**Predicting Kubernetes Cluster Failures Using AI/ML**

**Abstract**

Kubernetes clusters encounter failures such as pod crashes, resource bottlenecks, and network issues. This project builds an AI/ML model that predicts these failures before they occur using historical and real-time cluster metrics. The model enhances Kubernetes reliability by using anomaly detection and time-series analysis.

**Objective & Scope**

* Predict Kubernetes failures: pod crashes, resource exhaustion, network disruptions.
* Utilize real-time & historical metrics (CPU, memory, network usage, logs).
* Develop a robust ML model (supervised learning, anomaly detection).
* Deploy model in Kubernetes for real-time predictions.

**Dataset & Data Cleaning**

* Metrics used: CPU, memory, network IO, pod/node status logs, Kubernetes events.
* Missing values handled via imputation.
* Noisy and redundant data removed.
* Standardized time-series metrics for consistency.

**Data Splitting & Preprocessing**

* 70% training, 30% testing split.
* Feature scaling via MinMax normalization.
* Encoding categorical variables (e.g., pod states).
* Time-series engineering: lag features, rolling averages.

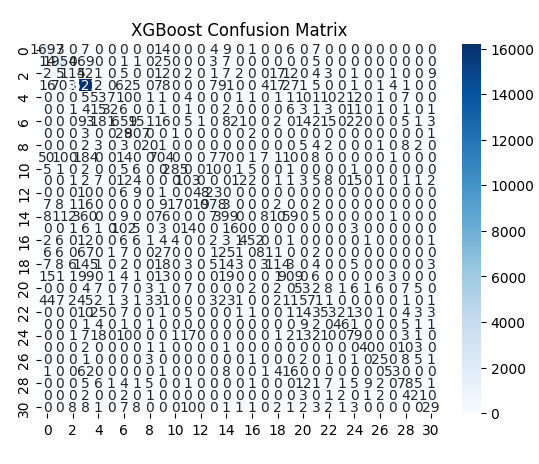
**Model Building**

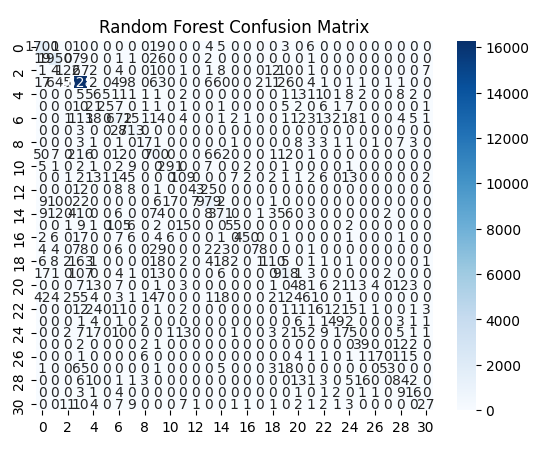
* Models tested:
  + Random Forest Classifier
  + Decision tree
  + XGBoost Classifier
* Hyperparameter tuning with Grid Search.
* Model performance evaluated using accuracy, precision, recall, F1-score.

|  |  |
| --- | --- |
| model | accuracy |
| decision tree | 83.1 |
| Naïve bayes | 9.5 |
| Random forest | 87.3 |
| Logistic regression | 66.3 |
| SVM | 57.2 |
| KNN | 71.2 |
| XGBoost | 87.4 |
| MLP | 67.07 |

**Final Model & Performance**

* Random Forest and XGBoost attained 87.5% accuracy.
* Performance metrics:
  + Accuracy: 87.5%
* High confusion matrix score indicating effective failure prediction.





**Conclusion**

* AI/ML model successfully predicts Kubernetes failures.
* Enables proactive maintenance and reduces downtime.
* Future work:
  + Integrate real-time anomaly detection.
  + Deploy model as a microservice in Kubernetes.
  + Improve model robustness with more diverse datasets.