Analysis Report

COMP-472 - Fall 2022 - Section

Mini Project 1

Instructor: Dr. Leila Kosseim

Team: K2+ no sushi

Kevin Nguyen(40158397)

The Kien Nguyen(40055738)

Vatanak So(40080207)

Table of contents

1. Analysis of the dataset

Because the data will be classified based on either sentiments or emotions, the analysis will assess the distribution for each type and how their data distribution can impact predictions and performance metrics.

1.1. Emotions

For the emotion category, the dataset is extremely imbalanced. Based on the below histogram, around 32% of the posts are classified as neutral. On the other side of the spectrum, there are classes that occupy less than 1% in the dataset namely “remorse”, “grief”, “pride”, “relief” and “nervousness”. The other classes only account for 5-10% each. Thus, the chance for a popular class to appear can be between 3 and upto 50 times more than a rare class.

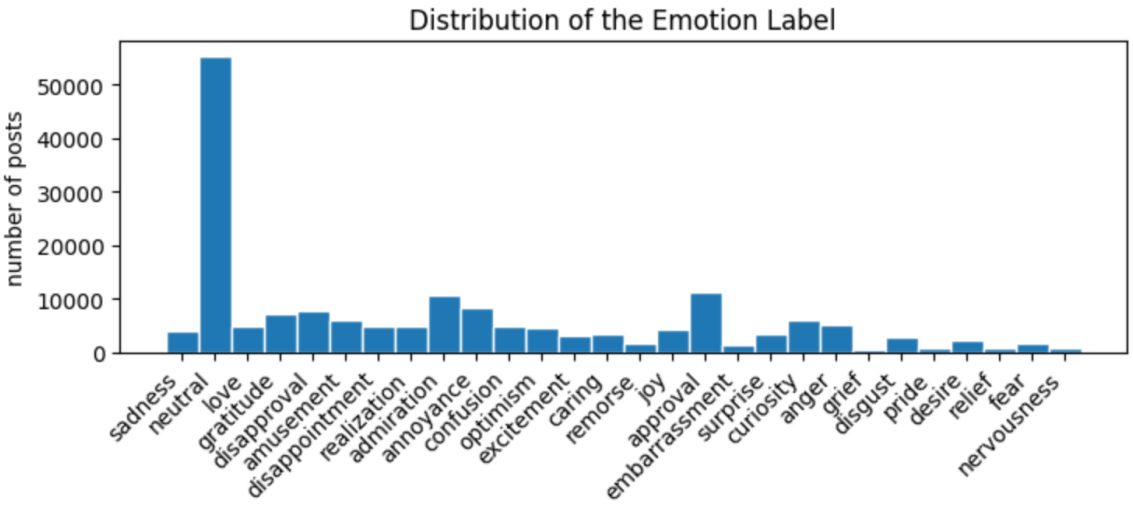

Figure 1. Distribution of emotions in the dataset

```
print(f"number of post {len(posts)}")

count =0
for emo in emotions:
  if emo == "neutral":
    count += 1

print(f"number of neutral post {count}")
print(f"% of neutral post {(count/len(posts)*100)}")
```

```
number of post 171820
number of neutral post 55298
% of neutral post 32.18368059597253
```

Figure 2. A code snippet which shows percentage of neutral posts in the dataset

```
number of relief post 788
% of relief post 0.45861948550808984
```

Figure 3. Distribution of the class “relief” in the dataset

This has a huge effect on how the models are trained. The major issue with this imbalanced dataset is that the models will be trained with a bias to classify posts as “neutral” due to the predominant distribution of the class. Models are more likely to have a high accuracy for “neutral” while that for other classes is modest. Given the small number of posts predicted as non-neutral classes, the precision and recall metrics for those classes will also be affected due to the bias. The recalls for rare classes tend to be very low because models will label most posts as “neutral” so that detecting all instances of a rare class is difficult. The precision should also be lowered, yet it is not as influenced by the imbalance as the recall does. On the other hand, both recall and precision for the popular class such as “neutral” are expected to be high.

1.2. Sentiments

For the sentiments category, the dataset is moderately imbalanced. Based on the below histogram, the top two popular classes are “neutral” and “positive” with around 32% and 34% of the posts respectively. The top two popular classes outnumber the class “negative” and the class “ambiguous” by 10% and 20%. Thus, although the distribution is not as imbalanced as the emotion category, the chance for a popular class to appear is still around 1.5 times and upto 3 times higher than a rare class.

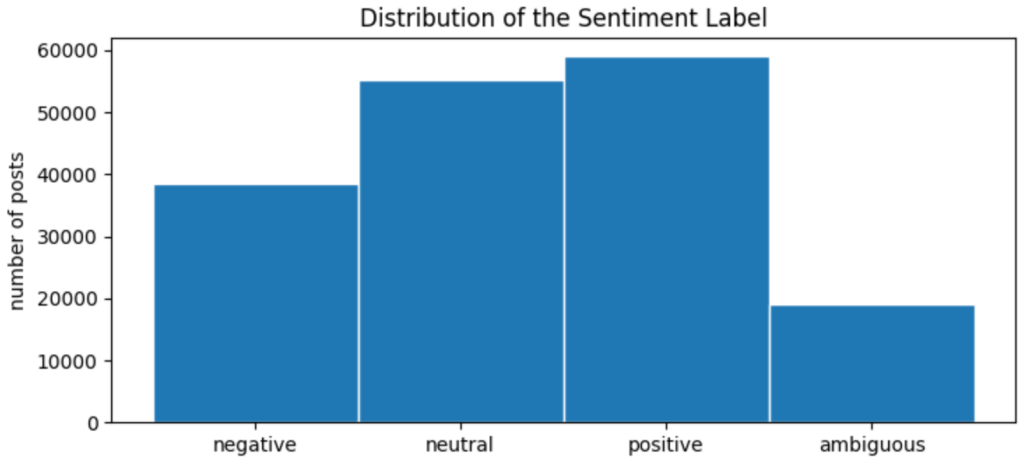

Figure 4. Sentiments distribution in the dataset

```
print(f"number of post {len(posts)} \n")

sen_map = {}
for sentiment in sentiments:
  if sen_map.get(sentiment) is None:
    sen_map[sentiment] = 0

  sen_map[sentiment] += 1

for sentiment in sen_map:
  print(f"number of {sentiment} post {sen_map[sentiment]}")
  print(f"% of {sentiment} post {(sen_map[sentiment]/len(posts)*100)} \n")
```

```
number of post 171820

number of negative post 38545
% of negative post 22.43336049353975

number of neutral post 55298
% of neutral post 32.18368059597253

number of positive post 58968
% of positive post 34.319636829239904

number of ambiguous post 19009
% of ambiguous post 11.063322081247817
```

Figure 5. Code snippet to extract sentiments distribution

The impact of imbalanced distribution on predictions and evaluation metrics are the same as above. However, because the imbalance is less extreme for the sentiment category, the precision and recall for less popular classes should be moderately high and accurate. Thus, the F-1 score for both weighted average and macro average are also expected to be high. The overall trend remains as both recall and precision for the popular class such as “neutral” are expected to be high while those for the less popular classes will be lower.

1.3. Metrics used for evaluation

Macro-average precision is our preferable metric. Since our dataset is imbalanced, we cannot rely on accuracy. The choice between recall and precision is based on our use case. Because the goal is to correctly identify the class for a post rather than find out the most posts for each class, precision is a preferred metric in our evaluation. To evaluate the overall performance, macro-average is preferred to weighted-average. Macro-average, which gives all classes the same weight, will fit our goal better and avoid bias to the extreme classes. Weighted averaged will give a score that relies significantly on the popular classes.

2. Model results analysis

Macro average Emotions

Model name	Macro-precision	Macro-recall	Macro-f1-score
Base MNB	0.38	0.14	0.17
TOP MNB without stopwords - Alpha: 0.5	0.35	0.17	0.20
Base MNB without stopwords	0.34	0.15	0.18
Top MNB - Alpha: 0.5	0.32	0.20	0.23
Base-dt without stopwords	0.30	0.28	0.28
Base DT	0.29	0.27	0.28
Top DT without stopwords	0.16	0.10	0.09
Top DT	0.12	0.10	0.09
Base MLP	0.03	0.03	0.03
Base MLP without stopwords	0.03	0.04	0.03
Base MLP with word embeddings (100 iterations)	0.40	0.20	0.23
Base MLP with word embeddings (100 iterations) - Wiki-300	0.32	0.17	0.20
Base MLP with word embeddings (100 iterations) - Twitter-25	0.32	0.09	0.09
Top MLP	0.01	0.04	0.02

Top MLP without stopwords	0.01	0.04	0.02
Top MLP with word embeddings	0.23	0.14	0.14

Macro-average Sentiments

Model name	Macro-precision	Macro-recall	Macro-f1-score
Base MNB	0.51	0.49	0.49
Base MNB without stopwords	0.51	0.48	0.48
Top MNB Alpha: 0.5	0.51	0.50	0.50
TOP MNB without stopwords - Alpha: 0.5	0.51	0.48	0.49
Base DT	0.52	0.54	0.53
Base-dt without stopwords	0.52	0.53	0.52
Top DT	0.34	0.30	0.23
Top DT without stop words	0.47	0.31	0.23
Base MLP embedding (100 iterations)	0.51	0.50	0.50
Base MLP embedding (100 iterations) - twitter-25	0.44	0.40	0.40
Base MLP embedding (100 iterations) - Wiki-300	0.50	0.46	0.46
Base MLP	0.25	0.25	0.25
Base MLP without stop words	0.25	0.25	0.25
Top MLP	0.09	0.25	0.13
Top MLP without stop words	0.09	0.25	0.13

Top MLP with word embeddings	0.22	0.25	0.20

2.1. Base MNB

2.1.1. Emotions

The macro precision and macro recall of the base MNB model are 0.38 and 0.14 respectively, giving a macro F1 score of 0.17. The scores here for precision and recall seem appropriate as this is just the base model with no additional parameters. Since the dataset has significantly more neutral posts, this has had a large impact on the results from the base MNB model. Due to most of the posts being neutral, the probability of a post being neutral in the total training dataset is much higher. In turn, the model will more often than not label a post as neutral. This is shown in the precision result of neutral posts with a result of 0.38. Only 38% out of 11028 posts labelled as neutral were truly neutral. This in turn has an effect on how the recall results of the other emotions are distributed. While most of the truly neutral posts that were labelled neutral as shown in the recall result of 0.85, the recall results for emotions with at least over 1000 posts such as confusion, approval, or disapproval had only 0.06, 0.09, and 0.08 respectively. This bias towards neutral posts can be explained by the vectorizer and its token frequencies. As taken from the entire posts dataset, words such as ‘it’ and ‘he’ have frequencies of 41895 and 13361 respectively to name a few. These are common words that can be used in pretty much any emotional context. The fact that a third of the posts are neutral and probably use almost all of these common words skews these words towards the neutral emotion. This in turn has an effect on truly labelled posts of different emotions that use these common words. The model will then have a larger probability of mislabelling them as neutral. Removing stop words changes the model only slightly by 1-2%. However, the model still outperforms a lot of the other models tested out. From the table above, the base MNB model is one of the top models in terms of precision and recall.

2.1.2. Sentiments

The macro precision and macro recall of the base MNB model for sentiments are 0.51 and 0.49 respectively, giving a macro F1 score of 0.49. In this dataset, the sentiments are much more equally balanced. Positive, neutral, and negative posts are distributed almost equally, while ambiguous posts take on only an 11th of the testing dataset. However, this distribution seems appropriate. Since the distribution of sentiments in the dataset is much more equal, the sentiments dataset does not share the same issue as that of the emotions dataset. In turn, the sentiment trained model has a more equal distribution of recall and precision results. Removing stop words changes the model only slightly by 1-2%. The model performs well against all other models. It is a contender with base DT and top MNB surprisingly as shown in the above table.

2.2. Top MNB

An analysis for the outputs of the Top-MNB classifiers.

2.2.1. Emotions

With the use of GridSearchCV for the parameter “alpha”, we receive the best value for alpha is 0.5 taken from the input list [0.5,0,0.25,0.75] (the default value for alpha is 1.0). As a result, when encountering an unknown probability, the classifier will assign 0.5 to it.

2.2.2. Sentiments

2.3. Base DT

2.3.1. Emotions

The macro precision and macro recall results were 0.29 and 0.27 respectively, giving a macro F1 score of 0.28. From the classification report, the recall and precision results of all the emotions are represented. None of the results have a score of 0.0. Surprisingly, the recall result of the neutral class with 0.43 is not the highest one even though it is the most overrepresented emotion in the dataset. The title of highest recall result goes to the gratitude emotion class with 0.72. Even with a significantly unbalanced dataset, the DT model seems to still represent most of the classes. In terms of its performance against the other models, its macro F1 score of 0.28 outperforms all other models. Therefore, the base DT model would be the recommended model for this dataset.

2.3.2. Sentiments

The macro precision and macro recall results were 0.52 and 0.54 respectively with a macro F1 score of 0.53. From the classification report, all of the sentiment classes are equally represented and distributed. The ambiguous class which only has 3828 support posts still has a recall and precision result of 0.49 and 0.36 respectively, which is significant because that class accounts for such a small percentage of the entire testing dataset of 34364 posts, being around 11%. These well-distributed results are shown through the macro F1 score of this data model, being 0.53; it has the biggest macro F1 score out of all the other trained models tested. Removing stop words makes insignificant changes to all the results: 0.52 for macro precision, 0.54 for macro recall, and 0.53 for macro F1 score. Base DT is the recommended model for this dataset.

2.4. Top DT

2.4.1 Emotions

The parameters that gave the best results from the top DT model were: {'criterion': 'gini', 'max_depth': 5, 'min_samples_split': 5}. The macro precision and macro recall results were 0.12

and 0.10 respectively, giving a macro F1 score of 0.09. The results for recall and precision are so poor because of the imbalance of neutral labelled posts in the training dataset. What is happening here is the same issue as that of the Base MNB emotions model. Common words used in all types of posts are being given low entropy neutral values, causing significantly increased information gain when labelling a post as neutral. What is strange is that the predictions are distributed to only 7 emotions being: admiration, amusement, gratitude, love, neutral, relief, and remorse. All other emotions' precision and recall results are at 0.0. A possible explanation could be that posts are first filtered as neutral posts, then from that node, only branch out to these 6 other emotions. Nonetheless, the top DT model with these hyper parameters is not viable for this dataset. This model underperforms against all other models except for base and top MLP. Its macro F1 score is incredibly low at 0.09.

2.4.1. Sentiments

The parameters that gave the best results from the top DT model were: {'criterion': 'gini', 'max_depth': 5, 'min_samples_split': 4}. The macro precision and macro recall results were 0.34 and 0.30 respectively, with a macro F1 score of 0.23. The results from this model are very strange. There seems to be no results whatsoever for the negative sentiment. The recall result for the ambiguous class is 0.0, yet it has a precision result of 0.25. The issue might have to do with the hyperparameters taken. Most of the recall results have been focused in the neutral class with 0.95. When removing the stop words, the macro precision and macro recall are 0.47 and 0.31 respectively with the same macro F1 score of 0.23. In the classification report with stop words removed, the recall result of the negative sentiment becomes 0.0. What can be said is that the top DT model is unfit for this dataset as it does not represent the ambiguous and negative classes at all. Its macro F1 score is also one of the lowest out of all the other sentiment trained models.

2.5. Base MLP

Embedding

2 others pre-trained models:

1. glove-wiki-gigaword-300: 'Pre-trained vectors based on Wikipedia 2014 + Gigaword, 5.6B tokens, 400K vocab, uncased, 300 dimension
2. glove-twitter-25:'Pre-trained vectors based on 2B tweets, 27B tokens, 1.2M vocab, uncased

Google-300: Pre-trained vectors trained on a part of the Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases. The phrases were obtained using a simple data-driven approach described in 'Distributed Representations of Words and Phrases and their Compositionality'

Emotions

The maximum iterations was 20 due to power constraint, along with the highly imbalanced dataset, have a huge impact on the performance. This problem can be tackled with more training time as shown with the result of the base-MLP with word embeddings on 100 iterations. Given enough time and power, MLP is one of the best algorithms for classification.

Sentiments

The distribution is relatively more spread out than that of the emotions, making the result significantly better at 25% precision as compared to about 3% for the emotions classification by the same model (without using word embeddings). The precision could be drastically higher at 51% using word embeddings with a much higher number of iterations at 100. It can be concluded that we can achieve a better performance from the base-MLP with a more evenly distributed data and more training time. Techniques such as word embeddings can also help reduce the dimensionality of the features which in turns makes the training more convenient and efficient.

2.6 Top MLP

The following hyperparameters are found using GridSearchCV

Emotions

Default

Hidden layers: (30, 50)
Activation: sigmoid
Optimiser: sgd

Without stop words

Hidden layers: (30, 50)
Activation: sigmoid
Optimiser: sgd

With word embeddings

Hidden layers: (10, 10, 10)
Activation: relu
Optimiser: adam

This algorithm performs the worst amongst all other algorithms. There can be a few reasons for this:

1. There was limited time and computing power upon training, so the maximum iterations was set to 20, rather than the default 100.
2. The activations sigmoid and ReLU suffer from vanishing gradients.

3. The number of hidden layers was set to 2 or 3, which makes the vanishing gradients worse as neurons may be long dead by the time they reach the output.
4. The data is highly imbalanced compared to sentiments, leading to the algorithm classifying the features into only the one class that is the most prevalent (i.e. neutral).

Sentiments

Default

Hidden layers: (30, 50)
Activation: sigmoid
Optimiser: sgd

Without stop words

Hidden layers: (10, 10, 10)
Activation: relu
Optimiser: sgd

With word embeddings

Hidden layers: (10, 10, 10)
Activation: relu
Optimiser: adam

The top-MLP for sentiment classification with word embeddings outperforms the other two methods. This is because the word embeddings shrunk down the feature space to significantly smaller, allowing more efficient training with the same amount of time. However, this model is still significantly behind the other algorithms like DT and MNB because it requires a lot more computing power and time, and the maximum number of iterations was only set to 20.

3. Team member contributions and responsibilities

Team Members	Contributions
Kevin Nguyen(40158397)	- 2.1 - 2.3.1 - 2.3.5 - 2.4.1(Base MNB) - 2.4.5(Top DT) - 2.5.1(Base MNB) - 2.5.5(Top DT) - 3.1 - 3.2 - 3.3 - 3.4
The Kien Nguyen (40055738)	- 2.2 (Data splitting) - 2.3.2 & 2.4.2 & 2.5.2 (Top MNB) - 2.3.4 & 2.4.4 & 2.5.4 (Base DT) - 3.5 (Base-MLP) - 3.7 - 3.8 - 4.1 - Review pull requests and resolve code conflict
Vatanak So (40080207)	- 1.2 - 1.3 - 2.3.3 - 2.3.6 - 2.4.3 (Base-MLP) - 2.4.6 (Top-MLP) - 2.5.3 (Base-MLP) - 2.5.6 (Top-MLP) - 3.6 - readme