A DEEP LEARNING SOLUTION FOR MORAL DECISION-MAKING USING NATURAL LANGUAGE PROCESSING

Dhruvisha Gosai

Table of Contents

Executive Summary

Usually, there is certain amount of hesitance when it comes to letting the machines make decisions – be it customer service with a bot or using an automated car. Humans would rather rely on humans for support. One such community support network system is observed on Reddit, a platform where people can come and share their opinions and seek feedback. A sub-reddit called 'Am I the Asshole?' (AITA) provides users opinions on whether the decision they made was ethical one or not.

A user labelled metadata with years' worth of information means a perfect opportunity to create a machine learning algorithm that could predict whether a decision made by someone was ethical or not. This research paper explores the Reddit metadata and extracts relevant information with the use of web scrapping. Employing the NLP methods, a deep learning neural networks algorithm is created to predict whether the post-verdict should have been "You're The Asshole" or "Not The Asshole".

1. Introduction

Since the growth of big data, the field of data science has seen prolific expansion and implementation in business and various technology platforms. This poses a question – does the machine learning (ML) model applied need to be ethical in decision-making?

Moral philosophy, a subcategory of philosophy that intends to answer what is right and wrong (Ethics Unwrapped, 2018) and provides a ground on what we ought to do. This is done by three primary branches, out of which deontology – an ethical theory that suggests that the morality of an action should be based on the action itself, whether the action was right or wrong, rather than basing it on consequences (Your Dictionary, n.d.). But for a computer to identify if a particular action was right or wrong on the moral ground is a tricky question to answer.

Even when a human attempts to answer a question dealing with morality, Plato's *Republic* suggests that "appearances are deceiving" and that the action of acting morally is in fact in the rational of self-interest of the individual (Richardson, 2003). Individuals often deviate to satisfying their own interests rather than make a moral decision.

In such moral uncertainty, in 2013, a photographer Marc Beaulac sought to find an answer to who was correct – him or his female coworkers about the office temperature settings. This was when he created a subreddit on a popular discussion website Reddit called 'Am I the Asshole?' (AITA). To overcome the empirical challenges to moral reasoning, AITA's goal was to provide users with honest moral opinion of their behavior, and actionable feedback for how to improve on it (Gordon, 2019). The users employ four options to deliver verdicts – "YTA," ("You're The Asshole"), "NTA" ("Not The Asshole"), "NAH" ("No Assholes Here"), or "ESH" ("Everyone Sucks Here") (Walsh, 2020). This process becomes inefficient when the users start to attack the individuals who post on there rather than providing constructive feedback.

With 97,000 posts (Sivek, 2021), the amount of metadata available provides a great base to create an ML model that utilises the human labelled data with the help of natural-language processing (NLP) to create a model that provides moral judgments. This crowdsourcing technique captures the

patterns of human moral sense; hence this research leverages the freely available data from the AITA subreddit with the aim to create an ML model using a Feed Forward Neural Network (FNN). The data is sourced using 'urllib' module to web scrape which is cleaned using NLP techniques before passing through the FNN.

2. Related work

Previous studies have been done for ethical dilemmas in machines and creating a framework for ethical decision-making by Massachusetts Institute of Technology (MIT) and University of Washington (UW) researchers.

The 'moral machine experiment' by MIT, published in 2020, focused on testing on simulated survey data generated with various underlying moral values. With the virtual participant size of 3,000 participants (Wiedeman, Wang, & Kruger, 2020), employed FNN resulted in the best performance by disregarding the distribution assumption. However, this robust model achieved high accuracy only on non-individual specific data, and there were no evaluation metrics specified. There was another study published in 2022 focusing on 'the morality problem in self-driving cars' (Chandak, et al., 2022) where the algorithm evaluated moral dilemmas for on-road situations.

In 2021, researchers from Allen Institute for Artificial Intelligence, UW, developed an experimental framework called the 'Delphi' (Jiang, et al., 2021) that used 1.5 million records scrapped from 4 different domains (including AITA subreddit) to create a corpus for everyday situations. Reinforcement learning technique was used for this model which rewards the algorithm based on the decision it makes and the performance evaluation was based on using 3-way classification setting (i.e., positive, discretionary, negative). With accuracy of 88.7%, surpassing another large language model called GPT-3 that has the prediction accuracy of 60.2%. Yet, Delphi struggled when asked questions with metaphorical meaning. For instance, when a software developer the system if she should die so she wouldn't burden her friends and family, Delphi said she should (Metz, 2021).

3. Data

To collate the data for this research, 'urllib' is a module that collects several modules for working with URLs WebCrawler was implemented. Data manipulation and model implementation was done using Python 3.8.5. Details of all the libraries and packages used can be found in Appendix. Before extracting the data from subreddit AITA, the link was verified on Cloudflare to ensure that there was no prohibition around web scrapping. Once it was cleared, decisions were made around the data consumption to restrict the amount of data to be processed.

For this research, only the data for top 10,000 high scored posts and their top 10 primary comments were chosen to be scrapped. The corpus was pre-processed to account for casing, spacing and punctuations, followed by lemmatization and detection of frequent chunks for n = 2 to n = max (Borrelli, Svartzman, & Lipizzi, 2020). The cleaned corpus was further passed through Natural Language processing (NLP) algorithms such as Non-negative Matrix Factorization (NMF) for topic modelling before sub-setting into training and testing datasets for Feedforward Neural Network (FNN) algorithm.

3.1 Data Generation via WebCrawler

For this research, reddit website (https://www.reddit.com/r/AmItheAsshole/) was required to be scraped for the data generation. To ensure there were no copyright restrictions and web scraping prohibition, the website was checked for the Terms of service along with passing the website through CloudFlare website. Both the tests provided a positive consensus. Given all the required data was available to be sourced from the website, there were no metadata supplementation were done.

3.1.1 Background

As this research focused on assessing if a decision was based on the moral justification, the post verdicts available for AITA subreddit served as the independent variable that was to be predicted. For this purpose, the initial analysis to extract reddit data, it was found that the most efficient way to web scrape the basic post details was via an API call and then passing the post URL to web scrape. This decision was made as the subreddit page had infinite scrolling, meaning the website information would load as the page was scrolled down. This proved to be a challenge when the initial web scrapping attempt was made which only extracted the 20 posts. Therefore, the decision was made to use API call to extract all the top posts by day, week, month, year, and all-time top post. Once the relevant information was derived, the additional information to extract the top 10 parent comments for a post was web scraped without any issues.

3.1.2 Methodology

The first step of data extraction with API call was done using the Python Reddit API Wrapper (PRAW) package. This package was useful to extract the basic post information such as the post 'Title', 'Post description', 'Unique ID', 'Post score', 'Total comments', 'Post URL'. The post URL was extracted as the result to be passed for further web scraping for the top 10 primary post comments.

Web scraping was done using the two packages 'urllib' and 'requests' which are designed for opening and reading the URLs. URLs obtained through PRAW were using the website format of '*https://www.reddit…*' but they were modified to be '*https://old.reddit…*' (the old reddit website format) because the HTML was simpler to decode and easier to scrape. When scraping, specific user-agent information was passed through as the header and sleep time to pause the calls was set to 1 second to reduce server-side blockages.

The URLs were passed in a loop to extract details for each post and the posts which had the tags of 'Meta' and 'Update' were excluded as these didn't fall under the category of moral decisions. It was ensured that extracted posts were unique, and they had some sort of verdict (excluded the posts which were too young for a verdict). The information retrieved was stored in a list which was then converted to a data frame using pd.Dataframe function. The final dataset consisted of 18 columns as shown in Figure 3.1.2 with final dataset size of 3,120 posts.

```
title
post
id
post_score
total_comments
post_url
post_verdict
comment_1
comment_2
comment_3
comment_4
comment_5
comment_6
comment_7
comment_8
comment_9
comment_10
concat_comment
```

	post_score	total_comments
count	3120.000000	3120.000000
mean	12465.381410	1514.011538
std	11624.336967	1526.669076
min	0.000000	10.000000
25%	1184.250000	199.000000
50%	12811.500000	1199.500000
75%	22141.250000	2342.250000
max	81022.000000	11656.000000

Figure 3.1.2: Columns names and extracted dataset size

3.1.3 Discussion and Limitations

The collected data provided a base to develop a corpus that could be used to create an ML algorithm for morality detection based on the AITA subreddit. There were no major hinderances in web scraping other than the initial testing which resulted in change of algorithm where web scraping of the subreddit page itself was more complex and time consuming than anticipated. This was due to the infinite scrolling feature that the website provides, meaning that only 20 posts were able to be retrieved using 'requests' and 'beautiful soup' packages. Therefore, the algorithm was modified to extract the posts via API call and then web scrape for each post for post description and comments.

An initial test for 10 posts was done before running the complete algorithm. Time constrain was the biggest limitation when web scraping as this algorithm took two hours to run. Hence, once the desired final output was as achieved, the final dataset stored as a dataframe was exported as a csv to ensure there is no requirement in future to re-run the web scraping procedure.

3.2 Data wrangling and Exploratory analysis

The harvested data with 3,120 records had to be transformed before it could be used as an input for the ML algorithm. Any records with a missing post-verdict (6 rows) were deleted from the harvested data. There was further removal of the rows which consisted of values other than 'YTA' and 'NTA', which brought the dataset count down to 2,971. At this stage, the aim was to clean the corpus and add additional features that could be used as part of the input. The spread of the 'post verdicts' (which would be used as the independent variable to be predicted) was assessed using frequency distribution (as shown in Figure 3.2.1). There was an imbalance with only 10.47% of the posts classed as 'YTA' – negative moral verdict and 84.94% classed as 'NTA' – positive moral verdict. This data skewness would mean that the accuracy of an ML algorithm would be better at predicting 'NTA' than 'YTA'. However, for this reason, when creating the ML model stratify = 'y' was used which segregated the data by the post-verdict when splitting it into test and training. Alongside, model was evaluated using the F1 score, precision, and recall instead of accuracy as discussed later in this paper.

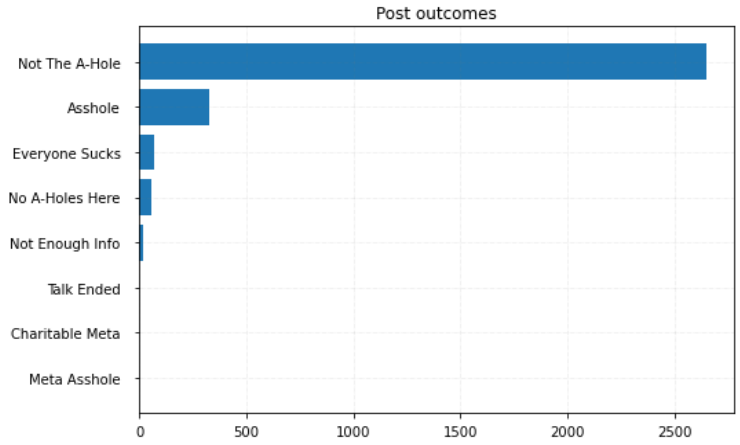

Figure 3.2.1: Frequency distribution of post-verdict

As part of the data pre-processing, all the required columns were dropped – this included individual comments and post URL. Two of the columns – post description and the concatenated comments, were passed through text preprocessing. Text preprocessing included removing of the URLs from the texts, along with any regular expressions (regex) such as special characters. Any stop words were also removed, and the text was lemmatized. Lemmatization was chosen over Porter Stemmer algorithm as stemming reduced the word to forms that were not logical. Text preprocessing was completed with only including nouns in the result as majority of the information was retrieved through just nouns.

As seen in the Figure 3.2.2, majority of the posts had a moral decision made involving the family. Reference to the 'family', 'husband', 'mom', and 'parent' were maximum. This alongside 'time', 'thing', 'year', and 'work' implies that the individuals made a decision based around these topics and were trying to understand if their decision was morally correct. This is when in the comments section, individuals provide their view whether the author of the post was correct in their decision or were they being not morally right. The unique post verdicts were added to a dictionary and the labels (0 to 7) were then put in the dataframe as it would be required for the ML model.

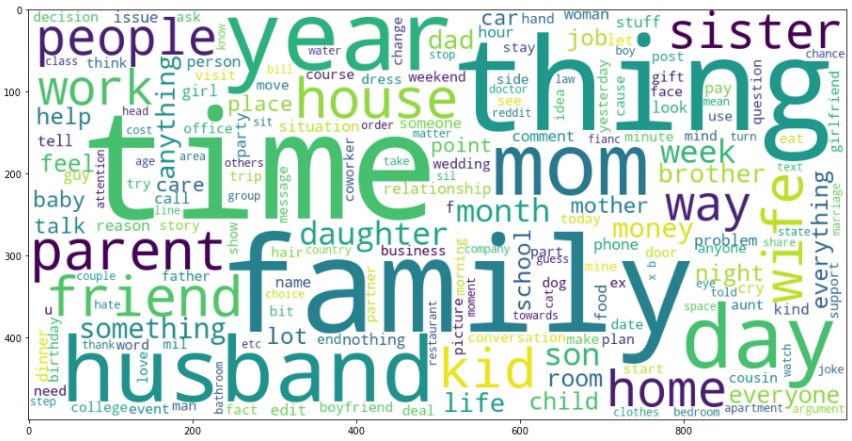

Figure 3.2.2: Word cloud for post description

The 'post description' that would provide majority of the details for the verdict of 'YTA' or 'NTA', this column was vectorized using the term frequency–inverse document frequency (TF-IDF) for 1500 features and n-grams ranging between 1 and 5. Feature names with the highest value were visualized to see if there were any n-grams that were significant, but as observed in Figure 3.2.3, none of the n-grams were part of top 50 words with highest TF-IDF scores.

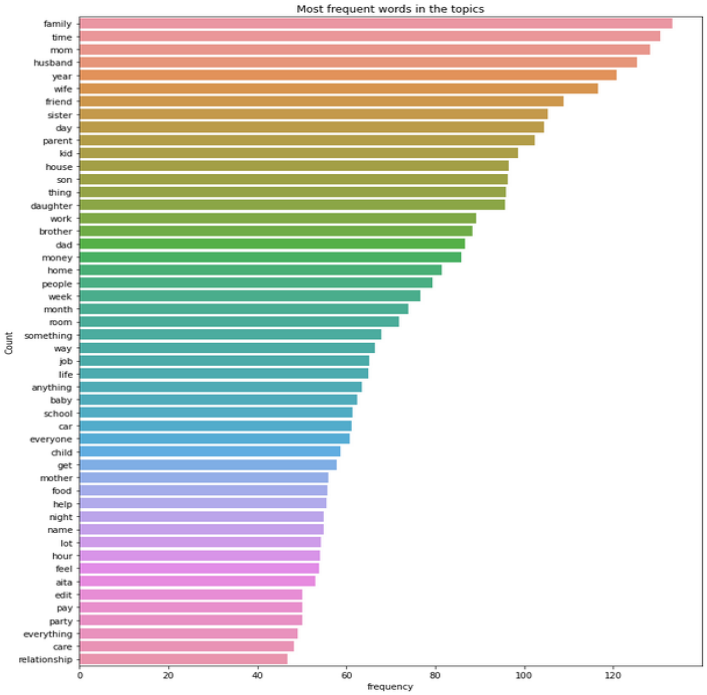

Figure 3.2.4: Top 50 words with highest TF-IDF scores

This TF-IDF vector values was normalized and passed through the non-negative matrix factorization (NMF) for 50 components/topics with the initialization procedure of non-negative double singular value decomposition (NNDSVD) at a random state of 1102. NNDSVD was chosen as it is better for sparse factors – matrices that may mostly consist of zero values (Brownlee, 2018).

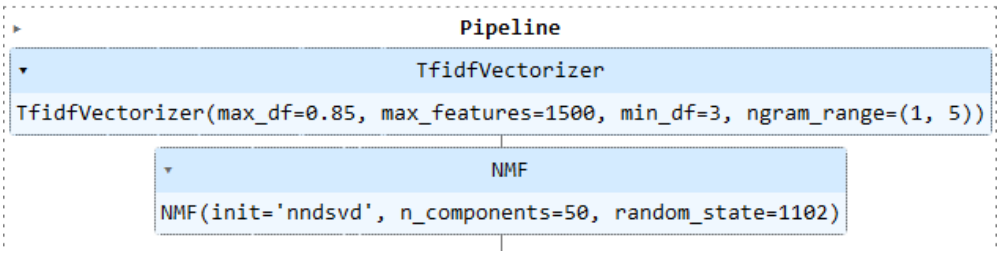

Figure 3.2.4: TF-IDF and NMF parameters

The result from the NMF decomposition was converted into a dataframe that consisted of each topic group with top 10 related words and the topic name as well (the top word in the topic). This was merged with the final dataset. Now, the clean dataset consisted of additional features such as labels, topic modelling, and cleaned post description and cleaned comments. Dataframe was then split into test and training with 20% set aside for testing and 80% was used for training. Out of the training data, 25% was reserved for validation purposes. Given the vectorized values of the post description from TF-IDF would be of different lengths, 'pad_sequences' function from keras package was used to pad zeros at the end of the sequence to make the values in the matrix the same size.

Using the Global Vectors for word representation (GloVe) – 'GLOVE 6B 50D Word Embeddings' sourced from Kaggle, an embedding matrix was created using the training dataset to be used as a hidden layer for the ML model.

4. Deep Learning Algorithm Implementation

With the maximum number of words in the column 'post description' being 1500 after the text preprocessing, depth-wise separable convolutional neural network (sepCNN) which has its architecture based on *Inception*. Inception module factors in cross-channel correlations and spatial correlations (Chollet, 2017). This meant that the network design would be able to identify patterns to predict the correct class of ethical decision – 'YTA' / 'NTA'. Due to the size of the data passed into the model was not huge, there was no requirement to run the model on GPU. CPU handled the model run effectively with no delays.

Using the TF-IDF vectorization of the 'post description' as the input, this model was created with 3 types of layers – input layer, hidden or dense layer, and output layer. Input layer consisted of the shape of the input data and an embedding layer where the GloVe embedding matrix mentioned in Section 3.2 was passed for weights. A dropout layer was added with the rate of 0.2 in between the embedding layer and the two sepCNN nets. This was to avoid overfitting of the data. The two sepCNN nets were created with output dimension of the layer set to 64, convolution window or kernel size of 8, and rectifier or ReLU activation function with a bias initializer and depthwaise initializer set to be 'random_uniform'. This initializer generates tensors with a uniform distribution (Keras, n.d.). Following the two sepCNN layers, a layer to down-sample using Max pooling operation for 1D temporal data (Keras, n.d.) was added.

Before creating the hidden layers, a dropout layer was added again with the same rate as before (rate of 0.2), and a flatten layer was included to convert the multidimensional output to a linear input into the dense layers. For the first iteration of the model, one hidden layer with ReLU activation function and 10 units for this layer was set. As for the final output layer, sigmoid function was selected as it maps logistic output (log odds) to the probabilities (Google Developers, n.d.) which is what is required for the model predictions.

```
Model: "Model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_76 (Embedding)    (None, 1500, 50)          510250

 dropout_34 (Dropout)        (None, 1500, 50)          0

 separable_conv1d_30 (Separa  (None, 1493, 64)         3664
 bleConv1D)

 separable_conv1d_31 (Separa  (None, 1486, 64)         4672
 bleConv1D)

 max_pooling1d_55 (MaxPoolin  (None, 743, 64)          0
 g1D)

 dropout_35 (Dropout)        (None, 743, 64)           0

 flatten_62 (Flatten)        (None, 47552)             0

 dense_95 (Dense)            (None, 10)                475530

 Output-Layer (Dense)        (None, 1)                 11

=================================================================
Total params: 994,127
Trainable params: 483,877
Non-trainable params: 510,250
_________________________________________________________________
```

```
model1.fit(X_train
           ,y_train
           ,validation_data = (X_val, y_val)
           ,class_weight=class_weights
           ,epochs=10
           ,verbose=2  # Logs once per epoch
           ,batch_size=100
          )
```

To compile the model, binary cross entropy loss function with ADAM optimizer was used. For this iteration of the model, epochs were set to 10 with batch size of 100. However, due to the data imbalance where 89.03% of the records were classed as 'NTA', compute_class_weight from scikit-learn was used to calculate the class weights and passed as part of the model execution.

```
Total: 2971
Positive (NTA): 2645 (89.03% of total)
Negative (YTA): 326  (10.97% of total)
```

As the results of the first iteration were underwhelming, certain hyperparameters were updated to see if there was any improvement in the performance. The second iteration was modified to have two additional layers with ReLU activation function instead of only one hidden layer. The output dimensions for the two sepCNN layers were updated to be 32 and kernel size updated to be 5 instead of 8. The optimizer was also updated to be Adam with Nesterov momentum (NADAM) with epochs of 150 and batch of 10.

```
Model: "Model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_78 (Embedding)     (None, 1500, 50)          510250

dropout_38 (Dropout)         (None, 1500, 50)          0

separable_conv1d_34 (Separa  (None, 1496, 32)          1882
bleConv1D)

separable_conv1d_35 (Separa  (None, 1492, 32)          1216
bleConv1D)

global_average_pooling1d_7   (None, 32)                0
(GlobalAveragePooling1D)

dropout_39 (Dropout)         (None, 32)                0

flatten_64 (Flatten)         (None, 32)                0

dense_99 (Dense)             (None, 12)                396

dense_100 (Dense)            (None, 10)                130

dense_101 (Dense)            (None, 2)                 22

Output-Layer (Dense)         (None, 1)                 3

_________________________________________________________________
Total params: 513,899
Trainable params: 3,649
Non-trainable params: 510,250
```

```python
# fit model
model2.fit(X_train
           ,y_train
           , validation_data = (X_val, y_val)
           ,class_weight=class_weights
           ,epochs=150
           ,verbose=2  # Logs once per epoch
           ,batch_size=10
          )
```

5. Results and Discussion

There were two iterations of the model run with modifications to the hyperparameter to assess the difference in performance. With significant imbalance in the categorical variable, there were certain options considered such as over-sample using SMOT, but the eventual decision was made to use the class weights passed as part of the model execution.

5.1 Performance evaluation

To overview the model performance, a confusion matrix was created for both the iterations to see how many observations were classed as ‘YTA’ and ‘NTA’. Looking at figure 5.1.1 (a) and 5.1.1 (b), both the models performed the same in classing all the 595 observations as NTA. The class weights did not provide enough weight to YTA and alternative method such as over-sampling may have provided a better base for improvement.

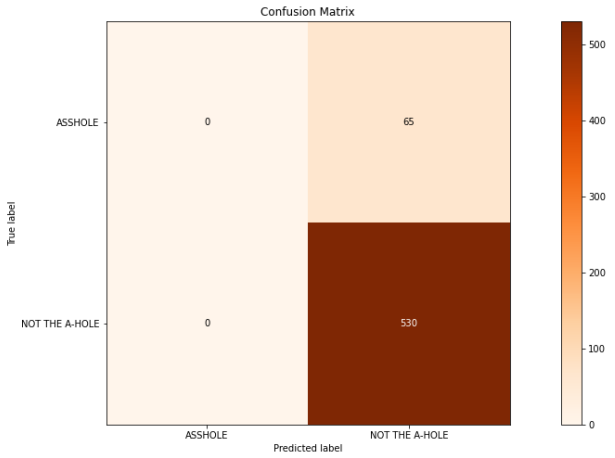

Figure 5.1.1 (a): Model iteration 1 results

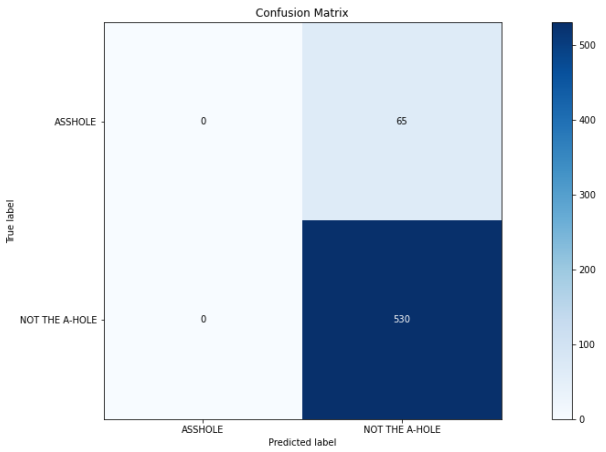

Figure 5.1.1 (b): Model iteration 2 results

Key performance evaluation metrics used for the model were precision, recall and F1 scores. By looking at the figures 5.1.2 (a) and (b), it was evident that there was no model improvement following the hyperparameter tuning. The data imbalance was a greater issue and couldn't be solved by just adding class weights. With the F1-score of 0.94, the model is deemed to be very efficient in predicting NTA. Although, with a score of zero for predicting YTA, overall model performance is not desirable.

```
Train Accuracy : 0.8911335578002245
Train Accuracy : 0.8872053872053872
Test  Accuracy : 0.8907563025210085

Classification Report :
               precision    recall  f1-score   support

NOT THE A-HOLE       0.89      1.00      0.94       530
      ASSHOLE        0.00      0.00      0.00        65

     accuracy                           0.89       595
    macro avg       0.45      0.50      0.47       595
 weighted avg       0.79      0.89      0.84       595
```

Figure 5.1.2 (a): Model 1 performance evaluation

```
Train Accuracy : 0.8911335578002245
Train Accuracy : 0.8872053872053872
Test  Accuracy : 0.8907563025210085

Classification Report :
               precision    recall  f1-score   support

NOT THE A-HOLE       0.89      1.00      0.94       530
      ASSHOLE        0.00      0.00      0.00        65

     accuracy                           0.89       595
    macro avg       0.45      0.50      0.47       595
 weighted avg       0.79      0.89      0.84       595
```

Figure 5.1.2 (b): Model 2 performance evaluation

5.2 Limitations

Data for this research was sourced from Reddit and the variable to be predicted was derived from the user inputs. If there was an update after the user posted on the AITA subreddit, the final post-verdict would be updated to be 'Update'. There were also instances where the posts were too young to have any tags. Such reliance on the user data causes the size of the corpus to go down. Another major drawback was data skewness with the post-verdict. There were approximately 11% of the posts that were YTA category. This meant that the model learnt how to predict NTA but for YTA, it lacked so much so that the model was not able to predict the class at all.

Based on the literature reviews and research based in the field of deep learning for NLP, sepCNN and CNN were deemed as the most appropriate models to predict the class accurately. However, even the research papers such as the Delphi experiment that used AITA data to build ML models, failed to explain how the data imbalance was addressed and how was the model evaluated. Reviewing the two iterations of the model with different hyperparameters, it is evident that sepCNN would not be a good choice when dealing with highly imbalanced data.

5.3 Recommendations

One of the major issues this model faced was data imbalance. SepCNN is not a robust model that could account for such bias. Potentially, using ML models such as XGBoost or Random Forest for instance could have improved the performance than deep learning models. Additionally, other sampling techniques such as under-sampling or over-sampling could be useful to balance the minority class.

6. Conclusion and Future work

People often question their judgement when their loved ones are emotionally impacted. Which is when they seek to understand whether their decision was correct or not. This paper aimed to solve this question of identifying if a person was right in their decision or not with the use of advance techniques such as web scrapping, natural language processing, and deep learning.

Using API and web scrapping tools, the collated data was cleaned via text preprocessing techniques to prepare the corpus for deep learning algorithm. SepCNN was employed as it is efficient with spatial relationships (OpenGenus IQ, n.d.). The employed neural networks provided a good prediction for NTA but failed to predict YTA.

There is a potential to extend this research to further look into robust solutions to data imbalance as it is a very common scenario where one category could have a lot more data than the other causing mis-predications.

References

Borrelli, D., Svartzman, G. G., & Lipizzi, C. (2020). Unsupervised acquisition of idiomatic units of symbolic natural language: An n-gram frequency-based approach for the chunking of news articles and tweets. *PLOS ONE 16(1): e0245404 (https://doi.org/10.1371/journal.pone.0245404)*.

Brownlee, J. (2018, Mar 14). *A Gentle Introduction to Sparse Matrices for Machine Learning*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/sparse-matrices-for-machine-learning/

Chandak, A., Aote, S., Menghal, A., Negi, U., Nemani, S., & Jha, S. (2022). Two-stage approach to solve ethical morality problem in self-driving cars. *AI & Society (https://doi.org/10.1007/s00146-022-01517-9)*.

Chollet, F. (2017). *Xception: Deep Learning with Depthwise Separable Convolutions.* Google, Inc.

Ethics Unwrapped. (2018, Dec 19). *Moral Philosophy*. Retrieved from Ethics Unwrapped: https://ethicsunwrapped.utexas.edu/glossary/moral-philosophy

Google Developers. (n.d.). *Machine Learning Glossary*. Retrieved from Google Developers: https://developers.google.com/machine-learning/glossary?utm_source=DevSite&utm_campaign=Text-Class-Guide&utm_medium=referral&utm_content=glossary&utm_term=sigmoid-function#sigmoid_function

Gordon, I. (2019, Aug 9). *How 'Am I the Asshole?' became the internet's most profound query*. Retrieved from Daily Dot: https://www.dailydot.com/unclick/am-i-the-asshole-aita-reddit-history/

Jiang, L., Hwang, J. D., Bhagavatula, C., Bras, R. L., Liang, J., Dodge, J., . . . Choi, Y. (2021, Oct 14). Can Machines Learn Morality? The Delphi Experiment. *arXiv (https://doi.org/10.48550/arxiv.2110.07574)*.

Keras. (n.d.). *Layer weight initializers*. Retrieved from Keras: https://keras.io/api/layers/initializers/

Keras. (n.d.). *MaxPooling1D layer*. Retrieved from Keras: https://keras.io/api/layers/pooling_layers/max_pooling1d/

Metz, C. (2021, Nov 19). *Can a Machine Learn Morality?* Retrieved from The New York Times: https://www.nytimes.com/2021/11/19/technology/can-a-machine-learn-morality.html

OpenGenus IQ. (n.d.). *When to use Convolutional Neural Networks (CNN)?* Retrieved from OpenGenus IQ: https://iq.opengenus.org/when-to-use-convolutional-neural-network-cnn/

Richardson, H. S. (2003, Sep 15). *Moral Reasoning*. Retrieved from Stanford Encyclopedia of Philosophy: https://plato.stanford.edu/entries/reasoning-moral/#MoraPrin

Sivek, S. C. (2021, Jul 30). *Am I the…Data Geek Who Analyzed Reddit AITA Posts? Yes.* Retrieved from Towards Data Science: https://towardsdatascience.com/am-i-the-data-geek-who-analyzed-reddit-aita-posts-yes-4954a8d37055

Walsh, K. (2020, July 27). *Reading Reddit Drama Helps Some People Leave Bad Relationships*. Retrieved from Vice: https://www.vice.com/en/article/y3z5av/reading-reddit-relationships-amitheasshole-aita-helps-some-leave-bad-relationships

Wiedeman, C., Wang, G., & Kruger, U. (2020). Modeling of moral decisions with deep learning. *Visual Computing for Industry, Biomedicine, and Art*, 3:27.

Your Dictionary. (n.d.). *Deontology definition*. Retrieved from Your Dictionary: https://www.yourdictionary.com/deontology

Appendix

Github Link

https://github.com/dhru-gosai/SepCNN-for-Reddit-AITA-using-NLP

Literature Analysis

1. Moral Machine Experiment (2020): Massachusetts Institute of Technology researchers
   - Data: tested with simulated Moral Machine survey data generated with various underlying distributions of moral values.
     - virtual participants = 3,000 participant; 1,000 set aside for test.
   - Method: Three models - deep learning model (Feedforward NN), Hierarchical Bayesian model (underlying distribution was assumed), Maximum likelihood model (no distribution assumptions are made)
   - Results: leverage a deep learning model combined with a maximum likelihood component to better extract both group trends and individual specific information from limited data, or could train deep neural networks to weight both moral and legal considerations

    - deep learning (DL): Best performance, highly adaptive to training examples, no assumptions regarding the distribution of moral values in a population, and Robust result
    - Hierarchical Bayesian (HBM): maximized the posterior probability, estimate used to predict other scenario decisions, HBM outperformed the ML model in all instances
    - Maximum likelihood model (ML): model's performance was mostly invariant of the underlying distribution (predictions only on limited, individual-specific data, without any prior assumption)
- Limitations: No use of NLP and real population, DL was only able to achieve high accuracy without individual-specific data, individually specific data would become increasingly important for accurate modeling as in-group variance increases. ML predicts the maximum likelihood decision, and not the decision itself.

2. <u>Delphi (Oct 2021):</u> Allen Institute for Artificial Intelligence, University of Washington researchers. Experimental framework based on deep learning
    - Data:
        - 1.5M Everyday situations (data scrapped from 4 domains)
            - Am I the Asshole? (AITA) subreddit
            - Confessions subreddit
            - ROCStories corpus
            - Dear Abby advice column
        - 144K Contexualised narratives
        - 28K Social justice and biases
        - 21K Unambiguous moral situations
    - Method: Descriptive model for commonsense moral reasoning trained in a bottom-up manner.
        - taught by COMMONSENSE NORM BANK, a compiled moral textbook customized for machines
        - trained from UNICORN, a T5-11B based neural language model specialized in commonsense question answering
        - input a query and responds with an answer is yes/no or free-form forms.
        - Lemmitisation
    - Results: accuracy = 88.7%
        - better GPT-3 that makes correct predictions 60.2% of the time
        - over four million queries to date from public
        - evaluation = 3-way classification setting (i.e., positive, discretionary, negative)
    - Limitations: with large volume of testing on the model by public queries, results have been surprisingly good but unsurprisingly biased
        - challenges around convoluted situations with long contexts
        - low performance in expressions where the literal expression deviates far from the metaphorical meaning.
            - example, "being all eyes and ears" predicts it as a "bad" action, and "telling someone to 'break a leg'" as a "rude" action.
            - When a software developer stumbled onto Delphi, she asked the system if she should die so she wouldn't burden her friends and family. Delphi said she should.

(https://www.nytimes.com/2021/11/19/technology/can-a-machine-learn-morality.html)

3. Morality problem in self-driving cars (May 2022): AI & Society (Journal of Knowledge, Culture and Communication)
   - Data: model is simulated for four lane traffic conditions
     - model is evaluated for light, medium, and heavy traffic
     - conditions to maximize the reward in different conditions
       - 15 cars for light traffic
       - 30 cars for medium traffic
       - 50 cars for heavy traffic
   - Method: reinforcement learning (Deep Q-Network - DQN)
     - Replay memory size is 10^6 and model is run 10 times
   - Results: 1000 people on their critical response to moral dilemmas for on-road situations.
     - gives a maximum reward of 76 under light traffic conditions
   - Limitations:
     - Model designed for moral dilemma situations only
     - performance evaluation done by glancing at 3 scenarios and not based on appropriate performance metrics
     - 1000 people surveyed for different on-road situations, but these responses were not compared against model outputs

Un-used Graphs

1. Word cloud not excluding the Nouns in text-preprocessing for post descriptions

2. Sentiment scores from post descriptions