Modern Data Analytics – G0Z39a (GROUP 22)
Elia Giorgi, Elias Iblisdir, Kaya Dogan, Tim Balcaen

1. Setting the scene

1.1 Story

Funding allocation for research is a complex process influenced by various factors, making it an important subject of analysis. Some countries and research areas may receive a larger share of the budget, while others secure fewer grants. A thorough investigation of the CORDIS Horizon Europe dataset may uncover how future funding opportunities relate to research area, location, previous research output, interdisciplinarity, novelty of the research and practical applicability. Moreover, insights could be yielded into how funding is distributed and where disparities exist. This latter point is essential point to scrutinize, given that the whole purpose of the Horizon Europe program is to foster innovation while promoting "industrial competitiveness"[1].

1.2 Research questions

While the dataset allows for a plethora of avenues of inquiry, for the purpose of this assignment we will attempt to answer an overarching question: "How can a young researcher improve the odds of obtaining a grant?". Furthermore, we will also investigate the effects of covariates which are uncontrollable for the researchers (e.g. university affiliation, research area) to have a full picture of what is really impacting the outcome of interest, without leaving any potential predictors aside. To this end, we will explore the following secondary research questions:

1. Which variables are most strongly associated with the amount of money granted for a project?
2. How is the number of grants distributed among the different fields of research? Also, how is the total amount of money distributed? Is that distribution much different from the previous one?
3. Analogously, how are the number of grants and the amount of funding distributed geographically in the EU?

2. Description of the dataset structure

2.1 CORDIS dataset

The CORDIS Horizon Europe dataset consists of multiple linked files containing information on research projects, participating organizations, deliverables, publications and more. For our analysis we will focus on the core project data (*projects.csv*) and link it with complementary datasets, such as *organizations.csv*, *deliverables.csv*, *publications.csv*, and *report_summaries.csv*. This will enable us to explore relationships between funding and project-level attributes such as research area, project duration and objectives, as well as institutional characteristics (organization type, location) and research output (publications, deliverables)

2.2 External data

We may also consider merging external data sources such as additional institutional information (e.g. university rankings, research funding history, department size) or region-level data (e.g. GDP, dominant economic sectors, R&D investment, average education levels). These additions could give a better understanding of patterns in funding distribution and research output across institutions and countries.

3. Methods/techniques

3.1 Data preparation

A key first step will be to ensure that the data is suitable for analysis by standardizing formats, handling missing values, cleaning inconsistent entries, and creating new, meaningful features. While the exact steps may evolve as the project develops, we outline some potential steps below.

[1] https://commission.europa.eu/funding-tenders/find-funding/eu-funding-programmes/horizon-europe_en

- **Type conversion**: ensures that all variables are stored in appropriate types. For example, date columns such as *startDate*, *endDate*, and *ecSignatureDate* will be converted using pandas.to_datetime().
- **Data cleaning**: normalizes text columns using string manipulations and regular expressions. This is particularly important for variables used in downstream NLP tasks (e.g. *title*, *objective*, or *deliverable description*).
- **Handling missing and zero values**: uses functions such as isnull() and value_counts() to identify missing data or zero values in key fields like *totalCost* and *ecMaxContribution*. Resorts to manual inspection via df.describe() and df.info() can be used to decide how to handle these cases.
- **Feature engineering**: Derives new features from existing variables, e.g. *ProjectDuration = endDate - startDate*, or groupby() to calculate total and average funding per country or institution. We may also create binary indicators for categorical fields such as project status or funding scheme or transform multi-topic fields into multiple columns using one-hot encoding or MultiLabelBinarizer.

3.2 Natural language processing of text fields

Several variables in the dataset contain unstructured text (project objectives, titles, deliverable descriptions). To extract useful information from these features, we can apply a selection of NLP techniques using pretrained models from Hugging Face. We outline some potential examples below:

- **Text embedding:** uses a sentence-transformer like all-MiniLM-L6-v2 to embed project objectives into numerical vectors. These can be used for similarity analysis, clustering, or visualization of related projects.
- **Keyword extraction:** extracts the most important keywords from a project's objective using KeyBERT. This helps summarize project focus and compare content across projects.
- **Topic modeling:** BERTopic combines embeddings with clustering to identify hidden topics in the text, helping to uncover research themes beyond the manually assigned topic codes.
- **Sentiment/tone classification (optional):** uses pretrained sentiment models to quantify tone in project descriptions. These scores can serve as input features in further analysis.

3.3 Machine learning

Using packages such as *scikit-learn*, we can apply both supervised and unsupervised machine learning techniques to uncover patterns in the data, explore relationships between project characteristics and funding outcomes, and support predictive modeling to address our research questions.

3.3.1 Preprocessing and Pipelines

For predictive purposes, the dataset shall have to be split into separate training and test sets and normalization/standardization routines shall be carried out only upon the training data, to avoid data leakage. We choose a test size of 0.2, and a random state of 22 for reproducibility. This separation shall be carried out at the onset of the preprocessing stage, and any inferences made for predictive purposes shall be yielded exclusively from the training set.

Preprocessing steps will be combined into streamlined workflows using the Pipeline class in *scikit-learn*, allowing for reproducible experiments and easy integration with cross-validation and hyperparameter tuning tools. This modular approach ensures that the entire workflow remains clean, traceable, and adaptable.

Considering that the dataset is quite sizeable (with most tables containing no fewer than 10000 entries), a potential challenge may arise from computations carried out with *pandas*. If the BlockManager consolidation sub-routines blow up memory and time requirements for unsuspectingly simple operations, workarounds will be considered by implementing *dask* or *polars*.

3.3.2 Unsupervised Learning

To explore hidden structures and patterns in the dataset, we will apply unsupervised learning techniques such as K-Means, DBSCAN or hierarchical clustering. These methods may help identify groups of similar projects or institutions based. This can be especially valuable for exploratory analysis and for detecting clusters of underfunded or particularly successful entities. Additionally, the e*uroSciVoc* topic hierarchy may be used to interpret and validate these clusters or serve as a reference classification.

3.3.3. Supervised Learning

For predictive modeling, we can use supervised learning techniques to estimate continuous outcomes and classify categorical targets. For example, we may apply regression models such as linear regression or random forests, or gradient boosting to predict *ecMaxContribution* based on the predictor variables and derived text features.In parallel, we may use classification models such as logistic regression or decision trees to categorize projects, with potential targets being a binary classification of high- vs. low-funded projects. These models can help us assess which factors are most predictive of funding outcomes and whether certain features consistently align with successful grants.

4. **Visualization approaches & dashboard vision**

Based on the envisioned analyses, graphical objects yielded from *Shiny/Plotly/Seaborn/Matplotlib* will be incorporated into a dashboard. A first graphical window of the dashboard will feature a map of Europe whereby one could click on different countries to see the investment received as well as the importance of each country as it pertains to specific fields of research. By clicking a country, a distribution of *ecMaxContribution* will appear. The Gini coefficient of a country's funding distribution will be encoded in a static color scale, that will indicate a measure of unequal funding. Additionally, a pie chart will appear revealing the results of the NLP analysis for all granted projects belonging to that specific country.

In a second graphical window, the countries will be compared with one another. To provide insight on the distributions of funds with respect to some other variable of reference (e.g. research area, research output, clusters yielded from the unsupervised learning modules, …), another possible visualization may consist in a set of Sankey diagrams. These shall be either integrated into the second geographical map (replacing the pie charts, which will be used in the first map), or displayed sequentially on their own after the first map. Sankey diagrams may also be implemented with *Plotly*, and allow for an easy and intuitive inference on behalf of the dashboard viewer.