```
import pandas as pd

df = pd.read_csv("spamsms.csv")

df.head(3)
```

```
  Class                                               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...
```

```
df.shape
```

```
(5574, 2)
```

```
(df["Class"].value_counts())
```

```
Class
ham     4827
spam     747
Name: count, dtype: int64
```

```
df.isnull().sum()
```

```
Class      0
Message    0
dtype: int64
```

```
df.info
```

```
<bound method DataFrame.info of      Class
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...
...    ...                                                ...
5569  spam  This is the 2nd time we have tried 2 contact u...
5570   ham                Will ü b going to esplanade fr home?
5571   ham  Pity, * was in mood for that. So...any other s...
5572   ham  The guy did some bitching but I acted like i'd...
5573   ham                         Rofl. Its true to its name

[5574 rows x 2 columns]>
```

TEXT PREPROCESSING (clean and transform the actual data to a formal for ml)

```
df["Message"] = df["Message"].str.lower()
```

REMOVING PUNCTUATION

```
df["Messsage"] = df["Message"].str.replace(r'[^\w\s]','',regex = True)

import nltk

from nltk.corpus import stopwords

stop = stopwords.words("english")

df["Message"] = df["Message"].apply(lambda x: ' ' .join([word for word
in x.split() if word not in (stop)]))

df
```

```
     Class                                                Message  \
0      ham  go jurong point, crazy.. available bugis n gre...
1      ham                      ok lar... joking wif u oni...
2     spam  free entry 2 wkly comp win fa cup final tkts 2...
3      ham          u dun say early hor... u c already say...
4      ham            nah think goes usf, lives around though
...    ...                                                ...
5569  spam  2nd time tried 2 contact u. u £750 pound prize...
5570   ham                       ü b going esplanade fr home?
5571   ham           pity, * mood that. so...any suggestions?
5572   ham  guy bitching acted like i'd interested buying ...
5573   ham                                 rofl. true name

                                              Messsage
0     go until jurong point crazy available only in ...
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 dont think he goes to usf he lives aroun...
...                                                 ...
5569  this is the 2nd time we have tried 2 contact u...
5570               will ü b going to esplanade fr home
5571  pity  was in mood for that soany other suggest...
5572  the guy did some bitching but i acted like id ...
5573                        rofl its true to its name

[5574 rows x 3 columns]
```

STEMMING

```
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()

df["Message"] = df["Message"].apply(lambda x: '
' .join([stemmer.stem(word) for word in x.split()]))
```

LEMMATIZATION

```
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

nltk.download("wordnet")
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\PC\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
True
```

```
df["Message"] = df["Message"].apply(lambda x: '
' .join([lemmatizer.lemmatize(word) for word in x.split()]))

ham = df[df['Class'] == 'ham' ]

spam = df[df['Class'] == 'spam' ]
```

Undersample the majority class (ham)

```
ham_sampled = ham.sample(len(spam), random_state = 42)

balanced_df = pd.concat([ham_sampled,spam])

balanced_df = balanced_df.sample(frac = 1 ,
random_state=42).reset_index(drop=True)
```

FEATURE EXTRACTION (converting text to numerical data)

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()

X = vectorizer.fit_transform(balanced_df['Message'])

y = balanced_df['Class'].apply(lambda x:1 if x == 'spam' else 0)
```

SPLITTING THE DATA

```
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 = 42)
```

CHECKING DATA SHAPE AFTER UNDERSAMPLING

```
print("Shape of X_train after undersampling:", X_train.shape)
print("Shape of y_train after undersampling:", y_train.shape)
```

```
Shape of X_train after undersampling: (1195, 4281)
Shape of y_train after undersampling: (1195,)

print("Class distribution in the training set after undersampling:")
print(pd.Series(y_train).value_counts())

Class distribution in the training set after undersampling:
Class
1    603
0    592
Name: count, dtype: int64
```

IMPORT NAIVE BAYES CLASSIFIER

```
from sklearn.naive_bayes import MultinomialNB

nb_model = MultinomialNB()

nb_model.fit(X_train,y_train)

MultinomialNB()

y_pred = nb_model.predict(X_test)
```

CHECK MODELS ACCURACY

```
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test , y_pred)

(f"Accuracy:{accuracy * 100:.2f}%")

'Accuracy:95.99%'

nb_model.score(X_test,y_test)*100

95.9866220735786
```

CLASSIFICATION PERFORMANCE OVERVIEW

```
from sklearn.metrics import classification_report

(classification_report(y_test , y_pred))

'              precision    recall  f1-score   support\n\n           0
0.96      0.97      0.96       155\n           1       0.96      0.95
0.96       144\n\n    accuracy                           0.96
299\n   macro avg       0.96      0.96      0.96       299\nweighted
avg       0.96      0.96      0.96       299\n'
```

TESTING WITH NEW MAILS

```
new_email = "Dear Valued Customer, Congratulations! You have been
selected as the winner of a **$1000 gift card**. This is an exclusive
offer just for you! To claim your prize, simply click the link below
and follow the instructions. Claim your gift now! Please note, this
offer is available for a limited time only, so act fast! Don’t miss
out on this incredible opportunity.Best regards"
new_email2 = " Jawad, your profile photo was changed"

new_email_vectorized = vectorizer.transform([new_email,new_email2])

prediction = nb_model.predict(new_email_vectorized)

for email, prediction in zip([new_email, new_email2], prediction):
    print(f"Email: {email}\nPrediction: {'Spam' if prediction == 1
else 'Not Spam'}\n")
```

```
Email: Dear Valued Customer, Congratulations! You have been selected
as the winner of a **$1000 gift card**. This is an exclusive offer
just for you! To claim your prize, simply click the link below and
follow the instructions. Claim your gift now! Please note, this offer
is available for a limited time only, so act fast! Don’t miss out on
this incredible opportunity.Best regards
Prediction: Spam

Email:  Jawad, your profile photo was changed
Prediction: Not Spam
```