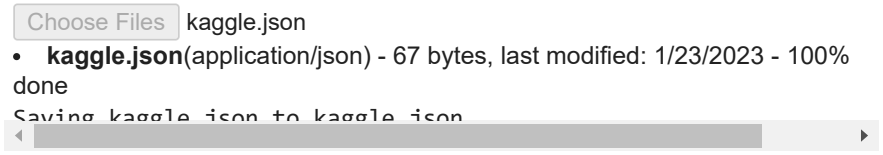
Lab - Spam Text Message Classification using Naive Bayes Classifier.

```
! pip install -q kaggle
```

```
from google.colab import files

files.upload()
```




```
! mkdir ~/.kaggle

! cp kaggle.json ~/.kaggle/
```

```
! chmod 600 ~/.kaggle/kaggle.json
```

```
! kaggle datasets download -d team-ai/spam-text-message-classification
```

```
Downloading spam-text-message-classification.zip to /content
  0% 0.00/208k [00:00<?, ?B/s]
100% 208k/208k [00:00<00:00, 28.3MB/s]
```

```
! mkdir Spam-Text-Classification
```

```
! unzip /content/spam-text-message-classification.zip -d /content/Spam-Text-Classification
```

```
Archive:  /content/spam-text-message-classification.zip
  inflating: /content/Spam-Text-Classification/SPAM text message 20170820 - Data.csv
```

Importing Important Libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
from sklearn.metrics import confusion_matrix, accuracy_score
```

Load The Downloaded Data.

This data has two columns named "Message" and "Category".
The "Category" column has non-spam (ham) and spam labels for each message.

```
data = pd.read_csv('/content/Spam-Text-Classification/SPAM text message 20170820 - Data.csv')
data.head()
```

	Category	Message
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...

```
data.shape
```

```
(5572, 2)
```

```
data['Message'] = data['Message'].str.lower() # Lower case all messages
```

```
data.head()
```

	Category	Message
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...

```
data['Message'] = data['Message'].str.replace('[^\w\s]', ' ', regex=True)  # Remove punctutions
```

```
data.head()
```

	Category	Message
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...

Here I perform the training and the testing. Then I build a pipeline and lastly, evaluate the results.

```
y = data['Category']
X = data['Message']
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

```
# Build model pipeline
model = make_pipeline(TfidfVectorizer(), MultinomialNB())
# Train the model using training data
model.fit(X_train, y_train)
# Predict the classes of test data
predicted_categories = model.predict(X_test)
```

```
# Accuracy score of the model
print("The model accuracy is:", accuracy_score(y_test, predicted_categories))
```

```
The model accuracy is: 0.9596412556053812
```

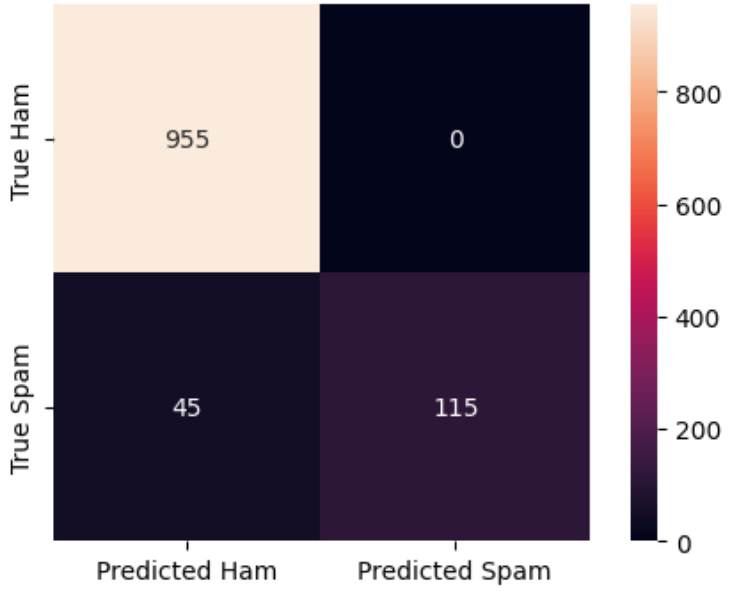
```
conf_matrix = confusion_matrix(y_test, predicted_categories)
```

```
plt.rcParams['figure.dpi']= 100

# Plot/draw confusion matrix

sns.heatmap(conf_matrix,square=True, annot=True, fmt="d", yticklabels=["True Ham", "True Spam"],
    xticklabels=["Predicted Ham", "Predicted Spam"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7de62e4d90>
```



Observations:

Analysis of the results:

- 955 Non-Spam (ham) mails have been correctly classified
- 115 Spam mails have been correctly classified
- 0 Non-Spam (ham) mails have been classified as Spam mails (False Positives or Type I Error)
- 45 Spam mails have been classified as Non-Spam (False Negatives or Type II Error)

The model **accuracy is 96% (rounded 0.9596412556053812)**, which is impressive for this problem.

Here I perform testing on some examples messages.

Example of a Spam message:

```
# Testing examples

# Spam Example
my_sentence = "WINNER!! As a valued network customer you have been selected to receivea £900 prize reward
p = model.predict([my_sentence])
print('Spam message' if p=='spam' else 'Ham/non-spam message')
```

```
Spam message
```

Example of a Non-spam / ham message:

```
# Non-spam example
my_sentence = "Nah I don't think he goes to usf, he lives around here though"
p = model.predict([my_sentence])
print('Spam message' if p=='spam' else 'Ham/non-spam message')
```

```
Ham/non-spam message
```

References Used:

https://www.kaggle.com/datasets/team-ai/spam-text-message-classification

https://colab.research.google.com/github/alvinntnu/python-notes/blob/master/nlp/naive-bayes.ipynb#scrollTo=EsveAy4T3flz