PRESIDENCY UNIVERSITY

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

School of Engineering

Department of Computer Science

Data Visualization

CSE - 367

A report on the project

Auto Plot WebApp

Batch of 2018 CSE

Under the guidance of-

Prof. Rama Krishna

Contents

Acknowledgement

We would like to extend our warm and heartfelt appreciation and thanks to our course instructor Prof. Rama Krishna who has encouraged us constantly to work on new aspects of Data Visualization and explore its related aspects. The insights that were received from our instructor were indeed very helpful and it reminded us how we can approach a real-life situation and provide a solution to it.

Secondly, I would like to thank all my classmates who suggested many improvements onto this project and who were very supportive in terms of constructively giving suggestions.

Lastly, I would like to thank all the members who were directly or indirectly involved in the completion of the project and for meeting the deadlines of the reviews in time.

Thank you.

Abstract

- The core functionality of the webapp is to provide a user a facility to visualize the data without having to code physically.
- The webapp has a simple interface and is capable of handling a dataset which can be a mix of simple to complex.
- There are additional features that enable a user to interact with plots like the time series plots and also visualize geo spatial data.

Members of the project

Sl. No.	Member Name	ID number
1	Sai Ram. K	20181CSE0621

Introduction

- In today's world there has been a tremendous amount of data that has been generated constantly. Organizations and companies are heavily relying on the data generated in order to get valuable insights in order to derive hidden facts.
- As the volume of the data generated increases, manually computing and deriving results is not feasible. This leads us to the use of modern tools that help us establish a connecting link in order to gain insights and present it in a meaningful perspective.
- This gave rise to the domain of data visualization that helps the users understand the distribution and orientation of the numerical data in pictorial forms for feasible visuals.
- This project aims to perform the different type of visuals by using various tools and packages in Python keeping the interactivity of the user to be the priority.
- The implementation would help individuals with non-tech background to easily navigate and generate visuals as they please in order to gain insights and also facilitate an entire report that can be saved locally for further analysis.

Objectives

The objectives that we intend to achieve are as follows:

- To facilitate the user with an easy and simple interface that can take an input of a CSV, TXT or XLS file.
- The user can select his desired function to perform from the sidebar of the webapp.
- The various options available are EDA, Plots, Interactive Mode and In-Depth report. Each of them has their own functionalities and can be used to generate plots.
- To implement additional features that include a button that can save the file or report generated to the local machine for viewing in offline mode or for presenting directly.

Motivation

The primary motivation for this webapp is to provide an environment wherein the working professionals can visualize their data in a quicker format without having to physically write any form of code in any type of tool. This webapp is also helpful for the non tech background individuals who can make use of it in order to generate reports or dashboards directly without having to employ an individual solely for the purpose of exploring the given data.

The motivation behind deploying this as a webapp is to make it available and handy for all individuals and keeping it open sourced in order to accept changes and modifications for the overall development of the application and its user interface.

Tools and Packages Used

In order to implement this webapp, we require a wide variety of tools that are responsible for the functionalities in this app. The version of python used in the making is version 3.9.0 and is deployed into Streamlit. We will now discuss the main modules in detail and also know it's working.

The modules used are as follows:

1 – **Streamlit: -**

Streamlit is an open-source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy, NumPy, pandas, Matplotlib etc. With Streamlit, no call-backs are needed since widgets are treated as variables. Data caching simplifies and speeds up computation pipelines. Streamlit watches for changes on updates of the linked Git repository and the application will be deployed automatically in the shared link.

2 – **Pandas: -**

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. It is suited for various kinds of data like tabular data, ordered and unordered time series data, arbitrary matrix data and any other form of observational / statistical data sets.

3 – **Matplotlib: -**

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

The pyplot API has a convenient MATLAB-style stateful interface. In fact, matplotlib was originally written as an open-source alternative for MATLAB. The OO API and its interface is more customizable and powerful than pyplot, but considered more difficult to use. As a result, the pyplot interface is more commonly used, and is referred to by default.

4 – **Seaborn: -**

Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas' data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

5 – **Wordcloud: -**

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analysing data from social network websites. Each word in this cloud has a variable font size and colour tone. Thus, this representation helps to determine words of prominence. A bigger font size of a word portrays its prominence more relative to other words in the cluster. Word Cloud can be built in varying shapes and sizes based on the creators' vision. The number of words plays an important role while creating a Word Cloud. A Word Cloud must always be semantically meaningful and must represent what it is meant for.

6 – **Pandas Profiling: -**

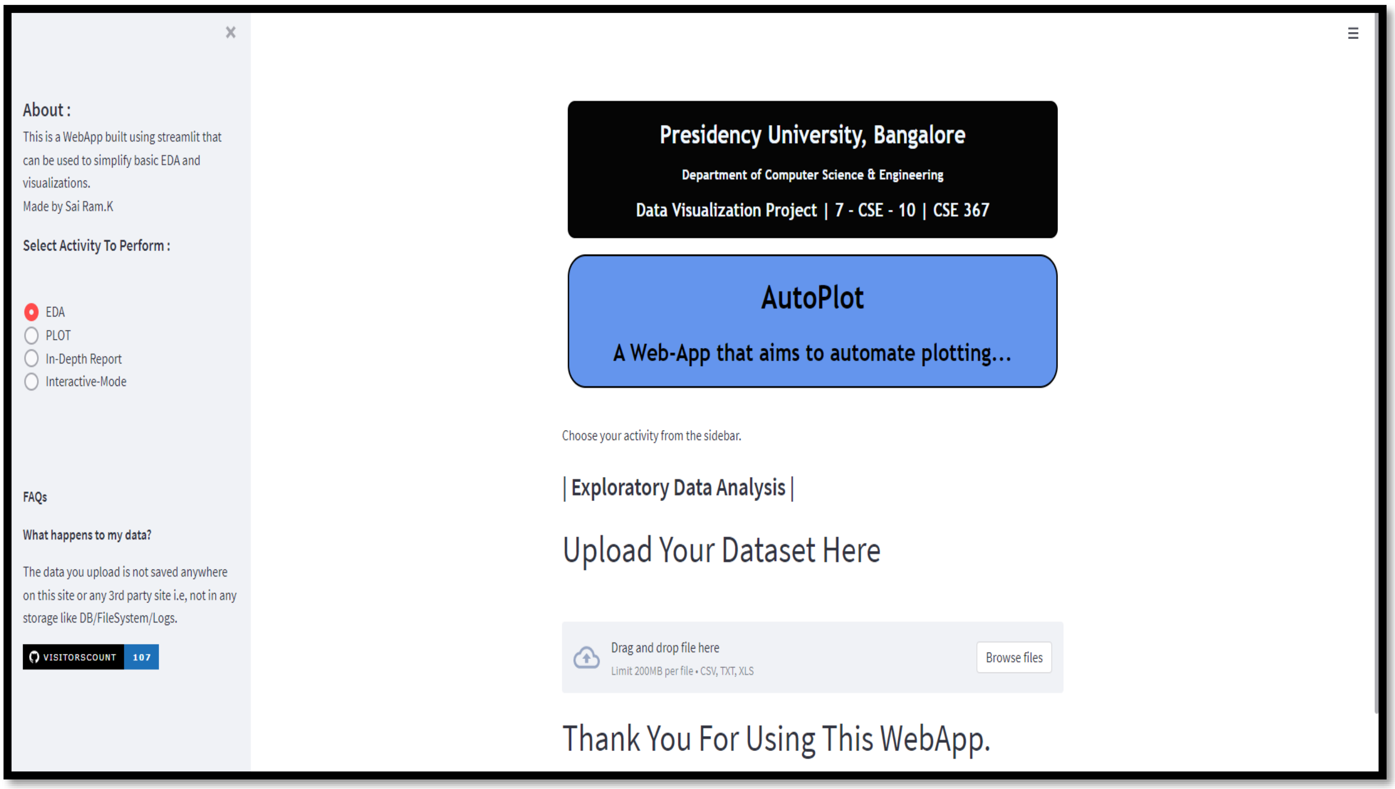
Pandas profiling is an open-source Python module with which we can quickly do an exploratory data analysis with just a few lines of code. it also generates interactive reports in web format that can be presented to any person, even if they don't know programming.

In short, what pandas profiling does is save us all the work of visualizing and understanding the distribution of each variable. It generates a report with all the information easily available.

Design of the Web-App

The design and construction of the webapp along with the interface would be the default layout of Streamlit. This allows the users to interact with the webapp in a simplistic manner so that there would be little to no conflicts. The various components that have been created under Streamlit would be discussed which gives an understanding of the design interface.

Home Screen of the web-app


About :

This is a WebApp built using streamlit that can be used to simplify basic EDA and visualizations.

Made by Sai Ram.K

Select Activity To Perform :

[✓ EDA] PLOT In-Depth Report Interactive-Mode

FAQs

What happens to my data?

The data you upload is not saved anywhere on this site or any 3rd party site i.e, not in any storage like DB/FileSystem/Logs.

VISITORSCOUNT 107

Presidency University, Bangalore

Department of Computer Science & Engineering

Data Visualization Project | 7 - CSE - 10 | CSE 367

AutoPlot

A Web-App that aims to automate plotting...

Choose your activity from the sidebar.

| Exploratory Data Analysis |

Upload Your Dataset Here

Drag and drop file here

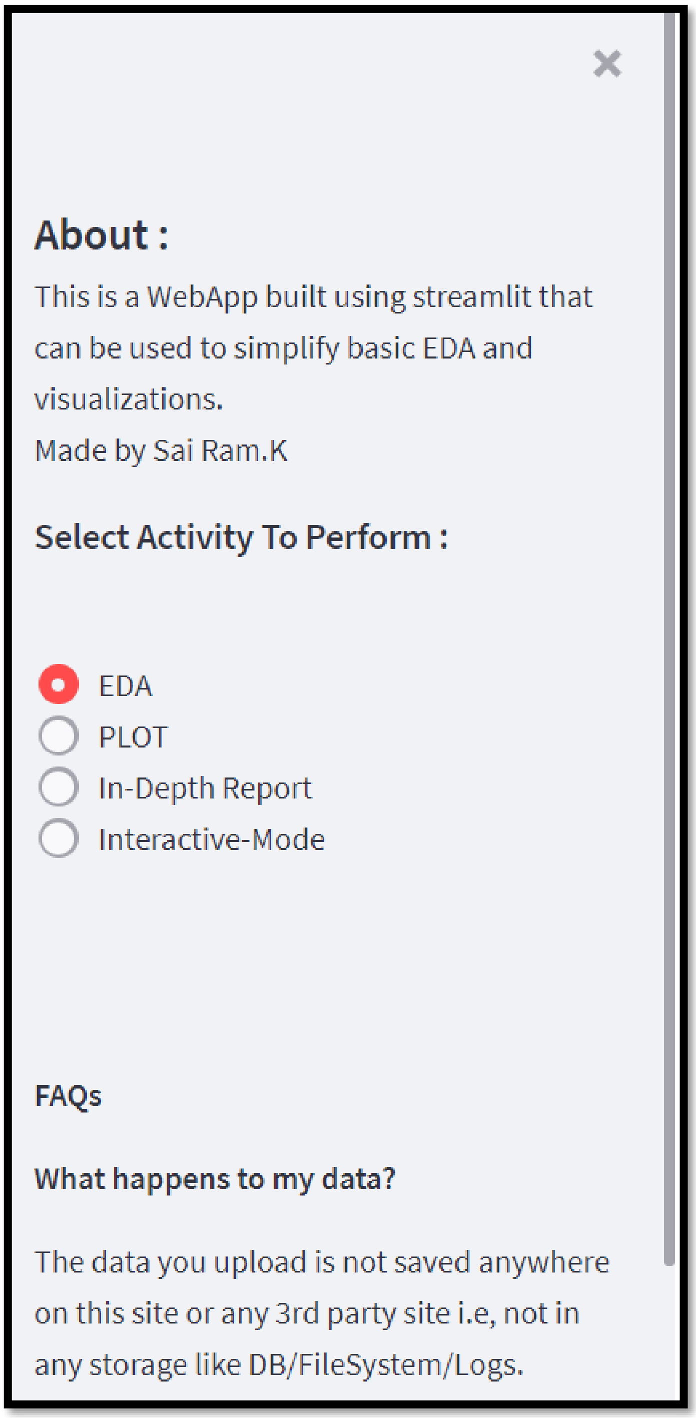
Limit 200MB per file • CSV, TXT, XLS

Browse files

Thank You For Using This WebApp.

The above image is the home screen of the webapp wherein the user can upload the dataset as well as explore all the features from the sidebar. Additional FAQs have also been included in case the user requires clarity of how to use the webapp.

Sidebar


About :

This is a WebApp built using streamlit that can be used to simplify basic EDA and visualizations.
Made by Sai Ram.K

Select Activity To Perform :

- [✓ EDA]
- PLOT
- In-Depth Report
- Interactive-Mode

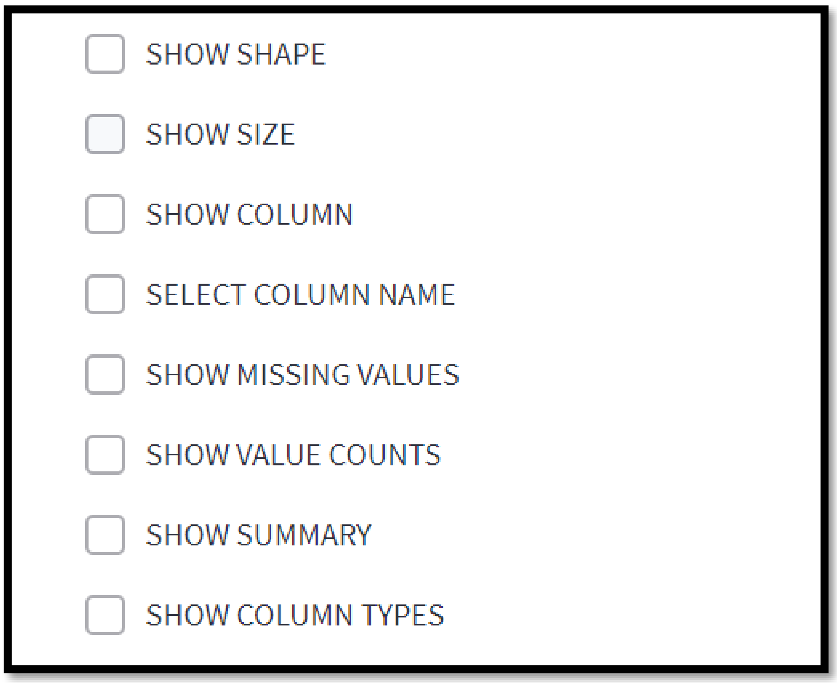
FAQs

What happens to my data?

The data you upload is not saved anywhere on this site or any 3rd party site i.e, not in any storage like DB/FileSystem/Logs.

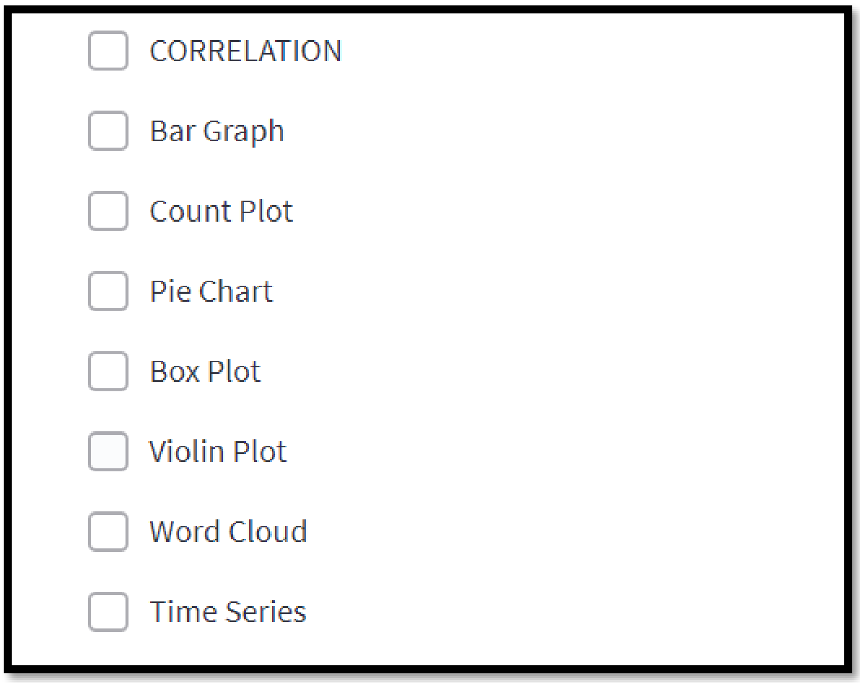
The sidebar of the webapp which allows the user to choose the desired mode of analysis to perform and to switch between modes.

EDA Mode



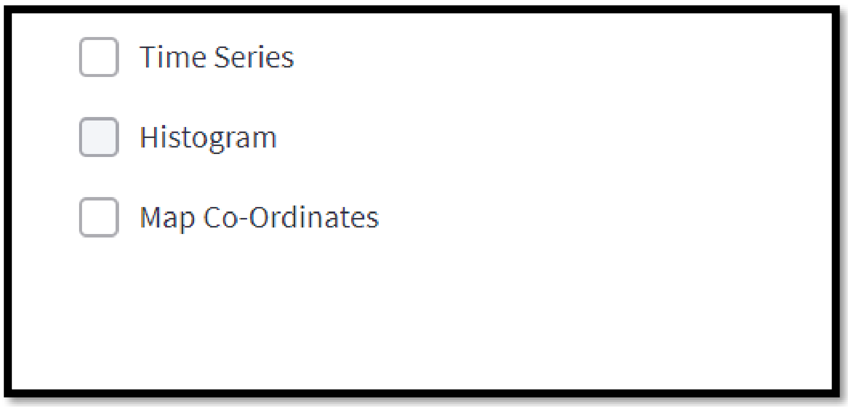
The functionalities of the Exploratory Data Analysis Mode are listed as above and user can choose the desired value to view.

Plots Mode



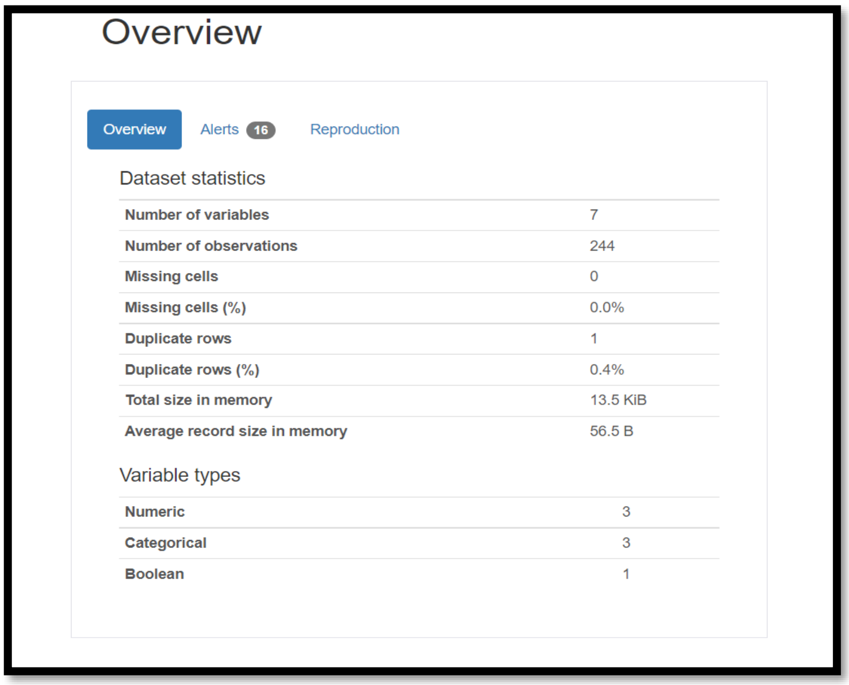
The functionalities of the Plots Mode are listed as above and user can choose the desired plot to view.

Interactive Mode



The functionalities of the Interactive Mode are listed as above and user can choose the desired plot to view.

In – Depth Report

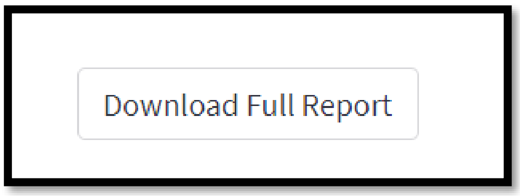


Dataset statistics

Number of variables	7
Number of observations	244
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	1
Duplicate rows (%)	0.4%
Total size in memory	13.5 KiB
Average record size in memory	56.5 B

Variable types

Numeric	3
Categorical	3
Boolean	1



The functionalities of the In-Depth Mode are listed as above and user can also choose to download a report.

Python Code Snippets

Importing Libraries & Boiler Plate:

```
import streamlit as st
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
import pandas_profiling
from wordcloud import WordCloud
import plotly.express as px
from streamlit_pandas_profiling import st_profile_report
import streamlit.components.v1 as components


plt.style.use('dark_background') ;
st.set_option('deprecation.showPyplotGlobalUse', False)
components.html(
    """
    <div class="intro" style="border: 2px solid black; border-radius: 25px; background-
color: cornflowerblue;font-family:Trebuchet MS,Garamond ;font-size: 18px;text-align:
center; ">
        <h1 style="margin: 0.25;"><strong>AutoPlot</strong></h1>
        <h2 style="margin: 0.25;">A Web-App that automates plots.</h2>
    </div>
    <br>
    <div class="college" style="border: 2px solid black; background-color: rgb(5, 5,
5);color: aliceblue;text-align: center;border-radius: 10px;font-family:Trebuchet
MS,Garamond;font-size: 14px">
        <h1><span>Presidency University, Bangalore</span></h1>
        <h3>Department of Computer Science & Engineering</h3>
        <h2>Data Visualization Project | 7 - CSE - 10 | CSE 367 </h2>
    </div>


    """,
    height=350,
)
```

EDA Mode:

```
    if choice == "EDA":
        st.subheader("|  Exploratory Data Analysis  |")
        st.write(""" <strong><p style="font-size: 42px">Upload Your Dataset 
Here</p></strong> """,unsafe_allow_html=True)
        dataset = st.file_uploader("" ,type = ["csv","txt","xls"])

        if dataset is not None:
            df = pd.read_csv(dataset , delimiter = ",")
            st.dataframe(df)
            if st.checkbox("SHOW SHAPE"):
                st.write(df.shape)
            if st.checkbox("SHOW SIZE"):
                st.write(df.size)
            if st.checkbox("SHOW COLUMN "):
                st.write(df.columns)
            if st.checkbox("SELECT COLUMN NAME"):
                select_columns = st.multiselect("Select Column" , df.columns)
                new_df = df[select_columns]
                st.dataframe(new_df)
            if st.checkbox("SHOW MISSING VALUES"):
                st.write(df.isna().sum())
            if st.checkbox("SHOW VALUE COUNTS"):
                column = st.selectbox("Select Columns" , df.columns)
                st.write(df[column].value_counts())
            if st.checkbox("SHOW SUMMARY"):
                st.write(df.describe())
            if st.checkbox("SHOW COLUMN TYPES"):
                column = st.selectbox("Select Columns" , df.columns,key="1")
                st.write(df[column].dtype)
```

Plots Mode:

```
    elif choice == "PLOT":
        st.subheader("|  Data Visualization  |")
        st.write(""" <strong><p style="font-size: 42px">Upload Your Dataset 
Here</p></strong> """,unsafe_allow_html=True)
        dataset = st.file_uploader("" ,type = ["csv","txt","xls"])

        if dataset is not None:
            df = pd.read_csv(dataset , delimiter = ",")
            st.dataframe(df)

            if st.checkbox("CORRELATION"):
                try:
                    st.write(sb.heatmap(df.corr() , annot = True,cmap="Blues"))
                    st.pyplot()
                except (ValueError,TypeError):
                    st.error("This Column does not correspond to a valid input data. 
Please Select a valid Column.")
            if st.checkbox("Bar Graph"):
                try:
                    x_axis = st.selectbox("Select x axis:" , df.columns)
                    x_axis = df[x_axis]
                    y_axis = st.selectbox("Select y axis:" , df.columns)
                    y_axis = df[y_axis]
                    st.write(sb.barplot(x_axis , 
y_axis,palette=['cyan','deeppink','cornflowerblue','coral']))
                    st.pyplot()
                    plt.xticks(rotation = 90)
                    plt.legend()
                    plt.grid()
                except (ValueError,TypeError):
                    st.error("This Column does not correspond to a valid input data. 
Please Select a valid Column.")

            if st.checkbox("Count Plot"):
                try:
                    c = st.selectbox("Select  axis:" , df.columns)
                    c_main = df[c]
                    st.write(sb.countplot(c_main,palette=['cyan','deeppink','cornflowerblu
e','coral','violet','crimson','yellow','lightcoral']))
                    st.pyplot()
                    plt.grid()
                    plt.xticks(rotation = 90)
                    plt.legend()
                except (ValueError,TypeError):
                    st.error("This Column does not correspond to a valid input data. 
Please Select a valid Column.")


            if st.checkbox("Pie Chart"):
                try:
                    col = st.selectbox("Select 1 column" , df.columns)
                    pie = df[col].value_counts().plot.pie(autopct = 
"%1.1f%%",colors=['cyan','deeppink','cornflowerblue','coral'])
                    st.write(pie)
                    st.pyplot()
                except (ValueError,TypeError):
                    st.error("This Column does not correspond to a valid input data. 
Please Select a valid Column.")
            if st.checkbox("Box Plot"):
                try:
                    col1 = st.selectbox("Select X column" , df.columns)
                    col2 = st.selectbox("Select Y column", df.columns)
                    box=sb.boxplot(x = col1, y = col2, data = 
df,notch=True,boxprops=dict(facecolor='r', color='cyan'),
capprops=dict(color='yellow'),flierprops=dict(color='g', 
markeredgecolor='r'),medianprops=dict(color='black'))
```

```
                    st.write(box)
                    st.pyplot()
                except (ValueError,TypeError):
                     st.error("These Columns do not correspond to a valid input data.
Please Select a valid Column.")
```

In-Depth Report:

```
    elif choice =="In-Depth Report":
            st.subheader("|  In - Depth Report  |")
            st.write(""" <strong><p style="font-size: 42px">Upload Your Dataset
Here</p></strong> """,unsafe_allow_html=True)
            dataset = st.file_uploader("" ,type = ["csv","txt","xls"])

            if dataset is not None:
                df = pd.read_csv(dataset , delimiter = ",")
                st.dataframe(df)
                pr = df.profile_report()
                st_profile_report(pr)
                export=pr.to_html()
                st.download_button(label="Download Full Report", data=export,
file_name='Report.html')
```

Interactive Mode:

```
    elif choice =="Interactive-Mode":
            st.subheader("|  Interactive Visualization  |")
            st.write(""" <strong><p style="font-size: 42px">Upload Your Dataset
Here</p></strong> """,unsafe_allow_html=True)
            dataset = st.file_uploader("" ,type = ["csv","txt","xls"])

            if dataset is not None:
               df = pd.read_csv(dataset , delimiter = ",")
               st.dataframe(df)
               if st.checkbox("Time Series"):
                    try:
                        col1 = st.selectbox("Select 1 column" , df.columns, key="2")
                        fig1 = px.line(df, x=df.index, y = df[col1])
                        fig1.update_layout(template="plotly_dark")
                        st.plotly_chart(fig1)
                    except (ValueError,TypeError):
                        st.error("The selected column is not in a time series format.
Please Select a valid Column.")

            if st.checkbox("Histogram"):
                    try:
                        column= st.selectbox("Select 1 column " , df.columns, key="6")
            his=px.histogram(df, x=column,title=column,
height=400,color_discrete_sequence=['#03DAC5'])
            his.update_layout(margin=dict(t=100, b=0, l=70, r=40),hovermode="x
unified",xaxis_tickangle=360,xaxis_title=' ', yaxis_title=" ",plot_bgcolor='#2d3035',
paper_bgcolor='#2d3035',\
            title_font=dict(size=40, color='#a5a7ab', family="Muli, sans-
serif"),font=dict(color='#FFFFFF',size=25),legend=dict(orientation="h", yanchor="bottom",
y=1.02, xanchor="right", x=1))
            st.write(his)
          except (ValueError,TypeError):
            st.error("The selected column is not in a valid format. Please
Select a valid Column.")
```

Conclusion & Real-World Application

A webapp that allows a user to upload the file and generate relevant EDA and plots facilitates a platform which serves as a tool to perform tasks efficiently. The files uploaded are not stored and hence it provides an additional layer of safety and trust to enable the users to explore their data. A situation wherein a technical case arises that does not require visualization as a primary focus and needs time to be coded manually can be replaced by the usage of this app as it is interactive and can be used in different use cases.

The webapp has also been deployed to the cloud in order to make it open-sourced and provide access to everyone. The webapp is currently in a beta stage and additional improvements would definitely be rolled out into the production in due course of time.

Link for the Web-App: - share.autoplot.py

Bibliography

Documentations:

- Streamlit - streamlit.io
- Pandas – pandas.pydata.org
- Seaborn - seaborn.pydata.org
- Stackoverflow – stackoverflow.com

Referred Web-Sites:

- Analytics India - https://analyticsindiamag.com/a-beginners-guide-to-streamlit
- Kaggle - https://www.kaggle.com/discussion
- Analytics Vidya - https://discuss.analyticsvidhya.com/
- YouTube (Tutorials) - https://www.youtube.com/

Blogs and Articles:

- Towards Data Science - https://towardsdatascience.com/streamlit-101
- Geeks for Geeks - https://www.geeksforgeeks.org/a-beginners-guide-to-streamlit/
- Medium - https://medium.com/streamlit/