Exploratory Data Analysis with Pandas Profiling

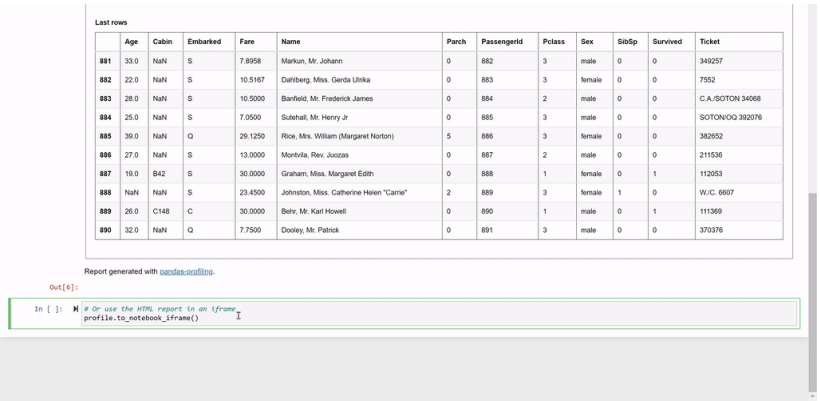
Albert Sanchez Lafuente Feb 9, 2020 · 3 min read ★

Pandas profiling is an open source Python module with which we can quickly do an **exploratory data analysis** with just a few lines of code. Besides, if this is not enough to convince us to use this tool, it also generates interactive reports in web format that can be presented to any person, even if they don't know programming.

In short, what pandas profiling does is save us all the work of visualizing and understanding the distribution of each variable. It generates a report with all the information easily available.

A picture is worth a thousand words

To show you clearly how it works, this is an example of a report generated by pandas profiling:



	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket
881	33.0	NaN	S	7.8958	Markun, Mr. Johann	0	882	3	male	0	0	349257
882	22.0	NaN	S	10.5167	Dahlberg, Miss. Gerda Ulrika	0	883	3	female	0	0	7552
883	28.0	NaN	S	10.5000	Banfield, Mr. Frederick James	0	884	2	male	0	0	C.A./SOTON 34068
884	25.0	NaN	S	7.0500	Sutehall, Mr. Henry Jr	0	885	3	male	0	0	SOTON/OQ 392076
885	39.0	NaN	Q	29.1250	Rice, Mrs. William (Margaret Norton)	5	886	3	female	0	0	382652
886	27.0	NaN	S	13.0000	Montvila, Rev. Juozas	0	887	2	male	0	0	211536
887	19.0	B42	S	30.0000	Graham, Miss. Margaret Edith	0	888	1	female	0	1	112053
888	NaN	NaN	S	23.4500	Johnston, Miss. Catherine Helen "Carrie"	2	889	3	female	1	0	W./C. 6607
889	26.0	C148	C	30.0000	Behr, Mr. Karl Howell	0	890	1	male	0	1	111369
890	32.0	NaN	Q	7.7500	Dooley, Mr. Patrick	0	891	3	male	0	0	370376

You can also check an interactive report clicking this link.

One of the strong points of the generated report are the warnings that appear at the beginning. It tells us the variables that contain NaN values, variables with many zeros, categorical variables with high cardinality, etc.

Section telling us the warnings

How to use pandas profiling

First step is to install it with this command:

```
pip install pandas-profiling
```

Then we generate the report using these commands:

```
from pandas_profiling import ProfileReport
prof = ProfileReport(df)
prof.to_file(output_file='output.html')
```

Here we are, it's been that simple. We can see the report generated in the *output.html* file.

Pandas profiling disadvantage

The main disadvantage of pandas profiling is its use with large datasets. With the increase in the size of the data the time to generate the report also increases a lot.

One way to solve this problem is to generate the report from only a part of all the data we have. It is important to make sure that the data selected to generate the report is representative of all the data we have, for example it could be the case that the first X rows of data contain only data from one category. In this example we would like to randomize the order of the data and select a representative sample.

An example with code:

```
from pandas_profiling import ProfileReport
#We only use the first 10000 data points
prof = ProfileReport(df.sample(n=10000))
prof.to_file(output_file='output.html')
```

Another alternative is to use the *minimum mode* that was introduced in version 2.4 of *pandas profiling*. You can check which version you have installed with this command:

```
pandas_profiling.version.__version__
```

With the minimum mode a simplified report will be generated with less information than the full one but it can be generated relatively quickly for a large dataset. This is the code to be used:

```
profile = ProfileReport(df, minimal=True)
profile.to_file(output_file="output_min.html")
```