In [35]:
```
# pip install pymupdf
# !pip install gensim
```

In [41]:
```
# pip install sentence-transformers
```

In [ ]:
```
import fitz  # PyMuPDF
import os
import nltk
nltk.download('punkt')      # Tokenizer
nltk.download('stopwords')  # Common words like "the", "and"
nltk.download('wordnet')    # For lemmatization

import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

from sentence_transformers import SentenceTransformer, util

import warnings
warnings.filterwarnings('ignore')

import pandas as pd
```

In [ ]:
```

```

PDF Reader

Extracting Text from PDF File

In [150…
```
def extract_text_from_pdf(pdf_path):
    text = ""
    with fitz.open(pdf_path) as doc:
        for page in doc:
            text += page.get_text()
    return text
```

Preprocessing Text obtained from the PDF

In [229…
```
def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'\S+@\S+', '', text)        # Remove emails
    text = re.sub(r'http\S+|www\S+', '', text) # Remove URLs
    text = re.sub(r'\d+', '', text)            # Remove digits
    text = re.sub(r'[^\w\s]', '', text)        # Remove punctuation

    tokens = word_tokenize(text)
    stop_words = set(stopwords.words('english'))
```

```
    tokens = [word for word in tokens if word not in stop_words]

    lemmatizer = WordNetLemmatizer()
    lemmatized = [lemmatizer.lemmatize(word) for word in tokens]

    return ' '.join(lemmatized)
```

In [154…
```
pdf_file_path = #YourFolder path here
```

Sample 1

In [155…
```
Sample_1_CV_Text = extract_text_from_pdf(os.path.join(pdf_file_path, "Sample
```

In [156…
```
Sample_1_CV_Text
```

Out[156…
```
' \n \n \n \nKRISTI \nLAAR \n \nREGISTERED NURSE \nCONTACT \n111 1st Avenue
\nRedmond, WA 65432 \n909.555.0100  \nkristi@example.com \nwww.interestings
ite.com \n \n \n \nCOMMUNICATION \nI have received several awards \nfor my
outstanding \ncommunication skills, including \nrecognition for providing
\nexceptional patient education \nand counseling. \n \n \n \nLEADERSHIP \nI
received the "Outstanding \nNursing Student" award during \nmy time in nurs
ing school, and I \nhave been recognized for my \ncontributions to patient
safety \nand satisfaction in my current \nrole. \nEDUCATION \nBellows Colle
ge | Madison, WI  \nBachelors of Science in Nursing \nRelevant coursework:
Anatomy and physiology, pharmacology, \nnursing ethics, and patient care ma
nagement.  \n \n \n \nEXPERIENCE \nNovember 20XX–October 20XX \nRegistered
Nurse | Pediatrics | Wholeness Healthcare \nDecember 20XX–November 20XX \nR
egistered Nurse | General Practice | Wholeness Healthcare \nSeptember 20XX–
August 20XX \nRegistered Nurse | General Practice | Tyler Stein MD  \nI hav
e a proven track record of delivering high-quality care while \nmaintaining
patient safety and satisfaction \n \n \n \nREFERENCES \n[Available upon req
uest]  \n \n'
```

In [157…
```
clean_resume_text_1 = preprocess_text(Sample_1_CV_Text)
```

In [158…
```
clean_resume_text_1
```

Out[158…
```
'kristi laar registered nurse contact st avenue redmond wa communication re
ceived several award outstanding communication skill including recognition
providing exceptional patient education counseling leadership received outs
tanding nursing student award time nursing school recognized contribution p
atient safety satisfaction current role education bellow college madison wi
bachelor science nursing relevant coursework anatomy physiology pharmacolog
y nursing ethic patient care management experience november xxoctober xx re
gistered nurse pediatrics wholeness healthcare december xxnovember xx regis
tered nurse general practice wholeness healthcare september xxaugust xx reg
istered nurse general practice tyler stein md proven track record deliverin
g highquality care maintaining patient safety satisfaction reference availa
ble upon request'
```

Sample 2

In [159…
```
Sample_2_CV_Text = extract_text_from_pdf(os.path.join(pdf_file_path, "Sample
clean_resume_text_2 = preprocess_text(Sample_2_CV_Text)
```

In [160…
```
clean_resume_text_2
```

Out[160…
```
'janna gardner main street chicago illinois human resource generalist year
experience assisting fulfilling organization staffing need requirement prov
en track record using excellent personal communication organization skill l
ead improve hr department recruit excellent personnel improve department ef
ficiency team player excellent communication skill high quality work driven
highly selfmotivated strong negotiating skill business acumen able work ind
ependently e x per ence xx present human resource generalist lamna healthca
re company chicago illinois review update revise company hiring practice va
cation human resource policy ensure compliance osha local state federal lab
or regulation creating maintaining positive responsive work environment rai
sed employee retention rate achieve greater employee retention year period
developed recruitment program successfully increase minority recruitment me
et affirmative action requirement led development team build deploy dedicat
ed recruitment website reduced yearoveryear recruitment cost june xx august
xx human resource intern wholeness healthcare boomtown ohio assisted recrui
tment outreach prospective employee organized conducted several seminar hos
pital employee educate update regarding available employment benefit option
arranged hospitalwide guest speaker symposium educate management new employ
ment law workplace confidence morale building technique administrative task
sk l l type word per minute proficient project management software team pla
yer excellent time management skill conflict management public speaking dat
a analytics e duca ti n may xx bachelor art human resource management jaspe
r university ft lauderdale fl gpa member university honor society acti v ti
e literature environmental conservation art yoga skiing travel'
```

Sample 3

In [161…
```
Sample_3_CV_Text = extract_text_from_pdf(os.path.join(pdf_file_path, "Sample
clean_resume_text_3 = preprocess_text(Sample_3_CV_Text)
```

In [162…
```
clean_resume_text_3
```

Out[162…
```
'takanori ito senior sale engineer objective replace text click start typin
g briefly state career objective summarize make stand use language job desc
ription keywords experience senior sale engineer relecloud xxxx describe re
sponsibility achievement term impact result use example keep short sale eng
ineer relecloud xxxx describe responsibility achievement term impact result
use example keep short skill list one strength list one strength list one s
trength list one strength list one strength intern relecloud xxxx describe
responsibility achievement term impact result use example keep short educat
ion b information technology jasper university xxxx okay brag gpa award hon
or feel free summarize coursework contact albany ny linkedin profile intere
stingsitecom b computer science mount flores college xxxx okay brag gpa awa
rd honor feel free summarize coursework certificate mount flores college xx
xx okay brag gpa award honor feel free summarize coursework'
```

Job Description

In [163…
```
jd_text = """
Job Title: Registered Nurse – Pediatric Department

Location: Redmond, WA
Company: Wholeness Healthcare
```

```
We are looking for a compassionate and dedicated Registered Nurse to join ou

Responsibilities:
- Provide direct patient care, administer medications, and assist with proce
- Maintain accurate patient records
- Educate patients and families about treatments and care plans
- Monitor patient health and report any changes
- Ensure a safe and clean healthcare environment

Qualifications:
- Bachelor's degree in Nursing (BSN) required
- Valid RN license in Washington State
- Minimum 1 year of nursing experience, pediatric experience preferred
- Excellent communication and leadership skills
- Strong attention to detail and a passion for patient care

Preferred:
- CPR/BLS certification
- Experience with electronic medical records (EMR)
"""
```

In [164…
```
clean_jd = preprocess_text(jd_text)
```

Comparing Resume Text and Job Description using TfidfVectorizer

In [165…
```
# def calculate_similarity(text1, text2):
#     vectorizer = TfidfVectorizer()
#     vectors = vectorizer.fit_transform([text1, text2])
#     similarity = cosine_similarity(vectors[0:1], vectors[1:2])
#     return similarity[0][0]
```

In [166…
```
# def calculate_tfidf_similarity(text1, text2):
#     vectorizer = TfidfVectorizer(
#         stop_words='english',
#         ngram_range=(1, 2),  # Include unigrams and bigrams
#         max_df=0.85
#     )
#     vectors = vectorizer.fit_transform([text1, text2])
#     similarity = cosine_similarity(vectors[0:1], vectors[1:2])
#     return similarity[0][0]
```

In [230…
```
def calculate_tfidf_similarity(text1, text2):
    texts = [text1, text2]
    vectorizer = TfidfVectorizer(ngram_range=(1, 2), stop_words='english')
    X = vectorizer.fit_transform(texts)
    sim = cosine_similarity(X[0:1], X[1:2])[0][0]
    return sim
```

In [231…
```
similarity_score_1 = calculate_similarity(clean_resume_text_1, clean_jd)
print(f"Similarity Score: {similarity_score_1:.2f}")
```

```
Similarity Score: 0.30
```

In [232…
```
similarity_score_2 = calculate_similarity(clean_resume_text_2, clean_jd)
print(f"Similarity Score: {similarity_score_2:.2f}")
```

```
Similarity Score: 0.08
```

In [233…
```
similarity_score_3 = calculate_similarity(clean_resume_text_3, clean_jd)
print(f"Similarity Score: {similarity_score_3:.2f}")
```

```
Similarity Score: 0.02
```

Comparing Resume Text and Job Description using Sentence Transformer

In [ ]:
```
model = SentenceTransformer('all-MiniLM-L6-v2')  # Fast and lightweight

def calculate_semantic_similarity(text1, text2):
    embeddings = model.encode([text1, text2])
    similarity = util.cos_sim(embeddings[0], embeddings[1])
    return float(similarity)
```

In [171…
```
semantic_score_1 = calculate_semantic_similarity(Sample_1_CV_Text, jd_text)
semantic_score_2 = calculate_semantic_similarity(Sample_2_CV_Text, jd_text)
semantic_score_3 = calculate_semantic_similarity(Sample_3_CV_Text, jd_text)
```

In [172…
```
print(semantic_score_1)
print(semantic_score_2)
print(semantic_score_3)
```

```
0.6585899591445923
0.4036364257335663
0.27140164375305176
```

In [ ]:
```
```

Data Analyst JD : Resume Scores

JD

In [248…
```
DA_JD_raw = extract_text_from_pdf(#YourFilePath here)
```

In [249…
```
DA_JD_preprocessed = preprocess_text(DA_JD_raw)
```

Resume

In [250…
```
resume_folder = r#YourFolderPath here
resume_files = [f for f in os.listdir(resume_folder) if f.lower().endswith('
resume_paths = [os.path.join(resume_folder, f) for f in resume_files]
```

In [251…
```
resume_files
```

Out[251…
```
['data-analyst-intern-resume-example.pdf',
 'data-analyst-resume-example.pdf',
 'entry-level-risk-adjustment-data-analyst-resume-example.pdf',
 'experienced-data-analyst-resume-example.pdf',
 'junior-data-analyst-resume-example.pdf',
 'Non_DA_Sample_1.pdf',
 'Non_DA_Sample_2.pdf',
 'Non_DA_Sample_3.pdf',
 'revenue-reporting-data-analyst-resume-example.pdf',
 'senior-data-analyst-resume-example.pdf',
 'senior-insurance-data-analyst-resume-example.pdf']
```

In [252…
```
resume_raw_texts = [extract_text_from_pdf(path) for path in resume_paths]
resume_clean_texts = [preprocess_text(text) for text in resume_raw_texts]
```

In [253…
```
def calculate_resume_scores(resume_filenames, resume_raw_texts, resume_clear
    results = []
    for i in range(len(resume_filenames)):
        filename = resume_filenames[i]
        raw_text = resume_raw_texts[i]
        clean_text = resume_clean_texts[i]
        tfidf_score = calculate_tfidf_similarity(clean_text, jd_clean)
        semantic_score = calculate_semantic_similarity(raw_text, jd_raw)
        results.append({
            "Resume": filename,
            "TF-IDF Score": round(tfidf_score, 3),
            "Semantic Score": round(semantic_score, 3)
        })
    df_scores = pd.DataFrame(results)
    return df_scores
```

In [254…
```
score_table = calculate_resume_scores(resume_files, resume_raw_texts, resume
```

In [255…
```
score_table
```

Out[255…

	Resume	TF-IDF Score	Semantic Score
0	data-analyst-intern-resume-example.pdf	0.139	0.631
1	data-analyst-resume-example.pdf	0.153	0.622
2	entry-level-risk-adjustment-data-analyst-resum...	0.223	0.609
3	experienced-data-analyst-resume-example.pdf	0.176	0.619
4	junior-data-analyst-resume-example.pdf	0.195	0.684
5	Non_DA_Sample_1.pdf	0.016	0.281
6	Non_DA_Sample_2.pdf	0.068	0.349
7	Non_DA_Sample_3.pdf	0.017	0.475
8	revenue-reporting-data-analyst-resume-example.pdf	0.157	0.660
9	senior-data-analyst-resume-example.pdf	0.187	0.608
10	senior-insurance-data-analyst-resume-example.pdf	0.288	0.598

In [256…

```
score_table['Rank_TF-IDF'] = score_table['TF-IDF Score'].rank(ascending=Fals
score_table['Rank_Semantic_Score'] = score_table['Semantic Score'].rank(asce
```

In [257…

```
score_table
```

Out[257…

	Resume	TF-IDF Score	Semantic Score	Rank_TF-IDF	Rank_Semantic_Score
0	data-analyst-intern-resume-example.pdf	0.139	0.631	8	3
1	data-analyst-resume-example.pdf	0.153	0.622	7	4
2	entry-level-risk-adjustment-data-analyst-resum...	0.223	0.609	2	6
3	experienced-data-analyst-resume-example.pdf	0.176	0.619	5	5
4	junior-data-analyst-resume-example.pdf	0.195	0.684	3	1
5	Non_DA_Sample_1.pdf	0.016	0.281	11	11
6	Non_DA_Sample_2.pdf	0.068	0.349	9	10
7	Non_DA_Sample_3.pdf	0.017	0.475	10	9
8	revenue-reporting-data-analyst-resume-example.pdf	0.157	0.660	6	2
9	senior-data-analyst-resume-example.pdf	0.187	0.608	4	7
10	senior-insurance-data-analyst-resume-example.pdf	0.288	0.598	1	8

In [258…

```
import plotly.express as px
fig = px.scatter(score_table, x='Rank_TF-IDF', y='Rank_Semantic_Score', hove
fig.show()
```