EMAIL SPAM FILTERING

**import** pandas **as** pd

*# Load the dataset (make sure to provide the correct path)*

df **=** pd**.**read\_csv("SMSSpamCollection", sep**=**'\t',names**=**['label','message'])

*# Show the first few rows of the dataset*

print(df**.**head())

*# Label encoding: spam=1, ham=0*

df['label']**=**df['label']**.**map({'spam':1,'ham':0})

*# train test*

**from** sklearn.model\_selection **import** train\_test\_split

X\_train,X\_test,y\_train, y\_test**=** train\_test\_split(df['message'], df['label'], test\_size**=**0.2, random\_state**=** 42)

*# text preprocessing*

**import** re

**def** preprocessing\_text(text):

text **=**text**.**lower()

text **=** re**.**sub(r'\d+','',text) *#remove numbers*

text **=** re**.**sub(r'\W+','',text) *#remove punctuations*

**return** text

*# apply preprocessing*

X\_train**=**X\_train**.**apply(preprocessing\_text)

X\_test**=**X\_test**.**apply(preprocessing\_text)

*#Tf\_IDF vectorization*

**from** sklearn.feature\_extraction.text **import** TfidfVectorizer

vectorizer**=**TfidfVectorizer(stop\_words**=**'english',max\_features**=**3000)

X\_train\_vec**=**vectorizer**.**fit\_transform(X\_train)

X\_test\_vec**=**vectorizer**.**transform(X\_test)

*# train neive bayes classifire*

**from** sklearn.naive\_bayes **import** MultinomialNB

classifier **=** MultinomialNB()

classifier**.**fit(X\_train\_vec,y\_train)

*# predict on test\_data*

y\_pred**=** classifier**.**predict(X\_test\_vec)

**from** sklearn.metrics **import** classification\_report

print(classification\_report(y\_test,y\_pred))

print(classification\_report(y\_test, y\_pred, target\_names**=**['ham', 'spam']))