Data Science Using Python

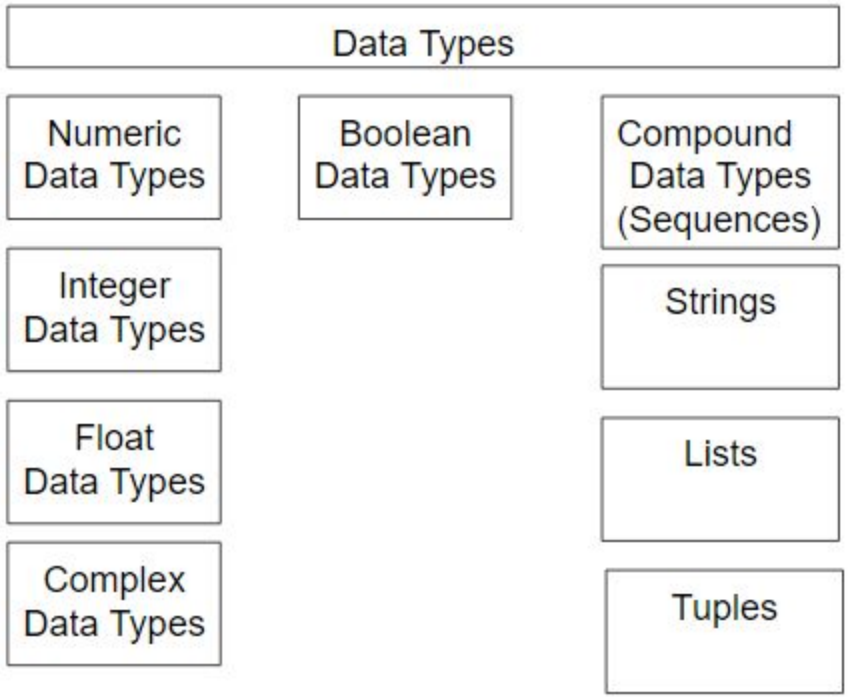
Module 1: Data Preprocessing

1. Missing Data
2. Categorical Data
3. Splitting data into training and testing set
4. Feature Scaling
5. Data Preprocessing Template

Module 2: ML for Data Science

1. **Regression**
2. **Classification**
3. **Clustering**
4. **Association Rule Mining**

#Data Types



 # Numeric Data Types

```
#Integer
0xA    # Hexadecimal Literals (0-9 and A-F)
Output: 10
```

```
0o10     # Octal Literals (digits 0-7)
Output : 8
0B1011   # Binary Literals (0 - 1)
Output : 11

#Float
1.2597 # Representing float data types
52.05-1.25 # Subtracting two float nos.
5.1e3   # representing exponent

#Complex
3+-7j
2.5+3.65j
X=5+7j
Y=0.2-7j
Z=X+Y
print (Z)
```

#Boolean Datatype

```
print(6>4)
print("F" not in "Python")
print(0==4)
```

#Compound Datatype

String Data Type

- A contiguous set of characters enclosed in the quotation marks is identified as a string.

List Data Types

- It is a collection of randomly typed objects with no fixed size and types.

Tuples Data Types

- It is an ordered sequence of elements or values.

#String Datatype

```
data="pqrstu"
data[1:4]
Output : qrs
```

Index from rear	-6	-5	-4	-3	-2	-1
Index from front	0	1	2	3	4	5
	p	q	r	s	t	u
Slice from front	1	2	3	4	5	:
Slice from rear	-5	-4	-3	-2	-1	:

```
Example_string=”String”
print(Example_string)          # Printing the value of variable;  Output: String
print(Example_string[2])       # printing value of index 2;          Output: r
print(Example_string[2:4])   # slicing the variable;                 Output: ri
```

#List Data Types

```
MyList = [10, 28.5, ‘Python’)
Example_list = [11, 22, 33, 44, 55]
print(Example_list)           # Printing the list;       Output : [11, 22, 33, 44, 55]
print(Example_list[2])       # Printing list index 2;  Output : 33
print(Example_list[1:4])    # Slicing the list;          Output : [22,33,44]
```

#Tuple Data Type

```
Example_tuple=(45,’Python’, 78.08)  # Creating a tuple
print (Example_tuple)       # Printing the tuple;       Output:(45,’Python’, 78.08)
print (Example_tuple[2])   # Printing tuple index 2; Output: 78.08
print(Example_tuple[0:1]) # Slicing the list;            Output: (45,)
```

#Program: Swapping two numbers using tuple assignment

```
num1 = 67                          # First number
num2 = 78                          # Second number
num1, num2 = num2, num1    # Tuple assignment
print(num1)                    # Printing first number after swapping;  Output: 78
print(num2)                  # Printing second number after swapping; Output: 67
```

#Dictionary

A dictionary in Python is an unordered set of items.
They are accessed using keys and not their positions.

```
MyDict = {}     # Empty Dictionary
MyDict[‘one’] = “This is Python”
MyDict[2] = “This is Python again”
Dict = {‘Roll No’ : 1, ‘Name’ : ‘Smith’ , ‘Age’ : 23}    # Creating dictionary
print(MyDict[‘one’])     # Prints value for ‘one’ key; Output: This is Python
```

print(MyDict[2]) # Prints value for 2 key; Output: This is Python again
print(Dict) # Prints Dictionary;
Output: {‘Roll No’ : 1, ‘Name’ : ‘Smith’ , ‘Age’ : 23}
print(Dict.keys()) #Prints all the keys
Output: dict_keys([‘Roll No’ , ‘Name’ , ‘Age’])
print(Dict.values()) # Print all the values; Output: dict_values([1, ‘Smith’, 23])

#Sets in Python
They are unordered, elements of set are not restricted to a key. ({})

MySet= {11,33} # Creating a set
print(MySet) # Printing set; Output: (33,11)
MySet.add(22) # Adding an element
print(MySet) # Printing set; Output: (33,11,22)
MySet.remove(11) # Deleting an element
print(MySet) # Printing set; Output: (33,22)

#Frozen Set: Elements are unchangeable, hashable
FSet1 = frozenset([11, 22, 33, 44]) # First frozenset
print(FSet1)
FSet2 = frozenset([33, 44, 22, 55])
print(FSet2)
FSet1.isdisjoint(FSet2) # Performing disjoint operation; Output : False

#Data Preprocessing

 # Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

#Check working directory

```
import os
os.getcwd()
```

#Output

```
'C:\\Users\\Mukesh  Yadav'
```

#Change working directory

```
os.chdir('C:\Users\Mukesh  Yadav\Desktop\Rizvi Colg\DataScience Using Python\Part 1 - Data Preprocessing')
os.getcwd()
```

#Output:

```
'C:\\Users\\Mukesh  Yadav\\Desktop\\Rizvi Colg\\DataScience Using Python\\Part 1 - Data Preprocessing'
```

#Importing the datasets

```
dataset = pd.read_csv('Data.csv')
X = dataset.iloc[:, :-1].values
Y = dataset.iloc[:, 3].values
```

#Display dataset

```
dataset
```

#Output of dataset:

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes
5	France	35.0	58000.0	Yes
6	Spain	NaN	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes

#Display X

```
X
```

#Output of X:

```
array([['France', 44.0, 72000.0],
       ['Spain', 27.0, 48000.0],
       ['Germany', 30.0, 54000.0],
       ['Spain', 38.0, 61000.0],
       ['Germany', 40.0, nan],
       ['France', 35.0, 58000.0],
       ['Spain', nan, 52000.0],
       ['France', 48.0, 79000.0],
       ['Germany', 50.0, 83000.0],
       ['France', 37.0, 67000.0]], dtype=object)
```

#Display Y

```
Y
```

#Output of Y:

```
array(['No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes'], dtype=object)
```

#Taking care of missing values

```
from sklearn.preprocessing import Imputer
imputer = Imputer(missing_values = 'NaN', strategy = 'mean', axis = 0)
imputer = imputer.fit(X[:, 1:3])
X[:, 1:3] = imputer.transform(X[:, 1:3])
```

#Output:

```
array([['France', 44.0, 72000.0],
       ['Spain', 27.0, 48000.0],
       ['Germany', 30.0, 54000.0],
       ['Spain', 38.0, 61000.0],
       ['Germany', 40.0, 63777.77777777778],
       ['France', 35.0, 58000.0],
       ['Spain', 38.77777777777778, 52000.0],
       ['France', 48.0, 79000.0],
       ['Germany', 50.0, 83000.0],
       ['France', 37.0, 67000.0]], dtype=object)
```

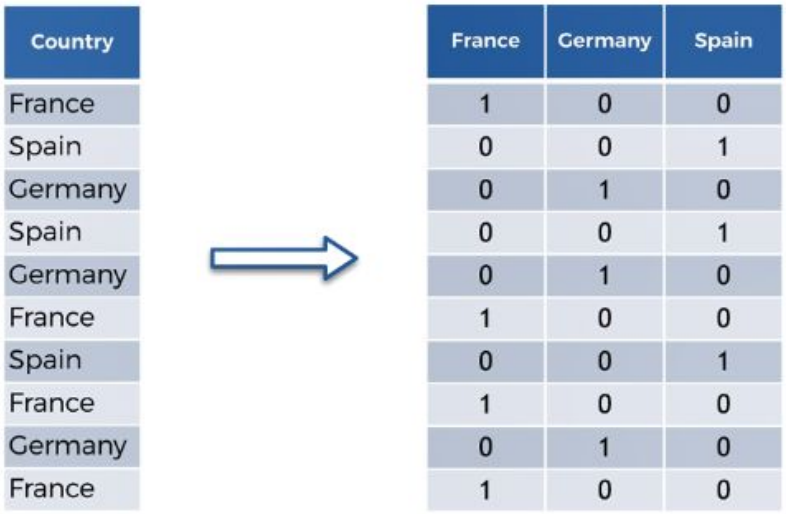
 # Encoding categorical data

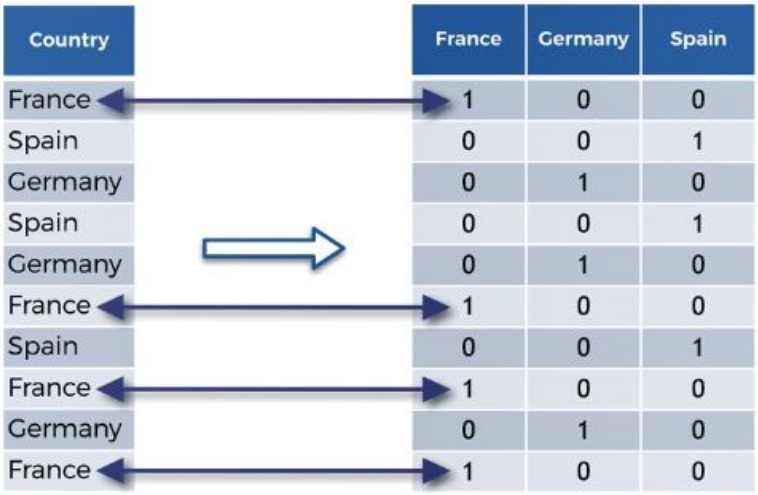
```
from sklearn.preprocessing import LabelEncoder
labelencoder_X = LabelEncoder()
X[:, 0] = labelencoder_X.fit_transform(X[:, 0])
X
```

#Output

```
array([[0L, 44.0, 72000.0],
       [2L, 27.0, 48000.0],
       [1L, 30.0, 54000.0],
       [2L, 38.0, 61000.0],
       [1L, 40.0, 63777.77777777778],
       [0L, 35.0, 58000.0],
       [2L, 38.77777777777778, 52000.0],
       [0L, 48.0, 79000.0],
       [1L, 50.0, 83000.0],
       [0L, 37.0, 67000.0]], dtype=object)
```

#Dummy Encoding





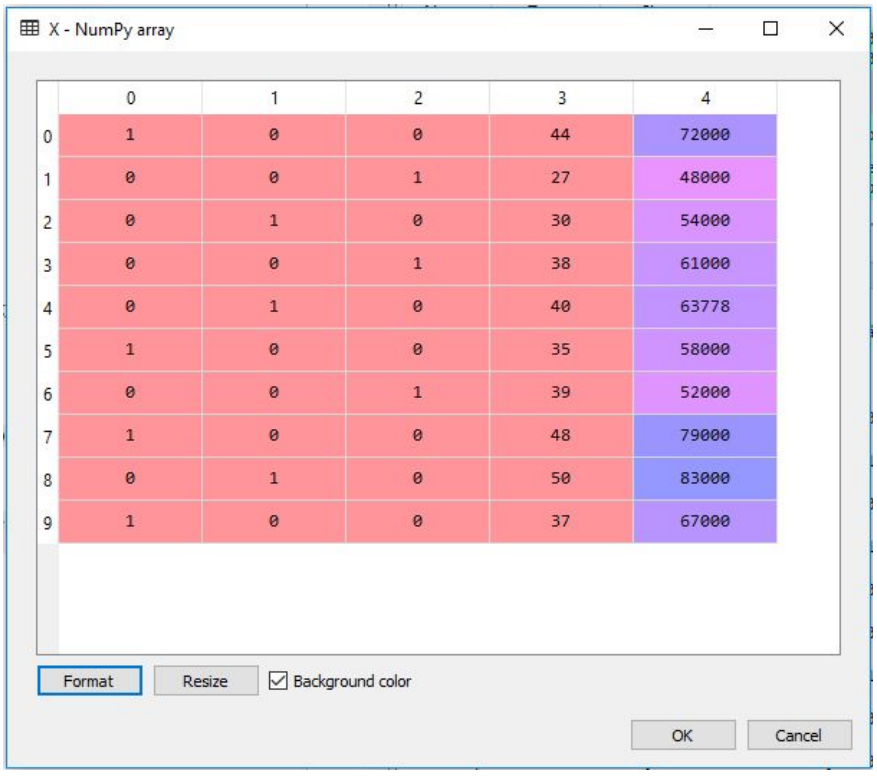
Country	France	Germany	Spain
France	1	0	0
Spain	0	0	1
Germany	0	1	0
Spain	0	0	1
Germany	0	1	0
France	1	0	0
Spain	0	0	1
France	1	0	0
Germany	0	1	0
France	1	0	0

#Dummy variables

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X = LabelEncoder()
X[:, 0] = labelencoder_X.fit_transform(X[:, 0])

onehotencoder = OneHotEncoder(categorical_features = [0])
X = onehotencoder.fit_transform(X).toarray()
```

#Output



 # Encoding the Dependent Variable

```
labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)
```

#Output:

dataset - DataFrame

Index	Country	Age	Salary	Purchased
0	France	44	7.2e+04	No
1	Spain	27	4.8e+04	Yes
2	Germany	30	5.4e+04	No
3	Spain	38	6.1e+04	No
4	Germany	40	nan	Yes
5	France	35	5.8e+04	Yes
6	Spain	nan	5.2e+04	No
7	France	48	7.9e+04	Yes
8	Germany	50	8.3e+04	No
9	France	37	6.7e+04	Yes

y - NumPy array

	0
0	0
1	1
2	0
3	0
4	1
5	1
6	0
7	1
8	0
9	1

 # Splitting the dataset into the Training set and Test set

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

 # Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)
```