**SOLVING A PROBLEM USING OPEN DATA**

**SOFTWARE DEVELOPMENT PROCESS**

**CODE**

import requests

import pandas as pd

import matplotlib.pyplot as plt

url\_unemp = "https://ec.europa.eu/eurostat/api/dissemination/statistics/1.0/data/une\_rt\_a?geo=LT&age=Y15-74&sex=T&unit=PC\_ACT"

resp\_unemp = requests.get(url\_unemp)

data\_unemp = resp\_unemp.json()

time\_index = data\_unemp["dimension"]["time"]["category"]["index"]

index\_to\_time = {v: k for k, v in time\_index.items()}

records = []

for idx\_str, val in data\_unemp["value"].items():

  idx = int(idx\_str)

  tc = index\_to\_time.get(idx)

  if tc is not None:

    records.append((tc, val))

df\_unemp = pd.DataFrame(records, columns=["year\_str", "unemployment"])

df\_unemp["year"] = pd.to\_datetime(df\_unemp["year\_str"], format="%Y").dt.year

df\_unemp = df\_unemp.sort\_values("year").reset\_index(drop=True)

url\_gdp = "https://ec.europa.eu/eurostat/api/dissemination/statistics/1.0/data/nama\_10\_gdp?geo=LT&na\_item=B1GQ&unit=CP\_MEUR"

resp\_gdp = requests.get(url\_gdp)

data\_gdp = resp\_gdp.json()

time\_index = data\_gdp["dimension"]["time"]["category"]["index"]

index\_to\_time = {v: k for k, v in time\_index.items()}

records\_gdp = []

for idx\_str, val in data\_gdp["value"].items():

  idx = int(idx\_str)

  tc = index\_to\_time.get(idx)

  if tc is not None:

    records\_gdp.append((tc, val))

df\_gdp = pd.DataFrame(records\_gdp, columns=["year\_str", "gdp\_meur"])

df\_gdp["year"] = pd.to\_datetime(df\_gdp["year\_str"], format="%Y").dt.year

df\_gdp = df\_gdp.sort\_values("year").reset\_index(drop=True)

df = pd.merge(df\_unemp[["year", "unemployment"]],

              df\_gdp[["year", "gdp\_meur"]],

              on="year", how="inner")

df["gdp\_growth\_pct"] = df["gdp\_meur"].pct\_change() \* 100

print(df.tail(10))

fig, ax1 = plt.subplots(figsize=(10,6))

ax1.set\_xlabel("Year")

ax1.set\_ylabel("Unemployment rate (%)", color="tab:red")

ax1.plot(df["year"], df["unemployment"], color="tab:red", marker="o", label="Unemployment Rate")

ax1.tick\_params(axis="y", labelcolor="tab:red")

ax2 = ax1.twinx()

ax2.set\_ylabel("GDP growth rate (%)", color="tab:blue")

ax2.plot(df["year"], df["gdp\_growth\_pct"], color="tab:blue", marker="s", linestyle="--", label="GDP Growth")

ax2.tick\_params(axis="y", labelcolor="tab:blue")

plt.title("Lithuania: Unemployment vs GDP Growth (Eurostat)")

fig.tight\_layout()

plt.grid(True, axis="x")

plt.show()

**OUTPUT**

A graph of the number of unemployment and gdp growth

AI-generated content may be incorrect.

**1. Problem Definition**

One of the classic questions in economics is how unemployment and GDP are connected. Usually, when economies grow fast, more jobs get created and unemployment goes down. On the flip side, during economic slowdowns, unemployment tends to rise.

In this project, I looked at Lithuania’s situation over the last couple of decades:  
*Does Lithuania’s GDP growth match up with changes in unemployment rates?*

This is relevant because Lithuania has gone through big shifts since joining the EU in 2004, including the 2008–09 financial crisis, the recovery years, and more recent shocks. Seeing these two indicators together can tell us how resilient the economy has been and what kind of challenges workers faced.

**2. Datasets Used**

I pulled both datasets from **Eurostat**, the official statistics portal of the European Union.

* **Unemployment Rate**
  + Title: *Unemployment rate by sex and age – annual data*
  + Code: une\_rt\_a
  + API endpoint:
  + https://ec.europa.eu/eurostat/api/dissemination/statistics/1.0/data/une\_rt\_a?geo=LT&age=Y15-74&sex=T&unit=PC\_ACT
  + Indicator: Annual unemployment rate (% of active population), Lithuania, ages 15–74.
* **GDP (current prices)**
  + Title: *GDP and main components (output, expenditure and income)*
  + Code: nama\_10\_gdp
  + API endpoint:
  + https://ec.europa.eu/eurostat/api/dissemination/statistics/1.0/data/nama\_10\_gdp?geo=LT&na\_item=B1GQ&unit=CP\_MEUR
  + Indicator: GDP in millions of euros (current prices), Lithuania.

Both datasets are free to use and published under Eurostat’s open data policy (CC-BY style).

**3. Data Cleaning and Processing**

The Eurostat API delivers data in JSON format with dimensions and indexes. To make it usable:

1. Extracted the time dimension and converted it into a year column.
2. Converted unemployment values and GDP values into numeric form.
3. Merged both datasets by year.
4. Calculated **GDP growth rate** as the year-to-year percentage change.

The main issue I faced was that Eurostat’s JSON format is a bit tricky, since it gives time indices rather than plain years. I solved this by mapping the indices back to year codes and converting them into datetime. Missing values weren’t a problem here, since Lithuania had complete data.

**4. Results and Visualizations**

Here’s the key chart I produced:

* **Red line (left axis):** Lithuania’s annual unemployment rate (%).
* **Blue dashed line (right axis):** Lithuania’s GDP growth rate (%).

**Observations:**

* During the 2008–2009 financial crisis, GDP growth collapsed (negative growth), and unemployment spiked above 15%.
* From 2010 to 2019, steady GDP growth lined up with falling unemployment, hitting record lows near 6%.
* The COVID-19 dip in 2020 briefly slowed GDP growth, but unemployment didn’t explode as much compared to 2009, probably due to stronger safety nets and EU recovery funding.
* In recent years (2021–2023), GDP growth bounced back, while unemployment stayed relatively stable, suggesting the labor market is more resilient now.

**5. Conclusions and Implications**

The analysis shows that Lithuania’s economy is strongly affected by global shocks (2009, 2020), but the connection between GDP growth and unemployment has weakened a bit over time.

Policy-wise, this might mean:

* Investments in labor market flexibility and EU support helped cushion unemployment during COVID-19.
* High GDP growth does not automatically mean ultra-low unemployment anymore — so policymakers should focus on structural issues (education, skills, migration) as much as raw GDP numbers.

Overall, the data confirms the “classic” link: recessions hurt jobs, but it also shows Lithuania has gotten better at protecting workers during downturns.

**6. Obstacles Faced**

* **Eurostat JSON structure**: Had to carefully extract years and values, since the API doesn’t give a simple CSV.
* **Units**: GDP is in millions of euros (current prices), so I focused on growth rates to make it meaningful.
* **Dual-axis plotting**: Needed to plot unemployment and GDP growth on different scales, otherwise the chart wouldn’t make sense.

**7. Licensing**

Eurostat data is open under the **Creative Commons Attribution (CC-BY 4.0)** license. That means it’s free to use as long as the source is credited.