INTRODUCTION TO PYTHON
Final Project

GDP Growth Forecast

Statement

Gross domestic product (GDP) is a monetary measure of the market value of all the final goods and services produced in a specific time period. GDP is often used as a metric for international comparisons as well as a broad measure of economic progress. It is often considered to be the "world's most powerful statistical indicator of national development and progress".

You are provided with a dataset containing over a thousand annual indicators of economic development (including *GDP* and *GDP growth*) in hundreds of countries around the world from 1960 to 2010, and you are asked to develop a model that can predict *GDP growth* using data from the past. You can use all the variables that you want from previous years, and you can use the model that you feel it is going to work better. The aim is to have a model that can predict *next year* GDP growth for each country in the world using the available data.

Data

The data is provided as a SQLite3 database containing the following tables:

- `Countries`: All the countries available with some information about them.
- `Indicators`: All the indicators available with some description about them.
- `CountryIndicators`: Country indicators values per year.
- `CountryNotes`: Specific notes about indicators in some countries.
- `Footnotes`: Specific notes about the value of some indicator in some countries at specific years.
- `IndicatorsNotes`: Specific notes about indicators in some years.
- `EstimatedGDPGrowth`: The table for you to store your predictions.

This database is a *slightly* touched version of the dataset available in this Kaggle.

Technical requirements

- The software must be executable through a `cli.py` file placed in the root folder of the project. This `cli.py` executable must accept one mandatory argument *task* (either `train` or `predict`) and perform, respectively, the following operations:

  - `train`: Retrieve the needed data from the tables in the database, build a proper dataset and train a machine learning model with it. The model must be persisted in a file (or files) placed in the `models` folder.
  - `predict`: Retrieve the needed data from the tables in the database, the model stored in the `models` folder, and perform a prediction for the next year. The results of this prediction must be stored in the `EstimatedGDPGrowth` of the database with the year corresponding to the prediction.

  You can add extra optional arguments for this command. For example, a *year* argument to be used in the `predict` task to select which year you want to make the prediction on.

- All the analysis of the data and the experiments you carry out during the exploratory phase should be done in Jupyter Notebooks placed them all in the `analysis` folder.

- You must write reports with the insights you find along the project. These reports must be done with Jupyter Notebooks (in the `analysis` folder) exported as HTML (in the `reports` folder). All reports (HTML) delivered in the `reports` folder must have the corresponding reproducible Jupyter Notebook (.ipynb) in the `analysis` folder. At least two reports are required:

  - One showing the model performance and explaining the results. Where it works well, where it does not and why are the key points here.
  - One other explaining the impact of the features in the model and making your conclusions out of it.

  You should write these reports for a non-technical (business) reader who can understand the conclusions that lie beyond the numbers.

- Your final model must not take more than 50 variables. Beyond that, there is no kind of extra requirement for the model. It is completely open to your criteria to design the model that is most effective and understandable from a business point of view.

- All the logging your software writes must be placed in the `logs` folder, with a different file for each execution.

- You can use all the third-party libraries your want as long as they are available in the PyPI repositories, i.e., as long as you can install them with pip. You must make your dependences explicit in a `requirements.txt` file at the root folder of the project.
- The project folder must contain a `README.md` file with a brief functional and technical description of the software, how to execute it and the name of the authors (name, surnames and NIU of every team member).
- Of course, all the code must be written in Python.

Teams

You must work in teams of two or three members. The name, surname and NIU of all team members must be written in the `README.md` file of your project. You must deliver only one copy of your project per team.

Delivery

You must deliver a *zip* file containing the root folder of your project through Virtual Campus. This folder must contain also the `db.sqlite3` databse with your predictions for 2011 already calculated and stored there. Do not include the virtual environment in the *zip* file (your environment should be reproducible through the `requirements.txt` file). Only The total size of the *zip* file must not exceed 300 MB.

The deadline to submit is November 30, 2020.

Grading

The evaluation of the project will be divided in 4 blocks:

- The correct operation of the software. Both methods `train` and `predict` must work without any errors. As long as they work properly, the whole grade from this part will be given (N_1 over 2.5).
- The quality of the reports. The clarity of the observed insights, graphical visualizations and the conclusions obtained will be valued (N_2 over 2.5).
- The quality of the model. Effectiveness of the model and originality in feature engineering will be valued (N_3 over 2.5).
- The quality of the code. Neatness, modularity and order of the code will be valued (N_4 over 2.5).

The final grade will be the sum of this 4 parts $N = N_1 + N_2 + N_3 + N_4$.