In [59]:

```
import os
import pandas as pd
import numpy as np
import re
import string
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
nltk.download('wordnet')
from autocorrect import Speller
spell = Speller(lang='en')
import warnings
warnings.filterwarnings("ignore")
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Embedding
from scipy.spatial import distance
from sklearn.metrics.pairwise import cosine_similarity
import spacy
import numpy as np
from numpy import dot
from numpy.linalg import norm
from sklearn.cluster import DBSCAN
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import MiniBatchKMeans
from sklearn.metrics import davies_bouldin_score
```

```
[nltk_data] Error loading stopwords: <urlopen error [Errno 8] nodename
[nltk_data]     nor servname provided, or not known>
[nltk_data] Error loading wordnet: <urlopen error [Errno 8] nodename
[nltk_data]     nor servname provided, or not known>
```

In [28]:

```
company_desc=pd.read_excel('/Users/sandeeppydi/Desktop/nlp/Company Descriptions.xlsx')
```

In [29]:

```
#Data Preprocessing of company_desc
#following standard steps for preprocessing
#Lower casing
#Removal of Punctuations
#Removal of Numbers
#Removal of Stopwords
#Lemmatization
#Removal of URLs
#Removal of HTML tags
#removing white spaces
#stripping
#sorting sentence
```

In [30]:

```
#going to work on short desription
data=company_desc.company_short_description
```

In [31]:

```
#Lower casing
data=data.str.lower()
```

In [32]:

```
#Removal of Punctuations
```

```
#Removal of Punctuations
def punct(s):
    return s.translate(str.maketrans('','',string.punctuation))
```

In [33]:

```
data=data.apply(punct)
```

In [34]:

```
#Special case for double quote characters not removed from above function
def punct_(s):
    return re.sub('[“”]','',s)
```

In [35]:

```
data=data.apply(punct_)
```

In [36]:

```
#Removal of Numbers
def numbers_remove(s):
    return re.sub('\d+','',s)
```

In [37]:

```
data=data.apply(numbers_remove)
```

In [38]:

```
#Removal of Stopwords
#remove stop words
stop=list(stopwords.words('english'))
def remove_stopwords(x):
    l=x.split(' ')
    m=[]
    for i in l:
        if i not in stop:
            m.append(i)
    return ' '.join(m)
```

In [39]:

```
data=data.apply(remove_stopwords)
```

In [40]:

```
#lemitization
w=WordNetLemmatizer()
def lemitization(x):
    m=[]
    l=re.split(' ',x)
    for i in l:
        m.append(w.lemmatize(i))
    return ' '.join(m)
```

In [41]:

```
data=data.apply(lemitization)
```

In [42]:

```
#remove urls
def remove_url(x):
    return re.sub('https?\S*\s*','',x)
```

In [43]:

```
data=data.apply(remove_url)
```

In [44]:

```
#remove html tags
def remove_html(x):
    return re.sub('<.*?>','',x)
```

In [45]:

```
data=data.apply(remove_html)
```

In [46]:

```
##removing white spaces
def remove_white_spaces(x):
    s=re.split('\s+',x)
    return ' '.join(s)
```

In [47]:

```
data=data.apply(remove_white_spaces)
```

In [48]:

```
#strip
def strip(s):
    return s.strip()
```

In [49]:

```
data=data.apply(strip)
```

In [50]:

```
#TFIDF vectorizer
v1=TfidfVectorizer()
```

In [51]:

```
sent_vec=v1.fit_transform(data).toarray()
```

In [70]:

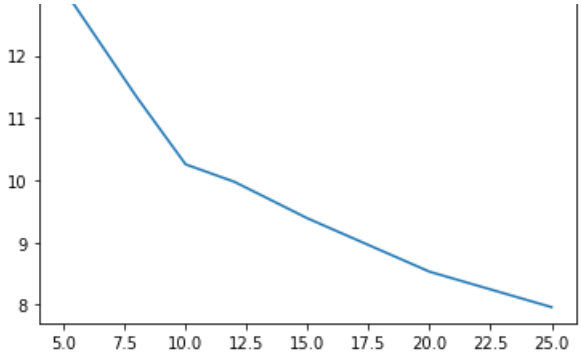
```
#try with differernt number of clusters
clusters=[5,8,10,12,15,20,25]
score=[]
label_list=[]
for i in l:
    d=MiniBatchKMeans(n_clusters=i,init_size=1000,batch_size=1000).fit(sent_vec)
    y=davies_bouldin_score(sent_vec,d.labels_)
    score.append(y)
    label_list.append(d.labels_)
```

In [71]:

```
import matplotlib.pyplot as plt
plt.plot(clusters,score)
```

Out[71]:

```
[<matplotlib.lines.Line2D at 0x11ddb5c90>]
```



In [ ]:

```
#score is minumum when cluster number is 25 where davies_bouldin_score is low
```

In [73]:

```
labels_=pd.Series((label_list[-1]))
```

In [74]:

```
company_desc['lables']=labels_
```

In [76]:

```
company_desc=company_desc.drop(columns=['company_short_description','company_description'])
```

In [78]:

```
company_desc.to_excel("prog_4_cluster.xlsx")
```

In [ ]:

```

```