COMP3222 Machine Learning Technologies Coursework

Tom Evans

January 2023

1 Introduction

In today's digital era, identifying misinformation is more vital than ever. With the relentless spread of fake news and posts on social media, the ability to use modern and advanced machine learning algorithms to distinguish between truth and deception is of great importance.

2 Problem Overview

This project aims to design and implement two machine-learning pipelines to classify posts within the 'MediaEval 2015 "verifying multimedia us" challenge dataset' (MediaEval 2015) as real or fake.

A fake post is defined as the following:

- Reposting of real multimedia, such as real photos from the past re-posted as being associated with a current event.
- Digitally manipulated multimedia.
- Synthetic multimedia, such as artworks or snapshots presented as real imagery.

The performance of these pipelines will be evaluated using the F1-Scores obtained after the algorithm has been trained on the training dataset and predicted on the test dataset.

3 Data Characterisation and Analysis

This section will analyse the contents of the MediaEval 2015 dataset looking to find patterns and trends that can be considered when designing the pipeline.

The MediaEval 2015 training and test set consist of social media posts with a 'real', 'fake' or 'humour' label. In this project, the humour label will be treated as fake making it a binary classification problem with 2 classes.

Both the training and test set contain 7 columns:

- tweetId
- tweetText
- userId
- imageId(s) - renamed to imageId for easier manipulation
- username
- timestamp
- label

There are 14483 entries in the training set and only 3781 entries in the test set, making the training set 3.83x larger. This also means that there are 18,264 total posts in the MediaEval 2015 dataset. However, of the 14483 posts in the training set, 1941 (13.4%) are duplicates (based on tweetText alone). There are 51 (1.34%) duplicates in the test set.

3.1 Events

The event to which the tweet refers is stored in the imageId column. As can be seen in Figure 1, the majority of tweets in the training set refer to the event 'SandyA' (9860) followed by 'SandyB' (2663) which - after looking at corresponding tweetText and timestamps - seem to be referring to Hurricane Sandy. The next highest is event 'boston' (546), assumingly regarding the Boston Marathon bombings all 'boston' tweets are from the month this happened (April 2013). The majority of tweets in the test set refer to the 'syrianboy' event (1786) and 'nepal' (1360) (Figure 1). Given the timestamp and tweetText, these seem to be referring to a video of a Syrian Boy saving a little girl (this video was confirmed to be fake [1]) and the Nepalese Earthquake in April 2015.

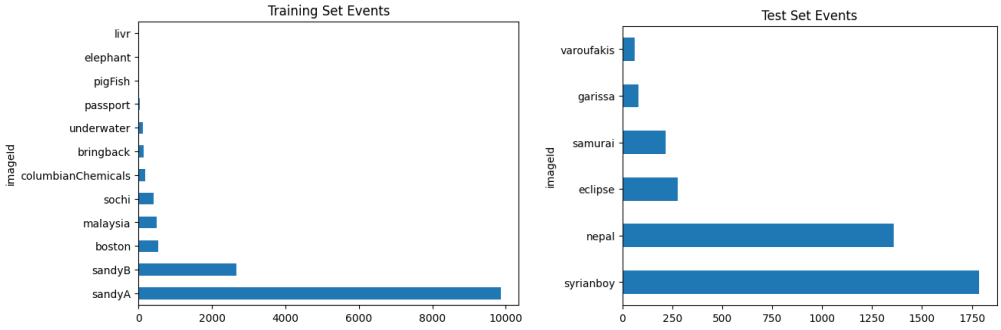

Figure 1: Bar Charts showing number of tweets per event in both training and test set.

3.2 Languages

There are tweets in several different languages in both the training and test sets. Figure 2 shows the top five most common languages in both sets with English being the highest in both - 77.0% and 74.0% respectively. The next most common in the training set are; Spanish (9.0%), Tagalog (2.1%) and French (1.6%) [2]. Somali (13.6%), Arabic (4.9%) and Spanish (1.6%) are the next most common for the test set [2]. Figure 3 shows that tweets in certain languages are more likely to be fake than others.

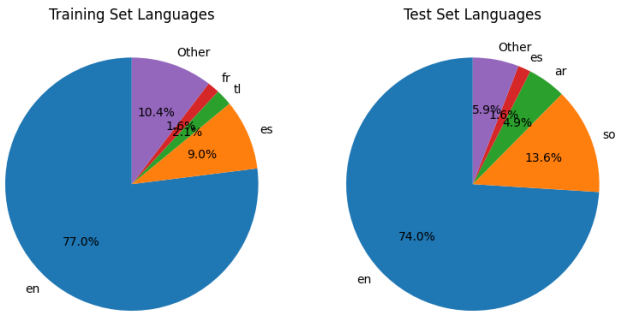

Figure 2: Pie Charts showing the top five languages in training and test set.

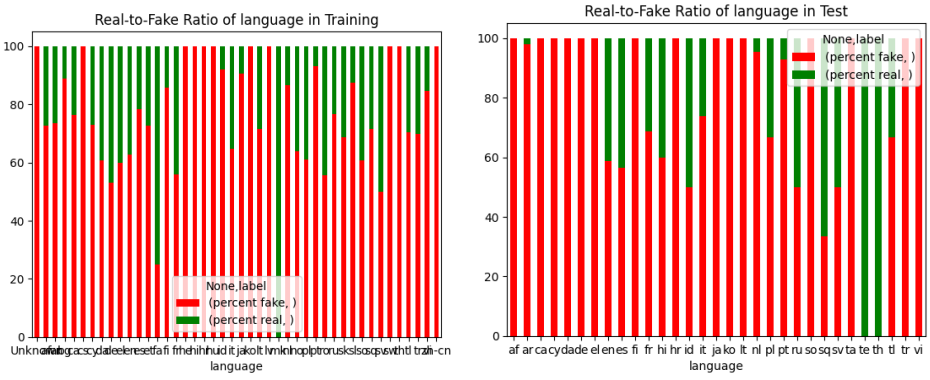

Figure 3: Stacked Bar Charts showing the relationship between label and language.

3.3 Tweet Lengths

The distribution of tweet lengths for the training and test set are shown as box plots in Figure 4.

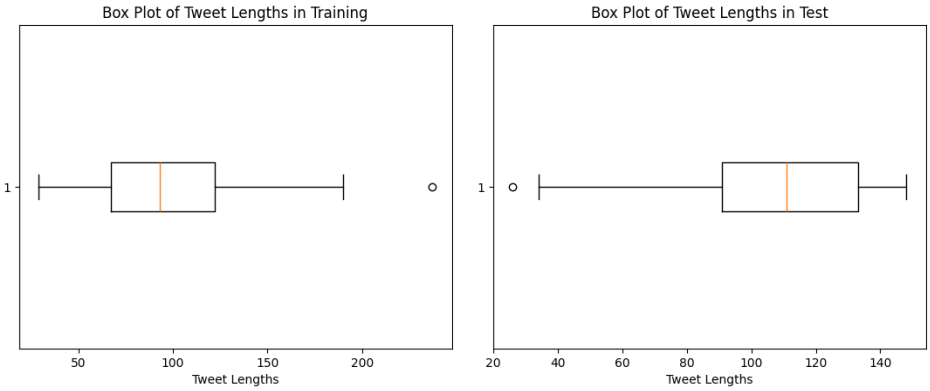

Figure 4: Box Plots showing tweet lengths in training and test set.

The median tweet length for the training set is roughly 90 compared to 110 for the test set. The lengths of the test tweets are also less spread out with a longest tweet of 148 characters vs 237 for the training set.

3.4 Hashtags

Figure 5 shows that - in general - the more hashtags there are in a tweet, the more likely it is to be fake.

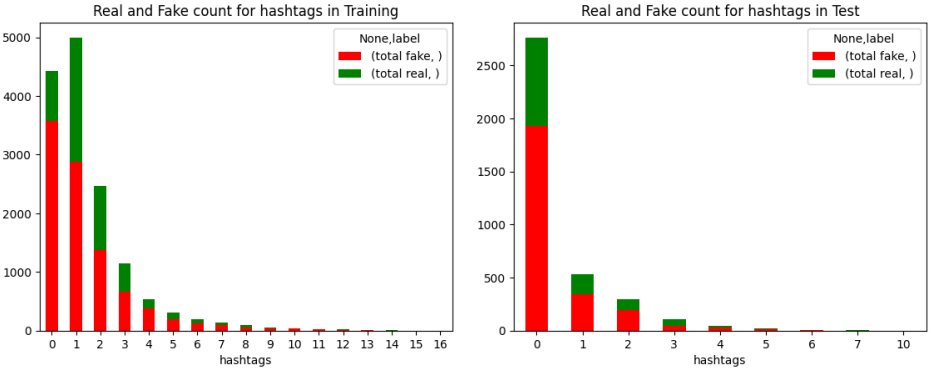

Figure 5: Bar charts showing number of Hashtags

3.5 URLs

Figure 6 shows that every tweet contains at least one URL (except a tiny few in the test set) and there is a clear trend with a tweet is fake or real based on the number of URLs in the tweet.

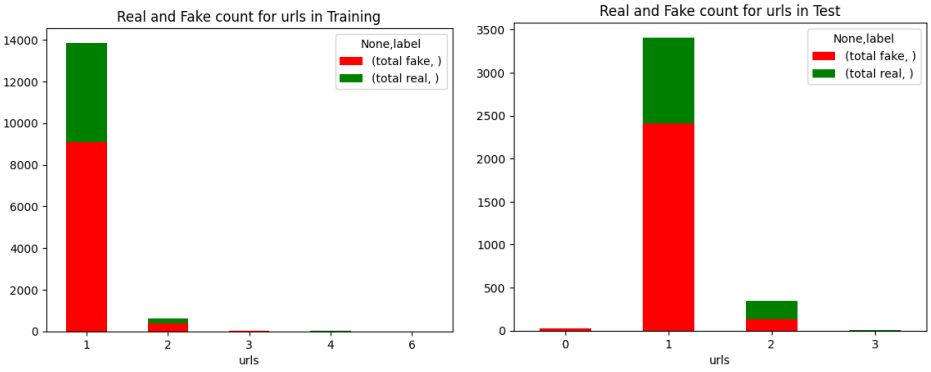

Figure 6: Bar charts showing number of URLs

3.6 Emojis

Figure 7 shows that most tweets don't contain emojis, however, the ones that do are more likely to be fake.

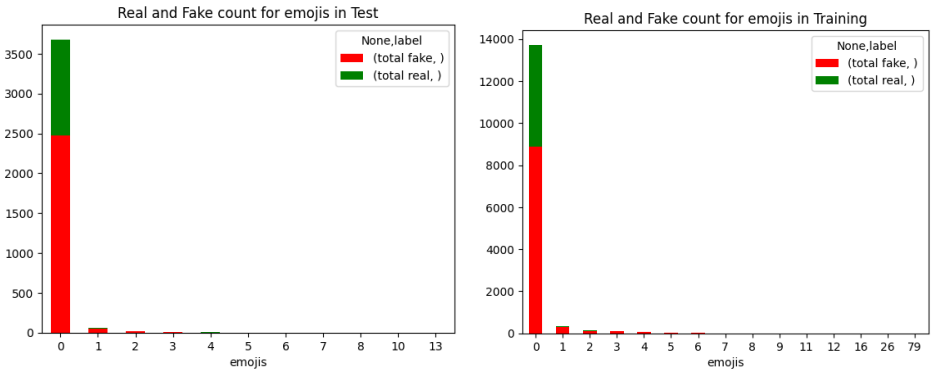

Figure 7: Bar charts showing the number of emojis.

3.7 Retweets

Figure 8 shows there is a slight trend between the number of retweets and whether the post is real or not.

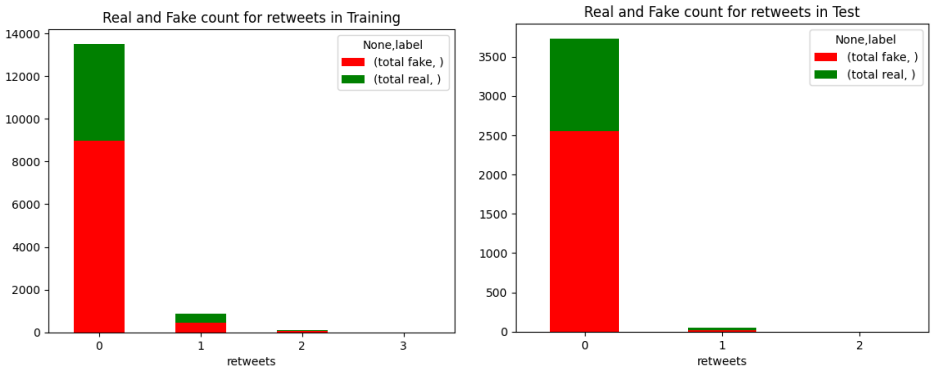

Figure 8: Bar charts showing the number of retweets.

3.8 Modified Tweets

Figure 9 shows that there is a minute number of modified tweets and therefore will not greatly impact the performance of the pipelines.

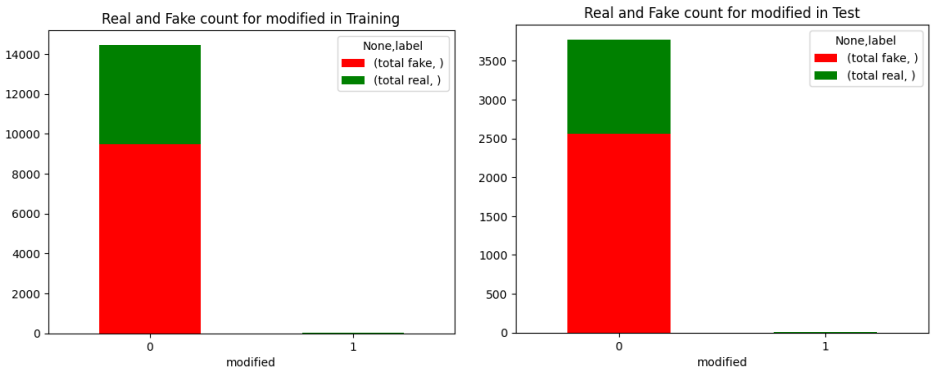

Figure 9: Bar charts showing the number of modified tweets.

3.9 Usernames

Figure 10 shows there is a slight trend between the number of usernames in a tweet and whether it is real or not.

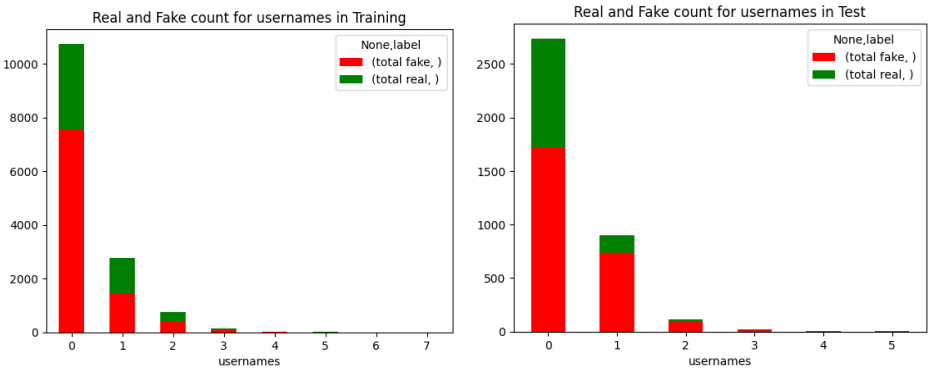

Figure 10: Bar charts showing the number of usernames.

3.10 Timestamps

Figure 11 shows there is a trend between the timestamp of the post and whether it is fake. This also shows that there is a possible under-representation in the range of dates within the MediaEval 2015 dataset (further expanded in Section 3.10).

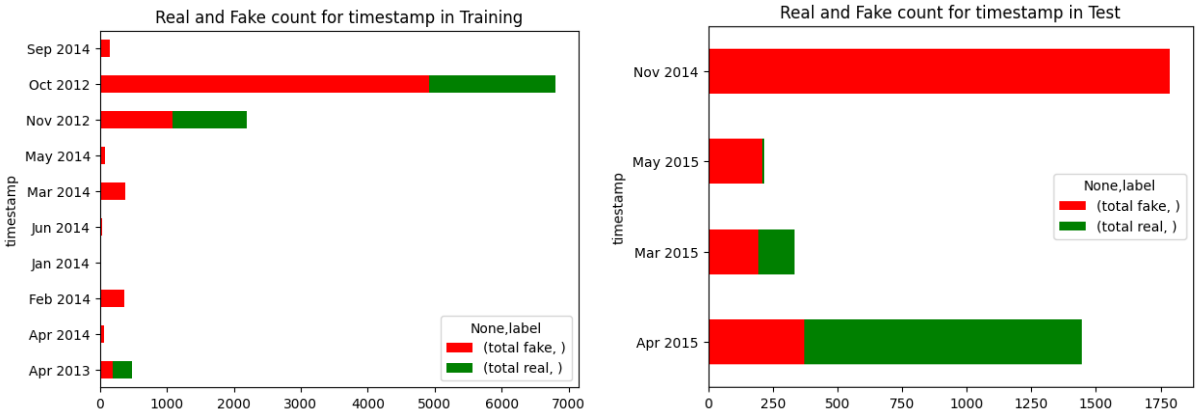

Figure 11: Bar charts showing the number of different timestamps.

3.11 Data Quality and Bias

As mentioned in Section 3.10, there is an under-representation of timestamps in the dataset. Most tweets (over 90%) are from October and November 2012 and all tweets from 2014 are labelled as fake, creating a bias. This lack of diversity may lead to overfitting, affecting the pipelines' performance on new data.

There is also a lack of diversity of events, most tweets in the training and test sets are about Hurricane Sandy and the Syrian Boy respectively. Different events cause different emotional reactions and public opinions which could affect the number of fake tweets produced or how they are written. Not representing enough unique events could once again create a bias and cause overfitting.

4 Pipeline Design

This section outlines the plan for the two pipeline designs, it will highlight different pre-processing, feature selection and dimension reduction techniques which will be tested in different configurations to achieve the best possible result.

4.1 Data Pre-Processing

Data pre-processing can have a significant impact on the performance of a supervised machine learning algorithm [3]. Machine Learning algorithms' success is usually dependent on the quality of data that they operate on, if it is inadequate or contains irrelevant information, they may produce less accurate results [3].

There are several data pre-processing techniques (Section 2 of Notebook) that can be combined and used in the pipelines to improve training results:

- **Text Filtering with Regex** is the process of removing irrelevant noise from text, therefore improving the quality of the data as only important features remain in the data. This is standard practice and has been used in text classification studies before [4].
- **Duplicate Removal** is important as they can introduce bias and skew the distribution of data [5].
- **Normalising Cases** is important is it reduces vocabulary size and introduces consistency. For example, this means that the words "HURRICANE" and 'hurricane' will now be considered the same word in the algorithms.
- **Tokenisation** is the process of turning text data as a string into a list of words, this strips unnecessary whitespace out of the training data and is also required to implement stopword removal, lemmatisation, POS tagging and NER.
- **Stopword Removal** involves removing common words from language that do not carry significant meaning to remove irrelevant data and reduce noise - such as "the" "and" "is" "in" and "to". Studies have also shown that stopword removal removes information that could mislead machine learning algorithms [6].
- **Lemmatisation** is highly similar to stopword removal as it groups words of similar meaning into one common word. For example, the lemma of "running" is "run". This once again reduces the dimensionality of the data by making the data vocabulary more consistent.
- **Part-of-Speech (POS) Tagging** is the process of assigning a grammatical category to each word in a given string. For example, the phrase "the cat is brown" would be tagged as "the DET cat NOUN is VERB brown ADJ". This process will add extra useful information to the training data which should improve results.
- **Named Entity Recognition (NER)** is similar to POS Tagging instead it classifies entities such as organisations or locations. NER proves invaluable in social media analysis because it aids in identifying and categorizing entities mentioned in social media posts, enabling deeper analysis of public opinion and trends [7].
- **Feature Manipulation** is performed in this task by appending features to the end of the processed data to keep some of the information that the stripped data was providing. For example, all punctuation will be removed from the tweets, but adding an emoji and hashtag count at the end of the cleaned text will allow the machine learning algorithm to use this information without it making the data noisy. More features such as the timestamp and length of the tweet will be added as well to give the pipelines more useful data to train on.

In the pipelines, the pre-processing has been split into three methods which can be tested separately and in combination:

- **CleanText**. This method removes punctuation, emojis, URLs, usernames and stopwords from tweetText using regex filtering. It also lemmatises the text, normalises whitespace (regex filtering) and sets all text to lowercase.
- **SmartCleanText** adds POS Tagging and NER to CleanText.

  Tweet: Probably the coolest pic of #hurricanesandy! #NBC #hurricane #sandy #weather #storm #statueofliberty http://t.co/RCf5ZKrn

  CleanText: probably coolest hurricanesandy nbc hurricane sandy weather storm statueofliberty

  SmartCleanText: probably rb coolest jjs hurricanesandy nn organization nbc nnp hurricane nn sandy nn weather nn storm nn statueofliberty nn

- **TweetInformation** concatenates and appends; tweet language, hashtag count, URL count, emoji count, retweet count, modified tweet count, username count, tweet length, and timestamp to the end of the cleaned text.

4.2 Feature Selection

Feature subset selection is the process of identifying and removing as much redundant information as possible [3], therefore, some of the pre-preprocessing methods used can also be considered feature selection such as text filtering and stopword removal. However, further feature selection methods were used and tested in the pipelines to achieve a greater result:

- **Bag-Of-Words (BoW)** is the most commonly used method of text classification where the occurrence of each word is used as a feature for training a classifier [8].
- **N-Grams** are contiguous sequences of n items (words, characters or symbols) within a string of text. The chosen value of n determines the number of adjacent items to be grouped starting at unigrams (1-grams) and then bigrams (2-grams), trigrams (3-grams) and so on (Figure 12). N-Grams works well with noisy data [9]

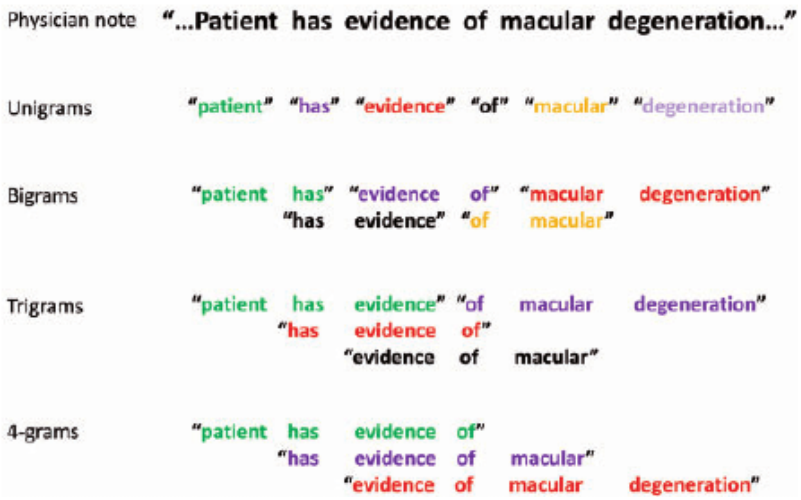

Figure 12: Iteration of N-Grams as n increases [10].

- **TF-IDF** determines how important a word is by weighing its frequency of occurrence in the document and computing how often the same word occurs in other documents [11]. TF-IDF is considered highly useful as it; catches important words, balances word importance, and is simple & easy, it is widely used [12].
- **SelectKBest** simply selects the K most important features from a given set of features based on a scoring function, it can be used after BoW, N-Grams or TF-IDF. SelectKBest is one of the most commonly used feature selection methods [13].

4.3 Dimension Reduction

Removing irrelevant and redundant information from data reduces the dimensionality of the data [3], therefore once again, the above-mentioned pre-processing techniques are considered a form of dimension reduction. However, like with feature selection, further dimension reduction techniques were added to the pipelines to see their effect:

- **TruncatedSVD**. Single-value decomposition is a fundamental matrix factorization technique which can be used to reduce the dimensionality of a term document matrix. It can help with identifying and removing noise from data as well as extracting the most important features [14].
- **Non-Negative Matrix Factorization (NMF)** is a matrix factorisation method which does not allow negative elements. This is useful for machine learning algorithms which require positive values.

4.4 Pipeline One - Multinomial Naive Bayes (MNB)

Multinomial Naive Bayes is a favoured algorithm for text classification since it performs well on datasets with high dimensionality [15] and works effectively with discrete features such as word counts. Pipeline 1 will use MNB for every run, changing the pre-processing, feature selection, dimensionality reduction and other parameters to optimise results while using it.

4.5 Pipeline Two - stochastic gradient descent (SGD)

SGD implements regularized linear models, by default it implements a support vector machine (SVM) [16] which is a model designed for binary classifications like this task [17]. Like Pipeline 1, Pipeline 2 will use SGD for all runs, changing other parameters to optimise results.

5 Evaluation

F1 Scores were used to evaluate the pipelines alongside the confusion matrices. F1 scores are calculated as follows:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

$$precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Table 1 aand Table 2 shows the process of training and testing the pipelines. The pipeline configuration is tweaked each time to try to improve results. 'Basic' refers to CleanText pre-processing whereas 'Smart' refers to SmartCleanText pre-processing. Results were averaged over five runs of each configuration.

ID	Pipeline 1 Configuration	F1 Score	Time Elapsed (s)
R0	TweetText, BoW	0.52	0.1
R1	Basic, BoW	0.48	0.1
R2	Basic, N-Grams(1,1)	0.48	0.1
R3	Basic, N-Grams(1,2)	0.50	0.2
R4	Basic, N-Grams(2,2)	0.65	0.2
R5	Basic, TF-IDF	0.88	0.1
R6	Basic, TF-IDF, SVD	n/a	n/a
R7	Basic, TF-IDF, NMF	(0.68)	10.4
R8	Smart, TF-IDF	0.89	0.1
R9	Basic, Tweet Information, TF-IDF	0.88	0.1
R10	Smart, Tweet Information, TF-IDF	0.89	0.2
R11	Smart, Tweet Information, TF-IDF, MNB(alpha=0.1)	0.51	0.3
R12	Smart, Tweet Information, TF-IDF, MNB(alpha=0.4)	0.76	0.2
R13	Smart, Tweet Information, TF-IDF, MNB(alpha=0.7)	0.88	0.2
R14	Smart, Tweet Information, TF-IDF, MNB(alpha=1 **DEFAULT**)	0.89	0.2
R15	Smart, Tweet Information, Month & Year, TF-IDF	0.91	0.2
R16	Smart, Tweet Information, Month & Year, TF-IDF, SelectKBest(k=1700)	0.91	0.3

Table 1: Iterations for Pipeline 1 using MultinomialNB as classifier. Highest scores - Gold, Silver, Bronze, (F1 Score) - Error in Result

ID	Pipeline 2 Configuration	F1 Score	Time Elapsed (s)
R0	TweetText, BoW	0.64	0.1
R1	Basic, BoW	0.84	0.1
R2	Basic, N-Grams(1,1)	0.85	0.1
R3	Basic, N-Grams(1,2)	0.65	0.2
R4	Basic, N-Grams(2,2)	(0.68)	0.1
R5	Basic, TF-IDF	0.85	0.1
R6	Basic, TF-IDF, SVD	(0.67)	0.8
R7	Basic, TF-IDF, NMF	(0.68)	10.9
R8	Smart, TF-IDF	0.80	0.1
R9	Smart, N-Grams(1,1)	0.80	0.1
R10	Basic, Tweet Information, TF-IDF	0.86	0.2
R11	Basic, Tweet Information, N-Grams(1,1)	0.84	0.2
R12	Smart, Tweet Information, TF-IDF	0.82	0.2
R13	Smart, Tweet Information, N-Grams(1,1)	0.87	0.2
R14	Smart, Tweet Information, Month & Year, TF-IDF	0.87	0.3
R15	Smart, Tweet Information, Month & Year, N-Grams(1,1)	0.86	0.3
R16	Basic, Tweet Information, Month & Year, TF-IDF	0.88	0.3
R17	Basic, Tweet Information, Month & Year, N-Grams(1,1)	0.84	0.4
R18	Basic, Tweet Information, Month & Year, TF-IDF, SelectKBest(k=1700)	0.89	0.3

Table 2: Iterations for Pipeline 2 using Stochastic Gradient Descent as classifier. Highest scores - Gold, Silver, Bronze, (F1 Score) - Error in Result

Initially both algorithms were trained with raw tweetText with the BoW feature selection. BoW was necessary as some form of vectorisation needs to take place before training with MNB and SGD. The algorithms performed modestly with no pre-processing which may have been caused by their suitability for text classification [15], [17].

The first three feature selection methods from Section 4.2 were then tested individually. MNB greatly prefers TF-IDF, obtaining a mean F1 score of 0.88, whereas, SGD performed consistently well with all three methods. MNB works by representing terms as feature vectors with each component corresponding to the frequency of a term, therefore, using n-grams may decrease performance as similar terms are added to the data, adding noise.

The dimensionality reduction techniques were then tested. TruncatedSVD did not work with MNB due to the matrix factorisation algorithm creating negative values in some elements. MNF executed, however, provided a misleading mean F1 score of 0.68. This result proved to be a common indication of an error while testing because it arose when the algorithm only predicted for one class, this is shown in the confusion matrix in Figure 13. MNF and SVD consistently caused this error when testing with SGD and also increased run time dramatically (especially NMF with 10+ seconds), therefore, they were no longer considered for the remaining tests, leaving dimensionality reduction to be inherently performed via the pre-processing methods.

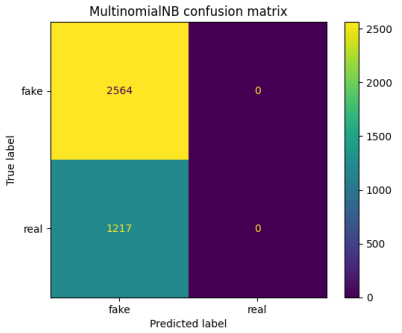

Figure 13: Confusion Matrix for single-class prediction error

Smart pre-processing was then tested in the pipelines in combination with the best-performing feature selection methods from previous tests. The introduction of POS and NER tagging greatly improved the accuracy of MNB with an average F1 score of 0.9. This may be due to these techniques adding valuable linguistic information which can be used as part of the feature vectors with MNB, improving results [18]. However, smart pre-pre-processing initially performed worse with both feature selection methods with SGD. SVMs (the model which SGD is implementing) can be sensitive to noise in the data [19] which could have possibly caused this dip in performance.

The algorithms were then tested with TweetInformation added to basic and smart pre-processing. This further improved MNB with both basic and smart processing, most likely due to the introduction of new feature vectors once again. Adding TweetInformation also improved SGD. Interestingly, it yielded the greatest results when paired with smart pre-processing and unigrams, contradicting the previous suggestion that SGD performs worse with noisier data. It may be the case that POS tags, NER tags and TweetInformation pair well together as TweetInformation could provide implied information which was lost when the tags were added (Section 4.1), such as tweet length.

The alpha parameter of MNB was then incrementally tested, however, it was clear that the default value of 1 yielded the best results. Changing the parameters for SGD was investigated however a non-default combination that yielded better results was not found so it has not been included.

Tweet timestamps in the form "{DATE} {MONTH}" were appended to Tweet Information. These improved results further, giving MNB its joint highest mean F1 score of 0.91 and SGD its second highest of 0.88 (with TF-IDF). However, this may be caused by overfitting the test data due to the under-representation of timestamps (as mentioned in Section 3.10). It could be argued that the timestamp of a tweet is not a good indicator of authenticity as opposed to the text body itself, but I believe that it is because it gives temporal context for specific news or events.

The last thing tested was adding SelectKBest feature selection to the pipeline configurations. I found that a K-value of roughly 1700 (found iteratively) produced the best results, keeping MNB at its highest F1 score of 0.91 and bringing SGD to its highest of 0.89. The confusion matrices from the best runs are shown in Figure 14.

Since the task of this project is to identify fake posts, correctly identifying a tweet as 'fake' is considered a True Positive (TP). Correctly identifying a real tweet is a True Negative (TN) and incorrectly identifying a fake as real and real as fake, is a False Positive (FP) and False Negative (FN) respectively. Figure 14 shows that MNB predicts more True Positives and less False Positives than SGD, however, SGD predicts more True Negatives and less False Negatives than MNB.

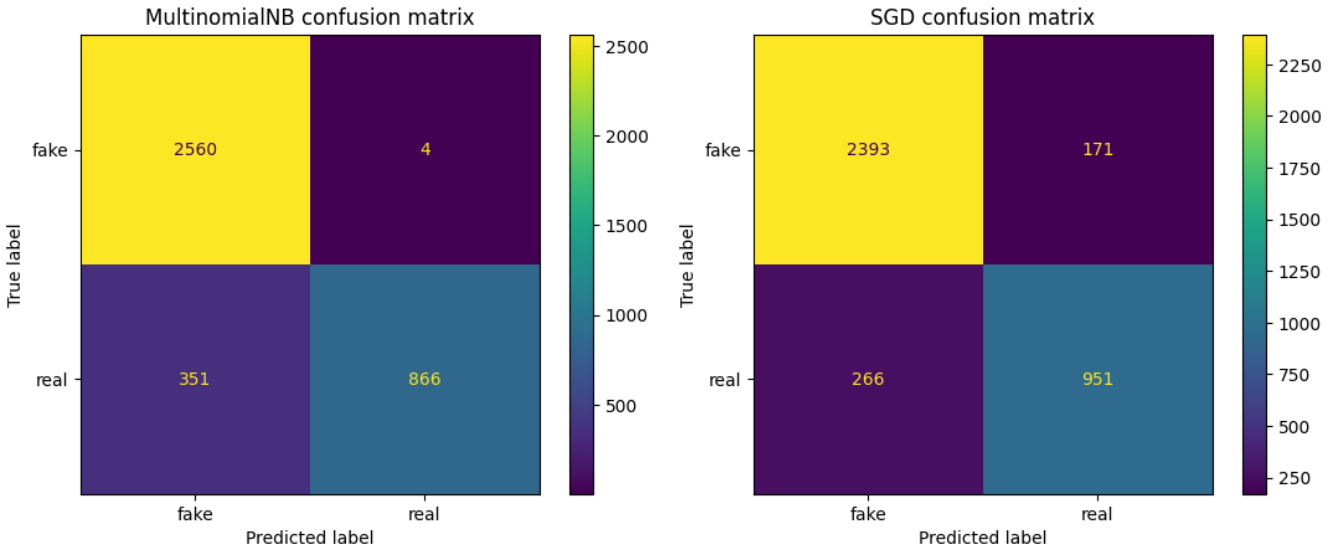

Figure 14: Confusion Matrices for best-performing pipeline configurations

6 Conclusion

These results do not suggest that one algorithm or pipeline configuration is objectively better than the other, if the goal is obtaining the highest F1 score possible then MNB may be chosen. Likewise, if the goal is to achieve as many True Positives as possible then MNB is the better choice. However, SGD provides a more balanced confusion matrix; there are a similar number of FPs and FNs, suggesting that it is a more robust algorithm.

When classifying fake social media posts, these results also suggest that N-Grams and TF-IDF are superior feature selection methods than bag-of-words, with TF-IDF outperforming N-Grams when more information is added to the training data. POS and NER tagging proved to be useful pre-processing techniques and improved accuracy, this could be due to the semantics of the words used in social being a large factor as to whether they are real or not. Appending tweet information - which may have been stripped from the original tweet in pre-processing - also proved to be a useful technique, by the important information without adding the noise from the original tweet. These results also suggest that SelectKBest is a robust feature selection method to improve the accuracy of machine learning algorithms.

However, these results do not clearly show whether the timestamp of a tweet is a good indication of whether it is fake or not. If the range of timestamps in the train and test was larger, the results may have given a better representation of this.

References

[1] [Online]. Available: https://www.bbc.co.uk/news/av/magazine-30043574.

[2] [Online]. Available: https://www.loc.gov/standards/iso639-2/php/code_list.php.

[3] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised leaning," *International journal of computer science*, vol. 1, no. 2, pp. 111–117, 2006.

[4] M. A. Abid, S. Ullah, M. A. Siddique, M. F. Mushtaq, W. Aljedaani, and F. Rustam, *Spam sms filtering based on text features and supervised machine learning techniques - multimedia tools and applications*, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11042-022-12991-0.

[5] Anishnama, *How duplicate entries in data set leads to ovetfitting?* 2023. [Online]. Available: https://medium.com/@anishnama20/how-duplicate-entries-in-data-set-leads-to-ovetfitting-2e3376e309c5.

[6] C. Silva and B. Ribeiro, "The importance of stop word removal on recall values in text categorization," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3, 2003, 1661–1666 vol.3. DOI: 10.1109/IJCNN.2003.1223656.

[7] M. Charles, *Named entity recognition (ner) in natural language processing*, 2023. [Online]. Available: https://medium.com/@cmugendi3/named-entity-recognition-ner-in-natural-language-processing-94f7c0cf7537.

[8] A. Sen, *Text classification-from bag-of-words to bert*, 2021. [Online]. Available: https://medium.com/analytics-vidhya/text-classification-from-bag-of-words-to-bert-1e628a2dd4c9.

[9] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:170740.

[10] M. Rastegar-Mojarad, S. Hebbring, Z. Ye, J. Mayer, C. Jacobson, and S. Lin, "Application of clinical text data for phenome-wide association studies (phewass)," *Bioinformatics*, Feb. 2015.

[11] S. Anala, *Text classification using tf-idf*, 2020. [Online]. Available: https://medium.com/swlh/text-classification-using-tf-idf-7404e75565b8.

[12] N. Ninja, *Tf-idf: Weighing importance in text*, 2023. [Online]. Available: https://letsdatascience.com/tf-idf/.

[13] K. D, *Optimizing performance: Selectkbest for efficient feature selection in machine learning*, 2023. [Online]. Available: https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48#:~:text=Then%2C%20it%20selects%20the%20K,when%20dealing%20with%20large%20datasets.&text=SelectKBest%20has%20two%20parameters%3A%20score%20function%20and%20k..

[14] [Online]. Available: https://www.sciencedirect.com/topics/mathematics/singular-value-decomposition.

[15] [Online]. Available: https://developer.ibm.com/tutorials/awb-classifying-data-multinomial-naive-bayes-algorithm/.

[16] [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

[17] V. K. Researcher, *All you need to know about support vector machines*, 2022. [Online]. Available: https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector-machine/.

[18] S. Li, *Named entity recognition and classification with scikit-learn*, 2018. [Online]. Available: https://towardsdatascience.com/named-entity-recognition-and-classification-with-scikit-learn-f05372f07ba2.

[19] "Generalized pinball loss svms," *Neurocomputing*, vol. 322, pp. 151–165, 2018. DOI: 10.1016/J.NEUCOM.2018.08.079.