MINI-PROJECT 1

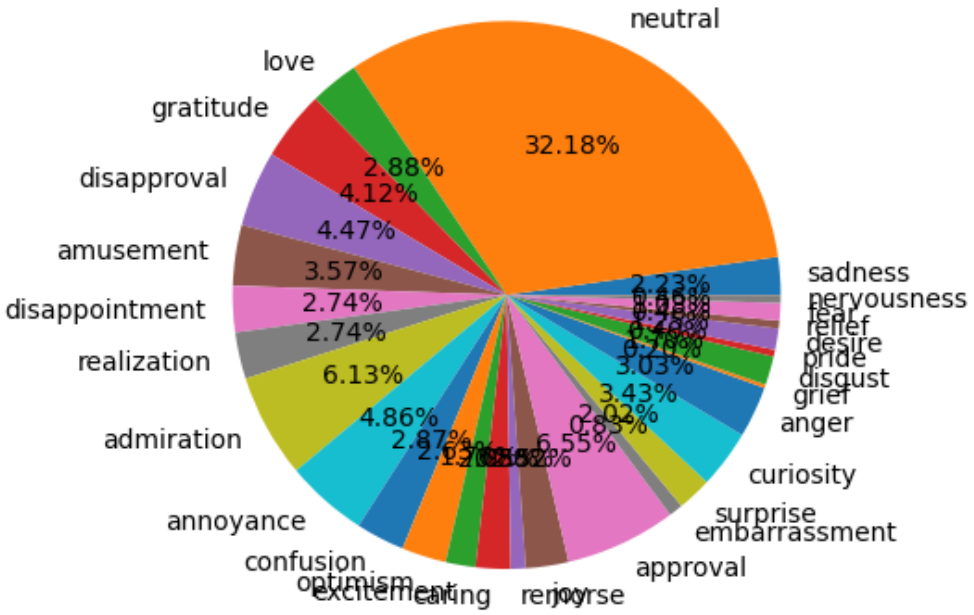
Report
presented to
Dr. Leila Kosseim

By
Krishna Patel 40176352
Rose Rutherford-Stone 27062516
Brianna Malpartida 40045115
COMP 472 section F

Concordia University
22 October 2022

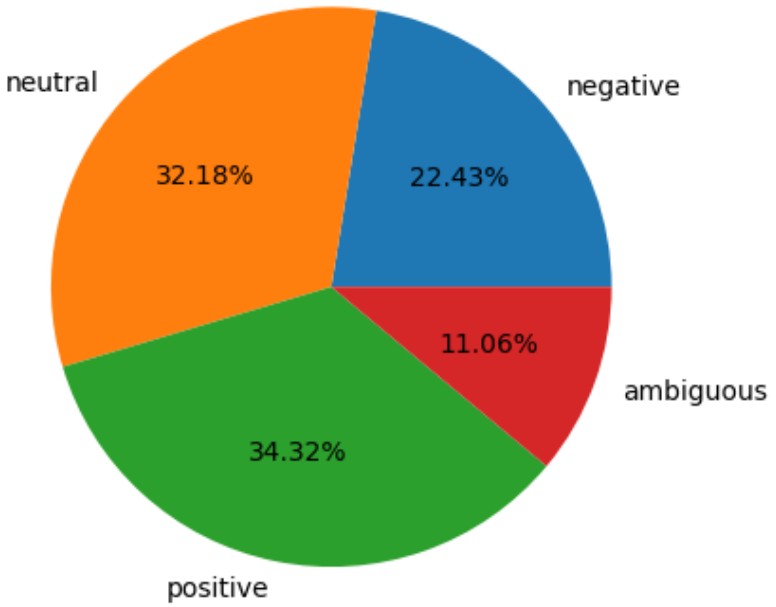
1.Dataset preparation and analysis

1.1.Distribution of emotions



Using matplot library, we were able to graph a pie chart of the distribution of emotions.

1.2.Distribution of sentiments



Using matplot library, we were able to graph a pie chart of the distribution of sentiments from the data set.

1.3. Comments on the choice of metric

As a preliminary analysis. We can see that the emotions metric is very skewed and has a large number of comments marked as ‘neutral’. Although the sentiments also have a big number of neutral comments, the distribution is overall much more even than emotions. As an initial hypothesis, it would be safe to say that sentiments will give us a higher accuracy than emotions because of the difference in distribution. We also noticed how although the data set consists of about 56k unique comments, the size of the dataset amounts to 171,820. This is likely due to comments having multiple labels associated to them, meaning that a single comment is counted many times. We will discuss in a later section the implication of this.

2. Words as Features

2.1. Size of the vocabulary

Size of the vocabulary was found to be 30449.

2.2.

2.3. Classification Results and Parameters

2.3.1. BASE-MNB

2.3.2. BASE-DT

2.3.3. BASE-MLP

2.3.4. TOP-MNB

2.3.5. TOP-DT

2.3.6. TOP-MLP

2.4. Results

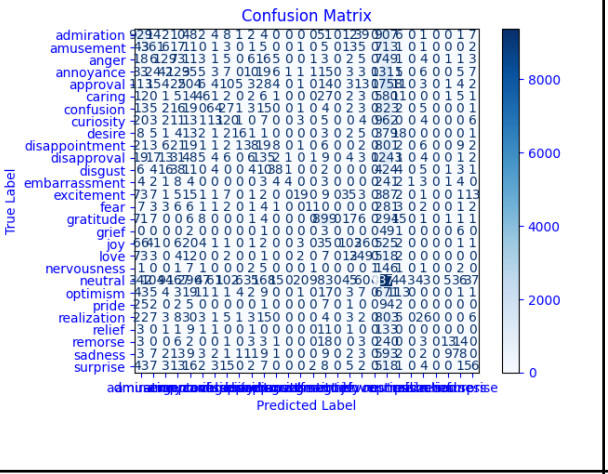
View Performance.txt file for full results. Analysis done in part 4.

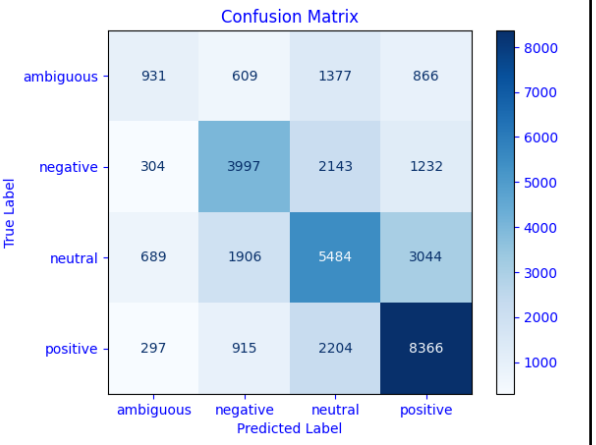
Figure 1. Accuracy Scores per classifier

Class	**BASE-MNB**	**BASE-DT**	**BASE-MLP**	**TOP-MNB**	**TOP-DT**	**TOP-MLP**
Sentiment	0.544436	0.542719	0.570713	0.544611	0.424136	0.565097
Emotion	0.385258	0.355576	0.437580	0.391514	0.393202	0.436009

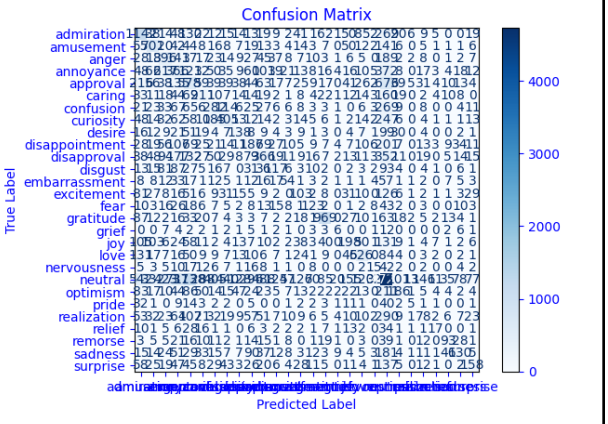
Figure 2. Confusion Matrices for each classifier

EMOTIONS	SENTIMENTS
Base-Mnb	Base-MB

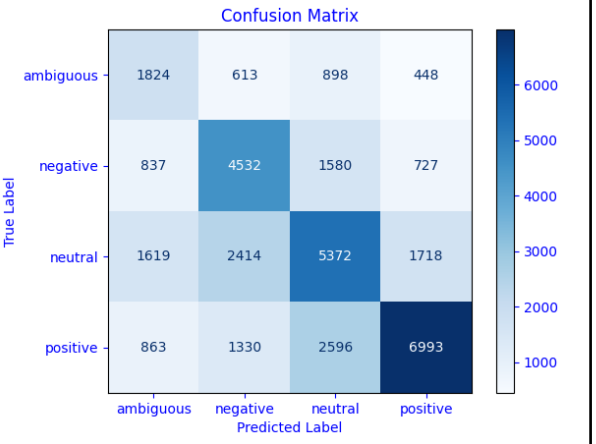




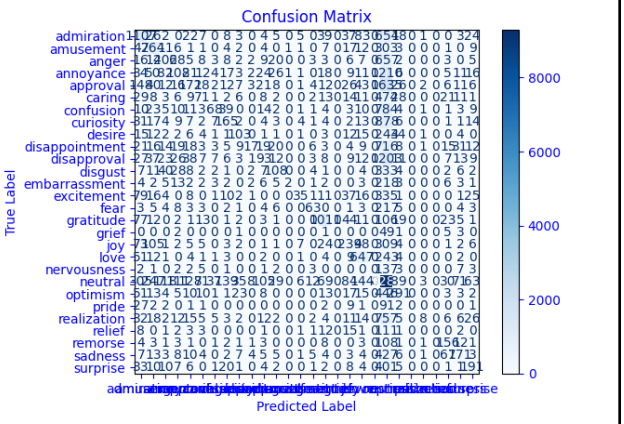
Base-DT



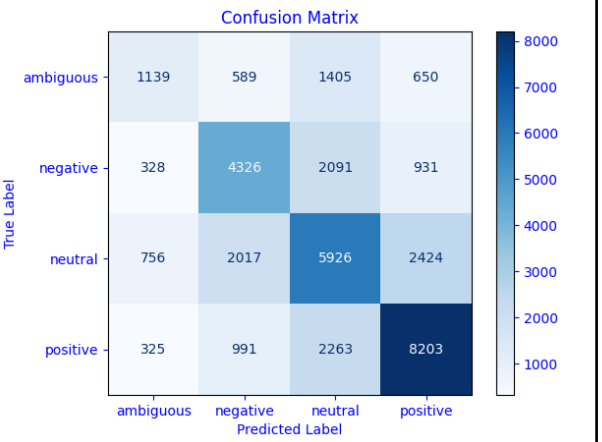
Base-DT



Base-MLP

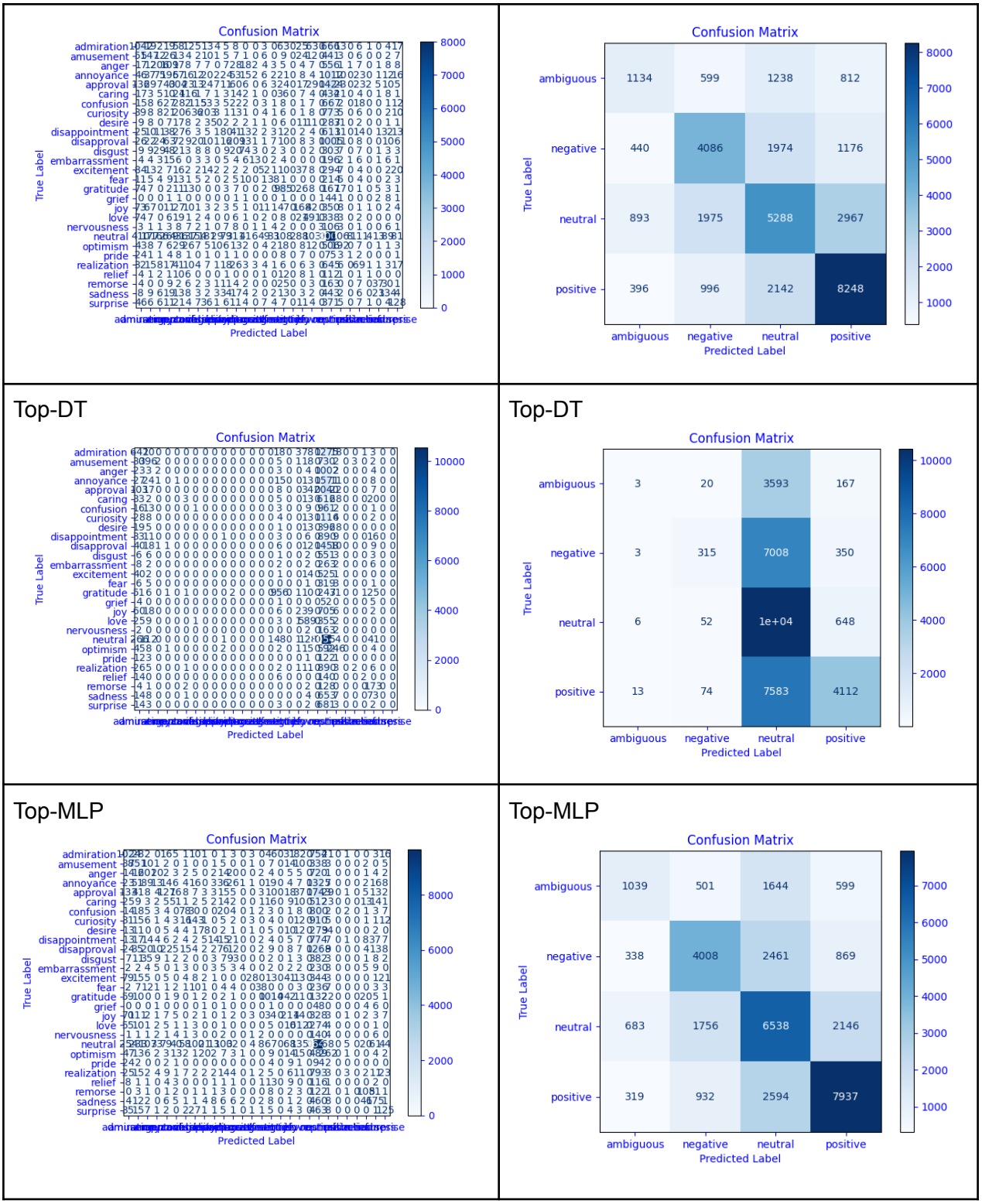


Base-MLP



Top-MNB

Top-MNB



2.5.Own exploration

For our own exploration, we decided to explore the tfidf and see how it compared with word frequencies. We did some research on tfidf, and found out it incorporates not only the text frequency but also the inverse document frequency. So unlike word frequency on its own, it checks how many times the word appears in the sentence, but also takes

the inverse (using log) of how many documents the word appears in. From this an importance weight can be determined, it takes into consideration how often the word occurs, and if it is too frequent or rare, it adjusts the weight.
In the code we used the tfidfvectorizer, this combines the countvectorizer and the tfidftransformer into one.
What we found was that this method performed similarly to the count vectorizer overall in terms of score.

Class	BASE-MNB	BASE-DT	BASE-MLP	TOP-MNB	TOP-DT	TOP-MLP
Sentiment	0.522087	0.539983	0.566959	0.530933	0.426754	0.565941
Emotion	0.345536	0.357292	0.443196	0.386101	0.394395	0.436009

3.Embeddings as Features

3.8 - Own Exploration

- "Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased), `"glove-twitter-25”`
  - Number of of tokens in training set: 2642159
  - Overall hit rate: 0.84
  - As explained later part 4, we were not able to run sklearn classifiers on embeddings as vectors.
- British National Corpus
  - “The British National Corpus (BNC) is a 100 million word collection of samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of British English from the later part of the 20th century, both spoken and written...The written part of the BNC (90%) includes, for example, extracts from regional and national newspapers, specialist periodicals and journals for all ages and interests, academic books and popular fiction, published and unpublished letters and memoranda, school and university essays, among many other kinds of text. The spoken part (10%) consists of orthographic transcriptions of unscripted informal conversations” (source)
  - Assuming the description, we were very surprised when we realized that none of the words from the reddit posts were found in this model. The hit rate being 0% is very unlikely, which means that we either ran into some issues when we downloaded the corpus from http://vectors.nlpl.eu/repository/#, or there is a mistake somewhere in our code.

4.Final Analysis

4.1.

From the first part of the project, we had come up with a hypothesis that the emotions models would perform poorly as opposed to the sentiment models. This can be explained by the fact that the number of posts marked as emotions were strongly skewed towards the 'neutral' label. From the pie chart we can see how the neutral label composed of 32% of the posts, while all other labels were below 7%. Sentiment had a much more evened out number of labels, making it a better classification. From Figure 1, this hypothesis has been proven true. We see how the accuracy scores show that the sentiment models performed 20% better than the emotion models.

Moreover, it was also noted as a preliminary observation that the dataset contained duplicate posts. Meaning that a Reddit comment could appear in the dataset multiple times with different labels. There are about 56k unique comments in the dataset, but the full size amounts to about 171k. This can lead to issues, especially for the sentiments, meaning that a post could be both labelled as positive and negative. In this case, the ambiguous count should have been higher but that is not the case. For emotions, the different labels are understandable due to the large number of emotion labels. A post could be both 'love' and 'admiration for example. Because of the nature of both labels, it is difficult to say if the duplicates in the data has a negative impact on the training of models.

4.2.

We will be comparing the **macro-F1 scores** as the macro-average is the most appropriate score since we assign equal importance to all classes despite an imbalanced dataset. (source)

BASE-MNB VS TOP-MNB

BASE-MNB	**TOP-MNB**
0.17 (emotion)	0.22 (emotion)
0.50 (sentiment)	0.51 (sentiment)

As we can see, the TOP-MNB performed slightly better than the BASE-MNB. The gridsearch parameter for the TOP models were the alpha hyperparameter. Both models for emotion and sentiment used {'alpha': 0.5} as their parameter. The alpha hyperparameter is used for smoothing in the Naive Bayes Classifier. If the alpha parameter is set too high, the model becomes biased towards the class with the most records. Choosing a low alpha parameter will make sure that the model is not under fitted (be a dumb model). This is especially important when the data is skewed as we have noticed for emotions in the previous section. This is the reason why we chose alpha parameters which were low. And by looking at the results of TOP-MNB for both classes, and by comparing them to BASE-MNB we do see a slight increase in

accuracy, which is probably a result of the skewed data, especially for emotion where we noticed a big skew as discussed in 4.1.

BASE-DT VS TOP DT

BASE-DT	**TOP-DT**
0.28 (emotion)	0.13 (emotion)
0.52 (sentiment)	0.27 (sentiment)

Parameters for emotions: {'criterion': 'gini', 'max_depth': 10, 'min_samples_split': 40}
Parameters for sentiments: {'criterion': 'gini', 'max_depth': 10, 'min_samples_split': 10}

From what we can observe, we see that the BASE-DTA performed much better than the TOP-DT models, for both emotions and sentiments.
- Criterion: whether the criterion is gini or entropy, it does not affect the performance of the model. The main difference between both functions being that entropy uses a logarithmic function, which makes entropy slower. But this has no relation to the scores.
-Max_depth: In order to avoid overfitting, we set the values of max_depth pretty low. The choices were either 2 or 10. I believe that if provided more time, we could have seen what an overfitted model would give us as macro-F1 score vs an under-fitted model and tried to figure out which was the optimal max_depth. After printing both the train accuracy scores and the test scores, we noticed that both were very similar. Meaning that max_depth did not cause overfitting. However, it is highly possible that the model was under fitted as the data was very large and that a max_depth of 10 was too low of a number. This could explain the low accuracies.

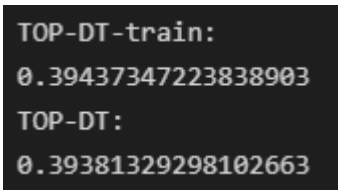


Figure 3. Train score vs Test score for TOP-DT emotion.
-min_samples_split: this hyperparameter can also lead to to underfitting of overfitting depending on the value used.
(source)

BASE-MLP VS TOP-MLP

BASE-MLP	**TOP-MLP**
0.25 (emotion)	0.22 (emotion)
0.53 (sentiment)	0.52 (sentiment)

Parameters for emotions: {'activation': 'tanh', 'hidden_layer_sizes': (30, 50), 'solver': 'adam'}
Parameters for sentiments: {'activation': 'relu', 'hidden_layer_sizes': (30, 50), 'solver': 'adam'}

As we can see the Base-MLP and top-MLP are very close in terms of macro-F1. However, we can see that TOP-MLP performed at a slightly lower rate than BASE-MLP. We believe that this could be explained to the fact that we set the Max_iter of the MLPClassifier to 1, since this classifier took a long time to run on our machines. This lead to our classifier not being able to converge. We also got a warning message about how 20 fits/80 fits failed and were set to NaN. This means that ¼ of the fits were not considered in the calculation of the TOP-MLP scores.

```
20 fits failed out of a total of 80.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.
```

```
C:\Users\Krish\AppData\Roaming\Python\Python310\site-packages\sklearn\neural_network\_multilayer_perceptron.py:702: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1) reached and the optimization
hasn't converged yet.
  warnings.warn(
```

EMBEDDINGS MLP VS WORD AS FEATURES MLP - EMOTIONS

	BASE-MLP	**TOP-MLP**
WORDS	0.44028	0.43604
EMBEDDINGS	0.32184844604818996	0.32184844604818996

EMBEDDINGS MLP VS WORD AS FEATURES MLP - SENTIMENTS

	BASE-MLP	**TOP-MLP**
WORDS	0.567	0.52
EMBEDDINGS	0.3428	0.3428

NOTE: After trying multiple models. We were not able to fully run the MLP models. When they did end up running on our local machines. The numbers for both BASE and TOP were the same as we can see with the tables above. More debugging and research would need to be done in order to see why we are not able to use the embeddings as a vector for the MLP model fit.

4.3.

Team Member Names + ID	**Summary of Contributions**
Krishna Patel + 40176352	- PART 1 - PART 2 (2.3, 2.4) - PART 3 (3.4, 3.5, 3.8) - PART 4 (4.1, 4.2)
Brianna Malpartida + 40045115	- PART 2 (2.3,2.4)

	- PART 3 (3.1, 3.5, 3.8) -
Rose Rutherford-Stone + 27062516	- PART 2 (2.1-2.3.3) - PART 3 (3.1-3.3, 3.5-3.7) - readme.md