GROUP NUMBER - 12

PYTHON PACKAGES

MEMBER 1 - JATAN SAHU (202218061)

MEMBER 2 - AJAY DAVE (202218017)

MEMBER 3 - ANUSHKA JAIN (202218057)

MEMBER 4 - VEDANT JOSHI (202003001)

Many Python packages can be useful for exploratory data analysis (EDA), depending on the specific needs and preferences of the user. Here are some of the most commonly used ones:

NumPy: for numerical operations and array manipulation.

Pandas: for data manipulation and analysis.

Matplotlib: for data visualisation and plotting.

Seaborn: for advanced data visualisation, including statistical graphics.

Plotly: for interactive and dynamic data visualisation.

Scikit-learn: for machine learning and predictive modelling.

Statsmodels: for statistical modelling and analysis.

SciPy: for scientific and technical computing.

Bokeh: for interactive and web-based data visualisation.

yellowbrick: for machine learning visualisation and diagnostics.

Speedml : SpeedML speeds up the process of machine learning by providing different functionalities which makes Exploratory Data Analysis an easy task.

Dataprep: DataPrep is an open-source library available for python that lets you prepare your data using a single library with only a few lines of code.

1. NumPy

Creator: Travis Olliphant

Dependencies: None (although it is often used in conjunction with other packages)

Associated functions:

Computing basic statistics (e.g., mean, median, variance)

Linear algebra operations

Random number generation

Array manipulation

2. Pandas

Creator: Wes McKinney

Dependencies: NumPy

Associated functions:

Data ingestion (e.g., reading CSV, Excel, or SQL data)

Data cleaning and preprocessing (e.g., handling missing values)

Data exploration (e.g., summarising data, creating pivot tables)

3. Matplotlib

Creator: John Hunter

Dependencies: NumPy

Associated functions:

Creating 2D visualisations (e.g., scatter plots, line plots)

Customising plot aesthetics (e.g., axis labels, legend, colours)

4. Seaborn

Creator: Michael Waskom

Dependencies: Matplotlib, Pandas, NumPy

Associated functions:

Creating complex visualisations (e.g., heatmaps, pair plots, box plots)

Customising plot aesthetics (e.g., colour palettes)

5. Plotly

Creator: Alex Johnson, Jack Parmer, Chris Parmer, and Matthew Sundquist

Dependencies: NumPy and optionally depends on Pandasfor data manipulation, and other libraries such as Flask and requests for web applications and API interactions.

Associated functions: creating line plots,scatter plots, bar charts, histograms, heatmaps, and more.

Alternatives: Seaborn, Bokeh etc.

6. Scikit-learn

Creator: David Cournapeau

Dependencies: NumPy, SciPy, Matplotlib, joblib (optional)

Associated functions:

Data preprocessing (e.g., scaling, encoding categorical variables)

Dimensionality reduction (e.g., PCA)

Clustering

Regression

Classification

7. Statsmodels

Creator: Skipper Seabold

Dependencies: NumPy, Pandas, SciPy, patsy

Associated functions:

Statistical modeling (e.g., linear regression, ANOVA, time series analysis)

Hypothesis testing

Model selection

8. SciPy

Creator: SciPy was created by a large group of contributors, with the initial work being done by Travis Oliphant.

Dependencies: NumPy is a dependency of SciPy, and the two packages are often used together. Some parts of SciPy rely on other packages such as BLAS, LAPACK, and FFT.

Associated functions:

SciPy is a library of numerical algorithms and mathematical tools for Python. It pr Mk ovides functionality for optimisation, integration, linear algebra, signal processing, and more.

Alternatives:

stats.ttest_ind(): stats.ttest_rel(), stats.ttest_1samp() signal.convolve(): signal.correlate(), signal.fftconvolve() integrate.quad(): integrate.fixed_quad(), integrate.romberg()

9. Bokeh

Creator: Bryan Van de Ven

Dependencies: NumPy, Jinja2, Tornado, PyYAML, Pillow.

Associated functions:

Interactive visualisation library, creating line plots, scatter plots, bar charts, and heatmaps, providing tools for creating interactive plots and dashboards in web browsers.

Alternatives:

figure(): output_file(), output_notebook() line(): circle(), rect() vplot(), hplot(): gridplot()

10. yellowbrick

Creator: Benjamin Bengfort & Rebecca Bilbro

Dependencies: NumPy optionally depends on Pandas for data manipulation and other libraries, such as Flask and Tornado, for web applications and server interactions.

Associated functions: creating scatter plots, line plots, bar charts, heatmaps, and more.

Alternatives: Snowflake, MemSQL, Hadoop, TensorFlow and PyTorch.

11. Speedml Library

We just need to import SpeedML and we can create a large variety of plots, clean the data, add and drop features, etc.

SpeedML imports and properly initializes the popular ML packages which are already defined in it including pandas, numpy, sklearn, xgboost, and matplotlib.

In this article, we will see how we can use SpeedML to analyze the data and prepare it for machine learning models.

Installing SpeedML :

pip install speedml

from speedml import Speedml

Initializing SpeedML :

sml = Speedml('dia_train.csv', 'dia_test.csv',target='Outcome')

Exploratory Data Analysis :

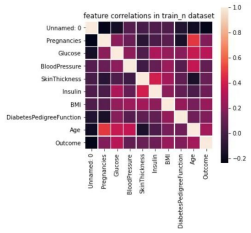
#data sample

sml.train.head()

	Unnamed: 0	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	0	6	148	72	35	0	33.6	0.627	50	0
1	1	1	85	66	29	0	26.6	0.351	31	1
2	2	8	183	64	0	0	23.3	0.672	32	1
3	3	1	89	66	23	94	28.1	0.167	21	1
4	4	0	137	40	35	168	43.1	2.288	33	1

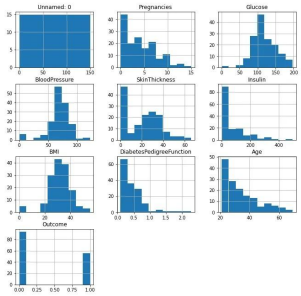
#Heatmap

sml.plot.correlate()



#Distribution Plot

sml.plot.distribute()



Similarly, we can create plots like continuous, ordinal, etc. Next, we will see how we can manipulate data and change it according to our requirements.

#Finding Feature density

sml.feature.density('Age')

sml.train[['Age', 'Age_density']].head()

	Age	Age_density
0	50	2
1	31	7
2	32	4
3	21	10
4	33	6

End Note :

Speedml to speed up the process of starting machine learning and various functions that are related to EDA and visualizations. Speedml is easy to use and most of the functions are callable using a single line of code which saves time and effort.

12. Dataprep

Why dataprep ?

DataPrep can be used to address multiple data-related problems, and the library provides numerous features through which every problem can be solved and taken care of.

Using the DataPrep Library, one can collect data from multiple data sources using the dataprep.connector module, perform intense exploratory analysis using the dataprep.eda module and clean and standardize datasets using the dataprep.clean module.

DataPrep automatically detects and highlights the insights present in the data, such as missing data, distinct count and statistics.

A whole detailed profile report can be created in a matter of seconds by using just a single line of code, which makes it ten times faster than other libraries to perform data preparation or EDA on.

Installation :

!pip install dataprep

Dependencies :

import pandas as pd

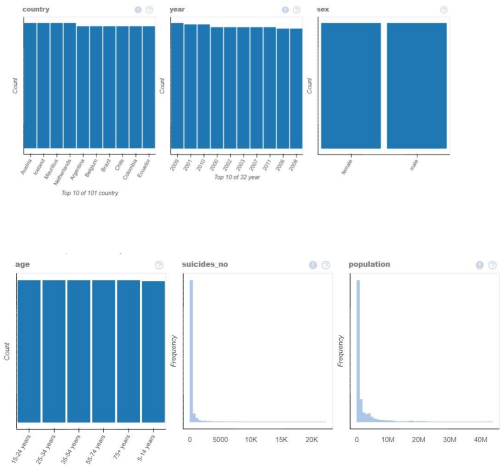
from dataprep.eda import plot, plot_correlation, plot_missing

As we want to create different plots from our dataset, we have imported plot, plot_correlation to create correlation graphs,plot_missing to plot the number of missing data.

Creating Visualizations :

Using DataPrep, we can create all the possible visualizations for the data using just a single line of code. Let us plot our loaded data and see what it looks like

```
df["year"] = df["year"].astype("category")
plot(df)
```



As we can see, the library itself detects the data and plots all the necessary data graphs in a single-window itself!

Auto Insight Generation :

You can also get a detailed plot for a single column with all its statistics to understand the column better,

```
plot(df, "gdp_per_capita ($)")
```

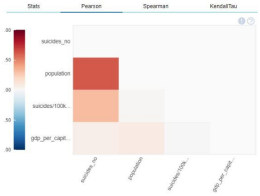
We also create a dataframe, taking care of the missing values and then create a correlation plot,

df_without_missing = df.dropna('columns')

plot_correlation(df_without_missing)

plot_correlation(df_without_missing, k=1)

plot_correlation(df_without_missing, value_range=(0,1)



Whether it be Pearson, Spearman or Kendall-Tau, any correlation graph can be easily plotted using the DataPrep library.

Final note :

importance of Data Preparation in Big Data Analytics and the necessary steps required to do so. We also explored a library known as DataPrep, and tested its different functionalities that might help during the Data Preparation and EDA phase

Contribution:

Jatan : worked on Numpy, Pandas, Matplotlib and Seaborn

Anushka : worked on Scikit-learn, Statsmodels and Plotly

Ajay : worked on SciPy, Bokeh and yellowbrick

Vedant : worked on the Speedml and Dataprep