ML - Assignment : Fake news detection

Name : Ashwin Vaswani
Institute: Birla Institute of Technology and Science (BITS Pilani) , Goa Campus
Email : ashwin.vaswani99@gmail.com / f20170960@goa.bits-pilani.ac.in
Contact : +91 7977572467

Problem Statement :

LIAR-PLUS is a benchmark dataset for fake news detection, released recently. This dataset has evidence sentences extracted automatically from the full-text verdict report written by journalists in Politifact. It consists of 12,836 short statements taken from POLITIFACT and labeled by humans for truthfulness, subject, context/venue, speaker, state, party, and prior history. For truthfulness, the LIAR dataset has six labels: pants-fire, false, mostly-false, half-true, mostly-true, and true. These six label sets are relatively balanced in size.

There are two tasks:
1. Binary classification task (true, false)
2. Six-way classification task (pants on fire, false, mostly false, half-true, mostly true, true)

Your task is to select a classifier (or a group of classifiers) that you think is suitable for this dataset and get the highest possible accuracy. You can choose different classifiers for the two tasks. You are free to use any ideas you think will be suitable for this dataset. For example, you may plot graphs for gaining insights about the features or clean/re-organise the data for feature extraction.

Overview:

Out of all the models I implemented , I got a maximum accuracy of 74% for Binary classification and 44% for Multi Class classification using Gradient Boosting which is better than the results given in the original paper.

How to run :

I have provided a jupyter notebook in the zip file attached which contains all the necessary code.

Methodology:

I first started off with Logistic Regression on all the features and got a base accuracy of 30% on Multi class classification task. I vectorized and converted to one hot encoding and used the same for logistic regression. I then proceeded by trying to select different features instead of all and as mentioned in the original paper, i tried all combinations such as S , SJ, S+MJ etc.
After improving on the accuracy, I proceeded to ensembling techniques and used Gradient Boosting which increased my accuracy to around 44%. After getting good accuracy with machine learning models, I proceeded to a Deep Learning approach. I implemented a Bi-LSTM which gave me decent accuracies similar to those mentioned in the original paper but not as good as the ensembling techniques mentioned above. I did not focus on optimization because of time and resource constraints and thus did not attach those in the Jupyter notebook attached. I can provide code for the same if required.

Here is a table with all the new results apart from those in the original paper :

Condition	Model	Binary		Six-way	
		Valid	Test	Valid	Test
All features	Logistic Regression	57.7%	56.7%	33.4%	30%
All features	Gradient Boosting	70%	74.4%	45%	43%
All features except Context and Subject	Logistic Regression	71.1%	74.1%	34.5%	31.5%
All features except Context and Subject	Gradient Boosting	71.2%	74.1%	46.5%	42.3%
Only Statement,context,justification and subject	Logistic Regression	57.7%	56.7%	25.3%	22.65%
Only Statement,context,justification and subject	Gradient Boosting	56.5%	59%	25%	25%
Only Statement and Justification	Logistic Regression	57.2%	57%	22.5%	21.5%
Only Statement and Justification	Gradient Boosting	71.2%	74.2%	46%	42.5%
All features	Basic Bi-LSTM	64.3%	63%	32.4%	32.1%

Conclusion:

From the table, it can be seen that the models when trained on All features performed better than the original paper. This can be explained logically from the idea that the credibility of the person speaking a fact or news is also important. Even better accuracies can be achieved given more time and resources and since this was a timed assignment, i could not optimise on the models. However the results from this work are sufficient to see that taking credibility of person as a feature improves our results to give SOTA accuracies and can be further improved with other ensembling or Deep Learning methods.

Libraries / References used:

I used sklearn's logistic regression initially and then moved onto ensemble techniques like Gradient boosting from sklearn.ensemble. I used numpy and pandas for basic data handling operations. I also used nltk's stopwords to clean the data.