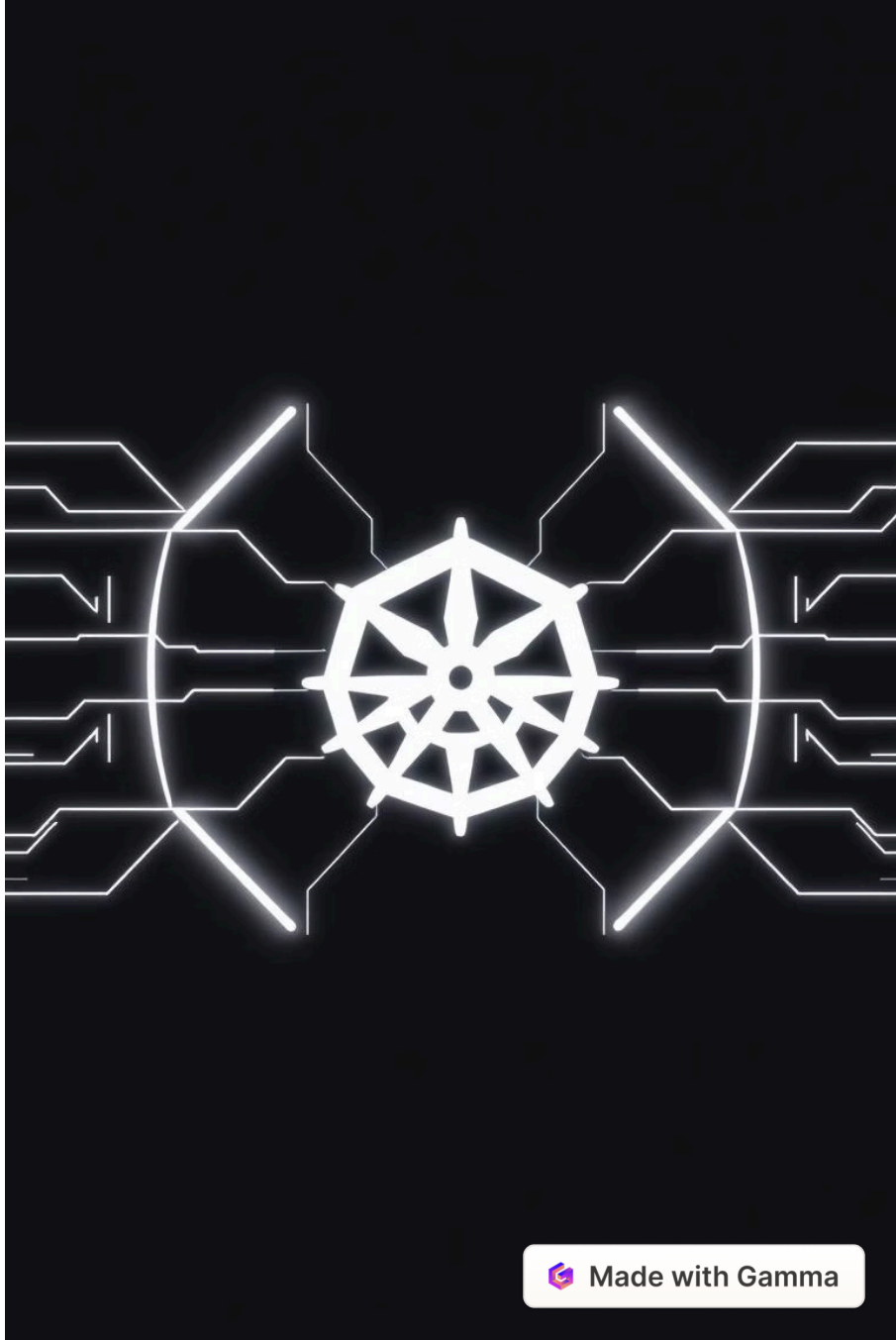
XGBoost for Predicting Kubernetes Issues

Team Envision Trails presents XGBoost for predicting Kubernetes issues. Kubernetes complexity leads to unpredictable failures and downtime. We use XGBoost to forecast potential issues and improve system stability.



Problem Statement: Kubernetes Challenges

Increased Downtime

Application downtime impacts revenue and user experience. Average downtime costs $5,600 per minute.

Reactive Firefighting

Difficult to predict and prevent failures proactively, leading to reactive firefighting.

Over-provisioning

Over-provisioning resources to avoid failures increases infrastructure costs by up to 30%.

Solution Overview: XGBoost Prediction

Data Collection

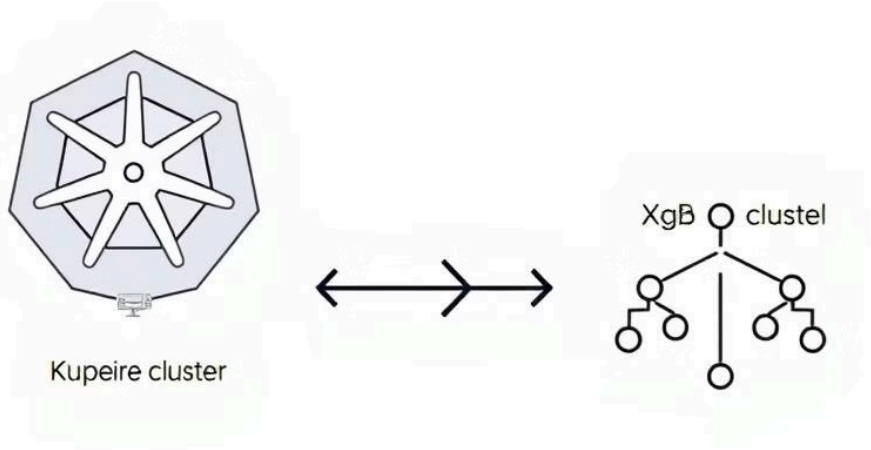
Collect real-time metrics and logs from Kubernetes clusters using Prometheus and Fluentd.

Feature Engineering

Preprocess data and engineer relevant features like CPU utilization and memory usage.

Model Training

Train an XGBoost model to predict potential issues like pod failures.



Data Handling: Feature Engineering

Data Sources

- Kubernetes API
- Prometheus (metrics)
- Fluentd (logs)

Feature Engineering

- Rolling average of CPU/Memory usage
- Rate of change of key metrics
- Categorical encoding of pod status

Preprocessing

- Handling missing values
- Scaling numerical features
- Data Cleaning

Model Training: XGBoost

1

XGBoost Algorithm

Gradient boosting framework for classification/regression.

2

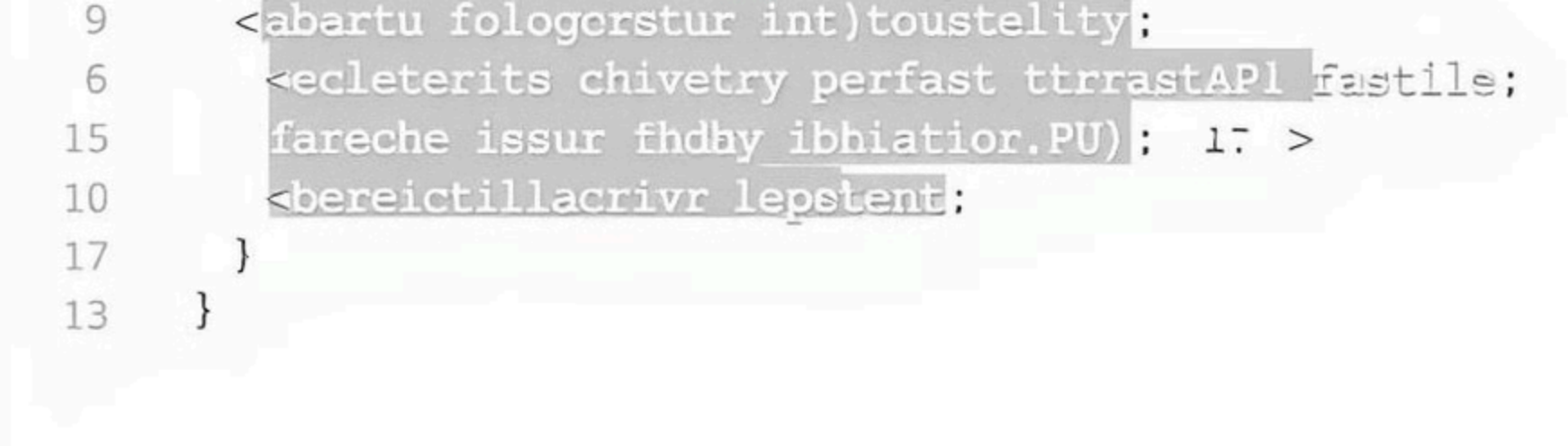
Hyperparameter Tuning

GridSearchCV for optimal parameters.

3

Evaluation Metrics

Precision, Recall, F1-score.



FastAPI Inference: Real-time Prediction

FastAPI

Lightweight Python framework for building APIs.

Endpoint

/predict for receiving JSON data containing metrics.

Output

Return a JSON response with the predicted probability and alert status.