DSDA

PBI Dashboard Sample Project-related questions

ML Project Sample Project-related questions

PBI Dashboard

# PBI Dashboard

## Areas covered in Project Explanation:

When Learners are introducing a Power BI dashboard project during their project explanation, they should include the following key elements in their introduction:

1. Project Overview

* **Project Title**: Clearly state the name of the project.
* **Objective**: Explain the purpose of the dashboard. For example, "The dashboard provides insights into sales trends and helps in decision-making for business growth."
* **Problem Statement**: Briefly describe the problem or business need the dashboard aims to solve. For example, "The sales team struggled to track real-time performance across regions, making it difficult to identify key growth areas."

2. Data Description

* **Data Source**: Mention where the data comes from.
* **Data Volume**: Provide an estimate of the dataset size or number of records.
* **Data Fields**: Highlight key fields or variables used (e.g., "sales, revenue, customer demographics, product categories").
* **Data Cleaning**: Briefly touch on steps taken to prepare the data, such as handling missing values or duplicate records.

3. Target Audience

* Define who will use the dashboard and how it will help them. For example, "The dashboard is designed for the sales team to monitor their performance and for senior management to identify high-performing regions."

4. Key Features

* Highlight the main features of the dashboard. For example:
  + Real-time updates
  + Drill-down functionality
  + Interactive filters
  + KPIs (e.g., total revenue, profit margin, etc.)
  + Trend analysis or predictions

5. Tools and Techniques

* Specify the tools used in the project (e.g., "Power BI for visualization, SQL for querying data, Excel for data preprocessing").
* Mention any advanced techniques or features used, such as DAX calculations, measures, or advanced visualizations.

6. Project Scope and Limitations

* **Scope**: Define what the project covers (e.g., "Analysis of regional sales data for the past year").
* **Limitations**: Acknowledge constraints, such as data granularity or unavailability of certain datasets.

7. Outcome/Expected Results

* Share the key insights or value provided by the dashboard. For example, "The dashboard identifies top-performing regions and products, enabling the sales team to focus on high-potential areas."

8. Future Enhancements (Optional)

* Suggest possible improvements or additional features. For example, "Integration with predictive analytics to forecast future sales trends."

By covering these elements, the Learner provides a comprehensive and professional introduction to their Power BI dashboard project.

## EXAMPLE SCENARIO (SALES DASHBOARD)

**Faculty:** Can you briefly introduce the project you worked on?  
**Learner:** The project is called the **"Sales Performance Dashboard."** My main goal was to create an interactive tool that helps track **sales** performance, spot trends, and ultimately empower **sales** teams to make data-driven decisions. It's all about making the process easier and faster, so they can focus more on strategy and less on manual reporting.

**Faculty:** What problem were you trying to solve with this project?  
**Learner:** The challenge I was addressing is that manual **sales** reporting is really time-consuming and often doesn’t provide real-time insights. This dashboard automates much of the process, so instead of spending hours compiling data, managers and executives can get immediate, actionable insights with just a few clicks.

**Faculty:** Can you explain the data you used and how you prepared it?  
**Learner:** I used a dataset from Kaggle, which includes **sales information like revenue, product details, regions, and time periods.** The data needed a little cleaning—there were missing values and duplicates to clean up. I used Python and libraries like Pandas to make sure everything was in top shape before feeding it into the dashboard. I wanted to ensure that the insights were as accurate as possible.

**Faculty:** Who’s the intended audience for this dashboard, and how does it benefit them?  
**Learner:** The dashboard was designed primarily for **sales managers and executives**. It helps them keep an eye on how their **sales** teams are performing in real-time, which regions or products are doing well, and where improvements are needed. In short, it makes their decision-making process much faster and more data-driven.

**Faculty:** What are the key features of your dashboard?  
Learner**:** There are several features I’m really proud of:

1. **KPIs:** I’ve included key performance indicators like Revenue, Profit Margin, and Units Sold.
2. **Drill-down capabilities:** Users can explore data by region, product, or time period.
3. **Charts:** For visualizing regional performance—easy to spot trends at a glance.
4. **Trend analysis:** Showing sales growth, which helps executives forecast and plan better.

**Faculty:** What outcomes did this project achieve?  
**Learner:** The dashboard has really helped improve decision-making. Managers are able to quickly identify top-performing regions and products, and they can act on this information almost immediately. It’s been a great tool for refining sales strategies and improving overall performance.

**Faculty:** How did you ensure the dashboard is easy to use?  
**Learner:** I wanted to make sure it was intuitive, so I kept things simple and clean. The key metrics are upfront, and all the drill-down options are easy to find. I also used tooltips and legends to explain the data points, so users don’t have to spend time figuring out what they’re looking at. My goal was for someone with no technical background to be able to use it with ease.

**Faculty:** What was the biggest challenge you faced, and how did you overcome it?  
**Learner:** One big challenge was making sure the data was clean and reliable. Sometimes the dataset had missing or inconsistent values, and that can really throw off the analysis. I spent quite a bit of time writing Python scripts to clean it up and ensure everything was accurate. Another challenge was making sure the dashboard performed smoothly with large datasets, but I optimized the queries and minimized the amount of data being processed at once to keep things fast and responsive.

**Faculty:** How do you see this dashboard evolving in the future?  
**Learner:** I think one of the next steps would be to integrate predictive analytics so that the dashboard can forecast future trends. This would give the sales team even more foresight when planning their strategies. I’d also love to add mobile capabilities, so the sales managers can access the dashboard on the go, which is something they’ve asked for a lot.

**Faculty:** What’s the most important takeaway from this project for you?  
**Learner:** For me, the biggest lesson is the power of visualizing data. It's one thing to have a lot of data, but turning it into a tool that can actually inform decisions is where the real value lies. This project really reinforced how data-driven decisions can be a game-changer for a business.

## Example Introduction for a Sales Dashboard

**Project Title**: **Sales Performance Dashboard**

**My Objective** was to create an interactive dashboard for tracking sales performance, identifying trends, and aiding data-driven decisions.

**Talking about my Problem Statement**: Manual sales reporting is time-consuming and lacks real-time insights. This dashboard addresses the need for automated, visual sales analysis to support decision-making.

Data was taken from kaggle, including sales amount, regions, products, and time dimensions. Cleaning involved handling missing values and duplicates.

I have designed this for sales managers and executives to monitor performance and identify areas for improvement.

**Key Features include**:

* KPIs: Revenue, Profit Margin, Units Sold.
* Drill-down by region, product, and time.
* Heatmap for regional performance.
* Trend analysis for sales growth.

**Outcome**:  
The dashboard highlights key insights, like top-performing regions and products, enabling better sales strategies.

ML Project

# ML Project

## Areas Covered in Machine Learning Project Explanation

When Learners introduce a Machine Learning (ML) project during their project explanation, they should include the following key elements in their introduction:

1. Project Overview

* **Project Title**: Clearly state the name of the project.  
   Example: *"Customer Churn Prediction Model"*
* **Objective**: Briefly explain the purpose of the project.  
   Example: *"The goal of this project is to predict whether a customer is likely to stop using a company's services based on historical data."*
* **Problem Statement**: Describe the problem or business need being addressed.  
   Example: *"Companies face significant revenue loss due to customer churn but often struggle to identify at-risk customers in time to take preventive actions."*

2. Data Description

* **Data Source**: Mention the source of the data.  
   Example: *"The dataset was obtained from Kaggle and contains customer demographics, subscription details, and usage patterns."*
* **Data Volume**: Provide an estimate of the size or number of records.  
   Example: *"The dataset includes 10,000 customer records with 20 features."*
* **Key Variables**: Highlight the main features used.  
   Example: *"Customer age, monthly charges, contract type, payment method, and tenure."*
* **Data Cleaning**: Briefly describe preprocessing steps, such as handling missing values, outlier removal, and feature engineering.

3. Target Audience

Define who benefits from the model and its purpose.  
 Example: *"This model is designed for customer success teams and marketing departments to proactively address customer retention by identifying high-risk accounts."*

4. Machine Learning Workflow

* **Model Selection**: Describe the approach taken.  
   Example: *"We used a classification model to predict churn, testing algorithms like Logistic Regression, Decision Trees, and Random Forests."*
* **Features and Techniques**:  
   Example: *"Feature scaling was applied to normalize data, and one-hot encoding was used for categorical variables. Feature selection was done using mutual information."*
* **Train-Test Split**: Explain how the data was divided.  
   Example: *"The dataset was split into 80% training and 20% testing to ensure the model was generalizable."*

5. Model Evaluation

* **Metrics Used**: Mention the metrics used to measure performance.  
   Example: *"The model was evaluated using accuracy, precision, recall, F1-score, and ROC-AUC."*
* **Results**: Share the outcomes of the model.  
   Example: *"The Random Forest model achieved an accuracy of 88% and an AUC of 0.92, making it the best-performing model."*

6. Project Scope and Limitations

* **Scope**: Define the project’s boundaries.  
   Example: *"The project focuses on predicting churn for customers of a single telecom provider based on historical data."*
* **Limitations**: Acknowledge constraints.  
   Example: *"The dataset may not capture behavioral trends that change over time, and external factors influencing churn (like competitors) are not included."*

7. Outcome/Expected Results

Explain the value delivered by the project.  
 Example: *"The model helps identify at-risk customers early, allowing the company to target retention campaigns more effectively and reduce churn by an estimated 15%."*

## Example Scenario: Customer Churn Prediction Project

**Faculty**: Can you briefly introduce the project you worked on?  
 **Learner**: The project is titled "Customer Churn Prediction Model." The main goal was to predict which customers are likely to stop using a company's services, helping the business reduce churn and improve retention strategies.

**Faculty**: What problem were you trying to solve with this project?  
 **Learner**: Churn is a critical issue for businesses, especially in subscription-based services. The problem we addressed was the inability to identify at-risk customers in time to take preventive action, leading to significant revenue loss.

**Faculty**: Can you explain the data you used and how you prepared it?  
 **Learner**: The dataset was from Kaggle, containing customer details like tenure, payment method, contract type, and monthly charges. We handled missing values using mean imputation and removed outliers using IQR. Categorical variables were encoded with one-hot encoding, and feature scaling was applied for numerical variables.

**Faculty**: What algorithms and techniques did you use?  
 **Learner**: We tested multiple models, including Logistic Regression, Random Forest, and XGBoost. After hyperparameter tuning, Random Forest performed the best, achieving an accuracy of 88%. We also used cross-validation to ensure the model's robustness.

**Faculty**: What were the key outcomes of the project?  
 **Learner**: The model identifies at-risk customers with high accuracy, allowing the retention team to focus on those accounts. This reduces churn by prioritizing interventions, potentially saving the company significant revenue.

**Faculty**: How do you see this project evolving?  
 **Learner**: In the future, I’d like to integrate customer interaction data from emails and call logs using natural language processing to enhance prediction accuracy. Also, deploying the model into a live dashboard for real-time churn prediction would be a great next step.

**Faculty**: What’s the most important takeaway from this project?  
 **Learner**: The biggest lesson is understanding how critical data preprocessing is for model performance. Also, the importance of selecting the right metric for evaluation based on the problem. For churn prediction, precision and recall were more critical than just accuracy.

## Example Introduction for a Machine Learning Project

**Project Title:** Customer Churn Prediction Model

The objective of this project is to develop a machine-learning model that predicts whether a customer is likely to stop using a company's services. This helps businesses proactively reduce churn and improve customer retention strategies.High customer churn rates are a major challenge for subscription-based businesses, leading to significant revenue loss. However, identifying at-risk customers early enough to take preventive action is often difficult. This project addresses this issue by creating a predictive model based on historical customer data.

The dataset includes 10,000 customer records with 20 features like Contract type, internet service, payment method, Monthly charges, tenure, and number of support tickets.

The primary users of this model are customer success managers and marketing teams. The model helps identify high-risk accounts, enabling timely interventions such as offering discounts, personalized outreach, or enhanced support to retain customers.

FSD

FSD- Java Sample Project-related questions

FSD - Python Sample Project-related questions

Java

# Areas Covered in Full-Stack Development - Java Project Explanation

When Learners introduce a Full-Stack Development (FSD) project during their project explanation, they should include the following key elements in their introduction:

1. Project Overview

**Project Title:** Clearly state the name of the project.  
 Example: *"E-commerce Website for Electronics"*

**Objective:** Briefly explain the purpose of the project.  
 Example: *"The goal of this project is to create a fully functional e-commerce platform that allows users to browse, search, and purchase electronic products while providing a seamless shopping experience."*

**Problem Statement:** Describe the problem or business need being addressed.  
 Example: *"Small businesses face challenges in reaching wider audiences and managing sales. This project aims to provide an online platform for electronics retailers to increase their visibility and streamline sales."*

2. Features and Functionalities

Highlight the main features of the application.  
 **Example Features:**

* **User Authentication:** Secure login and registration for users and admin.
* **Product Catalog:** Displays a dynamic catalog of products with categories and filters.
* **Shopping Cart:** Add, update, and remove products in the cart.
* **Payment Gateway Integration:** Simulated payment processing for order completion.
* **Admin Panel:** Manage product inventory, view orders, and generate sales reports.

3. Development Workflow

**Frontend Development:**

* Designed intuitive user interfaces with Bootstrap for responsiveness.
* Implemented dynamic elements like product filters and real-time cart updates using JavaScript.

**Backend Development:**

* Built RESTful APIs using Spring Boot for operations like product management, user authentication, and order processing.
* Implemented business logic for cart operations, discount calculations, and order history.

**Database Design:**

* Designed normalized tables in MySQL for products, users, orders, and payment records.
* Wrote efficient SQL queries for product search and order history retrieval.

4. Testing and Deployment

**API Testing:**

* Used Postman to test backend APIs for operations like product retrieval and order placement.

**Frontend Testing:**

* Validated user input forms and ensured responsiveness on multiple devices.

**Deployment:**

* Deployed the application on a local Apache Tomcat server, integrated with the MySQL database for live functionality.

5. Project Scope and Limitations

**Scope:** Define the boundaries of the project.  
 Example: *"The project is designed for small electronics retailers and focuses on providing basic e-commerce functionalities."*

**Limitations:** Acknowledge constraints.  
 Example: *"Currently, the payment gateway is simulated, and advanced features like real-time delivery tracking and customer reviews are not included."*

6. Outcome/Expected Results

Explain the value delivered by the project.  
 Example:

* *"Enables small businesses to transition online, increasing their customer reach."*
* *"Streamlines the shopping experience for users and automates sales processes for admins."*
* *"Provides a scalable platform that can be enhanced with additional features in the future."*

# Example Scenario: E-commerce Website for Electronics

**Faculty:** Can you briefly introduce the project you worked on?  
 **Learner:** The project is titled *"E-commerce Website for Electronics."* The main goal was to develop an online platform that allows users to browse, search, and purchase electronic products, while providing an admin panel for inventory management.

**Faculty:** What problem were you trying to solve with this project?  
 **Learner:** Many small electronics retailers lack an online presence, which limits their customer reach. This project provides a solution by creating a scalable and user-friendly platform for managing and selling products online.

**Faculty:** Can you explain the technology stack you used?  
 **Learner:** We used Java with Spring Boot for the backend to handle business logic and APIs. The frontend was built using HTML, CSS, Bootstrap, and JavaScript for an interactive user experience. MySQL was used for managing the database, and Postman helped test the backend APIs.

**Faculty:** What features does your application include?  
 **Learner:** Key features include user authentication, a product catalog with filtering options, a shopping cart for managing purchases, a payment gateway for simulated transactions, and an admin panel for managing products and orders.

**Faculty:** How did you ensure the application was secure and functional?  
 **Learner:** For security, we used hashed passwords for user authentication and restricted sensitive operations to admin roles. Functionality was tested using Postman for APIs and manual testing for the frontend interface.

**Faculty:** What were the key outcomes of the project?  
 **Learner:** The platform enables businesses to manage products and orders efficiently while providing users with an intuitive shopping experience. It bridges the gap between offline retailers and digital consumers.

**Faculty:** How do you see this project evolving?  
 **Learner:** In the future, I’d like to integrate real payment gateways, implement a recommendation system for products, and add features like customer reviews and real-time order tracking.

**Faculty:** What’s the most important takeaway from this project?  
 **Learner:** The biggest lesson was understanding the importance of seamless integration between frontend and backend, as well as the role of efficient database design in building scalable applications.

Python

# Areas Covered in Full-Stack Development - Python Django Project Explanation

When Learners introduce a Full-Stack Development (FSD) project during their project explanation, they should include the following key elements in their introduction:

1. Project Overview

**Project Title:** Clearly state the name of the project.  
 Example: *"E-commerce Platform for Electronics"*

**Objective:** Briefly explain the purpose of the project.  
 Example: *"The goal of this project is to build a robust e-commerce platform that allows users to browse and purchase electronics while providing features like user authentication, a shopping cart, and an admin panel for inventory management."*

**Problem Statement:** Describe the problem or business need being addressed.  
 Example: *"Many small electronics retailers struggle to reach customers beyond their local market. This project provides an online platform to expand their business and streamline their sales process."*

2. Features and Functionalities

Highlight the main features of the application.  
 **Example Features:**

* **User Authentication:** Secure user registration, login, and role-based access for admin and customers.
* **Product Catalog:** Dynamic product listing with categories, search, and filtering options.
* **Shopping Cart:** Add, update, and remove items from the cart with real-time price calculations.
* **Order Management:** Users can place orders, track their status, and view order history.
* **Payment Gateway Integration:** Simulated payment processing for order checkout.
* **Admin Panel:** Manage products, orders, and generate sales reports.

3. Development Workflow

**Frontend Development:**

* Designed responsive UI using HTML, CSS, Bootstrap, and JavaScript.
* Integrated dynamic elements such as product filtering and live cart updates using JavaScript.

**Backend Development:**

* Developed using Django to build robust models, views, and controllers for managing business logic.
* Created RESTful APIs with Django REST Framework (DRF) for product, cart, and order operations.
* Implemented session-based authentication and permissions for secure access.

**Database Design:**

* Used SQLite/MySQL for storing product, user, order, and payment data.
* Designed normalized tables and relationships between entities like products, users, and orders.
* Used Django ORM for efficient data manipulation.

4. Testing and Deployment

Testing:

* **Unit Testing:** Wrote test cases for individual Django views and models.
* **API Testing:** Used Postman to test REST APIs for CRUD operations on products, users, and orders.
* **Frontend Testing:** Manually tested user forms, UI responsiveness, and functionality across devices.

Deployment:

* Deployed the application on Heroku, connected to a live PostgreSQL database.
* Configured Django settings for production, including static file management and environment variables.

5. Project Scope and Limitations

**Scope:** Define the boundaries of the project.  
 Example: *"This project is designed for small electronics retailers who need an online presence and basic e-commerce functionality."*

**Limitations:** Acknowledge constraints.  
 Example: *"Currently, the platform uses simulated payment processing and lacks advanced features like real-time order tracking or AI-powered recommendations."*

6. Outcome/Expected Results

Explain the value delivered by the project.  
 Example:

* *"Provides small businesses with an online platform to expand their customer base."*
* *"Automates the shopping and sales process, saving time for both customers and admins."*
* *"Offers a scalable and customizable framework that can be enhanced with additional features."*

# Example Scenario: E-commerce Platform for Electronics

**Faculty:** Can you briefly introduce the project you worked on?  
 **Learner:** The project is titled *"E-commerce Platform for Electronics."* Its goal was to create an online shopping platform where users can browse and purchase electronic products while providing an admin interface for inventory management.

**Faculty:** What problem were you trying to solve with this project?  
 **Learner:** Small electronics retailers often lack the tools to reach a broader market. This project bridges that gap by offering a scalable and user-friendly platform for online sales and inventory management.

**Faculty:** Can you explain the technology stack you used?  
 **Learner:** We used Django for the backend, which provided a powerful framework for rapid development. The frontend was built using HTML, CSS, Bootstrap, and JavaScript for responsiveness and interactivity. The database was managed using MySQL, and APIs were tested with Postman.

**Faculty:** What features does your application include?  
 **Learner:** Key features include secure user authentication, a searchable product catalog, a shopping cart for managing purchases, simulated payment processing, and an admin panel for managing products and orders.

**Faculty:** How did you ensure the application was secure and functional?  
 **Learner:** We implemented session-based authentication to secure user data and restricted admin functionalities to authorized users. Testing was done using Django's built-in testing tools and Postman for API validation.

**Faculty:** What were the key outcomes of the project?  
 **Learner:** The platform offers a seamless shopping experience for users and efficient tools for admins to manage their inventory and orders. It helps small businesses expand their reach and optimize operations.

**Faculty:** How do you see this project evolving?  
 **Learner:** In the future, I would like to add real payment gateways, implement product recommendations using machine learning, and integrate features like customer reviews and order tracking.

**Faculty:** What’s the most important takeaway from this project?  
 **Learner:** The biggest lesson was understanding how Django's modular approach simplifies building scalable applications. I also learned the importance of testing APIs and the role of database optimization in improving performance.

Project List

### **DATA SCIENCE PROJECTS**

1. **Netflix Movie Recommendation System**
   * Dataset: Kaggle Netflix Movies Dataset
2. **Weather Analysis and Visualization**
   * Dataset: NOAA Weather Data
3. **Exploratory Data Analysis (EDA) on Titanic Dataset**
   * Dataset: Kaggle Titanic Dataset
4. **Stock Market Analysis (Predicting Trends)**
   * Dataset: Yahoo Finance Data
5. **House Price Prediction**
   * Dataset: Kaggle Ames Housing Dataset
6. **Learner Performance Analysis**
   * Dataset: UCI Learner Performance Dataset
7. **Customer Churn Analysis**
   * Dataset: Kaggle Customer Churn Dataset
8. **Iris Flower Classification**
   * Dataset: UCI Iris Dataset
9. **COVID-19 Data Analysis**
   * Dataset: Johns Hopkins COVID-19 Dataset
10. **Car Price Prediction**

* Dataset: Kaggle Car Price Dataset

1. **E-commerce Sales Analysis**
   * Dataset: Kaggle E-commerce Dataset
2. **Fraud Detection in Credit Card Transactions**
   * Dataset: Kaggle Credit Card Fraud Detection Dataset
3. **Retail Store Customer Segmentation**
   * Dataset: UCI Mall Customers Dataset
4. **HR Analytics for Employee Retention**
   * Dataset: Kaggle HR Analytics Dataset
5. **Airline Passenger Satisfaction Analysis**
   * Dataset: Kaggle Airline Passenger Satisfaction Dataset
6. **Sentiment Analysis on Product Reviews**
   * Dataset: Amazon Product Reviews
7. **Loan Eligibility Prediction**
   * Dataset: Kaggle Loan Prediction Dataset
8. **Flight Price Prediction**
   * Dataset: Kaggle Flight Price Dataset
9. **Heart Disease Prediction**
   * Dataset: UCI Heart Disease Dataset
10. **Predicting Diabetes in Patients**
    * Dataset: Kaggle Diabetes Dataset

### 

1. **Customer Lifetime Value Prediction**
   * Dataset: Online Retail Dataset
2. **Natural Language Processing (Text Summarization)**
   * Dataset: CNN/Daily Mail Dataset
3. **Real-Time Traffic Prediction**
   * Dataset: Uber Movement Data
4. **Time Series Analysis on Energy Consumption**
   * Dataset: UCI Household Energy Dataset
5. **Cancer Detection with Image Analysis**
   * Dataset: Kaggle Histopathologic Cancer Dataset
6. **Music Recommendation System**
   * Dataset: Million Song Dataset
7. **Predicting Employee Productivity using Time Tracking Data**
   * Dataset: Custom Time Tracking Datasets
8. **Object Detection for Self-Driving Cars**
   * Dataset: KITTI Vision Benchmark Dataset
9. **Fraudulent News Detection**
   * Dataset: Kaggle Fake News Dataset
10. **Electric Vehicle Charging Pattern Analysis**
    * Dataset: Kaggle Electric Vehicles Dataset
11. **Movie Recommendation System**
12. **YouTube Trending Video Analysis**
13. **Language Detection Tool**
14. **Twitter Sentiment Analysis**
15. **Chatbot with NLP**
16. **E-commerce Purchase Prediction**
17. **Spotify Playlist Recommendation**
18. **Real-Time Stock Market Dashboard**
19. **Personal Finance Dashboard**
20. **Food Delivery Analysis**
21. **Healthcare Analytics**
22. **Traffic Flow Prediction**
23. **Sales Forecasting**
24. **Supply Chain Optimization**
25. **Sports Performance Analysis**
26. **Electricity Demand Forecasting**
27. **Weather Pattern Prediction**
28. **Image Classification for Plant Disease Detection**
29. **Air Quality Prediction**
30. **Customer Satisfaction Prediction for Banks**

**DATA ANALYSIS PROJECT**

### **Beginner-Level Projects**

1. **Sales Dashboard for a Retail Store**
   * Dataset: Kaggle Superstore Dataset
   * Key Insights: Sales trends, product performance, and region-wise sales.
2. **COVID-19 Global Impact Dashboard**
   * Dataset: Johns Hopkins COVID-19 Dataset
   * Key Insights: Case trends, recovery rates, and region-wise spread.
3. **Employee Attrition Analysis**
   * Dataset: Kaggle HR Analytics Dataset
   * Key Insights: Identify factors leading to employee turnover.
4. **Learner Performance Dashboard**
   * Dataset: UCI Learner Performance Dataset
   * Key Insights: Analyze grades based on study hours, parental education, and school type.
5. **Customer Demographics Analysis**
   * Dataset: UCI Mall Customers Dataset
   * Key Insights: Group customers based on spending habits and demographics.
6. **E-commerce Performance Dashboard**
   * Dataset: Instacart Market Basket Dataset
   * Key Insights: Sales by category, user purchase patterns, and repeat orders.
7. **Stock Market Trends Dashboard**
   * Dataset: Yahoo Finance Data
   * Key Insights: Price trends, sector-wise performance, and volume analysis.
8. **Real Estate Market Analysis**
   * Dataset: Kaggle Housing Prices Dataset
   * Key Insights: Price distribution, factors affecting prices, and region-wise trends.
9. **Flight Delay Analysis**
   * Dataset: Kaggle Flight Delay Dataset
   * Key Insights: Identify causes and trends of delays across airlines and seasons.
10. **Hospital Patient Analysis**

* Dataset: Hospital Readmissions Dataset
* Key Insights: Admission trends, readmission rates, and hospital performance.

1. **Global Energy Consumption Dashboard**
   * Dataset: World Bank Energy Dataset
   * Key Insights: Analyze trends in energy consumption and production globally.
2. **Financial KPI Dashboard for Companies**
   * Dataset: Kaggle Financial Data
   * Key Insights: Monitor revenue, profit margins, and cash flow over time.
3. **Customer Churn Analysis**
   * Dataset: Kaggle Customer Churn Dataset
   * Key Insights: Visualize churn trends and key factors influencing customer retention.
4. **Air Pollution Analysis**
   * Dataset: WHO Air Quality Data
   * Key Insights: Trends in air quality index (AQI) by region and sources of pollution.
5. **Sports Performance Dashboard**
   * Dataset: Kaggle IPL Dataset
   * Key Insights: Player performance, team rankings, and match statistics.
6. **Crime Rate Analysis by City**
   * Dataset: US Crime Data
   * Key Insights: Trends in crime rates, types of crimes, and affected areas.
7. **Healthcare Resource Allocation Dashboard**
   * Dataset: Healthcare Cost and Utilization Data
   * Key Insights: Visualize resource utilization trends by department and treatment type.
8. **Travel and Tourism Trends**
   * Dataset: World Tourism Data
   * Key Insights: Tourist arrivals, spending habits, and country rankings.
9. **Education Trends Dashboard**
   * Dataset: UNESCO Education Dataset
   * Key Insights: Literacy rates, Learner enrollment, and teacher-Learner ratios.
10. **Electric Vehicle Market Analysis**
    * Dataset: EV Market Data
    * Key Insights: EV adoption trends, manufacturer performance, and region-wise analysis.
11. **YouTube Content Analysis**
    * Dataset: Kaggle YouTube Trending Dataset
    * Key Insights: Trends in content categories, views, and engagement.
12. **Netflix Viewer Patterns Dashboard**
    * Dataset: Netflix Dataset
    * Key Insights: Genre popularity, viewer trends, and release patterns.
13. **Online Food Delivery Analysis**
    * Dataset: Kaggle Food Delivery Dataset
    * Key Insights: Restaurant performance, customer preferences, and delivery trends.
14. **Music Listening Trends Dashboard**
    * Dataset: Spotify Data
    * Key Insights: Popular genres, artists, and listening trends by region.
15. **Retail Product Performance Analysis**
    * Dataset: Kaggle Walmart Dataset
    * Key Insights: Seasonal sales trends, top-performing products, and region-wise insights.
16. **Banking Customer Segmentation**
    * Dataset: Kaggle Banking Dataset
17. **Insurance Claim Analysis**
    * Dataset: Insurance Data
18. **Logistics and Supply Chain Dashboard**
    * Dataset: Kaggle Logistics Data
19. **Restaurant Performance Analysis**
    * Dataset: Restaurant Sales Data
20. **Healthcare Cost Analysis**
    * Dataset: Kaggle Healthcare Cost Data

### **Tools to Use:**

* **Power BI**: Build interactive dashboards with slicers, filters, and dynamic charts.
* **Tableau**: Use Tableau’s storytelling feature for narrative insights.
* **Data Sources**: Combine data from Excel, SQL, and APIs for advanced dashboards.

**FSD PROJECTS**

### 

1. **Personal Portfolio Website**
2. **To-Do List Application**
3. **Blog Website**
4. **Weather App (using APIs)**
5. **Quiz App with Score Tracking**
6. **Basic Calculator (Web-based)**
7. **Online Resume Builder**
8. **Expense Tracker**
9. **URL Shortener**
10. **Digital Clock**
11. **E-commerce Website (with Cart and Payment Integration)**
12. **Movie Review System (using Sentiment Analysis)**
13. **Library Management System**
14. **Social Media App (with Post and Comment Features)**
15. **Online Polling/Voting System**
16. **Chat Application (using WebSockets)**
17. **Fitness Tracker with Goals**
18. **Event Management System**
19. **Photo Gallery with Image Upload**
20. **Job Portal**
21. **CRM System (Customer Relationship Management)**
22. **Employee Management System**
23. **Learner Management System**
24. **Online Examination Portal**
25. **Course Registration System**
26. **Attendance Management System**
27. **Stock Price Tracker (using APIs)**
28. **Hospital Management System**
29. **Real-Time Weather Dashboard**
30. **Online Bookstore**
31. **Online Car Store**
32. **Content Management System (CMS)**
33. **Expense Splitter (for Groups)**
34. **Online Food Ordering System**
35. **AI-Powered Chatbot for a Website**
36. **Document Management System**
37. **Secure Login and Authentication System (using JWT)**
38. **Auction System for Online Bidding**
39. **Travel Booking Portal**
40. **E-learning Platform**
41. **News Aggregator Website (using APIs)**
42. **Recipe Sharing Website**
43. **Music Playlist Manager**
44. **Virtual Whiteboard for Collaborative Drawing**
45. **Language Translation App (using APIs)**
46. **Photo Editor (with Filters)**
47. **Meme Generator**
48. **Virtual Flashcards for Learning**
49. **Habit Tracker**
50. **Trivia Game with Leaderboard**
51. **Portfolio Analyzer for Cryptocurrency**

### **Tech Stack:**

For these projects, Learners can use:

* **Frontend:** HTML, CSS, JavaScript (or frameworks like React.js)
* **Backend:** Python Django / Java Springboot
* **Database:** MySQL, PostgreSQL, or MongoDB
* **APIs:** REST APIs, third-party APIs (like OpenWeather, Google Maps)
* **Deployment:** Docker, AWS, or Heroku