

SVM PYTHON CODE FOR NEWSGROUP:

from sklearn.datasets import fetch\_20newsgroups

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.svm import SVC

from sklearn.pipeline import make\_pipeline

from sklearn import metrics

# Load the datasets

train = fetch\_20newsgroups(subset='train', shuffle=True, random\_state=42)

test = fetch\_20newsgroups(subset='test', shuffle=True, random\_state=42)

# Create a pipeline that combines a TF-IDF vectorizer with an SVM classifier

model = make\_pipeline(TfidfVectorizer(), SVC(kernel='linear'))

# Train the model

model.fit(train.data, train.target)

# Predict the categories for the test data

predicted = model.predict(test.data)

# Calculate the accuracy

accuracy = metrics.accuracy\_score(test.target, predicted)

print(f'Accuracy: {accuracy:.2f}')

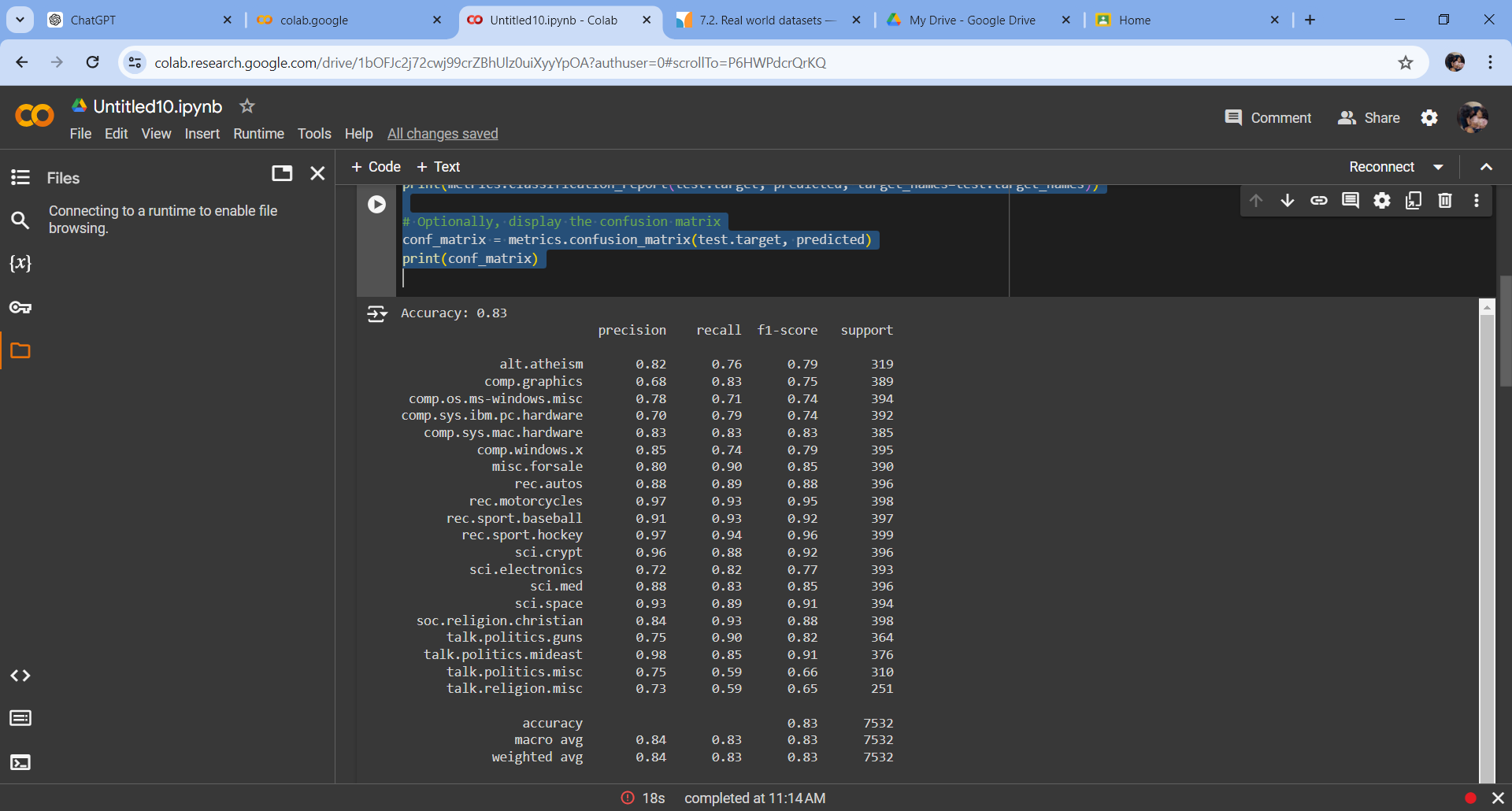
# Display the classification report

print(metrics.classification\_report(test.target, predicted, target\_names=test.target\_names))

# Optionally, display the confusion matrix

conf\_matrix = metrics.confusion\_matrix(test.target, predicted)

print(conf\_matrix)



FOR MOVIE REVIEW:

import os

import numpy as np

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.svm import SVC

from sklearn.pipeline import make\_pipeline

from sklearn import metrics

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

# Function to load data from the folder

def load\_data\_from\_folder(folder\_path):

    texts = []

    labels = []

    for label in os.listdir(folder\_path):

        label\_path = os.path.join(folder\_path, label)

        if os.path.isdir(label\_path):

            for filename in os.listdir(label\_path):

                file\_path = os.path.join(label\_path, filename)

                with open(file\_path, 'r', encoding='utf-8') as file:

                    texts.append(file.read())

                    labels.append(label)

    return texts, labels

# Load the dataset

folder\_path = '/content/drive/MyDrive/txt\_sentoken'  # Update this path to your dataset folder

texts, labels = load\_data\_from\_folder(folder\_path)

# Split the data into training and testing sets

train\_data, test\_data, train\_labels, test\_labels = train\_test\_split(texts, labels, train\_size=0.1, random\_state=42)

# Create a pipeline that combines a TF-IDF vectorizer with an SVM classifier

model = make\_pipeline(TfidfVectorizer(), SVC(kernel='linear'))

# Train the model

model.fit(train\_data, train\_labels)

# Predict the categories for the test data

predicted = model.predict(test\_data)

# Calculate the accuracy

accuracy = metrics.accuracy\_score(test\_labels, predicted)

print(f'Accuracy: {accuracy:}')

# Display the classification report

print(metrics.classification\_report(test\_labels, predicted))

# Optionally, display the confusion matrix

conf\_matrix = metrics.confusion\_matrix(test\_labels, predicted)

print(conf\_matrix)

