COMP 472 MP1 Analysis

4.1)

Given a significant difference in the distribution of emotions, this shows that some emotions that occur less are weakly represented within the dataset for the training phase. This affects a model's precision and recall metrics as it has fewer instances to infer for classes that occur less such as the emotion of “grief” during the testing phase. The most prevalent emotion appears “neutral”, approximately one-third of the total emotion outputs resulted in the emotion “neutral” this allows for posts that may have multiple emotions for an identical post to possibly favour neutral over other emotions since neutral is represented more than other emotions.

In terms of the sentiment dataset, we can observe a more equal distribution as depicted below. As a result of the more equal distribution of sentiments metrics such as recall and precision should perform better given that there are more cases to refer to when verifying for false negatives and false positives respectively.

By observing the dataset we can observe posts that are identical being tagged with different emotions and sentiments. This depicts that posts that are identical may not be classified consistently for either emotion or sentiment and may introduce some uncertainty or noise when determining a single result for either Emotion or Sentiment.

Post	Emotion	Sentiment
They have THE BEST tortillas too. Made right in front of you. Mmmmmm!	neutral	neutral
They have THE BEST tortillas too. Made right in front of you. Mmmmmm!	neutral	neutral
They have THE BEST tortillas too. Made right in front of you. Mmmmmm!	admiration	positive
They have THE BEST tortillas too. Made right in front of you. Mmmmmm!	admiration	positive
They have THE BEST tortillas too. Made right in front of you. Mmmmmm!	approval	positive

Table 1 Sample of outcomes for a specific posts

The following the the code that was used to generate Table 1
print(file.loc[file[0]=='They have THE BEST tortillas too. Made right in front of you. Mmmmmm!'])

Emotion	Occurrences
'neutral'	55298
'approval'	11259
'admiration'	10531
'annoyance'	8342
'disapproval'	7686
'gratitude'	7075
'amusement'	6130
'curiosity'	5885
'anger'	5202
'love'	4957
'confusion'	4938
'realization'	4714
'disappointment'	4706
'optimism'	4519
'joy'	4329
'sadness'	3827
'caring'	3523
'surprise'	3472
'excitement'	3020
'disgust'	2914
'desire'	2147
'fear'	1778
'remorse'	1510
'embarrassment'	1433

'nervousness'	796
'relief'	788
'pride'	690
'grief'	351

Table 2: Occurrences of Emotions

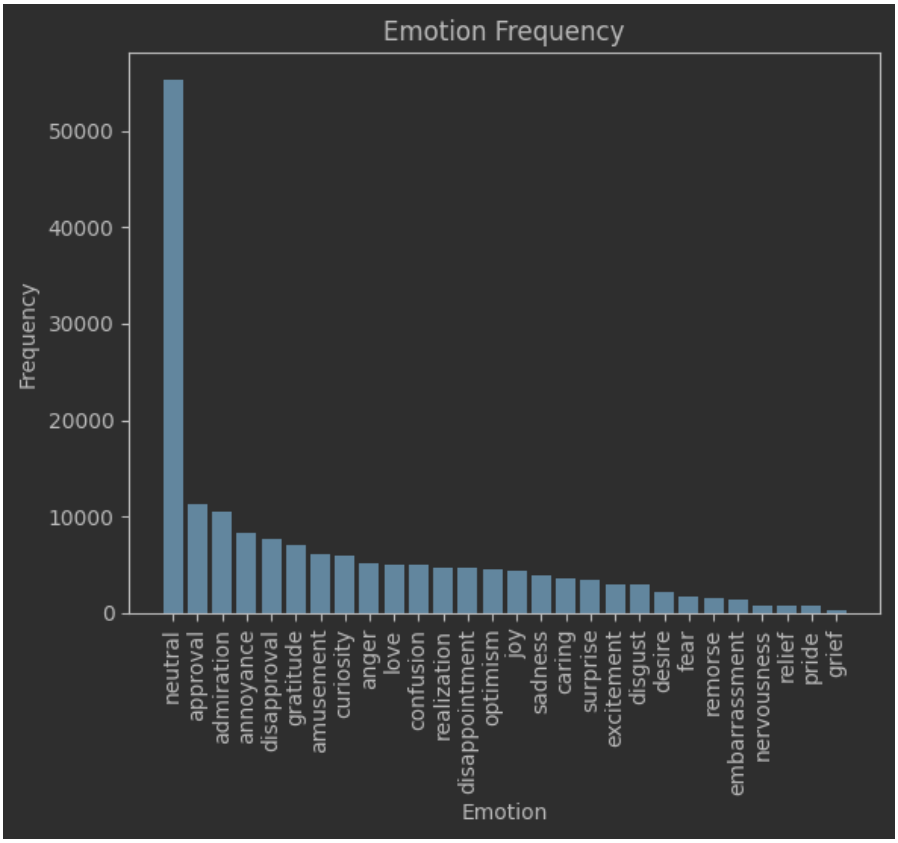

Figure 1: Number of Occurrences of Emotions

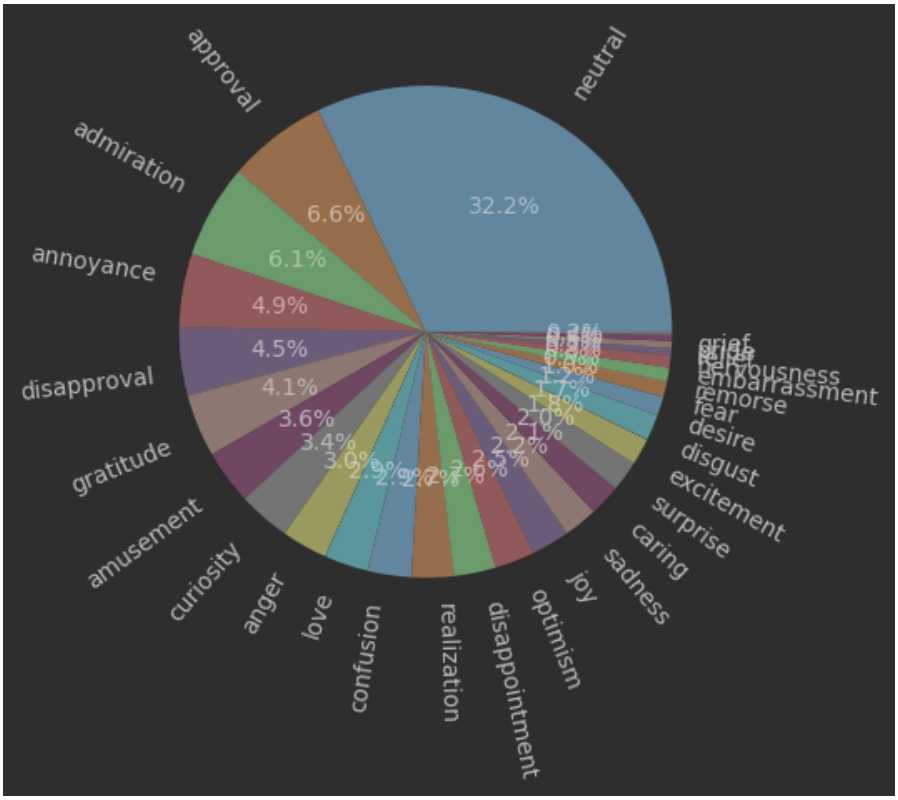

Figure 2: Relative Frequency of Emotions

Statistic	
mean	6136.428571
std	10041.271704
min	351.000000
25%	2054.750000
50%	4424.000000
75%	5946.250000
max	55298.000000

Table 3: Descriptive Statistics about Emotion Occurrences

Sentiment	Occurrences
'positive'	58968
'neutral'	55298
'negative'	38545
'ambiguous'	19009

Table 2: Occurrences of Sentiments

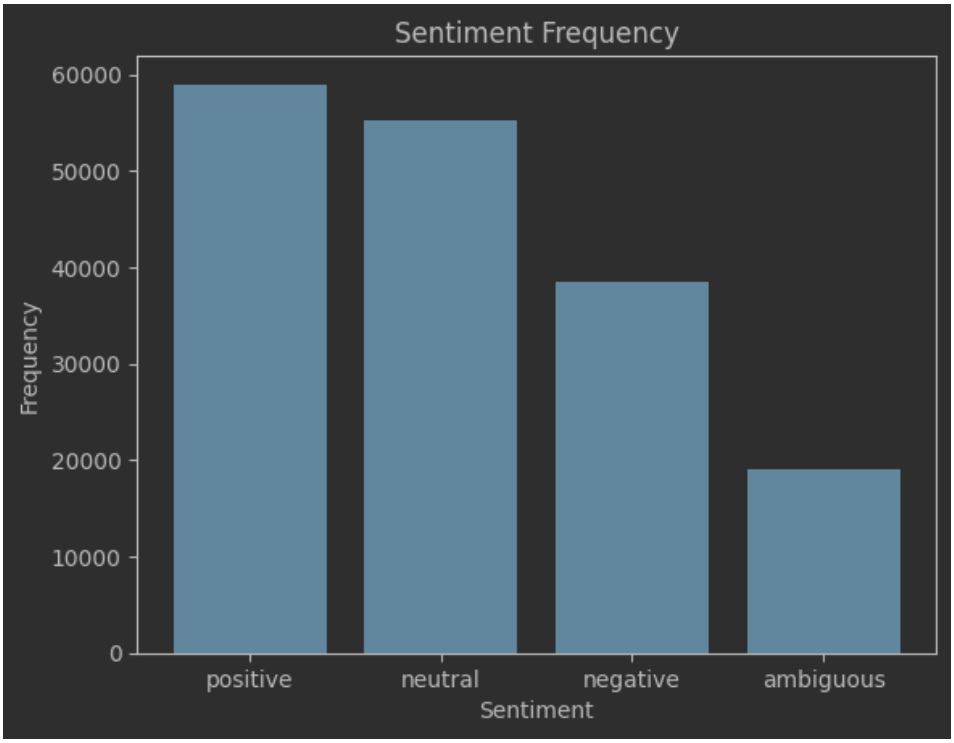

Figure 3: Number of Occurrences of Sentiments

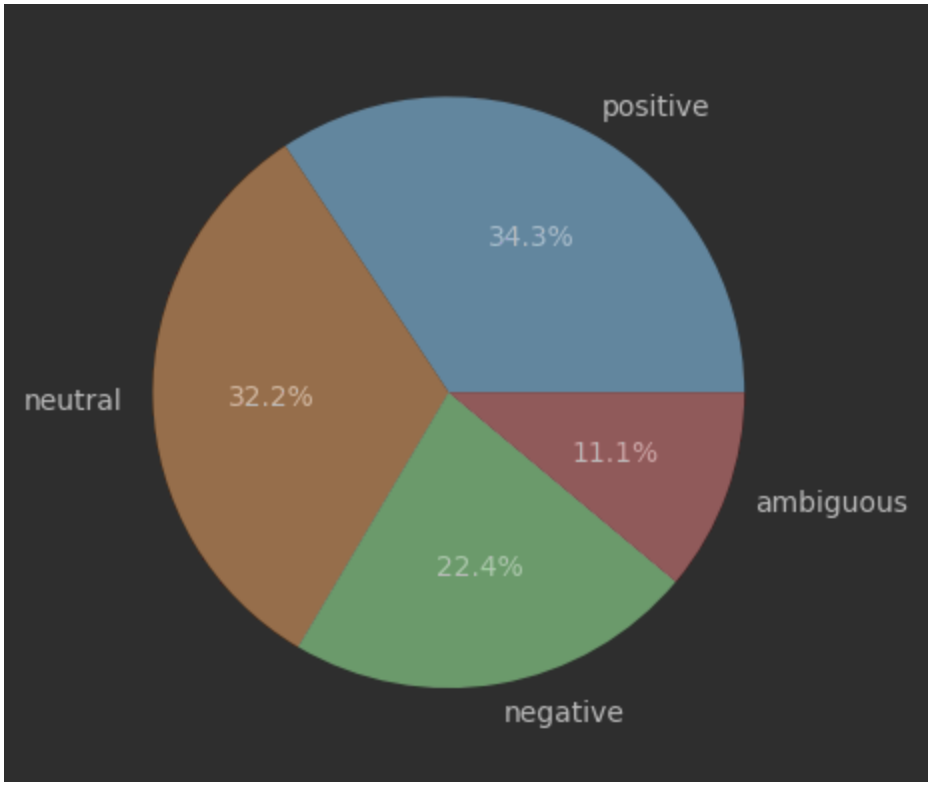

Figure 4: Relative Frequency of Sentiment

Statistic	
mean	42955.0000
std	18272.2452
min	19009.0000
25%	33661.0000
50%	46921.5000
75%	56215.5000
max	58968.0000

Table 4: Descriptive Statistics about Sentiment Occurrences

4.2)

Multinomial Naive Bayes Classifier

Comparing the Multinomial Naive Bayes Classifier using the different training test split we can observe that in terms of emotions the model did show very slight improvement for the metrics precision, recall and f1-score. Although the increase in performance is typically below 5% and may be a result of the training process. Further and repeating testing should be done for a more conclusive answer.

In relation to sentiment prediction, the model did improve when provided more training data but did not improve enough to be significant that cross-validation and more runs must be performed before reaching any conclusion that less training data would increase the model's effectiveness.

Data can be found in the "performance Excel.pdf" or performance.xlsx file.

Decision Tree Classifier

Comparing the Decision Tree Classifier using the different training test split we can observe that in terms of emotions the model did show improvement for the metrics precision, recall and f1-score. Increase in performance appears to be insignificant and may be a result of the training process. Further and repeating testing should be done for more conclusive findings.

In relation to sentiment prediction the model did improve with only a minor decrease in the recall of a single class when provided less training data but did not improve enough to be significant that cross-validation and more runs must be performed before reaching any conclusion that less training data would increase the models effectiveness.

Data can be found in the "performance Excel.pdf" or performance.xlsx file.

Multi-Layer Perceptron

Comparing the Multi-Layer Perceptron Classifier using the different training test split we can observe that in terms of emotions the model did suffer immensely for the metrics macro precision, macro recall and macro f1-score. Decrease in performance appears to be a significant decrease in precision of ~5%, decrease in recall of 23% and decrease of 19% in the f1-score. These metrics did not suffer as immensely when observing the weighted averages although decreases of 11%,11% and 15% respectively are still significant enough to conclude there may be a negative impact of providing more than 80% of the data for training in relation to emotions.

Although when it came to the sentiment prediction there was a significant increase in performance of the macro average for precision, recall and f1-score as followed 15%, 21% and 21%. This increase remains similar when observing the weighted averages of 15%, 16% and 17% respectively.

Data can be found in the “performance Excel.pdf” or performance.xlsx file.

Multinomial Naive Bayes Classifier using GridSearchCV

Comparing the Multinomial Naive Bayes Classifier using the different training test split we can observe that in terms of emotions the model did show very slight improvement for the metrics precision, recall and f1-score. Although the decrease in performance is typically below 5% note that recall did suffer significantly with all classes decreasing by a few percent. Further and repeated testing should be done for a more conclusive answer.

In relation to sentiment prediction the model did not improve when provided less training data to be significant that cross-validation and more runs must be performed before reaching any conclusion that less training data would increase the model’s effectiveness

Data can be found in the “performance Excel.pdf” or performance.xlsx file.

Decision Tree Classifier using GridSearchCV

Given the hyperparameters of "criterion":"gini","entropy", "Max_depth":100,3, "min_samples_split":12,5,30.

Emotion: The 80/20 split found the best configuration was max_depth=100 with min_sample_split of 30. While the 50/50 split found max_depth=100 with min_sample_split of 30.

Sentiment: The 80/20 split found the best configuration was max_depth=100 with min_sample_split of 5. While the 50/50 split found was max_depth=100 with min_sample_split of 5.

Comparing the Decision Tree using the different training test split we can observe that in terms of emotions the model did suffer immensely for the metrics macro precision, macro recall and macro f1-score. A decrease in performance appears to be a significant decrease in precision of ~4%, decrease in recall of 18% and decrease of 15% in the f1-score with every class suffering. These metrics did not suffer as immensely when observing the weighted averages although decreases of 10%,8% and 13% respectively; these are still significant enough to conclude there may be a negative impact of providing more than 80% of the data for training in relation to emotions.

Although when it came to the sentiment prediction there was a slight decrease in performance of the macro average for precision, recall and f1-score as followed 5%, 13% and 11%. This increase remains similar when observing the weighted averages of 6%, 9% and 9% respectively.

Data can be found in the “performance Excel.pdf” or performance.xlsx file.

Multi-Layer Perceptron using GridSearchCV

All Multi-Layer Perceptron using GridSearchCV models were set to max iterations of 5 instead of the default 200 to ensure that a complete model was produced. Efforts in favour of speeding up the GridSearchCV were made including allowing the library to run on all cores using n_jobs=-1 as well as allowing for early_stopping.

Comparing the Multi-Layer Perceptron Classifier using the different training test split we can observe that in terms of emotions the model did suffer for the metrics macro precision, macro recall and macro f1-score. A decrease in performance appears to be a significant decrease in precision of ~8%, decrease in recall of 2% and decrease of 2% in the f1-score. These metrics did not suffer as immensely when observing the weighted averages although decreases of 4%, 1% and 2% respectively.

Although when it came to the sentiment prediction there was no change in performance of the macro average for precision, recall and f1-score as followed 53%, 50% and 51%. This increase remains similar when observing the weighted averages of 55%, 55% and 55% respectively.

Note that the 80/20 split found the optimal alpha to be the activation function, hidden_layer_size and max_iter that were best had values of identity, 5 then 10 nodes and 5 iterations while the 50/50 split found the same hyperparameters when presented with the same options for the emotion models. Since the stochastic gradient descent did not have enough iterations to converge it was eliminated as it never produced a usable classifier here.

Data can be found in the “performance Excel.pdf” or performance.xlsx file.

Note: due to the limit of max_iterations being reduced stochastic gradient was not able to converge at 100 iterations but given enough time and resources may be better than the best estimators proposed currently.

Warning provided below did not result in a pickle file as the model was incomplete.
“C:\Users\chanj\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\neural_network_multilayer_perceptron.py:702: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.”

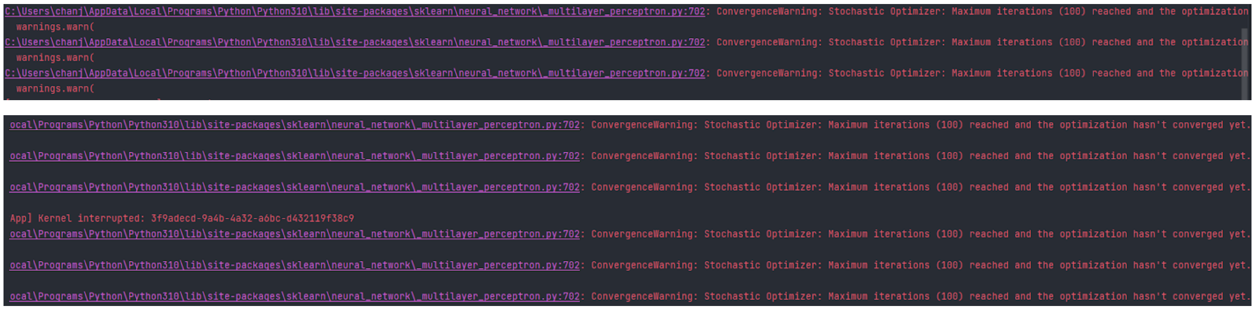

Figure 5: Convergence Errors

Embeddings as Features

Before further analyzing the results from the embeddings as features section, we would like to quickly discuss the models that were used. The first model being used is the word2vec-google-news-300 model which we were told to use. This model creates an output vector with 300 different values based on words that are input. Every word from a post was fed into the model, the values returned for each post were then averaged.

Since the Google model created a vector with 300 values, we wanted to make sure to test use 2 models that would be interesting to compare based on the size of the vector returned. The models that we chose are glove-twitter-100 which creates a vector containing 100 values, as well as glove-wiki-gigaword 300 which creates a vector containing 300 values like the word2vec.

We would like to see if both the Google and Wiki models perform better when compared to the Twitter model due to the vectors having more values allowing for a possibly better classification. Using the Wiki model which contains the same amount of values in its output vector as the Google model will be interesting to compare to see which of those models is the best.

Base MLP (Google, Twitter, Wiki)

The results shown below (for emotions) do not necessarily represent what we had initially thought was going to happen. We had thought that the value of Google and Wiki would have been closer due to the fact that both of their output vectors have 300 values in them compared to Twitter's 100. In reality, Wiki and Twitter scored nearly the same with the Google model being the worst performing model in every category that was measured. One of the biggest differences can be seen in the precision scores. This means that the Google model would guess the same value a repeated amount of times when it was incorrect. This may be linked to the issue that was raised in Table 1. There are posts that are linked to the same emotion, therefore, the model will be training the same values to different emotions. This can result in the values provided to the model being artificially inflated/deflated.

This however doesn't explain the huge difference that the Google and Wiki models have. This could simply be due to the fact that the values provided for the words by the Google model are less accurate than those provided by the Wiki in this specific scenario. This is especially relevant due to the fact that the Google model has almost the exact same values as the other two models when analyzing sentiments.

	Google	**Twitter**	**Wiki**
Score	0.3132840141524606	0.37331772966647164	0.38717506359835085
Accuracy	0.3132840141524606	0.37331772966647164	0.38717506359835085
Precision (Weighted)	0.10991460495201474	0.3636351303597701	0.3662660225736137
F1 Score (Weighted)	0.15242757960216918	0.26656652963011374	0.3054314369601219
F1 Score (Macro)	0.0178362445341089	0.12171919880877288	0.19573170590377284

Table 5: Emotion Comparison for 3 Models using Base-MLP

The table below shows a resurgence of the Google model. From being the worst performer in the previous table to being neck and neck with the Wiki model when checking the F1 scores. When training for sentiments, there is less room for error as there are only a handful of possible sentiments. We can also see how both of the 300 value models performed better than the 100 value model in the case where there are less possible values. There is also a reduction in the same posts having different classifications as shown in Table 1. It does still occur, but not as much allowing for less artificial inflation or deflation of values.

	Google	**Twitter**	**Wiki**
Score	0.5372122129405944	0.49637214745465186	0.5165063305944618
Accuracy	0.5372122129405944	0.5165063305944618	0.5165063305944618
Precision (Weighted)	0.5374237299734715	0.5065184711799884	0.5291221780220821
F1 Score (Weighted)	0.5219671309658458	0.4872978336397582	0.5129582214203394
F1 Score (Macro)	0.46449184791008546	0.4402642610201702	0.47357183479205495

Table 6: Sentiment Comparison for 3 Models using Base-MLP

Top MLP (Google Twitter, Wiki)

Unlike the results shown in the Base MLP models, Google was the best performing model in all categories in comparison to both the Twitter and Wiki models. We can also notice that the metrics for the Top MLP have increased in all models and categories compared to the Base MLP due to it being a better performing multi-layered perceptron. This was the case for both the emotion and sentiment metrics. Given that there is a more equal distribution in the sentiment dataset, values for metrics such as precision increased significantly. This can be seen when comparing Twitter's precision for emotion and sentiment.

While the Google model didn't outperform Twitter and Wiki by large amounts, there are significant differences between the emotion Precision and F1 Score (Macro) values of Google and Twitter. This could possibly be due to the fact that the Google model has 300 values in its output vector in comparison to Twitter only having 100. This means that the quantity of examples in Google outnumbers those found in the Twitter model which would result in better accuracy. This can also be the reason for the metrics in the Wiki model being closer to Google's and having better performance than the Twitter model.

	Google	**Twitter**	**Wiki**
Accuracy	0.3878359099775817	0.34339446053054895	0.35626133239749663
Precision	0.25383071639530835	0.1899110131432744	0.23083496323163072
F1 Score (Weighted)	0.27016455551657853	0.20833619664976433	0.23566305836078766
F1 Score (Macro)	0.1220581025419878	0.058257258816966244	0.08782347405468795

Table 7: Emotion Comparison for 3 Models using Top-MLP

The table below also shows that Google was the best performing model in all categories. Another observation that could be made is the fact that the performance in all categories and models has increased compared to the metrics found in the emotion comparison. As previously stated, this is due to the fact that there are much less sentiments than emotions, which results in a better distribution of results and thus higher scores.

	Google	**Twitter**	**Wiki**

Accuracy	0.528954493842257	0.4918253341522623	0.4970755103234485 4
Precision	0.5286138750581487	0.4876344160220643 7	0.4972254287168954
F1 Score (Weighted)	0.5194281396686699	0.4864398036763785	0.49385696006537
F1 Score (Macro)	0.4679402991192396	0.4537227084025612 4	0.4610463748652832

Table8: Sentiment Comparison for 3 Models using Top-MLP

4.3) In terms of the division of work, Jeffrey Chan was responsible for the execution and analysis of parts 1 and 2 while Maxime Giroux and James Gambino were responsible for the execution and analysis of part 3.