

# Flask

November 22, 2022

```
[1]: # pip install flask
```

```
[2]: # importing required libraries
import pandas as pd
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS, TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import f1_score
from sklearn.model_selection import train_test_split
```

```
[3]: # read the dataset
data = pd.read_csv('twitter_sentiments.csv')
# view the top rows
data.head()
```

```
[3]:
```

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation

```
[4]: # train test split
train, test = train_test_split(data, test_size = 0.2, stratify = data['label'],
    ↪random_state=21)

# get the shape of train and test split.
train.shape, test.shape
## >> ((25569, 3), (6393, 3))
```

```
[4]: ((25569, 3), (6393, 3))
```

```
[5]: # create a TF-IDF vectorizer object
tfidf_vectorizer = TfidfVectorizer(lowercase= True, max_features=1000,
    ↪stop_words=ENGLISH_STOP_WORDS)

# fit the object with the training data tweets
```

```
tfidf_vectorizer.fit(train.tweet)
```

```
[5]: TfidfVectorizer(max_features=1000,
                    stop_words=frozenset({'a', 'about', 'above', 'across', 'after',
                                         'afterwards', 'again', 'against', 'all',
                                         'almost', 'alone', 'along', 'already',
                                         'also', 'although', 'always', 'am',
                                         'among', 'amongst', 'amoungst', 'amount',
                                         'an', 'and', 'another', 'any', 'anyhow',
                                         'anyone', 'anything', 'anyway',
                                         'anywhere', ...}))
```

```
[6]: # transform the train and test data
train_idf = tfidf_vectorizer.transform(train.tweet)
test_idf  = tfidf_vectorizer.transform(test.tweet)
```

```
[7]: # create the object of LinearRegression Model
model_LR = LogisticRegression()

# fit the model with the training data
model_LR.fit(train_idf, train.label)

# predict the label on the training data
predict_train = model_LR.predict(train_idf)

# predict the model on the test data
predict_test = model_LR.predict(test_idf)

# f1 score on train data
f1_score(y_true= train.label, y_pred= predict_train)
## >> 0.4888178913738019

f1_score(y_true= test.label, y_pred= predict_test)
## >> 0.45751633986928114
```

```
[7]: 0.45751633986928114
```

```
[8]: # define the stages of the pipeline
pipeline = Pipeline(steps= [('tfidf', TfidfVectorizer(lowercase=True,
                                                       max_features=1000,
                                                       stop_words=
↳ ENGLISH_STOP_WORDS)),
                             ('model', LogisticRegression())])

# fit the pipeline model with the training data
pipeline.fit(train.tweet, train.label)
```

```
[8]: Pipeline(steps=[('tfidf',
                      TfidfVectorizer(max_features=1000,
                                      stop_words=frozenset({'a', 'about', 'above',
                                                           'across', 'after',
                                                           'afterwards', 'again',
                                                           'against', 'all',
                                                           'almost', 'alone',
                                                           'along', 'already',
                                                           'also', 'although',
                                                           'always', 'am', 'among',
                                                           'amongst', 'amoungst',
                                                           'amount', 'an', 'and',
                                                           'another', 'any',
                                                           'anyhow', 'anyone',
                                                           'anything', 'anyway',
                                                           'anywhere', ...}))),
                  ('model', LogisticRegression())])
```

```
[9]: # sample tweet
text = ["Virat Kohli, AB de Villiers set to auction their 'Green Day' kits from_
↪2016 IPL match to raise funds"]

# predict the label using the pipeline
pipeline.predict(text)
## >> array([0])
```

```
[9]: array([0], dtype=int64)
```

```
[10]: # import joblib
from joblib import dump
```

```
[11]: # dump the pipeline model
dump(pipeline, filename="text_classification.joblib")
```

```
[11]: ['text_classification.joblib']
```

```
[12]: # import joblib
from joblib import load

# sample tweet text
text = ["Virat Kohli, AB de Villiers set to auction their 'Green Day' kits from_
↪2016 IPL match to raise funds"]

# load the saved pipeline model
pipeline = load("text_classification.joblib")

# predict on the sample tweet text
```

```
pipeline.predict(text)
## >> array([0])
```

```
[12]: array([0], dtype=int64)
```

```
[13]: # !pip3 install tweepy
```

```
[14]: # import required libraries
import tweepy
import time
import pandas as pd
pd.set_option('display.max_colwidth', 1000)

# api key
api_key = '████████████████████████████████████████'
# api secret key
api_secret_key = '████████████████████████████████████████████████████████████████████████████████'
# access token
access_token = '████████████████████████████████████████████████████████████████████████████████'
# access token secret
access_token_secret = '████████████████████████████████████████████████████████████████████████████████'

# authorize the API Key
authentication = tweepy.OAuthHandler(api_key, api_secret_key)

# authorization to user's access token and access token secret
authentication.set_access_token(access_token, access_token_secret)

# call the api
api = tweepy.API(authentication, wait_on_rate_limit=True)
```

```
[15]: def get_related_tweets(text_query):
    # list to store tweets
    tweets_list = []
    # no of tweets
    count = 50
    try:
        # Pulling individual tweets from query
        for tweet in api.search_tweets(q=text_query, count=count):
            print(tweet.text)
            # Adding to list that contains all tweets
            tweets_list.append({'created_at': tweet.created_at,
                                'tweet_id': tweet.id,
                                'tweet_text': tweet.text})
        return pd.DataFrame.from_dict(tweets_list)

    except BaseException as e:
```

```
print('failed on_status,', str(e))
time.sleep(3)
```

```
[16]: from get_tweets import get_related_tweets
```

```
[17]: # importing the required libraries
from flask import Flask, render_template, request, redirect, url_for
from joblib import load

# load the pipeline object
pipeline = load("text_classification.joblib")

# function to get results for a particular text query
def requestResults(name):
    # get the tweets text
    tweets = get_related_tweets(name)
    # get the prediction
    tweets['prediction'] = pipeline.predict(tweets['tweet_text'])
    # get the value counts of different labels predicted
    data = str(tweets.prediction.value_counts()) + '\n\n'
    return data + str(tweets)
```

```
[18]: # start flask
app = Flask(__name__)

# render default webpage
@app.route('/')
def home():
    return render_template('home.html')

# when the post method detect, then redirect to success function
@app.route('/', methods=['POST', 'GET'])
def get_data():
    if request.method == 'POST':
        user = request.form['search']
        return redirect(url_for('success', name=user))

# get the data for the requested query
@app.route('/success/<name>')
def success(name):
    return "<xmp>" + str(requestResults(name)) + " </xmp> "
```

```
[ ]: app.run(debug=False)
```

```
* Serving Flask app '__main__' (lazy loading)
* Environment: production
```

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

\* Debug mode: off

\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

127.0.0.1 - - [22/Nov/2022 10:19:09] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [22/Nov/2022 10:19:14] "POST / HTTP/1.1" 302 -

127.0.0.1 - - [22/Nov/2022 10:19:15] "GET /success/trump HTTP/1.1" 200 -

@KK4176 @DTNutt @Leslieoo7 Lol every time you tweet you prove how ignorant you are. You can check my timeline to se... <https://t.co/l9NgAHaXBs>

RT @OccupyDemocrats: BREAKING: First Lady Jill Biden happily welcomes the Christmas tree to the White House, says it's "beautiful" and she...

RT @monicaonairtalk: The premiere that may have ended Donald Trump's candidacy & will hopefully lead to the indictment of Pfizer & Moderna e...

@ICheckertm @BulowMatthew @JD\_Wallen1 @RepAdamSchiff <https://t.co/Yl1qAivUpP>

RT @PalmerReport: 1) Manhattan DA Alvin Bragg is leaking that he's looking at criminally charging Donald Trump in the Stormy Daniels scanda...

RT @MattBinder: no doubt about it: Donald Trump was absolutely telling the truth about his meeting with Elon Musk.

he's doing it right no...

RT @Elizabe99819964: @tribelaw @AshaRangappa\_ I just hope the net is flung wide enough to include both the Senate and the House. There are...

RT @Bit2Loud: @TimRunsHisMouth @elonmusk They were, but trump (a suspect) compromised the investigation by having Maxwell intercepted in pr...

@GuntherEagleman My Prez Trump

RT @w\_terrence: Biden cares about black votes.

Trump cares about black people.

End of story.

RT @lavern\_spicer: She's making a strong case for Trump's candidacy with that statement. <https://t.co/YXMoK7fEs0>

@whitlock\_anne @lavern\_spicer Sounds like something a Republican would say. Got it?

@ilyrightback And him liking anti trump tweets back in 19/20 and being a child of a refugee and queer means nothing??? It's weird

@newkingej @w\_terrence Yep very true I like Trump but if Desantis runs I'm voting for him we need something new and fresh

RT @RadioFreeTom: For some left-leaning Twitter users, simply staying online could feel like a small step toward saving democracy, @kait\_ti...

RT @TheBabylonBee: Taliban Quits Twitter To Protest Return Of Trump

<https://t.co/zuOSBsmlMi>

RT @TheBabylonBee: Taliban Quits Twitter To Protest Return Of Trump

<https://t.co/zuOSBsmlMi>

@BrendaRuffin13 @LynnGre31717113 @LBJFKploitics @TeamPelosi Trump 2024

@OccupyDemocrats Funny coming from a guy who's said for year's he had hard evidence Trump was connected to Russia,... <https://t.co/D2a2lU4TH1>  
@Zeednuts @swtdreams1007 @Riflh97 @guelphgirlchris @lavern\_spicer No instead they had to hide his staff that wasn't... <https://t.co/diuFuZuD9d>  
If the DA is coming after Trump for this THAT just makes him more ADMIRABLE by the American people because we don't c... <https://t.co/oqDE2RB7rb>  
RT @LongTimeHistory: #wtpBLUE #ONEV1 #FreshResists

#### TODAY IN HISTORY

1986 - Oliver North & secretary shred documents implicating them in I...  
@carita\_paige @Steve\_Gallardo @CaptMarkKelly @katiehobbs @Adrian\_Fontes  
@krismayes Trump this, Trump that, every De... <https://t.co/64LmjR6JGV>  
RT @BaddCompani: Mike Pompeo interfered in the Peaceful Transfer of Power

Stating, and I basically quote:

'There will be a Peaceful Trans...  
RT @evoespueblo: Trump reivindica las políticas intervencionistas de la Casa Blanca y hace un llamado desesperado a "detener la expansión d...  
Column: Trump running in 2024 is a dream come true for Democrats. He's the weakest candidate out there <https://t.co/wpNB6ReLec>  
RT @OccupyDemocrats: BREAKING: First Lady Jill Biden happily welcomes the Christmas tree to the White House, says it's "beautiful" and she...  
@SariSwift @Benguin420 @AresOfRivia @RollingStone Not jack or trent...because they no like trump...  
RT @sandibachom: @ShotwellNancy I'm not going anywhere. I'm used to toxic environments. I was in Charlottesville and the Capitol and filme...  
RT @zerohedge: The Lords Of War: The Perils Facing Trump, Garland, & Smith In Washington's Legal Arms Race <https://t.co/OhKE2rtsbN>  
Donald Trump tried to stir novelists on behalf of a grocery store.  
RT @ClayTravis: Two years ago Donald Trump told CBS's Lesley Stahl on 60 Minutes the Hunter Biden laptop was real and should be covered by...  
RT @leftistlitwick: tbh i think the underlying reason behind the midterms going so badly for Republicans is that they misconstrued Trump wi...  
@Shorbri72 @wes120121 @ericareport You read current events like some Christians read the Bible. You don't get to pi... <https://t.co/QqpDYBBv16>  
RT @lindyli: Evangelicals gave us Trump

Evangelicals gave us the ChristoFascist Supreme Court that destroyed our reproductive rights

Evan...

Beltway Insider: Biden/G20, GOP/House, Russia/Ukraine, COVID/Vaccine Totals, Musk/Trump, Elizabeth Holmes <https://t.co/eBYyGCXF3r>  
RT @OccupyDemocrats: BREAKING: New York Times "Trump whisperer" Maggie Haberman reveals that "his heart" isn't in running because he "wants...  
RT @RollingStone: Jack White is no longer on Twitter: "So you gave Trump his

Twitter platform back. Absolutely disgusting, Elon." <https://t...>  
RT @MayoIsSpicyy: Trump is a fucking loser and the GOP knows it.  
RT @ddanpereira: Manhattan DA Alvin Bragg may be revisiting the Stormy Daniels hush money cover up led by Individual 1, Donald Trump. Only...  
This guy is desperate for attention, it's really great he found it in the arms of CNN.

<https://t.co/mFtzeuDq6p>  
@WmMcKinley25 @kyledcheney "The U.S. intelligence community's conclusion that Russia interfered in the 2016 electio... <https://t.co/1A11Za9LlA>  
RT @TribelSocial: Out official statement on Elon Musk reinstating Trump's Twitter account: "Unlike Musk, our main focus is not generating a...  
RT @ungerbn103: Trump stole and lied about 27 boxes of Top Secret and Classified Documents and he is allowed to run for President again.

E...  
RT @JudiciaryGOP: Trump was right again, and again, and again, and again, and again.  
@HouseGOP Trump imposed nearly \$80 billion worth of new taxes on Americans by levying tariffs on 1000s of products,... <https://t.co/0xoGtr4f4t>