Importing module

```
!pip install xai
!pip install xai_data
```

```
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
    Requirement already satisfied: xai in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: scipy<1.7.1 in /usr/local/lib/python3.7/dist-p
    Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/di
    Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: pandas<1.3.4 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: pyparsing in /usr/local/lib/python3.7/dist-pac
    Requirement already satisfied: numpy<1.21.3 in /usr/local/lib/python3.7/dist-
    Requirement already satisfied: cycler in /usr/local/lib/python3.7/dist-packag
    Requirement already satisfied: matplotlib<3.4.3 in /usr/local/lib/python3.7/d
    Requirement already satisfied: kiwisolver in /usr/local/lib/python3.7/dist-pa
    Requirement already satisfied: scikit-learn<1.0.1 in /usr/local/lib/python3.7
    Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/
    Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3
    Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
    ERROR: Could not find a version that satisfies the requirement xai_data (fror
    ERROR: No matching distribution found for xai_data
```

```
import sys, os
import pandas as pd
import numpy as np
from collections import defaultdict
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.pipeline import make_pipeline

sys.path.append("..")

import xai
import xai.data
```

Columns in Dataset

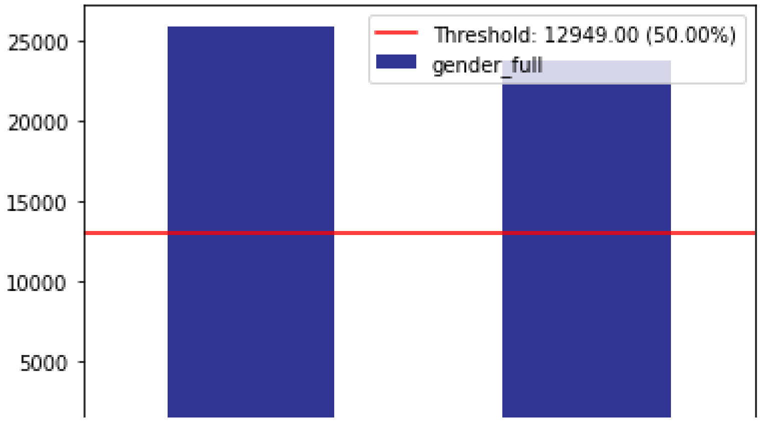
```
cat_columns = ["cat_columns", "job_title", "gender_full", "termreason_desc", "termtype_
                   "STATUS", "BUSINESS_UNIT"]
```

```
# Loading the dataset
dataframe = pd.read_csv("/content/drive/MyDrive/content/MFG10YearTerminationData.csv")
dataframe.tail()
```

	EmployeeID	recorddate_key	birthdate_key	orighiredate_key	ter
49648	8258	12/1/2015 0:00	5/28/1994	8/19/2013	
49649	8264	8/1/2013 0:00	6/13/1994	8/27/2013	
49650	8279	12/1/2015 0:00	7/18/1994	9/15/2013	
49651	8296	12/1/2013 0:00	9/2/1994	10/9/2013	
49652	8321	12/1/2014 0:00	11/28/1994	11/24/2013	

```
label = "loan"
protected_data = ["termreason_desc", "gender_full", "termtype_desc"]
```

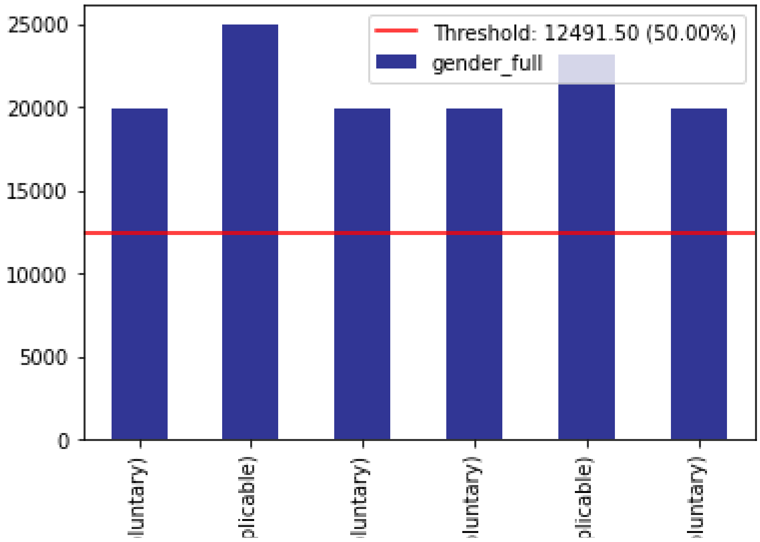
```
df_groups = xai.imbalance_plot(dataframe, "gender_full", categorical_cols=cat_columns)
```



Imbalance and balance data in term type description as per gender

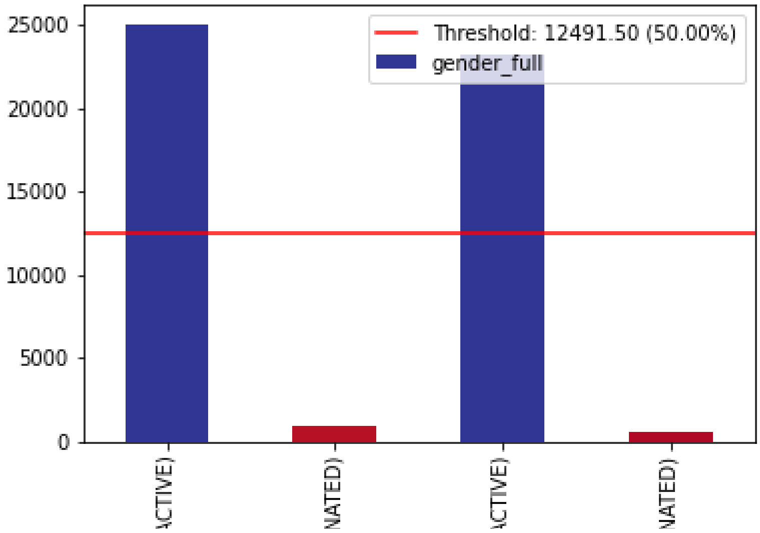
```
groups = xai.imbalance_plot(dataframe, "gender_full", "termtype_desc", categorical_cols
```

```
bal_dataframe = xai.balance(dataframe, "gender_full", "termtype_desc", upsample=0.8, ca
```

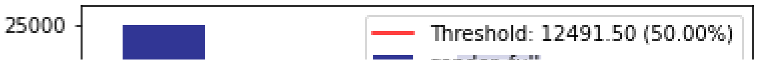


Imbalance and balance data of gender and status

```
groups = xai.imbalance_plot(dataframe, "gender_full", "STATUS", categorical_cols=cat_co
```

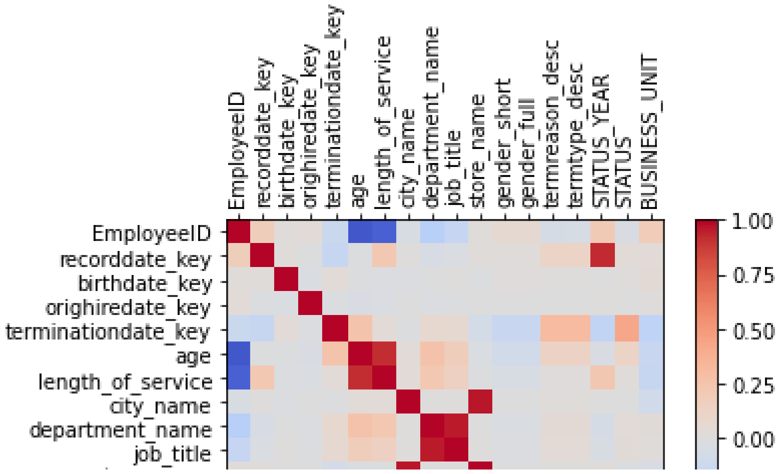


```
bal_dataframe = xai.balance(dataframe, "gender_full", "STATUS", upsample=0.8, categoric
```



Correlation using Xai library

```
_ = xai.correlations(dataframe, include_categorical=True, plot_type="matrix")
```



```
proc_dataframe = xai.normalize_numeric(bal_dataframe)
proc_dataframe = xai.convert_categories(proc_dataframe)
x = proc_dataframe.drop("STATUS", axis=1)
y = proc_dataframe["STATUS"]

x_train, y_train, x_test, y_test, train_idx, test_idx = \
    xai.balanced_train_test_split(
            x, y, "gender_full",
            min_per_group=300,
            max_per_group=500,
            categorical_cols=cat_columns)

x_train_for_display = bal_dataframe[train_idx]
x_test_data_for_display = bal_dataframe[test_idx]

print("Total number of rows: ", x_test.shape[0])

dataframe_test = x_test_data_for_display.copy()
dataframe_test["STATUS"] = y_test

_= xai.imbalance_plot(dataframe_test, "gender_full", "STATUS", categorical_cols=cat_col
```

Machine Learning models

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

clf=LogisticRegression()

#fitting the data
clf.fit(x_train, y_train)

#predicting
pred = clf.predict(x_test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

```
probabilities = clf.predict(x_test)
```

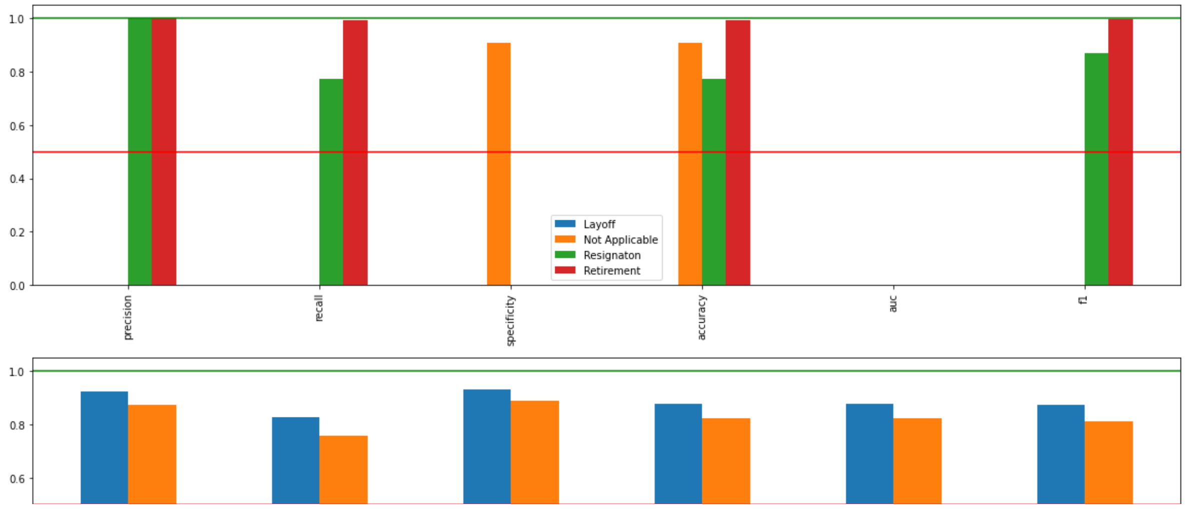
```
probabilities_test = clf.predict_proba(x_test)
```

```
x_test_data_for_display.columns
```

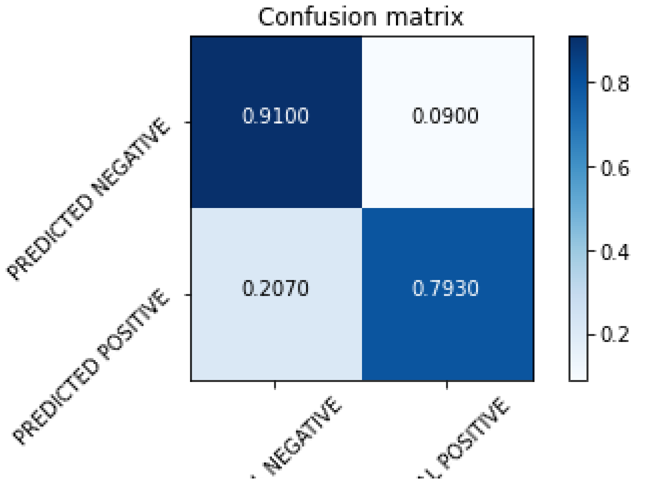
```
Index(['EmployeeID', 'recorddate_key', 'birthdate_key', 'orighiredate_key',
       'terminationdate_key', 'age', 'length_of_service', 'city_name',
       'department_name', 'job_title', 'store_name', 'gender_short',
       'gender_full', 'termreason_desc', 'termtype_desc', 'STATUS_YEAR',
       'STATUS', 'BUSINESS_UNIT'],
      dtype='object')
```

```
_ = xai.metrics_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=["gender_full", "termtype_desc"],
    categorical_cols=cat_columns)
```

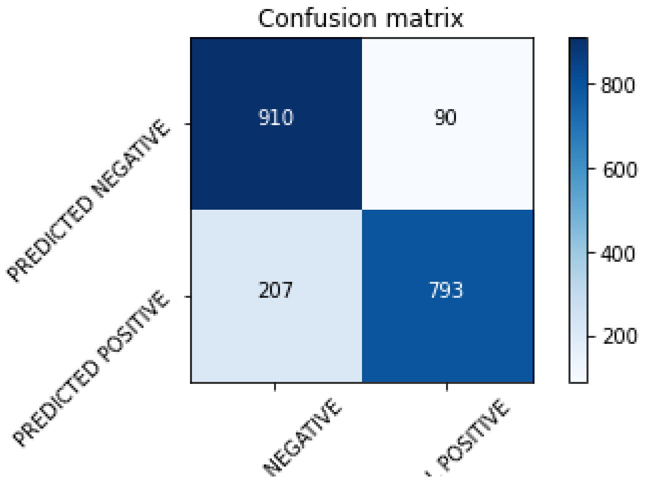
```
_ = [xai.metrics_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=[p],
    categorical_cols=cat_columns) for p in protected_data]
```



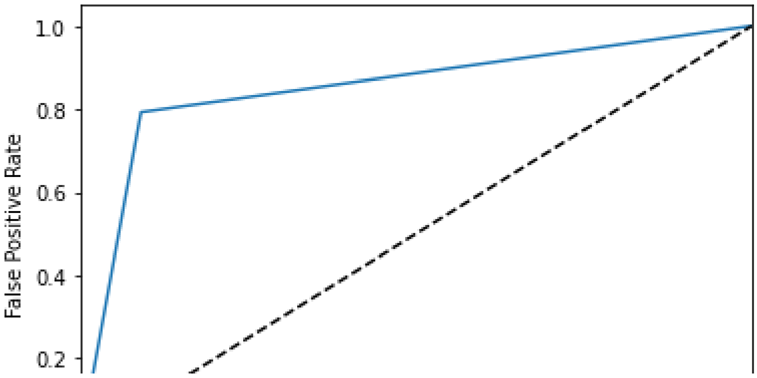
```
xai.confusion_matrix_plot(y_test, pred)
```



```
xai.confusion_matrix_plot(y_test, pred, scaled=False)
```



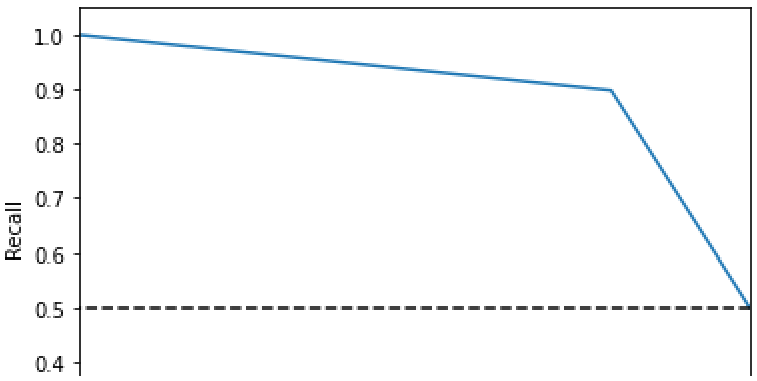
```
_ = xai.roc_plot(y_test, probabilities)
```



```
_ = [xai.roc_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=[p],
    categorical_cols=cat_columns) for p in protected_data]
```

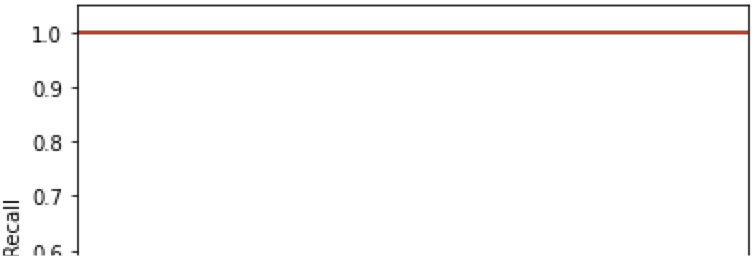
```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:1000
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
```

```
_= xai.pr_plot(y_test, probabilities)
```



```
_ = [xai.pr_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=[p],
    categorical_cols=cat_columns) for p in protected_data]
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:863:
  recall = tps / tps[-1]
```

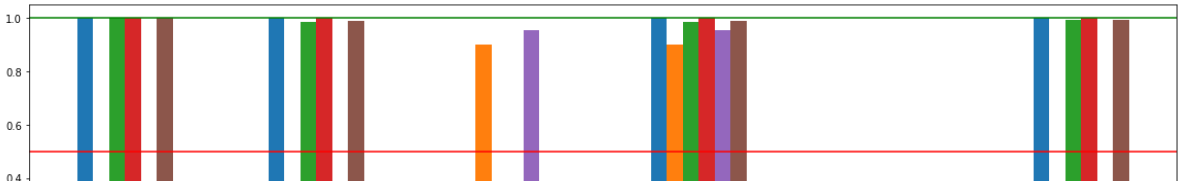


Support Vector Machine

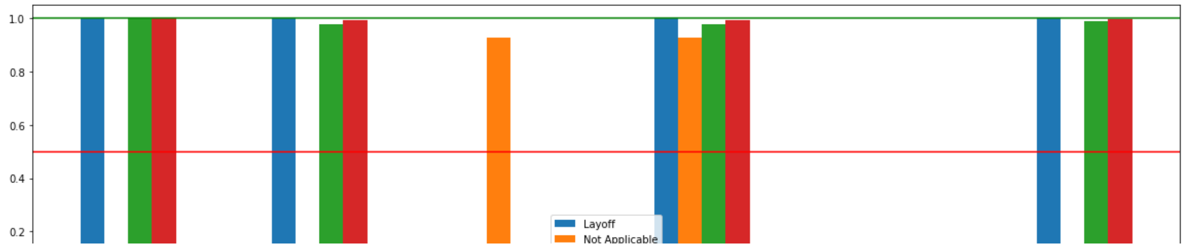
```
from sklearn.svm import SVC

clf=SVC()

#fitting the data
clf.fit(x_train, y_train)

#predicting
pred = clf.predict(x_test)
```

```
probabilities = clf.predict(x_test)
```

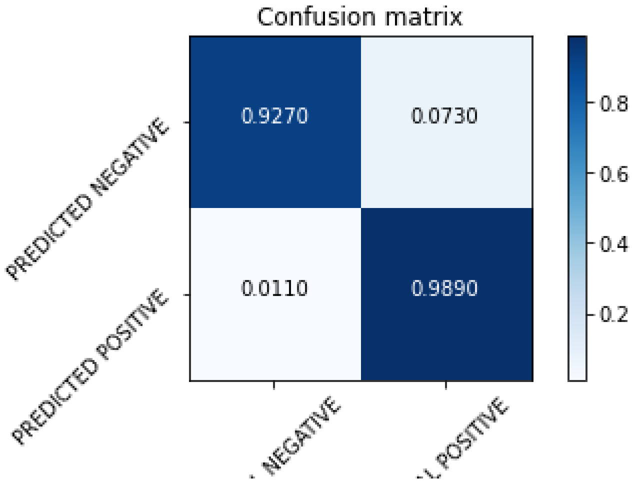
```
_ = xai.metrics_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=["gender_full", "termtype_desc"],
    categorical_cols=cat_columns)
```



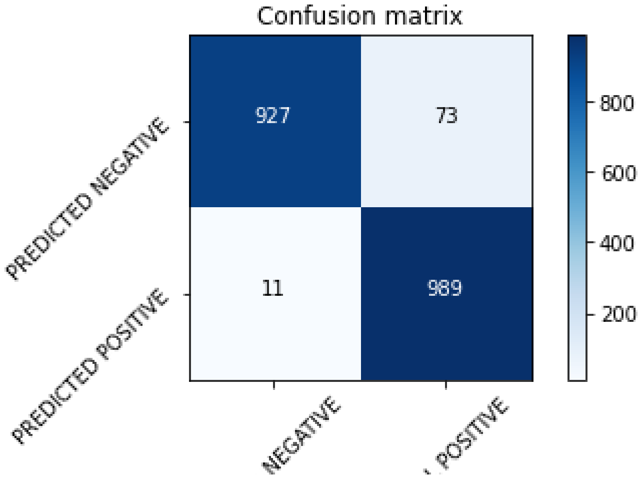
```
_ = [xai.metrics_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=[p],
    categorical_cols=cat_columns) for p in protected_data]
```



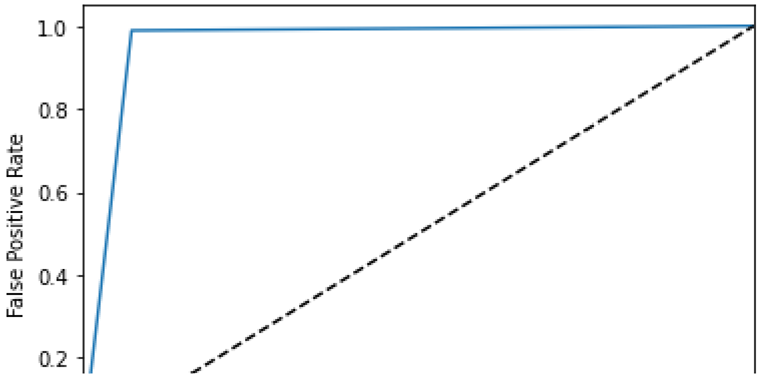
```
xai.confusion_matrix_plot(y_test, pred)
```



```
xai.confusion_matrix_plot(y_test, pred, scaled=False)
```

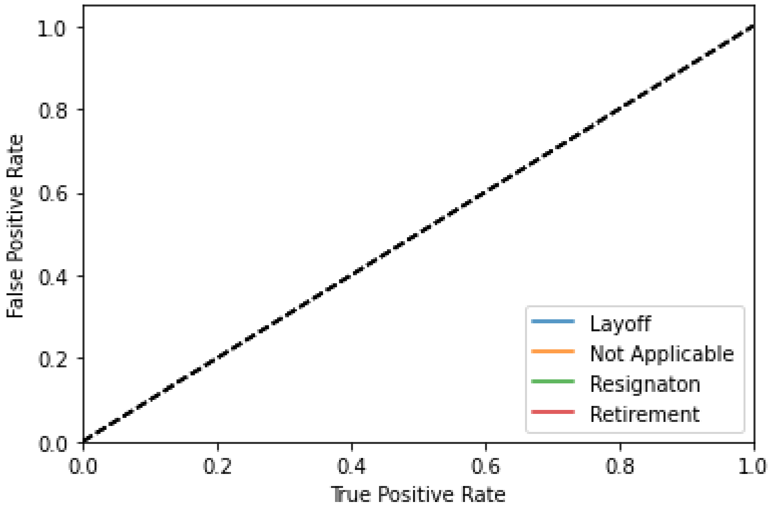


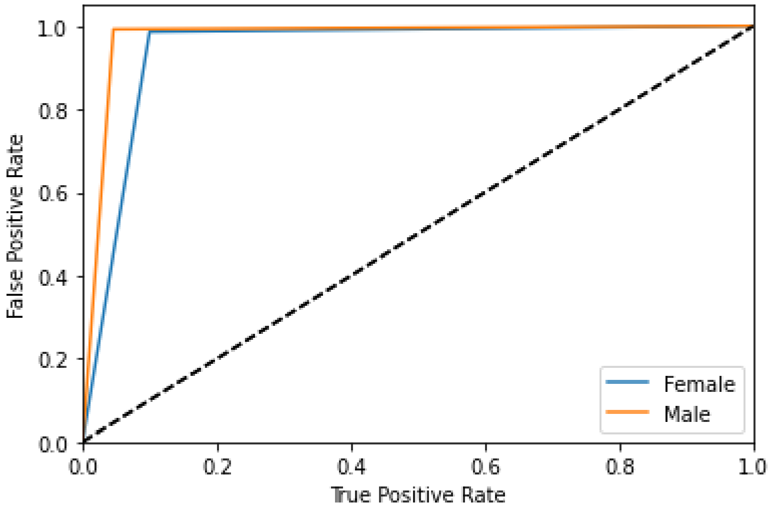
```
_ = xai.roc_plot(y_test, probabilities)
```



```
_ = [xai.roc_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=[p],
    categorical_cols=cat_columns) for p in protected_data]
```

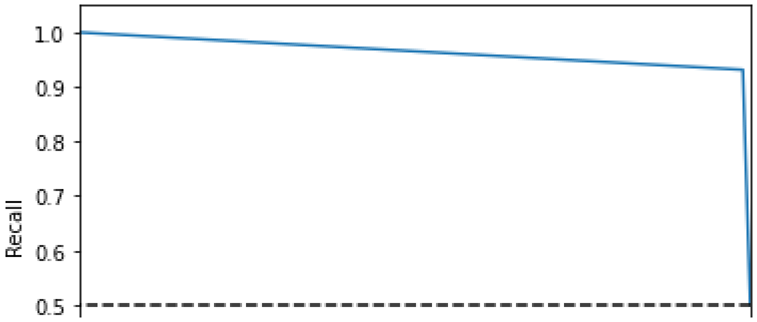
```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:1000
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
```





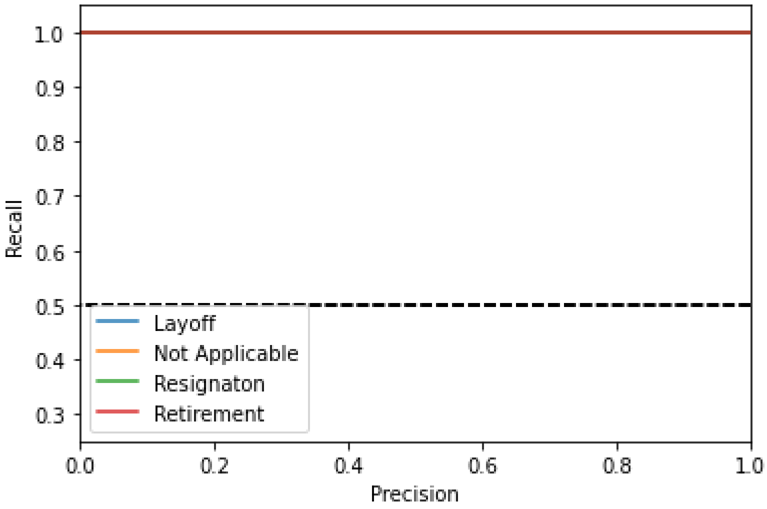
```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:1000
  UndefinedMetricWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:991:
  UndefinedMetricWarning,
```

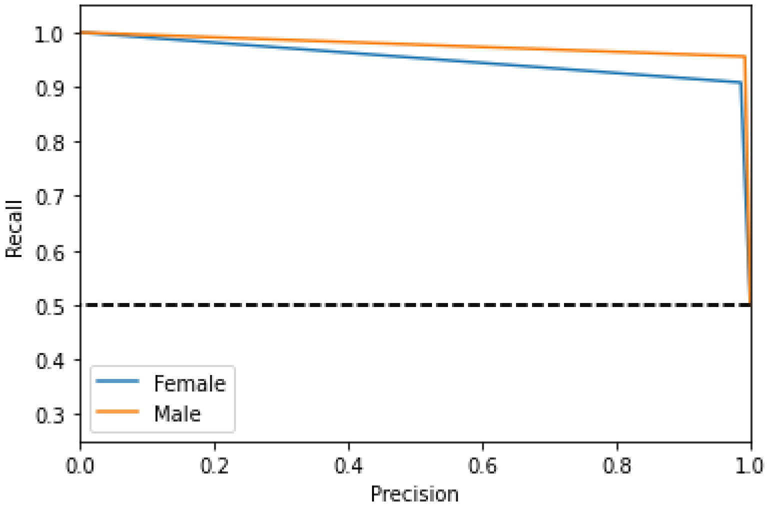
```
_= xai.pr_plot(y_test, probabilities)
```



```
_ = [xai.pr_plot(
    y_test,
    probabilities,
    df=x_test_data_for_display,
    cross_cols=[p],
    categorical_cols=cat_columns) for p in protected_data]
```

```
  /usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:863:
    recall = tps / tps[-1]
```





```
  /usr/local/lib/python3.7/dist-packages/sklearn/metrics/_ranking.py:863:
    recall = tps / tps[-1]
```