WEEK- 7: Applications-Practice:Iris dataset from scikit learn perform data exploration,preprocessing and splitting

Session No.3

# Data Cleaning With Python Pandas

In [7]:

**import**

numpy

**as**

np

**import**

pandas

**as**

pd

**import**

seaborn

**as**

sns

**import**

os

1

2

3

4

[8]:

1 print(os.listdir())

['.anaconda', '.bash\_history', '.conda', '.condarc', '.continuum', '.gitconf ig', '.idea', '.ipynb\_checkpoints', '.ipython', '.jupyter', '.lesshst', '.ma tplotlib', '.packettracer', '.viminfo', '.VirtualBox', '.vscode', '1st inter nal.ipynb', '87.py', 'aiml', 'AIML files', 'AIML\_CIE1-2.b.ipynb', 'anaconda 3', 'anakonda', 'AppData', 'Application Data', 'area.py', 'Assignment Week 4 & 5.ipynb', 'Atlassian', 'BFS.py', 'BOSTON\_KERAS.ipynb', 'Cal.csv', 'calc.p y', 'Cars Pro.ipynb', 'Cars Program.ipynb', 'CIE 2.b Ans.ipynb', 'CIE 2.ipyn b', 'CIE 3 Question Paper.ipynb', 'CIE-2.b.ipynb', 'CIE-3.ipynb', 'Cisco Pac ket Tracer 8.1.1', 'Company\_web', 'Confusion matrix and Accuracy.ipynb', 'Co ntacts', 'Cookies', 'Cross validation 1.ipynb', 'Data Integration 4Week.ipyn b', 'DataVisualization MATPLOTLIB.ipynb', 'DC with PP.ipynb', 'Decision Tre

e.ipynb', 'DFS.py', 'Documents', 'Downloads', 'Dtree BreastCancer.ipynb', 'E mp1.py', 'Emp11.py', 'Emp2.py', 'Emp3.py', 'Emp4.py', 'Emp5.py', 'Emp6.py', 'Emp7.py', 'Emp8.py', 'Emp9.py', 'ex.py', 'exp.py', 'exp1.py', 'exp2.py', 'e xp3.py', 'Factorial.py', 'Favorites', 'Fibonacci.py', 'first python.py', 'fi rst.py', 'Geometry.py', 'Grouping pandas .ipynb', 'Hash.py', 'hello.py.ipyn b', 'hello.txt', 'import libraries.py', 'IntelGraphicsProfiles', 'K-means Cl ustering.ipynb', 'LinearRegression.ipynb', 'LinearRegression1.ipynb', 'Linke dList.py', 'LinkedList1.py', 'Links', 'Local Settings', 'Logistic Regressio

n.ipynb', 'main.py', 'MediaGet2', 'ML Library.ipynb', 'Movie\_data.ipynb', 'M TCars.csv File.ipynb', 'Multiple Linear Regression.ipynb', 'Music', 'My Docu ments', 'NetHood', 'New Microsoft Excel Worksheet.xlsx', 'New Microsoft Word

Document.docx', 'NTUSER.DAT', 'ntuser.dat.LOG1', 'ntuser.dat.LOG2', 'NTUSER.

DAT{1c2b59c6-c5f5-11eb-bacb-000d3a96488e}.TM.blf', 'NTUSER.DAT{1c2b59c6-c5f5

df

**=**

pd

.

read\_csv

(

"C:\\Users\\maths\\aiml\\flights.csv"

)

1

-11eb-bacb-000d3a96488e}.TMContainer00000000000000000001.regtrans-ms', 'NTUS

ER.DAT{1c2b59c6-c5f5-11eb-bacb-000d3a96488e}.TMContainer0000000000000000000

2.regtrans-ms', 'ntuser.ini', 'Numpy DataFrame.ipynb', 'Numpy Moduls.ipynb',

'OneDrive', 'Pandas DataFrame.ipynb', 'pictures1.py', 'Polynomial Regressio

n.ipynb', 'Precision, Recall, F1 Score.ipynb', 'PrintHood', 'PriorityQueue.p y', 'PycharmProjects', 'python.py', 'python1.py', 'python2.py', 'python3.p y', 'python4.py', 'python5.py', 'python6.py', 'python7.py', 'python8. py.tx t', 'python9.py', 'Queue.py', 'Random Forest.ipynb', 'Recent', 'Reg no.43.ip ynb', 'Regression Matrics.ipynb', 'Saved Games', 'seaborn-data', 'Searches', 'SendTo', 'sh.py.ipynb', 'Shru', 'shru.DB', 'shru.main.py', 'shru.num.py', 'shru.py', 'shru.set.py', 'shru.tuple.py', 'shru1.py', 'shrushti.py', 'Simpl e Linear Regression .ipynb', 'skill test.py', 'sonu.DB', 'sonu.py', 'stack.p y', 'stack\_main.py', 'Start Menu', 'stu.py', 'Support Vector Machine.ipynb', 'Templates', 'testrepo', 'Time Series.ipynb', 'ubuntu-2022-07-10-14-26-58.lo g', 'Univariate Pro.ipynb', 'Untitled Folder', 'Untitled.ipynb', 'Untitled1 0.ipynb', 'Untitled11.ipynb', 'Untitled12.ipynb', 'Untitled13.ipynb', 'Untit led14.ipynb', 'Untitled15.ipynb', 'Untitled16.ipynb', 'Untitled17.ipynb', 'U ntitled18.ipynb', 'Untitled19.ipynb', 'Untitled2.ipynb', 'Untitled20.ipynb', 'Untitled21.ipynb', 'Untitled22.ipynb', 'Untitled23.ipynb', 'Untitled24.ipyn b', 'Untitled25.ipynb', 'Untitled26.ipynb', 'Untitled27.ipynb', 'Untitled28. ipynb', 'Untitled29.ipynb', 'Untitled3.ipynb', 'Untitled30.ipynb', 'Untitled 31.ipynb', 'Untitled32.ipynb', 'Untitled33.ipynb', 'Untitled4.ipynb', 'Untit led5.ipynb', 'Untitled6.ipynb', 'Untitled7.ipynb', 'Untitled8.ipynb', 'Untit led9.ipynb', 'usermodule.py', 'Videos', 'VirtualBox VM', 'VirtualBox VMs', 'VirtualBox VMs1', 'volume.py', 'Week-6.ipynb', '\_\_init\_.py']

In [9]:

[10]:

1 df

Out[10]:

**Unnamed: 0 year month passenger**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | 0 | 1949.0 | January | 112.0 |
| **1** | 1 | NaN | February | 118.0 |
| **2** | 2 | 1949.0 | March | NaN |
| **3** | 3 | 1949.0 | April | 129.0 |
| **4** | 4 | 1949.0 | May | 121.0 |
| **5** | 5 | 1949.0 | June | 113.0 |
| **6** | 6 | 1949.0 | July | 124.0 |
| **7** | 7 | 1949.0 | August | 126.0 |
| **8** | 8 | 1949.0 | Septmber | 132.0 |
| **9** | 9 | 1949.0 | Octomber | 116.0 |
| **10** | 10 | NaN | November | 114.0 |
| **11** | 11 | 1949.0 | December | 117.0 |

In [11]:

df

.

isnull

().

sum

()

1

Out[11]:

Unnamed: 0 0 year 2 month 0 passenger 1 dtype: int64

# Handling the program

**Step 1: Detecting NA N/A and na Values**

In [14]:

1. missing\_value**=**["N/a","na",np.nan]
2. df**=**pd.read\_csv("C:\\Users\\maths\\aiml\\flights.csv",na\_values**=**missing\_value) [15]:

1 df.isnull().sum()

Out[15]:

Unnamed: 0 0 year 2 month 0 passenger 1 dtype: int64 In [16]:

df

.

isnull

().

any

()

1

Out[16]:

Unnamed: 0 False year True month False passenger True dtype: bool In [19]:

sns

.

heatmap

(

df

.

isnull

()

,

yticklabels

**=**

**False**

,

annot

**=**

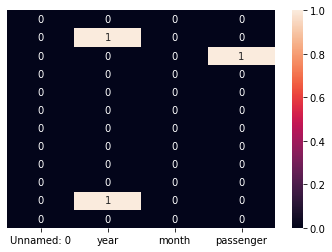
**True**

)

1

Out[19]:

<AxesSubplot:>



# Step 2: Lets learn how to to Remove this Values

In

[25]:

df1

**=**

pd

.

DataFrame

(

data

**=**

{

"year"

:[

1

,

np

.

nan

,

3

,

2

,

3

]

,

"month"

:[

22

,

np

.

nan

,

2

,

np

.

nan

,

22

]

})

1

2

3

4

[26]:

1df1

Out[26]:

**year**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **1** NaN | NaN |
| **2** 3.0 | 2.0 |
| **3** 2.0 | NaN |
| **4** 3.0 | 22.0 |

In [28]:

df1

.

dropna

()

1

Out[28]:

**year month**

**0** 1.0 22.0

**2** 3.0 2.0

**4** 3.0 22.0

In [29]:

df1

.

dropna

(

how

**=**

'all'

)

1

Out[29]: **year month**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **2** 3.0 | 2.0 |
| **3** 2.0 | NaN |
| **4** 3.0 | 22.0 |

[30]:

1df1

Out[30]:

**year**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **1** NaN | NaN |
| **2** 3.0 | 2.0 |
| **3** 2.0 | NaN |
| **4** 3.0 | 22.0 |

In [31]:

df1

.

fillna

(

0

)

1

Out[31]: **year month**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **1** 0.0 | 0.0 |
| **2** 3.0 | 2.0 |
| **3** 2.0 | 0.0 |
| **4** 3.0 | 22.0 |

In [32]:

*# Forward fill*

df1

.

fillna

(

method

**=**

'ffill'

)

1

2

Out[32]: **year month**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **1** 1.0 | 22.0 |
| **2** 3.0 | 2.0 |
| **3** 2.0 | 2.0 |
| **4** 3.0 | 22.0 |

[33]:

1df1

Out[33]:

**year**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **1** NaN | NaN |
| **2** 3.0 | 2.0 |
| **3** 2.0 | NaN |
| **4** 3.0 | 22.0 |

In [34]:

*# Backward fill*

df1

.

fillna

(

method

**=**

'bfill'

)

1

2

Out[34]: **year month**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **1** 3.0 | 2.0 |
| **2** 3.0 | 2.0 |
| **3** 2.0 | 22.0 |
| **4** 3.0 | 22.0 |

In [35]:

df1

.

interpolate

()

1

Out[35]: **year month**

|  |  |
| --- | --- |
| **0** 1.0 | 22.0 |
| **1** 2.0 | 12.0 |
| **2** 3.0 | 2.0 |
| **3** 2.0 | 12.0 |
| **4** 3.0 | 22.0 |

In [36]:

df\_drop

**=**

df

.

dropna

()

1

[37]: 1 df\_drop Out[37]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | 0 | 1949.0 | January | 112.0 |
| **1** | 1 | NaN | February | 118.0 |
| **2** | 2 | 1949.0 | March | NaN |
| **3** | 3 | 1949.0 | April | 129.0 |
| **4** | 4 | 1949.0 | May | 121.0 |
| **5** | 5 | 1949.0 | June | 113.0 |
| **6** | 6 | 1949.0 | July | 124.0 |
| **7** | 7 | 1949.0 | August | 126.0 |
| **8** | 8 | 1949.0 | Septmber | 132.0 |
| **9** | 9 | 1949.0 | Octomber | 116.0 |
| **10** | 10 | NaN | November | 114.0 |
| **11** | 11 | 1949.0 | December | 117.0 |

**Unnamed: 0 year month passenger**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | 0 | 1949.0 | January | 112.0 |
| **3** | 3 | 1949.0 | April | 129.0 |
| **4** | 4 | 1949.0 | May | 121.0 |
| **5** | 5 | 1949.0 | June | 113.0 |
| **6** | 6 | 1949.0 | July | 124.0 |
| **7** | 7 | 1949.0 | August | 126.0 |
| **8** | 8 | 1949.0 | Septmber | 132.0 |
| **9** | 9 | 1949.0 | Octomber | 116.0 |
| **11** | 11 | 1949.0 | December | 117.0 |

In [38]:

df

1

Out[38]:

**Unnamed: 0 year month passenger**

[39]:

Out[39]:

df

.

fillna

({

'year'

:

232323

})

1

2

3

**Unnamed: 0 year month passenger**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | 0 | 1949.0 | January | 112.0 |
| **1** | 1 | 232323.0 | February | 118.0 |
| **2** | 2 | 1949.0 | March | NaN |
| **3** | 3 | 1949.0 | April | 129.0 |
| **4** | 4 | 1949.0 | May | 121.0 |
| **5** | 5 | 1949.0 | June | 113.0 |
| **6** | 6 | 1949.0 | July | 124.0 |
| **7** | 7 | 1949.0 | August | 126.0 |
| **8** | 8 | 1949.0 | Septmber | 132.0 |
| **9** | 9 | 1949.0 | Octomber | 116.0 |
| **10** | 10 | 232323.0 | November | 114.0 |
| **11** | 11 | 1949.0 | December | 117.0 |

In [ ]:

**Training and Testing Data**

In

[74]:

Out[74]:

**import**

pandas

**as**

pd

df

**=**

pd

.

read\_csv

(

"C:\\Users\\maths\\aiml\\carPrice.csv"

)

df

.

head

()

1

2

3

**Mileage Age(yrs) Sell Price($)**

|  |  |  |
| --- | --- | --- |
| **0** 69000 | 6 | 18000 |
| **1** 35000 | 3 | 34000 |
| **2** 57000 | 5 | 26100 |
| **3** 225000 | 2 | 40000 |
| **4** 46000 | 4 | 31500 |

In

[75]:

In

[76]:

Out[76]:

**import**

matplotlib

.

pyplot

**as**

plt

**%**

matplotlib

inline

plt

.

scatter

(

df

[

'Mileage'

]

,

df

[

'Sell Price($)'

])

1

2

1

2

In

[77]:

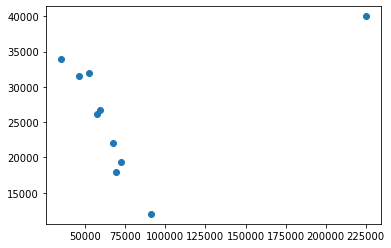
In

[78]:

<

matplotlib.collections.PathCollection at

0x2297842d130>



Out[78]:

x

**=**

df

[[

'Mileage'

,

'Age(yrs)'

]]

y

**=**

df

[

'Sell Price($)'

]

x

1

2

1

**Mileage Age(yrs)**

|  |  |
| --- | --- |
| **0** 69000 | 6 |
| **1** 35000 | 3 |
| **2** 57000 | 5 |
| **3** 225000 | 2 |
| **4** 46000 | 4 |
| **5** 59000 | 5 |
| **6** 52000 | 5 |
| **7** 72000 | 6 |
| **8** 91000 | 8 |
| **9** 67000 | 6 |

In [79]:

y

Out[79]:

1. 18000
2. 34000
3. 26100
4. 40000
5. 31500
6. 26750
7. 32000
8. 19300
9. 12000
10. 22000

Name: Sell Price($), dtype: int64 In [80]:

**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

)

In

[81]:

X: 10

X\_train: 8

print

(

"X:"

,

len

(

x

))

print

(

"X\_train:"

,

len

(

x\_train

))

print

(

"X\_test:"

,

len

(

x\_test

))

X\_test: 2 In [82]:

x\_train

Out[82]: **Mileage Age(yrs)**

**8** 91000 8

**2** 57000 5

**5** 59000 5

**0** 69000 6

**9** 67000 6

**3** 225000 2

**7** 72000 6

**6** 52000 5

In [83]:

x\_test

Out[83]: **Mileage Age(yrs)**

**4** 46000 4

**1** 35000 3

In [84]:

y\_train

Out[84]:

8 12000

2 26100

5 26750

0 18000

9 22000

3 40000

7 19300

6 32000

Name: Sell Price($), dtype: int64 In [85]:

y\_test

Out[85]:

4 31500

1 34000

Name: Sell Price($), dtype: int64 In