1. Load Dataset

import pandas as pd

import numpy as np

data = pd.read\_csv('labeledTrainData.tsv', sep='\t', encoding='latin-1')

2. Preprocessing

import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

stop = stopwords.words('english')

import re

def preprocessor(text):

text = re.sub('<[^>]\*>','',text) # Your code here

emoticons = re.findall('(?::|;|=)(?:-)?(?:\)|\(|D|P)', text)

text = (re.sub('[\W]+', ' ', text.lower()) + ' ' + ' '.join(emoticons).replace('-', ''))

return text

from nltk.stem import PorterStemmer

porter = PorterStemmer()

def tokenizer\_porter(text):

token = []

token = [porter.stem(word) for word in text.split()]

return token

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

X = data['review']

y = data['sentiment']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y)

from sklearn.pipeline import Pipeline

from sklearn.linear\_model import LogisticRegression

from sklearn.feature\_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(stop\_words=stop,

tokenizer=tokenizer\_porter,

preprocessor=preprocessor)

clf = Pipeline([('vect', tfidf),

('clf', LogisticRegression(random\_state=0))])

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

3. Create Model and Train

from sklearn.pipeline import Pipeline

from sklearn.linear\_model import LogisticRegression

from sklearn.feature\_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(stop\_words=stop,

tokenizer=tokenizer\_porter,

preprocessor=preprocessor)

clf = Pipeline([('vect', tfidf),

('clf', LogisticRegression(random\_state=0))])

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

4. Evaluate Model

y\_predict = clf.predict(X\_test)

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

from sklearn.metrics import multilabel\_confusion\_matrix

accuracy\_score(y\_test, y\_predict)

import seaborn as sns

sns.heatmap(confusion\_matrix(y\_test, y\_predict))

classification\_report(y\_test, y\_predict).split('\n')

5. Export Model

import pickle

import os

pickle.dump(clf, open(os.path.join('logisticRegression.pkl'), 'wb'), protocol=4)