ML Project - 2022

Fake News Detection using Machine Learning Algorithms

Group 2:
Akshat Wadhwa (2019231)
Shruti Jha (2019274)
Tarini Sharma (2019451)

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Problem Statement and Dataset

- **Problem Statement:**
  - Social media platforms like Twitter get manipulated by certain entities to promote biased opinions/fake news (such as spread of misinformation about vaccines during the COVID-19 pandemic)
  - Goal:
    - use ML algorithms for automated classification of news articles as fake or real
    - 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
- **DataSet:**
  - We have used the dataset available on Kaggle; 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 news and 1 for unreliable news)

Individual Contribution

1. Akshat Wadhwa (2019231)
    Pre-processing
    Feature extraction
    Models: Baseline model(Naive Bayes)

2. Shruti Jha (2019274)
    Pre-processing
    EDA
    Models: SVM, Decision Tree, MLP

3. Tarini Sharma (2019451)
    Pre-processing
    EDA
    Models: Logistic Regression, Passive Aggressive

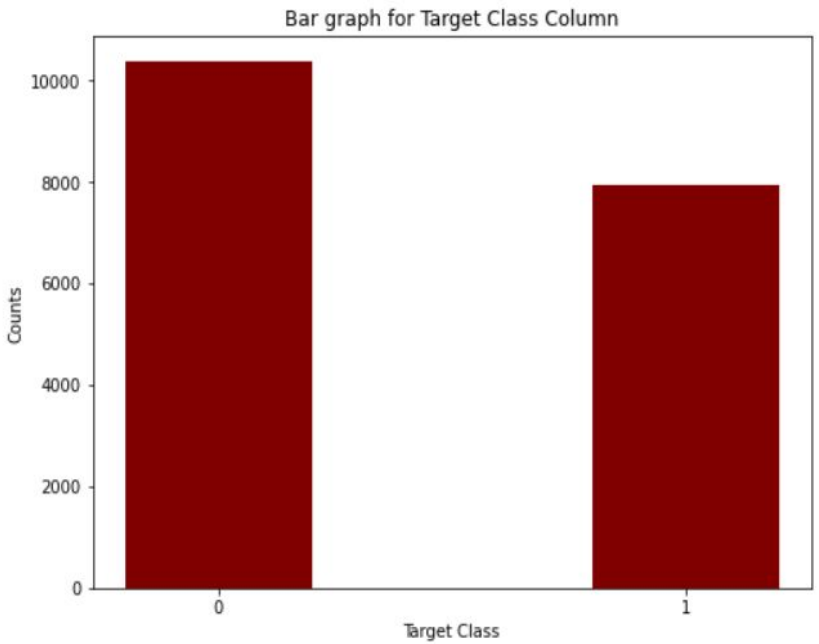
Pre-processing

1. Removed nulls + data samples containing only “ ” + “ “ + “\n”
2. No duplicate rows in dataset
3. Only kept A-Z and a-z English letters in news text field, replaced anything else with “ ”.
4. Drop stop words (eg. the, at, a)
5. Convert to root words [stemming]
6. Convert text data to numerical form using TFIDF Vectorizer

After preprocessing the dataset, training set size: 18211, test set size: 4544 samples

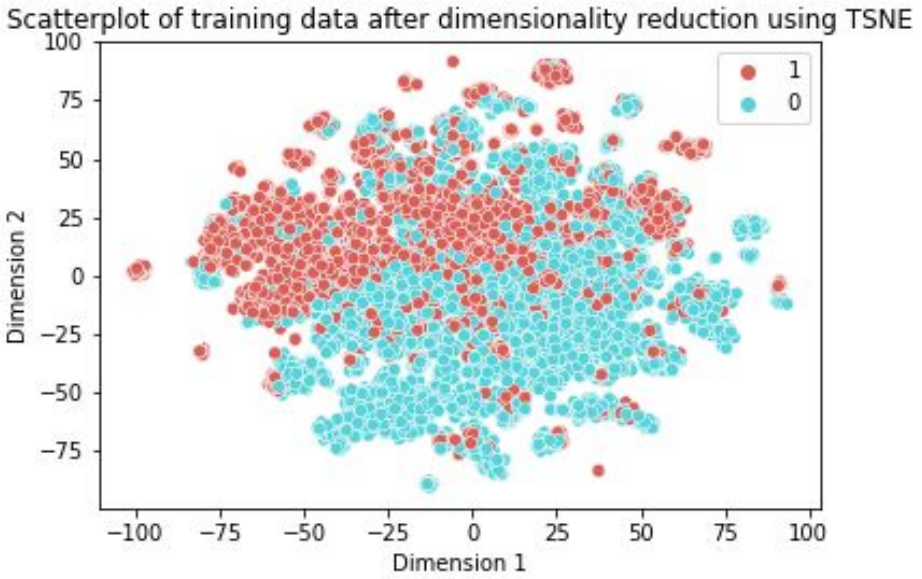
Exploratory Data Analysis (EDA)

Bar graph for target class



Class 1 is fake news class & Class 0 is real news class

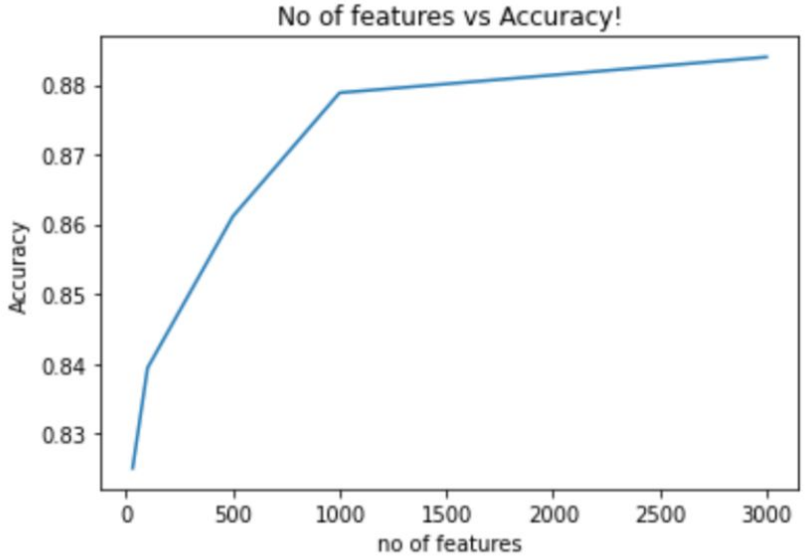
Separability of data:



The training data is not very well separated. There are quite a few number of samples of class 0 overlapping with samples of 1.

Feature Extraction

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



Note: Stratified kfold(k=5) used for train-validation set split.
Accuracy on y-axis is mean of accuracies on all 5 folds

Feature extraction using 3 methods; doc2vec, CountVectorizer and TF-IDF Vectorizer.

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

As TF-IDF gives the best results, we used TF-IDF for feature extraction for all the models.

Models: Optimal Hyperparameters

Hyperparameter tuning : feature reduction using PCA + applying Bayesian optimization

- 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

Evaluation Metrics

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

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

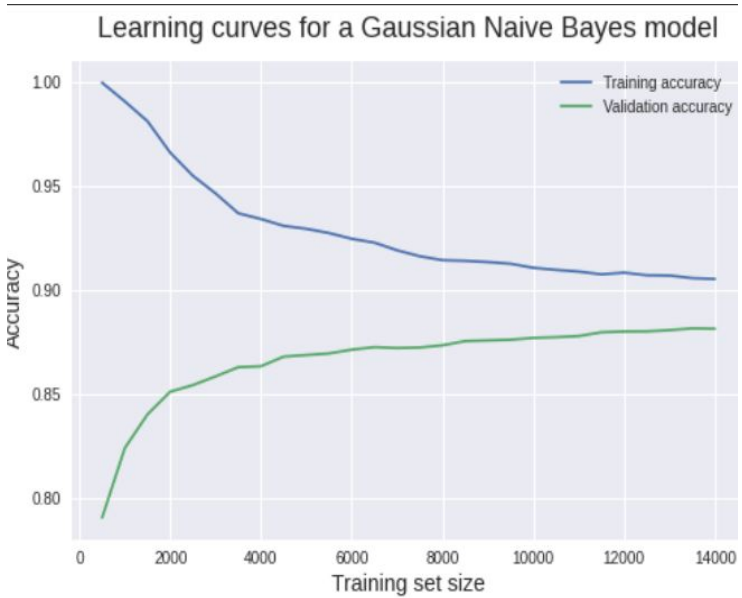
4) F1-score

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

Single-number evaluation metric : F1-score as harmonic mean of precision and recall. Satisficing metric: F1-score (threshold = 0.8), optimizing metric: accuracy

Results & Analysis - I

- Gaussian Naive Bayes
  - Validation set
    - Accuracy : 0.891268137324287
    - F1 score : 0.8853607640908908
  - Test set
    - Accuracy : 0.5801056338028169
    - F1 score : 0.5516917293233082



- SVM
  - Validation set
    - Accuracy : 0.959639611854439
    - F1 score : 0.9530845693078911
  - Test set
    - Accuracy : 0.6099911971830986
    - F1 score : 0.5864145129889237

Results & Analysis - II

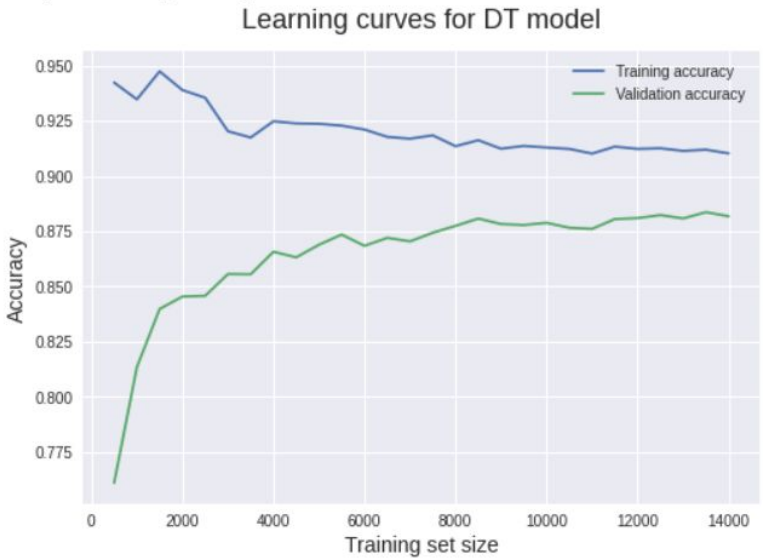
- Passive Aggressive
  - Validation set
    - Accuracy : 0.9409145867824718
    - F1 score : 0.93164202762820932
  - Test set
    - Accuracy : 0.6073943661971831
    - F1 score : 0.5825924192793636

- Logistic Regression
  - Validation set
    - Accuracy : 0.94239727352058058
    - F1 score : 0.933140345383244
  - Test set
    - Accuracy : 0.6034330985915493
    - F1 score : 0.5791686127977581

Results & Analysis - III

- Decision Tree
  - Validation set
    - Accuracy : 0.9056184059368971
    - F1 score : 0.88479364784546
  - Test set
    - Accuracy : 0.60656690140845
    - F1 score : 0.5759593037888536

- Multilayer Perceptron
  - Validation set
    - Accuracy :0.94629600402658
    - F1 score : 0.937633223842656
  - Test set
    - Accuracy : 0.60796654929577
    - F1 score : 0.5826341986110751



Future Work

1) Replace null values, instead of removing [eg, 1957/20387 nulls in author attribute in train set, replace null with -no author-]
2) Bagging(RandomForest) and Boosting(XGBoost) Classifiers mentioned in proposal left to be implemented, and its results analysed.
3) Reduce the high variance/overfitting models [increasing k, increase dataset size, regularization, early stopping]
4) Error analysis
5) One advanced model mentioned in proposal: LSTM left to be implemented, and its results analysed.
6) To try finding ‘polarity’ in text of fake and real news articles to be detected using sentiment analysis, and use it as a feature in the dataset to train models and compare results after using it.

Thank You!