ESILV LÉONARD DE VINCI | ÉCOLE D'INGÉNIEURS PARIS-LA DÉFENSE

Programming for Data Science (MESIIN471624)

Constantin TESTU, Walid KHERIJI, Hugo ALATRISTA-SALAS

Final project

1. Introduction

Information systems can be used to model many phenomena. The modeling task begins with the identification of requirements, including information requirements. Then, the objects that simplify reality are identified and described by their attributes or characteristics. These are then stored in data repositories (databases, data warehouses, data lakes, spreadsheets, etc.). Later, instances of the objects are created, and the database is populated. Data is essential to extract information to better understand the phenomena we are studying and to build indicators to support decision-making (Data-Driven Decision Making).

In this context, in this final project, the techniques learned in this course will be used to extract and visualize helpful information from different data sources.

2. Project goal

The aim of this project is to create a visualisation prototype showing four indicators extracted from the dataset at our disposal. For this purpose, the KDD process for analysing and extracting useful information from a real dataset should be used[1]. Each method or technique used in all steps of the KDD process should be implemented in a function - using *def():* - and add the parameters you think are necessary. Of course, you can reuse or improve methods implemented in labs during previous sessions. Finally, all the functions created must be called from a *__main__* function implemented at the end of the project, which also launches the visualization prototype. See the code box for inspiration to write your notebook. It is important to note that this course's material (slides, labs, Python code) can be used in this challenge.

3. About the dataset

In this project, the dataset at your disposal has been obtained from the NOAA platform (`https://www.ngdc.noaa.gov/hazard/earthqk.shtml`) and is associated with earthquakes recorded from the year 2150 BC to 2020. The dataset is available in CSV format along with this document and it contains spatial and temporal dimensions.

4. Tasks and its evaluation

The evaluation of this challenge is divided into positive and negative points. The following list describes how the points should be given, considering the main tasks to be implemented in this challenge.

1. [*2 points*] Data collection: Create a notebook, upload the data and store it in a structure that allows data manipulation. Explore the data in a way that allows you to fully understand it (using the data

[1] `https://www.kdnuggets.com/gpspubs/aimag-kdd-overview-1996-Fayyad.pdf`

summarization methods provided by Python). You can use the statements we have learned in the course. This stage should at least show the number of columns and the data type in each column, the number of rows and columns in the dataset, the number of missing data per column and the ranking between variables. This last result can be used in the next step.

2. [*10 points*] Indicators construction: Build four queries or indicators from the dataset and explain in detail what each one represents. These will be distributed as follows:

   - [*2 points*] An indicator using grouping queries from the dataset, for example, "the maximal number of ..." or "the sum of the ..."
   - [*2 points*] An indicator that uses data transformation methods, such as the z-score normalization, discretization, or others learned in the course. This group may also include results from the application of frequent pattern mining or other techniques you learn in other courses (e.g. machine learning).
   - [*6 points*] Two indicators using temporal or spatial data features (one of each, if the dataset allows). In this section you will use the forecasting and spatial clustering techniques you have seen in our course. You can also use other techniques learned in different courses.

   Note. All indicators should be explained: how they are calculated, what they represent, and how they will be interpreted.

3. [*4 points*] Visual representation: Visualize the four indicators using the libraries learned in this course (plotly, matplotlib or seaborn). Remember that these visualizations will be used in the next step. You can look the main lecture slides to remember all the visualization techniques.

4. [*4 points*] A visualization dashboard: Use Python Dash (`https://dash.plotly.com/`) to visualize a dashboard containing the four metrics built in the previous step. The dashboard should include the names of the team members and the name of the dataset used for this challenge.

5. [*2 extra points*] External dataset: You can enrich your result with an external dataset. You can also use it to interpret the indicators.

If all the previous tasks are completed correctly, the team will score 22 points. However, some mistakes can penalize the final punctuation. The most common errors are:

1. [*-2 points*] Failure to organize all the code into functions.

2. [*-2 points*] If at least one of the functions does not contain comments about its name, input, and output (see example in the code box). Remember also to add comments to make your code more understandable and facilitate the correction of the notebook. It is worth noting that if you decide to remove a feature, you will need to explain why you are removing it. All decisions should be described in your proposal.

3. [*-2 points*] If the dashboard does not work or there is no evidence that it works. Remember to send an HTML export of the dashboard or other evidence.

4. [*-2 points*] Each indicator should be described in human terms in a text cell (*markdown*). The penalty for each indicator description is -2 points.

5. [*-20 points*] If at least one member is not in the zoom session.

5. Deliverables

1. The notebook in HTML format[2]: The groups must show how they carried out the steps described above (see Section 2) in their notebooks. The grade for this work will depend on how creative they are and how well they follow the instructions given earlier. Comments describing key lines of code are also essential in this submission. This file should be named as *familyNameLeader_nameDataset.ipynb.html*

2. The dashboard in HTML format: Another file named *familyNameLeader_dashboard.html* will be uploaded. This file should prove that the dashboard works and clearly show the indicators proposed in this challenge.

6. Important information

This work must be completed by groups of at **least 4 (four)** and at **most 6 (six)** students. Each team will elect a leader who will upload the deliverables of this challenge. The deadline for the deliverables is at the end of this challenge (3 hours). The grade is a **team grade** (there is no individual grade), which means that all group members must be connected to Zoom during this challenge. The lecturer will be able to visit the groups and ask any group member how the work is progressing. If the group member is not available to respond then the group will automatically receive a **grade of zero (0)** as this is a group grade and it is the responsibility of all members to actively participate in the development of this work.

[2]If you are having problems converting your notebook into an HTML file, you can also send us the *.ipynb* file.

familyNameResponsable_nameDataset.ipynb

```
# Team members and dataset
Dataset: earthquakes
FirstName, FamilyName (team head)
FirstName, FamilyName (team member)
...

# Import libraries
import pandas as pd
import numpy as ny
...

# Function definitions

# Function for reading a CSV file
# Input: the path to the csv file
# Output: a dataframe
def load_data(file_path):
      bla bla bla # Try to comment on each line
      bla bla bla # especially if this line helps us to understand your project
      ...

# Function to bla bla bla
# Input: bla bla bla
# Output: bla bla bla
def function_1(df):
      bla bla bla
      ...

def function_2(df):
      bla bla bla
      ...

# Main block
if __name__ == "__main__":
      # Step 1: Data collection
      file_path = "myfile.csv"
      data = load_data(file_path)
      ...

      #Step 2: Indicators construction
      result = function_1(df) # this is only an example
      ...
```

May the force be with you !!!