mids2

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv("Mental-Health-Twitter.csv")

df.head()

df.shape

df.isnull().sum() # check for missing values

df = df[['post\_text']]

df.head()

df.describe()

df.shape

df["post\_text"] = df["post\_text"].apply(lambda x: " ".join(x.lower() for x in x.split()))

df["post\_text"] = df["post\_text"].str.replace(r"\d", "", regex=True)

df["post\_text"] = df["post\_text"].str.replace(r"[^\w\s]", "", regex=True)

!pip install textblob

!pip install nltk

from nltk.corpus import stopwords

from textblob import TextBlob

from textblob import Word

import nltk

nltk.download("stopwords")

from nltk.corpus import stopwords

sw = stopwords.words("english")

df["post\_text"] = df["post\_text"].apply(lambda x: " ".join(x for x in x.split() if x not in sw))

!pip install textblob

from textblob import TextBlob

!unzip /usr/share/nltk\_data/corpora/wordnet.zip -d /usr/share/nltk\_data/corpora/

from textblob import Word

nltk.download("wordnet")

nltk.download("omw-1.4")

df["post\_text"] = df["post\_text"].apply(lambda x: " ".join([Word(x).lemmatize()]))

!pip install spacy

!python -m spacy download en\_core\_web\_sm

!pip install -U NLTK

import nltk

nltk.download('punkt\_tab')

nltk.download('punkt')

from nltk.tokenize import word\_tokenize

df["tokens"] = df["post\_text"].apply(lambda x: TextBlob(x).words)

df.head()

blob\_emptylist = []

for i in df["post\_text"]:

blob = TextBlob(i).sentiment

blob\_emptylist.append(blob)

df2 = pd.DataFrame(blob\_emptylist)

df2.head()

df3 = pd.concat([df.reset\_index(drop=True), df2], axis=1)

df3.head()

df4 = df3[['post\_text','tokens','polarity']]

df4.head(6)

df4["Sentiment"] = np.where(df4["polarity"] >= 0 , "Positive", "Negative")

df4.head()

result = df4["Sentiment"].value\_counts()

sns.set\_style("whitegrid")

sns.despine(left=True, bottom=True)

result.plot(kind="bar", rot=0, color=["plum","cyan"]);

df4.groupby("Sentiment").count()

df4.groupby("polarity").max().head(5)

plt.figure(figsize=(8,4))

sns.histplot(df4['polarity'], bins=15, kde=False)

plt.xlabel('Polarity')

plt.ylabel('Count')

plt.title('Polarity Distribution')

plt.figure(figsize=(10,6))

sns.countplot(x='Sentiment', data=df4,order=df4['Sentiment'].value\_counts().index)

plt.xlabel('Sentiment')

plt.ylabel('Count')

plt.title('Sentiment Distribution')

plt.show()

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

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df4['post\_text'], df4['Sentiment'], test\_size=0.2, random\_state=42)

vectorizer = CountVectorizer()

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

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

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

knn.fit(X\_train, y\_train)

accuracy = knn.score(X\_test, y\_test)

print('Accuracy:', accuracy)

print('The accuracy of the KNN Classifier is',round(accuracy\_score(knn.predict(X\_test), y\_test)\*100,2),'%')

print(classification\_report(y\_test, knn.predict(X\_test)))

from sklearn.metrics import ConfusionMatrixDisplay

cm = confusion\_matrix(y\_test, knn.predict(X\_test))

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=knn.classes\_)

disp.plot()