In [ ]:

```
%matplotlib inline

import os
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
```

In [ ]:

```
eng_levels = [0, 1]
```

# Face Features

In [ ]:

```
base_dir = "../sound/features/"
```

In [ ]:

```
# Load data
df = pd.read_csv(os.path.join(base_dir, 'all.csv'))
labels = (df['label'] <= 0).astype(int) #binarize labels
df.head()
```

Out[ ]:

| | F0avg | F0std | F0max | F0min | F0skew | F0kurt | F0tiltavg | F0mseavg | F0tiltstd | F0msestd | ... | maxdurpause | min |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 107.251472 | 4.754879 | 112.142250 | 97.891090 | -1.044098 | -0.510219 | 38.107999 | 21.846451 | 0.000000 | 0.000000 | ... | 0.00 | |
| **1** | 107.249100 | 4.753147 | 112.142258 | 97.891144 | -1.044382 | -0.509233 | 38.172404 | 21.827407 | 0.000000 | 0.000000 | ... | 0.00 | |
| **2** | 114.073090 | 35.353394 | 238.759155 | 69.948997 | 1.999682 | 2.876741 | -39.423164 | 100.290732 | 227.348111 | 195.716531 | ... | 1.19 | |
| **3** | 131.886368 | 30.292049 | 241.647980 | 66.003014 | 0.802149 | 0.373328 | -31.940573 | 135.096570 | 237.915601 | 177.521635 | ... | 0.98 | |
| **4** | 118.834885 | 20.374716 | 223.293121 | 66.754921 | 1.245751 | 3.671583 | -79.806733 | 89.552508 | 227.226774 | 94.320115 | ... | 0.93 | |

5 rows × 105 columns

In [ ]:

```
print(len(df), len(labels))
```

```
216 216
```

In [ ]:

```
labels.value_counts()
```

Out[ ]:

```
0    131
1     85
Name: label, dtype: int64
```

In [ ]:

```
# Define Feature Series Ranges
r_f0 = range(1,7)
r_dur_voiced = range(80, 86)
r_dur_unvoiced = range(86, 92)

df_f0 = df.iloc[:, r_f0]
df_dur_voiced = df.iloc[:, r_dur_voiced]
df_dur_unvoiced = df.iloc[:, r_dur_unvoiced]
#df_f0.to_csv("../sound/reduced/f0.tsv", sep="\t", index=False)
```

In [ ]:

```
df_f0['label'] = labels.values
df_dur_voiced['label'] = labels.values
df_dur_unvoiced['label'] = labels.values

df = df.iloc[:, :-2]
df['label'] = labels.values
```

In [ ]:

```
df_f0.sample()
```

Out[ ]:

| | F0std | F0max | F0min | F0skew | F0kurt | F0tiltavg |
|---|---|---|---|---|---|---|
| **42** | 32.433376 | 231.657715 | 66.797943 | 1.016207 | 0.350995 | -217.466816 |

In [ ]:

```
df_dur_voiced.sample()
```

Out[ ]:

| | stddurvoiced | skwdurvoiced | kurtosisdurvoiced | maxdurvoiced | mindurvoiced | avgdurunvoiced |
|---|---|---|---|---|---|---|
| **130** | 0.199038 | 1.408875 | 2.379309 | 1.02 | 0.02 | 0.073143 |

In [ ]:

```
df_dur_unvoiced.sample()
```

Out[ ]:

| | stddurunvoiced | skwdurunvoiced | kurtosisdurunvoiced | maxdurunvoiced | mindurunvoiced | avgdurpause |
|---|---|---|---|---|---|---|
| **145** | 0.045 | -1.487246e-16 | -2.0 | 0.13 | 0.04 | 0.0 |

In [ ]:

```
df_all = pd.concat([df_f0.iloc[:, :-1],
df_dur_voiced.iloc[:, :-1],
df_dur_unvoiced],axis=1)
```

In [ ]:

```
feature_sets = {
    "F0": df_f0,
    "Duration of Voiced": df_dur_voiced,
    "Duration of UnVoiced": df_dur_unvoiced,
    "All Selected Features": df_all,
    "All Features": df
}
```

In [ ]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.model_selection import cross_val_score
from sklearn.metrics import f1_score
```

In [ ]:

```
#!pip install git+https://github.com/christophM/rulefit.git
from rulefit import RuleFit
```

In [ ]:

```
for title in feature_sets:
    dfc = feature_sets[title]
    not_zero_ind = ~(dfc == 0).all(axis=1)

    dfc = dfc.loc[not_zero_ind]
    labels = dfc['label'].loc[not_zero_ind]

    not_nan_index = ~dfc.isna().any(axis=1)
    dfc = dfc[not_nan_index]
    labels = labels[not_nan_index]

    scaler = StandardScaler()
    scaled_samples = scaler.fit_transform(dfc.iloc[:,:-1])

    X_train, X_test, y_train, y_test = train_test_split(scaled_samples, labels, test_size=0.2, random_state=42, stratif

    features = dfc.columns

    rf = RuleFit(model_type='r', rfmode='classify', max_iter=5000, n_jobs=-1) ## Classification task with only rule-bas
    rf.fit(X_train, y_train, feature_names=features)
    y_pred = rf.predict(X_test)
    res = f1_score(y_test, y_pred, average='weighted')
    print(title, "f1", res)
    rules = rf.get_rules()
    rules = rules[rules.coef != 0].sort_values("support", ascending=False)
    rules.to_csv("reports/interpret/pose/rule-%s.csv" % title)
```

```
F0 f1 0.6607449154618966
Duration of Voiced f1 0.7090844821491694
Duration of UnVoiced f1 0.515696113898938
All Selected Features f1 0.7116883116883116
All Features f1 0.47729163929400437
```

In [ ]:

```
"""
pca = PCA()
rf = RandomForestClassifier(n_estimators=100, n_jobs=-1)

blackbox_model = Pipeline([('pca', pca), ('rf', rf)])
"""
blackbox_model = SVC(gamma=2, C=1, probability=True)
```

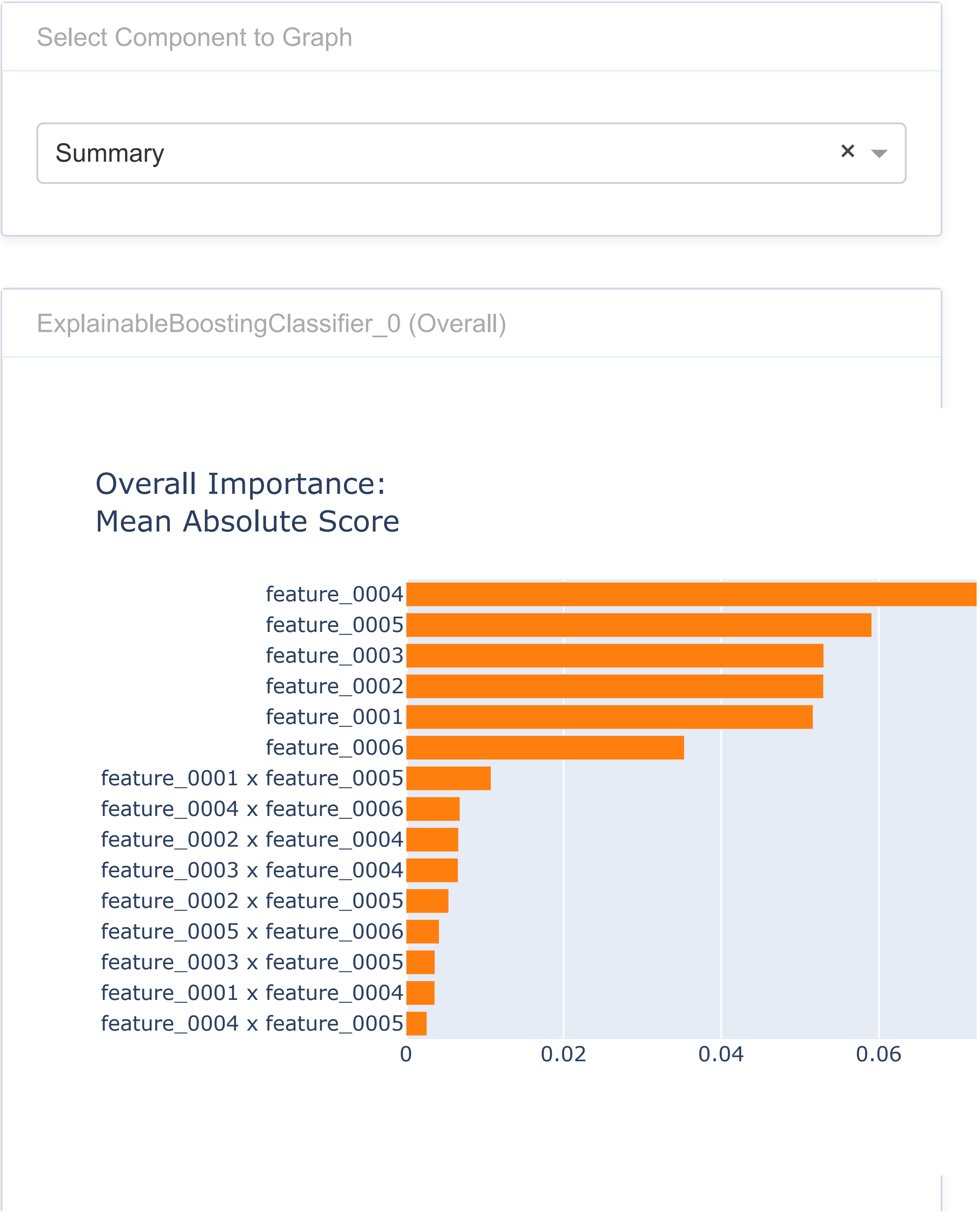
In [ ]:

```
from interpret import show
from interpret.perf import ROC
from interpret.blackbox import LimeTabular
from interpret import show
from interpret.blackbox import ShapKernel
from interpret.blackbox import MorrisSensitivity
from interpret.blackbox import PartialDependence
from interpret.glassbox import ExplainableBoostingClassifier
```

In [ ]:

```
for title in feature_sets:
    ebm = ExplainableBoostingClassifier()
    dfc = feature_sets[title]
    not_zero_ind = ~(dfc == 0).all(axis=1)

    dfc = dfc.loc[not_zero_ind]
    labels = dfc['label'].loc[not_zero_ind]

    not_nan_index = ~dfc.isna().any(axis=1)
    dfc = dfc[not_nan_index]
    labels = labels[not_nan_index]

    scaler = StandardScaler()
    scaled_samples = scaler.fit_transform(dfc.iloc[:,:-1])
    X_train, X_test, y_train, y_test = train_test_split(scaled_samples, labels, test_size=0.2, random_state=42, stratif

    ebm.fit(X_train, y_train)
    ebm_global = ebm.explain_global()
    show(ebm_global)
```

```
c:\Users\ASABUNCUOGLU13\Anaconda3\lib\site-packages\interpret\visual\udash.py:5: UserWarning:
The dash_html_components package is deprecated. Please replace
`import dash_html_components as html` with `from dash import html`
  import dash_html_components as html
c:\Users\ASABUNCUOGLU13\Anaconda3\lib\site-packages\interpret\visual\udash.py:6: UserWarning:
The dash_core_components package is deprecated. Please replace
`import dash_core_components as dcc` with `from dash import dcc`
  import dash_core_components as dcc
c:\Users\ASABUNCUOGLU13\Anaconda3\lib\site-packages\interpret\visual\udash.py:7: UserWarning:
The dash_table package is deprecated. Please replace
`import dash_table` with `from dash import dash_table`

Also, if you're using any of the table format helpers (e.g. Group), replace
`from dash_table.Format import Group` with
`from dash.dash_table.Format import Group`
  import dash_table as dt
```

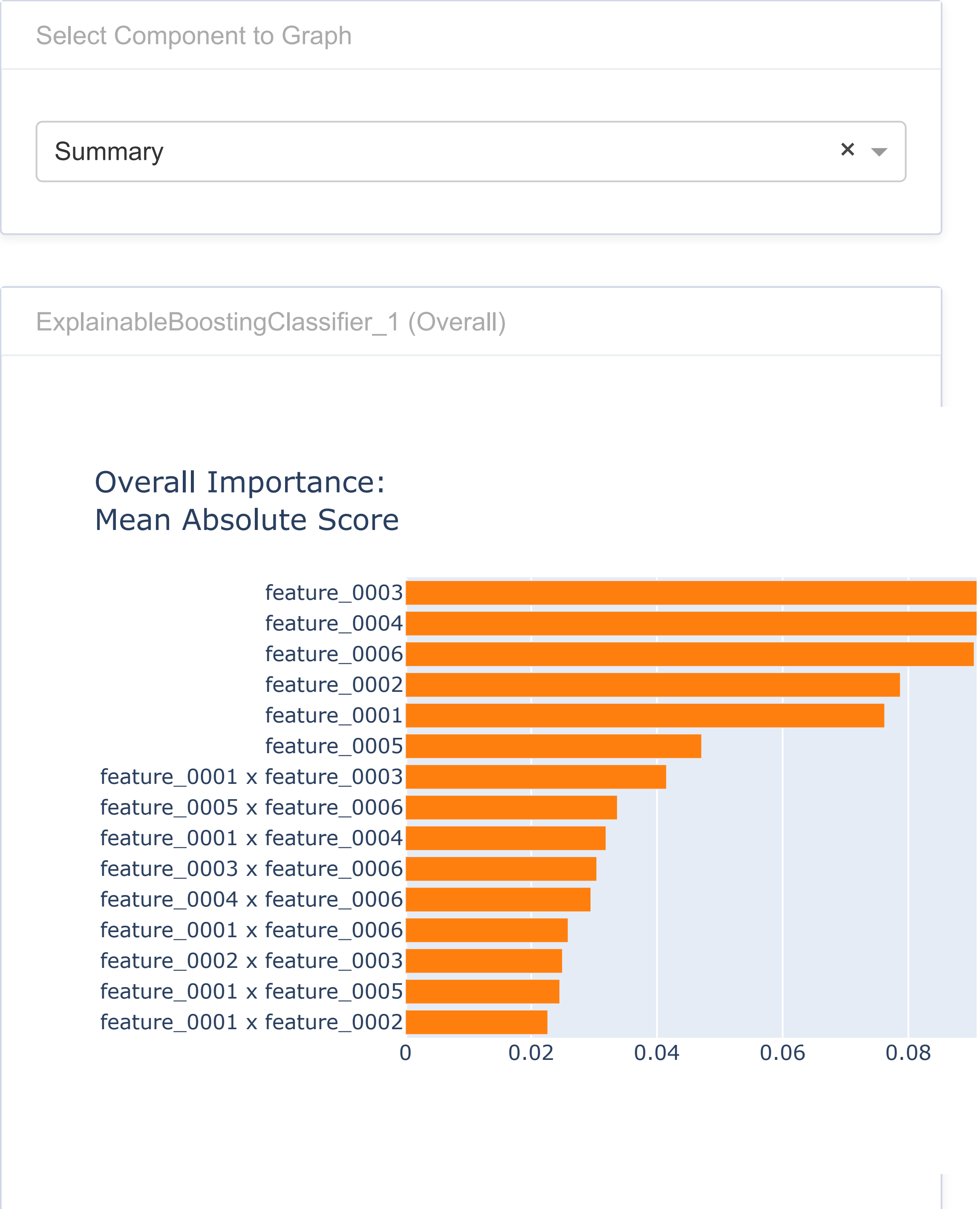


Select Component to Graph

Summary

ExplainableBoostingClassifier_0 (Overall)

×
Overall Importance:
Mean Absolute Score
feature_0004
feature_0005
feature_0003
feature_0002
feature_0001
feature_0006
feature_0001 x feature_0005
feature_0004 x feature_0006
feature_0002 x feature_0004
feature_0003 x feature_0004
feature_0002 x feature_0005
feature_0005 x feature_0006
feature_0003 x feature_0005
feature_0001 x feature_0004
feature_0004 x feature_0005
0
0.02
0.04
0.06

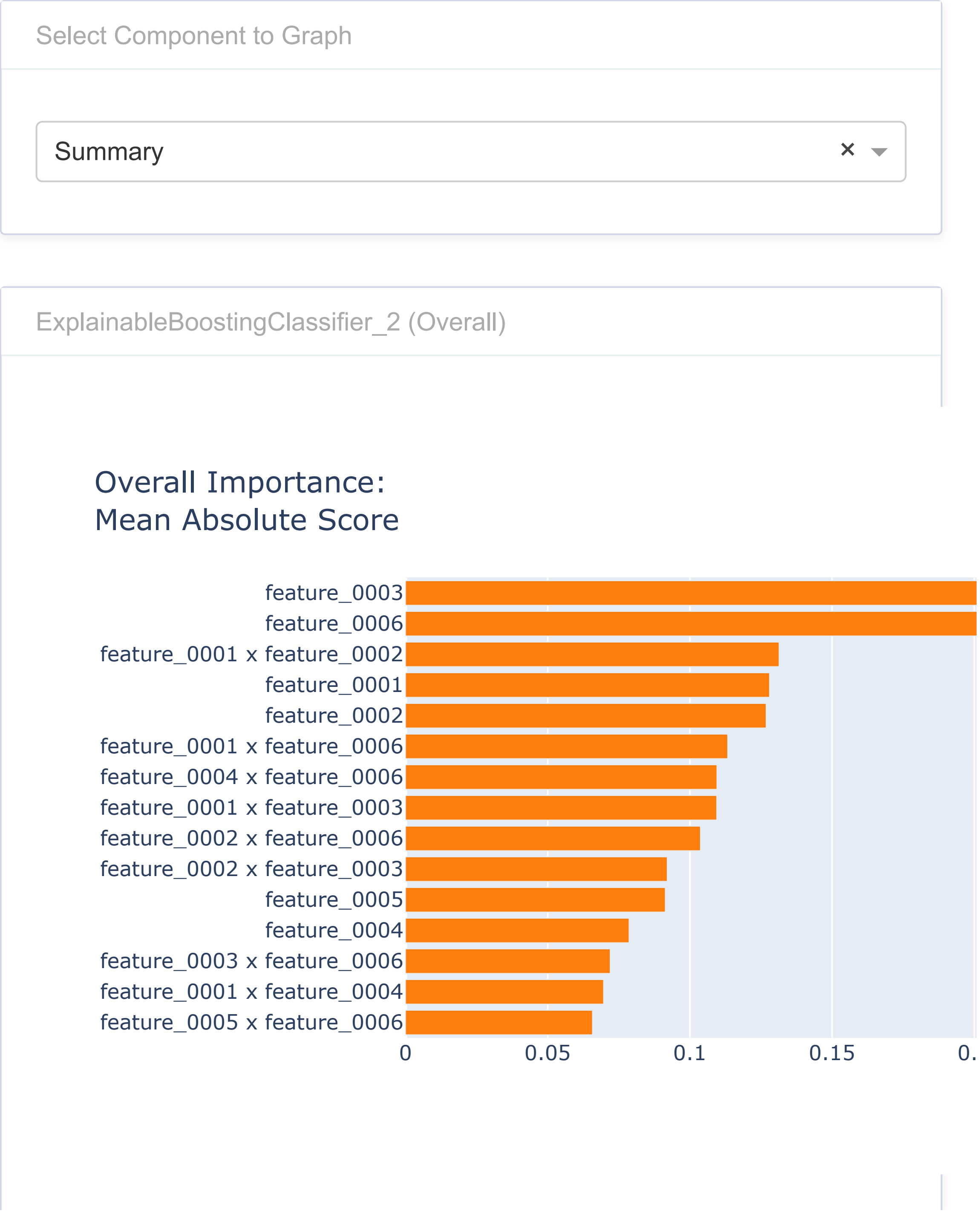


Select Component to Graph

Summary

ExplainableBoostingClassifier_1 (Overall)

Overall Importance:
Mean Absolute Score
feature_0003
feature_0004
feature_0006
feature_0002
feature_0001
feature_0005
feature_0001 x feature_0003
feature_0005 x feature_0006
feature_0001 x feature_0004
feature_0003 x feature_0006
feature_0004 x feature_0006
feature_0001 x feature_0006
feature_0002 x feature_0003
feature_0001 x feature_0005
feature_0001 x feature_0002
0
0.02
0.04
0.06
0.08



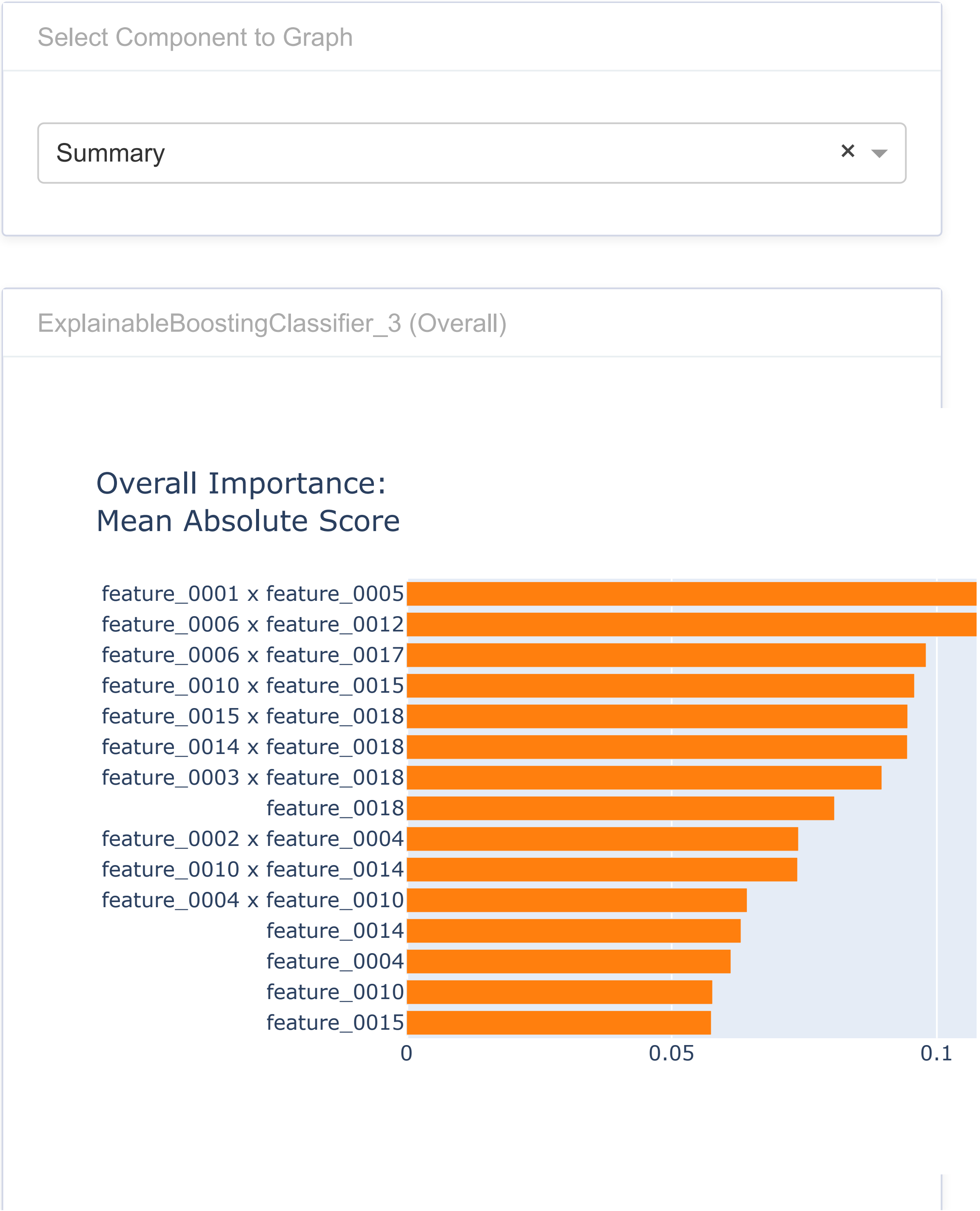
Select Component to Graph

Summary

ExplainableBoostingClassifier_2 (Overall)

Overall Importance:
Mean Absolute Score

×
feature_0003
feature_0006
feature_0001 x feature_0002
feature_0001
feature_0002
feature_0001 x feature_0006
feature_0004 x feature_0006
feature_0001 x feature_0003
feature_0002 x feature_0006
feature_0002 x feature_0003
feature_0005
feature_0004
feature_0003 x feature_0006
feature_0001 x feature_0004
feature_0005 x feature_0006
0
0.05
0.1
0.15
0.

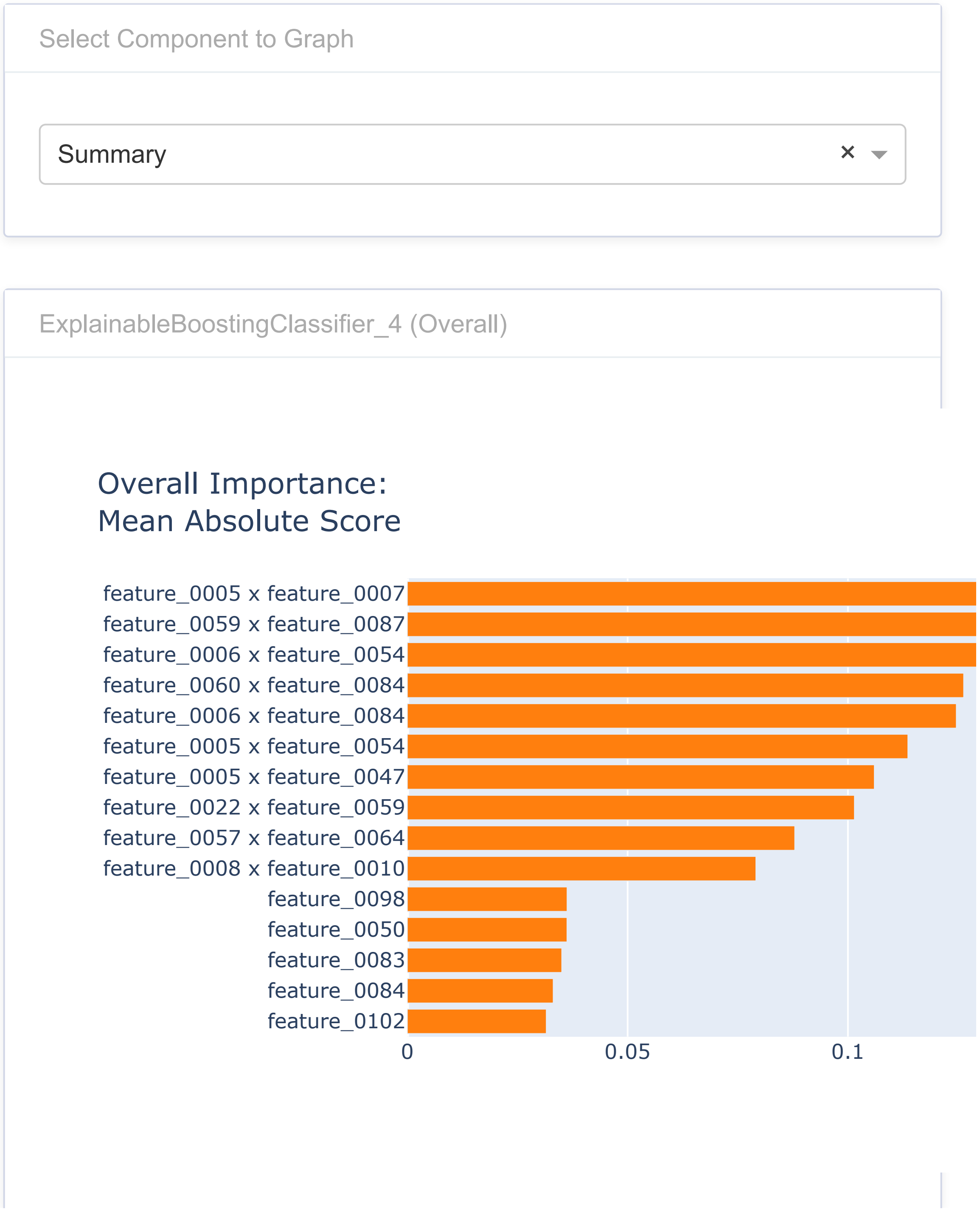


Select Component to Graph

Summary

ExplainableBoostingClassifier_3 (Overall)

Overall Importance:
Mean Absolute Score
feature_0001 x feature_0005
feature_0006 x feature_0012
feature_0006 x feature_0017
feature_0010 x feature_0015
feature_0015 x feature_0018
feature_0014 x feature_0018
feature_0003 x feature_0018
feature_0018
feature_0002 x feature_0004
feature_0010 x feature_0014
feature_0004 x feature_0010
feature_0014
feature_0004
feature_0010
feature_0015
0
0.05
0.1



Select Component to Graph

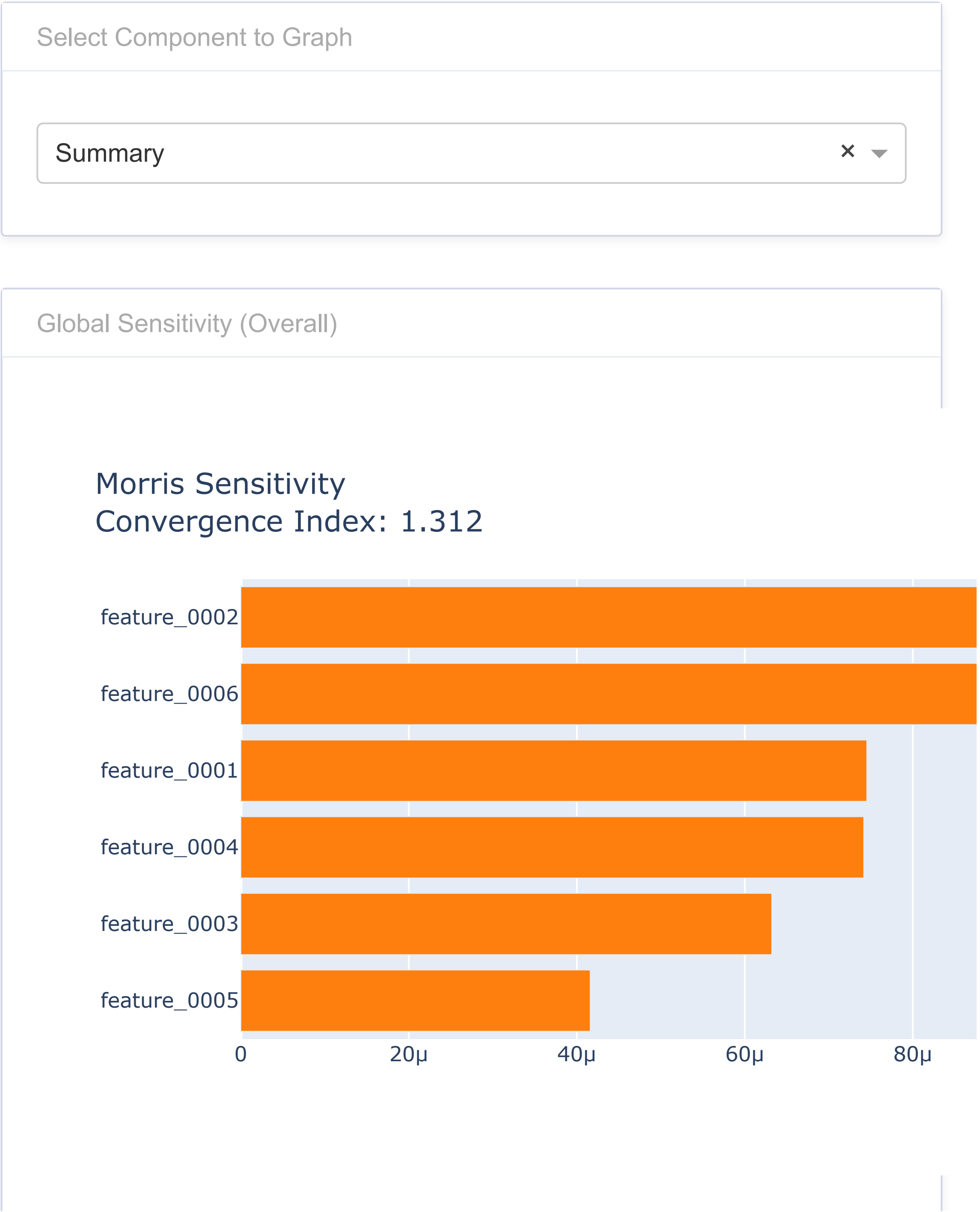
Summary

ExplainableBoostingClassifier_4 (Overall)

Overall Importance:
Mean Absolute Score
feature_0005 x feature_0007
feature_0059 x feature_0087
feature_0006 x feature_0054
feature_0060 x feature_0084
feature_0006 x feature_0084
feature_0005 x feature_0054
feature_0005 x feature_0047
feature_0022 x feature_0059
feature_0057 x feature_0064
feature_0008 x feature_0010
feature_0098
feature_0050
feature_0083
feature_0084
feature_0102
0
0.05
0.1

In [ ]:

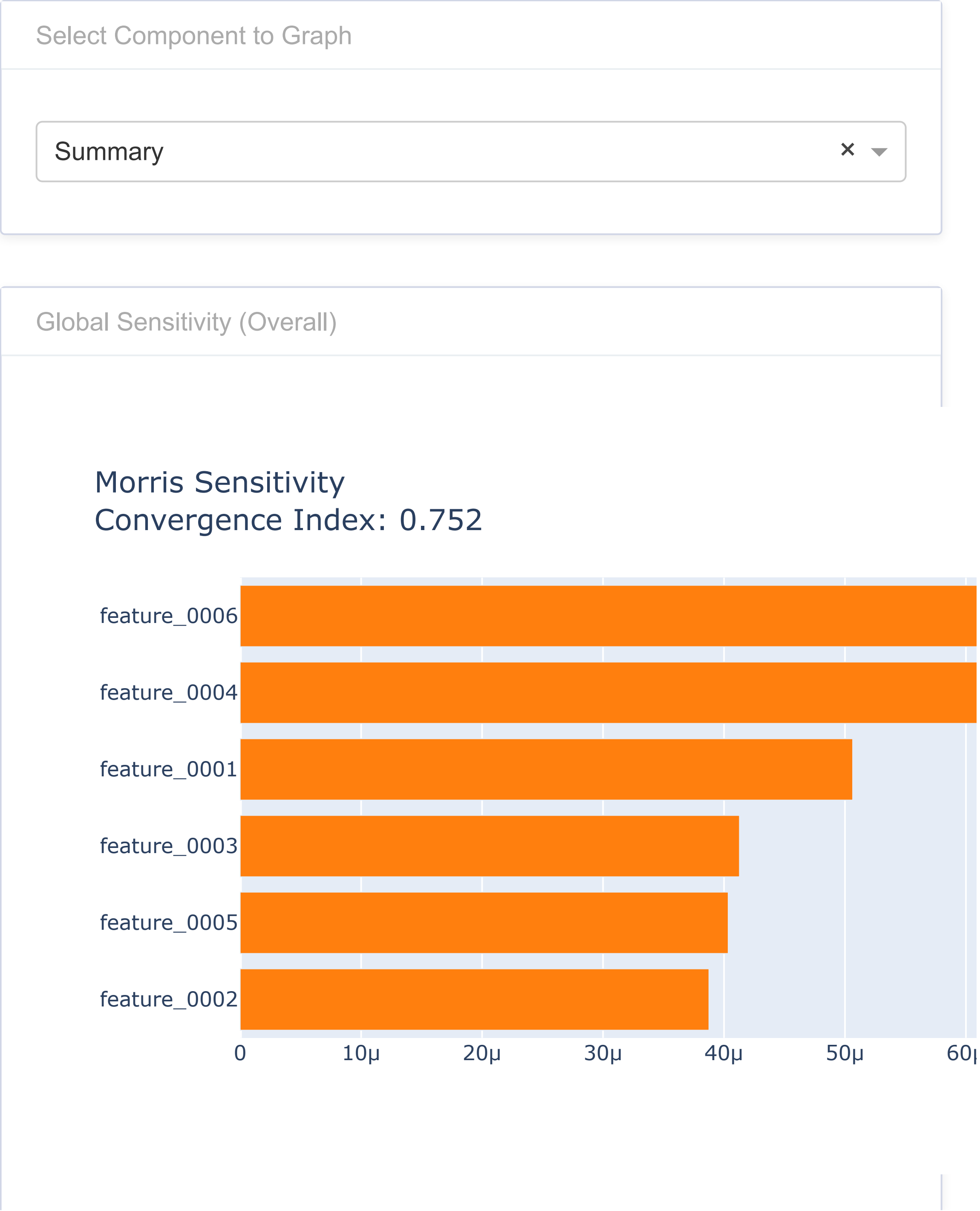
```
for title in feature_sets:
    dfc = feature_sets[title]
    not_zero_ind = ~(dfc == 0).all(axis=1)

    dfc = dfc.loc[not_zero_ind]
    labels = dfc['label'].loc[not_zero_ind]

    not_nan_index = ~dfc.isna().any(axis=1)
    dfc = dfc[not_nan_index]
    labels = labels[not_nan_index]

    scaler = StandardScaler()
    scaled_samples = scaler.fit_transform(dfc.iloc[:,:-1])

    X_train, X_test, y_train, y_test = train_test_split(scaled_samples, labels, test_size=0.2, random_state=42, stratif

    blackbox_model.fit(X_train, y_train)
    try:
        sensitivity = MorrisSensitivity(predict_fn=blackbox_model.predict_proba, data=X_train)
        sensitivity_global = sensitivity.explain_global(name="Global Sensitivity")

        show(sensitivity_global)

    except ValueError:
        print("zero-size array to reduction operation maximum which has no identity")
```



Select Component to Graph

Summary

Global Sensitivity (Overall)

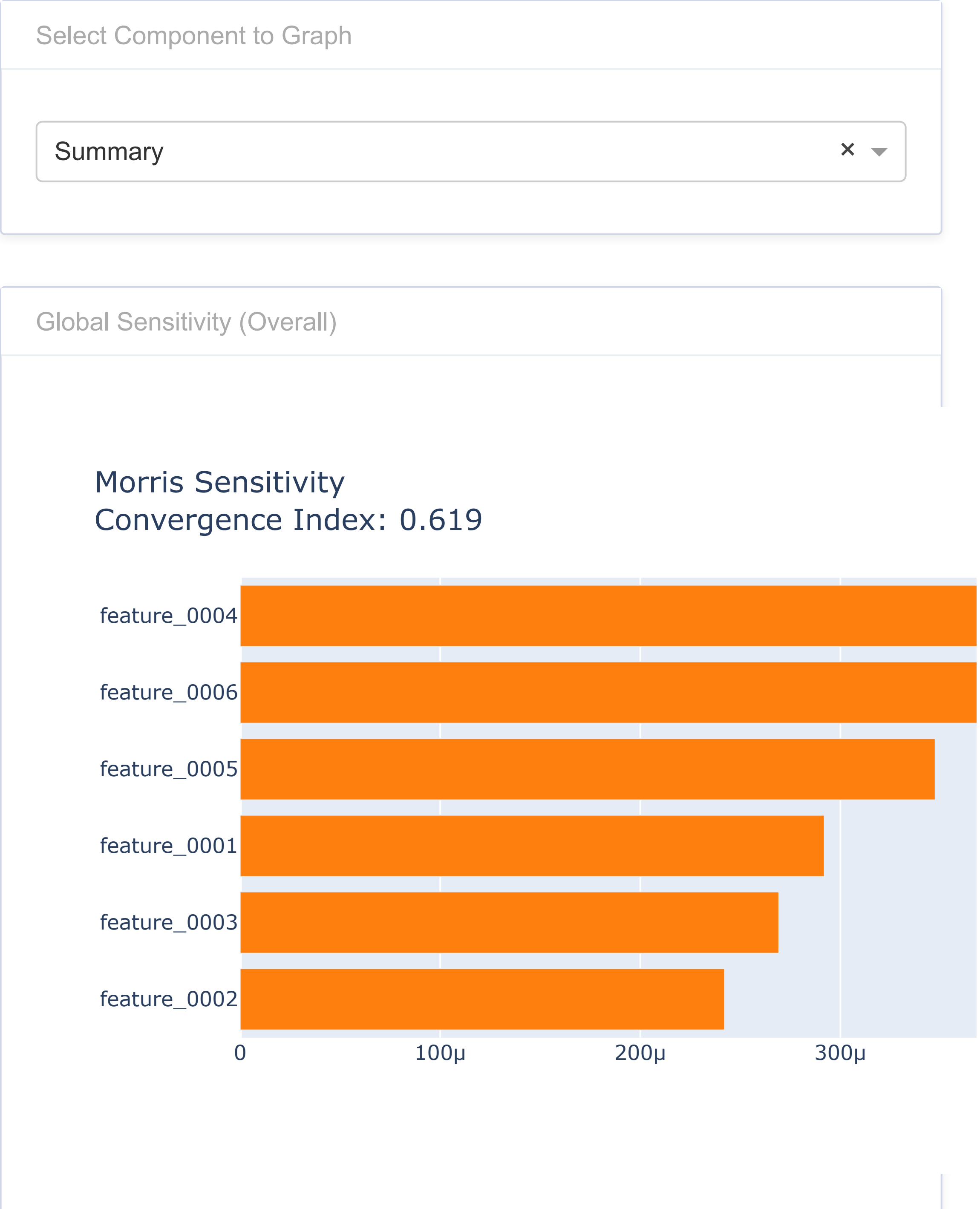


Select Component to Graph

Summary

Global Sensitivity (Overall)

Morris Sensitivity
Convergence Index: 0.752

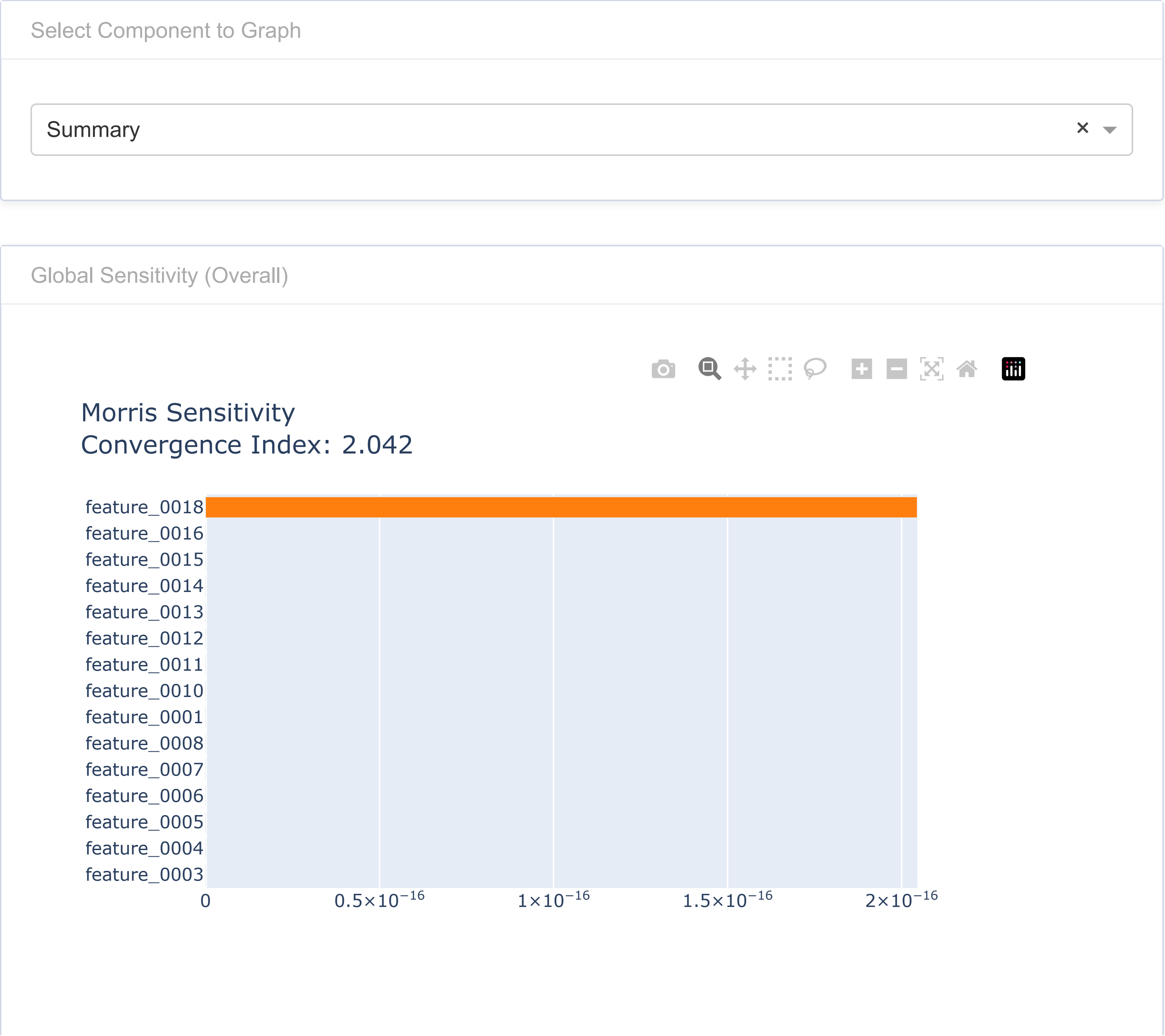


Select Component to Graph

Summary

Global Sensitivity (Overall)

Morris Sensitivity
Convergence Index: 0.619



Select Component to Graph

Summary

Global Sensitivity (Overall)

Morris Sensitivity
Convergence Index: 2.042

```
zero-size array to reduction operation maximum which has no identity
```

```
In [ ]:
```

```
In [ ]:
```