Hewlett Packard
Enterprise

Proactive precision

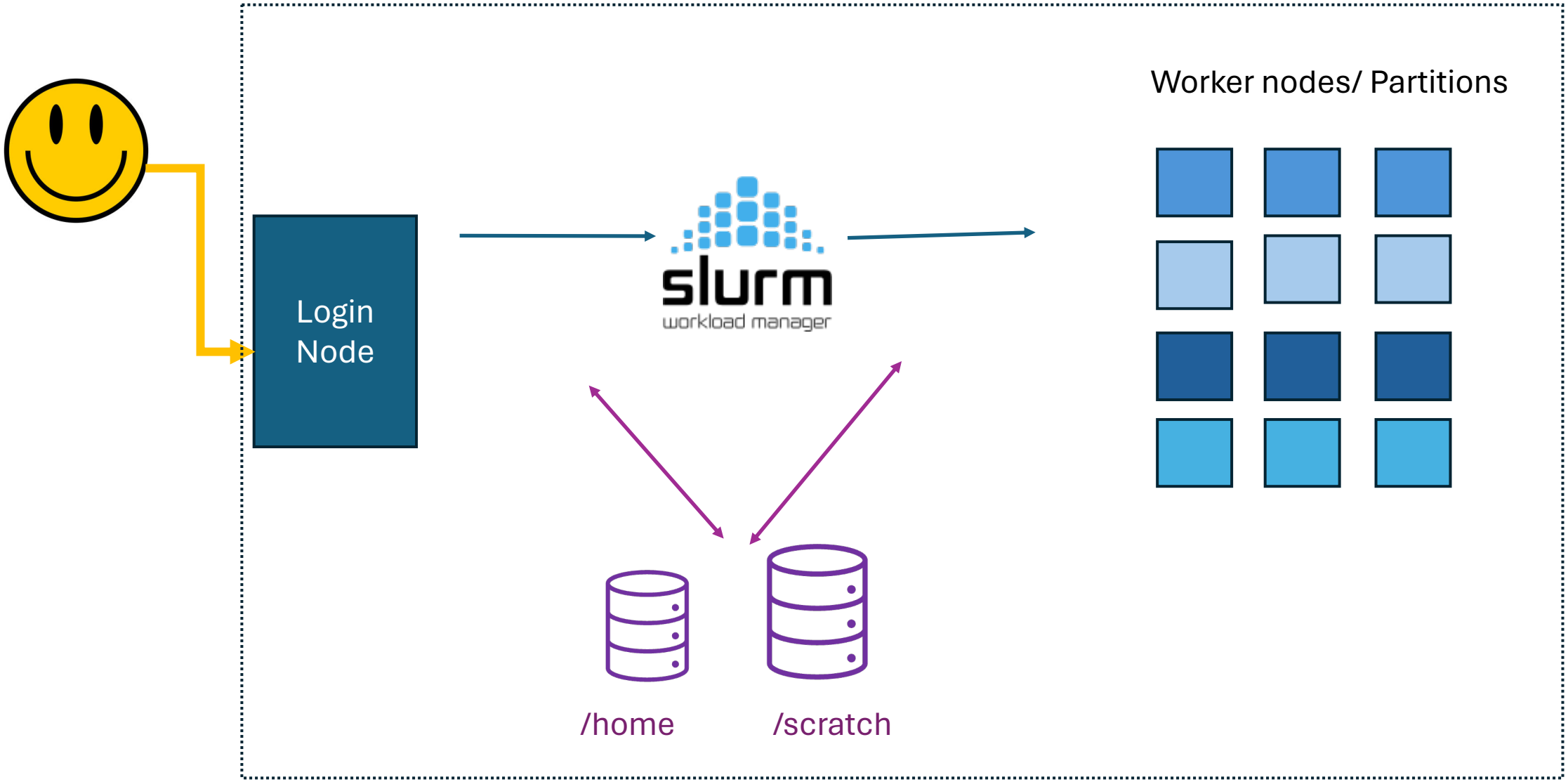
Enhancing High-Performance Computing with Early Job Failure Detection

Presenter: Saptashwa Mitra

Authors: Dipanwita Mallick, Siddhi Potdar, Saptashwa Mitra, Charlie Vollmer, and Nithin Singh Mohan

Problem Definition

Background


Login Node
slurm
workload manager
Worker nodes/ Partitions
/home
/scratch

Understanding the problem statement

What happens when there is lack of proactive job failure detection mechanisms ?

- ❌ Wasted compute resources
- ❌ Increased wait times
- ❌ Reduced user productivity
- ❌ Inefficient troubleshooting

Solution

Proactive job failure detection system

- accurately predict the likelihood of job failures in real-time

- ✓ Improved resource utilization
- ✓ Minimize downtime
- ✓ Improve system performance
- ✓ Enhanced troubleshooting capabilities

Solution to the user need

Proactive job failure detection system

- accurately predict the likelihood of job failures in real-time

✓Why my job is running slow?

Value Proposition

Hybrid Approach: Combining **supervised** and **unsupervised** learning for accurate predictions and deeper insights.

Oversampling Methodology: Addressing **class imbalance** in job failure data for reliable and accurate predictions.

User-Centric Design: Real-time results pipeline and **user-friendly interface** for actionable feedback to HPC stakeholders.

Post-Prediction Insights: Understanding failure causes through **model's feature interpretation**.

Continuous Retraining: **Adaptable** and **self-improving predictive model** that evolves with new data and system dynamics.

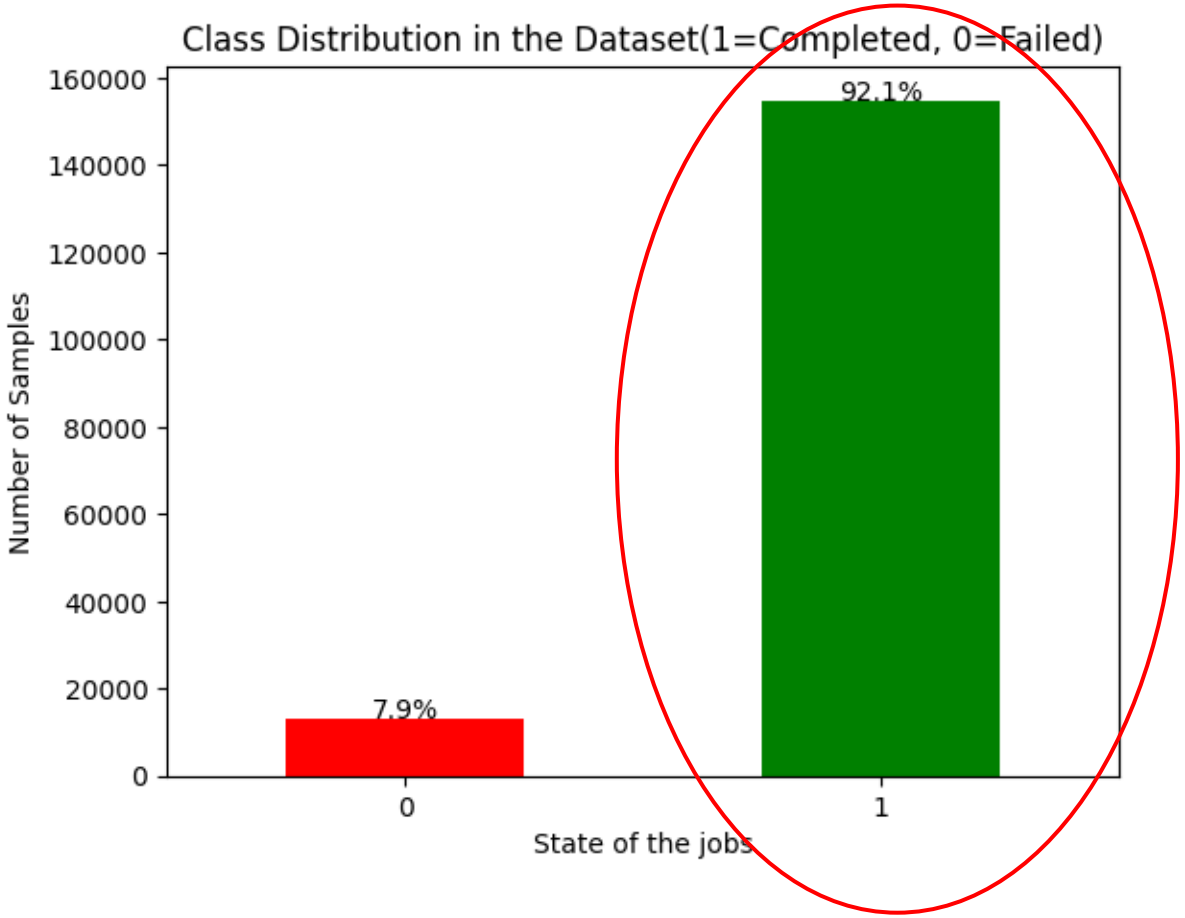
Methodology

Overview: System, Slurm and Data Collection

- System considered: **Hotlum**, with over **1027 nodes** and **9 groups**
- Slurm Accounting (sacct): Offers 100+ fields for analysis
- Focused features:
  - **User ID, Job Name, State, CPU, Memory, Nodes, CPUTime, Submit Time, Start Time, End Time, GID**
- Modelling predictions for **state_jobs**

job_id	job_name	user_id	start_time	end_time	state_jobs
502	shared_coll	mmoore	2022-11-08 10:28:50	2022-11-08 10:28:51	FAILED
503	shared_coll	mmoore	2022-11-08 10:32:12	2022-11-08 10:32:23	FAILED
504	shared_coll	mmoore	2022-11-08 10:34:11	2022-11-08 10:34:16	COMPLETED
505	hpl_sweep	jbaron	2022-11-08 11:04:08	2022-11-08 11:04:09	COMPLETED
506	hpl_sweep	jbaron	2022-11-08 11:06:26	2022-11-08 11:11:12	COMPLETED
507	shared_coll	mmoore	2022-11-08 11:10:03	2022-11-08 11:10:08	COMPLETED
508	shared_coll	mmoore	2022-11-08 11:33:08	2022-11-08 11:33:12	COMPLETED
509	shared_coll	mmoore	2022-11-08 11:33:50	2022-11-08 11:34:02	COMPLETED
510	shared_coll	mmoore	2022-11-08 11:34:54	2022-11-08 11:35:03	COMPLETED
511	shared_coll	mmoore	2022-11-08 11:37:40	2022-11-08 11:37:52	COMPLETED
512	shared_coll	mmoore	2022-11-08 11:39:49	2022-11-08 11:40:02	COMPLETED
513	shared_coll	mmoore	2022-11-08 11:40:55	2022-11-08 11:42:10	COMPLETED
514	shared_coll	mmoore	2022-11-08 11:43:27	2022-11-08 11:44:43	COMPLETED
515	shared_coll	mmoore	2022-11-08 11:44:56	2022-11-08 12:01:42	COMPLETED
516	shared_coll	mmoore	2022-11-08 12:01:43	2022-11-08 12:02:32	COMPLETED
517	hpcg_sweep	jbaron	2022-11-08 11:48:55	2022-11-08 12:09:00	COMPLETED
518	hpcg_sweep	jbaron	2022-11-08 11:53:32	2022-11-08 12:17:06	COMPLETED
519	hpcg_sweep	jbaron	2022-11-08 11:58:49	2022-11-08 12:18:59	CANCELLED by 213865
520	hpcg_sweep	jbaron	2022-11-08 12:05:52	2022-11-08 12:16:44	OUT_OF_MEMORY
521	hpcg_sweep	jbaron	2022-11-08 12:20:11	2022-11-08 12:52:20	CANCELLED by 213865
522	hostname	dchrist	2022-11-08 12:20:46	2022-11-08 12:20:46	COMPLETED
523	shared_coll	mmoore	2022-11-08 12:52:25	2022-11-08 12:53:12	COMPLETED
524	interactive	mmoore	2022-11-08 13:10:57	2022-11-08 13:11:06	FAILED
525	shared_coll	mmoore	2022-11-08 13:11:29	2022-11-08 13:12:59	COMPLETED
526	shared_coll	mmoore	2022-11-08 13:12:59	2022-11-08 13:13:55	COMPLETED
527	shared_coll	mmoore	2022-11-08 13:17:58	2022-11-08 13:18:02	CANCELLED by 14759

Skewness of the data



Data Preprocessing

Labeling and Problem Framing: Encoding **state_jobs** labels

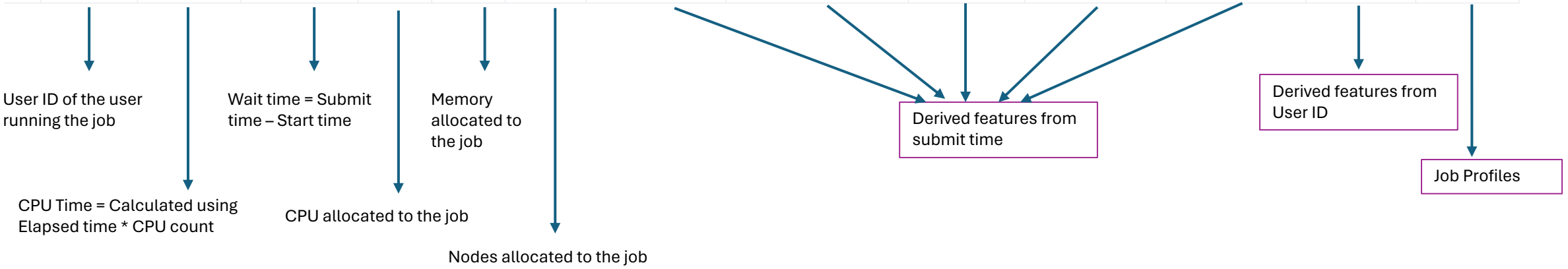
Preprocessing: **Eliminating features unimportant** to the prediction

Adding **Derived Features**: Job Profiles, User Segments

Data Ready!

Snapshot of the data ready for modeling

	user_id	cputime	wait_time	cpu	mem	node	submit_month	submit_day_of_week	submit...	submit_wee...	submit_ti...	segments	cluster
0	22	15001600	0	51200.0	89200.0	200	10	3	4	42	1	1	1
1	22	116992000	0	233984.0	407644.0	914	10	3	4	42	1	1	1
2	22	79104000	0	128000.0	223000.0	500	10	3	4	42	1	1	1
3	22	139392000	64	230400.0	401400.0	900	10	3	4	42	1	1	1
4	22	95846400	0	153600.0	267600.0	600	10	3	4	42	1	1	1



Modeling

Tree Based Models

- Capture complex relationships and patterns
- Work well with skewed data
- Robust to outliers
- Low risk of overfitting

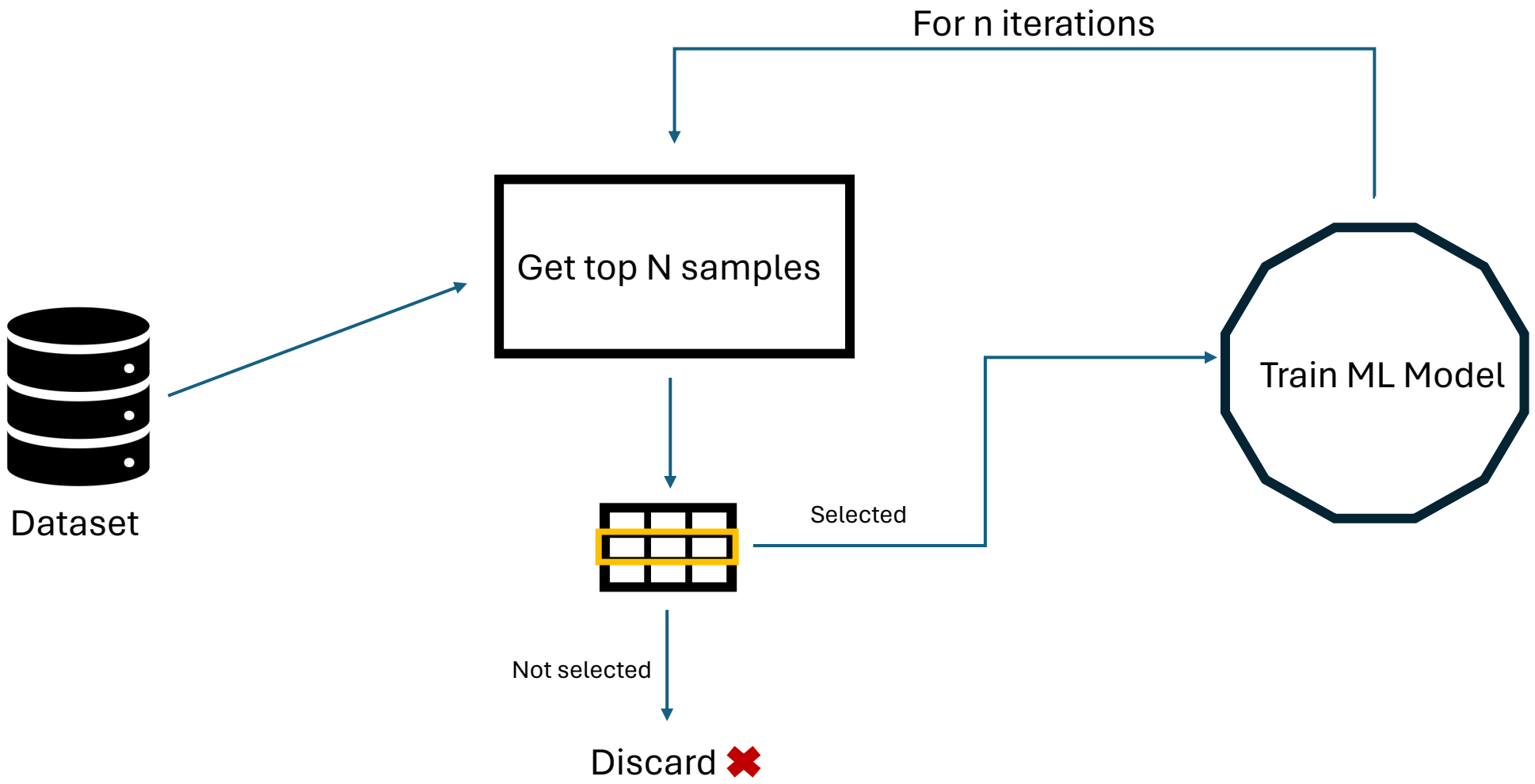
Oversampling

- Helps in handling skewness by:
  - Creating synthetic samples
  - Duplicating existing samples
- Improves model performance by assisting underrepresented class

Active Learning

- Dynamically selects top N data points using clustering
- Works well with limited data
- Robust to changes in data
- Faster, uses less iterations to train

Active Learning

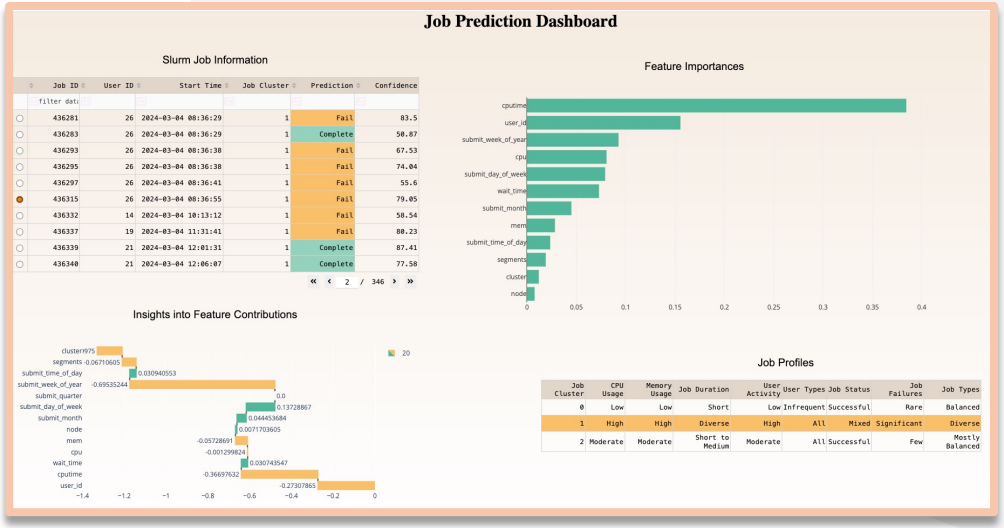


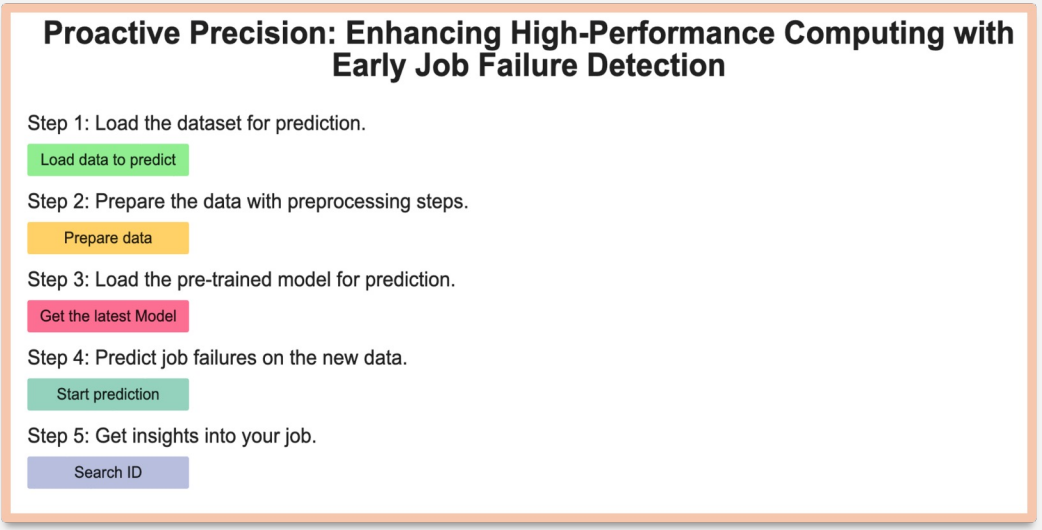
Results and Insights

Results

Method	Results	Accuracy
XG Boost	✓ Performs well on **completed** class ✓ Trains fast ✗ Incorrectly predicts **failed** jobs as completed	94%
Random Forest	✓ Performs extremely well on **completed and failed** classes ✗ Slow to train ✗ Overfits on the dataset	98%
XG Boost + Random Oversampling	✗ Performs extremely poorly on **failed** class ✗ Slow to train	58%
XG Boost + SMOTE Oversampling	✗ Performs extremely poorly on **failed** class ✗ Slow to train	70%
XG Boost + Active Learning	✓ Performs well on **completed** class ✓ Performs well on **failed** class ✓ Trains fast	97%

Model inference for end-user

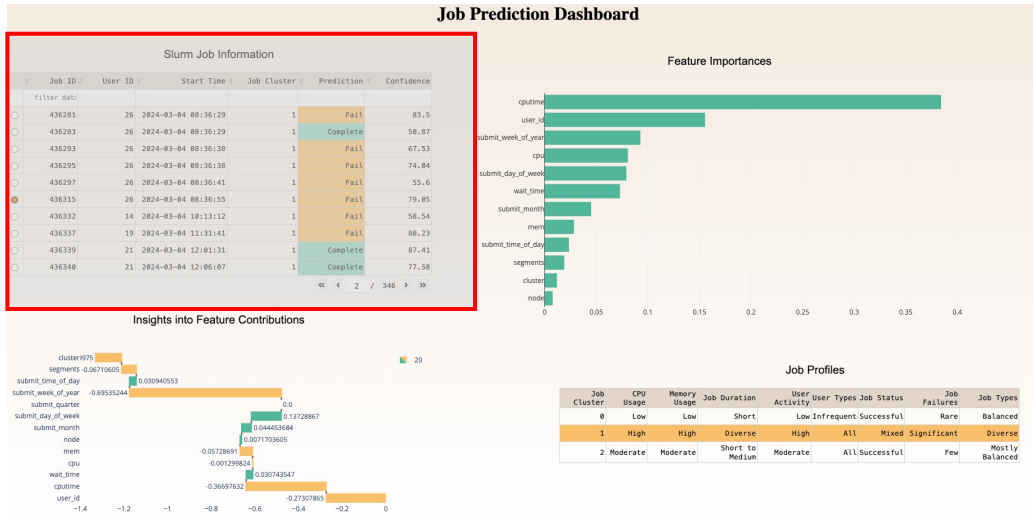




- ❑ Real-time visibility for proactive decision-making.
- ❑ Intuitive and accessible for all users.
- ❑ Allows customization and filtering.
- ❑ Provides a centralized platform to view and discuss.

- ❑ Modify and customize the analysis workflow.
- ❑ Streamline reproducibility, documentation.
- ❑ Integration with Libraries.
- ❑ Rapid Prototyping and Iteration.

Real-time analytics with dashboard

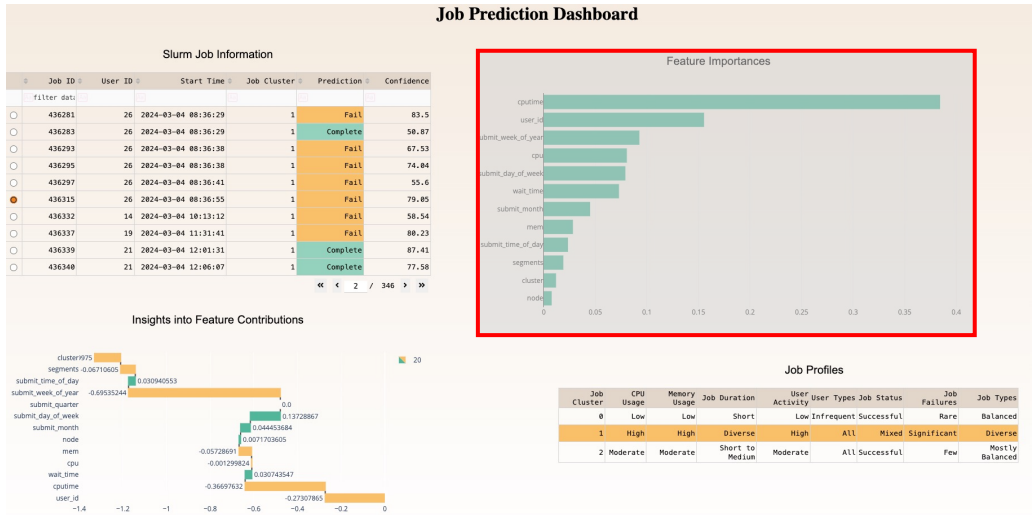


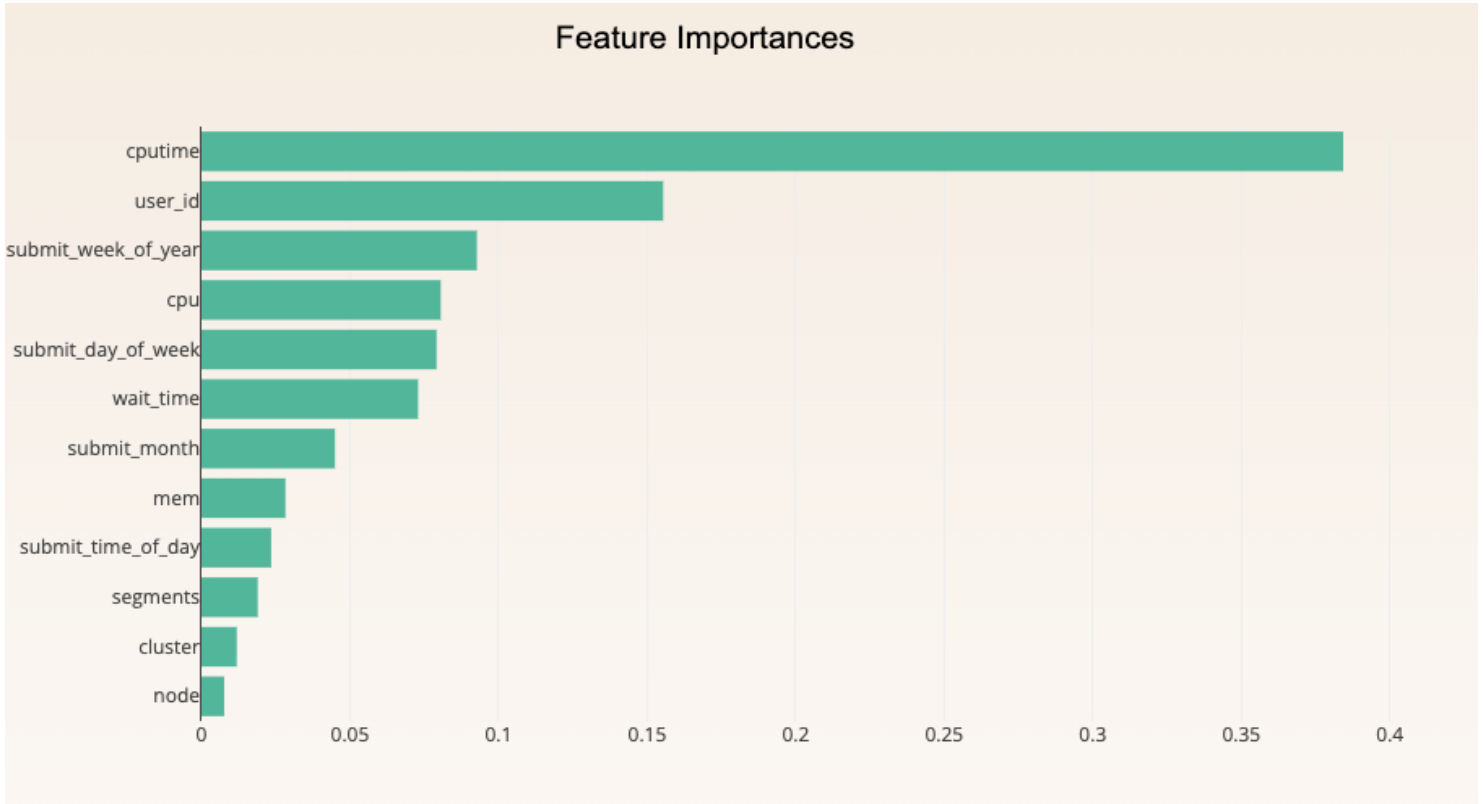
Slurm Job Information

	Job ID	User ID	Start Time	Job Cluster	Prediction	Confidence
	filter data					
○	436281	26	2024-03-04 08:36:29	1	Fail	83.5
○	436283	26	2024-03-04 08:36:29	1	Complete	50.87
○	436293	26	2024-03-04 08:36:38	1	Fail	67.53
○	436295	26	2024-03-04 08:36:38	1	Fail	74.04
○	436297	26	2024-03-04 08:36:41	1	Fail	55.6
●	436315	26	2024-03-04 08:36:55	1	Fail	79.05
○	436332	14	2024-03-04 10:13:12	1	Fail	58.54
○	436337	19	2024-03-04 11:31:41	1	Fail	80.23
○	436339	21	2024-03-04 12:01:31	1	Complete	87.41
○	436340	21	2024-03-04 12:06:07	1	Complete	77.58

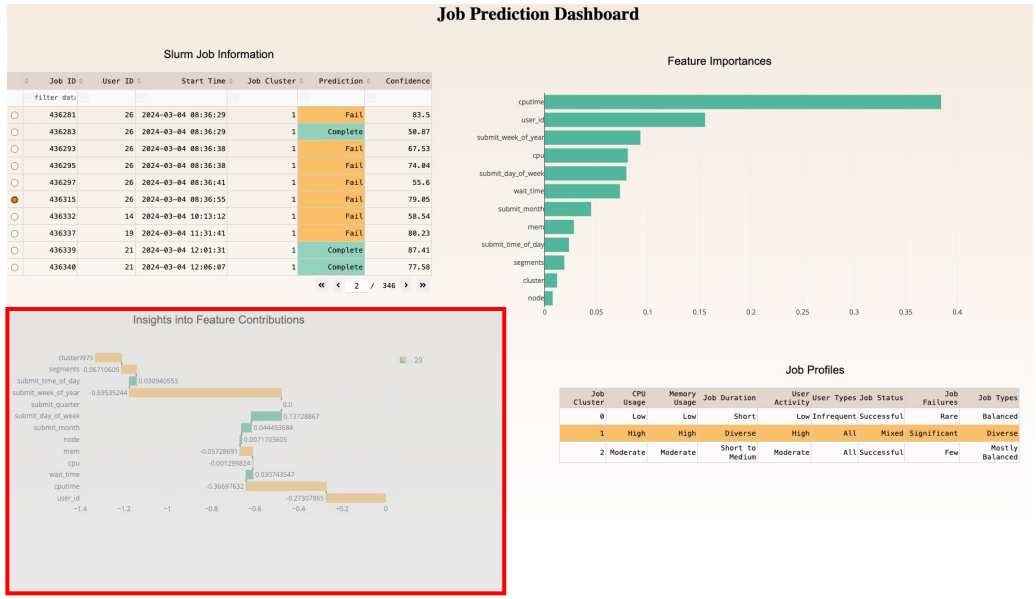
« ‹ 2 / 346 › »

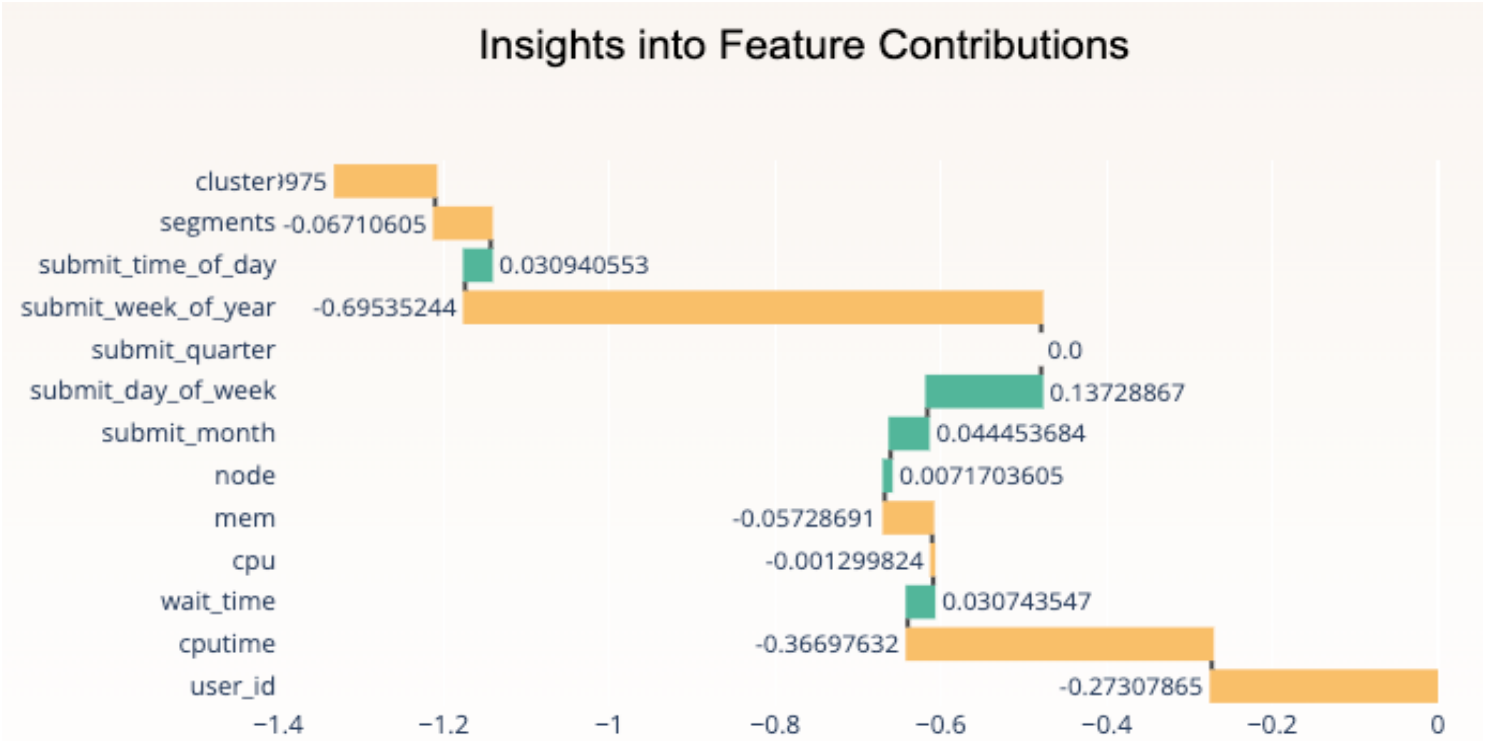
Real-time analytics with dashboard



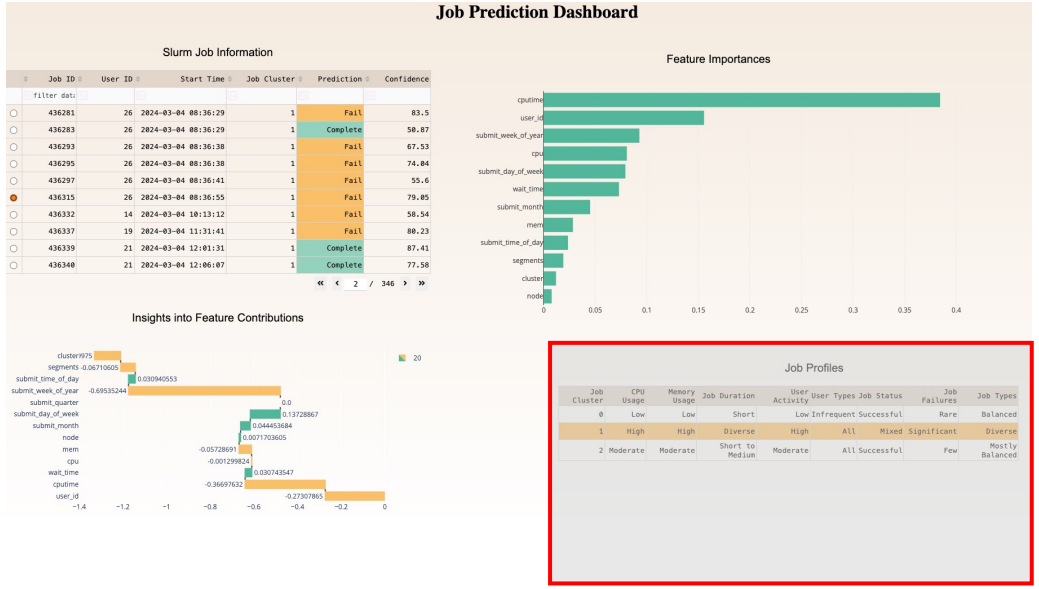


Real-time analytics with dashboard





Real-time analytics with dashboard



Job Profiles

Job Cluster	CPU Usage	Memory Usage	Job Duration	User Activity	User Types	Job Status	Job Failures	Job Types
0	Low	Low	Short	Low	Infrequent	Successful	Rare	Balanced
1	High	High	Diverse	High	All	Mixed	Significant	Diverse
2	Moderate	Moderate	Short to Medium	Moderate	All	Successful	Few	Mostly Balanced

Solving user pain points

Stakeholders	Interests and Needs	How Our Solution Offers
System Administrators	▪ Minimize downtime ▪ Efficient resource utilization ▪ **Need**: Optimize system performance	✓ Provides real-time insights ✓ Proactive failure detection ✓ Actionable management information(future scope)
Users and Researchers	▪ Reliability and performance of HPC jobs ▪ **Need:** Minimize job failures and delays	✓ Identifies potential issues ✓ Proactive failure detection
Data Scientists and ML Experts	▪ Advanced analytics and modeling techniques ▪ **Need:** Develop custom models	✓ Facilitates Jupyter notebook integration ✓ Enables data exploration and model experimentation
HPC Application Developers	▪ Create and optimize software applications that run on HPC systems. ▪ **Need**: Profile and debug their applications, identify performance bottlenecks, and optimize resource utilization.	✓ Insights into job behavior and resource usage ✓ Identification of potential failure points for app optimization(future scope)

Future Work

- **Expand data scope** - system logs, network performance metrics, and application-specific telemetry
- **Scalability assessments** of our model across diverse HPC platforms.
- Prioritize user experience improvements by **actively incorporating feedback**.
- Implement **cutting-edge machine learning techniques**, including deep learning and transfer learning.
- Dynamic **resource allocation recommendation**.

Thank you!