Machine Learning - Creating Dataset

Submitted By
Name: **Rathod Nishit Shailesh**
Register Number: **19112014**
Class: **5 BSc Data Science**

Create "Student Details Dataset" using Numpy and Pandas.

Question

1. Create a Dataframe with Following Fields: REGNO, AGE, YEAR, CLASS, DEPT, MARKS_PERC
2. Columns Description
   - For AGE, create a list with values ranging from 17 to 22
   - For YEAR, creata a list with values [1, 2, 3]
   - For CLASS, create a list with values: [BEA, BDS, BBA, BCOM]
   - For DEPT, create a list with [Data Science, Management, Commerce]
3. Reg No Rule:
   - I Year - 21 _ __
   - II Year - 20 _ __
   - III Year - 19 _ __
   - BEA - 1, BDS - 2, BBA - 3, BCOM - 4
   - Last Two Digits - Roll No (01 - 60)
4. Populate the Dataframe with Values
5. Write User Defined Functions to Add Students to the above courses.
6. The Min Age to be in First Year is 17, Second Year: 18, Third Year: 19
7. Marks Percentage is between 50% and 88%
8. Export it as a CSV

In [1]:
```python
import pandas as pd
import numpy as np
import random
```

In [2]:
```python
df = pd.DataFrame(columns = ['RegNo', 'Age', 'Year', 'Class', 'Dept', 'Marks_Perc'])
df
```

Out[2]:

RegNo	Age	Year	Class	Dept	Marks_Perc

In [3]:
```python
Age = [*range(17, 22, 1)]
Age
```

Out[3]:
```
[17, 18, 19, 20, 21]
```

In [4]:
```python
Year = [1, 2, 3]
Year
```

Out[4]:
```
[1, 2, 3]
```

In [5]:
```python
Class = ["BEA", "BDS", "BBA", "BCOM"]
Class
```

Out[5]:
```
['BEA', 'BDS', 'BBA', 'BCOM']
```

In [6]:
```python
Dept = ["Data Science", "Management", "Commerce"]
Dept
```

Out[6]:
```
['Data Science', 'Management', 'Commerce']
```

In [7]:
```python
for cls in Class: # Iterating over Class
    for yr in Year: # Iterating over Year

        # A Random Function to create number of Students in a Class
        Class_Strength = random.randrange(30, 60)

        # Creating Register Numbers for Students
        for Rno in range(1, Class_Strength):
            # First Two Digits
            YR = 22 - yr

            # Second Digit
            if cls == "BEA":
                CL = 1
                Dept = "Data Science"
            elif cls == "BDS":
                CL = 2
                Dept = "Data Science"
            elif cls == "BBA":
                CL = 3
                Dept = "Management"
            else:
                CL = 4
                Dept = "Commerce"

            # Making the Register Number
            RegNo = str(YR) + str(CL)+ str(Rno).zfill(2)

            # Age
            Age = random.randrange(16 + yr, 22)

            # Marks
            MarksPerc = round(random.uniform(50.0, 88.0), 2)

            Student = {}
            Student['RegNo'] = RegNo
            Student['Age'] = Age
            Student['Year'] = yr
            Student['Class'] = cls
            Student['Dept'] = Dept
            Student['Marks_Perc'] = MarksPerc

            df = df.append(Student, ignore_index = True)
```

In [8]:
```python
df.to_csv("StudentDetails.csv", index = False)
```

In [9]:
```python
df1 = pd.read_csv("StudentDetails.csv")
```

In [10]:
```python
df1.shape
```

Out[10]:
```
(500, 6)
```

In [11]:
```python
df1.columns
```

Out[11]:
```
Index(['RegNo', 'Age', 'Year', 'Class', 'Dept', 'Marks_Perc'], dtype='object')
```

In [12]:
```python
df1.dtypes
```

Out[12]:
```
RegNo           int64
Age             int64
Year            int64
Class          object
Dept           object
Marks_Perc    float64
dtype: object
```

In [13]:
```python
df1.astype({'RegNo': 'int64'}).dtypes
```

Out[13]:
```
RegNo           int64
Age             int64
Year            int64
Class          object
Dept           object
Marks_Perc    float64
dtype: object
```

In [14]:
```python
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   RegNo       500 non-null    int64
 1   Age         500 non-null    int64
 2   Year        500 non-null    int64
 3   Class       500 non-null    object
 4   Dept        500 non-null    object
 5   Marks_Perc  500 non-null    float64
dtypes: float64(1), int64(3), object(2)
memory usage: 23.6+ KB
```

In [15]:
```python
df1.head()
```

Out[15]:

	RegNo	Age	Year	Class	Dept	Marks_Perc
0	21101	21	1	BEA	Data Science	84.68
1	21102	19	1	BEA	Data Science	63.14
2	21103	21	1	BEA	Data Science	69.16
3	21104	17	1	BEA	Data Science	82.36
4	21105	20	1	BEA	Data Science	75.14

In [16]:
```python
df1.tail()
```

Out[16]:

	RegNo	Age	Year	Class	Dept	Marks_Perc
495	19444	20	3	BCOM	Commerce	57.10
496	19445	21	3	BCOM	Commerce	56.51
497	19446	20	3	BCOM	Commerce	81.60
498	19447	19	3	BCOM	Commerce	86.00
499	19448	19	3	BCOM	Commerce	54.44

In [17]:
```python
df1.describe()
```

Out[17]:

	RegNo	Age	Year	Marks_Perc
count	500.000000	500.000000	500.000000	500.000000
mean	20340.784000	19.432000	1.948000	69.627240
std	832.691198	1.221622	0.821368	11.089979
min	19101.000000	17.000000	1.000000	50.060000
25%	19417.750000	19.000000	1.000000	60.692500
50%	20323.500000	19.000000	2.000000	69.995000
75%	21219.250000	20.000000	3.000000	78.962500
max	21452.000000	21.000000	3.000000	87.990000

In [18]:
```python
df1.isna().sum()
```

Out[18]:
```
RegNo         0
Age           0
Year          0
Class         0
Dept          0
Marks_Perc    0
dtype: int64
```

Create "Clinic Details Dataset" using Numpy and Pandas.

Question

Dataset Structure: PATIENT_ID | TYPE | DEPARTMENT | EMERGENCY | DATE | BILL

1. Each Patient will have a Unique ID, based on the below conditions.
   - There are four sections:
   - First two Characters are either OP or IP
   - Second three Characters will show the Department
   - Third Section shows the Date
   - Fourth Section shows the Serial Number Eg: IP-ORT-10FEB21-001
2. Departments will be: General, Ortho, Neuro, Opthal.
3. Emergencies have either values: YES or NO
4. Dates are of the form: DD/MM/YYYY
5. Bill Amount Varies for each Department: Rs 100 - 600 for General Rs 300 - 1000 for Ortho Rs 500 - 1500 for Neuro Rs 200 - 400 for Opthal

Other Notes:

- No In-patients will have Emergencies
- It is observed that 20% of the cases in General Ward is Emergency, no other wards have emergencies
- Generally around 50 - 100 people visits each department a day

WRITE THE FUNCTION TO GENERATE THE DATASET FOR 7 DAYS, FROM ANY USER-INPUT DAY.

Sample Input Output

- Enter a Day: 10
- Enter a Month: FEB
- Enter a Year: 2021

PATIENT_ID | TYPE | DEPARTMENT | EMERGENCY | DATE | BILL

- IP-ORT-10FEB21-001 | In-Patient | Ortho | No | 10/02/2021 | Rs 592.00
- OP-GEN-10FEB21-001 | Out-Patient | General | Yes | 10/02/2021 | Rs 200.00

It should go upto FEB 16 (7 Days from Feb 10)

In [19]:
```python
import pandas as pd
import numpy as np
import random
from random import randrange
```

In [20]:
```python
cd = pd.DataFrame(columns = ['Patientid', 'Type', 'Department', 'Emergency', 'Date', 'Bill'])
cd
```

Out[20]:

Patientid	Type	Department	Emergency	Date	Bill

In [21]:
```python
Type = ["Out-Patient","In-Patient"]
Type
```

Out[21]:
```
['Out-Patient', 'In-Patient']
```

In [22]:
```python
Department = ["General", "Ortho", "Neuro", "Opthal"]
Department
```

Out[22]:
```
['General', 'Ortho', 'Neuro', 'Opthal']
```

In [23]:
```python
Emergency = ["Yes", "No"]
Emergency
```

Out[23]:
```
['Yes', 'No']
```

In [24]:
```python
Date = ["10/02/2021","11/02/2021","12/02/2021","13/02/2021","14/02/2021","15/02/2021","16/02/2021"]
Date
```

Out[24]:
```
['10/02/2021',
 '11/02/2021',
 '12/02/2021',
 '13/02/2021',
 '14/02/2021',
 '15/02/2021',
 '16/02/2021']
```

In [25]:
```python
for Dept in Department: # Iterating over Department
    for date in Date: # Iterating over Date
        for typ in Type: # Iterating over Type

            # Creating Patient Id
            Client_Strength = random.randrange(50, 100)
            for Pid in range(1, Client_Strength):

                if typ == "Out-Patient":
                    TP = "OP-"
                else:
                    TP = "IP-"

                if Dept == "General":
                    DP = "GEN-"
                    Bill = random.randrange(100, 600)
                elif Dept == "Ortho":
                    DP = "ORT-"
                    Bill = random.randrange(300, 1000)
                elif Dept == "Neuro":
                    DP = "NEU-"
                    Bill = random.randrange(500, 1500)
                else:
                    DP = "OPT-"
                    Bill = random.randrange(200, 400)

                if date == "10/02/2021":
                    dt = "10FEB2021-"
                elif date == "11/02/2021":
                    dt = "11FEB2021"
                elif date == "12/02/2021":
                    dt = "12FEB2021-"
                elif date == "13/02/2021":
                    dt = "13FEB2021-"
                elif date == "14/02/2021":
                    dt = "14FEB2021-"
                elif date == "15/02/2021":
                    dt = "15FEB2021-"
                elif date == "16/02/2021":
                    dt = "16FEB2021-"

                # Making the Patient Id.
                Patientid = str(TP) + str(DP)+ str(dt) + str(Pid).zfill(3)

                #Figuring out the Emergency Case
                for Emer in Emergency:
                    if typ == "In-Patient":
                        Emer = "No"
                    if typ == "Out-Patient":
                        Emer = "Yes"


                Patient = {}
                Patient['Patientid'] = Patientid
                Patient['Type'] = typ
                Patient['Department'] = Dept
                Patient['Emergency'] = Emer
                Patient['Date'] = date
                Patient['Bill'] = Bill

                cd = cd.append(Patient, ignore_index = True)
```

In [26]:
```python
cd.head()
```

Out[26]:

	Patientid	Type	Department	Emergency	Date	Bill
0	OP-GEN-10FEB2021-001	Out-Patient	General	Yes	10/02/2021	562
1	OP-GEN-10FEB2021-002	Out-Patient	General	Yes	10/02/2021	534
2	OP-GEN-10FEB2021-003	Out-Patient	General	Yes	10/02/2021	432
3	OP-GEN-10FEB2021-004	Out-Patient	General	Yes	10/02/2021	127
4	OP-GEN-10FEB2021-005	Out-Patient	General	Yes	10/02/2021	170

In [27]:
```python
cd.tail()
```

Out[27]:

	Patientid	Type	Department	Emergency	Date	Bill
3987	IP-OPT-16FEB2021-089	In-Patient	Opthal	No	16/02/2021	275
3988	IP-OPT-16FEB2021-090	In-Patient	Opthal	No	16/02/2021	213
3989	IP-OPT-16FEB2021-091	In-Patient	Opthal	No	16/02/2021	251
3990	IP-OPT-16FEB2021-092	In-Patient	Opthal	No	16/02/2021	338
3991	IP-OPT-16FEB2021-093	In-Patient	Opthal	No	16/02/2021	299

In [28]:
```python
cd.to_csv("ClientDetails.csv", index = False)
```

In [29]:
```python
cd1 = pd.read_csv("ClientDetails.csv")
```

In [30]:
```python
cd1.shape
```

Out[30]:
```
(3992, 6)
```

In [31]:
```python
cd1.columns
```

Out[31]:
```
Index(['Patientid', 'Type', 'Department', 'Emergency', 'Date', 'Bill'], dtype='object')
```

In [32]:
```python
cd1.sample(5)
```

Out[32]:

	Patientid	Type	Department	Emergency	Date	Bill
1652	IP-ORT-14FEB2021-036	In-Patient	Ortho	No	14/02/2021	955
3228	IP-OPT-11FEB2021074	In-Patient	Opthal	No	11/02/2021	354
3701	OP-OPT-15FEB2021-013	Out-Patient	Opthal	Yes	15/02/2021	329
2956	OP-OPT-10FEB2021-031	Out-Patient	Opthal	Yes	10/02/2021	345
3053	IP-OPT-10FEB2021-037	In-Patient	Opthal	No	10/02/2021	235

In [33]:
```python
cd1.dtypes
```

Out[33]:
```
Patientid     object
Type          object
Department    object
Emergency     object
Date          object
Bill           int64
dtype: object
```

In [34]:
```python
cd1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3992 entries, 0 to 3991
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Patientid   3992 non-null   object
 1   Type        3992 non-null   object
 2   Department  3992 non-null   object
 3   Emergency   3992 non-null   object
 4   Date        3992 non-null   object
 5   Bill        3992 non-null   int64
dtypes: int64(1), object(5)
memory usage: 187.2+ KB
```

In [35]:
```python
cd1.head(5)
```

Out[35]:

	Patientid	Type	Department	Emergency	Date	Bill
0	OP-GEN-10FEB2021-001	Out-Patient	General	Yes	10/02/2021	562
1	OP-GEN-10FEB2021-002	Out-Patient	General	Yes	10/02/2021	534
2	OP-GEN-10FEB2021-003	Out-Patient	General	Yes	10/02/2021	432
3	OP-GEN-10FEB2021-004	Out-Patient	General	Yes	10/02/2021	127
4	OP-GEN-10FEB2021-005	Out-Patient	General	Yes	10/02/2021	170

In [36]:
```python
cd1.describe()
```

Out[36]:

	Bill
count	3992.000000
mean	567.356713
std	344.279481
min	100.000000
25%	303.000000
50%	447.500000
75%	786.250000
max	1499.000000

In [37]:
```python
def main():
    print("====================================")
    print("\t\tMenu")
    print("====================================")
    print("Enter 1 To Search.")
    print("Enter 0 To Terminate the Search.")
    print("====================================")
    while True:
        loop = int(input("Enter your choice from the above menu?:"))
        if loop == 0:
            print("Thank you for searching.")
            break

        if loop == 1:
            print("-----------------------------------------------------")
            print("Note: Search by date between 10/02/2021 to 16/02/2021")
            print("=====================================================")
            date = str(input('Search by date in dd/mm/yyyy formate:'))
            cd2 = cd1[cd1['Date'].str.contains(date)]
            print(cd2)
main()
```

```
====================================
                Menu
====================================
Enter 1 To Search.
Enter 0 To Terminate the Search.
====================================
Enter your choice from the above menu?:1
-----------------------------------------------------
Note: Search by date between 10/02/2021 to 16/02/2021
=====================================================
Search by date in dd/mm/yyyy formate:13/02/2021
                 Patientid         Type Department Emergency        Date  Bill
460   OP-GEN-13FEB2021-001  Out-Patient    General       Yes  13/02/2021   168
461   OP-GEN-13FEB2021-002  Out-Patient    General       Yes  13/02/2021   574
462   OP-GEN-13FEB2021-003  Out-Patient    General       Yes  13/02/2021   302
463   OP-GEN-13FEB2021-004  Out-Patient    General       Yes  13/02/2021   398
464   OP-GEN-13FEB2021-005  Out-Patient    General       Yes  13/02/2021   156
...                    ...          ...        ...       ...         ...   ...
3521  IP-OPT-13FEB2021-079   In-Patient     Opthal        No  13/02/2021   255
3522  IP-OPT-13FEB2021-080   In-Patient     Opthal        No  13/02/2021   211
3523  IP-OPT-13FEB2021-081   In-Patient     Opthal        No  13/02/2021   292
3524  IP-OPT-13FEB2021-082   In-Patient     Opthal        No  13/02/2021   204
3525  IP-OPT-13FEB2021-083   In-Patient     Opthal        No  13/02/2021   355

[569 rows x 6 columns]
Enter your choice from the above menu?:1
-----------------------------------------------------
Note: Search by date between 10/02/2021 to 16/02/2021
=====================================================
Search by date in dd/mm/yyyy formate:15/02/2021
                 Patientid         Type Department Emergency        Date  Bill
760   OP-GEN-15FEB2021-001  Out-Patient    General       Yes  15/02/2021   501
761   OP-GEN-15FEB2021-002  Out-Patient    General       Yes  15/02/2021   444
762   OP-GEN-15FEB2021-003  Out-Patient    General       Yes  15/02/2021   105
763   OP-GEN-15FEB2021-004  Out-Patient    General       Yes  15/02/2021   280
764   OP-GEN-15FEB2021-005  Out-Patient    General       Yes  15/02/2021   392
...                    ...          ...        ...       ...         ...   ...
3811  IP-OPT-15FEB2021-052   In-Patient     Opthal        No  15/02/2021   297
3812  IP-OPT-15FEB2021-053   In-Patient     Opthal        No  15/02/2021   272
3813  IP-OPT-15FEB2021-054   In-Patient     Opthal        No  15/02/2021   323
3814  IP-OPT-15FEB2021-055   In-Patient     Opthal        No  15/02/2021   389
3815  IP-OPT-15FEB2021-056   In-Patient     Opthal        No  15/02/2021   355

[540 rows x 6 columns]
Enter your choice from the above menu?:0
Thank you for searching.
```

Thank you