4 TOOLS FOR AUTOMATIC EXPLORATORY DATA ANALYSIS (EDA) IN PYTHON ¶

1.Pandas Profiling

1.Pandas Profiling: Pandas Profiling is a simple-to-use open source Python package for exploratory data analysis. It produces a report of your dataframe in a variety of formats. While the pandas df.describe function is excellent, it does not provide a detailed report of your dataframe, you can perform a quick data analysis using Pandas profiling.

In [22]:
```
# installing package
!pip install pandas-profiling
```

In [4]:
```
#importing the data
import pandas as pd
df = pd.read_csv(r"C:\Users\DHARAVATH RAMDAS\Downloads\Dataset\Dataset\data1\Trav
```

In [21]:
```
#importing library
import pandas as pd
import numpy as np
import pandas_profiling
#from pandas_profiling import ProfileReport
#defining the library

pandas_profiling.ProfileReport(df)
```

Summarize dataset: 71/71 [00:15<00:00, 2.24it/s, 100% Completed]

Generate report structure: 1/1 [00:12<00:00, 12.65s/it] 100%

Render HTML: 100% 1/1 [00:03<00:00, 3.28s/it]

204881	1	< 0.1%
204880	1	< 0.1%
204879	1	< 0.1%
204878	1	< 0.1%

ProdTaken

Categorical

Distinct	2
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	38.3 KiB

Length

Max length	1
Median length	1
Mean length	1
Min length	1

Characters and Unicode

Out[21]:

```
The profile report generated above will give you a report of the whole columns
in the dataset – the number of missing values, skewness, categorical variables,
correlations and charts.
```

2. Dtale

2. Dtale: D-Tale combines a Flask back-end and a React front-end to provide an easy-to-use interface for viewing and analyzing Pandas data structures. It integrates with ipython notebooks and python/ipython terminals seamlessly. This tool currently supports the following Pandas objects: DataFrame, Series, MultiIndex, DatetimeIndex, and RangeIndex. As with Pandas profiling, it supports a variety of visualizations, including heatmaps, charts, and three-dimensional plots.

In [23]:
```
pip install dtale
```

importing our library and visualizing our report

In [7]:
```
import dtale

import dtale.app as dtale_app
dtale_app.USE_COLAB = True
dtale.show(df)
```

▶ 20 / 4888

	CustomerID	ProdTaken	Age	TypeofContact	CityTier	DurationOfPi
40	200040	0	35.00	Self Enquiry	1	1(
41	200041	0	27.00	Self Enquiry	1	8
42	200042	1	26.00	Self Enquiry	1	3
43	200043	0	27.00	Company Invited	3	14
44	200044	0	nan	Company Invited	1	6
45	200045	1	41.00	Self Enquiry	1	18
46	200046	0	34.00	Company Invited	3	
47	200047	0	37.00	Self Enquiry	1	25
48	200048	0	46.00	Company Invited	3	1
49	200049	0	35.00	Self Enquiry	1	14

It generates the report as a html file, so you click the link to take you to the site

3. Sweetviz

3. Sweetviz is a free, open-source Python library that generates beautiful, high-density visualizations for exploratory data analysis with only a few lines of code.

Sweetviz for sweet visualizations 😊

In [24]:
```
pip install sweetviz
```

In [17]:
```
#importing sweetviz and visualizing our data
import sweetviz as sv
sweet_report = sv.analyze(df)
sweet_report.show_html('sweetviz_report.html')
```

```
Done! Use 'show' commands to display/save.                    [100%] 00:02 -> (00:00 left)

Report sweetviz_report.html was generated! NOTEBOOK/COLAB USERS: the web browse
r MAY not pop up, regardless, the report IS saved in your notebook/colab files.
```

Out[17]:
```
<bound method DataframeReport.show_html of <sweetviz.dataframe_report.Dataframe
Report object at 0x00000157DCE4AAC0>>
```

In [18]:
```
df1 = sv.compare(df[102:], df[:102])
df1.show_html('Compare.html')
```

```
Done! Use 'show' commands to display/save.                    [100%] 00:03 -> (00:00 left)

Report Compare.html was generated! NOTEBOOK/COLAB USERS: the web browser MAY no
t pop up, regardless, the report IS saved in your notebook/colab files.
```

```
The result will be generated as a html file, so you have to check your notebook
files.
```

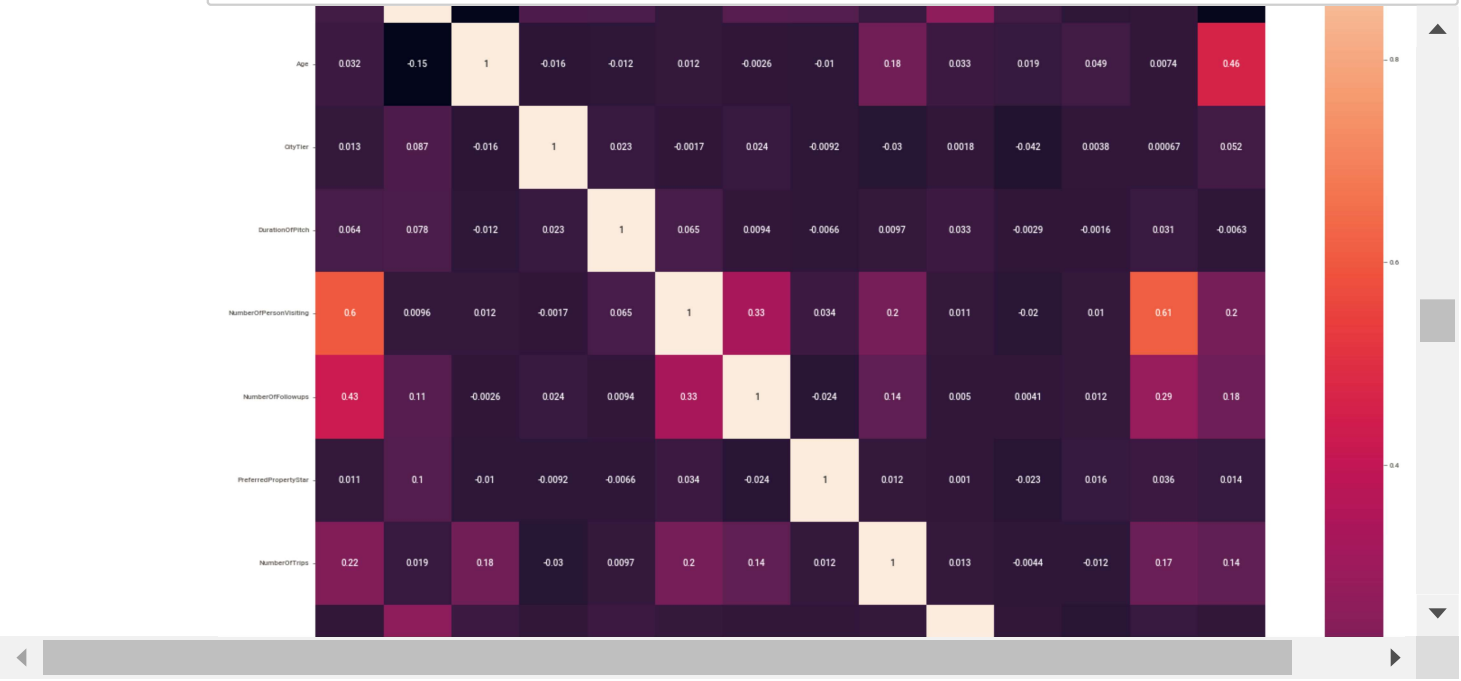
4. AutoViz

```
4. AutoViz: AutoViz automatically visualizes any dataset with a single line of
code. AutoViz is capable of identifying the most critical features and plotting
visually compelling visualizations solely on the basis of those automatically
selected features. Additionally, AutoViz is lightning fast, generating
visualization in a matter of seconds.
```

In [25]:
```
pip install autoviz
```

In [19]:

```
from autoviz.AutoViz_Class import AutoViz_Class
AV = AutoViz_Class()
%matplotlib inline
#you have to specify the target variable
dft = AV.AutoViz(r"C:\Users\DHARAVATH RAMDAS\Downloads\Dataset\Dataset\data1\Trav
```



Note that you don't even need Pandas to read the data. AutoViz will load it when you provide the path to the data set. Here's the report we generated with AutoViz.

AutoViz, you have many more plots (i.e., violin, boxplots and more) as well as statistical and probability values. However, the UI isn't as neat as others' reports, and you don't have access to interactive plots.