PrimeTagSvc

Jets, samples and Wednesday working meeting

Kaili Zhang

zhangkl@ihep.ac.cn

JOI Variables Ranking

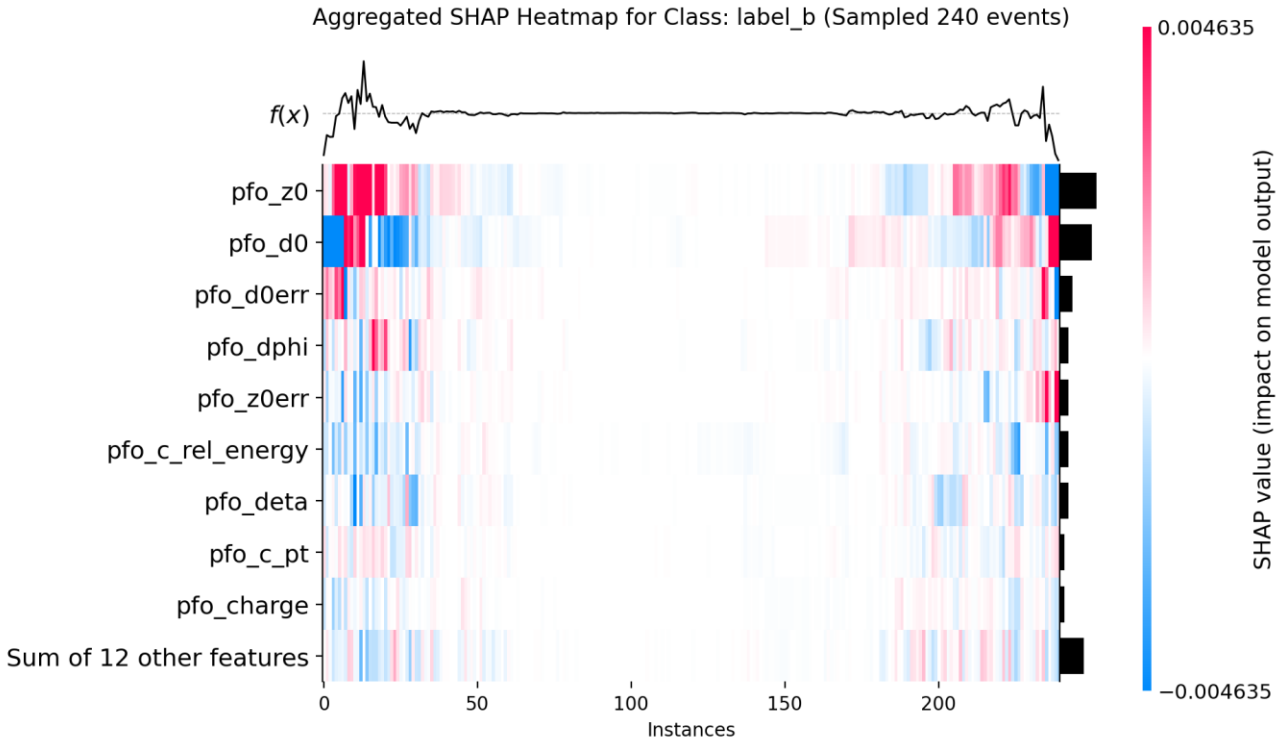
- 2 Methods tested:
  - Attribution: Shapley Value A Unified Approach to Interpreting Model Predictions(2017)
    - Contribution of feature to an individual prediction using game-theoretic Shapley Values. ->Local
  - Permutation: Random Forests(2001)
    - How much does lacking a feature impact global model performance?
    - Measures the drop in global model loss after shuffling a single feature's values. ->Global.

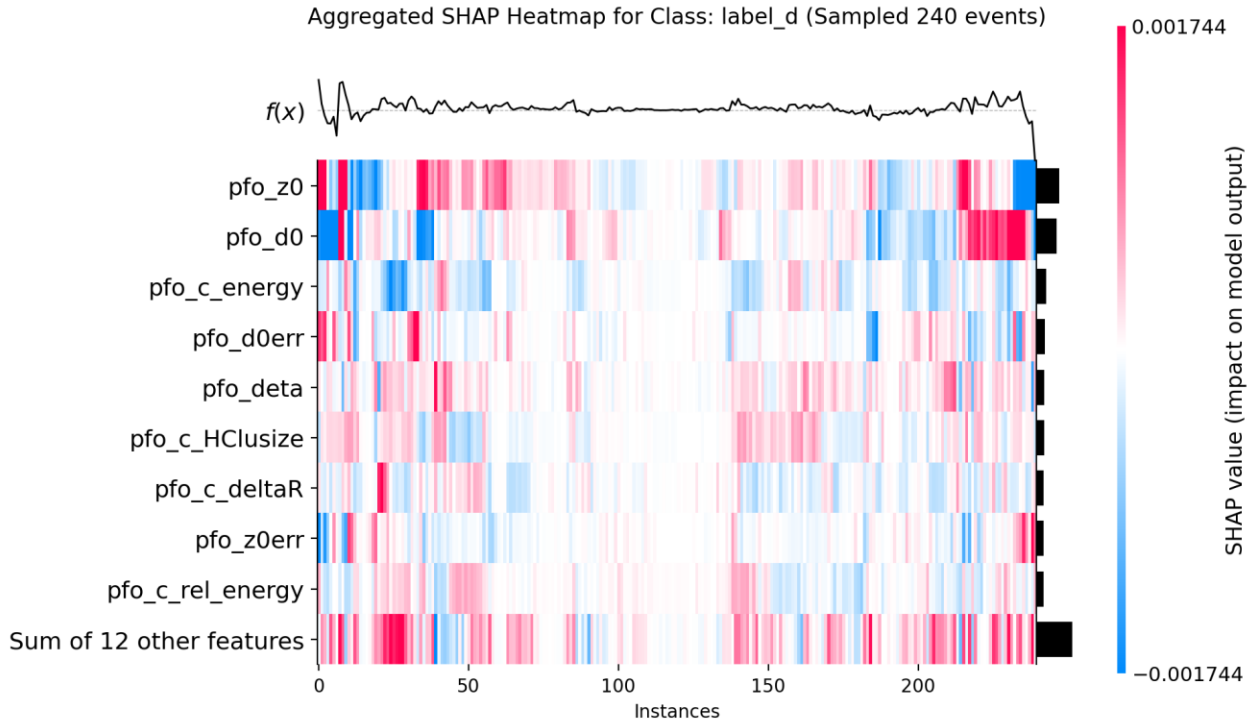
On Shap:

pimohan@ihep.ac.cn / ParT_Ranking · GitLab
Tested on XGBest_RecoID (TDR version).

High shap value: positive importance, making score higher.
Red: Variable itself with high value. Blue: variable itself with low value.
For heavy flavor b tagging and light quark d tagging, their ranking are almost same: impact parameters crucial.

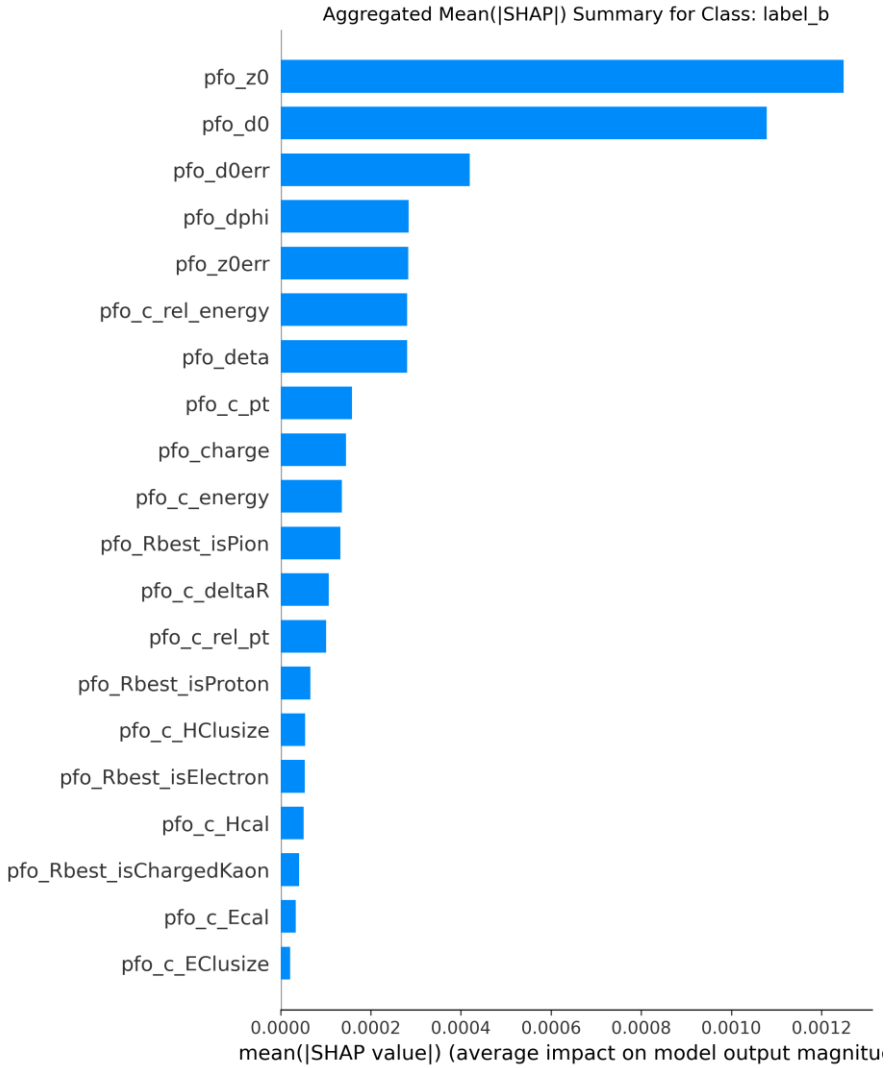
1. d0, z0 crucial. B tagging heavily rely on. (0.004). D tagging used are veto requirement.
2. Absolute importance smaller in d tagging (0.001). Other variables got more importance.

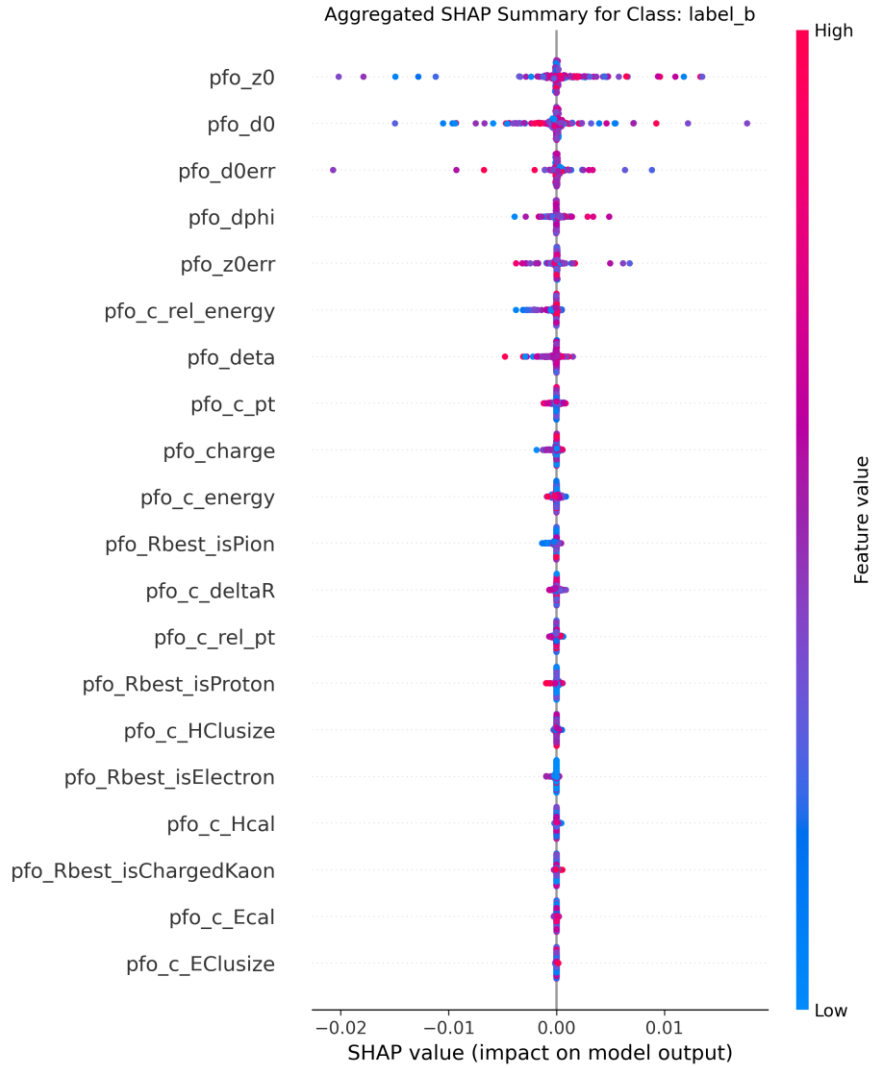




Shap analysis for b tagging

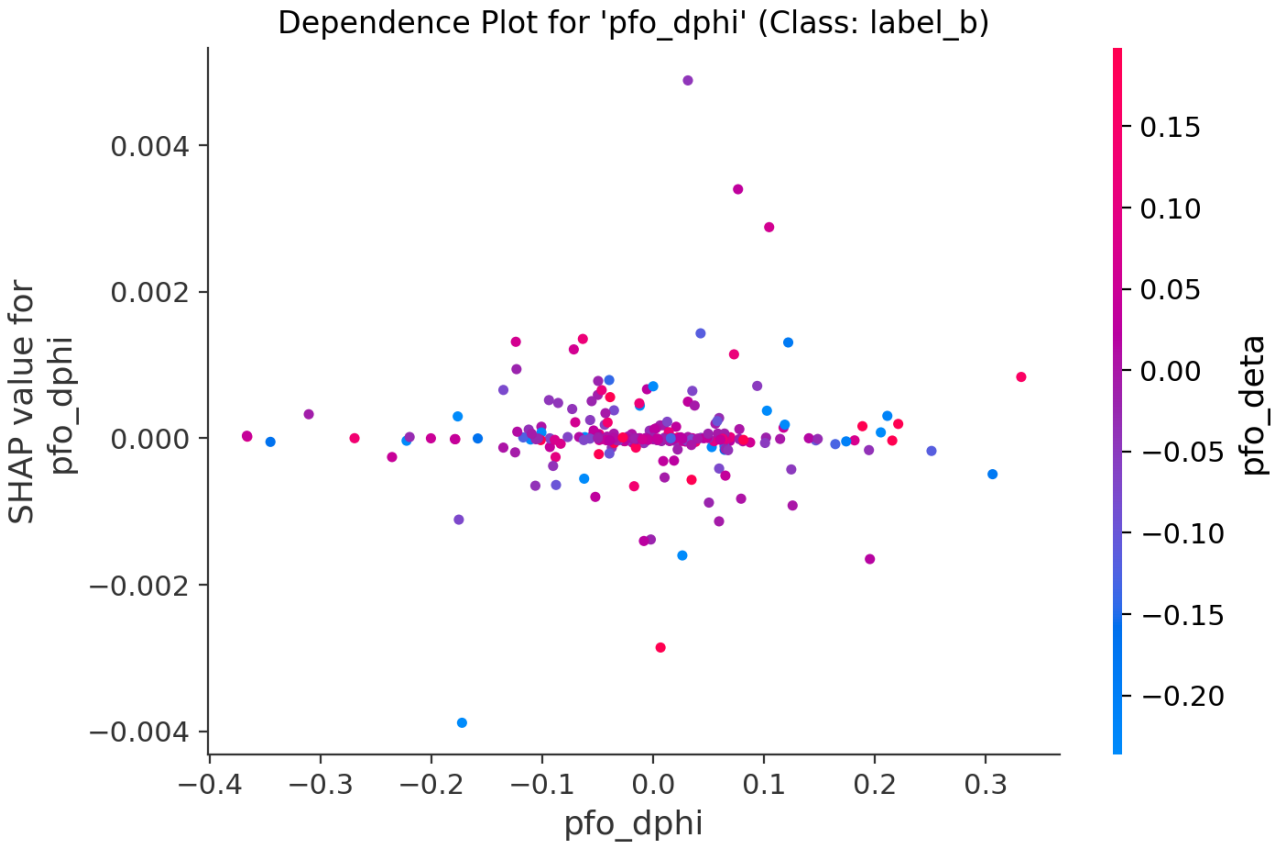
Note: Variables like isMuon, isElectron impact small, due to the overall ratio in one jet (<1%) is small.

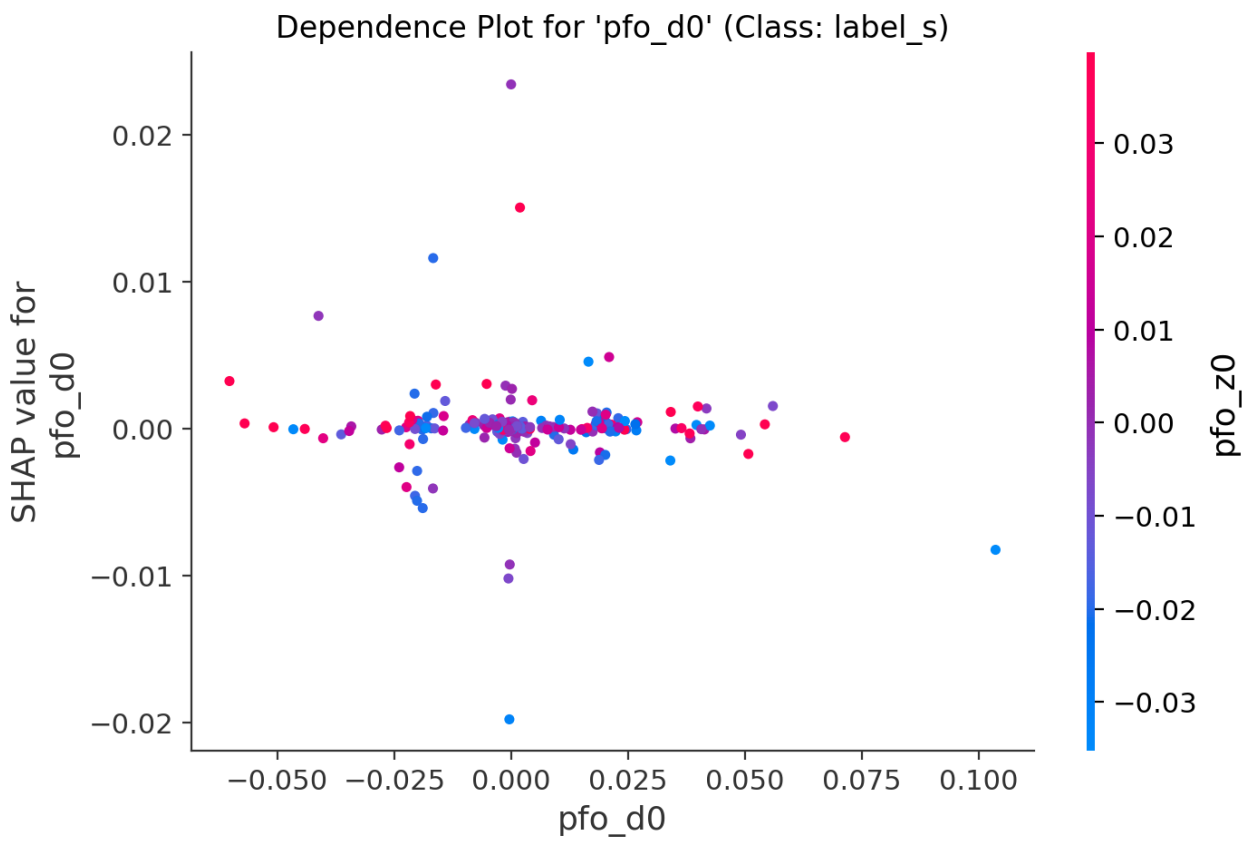




Shap 2d scattering plot

X: dphi value
Y: dphi Shap value (importance)
Z: deta, the most correlated variable value-> What model learned from pair effect.





Permutation ranking:

```
reco id:
 1. pfo_c_energy         : 0.2907
 2. pfo_c_pt             : 0.2273
 3. pfo_charge           : 0.1963
 4. pfo_Rbest_E          : 0.1874
 5. pfo_c_rel_pt         : 0.1786
 6. pfo_c_rel_energy     : 0.1783
 7. pfo_c_deltaR         : 0.1293
 8. pfo_Rbest_isPion     : 0.1173
 9. pfo_z0               : 0.1057
10. pfo_py               : 0.1051
11. pfo_px               : 0.1007
12. pfo_pz               : 0.0970
13. pfo_d0               : 0.0921
14. pfo_dphi             : 0.0818
15. pfo_deta             : 0.0717
16. pfo_Rbest_isChargedKaon: 0.0533
17. pfo_z0err            : 0.0329
18. pfo_d0err            : 0.0322
19. pfo_c_Ecal           : 0.0300
20. pfo_Rbest_isProton   : 0.0251
21. pfo_c_Hcal           : 0.0138
22. pfo_Rbest_isElectron: 0.0124
23. pfo_c_EClusize       : 0.0091
24. pfo_c_HClusize       : 0.0087
25. pfo_Rbest_isMuon     : 0.0073
```

- By shuffle the variable among all event
- Model lose the information in this variable.
- Baseline Loss on test data: 2.0478
- Baseline Accuracy on test data: 0.5576
- Losing Energy_ratio: 2.0478+0.2907
- Information redundancy: E_ratio and Pt_ratio, d0/z0 are correlated.
  - Shuffling A while B variable can do compensation
  - D0, Z0, D0err, Z0err all contribute to vertex information, so diluted.
  - So Permutation method more rely on “unique” features

Conclusion on variable rankings on TDR

- In total 25 features used
  - Impact parameter: D0, Z0, D0err, Z0err crucial. Especially for B/C tagging.
    - As d0/z0 dominates the tagging, it indicates that other variables lacking further development.
  - 4 momentum, and E ratio in jet, give unique information for model.
  - No variable gives negative impact——all should be kept to avoid information loss.
  - Conclusion is universal no matter using one true label sample only or merged sample.

Extension on ranking:

1.	c_IPSig	: 0.2086
2.	logE_ratio	: 0.2027
3.	d0err	: 0.1891
4.	z0err	: 0.1737
5.	rho_d0_phi	: 0.1572
6.	pz	: 0.1478
7.	py	: 0.1475
8.	px	: 0.1472
9.	E	: 0.1313
10.	c_IParea	: 0.1286
11.	Angle	: 0.1170
12.	R_PPhoton	: 0.0969
13.	c_z0err	: 0.0959
14.	d0	: 0.0830
15.	z0_sig	: 0.0826
16.	c_HcalL	: 0.0821
17.	d0_sig	: 0.0807
18.	Jade	: 0.0694
19.	R_PKaon	: 0.0588
20.	DR	: 0.0452
21.	pterr	: 0.0440
22.	logPt_ratio	: 0.0359
23.	c_d0err	: 0.0331
24.	R_PPion	: 0.0291
25.	Jm	: 0.0289
26.	EcalL	: 0.0221
27.	c_HClusize	: 0.0151
28.	Hcal_T0	: 0.0147
29.	rel_Eboost	: 0.0137
30.	Ecal_ratio	: 0.0128
31.	dphi	: 0.0115
32.	c_Dndx	: 0.0110
33.	Entropy	: 0.0104
34.	c_ecal_weta	: 0.0102
35.	Hcalweta	: 0.0082
36.	deta	: 0.0077

36.	deta	: 0.0077
37.	Hcal_T1	: 0.0075
38.	PID_entropy	: 0.0065
39.	pos	: 0.0047
40.	E_ratio	: 0.0045
41.	HcalL	: 0.0043
42.	c_HcalR90	: 0.0024
43.	R_PElectron	: 0.0022
44.	HcalR90	: 0.0019
45.	omega_sig	: 0.0014
46.	c_d0z0err	: 0.0014
47.	PID_max	: 0.0010
48.	rho_d0_z0	: 0.0007
49.	Ecalweta	: 0.0006
50.	PID_gap	: 0.0005
51.	c_ecal_wphi	: 0.0005
52.	z0	: 0.0005
53.	charge	: 0.0002
54.	Jy	: 0.0001
55.	Jx	: 0.0000
56.	Pt_ratio	: 0.0000
57.	Ecal_ecc	: -0.0000
58.	Ecalwphi	: -0.0000
59.	Ecalwtotal	: -0.0000
60.	Hcalwtotal	: -0.0000
61.	c_hcal_wphi	: -0.0000
62.	c_TOF	: -0.0001
63.	R_PProton	: -0.0002
64.	Jz	: -0.0002
65.	R_PMuon	: -0.0003
66.	Hcal_ecc	: -0.0003
67.	c_EClusize	: -0.0003
68.	c_hcal_weta	: -0.0005
69.	Hcalwphi	: -0.0006
70.	c_EcalR90	: -0.0008
71.	EcalR90	: -0.0012
72.	c_EcalL	: -0.0056
73.	wTrk	: -0.0678

https://code.ihep.ac.cn/zhangkl/jetorigin/-/blob/JetLevel/src/JetOrigin.cpp?ref_type=heads variable definitions can be found.

Overall 73 variables input in model.
The most important variable goes to impact parameter significance:

$$IP_{significance}\sqrt{\left(\frac{d_0}{d_0^{err}}\right)^2+\left(\frac{z_0}{z_0^{err}}\right)^2}$$

Variable z0, as it can be replaced, 52nd.
Furthermore, serval parameters gives negative impact.

However, it learnes fast in the first epoch:

Epoch #0: Current validation metric: 0.51043 (best: 0.51043)
Epoch #12: Current validation metric: 0.55340 (best: 0.55340)
Epoch #29: Current validation metric: 0.55925 (best: 0.55997)

Extension on ranking 2:

1.	c_IPSig	: 0.2086
2.	logE_ratio	: 0.2027
3.	d0err	: 0.1891
4.	z0err	: 0.1737
5.	rho_d0_phi	: 0.1572
6.	pz	: 0.1478
7.	py	: 0.1475
8.	px	: 0.1472
9.	E	: 0.1313
10.	c_IParea	: 0.1286
11.	Angle	: 0.1170
12.	R_PPhoton	: 0.0969
13.	c_z0err	: 0.0959
14.	d0	: 0.0830
15.	z0_sig	: 0.0826
16.	c_HcalL	: 0.0821
17.	d0_sig	: 0.0807
18.	Jade	: 0.0694
19.	R_PKaon	: 0.0588
20.	DR	: 0.0452
21.	pterr	: 0.0440
22.	logPt_ratio	: 0.0359
23.	c_d0err	: 0.0331
24.	R_PPion	: 0.0291
25.	Jm	: 0.0289
26.	EcalL	: 0.0221
27.	c_HClusize	: 0.0151
28.	Hcal_T0	: 0.0147
29.	rel_Eboost	: 0.0137
30.	Ecal_ratio	: 0.0128
31.	dphi	: 0.0115
32.	c_Dndx	: 0.0110
33.	Entropy	: 0.0104
34.	c_ecal_weta	: 0.0102
35.	Hcalweta	: 0.0082
36.	deta	: 0.0077

36.	deta	: 0.0077
37.	Hcal_T1	: 0.0075
38.	PID_entropy	: 0.0065
39.	pos	: 0.0047
40.	E_ratio	: 0.0045
41.	HcalL	: 0.0043
42.	c_HcalR90	: 0.0024
43.	R_PElectron	: 0.0022
44.	HcalR90	: 0.0019
45.	omega_sig	: 0.0014
46.	c_d0z0err	: 0.0014
47.	PID_max	: 0.0010
48.	rho_d0_z0	: 0.0007
49.	Ecalweta	: 0.0006
50.	PID_gap	: 0.0005
51.	c_ecal_wphi	: 0.0005
52.	z0	: 0.0005
53.	charge	: 0.0002
54.	Jy	: 0.0001
55.	Jx	: 0.0000
56.	Pt_ratio	: 0.0000
57.	Ecal_ecc	: -0.0000
58.	Ecalwphi	: -0.0000
59.	Ecalwtotal	: -0.0000
60.	Hcalwtotal	: -0.0000
61.	c_hcal_wphi	: -0.0000
62.	c_TOF	: -0.0001
63.	R_PProton	: -0.0002
64.	Jz	: -0.0002
65.	R_PMuon	: -0.0003
66.	Hcal_ecc	: -0.0003
67.	c_ECLusize	: -0.0003
68.	c_hcal_weta	: -0.0005
69.	Hcalwphi	: -0.0006
70.	c_EcalR90	: -0.0008
71.	EcalR90	: -0.0012
72.	c_EcalL	: -0.0056
73.	wTrk	: -0.0678

https://code.ihep.ac.cn/zhangkl/jetorigin/-/blob/JetLevel/src/JetOrigin.cpp?ref_type=heads variable definitions can be found.

Then, one version with less variable (35) tested.

Now, it learns slowly in the first epoch, but finally:

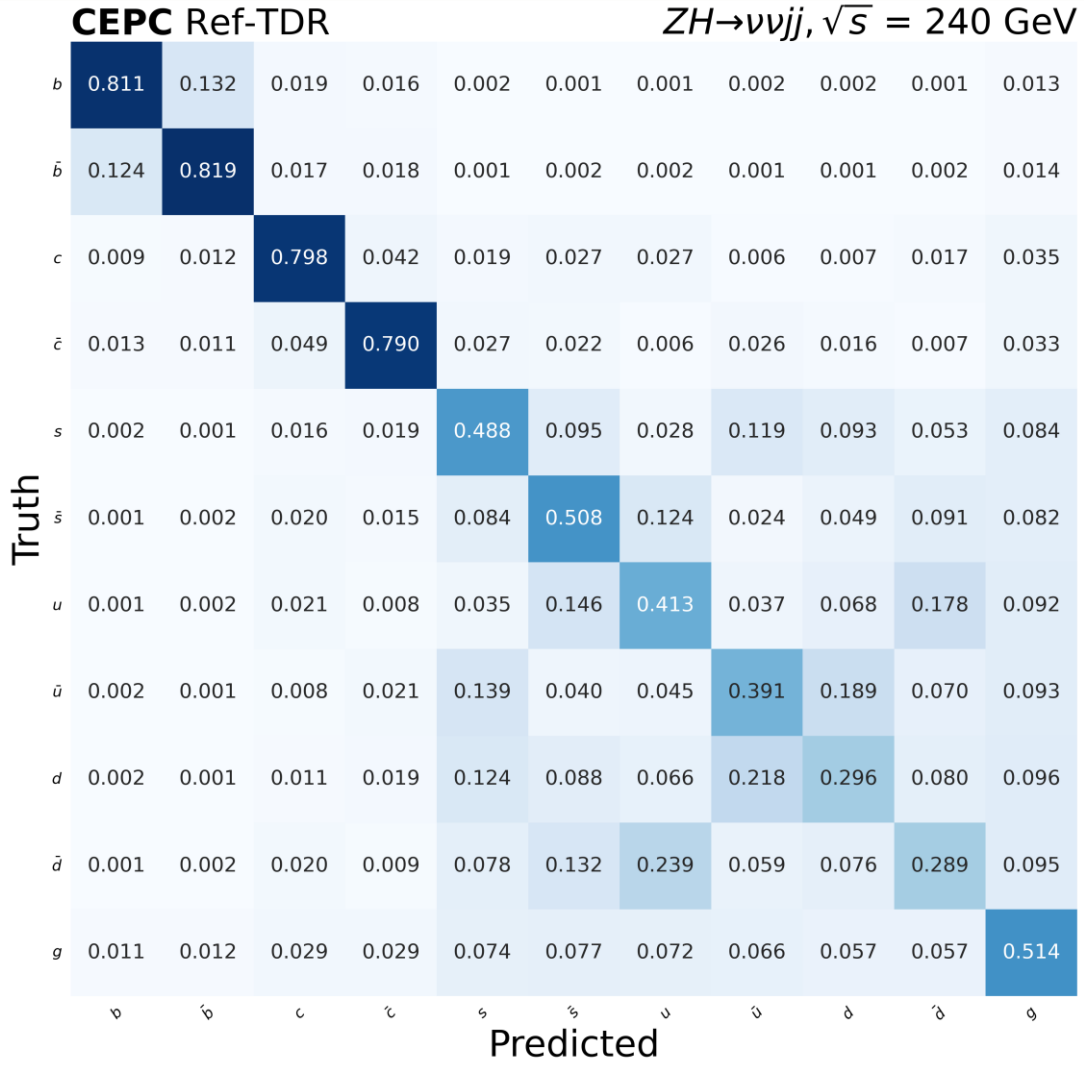
Epoch #0: Current validation metric: 0.50171 (best: 0.50171)
Epoch #12: Current validation metric: 0.53858 (best: 0.53858)
Epoch #29: Current validation metric: 0.56154 (best: 0.56158)

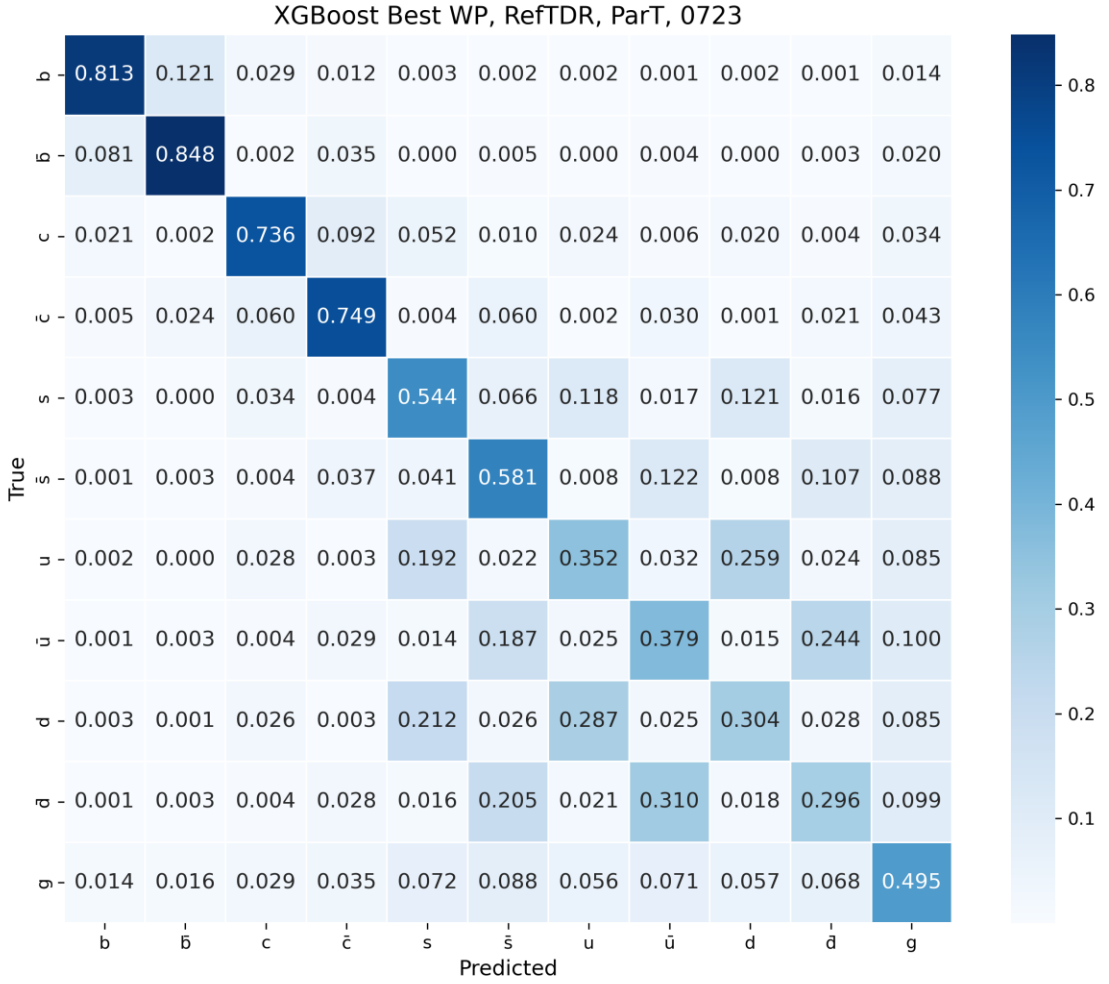
It proved that feature engineering matters for final performance.

d0 and log(d0):
-> We can do sth with these variables. But need check.

Matrix difference between TDR and 0723

Though better metric, new result gets bad performance in cc, and larger charge inconsistence between b/bar.
Also showing interesting pattern between u/d/s. -> Indicating overfit? Under further check.





PrimeTagSvc: latest: v2.5

- https://code.ihep.ac.cn/zhangkl/PrimeTagSvc
- With

  /hpcfs/cepc/higgsgpu/zhangkl/training/onnx_model/H_M11_v35_0728.onnx

- Make sure same bahavior when training;
- Performance still not same, but is_b, is_c, is_lightquark should no big performance degradement.
- Please check.

Interface for output;

```
std::vector<float> GetProb() const override;
int get_type_M11() const override;
int get_type_M6() const override;

// M11 Boolean functions (11 total)
bool is_b_quark() const override;
bool is_b_bar_quark() const override;
bool is_c_quark() const override;
bool is_c_bar_quark() const override;
bool is_s_quark() const override;
bool is_s_bar_quark() const override;
bool is_u_quark() const override;
bool is_u_bar_quark() const override;
bool is_d_quark() const override;
bool is_d_bar_quark() const override;
bool is_gluon() const override;

// M6 Boolean functions (6 total)
bool is_b_jet() const override;
bool is_c_jet() const override;
bool is_s_jet() const override;
bool is_u_jet() const override;
bool is_d_jet() const override;
bool is_gluon_jet() const override;
```