



Computer vision in the new era of Artificial Intelligence and Deep Learning

Visión por computador en la nueva era de la Inteligencia Artificial y el Deep Learning

Rubén Usamentiaga*, Alberto Fernández°

*** University of Oviedo**

° TSK

Gijón (Spain)
5 – 16 April 2021



<https://github.com/albertofernandezvillan/computer-vision-and-deep-learning-course>

Pandas

Introduction to pandas

Minimal example with pandas and scikit-learn



- [pandas introduction.ipynb](#)
- [pandas and scikit learn introduction.ipynb](#)



- [pandas introduction.ipynb](#)
- [pandas and scikit learn introduction.ipynb](#)

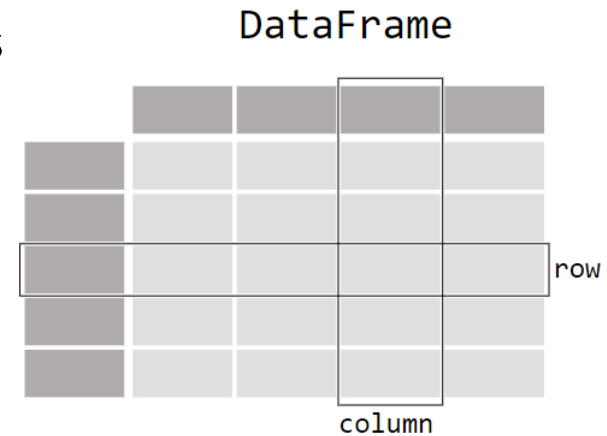


<https://github.com/albertofernandezvillan/computer-vision-and-deep-learning-course>

Anatomy of a DataFrame

The Pandas DataFrame is a structure that contains two-dimensional data and its corresponding labels.

It is composed of three different components, the index, columns, and the data.



columns
axis=1

column name

more columns to display

index label

index
axis=0

missing values

data
(values)

	color	director_name	num_critic_for_reviews	duration	...	actor_2_facebook_likes	imdb_score	aspect_ratio	movie_facebook_likes
0	Color	James Cameron	723.0	178.0	...	936.0	7.9	1.78	33000
1	Color	Gore Verbinski	302.0	169.0	...	5000.0	7.1	2.35	0
2	Color	Sam Mendes	602.0	148.0	...	393.0	6.8	2.35	85000
3	Color	Christopher Nolan	813.0	164.0	...	23000.0	8.5	2.35	164000
4	NaN	Doug Walker	NaN	NaN	...	12.0	7.1	NaN	0

Exploring a DataFrame

```
data = np.array([np.arange(3),  
                 np.arange(3,6),  
                 np.arange(6,9),  
                 np.arange(9,12)])
```

```
df = pd.DataFrame(data=data,  
                  index=["i1", "i2", "i3", "i4"],  
                  columns=["c1", "c2", "c3"])
```

df.dtypes

```
c1  int64  
c2  int64  
c3  int64
```

df.shape

(4, 3)

(rows: axis=0, cols: axis=1)

df.empty

False

df.size

12

df.describe()

	c1	c2	c3
count	4.000000	4.000000	4.000000
mean	4.500000	5.500000	6.500000
std	3.872983	3.872983	3.872983
min	0.000000	1.000000	2.000000
25%	2.250000	3.250000	4.250000
50%	4.500000	5.500000	6.500000
75%	6.750000	7.750000	8.750000
max	9.000000	10.000000	11.000000

df.info()

```
<class 'pandas.core.frame.DataFrame'>  
Index: 4 entries, i1 to i4  
Data columns (total 3 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0    c1         4 non-null    int64  
1    c2         4 non-null    int64  
2    c3         4 non-null    int64  
dtypes: int64(3)  
memory usage: 288.0+ bytes
```

df.columns

	c1	c2	c3
i1	0	1	2
i2	3	4	5
i3	6	7	8
i4	9	10	11

df.head(2)

	c1	c2	c3
i1	0	1	2
i2	3	4	5

df.tail(1)

	c1	c2	c3
i4	9	10	11

df.index

df.values

df.isnull()

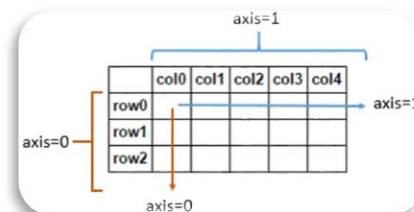
	c1	c2	c3
i1	False	False	False
i2	False	False	False
i3	False	False	False
i4	False	False	False

df.isnull().sum(axis=0)

	c1
c1	0
c2	0
c3	0

df.isnull().sum(axis=1)

	i1
i1	0
i2	0
i3	0
i4	0



Working with a DataFrame

All the operations are referred to the initial DataFrame

```
data = np.array([np.arange(3), np.arange(3,6), np.arange(6,9),  
np.arange(9,12),np.arange(3),[100, np.nan, np.nan]])
```

```
df = pd.DataFrame(data=data,index=["i1", "i2", "i3", "i4", "i5", "i6"],columns=["c1", "c2", "c3"])
```

```
df2 = df.append(df)
```

```
df.insert(1, 'new_column', df['c1'] * 10)
```

	c1	c2	c3
i1	0.0	1.0	2.0
i2	3.0	4.0	5.0
i3	6.0	7.0	8.0
i4	9.0	10.0	11.0
i5	0.0	1.0	2.0
i6	100.0	NaN	NaN

	c1	c2	c3
i1	0.0	1.0	2.0
i2	3.0	4.0	5.0
i3	6.0	7.0	8.0
i4	9.0	10.0	11.0
i5	0.0	1.0	2.0
i6	100.0	NaN	NaN

	c1	new_column	c2	c3
i1	0.0	0.0	1.0	2.0
i2	3.0	30.0	4.0	5.0
i3	6.0	60.0	7.0	8.0
i4	9.0	90.0	10.0	11.0
i5	0.0	0.0	1.0	2.0
i6	100.0	1000.0	NaN	NaN

```
df.fillna(123.0, inplace = True)
```

	c1	c2	c3
i1	0.0	1.0	2.0
i2	3.0	4.0	5.0
i3	6.0	7.0	8.0
i4	9.0	10.0	11.0
i5	0.0	1.0	2.0
i6	100.0	123.0	123.0

```
df.rename(columns={"c1": "C1"})
```

```
df.rename(index={"i1": "I1"})
```

	C1	c2	c3
I1	0.0	1.0	2.0
i2	3.0	4.0	5.0

```
df.drop_duplicates(inplace=True)
```

```
df.dropna(axis=0, inplace = True)
```

	c1	c2	c3
i1	0.0	1.0	2.0
i2	3.0	4.0	5.0
i3	6.0	7.0	8.0
i4	9.0	10.0	11.0
i5	0.0	1.0	2.0

All rows with nan values will be deleted

'first' by default

	c1	c2	c3
i1	0.0	1.0	2.0
i2	3.0	4.0	5.0
i3	6.0	7.0	8.0
i4	9.0	10.0	11.0
i6	100.0	NaN	NaN

Selecting subsets of data in pandas DataFrame

```
numpy_data = np.array([np.arange(3), np.arange(3,6), np.arange(6,9), np.arange(9,12)])  
  
df = pd.DataFrame(data=numpy_data, index=["index1", "index2", "index3", "index4"],  
columns=["column1", "column2", "column3"])
```



	column1	column2	column3
index1	0	1	2
index2	3	4	5
index3	6	7	8
index4	9	10	11

```
df[["column1","column2"]]
```

	column1	column2
index1	0	1
index2	3	4
index3	6	7
index4	9	10

**Selecting multiple columns
returns a DataFrame**

**Selecting a single column
returns a Series**

```
df["column1"]
```

index1	0
index2	3
index3	6
index4	9

```
df.loc[:,["column1"]]
```

```
df.iloc[:, [0]]
```

	column1
index1	0
index2	3
index3	6
index4	9

```
df.loc["index1":"index3",:]
```

```
df.iloc[0:3, :]
```

	column1	column2	column3
index1	0	1	2
index2	3	4	5
index3	6	7	8

```
df.loc["index2","column1"]
```

```
df.iloc[1, 0]
```

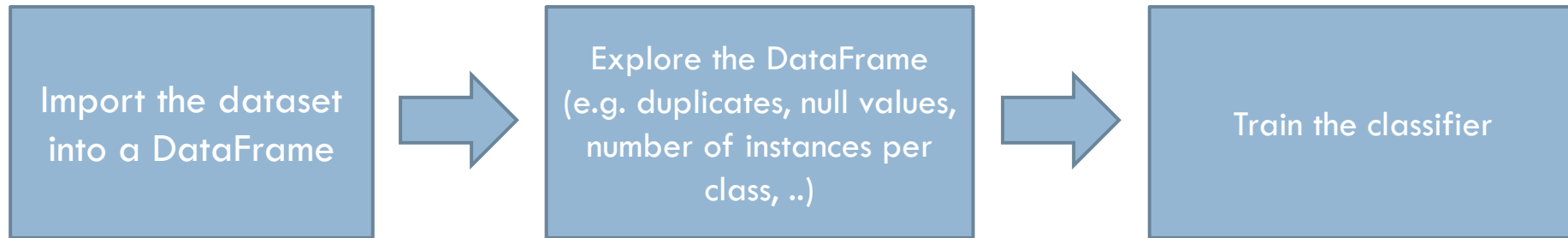
3

```
df.loc[["index2"],["column1"]]
```

```
df.iloc[[1], [0]]
```

	column1
index2	3

Minimal example using both scikit-learn and pandas for classification



In this [example](#), we have performed a minimal example using both pandas and scikit-learn to tackle a classification problem.

Note also that we have only trained the classifier but other points are missing (e.g. making predictions using the trained model, measuring the performance, model persistence,...). See [this notebook](#), where aforementioned topics are covered using also this dataset.

Pandas

Introduction to pandas

Minimal example with pandas and scikit-learn



<https://github.com/albertofernandezvillan/computer-vision-and-deep-learning-course>