Python Programming for Machine Learning (2)-Data Frame

Parcours Progis

Etudes, Medias, communication, Marketing

Bahareh Afshinpour

18.11.2024

Data Frames attributes

Python objects have *attributes* and *methods*.

df.attribute	description
dtypes	list the types of the columns
columns	list the column names
axes	list the row labels and column names
ndim	number of dimensions
size	number of elements
shape	return a tuple representing the dimensionality
values	numpy representation of the data

Data Frames methods

Unlike attributes, python methods have *parenthesis*.
All attributes and methods can be listed with a *dir()* function: `dir(df)`

df.method()	description
head([n]), tail([n])	first/last n rows
describe()	generate descriptive statistics (for numeric columns only)
max(), min()	return max/min values for all numeric columns
mean(), median()	return mean/median values for all numeric columns
std()	standard deviation
sample([n])	returns a random sample of the data frame
dropna()	drop all the records with missing values

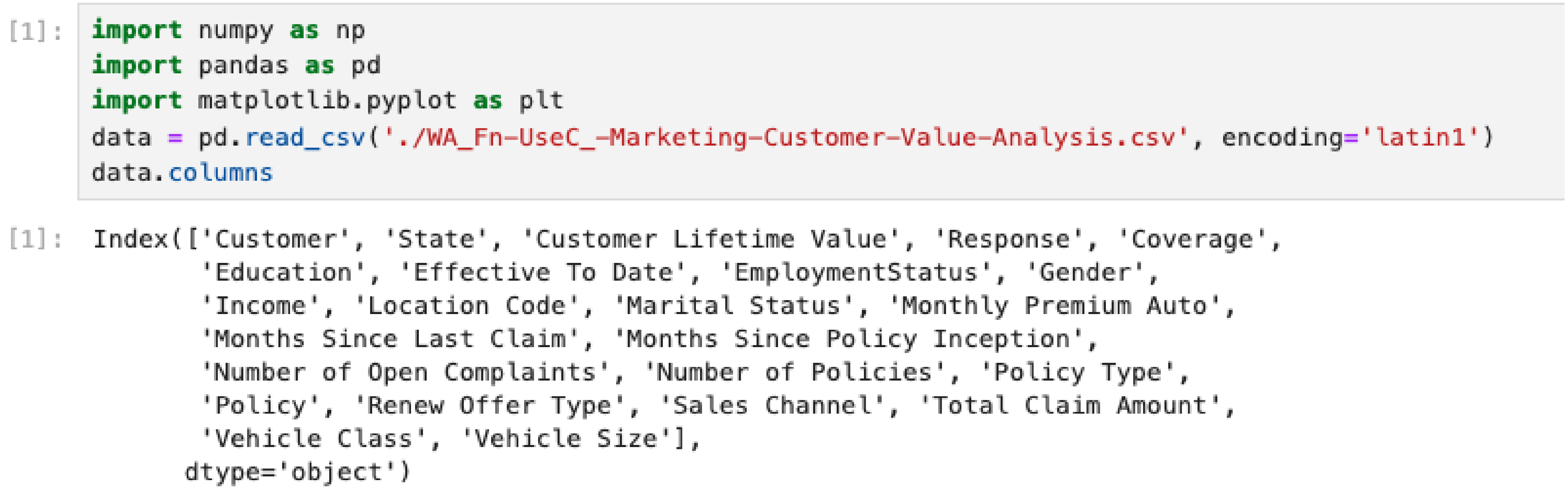
Exploring data frames

You can have a look at the first five rows of your dataframe with **.head()**:

```
#List first 5 records
df.head()
```

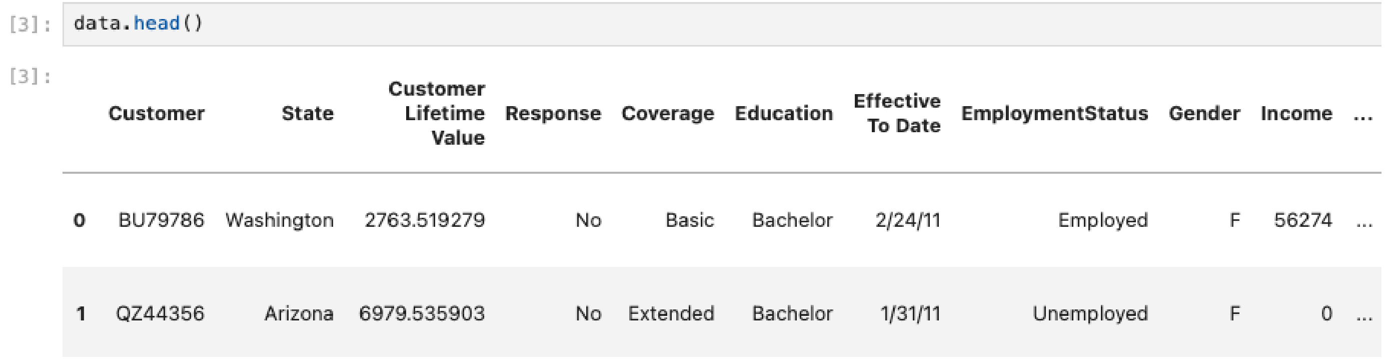
```
   rank  discipline  phd  service   sex  salary
0  Prof           B   56       49  Male  186960
1  Prof           A   12        6  Male   93000
2  Prof           A   23       20  Male  110515
3  Prof           A   40       31  Male  131205
4  Prof           B   20       18  Male  104800
```

You also use the **.shape** attribute of the DataFrame to see its **dimensionality**. The result is a tuple containing the number of rows and columns.



```
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     data = pd.read_csv('./WA_Fn-UseC_-Marketing-Customer-Value-Analysis.csv', encoding='latin1')
     data.columns
```

```
[1]: Index(['Customer', 'State', 'Customer Lifetime Value', 'Response', 'Coverage',
            'Education', 'Effective To Date', 'EmploymentStatus', 'Gender',
            'Income', 'Location Code', 'Marital Status', 'Monthly Premium Auto',
            'Months Since Last Claim', 'Months Since Policy Inception',
            'Number of Open Complaints', 'Number of Policies', 'Policy Type',
            'Policy', 'Renew Offer Type', 'Sales Channel', 'Total Claim Amount',
            'Vehicle Class', 'Vehicle Size'],
           dtype='object')
```



```
[3]: data.head()
```

[3]:

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	EmploymentStatus	Gender	Income	...
0	BU79786	Washington	2763.519279	No	Basic	Bachelor	2/24/11	Employed	F	56274	...
1	QZ44356	Arizona	6979.535903	No	Extended	Bachelor	1/31/11	Unemployed	F	0	...

Selecting a column in a Data Frame

Method 1: Subset the data frame using column name:

```
df['sex']
```

Method 2: Use the column name as an attribute:

```
df.sex
```

Note: there is an attribute *rank* for pandas data frames, so to select a column with a name "rank" we should use method 1.

Data Frame data types

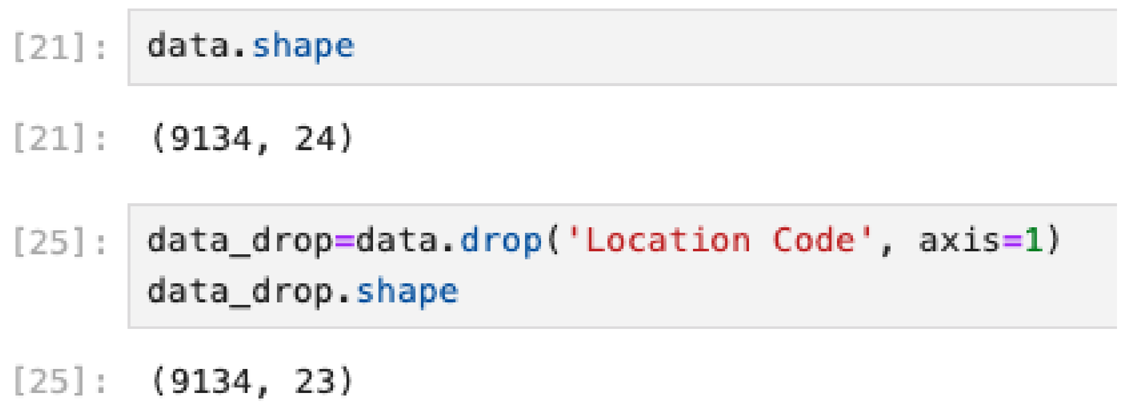
```
#Check a particular column type
df['salary'].dtype
```

```
#Check types for all the columns
df.dtypes
```

```
rank          object
discipline    object
phd           int64
service       int64
sex           object
salary        int64
dtype: object
```

DataFrame drop() Method

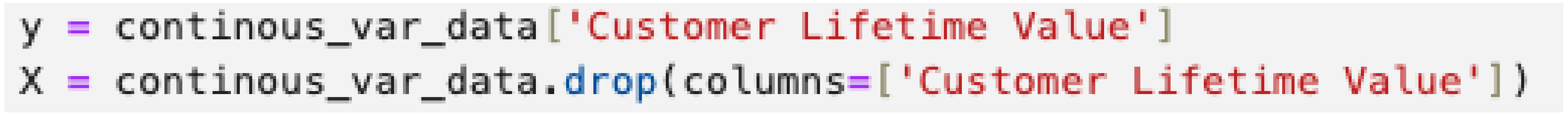
- Suppose one column x has many missing values, so we want to drop it.
- we can drop a single column from a DataFrame using the **.drop()** .



```
[21]: data.shape

[21]: (9134, 24)

[25]: data_drop=data.drop('Location Code', axis=1)
      data_drop.shape

[25]: (9134, 23)
```

Defining Independent and Target features

How to take our data frame and split it into two important parts.



```
y = continous_var_data['Customer Lifetime Value']
X = continous_var_data.drop(columns=['Customer Lifetime Value'])
```

Checking for the Missing values

Missing values can be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located.

Numerical features

A	B
22	34
36	78
15	66
	78

Replace with mean

A	B
22	34
36	78
15	66
24,3	78
24,3	64

Categorical features

C	D
A	P
A	P
A	Q
B	R
	p

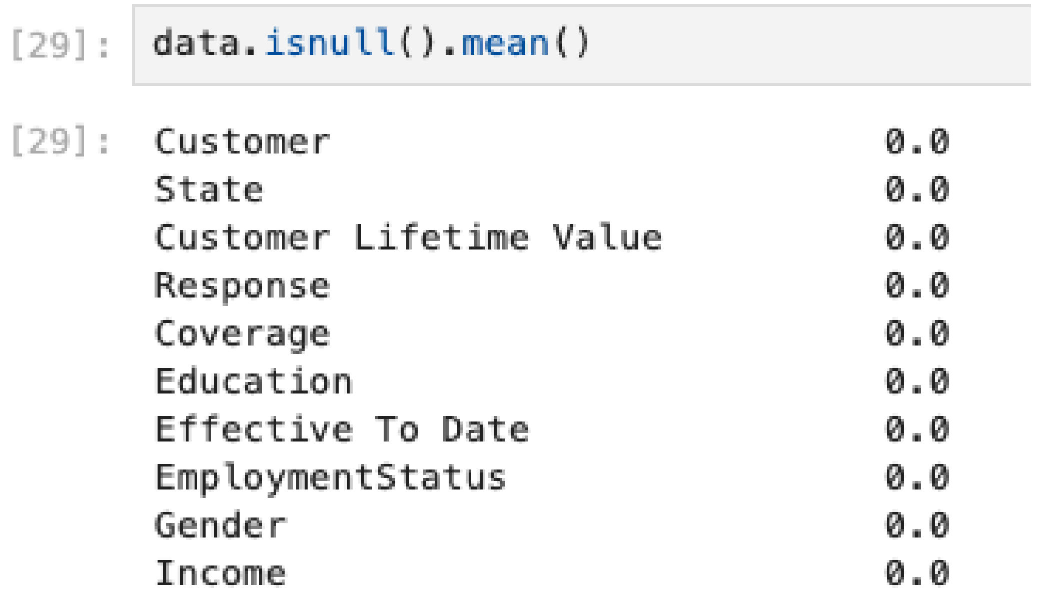
Replace with mode

C	D
A	P
A	P
A	Q
B	R
A	P

isnull(), Detect missing values for an array-like object.

Checking for the Missing values

- Mean simply returns the sum of the values divided by the number of values.



```
[29]: data.isnull().mean()

[29]: Customer                        0.0
      State                           0.0
      Customer Lifetime Value         0.0
      Response                        0.0
      Coverage                        0.0
      Education                       0.0
      Effective To Date               0.0
      EmploymentStatus                0.0
      Gender                          0.0
      Income                          0.0
```

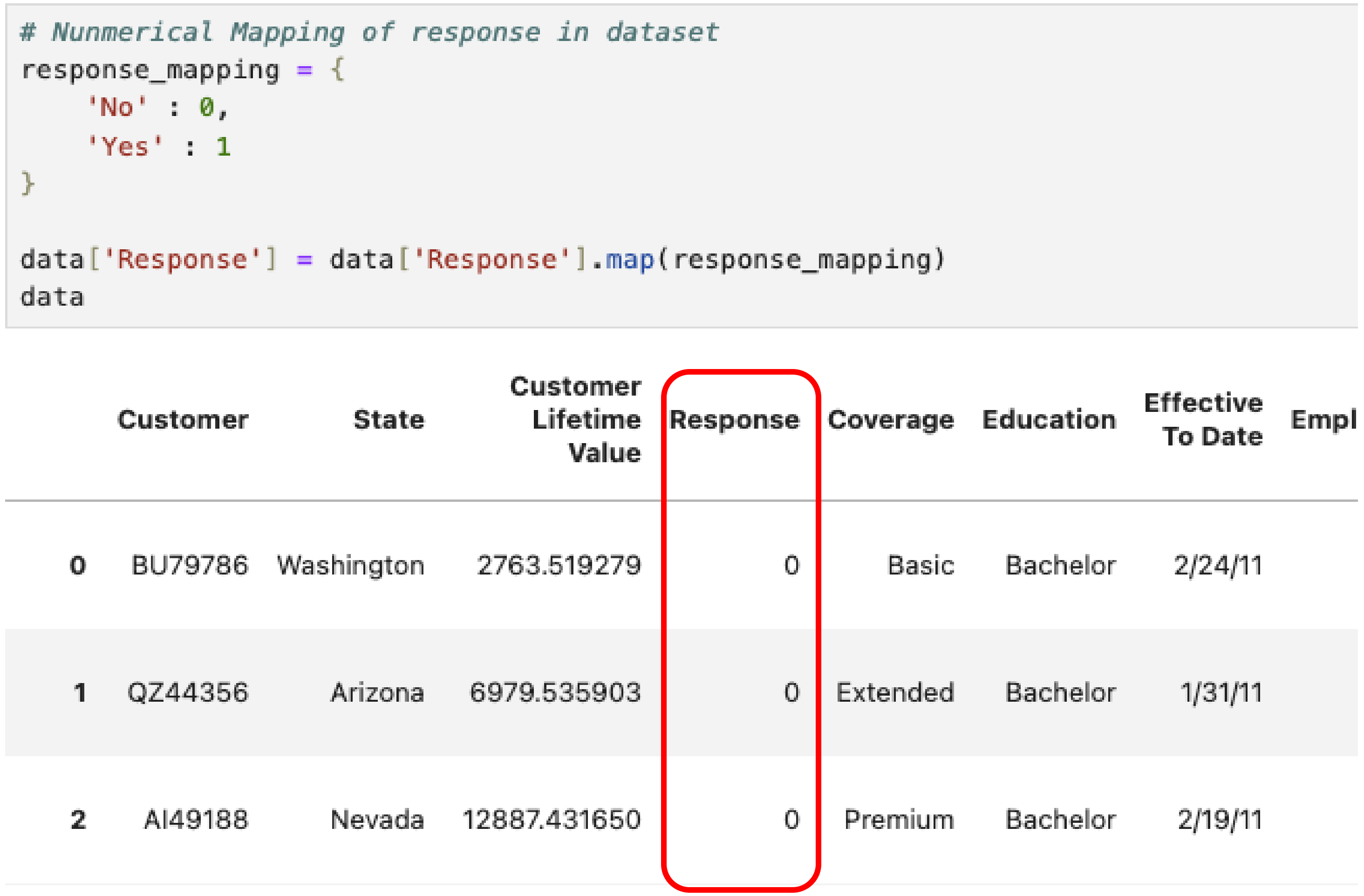
data.isnull().mean()

This data does not have any missing data.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

DataFrame.map()

Used for substituting each value in a Series with another value



```
# Nunmerical Mapping of response in dataset
response_mapping = {
    'No' : 0,
    'Yes' : 1
}

data['Response'] = data['Response'].map(response_mapping)
data
```

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	Empl
0	BU79786	Washington	2763.519279	0	Basic	Bachelor	2/24/11	
1	QZ44356	Arizona	6979.535903	0	Extended	Bachelor	1/31/11	
2	AI49188	Nevada	12887.431650	0	Premium	Bachelor	2/19/11	

End