Pandas-Profiling Package as Visuzalization

Instructor Notes, YB, RIT

Generating summary report with Pandas-Profiling

Table of Contents

What is Pandas-Profiling Pack?

Visit the main resource.

- Generates profile reports with descriptive statistics and simple static visualization from a pandas DataFrame.

- The pandas df.describe() function is great but a little basic and limiteed for serious exploratory data analysis. pandas_profiling extends the pandas DataFrame with df.profile_report() for quick data analysis.

- For each column the following statistics - if relevant for the column type - are presented in an interactive HTML report:

  - Type inference: detect the types of columns in a dataframe.
  - Essentials: type, unique values, missing values

    - Quantile statistics like minimum value, Q1, median, Q3, maximum, range, interquartile range
    - Descriptive statistics like mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness
    - Most frequent values
    - Histogram
    - Correlations highlighting of highly correlated variables, Spearman, Pearson and Kendall matrices
    - Missing values matrix, count, heatmap and dendrogram of missing values
    - Text analysis learn about categories (Uppercase, Space), scripts (Latin, Cyrillic) and blocks (ASCII) of text data.
    - File and Image analysis extract file sizes, creation dates and dimensions and scan for truncated images or those containing EXIF information.
- Go to Examples on the web. Spend some time here.

Install

- Open Terminal and run this code
- This will load Pandas-profiling. Do not run here.
- Use conda, not pip, because anaconda is our navigator
- I recommend to create a new environemnt if you are not beginner in Jupyter or Python

> conda install -c conda-forge pandas-profiling

- Install using conda. However, you can do it with pip as below: use pip3 or just pip if not working

> !pip install pandas-profiling

Creating a new environment for ISTE-782 Python work

- Open terminal: run the code one by one after reading the outputs.

- List the environments found

> conda env list

- List the Python versions

  > conda search --full-name python

- Create a new environment, myvaenv. This will take time

  > conda create -n myVAenv python=3.11 anaconda

- Check if the new env created

  > conda env list
  >
  > conda list -n myvaenv

- You should ACTIVATE the env before the start

  > conda activate myvaenv

- Once you are done, deactivate the env

  > conda deactivate #or just deactivate

Practice

In [ ]:

```
# Check the working directory
!pwd
```

In [ ]:

```
# import packs
import pandas as pd
import pandas_profiling as pdp
```

In [ ]:

```
# when xlsx file import
#conda install openpyxl

# use pd.red.excel('.xlsx')
# then make pd.DataFrame(df, columns=['','',...])
```

In [ ]:

```
# See the Titanic data set: download under the same directory
df_titanic = pd.read_csv('train-titanic.csv',
                         index_col='PassengerId') #comma-seperated values=csv
```

In [ ]:

```
# run each one by one

df_titanic.info() #types and all missing
#df_titanic.shape #dim
#df_titanic.head(10) #first 10 observation,
#df_titanic.tail(10) #last 10 observation,
#df_titanic.describe() #show only numerical summaries
#df_titanic.describe(include='all') #include all variables including numerical and categorical
```

In [ ]:

```
df_titanic.shape
```

In [ ]:

```
# Did you import the Titanic data set, train-titanic.csv?
report1 = pdp.ProfileReport(df_titanic, title='Pandas Profiling Report - Simple')
report1.to_file("report1.html")
report1
```

- Then open the html files under the folder and explore the results

Explore Deeper

The example code below loads the explorative configuration file, that includes many features for text (length distribution, unicode information), files (file size, creation time) and images (dimensions, exif information).

In [ ]:

```
# Use minimal=True for large data set
report2 = pdp.ProfileReport(df_titanic, title='Pandas Profiling Report - Detailed',
                            explorative=True, html={'style':{'full_width':True}})
report2.to_file("report2.html")
report2
```

- Then open the html files under the folder and explore the results

Minimal Report in case large data

In [ ]:

```
# Use minimal=True for large data set
report3 = pdp.ProfileReport(df_titanic, title='Pandas Profiling Report',
                            minimal=True) #change to False for full report
report3.to_file("report3.html")
report3
```

- Then open the html files under the folder and explore the results

In [ ]:

```
# Optional: Try advanced codes with html, playing with style etc.:

#html={'style':{'full_width':True}}

# Focus on Target variable
```

Your Turn

- Just play and include the codes for future reference. Hope

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

References

- Pandas-profiling

- Practice with this Kaggle notebook