CAPSTONE PROJECT

PREDICTIVE MAINTENANCE OF INDUSTRIAL MACHINERY

Presented By:
1. Rishi Bhandari-Graphic Era Deemed to be University-CSE

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Develop a predictive maintenance model for a fleet of industrial machines to anticipate failures before they occur. This project will involve analyzing sensor data from machinery to identify patterns that precede a failure. The goal is to create a classification model that can predict the type of failure (e.g., tool wear, heat dissipation, power failure) based on real-time operational data. This will enable proactive maintenance, reducing downtime and operational costs.

PROPOSED SOLUTION

- **The proposed system aims to address the challenge of predicting industrial equipment failures by accurately classifying the type of failure before it occurs. This involves leveraging data analytics and machine learning techniques to forecast equipment health. The solution will consist of the following components:**
- **Data Collection:**
  - Gather historical operational data from industrial machinery, including sensor readings (e.g., pressure, temperature, vibration) and their corresponding, labeled failure types.
  - Plan to utilize real-time data streams from active machinery sensors to enhance live prediction accuracy and enable early warnings.
- **Data Preprocessing:**
  - Clean and preprocess the collected sensor data to handle missing values, outliers, and inconsistencies. This process was largely automated using the IBM AutoAI tool.
  - Feature engineering to automatically extract relevant features from the data that might impact equipment failure, a key step performed by AutoAI to improve model performance.
- **Machine Learning Algorithm:**
  - Implement a machine learning classification algorithm. The AutoAI experiment automatically tested multiple models and identified a Batched Tree Ensemble Classifier (Snap Random Forest) as the optimal model.
  - Consider the various sensor readings as the input factors to determine the final prediction for the failure type.

edunet
foundation

- **Deployment:**
  - Develop a dashboard or user-friendly interface that provides real-time predictions on the health status of machinery.
  - Deploy the solution on a scalable and reliable platform, IBM Cloud, using a Deployment Space to serve the model as a live API for real-time predictions.
- **Evaluation:**
  - Assess the model's performance using appropriate classification metrics such as Accuracy, Precision, Recall, and F1-Score. The selected model achieved an optimized accuracy of 99.5%.
  - Fine-tune the model periodically with new data and feedback to ensure continuous monitoring and high prediction accuracy.
- Result:
  - The primary result is a highly accurate (99.5%) and fully deployed predictive maintenance model. The system is capable of successfully classifying potential machinery failure types from operational data, enabling proactive maintenance scheduling and reducing costly, unscheduled downtime.

SYSTEM APPROACH

- System requirements

Hardware: A standard development machine with multi-core CPU, minimum of 8GB of RAM and 50GB of available storage space. Cloud-based hardware resources are managed and provided by the IBM Cloud platform.

Software: Operating System, Core language(Python)

Library required to build the model : -

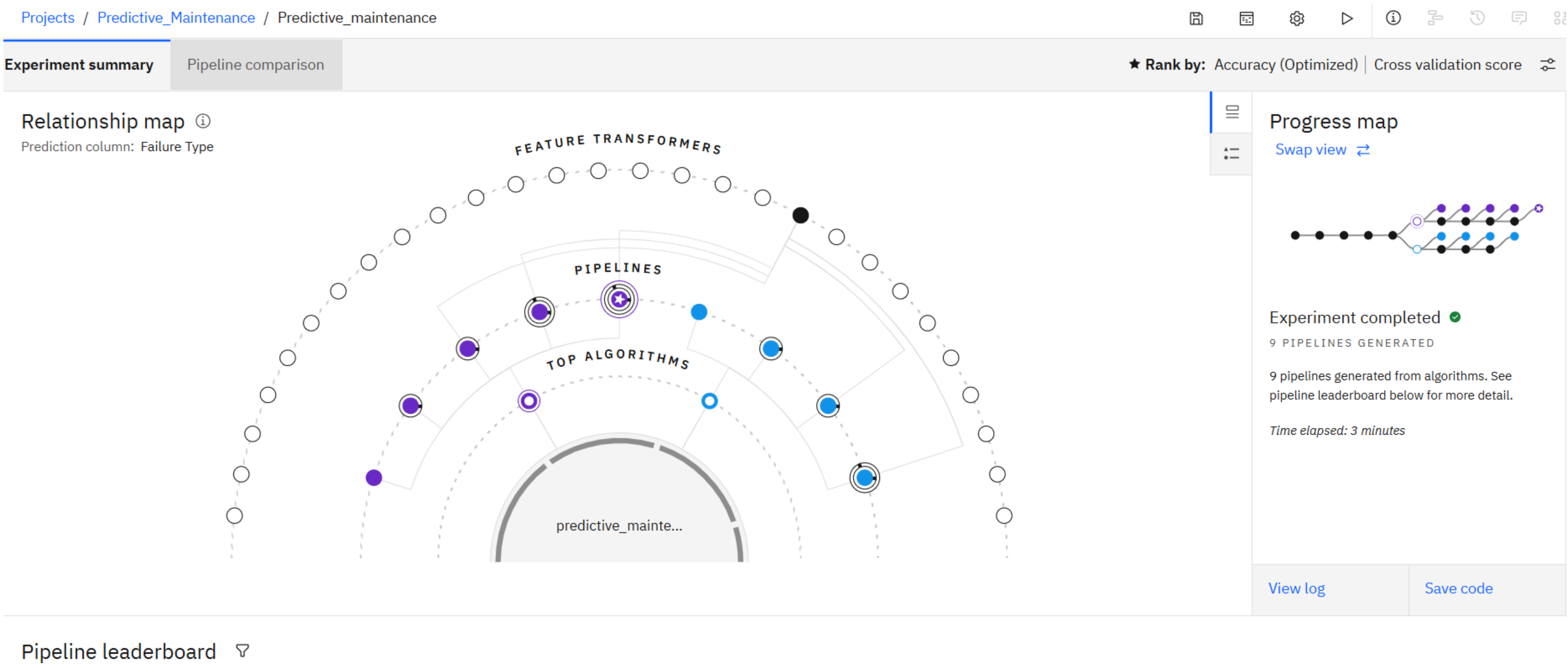
- **Data Manipulation and Analysis:** pandas and numpy
- **Data Visualization:-** matplotlib and seaborn
- **Machine Learning :-** scikit-learn

Cloud Integration :- The official IBM client for programmatically interacting with the deployed model, sending scoring requests and managing deployments.

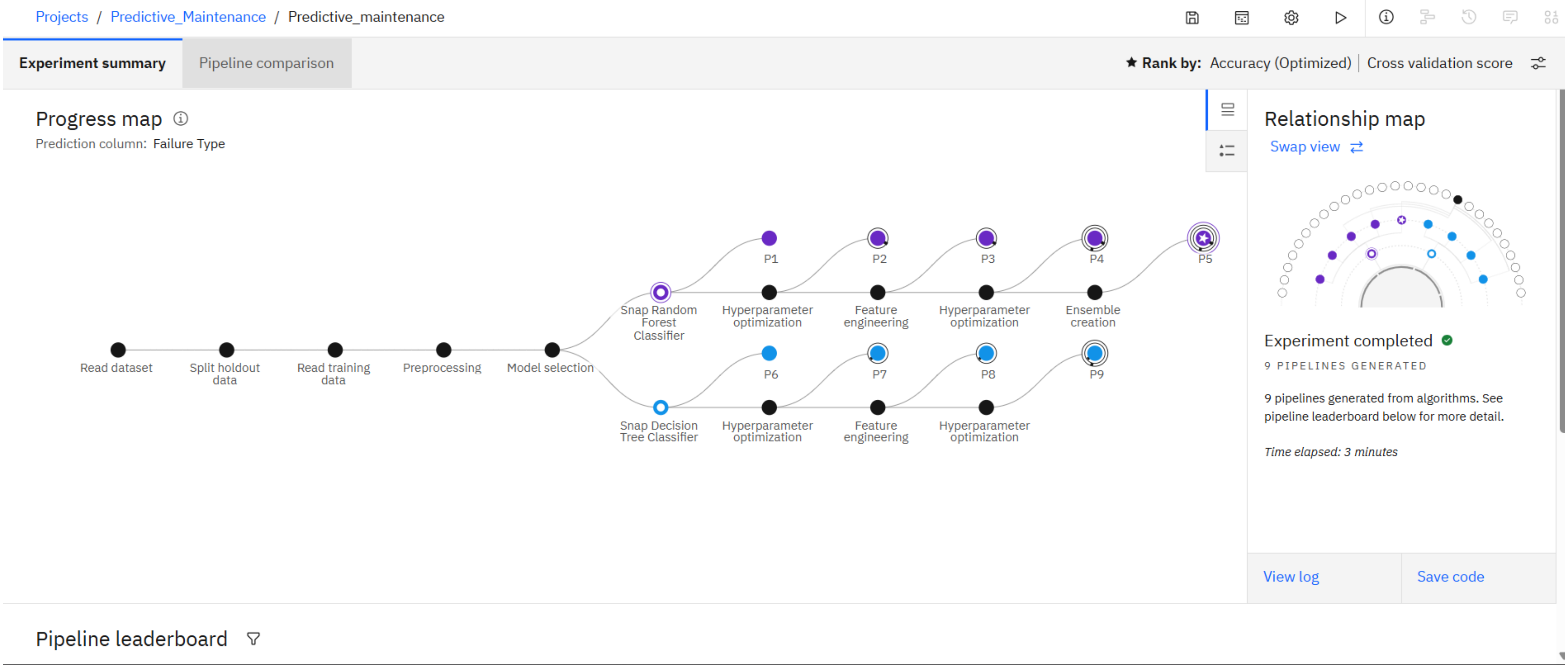
ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**
  - We went with a powerful version of the **Random Forest** algorithm. We chose this because it's known for being great at finding patterns in complex data like in the case of ours. In the end, it wasn't just a hunch—IBM's AutoAI tool ran a competition between several models, and the Random Forest came out as the clear winner with the highest accuracy.
- **Data Input:**
  - The algorithm uses a set of features derived from the industrial machinery's historical operational data. These inputs consist of multiple time-series sensor readings that create a comprehensive snapshot of the machine's health state, such as: Pressure, Vibration, Temperature, Rotational Speed and Torque.
- **Training Process:**
  - The entire training process was automated and optimized using the **IBM AutoAI** tool. It automatically handled advanced techniques like creating new, more impactful features from the raw data (**feature engineering**), finding the ideal model settings (**hyperparameter tuning**), and using **cross-validation** to ensure the 99.5% accuracy was reliable and not just a fluke.
- **Prediction Process:**
  - The trained and deployed model makes predictions via a secure API endpoint. To get a prediction, a new set of sensor readings from a machine is sent to this API. The model processes these inputs through its ensemble of decision trees. In its deployed state, the system is designed to use **real-time data inputs.** A continuous stream of live sensor data from an active machine can be fed to the model's API, allowing it to provide an instantaneous health assessment and predict potential failures as soon as the data patterns emerge.

RESULT



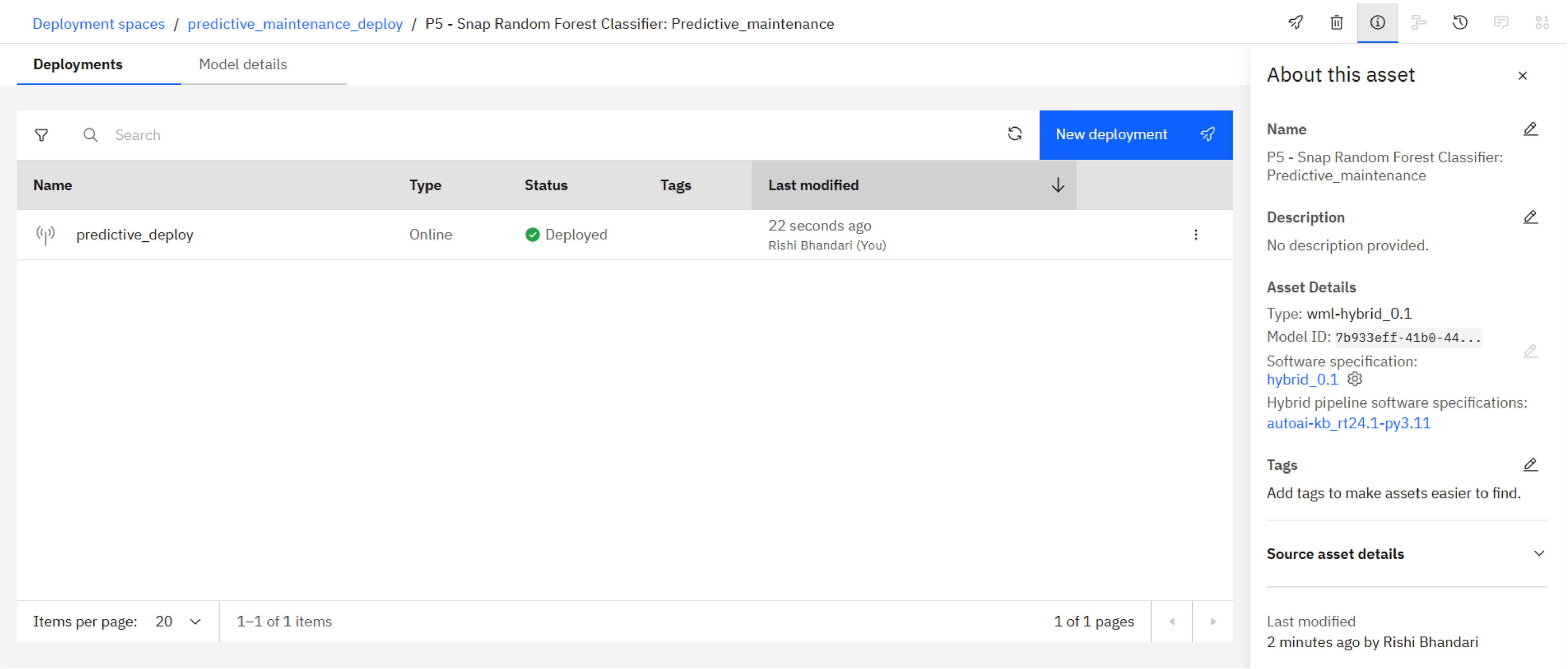
Projects / Predictive_Maintenance / Predictive_maintenance
Experiment summary
Pipeline comparison
★ Rank by: Accuracy (Optimized) | Cross validation score
Relationship map
Prediction column: Failure Type
FEATURE TRANSFORMERS
PIPELINES
TOP ALGORITHMS
predictive_mainte...
Progress map
Swap view
Experiment completed
9 PIPELINES GENERATED
9 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 3 minutes
View log
Save code
Pipeline leaderboard



Projects / Predictive_Maintenance / Predictive_maintenance
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Progress map
Prediction column: Failure Type
Read dataset
Split holdout data
Read training data
Preprocessing
Model selection
Snap Random Forest Classifier
Snap Decision Tree Classifier
Hyperparameter optimization
Feature engineering
Hyperparameter optimization
Ensemble creation
P1
P2
P3
P4
P5
P6
P7
P8
P9
Relationship map
Swap view
Experiment completed
9 PIPELINES GENERATED
9 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 3 minutes
View log
Save code
Pipeline leaderboard

Pipeline leaderboard

	Rank	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	**Pipeline 5**	Batched Tree Ensemble Classifier (Snap Random Forest Classifier)	INCR	0.995	HPO-1 FE HPO-2 BATCH	00:00:43
	2	**Pipeline 4**	Snap Random Forest Classifier		0.995	HPO-1 FE HPO-2	00:00:41
	3	**Pipeline 3**	Snap Random Forest Classifier		0.995	HPO-1 FE	00:00:32
	4	**Pipeline 9**	Snap Decision Tree Classifier		0.994	HPO-1 FE HPO-2	00:00:04



edunet foundation

UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target
249	L47428	L	298	308.3	1362	56.8	216	1
208	M15067	M	298.4	308.7	1421	60.7	119	1
501	M15360	M	297.5	309.2	1804	27.4	215	0
623	M15482	M	298.4	310.2	1725	32	80	0
717	H30130	H	297.4	308.6	1420	45.8	120	0
848	L48027	L	296.4	307.4	2833	5.6	213	1
927	L48106	L	295.6	306.1	1372	55.6	215	1
6092	L53271	L	300.9	310.7	1412	57.5	16	0

Prediction results

Close

Display format for prediction results

Table view JSON view

Show input data

	prediction	probability
1	Overstrain Failure	[0.0030303031206130983,0,0.9969696998596191,0,0,-2.9802322831784522e-9]
2	Power Failure	[0,0,0,1,0,0]
3	No Failure	[0,0.9998846530914307,0,0,0.00011534024961292744,6.658956386296211e-9]
4	No Failure	[0,1,0,0,0,0]
5	No Failure	[0,1,0,0,0,0]
6	Power Failure	[0,0,0,1,0,0]
7	Overstrain Failure	[0.0030303031206130983,0,0.9969696998596191,0,0,-2.9802322831784522e-9]
8	No Failure	[0,0.9997901439666749,0,0,0.00020986357703804971,-7.543712987612139e-9]
9		

CONCLUSION

Key Findings & Solution Effectiveness

- **High Accuracy Achieved:** The solution, built using **IBM AutoAI**, successfully produced a predictive model with **99.5% accuracy** for classifying machinery failure types. The top-performing algorithm was a **Batched Tree Ensemble Classifier**.
- **Rapid & Effective Implementation:** The entire process of data analysis, feature engineering, model training, and optimization was automated and completed in **under 3 minutes**. This demonstrates an extremely effective and efficient pathway from data to a deployed, high-value AI solution.

Challenge :- The current model was trained on a historical dataset. The primary operational challenge is integrating a live, continuous stream of data from active machinery.

Improvement :- Connect the model to live IoT data streams for real-time monitoring and prediction.

FUTURE SCOPE

This project successfully validated a powerful, data-driven approach to predictive maintenance, establishing a robust foundation for a truly intelligent operational ecosystem.

Core Achievement:

- A highly accurate (**99.5%**) model for classifying machinery failure types was rapidly developed and deployed using **IBM AutoAI**.
- This demonstrates the viability of automating the machine learning pipeline to turn operational data into actionable intelligence efficiently.

Future Work:

- Incorporate additional data sources, such as unstructured **maintenance logs** (via NLP) and **real-time IoT sensor streams**, to create a more holistic and resilient predictive system covering multiple, complex failure signals.
- Deploying lightweight versions of the model directly on edge devices near the machinery. This will enable immediate, low-latency predictions and actions without relying on constant cloud connectivity.

REFERENCES

In developing the predictive maintenance solution for industrial machinery, several key resources played a crucial role in shaping the approach. The foundational concepts were drawn from academic works like Susto et al.'s study on machine learning classifiers for predictive maintenance and Zhang et al.'s comprehensive survey of data-driven maintenance strategies. Practical implementation was guided by documentation from Scikit-learn, which provided robust tools for model training, evaluation, and preprocessing. Additionally, best practices in handling real-world data—such as dealing with missing values and scaling—were adopted from industry blogs and resources like Machine Learning Mastery. The deployment phase benefited from IBM Cloud's official documentation, which offered clear guidance on integrating machine learning models into cloud-based environments using IBM Watson Studio.

IBM CERTIFICATIONS

Getting Started with Artificial Intelligence
IBM SkillsBuild

In recognition of the commitment to achieve professional excellence

Rishi Bhandari

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/b356bd48-8303-4120-b42b-89bf17583ffd

IBM

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Rishi Bhandari

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 21, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/9b153ae9-b43f-4930-858d-6dc9368b34a0

IBM

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Rishi Bhandari

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT) **Learning hours:** 20 mins

THANK YOU