import dataset

```
[ ] import pandas as pd
```

```
[ ] import nltk
    import re
    import string
    string.punctuation
    from nltk.stem.porter import PorterStemmer
    from nltk.corpus import stopwords
    from nltk.stem import WordNetLemmatizer
    import nltk
    nltk.download('wordnet')

    def tokenization(text):
        tokens = text.split(' ')
        return tokens

    def remove_punctuation(text):
        punctuationfree=[]
        for t in text:
            punctuationfree.append(''.join(i for i in t if i not in string.punctuation))
        return punctuationfree


    nltk.download('stopwords')
    stopwords=stopwords.words('english')

    def remove_stopwords(text):
        output= [i for i in text if i not in stopwords]
        return output


    porter_stemmer = PorterStemmer()

    def stemming(text):
        stem_text = [porter_stemmer.stem(word) for word in text]
        return stem_text

    def lematize(text):
        lemmatizer = WordNetLemmatizer()
        lem_text=[lemmatizer.lemmatize(word) for word in text]
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
[ ] """import json

    with open('dict.json') as json_file:
        data = json.load(json_file)"""
```

```
'import json\n\nwith open('dict.json') as json_file:\n    data = json.load(json_file)'
```

```
[ ] import warnings
    warnings.filterwarnings('ignore')
```

```
[ ] import pandas as pd

    df = pd.read_excel(r'Data.xlsx')
```

```
[ ] df['Processed Text']=df['Summary Text'].str.replace('\d+', '')
    df['Processed Text']= df['Processed Text'].apply(lambda x: x.lower())
    df['Processed Text']=df['Processed Text'].apply(lambda x: tokenization(x))
    df['Processed Text']=df['Processed Text'].apply(lambda x: remove_punctuation(x))
    df['Processed Text']=df['Processed Text'].apply(lambda x: remove_stopwords(x))
    df['Processed Text']=df['Processed Text'].apply(lambda x: stemming(x))
```

```
[ ] """nltk.download('omw-1.4')
    df['Processed Text']=df['Processed Text'].apply(lambda x: lematize(x))"""
```

```
'nltk.download('omw-1.4')\ndf['Processed Text']=df['Processed Text'].apply(lambda x: lematize(x))'
```

```
[ ] df['Processed Text']=df['Processed Text'].apply(lambda x: ' '.join(x))
```

```
[ ] df.head(5)
```

	CVE ID Number	Vulnerability Type	Summary Text	Processed Text
0	CVE-2022-43766	Denial of Service	Apache IoTDB version 0.12.2 to 0.12.6, 0.13.0 ...	apach iotdb version vulner denial servic a...
1	CVE-2022-43365	Denial of Service	IP-COM EW9 V15.11.0.14(9732) was discovered to...	ipcom ew v discov contain buffer overflow form...
2	CVE-2022-43035	Denial of Service	An issue was discovered in Bento4 v1.6.0-639. ...	issu discov bento v heapbufferoverflow apdecat...
3	CVE-2022-43033	Denial of Service	An issue was discovered in Bento4 1.6.0-639. T...	issu discov bento bad free compon aphdlratoma...
4	CVE-2022-42969	Denial of Service	The py library through 1.11.0 for Python allow...	py librari python allow remot attack conduct ...

```
[ ] df.shape
```

```
(175850, 4)
```

```
[ ] df.sample(10)
```

	CVE ID Number	Vulnerability Type	Summary Text	Processed Text
146884	CVE-2008-0555	Bypass	The ExpandCert function in Apache-SSL before a...	expandcert function apachessl apachessl proper...
89127	CVE-2008-3143	Overflow	Multiple integer overflows in Python before 2....	multipl integ overflow python might allow con...
16037	CVE-2013-5385	Denial of Service	The OSPF implementation in IBM i 6.1 and 7.1, ...	ospf implement ibm zo zseri server network o...
10042	CVE-2016-8518	Denial of Service	A remote denial of service vulnerability in HP...	remot denial servic vulner hpe system insight ...
68168	CVE-2005-1670	Code Execute	Unknown vulnerability in Extreme BlackDiamond ...	unknown vulner extrem blackdiamond switch ru...
67784	CVE-2005-2927	Code Execute	Stack-based buffer overflow in ppp in SCO Unix...	stackbas buffer overflow ppp sco unixwar pos...
98474	CVE-2013-7490	Memory Corruption	An issue was discovered in the DBI module befo...	issu discov dbi modul perl use mani argument ...
103148	CVE-2019-14702	SQL Injection	An issue was discovered on MicroDigital N-seri...	issu discov microdigit nseri camera firmwar s...
130534	CVE-2007-5183	Cross Site Scripting (XSS)	Cross-site scripting (XSS) vulnerability in Ma...	crosssit script xss vulner mailboxmw odysseysu...
121344	CVE-2018-0098	Cross Site Scripting (XSS)	A vulnerability in the web-based management in...	vulner webbas manag interfac cisco wap wireles...

```
[ ]
```

Creating DataSet

```
[ ] from collections import defaultdict
    dic=defaultdict()
```

```
[ ] for x in set(df['Vulnerability Type']):
        print(x)
```

```
Bypass
Security Vulnerabilities
Overflow
Directory Traversal
Memory Corruption
gain priviledge
Http Response Splitting
Cross Site Scripting (XSS)
Gain Information
CSRF
SQL Injection
File Inclusion
Denial of Service
Code Execute
```

```
[ ] # creating short keyword, vectorizing the dataset(Tfidf vectorizer can also be used)

    new={
        'Denial of Service': 'DoS',
        'Cross Site Scripting (XSS)': 'XSS',
         'CSRF': 'CSRF',
         'Memory Corruption': 'Mem C',
         'Directory Traversal': 'Dir T',
         'Code Execute': 'Code Exec',
         'Bypass': 'Bypass',
         'Http Response Splitting': 'HTTP',
         'File Inclusion': 'File Inc',
         'Overflow': 'OverF',
         'SQL Injection': 'SQLI',
         'gain priviledge': 'Gain Priv',
         'Security Vulnerabilities': 'Sec Vul',
         'Gain Information': 'Gain inf',
    }
```

```
[ ] #df.drop(['Summary Text'],axis=1,inplace=True)
    for item in new.keys():
      df[new[item]] = [1 if x ==item else 0 for x in df['Vulnerability Type']]      #one of the 14 vulnerability type will get a 1 and all others appointed to 0,meaning the one which has 1 in DoS implies the vulnerability for the text/iss
```

```
[ ] df.sample(10) #one can see that as we go along the row only one vulnerability type recieves a bool 1 indicating current summary has the saem vulnerability type
```

	CVE ID Number	Vulnerability Type	Summary Text	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain inf
17834	CVE-2012-3562	Denial of Service	Opera before 12.00 Beta allows user-assisted r...	opera beta allow userassist remot attack caus...	1	0	0	0	0	0	0	0	0	0	0	0	0	0
62857	CVE-2007-4537	Code Execute	Heap-based buffer overflow in the Huffman deco...	heapbas buffer overflow huffman decompress alg...	0	0	0	0	0	1	0	0	0	0	0	0	0	0
52291	CVE-2013-3871	Code Execute	Microsoft Internet Explorer 6 through 10 allow...	microsoft internet explor allow remot attack...	0	0	0	0	0	1	0	0	0	0	0	0	0	0
12805	CVE-2015-5846	Denial of Service	IOKit in the kernel in Apple iOS before 9 allo...	iokit kernel appl io allow attack execut arbi...	1	0	0	0	0	0	0	0	0	0	0	0	0	0
30119	CVE-2022-21878	Code Execute	Windows Geolocation Service Remote Code Execut...	window geoloc servic remot code execut vulner	0	0	0	0	0	1	0	0	0	0	0	0	0	0
35508	CVE-2020-15294	Code Execute	Compiler Optimization Removal or Modification ...	compil optim remov modif securitycrit code vul...	0	0	0	0	0	1	0	0	0	0	0	0	0	0
78833	CVE-2017-18070	Overflow	In wma_ndp_end_response_event_handler(), the v...	wmandpendresponseeventhandl variabl lenendrsp ...	0	0	0	0	0	0	0	0	0	1	0	0	0	0
117095	CVE-2020-6163	Cross Site Scripting (XSS)	The WikibaseMediaInfo extension 1.35 for Media...	wikibasemediainfo extens mediawiki allow xss ...	0	1	0	0	0	0	0	0	0	0	0	0	0	0
149261	CVE-2020-26539	Gain Information	An issue was discovered in Foxit Reader and Ph...	issu discov foxit reader phantompdf multipl i...	0	0	0	0	0	0	0	0	0	0	0	0	0	1
130541	CVE-2007-5127	Cross Site Scripting (XSS)	Multiple cross-site scripting (XSS) vulnerabil...	multipl crosssit script xss vulner simpgb all...	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Supervised Sentiment Analysis

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
```

```
[ ] #creating vectorizer
    vectorizer = TfidfVectorizer(ngram_range=(1,2),
                                 max_features=2000)
```

```
[ ] %%time
    vectorizer.fit(df['Processed Text'])
```

```
CPU times: user 11.7 s, sys: 210 ms, total: 11.9 s
Wall time: 11.9 s
TfidfVectorizer(max_features=2000, ngram_range=(1, 2))
```

```
[ ] len(vectorizer.vocabulary_)
```

```
2000
```

```
[ ] vectorizer.get_feature_names_out()[:100]
```

```
array(['abil', 'abil caus', 'abl', 'abl execut', 'absolut', 'ac',
       'accept', 'access', 'access control', 'access data',
       'access restrict', 'access sensit', 'access via', 'account',
       'achiev', 'acrobat', 'acrobat acrobat', 'acrobat reader',
       'acrobat window', 'action', 'action indexphp', 'activ', 'activex',
       'activex control', 'ad', 'adapt', 'adapt secur', 'add', 'addit',
       'addit execut', 'address', 'address improv', 'adjac', 'admin',
       'administr', 'administr credenti', 'administr execut',
       'administr privileg', 'administr request', 'adminphp', 'adob',
       'adob acrobat', 'adob air', 'adob flash', 'adob reader',
       'adob shockwav', 'advanc', 'advisori', 'affect', 'affect devic',
       'affect icloud', 'affect instal', 'affect issu', 'affect itun',
       'affect maco', 'affect product', 'affect safari', 'affect softwar',
       'affect system', 'affect tvo', 'affect version', 'affect watcho',
       'agent', 'air', 'air adob', 'air sdk', 'aix', 'ajax', 'aka',
       'aka bug', 'aka intern', 'aka internet', 'aka microsoft',
       'aka script', 'aka window', 'alloc', 'allow', 'allow arbitrari',
       'allow attack', 'allow authent', 'allow contextdepend',
       'allow high', 'allow local', 'allow mainthemiddl', 'allow owner',
       'allow physic', 'allow remot', 'allow unauthent', 'allow user',
       'allow userassist', 'allow xss', 'alreadi', 'also', 'also affect',
       'alter', 'alter intend', 'amount', 'android', 'android adob',
       'android allow'], dtype=object)
```

```
[ ] %%time
    inputs = vectorizer.transform(df['Processed Text'])
```

```
CPU times: user 7.73 s, sys: 30.8 ms, total: 7.76 s
Wall time: 7.77 s
```

```
[ ] inputs.shape
```

```
(175850, 2000)
```

```
[ ] inputs.toarray()[0][:50]
```

```
array([0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.30053715, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.19479423, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ])
```

```
[ ] targets=df['DoS']
```

Logistic Regression

```
[ ] from sklearn.linear_model import LogisticRegression #testing logistic reg. model
```

```
[ ] model = LogisticRegression()
```

```
[ ] inputs.shape,targets.shape
```

```
((175850, 2000), (175850,))
```

```
[ ] model.fit(inputs,targets)
```

```
LogisticRegression()
```

```
[ ] preds = model.predict(inputs)   #predicting the performance on the same data
```

```
[ ] df7=pd.DataFrame([targets,preds]).T
```

```
df7
```

	DoS	Unnamed 0
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
175845	0	0
175846	0	0
175847	0	0
175848	0	0
175849	0	0

175850 rows × 2 columns

```
[ ] from collections import Counter
    a,b=Counter(targets),Counter(preds)
```

```
[ ] a,b
```

```
(Counter({1: 28100, 0: 147750}), Counter({1: 25241, 0: 150609}))
```

```
[ ] df2=df.copy()
    df2['DoS1']=preds
    #df2.shape,b.shape
```

```
[ ] df2.sample(10)
```

	CVE ID Number	Vulnerability Type	Summary Text	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain inf	DoS1
130542	CVE-2007-5121	Cross Site Scripting (XSS)	Cross-site scripting (XSS) vulnerability in JS...	crosssit script xss vulner jspwiki beta allow ...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
47185	CVE-2016-4778	Code Execute	The kernel in Apple iOS before 10, OS X before...	kernel appl io os x tvo watcho allow attac...	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
129613	CVE-2008-5011	Cross Site Scripting (XSS)	Multiple cross-site scripting (XSS) vulnerabil...	multipl crosssit script xss vulner ibm lotu qu...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
92367	CVE-2003-0580	Overflow	Buffer overflow in uvadmsh in IBM U2 UniVerse ...	buffer overflow uvadmsh ibm u univers earlier...	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
121042	CVE-2018-2432	Cross Site Scripting (XSS)	SAP BusinessObjects Business Intelligence (BI ...	sap businessobject busi intellig bi launchpad ...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
170018	CVE-2009-3759	CSRF	Multiple cross-site request forgery (CSRF) vul...	multipl crosssit request forgeri csrf vulner s...	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
100311	CVE-2007-3806	Memory Corruption	The glob function in PHP 5.2.3 allows context-...	glob function php allow contextdepend attack ...	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
93274	CVE-2000-0968	Overflow	Buffer overflow in Half Life dedicated server ...	buffer overflow half life dedic server build ...	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
138766	CVE-2006-1953	Directory Traversal	Directory traversal vulnerability in Caucho Re...	directori travers vulner caucho resin window...	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
51918	CVE-2013-6283	Code Execute	VideoLAN VLC Media Player 2.0.8 and earlier al...	videolan vlc media player earlier allow remot...	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

```
[ ] df2.head()
```

	CVE ID Number	Vulnerability Type	Summary Text	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain inf	DoS1
0	CVE-2022-43766	Denial of Service	Apache IoTDB version 0.12.2 to 0.12.6, 0.13.0 ...	apach iotdb version vulner denial servic a...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	CVE-2022-43365	Denial of Service	IP-COM EW9 V15.11.0.14(9732) was discovered to...	ipcom ew v discov contain buffer overflow form...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	CVE-2022-43035	Denial of Service	An issue was discovered in Bento4 v1.6.0-639. ...	issu discov bento v heapbufferoverflow apdecat...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	CVE-2022-43033	Denial of Service	An issue was discovered in Bento4 1.6.0-639. T...	issu discov bento bad free compon aphdlratoma...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	CVE-2022-42969	Denial of Service	The py library through 1.11.0 for Python allow...	py librari python allow remot attack conduct ...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Predictiing similarly on all supervised models

Tree

```
[ ] %%time
    from sklearn import tree
    model_2 = tree.DecisionTreeClassifier()
    model_2.fit(inputs,targets)
```

```
CPU times: user 46 s, sys: 83 ms, total: 46.1 s
Wall time: 46.1 s
DecisionTreeClassifier()
```

```
[ ] preds_2 = model_2.predict(inputs)
```

```
[ ] from collections import Counter
    a,b=Counter(targets),Counter(preds_2)
```

```
[ ] a,b
```

```
(Counter({1: 28100, 0: 147750}), Counter({1: 16486, 0: 159364}))
```

Multinomial NB

```
[ ] #using Naive bayes classifier for prediction
    from sklearn.naive_bayes import MultinomialNB
```

```
[ ] model_3 = MultinomialNB()
```

```
model_3.fit(inputs,targets)
MultinomialNB()
pred_3 = model_3.predict(inputs)
```

```
from collections import Counter
a,b=Counter(targets),Counter(pred_3)
```

```
a,b
```

```
(Counter({1: 28100, 0: 147750}), Counter({0: 137806, 1: 38044}))
```

```
df_3=df.copy()
df_3['DoS1']=pred_3
```

```
df_3.sample(10)
```

	CVE ID Number	Vulnerability Type	Summary Text	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain inf	DoS1
127163	CVE-2012-2075	Cross Site Scripting (XSS)	Cross-site scripting (XSS) vulnerability in th...	crosssit script xss vulner contact save modul ...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
164926	CVE-2007-6717	gain priviledge	Buffer overflow in tftp in bos.net.tcp.client ...	buffer overflow tftp bosnettcpcli ibm aix al...	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
121486	CVE-2017-18832	Cross Site Scripting (XSS)	Certain NETGEAR devices are affected by stored...	certain netgear devic affect store xss affect ...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
76437	CVE-2019-7985	Overflow	Adobe Photoshop CC versions 19.1.8 and earlier...	adob photoshop cc version earlier earlier he...	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
175218	CVE-2009-3597	Security Vulnerabilities	Digitaldesign CMS 0.1 stores sensitive informa...	digitaldesign cm store sensit inform web root...	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
54712	CVE-2011-4047	Code Execute	The Dell KACE K2000 System Deployment Applianc...	dell kace k system deploy applianc allow remot...	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
149281	CVE-2020-25772	Gain Information	An out-of-bounds read information disclosure v...	outofbound read inform disclosur vulner trend ...	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
71272	CVE-2022-40860	Overflow	Tenda AC15 router V15.03.05.19 contains a stac...	tenda ac router v contain stack overflow vulne...	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5117	CVE-2019-2765	Denial of Service	Vulnerability in the Oracle Solaris product of...	vulner oracl solari product oracl system compo...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
87204	CVE-2011-0918	Overflow	Stack-based buffer overflow in the NRouter (ak...	stackbas buffer overflow nrouter aka router se...	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
```

```
clf = RandomForestClassifier(n_estimators = 100)
#clf.fit(inputs,targets)
#pred_3=-clf.predict(inputs)
```

```
```

ALL MODELS TOGETHER

```
from sklearn.linear_model import LogisticRegression
from sklearn import tree
#using Naive bayes classifier for prediction
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
```

```
"""model_4 = SVC(C= .1, kernel='linear', gamma= 1)
model_4.fit(inputs,targets)
preds_4=model_4.predict(inputs)
Counter(preds_4)"""
```

```
'model_4 = SVC(C= .1, kernel=\'linear\', gamma= 1)\nmodel_4.fit(inputs,targets)\npreds_4=model_4.predict(inputs)\nCounter(preds_4)'
```

```
%%time
items=['DoS','XSS','CSRF','Mem C','Dir T','Code Exec','Bypass','HTTP','File Inc','OverF','SQLI','Gain Priv','Sec Vul','Gain inf']
from collections import defaultdict
dic=defaultdict(list)
for item in items:
  # training best working models and predicting the outputs
  targets=df[item]
  dic[item].append(targets)
  model = LogisticRegression()
  model.fit(inputs,targets)
  preds=model.predict(inputs)
  dic[item].append(preds)

  model_2 = tree.DecisionTreeClassifier()
  model_2.fit(inputs,targets)
  preds_2=model_2.predict(inputs)
  dic[item].append(preds_2)

  model_3 = MultinomialNB()
  model_3.fit(inputs,targets)
  preds_3 = model_3.predict(inputs)
  dic[item].append(preds_3)
  from sklearn.svm import SVC

  # Building a Support Vector Machine on train data
""" model_4 = SVC(C= .1, kernel='linear', gamma= 1)
  model_4.fit(inputs,targets)
  preds_4=model_4.predict(inputs)
  dic[item].append(preds_4)"""
```

```
CPU times: user 15min 10s, sys: 1.34 s, total: 15min 11s
Wall time: 15min 12s
' model_4 = SVC(C= .1, kernel=\'linear\', gamma= 1)\n  model_4.fit(inputs,targets)\n  preds_4=model_4.predict(inputs)\n  dic[item].append(preds_4)'
```

```
from collections import Counter
for item in dic.keys():
  print('\n')
  print(item)
  for pred in dic[item]:
    print(Counter(pred))
  print('\n')
```

```
DoS
Counter({0: 147750, 1: 28100})
Counter({0: 150609, 1: 25241})
Counter({0: 159364, 1: 16486})
Counter({0: 137806, 1: 38044})



XSS
Counter({0: 152850, 1: 23000})
Counter({0: 152455, 1: 23395})
Counter({0: 155382, 1: 20468})
Counter({0: 152037, 1: 23813})



CSRF
Counter({0: 172050, 1: 3800})
Counter({0: 172110, 1: 3740})
Counter({0: 172940, 1: 2910})
```

```
Counter({0: 172008, 1: 3842})



Mem C
Counter({0: 169100, 1: 6750})
Counter({0: 174026, 1: 1824})
Counter({0: 175254, 1: 596})
Counter({0: 158393, 1: 17457})



Dir T
Counter({0: 170250, 1: 5600})
Counter({0: 170534, 1: 5316})
Counter({0: 171312, 1: 4538})
Counter({0: 170197, 1: 5653})



Code Exec
Counter({0: 132800, 1: 43050})
Counter({0: 141580, 1: 34270})
Counter({0: 158465, 1: 17385})
Counter({0: 132723, 1: 43127})



Bypass
Counter({0: 167250, 1: 8600})
```

```
frame=pd.DataFrame(df,columns=['CVE ID Number','Vulnerability Type','Processed Text'])
```

```
frame2=pd.DataFrame(df,columns=['CVE ID Number','Vulnerability Type','Processed Text'])
```

```
#items=['DoS','XSS','CSRF','Mem C','Dir T','Code Exec','Bypass','HTTP','File Inc','OverF','SQLI','Gain Priv','Sec Vul','Gain inf']
```

```
for item in items:
  #frame[item]=df[item]
  print(item)
  for it in dic[item][1:]:
    print(Counter(it)[1])
  ite=max(dic[item][1:],key=lambda x : Counter(x)[1])
  print('ans=',Counter(ite),'\n')
  frame[item]=ite
  frame2[item]=dic[item][3]

#dic['DoS'][3]
```

```
DoS
25241
16486
38044
ans= Counter({0: 137806, 1: 38044})

XSS
23395
20468
23813
ans= Counter({0: 152037, 1: 23813})

CSRF
3740
2910
3842
ans= Counter({0: 172008, 1: 3842})

Mem C
1824
596
17457
ans= Counter({0: 158393, 1: 17457})

Dir T
5316
4538
5653
ans= Counter({0: 170197, 1: 5653})

Code Exec
34270
17385
43127
ans= Counter({0: 132723, 1: 43127})

Bypass
8453
6389
7731
ans= Counter({0: 167397, 1: 8453})

HTTP
124
115
71
ans= Counter({0: 175726, 1: 124})

File Inc
1547
174
4115
ans= Counter({0: 171735, 1: 4115})

OverF
13850
5863
30753
```

```

```

```
def fn(row):
  ans=[]
  for item in items:
    if row[item]==1:
      ans.append(item)
  return ans
```

```
frame['all']=frame.apply(lambda row: fn(row), axis=1)
```

```
frame2['all']=frame2.apply(lambda row: fn(row), axis=1)
```

```
import numpy as np
def fn2(lst):
  return [i for i in new.keys() if new[i] in lst]
def fn3(x):
  return np.array(x)
```

```

```

```
frame['nall']=frame['all'].apply(lambda x: fn2(x))
#final['nall']=final['nall'].apply(lambda x: fn3(x))
frame2['nall']=frame2['all'].apply(lambda x: fn2(x))
```

```
frame.sample(10)
```

	CVE ID Number	Vulnerability Type	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain inf	all	nall
124557	CVE-2015-1264	Cross Site Scripting (XSS)	crosssit script xss vulner googl chrome allow...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	[XSS]	[Cross Site Scripting (XSS)]
7801	CVE-2017-13796	Denial of Service	issu discov certain appl product io affect sa...	1	0	0	1	0	0	0	0	0	1	0	0	0	0	[DoS, Mem C, OverF]	[Denial of Service, Memory Corruption, Overflow]

113426	CVE-2021-38316	Cross Site Scripting (XSS)	wp academ peopl list wordpress plugin vulner r...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	[XSS]	[Cross Site Scripting (XSS)]
13734	CVE-2015-1240	Denial of Service	gpublinkwebgraphicscontextdimplcc webgl implem...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	[DoS]	[Denial of Service]
23271	CVE-2007-5241	Denial of Service	buffer overflow netcsmacdex hp openvm earlier...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	[DoS]	[Denial of Service]
77054	CVE-2018-17230	Overflow	exivuldata typescpp exiv v allow remot attack ...	1	0	0	0	0	0	0	0	0	1	0	0	0	0	[DoS, OverF]	[Denial of Service, Overflow]
28102	CVE-2022-43774	Code Execute	handlerpagepkid class delta electron diaenergi...	0	0	0	0	0	0	0	0	0	0	1	0	0	0	[SQLI]	[SQL Injection]
106040	CVE-2010-2335	SQL Injection	sql inject vulner indexphp yamamah photo galle...	0	0	0	0	0	1	0	0	0	0	1	0	0	0	[Code Exec, SQLI]	[Code Execute, SQL Injection]
77453	CVE-2018-13598	Overflow	minttoken function smart contract implement se...	0	0	0	0	0	0	0	0	0	1	0	0	0	0	[OverF]	[Overflow]
135787	CVE-2018-15782	Directory Traversal	quick setup compon rsa authent manag version p...	0	0	0	0	1	0	0	0	0	0	0	0	0	0	[Dir T]	[Directory Traversal]

```
[ ] cnt=0
    for item in frame['nall']:
      if len(item)==0:
        cnt+=1
    cnt
```

```
22155
```

```
[ ] cnt=0
    for item in frame2['nall']:
      if len(item)==0:
        cnt+=1
    cnt
```

```
24572
```

```
[ ] frame2.sample(10)
```

	CVE ID Number	Vulnerability Type	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain inf	all	nall
143811	CVE-2016-1779	Bypass	webkit appl io safari allow remot attack byp...	0	0	0	0	0	0	1	0	0	0	0	0	0	0	[Bypass]	[Bypass]
62183	CVE-2007-6466	Code Execute	multipl sql inject vulner indexphp freewebshop...	0	0	0	0	0	0	0	0	0	0	1	0	0	0	[SQLI]	[SQL Injection]
116450	CVE-2020-14223	Cross Site Scripting (XSS)	hcl digit experi suscept crosssit script xs...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	[XSS]	[Cross Site Scripting (XSS)]
74445	CVE-2020-22907	Overflow	stack overflow vulner function jsievalcodesub ...	1	0	0	0	0	0	0	0	0	1	0	0	0	0	[DoS, OverF]	[Denial of Service, Overflow]
105262	CVE-2013-3961	SQL Injection	sql inject vulner editeventphp simpl php agend...	0	0	0	0	0	1	0	0	0	0	1	0	0	0	[Code Exec, SQLI]	[Code Execute, SQL Injection]
146892	CVE-2008-0169	Bypass	pluginpasswordauthpm aka passwordauth plugin i...	0	0	0	0	0	0	1	0	0	0	0	0	0	0	[Bypass]	[Bypass]
52365	CVE-2013-3482	Code Execute	stackbas buffer overflow rfreporterror functio...	1	0	0	0	0	0	0	0	0	1	0	0	0	0	[DoS, OverF]	[Denial of Service, Overflow]
47656	CVE-2016-2804	Code Execute	multipl unspecifi vulner browser engin mozilla...	1	0	0	1	0	0	0	0	0	0	0	0	0	0	[DoS, Mem C]	[Denial of Service, Memory Corruption]
55663	CVE-2011-0130	Code Execute	webkit use appl itun window allow maninthemid...	1	0	0	1	0	0	0	0	0	0	0	0	0	0	[DoS, Mem C]	[Denial of Service, Memory Corruption]
160662	CVE-2004-1606	Gain Information	slxwebdll saleslogix allow remot attack caus ...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	[DoS]	[Denial of Service]

```
[ ] frame.sample(5)
```

	CVE ID Number	Vulnerability Type	Processed Text	DoS	XSS	CSRF	Mem C	Dir T	Code Exec	Bypass	HTTP	File Inc	OverF	SQLI	Gain Priv	Sec Vul	Gain inf	all	nall
70061	CVE-2002-1244	Code Execute	format string vulner pablo ftp server possib...	1	0	0	0	0	1	0	0	0	0	0	0	0	0	[DoS, Code Exec]	[Denial of Service, Code Execute]
98912	CVE-2012-3983	Memory Corruption	multipl unspecifi vulner browser engin mozilla...	1	0	0	1	0	0	0	0	0	0	0	0	0	0	[DoS, Mem C]	[Denial of Service, Memory Corruption]
5056	CVE-2019-3804	Denial of Service	found cockpit version use glib base decod fun...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	[DoS]	[Denial of Service]
123778	CVE-2015-9366	Cross Site Scripting (XSS)	custom url track addon ithem exchang wordpres...	0	1	0	0	0	0	0	0	0	0	0	0	0	0	[XSS]	[Cross Site Scripting (XSS)]
24014	CVE-2007-0686	Denial of Service	intel bg wireless minipci driver wnsi allow ...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	[DoS]	[Denial of Service]

```
[ ] final=pd.DataFrame(frame,columns=['CVE ID Number','Vulnerability Type','Processed Text','nall'])
```

```
[ ] final.sample(5)
```

	CVE ID Number	Vulnerability Type	Processed Text	nall
114093	CVE-2021-28380	Cross Site Scripting (XSS)	aimeo aka aimeo shop ecommerc framework extens...	[Cross Site Scripting (XSS)]
50030	CVE-2015-0782	Code Execute	sql inject vulner schedulequeri method schedul...	[Code Execute, SQL Injection]
63527	CVE-2007-2598	Code Execute	sql inject vulner printphp simplenew final al...	[Code Execute, SQL Injection]
119928	CVE-2018-17193	Cross Site Scripting (XSS)	messagepagejsp error page use valu http reques...	[Cross Site Scripting (XSS)]
56779	CVE-2010-2153	Code Execute	unrestrict file upload vulner admincodetcefunc...	[Code Execute, Security Vulnerabilities]

DEEP LEARNING LSTM

```
This is a try to implement lstm without feature engineering and finetuning.
It gave worst results.In the next notebooks works have been made to finetune it.
```

```
[ ] tweet = df['Processed Text'].values
```

```
[ ] from tensorflow.keras.preprocessing.text import Tokenizer

    tokenizer = Tokenizer(num_words=5000)

    tokenizer.fit_on_texts(tweet)
```

```
[ ] from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import LSTM,Dense, Dropout, SpatialDropout1D
    from tensorflow.keras.layers import Embedding


    embedding_vector_length = 32
    model = Sequential()
    vocab_size=len(tokenizer.word_index) + 1
    model.add(Embedding(vocab_size, embedding_vector_length, input_length=200))
    model.add(SpatialDropout1D(0.25))
    model.add(LSTM(50, dropout=0.5, recurrent_dropout=0.5))
    model.add(Dropout(0.2))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy',optimizer='adam', metrics=['accuracy'])

    print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 200, 32)           4452896
```

```
 spatial_dropout1d (SpatialD  (None, 200, 32)          0
 ropout1D)

 lstm (LSTM)                 (None, 50)                16600

 dropout (Dropout)           (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 4,469,547
Trainable params: 4,469,547
Non-trainable params: 0
_________________________________________________________________
None
```

```
[ ] encoded_docs = tokenizer.texts_to_sequences(tweet)
```

```
sentiment_label=df['DoS'].factorize()
sentiment_label

from tensorflow.keras.preprocessing.sequence import pad_sequences

padded_sequence = pad_sequences(encoded_docs, maxlen=200)
```

```
[ ] history = model.fit(padded_sequence,sentiment_label[0], epochs=1, batch_size=10000)
```

```
18/18 [==============================] - 371s 20s/step - loss: 0.1411 - accuracy: 0.9298
```

```
[ ] res=model.predict(padded_sequence)
```

```
5496/5496 [==============================] - 213s 39ms/step
```

```
[ ] res
```

```
array([[0.0479231 ],
       [0.5103224 ],
       [0.0881662 ],
       ...,
       [0.999869  ],
       [0.9998912 ],
       [0.99994564]], dtype=float32)
```

```
[ ] pivot=0.3
```

```
[ ] new_res=[1 if x>=pivot else 0 for x in res]
    print(new_res)
    new_res.count(1),new_res.count(0)
```

```
[0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
(156605, 19245)
```

```
[ ] list(sentiment_label[0]).count(1),list(sentiment_label[0]).count(0)
```

```
(147750, 28100)
```

```
[ ] from sklearn.metrics import accuracy_score
    print(accuracy_score(new_res,sentiment_label[0]))
```

```
0.9316178561273812
```

```
[ ] df1=pd.DataFrame(df,columns=['CVE ID Number','Vulnerability Type','Processed Text','DoS'])
```

```
[ ] df1['DoS1']=list(new_res)
```

```
[ ] df1.sample(5)
```

	CVE ID Number	Vulnerability Type	Processed Text	DoS	DoS1
9148	CVE-2017-5699	Denial of Service	input validation error intel minnowboard firm...	1	0
140762	CVE-2021-3541	Bypass	flaw found libxml exponential entity expansion...	0	0
86180	CVE-2012-3645	Overflow	webkit used apple safari allows remote attack...	0	1
132009	CVE-2006-1911	Cross Site Scripting (XSS)	crosssite scripting xss vulnerability mybb myb...	0	1
144527	CVE-2015-0929	Bypass	timehtm web interface servision hvg video gate...	0	1

```
[ ] df1.head(10)
```

	CVE ID Number	Vulnerability Type	Processed Text	DoS	DoS1
0	CVE-2022-43766	Denial of Service	apache iotdb version vulnerable denial ser...	1	0
1	CVE-2022-43365	Denial of Service	ipcom ew v discovered contain buffer overflow ...	1	1
2	CVE-2022-43035	Denial of Service	issue discovered bento v heapbufferoverflow ap...	1	0
3	CVE-2022-43033	Denial of Service	issue discovered bento bad free component aph...	1	0
4	CVE-2022-42969	Denial of Service	py library python allows remote attackers con...	1	0
5	CVE-2022-42300	Denial of Service	issue discovered veritas netbackup related ve...	1	0
6	CVE-2022-42299	Denial of Service	issue discovered veritas netbackup related ve...	1	0
7	CVE-2022-41715	Denial of Service	programs compile regular expressions untrusted...	1	0
8	CVE-2022-41665	Denial of Service	vulnerability identified sicam p versions v s...	1	1
9	CVE-2022-41556	Denial of Service	resource leak gwbackendc lighttpd could lead...	1	0

```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='acc')
plt.plot(history.history['val_accuracy'], label='val_acc')
plt.legend()
plt.show()

plt.savefig("Accuracy plot.jpg")
```

```
[ ] import nltk
    import pandas as pd
    from textblob import Word
    from nltk.corpus import stopwords
    from sklearn.preprocessing import LabelEncoder
    from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
    from keras.models import Sequential
    from keras.preprocessing.text import Tokenizer
    from keras_preprocessing.sequence import pad_sequences
    from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
    from sklearn.model_selection import train_test_split
    #Model Building
    model = Sequential()
    model.add(Embedding(500, 120, input_length = X.shape[1]))
    model.add(SpatialDropout1D(0.4))
    model.add(LSTM(704, dropout=0.2, recurrent_dropout=0.2))
    model.add(Dense(352, activation='LeakyReLU'))
    model.add(Dense(3, activation='softmax'))
    model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics = ['accuracy'])
    print(model.summary())
    #Model Training
    #model.fit(X,y, epochs = 20, batch_size=32, verbose =1)
    #Model Testing
    #model.evaluate(X,y)
```