

Soal

1. [25 points] Project Environment Setup
 - a. [5 points] Buat folder project

A screenshot of a terminal window titled "trmux". The command "tree" is run in the directory "/Code/_data_portfolios/benning_mlprocess". The output shows the following directory structure:

```
benning@MacBook-Pro ~/_Code/_data_portfolios/benning_mlprocess
└── tree
    ├── data
    │   ├── interim
    │   ├── processed
    │   └── raw
    ├── models
    └── src

7 directories, 0 files
```

The terminal window has a dark background and light-colored text. The title bar says "trmux". The bottom status bar shows the session number (7), user (benning), and various tmux key bindings. The system tray at the top right shows the date (26-Jan-26) and time (02:37:31).

b. [5 points] Buat python virtual environment (venv)

```
7 directories, 0 files
└─ bening@MacBook-Pro ~-/Code/_data_portfolios/bening_mlprocess
    └─ python -m venv bening_venv
        └─ bin
            └─ python3.12 → /usr/local/opt/python@3/bin/python3.12: No module named venv
        → python3.12 -m venv bening_venv
    → bening@MacBook-Pro ~-/Code/_data_portfolios/bening_mlprocess
    → tree

bening_venv
├── bin
│   ├── activate
│   ├── activate.csh
│   ├── activate.fish
│   ├── Activate.ps1
│   ├── pip
│   ├── pip3
│   ├── pip3.12
│   ├── python → python3.12
│   ├── python3 → python3.12
│   ├── python3.12 → /usr/local/opt/python@3.12/bin/python3.12
├── include
└── lib
    └── python3.12
        └── site-packages
            ├── pip
            │   ├── __init__.py
            │   ├── _internal.py
            │   ├── _main_.py
            │   ├── _pip_runner_.py
            │   ├── _pycache_
            │   │   ├── __init___.cpython-312.pyc
            │   │   ├── __main___.cpython-312.pyc
            │   │   ├── __pip_runner___.cpython-312.pyc
            │   ├── _internal
            │   │   ├── __init__.py
            │   │   ├── _pycache_
            │   │   │   ├── __init___.cpython-312.pyc
            │   │   │   ├── build_env.cpython-312.pyc
            │   │   │   ├── cache.cpython-312.pyc
            │   │   │   ├── configuration.cpython-312.pyc
            │   │   │   ├── exceptions.cpython-312.pyc
            │   │   │   ├── main.cpython-312.pyc
            │   │   │   ├── pyproject.cpython-312.pyc
            │   │   │   ├── test_outdated_check.cpython-312.pyc
            │   │   │   └── wheel_builder.cpython-312.pyc
            │   ├── build_env.py
            │   ├── cache.py
            │   ├── cli
            │   │   ├── __init__.py
            │   │   ├── _pycache_
            │   │   │   ├── __init___.cpython-312.pyc
            │   │   │   ├── __autocompletion__.cpython-312.pyc
            │   │   └── __init__.py
            └── __pycache__
                └── __init___.cpython-312.pyc
└── titanic
    └── i_resume
        └── 2:houseprice
            └── 3:ml-utils
                └── 4:diabetest-5:pacmann*
92:37:56 [945/1984] 02:38:44 24-Jan-26 MacBook-Pro.local
```

c. [5 points] Update PIP

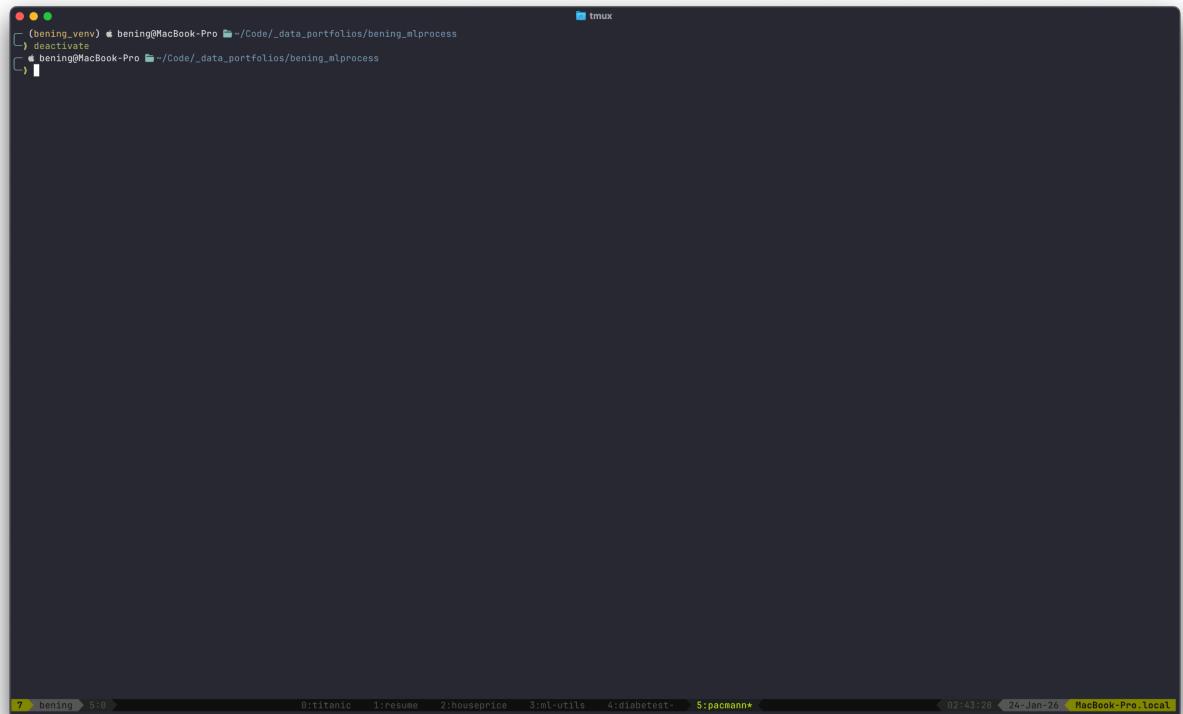
```
[bening@MacBook-Pro ~] cd ~/Code/_data_portfolios/bening_mlprocess
[bening@MacBook-Pro ~] source bening_venv/bin/activate
(bening_venv) [bening@MacBook-Pro ~] cd ~/Code/_data_portfolios/bening_mlprocess
[bening_venv] which pip
/usr/local/Code/_data_portfolios/bening_mlprocess/bening_venv/bin/pip
(bening_venv) [bening@MacBook-Pro ~] /usr/local/Code/_data_portfolios/bening_mlprocess
[bening_venv] pip install --upgrade pip
Requirement already satisfied: pip in ./bening_venv/lib/python3.12/site-packages (25.1.1)
Collecting pip
  Using cached pip-25.3-py3-none-any.whl.metadata (4.7 kB)
Using cached pip-25.3-py3-none-any.whl (1.8 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 25.1.1
    Uninstalling pip-25.1.1:
      Successfully uninstalled pip-25.1.1
Successfully installed pip-25.3
(bening_venv) [bening@MacBook-Pro ~] cd ~/Code/_data_portfolios/bening_mlprocess
[bening_venv]
```

d. [5 points] Install dependencies



```
Found existing installation: pip 25.1.1
Uninstalling pip-25.1.1...
Successfully uninstalled pip-25.1.1
Successfully installed pip-25.3
(bening_mlprocess) ➜ pip install pandas scikit-learn imblearn joblib numpy scipy seaborn fastapi jupyterlab requests
Collecting pandas
  Downloading pandas-1.0.0-cp312-cp312-macosx_10_13_x86_64.whl.metadata (79 kB)
Collecting scikit-learn
  Using cached scikit-learn-1.0.0-cp312-cp312-macosx_10_13_x86_64.whl.metadata (11 kB)
Collecting joblib
  Using cached joblib-0.0-py2.py3-none-any.whl.metadata (355 bytes)
Collecting joblib
  Using cached joblib-1.5.3-py3-none-any.whl.metadata (5.5 kB)
Collecting numpy
  Using cached numpy-1.17.0-cp312-cp312-macosx_10_13_x86_64.whl.metadata (62 kB)
Collecting scipy
  Using cached scipy-1.17.0-cp312-cp312-macosx_10_13_x86_64.whl.metadata (62 kB)
Collecting seaborn
  Using cached seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Collecting fastapi
  Using cached fastapi-0.128.0-py3-none-any.whl.metadata (30 kB)
Collecting jupyterlab
  Downloading jupyterlab-4.5.3-py3-none-any.whl.metadata (16 kB)
Collecting requests
  Using cached requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting python-datetimeutil>=2.9.2 (from pandas)
  Using cached python-datetimeutil-2.9.2 (from pandas)
Collecting threadpoolctl>=3.2.0 (from scikit-learn)
  Using cached threadpoolctl-3.2.0-py3-none-any.whl.metadata (13 kB)
Collecting imbalanced-learn (from imblearn)
  Downloading imbalanced-learn-0.14.1-py3-none-any.whl.metadata (8.9 kB)
Collecting metaplotlib<0.10.0
  Using cached metaplotlib-0.10.0-cp312-cp312-macosx_10_13_x86_64.whl.metadata (52 kB)
Collecting starlette<0.51.0,>=0.40.0 (from fastapi)
  Using cached starlette-0.50.0-py3-none-any.whl.metadata (6.3 kB)
Collecting pydantic>=2.7.0 (from fastapi)
  Using cached pydantic-2.12.5-py3-none-any.whl.metadata (90 kB)
Collecting typing-extensions>4.8.0 (from fastapi)
  Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting aiofiles>2.0.0 (from fastapi)
  Using cached aiofiles-2.0.0-py3-none-any.whl.metadata (6.6 kB)
Collecting anyio<3.6.2 (from starlette<0.51.0,>=0.40.0->fastapi)
  Using cached anyio-3.1.2-py3-none-any.whl.metadata (4.3 kB)
Collecting idna>=2.8 (from anyio<3.6.2->starlette<0.51.0,>=0.40.0->fastapi)
  Using cached idna-3.11-py3-none-any.whl.metadata (8.4 kB)
Collecting httpx<0.28.0 (from jupyterlab)
  Using cached httpx-0.28.0-py3-none-any.whl.metadata (5.3 kB)
Collecting httpx<0.28.0 (from jupyterlab)
  Using cached httpx-0.28.0-py3-none-any.whl.metadata (7.1 kB)
Collecting ipykernel>=6.30.0,>=5.5.0 (from jupyterlab)
  Using cached ipykernel-7.1.0-py3-none-any.whl.metadata (4.5 kB)
Collecting jinja2>=3.0.3 (from jupyterlab)
  Using cached jinja2-3.1.0-py3-none-any.whl.metadata (2.9 kB)
Collecting jupyter-core (from jupyterlab)
```

e. [5 points] Non aktifkan venv Anda



```
(bening_venv) ➜ bening@MacBook-Pro ~/_Code/_data_portfolios/bening_mlprocess
└─ deactivate
└─ bening@MacBook-Pro ~/_Code/_data_portfolios/bening_mlprocess
```

2. [25 points] Rangkum permasalahan bisnis

- Rangkum [mockup interview](#) antara user dan DS di bagian akhir dari dokumen ini
- **Rangkuman Anda harus bisa menjawab pertanyaan berikut:**

PERTANYAAN	JAWABAN
Latar belakang permasalahan bisnis	<i>Bank memiliki masalah dengan sistem risiko kredit yang saat ini masih menerima pinjaman-pinjaman yang berpotensi sebagai masalah.</i>
Objektif bisnis yang ingin dicapai	<i>Menurunkan angka nasabah-nasabah yang bermasalah</i>
Metrik pengukuran bisnis yang akan digunakan	<i>Total Non-Performing Loan</i>
Kandidat solusi machine learning yang akan dibangun	<i>Kasus klasifikasi yang berfokus pada ensemble models (Random Forest, XGBoost, Catboost) dengan menggunakan linear models sebagai baseline.</i>
Metrik pengukuran machine learning yang akan digunakan	<i>Recall</i>

3. [25 points] Persiapan Dataset

```
(.venv) $ bening@MacBook-Pro ~/Code/_data_portfolios/credit-risk-classification
$ cd ..
$ tree
.
├── data
│   ├── interim
│   ├── processed
│   └── raw
│       └── credit_risk_dataset.csv
└── data_preparation.ipynb
└── models
└── src

7 directories, 2 files
(.venv) $ bening@MacBook-Pro ~/Code/_data_portfolios/credit-risk-classification
$
```

The screenshot shows a Jupyter Notebook interface with the following details:

- File Structure:** The left sidebar shows a tree view of the project structure, including a `data` directory with `interim`, `processed`, and `raw` sub-directories, and a `data_preparation.ipynb` notebook.
- Kernel:** Python 3 (ipykernel)
- Cell 5:** Displays the code for loading a CSV file into a pandas DataFrame:

```
import pandas as pd

def load_data(fname: str) -> pd.DataFrame:
    """
    Load a CSV file into a pandas DataFrame and display its dimensions.

    :param fname: The file path or buffer of the CSV file to be read.
    :type fname: str
    :return: A DataFrame containing the loaded data.
    :rtype: pandas.DataFrame
    """
    data = pd.read_csv(fname)
    print(f'Data Shape: [{data.shape}]')
    return data
```
- Cell 6:** Displays the output of the previous cell, showing the first few rows of the DataFrame:

```
[5]: 0.0s
[6]: Data Shape: [(32581, 12)]
[6]:
```

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_default_on
0	22	59000	RENT	123.0	PERSONAL	D	35000	16.02	1	0.59	
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0	0.10	
2	25	9600	MORTGAGE	1.0	MEDICAL	C	5500	12.87	1	0.57	
3	23	65500	RENT	4.0	MEDICAL	C	35000	15.23	1	0.53	
4	24	64400	RENT	6.0	MEDICAL	C	35000	14.27	1	0.55	
- Bottom Bar:** Includes buttons for Run Testcases, Live Share, Select Postgres Server, Cell 6 of 6, Go Live, and Prettier.

credit-risk-classification

Split Input/Output Dataset

```

def split_input_output(
    data: pd.DataFrame,
    target_col="loan_status"
) -> tuple[pd.DataFrame, pd.Series]:
    """
    Split a DataFrame into features (X) and target (y).

    :param data: Input DataFrame.
    :type data: pd.DataFrame
    :param target_col: Target column name, defaults to "loan_status".
    :type target_col: str, optional
    :return: Feature set (X) and target series (y).
    :rtype: tuple[pd.DataFrame, pd.Series]
    """
    X = data.drop(target_col, axis=1)
    y = data[target_col]
    print(f"Original data shape: {data.shape}")
    print(f"X data shape: {X.shape}")
    print(f"y data shape: {y.shape}")
    return X, y

```

[13] ✓ 0.0s

```

X, y = split_input_output(data=data)
X.head()

```

[14] ✓ 0.0s

... Original data shape: (32581, 12)
X data shape: (32581, 11)
y data shape: (32581,)

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_percent_income	cb_person_default_on_file	cb_per
0	22	59000	RENT	12.0	PERSONAL	B	35000	16.02	0.59	Y	
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0.10	N	
2	25	9600	MORTGAGE	1.0	MEDICAL	C	5500	12.87	0.57	N	
3	23	65500	RENT	4.0	MEDICAL	C	35000	16.23	0.53	N	
4	24	54400	RENT	8.0	MEDICAL	C	36000	14.27	0.55	Y	

Python

Spaces: 4 Cell 8 of 10 Go Live Prettier

credit-risk-classification

Split Train and Test Dataset

```

from sklearn.model_selection import train_test_split

def split_train_test(
    X: pd.DataFrame,
    y: pd.Series,
    test_size: float,
    random_state: int | None=None
) -> tuple[pd.DataFrame, pd.DataFrame, pd.Series, pd.Series]:
    """
    Split the dataset into X_train, X_test, y_train, y_test.

    :param X: A feature dataset.
    :type X: pd.DataFrame
    :param y: A target dataset.
    :type y: pd.Series
    :param test_size: Represents the number of test samples.
    :type test_size: float
    :param random_state: Controls the shuffling applied to the data, defaults to None.
    :type random_state: int, optional
    :return: X_train, X_test, y_train, y_test. In that order.
    :rtype: tuple[pd.DataFrame, pd.DataFrame, pd.DataFrame, pd.DataFrame]
    """
    X_train, X_test, y_train, y_test = train_test_split(
        X,
        y,
        test_size=test_size,
        stratify=y,
        random_state=random_state
    )
    print(f"X train shape: {X_train.shape}")
    print(f"X test shape: {X_test.shape}")
    print(f"y train shape: {y_train.shape}")
    print(f"y test shape: {y_test.shape}\n")
    return X_train, X_test, y_train, y_test

```

[22] ✓ 0.0s

```

X_train, X_non_train, y_train, y_not_train = split_train_test(X, y, 0.2, random_state=42)
X_valid, X_test, y_valid, y_test = split_train_test(X_non_train, y_not_train, 0.5, random_state=42)

```

[23] ✓ 0.0s

... X train shape: (26064, 11)
X test shape: (6517, 11)
y test shape: (26064,)
y test shape: (6517,)
X train shape: (3258, 11)
X test shape: (3259, 11)
y test shape: (3258,)
y test shape: (3259,)

Python

Spaces: 4 Cell 12 of 13 Go Live Prettier

The screenshot shows a Jupyter Notebook interface with the following details:

- EXPLORER:** Shows a file tree for a project named "CREDIT-RISK-CLASSIFICATION". The "data" directory contains "interim", "processed", and "raw" sub-directories, each with various files like CSVs and pickles. A "models" directory is also present.
- CELLS:** There are two code cells open:
 - Cell 24:** Titled "Serialize Data", containing Python code to serialize data frames and series into files using joblib. It includes a docstring and several calls to `serialize_data` with different parameters (X_train, y_train, X_test, y_test, X_valid, y_valid).
 - Cell 25:** An empty cell with the status "0.0s".
- STATUS BAR:** Shows "Ln 1, Col 90" and "Cell 16 of 16".

4. [25 points] Buat python utility script

Initialization

```
import pandas as pd
import src.utils as utils

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', None)
```

Data Preparation

```
FNAME = './data/raw/credit_risk_dataset.csv'
data = utils.load_data(fname=FNAME)
data.head()
```

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_default_on_file
0	22	59000	RENT	123.0	PERSONAL	D	35000	16.02	1	0.59	
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0	0.10	
2	25	9600	MORTGAGE	1.0	MEDICAL	C	6500	12.87	1	0.57	
3	23	65500	RENT	4.0	MEDICAL	C	35000	15.23	1	0.53	
4	24	54400	RENT	8.0	MEDICAL	C	35000	14.27	1	0.55	

Split Feature and Target Dataset

```
TARGET_COL = "loan_status"
X, y = utils.split_feature_target(data=data, target_col=TARGET_COL)
X.head()
```

Original data shape: (32581, 12)
X data shape: (32581, 11)
y data shape: (32581,)

Split Feature and Target Dataset

```
TARGET_COL = "loan_status"
X, y = utils.split_feature_target(data=data, target_col=TARGET_COL)
X.head()
```

Original data shape: (32581, 12)
X data shape: (32581, 11)
y data shape: (32581,)

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_percent_income	cb_person_default_on_file	cb_per...
0	22	59000	RENT	123.0	PERSONAL	D	35000	16.02	0.59	Y	
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0.10	N	
2	25	9600	MORTGAGE	1.0	MEDICAL	C	6500	12.87	0.57	N	
3	23	65500	RENT	4.0	MEDICAL	C	35000	15.23	0.53	N	
4	24	54400	RENT	8.0	MEDICAL	C	35000	14.27	0.55	Y	

Split Train and Test Dataset

```
X_train, X_non_train, y_train, y_not_train = utils.split_train_test(X, y, 0.2, random_state=42)
X_valid, X_test, y_valid, y_test = utils.split_train_test(X_non_train, y_not_train, 0.5, random_state=42)
```

X train shape: (26064, 11)
X test shape: (6517, 11)
y train shape: (26064,)
y test shape: (6517,)

X train shape: (3258, 11)
X test shape: (3259, 11)
y train shape: (3258,)
y test shape: (3259,)

Serialize Data

```
DATA_PATH = "./data/raw"
```

Mockup Percakapan

Pengguna	Hai, Pak David! Terima kasih sudah menyempatkan waktu untuk bertemu.
David	Hai! Tentu saja, tidak masalah. Saya senang bisa membantu. Ada yang bisa saya bantu?
Pengguna	Jadi begini, Pak David, kami di bank sedang memiliki masalah dengan risiko kredit, nih. Kami ingin bikin sistem prediksi yang bisa bantu kita lebih cepet nangkep pinjaman yang berpotensi jadi masalah, gitu.
David	Ah, mengerti. Jadi tujuannya adalah untuk lebih akurat dalam memprediksi pinjaman bermasalah. Keren, tujuan bisnisnya apa nih?
Pengguna	Ya, tentu aja. Kami pengen turunin jumlah NPL (Non-Performing Loan) dan lebih cepet deteksi pinjaman yang berisiko biar bisa langkah preventif lebih awal.
David	Oke, paham. Jadi pengukuran keberhasilannya nanti gimana?
Pengguna	Gampang, nih. Yang jelas pengen liat jumlah NPL yang beneran turun.
David	Nah, kalo soal solusi machine learning, ada preferensi khusus gak?
Pengguna	Hmm, nggak terlalu sih. Yang penting bisa handle data dalam jumlah besar dan kita bisa coba beberapa model yang beda-beda.
David	Bagus. Terus kalo pengen evaluasi kinerjanya gimana?
Pengguna	Ya, kita memperbolehkan adanya false alarm. Ini karena nantinya kita akan saring lagi yang diprediksi sebagai NPL, apakah emang benar akan NPL? Dan kita juga akan beri treatment agar meminimalisir nasabah yang akan NPL.
David	Terakhir, ada insight apa aja dari eksplorasi data awalnya?
Pengguna	Udah ada beberapa pola menarik terkait perilaku pembayaran pelanggan dan faktor risiko lainnya. Mudah-mudahan nanti model bisa nangkep pola-pola itu dengan baik.
David	Keren, terima kasih udah sharing. Dengan info yang udah kamu kasih, saya bakal buat solusi yang sesuai dengan kebutuhan kamu.
Pengguna	Mantap, Pak David! Kami bener-bener berharap bisa kerja sama dengan Anda.
David	Sama-sama, semoga bisa ngasih hasil yang memuaskan. Kalo ada pertanyaan atau hal lain yang perlu dibahas, jangan ragu buat hubungi gue lagi ya.
Pengguna	Deal! Makasih banyak, Pak David. Sampai ketemu lagi!

David

Sama-sama! Sampai jumpa dan semoga proyeknya lancar ya!