Project Document for Data Analytics Team Project

Data Analysis Web-based Dashboard to show Stock Market Activity for 5 specific market sectors

The dashboard charts should be able to start to give enough data that could be cleaned and normalized to predict if the stock price will go up or down based on activity [and maybe historic trends]. The machine learning component would be to take the data once it’s cleaned and normalized and push it through 3 different learning models to see what the predictions are from each model. To make this work, we need to identify a learning set – which could be based on 7 days of data, 30 days of data and 365 days of data – this would be the next phase of development.

https://maternal-mortality-project.herokuapp.cocd m/methodology - example

* Web assets and libraries
* Bootswatch – theme – Spacelab <https://bootswatch.com/spacelab/>
  + - Bootstrap current release is v5.0.2 <https://getbootstrap.com/>
    - D3.js – release version – <https://cdnjs.cloudflare.com/ajax/libs/d3/4.11.0/d3.js>
      * Used with Bootstrap version 3.3.7 [assignment 14 – UFO Search]
* D3 assignment with plotly.js [assignment 15 – Belly button]
  + Used Bootstrap version 3.3.7
  + <https://d3js.org/d3.v5.min.js>
  + <https://d3js.org/d3-scale-chromatic.v0.3.min.js> - for colors
  + Plotly.js – release version – <https://cdn.plot.ly/plotly-lastest.min.js>
* D3 assignment for dynamic scatter plot [assignment 16 – Scatterplots]
  + Used bootstrap 4.1.3
  + <https://code.jquery.com/jquery-3.3.1.slim.min.js>
  + <https://cdnjs.cloudflare.com/ajax/libs/popper.js/4.1.3/js/bootstrap.min.js>
  + <https://d3js.org/d3.v5.min.js>
  + <https://cdnjs.cloudflare.com/ajax/libs/d3-tip/0.9.1/d3-tip.js>
    - Javascript that is new – chart.js [has a machine learning component]
    - To test javascript without the Flask API – use python -m http.server in gitBash
* Each chart has its own javascript code
  + Bubble\_chart.js – Norman - Dashboard
  + Line\_chart.js – Emisael [with two y axes] – Dashboard – two inputs
  + Candlestick\_chart.js – Rupesh -Dashboard
  + Dynamic\_Scatter\_Plot.js – Eve/John -Dashboard – shows stocks by industry sector
  + DailyActivityExchanges\_Chart.js Eve -shows on Homepage and Dashboard
  + Optional – trend prediction for industry or individual stock – will it go up or down
    - Three learning models
    - Normalization step for data
    - Learning set - identified

[assumptions: Using HTML5, CSS and Javascript with Bootstrap components that are tested and compatible]

* Webpage template has navbar with menu, Header, page content section, Footer

[give links to the templates folder on git]

* Master Page
* Navbar menu will be horizontal with
* Home, Home page is a splash page with introduction
* Dashboard, [Layout will get done last - each chart on the dashboard can be clicked through to display on its own page] – 4 chart pages will be set up
* Methodology, [Describes data sources, data collection, data cleaning; APP architecture using Flask API program flow, how the webpage listeners are triggered to access data and display it on a chart or page]
  + - * Methodology page should have an entry from each person for the work they did
* If there is a machine learning component – the machine learning can be done on python or javascript or both
* About Us [grid layout] – send in the picture you want on the page
* Flask API
  + Needs Requirements.txt for the Heroku build
  + Needs Runtime.txt
  + Needs Procfile to set up app instruction
  + app.py – wrapper for Flask application – set up and deploy initial flask api to confirm a webpage can be loaded on Heroku and data can be sent to Mongo db and retrieved.
  + db.py – database create instruction if needed
  + ##.py – other functions that are used for processing data as needed – will be needed for machine learning step
* Data Repositories
  + MongoDB – for saving historic data by stock and close date
  + S3 bucket to save daily processing files that are used by Heroku