PredictElections.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Predict US Elections with Sentiment Analysis

```
[0] import pandas as pd
    import numpy as np
    import seaborn as sns
    import matplotlib.pyplot as plt
    from textblob import TextBlob
    from wordcloud import WordCloud
    import plotly.graph_objects as go
    import plotly.express as px

    trump_reviews = pd.read_csv("Trumpall2.csv")
    biden_reviews = pd.read_csv("Bidenall2.csv")
```

Now let's have a quick look at the first 5 rows from both the datasets:

```
[2] print(trump_reviews.head())
    print(biden_reviews.head())
```

```
                user                                               text
0      manny_rosen   @sanofi please tell us how many shares the CF...
1        osi_abdul   https://t.co/pt5B6Lqsf2  Like, comment, RT RP...
2          Patsyrw   Your AG Barr is as useless &amp; corrupt as y...
3  seyodebrahimi_m   Mr. Trump! Wake Up!  Most of the comments bel...
4   Jamaica82540477   After 4 years you think you would have figure...
                user                                               text
0   MarkieDoDer3     @JoeBiden And we'll find out who won in 2020,...
1   K8J72279616  @JoeBiden Your Democratic Nazi Party cannot be...
2       OldJaceA                                  @JoeBiden So did lying Barr
3   penblogger  @JoeBiden It's clear you didnt compose this tw...
4  Aquarian8264          @JoeBiden I will vote in person thank you.
```

Now, I will get started with sentiment analysis. I will do it by using the Textblob package in Python. Here I will use this package to perform simple text classification in either positive or negative on the basis of sentiment analysis:

```
[3]
    textblob1 = TextBlob(trump_reviews["text"][10])
    print("Trump :",textblob1.sentiment)
    textblob2 = TextBlob(biden_reviews["text"][500])
    print("Biden :",textblob2.sentiment)
```

```
Trump : Sentiment(polarity=0.15, subjectivity=0.3125)
Biden : Sentiment(polarity=0.6, subjectivity=0.9)
```

```
[4] def find_pol(review):
        return TextBlob(review).sentiment.polarity
    trump_reviews["Sentiment Polarity"] = trump_reviews["text"].apply(find_pol)
    print(trump_reviews.tail())

    biden_reviews["Sentiment Polarity"] = biden_reviews["text"].apply(find_pol)
    print(biden_reviews.tail())
```

```
                user  ... Sentiment Polarity
2783          addiva1  ...              0.000
2784         hidge026  ...              0.000
2785   SpencerRossy   ...              0.225
2786  ScottyMcPherson  ...              0.000
2787          bjklinz  ...             -0.500

[5 rows x 3 columns]
                user  ... Sentiment Polarity
2535    meryn1977  ...               0.15
2536  BSNelson114  ...               0.00
2537       KenCapel  ...               0.00
2538   leslyeHale  ...               0.10
2539      rerickre  ...               0.20

[5 rows x 3 columns]
```

Polarity ranges from -1 to +1(negative to positive) and tells whether the text has negative sentiments or positive sentiments. Polarity tells about factual information.

Sentiment Polarity on Both the candidates:

```
[5] trump_reviews["Expression Label"] = np.where(trump_reviews["Sentiment Polarity"]>0, "positive", "negative")
    trump_reviews["Expression Label"][trump_reviews["Sentiment Polarity"]==0]="Neutral"
    print(trump_reviews.tail())

    biden_reviews["Expression Label"] = np.where(biden_reviews["Sentiment Polarity"]>0, "positive", "negative")
    biden_reviews["Expression Label"][trump_reviews["Sentiment Polarity"]==0]="Neutral"
    print(biden_reviews.tail())
```

```
                user  ... Expression Label
2783          addiva1  ...          Neutral
2784         hidge026  ...          Neutral
2785   SpencerRossy   ...         positive
2786  ScottyMcPherson  ...          Neutral
2787          bjklinz  ...         negative

[5 rows x 4 columns]
                user  ... Expression Label
2535    meryn1977  ...          Neutral
2536  BSNelson114  ...          Neutral
2537       KenCapel  ...         negative
2538   leslyeHale  ...          Neutral
2539      rerickre  ...         positive

[5 rows x 4 columns]
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Drop all the tweets with neutral polarity from both the datasets to balance the data equally. I will also perform some data cleaning operations so that at the can we can easily predict the US Elections:

```
[6] reviews1 = trump_reviews[trump_reviews['Sentiment Polarity'] == 0.0000]
    print(reviews1.shape)

    cond1=trump_reviews['Sentiment Polarity'].isin(reviews1['Sentiment Polarity'])
    trump_reviews.drop(trump_reviews[cond1].index, inplace = True)
    print(trump_reviews.shape)

    reviews2 = biden_reviews[biden_reviews['Sentiment Polarity'] == 0.0000]
    print(reviews2.shape)

    cond2=biden_reviews['Sentiment Polarity'].isin(reviews1['Sentiment Polarity'])
    biden_reviews.drop(biden_reviews[cond2].index, inplace = True)
    print(biden_reviews.shape)
```

```
(1464, 4)
(1324, 4)
(1509, 4)
(1031, 4)
```

Balance both the datasets:

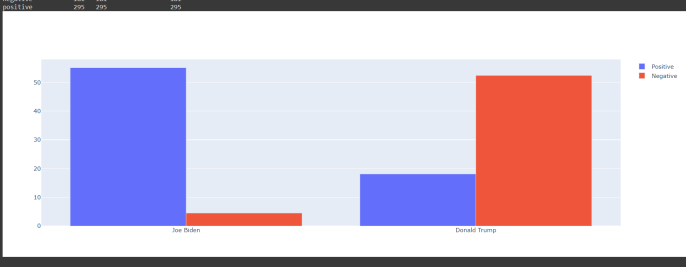
```
[7] # Donald Trump
    np.random.seed(10)
    remove_n =324
    drop_indices = np.random.choice(trump_reviews.index, remove_n, replace=False)
    df_subset_trump = trump_reviews.drop(drop_indices)
    print(df_subset_trump.shape)
    # Joe Biden
    np.random.seed(10)
    remove_n =31
    drop_indices = np.random.choice(biden_reviews.index, remove_n, replace=False)
    df_subset_biden = biden_reviews.drop(drop_indices)
    print(df_subset_biden.shape)
```

```
(1000, 4)
(1000, 4)
```

analyze the data to predict the US Elections, by analyzing the number of positive and negative sentiments in both the accounts:

```
[8] count_1 = df_subset_trump.groupby('Expression Label').count()
    print(count_1)

    negative_per1 = (count_1['Sentiment Polarity'][0]/1000)*10
    positive_per1 = (count_1['Sentiment Polarity'][1]/1000)*100

    count_2 = df_subset_biden.groupby('Expression Label').count()
    print(count_2)

    negative_per2 = (count_2['Sentiment Polarity'][0]/1000)*100
    positive_per2 = (count_2['Sentiment Polarity'][1]/1000)*100

    Politicians = ['Joe Biden', 'Donald Trump']
    lis_pos = [positive_per1, positive_per2]
    lis_neg = [negative_per1, negative_per2]

    fig = go.Figure(data=[
        go.Bar(name='Positive', x=Politicians, y=lis_pos),
        go.Bar(name='Negative', x=Politicians, y=lis_neg)
    ])
    # Change the bar mode
    fig.update_layout(barmode='group')
    fig.show()
```

```
                     text  Sentiment Polarity
Expression Label
negative              449   449                 449
positive              551   551                 551
                     text  Sentiment Polarity
Expression Label
Neutral               524   524                 524
negative              181   181                 181
positive              295   295                 295
```



*From the above figure, it is very clear that **Joe Biden** is getting more Positive tweets and less negative tweets as compared to Donald Trump. So it will not be wrong to conclude that Joe Biden is more prefered by the people to win the US Presidential Elections than Donald Trump.*

0s completed at 10:09 AM