

## MLL Assignment 1

```
(base) C:\Users\nilesh>conda create --name ml_lab python=3.10
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 24.1.2

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=24.1.2

## Package Plan ##

environment location: C:\Users\nilesh\anaconda3\envs\ml_lab

added / updated specs:
- python=3.10

The following packages will be downloaded:

  package          build
  -----          -----
bzip2-1.0.8           h2bbff1b_5      78 KB
openssl-3.0.13         h2bbff1b_0      7.4 MB
tzdata-2024a          h04d1e81_0     116 KB
xz-5.4.6              h8cc25b3_0     587 KB
  -----
                           Total:        8.2 MB

The following NEW packages will be INSTALLED:

bzip2                  pkgs/main/win-64::bzip2-1.0.8-h2bbff1b_5
ca-certificates         pkgs/main/win-64::ca-certificates-2023.12.12-haa95532_0
libffi                 pkgs/main/win-64::libffi-3.4.4-hd77b12b_0
openssl                pkgs/main/win-64::openssl-3.0.13-h2bbff1b_0
pip                    pkgs/main/win-64::pip-23.3.1-py310haa95532_0
python                 pkgs/main/win-64::python-3.10.13-he1021f5_0
setuptools              pkgs/main/win-64::setuptools-68.2.2-py310haa95532_0
sqlite                 pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
tk                      pkgs/main/win-64::tk-8.6.12-h2bbff1b_0
tzdata                 pkgs/main/noarch::tzdata-2024a-h04d1e81_0
vc                      pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime           pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel                  pkgs/main/win-64::wheel-0.41.2-py310haa95532_0
xz                      pkgs/main/win-64::xz-5.4.6-h8cc25b3_0
zlib                   pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
```

```
(base) C:\Users\ Nilesh> conda activate ml_lab
(ml_lab) C:\Users\ Nilesh> pip install numpy
Collecting numpy
  Downloading numpy-1.26.4-cp310-cp310-win_amd64.whl.metadata (61 kB)
    10.0/61.0 kB 0.0 MB/s eta 0:00:00
  Downloading numpy-1.26.4-cp310-cp310-win_amd64.whl (15.8 MB)
    15.8/15.8 kB 11.5 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.26.4

(ml_lab) C:\Users\ Nilesh> pip install pandas
Collecting pandas
  Downloading pandas-2.2.1-cp310-cp310-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy<2,>=1.22.4 in c:\users\ Nilesh\anaconda3\envs\ml_lab\lib\site-packages (from pandas) (1.26.4)
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2024.1 (from pandas)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Downloading pandas-2.2.1-cp310-cp310-win_amd64.whl (11.6 MB)
  11.6/11.6 kB 0.1 MB/s eta 0:00:00
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
  229.9/229.9 kB 14.6 MB/s eta 0:00:00
Downloading pytz-2024.1-py2.py3-none-any.whl (505 kB)
  505.5/505.5 kB 7.9 MB/s eta 0:00:00
Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
  345.4/345.4 kB 7.3 MB/s eta 0:00:00
Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, tzdata, six, python-dateutil, pandas
Successfully installed pandas-2.2.1 python-dateutil-2.9.0 post0 pytz-2024.1 six-1.16.0 tzdata-2024.1
```

```
(ml_lab) C:\Users\ Nilesh> pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.8.3-cp310-cp310-win_amd64.whl.metadata (5.9 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Using cached contourpy-1.2.0-cp310-cp310-win_amd64.whl.metadata (5.8 kB)
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.49.0-cp310-cp310-win_amd64.whl.metadata (162 kB)
    162.3/162.3 kB 0.0 MB/s eta 0:00:00
Collecting kisilver>=1.3.1 (from matplotlib)
  Using cached kisilver-1.4.0-cp310-cp310-win_amd64.whl.metadata (6.5 kB)
Requirement already satisfied: numpy<2,>=1.21 in c:\users\ Nilesh\anaconda3\envs\ml_lab\lib\site-packages (from matplotlib) (1.26.4)
Collecting packaging>=20.0 (from matplotlib)
  Using cached packaging-23.2-py3-none-any.whl.metadata (3.2 kB)
Collecting pillow>=8 (from matplotlib)
  Using cached pillow-10.2.0-cp310-cp310-win_amd64.whl.metadata (9.9 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Using cached pyparsing-3.1.1-py3-none-any.whl.metadata (5.1 kB)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\ Nilesh\anaconda3\envs\ml_lab\lib\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\ Nilesh\anaconda3\envs\ml_lab\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Downloading matplotlib-3.8.3-cp310-cp310-win_amd64.whl (7.6 MB)
  7.6/7.6 kB 0.5 MB/s eta 0:00:00
Using cached contourpy-1.2.0-cp310-cp310-win_amd64.whl (184 kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.49.0-cp310-cp310-win_amd64.whl (2.2 MB)
  2.2/2.2 kB 0.1 MB/s eta 0:00:00
Using cached kisilver-1.4.0-cp310-cp310-win_amd64.whl (56 kB)
Using cached packaging-23.2-py3-none-any.whl (53 kB)
Using cached pillow-10.2.0-cp310-cp310-win_amd64.whl (2.6 MB)
Using cached pyparsing-3.1.1-py3-none-any.whl (103 kB)
Installing collected packages: pyparsing, pillow, packaging, kisilver, fonttools, cycler, contourpy, matplotlib
Successfully installed contourpy-1.2.0 cycler-0.12.1 fonttools-4.49.0 kisilver-1.4.0 matplotlib-3.8.3 packaging-23.2 pillow-10.2.0 pyparsing-3.1.1
```

```
(ml_lab) C:\Users\ Nilesh> pip install scikit-learn
Collecting scikit-learn
  Downloading scikit_learn-1.4.1.post1-cp310-cp310-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy<2,>=1.19.5 in c:\users\ Nilesh\anaconda3\envs\ml_lab\lib\site-packages (from scikit-learn) (1.26.4)
Collecting scipy>=1.6.0 (from scikit-learn)
  Downloading scipy-1.12.0-cp310-cp310-win_amd64.whl.metadata (60 kB)
    60.4/60.4 kB 0.0 MB/s eta 0:00:00
Collecting joblib>=1.2.0 (from scikit-learn)
  Using cached joblib-1.3.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=2.0.0 (from scikit-learn)
  Downloading threadpoolctl-3.3.0-py3-none-any.whl.metadata (12 kB)
  Downloading scikit_learn-1.4.1.post1-cp310-cp310-win_amd64.whl (10.6 MB)
    10.6/10.6 kB 0.0 MB/s eta 0:00:00
Using cached joblib-1.3.2-py3-none-any.whl (302 kB)
Downloading scipy-1.12.0-cp310-cp310-win_amd64.whl (48.2 kB)
  48.2/48.2 kB 0.0 MB/s eta 0:00:00
  Downloading threadpoolctl-3.3.0-py3-none-any.whl (17 kB)
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
Successfully installed joblib-1.3.2 scikit-learn-1.4.1.post1 scipy-1.12.0 threadpoolctl-3.3.0
```

# Numpy

```
[3]: import numpy as np
```

```
[2]: print(np.__version__)
```

1.26.3

```
[36]: test = np.array([1,2,3,4,5])  
test1 = np.array([(1,2,3), (4,5,6)])
```

```
[10]: print(test)  
print(test1)
```

```
[1 2 3 4 5]  
[[1 2 3]  
 [4 5 6]]
```

```
[21]: print("shape of the array:", test.shape)  
print("length of the array: ", len(test))  
print("dimensions of the array: ", test.ndim  
) print("data type of array a: ", test.dtype)
```

```
shape of the array: (5,)  
length of the array: 5  
dimensions of the array: 1  
data type of array a: float64
```

```
[22]: print("shape of the array:", test1.shape)  
print("length of the array: ", len(test1))  
print("dimensions of the array: ", test1.ndim  
) print("data type of array a: ", test1.dtype)
```

```
shape of the array: (2, 3)  
length of the array: 2  
dimensions of the array: 2  
data type of array a: int32
```

```
[23]: c = test.astype(int)#convert array into other  
datatype print(c)
```

```
[1 2 3 4 5]

[27]: d = np.array([(1,2,3), (4,5,6)], [(7,8,9), (10,11,12)])
print(d)
print("shape of the array:", d.shape)
print("length of the array: ", len(d))
print("dimensions of the array: ", d.ndim)
print("data type of array a: ", d.dtype)
```

```
[[[ 1  2  3]
 [ 4  5  6]]

 [[ 7  8  9]
 [10 11 12]]]

shape of the array: (2, 2, 3)
length of the array: 2
dimensions of the array: 3
data type of array a: int32
```

```
[13]: t1 = np.zeros((3,4))
print(t1)
t2 = np.ones((3,4))
print(t2)
f = np.arange(10,25,2)
print(f)
h = np.linspace(0,2,9) # equal distance from each element from 1 to 2
print(h)
r = np.random.random((2,3))
print(r)
e = np.empty((2,2))
print(e)
i = np.eye(3)
print(i)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
[10 12 14 16 18 20 22 24]
[0. 0.25 0.5 0.75 1. 1.25 1.5 1.75 2. ]
[[0.97466918 0.94023064 0.51741172]
 [0.38217297 0.42256136 0.00248236]]
[[2.54639495e-313 3.39519327e-313]
 [4.24399158e-313 5.09278990e-313]]
[[1. 0. 0.]
 [0. 1. 0.]]
```

```
[0. 0. 1.])
```

```
[19]: #loading and saving array  
np.save("D:\MIT ADT\Third Year Sem - 2\ML LAB\my_array", test)  
  
temp = np.load("D:\MIT ADT\Third Year Sem - 2\ML LAB\my_array.npy")  
print(temp)
```

```
[1.6 2. 3. 4. 5.]
```

```
[37]: temp.sum()  
print(temp.min())  
  
print(temp.max(axis=0))  
print(test1)  
print("test 1 - 0: ", test1.max(axis=0))  
print("test 1 - 1: ", test1.max(axis=1))  
  
print(np.median(test))  
print(np.std(test))  
  
print(np.transpose(test1))
```

```
1.6  
5.0  
[[1 2 3]  
 [4 5 6]]  
test 1 - 0: [4 5 6]  
test 1 - 1: [3 6]  
3.0  
1.4142135623730951  
[[1 4]  
 [2 5]  
 [3 6]]
```

```
[34]: test[2:]
```

```
[34]: array([3, 4, 5])
```

```
[43]: k = (test1.ravel()) #more dimension to 1d flattning of the matrix  
print(k)  
  
r = k.reshape(2,3)# should be proportion  
print(r)
```

```
[1 2 3 4 5 6]  
[[1 2 3]  
 [4 5 6]]
```

```
[59]: my_array = np.array([1,2,3,4,5])
z = (np.resize(my_array,2))
print(z)

print(np.append(my_array, 8))

print(np.insert(my_array,[2], 5))

print(np.delete(my_array,[2]))

print(np.dot(test, test))
```

```
[1 2]
[1 2 3 4 5 8]
[1 2 5 3 4 5]
[1 2 4 5]
```

```
55
```

```
[ ]:
```

# Pandas

```
[3]: import pandas as pd  
import numpy as np  
from pandas import DataFrame
```

```
[4]: df = DataFrame({  
    'Name': ['Abc', 'Pqr', 'Xyz'],  
    'Age':[10,20,30]  
})  
  
df
```

```
[4]:   Name  Age  
0   Abc   10  
1   Pqr   20  
2   Xyz   30
```

```
[ ]:
```

```
[5]: df.Name
```

```
[5]: 0    Abc  
1    Pqr  
2    Xyz  
Name: Name, dtype: object
```

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

```
[6]: (3, 2)
```

```
[7]: df.Age
```

```
[7]: 0    10  
1    20  
2    30  
Name: Age, dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
--- 
 0   Name     3 non-null      object  
 1   Age      3 non-null      int64  
dtypes: int64(1), object(1)
memory usage: 176.0+ bytes
```

```
[9]: df1 = pd.read_csv("D:\\MIT ADT\\Third Year Sem - 2\\ML LAB\\Assign 2\\diabetes.csv")
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype    
--- 
 0   Pregnancies      768 non-null    int64    
 1   Glucose          768 non-null    int64    
 2   BloodPressure    768 non-null    int64    
 3   SkinThickness    768 non-null    int64    
 4   Insulin          768 non-null    int64    
 5   BMI              768 non-null    float64  
 6   DiabetesPedigreeFunction 768 non-null    float64  
 7   Age              768 non-null    int64    
 8   Outcome          768 non-null    int64    
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
[10]: df1.tail(10) #bottom 5 records
```

```
Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI \
758          1        106            76             0         0  37.5
759          6        190            92             0         0  35.5
760          2        88             58            26         16  28.4
761          9        170            74            31         0  44.0
762          9        89             62             0         0  22.5
763         10        101            76            48        180  32.9
764          2        122            70            27         0  36.8
765          5        121            72            23        112  26.2
766          1        126            60             0         0  30.1
767          1         93            70            31         0  30.4

DiabetesPedigreeFunction  Age  Outcome
758                  0.197    26      0
759                  0.278    66      1
```

```
760           0.766   22      0
761           0.403   43      1
762           0.142   33      0
763           0.171   63      0
764           0.340   27      0
765           0.245   30      0
766           0.349   47      1
767           0.315   23      0
```

```
[11]: df1.head() #top 5 records
```

```
[11]:    Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI \
0            6        148          72            35          0  33.6
1            1        85           66            29          0  26.6
2            8        183          64            0          0  23.3
3            1        89           66            23         94  28.1
4            0        137          40            35         168 43.1

          DiabetesPedigreeFunction  Age  Outcome
0                  0.627    50      1
1                  0.351    31      0
2                  0.672    32      1
3                  0.167    21      0
4                  2.288    33      1
```

```
[12]: df1.describe() #for the columns only for numerical data
```

```
[12]:    Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin \
count    768.000000  768.000000  768.000000  768.000000  768.000000
mean     3.845052  120.894531  69.105469  20.536458  79.799479
std      3.369578  31.972618  19.355807  15.952218 115.244002
min      0.000000  0.000000  0.000000  0.000000  0.000000
25%     1.000000  99.000000  62.000000  0.000000  0.000000
50%     3.000000  117.000000  72.000000  23.000000 30.500000
75%     6.000000  140.250000  80.000000  32.000000 127.250000
max     17.000000  199.000000 122.000000  99.000000 846.000000

          BMI  DiabetesPedigreeFunction  Age  Outcome
count    768.000000                 768.000000  768.000000  768.000000
mean     31.992578                 0.471876  33.240885  0.348958
std      7.884160                  0.331329  11.760232  0.476951
min      0.000000                 0.078000  21.000000  0.000000
25%    27.300000                 0.243750  24.000000  0.000000
50%    32.000000                 0.372500  29.000000  0.000000
75%    36.600000                 0.626250  41.000000  1.000000
max    67.100000                 2.420000  81.000000  1.000000
```

```
[13]: df1.columns
```

```
[13]: Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
       dtype='object')
```

```
[14]: df1.columns[1]
```

```
[14]: 'Glucose'
```

```
[15]: df1.columns.values
```

```
[15]: array(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
       'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
       dtype=object)
```

```
[16]: print(len(df1.columns.values))
```

9

```
[17]: print(df['Age'])
print(df1[['Age','Outcome']])
```

```
0    10
1    20
2    30
Name: Age, dtype: int64
      Age  Outcome
0      50        1
1      31        0
2      32        1
3      21        0
4      33        1
..   ...
763    63        0
764    27        0
765    30        0
766    47        1
767    23        0
```

[768 rows x 2 columns]

```
[18]: X = df1.drop('Outcome', axis=1)
X
```

```
[18]:    Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI \
0           6         148            72            35          0   33.6
```

```
1          1    85      66      29      0  26.6
2          8   183      64      0      0  23.3
3          1    89      66     23    94  28.1
4          0   137      40     35  168  43.1
...
763        10   101      76      48    180  32.9
764        2   122      70      27      0  36.8
765        5   121      72      23   112  26.2
766        1   126      60      0      0  30.1
767        1    93      70      31      0  30.4
```

	DiabetesPedigreeFunction	Age
0	0.627	50
1	0.351	31
2	0.672	32
3	0.167	21
4	2.288	33
...	...	...
763	0.171	63
764	0.340	27
765	0.245	30
766	0.349	47
767	0.315	23

[768 rows x 8 columns]

```
[19]: Y = df1.Outcome  
Y
```

```
[19]: 0      1
1      0
2      1
3      0
4      1
...
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

```
[20]: df1.shape
```

(768, 9)