Solution 1- Web application development using Python.

Done:

The application is built using Flask, Python 3.6, HTML and CSS. It is a web page that consists of a text field to take input from user as a single review, analyze the result and print its output. We trained our classifiers, which for now is including simple Naïve Bayes and Bernouli Naïve Bayes classifier, on 80% of our labeled data and put the rest 20% for testing. We got **70%** accuracy on testing data with simple Naïve Bayes and **67%** testing accuracy with the help of simple Bernouli Naïve Bayes. Up to now, the results we obtained is promising and shows that we can optimize our application.

Doing:

We are adding an option to upload a text file containing multiple reviews rather than getting just one review from the user, the sentiment analyzer then analyses all the reviews in the file and labels them as positive or negative and then return a file that can be downloaded by the user which contains all the reviews analyzed and labelled as positive or negative. We have also looked at the prospective of providing and option to the users to enter the URL of the product and all the reviews for that product will be analyzed and presented to the user.

Roadblocking:

For adding the option of adding the URL, we faced some problems since we could not find a tool or way to integrate our program which can dynamically extract the reviews present at the website, given just the URL. But we tried amazon_scraper 0.3.3 which is a hybrid web scraper/ API client for python which is great at extracting the reviews of product dynamically and returns an array of reviews. We are currently testing this as the output needs to be refined in a standard format to be provided to the classifier. Once done, we will just integrate it with the web page code and the implementation for the web application will be ready for integration testing.

Solution 2: Chrome extension

One of the implementations of our product will be through a Google Chrome extension. The Google Chrome extension will be installed like any other Chrome extension - through the Chrome web store. Once installed, our extension will show up as an icon on the Google Chrome taskbar. Clicking this icon while viewing an Amazon page about a product will cause a window to pop up, displaying a summary of the reviews for that product. Clicking the icon while viewing any other page, including the Amazon homepage, will simply print the message, "No Amazon product detected".

Done:

We have constructed the basic skeleton of a Google Chrome extension with a couple of functions working. Our extension is so far able to retrieve the URL and a picture of the webpage the user is currently viewing, which we will later be able to send to our Amazon API to retrieve reviews of products. Our Google Chrome extension also has a popup window that prints the retrieved URL and picture.
The sentiment analysis portion of our Google Chrome extension project, which uses the same Python code from our website-oriented project discussed previously, is mostly complete. It can learn from a training set of data and return positive/neutral/negative scores when new sentences are inputted (results can be found in Table1). These scores can then easily be translated to a new star rating and displayed to the user.

Doing:

The summary of reviews displayed through our Google Chrome extension will show the percentages of positive, neutral, and negative reviews based on our sentiment analysis. A breakdown of positive, neutral, and negative reviews over time will also be displayed to show the user how sentiment towards the product has changed over time. Also, included will be a new star rating which will be calculated based on our sentiment analysis. Finally, pictures which have been posted by other reviewers will be displayed to help the consumer get a better idea of what the product looks like in real life.

Roadblocking:

A major hurdle which we are currently working on is figuring out how to run Python code from the Google Chrome extension and use it to send data back. Since we are using Python for its natural language processing abilities in our sentiment analysis, it is vital we come up with a solution to this problem quickly. So far, we have found one potential solution to this problem online, which we are still testing. The other challenge we are facing is the retrieval and display of pictures from other user reviews. The API we are currently using doesn't have any obvious way of doing this, so we might need to find some other API to accomplish this task or find some other kind of workaround.

Overall this part of the project has good headway because we are using a lot of the same code from our first implementation of our software project and have a basic skeleton of a Chrome extension with a few working functions. Still, there remains the major technical hurdle of figuring out how to run Python code through a Google Chrome Extension (which is traditionally written in Javascript), as well as how to retrieve pictures through the API.

Solution 3: Application for android

Done:

We are also trying to integrate our method as an android application. We have already decided about the interface and have also the idea of implementing Naïve Bayes algorithm in java, which we will use it in the application programming.

Doing:

In the application, there will be an option to whether copy single review and paste to check the efficiency of the app or you can copy and paste the URL of the webpage and see the result. But this method will be only applicable for amazon item review.

The application interface would be like a textbox to copy and paste single review or you can copy and paste the URL of the webpage. **You must copy and paste the URL from amazon website not from amazon app**. After copying a single review, the app will first train itself on the already provided dataset and then analyze the review to give the result as a single word of positive or negative.

If you copy and paste the URL from amazon webpage, then it will first scrap the reviews from the webpage to store it on and then will train the application based on dataset provide and then will analyze the scrapped reviews and give result in form of pie chart.

The app will be available as apk file or if applicable then also on the google store. It will only available for android and not for another OS's. The idea for making an android application came to our mind because almost all people have a smartphone these days and not every time we have access to a computer so other two methods will not be that beneficial if compared to usability and availability. Anyone, having an android mobile can use it with ease and can get the result in seconds. This will also ease the method of application since making an android application is very common and can be done easily.

Roadblocking:

Every time you start the application to check either one review or paste the URL to check reviews in a bulk, you need to give time to the application to first train itself on the dataset provided and then only it can answer your queries. We are trying to make it like it does not have to train again and again but we are not yet sure but we will try our best to do that because it will ease its use.