Learning to predict the stance of news articles

Justin Ty z5232245

Introduction

The prevelance of fake news in the media has been an increasing issue in recent times. A Fake News Challenge (FNC) was created in 2017 in an attempt to address this issue by using automated systems. The goal of the challenge was to build an AI system that could automatically detect the stance of news articles. This project for COMP9417 was an attempt to build an end-to-end process which could take news headlines and articles to predict their stance.

Implementation

Dataset and Labels

The provided dataset from the challenge contained news article bodies and headlines. The goal of the challenge is to build a classifier that can detect the stance between the body and headline. The stance of the dataset can be one of the following:

- Agree - If the headline and article body agree
- Disagree - If the headline and article body disagree
- Discusses - If there was a discussion of the topic
- Unrelated - If the headline and body were not related.

The nature of the problem could have been modelled in stages but Chaudhry (2018) noted that this does not improve results and only adds complexity. As such, the models in this project treated this as a 4-class classification problem.

Feature Engineering

The main approach for the features was to use a bag of words representation of words in the article. A TFIDF Vectorizer was used to generate this bag-of-words. This tranformation made more frequent words and stop words less important, while giving less occuring words more weight.

A large vocabulary with a one-hot-encoding resulted in a very sparse dataset. In order to improve performance and reduce noise, dimensionality reduction was applied to the bag of words using Latent Semantic Analysis.

Additionally, some feature engineering scripts were borrowed fromt the original challenge. Cooccurence of words between were the main features created by the original utility functions from the challenge. The borrowed features from the original repository include the count of refutes and overlaps between the article body and headline.

Modelling

The baseline from the competition used a Gradient Boosting Classifier with 200 estimators.

Another simple benchmark was a Simple Decision Tree. This model was prone to overfitting as its training scores were very high yet its validation scores are low. The learning curves in the appendix confirm this.

To avoid the overfitting of a single decision tree learner, a randomised cross validation search was used to tune its hyper parameters. This method allowed a single tree to be optimised and tuned by k fold cross validation. The hyper paramters include the tree's max depth and minimum items for a split.

Neural networks and gradient boost proved to be popular and effective in the competition. SOLAT in the SWEN (Sean et al. 2017) by Talos Intelligence was the winner of the FNC which combined both Gradient Boosting and a Convolutional Neural Network with Google's pretained Word2Vec. The second, Athene (Hanselowski 2017) also used an ensemble of Multilayer Linear Perceptrons. In this project, a simple dense neural network was implemented without ensembling.

Cross validation

Validation and learning plots were generated to aid with decisions on the bias variance tradeoff. These can be seen in the appendix.

Scoring

Some scoring utility functions were borrowed from the challenge repository in order to compare this project's models against the baselines. This is referred to as the FNC scores.

Hanselowski (2018) noted that the original scoring functions were not appropriate for validating the document-level stance detection task. Instead, it was their recommendation to use F1 score with macro averaging. This is so that the scores are not affected by majority classes.

Findings by Hanselowski (2018) mentioned that some classifiers do not create predictions for the Disagree class while yielding high scores. The majority classes were Discuss and Unrelated which the FNC score tend to favour.

For this project, the FNC, F1, and accuracy scores were recorded. The detailed logs of the application were also saved to in the archive that shows their confusion matrices.

Experimentation

Boosted ensemble methods seem to work very well especially for an unbalanced dataset such as this. Additional ensemble methods were also used such as Adaboost, Random Forest were used. These methods show comparable results against the boosted baseline.

In addition, XGBoost was also trained and tested. It has the benefit of being fast and scalable with the focus on speed and performance. It also contains regularisation parameters which help against overfitting. If the data is structured properly, it can be very efficient for sparse data which is suitable for a bag-of-words approach.

A simple neural network was also implemented in this project using a dense network in Keras. It was found that the scores for the neural network in this project does not score as well as the other ensemble models. This may be due to the lack of quality features. This could be futher improved with more background in NLP feature engineering.

Hanselowski (2018) noted, these some models tend to score well for the FNC score but do not create as much predictions for the Disgree class, yielding lower F1 scores. These observations were also made on the neural network trained on this project.

A summary of FNC scores, F1 Scores, and Accuracy is listed below. For comparison, Hanselowski (2018) recorded the scores of the challenge participants are as follows:

Scores of Challenge Participants

System	FNC	F1 Macro
Talos Comb	0.820	0.582
Talso Tree	0.830	0.570
Talos CNN	0.502	0.308
Athene	0.820	0.604

Scores from this project

System	FNC	F1 Macro	Accuracy	Training Time (seconds)
Simple Decision Tree	49.55%	29.83%		4.388067007
Gradient Boosting Model	TODO			597.99
XGBoost				33.44
Decision Tree with RandomCV	57.82%	36.65%	71.76%	

Search				
Dense Neural Network				

The scores of this project do not exceed that of the winners of the competition. This can be attributed to the competition winners use pretrained vectors and better feature engineering from an NLP stand point.

In terms of performance and accuracy, gradient boosting modles

Conclusion and further recommendation

This project has shown an end-to-end process of feature engineering and predicting the stance of news articles without the use of pretrained materials. The process included transforming raw text data into vectorised features for a model to train from.

This project was meant for education purposes without using pretained materials. In terms of FNC and F1 scores, the models from this project do not score as high as the winners of the original challenge.

On the other hand, the test accuracy of the models from this project were good.

Ensemble estimators from this project have drawn out the best results. GBM and XGBoost were able to predict the correct stance of 80% of the test data.

In particular, XGBoost was performant and good preventing overfitting with the use of an early stop. XGBoost was a significantly more performant and slightly more accurate estimator than the baseline GBM.

Futher improvements could be done with this project. Areas of improvement can be done with better featuring engineering in the NLP space.

References

Andreas Hanselowski , et al (2018) *A Retrospective Analysis of the Fake News Challenge Stance Detection Task*, : Research Training Group AIPHES Computer Science Department, Technische Universität Darmstadt. https://arxiv.org/pdf/1806.05180.pdf

Melanie Tosik, et al (2018) *Debunking Fake News One Feature at a Time*, : Department of Computer Science New York University. https://arxiv.org/pdf/1808.02831.pdf

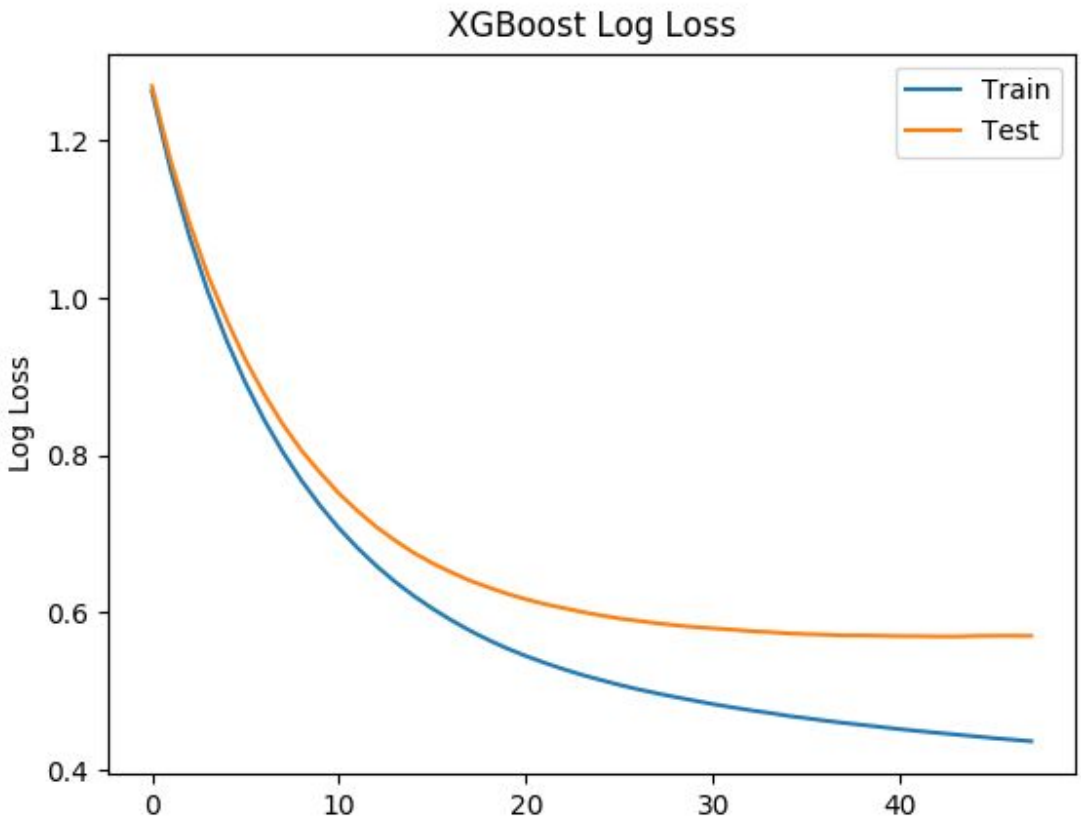
Ali K. Chaudhry, Darren Baker, Philipp Thun-Hohenstein (2018) *Stance Detection for the Fake News Challenge: Identifying Textual Relationships with Deep Neural Nets*, : Stanford. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2760230.pdf

Sean , Baird, Doug Sibley, Yuxi Pan (2018) *Talos Targets Disinformation with Fake News Challenge Victory,* Talos Intelligence https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html

Fake New Challenge http://www.fakenewschallenge.org/

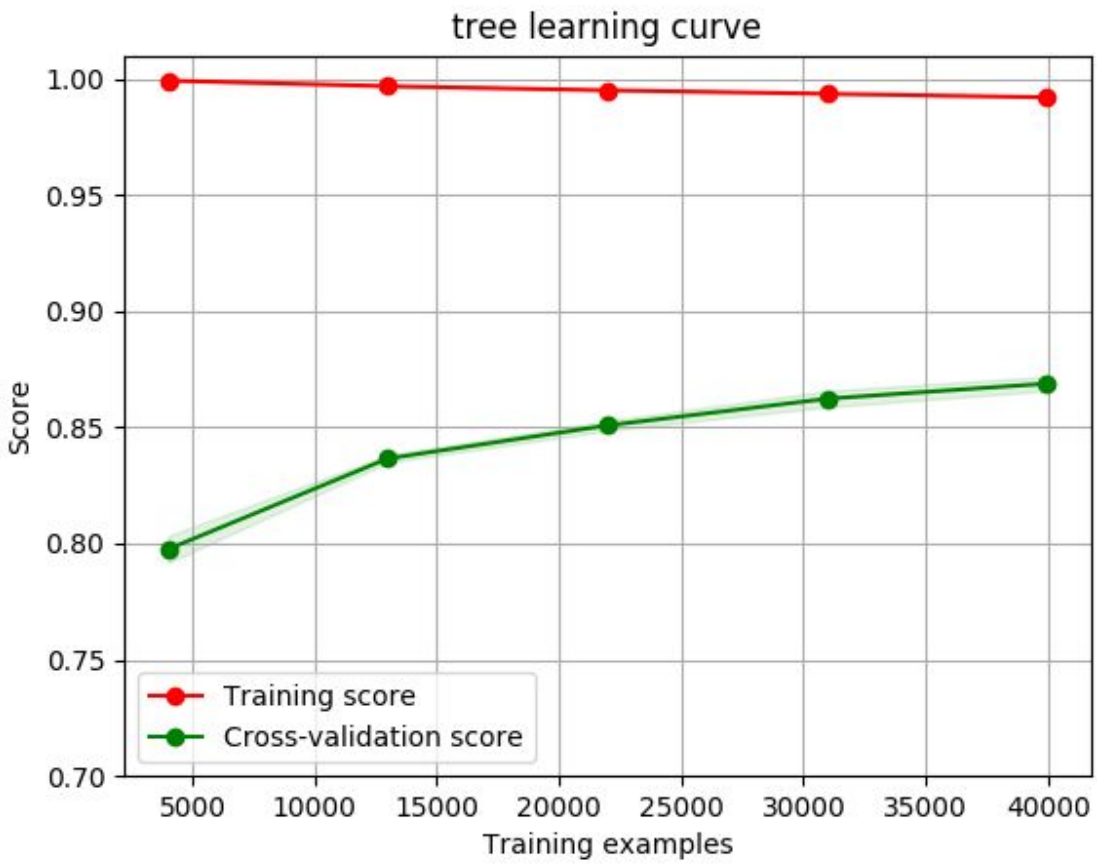
Tianqi Chen and Carlos Guestrin. (2016). *XGBoost: A Scalable Tree Boosting System*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, 785-794. DOI: https://doi.org/10.1145/2939672.2939785

Appendix



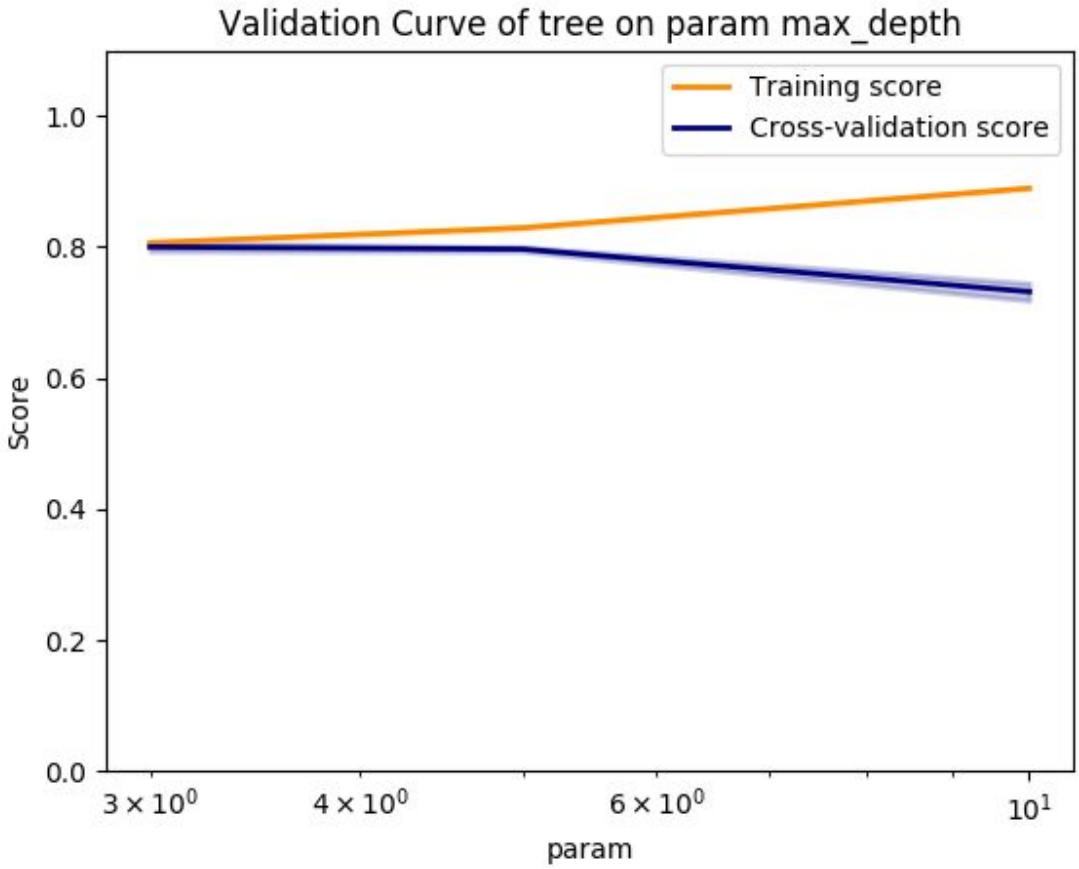
An early stop[1] was applied to XGBoost. This prevents anymore learning to avoid overfitting. If the log loss (cross entropy) does not decrease after 5 rounds, then the training stops.

[1] https://machinelearningmastery.com/avoid-overfitting-by-early-stopping-with-xgboost-in-python/



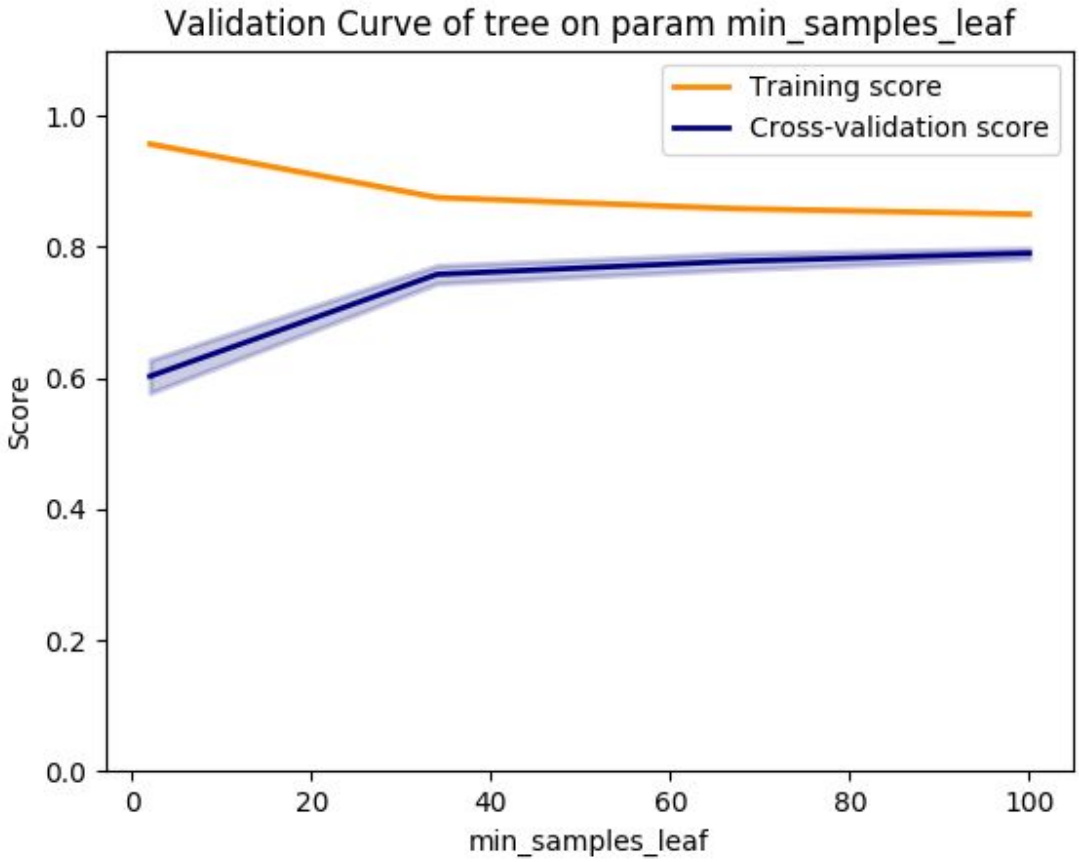
The learning curve[2] that the Simple Decision Tree overfits the training set. It scores almost 100% on the training set while it's cross validation score remained low.

[2] https://scikit-learn.org/stable/auto_examples/model_selection/plot_validation_curve.html



The validation plot[3] suggests that the max depth for a tree is around 5. Any deeper, the model will overfit the training set but degrade on cross validation.

3 https://scikit-learn.org/stable/auto_examples/model_selection/plot_validation_curve.html#sphx-glr-auto-examples-model-selection-plot-validation-curve-py



This suggests that the minimum number of samples in the leaf should be around 50.