

# Building Interactive Dashboards



Sharad B Nalawade

# Trainer Profile: Sharad Nalawade

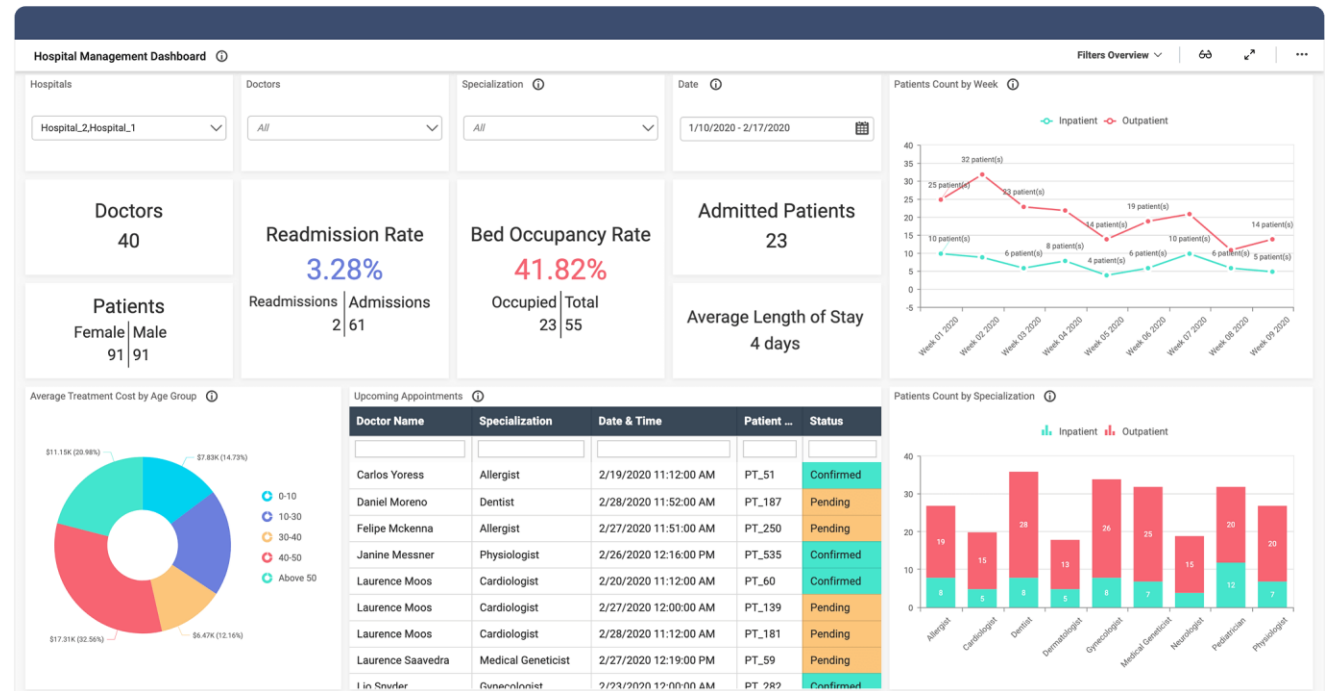
---



- 35 years in the IT industry
- Founder – ***Ultitude Consulting***, India
- President – ***Ambharii Labs***, USA
- Stints at ***Cisco, Accenture, Symantec, Wipro, C-DoT and CMC Ltd.***
- Consultant on emerging technologies – Data Science, Machine Learning, Cloud Computing, RPA, Digital Transformation, Solution Design and Enterprise Architecture
- Author of three books on Science and Technologies
- Visiting Faculty at IIT-Jodhpur, IIM-Rohtak, IIM-Indore, IIM-Lucknow

# What is an Interactive Dashboard?

- A single window to critical performance parameters, be it business or operations
- It helps to monitor and track changes as they happen
- A great tool for date driven decision making



# Differences between Static and Interactive Dashboard

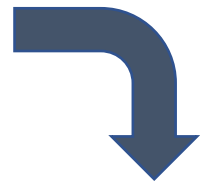
<b>Static</b>	<b>Interactive</b>
Data is stale unless updated manually	Data is real time, as it happens
Answers specific questions	Answers any question interactively across multiple levels
Need to generate multiple reports to get for different data sources	Combined view of multiple data sources
Needs time for analysis	Faster decision making
Static interface	Visually dynamic
Limited trend and what-if analysis	Explorative data analytics and storytelling

# Top Dashboard Building Tools/Languages

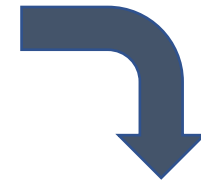


# Know your customer through their data

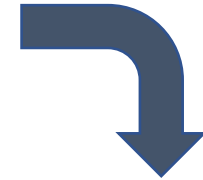
Know your audience



Collect relevant datasets

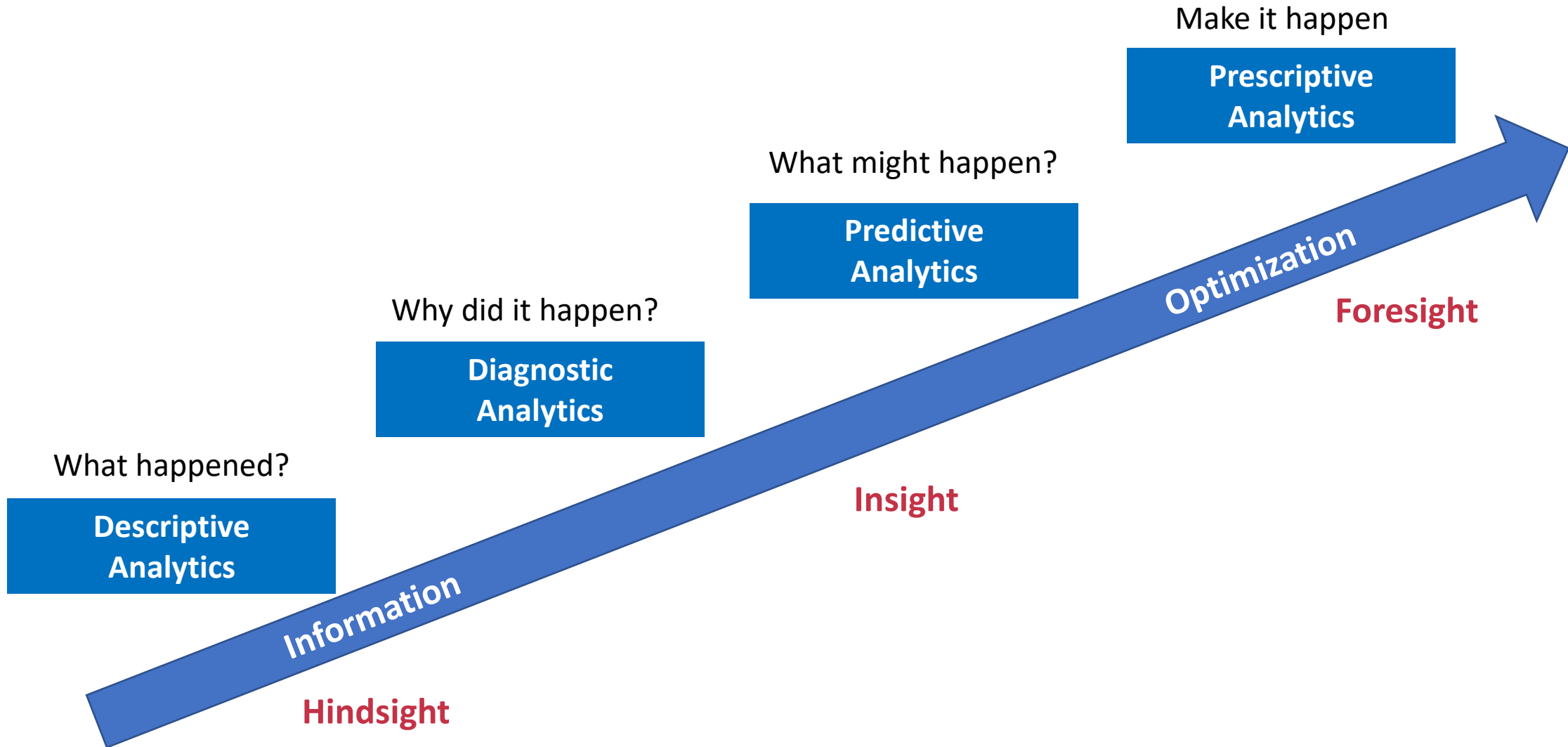


Choose the right analysis

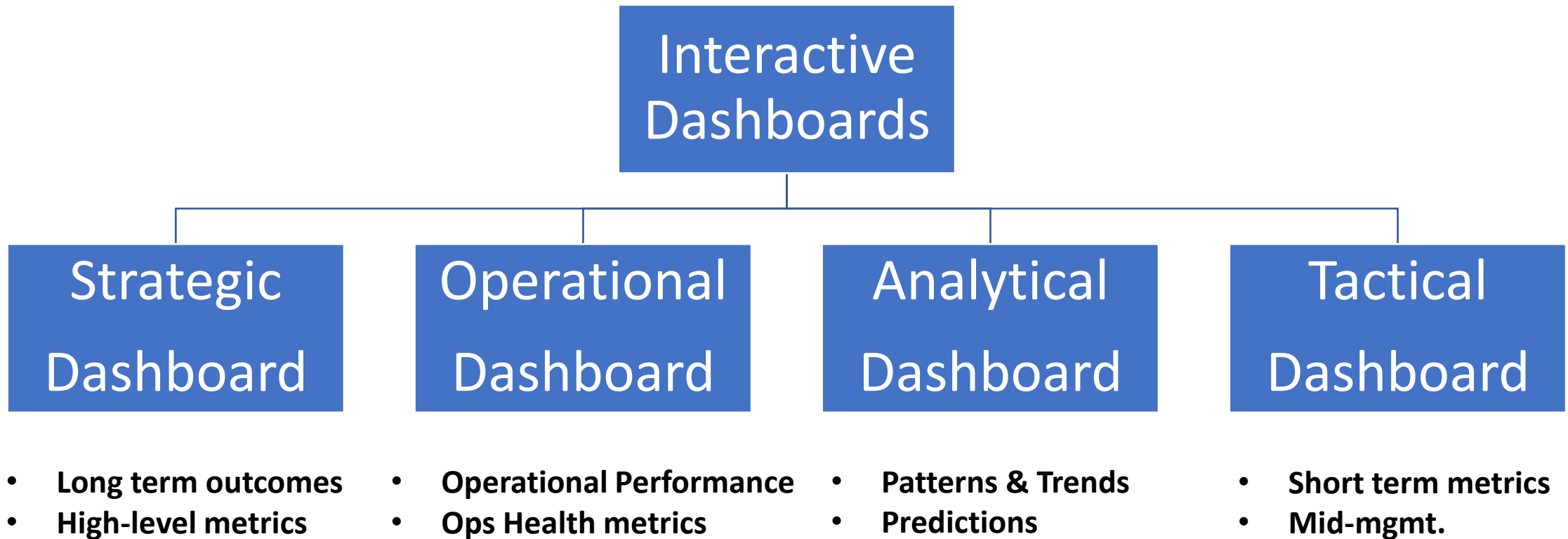


Choose the right chart

# Types of Analytics



# Types of Dashboards



# Role based Dashboards

SN	Dashboard	KPIs
1	Management Dashboard	<ul style="list-style-type: none"> <li>Quarterly Revenue</li> <li># of customers added</li> <li>Avg revenue per customer</li> <li>Customer Acquisition Cost</li> </ul>
2	CMO Dashboard	<ul style="list-style-type: none"> <li># Leads generated (MQL &amp; SQL)</li> <li>Conversion Rate</li> <li>Avg Ad spend per customer</li> </ul>
3	CFO Dashboard	<ul style="list-style-type: none"> <li>.....</li> </ul>
4	Manufacturing Production Dashboard	<ul style="list-style-type: none"> <li>Quantity/Volume produced per a time period</li> <li>Product backlogs</li> <li># Defects</li> <li>Manufacturing Cost 6</li> </ul>
5	CTO Dashboard	
6	Talent Management Dashboard	<ul style="list-style-type: none"> <li>Demand versus Supply</li> <li>Talent attrition rate</li> <li>Talent Rating</li> </ul>
7	Hospital Admin Dashboard	-----

# Building Interactive Dashboards with Python



Setting up Python Environment

# Setting up Python Environment

- **Anaconda** is a open-source Data science platform
- Includes popular Python packages like **NumPy** and **Pandas**
- **Jupyter Notebook** – Open source web based interface
- **JupyterLab** – Full-pledged IDE for Python
- Both support Interactive interface with code, narrative text and visualization

# Setting up Python Environment



Install  
Anaconda

Run  
Command  
Line

Open  
JupyterLab

# Setting up Python Environment

After installing Anaconda, go to Anaconda command line prompt and launch JupyterLab

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\akars>jupyter lab_
```

# Setting up Python Environment

The image shows a web browser window displaying the JupyterLab interface. The browser's address bar shows the URL `localhost:8888/lab`. The JupyterLab interface includes a top menu bar with options: File, Edit, View, Run, Kernel, Tabs, Settings, and Help. Below the menu is a 'Launcher' tab. The main content area is titled 'Training/IIM-Rohtak Data Visualization' and contains three options: 'Notebook' (with a notebook icon), 'Python 3' (with the Python logo), and 'Console' (with a terminal icon). The bottom status bar shows '0 \$ \_ 0' and a gear icon. The Windows taskbar at the bottom displays the system tray with weather information (20°C Cloudy), search, and various application icons. The system clock shows the time as 17:30 on 02/11/2022.

# What is Dash



<https://dash.plotly.com/introduction>

Developed by **Plotly** company

Dash is the original low-code framework for rapidly building data apps in Python, R, Julia, and F# (experimental).

Dash is built on top of **Plotly, Flask and React**

Dash apps are rendered in the web browser. You can deploy your apps to VMs or Kubernetes clusters and then share them through URLs.

Since Dash apps are viewed in the web browser, Dash is inherently cross-platform and mobile ready.

# What is Dash

Built on top of **Plotly.js**, **React**, and **Flask**, Dash ties modern UI elements like ***dropdowns***, ***sliders*** and ***graphs*** directly to your analytical python code.

Dash has two major components: ***Layout*** and ***Interactivity***

**1. dash\_core\_components** – For interactivity (*dropdowns, radio buttons, sliders, graphs, etc.* using **callback functions**)

**2. Dash\_html\_components** – HTML tags for app layout

Dash apps use ***callback functions***: functions that are automatically called by Dash whenever an input component's property changes, in order to update some property in another component (the output).

# Interactive Dashboards Demos

Marketing Lead Trends Dashboard

<https://www.inetsoft.com/sree/app/viewer/view/global/gallery/Marketing%20Leads>

Call Centre Team Performance Dashboard

<https://www.inetsoft.com/sree/app/viewer/view/global/gallery/Call%20Center%20Analysis>

Call Centre Monitoring Dashboard

<https://www.inetsoft.com/sree/app/viewer/view/global/gallery/Call%20Center%20Monitoring>

Executive Sales Summary Dashboard

[https://www.inetsoft.com/evaluate/bi\\_visualization\\_gallery/dashboard.jsp?dbIdx=8](https://www.inetsoft.com/evaluate/bi_visualization_gallery/dashboard.jsp?dbIdx=8)

Global Visual Dashboard

[Global Visual Analysis \(inetsoft.com\)](https://www.inetsoft.com)

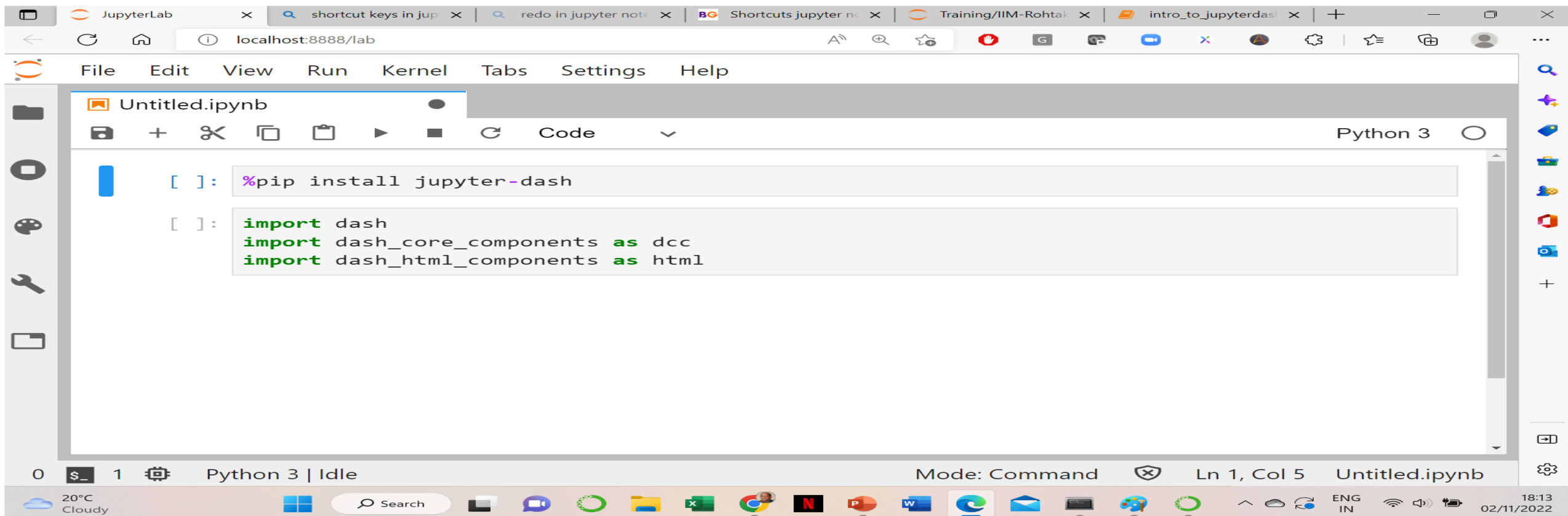
Source: Inetsoft.com

# Why Dash

- Various data visualization to dashboards
- Interactive
- Highly customizable
- Easy to share
- Open source

<https://dash.gallery/Portal/>

# Setting up Python Dash Environment



The screenshot shows the JupyterLab interface in a web browser. The address bar shows 'localhost:8888/lab'. The main area contains a code editor with two code cells. The first cell contains the command to install 'jupyter-dash'. The second cell contains Python code to import 'dash', 'dash\_core\_components' (aliased as 'dcc'), and 'dash\_html\_components' (aliased as 'html'). The interface includes a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help), a toolbar with icons for file operations, and a status bar at the bottom showing 'Python 3 | Idle', 'Mode: Command', 'Ln 1, Col 5', and 'Untitled.ipynb'. The Windows taskbar is visible at the bottom of the screen.

```
[ ]: %pip install jupyter-dash

[ ]: import dash
import dash_core_components as dcc
import dash_html_components as html
```

You could also use PyCharm or Visual Studio Code editors

# HTML Components

## **Dash HTML Components – dash.html**

## **Dash Core Components – dash.dcc**

HTML Documents contain following elements:

- Section Headings: h1, h2, h3, h4, h5, h6
- Paragraph: p
- Content division: div
- Line break: br
- Anchor: a

# HTML Tags

- HTML tags define the HTML elements
- Examples:

```
<p>This is a paragraph</p>
```

```
<h1>This is Heading 1</h1>
```



start tag



content



end tag

Practice Code: [Create Dashboard to take you to Flipkart.com](#)

# Core Dash Components Module

- Higher level components that can be interactive
- Written with JavaScript, HTML, and CSS etc
- `from dash import dcc`
  
- Examples:
  - Dropdowns: `dcc.Dropdown`
  - Radio items: `dcc.RadioItems`
  - Graphs: `dcc.Graph`
  - Text input boxes: `dcc.Input`
  - Sliders and Range Sliders: `dcc.Slider` and `dcc.RangeSlider`

Practice Code: Create 'World Happiness Dashboard'

# Callback Functions

Call-back functions are automatically called when an input component's property changes, and updates the output component's property's

## Examples:

- Input box's value property updates output text

Change this text<sup>Ⓜ</sup>

Text: Change this text

# Interactive Dashboard with Panel



# Panel

[Panel](#) provides tools for easily composing widgets, plots, tables, and other viewable objects and controls into custom analysis tools, apps, and dashboards.

Panel works with visualizations from [Altair](#), [Bokeh](#), [HoloViews](#), [Matplotlib](#), [Plotly](#), [pydeck](#), [PyVista](#) and many other Python plotting libraries

[panel · PyPI](#)

Questions?