Caterpillar Hackathon

1. **Team Name:** Code Crashers

2. **Team Member Names:**

   - ❖ Supreetha H R
   - ❖ Shreya S
   - ❖ Kausthubha Gajula

3. **Problem Statement chosen:**

Predicting component failures based on history and usage data.

One of our core initiatives is to use telematics data collected from customers' equipment to proactively detect potential present or future equipment/component failures. Students will have the opportunity to develop a smart AI solution that reproduces and possibly surpasses a human expert's thought processes in predicting equipment/component failures using telematics data (specifically Product Link). The student team will have the freedom to identify recommendations they feel will make the greatest impact to equipment efficiencies for Caterpillar's global customers.

4. **Describe the Solution:**

- ***Customizable Failure Prediction Models***

We could develop tailored predictive models for each type of machine and its components, making our predictions more accurate. Plus, we'd implement adaptive thresholds that adjust based on past data and current operating conditions, ensuring our models stay relevant and effective.

- ***Automated Maintenance Scheduling***

With an automated scheduling tool, you'd get maintenance plans generated based on our predictions and your operational needs. This system would also recommend the best way to allocate resources, such as parts and personnel, to keep your maintenance operations efficient and well-organized.

- ***Repair History Recommendations***

Leveraging past repair data, our platform would provide tailored recommendations for upcoming repairs. By analysing historical repair trends and recurring issues, the system can suggest proactive measures and repairs before problems escalate, helping to prevent unexpected breakdowns and improve overall equipment reliability.

5. Tools and Technology used:

Data Preprocessing: Leverage tools such as Pandas and NumPy for cleaning, transforming, and preparing data. Proper preprocessing ensures that your data is ready for effective model training.

ML Tools and Technologies: Use Scikit-learn for implementing and evaluating traditional machine learning models, Logistic regression, XGBoost for high-performance gradient boosting, and LightGBM for scalable, efficient learning

Model Implementation and Deployment: Train and fine-tune models, evaluate their performance, and deploy them using platforms like TensorFlow Serving or Flask APIs. Continuously monitor and update models to maintain accuracy and relevance.

6. How are you approaching?

Data Collection and Preparation:

We extended the dataset by generating additional rows to ensure a robust analysis.

Model Development:

For model development, we use machine learning techniques to predict potential component failures. Initially, we apply logistic regression to handle smaller datasets. We train our models on historical data to detect patterns associated with component failures and assess their performance using metrics such as accuracy and F1-score. This approach ensures that our models are well-calibrated and effective in predicting failures.

System Implementation:

The system architecture includes a data ingestion pipeline, predictive modelling layer, and a user-friendly interface for monitoring and scheduling maintenance. We utilize Python for developing our models and custom scripts for data processing and integration. The user interface is built using a web framework, providing real-time insights and notifications.

7. Key features and Unique Selling Ponit of your Product:

- Customized Failure Prediction Models
- Smart Maintenance Scheduling

8. Milestones achieved as of 10th August 11:00 am:

Data Collection and Preparation:

Collected comprehensive data from various machines, capturing critical parameters.

Extended dataset columns to include: Id, Date, Machine Type, Component, Engine Temp, Fuel Temp, Exhaust Gas Temp, Engine Oil Pressure, Transmission Pressure, Brake Pressure, Engine Speed (RPM), Fuel Level, Oil Level, Pedal Sensor Reading, Hydraulic Pump Rate, Operational Hours, Load Conditions, Failure History, and Repair History.

Pre-processed the data, including handling missing values, correcting inconsistencies, and formatting for machine learning.

Model Development:

Developed initial logistic regression models for predicting component failures.

Implemented random forest algorithms to enhance predictive accuracy for larger datasets.

Trained and evaluated models on historical data to identify patterns associated with failures.

9. Outstanding milestones for the next 24 hours:

Repair History Recommendations:

We'll start by analyzing historical repair data to identify common patterns and issues. This helps us understand recurring problems and trends.

Using these insights, we'll create a recommendation system that suggests proactive repairs and maintenance actions. This feature will help users address potential issues before they escalate.

Finally, we'll integrate these recommendations into the platform, ensuring they are easily accessible and actionable for users.

Predictive Maintenance Platform:

Our first step will be to outline the platform's structure, focusing on how we'll gather and process equipment data, and how this data will be presented to users.

We'll begin developing the platform, setting up data flows, implementing algorithms, and designing a user-friendly dashboard to display equipment health and predictions.

Customizable Failure Prediction Models:

We'll prepare the data by cleaning and normalizing it, making it suitable for building predictive models.

We'll experiment with different machine learning models to find the best one for our data. We'll adjust and fine-tune these models to enhance their accuracy.

Once the best models are selected, we'll integrate them into the platform and conduct tests to confirm that they provide accurate and useful predictions.

Automated Maintenance Scheduling:

We'll determine the requirements for the automated scheduling tool, including how it will generate maintenance schedules and allocate resources effectively.

We'll develop this scheduling tool, incorporating it with our predictive models to create efficient maintenance plans and resource recommendations.

After building the tool, we'll test its functionality to ensure it operates smoothly and make improvements based on feedback to optimize its performance.