JOB DESCRIPTION BASED RESUME MATCHER USING SENTIMENT ANALYSIS, TEXT SUMMARIZATION, KEYWORD EXTRACTION

1. IMPORT LIBRARIES

> In this part, we will import all required packages and libraries which we will be using for further computation.

```
import numpy as nmp
import pandas as pnd
import nltk
import matplotlib.pyplot as pl0t
import seaborn as sbns
import re, os, string
import spacy
import warnings
warnings.filterwarnings("ignore")

nltk.download('vader_lexicon')
nltk.download('stopwords')
nltk.download('punkt')

from nltk import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import TweetTokenizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer as sian
from pprint import PrettyPrinter
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

In the above code we are importing numpy,pandas,nltk,matplotlib.pyplot,seaborn,spacy,warnings,vander_lexicon,stopwords,punkt and from nltk we are importing word_tokenize, stopwords, porterstemmer,tweettokenizer,sentimentIntensityAnalyzer,prettyprinter and Tfidvectorizer from python library.

2. READ AND PRE-PROCESS DATA

> In this part, we read the job description dataset using pandas. We then remove unwanted rows and columns. We remove the empty filled rows.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

In the above code we are mounting the google drive for copy the path of dataset.

```
job_dataset = pnd.read_csv('/content/drive/MyDrive/Colab Notebooks/monster_com-job_sample.csv')
job_dataset.head()
```

	country	country_code	date_added	has_expired	job_board	job_description	job_title	job_type	lc
0	United States of America	US	NaN	No	jobs.monster.com	TeamSoft is seeing an IT Support Specialist to...	IT Support Technician Job in Madison	Full Time Employee	M W
1	United States of America	US	NaN	No	jobs.monster.com	The Wisconsin State Journal is seeking a flexi...	Business Reporter/Editor Job in Madison	Full Time	M W
2	United States of America	US	NaN	No	jobs.monster.com	Report this job About the Job DePuy Synthes Co...	Johnson & Johnson Family of Companies Job Appl...	Full Time, Employee	Cor me Jo
3	United States of America	US	NaN	No	jobs.monster.com	Why Join Altec? If you're considering a career...	Engineer - Quality Job in Dixon	Full Time	Di
4	United States of America	US	NaN	No	jobs.monster.com	Position ID# 76162 # Positions 1 State CT C...	Shift Supervisor - Part-Time Job in Camphill	Full Time Employee	C

In the above part we are reading the "jobdescription.csv" file by "pandas" pnd variable by copying the path of dataset from google drive which we had uploaded in the previous step.

```
job_data = job_dataset[['job_description','uniq_id']].copy()

job_data = job_data.dropna()

#Lower-case all descriptions
job_data.title = job_data.job_description.str.lower()

#Remove handlers
job_data.title = job_data.job_description.apply(lambda x:re.sub('@[^\s]+','',x))

# Remove URLS
job_data.desc = job_data.job_description.apply(lambda x:re.sub(r"http\S+", "", x))

# Remove all the special characters
job_data.desc = job_data.job_description.apply(lambda x:' '.join(re.findall(r'\w+', x)))

#remove all single characters
job_data.desc = job_data.job_description.apply(lambda x:re.sub(r'\s+[a-zA-Z]\s+', '', x))

# Substituting multiple spaces with single space
job_data.desc = job_data.job_description.apply(lambda x:re.sub(r'\s+', ' ', x, flags=re.I))

#Remove Time From Timestamp
#job_data.timestamp = pd.to_datetime(job_data.timestamp).dt.date
```

Here we are dropping or removing the special characters,single characters, single space, multiple spaces and handlers etc from the job description dataset as a part of cleaning or preprocessing.

```
job_dataset.drop(['country', 'country_code','date_added','has_expired','job_board','job_type','location','organization','page_url','sector'],
```

	job_description	job_title	salary	uniq_id
0	TeamSoft is seeing an IT Support Specialist to...	IT Support Technician Job in Madison	NaN	11d599f229a80023d2f40e7c52cd941e
1	The Wisconsin State Journal is seeking a flexi...	Business Reporter/Editor Job in Madison	NaN	e4cbb126dabf22159aff90223243ff2a
2	Report this job About the Job DePuy Synthes Co...	Johnson & Johnson Family of Companies Job Appl...	NaN	839106b353877fa3d896ffb9c1fe01c0
3	Why Join Altec? If you're considering a career...	Engineer - Quality Job in Dixon	NaN	58435fcab804439efdcaa7ecca0fd783

Here we are dropping some extra coulumns from job description dataset such as "country","country_code","date_expired","job_board","job_type".

```
job_dataset.dropna(subset=['job_description'], inplace=True)
```

This is a major premier   Assistant Vice President -   120,000.00 -

Here we are dropping the null values of "job description" column, which is a part of data preprocessing.

Luxury homebuilder in   Accountant Job in   45,000.00 - 60,000.00

3. SENTIMENT ANALYSIS

> Here we perform the sentiment analysis on the above job descriptions. We do the lexicon based sentiment analysis , not the machine learning based sentiment analysis as the available dataset does not contain any pre trained data. We use the VaderSentiment for the current task. Also, we use the textblob to do the above task.

> A set of words or phrases and the sentiment scores they correspond with make up VaderSentiment's vocabulary. The scores range from -1 to 1, with a score of 1 representing a strongly favorable feeling and a score of 0 representing a neutral sentiment. Also, the lexicon contains guidelines for dealing with negations, intensifiers, and other linguistic elements that can influence the tone of a document.

```
pip install vaderSentiment
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting vaderSentiment
  Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl (125 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.0/126.0 KB 11.4 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from vaderSentiment) (2.27.1)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests->vaderSentiment) (2.0.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests->vaderSentiment) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests->vaderSentiment) (1.26.15)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->vaderSentiment) (3.4)
Installing collected packages: vaderSentiment
Successfully installed vaderSentiment-3.3.2
```

In the above step we are installing vadersentimentfrom the python library.

```
import vaderSentiment
# calling SentimentIntensityAnalyzer object
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
Sent_Analyser = SentimentIntensityAnalyzer()
```

In the above block of code we are importing SentimentInyensityAnalyzer from vederSentiment and defining the function SentimentIntensityAnalyzer() for the variable Snet_Analyser.

```
# Using polarity scores for knowing the polarity of each text
def sentiment_analyzer_score(sentence):
    score = Sent_Analyser.polarity_scores(sentence)
    print("{:-<40} {}".format(sentence, str(score)))
```

Here we are predicting the score for a particular job description sentence by defining polarity_scores in the function.

```
job_dataset["sentiment"] = job_dataset["job_description"].apply(lambda review: Sent_Analyser.polarity_scores(review))
```

```
job_dataset.head()
```

	country	country_code	date_added	has_expired	job_board	job_description	job_title	job_type	lc
0	United States of America	US	NaN	No	jobs.monster.com	TeamSoft is seeing an IT Support Specialist to...	IT Support Technician Job in Madison	Full Time Employee	M W
1	United States of America	US	NaN	No	jobs.monster.com	The Wisconsin State Journal is seeking a flexi...	Business Reporter/Editor Job in Madison	Full Time	M W
2	United States of America	US	NaN	No	jobs.monster.com	Report this job About the Job DePuy Synthes Co...	Johnson & Johnson Family of Companies Job Appl...	Full Time, Employee	Cor me Jo
3	United States of America	US	NaN	No	jobs.monster.com	Why Join Altec? If you're considering a career...	Engineer - Quality Job in Dixon	Full Time	Di
4	United States of America	US	NaN	No	jobs.monster.com	Position ID# 76162 # Positions 1 State CT C...	Shift Supervisor - Part-Time Job in Camphill	Full Time Employee	C

Here are diplaying the sentiment analysis that we had performed for the job description dataset which we had uploaded in the above step.

```
from textblob import TextBlob
```

Here we are importing the textblob from Textblob.

```
for text in job_dataset["job_description"]:
  if(TextBlob(text).sentiment.polarity>0):
    job_dataset["sentiment_class"]="positive"
  elif(TextBlob(text).sentiment.polarity<0):
    job_dataset["sentiment_class"]="negative"
  else:
    job_dataset["sentiment_class"]="neutral"
```

Here we are predicting the sentiment polarity for the particular "job description" for the data in the job description dataset.eg, {"positive',"Negative","Neutral"}.

```
job_dataset.head()
```

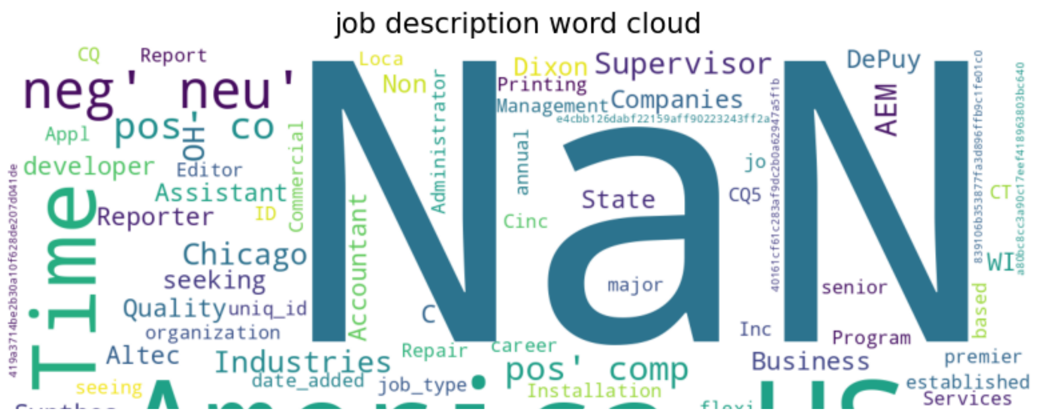
	country	country_code	date_added	has_expired	job_board	job_description	job_title	job_type	lo
0	United States of America	US	NaN	No	jobs.monster.com	TeamSoft is seeing an IT Support Specialist to...	IT Support Technician Job in Madison	Full Time Employee	M W
1	United States of America	US	NaN	No	jobs.monster.com	The Wisconsin State Journal is seeking a flexi...	Business Reporter/Editor Job in Madison	Full Time	M W
2	United States of America	US	NaN	No	jobs.monster.com	Report this job About the Job DePuy Synthes Co...	Johnson & Johnson Family of Companies Job Appl...	Full Time, Employee	Cor me Jo
3	United States of America	US	NaN	No	jobs.monster.com	Why Join Altec? If you're considering a career...	Engineer - Quality Job in Dixon	Full Time	Di
4	United States	US	NaN	No	jobs.monster.com	Position ID# 76162 # Positions	Shift Supervisor -	Full Time	C

Here we are printing the Info about the dataset after performing the sentiment analysis.

```
from wordcloud import WordCloud
positive = job_dataset[job_dataset['sentiment']==1]

pl0t.rcParams['figure.figsize']=(10,10)
pl0t.style.use('fast')


wc=WordCloud(background_color='white',width=1200,height=1200).generate(str(job_dataset))
pl0t.title('job description word cloud',fontsize=15)
pl0t.imshow(wc)
pl0t.axis('off')
pl0t.show()
```



Here we are displaying the Visualization of sentiment analysis that we had performed in the above step.

```
# total number of positive and negative sentiments
l = len(job_dataset[job_dataset['sentiment_class'] == 'negative'])
print(f"negative sentences= {l} ")

m = len(job_dataset[job_dataset['sentiment_class'] == 'positive'])
print(f" positive sentences= {m}")

m = len(job_dataset[job_dataset['sentiment_class'] == 'neutral'])
print(f" neutral sentences= {m}")
```

```
    negative sentences= 0 
     positive sentences= 22000
     neutral sentences= 0
```

Here we are predicting the "positive sentences","negative sentences" and "neutral sentences" that are present in the dataset.

```
job_dataset['sentiment_class'] = job_dataset['sentiment_class'].map({'positive':1,'negative':-1,'neutral':0},na_action=None)
count = sbns.countplot(data=job_dataset,x='sentiment_class',order=job_dataset['sentiment_class'].value_counts().index)
pl0t.show()
```

Above graph depicts the Graphical visualzation of "positive sentences" that we had predicted for the dataset.

4. TEXT SUMMARIZATION

> In this part, we do the text summarization on the job description dataset. For every job description form the dataset, we extract the brief summary and add it to the dataframe. We create a spacy pipeline and define a summary function.

> Text summarization is a natural language processing (NLP) technique that includes condensing a text while keeping the key points. Text summary aims to provide a condensed version of a text that captures the key ideas and is simpler to read and comprehend.

```
! pip install pytextrank
import pytextrank #ranking text
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pytextrank
  Downloading pytextrank-3.2.4-py3-none-any.whl (30 kB)
Requirement already satisfied: graphviz>=0.13 in /usr/local/lib/python3.9/dist-packages (from pytextrank) (0.20.1)
Requirement already satisfied: networkx[default]>=2.6 in /usr/local/lib/python3.9/dist-packages (from pytextrank) (3.0)
Requirement already satisfied: scipy>=1.7 in /usr/local/lib/python3.9/dist-packages (from pytextrank) (1.10.1)
Requirement already satisfied: pygments>=2.7.4 in /usr/local/lib/python3.9/dist-packages (from pytextrank) (2.14.0)
Collecting icecream>=2.1
  Downloading icecream-2.1.3-py2.py3-none-any.whl (8.4 kB)
Requirement already satisfied: spacy>=3.0 in /usr/local/lib/python3.9/dist-packages (from pytextrank) (3.5.1)
Collecting executing>=0.3.1
  Downloading executing-1.2.0-py2.py3-none-any.whl (24 kB)
Collecting asttokens>=2.0.1
  Downloading asttokens-2.2.1-py2.py3-none-any.whl (26 kB)
Collecting colorama>=0.3.9
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: matplotlib>=3.4 in /usr/local/lib/python3.9/dist-packages (from networkx[default]>=2.6->pytextrank) (
Requirement already satisfied: pandas>=1.3 in /usr/local/lib/python3.9/dist-packages (from networkx[default]>=2.6->pytextrank) (1.4.
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.9/dist-packages (from networkx[default]>=2.6->pytextrank) (1.22
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (4.65.0)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (3.1.2)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pyt
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (6.3
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (8.1.9)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (1.
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (0.7.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (23.0)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (2.0.7)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (2.0.
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (67.6.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (3.3.
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (0.10.1)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (1.1.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (2.27
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (3.0.8)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (2.4.6)
Requirement already satisfied: six in /usr/local/lib/python3.9/dist-packages (from asttokens>=2.0.1->icecream>=2.1->pytextrank) (1.1
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=
Requirement already satisfied: importlib-resources>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=2.6
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=2.6-
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[defaul
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas>=1.3->networkx[default]>=2.6->pyt
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.9/dist-packages (from pydantic!=1.8,!=1.8.1,<1.11.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.13.0->spacy>=3.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.13.0->spacy>=3.0->pyte
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.13.0->spa
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.13.0->spacy>
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.9/dist-packages (from thinc<8.2.0,>=8.1.8->spacy>=
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.9/dist-packages (from thinc<8.2.0,>=8.1.8->spacy>=3.0->py
```

In the above code we are installing the pytextrank from the python library.

```
pp = PrettyPrinter()
```

Here we are defining prettyprinter() function for pp variable.

```
job_dataset.info() # infor about the data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22000 entries, 0 to 21999
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   country          22000 non-null  object
 1   country_code     22000 non-null  object
 2   date_added       122 non-null    object
 3   has_expired      22000 non-null  object
 4   job_board        22000 non-null  object
 5   job_description  22000 non-null  object
 6   job_title        22000 non-null  object
 7   job_type         20372 non-null  object
 8   location         22000 non-null  object
 9   organization     15133 non-null  object
 10  page_url         22000 non-null  object
 11  salary           3446 non-null   object
 12  sector           16806 non-null  object
 13  uniq_id          22000 non-null  object
 14  sentiment        22000 non-null  object
 15  sentiment_class  22000 non-null  int64
dtypes: int64(1), object(15)
memory usage: 2.7+ MB
```

Here we are printing the Information about the Job description about the Job dataset.

```
# Create spaCy pipeline and add textrank to it
import spacy.cli

spacy.cli.download("en_core_web_lg")
nlp = spacy.load("en_core_web_lg")
nlp.add_pipe("textrank")
```

```
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_lg')
<pytextrank.base.BaseTextRankFactory at 0x7fa7b0dedc70>
```

Here we are importing the spacy.cli library and downloading "en_core_web_lg" &"en_core_web_lg" corpus data from python library.

```
def summary_for_article(num,prin=False):

    sum = "" # collecting the summary from the generator
    document = nlp(job_dataset.job_description[num]) #apply the pipeline

    for i in document._.textrank.summary(limit_phrases=10, limit_sentences=1): #get the summary
        sum+=str(i)

    phrases_n_ranks = [ (phrase.chunks[0], phrase.rank) for phrase in document._.phrases] # get important phrases

    if prin:
        print(job_dataset.job_description[num])
        print("\n_______ to ______\n")
        print(sum)


    return sum
```

Here we are assigning the ranks for each and every row present in job description dataset after performing the sentiment analysis.

```
for i in range(0,100):
    print("\n....",i,"")
    summary_for_article(i,True);
    job_dataset["summary"]=summary_for_article(i,True)
```

```
_______ to ______

Background experience in the financial services industry and experience with Iterative Development methodologies is highly desirable
The Judge Group is looking for a Scrum Master for our client in Denver. Please email Josh Freidus at Jfreidus@judge.com for more det

_______ to ______

Background experience in the financial services industry and experience with Iterative Development methodologies is highly desirable

.... 96
Financial Advisor Northwestern MutualOur financial advisory firm is seeking new Financial Advisors to join growing practice. Strong a

_______ to ______

They strive to understand their clients' goals and dreams in order to develop comprehensive financial solutions that will help their
Financial Advisor Northwestern MutualOur financial advisory firm is seeking new Financial Advisors to join growing practice. Strong a

_______ to ______

They strive to understand their clients' goals and dreams in order to develop comprehensive financial solutions that will help their

.... 97
Job Title: Personal Banker (SAFE) 1 - Bear ValleyJob ID Number: 5205451Schedule Type: Reg-TimeWork Hours: 40Location: Denver,COQuali

_______ to ______

Bankers have the ability to resolve difficult customer situations effectively while delivering friendly customer service and ensurin
Job Title: Personal Banker (SAFE) 1 - Bear ValleyJob ID Number: 5205451Schedule Type: Reg-TimeWork Hours: 40Location: Denver,COQuali

_______ to ______

Bankers have the ability to resolve difficult customer situations effectively while delivering friendly customer service and ensurin

.... 98
*****THIS POSITION IS IN Decatur, IL. PLEASE APPLY ONLY IF YOU ARE INTERESTED FOR THAT LOCATION*****Volt has been serving some of the

_______ to ______

Qualifications: Design experience with automotive,industrial equipment, or machinery Strong communications skills Experience leading
*****THIS POSITION IS IN Decatur, IL. PLEASE APPLY ONLY IF YOU ARE INTERESTED FOR THAT LOCATION*****Volt has been serving some of the

_______ to ______

Qualifications: Design experience with automotive,industrial equipment, or machinery Strong communications skills Experience leading

.... 99
MOUNTAIN, LTD. is currently seeking an OSP Field Engineer for a leading communications company in the Fort Collins, CO area.Position

_______ to ______

The successful candidate will have strong fielding skills and all necessary tools, and be capable of acquiring necessary permits and
MOUNTAIN, LTD. is currently seeking an OSP Field Engineer for a leading communications company in the Fort Collins, CO area.Position

_______ to ______

The successful candidate will have strong fielding skills and all necessary tools, and be capable of acquiring necessary permits and
```

Here we are printing Text Summary of Job description Dataset as a part of Text Summarization.

5. KEYWORD EXTRACTION FROM RESUME DATASET

In this part, we extract the keywords from the resume dataset which is required for the other part of our project. We first read and pre-process our data and then apply different pre-processing functions on the data. We also use TfIdfVectorizer and extract the top keywords from the resume description and then add it to the dataframe.Keyword extraction involves locating the most crucial

> words or phrases in a text. To help with tasks like document categorization, topic modeling, and information retrieval, keyword extraction aims to identify the major subjects or themes in a document or corpus.

```
def get_stopwords_list(stop_file_path):
    """load stop words """

    with open(stop_file_path, 'r', encoding="utf-8") as f:
        stop_words = f.readlines()
        stopwords_set = set(m.strip() for m in stop_words)
        return list(frozenset(stopwords_set))
```

Here we are removing the stopwords for "resume.csv" dataset which is used as the process of extracting the keywords from the given sample resumes.

```
def clean_text(text):
    """Doc cleaning"""

    # Lowering text
    text = text.lower()

    # Removing punctuation
    text = "".join([c for c in text if c not in PUNCTUATION])

    # Removing whitespace and newlines
    text = re.sub('\s+',' ',text)

    return text
```

Here we are performing the pre-processing for "Resume.csv" dataset by adding the punctuation as well as converting the text to lower and adding the punctuation to the text present in dataset.

```
def sort_coo(coo_matrix):
    """Sort a dict with highest score"""
    tuples = zip(coo_matrix.col, coo_matrix.data)
    return sorted(tuples, key=lambda x: (x[1], x[0]), reverse=True)

def extract_topn_from_vector(feature_names, sorted_items, topn=10):
    """get the feature names and tf-idf score of top n items"""

    #use only topn items from vector
    sorted_items = sorted_items[:topn]

    score_vals = []
    feature_vals = []

    # word index and corresponding tf-idf score
    for idx, score in sorted_items:

        #keep track of feature name and its corresponding score
        score_vals.append(round(score, 3))
        feature_vals.append(feature_names[idx])

    #create a tuples of feature, score
    results= {}
    for idx in range(len(feature_vals)):
        results[feature_vals[idx]]=score_vals[idx]

    return results
```

Here we are intializing the features and functions that are required for extraction of keywords from "resume.csv" dataset.

```
def get_keywords(vectorizer, feature_names, doc):
    """Return top k keywords from a doc using TF-IDF method"""

    #generate tf-idf for the given document
    tf_idf_vector = vectorizer.transform([doc])

    #sort the tf-idf vectors by descending order of scores
    sorted_items=sort_coo(tf_idf_vector.tocoo())
    #extract only TOP_K_KEYWORDS
    keywords=extract_topn_from_vector(feature_names,sorted_items,TOP_K_KEYWORDS)

    return list(keywords.keys())
```

Here we are extracting the keywords from "Resume.csv" dataset by initializing Vectorizer funtion to tf_idf_vector variable.

```
# Constants
PUNCTUATION = """!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""
TOP_K_KEYWORDS = 10 # top k number of keywords to retrieve in a ranked document
STOPWORD_PATH = "/content/drive/MyDrive/Colab Notebooks/stopwords.txt"
Resume_PATH = "/content/drive/MyDrive/Colab Notebooks/Resume.csv"
```

Assigning path to each and every and variable such as "Resume_Path" and " Stopword_Path".

```
data = pnd.read_csv(Resume_PATH)
data.drop(["Resume_html"],axis=1)
data.head()
```

	ID	Resume_str	Resume_html	Category
0	16852973	HR ADMINISTRATOR/MARKETING ASSOCIATE\...	<div class="fontsize fontface vmargins hmargin...	HR
1	22323967	HR SPECIALIST, US HR OPERATIONS ...	<div class="fontsize fontface vmargins hmargin...	HR
2	33176873	HR DIRECTOR Summary Over 2...	<div class="fontsize fontface vmargins hmargin...	HR
3	27018550	HR SPECIALIST Summary Dedica...	<div class="fontsize fontface vmargins hmargin...	HR
4	17812897	HR MANAGER Skill Highlights ...	<div class="fontsize fontface vmargins hmargin...	HR

Here we are performing some data mining process such as Cleaning the text of Resume_html for "Resume.csv" Dataset by dropping the data of "Resume_htmL" as a part of cleaned,labelled,punctuated text.

```
data.dropna(subset=['Resume_str'], inplace=True)
```

```
data['Resume_str'] = data['Resume_str'].apply(clean_text)
data.head()
```

	ID	Resume_str	Resume_html	Category
0	16852973	hr administratormarketing associate hr admini...	<div class="fontsize fontface vmargins hmargin...	HR
1	22323967	hr specialist us hr operations summary versat...	<div class="fontsize fontface vmargins hmargin...	HR
2	33176873	hr director summary over 20 years experience ...	<div class="fontsize fontface vmargins hmargin...	HR
3	27018550	hr specialist summary dedicated driven and dy...	<div class="fontsize fontface vmargins hmargin...	HR
4	17812897	hr manager skill highlights hr skills hr depa...	<div class="fontsize fontface vmargins hmargin...	HR

Here we are performing some data mining process such as Cleaning the text of Resume_str for "Resume.csv" Dataset by dropping the data of "Resume_str" as a part of cleaned,labelled,punctuated text.

```
corpora = data['Resume_str'].to_list()
```

Here we are transforming the text of "Resume_str" column to list format.

```
#load a set of stop words
stopwords=get_stopwords_list(STOPWORD_PATH)

# Initializing TF-IDF Vectorizer with stopwords
vector = TfidfVectorizer(stop_words=stopwords, smooth_idf=True, use_idf=True)

# Creating vocab with our corpora
# Exlcluding first 10 docs for testing purpose
vector.fit_transform(corpora[10::])
```

```
# Storing vocab
ftr_names = vector.get_feature_names_out()
```

Finally we are removing stopwords and null values present in the text of 'Resume.csv" dataset by assigning "stopwords.txt" path to TfidVrctorizer function.

```
rslt = []
for docment in corpora[0:10]:
    KW_dataset = {}
    KW_dataset['Resume_str'] = docment
    KW_dataset['top_keywords'] = get_keywords(vector, ftr_names, docment)
    rslt.append(KW_dataset)

output = pnd.DataFrame(rslt)
output
```

	Resume_str	top_keywords
0	hr administratormarketing associate hr admini...	[marketing, dec, medical, relations, customer,...
1	hr specialist us hr operations summary versat...	[marketing, hr, sharepoint, materials, brochur...
2	hr director summary over 20 years experience ...	[hris, friends, hr, kansas, adjutant, topeka, ...
3	hr specialist summary dedicated driven and dy...	[call, 10key, touch, customer, hr, comments, w...
4	hr manager skill highlights hr skills hr depa...	[hr, employee, human, benefits, jan, compensat...
5	hr generalist summary dedicated and focused a...	[nonimmigrant, uscis, petitions, 112008, perfo...
6	hr manager summary human resources manager ex...	[hr, training, staff, tesol, development, huma...
7	hr manager professional summary senior hr pro...	[employee, benefits, human, employees, resourc...
8	hr specialist summary possess 15 years of exp...	[hr, statewide, salary, recruitment, pay, comp...
9	hr clerk summary translates business vision i...	[hr, shrm, employee, compensation, administrat...

Here we are displaying the "Keywords" for each and every resume present in the "resume.csv" dataset after cleaning the text of "Resume_str" and "Resume_html" columns.

End of Project Increment1

✓ 0s completed at 2:43 AM