**Website Development Project Log**

***Week 2 (January 9 – January 16, 2022)***

Create a website that will showcase data analyses conducted on historical Olympics data in a visually appealing and interactive manner & generate excitement and enthusiasm for the upcoming 2022 Winter Olympics in Beijing (Beijing 2022)

The following items detail the various activities conducted within the second week of the group project:

**Website Creation Tasks**

* Successfully created organization file structure for website (static/js/images/css + templates/)
* Successfully created parent html file to serve as home page based upon assistance from websites and NYT GitHub project
  + Developed banner with multicolored text
  + Developed colorful navigation bar with links to other pages that will be a part of the website
* Successfully created “trends”, “visualizations”, and “interact” html pages and linked them to parent page
  + Successfully resolved issues around site navigation so that all 4 links work and the user can click between pages successfully
* Began reviewing Weekly Challenge #10 Mission to Mars html code as a model for how to scrape the Beijing 2022 website for the latest news on the upcoming Winter Olympics
  + Identified a potential challenge with scraping Olympics website, i.e., permission and copyright
    - Resolved this issue 1/10/22 after discussing during OH by clarifying that the Olympics robots.txt allows for scraping
  + Reviewed the Beijing 2022 website to identify elements to scrape
  + Determined that web scraping would not be able to be utilized efficiently given the design of the Beijing 2022 page
  + Decided to include the Beijing 2022 website banner in the homepage and a link to Beijing 2022

Graphical user interface, text

Description automatically generated

* Began working on “interact” html page creative elements
  + Developed a fly-in text statement for audio player, which later was changed to static text in red
  + Centered audio player on page
  + Created the hyperlink to the music source
  + Identified a challenge with audio (i.e., audio players do not consistently play when you click on other pages)
    - Researched into options for how to handle this
    - Determined that the audio player should be housed on the “Interact!” page as an interactive feature with some history about John Williams’s iconic piece commissioned and premiered in 1984.
    - Successfully updated pages so that the “Interact!” page only includes music

Graphical user interface, text, application

Description automatically generated

* Successfully created “Trends” page
  + Discussed Elisabeth’s work with Tableau and whether or not Tableau dashboards could be uploaded to the webpage during class on Monday 1/10/22
  + Elisabeth provided the embed code for preliminary visualizations to test
  + Successfully built embed code into the “Trends” page and shared final results with team via Slack
  + Successfully updated page to include visually appealing elements, e.g., having header of “Dive into Olympics Dashboard!” as a fly-in text and a border around the Tableau element

Graphical user interface, application

Description automatically generatedMap

Description automatically generated

**Website Deployment**

* Inquired about whether JavaScript can be used on Heroku as well as best deployment technology to use for webpage during OH
  + Ed agreed to provide a lesson on webpage deployment and Heroku during class on Wednesday (1/12/22)
* Began exploring how to deploy machine learning models in Flask
  + The pickle library for serializing standard Python objects.
    - Benefits:
    - The pickle module keeps track of the objects it has already serialized, so that later references to the same object won’t be serialized again, thus allowing for faster execution time.
    - Allows saving model in very little time.
    - Good For small models with fewer parameters like the one we used.
  + The joblib library for efficiently serializing Python objects with NumPy arrays.
    - Ideal for the large models having many parameters and can have large NumPy arrays in the backend.
    - Can only save the file to disk and not to a string.
    - Works similar to pickle `dump` and `load`
    - Most fitted for sklearn estimators.
  + Successfully concluded that it looks like joblib is the one we should use!
  + Successfully tested joblib in a practice file using logistic regression with Module 17 to open and then run the model based on “test” data!

Graphical user interface, text, application, email

Description automatically generated

* Attended Ed’s lecture on webpage deployment on Wednesday (1/12/22) and worked with team on important issues related to EDA and ML as well as GitHub
* Attended OH on Saturday (1/15/22) to discuss EDA and ML issues
* Began working on “interact” html page ML elements
  + Completed preliminary coding for HTML to capture user input, which included drop-down menus
  + Built user input forms with borders for Summer and Winter Olympics
  + Discussed with Nikhila on Sunday (1/16/22) a way to encode data that could be used for the webpage, i.e., Label Encoder after research results obtained by Geeks for Geeks <https://www.geeksforgeeks.org/deploy-machine-learning-model-using-flask/>

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated