Industrial PhD

New applications for machine learning and transfer learning algorithms in the manufacturing industry and in particle physics.

Lorena Dieste

TRIPLEALPHA innovation

IGFAE Instituto Galego de Física de Altas Enerxías

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

INDEX

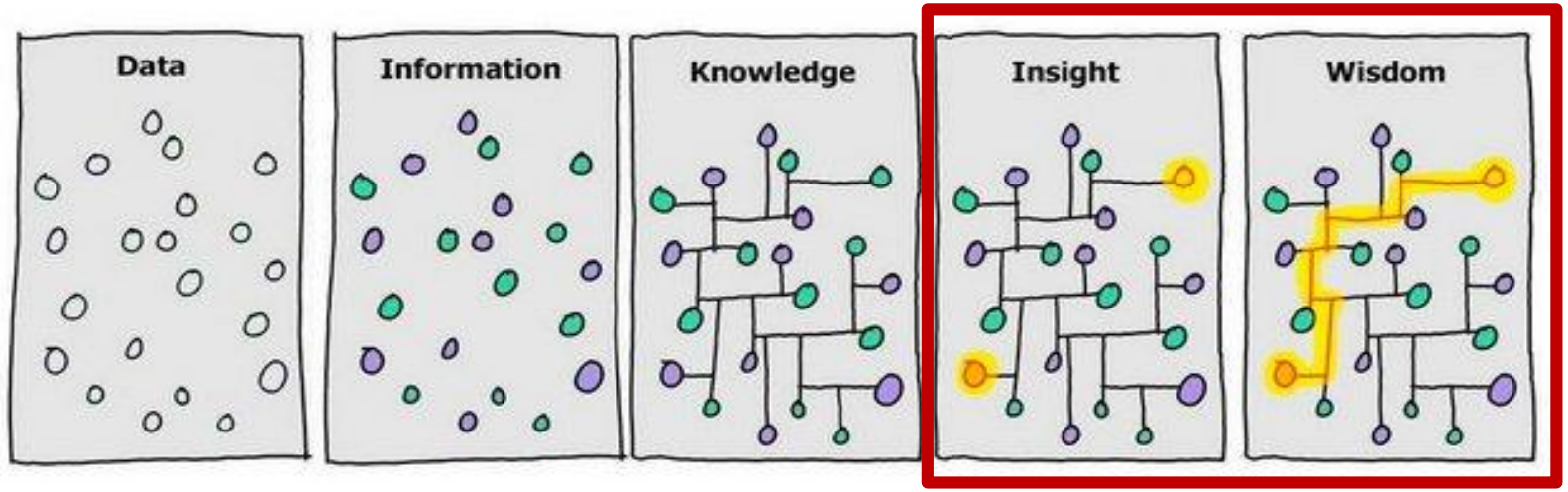
- Introduction
- TMVA vs. SKlearn
- The LHCb Analysis
- Future work

Introduction

An **industrial PhD student** participates in an industrial research or experimental development project directly related to his thesis, developed in a company and in a university.

- Which company? → TripleAlpha
- **TripleAlpha** is a technological firm specialized in data analysis applied to Production and Operations.
- Which center? → **IGFAE**, LHCb experiment.
- The Galician Institute of High Energy Physics is a joint research center of University of Santiago de Compostela and Xunta de Galicia.

"We help companies to be more productive and profitable thanks to the optimization of production and logistic processes"



Merging Data Science with Manufacturing expertise

The title of the thesis is:

- *New applications for machine learning and transfer learning algorithms in the manufacturing industry and in particle physics.*

The main goal is to find and use machine learning techniques that can be applied to improve the results for different fields, in the case of this study we will focus on **particle physics** and the **manufacturing industry**.

TMVA vs. Sklearn

2. TMVA vs Python

The first step is to compare the tools that are used for particle physics and find the one that gets better results:

- TMVA a ROOT (open-source data analysis framework) tool used by high energy physics and others.
- Python Machine Learning Tools as Scikit-Learn

VS.

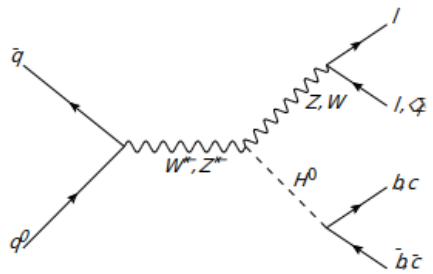
- We will use different types of topologies representing different types of problems of High Energy Physics.

We studied three different types of topologies for this analysis:

1) KS0 → π^+ π^- In this exercise our goal is to discriminate signal vs. Background.

2)Ds→ η' (→ μ^+ μ^-) + π . In this exercise our goal is to discriminate signal vs.. Background.

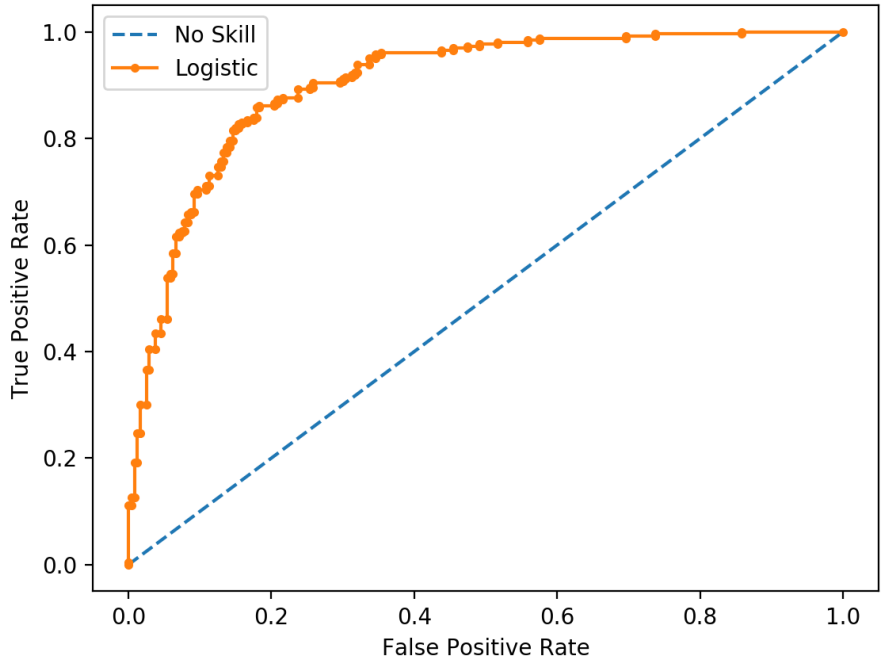
3)W+bb and tt cross sections. In this excercise our goal is to discriminate them with each other.



For now we use Monte Carlo LHCb to get the data for the analysis.

To compare the quality of the results we used two figures of merit:

- The **ROC curve**: a plot that confronts the rate of True Positives vs. False Positives



- The **correlation** between the results and the mass

2. TMVA vs Python

To find our model we used **Grid search.** Grid search is used to find the optimal hyperparameters of a model which results in the most ‘accurate’ predictions.

In our case first we tried different models with different algorithms as:

- BDT with AdaBoost
- BDT with GradientBoosting
- BDT with XGBoost
- RandomForestClassifier
- Neural Network
- Logistic Regression.

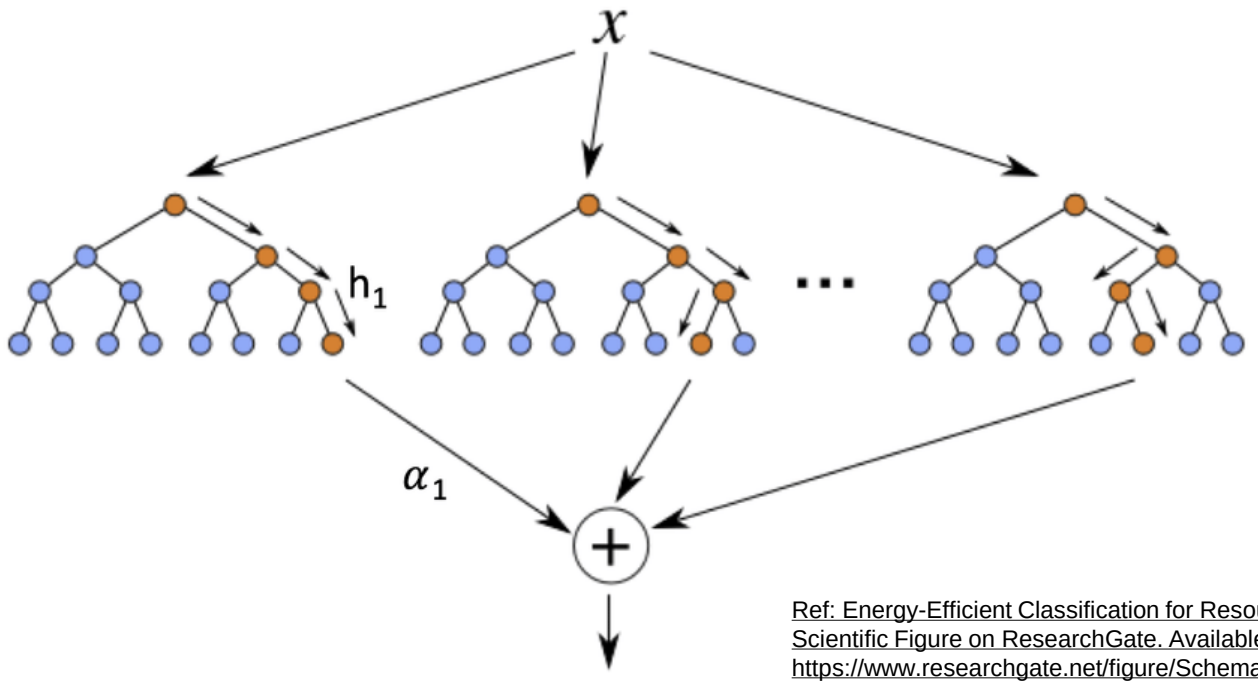
```
for c in [2^-5, 2^-3,...,2^15]:
 for g in [2^-15, 2^-13, ..., 2^5]:
  for train, test in partition:
    model = svm_train(train, c, g)
    score = svm_predict(test, model)
    cv_list.insert (score)
  scores_list.insert(mean(cv_list),c,g)
print max(scores_list)
```

For each one of this we run a Grid Search to find the optimal hyperparameters and after the process we selected the algorithm with the best results.

2. TMVA vs Python

The algorithm we decided to use in both platforms was **BDT,** using boosting **AdaBoost.** We also tried Gradient Boosting and XGBoost.

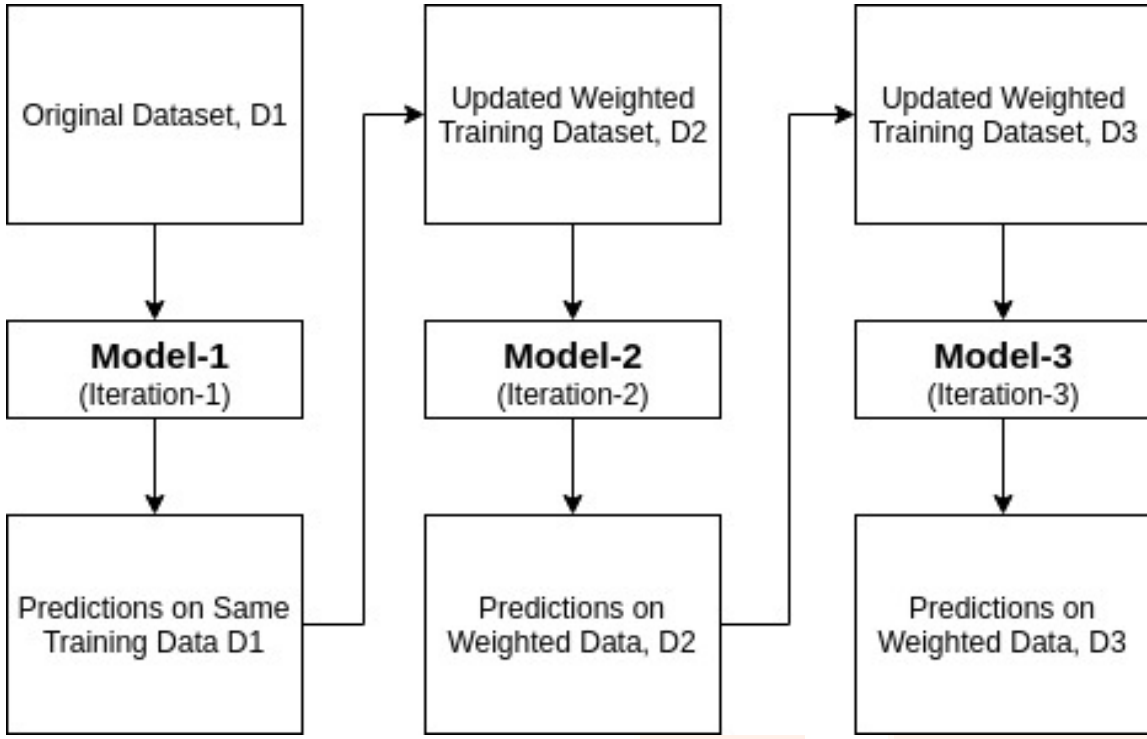
- BDT (Boosted Decission Trees) : is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an **ensemble of weak prediction models**, typically decision trees.



Ref: Energy-Efficient Classification for Resource-Constrained Biomedical Applications - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Schematic-diagram-of-a-boosted-ensemble-of-decision-trees_fig2_325632132 [accessed 8 Oct, 2020]

Ada-boost or **Adaptive Boosting** is an iterative ensemble classifier, it builds a strong classifier by combining multiple poorly performing classifiers so that you will get a high accuracy strong classifier.

- It assigns the higher weight to wrong classified observations.
- The more accurate classifier will get high weight.
- This process iterate until the complete training data fits:
  - without any error
  - reached to the specified maximum number of estimators.
- To classify, perform a "vote" across all of the learning algorithms you built.



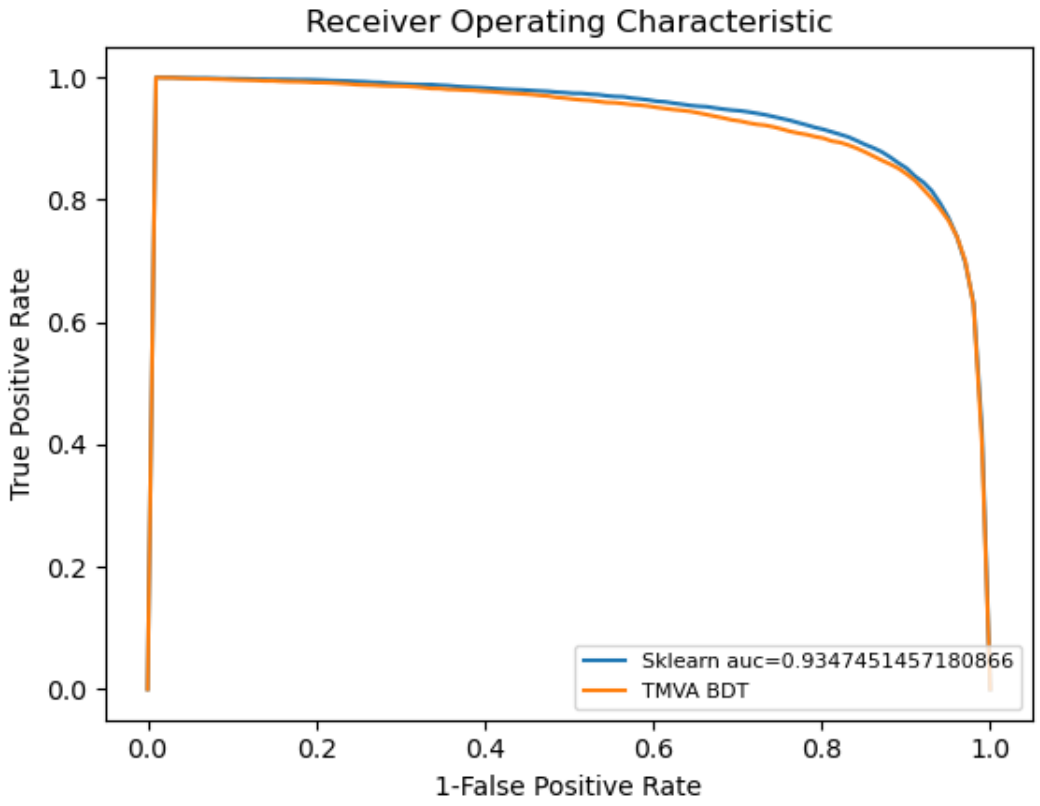
KS0 → $\pi^+ \pi^-$ **variables** :

- 'd1_ip'
- 'd1_pt'
- 'd2_ip'
- 'd2_pt'
- 'doca'
- 'ipchi2'
- 'pt'

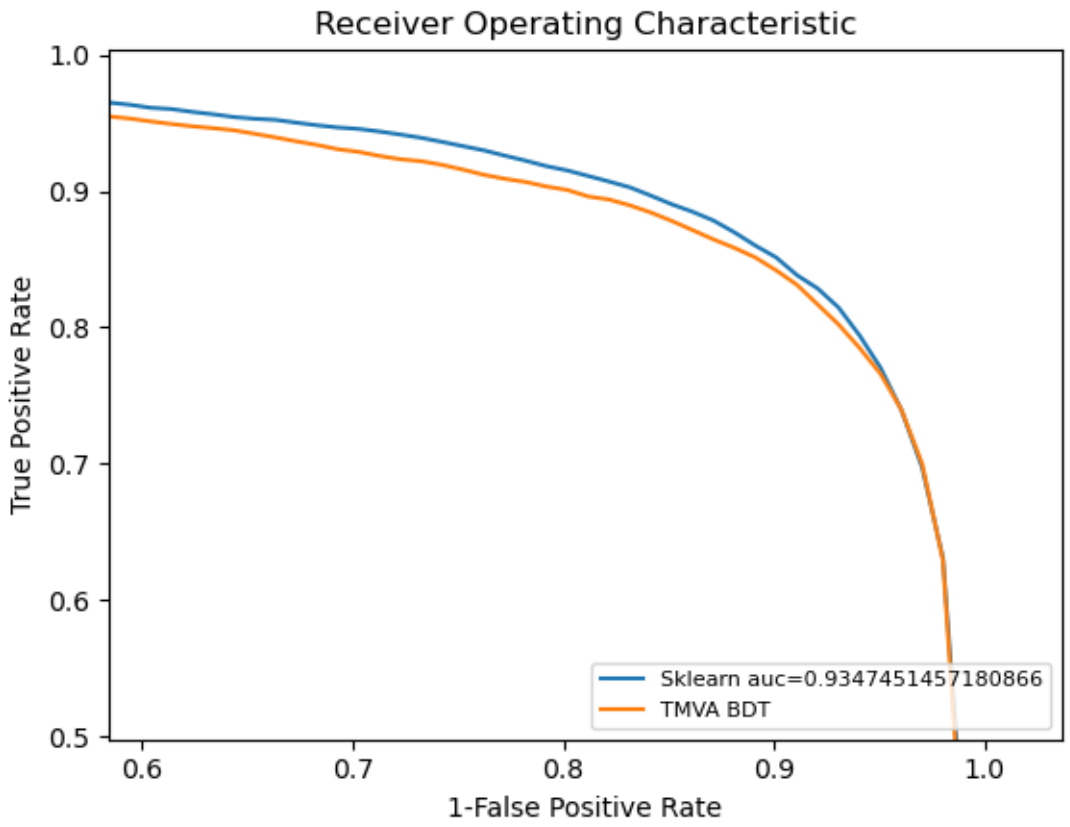
Correlation:

- 'mass'

KS0 → $\pi^+ \pi^-$ results **ROC curve**:

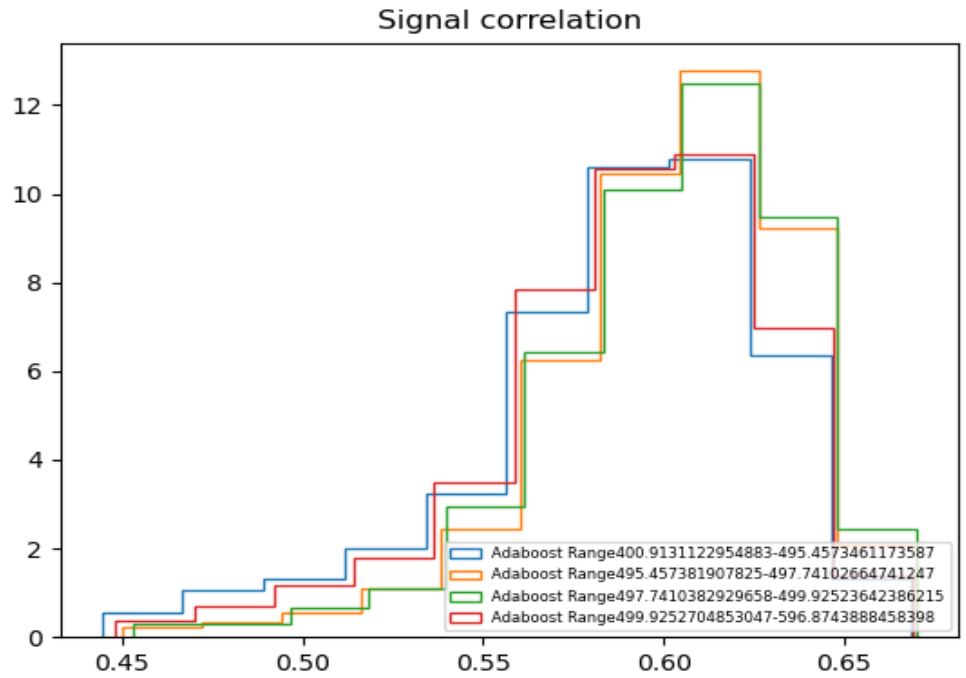


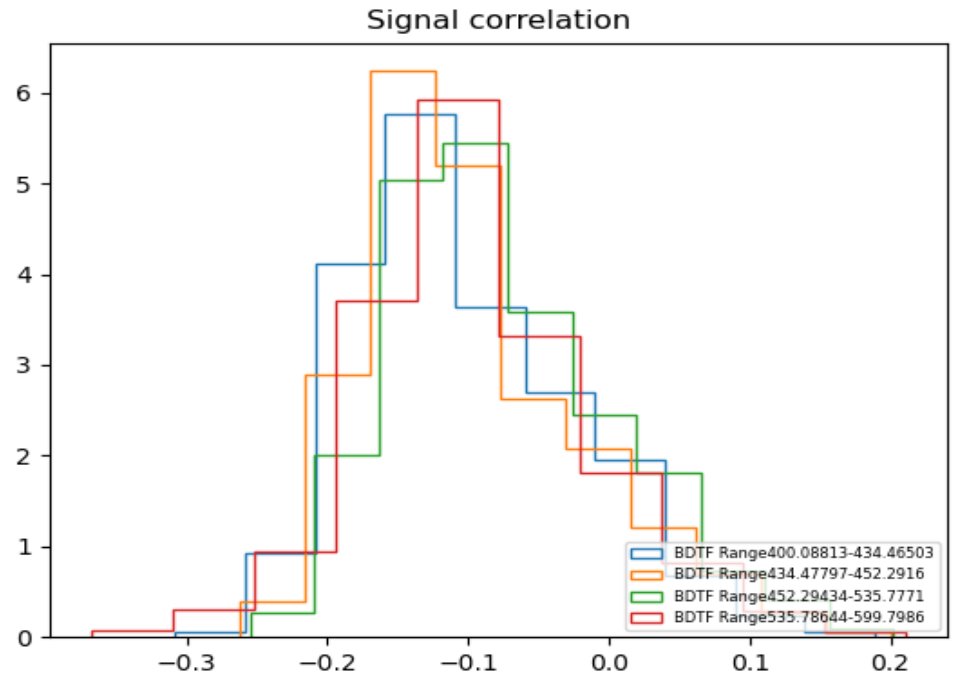
Zoom of the results:



The curve for sklearn is slightly better

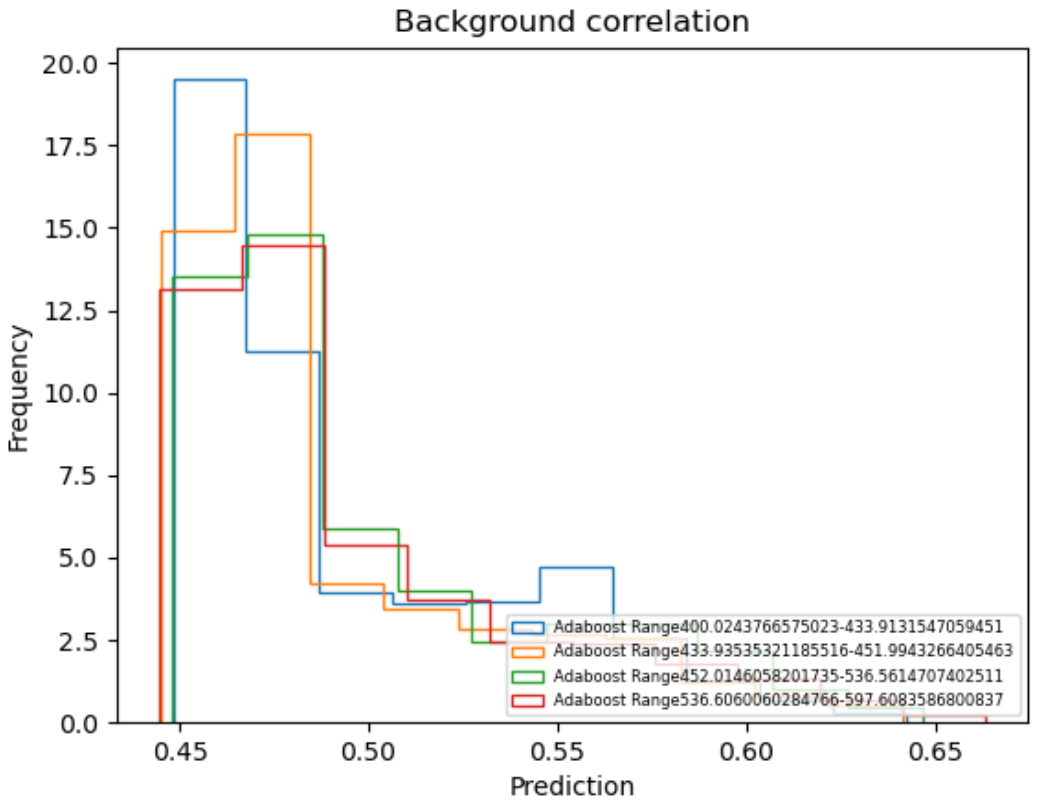
KS0 → $\pi^+ \pi^-$ results for signal **correlation**:

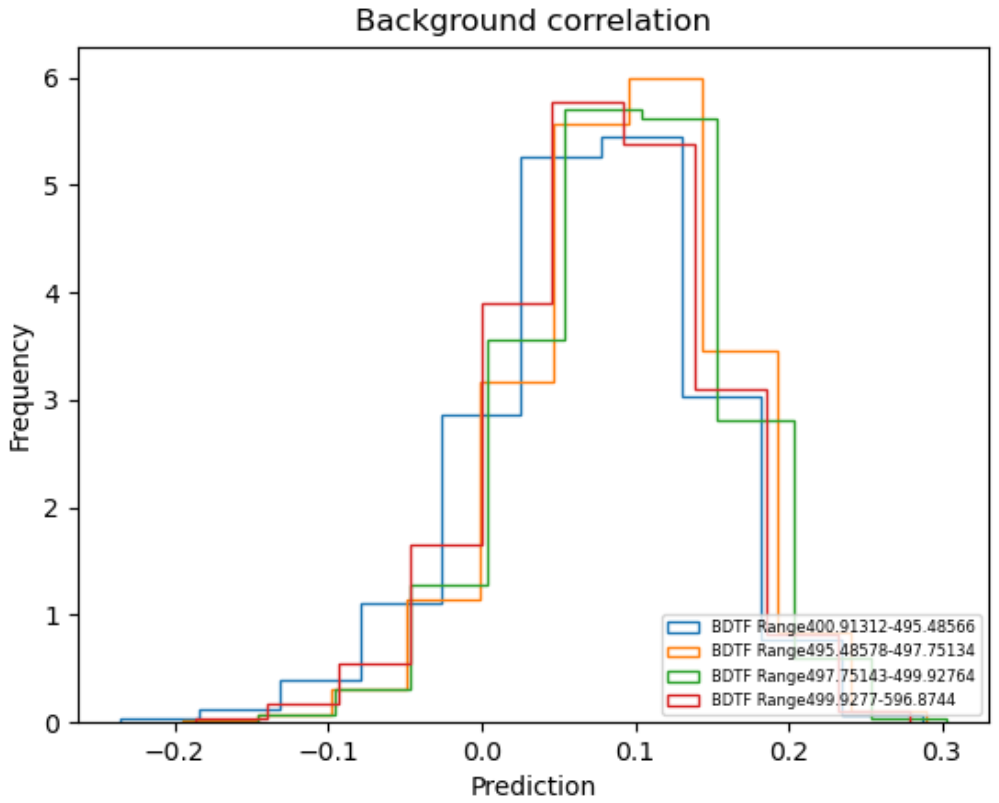




We can observe that there is no correlation.

KS0 → π^+ π^- results for background **correlation**:





We can observe that there is no correlation.

2. TMVA vs Python

Ds→ η' (→ $\pi^+ \pi^- \mu^+ \mu^-$) + π variables :

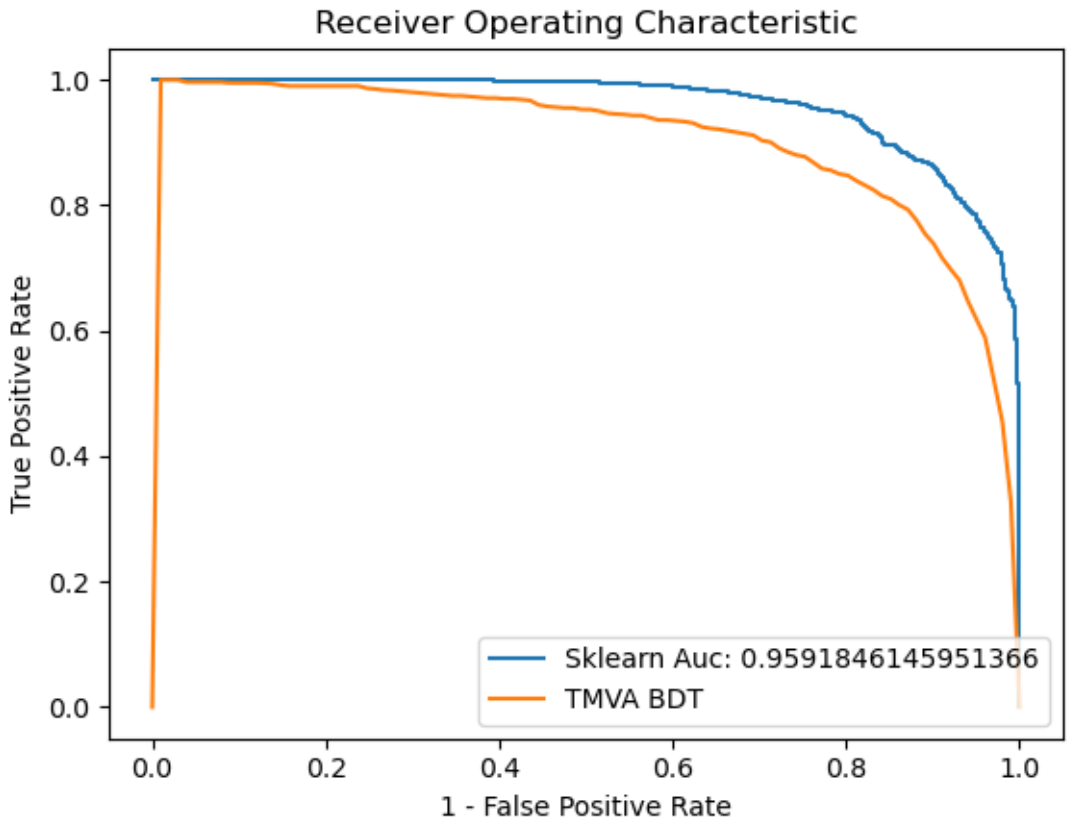
- 'etap_IPCHI2_OWNPV'
- 'etap_FDCHI2_OWNPV'
- 'etap_VCHI2PERDOF'
- 'etap_PT'
- 'Ds_IPCHI2_OWNPV'
- 'Ds_FDCHI2_OWNPV',
- 'Ds_VCHI2PERDOF'
- 'Ds_PT'
- 'pi_PT',
- 'pi_IPCHI2_OWNPV'

- 'pip_eta_PT'
- 'pip_eta_IPCHI2_OWNPV'
- 'pim_eta_PT'
- 'pim_eta_IPCHI2_OWNPV'
- 'mup_PT'
- 'mup_IPCHI2_OWNPV'
- 'mum_PT'
- 'mum_IPCHI2_OWNPV']

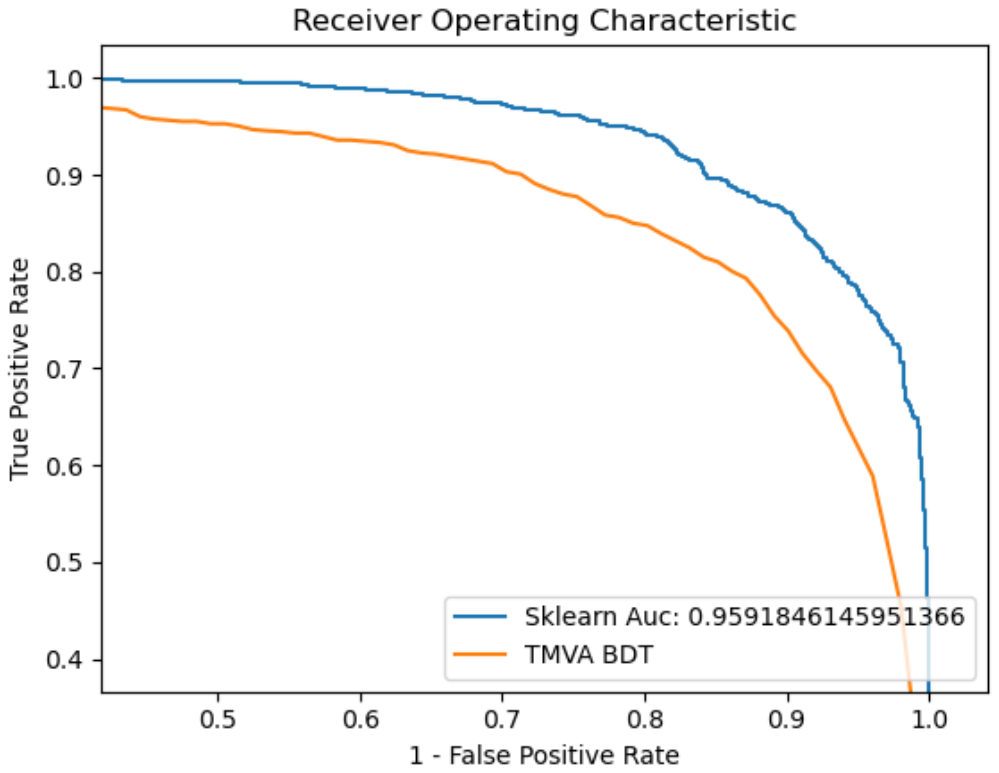
Correlation:

- 'Ds_M'
- 'etap_M'

In $Ds \rightarrow \eta' (\rightarrow \mu^+ \mu^-) + \pi$ we used a K-fold to be able to use all the data for training, the results of the **ROC curve**:

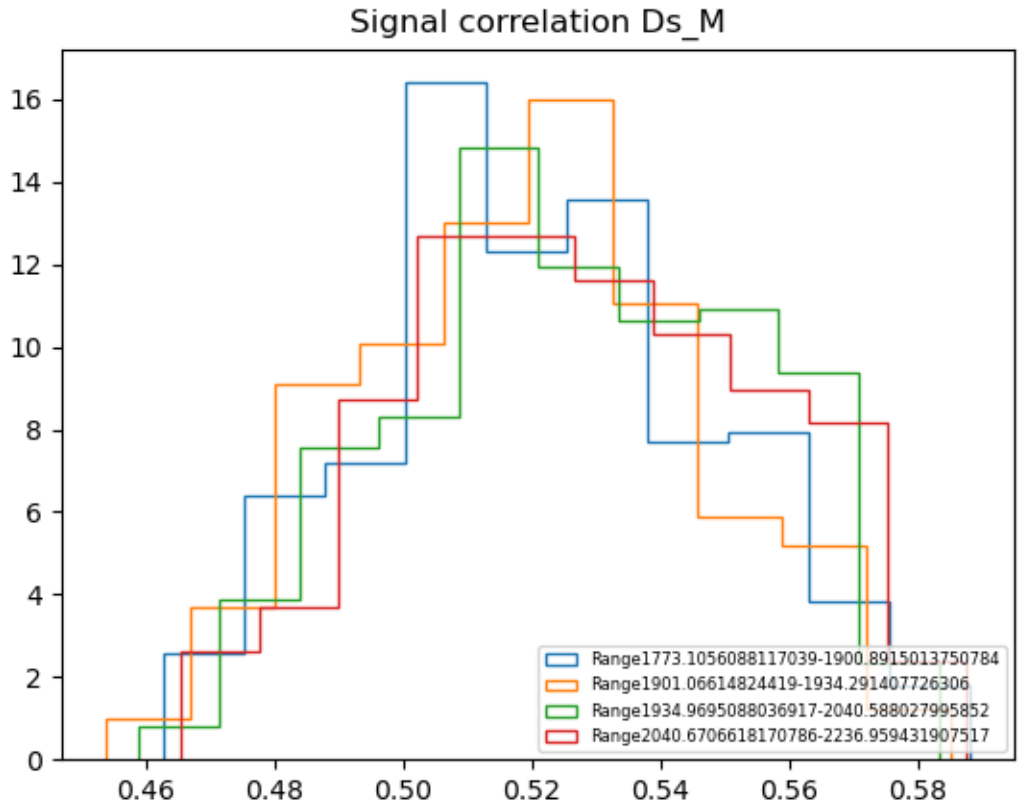


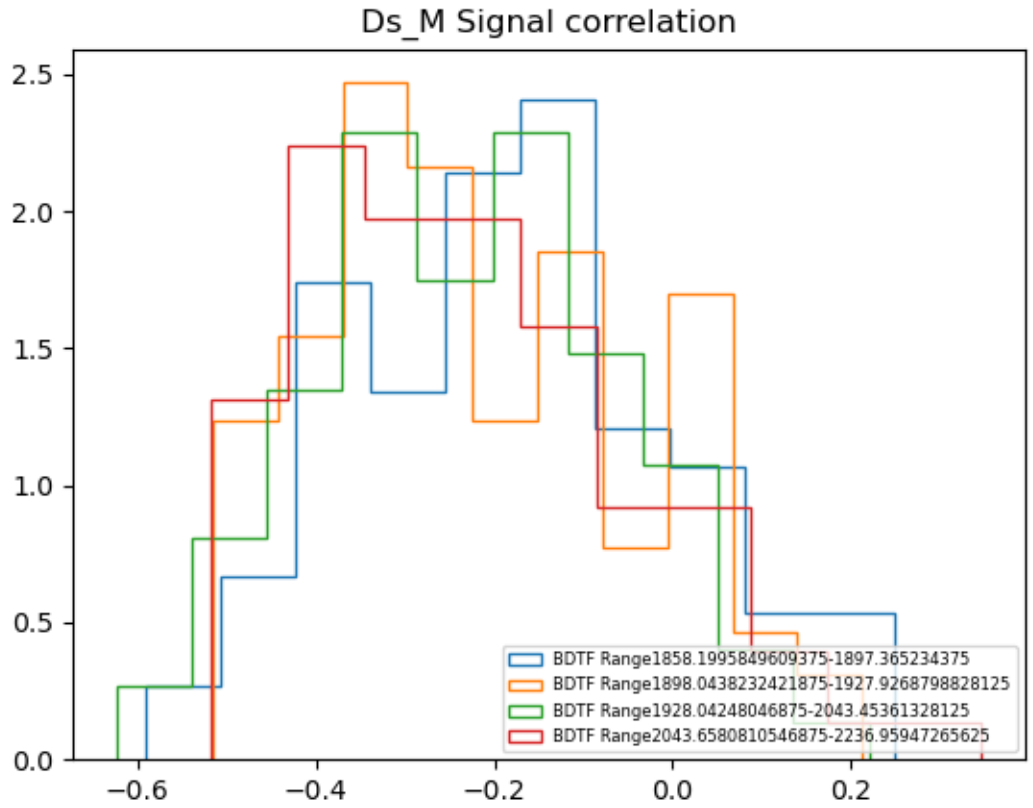
Zoom of the results:



The curves for sklearn have a visible improvement

Second experiment results for signal **correlation**:

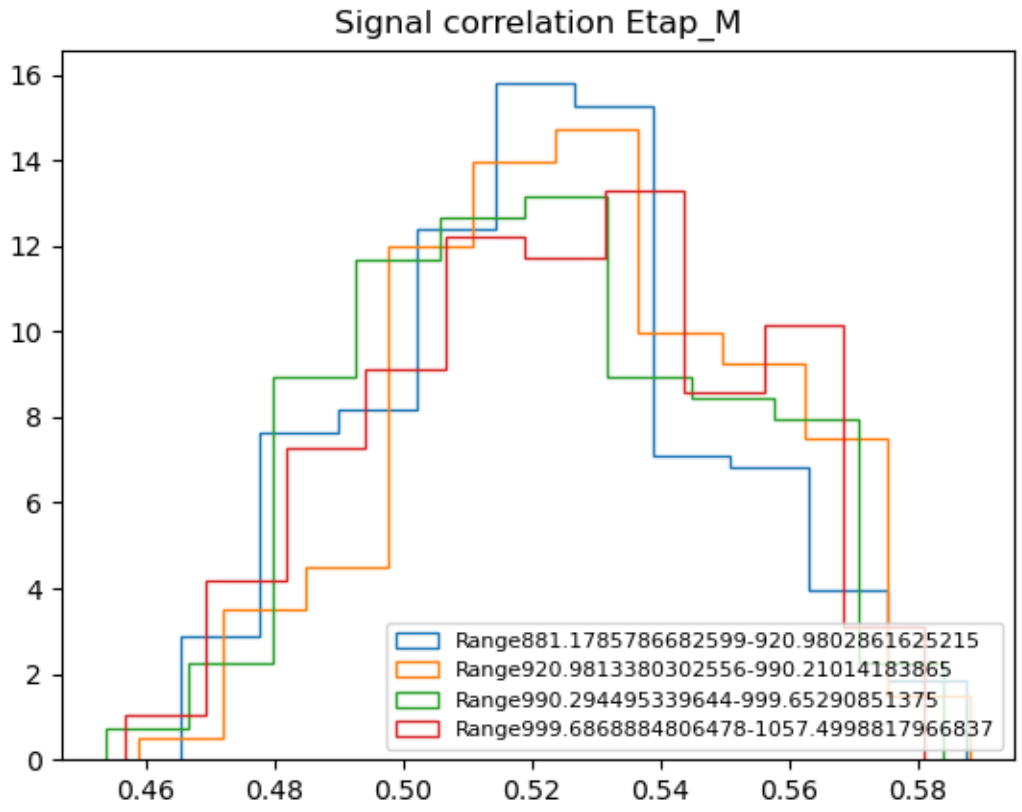


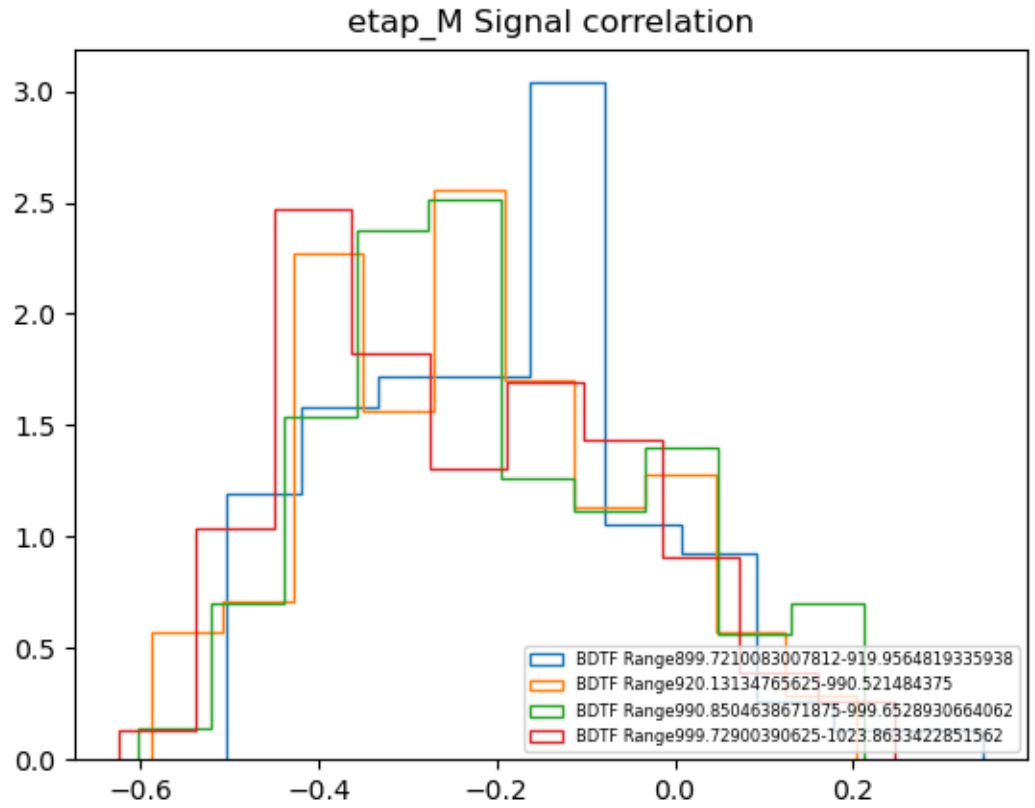


We can observe that there is no correlation.

2. TMVA vs Python

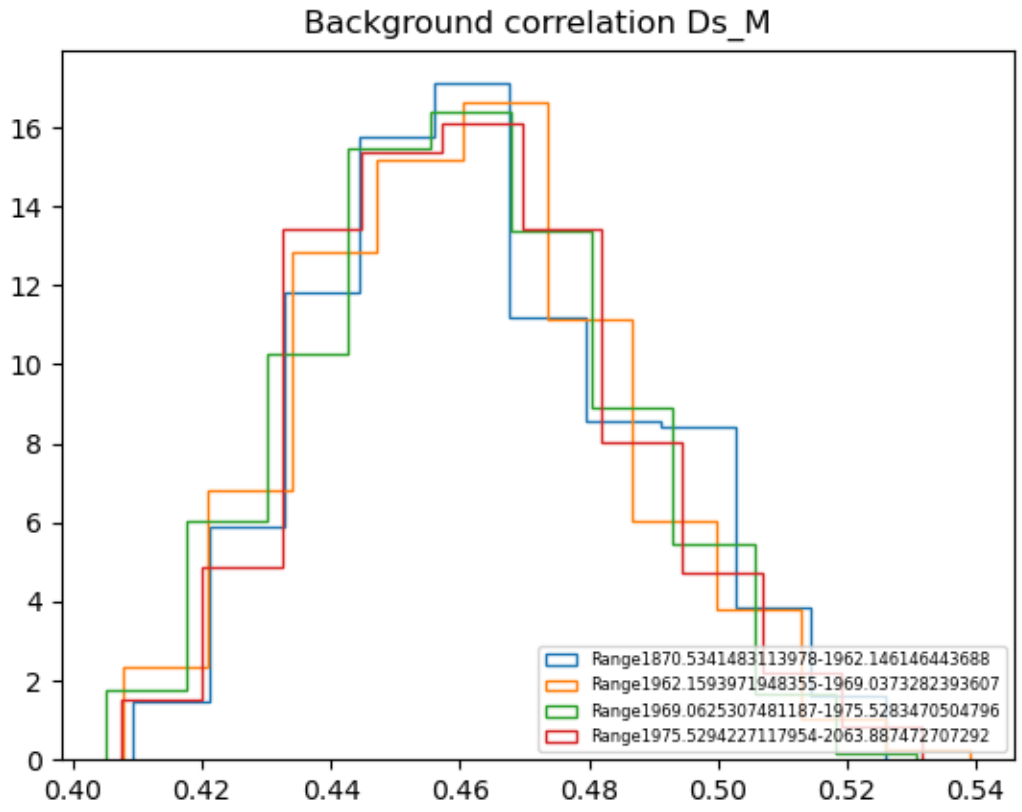
In this example the particle decays from one corps to another and we have two masses, Ds_M and Etap_M

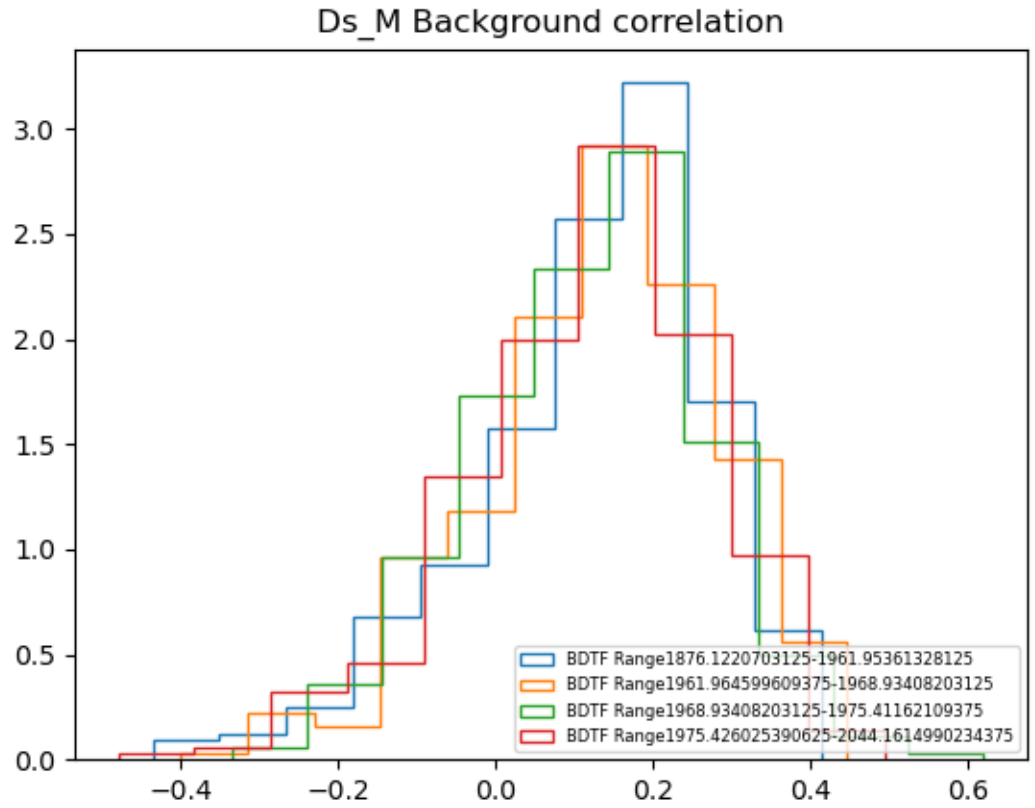




We can observe that there is no correlation.

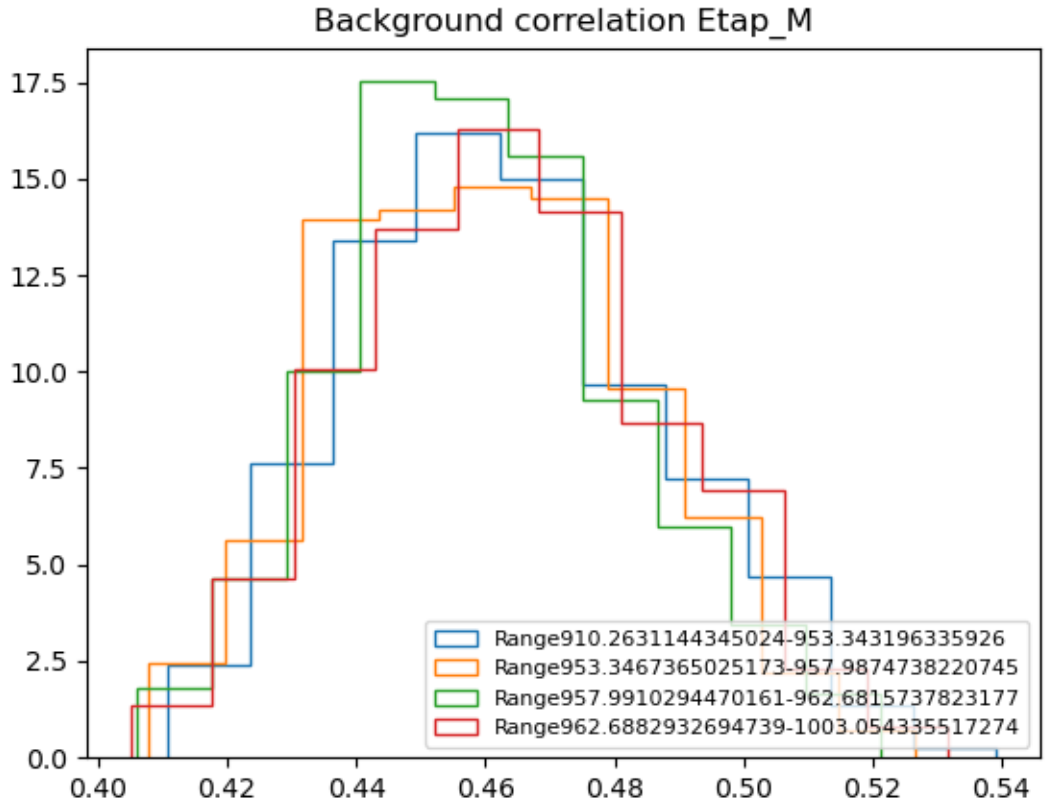
Second experiment results for background **correlation**:

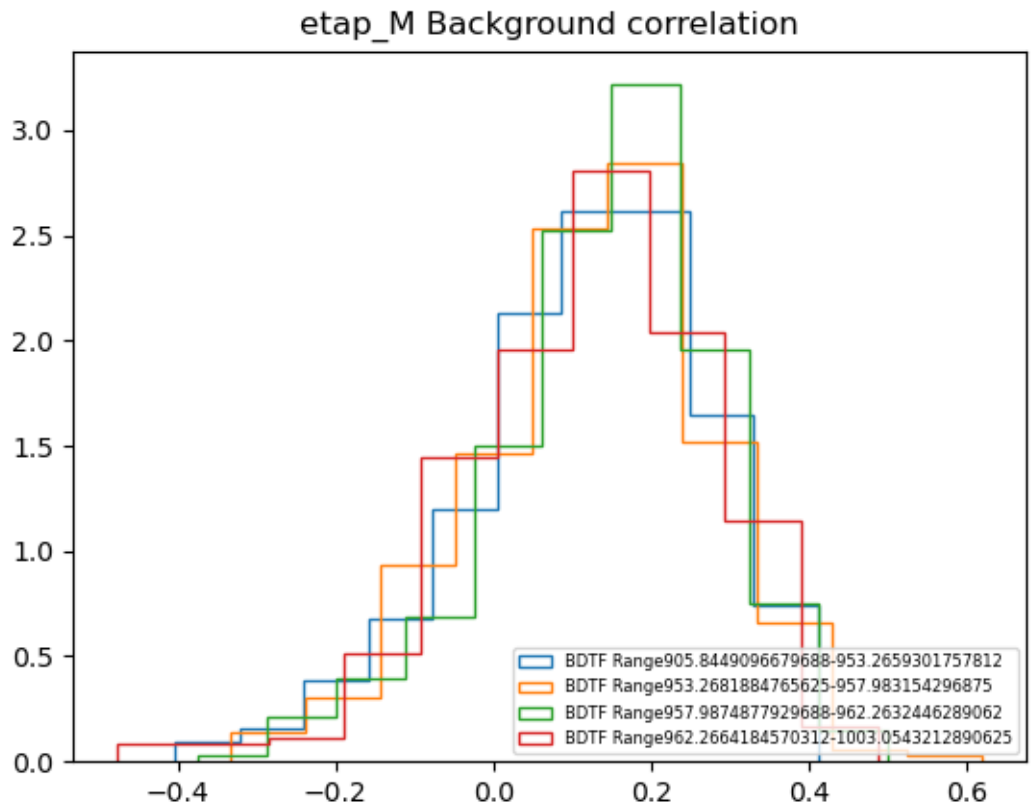




We can observe that there is no correlation.

Second experiment results for background **correlation**:





We can observe that there is no correlation.

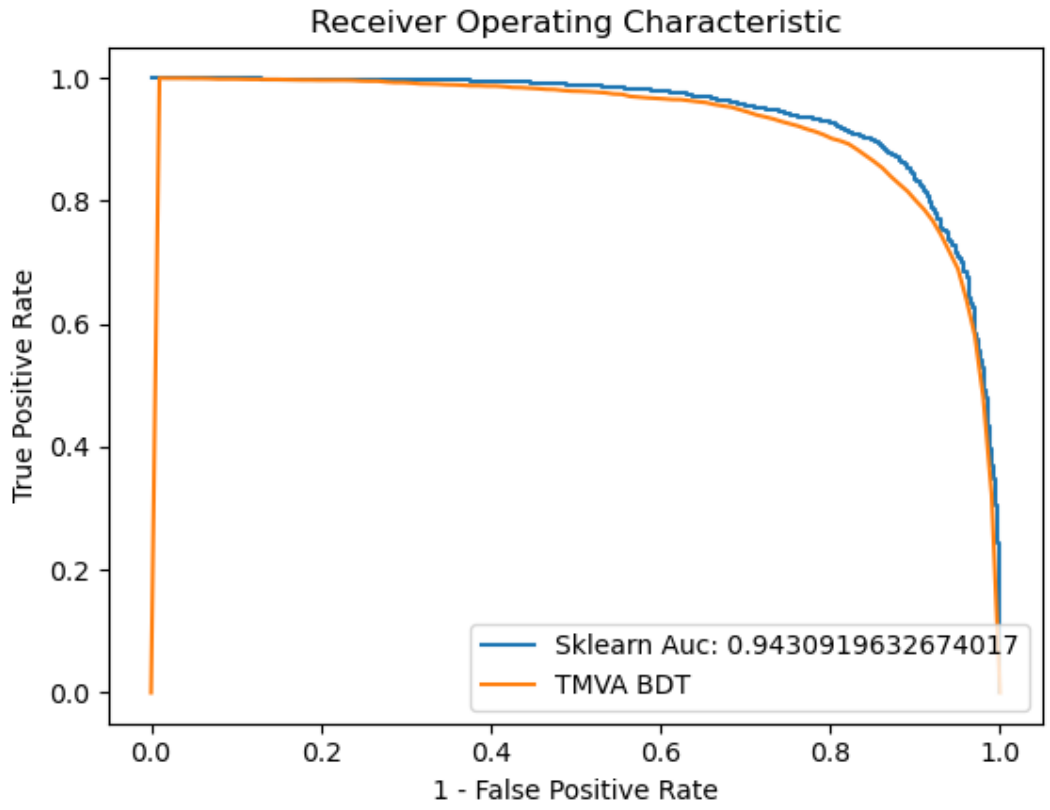
W+bb and tt ：

- "Lepton_ETA"
- "Lepton_PT"
- "DiJet_PT"
- "HP_DR_JET0_LEPTON"
- "HP_DR_JET1_LEPTON"
- "Jet0_JetWidth"
- "Jet1_JetWidth"
- "HP_CosThetaBoost"
- "HP_DR_JET0_JET1"
- "ULE_CELLjet0_PT"
- "PF_ETA_VAR_ERatio_12_3456"

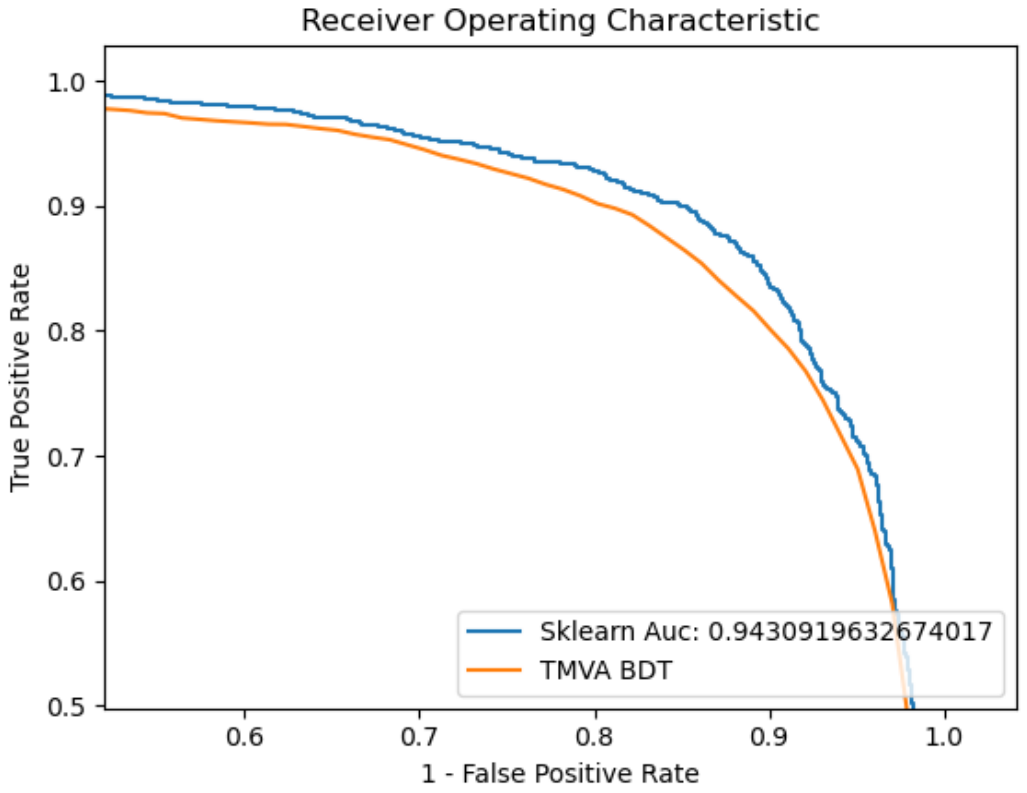
Correlation:

- "DiJet_M"

W+bb and tt results **ROC curve**:

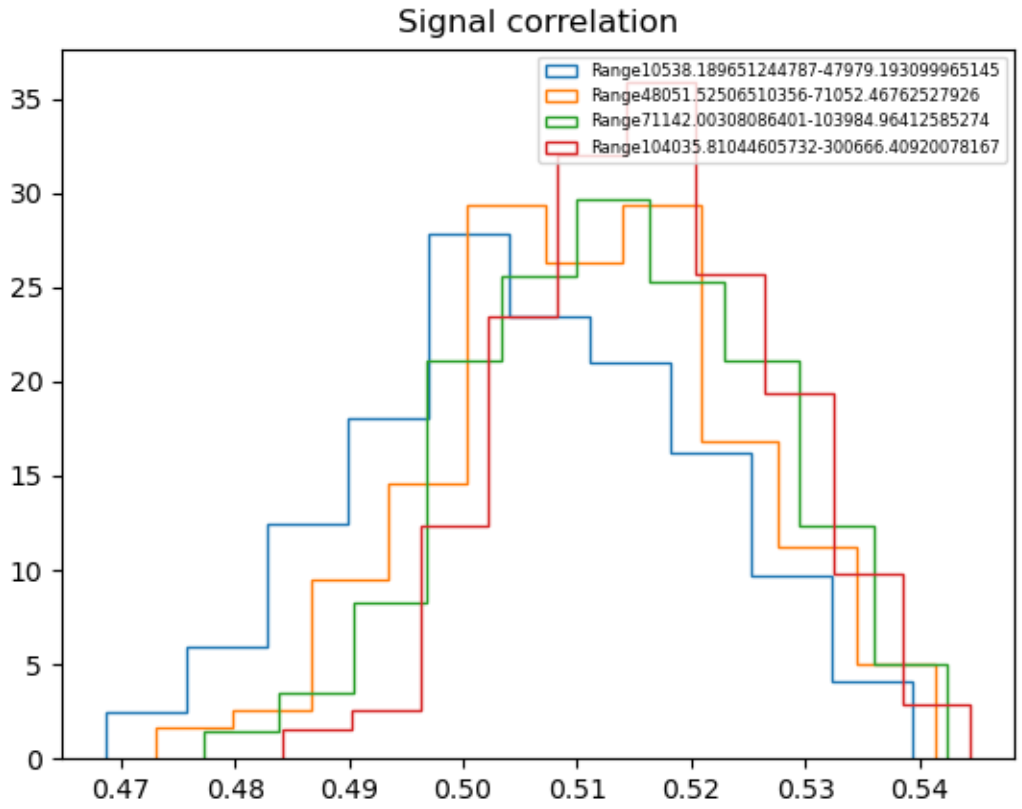


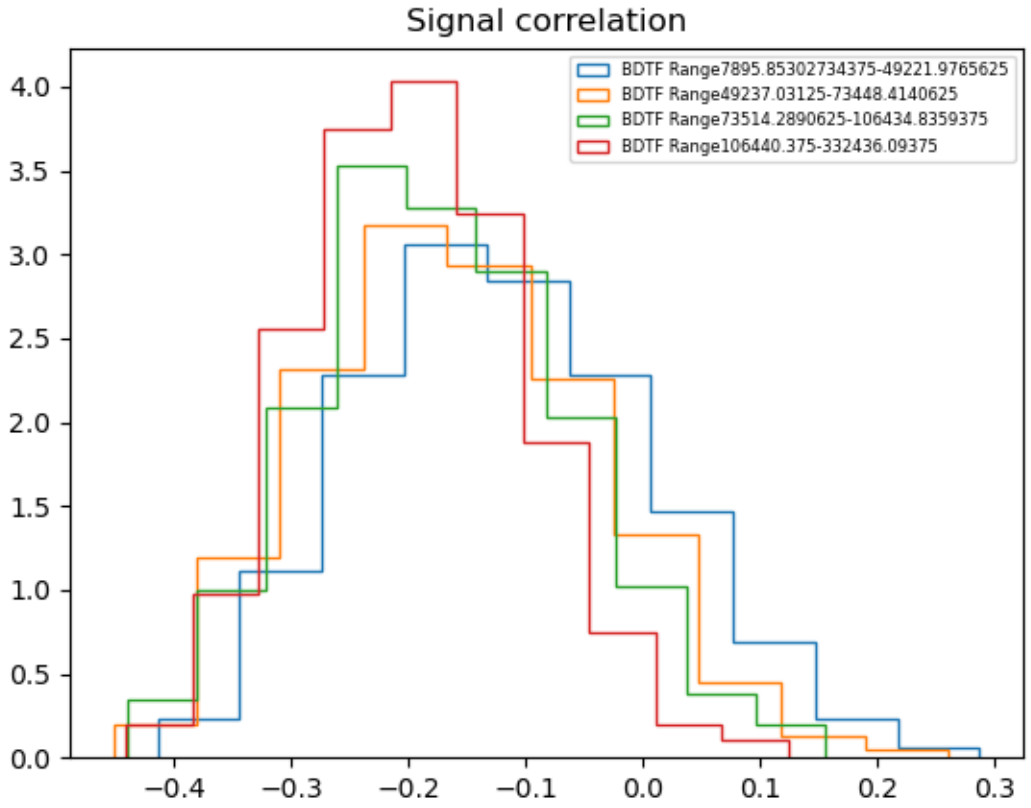
Zoom of the results:



The curve for sklearn is better than the TMVA's

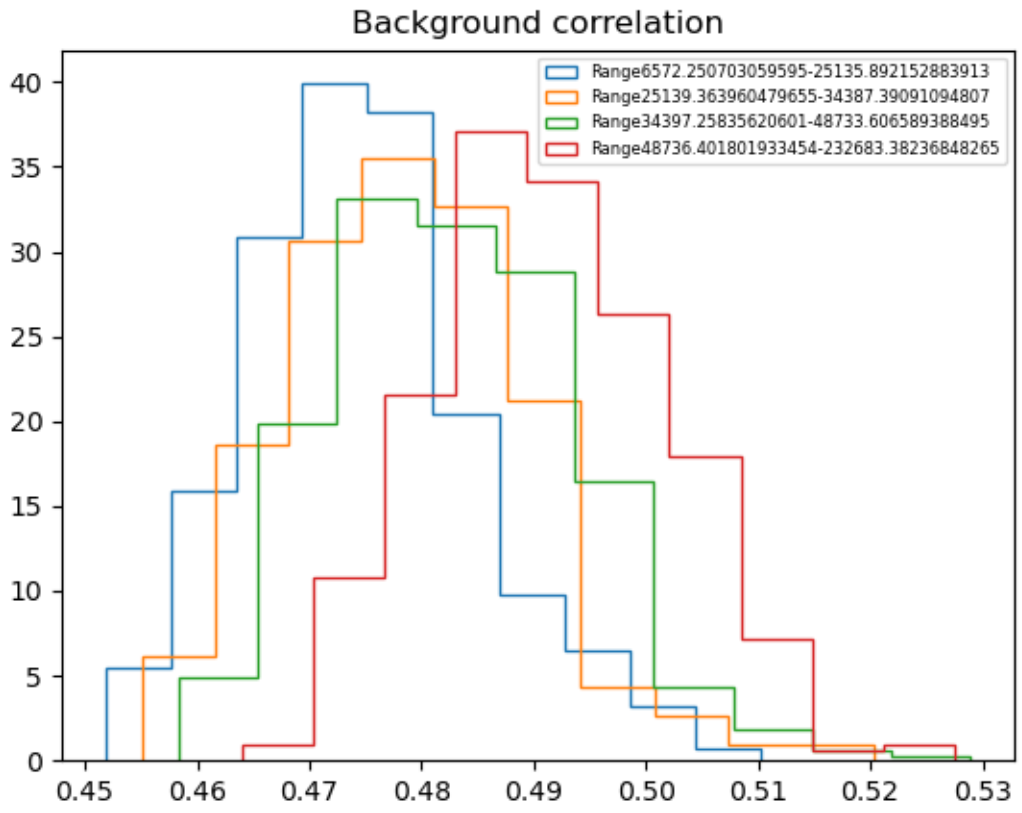
Third experiment results for signal **correlation**:

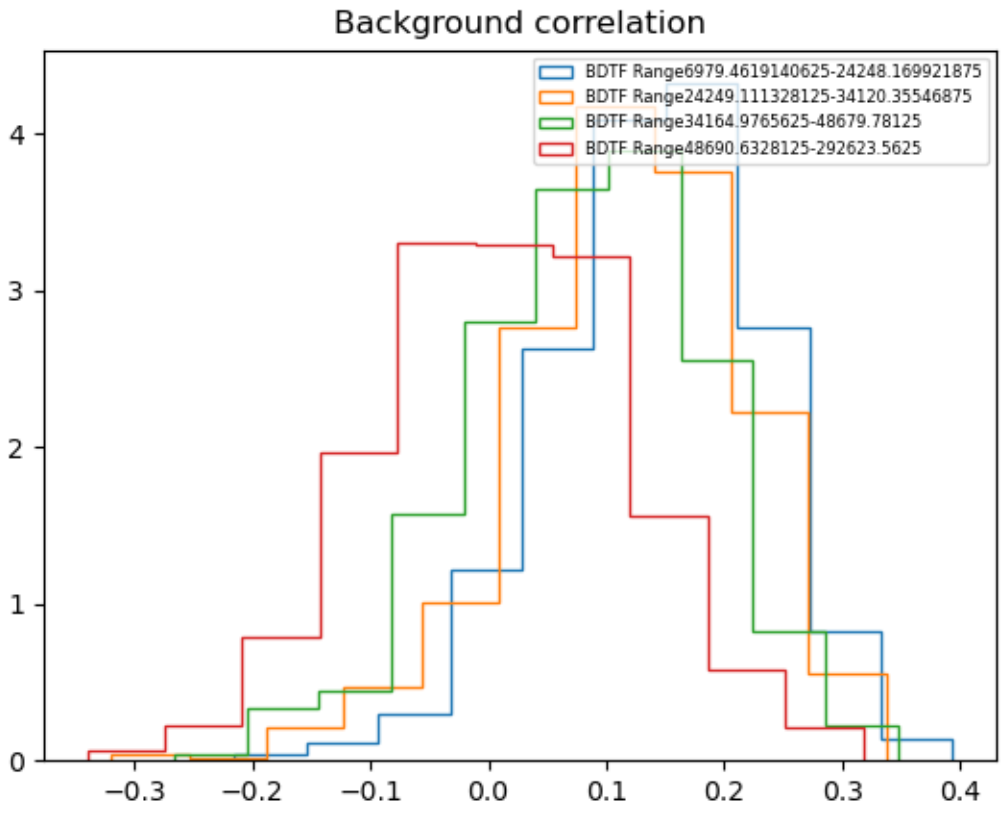




We can observe that there is no correlation in both results.

Third experiment results for background **correlation**:





We can observe that there is correlation in both results.

2. Conclusions

- After three different results we concluded that the best results were available using Scikit-Learn instead of TMVA.
  - The ROC Curves in all cases we better for the scikit-learn results using the same parameters
  - The correlation with the mass behaved similarly enough in both cases
  - In conclusion the results obtained using Scikit-learn are prefered than the TMVA ones for the same problem and same specifications.

2. Future Work

1) We need to keep proving with different data that this conclusion is correct and not just a singular occasion in this specific data.

2) We will use Pythia to generate data for the next analysis in order to avoid the use of LHCb data that can't be used for publications outside the CERN.

3) We need to prove why the results under the same specifications work better for our module in scikit-learn Python than our module in TMVA Python.

LHCb Analysis

Another line of work of the thesis is to help in different analysis in the LHCb collaboration.

We got three different groups of data:

- $Ds \rightarrow \eta' (\rightarrow \pi^+ \pi^- \mu^+ \mu^-) + \pi$
- $Ds \rightarrow \eta' (\rightarrow \mu^+ \mu^-) + \pi$
- $Ds \rightarrow \eta (\rightarrow \mu^+ \mu^-) + \pi$

In this case we want to measure the branching fractions of the decays that are theoretically predicted, this experimental measurement is useful for understanding Chiral Perturbation Theory. [1]

1.**GAN, L., KUBIS, B., PASSEMAR, E. Y TULIN, S.**
Precision tests of fundamental physics with η and $\eta^\prime$ mesons

Our goal was to create a Python module, using the same Machine Learning algorithm as before (BDT), to get the results of the classification for the three different types of data:

- $Ds \rightarrow \eta' (\rightarrow \pi^+ \pi^- \mu^+ \mu^-) + \pi$
- $Ds \rightarrow \eta' (\rightarrow \mu^+ \mu^-) + \pi$
- $Ds \rightarrow \eta (\rightarrow \mu^+ \mu^-) + \pi$

- We used as data the LHCb RUN2 dataset including 2016, 2017 and 2018. As the data we had was not extensive we decide to use a K-fold of the different years as training for the others. We compared the variables of the different years to prove that the data was similar enough to train the models.

Ds→ η' (→ $\pi^+ \pi^- \mu^+ \mu^-$) + π variables :

- 'etap_IPCHI2_OWNPV'
- 'etap_FDCHI2_OWNPV'
- 'etap_VCHI2PERDOF'
- 'etap_PT'
- 'Ds_IPCHI2_OWNPV'
- 'Ds_FDCHI2_OWNPV',
- 'Ds_VCHI2PERDOF'
- 'Ds_PT'
- 'pi_PT',
- 'pi_IPCHI2_OWNPV'

- 'pip_eta_PT'
- 'pip_eta_IPCHI2_OWNPV'
- 'pim_eta_PT'
- 'pim_eta_IPCHI2_OWNPV'
- 'mup_PT'
- 'mup_IPCHI2_OWNPV'
- 'mum_PT'
- 'mum_IPCHI2_OWNPV']

Correlation:

- 'Ds_M'
- 'etap_M'

Ds→ η' (→ $\mu^+ \mu^-$) + π' :

- 'etap_IPCHI2_OWNPV'
- 'etap_FDCHI2_OWNPV'
- 'etap_VCHI2PERDOF'
- 'etap_PT'
- 'Ds_IPCHI2_OWNPV'
- 'Ds_FDCHI2_OWNPV'
- 'Ds_VCHI2PERDOF'
- 'Ds_PT'
- 'pi_PT'
- 'pi_IPCHI2_OWNPV'

Correlation:

- 'Ds_M'
- 'etap_M'

Ds→ η (→ $\mu^+ \mu^-$) + π:

- 'eta_IPCHI2_OWNPV'
- 'eta_FDCHI2_OWNPV'
- 'eta_VCHI2PERDOF'
- 'eta_PT','Ds_IPCHI2_OWNPV'
- 'Ds_FDCHI2_OWNPV'
- 'Ds_VCHI2PERDOF'
- 'Ds_PT'
- 'pi_PT'
- 'pi_IPCHI2_OWNPV'

Correlation:

- 'Ds_M'
- 'eta_M'

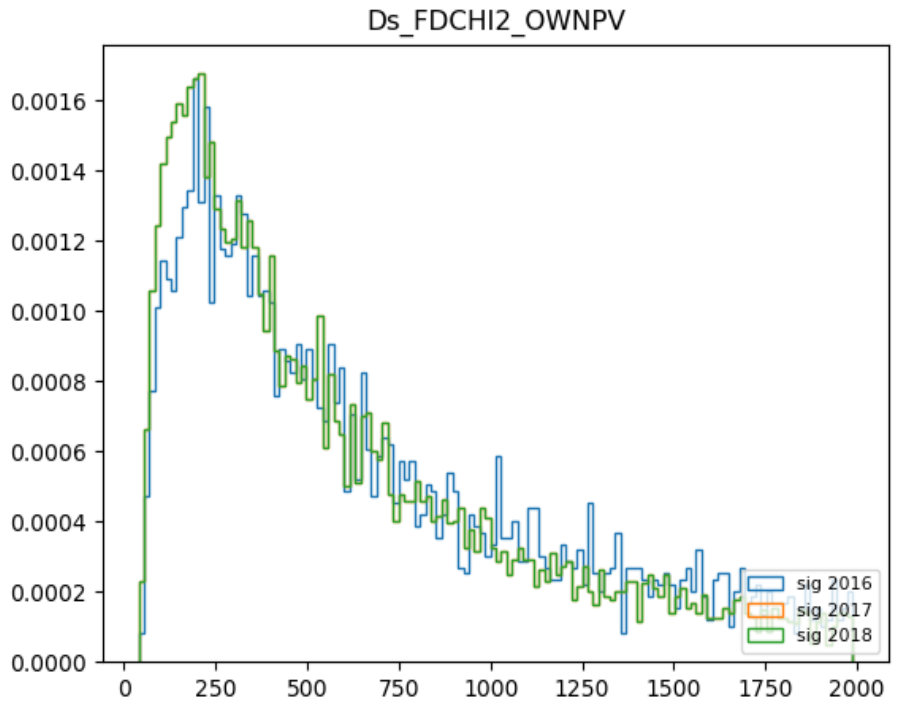
Isolation for Ds→ η (→ $\mu^+ \mu^-$) + π
And Ds→ η' (→ $\mu^+ \mu^-$) + π:

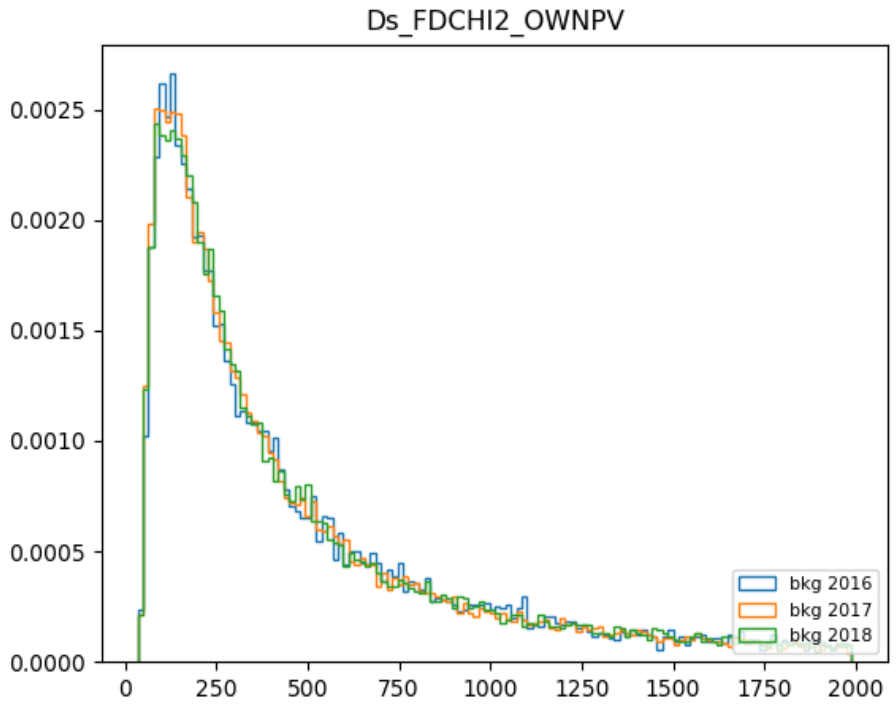
- 'Ds_VertexIsolation_VTXISODCHI2ONETRACK'
- 'Ds_VertexIsolation_VTXISODCHI2TWOTRACK'
- 'pi_TrackIsolationBDT_TRKISOBDTTHIRDVALUE'

Isolation for Ds→ η' (→ $\pi^+ \pi^- \mu^+ \mu^-$) + π

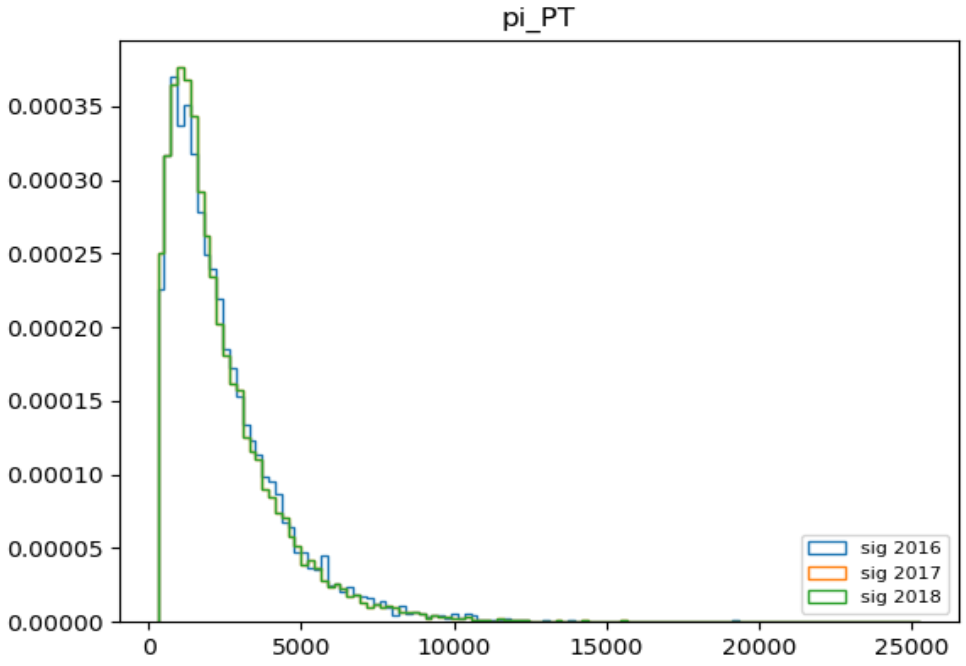
- 'Ds_VertexIsolation_VTXISODCHI2ONETRACK'
- 'Ds_VertexIsolation_VTXISODCHI2TWOTRACK'

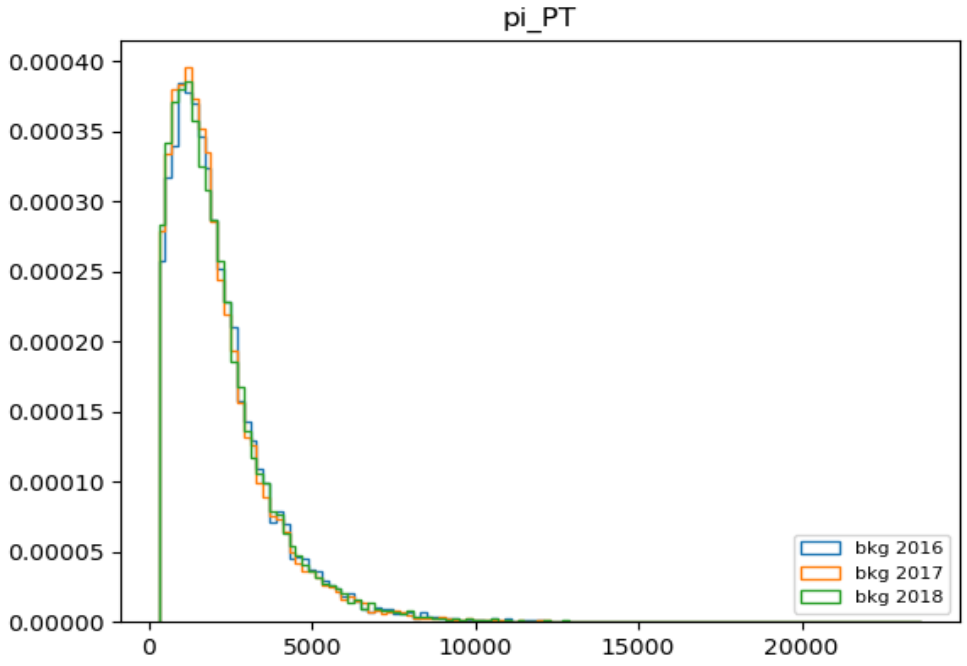
$Ds \rightarrow \eta (\rightarrow \mu^+ \mu^-) + \pi$



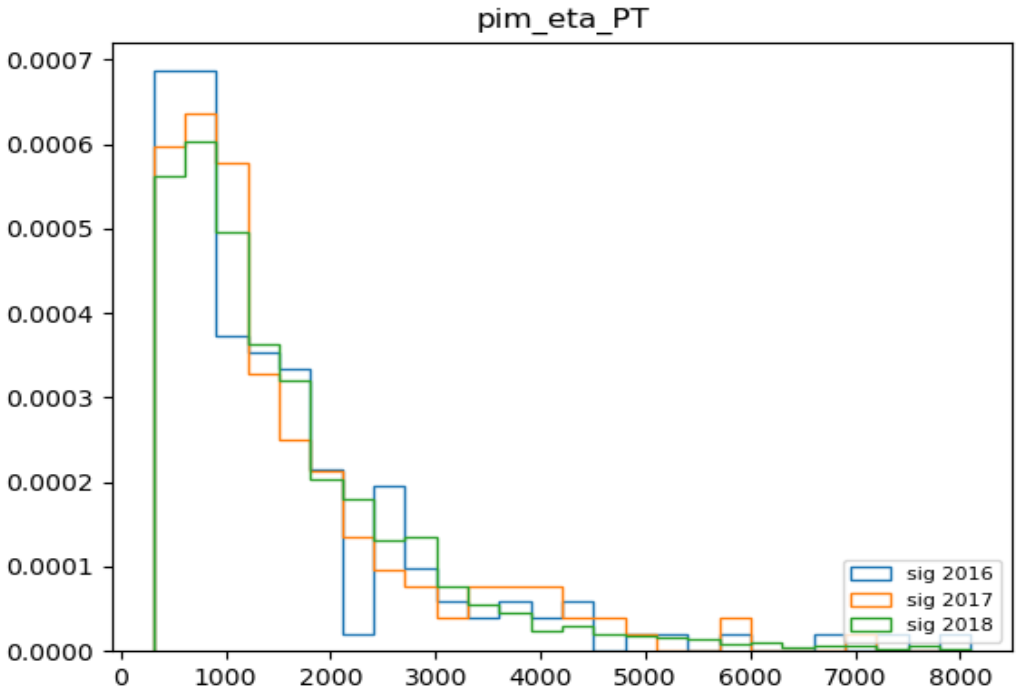


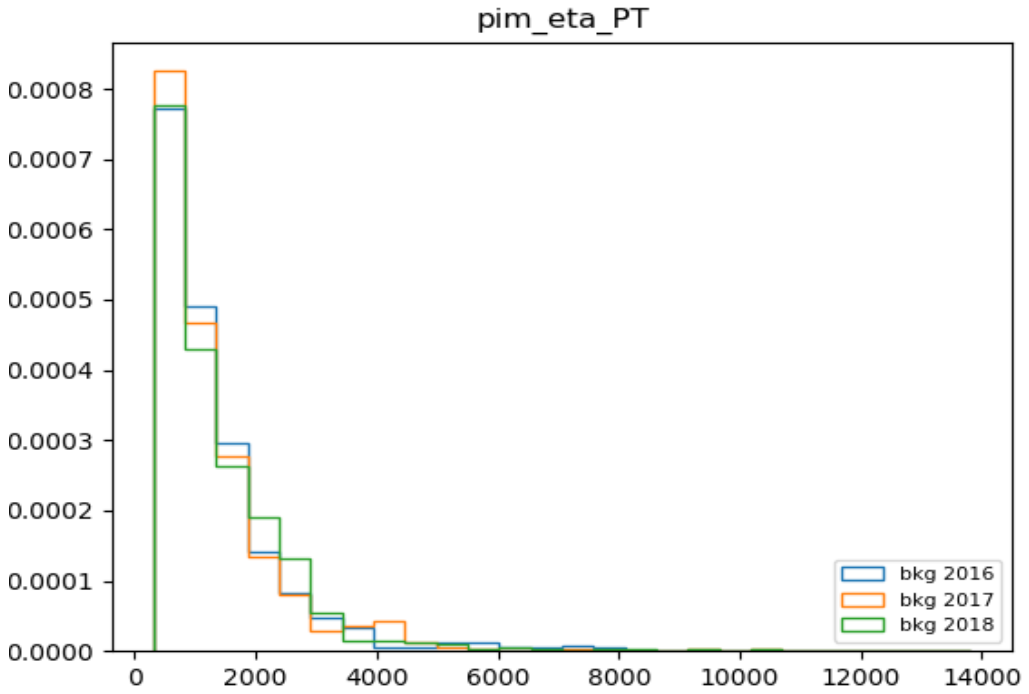
- $Ds \rightarrow \eta' (\rightarrow \mu^+ \mu^-) + \pi$





$Ds \rightarrow \eta' (\rightarrow \pi^+ \pi^- \mu^+ \mu^-) + \pi$



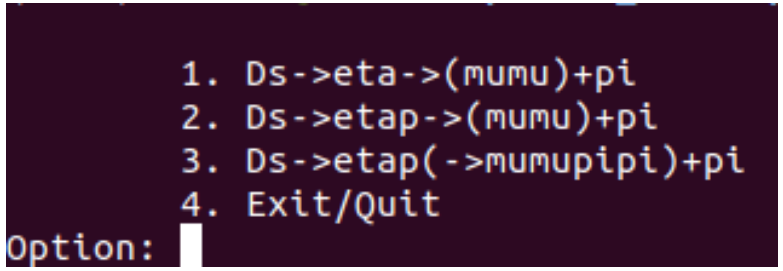


To measure the quality of the results we used same options as before:

1)The **ROC curve**: a plot that confronts the rate of True Positives vs. False Positives

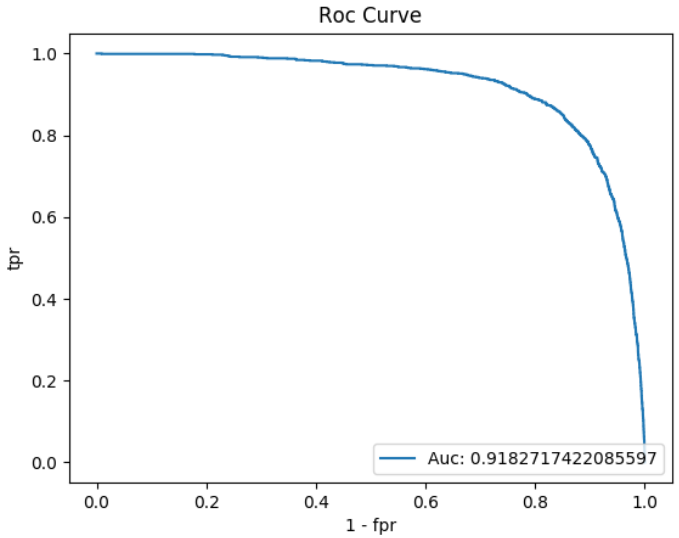
1)The **correlation** between the results and the mass

We got all the results for each analysis and each year (2016, 2017, 2018), in total 9 different classifiers.

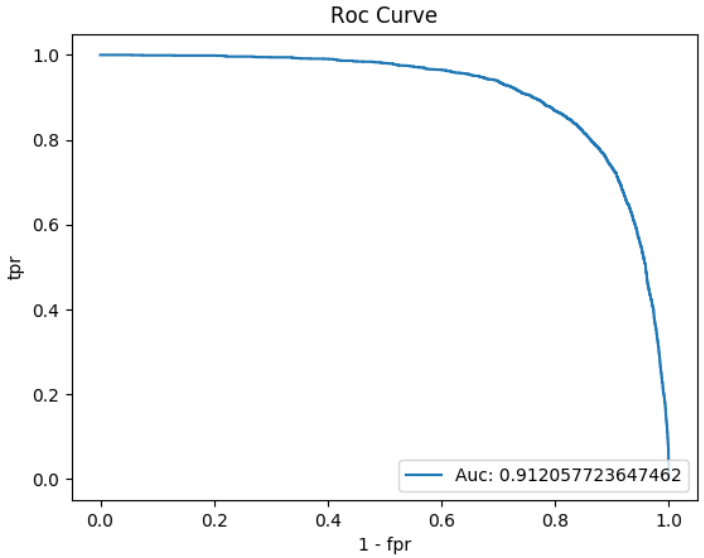


$$Ds \rightarrow \eta (\rightarrow \mu^+ \mu^-) + \pi$$

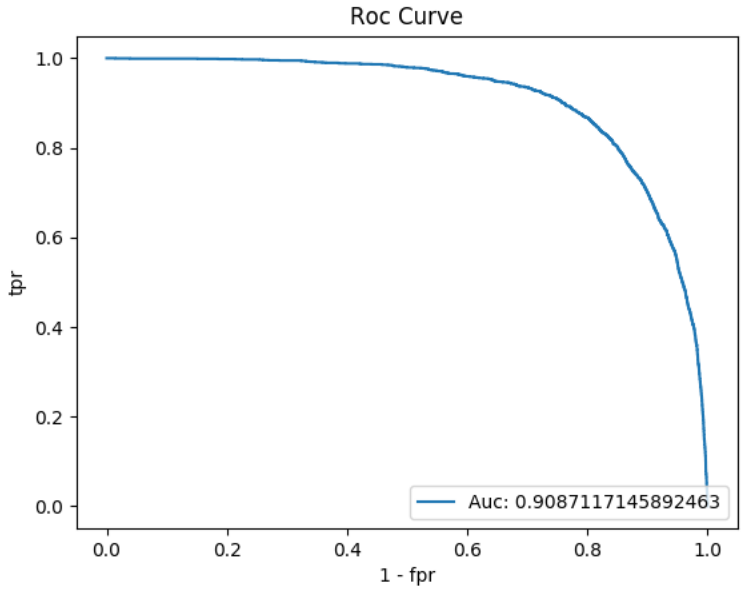
2016



2017

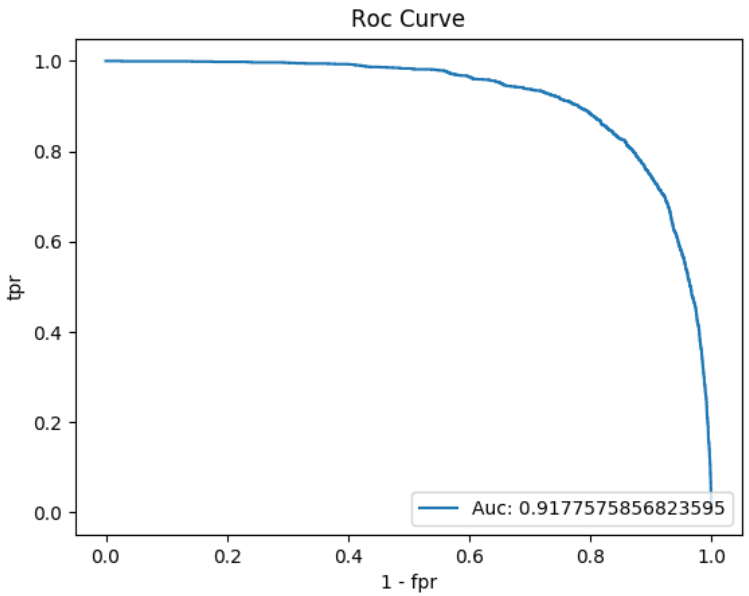


2018

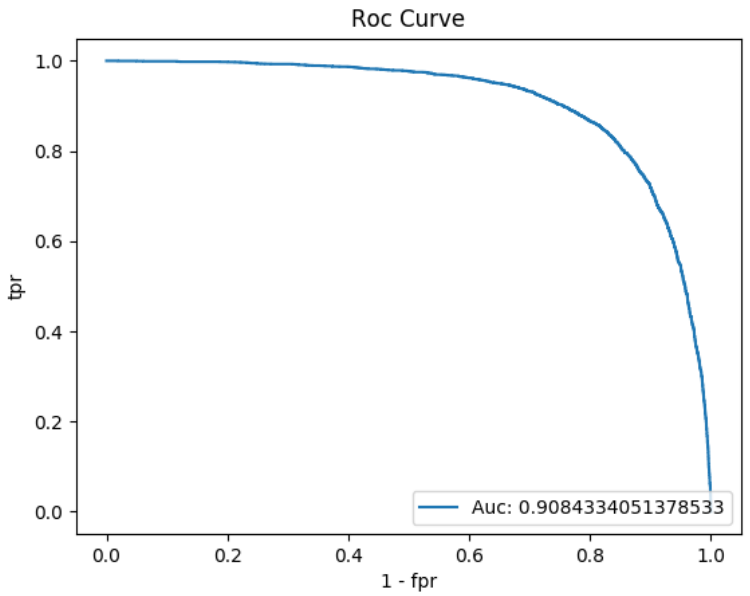


$Ds \rightarrow \eta' (\rightarrow \mu^+ \mu^-) + \pi$

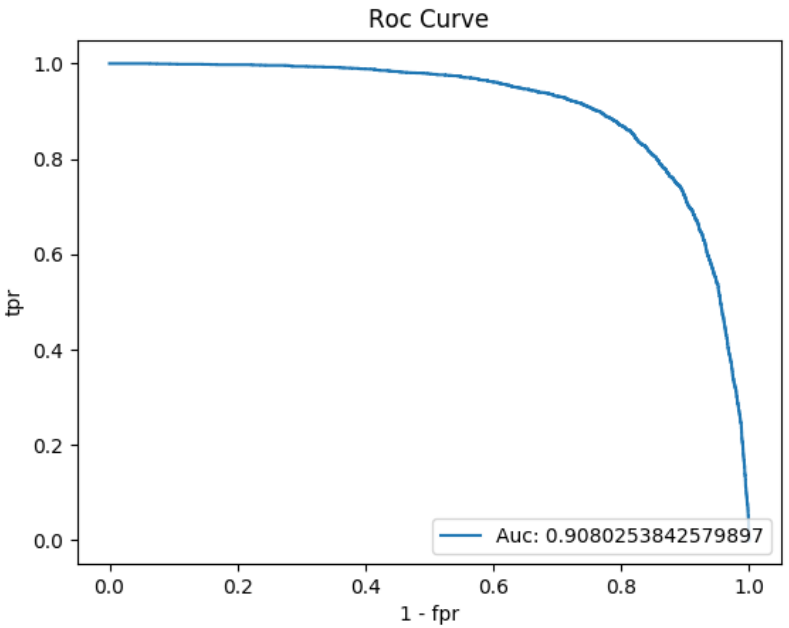
2016



2017

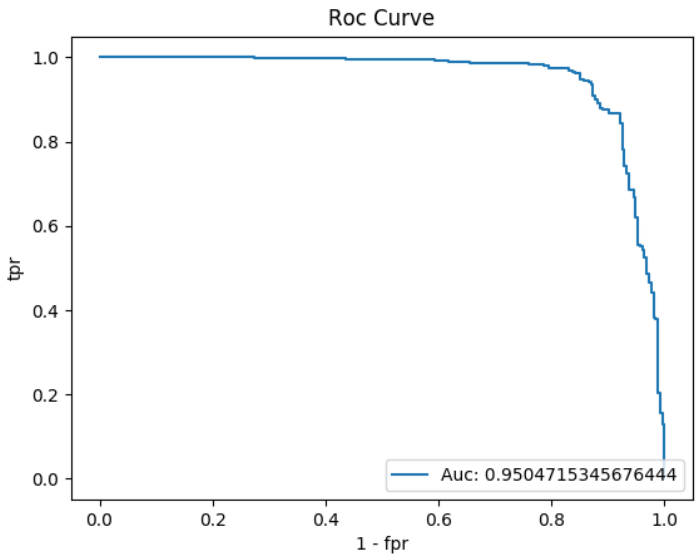


2018

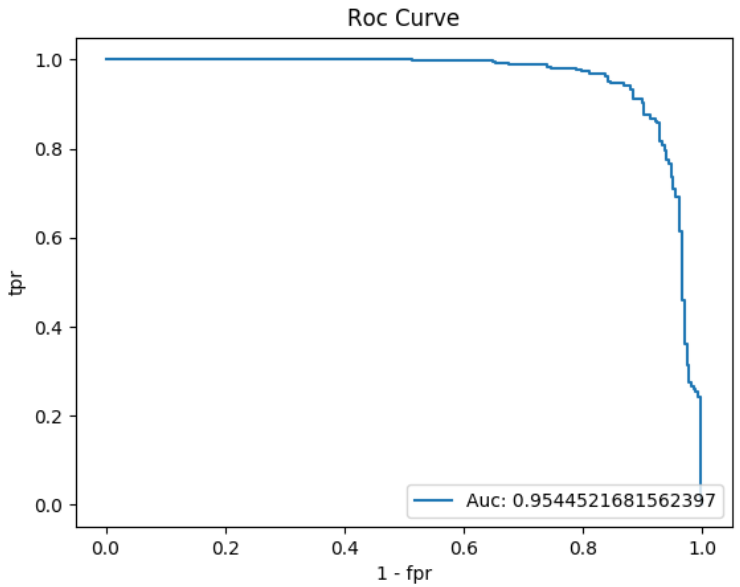


$$Ds \rightarrow \eta' (\rightarrow \pi^+ \pi^- \mu^+ \mu^-) + \pi$$

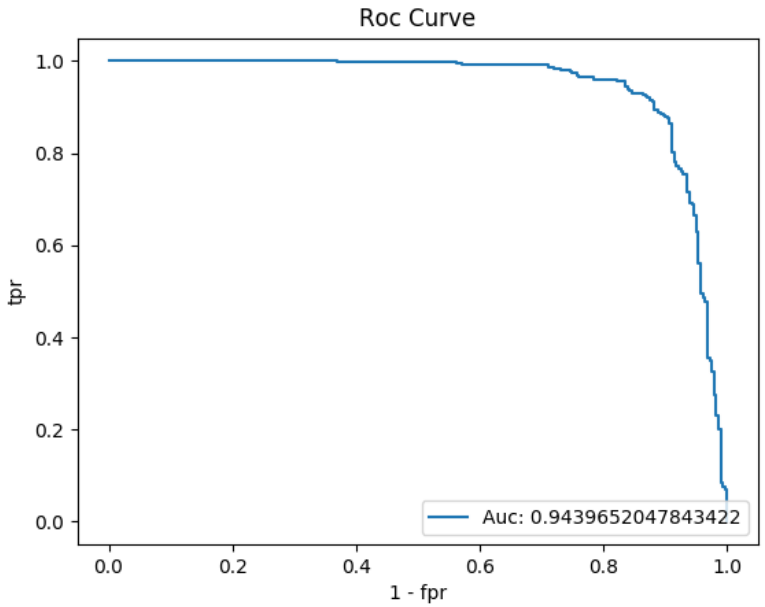
2016



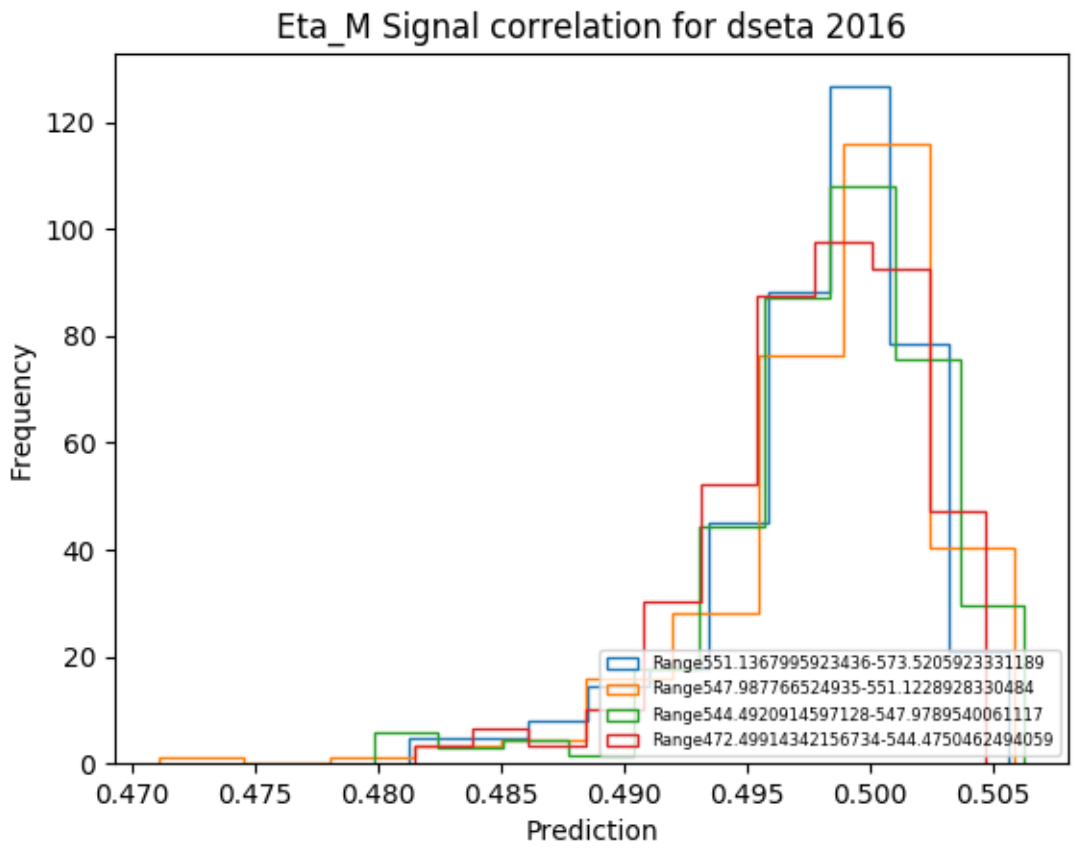
2017

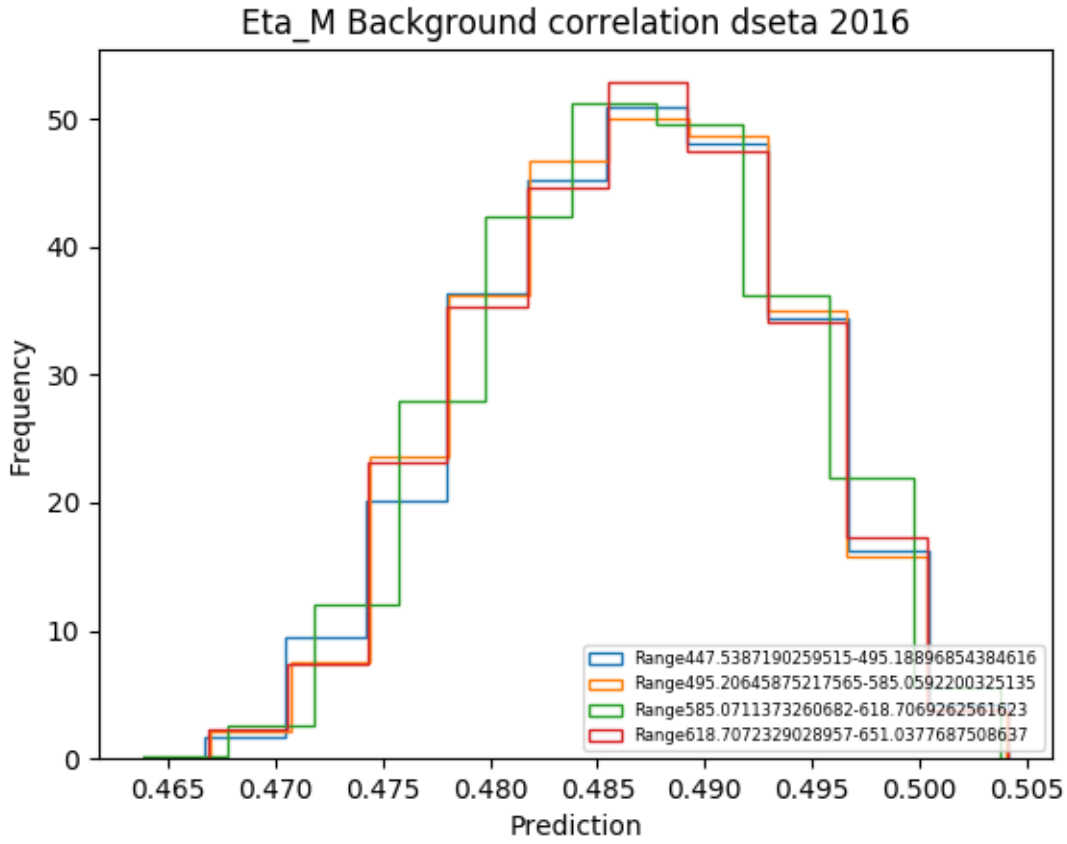


2018

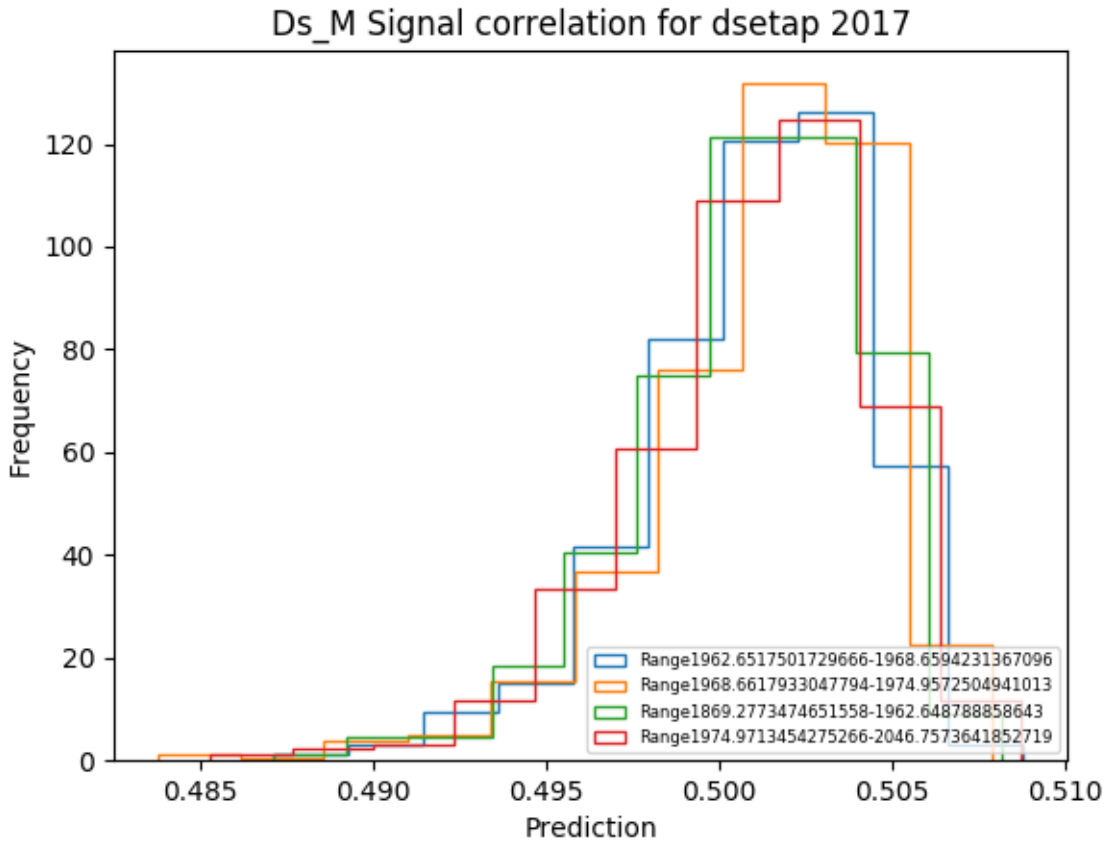


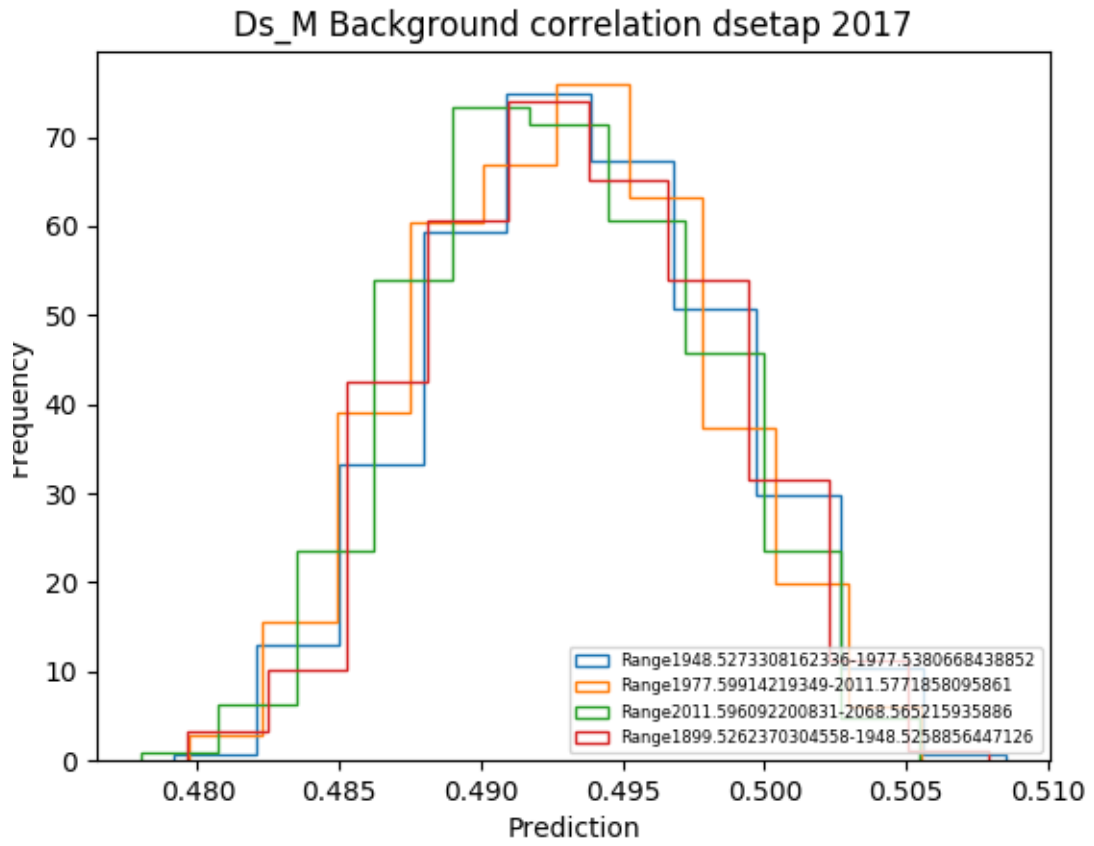
3. Ds→ η (→ $\mu^+ \mu^-$) + π
eta_M 2016



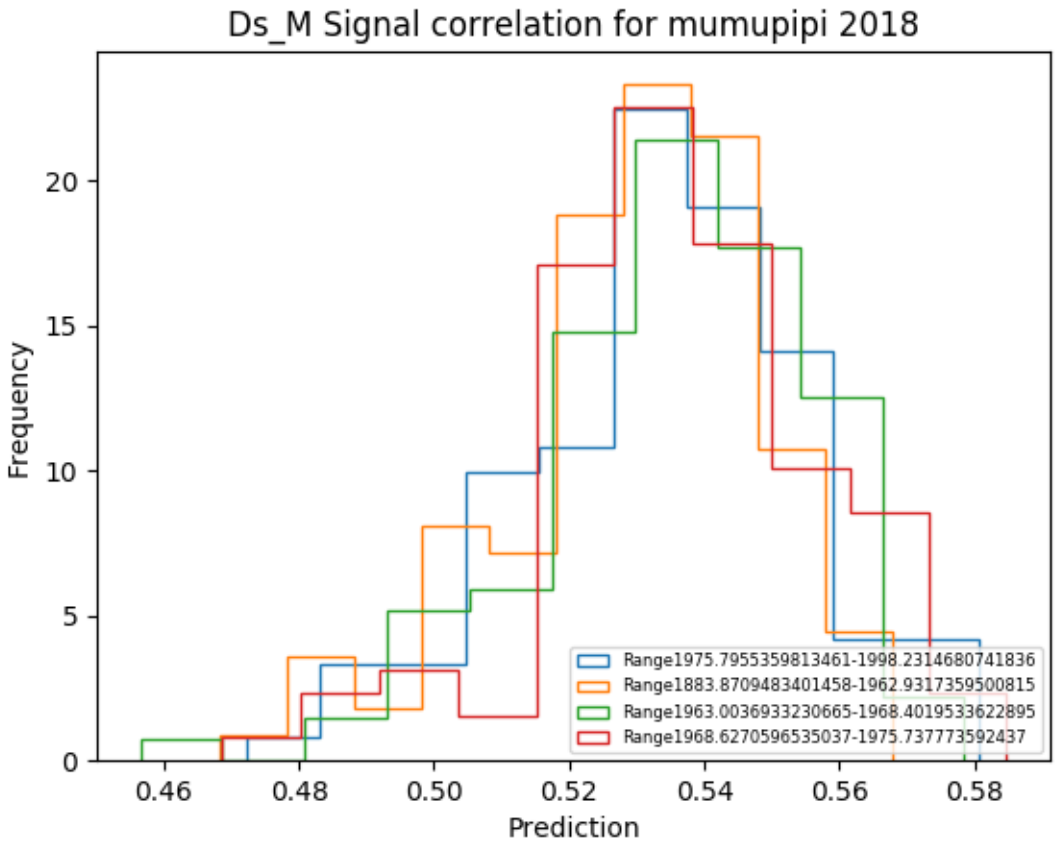


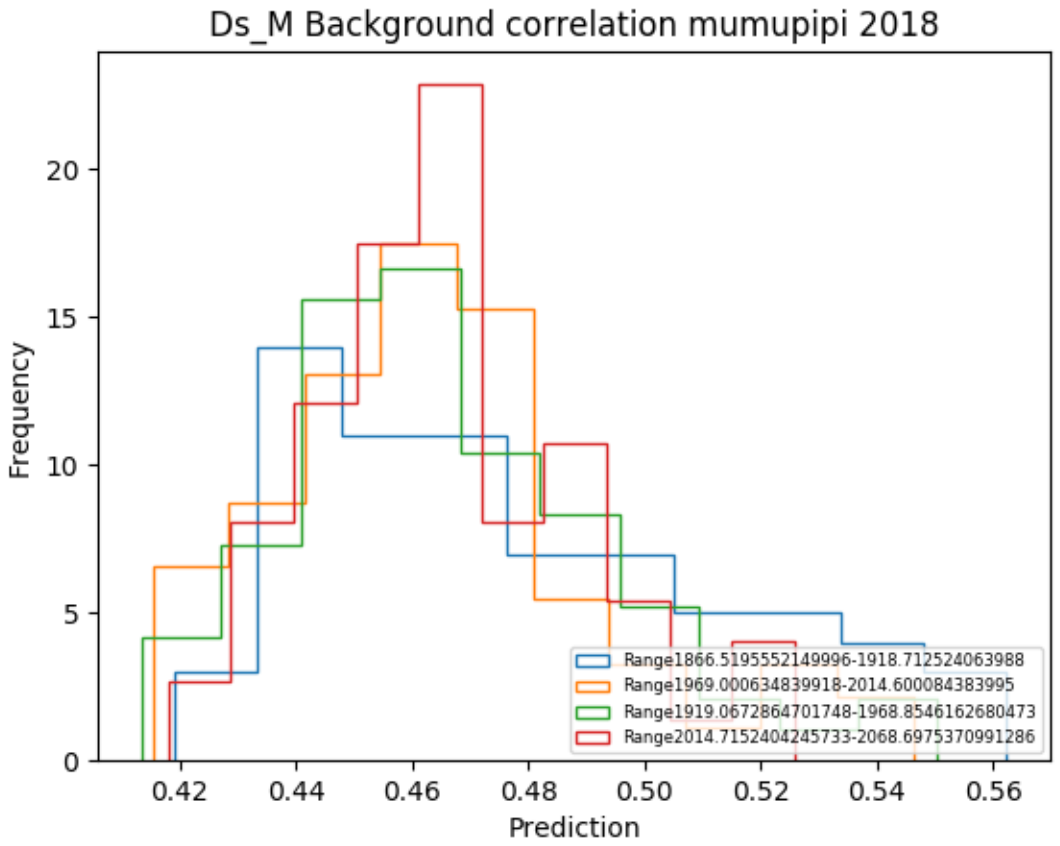
3. Ds→ η' (→ $\mu^+ \mu^-$) + π
Ds_M 2017





3. Ds→ η' (→π⁺ π⁻ μ⁺ μ⁻) + π
Ds_M 2018





Future work

As we showed before the future lines of work are:

1)The necessity of proving with different data that this conclusion is correct and not just a singular occasion in this specific data.

2) Proving why the results under the same speifications work better for our module in scikit-learning Python than our module in TMVA Python.

3) The development of Machine Learning model to solve problems related to the industry.

1) So far we started a project related with a company in the textile sector to detect and solve failures in the system.


TRIPLEALPHA
innovation