Warayut Muknumporn (wm2n21@soton.ac.uk 32314914)

Machine Learning Technologies Coursework

I. Introduction and Data Analysis

Popular social media platforms like Twitter and Facebook have become a vital element of the journalistic and news dissemination process in recent years. Nowadays, social media is a part of everyday life. In addition to being a personal space for people, There is also a lot of news and information, both real and fake. So how to know The information received is true or false. This report is a real-to-fake separate trial of Twitter data collected in 2015 using a dataset MediaEval 2015 that was provided by Coursework Comp 6246. The MediaEval 2015 dataset consists of social media posts for which the social media identifiers are shared along with the post text and some additional characteristics of the post. The report is about analysis use cases, designing algorithm machine learning and evaluating results.

• Problem Characterization

Definition of fake posts:

i. Reposting of real multimedia, such as real photos from the past reposted as being associated with a current event
ii. Digitally manipulated multimedia
iii. Synthetic multimedia, such as artworks or snapshots presented as real imagery

The goal of this study is to assess five different machine learning algorithm designs for classifying posts in the "MediaEval 2015" dataset, describe each algorithm design for three strengths and weaknesses, then compare and rank all five algorithm designs. In terms of Algorithm design, including of pre-processing, feature selection, dimensionality reduction and a machine learning algorithm.

As a result, the characteristics that will be used will be based on natural language processing (NLP) and various kinds of word count vectorisation.

This report is working with static data and the algorithm's conclusions. The algorithm's implementation would not need to have any computational speed criteria. The F1 score is the primary performance metric used to compare different methods to this technique. The usage of confusion matrices and accuracy will also give further insight into the performance.

• Data Characterization

The dataset 'MediaEval 2015' is consists of a set of ground truth labels consisting of real and fake (Humour label should be treated as a Fake label). The original three classifications will be reduced to two: 'fake' and 'real'. The dataset is divided into two parts, trainset and testset, in the form of a text file.

```
<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: Information of Trainset

The trainset section contains all data of 14277 columns and has a data format, as shown in figure 1.

```
<class 'pandas.core.frame.DataFrame'>
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 2: Information of Testset

In the testset section, The data consists of 3755 columns and contains the following information as in figure 2.

They're both set consists of 18,032 columns and separating the contents of the ImageId(s) by grouping the same content together. The information can be divided into two parts.

```
imgs
boston                 546
bringback              131
columbianChemicals     185
elephant                13
livr                     9
malaysia               501
passport                46
pigFish                 14
sandyA                9695
sandyB                2621
sochi                  402
underwater             112
```

Figure 3: Post per image/event for trainset

Part of trainset contains 12 events which are boston, bringback, columbianChemicals, elephant, livr, malaysia, passport, pigFish, sandyA, sandyB, sochi, underwater. Shown in figure 3. From the information it is evident that The information is biased. The total information about SandyA and SandyB is 12316 posts. This means that repeating keywords like 'sandyA' and 'SandyB' in these articles would be learnt as characteristics that would not apply to any other occurrences.

```
imgs
eclipse        277
garissa         77
nepal         1353
samurai        218
syrianboy     1769
varoufakis      61
```

Figure 4: Post per image/event for testset

Part of testset consists of 6 events which are eclipse, garissa, nepal, samurai, syrianboy, varoufakis. Shown in Figure 4. It is the same a trainset. The information is biased by The 'nepel' and 'syrianboy' data are a total of 3122 posts which mean those two events accounted for about 83% of the total.

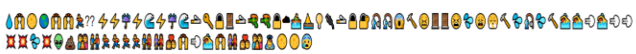


Figure5: icon in posts

Due to the noise in the data, the language detection is likely to be inaccurate. Furthermore, the characters 'amp;' and 'n' such as (':', ';', '[', ']', '"', '"', '(', ')', '.', '?', '#', '@', '...') occur at random in certain tweets. The effect of this noise can also be seen in the fact that Google's language identification failed to recognize the language of the following tweet, which was full of emojis and misspelled words: such as figure 5.

II. Algorithm design

- Data Preprocessing

The first process to do in data preprocessing is data cleaning by dropping some row that does not affect the training dataset process e.g. tweetId, userId, username. Next, because the task is to create a binary classifier, the following step was to modify the label 'humor' to 'fake.'

Therefore the data is highly biased. As explained above, on some events such as 'SandyA', 'SandyB', 'Nepal', 'Syrianboy'. Most of which are caused by retweets of the same post. To keep the data from being too biased, delete duplicate posts. from the experimental training In conclusion, deleting retweets has a positive effect on the performance of the result .thus reasonable

Emojis are another common element to consider in tweets, as previously discussed. Because emojis don't provide a lot of information, removing them would minimize symbolic noise in the data and improve the textual substance of the tweet.

The real multimedia is conveyed as a link within the tweet text. Because such links would be useless to the categorization system, all links in the tweet content were eliminated. Further cleaning was done to remove nonsensical symbols by deleting 'amp;' and 'n' such as (':', ';', '[', ']', '"', '"', '(', ')', '.', '?', '#', '@', '...') from the text, since they appeared in a large number of tweets as noted in data characterization[16].

Next is, after eliminating all retweets, the remaining @username (tag name) were also removed because they don't transmit much information and might be utilized by both fake and real postings[16].

Before going on to the next stage, a fast whitespace repair was conducted because deleting symbols and text from a tweet sometimes leave excessive whitespace.

The next step was to eliminate stopwords. Before being removed, the list of stopwords was expanded with punctuation. The result was assigned to a new column called 'filterPosts.' After that, the data in 'filterPosts' was lemmatized and sent to 'Lemmatising-nltk.' [20].After analyzing the data, it appeared that several preparation processes were unnecessary. One of them is spelling correction, as erroneous spelling appeared to hint to the post is a fake. Thus the original spelling was kept such that the algorithm could have a better prediction[16].

In part of multiple languages, There are many features in non-English, so it is important and will affect the training model process. The report decided to keep this information and use it as a separate feature.

The last step for preprocessing, After all of the preceding procedures have been completed; the next step is to deal with the resulting data bias, after going over the data from each procedure. Labelling is used to segregate the generated trainset dataset, resulting in the following: 7899 posts with the label 'false' and 3651 posts with the label 'real.' To establish equality, a new dataset was created by picking 3651 posts at random from the label 'fake' and merging them with the label 'real' to create a total of 7302 posts.

- Algorithm Training model

Find information and experiment from the case study Although understanding algorithm It's just that it's hard to tell if the algorithm that is being used is suitable for the provided dataset. Various potential combinations were investigated and tried with the application of the preprocessing processes outlined in the preceding section in order to determine the optimal vectorisation strategy and training model. Algorithm for the trained model has been chosen to test a total of 6 together.

i. MultinomialNB

Due to its computational economy and reasonably strong results, Multinomial Naive Bayes is frequently employed for text categorization issues like this [5]. The multinomial Naive Bayes classifier is good for discrete feature classification [7]. Normally, integer feature counts are required for the multinomial distribution. Fractional counts, such as TF-IDF, may also function in practice.

ii. BernoulliNB

Another helpful nave Bayes model is Bernoulli Nave Bayes. The model assumes that the characteristics are binary (0s and 1s) in character. Text classification with the 'bag of words' model is an application of Bernoulli Nave Bayes classification[8].

iii. PassiveAggressiveClassifier

For large-scale learning, passive-aggressive algorithms are commonly utilized. It's one of the few 'online-learning algorithms' on the market. In contrast to batch learning, where the full training dataset is used at once, online machine learning algorithms take the input data in a sequential sequence and update the machine learning model step by step. This is highly beneficial in circumstances where there is a large quantity of data and training the full dataset is computationally impossible due to the sheer bulk of the data [9]. It operates by responding passively to accurate classifications and aggressively to any misclassifications.

iv. SGDClassifier

The Stochastic Gradient Descent (SGD) to regularized linear techniques can aid in the development of a classification and regression estimator. [10] Unlike gradient descent, which analyzes the entire training data, stochastic gradient descent considers just 1 random point when modifying weights. As a result, when dealing with big data sets, stochastic gradient descent is substantially quicker than gradient descent.

v. SVC

A Linear SVC (Support Vector Classifier) is designed to fit to the data that supply and provide a "best fit" hyperplane that divides or categorizes the data. Following that, the classifier may input some characteristics to check the "predicted" class once the hyperplane is obtained. As a result, this algorithm is a good fit for our needs[11].

vi. AdaBoostClassifier

An AdaBoost classifier is a meta-estimator that starts by fitting a classifier on the original dataset, then fits further copies of the classifier on the same dataset, but adjusts the weights of poorly classified instances such that future classifiers focus more on difficult situations[13]

- Feature Processing

Three different vectorisation algorithms were tested and analyzed for feature extraction [1].

i.Bag-of-Words

The bag-of-words model is a representation for natural language processing and information retrieval that simplifies things (IR). A text is represented in this paradigm as a bag of its words, which ignores syntax and even word order while maintaining multiplicity. Computer vision has also employed the bag-of-words concept. The bag-of-words model is widely used in document classification approaches, where the occurrence of each word is utilized for training a classifier[12].

ii.N-Gram

A n-element continuous sequence from a given sets from a given sample of text or audio is known as an n-gram (Q-gram). The elements might be phonemes, syllables, letters, words, or base pairs, depending on the application. n-grams are often retrieved from a text or audio corpus. N-grams are also known as shingles when the components are words[14].

iii.TF-IDF

The TF-IDF (term frequency-inverse document frequency) statistic examines the relevance of a word to a document in a collection of documents. As a weighting factor, it's often utilized in information retrieval, text mining, and user modelling searches[16]. The amount of times a word appears in a document directly affects the TF–IDF value. But is offset by the number of documents in the corpus that include the term. This helps compensate for certain words appearing more frequently than others. One of the most prevalent term-weighting techniques currently is TF–IDF.

- Algorithm Designs

From the process from the beginning, data preprocessing is used to set up a dataset for the training model. Algorithm designs that can classify posts within the dataset. by matching Feature Processing and Algorithm Training model. Resulting in the results as shown in Table 1.(For Table row Confusion Matrix can find more information in APPENDIX [2])

Init	Clf	Confusion Matrix	F1 Score	Acc
Bag-of-Words	MultinomialNB	✗	0.433	0.482
Bag-of-Words	BernoulliNB	✗	0.479	0.486
Bag-of-Words	PassiveAggressiveClassifier	✓	0.85	0.806
Bag-of-Words	SGDClassifier	✓	0.856	0.806
Bag-of-Words	SVC	✗	0.793	0.659
Bag-of-Words	AdaBoostClassifier	✗	0.807	0.677
N-Gram	MultinomialNB	✗	0.546	0.463
N-Gram	BernoulliNB	✗	0.407	0.427
N-Gram	PassiveAggressiveClassifier	✗	0.64	0.509
N-Gram	SGDClassifier	✗	0.48	0.428
N-Gram	SVC	✗	0.846	0.759
N-Gram	AdaBoostClassifier	✗	0.675	0.525
TF-IDF	MultinomialNB	✓	0.9	0.862
TF-IDF	BernoulliNB	✗	0.445	0.468
TF-IDF	PassiveAggressiveClassifier	✓	0.854	0.811
TF-IDF	SGDClassifier	✓	0.896	0.855
TF-IDF	SVC	✗	0.796	0.669
TF-IDF	AdaBoostClassifier	✗	0.807	0.677

Tabel 1: Algorithm Design and Result

From the results of the experiment that came out, 5 algorithm designs were obtained. that can classify the dataset 'MediaEval2015' which are :

1) PassiveAggressiveClassifier + Bag-of-Words
2) SGDClassifier + Bag-of-Words
3) MultinomialNB + TF-IDF
4) PassiveAggressiveClassifier + TF-IDF
5) SGDClassifier + TF-IDF

Those 5 algorithm designs used same data preprocessing

III. Evaluation

This section will cover three strengths and three weaknesses, as well as compare and contrast all five algorithm designs. Rank the 5 algorithm designs in order of task suitability. These rankings are based on three strengths and three weaknesses for each algorithm design, using the F1 score and accuracy as evidence.

- Rank and 3 strengths & weaknesses

Rank 1 MultinomialNB + TF-IDF

3 strengths

- MultionmialNB suitable with a feature section with Fractional counts such as TF-IDF
- MultionmialNB. It is highly scalable and can easily handle large datasets.[21]
- Multionmial NB method is a strong tool for analyzing text input and solving issues with numerous classes.[21]

3 weaknesses

- This algorithm's prediction accuracy is lower than that of other probability algorithms.[21]
- It isn't appropriate for regression. Multionmial Naive Bayes technique can only be used to classify textual input and cannot be used to predict numerical values.[21]
- MultionmialNB is not suitable for bag-of word other feature

Rank 2 SGDClassifier + TF-IDF

3 strengths

- TF-IDF help sort data into categories, as well as extract keywords, count of corpus. This reduces the problem of repeating words in tweet text.
- SGD optimization allow support vector machine to learn over a small batch of data for large scale data.
- Because only one sample is processed at a time, it is computationally efficient.[22]

3 weaknesses

- SGDClassifier require appropriate hyperparameter tuning to achieve the best performance.[19]
- TF-IDF is only useful as a lexical level feature
- TF-IDF Cannot capture semantics

Rank 3 PassiveAggressiveClassifier + TF-IDF

3 strengths

- PAC algorithm has proven to be a very effective and popular method for online learning to solve many problems in the real world [17]
- PAC algorithm perfect for classifying massive streams of data
- Some basic metric to extract the most descriptive terms in a document

3 weaknesses

- The inverse document frequency computation causes a "zero value problem." If the word of interest appears in all papers, the tf-idf value will be 0 by default.[24]
- Do not know the relationship between words
- the feature vector is significantly increased due to amount of different word

Rank 4 SGDClassifier + Bag-of-Words

3 strengths

- Because of its simplicity, Bag of Words is still commonly utilized. NLP researchers frequently start using Bag of Words to gain a sense of how well their work is performing before moving on to better word embeddings.[18]
- Ease of implementation [19]
- Because the network processes just one training sample, SGDClassifier is easy to put into memory.

3 weaknesses

- SGD is sensitive to feature scaling.[19]
- The steps taken towards the minima are highly loud due to frequent updates. This can cause the gradient to slant in unexpected directions.[22]
- Because of the noisy stages, convergence to the loss function minima may take longer.[22]

Rank 5 PassiveAggressiveClassifier + Bag-of-Words

3 strengths

- Keep the model and don't make any adjustments if the forecast is right; the data, in this case, is insufficient to generate any model adjustments.[9]
- Support steaming data real-time [23]
- Simple and good speed performance

3 weaknesses

- Stop words include often used terms like 'the,' 'is,' and so on. When such terms are removed from a lexicon, the vectors get smaller [18].
- There is no word count that may be repeated. For this dataset, the outcome is not as excellent as TF-IDF.
- To produce vectors for huge documents, consider using the Bag of Words approach. The resulting vectors will be big in size and include an excessive number of null values, resulting in sparse vectors [18].

- Compare all 5 algorithm designs

From the ranking comparison, it can be seen that TF-IDF is more suitable for checking fake tweets than Bag-of-word because it counts repetitions which makes it more effective at correcting repetitions. up from tweet text. MultinomialNB + TF-IDF has the best performance. Due to the compatibility of feature sections and machine learning algorithms. In terms of machine learning algorithms from the results can tell SGDClassifier is more suitable for filtering text effect than PassiveAggressiveClassifier, and Support vector machines can learn over a small batch of data for large scale data using SGD optimization.

IV. Conclusion

Fake posts. It's a cause for misunderstandings and sometimes severe consequences. It is crucial to find information to have an algorithm to differentiate between post real or fake. to increase credibility. for this report, has been told how to Create an algorithm design in detail from how to do it. Data preprocessing until design algorithm design for use with datasets 'MediaEval2015' with the results of the resulting experiment. In every part of the design algorithm. They are all equally important. Any change in any part may cause a different result of the algorithm. As a result of the assignment, it can be determined that TF-IDF is the best method for addressing these postings or new datasets, whether real or fake. The key factor to this is because TF-IDF has a word counting process and reduces the problem of duplicates that can cause confusion. The matching algorithm is equally important. The results can tell that the matching design algorithm appropriate and compatible get the best results such as MultinomialNB + TF-IDF algorithm design.

For the idea for future improvement. A new research area of NLP is involved with the deep learning method. In deep learning, the text input was fed directly to the network. The network will extract the feature automatically instead of using a handcrafted feature in a typical machine learning method. For an example of feature extraction, TensorFlow Keras library provides an embedding layer that encodes the text input to vector, which adapt itself along with the learning process. In a classification algorithm, a deep learning model can be built with various architecture depending on the complexity of the task. However, creating a deep learning model and hyperparameter tuning is difficult to master.

V. References

[1] scikit-learn: Feature extraction: [Online].Available: https://scikit-learn.org/stable/modules/feature_extraction.html [Accessed January, 10 2022].

[2] Samarth Agrawal: Understanding the Confusion Matrix from Scikit learn: towards data science: Jan 1, 2021: https://towardsdatascience.com/understanding-the-confusion-matrix-from-scikit-learn-c51d88929c79

[3] Thomas Wood: F-Score [Online].Available: https://deepai.org/machine-learning-glossary-and-terms/f-score [Accessed January 12, 2022]

[4] MediaEval Benchmarking Initiative for Multimedia Evaluation: MediaEval 2015 [Online]. Available: http://www.multimediaeval.org/mediaeval2015/ [Accessed January 8, 2022]

[5] Frank, Eibe; Bouckaert, Remco R.: Naïve Bayes for text classification with unbalanced classes : Springer-Verlag Berlin Heidelberg 2006 : DOI https://doi.org/10.1007/11871637_49

[6] GOWTHAM CH : How I was using Naive Bayes (Incorrectly) till now — Part-1: towards data science Jan 30, 2021 : [Online] https://towardsdatascience.com/how-i-was-using-naive-bayes-incorrectly-till-now-part-1-4ed2a7e2212b

[7] scikit-learn: MultinomialNB: [Online].Available: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html [Accessed January, 14 2022].

[8] scikit-learn: BernoulliNB: [Online].Available: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html [Accessed January, 14 2022].

[9] GeeksforGeeks : Passive Aggressive Classifiers: [Online].Available: https://www.geeksforgeeks.org/passive-aggressive-classifiers/ [Accessed January, 14 2022].

[10] Vinay Patlolla : How to make SGD Classifier perform as well as Logistic Regression using parfit :toward data science[Online] Available https://towardsdatascience.com/how-to-make-sgd-classifier-perform-as-well-as-logistic-regression-using-parfit-cc10bca2d3c4 [Accessed January, 14 2022].

[11] Linear SVC Machine learning SVM example with Python :[Online] https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/ [Accessed January, 14 2022].

[12] Jason Brownlee : A Gentle Introduction to the Bag-of-Words Model : Deep Learning for Natural Language Processing: October 9, 2017 : https://machinelearningmastery.com/gentle-introduction-bag-words-model/

[13] scikit-learn: AdaBoostClassifier: [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html [Accessed January, 14 2022].

[14] Speech and Language Processing. Daniel Jurafsky & James H. Martin.:CH3 N-gram Language Models: Copyright © 2021. All rights reserved. Draft of December 29, 2021. https://web.stanford.edu/~jurafsky/slp3/3.pdf

[15] William Scott : TF-IDF from scratch in python on a real-world dataset.: towards data science[Online]: Feb 15, 2019 :

https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

[16] Christina Boididou, Symeon Papadopoulos, Markos Zampoglou, Lazaros Apostolidis : Detection and visualization of misleading content on Twitter : March 2018International Journal of Multimedia Information Retrieval 7(3) https://link.springer.com/article/10.1007/s13735-017-0143-x

[17] K.Nagashri, J. Sangeetha : Fake News Detection Using Passive-Aggressive Classifier and Other Machine Learning Algorithms: 13 June 2021 : pp 221-233 Electronic ISSN 1876-1119 : https://link.springer.com/chapter/10.1007%2F978-981-33-6987-0_19#enumeration

[18] Naman Swarnkar: Bag of Words: Approach, Python Code, Limitations : quantinsti [Online] Aug 05, 2020 : https://blog.quantinsti.com/bag-of-words/

[19] scikit-learn: Stochastic Gradient Descent: [Online]. Available: https://scikit-learn.org/stable/modules/sgd.html [Accessed January, 17 2022].

[20] Stemming and Lemmatization - NLTK Book [Online]: https://www.nltk.org/book/ch03.html [Accessed January, 17 2022].

[21] Shriram : Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2022 [Online] : JAN 3, 2021 : https://www.upgrad.com/blog/multinomial-naive-bayes-explained/

[22] Advantages and Disadvantages of Stochastic Gradient Descentv: [Online] Aug. 24, 2020 : https://www.asquero.com/article/advantages-and-disadvantages-of-stochastic-gradient-descent/

[23] Jing Lu, Peilin Zhao & Steven C. H. Hoi : Online Passive-Aggressive Active learning : Machine Learning volume 103, pages141–183 (2016)

https://link.springer.com/article/10.1007/s10994-016-5555-y

[24] Jack Scott : What's in a word? : towards data science: Nov 1, 2020 [Online]

https://towardsdatascience.com/whats-in-a-word-da7373a8ccb

Appendix

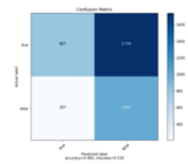
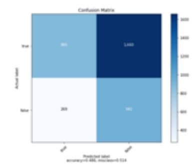
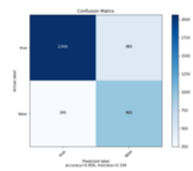
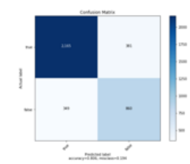
Init	Clf	Confusion Matrix	F1 Score
Bag-of-Words	MultinomialNB		TP: 807 FP: 207 TN: 1002 FN: 1739 f1: 0.453
Bag-of-Words	BernoulliNB		TP: 886 FP: 269 TN: 940 FN: 1660 f1: 0.479
Bag-of-Words	PassiveAggressiveClassifier		TP: 2066 FP: 249 TN: 960 FN: 480 f1: 0.850
Bag-of-Words	SGDClassifier		TP: 2165 FP: 349 TN: 860 FN: 381 f1: 0.856

Table2. Confusion Matrix

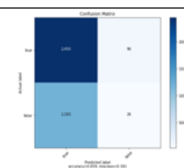
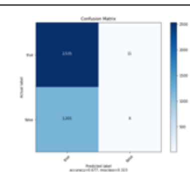
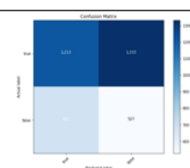
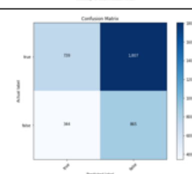
Bag-of-Words	SVC		TP: 2450 FP: 1183 TN: 26 FN: 96 f1: 0.793
Bag-of-Words	AdaBoostClassifier		TP: 2535 FP: 1201 TN: 8 FN: 11 f1: 0.807
N-Gram	MultinomialNB		TP: 1213 FP: 682 TN: 527 FN: 1333 f1: 0.546
N-Gram	BernoulliNB		TP: 739 FP: 344 TN: 865 FN: 1807 f1: 0.407

Table3. Confusion Matrix

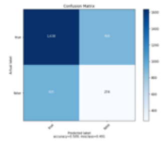
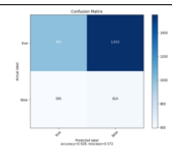
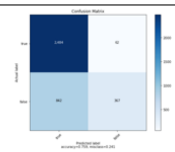
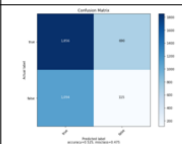
N-Gram	PassiveAggressiveClassifier		TP: 1638 FP: 935 TN: 274 FN: 908 f1: 0.640
N-Gram	SGDClassifier		TP: 993 FP: 595 TN: 614 FN: 1553 f1: 0.480
N-Gram	SVC		TP: 2484 FP: 842 TN: 367 FN: 62 f1: 0.846
N-Gram	AdaBoostClassifier		TP: 1856 FP: 1094 TN: 115 FN: 690 f1: 0.675

Table4, Confusion Matrix

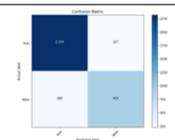
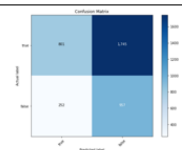
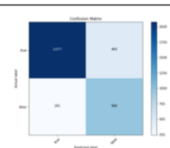
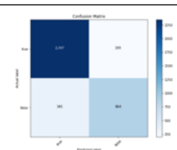
TF-IDF	MultinomialNB		TP: 2319 FP: 290 TN: 919 FN: 227 f1: 0.900
TF-IDF	BernoulliNB		TP: 801 FP: 252 TN: 957 FN: 1745 f1: 0.445
TF-IDF	PassiveAggressiveClassifier		TP: 2077 FP: 241 TN: 968 FN: 469 f1: 0.854
TF-IDF	SGDClassifier		TP: 2347 FP: 345 TN: 864 FN: 199 f1: 0.896

Table5, Confusion Matrix

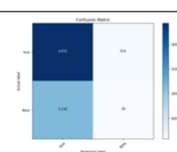
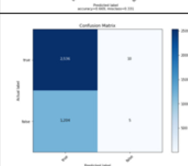
TF-IDF	SVC		TP: 2432 FP: 1130 TN: 79 FN: 114 f1: 0.796
TF-IDF	AdaBoostClassifier		TP: 2536 FP: 1204 TN: 5 FN: 10 f1: 0.807

Table6, Confusion Matrix