Yiyang Lu
30979358

January 25, 2024

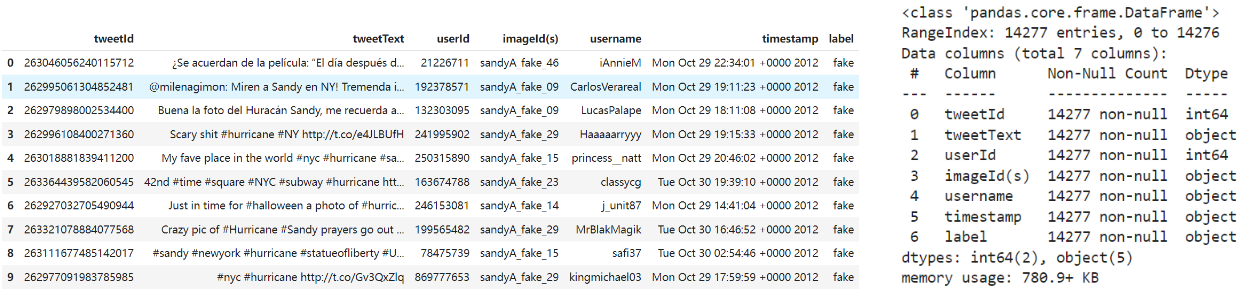
COMP3222 Machine Learning Technologies

Coursework Final Report

1 Introduction and data analysis

The aim of the project is to analysis the dataset from the MediaEval 2015 "verifying multimedia use" task which is to train two machine learning models as a tool, so that the fake posts on the social media can be detected. According to the requirements, the project would only use the text and metadata in the dataset to train the models.

From the dataset the problem is a classification problem. The models used in the project including machine learning and deep learning methods, both are supervised methods according to the labels in the dataset provided.



	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
3	262996108400271360	Scary shit #hurricane #NY http://t.co/e4JLBUfH	241995902	sandyA_fake_29	Haaaaarryyy	Mon Oct 29 19:15:33 +0000 2012	fake
4	263018881839411200	My fave place in the world #nyc #hurricane #sa...	250315890	sandyA_fake_15	princess__natt	Mon Oct 29 20:46:02 +0000 2012	fake
5	263364439582060545	42nd #time #square #NYC #subway #hurricane htt...	163674788	sandyA_fake_23	classycg	Tue Oct 30 19:39:10 +0000 2012	fake
6	262927032705490944	Just in time for #halloween a photo of #hurric...	246153081	sandyA_fake_14	j_unit87	Mon Oct 29 14:41:04 +0000 2012	fake
7	263321078884077568	Crazy pic of #Hurricane #Sandy prayers go out ...	199565482	sandyA_fake_29	MrBlakMagik	Tue Oct 30 16:46:52 +0000 2012	fake
8	263111677485142017	#sandy #newyork #hurricane #statueofliberty #U...	78475739	sandyA_fake_15	safi37	Tue Oct 30 02:54:46 +0000 2012	fake
9	262977091983785985	#nyc #hurricane http://t.co/Gv3QxZlq	869777653	sandyA_fake_29	kingmichael03	Mon Oct 29 17:59:59 +0000 2012	fake

```
<class 'pandas.core.frame.DataFrame'>
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
```

Figure 1: First Ten Rows and Info of The Training set

The dataset available contains a training set and a test set, both of them are csv files. As in the Figure 1, both of the datasets contains seven columns: tweetId, tweetText, userId, imageId(s), username, timestamp and label. There are 14277 tweets provided in the training set and 3755 in the test set. Both datasets contain no missing value in any of the column. The shape of the training and testing datasets are (14277, 7) and (3755, 7).

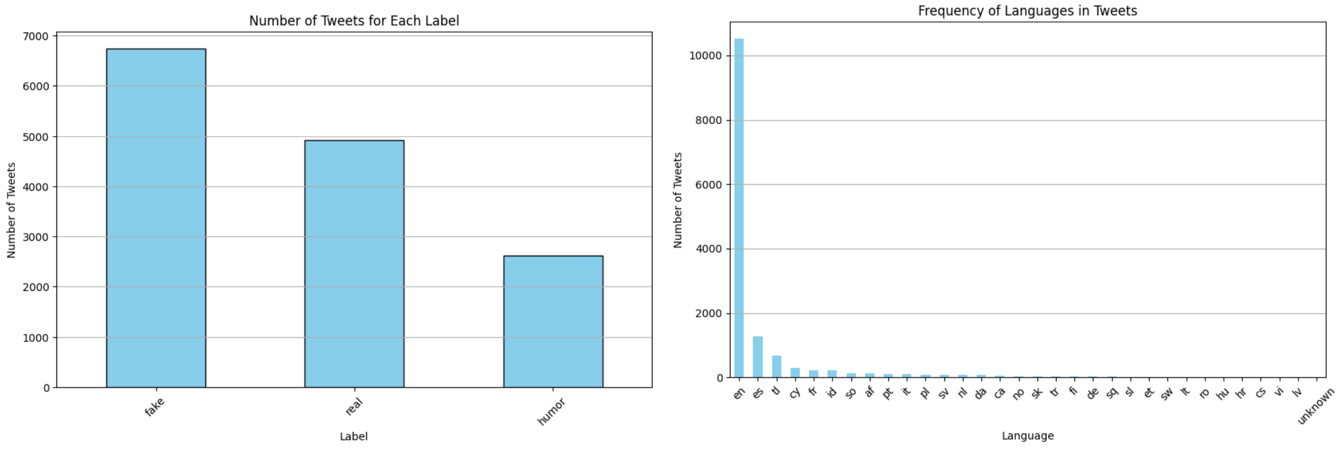


Figure 2: Distribution of Labels and language in the tweetText column

Figure 2 is the distribution of three labels and the languages used in the tweet texts column of the training set. The numbers of fake, real and humor labels are 6742, 4921 and 2614, the humor label will be counted as fake label as well. The ratio of fake and real posts is nearly 2:1, so there would be a bias towards the fake posts. The language distribution shows that the large majority of the posts are in English, there would be a minor influence to the performance of the models to discard the non-English posts in the training of the models.

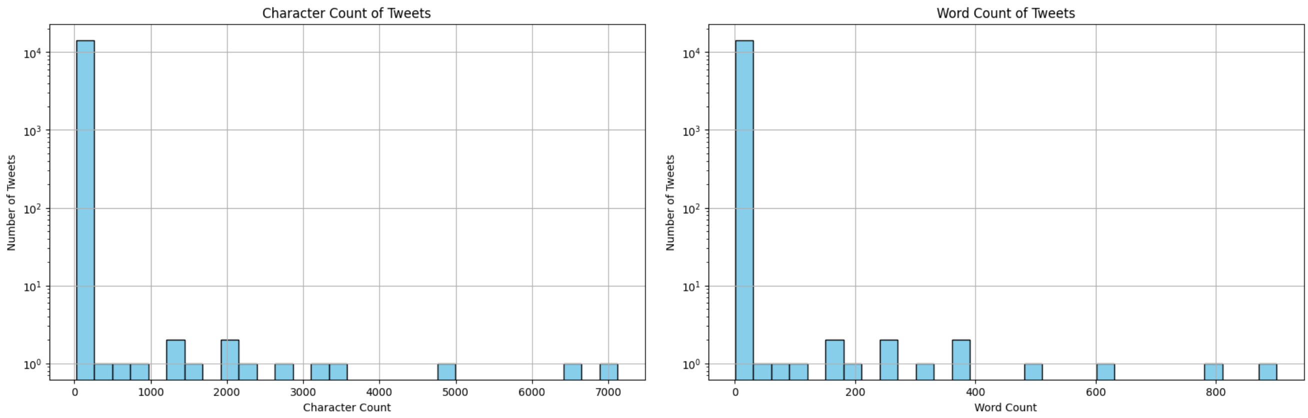


Figure 3: Length and Word Count of the tweetText

The maximum length of the English tweetText in the training set is 7125 characters and it contains 901 words, the minimum length and word count is 26 and 1. The majority of the tweet content has very short length and small word count, below 1000 characters and 100 words. The assumption for now is that the content length and fake posts have certain relationship, so the length of the fake posts and real posts would be compared as following Figue 4.

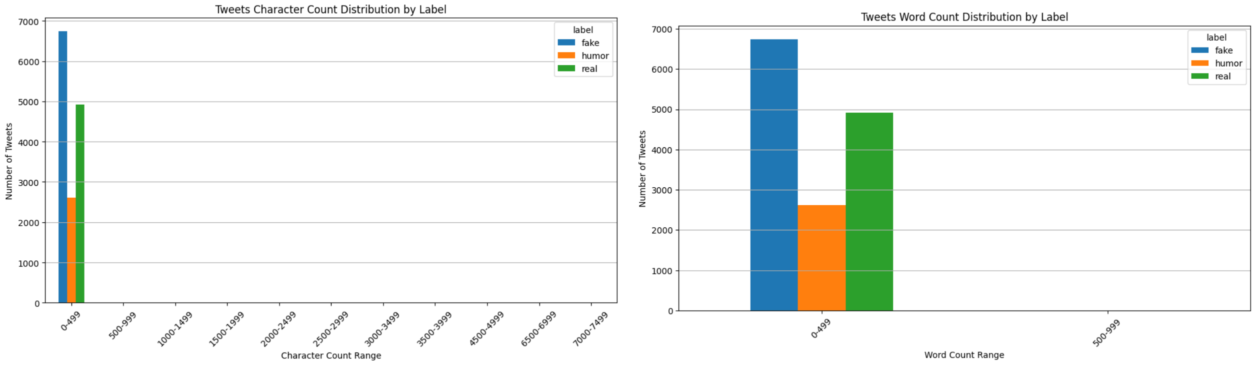


Figure 4: Length and Word Count of the tweetText For Each Label

Due to the uneven distribution of the tweets length, the direct count of the number of tweets cannot show the relation between different labels. The use of mean can better address this issue as shown in Figure 5.

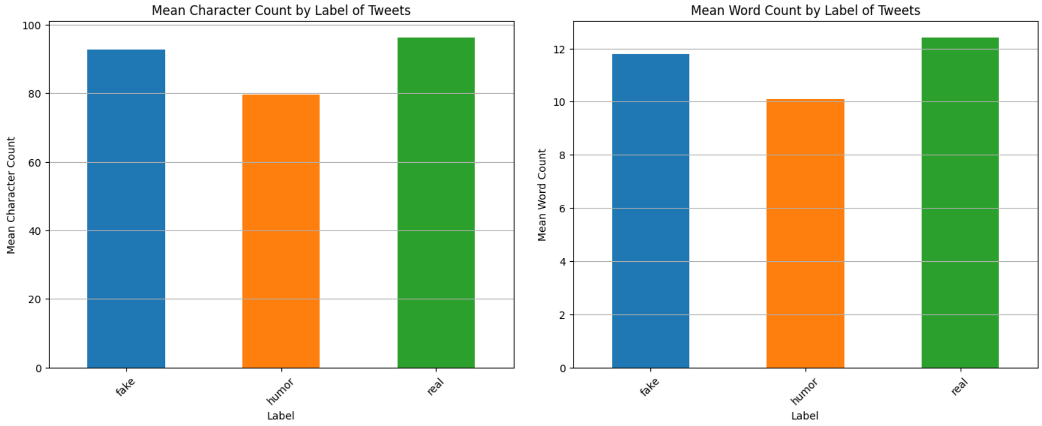
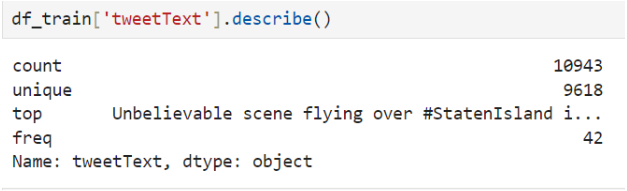


Figure 5: Mean Length and Word Count of the tweetText For Each Label

It seems that real tweets tend to be longer than the fake and humor tweets.



```
df_train['tweetText'].describe()
```

```
count                                          10943
unique                                          9618
top       Unbelievable scene flying over #StatenIsland i...
freq                                              42
Name: tweetText, dtype: object
```

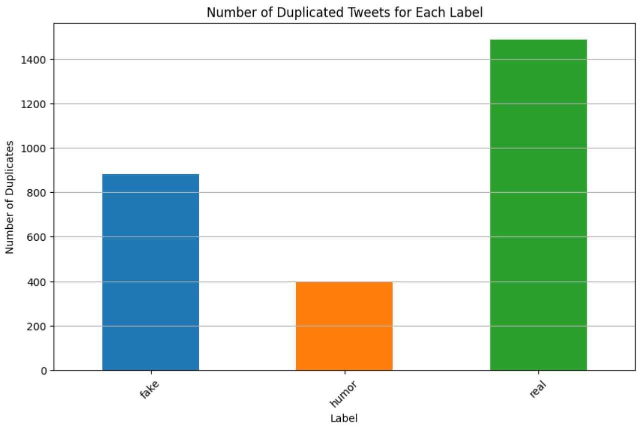


Figure 6: Duplicated English Tweets

The training dataset is also biased towards the real tweets in terms of duplication of tweetText. There are 1325 duplicated English tweetText, and more real labeled tweets are duplicated than the fake and humor labels.

The evaluation method including the F1 score as the final judgement of the performance of the models, but other scores might also be included to adjust the bias of the dataset, such as recall and precision scores.

2 Related Work

A review of the machine learning techniques in detecting fake news compared the common algorithms performance in the area, the highest accuracy of nearly 100% is the approach that used SVM, Neural Network and Naïve Bayes on Twitter messages[1]. In the survey of evaluating machine learning techniques, they also mentioned the use of Decision Tree and Random Forest algorithms to detect the fake news on social media[2]. The experiment of detecting fake news by [3] used three algorithms including SVM, Naïve Bayes and Logistic Regression, the result shows that SVM and Naïve Bayes have the best result.

According to the tool of detecting fake news[4], the researchers used three feature selection models for text content: Bag-of-Words, N-gram and Term Frequency-Inverse Document Frequency(TF-IDF). Their result shows that the combination of linear SVM and TF-IDF works well on both content and title detection accuracy. Many researchers used content features like tweet length, word count and whether URL presents in the content as the feature used in the classifiers[5][6].

3 Pipeline design

3.1 Feature Selection

In this project, the dataset's limitation that excludes the need for image processing leads to the omission of the imageId(s) column. For the username and userId, since each username is corresponding to one unique userId, use only one of them is enough, in this project the username column is used for as a feature. From the related work, the most common feature used for similar domain is the content, thus the primary focus will be on the tweetText column, which will serve as the sole feature utilized for analysis. Both the words and length of the tweetText will be consider as the feature in the training process.

3.2 Data Preprocessing

Initially, only English tweets were retained for training by using the langdetect library to filter out non-English content. However, due to potential inconsistencies in language detection, all languages were eventually included in the training of both algorithms, with stopwords removal applied to English text only.

In the first iteration, the data preprocessing step is to clean the data and remove stop words according to the NLTK stopwords library. These are the most common steps mentioned in several papers[7][8]. Data cleaning including convert the tweetText in to lowercase and remove URLs and non-alphabetic characters.

In the second iteration, tokenization and lemmatization are added to the data preprocessing step, these steps are also mentioned in [7][8]. Tokenization is the step that split the text by words and make it possible to evaluate each word individually. Lemmatization reduces the words back to their base forms, and reduces the complexity of the string. Punkt and WordNet are the free libraries used for these two steps. The impact of this change would be discussied later in the document.

3.3 Feature Extraction

Feature extraction is the process of transforming raw data into a set of numerical features that can be used to train a machine learning model[6]. In this project TF-IDF would be used as the only feature extraction method in both iterations. TF-IDF is a numerical statistic used in text mining and information retrieval to reflect how important a word is to a document in a collection or corpus[3].

$$TF(t,d) = \frac{Number\ of\ times\ term\ t\ appears\ in\ document\ d}{Total\ number\ of\ terms\ in\ document\ d}$$

$$IDF(t,D) = log\frac{Total\ number\ of\ documents\ D}{Number\ of\ documents\ with\ term\ t\ in\ it}$$

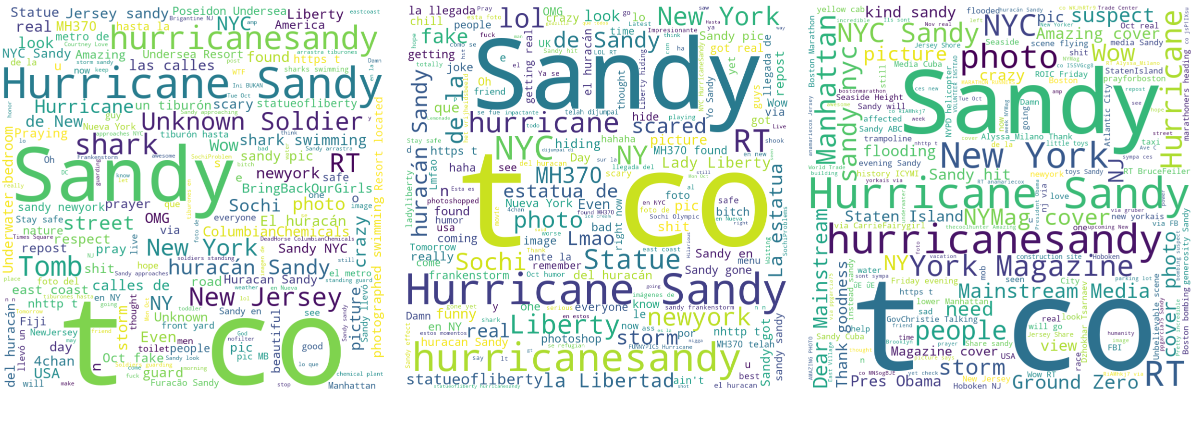


Figure 7: English Word cloud of Three Labels

As the dataset is not very large, to prevent overfitting, the max feature hyperparameter is applied as 4000 which is a random guess from the size of the training set, it will only keep the 4000 most frequent terms and discard the rest. This would be tuned to find the best number in the evaluation step.

3.4 Algorithm Design

As mentioned in previous, SVM and random forest are the most common machine learning models for fake text classification problems. Due to the limitation of my GPU and time available, deep learning models were not considered as they could be quite computational expensive and time consuming. Consider the size and the form of the datasets given, machine learning approach is enough to handle the classification. If image process is needed, then a deep learning model like Convolutional Neural Networks(CNN) could be better.

3.4.1 SVM Model

SVM stands for Support Vector Machine, it is a common algorithm for classification problem. It uses hyper-lines to separate the data space by the feature(s), obtain the minimal classification error[9]. In the first iteration, different feature combinations within SVMs to assess their impact on classification performance is explored. SVM_1 used both semantic content (through TF-IDF) and structural properties (through text length) of the tweetText, SVM_2 used the text length only, and SVM_3 used TF-IDF. The classifier used is SVC from scikit-learn library. These design choices are grounded in the premise that different aspects of tweet texts can provide different benefits for classification, a concept supported by existing research in the field of text analysis[5].

3.4.2 Random Forest Model

Random Forest was selected as the second algorithm due to its proficiency in managing biased datasets and has capability to mitigate overfitting. By applying the ensemble learning approach to decision trees, Random Forest kept the benefits of decision tree, and more robust for high dimensional data[10]. Similar features were used in the first iteration, RF_1 used TF-IDF and text length, RF_2 used text length and RF_3 used TD-IDF. The classifier used is RandomForestClassifier from scikit-learn library.

4 Evaluation and Parameter Tuning

4.0.1 SVM Model

F1 Metric	**SVM_1**	**SVM_2**	**SVM_3**
Accuracy	0.68	0.68	0.84
Macro Avg	0.40	0.40	0.80
Weighted Avg	0.55	0.55	0.83

Table 1: F1 Scores Comparison of SVM Models From Scikit-learn

From the F1 scores we can see that the length of the tweetText might be a downside for the classification of fake and real tweets. This is may be due to the highly complexity leads to overfitting, it suggests that the semantic content of the tweets is more informative than their structural properties, and that SVM_3's focus on semantic features allows it to more effectively discriminate between fake and real labels.

To find the best hyperparameter, grid search and cross validation is used for tuning process[9][11]. I tried the basic regularization parameters $\mathcal{C}$ and γ with 5-fold cross validation at first, but the result remained the same. Then follow a similar choice of max-df and min-df as in [7], these two parameters specifies the maximum and minimum document frequency allowed for any term. Also n-gram parameter for the n-values as mentioned in [3]. Also the kernel type to be used in the SVM is also added, this is just from my curious of the kernel function. Unfortunately, the F1-scores with these best parameters even drops, which I think is due to overfitting and the bias of the dataset, then I addressed a stratified K-fold method with k=3 to try to reduce the influence of the bias, the result is still worse than the first try. But in this iteration, the max feature number for TF-IDF I find ideal for SVM is 4500 rather than 4000.

For further improvement, what I came out was to set more features from the dataset to the model, maybe there's other relations between the labels and other features, so I pick username and timestamp, cause maybe certain people would post fake or real tweets only, or at certain time the fake posts would be released because of the company work time behind them. To try to constrain the complexity of the model, I only use the combination of two features as the maximum feature selection, along with the best parameters obtained from the previous grid search.

F1 Metric	**tweetText**	**tweetText & username**	**tweetText & timestamp**
Accuracy	0.67	0.83	0.66
Macro Avg	0.42	0.79	0.43
Weighted Avg	0.56	0.83	0.56

Table 2: F1 Scores Comparison of SVM Models With Different Features

Here we can see that the combination of the 'tweetText' and 'username' columns has the highest performance. These could be used in the last iteration of the training process along with other adjustments on the training and testing data.

4.0.2 Random Forest Model

F1 Metric	**RF_1**	**RF_2**	**RF_3**
Accuracy	0.63	0.66	0.66
Macro Avg	0.41	0.42	0.41
Weighted Avg	0.54	0.55	0.55

Table 3: F1 Scores Comparison of Random Forest Models From Scikit-learn

The classification result of Random Forest algorithm is evenly spread between different feature extraction combinations. A similar evaluation process is followed for the Random Forest algorithm. The grid parameters picked at first was only n-estimators and the max-depth of the forest, and then the max feature of TF-IDF and class weight also added for addressing the dataset bias. However, after several times of the grid search and cross validation, the result remains nearly the same. The best hyperparameters obtain for the TF-IDF max feature was 3850 which is different from the SVM algorithm's.

I also tried the combination of two different columns of the training dataset for the Random Forest algorithm, yet the results did not improve nor drop.

F1 Metric	tweetText	tweetText & username	tweetText & timestamp
Accuracy	0.66	0.67	0.67
Macro Avg	0.41	0.43	0.41
Weighted Avg	0.55	0.56	0.55

Table 4: F1 Scores Comparison of Random Forest Models With Different Features

5 Iteration

As mentioned before, the second iteration of the training process included a further preprocessing of the data, which added tokenization and lemmatization, combining with the best parameters and feature selection obtained in the previous iteration. The data preprocessing also dropped the duplicated content rows to improve the data quality. This time, the F1 scores of the Random Forest model has a significant improvement, even better than SVM model.

F1 Metric	SVM	Random Forest
Accuracy	0.85	0.89
Macro Avg	0.82	0.86
Weighted Avg	0.85	0.88

Table 5: Final F1 Scores Comparison of SVM and Random Forest Models

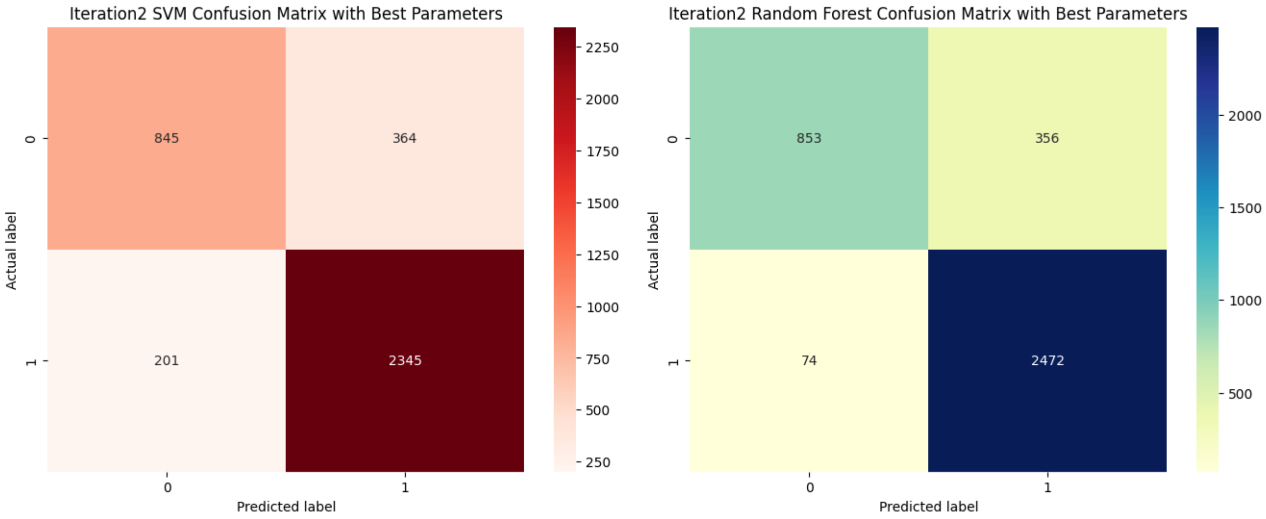


Figure 8: Confusion Matrix of the Best SVM and Random Forest Model

6 Conclusion

The project has several iterations on different steps of the training process. The final selected features that obtain the best performances are the content analysis using TF-IDF, and the combination of 'tweetText' and 'username'. Data preprocessing is also crucial, although some algorithm might be more sensitive to the data quality and different preprocessing methods. For this classification of the social media content using only text-based data, tokenization and lemmatization is necessary for Random Forest models.

For the iteration and comparison process of the training machine learning models, it is important to control the variables, and keep each iteration with minor changes to select and understand the meaning of each influence factor.

The for the further improvement on this task, I would like to try different feature extraction technique like CountVectorizer[12]. Other features like the present of URLs in 'tweetText' could be useful for the training as well. Select a different algorithm like ANN could obtain a better model.

References

[1] M. Choudhary, S. Jha, D. Saxena, A. K. Singh, *et al.*, "A review of fake news detection methods using machine learning," in *2021 2nd International Conference for Emerging Technology (INCET)*, IEEE, 2021, pp. 1–5.

[2] N. A. Patel and R. Patel, "A survey on fake review detection using machine learning techniques," in *2018 4th international Conference on computing Communication and automation (ICCCA)*, IEEE, 2018, pp. 1–6.

[3] E. M. Mahir, S. Akhter, M. R. Huq, *et al.*, "Detecting fake news using machine learning and deep learning algorithms," in *2019 7th international conference on smart computing & communications (ICSCC)*, IEEE, 2019, pp. 1–5.

[4] B. Al Asaad and M. Erascu, "A tool for fake news detection," in *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, IEEE, 2018, pp. 379–386.

[5] S. Krishnan and M. Chen, "Identifying tweets with fake news," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, IEEE, 2018, pp. 460–464.

[6] N. Y. Hassan, W. H. Gomaa, G. A. Khoriba, and M. H. Haggag, "Supervised learning approach for twitter credibility detection," in *2018 13th International conference on computer engineering and systems (ICCES)*, IEEE, 2018, pp. 196–201.

[7] U. Parida, M. Nayak, and A. K. Nayak, "News text categorization using random forest and naıve bayes," in *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON)*, IEEE, 2021, pp. 1–4.

[8] R. Jehad and S. A. Yousif, "Fake news classification using random forest and decision tree (j48)," *Al-Nahrain Journal of Science*, vol. 23, no. 4, pp. 49–55, 2020.

[9] P. Lameski, E. Zdravevski, R. Mingov, and A. Kulakov, "Svm parameter tuning with grid search and its impact on reduction of model over-fitting," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 15th International Conference, RSFDGrC 2015, Tianjin, China, November 20-23, 2015, Proceedings*, Springer, 2015, pp. 464–474.

[10] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 272, 2012.

[11] C.-W. Hsu, C.-C. Chang, C.-J. Lin, *et al.*, *A practical guide to support vector classification*, 2003.

[12] B. Sumathi *et al.*, "Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020.