**CAPSTONE PROJECT**

**DOCUMENTATION**

**End-to-End Data Analytics Project**

**Course Data Analysis**

**Name :** Nivash C

**Course :** Data Analytics

**Batch Number :** 26

**Mode of Study :** Offline

**Roll Number :** 160324CBR004

**Table of Contents**

1. Introduction

2. Aim

3. Business Problem / Problem Statement

4. Project Workflow

5. Web Scraping with Python

6. Data Cleaning in PowerQuery

7. Storing Data in MySQL Database

8. Creating a Power BI Dashboard

9. Writing Insights and Final Report

10. Conclusion

**1. Introduction**

The project involves an end-to-end data analytics solution aimed at extracting valuable insights from edX course data. The process begins with web scraping to collect various course details such as titles, ratings, enrollments, and pricing. The scraped data is then stored in a structured format within a MySQL database, which serves as the backend for data analysis and visualization. Finally, an interactive Power BI dashboard is developed to present the data in a visually compelling manner, allowing users to explore insights into course popularity, pricing trends, and more.

**2. Aim**

The primary aim of this project is to analyze course data from the edX platform to provide actionable insights into online education trends. By leveraging web scraping, database management, and data visualization tools, the project seeks to inform educational institutions and learners about the dynamics of course offerings, helping stakeholders make data-driven decisions about course development, marketing strategies, and enrollment optimization.

**3. Business Problem / Problem Statement**

In the competitive landscape of online education, understanding the factors that drive course popularity and learner engagement is crucial. However, the vast amount of data available on platforms like edX is often underutilized due to the lack of systematic analysis. The business problem addressed by this project is the need for a comprehensive analysis of course characteristics, such as ratings, pricing, and enrollments, to uncover patterns and trends that can inform strategic decisions for course providers and learners. By identifying these trends, the project aims to enhance the value proposition of online courses, optimize pricing strategies, and improve learner satisfaction.

**4. Project Workflow**

The project workflow consists of the following key steps:

1. **Web Scraping with Python:**  
    Using Selenium and BeautifulSoup, the project begins with the extraction of course data from edX. This step involves navigating through multiple pages, collecting detailed information about each course, and saving the data in a structured format (CSV).
2. **Data Storage in MySQL Database:**  
    After data collection, the next step is to design and create a MySQL database schema to store the scraped data. The cleaned data is imported into the database, which serves as the foundation for further analysis and visualization.
3. **Data Visualization with Power BI:**  
    Power BI is connected to the MySQL database to create an interactive dashboard. This dashboard includes various visualizations such as charts and filters that allow users to explore course data dynamically. Key metrics such as average ratings, price distributions, and enrollment patterns are visualized to provide actionable insights.
4. **Insight Generation and Reporting:**  
    The final step involves analyzing the visualizations to generate insights into course trends. A comprehensive report is prepared, detailing the methodology, findings, and implications of the analysis. The project concludes with a presentation summarizing the key insights and recommendations for stakeholders.

**5. Web Scraping with Python**

**Overview:**

Web scraping is a method used to extract data from websites and store it in a structured format for further analysis. In this project, Python's powerful web scraping libraries, Selenium and BeautifulSoup, were utilized to collect detailed information on courses offered on the edX platform. This step is crucial as it forms the foundation for subsequent data analysis and visualization, providing raw data that reflects the current state of online education offerings.

**Objectives of Web Scraping:**

The primary objective of the web scraping process was to gather comprehensive data on edX courses, including:

**Course Name:** The title of the course, which provides insight into the subject matter and focus.

**Rating:** The average rating given by students, indicating the perceived quality of the course.

**Number of Ratings:** The total number of ratings, reflecting the level of student engagement.

**Duration:** The length of the course in weeks, which helps categorize courses based on time commitment.

**Enrollment Numbers:** The number of students enrolled, serving as a proxy for course popularity.

**Institution:** The organization offering the course, providing a way to analyze courses by provider.

**Subject:** The subject area of the course, allowing for analysis across different fields of study.

**Level**: The difficulty level of the course (e.g., beginner, intermediate), which is useful for segmenting the data.

**Language:** The language in which the course is offered, which is relevant for accessibility analysis.

**Price:** The cost of the course, either in local currency or USD, important for understanding pricing strategies.

**Tools and Libraries Used:**

**Selenium:** A powerful tool for automating web browsers, Selenium was used to navigate through the edX course pages, manage dynamic content loading, and handle JavaScript-rendered elements. Selenium’s ability to control a browser programmatically was essential for simulating human-like browsing behavior, such as clicking on course links and scrolling through pages.

**BeautifulSoup:** This library was used for parsing the HTML content obtained from Selenium. BeautifulSoup provides a simple way to extract information from HTML and XML documents, making it easier to navigate the parsed document tree and retrieve the required data fields.

**Python Regular Expressions (re module):** Regular expressions were used to extract specific patterns from text data, such as numerical values for ratings, enrollments, and prices.

**Pandas:** A data manipulation library, Pandas was used to store the scraped data in a structured format (DataFrame) and eventually save it as a CSV file, which is suitable for database import and further analysis.

**Process Details:**

1. **Initialization of WebDriver:** The web scraping process began by initializing a Selenium WebDriver for Chrome. This WebDriver was configured to automate the browsing actions, such as loading pages, clicking links, and waiting for content to load fully.

**Python Code**

from selenium import webdriver

driver = webdriver.Chrome()

1. **Navigating and Extracting Course Data:** The script iteratively navigated through the specified range of pages on the edX search results. For each page, the following steps were performed:

* **Page Load and Content Capture:** The script navigated to each page URL, waited for content to fully load using a sleep function to handle dynamic content, and captured the page source.
* **Parsing with BeautifulSoup**: The captured HTML content was parsed using BeautifulSoup to identify and extract course-specific data elements, such as titles, ratings, and enrollment numbers.

**Python Code**

# Load the webpage

driver.get(f"https://www.edx.org/search?tab=course&page={i}")

time.sleep(13) # Allow time for the page to fully load

# Parse the page content

soup = BeautifulSoup(driver.page\_source, "html.parser")

* **Data Extraction:** For each course card on the page:
* **Link Extraction:** Extracted the course link and navigated to the course detail page to gather more specific information.
* **Field Extraction:** Extracted relevant data fields using BeautifulSoup selectors. The extraction logic handled potential variations in HTML structure by including checks and fallback options (try-except blocks) for missing data.

**Python Code**

# Example of extracting course title and rating

con = soup.find('div', class\_="page course-info-page")

title.append(con.find('h1').string)

rating = con.find('div', class\_="h5 ml-1 mr-3 mb-0")

ratings.append(rating.get\_text(strip=True).split()[0] if rating else 'None')

* **Error Handling and Data Validation:** Implemented error handling to ensure that missing or malformed data did not cause the script to fail. For example, if a specific data field like ratings was not found, a default value ('None') was appended to the list, ensuring consistent data length and structure.

**Python Code**

# Handling missing enrollment data

try:

enroll = con.find('div', {'class': 'small', "data-test-id": "selector-subheading"}).text.strip(" ")

enrollment.append(int(''.join(re.findall(r'\d+', enroll))) if enroll else 'None')

except:

enrollment.append('None')

* **Storing Data in CSV Format:** After scraping all the required data, the results were stored in a Pandas DataFrame and saved to a CSV file named 'Edx.csv'. This structured data file served as the input for subsequent steps in the project, such as database integration and visualization.

**Python Code**

# Convert the collected data into a DataFrame and save to CSV

dict\_1 = {

"Course\_Name": title,

"Rating": ratings,

# ... other fields ...

}

final = pd.DataFrame(dict\_1)

final.to\_csv('Edx.csv', index=False)

1. **Challenges and Solutions:**

**Dynamic Content Loading**: edX course pages load content dynamically, which required careful management of wait times to ensure all content was fully loaded before scraping. This was handled using time.sleep() strategically after page loads and interactions.

**Variations in HTML Structure:** Some courses had missing fields or variations in how data was presented. This was mitigated by using flexible BeautifulSoup selectors and incorporating error handling to skip or substitute missing data points.

1. **Outcome:**

The web scraping step successfully extracted a comprehensive dataset of edX courses, capturing all relevant fields in a consistent and structured manner. This dataset provided the foundation for further analysis, enabling insights into course offerings, pricing, and popularity trends on the edX platform.

**6. Data Cleaning in Power Query**

**Overview:**

The data cleaning step was performed using Power Query in Power BI, where the raw data from the CSV file was transformed into a clean and structured format suitable for analysis. This process involved various data cleaning tasks, such as correcting data types, removing unnecessary text, and standardizing language values. Effective data cleaning ensures that the dataset is accurate, consistent, and ready for meaningful visualizations and insights in Power BI.

**Objectives:**

* To clean and prepare the scraped data for analysis by addressing inconsistencies, converting data types, and ensuring uniformity across all fields.
* To enhance data quality and reliability, making it suitable for creating accurate visualizations in the Power BI dashboard.

**Data Cleaning Tasks Performed:**

1. **Course Name -** Splitting the Institute Name from Title:
   * In many course titles, the institution name was included alongside the course name (e.g., "HarvardX: Remote Work Revolution for Everyone").
   * Using Power Query, the course name was split to separate the institution from the course title, enhancing clarity and allowing for better categorization and analysis by institution.
2. **Ratings** - Removed the Word "Stars" and Changed to Numeric Data Type:
   * Course ratings were initially presented as text strings that included the word "stars" (e.g., "4.5 stars").
   * Power Query was used to remove the word "stars" and convert the rating to a numeric data type, facilitating numerical analysis and visualization of ratings.
3. **Number of Ratings -** Removed 'Ratings' Text and Changed to Numeric Data Type:
   * The number of ratings field originally contained text entries such as "123 ratings".
   * The text "ratings" was removed, and the remaining numbers were converted to a numeric data type, ensuring consistency and enabling accurate aggregation and analysis.
4. **Number of Enrollment -** Changed Data Type to Whole Number:
   * Enrollment numbers were initially stored as text, which could cause errors in analysis and visualization.
   * Power Query was used to change the data type of enrollment numbers to whole numbers, allowing for correct summation and filtering in the Power BI dashboard.
5. **Language -** Converted Language Words to English:
   * Some language values were in non-English words or encoded characters (e.g., "EspaÃ±ol" instead of "Spanish", "FranÃ§ais" instead of "French").
   * Power Query was used to standardize language values by converting them into English, improving the dataset’s readability and consistency.
6. **Price -** Converted to Whole Number from Text:
   * The price data was initially stored as text, which included symbols or unnecessary characters.
   * The text was cleaned, and prices were converted into whole numbers, enabling accurate analysis of pricing trends and comparisons across different courses.

**Process Details:**

* **Splitting and Text Removal:**

Power Query’s text manipulation functions, such as split column by delimiter and replace values, were used to clean up course names, ratings, and the number of ratings fields.

* **Data Type Conversion:**

Data type conversion tools in Power Query were utilized to change text fields into appropriate numeric data types (e.g., whole numbers, decimals) to facilitate accurate calculations in Power BI.

* **Standardization:**

To ensure consistency across the dataset, language values were standardized using replace functions in Power Query, converting all entries to their English equivalents.

**Outcome:**

The data cleaning process in Power Query resulted in a well-structured and consistent dataset, free of text inconsistencies and ready for analysis. By addressing data type mismatches and standardizing key fields, the cleaned data enabled more accurate and meaningful visualizations in the Power BI dashboard. This preparation step was essential to ensuring that the subsequent analysis provided reliable insights, supporting data-driven decisions for course providers and learners on the edX platform.

**7. Storing Data in MySQL Database**

**Overview:**

After scraping data from edX, the next step was to store this data in a MySQL database. Storing the data in MySQL allows for organized, efficient management and provides a backend that integrates seamlessly with tools like Power BI for advanced data visualization. This phase involves connecting to the MySQL database using Python and importing the scraped data into a single table directly from a CSV file.

**Objectives:**

* To establish a connection between Python and MySQL for data storage.
* To create a single table in MySQL automatically from the DataFrame using pandas.DataFrame.to\_sql(), simplifying the data import process.
* To ensure that all data is accurately stored in a structured format, ready for querying and analysis.

**Tools and Libraries Used:**

1. **MySQL:** A relational database management system used for storing the scraped data, providing a structured environment for data management and retrieval.
2. **SQLAlchemy:** A Python SQL toolkit that provides a convenient interface for connecting to the MySQL database, enabling the seamless import of data from Python into MySQL.
3. **Pandas:** A Python data analysis library used to read the CSV file and interact with the database via SQLAlchemy, leveraging the to\_sql() function to create and populate the table automatically.

**Process Details:**

**1.Connecting to MySQL Database Using Python:**

The connection to MySQL was established using SQLAlchemy, which simplifies the process of setting up a database connection with the appropriate credentials and configuration.

**2.Automatically Creating and Populating the MySQL Table:**

The pandas.DataFrame.to\_sql() function was used to both create the table and insert the data in one step. This function automatically generates a table schema based on the DataFrame's structure, making it an efficient choice for straightforward data import tasks.

**Python Code for MySQL Connection and Data Import:**

import pandas as pd

from sqlalchemy import create\_engine

# Prompt the user to input the file path for the CSV containing the scraped data

file = input('Enter the file path: ')

# Read the CSV file using Pandas

df = pd.read\_csv(file, encoding='latin1')

# Define the MySQL connection string with the appropriate credentials

path = "mysql://root:Nivash%4003@localhost/Nivash\_c" # Replace with your MySQL credentials and database name

engine = create\_engine(path)

# Prompt the user to input the table name for storing the data

table\_name = input("Enter table name: ")

# Try to import the data into MySQL

try:

# Use Pandas to\_sql method to automatically create the table and insert data

df.to\_sql(table\_name, con=engine, index=False, if\_exists='replace') # 'if\_exists' parameter ensures the table is replaced if it already exists

print("Data uploaded successfully.")

except Exception as e:

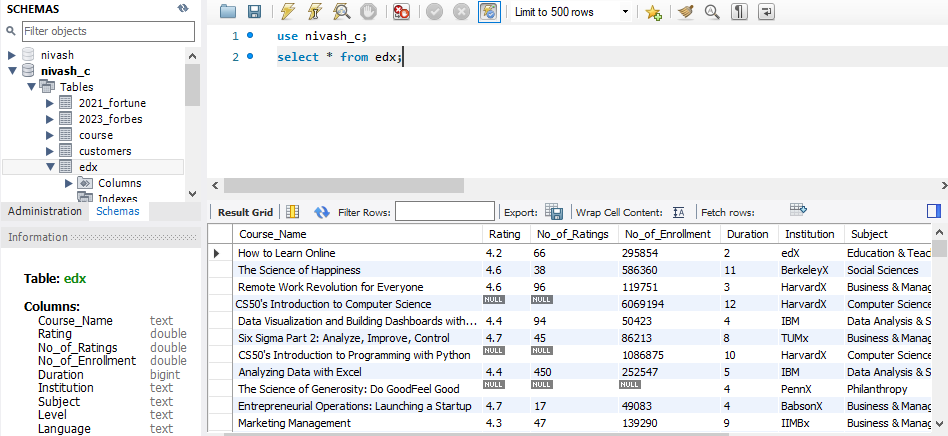
print(f"Error occurred: {e}")

* + **Connection Setup:** The script uses SQLAlchemy’s create\_engine() function to connect to the MySQL database using the provided credentials and connection string.
  + **Data Import with Table Creation:** The to\_sql() function from Pandas is used to import the DataFrame directly into MySQL. This function automatically creates the table based on the DataFrame's structure (column names and data types) if it doesn't exist, or replaces the table if specified.
  + **Error Handling:** The script includes a try-except block to catch and report any errors that occur during the data import process, ensuring that issues like connection errors or data type mismatches are managed gracefully.

**3. Validation and Testing:**

After importing the data, simple validation checks were performed:

* 1. **Row Count Validation:** Ensured that the number of rows in the MySQL table matches the CSV file.
  2. **Data Accuracy Checks:** Queried the database to verify that key fields such as course names, ratings, and prices were imported correctly without missing values.



**Outcome:**

The data was successfully stored in MySQL using a single table, simplifying the overall process by leveraging the automatic table creation capabilities of pandas.DataFrame.to\_sql(). This approach provided an efficient and effective way to manage the scraped data, making it readily available for querying, analysis, and integration with Power BI for visualization.

By using SQLAlchemy and Pandas, the transition from data collection to storage was streamlined, ensuring that the data was accurately organized and accessible for subsequent analysis and reporting phases of the project. This setup supports ongoing updates and scalability, allowing new data to be easily appended or replaced as required.

**8. Creating a Power BI Dashboard**

**Overview:**

In this step, an interactive Power BI dashboard was created to visualize the cleaned and structured data from the MySQL database. The dashboard provides an engaging interface for users to explore various aspects of edX course offerings, such as subject popularity, course pricing, and enrollment trends. The report consists of two pages, with the main dashboard featuring key visualizations and a secondary page providing details about edX and links to social media for further queries.

**Objectives:**

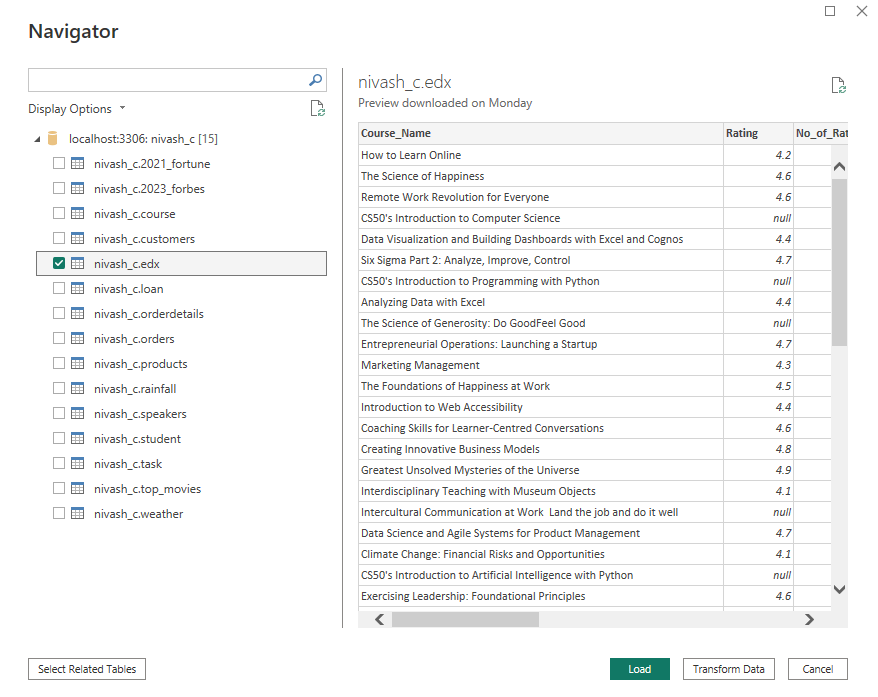
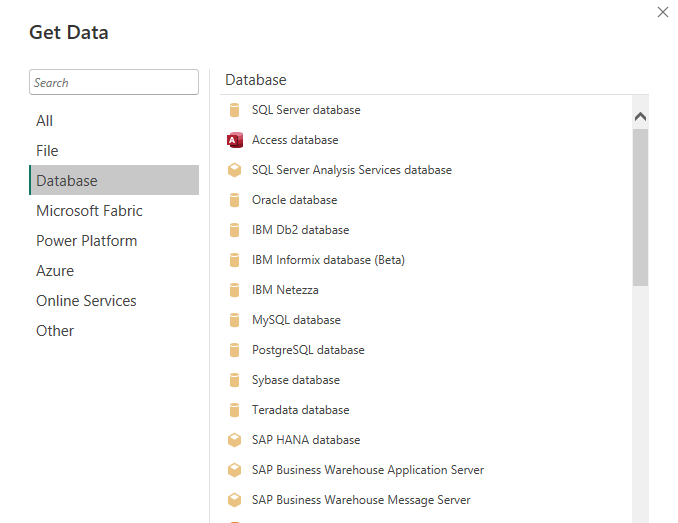
* To connect Power BI to the MySQL database containing the cleaned data and visualize key metrics that reflect trends and insights about edX courses.
* To design a dashboard with intuitive navigation and interactive elements that allow users to filter and explore the data by level and language.
* To create a second page that includes information about edX and the analyst’s contact details for further engagement.

**Tools and Features Used:**

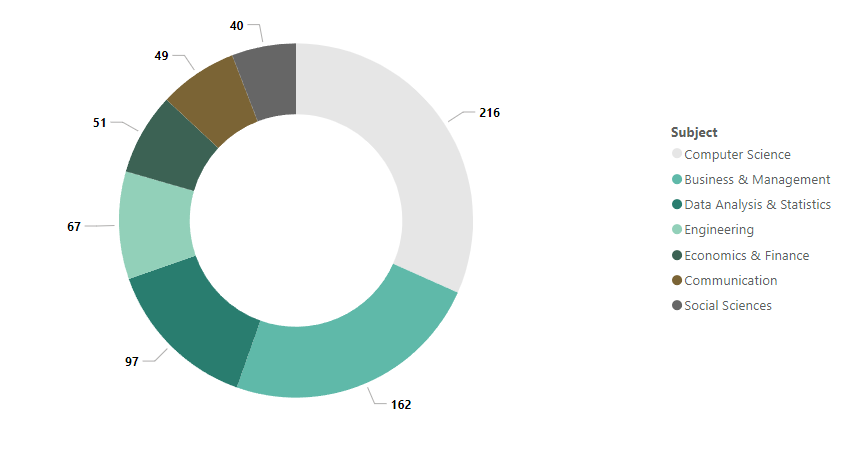
1. Power BI Desktop: Used for creating the visualizations and designing the layout of the dashboard.
2. MySQL Connector: Established the connection between Power BI and the MySQL database to import the data directly into Power BI.
3. Visual Elements in Power BI: Various chart types, KPIs, and tables were used to create a comprehensive view of the data.

**Process Details:**

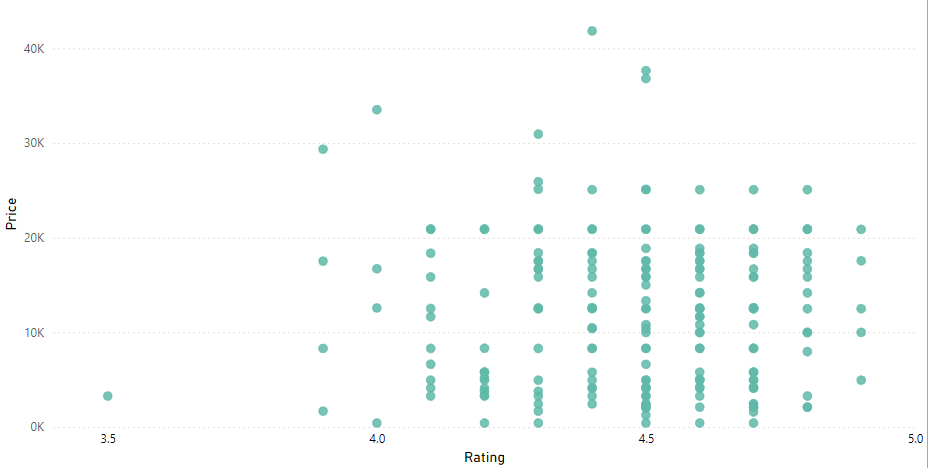
1. **Connecting Power BI to MySQL:**
   * Power BI was connected to the MySQL database using the MySQL connector, allowing for the seamless import of the cleaned course data.
   * The table with the course data was selected, ensuring that the visualizations in Power BI reflected the most up-to-date information



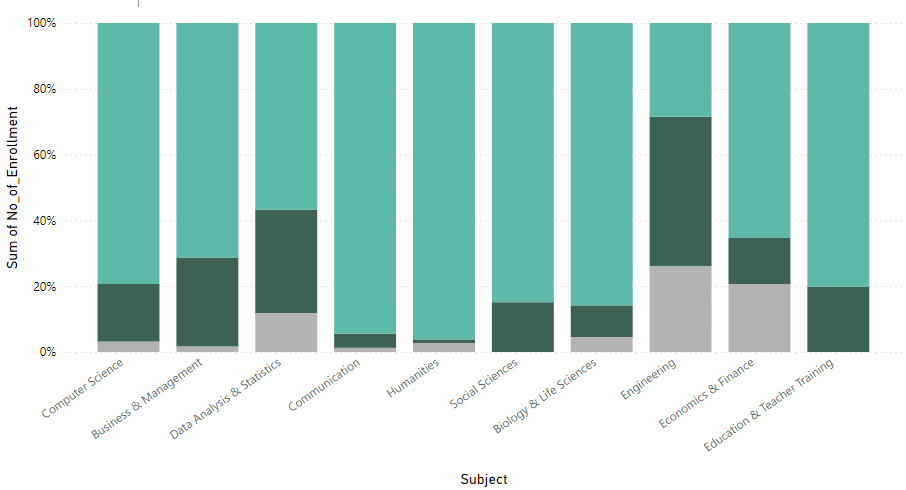
1. **Creating Visualizations:**
   * **Donut Chart -** Count of Course Name by Subject:
     + This chart visualizes the distribution of courses across different subjects, showing that Computer Science (31.67%) and Business and Management (23.75%) are the most prevalent subjects.



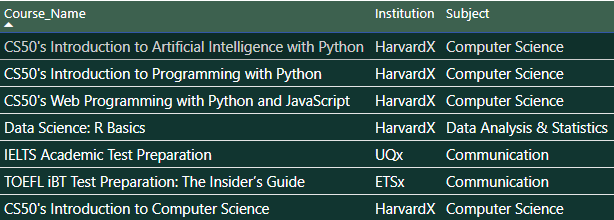
* + **Scatter Plot -** Rating vs. Price:
    - This plot illustrates the relationship between course ratings and price, revealing that most courses are clustered within the 4 to 5 rating range and priced at or below 20k.



* + **100% Stacked Column Chart -** Sum of Enrollments by Subject and Level:
    - This chart shows the enrollment distribution across subjects and course levels, indicating that introductory-level courses dominate enrollments in most subjects, followed by intermediate and advanced levels.



* + **Table -** Course Name, Institution, and Subject:
    - A table that lists the top 7 institutions by enrollment, providing a detailed view of which institutions are leading in attracting learners**.**

****

* + **KPIs (Key Performance Indicators):**
    - **KPI 1 -** Average Duration**:**

A gauge chart showing the average course duration is 5.88 weeks.

* + - **KPI 2 -** Average Price:

A gauge chart showing the average course price is 11.85k.

* + - **KPI 3 -** Average Rating:

A gauge chart showing the average rating of courses is 4.50.

* + - **KPI 4 -** Average Enrollment:

A gauge chart showing that the average number of enrollments is 83.07 lakh.

* + - **KPI 5 -** Total Courses**:**

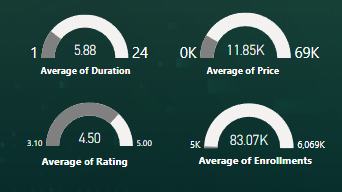
Displaying the total number of courses as 1000.

* + - **KPI 6 -** Total Institutions:

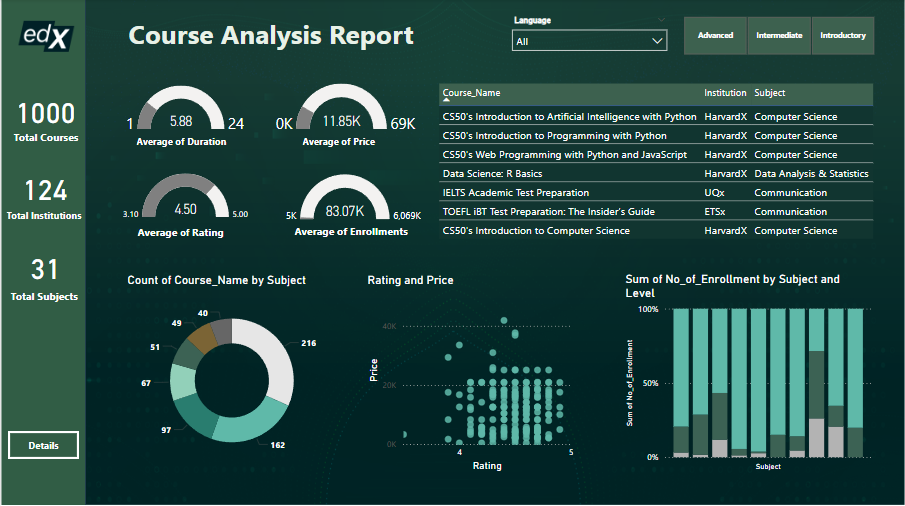
Showing the total number of institutions offering courses as 124.

* + - **KPI 7 -** Total Subjects:

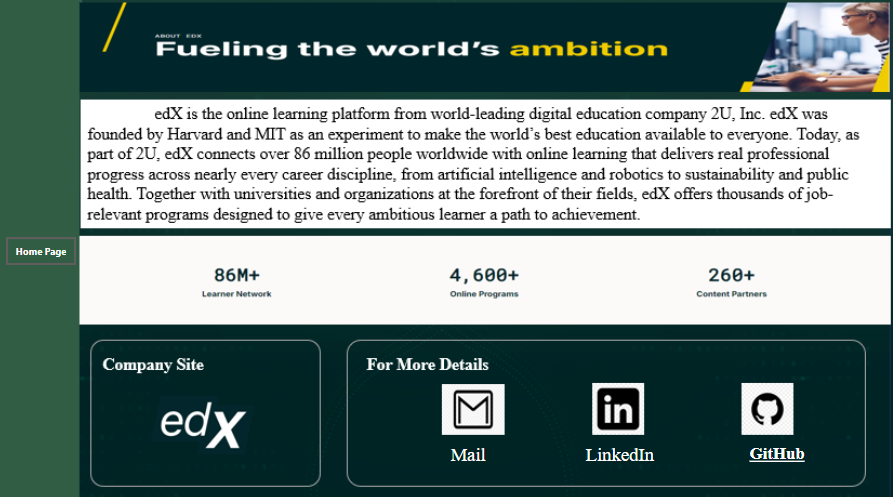
Indicating the total number of subjects available as 31.



1. **Dashboard Layout and Interactivity:**
   * **Page 1** - Main Dashboard**:**
     + The main dashboard includes all the key visualizations and KPIs, with filters for course level and language to allow users to refine the data view according to their interests.
     + A navigation button is provided to switch to the second page, enhancing the user experience with easy access to more detailed information.



* + **Page 2** - Information and Contact Details:
    - The second page contains details about edX and the company, along with links to social media profiles for user queries and further engagement.
    - A button on this page allows users to return to the main dashboard, ensuring seamless navigation between pages.



1. **Testing and Validation:**
   * The dashboard was tested to ensure that all visualizations correctly reflected the data from the MySQL database and that filters and buttons functioned as intended.
   * Performance testing ensured that the dashboard loads efficiently and provides a smooth user experience.

**Outcome:**

The Power BI dashboard provides a dynamic, interactive platform for exploring edX course data, highlighting key trends and metrics. The inclusion of KPIs and various charts allows stakeholders to quickly grasp the most important insights, while the filters and navigation buttons enhance the interactivity and usability of the report.

**9. Writing Insights and Final Report**

**Overview:**

This step involved analyzing the visualizations created in the Power BI dashboard to extract actionable insights and compiling them into a final report. The report summarizes the findings from the data analysis and provides recommendations for course providers and other stakeholders based on the observed trends.

**Insights from the Visualizations:**

1. **Subject Popularity:**
   * The donut chart revealed that Computer Science and Business and Management are the leading subjects on edX, comprising 31.67% and 23.75% of the total courses, respectively. This indicates a high demand for courses in these areas, reflecting current job market trends and learner interest in tech and business skills.
2. **Rating vs. Price Relationship:**
   * The scatter plot showed that the majority of courses are rated between 4 and 5, with most of these priced at or below 20k. This suggests that highly rated courses tend to be affordably priced, potentially driving higher enrollment due to their perceived value.
3. **Enrollment by Subject and Level:**
   * The 100% stacked column chart highlighted that introductory-level courses dominate enrollments across almost all subjects. This suggests a strong interest from beginners or those new to a field, emphasizing the importance of offering accessible entry-level courses.
4. **Top Institutions by Enrollment:**
   * The table visual listed the top 7 institutions based on enrollment numbers, showcasing the leading providers on edX. This insight can help new course providers benchmark their performance and understand the competitive landscape.
5. **Key Performance Indicators:**
   * **Average Course Duration:** At 5.88 weeks, the average course duration suggests that courses are generally designed to be completed in a manageable timeframe, appealing to busy learners.
   * **Average Price:** With an average price of 11.85k, the courses are positioned to be affordable for a wide range of learners.
   * **Average Rating:** The average course rating of 4.50 indicates overall satisfaction and high quality across the course offerings.
   * **Average Enrollment:** An average enrollment of 83.07 lakh highlights the scale of learner engagement on the platform.
   * **Total Metrics:** The dashboard also highlighted that there are 1000 courses, 124 institutions, and 31 subjects available, providing a diverse range of learning opportunities.

**Final Report and Recommendations:**

* The final report documents these insights, detailing the methodology and findings from the Power BI dashboard.
* **Recommendations:**
  + **Expand High-Demand Subjects:** Course providers should consider expanding offerings in high-demand subjects like Computer Science and Business to capture a larger market share.
  + **Focus on Entry-Level Courses:** Given the high enrollment in introductory courses, there is an opportunity to attract new learners by offering more beginner-friendly content.
  + **Leverage Ratings for Marketing:** Highlighting highly rated courses in marketing efforts can help attract more enrollments, as learners are drawn to courses with proven satisfaction.
  + **Price Optimization:** Providers should ensure that course pricing remains competitive, especially for highly rated courses, to maintain or increase enrollment numbers.

**10. Conclusion**

The insights derived from the Power BI dashboard offer valuable guidance for course providers on edX, helping them tailor their offerings to meet learner preferences and market demands. By leveraging these insights, providers can enhance the appeal of their courses, optimize pricing strategies, and ultimately drive greater engagement on the platform. The final report consolidates these findings and recommendations, providing a strategic roadmap for stakeholders looking to make data-driven decisions in the online education space.