import pandas as pd

data = pd.read\_csv('https://raw.githubusercontent.com/mohitgupta-omg/Kaggle-SMS-Spam-Collection-Dataset-/master/spam.csv', encoding='latin-1')

data.head()

data.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)

data.columns = ['label', 'text']

data.head()

**Exploratory Data Analysis (EDA)**

Let’s do some basic EDA to see if there are missing values in the dataset and what’s the target balance.

# check missing values

data.isna().sum()

data.shape

data['label'].value\_counts(normalize = True).plot.bar()

**Text Preprocessing**

This is where all text cleaning takes place. It’s a loop that iterates through all 5,572 documents and does the following:

* Remove all special characters
* Lowercase all the words
* Tokenize
* Remove stopwords
* Lemmatize

# text preprocessing

# download nltk

import nltk

nltk.download('all')

# create a list text

text = list(data['text'])

# preprocessing loop

import re

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

corpus = []

for i in range(len(text)):

r = re.sub('[^a-zA-Z]', ' ', text[i])

r = r.lower()

r = r.split()

r = [word for word in r if word not in stopwords.words('english')]

r = [lemmatizer.lemmatize(word) for word in r]

r = ' '.join(r)

corpus.append(r)

#assign corpus to data['text']

data['text'] = corpus

data.head()

**Train-test-split**

Let’s split the dataset into train and test before feature extraction.

# Create Feature and Label sets

X = data['text']

y = data['label']

# train test split (66% train - 33% test)

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=123)

print('Training Data :', X\_train.shape)

print('Testing Data : ', X\_test.shape)

**Feature Extraction**

Here, we use the Bag of Words model (CountVectorizer) to convert the cleaned text into numeric features. This is needed for training the machine learning model.

# Train Bag of Words model

from sklearn.feature\_extraction.text import CountVectorizer

cv = CountVectorizer()

X\_train\_cv = cv.fit\_transform(X\_train)

X\_train\_cv.shape

**Model Training and Evaluation**

In this part, we are training a Logistic Regression model and evaluating the confusion matrix of the trained model.

# Training Logistic Regression model

from sklearn.linear\_model import LogisticRegression

lr = LogisticRegression()

lr.fit(X\_train\_cv, y\_train)

# transform X\_test using CV

X\_test\_cv = cv.transform(X\_test)

# generate predictions

predictions = lr.predict(X\_test\_cv)

predictions

# confusion matrix

import pandas as pd

from sklearn import metrics

df = pd.DataFrame(metrics.confusion\_matrix(y\_test,predictions), index=['ham','spam'], columns=['ham','spam'])

df