In [3]:
```
import pandas as pd
import numpy as np
```

In [4]:
```
df = pd.read_csv('UpdatedResumeDataSet.csv')
```

In [5]:
```
df.head()
```

Out[5]:

	Category	Resume
0	Data Science	Skills * Programming Languages: Python (pandas...
1	Data Science	Education Details \r\nMay 2013 to May 2017 B.E...
2	Data Science	Areas of Interest Deep Learning, Control Syste...
3	Data Science	Skills â¢ R â¢ Python â¢ SAP HANA â¢ Table...
4	Data Science	Education Details \r\n MCA YMCAUST, Faridab...

In [6]:
```
df['Category'].value_counts()
```

Out[6]:
```
Java Developer               84
Testing                      70
DevOps Engineer              55
Python Developer             48
Web Designing                45
HR                           44
Hadoop                       42
Blockchain                   40
ETL Developer                40
Operations Manager           40
Data Science                 40
Sales                        40
Mechanical Engineer          40
Arts                         36
Database                     33
Electrical Engineering       30
Health and fitness           30
PMO                          30
Business Analyst             28
DotNet Developer             28
Automation Testing           26
Network Security Engineer    25
SAP Developer                24
Civil Engineer               24
Advocate                     20
Name: Category, dtype: int64
```

In [7]:
```
df['Category'].nunique()
```

Out[7]:
```
25
```

In [ ]:
```

```

In [8]:
```
df.isnull()
```

Out[8]:

	Category	Resume
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
...	...	...
957	False	False
958	False	False
959	False	False
960	False	False
961	False	False

962 rows × 2 columns

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Category    0
Resume      0
dtype: int64
```

In [10]:
```
df.duplicated().sum()
```

Out[10]:
```
796
```

In [11]:
```
duplicate_rows =df.duplicated(subset=['Resume', 'Category'], keep=False)
```

In [12]:
```
print(duplicate_rows)
```

```
0      True
1      True
2      True
3      True
4      True
       ...
957    True
958    True
959    True
960    True
961    True
Length: 962, dtype: bool
```

In [13]:
```
duplicateRows = df[~duplicate_rows]
```

In [14]:
```
print(duplicateRows)
```

```
            Category                                             Resume
602  DevOps Engineer  Technical Skills Key Skills MS Technology .Net...
603  DevOps Engineer  Core skills â¢ Project / Program Management â...
604  DevOps Engineer  Total IT Experience 15 years. Core expertise i...
605  DevOps Engineer  TECHNICAL SKILLS â¢ HP ALM, RTC and JIRA â¢ ...
```

In [15]:
```
df.dropna().iloc[0]
```

Out[15]:
```
Category                                        Data Science
Resume      Skills * Programming Languages: Python (pandas...
Name: 0, dtype: object
```

In [16]:
```
df['Resume'].iloc[0]
```

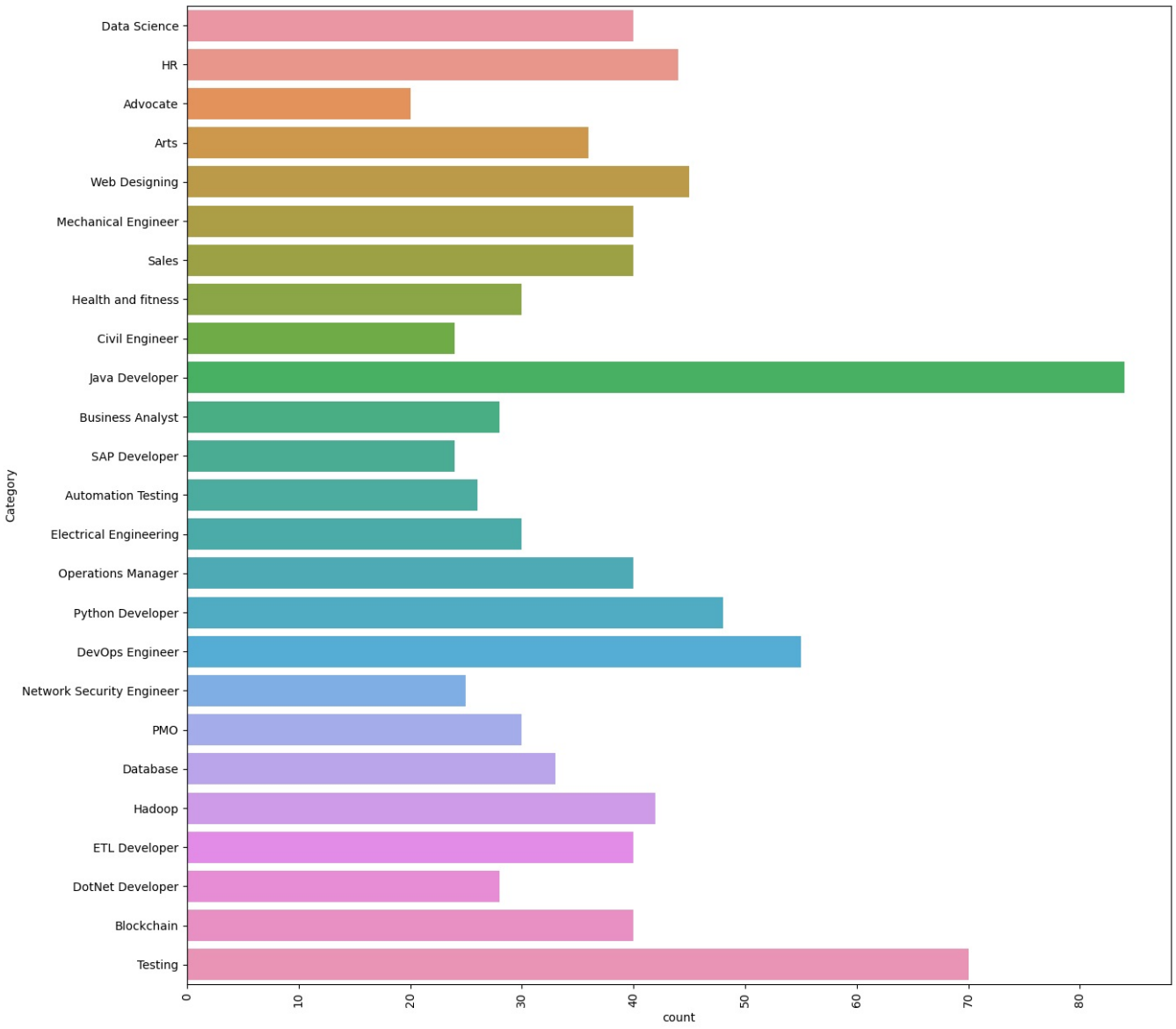
Out[16]:
```
'Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript
/JQuery. * Machine learning: Regression, SVM, NaÃ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techni
ques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language processing, Dimensionality reducti
on, Topic Modelling (LDA, NMF), PCA & Neural Nets. * Database Visualizations: Mysql, SqlServer, Cassandra, Hbas
e, ElasticSearch D3.js, DC.js, Plotly, kibana, matplotlib, ggplot, Tableau. * Others: Regular Expression, HTML,
CSS, Angular 6, Logstash, Kafka, Python Flask, Git, Docker, computer vision - Open CV and understanding of Deep
learning.Education Details \r\n\r\nData Science Assurance Associate \r\n\r\nData Science Assurance Associate -
Ernst & Young LLP\r\nSkill Details \r\nJAVASCRIPT- Exprience - 24 months\r\njQuery- Exprience - 24 months\r\nPy
thon- Exprience - 24 monthsCompany Details \r\ncompany - Ernst & Young LLP\r\ndescription - Fraud Investigation
s and Dispute Services   Assurance\r\nTECHNOLOGY ASSISTED REVIEW\r\nTAR (Technology Assisted Review) assists in
accelerating the review process and run analytics and generate reports.\r\n* Core member of a team helped in de
veloping automated review platform tool from scratch for assisting E discovery domain, this tool implements pre
dictive coding and topic modelling by automating reviews, resulting in reduced labor costs and time spent durin
g the lawyers review.\r\n* Understand the end to end flow of the solution, doing research and development for c
lassification models, predictive analysis and mining of the information present in text data. Worked on analyzi
ng the outputs and precision monitoring for the entire tool.\r\n* TAR assists in predictive coding, topic model
ling from the evidence by following EY standards. Developed the classifier models in order to identify "red fla
gs" and fraud-related issues.\r\n\r\nTools & Technologies: Python, scikit-learn, tfidf, word2vec, doc2vec, cosi
ne similarity, NaÃ¯ve Bayes, LDA, NMF for topic modelling, Vader and text blob for sentiment analysis. Matplot
lib, Tableau dashboard for reporting.\r\n\r\nMULTIPLE DATA SCIENCE AND ANALYTIC PROJECTS (USA CLIENTS)\r\nTEXT
ANALYTICS - MOTOR VEHICLE CUSTOMER REVIEW DATA * Received customer feedback survey data for past one year. Perf
ormed sentiment (Positive, Negative & Neutral) and time series analysis on customer comments across all 4 categ
ories.\r\n* Created heat map of terms by survey category based on frequency of words * Extracted Positive and N
egative words across all the Survey categories and plotted Word cloud.\r\n* Created customized tableau dashboar
ds for effective reporting and visualizations.\r\nCHATBOT * Developed a user friendly chatbot for one of our Pr
oducts which handle simple questions about hours of operation, reservation options and so on.\r\n* This chat bo
t serves entire product related questions. Giving overview of tool via QA platform and also give recommendation
responses so that user question to build chain of relevant answer.\r\n* This too has intelligence to build the
pipeline of questions as per user requirement and asks the relevant /recommended questions.\r\n\r\nTools & Tech
nologies: Python, Natural language processing, NLTK, spacy, topic modelling, Sentiment analysis, Word Embedding
, scikit-learn, JavaScript/JQuery, SqlServer\r\n\r\nINFORMATION GOVERNANCE\r\nOrganizations to make informed de
cisions about all of the information they store. The integrated Information Governance portfolio synthesizes in
telligence across unstructured data sources and facilitates action to ensure organizations are best positioned
to counter information risk.\r\n* Scan data from multiple sources of formats and parse different file formats,
extract Meta data information, push results for indexing elastic search and created customized, interactive das
hboards using kibana.\r\n* Preforming ROT Analysis on the data which give information of data which helps ident
ify content that is either Redundant, Outdated, or Trivial.\r\n* Preforming full-text search analysis on elasti
c search with predefined methods which can tag as (PII) personally identifiable information (social security nu
mbers, addresses, names, etc.) which frequently targeted during cyber-attacks.\r\nTools & Technologies: Python,
Flask, Elastic Search, Kibana\r\n\r\nFRAUD ANALYTIC PLATFORM\r\nFraud Analytics and investigative platform to r
eview all red flag cases.\r\nâ\x80¢ FAP is a Fraud Analytics and investigative platform with inbuilt case manag
er and suite of Analytics for various ERP systems.\r\n* It can be used by clients to interrogate their Accounti
ng systems for identifying the anomalies which can be indicators of fraud by running advanced analytics\r\nTool
s & Technologies: HTML, JavaScript, SqlServer, JQuery, CSS, Bootstrap, Node.js, D3.js, DC.js'
```

In [17]:
```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt

plt.figure(figsize=(15,15))
plt.xticks(rotation=90)
```

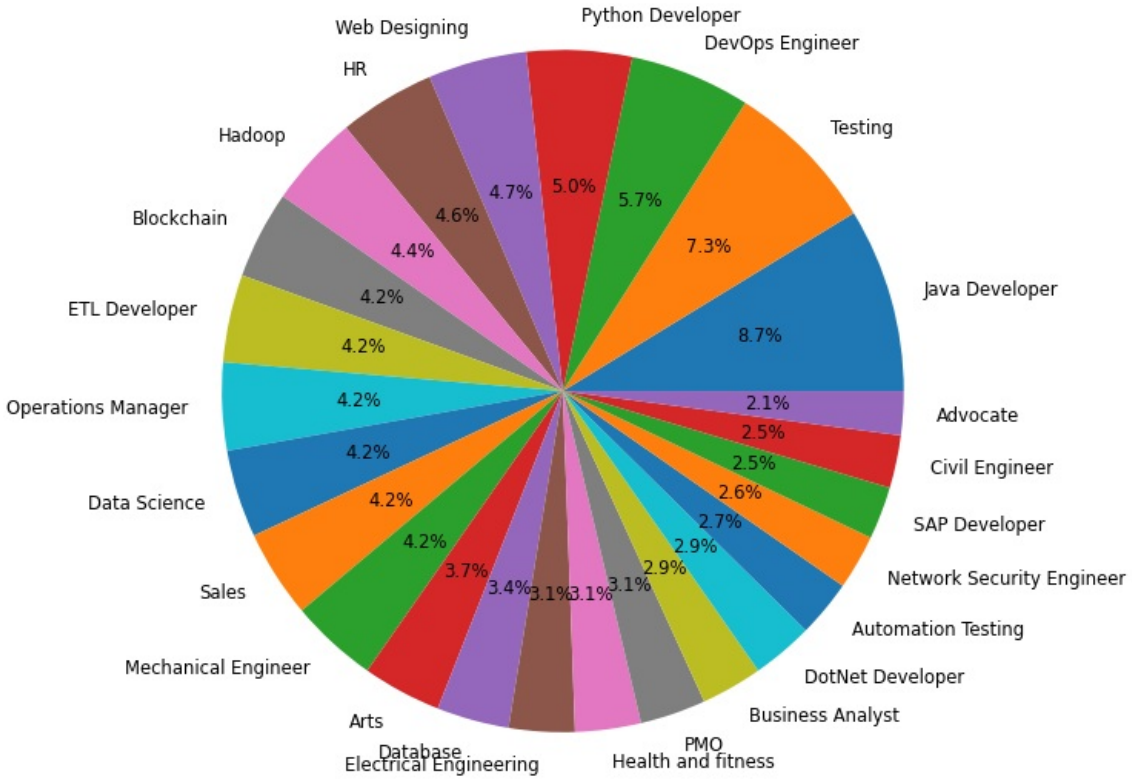
```
sns.countplot(y="Category", data=df)

#Pie-chart
targetCounts = df['Category'].value_counts().reset_index()['Category']
targetLabels  = df['Category'].value_counts().reset_index()['index']
```



In [18]:
```
fig, ax = plt.subplots()

ax.pie(targetCounts, labels=targetLabels, autopct='%1.1f%%',
       textprops={'size': 'smaller'}, radius=1.5)
plt.show()
```



In [19]:
```
import re
import warnings
warnings.filterwarnings('ignore')
def cleanResume(dfText):
    dfText = re.sub(r'https?://\S+|www\.\S+', ' ', dfText)  # remove URLs
    dfText = re.sub(r'RT|cc', ' ', dfText)  # remove RT and cc
    dfText = re.sub(r'#S+', '', dfText)  # remove hashtags
    dfText = re.sub(r'@\S+', '  ', dfText)  # remove mentions
    dfText = re.sub(r'[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[]^_`{|}~"""), ' ', dfText)  # remove punctuat
    dfText = re.sub(r'[^x00-x7f]',r' ', dfText)
    dfText = re.sub(r's+', ' ', dfText)  # remove extra whitespace
    return dfText

df['cleaned_resume'] = df['Resume'].apply(lambda x: cleanResume(x))
```

In [20]:
```
print(df['cleaned_resume'])
```

```
0      Skill    Programming Language   P thon  panda ...
1      Education Detail    Ma  2013 to Ma  2017 B E  ...
2      Area  of Intere t Deep Learning  Control S  te...
3      Skill      R     P thon     SAP HANA     Table...
4      Education Detail     MCA   YMCAUST   Faridabad...
                              ...
957    Computer Skill       Proficient in MS office  ...
958        Willingne  to a ept the challenge       Po...
959    PERSONAL SKILLS     Quick learner      Eagerne...
960    COMPUTER SKILLS   SOFTWARE KNOWLEDGE MS Power ...
961    Skill Set OS Window  XP 7 8 8 1 10 Databa e MY...
Name: cleaned_resume, Length: 962, dtype: object
```

In [21]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Category'] = le.fit_transform(df['Category'])
```

In [22]:
```
print(df['Category'])
```

```
0       6
1       6
2       6
3       6
4       6
       ..
957    23
958    23
959    23
960    23
961    23
Name: Category, Length: 962, dtype: int32
```

In [23]:
```
print(df['Category'], le.classes_)
```

```
0       6
1       6
2       6
3       6
4       6
       ..
957    23
958    23
959    23
960    23
961    23
Name: Category, Length: 962, dtype: int32 ['Advocate' 'Arts' 'Automation Testing' 'Blockchain' 'Business Analys
t'
 'Civil Engineer' 'Data Science' 'Database' 'DevOps Engineer'
 'DotNet Developer' 'ETL Developer' 'Electrical Engineering' 'HR' 'Hadoop'
 'Health and fitness' 'Java Developer' 'Mechanical Engineer'
 'Network Security Engineer' 'Operations Manager' 'PMO' 'Python Developer'
 'SAP Developer' 'Sales' 'Testing' 'Web Designing']
```

In [24]:
```
print('Category')
```

```
Category
```

In [27]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words=['the', 'and', 'of', ...], max_features=1500, sublinear_tf=True)
requiredText  = df['cleaned_resume'].values
WordFeatures = tfidf_vectorizer.fit_transform(requiredText)
```

In [28]:
```
print("TF-IDF Matrix Shape:", WordFeatures.shape)
```

```
TF-IDF Matrix Shape: (962, 1500)
```

In [29]:
```
requiredTarget = df['Category'].values
```

In [30]:
```
print(df['Category'].values)
```

```
[ 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6 12 12 12 12 12 12 12 12
 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
 12 12 12 12 12 12 12 12 12 12 12 12  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 24 24 24 24
 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 16 16 16 16 16 16 16 16
 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
 16 16 16 16 16 16 16 16 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
 22 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
 14 14 14 14 14 14 14  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5
  5  5  5  5  5  5  5 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15  4  4  4  4  4
  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4 21
 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21  2
  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
  2 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
 11 11 11 11 11 11 11 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 20
 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  8
  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8
  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8
  8  8  8  8  8  8 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
 17 17 17 17 17 17 17 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19
 19 19 19 19 19 19 19 19 19 19 19 19 19  7  7  7  7  7  7  7  7  7  7  7
  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7 13 13 13
 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 10 10 10 10 10  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
  9  9  9  9  9  9  9  9  9  9  9  9  3  3  3  3  3  3  3  3  3  3  3  3
  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
  3  3  3  3 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
 23 23]
```

In [31]:
```
requiredText  = df['cleaned_resume'].values
```

In [32]:
```
print("Values of requiredText", requiredText[0])
```

```
Values of requiredText Skill    Programming Language   P thon  panda   nump    cip    cikit learn  matplotlib
Sql  Java  JavaScript JQuer     Machine learning  Regre ion  SVM  Na  ve Ba e   KNN  Random Fore t  Deci ion Tr
ee   Boo ting technique   Clu ter Anal  i   Word Embedding  Sentiment Anal  i   Natural Language proce ing  Dim
en ionalit  reduction  Topic Modelling  LDA  NMF   PCA   Neural Net     Databa e Vi uali ation   M  ql  SqlServ
er  Ca andra  Hba e  Ela ticSearch D3 j   DC j   Plotl   kibana  matplotlib  ggplot  Tableau    Other   Regular
Expre ion  HTML  CSS  Angular 6  Log ta h  Kafka  P thon Fla k  Git  Docker  computer vi ion   Open CV and unde
r tanding of Deep learning Education Detail      Data Science A urance A ociate     Data Science A urance A oci
ate   Ern t   Young LLP  Skill Detail    JAVASCRIPT  Exprience   24 month   jQuer   Exprience   24 month   P th
on  Exprience   24 month Compan  Detail    compan    Ern t   Young LLP  de cription   Fraud Inve tigation  and
Di pute Service    A urance  TECHNOLOGY ASSISTED REVIEW  TAR  Technolog  A i ted Review  a i t  in a elerating
the review proce  and run anal tic  and generate report      Core member of a team helped in developing automat
ed review platform tool from  cratch for a i ting E di cover  domain  thi  tool implement  predictive coding an
d topic modelling b  automating review   re ulting in reduced labor co t  and time  pent during the law er  rev
iew     Under tand the end to end flow of the  olution  doing re earch and development for cla ification model
predictive anal  i  and mining of the information pre ent in text data  Worked on anal  ing the output  and pre
ci ion monitoring for the entire tool     TAR a i t  in predictive coding  topic modelling from the evidence b
following EY  tandard   Developed the cla ifier model  in order to identif   red flag   and fraud related i ue
Tool    Technologie   P thon   cikit learn  tfidf  word2vec  doc2vec  co ine  imilarit   Na  ve Ba e   LDA  NMF
for topic modelling  Vader and text blob for  entiment anal  i   Matplot lib  Tableau da hboard for reporting
MULTIPLE DATA SCIENCE AND ANALYTIC PROJECTS  USA CLIENTS   TEXT ANALYTICS   MOTOR VEHICLE CUSTOMER REVIEW DATA
Received cu tomer feedback  urve  data for pa t one  ear  Performed  entiment  Po itive  Negative   Neutral  an
d time  erie  anal  i  on cu tomer comment  acro  all 4 categorie      Created heat map of term  b   urve  cate
gor  ba ed on frequenc  of word    Extracted Po itive and Negative word  acro  all the Surve  categorie  and pl
otted Word cloud     Created cu tomi ed tableau da hboard  for effective reporting and vi uali ation    CHATBOT
Developed a u er friendl  chatbot for one of our Product  which handle  imple que tion  about hour  of operatio
n  re ervation option  and  o on     Thi  chat bot  erve  entire product related que tion   Giving overview of
tool via QA platform and al o give recommendation re pon e   o that u er que tion to build chain of relevant an
wer     Thi  too ha  intelligence to build the pipeline of que tion  a  per u er requirement and a k  the relev
ant  recommended que tion      Tool    Technologie   P thon  Natural language proce ing  NLTK   pac   topic mod
elling  Sentiment anal  i   Word Embedding   cikit learn  JavaScript JQuer   SqlServer    INFORMATION GOVERNANC
E  Organi ation  to make informed deci ion  about all of the information the   tore  The integrated Information
Governance portfolio   nthe i e  intelligence acro  un tructured data  ource  and facilitate  action to en ure
organi ation  are be t po itioned to counter information ri k     Scan data from multiple  ource  of format  an
d par e different file format   extract Meta data information  pu h re ult  for indexing ela tic  earch and cre
ated cu tomi ed  interactive da hboard  u ing kibana     Preforming ROT Anal  i  on the data which give informa
tion of data which help  identif  content that i  either Redundant  Outdated  or Trivial     Preforming full te
xt  earch anal  i  on ela tic  earch with predefined method  which can tag a   PII  per onall  identifiable inf
ormation   ocial  ecurit  number   addre e   name   etc   which frequentl  targeted during c ber attack    Tool
Technologie   P thon  Fla k  Ela tic Search  Kibana    FRAUD ANALYTIC PLATFORM  Fraud Anal tic  and inve tigati
ve platform to review all red flag ca e        FAP i  a Fraud Anal tic  and inve tigative platform with inbuilt
ca e manager and  uite of Anal tic  for variou  ERP    tem      It can be u ed b  client  to interrogate their
A ounting    tem  for identif ing the anomalie  which can be indicator  of fraud b  running advanced anal tic
Tool    Technologie   HTML  JavaScript  SqlServer  JQuer   CSS  Boot trap  Node j   D3 j   DC j
```

In [33]:
```
print(df)
```

```
     Category                                             Resume  \
0           6  Skills * Programming Languages: Python (pandas...
1           6  Education Details \r\nMay 2013 to May 2017 B.E...
2           6  Areas of Interest Deep Learning, Control Syste...
3           6  Skills â¢ R â¢ Python â¢ SAP HANA â¢ Table...
4           6  Education Details \r\n MCA   YMCAUST,  Faridab...
..        ...                                                ...
957        23  Computer Skills: â¢ Proficient in MS office (...
958        23  â Willingness to accept the challenges. â ...
959        23  PERSONAL SKILLS â¢ Quick learner, â¢ Eagerne...
960        23  COMPUTER SKILLS & SOFTWARE KNOWLEDGE MS-Power ...
961        23  Skill Set OS Windows XP/7/8/8.1/10 Database MY...

                                        cleaned_resume
0    Skill    Programming Language   P thon  panda ...
1    Education Detail    Ma  2013 to Ma  2017 B E  ...
2    Area  of Intere t Deep Learning  Control S  te...
3    Skill      R     P thon     SAP HANA     Table...
4    Education Detail     MCA   YMCAUST   Faridabad...
..                                                 ...
957  Computer Skill       Proficient in MS office  ...
958      Willingne  to a ept the challenge       Po...
959  PERSONAL SKILLS     Quick learner      Eagerne...
960  COMPUTER SKILLS   SOFTWARE KNOWLEDGE MS Power ...
961  Skill Set OS Window  XP 7 8 8 1 10 Databa e MY...

[962 rows x 3 columns]
```

In [34]:
```
df.head()
```

Out[34]:

	Category	Resume	cleaned_resume
0	6	Skills * Programming Languages: Python (pandas...	Skill Programming Language P thon panda ...
1	6	Education Details \r\nMay 2013 to May 2017 B.E...	Education Detail Ma 2013 to Ma 2017 B E ...
2	6	Areas of Interest Deep Learning, Control Syste...	Area of Intere t Deep Learning Control S te...
3	6	Skills â¢ R â¢ Python â¢ SAP HANA â¢ Table...	Skill R P thon SAP HANA Table...
4	6	Education Details \r\n MCA YMCAUST, Faridab...	Education Detail MCA YMCAUST Faridabad...

In [35]:
```
from sklearn.model_selection import train_test_split
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.multiclass import OneVsRestClassifier
X_train, X_test, y_train, y_test = train_test_split(WordFeatures, requiredTarget, test_size=0.2, random_state=0
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
knn_classifier = KNeighborsClassifier()
clf = OneVsRestClassifier(knn_classifier)
clf.fit(X_train, y_train)
pred = clf.predict(X_test)
```

```
(769, 1500)
(193, 1500)
(769,)
(193,)
```

In [36]:
```
from sklearn import metrics
print('Accuracy of KNeighbors Classifier on training set: {:.2f}'.format(clf.score(X_train, y_train)))
print('Accuracy of KNeighbors Classifier on test set: {:.2f}'.format(clf.score(X_test, y_test)))
print("n Classification report for classifier %s:n%sn" % (clf, metrics.classification_report(y_test, pred)))
```

```
Accuracy of KNeighbors Classifier on training set: 0.99
Accuracy of KNeighbors Classifier on test set: 0.99
n Classification report for classifier OneVsRestClassifier(estimator=KNeighborsClassifier()):n              pre
cision    recall  f1-score   support

           0       1.00      1.00      1.00         3
           1       1.00      1.00      1.00         3
           2       1.00      0.80      0.89         5
           3       1.00      1.00      1.00         9
           4       1.00      1.00      1.00         6
           5       0.83      1.00      0.91         5
           6       1.00      1.00      1.00         9
           7       1.00      1.00      1.00         7
           8       1.00      0.91      0.95        11
           9       1.00      1.00      1.00         9
          10       1.00      1.00      1.00         8
          11       0.90      1.00      0.95         9
          12       1.00      1.00      1.00         5
          13       1.00      1.00      1.00         9
          14       1.00      1.00      1.00         7
          15       1.00      1.00      1.00        19
          16       1.00      1.00      1.00         3
          17       1.00      1.00      1.00         4
          18       1.00      1.00      1.00         5
          19       1.00      1.00      1.00         6
          20       1.00      1.00      1.00        11
          21       1.00      1.00      1.00         4
          22       1.00      1.00      1.00        13
          23       1.00      1.00      1.00        15
          24       1.00      1.00      1.00         8

    accuracy                           0.99       193
   macro avg       0.99      0.99      0.99       193
weighted avg       0.99      0.99      0.99       193
n
```

In [37]:
```
from sklearn.ensemble import RandomForestClassifier
X_train, X_test, y_train, y_test = train_test_split(WordFeatures, requiredTarget, test_size=0.2, random_state=0
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
rfc = RandomForestClassifier(n_estimators=100, random_state=0)
rfc.fit(X_train, y_train)
pred = rfc.predict(X_test)
accuracy = rfc.score(X_test, y_test)
print("Accuracy:", accuracy)
```

```
(769, 1500)
(193, 1500)
(769,)
(193,)
Accuracy: 1.0
```

In [38]:
```
print(X_train.dtype)
```

```
float64
```

In [39]:
```
print(y_train.dtype)
```

```
int32
```

In [40]:
```
import pickle

# Save the TF-IDF vectorizer
with open('tfidf.pkl', 'wb') as f:
    pickle.dump(tfidf_vectorizer, f)
```

```
# Save the classifier
with open('clf.pkl', 'wb') as f:
    pickle.dump(clf, f)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: