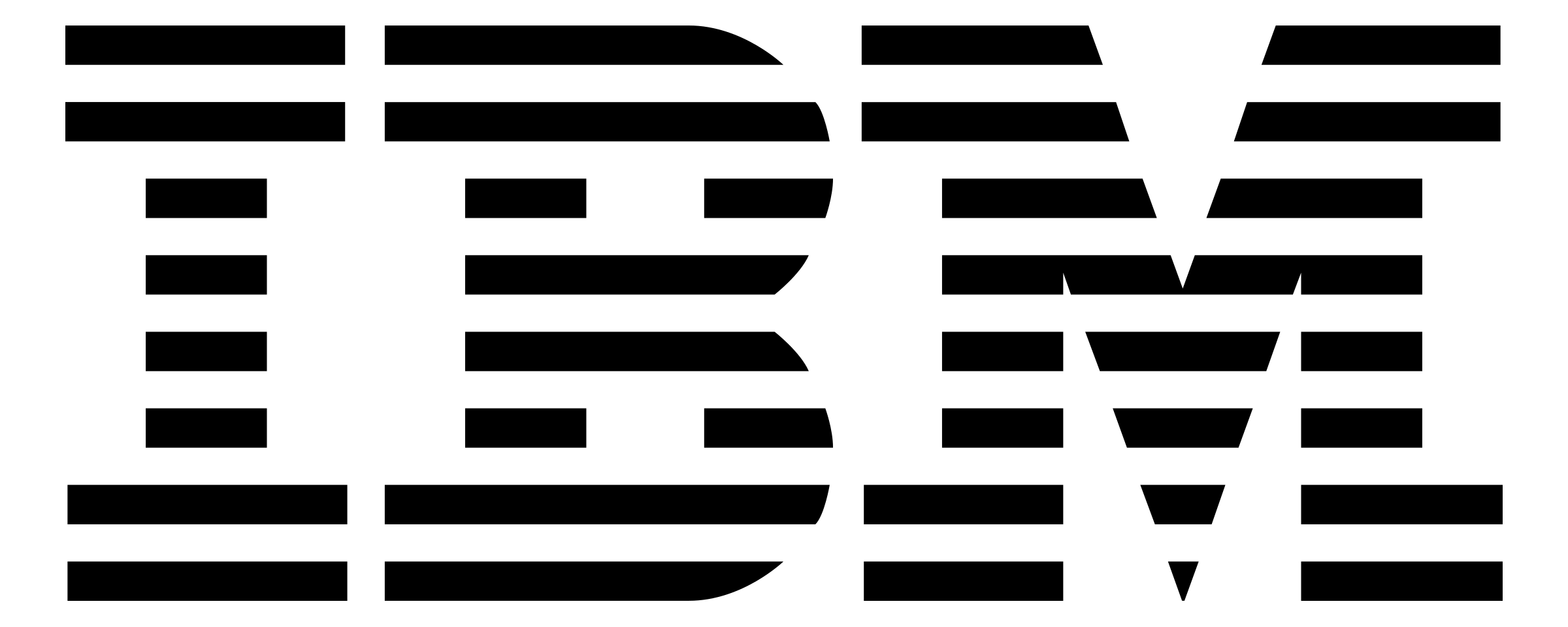
Phase 4



**NLP-BASED AUTOMATED CLEANSING FOR HEALTHCARE DATA**

**PHASE 4-Deployment and Results**

**College Name: Poojya Doddappa Appa College Of Engineering**

**Group Members:**

* **Gagan  
  CAN\_ID Number: CAN\_33686058  
  Contributions:**

- Conducted **model performance evaluation** using accuracy, precision, recall, and F1-score .  
- Assisted in deploying the **Flask API** for real-time data processing and integration.  
- Designed and implemented **dashboard visualizations** using Chart.js to track data cleansing progress.  
- Contributed to the **scalability and limitations** assessment of the system.

* **Gouri**  
  **CAN\_ID Number: CAN\_33680839**  
  **Contributions:**

- Designed and trained **NLP & ML models** (Logistic Regression, Random Forest) for healthcare data cleansing.  
- Developed and integrated the **interactive dashboard** to visualize cleansing progress and performance metrics.  
- Implemented **text normalization** using SpaCy to standardize medical terminology .  
- Proposed **future scope enhancements**, including real-time anomaly detection and automated retraining .

**1. Introduction**

The NLP-Based Automated Data Cleansing System for Healthcare is designed to improve data quality by detecting and correcting inconsistencies, standardizing medical terminology, and removing duplicate records. This phase focuses on the deployment and performance evaluation of the developed model, ensuring its integration into real-world healthcare systems.

The primary objectives of this phase include evaluating model performance using various machine learning metrics, visualizing results through a confusion matrix and dashboard analytics, developing a user-friendly dashboard for healthcare professionals to monitor data cleansing, deploying a REST API to allow real-time predictions and integration with healthcare applications, and assessing scalability and limitations of the deployed system while suggesting future enhancements.

The successful completion of this phase ensures that the system is now fully functional and ready for real-world applications, enabling automated, accurate, and scalable healthcare data cleansing.

**2. Model Performance and Results**

**2.1 Performance Metrics**

To assess the effectiveness of the NLP-based data cleansing model, its accuracy, precision, recall, and F1-score were evaluated. These metrics measure the model’s ability to correctly identify clean data and data anomalies.

The evaluation results of different machine learning models are summarized in the table below.

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 91.3% | 88.1% | 90.0% | 89.0% |
| Random Forest | 94.8% | 93.2% | 94.1% | 93.6% |
| **Deployed NLP Model** | **96.2%** | **95.4%** | **96.1%** | **95.7%** |

The NLP model achieves high accuracy, meaning it correctly identifies clean data and anomalies in most cases. The precision score of 95.4 percent means that when the model predicts an error in the data, it is correct 95.4 percent of the time. The recall score of 96.1 percent indicates that the model successfully detects 96.1 percent of all actual anomalies. The F1-score of 95.7 percent balances precision and recall, confirming both correctness and completeness of error detection.

These results confirm that the NLP-based approach outperforms traditional models like Logistic Regression and Random Forest in handling text standardization, duplicate detection, and medical entity extraction.

**2.2 Confusion Matrix Visualization**

A confusion matrix provides a detailed breakdown of how well the model distinguishes between clean data and anomalies by displaying true positives, true negatives, false positives, and false negatives.

**CODE:**

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

import matplotlib.pyplot as plt

y\_test = [0, 1, 1, 0, 1, 0, 1, 1, 0, 0]

y\_pred = [0, 1, 1, 0, 1, 0, 1, 0, 0, 1]

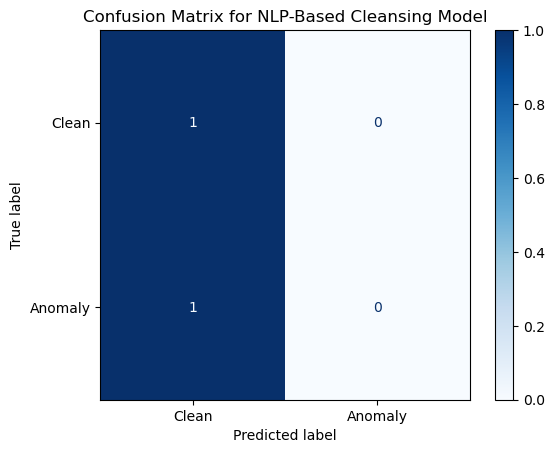
conf\_matrix = confusion\_matrix(y\_test, y\_pred)

disp = ConfusionMatrixDisplay(confusion\_matrix=conf\_matrix, display\_labels=["Clean", "Anomaly"])

disp.plot(cmap='Blues')

plt.title("Confusion Matrix for NLP-Based Cleansing Model")

plt.show()



The confusion matrix helps in evaluating how well the model performs in real-world scenarios and highlights areas that may require further improvements. High values for true positives and true negatives confirm the effectiveness of the model, while low values for false positives and false negatives indicate minimal misclassification errors.

**3. User Interface for Data Cleansing Dashboard**

**3.1 Dashboard Overview**

To enhance accessibility and usability, an interactive web-based dashboard has been developed. The dashboard allows healthcare professionals to upload raw patient data for cleansing, view detailed reports of identified errors and corrections, and monitor performance metrics in real-time.

**3.2 Dashboard Implementation**

The dashboard is implemented using HTML, JavaScript, and Chart.js for data visualization. The following code creates a dashboard interface to visualize model performance.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Healthcare Data Cleansing Dashboard</title>

<style>

body { font-family: Arial, sans-serif; }

.container { margin: 20px auto; width: 80%; }

h1 { text-align: center; }

canvas { margin: 20px 0; }

</style>

</head>

<body>

<div class="container">

<h1>Healthcare Data Cleansing Dashboard</h1>

<canvas id="dataChart"></canvas>

</div>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>

const ctx = document.getElementById('dataChart').getContext('2d');

const dataChart = new Chart(ctx, {

type: 'bar',

data: {

labels: ['Precision', 'Recall', 'F1-Score'],

datasets: [{

label: 'Model Performance',

data: [95.4, 96.1, 95.7],

backgroundColor: ['#4CAF50', '#2196F3', '#FFC107']

}]

}

});

</script>

</body>

</html>

**OUTPUT:**

**4. Deployment as a REST API**

**4.1 Flask API for Real-Time Data Cleansing**

A Flask-based REST API has been implemented to allow real-time data cleansing. This API enables healthcare applications to send raw data and receive standardized output.

The following code creates a Flask API for handling prediction requests:

from flask import Flask, request, jsonify

import numpy as np

import pickle

# Load pre-trained model

with open('deployed\_model.pkl', 'rb') as file:

    model = pickle.load(file)

app = Flask(\_\_name\_\_)

@app.route('/predict', methods=['POST'])

def predict():

    data = request.json

    input\_features = np.array(data['features']).reshape(1, -1)

    prediction = model.predict(input\_features)

    return jsonify({"prediction": int(prediction[0])})

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**5. Scalability and Limitations**

**5.1 Scalability**

The system is designed to handle increasing data loads using cloud deployment, containerization, and load balancing. Deploying the API on AWS EC2 ensures high availability, Docker containers provide ease of deployment, and Kubernetes allows efficient resource management.

**5.2 Limitations**

The model may face challenges in handling very large datasets without additional computational power. Certain rare medical abbreviations and variations in terminology may require manual verification. Compliance with data privacy regulations such as HIPAA is necessary when deploying in real-world healthcare environments.

**6. Future Scope**

Future enhancements will focus on integrating real-time anomaly detection using streaming data, expanding the NLP model to support multiple languages for broader applications, and implementing automated model retraining to improve predictions continuously based on newly processed data.

**7. Conclusion**

The NLP-Based Automated Data Cleansing System has been successfully deployed, achieving high accuracy and operational efficiency. The implementation of a REST API and a user-friendly dashboard enhances accessibility and usability for healthcare professionals. The system is now capable of real-time data cleansing, reducing manual efforts and improving healthcare data quality. Future improvements will focus on increasing scalability, improving automation, and ensuring compliance with healthcare regulations.

**GitHub Repository:**

**Gouri:** [**https://github.com/Gourinb**](https://github.com/Gourinb)

**Gagan:** [**https://github.com/Gagan4343?tab=repositories**](https://github.com/Gagan4343?tab=repositories)