Fake News Detection Using Machine Learning Algorithms

Akshat Wadhwa
IIIT Delhi
New Delhi,India
akshat19231@iiitd.ac.in

Shruti Jha
IIIT Delhi
New Delhi, India
shruti19274@iiitd.ac.in

Tarini Sharma
IIIT Delhi
New Delhi, India
tarini19451@iiitd.ac.in

1. Introduction

With the rise in popularity of the Internet, social media platforms like Facebook and Twitter have become new sources of information for people. But these online media platforms are being manipulated by certain entities to spread distorted facts. This phenomenon is known as fake news [6]. Fake news can lead to serious problems like the spread of misinformation about vaccines during the COVID-19 pandemic that led to issues in maintaining public health [5]. In this project, we aim to use machine learning algorithms for automated classification of news articles as fake or real. We aim to explore various textual properties in natural language processing on the dataset, which we will use to train different ML models and ensemble methods and evaluate their performance to determine the best model for this learning task.

2. Related Work

In [8], the authors performed very minimal preprocessing: lowercasing the dataset and dropping stopwords. In order to train the models, a 5-fold split was also performed (80% training, 20% test). The models used are CNN, XGBoost, SVM, Naive Bayes and Random Forest. After training using the Keras API and TensorFlow module, the Accuracy, Precision, Recall and F1-Score are calculated (Figure 5). In paper [7], in addition to lowercasing and removing stopwords, empty strings and missing labels are also handled. The models used are SVM, CNN and LSTM on the metrics Accuracy, Precision and Recall (Figure 6). In [9], the preprocessing is done using word2vec (after dropping the stopwords) from the gensim library. The models used are Feedforward Neural Network, CNN with one convolution layer, CNN with multiple convolution layers and LSTM on the metrics Accuracy, Precision, Recall and F1-score. The metrics are calculated on two different aspects of this dataset: in the first part, the models are trained using the full text of each article and in the latter part models are trained on the title of each article (Figures 7 and 8).

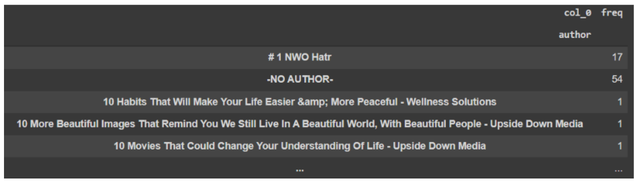
3. Dataset and Evaluation

3.1. Dataset Information

We have used the dataset available on Kaggle [1], which has two files, train.csv (20387 training samples) and test.csv (5127 testing samples). The testing data has four attributes: id, title, author, text and the training data has the additional column of class label (0 for reliable/real news and 1 for unreliable/fake news). There is a slight imbalance in the class distribution in training data, with 10361 samples of class 0 and 7850 samples of class 1 (see Figure 9).

3.2. Pre-processing and feature extraction

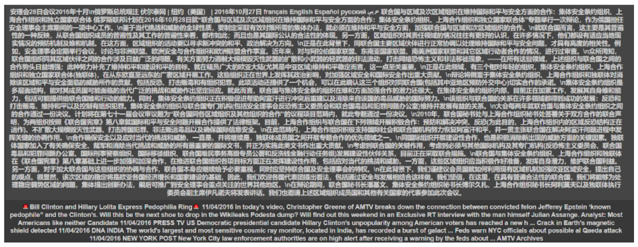
Before extracting features, we remove the NaN values and data samples containing only white spaces or "\n". The entire text is converted to lowercase and English letters from a-z are kept, everything else replaced with space. Then stemming is performed to reduce the words in the dataset to their basic roots. Next we drop stop words like "an", "the" etc. [2] since they do not carry any significant meaning. Lastly, we vectorize the document by sklearn's TF-IDF Vectorizer [10]. We explored three different vectorizers for feature extraction before choosing TF-IDF because it gave best results on validation set, as seen in Figure 10. Author attribute had 1957/20800 NaN values in the train set and 500/5200 NaN values in test set. Instead of removing these, replaced NaN values with '-no author-' which helped increase the dataset size. After preprocessing, we



col_0	freq
author	
# 1 NWO Hatr	17
-NO AUTHOR-	54
10 Habits That Will Make Your Life Easier &amp; More Peaceful - Wellness Solutions	1
10 More Beautiful Images That Remind You We Still Live In A Beautiful World, With Beautiful People - Upside Down Media	1
10 Movies That Could Change Your Understanding Of Life - Upside Down Media	1
...	...

Figure 1. Entries with no-author in the author field

had 5040 test samples and 20126 training samples, upon which stratified 5 fold cross validation was performed with 4 folds[16100 samples] used to train the model and 1 fold


安理会28日会议2016年十月\n俄罗斯总统观注 伏尔泰网 | 纽约（美国） | 2016年10月27日 français English Español русский عربي 联合国与区域及次区域组织在维持国际和平与安全方面的合作：集体安全条约组织、上海合作组织和独立国家联合体 俄罗斯联邦计划在2016年10月28日就"联合国与区域及次区域组织在维持国际和平与安全方面的合作：集体安全条约组织、上海合作组织和独立国家联合体"专题举行一次辩论，作为俄国担任安全理事会主席期间的一项中心工作。\n鉴于当代挑战和威胁的全球性质，要制定采取有效对策所需的集体办法，就必须在维持和平与安全方面，加强联合国与区域和次区域组织的合作。\n就联合国而言，这主要是其普遍性的一种反映，从联合国组织成员的普遍性及其工作的普遍性来看，都市如此；而且也是其国际公认的合法性的体现。另一方面，区域组织对其责任领域的情况往往有更好的认识，在许多情况下，他们都设有适应当地现实情况的预防机制及维和机制。在这方面，区域组织的活动要以寻求新冲突的和平、政治解决为方向。\n正是在此背景下，同联合国主要区域伙伴进行正常协调以处理维持国际和平与安全问题，才具有高度的相关性。例如，安全理事会定期举行会议，讨论与非洲联盟、欧洲安全与合作组织和欧洲联盟合作事宜。近年来，对与阿拉伯国家联盟、东南亚国家联盟、南美洲国家联盟和其它区域行动者合作的情况，进行过审查。\n众所周知，联合国组织同其区域伙伴之间的合作涉及日益广泛的问题。有关方面努力遏制大规模毁灭性武器的扩散和小武器的轻武器的非法流动，打击网络恐怖主义和非法移徙现象，——在所有这些领域，上述组织与联合国之间的合作势头日益增强；此种努力补充了维持和平和建设和平的目标。就在幅员广大的欧亚大陆(尤其是中亚区域)维持和平稳定而言，这一点至关重要。\n正是在此领域，有三个相对年轻的组织，集体安全条约组织、上海合作组织和独立国家联合体(独联体)，在从东欧直至远东的广袤区域开展工作。这些组织正在世界上发挥其政治影响，对加强区域安全和国际安全作出重大贡献。\n辩论将侧重于集体安全条约组织、上海合作组织和独联体对消除该区域和平与安全面临的威胁所作的贡献，包括反恐、打击贩毒和有组织犯罪。此项活动还提供了一个机会，可以在此确认这三个组织对同联合国(包括其中亚地区预防外交中心)切实合作的承诺。\n集体安全条约组织是多层面结构，能对其成员国可能面临的当代广泛的挑战和威胁作出坚定回应。就此而言，联合国与集体安全条约组织在维和方面加强合作的潜力还很大。在集体安全条约组织内部，目前正在加紧工作，发展其自身维和能力，包括可能提供给联合国维和行动的能力。同时，集体安全条约组织正在积极促进帮助阿富汗进行冲突后重建以及消除来自该国的毒品威胁的国际努力。\n该组织与联合国的关系在许多领域都得到成功的发展：反恐和打击贩毒、维持和平以及控制有组织犯罪。集体安全条约组织与联合国专门机构(包括安全理事会反恐怖主义委员会和联合国毒品和犯罪问题办公室)维持并发展有益的关系。\n大会每两年就联合国与集体安全条约组织之间的合作通过一份决议。计划将在第七十一届会议审议题为"联合国同各区域组织及其他组织的合作"的议程项目范畴内，就此专题通过一份决议。\n2010年，联合国秘书处与上海合作组织秘书处签署关于双方合作的联合声明，为两组织按照《联合国宪章》第八章就国际和平与安全问题开展合作提供了法律框架。目前，上海合作组织与联合国在下列领域开展积极合作：预防和解决冲突、反恐(为此目的，上海合作组织内的区域反恐结构正在运作)、不扩散大规模毁灭性武器、打击跨国犯罪、非法贩运毒品以及确保国际信息安全。\n在此范畴内，上海合作组织积极支持国际社会和联合国机构努力恢复阿富汗和平，并一贯主张联合国在解决阿富汗问题进程中发挥关键的协调作用。\n合作确保安全以及应对当代的挑战和威胁，一直是，并将继续是，独联体成员国之间开展专题合作的优先领域之一。\n同国际组织开展建设性合作，也是积极消除新出现的威胁方面的关键因素。独联体国家加入了有关确保安全、裁军和消除当代挑战和威胁的所有最重要的国际文书，并正为实施此类文书作出重大贡献。\n考虑到联合国的关键作用，考虑到必须与其他国际机构及其专门机构(反恐怖主义委员会、联合国毒品和犯罪问题办公室、国际刑事警察组织、国际移徙组织、联合国难民事务高级专员公署和反洗钱金融行动任务组)发展建设性伙伴关系，目前正在采取联合措施。\n联合国与集体安全条约组织、上海合作组织和独联体在《联合国宪章》第八章基础上进一步加强和加深合作，在推进联合国组织各项目标方面正在发挥建设性作用，包括应对当代的挑战和威胁。一方面，这些区域组织应当积极作好准备，发挥自身潜力，维护联合国利益。另一方面，对于加大联合国与这些组织的协调与合作，联合国本身应继续给予必要重视，同时应坚持联合国及安全理事会的特权。\n在此背景下，我们谨建议会员国就如何利用现有区域机制加强欧亚区域安全，提出自己的观点。很显然，该次区域的稳定将是欧亚各国经济增长和国家建设的基础，因此，我们欢迎各国代表团提出看法，包括通过安全与发展相结合这样做。我们坚信，在这里，在具有普遍合法性的联合国，我们将能够为处理稳定薄弱区域的问题，集体提出创新办法，嗣后可推广到安全理事会重点关注的世界其他地区。\n在辩论期间，联合国秘书长潘基文、集体安全条约组织秘书长博尔久扎、上海合作组织秘书长阿利莫夫以及独联体执行委员会副主席伊凡诺夫将发表讲话。我们也邀请上述区域组织成员国和其他有关国家的代表参加此次会议。

🚨Bill Clinton and Hillary Lolita Express Pedophilia Ring🚨 11/04/2016 In today's video, Christopher Greene of AMTV breaks down the connection between convicted felon Jefferey Epstein 'known pedophile" and the Clinton's. Will this be the next shoe to drop in the Wikileaks Podesta dump? Will find out this weekend in an Exclusive RT interview with the man himself Julian Assange. Analyst: Most Americans like neither Candidate 11/04/2016 PRESS TV US Democratic presidential candidate Hillary Clinton's unpopularity among American voters has reached a new h ... Crack in Earth's magnetic shield detected 11/04/2016 DNA INDIA The world's largest and most sensitive cosmic ray monitor, located in India, has recorded a burst of galact ... Feds warn NYC officials about possible al Qaeda attack 11/04/2016 NEW YORK POST New York City law enforcement authorities are on high alert after receiving a warning by the feds about ... AMTV Archives

Figure 2. Texts with Chinese and special characters

kept for validation[4026 samples]. Some texts are not in English (Chinese, or have some symbols).

3.3. Evaluation Metrics

Evaluation metrics used are accuracy, recall, precision and F1 score. We also plot the confusion matrix and AUC-ROC curves of the models to better compare them.
To evaluate models, a single-number evaluation metric is needed. Since F1-score is the harmonic mean of precision and recall, we choose F1 score as a satisficing metric(threshold=0.6) and accuracy as the optimizing metric. Formulas to determine these metrics are in Figure 11.

4. Analysis and Progress

4.1. EDA

We checked the separability of training data by reducing the dimensions of data to 2 using t-SNE and plotting the scatterplot (Figure 12). We see that the data is not very well separated and there are quite a few samples of class 0 overlapping with samples of class 1. We also plotted the word clouds for the reliable news articles and the fake news articles as shown in Figures 13 and 14. There are some general words like "one", "time", "many" present in both real and fake news in equal weightage. To decide the number of features to extract in the tf-idf vectorizer, we plotted the mean dev set (for 5-fold) accuracy with Naive Bayes model and got the best result for 3000 features (Figure 15).

4.2. Models and Learning methods

We have explored 6 ML models: Naives bayes (baseline), Logistic Regression [learning methods used: LBFGS (variant of Newton's method using approximated Hessian matrix) and SGD], Passive-Aggressive Classifier, SVM [learning methods used: Sequential Minimal Optimisation (SMO) and SGD], Decision Tree and Multi-layer Perceptron [learning methods used: SGD and Quasi-Newton method]. We also explored an RNN based (LSTM) model. [3] We also explored two ensemble methods: XGBoost (boosting method) and Random Forest (bagging method).
We used stratified k-fold cross validation (k=5) while training all models and selected the model with best dev set accuracy while satisfying threshold of 0.9 on f1 score.

4.2.1 Hyperparameter Tuning

For hyperparameter tuning, we used Bayesian Optimization implemented in sklearn to get the best hyperparameters for each model (see Figure 16). Since we did not have enough RAM to fit the whole dataset, we used PCA decomposition to reduce the data dimensionality to 100 dimensions.

4.3. Challenges and their Resolutions

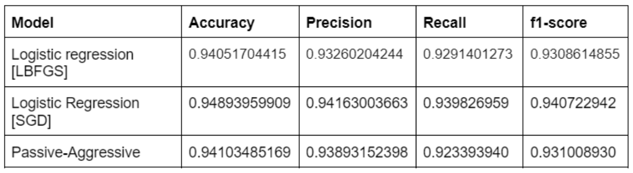
Overfitting was detected using learning curves. (see fig. 17-24) Naive Bayes has a high bias of about 10% and a low variance of about 2%. The model is underfitting and so we use advanced models. Logistic regression[LBFGS] has a low bias of 0.01%(assuming desired performance to be 100% accuracy) and has a variance of 6%. Logistic regression[SGD] has a bias of 2% and variance of 3.4%. Passive Aggressive model has bias of 1.3% and variance of 4.7%. Random Forest model has a bias of 0.01% and variance of about 5%. XGBoost has 2% bias and 3% variance. All the models except Naive Bayes is fitting well to the data. To reduce overfitting while training, we tried to add regularization(L2 regularization for logistic regression, early stopping in SGD). We also tried decreasing the number of features which decreased the dev set accuracy as shown in EDA). Reducing the number of features to 100 increased bias by 14.%

4.4. Covariate shift check between validation set and test set

Preprocessed the train(and test) set, added a column with targetDataset=0(and 1) and dropped the classLabel from train dataset. Concatenated and shuffled the 2 datasets and used a RF classifier to classify combined dataset based on targetDataset. An ROC value of 0.505 was observed which implies that the datasets do not have much covariate shift.

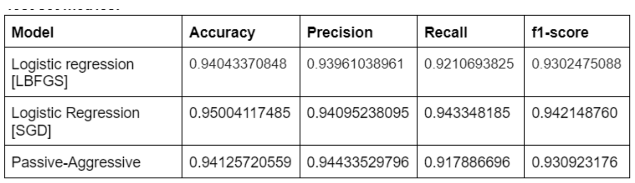
4.5. Sentiment Analysis

Sentiment analysis of 'text' attribute using vader and textblob. Using textblob, extracted features maxpos, minneg, mean_sentiment, mean_subjectivity. Textblob.sentiment.polarity returns polarity of text in the range [-1,1] where -1 denotes negative sentiment and +1 denotes positive sentiment. Textblob.sentiment.subjectivity returns the amount of personal opinion and factual information in the range [0,1] where 1 denotes that the text contains more personal opinion rather than factual information. [4] Maxpos is the maximum polarity over all sentences in the 'text' attribute of a single datasample. Minneg denotes the minimum polarity, mean_sentiment denotes the mean polarity. Mean_subjectivity is the mean of subjectivity over all sentences in the datasample 'text' attribute. Trained



Model	Accuracy	Precision	Recall	f1-score
Logistic regression [LBFGS]	0.94051704415	0.93260204244	0.9291401273	0.9308614855
Logistic Regression [SGD]	0.94893959909	0.94163003663	0.939826959	0.940722942
Passive-Aggressive	0.94103485169	0.93893152398	0.923393940	0.931008930

Figure 3. Metrics on the validation set



Model	Accuracy	Precision	Recall	f1-score
Logistic regression [LBFGS]	0.94043370848	0.93961038961	0.9210693825	0.9302475088
Logistic Regression [SGD]	0.95004117485	0.94095238095	0.943348185	0.942148760
Passive-Aggressive	0.94125720559	0.94433529796	0.917886696	0.930923176

Figure 4. Metrics on the test set

logistic regression(LBFGS) on the extracted features which results in the following metrics - Accuracy: 0.63107, Precision: 0.601618, Recall: 0.426114, F1-score: 0.498881431. Using vader, extracted features pos, neg, neu, compound, where pos denotes positive sentiment score, neg denotes negative sentiment score and neu denotes neutral sentiment score in the range [0,1] (all sum upto 1). Compound denotes the compound score calculated as the sum of normalized sentiment scores. Trained logistic regression(LBFGS) on the extracted features which results in Accuracy: 0.6184, Precision: 0.6898, Recall: 0.20828, F1-score: 0.31996.

Because of better performance of textblob, concatenated the 4 features from textblob with the 3000 features from TF-IDF and trained on Logistic Regression(LBFGS and SGD) and Passive Aggressive Classifier (shown in figure below).

5. Results and Interpretation

5.1. Results

Figures 45 and 46 contain the value of the metrics mentioned in the section 3 for the validation and test set. Figure 47 contains the accuracy of the models on the test set provided by the Kaggle competition from which the dataset was taken. [1] Gaussian Naive Bayes is underfitting as can be inferred from the learning curve and relatively low training, dev and test accuracy for Naive Bayes. Since we needed to submit our predictions on kaggle to get the accuracy, we decided to do a 80-20 train test split to test our models on other metrics (precision, recall, f1-score, roc curve, confusion matrix) With weighted accuracy as the optimizing metric and F1-score as the satisficing metric(with threshold 0.9), the best model is SVM(SMO) with a test accuracy of about 0.9548

5.2. Interpretation

Looking at dev set samples which are mislabelled as real, it was noticed that the dev set contains some key words(essential for a human classifying the text as fake or real) not present in the train set. Eg, in the validation sample with text “pokemon go players are inadvertently stopping people committing suicide in japan“, “pokemon” and “inadvertently” are not present in the features extracted using TF-IDF vectorizer. This implies we need to make the dataset more diverse in the future. Performance of gaussian naive bayes on the test set is better as comparable to paper [6] (difference in accuracy about 28.8%). Performance of SVM on test set is better than the papers [5] and [6] with difference in accuracy of about 35% and 32%. Performance of random forest and xgboost is better as compared to [6] (difference of about 16% and 6% respectively). This can be attributed to the fact that the authors performed very minimal preprocessing - lowercasing the whole dataset and dropping stopwords. Performance of LSTM is comparable to [8] (difference in performance of)
On comparing the AUC-ROC curves (see figures 25-34) for the various baseline (Gaussian Naive Bayes) and advanced models implemented, we see that the the area under the ROC curve for baseline model is less as compared to the the area of the other advanced models like logistic regression, SVM, etc., and the ensemble methods explored like XGBoost and Random Forest. This shows that the advanced models are performing better on the test set as compared to the baseline model, as we would have expected.
Consider class 0 (real news) as the negative class and class 1 (fake news) as the positive class. First analyzing the confusion matrix of our baseline model (Gaussian Naive Bayes), we get that the number of true positives and true negatives are almost equal, as well as the number of false positives and false negatives are also almost equal. Also the total number of samples that it is misclassifying is also very high. This shows that the baseline model is not performing very well on the test set, since it is equally classifying and misclassifying the test data samples (figure 35). For the other advanced models: Logistic regression, Passive Aggressive Classifier, SVM, etc., we can see that the number of true positives and true negatives are quite high, while the number of false positives and false negatives are relatively lower (see figures 36-44). Thus these advanced models give better evaluation metric scores on the test set as well.

6. Contributions

6.1. Deliverables

During the course of the project, the following deliverables were promised and derlivered by the corresponding team members:

Shruti - Promised deliverable: EDA, preprocessing, training+hyperparameter tuning of models (SVM, Decision Trees, MLP), model selection, ensemble approaches, drawing final conclusions+report writing.

Delivered deliverables: EDA, preprocessing, training+hyperparameter tuning of models (SVM, Decision Trees, MLP), model selection, ensemble approaches, drawing final conclusions+report writing.

Tarini - Promised deliverables: training+hyperparameter tuning of models (Logistic regression[LBFGS, SGD], Passive Aggressive Classifier, (XGBoost) ensemble approach), model selection, conclusion+report writing.

Delivered deliverables: EDA, training+hyperparameter tuning of models (Logistic regression[LBFGS, SGD], Passive Aggressive Classifier, XGBoost, Random Forest (ensemble approaches)), error analysis, covariate shift, sentiment analysis, model selection, conclusion+report writing.

Akshat - Promised deliverables: Preprocessing, training+hyperparameter tuning of models (Naive Bayes, LSTM), model selection, evaluation metrics, error analysis, drawing final conclusions+report writing.

Delivered deliverables: Pre-processing, Vectorization of data, training+hyperparameter tuning of models (Gaussian Naive Bayes and LSTM), evaluation metrics, drawing final conclusions+report writing.

6.2. File links

During the course of the project, the following files were made by the corresponding team members:
Tarini :

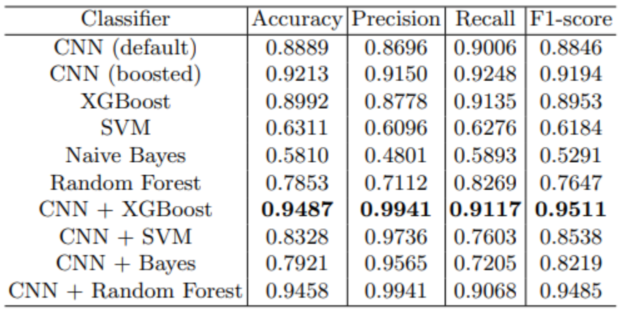
- Models(training + Hyperparameter tuning)
- Learning curves
- Error analysis
- Preprocessed train and test datasets

Akshat :

- TFIDF + Hyperparam tuning
- LSTM
- Count Vectorizer
- Word2vec

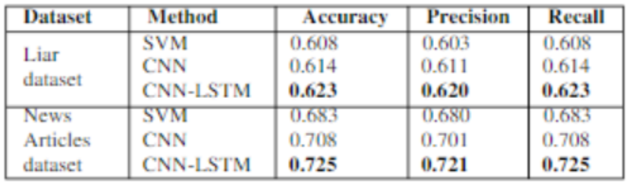
Shruti :

- EDA
- Hyperparameter Tuning
- Models (SVM, Decision Trees, MLP)
- Learning methods explored for models
- Learning curves



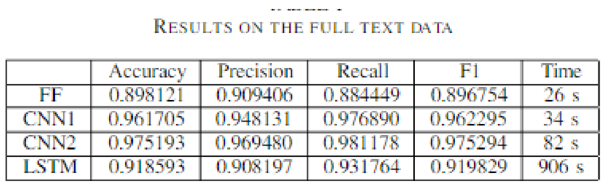
Classifier	Accuracy	Precision	Recall	F1-score
CNN (default)	0.8889	0.8696	0.9006	0.8846
CNN (boosted)	0.9213	0.9150	0.9248	0.9194
XGBoost	0.8992	0.8778	0.9135	0.8953
SVM	0.6311	0.6096	0.6276	0.6184
Naive Bayes	0.5810	0.4801	0.5893	0.5291
Random Forest	0.7853	0.7112	0.8269	0.7647
CNN + XGBoost	**0.9487**	**0.9941**	**0.9117**	**0.9511**
CNN + SVM	0.8328	0.9736	0.7603	0.8538
CNN + Bayes	0.7921	0.9565	0.7205	0.8219
CNN + Random Forest	0.9458	0.9941	0.9068	0.9485

Figure 5. Evaluation Metrics results in referenced paper [8]



Dataset	Method	Accuracy	Precision	Recall
Liar dataset	SVM	0.608	0.603	0.608
	CNN	0.614	0.611	0.614
	CNN-LSTM	**0.623**	**0.620**	**0.623**
News Articles dataset	SVM	0.683	0.680	0.683
	CNN	0.708	0.701	0.708
	CNN-LSTM	**0.725**	**0.721**	**0.725**

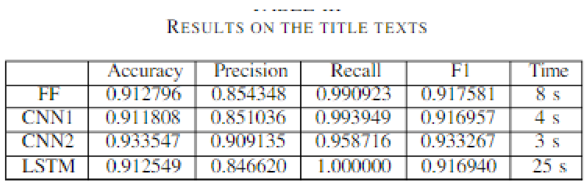
Figure 6. Evaluation Metrics results in the referenced paper [7]



RESULTS ON THE FULL TEXT DATA

	Accuracy	Precision	Recall	F1	Time
FF	0.898121	0.909406	0.884449	0.896754	26 s
CNN1	0.961705	0.948131	0.976890	0.962295	34 s
CNN2	0.975193	0.969480	0.981178	0.975294	82 s
LSTM	0.918593	0.908197	0.931764	0.919829	906 s

Figure 7. Metrics of the models trained only on text field as given in the referenced paper [9]



RESULTS ON THE TITLE TEXTS

	Accuracy	Precision	Recall	F1	Time
FF	0.912796	0.854348	0.990923	0.917581	8 s
CNN1	0.911808	0.851036	0.993949	0.916957	4 s
CNN2	0.933547	0.909135	0.958716	0.933267	3 s
LSTM	0.912549	0.846620	1.000000	0.916940	25 s

Figure 8. Metrics of the models trained on the title field as given in the referenced paper [9]

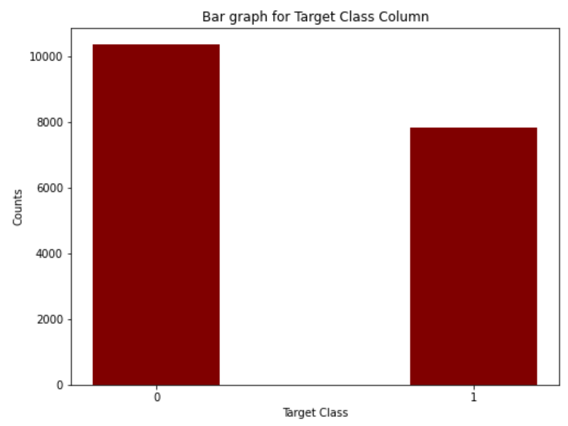
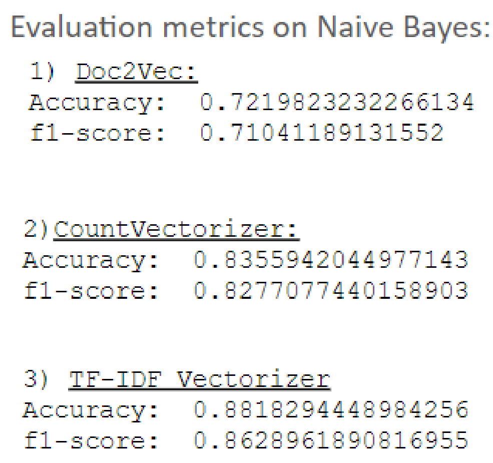


Figure 9. Bar Graph for Target class distribution (Class 0 is real news and Class 1 is fake news)



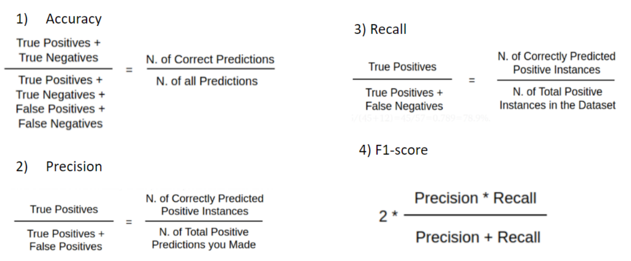
Evaluation metrics on Naive Bayes:

```
 1) Doc2Vec:
 Accuracy:   0.7219823232266134
 f1-score:   0.71041189131552


2)CountVectorizer:
Accuracy:   0.8355942044977143
f1-score:   0.8277077440158903


3) TF-IDF Vectorizer
Accuracy:   0.8818294448984256
f1-score:   0.8628961890816955
```

Figure 10. Validation accuracy using Baseline model of the three different vectorizers explored for feature extraction.



1) Accuracy

$$\frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} = \frac{\text{N. of Correct Predictions}}{\text{N. of all Predictions}}$$

2) Precision

$$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Predictions you Made}}$$

3) Recall

$$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Instances in the Dataset}}$$

4) F1-score

$$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 11. Formulae for the evaluation metrics used by us.

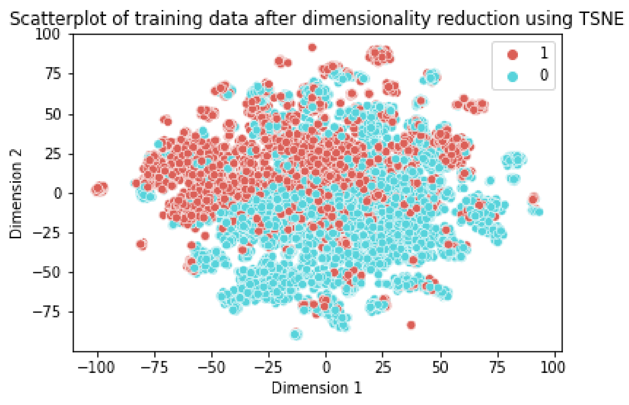


Figure 12. Scatterplot of the training data using t-SNE to check separability of data.

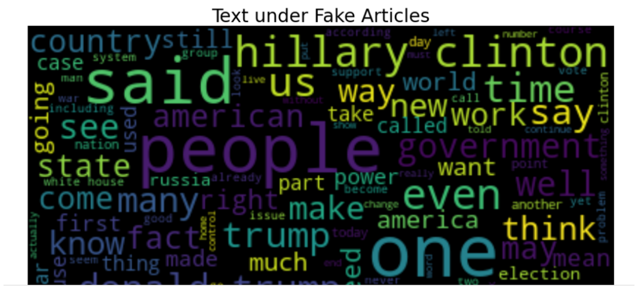


Figure 13. Word cloud for text of Fake articles

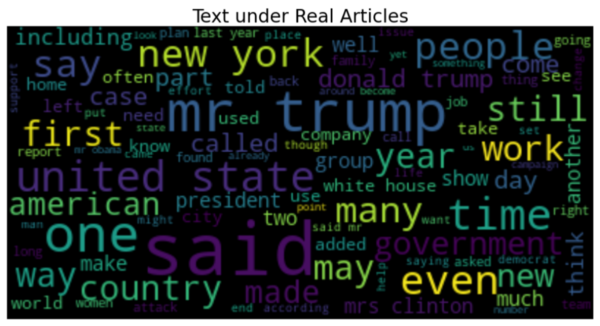


Figure 14. Word cloud for text of Real articles

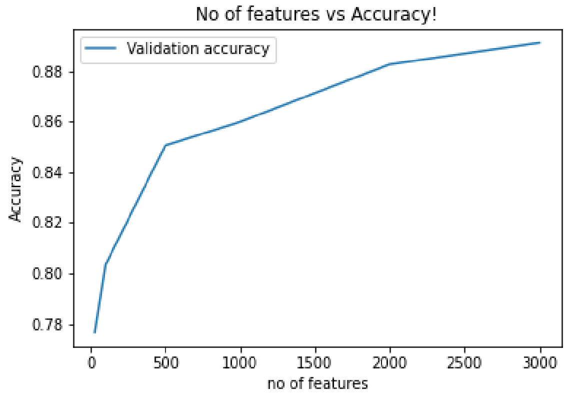
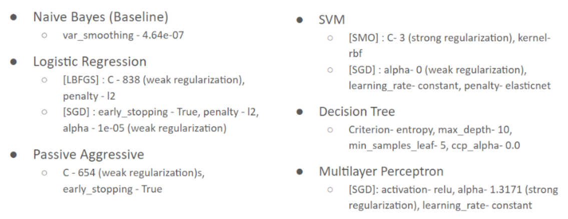


Figure 15. Number of features(in tf-idf vectorization of training data) Vs Mean Accuracy of Naive Bayes model(on validation set)



- Naive Bayes (Baseline)
  - var_smoothing - 4.64e-07
- Logistic Regression
  - [LBFGS] : C - 838 (weak regularization), penalty - l2
  - [SGD] : early_stopping - True, penalty - l2, alpha - 1e-05 (weak regularization)
- Passive Aggressive
  - C - 654 (weak regularization)s, early_stopping - True
- SVM
  - [SMO] : C- 3 (strong regularization), kernel- rbf
  - [SGD] : alpha- 0 (weak regularization), learning_rate- constant, penalty- elasticnet
- Decision Tree
  - Criterion- entropy, max_depth- 10, min_samples_leaf- 5, ccp_alpha- 0.0
- Multilayer Perceptron
  - [SGD]: activation- relu, alpha- 1.3171 (strong regularization), learning_rate- constant

Figure 16. Best Hyperparameters obtained using Bayesian Optimisation Hyperparameter tuning of models

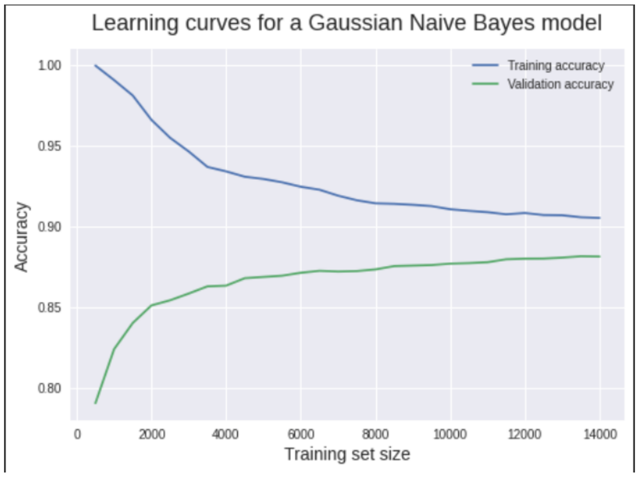


Figure 17. Learning Curve for Naive Bayes

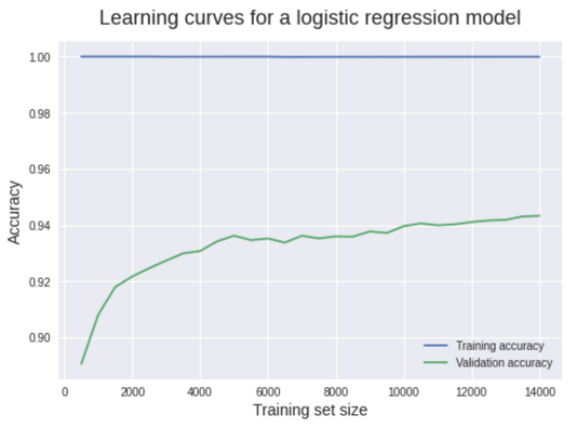


Figure 18. Learning Curve for Logistic Regression

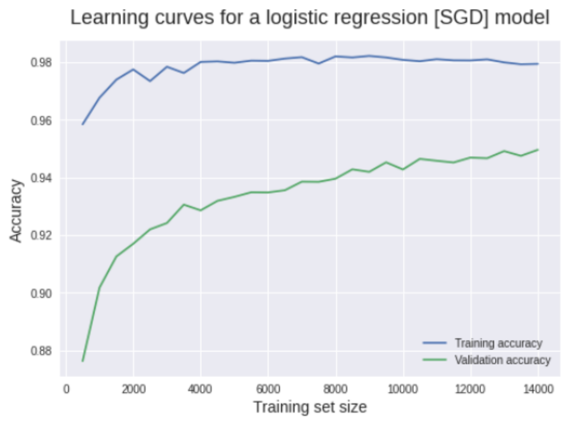


Figure 19. Learning Curve for Logistic regression [SGD]

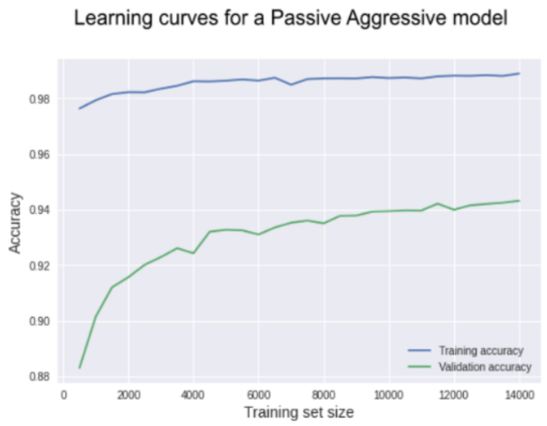


Figure 20. Learning Curve for the Passive Aggressive model

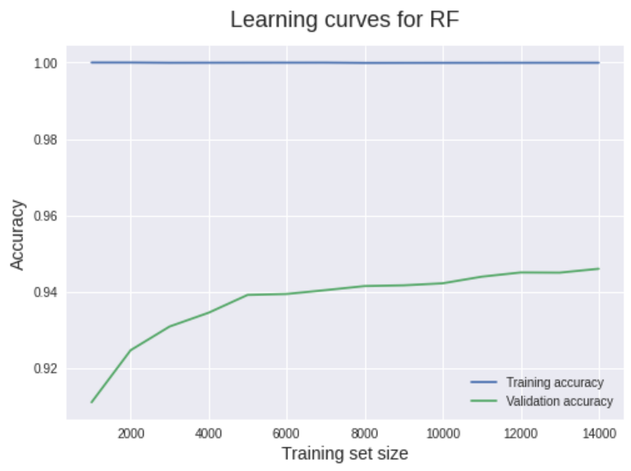


Figure 21. Learning Curve for the Random Forest model

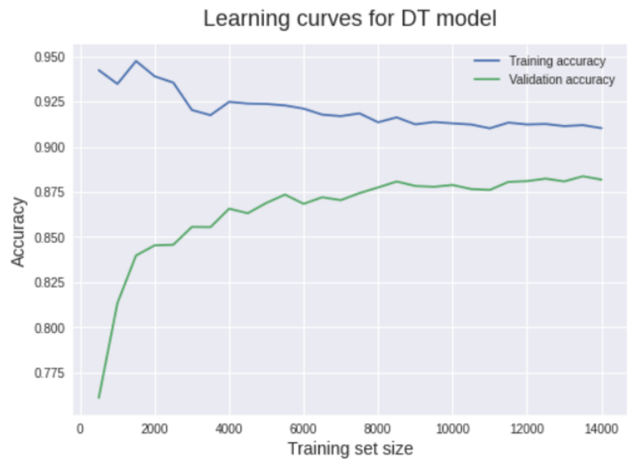


Figure 24. Learning Curve for the Decision Tree model

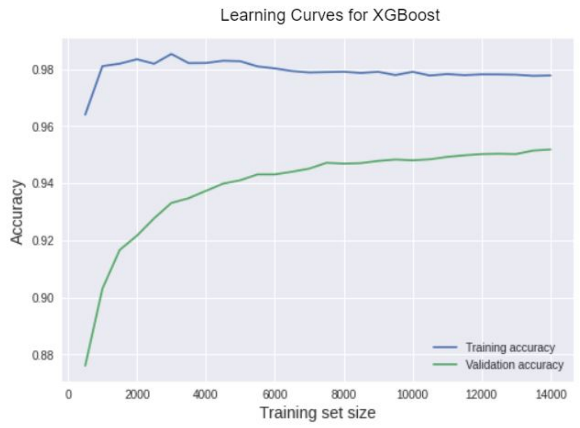


Figure 22. Learning Curve for the XGBoost model

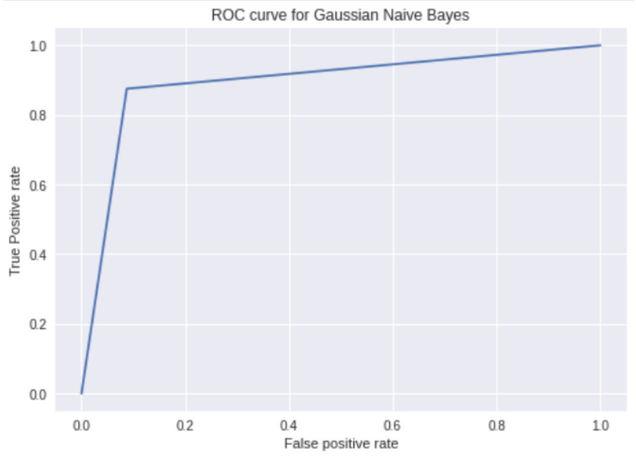


Figure 25. ROC curve for Gaussian Naive Bayes

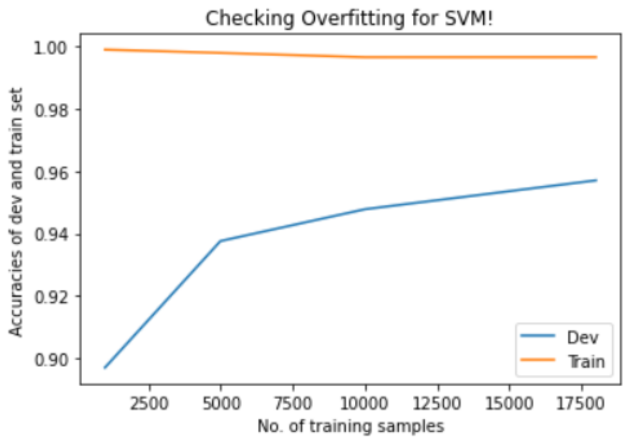


Figure 23. Learning Curve for the SVM model

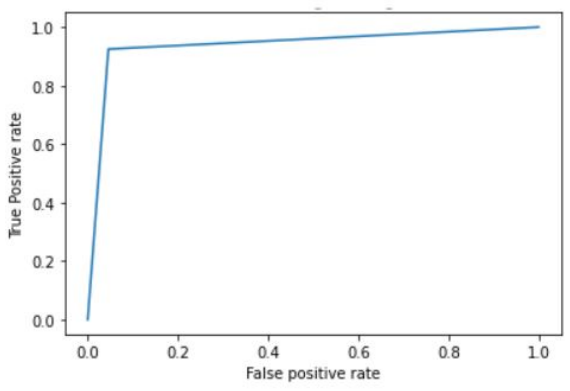


Figure 26. ROC curve for Random Forest

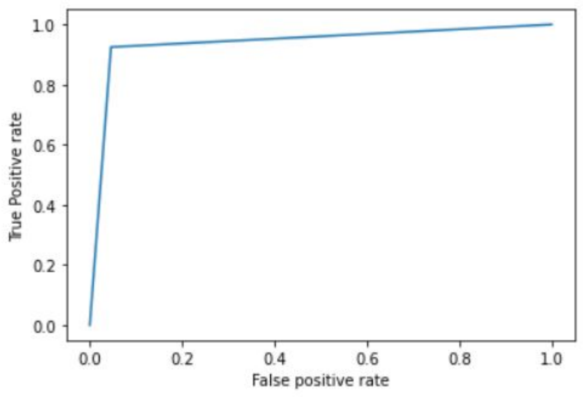


Figure 27. ROC curve for XGBoost

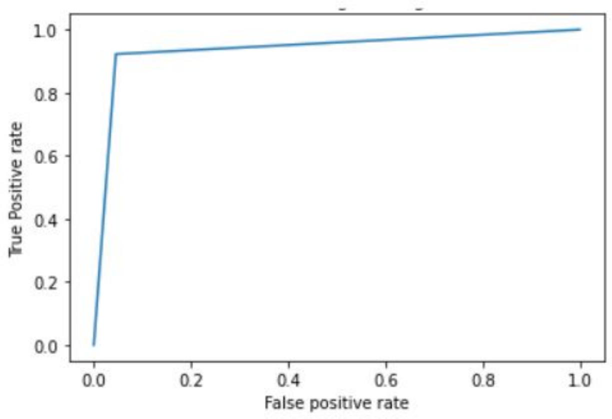


Figure 30. ROC curve for Logistic Regression [LBFGS]

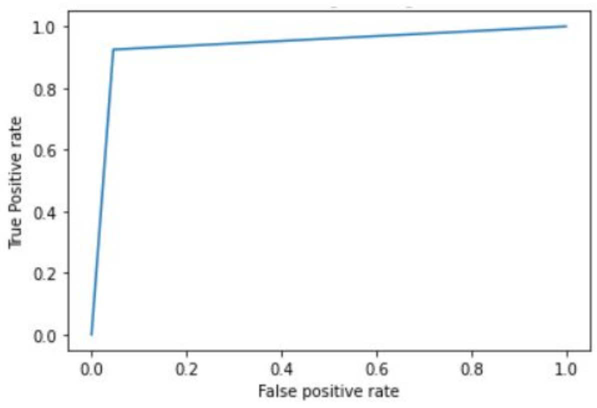


Figure 28. ROC curve for Passive Aggressive Classifier

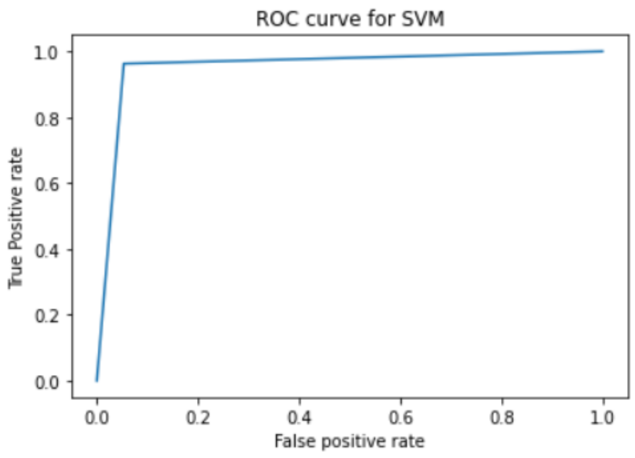


Figure 31. ROC curve for SVM

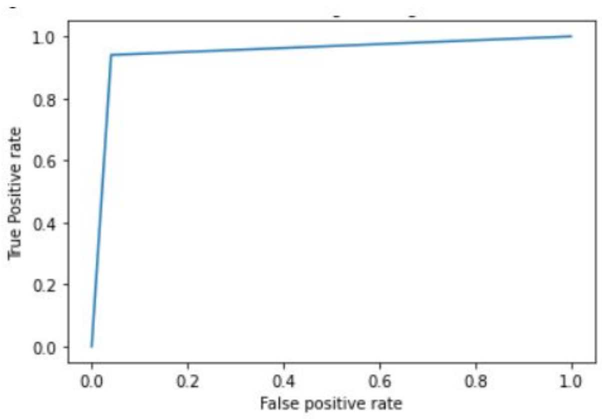


Figure 29. ROC curve for Logistic Regression [SGD]

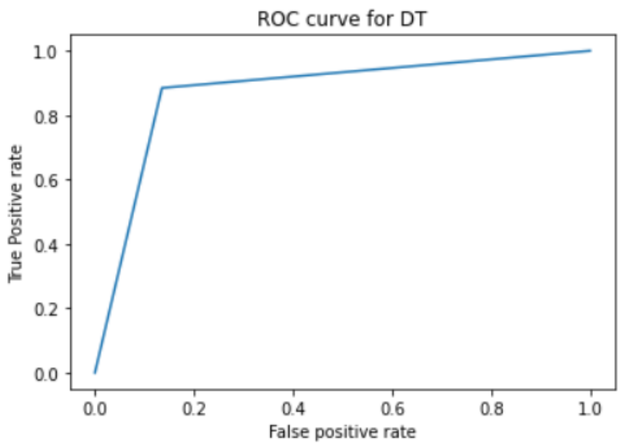


Figure 32. ROC curve for Decision Tree

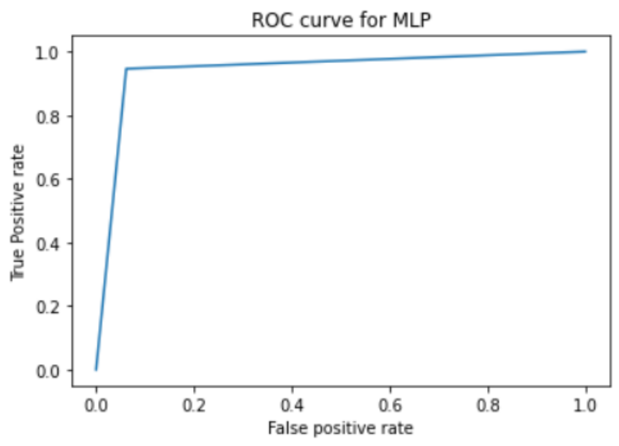


Figure 33. ROC curve for MLP

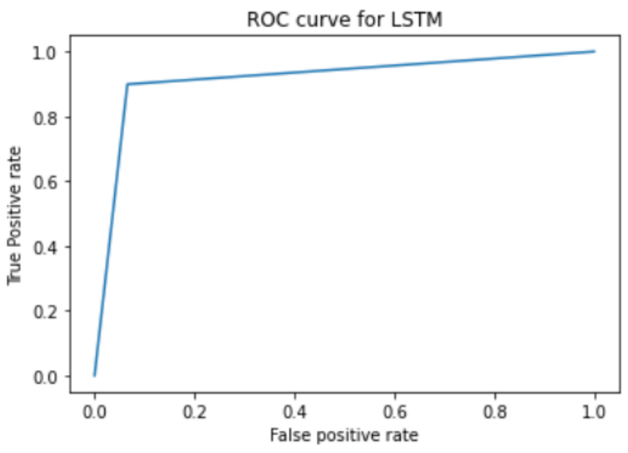


Figure 34. ROC curve for LSTM

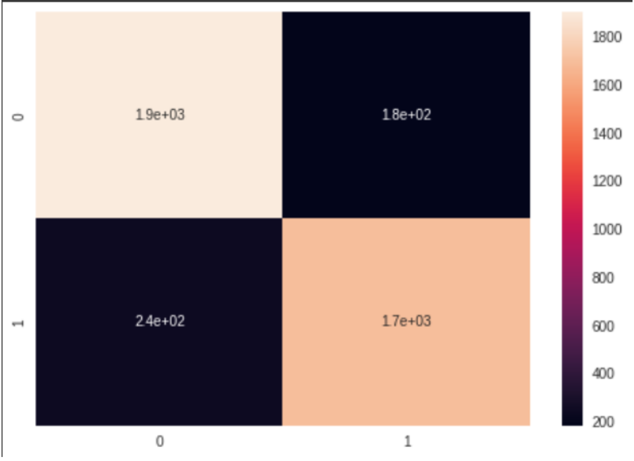


Figure 35. Confusion matrix for Gaussian Naive Bayes

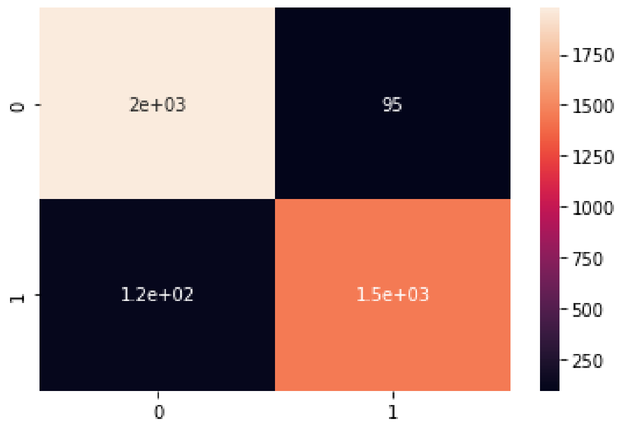


Figure 36. Confusion matrix for Random Forest

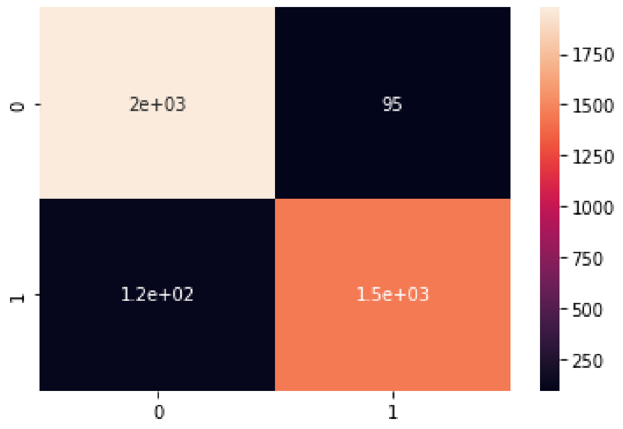


Figure 37. Confusion matrix for XGBoost

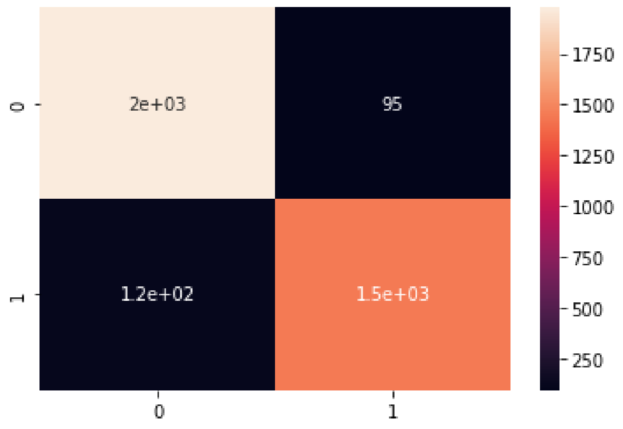


Figure 38. Confusion matrix for Passive Aggressive Classifier

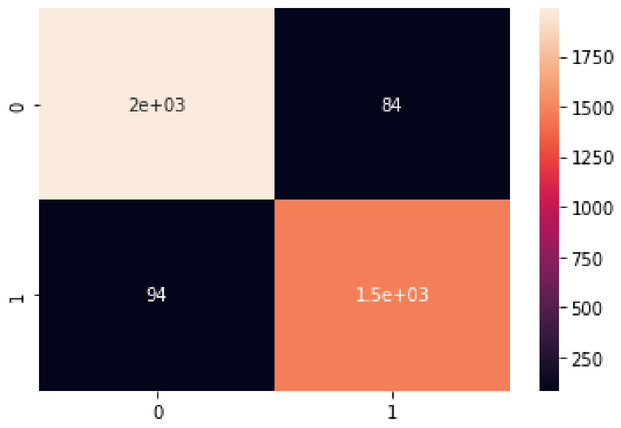


Figure 39. Confusion matrix for Logistic Regression [SGD]

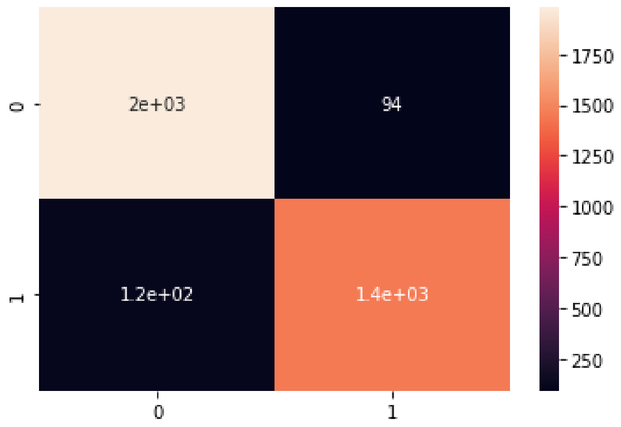


Figure 40. Confusion matrix for Logistic Regression [LBFGS]

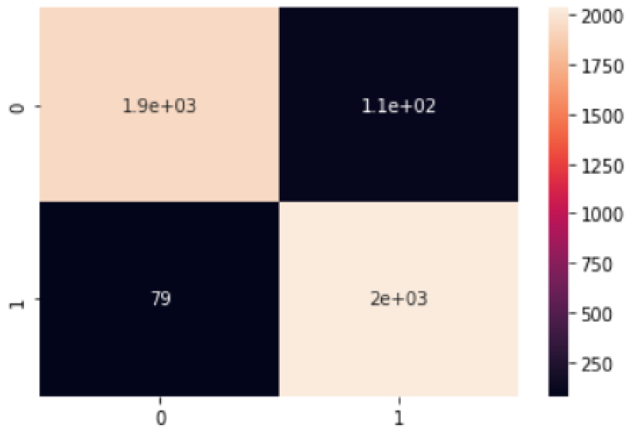


Figure 41. Confusion matrix for SVM

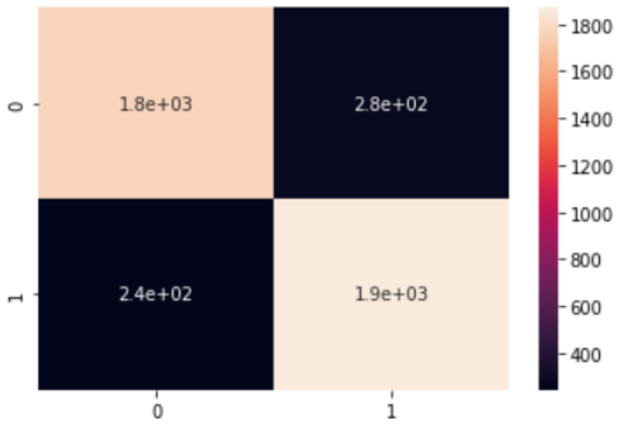


Figure 42. Confusion matrix for Decision Tree

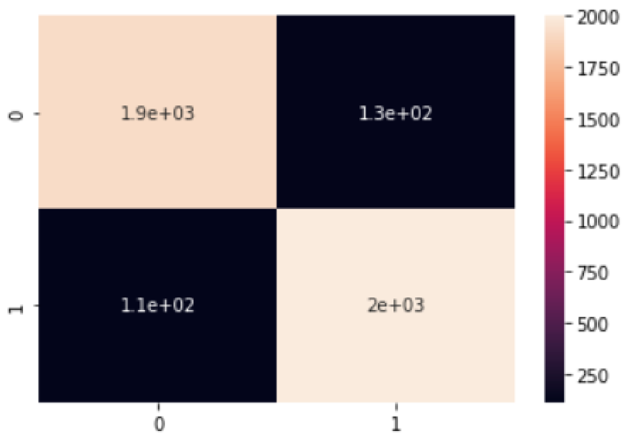


Figure 43. Confusion matrix for MLP

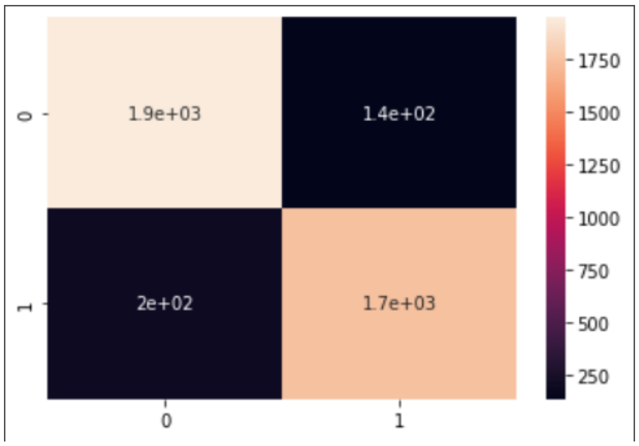
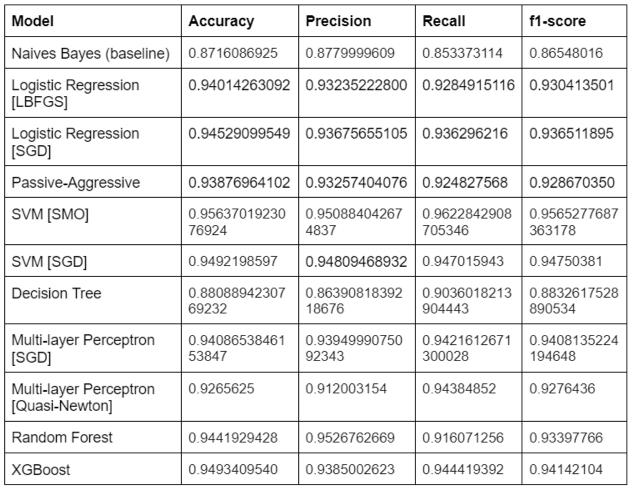
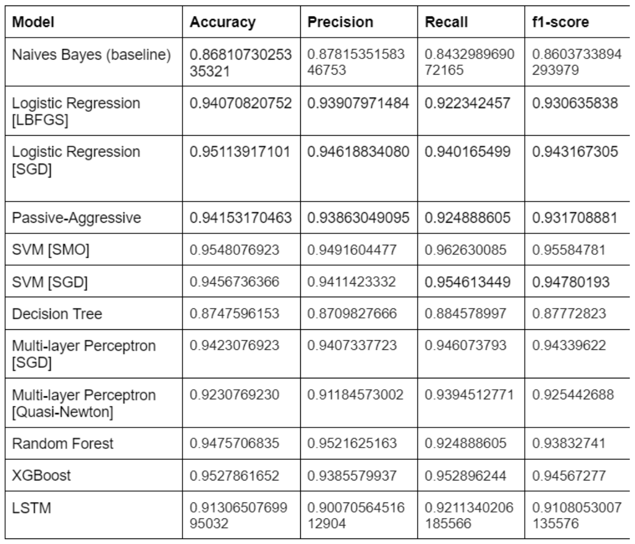


Figure 44. Confusion matrix for LSTM



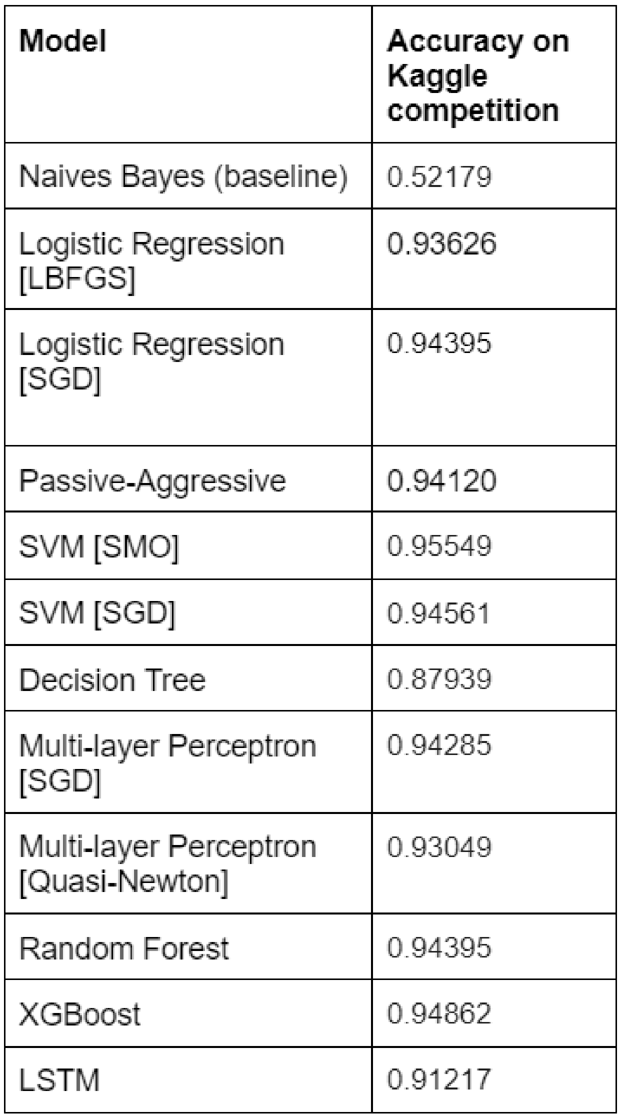
Model	Accuracy	Precision	Recall	f1-score
Naives Bayes (baseline)	0.8716086925	0.8779999609	0.853373114	0.86548016
Logistic Regression [LBFGS]	0.94014263092	0.93235222800	0.9284915116	0.930413501
Logistic Regression [SGD]	0.94529099549	0.93675655105	0.936296216	0.936511895
Passive-Aggressive	0.93876964102	0.93257404076	0.924827568	0.928670350
SVM [SMO]	0.9563701923076924	0.950884042674837	0.9622842908705346	0.9565277687363178
SVM [SGD]	0.9492198597	0.94809468932	0.947015943	0.94750381
Decision Tree	0.8808894230769232	0.8639081839218676	0.9036018213904443	0.8832617528890534
Multi-layer Perceptron [SGD]	0.9408653846153847	0.9394999075092343	0.9421612671300028	0.9408135224194648
Multi-layer Perceptron [Quasi-Newton]	0.9265625	0.912003154	0.94384852	0.9276436
Random Forest	0.9441929428	0.9526762669	0.916071256	0.93397766
XGBoost	0.9493409540	0.9385002623	0.944419392	0.94142104

Figure 45. Performance of our models on validation set (5-fold)



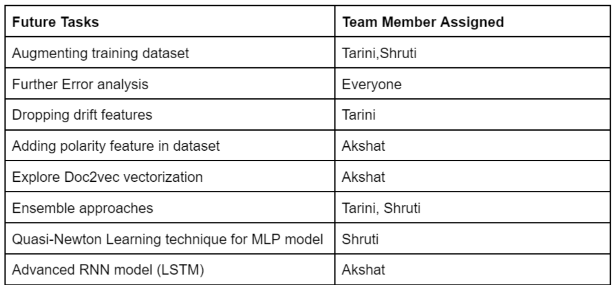
Model	Accuracy	Precision	Recall	f1-score
Naives Bayes (baseline)	0.8681073025335321	0.8781535158346753	0.843298969072165	0.8603733894293979
Logistic Regression [LBFGS]	0.94070820752	0.93907971484	0.922342457	0.930635838
Logistic Regression [SGD]	0.95113917101	0.94618834080	0.940165499	0.943167305
Passive-Aggressive	0.94153170463	0.93863049095	0.924888605	0.931708881
SVM [SMO]	0.9548076923	0.9491604477	0.962630085	0.95584781
SVM [SGD]	0.9456736366	0.9411423332	0.954613449	0.94780193
Decision Tree	0.8747596153	0.8709827666	0.884578997	0.87772823
Multi-layer Perceptron [SGD]	0.9423076923	0.9407337723	0.946073793	0.94339622
Multi-layer Perceptron [Quasi-Newton]	0.9230769230	0.91184573002	0.9394512771	0.925442688
Random Forest	0.9475706835	0.9521625163	0.924888605	0.93832741
XGBoost	0.9527861652	0.9385579937	0.952896244	0.94567277
LSTM	0.9130650769995032	0.9007056451612904	0.9211340206185566	0.9108053007135576

Figure 46. Performance of our models on the test set.



Model	Accuracy on Kaggle competition
Naives Bayes (baseline)	0.52179
Logistic Regression [LBFGS]	0.93626
Logistic Regression [SGD]	0.94395
Passive-Aggressive	0.94120
SVM [SMO]	0.95549
SVM [SGD]	0.94561
Decision Tree	0.87939
Multi-layer Perceptron [SGD]	0.94285
Multi-layer Perceptron [Quasi-Newton]	0.93049
Random Forest	0.94395
XGBoost	0.94862
LSTM	0.91217

Figure 47. Performance of our models on the test set provided by Kaggle.



Future Tasks	Team Member Assigned
Augmenting training dataset	Tarini,Shruti
Further Error analysis	Everyone
Dropping drift features	Tarini
Adding polarity feature in dataset	Akshat
Explore Doc2vec vectorization	Akshat
Ensemble approaches	Tarini, Shruti
Quasi-Newton Learning technique for MLP model	Shruti
Advanced RNN model (LSTM)	Akshat

Figure 48. Future Tasks distribution amongst the team

References

[1] Dataset Available Here (Click). 1, 3

[2] Pre-processing in NLP (Click). 1

[3] Referred LSTM's implementation from this link (Click). 2

[4] TextBlob and Sentiment Analysis referred from here (Click). 2

[5] A. Salma et al. B. Zapan, B. Sajib. Effects of misinformation on covid-19 individual responses and recommendations for resilience of disastrous consequences of misinformation, 2020. Volume 8, ISSN 2590-0617. 1

[6] Y. Benkler et al. D. M. J. Lazer, M. A. Baum. The science of fake news, 2018. Science, vol. 359, no. 6380, pp. 1094–1096. 1

[7] Zineb Giordano Silvia. Drif, Ahlem Ferhat Hamida. Fake news detection method based on text-features, 2019. 1, 4

[8] Anna Marciniak Martyna Tarczewska and Agata Gie lczyk. Fake or real? the novel approach to detecting online disinformation based on multi ml classifiers. 2020. 1, 4

[9] Viera Maslej Krešňáková, Martin Sarnovsky, and Peter Butka. Deep learning methods for fake news detection. 11 2019. 1, 4

[10] Ray Oshikawa, Jing Qian, and William Yang Wang. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*, 2018. 1