Machine Learning with Apache Spark

Example of a complete ML pipeline

Matteo Migliorini
University of Padua, CERN IT-DB

Use case

- Topology classification with deep learning to improve real time event selection at the LHC [https://arxiv.org/abs/1807.00083]

- Improve the purity of data samples selected in real time at the Large Hadron Collider

- Different data representation has been considered to train different multi-class classifiers:

    - Both raw data and high-level features are utilised

Machine Learning Pipeline

The goals of this work are:

- Produce an example of a ML pipeline using Spark
- Test the performances of Spark at each stage

Machine Learning Pipeline

The goals of this work are:

- Produce an example of a ML pipeline using Spark
- Test the performances of Spark at each stage

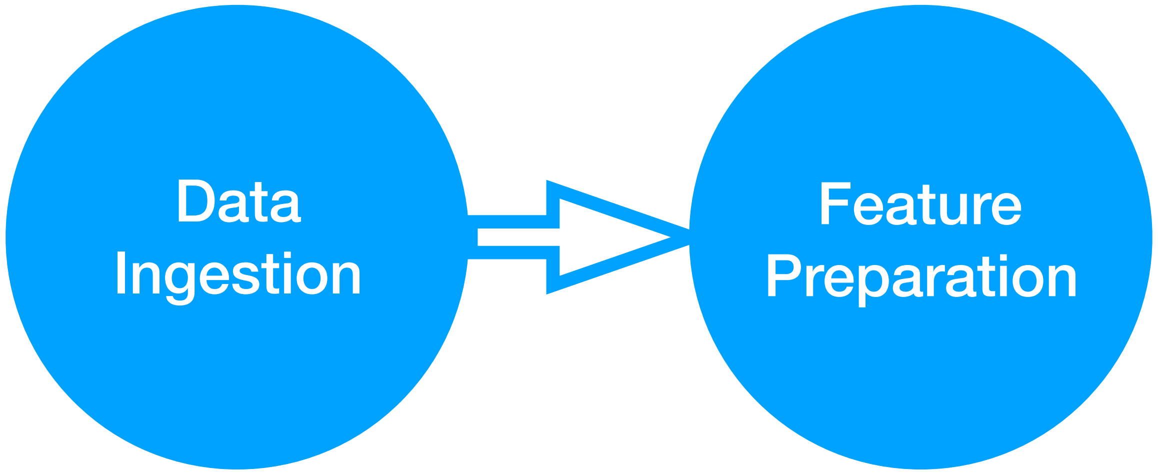
Data Ingestion

- Read Root Files from EOS
- Produce HLF and LLF datasets

Machine Learning Pipeline

The goals of this work are:

- Produce an example of a ML pipeline using Spark
- Test the performances of Spark at each stage



Data Ingestion

- Read Root Files from EOS
- Produce HLF and LLF datasets

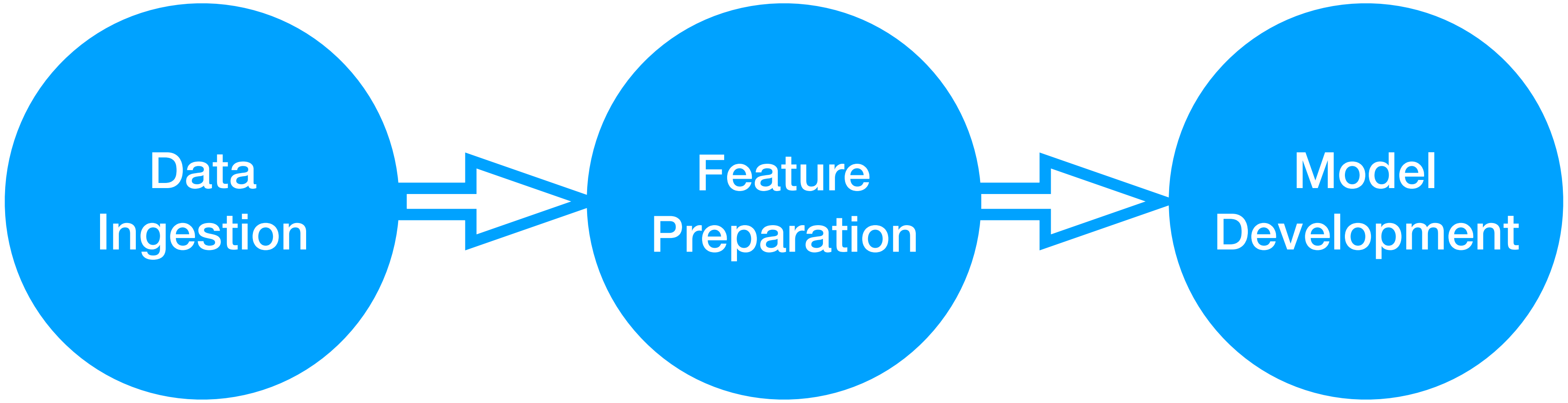
Feature Preparation

- Produce the input for each classifier

Machine Learning Pipeline

The goals of this work are:

- Produce an example of a ML pipeline using Spark
- Test the performances of Spark at each stage



- Read Root Files from EOS
- Produce HLF and LLF datasets

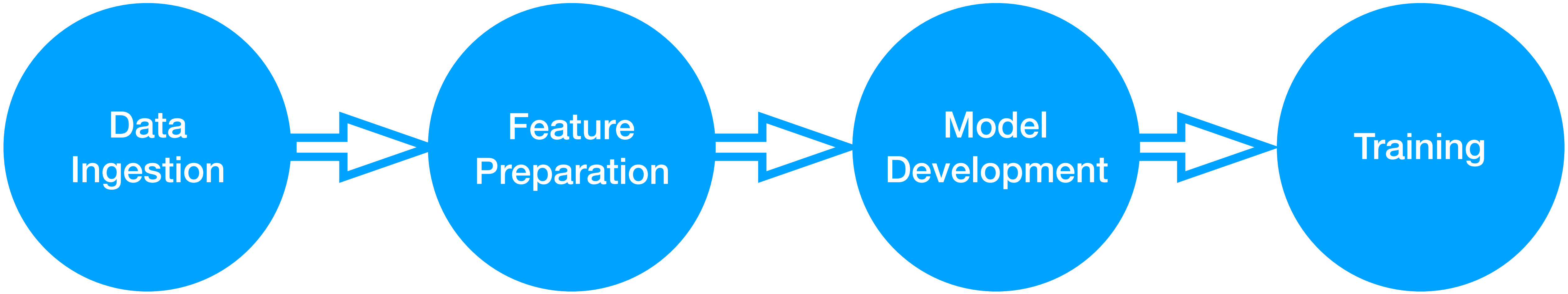
- Produce the input for each classifier

- Find the best model using Grid/Random search

Machine Learning Pipeline

The goals of this work are:

- Produce an example of a ML pipeline using Spark
- Test the performances of Spark at each stage



Data Ingestion

Feature Preparation

Model Development

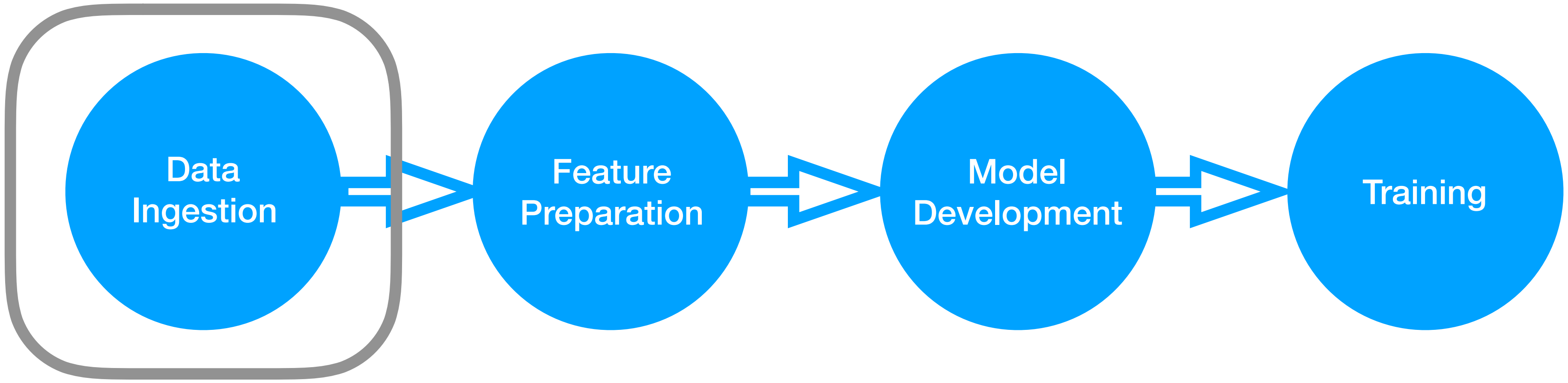
Training

- Read Root Files from EOS
- Produce HLF and LLF datasets

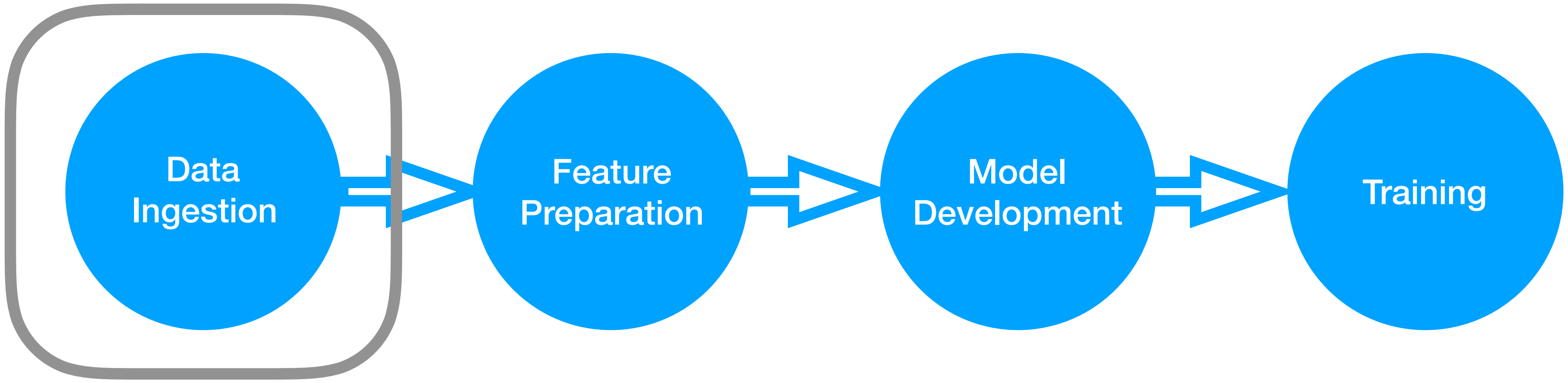
- Produce the input for each classifier

- Find the best model using Grid/Random search

- Train the best model on the entire dataset



Input Size: ~2 TBs



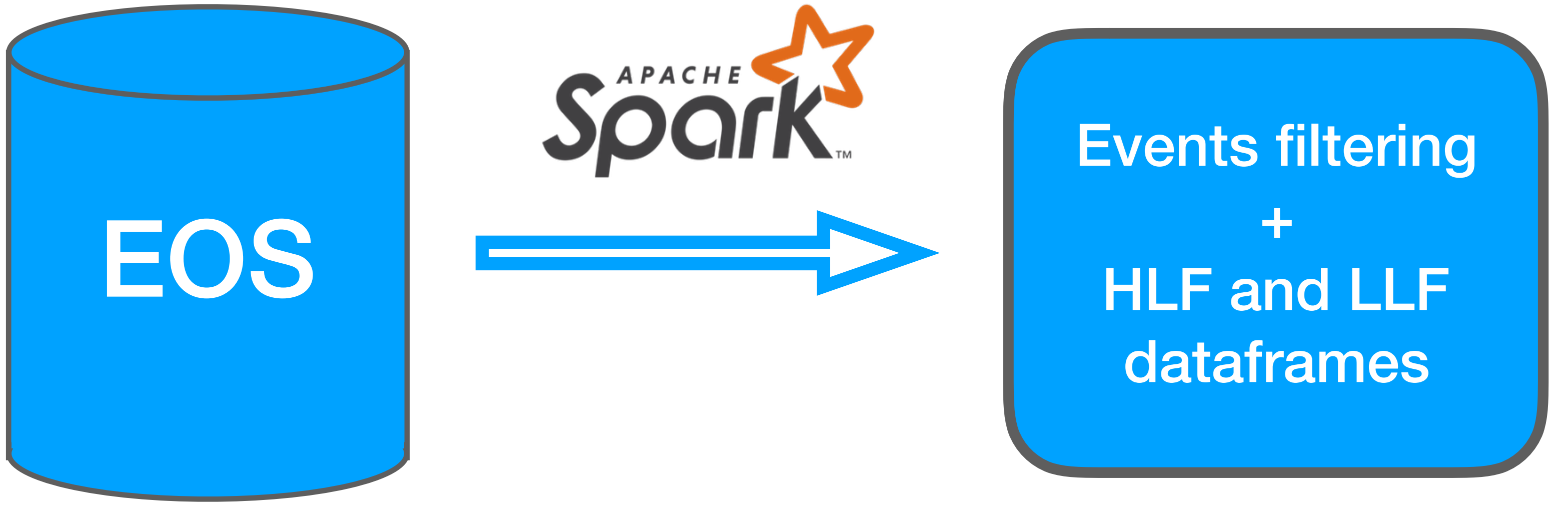
Data Ingestion

Feature Preparation

Model Development

Training

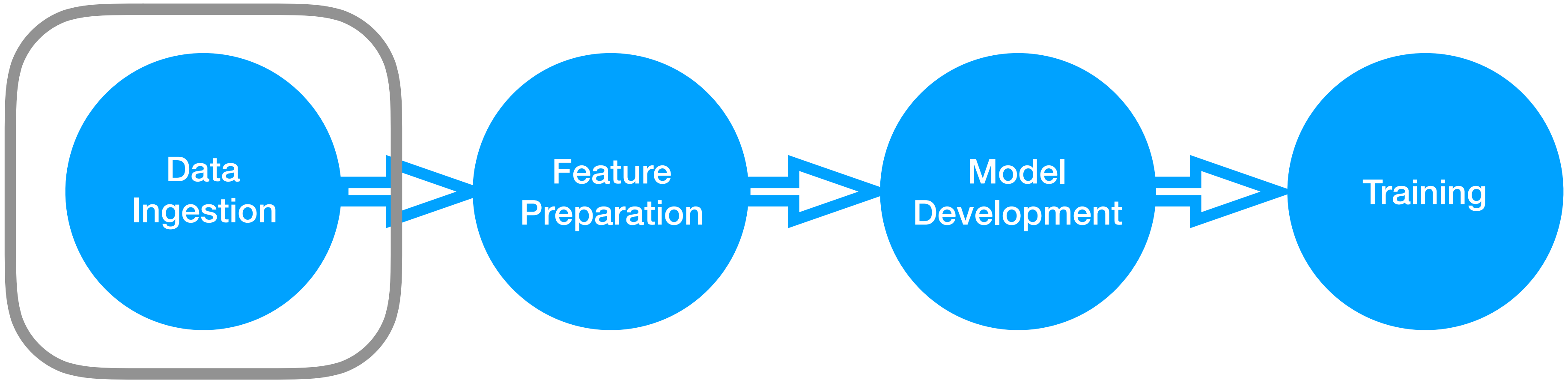
Input Size: ~2 TBs



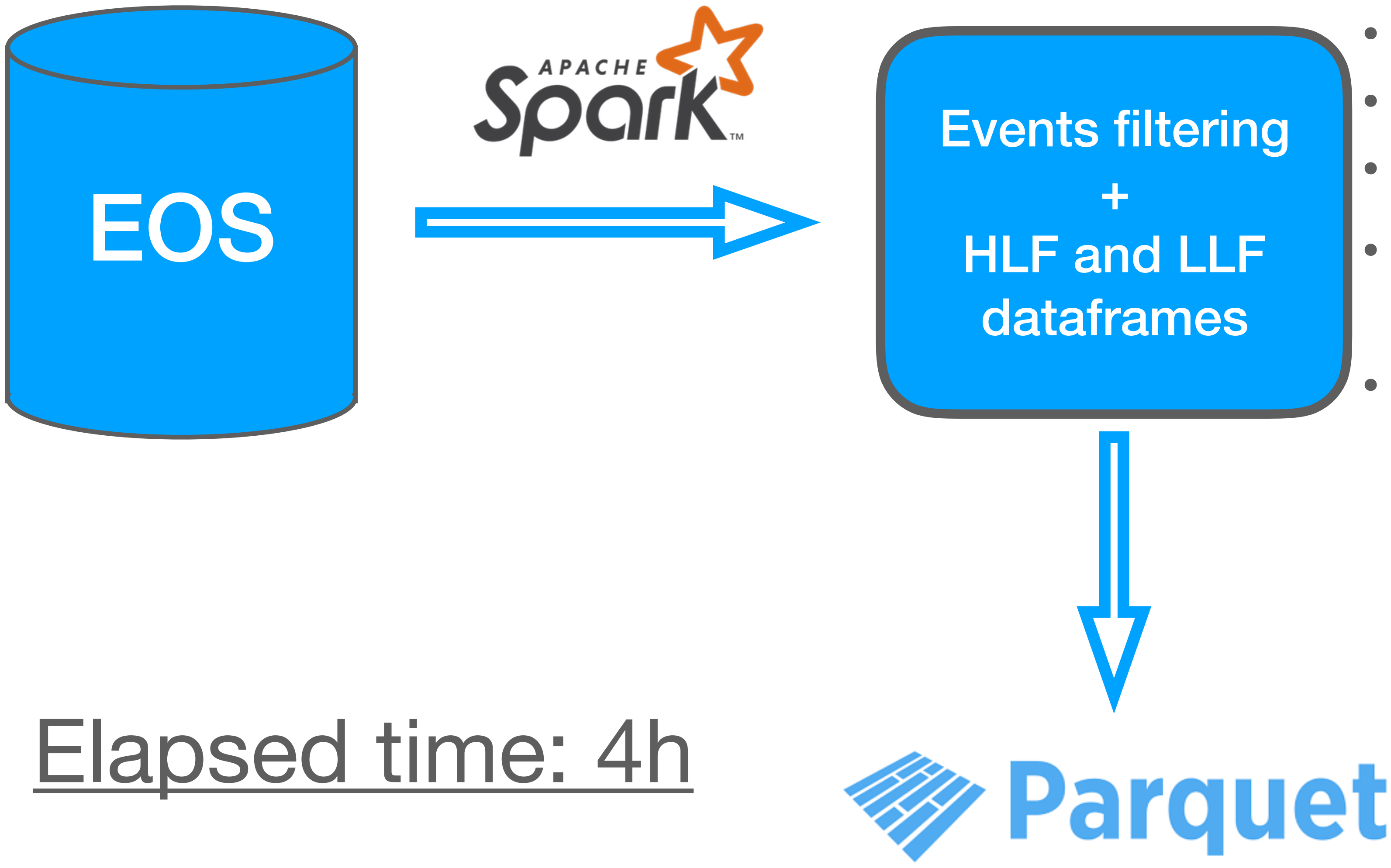
EOS

Events filtering
+
HLF and LLF dataframes

- Electron or Muon with p_T>23 GeV
- Particle-based isolation<0.45
- …
- LLF: list of 801 particles, each characterised by 19 features
- 14 HLF features



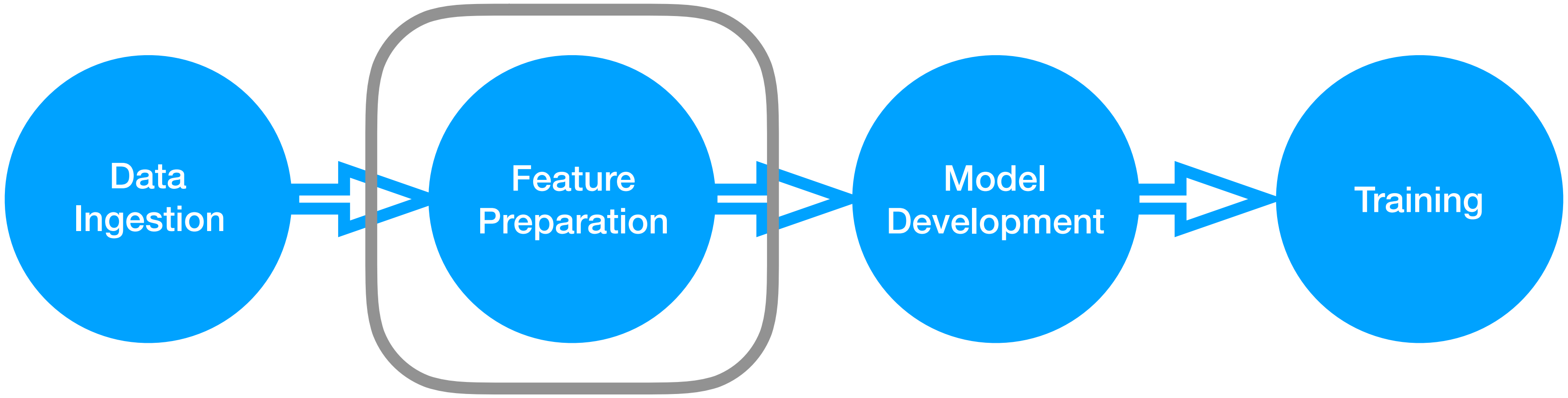
Input Size: ~2 TBs



- Electron or Muon with $p_T>23$ GeV
- Particle-based isolation<0.45
- …
- LLF: list of 801 particles, each characterised by 19 features
- 14 HLF features

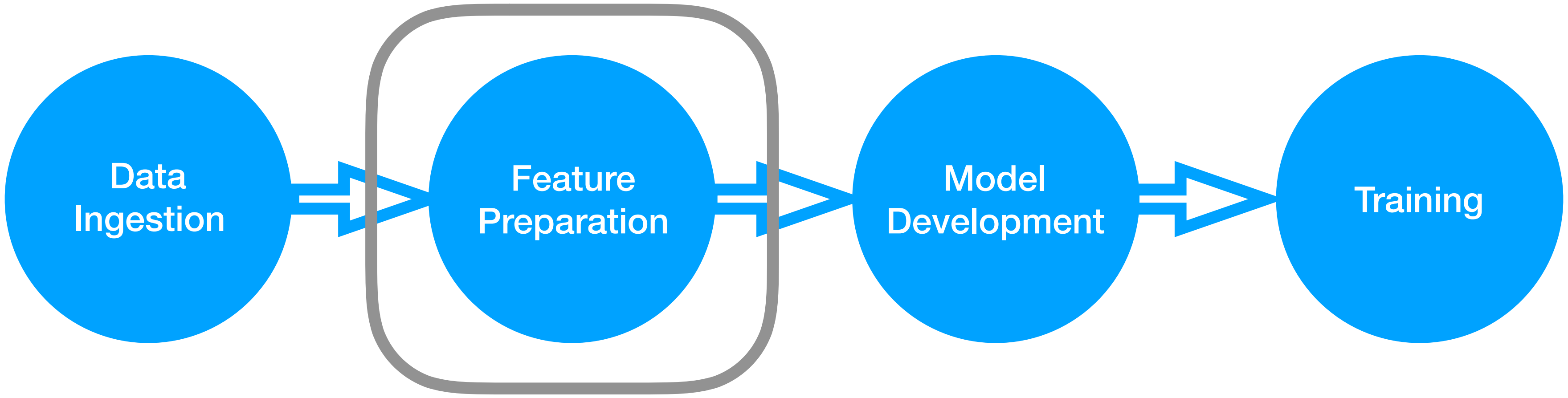
Elapsed time: 4h

Output Size: 750 GBs



Parquet

Start from the output of the previous stage

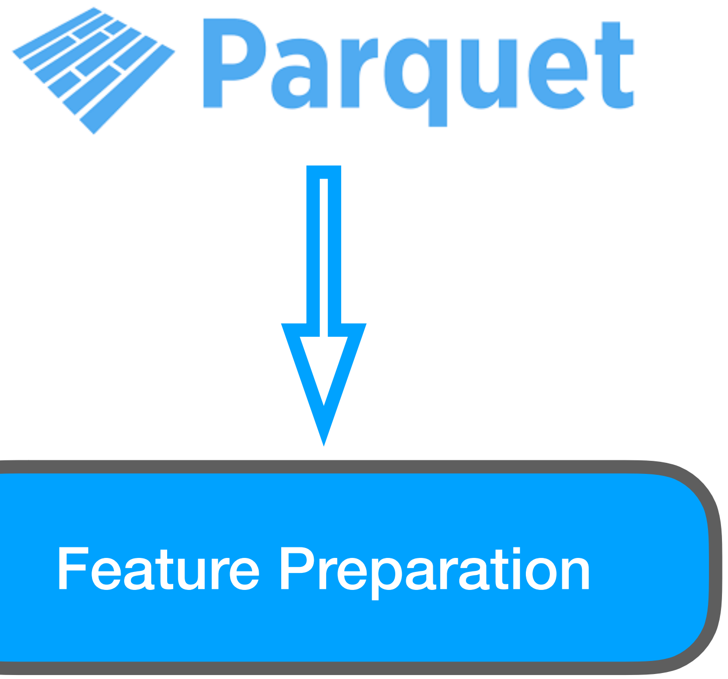


Data Ingestion

Feature Preparation

Model Development

Training

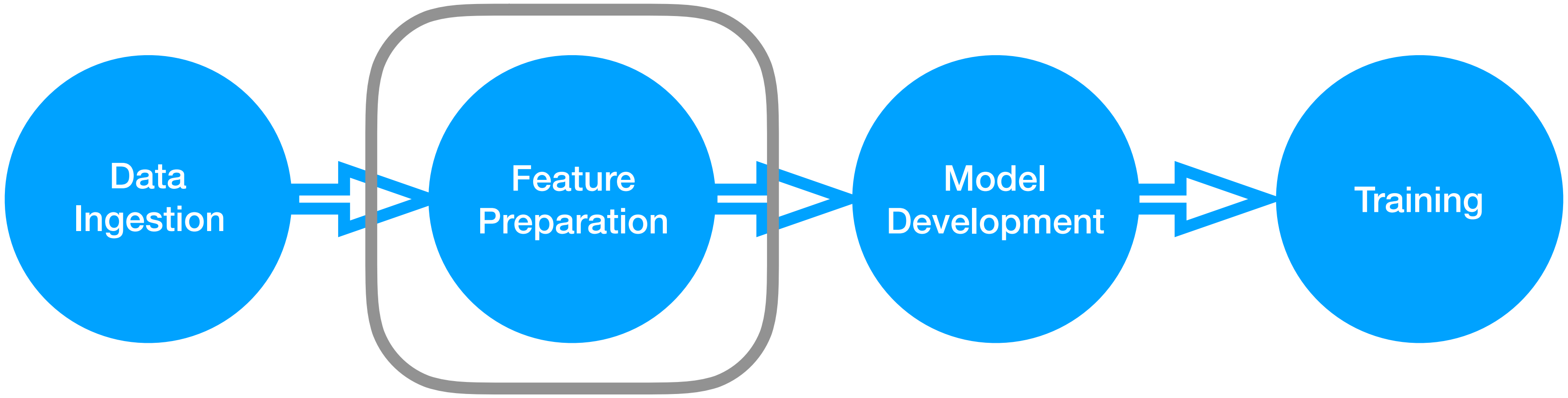


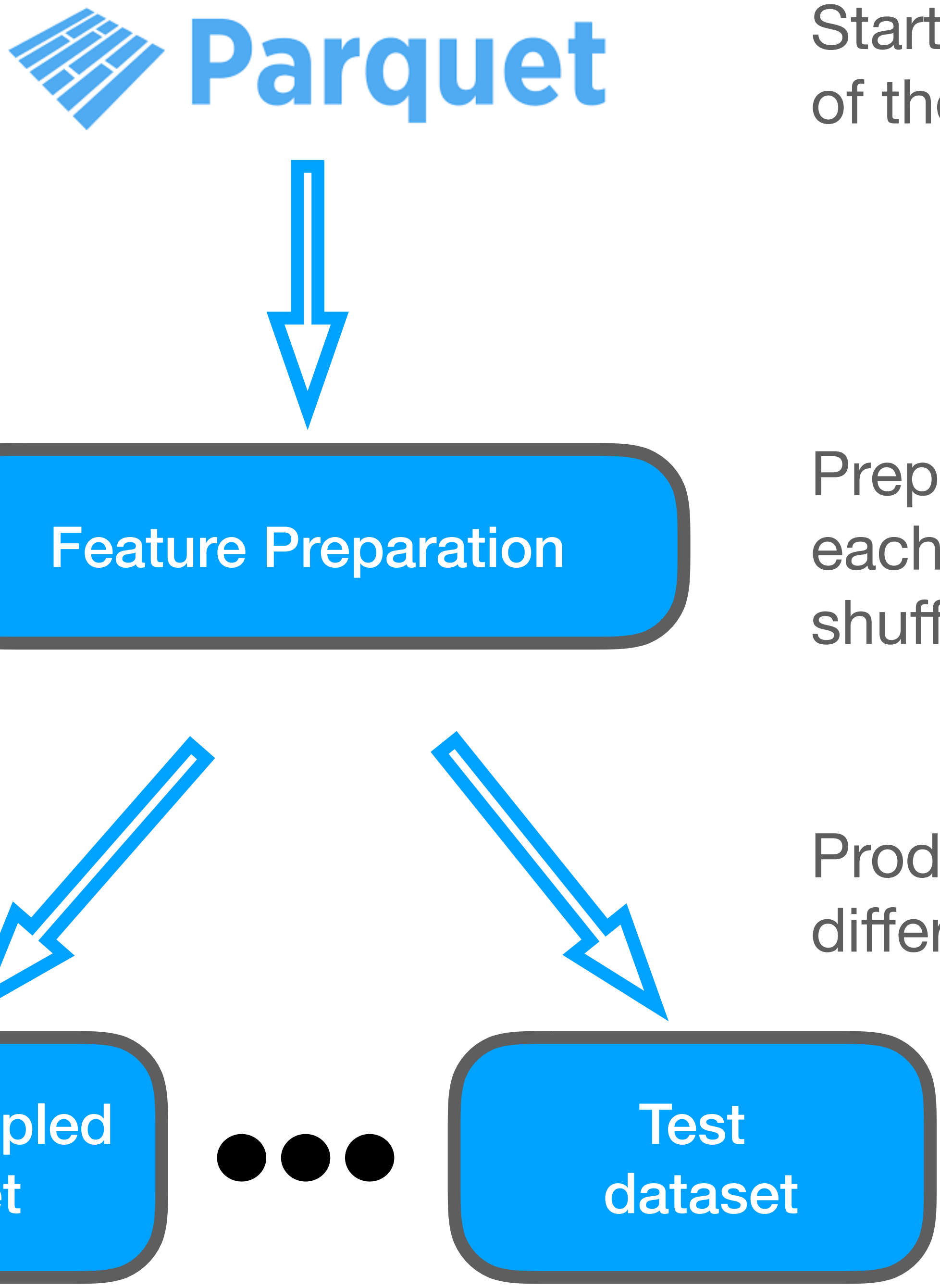
Parquet

Feature Preparation

Start from the output of the previous stage

Prepare the input for each classifier and shuffle the dataframe



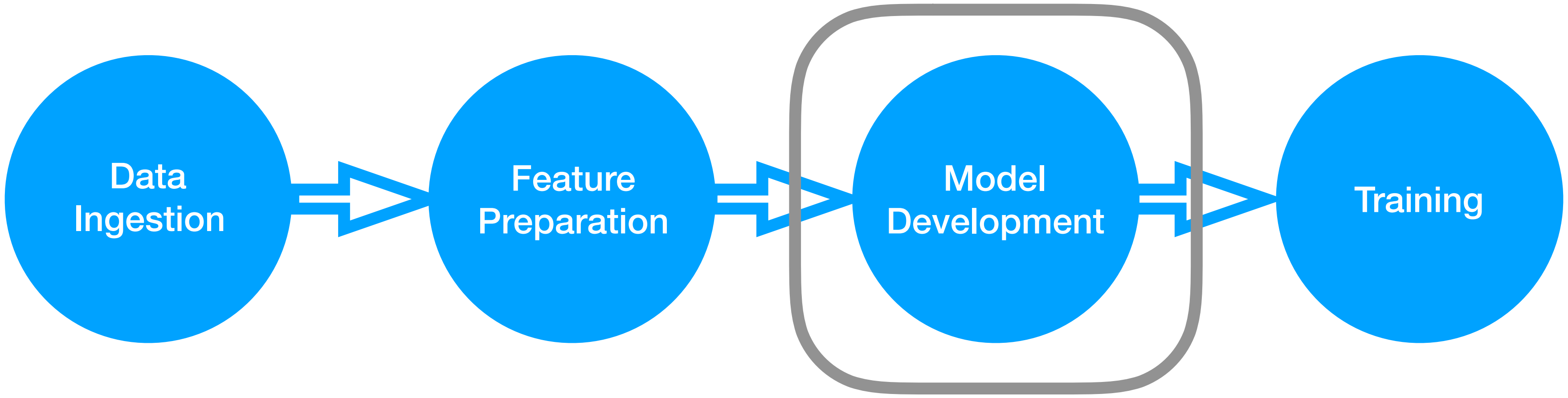


Start from the output of the previous stage

Elapsed time: 2h

Prepare the input for each classifier and shuffle the dataframe

Produce samples of different sizes



Data Ingestion

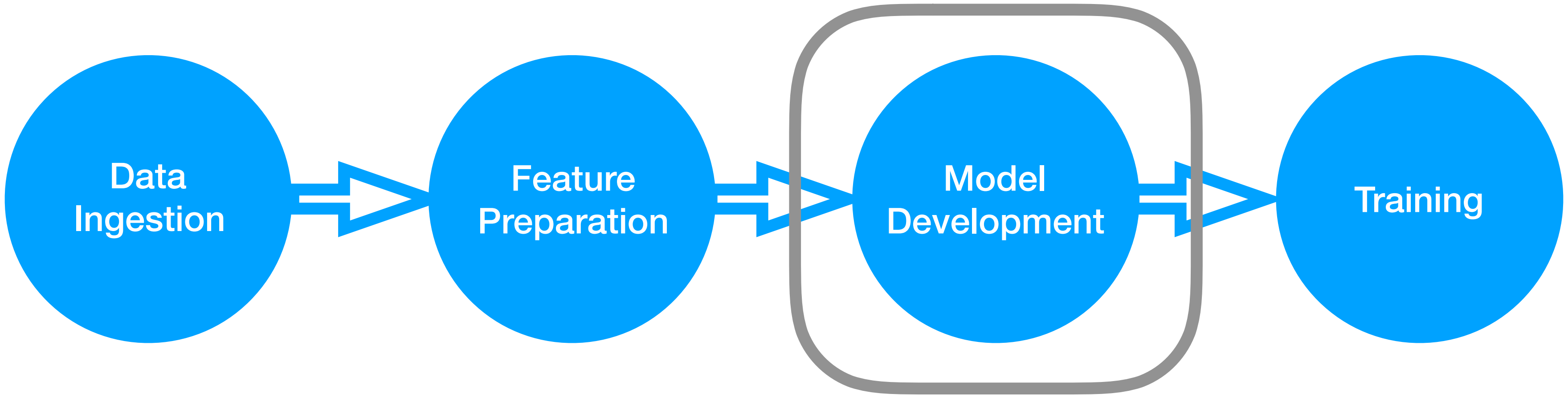
Feature Preparation

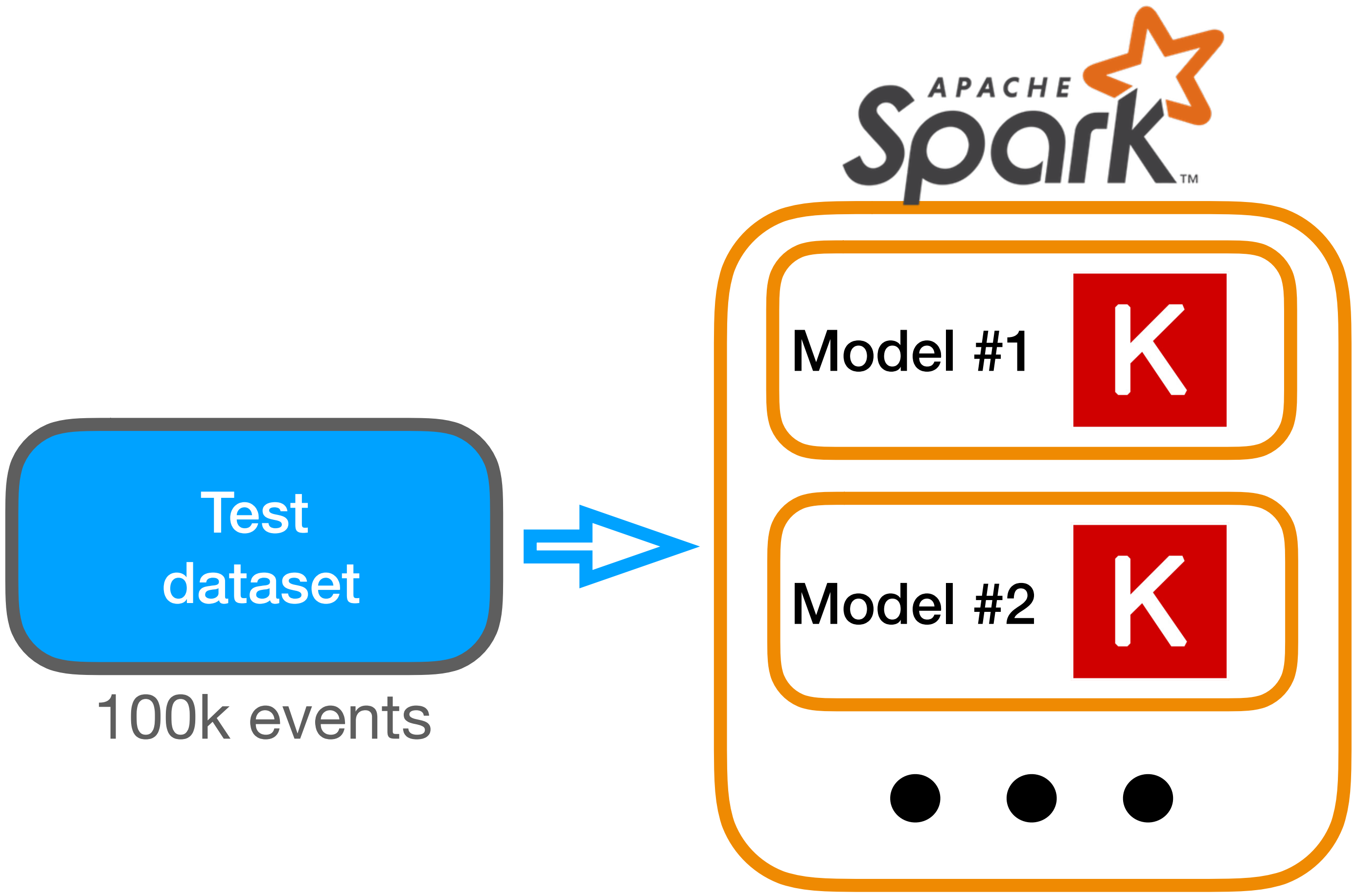
Model Development

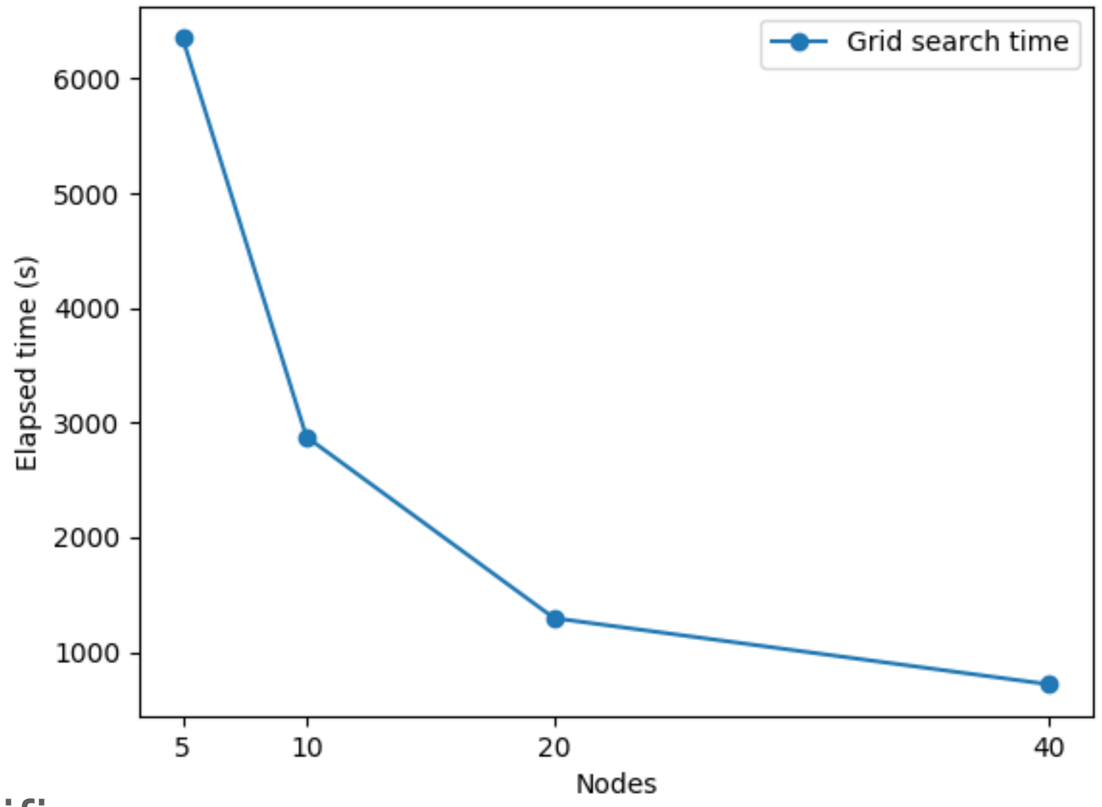
Training

Test dataset

100k events



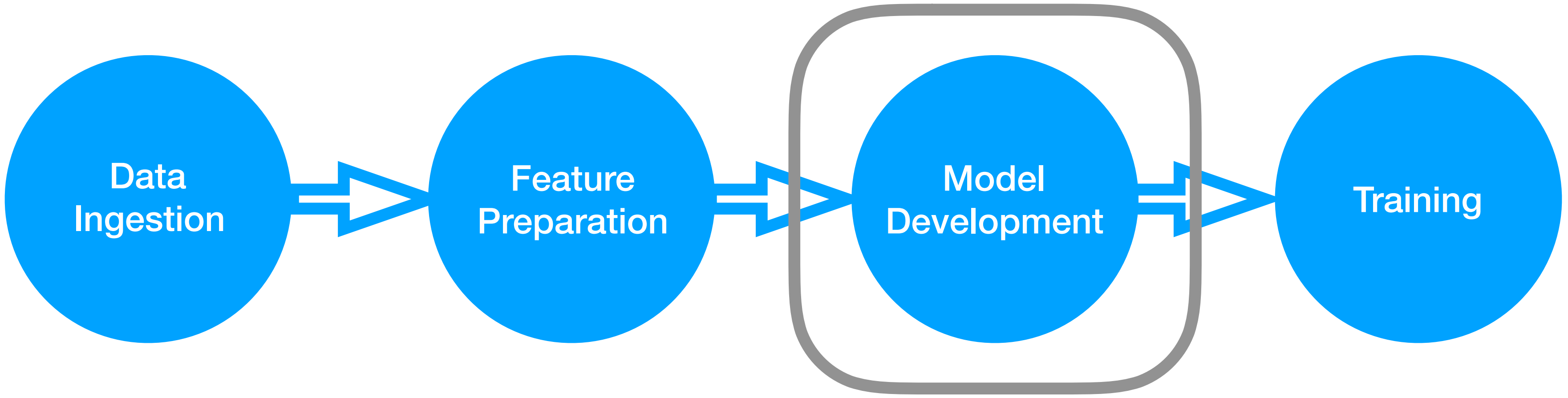




Tests made with the HLF classifier:

- Trained 162 different models changing topology and training parameters
- 3-fold cross validation

Each node (executor) has two cores

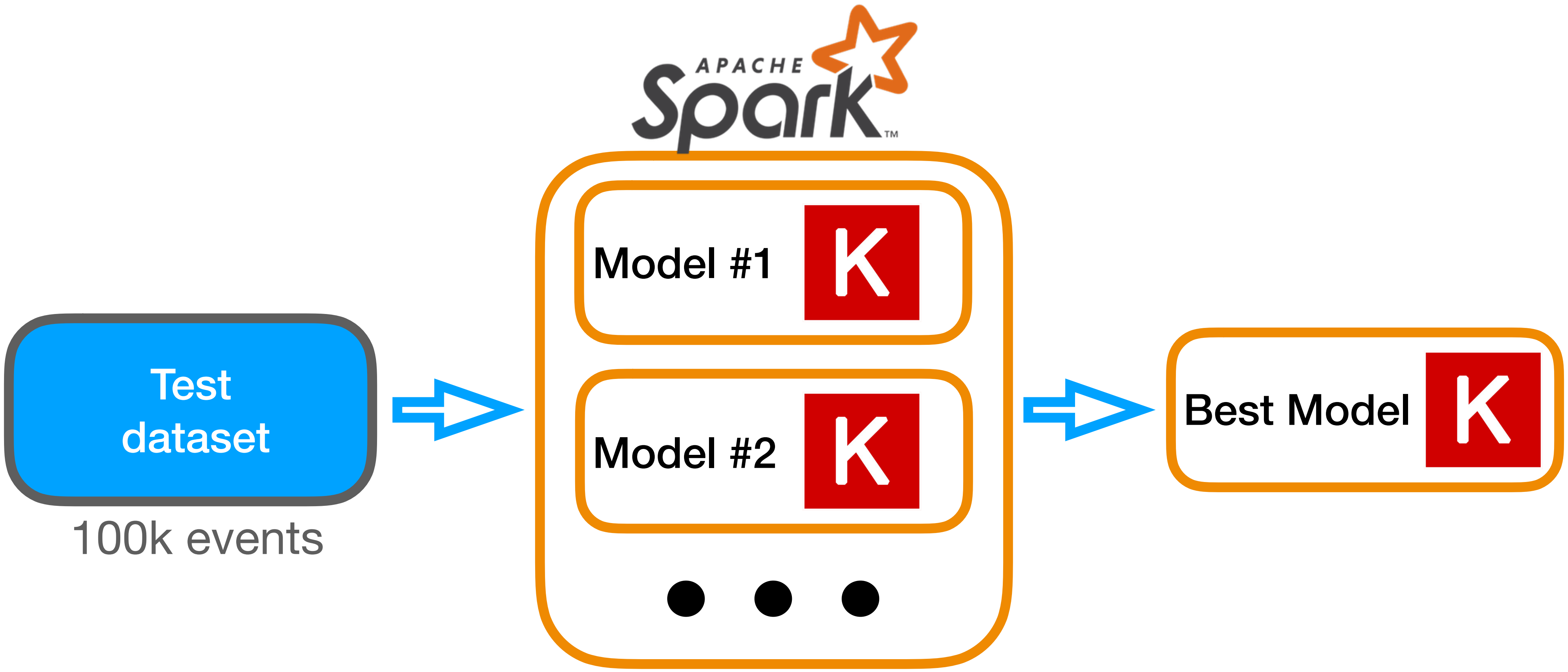


Data Ingestion

Feature Preparation

Model Development

Training



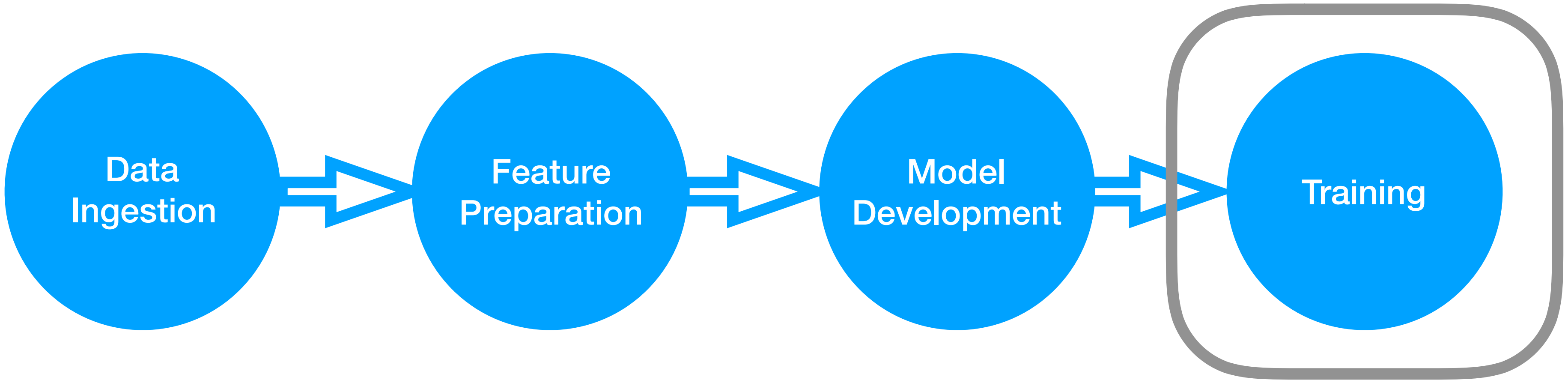
Test dataset

100k events

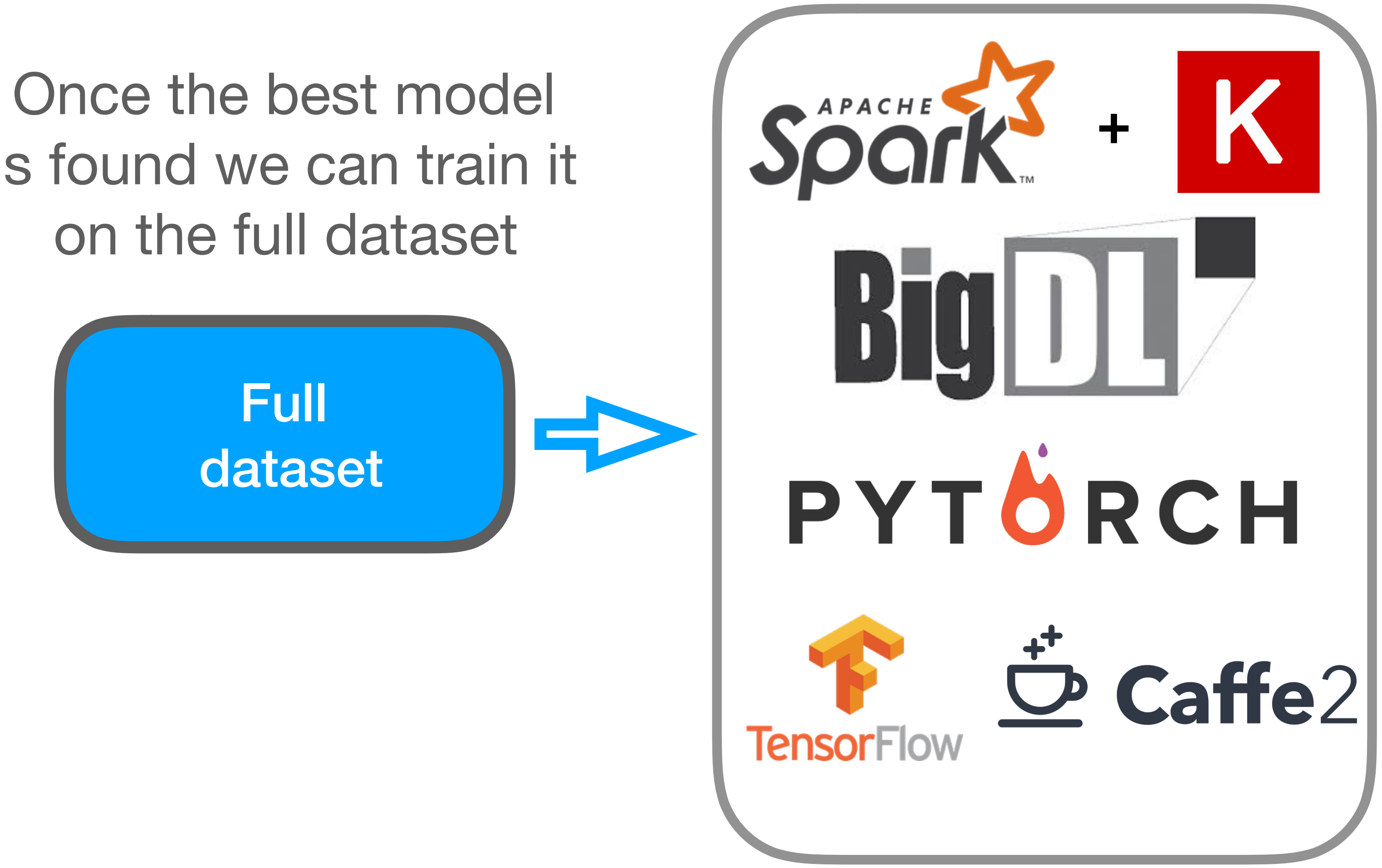
Model #1

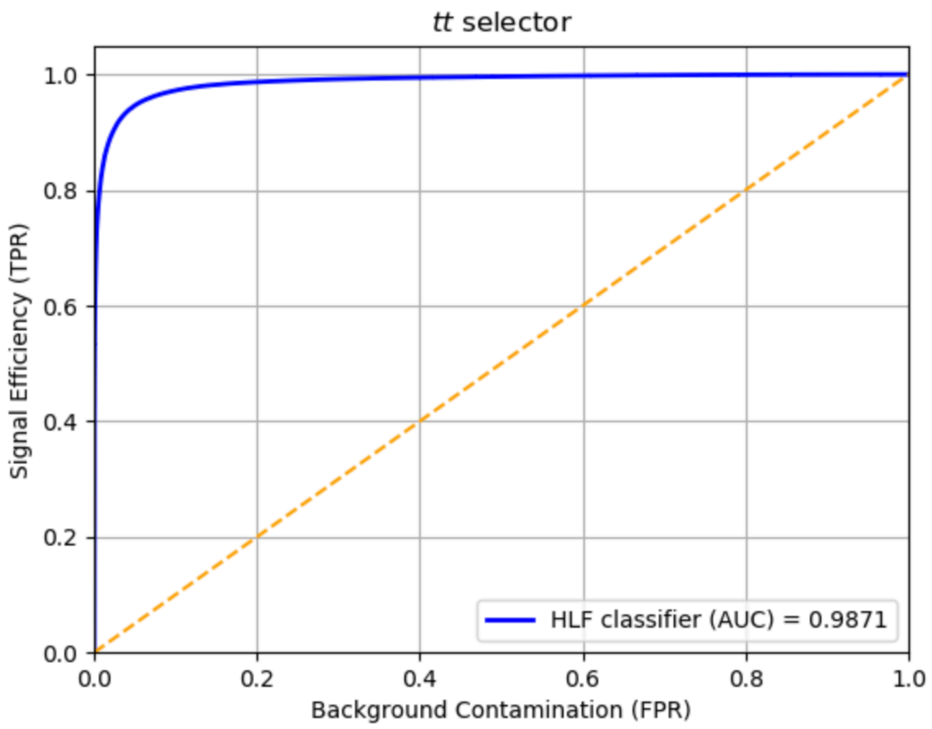
Model #2

Best Model



Once the best model is found we can train it on the full dataset

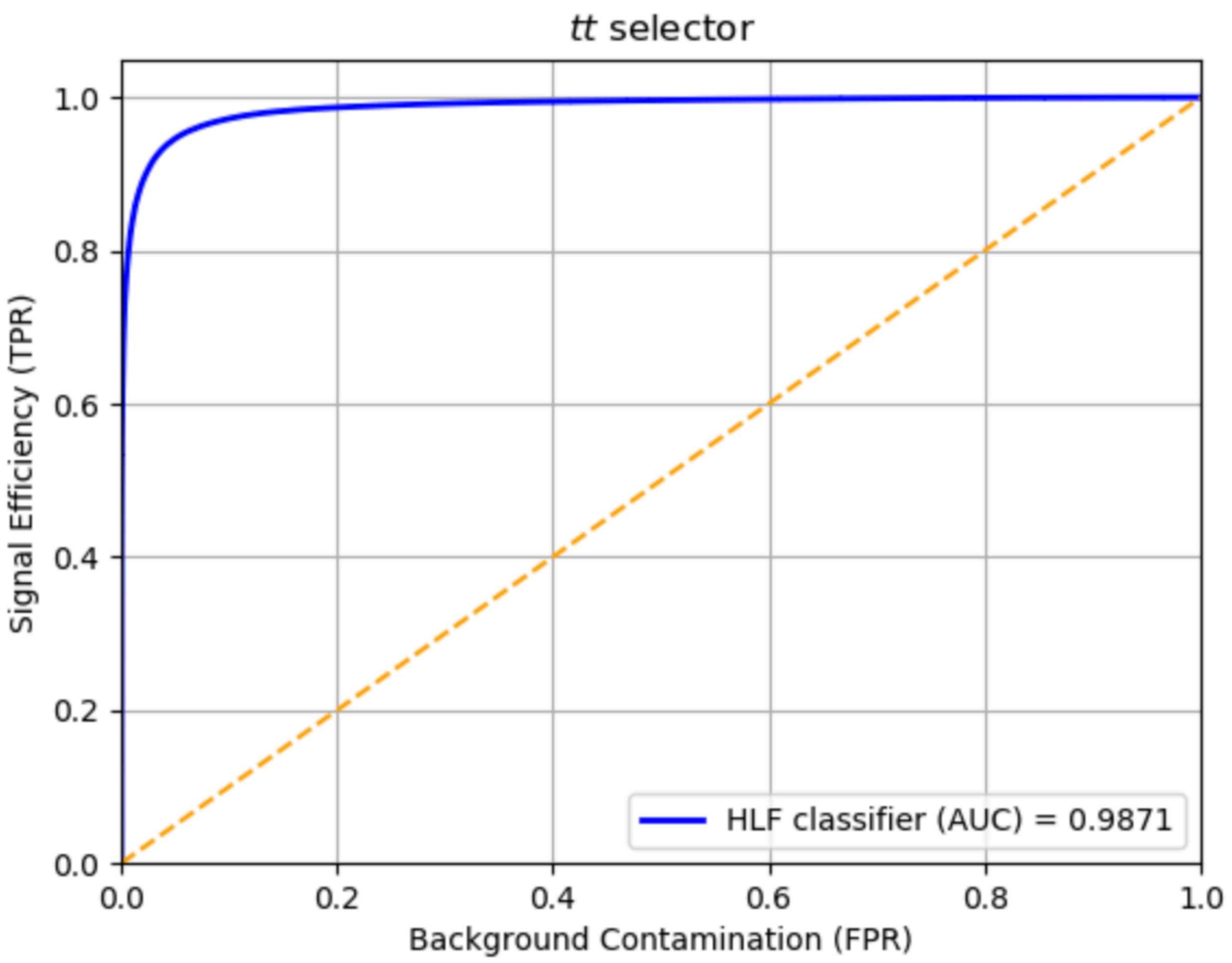


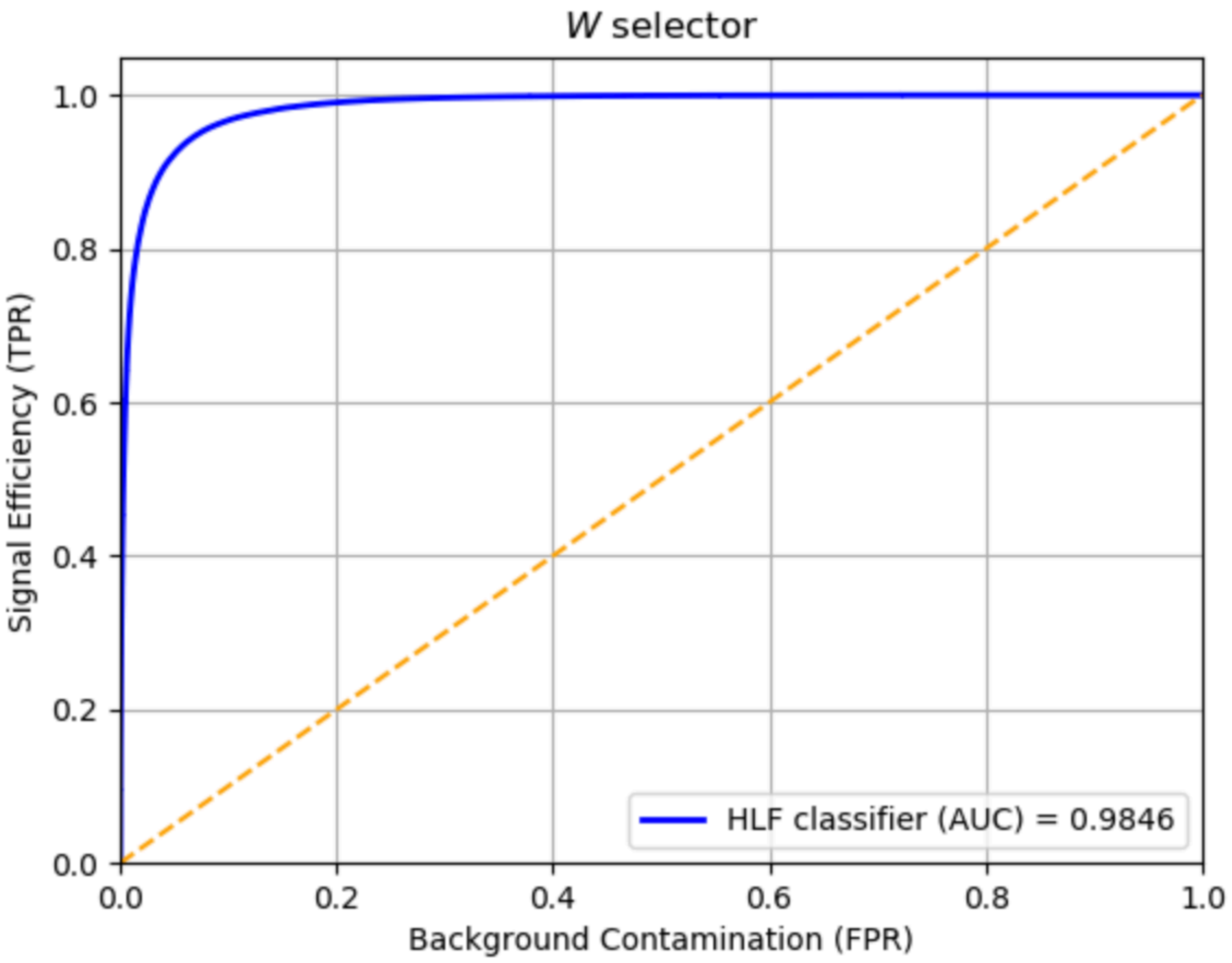


Different tools that can be used to train the best model

Result of the first tests

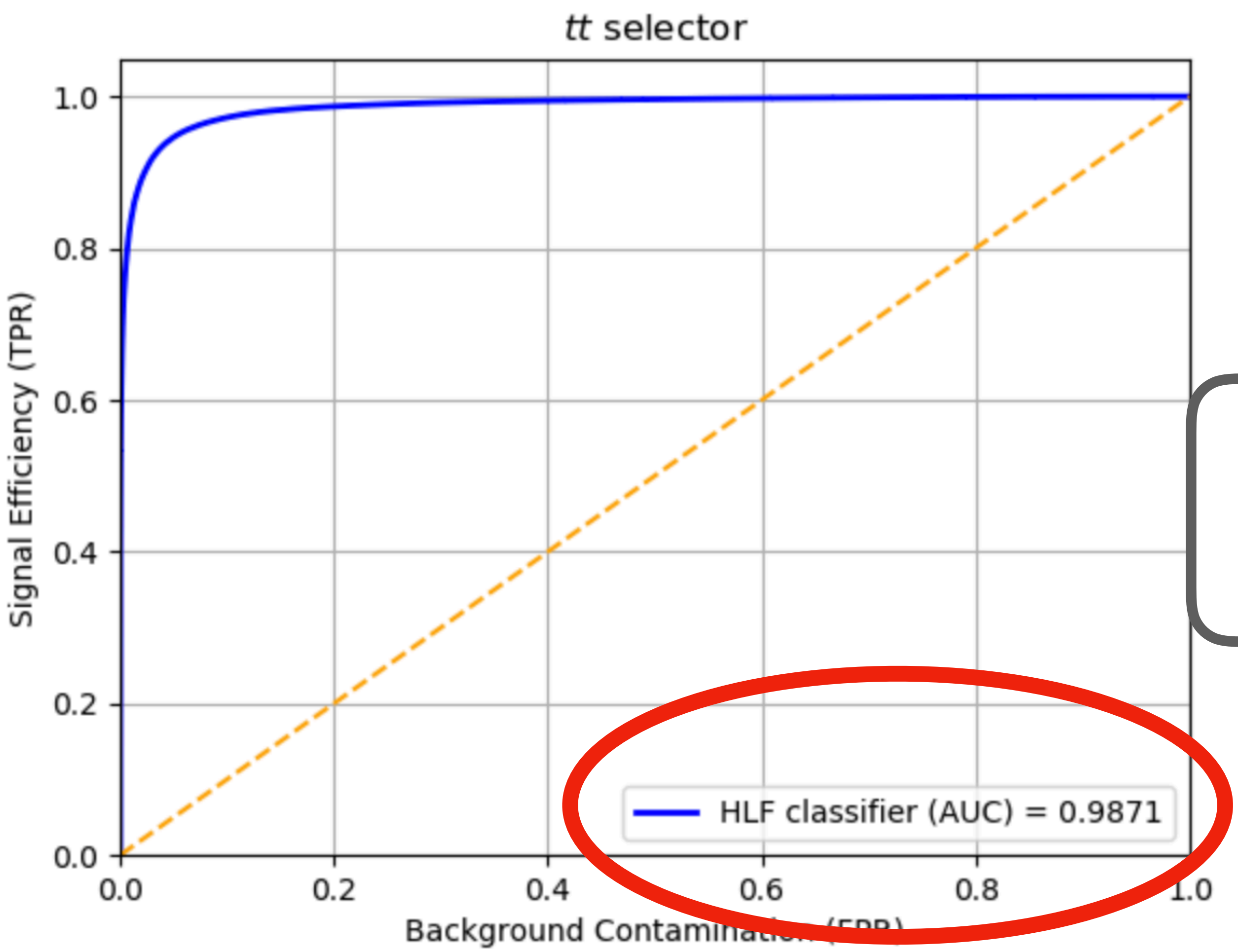
- Trained HLF classifier on the “Undersampled dataset” (Equal number of events for each class)
  - Training with ~4M events took 17 mins using dist-keras with 20 executors (2 cores each)

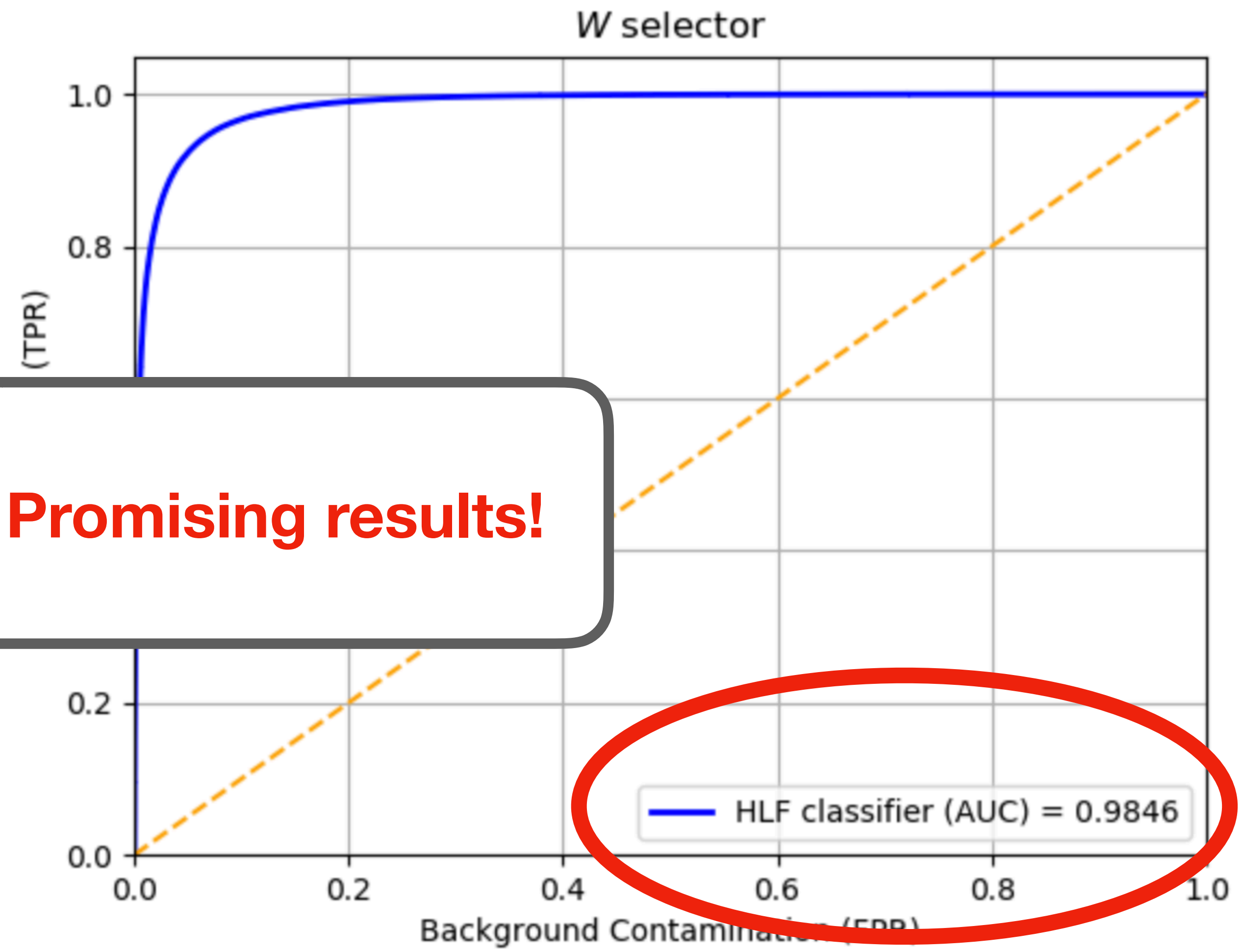




Result of the first tests

- Trained HLF classifier on the “Undersampled dataset” (Equal number of events for each class)
    - Training with ~4M events took 17 mins using dist-keras with 20 executors (2 cores each)





Promising results!

Comments

- The pipeline works!

- For the first three stages Spark works very well

    ✓ Data Ingestion, Feature Preparation, Model Development

    - There is still room for improvement: Ongoing studies on UDF performances

    ◉ Training at scale

- It is possible to deploy an end to end ML pipelines using Spark

Easy to use!

- Use industry standard tools
  - Python & Spark
  - Notebooks [https://github.com/Mmiglio/SparkML]
- Notebooks are a great tool!
  - They help keeping the code organised, embed documentation and graphs
  - Easy to share and collaborate

Further work

- Train the HLF classifier on a bigger sample and test different configurations (#Executor/#Cores)
- Test Particle-Sequence and Inclusive classifiers on a bigger dataset
    - Results obtained using a small sample are consistent with the ones from the paper
    - Train them on a bigger dataset
- Add the image classifier