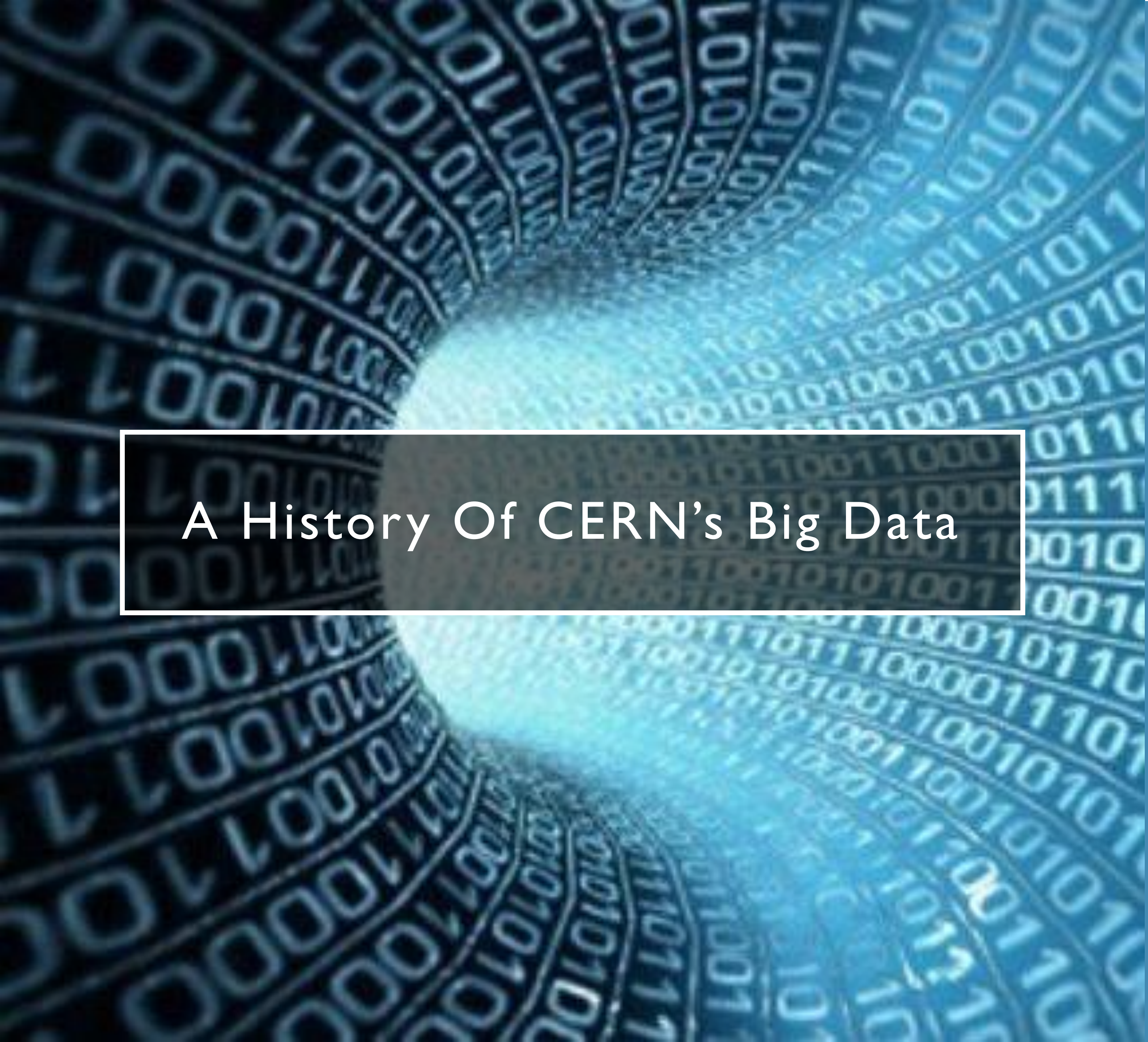
THE FLAVORS OF PHYSICS

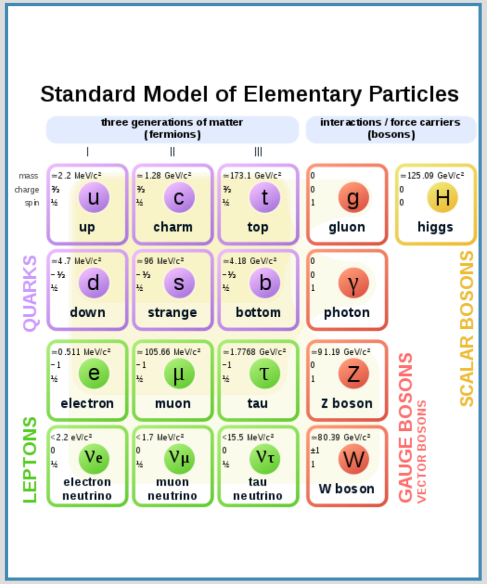
Jeanette Henry
Masters of Science in Data Science
Practicum 1

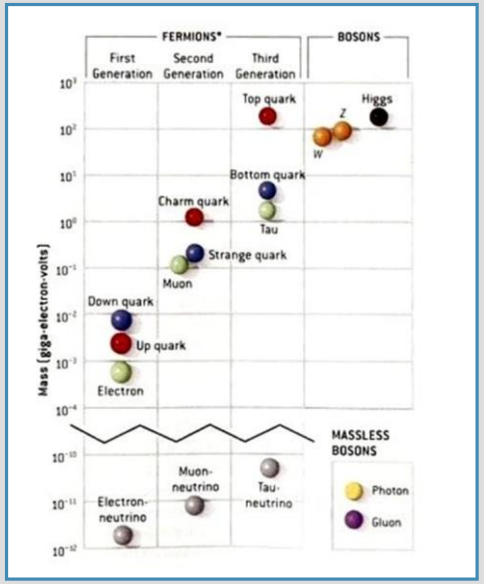


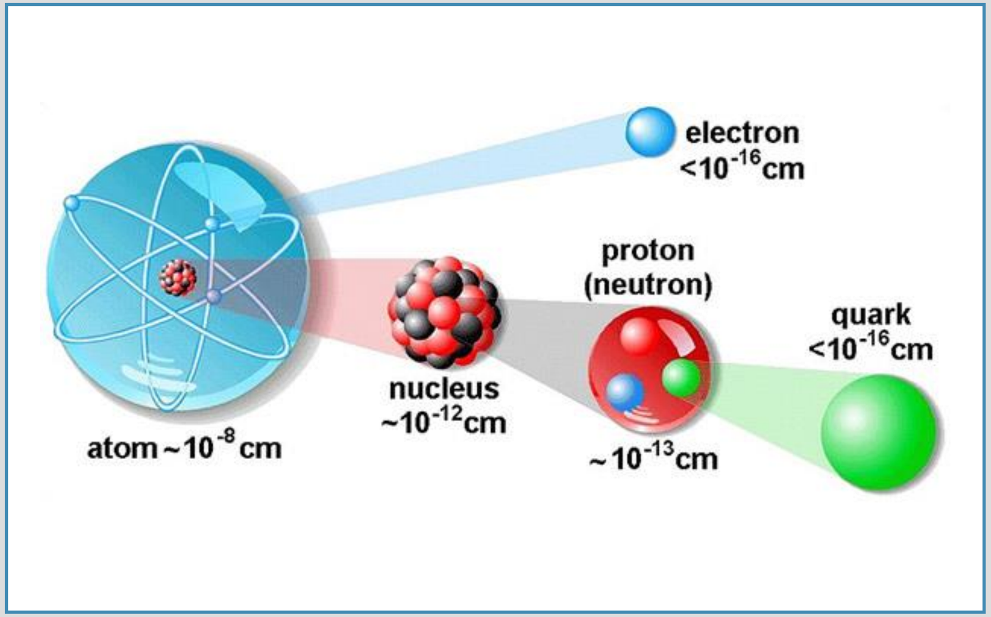
A History Of CERN's Big Data

CERN – The European Organization for Nuclear Research

- 1989 – CERN invented the World Wide Web to meet the demand for sharing very large amounts of scientific information between scientists around the world
- 1998 – CERN began building the Large Hadron Collider (**LHC**)
- 2010 – First data was collected from the LHC, producing 300 GByte/s of data
- 2012 – CERN confirms the Higgs boson was discovered at the LHC
- 2018 – CERN observes the Higgs boson decay into a pair of **tau leptons**







LARGE HADRON COLLIDER

Popular Explanation:

Physicists are using the collider to reproduce the conditions just after the big bang to see what the universe was like when it just started!

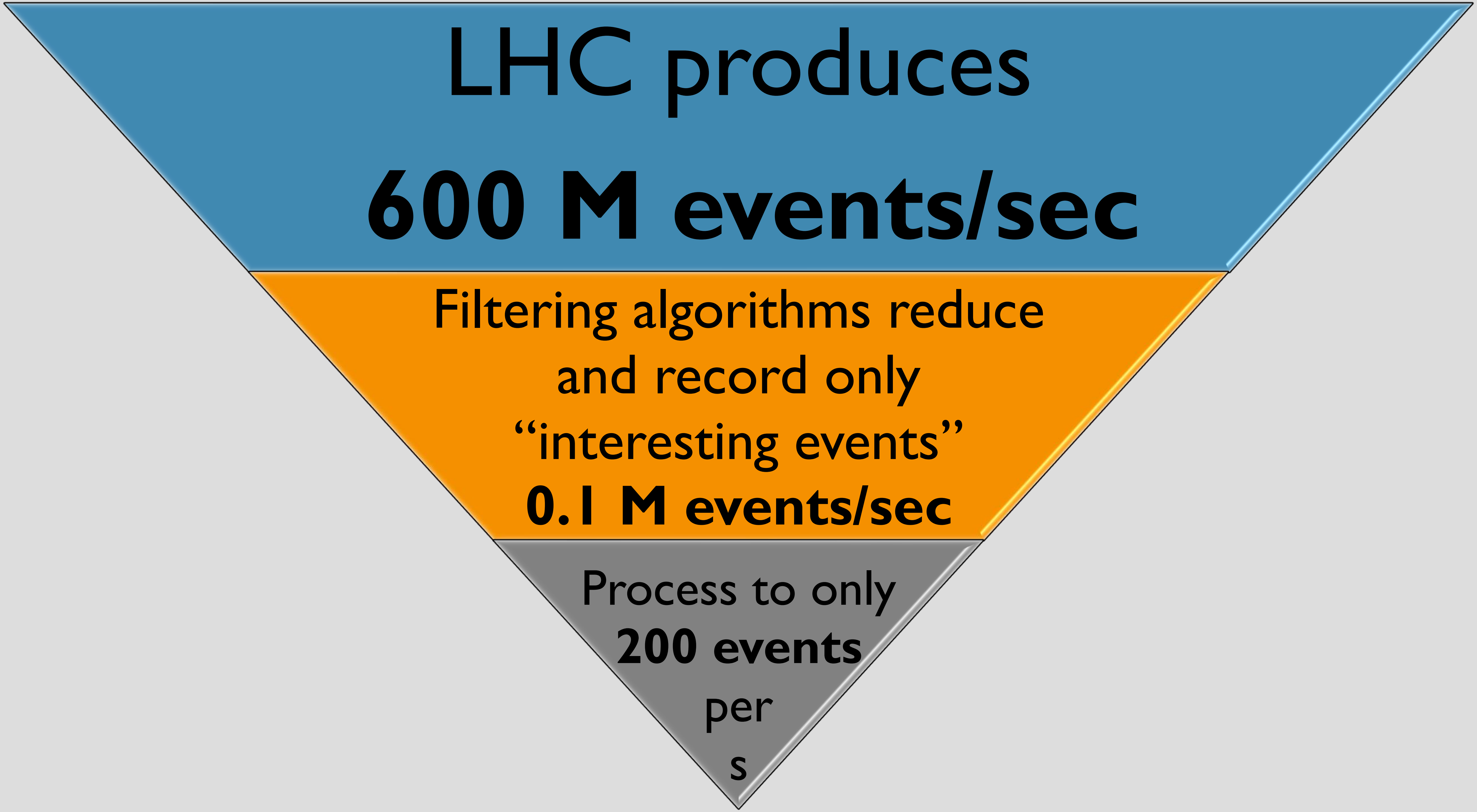
More Accurate Explanation:

We are trying to understand the most basic laws of nature.

BEYOND
THE STANDARD MODEL OF PHYSICS

- Many questions remain unanswered by the current Standard Model of Particle physics, we need models.
- **Physics beyond the Standard Model** (**BSM**) is needed to explain
  - The Nature of Dark Matter
  - Origin of Mass
  - Matter-antimatter asymmetry
- How do we open the door to BSM without knowing what to look for?
- **Charged lepton flavor violations (cLFV) have never been observed.**
- Experiments to observe cLFV would be a CLEAR sign of NEW PHYSICS beyond the Standard Model.

LEPTON FLAVOR VIOLATIONS


LHC produces
600 M events/sec
Filtering algorithms reduce and record only "interesting events"
0.1 M events/sec
Process to only
200 events per s

cLFV EXPERIMENT

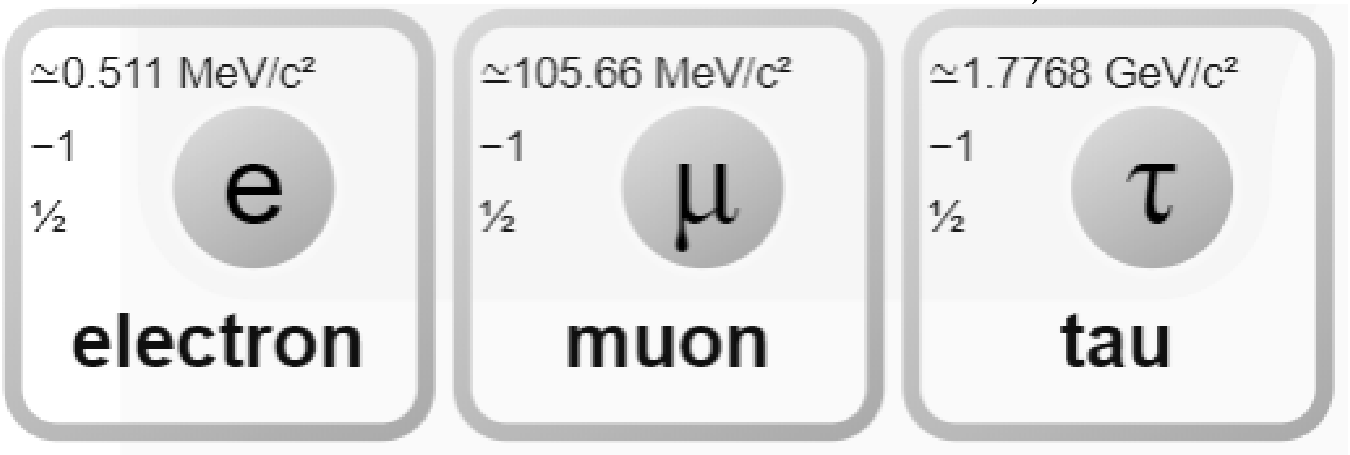
Predict the probability that a LHC event is the

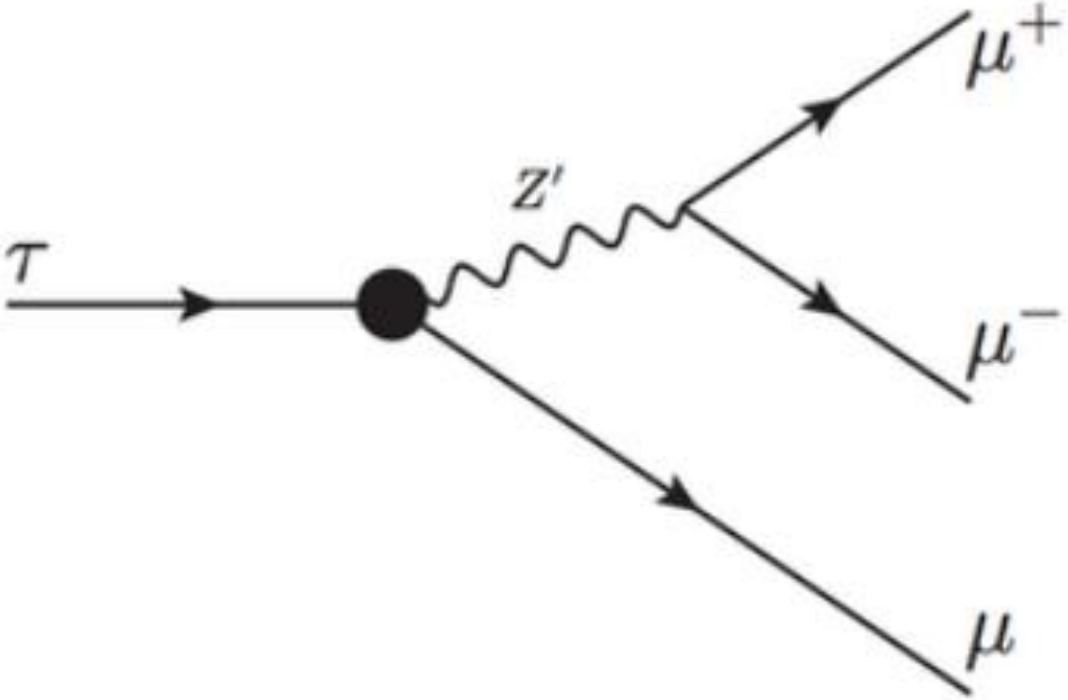
tau → 3 mu

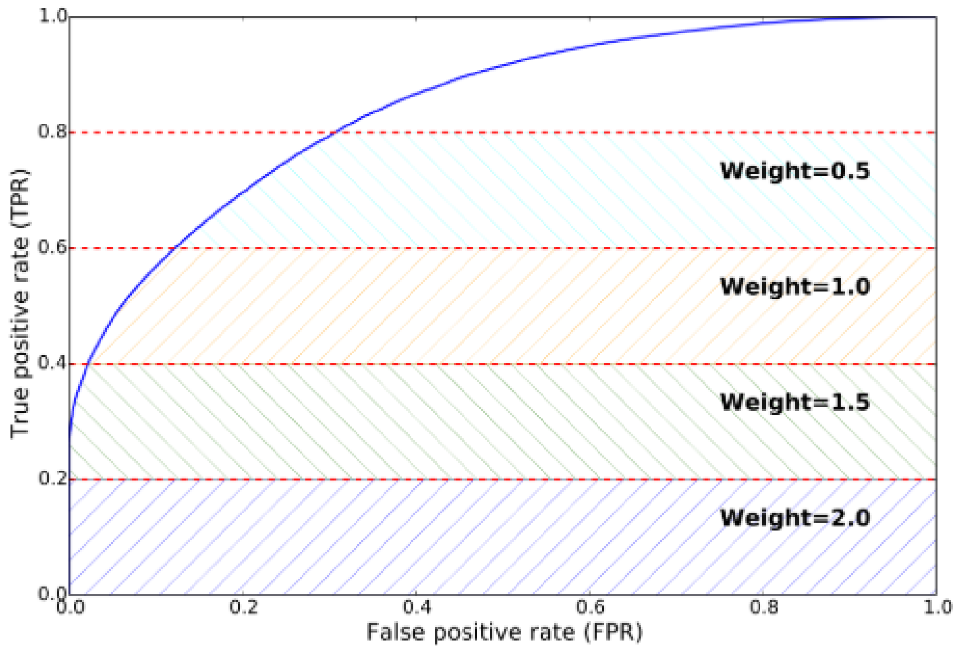
decay

1 e$^-$ 200x e$^-$ 3,478x e$^-$

LEPTONS







MODEL EVALUATION

Weighted AUC Score

Note: Kaggle competition's weighted AUC is calculated only for events (simulated signal events for tau->μμμ and real background events for tau->μμμ) with min_ANNmuon > 0.4

DATA

Training.csv

67,553 events with

51 variables

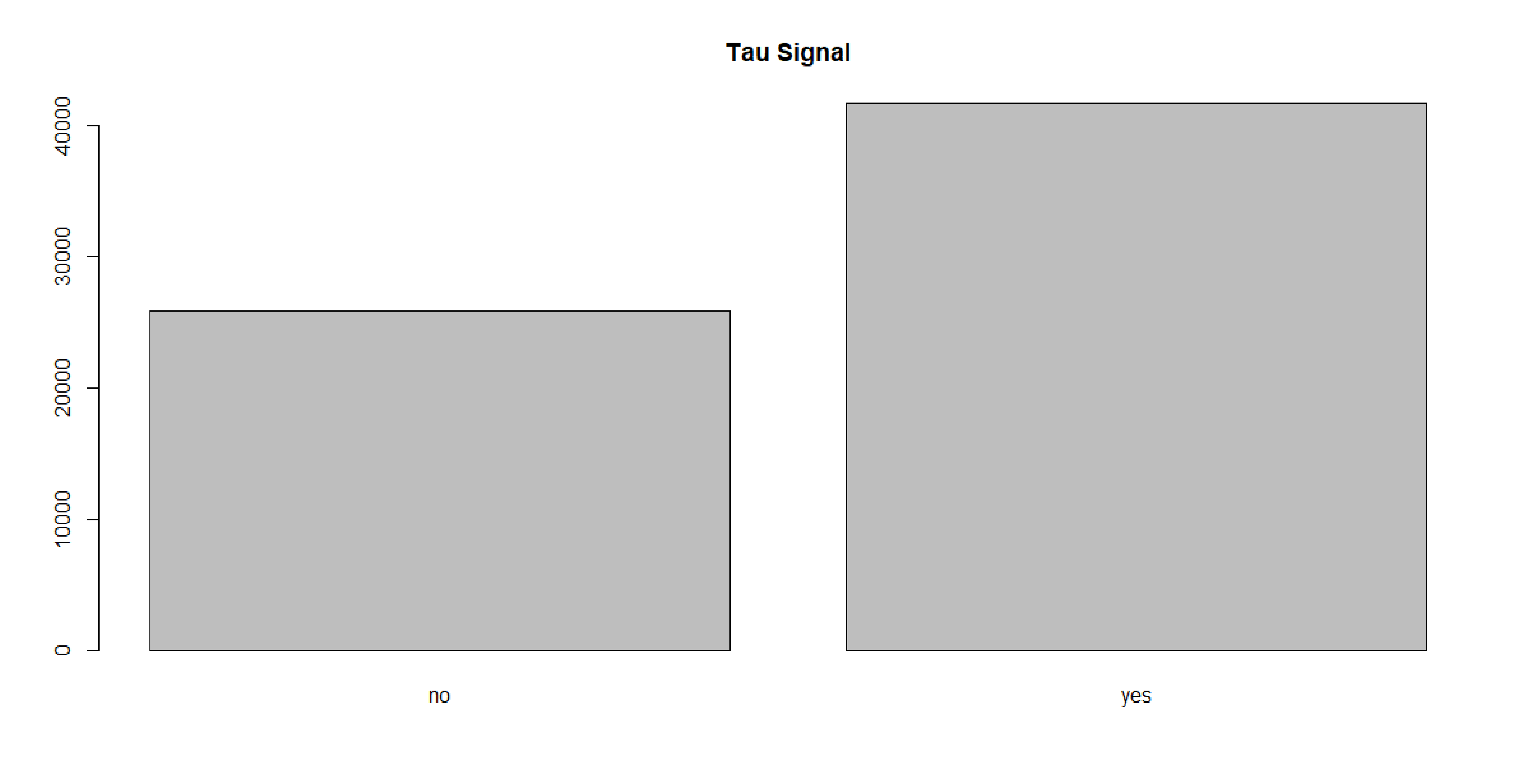
Test.csv

855,819 events with

47 variables

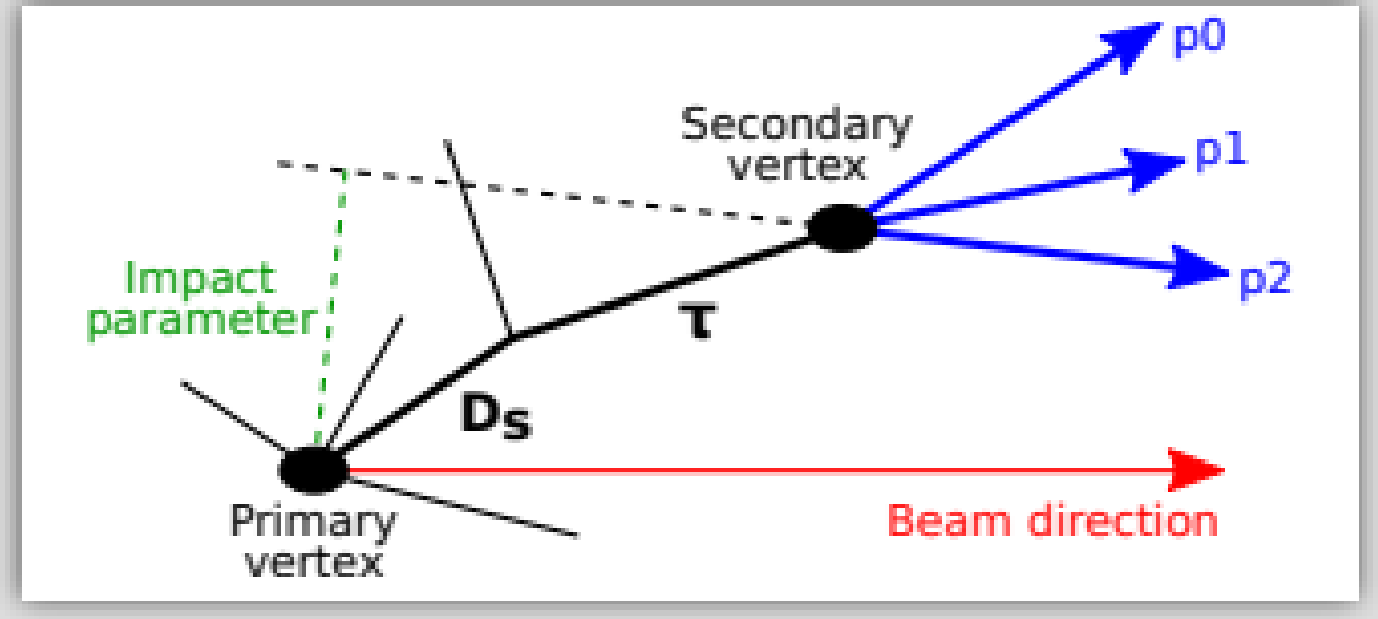
Target Variable:

"Signal"



TEST VARIABLES

- 10 kinematic features (momenta, transverse momenta, and pseudorapidities of p0, p1, and p2; transverse momentum of the mother particle)
- 35 "geometric" features (impact parameters, track isolation variables, flight distance and lifetime of the mother particle, etc.)
- SPDhits (number of hits in the Scintillating Pad Detector, SPD)



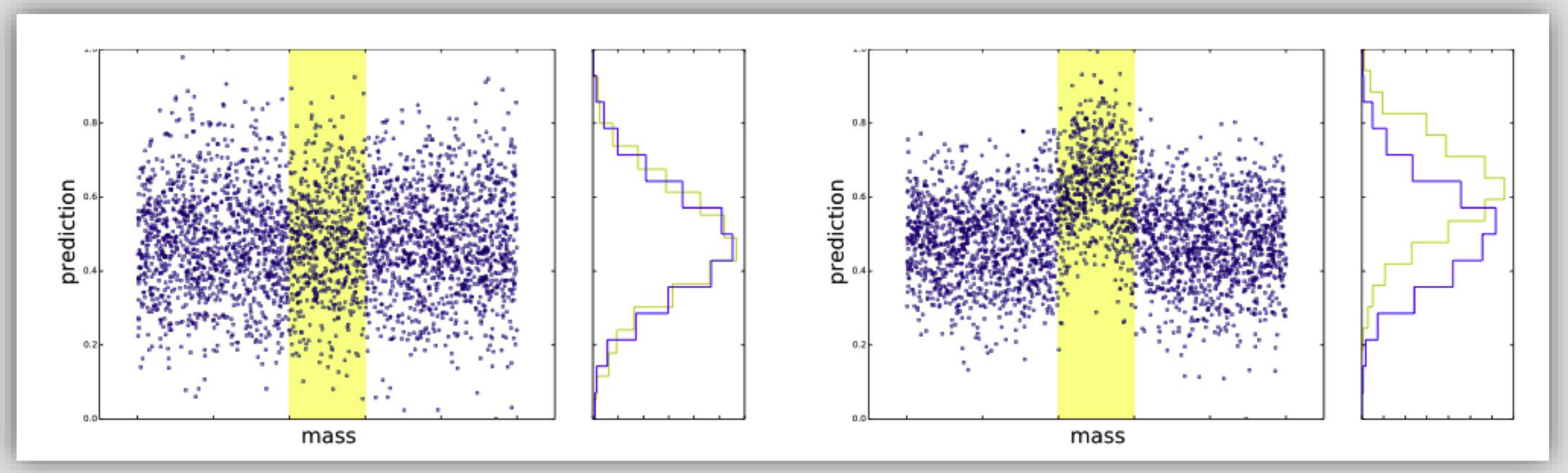
OMITTED VARIABLES

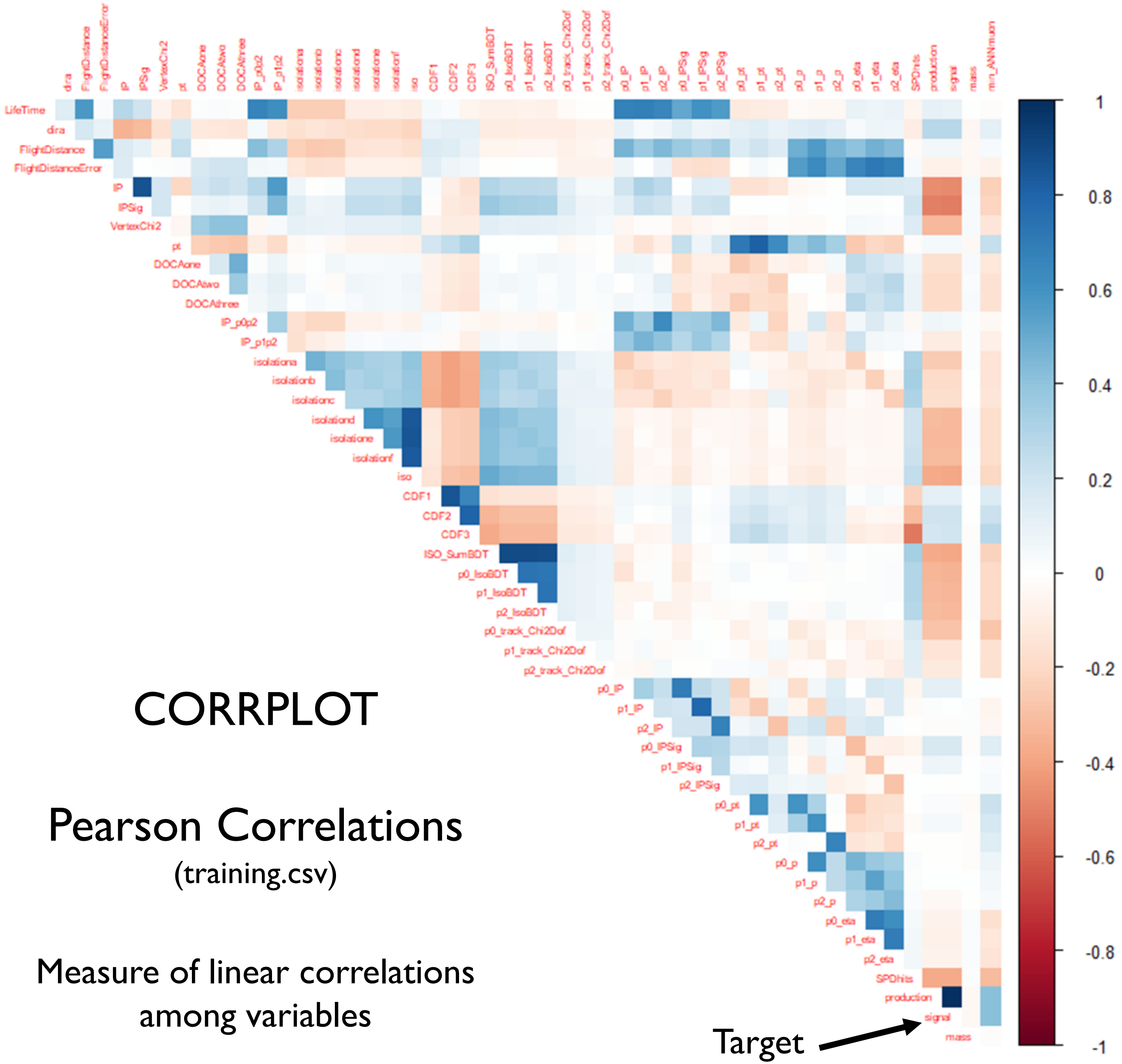
Omitted by CERN:

- Signal – Target Variable
- Production - τ production mechanism
- Min_ANNmuon – CERN's neural network prediction for muon
- Mass – Avoid training bias on an approximated number

Omitted by me:

- SPDhits - hits in the Scintillating Pad Detector, SPD
  - Causes model to exploit simulated data imperfections and fail Kaggle's "agreement test"





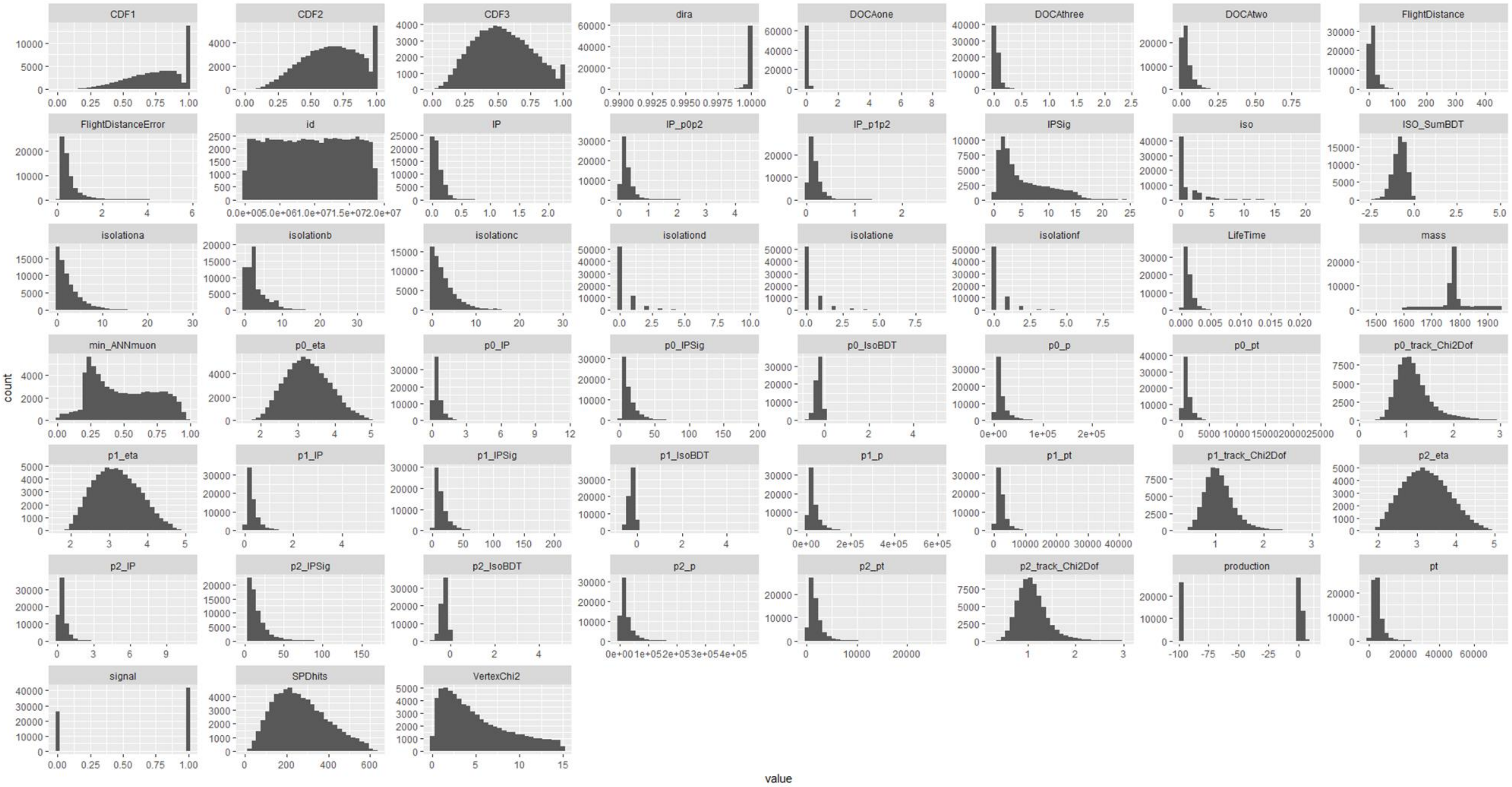
CORRPLOT

Pearson Correlations

(training.csv)

Measure of linear correlations among variables

VALUES OF ALL VARIABLES IN TRAIN.CSV

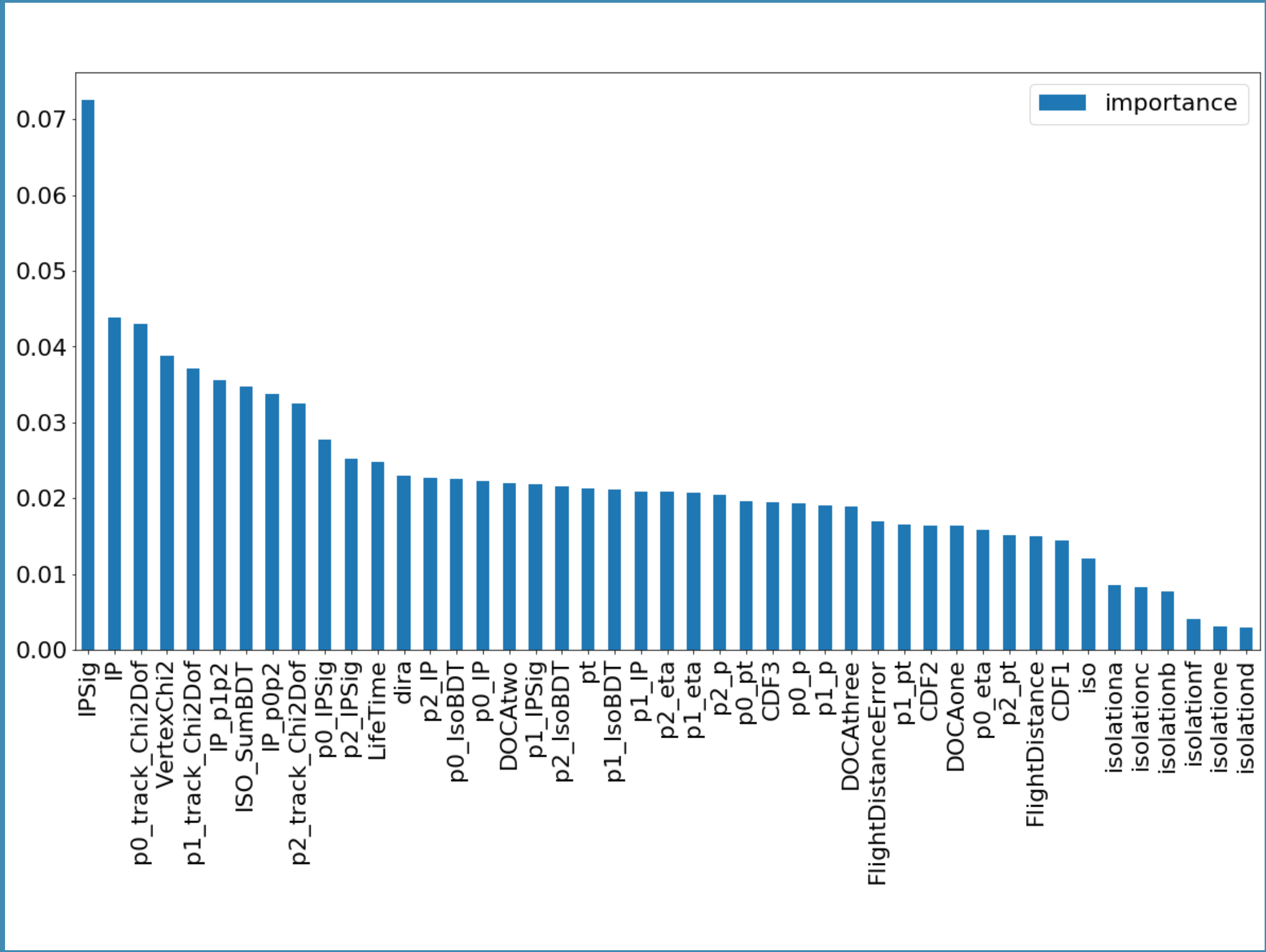


FEATURE IMPORTANCES

Python package: from **sklearn.ensemble** import **GradientBoostingClassifier**

1. Split train data into training and validation sets
2. Train a baseline model on all 46 testing variables
3. Sort through the "feature_importances_"

```
GBC = GradientBoostingClassifier(n_estimators=200,
                                 learning_rate=0.1,
                                 subsample=0.4,
                                 min_samples_leaf=3,
                                 max_depth=6,
                                 random_state=11)
GBC.fit(tn[variables], tn['signal'])
```



OPTIMIZING VARIABLE SELECTION

How can removing variables increase the weighted AUC score of model?

```
model = GradientBoostingClassifier(n_estimators=200, learning_rate=0.1, subsample=0.4,
                                   min_samples_leaf=3, max_depth=6, random_state=11)
model.fit(X_train, y_train)
# make predictions for test data and evaluate
val_probs = model.predict_proba(X_test)[:, 1]
roc_auc = evaluation.roc_auc_truncated(y_test, val_probs)
print("AUC: %.6f" % (roc_auc))
# Fit model using each importance as a threshold
thresholds = sort(model.feature_importances_)
for thresh in thresholds:
        # select features using threshold
    selection = SelectFromModel(model, threshold=thresh, prefit=True)
    select_X_train = selection.transform(X_train)
    # train model
    selection_model = GradientBoostingClassifier(n_estimators=200, learning_rate=0.1, subsample=0.4,
                                   min_samples_leaf=3, max_depth=6, random_state=11)
    selection_model.fit(select_X_train, y_train)
    # evaluate model
    select_X_test = selection.transform(X_test)

    val_probs = selection_model.predict_proba(select_X_test)[:, 1]
    roc_auc = evaluation.roc_auc_truncated(y_test, val_probs)

    print("Thresh=%.6f, n=%d, AUC: %.6f" % (thresh, select_X_train.shape[1], roc_auc))
```

Baseline AUC: 0.984050
Selection AUC: 0.985136

FEAT. IMPORTANCE THRESHOLDS

	importance
isolationd	0.002999
isolatione	0.003026
isolationf	0.004032
isolationb	0.007695
isolationc	0.008320
isolationa	0.008556
iso	0.011978
CDF1	0.014434
FlightDistance	0.015011
p2_pt	0.015169
p0_eta	0.015775
DOCAone	0.016383
CDF2	0.016421
p1_pt	0.016519
FlightDistanceError	0.016910
DOCAthree	0.018937
p1_p	0.019053
p0_p	0.019255
CDF3	0.019451
p0_pt	0.019558
p2_p	0.020499
p1_eta	0.020732
p2_eta	0.020832
p1_IP	0.020892
p1_IsoBDT	0.021180
pt	0.021221
p2_IsoBDT	0.021568
p1_IPSig	0.021841
DOCAtwo	0.022036
p0_IP	0.022199
p0_IsoBDT	0.022571
p2_IP	0.022666
dira	0.022913
LifeTime	0.024793
p2_IPSig	0.025179
p0_IPSig	0.027679
p2_track_Chi2Dof	0.032479
IP_p0p2	0.033720
ISO_SumBDT	0.034714
IP_p1p2	0.035592
p1_track_Chi2Dof	0.037100
VertexChi2	0.038781
p0_track_Chi2Dof	0.042942
IP	0.043879
IPSig	0.072506

VARIABLE SELECTION LOOP

Basline AUC: 0.984050

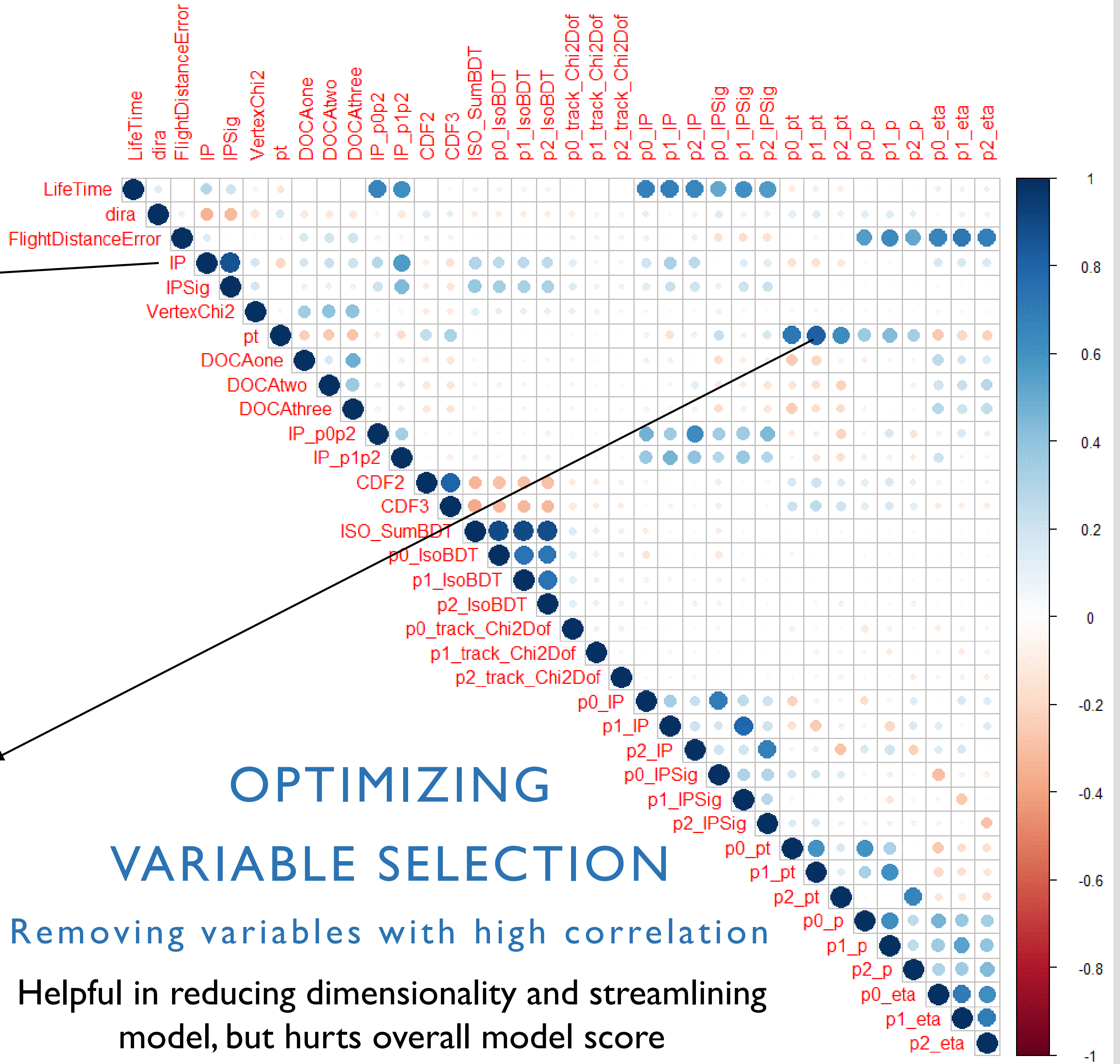
Thresh=0.002999, n=45, AUC: 0.984050
Thresh=0.003026, n=44, AUC: 0.983583
Thresh=0.004032, n=43, AUC: 0.984252
Thresh=0.007695, n=42, AUC: 0.984945
Thresh=0.008320, n=41, AUC: 0.985029
Thresh=0.008556, n=40, AUC: 0.983928
Thresh=0.011978, n=39, AUC: 0.982815
Thresh=0.014434, n=38, AUC: 0.983715
Thresh=0.015011, n=37, AUC: 0.983858
Thresh=0.015169, n=36, AUC: 0.985136
Thresh=0.015775, n=35, AUC: 0.984464
Thresh=0.016383, n=34, AUC: 0.984668
Thresh=0.016421, n=33, AUC: 0.983949
Thresh=0.016519, n=32, AUC: 0.984217
Thresh=0.016910, n=31, AUC: 0.984202
Thresh=0.018937, n=30, AUC: 0.983723
Thresh=0.019053, n=29, AUC: 0.983153
Thresh=0.019255, n=28, AUC: 0.983194
Thresh=0.019451, n=27, AUC: 0.982219
Thresh=0.019558, n=26, AUC: 0.983566
Thresh=0.020499, n=25, AUC: 0.984289
Thresh=0.020732, n=24, AUC: 0.983030
Thresh=0.020832, n=23, AUC: 0.982702
Thresh=0.020892, n=22, AUC: 0.981903
Thresh=0.021180, n=21, AUC: 0.982184
Thresh=0.021221, n=20, AUC: 0.983123
Thresh=0.021568, n=19, AUC: 0.981621
Thresh=0.021841, n=18, AUC: 0.981509
Thresh=0.022036, n=17, AUC: 0.980400
Thresh=0.022199, n=16, AUC: 0.981149
Thresh=0.022571, n=15, AUC: 0.979629
Thresh=0.022666, n=14, AUC: 0.980524
Thresh=0.022913, n=13, AUC: 0.980108
Thresh=0.024793, n=12, AUC: 0.980628
Thresh=0.025179, n=11, AUC: 0.978924
Thresh=0.027679, n=10, AUC: 0.978799
Thresh=0.032479, n=9, AUC: 0.977865
Thresh=0.033720, n=8, AUC: 0.977296
Thresh=0.034714, n=7, AUC: 0.973143
Thresh=0.035592, n=6, AUC: 0.965965
Thresh=0.037100, n=5, AUC: 0.961653
Thresh=0.038781, n=4, AUC: 0.962452
Thresh=0.042942, n=3, AUC: 0.954894
Thresh=0.043879, n=2, AUC: 0.951143
Thresh=0.072506, n=1, AUC: 0.948675

	importance
IPSig	0.072506
IP	0.043879
p0_track_Chi2Dof	0.042942
VertexChi2	0.038781
p1_track_Chi2Dof	0.037100
IP_p1p2	0.035592
ISO_SumBDT	0.034714
IP_p0p2	0.033720
p2_track_Chi2Dof	0.032479
p0_IPSig	0.027679
p2_IPSig	0.025179
LifeTime	0.024793
dira	0.022913
p2_IP	0.022666
p0_IsoBDT	0.022571
p0_IP	0.022199
DOCAtwo	0.022036
p1_IPSig	0.021841
p2_IsoBDT	0.021568
pt	0.021221
p1_IsoBDT	0.021180
p1_IP	0.020892
p2_eta	0.020832
p1_eta	0.020732
p2_p	0.020499
p0_pt	0.019558
CDF3	0.019451
p0_p	0.019255
p1_p	0.019053
DOCAthree	0.018937
FlightDistanceError	0.016910
p1_pt	0.016519
CDF2	0.016421
DOCAone	0.016383
p0_eta	0.015775
p2_pt	0.015169
FlightDistance	0.015011
CDF1	0.014434
iso	0.011978
isolationa	0.008556
isolationc	0.008320
isolationb	0.007695
isolationf	0.004032
isolatione	0.003026
isolationd	0.002999

Current high AUC: 0.985136

Removing 'IP': AUC: 0.984291

Removing 'p1_pt': AUC: 0.984620



OPTIMIZING VARIABLE SELECTION

Removing variables with high correlation

Helpful in reducing dimensionality and streamlining model, but hurts overall model score

READY TO SUBMIT?

CORRELATION TEST

- Check that model is not correlated with mass

- PASS!

```
check_correlation = pandas.read_csv(folder + 'check_correlation.csv',
                                    index_col='id')
correlation_probs = model.predict_proba(check_correlation[variables])[:, 1]
cvm = evaluation.compute_cvm(correlation_probs, check_correlation['mass'])
print ('CvM metric', cvm, cvm < 0.002)
```

```
CvM metric 0.000907392904567 True
```

AGREEMENT TEST

- Check that model is not bias on simulated data

- Fail!!!

```
check_agreement = pandas.read_csv(folder + 'check_agreement.csv', index_col='id')
agreement_probs = model.predict_proba(check_agreement[variables])[:, 1]

ks = evaluation.compute_ks(
    agreement_probs[check_agreement['signal'].values == 0],
    agreement_probs[check_agreement['signal'].values == 1],
    check_agreement[check_agreement['signal'] == 0]['weight'].values,
    check_agreement[check_agreement['signal'] == 1]['weight'].values)
print('KS metric', ks, ks < 0.09)
```

```
KS metric 0.119832313137 False
```

MODEL #2

- Reduce the training variables further to reduce bias

variables2 = ['IPSig', 'p0_track_Chi2Dof', 'p1_track_Chi2Dof', 'p2_track_Chi2Dof', 'VertexChi2', 'ISO_SumBDT’]

model2 = GradientBoostingClassifier(n_estimators=200, learning_rate=0.1, subsample=0.4, min_samples_leaf=3, max_depth=6, random_state=11)

model2.fit(X_train2, y_train)

Passes Correlation test

and

Passes Agreement Test

COMBINE THE PREDICTIONS

28% MODEL 1 + 72% MODEL 2

Check correlation test

In [96]:
```
check_correlation = pandas.read_csv(folder + 'check_correlation.csv', index_col='id')
correlation_probs =  (0.28*(model.predict_proba(check_correlation[variables])[:, 1])+
                      0.72*(model2.predict_proba(check_correlation[variables2])[:, 1]))
cvm = evaluation.compute_cvm(correlation_probs, check_correlation['mass'])
print ('CvM metric', cvm, cvm < 0.002)
```

```
CvM metric 0.001037191579 True
```

Check agreement test

In [95]:
```
check_agreement = pandas.read_csv(folder + 'check_agreement.csv', index_col='id')
agreement_probs = (0.28*(model.predict_proba(check_agreement[variables])[:, 1])+
                   0.72*(model2.predict_proba(check_agreement[variables2])[:, 1]))

ks = evaluation.compute_ks(
    agreement_probs[check_agreement['signal'].values == 0],
    agreement_probs[check_agreement['signal'].values == 1],
    check_agreement[check_agreement['signal'] == 0]['weight'].values,
    check_agreement[check_agreement['signal'] == 1]['weight'].values)
print('KS metric', ks, ks < 0.09)
```

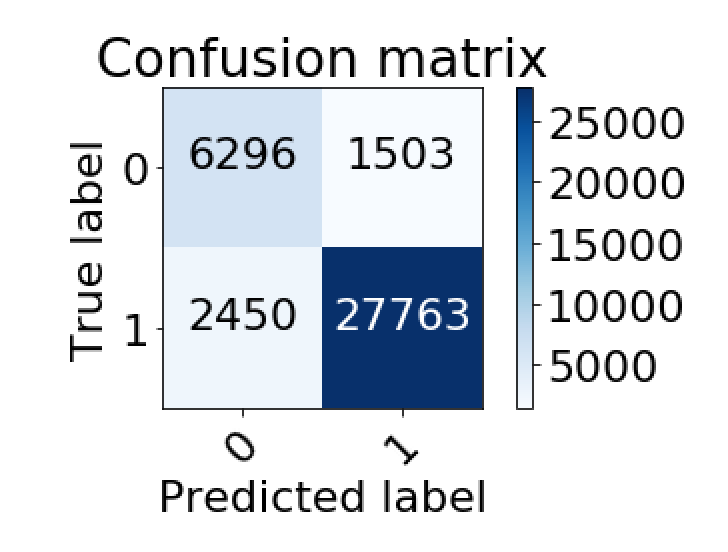
```
KS metric 0.0896123420242 True
```

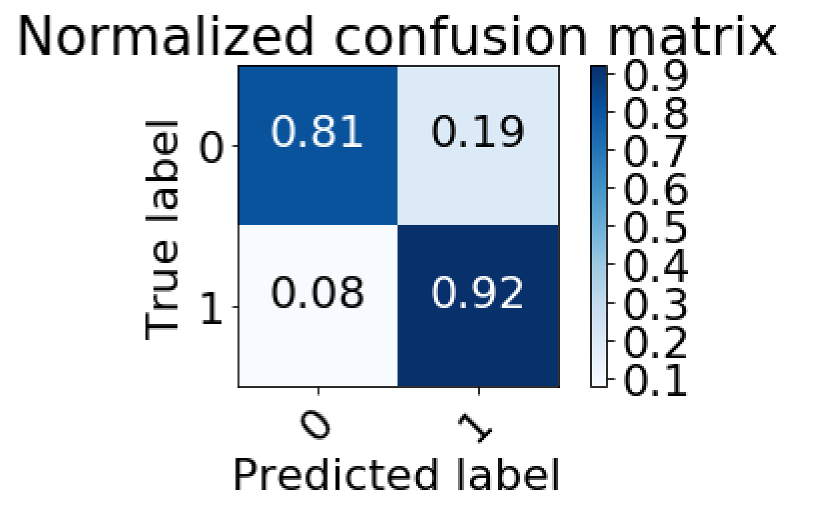
Compute weighted AUC on the training data with min_ANNmuon > 0.4

In [99]:
```
tn_eval = tn[tn['min_ANNmuon'] > 0.4]
tn_probs = (0.28*(model.predict_proba(tn_eval[variables])[:, 1])+
            0.72*(model2.predict_proba(tn_eval[variables2])[:, 1]))

roc_auc = evaluation.roc_auc_truncated(tn_eval['signal'], tn_probs)
print( 'AUC', roc_auc)
```

```
AUC 0.98983411521
```





SUBMISSION TO KAGGLE

```
test = pandas.read_csv(folder + 'test.csv', index_col='id')
result = pandas.DataFrame({'id': test.index})
result['prediction'] = (0.28*(model.predict_proba(test[variables])[:, 1])+
                        0.72*(model2.predict_proba(test[variables2])[:, 1]))
```

```
result.to_csv('JH101618.csv', index=False, sep=',')
```

Submission and Description	Private Score
JH101618.csv 18 minutes ago by JeanetteHenry Ensemble	0.979988

This competition is brought to you by:

CERN

LHCb

SCHOOL OF DATA ANALYSIS


THANK YOU