topic_analysis

August 6, 2020

```
[89]: import pandas as pd
      import re
      from sklearn.feature_extraction.text import CountVectorizer
      from textblob import TextBlob
      import nltk
      from nltk.corpus import stopwords
      import ast
      import matplotlib.pyplot as plt
      import seaborn as sns
      from datetime import datetime,date
      import json
      import numpy as np
      df_trump=pd.read_csv("tweets_donald_trump.csv",sep=",")
      df_biden=pd.read_csv("tweets_joe_biden.csv",sep=",")
      df_trump=df_trump.dropna(thresh=2)
      df_biden=df_biden.dropna(thresh=2)
      from sklearn.decomposition import LatentDirichletAllocation
      from sklearn.feature_extraction.text import CountVectorizer
      from prettytable import PrettyTable


      def clean_tweet(tweet):
           """funzione pulizia tweet tramite regular expressione"""
             return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t]) |(\w+:\/\/
       ↪\S+)", " ", tweet).split())


      def prepr(df):
          df=df[df['text']!="text"]#aggiungendo i dati al csv venivano aggiunti␣
       ↪nuovamente i nomi delle colonne, questa riga di codice mi permette di␣
       ↪rimuoverle
          df['text'] = df['text'].apply(clean_tweet)#applico funzione per la pulizia␣
       ↪dei tweet
          df.drop(df.columns[7:], axis=1, inplace=True)#cancello colonne inutili.␣
       ↪problema causato dal fatto che inizialmente stavo salvando i tweet in modo␣
       ↪diverso
          df=df.loc[2:]
```

```
    df=df.dropna(thresh=3)#rimuove le righe con all'interno almeno 3 NA
    df=df[df['sentiment'].str.startswith('{')]#scelgo per la colonna sentiment␣
 ↪solo le righe che iniziano con "{" perché tweepy restituisce un dizionario
    df['sentiment']=df['sentiment'].apply(ast.literal_eval)#trasforma una␣
 ↪stringa contenente un dizionario, in un dizionario
    df=pd.concat([df.drop(['sentiment'], axis=1), df['sentiment'].apply(pd.
 ↪Series)], axis=1)#divide il dizionario in delle colonne che hanno come␣
 ↪etichetta la chiave del dizionario e all'interno dei campi della colonna␣
 ↪vengono inseriti i valori del dizionario
    return df

df_trump=prepr(df_trump)#applico funzione appena creata
df_biden=prepr(df_biden)#applico funzione appena creata)


"""aggiungo la colonna candidato, mi servirà successivamente quando creerò un␣
 ↪unico dataframe dei tweet dei due candidati
"""
df_biden['candidato'] = 'Joe Biden'
df_trump['candidato'] = 'Donald Trump'
```

[90]:

```
"""rimuovo altri pattern dal text dei tweet tramite le regex"""
df_biden['text'] = df_biden['text'].str.lower()\
          .str.replace('(@[a-z0-9]+)\w+',' ')\
          .str.replace('(http\S+)', ' ')\
          .str.replace('([^0-9a-z \t])',' ')\
          .str.replace(' +',' ')\
          .str.replace('rt', '')\




tweetsSentiment=df_biden.to_dict('records')
```

[91]:

```
def topic_modeling(tweets=None):
    """funzione che prima vettorizza i tweet e poi applica la topic con la␣
 ↪LDA"""
    if not tweets:
        tweets = []
    tf_vectorizer = CountVectorizer(
        max_df=0.95,
        min_df=2,
        max_features=1000,
        stop_words='english'
    )
    tf = tf_vectorizer.fit_transform(tweets)
    tf_feature_names = tf_vectorizer.get_feature_names()
    no_topics = 5
    lda = LatentDirichletAllocation(n_components=no_topics,
                                    max_iter=5,
                                    learning_method='online',
                                    learning_offset=50.,
                                    random_state=0).fit(tf)
    for topic_idx, topic in enumerate(lda.components_):
        print("Topic %d:" % (topic_idx))
        print(" ".join([tf_feature_names[i]
                        for i in topic.argsort()[:-10 - 1:-1]]))
```

[92]:

```
"""tweet di biden"""
positiveTweets = [tweet['text'] for tweet in tweetsSentiment if tweet['sign']␣
 ↪== 'positive']
negativeTweets = [tweet['text'] for tweet in tweetsSentiment if tweet['sign']␣
 ↪== 'negative']
neutralTweets = [tweet['text'] for tweet in tweetsSentiment if tweet['sign'] ==␣
 ↪'neutral']
```

[93]:

```
"""topic dei tweet positivi"""
print("Positive Tweets Trump %d" % len(positiveTweets))
topic_modeling(tweets=positiveTweets)
```

```
Positive Tweets Trump 9269
Topic 0:
biden joe trump amp president leader defund police america safe
Topic 1:
world joe great states wish china communist depament retweet million
Topic 2:
biden joe trump president america really new years election vote
Topic 3:
joe biden just want doesn know pay racist crimes amp
Topic 4:
joe biden american people bide tweeting competent intelligent sees humane
```

[94]:

```
"""topic dei tweet negativi"""
print("Negative Tweets %d" % len(negativeTweets))
topic_modeling(tweets=negativeTweets)
```

```
Negative Tweets 5353
Topic 0:
joe biden trump vote donald president voting white mail 19
Topic 1:
biden joe black long lives don years today record family
Topic 2:
biden joe china wallace chris weak democrats police amp imagine
Topic 3:
```

```
biden joe american destroy kids claims fall closed silent schools
Topic 4:
biden joe rep nunes devin foreign think fixed dc half
```

```
[95]: """topic dei tweet neutrali"""
      print("Neutral Tweets %d" % len(neutralTweets))
      topic_modeling(tweets=neutralTweets)
```

```
Neutral Tweets 8798
Topic 0:
biden joe trump president know donald like voting years doesn
Topic 1:
joe biden jobs president need created hunter went create proven
Topic 2:
biden joe election think america honor come yes words april
Topic 3:
joe biden trump don people president didn saying did pro
Topic 4:
joe biden america radical police left vote puppet trump make
```

```
[36]: """rimuovo altri pattern dal text dei tweet tramite le regex"""
      df_trump['text'] = df_trump['text'].str.lower()\
                .str.replace('(@[a-z0-9]+)\w+',' ')\
                .str.replace('(http\S+)', ' ')\
                .str.replace('([^0-9a-z \t])',' ')\
                .str.replace(' +',' ')\
                .str.replace('rt', '')\




      tweetsSentiment=df_trump.to_dict('records')
```

```
[37]: """tweet di trump"""
      positiveTweets = [tweet['text'] for tweet in tweetsSentiment if tweet['sign']␣
       ↪== 'positive']
      negativeTweets = [tweet['text'] for tweet in tweetsSentiment if tweet['sign']␣
       ↪== 'negative']
      neutralTweets = [tweet['text'] for tweet in tweetsSentiment if tweet['sign'] ==␣
       ↪'neutral']
```

```
[39]: """topic dei tweet positivi"""
      print("Positive Tweets Trump %d" % len(positiveTweets))
      topic_modeling(tweets=positiveTweets)
```

```
Positive Tweets Trump 9844
Topic 0:
trump donald president new jr america 2020 say legal video
Topic 1:
```

```
trump donald people election biden likely turn amp joe defended
Topic 2:
trump donald days 30 putin really long silence agree bountygate
Topic 3:
vote trump donald want early wewillvote loudly piss mail friend
Topic 4:
trump donald right president time years obama like daddy just
```

[40]:

```
"""topic dei tweet negativi"""
print("Negative Tweets %d" % len(negativeTweets))
topic_modeling(tweets=negativeTweets)
```

```
Negative Tweets 6192
Topic 0:
trump donald president election mail new country fraud 2020 america
Topic 1:
trump donald make single don going good knows disgusting evidence
Topic 2:
trump donald days past joe trying biden lives america matter
Topic 3:
trump donald hate federal sent agents called symbol 31 blacklivesmatter
Topic 4:
trump donald fuck people time white november said justice just
```

[41]:

```
"""topic dei tweet neutrali"""

print("Negative Tweets %d" % len(neutralTweetsTweets))
topic_modeling(tweets=neutralTweets)
```

```
Negative Tweets 6192
Topic 0:
trump donald president election mail new country fraud 2020 america
Topic 1:
trump donald make single don going good knows disgusting evidence
Topic 2:
trump donald days past joe trying biden lives america matter
Topic 3:
trump donald hate federal sent agents called symbol 31 blacklivesmatter
Topic 4:
trump donald fuck people time white november said justice just
```