BIOS821_Final_Project

November 1, 2019

1 Final Project

The goal of the final project is to give students an opportunity to implement all skills discussed in BIOS 821 and to demonstrate they can identify software tools that can be leveraged to solve components of reserach projects.

2 Overall Goal

In this project, we will use the freely available European Soccer Database. Your goal is to build a working analytic data set that could be used to create a descriptive figure to visualize trends in the data. We have not discussed creating figures in this course, and you will NOT be expected to do so for this project. Instead, you will be expected to build an analytic data set from the raw data files that could be used to create such a figure. In doing so, you will be expected to explore the data, modify the data by creating new features, and build a pipeline that would support the creation of such a descriptive figure. You will be graded according to the elements listed in `Project Components`. Ensure that your solution meets these requirements. Beyond those components, you should adhere to the following principles:

1. **Everything Programmatic** - Nothing should be downloaded from the internet via your browser. *The only exception is obtaining a copy of the European Soccer Database!* There should be a script or function that you can point to that retreives everything else needed for this project.
2. **Everything Reproducible** - the environment where you execute code needs to be documented. Ideally this will be a `docker` container. If you cannot use docker, find a way to communicate the exact environment where your code will execute as expected.
3. **Everything Versioned** - as you create code, use `git` to commit often. Make your commit messages meaningful. We will not use the fork and pull method. Instead, you will create a project at: `https://gitlab.oit.duke.edu/`. **When it is time to turn in your project, you will send me the link to this gitlab project.**

3 Project Components

The final project is comprised of two deliverables: 1. An in-person code review. 2. GitLab repository (code, notebooks, files)

The final grade will be a weighted average of these two components. The project (i.e. code, notebooks, files) will count towards `80%` of your grade and the interview portion `20%`.

3.1 Code Review Rubric

You will need to sign-up for an interview for this portion of the project using Google docs spreadsheet to be posted after class on Thursday Nov 7, 2019.

This portion of the project will consist of an in-person code review. All questions asked will come from the work that you submit. Grade assignment is as follow:

1. `[90-100]`: Clear explanations, effective for both content matter experts and people outside the discipline, integrated general principles and details
2. `[80-90)` : Explanations that were sound, but not fully understandable to people outside the discipline (e.g., too technical)
3. `[70-80)` : Explanations that cover some content accurately but were difficult to follow (e.g., responded to something other than question asked).
4. `[0-70)` : Answered most or all questions incorrectly.

3.2 Project Rubric

Your project must comply with the components listed below. The points for each component are given within this section.

3.2.1 [1] `git` started (10 POINTS)

Create your project in https://gitlab.oit.duke.edu. Store all files, code, and notebooks in this project.

Many data scientists and developers adopt the mantra of `commit often`. By commiting often, you are ensuring that no commit is too large to review and that you don't accidentally overwrite functioning code. This section of the project will be graded by examining your cadence of commiting, commit history messages, and ability to use a central repository that is remote.

You should use the `ssh` protocol, not `https` to demonstrate your knowledge of this topic.

3.2.2 [2] `data` retrieval (30 POINTS)

You may download the sqlite database via your browser from here: https://www.kaggle.com/hugomathien/soccer/download. This is the one and only time that we will use a manual process in this project. Since we are doing so, how might you communicate this exception to those who review your project on gitlab? **(5 POINTS)** This database (and it's corresponding compressed version) should never make it to gitlab. It should only reside on your local copy of the project. **Ensure that this database can never be uploaded to gitlab. (5 POINTS)**

Task 2.A (10 POINTS) You will need to create a script that meets the following specifications: 1. Takes 3 positional parameters. 2. Conditionally downloads the file: `https://gitlab.oit.duke.edu/bios821/european_soccer_database/raw/master/esdb.md5` and stores it according to one of the positional parameters, if and only if it does not already exist. 3. The other 2 positiional parameters should be used to optionally move and decompress the database already on disk (i.e. soccer.zip). Only execute this code if the `md5` sum within the esdb.md5 file matches that of the database you downloaded (i.e. soccer.zip).

Task 2.B (10 POINTS) Augment your data by geocoding each location in the `country` table. This should be done by creating a new table called `latlong`. This table should contain 5 columns: `id`, `country_id`, `country_name`, `lat`, and `long`. You can use mapbox forward geocoding to accomplish this. Remember, you will need to accomplish all of this in a reproducible manner; specifically, you will need to create code that accomplishes this task. Simply entering commands in a bash or python console will work, but that approach not count toward full marks on this component. You will need to sign up for an account on mapbox, but you are not required to enter credit card information. Part of this process will require you to use sensitive information (i.e. your token) in your RESTful `api` calls. Your code will need to know how to handle sensitive information without actually exposing it in the code.

3.2.3 [3] `package` it (30 POINTS)

We've talked about modularity and composability in class. In this component, you are required to package the fuctionality in Task 2.B. You will need to create your own python package that contains the following components: 1. Well-defined application programming interface (api) 2. Uses Object Oriented Programming concepts, including at least one subclass and one class property

3.2.4 [4] `pipeline` build (30 POINTS)

In this component, you will need to build a pipeline that executes Component [1], [2], and [3] as well as implements new programming tasks that builds from these components.

Use a *single* jupyter notebook to accomplish the following tasks in the order give below:

1. In Component [2] Task 2.A, you developed a script that conditionally decompressed the database and moved it to a new location. In this jupyter notebook, execute that script and demonstrate how it works.
2. Utilize the package that you developed in Component [3]. By executing this code you will be adding a new table to your database.
3. Recall that a goal of this project is to build a working analytic data set that could be used to create a descriptive figure to visualize trends in the data. Suppose you wanted to display the average number of goals scored per season for each country using a line plot (i.e., each country's trajectory would be represented as a line in the figure where the y-axis is the average and the x-axis is the season). Complete the task below to create a data set that would support the creation of this figure. You do NOT need to generate a figure.
    a. Derive a `sql` query that returns a data set that contains a column for `country`, a column for `season`, and a column for `average number of goals scored per game`

for each country-season combination.

4. Using the data set built in the previous task, use pandas to:
   a. Create a new feature called good_season that takes the value 1 if the average number scored per game in a given season is greater than `2.75`.

   b. Create a summary data frame that displays the number of "good seasons" for each country.