**Spam/Ham Classification using Naïve Bayes**

import numpy as np

import pandas as pd

emails = pd.read\_csv('./emails.csv')

#emails[:10]

def process\_email(text):

text = text.lower()

return list(set(text.split()))

emails['words'] = emails['text'].apply(process\_email)

num\_emails = len(emails)

num\_spam = sum(emails['spam'])

print("Number of emails:", num\_emails)

print("Number of spam emails:", num\_spam)

print()

# Calculating the prior probability that an email is spam

print("Probability of spam:", num\_spam/num\_emails)

print()

model = {}

# Training process

for index, email in emails.iterrows():

for word in email['words']:

if word not in model:

model[word] = {'spam': 1, 'ham': 1}

if word in model:

if email['spam']:

model[word]['spam'] += 1

else:

model[word]['ham'] += 1

def predict\_bayes(word):

word = word.lower()

num\_spam\_with\_word = model[word]['spam']

num\_ham\_with\_word = model[word]['ham']

return 1.0\*num\_spam\_with\_word/(num\_spam\_with\_word + num\_ham\_with\_word)

print("Prediction using Bayes for word sale",predict\_bayes("sale"))

print("Prediction using Bayes for word lottery",predict\_bayes("lottery"))

print()

def predict\_naive\_bayes(email):

total = len(emails)

num\_spam = sum(emails['spam'])

num\_ham = total - num\_spam

email = email.lower()

words = set(email.split())

spams = [1.0]

hams = [1.0]

for word in words:

if word in model:

spams.append(model[word]['spam']/num\_spam\*total)

hams.append(model[word]['ham']/num\_ham\*total)

prod\_spams = np.compat.long(np.prod(spams)\*num\_spam)

prod\_hams = np.compat.long(np.prod(hams)\*num\_ham)

return prod\_spams/(prod\_spams + prod\_hams)

print("Prediction using NaiveBayes for word lottery sale",predict\_naive\_bayes("lottery sale"))

print("Prediction using NaiveBayes for word asdfgh",predict\_naive\_bayes("asdfgh"))

print("Prediction using NaiveBayes ",predict\_naive\_bayes('Hi mom how are you'))

**OUTPUT:**

**Number of emails: 5728**

**Number of spam emails: 1368**

**Probability of spam: 0.2388268156424581**

**Prediction using Bayes for word sale 0.48148148148148145**

**Prediction using Bayes for word lottery 0.9**

**Prediction using NaiveBayes for word lottery sale 0.9638144992048691**

**Prediction using NaiveBayes for word asdfgh 0.2388268156424581**

**Prediction using NaiveBayes 0.12554358867164464**