CAPSTONE PROJECT
Machine Learning Engineer Nanodegree

June 15, 2021

Domain Background
Fake news became a very critical problem in today's political and social life. Disinformation continues to feed intolerance and polarize our societies.
In this project, I plan to build a classification model able to separate the fake news from the real ones. The project will use the XGBoost classifier to accomplish its objective. Although Sequetial (keras) models were shown to be highly effective for these data,[1] my projects aims to employ XGBoost classifier to see if it can do beter.
This project is inspired by Data Flair example of detecting Fake/Real News.

Problem Statement
The objective is to build a classifier model distinguishing between fake and real news. The particular tasks that will be achieved in the project are the following:

1. Download and clean the fake/real news data from Kaggle: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset
2. Implement two vectorizers: TF-IDF and Count Vectorizer and then test which of these result in better model performance.
3. Train a XGBoost classifier (choose the best performing vectorizer)
4. Tune hyperparameters
5. Test model performance on the test part of the data.

Datasets and Inputs
The project will use the data from Kaggle on fake and real news. The data can be downloaded from here: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset. This data consists of two parts, one is called 'real', and the other one—'fake.' The data contains title of the article, text of the article, subject, and date of the article. This is an appropriate dataset for the use for classification of fake vs real news.

Solution Statement
The project will use either tf-idf or count vectorizer (depending on their performance) with the XGBoost classifier. The main evaluation metric will be accuracy, but precision and recall will be reported as well. These metrics are common in evaluating the performance of classifier models. Seeds will be used to make the results replicable.

Benchmark Model
The model with tf-idf and XGBoost will serve as a benchmark for further hyperparameter tuning. The initial parameters will be:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
```

[1] https://medium.com/@ODSC/deep-learning-finds-fake-news-with-97-accuracy-d774ca977b0d

learning_rate=0.3, max_delta_step=0, max_depth=6,
min_child_weight=1, missing=nan, monotone_constraints='()',
n_estimators=100, n_jobs=16, num_parallel_tree=1, random_state=42,
reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=42,
subsample=1, tree_method='exact', validate_parameters=1,
verbosity=None)

Evaluation Metrics

GridSearchCV will be used for hyperparameter tuning. In tuning, Area Under the Receiver Operating Characteristic Curve (ROC AUC) will be used for evaluation. For the model itself, the main evaluation metric will be accuracy, but precision and recall will be reported as well. These metrics are common in evaluating the performance of classifier models.

Project Design

The text analysis problems require the transformation of the text in a way that can letter be fed to models. Among possible vectorization possibilities, the project chose two common ones, in particular TF-IDF and Count vectorization for data preprocessing. Because this is a classification problem, XGBoostClassifier can be used as one of the possible models for classification.