ITD105 – Big Data Analytics
Lab Exercises #1
Exploratory Data Analysis (EDA) of Student Exam Performance

Instructions:

1. Install Required Libraries: Ensure you have the following libraries installed:

```
pip install streamlit pandas matplotlib seaborn plotly
```

2. Download the dataset (student-mat.csv)
Link - https://drive.google.com/drive/folders/1Bz9q37BB20PJSWsdGH__cshZGfPKSpHd?usp=sharing

3. Create a new Python file (e.g., student_performance.py) and set up the basic structure of your Streamlit app

4. Activity Tasks
   a. Load the dataset into the Streamlit app.
   b. Display the first few rows of the dataset.
   c. Show dataset information (e.g., data types, missing values).
   d. Generate summary statistics for the dataset.
   e. Create a heatmap to visualize correlations between features.
   f. Display a boxplot for exploratory visualization of numeric features.
   g. Use Plotly to create an interactive scatter plot of student performance.

5. Questions
   a. Which features have the highest correlation with the final exam scores (G1, G2, G3)?
      - The G2 has the highest correlation with G3 with a correlation 0.90
      - The G1 has the highest correlation with G2 with a correlation of 0.85,
      - The G3 has the highest correlation with G1 with a correlation of 0.80.
   b. How does study time correlate with exam performance?
      - Study time and G3 (final exam score) have a weakly positive correlation of 0.10, while the study time and G1 has the highest correlation with 0.16 and for the study time and G2 with a correlation with 0.14. This implies that higher exam performance is correlated with longer study sessions.
   c. What insights can you draw from the boxplot?
      - The range of ages is 16 to 18. The mother's education is marginally less than the father's. Study time and travel time are almost equal. Family bonds are strong. Plus, there's more leisure time. There's a sense of balance to going out with friends. There is a small increase in alcohol consumption on the weekends compared to the workdays. There are minimal absences and good health. While G2 performs little better, G1 and G3 are fairly close.
   d. How does gender impact the final exam score?
      - The gender barely affects the final exam.

Grade Booster:

- Add more interactive widgets in Streamlit to filter the dataset (e.g., by gender or parental education).

- Use other visualization techniques such as (1) bar charts and (2) pair plots to analyze other features.

- Make your Streamlit app look more like a dashboard, you can organize the layout using columns, tabs, and other widgets for filtering and interacting with the data.

Submit the following:

1. Source code

```
import io
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Title of the app
st.title('Exploratory Data Analysis with Streamlit')

# File uploader
uploaded_file = st.file_uploader("Upload CSV file here", type="csv")

if uploaded_file is not None:
    # Load the data
    df = pd.read_csv(uploaded_file, delimiter=";")

    # Display the first few rows of the dataframe
    st.subheader('Data Preview')
    st.write(df.head())

    # Display summary statistics
    st.subheader('Summary Statistics')
    st.write(df.describe())

    # Create two columns for Data Info and Missing Values
    col1, col2 = st.columns(2)

    # Data Info in the first column
    with col1:
        st.subheader('Data Info')
        buffer = io.StringIO()
        df.info(buf=buffer)
        s = buffer.getvalue()
        filtered_info = "\n".join(s.split('\n')[1:])
        st.text(filtered_info)

    # Missing Values in the second column
    with col2:
        st.subheader('Missing Values')
        st.write(df.isnull().sum())
        df = df.fillna(df.select_dtypes(include=[float,
int]).mean())

    # Create two columns for Pie Chart and Heatmap
    col1, col2 = st.columns(2)

    # Pie Chart in the first column
    with col1:
        st.subheader('Pie Chart ')
        categorical_cols =
df.select_dtypes(include=['object']).columns
        if len(categorical_cols) > 0:
            selected_col = st.selectbox('Select a categorical column
for pie chart:', categorical_cols)

            if selected_col:
                category_counts = df[selected_col].value_counts()
                fig, ax = plt.subplots(figsize=(3,3))
                ax.pie(category_counts,
labels=category_counts.index, autopct='%1.1f%%', startangle=140,
textprops={'fontsize': 4},
                       wedgeprops=dict(width=0.5))
                st.pyplot(fig)
        else:
            st.write("No categorical columns found in the dataset.")

    # Heatmap in the second column
    with col2:
        st.subheader('Correlation Heatmap')
        corr = df.select_dtypes(include=[float, int]).corr()
        fig, ax = plt.subplots(figsize=(10, 8))
        sns.heatmap(corr, annot=True, fmt='.2f', cmap='coolwarm',
ax=ax,
                    annot_kws={"size": 8}, cbar_kws={'shrink': .8},
                    linewidths=0.5, linecolor='gray')

        # Rotate labels
        ax.set_xticklabels(ax.get_xticklabels(), rotation=45,
ha='right', fontsize=10)
        ax.set_yticklabels(ax.get_yticklabels(), rotation=0,
fontsize=10)

        st.pyplot(fig)
    # Create two columns for Scatter Plot and Bar Chart
    col1, col2 = st.columns(2)

    # Scatter Plot in the first column
    with col1:
        st.subheader('Scatter Plot')
        num_cols = df.select_dtypes(include=[np.number]).columns
        x_col = st.selectbox('Select X-axis column', num_cols)
        y_col = st.selectbox('Select Y-axis column', num_cols)

        fig, ax = plt.subplots(figsize=(8, 6))
        sns.scatterplot(x=df[x_col], y=df[y_col], ax=ax)
        ax.set_title(f'Scatter Plot of {x_col} vs {y_col}')
        st.pyplot(fig)

    # Bar Chart in the second column
    with col2:
        st.subheader('Bar Chart')
        if len(categorical_cols) > 0:
            selected_col = st.selectbox('Select a categorical column
for bar chart or "Show All":', ['Show All'] +
list(categorical_cols))

            if selected_col:
                if selected_col == 'Show All':
                    combined_counts = pd.DataFrame()
                    for col in categorical_cols:
                        counts =
df[col].value_counts().reset_index()
                        counts.columns = ['Category', 'Count']
                        counts['Source Column'] = col
                        combined_counts =
pd.concat([combined_counts, counts])

                    fig, ax = plt.subplots(figsize=(12, 8))
                    sns.barplot(x='Category', y='Count', hue='Source
Column', data=combined_counts, ax=ax)
                    ax.set_xlabel('Category')
                    ax.set_ylabel('Count')
                    ax.set_title('Combined Bar Chart of All
Categorical Columns')
                    st.pyplot(fig)
                else:
                    category_counts =
df[selected_col].value_counts()
                    fig, ax = plt.subplots(figsize=(10, 6))
                    category_counts.plot(kind='bar', ax=ax)
                    ax.set_xlabel('Category')
```

```python
                    ax.set_ylabel('Count')
                    ax.set_title(f'Bar Chart of {selected_col}')
                    st.pyplot(fig)
        else:
            st.write("No categorical columns found in the dataset.")

    # Create three columns for Histograms, Density Plots, and Box
and Whisker Plots
    col1, col2, col3 = st.columns(3)

    # Plot histograms in the first column
    with col1:
        st.subheader('Histograms')
        num_cols = df.select_dtypes(include=[np.number]).columns
        for col in num_cols:
            fig, ax = plt.subplots(figsize=(6, 4))
            df[col].hist(ax=ax, bins=20)
            ax.set_title(f'Histogram of {col}')
            st.pyplot(fig)

    # Plot density plots in the second column
    with col2:
        st.subheader('Density Plots')
        for col in num_cols:
            fig, ax = plt.subplots(figsize=(6, 4))
            sns.kdeplot(df[col], ax=ax, fill=True)
            ax.set_title(f'Density Plot of {col}')
            st.pyplot(fig)

    # Plot box and whisker plots in the third column
    with col3:
        st.subheader('Box and Whisker Plots')
        for col in num_cols:
            fig, ax = plt.subplots(figsize=(6, 4))
            sns.boxplot(x=df[col], ax=ax)
            ax.set_title(f'Box and Whisker Plot of {col}')
            st.pyplot(fig)
            #macala normailah itd105 it4d
```

2. Video Screen record of your output (Max of 2mins only)
   Link of the Video
   https://drive.google.com/file/d/1roGAjFtBzwCI5mjisrV1VajFoTxVokBS/view?usp=sharing