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)
  - Since we submitted our predictions on kaggle competitions which only returns accuracy metric, we did 80-20 train-test split to test our models on other metrics as well (precision, recall, f1-score, roc-curve, confusion matrix).

Progress till Interim Submission

- **Preprocessing of dataset:** Removed null samples, no duplicate rows in dataset, Only kept A-Z and a-z English letters in news text field, dropped stop words, stemming, vectorise data to numerical form using TF-IDF vectorizer.
- **EDA:** Imbalance in class distribution(10361 samples of class 0{real} and 7850 samples of class 1{fake}, t-SNE scatterplot shows data not very well separable, word clouds made for train and test set show almost same weightage for words in both real and fake news set.
- **Feature Extraction:** Compare 3 methods to convert text data to vectorized format: doc2vec, CountVectorizer, TF-IDFVectorizer. Chose TF-IDFVectorizer with 3000 features extracted since baseline model (Naive Bayes) gives best validation set metrics (accuracy and f1-score) for this method.
- **Evaluation Metrics:** Single-number evaluation metric : F1-score as harmonic mean of precision and recall. Satisficing metric: F1-score (threshold = 0.88), optimizing metric: accuracy
- **ML Models explored (total 6):** Naive Bayes, SVM (learning methods: smo, sgd), MLP (learning method: sgd), Decision Trees, Logistic Regression (learning methods: lbfgs, dgd), Passive Aggressive classifier.
- **Hyperparameter tuning:** Found optimal parameters using Bayesian Optimization.
- **Results:** Calculated accuracy and f1-score on validation set (stratified k=5-fold) for the 6 models explored. Best accuracy for SVM (95.96%), and best f1-score for SVM (95.3%).

Progress after Interim Submission - Approaches (I)

After the intermediate submission, we worked on the following additional models.

- XGBoost

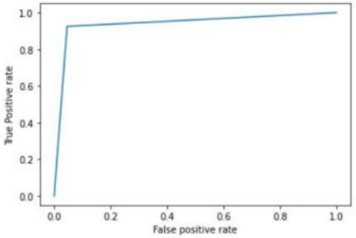


Figure 27. ROC curve for XGBoost

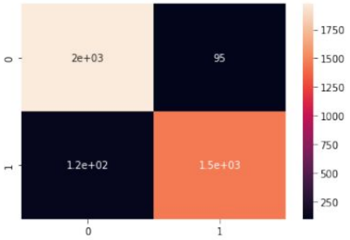


Figure 37. Confusion matrix for XGBoost

- Random Forests

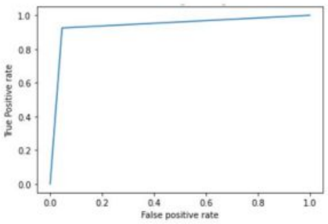


Figure 26. ROC curve for Random Forest

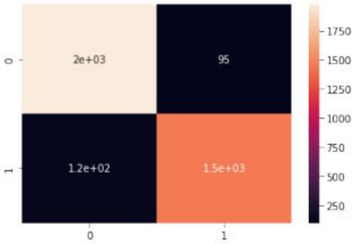
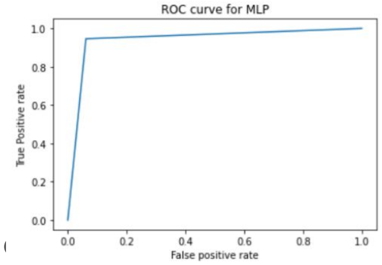


Figure 36. Confusion matrix for Random Forest

- MLP (Quasi-Newton learning)



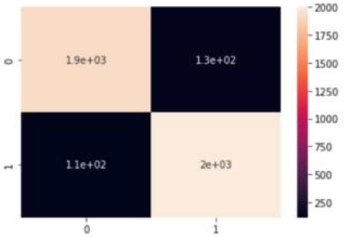
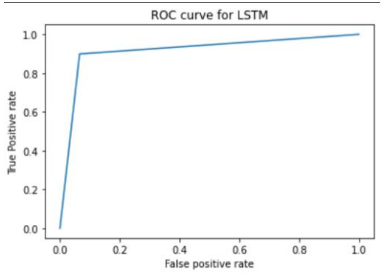
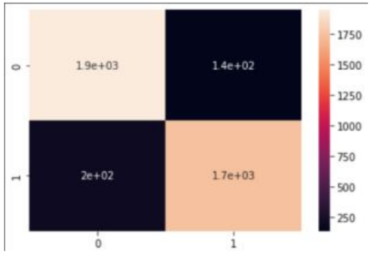


Figure 43. Confusion matrix for MLP

- LSTM

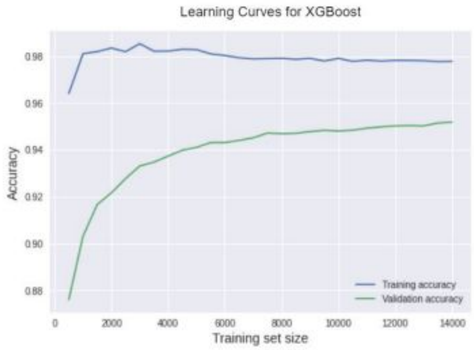




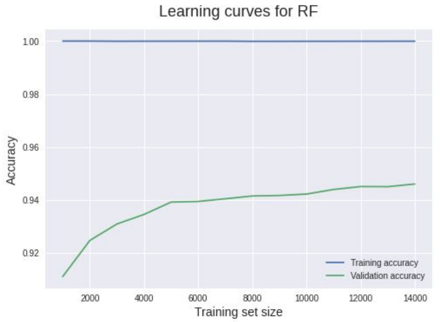
Progress after Interim Submission - Approaches (II)

We performed hyperparameter tuning on two of the new models.

- XGBoost
  - Hyperparameters best suited:
    - L1 Regularisation (α) and L2 Regularisation (λ) used.
    - Learning rate(η) was lower than default (0.3)
    - Maxdepth of tree - 10
    - Number of estimators - 100
  - Learning Curve :



- Random Forests
  - Hyperparameters best suited:
    - Number of estimators-174
    - ccp_alpha=0.0 (Pruning was not performed)
  - Learning Curve :



Progress after Interim Submission - Results

We tabulated the values of the evaluation metrics on our own split train-val-test sets from the training set provided. Also, we tabulated the accuracies from the Kaggle competitions test set.

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

Table. Metrics on the validation set

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

Table. Metrics 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

Table. Accuracy on Kaggle test set

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 95.48%.

Analysis and Ablation (I)

- TextBlob(maxpos, minneg, mean_sentiment, mean_subjectivity) vs Vader(pos, neg, neu, compound)
  - Trained on Logistic Regression
  - Textblob : Accuracy: 0.63107, Precision: 0.601618, Recall: 0.426114, F1-score: 0.498881431
  - Vader : Accuracy: 0.6184, Precision: 0.6898, Recall: 0.20828, F1-score: 0.31996
- 3000 features extracted from TF-IDF + 4 features from TextBlob -> trained on Logistic Regression(LBFGS, SGD) and Passive Aggressive Classifier

Val set metrics:

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

Test set metrics:

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

Analysis and Ablation (II)

- Gaussian naive bayes- underfit (high bias, low variance), advanced models- fit to the data well (low bias, low variance)
- Accuracy comparison
  - Naive bayes: 28.8% from [6]
  - SVM: 35% and 32% from [5], [6]
  - Random Forest, XGBoost: 16% and 6% from [6]
  - LSTM:-0.5% from [8]
- Sentiment analysis
  - Removing this component did not lead to any difference on the dev set metrics
- Dataset needs to be diverse
  - Mislabelled dev samples -> contains keywords (essential for a human classifying the text as fake or real) not present in the train set
  - “pokemon go players are inadvertently stopping people committing suicide in japan“

Individual Contributions

Team Member	Tasks/Deliverables Completed
Akshat Wadhwa	Pre-processing, Feature extraction, ML models explored (Gaussian Naive Bayes and LSTM), Hyperparameter tuning, Evaluation metrics, Drawing final conclusions and report writing.
Shruti Jha	Preprocessing, Exploratory Data Analysis, ML models explored with different learning methods (SVM [SGD, SMO], Decision Trees, MLP [SGD, Quasi-Newton]), Model selection, Drawing final conclusions and report writing.
Tarini Sharma	EDA, ML models explored with different learning methods (Logistic regression [LBFGS, SGD], Passive Aggressive Classifier, Ensemble methods {XGBoost, Random Forest}), Error analysis, Covariate shift check, Sentiment analysis, Drawing final conclusions and report writing.

Thank You!