**Introduction:**

* The web application is built on python and its library. This application takes in any dataset and perform EDA automatically.
* The dataset we have chosen to perform EDA and built a visualization is Expenditure data.
* From this dataset our goal is to perform EDA using the above python web application.
* We have also performed some EDA tasks manually on this data set to compare it with the result of our app.

**Introduction to the Dataset:**

* Shop Customer Data is a detailed analysis of an imaginative shop's ideal customers. It helps a business to better understand its customers. The owner of a shop gets information about Customers through membership cards.
* Dataset consists of 2000 records and 8 columns: Customer ID, Gender, Age, Annual Income, Spending Score (Score assigned by the shop, based on customer behavior and spending nature), Profession, Work Experience - in years, Family Size.

**Methodology:**

* Get a data from secondary research method (Kaggle, Twitter).
* Perform EDA manually by using pandas.
* Creating the web application with the help of Streamlit library.
* Creating a profiling with pandas profiling library.
* After creating an application, we have given dataset as an input for automatic EDA process through the web application.
* Finding the statistical analysis from both the method (automated as well as manual Visualization).

**Tools/Toolkit:**

* Python
* Pandas
* NumPy
* Streamlit library
* Pandas profiling library
* Seaborn
* Matplot.lib

**Agile framework:**

* Building an Agile framework for data cleaning can involve the following steps:
* Define the scope: Clearly define the scope of the data cleaning project, including the specific data sets that need to be cleaned, the cleaning tasks that need to be performed, and the objectives of the project.
* Form a cross-functional team: Assemble a cross-functional team with members from different departments such as data analysts, developers, and quality assurance, to work together to achieve the project objectives.
* Prioritize the backlog: Prioritize the backlog of data cleaning tasks based on their importance, dependencies, and risk
* Plan sprints: Plan sprints, which are short cycles of development, typically lasting 2 weeks, during which a specific set of tasks will be completed.
* Hold daily stand-up meetings: Hold daily stand-up meetings to discuss progress, identify and resolve any issues, and plan for the next day's tasks.
* Implement Continuous Integration and Continuous Deployment: Implement continuous integration and continuous deployment (CI/CD) practices to automate the data cleaning process, allowing for faster and more efficient data cleaning.
* Conduct regular reviews and retrospectives: Conduct regular reviews and retrospectives to evaluate the progress of the project, gather feedback.

**Description about App:**

This is a Python script that creates a web application using the Streamlit library and the pandas-profiling library to perform exploratory data analysis (EDA) on a dataset. The web application allows users to upload CSV or Excel files, merge them if needed, and generate a pandas profiling report for the resulting data.

* Here is a brief overview of the code:
* The first few lines import the necessary libraries: numpy, pandas, streamlit, pandas-profiling, and streamlit-pandas-profiling.
* The next section of code sets up the web application interface, including the app title and file upload widgets.
* The load\_csv and load\_excel functions are defined to load CSV and Excel files, respectively. These functions are decorated with the @st.cache decorator to improve performance by caching the loaded data.
* The code checks if a file has been uploaded and loads it into a DataFrame object. If a merge file is also uploaded, the code allows users to select columns to connect the two tables and the type of join to perform. If no merge file is uploaded, the code performs EDA on the uploaded file only.
* When the user clicks the "Run EDA" button, the code generates a pandas profiling report for the uploaded data, optionally merged with the other file. The report is displayed using the st\_profile\_report function from the streamlit-pandas-profiling library.
* To run this code, we can save it to a Python file with a .py extension and run it using a Python interpreter. Note that we will need to have the required libraries installed, which we can do using pip or conda. Once the code is running, we can access the web application by opening a web browser and navigating to the URL provided by the Streamlit library.

**Outputs:**

* Users can upload CSV or Excel files and merge them if necessary.
* Users can generate a pandas profiling report for the uploaded data, which includes statistics such as count, mean, and standard deviation for numerical columns, and frequency tables for categorial columns.
* Users can view interactive visualization of the data, such as histograms, scatter plots, and correlation matrices.
* Users can download the pandas profiling report as an HTML file.

**Outcomes:**

* The web app can help users quickly and easily understand the structure and content of their data, identify potential issues or inconsistencies, and gain insights that can inform further analysis or decision making.
* The web app can streamline the EDA process, saving users time and efforts compared to manually exploring the data using code or other tools.

**User Module:**

The user module will be responsible for handling user interactions with the web app. It will include the following functionalities:

* User login and authentication.
* Uploading a product CSV file.
* Viewing the uploaded product data.
* Selecting and filtering the data based on different attributes.
* Viewing the details of a specific product.
* Adding a new product to the existing dataset.
* Updating an existing product information.
* Deleting a product from the dataset.

**Testing module:**

The testing module will be responsible for testing the functionalities of the web app. It will include the following functionalities:

* Testing the user module functionalities.
* Testing the functionalities of merging datasets.
* Testing the pandas profiling report functionalities.
* Testing the security and authentication functionalities.
* Testing the responsiveness of the web app to different screen sizes.
* Testing the performance of the web app under different load conditions.

**Admin module:**

The admin module will be responsible for managing the web app and its content. It will include the following functionalities:

* Managing user accounts and permissions
* Monitoring the usage and performance of the web app
* Managing and updating the dataset used by the web app
* Configuring the settings and parameters of the web app
* Backing up and restoring the web app data and settings
* Updating the libraries and dependencies used by the web app

**Risk Management:**

* **Data Privacy:** If you are working with sensitive or personal data, you need to be careful about protecting it from unauthorized access or disclosure. Make sure that you comply with all applicable data protection laws and regulations, and only share the data with authorized individuals.
* **Data Quality:** The quality of the dataset can significantly affect the results of your analysis. You should carefully review the data to ensure that it is accurate, complete, and relevant to the project. You may need to perform data cleaning or preprocessing steps to prepare the data for analysis.
* **Algorithm Selection:** The choice of algorithms and techniques used for data analysis can have a significant impact on the accuracy and reliability of the results. Make sure that you select appropriate algorithms and techniques that are suitable for your specific data and objectives.
* **User Experience:** When developing a web app, it's important to keep the end-user in mind. You need to ensure that the app is intuitive and easy to use, and that the results are presented in a clear and meaningful way.
* **Technical Issues:** During the development and deployment of the app, you may encounter technical issues such as bugs, errors, or compatibility problems. You need to have a plan in place to address these issues quickly and effectively.
* **Resource Limitations:** The size of the dataset and the complexity of the analysis can require a significant amount of computational resources. Make sure that you have access to the necessary hardware and software resources to perform the analysis efficiently and effectively.

**Future Scope:**

* The current version of the web app is limited to basic EDA functionality, such as data cleaning, visualization, and profiling. Future development could include more advanced analysis techniques, such as machine learning modules, natural language process, or time series analysis.
* The web app could be extend to support more data formats, such as JSON, PDF files, Google docs, etc.
* The web app could include more customization options, such as the ability to select which columns to include in the profiling report or which visualization to generate.
* The web app could be integrated with other tools or platforms, such as Jupyter notebooks or cloud-based data storage solutions.

**Conclusion:**

* The web application saves time in EDA compared to manually coding and going through the EDA process.
* The web application is accurate and is a reliable in performing EDA comparing with the manual EDA on python.
* Web application can perform EDA on any dataset and does not depend on the type of business or client.
* The EDA report can be shared between teams once its generated like manual process.

**Literature Review:**

* Aindrila Ghosha et al. [1] have examined the different data exploration tool for exploratory analysis. They have described some of the data exploration tool.
* Author John T. Behrens [2] has described about the difference between classical data analysis and exploratory data analysis using different visualization method.
* Chokey Wangmo [3] has done an exploratory study on bank lending to SME sector in Bhutan.
* Matthew Ntow-Gyamfi et al. [4] has done an exploratory study on Credit risk and loan default among Ghanaian banks
* X.Francis Jency et al. [5] have done exploratory data analysis for loan prediction depending upon the nature of the client they have used machine learning techniques for predictive data analysis.K. Ulaga Priyal et al. [6] has done exploratory analysis on prediction of loan privilege for customers using random forest. They have used R programming for exploratory data analysis.
* Bogumil M. Konopka et al. [7] has done exploratory data analysis of a clinical study group. Development of a procedure for exploring multidimensional data.

**Thank You**