft')
               svc                        nb                         et
▾          SVC               ▾                  ▾              ExtraTreesClassifier
SVC(gamma=1.0, kernel='sigmoi   MultinomialNB    ExtraTreesClassifier(n_estimators=50, random_state=
d')                            MultinomialNB    2)
                               ()
```

In [210]:

```
# y_pred = voting.predict(X_test)
# print("Accuracy",accuracy_score(y_test,y_pred))
# print("Precision",precision_score(y_test,y_pred))
```

In [211]:

```
new_df_scaled
```

Out[211]:

	Algorithm	Accuracy	Precision	Accuracy_scaling_x	Precision_scaling_x	Accuracy_scaling_y	Precision_scaling_y
0	KN	0.925388	1.000000	0.925388	1.000000	0.925388	1.000000
1	RF	0.971899	1.000000	0.971899	1.000000	0.971899	1.000000
2	ETC	0.971899	0.978022	0.971899	0.978022	0.971899	0.978022
3	xgb	0.975775	0.950495	0.975775	0.950495	0.975775	0.950495
4	GBDT	0.956395	0.949367	0.956395	0.949367	0.956395	0.949367
5	DT	0.942829	0.938462	0.942829	0.938462	0.942829	0.938462
6	BgC	0.966085	0.935484	0.966085	0.935484	0.966085	0.935484
7	LR	0.971899	0.930693	0.971899	0.930693	0.971899	0.930693
8	AdaBoost	0.969961	0.920792	0.969961	0.920792	0.969961	0.920792
9	NB	0.982558	0.915254	0.982558	0.915254	0.982558	0.915254
10	SVC	0.933140	0.688000	0.933140	0.688000	0.933140	0.688000

final Pipline create

select RF Alog

In [ ]:

```

```