```
! pip install evidently
```

```
Collecting mypy-extensions>=0.3.0 (from typing-inspect>=0.9.0->evidently)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.10/dist-packages (from uvicorn>=0.22.0->uvicorn[standard]>=0.22.0->evidently) (0.14.0)
Collecting httptools>=0.6.3 (from uvicorn[standard]>=0.22.0->evidently)
  Downloading httptools-0.6.4-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.6 kB)
Collecting python-dotenv>=0.13 (from uvicorn[standard]>=0.22.0->evidently)
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Collecting uvloop!=0.15.0,!=0.15.1,>=0.14.0 (from uvicorn[standard]>=0.22.0->evidently)
  Downloading uvloop-0.21.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Collecting watchfiles>=0.13 (from uvicorn[standard]>=0.22.0->evidently)
  Downloading watchfiles-1.0.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Collecting websockets>=10.4 (from uvicorn[standard]>=0.22.0->evidently)
  Downloading websockets-14.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.7 kB)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio>=3->litestar>=2.8.3->evidently) (1.3.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.12->cryptography>=43.0.1->evidently) (2.22)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx>=0.22->litestar>=2.8.3->evidently) (1.0.7)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=13->evidently) (0.1.2)
Collecting faker (from polyfactory>=2.6.3->litestar>=2.8.3->evidently)
  Downloading Faker-33.1.0-py3-none-any.whl.metadata (15 kB)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=1.3.5->pandas[parquet]>=1.3.5->evidently) (1.16.0)
Downloading evidently-0.4.40-py3-none-any.whl (3.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.5/3.5 MB 24.8 MB/s eta 0:00:00
Downloading deprecation-2.1.0-py2.py3-none-any.whl (11 kB)
Downloading dynaconf-3.2.6-py2.py3-none-any.whl (231 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 231.1/231.1 kB 17.5 MB/s eta 0:00:00
Downloading iterative_telemetry-0.0.9-py3-none-any.whl (10 kB)
Downloading litestar-2.13.0-py3-none-any.whl (555 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 555.5/555.5 kB 30.3 MB/s eta 0:00:00
Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Downloading ujson-5.10.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (53 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.6/53.6 kB 4.1 MB/s eta 0:00:00
Downloading uuid6-2024.7.10-py3-none-any.whl (6.4 kB)
Downloading uvicorn-0.32.1-py3-none-any.whl (63 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 63.8/63.8 kB 5.3 MB/s eta 0:00:00
Downloading watchdog-6.0.0-py3-none-manylinux2014_x86_64.whl (79 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.1/79.1 kB 6.3 MB/s eta 0:00:00
Downloading httptools-0.6.4-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (442 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 442.1/442.1 kB 21.9 MB/s eta 0:00:00
Downloading litestar_htmx-0.3.0-py3-none-any.whl (8.9 kB)
Downloading msgspec-0.18.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (210 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 210.3/210.3 kB 16.0 MB/s eta 0:00:00
Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Downloading polyfactory-2.18.1-py3-none-any.whl (59 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.3/59.3 kB 4.9 MB/s eta 0:00:00
Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Downloading uvloop-0.21.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.8/3.8 MB 62.9 MB/s eta 0:00:00
Downloading watchfiles-1.0.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (442 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 442.6/442.6 kB 28.8 MB/s eta 0:00:00
Downloading websockets-14.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (168 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 168.2/168.2 kB 13.2 MB/s eta 0:00:00
Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Downloading rich_click-1.8.4-py3-none-any.whl (35 kB)
Downloading Faker-33.1.0-py3-none-any.whl (1.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 58.9 MB/s eta 0:00:00
Installing collected packages: appdirs, websockets, watchdog, uvloop, uvicorn, uuid6, ujson, python-dotenv, mypy-extensions, msgspec, httptools, dynaconf, deprecation, watchfiles, t
Successfully installed appdirs-1.4.4 deprecation-2.1.0 dynaconf-3.2.6 evidently-0.4.40 faker-33.1.0 httptools-0.6.4 iterative-telemetry-0.0.9 litestar-2.13.0 litestar-htmx-0.3.0 ms
```

```
! jupyter nbextension install --sys-prefix --symlink --overwrite --py evidently
```

```
Installing /usr/local/lib/python3.10/dist-packages/evidently/nbextension/static -> evidently
Symlinking: /usr/share/jupyter/nbextensions/evidently -> /usr/local/lib/python3.10/dist-packages/evidently/nbextension/static
- Validating: OK

    To initialize this nbextension in the browser every time the notebook (or other app) loads:

          jupyter nbextension enable evidently --py --sys-prefix
```

```
! jupyter nbextension enable evidently --py --sys-prefix
```

```
Enabling notebook extension evidently/extension...
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
Paths used for configuration of notebook:

      - Validating: OK
Paths used for configuration of notebook:
        /usr/etc/jupyter/nbconfig/notebook.json
```

```
try:
    import evidently
except:
    !pip install git+https://github.com/evidentlyai/evidently.git


import pandas as pd
import numpy as np

from sklearn import datasets, ensemble, model_selection

from evidently import ColumnMapping
from evidently.test_suite import TestSuite
from evidently.tests import *
from evidently.test_preset import NoTargetPerformanceTestPreset
from evidently.test_preset import DataQualityTestPreset
from evidently.test_preset import DataStabilityTestPreset
from evidently.test_preset import DataDriftTestPreset
from evidently.test_preset import RegressionTestPreset
from evidently.test_preset import MulticlassClassificationTestPreset
from evidently.test_preset import BinaryClassificationTopKTestPreset
from evidently.test_preset import BinaryClassificationTestPreset
from evidently.report import Report
from evidently.metric_preset import DataDriftPreset
from evidently.metric_preset import DataQualityPreset
from evidently.metric_preset import RegressionPreset
from evidently.metric_preset import ClassificationPreset
from evidently.metric_preset import TargetDriftPreset


import warnings
warnings.filterwarnings('ignore')
```

1. Dataset for Data Quality and Integrity

```
#Dataset for Data Quality and Integrity

adult_data = datasets.fetch_openml(name='adult', version=2, as_frame=True)
adult = adult_data.frame
```

```
adult.head()
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K

Next steps: Generate code with `adult` View recommended plots New interactive sheet

```
adult.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             48842 non-null  int64
 1   workclass       46043 non-null  category
 2   fnlwgt          48842 non-null  int64
 3   education       48842 non-null  category
 4   education-num   48842 non-null  int64
 5   marital-status  48842 non-null  category
 6   occupation      46033 non-null  category
 7   relationship    48842 non-null  category
 8   race            48842 non-null  category
 9   sex             48842 non-null  category
 10  capital-gain    48842 non-null  int64
 11  capital-loss    48842 non-null  int64
 12  hours-per-week  48842 non-null  int64
 13  native-country  47985 non-null  category
 14  class           48842 non-null  category
dtypes: category(9), int64(6)
memory usage: 2.7 MB
```

```
adult.education.value_counts()
```

education	count
HS-grad	15784
Some-college	10878
Bachelors	8025
Masters	2657
Assoc-voc	2061
11th	1812
Assoc-acdm	1601
10th	1389
7th-8th	955
Prof-school	834
9th	756
12th	657
Doctorate	594
5th-6th	509
1st-4th	247
Preschool	83

dtype: int64

```
adult_ref = adult[~adult.education.isin(['Some-college', 'HS-grad', 'Bachelors'])]
adult_cur = adult[adult.education.isin(['Some-college', 'HS-grad', 'Bachelors'])]

adult_cur.iloc[:2000, 3:5] = np.nan


adult_ref.shape , adult_cur.shape
```

```
((14155, 15), (34687, 15))
```

```
adult_cur.iloc[:2000 , 3:5]
```

	education	education-num
1	NaN	NaN
3	NaN	NaN
4	NaN	NaN
6	NaN	NaN
8	NaN	NaN
...	...	...
2813	NaN	NaN
2814	NaN	NaN
2815	NaN	NaN
2816	NaN	NaN
2817	NaN	NaN

2000 rows × 2 columns

2. Dataset for Regression

```
housing_data = datasets.fetch_california_housing(as_frame=True)
housing = housing_data.frame
```

```
housing.head()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

Next steps: Generate code with housing | View recommended plots | New interactive sheet

```
housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   MedInc       20640 non-null  float64
 1   HouseAge     20640 non-null  float64
 2   AveRooms     20640 non-null  float64
 3   AveBedrms    20640 non-null  float64
 4   Population   20640 non-null  float64
```

```
 5   AveOccup     20640 non-null  float64
 6   Latitude     20640 non-null  float64
 7   Longitude    20640 non-null  float64
 8   MedHouseVal  20640 non-null  float64
dtypes: float64(9)
memory usage: 1.4 MB
```

```
housing.shape
```

```
(20640, 9)
```

```
housing.rename(columns={'MedHouseVal': 'target'}, inplace=True)
housing['prediction'] = housing_data['target'].values + np.random.normal(0, 3, housing.shape[0])
```

```
housing.head()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	target	prediction
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526	6.567026
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585	1.235988
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521	0.777297
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413	4.034504
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422	-0.511835

Next steps: Generate code with housing | View recommended plots | New interactive sheet

```
housing_ref = housing.sample(n=5000, replace=False)
housing_cur = housing.sample(n=5000, replace=False)
```

```
housing_ref.describe()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	target	prediction
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.00000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	3.852558	28.444800	5.435699	1.097594	1451.12560	3.209105	35.617538	-119.550902	2.070375	2.058954
std	1.912136	12.570416	2.774699	0.487576	1229.41989	17.581415	2.143159	2.006215	1.151730	3.207989
min	0.499900	1.000000	1.130435	0.600000	17.00000	1.161290	32.550000	-124.300000	0.149990	-9.946500
25%	2.544000	18.000000	4.433418	1.007015	785.75000	2.427801	33.930000	-121.760000	1.201000	-0.074769
50%	3.515600	29.000000	5.220024	1.050188	1167.00000	2.822435	34.220000	-118.460000	1.804500	2.061700
75%	4.746375	37.000000	6.065431	1.099508	1742.50000	3.291755	37.730000	-117.990000	2.657250	4.188197
max	15.000100	52.000000	141.909091	25.636364	35682.00000	1243.333333	41.810000	-114.570000	5.000010	14.536211

```
housing_cur.describe()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	target	prediction
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	3.849595	28.676400	5.426195	1.098979	1416.916400	2.944566	35.625728	-119.563056	2.066844	2.030391
std	1.906162	12.496468	2.769500	0.492697	1171.739456	1.469293	2.129458	2.003548	1.154078	3.166248
min	0.499900	1.000000	1.130435	0.375000	3.000000	0.692308	32.550000	-124.250000	0.225000	-10.470058
25%	2.549825	18.000000	4.446270	1.006025	792.750000	2.426918	33.940000	-121.792500	1.186750	-0.131569
50%	3.522700	29.000000	5.210694	1.050584	1152.500000	2.809281	34.250000	-118.470000	1.792500	2.029548
75%	4.690950	37.000000	6.042019	1.100228	1716.250000	3.277146	37.710000	-118.020000	2.654250	4.160654
max	15.000100	52.000000	141.909091	25.636364	35682.000000	83.171429	41.950000	-114.560000	5.000010	14.345086

3. Dataset for Binary Probabilistic Classifcation

```
bcancer_data = datasets.load_breast_cancer(as_frame=True)
bcancer = bcancer_data.frame
```

```
bcancer.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	sym
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	(
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	(
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	(
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	(
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	(

5 rows × 31 columns

```
bcancer.target.value_counts()
```

target	count
1	357
0	212

dtype: int64

```
bcancer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
```

```
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   mean radius             569 non-null    float64
 1   mean texture            569 non-null    float64
 2   mean perimeter          569 non-null    float64
 3   mean area               569 non-null    float64
 4   mean smoothness         569 non-null    float64
 5   mean compactness        569 non-null    float64
 6   mean concavity          569 non-null    float64
 7   mean concave points     569 non-null    float64
 8   mean symmetry           569 non-null    float64
 9   mean fractal dimension  569 non-null    float64
 10  radius error            569 non-null    float64
 11  texture error           569 non-null    float64
 12  perimeter error         569 non-null    float64
 13  area error              569 non-null    float64
 14  smoothness error        569 non-null    float64
 15  compactness error       569 non-null    float64
 16  concavity error         569 non-null    float64
 17  concave points error    569 non-null    float64
 18  symmetry error          569 non-null    float64
 19  fractal dimension error 569 non-null    float64
 20  worst radius            569 non-null    float64
 21  worst texture           569 non-null    float64
 22  worst perimeter         569 non-null    float64
 23  worst area              569 non-null    float64
 24  worst smoothness        569 non-null    float64
 25  worst compactness       569 non-null    float64
 26  worst concavity         569 non-null    float64
 27  worst concave points    569 non-null    float64
 28  worst symmetry          569 non-null    float64
 29  worst fractal dimension 569 non-null    float64
 30  target                  569 non-null    int64  
dtypes: float64(30), int64(1)
memory usage: 137.9 KB
```

```
bcancer_ref = bcancer.sample(n=300, replace=False)
bcancer_cur = bcancer.sample(n=200, replace=False)

bcancer_label_ref = bcancer_ref.copy(deep=True)
bcancer_label_cur = bcancer_cur.copy(deep=True)
```

```
bcancer_data.feature_names.tolist()
```

```
['mean radius',
 'mean texture',
 'mean perimeter',
 'mean area',
 'mean smoothness',
 'mean compactness',
 'mean concavity',
 'mean concave points',
 'mean symmetry',
 'mean fractal dimension',
 'radius error',
 'texture error',
 'perimeter error',
 'area error',
 'smoothness error',
 'compactness error',
```

```
        'concavity error',
        'concave points error',
        'symmetry error',
        'fractal dimension error',
        'worst radius',
        'worst texture',
        'worst perimeter',
        'worst area',
        'worst smoothness',
        'worst compactness',
        'worst concavity',
        'worst concave points',
        'worst symmetry',
        'worst fractal dimension']
```

```
model = ensemble.RandomForestClassifier(random_state=1, n_estimators=10)
model.fit(bcancer_ref[bcancer_data.feature_names.tolist()], bcancer_ref.target)
```

```
RandomForestClassifier(n_estimators=10, random_state=1)
```

```
bcancer_ref['prediction'] = model.predict_proba(bcancer_ref[bcancer_data.feature_names.tolist()])[:, 1]
bcancer_cur['prediction'] = model.predict_proba(bcancer_cur[bcancer_data.feature_names.tolist()])[:, 1]
```

```
bcancer_ref.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	d
111	12.63	20.76	82.15	480.4	0.09933	0.12090	0.106500	0.060210	0.1735	0.07070	...	89.00	527.4	0.12870	0.22500	0.221600	0.11050	0.2226	
8	13.00	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2350	0.07389	...	106.20	739.3	0.17030	0.54010	0.539000	0.20600	0.4378	
333	11.25	14.78	71.38	390.0	0.08306	0.04458	0.000974	0.002941	0.1773	0.06081	...	82.08	492.7	0.11660	0.09794	0.005518	0.01667	0.2815	
397	12.80	17.46	83.05	508.3	0.08044	0.08895	0.073900	0.040830	0.1574	0.05750	...	90.72	591.0	0.09534	0.18120	0.190100	0.08296	0.1988	
90	14.62	24.02	94.57	662.7	0.08974	0.08606	0.031020	0.029570	0.1685	0.05866	...	102.90	803.7	0.11150	0.17660	0.091890	0.06946	0.2522	

5 rows × 32 columns

```
bcancer_cur.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	d
397	12.800	17.46	83.05	508.3	0.08044	0.08895	0.07390	0.04083	0.1574	0.05750	...	90.72	591.0	0.09534	0.18120	0.19010	0.08296	0.1988	
158	12.060	12.74	76.84	448.6	0.09311	0.05241	0.01972	0.01963	0.1590	0.05907	...	84.08	532.8	0.12750	0.12320	0.08636	0.07025	0.2514	
149	13.740	17.91	88.12	585.0	0.07944	0.06376	0.02881	0.01329	0.1473	0.05580	...	97.19	725.9	0.09711	0.18240	0.15640	0.06019	0.2350	
471	12.040	28.14	76.85	449.9	0.08752	0.06000	0.02367	0.02377	0.1854	0.05698	...	87.24	567.6	0.10410	0.09726	0.05524	0.05547	0.2404	
416	9.405	21.70	59.60	271.2	0.10440	0.06159	0.02047	0.01257	0.2025	0.06601	...	68.73	359.4	0.15260	0.11930	0.06141	0.03770	0.2872	

5 rows × 32 columns

```
bcancer_label_ref['prediction'] = model.predict(bcancer_label_ref[bcancer_data.feature_names.tolist()])
bcancer_label_cur['prediction'] = model.predict(bcancer_label_cur[bcancer_data.feature_names.tolist()])
```

```
bcancer_label_ref.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	d
111	12.63	20.76	82.15	480.4	0.09933	0.12090	0.106500	0.060210	0.1735	0.07070	...	89.00	527.4	0.12870	0.22500	0.221600	0.11050	0.2226	
8	13.00	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2350	0.07389	...	106.20	739.3	0.17030	0.54010	0.539000	0.20600	0.4378	
333	11.25	14.78	71.38	390.0	0.08306	0.04458	0.000974	0.002941	0.1773	0.06081	...	82.08	492.7	0.11660	0.09794	0.005518	0.01667	0.2815	
397	12.80	17.46	83.05	508.3	0.08044	0.08895	0.073900	0.040830	0.1574	0.05750	...	90.72	591.0	0.09534	0.18120	0.190100	0.08296	0.1988	
90	14.62	24.02	94.57	662.7	0.08974	0.08606	0.031020	0.029570	0.1685	0.05866	...	102.90	803.7	0.11150	0.17660	0.091890	0.06946	0.2522	

5 rows × 32 columns

```
bcancer_label_cur.head()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	d
397	12.800	17.46	83.05	508.3	0.08044	0.08895	0.07390	0.04083	0.1574	0.05750	...	90.72	591.0	0.09534	0.18120	0.19010	0.08296	0.1988	
158	12.060	12.74	76.84	448.6	0.09311	0.05241	0.01972	0.01963	0.1590	0.05907	...	84.08	532.8	0.12750	0.12320	0.08636	0.07025	0.2514	
149	13.740	17.91	88.12	585.0	0.07944	0.06376	0.02881	0.01329	0.1473	0.05580	...	97.19	725.9	0.09711	0.18240	0.15640	0.06019	0.2350	
471	12.040	28.14	76.85	449.9	0.08752	0.06000	0.02367	0.02377	0.1854	0.05698	...	87.24	567.6	0.10410	0.09726	0.05524	0.05547	0.2404	
416	9.405	21.70	59.60	271.2	0.10440	0.06159	0.02047	0.01257	0.2025	0.06601	...	68.73	359.4	0.15260	0.11930	0.06141	0.03770	0.2872	

5 rows × 32 columns

4. Dataset for Multiclass Classifcation

```
iris_data = datasets.load_iris(as_frame=True)
iris = iris_data.frame
```

```
iris.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Next steps: Generate code with `iris` View recommended plots New interactive sheet

```
iris.target.value_counts()
```

target	count
0	50
1	50
2	50

dtype: int64

```
iris_ref = iris.sample(n=75, replace=False)
iris_cur = iris.sample(n=75, replace=False)
```

```
model = ensemble.RandomForestClassifier(random_state=1, n_estimators=3)
model.fit(iris_ref[iris_data.feature_names], iris_ref.target)
```

```
RandomForestClassifier
RandomForestClassifier(n_estimators=3, random_state=1)
```

```
iris_ref['prediction'] = model.predict(iris_ref[iris_data.feature_names])
iris_cur['prediction'] = model.predict(iris_cur[iris_data.feature_names])
```

```
iris_ref.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	prediction
76	6.8	2.8	4.8	1.4	1	1
14	5.8	4.0	1.2	0.2	0	0
87	6.3	2.3	4.4	1.3	1	1
44	5.1	3.8	1.9	0.4	0	0
55	5.7	2.8	4.5	1.3	1	1

Next steps: Generate code with iris_ref | View recommended plots | New interactive sheet

```
iris_cur.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	prediction
17	5.1	3.5	1.4	0.3	0	0
44	5.1	3.8	1.9	0.4	0	0
8	4.4	2.9	1.4	0.2	0	0
63	6.1	2.9	4.7	1.4	1	1
30	4.8	3.1	1.6	0.2	0	0

Next steps: Generate code with iris_cur | View recommended plots | New interactive sheet

5. Leveraging Test presets and List of Tests

A. Data stability

```
data_stability = TestSuite(tests=[
    DataStabilityTestPreset(),
])

data_stability.run(reference_data=adult_ref, current_data=adult_cur)
data_stability
```

39	29	0	10	0
Tests	Success	Warning	Fail	Error

All tests

Number of Rows

The number of rows is 34687. The test threshold is eq=1.42e+04 ± 1.42e+03.

Number of Columns Details

The number of columns is 15. The test threshold is eq=15.

Column Types Details

The number of columns with a type mismatch is 1 out of 15.

The Share of Missing Values in a Column

The share of missing values in the column **capital-gain** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **hours-per-week** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **age** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **fnlwgt** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **education-num** is 0.0577. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **capital-loss** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **relationship** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **race** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **native-country** is 0.0162. The test threshold is lte=0.0208 ± 0.00208.

The Share of Missing Values in a Column

The share of missing values in the column **workclass** is 0.0549. The test threshold is lte=0.0633 ± 0.00633.

The Share of Missing Values in a Column

The share of missing values in the column **occupation** is 0.055. The test threshold is lte=0.0637 ± 0.00637.

The Share of Missing Values in a Column

The share of missing values in the column **class** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **education** is 0.0577. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **marital-status** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **sex** is 0. The test threshold is lte=0 ± 1e-12.

Share of Out-of-Range Values

Details

The share of values out of range in the column **capital-gain** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-Range Values

Details

The share of values out of range in the column **hours-per-week** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-Range Values

Details

The share of values out of range in the column **age** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-Range Values

Details

The share of values out of range in the column **fnlwgt** is 0.000115 (4 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-Range Values

Details

The share of values out of range in the column **education-num** is 0 (0 out of 32687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-Range Values

Details

The share of values out of range in the column **capital-loss** is 8.65e-05 (3 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values

Details

The share of values out of list in the column **relationship** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values

Details

The share of values out of list in the column **race** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values

Details

The share of values out of list in the column **native-country** is 0.0163 (564 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values — Details

The share of values out of list in the column **workclass** is 0.0549 (1903 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values — Details

The share of values out of list in the column **occupation** is 0.055 (1907 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values — Details

The share of values out of list in the column **class** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values — Details

The share of values out of list in the column **education** is 1 (34687 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values — Details

The share of values out of list in the column **marital-status** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values — Details

The share of values out of list in the column **sex** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Mean Value Stability — Details

The mean value of the column **capital-gain** is 844. The expected range is from -1.82e+04 to 2.15e+04

Mean Value Stability — Details

The mean value of the column **hours-per-week** is 40.5. The expected range is from 13.4 to 67

Mean Value Stability — Details

The mean value of the column **age** is 38. The expected range is from 11.3 to 69.3

Mean Value Stability — Details

The mean value of the column **fnlwgt** is 1.89e+05. The expected range is from 2.1e+04 to 4.04e+05

The mean value of the column **fnlwgt** is 1.89e+05. The expected range is from -2.1e+04 to 4.04e+05

Mean Value Stability **Details**

The mean value of the column **education-num** is 10.2. The expected range is from 1.56 to 17.8

Mean Value Stability **Details**

The mean value of the column **capital-loss** is 83.3. The expected range is from -763 to 958

```
#test preset in a JSON format
data_stability.json()
```

'{"version": "0.4.40", "tests": [{"name": "Number of Rows", "description": "The number of rows is 34687. The test threshold is eq=1.42e+04 \\u00b1 1.42e+03.", "status": "FAIL", "grou
p": "data_integrity", "parameters": {"condition": {"eq": {"value": 14155, "relative": 0.1, "absolute": 1e-12}}, "value": 34687.0}}, {"name": "Number of Columns", "description": "The
number of columns is 15. The test threshold is eq=15.", "status": "SUCCESS", "group": "data_integrity", "parameters": {"condition": {"eq": 15}, "value": 15.0}}, {"name": "Column Type
s", "description": "The number of columns with a type mismatch is 1 out of 15.", "status": "FAIL", "group": "data_integrity", "parameters": {"columns": [{"actual_type": "int64", "col
umn_name": "age", "expected_type": "int64"}, {"actual_type": "CategoricalDtypeType", "column_name": "workclass", "expected_type": "CategoricalDtypeType"}, {"actual_type": "int64", "c
olumn_name": "fnlwgt", "expected_type": "int64"}, {"actual_type": "CategoricalDtypeType", "...'

```
#test preset as a python object
data_stability.as_dict()
```

```
          'status': 'SUCCESS',
          'group': 'data_quality',
          'parameters': {'column_name': 'hours-per-week',
           'current_mean': 40.51,
           'n_sigmas': 2,
           'reference_mean': 40.21,
           'reference_std': 13.42}},
         {'name': 'Mean Value Stability',
          'description': 'The mean value of the column **age** is 38. The expected range is from 11.3 to 69.3',
```

B. Data Quality

```
data_quality = TestSuite(tests=[
    DataQualityTestPreset(),
])

data_quality.run(reference_data=adult_ref, current_data=adult_cur)
data_quality
```

33	25	0	8	0
Tests	Success	Warning	Fail	Error

All tests

The Share of Missing Values in a Column

The share of missing values in the column **capital-gain** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **hours-per-week** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **age** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **fnlwgt** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **education-num** is 0.0577. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **capital-loss** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **relationship** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **race** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **native-country** is 0.0162. The test threshold is lte=0.0208 ± 0.00208.

The Share of Missing Values in a Column

The share of missing values in the column **workclass** is 0.0549. The test threshold is lte=0.0633 ± 0.00633.

The Share of Missing Values in a Column

The share of missing values in the column **occupation** is 0.055. The test threshold is lte=0.0637 ± 0.00637.

The Share of Missing Values in a Column

The share of missing values in the column **class** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **education** is 0.0577. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **marital-status** is 0. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **sex** is 0. The test threshold is lte=0 ± 1e-12.

Share of the Most Common Value

Details

The most common value in the column **capital-gain** is 0. Its share is 0.923. The test threshold is eq=0.903 ± 0.0903.

Share of the Most Common Value

Details

The most common value in the column **hours-per-week** is 40. Its share is 0.479. The test threshold is eq=0.438 ± 0.0438.

Share of the Most Common Value — Details

The most common value in the column **age** is 23. Its share is 0.0312. The test threshold is eq=0.0407 ± 0.00407.

Share of the Most Common Value — Details

The most common value in the column **fnlwgt** is 203488. Its share is 0.0006. The test threshold is eq=0.0005 ± 5e-05.

Share of the Most Common Value — Details

The most common value in the column **education-num** is 9.0. Its share is 0.429. The test threshold is eq=0.188 ± 0.0188.

Share of the Most Common Value — Details

The most common value in the column **capital-loss** is 0. Its share is 0.955. The test threshold is eq=0.949 ± 0.0949.

Share of the Most Common Value — Details

The most common value in the column **relationship** is Husband. Its share is 0.394. The test threshold is eq=0.426 ± 0.0426.

Share of the Most Common Value — Details

The most common value in the column **race** is White. Its share is 0.857. The test threshold is eq=0.851 ± 0.0851.

Share of the Most Common Value — Details

The most common value in the column **native-country** is United-States. Its share is 0.918. The test threshold is eq=0.847 ± 0.0847.

Share of the Most Common Value — Details

The most common value in the column **workclass** is Private. Its share is 0.707. The test threshold is eq=0.663 ± 0.0663.

Share of the Most Common Value — Details

The most common value in the column **occupation** is Adm-clerical. Its share is 0.135. The test threshold is eq=0.209 ± 0.0209.

Share of the Most Common Value — Details

The most common value in the column **class** is <=50K. Its share is 0.773. The test threshold is eq=0.731 ± 0.0731.

Share of the Most Common Value

Details

The most common value in the column **education** is HS-grad. Its share is 0.429. The test threshold is eq=0.188 ± 0.0188.

Share of the Most Common Value

Details

The most common value in the column **marital-status** is Married-civ-spouse. Its share is 0.449. The test threshold is eq=0.482 ± 0.0482.

Share of the Most Common Value

Details

The most common value in the column **sex** is Male. Its share is 0.661. The test threshold is eq=0.686 ± 0.0686.

Number of Constant Columns

Details

The number of constant columns is 0. The test threshold is lte=0.

Number of Duplicate Columns

The number of duplicate columns is 0. The test threshold is lte=0.

Number of Duplicate Rows

The number of duplicate rows is 38. The test threshold is eq=27 ± 2.7.

C. Data Drift

```
data_drift = TestSuite(tests=[
    DataDriftTestPreset(stattest='psi'),
])

data_drift.run(reference_data=adult_ref, current_data=adult_cur)
data_drift
```

16	13	0	3	0
Tests	Success	Warning	Fail	Error

All tests

Share of Drifted Columns Details

The drift is detected for 20% features (3 out of 15). The test threshold is lt=0.3

Drift per Column Details

The drift score for the feature **relationship** is 0.008. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **race** is 0.003. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **native-country** is 0.1. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **workclass** is 0.013. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **occupation** is 0.192. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **class** is 0.009. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **education** is 15. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **marital-status** is 0.01. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **sex** is 0.003. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **capital-gain** is 0.029. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **hours-per-week** is 0.016. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **age** is 0.055. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **fnlwgt** is 0.004. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **education-num** is 15. The drift detection method is PSI. The drift detection threshold is 0.1.

Drift per Column Details

The drift score for the feature **capital-loss** is 0.006. The drift detection method is PSI. The drift detection threshold is 0.1.

D. test suite to monitor a model's performance without actual target values (labels)

```
no_target_performance = TestSuite(tests=[
    NoTargetPerformanceTestPreset(columns=['education-num', 'hours-per-week'],  num_stattest='ks', cat_stattest='psi'),
])

no_target_performance.run(reference_data=adult_ref, current_data=adult_cur)
no_target_performance
```

8	5	0	3	0
Tests	Success	Warning	Fail	Error

All tests

Share of Drifted Columns Details

The drift is detected for 46.7% features (7 out of 15). The test threshold is lt=0.3

Column Types Details

The number of columns with a type mismatch is 1 out of 15.

The Share of Missing Values in a Column

The share of missing values in the column **education-num** is 0.0577. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **hours-per-week** is 0. The test threshold is lte=0 ± 1e-12.

Share of Out-of-Range Values Details

The share of values out of range in the column **education-num** is 0 (0 out of 32687). The test threshold is eq=0 ± 1e-12.

Share of Out-of-Range Values Details

The share of values out of range in the column **hours-per-week** is 0 (0 out of 34687). The test threshold is eq=0 ± 1e-12.

Mean Value Stability Details

The mean value of the column **education-num** is 10.2. The expected range is from 1.56 to 17.8

Mean Value Stability Details

The mean value of the column **hours-per-week** is 40.5. The expected range is from 13.4 to 67

E. Regression model performance

```
regression_performance = TestSuite(tests=[
    RegressionTestPreset()
])

regression_performance.run(reference_data=housing_ref, current_data=housing_cur)
regression_performance
```

4	4	0	0	0
Tests	Success	Warning	Fail	Error

All tests

Mean Error (ME) Details

The ME is -0.0365. The test threshold is eq=0 ± 0.295.

Mean Absolute Error (MAE) Details

The MAE is 2.34. The test threshold is eq=2.36 ± 0.236

Root Mean Square Error (RMSE) Details

The RMSE is 2.95. The test threshold is eq=2.96 ± 0.296.

Mean Absolute Percentage Error (MAPE) Details

The MAPE is 1.57e+02. The test threshold is eq=157 ± 15.7.

F. Multiclass Classification Model Performance

```
classification_performance = TestSuite(tests=[
   MulticlassClassificationTestPreset(stattest='psi')
])

classification_performance.run(reference_data=iris_ref, current_data=iris_cur)
classification_performance
```

10	10	0	0	0
Tests	Success	Warning	Fail	Error

All tests

Accuracy Score Details

The Accuracy Score is 0.947. The test threshold is eq=0.973 ± 0.195

F1 Score Details

The F1 Score is 0.951. The test threshold is eq=0.971 ± 0.194

Precision Score by Class Details

The precision score of the label **0** is 1. The test threshold is eq=1 ± 0.2

Precision Score by Class Details

The precision score of the label **1** is 0.893. The test threshold is eq=0.964 ± 0.193

Precision Score by Class Details

The precision score of the label **2** is 0.963. The test threshold is eq=0.95 ± 0.19

Recall Score by Class Details

The recall score of the label **0** is 1. The test threshold is eq=1 ± 0.2

Recall Score by Class Details

The recall score of the label **1** is 0.962. The test threshold is eq=0.964 ± 0.193

Recall Score by Class Details

The recall score of the label **2** is 0.897. The test threshold is eq=0.95 ± 0.19

Number of Rows

The number of rows is 75. The test threshold is eq=75 ± 7.5.

Drift per Column

Details

The drift score for the feature **target** is 0.075. The drift detection method is PSI. The drift detection threshold is 0.1.

G. Binary Classification Model Peformance

```
label_binary_classification_performance = TestSuite(tests=[
    BinaryClassificationTestPreset(),
])

label_binary_classification_performance.run(reference_data=bcancer_label_ref, current_data=bcancer_label_cur)
label_binary_classification_performance
```

5	5	0	0	0
Tests	Success	Warning	Fail	Error

All tests

Drift per Column Details

The drift score for the feature **target** is 0.513. The drift detection method is Z-test p_value. The drift detection threshold is 0.05.

Precision Score Details

The Precision Score is 0.971. The test threshold is eq=1 ± 0.2

Recall Score Details

The Recall Score is 0.978. The test threshold is eq=1 ± 0.2

F1 Score Details

The F1 Score is 0.974. The test threshold is eq=1 ± 0.2

Accuracy Score Details

The Accuracy Score is 0.965. The test threshold is eq=1 ± 0.2

```
prob_binary_classification_performance = TestSuite(tests=[
    BinaryClassificationTestPreset(stattest='psi', probas_threshold=0.89),
])

prob_binary_classification_performance.run(reference_data=bcancer_ref, current_data=bcancer_cur)
prob_binary_classification_performance
```

6	6	0	0	0
Tests	Success	Warning	Fail	Error

All tests

Drift per Column Details

The drift score for the feature **target** is 0.004. The drift detection method is PSI. The drift detection threshold is 0.1.

ROC AUC Score Details

The ROC AUC Score is 0.98. The test threshold is eq=1 ± 0.2

Precision Score Details

The Precision Score is 0.976. The test threshold is eq=1 ± 0.2

Recall Score Details

The Recall Score is 0.919. The test threshold is eq=0.948 ± 0.19

Accuracy Score Details

The Accuracy Score is 0.93. The test threshold is eq=0.967 ± 0.193

F1 Score Details

The F1 Score is 0.947. The test threshold is eq=0.974 ± 0.195

```
binary_topK_classification_performance = TestSuite(tests=[
    BinaryClassificationTopKTestPreset(k=10, stattest='psi'),
])

binary_topK_classification_performance.run(reference_data=bcancer_ref, current_data=bcancer_cur)
```

```
binary_topK_classification_performance
```

7	6	0	1	0
Tests	Success	Warning	Fail	Error

All tests

Accuracy Score Details

The Accuracy Score is 0.905. The test threshold is eq=0.89 ± 0.178

Precision Score Details

The Precision Score is 0.983. The test threshold is eq=1 ± 0.2

Recall Score Details

The Recall Score is 0.874. The test threshold is eq=0.83 ± 0.166

F1 Score Details

The F1 Score is 0.925. The test threshold is eq=0.907 ± 0.181

Drift per Column Details

The drift score for the feature **target** is 0.004. The drift detection method is PSI. The drift detection threshold is 0.1.

ROC AUC Score Details

The ROC AUC Score is 0.98. The test threshold is eq=1 ± 0.2

Logarithmic Loss Details

The Logarithmic Loss is 0.429. The test threshold is eq=0.0422 ± 0.00845

H. List of all Data Integrity Test

```
#dataset-level tests

data_integrity_dataset_tests = TestSuite(tests=[
    TestNumberOfColumns(),
    TestNumberOfRows(),
    TestNumberOfMissingValues(),
    TestShareOfMissingValues(),
    TestNumberOfColumnsWithMissingValues(),
    TestNumberOfRowsWithMissingValues(),
    TestShareOfColumnsWithMissingValues(),
    TestShareOfRowsWithMissingValues(),
    TestNumberOfDifferentMissingValues(),
    TestNumberOfConstantColumns(),
    TestNumberOfEmptyRows(),
    TestNumberOfEmptyColumns(),
    TestNumberOfDuplicatedRows(),
    TestNumberOfDuplicatedColumns(),
    TestColumnsType(),

])

data_integrity_dataset_tests.run(reference_data=adult_ref, current_data=adult_cur)
data_integrity_dataset_tests
```

15	6	0	9	0
Tests	Success	Warning	Fail	Error

All tests

Number of Columns Details

The number of columns is 15. The test threshold is eq=15.

Number of Rows

The number of rows is 34687. The test threshold is eq=1.42e+04 ± 1.42e+03.

The Number of Missing Values Details

The number of missing values is 8373. The test threshold is lte=5.13e+03 ± 513.

Share of Missing Values Details

The share of missing values is 0.0161. The test threshold is lte=0.00985 ± 0.000985.

The Number of Columns With Missing Values Details

The number of columns with missing values is 5. The test threshold is lte=3.

The Number Of Rows With Missing Values

The number of rows with missing values is 4281. The test threshold is lte=2.9e+03 ± 290.

The Share of Columns With Missing Values Details

The share of columns with missing values is 0.333. The test threshold is lte=0.2.

The Share of Rows With Missing Values

The share of rows with missing values is 0.123. The test threshold is lte=0.0837 ± 0.00837.

Different Types of Missing Values

Details

The number of differently encoded types of missing values is 1. The test threshold is eq=1.

Number of Constant Columns

Details

The number of constant columns is 0. The test threshold is lte=0.

Number of Empty Rows

Number of Empty Rows is 0. The test threshold is eq=0 ± 1e-12.

Number of Empty Columns

Details

Number of Empty Columns is 0. The test threshold is lte=0.

Number of Duplicate Rows

The number of duplicate rows is 38. The test threshold is eq=27 ± 2.7.

Number of Duplicate Columns

The number of duplicate columns is 0. The test threshold is lte=0.

Column Types

Details

The number of columns with a type mismatch is 1 out of 15.

```
#column-level tests

data_integrity_column_tests = TestSuite(tests=[
    TestColumnNumberOfMissingValues(column_name='education'),
    TestColumnShareOfMissingValues(column_name='education'),
    TestColumnNumberOfDifferentMissingValues(column_name='education'),
    TestColumnAllConstantValues(column_name='education'),
    TestColumnAllUniqueValues(column_name='education'),
    TestColumnRegExp(column_name='education',reg_exp='^[0..9]'),
    TestCategoryShare(column_name='education', category='Some-college', lt=0.5),
    TestCategoryShare(column_name='age', category=27., lt=0.5)
])

data_integrity_column_tests.run(reference_data=adult_ref, current_data=adult_cur)
data_integrity_column_tests
```

8	4	0	4	0
Tests	Success	Warning	Fail	Error

All tests

The Number of Missing Values in a Column

The number of missing values in the column **education** is 2000. The test threshold is lte=0 ± 1e-12.

The Share of Missing Values in a Column

The share of missing values in the column **education** is 0.0577. The test threshold is lte=0 ± 1e-12.

Different Types of Missing Values in a Column Details

The number of differently encoded types of missing values in the column **education** is 1. The test threshold is lte=0.

All Constant Values in a Column Details

The number of the unique values in the column **education** is 3 out of 34687

All Unique Values in a Column Details

The number of the unique values in the column **education** is 3 out of 34687

RegExp Match Details

The number of the mismatched values in the column **education** is 32687. The test threshold is eq=3.28e+04 ± 3.28e+03.

Share of category Details

The share of category 'Some-college' in the column **education** is 0.296 (10260 out of 34687). The test threshold is lt=0.5.

Share of category Details

The share of category '27.0' in the column **age** is 0.0271 (941 out of 34687). The test threshold is lt=0.5.

I. List of all Data Quality Tests

```
#dataset-level tests

data_quality_dataset_tests = TestSuite(tests=[
    TestConflictTarget(),
    TestConflictPrediction(),
    TestTargetPredictionCorrelation(),
    TestHighlyCorrelatedColumns(),
    TestTargetFeaturesCorrelations(),
    TestPredictionFeaturesCorrelations(),
    TestCorrelationChanges(),
])

data_quality_dataset_tests.run(reference_data=iris_ref, current_data=iris_cur)
data_quality_dataset_tests
```

7	7	0	0	0
Tests	Success	Warning	Fail	Error

All tests

Test number of conflicts in target

Target is stable

Test number of conflicts in prediction

Prediction is stable

Correlation between Target and Prediction

The correlation between the target and prediction is 0.931. The test threshold is eq=0.958 ± 0.25.

Highly Correlated Columns Details

The maximum correlation is 0.952. The test threshold is eq=0.972 ± 0.0972.

Correlation between Target and Features Details

The maximum correlation is 0.931. The test threshold is eq=0.958 ± 0.0958.

Correlation between Prediction and Features Details

The maximum correlation is 0.931. The test threshold is eq=0.958 ± 0.0958.

Change in Correlation Details

The number of correlation violations is 0. The test threshold is eq=0.

```
#column-level tests

data_quality_column_tests = TestSuite(tests=[
    TestColumnValueMin(column_name='education-num'),
    TestColumnValueMax(column_name='education-num'),
    TestColumnValueMean(column_name='education-num'),
    TestColumnValueMedian(column_name='education-num'),
    TestColumnValueStd(column_name='education-num'),
    TestNumberOfUniqueValues(column_name='education'),
    TestUniqueValuesShare(column_name='education'),
    TestMostCommonValueShare(column_name='education'),
    TestMeanInNSigmas(column_name='education-num'),
    TestValueRange(column_name='education-num'),
    TestNumberOfOutRangeValues(column_name='education-num'),
    TestShareOfOutRangeValues(column_name='education-num'),
    TestValueList(column_name='education'),
    TestNumberOfOutListValues(column_name='education'),
    TestShareOfOutListValues(column_name='education'),
    TestColumnQuantile(column_name='education-num', quantile=0.25),
    TestShareOfOutListValues(column_name='education-num'),
])

data_quality_column_tests.run(reference_data=adult_ref, current_data=adult_cur)
data_quality_column_tests
```

17	8	0	9	0
Tests	Success	Warning	Fail	Error

All tests

Min Value Details

The minimum value of the column **education-num** is 9.0 The test threshold is gte=1.

Max Value Details

The maximum value of the column **education-num** is 13.0. The test threshold is lte=16.

Mean Value Details

The mean value of the column **education-num** is 10.2. The test threshold is eq=9.68 ± 0.968.

Median Value Details

The median value of the column **education-num** is 10. The test threshold is eq=11 ± 1.1.

Standard Deviation (SD) Details

The standard deviation of the column **education-num** is 1.57. The test threshold is eq=4.06 ± 0.406.

Number of Unique Values Details

The number of the unique values in the column **education** is 3. The test threshold is eq=13 ± 1.3.

Share of Unique Values Details

The share of the unique values in the column **education** is 0.0001. The test threshold is eq=0.0009 ± 9e-05.

Share of the Most Common Value Details

The most common value in the column **education** is HS-grad. Its share is 0.429. The test threshold is eq=0.188 ± 0.0188.

Mean Value Stability

Details

The mean value of the column **education-num** is 10.2. The expected range is from 1.56 to 17.8

Value Range

Details

All values in the column **education-num** are within range

Number of Out-of-Range Values

Details

The number of values out of range in the column **education-num** is 0. The test threshold is eq=0 ± 1e-12.

Share of Out-of-Range Values

Details

The share of values out of range in the column **education-num** is 0 (0 out of 32687). The test threshold is eq=0 ± 1e-12.

Out-of-List Values

Details

The column **education** has values out of list.

Number Out-of-List Values

Details

The number of values out of list in the column **education** is 34687. The test threshold is eq=0 ± 1e-12.

Share of Out-of-List Values

Details

The share of values out of list in the column **education** is 1 (34687 out of 34687). The test threshold is eq=0 ± 1e-12.

Quantile Value

Details

The 0.25 quantile value of the column **education-num** is 9. The test threshold is eq=6 ± 0.6.

Share of Out-of-List Values

Details

The share of values out of list in the column **education-num** is 1 (34687 out of 34687). The test threshold is eq=0 ± 1e-12.

J. List of all Data Drift Tests

```
#dataset-level tests

data_drift_dataset_tests = TestSuite(tests=[
    TestNumberOfDriftedColumns(),
    TestShareOfDriftedColumns(),
])

data_drift_dataset_tests.run(reference_data=adult_ref, current_data=adult_cur)
data_drift_dataset_tests
```

2	0	0	2	0
Tests	Success	Warning	Fail	Error

All tests

Number of Drifted Features — Details

The drift is detected for 5 out of 15 features. The test threshold is lt=5.

Share of Drifted Columns — Details

The drift is detected for 33.3% features (5 out of 15). The test threshold is lt=0.3

```
#column-level tests

data_drift_column_tests = TestSuite(tests=[
    TestColumnDrift(column_name='education-num', stattest='psi', stattest_threshold=0.3)
])

data_drift_column_tests.run(reference_data=adult_ref, current_data=adult_cur)
data_drift_column_tests
```

1	0	0	1	0
Tests	Success	Warning	Fail	Error

All tests

Drift per Column Details

The drift score for the feature **education-num** is 15. The drift detection method is PSI. The drift detection threshold is 0.3.

K. List of all Regression Performance Test

```
#dataset-level tests
regression_performance_dataset_tests = TestSuite(tests=[
    TestValueMAE(),
    TestValueRMSE(),
    TestValueMeanError(),
    TestValueMAPE(),
    TestValueAbsMaxError(),
    TestValueR2Score()
])

regression_performance_dataset_tests.run(reference_data=housing_ref, current_data=housing_cur)
regression_performance_dataset_tests
```

6	6	0	0	0
Tests	Success	Warning	Fail	Error

All tests

Mean Absolute Error (MAE) Details

The MAE is 2.34. The test threshold is eq=2.36 ± 0.236

Root Mean Square Error (RMSE) Details

The RMSE is 2.95. The test threshold is eq=2.96 ± 0.296.

Mean Error (ME) Details

The ME is -0.0365. The test threshold is eq=0 ± 0.295.

Mean Absolute Percentage Error (MAPE) Details

The MAPE is 1.57e+02. The test threshold is eq=157 ± 15.7.

Max Absolute Error Details

The Max Absolute Error is 11.5. The test threshold is lte=10.6 ± 1.06.

R2 Score Details

The R2 score is -5.54. The test threshold is eq=-5.62 ± 0.562.

L. List of all Multiclass classification performance test

```
#dataset-level tests

classification_performance_dataset_tests = TestSuite(tests=[
    TestAccuracyScore(),
    TestPrecisionScore(),
    TestRecallScore(),
    TestF1Score(),
    TestPrecisionByClass(label=0),
    TestPrecisionByClass(label=1),
    TestPrecisionByClass(label=2),
    TestRecallByClass(label=0),
    TestRecallByClass(label=1),
    TestRecallByClass(label=2),
    TestF1ByClass(label=0),
    TestF1ByClass(label=1),
    TestF1ByClass(label=2),
])

classification_performance_dataset_tests.run(reference_data=iris_ref, current_data=iris_cur)
classification_performance_dataset_tests
```

13	13	0	0	0
Tests	Success	Warning	Fail	Error

All tests

Accuracy Score Details

The Accuracy Score is 0.947. The test threshold is eq=0.973 ± 0.195

Precision Score Details

The Precision Score is 0.952. The test threshold is eq=0.971 ± 0.194

Recall Score Details

The Recall Score is 0.953. The test threshold is eq=0.971 ± 0.194

F1 Score Details

The F1 Score is 0.951. The test threshold is eq=0.971 ± 0.194

Precision Score by Class Details

The precision score of the label **0** is 1. The test threshold is eq=1 ± 0.2

Precision Score by Class Details

The precision score of the label **1** is 0.893. The test threshold is eq=0.964 ± 0.193

Precision Score by Class Details

The precision score of the label **2** is 0.963. The test threshold is eq=0.95 ± 0.19

Recall Score by Class Details

The recall score of the label **0** is 1. The test threshold is eq=1 ± 0.2

Recall Score by Class **Details**

The recall score of the label **1** is 0.962. The test threshold is eq=0.964 ± 0.193

Recall Score by Class **Details**

The recall score of the label **2** is 0.897. The test threshold is eq=0.95 ± 0.19

F1 Score by Class **Details**

The F1 score of the label **0** is 1. The test threshold is eq=1 ± 0.2

F1 Score by Class **Details**

The F1 score of the label **1** is 0.926. The test threshold is eq=0.964 ± 0.193

F1 Score by Class **Details**

The F1 score of the label **2** is 0.929. The test threshold is eq=0.95 ± 0.19

M. List of all Binary Classification performance test

```
#dataset-level tests

prob_classification_performance_dataset_tests = TestSuite(tests=[
    TestAccuracyScore(),
    TestPrecisionScore(),
    TestRecallScore(),
    TestF1Score(),
    TestRocAuc(),
    TestLogLoss(),
    TestPrecisionByClass(label=0),
    TestPrecisionByClass(label=1),
    TestRecallByClass(label=0),
    TestRecallByClass(label=1),
    TestF1ByClass(label=0),
    TestF1ByClass(label=1),

])

prob_classification_performance_dataset_tests.run(reference_data=bcancer_ref, current_data=bcancer_cur)
prob_classification_performance_dataset_tests
```

12	11	0	1	0
Tests	Success	Warning	Fail	Error

All tests

Accuracy Score Details

The Accuracy Score is 0.97. The test threshold is eq=0.99 ± 0.198

Precision Score Details

The Precision Score is 0.957. The test threshold is eq=0.985 ± 0.197

Recall Score Details

The Recall Score is 1. The test threshold is eq=1 ± 0.2

F1 Score Details

The F1 Score is 0.978. The test threshold is eq=0.992 ± 0.198

ROC AUC Score Details

The ROC AUC Score is 0.98. The test threshold is eq=1 ± 0.2

Logarithmic Loss Details

The Logarithmic Loss is 0.429. The test threshold is eq=0.0422 ± 0.00845

Precision Score by Class Details

The precision score of the label **0** is 1. The test threshold is eq=1 ± 0.2

Precision Score by Class Details

The precision score of the label **1** is 0.957. The test threshold is eq=0.985 ± 0.197

Recall Score by Class

Details

The recall score of the label **0** is 0.908. The test threshold is eq=0.972 ± 0.194

Recall Score by Class

Details

The recall score of the label **1** is 1. The test threshold is eq=1 ± 0.2

F1 Score by Class

Details

The F1 score of the label **0** is 0.952. The test threshold is eq=0.986 ± 0.197

F1 Score by Class

Details

The F1 score of the label **1** is 0.978. The test threshold is eq=0.992 ± 0.198

6. Metric Preset and List of Metrics

A. Create the Data Drift Metric Report

```
data_drift_report = Report(metrics=[
    DataDriftPreset(num_stattest='ks', cat_stattest='psi', num_stattest_threshold=0.2, cat_stattest_threshold=0.2),
])

data_drift_report.run(reference_data=adult_ref, current_data=adult_cur)
data_drift_report
```

Dataset Drift

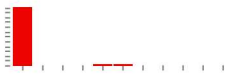
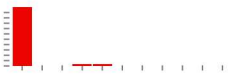
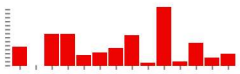
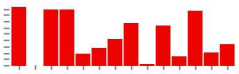
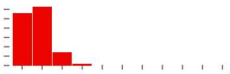
Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

15	6	0.4
Columns	Drifted Columns	Share of Drifted Columns

Data Drift Summary

Drift is detected for 40.0% of columns (6 out of 15).

Search

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
education	cat			Detected	PSI	15.016729
capital-loss	num			Not Detected	K-S p_value	0.346362
occupation	cat			Not Detected	PSI	0.191829
native-country	cat			Not Detected	PSI	0.099832
fnlwgt	num			Detected	K-S p_value	0.027831
workclass	cat			Not Detected	PSI	0.013186

marital-status	cat		Not Detected	PSI	0.010267
class	cat		Not Detected	PSI	0.009418
relationship	cat		Not Detected	PSI	0.007765
race	cat		Not Detected	PSI	0.003051

Rows per page: 10 rows 1-10 of 15

```
data_drift_report.save_html('data_drift_report.html')
```

```
data_drift_report.json()
```


```
'{"version": "0.4.40", "metrics": [{"metric": "DatasetDriftMetric", "result": {"drift_share": 0.5, "number_of_columns": 15, "number_of_drifted_columns": 6, "share_of_drifted_column
s": 0.4, "dataset_drift": false}}, {"metric": "DataDriftTable", "result": {"number_of_columns": 15, "number_of_drifted_columns": 6, "share_of_drifted_columns": 0.4, "dataset_drift":
false, "drift_by_columns": {"age": {"column_name": "age", "column_type": "num", "stattest_name": "K-S p_value", "stattest_threshold": 0.2, "drift_score": 2.704668436775831e-60, "drif
t_detected": true, "current": {"small_distribution": {"x": [17.0, 24.3, 31.6, 38.9, 46.2, 53.5, 60.8, 68.1, 75.4, 82.7, 90.0], "y": [0.02471021672878118, 0.025839691234843417, 0.0262
859521410848, 0.025211766596857754, 0.015942967066340047, 0.010173168977679455, 0.006152871609947434 4, 0.0018640278561586543, 0.000568686464590777, 0.0002369526935794904]}}, "referen
ce": {"small_distribution": {"x": [17.0, 24.3, 31.6, 38.9, 46.2, 53.5, 60.8, 68.1, 75.4, 82.7, "
```


```
data_drift_report.save_json('data_drift_report.json')
```

```
data_drift_report.as_dict()
```


```
{'metrics': [{'metric': 'DatasetDriftMetric',
   'result': {'drift_share': 0.5,
    'number_of_columns': 15,
    'number_of_drifted_columns': 6,
    'share_of_drifted_columns': 0.4,
    'dataset_drift': False}},
  {'metric': 'DataDriftTable',
   'result': {'number_of_columns': 15,
    'number_of_drifted_columns': 6,
    'share_of_drifted_columns': 0.4,
    'dataset_drift': False,
    'drift_by_columns': {'age': {'column_name': 'age',
      'column_type': 'num',
      'stattest_name': 'K-S p_value',
      'stattest_threshold': 0.2,
      'drift_score': 2.704668436775831e-60,
      'drift_detected': True,
      'current': {'small_distribution': {'x': [17.0,
```

```
                24.3,
                31.6,
                38.9,
                46.2,
                53.5,
                60.8,
                68.1,
                75.4,
                82.7,
                90.0],
               'y': [0.02471021672878118,
                0.025839691234843417,
                0.0262859521410848,
                0.025211766596857754,
                0.015942967066340047,
                0.010173168977679455,
                0.0061528716099474344,
                0.0018640278561586543,
                0.000568686464590777,
                0.00023695269357949041]}},
             'reference': {'small_distribution': {'x': [17.0,
                24.3,
                31.6,
                38.9,
                46.2,
                53.5,
                60.8,
                68.1,
                75.4,
                82.7,
                90.0],
               'y': [0.02104876054252575,
                0.020739077628796638,
                0.02384558435714183,
                0.026835959992838568,
                0.018658395552179158,
                0.012580868370245284,
                0.00869047676652328,
                0.0029516652714806184,
                0.0012871961019863...
```

```
data_drift_report = Report(metrics=[
    DataDriftPreset(num_stattest='ks',
                    cat_stattest='psi',
                    num_stattest_threshold=0.2,
                    cat_stattest_threshold=0.2),
],
  options={"render": {"raw_data": True}}
                          )

data_drift_report.run(reference_data=adult_ref, current_data=adult_cur)
data_drift_report
```

Dataset Drift

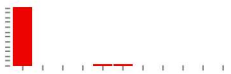
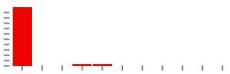
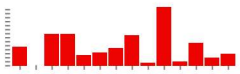
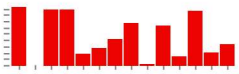
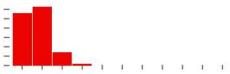
Dataset Drift is NOT detected. Dataset drift detection threshold is 0.5

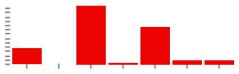
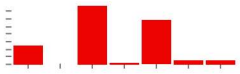
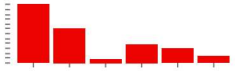
15	6	0.4
Columns	Drifted Columns	Share of Drifted Columns

Data Drift Summary

Drift is detected for 40.0% of columns (6 out of 15).

Search

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
education	cat			Detected	PSI	15.016729
capital-loss	num			Not Detected	K-S p_value	0.346362
occupation	cat			Not Detected	PSI	0.191829
native-country	cat			Not Detected	PSI	0.099832
fnlwgt	num			Detected	K-S p_value	0.027831
workclass	cat			Not Detected	PSI	0.013186

>	marital-status	cat			Not Detected	PSI	0.010267
>	class	cat			Not Detected	PSI	0.009418
>	relationship	cat			Not Detected	PSI	0.007765
>	race	cat			Not Detected	PSI	0.003051

Rows per page: 10 rows 1-10 of 15

B. Create the Data Quality Metric Report

```
data_quality_report = Report(metrics=[
    DataQualityPreset(),
])

data_quality_report.run(reference_data=adult_ref, current_data=adult_cur)
data_quality_report
```

Dataset Summary

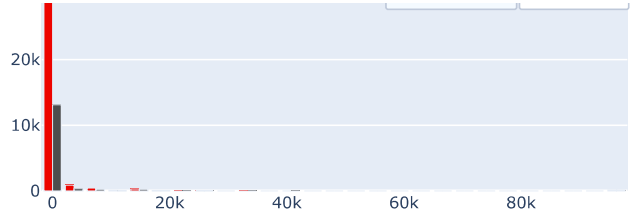
Metric	Current	Reference
id column	None	None
target column	None	None
prediction column	None	None
date column	None	None
number of columns	15	15
number of rows	34687	14155
missing values	8373	2092
categorical columns	9	9
numeric columns	6	6
text columns	0	0
datetime columns	0	0
empty columns	0	0
constant columns	0	0
almost constant features	1	0
duplicated columns	0	0
almost duplicated features	0	0

	current	reference
count	34687	14155
mean	844.27	1654.45
std	6159.97	9908.08
min	0.0	0.0
25%	0.0	0.0

capital-gain

num

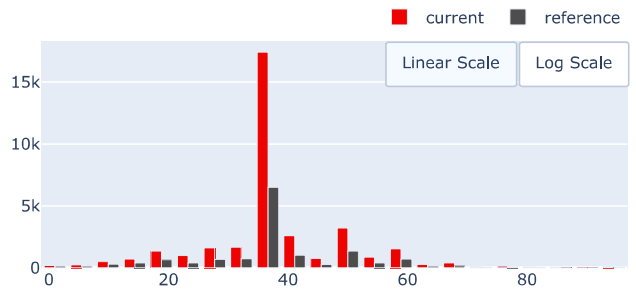
50%	0.0	0.0
75%	0.0	0.0
max	99999.0	99999.0
unique	117 (0.34%)	114 (0.81%)
most common	0.0 (92.33%)	0.0 (90.3%)
missing	0 (0.0%)	0 (0.0%)
infinite	0 (0.0%)	0 (0.0%)



hours-per-week

num

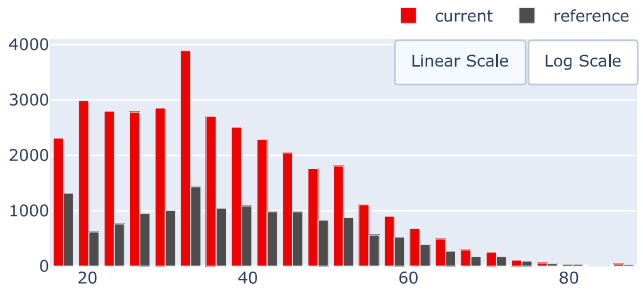
	current	reference
count	34687	14155
mean	40.51	40.21
std	11.94	13.42
min	1.0	1.0
25%	40.0	38.0
50%	40.0	40.0
75%	45.0	45.0
max	99.0	99.0
unique	95 (0.27%)	84 (0.59%)
most common	40.0 (47.88%)	40.0 (43.77%)
missing	0 (0.0%)	0 (0.0%)



infinite	0 (0.0%)	0 (0.0%)

age
num

	current	reference
count	34687	14155
mean	37.97	40.3
std	13.31	14.52
min	17.0	17.0
25%	27.0	29.0
50%	36.0	39.0
75%	47.0	50.0
max	90.0	90.0
unique	73 (0.21%)	73 (0.52%)
most common	23.0 (3.12%)	17.0 (4.07%)
missing	0 (0.0%)	0 (0.0%)
infinite	0 (0.0%)	0 (0.0%)

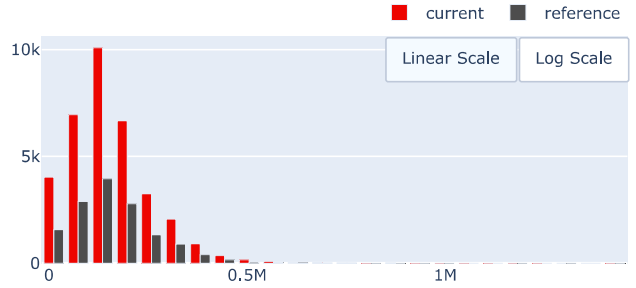


	current	reference
count	34687	14155
mean	188997.37	191298.05
std	105374.9	106149.45
min	12285.0	13769.0

fnlwgt

num

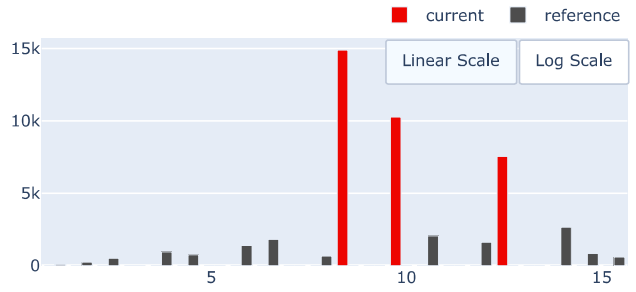
25%	117618.0	117295.0
50%	177675.0	179973.0
75%	236393.5	240462.5
max	1490400.0	1455435.0
unique	22172 (63.92%)	11585 (81.84%)
most common	203488.0 (0.06%)	238917.0 (0.05%)
missing	0 (0.0%)	0 (0.0%)
infinite	0 (0.0%)	0 (0.0%)



education-num

num

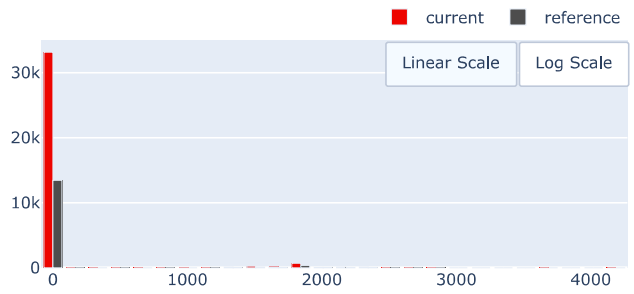
	current	reference
count	32687	14155
mean	10.24	9.68
std	1.57	4.06
min	9.0	1.0
25%	9.0	6.0
50%	10.0	11.0
75%	10.0	14.0
max	13.0	16.0
unique	3 (0.01%)	13 (0.09%)
most common	9.0 (42.93%)	14.0 (18.77%)



missing	2000 (5.77%)	0 (0.0%)
infinite	0 (0.0%)	0 (0.0%)

capital-loss
num

	current	reference
count	34687	14155
mean	83.32	97.74
std	391.26	430.27
min	0.0	0.0
25%	0.0	0.0
50%	0.0	0.0
75%	0.0	0.0
max	4356.0	3900.0
unique	92 (0.27%)	85 (0.6%)
most common	0.0 (95.5%)	0.0 (94.9%)
missing	0 (0.0%)	0 (0.0%)
infinite	0 (0.0%)	0 (0.0%)

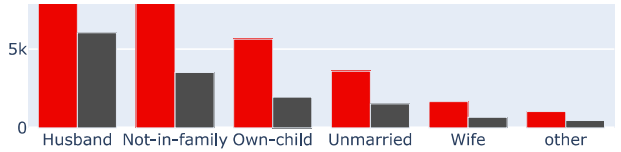


relationship

	current	reference
count	34687	14155
unique	6 (0.02%)	6 (0.04%)
most common	Husband (39.44%)	Husband (42.63%)

cat

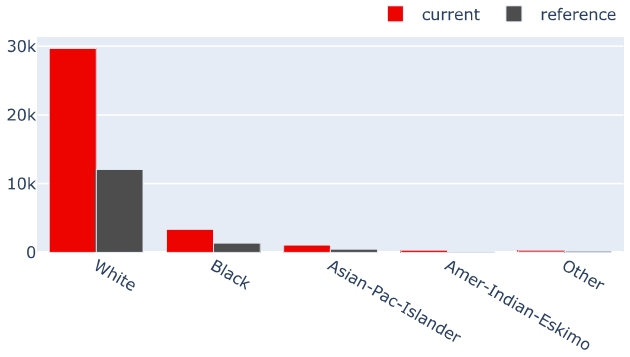
most common	Husband (39.44%)	Husband (42.05%)
missing	0 (0.0%)	0 (0.0%)
new categories	0	
missing categories	0	



race

cat

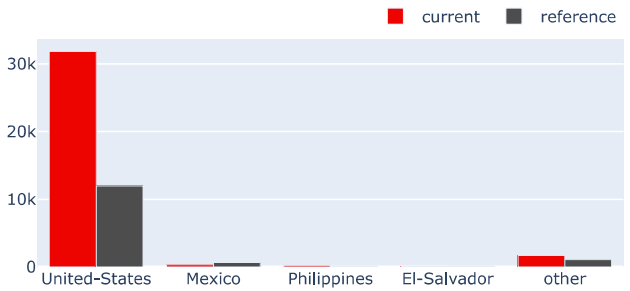
	current	reference
count	34687	14155
unique	5 (0.01%)	5 (0.04%)
most common	White (85.65%)	White (85.14%)
missing	0 (0.0%)	0 (0.0%)
new categories	0	
missing categories	0	



native-country

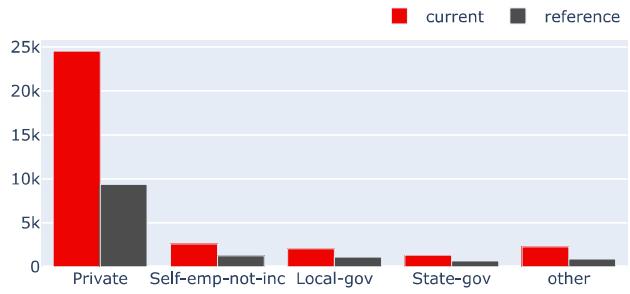
cat

	current	reference
count	34124	13861
unique	41 (0.12%)	40 (0.28%)
most common	United-States (91.82%)	United-States (84.66%)
missing	563 (1.62%)	294 (2.08%)
new categories	1	
missing categories	0	



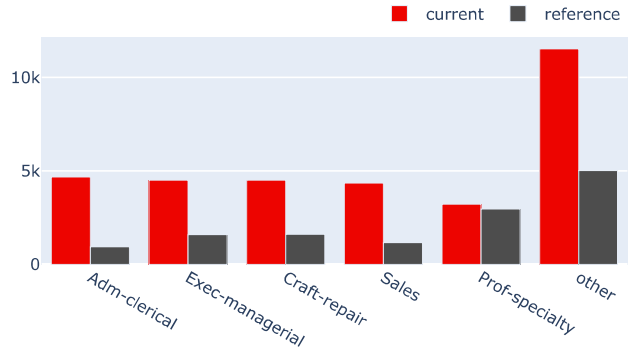
workclass
cat

	current	reference
count	32784	13259
unique	8 (0.02%)	8 (0.06%)
most common	Private (70.69%)	Private (66.31%)
missing	1903 (5.49%)	896 (6.33%)
new categories	0	
missing categories	0	



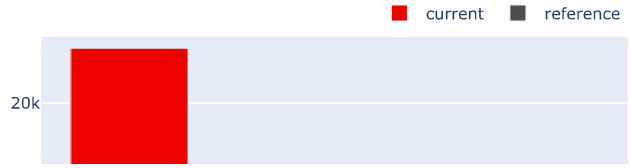
occupation
cat

	current	reference
count	32780	13253
unique	14 (0.04%)	14 (0.1%)
most common	Adm-clerical (13.46%)	Prof-specialty (20.88%)
missing	1907 (5.5%)	902 (6.37%)
new categories	0	
missing categories	0	



class
cat

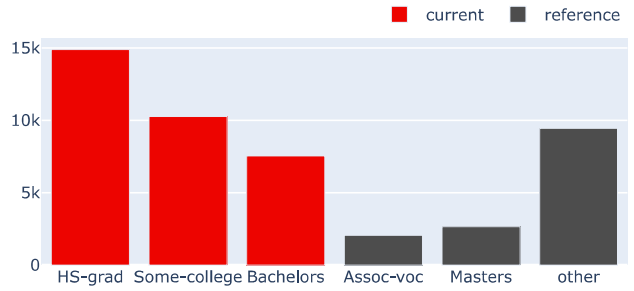
	current	reference
count	34687	14155
unique	2 (0.01%)	2 (0.01%)
most common	<=50K (77.29%)	<=50K (73.1%)



missing	0 (0.0%)	0 (0.0%)
new categories	0	
missing categories	0	

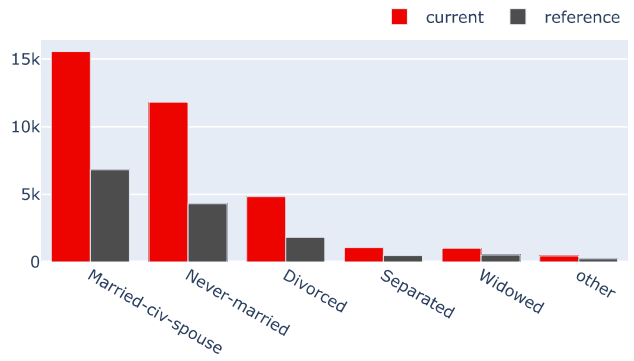
education
cat

	current	reference
count	32687	14155
unique	3 (0.01%)	13 (0.09%)
most common	HS-grad (42.93%)	Masters (18.77%)
missing	2000 (5.77%)	0 (0.0%)
new categories	4	
missing categories	13	



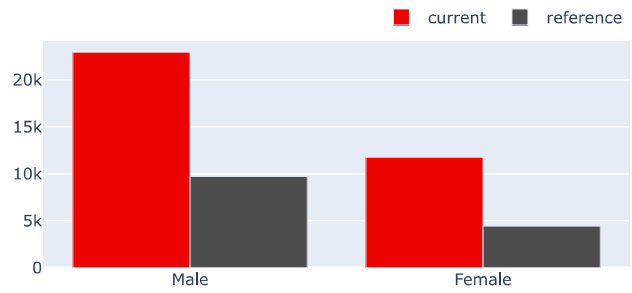
marital-status
cat

	current	reference
count	34687	14155
unique	7 (0.02%)	7 (0.05%)
most common	Married-civ-spouse (44.85%)	Married-civ-spouse (48.19%)
missing	0 (0.0%)	0 (0.0%)
new categories	0	
missing categories	0	



sex
cat

	current	reference
count	34687	14155
unique	2 (0.01%)	2 (0.01%)
most common	Male (66.12%)	Male (68.63%)
missing	0 (0.0%)	0 (0.0%)
new categories	0	
missing categories	0	



Dataset Missing Values

8373 (1.609%)
Missing values (Current data)

2092 (0.985%)
Missing values (Reference data)

Current dataset

Table Histogram

Value	Count
education	2000
education-num	2000
occupation	1907
workclass	1903
native-country	563
age	0
fnlwgt	0

marital-status	0
relationship	0
race	0
sex	0
capital-gain	0
capital-loss	0
hours-per-week	0
class	0

Reference dataset

Table Histogram

Value	Count
occupation	902
workclass	896
native-country	294
age	0
fnlwgt	0
education	0
education-num	0
marital-status	0
relationship	0
race	0
sex	0
capital-gain	0
capital-loss	0
hours-per-week	0
class	0

```
data_quality_report.save_html('data_quality_report.html')
```

C. Create the Regression Performance Metric Report

```
regression_performance_report = Report(metrics=[
    RegressionPreset(),
])

regression_performance_report.run(reference_data=housing_ref.sort_index(), current_data=housing_cur.sort_index())
regression_performance_report
```

Regression Model Performance. Target: 'target'

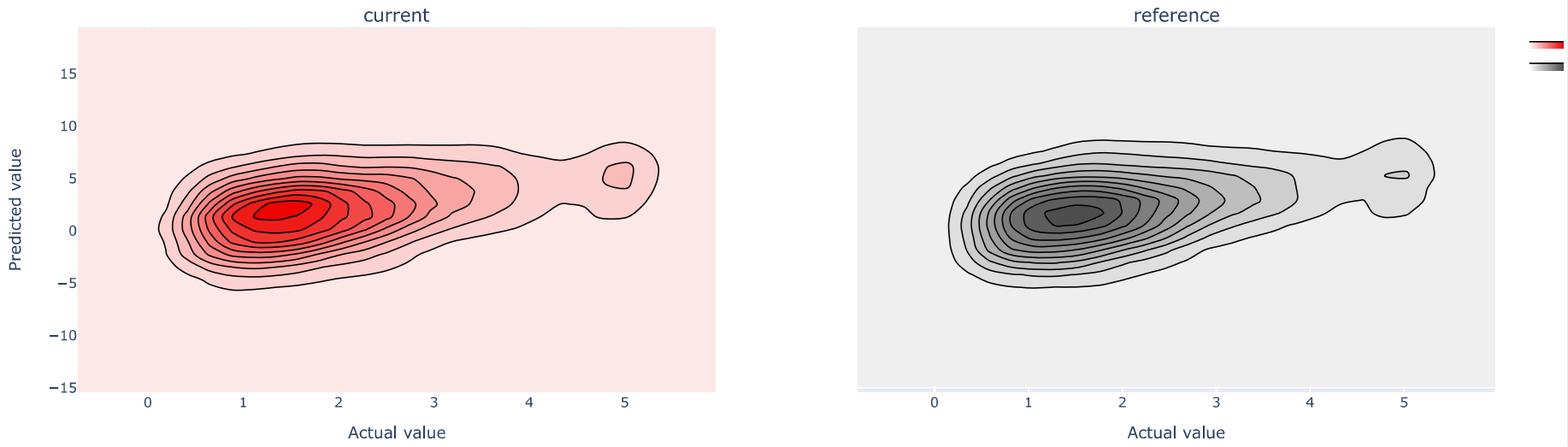
Current: Model Quality (+/- std)

ME	MAE	MAPE
-0.04 (2.95)	2.34 (1.8)	156.62 (1.7)

Reference: Model Quality (+/- std)

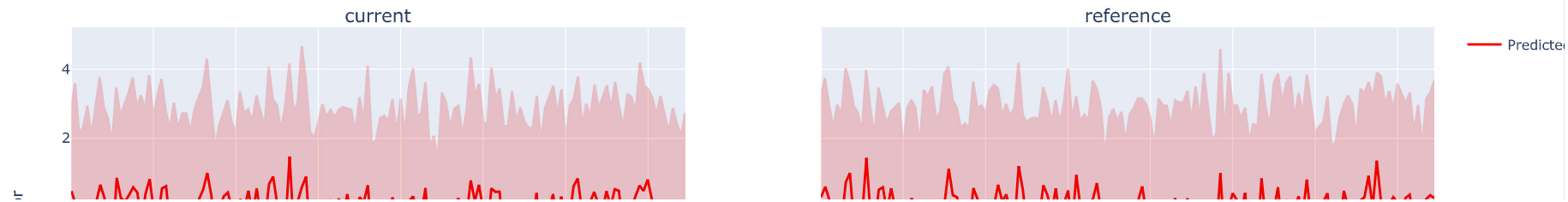
ME	MAE	MAPE
-0.01 (2.96)	2.36 (1.8)	156.7 (1.74)

Predicted vs Actual

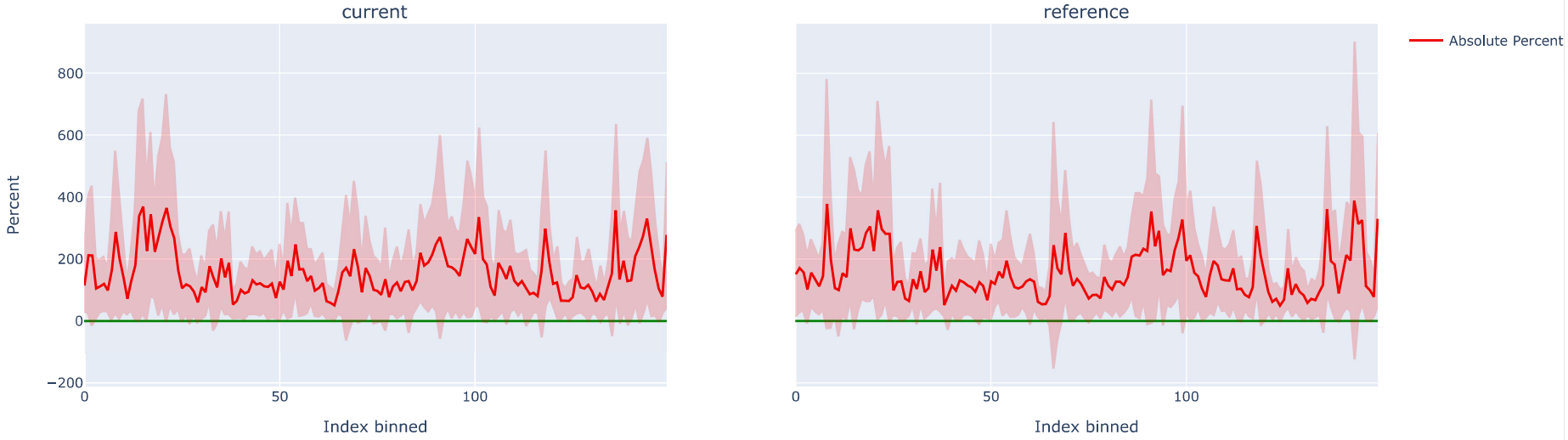


Predicted vs Actual in Time

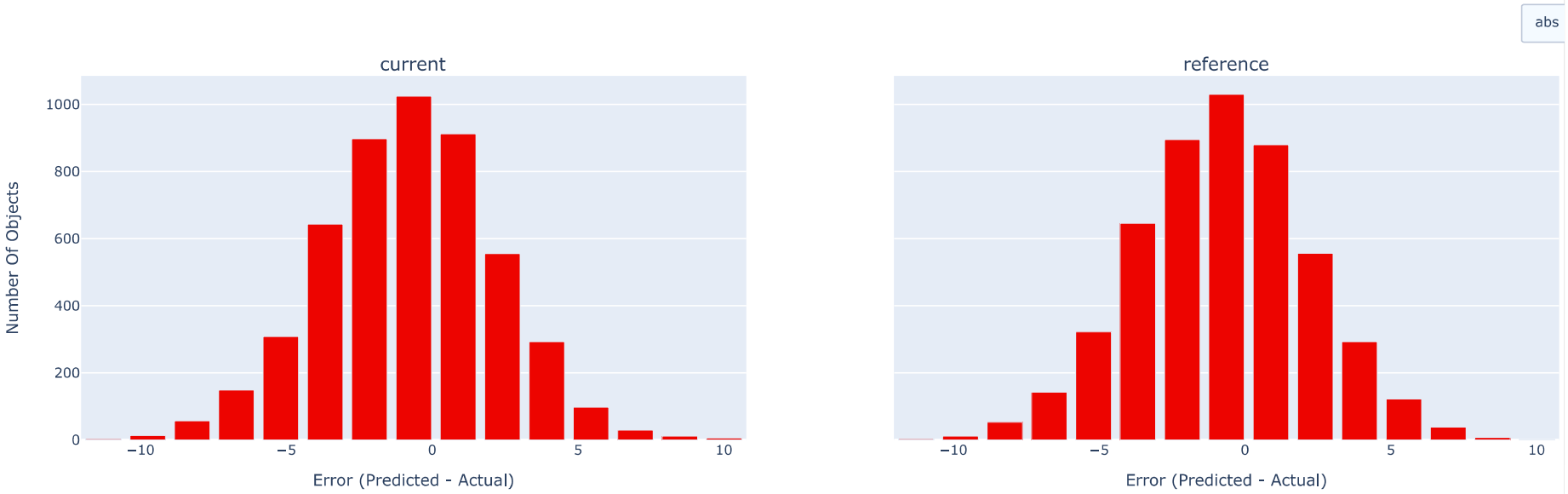
Error (Predicted - Actual)



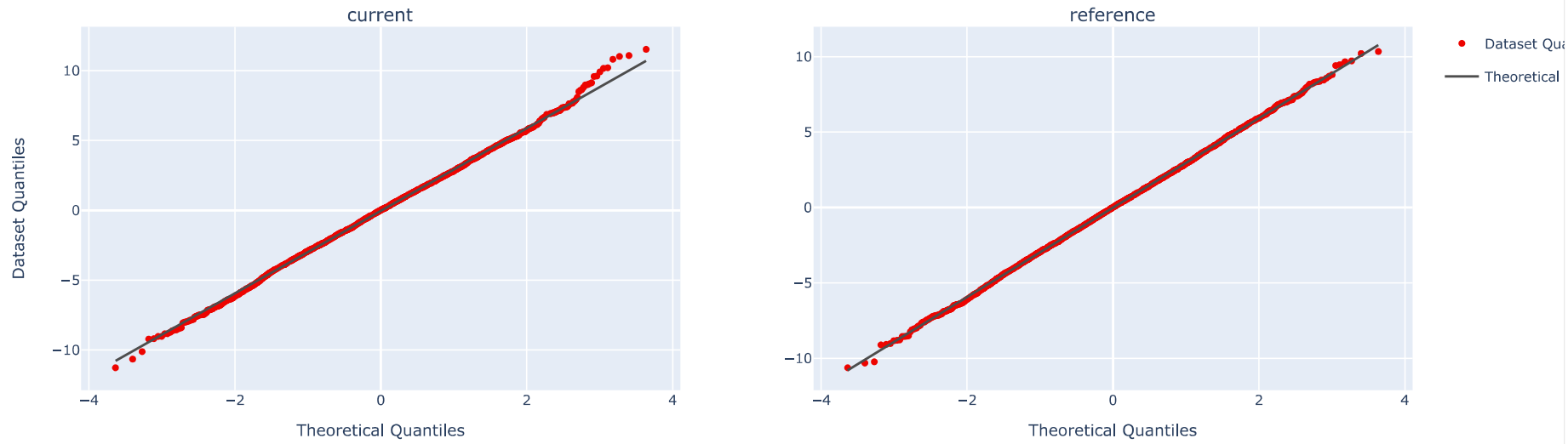
Absolute Percentage Error



Error Distribution



Error Normality



Error Bias Table

Current: Mean Error per Group (+/- std)