CHAPTER 12

Matplotlib and Seaborn

It is critical to be able to plot the observations and variables of a dataset before subjecting the dataset to some machine learning algorithm or another. Data visualization is essential to understand your data and to glean insights into the underlying structure of the dataset. These insights help the scientist in deciding with statistical analysis or which learning algorithm is more appropriate for the given dataset. Also, the scientist can get ideas on suitable transformations to apply to the dataset.

In general, visualization in data science can conveniently be split into **univariate** and **multivariate** data visualizations. Univariate data visualization involves plotting a single variable to understand more about its distribution and structure, while multivariate plots expose the relationship and structure between two or more variables.

Matplotlib and Seaborn

Matplotlib is a graphics package for data visualization in Python. Matplotlib has arisen as a key component in the Python data science stack and is well integrated with NumPy and Pandas. The **pyplot** module mirrors the MATLAB plotting commands closely. Hence, MATLAB users can easily transit to plotting with Python.

Seaborn, on the other hand, extends the Matplotlib library for creating beautiful graphics with Python using a more straightforward set of methods. Seaborn is more integrated for working with Pandas DataFrames. We will go through creating simple essential plots with Matplotlib and seaborn.

Pandas Plotting Methods

Pandas also has a robust set of plotting functions which we will also use for visualizing our dataset. The reader will observe how we can easily convert datasets from NumPy to Pandas and vice versa to take advantage of one functionality or the other. The plotting features of Pandas are found in the **plotting** module.

E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*,
https://doi.org/10.1007/978-1-4842-4470-8_12

There are many options and properties for working with **matplotlib**, **seaborn**, and **pandas.plotting** functions for data visualization, but as is the theme of this material, the goal is to keep it simple and give the reader just enough to be dangerous. Deep competency comes with experience and continuous usage. These cannot really be taught.

To begin, we will load Matplotlib by importing the **pyplot** module from the **matplotlib** package and the **seaborn** package.

```
import matplotlib.pyplot as plt
import seaborn as sns
```

We'll also import the **numpy** and **pandas** packages to create our datasets.

```
import pandas as pd
import numpy as np
```

Univariate Plots

Some common and essential univariate plots are line plots, bar plots, histograms and density plots, and the box and whisker plot, to mention just a few.

Line Plot

Let's plot a sine graph of 100 points from the negative to positive **exponential** range. The **plot** method allows us to plot lines or markers to the figure. The outputs of the sine and cosine line plot are shown in Figure 12-1 and Figure 12-2, respectively.

```
data = np.linspace(-np.e, np.e, 100, endpoint=True)
# plot a line plot of the sine wave
plt.plot(np.sin(data))
plt.show()
# plot a red cosine wave with dash and dot markers
plt.plot(np.cos(data), 'r-.')
plt.show()
```