male_female_employment_analysis_using_topic_modeling

June 23, 2020

Topic Modelling of tweets on the job opportunity available for male and females. This analysis is basically tweets collected from online between January 1st and June 20 2020. It is about the job opportunity available for both male and females. A EDA was carried out as well as a topic modelling using the Latent Dirichlet Allocation (LDA) method. The LDA is often used for topic modelling to classify text in a document to a particular topic, thereby alowing one to have an idea of what is been said at a glance. here, out document is the tweet. For the tweets, we first searched for words within the context of job opportunity for any gender, we then went futher to specify job opportunity for male, and female.

[ ]:
```
#!pip3 install plotly
#!pip3 install pyLDAvis
```

[11]:
```
# Importing modules
import os
import re
import pandas as pd
import numpy as np
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly as py
import plotly.graph_objs as go
import gensim
from gensim import corpora, models, similarities
import logging
import tempfile
from nltk.corpus import stopwords
from string import punctuation
from collections import OrderedDict
import seaborn as sns
import pyLDAvis.gensim
import matplotlib.pyplot as plt
%matplotlib inline
import string
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.corpus import stopwords
from string import punctuation
from nltk.tokenize import TweetTokenizer
from nltk import tokenize
```

```
from wordcloud import WordCloud
from PIL import Image
import warnings
warnings.filterwarnings("ignore")
```

[47]:
```
#os.chdir('..')
# Read data into papers
tweet1 = pd.read_csv('malejob.csv') #jobs for male
tweet2 = pd.read_csv('femalejob.csv') #jobs for female
tweet3 = pd.read_csv('any_gender.csv') #jobs offer available

# Print head
tweet1.head(2)
```

[47]:
```
   Unnamed: 0    screen_name                      username  \
0           0  SimbaRashe_OG                   Simba Rashe
1           1  navyservant56  navyservant56 a.k.a. "B. A."

               user_id             tweet_id  \
0  1134271763947413504  1249098583707541504
1  1215055072272506880  1249092591334961153

                                tweet_url            timestamp  \
0  /SimbaRashe_OG/status/1249098583707541504  2020-04-11 22:14:44
1  /navyservant56/status/1249092591334961153  2020-04-11 21:50:55

   timestamp_epochs                                       text  \
0        1586643284  Lerato,  fake account,  u always re-posting th…
1        1586641855  Persecution is alive and all around. I am bein…

                                        text_html  … has_media img_urls  \
0  <p class="TweetTextSize js-tweet-text tweet-te…  …     False       []
1  <p class="TweetTextSize js-tweet-text tweet-te…  …     False       []

   video_url likes  retweets  replies  is_replied  is_reply_to  \
0        NaN     0         0        0       False         True
1        NaN     0         0        0       False        False

   parent_tweet_id                                    reply_to_users
0     1.248967e+18  [{'screen_name': 'uLerato_pillay', 'user_id': …
1              NaN                                                []

[2 rows x 22 columns]
```

[ ]:

[51]:
```
frames = [tweet1,tweet2,tweet3]
tweets = pd.concat(frames)
```

[26]:
```
#!pip3 install chart-studio
```

[52]:
```
tweets['text'][0]
```

[52]:
```
0    Lerato,  fake account,  u always re-posting th…
0    It should preferably be the best person for th…
0    Whether its assistance with your resume and co…
Name: text, dtype: object
```

[53]:
```
len(tweets)
```

[53]: 16009

[54]:
```
# sorting by first name
tweets.sort_values("text", inplace = True)

# dropping ALL duplicte values
tweets.drop_duplicates(subset ="text",
                     keep = False, inplace = True)

# displaying data
tweets.head(2)
```

[54]:
```
       Unnamed: 0      screen_name                       username  \
11351       11351  BobbyBreadcrumb  Goddess Berlin's Thong fund
2168         2168       SnowDragon_      Dana Whites first facial

                   user_id             tweet_id  \
11351  1179219480733507584  1270727329283702784
2168   1143317424550400001  1244432532940169217

                                     tweet_url            timestamp  \
11351  /BobbyBreadcrumb/status/1270727329283702784  2020-06-10 14:39:38
2168       /SnowDragon_/status/1244432532940169217  2020-03-30 01:13:30

       timestamp_epochs                                              text  \
11351        1591799978  \n\nEven though she again turned down my marri…
2168         1585530810  \n\nI'll take catfish all day. \n\nI said It o…

                                               text_html  … has_media  \
11351  <p class="TweetTextSize js-tweet-text tweet-te…  …     False
2168   <p class="TweetTextSize js-tweet-text tweet-te…  …     False

      img_urls  video_url likes  retweets  replies  is_replied  is_reply_to  \
```

```
11351        []        NaN     1          1         0       False         False
2168         []        NaN     1          0         0       False          True

       parent_tweet_id                                    reply_to_users
11351              NaN                                                []
2168      1.244246e+18  [{'screen_name': 'mohamed601phm', 'user_id': '…

[2 rows x 22 columns]
```

[55]:
```python
# This is the new data we are woking with now, after removing duplicates
len(tweets)
```

[55]: 14148

[56]:
```python
# Separating the time variable by hour, day, month and year for further␣
↪analysis using datetime

tweets['timestamp'] = pd.to_datetime(tweets['timestamp'])
tweets['hour'] = tweets['timestamp'].apply(lambda x: x.hour)
tweets['month'] = tweets['timestamp'].apply(lambda x: x.month)
tweets['day'] = tweets['timestamp'].apply(lambda x: x.day)
tweets['year'] = tweets['timestamp'].apply(lambda x: x.year)
tweets['length'] = tweets["text"].apply(len)
tweets['num_of_words'] = tweets["text"].str.split().apply(len)
# addding 1 column for counting
# (dodanie 1 kolumny do zliczania)
tweets['dummy_count'] = 1
tweets.head(5)
```

[56]:
```
       Unnamed: 0      screen_name                        username  \
11351       11351  BobbyBreadcrumb  Goddess Berlin’s Thong fund
2168         2168      SnowDragon_     Dana Whites first facial
3140         3140      cyanhearted                         Nana
220           220          He3Man7                      William
222           222          He3Man7                      William

                   user_id             tweet_id  \
11351  1179219480733507584  1270727329283702784
2168   1143317424550400001  1244432532940169217
3140    924788731546161152  1251049716416200704
220               67532070  1247898964394823683
222               67532070  1247894368540745730

                                        tweet_url           timestamp  \
11351  /BobbyBreadcrumb/status/1270727329283702784 2020-06-10 14:39:38
2168       /SnowDragon_/status/1244432532940169217 2020-03-30 01:13:30
3140       /cyanhearted/status/1251049716416200704 2020-04-17 07:27:50
```

```
220           /He3Man7/status/1247898964394823683 2020-04-08 14:47:52
222           /He3Man7/status/1247894368540745730 2020-04-08 14:29:36

       timestamp_epochs                                               text  \
11351        1591799978  \n\nEven though she again turned down my marri…
2168         1585530810  \n\nI’ll take catfish all day. \n\nI said It o…
3140         1587108470  \n\nhow unfair life's been to the male species…
220          1586357272  \nMake him negatively twitter famous. He attac…
222          1586356176  \nMake him negatively twitter famous. He attac…

                                              text_html  … is_reply_to  \
11351  <p class="TweetTextSize js-tweet-text tweet-te…  …       False
2168   <p class="TweetTextSize js-tweet-text tweet-te…  …        True
3140   <p class="TweetTextSize js-tweet-text tweet-te…  …        True
220    <p class="TweetTextSize js-tweet-text tweet-te…  …        True
222    <p class="TweetTextSize js-tweet-text tweet-te…  …       False

      parent_tweet_id                                     reply_to_users hour  \
11351             NaN                                                 []   14
2168     1.244246e+18  [{'screen_name': 'mohamed601phm', 'user_id': '…    1
3140     1.251035e+18  [{'screen_name': 'SpycieYrn', 'user_id': '1029…    7
220      1.247275e+18  [{'screen_name': 'shaunking', 'user_id': '7551…   14
222               NaN                                                 []   14

       month  day  year  length  num_of_words  dummy_count
11351      6   10  2020     259            46            1
2168       3   30  2020     274            53            1
3140       4   17  2020     195            37            1
220        4    8  2020     267            42            1
222        4    8  2020     326            44            1

[5 rows x 29 columns]
```

[57]:
```python
# Who twitted most about male and female employement opportunities in the last␣
↪6 months

grouped = pd.DataFrame(tweets.groupby('username').size().rename('counts')).
↪sort_values('counts', ascending=False)
grouped.head(10)
```

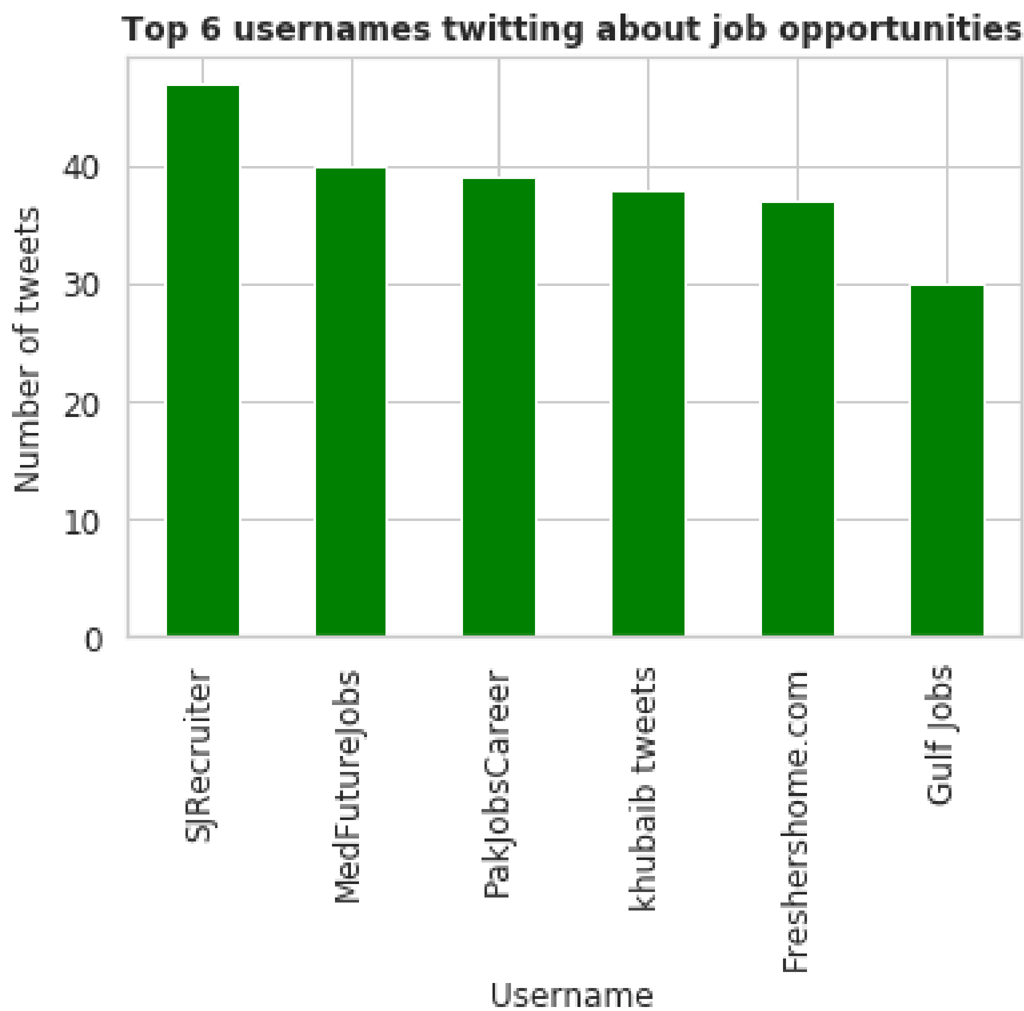
[57]:
```
                   counts
username
SJRecruiter            47
MedFutureJobs          40
PakJobsCareer          39
khubaib tweets         38
Freshershome.com       37
```

```
Gulf Jobs                30
                   23
TWG International        20
Jobswebpk                18
freezonejobs             16
```

Observation: These are the first 10 people/organization that tweeted most about the jobs available and or remote job opportunities for male and female in the past 6 months. `SJRecruiter` has 47 count followed by `MedFutureJobs`. Meaning that this recruiters or individual are worth looking out for. The visualization is as follows:

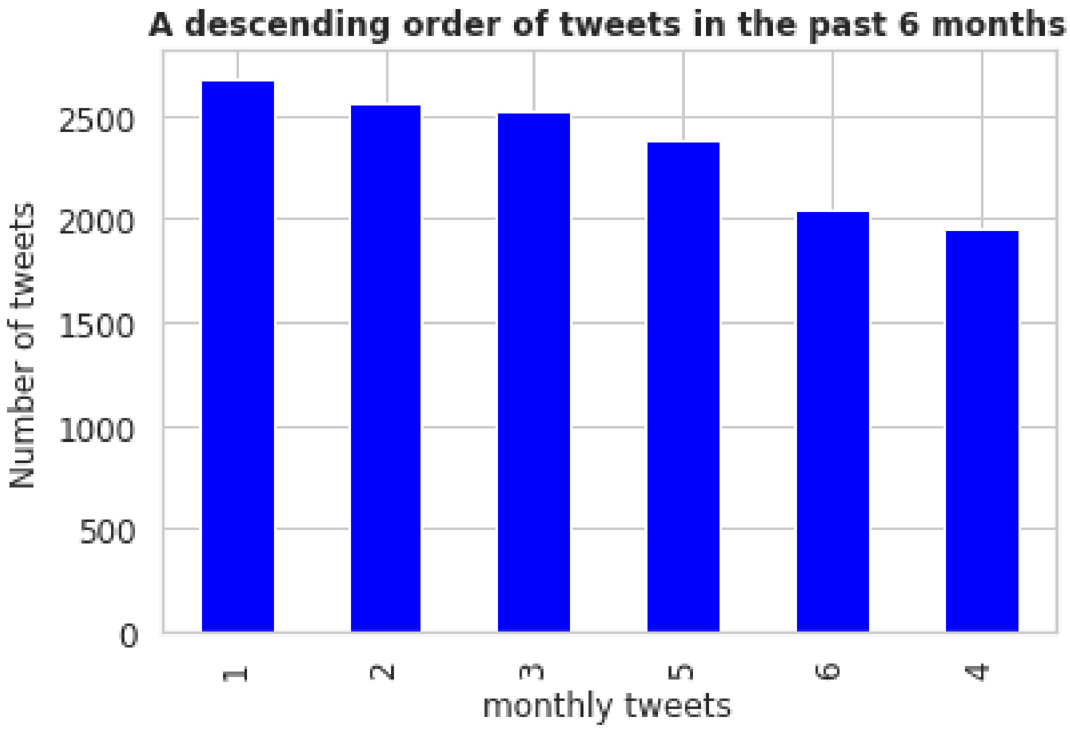
[58]:
```
# Who twitted most about male and female employement opportunities in the last␣
↪6 months

get_ipython().magic('matplotlib inline')
tweets_by_username = tweets['username'].value_counts()

fig, ax = plt.subplots()
ax.tick_params(axis='x', labelsize=12)
ax.tick_params(axis='y', labelsize=12)
ax.set_xlabel('Username', fontsize=12)
ax.set_ylabel('Number of tweets' , fontsize=12)
ax.set_title('Top 6 usernames twitting about job opportunities', fontsize=12,␣
↪fontweight='bold')
tweets_by_username[:6].plot(ax=ax, kind='bar', color='green')
```

[58]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4a2aa40450>
```



[59]:
```
get_ipython().magic('matplotlib inline')
tweets_by_username = tweets['month'].value_counts()

fig, ax = plt.subplots()
ax.tick_params(axis='x', labelsize=12)
ax.tick_params(axis='y', labelsize=12)
ax.set_xlabel('monthly tweets', fontsize=12)
ax.set_ylabel('Number of tweets' , fontsize=12)
ax.set_title('A descending order of tweets in the past 6 months', fontsize=12,␣
 ↪fontweight='bold')
tweets_by_username[:15].plot(ax=ax, kind='bar', color='blue')
```

[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4a2ad580d0>
```



Observation: The bar chart above shows the number of tweets as regarding job opportunities for male, females, and any gender. From the plot, is is observed that most people tweet about this topic more in the first month of the year. meaning that employers take in more employee mostly at the begining of the year, and this trend seems to decrease as the months goes by. This is a good indication that, job seekers should shoot their shoot mostly at the begining of the year.

Preprocessing data using NLTK: Clean, Tokenize, Remove stopwords, Stem, Lemmatize tweets

[124]:
```
#!pip3 install wordcloud
```

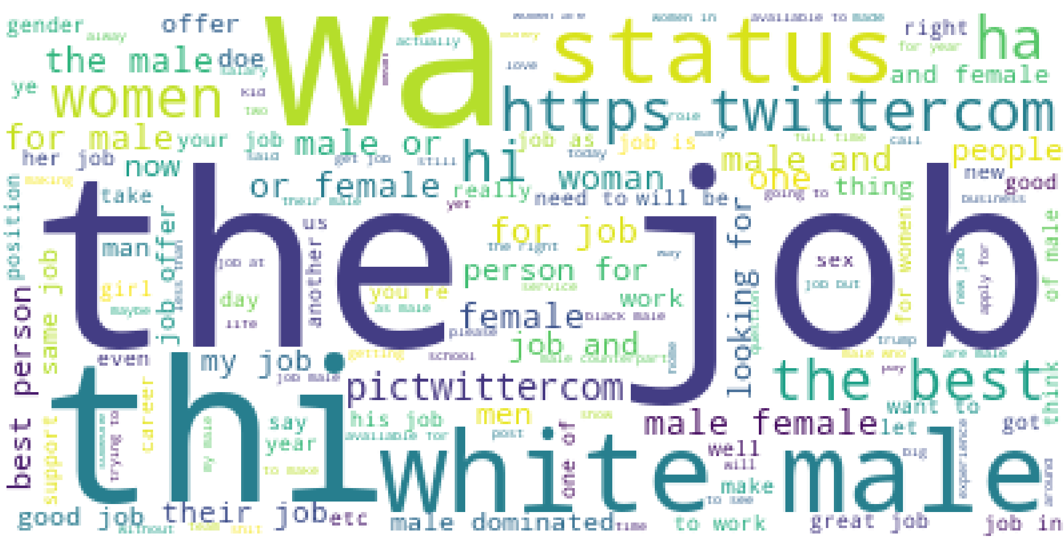
[60]:
```
# Remove punctuation
tweets['tweets_text_processed'] = tweets['text'].map(lambda x: re.sub('[,\.!?
↪]', '', x))
# Convert the titles to lowercase
tweets['tweets_text_processed'] = tweets['tweets_text_processed'].map(lambda x:␣
↪x.lower())
# Print out the first rows of papers
tweets['tweets_text_processed'].head()
```

[60]:
```
11351    \n\neven though she again turned down my marri…
2168     \n\ni'll take catfish all day \n\ni said it on…
```

```
3140      \n\nhow unfair life's been to the male species…
220       \nmake him negatively twitter famous he attack…
222       \nmake him negatively twitter famous he attack…
Name: tweets_text_processed, dtype: object
```

[62]:
```python
# Import the wordcloud library
from wordcloud import WordCloud

# Join the different processed titles together.
long_string = ','.join(list(tweets['tweets_text_processed'].values))
# Create a WordCloud object
wordcloud = WordCloud(background_color="white", max_words=500, contour_width=3, contour_color='steelblue')
# Generate a word cloud
wordcloud.generate(long_string)
# Visualize the word cloud
wordcloud.to_image()
```

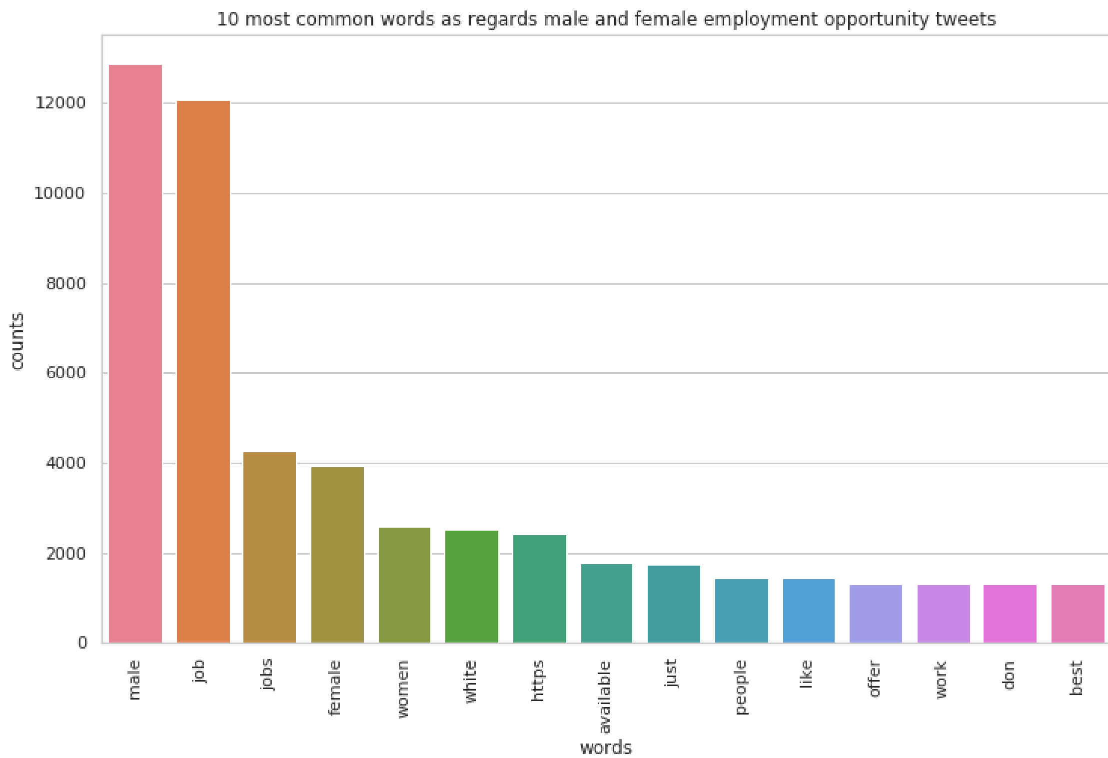
[62]:



Observations: The above diagram is called a wordcloud, we try to see the 500 most frequent words in the tweet. as we can see the words like job, white, are mostly common. however, we can deduce that the word male is mentioned more than the word female, this is so visible in the right hand coner of the plot as wee as the top left hand coner of the plot. Maybe male employees are more prefered compared to women.

[63]:
```python
# Load the library with the CountVectorizer method
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
import matplotlib.pyplot as plt
```

```
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline

# Helper function
def plot_10_most_common_words(count_data, count_vectorizer):
    words = count_vectorizer.get_feature_names()
    total_counts = np.zeros(len(words))
    for t in count_data:
        total_counts+=t.toarray()[0]

    count_dict = (zip(words, total_counts))
    count_dict = sorted(count_dict, key=lambda x:x[1], reverse=True)[0:15]
    words = [w[0] for w in count_dict]
    counts = [w[1] for w in count_dict]
    x_pos = np.arange(len(words))

    plt.figure(2, figsize=(12, 12/1.6180))
    plt.subplot(title='10 most common words as regards male and female␣
 ↪employment opportunity tweets')
    sns.set_context("notebook", font_scale=1, rc={"lines.linewidth": 1.5})
    sns.barplot(x_pos, counts, palette='husl')
    plt.xticks(x_pos, words, rotation=90)
    plt.xlabel('words')
    plt.ylabel('counts')
    plt.show()

# Initialise the count vectorizer with the English stop words
count_vectorizer = CountVectorizer(stop_words='english')

# Fit and transform the processed titles
count_data = count_vectorizer.fit_transform(tweets['tweets_text_processed'] )

# Visualise the 10 most common words
plot_10_most_common_words(count_data, count_vectorizer)
```



Observation: The bar chart above is the bar representation if what we had before in the would cloud, this is detailed in the sence that, we would not only see the most occuring words, but we would also see it frequency (the number of times it appears). From the chart, it is obvious that the word male occur more frequencly than that of female. The frequency of count of the word male is above 12000, while that of female is slightly above 2000. Could this be a gender bias in terms of job opportunities? This would be an interesting topic to look into.

[66]:
```
import warnings
warnings.simplefilter("ignore", DeprecationWarning)
# Load the LDA model from sk-learn
from sklearn.decomposition import LatentDirichletAllocation as LDA

# Helper function
def print_topics(model, count_vectorizer, n_top_words):
    words = count_vectorizer.get_feature_names()
    for topic_idx, topic in enumerate(model.components_):
        print("\nTopic #%d:" % topic_idx)
        print(" ".join([words[i]
                        for i in topic.argsort()[:-n_top_words - 1:-1]]))

# Tweak the two parameters below
number_topics = 10
```

```python
number_words = 20

# Create and fit the LDA model
lda = LDA(n_components=number_topics, n_jobs=-1)
lda.fit(count_data)
# Print the topics found by the LDA model
print("Topics found via LDA:")
print_topics(lda, count_vectorizer, number_words)
```

```
Topics found via LDA:

Topic #0:
job male https available offer pictwittercom http online female apply men
position status twittercom new women time day opening 2020

Topic #1:
male jobs job female https 2020 work pictwittercom day workers offer available
health need like pay hours police people working

Topic #2:
job male https status twittercom female jobs work best pictwittercom white time
good woman looking available like person man just

Topic #3:
job male offer nurses available female home 80 sir nursing work start 20 remote
time pmoindia save_male_nurses got today http

Topic #4:
male job white female just like don good women people doing jobs know woman best
think got man person ve

Topic #5:
male job women female jobs men best people white gender person black pay just
don dominated https equal paid woman

Topic #6:
job male jobs status twittercom https available people offer pictwittercom old
help lockdown just man 50 100 like white period

Topic #7:
male jobs female job https apply http looking years 2020 experience english
hiring assistant required location teachers sales saudi salary

Topic #8:
available offer job jobs offers https pictwittercom http new time career male
help need opportunities looking apply free bitly positions
```

Topic #9:
male job women work jobs like time status https twittercom want make woman men don sex working right really dominated

Observation: From the above LDA analysis, we generated 10 different topics,with 15 words each, these two parameters can be tweak according to out needs, but from what we have displayed, it is obvious that the last topic #3 is about male nurse neede, this is so interesting, as jobs like that were mostly allocated to females. Topic #7 is about hiring an experienced female English teacher at Saudi Arabia. Topic #9 is probaly talking about the sex work being dominated by females.

Conclusion: We have been able to scrap tweets from the first day of January up til the 20 day of Juneseen the various tweets from people around the world as regards male and female job oppotunities and postions, from the EDA, it was evidence that the word male occured most in the tweets than that of female. Hence, aside for the topic modelling, further analysis needs to be carried out on this data, as there are still some interesting findings to get from it.