

# Oasis Infobyte - OIBSIP - Data Science

## Task 2 : Email Spam Detection Using with Machine Learning

Intern - Shravani Mahesuni

Problem Statement : Each of us has received spam emails at some point. A sort of email known as spam mail, sometimes known as junk mail, is one that is sent to a large number of people at once and typically includes cryptic messages, con games, or—most dangerously—phishing content. Python will be used in this project to create an email spam detector. After that, employ machine learning to teach the spam detector to identify and categorize emails as spam or not spam.

```
In [2]: # Importing Python libraries from SciKit Learn for analyzing models
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, f1_score, recall_score
```

scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines.

```
In [3]: # Loading Dataset / Reading dataset
df = pd.read_csv("spam.csv", encoding="latin1")
df

Out[3]:
```

	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	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
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will i_b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows x 5 columns

```
In [9]: # Print Shape (Get the number of rows and columns)
df.shape

Out[9]: (5169, 5)

In [10]: # get the columns names
df.columns

Out[10]: Index(['v1', 'v2', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')
```

```
In [14]: # Check for number of rows and columns present
print('rows-->', df.shape[0])
print('columns-->', df.shape[1])

rows--> 5169
columns--> 5

In [15]: # Check out the number of null values in the df.
df.isnull().sum()

Out[15]:
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
	0	0	5126	5159	5164
	dtype: int64	dtype: int64	dtype: int64	dtype: int64	dtype: int64

As you can see, there are a significant number of missing entries in the relevant columns Unnamed:2, Unnamed:3, Unnamed:4 more than 99%—so we must get rid of these null values fields.

```
In [17]: #removing those columns which we don't want in calculation
df.drop(columns=df[["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"]], axis=1, inplace=True)

In [18]: print(df)

v1 v2
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...
3 ham U dun say so early hor... U c already then say...
4 ham Nah I don't think he goes to usf, he lives aro...
... ..
5567 spam This is the 2nd time we have tried 2 contact u...
5568 ham Will i_b going to esplanade fr home?
5569 ham Pity, * was in mood for that. So...any other s...
5570 ham The guy did some bitching but I acted like i'd...
5571 ham Rofl. Its true to its name

[5169 rows x 2 columns]

As you can see here that extra columns successfully reduced !

In [19]: # Checking after reduction
df.shape

Out[19]: (5169, 2)
```

```
In [32]: # renaming the column names to make them clear
df.columns = ['spam/ham', 'msg']

In [33]: df.loc[df['spam/ham'] == 'spam', 'spam/ham'] = 0
df.loc[df['spam/ham'] == 'ham', 'spam/ham'] = 1

In [35]: df

Out[35]:
```

	spam/ham	msg
0	1	Go until jurong point, crazy.. Available only ...
1	1	Ok lar... Joking wif u oni...
2	0	Free entry in 2 a wkly comp to win FA Cup fina...
3	1	U dun say so early hor... U c already then say...
4	1	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	0	This is the 2nd time we have tried 2 contact u...
5568	1	Will i_b going to esplanade fr home?
5569	1	Pity, * was in mood for that. So...any other s...
5570	1	The guy did some bitching but I acted like i'd...
5571	1	Rofl. Its true to its name

5169 rows x 2 columns

```
In [36]: x = df.msg
x

Out[36]:
```

0 Go until jurong point, crazy.. Available only ...  
1 Ok lar... Joking wif u oni...  
2 Free entry in 2 a wkly comp to win FA Cup fina...  
3 U dun say so early hor... U c already then say...  
4 Nah I don't think he goes to usf, he lives aro...  
... ..  
5567 This is the 2nd time we have tried 2 contact u...  
5568 Will i\_b going to esplanade fr home?  
5569 Pity, \* was in mood for that. So...any other s...  
5570 The guy did some bitching but I acted like i'd...  
5571 Rofl. Its true to its name  
Name: msg, Length: 5169, dtype: object

```
In [37]: y = df['spam/ham']
y

Out[37]:
```

0 1  
1 1  
2 0  
3 1  
4 1  
... ..  
5567 0  
5568 1  
5569 1  
5570 1  
5571 1  
Name: spam/ham, Length: 5169, dtype: object

```
In [38]: #Devide the whole dataset into training and testing set for model training
from sklearn.model_selection import train_test_split

In [39]: xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=3)
```

```
In [40]: print(x.shape)
print(xtrain.shape)
print(xtest.shape)

(5169,)
(4135,)
(1034,)
```

```
In [41]: xtrain, xtest

Out[41]:
```

(4443, 1) COME BACK TO TAMPA FFFUUUUUUU  
982 Congrats! 2 mobile 3G Videophones R yours. cal...  
3822 Please protect yourself from e-threats. SIB ne...  
3924 As if i wasn't having enough trouble sleeping.  
4927 Just hoping that wasn't too pissed up to re...  
... ..  
806 sure, but make sure he knows we ain't smokin yet  
990 26th OF JULY  
1723 Hi Jon, Pete here, I've bin 2 Spain recently & ...  
3519 No it will reach by 9 only. She telling she wi...  
1745 Iâ cool ta luv but v.tired 2 cause i have be...  
Name: msg, Length: 4135, dtype: object  
4994 Just looked it up and addie goes back Monday, ...  
4292 You best watch what you say cause I get drunk ...  
4128 We i'm not workin. Once i get job  
4429 Yar lor... How u noe? U used dat route too?  
660 Under the sea, there lays a rock. In the rock,...  
... ..  
4003 Well there's a pattern emerging of my friends ...  
1107 From someone not to smoke when every time I've...  
5413 Nite nite pocay wocay luv u more than n e thin...  
1413 Dear U've been invited to XCHAT. This is our f...  
4998 Hmph. Go head, big baller.  
Name: msg, Length: 1034, dtype: object)

```
In [42]: ytrain, ytest

Out[42]:
```

(4443, 1)  
982 0  
3822 1  
3924 1  
4927 1  
... ..  
806 1  
990 1  
1723 1  
3519 1  
1745 1  
Name: spam/ham, Length: 4135, dtype: object  
4994 1  
4292 1  
4128 1  
4429 1  
660 1  
... ..  
4003 1  
1107 1  
5413 1  
1413 0  
4998 1  
Name: spam/ham, Length: 1034, dtype: object)

We must convert all of the text input into numbers since, as far as we are aware, machine learning algorithms only work well with respect to numbers. I'll utilize the TfidfVectorizer method from Sklearn's feature\_extraction section to accomplish this.

```
In [43]: feat_vect=TfidfVectorizer(min_df=1, stop_words='english', lowercase=True)
feat_vect

Out[43]:
```

TfidfVectorizer  
TfidfVectorizer(stop\_words='english')

```
In [44]: ytrain=ytrain.astype('int')
ytest=ytest.astype('int')
```

```
In [45]: xtrain_vec = feat_vect.fit_transform(xtrain)
```

```
In [47]: xtest_vec = feat_vect.transform(xtest)
```

```
In [51]: print(xtrain)

4443 COME BACK TO TAMPA FFFUUUUUUU
982 Congrats! 2 mobile 3G Videophones R yours. cal...
3822 Please protect yourself from e-threats. SIB ne...
3924 As if i wasn't having enough trouble sleeping.
4927 Just hoping that wasn't too pissed up to re...
... ..
806 sure, but make sure he knows we ain't smokin yet
990 26th OF JULY
1723 Hi Jon, Pete here, I've bin 2 Spain recently & ...
3519 No it will reach by 9 only. She telling she wi...
1745 Iâ cool ta luv but v.tired 2 cause i have be...
Name: msg, Length: 4135, dtype: object
```

```
In [53]: xtrain_vec

Out[53]: <4135x7378 sparse matrix of type '<class 'numpy.float64'>'
with 31488 stored elements in Compressed Sparse Row format>
```

```
In [54]: #printing x train data
print(xtrain_vec)

(0, 2697) 0.7285755344386542
(0, 6409) 0.5958532917415522
(0, 1825) 0.35592482233751443
(1, 5438) 0.27399320458839144
(1, 4583) 0.27399320458839144
(1, 4438) 0.22516921191243092
(1, 5036) 0.27399320458839144
(1, 2274) 0.27399320458839144
(1, 2920) 0.23390504161994488
(1, 3618) 0.27399320458839144
(1, 4984) 0.19732502227978832
(1, 4180) 0.23390504161994488
(1, 7137) 0.24133495616477563
(1, 6940) 0.27399320458839144
(1, 283) 0.27399320458839144
(1, 6941) 0.27399320458839144
(1, 453) 0.25698446420786897
(1, 4333) 0.15929789793058355
(1, 1805) 0.22516921191243092
(2, 953) 0.26160275768603725
(2, 4856) 0.26160275768603725
(2, 5786) 0.26160275768603725
(2, 2459) 0.22436535516409714
(2, 4960) 0.26160275768603725
(2, 5976) 0.1902832473629628
:
(4132, 6862) 0.110853927369947865
(4132, 5612) 0.14854399893836668
(4132, 3865) 0.16898098428277844
(4133, 6457) 0.61541778208806059
(4133, 5320) 0.5530764956488926
(4133, 2311) 0.4238274689992768
(4133, 3771) 0.36842584696755415
(4134, 4632) 0.2852228597337175
(4134, 3508) 0.2852228597337175
(4134, 5982) 0.2623257437582278
(4134, 3585) 0.2718289051333927
(4134, 6095) 0.2623257437582278
(4134, 4908) 0.24893178953790301
(4134, 1608) 0.2289889687279293
(4134, 5711) 0.21865344863808088
(4134, 6383) 0.24893178953790301
(4134, 3967) 0.23205740285833368
(4134, 2297) 0.23205740285833368
(4134, 6596) 0.235378353175782
(4134, 5998) 0.25495454185338234
(4134, 7181) 0.20454345297905668
(4134, 1923) 0.19363439583175862
(4134, 1571) 0.18164833709350694
(4134, 4068) 0.20454345297905668
(4134, 3038) 0.13885722635220862
```

```
In [55]: xtest_vec

Out[55]: <1034x7378 sparse matrix of type '<class 'numpy.float64'>'
with 7012 stored elements in Compressed Sparse Row format>
```

```
In [56]: #printing x test data
print(xtest_vec)

(0, 6284) 0.43430701953205156
(0, 4357) 0.4264504812056483
(0, 3999) 0.45410391508126108
(0, 3685) 0.21875536593912145
(0, 3088) 0.3755589393427584
(0, 796) 0.4841597716958077
(1, 7050) 0.41978523395044104
(1, 5656) 0.3549971211138654
(1, 2369) 0.5017364019285492
(1, 1608) 0.47304204171951914
(1, 1254) 0.398046282326562
(2, 7221) 0.7923997102028898
(2, 3640) 0.6100022125126892
(3, 7292) 0.4854329061592562
(3, 6879) 0.504261162123145
(3, 4574) 0.45819212318042857
(3, 4009) 0.3213876208979908
(3, 2080) 0.43567694225913534
(4, 7218) 0.2292199402753507
(4, 5680) 0.23604783770184097
(4, 5540) 0.5112231722851113
(4, 4823) 0.597985350067042
(4, 2508) 0.5052580430200418
(5, 1132) 0.6794018382139002
:
(1031, 5191) 0.3701308998039074
(1031, 4585) 0.6439113570053595
(1031, 4068) 0.3041309124908008
(1031, 510) 0.62409124971092943
(1032, 7339) 0.2632164916232764
(1032, 7271) 0.20958482923756
(1032, 6907) 0.2578213890193124
(1032, 691) 0.17642080640756436
(1032, 6764) 0.16357649120381332
(1032, 6298) 0.24905393871920117
(1032, 5583) 0.236096235154502
(1032, 4406) 0.2773798445273517
(1032, 3848) 0.25783213890193124
(1032, 3538) 0.24905393871920117
(1032, 2720) 0.23900544751939748
(1032, 2009) 0.17318001901901607
(1032, 1899) 0.1877453530769227
(1032, 1608) 0.19878732825316712
(1032, 1971) 0.21945854189397707
(1032, 674) 0.24537378511987706
(1032, 394) 0.24905393871920117
(1032, 316) 0.2016158353905777
(1032, 302) 0.1936225393041707
(1033, 3198) 0.730768408492214
(1033, 1272) 0.6794018382139002
```

```
In [57]: log1 = LogisticRegression()

In [58]: log1.fit(xtrain_vec, ytrain)
```

```
Out[58]: LogisticRegression
LogisticRegression()

In [59]: log1.score(xtrain_vec, ytrain)

Out[59]: 0.962273278904474
```

```
In [60]: log1.score(xtest_vec, ytest)

Out[60]: 0.960348162475822
```

```
In [61]: pred_log1 = log1.predict(xtest_vec)
pred_log1

Out[61]: array([1, 1, 1, ..., 1, 0, 1])
```

```
In [62]: from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

In [63]: accuracy_score(ytest, pred_log1)

Out[63]: 0.960348162475822
```

```
In [64]: confusion_matrix(ytest, pred_log1)

Out[64]: array([[ 99, 41],
[ 0, 894]], dtype=int64)
```

```
In [65]: print(classification_report(ytest, pred_log1))

precision recall f1-score support

0 1.00 0.71 0.83 140
1 0.96 1.00 0.98 894

accuracy 0.98 0.05 0.96 1034
macro avg 0.98 0.96 0.96 1034
weighted avg 0.96 0.96 0.96 1034
```

### Final Classification Report analysis

```
In [1]:
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[1], line 1
----> 1 df.info()

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

```
In [2]:
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[2], line 1
----> 1 df.info()

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

```
In [ ]:
```