```
#importing required packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
import string
from nltk.tokenize import word_tokenize
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
#dataset: https://www.kaggle.com/karthickveerakumar/spam-filter
emails = pd.read_csv('emails.csv')
emails
```

	text	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1
...	...	...
5723	Subject: re : research and development charges...	0
5724	Subject: re : receipts from visit jim , than...	0
5725	Subject: re : enron case study update wow ! a...	0
5726	Subject: re : interest david , please , call...	0
5727	Subject: news : aurora 5 . 2 update aurora ve...	0

5728 rows × 2 columns

```
emails.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5728 entries, 0 to 5727
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   text    5728 non-null   object
 1   spam    5728 non-null   int64
```

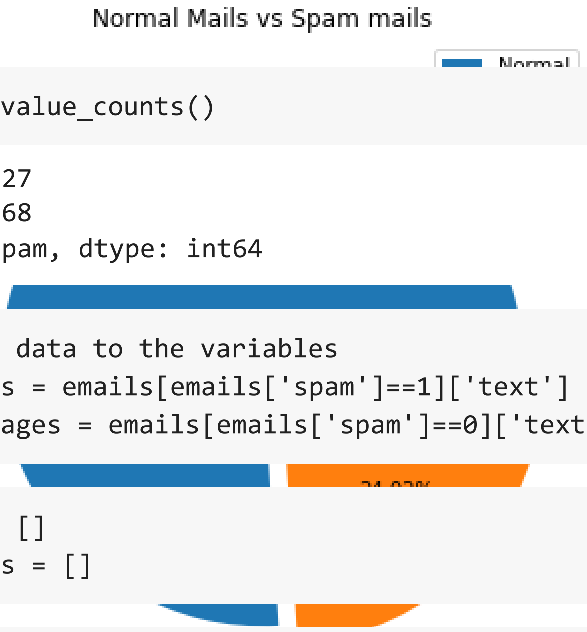
```
dtypes: int64(1), object(1)
memory usage: 89.6+ KB
```

```
emails = emails.drop_duplicates(keep = 'last') #remove all duplicate emails from the dataf
emails
```

	text	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1
...	...	...
5723	Subject: re : research and development charges...	0
5724	Subject: re : receipts from visit jim , than...	0
5725	Subject: re : enron case study update wow ! a...	0
5726	Subject: re : interest david , please , call...	0
5727	Subject: news : aurora 5 . 2 update aurora ve...	0

5695 rows × 2 columns

```
#data visualization using matplotlib
emails.spam.value_counts().plot(kind='pie',
                                    explode=[0,.1],
                                    figsize=(6,6),
                                    autopct='%.2f%%')
plt.title('Normal Mails vs Spam mails')
plt.legend(['Normal','Spam'])
plt.show()
```



```
emails.spam.value_counts()
```

```
0    4327
1    1368
Name: spam, dtype: int64
```

```
# allocating data to the variables
spam_messages = emails[emails['spam']==1]['text']
notspam_messages = emails[emails['spam']==0]['text']
```

```
spam_words = []
notspam_words = []
```

```
#creating a function for tokenizing the text using nltk
def tokenize_spam_words(text):
    words = [w.lower() for w in word_tokenize(text) if w.lower() not in stopwords.words('e
    spam_words.extend(words)

def tokenize_notspam_words(text):
    words = [w.lower() for w in word_tokenize(text) if w.lower() not in stopwords.words('e
    notspam_words.extend(words)
```

```
#tokenizing the spam messages
spam_messages.apply(tokenize_spam_words)
print(spam_words[:100])
```

```
['subject', 'naturally', 'irresistible', 'corporate', 'identity', 'lt', 'really', 'ha
```

```
#tokenizing the not spam messages
notspam_messages.apply(tokenize_notspam_words)
print(notspam_words[:100])
```

```
['subject', 'hello', 'guys', 'bugging', 'completed', 'questionnaire', 'one', 'page',
```

```
#stemming
from nltk.stem import PorterStemmer
```

```
stemmer = PorterStemmer()
```

```
# creating a function for stemming the words
def cleanup_text(message):
    message = message.translate(str.maketrans('','',string.punctuation))
    words = [stemmer.stem(w) for w in message.split() if w.lower() not in stopwords.words(
    return ' '.join(words)
```

```
emails.text = emails.text.apply(cleanup_text)
```

```
emails.head()
```

	text	spam
0	subject natur irresist corpor ident lt realli ...	1
1	subject stock trade gunsling fanni merril muzo...	1
2	subject unbeliev new home made easi im want sh...	1
3	subject 4 color print special request addit in...	1
4	subject money get softwar cd softwar compat gr...	1

```
#feautre extraction using count vectorizer
from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer(stop_words = 'english')
```

```
features = vect.fit_transform(emails.text)
features.shape
```

```
(5695, 29096)
```

```
# saving the feautures using the pickle
import pickle

with open('count_vectorizer.pkl','wb') as f:
    pickle.dump(vect,f)
print('done')
```

```
done
```

```
# data preprocessing for training the model
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
```

```
#labeling the data
category =  LabelEncoder()
emails.spam = category.fit_transform(emails.spam)
```

```
emails.head()
```

	text	spam
0	subject natur irresist corpor ident lt realli ...	1

```
#splitting the data into training and testing data
x_train, x_test, y_train,y_test = train_test_split(features.toarray(), emails.spam,test_si
```

2 subject 4 color print special request addit in ... 1

```
#creating a machine learning model
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
```

```
model = GaussianNB()
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
```

```
#confusion matrix
confusion_matrix(y_test,y_pred)
```

```
array([[881,  15],
       [ 16, 227]])
```

```
#saving the builded model using pickle
import pickle
with open('spam_classifier.pkl','wb') as f:
    pickle.dump(model,f)
print('done')
```

```
done
```