In [ ]:

```
# !pip install piml
```

```
Requirement already satisfied: piml in /usr/local/lib/python3.10/dist-packages (0.6.0.pos
t2)
Requirement already satisfied: ipykernel in /usr/local/lib/python3.10/dist-packages (from
piml) (5.5.6)
Requirement already satisfied: ipywidgets>=7.7.0 in /usr/local/lib/python3.10/dist-packag
es (from piml) (7.7.1)
Requirement already satisfied: joblib>=1.1.0 in /usr/local/lib/python3.10/dist-packages (
from piml) (1.4.2)
Requirement already satisfied: ipython>=7.12.0 in /usr/local/lib/python3.10/dist-packages
(from piml) (7.34.0)
Requirement already satisfied: numpy<1.24.0,>=1.21.4 in /usr/local/lib/python3.10/dist-pa
ckages (from piml) (1.23.5)
Requirement already satisfied: scipy<=1.10.1,>=1.5.3 in /usr/local/lib/python3.10/dist-pa
ckages (from piml) (1.10.1)
Requirement already satisfied: pandas<2.0.0,>=1.3.5 in /usr/local/lib/python3.10/dist-pac
kages (from piml) (1.5.3)
Requirement already satisfied: matplotlib<3.8.0,>=3.1.2 in /usr/local/lib/python3.10/dist
-packages (from piml) (3.7.1)
Requirement already satisfied: seaborn>=0.11.2 in /usr/local/lib/python3.10/dist-packages
(from piml) (0.13.1)
Requirement already satisfied: xlrd>=1.2.0 in /usr/local/lib/python3.10/dist-packages (fr
om piml) (2.0.1)
Requirement already satisfied: scikit-learn<1.4.0,>=0.24.2 in /usr/local/lib/python3.10/d
ist-packages (from piml) (1.3.2)
Requirement already satisfied: xgboost>=1.4.2 in /usr/local/lib/python3.10/dist-packages
(from piml) (2.1.1)
Requirement already satisfied: statsmodels>=0.12.2 in /usr/local/lib/python3.10/dist-pack
ages (from piml) (0.14.4)
Requirement already satisfied: lime>=0.2.0.1 in /usr/local/lib/python3.10/dist-packages (
from piml) (0.2.0.1)
Requirement already satisfied: shap>=0.39.0 in /usr/local/lib/python3.10/dist-packages (f
rom piml) (0.46.0)
Requirement already satisfied: torch>=1.11.0 in /usr/local/lib/python3.10/dist-packages (
from piml) (2.4.1+cu121)
Requirement already satisfied: pygam==0.8.0 in /usr/local/lib/python3.10/dist-packages (f
rom piml) (0.8.0)
Requirement already satisfied: natsort>=8.2.0 in /usr/local/lib/python3.10/dist-packages
(from piml) (8.4.0)
Requirement already satisfied: psutil>=5.9.0 in /usr/local/lib/python3.10/dist-packages (
from piml) (5.9.5)
Requirement already satisfied: dill>=0.3.6 in /usr/local/lib/python3.10/dist-packages (fr
om piml) (0.3.9)
Requirement already satisfied: packaging>=20.5 in /usr/local/lib/python3.10/dist-packages
(from piml) (24.1)
Requirement already satisfied: networkx>=2.6.3 in /usr/local/lib/python3.10/dist-packages
(from piml) (3.3)
Requirement already satisfied: numba<0.57.0 in /usr/local/lib/python3.10/dist-packages (f
rom piml) (0.56.4)
Requirement already satisfied: jupyter-client<=7.4.9 in /usr/local/lib/python3.10/dist-pa
ckages (from piml) (6.1.12)
Requirement already satisfied: optbinning>=0.17.3 in /usr/local/lib/python3.10/dist-packa
ges (from piml) (0.19.0)
Requirement already satisfied: momentchi2 in /usr/local/lib/python3.10/dist-packages (fro
m piml) (0.1.8)
Requirement already satisfied: future in /usr/local/lib/python3.10/dist-packages (from py
gam==0.8.0->piml) (1.0.0)
Requirement already satisfied: progressbar2 in /usr/local/lib/python3.10/dist-packages (f
rom pygam==0.8.0->piml) (4.5.0)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-package
s (from ipython>=7.12.0->piml) (71.0.4)
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-packages (fro
m ipython>=7.12.0->piml) (0.19.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from
ipython>=7.12.0->piml) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (fr
```

```
Requirement already satisfied: picklesha
om ipython>=7.12.0->piml) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/dist-packages
(from ipython>=7.12.0->piml) (5.7.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local
/lib/python3.10/dist-packages (from ipython>=7.12.0->piml) (3.0.48)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from
ipython>=7.12.0->piml) (2.18.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from
ipython>=7.12.0->piml) (0.2.0)
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packag
es (from ipython>=7.12.0->piml) (0.1.7)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (fr
om ipython>=7.12.0->piml) (4.9.0)
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.10/dist-
packages (from ipywidgets>=7.7.0->piml) (0.2.0)
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/python3.10/dis
t-packages (from ipywidgets>=7.7.0->piml) (3.6.9)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.10/dis
t-packages (from ipywidgets>=7.7.0->piml) (3.0.13)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/dist-packages (f
rom ipykernel->piml) (6.3.3)
Requirement already satisfied: jupyter-core>=4.6.0 in /usr/local/lib/python3.10/dist-pack
ages (from jupyter-client<=7.4.9->piml) (5.7.2)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.10/dist-packages (from
jupyter-client<=7.4.9->piml) (24.0.1)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.10/dist-pac
kages (from jupyter-client<=7.4.9->piml) (2.8.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from lime
>=0.2.0.1->piml) (4.66.5)
Requirement already satisfied: scikit-image>=0.12 in /usr/local/lib/python3.10/dist-packa
ges (from lime>=0.2.0.1->piml) (0.24.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-package
s (from matplotlib<3.8.0,>=3.1.2->piml) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (f
rom matplotlib<3.8.0,>=3.1.2->piml) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packag
es (from matplotlib<3.8.0,>=3.1.2->piml) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packag
es (from matplotlib<3.8.0,>=3.1.2->piml) (1.4.7)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (
from matplotlib<3.8.0,>=3.1.2->piml) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-package
s (from matplotlib<3.8.0,>=3.1.2->piml) (3.1.4)
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in /usr/local/lib/python3.10/di
st-packages (from numba<0.57.0->piml) (0.39.1)
Requirement already satisfied: ortools>=9.4 in /usr/local/lib/python3.10/dist-packages (f
rom optbinning>=0.17.3->piml) (9.7.2996)
Requirement already satisfied: ropwr>=1.0.0 in /usr/local/lib/python3.10/dist-packages (f
rom optbinning>=0.17.3->piml) (1.0.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (f
rom pandas<2.0.0,>=1.3.5->piml) (2024.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-pac
kages (from scikit-learn<1.4.0,>=0.24.2->piml) (3.5.0)
Requirement already satisfied: slicer==0.0.8 in /usr/local/lib/python3.10/dist-packages (
from shap>=0.39.0->piml) (0.0.8)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (fr
om shap>=0.39.0->piml) (2.2.1)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-packages (f
rom statsmodels>=0.12.2->piml) (0.5.6)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from
torch>=1.11.0->piml) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist
-packages (from torch>=1.11.0->piml) (4.12.2)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from tor
ch>=1.11.0->piml) (1.13.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from to
rch>=1.11.0->piml) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from to
rch>=1.11.0->piml) (2024.6.1)
Requirement already satisfied: nvidia-nccl-cu12 in /usr/local/lib/python3.10/dist-package
s (from xgboost>=1.4.2->piml) (2.23.4)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-pack
```

Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython>=7.12.0->piml) (0.8.4)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from jupyter-core>=4.6.0->jupyter-client<=7.4.9->piml) (4.3.6)
Requirement already satisfied: absl-py>=0.13 in /usr/local/lib/python3.10/dist-packages (from ortools>=9.4->optbinning>=0.17.3->piml) (1.4.0)
Requirement already satisfied: protobuf>=4.23.3 in /usr/local/lib/python3.10/dist-packages (from ortools>=9.4->optbinning>=0.17.3->piml) (5.28.2)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.6->statsmodels>=0.12.2->piml) (1.16.0)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages (from pexpect>4.3->ipython>=7.12.0->piml) (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=7.12.0->piml) (0.2.13)
Requirement already satisfied: cvxpy>=1.1.14 in /usr/local/lib/python3.10/dist-packages (from ropwr>=1.0.0->optbinning>=0.17.3->piml) (1.5.3)
Requirement already satisfied: imageio>=2.33 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime>=0.2.0.1->piml) (2.35.1)
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime>=0.2.0.1->piml) (2024.9.20)
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime>=0.2.0.1->piml) (0.4)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.10/dist-packages (from widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (6.5.5)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0->piml) (2.1.5)
Requirement already satisfied: python-utils>=3.8.1 in /usr/local/lib/python3.10/dist-packages (from progressbar2->pygam==0.8.0->piml) (3.9.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.11.0->piml) (1.3.0)
Requirement already satisfied: osqp>=0.6.2 in /usr/local/lib/python3.10/dist-packages (from cvxpy>=1.1.14->ropwr>=1.0.0->optbinning>=0.17.3->piml) (0.6.7.post0)
Requirement already satisfied: ecos>=2 in /usr/local/lib/python3.10/dist-packages (from cvxpy>=1.1.14->ropwr>=1.0.0->optbinning>=0.17.3->piml) (2.0.14)
Requirement already satisfied: clarabel>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from cvxpy>=1.1.14->ropwr>=1.0.0->optbinning>=0.17.3->piml) (0.9.0)
Requirement already satisfied: scs>=3.2.4.post1 in /usr/local/lib/python3.10/dist-packages (from cvxpy>=1.1.14->ropwr>=1.0.0->optbinning>=0.17.3->piml) (3.2.7)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (23.1.0)
Requirement already satisfied: nbformat in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (5.10.4)
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (6.5.4)
Requirement already satisfied: nest-asyncio>=1.5 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (1.6.0)
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (0.18.1)
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (0.21.0)
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (1.1.0)
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/python3.10/dist-packages (from nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (0.2.4)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (4.12.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (0.4)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (0.3.0)

```
iml) (0.3.0)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.10/dist-packag
es (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->pim
l) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.10/dist-packages
(from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml)
(0.10.0)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.10/dist-pac
kages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->
piml) (1.5.1)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-packages (from
nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (1.3.0
)
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.10/dist-pac
kages (from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml
) (2.20.0)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.10/dist-packages
(from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (4.2
3.0)
Requirement already satisfied: qdldl in /usr/local/lib/python3.10/dist-packages (from osq
p>=0.6.2->cvxpy>=1.1.14->ropwr>=1.0.0->optbinning>=0.17.3->piml) (0.1.7.post4)
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.10/dist-pac
kages (from argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->p
iml) (21.2.0)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (
from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7
.7.0->piml) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/pyt
hon3.10/dist-packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextensio
n~=3.6.0->ipywidgets>=7.7.0->piml) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-pack
ages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets>=7.7.0->piml) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages
(from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=
7.7.0->piml) (0.20.0)
Requirement already satisfied: jupyter-server<3,>=1.8 in /usr/local/lib/python3.10/dist-p
ackages (from notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->widgetsnbextension
~=3.6.0->ipywidgets>=7.7.0->piml) (1.24.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-packages (fr
om argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets>=7.7.0->piml) (1.17.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (
from beautifulsoup4->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets
>=7.7.0->piml) (2.6)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (f
rom bleach->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->
piml) (0.5.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from
cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.
0->ipywidgets>=7.7.0->piml) (2.22)
Requirement already satisfied: anyio<4,>=3.1.0 in /usr/local/lib/python3.10/dist-packages
(from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->wi
dgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (3.7.1)
Requirement already satisfied: websocket-client in /usr/local/lib/python3.10/dist-package
s (from jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook>=4.4.1->
widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (1.8.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from
anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->notebook
>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (f
rom anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->note
book>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages
(from anyio<4,>=3.1.0->jupyter-server<3,>=1.8->notebook-shim>=0.2.3->nbclassic>=0.4.7->no
tebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7.7.0->piml) (1.2.2)
```

Data Pipeline

In [18]:

```
from piml import Experiment
```

In [3]:

```
exp_tw_credit = Experiment()
exp_bike_sharing = Experiment()
exp_tw_credit.data_loader("TaiwanCredit", silent=True)
exp_bike_sharing.data_loader("BikeSharing", silent=True)
```

In [24]:

```
tw_credit = exp_tw_credit
bike_sharing = exp_bike_sharing
```

In [4]:

```
exp = Experiment()
```

In [30]:

```
exp.data_loader()
```

In [27]:

```
exp.data_summary()
```

In [8]:

```
exp.data_loader()
```

In [9]:

```
exp.data_summary()
```

Q1 (5pts): Using PiML, show data summary tables for both data sets. What is the max value of `cnt` in the bike share data? How many unique values are there in the `EDUCATION` data point of the Taiwan credit data?

- **For Taiwan Credit the number of unique values for EDUCATION = 4**
-

```
exp.data_summary()
```

✓ 0.5s

Data Shape:(30000, 24)

Numerical Attributes | Categorical Attributes

	name	n_missing	n_unique	top1	top2	top3	n_others
0	SEX	0	2	2.0 : 18112	1.0 : 11888	0	0
1	EDUCATION	0	4	2.0 : 14030	1.0 : 10585	3.0 : 4917	468

2	MARRIAGE	0	3	2.0 : 15964	1.0 : 13659	0.0 : 377	0
3	FlagDefault	0	2	0.0 : 23364	1.0 : 6636	0	0

- **For Bike Sharing the max value of cnt is = 977.0**
-

```
exp.data_summary()
```

✓ 0.4s

Data Shape:(17379, 13)

Numerical Attributes | Categorical Attributes

	name	n_missing	mean	std	min	q1	median	q3	max
0	mnth	0.0	6.5378	3.4388	1.0	4.0	7.0	10.0	12.0
1	hr	0.0	11.5468	6.9144	0.0	6.0	12.0	18.0	23.0
2	weekday	0.0	3.0037	2.0058	0.0	1.0	3.0	5.0	6.0
3	temp	0.0	0.497	0.1926	0.02	0.34	0.5	0.66	1.0
4	atemp	0.0	0.4758	0.1719	0.0	0.3333	0.4848	0.6212	1.0
5	hum	0.0	0.6272	0.1929	0.0	0.48	0.63	0.78	1.0
6	windspeed	0.0	0.1901	0.1223	0.0	0.1045	0.194	0.2537	0.8507
7	cnt	0.0	189.4631	181.3876	1.0	40.0	142.0	281.0	977.0

Model Pipeline

In [32]:

```
from xgboost import XGBClassifier, XGBRegressor
from sklearn.neural_network import MLPClassifier, MLPRegressor

exp_bike_sharing.data_prepare(random_state=100, target="cnt", test_ratio=0.2, task_type=
"regression", split_method="random", silent=True)
exp_tw_credit.data_prepare(random_state=100, target="FlagDefault", test_ratio=0.2, task_t
ype="classification", split_method="random", silent=True)

exp_bike_sharing.model_train(model=XGBRegressor(max_depth=5, n_estimators=500), name="XG
B_Bike_Sharing")
exp_bike_sharing.model_train(model=MLPRegressor(hidden_layer_sizes=[10]*2, activation="r
elu", random_state=0, early_stopping=True), name="DNN_Bike_Sharing")

exp_tw_credit.model_train(model=XGBClassifier(max_depth=5, n_estimators=500), name="XGB_T
W_Credit")
exp_tw_credit.model_train(model=MLPClassifier(hidden_layer_sizes=[10]*2, activation="relu
", random_state=0, early_stopping=True), name="DNN_TW_Credit")
```

Q2 (5pts): What does the `random_state` parameter do? Will the XGB model for the bike sharing data turn out identical every time you run the code above?

- **random_state**
  - **The random_state parameter in machine learning code controls the randomness in processes like data**

splitting and model initialization. By setting it to a specific number, we ensure that these random steps will always produce the same results. This makes experiments reproducible, so we can tell if changes we make to the code actually improve performance.
    - In machine learning, many processes involve randomness, like splitting data into training and testing sets, initializing model parameters, or selecting features. random_state lets us control this randomness.
    - By setting random_state to a specific number (like 100 here), we ensure that these random processes will produce the same results every time you run the code. This makes experiments more reproducible and helps identify if changes that are made actually impact the model's performance.
    - If we don't specify random_state, the randomness will be based on the system's current time, which means the results will vary each time.
- Will the XGB Model be Identical?
    - The XGBoost model for bike sharing will not be completely identical every time we run it. This is because even though the data split will be the same due to random_state, XGBoost still has internal randomness in its training process.
    - The data will be split into training and testing sets using the same random seed.
    - The XGBoost model might have internal randomness in its initialization. However, random_state is not used directly in the XGBRegressor object. So the model initialization is still random.The training process itself can involve randomness, depending on the XGBoost algorithm's implementation.

Q3 (2.5pts): What are the hyperparameters specified in each type of model?

- XGBoost Models (Regression and Classification):
    - max_depth: Controls the maximum depth of each individual decision tree in the ensemble. Higher values allow the model to learn more complex relationships but increase the risk of overfitting. Here it's set to 5.
    - n_estimators: Determines the number of decision trees in the ensemble. More trees generally lead to better performance but also increase training time. Here it's set to 500.
- MLP Models (Regression and Classification):
    - hidden_layer_sizes: Defines the number of neurons in each hidden layer of the neural network. Here, it's set to [10]*2, creating two hidden layers with 10 neurons each.
    - activation: Specifies the activation function used in the hidden layers. relu common as it can handle non-linear relationships in the data.
    - random_state: Controls the random initialization of the neural network's weights. Setting it to 0 ensures consistent weight initialization between runs.
    - early_stopping: This parameter enables early stopping, a technique that prevents overfitting by stopping training when the model's performance on a validation set starts to decrease.

Bike sharing dataset with XGB model and DNN model

Q4 (10 pts): Compute the reliability table (including empirical coverage and average bandwidth) of the two Bike Sharing models; which model is more reliable?

In [46]:

```
# XGB
reliability_xgb = exp_bike_sharing.model_diagnose(model="XGB_Bike_Sharing", show="reliability_table")

# DNN
reliability_dnn = exp_bike_sharing.model_diagnose(model="DNN_Bike_Sharing", show="reliability_table")
```

	Empirical Coverage	Average Bandwidth
0	0.8813	0.1042

	Empirical Coverage	Average Bandwidth
0	0.8856	0.1874

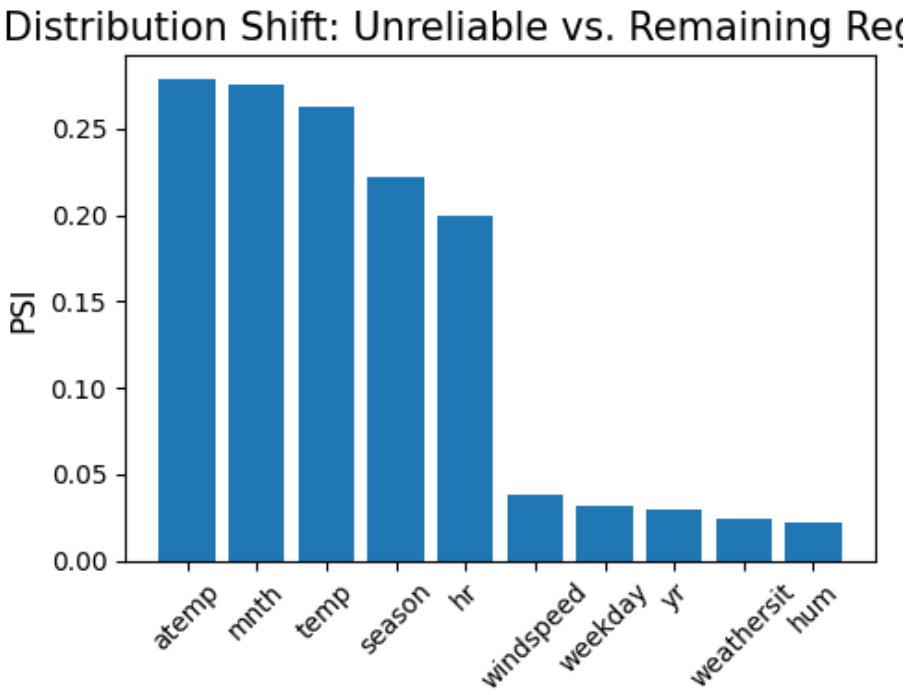
	Empirical Coverage	Average Bandwidth

XGBoost has a narrower bandwith than the DNN Model. XGB would be more reliable than the DNN model

Q5 (5 pts): Use the `reliability_distance` function in PiML on the XGB model of the bike sharing data. List the top 5 features that have the largest distributional distance between unreliable regions and reliable regions based on PSI score. (You can use the defaults for all other arguments of `model_diagnose`)

In [37]:

```
exp_bike_sharing.model_diagnose(model="XGB_Bike_Sharing", show="reliability_distance", al
pha=0.1,
                    threshold=1.1, distance_metric="PSI", figsize=(5, 4))
```

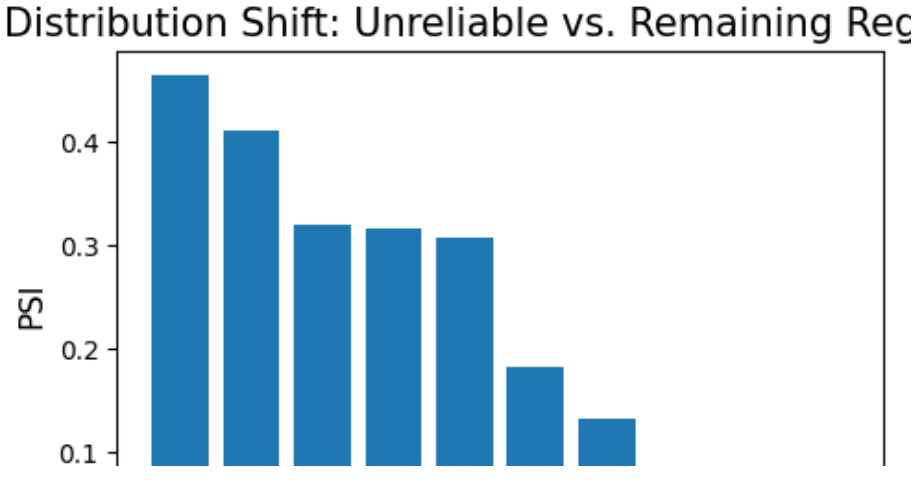


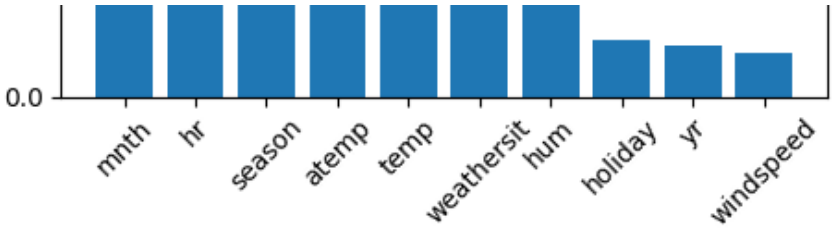
The top 5 features of ths XGB are: XGB: atemp, mnth, temp, season, hr

Q6 (5 pts): Do the same analysis as in Q5, but use a threshold of 1.3. Why did the PSI values increase?

In [38]:

```
exp_bike_sharing.model_diagnose(model="XGB_Bike_Sharing", show="reliability_distance", al
pha=0.1,
                    threshold=1.3, distance_metric="PSI", figsize=(5, 4))
```



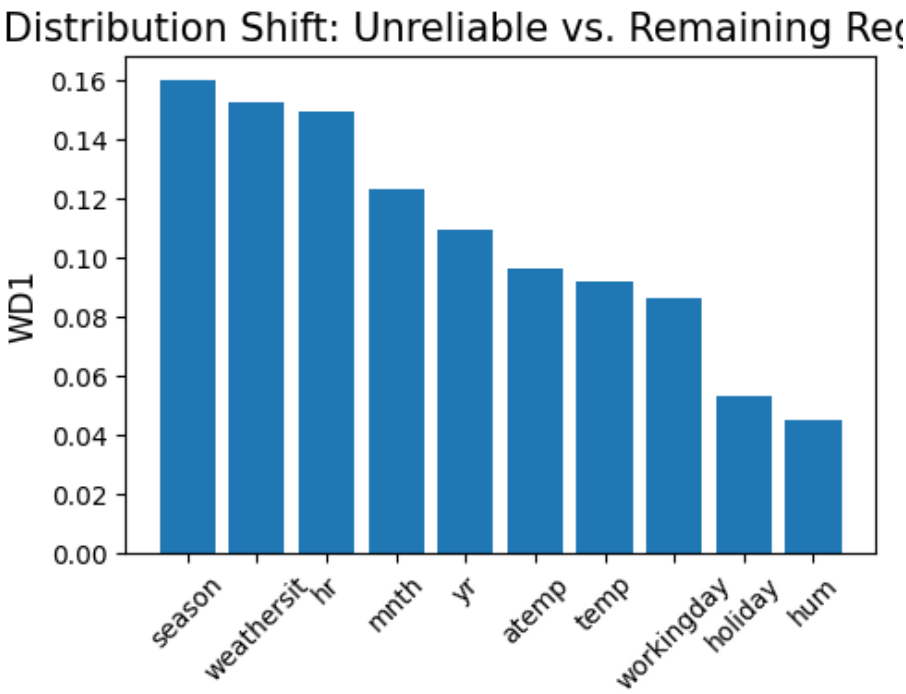


- **The PSI values increased because the threshold for defining a region as "unreliable" was changed from 1.0 to 1.3.**
- **[PSI (Population Stability Index) measures how much the distribution of a variable has changed between two populations (e.g., training vs. testing data). A higher threshold means a larger difference is needed to be considered unreliable.]**
- **A higher threshold (1.3) means that more regions are now considered "unreliable." This is because a larger change in the distribution is needed to trigger the "unreliable" label. Since more regions are now considered unreliable, the overall impact on the distribution is greater, resulting in higher PSI values.**

Q7 (2.5 pts): Do the same analysis as in Q6, but use the Wasserstein distance measure instead of PSI. Are the results the same?

In [40]:

```
exp_bike_sharing.model_diagnose(model="XGB_Bike_Sharing", show="reliability_distance", al
pha=0.1, threshold= 1.3, distance_metric='WD1',figsize=(5, 4))
```



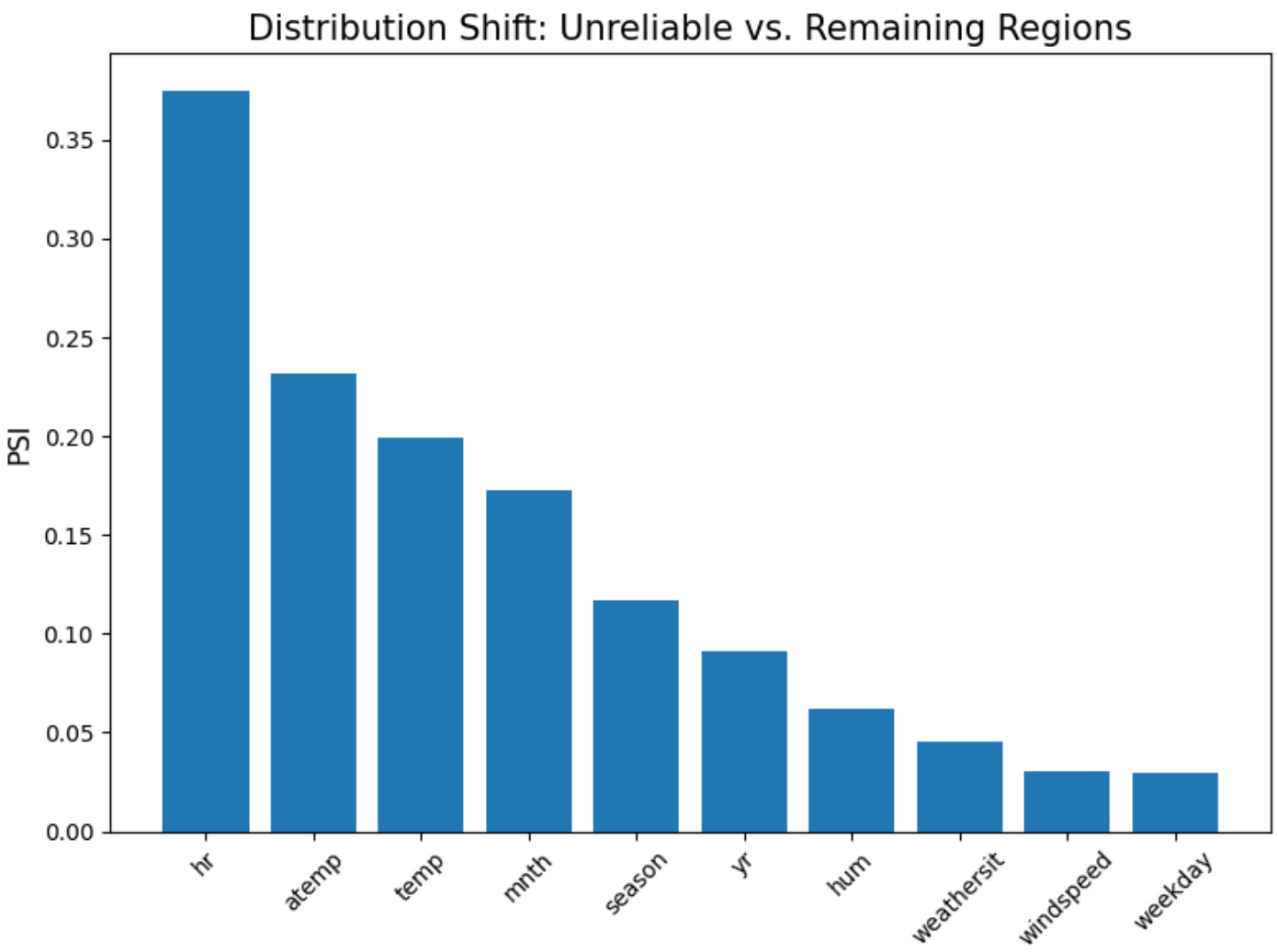
- **No, the results of the PSI and Wasserstein distance analyses are not the same.**
- **While both measure distribution shifts, they focus on different aspects:**
  - **PSI: Detects changes in the proportions of values within a distribution.**
  - **Wasserstein Distance: Measures the overall shape change in the distribution.**
- **They highlight different features as potentially unreliable, leading to different results.**

Q8 (5 pts): Do the same analysis as in Q5, but for the DNN model. Are the results the same as Q5? Why or why not?

In [41]:

```
exp_bike_sharing.model_diagnose(model="DNN_Bike_Sharing", show="reliability_distance")
```

```
# exp_bike_sharing.model_diagnose(model="XGB_Bike_Sharing", show="reliability_distance",
#                                 alpha=0.1,threshold=1.1, distance_metric="PSI", figsize=
(5, 4))
```



- **The results are not the same. The DNN model shows more sensitivity to the 'hr' feature compared to the XGB model.**

  - DNNs have a more complex structure and can capture more time-dependent patterns, which makes them more sensitive to changes in hourly trends.

  - XGBoost relies on tree splitting algorithms, which might be simpler and less sensitive to subtle changes in hourly patterns compared to the DNN's ability to handle non-linear relationships.

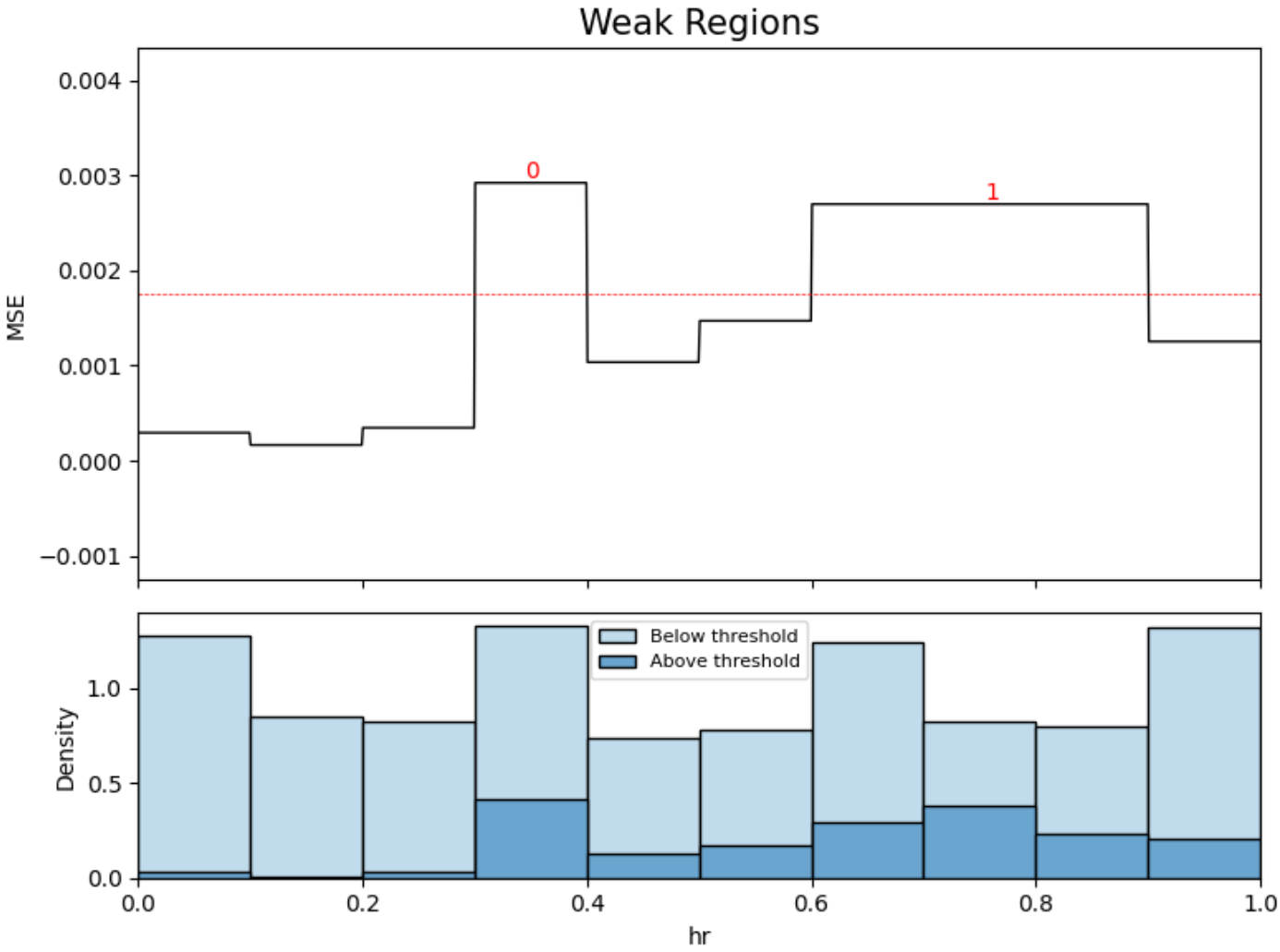
Q9 (5 pts): How could the analysis above help you if you were in charge of monitoring a model in production?

- **This analysis helps monitor model stability in production by identifying features with significant distribution shifts.**
- **Features with high PSI or Wasserstein Distance indicate potential data drift, which can lead to model performance degradation.**
- **Monitoring these features helps identify the need to retrain the model or adapt the features to maintain accuracy. These insights also help detect the severity of model drift, pinpoint the features causing it, and assess the overall stability over time, contributing to fairness and mitigating biases.**

Q10 (5 pts): For the `XGB_Bike_Sharing` model, write the PiML code to show the weak regions of the `hr` feature. Use MSE as the measurement metric and include the test data in the results. Use the `histogram` method for slicing.

In [42]:

```
exp_bike_sharing.model_diagnose(model="XGB_Bike_Sharing", show="weakspot", slice_method="
histogram", slice_features=["hr"], metric="MSE", use_test=True)
```



Q11 (5 pts): Is hour 3 or hour 4 part of a weak region in the analysis of Q10?

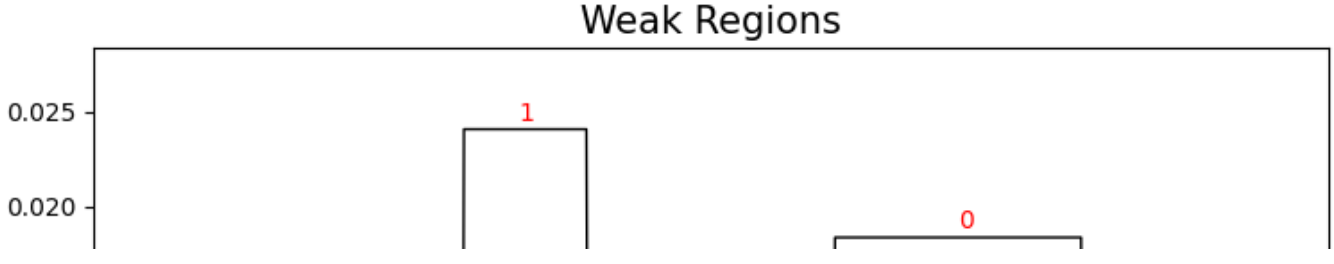
Neither hour 3 nor hour 4 are part of a weak region in the analysis.

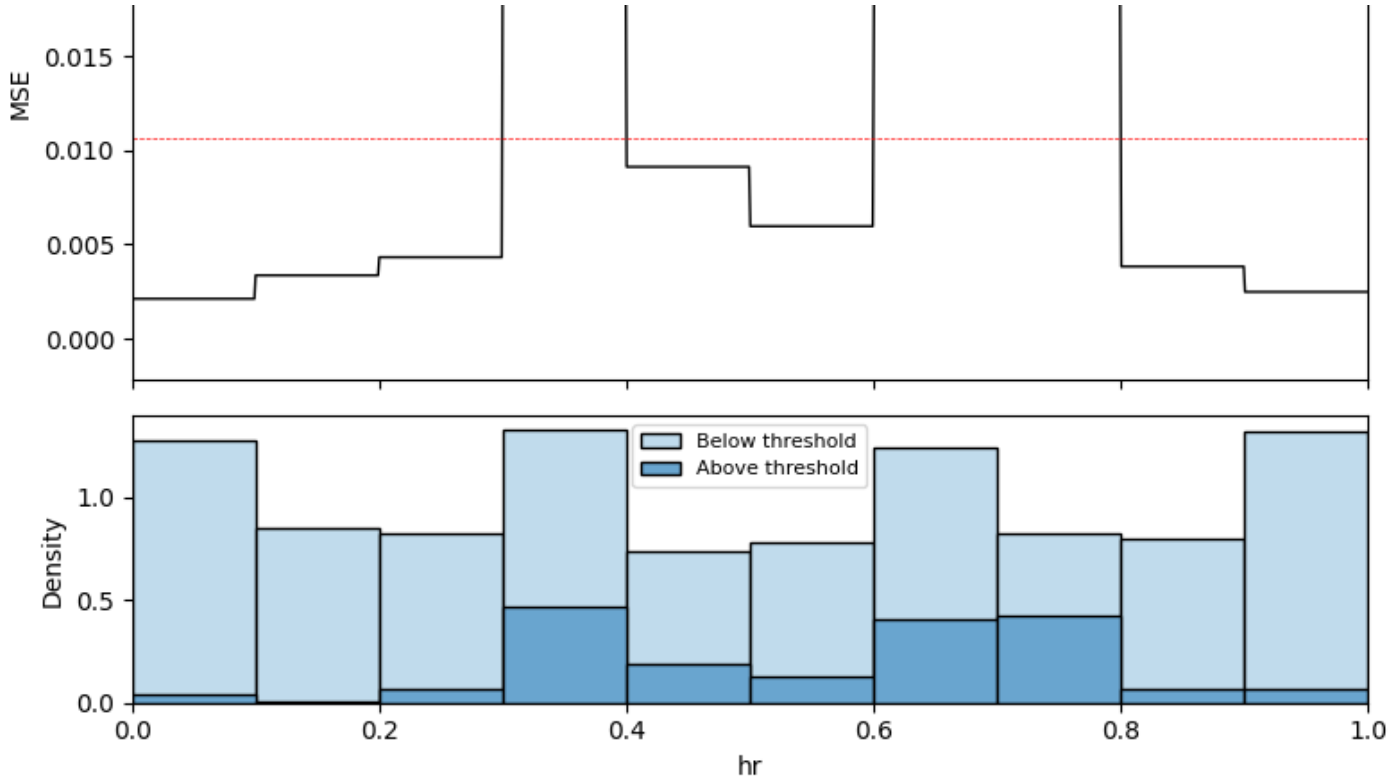
The top plot shows that both hours 3 and 4 have MSE values above the threshold. The bottom plot shows that both hours 3 and 4 are in the "above threshold" region of the histogram. This means they are considered stronger regions, not weak ones.

Q12 (5 pts): If you do the same analysis of Q10 for the DNN model, does it have the same weak regions for the `hr` feature?

In [43]:

```
exp_bike_sharing.model_diagnose(model="DNN_Bike_Sharing", show="weakspot", slice_method="
histogram", slice_features=["hr"], metric="MSE", use_test=True)
```





The DNN model does not have the exact same weak regions as the XGB model for the 'hr' feature. DNN has more pronounced errors around certain hours compared to the XGB model, suggesting that it is capturing the temporal patterns differently.

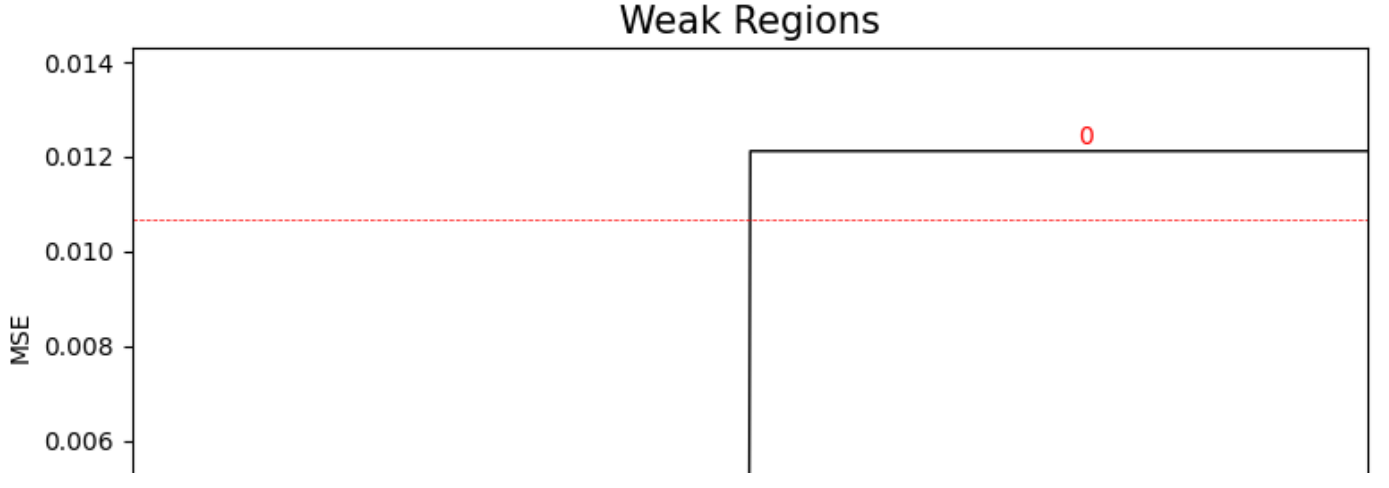
Q13 (5 pts): Do these weak regions mean that the model should not be used? What might be done to improve the model in these regions?

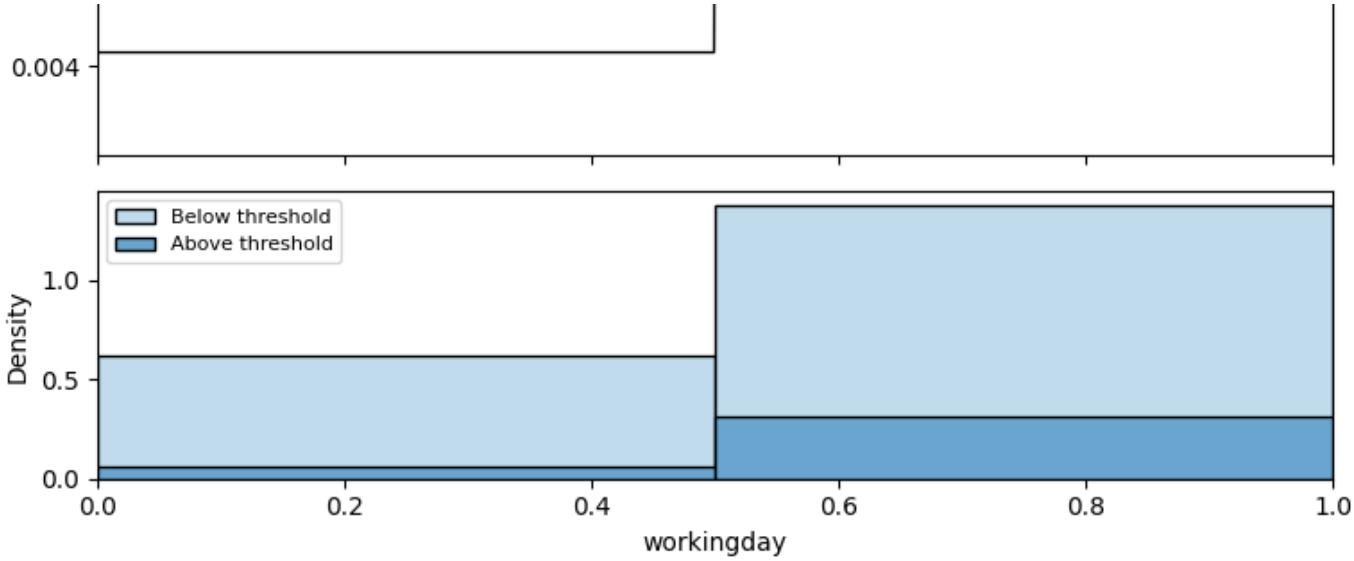
Weak regions in a model don't necessarily mean it's unusable. They indicate areas where the model is less reliable, especially for specific time periods and may not generalize well. To improve the model in these regions, consider techniques like feature engineering, categorization of time variables, adding data to weak regions, resampling, or fine-tuning model parameters for better generalization.

Q14 (10 pts): For the DNN model on the bike sharing data, analyze the weak spots for the `workingday` feature, based on testing data and MSE metric and historgram slicing. What is the number of test samples in the weak region? What is the difference from the test samples in the weak region to the MSE of the test data in the overall model?

In [44]:

```
exp_bike_sharing.model_diagnose(model="DNN_Bike_Sharing", show="weakspot", slice_method="
histogram", slice_features=["workingday"], metric="MSE", use_test=True, return_data=True
)
```





Out[44]:

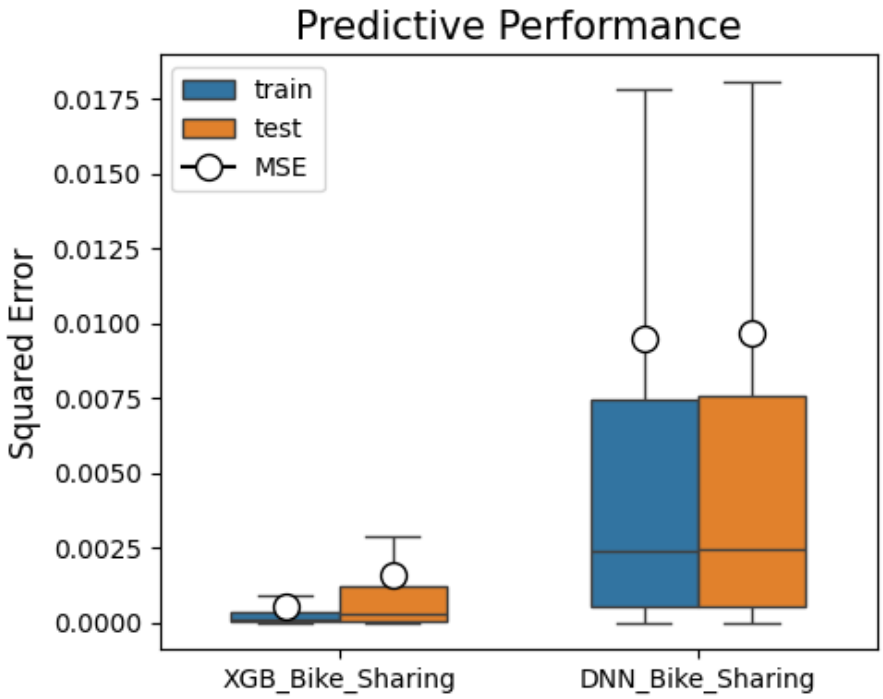
```
<piml.utils.TestResult at 0x1d4a430b3a0>
```

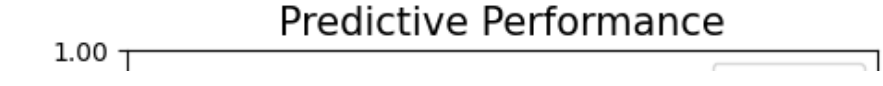
The weak region in the 'workingday' feature is for working days (value of 1). Approximately 50% of the test data falls in this weak region. The MSE in the weak region is higher (by about 0.007) than in the non-weak region (non-working days), indicating that the model performs worse on working days.

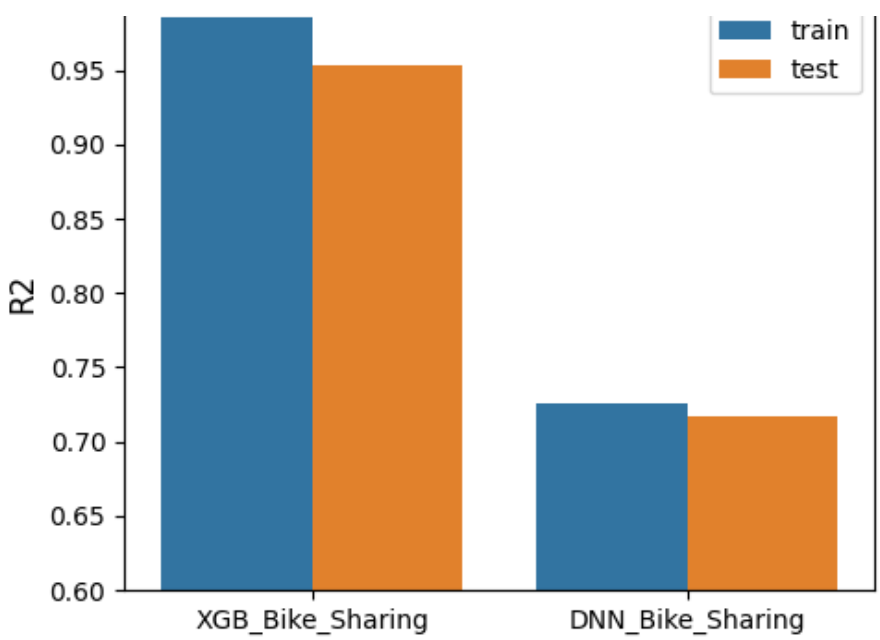
Q15 (10 pts): Compare the MSE and R2 for the two models used in biking sharing data set, for both training sample and testing. Plot the box plot for MSE and bar plot for R2. Which model performs better based on out-of-sample evaluation metrics?

In [45]:

```
# MSE
exp_bike_sharing.model_compare(models=["XGB_Bike_Sharing", "DNN_Bike_Sharing"], show="acc
uracy_plot", metric="MSE", figsize=(5, 4))

#R2
exp_bike_sharing.model_compare(models=["XGB_Bike_Sharing", "DNN_Bike_Sharing"], show="acc
uracy_plot", metric="R2", figsize=(5, 4))
```







- **Based on the box-plots and predictive performance on the bar graph, the XGB model is performing better compared to the DNN.**
- **XGB has lower mean squared error and higher R2, likely meaning that it is better at generalizing the data overall and making more accurate predictions on the unseen data.**

Taiwan (TW) credit dataset with XGB model and DNN model

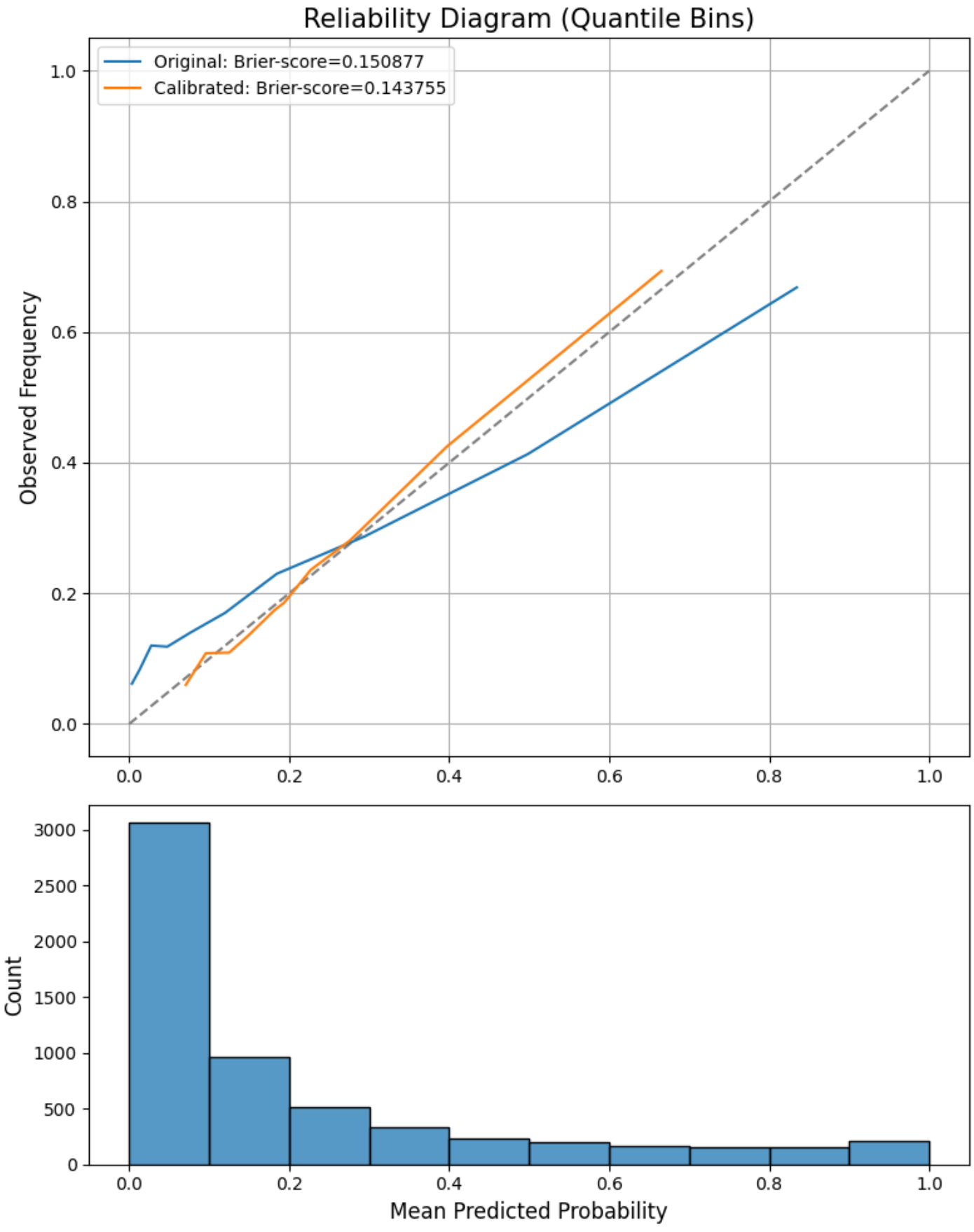
Q16 (10 pts): Different from regression models, what may need to be calibrated in a classification model? Use PiML to plot the reliability diagrams (or calibration curves) for both of the Taiwan credit data set models. Then add a copy of that plot with 100 bins. Which model is more reliable before calibration for the Taiwan credit data set?

In [47]:

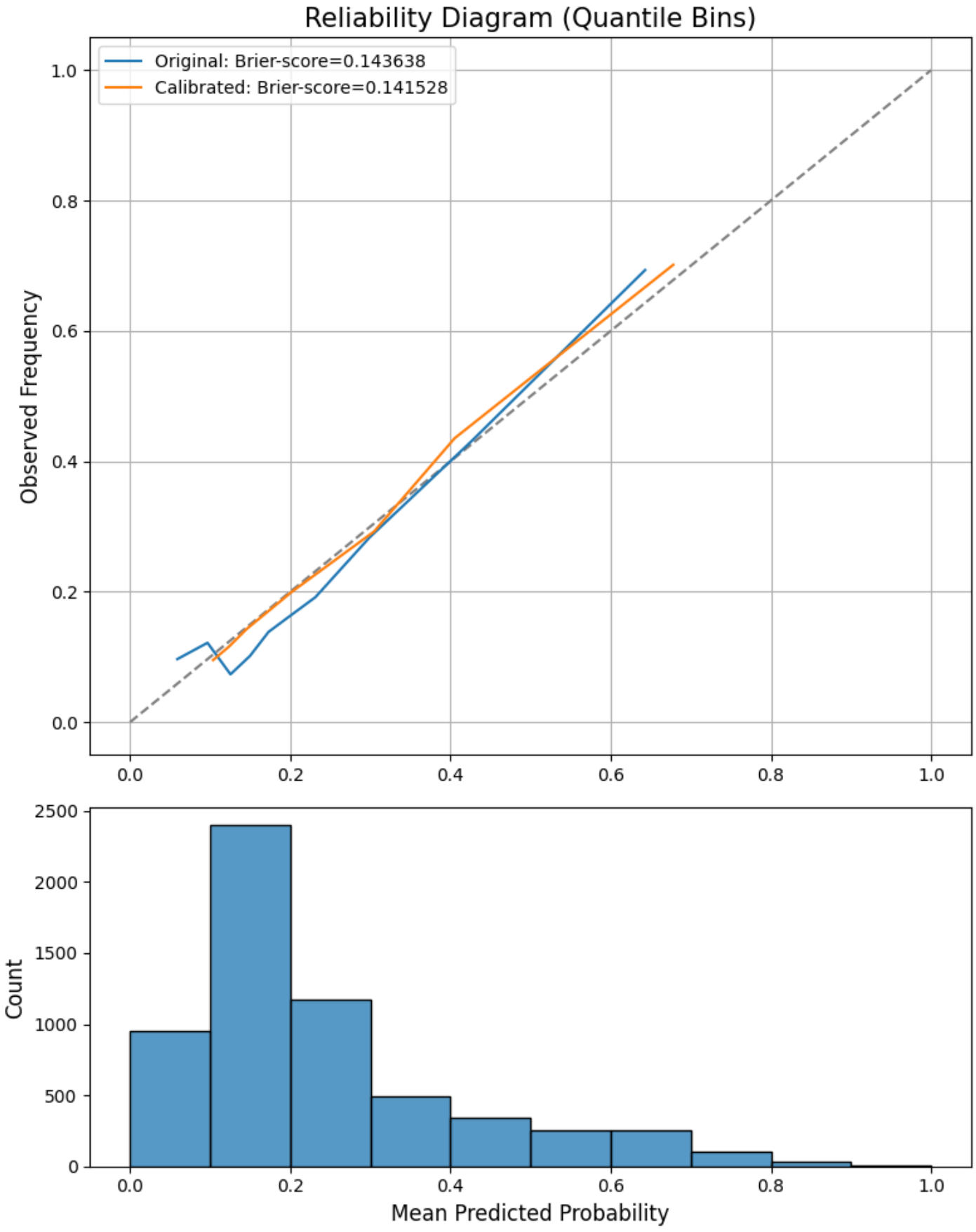
```
print("------------------------------------------------------------------------------------
-------------------------------------------------------")

print("Reliability Diagram - XGB - Taiwan Credit")
exp_tw_credit.model_diagnose(model="XGB_TW_Credit", show="reliability_perf", figsize=(8,
10))

print("------------------------------------------------------------------------------------
-------------------------------------------------------")

print("Reliability Diagram - DNN - Taiwan Credit")
exp_tw_credit.model_diagnose(model="DNN_TW_Credit", show="reliability_perf", figsize=(8,
10))

print("------------------------------------------------------------------------------------
-------------------------------------------------------")
print("------------------------------------------ Using 100 Bins ----------------------
-------------------------------------------------------")
print("------------------------------------------------------------------------------------
-------------------------------------------------------")

print("Reliability Diagram - XGB - Taiwan Credit 100 Bins")
exp_tw_credit.model_diagnose(model="XGB_TW_Credit", show="reliability_perf", bins=100, fi
gsize=(8, 10))

print("------------------------------------------------------------------------------------
-------------------------------------------------------")

print("Reliability Diagram DNN Taiwan Credit 100 Bins")
exp_tw_credit.model_diagnose(model="DNN_TW_Credit", show="reliability_perf", bins=100, fi
```

```
gsize=(8, 10))

print("-------------------------------------------------------------------------------------------
-------------------------------------------------------")
```

```
-------------------------------------------------------------------------------------------
-------------------------------------------------
Reliability Diagram - XGB - Taiwan Credit
```

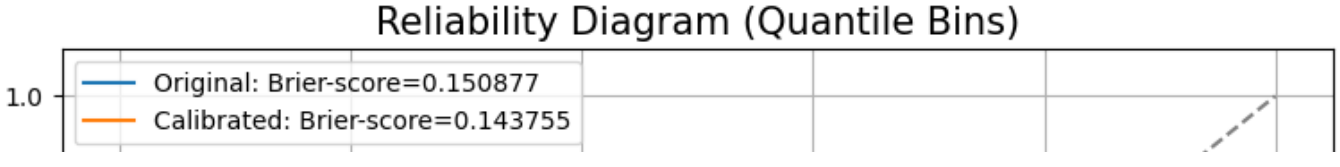


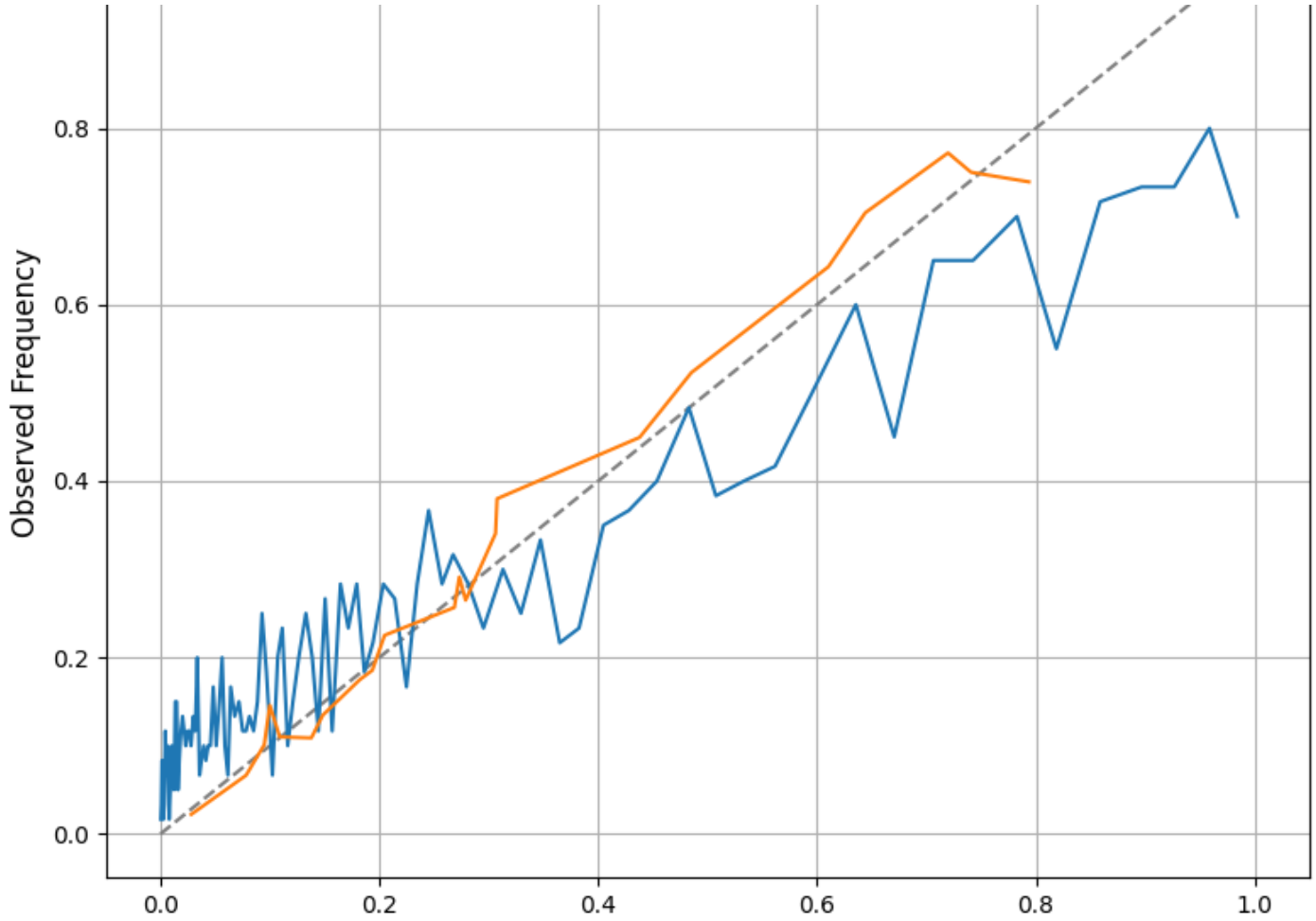

```
-------------------------------------------------------------------------------------------
-------------------------------------------------
Reliability Diagram - DNN - Taiwan Credit
```

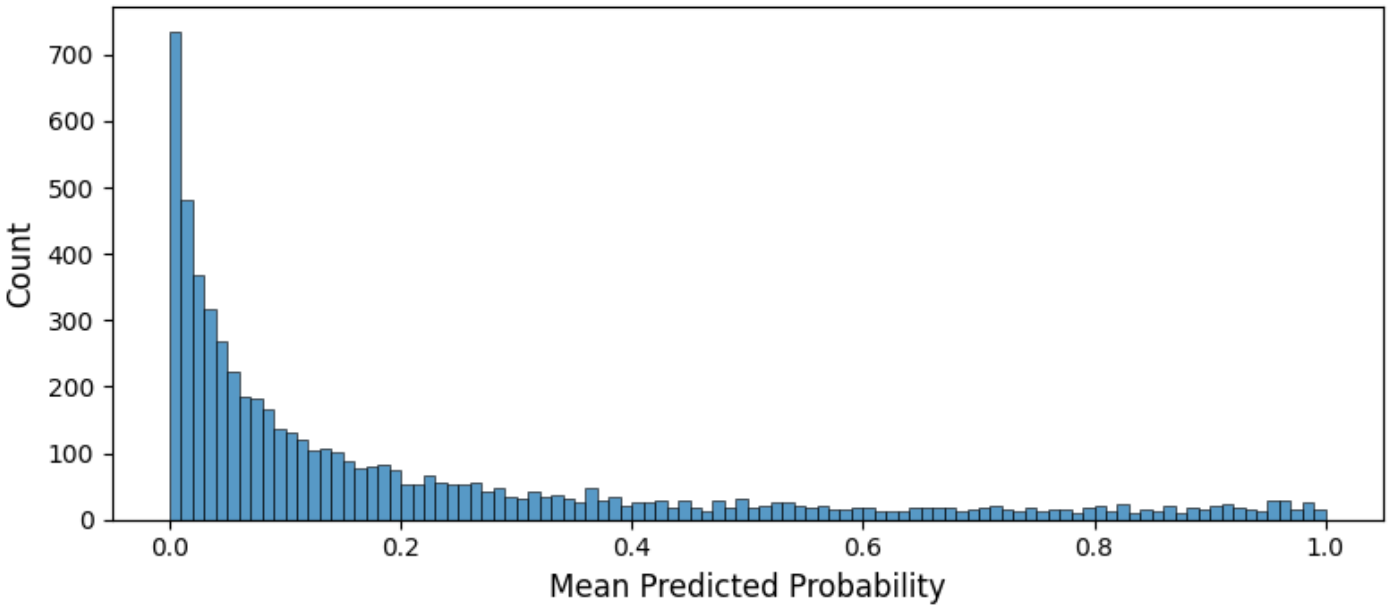


---------- Using 100 Bins ----------

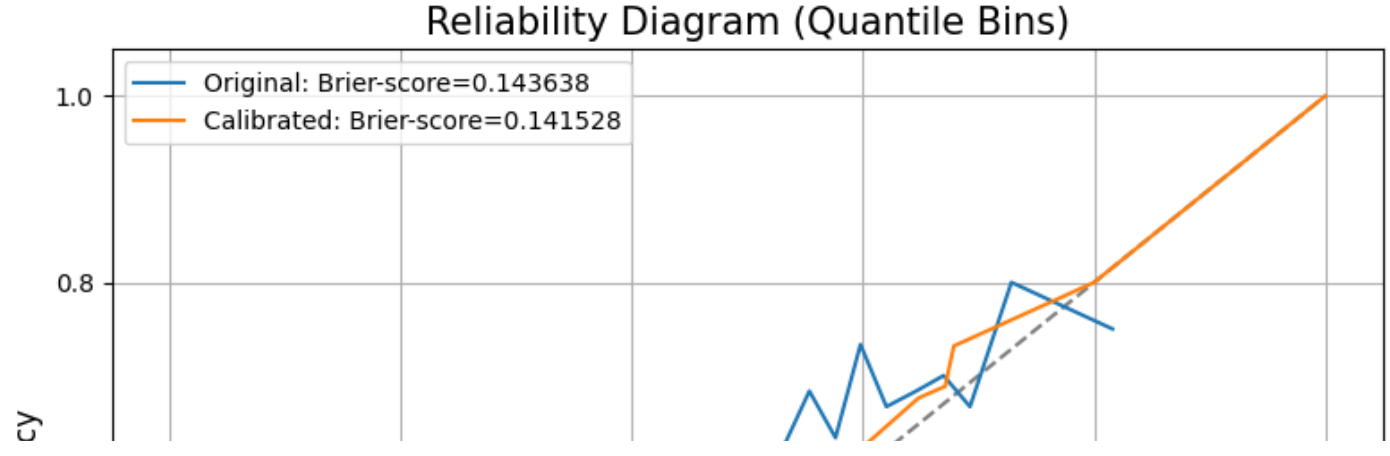
Reliability Diagram - XGB - Taiwan Credit 100 Bins

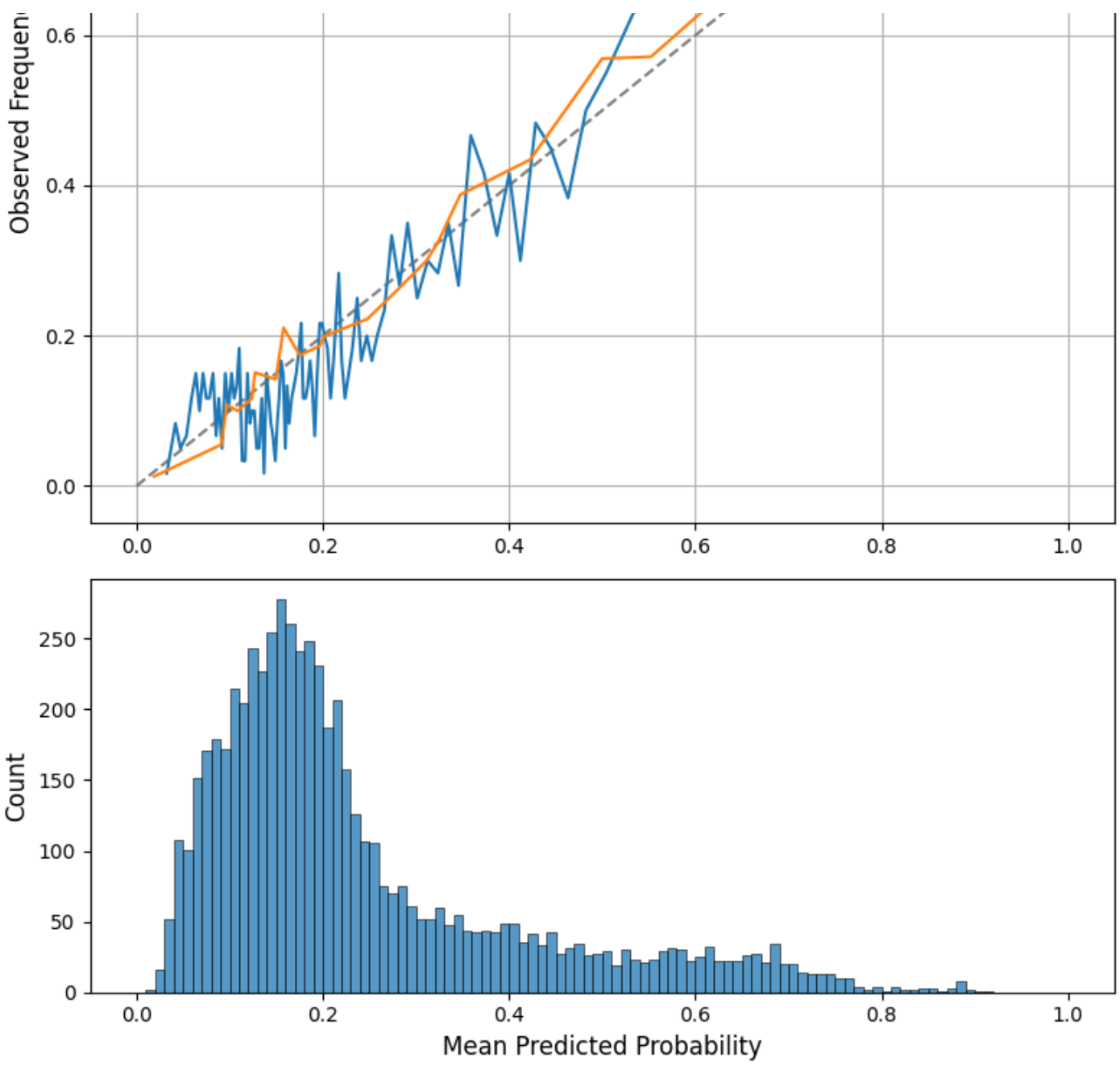






Reliability Diagram DNN Taiwan Credit 100 Bins





- **In classification models, we need to calibrate**
  - **Predicted Probabilities: To ensure they reflect the true likelihood of belonging to each class.**
  - **Decision Threshold: To account for imbalanced classes, finding a balance between minimizing false positives and false negatives.**

DNN model is more accurate before calibration.

- **The DNN's calibration curve is closer to the ideal diagonal line than the XGB model's curve. This indicates that the DNN model's predicted probabilities more closely align with the actual observed frequencies.**
- **The DNN model has a slightly better Brier score than the XGB model, further supporting the idea that its predicted probabilities are more accurate.**
- **While both models exhibit some degree of miscalibration, the DNN model appears to be more reliable in terms of its predicted probabilities before any calibration is applied.**

Q17 (5 pts): For the two models of TW credit dataset, which model performs better and why? Does any model potentially have overfitting problem and why?

- **The DNN model for the TW credit dataset performs better before calibration, as indicated by its closer calibration curve and slightly better Brier score. However, both models, especially the DNN have potential for overfitting.**

- **Overfitting is a concern with complex models like DNNs due to their high flexibility and the potential for memorizing the training data too well. The imbalanced nature of the TW credit dataset can exacerbate overfitting.**
- **To assess overfitting, compare the performance on the training set versus the test set. A significant drop in performance on the test set suggests overfitting. We can consider using regularization techniques (like dropout or other techniques) during training to mitigate this issue.**

In [48]:

```
# End of Assignment - Uma Chavali
```

In [ ]:

```

```