COMP3222

MACHINE LEARNING TECHNOLOGIES

COURSEWORK REPORT

February 5th, 2024

ab3u21

Introduction and data analysis

This report focuses on the binary classification of tweet posts provided by a slice of the MediaEval dataset, where labels refer to the veracity of tweets and their images. To separate tweets into different classes, tweet attributes such as including text, tweet metadata, and images are considered. For the sake of this objective, let us first understand the dataset, its characteristics, features, and data. This process will help us understand what pipeline would be most effective to classify the data.

Format

Training and testing datasets are provided as tab separated value (.tsv) files, where columns are separated by tabs. Importing the datasets, they can be read and treated as comma separated value files (.csv) and translated to a Data Frame using the Pandas library. Figure I shows the datatype, value count, and null value count for each feature in the datasets. We notice that we have two numerical features, while others are marked as objects. We will have to numericize other features should we consider using them. We can also infer the shape of these datasets to be (14276, 7) and (3755,7), and neither of them contain null values.

Training

```
RangeIndex: 14277 entries, 0 to 14276
Data columns (total 7 columns):
 #   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
dtypes: int64(2), object(5)
memory usage: 780.9+ KB
```

Testing

```
RangeIndex: 3755 entries, 0 to 3754
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   tweetId     3755 non-null   int64
 1   tweetText   3755 non-null   object
 2   userId      3755 non-null   int64
 3   imageId(s)  3755 non-null   object
 4   username    3755 non-null   object
 5   timestamp   3755 non-null   object
 6   label       3755 non-null   object
dtypes: int64(2), object(5)
memory usage: 205.5+ KB
```

Figure I – Data format of training dataset

Figure II provides insight on the content of each feature. We notice that the 'tweetText' column, , is very noisy as it contains links, hashtags, emojis, and special characters. We can also notice that the timestamp, could be converted to a Date Time format and numericized for a Time-Based analysis, which (Shahid et al., 2022) identify as a valuable strategy to identify fake news.

	tweetId	tweetText	userId	imageId(s)	username	timestamp	label
0	263046056240115712	¿Se acuerdan de la película: "El día después d...	21226711	sandyA_fake_46	iAnnieM	Mon Oct 29 22:34:01 +0000 2012	fake
1	262995061304852481	@milenagimon: Miren a Sandy en NY! Tremenda i...	192378571	sandyA_fake_09	CarlosVerareal	Mon Oct 29 19:11:23 +0000 2012	fake
2	262979898002534400	Buena la foto del Huracán Sandy, me recuerda a...	132303095	sandyA_fake_09	LucasPalape	Mon Oct 29 18:11:08 +0000 2012	fake

Figure II – Head of training dataset

Volume

Volume and occurrence statistics for the training and testing dataset can be seen in Figure III. In total we have a corpus of 18032 documents. We notice that the training set holds 1901 non-unique tweets, while the testing dataset holds 49 non-unique tweets, though many more differ just slightly due to retweets. (Shahid et al., 2022) identify duplicate entries as indicative of misinformation campaigns and coordinated attacks, hence, text frequency will be later considered as a feature for classification to also avoid overfitting our models.

Training

	tweetText	imageId(s)	username	timestamp	label
count	14277	14277	14277	14277	14277
unique	12376	377	13498	13909	3
top	Unbelievable scene flying over #StatenIsland i...	sandyA_fake_29	SAGandAFTRA	Tue Oct 30 00:31:14 +0000 2012	fake
freq	42	1100	16	4	6742

Testing

	tweetText	imageId(s)	username	timestamp	label
count	3755	3755	3755	3755	3755
unique	3706	88	3553	3449	2
top	J'aime une vidéo @YouTube : "SYRIA! SYRIAN HER...	syrianboy_1	_WTFVideos	Sat Apr 25 18:05:05 +0000 2015	fake
freq	4	1769	23	13	2546

Figure III – Countable frequencies of datasets

Quality

The multilingual features in the MediaEval dataset can be used to predict future fake posts in languages other than English. Figure IV shows the distribution of different languages. Both datasets are plotted as bar charts to compare their distributions. We can see a negligible number of entries are in different languages.

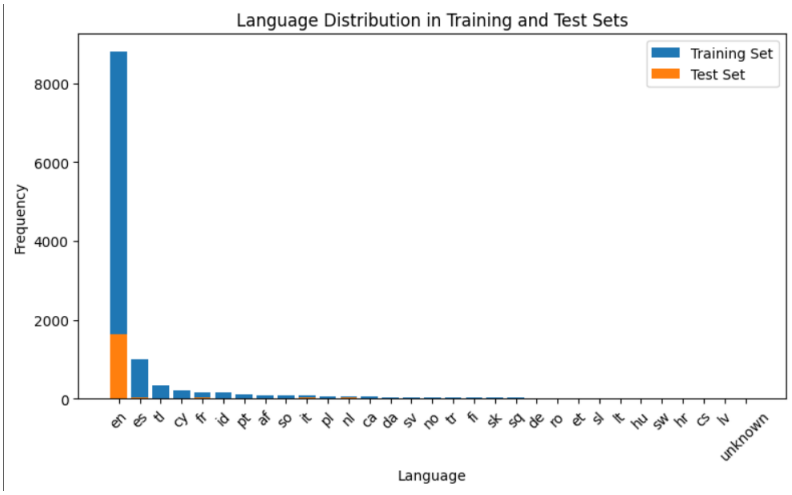


Figure IV – Language distributions in training and test datasets

Label values are three in the training dataset, which include the 'humour' category. However, the scope of this project is to perform binary classification, and thus we will join the fae and humour labels. Figure V shows the disparity between labels through pie charts. After merging, we see ~65% of our dataset is fake labels. This might bias our models towards future data, however, we don't know the full distribution of real/fake data on twitter in 2015, we can only assume there are more fake tweets than real ones.

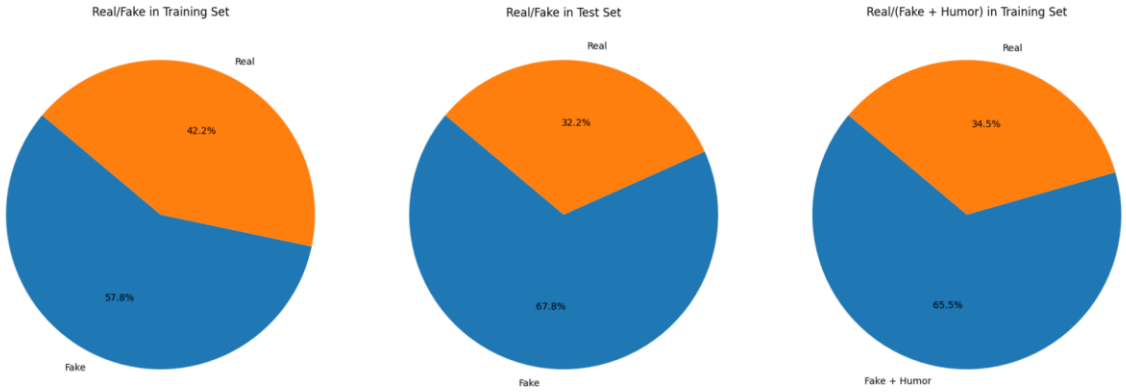


Figure V – Label proportions in training and test datasets

Lastly, let us consider is average tweet length and its distribution. Figure VI shows the distributions of tweet length after tokenization for both datasets. It is interesting to note how some datapoints in both sets have lengths greater than 140, which was the character limit for Twitter posts in 2015 (Alex Hern, 2015) which points to posts being fake. In fact, document length is another content-based feature used in fake news detection(Capuano et al., 2023).

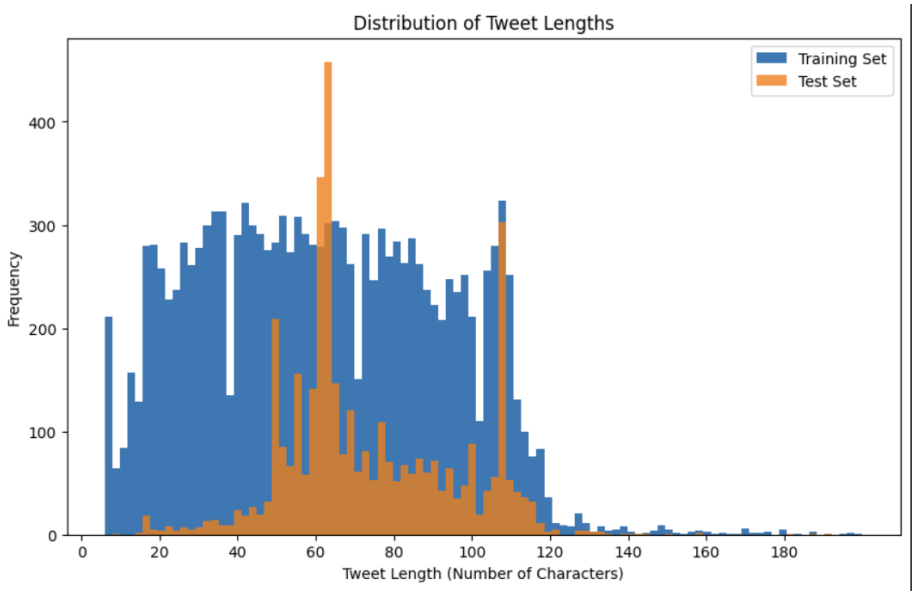
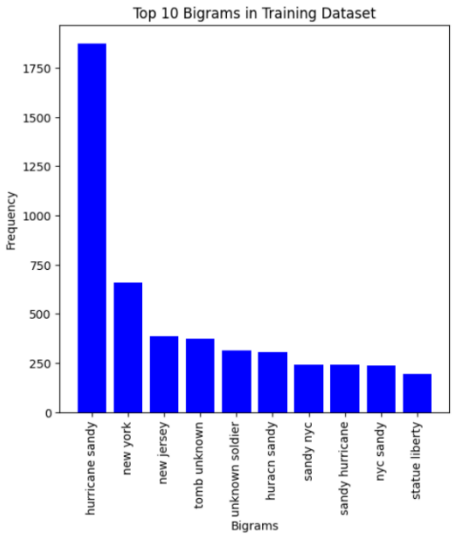


Figure VI – Distribution of tweet lengths

Bias

The training dataset appears to be heavily biased towards specific keywords. Figure VII shows a disproportionate majority of tweets talk about hurricane sandy, while the test dataset has a more even spread of bigrams in Figure VIII. This indicates that the models will need dimensionality reduction techniques and feature engineering to avoid overfitting.



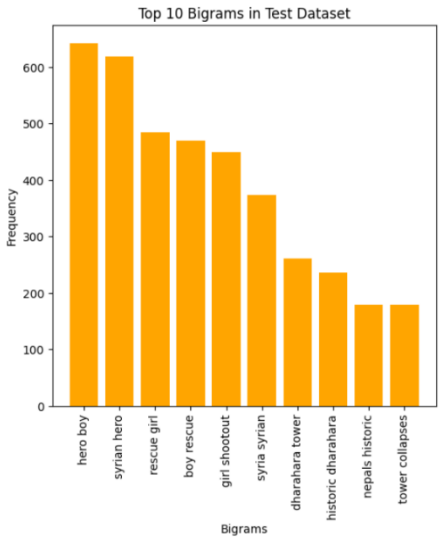


Figure VII – Top 10 bigrams in training dataset

Figure VIII – Top 10 bigrams in testing dataset

Pipeline design 1

The first pipeline uses a SVM classifier to classify tweets based on the following features: TF-IDF, user ID, tweet length, frequency of occurrence, and a rudimentary sentiment analysis score.

Data cleaning and preprocessing follows this pipeline:

1. Datasets are imported as .txt files and interpreted using the pd.read_csv function, whereupon they are transformed into data frames for ease of processing.
2. After data characterization text is converted to lowercase and cleaned of noise: URLs, newlines, double whitespaces, user references, and punctuation are removed, while emojis are translated to a few content words in English.
3. Text is then tokenized by splitting on whitespace characters.
4. Stop words are removed to preserve content words.

Further data characterization is carried out to investigate tweet length distributions and training dataset bias. To front this problem the training dataset distribution has been changed to only maintain the tweets with lowest cosine similarity. This entails that each of the selected tweets has a

high variance over the target variable, and it prevents our model overfitting to the bias seen in Figure IV.

Feature engineering has been carried out in the following way:

1. A TF-IDF sparse matrix is generated representing the number of times each token appears in each document. The matrix is then condensed following two criteria: it should represents up to 200 dimensions, and only allows vector representations of text which do not appear in more than 80% of the corpus. This should dampen the bias we discussed in the previous section.
2. The Support Vector Machine created works with a Radial basis Function kernel which allows it to consider multiple dimensions when drawing a separation hyperplane. This is done in conjunction with fine-tuned parameters.

Pipeline design 2

The second pipeline uses n-grams of words, tweet length, tweet frequency and a sentiment score as features to make a classification. The pipeline is identical until after the data cleaning section. It differs in the following steps:

1. During further Exploratory Data Analysis, tweet lengths and frequencies are calculated.
2. These metrics are turned into features during the Feature Engineering Stage and are subsequently normalized from -1 to 1 values.
3. To prevent training dataset bias from affecting the classifier, the majority class of false tweets is under-sampled, removing without replacement all excess documents that contributed to a disproportion between real and false classes in the training dataset.
4. Features are reshaped into column vectors to be processed and the model makes its prediction.

Evaluation

Performance metrics and relative confusion matrices are visible in Figure IX and X. We can observe that the models provide a satisfactory result: they outperform a weak classifier and function according to the scope of this project.

One consideration to be made is that, while both models work on the test dataset, there is evidence to indicate that the Logistic Classifier might perform better because it was trained on an even distribution of target variables, and thus was not biased by their proportion. On the other hand, the Logistic Classifier was more complicated to program and fine-tune due to the many features it was relying on.

By the same token, it seems the SVM learned the data distribution better than its logistic counterpart, but it is still unclear to me if it would outperform it on a differently distributed dataset.

Figure IX – Performance metrics for SVM classifier

```
Precision: 0.7200321802091714
Accuacy: 0.8237017310252996
Recall: 0.7402812241521919
F1 Score: 0.7300163132137031
True Negatives (TN): 2198
False Positives (FP): 348
False Negatives (FN): 314
True Positives (TP): 895
```

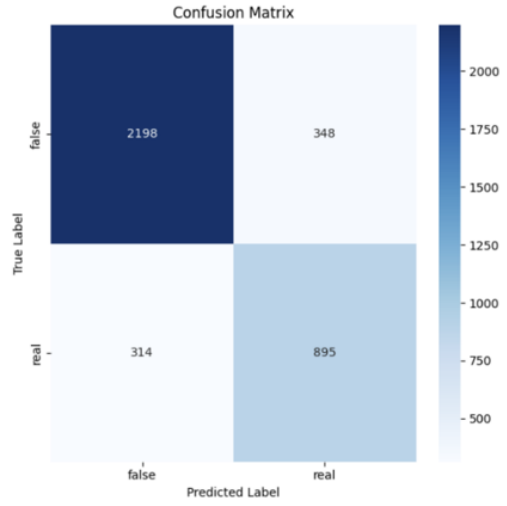
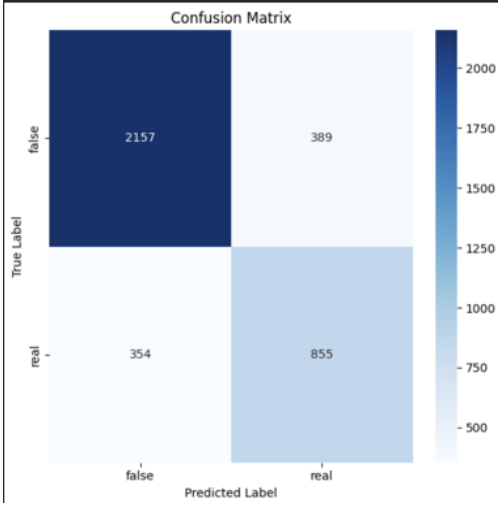


Figure X – Performance metrics for Logistic Classifier

```
Precision: 0.6872990353697749
Accuacy: 0.8021304926764314
Recall: 0.707196029776675
F1 Score: 0.6971055849979617
True Negatives (TN): 2157
False Positives (FP): 389
False Negatives (FN): 354
True Positives (TP): 855
```



Conclusion

Last considerations go towards the possibility to better curate data, by translating it for example, and by inferring new features from the datasets. In particular, it would have been interesting to merge the TF-IDF matrix with the features generated for the Logistic Classifier, which might have yielded a more effective model.

Lastly, other features such as POS tagging and more advanced Word Embedding models, such as Word2Vec would have surely increased performance in both models(Middleton, n.d.).

References

Alex Hern. (2015, June 12). Twitter will remove 140-word character limit in direct messages. *The Guardian*.

Capuano, N., Fenza, G., Loia, V., & Nota, F. D. (2023). Content-Based Fake News Detection With Machine and Deep Learning: a Systematic Review. In *Neurocomputing* (Vol. 530, pp. 91–103). Elsevier B.V. https://doi.org/10.1016/j.neucom.2023.02.005

Middleton, S. E. (n.d.). *Extracting Attributed Verification and Debunking Reports from Social Media: MediaEval-2015 Trust and Credibility Analysis of Image and Video*.

Shahid, W., Li, Y., Staples, D., Amin, G., Hakak, S., & Ghorbani, A. (2022). Are You a Cyborg, Bot or Human?—A Survey on Detecting Fake News Spreaders. *IEEE Access*, *10*, 27069–27083. https://doi.org/10.1109/ACCESS.2022.3157724