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]   Package vader_lexicon is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

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.

```
job_dataset = pnd.read_csv('/content/sample_data/monster_com-job_sample.csv')
job_dataset.head()
```

	country	country_code	date_added	has_expired	job_board	job_descriptior
0	United States of America	US	NaN	No	jobs.monster.com	TeamSoft is seeing an IT Support Specialist to..
1	United States of America	US	NaN	No	jobs.monster.com	The Wisconsir State Journal is seeking a flexi..
2	United States	US	NaN	No	jobs.monster.com	Report this job About the Job

```
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_i
0	TeamSoft is seeing an IT Support Specialist to...	IT Support Technician Job in Madison	NaN	11d599f229a80023d2f40e7c52cd941
1	The Wisconsin State Journal is seeking a flexi...	Business Reporter/Editor Job in Madison	NaN	e4cbb126dabf22159aff90223243ff2
2	Report this job About the Job DePuy Synthes Co...	Johnson & Johnson Family of Companies Job Appl...	NaN	839106b353877fa3d896ffb9c1fe01c
3	Why Join Altec? If you're considering a career...	Engineer - Quality Job in Dixon	NaN	58435fcab804439efdcaa7ecca0fd78
4	Position ID# 76162 # Positions 1 State CT C...	Shift Supervisor - Part-Time Job in Camphill	NaN	64d0272dc8496abfd9523a8df63c184
...	...	...	...	
21995	This is a major premier Cincinnati based finan...	Assistant Vice President - Controller Job in C...	120,000.00 - 160,000.00 $ /yearbonus	a80bc8cc3a90c17eef418963803bc64
			45,000.00	

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

job_dataset = job_dataset.dropna()

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

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

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

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

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

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

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

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 4.2 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: 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)
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)
Installing collected packages: vaderSentiment
Successfully installed vaderSentiment-3.3.2
```

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

```
# 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)))
```

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

```
job_dataset.head()
```

	job_description	uniq_id	sentiment
0	TeamSoft is seeing an IT Support Specialist to...	11d599f229a80023d2f40e7c52cd941e	{'neg': 0.009, 'neu': 0.822, 'pos': 0.169, 'co...
1	The Wisconsin State Journal is seeking a flexi...	e4cbb126dabf22159aff90223243ff2a	{'neg': 0.0, 'neu': 0.868, 'pos': 0.132, 'comp...
2	Report this job About the Job	839106b353877fa3d896ffb9c1fe01c0	{'neg': 0.004, 'neu': 0.857, 'pos': 0.139

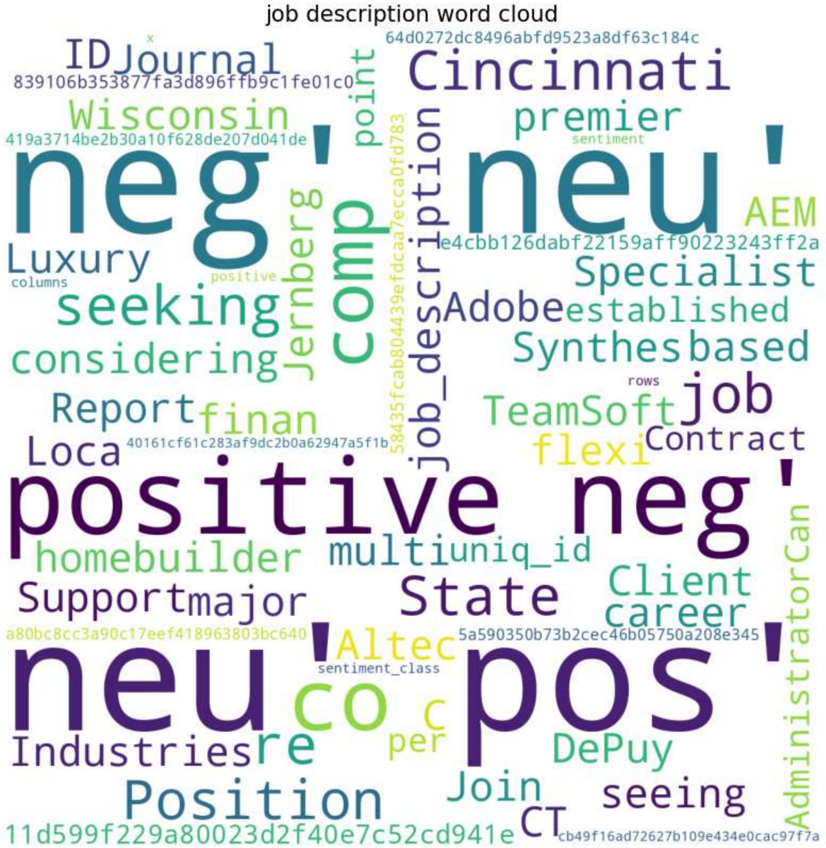
```
from textblob import TextBlob
```

```
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"
```

```
job_dataset.head()
```

	job_description	uniq_id	sentiment	sentiment_class
0	TeamSoft is seeing an IT Support Specialist to...	11d599f229a80023d2f40e7c52cd941e	{'neg': 0.009, 'neu': 0.822, 'pos': 0.169, 'co...	positive

```
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()
```

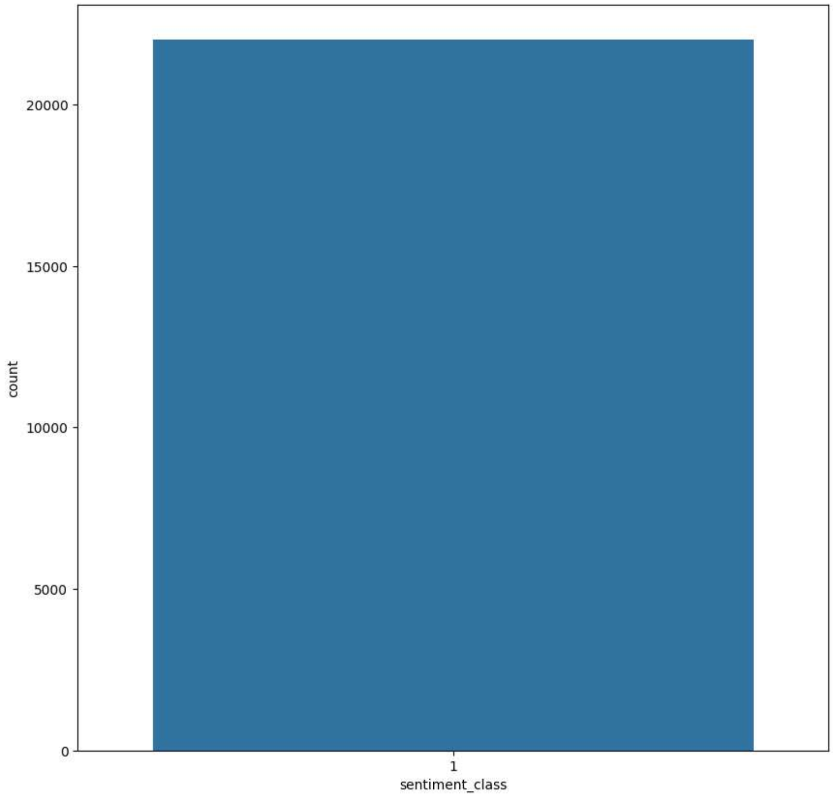


```
# total number of positive and negative sentiments
l = len(job_dataset[job_dataset['sentiment_class'] == 'negative'])
print(f"negative sentences= {l} ")

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
```

```
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()
```



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
```

```
    Requirement already satisfied: scipy>=1.7 in /usr/local/lib/python3.9/dist-packages (from pytextrank) (1.10.1)
    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 colorama>=0.3.9
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
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)
Requirement already satisfied: pandas>=1.3 in /usr/local/lib/python3.9/dist-packages (from networkx[default]>=2.6->pytextrank) (1.4.4
Requirement already satisfied: matplotlib>=3.4 in /usr/local/lib/python3.9/dist-packages (from networkx[default]>=2.6->pytextrank) (3
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: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (1
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: 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: 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: 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: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (1.0
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: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (23.0)
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->pyte
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (3.1.2)
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) (3.3.0
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: 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: 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: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (3
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: pathy>=0.10.0 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (0.10.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.9/dist-packages (from spacy>=3.0->pytextrank) (2.0.8
Requirement already satisfied: six in /usr/local/lib/python3.9/dist-packages (from asttokens>=2.0.1->icecream>=2.1->pytextrank) (1.16
Requirement already satisfied: kiwisolver>=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: cycler>=0.10 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=2.6->
Requirement already satisfied: importlib-resources>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[d
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=2
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx[default]>=2
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas>=1.3->networkx[default]>=2.6->pyte
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.9/dist-packages (from pydantic!=1.8,!=1.8.1,<1.11.0
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: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests<3.0.0,>=2.13.0->spacy>=3.0
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: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.9/dist-packages (from thinc<8.2.0,>=8.1.8->spacy>=3
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
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.9/dist-packages (from typer<0.8.0,>=0.3.0->spacy>=3.0->py
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from jinja2->spacy>=3.0->pytextrank) (2.1.2
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.9/dist-packages (from importlib-resources>=3.2.0->matplotlib>=3.
Installing collected packages: executing, colorama, asttokens, icecream, pytextrank
```

```
pp = PrettyPrinter()
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22000 entries, 0 to 21999
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   job_description  22000 non-null  object
 1   uniq_id          22000 non-null  object
 2   sentiment        22000 non-null  object
 3   sentiment_class  22000 non-null  int64 
dtypes: int64(1), object(3)
memory usage: 687.6+ KB
```

```
# 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 0x7f2097bb2f40>
```

```
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


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

```
If you have a passion for customer service along with trouble shooting experience you are encouraged to apply!Volt Offers: Competitive

.... 92
Chamberlin Roofing & Waterproofing is an established commercial specialty contractor that provides roofing and sheet metal, waterproof

_______ to ______

Good organizational and communication skills Have good math and writing skills Work well as an essential team member   Job Purpose:Pro

.... 93
About Agfa HealthCare   Agfa HealthCare, a member of the Agfa-Gevaert Group, is a leading global provider of diagnostic imaging and  I

_______ to ______

Take lead responsibility for a specific custom development project within Professional Services Solution Architects organization Consu

.... 94
BASIC FUNCTION AND SCOPE OF JOBConceives designs and develops power conversion, power control, and system communication products.  The

_______ to ______

WORK PERFORMED-Development of detailed analytical design analysis-Preparation of electronic schematic diagrams and part lists-Particip

.... 95
The Judge Group is looking for a Scrum Master for our client in Denver. Please email Josh Freidus at Jfreidus@judge.com for more deta

_______ 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 ap

_______ to ______

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

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

_______ to ______

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

.... 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 c

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

_______ to ______

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

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))


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


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


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())
```

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

```
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 Dedi...	<div class="fontsize fontface	HR

```
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	<div class="fontsize fontface	HR

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

```
#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()
```

Double-click (or enter) to edit

```
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	[illegible]	[employee, benefits, human, employees,

End of Project Increment1

✓ 0s completed at 10:50 PM