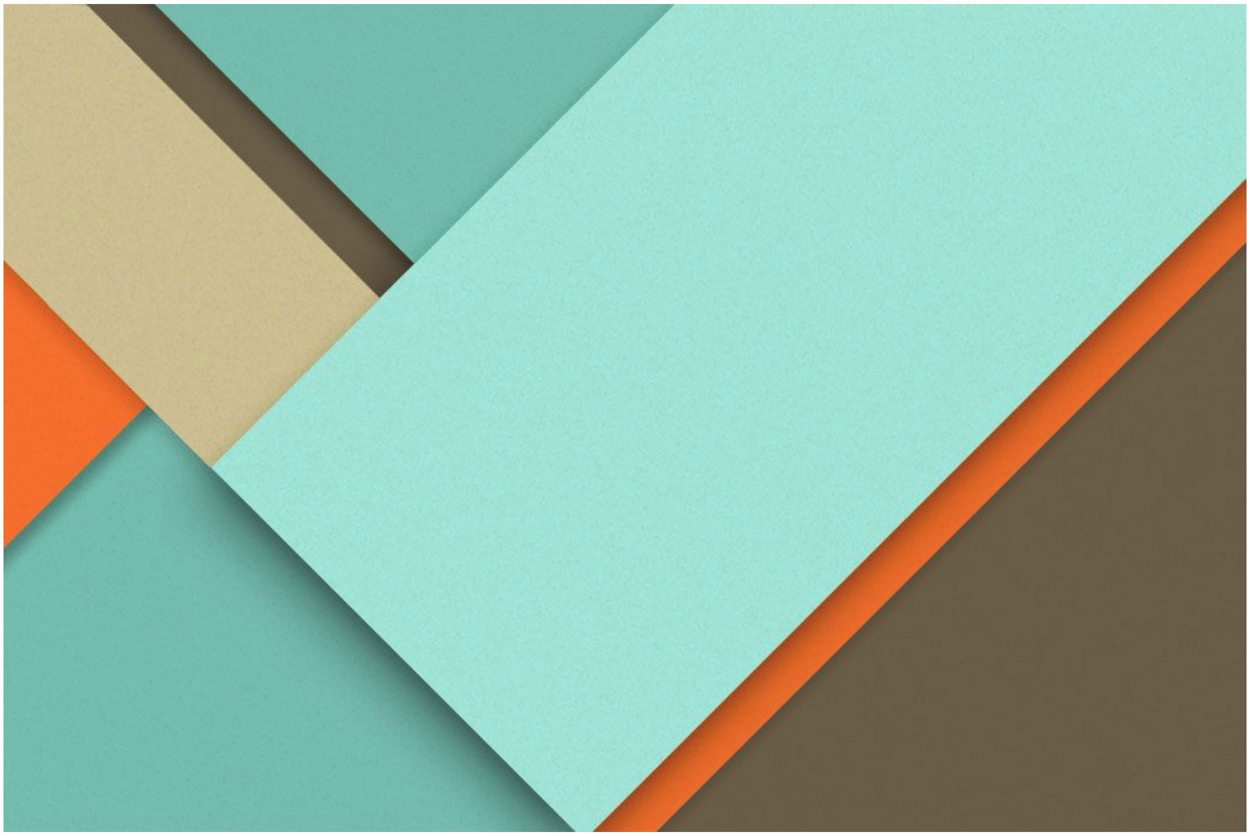


Amazon Product Review Analyser

Devin Thomas

Contents

References

[1] AWS Amazon Comprehend: https://aws.amazon.com/comprehend/?nc2=type_a

[2] Stanford Research Paper: http://cs229.stanford.edu/proj2018/report/122.pdf

[3] Fakespot: https://www.fakespot.com

[4] Python: https://www.python.org

[5] Django: https://www.djangoproject.com

[6] React: https://reactjs.org

[7] React Bootstrap: https://react-bootstrap.github.io

[8] BeautifulSoup: https://pypi.org/project/beautifulsoup4/

Analysis

The Problem

In the first quarter of 2018, e-commerce increased by 2% but rose to a 20% increase in the first quarter of 2019. Presently, a larger number of people are moving from physical shopping to online shopping and as a result e-commerce companies have benefited greatly. Amazon, the largest e-commerce company in the world, saw net sales rise by 31% to $232.9 Billion last year. However, an issue with online shopping is that we cannot guarantee the quality of the products that we are buying because all we see is a picture of the item. As a result, we often rely on other buyers and their experience to form our own decisions and ultimately buy or ignore the product. Several factors come into account such as country of origin, seller, reviews, ratings and number of both reviews and ratings. All these factors allow us to make justified decisions to ascertain if the product is worth it.

My solution proposes to streamline the process of deciding whether a product is beneficial and if the user is likely to be satisfied with it. By allowing the user to select a product from Amazon's website, my solution will analyse the entirety of the information provided about the product in order to give a percentage confidence that the product is good or not. This complete solution is not currently available anywhere due to the niche nature of it. However, it is essential as it would provide a good tool in being able to trust products that appear on Amazon. By buying the best product first, it will save the user from having to return items and buy another; a process that is laborious and tedious. In cutting down this process the user can therefore streamline the process of online shopping through Amazon. Additionally, businesses would benefit by looking at their buyers' opinions about their products and researchers would also be able to use the information to do in-depth analyses of market trends which may lead to better predictions of the stock market.

A computer is best suited to solving this problem as there is an inherent bias that lies within humans when making decisions. For example, we may decide that the product with the best rating is the best when in fact another cheaper but identical product would be ideal as it saves money. The system will be required to read through possibly thousands of reviews and determine the sentiment of each review in order to depict whether a product is of good quality. A computer can therefore read these reviews much faster and gain a less biased view of the product. Additionally, the system can use computational features such as objects and classes to represent each review displayed and sequentially work through each review.

Computational methods allow this problem to be solved as a physical method would not be efficient or simple to implement. This solution will involve large amounts of calculations and computations which would be impossible for humans to do alone. The repeated iterative process of reading reviews to determine their sentiment and report back to the user would be much more effectively done by a computer. Additionally, a computer follows a strict sequential nature and utilises fast processing speeds. This means that all of the calculations will be dealt with quickly and efficiently so that the user experiences little to no delay - something that would be very prevalent if solved by physical or human methods.

Currently, there are no non-computational methods that solve this solution due to the fact that Amazon is an online company and all of their services are online. The only two ways to determine sentiment are for a human to read reviews or create a machine that will read them and create links to decide. The former method is too laborious and so the latter would be the best solution to read through products with an excess of 1000 reviews.

Target Market

This product would be aimed towards shoppers who use Amazon. Therefore, the demographic of this product would be the same as all users of Amazon's services. This includes the 310 Million active users of Amazon who are looking to gain an insight into whether the product that they intend to buy is good quality and worth the money. Narrowing this target market down we can argue that it would be more useful for all shoppers who are aged 18 and above considering the fact that in order to buy an item online, you must be of this age. Users are likely to be technologically advanced due to the nature of buying online. Therefore, the system must be easily navigable and simple to use. This allows elderly users who may not be as familiar with technology to easily use the system.

Additionally, other users may include businesses who are looking to find out about their product and the overall sentiment about it. This will provide them with a view of the public's opinion without having to go out and survey. Being a free solution, the businesses would save money as they do not have to spend money on sending out surveys and handling with the incoming data.

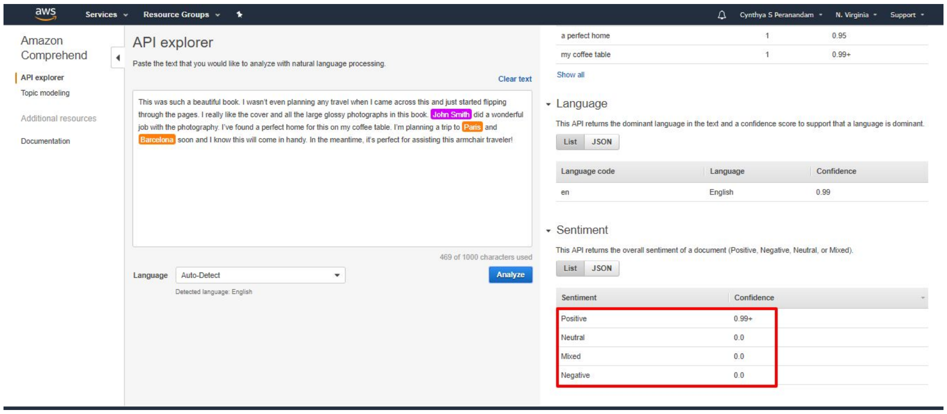
Research

Pre-existing Solutions

Currently, there are many researchers who have produced papers which carry out a very similar experiment in which they sample many Amazon reviews and attempt to evaluate the sentiment of the product whilst predicting whether the review is a highly rated or lowly rated product.

As such, I have aimed to research three different solutions that all encompass some form of my solution.

AWS Amazon Comprehend



API explorer

Paste the text that you would like to analyze with natural language processing.

This was such a beautiful book. I wasn't even planning any travel when I came across this and just started flipping through the pages. I really like the cover and all the large glossy photographs in this book. John Smith did a wonderful job with the photography. I've found a perfect home for this on my coffee table. I'm planning a trip to Paris and Barcelona soon and I know this will come in handy. In the meantime, it's perfect for assisting this armchair traveler!

469 of 1000 characters used

Language: Auto-Detect

Detected language: English

a perfect home	1	0.95
my coffee table	1	0.99+

Language

This API returns the dominant language in the text and a confidence score to support that a language is dominant.

Language code	Language	Confidence
en	English	0.99

Sentiment

This API returns the overall sentiment of a document (Positive, Negative, Neutral, or Mixed).

Sentiment	Confidence
Positive	0.99+
Neutral	0.0
Mixed	0.0
Negative	0.0

Amazon Comprehend[1] already is a very good solution to the problem at hand. It will provide a confidence for each of the different possible sentiments in a very clear and user friendly way (highlighted in red). The user inputs a review and the page will return different details such as the language, sentiment, entities such as places or people and keyphrases.

However, there are a few disadvantages to this service. Firstly, to use this service, you are required to pay in order to access Amazon Comprehend. Though it is free to create an AWS account, you must give your credit card details so that you can pay for the other services provided. For many, the added expenses of this will be enough to turn them away from using the service. As a result, I believe that my solution should be free and easily accessible to all. This is so that using it does not become a hassle and a greater expense than it is worth. Additionally, you can only analyse one review at a time in the example above and so it does not give an overall depiction of the quality of a product. Due to this, it is very unlikely that people will individually input each review and assess whether each one is positive.

Good Features	Missing/Bad Features
Various analytics create a strong analysis of each individual review.	Users have to analyse one review at a time.

More analyses than just sentiment.	Costs money.
	Unless you are a large company, the analytics are not useful.

Stanford Research Paper

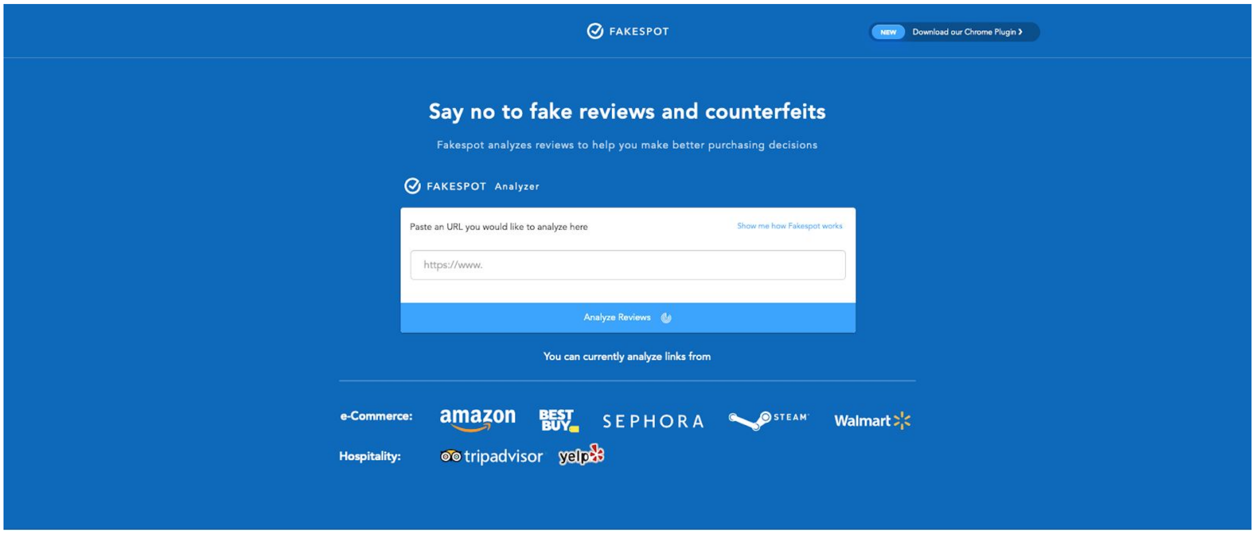
Another already existing solution to the problem is addressed in a paper published by researchers at Stanford[2]. Within this paper, the three researchers aimed to classify the positive and negative reviews of customers across a range of different products. They then built a supervised learning model to polarise the vast amount of reviews.

By testing a wide variety of different machine learning algorithms such as Naive Bayes analysis, K-Nearest Neighbours and a Recurrent Neural Network. In the end, they used 13 different models and gained various test accuracies between 52% to 71%.

To begin, the researchers took their dataset and preprocessed it by removing all the irrelevant fields such as reviews.dateAdded - representing the date the review was published to Amazon. An issue with their dataset was that it is heavily biased towards 5 star reviews. There were 23,775 4.8 - 5 star reviews compared to only 410 0 - 1.2 star reviews. Additionally, the researchers noticed that there were repeated samples in the dataset. This introduces more bias to the dataset and can therefore lead to model overfitting - an issue that they ended up experiencing.

Good Features	Missing/Bad Features
Various different models were tested for different accuracies and results.	Only a research paper - no tangible end product.
Utilises a large dataset of 34660 data points and so reliable models were more likely to be produced than from a small dataset.	Best model test accuracy: 71.5% Worst model test accuracy: 52.4% Overfitting was a large factor and so they produced low test accuracy. They have stated that this was due to data imbalance and a dataset that wasn't large enough.
	Heavily biased dataset.

Fakespot



Say no to fake reviews and counterfeits

Fakespot analyzes reviews to help you make better purchasing decisions

Paste an URL you would like to analyze here

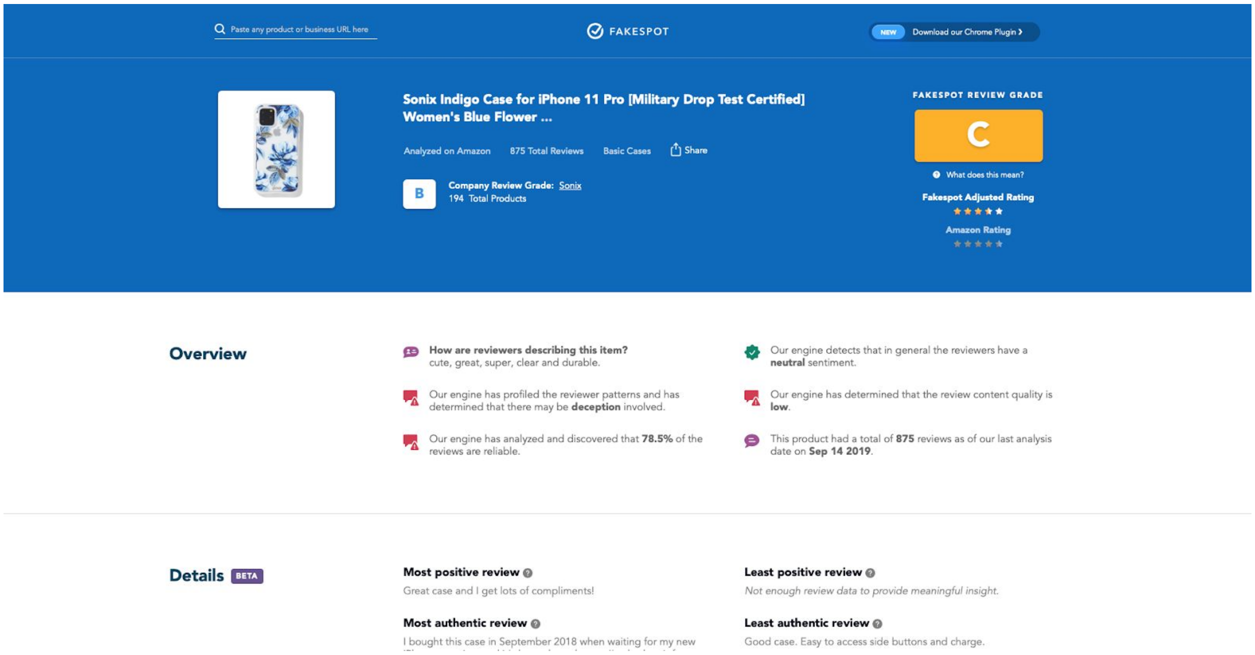
https://www.

You can currently analyze links from

Fakespot[3] is an online service that aims to provide consumers with a new way of filtering product reviews in order to find out what real users are saying about various products online. Their technology analyses millions of product reviews whilst looking for suspicious patterns and incentivised reviews. As a result, they then flag reviews that come up as unreliable.

The previous two solutions, though fit the end result better, lack in presentation and overall usability. This is what Fakespot makes up for; though it only looks at reviews for falsity and unreliability, it is presented in a much better format and analyses all the reviews of a product rather than individually.

The way in which the user inputs the URL of the amazon product that they are looking at is a very good and easy to use method and I believe this would be a good way to source the product from the user. As a result, validation will have to be carried out in order to ensure only Amazon products are used to begin with. Like Fakespot, it may be beneficial in the future to support the use of multiple different e-commerce sites.



Paste any product or business URL here

FAKESPOT

NEW Download our Chrome Plugin ›

Sonix Indigo Case for iPhone 11 Pro [Military Drop Test Certified] Women's Blue Flower ...

Analyzed on Amazon 875 Total Reviews Basic Cases Share

B **Company Review Grade:** Sonix
194 Total Products

FAKESPOT REVIEW GRADE

C

What does this mean?

Fakespot Adjusted Rating
★★★★★

Amazon Rating
★★★★★

Overview

How are reviewers describing this item?
cute, great, super, clear and durable.

Our engine has profiled the reviewer patterns and has determined that there may be **deception** involved.

Our engine has analyzed and discovered that **78.5%** of the reviews are reliable.

Our engine detects that in general the reviewers have a **neutral** sentiment.

Our engine has determined that the review content quality is **low**.

This product had a total of **875** reviews as of our last analysis date on **Sep 14 2019**.

Details BETA

Most positive review
Great case and I get lots of compliments!

Most authentic review
I bought this case in September 2018 when waiting for my new

Least positive review
Not enough review data to provide meaningful insight.

Least authentic review
Good case. Easy to access side buttons and charge.

Above is an example of the output of analysing the reviews for a specific product. As shown, the user is able to see a review grade for the product based on existing reviews and creates an adjusted rating to the one on Amazon. This is therefore likely to give the user a more holistic view on the product and allow them to make a better decision on whether they want to buy the product.

Good Features	Missing/Bad Features
Good, intuitive User Interface and User Experience.	Only looks at the reliability of reviews for products and not the quality or positivity of reviews.
Free to use with no additional payments for other features.	
Distinguishes between real and fake reviews and products.	

Evaluation

Looking at all of the pre-existing solutions, the clearest similarity is that they have different demographics but inherently use similar technologies. AWS Amazon Comprehend is aimed towards businesses looking to analyse their products and reviews on those products. The research paper from Stanford is intended as research so that it may be useful to others in the future. As such, there is no interaction between users and their solution. Finally, Fakespot is designed with individual buyers in mind. With it's simple User Interface and easy User Experience it outperforms the other two solutions though it lacks in the overall intention of the solution. By this, I am suggesting that Fakespot does not analyse reviews for whether they are positive or negative but for the falsity of reviews. It does have a section which displays the most positive but this is unhelpful as it has no gauge of the overall sentiment on the product from all of the reviews. Therefore, my solution will be aimed at all three types of demographics: individual users, businesses and researchers. By creating a useful and easy to manipulate web page, different users will be able to use it to accommodate their different needs. The solution will deliver on reporting if the product is good to the buyers and the businesses

Additionally, out of the three solutions presented, only the research paper provided a form of solution and how it was achieved. Both AWS Amazon Comprehend and Fakespot are proprietary and most likely profit from the service that they are dealing; as such they are unlikely to share the technology behind how they work. Due to this, my solution will aim to look and report like Fakespot whilst utilising the inner technology behind the research paper.

Another feature which is evident in both Fakespot and AWS Amazon Comprehend is the variety of different analytics. AWS Amazon Comprehend provides a variety of features in addition to the sentiment analysis such as entity detection and language. Comparatively, Fakespot will provide the details, helpful insights and similar products in addition to the sentiment analysis of the reviews. In order to most effectively help the user, multiple different facets of the information should be provided. In my solution I will begin with the basics and as the project develops I will integrate new features such as “Best Review” and “Worst Review”.

The Proposed Solution

Solution

Firstly, the solution will take inspiration from all three of the existing solutions so that it provides a service that is both complex computationally but simple for the user to interact with.

The user interface will take inspiration from the simplicity of the Fakespot website. This is because the intuitiveness of their solution is unparallelled and significantly better than the other two solutions. Though the website will be easy to use, I will implement a simple help button in order to aid those who are confused or in need of help. By making the results simple to manipulate and read, a reader range of people will be able to use the solution without difficulty. Within the web page, there will be a search box and submit button so that the user can search for various different Amazon products. Additionally, the product that they search for will be displayed on the screen so that they can both analyse the product themselves as well as read the systems analysis of the product and its reviews.

Next, the inner technology of the Stanford research paper will be utilised to analyse the sentiment of all the reviews for a specific product. This is something that the other two lack. A machine learning model will be responsible to analyse all incoming reviews and output a decision on the overall rating of the product. The model will be trained by a dataset that is created by scraping random reviews from random products on Amazon.

The development of the solution will be split up into various sequential sprints. By following the agile method, features can be segmented and developed individually so that they can eventually be integrated together into one seamless service.

The Sprints

Sprint 1

Sprint 1 will consist of creating the front end. I will use React and React Bootstrap in order to create an accessible and scalable web page that others can easily use with very little instruction. It should be intuitive and easy to use whilst being laid out in a readable format. This format and design will be explored further during the sprint with sketches of possible layouts.

Sprint 2

Sprint 2 will be the setting up of the web scraping in order to create a file for the machine learning model to train with. This will be done in Python and will either use an Amazon Reviews API, a pre-existing dataset or creating my own using a python package called BeautifulSoup.

Sprint 3

Sprint 3 will be the first iteration of the sentiment analysis. I will start with the simple implementation of natural language processing by creating a set of positive, negative, amplifier and negating words and assigning a score and multiplier to them. The overall score produced from a review will determine whether the review is positive or negative and an overall score will be averaged from all the reviews for a product. This will then create a rating for the product and return it to the user through the web page.

Sprint 4

Sprint 4 will aim to implement a form of machine learning into the sentiment analysis. This will include learning about the machine learning models such as the ones implemented in the Stanford research paper as well as developing my own to produce outputs that I require.

Sprint 5

Sprint 5 will be the implementation of additional features such as “Best Review” and “Worst Review”. Some potential features are listed below.

1	Best and Worst Review
2	Include number of reviews as a factor in analysis

3	Include seller location and seller as a factor in analysis (more trusted means better chance of a good product)
4	Similar products displayed
5	Similar product ratings displayed alongside
6	Training whilst deployed - users help to fine tune the system
7	Break up reviews into phrases and establish sentiments for those phrases
8	Account system to save products and reviews

Risks

Feasibility Study

Economic Aspect

The project will not cost anything to complete. This is because the hardware and software required are already available and open source. Some cost may incur from using an AWS account, however, I do not intend on opening a free AWS account and paying to test the sentiment analysis feature.

Technical Aspect

The project is possible and can be completed. Some training and learning will be required when web scraping as currently there are two possible routes: either finding an Amazon product review API or scraping them myself. I already have experience in Django, Python and React and so the main part of the training will be learning the models to use when carrying out the sentiment analysis.

Social Aspect

The solution will be very useful to buyers as they are the main stakeholders in this product. However, businesses may dislike the product as it might dissuade buyers from buying their products if they do not have an overall positive review. Businesses, however, can then use the product in order to alter the product and take into account the various buyer reviews to improve their score and sell more stock.

Time Constraints

Creating the front end should not be too time consuming as it will be a fairly straightforward webpage. Future features can be added at a later date after the main sentiment analysis feature has been implemented. The model will have to be trained as well which may take a long time once it has been developed. Having about 4-6 months to complete the project, I will focus on completing the first three sprints and then adding the additional features when the main priority has been successfully completed.

Limitations

The first limitation may be time management. Having around 4-6 months to develop a working solution, it is essential to focus on the most important features first before developing the additional, helper features. For example, it is necessary to implement the sentiment analysis but this can be done in a number of ways. The easiest method would be to create a database of positive, negative, amplifier and negating words and create an algorithm that can analyse the appearances of these words in reviews. Being a simple solution, this may be developed quickly and soon I would be able to produce a better machine learning model to analyse the reviews. However, having the existing solution already in place, there would be no issue of delays or producing a half-finished solution as we would be extending features once the base amount is complete.

Another limitation caused by time would be the time allowed for training the machine learning model and the availability of test data. It might be beneficial to find a dataset online similar to the Stanford research paper. However, their dataset was heavily biased and so this caused overfitting in their model. Alternatively, I could scrape the reviews myself and fill quotas. For example, we would create a file of 10,000 1 star reviews, 10,000 2 star reviews and so on. This way, there is no inherent bias from the number of samples. The issue here would be the time taken to generate the dataset which would take a considerable amount of time.

One issue that may arise is the actual effectiveness of the solution. As the machine learning models may be hard to research and properly implement and train, the results that the solution produces may not be accurate or reliable. The machine learning model aims to solve the issue of the existing solutions not providing an answer to if the user should buy the product or not and returning a rating.

Requirements

Software

In order to produce the solution, I will aim to work in a variety of languages. These will be Python[4] with the Django[5] framework for the backend whilst, for the frontend, React[6] will be used with React Bootstrap[7] to produce a clean user interface that is easy to interact with. A variety of different modules will be required. BeautifulSoup[8] will be required in order to carry out the web scraping for the amazon reviews and ratings. This will incorporate bootstrap in order to make the resulting webpage easy to resize and use on different sized devices. As both the frontend and backend are web-based and are hosted locally, the system will eventually be available on all devices and web browsers that have javascript enabled.

Hardware

The hardware required is very minimal. I will be using an Apple Mac in order to develop the solution. The Mac has a 2.9 GHz processor with 8GB RAM. Essentials such as a mouse and keyboard are required too. The documentation will be stored online on Google Drive whilst the solution will be stored locally on the Mac harddrive and can be transported using another medium such as a USB flash drive.

Success Criteria

Criteria Reference	Criteria	Reason	Test Strategy
1	**The User Interface**		
1.1	Display a URL search box and submit button.	Allow the user to search for a specific Amazon product.	There should be a text input box with a submit button.
1.2	Return the first image of the product and all relevant information.	Allow the user to read a summary of the product.	An image should appear as well as a card holding the product information.
1.3	Display a result.	The user must be able to see the results of the analysis of the reviews.	A calculated rating should appear as well as the confidence that the product is positive or negatively rated.
1.4	Restart search.	The user must be allowed to create multiple analyses for different products.	A restart button should be included or the search bar and submit button must always appear at the top of the page.
2	**Web Scraping**		
2.1	Take a URL from the user.	The scraper must be able to take all of the reviews from the product provided by the user.	The scraper will be able to find the page referenced by the user and eventually extract the relevant details.
2.2	Access the reviews.	The reviews must be taken from the Amazon site and then analysed by the software.	Sanitise the incoming reviews and store them locally. They can then be output to check existence.
2.3	Dataset creation.	In order to train and test the machine learning model, it will	One file will be created that includes a mixture of 50,000 reviews of

		have to analyse thousands of reviews to gain an overall view on what is positive and negative.	different stars. These will then be fed into the model to train and test it.
3	**Sentiment Analysis**		
3.1	Analyse an individual review for the words in it.	The system must be able to distinguish between good and bad reviews.	A score is returned for the review based on the occurring words within it.
3.2	Incoming reviews must be sanitised and prepared before analysis.	Words may not match if there is capitalisation or if they are misspelt.	The sanitisation function will be tested to ensure it is effectively working before use.
3.3	The machine learning model is trained	The model must be trained in order to effectively analyse reviews to an acceptable level of accuracy and confidence	After training, the system will be tested and it will output an accuracy. Once the accuracy is acceptable, it will be ready for use.
3.4	The machine learning model is tested	The model must then be tested to ensure acceptable accuracy and error.	An accuracy of 80% and above will be ideal but this will vary depending on the dataset created.
4	**Additional Features**		
4.1	Best and Worst Reviews are displayed.	These are displayed so that the user can gain more information about the product they may buy.	Create a fake product with fake reviews and submit that product. The most positive and negative reviews will be output and then compared to see whether they are working.
4.2	Number of reviews included in analysis.	More reviews normally highlights that the product is popular and	Create a coefficient that incorporates the number of reviews into

		better value	the overall outcome. Test this with very popular products and less popular ones.
4.3	Display similar products.	The user may look at an untrusted or unreliable product and so providing other ones may be beneficial to the user.	The system can search Amazon for the product tags and return the three most popular ones. Alternatively, the first page could all be analysed and the top three returned but this will add to computation time and increase delay.
4.4	Model is trained whilst in use.	Users can train the model whilst it is being used in order to improve accuracy.	Every request made by the user can then add to the training set and the next request can therefore test the model. As a result, the accuracy should steadily improve over time.

Sprint 1