

In this notebook we have a first look at the data set given.

```
• begin
•   using Pkg
•   Pkg.activate(joinpath(Pkg.devdir(), "MLCourse"))
•   using CSV, DataFrames, Distributions, Plots, MLJ, MLJLinearModels, Random,
•   OpenML
•   end
```

Exploration

We load the precipitation data from a csv file on the haddisk to a DataFrame. Our goal is to predict whether there is some precipitation (rain, snow etc.) on the next day in Pully, getting measurements from different weather stations in Switzerland.

Training data properties

`precipitation =`

	ABO_radiation_1	ABO_delta_pressure_1	ABO_air_temp_1	ABO_sunshine_1	ABO_win
1	-0.166667	-1.2	-5.68333	0.0	2.08333
2	0.333333	0.2	5.16667	0.0	1.43333
3	3.83333	-0.3	9.61667	0.0	1.35
4	16.5	-0.3	6.16667	33.0	0.983333
5	4.66667	-0.1	9.3	0.0	3.9
6	5.33333	-0.6	9.01667	0.0	7.6
7	1.16667	missing	0.0333333	0.0	2.46667
8	9.83333	0.3	7.38333	0.0	7.98333
9	1.66667	-0.3	-0.75	0.0	1.98333
10	10.1667	-1.2	4.81667	0.0	1.8
⋮ more					
3176	17.3333	0.1	14.9333	28.0	4.48333

```
• precipitation = CSV.read(joinpath(@__DIR__, "..", "data", "project",
•   "trainingdata.csv"), DataFrame)
```

	variable	mean	min	median	max	nmissing	
1	:ABO_radiation_1	6.25181	-1.66667	2.83333	40.1667	1	Union{M
2	:ABO_delta_pressure_1	-0.51594	-8.0	-0.5	8.8	58	Union{M
3	:ABO_air_temp_1	4.3429	-21.1833	4.88333	24.7	3	Union{M
4	:ABO_sunshine_1	5.95274	0.0	0.0	79.0	2	Union{M
5	:ABO_wind_1	3.69496	0.0	2.05	41.4	11	Union{M
6	:ABO_wind_direction_1	209.203	0.0	225.0	333.0	11	Union{M
7	:ALT_radiation_1	4.66535	-1.66667	1.16667	36.3333	0	Float64
8	:ALT_delta_pressure_1	-0.168485	-10.9	-0.2	10.0	60	Union{M
9	:ALT_air_temp_1	8.26916	-12.8667	8.75	26.9667	0	Float64
10	:ALT_sunshine_1	3.23339	0.0	0.0	51.0	1	Union{M
	: more						
529	:precipitation_nextday	0.426322	false	0.0	true	0	Bool

◀ describe(precipitation) ▶

▶ (3176, 529)

• size(precipitation)

The training data contains:

- 3176 observations
- 529 predictors

The values of variables are Float64, and the value of y is a boolean.

Comparison with the other data sets

Sample submission

```
precipitation_ss =
```

	id	precipitation_nextday
1	1	0.564894
2	2	0.560655
3	3	0.777565
4	4	0.968714
5	5	0.110537
6	6	0.344311
7	7	0.831808
8	8	0.508994
9	9	0.558186
10	10	0.57066
⋮ more		
1200	1200	0.426492

```
• precipitation_ss = CSV.read(joinpath(@__DIR__, "..", "data", "project",  
"sample_submission.csv"), DataFrame)
```

```
► (1200, 2)
```

```
• size(precipitation_ss)
```

The 2 predictors correspond of id and the precipitation_nextday, so the id does not count in the total of variables.

Test data

```
• precipitation_test = CSV.read(joinpath(@__DIR__, "..", "data", "project",  
"testdata.csv"), DataFrame);
```

```
► (1200, 528)
```

```
• size(precipitation_test)
```

Comparison

These values correspond to the training ones, as the sample submission data contains the precipitation_nextday variable (y), and the test data the 528 variables.

Precipitation_nextday properties

=====

```
y =  
►[false, false, false, false, false, true, false, true, true, true, false, false, false, t  
• y = precipitation.precipitation_nextday[1:end]
```

The y value is a boolean. Let's compare the ratio between true and false values:

True

```
• true_val = precipitation[precipitation.precipitation_nextday .== 1, :]; # select  
  only true
```

```
►(1354, 529)
```

```
• size(true_val)
```

False

```
• false_val = precipitation[precipitation.precipitation_nextday .== 0, :]; # select  
  only false
```

```
►(1822, 529)
```

```
• size(false_val)
```

Comparison

There are 1354 true values and 1822 false ones. The total of both numbers is 3176, which corresponds to the size of training data found before. This means that there is no missing value for y.



