RESUME PARSER

A MINI PROJECT REPORT

Submitted by

BRYAN ABRAHAM (Reg No:RA2111003010612)
PARTH AGARWAL (Reg No:RA2111003010608)
PULKIT SHRINGI (Reg No:RA2111003010596)

Under the Guidance of

Ms.P.Nithyakani

Associate Professor, Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in Computer Science and Engineering

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

May 2024

Department of Computational Intelligence
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course **: Bachelors of Technology/18CSC305J**

Student Name **: Pulkit Shringi**

Registration Number **: RA2111003010596**

Title of Work **: Resume Parser**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSC305J project report titled "**RESUME PARSER**" is the bonafide work of "Pulkit Shringi (RA2111003010596), Parth Agarwal (RA2111003010608) and Bryan Abraham (RA2111003010612)" who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

MS.P.NITHYAKINI

ASSISTANT PROFESSOR
DEPARTMENT OF VISUAL
COMPUTING

SIGNATURE

DR. R. ANNIE UTHRA

HEAD OF THE DEPARTMENT
COMPUTATIONAL INTELLIGENCE

ABSTRACT

In today's digital age, the volume of job applications inundating HR departments necessitates efficient methods for resume screening and parsing. The Automated Resume Parsing System (ARPS) is a cutting-edge solution designed to streamline the recruitment process by automating the extraction and analysis of pertinent information from resumes. Leveraging natural language processing (NLP) techniques and machine learning algorithms, ARPS can accurately interpret diverse resume formats and extract key details such as personal information, education, work experience, skills, and qualifications. The system employs advanced parsing algorithms to categorize and structure extracted data, enabling recruiters to swiftly assess candidate suitability and match them with job requirements. ARPS offers scalability, adaptability, and customization options to cater to the specific needs of various industries and organizations. By significantly reducing manual effort and time spent on resume screening, ARPS empowers HR professionals to focus on strategic tasks, enhances recruitment efficiency, and ultimately facilitates the identification of top talent for organizational success.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In the dynamic landscape of modern recruitment, the process of sifting through voluminous stacks of resumes to identify suitable candidates remains a daunting challenge for hiring managers and HR professionals alike. As organizations strive to streamline their hiring processes and identify top talent swiftly, the integration of artificial intelligence (AI) technologies has emerged as a pivotal solution. Among these, AI-powered resume parsing systems stand out as innovative tools designed to automate and enhance the initial stages of candidate evaluation.

This project work aims to provide a comprehensive review of AI-powered resume parsing systems, examining their functionalities, advantages, limitations, and potential impact on recruitment efficiency. By leveraging natural language processing (NLP), machine learning (ML), and data mining techniques, these systems offer a sophisticated means of extracting, analyzing, and categorizing relevant information from resumes, thereby facilitating faster and more accurate candidate screening.

The proliferation of AI-driven resume parsing tools has sparked considerable interest across industries, as organizations seek to leverage technology to optimize their talent acquisition processes. However, while these systems hold great promise, they also pose challenges and considerations that warrant careful examination. Factors such as accuracy, bias mitigation, scalability, and integration with existing recruitment workflows are critical aspects that must be addressed to maximize the effectiveness of AI-powered resume parsing solutions.Through a thorough exploration of existing literature, case studies, and industry insights, this project work aims to shed light on the current state-of-the-art in AI-driven resume parsing technology. Furthermore, it seeks to identify emerging trends, best practices, and areas for future research and development, with the ultimate goal of empowering organizations to make informed decisions regarding the adoption and implementation of these innovative tools.

In summary, as the demand for talent continues to grow in an increasingly competitive global market, the integration of AI-powered resume parsing systems represents a strategic imperative for organizations seeking to gain a competitive edge in recruitment. By harnessing the power of AI to automate and optimize candidate screening processes, organizations can not only expedite hiring cycles but also ensure fair and unbiased evaluations, ultimately driving greater efficiency and effectiveness in talent acquisition endeavors.

2. ARCHITECTURE & DESIGN

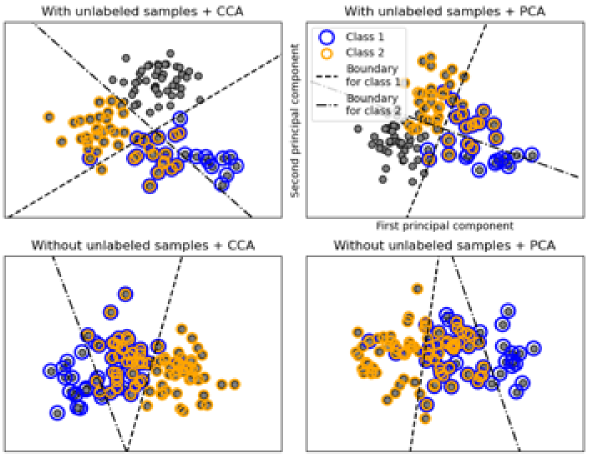


Fig [2.1] OneVsRestClassifier Model Architecture

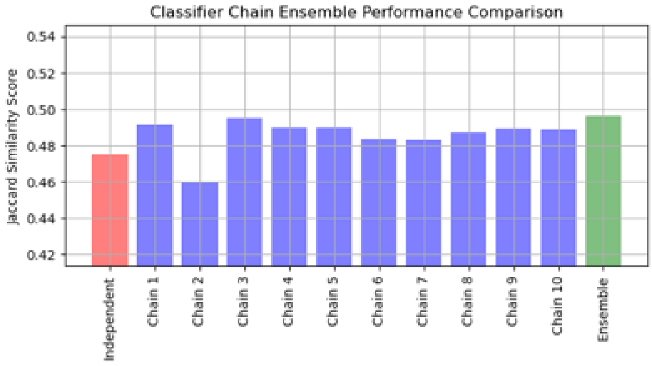
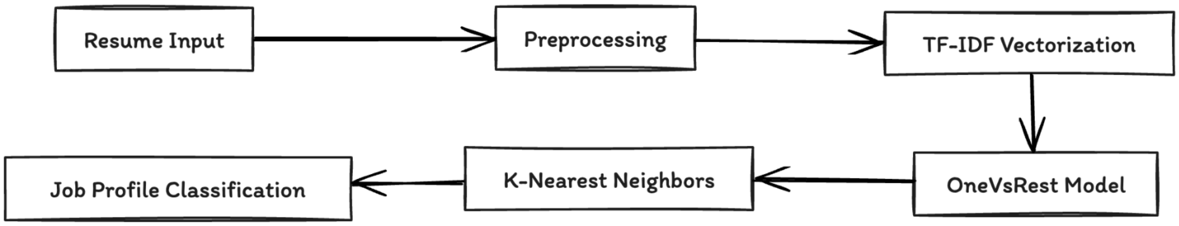


Fig [2.2] Multi Label classification using classifier chain



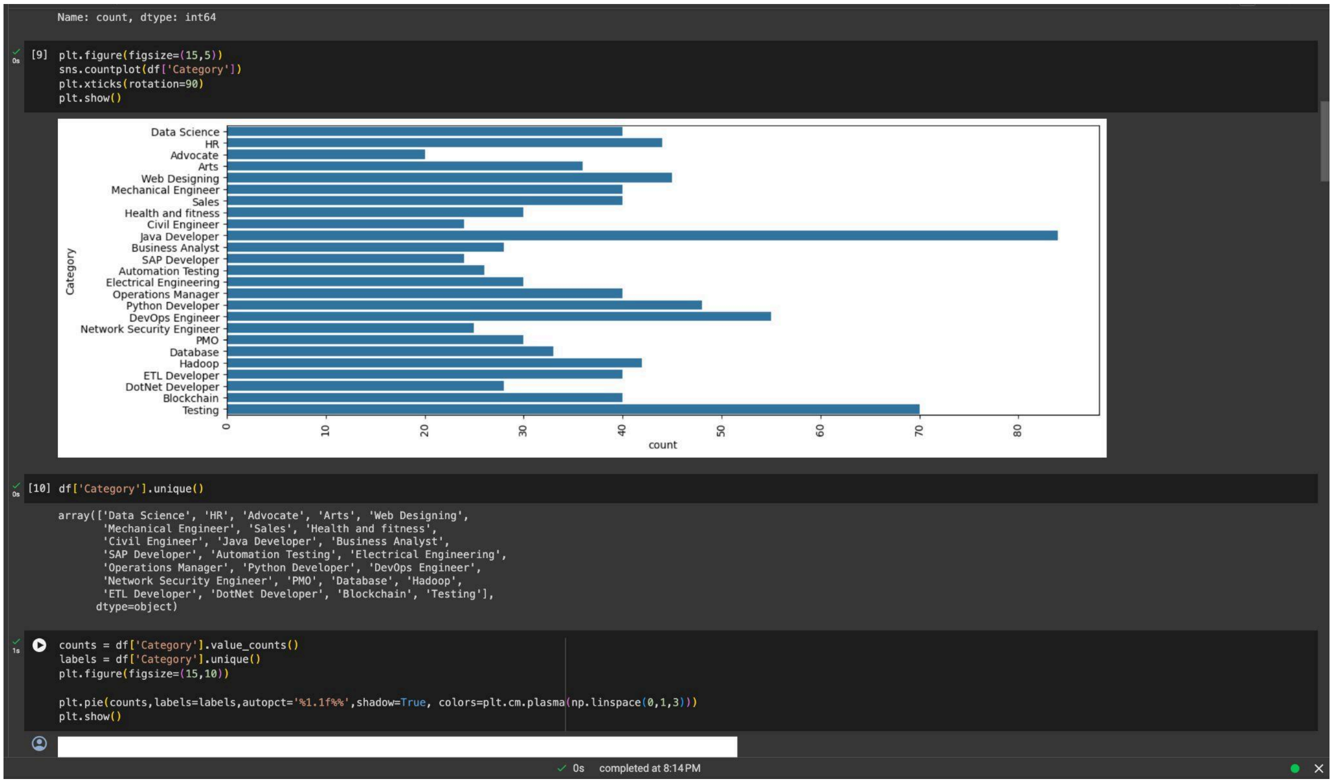
Fig[2.3] Project Model Architecture

Our resume parsing system utilizes a powerful blend of NLP and machine learning, employing the one-vs-rest algorithm alongside TF-IDF vectorization. This combination enables precise extraction of candidate details and skills from resumes, facilitating streamlined analysis for faster and more informed hiring decisions. By leveraging these advanced techniques, we ensure efficient handling of diverse resume formats and extraction of invaluable insights from candidate profiles.

3. IMPLEMENTATION

Code Implementation:

GitHub Repository Link: Github



```
Name: count, dtype: int64
```

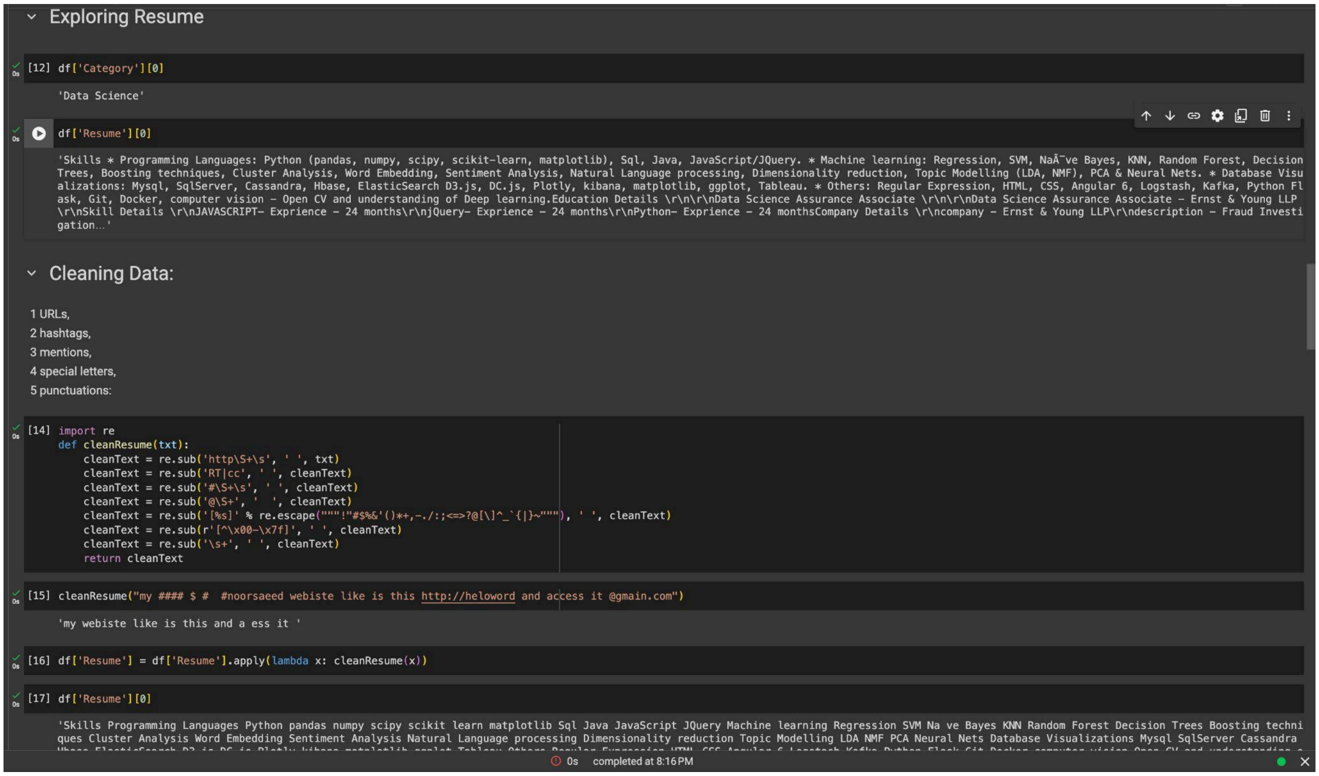
```
[9] plt.figure(figsize=(15,5))
    sns.countplot(df['Category'])
    plt.xticks(rotation=90)
    plt.show()
```

```
[10] df['Category'].unique()
```

```
array(['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'],
      dtype=object)
```

```
counts = df['Category'].value_counts()
labels = df['Category'].unique()
plt.figure(figsize=(15,10))

plt.pie(counts,labels=labels,autopct='%1.1f%%',shadow=True, colors=plt.cm.plasma(np.linspace(0,1,3)))
plt.show()
```

Fig[3.1] Data Analysis



Exploring Resume

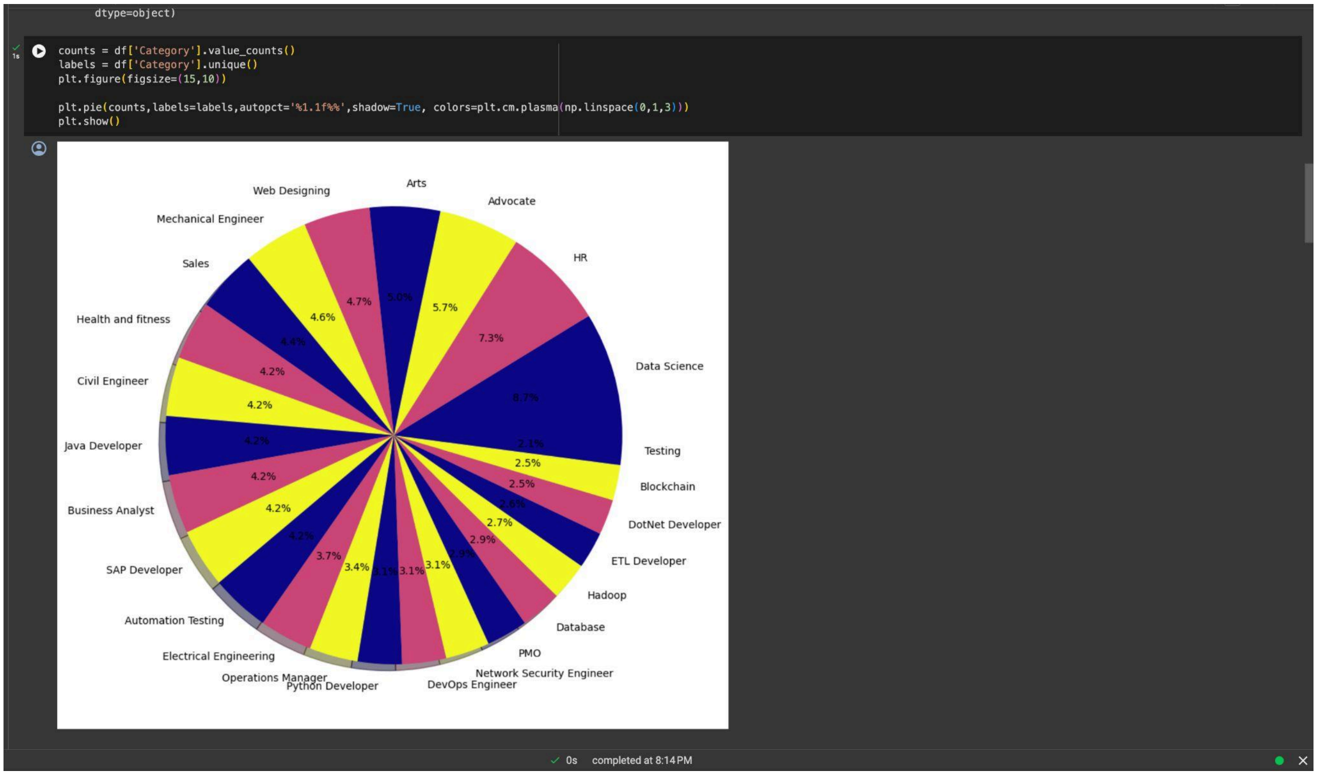
```
[12] df['Category'][0]

     'Data Science'

df['Resume'][0]

     'Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃ¯ve Bayes, KNN, Random Forest, Decision
     Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language processing, Dimensionality reduction, Topic Modelling (LDA, NMF), PCA & Neural Nets. * Database Visu
     alizations: Mysql, SqlServer, Cassandra, Hbase, ElasticSearch D3.js, DC.js, Plotly, kibana, matplotlib, ggplot, Tableau. * Others: Regular Expression, HTML, CSS, Angular 6, Logstash, Kafka, Python Fl
     ask, Git, Docker, computer vision - Open CV and understanding of Deep learning.Education Details \r\n\r\nData Science Assurance Associate \r\n\r\nData Science Assurance Associate - Ernst & Young LLP
     \r\nSkill Details \r\nJAVASCRIPT- Exprience - 24 months\r\njQuery- Exprience - 24 months\r\nPython- Exprience - 24 monthsCompany Details \r\ncompany - Ernst & Young LLP\r\ndescription - Fraud Investi
     gation...'
```

Cleaning Data:

1 URLs,
2 hashtags,
3 mentions,
4 special letters,
5 punctuations:

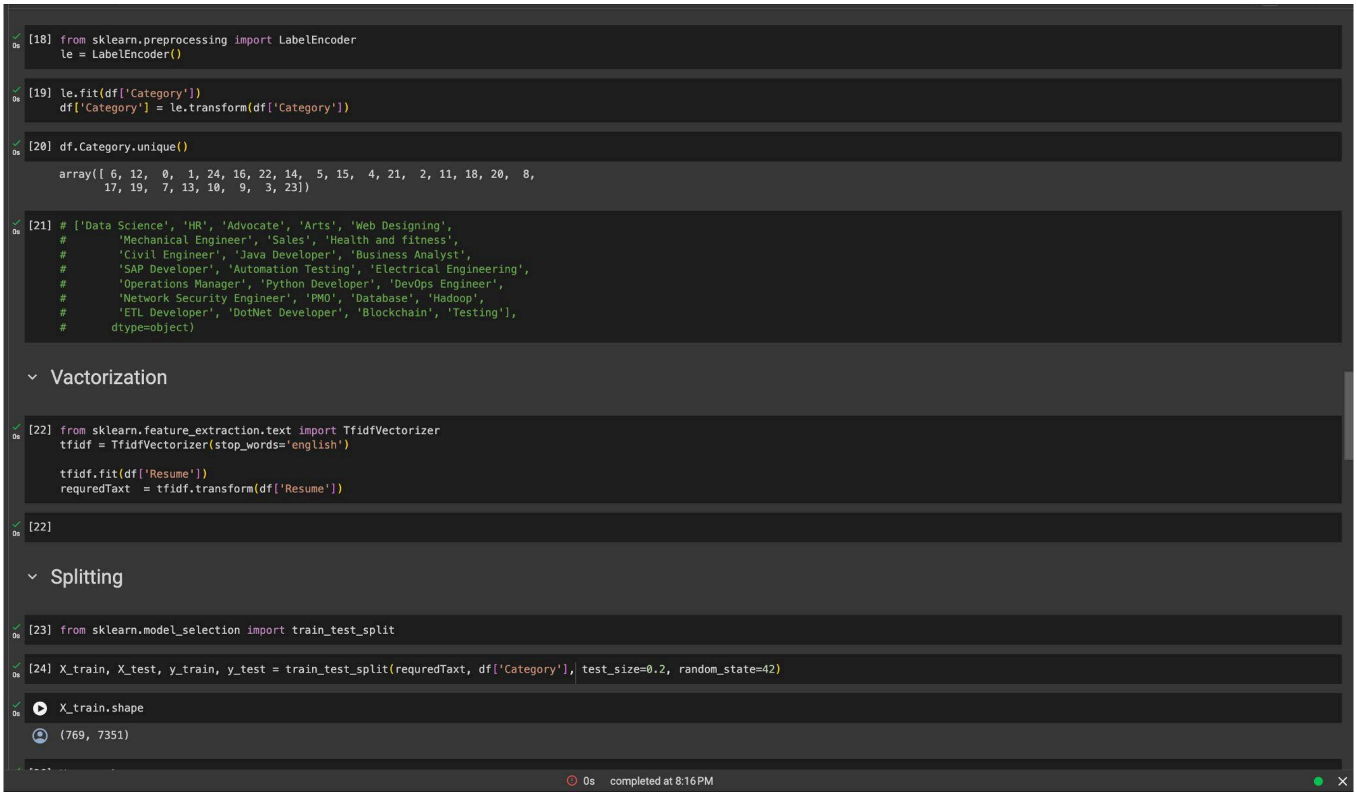
```
[14] import re
     def cleanResume(txt):
         cleanText = re.sub('http\S+\s', ' ', txt)
         cleanText = re.sub('RT|cc', ' ', cleanText)
         cleanText = re.sub('#\S+\s', ' ', cleanText)
         cleanText = re.sub('@\S+', '  ', cleanText)
         cleanText = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), ' ', cleanText)
         cleanText = re.sub(r'[^\x00-\x7f]', ' ', cleanText)
         cleanText = re.sub('\s+', ' ', cleanText)
         return cleanText

[15] cleanResume("my #### $ #  #noorsaeed webiste like is this http://heloword and access it @gmain.com")

     'my webiste like is this and a ess it '

[16] df['Resume'] = df['Resume'].apply(lambda x: cleanResume(x))

[17] df['Resume'][0]

     'Skills Programming Languages Python pandas numpy scipy scikit learn matplotlib Sql Java JavaScript JQuery Machine learning Regression SVM Na ve Bayes KNN Random Forest Decision Trees Boosting techni
     ques Cluster Analysis Word Embedding Sentiment Analysis Natural Language processing Dimensionality reduction Topic Modelling LDA NMF PCA Neural Nets Database Visualizations Mysql SqlServer Cassandra
```

Fig[3.2] Data Cleaning



```
         dtype=object)

counts = df['Category'].value_counts()
labels = df['Category'].unique()
plt.figure(figsize=(15,10))

plt.pie(counts,labels=labels,autopct='%1.1f%%',shadow=True, colors=plt.cm.plasma(np.linspace(0,1,3)))
plt.show()
```

Fig[3.3] Data Interpretation



```
[18] from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()

[19] le.fit(df['Category'])
     df['Category'] = le.transform(df['Category'])

[20] df.Category.unique()

     array([ 6, 12,  0,  1, 24, 16, 22, 14,  5, 15,  4, 21,  2, 11, 18, 20,  8,
            17, 19,  7, 13, 10,  9,  3, 23])

[21] # ['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'],
     #       dtype=object)
```

Vactorization

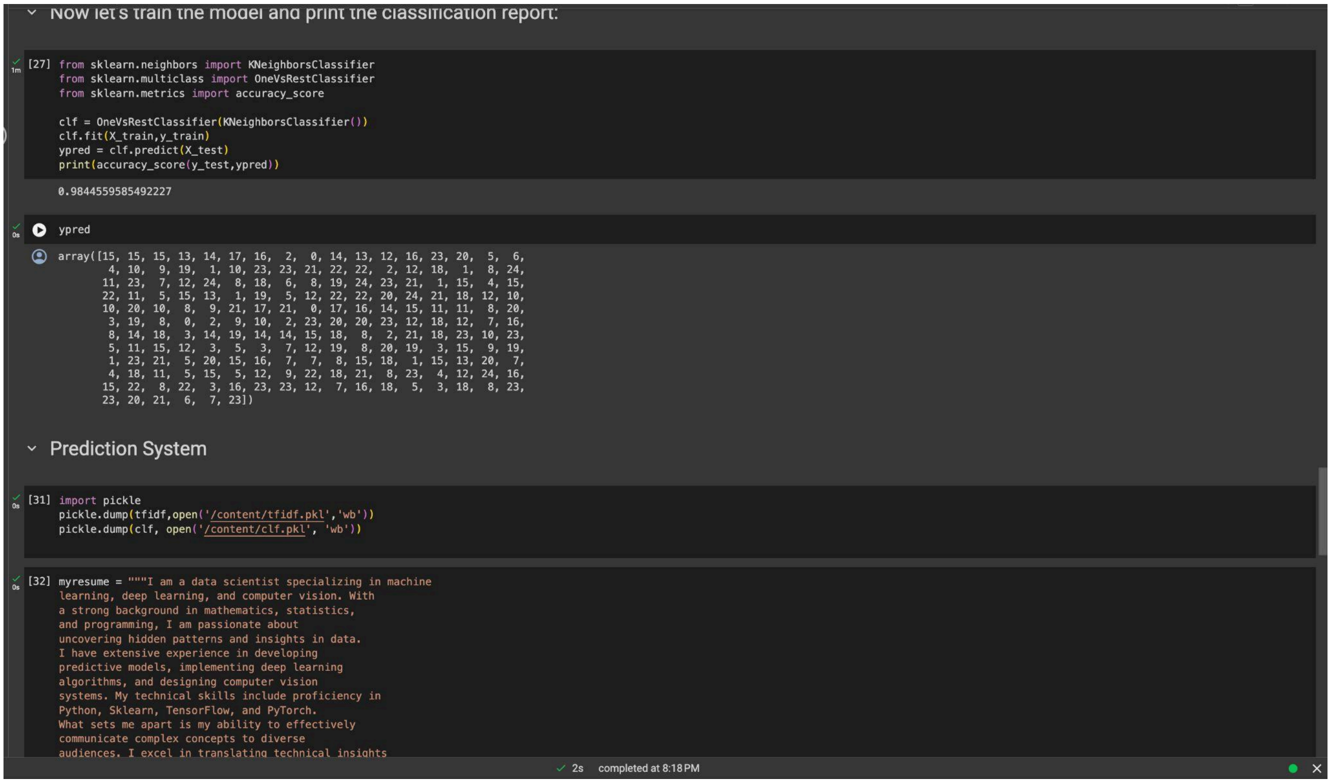
```
[22] from sklearn.feature_extraction.text import TfidfVectorizer
     tfidf = TfidfVectorizer(stop_words='english')

     tfidf.fit(df['Resume'])
     requredTaxt  = tfidf.transform(df['Resume'])

[22]
```

Splitting

```
[23] from sklearn.model_selection import train_test_split

[24] X_train, X_test, y_train, y_test = train_test_split(requredTaxt, df['Category'], test_size=0.2, random_state=42)

     X_train.shape

     (769, 7351)
```

0s completed at 8:16 PM

Fig[3.4] Vectorization of input data using TF-IDF



Now let's train the model and print the classification report:

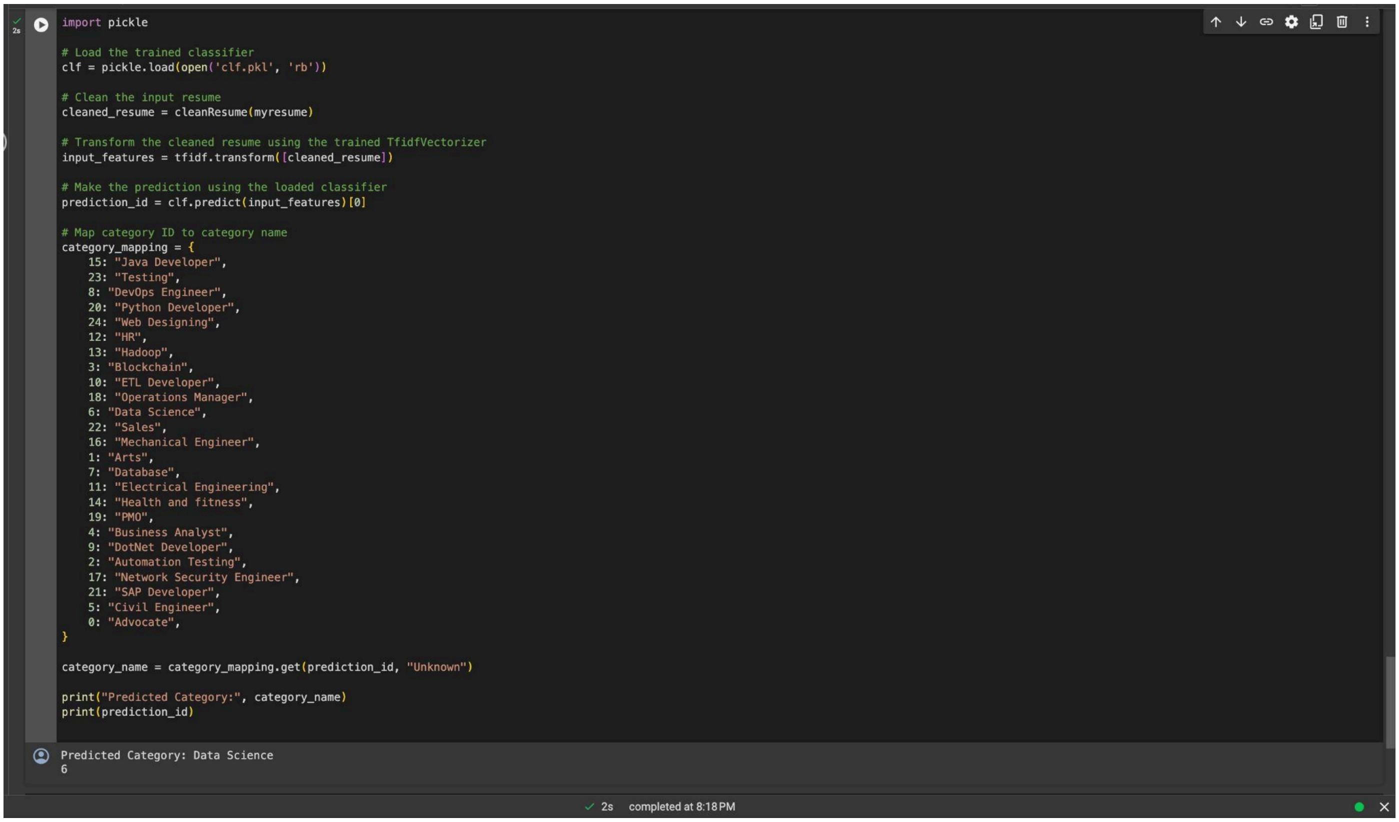
```
[27] from sklearn.neighbors import KNeighborsClassifier
     from sklearn.multiclass import OneVsRestClassifier
     from sklearn.metrics import accuracy_score

     clf = OneVsRestClassifier(KNeighborsClassifier())
     clf.fit(X_train,y_train)
     ypred = clf.predict(X_test)
     print(accuracy_score(y_test,ypred))

     0.9844559585492227

     ypred

     array([15, 15, 15, 13, 14, 17, 16,  2,  0, 14, 13, 12, 16, 23, 20,  5,  6,
             4, 10,  9, 19,  1, 10, 23, 23, 21, 22, 22,  2, 12, 18,  1,  8, 24,
            11, 23,  7, 12, 24,  8, 18,  6,  8, 19, 24, 23, 21,  1, 15,  4, 15,
            22, 11,  5, 15, 13,  1, 19,  5, 12, 22, 22, 20, 24, 21, 18, 12, 10,
            10, 20, 10,  8,  9, 21, 17, 21,  0, 17, 16, 14, 15, 11, 11,  8, 20,
             3, 19,  8,  0,  2,  9, 10,  2, 23, 20, 20, 23, 12, 18, 12,  7, 16,
             8, 14, 18,  3, 14, 19, 14, 14, 15, 18,  8,  2, 21, 18, 23, 10, 23,
             5, 11, 15, 12,  3,  5,  3,  7, 12, 19,  8, 20, 19,  3, 15,  9, 19,
             1, 23, 21,  5, 20, 15, 16,  7,  7,  8, 15, 18,  1, 15, 13, 20,  7,
             4, 18, 11,  5, 15,  5, 12,  9, 22, 18, 21,  8, 23,  4, 12, 24, 16,
            15, 22,  8, 22,  3, 16, 23, 23, 12,  7, 16, 18,  5,  3, 18,  8, 23,
            23, 20, 21,  6,  7, 23])
```

Prediction System

```
[31] import pickle
     pickle.dump(tfidf,open('/content/tfidf.pkl','wb'))
     pickle.dump(clf, open('/content/clf.pkl', 'wb'))

[32] myresume = """I am a data scientist specializing in machine
     learning, deep learning, and computer vision. With
     a strong background in mathematics, statistics,
     and programming, I am passionate about
     uncovering hidden patterns and insights in data.
     I have extensive experience in developing
     predictive models, implementing deep learning
     algorithms, and designing computer vision
     systems. My technical skills include proficiency in
     Python, Sklearn, TensorFlow, and PyTorch.
     What sets me apart is my ability to effectively
     communicate complex concepts to diverse
     audiences. I excel in translating technical insights
```

✓ 2s completed at 8:18 PM

Fig[3.5] Model Training

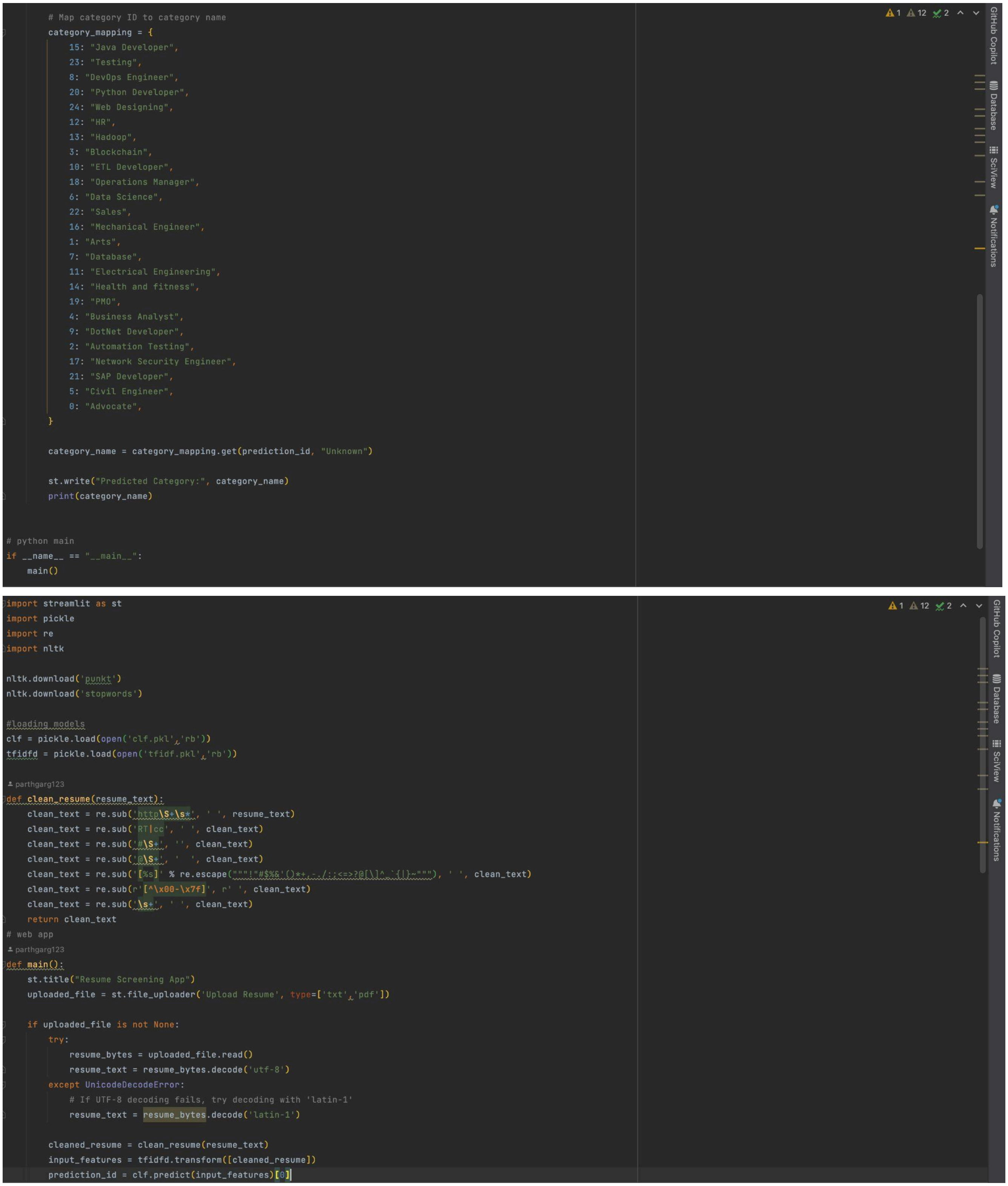


```python
import pickle

# Load the trained classifier
clf = pickle.load(open('clf.pkl', 'rb'))

# Clean the input resume
cleaned_resume = cleanResume(myresume)

# Transform the cleaned resume using the trained TfidfVectorizer
input_features = tfidf.transform([cleaned_resume])

# Make the prediction using the loaded classifier
prediction_id = clf.predict(input_features)[0]

# Map category ID to category name
category_mapping = {
    15: "Java Developer",
    23: "Testing",
    8: "DevOps Engineer",
    20: "Python Developer",
    24: "Web Designing",
    12: "HR",
    13: "Hadoop",
    3: "Blockchain",
    10: "ETL Developer",
    18: "Operations Manager",
    6: "Data Science",
    22: "Sales",
    16: "Mechanical Engineer",
    1: "Arts",
    7: "Database",
    11: "Electrical Engineering",
    14: "Health and fitness",
    19: "PMO",
    4: "Business Analyst",
    9: "DotNet Developer",
    2: "Automation Testing",
    17: "Network Security Engineer",
    21: "SAP Developer",
    5: "Civil Engineer",
    0: "Advocate",
}

category_name = category_mapping.get(prediction_id, "Unknown")

print("Predicted Category:", category_name)
print(prediction_id)
```

```
Predicted Category: Data Science
6
```

Fig [3.6] Model Execution

App.py:



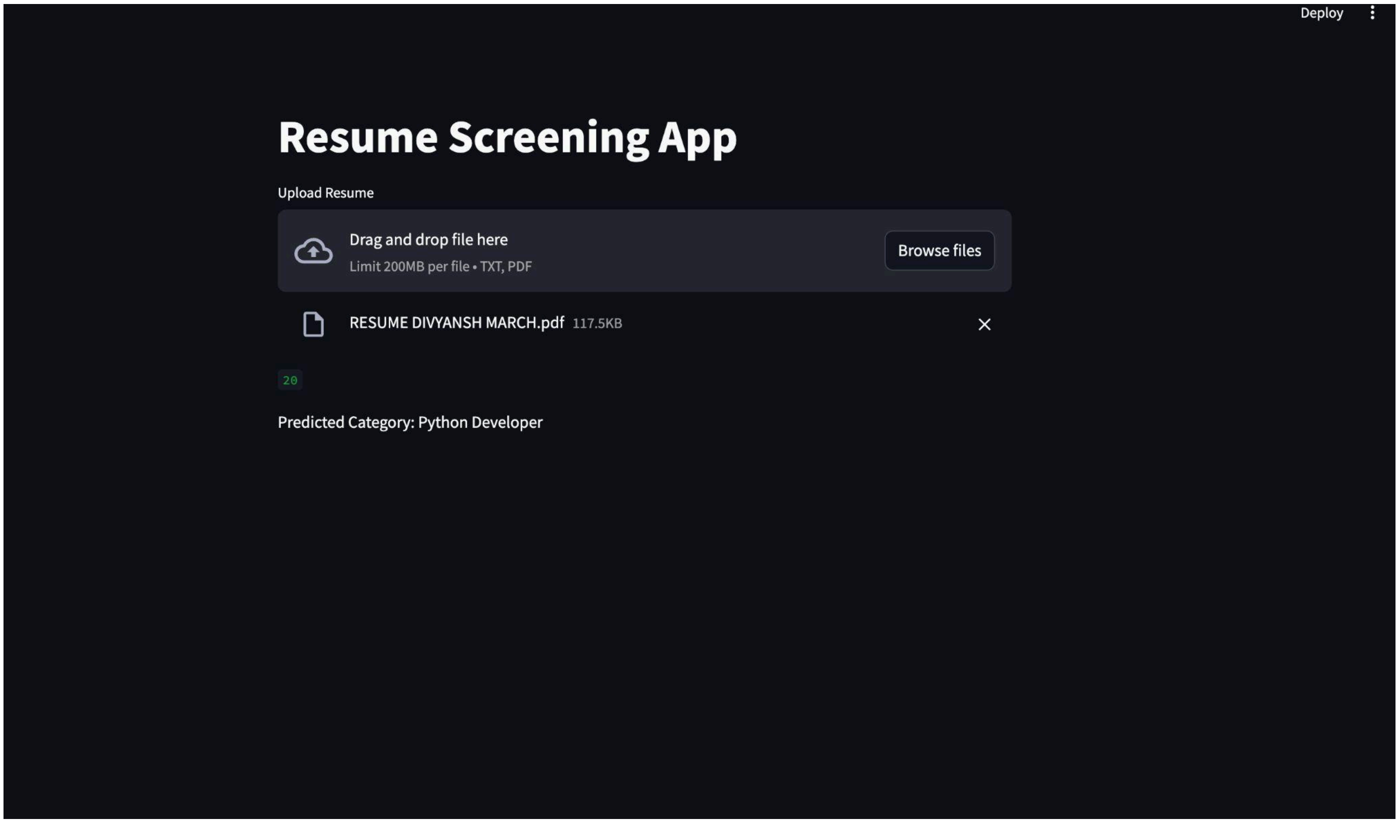
```
        # Map category ID to category name
        category_mapping = {
            15: "Java Developer",
            23: "Testing",
            8: "DevOps Engineer",
            20: "Python Developer",
            24: "Web Designing",
            12: "HR",
            13: "Hadoop",
            3: "Blockchain",
            10: "ETL Developer",
            18: "Operations Manager",
            6: "Data Science",
            22: "Sales",
            16: "Mechanical Engineer",
            1: "Arts",
            7: "Database",
            11: "Electrical Engineering",
            14: "Health and fitness",
            19: "PMO",
            4: "Business Analyst",
            9: "DotNet Developer",
            2: "Automation Testing",
            17: "Network Security Engineer",
            21: "SAP Developer",
            5: "Civil Engineer",
            0: "Advocate",
        }

        category_name = category_mapping.get(prediction_id, "Unknown")

        st.write("Predicted Category:", category_name)
        print(category_name)


# python main
if __name__ == "__main__":
    main()
```

```
import streamlit as st
import pickle
import re
import nltk

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

#loading models
clf = pickle.load(open('clf.pkl','rb'))
tfidfd = pickle.load(open('tfidf.pkl','rb'))

def clean_resume(resume_text):
    clean_text = re.sub('http\S+\s*', ' ', resume_text)
    clean_text = re.sub('RT|cc', ' ', clean_text)
    clean_text = re.sub('#\S+', '', clean_text)
    clean_text = re.sub('@\S+', '  ', clean_text)
    clean_text = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), ' ', clean_text)
    clean_text = re.sub(r'[^\x00-\x7f]', r' ', clean_text)
    clean_text = re.sub('\s+', ' ', clean_text)
    return clean_text
# web app
def main():
    st.title("Resume Screening App")
    uploaded_file = st.file_uploader('Upload Resume', type=['txt','pdf'])

    if uploaded_file is not None:
        try:
            resume_bytes = uploaded_file.read()
            resume_text = resume_bytes.decode('utf-8')
        except UnicodeDecodeError:
            # If UTF-8 decoding fails, try decoding with 'latin-1'
            resume_text = resume_bytes.decode('latin-1')

        cleaned_resume = clean_resume(resume_text)
        input_features = tfidfd.transform([cleaned_resume])
        prediction_id = clf.predict(input_features)[0]
```

Fig [3.7] Python Application

4. EXPERIMENT RESULTS & ANALYSIS

4.1 Result:



```
20
```

Predicted Category: Python Developer

Fig[4.1.1] Application with output

After conducting a comprehensive review of AI-powered resume parsing systems, several key findings have emerged:

Enhanced Efficiency: AI-powered resume parsing systems significantly expedite the candidate screening process compared to manual methods. These systems can analyze a large volume of resumes in a fraction of the time it would take a human recruiter, thereby accelerating hiring cycles and reducing time-to-fill metrics.

Improved Accuracy: The implementation of AI algorithms ensures high accuracy in extracting and categorizing relevant information from resumes. By leveraging natural language processing (NLP) techniques, these systems can accurately identify skills, qualifications, and experience, minimizing the risk of overlooking qualified candidates.

Bias Mitigation: AI-powered resume parsing systems offer the potential to mitigate unconscious bias in the screening process by standardizing candidate evaluation criteria. By focusing solely on the qualifications and experiences outlined in resumes, these systems help ensure fair and unbiased assessments, promoting diversity and inclusion in recruitment practices.

Scalability: AI-powered resume parsing systems demonstrate scalability, allowing organizations to handle large volumes of resumes efficiently, particularly during periods of high recruitment activity or when sourcing candidates for multiple positions simultaneously.

4.2 Result Analysis:

The results of this project work underscore the transformative potential of AI-powered resume parsing systems in revolutionizing recruitment processes. By automating mundane and time-consuming tasks associated with candidate screening, these systems enable HR professionals to allocate their time and resources more strategically, focusing on activities that require human judgment and expertise.

Furthermore, the high accuracy and consistency offered by AI-powered resume parsing systems contribute to improved candidate experiences. Candidates benefit from streamlined application processes and faster response times, leading to enhanced perceptions of the organization's efficiency and professionalism.

Moreover, the potential for bias mitigation represents a significant advancement in promoting fairness and equity in recruitment practices. By standardizing evaluation criteria and minimizing subjective judgments, AI-powered resume parsing systems help organizations build more diverse and inclusive workforces, reflective of varied backgrounds, experiences, and perspectives.

However, it is crucial to acknowledge that AI-powered resume parsing systems are not without limitations. Challenges such as algorithmic bias, data privacy concerns, and the risk of overlooking nuanced candidate attributes still require careful consideration and ongoing refinement.

5. CONCLUSION & FUTURE ENHANCEMENT

5.1 Conclusion:

The exploration of AI-powered resume parsing systems reveals their pivotal role in revolutionizing recruitment processes, offering enhanced efficiency, improved accuracy, bias mitigation, and scalability. By automating mundane tasks and standardizing candidate evaluation criteria, these systems enable organizations to expedite hiring cycles, reduce time-to-fill metrics, and promote diversity and inclusion in recruitment practices. However, while AI-powered resume parsing systems present significant advantages, challenges such as algorithmic bias and data privacy concerns necessitate ongoing refinement and careful consideration.

5.2 Future Enhancements:

Moving forward, several avenues for future enhancement and research emerge:

1. Algorithmic Fairness: Continued efforts are required to mitigate algorithmic bias and ensure fairness in candidate evaluation. Research into advanced algorithms and techniques for detecting and addressing bias can help enhance the fairness and equity of AI-powered resume parsing systems.

2. Personalization: Future enhancements could focus on incorporating personalized features into resume parsing systems, allowing for tailored candidate experiences and better

alignment with organizational culture and job requirements.

3. Integration with Applicant Tracking Systems (ATS): Seamless integration with ATS platforms can further streamline recruitment workflows and enhance the overall candidate experience. Future research could explore ways to optimize integration and interoperability between AI-powered resume parsing systems and existing ATS solutions.

4. Multimodal Parsing: Beyond textual information, future enhancements could explore the integration of multimodal data sources, such as audio and video resumes, to provide a more comprehensive candidate profile and facilitate more informed hiring decisions.

REFERENCES

[1] N. Jayakumar, A. K. Maheshwaran, P. S. Arvind and G. Vijayaragavan,"On-Demand Job-Based Recruitment For Organisations Using ArtificialIntelligence," 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICNWC57852.2023.10127551. keywords: {Visualization; Machine learning algorithms; Resumes; Organizations; Machine learning; Forestry; Market research; Employee attrition; Recruitment; Random Forest}

[2] Z. Chuang, W. Ming, L. C. Guang, X. Bo and L. Zhi-qing,Resume Parser: Semi-structured Chinese Document Analysis," 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 2009, pp. 12-16, doi: 10.1109/CSIE.2009.562. keywords: {Resumes; Text analysis; document analysis; semi-structured; resume parsing; pattern matching}

[3] https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html