COMP3222 Machine Learning Technologies Coursework 2023/24

by Aristotelis Loucaides (al16g20)

Introduction and Data Analysis

With the creation of social media, viral news and controversial information are shared to millions of different users in seconds. When a high impact event takes place, it is hard pinpointing whether it represents real information or even presents real images but in a false context. The aim of this task is to implement a pipeline to classify social media posts from the MediaEval 2015 dataset as real or fake.

Dataset Characterization

Format

The dataset comes with two folders, the training set and the test set formatted in a .txt file. Each line in the text folder has a unique entry, and the columns are separated by tab ('\t'). The data columns for the dataset are: tweetId, tweetText, userId, imageId(s), username, timestamp, label.

#	Column	Non-Null Count	Dtype
0	tweetId	14277 non-null	int64
1	tweetText	14277 non-null	object
2	userId	14277 non-null	int64
3	imageId(s)	14277 non-null	object
4	username	14277 non-null	object
5	timestamp	14277 non-null	object
6	label	14277 non-null	object

Figure 1: Data columns of training set

Volume

The training set consists of 14277 entries and the test set of 3755, combined they have 18,032 different entries.

Quality

There doesn't seem to be an entry with missing values and the data types are appropriate for each column. On the other hand, a big number of tweetText entries appear to be retweets of original posts, seen from "RT" and "via". If our model is too sensitive to duplicate information, or introduces bias, then we should remove these entries completely.

```
UserId: 245699669
TweetText: Insanely beautiful. Wow. RT @AwkwardGoogle: Underwater bedroom. Who wouldn't like to live in a place like this htt
p://t.co/21QjhyLlCy

UserId: 895081556
TweetText: Dad left stranded in South Korea after his toddler doodled over his passport. via /r/funny http://t.co/TvCktsNb1I ht
tp://t.co/YUSKQW78dD
```

Figure 2: Reposted tweets

When we detect the language for each tweet using the langdetect library we get the following results:

```
{'en': 10950, 'es': 1293, 'tl': 321, 'fr': 213, 'id': 179, 'pt': 167, 'cy': 125, 'so': 122, 'de': 122, 'it': 96, 'nl': 86, 'a
r': 80, 'af': 78, 'ru': 60, 'sv': 42, 'pl': 38, 'ca': 35, 'no': 34, 'tr': 34, 'da': 22, 'ja': 20, 'th': 18, 'sk': 13, 'vi': 13,
'et': 13, 'sw': 11, 'fi': 11, 'sl': 10, 'sq': 9, 'zh-cn': 9, 'bg': 9, 'ko': 7, 'lt': 7, 'ro': 7, 'hu': 5, 'el': 5, 'hr': 4, 'f
a': 3, 'he': 1, 'cs': 1, 'Unknown': 1, 'mk': 1, 'hi': 1, 'lv': 1}
```

Figure 3: Languages detected and frequency in training set

Depending on our model the language diversity can be both a strength and a challenge. In the case of MediaEval 2015 dataset, the languages used are coefficient to the event covered, for example in the event ‘malaysia’ a big number of tweets are written in Indonesian (which is a language spoken geographically close to this high compact event). Thus language-specific analysis might be necessary for the context and labeling value of the event, or it can cause an imbalance which might impact the performance of the model.

There is a label discrepancy in our ‘label’ column, where the training set includes the values ‘fake’, ‘real’ and humor, whereas the test set only has ‘fake’ and ‘real’.

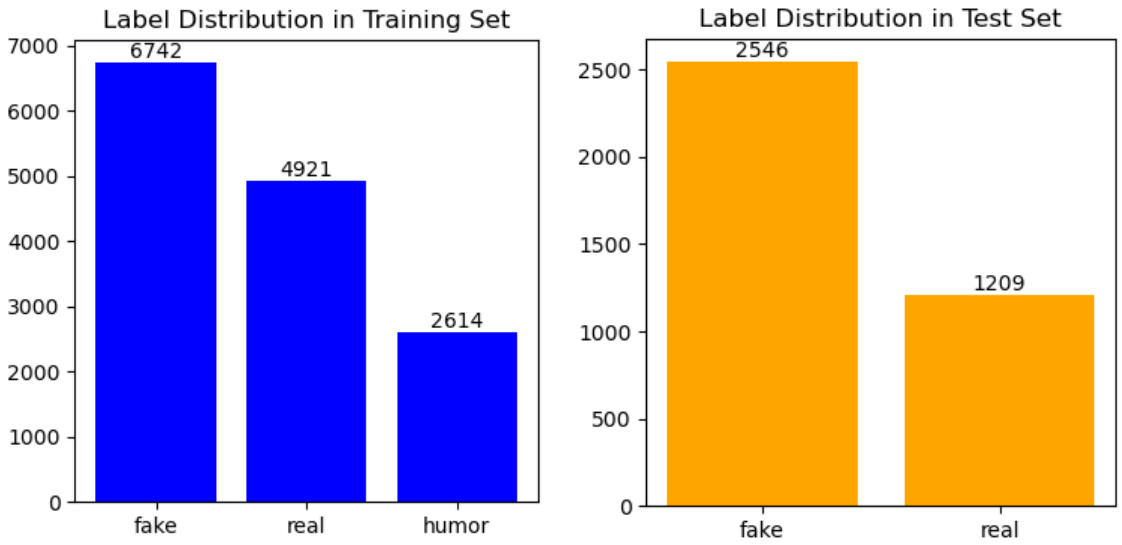


Figure 4: Label distribution in dataset

These inconsistencies can introduce challenges during the evaluation step as the model may not recognise the humor category during the training phase. As tweets labeled as ‘humor’ seem to be satirical and not essentially representative of the truth, they can be potentially labeled as ‘fake’.

There are other features to consider, such as emojis, links and whitespaces, these will be mentioned and tackled accordingly in the data pre-processing section.

Bias

We can identify bias in our dataset as certain events are disproportionately represented or underrepresented.

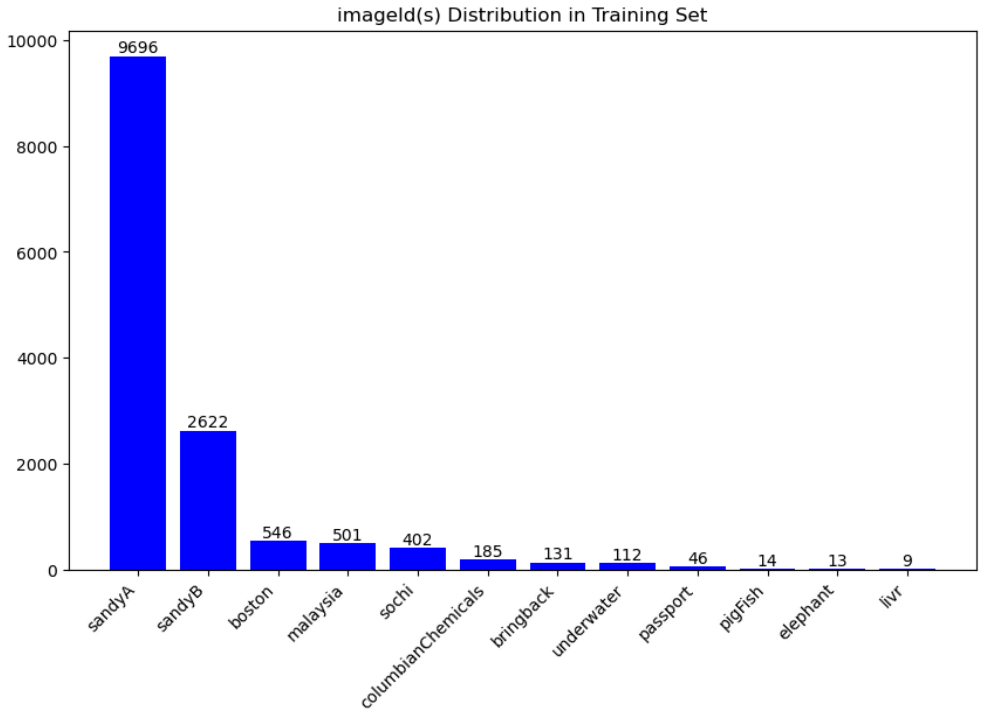


Figure 5: imageId(s) distribution in training set

As can be seen from the graph, “sandyA” and “sandyB”, which both cover the event of hurricane Sandy, have a total number of 12318 posts. Comparing this number to the total number of entries in our training set, 86% of our data only covers the event about hurricane Sandy.

In contrast, the events “passport”, “pigFish”, “elephant” and “livr” only amount to less than 1% of the represented data each.

Pipeline Design

Dataset Pre-processing

The first step was to turn the humor values in our training set to fake. The test set didn’t have such values in its label column and as humorous posts don’t necessarily represent the truth they were labeled as fake.

Whatever tweet included "(via @\w+)|(via+\s+/r/[A-Za-z]+)|(via \w+)|(@\w+)|RT|&amp" were removed from the tweet. Links, emojis and newline characters were also removed from tweetText. The goal was to only have words and phrases that indicated towards the event and would provide value to the classification algorithm.

As a final step, all the whitespace was removed from the text and the text was turned to lowercase.

Feature Selection

After going through the whole process of our pre-processing, the decided feature for our machine learning algorithm is ‘tweetText’. The target values will be the ‘label’ data column. ‘tweetText’ provides important context and keywords that showcase whether an event is possible real or fake, thus was the choice for our selected feature.

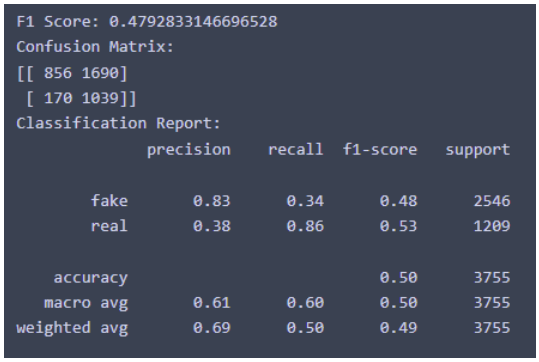
Machine Learning Algorithm

After cleaning our dataset, the next step is to choose our text vectorization technique. Its purpose is to convert text to a format that can be processed efficiently by machine learning algorithms [1]. Our first choice of vectorization method is Term Frequency-Inverse Document Frequency (TF-IDF) and the second Bag of Words (BoW). These two seem to perform well under logistic regression [1], which is our case as we will have to label as either real or fake.

For the text classification, Multinomial Naive Bayes and Bernoulli Naive Bayes are considered. After making the choice of vectorization techniques, we will test and compare which text classification performs better. “Multinomial Naïve Bayes classifier works on the concept of term frequency which means that how many times does the word occur in a document.”. “On the other hand, Bernoulli Naïve Bayes Classifier works on the binary concept that whether the term occurs in a document or not but unlike Multinomial Naïve Bayes, it does not tell about the term frequency.” [2].

Evaluation

The first step, we tested Multinomial Naive Bayes with BoW. The model has a high recall for real tweets (0.86), but low recall for fake tweets (0.32). The f1 score is significantly low at 0.47, which shows there is significant improvement to be made.



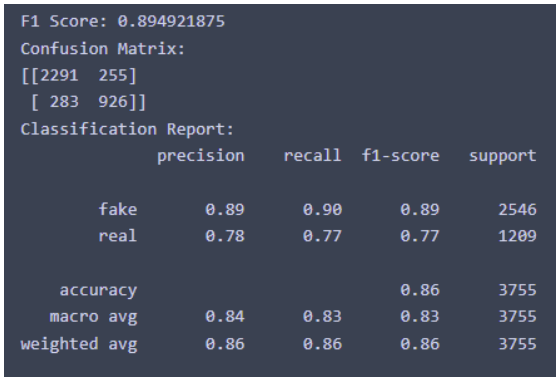
```
F1 Score: 0.4792833146696528
Confusion Matrix:
[[ 856 1690]
 [ 170 1039]]
Classification Report:
              precision    recall  f1-score   support

        fake       0.83      0.34      0.48      2546
        real       0.38      0.86      0.53      1209

    accuracy                           0.50      3755
   macro avg       0.61      0.60      0.50      3755
weighted avg       0.69      0.50      0.49      3755
```

Figure 6: MultinomialNB with BoW f1 score and confusion matrix

Next, we changed our vectorization technique to TF-IDF. This made a significant improvement. Our f1 score jumped to 0.89, indicating a strong ability to discriminate between real and fake, the model showed high recalls for both real and fake.



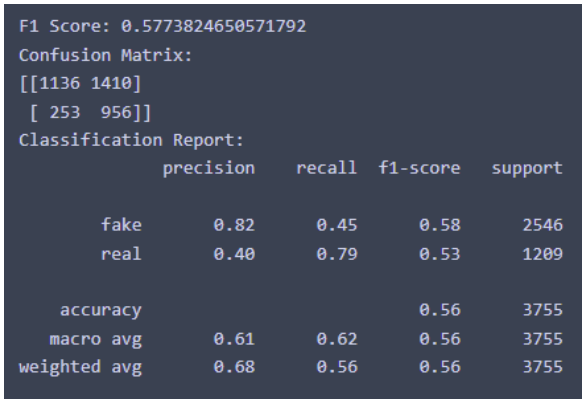
```
F1 Score: 0.894921875
Confusion Matrix:
[[2291  255]
 [ 283  926]]
Classification Report:
              precision    recall  f1-score   support

        fake       0.89      0.90      0.89      2546
        real       0.78      0.77      0.77      1209

    accuracy                           0.86      3755
   macro avg       0.84      0.83      0.83      3755
weighted avg       0.86      0.86      0.86      3755
```

Figure 7: MultinomialNB with TF-IDF f1 score and confusion matrix

G. Singh et al. [2] research resulted that Multinomial and Bernoulli Naive Bayes perform not much differently given the same dataset, which is why Bernoulli was chosen as a possible classification method in this project. Suprisingly, The results were not very promising, the f1 score achieved was only 0.57.



```
F1 Score: 0.5773824650571792
Confusion Matrix:
[[1136 1410]
 [ 253  956]]
Classification Report:
              precision    recall  f1-score   support

        fake       0.82      0.45      0.58      2546
        real       0.40      0.79      0.53      1209

    accuracy                           0.56      3755
   macro avg       0.61      0.62      0.56      3755
weighted avg       0.68      0.56      0.56      3755
```

Figure 8: BernoulliNB with TF-IDF f1 score and confusion matrix

Conclusion

Findings

The TF-IDF vectorization technique heavily outweighs the increase in performance compared to Bag of Words when used with Multinomial Naive Bayes.

Areas for improvement

At the early stages of the project, dimensionality reduction was tried but was too computationally expensive. Even though preprocessing was done thoroughly, there can still be made improvements.

Lessons learned

The choice of text vectorization technique can have a significant impact on the performance of the text classification algorithm. Although achieving promising results with one of the algorithms, there is still room for improvement, shown by the other cases where certain methods ded not meet expectations.

References

[1] A. Verma, N. Goyal, P. Bansal and P. Gambhir, "Comparing the performance of various Encoder Models and Vectorization Techniques on Text Classification," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-7, doi: 10.1109/ICCCNT56998.2023.10307214.

[2] G. Singh, B. Kumar, L. Gaur and A. Tyagi, "Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification," 2019 International Conference on Automation, Computational and Technology Management (ICACTM), London, UK, 2019, pp. 593-596, doi: 10.1109/ICACTM.2019.8776800.