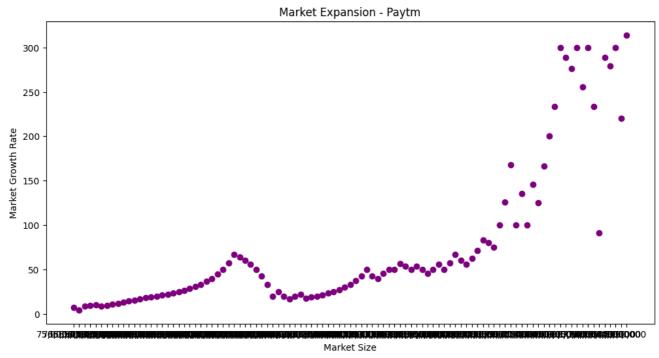
```
# Importing modules
import pandas as pd
import matplotlib.pyplot as plt
from ipywidgets import interact, widgets
import numpy as np
import seaborn as sns
# Loading dataset
data = pd.read_csv("/content/drive/MyDrive/Project Dataset/StartupsDataset.csv")
data["Revenue"] = pd.to_numeric(data["Revenue"], errors='coerce')
# Operational Efficiency (Bar chart)
def operational efficiency(startup name):
    productivity = data["Employee Productivity"]
    plt.bar(productivity.index, productivity.values, color="green")
    plt.title(f"Employee Productivity Over Time - {startup_name}")
    plt.xlabel("Time")
   plt.ylabel("Productivity")
   plt.ylim([productivity.min() * 0.9, productivity.max() * 1.1])
   plt.grid(True)
   plt.show()
    plt.close()
# Market Expansion (Scatter Plot)
def market_expansion(startup_name):
    market data = data[['Market Size', 'Market Growth Rate']].dropna()
    plt.figure(figsize=(12, 6)) # Increase figure size
    plt.scatter(market_data["Market Size"], market_data["Market Growth Rate"], color="purple")
   plt.title(f"Market Expansion - {startup_name}")
   plt.xlabel("Market Size")
   plt.ylabel("Market Growth Rate")
   plt.xticks(np.arange(0, market_data["Market Size"].max() + 500, 500), rotation=75)
    plt.grid(True)
   plt.tight layout()
    plt.show()
    plt.close()
# Customer Satisfaction (Heatmap)
def customer_satisfaction(startup_name):
   csat = data["Customer Satisfaction"].values.reshape(-1, 1)
    sns.heatmap(csat, annot=True, cmap="YlGnBu", cbar=True, linewidths=0.5)
   plt.title(f"Customer Satisfaction Heatmap - {startup_name}")
   plt.xlabel("Customer Satisfaction")
   plt.ylabel("Time")
   plt.show()
   plt.close()
# UI options
chart_options = ["Operational Efficiency", "Market Expansion", "Customer Satisfaction"]
startup_name_input = widgets.Text(description="Startup Name:")
chart_selector = widgets.SelectMultiple(options=chart_options, description="Charts:")
# Display selected charts based on user input
def display_selected_charts(startup_name, selected_charts):
    for chart in selected charts:
        if chart == "Operational Efficiency":
           operational_efficiency(startup_name)
        elif chart == "Market Expansion":
            market_expansion(startup_name)
        elif chart == "Customer Satisfaction":
            customer_satisfaction(startup_name)
# Create interactive UI
interact(display_selected_charts, startup_name=startup_name_input, selected_charts=chart_selector);
```

```
Startup Na...
          Charts:
                  Operational Efficiency
                  Market Expansion
                  Customer Satisfaction
     TypeError
                                                  Traceback (most recent call last)
     /usr/local/lib/python3.10/dist-packages/ipywidgets/widgets/interaction.py in update(self, *args)
                                  value = widget.get_interact_value()
         255
                              self.kwargs[widget._kwarg] = value
self.result = self.f(**self.kwargs)
         256
     --> 257
         258
                               show_inline_matplotlib_plots()
                              if self.auto_display and self.result is not None:
         259
                                          1 frames
     <ipython-input-6-a010b76d0314> in market_expansion(startup_name)
                 plt.ylabel("Market Growth Rate")
          31
          32
                 plt.xticks(np.arange(0, market_data["Market Size"].max() + 500, 500), rotation=75)
     ---> 33
          34
                 plt.grid(True)
     TypeError: can only concatenate str (not "int") to str
```





https://colab.research.google.com/drive/1wmmn2BopiraGd_qOtG7brt1q0KrSg9i4#printMode=true