Machine Learning Engineering Nanodegree

Capstone Proposal

Amrita Sharma
May 27th, 2020

Domain Background

This project is chosen from Kaggle dataset to classify between real and fake news. The challenge here is to create an algorithm able to determine if a news article is fake news or not. Given the current scenario of COVID-19 pandemic, recession, and the growing unemployment, it has become crucial to identify between fake and real news, which was the inspiration to choose this domain. Since we are constantly exposed to numerous news sources daily, an algorithm such as this would help identify real and fake news.

Problem Statement

The problem is to create an algorithm to classify if a news article is a fake news or real news.

Datasets and Inputs

Kaggle dataset contains two files Fake.csv and True.csv. Both the .csv files contain columns such as title, text, subject, and date. Primarily text column will be used for feature engineering in the Machine Learning Model. Other columns such as news titles, news subjects, and news dates can be used as features too; however, this project is focused on the text of the news article. For more details about the input data, README.md file can be referred.

Solution Statement

Firstly, we will load the two .csv files and merge the two Fake.csv and True.csv files into a single DataFrame. Before merging the two files, Fake.csv will have an additional column "label" with values as "fake," and True.csv will have an additional column "label" with value as "true" for all the corresponding rows. Additional "label" column will ensure that the dataset has appropriate labels for each news record.

We will clean the DataFrame by removing null, missing values, and outliers. Since news text is used for feature engineering, necessary steps will be taken, such as removing stop words, punctuations, numbers, and special characters. Stemming and lemmatization will also be done for tokenization purposes to have rich text features.

We will then use the news text column for feature engineering in the classification models. For feature engineering, we will try TF-IDF, Count-Vectorization, and N-Grams methods.

For text classification, models such as Naïve-Bayes, Support Vector Machines (SVC), and Deep Learning models such as Fully Connected Feed-forward network will be created. We will also try to use ensemble methods if required to boost model performance in the final stages.

Benchmark model

Logistic Regression can be used as a benchmark model to compare the results of other models. Logistic Regression can be used for the classification of both text and numerical data. Hence Logistic Regression can be considered as a baseline and would be a good starting point to compare with the other models with more advanced feature engineering techniques and data representation.

Evaluation Metrics

We will use accuracy and precision as evaluation metrics. Test Accuracy will be used because we want to see how the model performs on test data. Furthermore, precision will also be used because we want to classify as much real news as possible.

Project Design

The project is divided into three parts. Firstly, the data exploration part will consist of loading the fake and real news data and label them, respectively. Merge the two files to make a common DataFrame. Clean the dataset to remove null, missing, and outlier values. Clean the text column for data exploration purposes. Data Exploration is primarily to understand and gain intuition about what the data consists of. We will extract the word clouds of the top 100 words of real and fake news to understand to get an intuition of the news data. We will also check the different categories of news data and the time duration for the news article.

Secondly, the feature engineering part will consist of extracting critical features from news text data such as n-grams, TF-IDF features, and Count Vectorization. We will also try the readability score for the news article to understand if it has any impact on the model.

Thirdly, the training part will consist of splitting the data into train and test. We will try the Machine Learning models and Deep Learning Models for text classification to compare performance. We are also planning to use ensemble models and methods to boost model performance.

Lastly, model performances will be compared on the test data with a focus on accuracy and precision metric to choose the best model.

Hence above is the project design I have in mind; however, there might be changes in the feature engineering phase and machine learning modeling phase since such decisions can be taken based on model performance tuning in the final steps.