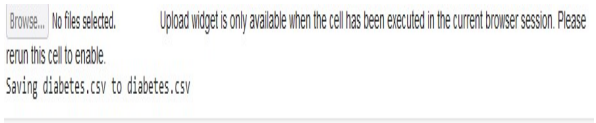
AI Programming

[Day-2]

Program 1: Pima_Indian_Dataset To build a Pima Indian dataset: download data, keep in folders. Note: Data can be in any format (XLS, CSV, etc).

Solution:

```
from google.colab import files
uploaded = files.upload()
```




```
import pandas as pd
df = pd.read_csv('https://github.com/npradaschnor/Pima-Indians-Diabetes-Dataset/raw/master/diabetes.csv')
df.to_csv('diabetes.csv')
print('Success!')
```

```
Success!
```

```
from google.colab import files
files.download('diabetes.csv')

import shutil
import os
from google.colab import files

print("Please upload kaggle.json")
uploaded = files.upload()

path = '/root/.kaggle/'
if not os.path.exists(path):
  os.mkdir(path)
shutil.move("kaggle.json", path)
print("Success!")
```


```
! kaggle datasets download uciml/pima-indians-diabetes-database
import zipfile as zip
f = zip.ZipFile("pima-indians-diabetes-database.zip")
f.extractall()
print('Success!')
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Downloading pima-indians-diabetes-database.zip to /content
  0% 0.00/8.91k [00:00<?, ?B/s]
100% 8.91k/8.91k [00:00<00:00, 25.5MB/s]
Success!
```

```
import os
import pandas as pd

df = pd.read_csv('diabetes.csv')
cols = df.columns.tolist()
p = []
n = []
for row in df.index:
  data = df.loc[row].tolist()
  outcome = int(data[-1])
  if outcome == 1: p.append(data)
  else: n.append(data)
dfp = pd.DataFrame(p, columns = cols).drop(labels=['Outcome'], axis=1)
dfn = pd.DataFrame(n, columns = cols).drop(labels=['Outcome'], axis=1)
print(dfp, '\n', dfn)
if not os.path.exists('Positive/'): os.mkdir('Positive/')
dfp.to_csv("Positive/diabetes.csv")
if not os.path.exists('Negative/'): os.mkdir('Negative/')
dfn.to_csv("Negative/diabetes.csv")
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6.0    148.0           72.0           35.0      0.0  33.6
1            8.0    183.0           64.0            0.0      0.0  23.3
2            0.0    137.0           40.0           35.0    168.0  43.1
3            3.0     78.0           50.0           32.0     88.0  31.0
4            2.0    197.0           70.0           45.0    543.0  30.5
..           ...      ...            ...            ...      ...   ...
263          1.0    128.0           88.0           39.0    110.0
36.5
264          0.0    123.0           72.0            0.0      0.0  36.3
265          6.0    190.0           92.0            0.0      0.0  35.5
266          9.0    170.0           74.0           31.0      0.0  44.0
267          1.0    126.0           60.0            0.0      0.0  30.1

     DiabetesPedigreeFunction   Age
0                       0.627  50.0
```

```
1              0.672  32.0
2              2.288  33.0
3              0.248  26.0
4              0.158  53.0
..               ...   ...
263            1.057  37.0
264            0.258  52.0
265            0.278  66.0
266            0.403  43.0
267            0.349  47.0

[268 rows x 8 columns]
     Pregnancies  Glucose  BloodPressure  SkinThickness
Insulin   BMI  \
0            1.0    85.0           66.0           29.0
0.0  26.6
1            1.0    89.0           66.0           23.0
94.0  28.1
2            5.0   116.0           74.0            0.0
0.0  25.6
3           10.0   115.0            0.0            0.0
0.0  35.3
4            4.0   110.0           92.0            0.0
0.0  37.6
..           ...     ...            ...            ...
...  ...
495          9.0    89.0           62.0            0.0
0.0  22.5
496         10.0   101.0           76.0           48.0
180.0
32.9
497          2.0   122.0           70.0           27.0
0.0  36.8
498          5.0   121.0           72.0           23.0
112.0
26.2
499          1.0    93.0           70.0           31.0
0.0  30.4

     DiabetesPedigreeFunction   Age
0                       0.351  31.0
1                       0.167  21.0
2                       0.201  30.0
3                       0.134  29.0
4                       0.191  30.0
..                        ...   ...
495                     0.142  33.0
496                     0.171  63.0
497                     0.340  27.0
498                     0.245  30.0
499                     0.315  23.0

[500 rows x 8 columns]
```

```
import warnings
warnings.simplefilter(action='ignore',
category=FutureWarning)
# supresses deprecated pandas.DataFrame.append
warning

import os
import pandas as pd

df = pd.read_csv('diabetes.csv')
cols = df.columns.tolist()
dfp = pd.DataFrame(columns = cols)
dfn = pd.DataFrame(columns = cols)
```

```
for row in df.index:
  data = df.loc[row].tolist()
  app = {}
  for i in range(len(data)):
    app[cols[i]] = data[i]
  outcome = int(data[-1])
  if outcome == 1: dfp = dfp.append(app,
ignore_index=True)
  else: dfn = dfn.append(app, ignore_index=True)
dfp = dfp.drop(labels=['Outcome'], axis=1)
dfn = dfn.drop(labels=['Outcome'], axis=1)
print(dfp, '\n', dfn)
if not os.path.exists('Positive/'): os.mkdir('Positive/')
dfp.to_csv("Positive/diabetes.csv")
if not os.path.exists('Negative/'): os.mkdir('Negative/')
dfn.to_csv("Negative/diabetes.csv")
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness
Insulin   BMI  \
0            6.0    148.0           72.0           35.0
0.0  33.6
1            8.0    183.0           64.0            0.0
0.0  23.3
2            0.0    137.0           40.0           35.0
168.0  43.1
3            3.0     78.0           50.0           32.0
88.0  31.0
4            2.0    197.0           70.0           45.0
543.0  30.5
..           ...     ...            ...            ...
...  ...
263          1.0    128.0           88.0           39.0
110.0
36.5
264          0.0    123.0           72.0            0.0
0.0  36.3
265          6.0    190.0           92.0            0.0
0.0  35.5
266          9.0    170.0           74.0           31.0
0.0  44.0
267          1.0    126.0           60.0            0.0
0.0  30.1

     DiabetesPedigreeFunction   Age
0                       0.627  50.0
1                       0.672  32.0
2                       2.288  33.0
3                       0.248  26.0
4                       0.158  53.0
..                        ...   ...
263                     1.057  37.0
264                     0.258  52.0
265                     0.278  66.0
266                     0.403  43.0
267                     0.349  47.0

[268 rows x 8 columns]
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness
Insulin   BMI  \
0            1.0    85.0           66.0           29.0
0.0  26.6
1            1.0    89.0           66.0           23.0
94.0  28.1
2            5.0   116.0           74.0            0.0
0.0  25.6
3           10.0   115.0            0.0            0.0
0.0  35.3
4            4.0   110.0           92.0            0.0
0.0  37.6
..           ...     ...            ...            ...
...  ...
495          9.0    89.0           62.0            0.0
0.0  22.5
```

```
496      10.0   101.0       76.0       48.0   180.0
32.9
497       2.0   122.0       70.0       27.0     0.0  36.8
498       5.0   121.0       72.0       23.0   112.0
26.2
499       1.0    93.0       70.0       31.0     0.0  30.4

     DiabetesPedigreeFunction    Age
0                       0.351   31.0
1                       0.167   21.0
2                       0.201   30.0
3                       0.134   29.0
4                       0.191   30.0
..                        ...    ...
495                     0.142   33.0
496                     0.171   63.0
497                     0.340   27.0
498                     0.245   30.0
499                     0.315   23.0

[500 rows x 8 columns]
```

Program 2: Read the data from two different folders. Automate the process. Avoid mentioning the paths of the data manually. Use built in functions of popular libraries for reading.

```
import pandas as pd

print("Which dataset would you like to read?\n0 All Reports\n1
Positive Reports\n-1 Negative Reports")
f = int(input("Input file number: "))
path = "diabetes.csv"
if f<0: path = "Negative/" + path
if f>0: path = "Positive/" + path
df = pd.read_csv(path)
print("\nViewing " + path)
print(df)
```

```
Which dataset would you like to read?
0 All Reports
1 Positive Reports
-1 Negative Reports
Input file number: -1

Viewing Negative/diabetes.csv
    Unnamed: 0  Pregnancies  Glucose  BloodPressure
SkinThickness  Insulin  \
0            0          1.0     85.0           66.0
29.0     0.0
1            1          1.0     89.0           66.0
23.0    94.0
2            2          5.0    116.0           74.0
0.0     0.0
3            3         10.0    115.0            0.0
0.0     0.0
4            4          4.0    110.0           92.0
0.0     0.0
..         ...          ...      ...            ...
...     ...
495        495          9.0     89.0           62.0
0.0     0.0
496        496         10.0    101.0           76.0
48.0   180.0
497        497          2.0    122.0           70.0
27.0     0.0
498        498          5.0    121.0           72.0
23.0   112.0
499        499          1.0     93.0           70.0
31.0     0.0

     BMI  DiabetesPedigreeFunction  Age
0   26.6                     0.351   31.0
1   28.1                     0.167   21.0
2   25.6                     0.201   30.0
3   35.3                     0.134   29.0
4   37.6                     0.191   30.0
..   ...                       ...    ...
495 22.5                     0.142   33.0
496 32.9                     0.171   63.0
497 36.8                     0.340   27.0
498 26.2                     0.245   30.0
499 30.4                     0.315   23.0

[500 rows x 9 columns]
```

Program 3: **View all the data columns and rows. See what can be features or affect the learning process according to point.**

```
import pandas as pd
f = open("diabetes.csv")
df = pd.read_csv("diabetes.csv")
print(df)
```

```
     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
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
..                        ...   ...
...
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

[768 rows x 9 columns]
```

```
import pandas as pd
df = pd.read_csv("diabetes.csv")
print(df.shape)
```

```
(768, 10)
```

```
import pandas as pd
f = open("diabetes.csv")
df = pd.read_csv("diabetes.csv")
print(df.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
None
```

```
import pandas as pd
df = pd.read_csv('diabetes.csv')

print(df.describe())
```

	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

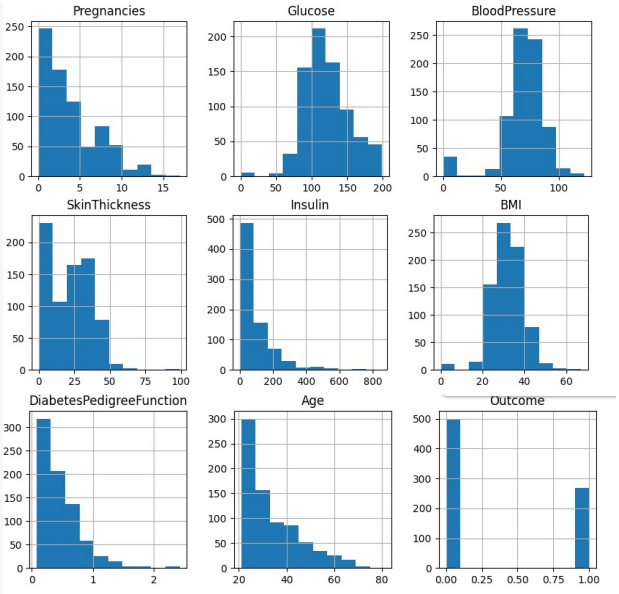
	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	31.992578	0.471876	33.240885	0.348958
std	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.078000	21.000000	0.000000
25%	27.300000	0.243750	24.000000	0.000000
50%	32.000000	0.372500	29.000000	0.000000
75%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

```
import pandas as pd
df = pd.read_csv('diabetes.csv')
df.hist(figsize = (10, 10))
```

```
array([[<Axes: title={'center': 'Pregnancies'}>,
        <Axes: title={'center': 'Glucose'}>,
        <Axes: title={'center': 'BloodPressure'}>],
       [<Axes: title={'center': 'SkinThickness'}>,
        <Axes: title={'center': 'Insulin'}>,
        <Axes: title={'center': 'BMI'}>],
       [<Axes: title={'center': 'DiabetesPedigreeFunction'}>,
        <Axes: title={'center': 'Age'}>,
        <Axes: title={'center': 'Outcome'}>]], dtype=object)
```

Or type following

```
array([[<Axes: title={'center':
'Pregnancies'}>,
        <Axes: title={'center': 'Glucose'}>,
        <Axes: title={'center':
'BloodPressure'}>],
       [<Axes: title={'center':
'SkinThickness'}>,
        <Axes: title={'center': 'Insulin'}>,
        <Axes: title={'center': 'BMI'}>],
       [<Axes: title={'center':
'DiabetesPedigreeFunction'}>,
        <Axes: title={'center': 'Age'}>,
        <Axes: title={'center': 'Outcome'}>]],
dtype=object)
```



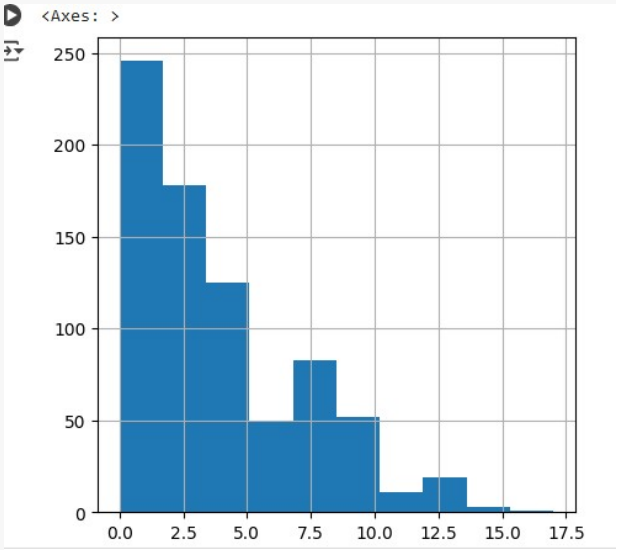
Program 4: **Find Print only pregnancy data.**

```
import pandas as pd
f = open("diabetes.csv")
df = pd.read_csv("diabetes.csv", usecols =
['Pregnancies'])
n = int(input("Enter the number of rows to
print: "))
print(df.head(n))
```

```
Enter the number of rows to print: 20
    Pregnancies
0             6
1             1
2             8
3             1
4             0
5             5
6             3
7            10
8             2
9             8
10            4
11           10
```

```
12              10
13               1
14               5
15               7
16               0
17               7
18               1
19               1
```

```
import pandas as pd
df = pd.read_csv('diabetes.csv')
df['Pregnancies'].hist(figsize = (5, 5))
```



Program 5: **Create different variants of the dataset:**
Print only 8 features (first)
What happens when we use -1 in feature space?
Print 100 learnable and seven dimensions of data.

```
import pandas as pd
df = pd.read_csv('diabetes.csv')
df = df[df.columns[1:9]]
print(df)
```

```
     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]
```

```
import pandas as pd
df = pd.read_csv('diabetes.csv')
print(df[-1:])
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
767                     0.315   23        0
```

```
import pandas as pd
df = pd.read_csv('diabetes.csv')
df = df[df.columns[1:8]]
df = df.head(100)
print(df)
```

```
    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
..          ...      ...            ...            ...      ...   ...
95            6      144             72             27      228  33.9
96            2       92             62             28        0  31.6
97            1       71             48             18       76  20.4
98            6       93             50             30       64  28.7
99            1      122             90             51      220  49.7

    DiabetesPedigreeFunction
0                      0.627
1                      0.351
2                      0.672
3                      0.167
4                      2.288
..                       ...
95                     0.255
96                     0.130
97                     0.323
```

Program 6: Store the datasets physically into disks numpy array files (in npy format) and print head tail, and shape of data. Use numpy to print data in array and print interleaved format.

```
import numpy as np
from google.colab import files

ds = np.genfromtxt('diabetes.csv',
delimiter=',')
np.save('diabetes.npy', ds)
files.download('diabetes.npy')



import numpy as np
np.set_printoptions(suppress=True)
```

```
ds = np.load('diabetes.npy')
print('HEAD')
print(ds[:5])
print('\nTAIL')
print(ds[-5:])
print('\nSHAPE')
print(ds.shape)
```

```
HEAD
[[  nan   nan   nan   nan   nan   nan   nan   nan   nan]
 [  6.   148.    72.    35.     0.    33.6    0.627  50.     1.  ]
 [  1.    85.    66.    29.     0.    26.6    0.351  31.     0.  ]
 [  8.   183.    64.     0.     0.    23.3    0.672  32.     1.  ]
 [  1.    89.    66.    23.    94.    28.1    0.167  21.     0.  ]]

TAIL
[[ 10.   101.    76.    48.   180.    32.9    0.171  63.     0.  ]
 [  2.   122.    70.    27.     0.    36.8    0.34   27.     0.  ]
 [  5.   121.    72.    23.   112.    26.2    0.245  30.     0.  ]
 [  1.   126.    60.     0.     0.    30.1    0.349  47.     1.  ]
 [  1.    93.    70.    31.     0.    30.4    0.315  23.     0.  ]]

SHAPE
(769, 9)
```

```
import numpy as np

ds = np.genfromtxt('diabetes.csv',
delimiter=',')
print(ds)
```

```
[[  nan   nan   nan ...   nan   nan   nan]
 [  6.   148.    72.  ...  0.627  50.    1.  ]
 [  1.    85.    66.  ...  0.351  31.    0.  ]
 ...
 [  5.   121.    72.  ...  0.245  30.    0.  ]
 [  1.   126.    60.  ...  0.349  47.    1.  ]
 [  1.    93.    70.  ...  0.315  23.    0.  ]]
```

AI Programming

[Day-3]