Spam Detection with Machine Learning

Detecting spam alerts in emails and messages is one of the main applications that every big tech company tries to improve for its customers. Apple's official messaging app and Google's Gmail are great examples of such applications where spam detection works well to protect users from spam alerts. So, if you are looking to build a spam detection system, this article is for you. In this article, I will walk you through the task of Spam Detection with Machine Learning using Python.

Spam Detection

Whenever you submit details about your email or contact number on any platform, it has become easy for those platforms to market their products by advertising them by sending emails or by sending messages directly to your contact number. This results in lots of spam alerts and notifications in your inbox. This is where the task of spam detection comes in.

Spam detection means detecting spam messages or emails by understanding text content so that you can only receive notifications about messages or emails that are very important to you. If spam messages are found, they are automatically transferred to a spam folder and you are never notified of such alerts. This helps to improve the user experience, as many spam alerts can bother many users.

Spam Detection using Python

Hope you now understand what spam detection is, now let's see how to train a machine learning model for detecting spam alerts using Python. I'll start this task by importing the necessary Python libraries and the dataset you need for this task

In [170…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import cv2

%matplotlib inline
```

Data Collection

In [171…

```
data = pd.read_csv(r".\\data\spam_messege.csv", encoding="latin-1")
data.head()
```

Out[171…

	class	message	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

In [172…

```
print(data.head())
```

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

  Unnamed: 3 Unnamed: 4
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3        NaN        NaN
4        NaN        NaN
```

Data Pre-Processing

In [173…

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
```

```
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   class       5572 non-null   object
 1   message     5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

In [174…]
```
data.describe()
```

Out[174…]

	class	message	Unnamed: 2	Unnamed: 3	Unnamed: 4
count	5572	5572	50	12	6
unique	2	5169	43	10	5
top	ham	Sorry, I'll call later	bt not his girlfrnd... G o o d n i g h t . . .@"	MK17 92H. 450Ppw 16"	GNT:-)"
freq	4825	30	3	2	2

In [175…]
```
print(data.describe())
```

```
       class                 message  \
count   5572                    5572
unique     2                    5169
top      ham  Sorry, I'll call later
freq    4825                      30

                                        Unnamed: 2  \
count                                           50
unique                                          43
top      bt not his girlfrnd... G o o d n i g h t . . .@"
freq                                             3

                  Unnamed: 3 Unnamed: 4
count                     12          6
unique                    10          5
top      MK17 92H. 450Ppw 16"    GNT:-)"
freq                      2          2
```

In [176…]
```
data.columns
```

Out[176…]
```
Index(['class', 'message', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')
```

From this dataset, class and message are the only features we need to train a machine learning model for spam detection, so let's select these two columns as the new dataset

In [177…]
```
data = data[["class", "message"]]
```

In [178…]
```
data[["class"]].value_counts()
```

Out[178…]
```
class
ham      4825
spam      747
dtype: int64
```

In [179…]
```
data[["message"]].value_counts().head(5)
```

Out[179…]
```
message
Sorry, I'll call later
30
I cant pick the phone right now. Pls send a message
12
Ok...
10
Your opinion about me? 1. Over 2. Jada 3. Kusruthi 4. Lovable 5. Silent 6. Spl character 7. Not matured 8. Stylish 9. Simple Pls reply..
4
Wen ur lovable bcums angry wid u, dnt take it seriously.. Coz being angry is d most childish n true way of showing deep affection, care n luv!.. ketto da manda... Have nice day da.     4
dtype: int64
```

Feature Selection

In [180…]
```
feature = np.array(data["message"])
target = np.array(data["class"])
```

In [181…]
```
from sklearn.feature_extraction.text import CountVectorizer
fem = CountVectorizer()
feature = fem.fit_transform(feature)
```

Spliting Data

In [182…]
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(feature, target, test_size=0.33,random_state=42)
```

In [183…]
```
xtest.shape, xtrain.shape
```

Out[183…]
```
((1839, 8710), (3733, 8710))
```

In [184…]
```
ytest.shape, ytrain.shape
```

Out[184… ((1839,), (3733,))

Choosing Model & Training The Model

In [185…
```
from sklearn.naive_bayes import MultinomialNB
nbm = MultinomialNB()
nbm.fit(xtrain, ytrain.ravel())
```

Out[185… MultinomialNB()

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Predicting Test Data

In [186…
```
predictions = nbm.predict(xtest)
```

In [187…
```
predictions
```

Out[187… array(['spam', 'ham', 'spam', ..., 'ham', 'ham', 'spam'], dtype='<U4')

Now let's test this model by taking a user input as a message to detect whether it is spam or not

Enter a message: You won $40 cash price

In [188…
```
sample = input('Enter a message:')
data = fem.transform([sample]).toarray()
print(nbm.predict(data))
```

['spam']

Summary

So this is how you can train a machine learning model for the task of detecting whether an email or a message is spam or not. A Spam detector detects spam messages or emails by understanding text content so that you can only receive notifications about messages or emails that are very important to you. I hope you liked this article on the task of detecting spam alerts with machine learning using Python. Feel free to ask your valuable questions in the comments section below.

Sheikh Rasel Ahmed

Data Science || Machine Learning || Deep Learning || Artificial Intelligence Enthusiast

In [190…
```
# LinkedIn - https://www.linkedin.com/in/shekhnirob1

# GitHub - https://github.com/Rasel1435

# Behance - https://www.behance.net/Shekhrasel2513
```