Data Science - Individual Delivery Guide
April / July 2021 - EDA - The Bridge

INSTRUCTOR: Gabriel Vázquez Torres
gabriel@thebridgeschool.es

TEACHER: Clara Piniella Martinez
clara.piniella@thebridgeschool.es

TEACHER: Borja Puig de la Bellacasa
borja@thebridgeschool.es

Delivery explanation

This individual delivery aims to practice different concepts about EDA and APIs. Also, the project will be presented.

The student must choose a subject that he/she prefers. Idically, this delivery will be extended with the Unity 2 and 3 of the temary. This is, apart from this EDA delivery, there will be more deliveries focused on Machine Learning (Unity 2) and Data Science as Product (Unity 3).

Requirements

The next requirements are mandatories:

1. The project must give an answer to a **hypothesis** (explained below).
2. The student will do a presentation and have to document all steps he/she does.
3. The student must divide the tasks that he/she has to do.
4. It is mandatory for the student to use *trello* (or other related) to manage the tasks in different status: TODO, DOING and DONE. BACKLOG and REVIEW are optionals. You have to add all teachers to the board of the project. The best advice we can give you

is that you spend the most possible time defining well the steps of your project. Otherwise you will have to repeat processes many times.

5. The delivery must be sent before 02/06/2021 at 23:59.
6. The delivery must be sent in a *.zip* file by email/classroom with this structure:
    a. A folder **src/** that contains all the source code.
    b. A folder **documentation/** that contains all the documents related to documentation (pdf, presentation, ...).
    c. A folder **resources/** that contains other useful content (images,...)
    d. A folder **reports/** that contains all related to created reports such as figures, html, pdf, etc
    e. A folder **notebooks/** that contains notebooks for your tests.
    f. A folder **data/** that contains the data of the project.
    g. A folder **src/utils/** that contains all the modules used by the *main* file.
    h. A file **src/main.ipynb** that contains all the functionality. This file must only contain imports, pandas, matplotlib, requests,... and calls to your **src/utils/*** modules.
    i. A folder **src/dashboard/** that contains an "*app.py*" file that will run a streamlit dashboard using the necessary command.
    j. There are, at least, these modules inside **src/utils/** :
        i. "folders_tb.py" that contains the generic functionality related to open, create, read and write files.
        ii. "visualization_tb.py" that contains the generic functionality related to pandas, matplotlib, seaborn and other libraries focus on visualizations.

iii. “mining_data_tb.py” that contains the generic functionality related to collect data, clean data and others (wrangling methods such us working with multiples jsons)

iv. “dashboard_tb.py” that contains the functionality related to the dashboard.

v. “apis_tb.py” that contains the generic functionality related to working with APIs.

vi. Others that the student needs.

7. A file **src/api/server.py** that contains the functionality that starts the Flask API. This file only will be executed if it is passed an argument *“-x 8642” (argparse)*, otherwise will show “wrong password”. In the API there is, at least, this GET functions:

    a. One that must allow you to receive a ***token_id*** value and, if ***token_id*** is equal to S, return the jsons that contains the logic explained below. Otherwise, return a string with a message of error.

        i. S is the DNI of the student starting with the letter: Example: “B80070012”.

        ii. The json returned by flask must be the cleaned data of your data.

Hypothesis

Normally, the goal in an EDA project is answering a question or demonstrating an axiom. This is, giving all necessary reasons to explain why the answer to the question is *one specifically* and refute or reaffirm an axiom.

One example of a hypothesis in the project of covid-19 could be:

> *We believe that the alarm state of each country has an impact on the progression of daily infection.*

Presentation

All students must do a presentation about its project. The presenter will use a presentation file (no PDF or code directly) to explain all the steps of the workflow with graphs.

The duration of the presentation won't be longer than 10 minutes so it is really important and necessary to explain the essential points of the work.

The judge will stop the presentation if required.

The project steps

The idea of the project consist in different steps:

1. Find the subject: the student must find the project itself. This is something he/she wants to do.
2. Find the data related to the project: research where it can be and if it is accessible from the public. Focus on this and Data Wrangling.
3. Define a hypothesis: find something you can conclude with your data.
4. Define the necessary steps to demonstrate or not your hypothesis.
5. With the code structure defined and using Python:
    a. Get your data. Maybe you need to use an API, maybe a file. Data Wrangling.
    b. Clean your data. Detects outliers, rare values and reemplace NaN values if needed.
    c. Draw all graphs you need both to understand your data and to show the necessary results.
    d. Explain why from your graphs and others results it can be argued the conclusion.
6. Document all steps, zip the necessary files and send it to teachers' mails & classroom.

NOTE: Do all steps finishing the criteria requirements.

The resources

With the goal of finding all the necessary resources, the student can search all over the internet.

There are pages where you can find both good examples of EDA projects and datasets:

- Kaggle: here you can find millions of examples with millions of datasets. There are different parts where you can learn from novices or experts.
- Googledatasetsearch: here you can find millions of datasets. It is a good page if you want to find the data you need.
- GoogleApis: here you have many apis from different subjects to get data.
- City council pages, statistics pages and thousands of APIs you can find on the Internet.
- Statista
- https://data.world/
- https://ourworldindata.org/
- http://www.fao.org/statistics/databases/es/
- Add the words: API or CSV or JSON when you search on Google
- Other ones in EDA delivery part 1 and Discord.

If the student has not any inspiration, then we can recommend the next EDA subjects:

- Analyzing tweets to determine if some event changes the tendencies.
- Analyzing films datasets to conclude if there are more female actresses or romantic films.
- Analyzing sport datasets to conclude if Messi, Lebron James or Fernando Alonso are the best in their sports.
- Analyzing disease datasets to conclude if there are relationships between symptoms and number of deaths (or other relationship).
- Analyzing climate datasets to conclude if climate change is real.
- Analyzing video datasets to conclude if the funny videos have the most views.

Evaluation criteria

For this delivery, there are different delivery options. Each student must choose what delivery they want to do. **C** is the minimum requirement for this delivery. There is a hierarchy in the options: **C→ B→ A→ A+***

It is not allowed to do:

- B without C
- A without B and C
- A+ without A, B and C

Option C

Apart from all requirements that are written in the **Requirements** section, there are the next mandatory exercises:

1. Document all steps. Structure your code to keep it cleaned using good practices.
2. Use Trello to manage your project
3. The dashboard created must have at least three menu sections. The first one is the welcome, the second one is related to visualize your cleaned data and the third one must call your own flask API and show the json returned.
4. Collect the data. Try to do that each call, it collects the last updated data.
5. Determine and explain if the data is cleaned. If not, then clean it.
6. Show different tendencies for each column in your dataset.
7. Represent, in a pie chart, the time you have needed for each point in the **The project steps** section.
8. Show five graphs in which you can conclude your hypothesis
9. Answer the questions:
    a. Was it possible to demonstrate the hypothesis? Why?
    b. What can you conclude about your data study?
    c. What would you change if you need to do another EDA project?
    d. What do you learn doing this project?

Option B

1. Show the histogram of each column of your dataset with *bins*=5. How are the ranges painted?
2. Which are the columns with the highest correlation? Draw the correlation matrix.
3. Use Matplotlib functions to show all graphs. No pandas directly.
4. Use your dataframe cleaned to insert its values to a MySQL database. The info of the database will be in the *"sql"* channel on Discord. To do that, you will need to create a Table with the dataframe columns. The name of the table is your entire name using "_" between words. Ejemplo: *"pepito_perez_galdos"*

Option A

1. Research to save each plot in local files.
2. Use distributed modules for each functionality. The jupyter notebooks must not have any loop or functions. It only must have the initial imports and the call to necessary functions.
3. Apart from matplotlib, use seaborn to show the graphs.
4. Answer the questions:
    a. Are there outliers or some rare data?
    b. What are the columns that have more repeated values?

Option A+

There are different A+. You can do the ones you want:

1. Create a pull request for the entire project.
2. Are there more URLs from where to collect your data? Explain why. If so, then collect it and merge it with your data.
3. In order to practice OOP and engineering/architecture concepts in computing, define all the functions inside classes and make the program functional using them.
4. Use a program to create the *classes diagram*.
5. Realize a webscraping to a web to get some data related to your project.
6. From a *.py* create the functionality to get your cleaned data from the MySQL database. The *.py* file must be called using argparse.
7. Research and create the Gantt Diagram using a program.