```
import re
import tweepy
from tweepy import OAuthHandler
from textblob import TextBlob


class TwitterClient(object):
    def __init__(self):
        consumer_key = [redacted]
        consumer_secret = [redacted]
        access_token = [redacted]
        access_token_secret = [redacted]
        try:
            self.auth = OAuthHandler(consumer_key, consumer_secret)
            self.auth.set_access_token(access_token, access_token_secret)
            self.api = tweepy.API(self.auth)
        except:
            print("Error: Authentication Failed")
    def clean_tweet(self, tweet): #to remove links
        return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t]) | (\w +:\ / \ /
\S +)", " ", tweet).split())
    def get_tweet_sentiment(self, tweet): #dataset
        analysis = TextBlob(self.clean_tweet(tweet))
        # set sentiment
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'
    def get_tweets(self, query, count=10): #fetch the dataset and train them
        tweets = []
        try:
            fetched_tweets = self.api.search(q=query, count=count)
            for tweet in fetched_tweets:
                parsed_tweet = {}
                parsed_tweet['text'] = tweet.text
                parsed_tweet['sentiment'] = self.get_tweet_sentiment(tweet.text)
                if tweet.retweet_count > 0:
                    # if tweet has retweets, ensure that it is appended only once
                    if parsed_tweet not in tweets:
                        tweets.append(parsed_tweet)
                else:
                    tweets.append(parsed_tweet)
            return tweets
        except tweepy.TweepError as e:
            # print error (if any)
            print("Error : " + str(e))
def main():
    api = TwitterClient()
    tweets = api.get_tweets(query='Donald Trump', count=200)
    ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
    print("Positive tweets percentage: {} %".format(100 * len(ptweets) / len
(tweets)))
    ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
    print("Negative tweets percentage: {} %".format(100 * len(ntweets) / len
(tweets)))
    print("\n\nPositive tweets:")
    for tweet in ptweets[:1]:
        print(tweet['text'])
    print("\n\
nNegative tweets:")
    for tweet in ntweets[:10]:
        print(tweet['text'])
if __name__ == "__main__":
    main()
```