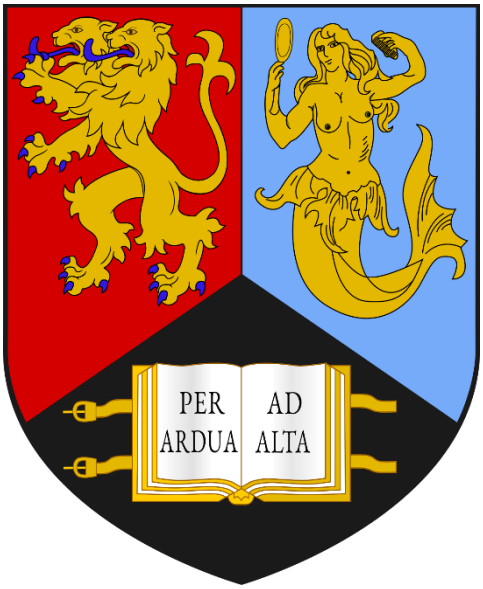
User Review Bias Detection In Metacritic



By Jonathan Bentham

Student ID: 2107241

MSc Computer Science (Conversion)

Supervisor: Dr Ian Kenny

School of Computer Science

Date: September 3rd 2020

Acknowledgments

I would like to thank Dr Ian Kenny for his guidance and advice throughout this project.

Abstract

The most controversial game of 2020, The Last of Us: Part II sparked disparity amongst critics and users around the world and this is evident on Metacritic, the review aggregator website. The game scored 94 out of 100 overall from critics – one of the most highly acclaimed games ever. However, the overall user score was 5.6 out of 10 denoting a low-average review score. This study was designed to check for bias levels of user reviews on the site. Two deep neural networks were created. The first model was created using reviews scraped from Metacritic, by adapting a web scraping algorithm. The second model was trained using reviews for Amazon fine foods. The model trained on Amazon data led to a higher degree of accuracy than the Metacritic model. As the Amazon model had the highest accuracy, a Long Short Term Memory (LSTM) layer was added. However, addition of the LSTM layer led to a lower accuracy of predictions, even though this took over six hours to run. An in-depth bias was carried out on three recent games: The Last of Us: Part II, Death Stranding and Ghost of Tsushima, where the predictions of bias on reviews were manually checked for accuracy. The manual check of the three games saw an average of 45 percent accuracy for the Amazon models predictions, which differed greatly from the 74 percent accuracy on training data. The in-depth analyses again showed that the Amazon model without an LSTM provided better predictions than the Metacritic model. The bias percentage was then checked on the best and worst 25 games on Metacritic by user score. The percentage of bias ranged from 4 to 57 percent on the Amazon model. The Metacritic model showed a bias range between 15 to 81 percent. The Evil Within 2 was found to have the largest percentage of bias overall. The average level of bias on the most accurate model was 25 percent for the worst 25 games and 41 percent for the top 25 games. Therefore, the disparity between user and critic review scores appears to be partially if not completely down to biased user reviews.

Contents

List of Figures

List of Tables

1 Introduction

The aim of this project was to detect the percentage of bias in user reviews of video games on Metacritic.com (Metacritic). Metacritic is an online review source which aggregates consumer and critic reviews of music, films, games, and books. There is now a body of evidence indicating that positive consumer reviews have an impact on sales. For example, a study by Ye, Law and Gu (2009) revealed a significant relationship between online consumer reviews and hotel room sales. Of relevance to this project were the findings of Zhu and Zang (2010) who found "online reviews are more influential for less popular games and for games whose players have greater internet experience".

Two Deep Neural Networks were created for this study, using two different datasets for a sentiment analysis. The first model was trained using an Amazon dataset reviewing fine foods on the Kaggle website. The other model was trained using data scraped from Metacritic using 22 games. Both models are used for a multi-class classification of three classes, to detect whether the written sentiment of a review is either: positive, negative, or average.

To scrape the Metacritic reviews, a problematic, current web scraping algorithm for the site was adapted. This was problematic as it would not scrape the data for the games of interest for this study. This was solved, and the algorithm was made to run significantly faster than previously. The web scraper was not only used to train the Metacritic model, but all predictions used in this study were carried out on user reviews from the site.

From the predictions a bias percentage was found for each game, by looking at how many predictions matched the actual review type. Therefore, if a game review was average and the sentiment predicted was positive, this was classed as a biased review. Three games were analysed in depth: The Last of Us: Part II (TLOU2), Death Stranding, and Ghost of Tsushima. From these in-depth analyses, the model which tended to make a more accurate bias detection was revealed. Additionally, it became apparent that occasionally it was worth using both models collectively to detect the level of bias for the game. The predictions were then carried out on a larger scale of the best and worst 25 games on PlayStation 4 (PS4) from Metacritic (by user score) with at least 100 written reviews.

The background of this report includes a brief overview of Metacritic and introduces the games that were collected to train the Metacritic model and the user reviews that were examined for bias. Sentiment analysis, word embeddings, Long Short Term Memory (LSTM) and the Rectified Linear Unit (ReLU) activation function are described in the context of this report. This is followed by a literature review of sentiment analysis and web scraping techniques. The methodology section explains the techniques used to train the models, the pre-processing technique employed and how predictions were made on the datasets. The results and analysis section describe the accuracy of models. A manual check of predictions was carried out within the in-depth bias analysis to check for the percentage of error within the models. The level of bias amongst the best and worst reviewed games on Metacritic is reported. The strengths and limitations of this report, as well as future work, are examined in the discussion. The conclusion of the study provides an overview of this study.

The following hypotheses were constructed to be later tested in the results and analysis section of the report:

Hypothesis One: The Metacritic model will predict bias to a higher level of accuracy than the Amazon model.

Hypothesis Two: Adding an LSTM layer to the models will lead to higher accuracy in predictions.

Hypothesis Three: The top 25 PlayStation 4 games by user score have a higher percentage of bias than the lowest 25 PlayStation 4 games on the most accurate model.

2 Background

2.1 Metacritic

Metacritic accumulates professional and user review scores from multiple online media review sources. The critic reviews are based on a score between 0 and 100, however this study looked at the Metacritic user score and only video games (rather than other media). The user score allows the regular users of a game to leave a written review and rate on a scale between 0 and 10, where:

- A score of 0 – 4 denotes a negative review
- A score of 5 – 7 denotes an average review
- A score of 8 – 10 denotes a positive review

It is important to note that each user score has equal weighting, therefore in-depth reviews that are paragraphs long carry the same weighting as scores that are a sentence long. These reviews are posted anonymously under a username of users' choosing. The study by Straat and Verhagen (2017) describes how Metacritic was mentioned in various game blogs and online blogs discussing the validity and value of professional reviews (Shreirer, 2015).

A variety of games were extracted from Metacritic. One of the most recent and most controversial games to be reviewed on the site was TLOU2. The game has sparked disparity between critics and users. TLOU2 scored 94 out of 100 as a cumulative average of 118 critics (a highly positive score), whereas 114,045 users gave a cumulative review score of 5.6 which denoted a low average review score. The user score for this had been previously lower, however the staff at Metacritic attempt to remove biased reviews during the first few weeks of a games release.

The work of Subhan (2020) suggests two potential solutions to the bias in the user reviews. The first solution was to develop a system to check that users had purchased the game, and the second was to completely abolish the user review system. The author found that the first did not completely remove review bias, and the second took away the opinion of the fair reviewer.

Therefore, a system to detect review bias seemed like a more logical option.

2.2 Video games within the study

Two models were created in the study to detect bias, one of which was trained on Metacritic data. Before extracting data, at least 5,000 average reviews needed to be extracted in total (average review numbers are significantly lower than positive and negative). Therefore, the following games were included:

Table 1 - Game reviews trained on Metacritic model

Title	Year of Release	Platform	User Score
The Last of Us: Part II	2020	PlayStation 4	5.6
Pokémon Sword	2019	Nintendo Switch	4.6
Modern Warfare	2019	PlayStation 4	3.3
Death Stranding	2019	PlayStation 4	7.3
Overwatch	2016	PC	6.5
Red Dead Redemption 2	2018	PlayStation 4	8.4
Minecraft	2011	PC	7.8
Star Wars Battlefront	2015	PlayStation 4	5.0
Call of Duty: Black Ops 3	2015	PlayStation 4	4.9
Assassin's Creed Odyssey	2018	PlayStation 4	6.3
No Man's Sky	2016	PlayStation 4	4.7
Halo 5	2015	Xbox One	6.4
Borderlands 3	2019	PC	5.2
Assassin's Creed Origins	2017	PlayStation 4	7.2
The Order: 1886	2015	PlayStation 4	6.7
Days Gone	2019	PlayStation 4	8.2
Grand Theft Auto V	2014	PlayStation 4	8.4
Batman: Arkham Knight	2015	PlayStation 4	7.9
Animal Crossing: New Horizons	2020	Nintendo Switch	5.4
The Last Guardian	2016	PlayStation 4	7.9
Resident Evil 3	2020	PlayStation 4	6.5
Gears of War 4	2016	Xbox One	7.0

The top 25 PS4 games by User Score (with over 100 written reviews) were tested using the models:

Table 2 – Top 25 games by user score for PS4

Title	Year of Release	User Score
Ghost of Tsushima	2020	9.2
The Witcher 3: Wild Hunt	2015	9.2
The Last of Us Remastered	2014	9.2
God of War	2018	9.1
SpongeBob SquarePants: Battle for Bikini Bottom – Rehydrated	2020	9.0
Resident Evil 2	2019	8.9
Astro Bot: Rescue Mission	2018	8.9
NieR: Automata	2017	8.9
Bloodborne	2015	8.9
Detroit: Become Human	2018	8.9
Dark Souls III	2016	8.8
Dreams	2020	8.7
Marvel's Spider-Man	2018	8.7
Dragon Quest XI	2018	8.7

Persona 5	2017	8.7
Life is Strange	2015	8.6
A Plague Tale: Innocence	2019	8.5
The Evil Within 2	2017	8.5
Titanfall 2	2016	8.5
Uncharted 4: A Thief's End	2016	8.5
Rachet & Clank	2016	8.5
Uncharted: The Nathan Drake Collection	2015	8.5
Rocket League	2015	8.5
Devil May Cry 5	2019	8.4
Red Dead Redemption 2	2018	8.4

The lowest rated 25 PS4 games by User Score (with over 100 written reviews) were tested using the models:

Table 3 - Worst 25 games by user score for PS4

Title	Year of Release	User Score
Madden NFL 21	2020	0.3
NBA 2K20	2019	1.2
FIFA 20	2019	1.2
Star Wars Battlefront II	2017	1.4
Metal Gear Survive	2018	1.4
Tony Hawk's Pro Skater 5	2015	1.5
Madden NFL 20	2019	1.5
WWE 2K20	2019	1.5
NBA 2K18	2017	1.7
FIFA 19	2018	1.9
Madden NFL 19	2018	2.1
Battlefield V	2018	2.3
NBA 2K19	2018	2.7
Fallout 76	2018	2.8
Call of Duty: Modern Warfare	2019	3.3
Anthem	2019	3.4
FIFA 18	2017	3.5
Street Fighter V	2016	3.6
Fortnite	2017	3.6
Mortal Kombat 11	2019	3.6
Call of Duty Ghosts	2013	3.9
Call of Duty: Infinite Warfare	2016	3.9
Call of Duty: Black Ops 4	2018	4.0
Far Cry: New Dawn	2019	4.1
Need for Speed: Payback	2017	4.2

2.3 Sentiment Analysis

Sentiment analysis is the area of study that analyses people's opinions, sentiments, evaluations, attitudes, and emotions from text. It essentially appears to be the most popular research area in Natural Language Processing (NLP), and the research has gone beyond computer science to areas such as social sciences due to its influence on society. Sentiment analysis is used in many businesses because opinions are the main reasons for the way humans behave as consumers (Liu, 2012).

There are many types of sentiment analysis. It can be used to detect polarity (this study), but it can also detect feelings, emotions, or even intentions. Rather than just looking at positive and negative sentiment, this study also classifies average comments to capture a greater range of review types.

The three main types of algorithm used for a sentiment analysis are rule-based, automatic or a hybrid of the two. In this study an automatic approach is used, which relies on machine learning techniques. The below figure shows how a sentiment analysis with a machine learning approach works:

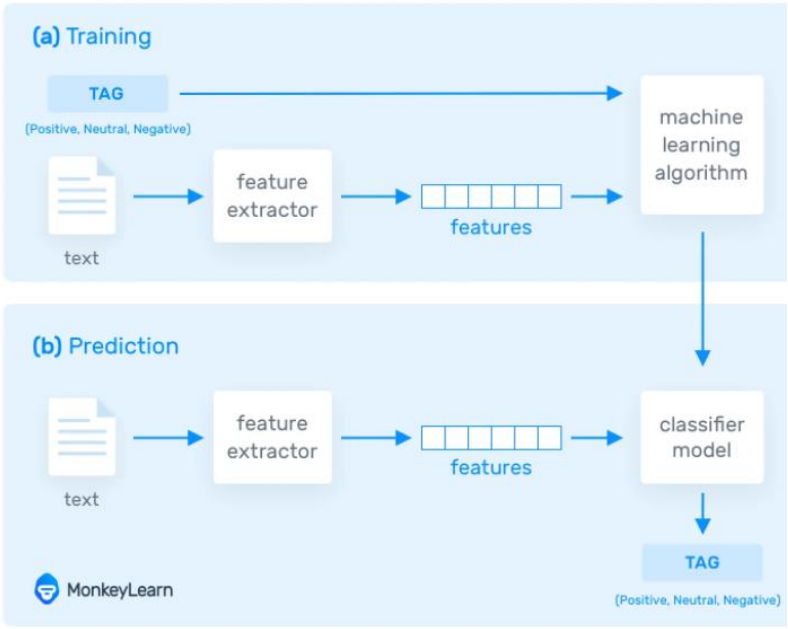


Figure 14 - How does sentiment analysis work? (MonkeyLearn, 2020)

The above figure shows the training process in part (a), where the model learns to associate the text input with corresponding output (i.e. positive, negative, average - tags). The feature extractor embeds the words into feature vectors (description in Literature Review). The tags are then paired with the embedding vectors and a machine learning algorithm is used on the vectors to create a model. In the prediction process in part (b), the feature extractor is used to transform unseen text into vectors of their features, which are passed via the same model to generate tags.

2.4 Word Embeddings

For this report, the sentiment of reviews (positive, negative, or average) were detected. To predict sentiment by means of machine learning, the text needed to be converted into dense vectors (i.e. a vector with mainly non-zero values) so that a computer can understand the meaning of the word. One-hot encoding was applied to create tags for the models which changed the review type to numbers, leaving vectors with a high dimensionality. The problem with one-hot encoding is that it thinks similar words are completely different features, however word embeddings lower the dimensionality and "the semantic relationships between words are reflected in the distance and direction of the vectors", as the work of Carremans (2018) shows.

Rather than doing a large amount of pre-processing, where it is necessary to tokenize, stem and remove punctuation (i.e. the steps before you convert text into numbers), it is now possible to use pre-trained models that have already completed pre-processing. The pre-trained model used for this report was the Universal Sentence Encoder (USE) (Cer et al., 2018), for reasons discussed in the Literature Review. Each sentence that is passed to the model was encoded as a vector with 512 elements.

2.5 Long Short Term Memory Networks

Long Short Term Memory networks (LSTMs) are a type of Recurrent Neural Network (RNN), that are capable of learning long-term dependencies. The work of Hochreiter and Schmidhuber (1997) first introduced LSTMs, and their work has been refined since this time. The initial idea behind their development was to solve the long-term dependency problem that RNNs have, where information is remembered for long periods of time. The work of Olah (2015) shows the difference between a standard RNN and an LSTM. Below shows a standard RRN:

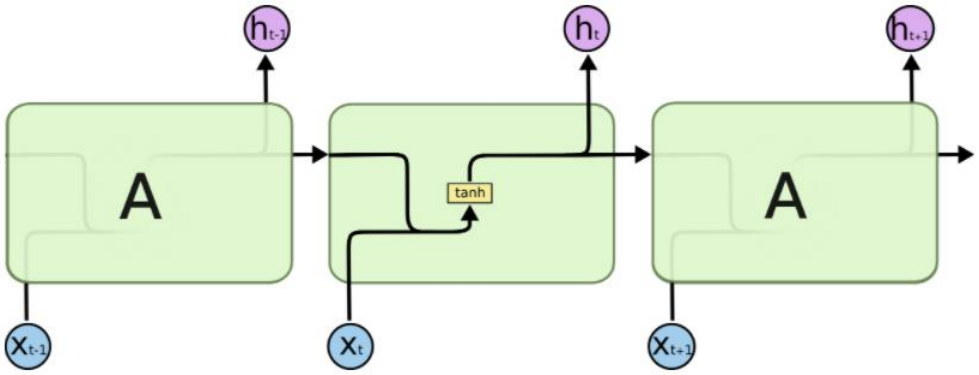


Figure 15 - Standard RNN (Olah, 2015)

The figure above shows how all RNNs have a chain of repeating modules of neural network. In the standard RNN an example of a single tanh layer is shown, indicating a very simple architecture compared to the RNN with an LSTM layer below:

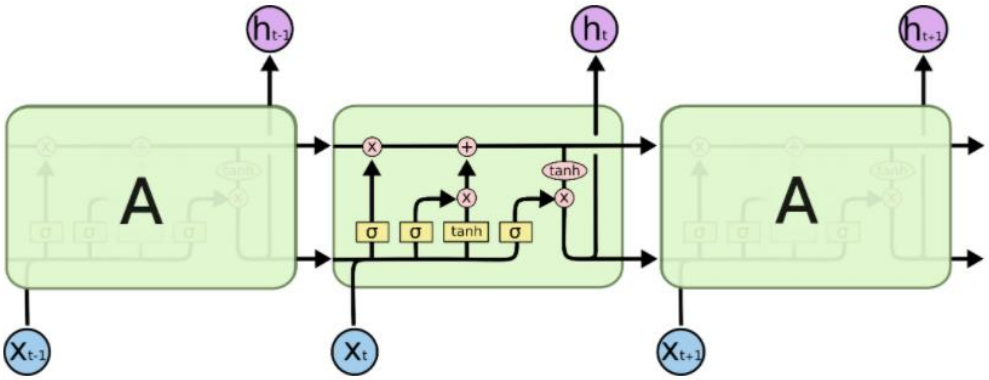


Figure 16 - RNN with LSTM layer (Olah, 2015)

The chain layer in the above figure is more complex than without an LSTM because the repeating module has a more complicated structure. Instead of one layer, there are four. The findings of Olah (2015) show that having an LSTM layer works better for most tasks. From the figure above, in the LSTM layer there are inner recurrent activations which actualize inner cell memory – which is controlled by the recurrent activation function; this is usually a sigmoid function. The final output layer is then computed by applying the activation function, in the above case by default is tanh.

2.6 The ReLU activation function

It is important to choose activation functions carefully with deep neural networks as this has a great impact on the performance of the model. The most used activation function is the ReLU. There are many other activation functions such as Sigmoid, however none of them have been able to outperform ReLU (Cer et al., 2018). The work of Wang (2019) shows that the vanishing gradient can be solved by using the ReLU activation function, whereas functions such as Sigmoid do nothing to solve this. The vanishing gradient problem needs to be solved so that the accuracy and loss of the models are the best they can possibly be.

More detail can be seen on why the ReLU activation function is used when looking at the graph of the function. See the figure below.

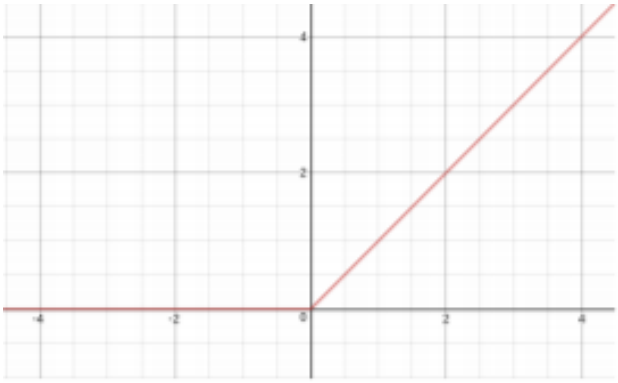


Figure 17 - ReLU activation function (Agarap, 2018):

Firstly, the definition of the ReLU activation is as follows:

$$h = \max(0, a) \ \ where\ a = Wx + b \qquad (1)$$

The reason for the lower likelihood of the gradient vanishing is due to when a > 0; in this instance the gradient will have constant value. When comparing this to a sigmoid activation the gradient becomes increasingly lower as the absolute value of x grows. Therefore, the constant gradient of ReLUs lead to faster learning than Sigmoid. The second benefit of ReLUs is sparsity – most of the weights are zero, leading usually to an increase in space and time efficiency.

3 Literature Review

3.1 Sentiment Analysis

A systematic literature review of sentiment analysis by Kumar and Jaiswal (2019) on Twitter using Soft Computing (SC) techniques, showed that SC techniques are generally split into five categories: Machine Learning (ML), Neural Networks, Evolutionary Computation, Fuzzy Logic and Probabilistic Reasoning. Another systematic survey by Jain and Singh (2018) was more general and not specific to SC techniques. Their study found issues and challenges across recent research work including: most work carried out on sentiment analysis is on English text data; sarcastic text struggles to be detected; performance of sentiment analysis suffers from fake or spam reviews; and poor grammar and punctuation can lead to incorrect classification.

3.1.1 Binary Sentiment Classification

The sentiment analysis performed for this report was influenced by the work of Velkov (2019), which used hotel review data scraped from Booking.com. The dataset contains 515,000 customer reviews stored in a csv file. The file contains review scores from 0 to 10 with their corresponding text review. The data is read into python using the 'pandas' library and the dataframe is given a new column with the reviews in corresponding review type, whether it is positive or negative based on a lambda expression.

As there were a large number more positive than negative reviews, the author created two separate dataframes with the same number of positive and negative reviews and appended them to one dataframe. Doing this led to the same number of positive and negative reviews. This is a good idea as there was no bias when training the model, i.e. the model would be more likely to predict a positive review if the model is trained on significantly more positive than negative reviews. This reduces the number of reviews from over 500,000 to only 160,000, however this is still a large amount of data to train models on.

The data was then encoded using one-hot encoding; therefore, each review is represented by a NumPy array of two elements which consist of a one and a zero. The one in the first position indicates a negative review. The 'train_test_split' function is then used from sklearn, which enables the author to split the reviews into training and test datasets. Additionally, the author decided to split the data to test ten percent of the reviews. The author then goes onto embed the training data and the test data by using the previously mentioned third iteration of the USE. This then converts the training data into 156,331 vectors of length 512, and the output of the training data has the same number of elements, but they have been converted using one-hot encoding.

The sentiment analysis itself is then carried out using Keras, and has the following network architecture:

1) Input layer: Start with the fully connected layer, with 256 neurons, input shape is the same as X_train, and the Rectified Linear Unit (ReLU) activation function is used.
2) 2nd layer: Dropout layer with 0.5 dropout rate

3) 3rd layer: Same as input layer but has half the neurons.
4) 4th layer: Dropout layer with 0.5 dropout rate
5) Output layer: Outputs two neurons for positive and negative, and a 'softmax' activation function is used.

The model strangely does not use 'binary crossentropy' for loss function, but uses a 'categorical crossentropy', which should be used when classifying more than two classes. After applying an adaptive learning rate of 0.001 the model is then ready to fit. The author uses ten epochs and uses ten percent of the training data as validation data. The training and testing datasets are shuffled when inputted into the network and a batch size of 16 is used. The model gets 82 percent accuracy on the test set and on the validation set, which is very good. From this, predictions can be made from the model and it can be compared against the actual review type.

3.1.2 Aspect Based Sentiment Analysis

The work of Straat and Verhagen (2017) looked at user created game reviews for sentiment analysis to look at user attitudes towards video games on Metacritic. The hypothesis tested was: "There is no relationship between the values of character, combat or story and the overall review rating". The study collected user reviews from two video game franchises and the most used words in the reviews were derived. The words in this instance, called aspects, were analysed through a sentiment analysis. The aspects were determined through a word frequency analysis of all user reviews, and the study found that the most common aspects were 'combat', 'story' and 'character'.

An aspect-based sentiment analysis was performed when user sentiment of certain aspects of a "multi-aspect entity" were to be measured. Here the author stated that video games had multiple aspects. After the aspects were determined the sentiment analysis was performed through an online crowdsourcing service, via a manual read through.

The output of the sentiment analysis then showed whether the sentiment is positive, average, or neutral, as within this study. The manual sentiment analysis was quite literally showing an evaluator the written review that contains one of the aspects and then getting them to state whether it is a positive, negative, or average one. Using a manual based approach for the present study is not an option due to the large datasets.

The results of the Straat and Verhagen (2017) study showed that if an aspect occurs in a review, the sentiment of that aspect will reflect the rating of the review. The null hypothesis was therefore rejected for all games that were examined in the study, other than two which had small datasets. These results imply that the aspects reflect areas in games that are disliked by users. Additionally, the high frequency of certain aspects implies that these areas are most important to users.

3.2 Web Scraping

The work of Ong (2019) as seen in Appendix B, built a simplistic guide to scraping Metacritic game reviews using the package BeautifulSoup (BS) in Python. These reviews were then saved to a dataframe and Pokémon Sword reviews were scraped. After importing the packages, the author figured out the web page's HTML structure and found that the written text in the reviews is concealed within a span tag. The reason for using the package BS is that it can find reviews within the span tags. However, the issue is that not all span tags contain written reviews, thus the author went on to tell BS to "find an outer tag that is review-specific and then find a span tag within". A scroll through the page's HTML, showed that the text is nested within 'div' classes. The author proceeded to parse the page using BS by making a URL request and then parsing the response into BS.

For the current study just the rating and review were needed, however this scraping also extracts the name and date which is useful when trying to detect user spam.

Ong (2019) proceeded to instruct BS to find the relevant information and append them to the dictionary's list, which included: name, date, rating, and review. To do this the author used 'find all' to get all 'review_content' tags within the page. This finds every review on the page. An 'if-else' statement was used in the code, where the else was directed at normal-sized reviews, and the 'if' was used when the review was of such a length, the user needs to click to see the full review. Additionally, as each review on Metacritic has the same elements, they are appended to the desired list. Following this the reviews are appended to a dataframe; however, this required a for loop for the number of pages of written reviews.

4 Methodology

A sentiment analysis was carried about by using Amazon reviews for one model and Metacritic reviews for the other model. All code in this project is coded in Google Colab using Python. Within Python, libraries such as Keras and sklearn were used to create the deep neural networks, as well as using BeautifulSoup to extract the Metacritic data.

4.1 Data Collection

4.1.1 Metacritic Model

During the Literature Review for this report, a web scraper for Metacritic was found. However, a problem was found when trying to scrape reviews for TLOU2. There were over 500 pages of reviews for this dataset, far more than the 23 pages of the Pokémon Shield reviews extracted by Ong (2019). The code was first run on TLOU2 for 20 pages and a 'NoneType' error was given (see Appendix C7, section: Test Using Old vs New Scraper), where the "object has no attribute 'text'". After going to the web page for Metacritic and analysing a few of the review pages, the issue was that the site allows reviews without written text. An 'if-else' statement was added to the code so that once the scraper was inside a review it simply moves to the next body of code.

The second issue with code was that it scraped data slowly. In the case of extracting 300 pages of reviews from TLOU2, it took approximately nine minutes. To solve this issue threads were added to the code. This was done by firstly building a chunkIt method to break the review into the desired number of threads (for details again see Appendix C7). By allowing threads, this enabled the data to be extracted over three times faster. For example, extracting 300 pages of reviews only took two minutes eleven seconds. Thus, saving almost seven minutes.

These changes led to a working, fast, Metacritic web scraping algorithm that was used to collect data for the Metacritic model and data that was used for testing models. Once the data had been converted to a dataframe, it was extracted to a csv.

The first model added to the dataset was TLOU2 which had 31,309 reviews. The issue was that this only had 1174 average reviews and each category needs to be the same length when training to reduce bias in predictions. Therefore, a further 21 games were collected and added to the dataframe so that upon reducing the length of the dataframe there would be over 5,000 average reviews. Upon combining each of the 22 csvs (discussed in the background) to a dataframe, there were 67,282 reviews. The data consisted of reviews from a period of nine years (November 2011 – July 2020).

4.1.2 Amazon Model

The dataset used for the Amazon Model consisted of reviews of fine foods from Amazon (this was taken from Kaggle, see Appendix A). The data had reviews from a period of over ten years (October 1999 – October 2012). Additionally, the dataset consisted of 568,454 reviews from 74,258 products, therefore there was much more data here than the Metacritic model.

4.2 Pre-processing

Pre-processing for the Metacritic and Amazon models was similar; however, it differed in parts. Once the data for both models were in csv format, there was an immediate issue with the reviews that needed to be solved; the reviews were multilingual. The initial plan was to convert every review in the csv to English by cycling through a dataframe using the Google Translate API. However, to do this with 100,000s of reviews in multiple languages led to a HTTP error of "Too Many Requests".

This would usually be solved by making the translation slow, by using a for loop with sleep cycles. However, the API only allowed so many translations in a time frame. The solution to this was to open Google Sheets for every csv and create a new column that used the following formula for each review in the dataframe:

=GOOGLETRANSLATE(C2, "AUTO", "EN")

The following formula translated the review from any language that Google Translate recognised and translated the review to English. After going through each spreadsheet and making sure that it was applied to every review in the spreadsheet, the old review column could be deleted, and this new column could be named 'review'. The reviews were then imported to different notebooks according to the model (Amazon Model see Appendix C1, Metacritic Model see Appendix C2).

Once the Metacritic data was loaded into a notebook the reviews were separated into three classes: Negative, Average, Positive. The three classes were based on the Metacritic website (see Introduction). These categories were then put into a review type column in the dataframe for each of the reviews. The Amazon reviews were on a scale of one to five. Therefore, this was normalised so that scores of one and two are negative scores, scores of three are average, and scores of four and five are positive.

Upon doing this the number of positive, average, and negative reviews were counted and unsurprisingly the number of average reviews was the lowest in both notebooks with 5,290 average reviews for Metacritic and 42,640 for Amazon. For both notebooks three different dataframes for the review types were created and then appended to one dataframe (adapted from the work in the Literature Review). Countplots were created to make sure there were the

same number of reviews of each type in the new dataframe. The figure below shows the Countplot for the Amazon notebook:

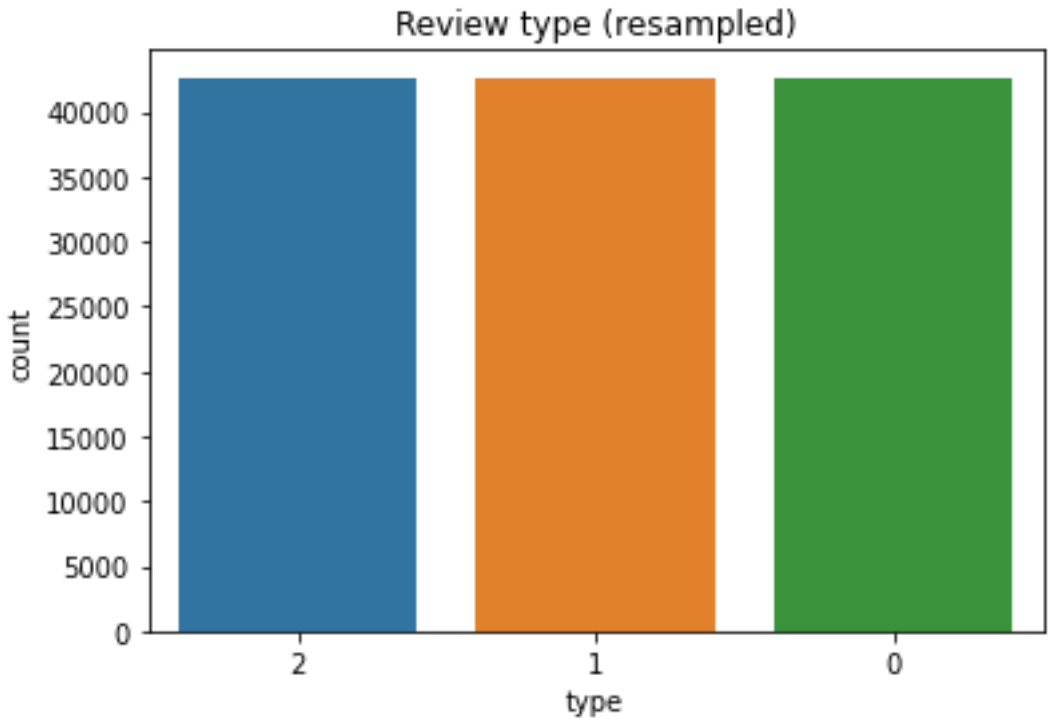


Figure 18 - Amazon review dataframe counts

The review types for Metacritic were one-hot encoded and were in a form so that the reviews could be embedded. Further, the 'train_test_split' function was used in both notebooks so that there was training data and test data. Both notebooks chose to have a test size of ten percent for their models. The embeddings for 'X_train' and 'X_test', were embedded in a similar way to the Literature Review, however, the newer iteration of the USE required different syntax to generate the embeddings. The embeddings were saved for the Amazon notebook because there were over 100,000 reviews to be embedded. However, this was not necessary with the Metacritic notebook due to the time low time to embed.

4.3 The Models

Both models used the same network architecture, however certain parameters vary when fitting the models to attain a better accuracy. Prior to building a neural network, it was important to choose which library to use between TensorFlow and Keras. Keras is built on top of TensorFlow and allows for faster network building due to the powerful 'Model' and 'Sequential' API's. Additionally, Python is far more user friendly than TensorFlow. The only downside was that Keras has slightly less functionality, but this was not an issue for the network being built.

Below shows the model architecture for the Metacritic and Amazon models:

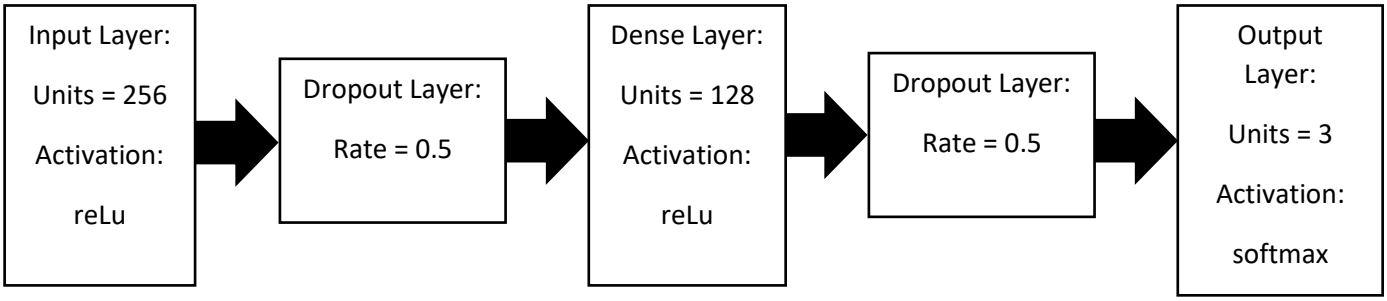


Figure 19 - Visualisation of Metacritic and Amazon Models

Layer 1: An input layer with an input shape matching the shape of the USE, therefore shape: (512,) and with 256 neurons (half the size of the input), with a ReLU activation function.

Layer 2: A 256-layer dense network which takes in the input from layer 1, and a dropout of 0.5 is added here to prevent overfitting.

Layer 3: A 128-layer dense network (halves the neurons), with a ReLU activation function.

Layer 4: A 128-layer dense network which takes in the input from layer 3, and a dropout of 0.5 is added here to prevent overfitting.

Layer 5: A three-layer dense network with softmax activation. Each class if used to represent a sentiment category, with class one representing negative, class two representing average, and class three representing positive.

The ReLU activation function has been used here because it is non-linear and helps complex relationships in the reviews to be captured.

An 'Adam' optimiser was used on both models, with a lower learning rate on the Metacritic model as it needed to spend more time learning the data as there was far less data being trained. As there were three classes, this was a multi-class classification (more than two exclusive classes) and thus a categorical crossentropy loss function was used.

When fitting the models, they both used early stopping to ensure that the validation loss of the model did not rise too much, if it did the training process would stop. Therefore, the number of epochs inputted was a random high number for each, as the model would stop running long before getting to these epochs.

The two models differed firstly on the validation data split, where only ten percent of the Amazon model was used for validation, against 20 percent for the Metacritic model. The reason for the difference in size of these splits was because the Amazon model only had 14,283 reviews for training, thus it was better to increase the validation size. If the size was not increased the model had more of a chance of reusing the same batches of data for validation during an epoch. Lower batch sizes led to better accuracy and better loss but require longer to run (Shen, 2018). Therefore, the Metacritic model was given a smaller batch size of 16, whereas Amazon model had batch sizes of 32. The aim was to try and get similar running times when fitting both models.

4.4 Amazon Model with LSTM Layer

An additional model was created for Amazon for reasons discussed in the Results section. However, prior to doing this, the shape of the training and testing embeddings must match. Therefore, the embeddings were reshaped in the Amazon with LSTM notebook (see Appendix C7). Below shows the network architecture for this Amazon model:

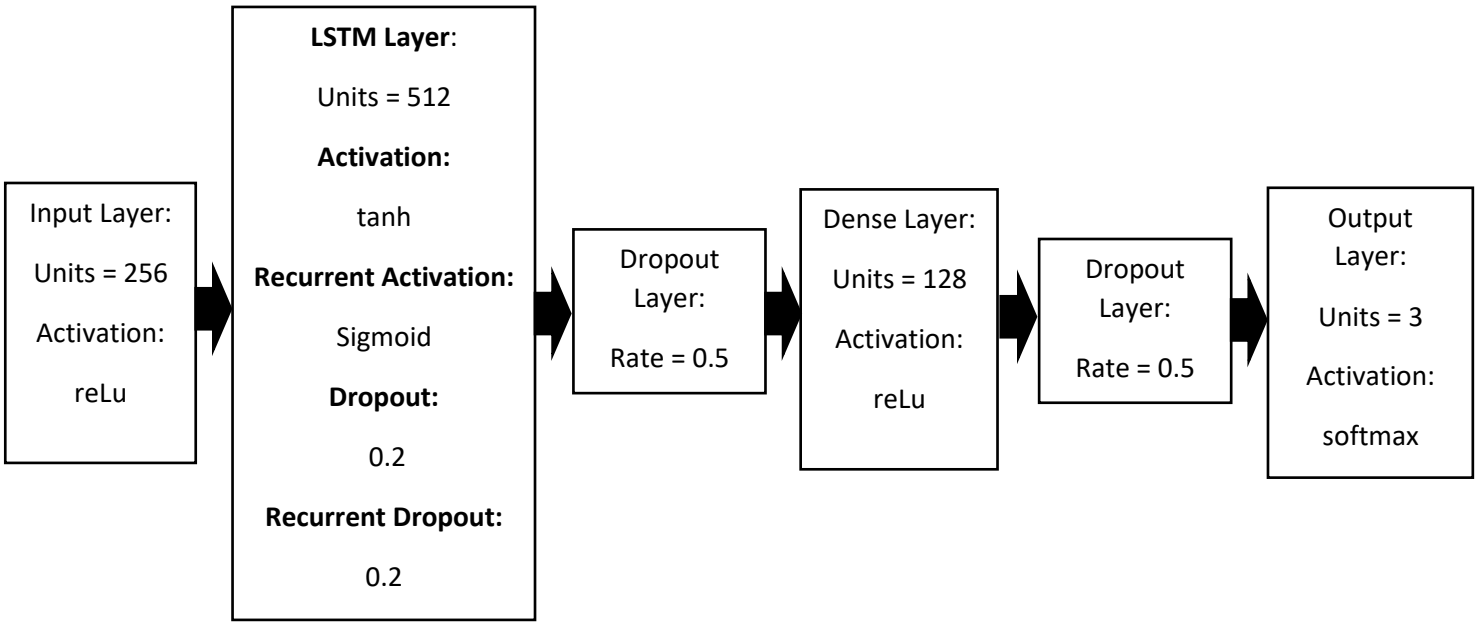


Figure 20 - Amazon model with LSTM layer

The idea of the additional model was to look at how the addition of an LSTM layer would affect the accuracy of the model. The model is identical to the first Amazon model other than the input shape of the first layer that has been changed so that it fits the shape of the new embeddings. There is also a new layer after the input layer, the LSTM layer. This layer outputs 512 neurons (the length of vectors in the USE).

The activation function used was 'tanh' (the default activation function in Keras). This can be used to overcome the vanishing gradient problem (discussed in the Literature Review). The reason for this was because the second derivative of the 'tanh' function can sustain for a long range before it gets to zero. Secondly, for the recurrent activation (the activation that happens inside the LSTM layer, rather than the activation on the output) a sigmoid activation function was used. As a Sigmoid function either outputs 1 or 0, in this case it could be used to forget or remember information.

4.5 Making Predictions

There were three games where bias detection of the model was analysed in-depth:

- The Last of Us: Part II (see Appendix C4 & C8) – Metacritic and Amazon model with and without LSTM layer
- Death Stranding (see Appendix C5) – Metacritic and Amazon model
- Ghost of Tsushima (see Appendix C6) – Metacritic and Amazon model

All predictions were carried out in the same way, other than the bias detection made on TLOU2 using the Amazon LSTM model. The reviews were firstly loaded in and this time, they did not need to be put into a smaller dataframe as every bit of data was being used to make predictions. However, this meant that the 'train_test_split' function did not automatically reshape data to go into the embedder used when training the models. Therefore, the data was manually reshaped so that it could go into the same embedder. In the case of TLOU2, this involved changing the shape from (31309, 1) to (31309,), which allowed the data to be embedded. Once TLOU2 reviews were embedded they had shape: (31309, 512). For three out of four notebooks the shape was fine, however in the same way as when embedding the Amazon LSTM model, the reviews for the LSTM notebook needed to be embedded to shape: (31309, 512, 1).

The Amazon and Metacritic models were then loaded. For each model a for loop was created to append to a list the prediction of positive, negative, or average based on whether the value in that position was less than or greater than 0.33. The lists were then appended to columns in the dataframe holding the reviews, ratings, review type. Therefore, the new dataframe now had the level of bias predicted from the model for Amazon and Metacritic. The reviews could now be used to detect whether there was bias by checking whether the actual review score matched the prediction made by the models. Separate dataframes were created for where only the Amazon model detected bias, only the Metacritic Model detected bias and where they both detected biases. Likewise, in a separate notebook the same was done for the Amazon model with an LSTM layer. The models were examined for where they correctly detected bias by selecting a random starting element in the dataframe and manually checking twenty adjacent reviews in the dataframe for where bias is detected in both models together and separately.

There are a further 25 games (see Background) that were scraped for bias detection on the Metacritic and Amazon models. The games were collected from a period of five years (2015 – 2020). The level of bias was detected in the same way as for the in-depth bias review, however for each game, the level of bias for the Amazon and Metacritic models was shown for each game in tabular format, as well as pie charts within the notebook (see Appendix C7 for pie chart code). The Amazon model with an LSTM layer is not used here for reasons discussed in the results of the study.

5 Results and Analysis

5.1 Metacritic Model

The Metacritic model ran for 82 epochs before stopping, due to the validation loss starting to increase and thus being stopped by the early stopping implemented. Each epoch ran for two seconds before moving onto a new batch of data in the group. By the 82nd epoch, the model produced the following accuracy and loss values for the training and validation data:

Training Loss: 0.7741 – **Training Accuracy:** 0.6898 (From 11426 reviews)

Validation Loss: 0.7676 – **Validation Accuracy:** 0.6818 (From 2857 reviews)

Below the graphs of the accuracy and loss for training and validation data for the Metacritic model have been plotted:

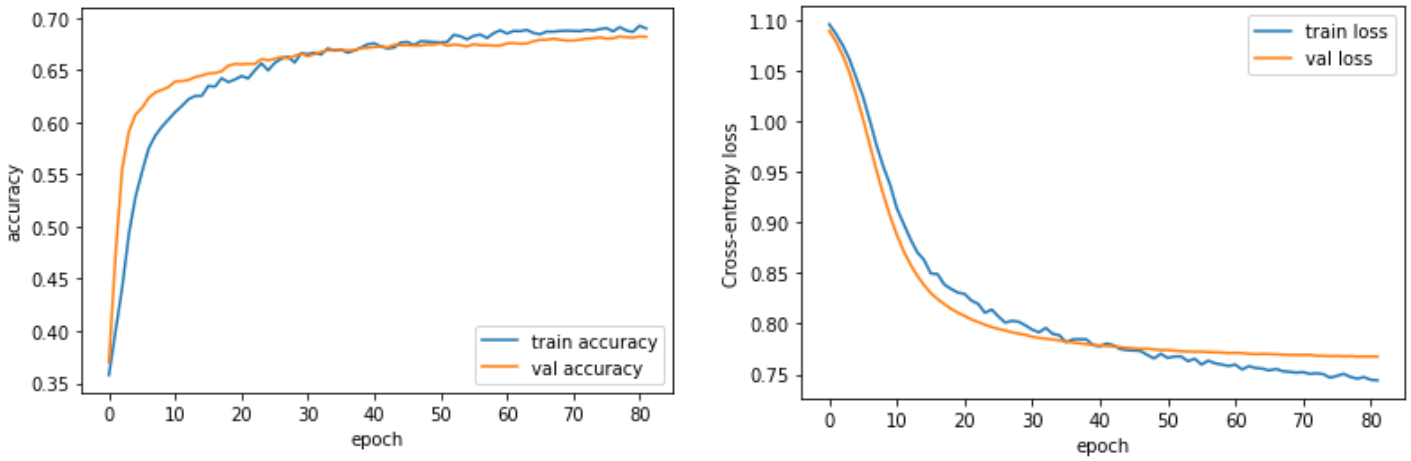


Figure 21 - Accuracy and loss for training and validation data (Metacritic model)

In both plots the model has comparable performance on both the training and validation datasets. There is no sign that in either of the plots that the lines are starting to depart consistently. Therefore, the training and validation sets are similar, even though they are taken from different parts of the dataset. If the two lines on either of the graphs were separating it could be a sign of overfitting. The low level of bias in the model can be seen further when looking at the 68 percent accuracy the model achieves on the test data on 50 reviews.

5.2 Amazon Model

The Amazon model ran for 27 epochs before stopping, due to the validation loss starting to increase. Each epoch ran for approximately 15 seconds before moving onto a new batch of data in the group. By the 27th epoch, the model produced the following accuracy and loss values for the training and validation data:

Training Loss: 0.5230 – **Training Accuracy:** 0.7850 (From 113976 reviews)

Validation Loss: 0.6291 – **Validation Accuracy:** 0.7391 (From 1152 reviews)

Below the graphs of the accuracy and loss on training and validation data for the Amazon model have been plotted:

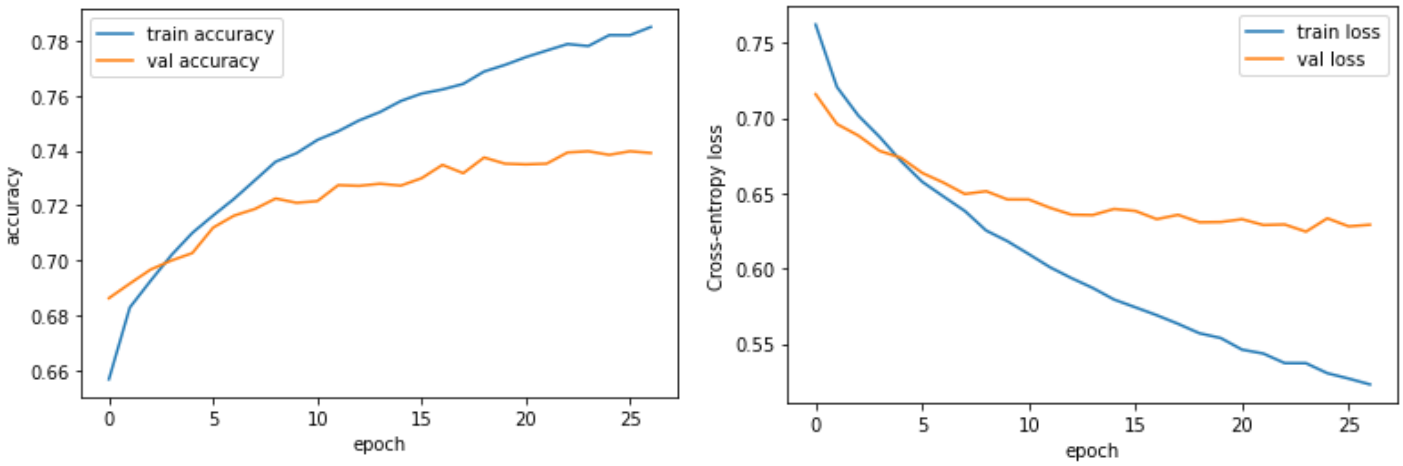


Figure 22 - Accuracy and loss for training and validation data (Amazon Model)

The above shows that the training and validation data differ, due to the lines not decaying or growing at the same rate. For example, the training loss is much higher than the validation loss. One of the reasons for the difference is because the epochs only stopped running once the validation loss started to increase. The issue here is the validation loss for the model falls slower than the training loss, so the model does not stop fitting. However, the model achieved an accuracy of 74 percent on the testing data. There was a higher level of variance between the train, validation, and test accuracy on the Amazon model than the Metacritic model. The accuracy of the training data on the Amazon model was significantly higher.

5.2.1 Addition of LSTM Layer

The Amazon model with an LSTM layer ran for 34 epochs before stopping. Following this, it started to increase. Each epoch ran for approximately 11 minutes before moving onto a new batch of data, which led to the model running for almost seven hours over 34 epochs. By the 34th epoch, the model produced the following accuracy and loss values on the training and validation data:

Training Loss: 0.6870 – **Training Accuracy:** 0.7014 (From 113976 reviews)

Validation Loss: 0.8175 – **Validation Accuracy:** 0.6556 (From 1152 reviews)

Below the graphs of the accuracy and loss for training and validation data for the Metacritic model have been plotted:

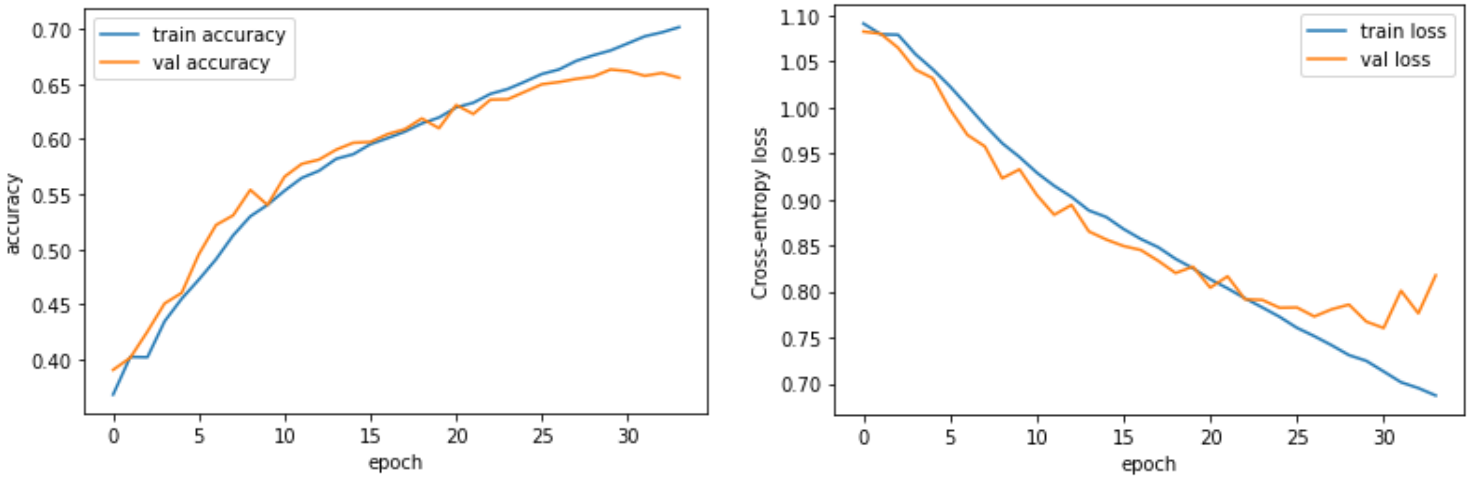


Figure 23 - Accuracy and loss for training and validation data (Amazon Model with LSTM layer)

In both plots the model has comparable performance on both the train and validation datasets. The graphs look better here than for the model without the LSTM layer because the training and validation lines tend to agree with each other. However, the spike at the end still leads to variance between the training and validation loss. Additionally, the testing data is the same as the validation dataset for validation accuracy and loss.

5.3 In-depth Bias Analysis

5.3.1 The Last of Us: Part II

For TLOU2, all 31,309 of the reviews were used to make predictions. The dataset had 15,722 positive reviews, 14,413 negative and 1,174 average reviews. Word clouds were created for each review type.

Below shows the WordCloud for negative reviews:

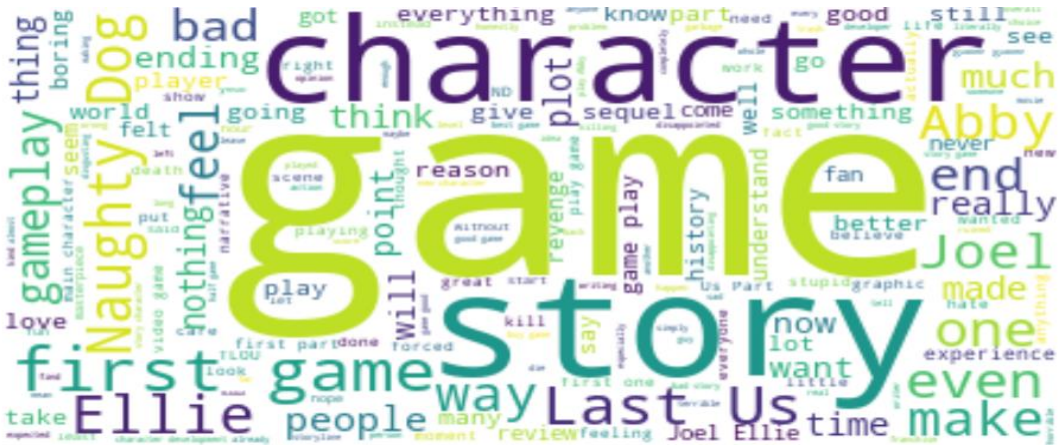


Figure 24 - TLOU2 negative review WordCloud

The above figure shows the most used words in negative reviews of TLOU2 reviews. The larger words indicate words that are used the most. From the above there are many users that associate terms such as ‘story’ and ‘game’ with negative reviews.

Below shows the WordCloud average reviews:

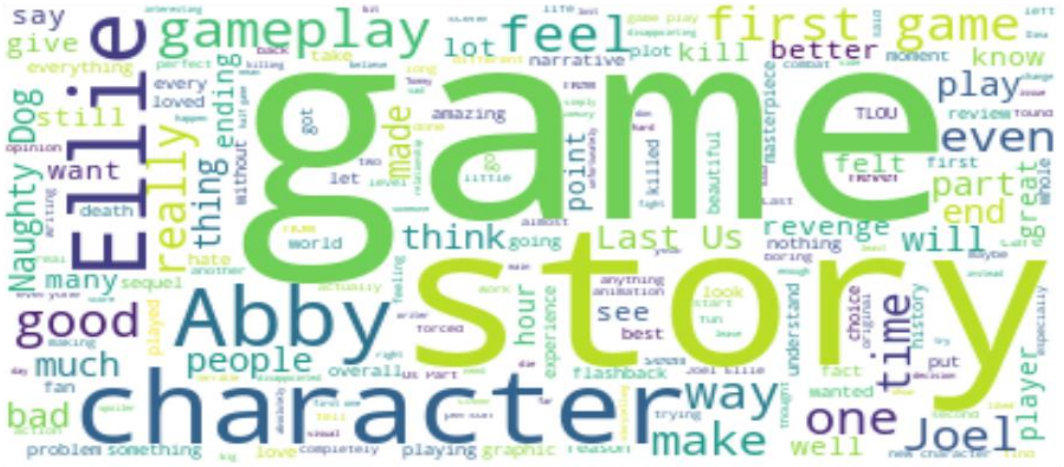


Figure 25 - TLOU2 average review WorldCloud

The above shows the most frequent words used in average TLOU2 reviews. In this WordCloud there is no mention of the word ‘character’. However, terms ‘game’ and ‘story’ are present, just as in the negative reviews WordCloud. The names of characters in the game are used frequently with average reviews such as ‘Ellie’ or ‘Abby’.

Below shows a WordCloud for the positive reviews:

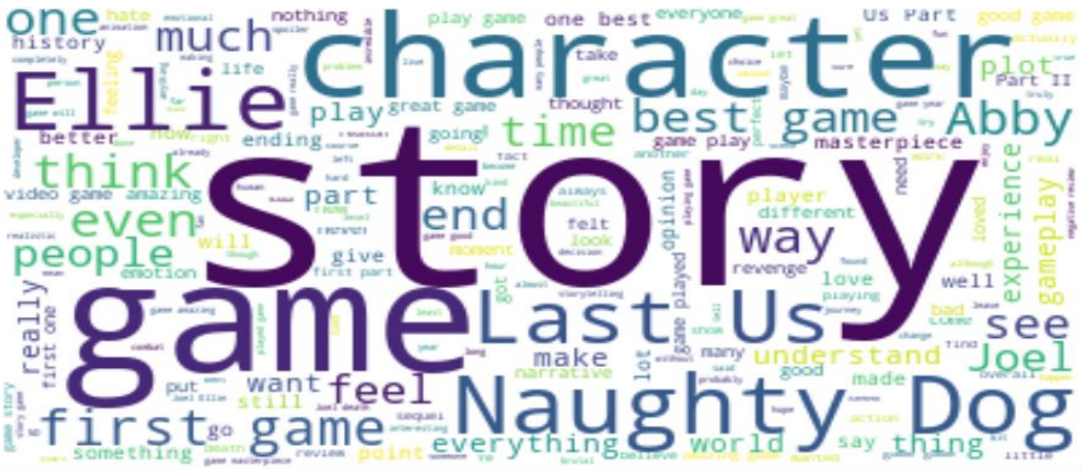


Figure 26 - TLOU2 Positive Review Cloud

The ‘story’ and ‘game’ itself are associated with positive sentiment, just as in the negative and average reviews. However, there appears to be a more common reference to the first game (‘The Last of Us’) and more references to the developer of the game.

For bias prediction there were three cases: only the Amazon model predicted bias, only the Metacritic model predicted bias and both models predicted bias together. The three categories were combined revealing that 22,138 out of 31,309 (71 percent) reviews have potential bias (where a review type that is predicted did not match the actual review type).

The models were examined for where they correctly detected bias by selecting a random starting element in the dataframe and manually checking twenty reviews in each of the categories:

1) **Only Amazon model detected bias:**

The Amazon model found bias where the Metacritic model did not in 5,789 reviews. Through analysing 20 elements adjacent to each other in the dataframe, 60 percent of the reviews were biased through the manual check.

2) **Only Metacritic model detected bias:**

The Metacritic model found bias in 9,336 reviews, where the Amazon model did not. Through analysing 20 adjacent elements in the dataframe, 20 percent of the reviews were biased through the manual check.

3) **Amazon model and Metacritic model detected bias:**

The Amazon and Metacritic model together found bias in 7,013 reviews. Through analysing 20 adjacent elements in the dataframe, 45 percent of the reviews were biased through the manual check.

4) **Amazon model with LSTM layer detected bias:**

The Amazon model with an LSTM layer found bias in 14,587 reviews. Through analysing 20 adjacent elements in the dataframe, 40 percent of the reviews were biased through the manual check. Therefore, this was 20 percent less accurate than the Amazon model without an LSTM layer and was not used for further predictions.

Below shows a table where bias was detected correctly across the models:

Table 4 - Correct bias detection across models for TLOU2

Review	Review Type	Model Detected by	Why?
"Big disappointment. Part 1 was a great game. Stupid plot, clearly more focused on their own agenda instead of making a great game for gamers."	Average (5/10)	Amazon	The review only contained negative sentiment.

“you know one advantage that TLOU2 has is that it has good graphics other than that story, game mechanics, etc. it is bad…”	Negative (1/10)	Metacritic	The review mentioned an advantage but gives an extremely low score
“Don't listen to all the negative user reviews - they haven't even played the game.”	Positive (10/10)	Amazon and Metacritic	The user did not mention the sentiment of the game.
“Good graphics, but story and gameplay outdated. TLOU was way better I think they tried to much about this game.”	Negative (0/10)	Amazon with LSTM layer	The user mentioned positive sentiment but still gave the lowest possible score.

5.3.2 Ghost of Tsushima

For Ghost of Tsushima, 6,471 reviews were used to make predictions on. The dataset had 6,099 positive reviews, 208 negative and 164 average reviews. WordClouds were created for each review type.

Below shows the WordCloud for negative reviews:

Figure 14 - Ghost of Tsushima negative review WordCloud

The above Figure shows that terms such as ‘boring’ or ‘combat’ or ‘gameplay’ are commonly used in the negative reviews for Ghost of Tsushima.

Below shows the average reviews WordCloud:

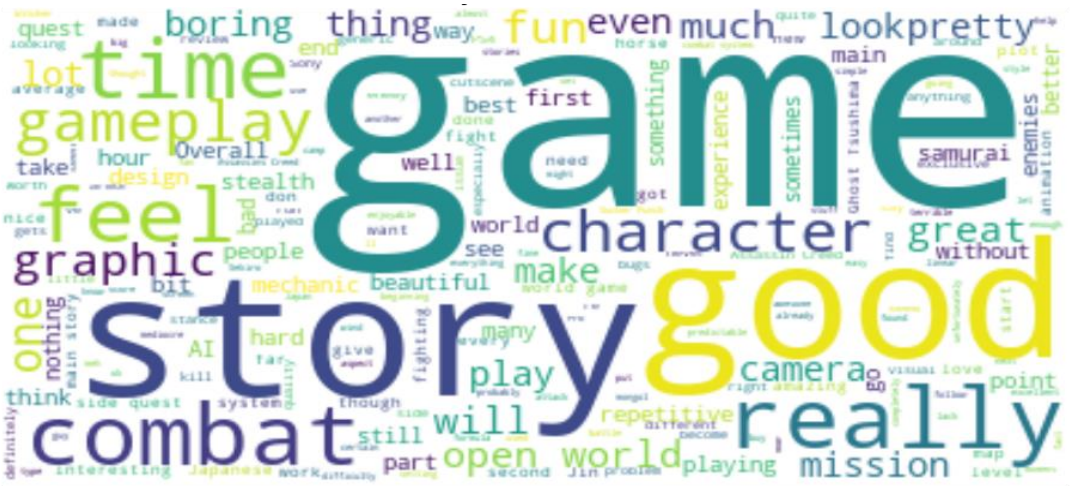


Figure 15 - Ghost of Tsushima average review WordCloud

The above shows that words such as ‘story’ and ‘combat’ are commonly associated with average reviews for the game. Additionally, words such as ‘good’ add to this average sentiment.

Below shows the positive reviews WordCloud:

Figure 16 - Ghost of Tsushima positive review WordCloud

The above shows that terms such as ‘open world’ and the developer of the game, ‘Sucker Punch’ were commonly used words used in positive reviews for the game. Additionally, ‘combat’ was also used here, just as in the average reviews.

There were again three categories for bias detection. The categories combined imply that 2,734 out of 6,471 (42 percent) reviews had potential bias. The models were then examined for where they correctly and incorrectly detect bias by manually checking twenty reviews for bias in each category:

1) **Only Amazon model detects bias:**

The Amazon model found bias in 898 reviews, where the Metacritic model did not. Through analysing 20 adjacent elements in the dataframe, 35 percent of the reviews were biased through the manual check.

2) Only Metacritic model detects bias:

The Metacritic model found bias in 484 reviews, where the Amazon model did not. Through analysing 20 adjacent elements in the dataframe, 30 percent of the reviews were biased through the manual check.

3) Amazon model and Metacritic model detect bias:

The Amazon and Metacritic model together found bias in 1,352 reviews. Through analysing 20 adjacent elements in the dataframe, 50 percent of the reviews were biased through the manual check.

Below shows a table where bias was detected correctly across the models:

Table 5 - Correct bias detection across models for Ghost of Tsushima

Review	Review Type	Model Detected by	Why?
"One of the open world that I loved it, not being a lover of the open world, the game structured in a spectacular way even if it is to be improved a little bit on the graphics, but the rest is beautiful."	Positive (10/10)	Amazon	The user mentioned negative sentiment but gave the game a perfect score.
"Gameplay is extremely fantastic. Although the game is long…"	Positive (10/10)	Metacritic	The user mentioned negative sentiment but gave the game a perfect score.
"Pretty looking game, with decent gameplay, not half-bad story like Abby of Us…"	Positive (10/10)	Amazon and Metacritic	"Decent" gameplay, does not justify a perfect score.

5.3.3 Death Stranding

For Death Stranding, 7,187 reviews were used to make predictions on. This dataset has 4,799 positive reviews, 2,011 negative and 399 average reviews. Word clouds were created for each review type.

Below shows the WordCloud for negative reviews:

Figure 17 - Death Stranding negative review WordCloud

The above shows that words such as the developer, 'Kojima' or 'story' were commonly used words in negative reviews and indicate a large portion of users are not happy with the developer.

Below shows a WordCloud for average reviews:

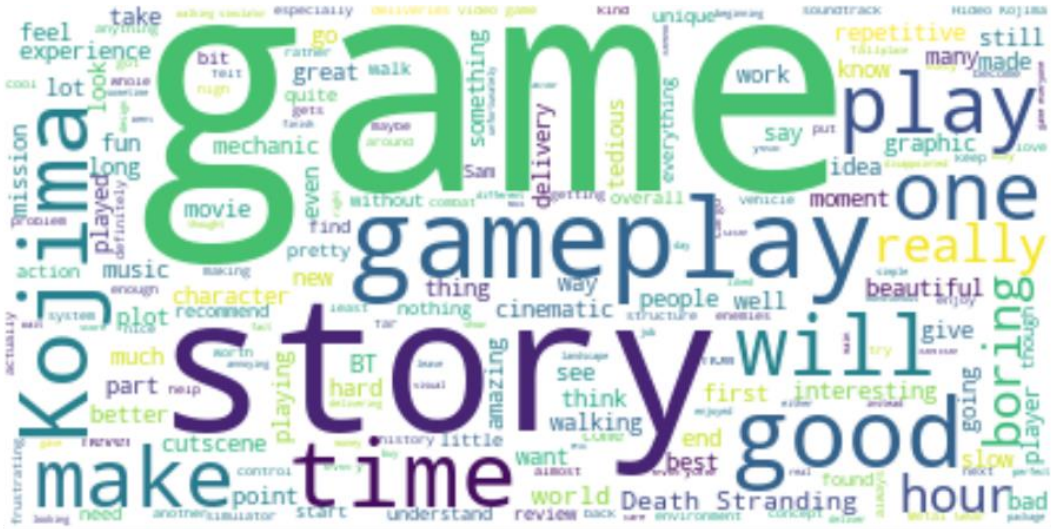


Figure 18 - Death Stranding average review WordCloud

The above shows that 'gameplay' and 'story' are frequently used in average reviews. Additionally, the developer 'Kojima' is used less frequently than in negative reviews.

Below shows a WordCloud for positive reviews:

Figure 19 - Death Stranding positive review WordCloud

The above WordCloud shows that for positive reviews one of the most used words is the 'story' and the 'gameplay' is also used frequently. The developers name is used the least in the positive reviews. Therefore, a huge number of people when rating the game badly blame the developer.

There were again three categories for bias detection. The categories combined imply that 4,453 out of 7,187 (62 percent) reviews have potential bias. The models were examined for where they correctly and incorrectly detect bias by manually checking twenty reviews for bias for each category:

Only Amazon model detects bias:

The Amazon model found bias in 1,663 reviews, where the Metacritic model did not. Through analysing 20 adjacent elements in the dataframe, 40 percent of the reviews were biased through the manual check.

Only Metacritic model detects bias:

The Metacritic model found bias in 1,283 reviews, where the Amazon model did not. Through analysing 20 adjacent elements in the dataframe, 30 percent of the reviews were biased through the manual check.

Amazon model and Metacritic model detect bias:

The Amazon and Metacritic model together found bias in 1,507 reviews. Through analysing 20 adjacent elements in the dataframe, 45 percent of the reviews were biased through the manual check.

Below shows a table where bias was detected correctly across the models:

Table 6 - Correct bias detection across models for Death Stranding

Review	Review Type	Model Detected by	Why?
"This game is very boring, confused and tedious, the main mechanics in the game is just walking, the combats are very simple (if you can find some combat). Graphically the game is really impressive, and the cut scenes and landscape are beautiful, the sound as well, but the rest is a waste of time and money"	Negative (4/10)	Amazon	The user mentioned multiple positive features of the game but leaves a negative review score (on Metacritic's scale)
"This game is proof that GAMEPLAY in a videogame is truly important, I feel bad for bashing this game because the world itself looks pretty…"	Negative (1/10)	Metacritic	The user heavily complemented one aspect of the game but leaves a low review score.
"Whoever said it was right. This game is just a 60-dollar Netflix drama with commercials (the deliveries) as an in-between each 2 hour long cut scene. I will still platinum it since it's a rather time consuming but easy one."	Average (6/10)	Amazon and Metacritic	The user only spoke negatively of the game; therefore, a negative score should follow.

5.4 Top 25 PS4 Games

The percentage of bias detected for the top 25 PS4 games by user score was analysed by looking separately at the Metacritic and Amazon models. Green percentages represent the lowest level of bias for that category, red percentages represent the highest level of bias for the category:

Table 7 - Top 25 PS4 games review bias

Title	Number of Reviews	Amazon Bias (%)	Metacritic Bias (%)	Average Bias (%)
Ghost of Tsushima	6471	34.8	28.4	32
The Witcher 3: Wild Hunt	1826	36.4	32.2	34
The Last of Us Remastered	2415	30.7	24.8	28
God of War	3668	34.1	24.8	30
SpongeBob SquarePants: Battle for Bikini Bottom – Rehydrated	701	53.2	44.9	49
Resident Evil 2	709	50.2	49.8	50
Astro Bot: Rescue Mission	137	41.6	16.8	29
NieR: Automata	703	41.3	33.6	37
Bloodborne	1928	38.5	33.4	36
Detroit: Become Human	1013	42.7	33.1	38
Dark Souls III	464	36.6	36.9	37
Dreams	477	32.5	22.6	27
Marvel's Spider-Man	1545	33.9	50.7	42
Dragon Quest XI	185	33.0	45.4	39
Persona 5	897	29.0	36.0	33
Life is Strange	246	43.9	39.4	42
A Plague Tale: Innocence	182	40.1	45.1	43
The Evil Within 2	199	56.8	54.8	56
Titanfall 2	352	47.7	58.8	53
Uncharted 4: A Thief's End	2194	39.0	34.4	37
Rachet & Clank	363	42.1	49.6	46
Uncharted: The Nathan Drake Collection	338	46.2	45.3	46
Rocket League	352	40.1	38.1	39
Devil May Cry 5	283	49.1	49.8	49
Red Dead Redemption 2	3338	40.1	39.3	40
Average (%)		41	39	40

From the above table, the Metacritic and Amazon models were very similar in terms of their bias detection levels. From the in-depth bias analyses, the Amazon model was more accurate on its predictions than the Metacritic model and was accurate 45 percent of the time. Therefore, the top 25 PS4 games have 18 percent bias according to the Amazon model, once the error is

applied to the bias percentage. The Metacritic model was less accurate in the in-depth bias analysis and is used here as a comparison only.

Below shows the most and least biased titles for the top 25 PS4 games:

The most bias titles according to:

- The Metacritic model – Titanfall 2
- The Amazon model – The Evil Within 2
- Both models collectively – The Evil Within 2

The least bias titles according to:

- The Metacritic model - Astrobot: Rescue Mission
- The Amazon model - Persona 5
- Both models collectively – Dreams

5.5 Worst 25 PS4 Games

The percentage of bias detected for the worst 25 PS4 games by user score was analysed by looking separately at the Metacritic and Amazon models.

Table 8 - Worst 25 PS4 games review bias

Title	Number of Reviews	Amazon Bias (%)	Metacritic Bias (%)	Average Bias (%)
Madden NFL 21	262	3.8	16.8	10
NBA 2K20	557	12.0	15.4	14
FIFA 20	1626	12.1	19.1	16
Star Wars Battlefront II	2379	18.1	20.9	20
Metal Gear Survive	243	21.4	25.5	23
Tony Hawk's Pro Skater 5	116	19.0	60.3	41
Madden NFL 20	318	12.3	76.7	45
WWE 2K20	130	16.2	78.5	47
NBA 2K18	240	15.8	74.2	32
FIFA 19	1626	12.1	80.9	47
Madden NFL 19	122	19.7	68.9	45
Battlefield V	770	29.1	69.1	49
NBA 2K19	143	16.8	76.2	47
Fallout 76	1570	28.7	62.0	45
Call of Duty: Modern Warfare	3928	28.4	77.7	53
Anthem	386	35.5	62.7	49
FIFA 18	183	36.6	69.4	53

Street Fighter V	447	29.3	61.1	45
Fortnite	276	30.1	64.5	47
Mortal Kombat 11	750	27.6	73.7	51
Call of Duty Ghosts	349	47.0	59.9	53
Call of Duty: Infinite Warfare	381	42.5	62.7	53
Call of Duty: Black Ops 4	504	32.9	64.1	49
Madden NFL 21	144	38.2	67.4	53
NBA 2K20	227	30.4	64.3	47
Average (%)		25	59	41

There was huge variance on the level of bias detected between the negatively reviewed games. An example of this was FIFA 19, where the Amazon model detects 12 bias and the Metacritic model detects 81 percent, indicating vastly different predictions.

Due to the large variance between games an additional manual check was conducted on FIFA 19. The manual check showed, as expected, the Amazon model to be more accurate. The Amazon model was accurate 45 percent of the time and the Metacritic model was accurate 15 percent of the time. This shows that the Amazon model is less erroneous at analysing review bias for lower rated games. Thus, it is assumed that the level of bias for the worst 25 games for Metacritic is on average 11 percent after accounting for the error on the Amazon model.

Below shows the most and least biased titles for the worst 25 PS4 games:

The most bias titles according to:

- The Metacritic model – FIFA 19
- The Amazon model – Call of Duty: Ghosts
- Both models collectively – Call of Duty: Ghosts

The least bias titles according to:

- The Metacritic model – NBA 2K20
- The Amazon model – Madden NFL 21
- Both models collectively – Madden NFL 21

6 Discussion

6.1 Achievements

The most important achievement was learning the language, Python. In doing this, I became knowledgeable in using Google Colab and realised the use of a GPU when running models and embedding words. Further, I learnt how to carry out a sentiment analysis by using TensorFlow 2, USE, sklearn and Keras. In doing so, a strong knowledge of building neural networks was built. I adapted a web scraping algorithm to work for scraping Metacritic reviews when there is no text and made the current algorithm over three times faster than it previously was.

There were three hypotheses in the report. Hypothesis One was rejected as the Amazon model had higher accuracy than Metacritic model. The main reason for this will likely be because the Amazon model is trained on over 100,000 more reviews. However, there could be additional factors such as the Amazon reviews are less biased than the Metacritic model. This is highly likely after the bias analysis showed that all the Metacritic games studied showed at least some level of bias.

Hypothesis Two was also rejected as the Amazon model without an LSTM layer was more accurate. The reason for the lower level of accuracy from the layer could be due to the increase in complexity in the model not being necessary. The model without the addition of the LSTM layer train much faster and to a higher degree of accuracy.

Hypothesis Three failed to be rejected as the top 25 PS4 games had a higher level of bias on the Amazon model than the lowest rated PS4 games. The model found 41percent bias on the top 25 Metacritic games, compared to only 25 percent bias on the lowest rated 25 Metacritic games. This 16 percent difference in bias implies that users justify their choice of review score more when discussing games that they dislike.

Out of the fifty titles analysed, the least biased was Madden NFL 21, one of the most recently released titles. The most biased title was The Evil Within 2. FIFA 19 showed the highest level of bias out of all games tested on the Metacritic model. However, these detections of bias were not accurate.

6.2 Future Work

There are various ways to build on the work done in this report. The first of which would be to scrape more data from Metacritic and train the Metacritic model on this. This would lead to a more accurate model due to more training data. An additional way the work could have been improved, is by using the K-Folds Cross Validation for training the models. This style of validation generally leads to a less biased model compared to other methods of training. The reason for the lower bias using this technique, is because it ensures every observation from the original dataset has the chance of appearing in the training and testing set.

If the USE were not used, and more time was spent towards working on the embeddings, then this would lead to a higher accuracy across all models. Many bias detections were found when

the user aims their sentiment towards another game, when in fact they only spoke negatively of the game in the review. Not using pre-trained word embeddings in future work will solve this.

Additionally, from the literature review the model struggled where expected and had issues with detecting sarcastic text and detect fake or spam reviews, as well as misclassifications from poor grammar and punctuation. Therefore, more research is needed in these areas.

7 Conclusion

This study set out to find how biased the user reviews on Metacritic are, because there is often disparity between critic and user reviews on the site. The problem was observed by building two models, one using data from Metacritic and the other using data from Amazon reviews. The Amazon model had better accuracy for predictions than the Metacritic model. These findings alone indicate potential bias in the methods used by the site.

When looking at top 25 and lowest 25 rated PS4 games by user reviews, the percentage of bias ranged from 4 to 57 percent on the Amazon model. And the less accurate Metacritic model showed a bias range for the games examined between 15 to 81 percent bias. On the more accurate model it finds that The Evil Within 2 is the most biased game in terms of user reviews. The average level of bias throughout the Metacritic user reviews is 25 percent for the worst 25 games and 41 percent for the top 25 games. This shows the disparity between user and critic review scores is partially if not completely down to biased user reviews.

The LSTM layer added to the Amazon model made the prediction accuracy worse, and it took six hours longer to run than the model without the LSTM layer. The Metacritic model was less accurate even though it was predicting Metacritic reviews. Thus, having over 100,000 reviews more on the Amazon dataset greatly helped the accuracy.

With additional time I would not have used the USE and would have found a different way to embed the sentences. It would also be interesting to test bias on a larger scale of datasets or even on the entirety of the Metacritic site, which appears to have a considerable influence on consumer behaviour.

References

Agarap, A., 2018. *Deep Learning Using Rectified Linear Units (Relu)*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1803.08375> [Accessed 27 August 2020].

Carremans, B., 2018. *Word Embeddings For Sentiment Analysis*. [online] Medium. Available at: <https://towardsdatascience.com/word-embeddings-for-sentiment-analysis-65f42ea5d26e> [Accessed 25 August 2020].

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B. and Kurzweil, R., 2018. *Universal Sentence Encoder*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1803.11175> [Accessed 24 August 2020].

Hochreiter, S. and Schmidhuber, J., 1997. *Long Short-Term Memory*. [ebook] Available at: <http://www.bioinf.jku.at/publications/older/2604.pdf> [Accessed 27 August 2020].

Jain, S. and Singh, P., 2018. *Systematic Survey On Sentiment Analysis - IEEE Conference Publication*. [online] Available at: <https://ieeexplore.ieee.org/abstract/document/8703370> [Accessed 3 September 2020].

Kumar, A. and Jaiswal, A., 2019. Systematic literature review of sentiment analysis on Twitter using soft computing techniques. Concurrency and Computation: Practice and Experience, 32(1).

Liu, B., 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1), pp.1-167.

MonkeyLearn. 2020. *Sentiment Analysis*. [online] Available at: <https://monkeylearn.com/sentiment-analysis/> [Accessed 25 August 2020].

Olah, C., 2015. *Understanding LSTM Networks*. [online] Colah.github.io. Available at: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> [Accessed 27 August 2020].

Ong, A., 2019. *Web Scraping Metacritic Reviews Using Beautifulsoup*. [online] Medium. Available at: <https://towardsdatascience.com/web-scraping-metacritic-reviews-using-beautifulsoup-63801bbe200e> [Accessed 24 August 2020].

Ramachandran, P., Zoph, B. and Le, Q., 2017. *Searching For Activation Functions*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1710.05941> [Accessed 27 August 2020].

Schreier, J., 2015. *Metacritic Matters: How Review Scores Hurt Video Games*. [online] Kotaku. Available at: <https://kotaku.com/metacritic-matters-how-review-scores-hurt-video-games-472462218> [Accessed 22 August 2020].

Shen, K., 2018. *Effect Of Batch Size On Training Dynamics*. [online] Medium. Available at: <https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e> [Accessed 26 August 2020].

Straat, B. and Verhagen, H., 2017. *Using User Created Game Reviews For Sentiment Analysis: A Method For Researching User Attitudes*. [ebook] Available at: <http://ceur-ws.org/Vol-1956/GHItaly17_paper_01.pdf> [Accessed 21 August 2020].

Subhan, I., 2020. [online] Theboar. Available at: <https://theboar.org/2020/07/what-the-last-of-us-part-ii-tells-us-about-metacritic/#:~:text=The%20Last%20of%20Us%20Part%20II%20(TLOU2)%20was%20always%20going,other%20writers%20at%20The%20Boar.&text=In%20reality%2C%20Metacritic%20exacerbates%20a,as%20a%20metric%20of%20success.> [Accessed 22 August 2020].

Valkov, V., 2019. *Sentiment Analysis With Tensorflow 2 And Keras Using Python*. [online] Curiousily.com. Available at: <https://www.curiousily.com/posts/sentiment-analysis-with-tensorflow-2-and-keras-using-python/> [Accessed 24 August 2020].

Wang, C., 2019. *The Vanishing Gradient Problem*. [online] Medium. Available at: <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484> [Accessed 27 August 2020].

Ye, Q., Law, R. and Gu, B., 2009. The impact of online user reviews on hotel room sales. *International Journal of Hospitality Management*, 28(1), pp.180-182.

Zhu, F. and Zhang, X., 2010. Impact of Online Consumer Reviews on Sales: The Moderating Role of Product and Consumer Characteristics. *Journal of Marketing*, 74(2), pp.133-148.

Appendices

Appendix A

Amazon Dataset – Kaggle: https://www.kaggle.com/snap/amazon-fine-food-reviews

Appendix B

Problematic Web Scraper:
https://gist.github.com/adelweiss/3c13890420662b5041210c48792b4764#file-metacritic-web-scraper

Appendix C

The address of the project's git repository: https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2019/jxb1272 which contains:

- C1: The Amazon Model in a notebook (.pynb)
- C2: The Metacritic Model in a notebook (.pynb)
- C3: The Metacritic Web Scraper in a notebook (.pynb)
- C4: In-depth bias analysis of The Last of Us: Part II in a notebook (.pynb)
- C5: In-depth bias analysis of Death Stranding in a notebook (.pynb)
- C6: In-depth bias analysis of Ghost of Tsushima in a notebook (.pynb)
- C7: Top 25 PS4 games bias (.py)
- C7: The Amazon Model (with LSTM) in a notebook (.pynb)
- C8: In depth bias analysis of The Last of Us: Part II in a notebook: LSTM (.pynb)

The above code has already been executed and is in notebook form, this requires access to my own Google Drive, thus the output has already been shown. Appendix C7 is too large a file to show as a notebook, however this uses identical methods to the ones used in the in-depth bias analyses.