CP_Project

February 16, 2026

[3]:
```
%pip install seaborn
%pip install kagglehub
%pip install scikit-learn
%pip install numpy
%pip install matplotlib
%pip install pandas
%pip install xgboost
```

```
Collecting seaborn
  Using cached seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Collecting numpy!=1.24.0,>=1.20 (from seaborn)
  Downloading numpy-2.4.2-cp313-cp313-macosx_14_0_arm64.whl.metadata (6.6 kB)
Collecting pandas>=1.2 (from seaborn)
  Downloading pandas-3.0.0-cp313-cp313-macosx_11_0_arm64.whl.metadata (79 kB)
Collecting matplotlib!=3.6.1,>=3.4 (from seaborn)
  Downloading matplotlib-3.10.8-cp313-cp313-macosx_11_0_arm64.whl.metadata (52
kB)
Collecting contourpy>=1.0.1 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading contourpy-1.3.3-cp313-cp313-macosx_11_0_arm64.whl.metadata (5.5
kB)
Collecting cycler>=0.10 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading fonttools-4.61.1-cp313-cp313-macosx_10_13_universal2.whl.metadata
(114 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading kiwisolver-1.4.9-cp313-cp313-macosx_11_0_arm64.whl.metadata (6.3
kB)
Requirement already satisfied: packaging>=20.0 in ./cp-env/lib/python3.13/site-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (26.0)
Collecting pillow>=8 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading pillow-12.1.1-cp313-cp313-macosx_11_0_arm64.whl.metadata (8.8 kB)
Collecting pyparsing>=3 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Using cached pyparsing-3.3.2-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: python-dateutil>=2.7 in ./cp-
env/lib/python3.13/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(2.9.0.post0)
Requirement already satisfied: six>=1.5 in ./cp-env/lib/python3.13/site-packages
```

```
(from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Using cached seaborn-0.13.2-py3-none-any.whl (294 kB)
Downloading matplotlib-3.10.8-cp313-cp313-macosx_11_0_arm64.whl (8.1 MB)
                         8.1/8.1 MB
230.5 kB/s  0:00:34m0:00:0100:02
Downloading contourpy-1.3.3-cp313-cp313-macosx_11_0_arm64.whl (274 kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.61.1-cp313-cp313-macosx_10_13_universal2.whl (2.8 MB)
                         2.8/2.8 MB
280.2 kB/s  0:00:10 eta 0:00:01
Downloading kiwisolver-1.4.9-cp313-cp313-macosx_11_0_arm64.whl (64 kB)
Downloading numpy-2.4.2-cp313-cp313-macosx_14_0_arm64.whl (5.2 MB)
                         5.2/5.2 MB
424.1 kB/s  0:00:12 eta 0:00:01
Downloading pandas-3.0.0-cp313-cp313-macosx_11_0_arm64.whl (9.9 MB)
                         9.9/9.9 MB
388.8 kB/s  0:00:25m0:00:0100:02
Downloading pillow-12.1.1-cp313-cp313-macosx_11_0_arm64.whl (4.7 MB)
                         4.7/4.7 MB
291.6 kB/s  0:00:15 eta 0:00:01
Using cached pyparsing-3.3.2-py3-none-any.whl (122 kB)
Installing collected packages: pyparsing, pillow, numpy, kiwisolver, fonttools,
cycler, pandas, contourpy, matplotlib, seaborn
                         10/10
[seaborn]9/10 [seaborn]ib]
Successfully installed contourpy-1.3.3 cycler-0.12.1 fonttools-4.61.1
kiwisolver-1.4.9 matplotlib-3.10.8 numpy-2.4.2 pandas-3.0.0 pillow-12.1.1
pyparsing-3.3.2 seaborn-0.13.2

[notice] A new release of pip is
available: 25.3 -> 26.0.1
[notice] To update, run:
/Users/nyeinchanaung/Downloads/ML:CP Project/cp-env/bin/python -m

pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Collecting kagglehub
  Downloading kagglehub-1.0.0-py3-none-any.whl.metadata (40 kB)
Collecting kagglesdk<1.0,>=0.1.14 (from kagglehub)
  Using cached kagglesdk-0.1.15-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: packaging in ./cp-env/lib/python3.13/site-
packages (from kagglehub) (26.0)
Requirement already satisfied: pyyaml in ./cp-env/lib/python3.13/site-packages
(from kagglehub) (6.0.3)
Requirement already satisfied: requests in ./cp-env/lib/python3.13/site-packages
(from kagglehub) (2.32.5)
Collecting tqdm (from kagglehub)
  Using cached tqdm-4.67.3-py3-none-any.whl.metadata (57 kB)
```

```
Collecting protobuf (from kagglesdk<1.0,>=0.1.14->kagglehub)
  Using cached protobuf-6.33.5-cp39-abi3-macosx_10_9_universal2.whl.metadata
(593 bytes)
Requirement already satisfied: charset_normalizer<4,>=2 in ./cp-
env/lib/python3.13/site-packages (from requests->kagglehub) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in ./cp-env/lib/python3.13/site-
packages (from requests->kagglehub) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./cp-
env/lib/python3.13/site-packages (from requests->kagglehub) (2.6.3)
Requirement already satisfied: certifi>=2017.4.17 in ./cp-
env/lib/python3.13/site-packages (from requests->kagglehub) (2026.1.4)
Downloading kagglehub-1.0.0-py3-none-any.whl (70 kB)
Using cached kagglesdk-0.1.15-py3-none-any.whl (160 kB)
Using cached protobuf-6.33.5-cp39-abi3-macosx_10_9_universal2.whl (427 kB)
Using cached tqdm-4.67.3-py3-none-any.whl (78 kB)
Installing collected packages: tqdm, protobuf, kagglesdk, kagglehub
                         4/4
[kagglehub]/4 [kagglesdk]
Successfully installed kagglehub-1.0.0 kagglesdk-0.1.15 protobuf-6.33.5
tqdm-4.67.3

[notice] A new release of pip is
available: 25.3 -> 26.0.1
[notice] To update, run:
/Users/nyeinchanaung/Downloads/ML:CP Project/cp-env/bin/python -m
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Collecting scikit-learn
  Downloading scikit_learn-1.8.0-cp313-cp313-macosx_12_0_arm64.whl.metadata (11
kB)
Requirement already satisfied: numpy>=1.24.1 in ./cp-env/lib/python3.13/site-
packages (from scikit-learn) (2.4.2)
Collecting scipy>=1.10.0 (from scikit-learn)
  Downloading scipy-1.17.0-cp313-cp313-macosx_14_0_arm64.whl.metadata (62 kB)
Collecting joblib>=1.3.0 (from scikit-learn)
  Using cached joblib-1.5.3-py3-none-any.whl.metadata (5.5 kB)
Collecting threadpoolctl>=3.2.0 (from scikit-learn)
  Using cached threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Downloading scikit_learn-1.8.0-cp313-cp313-macosx_12_0_arm64.whl (8.0 MB)
                         8.0/8.0 MB
242.9 kB/s  0:00:31m0:00:0100:02
Using cached joblib-1.5.3-py3-none-any.whl (309 kB)
Downloading scipy-1.17.0-cp313-cp313-macosx_14_0_arm64.whl (20.1 MB)
                         20.1/20.1 MB
351.2 kB/s  0:00:58m0:00:0100:03
Using cached threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
```

```
                         4/4 [scikit-
learn][0m [scikit-learn]
Successfully installed joblib-1.5.3 scikit-learn-1.8.0 scipy-1.17.0
threadpoolctl-3.6.0

[notice] A new release of pip is
available: 25.3 -> 26.0.1
[notice] To update, run:
/Users/nyeinchanaung/Downloads/ML:CP Project/cp-env/bin/python -m
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: numpy in ./cp-env/lib/python3.13/site-packages
(2.4.2)

[notice] A new release of pip is
available: 25.3 -> 26.0.1
[notice] To update, run:
/Users/nyeinchanaung/Downloads/ML:CP Project/cp-env/bin/python -m
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: matplotlib in ./cp-env/lib/python3.13/site-
packages (3.10.8)
Requirement already satisfied: contourpy>=1.0.1 in ./cp-env/lib/python3.13/site-
packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in ./cp-env/lib/python3.13/site-
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in ./cp-
env/lib/python3.13/site-packages (from matplotlib) (4.61.1)
Requirement already satisfied: kiwisolver>=1.3.1 in ./cp-
env/lib/python3.13/site-packages (from matplotlib) (1.4.9)
Requirement already satisfied: numpy>=1.23 in ./cp-env/lib/python3.13/site-
packages (from matplotlib) (2.4.2)
Requirement already satisfied: packaging>=20.0 in ./cp-env/lib/python3.13/site-
packages (from matplotlib) (26.0)
Requirement already satisfied: pillow>=8 in ./cp-env/lib/python3.13/site-
packages (from matplotlib) (12.1.1)
Requirement already satisfied: pyparsing>=3 in ./cp-env/lib/python3.13/site-
packages (from matplotlib) (3.3.2)
Requirement already satisfied: python-dateutil>=2.7 in ./cp-
env/lib/python3.13/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in ./cp-env/lib/python3.13/site-packages
(from python-dateutil>=2.7->matplotlib) (1.17.0)

[notice] A new release of pip is
available: 25.3 -> 26.0.1
[notice] To update, run:
```

```
/Users/nyeinchanaung/Downloads/ML:CP Project/cp-env/bin/python -m
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: pandas in ./cp-env/lib/python3.13/site-packages
(3.0.0)
Requirement already satisfied: numpy>=1.26.0 in ./cp-env/lib/python3.13/site-
packages (from pandas) (2.4.2)
Requirement already satisfied: python-dateutil>=2.8.2 in ./cp-
env/lib/python3.13/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in ./cp-env/lib/python3.13/site-packages
(from python-dateutil>=2.8.2->pandas) (1.17.0)

[notice] A new release of pip is
available: 25.3 -> 26.0.1
[notice] To update, run:
/Users/nyeinchanaung/Downloads/ML:CP Project/cp-env/bin/python -m
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Collecting xgboost
  Downloading xgboost-3.2.0-py3-none-macosx_12_0_arm64.whl.metadata (2.1 kB)
Requirement already satisfied: numpy in ./cp-env/lib/python3.13/site-packages
(from xgboost) (2.4.2)
Requirement already satisfied: scipy in ./cp-env/lib/python3.13/site-packages
(from xgboost) (1.17.0)
Downloading xgboost-3.2.0-py3-none-macosx_12_0_arm64.whl (2.3 MB)
                          2.3/2.3 MB
224.7 kB/s  0:00:11 eta 0:00:02
Installing collected packages: xgboost
Successfully installed xgboost-3.2.0

[notice] A new release of pip is
available: 25.3 -> 26.0.1
[notice] To update, run:
/Users/nyeinchanaung/Downloads/ML:CP Project/cp-env/bin/python -m
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

[1]:

```python
#import libraries
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, StratifiedKFold,␣
↪cross_validate, GridSearchCV
from sklearn.preprocessing import OneHotEncoder, StandardScaler
```

```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import (
    classification_report, confusion_matrix, ConfusionMatrixDisplay,
    roc_auc_score, RocCurveDisplay
)

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
```

[4]:

```
#load data
Datapath ='Data/CSE_student_performances.csv'
data = pd.read_csv(Datapath)
#explore data
print(data.head())
print(data.info())
print(data.describe())

# 2) Missing values (overall + per column)
missing_per_col = data.isna().sum().sort_values(ascending=False)
print("\nMissing values per column (top 15):")
print(missing_per_col.head(15))

# 3) Basic types
print("\nDtypes:\n", data.dtypes)

# 4) Quick numeric summary
display(data.describe(include="number").T)

# 5) Quick categorical summary (top categories)
cat_cols_guess = data.select_dtypes(include=["object", "category"]).columns
for c in cat_cols_guess[:10]:  # show first 10 only
    print(f"\nColumn: {c}")
    print(data[c].value_counts(dropna=False).head(10))
```

```
   Age   Gender AcademicPerformance TakingNoteInClass DepressionStatus  \
0   23     Male             Average                No        Sometimes
1   23     Male           Excellent         Sometimes              Yes
2   24     Male             Average                No        Sometimes
3   20   Female                Good               Yes        Sometimes
4   24   Female             Average               Yes              Yes

  FaceChallangesToCompleteAcademicTask LikePresentation  SleepPerDayHours  \
0                                  Yes              Yes                12
1                                   No              Yes                 8
2                            Sometimes               No                 8
3                                  Yes               No                 5
4                                  Yes              Yes                 5
```

```
   NumberOfFriend LikeNewThings
0             NaN           Yes
1            80.0           Yes
2            10.0           Yes
3            15.0           Yes
4             2.0           Yes
<class 'pandas.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 10 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   Age                                  99 non-null     int64
 1   Gender                               99 non-null     str
 2   AcademicPerformance                  99 non-null     str
 3   TakingNoteInClass                    99 non-null     str
 4   DepressionStatus                     99 non-null     str
 5   FaceChallangesToCompleteAcademicTask  99 non-null     str
 6   LikePresentation                     99 non-null     str
 7   SleepPerDayHours                     99 non-null     int64
 8   NumberOfFriend                       95 non-null     float64
 9   LikeNewThings                        99 non-null     str
dtypes: float64(1), int64(2), str(7)
memory usage: 7.9 KB
None
             Age  SleepPerDayHours  NumberOfFriend
count  99.000000         99.000000       95.000000
mean   22.515152          6.717172       16.189474
std     1.560767          1.738169       25.397811
min    20.000000          4.000000        0.000000
25%    21.000000          5.000000        3.000000
50%    23.000000          7.000000        6.000000
75%    24.000000          8.000000       15.000000
max    25.000000         12.000000      100.000000

Missing values per column (top 15):
NumberOfFriend                          4
Age                                     0
Gender                                  0
AcademicPerformance                     0
TakingNoteInClass                       0
DepressionStatus                        0
FaceChallangesToCompleteAcademicTask    0
LikePresentation                        0
SleepPerDayHours                        0
LikeNewThings                           0
dtype: int64
```

```
Dtypes:
 Age                                       int64
Gender                                      str
AcademicPerformance                         str
TakingNoteInClass                           str
DepressionStatus                            str
FaceChallangesToCompleteAcademicTask        str
LikePresentation                            str
SleepPerDayHours                          int64
NumberOfFriend                          float64
LikeNewThings                               str
dtype: object
                  count       mean        std   min   25%   50%   75%    max
Age                99.0  22.515152   1.560767  20.0  21.0  23.0  24.0   25.0
SleepPerDayHours   99.0   6.717172   1.738169   4.0   5.0   7.0   8.0   12.0
NumberOfFriend     95.0  16.189474  25.397811   0.0   3.0   6.0  15.0  100.0

Column: Gender
Gender
Male      56
Female    43
Name: count, dtype: int64

Column: AcademicPerformance
AcademicPerformance
Average          45
Good             41
Excellent         9
Below average     4
Name: count, dtype: int64

Column: TakingNoteInClass
TakingNoteInClass
Yes          61
Sometimes    26
No           12
Name: count, dtype: int64

Column: DepressionStatus
DepressionStatus
Sometimes    44
Yes          34
No           21
Name: count, dtype: int64

Column: FaceChallangesToCompleteAcademicTask
FaceChallangesToCompleteAcademicTask
```

```
Yes          37
No           31
Sometimes    31
Name: count, dtype: int64

Column: LikePresentation
LikePresentation
Yes    69
No     30
Name: count, dtype: int64

Column: LikeNewThings
LikeNewThings
Yes    89
No     10
Name: count, dtype: int64
```

```
/var/folders/_r/1mtvh9nn1ns69vtxm47ygl1c0000gn/T/ipykernel_15776/2410881801.py:2
1: Pandas4Warning: For backward compatibility, 'str' dtypes are included by
select_dtypes when 'object' dtype is specified. This behavior is deprecated and
will be removed in a future version. Explicitly pass 'str' to `include` to
select them, or to `exclude` to remove them and silence this warning.
See https://pandas.pydata.org/docs/user_guide/migration-3-strings.html#string-
migration-select-dtypes for details on how to write code that works with pandas
2 and 3.
  cat_cols_guess = data.select_dtypes(include=["object", "category"]).columns
```

[5]:

```python
#Define features and target
X = data.drop(columns=['DepressionStatus'])
y = data['DepressionStatus']

print("X shape:", X.shape)
print("y shape:", y.shape)

print("\nTarget distribution:")
print(y.value_counts())

print("Unique target values:", y.unique())
y = y.astype(str).str.strip().str.lower()


# Binary encoding
y = y.map({
    "yes": 1,
    "sometimes": 1,
    "no": 0
})
```

```python
print("New target distribution:")
print(y.value_counts())

# STEP 4: Detect feature types
# ==========================================

numeric_cols = X.select_dtypes(include=["int64", "float64"]).columns.tolist()
categorical_cols = X.select_dtypes(include=["object", "category", "string"]).
 ↪columns.tolist()

print("Numeric columns:", numeric_cols)
print("Categorical columns:", categorical_cols)

X_encoded = pd.get_dummies(X, columns=categorical_cols, drop_first=True)

print("Shape after encoding:", X_encoded.shape)
display(X_encoded.head())

print("Final X shape:", X_encoded.shape)
print("Final y shape:", y.shape)

# Make sure no missing values
print("Missing values in X:", X_encoded.isna().sum().sum())
print("Missing values in y:", y.isna().sum())

missing_per_column = X_encoded.isna().sum()
print(missing_per_column[missing_per_column > 0])

X_encoded["NumberOfFriend"] = X_encoded["NumberOfFriend"].fillna(
    X_encoded["NumberOfFriend"].median())
print("Missing values in NumberOfFriend after imputation:",
      X_encoded["NumberOfFriend"].isna().sum())
```

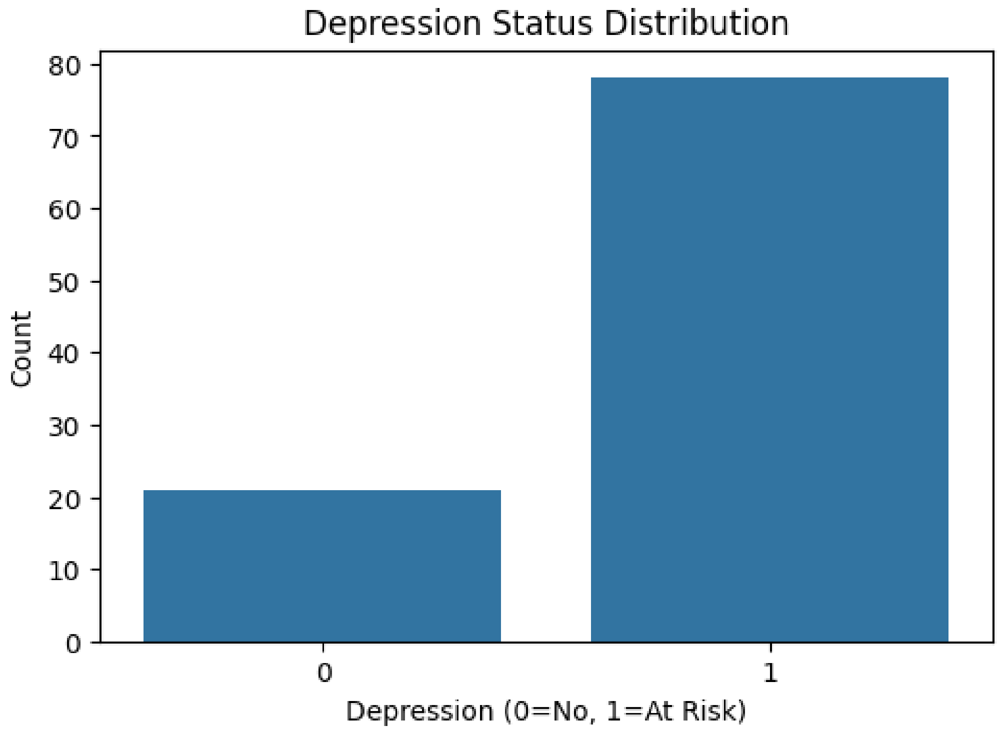
```
X shape: (99, 9)
y shape: (99,)

Target distribution:
DepressionStatus
Sometimes    44
Yes          34
No           21
Name: count, dtype: int64
Unique target values: <StringArray>
['Sometimes', 'Yes', 'No']
Length: 3, dtype: str
New target distribution:
DepressionStatus
1    78
```

```
0    21
Name: count, dtype: int64
Numeric columns: ['Age ', 'SleepPerDayHours', 'NumberOfFriend']
Categorical columns: ['Gender', 'AcademicPerformance', 'TakingNoteInClass',
'FaceChallangesToCompleteAcademicTask', 'LikePresentation', 'LikeNewThings']
Shape after encoding: (99, 13)

   Age   SleepPerDayHours  NumberOfFriend  Gender_Male  \
0    23                12             NaN         True
1    23                 8            80.0         True
2    24                 8            10.0         True
3    20                 5            15.0        False
4    24                 5             2.0        False

   AcademicPerformance_Below average  AcademicPerformance_Excellent  \
0                              False                          False
1                              False                           True
2                              False                          False
3                              False                          False
4                              False                          False

   AcademicPerformance_Good  TakingNoteInClass_Sometimes  \
0                     False                        False
1                     False                         True
2                     False                        False
3                      True                        False
4                     False                        False

   TakingNoteInClass_Yes  FaceChallangesToCompleteAcademicTask_Sometimes  \
0                  False                                           False
1                  False                                           False
2                  False                                            True
3                   True                                           False
4                   True                                           False

   FaceChallangesToCompleteAcademicTask_Yes  LikePresentation_Yes  \
0                                      True                  True
1                                     False                  True
2                                     False                 False
3                                      True                 False
4                                      True                  True

   LikeNewThings_Yes
0               True
1               True
2               True
3               True
4               True
```

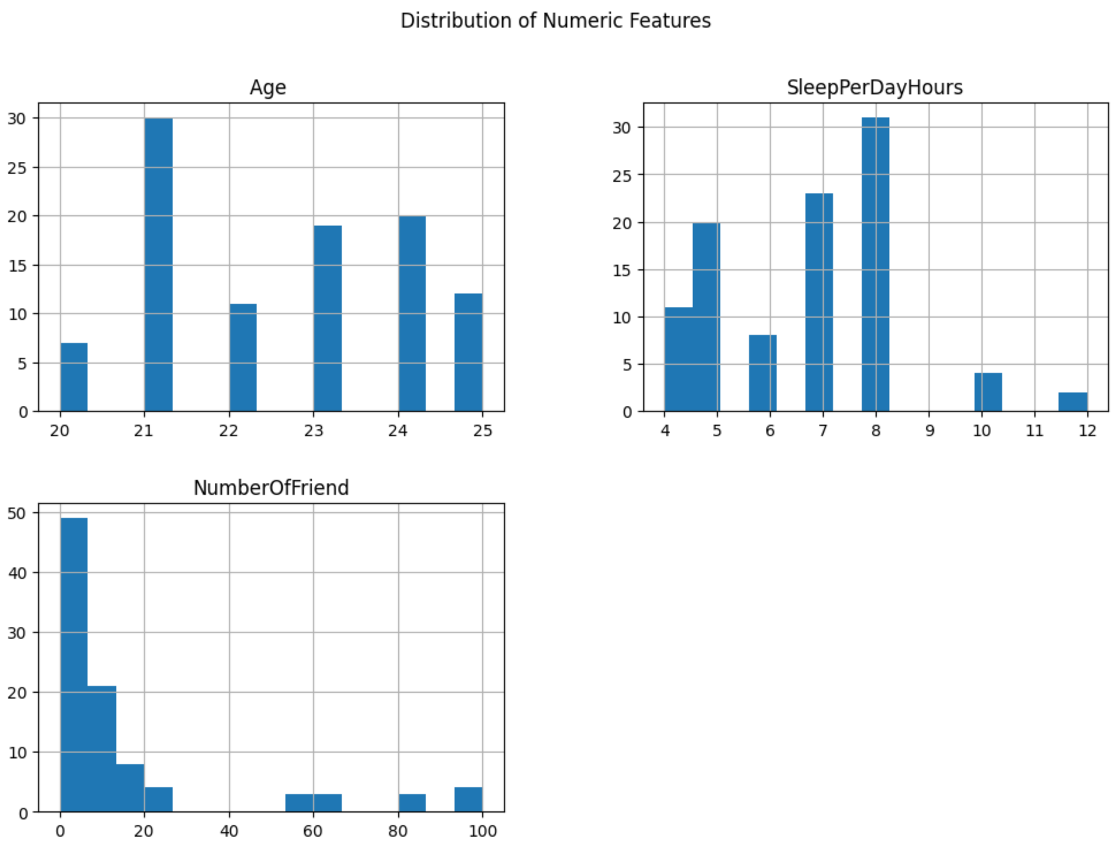
```
Final X shape: (99, 13)
Final y shape: (99,)
Missing values in X: 4
Missing values in y: 0
NumberOfFriend    4
dtype: int64
Missing values in NumberOfFriend after imputation: 0
```

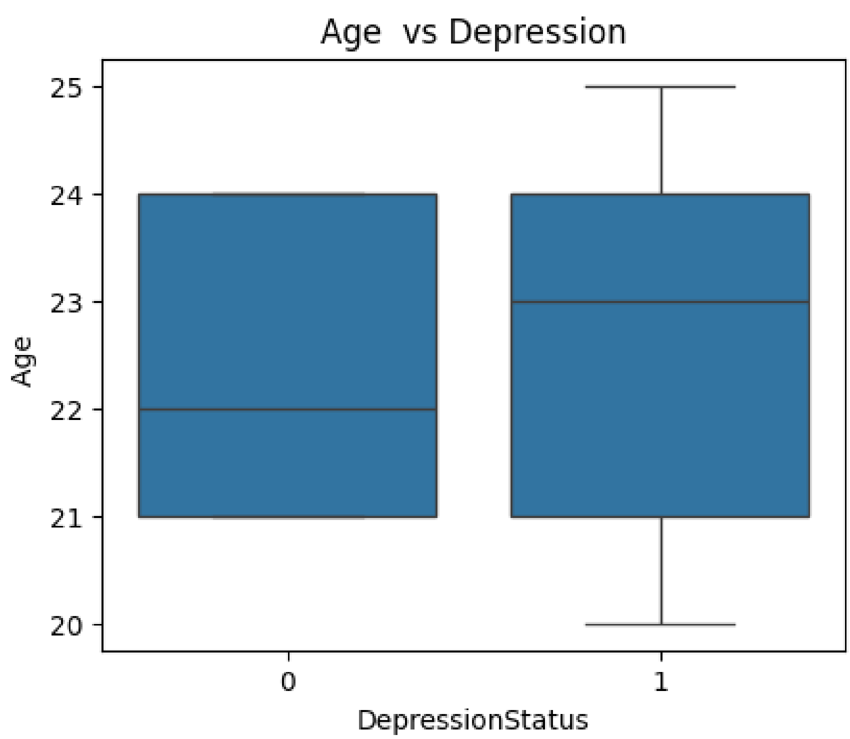
[6]:
```python
#EDA

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(6,4))
sns.countplot(x=y)
plt.title("Depression Status Distribution")
plt.xlabel("Depression (0=No, 1=At Risk)")
plt.ylabel("Count")
plt.show()

print(y.value_counts(normalize=True))

numeric_cols = X.select_dtypes(include=["int64", "float64"]).columns

X[numeric_cols].hist(figsize=(12,8), bins=15)
plt.suptitle("Distribution of Numeric Features")
plt.show()

for col in numeric_cols:
    plt.figure(figsize=(5,4))
    sns.boxplot(x=y, y=X[col])
    plt.title(f"{col} vs Depression")
    plt.show()

categorical_cols = X.select_dtypes(include=["object", "category", "string"]).
 ↪columns

for col in categorical_cols:
    plt.figure(figsize=(6,4))
    sns.countplot(x=col, hue=y, data=data)
    plt.title(f"{col} vs Depression")
    plt.xticks(rotation=45)
    plt.show()

plt.figure(figsize=(8,6))
sns.heatmap(X[numeric_cols].corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix (Numeric Features)")
```

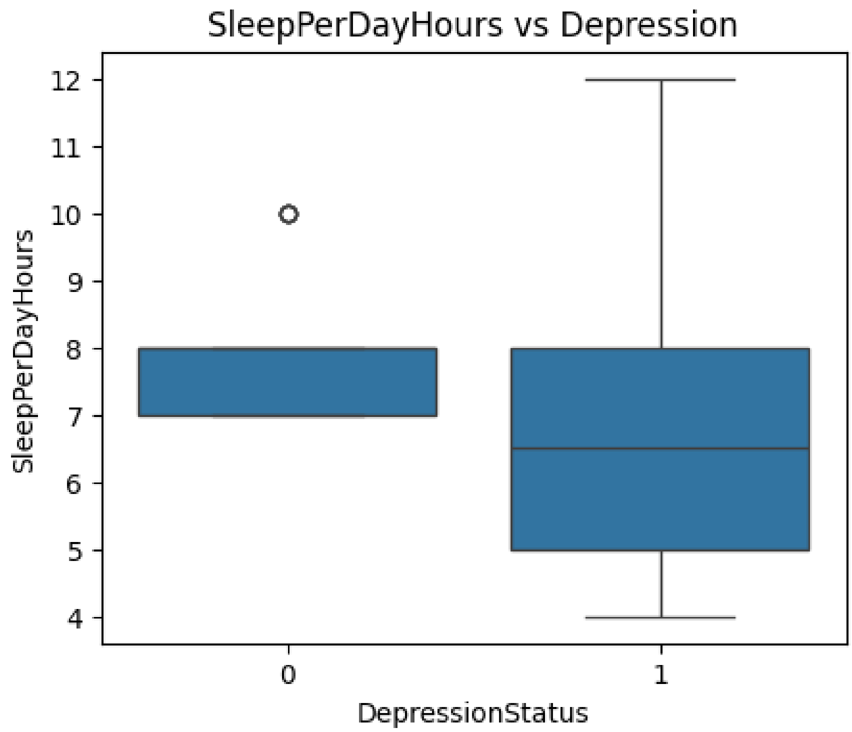
```
plt.show()
```

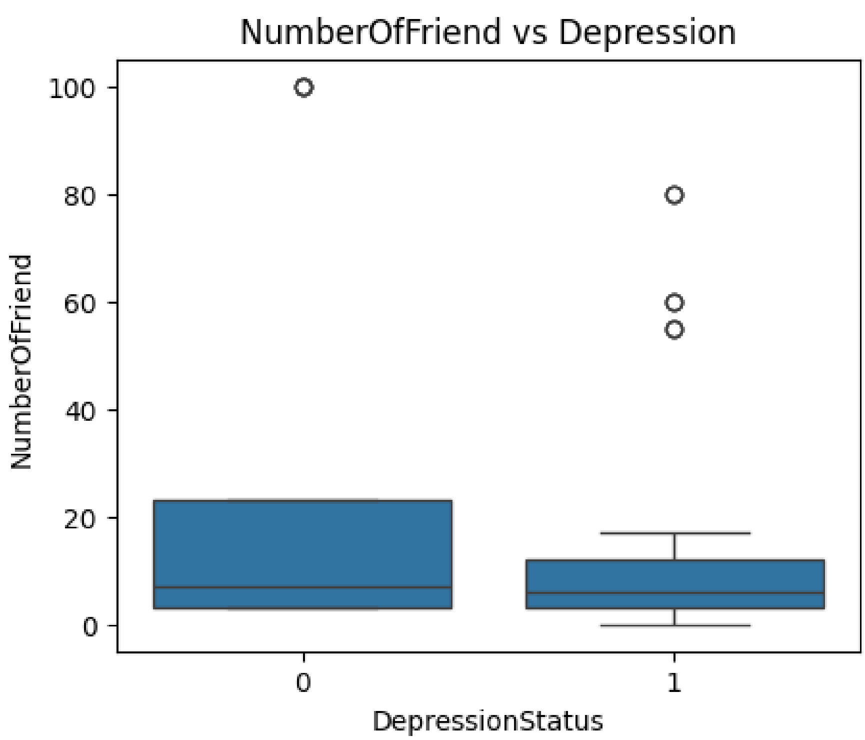


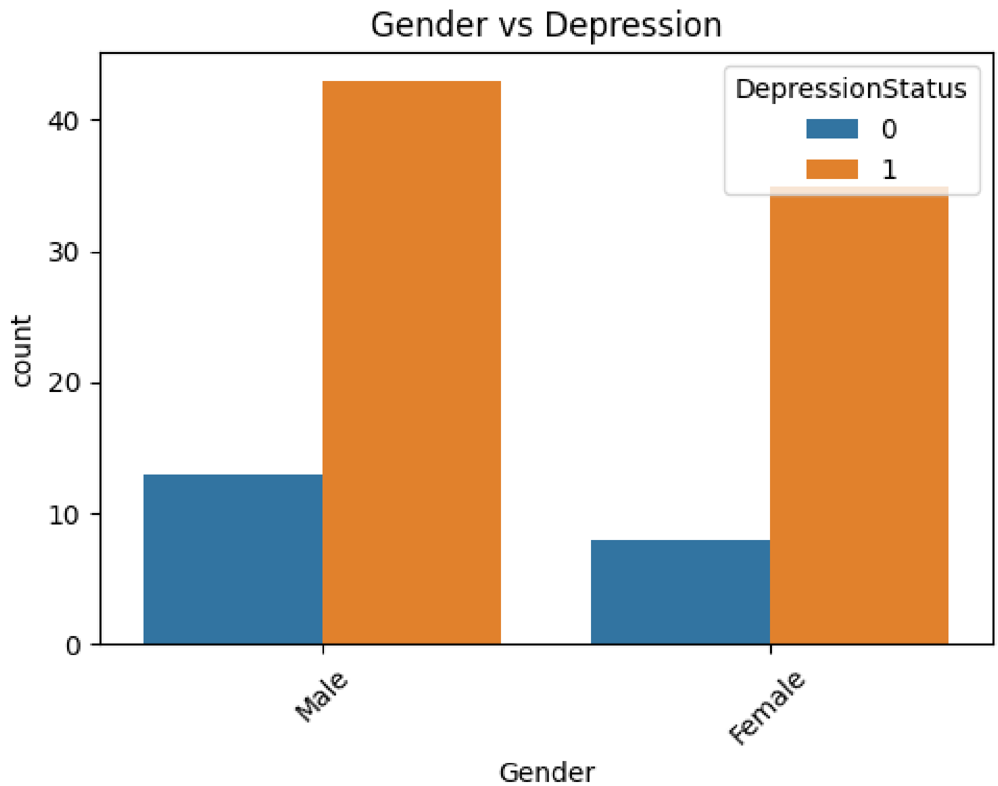

```
DepressionStatus
1    0.787879
0    0.212121
Name: proportion, dtype: float64
```

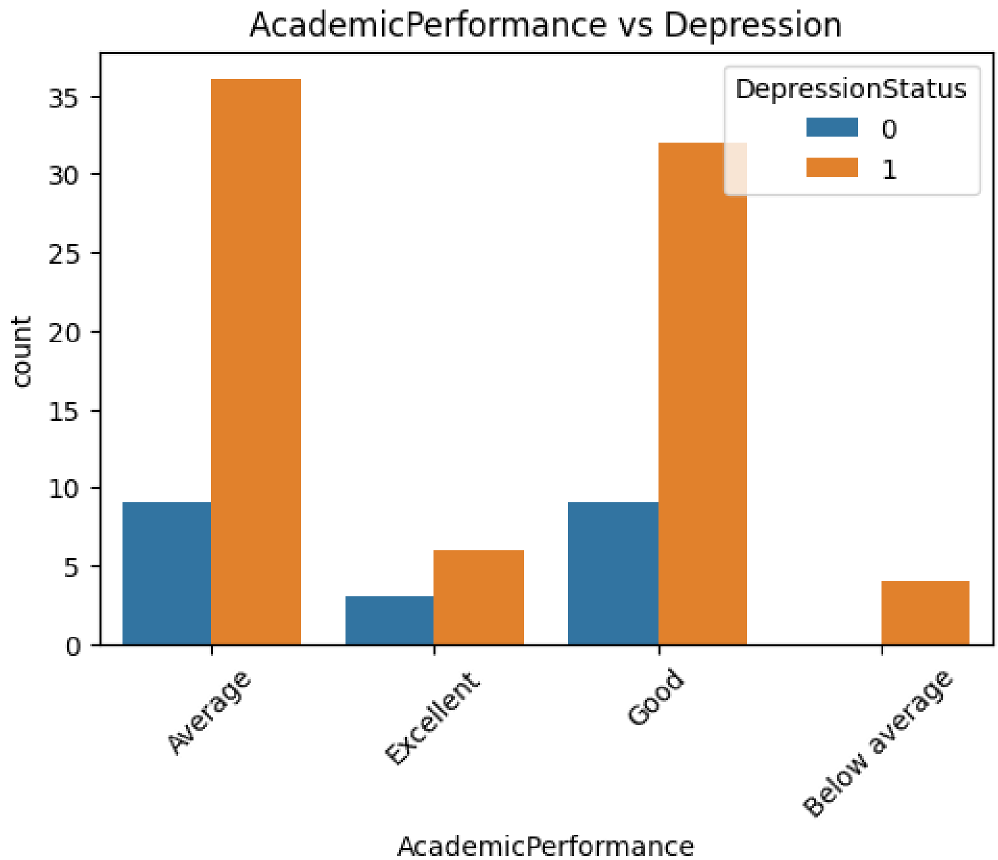

Distribution of Numeric Features
Age
SleepPerDayHours
NumberOfFriend

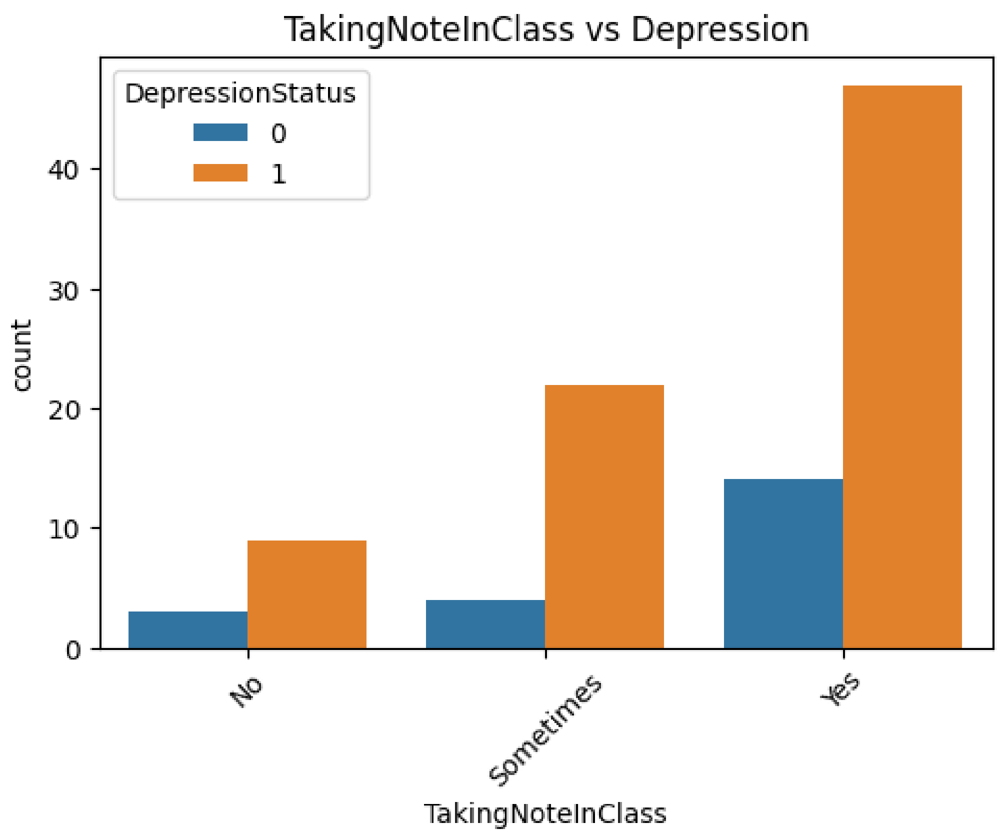

Age vs Depression
Age
DepressionStatus

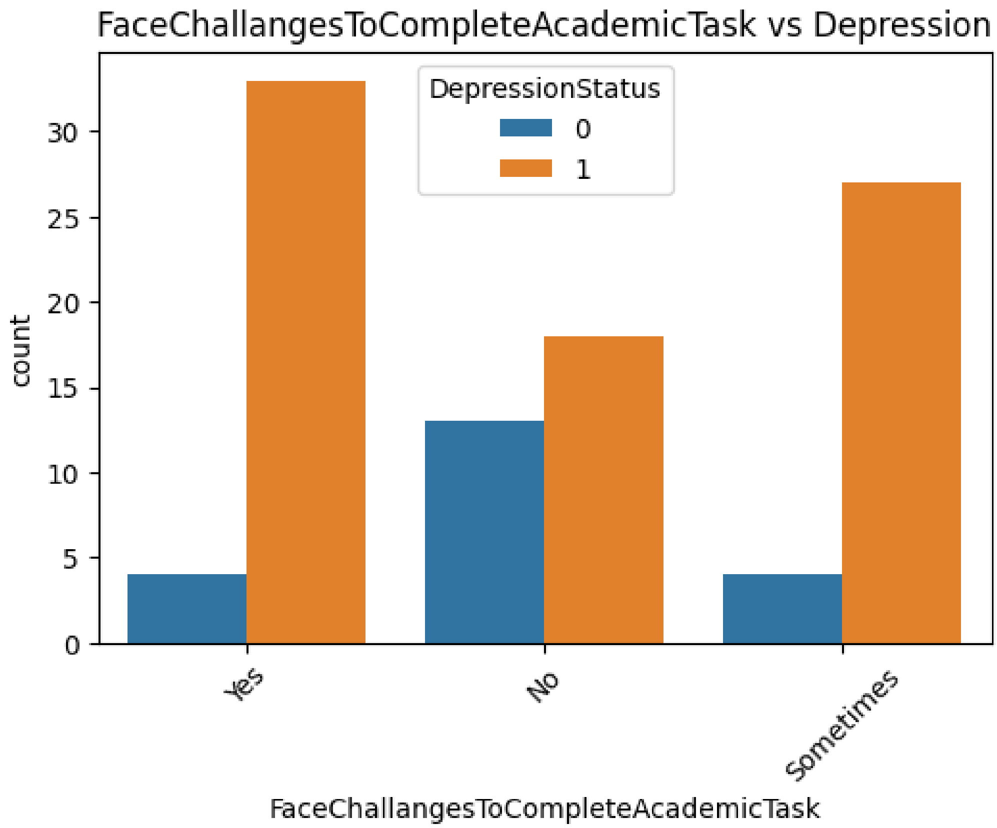

SleepPerDayHours vs Depression
SleepPerDayHours
12
11
10
9
8
7
6
5
4
0
1
DepressionStatus


NumberOfFriend vs Depression
NumberOfFriend
100
80
60
40
20
0
0
1
DepressionStatus

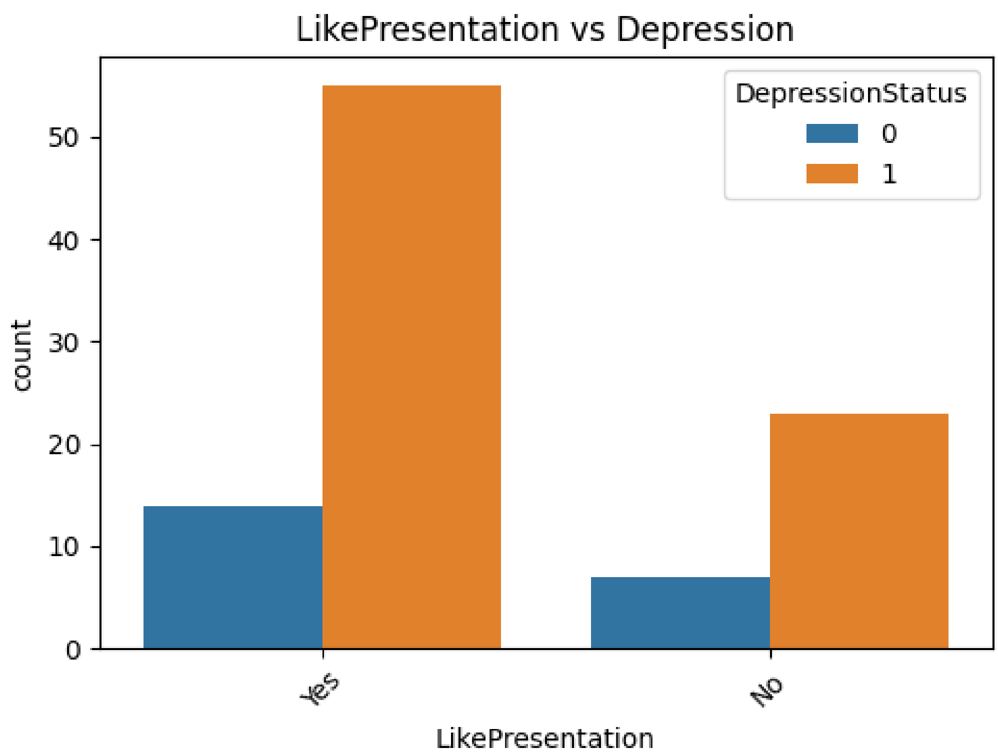

Gender vs Depression
DepressionStatus
0
1
40
30
20
10
0
count
Male
Female
Gender

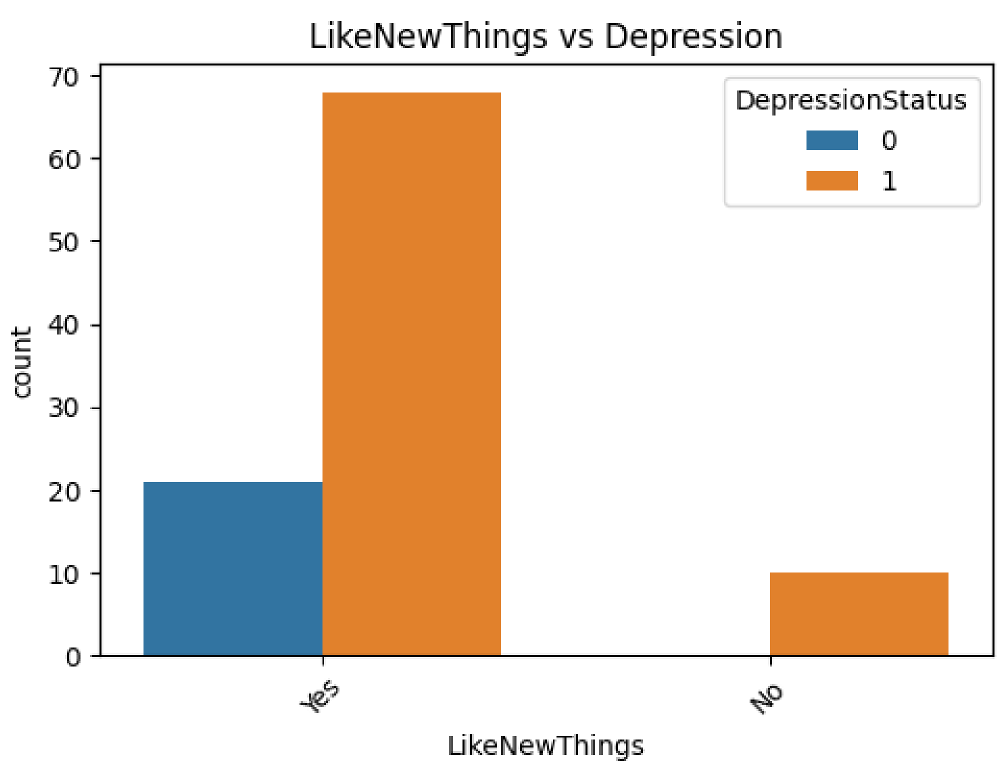

AcademicPerformance vs Depression
DepressionStatus
0
1
count
35
30
25
20
15
10
5
0
Average
Excellent
Good
Below average
AcademicPerformance

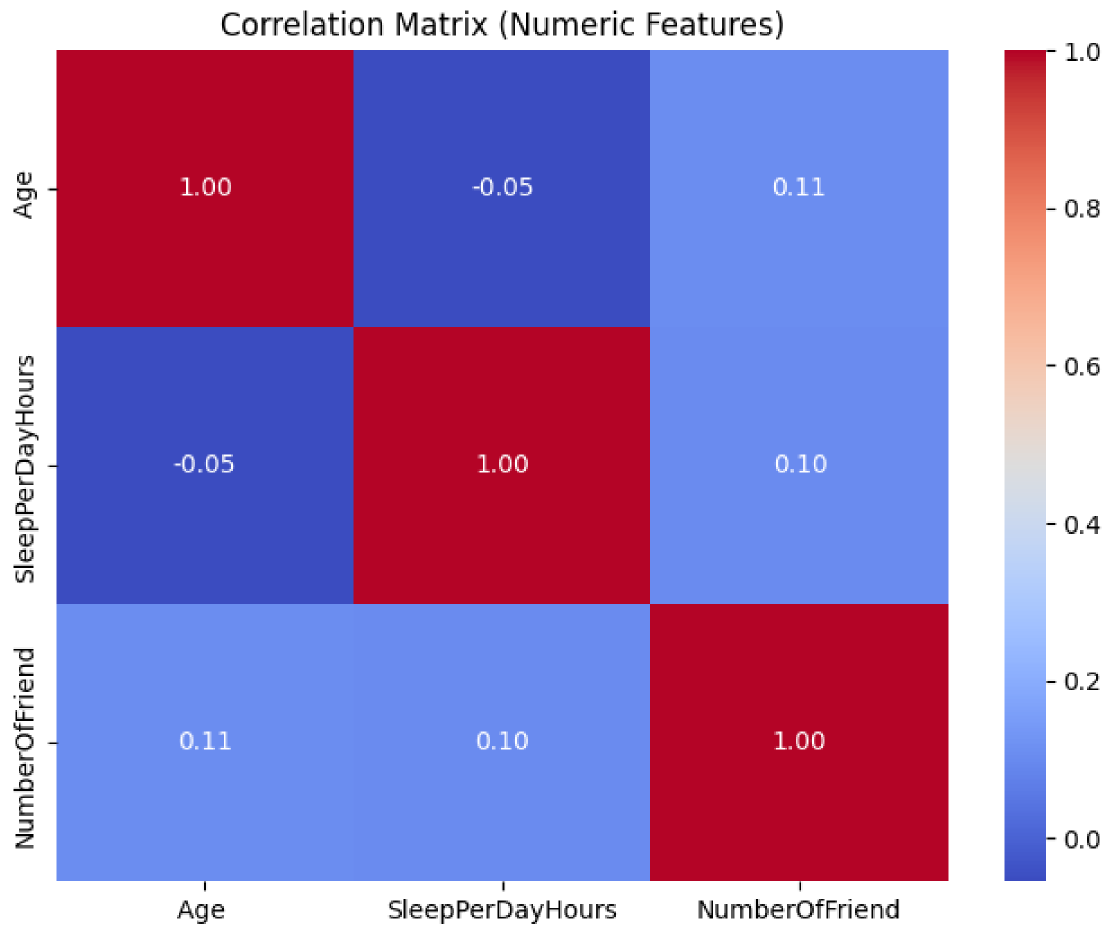

TakingNoteInClass vs Depression
DepressionStatus
0
1
count
0
10
20
30
40
No
Sometimes
Yes
TakingNoteInClass



FaceChallangesToCompleteAcademicTask vs Depression
DepressionStatus
0
1
count
30
25
20
15
10
5
0
Yes
No
Sometimes
FaceChallangesToCompleteAcademicTask


LikePresentation vs Depression
DepressionStatus
0
1
count
0
10
20
30
40
50
Yes
No
LikePresentation


LikeNewThings vs Depression
DepressionStatus
0
1
count
0
10
20
30
40
50
60
70
Yes
No
LikeNewThings



EDA Summary and Next-Step Decisions 1) Data situation (what the dataset looks like) * Size: 99 student records with 9 original features (mixed numeric + categorical). After one-hot encoding, features expanded to 13 columns. * Data types: * Numeric: Age, SleepPerDayHours, NumberOfFriend * Categorical: Gender, AcademicPerformance, TakingNoteInClass, FaceChallangesToCompleteAcademicTask, LikePresentation, LikeNewThings * Missing data: Only NumberOfFriend had 4 missing values; it was imputed using the median (missing values became 0). 2) Target variable (DepressionStatus) and class balance * Original target had 3 levels: Sometimes (44), Yes (34), No (21). * For modeling, it was converted into binary: * At Risk (1): Yes + Sometimes → 78 (78.8%) * Not Depressed (0): No → 21 (21.2%) * Implication: The target is imbalanced, so accuracy alone is not reliable. We should emphasize F1-score, Recall, and ROC-AUC. 3) What we learned from distributions and relationships * SleepPerDayHours vs Depression: The boxplot shows a noticeable difference between groups; sleep appears to be a potentially meaningful predictor. * NumberOfFriend: The histogram and boxplot show a highly skewed distribution with large outliers (values up to ~100). This feature may influence models and should be handled carefully. * Age: Only small differences between groups; likely a weaker predictor. * Categorical vs Depression: Some categories show visible differences (e.g., challenges completing tasks and academic performance patterns), but several features are not strongly separated visually. * Correlation (numeric): Correlations are close to zero (no strong multicollinearity), so numeric variables are not redundant. 4) Decision process for next steps (modeling plan) 1. Address class imbalance * Use Stratified splitting and Stratified K-Fold Cross-Validation. * Use class_weight="balanced" (at least for Logistic Regression and Random Forest). 2. Use leakage-safe preprocessing * Put imputation + scaling + encoding inside a Pipeline (not manual preprocessing before splitting). 3. Start with a baseline, then compare * Baseline: Logistic Regression (interpretable). * Compare with: Random Forest (handles nonlinearity and outliers better). * Optional: XGBoost only with careful tuning due to small sample size. 4. Evaluate with robust metrics * Report F1, Recall, and ROC-AUC (not accuracy only), plus confusion matrix. 5. Interpretability * Use Logistic Regression coefficients and/or permutation importance to identify key predictors. Conclusion: The dataset is small but clean and usable. The main challenges are class imbalance and skew/outliers in NumberOfFriend. The next step is to implement a Pipeline + Stratified 5-fold CV and compare baseline vs ensemble models using F1/Recall/ROC-AUC.

[10]:
```python
#prepocessing data pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Detect columns from ORIGINAL X
numeric_cols = X.select_dtypes(include=["int64", "float64"]).columns
categorical_cols = X.select_dtypes(include=["object", "category", "string"]).
 ↪columns

print("Numeric:", numeric_cols)
print("Categorical:", categorical_cols)

# Numeric pipeline → Impute missing values + Scale
numeric_pipeline = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())
])

# Categorical pipeline → Impute + OneHotEncode
categorical_pipeline = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="most_frequent")),
    ("encoder", OneHotEncoder(handle_unknown="ignore"))
])

# 2  Combine with ColumnTransformer

preprocessor = ColumnTransformer(
    transformers=[
        ("num", numeric_pipeline, numeric_cols),
        ("cat", categorical_pipeline, categorical_cols)
    ]
```

```
)

# 3  Train-Test Split
# ===============================

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=42,
    stratify=y
)
```

```
Numeric: Index(['Age ', 'SleepPerDayHours', 'NumberOfFriend'], dtype='str')
Categorical: Index(['Gender', 'AcademicPerformance', 'TakingNoteInClass',
       'FaceChallangesToCompleteAcademicTask', 'LikePresentation',
       'LikeNewThings'],
      dtype='str')
```

[11]:

```
#  Baseline pipeline (simple, interpretable)
from sklearn.linear_model import LogisticRegression

baseline_pipeline = Pipeline(steps=[
    ("preprocessing", preprocessor),
    ("model", LogisticRegression(max_iter=5000, random_state=42))
])



# Train Model

baseline_pipeline.fit(X_train, y_train)

# 6  Evaluate Baseline

from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix

y_pred = baseline_pipeline.predict(X_test)

print("Baseline Logistic Regression Results")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Baseline Logistic Regression Results
Accuracy: 0.9
```

```
Confusion Matrix:
 [[ 3  1]
 [ 1 15]]

Classification Report:
               precision    recall  f1-score   support

           0       0.75      0.75      0.75         4
           1       0.94      0.94      0.94        16

    accuracy                           0.90        20
   macro avg       0.84      0.84      0.84        20
weighted avg       0.90      0.90      0.90        20
```

[12]:

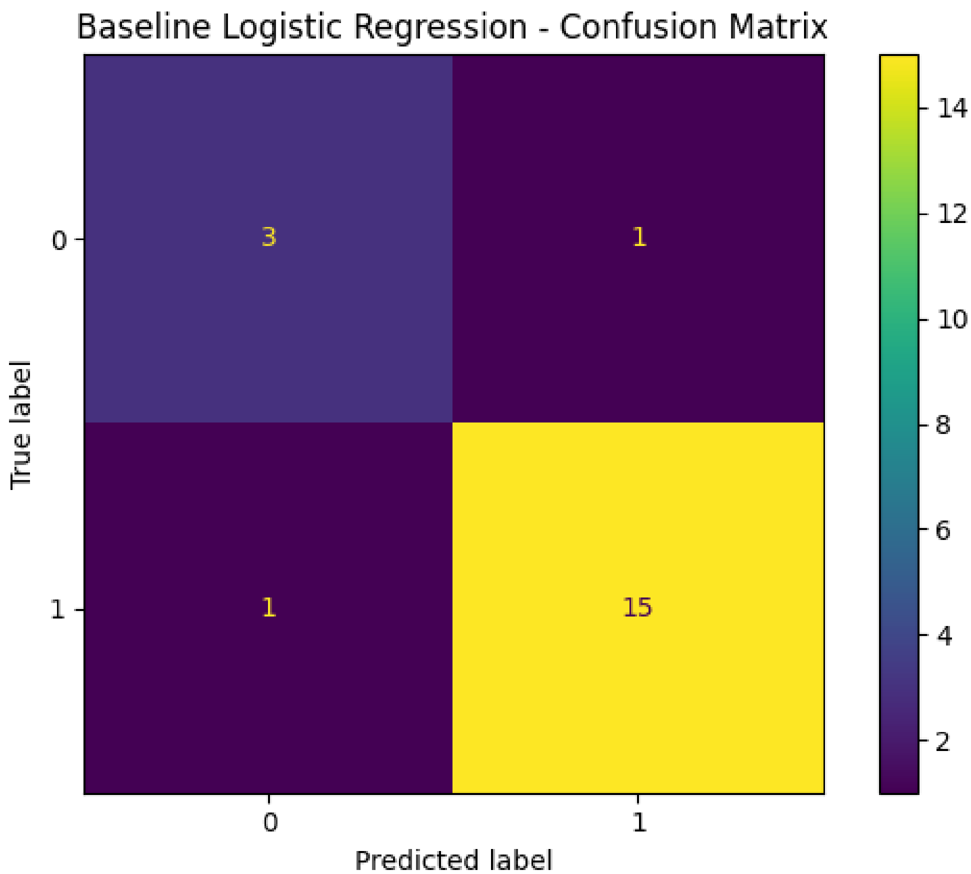
```
# ==============================
# Baseline model evaluation + plots (Matplotlib only)
# Works for binary or multi-class targets
# Assumes you already have:
# baseline_pipeline, X_test, y_test
# ==============================

import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import (
    ConfusionMatrixDisplay,
    accuracy_score,
    classification_report,
    roc_curve,
    auc,
    RocCurveDisplay,
    precision_recall_curve,
    PrecisionRecallDisplay
)

# ---- 1) Predictions
y_pred = baseline_pipeline.predict(X_test)

print("=== Baseline: Logistic Regression ===")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# ---- 2) Confusion Matrix Plot (works for binary + multiclass)
plt.figure(figsize=(7, 6))
ConfusionMatrixDisplay.from_estimator(
    baseline_pipeline,
    X_test,
```

```
    y_test,
    values_format="d",   # use "d" for counts; change to ".2f" if you want␣
 ↪normalized
)
plt.title("Baseline Logistic Regression - Confusion Matrix")
plt.tight_layout()
plt.show()

# ---- 3) ROC + Precision-Recall (ONLY for binary classification)
# If your target has more than 2 classes, we skip ROC/PR here (can add␣
 ↪One-vs-Rest later)
classes = np.unique(y_test)

if len(classes) == 2 and hasattr(baseline_pipeline, "predict_proba"):
    # Probability of the positive class
    y_prob = baseline_pipeline.predict_proba(X_test)[:, 1]

    # ROC Curve
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    roc_auc = auc(fpr, tpr)

    plt.figure(figsize=(7, 6))
    RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=roc_auc).plot()
    plt.title("Baseline Logistic Regression - ROC Curve")
    plt.tight_layout()
    plt.show()

    # Precision-Recall Curve
    precision, recall, _ = precision_recall_curve(y_test, y_prob)

    plt.figure(figsize=(7, 6))
    PrecisionRecallDisplay(precision=precision, recall=recall).plot()
    plt.title("Baseline Logistic Regression - Precision-Recall Curve")
    plt.tight_layout()
    plt.show()

else:
    print("\n[Info] ROC/PR plots are shown only for binary classification.")
    print("       Your target appears to have", len(classes), "classes:",␣
 ↪classes)
    print("       If you want, I can add multiclass ROC (One-vs-Rest) plots␣
 ↪next.")
```

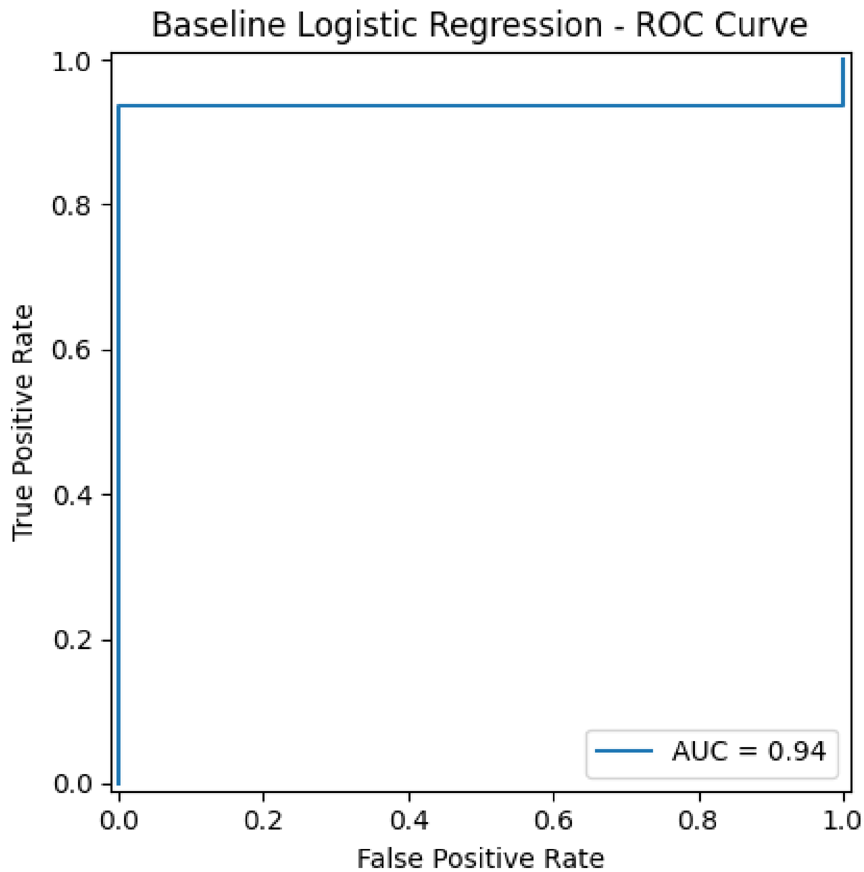
```
=== Baseline: Logistic Regression ===
Accuracy: 0.9

Classification Report:
```

```
              precision    recall  f1-score   support

           0       0.75      0.75      0.75         4
           1       0.94      0.94      0.94        16

    accuracy                           0.90        20
   macro avg       0.84      0.84      0.84        20
weighted avg       0.90      0.90      0.90        20
```

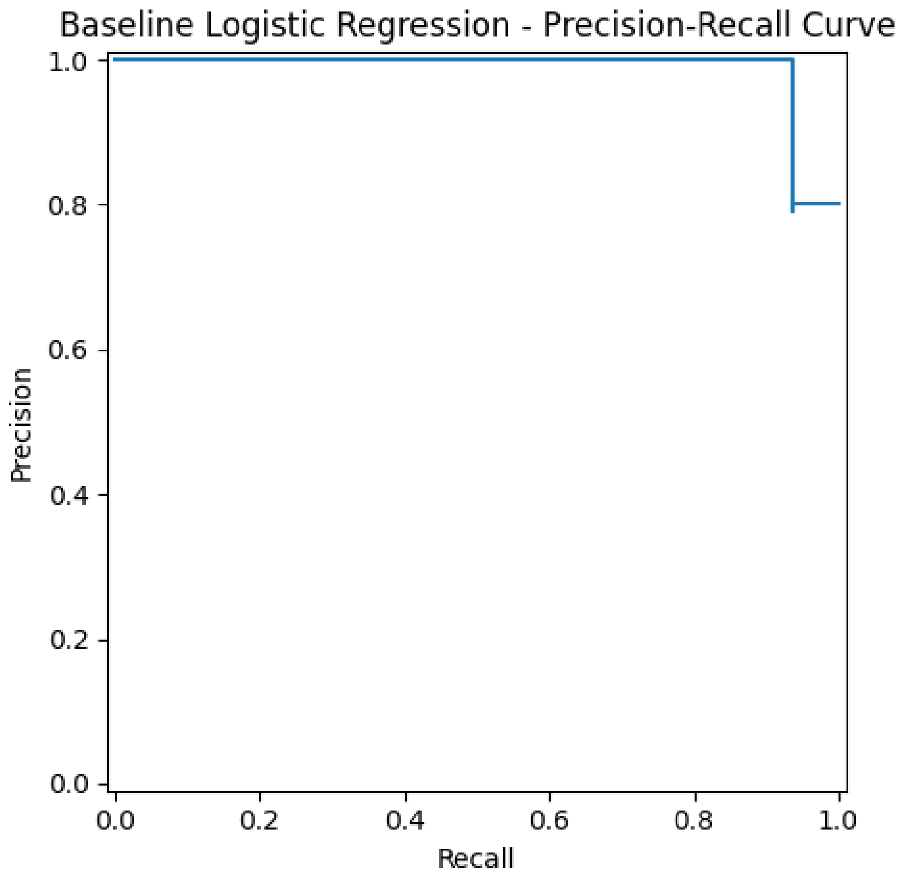
<Figure size 700x600 with 0 Axes>



<Figure size 700x600 with 0 Axes>



<Figure size 700x600 with 0 Axes>


Baseline Logistic Regression - Precision-Recall Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0