	importing the libraries				
In [1]:	<pre>import numpy as np import pandas as pd import nltk</pre>				
	read the data set				
In [2]:	<pre>sms=pd.read_csv('spam.csv', encoding='latin-1') sms.head()</pre>				
Out[2]:	v1 v2 Unnamed: 2 Unnamed: 3 Unnamed: 4 0 ham Go until jurong point, crazy Available only NaN NaN NaN	_			
	1 ham Ok lar Joking wif u oni NaN NaN NaN 2 spam Free entry in 2 a wkly comp to win FA Cup fina NaN NaN NaN 3 ham Hall dup say so early har. Hall dup say so early har. Hall dup say so early har. NaN NaN NaN				
	 3 ham U dun say so early hor U c already then say NaN NaN NaN 4 ham Nah I don't think he goes to usf, he lives aro NaN NaN NaN NaN 				
In [3]:	<pre>sms=sms.drop(["Unnamed: 2","Unnamed: 3","Unnamed: 4"],axis=1) sms=sms.rename(columns={"v1":"label","v2":"text"}) sms.head()</pre>				
Out[3]:	label text 0 ham Go until jurong point, crazy Available only				
	 1 ham Ok lar Joking wif u oni 2 spam Free entry in 2 a wkly comp to win FA Cup fina 2 ham Aldura say on early box. Handkoody them as your 				
	3 ham U dun say so early hor U c already then say4 ham Nah I don't think he goes to usf, he lives aro				
	Explore the data				
In [4]:	<pre>print(" no of rows", len(sms)) no of rows 5572</pre>				
<pre>In [5]: Out[5]:</pre>	sms.label.value_counts() ham 4825 spam 747				
In [6]:	Name: label, dtype: int64 sms.describe()				
Out[6]:	count 5572 5572				
	top ham Sorry, I'll call later freq 4825 30				
In [7]:	<pre>sms['length']=sms['text'].apply(len) sms.head()</pre>				
Out[7]:	label text length O ham Go until jurong point, crazy Available only 111				
	1hamOk lar Joking wif u oni292spamFree entry in 2 a wkly comp to win FA Cup fina155				
	 ham U dun say so early hor U c already then say ham Nah I don't think he goes to usf, he lives aro 61 				
	Plot				
In [8]:	<pre>import matplotlib.pyplot as plt import seaborn as sns</pre>				
Out[8]:	<pre>sms['length'].plot(bins=50, kind='hist',figsize=(10,7)) <axes: ylabel="Frequency"></axes:></pre>				
	1400 -				
	1200 -				
	1000 -				
	edneucy				
	은 600 -				
	400 -				
	200 -				
In [9]:	0 200 400 6 sms.hist(column='length', by='label', bins=50, figsize=(15,8))	500 8	00		
Out[9]:	<pre>array([<axes: 'ham'}="" title="{'center':">, <axes: 'spam'}="" title="{'center':"></axes:></axes:></pre>],			
	ham	140 -		spam	
	1400 -				
	1200 -	120 -			
	1000 -	100 -			
	800 -	80 -			
	600 -	60 -			
	400 -	60 - 40 -			
	400 -	40 -			
To [40].	400-200-000	40 - 20 -	05	150 -	200 -
In [10]:	<pre>sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprec</pre>	ationWarning: In a f	uture version, `df.iloo	c[:, i] = newvals` will attempt	to set the v
In [10]:	<pre>sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head()</pre>	ationWarning: In a f	uture version, `df.iloo	c[:, i] = newvals` will attempt	to set the v
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecalues inplace instead of always setting a new array. To retain the old be setitem(i, newvals)` sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a f	uture version, `df.iloo	c[:, i] = newvals` will attempt	to set the v
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecalues inplace instead of always setting a new array. To retain the old be setitem(i, newvals)' sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a f	uture version, `df.iloo	c[:, i] = newvals` will attempt	to set the v
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecalues inplace instead of always setting a new array. To retain the old be setitem(i, newvals)` sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a f	uture version, `df.iloo	c[:, i] = newvals` will attempt	to set the v
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecalues inplace instead of always setting a new array. To retain the old be setitem(i, newals)' sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a f	uture version, `df.iloo	c[:, i] = newvals` will attempt	to set the v
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecalues inplace instead of always setting a new array. To retain the old b setitem(i, newvals) sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the vunique, `df.i
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprec alues inplace instead of always setting a new array. To retain the old b setitem(i, newvals) sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Depred alues inplace instead of always setting a new array. To retain the old be settiem(i, newvals)' sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the vunique, `df.i
Out[10]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecallues inplace instead of always setting a new array. To retain the old be setitem(i, newvals)' sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the vunique, `df.i
Out[10]: In [11]:	sms.loc[:, 'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecatives inplace instead of always setting a new array. To retain the old by settiem(i, newvals) sms.loc[:, 'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the vunique, `df.i
Out[10]: In [11]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Depred alues inplace instead of always setting a new array. To retain the old be settiem(1, newvals)' sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]:	sms.loc[;,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecatives inplace instead of always setting a new array. To retain the old bestitem(i, newvals)' sms.loc[;,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]:	sms.loc[:, 'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\INDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: Deprecatives inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace instead of always setting a new array. To retain the old best-time inplace in least length text length 1 0 Go until jurong point, crazy. Available only	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]: Out[13]:	sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) sms.head() C:\Users\NDIA\AppData\Local\Temp\ipykernel_7232\1727894437.py:1: beprea lues inplace instead of always setting a new array. To retain the old bestitem(i, newsls)' sms.loc[:,'label']=sms.label.map({'ham':0, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]: Out[13]:	sms.loc[:,'label']=sms.label.map(('ham':0, 'spam':1)) sms.head() C:\Users\Indianal\topata\Local\Temp\iny\ternel_723\\727894437.py:1: Deprecalues inplace instead of always setting a new array. To retain the old be settiem(i, newvals) sms.loc[:,'label']=sms.label.map(('ham':0, 'spam':1)) label text length 0 Go until jurong point, crazy. Available only 111 1 O Ok lar Joking wift uni 29 2 1 Free entry in 2 a wkly comp to win FA Cup fina 155 3 O Udun say so early hor U c already then say 49 4 O Nah I don't think he goes to usf, he lives aro 61 Implement Bag of Words from sklearn.feature_extraction.text import Countvectorizer from sklearn.model_selection import train_test_split count=Countvectorizer() input=['REMINDER FROM 02: To get 2.50 pounds free call credit and detail text=count.fit_transform(sms['text']) x_train, x_test, y_train, y_test= train_test_split(text, sms['label'], t text vith 73916 stored elements in Compressed Sparse Row format> print(x_train.shape) print(x_text.shape) print(x_text.shape) input=ext[5571] (4457, 8672) implementation of ML Model from sklearn.neural_network import MLPClassifier model=MLPClassifier() model.fit(x_train, y_train) *MLPClassifier	ationWarning: In a fehavior, use either	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]: Out[13]:	sms.loc[:,'label']=sms.label.map(('ham':0, 'spam':1)) sms.head() C:\Users\INDIA\AppData\Local\Temp\inp\text{remp\inp\text{remp\inp\text{rain}}, roretain the old by settlem(i, newvals) sms.loc[:,'label']=sms.label.map(('ham':0, 'spam':1)) label text length 0	ationWarning: In a fehavior, use either est_size=0.20, rando	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]: In [14]:	sms.loc[:,'label']=sms.label.map(('ham':e, 'spam':1)) sms.head() C:\Users\INDIA\AppData\Loca\Temp\ipykernel_7232\1727894437.py:1: Depred alues inplace instead of always setting a new array. To retain the old b setitem(i, newvals)' sms.loc[:,'label']=sms.label.map({'ham':e, 'spam':1}) label text length 0	ationWarning: In a fehavior, use either est_size=0.20, rando	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]: In [14]:	sms.loc(; 'label']=sms.label.map(('ham':0, 'spam':1)) sms.head() C:\Users\INDIA\Appdata\Loca\Temp\ipykernel 7232\1727894437.py:1: Deprecatives implace instead of always setting a new array. To retain the old be settlem(1, newvals) sms.loc(; 'label']=sms.label.map(('ham':0, 'spam':1)) label text length 0	ationWarning: In a fehavior, use either est_size=0.20, rando	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]:	sms.loc[:,'label']=sms.label.map(['ham':8, 'spam':1)) sms.head() C'.\Users\INDIA\AppData\Local\Temp\lpykernel_7232\1727894437.py:1: Deprecatues implace instead of always setting a new array. To retain the old b settient(in newvals) sms.loc[:,'label']=sms.label.map(['ham':8, 'spam':1)) sms.loc[:,'label']=sms.label.map(['ham':8, 'spam':1)) sms.loc[:,'label']=sms.label.map(['ham':8, 'spam':1)) sms.loc[:,'label']=sms.label.map("ham':8, 'spam':1)] sms.loc[ationWarning: In a fiehavior, use either e, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]: In [14]:	sms.loc[:,'label']=sms.label.map('ham':0, 'spam':1)) sms.head() C.Vlser*NLNDIAA/AppDatatucallYcemphipykernel_22221727894427,py:1: Deprecalues injace instead of always setting a new array. To retain the old be settien(1, newvals)' sms.loc[:,'label']=sms.label.map(('ham':0, 'spam':1)) label text length 0	ationWarning: In a fehavior, use either ee, f1_score ee, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]:	sms.loc[:,'label']=sms.label.map({'ham':8, 'spam':1}) sms.head() C:\Users\NtDIANAppbata\Local\Temp\inykernel_732\172\footnote{1} body sms.loc[:,'label']=sms.label.map({'ham':8, 'spam':1}) sms.loc[:,'label']=sms.label.map({'ham':8, 'spam':1}) sms.loc[:,'label']=sms.label.map(('ham':9, 'spam':1)) smb.loc[:,'label']=sms.label.map(('ham':9, 'spam':1)) sms.loc[:,'label']=sms.label.map(('ham':9, 'spam':1)) sms.label text length	ationWarning: In a fehavior, use either ee, f1_score ee, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]:	sms.loc[; 'label' sms.label.map(['ham':0, 'spam':1]) sms.locd() C:Users\INDIA\AppBata\Local\Temp\ipykernel_723\\T27894437.py:1: Deprecations and interest of always setting a new array. To retain the old bestitetie, newals] sms.loc[; 'label']=sms.label.map(('ham':0, 'spam':1)) tabel text length 0 Countiljuroppoint crazy.Available only. 111 1 O CharJoining wifu onl 29 2 1 Free entry in 2 arkly compto win FA Cup fina 155 3 O U dis asy so early hor Ur already them say 43 4 O Nan I don't think he goes to usf. he lives are 61 Implement Bag of Words from sklearn.feature_extraction.text import CountVectorizer from sklearn.model_selection import frain_test_split count=CountVectorizer() Input=['REKINDER FROM 02: To get 2.50 pounds free call credit and detail text=count.filt_transform(smst'(ext')) x.train, x.test, y.train, y.test=train_test_split(text, sms['label'), t text <pre>cstsystems artrix of type 'cclass 'numpy.int64's' with 73816 stored elements in Compressed Sparse Row format> print(x.train.shape) print(x.test.shape) input=text[5572] (4457, 8672) implementation of ML Model from sklearn.neural_network import MLPClassifier model=MLPClassifier() model-MLPClassifier() w.HPClassifier() #HPClassifier() #HPClassifier() print("Mecuracy soore: ()" format(precall score(y.test, prediction)) print("Mecall score: ()" format(precall score(y.test, prediction)) print("Mecall score: ()" format(precall score(y.test, prediction)) mint("Mecall score: ()" format(precall score(y.test, prediction)) mint("Mecall score: ()" format(precall score(y.test, prediction)) mint("Mecall score: ()" format(precall score(y.test, prediction)) print("Mecall score: ()" format(precall score(y.test, prediction)) mint("Mecall score: ()" format(precall score(y.test, p</pre>	ationWarning: In a fehavior, use either ee, f1_score ee, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]:	sms.loc[; 'label']=sms.label.map(('ham':8, 'spam':1)) sms.label; sms.label.map(('ham':8, 'spam':1)) sms.label; smplace instead of always setting a new array. To retain the old besetted(), newvals]' sms.loc[; 'label']=sms.label.map(('ham':8, 'spam':1)) label to the string and	ationWarning: In a fehavior, use either e, f1_score e, f1_score e, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]: In [16]:	sms.loc(: 'label']=sms.label.map(('nam':8, 'spam':1)) sms.loc(: 'label']=sms.label.map(('nam':8, 'spam':1)) sms.loc(: 'label']=sms.label.map(('nam':8, 'spam':1)) sms.loc(: 'label']=sms.label.map(('ham':8, 'spam':1)) tabel text length 0	ationWarning: In a fehavior, use either se, f1_score re, f1_score re, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]: In [16]:	ses.loc[:,'label']=ses.label.map(('ham':e, 'spam':1)) ses.loc[:,'label']=ses.label.map(('ham':e, 'spam':1)) c.\Users\text{INDTA\Apporta\Local\Temp\iny\text{ser.local}} c.\Users\text{INDTA\Apporta\Local\Temp\iny\text{ser.local}}, sem:'1)) inbel c.\Users\text{INDTA\Apporta\Local\Temp\iny\text{ser.local}}, sem:'1) inbel c.\Users\text{INDTA\Apporta\Local\Temp\iny\text{ser.local}}, sem:'1) inbel c.\Users\text{local\temp\iny\text{ser.local\temp\iny\text{ser.local}}} c.\Userslocal\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\text{ser.local\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\iny\temp\i	ationWarning: In a fehavior, use either se, f1_score re, f1_score re, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: Out[11]: In [12]: In [14]: In [16]:	sns.loc[:,'label']sss.label.map(('ham':o, 'span':1)) sns.head() C-Wiers/NDIAN/MyDotat.boral/TempAtpyberrel. 7229/172784437.py:1: Depretations.head() subset inplace indexed of always setting a new array. To retain the old be settled(i, newvals) sms.loc[:,'label']sss.label.map(('ham':o, 'span':1)) label to the subset of th	ationWarning: In a fehavior, use either se, f1_score re, f1_score re, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]: In [16]:	sns.loc[:,'label']sss.label.map(('ham':o, 'span':1)) sns.head() C-Wiers/NDIAN/MyDotat.boral/TempAtpyberrel. 7229/172784437.py:1: Depretations.head() subset inplace indexed of always setting a new array. To retain the old be settled(i, newvals) sms.loc[:,'label']sss.label.map(('ham':o, 'span':1)) label to the subset of th	ationWarning: In a fehavior, use either se, f1_score re, f1_score re, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [13]: In [13]: In [14]: In [16]: In [17]:	ans.ioc[; 'label']sens.label.map('ham':0, 'spam':1)) sens.ioc[c, 'label']sens.label.map('ham':0, 'spam':1)) sens.ioc[c, 'label']sens.label.map('ham':0, 'spam':1)) sens.ioc[c, 'label']sens.label.map('ham':0, 'spam':1)) label tost length to	ationWarning: In a fehavior, use either se, f1_score re, f1_score re, f1_score	uture version, `df.iloo `df[df.columns[i]] = ne	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [12]: In [14]: In [16]: Out[19]:	ses.loc[s.*label*]sess.label.map("inan*20, "span*12)] ses.leca() C. Users*NDOCA-NDOPA Applicational New Proprint ("span*21) ses.leca() C. Users*NDOCA-NDOPA Applicational New Proprint ("span*21) sestimatic, indicational New Proprint ("span*21) 1	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out [10]: In [11]: Out [13]: In [14]: In [15]: In [16]: In [17]:	sms. loc(, 'label')sms. label.mag(('mam':o, 'span':1)) sms. loc(), 'label')sms. label.mag(('mam':o, 'span':1)) sms. label.mag(('mam':o, 'span':n, 'span':1)) sms. label.mag(('mam':o, 'span':n, 'span':n, 'span':1)) sms. label.mag(('mam':o, 'span':n, 's	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out [10]: In [11]: Out [13]: In [12]: In [15]: In [16]: In [17]:	sms.loci; 'label']smms.lamel.map('ham':0, 'spam':1) sms.head() sms	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out [10]: In [11]: Out [13]: In [14]: In [15]: In [16]: In [17]:	ans. loc(:, 'label')sans. label.nap('man'io, 'span'il)) sins. Read() s	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out [10]: In [11]: Out [13]: In [14]: In [15]: In [16]: In [17]:	sms.loc(; 'label') smms.label.man('ham's, 'span's) sms.loc(; 'label') smms.label.man('ham's, 'span's) sms.loc(; 'label') sms.label.man('ham's, 'span's) sms.label') sms.	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [13]: In [14]: In [15]: In [16]: In [20]: In [21]: Out[21]:	smaller(s, "label") smaller(smaller) specifies (smaller) Childer Stabel "Jesus Label mapy ("ther"s, "specifies) section (smaller) section	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the v unique, `df.i
Out[10]: In [11]: Out[11]: In [13]: In [14]: In [16]: In [17]: In [20]: In [21]: Out[21]:	ss. loci; label* jesms.label.mau("man"s, "spam"(2)) ss. loci; label* jesms.label.mau("man"s, "spam"(2)) ss. head; label* jesms.label.mau("man"s, "spam"(2)) ss. head; label* jesms.label.mau("man"s, "spam"(2)) sse loci and jesms.label.mau("man"s, "spam"(3)) label* sess.loci; label* jesms.label.mau("man"s, "spam"(3)) complement Bag of Words label* sess.loci; label* sess.loci and sess.lo	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the vunique, `df.i
Out [10]: In [11]: Out [11]: In [13]: In [13]: In [16]: In [16]: In [20]: In [21]: Out [21]: Out [21]:	ses_locitimed_ines_label_nep(('nan'lo, 'spew'll) ses_loci_timed_ines_label_nep(('nan'lo, 'spew'll) ses_loci_timed_ines_label_nep(('nan'lo, 'spew'll) ses_loci_timed_ines_label_nep(('nan'lo, 'spew'll) ses_loci_timed_ines_label_nep(('nan'lo, 'spew'll) ses_loci_timed_ines_label_nep('han'lo, 'spew'll) bes_loci_timed_ines_label_nep('han'lo, 'spew'll) bes_loci_timed_ines_label_nep('han'lo, 'spew'll) 1	ationWarning: In a fehavior, use either s of great offers pl dest_size=0.20, rando re, f1_score re, f1_score	s reply 2 this text win state=1)	c[:, i] = newvals` will attempt ewvals` or, if columns are non-	to set the vunique, `df.i