SHETH L.U.J. & SIR M.V. COLLEGE

Rajanish bhardwaj | T073

Practical No. 2

Aim: Data Frames and Basic Data Pre-processing

- Read data from CSV and JSON files into a data frame.
- Perform basic data pre-processing tasks such as handling missing values and outliers.
- Manipulate and transform data using functions like filtering, sorting, and grouping.

2.1 Loading a Sample Dataset

Problem :- You want to load a prexisting sample dataset

```
from sklearn import datasets

digits = datasets.load_digits()

features = digits.data
target = digits.target
features[0]

```

```
array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
       15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
       12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
        0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
       10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])
```

2.2 Creating a Simulated Dataset

Problem :- You need to generate a dataset of simulated data

```
from sklearn.datasets import make_regression

features, target, coefficients = make_regression(n_samples = 100,
                                                 n_features = 3,
                                                 n_informative= 3,
                                                   n_targets = 1,
                                                   noise = 0.0,
                                                   coef = True,
                                                   random_state = 1)
print("feature matrix \n{}".format(features[:3]))
print("target vector \n{}".format(target[:3]))
```

```
feature matrix
[[ 1.29322588 -0.61736206 -0.11044703]
 [-2.793085    0.36633201  1.93752881]
 [ 0.80186103 -0.18656977  0.0465673 ]]
target vector
[-10.37865986  25.5124503   19.67705609]
```

```
from sklearn.datasets import make_classification
features, target = make_classification(n_samples=100,
                                       n_features=3,
                                       n_informative=3,
                                       n_redundant=0,
                                       n_classes=2,
                                       weights=[.25,.75],
                                       random_state=1)
print("feature matrix \n{}".format(features[:3]))
print("target vector \n{}".format(target[:3]))
```

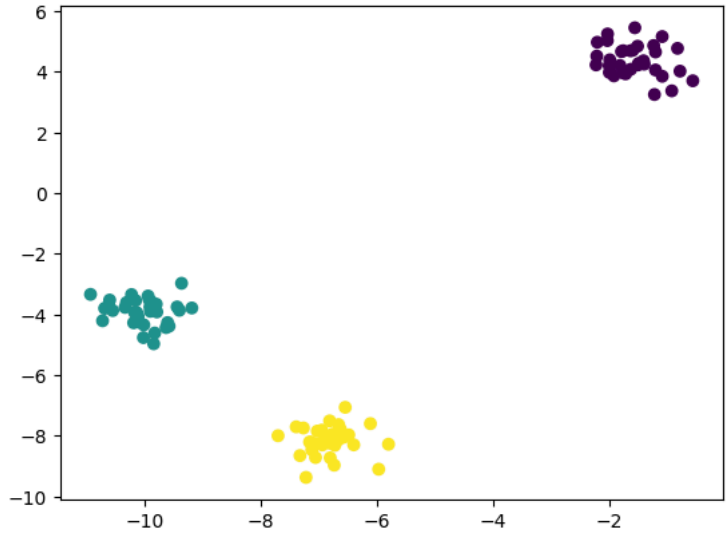
```
feature matrix
[[ 1.06354768 -1.42632219  1.02163151]
 [ 0.23156977  1.49535261  0.33251578]
 [ 0.15972951  0.83533515 -0.40869554]]
target vector
[1 0 0]
```

```
from sklearn.datasets import make_blobs
features, target = make_blobs(n_samples=100,
                              n_features=2,
                              centers=3,
                              cluster_std=0.5,
                              shuffle=True,
                              random_state=1)
```

```
print("feature matrix \n{}".format(features[:3]))
print("target vector \n{}".format(target[:3]))

import matplotlib.pyplot as plt
plt.scatter(features[:,0], features[:,1], c=target)
plt.show()
```

```
feature matrix
[[ -1.22685609   3.25572052]
 [ -9.57463218  -4.38310652]
 [-10.71976941  -4.20558148]]
target vector
[0 1 1]
```



2.3 Loading a CSV File

Problem :- You need to import a comma-separated values (CSV) file.

```
import pandas as pd

df=pd.read_csv("College_Marks_Dataset.csv")

df.head(2)
```

	Student_ID	Name	Class	SSC_Marks	HSC_Marks	College_Marks	Attendance_Percentage	Grade
0	S1000	Student_0	Commerce	535	452	692	84.71	C
1	S1001	Student_1	Commerce	494	535	551	81.99	D

2.4 Loading an Excel File

Problem :- You need to import an Excel spreadsheet

```
import pandas as pd
df=pd.read_excel("exceldata.xlsx")
df.head(2)
```

	1	rajnish
0	2	gautam
1	3	ram

2.5 Loading a JSON File

Problem :- You need to load a JSON file for data preprocessing

```
import pandas as pd
df=pd.read_json("iris.json")
df.head(2)
```

	sepalLength	sepalWidth	petalLength	petalWidth	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa

2.6 Querying a SQL Database

Problem :- You need to load data from a databaseu sing structured query language

```
 1 import pandas as pd
 2 import sqlite3
 3
 4 con = sqlite3.connect("sqlite-sakila.db")
 5
 6 df = pd.read_sql_query("SELECT * FROM users", con)
 7 print("Data in 'users' table:")
 8 display(df)
 9
10 con.close()
```

```
Data in 'users' table:
```

	id	name
0	1	rajnish
1	2	gautam
2	3	ram