Resume/CV Parsing

Harshitha Girish – 011809582

Mahesh Kumar Srinivas - 011845961

Sharath Kumar Karnati - 011852253

Sreehari Guruprasad – 011809587

Introduction:

- Hundreds of resumes flood HR desks for each job opening, overwhelming recruiters and slowing down candidate selection.
- Resume parser/analyzers offer a solution by efficiently transforming large volumes of resume data into organized formats, extracting key details such as skills and experience.
- This not only streamlines HR processes but also aids job seekers by providing insights, identifying errors, and suggesting improvements to enhance their resumes.
- Our project aims to develop and deploy a resume parser tool using advanced technologies like NLP and machine learning to automatically extract and organize crucial data from various resume formats.

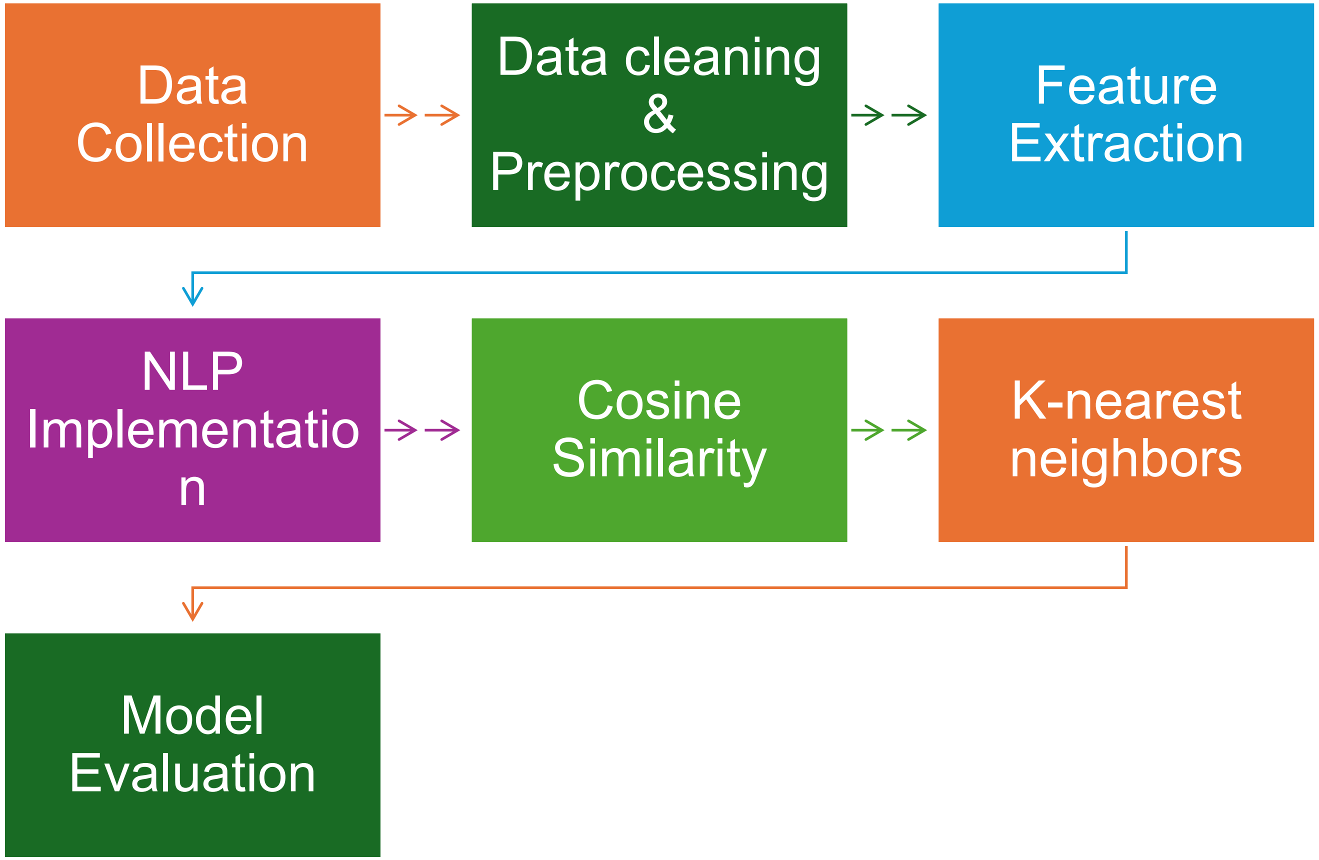
Objective:

- Our goal is to develop a tool capable of extracting essential details from resumes, including contact information, education, experience, and skills.
- This project aims to assist both recruiters and candidates by streamlining the job process, allowing them to focus their time and efforts on more productive tasks.

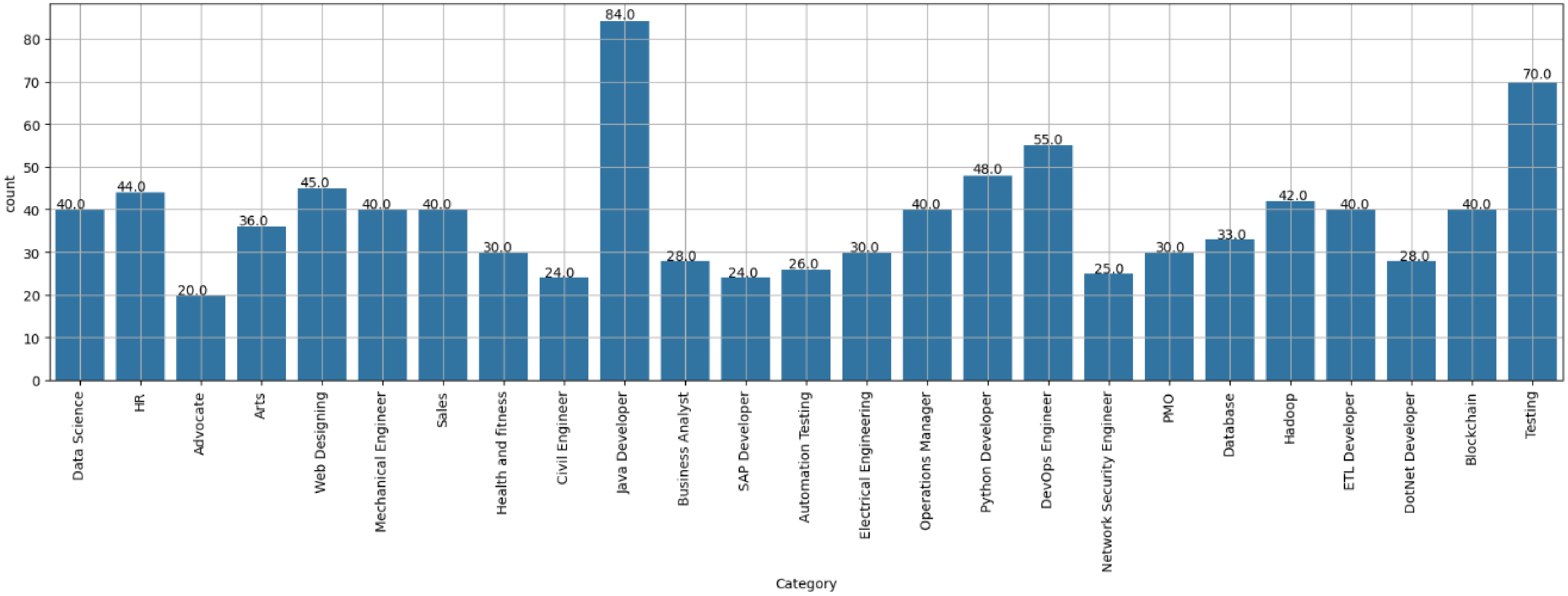
Dataset:

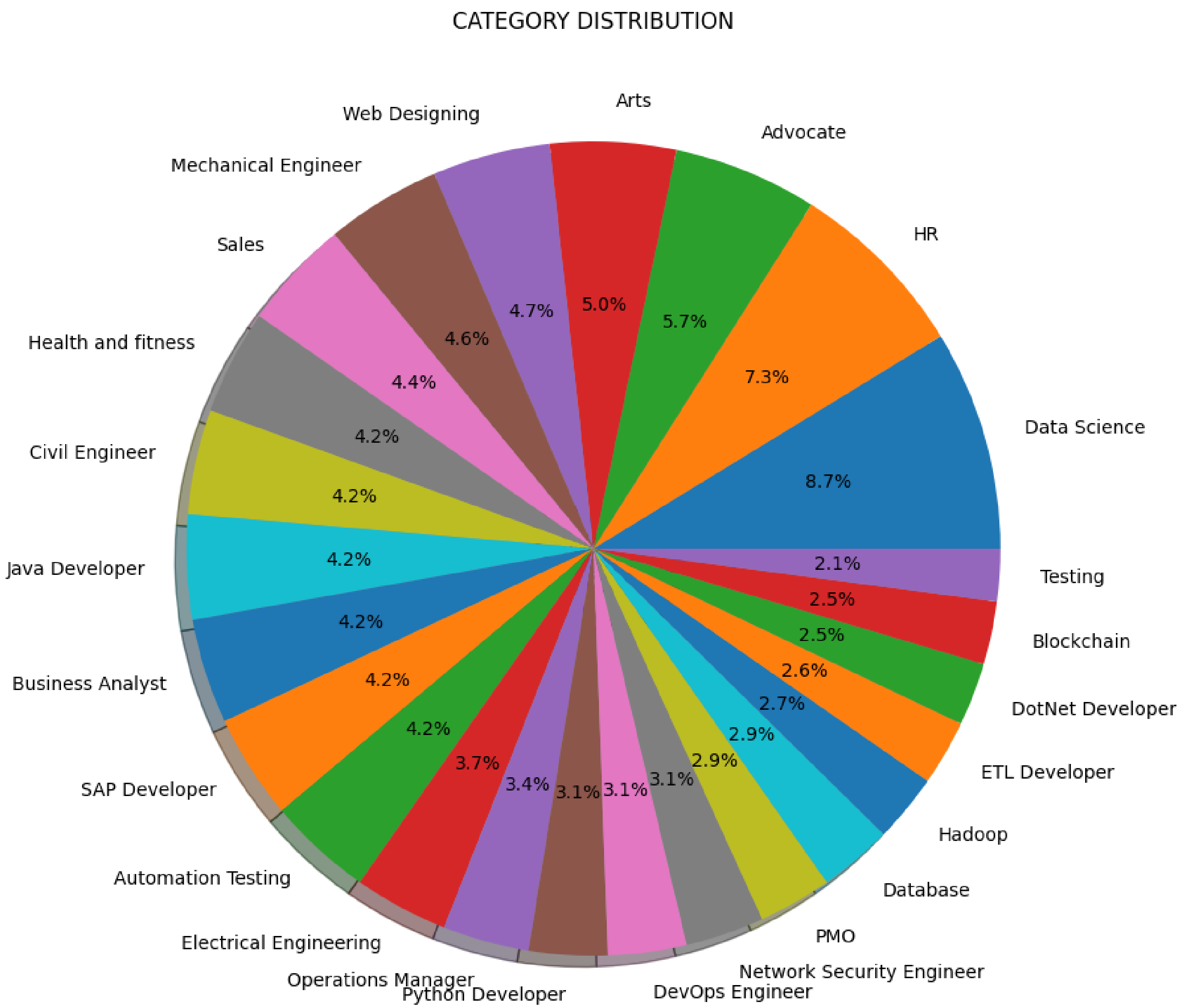
- We utilized a Kaggle dataset containing over 2400 resumes for our project.
- The process involved extracting text from PDF files using PyPDF2, converting it to lowercase, removing special characters and punctuation, tokenizing the text, eliminating stop words, lemmatizing words, and finally, vectorizing the text for machine learning analysis.
- This comprehensive approach ensured the extraction of meaningful information while standardizing and preparing the data for analysis.

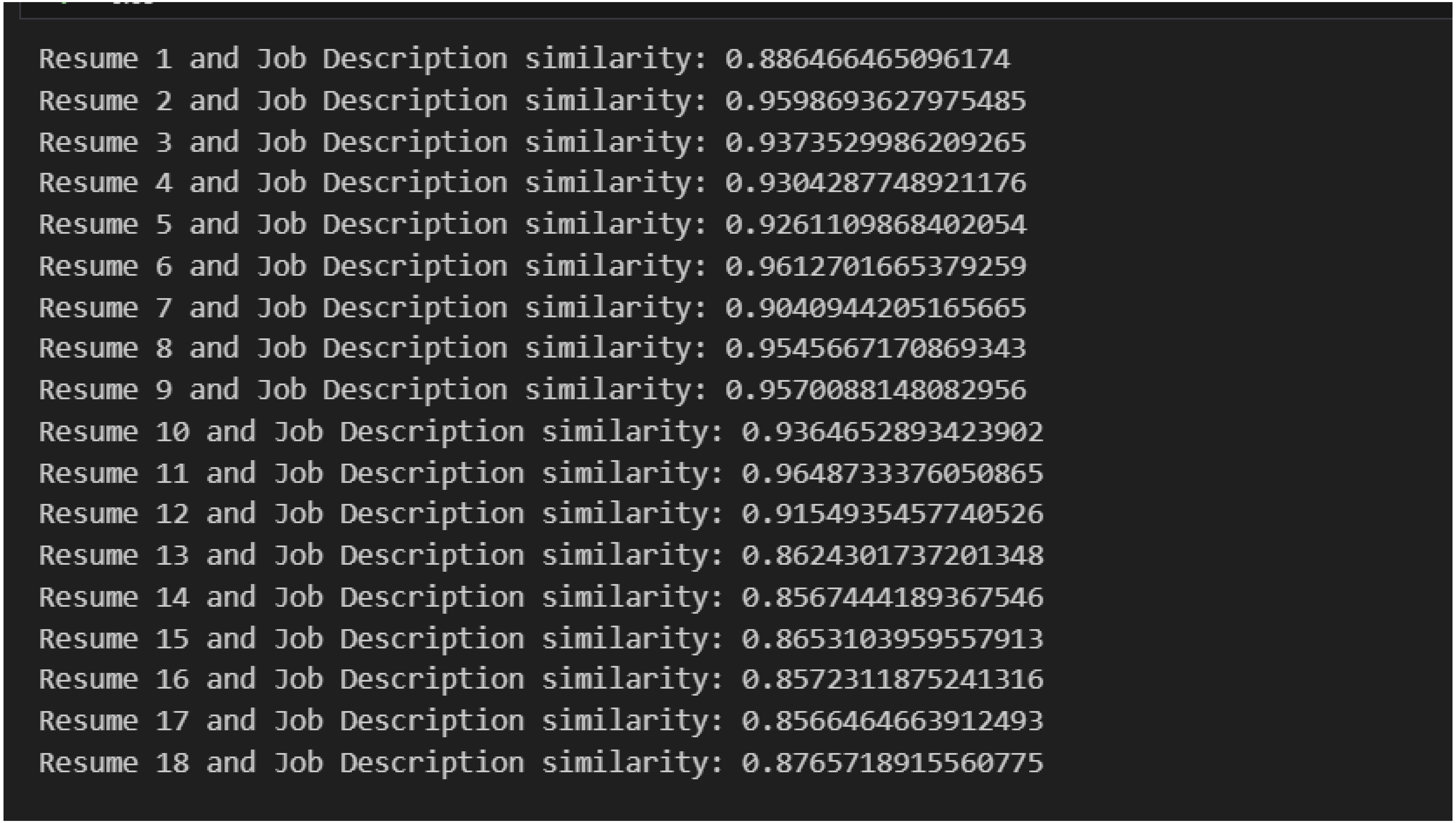
Methodolog y:



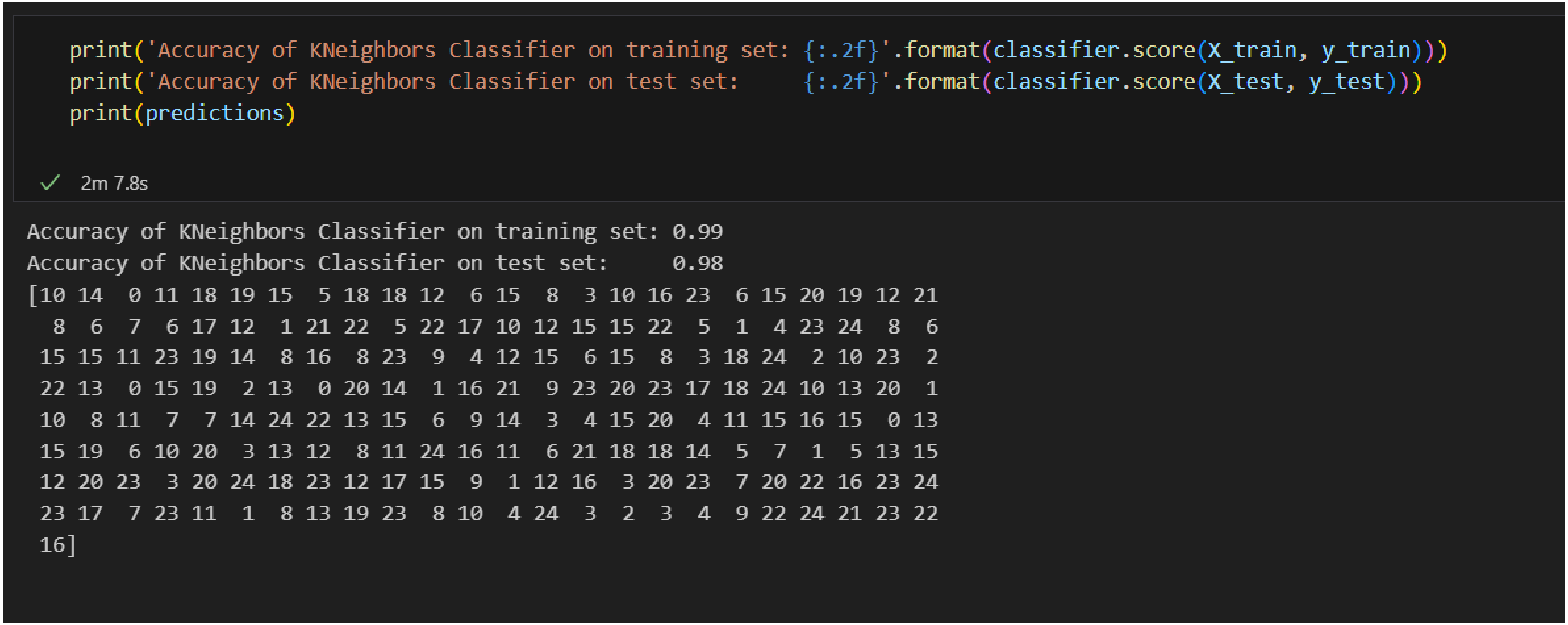
Results:


count
0
10
20
30
40
50
60
70
80
40.0
44.0
20.0
36.0
45.0
40.0
40.0
30.0
24.0
84.0
28.0
24.0
26.0
30.0
40.0
48.0
55.0
25.0
30.0
33.0
42.0
40.0
28.0
40.0
70.0
Data Science
HR
Advocate
Arts
Web Designing
Mechanical Engineer
Sales
Health and fitness
Civil Engineer
Java Developer
Business Analyst
SAP Developer
Automation Testing
Electrical Engineering
Operations Manager
Python Developer
DevOps Engineer
Network Security Engineer
PMO
Database
Hadoop
ETL Developer
DotNet Developer
Blockchain
Testing
Category


CATEGORY DISTRIBUTION
Data Science
8.7%
HR
7.3%
Advocate
5.7%
Arts
5.0%
Web Designing
4.7%
Mechanical Engineer
4.6%
Sales
4.4%
Health and fitness
4.2%
Civil Engineer
4.2%
Java Developer
4.2%
Business Analyst
4.2%
SAP Developer
4.2%
Automation Testing
4.2%
Electrical Engineering
3.7%
Operations Manager
3.4%
Python Developer
3.1%
DevOps Engineer
3.1%
Network Security Engineer
3.1%
PMO
2.9%
Database
2.9%
Hadoop
2.7%
ETL Developer
2.6%
DotNet Developer
2.5%
Blockchain
2.5%
Testing
2.1%



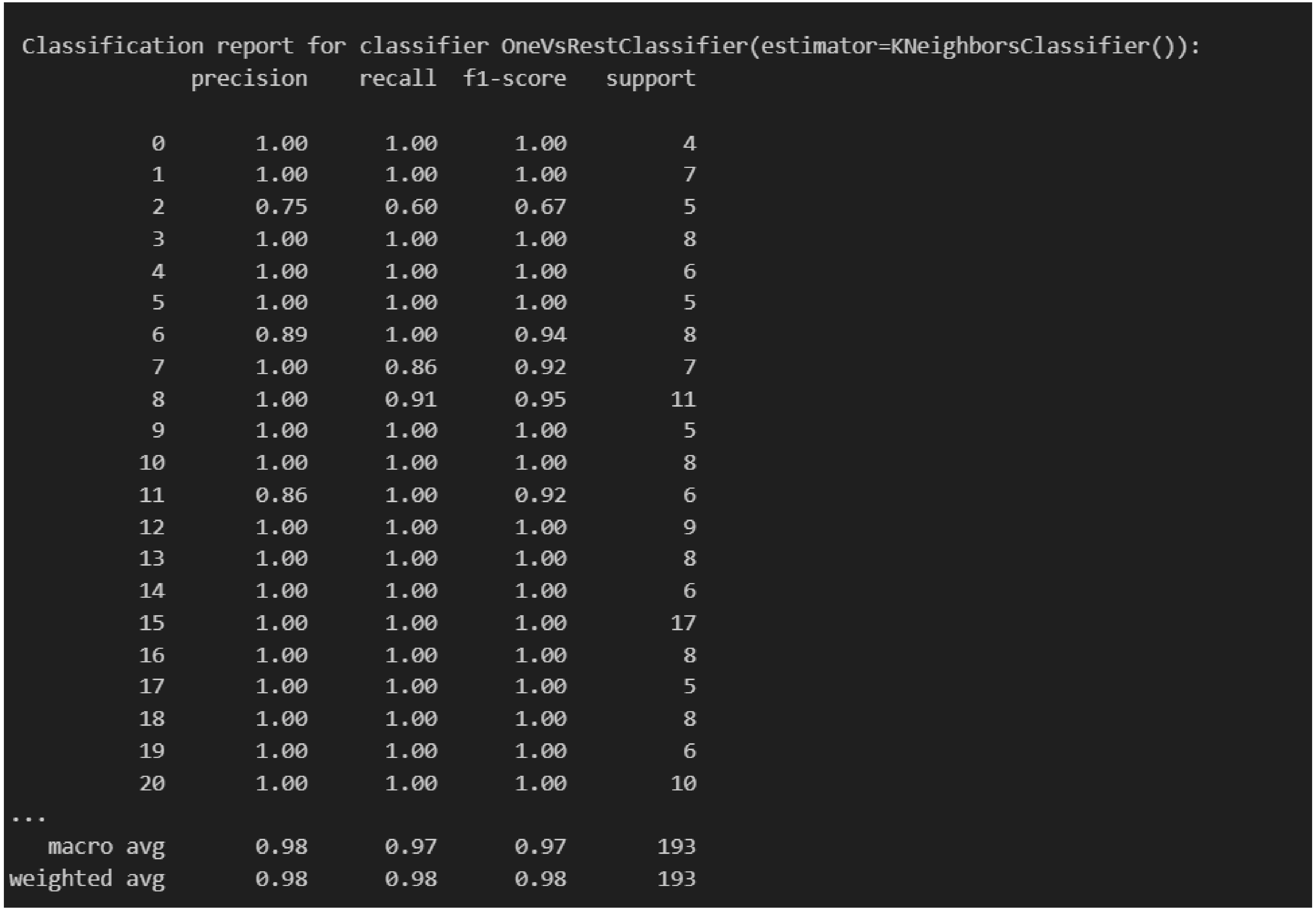
```
Resume 1 and Job Description similarity: 0.886466465096174
Resume 2 and Job Description similarity: 0.9598693627975485
Resume 3 and Job Description similarity: 0.9373529986209265
Resume 4 and Job Description similarity: 0.9304287748921176
Resume 5 and Job Description similarity: 0.9261109868402054
Resume 6 and Job Description similarity: 0.9612701665379259
Resume 7 and Job Description similarity: 0.9040944205165665
Resume 8 and Job Description similarity: 0.9545667170869343
Resume 9 and Job Description similarity: 0.9570088148082956
Resume 10 and Job Description similarity: 0.9364652893423902
Resume 11 and Job Description similarity: 0.9648733376050865
Resume 12 and Job Description similarity: 0.9154935457740526
Resume 13 and Job Description similarity: 0.8624301737201348
Resume 14 and Job Description similarity: 0.8567444189367546
Resume 15 and Job Description similarity: 0.8653103959557913
Resume 16 and Job Description similarity: 0.8572311875241316
Resume 17 and Job Description similarity: 0.8566464663912493
Resume 18 and Job Description similarity: 0.8765718915560775
```



```
print('Accuracy of KNeighbors Classifier on training set: {:.2f}'.format(classifier.score(X_train, y_train)))
print('Accuracy of KNeighbors Classifier on test set:     {:.2f}'.format(classifier.score(X_test, y_test)))
print(predictions)
```

✓ 2m 7.8s

```
Accuracy of KNeighbors Classifier on training set: 0.99
Accuracy of KNeighbors Classifier on test set:     0.98
[10 14  0 11 18 19 15  5 18 18 12  6 15  8  3 10 16 23  6 15 20 19 12 21
  8  6  7  6 17 12  1 21 22  5 22 17 10 12 15 15 22  5  1  4 23 24  8  6
 15 15 11 23 19 14  8 16  8 23  9  4 12 15  6 15  8  3 18 24  2 10 23  2
 22 13  0 15 19  2 13  0 20 14  1 16 21  9 23 20 23 17 18 24 10 13 20  1
 10  8 11  7  7 14 24 22 13 15  6  9 14  3  4 15 20  4 11 15 16 15  0 13
 15 19  6 10 20  3 13 12  8 11 24 16 11  6 21 18 18 14  5  7  1  5 13 15
 12 20 23  3 20 24 18 23 12 17 15  9  1 12 16  3 20 23  7 20 22 16 23 24
 23 17  7 23 11  1  8 13 19 23  8 10  4 24  3  2  3  4  9 22 24 21 23 22
 16]
```



```
 Classification report for classifier OneVsRestClassifier(estimator=KNeighborsClassifier()):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           1       1.00      1.00      1.00         7
           2       0.75      0.60      0.67         5
           3       1.00      1.00      1.00         8
           4       1.00      1.00      1.00         6
           5       1.00      1.00      1.00         5
           6       0.89      1.00      0.94         8
           7       1.00      0.86      0.92         7
           8       1.00      0.91      0.95        11
           9       1.00      1.00      1.00         5
          10       1.00      1.00      1.00         8
          11       0.86      1.00      0.92         6
          12       1.00      1.00      1.00         9
          13       1.00      1.00      1.00         8
          14       1.00      1.00      1.00         6
          15       1.00      1.00      1.00        17
          16       1.00      1.00      1.00         8
          17       1.00      1.00      1.00         5
          18       1.00      1.00      1.00         8
          19       1.00      1.00      1.00         6
          20       1.00      1.00      1.00        10
...
   macro avg       0.98      0.97      0.97       193
weighted avg       0.98      0.98      0.98       193
```

Future Scope/Conclusion:

- In conclusion, this project results show that the effectiveness of the resume parser in helping both the efficiency and accuracy of screening process. By using Spacy for Natural Processing Language, Cosine Similarity for resume ranking and K-Nearest Neighbors for candidate matching, our application outperforms the traditional way of resume screening.
- Moving forward, our main aim is to build a web application that can be user friendly. We also plan to incorporate additional customizable features and filters. Overall, this approach is a promising opportunity to transform the manual screening to a more automated task.

Reference

- https://www.irjmets.com/uploadedfiles/paper//issue_4_april_2023/37428/final/fin_irjmets1683342426.pdf
- https://www.researchgate.net/publication/361772014_RESUME_PARSER
- https://arxiv.org/abs/1301.3781

THANK YOU