```
!pip install fairlearn
```

```
Collecting fairlearn
  Downloading fairlearn-0.12.0-py3-none-any.whl.metadata (7.0 kB)
Requirement already satisfied: numpy>=1.24.4 in /usr/local/lib/python3.11/dist-packages (from fairlearn) (2.0.2)
Requirement already satisfied: pandas>=2.0.3 in /usr/local/lib/python3.11/dist-packages (from fairlearn) (2.2.2)
Requirement already satisfied: scikit-learn>=1.2.1 in /usr/local/lib/python3.11/dist-packages (from fairlearn) (1.6.1)
Requirement already satisfied: scipy>=1.9.3 in /usr/local/lib/python3.11/dist-packages (from fairlearn) (1.16.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=2.0.3->fairlearn) (2.9.0.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=2.0.3->fairlearn) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=2.0.3->fairlearn) (2025.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.2.1->fairlearn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.2.1->fairlearn) (3.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas>=2.0.3->fairlear
Downloading fairlearn-0.12.0-py3-none-any.whl (240 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 240.0/240.0 kB 4.4 MB/s eta 0:00:00
Installing collected packages: fairlearn
Successfully installed fairlearn-0.12.0
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from fairlearn.metrics import (
    MetricFrame,
    selection_rate,
    demographic_parity_difference,
    equalized_odds_difference
)
from sklearn.metrics import accuracy_score
from fairlearn.reductions import ExponentiatedGradient, DemographicParity

np.random.seed(42)
```

```
n = 200
experience = np.random.choice(["low", "medium", "high"], n)
test_score = np.random.randint(0, 11, n)
interview_score = np.random.randint(0, 11, n)
gender = np.random.choice(["male", "female"], n)

hired_balanced = [np.random.choice([0, 1], p=[0.5, 0.5]) for _ in range(n)]

df_balanced = pd.DataFrame({
    "experience": experience,
    "test_score": test_score,
    "interview_score": interview_score,
    "gender": gender,
    "hired": hired_balanced
})
```

```
hired_imbalanced = []
for g in gender:
    if g == "male":
        hired_imbalanced.append(np.random.choice([0, 1], p=[0.3, 0.7]))
    else:
        hired_imbalanced.append(np.random.choice([0, 1], p=[0.7, 0.3]))

df_imbalanced = pd.DataFrame({
    "experience": experience,
    "test_score": test_score,
    "interview_score": interview_score,
    "gender": gender,
    "hired": hired_imbalanced
})
```

```
def evaluate_fairness(df, title, model):
    X = df[["experience", "test_score", "interview_score", "gender"]]
    y = df["hired"]
    sensitive_feature = df["gender"]
    X_train, X_test, y_train, y_test, sf_train, sf_test = train_test_split(
        X, y, sensitive_feature, test_size=0.3, random_state=42
    )

    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    metric_frame = MetricFrame(
        metrics={"accuracy": accuracy_score, "selection_rate": selection_rate},
        y_true=y_test,
        y_pred=y_pred,
        sensitive_features=sf_test
    )

    dp_diff = demographic_parity_difference(y_test, y_pred, sensitive_features=sf_test)
    eo_diff = equalized_odds_difference(y_test, y_pred, sensitive_features=sf_test)

    print(f"\n===== {title} Dataset =====")
    print("Overall Metrics:", metric_frame.overall)
    print("By Group Metrics:\n", metric_frame.by_group)
    print(f"Demographic Parity Difference: {dp_diff:.4f}")
    print(f"Equalized Odds Difference: {eo_diff:.4f}")



pipeline = Pipeline([
    ("encoder", OneHotEncoder(drop="first")),
    ("model", LogisticRegression(solver="liblinear"))
])

evaluate_fairness(df_balanced, "Balanced", pipeline)
evaluate_fairness(df_imbalanced, "Imbalanced", pipeline)
```

```
===== Balanced Dataset =====
Overall Metrics: accuracy          0.483333
selection_rate    0.416667
dtype: float64
By Group Metrics:
         accuracy  selection_rate
gender
female  0.481481        0.296296
male    0.484848        0.515152
Demographic Parity Difference: 0.2189
Equalized Odds Difference: 0.2500

===== Imbalanced Dataset =====
Overall Metrics: accuracy          0.700000
selection_rate    0.633333
dtype: float64
By Group Metrics:
         accuracy  selection_rate
gender
female  0.518519        0.222222
male    0.848485        0.969697
Demographic Parity Difference: 0.7475
Equalized Odds Difference: 0.9655
```