esipap

European School of Instrumentation
in Particle & Astroparticle Physics

TMVA Lab

Dr. Karolos Potamianos
University of Oxford & CERN

November 25, 2021

Dr. Karolos POTAMIANOS

karolos.potamianos@cern.ch

@kpotamianos

http://linkedin.com/in/karolos

Education

- BSc./MSc. in Applied Physics
  @ Ecole Polytechnique, ULB, Belgium
- BSc. in Business Administration
  @ Solvay Business School, ULB, Belgium
- PhD in Particle Physics
  @ Purdue University, USA

Research

- 2012 – Present :: Researcher
  @ CERN
- 2012 – 2017 :: Postdoctoral Research Fellow
  @ Lawrence Berkeley National Laboratory (LBNL), USA
- 2017 – 2020 :: Research Fellow
  @ Deutsches Elektronen-Synchrotron (DESY), GER
- **2020 – Present :: Ernest Rutherford Fellow**
  @ University of Oxford, UK
- **2020 – Present :: gluoNNet**

Scope of the TMVA Lab

- Explore of the TMVA package for training machine learning algorithms
- Exercise these tools and get familiar with them
- Be a good starting point towards solving YOUR machine learning needs

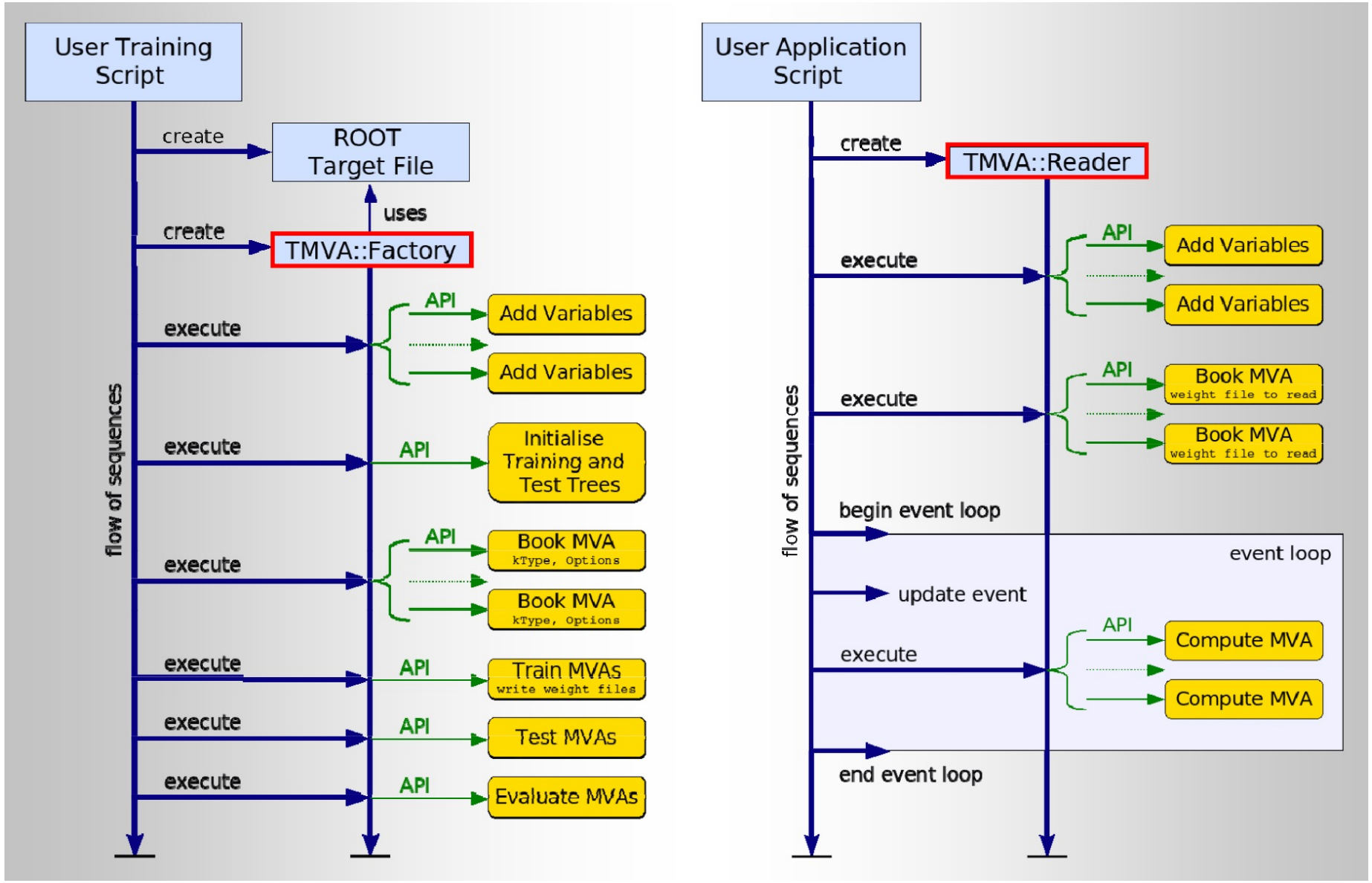
- There are many excellent resources online (the fact that machine learning is a subfield of computer science helps) to experiment with machine learning

- During the tutorials in this lab, we'll go over some (biased, non-exhaustive) applications of machine learning, to begin with
- **If you have a problem that you'd like to address, feel free to bring it forward and experiment with it during the tutorials**

TMVA :: Toolkit for Multivariate Data Analysis with ROOT

- **ROOT:** is the analysis framework used by most (HEP)-physicists
- **TMVA:** rather than implementing MVA techniques and making them available in ROOT:
  - Have one common platform / interface for all MVA classifiers
  - Have common data pre-processing capabilities
  - Train and test all classifiers on same data sample and evaluate consistently
  - Provide common analysis (ROOT scripts) and application framework
  - Provide access with and w/o ROOT, through macros, C++ executables or Python
- **Integral part of ROOT:** https://root.cern/manual/tmva/

TMVA :: Code Flow for *Training* and *Application* Phases



Source: A. Hoecker, Multivariate Data Analysis with TMVA

TMVA :: Simple Example for *Training*

```
void TMVAnalysis( )
{
   TFile* outputFile = TFile::Open( "TMVA.root", "RECREATE" );

   TMVA::Factory *factory = new TMVA::Factory( "MVAnalysis", outputFile,"!V");

   TFile *input = TFile::Open("tmva_example.root");

   factory->AddSignalTree        ( (TTree*)input->Get("TreeS"), 1.0 );
   factory->AddBackgroundTree ( (TTree*)input->Get("TreeB"), 1.0 );

   factory->AddVariable("var1+var2", 'F');
   factory->AddVariable("var1-var2", 'F');
   factory->AddVariable("var3", 'F');
   factory->AddVariable("var4", 'F');

   factory->PrepareTrainingAndTestTree("", "NSigTrain=3000:NBkgTrain=3000:SplitMode=Random:!V" );

   factory->BookMethod( TMVA::Types::kLikelihood, "Likelihood",
                        "!V:!TransformOutput:Spline=2:NSmooth=5:NAvEvtPerBin=50" );

   factory->BookMethod( TMVA::Types::kMLP, "MLP", "!V:NCycles=200:HiddenLayers=N+1,N:TestRate=5" );

   factory->TrainAllMethods();
   factory->TestAllMethods();
   factory->EvaluateAllMethods();

   outputFile->Close();
   delete factory;
}
```

← create *Factory*

← give training/test trees

← register input variables

← select MVA methods

← train, test and evaluate

→ TMVA tutorial

Source: A. Hoecker, Multivariate Data Analysis with TMVA

TMVA :: Simple Example for *Application*

```
void TMVApplication( )
{
   TMVA::Reader *reader = new TMVA::Reader("!Color");

   Float_t var1, var2, var3, var4;
   reader->AddVariable( "var1+var2", &var1 );
   reader->AddVariable( "var1-var2",  &var2 );
   reader->AddVariable( "var3", &var3 );
   reader->AddVariable( "var4", &var4 );

   reader->BookMVA( "MLP classifier",  "weights/MVAnalysis_MLP.weights.txt" );

   TFile *input = TFile::Open("tmva_example.root");
   TTree* theTree = (TTree*)input->Get("TreeS");

   // … set branch addresses for user TTree
   for (Long64_t ievt=3000; ievt<theTree->GetEntries();ievt++) {
      theTree->GetEntry(ievt);

      var1 = userVar1 + userVar2;
      var2 = userVar1 - userVar2;
      var3 = userVar3;
      var4 = userVar4;

      Double_t out =  reader->EvaluateMVA(  "MLP classifier"  );

      // do something with it …
   }
   delete reader;
}
```

← create *Reader*

← register the variables

← book classifier(s)

← prepare event loop

← compute input variables

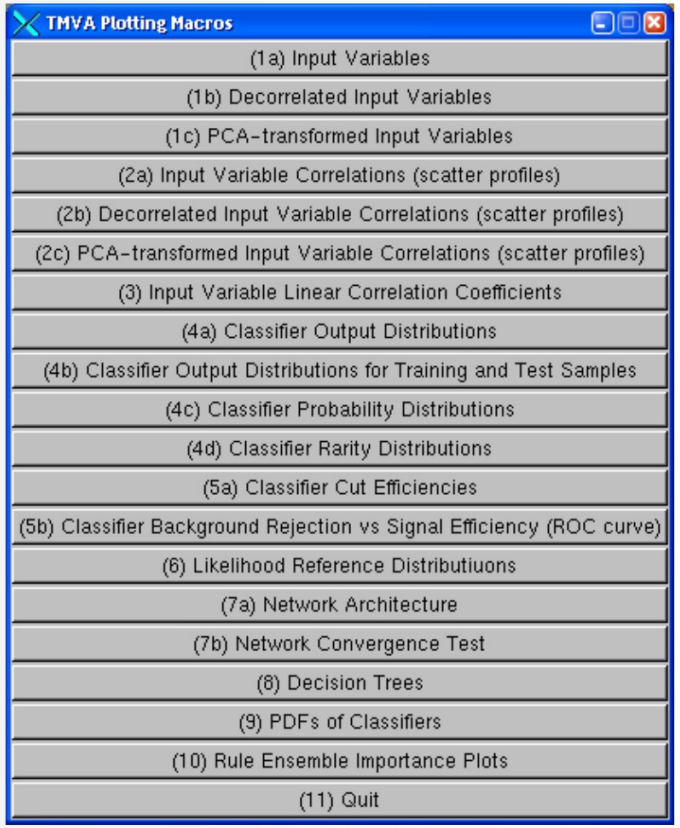
← calculate classifier output

→ TMVA tutorial

Source: A. Hoecker, Multivariate Data Analysis with TMVA

TMVA :: MVA *Evaluation* Framework

- **TMVA** is not only a collection of classifiers, but an MVA framework
  - Provides evaluation scrips (through GUI)



Plot all signal (S) and background (B) input variables with and without pre-processing

Correlation scatters and linear coefficients for S & B

Classifier outputs (S & B) for test and training samples (spot overtraining)

Classifier *Rarity* distribution

Classifier significance with optimal cuts

B rejection versus S efficiency

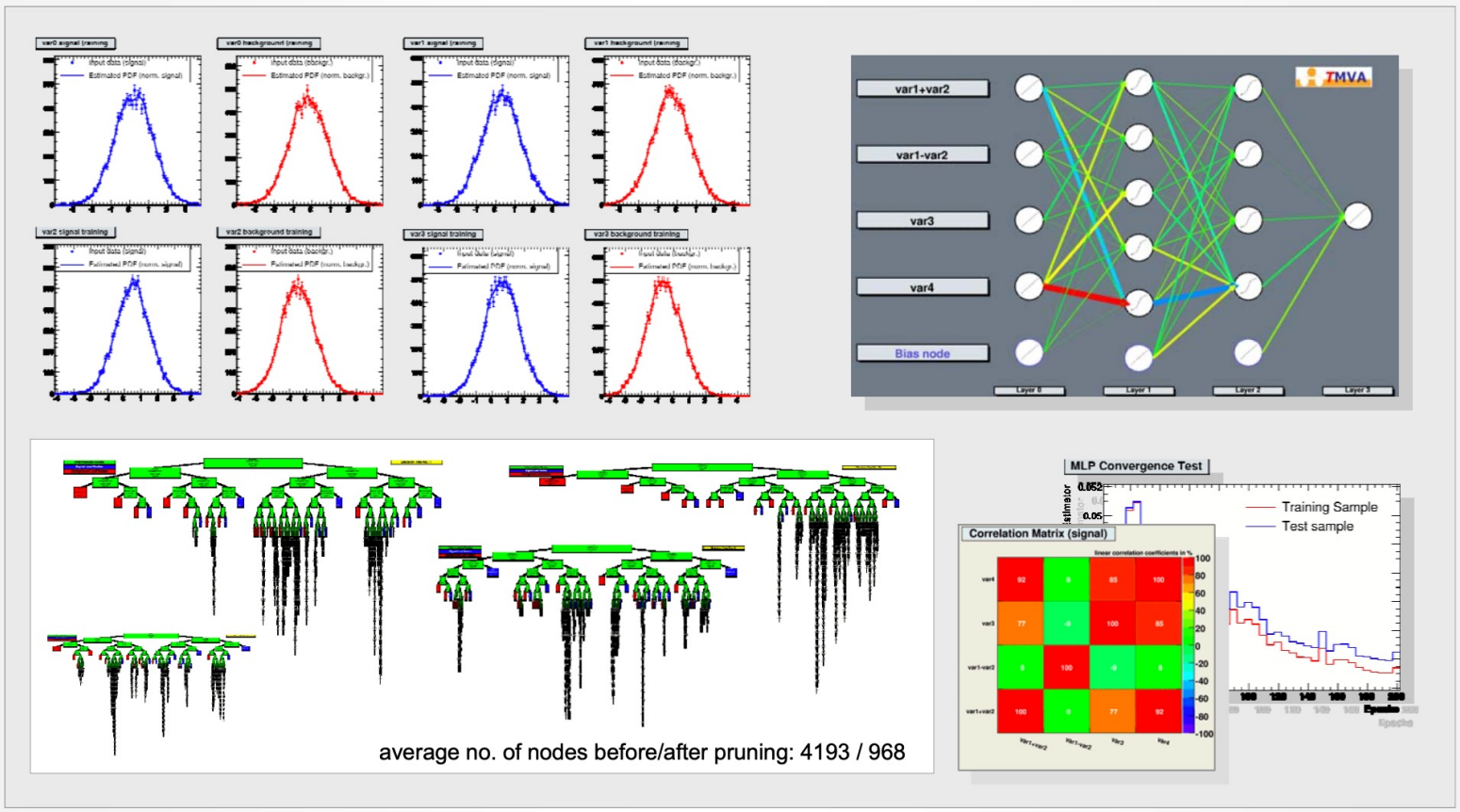
Classifier-specific plots:
- Likelihood reference distributions
- Classifier PDFs (for probability output and Rarity)
- Network architecture, weights and convergence
- Rule Fitting analysis plots

- Visualise decision trees

esipap
European School of Instrumentation in Particle & Astroparticle Physics

TMVA :: MVA *Evaluation* Framework

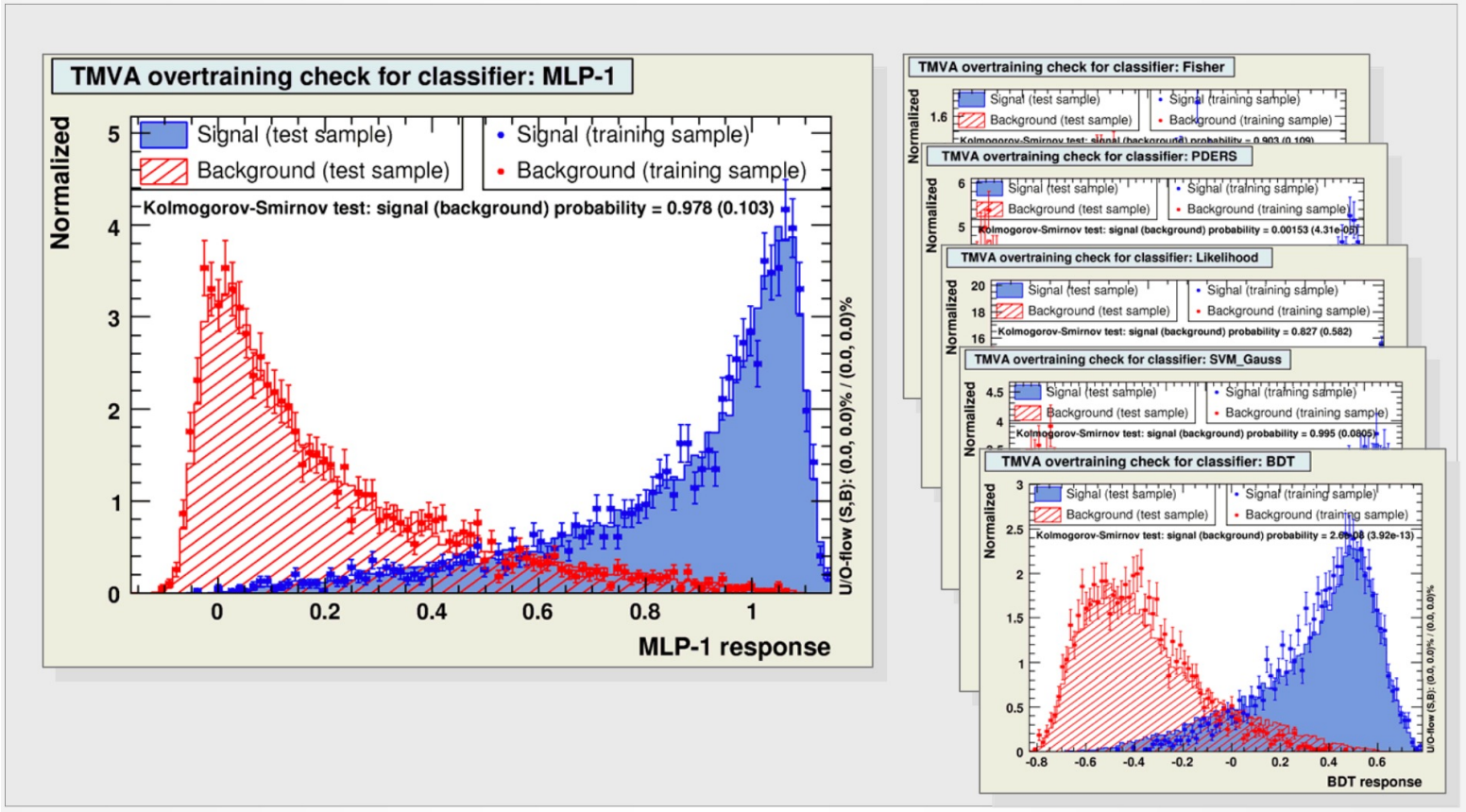
- **TMVA** is not only a collection of classifiers, but an MVA framework
  - Evaluation of Classifiers



esipap
European School of Instrumentation in Particle & Astroparticle Physics

TMVA :: MVA *Evaluation* Framework

- **TMVA** is not only a collection of classifiers, but an MVA framework
  - Evaluation of Classifier outout distributions for test and training samples

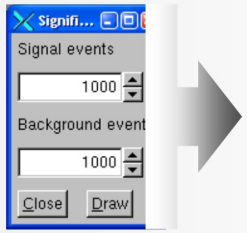


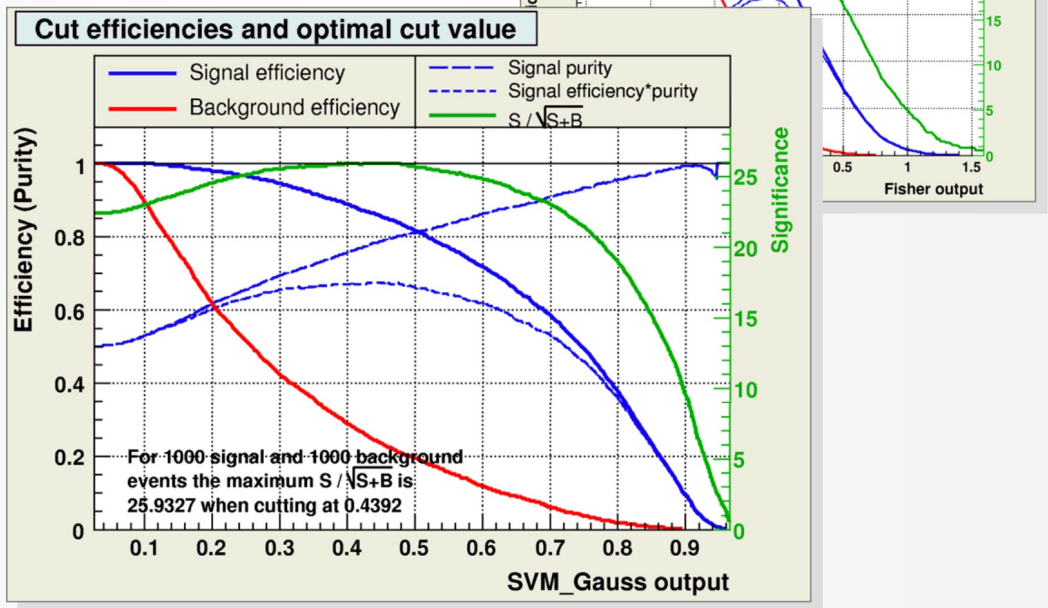
TMVA :: MVA *Evaluation* Framework

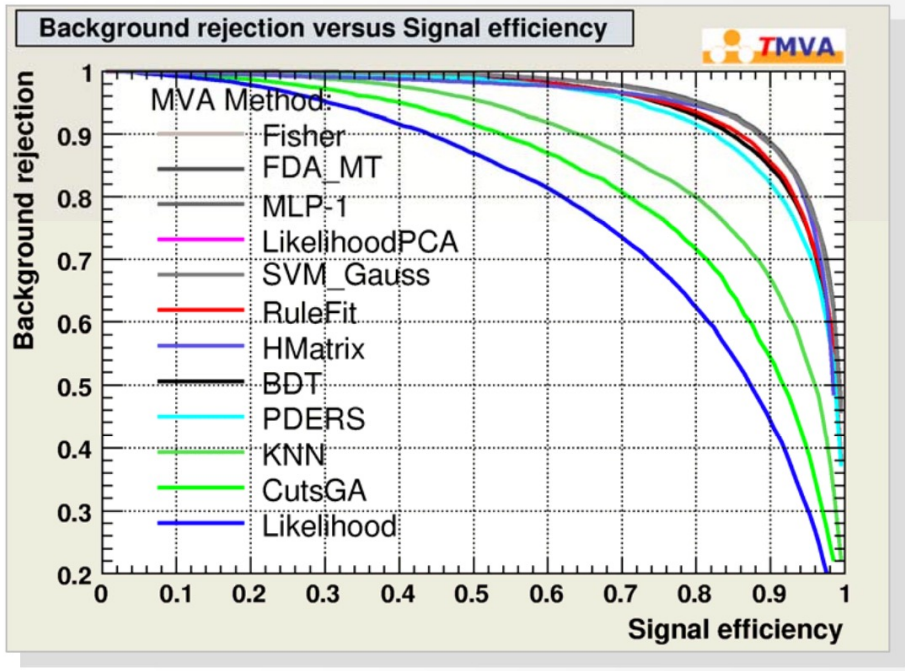
- **TMVA** is not only a collection of classifiers, but an MVA framework
  - Comparison and optimal cut for Classifiers

Optimal cut for each classifiers …

Determine the optimal cut (working point) on a classifier output







esipap
European School of Instrumentation in Particle & Astroparticle Physics

Before we proceed …

- Feel free to work in teams if you desire, ask questions at any time, and help answer each other's questions! [doesn't really apply with COVID but we can nevertheless make it happen if you so desire]

- Machine Learning is a rapidly growing field (esp. in deep learning as we'll see later): nobody knows everything, and there are new ideas each week!

- Luckily, there's a very active community of people online, with lots of tutorials, blog pots, videos, etc.

- **ML/DL is also about sharing and collaborating with others!**

esipap
European School of Instrumentation in Particle & Astroparticle Physics

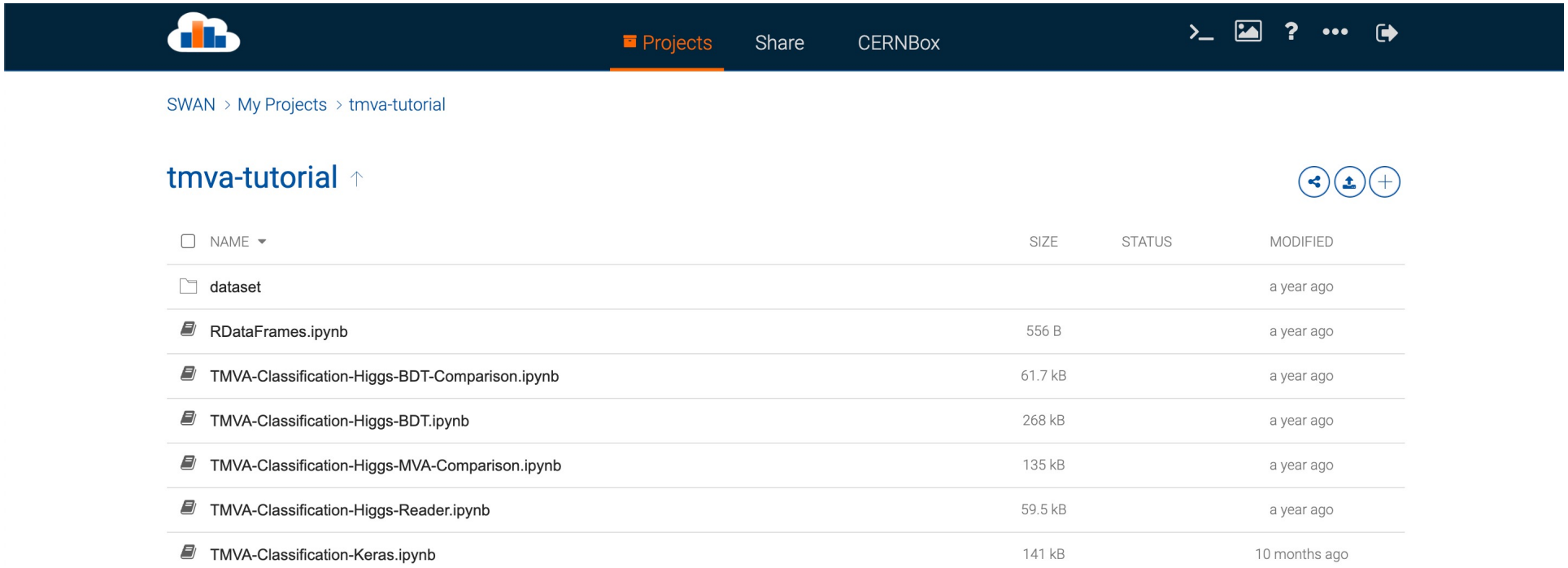
TMVA Lab :: Tutorial

- Slides and instructions are available here:
  - https://gitlab.cern.ch/karolos-potamianos/esipap/tmva-tutorial
- The repository contains some exercises you can run
  - We will go over them during the tutorial and the Q&A
- Additional tutorial material is available here:
  - https://root.cern/doc/master/group__tutorial__tmva.html

- There are several ways to run (and do your work in the future):
  - **Run on SWAN**: https://swan.cern.ch/ (requires CERN account)
  - **Run on Docker**: https://hub.docker.com/r/rootproject/root
  - **Run on lxplus** (requires CERN account)
  - **Install ROOT (>= 6.22) and Python3** locally or on the remote machine you're working on, manually or via a package manager (e.g. Anaconda)

TMVA Lab :: Running with SWAN

- CERN offers a **S**ervice for **W**eb based **AN**alysis (SWAN): https://swan.cern.ch
  - Note that you need a CERN account to use it
- Once your session is started, and you've chosen the configuration (default is OK), you can use the "Download from git" button (see right) and point to
  - https://gitlab.cern.ch/karolos-potamianos/esipap/tmva-tutorial



esipap
European School of Instrumentation in Particle & Astroparticle Physics

TMVA Lab :: Running with Docker

- **Docker** is a system to manage and run containers. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.
- Very flexible and allows runing from any machine with an internet connection and Docker installed; to get Docker, check out https://docs.docker.com/get-docker/

- You can run the latest version of ROOT and Python3 by running:
  - **`docker run -it rootproject/root:6.24.06-centos7 /bin/bash`**
  - You'll need a terminal application to execute the line above
  - The container will be downloaded automatically if necessary

TMVA Lab :: Running with Docker

- You might want to link the local folder inside the container so that your data is saved to your local computer once the container is closed (else the data will be lost)
- Once you **git clone** the course material, you can start running it

```
ESIPAP % git clone https://gitlab.cern.ch/karolos-potamianos/esipap/tmva-tutorial

ESIPAP % docker run -it --user $(id -u) -p 8888:8888/tcp -v "${PWD}/tmva-tutorial":/tmva-tutorial rootproject/root:6.24.06-centos7 /bin/bash
```

TMVA Lab :: Running with Docker :: Jupyter Notebook

- You can also spawn the Jupyter Notebook sever with root using (in the container):
  - root --notebook
  - You might need additional setup for this (depending on your configuration)

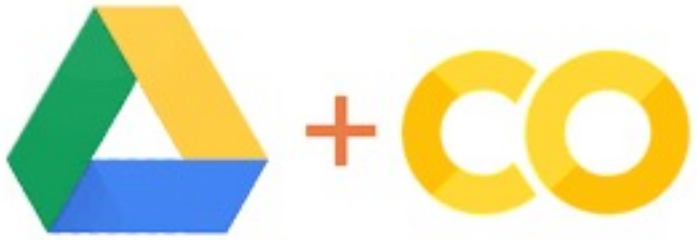
- You can also enable graphics in your container, following the instructions at:
  - https://hub.docker.com/r/rootproject/root
  - These instructions differ depending on your platform (Linux, Windows or macOS)

Google Colaboratory :: https://colab.research.google.com

- Colab is a Python development environment that runs in the browser using Google Cloud (for free)
  - Very convenient for sharing code and work together
  - Uses the concept of Python Notebooks (like Jupyter)

- You can also import code from GitHub!

- Note: you don't need to be an expert on Python to use it for ML (though it helps to know it) as most of the syntax is straightforward

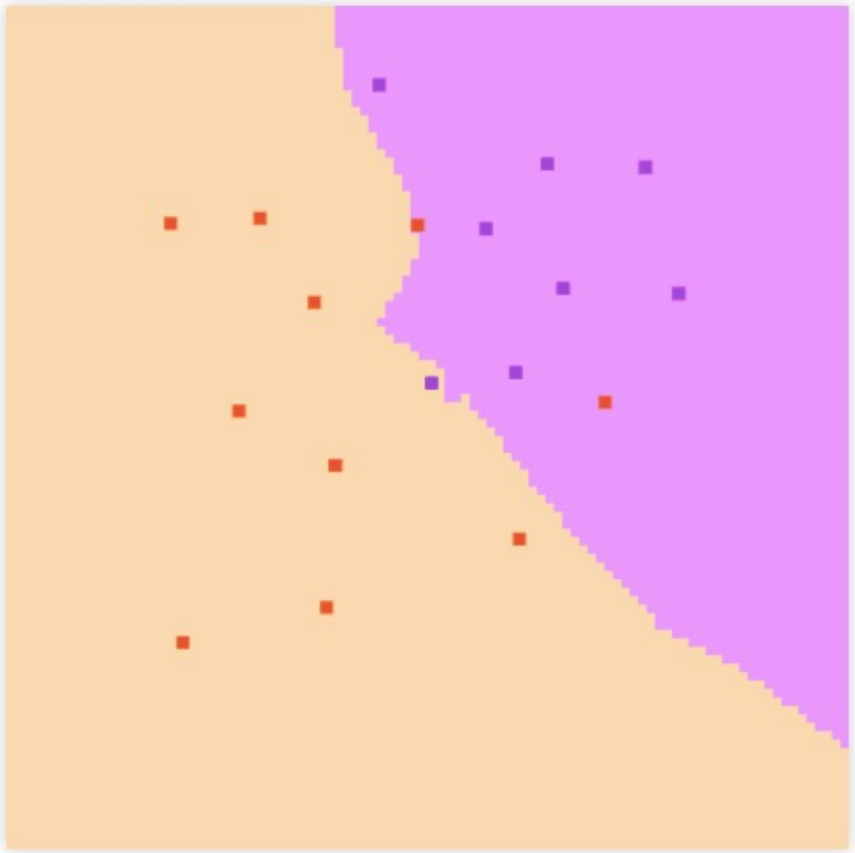


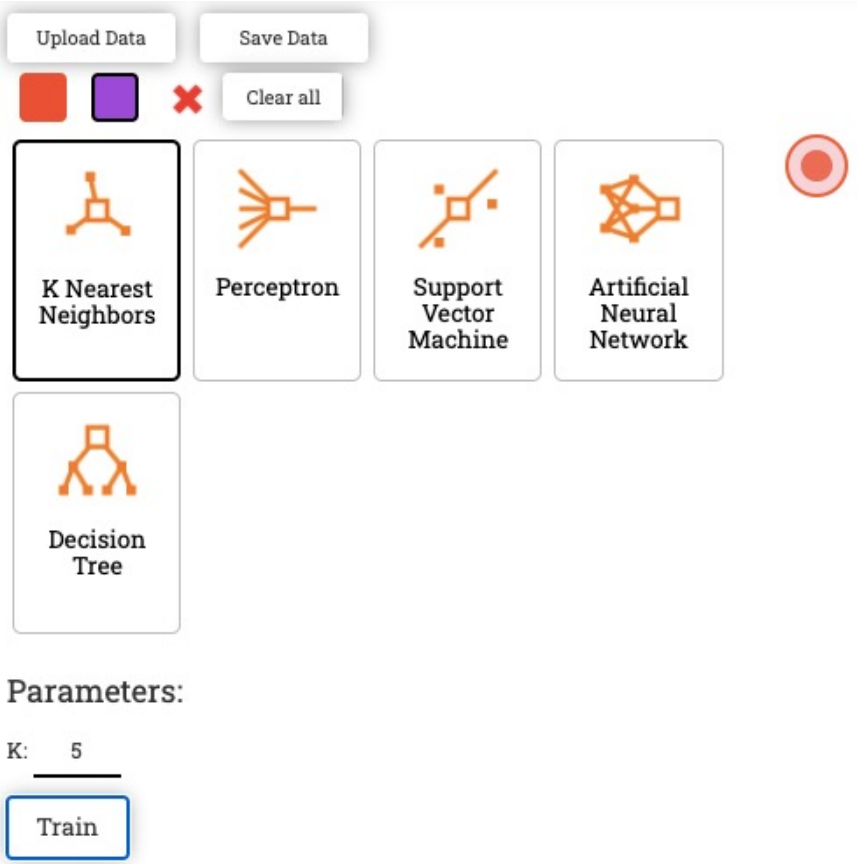
ADDITIONAL MATERIAL

esipap
European School of Instrumentation in Particle & Astroparticle Physics

ML Playground :: https://ml-playground.com

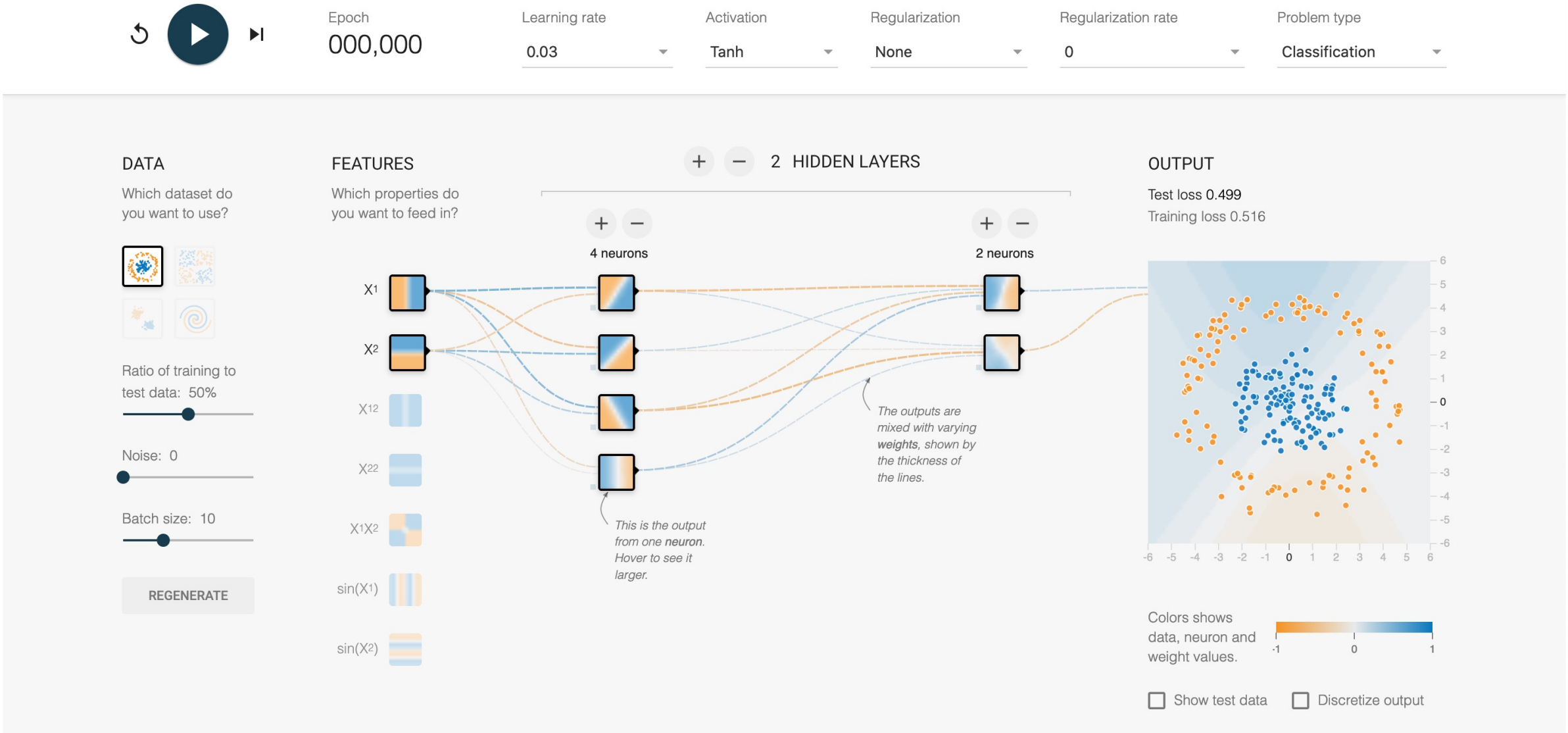
- Website to play live with some of the ML algorithms to get a feeling on how they behave





esipap
European School of Instrumentation in Particle & Astroparticle Physics

Tensorflow Playground :: https://playground.tensorflow.org



Scikit-Learn :: http://scikit-learn.org

- Good to **start on ML**, limited support for NN/DL, no GPU support
  - Simple and efficient tools for data mining and data analysis
  - Accessible to everybody, and reusable in various contexts
  - Built on NumPy, SciPy, and matplotlib
  - Open source, commercially usable - BSD license

esipap
European School of Instrumentation in Particle & Astroparticle Physics

Scikit-Learn :: Classifier Comparison