**Code for Steps 1-5:**

import tweepy

key = "T4RedjB2A62I8TreKYpvsYDdO"

secret = "yVs7UQbiMjCMMTtBcuzFsK0FNMjWwm0QIgVQ1fqO5cBlDuQCXd"

twitter\_handle\_name = ["nytimes","biden4pres","POTUS","TonyFratto","BernieSanders"] #select 5 tweeter user accounts

auth = tweepy.OAuthHandler(key, secret) #Referencing auth to API variable

api = tweepy.API(auth) #Referencing auth to API variable

goodWords = [' like ', ' love ', ' nice ', ' sweet ', ' good ', ' happy ', ' joy ', ' yeah ', ' awesome ', ' wonderful ',

' laugh ', ' yes ']

badWords = [' hate ', ' no ', ' bad ', ' dirty ', ' sad ', ' nope ', ' terrible ', ' not ', ' horrible ', ' sucks ', ' awful ', ' yuck ',

' nah '] #define lists of good and bad word

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer #import libraries

from textblob import TextBlob

from html import unescape

with open('twitter\_scores.csv', 'w') as f: #writing to a new csv file

f.write("Name account"+','+"Sentiment"+','+"Vader"+','+"Textblob"+','+"Retweet"+"\n") #create title in first row of csv file

for name in twitter\_handle\_name: #for loop for each user

tweet\_data = api.user\_timeline(screen\_name=name, tweet\_mode="extended", count=200) #reference tweeter\_data from api and pull 200 tweets per user

for tweet in tweet\_data: # for loop for each tweet

words = unescape(tweet.full\_text.lower()) #defining a variable to represent full tweet text in all lowercase letters

retweet = tweet.retweet\_count #defining a variable to represent virality

goodcount = 0 #creating a startpoint to add each tweet containing a good word

badcount = 0 #creating a startpoint to add each tweet containing a bad word

for good in goodWords: #for loop for each word in good word list

goodcount= goodcount + words.count(good) #add to good count variable when a good word appears in tweet

for bad in badWords: #for loop for each word in bad word list

badcount= badcount + words.count(bad) #add to bad count variable when a bad word appears in tweet

sentiment = goodcount - badcount #sentiment score calculation

analyzer = SentimentIntensityAnalyzer() #defining a variable to represent vader library

wordvader = tweet.full\_text #defining a viarable to represent words from tweet text

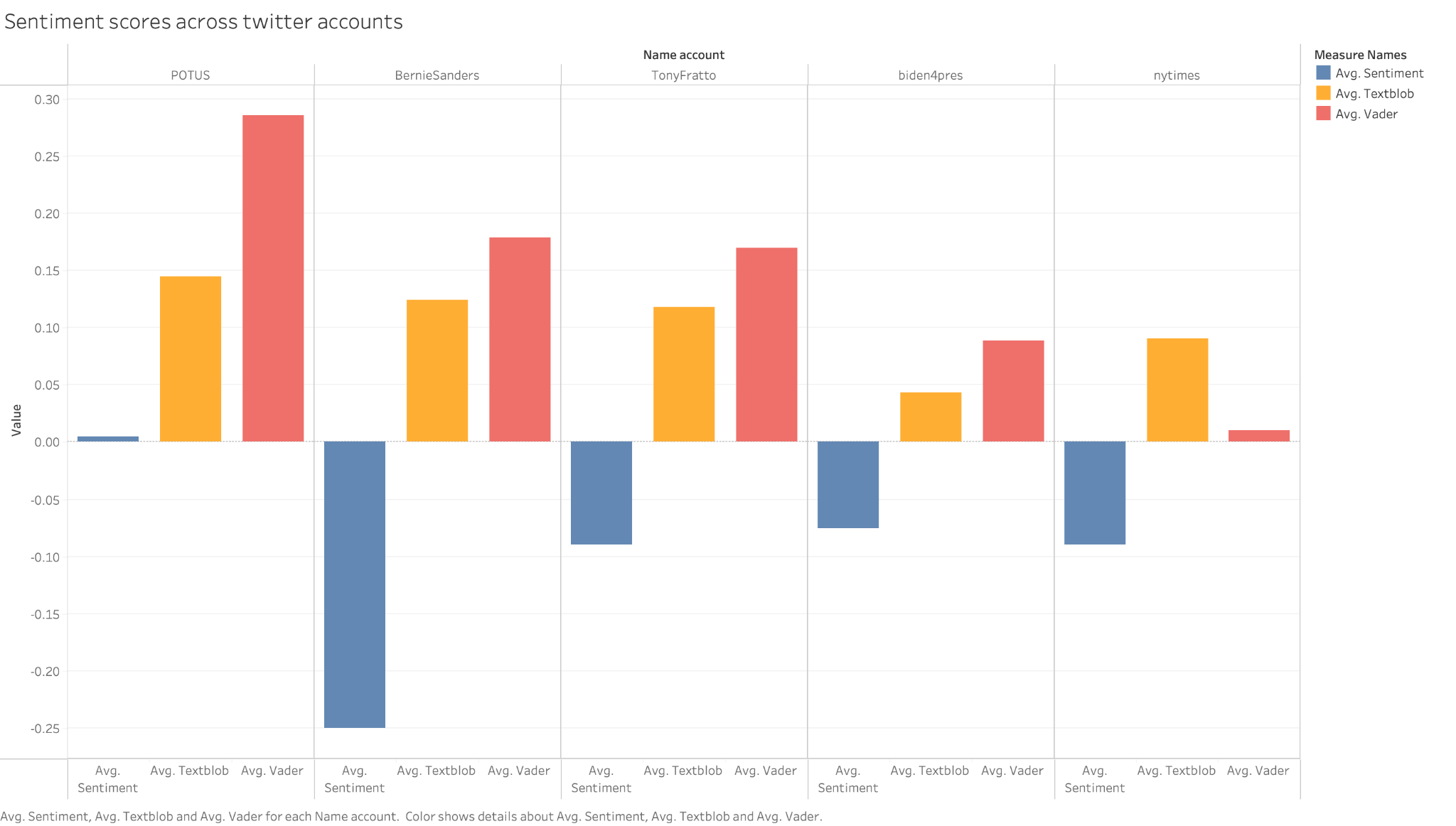
word = TextBlob(tweet.full\_text) #defining a variable to represent textblob library

f.write('"'+name+'"'+','+str(sentiment)+','+str(analyzer.polarity\_scores(wordvader)["compound"])+','+str(round(word.sentiment.polarity,4))+','+str(retweet)+"\n")

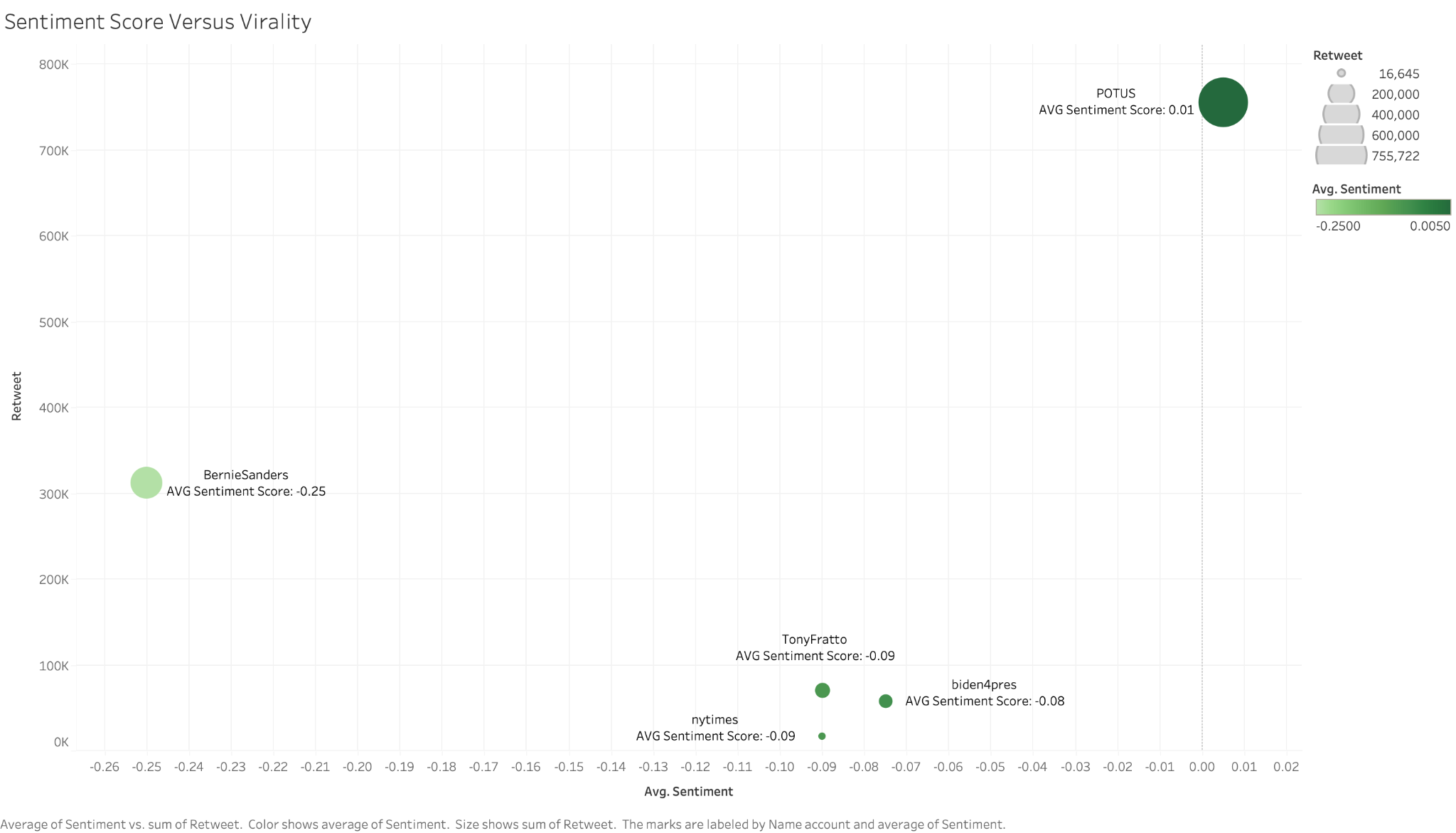
#for above function write scores for each tweet in csv file according to each scoring method

**Step 6 Tableau Visualizations:**

We created a side-by-side bar chart to compare sentiment score averages across the different methods for each of the Twitter accounts we analyzed:



Then, we created a scatter plot showing the relative virality of the accounts by basing dot size on retweet count, paired with each user’s average sentiment score that we coded, which is reflected in the color intensity:



**Part III Weather API Code:**

import csv

import meteostat #import weather data

from datetime import datetime

s = datetime(2020,1,1) #start date

e = datetime(2020,12,31) #end date

loc = meteostat.Point(42.348495, -83.060303, 70) #location coordinates

data = meteostat.Daily(loc, s, e) #pull weather data from loc (detroit) between start and end date

data = data.fetch() #update the data

measures = ["tmin", "tmax"] #defining measure variable

table = data[measures] #grab min and max temperature from weather api

print(table)