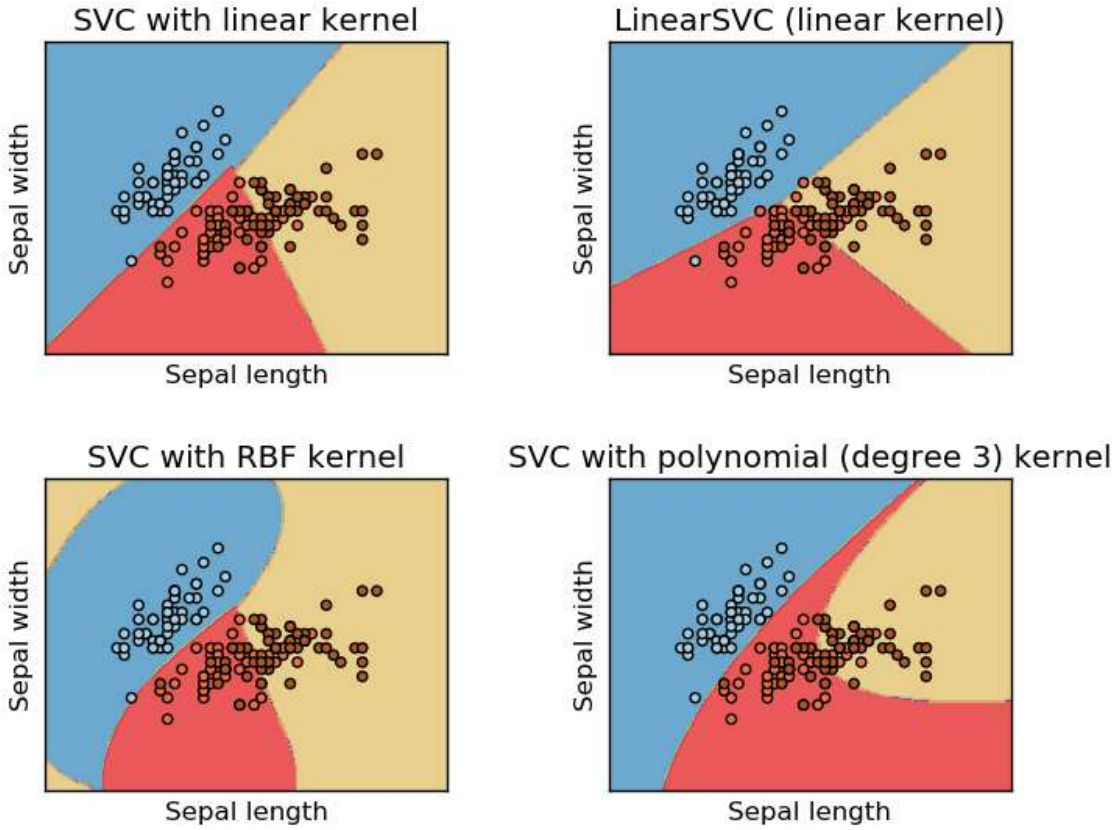
Zack Jaffe-Notier
CS 406 2-Credit Project

Machine Learning Fundamentals

This project will utilize the data collected during my first 2-credit 406 project, which focused on data gathering and visualization. While I had originally planned for the first project to cover the fundamentals of statistics, far more time was needed for learning data collection and cleaning than was expected. This actually works out nicely because every "Intro to ML" course I find first reviews statistical concepts like multivariable linear regression, Bayesian inferences, and regularization. After (hopefully) getting a better grasp of what these techniques are and how they are useful, I will begin to familiarize myself with the different types of Machine Learning models available and which one would be best suited for my needs. However, if the complexity of the best suited model is beyond my current ability, I may find myself implementing far more basic ML methods to achieve my goals; my goal for this project being a trained model that is capable of predicting the community consensus (rating) of a game based on a myriad of factors (age range, publisher, number of players…). While this will be the primary goal, if it is possible to re-target the model to predict other aspects of the game (classification, community engagement, price, etc.) based on other factors (mechanics, theming, target age range), then I will also train models for these purposes. The user should be able to confidently predict at least one of these values – rating, classification, price, or engagement (number of comments) – with the model after it is trained when given other details about the game.

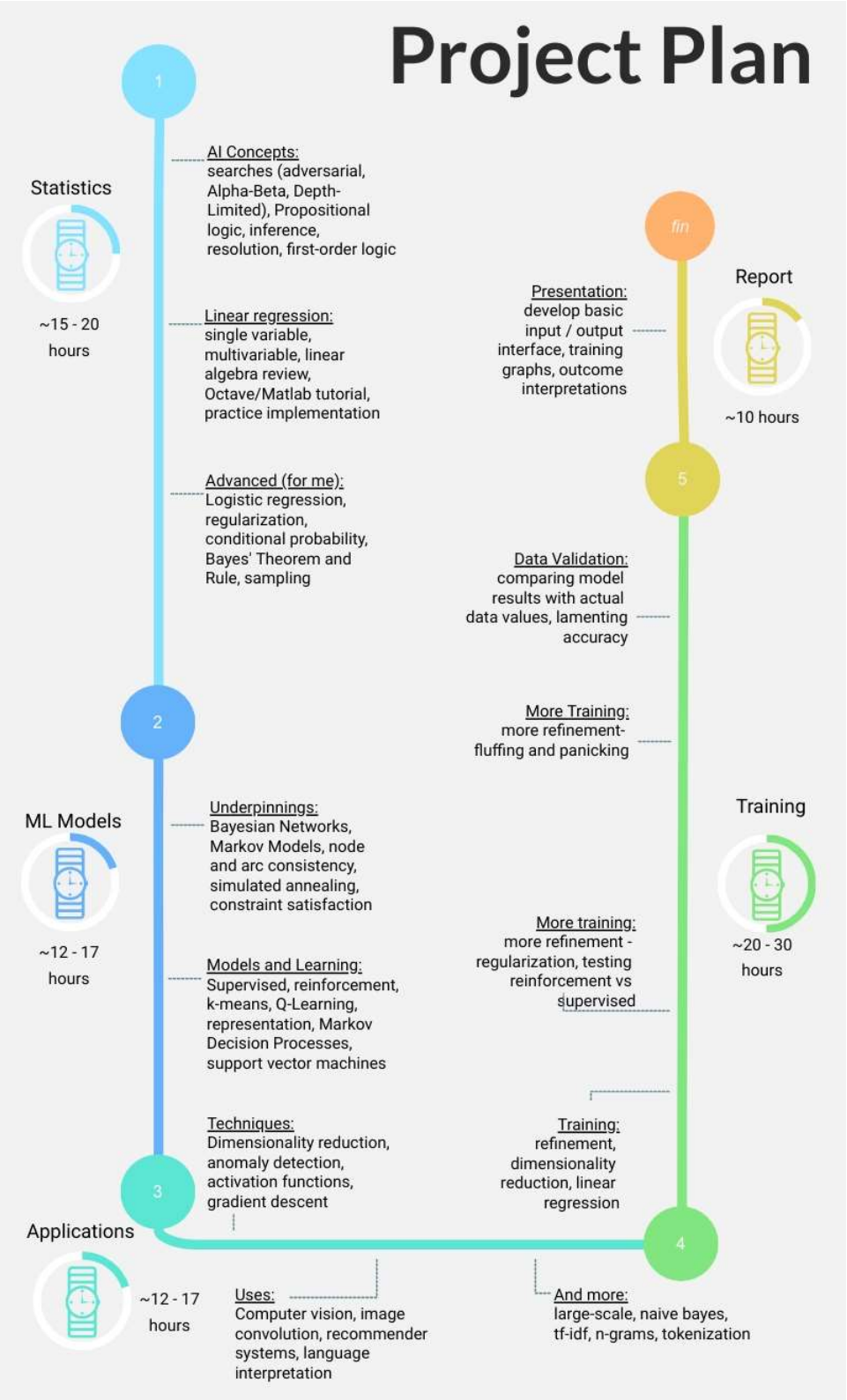
In order to accomplish this, I will be following along with Andrew Ng's Intro to Machine Learning course to learn the more practical aspects of ML while also following along with CS50's Intro to AI course in order to better familiarize myself with the concepts and terminology so that I may better implement what I'm learning. In order to follow along with these courses I will need to use platforms like MatLab to practice certain statistical methods. I plan for everything to be written in Python in PyCharm, as it has excellent handling for packages like Pandas that will allow me to spend more time coding and less time setting up a development environment. Pandas also allows for simple yet attractive graphs to be produced, which will be necessary throughout the training process in order to determine which methods are most accurate at predicting various characteristics of the game. For example, these are various model predictions of Wild Iris flower types based on physical attributes of the plants:



From https://scikit-learn.org/0.16/auto_examples/svm/plot_iris.html

This is an example of “boundary training” where each region is a particular classification. Hopefully I will be able to generate similar regions for board game classifications, but it will require a tedious process of finding the most statistically significant factors without over-fitting to the set of training data.

I have broken down the expected workloads and organized them into a timeline. This timeline is also representative of one that will be part of the final project, except it will reflect the actual amount of time spent on each subject and what precisely was covered.



Made with https://www.visme.co/timeline-maker/

Last quarter I did not leave myself any time to create a basic interface for users to interact with, as I was primarily focused on getting my code to even work while gathering data, cleaning, and interpreting it. This quarter I plan to place a higher prioritization on that, even though it is doubtful this will be shown to anyone outside of school. I will personally feel better about the project if users can have a concrete experience and something to actually "use" as opposed to having python script that needs to have variables adjusted for each run to produce new output. While the vast majority of the time will be spent wrapping my head around the fundamentals of this field, I am determined to have this project predict at least one of the target elements with reasonable accuracy for the user. If it ends up being useful to the user, I may also generate a series of graphical representations of where their hypothetical game falls in the field of data used for training and how the model is interpreting it to give them the answer that it determines is best. The visualization would need to be custom for the type of output being sought (eg. A boundary training graph for game classification), so this may be beyond the scope of the project.