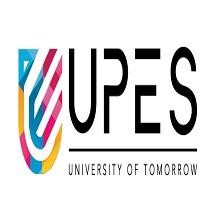
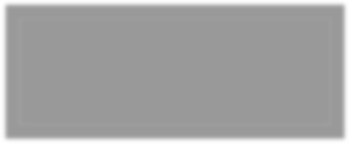
UNIVERSITY OF PETROLEUM & ENERGY STUDIES



**SCHOOL OF COMPUTER SCIENCE**

**NATURAL LANGUAGE PROCESSING PROJECT**

**NAME:** Nihar

**BATCH:** 1

**SAP ID:** 500091967

**ENROLLMENT NO.:** R2142210525

**COURSE:** B. TECH (CSE)-H

**SEMESTER:** VI

**YEAR**: III

**TOPIC**: Resume Screening using NLP and KNN

**SUBJECT:** NATURAL LANGUAGE PROCESSING

**SUBMITTED TO:** Dr. TOUSEEF IQBAL

ASSISTANT PROFESSOR,

UPES

Table of Content

|  |  |  |
| --- | --- | --- |
| **Topic** | | **Page No** |
| 1 | Introduction | 3 |
|  | 1.1 Purpose of the Project |
|  | 1.2 Target Beneficiary |
|  | 1.3 Project Scope |
| 2 | Project Description | 4-5 |
|  | 2.1 Dataset |
|  | 2.2 Functions used in code |
|  | 2.3 SWOT Analysis |
|  | 2.4 Project Features |
| 3 | System Requirements | 6 |
| 4 | Running Code | 7-11 |
|  | 3.1 CODE |
|  | 3.2 Flowchart |
| 5. | Conclusion & Future Enhancements | 12 |
|  | 4.1 Conclusion |
|  | 4.2 Future SCOPE |

Page-2

# TITLE: Resume Screening using NLP and KNN

1. **INTRODUCTION**

* **Purpose of the Project**

The purpose of my project is to develop a resume analysis system that automates the process of categorizing resumes based on job titles, checking for required skills, and analyzing the sentiment of resumes. The main aims of the project is to improve the efficiency and accuracy of the recruitment process by helping recruiters quickly identify suitable candidates for specific job roles.

* **Target Beneficiary**

The target beneficiaries of my project are recruiters, HR professionals, and hiring managers who are involved in the recruitment process. By automating the resume analysis and categorization tasks they can easily analysis which candidate is best for the given post.

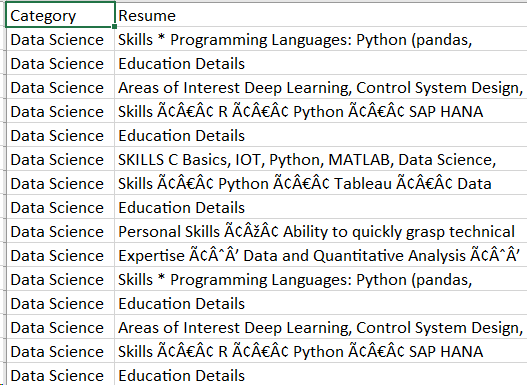
* **Project Scope**

The project scope includes automating resume categorization into job titles, checking for required skills, and analyzing sentiment of words used in recruiters Resume. It aims to provide a user an interface for easy resume upload, categorization, and analysis.

Page-3

# Project Description

## Dataset:



## Functions used in the code

The are the functions used by me in code:

* cleanResume(txt): This is a custom function you defined to clean resume text. It uses regular expressions to remove URLs, RT, cc, hashtags, mentions, special characters, non-ASCII characters, and multiple whitespaces from the text.
* word\_tokenize(text): This function from NLTK's tokenize module is used to tokenize the resume text into words.
* SentimentIntensityAnalyzer(): This function creates an instance of the SentimentIntensityAnalyzer class from NLTK's sentiment.vader module, which is used for sentiment analysis.
* sid.polarity\_scores(word)['compound']: This is used to calculate the sentiment score (specifically, the compound score) for each word in the resume text using the polarity\_scores method of the SentimentIntensityAnalyzer class.
* ‘re.findall(r'\b\w+[a-z]\*\b', myresume.lower()): This uses a regular expression to find all words in the resume text, converting them to lowercase. This is part of the process for analyzing the sentiment of words in the resume.

Page-4

* **SWOT Analysis**
* Strength:
* Automating resume screening saves time and resources for recruiters.
* Machine learning algorithms can improve the accuracy of job role predictions.
* Reducing bias in hiring decisions can lead to fairer outcomes.
* Weaknesses:
* The system may not be able to capture nuanced information or context from resumes.
* It could be challenging to ensure the system is consistently updated with the latest job market trends and requirements.
* Opportunities:
* The project could be expanded to include more advanced natural language processing (NLP) techniques for better resume analysis.
* Collaboration with HR departments and recruitment agencies could lead to the integration of the system into existing recruitment workflows.
* Threats:
* Resistance to change from traditional recruitment processes.
* Potential data privacy concerns if sensitive information is not handled properly.
* Competition from other automated recruitment solutions.
* **Project Features**

1. Resume Categorization: Utilizes machine learning to automatically categorize resumes into specific job titles or categories, aiding recruiters in efficiently sorting and organizing large volumes of resumes.
2. Skill Matching: Implements algorithms to scan resumes for specific skills and qualifications required for various job roles, helping recruiters identify candidates that closely match the job requirements.
3. Sentiment Analysis: Utilizes natural language processing techniques to analyze the sentiment of resumes, providing recruiters with insights into the tone and attitude of the candidates.

Page-5

**3 System Requirements**

**Hardware Requirements:**

• Operating System (OS): You can use Windows, macOS, or Linux (e.g., Ubuntu) based on your preference.

• Processor: Intel Core 2 Duo, Athlon X2, or better

• Memory: 8GB RAM

• Storage: Sufficient storage space for code and dataset. SSDs are preferable for faster execution and data access.

**Software Requirements:**

1. VS Code:- A versatile code editor with features for debugging, syntax highlighting, and Git integration, favored for its lightweight design and extensibility.

2. Jupyter:- A popular web-based interactive computing environment, known for its support of various programming languages, particularly Python, and its integration with data science libraries.

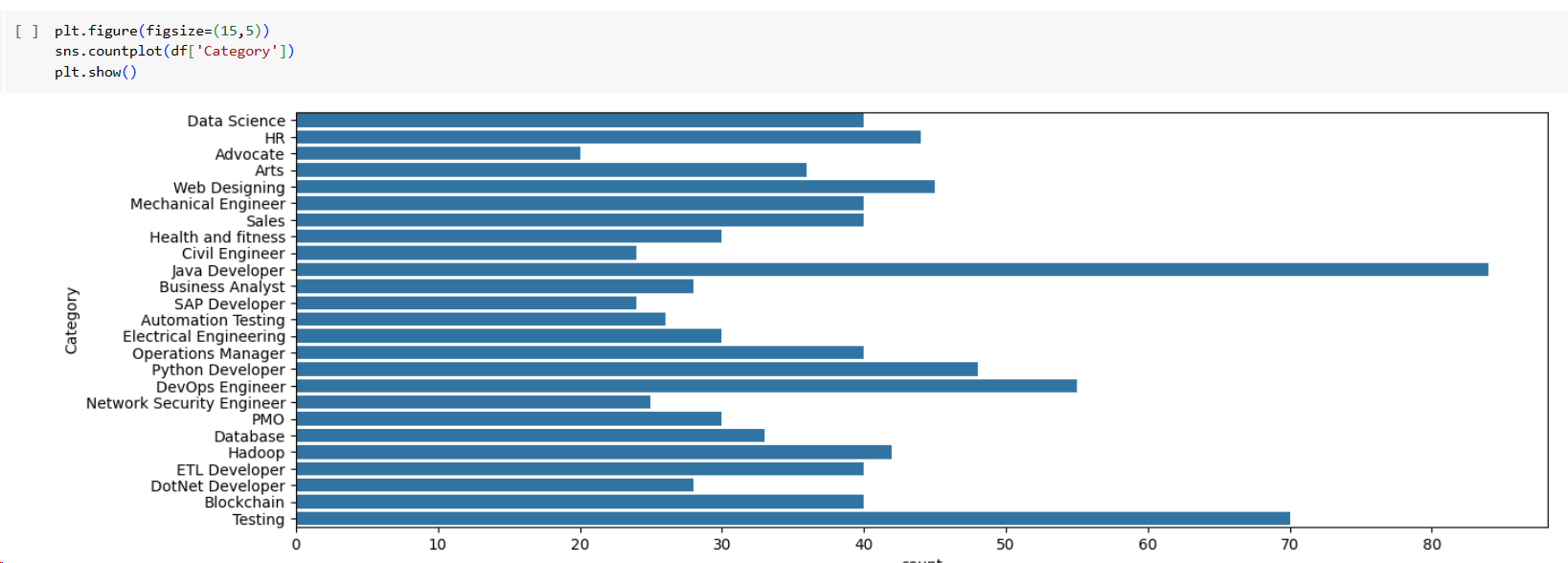
Page-6

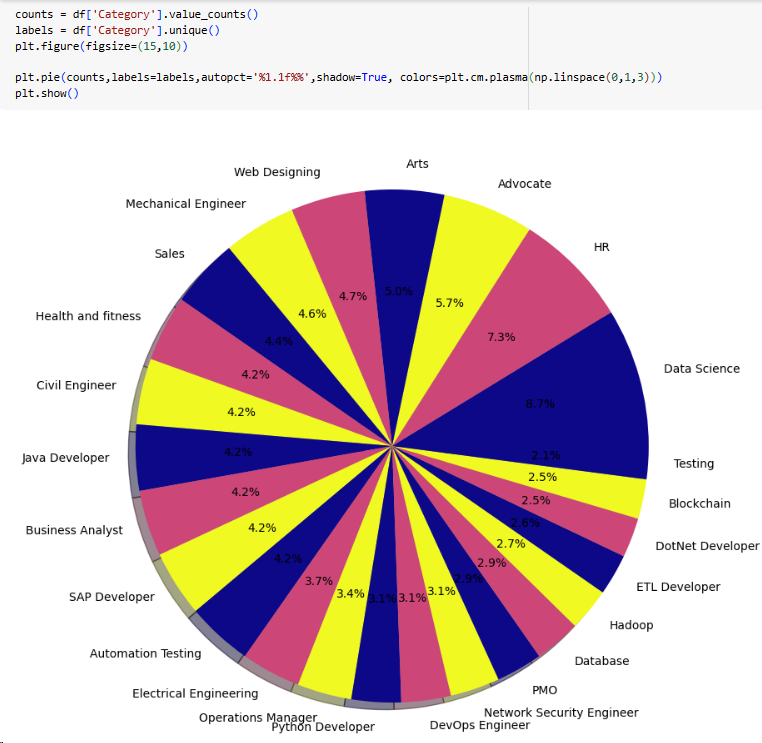
* **Running Code**
* **CODE:-**

|  |
| --- |
| **import numpy as np**  **import pandas as pd**  **import matplotlib.pyplot as plt**  **import seaborn as sns**  **import re**  **import nltk**  **from sklearn.model\_selection import train\_test\_split**  **from sklearn.neighbors import KNeighborsClassifier**  **from sklearn.multiclass import OneVsRestClassifier**  **from sklearn.metrics import accuracy\_score**  **import pickle**  **from nltk.sentiment.vader import SentimentIntensityAnalyzer**  **from nltk import word\_tokenize**  **nltk.download('punkt')**  **nltk.download('vader\_lexicon')**  **def cleanResume(txt):**  **cleanText = re.sub('http\S+\s', ' ', txt)**  **cleanText = re.sub('RT|cc', ' ', cleanText)**  **cleanText = re.sub('#\S+\s', ' ', cleanText)**  **cleanText = re.sub('@\S+', ' ', cleanText)**  **cleanText = re.sub('[%s]' % re.escape("""!"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~"""), ' ', cleanText)**  **cleanText = re.sub(r'[^\x00-\x7f]', ' ', cleanText)**  **cleanText = re.sub('\s+', ' ', cleanText)**  **return cleanText**  **df = pd.read\_csv('/content/UpdatedResumeDataSet.csv')**    Page-7  **plt.figure(figsize=(15,5))**  **sns.countplot(df['Category'])**  **plt.show()**  **df['Resume'] = df['Resume'].apply(lambda x: cleanResume(x))**  **from sklearn.feature\_extraction.text import TfidfVectorizer**  **tfidf = TfidfVectorizer(stop\_words='english')**  **tfidf.fit(df['Resume'])**  **requiredtext = tfidf.transform(df['Resume'])**  **X\_train, X\_test, y\_train, y\_test = train\_test\_split(requiredtext, df['Category'], test\_size=0.2, random\_state=42)**  **clf = OneVsRestClassifier(KNeighborsClassifier())**  **clf.fit(X\_train, y\_train)**  **ypred = clf.predict(X\_test)**  **print("Accuracy:", accuracy\_score(y\_test, ypred))**  **pickle.dump(tfidf, open('tfidf.pkl', 'wb'))**  **pickle.dump(clf, open('clf.pkl', 'wb'))**  **sid = SentimentIntensityAnalyzer()**  **resume = myresume**  **words = word\_tokenize(resume)**  **positive\_count = 0**  **negative\_count = 0**  **neutral\_count = 0**  **for word in words:**  **sentiment\_score = sid.polarity\_scores(word)['compound']**  **if sentiment\_score > 0:**  Page-8  **positive\_count += 1**  **elif sentiment\_score < 0:**  **negative\_count += 1**  **else:**  **neutral\_count += 1**  **print("Positive words:", positive\_count)**  **print("Negative words:", negative\_count)**  **print("Neutral words:", neutral\_count)** |

Page-9

* **Result:**

****

****

****

****

Page-10

### FLOWCHART

Page-11

# CONCLUSION & FUTURE SCOPE

### CONCLUSION

In conclusion, my project aims to revolutionize the recruitment process by automating key tasks such as resume categorization, skill matching, and sentiment analysis. By leveraging machine learning and natural language processing techniques, the system streamlines the screening process, saving time and effort for recruiters. The user-friendly interface and scalability make it adaptable to various recruitment scenarios. Overall, my project addresses key challenges in recruitment, providing a valuable tool for recruiters to identify and select the best candidates for job positions.

### FUTURE SCOPE

* + Implement advanced NLP techniques for deeper resume analysis, such as sentiment analysis and entity recognition.
  + Provide personalized job recommendations based on the analyzed resume data.
  + Integrate the system with HR systems and job portals, and add a feedback mechanism for continuous improvement.

Page-12