**Introduction to Matplotlib**

**Data visualisation** is an important skill to possess for anyone trying to extract and communicate insights from data. Great business narratives and presentations often stem from brilliant visualisations that convey the key ideas in a concise and aesthetic manner. In the field of machine learning, visualisation plays a key role throughout the entire process of analysis - to obtain relationships, observe trends and portray the final results as well. Data visualisation is a crucial step in the process of data analysis

## **The Necessity of Data Visualisation**

**“There are three kinds of lies: lies, damned lies, and statistics.” - Mark Twain**

It is essential to appreciate why it is so important to ‘look’ at the data from the perspective of plots and graphs. To begin with, it is difficult for the human eye to decipher patterns from raw numbers only. Sometimes, even the statistical information summarised from the data may mislead you to wrong conclusions. Therefore, you should visualise the data often to understand how different features are behaving. Data Visualization helps us understand flows and structures in an intuitive way.

**Facts and Dimensions**

Graphics and visuals, when used intelligently and innovatively, can convey a lot more than what raw data alone can. Matplotlib serves the purpose of providing multiple functions to build graphs from the data stored in your lists, arrays, etc.

There are two types of data that helps to create charts and plots effectively.

* Facts
* Dimensions

Facts and dimensions are different types of variables that help you interpret data better. Facts are numerical data, and dimensions are metadata. Metadata explains the additional information associated with the factual variable. Both facts and dimensions are equally important for generating actionable insights from a given data set. For example, in a data set about the height of students in a class, the height of the students would be a fact variable, whereas the gender of the students would be a dimensional variable. You can use dimensions to slice data for easier analysis.

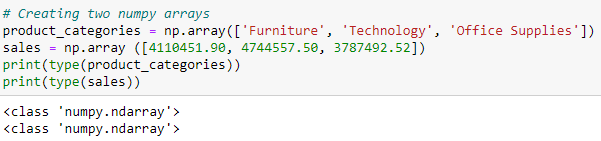
Plots are used to convey different ideas. For example, you can use certain plots to visualise the spread of data across two variables and other plots to gauge the frequency of a label. Depending on the objective of your visualisation task, you can choose an appropriate plot.

**Bar Graph**

To get the insights of sales across each category. We are using a Bar Graph as it uses the bars to show comparisons between the categories of data. First and foremost, NumPy and Matplotlib libraries should be imported, the sub package pyplot is used to build plots and graphs.



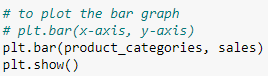
Let’s have two NumPy arrays:

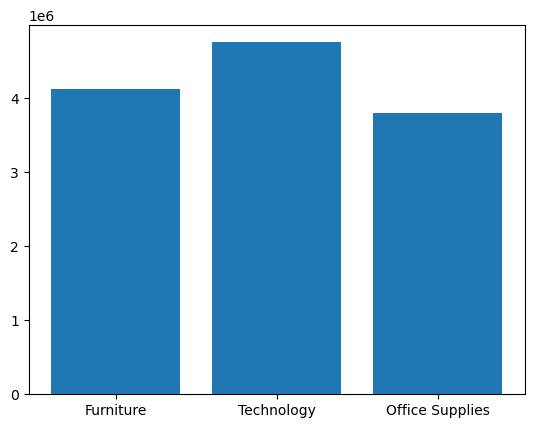


Matplotlib allows you to use a simple and intuitive workflow to create plots. The important Matplotlib commands used in the video above are as follows:

* plt.bar (x\_component, y\_component): Used to draw a bar graph
* plt.show(): Explicit command required to display the plot object

Plotting the Bar Graph:





A bar graph is helpful when there is a need to visualise a numeric feature (fact) across multiple categories. In the example covered we plotted the sales amount (numeric feature) under three different product categories. Using the bar graph, could easily distinguish between the performance of these categories.

The following code can be used to add the title and x, y labels to the graph.

* plt.xlabel(), plt.ylabel(): Specify labels for the x and y axes
* plt.title(): Add a title to the plot object

To make the charts more appealing different attributes such as font size and colour can be used. Adding labels and a title to plot helps the audience interpret the graphs easily and also relays the required information to the viewer. For example, you can use the following code to change the values and ticks on the x and y axes of a graph:

plt.yticks(tick\_values, tick\_labels)



