ROLL NO – 162010022

PRACTICAL – 7
TREND ANALYSIS

AIM:- To identifying the trend in the given dataset for different entities like people, organizations etc

In [103]:
```
import lda
from sklearn.feature_extraction.text import CountVectorizer
```

In [104]:
```
import logging
logging.getLogger("lda").setLevel(logging.WARNING)
```

In [105]:
```
cvectorizer = CountVectorizer(min_df=10, max_features=10000, tokenizer=tokenizer, ngram_range=(1,2))
cvz = cvectorizer.fit_transform(df['summary'])

n_topics = 20
n_iter = 2000
lda_model = lda.LDA(n_topics=n_topics, n_iter=n_iter)
X_topics = lda_model.fit_transform(cvz)
```

```
WARNING:lda:all zero row in document-term matrix found
```

In [106]:
```
n_top_words = 8
topic_summaries = []

topic_word = lda_model.topic_word_  # get the topic words
vocab = cvectorizer.get_feature_names()
for i, topic_dist in enumerate(topic_word):
    topic_words = np.array(vocab)[np.argsort(topic_dist)][:-(n_top_words+1):-1]
    topic_summaries.append(' '.join(topic_words))
    print('Topic {}: {}'.format(i, ' '.join(topic_words)))
```

```
Topic 0: one said people like time even would also
Topic 1: party yadav akhilesh mulayam samajwadi singh samajwadi party chief
Topic 2: court supreme supreme court case justice said order high
Topic 3: appeared post appeared first first said firstpost state first firstpost
Topic 4: rs crore bank rs crore tax said banks india
Topic 5: minister said state government chief chief minister meeting jallikattu
Topic 6: said india china world countries global would new
Topic 7: minister prime prime minister modi said narendra narendra modi minister narendra
Topic 8: party congress bjp assembly said election elections chief
Topic 9: percent us index points stocks ket dollar kets
Topic 10: indian uary students th india school education event
Topic 11: said government state development project water also power
Topic 12: police said district two arrested incident police said today
Topic 13: first india team cricket test second match captain
Topic 14: per rs cent per cent rs per prices ket today
Topic 15: air said degrees airport two due celsius last
Topic 16: us trump president said donald presidentelect donald trump presidentelect donald
Topic 17: india new company first said mobile million technology
Topic 18: film actor khan actress star films movie also
Topic 19: company uary bse ltd board tata quarter meeting
```

In [114]:
```
tsne_lda = tsne_model.fit_transform(X_topics)
```

In [ ]:
```
tsne_lda
```

In [116]:
```
doc_topic = lda_model.doc_topic_
lda_keys = []
for i, tweet in enumerate(df['summary']):
    lda_keys += [doc_topic[i].argmax()]
```

In [117]:
```
plot_lda = bp.figure(plot_width=700, plot_height=600, title="LDA topic visualization",
    tools="pan,wheel_zoom,box_zoom,reset,hover,previewsave",
    x_axis_type=None, y_axis_type=None, min_border=1)
```

In [118]:
```
lda_df = pd.DataFrame(tsne_lda, columns=['x','y'])
lda_df['summary'] = df['summary']
```

In [119]:
```
lda_df['topic'] = lda_keys
lda_df['topic'] = lda_df['topic'].map(int)
```

In [128]:
```
ds1 = ColumnDataSource(data=dict(x=lda_df['x'],
                                 y=lda_df['y'],
                                 color=list(colormap[lda_df['topic']]))))
```

In [129]:
```
plot_lda.scatter(x='x', y='y',color='color'
                 ,source=ds1)
hover = plot_lda.select(dict(type=HoverTool))
hover.tooltips={"description": "@summary", "topic": "@topic"}
show(plot_lda)
```

In [ ]:
```
lda_df
```

In [131]:
```
topicsName={}
topicsName[0]='general'
topicsName[1]='politics'
topicsName[2]='government'
topicsName[3]='general'
topicsName[4]='economy'
topicsName[5]='government'
topicsName[6]='international'
topicsName[7]='politics'
topicsName[8]='politics'
topicsName[9]='economy'
topicsName[10]='education'
topicsName[11]='development'
topicsName[12]='crime'
topicsName[13]='sports'
topicsName[14]='economy'
topicsName[15]='general'
topicsName[16]='politics'
topicsName[17]='technology'
topicsName[18]='bollywood'
topicsName[19]='business'
def generateTopics(number):
    return topicsName[number]
```

In [132]:
```
df['topics']=lda_df['topic'].apply(lambda x:generateTopics(x))
```

In [133]:
```
df[['summary','topics']].head()
```

Out[133]:

	summary	topics
0	The Army and the Air Force got new chiefs on ...	politics
1	Utility needs crore to pay salary two ths pen...	development
2	More than people on board a bus operated by ...	crime
3	The year saw a diverse range of movies being r...	bollywood
4	Beijing China will ban the processing and sale...	international

In [134]:
```
t_NumTopics=df['topics'].values
```

In [135]:
```
countTrend=Counter(t_NumTopics)
```

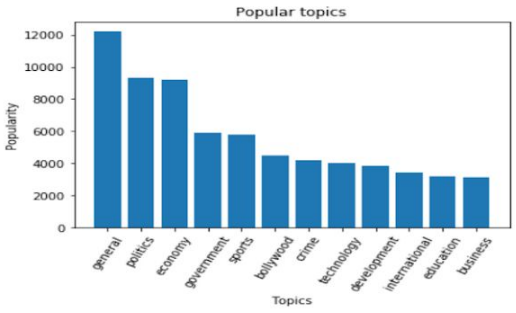
In [136]:
```
countTrend
```

Out[136]:
```
Counter({'bollywood': 4468,
         'business': 3139,
         'crime': 4209,
         'development': 3814,
         'economy': 9193,
         'education': 3159,
         'general': 12227,
         'government': 5879,
         'international': 3418,
         'politics': 9311,
         'sports': 5756,
         'technology': 3984})
```

In [137]:
```
countTrend=countTrend.most_common()
```

In [138]:
```
x_topics=[i[0] for i in countTrend[:]]
y_values=[i[1] for i in countTrend[:]]
```

In [139]:
```
import matplotlib.pyplot as plt
import numpy as np
index=np.arange(len(x_topics))
plt.bar(index, y_values)
plt.xlabel('Topics', fontsize=10)
plt.ylabel('Popularity', fontsize=10)
plt.xticks(index, x_topics, fontsize=10, rotation=60)
plt.title('Popular topics')
plt.show()
```



In [144]:
```
def getDay(text):
    f='%Y-%m-%d'
    text=str(text).split(' ')[0]
    if '$' in text:
        text=text[1:]
    currdate=datetime.datetime.strptime(text, f).date()
    return currdate.day
```

In [145]:
```
def getMonth(text):
    f='%Y-%m-%d'
    text=str(text).split(' ')[0]
    if '$' in text:
        text=text[1:]
    currdate=datetime.datetime.strptime(text, f).date()
    return currdate.month
```

In [146]:
```
def getYear(text):
    f='%Y-%m-%d'
    text=str(text).split(' ')[0]
    if '$' in text:
        text=text[1:]
    currdate=datetime.datetime.strptime(text, f).date()
    return currdate.year
```

In [147]:
```
df['day']=df['Date'].apply(lambda x:getDay(x))
df['month']=df['Date'].apply(lambda x:getMonth(x))
df['year']=df['Date'].apply(lambda x:getYear(x))
```

In [148]:
```
df[['Date','day','month','year']].head()
```

Out[148]:

	Date	day	month	year
0	2017-01-01 00:00:00	1	1	2017
1	2017-01-01 00:00:00	1	1	2017
2	2017-01-01 00:00:00	1	1	2017
3	2017-01-01 00:00:00	1	1	2017

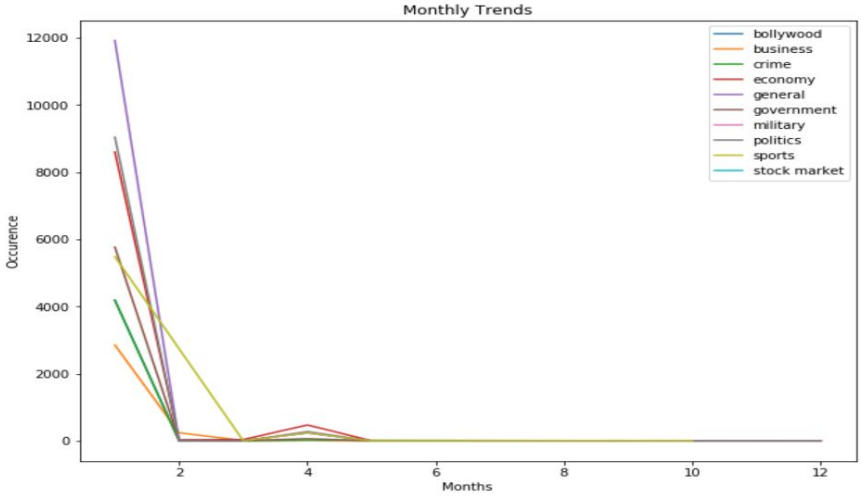
In [149]:
```
g1=pd.DataFrame({'count' : df.groupby( [ "topics", "month"] ).size()}).reset_index()
```

In [150]:
```
x1=g1[g1['topics']=='bollywood']['month'].values
x2=g1[g1['topics']=='business']['month'].values
x3=g1[g1['topics']=='crime']['month'].values
x4=g1[g1['topics']=='economy']['month'].values
x5=g1[g1['topics']=='general']['month'].values
x6=g1[g1['topics']=='government']['month'].values
x7=g1[g1['topics']=='military']['month'].values
x8=g1[g1['topics']=='politics']['month'].values
x9=g1[g1['topics']=='sports']['month'].values
x10=g1[g1['topics']=='stock market']['month'].values


y1=g1[g1['topics']=='bollywood']['count'].values
y2=g1[g1['topics']=='business']['count'].values
y3=g1[g1['topics']=='crime']['count'].values
y4=g1[g1['topics']=='economy']['count'].values
y5=g1[g1['topics']=='general']['count'].values
y6=g1[g1['topics']=='government']['count'].values
y7=g1[g1['topics']=='military']['count'].values
y8=g1[g1['topics']=='politics']['count'].values
y9=g1[g1['topics']=='sports']['count'].values
y10=g1[g1['topics']=='stock market']['count'].values

plt.plot(x1,y1,label='bollywood')
plt.plot(x2,y2,label='business')
plt.plot(x3,y3,label='crime')
plt.plot(x4,y4,label='economy')
plt.plot(x5,y5,label='general')
plt.plot(x6,y6,label='government')
plt.plot(x7,y7,label='military')
plt.plot(x8,y8,label='politics')
plt.plot(x9,y9,label='sports')
plt.plot(x10,y10,label='stock market')
```

Out[150]: <matplotlib.legend.Legend at 0x5e77d710>



In [151]:
```
listPerson=df['PERSON'].values.tolist()
```

In [152]:
```
listPerson1=list(filter(bool,listPerson))
```

In [153]:
```
listPerson1=','.join(listPerson1)
```

In [154]:
```
listPerson1=listPerson1.replace(',Modi',',Narendra Modi')
```

In [155]:
```
listPerson1=listPerson1.replace(',Narendra Modis',',Narendra Modi')
```

In [156]:
```
listPerson1=listPerson1.replace(',Date','')
```

In [157]:
```
listPerson1=listPerson1.replace(',Jan','')
```

In [158]:
```
listPerson1=listPerson1.replace(',Yadav',',Akhilesh Yadav')
listPerson1=listPerson1.replace(',Singh',',Mulayam Singh')
```

In [159]:
```
new_listPerson1=""
iterator=listPerson1.split(',')
for word in iterator:
    if len(word.split(' '))==1:
        for existing_word in iterator:
            if word in existing_word:
                word=existing_word
                new_listPerson1+=word+","
    elif word[-1]=='s':
        word=word[:-1]
        new_listPerson1+=word+","
    else:
        new_listPerson1+=word+","
```

In [ ]:
```
new_listPerson1
```

In [160]:
```
new_listPerson1=new_listPerson1.replace(',Embargo Syndicate Hide',"")
```

In [161]:
```
new_listPerson1=new_listPerson1.replace(',Donald Trumps',",Donald Trump")
```

In [162]:
```
new_listPerson1=new_listPerson1.replace(',Narendra Modis',",Narendra Modi")
```

In [163]:
```
new_listPerson1=new_listPerson1.replace(',Barack Obamas',",Barack Obama")
```

In [164]:
```
new_listPerson1=new_listPerson1.replace(',Kejriwals',",Arvind Kejriwal")
```

In [165]:
```
new_listPerson1=new_listPerson1.replace(',Arvind Kejriwals',",Arvind Kejriwal")
```

In [173]:
```
new_listPerson1=new_listPerson1.replace(',Article Images Short',",")
```

In [174]:
```
new_listPerson1=new_listPerson1.replace(',Mahendra Singh Dhonis',",Mahendra Singh Dhoni")
```

In [219]:
```
counter = Counter(new_listPerson1.split(','))
```

In [220]:
```
new_count=counter.most_common(20)
```

In [223]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [224]:
```
persons=[i[0] for i in new_count[:10]]
values=[i[1] for i in new_count[:10]]
```

In [223]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [224]:
```
persons=[i[0] for i in new_count[:10]]
values=[i[1] for i in new_count[:10]]
```

In [228]:
```
index=np.arange(len(persons))
plt.bar(index, values)
plt.xlabel('Persons', fontsize=10)
plt.ylabel('Word Count', fontsize=10)
plt.xticks(index, persons, fontsize=10, rotation=70)
plt.title('Popular persons')
plt.show()
```