The Big Idea: *What is the main idea of your project? What topics will you explore and what will you generate? What is your* ***minimum viable product*** *? What is a* ***stretch goal*** *?*

For our project we are using past product history data from Google Shopping, Amazon, Ebay, etc. (the exact sites are still unknown) in order to find when the best time it is to purchase a product. We will explore data scraping and data analysis for prediction.
Minimum Viable Product: Analyze past data that predicts the best time for a product to be purchased.
Stretch Goal: Analyze sentiment in reviews and other factors that might affect price patterns. Create a GUI and visualize the data in a helpful way.

Learning Goals: *What are your individual learning goals for this project?*
Samantha Young - Getting better with GUI and utilizing apis
Noah Rivkin - Create predictive/trend analysis algorithm
Vicky McDermott - Get better at using unfamiliar APIs and understanding/analyzing trends
Katya Donovan - Improve at using APIs and collaboration (merging codes) as well as analyzing data to make predictions

Implementation Plan: *This will probably be pretty vague initially. Perhaps at this early juncture you will have identified a library or a framework that you think will be useful for your project***.** *If you don't have any idea how you will implement your project, provide a rough plan for how you will determine this information.*

Part 1: Data Collection
Includes Price Data and Sentiment Data
Part 2: Data Analysis and Prediction
Finding trends in prices and finding the most positive reviews and finding the intersection between the two
Part 3: Visualization of the data on a webpage. Including user interaction of picking data.
Part 4: Sending notifications/email about when it is the best time to buy.

We are planning to use the Google Shopping API to look at product price history. We will have a section of our code dedicated to pulling the data, a section for analyzing and predicting the data, and then one for showing the data. We will research various libraries for user interaction and visualization and choose one that seems appropriate for our data.

Project schedule: *You have 6 weeks (roughly) to finish the project. Sketch out a rough schedule for completing the project. Depending on your project, you may be able to do this in great specificity or you may only be able to give a broad outline. Additionally, longer projects come with increased uncertainty, and this schedule will likely need to be refined along the way.*
By March 27 - Determine where we are getting data (or how we are faking it) and start collecting some price data. Write and prepare for architectural review. Have a good idea of what the classes and functions we want to include in our project are.

By April 3 - Be able to get use data to do something interesting - look at patterns, find lowest price and time of year based on an input product, etc.
By April 10 - Have a basic GUI working and start thinking about how we want to visualize the data. Start looking into accessing review data or some other interesting factors which may affect price patterns.
By April 17 - Start working on project website and improving project towards our stretch goals. See if we can use machine learning to improve our algorithm and help it learn from itself and the patterns it sees as time goes by.
By April 24 - Polish off project and get ready to present it to the class. Finish off project website and start project poster.
By April 30 - Turn in project website and project poster.

Collaboration plan: *How do you plan to collaborate with your teammates on this project? Will you split tasks up, complete them independently, and then integrate? Will you pair program the entire thing? Make sure to articulate your plan for successfully working together as a team. This might also include information about any software development methodologies you plan to use (e.g. agile development). Make sure to make clear why you are choosing this particular organizational structure.*

We plan to split the project into four basic parts in the beginning and have tri-weekly check-ins where we meet up and share what we have done. That way we are up to date about where others are at in the project and can change code depending on the changes that others have made. In addition we will have a longer weekly meeting, where we work together for a couple of hours to make sure that everyone knows how to proceed. Our current plan is to separate the code into two main parts, with four sub groups, and have a pair try to program together most of the time. We are choosing this particular organizational structure because we all like to work with others while we code in order to ensure that the code is understandable and that it will merge well with other's code. We will also comment our code a lot! We are choosing this particular strategy because it ensures strong collaboration between two people all the time, and then has the pairs collaborate. That way, no one will be lost, and code will be compatible more easily. Plus in terms of scheduling, it is much easier to schedule between two people than with four.

Risks: *What do you view as the biggest risks to the success of this project?*

One major risk to the success of this project is the possibility of work overload. We have attempted to minimize this risk by setting a achievable MVP, so that even in the case of difficulties like a sick team member or similar problems we will be able to complete the MVP. Another risk is a lack of available data. Our preliminary investigations have shown that there is online data available, but it is still uncertain whether there is sufficient data to find clear trends with a reasonable degree of certainty (at least 2 standard deviations, preferably). However, we feel certain that we can find enough data to accomplish our MVP.

Additional Course Content: *What are some topics that we might cover in class that you think would be especially helpful for your project?*

It would be helpful to learn more about predictive algorithms. We have the skills we need about APIs and internet scraping. However, learning about servers and web applications would be very useful.