Sales dashboard report

import streamlit as st

import pandas as pd

import plotly.express as px

# Load data

df = pd.read\_csv('C:/Users/Dell/Desktop/Deployment/Sales\_Data.csv')

df['Date'] = pd.to\_datetime(df['Date'], infer\_datetime\_format=True, dayfirst=True, errors='coerce')

df['Month'] = df['Date'].dt.strftime('%b')

df['Amount'] = df['Amount'].str.replace(',', '').astype(float).astype(int)

df['Price'] = df['Price'].str.replace(',', '').astype(float).astype(int)

# Set page configuration

st.set\_page\_config(

    page\_title="Sales Dashboard Report",

    page\_icon=":bar\_chart:",

    layout="wide"

)

# Sidebar

sidebar\_menu = st.sidebar.selectbox("Navigation", ["Main", "Data", "Data Analysis", "Conclusion"])

# Main section

if sidebar\_menu == "Main":

    st.title(":bar\_chart: Sales Dashboard Report")

    st.markdown("##")

    st.write("""

    Here is the sales data of a particular company that sells electronic Gadgets in different cities of India.

    We have the records from the first week of January 2021 to the last week of December 2021. Our objectives are:

    (1) Amount of Sales done by each Sales Representative

    (2) Amount of Productwise sales in each month of the year 2021

    (3) Amount of sales happened in monthwise of the year 2021

    (4) Amount of Sales by category

    (5) Amount of Product sold on a particular day

    """)

    st.divider()

# Data section

elif sidebar\_menu == "Data":

    st.markdown("## Data Introduction ")

    st.write(""" The data contain

    dates are in the First Column product

    name in the second column category in

    the third column name of sales

    representative in the fourth column name

    of city in the fifth column number of

    units sold in the sixth column unit price

    and amount in the third and second  last column & so on ...... There dataset contains 1559 rows and 9 columns""")

    # Display data table

    st.write("### Sales Data Table")

    st.dataframe(df)

# Data Analysis section

elif sidebar\_menu == "Data Analysis":

    st.markdown("## Data Analysis ")

    st.sidebar.header("Choose your filter:")

    # Sidebar filters

    months\_list = df['Month'].dropna().unique()

    months\_list.sort()  # Sorting the list

    selected\_month = st.sidebar.multiselect(

        "Pick the Month:",

        options=months\_list,

        default=months\_list

    )

    selected\_product = st.sidebar.multiselect(

        "Pick the Product:",

        options=df["Product"].unique(),

        default=df["Product"].unique()

    )

    selected\_category = st.sidebar.multiselect(

        "Pick the category:",

        options=df["Category"].unique(),

        default=df["Category"].unique()

    )

    selected\_sales\_rep = st.sidebar.multiselect(

        "Pick the Sales\_Rep:",

        options=df["Sales Rep"].unique(),

        default=df["Sales Rep"].unique()

    )

    # Filter data based on user selection

    filtered\_data = df[df['Month'].isin(selected\_month) & df['Product'].isin(selected\_product) &

                    df['Category'].isin(selected\_category) & df['Sales Rep'].isin(selected\_sales\_rep)]

    # Modify the 'Date' column to remove the time portion

    filtered\_data['Date'] = filtered\_data['Date'].dt.strftime('%Y-%m-%d')

    # Organize the layout using rows and columns

    col1, col2 = st.columns([1, 1])

    with col1:

        # Sales by salesRep (bar chart)

        sales\_Rep = filtered\_data.groupby(['Sales Rep'])['Amount'].sum().reset\_index()

        sales\_by\_salesRep = px.bar(sales\_Rep, x="Amount", y="Sales Rep", orientation="h", title="Sales by SalesRep")

        st.plotly\_chart(sales\_by\_salesRep)

        # Monthwise sales (bar chart)

        Monthly\_sales = filtered\_data.groupby(['Month'])['Amount'].sum().reset\_index()

        Monthwise\_sales = px.bar(Monthly\_sales, x='Amount', y='Month', orientation="h", title='Monthwise Sales')

        st.plotly\_chart(Monthwise\_sales)

    with col2:

        # Productwise sales (bar chart)

        sales\_by\_productwise = filtered\_data.groupby(by=["Product"])["Amount"].sum().reset\_index()

        productwise\_sale = px.bar(sales\_by\_productwise, x="Amount", y="Product", orientation="h", title="Productwise Sales")

        st.plotly\_chart(productwise\_sale)

        # Sales by category (pie chart)

        category\_data = filtered\_data.groupby(['Category'])['Amount'].sum().reset\_index()

        by\_category = px.pie(category\_data, values='Amount', names='Category', title='Category by Sales')

        st.plotly\_chart(by\_category)

    # 3D Scatter plot

    scatter\_3d = px.scatter\_3d(

        filtered\_data,

        x='Date',

        y='Product',

        z='Amount',

        color='Amount',

        size='Amount',

        title='3D Scatter Plot - Date vs Product vs Amount'

    )

    st.plotly\_chart(scatter\_3d)

# Conclusion section

elif sidebar\_menu == "Conclusion":

    st.markdown("## Conclusion ")

    st.write(""" from the analysis, we conclude

 that the bar chart gives the Amount of Sales done by each Sales Representative,

    Amount of Productwise sales in each month of the year 2021, Amount of sales happened in monthwise of the year 2021

    and pie chart gives Amount of Sales by category and 3D scatter plot gives Amount of Product sold on a particular day.

    so our objective is fulfilled.""")