ML-Silver

* Topic
  + I intend on creating a machine learning API and implementing that into a website simple enough that anyone with a browser can create and use their own machine learning models or use models created by other users. The plan is to have 2-4 different machine learning algorithms for the user to choose from when creating a model. User’s will be able to create their own models based on the data they provide and can either choose to flag their model as private or public (meaning any other user can access it). From there the learned model will be saved to the cloud using MongoDB Atlas. Any user will also be able to browse or search the list of publicly available models and can choose to just use them in the browser or to download them for personal use.
* Core Requirements
  + ML API (Python):
    - Data Manipulation from a CSV with pandas
    - Decision Tree Classifier (predict continuous values)
    - Decision Tree Regression (predict discrete values)
    - saving the .dot file and .joblib file to the database
  + Python Backend:
    - Saving users to DB
    - Loading users from DB
  + JS Backend:
    - importing/exporting the .dot file and .joblib
    - render .dot files
  + Svelte Frontend:
    - Login Page
    - Model Creation Page
    - Personal Library Page
    - Model Viewing Page
* Stretch Requirements
  + ML API (Python):
    - Being able to set models as public or private
  + HTML/JS:
    - Ability to view, download, or use other people’s models
* Tech Stack
  + Python
    - Scikit-Learn
    - Pandas
    - Flask
  + HTML/CSS/JS
    - Svelte
  + Express/Node
  + MongoDB Atlas
  + Docker
  + Nginx/Eurika
  + RabbitMQ
* Proposed schedule
  + Week 1: to finish the data manipulation with pandas (API)
    - I will use pandas (python library) to take in the user’s csv file and clean up any unusable or null data
  + Weeks 2-3: to finish the machine learning generation (API)
    - Use pandas and scikit to split up the data into input columns and output columns
    - Specify which learning model type it is
    - Choose from n different trained options
    - Input data and get projected models back
  + Weeks 4-5:
    - Use scikit to create tree model files (.dot) and learned instructions files (.joblib)
    - Set up importing and exporting .joblib files
  + Week 6:
    - Dockerize the backend and start connecting it to calls on the front end
    - Set up front end to have small endpoints to take in files for testing
    - Add endpoints for creating an account
  + Week 7:
    - Create a better formatted login screen, model creation screen, and model library screen
  + Week 8:
    - Connect the front end to the backend
    - Start passing files from front end to back end and vice versa
  + Weeks 9-10 (flex):
    - Start flex goals
    - Create visibility modifies for the models
    - Create a global models page and add endpoints to connect all of the models to it