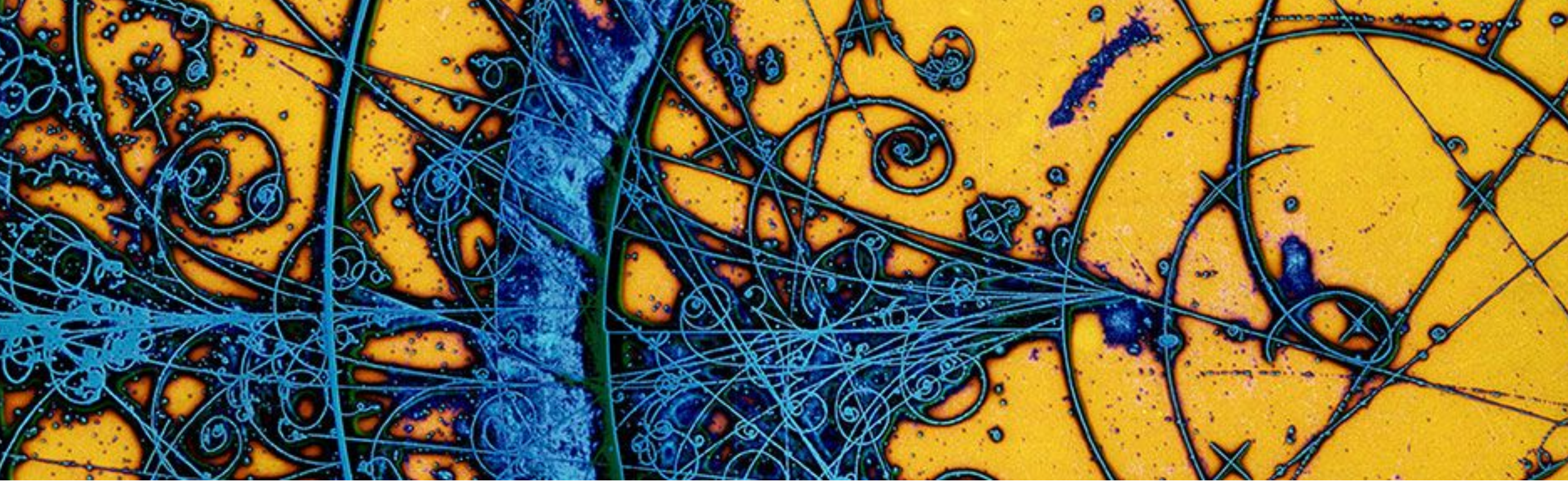


Automated selection of particle-jet features for data analysis in High Energy Physics experiments

deepPP

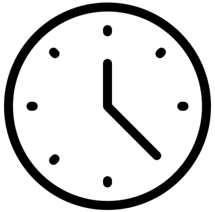
A. Di Luca, F. M. Follega, M. Cristoforetti, R. Iuppa

40th International Conference on High Energy Physics - 28th July 2020

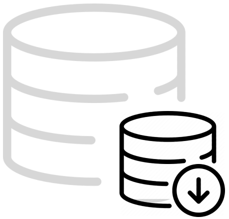
Feature selection for classification problems

When dealing with **new classification problem** in which **machine learning algorithms** will be applied, one of the **crucial** step is **feature selection**.

Reducing unnecessary dimensionality is a key point for:



Time constraints



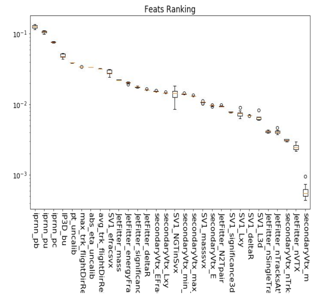
Limited sample size

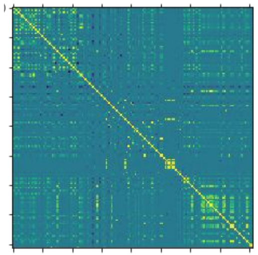
Best understanding

Feature selection for classification problems

A possible approach to select **the most relevant** **non-correlated** features as ranked by a decision tree algorithm.

Yandex CatBoost





Preliminary steps → Feature ranking → Correlated variable removal → Final model

Preliminary steps

Main dataset production

- **Store all the variables** we want to test.
- Apply selections (pT, eta, etc.)
- **Populate** each class **equally**.
- Scale variables in order to limit their range in a smaller one

Standard Scaler	$\dfrac{x_i - \text{mean}(\boldsymbol{x})}{\text{stdev}(\boldsymbol{x})}$
MinMax Scaler	$\dfrac{x_i - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}$

Decision tree hyper parameter tuning

Grid hyperparameter search to optimize **CatBoost** classificator. By doing the obtained feature ranking is relevant for the feature selection.

CATBOOST OPTIMIZATION

```
[15]: model = CatBoostClassifier(loss_function='Logloss',
                                 task_type="GPU",
                                 devices='0:1')
```

```
[16]: grid = {'learning_rate': [0.01, 0.1],
              'depth': [6, 10, 15],
              'l2_leaf_reg': [1, 3, 5, 7, 9]}

      randomized_search_result = model.randomized_search(grid,
                                                         X=X_train,
                                                         y=Y_train,
                                                         plot=True)
```

Feature ranking

A **tool** that could help in **tackling** modern high energy physics **challenges**.

CatBoost

Yandex CatBoost

CatBoost is a machine learning algorithm that uses gradient boosting on decision trees.

- During training, after evaluating the first tree, the weights of those observations that are difficult to classify are increased.

Support GPU training

Random Forest

Random forest is an ensemble learning method for classification or regression.

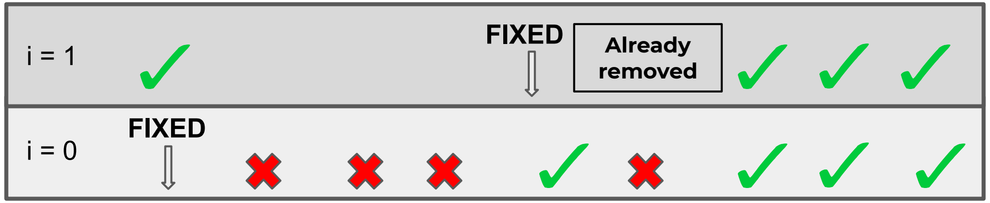
- At training time, multitude of decision trees are evaluated and the output is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Do they share **any common behaviour**?

Correlated variable removal

- We expect to have **highly correlated** variable in **close** positions in feature ranking.
- We define a **threshold** on the correlation value and start **removing variables following the obtained feature ranking.**

  Removing correlated variables **improve the significance** and prevent introducing **undesidered noise.**



	var 1	var 2	. . .						var N
var 1	1	0.97	0.99	0.97	0.92	0.95	0.83	-0.24	-0.78
var 2	0.97	1	0.92	0.94	0.94	0.89	0.79	-0.23	-0.73
.	0.99	0.92	1	0.96	0.87	0.97	0.84	-0.24	-0.79
.	0.97	0.94	0.96	1	0.96	0.98	0.86	-0.26	-0.8
.	0.92	0.94	0.87	0.96	1	0.87	0.81	-0.27	-0.74
.	0.95	0.89	0.97	0.98	0.87	1	0.85	-0.23	-0.8
.	0.83	0.79	0.84	0.86	0.81	0.85	1	-0.21	-0.96
.	-0.24	-0.23	-0.24	-0.26	-0.27	-0.23	-0.21	1	-0.083
var N	-0.78	-0.73	-0.79	-0.8	-0.74	-0.8	-0.96	-0.083	1

Correlation threshold = 0.85

Benchmark application

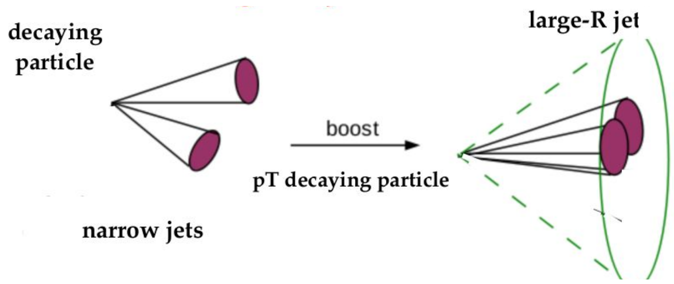
Boosted H→bb tagging

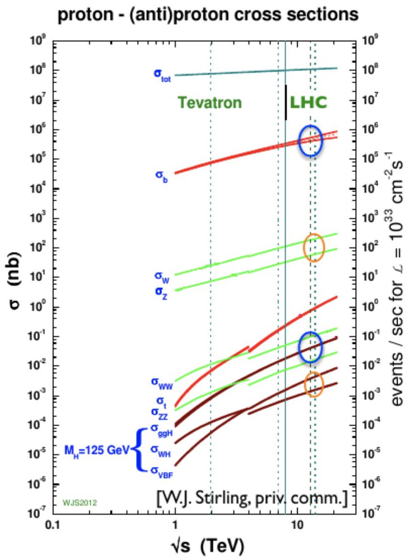
The **H(bb)** channels accounts for **58% of the total Higgs boson decays**.

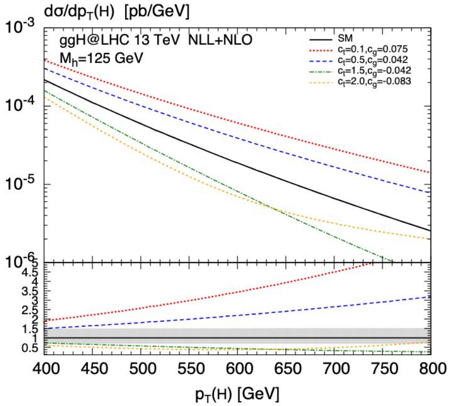
Huge irreducible background coming from QCD multi-jet production in *pp* collisions.

Boosted regime is a nice place where to look for **BSM effects**.

We developed an **H→bb tagger** for *pp* collision experiments based on a deep neural network to identify jets that contains both the b quarks from boosted H decay.







Modeling BSM effects on the Higgs pT spectrum

Pseudo detector simulation

Development of a fast and reliable framework to make a pseudo-experiment.

PYTHIA8
generation of high-energy physics events

+

DELPHES
detector response
fast simulation

RAVE
secondary vertex reconstruction

Microsoft Azure

Run on **Azure VM**
Ubuntu 18.04-LTS
Standard NC6_Promo
[6 vcpus, 56 GiB memory, 1 GPU]

Simulation time

<1 day
> 9M simulated events

Simulated data and object reconstruction

- ATLAS-like detector
- Fast high pT production allowed by having used the following processes in Pythia 8:
  - **Signal:** $q\ g \rightarrow H\ q$
  - **Background:** Hard QCD

Large radius jet (Large R jet)

- Anti-kT jet
- R = 1

Variable radius track jet [arXiv:0903.0392]

- R_{MAX} = 0.4
- R_{MIN} = 0.02
- Rho = 30

Large R jet

Jet substructures	Kinematics variable
Charge	P_T, η, ϕ
Angularity	Kt_ΔR
N-subjettiness	$\Delta\eta$
PlanarFlow	$\Delta\phi$
C2, D2	EhadOverEem
Aplanarity	
ZCut	

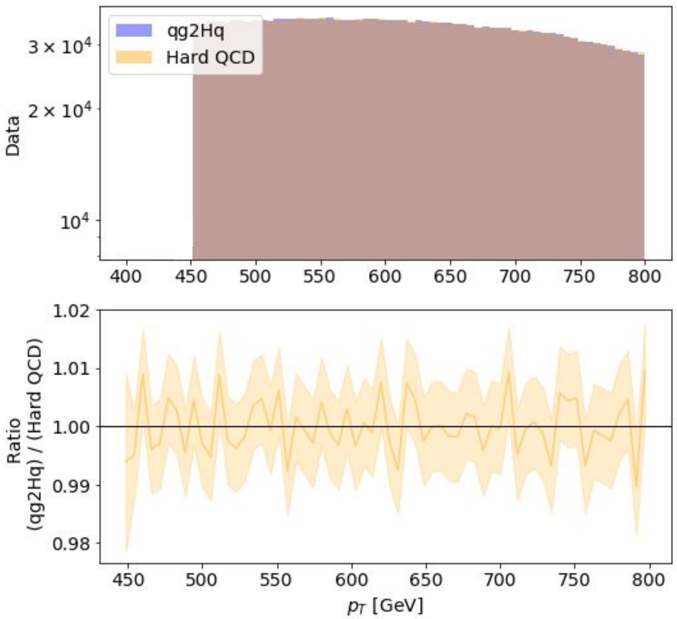
Variable-R track jet

Secondary vertex	Kinematics variable
secvtx_energy	P_T, η, ϕ
secvtx_mass	$\Delta\eta$, $\Delta\phi$
secvtx_sig3D	BTag
secvtx_Lxy	
secvtx_L3D	

Dataset production

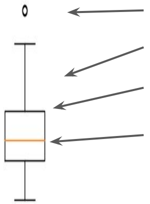
Event selection

- **Large-R jet**
  - p_T > 450 Gev/c^2
  - $|\eta|$ < 2
- **VR track jet**
  - 2 highest p_T contained in Large-R jet

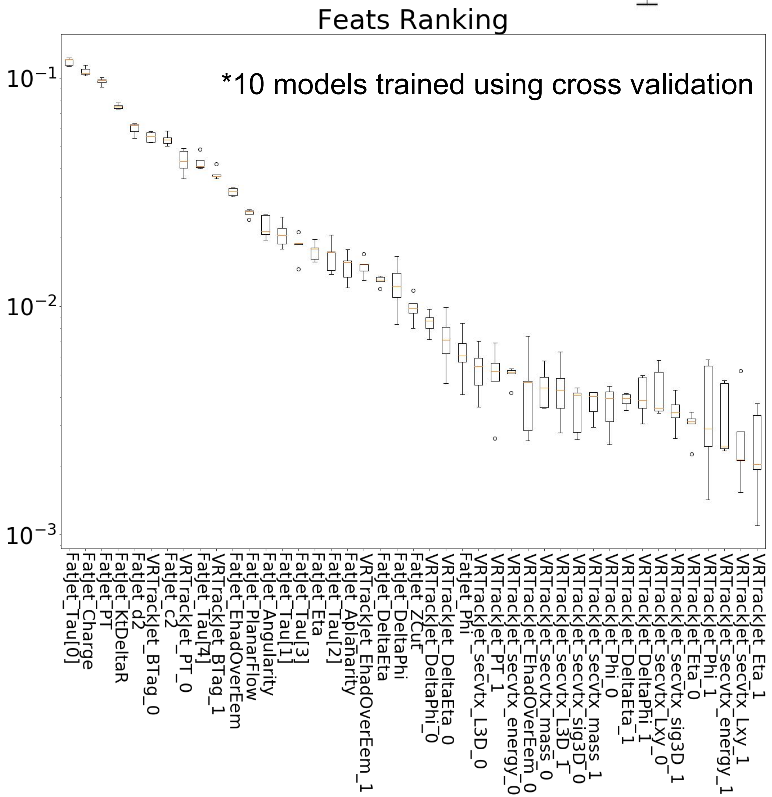


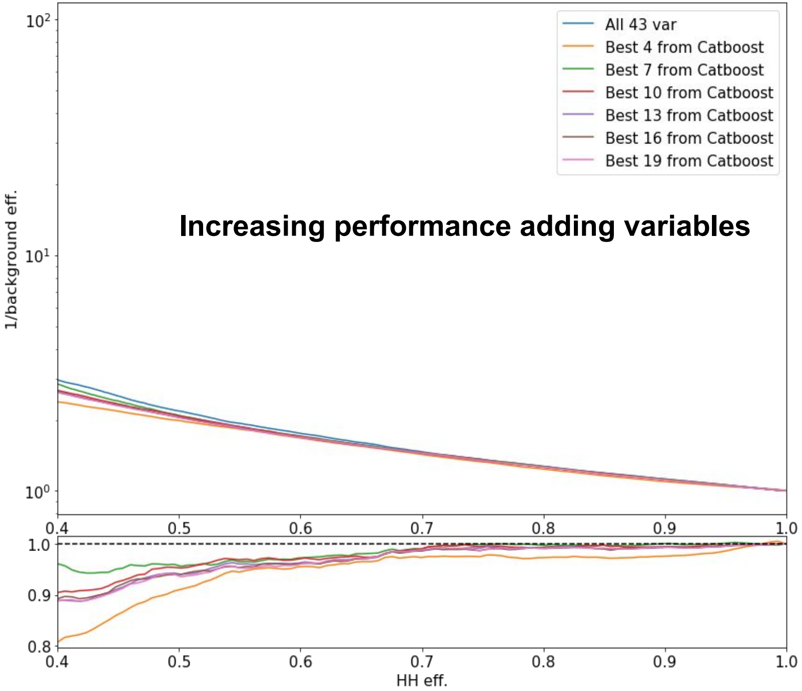
Flat pT spectrum for signal and background large-R jet.

Feature ranking



Yandex CatBoost

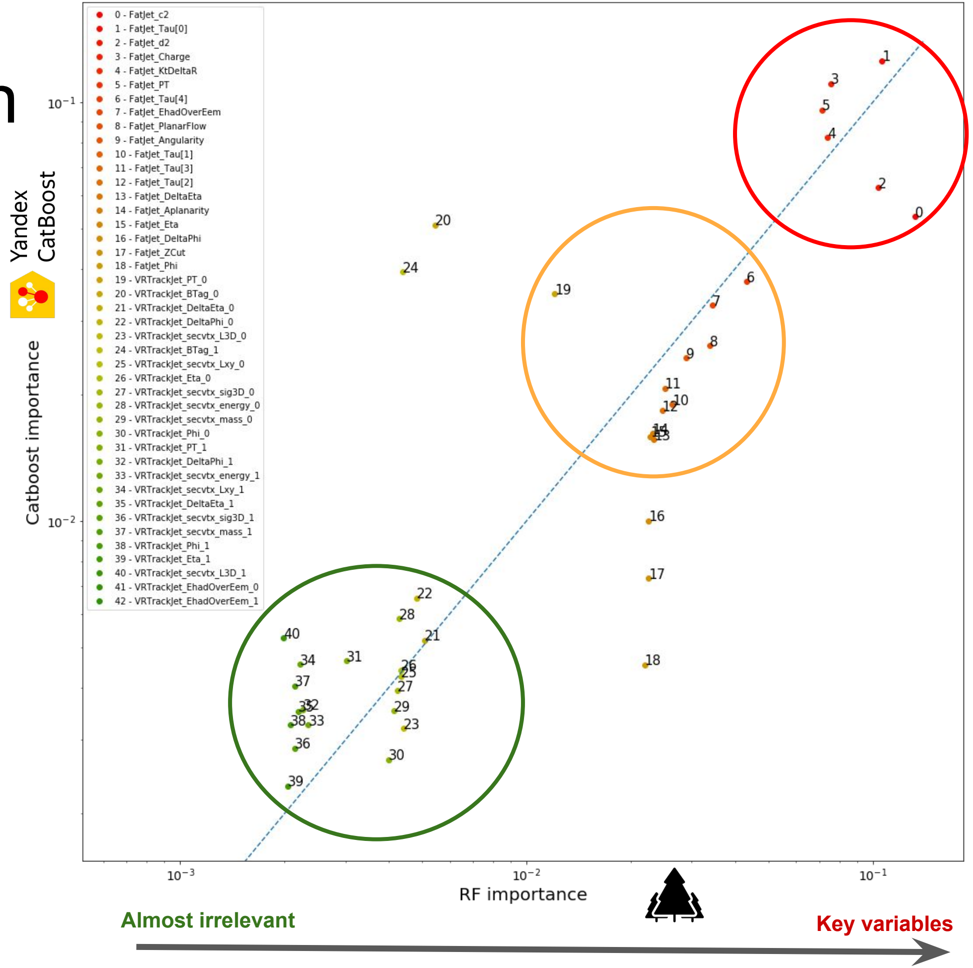




Feature ranking comparison

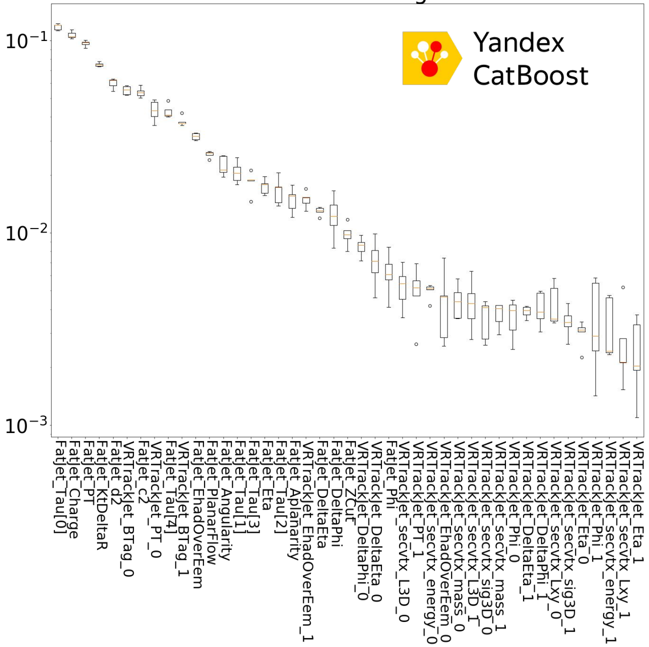
Presence of clusters of variables close to dotted line means compatible feature importance

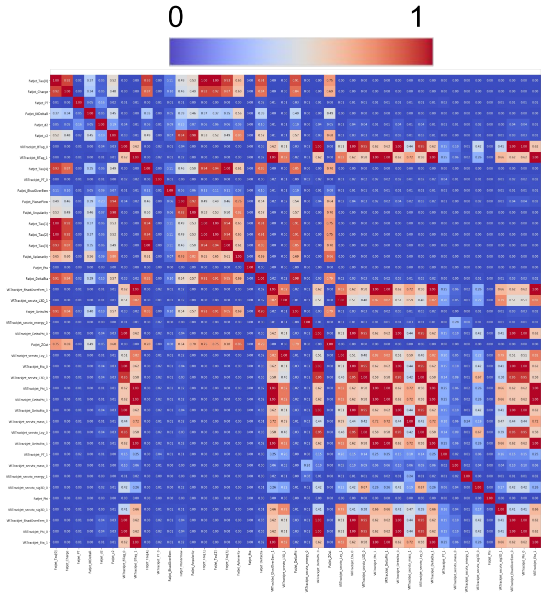
- **KEY** variables for the tagging
- **Almost irrelevant** variables



Correlated variable removal

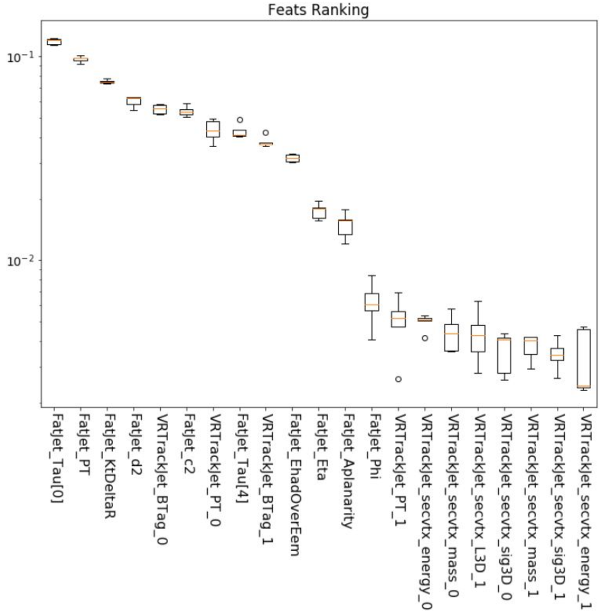
Start: 43 variables



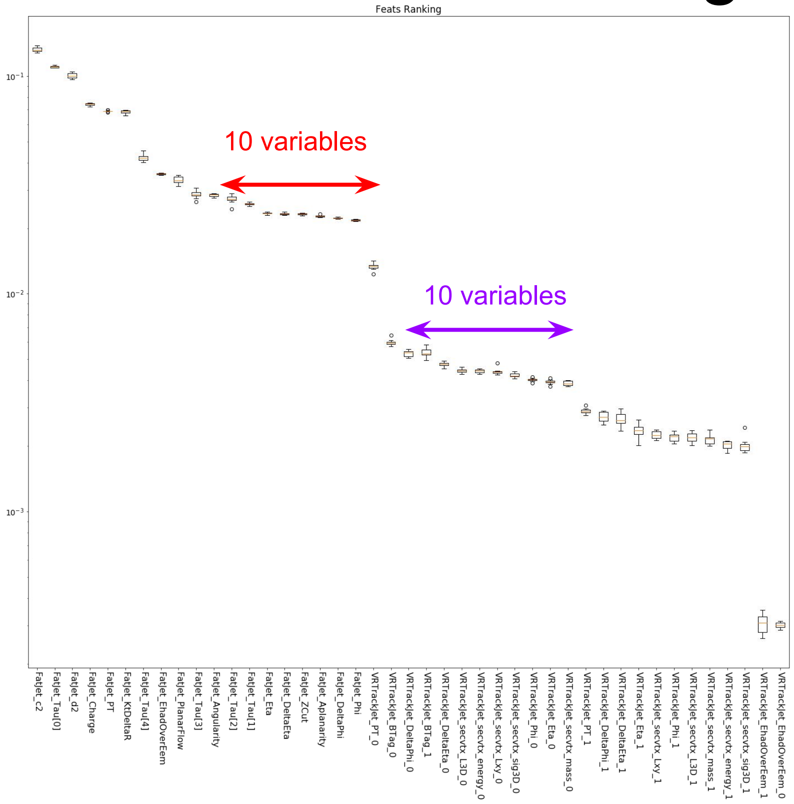


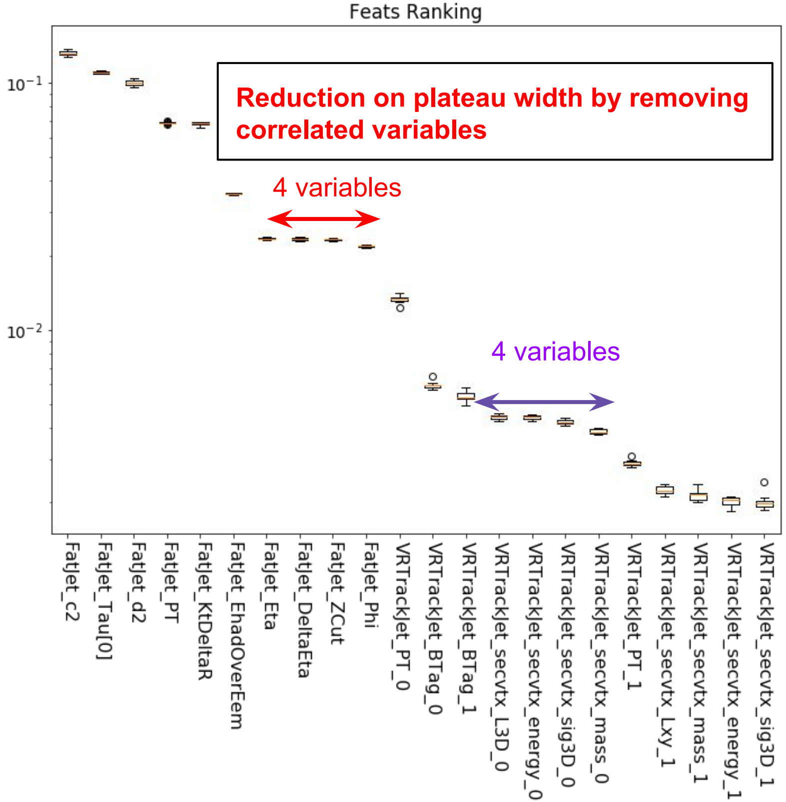
Correlation threshold = 0.85

End: 22 variables



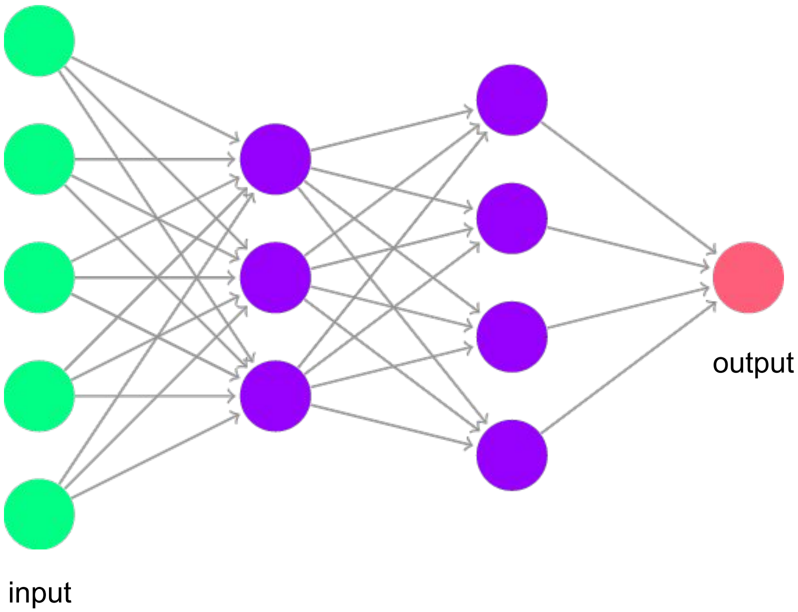
Random forest ranking





Network architecture

- Fully connected layers
  - 6 hidden layers
  - 128 nodes per layer
  - SELU activation function
- Output [0,1]
- Framework: Pytorch
- We kept the **hidden** network architecture constant
- **EarlyStopping** to avoid overfitting



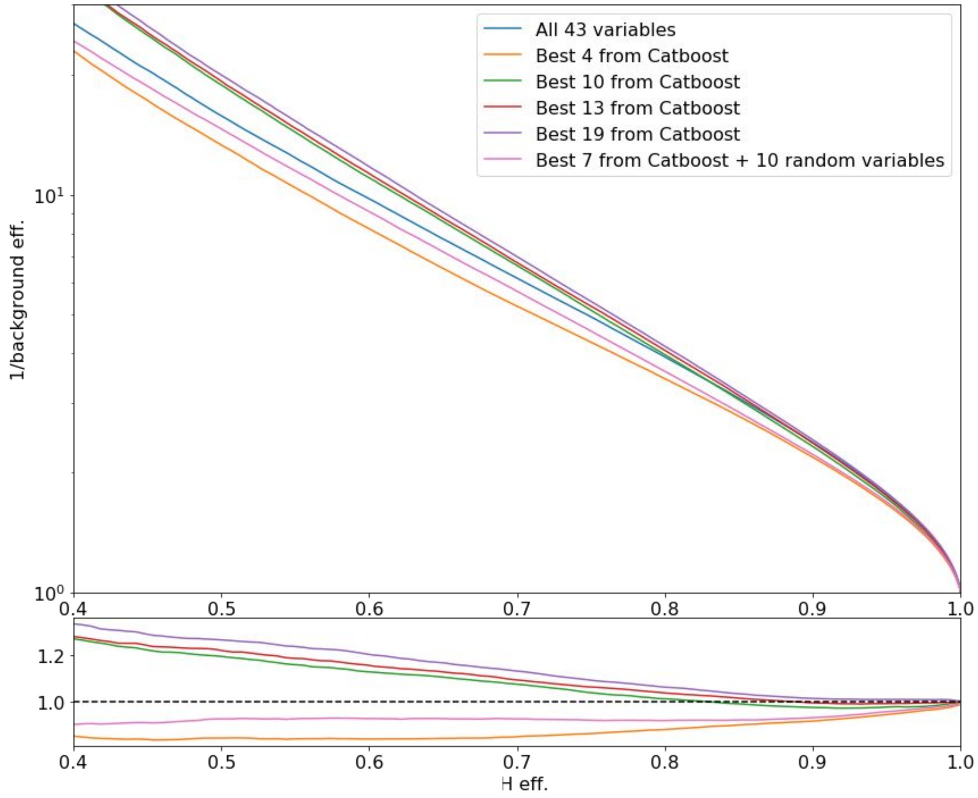
Performance

Performances using the **best 7 variable plus 10 random** variable are **much** lower than other models. Model trained using **top 19 ranked variables** is performing better than the **complete model***

We're using a feature ranking obtained using a **totally different** algorithm.

***Reducing the dimensionality** may well improve the performance, helping the algorithm to **escape accidental local minima** and find the global one. Bugs in vertex parameter definitions are under investigation

*Model re-trained when using new set of variable.

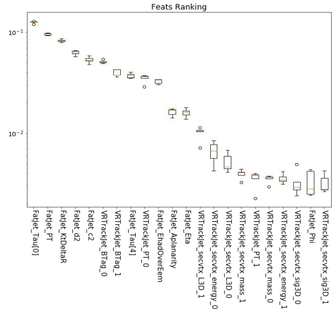


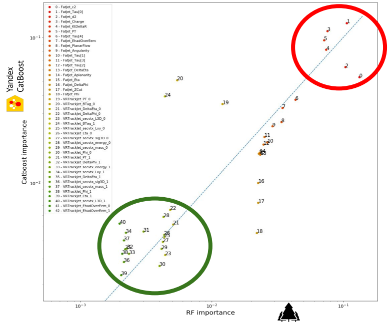
Conclusions

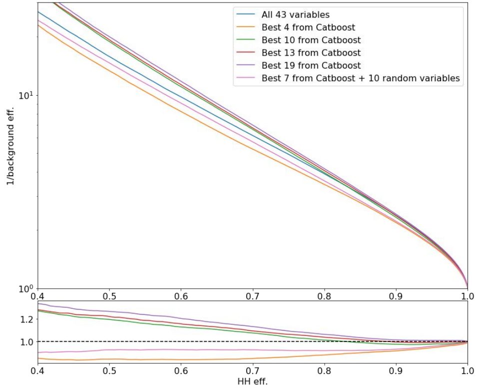
We showed a possible approach for feature selection, containing **the most relevant non-correlated** features as ranked by a decision tree algorithm.

- Both most and less relevant features are compatible using different ranking algorithm.
- Different NN models were trained using different variable combination to highlight the relevance of the feature ranking obtained using a totally different algorithm.

A method to dig into details of event N-tuples of modern high energy physics experiments is under development, with focus on **reducing dimensionality and correlation**.







Backup

Feature ranking

RF ranking

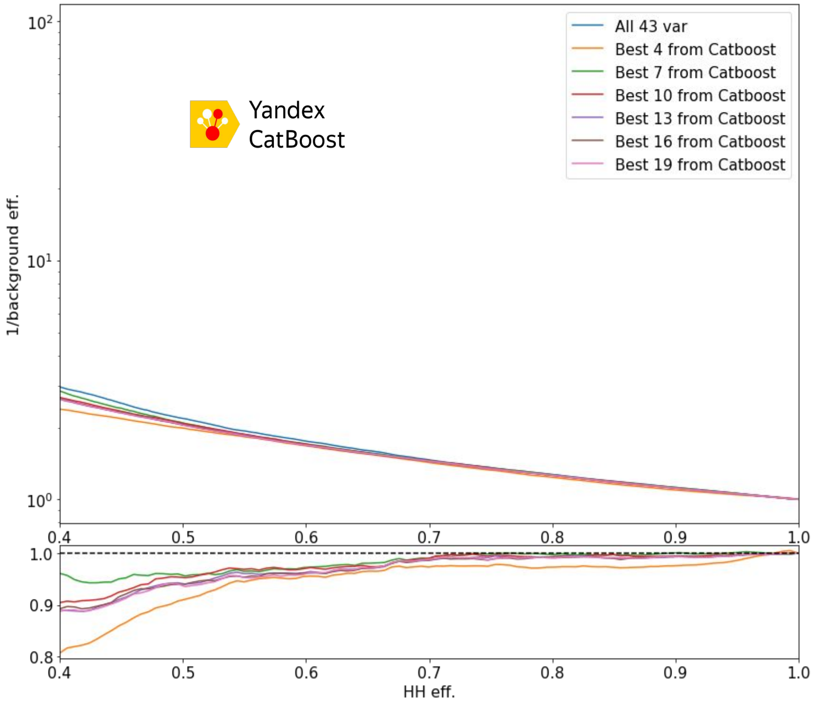
The impurity-based feature importances.

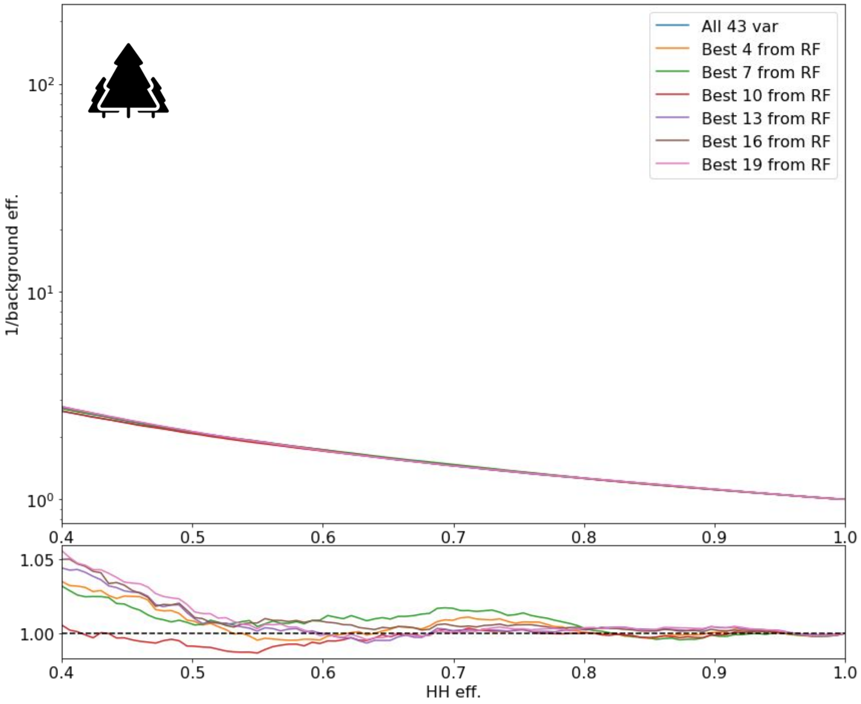
The higher, the more important the feature. The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance.

CatBoost ranking

For each feature, it based on how much on average the prediction changes if the feature value changes. The bigger the value of the importance the bigger on average is the change to the prediction value, if this feature is changed.

CatBoost and RF performances





Selected features