# KUBERNETES DATA ANALYSIS

A Q

# TABLE OF CONTENTS

# INTRODUCTION

Kubernetes data analysis

# PROBLEM STATEMENT

In a Kubernetes environment, failures can manifest in multiple ways

1.

## Node or pod failures

Node or pod failures in Kubernetes can disrupt workloads, causing downtime and requiring quick recovery to maintain cluster stability

2.

## Resource exhaustion

CPU, memory, or disk depletion, can degrade performance, cause pod evictions, and lead to application failures if not managed properly

3.

## Network or connectivity issues

Disrupt communication between pods, services, and external systems, leading to latency, timeouts, or complete service failures

4.

## Service disruptions based on logs and events

Provide insights into failures, crashes, or resource constraints affecting application performance and availability.

## DATA SET

To build an effective machine learning model, structured log data is required. Kubernetes logs, however, are often unstructured and difficult to analyze directly

### GROK

A pattern-matching tool
Used to extract meaningful features from raw log files.
Allows us to filter and structure log entries by identifying timestamps, error codes, resource usage metrics, and pod statuses

- The extracted data is preprocessed to remove noise and irrelevant information.
- Only significant attributes are retained for model training.
- The prepared dataset serves as the foundation for training the Random Forest Classifier.
- The model learns to recognize failure patterns within Kubernetes clusters.

## RANDOM FOREST

### Dataset Preparation

- The process begins with loading and preprocessing the dataset.
- The dataset is split into training and testing sets for model evaluation.

### Model Training

- The Random Forest Classifier is trained on extracted log features.
- It learns patterns associated with past failures to make predictions.

### Model Evaluation

After training, the model is evaluated based on key performance metrics, including:

- Accuracy
- Precision
- Recall

### Feature Importance & Predictive Accuracy

- Random Forest provides insights into feature importance, helping system administrators identify key failure factors.
- By leveraging an ensemble learning approach, the classifier enhances predictive accuracy.
- This makes it a reliable tool for proactive Kubernetes failure detection.

## MAKING MODEL

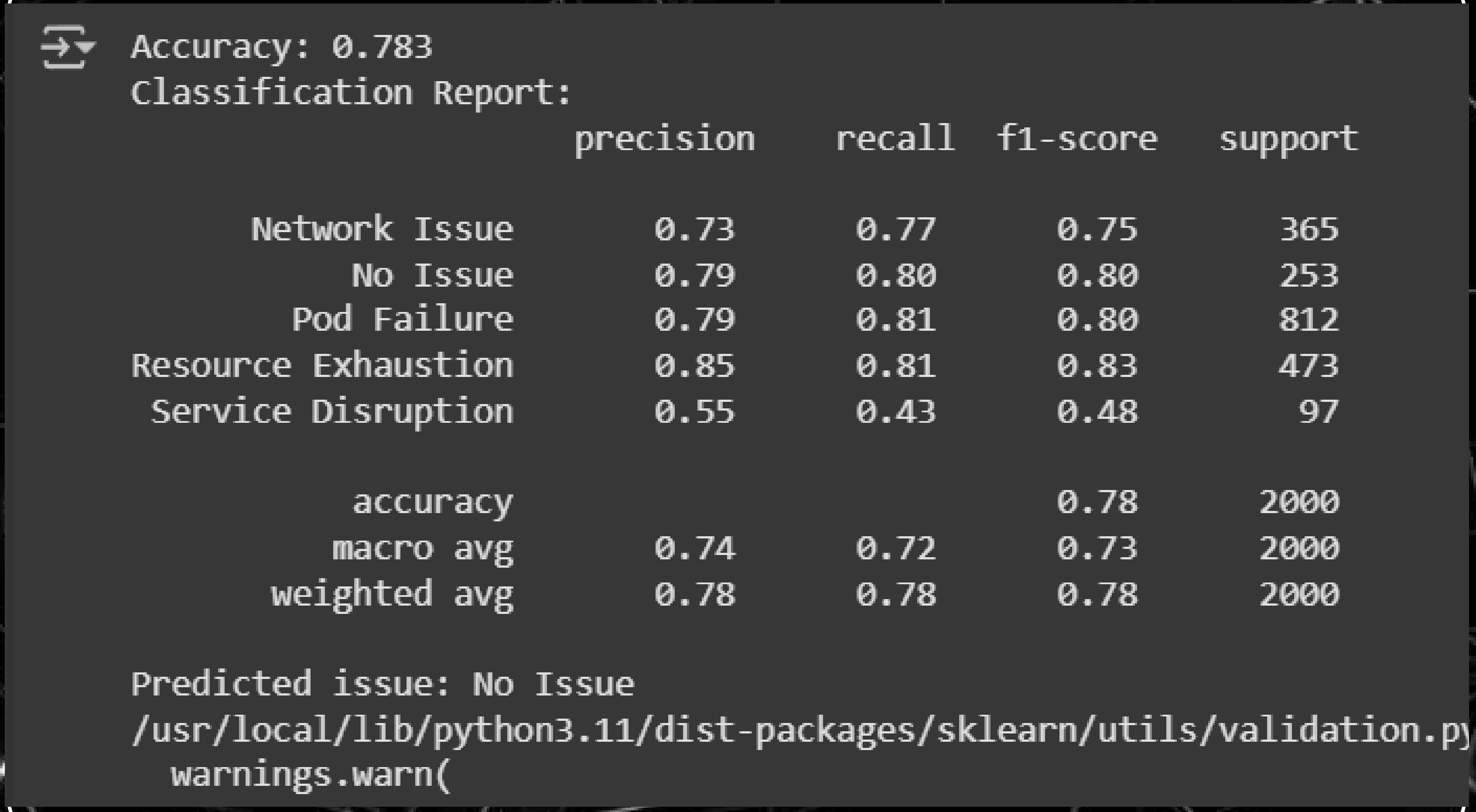
The Random Forest Classifier is chosen for this project due to its ability to handle high-dimensional data and its robustness against overfitting

CONCLUSION

## RESULTS

### 80% ACCURACY

By using the RandomForestClassifier and training it using the generated dataset we receive a model of upto 80% accuracy only with one iteration.



```
Accuracy: 0.783
Classification Report:
                     precision    recall  f1-score   support

      Network Issue       0.73      0.77      0.75       365
           No Issue       0.79      0.80      0.80       253
        Pod Failure       0.79      0.81      0.80       812
Resource Exhaustion       0.85      0.81      0.83       473
 Service Disruption       0.55      0.43      0.48        97

           accuracy                           0.78      2000
          macro avg       0.74      0.72      0.73      2000
       weighted avg       0.78      0.78      0.78      2000

Predicted issue: No Issue
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py
  warnings.warn(
```

# OUR TEAM

Shashi Vadhan M

Mohd Mustaneer Nabeel

Ridha Mohammadi

Aakifa Aimen