Predict CMS Data Popularity to Improve Its Availability for Physics Analysis

Andrii Len

Taras Shevchenko National University of Kyiv

Dmytro Kovalskyi, Rahul Chauhan, Hasan Ozturk

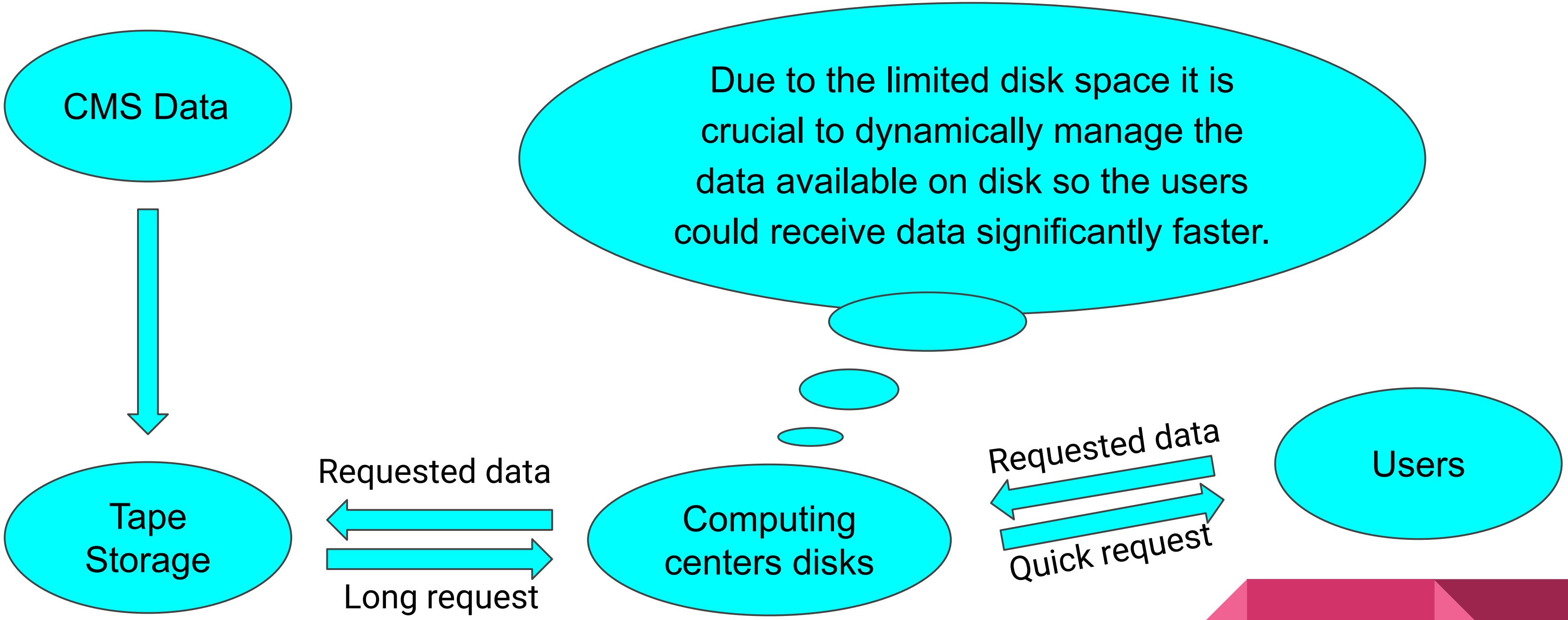
MIT, CERN

IRIS-HEP Summer Fellowship

October 16, 2023

Introduction

CMS data management



A typical question that we want to answer:

Which datasets we can delete from cache?

Plan of the project

- ➢ Collection of the data usage data
- ➢ Feature Engineering
- ➢ Searching for the best Machine Learning approach
- ➢ Model evaluation

Data gathering. Data structure and Features selection

Data Extraction with Spark

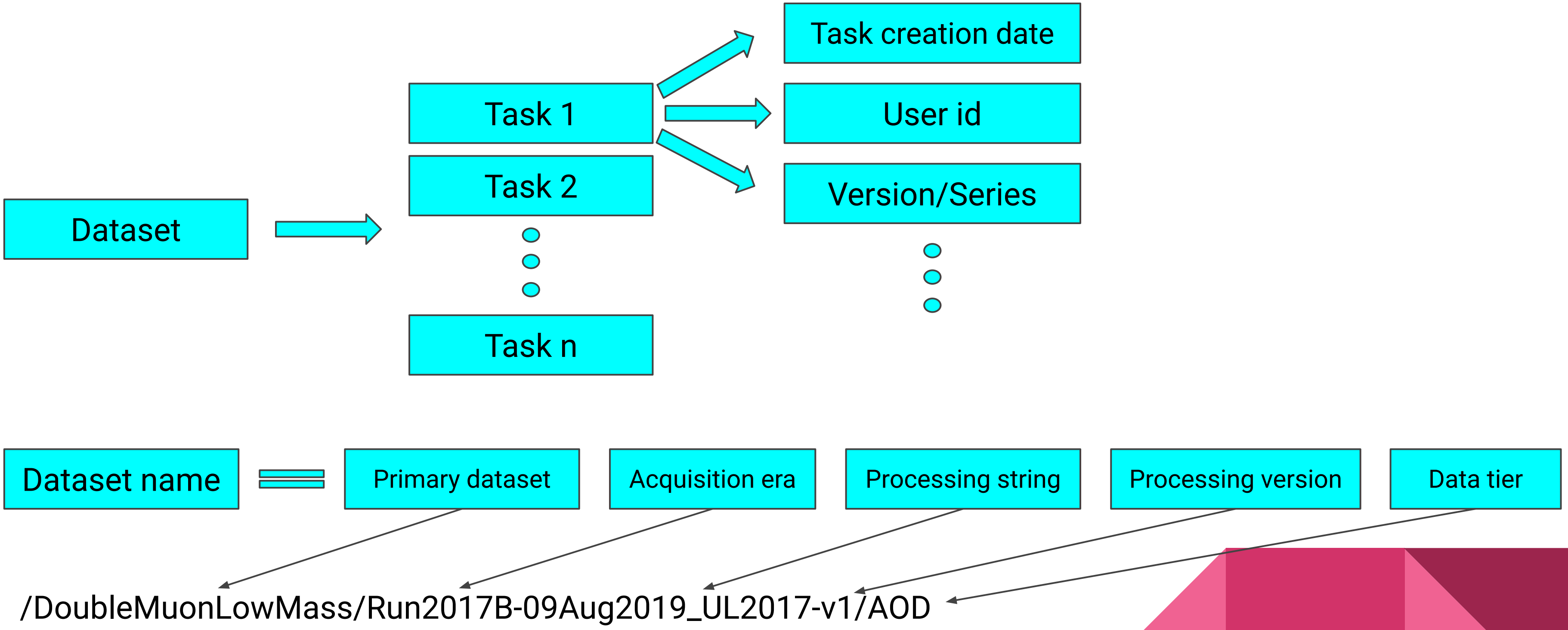
Extracted columns from CRAB:

- ❑ CRAB_Workflow
- ❑ DESIRED_CMSDataset
- ❑ CMSSWMajorVersion
- ❑ CMSSWRealeaseSeries
- ❑ CRAB_TaskCreationDate
- ❑ CRAB_UserHN

Historical data taken:

- ❖ 2020 (01.06 - 31.12) 7 months
- ❖ 2021 full
- ❖ 2022 full
- ❖ 2023 (01.01 - 31.08) 8 months

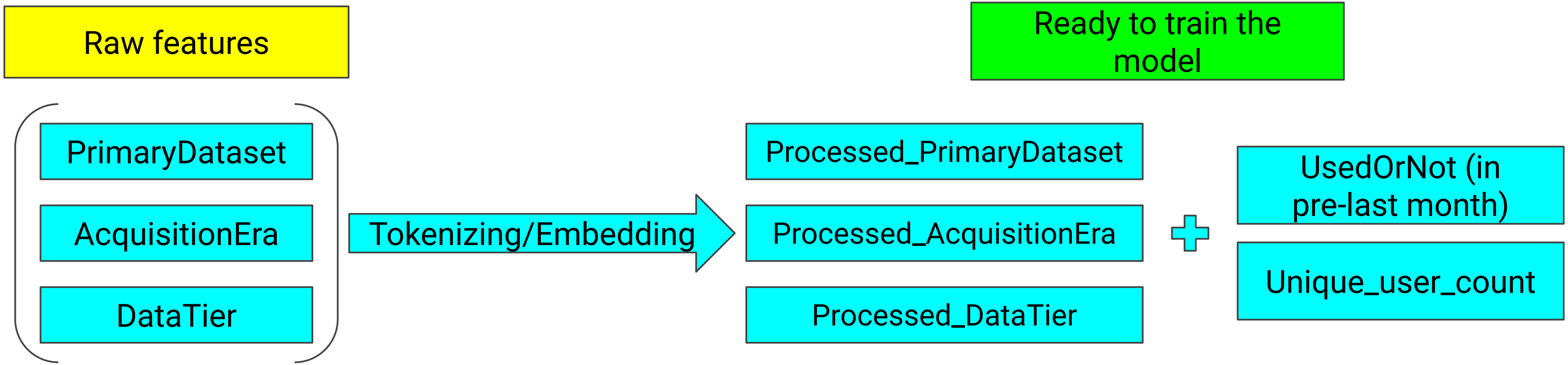
Data structure



Features we tried

1. ‘Counts’ - number of tasks per dataset.
2. ‘Earliest_time’, ’Latest_time’ - first and last time when the dataset was used (dataset usage time frame).
3. ‘UsedOrNot’ - 1 or 0 based on if the dataset was used in previous month.
4. ‘Usage_n_last)month’ - sequences of frequency of data usage during n last months (for example, if n = 4: [2, 0, 14, 4])
5. ‘Unique_user_count’ - How many different users used a particular dataset.
6. ‘PrimaryDataset’, ‘AcquisitionEra’, ‘ProcessingString’, ‘ProcessingVersion’, ‘DataTier’

Machine learning: Preprocessing



- Tokenizer creates dictionary of elements and turns a string into a vector of numbers.
- Embedding layers are used to map discrete tokens or integers into a continuous vector space. They are an additional layers in ML model that learn alongside with main layers and allow the model to capture semantic relationships between tokens.

Model Training and Performance Evaluation

Machine Learning Model Architecture and Tools

Model Tools:

1. Programming Language: Python
2. Libraries: TensorFlow/Keras (for model development)
3. Other Tools: tokenizer (for word indexing)

Model Components:

1. Input Layers: 5
2. Embedding Layers: 2 (PrimaryDataset, AcquisitionEra)
3. Flatten Layers: 2 (Embedded PrimaryDataset and AcquisitionEra)
4. Dense layers for feature transformation: 3
5. Activation functions: ReLU.
6. Output Layer: Single neuron output layer with sigmoid activation for binary classification.

Model Training:

1. Optimizer: Adam
2. Loss Function: Binary Cross-Entropy
3. Metrics: Accuracy
4. Training on training data for 8 epochs with a batch size of 128.

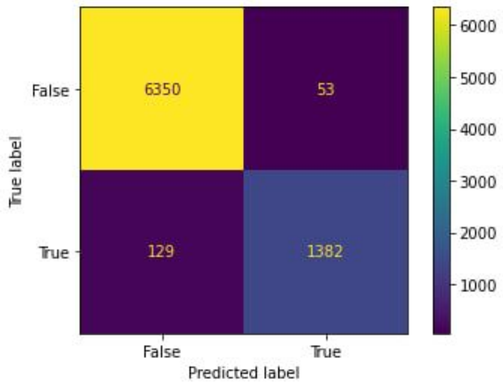
https://gitlab.cern.ch/cmsdmops/CMSDataPopularity

Take a look on everything we have done on our gitlab

Our evaluation metrics

General:

- **Precision**: (True Positive)/(True Positive+False Positive).
- **Recall**: (True Positive)/(True Positive+False Negative).
- **F1 Score**: 2*(Precision*Recall/(Precision + Recall)).



Specially for our data:

- ‘Unused, Unused’ - number of datasets that were not used in previous month and not used in current month.
    - Predicted used (Wrong)
    - Predicted unused (Correct)
- ‘Unused, Used’
- ‘Used, Unused’
- ‘Used, Used’

Usage in previous month	Unused		Used	
Unused	Predicted as Used: 42	Predicted as Unused: 9381	Predicted as Used: 609	Predicted as Unused: 80
Used	Predicted as Used: 13	Predicted as Unused: 336	Predicted as Used: 276	Predicted as Unused: 28

Usage in current (target) month

Precision: 0.94

Recall: 0.89

F1 Score: 0.92

Caching testing

Current caching algorithm - LRU

LRU - Least Recently Used

Incoming datasets: A -> B -> C -> B -> D

Cache size 3

	Step 1	Step 2	Step 3
Most recently used	C	B	D
	B	C	B
Least recently used	A	A	C

Caching with ML model

Incoming datasets: A -> B -> C -> D -> E

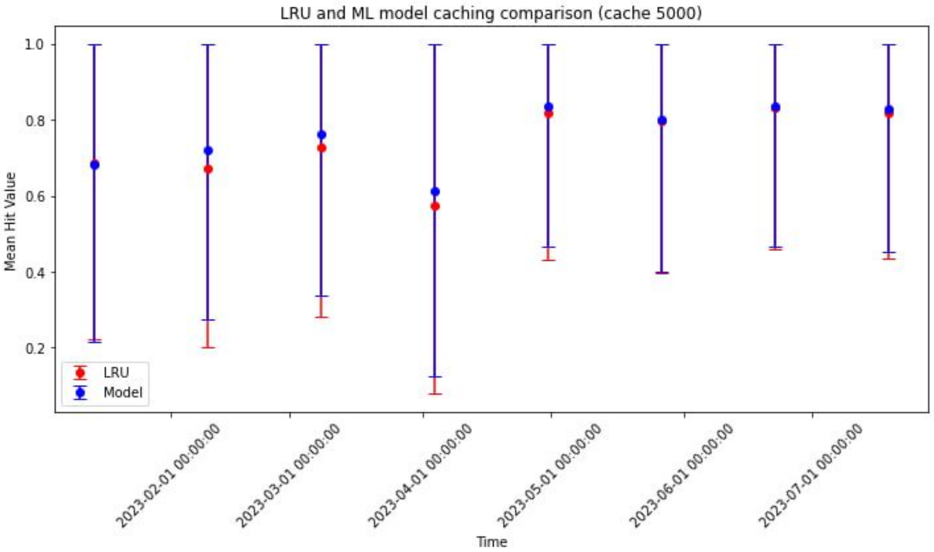
Predictions:

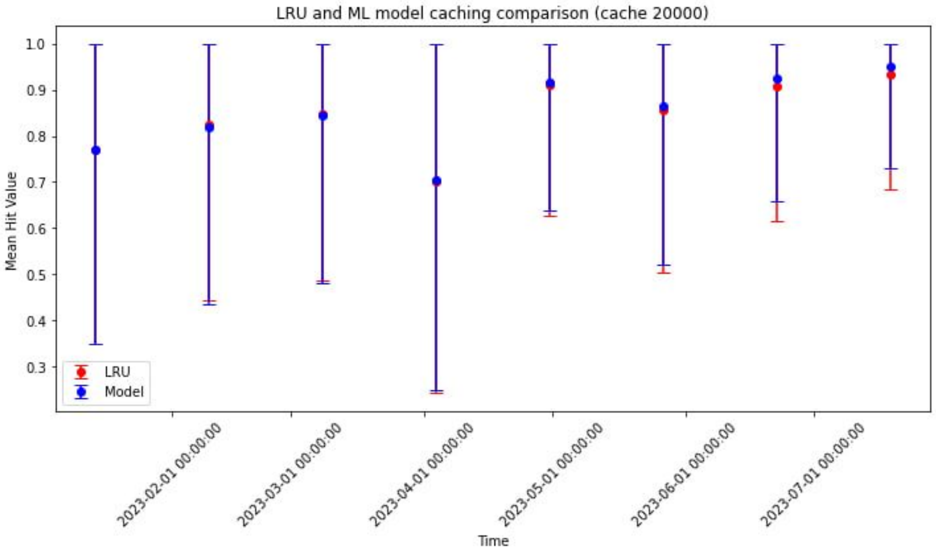
A - 0.44
B - 0.87
C - 0.98
D - 0.69
E - 0.94

Cache size 3

	Step 1	Step 2	Step 3
Highest prediction	C	C	C
	B	B	E
Lowest prediction	A	D	B

LRU and ML model caching comparison





If incoming dataset is in cache - we call it hit and store this event as "True" or "1". If dataset is not in cache - we call it miss, put this dataset in cache while deleting some other dataset (which one to delete is decided based on specific algorithm) and store as 'False' or '0'.

Hit/miss ratio comparison we can see on plots.

Summary

Summary

- CMS has more useful data than it can keep in the disk storage
  - When data is not present on disk it needs to be recalled from tape, which is a slow operation
- Project Objective
  - Explore how well Machine Learning algorithms can predict data popularity based on the current patterns and metadata of datasets already in use
    - Primary Dataset name, Acquisition Era and Data Tier etc
- Built a model using fully connected Neural Net
  - Popularity information was extracted from user crab jobs
  - Managed to achieve high Precision and Recall values using all historic data with feature embedding and the last month of data as a target
- Tested the model in a data cache application
  - Our model outperformed LRU when frequent model retraining is used
  - LRU model gives a similar performance and given its simplicity is a better choice for data caching application
- More work is needed to explore full potential of this approach

Thank you for your attention!