Nowcasting Socio-Economic Trends: An LSTM-based Approach Leveraging Google Trends, GDELT, and Eurostat Data

Cristina Jiménez, Ayah Dahmani, Ricardo González Otal and Juan Miguel López Piñero

1. Overview
2. Introduction
3. Aim and Objectives
4. PyTrends API
   4.1. API Overview
   4.2. Setup and Configuration
   4.3. Retrieving Google Trends Data
   4.4. Data Cleaning and Preprocessing
   4.5. Challenges and Solutions
5. GDELT API
   5.1. API Overview
   5.2. Accessing Extracting and Transforming the GDELT Database
   5.3. Challenges and Solutions
6. Eurostat API
   6.1. API Overview
   6.2. Authenticating, Retrieving Socio-Economic, Data Cleaning and Handling Missing Values
   6.3. Challenges and Solutions
7. Data Integration
   7.1. Merging Data from Multiple Sources
   7.2. Handling Inconsistencies and Mismatches
   7.3. Feature Engineering and Selection

1.OVERVIEW

This project aims to develop nowcasting models using advanced machine learning techniques, such as random forests, extreme gradient boosting, stacked ensembles, and neural networks. These models will be employed to predict, in real-time, a set of socio-economic variables in relation to labor market integration, providing valuable insights for decision-making processes. Moreover, this project will serve as a tool for new graduates from the Data Science program 2023-2024 of The Bridge School, allowing them to apply and showcase their skills in machine learning and data analysis.

2.INTRODUCTION

In today's rapidly evolving world, the ability to predict and understand socio-economic trends in real-time has become increasingly valuable. This project, created by the company Rambee and assigned to a team of four graduates from The Bridge School, aims to develop a nowcasting model based on machine learning techniques to forecast a specific socio-economic variable using big data obtained from Google Trends, the Global Database on Events, Language, and Tone (GDELT), and Eurostat.

Nowcasting,("now" and "forecasting"), refers to the process of predicting the present or the very near future by analyzing current and historical data. This approach is particularly useful in situations where traditional forecasting methods may not be as effective due to the lack of timely data or the need for more immediate insights.

The main goal of this project is to develop and validate an LSTM neural network nowcasting model for sequential data analysis. The model will utilize three key data sources: Google Trends for search volume trends across countries, GDELT for sentiment indicators and topic popularity from news articles, and Eurostat for official socio-economic statistics from the European Union. By combining these diverse data sources, the model aims to provide insights into consumer behavior, sentiment trends, and socio-economic indicators for forecasting purposes.

3.AIMS AND OBJECTIVES

The primary tasks involved in this project are as follows:

1. Extract information on search volume queries from Google Trends for a predefined set of categories, with weekly frequency, across different countries.
2. Retrieve sentiment indicators and topic popularity rates from GDELT for a series of relevant socio-economic themes, in the form of "Article Tone" and "Topic Popularity Rate" data.
3. Obtain relevant socio-economic data from Eurostat for the variable of interest.
4. Construct and validate an LSTM-based nowcasting model for a specific socio-economic variable of interest, using the extracted data from Google Trends, GDELT, and Eurostat as predictors.

The final nowcasting model will enable a deeper understanding of current trends and facilitate proactive decision-making processes.

4. PYTRENDS API

5.1. API Overview
5.2. Accessing the GDELT Database
5.3. Extracting Sentiment Indicators and Topic Popularity
5.4. Data Transformation and Formatting
5.5. Challenges and Solutions

5. GDELT API

5.1. API Overview

The Global Database of Events, Language, and Tone (GDELT) is a massive open-source database that monitors, captures and codes events from news articles and classify them into a structured database; these attributes make GDELT a valuable resource for analysts, and data scientists interested in studying global events. In the following section, we will explore how the team managed to access the GDELT database, extract relevant data, transform and format it for analysis, and address common challenges encountered while working with this data source.

5.2. Accessing Extracting and Transforming the GDELT Database

The team objective was building an API that could extract and analyze data from GDELT , accessing and working with such a vast and complex dataset has been a challenge.

After familiarizing ourselves with the GDELT documentation and understanding the different types of data available, we decided to focus our efforts on extracting sentiment indicators (tone) and topic popularity data. These two aspects would provide valuable insights into public perception and media coverage of various events and topics.

We chose to build our API using the FastAPI framework, because it offered a modern and efficient approach to creating web applications and APIs. The first step was to set up the project structure and install the necessary dependencies, including the GDELT doc library, which would serve as our gateway to the GDELT database.

With the project setup complete, we began designing the API endpoints. We wanted to provide users with the flexibility to extract data at different time intervals – daily, monthly, and quarterly. This would cater to various use cases and allow for more granular or aggregated analysis as needed.

The first endpoint we built was */diary/extraction*, which allowed users to extract daily data for a specific keyword and country. We implemented filters to narrow down the search results based on the provided parameters. Users could also choose to download the extracted data as a CSV file for further analysis or storage.

Next, we tackled the */monthly/extraction and /quarterly/*extraction endpoints. These endpoints would aggregate the daily data into monthly and quarterly summaries, respectively. We used pandas' grouping and aggregation capabilities to achieve this, ensuring that the data was correctly grouped and summed based on the specified time intervals.

To streamline the data extraction process, we created a */project* endpoint that would automate the extraction and aggregation of data for a predefined list of countries. This endpoint would

generate CSV files for each country, containing the tone and popularity data at monthly and quarterly intervals.

As our API grew more complex, we recognized the need for data cleaning and preprocessing. We implemented a /clean endpoint that would take the extracted data, handle missing dates, remove duplicates, and ensure that the data was in a consistent format for further analysis.

Finally, we added a /mean endpoint that would calculate the mean values for a specified column across multiple CSV files. This would allow users to easily obtain the average tone or popularity for a given time period, providing a high-level overview of the data.

5.3. Challenges and Solutions

As we progressed, we encountered several challenges. One of the most significant hurdles was……..

Throughout the development process, we encountered numerous challenges and learned valuable lessons. We had to adapt our approach based on the limitations and quirks of the GDELT API, and we had to be creative in our solutions to overcome obstacles. Collaboration and effective communication within the team were crucial to our success.

In the end, we were proud to have built a robust and feature-rich API that could extract, aggregate, clean, and analyze data from the GDELT database. Our API would serve as a valuable tool for researchers, data scientists and analysts

6. Eurostat API

6.1. API Overview

Once again, we chose to build our API using the FastAPI framework, as it had proven to be a reliable and efficient choice for our previous project with the GDELT database. We set up the project structure and installed the necessary dependencies, including the pandas library for data manipulation and the requests library for making HTTP requests.

6.2. Authenticating, Retrieving Socio-Economic Indicators, Data Cleaning and Handling Missing Values

Upon receiving the response from the Eurostat API, we saved the compressed data to a local file and then decompressed it using the gzip module. This decompressed data was then read into a pandas DataFrame for further processing.

We quickly realized that the raw data from Eurostat required extensive cleaning and preprocessing. We had to remove unnecessary columns, handle missing values, and round numerical values to a specific number of decimal places. Additionally, we had to map the time periods from quarters to specific months, as our analysis required monthly data.

One of the challenges we faced was filtering the data based on specific criteria. We wanted to focus on the employment rates of foreign citizens who were actively employed. To achieve this, we applied multiple filters to the DataFrame, selecting only the relevant rows based on citizenship status and employment status.

Next, we needed to split the data by country and gender, as our analysis required separate datasets for each combination of country and gender. We created a dictionary of DataFrames, with each key representing a country and the corresponding value being a list of two DataFrames (one for males and one for females).

As we progressed, we encountered several issues, such as countries with missing or incomplete data. To address this, we implemented functions to identify and remove DataFrames with zero or missing values, ensuring that our analysis would be based on reliable and complete data.

Once the data was cleaned and preprocessed, we saved the resulting DataFrames to separate folders for raw, preprocessed, and processed data. This organization would make it easier to track the different stages of data processing and facilitate future analysis or updates.

To further enhance our analysis, we implemented functions to complete missing time series data and generate a combined series representing the European Union as a whole. This would allow

us to analyze trends and patterns not only at the country level but also at the broader regional level.

6.3. Challenges and Solutions

Throughout the development process, we encountered numerous challenges and learned valuable lessons. We had to adapt our approach based on the quirks and limitations of the Eurostat API, and we had to be creative in our solutions to overcome obstacles. Collaboration and effective communication within the team were crucial to our success.

In the end, we were proud to have built a robust and feature-rich API that could extract, clean, preprocess, and analyze data from the Eurostat database. Our API would serve as a valuable tool for researchers, analysts, and data scientists interested in studying employment trends and labor force dynamics in the European Union.