ESILV
Paris, France

Track UAV- Team 13
IBM Hackathon

Branly Djime
Akram Hanouz
Marwan Benis
Jules Jouvin
Zakaria Jalal
Lucas Perrier

November 7th 2025

Outline

- Context
- Business Use Case
- Solutions
- Added Value
- Next Steps

Context

What is DavinciHive?

DaVinci Hive is a **student association** dedicated to the world of drones and embedded technologies

What challenges does DavinciHive face?

Drone failures can lead to **loss of equipment**, service interruptions, and risks to people.

How can DavinciHive face their challenges?

Davinci Hive can build **preventive maintenance** and **early fault detection** solutions by using flight data to detect the presence of a fault with machine learning algorithms.

Who are we?

We are **ESILV students** participating in the IBM hackathon held for all Data and AI students.

What is the IBM hackathon?

A 3 day hackathon held at ESILV with 100 participants giving students access to the IBM cloud platform with 4 challenges to choose from.

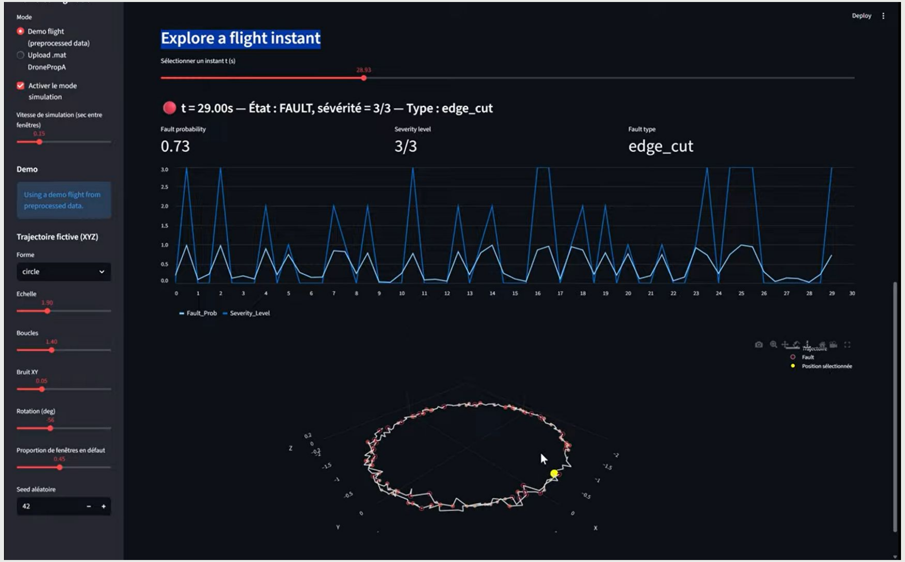
DAVINCI HIVE

Business Use Case

DAVINCI HIVE

Implementing a preventive maintenance approach using machine learning:

- We implement a machine learning model using flight data to predict fault type.
- By treating the data as a classification problem, we use our machine learning model as a pre-flight check for faults.
- By training more accurate machine learning models, we can have more confidence in our preventive machine learning based measures.



Solutions

DAVINCI HIVE

Machine learning models:

- KNN
- Xgboost
- CNN 1D
- IBM cloud Auto.ai

Methodology:

- Train/test split
- Comparative evaluation metrics F1, Accuracy, MAE between models
- Confusion Matrix as sanity check

Data:

- Fast Fourier Transform as a data processing tool
- All 4 GB of available flight data used for training

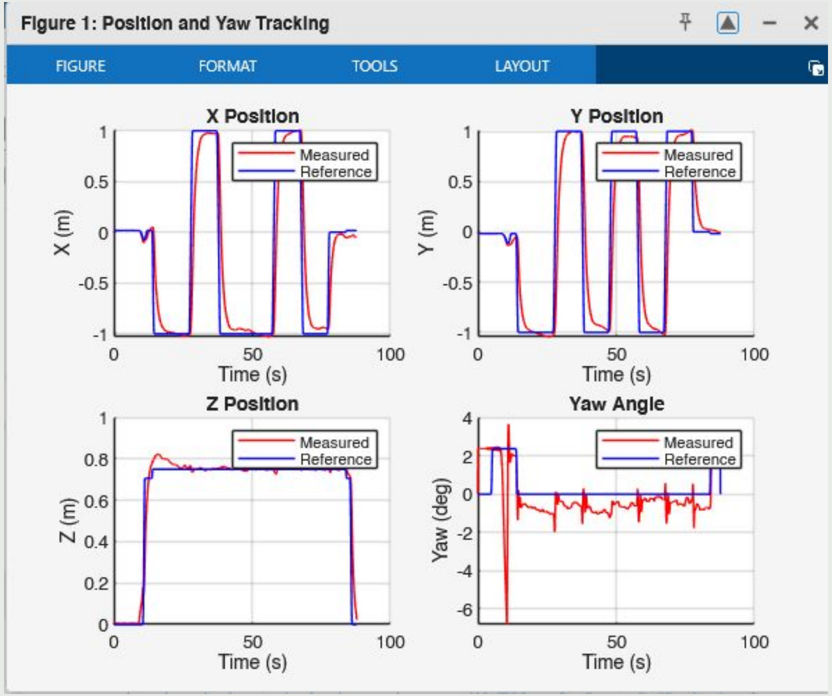
Flight Data Visualization:

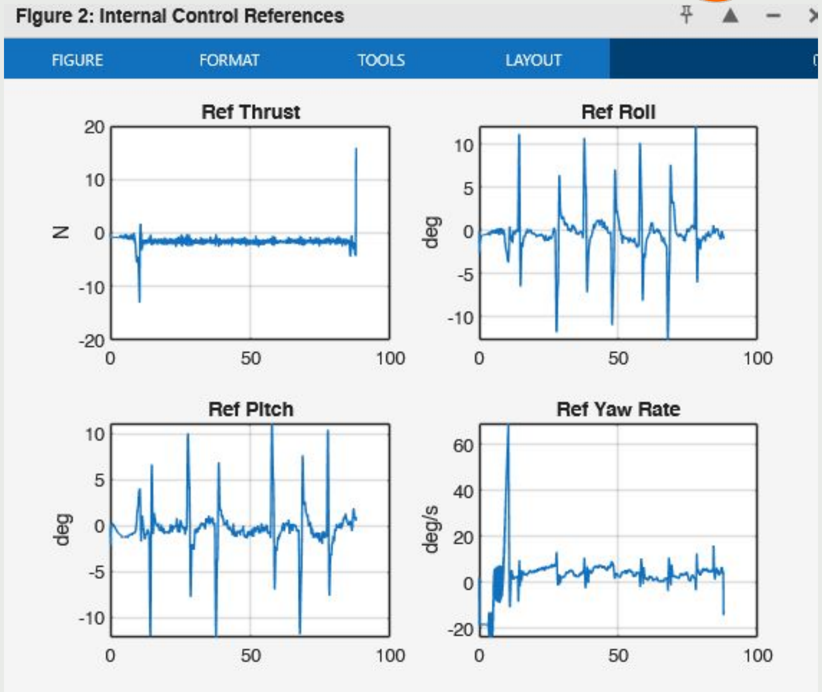
- Power BI based visualization dashboard for known flight data with KPI (EDA)

Prediction Visualization:

- Stream lit app interface for flight data analysis and real time prediction from XGboost model

Exploration on MatLab





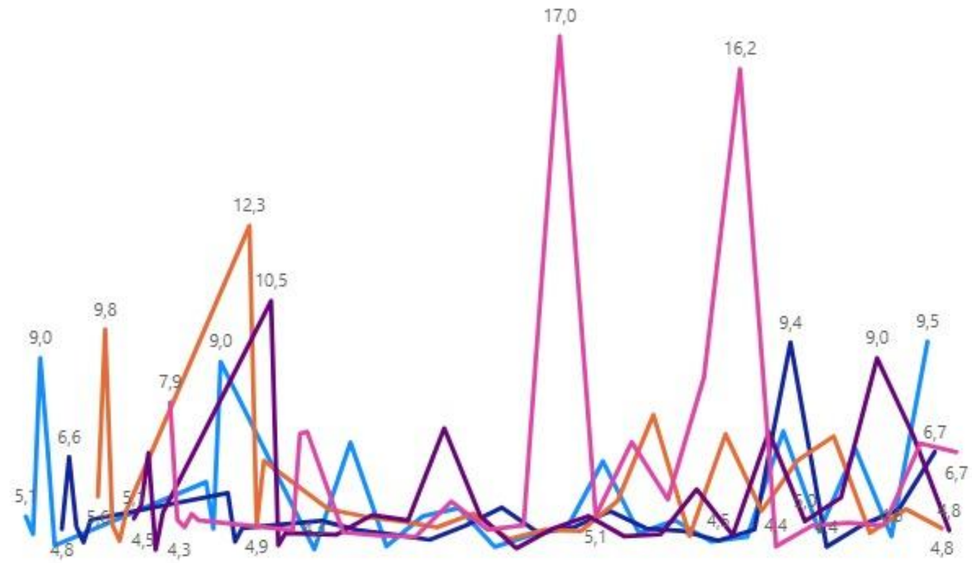
Overview

130
flights

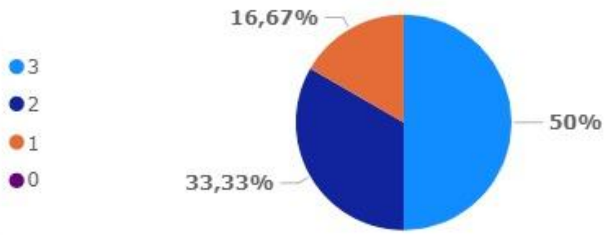
69,23 %
faulty

1,38
average severity

Mean acceleration per trajectory



Fault type distribution



Top 10 worst flights

Drone	FaultType	Severity	MotorVariance
1	3	1	224,57
1	3	2	250,36
1	3	1	253,93
1	2	2	260,33
1	2	3	263,47
1	2	2	273,24
1	0	0	274,71
1	3	3	280,95
1	0	0	281,10
1	1	2	282,68

Dataset Generation

DAVINCI HIVE

Source: 130 indoor experimental flights on a commercial drone with injected faults.

Injected faults:

- 3 types → *crack*, *edge cut*, *surface cut*
- 3 severity levels + 1 healthy condition

Flight conditions:

- 5 trajectories (cross, square, step climb, direct climb, yaw ±45°/±90°)
- 2 speed scenarios: SP1 (2.0 m/s), SP2 (0.33 m/s)

Raw files (.mat): 3 matrix: QDrone_data, commander_data, stabilizer_data

- Filename encodes fault type, severity, speed, trajectory

Preprocessing (main_prepare_data.py):

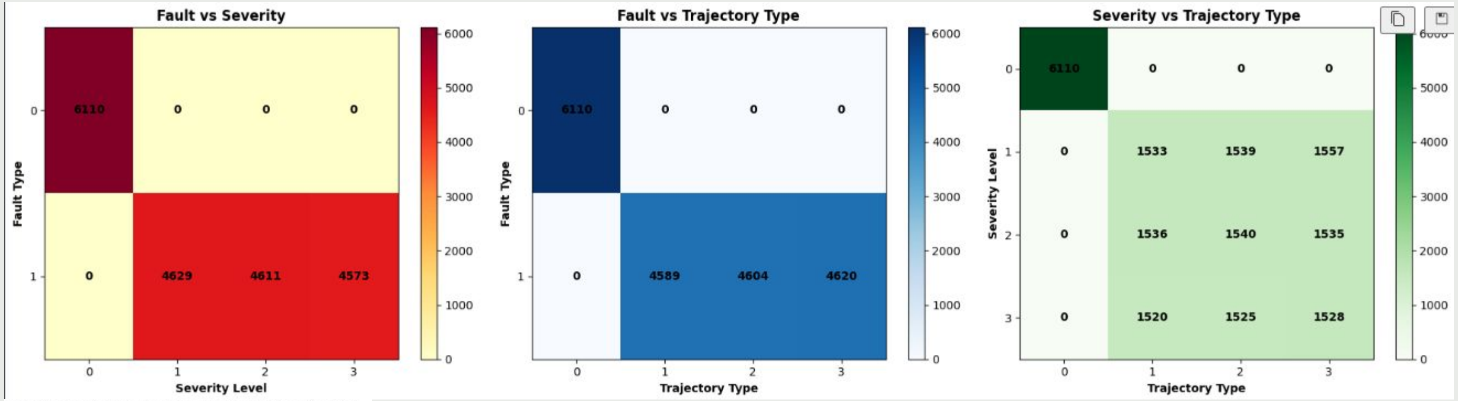
1. Load all flights & **resample to 100 Hz** (linear interpolation)
2. **Global normalization** (mean/std)
3. **Sliding window segmentation:** 1 s windows (100 samples) with 50% overlap
4. **Label generation** (fault, type, severity, trajectory, speed, drone_id)

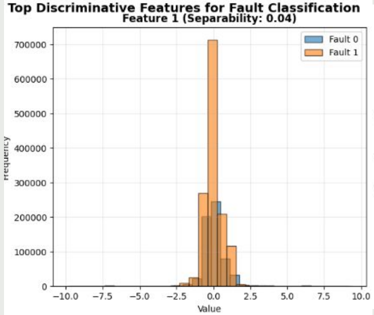
Final dataset:

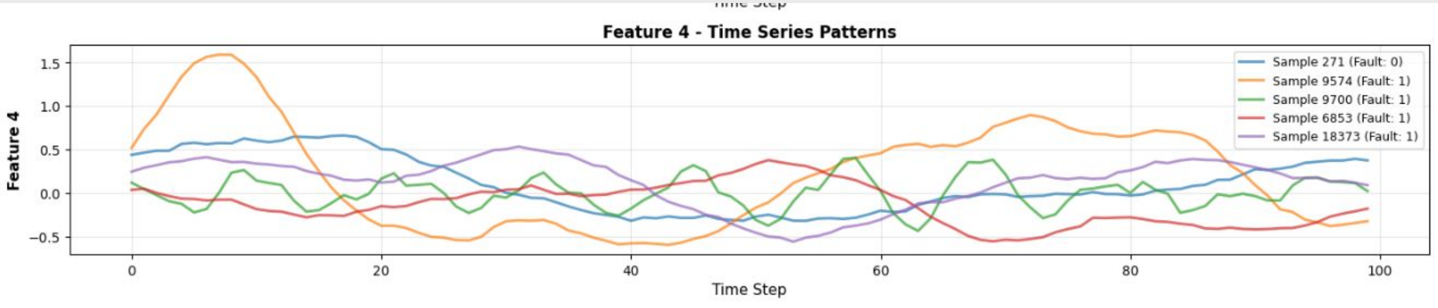
- X_windows: (19 923, 100, 111) → ~20k normalized windows
- y_fault, y_type, y_sev: corresponding label vectors
- Exported as .npy, .npz, .json

Usage: uploaded to **watsonx.ai** for model training (XGBoost, CNN MTL, KNN CUDA)

Exploration extracted features







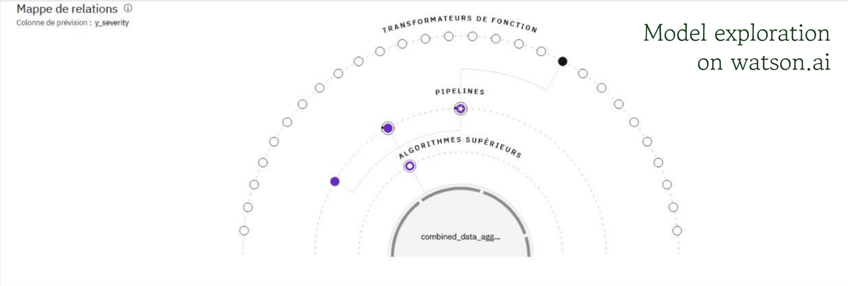
Difficulties Faced

Data Pre-processing:
The large feature space means the data dimensionality is difficult to process and train on. To capture essential information we use a Fast Fourier Transform to pre process our data into (N,T,frequency) format. The fast fourier transform efficiently turns a multi-dimensional signal into the frequency domain.

Model Training Time:
Cpu long training time (more than 2 hours), random forest (scikit learn), so we used xgboost (XGboost library) in order to use a gpu.

Davinci Hive can build preventive maintenance and early fault detection solutions by using flight data to detect the presence of a fault with machine learning algorithms

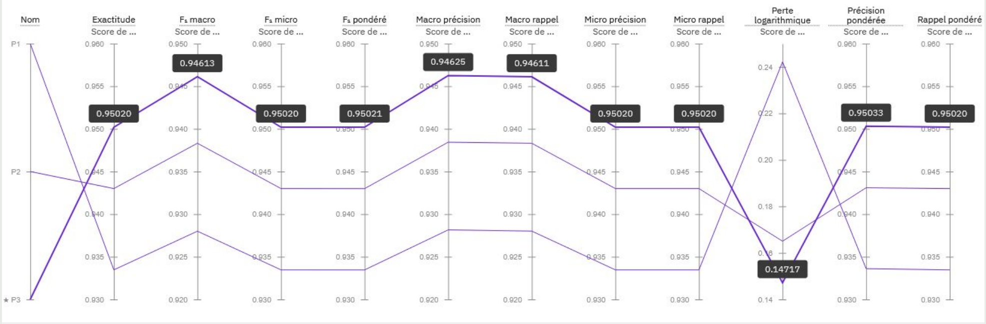
Watsonx.ai - auto.ai



Model exploration on watson.ai

Classement de pipeline

	Rang	Nom	Algorithme	Spécialisation	Exactitude (Optimisé) Validation Croisée	Améliorations	Heure de création
★	1	Pipeline 3	Discriminant LGBM		0.950	HPO-1 FE	00:30:57
	2	Pipeline 2	Discriminant LGBM		0.943	HPO-1	00:09:40
	3	Pipeline 1	Discriminant LGBM		0.933	Aucun	00:01:22

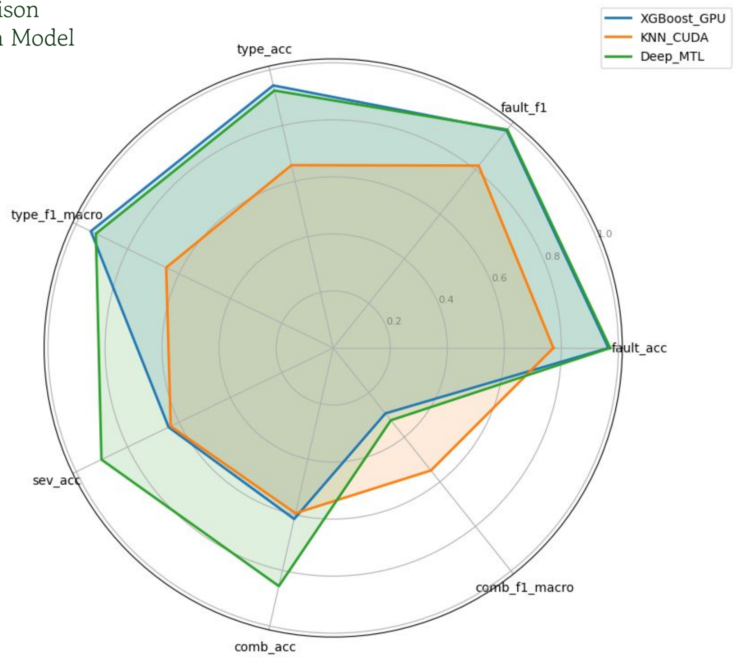


Metrics for the three different pipelines

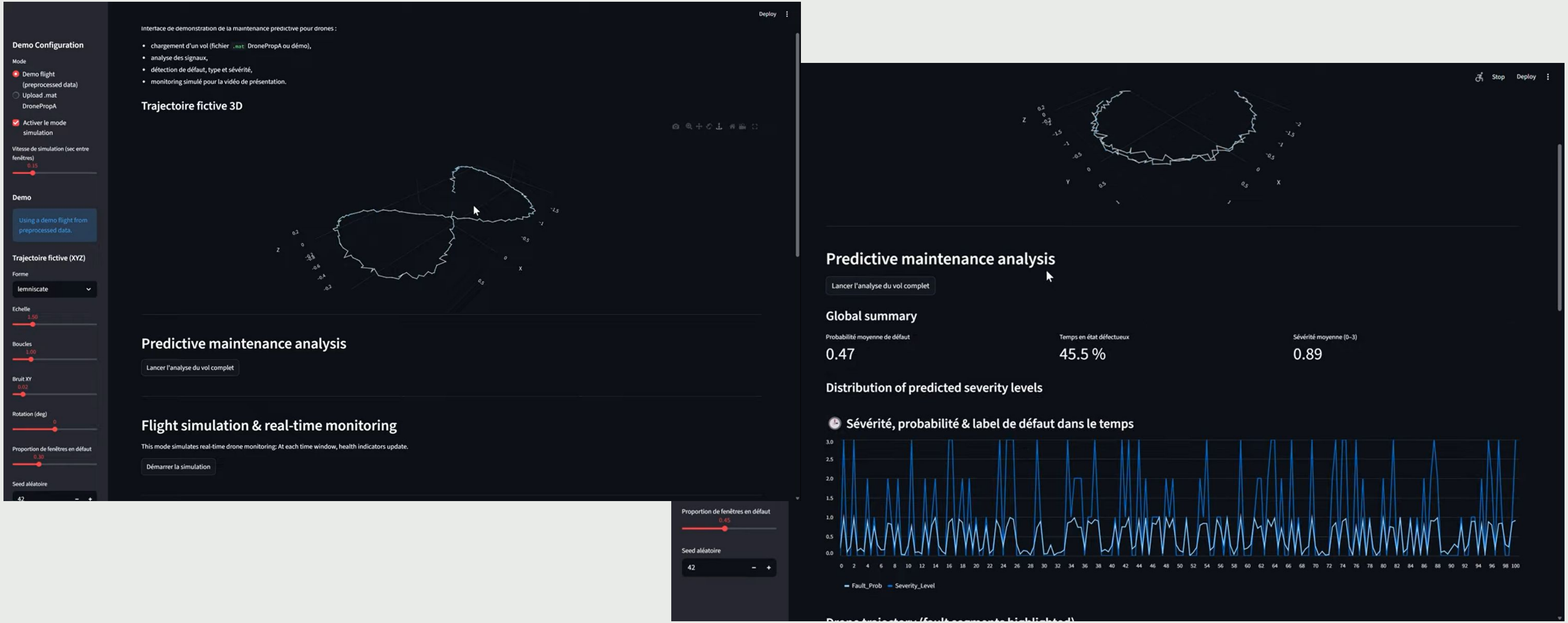
Custom Model Comparison

Metric	XGBoost_GPU	KNN_CUDA	Deep_MTL
Fault Accuracy	0.965	0.772	0.971
Fault F1	0.975	0.818	0.979
Type Accuracy	0.944	0.657	0.926
Type F1	0.943	0.650	0.924
Severity MAE	0.371	0.615	0.146
Severity Accuracy	0.640	0.632	0.902
Combined Accuracy	0.615	0.594	0.856
Combined F1	0.294	0.549	0.324
Training Time	~25min	< 1sec	~5min

Radar Chart Comparison
Between Our Custom Model



StreamLit App



Added Value

Scalable implementation:
- New drone flight regimes can be added to training data and model trained high volume dataset

Multiple ML models considered:
- By considering and testing different ML model types (classic, deep, IBM auto ai) we provide benchmark results for fault prediction in drones

CNN adaptation:
- Codebase adapts CNN deep learning architecture to use case

Data Visualization:
- Power BI based interface allows users to understand flight data

User Interface:
- Stream lit app to visualize predictions from trained ML model, showcasing business use case and serving as a user application

Next Steps

Increase model accuracy:

- Change ML model pipeline to increase model accuracy to have more confidence in application usage.

Collect more flight data in varied scenarios:

- Varied flight scenarios and increased number of samples increases confidence in usability of our solution.

Use an ML based approach to identify causes of faults:

- Conceive of ML methodology to identify causes of faults as an alternative preventative measure approach

Use MLops or IBM tools to integrate ML pipeline in generalized pipeline:

- Automate data addition, model retraining, model selection and user interface integration into one integrated application

Thank You!