2-midterm

September 26, 2023

0.1 Task 2 (1)

print your student id, full and name below.

Expected answer

```
st12xxxx
Firstname Lastname
```

st123131
Nattabude Tanasansurapong

0.2 Task 3 (1)

Continue from the Task 1, show your current version of numpy, pandas, and sklearn

[ ]:
```
import numpy as np
import pandas as pd
import sklearn
import matplotlib as mpl
import matplotlib.pyplot as plt

print(f"     numpy version: {np.__version__}")
print(f"    pandas version: {pd.__version__}")
print(f"   sklearn version: {sklearn.__version__}")
print(f"matplotlib version: {mpl.__version__}")
```

```
     numpy version: 1.26.0
    pandas version: 2.1.1
   sklearn version: 1.3.1
matplotlib version: 3.8.0
```

0.3 Task 4 (1)

Print a url to your `ait-ml-2023-midterm` image from `DockerHub`.

Expected answer

`https://hub.docker.com/repository/xxxxxxxxxxx`

https://hub.docker.com/layers/pohfori/ait-ml-2023-midterm/latest/images/sha256:3b606308bbd235d0f4f9e36ca7

0.4 Task 5 (1)

So far so good?

In the folder `dataset`, there is a file `data.csv`. Load the file into a `Pandas` dataframe.

Take a look at the data. It supposes to **4** columns [y,x1,x2,x3] with 200 rows.

[ ]:
```
df = pd.read_csv('dataset/data.csv')
```

[ ]:
```
df.head()
```

[ ]:
```
    y     x1    x2  x3
0   1  20.83  6.80  40
1   1  22.05  2.26  40
2   0  14.33  2.27  30
3   1  22.80  6.96  30
4   0  10.78  0.39  40
```

[ ]:
```
df.shape
```

[ ]:
```
(200, 4)
```

[ ]:
```
df.describe()
```

[ ]:
```
                y          x1         x2          x3
count  200.000000  200.000000  200.00000  200.000000
mean     0.500000   16.048350    5.25540   25.900000
std      0.501255    5.343838    3.00317   11.261622
min      0.000000    7.680000    0.00000   10.000000
25%      0.000000   11.535000    2.70750   20.000000
50%      0.500000   15.320000    5.34000   30.000000
75%      1.000000   20.490000    7.99500   40.000000
max      1.000000   25.510000    9.96000   40.000000
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   y       200 non-null    int64
 1   x1      200 non-null    float64
 2   x2      200 non-null    float64
 3   x3      200 non-null    int64
dtypes: float64(2), int64(2)
memory usage: 6.4 KB
```

[ ]:
```
# no null data
```

0.5 Task 6 (1)

Answer the following questions

1. What is the range of each columns (min, max)?

- Answer x1 => (7.68,25.51) x2 => (0,9.96) x3 => (10,40) y => (0,1)

2. Based on the value of each columns, which one is Discrete and which one is Category?

- Answer discrete or catagoty => x3,y continuous => x1,x2

3. Based on the range of y, how many classes are there? and what are they?

- Answer 2 classes 0,1

0.6 Task 7 (1)

Plot three scatter plots. - x1, x2 and color as y - x1, x3 and color as y - x2, x3 and color as y

[ ]:
```
! pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.12.2-py3-none-any.whl (293 kB)
                           293.3/293.3
kB 2.7 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: numpy!=1.24.0,>=1.17 in
/usr/local/lib/python3.11/site-packages (from seaborn) (1.26.0)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.11/site-
packages (from seaborn) (2.1.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in
/usr/local/lib/python3.11/site-packages (from seaborn) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/site-
packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(4.42.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.11/site-
packages (from matplotlib!=3.6.1,>=3.1->seaborn) (10.0.1)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(3.1.1)
```

```
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.1->seaborn)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/site-
packages (from pandas>=0.25->seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.11/site-
packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/site-
packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Installing collected packages: seaborn
Successfully installed seaborn-0.12.2
WARNING: Running pip as the 'root' user can result in broken permissions

and conflicting behaviour with the system package manager. It is recommended to

use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

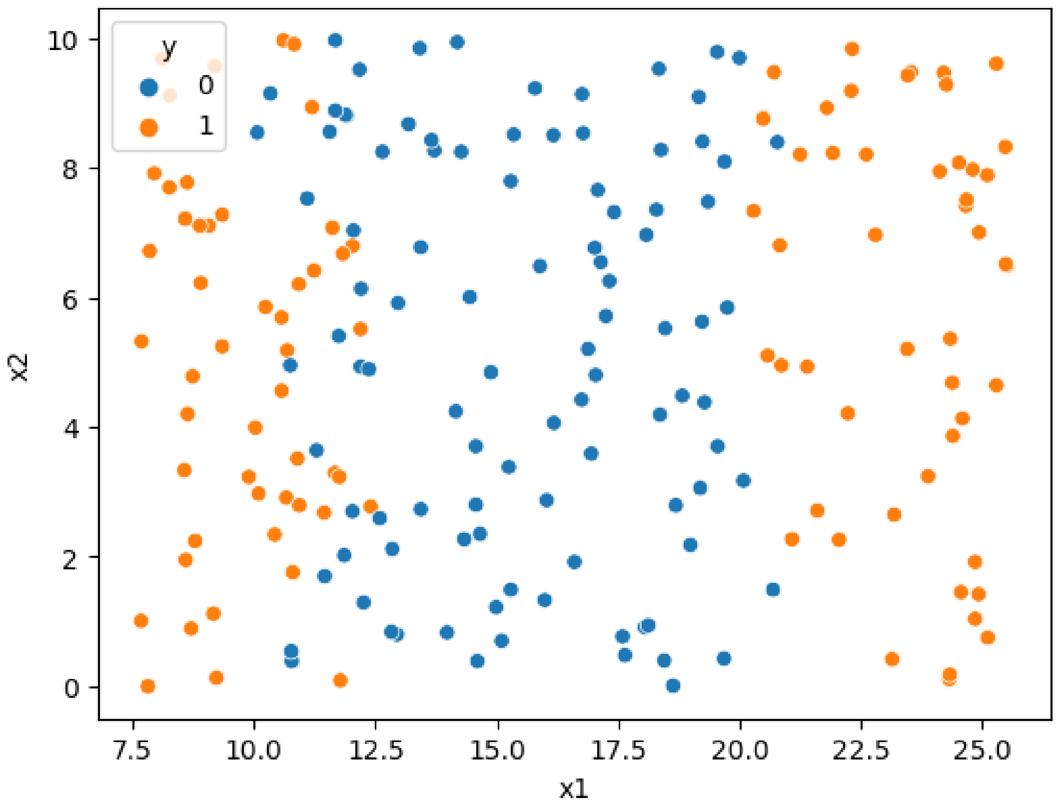
[ ]:

```
import seaborn as sns

sns.scatterplot(data=df, x='x1', y='x2',hue='y')
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

[ ]:

```
<Axes: xlabel='x1', ylabel='x2'>
```

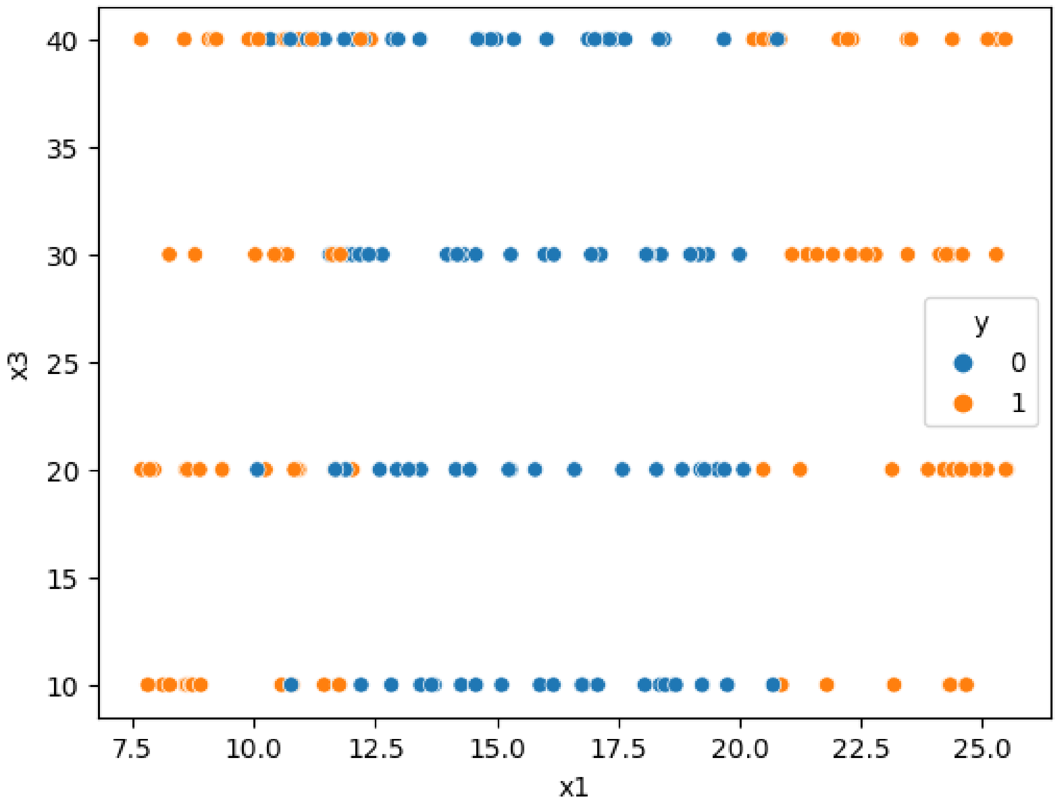



```
[ ]: sns.scatterplot(data=df, x='x1', y='x3',hue='y')
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
```

```
  if pd.api.types.is_categorical_dtype(vector):
```

[ ]: <Axes: xlabel='x1', ylabel='x3'>

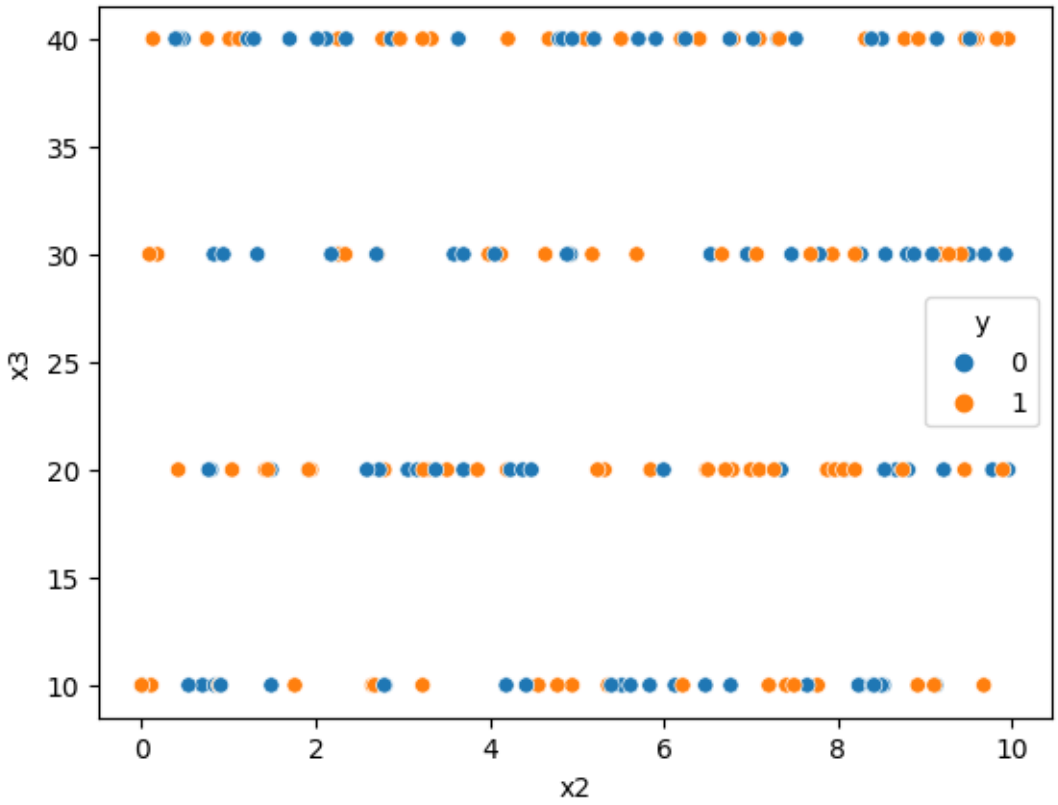


[ ]:
```
sns.scatterplot(data=df, x='x2', y='x3',hue='y')
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

[ ]: <Axes: xlabel='x2', ylabel='x3'>



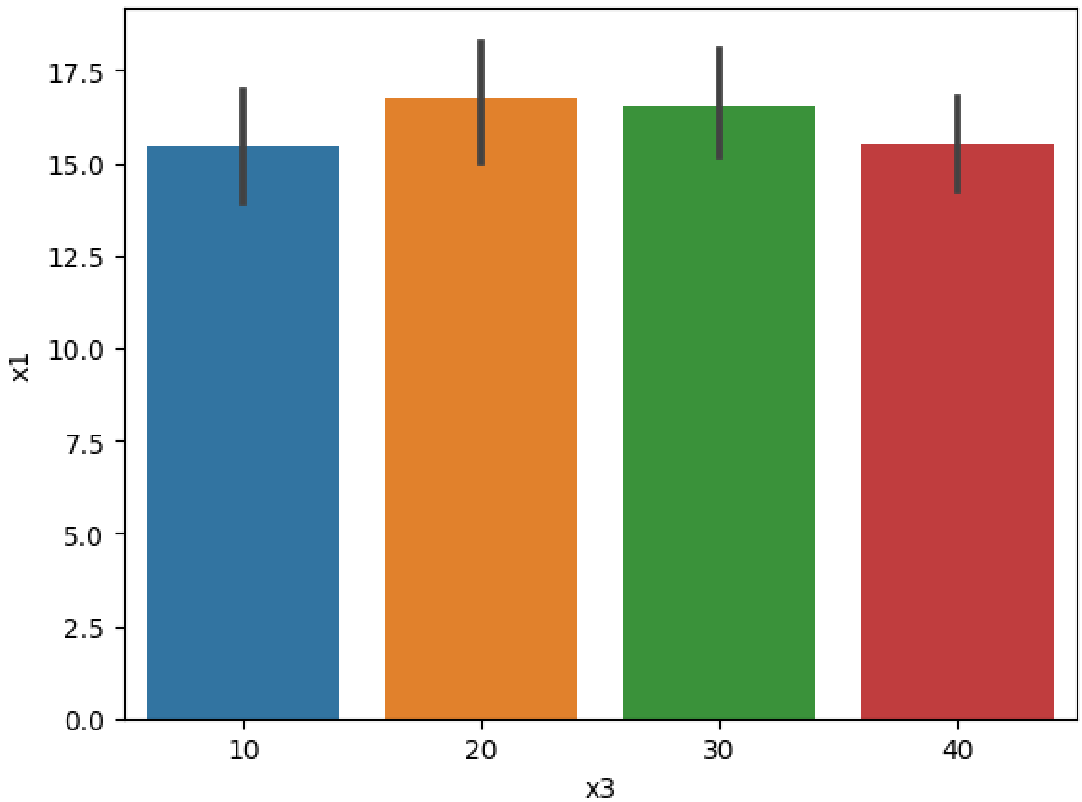
0.7 Task 8 (10)

Perform Exploratory Data Analysis (EDA) on the Dataset.

At the end, report which feature and classifier is best. (explain as best as you can. short answer get 0.)

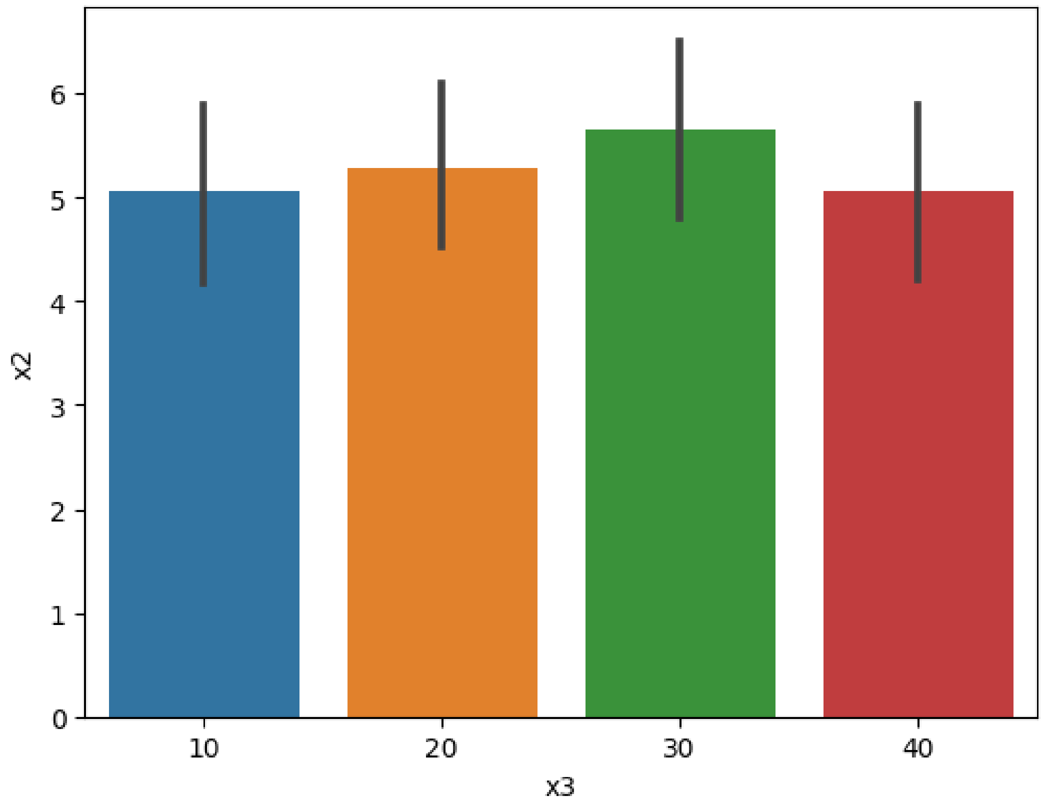
[ ]:
```
# bar plot

#2. for numerical type, lets plot some a bar plot with Loan Status


for col in ["x1","x2","x3"]:
    sns.barplot(x = df['x3'], y = df[col])
    plt.show()
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

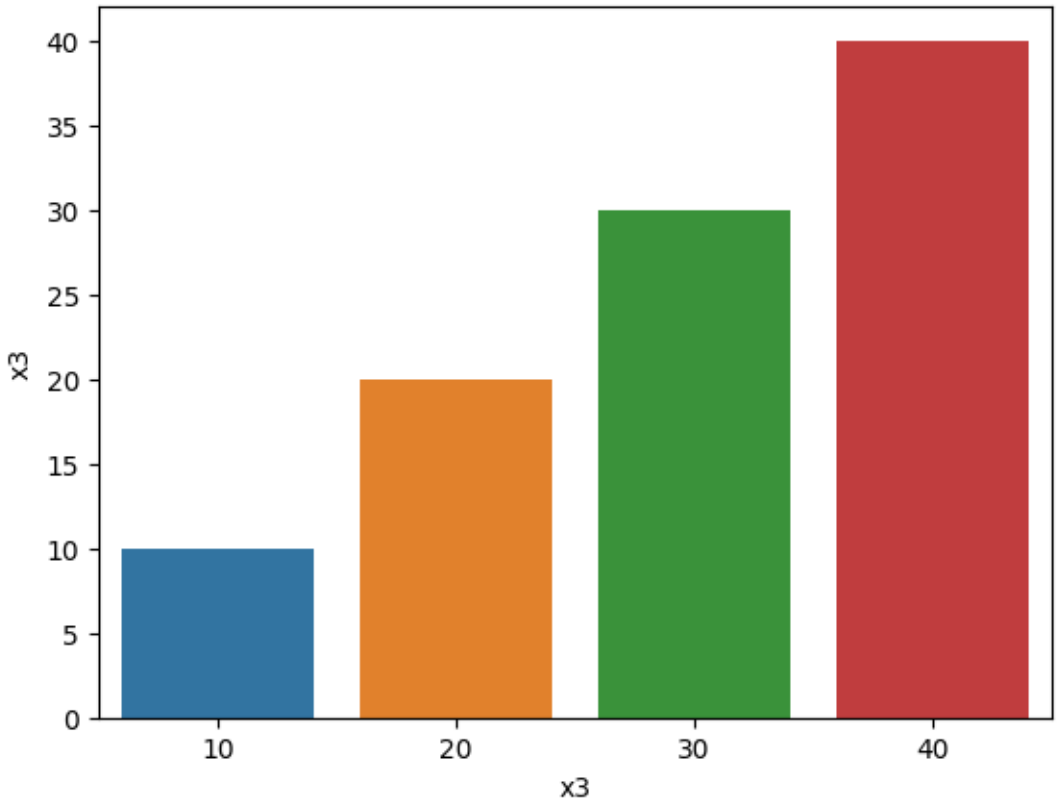



```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
```

```
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

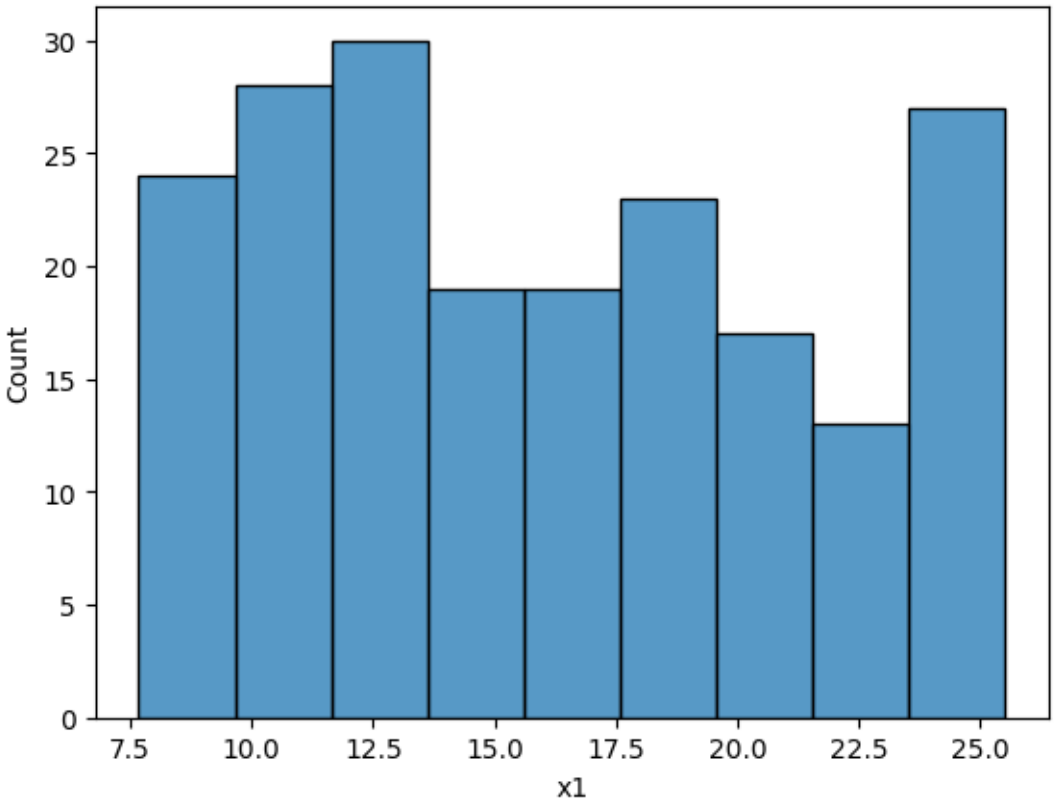



```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

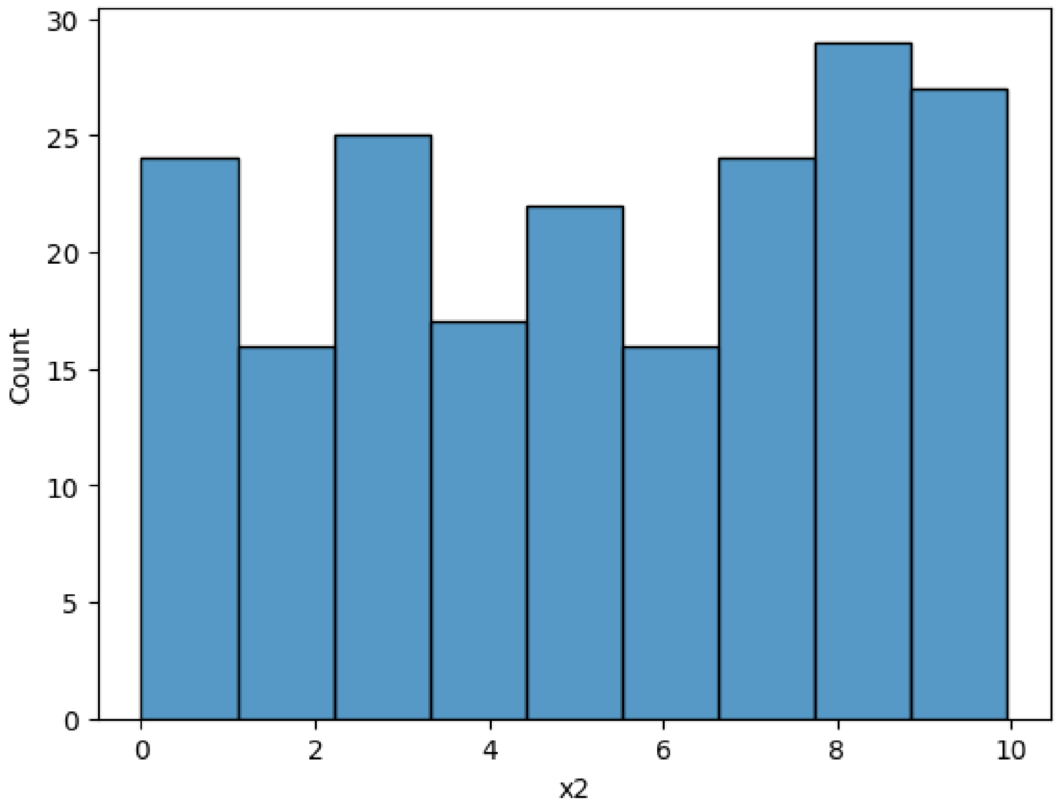


[ ]:
```
for col in ["x1","x2"]:
    sns.histplot(data=df, x=col)
    plt.show()
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

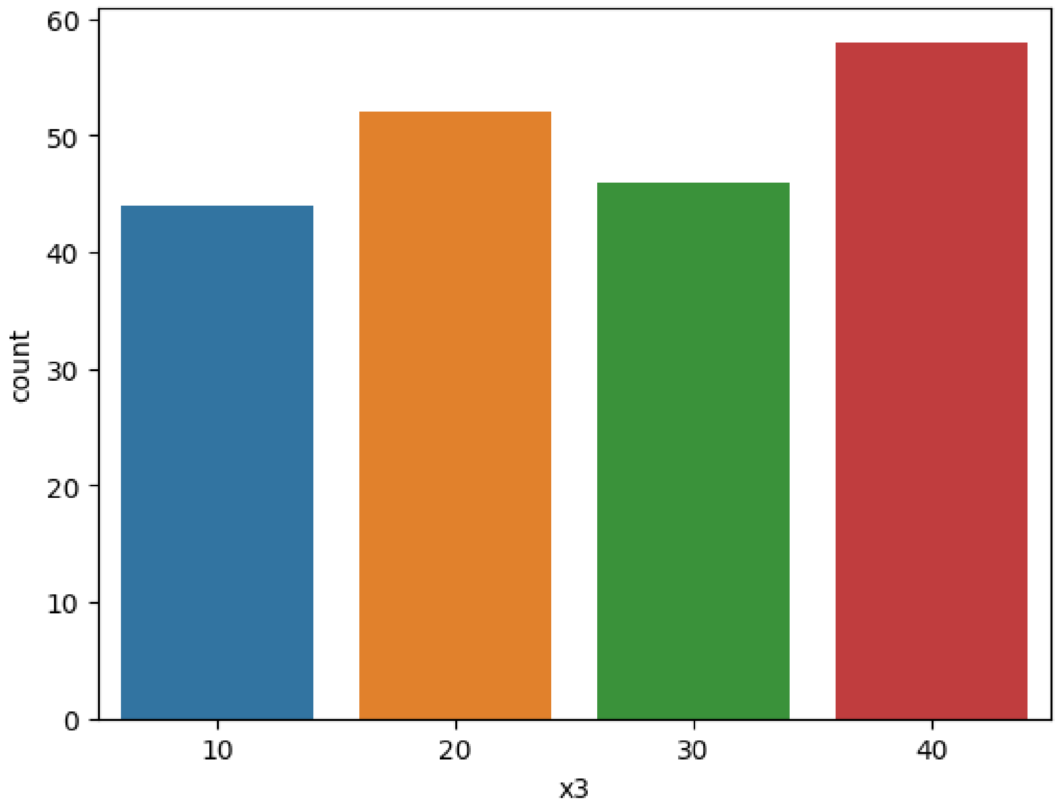



```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

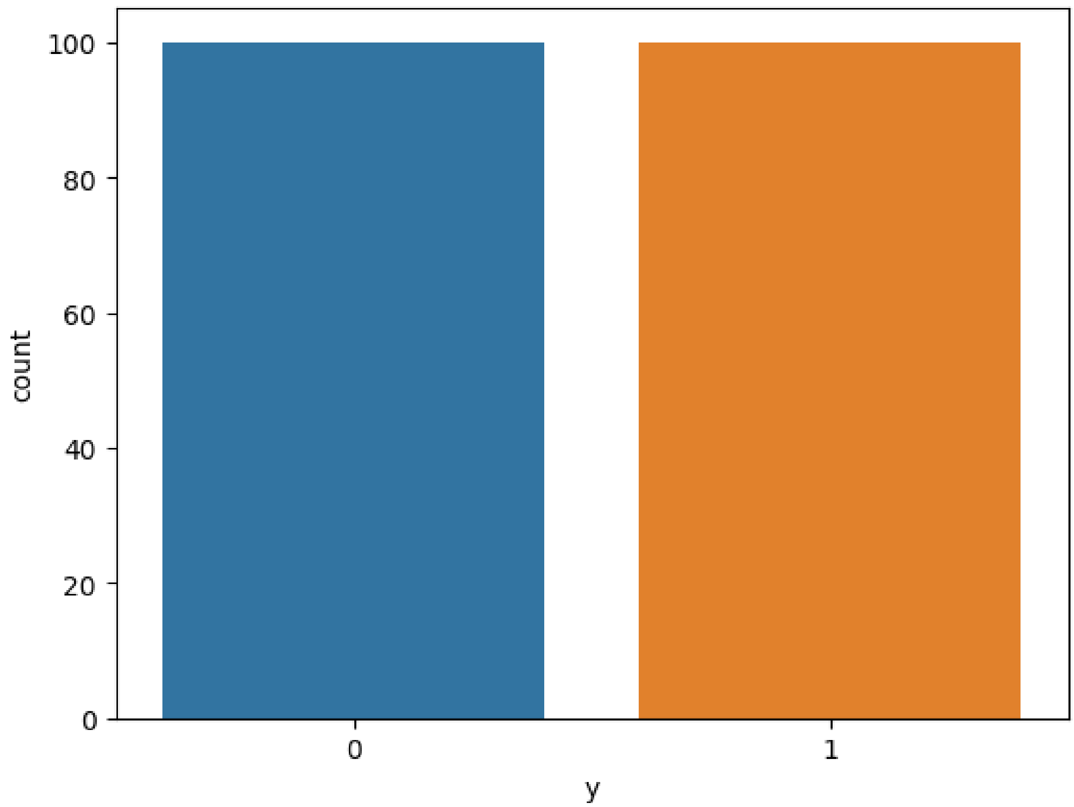


[ ]:
```
for col in ["x3",'y']:
    sns.countplot(data=df, x=col)
    plt.show()
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```




```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



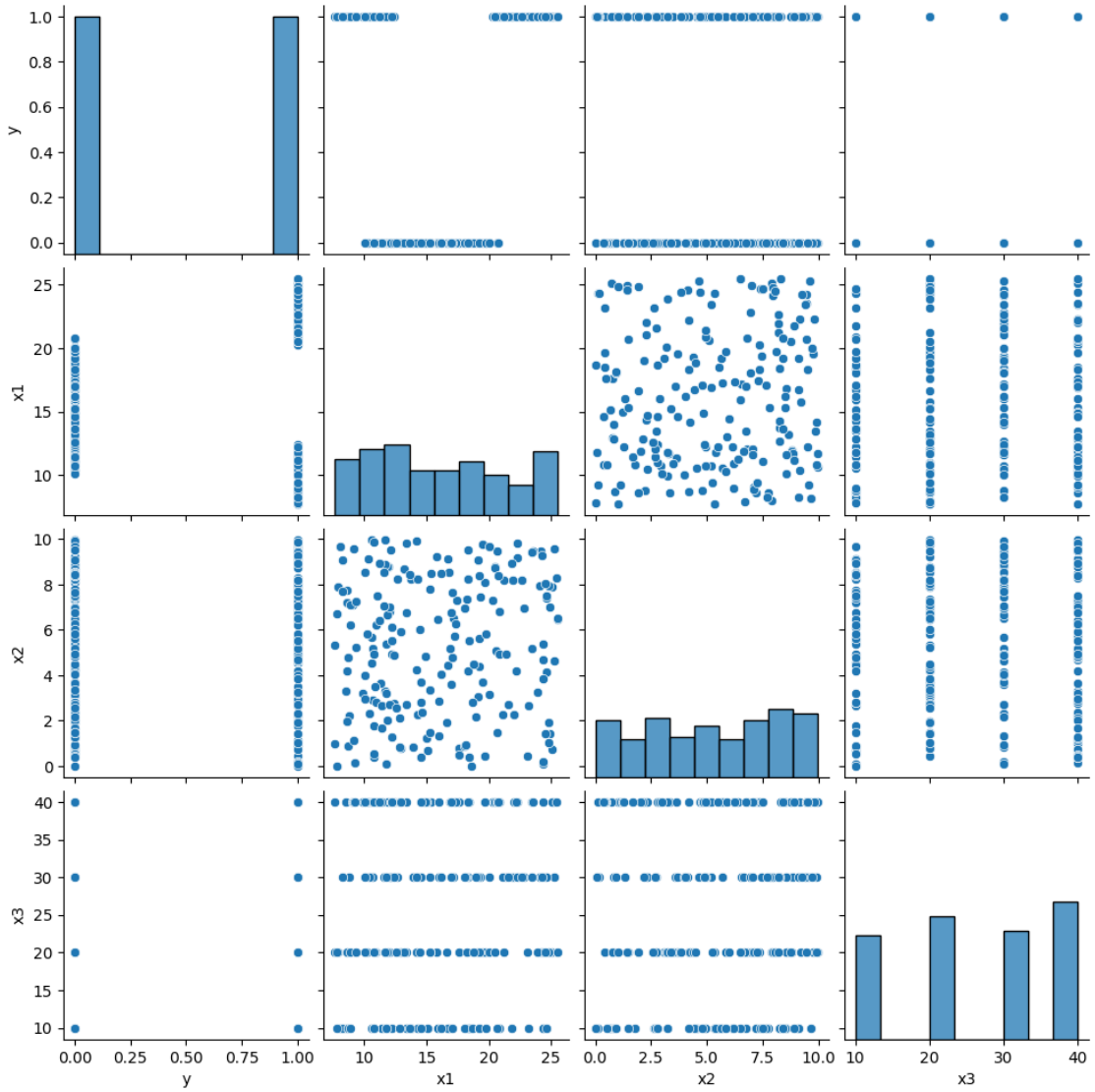
[ ]:
```
sns.pairplot(df)
plt.show()
```

```
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
```

```
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
```

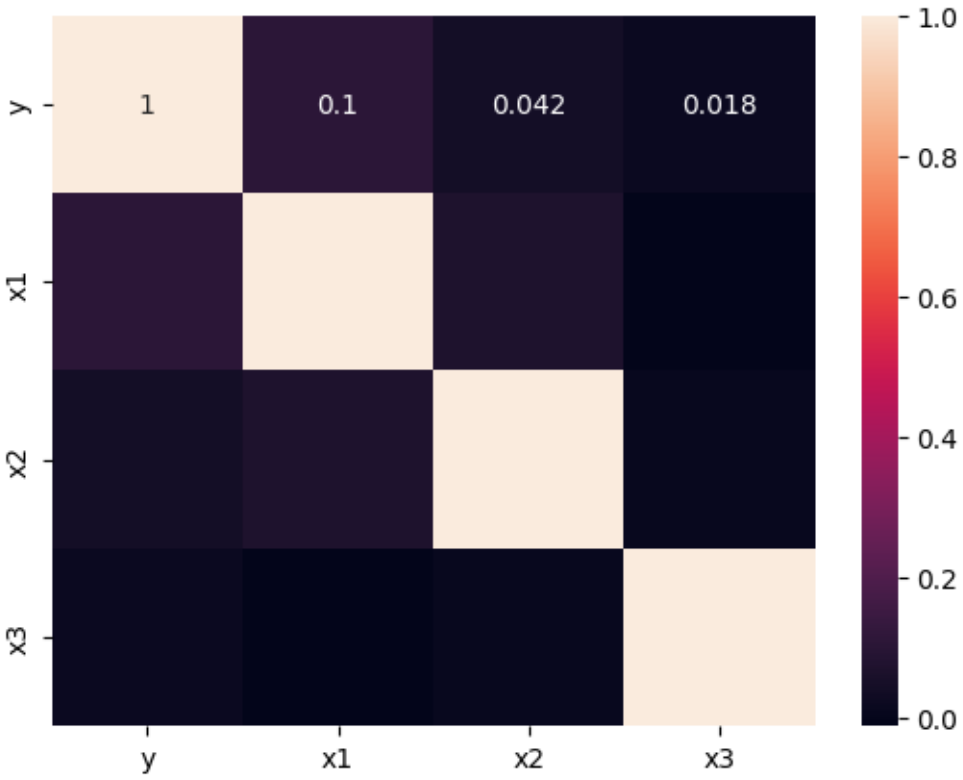
```
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
```

```
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/usr/local/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



[ ]:
```
sns.heatmap(df.corr(), annot=True)
```

[ ]: <Axes: >



Your report

- label looked balance
- distribution seem not normal
- the feature is not much correlated the most correlate is x1 = 0.1
- from scatter plot the x1 seem to can seperate the class 0 and 1 (from 10 to 20 is likly to be class 0)

0.8 Task 9 (10)

Perform Preprocessing and Data splitting (80:20).

Argue your choice. (No explanation, no score.)

Becareful with the data leakage and imbalance data when split.

[ ]:
```
# preprocess
# lets do something with x1 like abs different from 15 since x1 can seperate 0␣
↪10-20

df['x1-new'] = df['x1'].apply(lambda x:abs(x-15))
```

[ ]:
```
df[:5]
```

[ ]:

	y	x1	x2	x3	x1-new
0	1	20.83	6.80	40	5.83
1	1	22.05	2.26	40	7.05
2	0	14.33	2.27	30	0.67
3	1	22.80	6.96	30	7.80
4	0	10.78	0.39	40	4.22

[ ]:
```
# data splite
X = df[['x1-new','x2','x3']]
y = df["y"]
# use all feature

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,␣
 ↪random_state = 96, stratify=y)
# used satify to split balancly
```

[ ]:
```
# data splite
X = df[['x1','x2','x3']]
y = df["y"]
# use all feature

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,␣
 ↪random_state = 96, stratify=y)
# used satify to split balancly
```

[ ]:
```
X_train.describe()
```

[ ]:

	x1-new	x2	x3
count	160.000000	160.000000	160.000000
mean	4.655750	5.315937	26.000000
std	2.790718	2.986697	11.170491
min	0.010000	0.010000	10.000000
25%	2.545000	2.760000	20.000000
50%	4.210000	5.565000	30.000000
75%	6.402500	7.995000	40.000000
max	10.510000	9.960000	40.000000

[ ]:
```
y_train.value_counts()
```

[ ]:
```
y
0    80
1    80
Name: count, dtype: int64
```

[ ]:
```
df.isna().sum()
```

[ ]:
```
y         0
x1        0
x2        0
x3        0
x1-new    0
dtype: int64
```

[ ]:
```
# imputation
# if null is 0 maybe no need
```

[ ]:
```
# i don't think we should scale because x3 is catagory
```

0.9 Task 10 (10)

Perform model selection to find the best model that suited this dataset.

To save you from insanity, you don't need to perform `GridSearch`.

Validate the result using accuracy, precision-recall, f1-score, and confusion matrix.

Explain the result.

[ ]:
```
# Model to train. You can add more if you want.
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

[ ]:
```
# use x1
#any random_state you can use.....up to you
lr = LogisticRegression(random_state=96)
rf = RandomForestClassifier(random_state=96)
sv = SVC(random_state=96)
gnb = GaussianNB()

models = [lr, rf, sv,gnb]

#3.2 perform cross validation using KFold
from sklearn.model_selection import KFold, cross_val_score

kfold = KFold(n_splits = 5, shuffle = True, random_state=999)

for model in models:
    score = cross_val_score(model, X_train, y_train, cv=kfold,␣
 ↪scoring='accuracy')  #f1, recall, precision, accuracy
    print("Scores: ", score, "- Scores mean: ", score.mean(), "- Scores std␣
 ↪(lower better): ", score.std())  #out of 1 ; 1 means perfect accuracy
```

```
    #lr, rf, sv
```

```
Scores:  [0.5     0.375   0.375   0.46875 0.375  ] - Scores mean:  0.41875 -
Scores std (lower better):  0.054486236794258416
Scores:  [0.8125  0.90625 0.75    0.84375 0.875  ] - Scores mean:  0.8375 -
Scores std (lower better):  0.053764532919016415
Scores:  [0.78125 0.71875 0.8125  0.6875  0.71875] - Scores mean:  0.74375 -
Scores std (lower better):  0.045927932677184584
Scores:  [0.84375 0.71875 0.875   0.78125 0.78125] - Scores mean:  0.8 - Scores
std (lower better):  0.054486236794258416
```

[ ]:
```
model = rf.fit(X_train,y_train)
```

[ ]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test,model.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.86      0.95      0.90        20
           1       0.94      0.85      0.89        20

    accuracy                           0.90        40
   macro avg       0.90      0.90      0.90        40
weighted avg       0.90      0.90      0.90        40
```

[ ]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,model.predict(X_test))
```

[ ]:
```
array([[19,  1],
       [ 3, 17]])
```

[ ]:
```
# use new x1
#any random_state you can use.....up to you
lr = LogisticRegression(random_state=96)
rf = RandomForestClassifier(random_state=96)
sv = SVC(random_state=96)
gnb = GaussianNB()

models = [lr, rf, sv,gnb]

#3.2 perform cross validation using KFold
from sklearn.model_selection import KFold, cross_val_score

kfold = KFold(n_splits = 5, shuffle = True, random_state=999)

for model in models:
```

```
    score = cross_val_score(model, X_train, y_train, cv=kfold,␣
↪scoring='accuracy')  #f1, recall, precision, accuracy
    print("Scores: ", score, "- Scores mean: ", score.mean(), "- Scores std␣
↪(lower better): ", score.std())  #out of 1 ; 1 means perfect accuracy
    #lr, rf, sv
```

```
Scores:  [0.8125  0.78125 0.8125  0.84375 0.875  ] - Scores mean:  0.825 -
Scores std (lower better):  0.0318688719599549
Scores:  [0.8125  0.75    0.8125  0.78125 0.875  ] - Scores mean:  0.80625 -
Scores std (lower better):  0.0414578098794425
Scores:  [0.8125  0.8125  0.75    0.8125  0.78125] - Scores mean:  0.79375 -
Scores std (lower better):  0.024999999999999998
Scores:  [0.84375 0.75    0.84375 0.84375 0.875  ] - Scores mean:  0.83125 -
Scores std (lower better):  0.04238956239453293
```

[ ]:
```
model = gnb.fit(X_train,y_train)
from sklearn.metrics import classification_report

print(classification_report(y_test,model.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.86      0.95      0.90        20
           1       0.94      0.85      0.89        20

    accuracy                           0.90        40
   macro avg       0.90      0.90      0.90        40
weighted avg       0.90      0.90      0.90        40
```

report from best to bad random forrest, svm , naive baye and logistic regression
* since data not have much linear corelation (the most correlate is x1 0.1 which is small) logistic regression seem not so good
* the data have bad distribution that maybe why naive bayes is not so good * svm since the x1 can seperate why class 0 with 10-20 svm can find the line in higher dimension that can seperate the class * random forrest alway good because it complex

if we use new x1 (dif from 15) from best to bad naive baye,logistic regression ,random forrestand and svm
* because now it can seperate use 1 line logistic regression and naive bayes huge improve

0.10 Last Task (5)

- Export the file to PDF (4)
- Name the PDF using this format `<st12xxxx>-midterm.pdf` (1).
- Submit the PDF to the Moodle platform.