COMP 472 Mini-Project 1

Robin Laliberté-Beaupré - 40180181
Sean Mckenzie - 40068618
Hristo Marinov - 40156820

Analysis

4.1 An analysis of the dataset given on Moodle. If there is anything particular about these datasets that might have an impact on the metric to use or the performance of some models (see task 1.3), explain it.

Upon analyzing the data set and determining the classes for the post's emotions and sentiments, it's apparent that the dataset is heavily unbalanced. For sentiments, there is roughly the same amount of "neutral" and "positive" posts, but less "negative" and "ambiguous" posts. The emotion classes are even more unbalanced with neutral "emotions" for posts being the most common and every other emotion class being smaller in comparison.

An unbalanced data set such as this can lead to metrics like accuracy, precision, recall to be more inaccurate as those metrics are best used to evaluate classification models for balanced datasets. This is clear in the results of the models tested where all models suffered from subpar accuracy and no model had any accuracy calculated for emotions being over 50%. The macro and weighted averages were also vastly lowered because of the imbalance with some tests, such as having some of the emotions dataset having a macro average dropping under 20%.

4.2 An analysis of the results of all the models for both classification tasks. In particular, compare and contrast the performance of each model with one another, and with the datasets. Please note that your discussion must be analytical. This means that in addition to stating the facts (e.g. the macro-F1 has this value), you should also analyze them (i.e. explain why some metric seems more appropriate than another, or why your model did not do as well as expected.) Tables, graphs and contingency tables to backup your claims would be very welcome here.

Performance Metrics

Macro vs weighted average:

We notice a large variation between the macro-average and weighted-average metrics for the emotion models. This is noteworthy as it is a direct result of the unbalanced nature of the emotions dataset. If we were to look only at weighted averages, we would get an inaccurate picture of the usefulness of the model as the results are skewed by the high presence of the “neutral” class. It is, therefore, more useful to look at the macro-average for the emotion models as this gives us a better idea of how the models perform for less represented classes.

(There is much less variation between the macro-average and weighted-average metrics for the sentiment models. This is as expected given that this portion of the dataset is fairly well-balanced. However, some slight variation remains due to the over-representation of the “neutral” and “positive” classes.)

Precision vs recall:

We first note that for all sentiment models the results of all metrics are very similar. All the results are also very close to 0.50. We conclude from this observation that the models, when working with a relatively balanced dataset, all make the conservative choice of maximizing recall and precision equally.

In contrast, the models dealing with the emotions dataset all have a (macro-average) precision score significantly higher than the recall score. Given our conclusion from the sentiment models, we know that the classifiers by default do not prioritize one metric over the other. Therefore, the variation between precision and recall must be explained by differences between the datasets. The main difference is the unbalanced nature of the emotions dataset. Precision, therefore, is skewed upwards by the over-representation of one class (neutral). Indeed, the ease with which the models can correctly predict the "neutral" class inflates the precision score. Recall, on the other hand, is skewed downwards by the less represented classes. Since the model is much more likely to guess "neutral" than the other classes, the number of false negatives for the less represented classes will inevitably be very high. This decreases the recall score.

Since there are a lot of classes in the emotions part of the dataset, and only one of them is over-represented, recall seems more appropriate as a metric as it rates the models based on their ability to judge *all* classes without favoring the "neutral" one. However, which metric is more useful ultimately depends on the intended application of the models.

F1-score:

Since precision and recall are weighted equally in calculating the f1-scores, the results for this metric simply follow from our previous observations of precision and recall. In the emotion models, the high number of under-represented classes gives f1-scores closer to the recall score, and much lower than the precision score. In the sentiment models, the balanced nature of the dataset gives f1-scores almost identical to both precision and recall.

Multinomial Naive Bayes (2.3.1, 2.3.3)

Analysis:

In the previous section above about performance metrics, it was noted that the results have been drastically lowered because of the imbalance in data, with the emotions dataset having the most imbalance. This leads to the macro and weighted averages dipping lower than the averages for the sentiments dataset.

Upon comparison with each other, we can confirm that even if the precision for the emotions were less than the sentiment's precision, the weighted recall average is vastly lower and brings down the macro f1-score average the most. The base emotion model ended up being more precise than the top model, but the top had a better recall value. Even if these 2 emotions dataset models had minor differences in their overall metric values, the base and top sentiments were identical after the tests. This is because the GridSearch found an alpha value of 1 to be the best, which is the default for the base model.

Results:

Base Emotions

	precision	recall	f1-score	support
accuracy			0.38	34364
macro avg	0.34	0.14	0.16	34364
weighted avg	0.37	0.38	0.31	34364

Base Sentiments

	precision	recall	f1-score	support
accuracy			0.54	34364
macro avg	0.51	0.49	0.49	34364
weighted avg	0.53	0.54	0.53	34364

Top Emotions

	precision	recall	f1-score	support
accuracy			0.39	34364
macro avg	0.31	0.19	0.22	34364
weighted avg	0.36	0.39	0.34	34364

Top Sentiments

	precision	recall	f1-score	support
accuracy			0.54	34364
macro avg	0.51	0.49	0.49	34364
weighted avg	0.53	0.54	0.53	34364

Decision Trees (2.3.2, 2.3.4)

Analysis:

As expected, the emotions dataset models had objectively lower macro and weighted averages because of the imbalance of classes. Between the base and top decision tree models, some averages were greater than the other while others were lower.

For precision, the top model using the emotions dataset was more precise, with the sentiment dataset being almost identical between base and top decision trees. The recall was slightly smaller for the top sentiments compared to the base, but the biggest difference was with the emotions dataset. The base model had a higher macro average but a lower weighted compared to the top model. The recall scores were greatly smaller than the precision scores for emotions because the overabundance of “neutral” posts lead to more false positives for the other emotion classes in the dataset.

Due to the imbalance of emotion classes and the difference between precision and recall scores, the f1-score was also significantly lower for the models using the emotions dataset.

Results:

Base Emotions

	precision	recall	f1-score	support
accuracy			0.36	34364
macro avg	0.29	0.28	0.27	34364
weighted avg	0.37	0.36	0.36	34364

Base Sentiments

	precision	recall	f1-score	support
accuracy			0.54	34364
macro avg	0.51	0.53	0.52	34364
weighted avg	0.55	0.54	0.54	34364

Top Emotions

	precision	recall	f1-score	support
accuracy			0.42	34364
macro avg	0.37	0.18	0.20	34364
weighted avg	0.40	0.42	0.33	34364

Top Sentiments

	precision	recall	f1-score	support
accuracy			0.52	34364
macro avg	0.51	0.47	0.48	34364
weighted avg	0.54	0.52	0.52	34364

Multi-Layered Perceptron: Words as features (2.3.3, 2.3.5)

Analysis:

The Base MLP was interrupted during fitting in order to get results more quickly. This means that the weights did not have time to be fully adjusted. We can, therefore, assume that the performance is worse than if it had time to be fully fit.

Despite this, the performance obtained is similar to the one obtained by fully running the other models (MNB, DT). This can be clearly seen by comparing the f1-score for all the MLPs with the f1-scores of the other models. More precisely, we notice that the MLPs generally provide better recall scores than the other models. This may be due to the perceptron classifier being better at adjusting for less represented classes.

The Top MLPs were given a low maximum number of iterations (8) when fitting to obtain results more quickly. As with the Base MLPs, this means that the weights did not have time to be fully

adjusted. This likely explains why the performance of the Top MLPs is not significantly better than the performance of the base version.

As with all the models explored, the improvements in performance of the Top model are only meaningful for the emotions portion. This can likely be explained by the hyper-parameters chosen by the Top model being better at dealing with the unbalanced nature of the emotions dataset. Since the sentiment dataset is fairly balanced, the different hyper-parameters had little impact on overall performance.

Overall, what is perhaps most noteworthy is that despite the limitations of our MLP models (interrupts, max iterations), they perform at least as well, and usually better, than the other models. This is true for both the Base and Top versions. This is the expected outcome given that the increased computational completeness of the MLP should, in theory, give better prediction performance than the simpler MNBs and DTs.

Results:

Base Emotions

	precision	recall	f1-score	support
accuracy			0.38	34364
macro avg	0.28	0.28	0.28	34364
weighted avg	0.37	0.38	0.37	34364

Base Sentiments

	precision	recall	f1-score	support
accuracy			0.56	34364
macro avg	0.53	0.52	0.52	34364
weighted avg	0.56	0.56	0.55	34364

Top Emotions

{'activation': 'logistic', 'hidden_layer_sizes': (30, 50), 'max_iter': 8, 'solver': 'adam'}

	precision	recall	f1-score	support
accuracy			0.44	34364
macro avg	0.38	0.25	0.26	34364
weighted avg	0.41	0.44	0.37	34364

Top Sentiments:

{'activation': 'logistic', 'hidden_layer_sizes': (30, 50), 'max_iter': 8, 'solver': 'adam'}

	precision	recall	f1-score	support
accuracy			0.56	34364
macro avg	0.54	0.50	0.51	34364
weighted avg	0.56	0.56	0.56	34364

Stop Words (2.5)

Analysis:

The experiment we chose to explore with was to modify the CountVectorizer to remove all stop words in the English language. The results between the regular tests in 2.3 and the new tests with no stop words were minimal.
Below is an example using the *Multinomial Naive Bayes* models as it showcases that there isn't much of a difference with or without stop words.

Results:

Base Emotions

	precision	recall	f1-score	support
accuracy			0.38	34364
macro avg	0.34	0.14	0.16	34364
weighted avg	0.37	0.38	0.31	34364

Base Emotions (No Stop Words)

accuracy			0.39	34364
macro avg	0.34	0.15	0.18	34364
weighted avg	0.37	0.39	0.31	34364

Base Sentiments

	precision	recall	f1-score	support
accuracy			0.54	34364
macro avg	0.51	0.49	0.49	34364
weighted avg	0.53	0.54	0.53	34364

Base Sentiments (No Stop Words)

accuracy			0.53	34364
macro avg	0.50	0.48	0.48	34364
weighted avg	0.52	0.53	0.52	34364

Top Emotions

	precision	recall	f1-score	support
accuracy			0.39	34364
macro avg	0.31	0.19	0.22	34364
weighted avg	0.36	0.39	0.34	34364

Top Emotions (No Stop Words)

accuracy			0.39	34364
macro avg	0.31	0.20	0.22	34364
weighted avg	0.36	0.39	0.34	34364

Top Sentiments

	precision	recall	f1-score	support
accuracy			0.54	34364
macro avg	0.51	0.49	0.49	34364
weighted avg	0.53	0.54	0.53	34364

Top Sentiments (No Stop Words)

accuracy			0.53	34364
macro avg	0.50	0.48	0.48	34364
weighted avg	0.52	0.53	0.52	34364

Multi-Layered Perceptron: Word embeddings (3.5, to 3.7)

Analysis:

We first note that perhaps the most noticeable change when using word embeddings compared to words as features was the speed at which the model was able to fit itself to the dataset. This had the major benefit of letting us run the models fully, which we could not do with the regular words as features versions.

The second noticeable change is that the models generally performed slightly worse across all metrics when using word embeddings rather than words as features. This is not unexpected given that using the average embeddings of posts rather than literal word frequencies will inevitably cause us to lose some precision. Indeed, in the original words as features version, the MLP was fed a vector with 30000 words. On the other hand, the word embeddings version received a vector with 300 entries. This means that some posts with similar embeddings but different classifications could more easily be misclassified by the model.

Despite this loss in precision, the decrease in performance is marginal, which showcases the strength of the embedding models.

Results:

Base Emotions

	precision	recall	f1-score	support
accuracy			0.41	34346
macro avg	0.36	0.22	0.25	34346
weighted avg	0.38	0.41	0.35	34346

Base Sentiments

	precision	recall	f1-score	support
accuracy			0.54	34346
macro avg	0.51	0.51	0.49	34346
weighted avg	0.53	0.54	0.53	34346

Top Emotions

	precision	recall	f1-score	support
accuracy			0.39	34346
macro avg	0.26	0.13	0.13	34346
weighted avg	0.34	0.39	0.28	34346

Top Sentiments

	precision	recall	f1-score	support
accuracy			0.52	34346
macro avg	0.50	0.46	0.47	34346
weighted avg	0.52	0.52	0.52	34346

Embeddings exploration (3.8)

We tested the models with the following additional embedding models:

- Fasttext-wiki-news-subwords-300
- Glove-twitter-200

Unfortunately, the results were rather underwhelming. Performance of the MLPs remained roughly the same as with the default word2vec-google-news-300 pre-trained model. This remained across all versions of the MLPs. We did notice a slight correlation between the size of the vocabulary and the performance of the models with the emotions dataset. Indeed, the google news model, which has the largest vocabulary of the three (2.8m words), had better overall performance with the emotions dataset than the other two. The other two models, which have around a 1m word vocabulary, had slightly worse performance. This would not be unexpected as smaller vocabularies could give less precise embeddings for the posts, which makes it harder for the model to get accurate predictions.

4.3 In the case of teamwork, a description of the responsibilities and contributions of each team member.

Sean Mckenzie - 40068618:

- Dataset Preparation & Analysis (1.1 - 1.3)
- Decision Trees for Words as Features (2.3.2 and 2.3.5)
- Analysis of Dataset (4.1)
- Quick code reviewing and testing

- Analysis of MNB and DT (4.2)
- Recorded results most models (4.2)

Robin Laliberté-Beaupré - 40180181:

- Base and Top MLP for words as features (2.3.3 and 2.3.6)
- Base and Top MLP for word embeddings (3.5 to 3.7)
- Embeddings exploration (3.8)
- Analysis of Metrics (4.2)
- Analysis of MLPs (4.2)

Hristo Marinov - 40156820:

- Data processing, testing split (2.1 and 2.2)
- Base and Top MNB for words as features (2.3.1 and 2.3.4)
- Loading dataset, cleaning up, tokenizing and computing the embeddings of Reddit posts, as well as displaying token count and hit rates (3.1 to 3.4)
- Analysis editing