Model Development Phase Template

Date	15 Nov 2024
Team ID	739938
Project Title	AI Enabled Candidate Resume Screening
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code for AI Enabled Candidate Resume Screening will be shared through a future screenshot, showcasing the process of text preprocessing, vectorization, and training logistic regression models. The model uses natural language processing and machine learning techniques to analyze resume content and predict candidate suitability. The model validation and evaluation report will summarize the performance of multiple models, including their accuracy, precision, recall, F1 score with the details presented via respective screenshots.

Initial Model Training Code (5 marks):

Paste the screenshot of the model training code

```
from pyresparser import ResumeParser
import smtplib


# SMTP initialization for Outlook
s = smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()
s.login("aravindkumarkorem@gmail.com", "aysg vlyp yzhn nykw")

SUBJECT = "Interview Call"
python_skills = ["ml", "ai", "matplotlib", "seaborn",
                 "python", "regression", "algorithms",
                 "pandas", "data analysis", "keras",
                 "tensorflow", "artificial intelligence",
                 "data visualization", "opencv"]
java_skills = ["java", "object-oriented programming",
               "data structures", "Algorithms",
               "spring framework", "hibernate",
               "SQL", "multithreading", "JavaFX",
               "RESTful API", "Maven", "JUnit", "JSP",
               "Servlets","Javascript","Ai", "OOP"]
data_Scientist = ["Machine Learning (ML)", "Deep Learning (DL)",
                  "Data Mining", "Statistical Analysis"
```

```
data_Scientist = ["Machine Learning (ML)", "Deep Learning (DL)",
                  "Data Mining", "Statistical Analysis",
                  "Data Visualization", "Natural Language Processing (NLP)",
                  "Big Data", "Predictive Modeling", "Feature Engineering",
                  "Regression Analysis", "Classification Algorithms",
                  "Clustering Algorithms", "Time Series Analysis",
                  "Dimensionality Reduction", "Ensemble Methods", "Neural Networks",
                  "Python for Data Science", "R Programming", "SQL", "Tableau",
                  "Apache Spark", "Hadoop"]
```

```
data = ResumeParser("C:/Users/aravi/OneDrive/Desktop/resume_screeninggg/resumeAravindkumar.pdf").get_extracted_da
# Display the extracted data
print(data)
def extract_skills(filename):
    data = ResumeParser(filename).get_extracted_data()
    name = data['name']
    email = data['email']
    skills = data['skills']
    actual_skills = [i.lower() for i in skills]
    return name, email, actual_skills


def extract_education(filename):
    data = ResumeParser(filename).get_extracted_data()
    education = []
    if 'education' in data:
        education = data['education']
    return education
```

```
def extract_certificates(filename):
    data = ResumeParser(filename).get_extracted_data()
    certificates = []
    if 'certificate' in data:
        certificates = data['certificate']
    return certificates


def compare_skills(appliedJob, skills):
    skills_matched = []
    if appliedJob == "AI/ML Dev":
        for ele in skills:
            if ele in python_skills:
                skills_matched.append(ele)
    if appliedJob == "Java Dev":
        for ele in skills:
            if ele in java_skills:
                skills_matched.append(ele)
    if appliedJob == "Data Scientist":
        for ele in skills:
            if ele in data_Scientist:
                skills_matched.append(ele)
    return skills_matched
```

```
def send_email(email, name, is_rejected, appliedJob):
    if is_rejected:
        TEXT = f"Hello {name}, \n\nThanks for applying to the job post {appliedJob} . Your candidature is " \
               f"rejected.\n\n\n\nThanks and Regards,\n\nTalent Acquisition Team,\n\nSmartInternz by Smartbridge"
    else:
        TEXT = f"Hello {name}, \n\nThanks for applying to the job post {appliedJob}. Your skills match our " \
               f"requirements. Kindly let us know the available time for the initial round of " \
               f"interview.\n\n\n\nThanks and Regards,\n\nTalent Acquisition Team,\n\nSmartInternz by Smartbridge
    message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
    s.sendmail("aravindkumarkorem@gmail.com", email, message)
    s.quit()
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Model 1	Spacy is a powerful, open-source library designed for advanced natural language processing (NLP) in Python. It was developed by Matthew Honnibal and Ines Montani, and first released in 2015. Unlike other NLP libraries such as NLTK, which are often used for educational purposes, spaCy focuses on providing tools suitable for production environments, making it ideal for building applications that require efficient text processing and understanding.