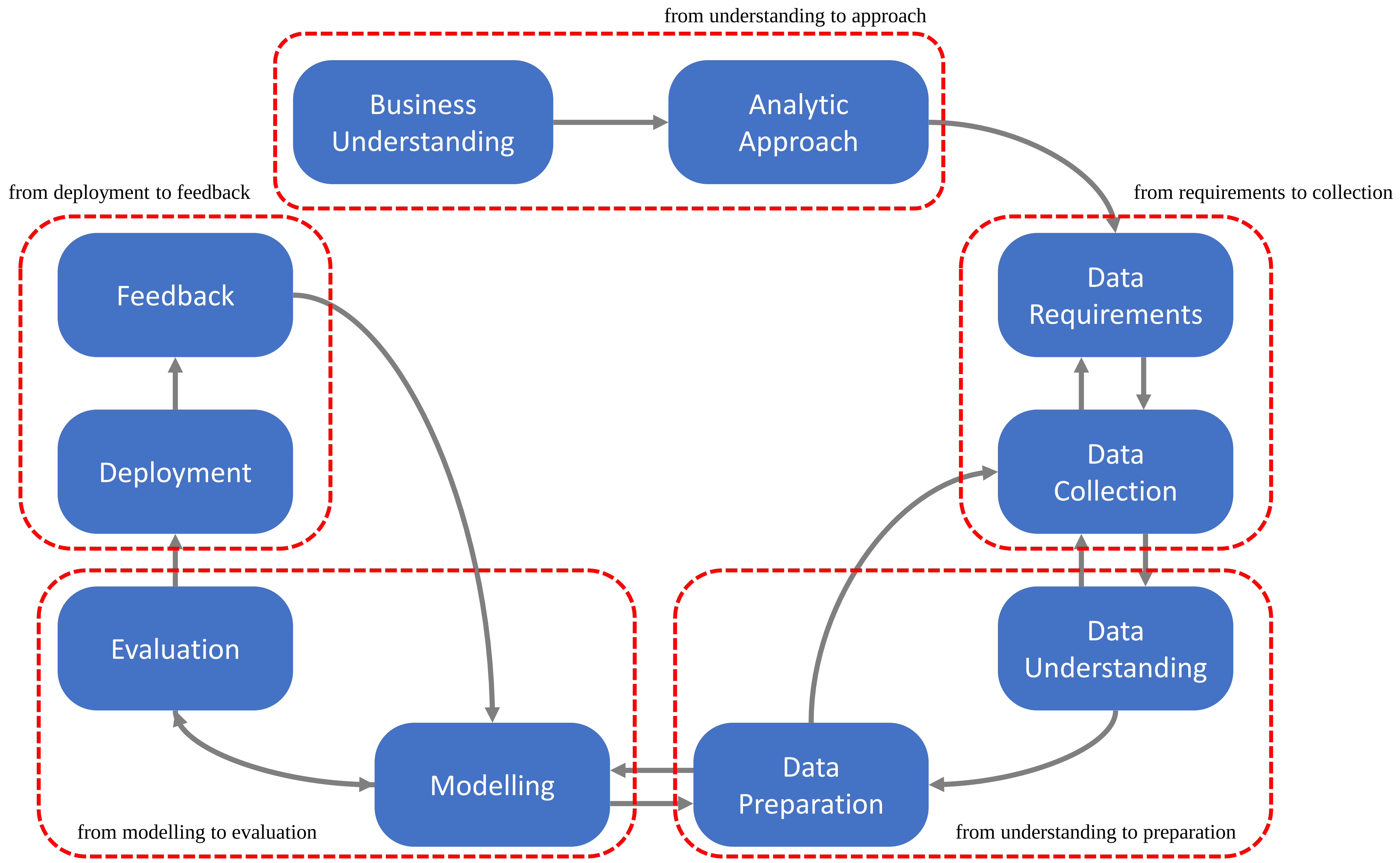
from understanding to approach
Business Understanding
Analytic Approach
from deployment to feedback
from requirements to collection
Feedback
Data Requirements
Deployment
Data Collection
Evaluation
Data Understanding
Modelling
Data Preparation
from modelling to evaluation
from understanding to preparation

An area plot also known as an area chart or graph is a type of plot that depicts accumulated totals using numbers or percentages over time. It is based on the line plot and is commonly used when trying to compare two or more quantities. So how can we generate an area plot with Matplotlib?

Histograms

A histogram is a way of representing the frequency distribution
of a numeric dataset. The way it works is it partitions the spread of the
numeric data into bins, assigns each datapoint in the dataset to a bin, and then
counts the number of datapoints that have been assigned to each bin.
Now, the first step in creating the histogram is partitioning the horizontal axis in, say, ten bins of equal width, and then we construct the histogram by counting how many datapoints have a value that is between the limits of the first bin, the second bin, the third bin, and so on.

Basic Plotting with Matplotlib

Matplotlib is a well-established data visualization library that is well supported in different environments such as in Python scripts, in the Python shell, web application servers, in graphical user interface toolkits as well as the Jupyter notebook.

Line Plots

Line plot is a plot in the form of a series of data points connected by straight line segments. It is one of the most basic types of charts and is common in many fields not just data science.

Scatter Plots

A scatter plot is a type of plot that displays values pertaining to typically two variables against each other. Usually, it is a dependent variable to be plotted against an independent variable in order to determine if any correlation between the two variables exists. For example, here is a scatter plot of income versus education, and by looking at the plotted data, one can conclude that an individual with more years of education is likely to earn a higher income than an individual with fewer years of education. So how can we create a scatterplot with Matplotlib?

Waffle Charts

A waffle chart is a great way to visualize data in relation to a whole or to highlight progress against a given threshold. Unfortunately, Matplotlib does not have a built-in function to create waffle charts.

Word Clouds

A word cloud is simply a depiction of the importance of different words in the body of text. A word cloud works in a simple way; the more a specific word appears in a source of textual data the bigger and bolder it appears in the world cloud. So given some text data on recruitment, for example, we generate a cloud of words like this. This cloud is telling us that words such as recruitment, talent, candidates, and so on, are the words that really stand out in these text documents. And assuming that we didn't know anything about the content of these documents, a word cloud can be very useful to assign a topic to some unknown textual data. Unfortunately, just like waffle charts, Matplotlib does not have a built-in function to generate word clouds.

Introduction to Data Visualization

Data visualization is a way to show a complex data in a form that is graphical and easy to understand this can be especially useful when one is trying to explore the data and getting acquainted with it also since a picture is worth a thousand words then plots and graphs can be very effective in conveying a clear description of the data especially when disclosing findings to an audience or sharing the data with other pure data scientists also they can be very valuable when it comes to supporting any recommendations.

Introduction to Matplotlib

Matplotlib is one of the most widely used, if not the most popular data visualization library in Python. It was created by John Hunter, who was a neurobiologist and was part of a research team that was working on analyzing Electrocorticography signals, ECoG for short. The team was using a proprietary software for the analysis. However, they had only one license and were taking turns in using it. So, in order to overcome this limitation, John set out to replace the proprietary software with a MATLAB based version that could be utilized by him and his teammates, and that could be extended by multiple investigators. As a result, Matplotlib was originally developed as an ECoG visualization tool, and just like MATLAB, Matplotlib was equipped with a scripting interface for quick and easy generation of graphics, represented by pyplot. We will learn more about this in a moment. Now Matplotlib's architecture is composed of three main layers: the back-end layer, the artist layer where much of the heavy lifting happens and is usually the appropriate programming paradigm when
writing a web application server, or a UI application, or perhaps a script to be shared with other developers, and the scripting layer, which is the appropriate
layer for everyday purposes and is considered a lighter scripting interface
to simplify common tasks and for a quick and easy generation of graphics and
plots.

Bar Charts

A bar chart is a very popular visualization tool. Unlike a histogram, a bar chart also known as a bar graph is a type of plot where the length of each bar is proportional to the value of the item that it represents. It is commonly used to compare the values of a variable at a given point in time. For example, say we're interested in visualizing in a discrete fashion how immigration from Iceland to Canada looked like from 1980 to 2013. One way to do that is by building a bar chart where the height of the bar represents the total immigration from Iceland to Canada in a particular year. So how do we do that with Matplotlib.

Pie Charts

A pie chart is a circular statistical graphic divided into slices to illustrate numerical proportion. For example, here is a pie chart of the Canadian federal election back in 2015 were the Liberals in red won more than 50% of the seats in the House of Commons. That is why the red color occupies more than half of the circle.

Box Plots

A boxplot is a way of statistically representing the distribution of given data through 5 main dimensions. The first dimension is minimum, which is the smallest number in the sorted data. Its value can be obtained by subtracting 1.5 times the IQR where IQR is interquartile range from the first quartile. The second dimension is first quartile which is 25% of the way through the sorted data.

Seaborn and Regression Plots

Seaborn makes creating plots very efficient. Therefore, with Seaborn you
can generate plots with code that is 5 times less than with Matplotlib.
With Seaborn, you can do all this with literally one line of code. The
way to do this, we first import Seaborn and let's import it as sns. Then, we call
the Seaborn regplot function. We basically tell it to use the dataframe
df_total and to plot the column year on the horizontal axis and
the column total on the vertical axis.

Introduction to Folium

Folium, you can create a map of any location in the world as long as you
know its latitude and longitude values. You can also create a map and
superimpose markers as well as clusters of markers on top of the map for cool
and very interesting visualizations. You can also create maps of different styles
such as street level map, stamen map, and a couple others which we will look
into in just a moment.

Choropleth Maps

A choropleth map is a thematic map in which areas are shaded
or patterned in proportion to the measurement of the statistical variable
being displayed on the map, such as population density or per capita income.
The higher the measurement the darker the color.

Dashboarding Overview

With real-time visuals on the dashboard, understanding business moving parts becomes easy. Based on the report type and data, suitable graphs and charts can be created in one central location. This provides an easy way for stakeholders to understand what is going right, wrong, and what needs to be improved. Getting the big-picture in one place can help businesses make informed decisions. This improves business performance.
In general, the best dashboards answer critical business questions.

A data scientist should be able to create and deliver a story around the finding
in a way stakeholders can easily understand. With that in mind, dashboards are the way to go.

Let's take a look at web-based dashboarding tool options available in Python.
Dash is a python framework for building web analytic applications. It is written on top
of Flask, Plotly.js, and React.js.

Introduction to Plotly

Plotly python is built on top of Plotly JavaScript library and includes chart
types like statistical, financial, maps, scientific, and 3-dimensional data.
The web-based visualizations created using Plotly python can be displayed in Jupyter notebook, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash. The focus of this lesson will be on two of the Plotly sub-modules: Plotly Graph Objects and Plotly Express Plotly Graph Objects is the low-level
interface to figures, traces, and layout. The Plotly graph objects module provides
an automatically generated hierarchy of classes (figures, traces, and layout) called graph objects.

These graph objects are used for representing figures with a top-level class Plotly. graph objects. Figure. Plotly express is a high-level wrapper for Plotly. It is a recommended starting point for creating most common figures provided by Plotly using a simpler syntax. It uses graph objects internally.

Introduction to Dash

Dash is a Open-Source User Interface Python library for creating reactive, web-based applications. It is enterprise-ready and a first-class member of Plotly’s open-source tools. Dash applications are web servers running Flask and communicating JSON packets over HTTP requests.
Dash’s frontend renders components using React.js. It is easy to build a Graphical User Interface
using dash as it abstracts all technologies required to build the applications.
Dash is Declarative and Reactive. Dash output can be rendered in web
browser and can be deployed to servers. Dash uses a simple reactive decorator
for binding code to the UI. This is inherently mobile and cross-platform ready.
Let's say you are planning to create an application to answer a business question.
As a first step, you need to determine the layout of the application.
Decide which chart to use and where to place for example. This is called `layout` part in dash. The second part is to add interactivity to the application. There are two components of Dash First is `Core components` We can import core components as dcc using this import statement Next is `HTML Components` We can import html components as
html using this import statement Let's explore these further. The dash_html_components library has a component for every HTML tag.

Make dashboards interactive

First, create a function that will perform operations to return the desired result for the output component. Decorate the callback function with @app. callback decorator. This takes two parameters. Output: This sets result returned from the callback function to a component id Input: This set input provided to the callback function to a component id from here we will connect input and output to desired properties. We will see this in action with an example using the airline data. The use case here is to extract the top 10 airline carriers in the provided input year selected by the number of flights. Based on the input year, the output will change. First, we import the required packages. As seen before, we will import pandas, dash, dash core, and HTML components. The new entry here is dash dependencies.
From dash dependencies, we will import input and output that we will use in the callback function. We read the airline data into the pandas dataframe. We load our dataframe at the start of the app and can be read inside the callback function.
We will start designing the dash application layout by adding components.
First, we will provide the title to the dash app using the HTML heading component H1 and style it using the style parameter. Next, we will add an HTML division and text input Core component. In-Dash, the inputs
and outputs of the application are simply the properties of a particular component.