Comparison between basic classifiers for Fake News Detection

Vlad Cociorva, Iustina Filipescu, Rares Gramescu, Rares Radu

20 April 2021

1 Introduction

In recent years, due to the development of online social platforms, fake news for various commercial and political purposes has been appearing in large numbers and widespread in the online world. With deceptive words, online social network users can get infected by these online fake news easily, which has brought about tremendous effects on the offline society already.

For the purpose of experimenting different methods of detecting fake news we used the following classifiers Naive-Bayes [**?**], K Nearest Neighbors [**?**], Support Vector Machine [**?**] and Decision Tree [**?**]. For comparing the results, we used F1-Score, Accuracy Score, Recall Score and Confusion Matrix.

2 Data Processing

2.1 Data Analysis

The dataset we used consisted of 50 labeled examples of news. Every row had the following columns **id**, **title**, **text** and the **rating** (true, false or partially false). The most popular words in the dataset can be seen in the following word-cloud:

2.2 Pre-Processing

The **title** and **text** columns were merged in a single column to get all the text of the news article. The newly formed **total** column was then processed by removing every stopword and lemmatizating each word using the *nltk* [**?**] library. Each word was converted into lowercase and all the extra whitespaces were removed as well.

2.3 Feature Extraction

For extracting the features from the **total** column we used TF-IDF (term frequency-inverse document frequency) vectorizer. TF-IDF enables us to gives us a way to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm.

2.4 Data split

The dataset was split into 80% for training and 20% was used for testing.

3 Methods

3.1 Naive-Bayes

We used Multinomial Naive-Bayes. Multinomial Naïve Bayes uses term frequency i.e. the number of times a given term appears in a document.

3.2 KNN

KNN is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption for underlying data distribution. In other words, the model structure determined from the dataset. This will be very helpful in practice where most of the real world datasets do not follow mathematical theoretical assumptions. Lazy algorithm means it does not need any training data points for model generation.

3.3 SVM

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

3.4 Decision Tree

A Decision Tree is constructed by asking a serious of questions with respect to a record of the dataset we have got. Each time an answer is received, a follow-up question is asked until a conclusion about the class label of the record. The series of questions and their possible answers can be organised in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edges.

4 Results

Naive-Bayes Accuracy Score = 0.7
KNN Accuracy Score = 0.7
Decision Tree Accuracy Score = 0.6
SVM Accuracy Score = 0.3

Naive-Bayes	precision	recall	f1-score	support
FALSE	0.75	0.75	0.75	4
TRUE	0.5	1.0	0.66	1
partially false	0.75	0.6	0.66	5
macro avg	0.66	0.783	0.694	10
weighted avg	0.725	0.7	0.7	10

Table 1: Naive-Bayes Classification Report

KNN	precision	recall	f1-score	support
FALSE	0.75	0.75	0.75	4
TRUE	0	0	0	1
partially false	0.66	0.8	0.72	5
macro avg	0.47	0.51	0.49	10
weighted avg	0.63	0.7	0.66	10

Table 2: KNN Classification Report

Decision Tree	precision	recall	f1-score	support
FALSE	0.75	0.75	0.75	4
TRUE	0	0	0	1
partially false	0.66	0.8	0.72	5
macro avg	0.47	0.51	0.49	10
weighted avg	0.63	0.7	0.66	10

Table 3: Decision Tree Classification Report

SVM	precision	recall	f1-score	support
FALSE	0.285	1.0	0.44	2
TRUE	1.0	0.33	0.5	3
partially false	0	0	0	5
macro avg	0.42	0.44	0.31	10
weighted avg	0.35	0.3	0.23	10

Table 4: SVM Classification Report

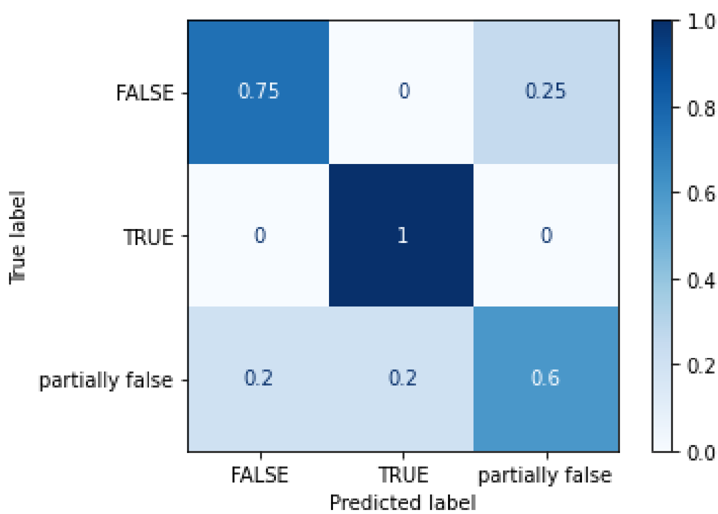

Figure 1: Naive-Bayes Confusion Matrix

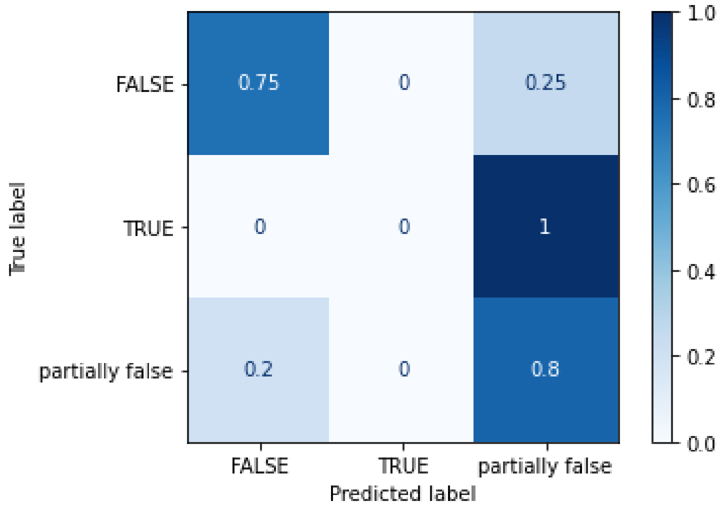

Figure 2: KNN Confusion Matrix

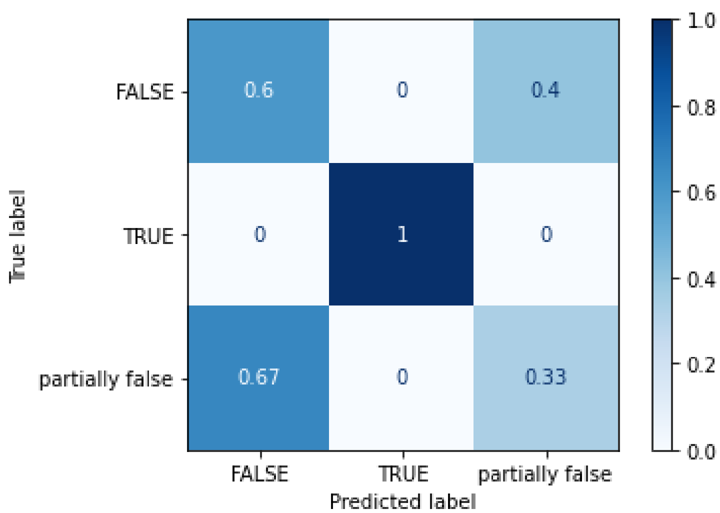

Figure 3: Decision Tree Confusion Matrix

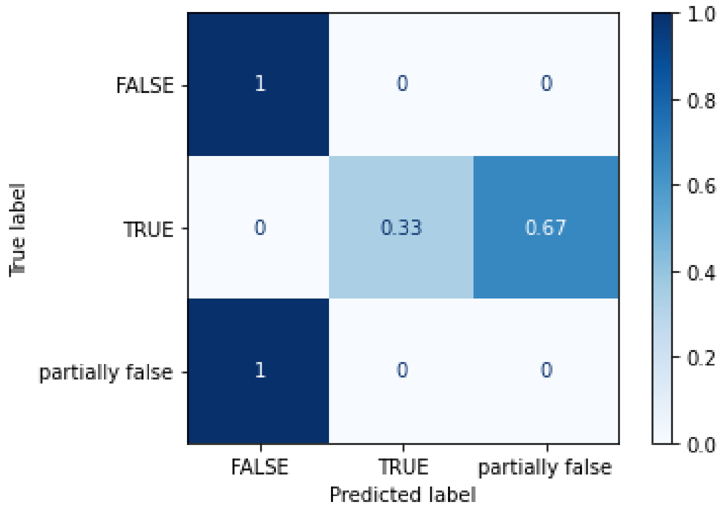

Figure 4: SVM Confusion Matrix