

Project Explanation

Step 1 – Introduction

- Start by providing a brief overview of the project, emphasizing its significance.
- Mention the problem or question the project aimed to address.
- Highlight the business context and objectives that led to the initiation of the project.

Step 2 - Data Collection

- Gather relevant data from various sources.
- This may involve obtaining data from databases, APIs, files, or other means.

Step 3 Data Cleaning and Preprocessing

- Clean and preprocess the raw data to handle missing values, outliers, and inconsistencies.
- This step is crucial for ensuring the quality of the data used for analysis.

Step 4 Exploratory Data Analysis (EDA)

- Perform exploratory data analysis to understand the characteristics of the data.
- Use visualizations, and summary metrics to identify patterns, trends, and outliers.

Step 5 Data Transformation

- Conduct data transformations and manipulations to create derived variables or features that enhance the analysis.
- This step may involve aggregations, merging datasets, or creating new variables to gain deeper insights.

Step 6 Model Deployment (if applicable)

- Describe the process of deploying the model in a production environment.
- Highlight the impact of the deployed model on business processes.

Step 7 Results and Impact

- Clearly communicate the results of your analysis and the impact on the business or problem at hand.
- Use quantitative metrics to demonstrate the success of the project.

Step 8 Challenges and Learnings

- Discuss any challenges you faced during the project and how you overcame them.
- Highlight key learnings and improvements for future projects.

Project Explanation – Covid 19 Impact Analysis

Introduction

- The "Covid-19 Impact Analysis" project uses Python's Dash library web deployment to create an interactive dashboard.
- This dashboard visually represents key Covid-19 metrics such as total cases, active cases, recovered cases, and total deaths according to different states of India.
- It also analyzes the availability of essential items like masks, sanitizers, and oxygen. The goal is to make complex pandemic data easy to understand through interactive visuals.

Data Collection

- I gathered a curated dataset from a CSV file, encompassing comprehensive information on Covid-19 cases.

Libraries Used

- numpy and pandas for data manipulation and analysis.
- plotly.graph_objs and plotly.express for creating interactive plots.
- dash for web application development.
- dash_core_components, dash_html_components for creating the layout components.
- Input, Output from dash.dependencies for handling callbacks.

Data Loading

- The project starts by loading a dataset (state_wise_daily.csv) using Pandas, containing information on Covid-19 cases.

- The dataset is structured to include data on different statuses like Confirmed, Recovered, and Deceased.

Dashboard Layout

HTML

- HTML (HyperText Markup Language) is the standard markup language for creating web pages.
- In the context of this project, HTML is used to define the structure of the dashboard, dividing it into various sections.

Dash Components

- Dash provides a Python framework for building interactive web applications. Dash components are used to create the different elements of the dashboard.

Divs and Classes:

- Divs (Division):
 - Divs are HTML elements used to group and structure content on a web page.
 - In this project, divs are employed to create distinct sections of the dashboard, separating content logically.
- Classes:
 - Classes in HTML are used to apply styles to multiple elements with the same design.
- The top section displays overall Covid-19 statistics with dynamically updating total cases, active cases, recovered cases, and total deaths.
- The second section presents line charts showing the trend of essential commodities (mask, sanitizer, oxygen) over different statuses (Confirmed, Recovered, Deceased).
- The third section showcases a dropdown-enabled pie chart illustrating the distribution of Covid-19 cases in different zones (Red, Blue, Green, Orange).
- The final section allows users to choose a status (Confirmed, Hospitalized, Recovered, Deceased) from a dropdown to view a bar chart depicting the total count for each state.

Styling and External Dependencies

- Bootstrap is utilized for external styling to enhance the visual appeal of the dashboard.
- External stylesheet links are added to the Dash app for improved styling.

Running the Application

- The application is launched using `app.run_server(debug=True)`
- The application is launched using `app.run_server(debug=True)`. The dashboard opens in the web browser.

Challenges Faced and Solutions

- Challenge: Implementing dynamic updates for graphs based on user interactions required careful handling of callbacks and data dependencies.
- Dash's callback functionality was used effectively.

Conclusion

This project not only enhances my skills in data analysis and web development but also serves as a practical application of data analytics and science in addressing real-world challenges.