Research Paper

Fake News Detection
Using
Machine Learning Algorithm

Date-: 20/01/2022

Research Article

Fake News Detection Using Machine Learning Algorithms

By-: Aniket Dwivedi

Department of Computer Science, PDM University, Bahadurgarh, Haryana

Correspondence should be addressed to ani22ket16@gmail.com

Written-: 20-01-2022

Table Of Contents

1. Introduction

Fake news is a term that has come to mean different things to different people. At its core, we are defining fake news as those news stories that are false**:** the story itself is fabricated**,** with no verifiable facts, sources or quotes**.** Sometimes these stories may be propaganda that is intentionally designed to mislead the reader or maybe designed as “clickbait” written for economic incentives. In recent years, fake news stories have proliferated via social media, in part because they are so easily and quickly shared online.

The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for information dissemination that has never been witnessed in human history before. Besides other use cases, news outlets benefitted from the widespread use of social media platforms by providing updated news in near real-time to their subscribers. The news media evolved from newspapers, tabloids, and magazines to a digital form such as online news platforms, blogs, social media feeds, and other digital media formats. It became easier for consumers to acquire the latest news at their fingertips. Facebook referrals account for 70% of traffic to news websites. These social media platforms in their current state are extremely powerful and useful for their ability to allow users to discuss and share ideas and debate over issues such as democracy, education, and health. However, such platforms are also used with a negative perspective by certain entities commonly for monetary gain and in other cases for creating biased opinions, manipulating mindsets, and spreading satire or absurdity. The phenomenon is commonly known as fake news.

There has been a rapid increase in the spread of fake news in the last decade, most prominently observed in the 2016 US elections. Such proliferation of sharing articles online that do not conform to facts has led to many problems not just limited to politics but covering various other domains such as sports, health, and also science. One such area affected by fake news is the financial markets, where a rumor can have disastrous consequences and may bring the market to a halt. Our ability to take a decision relies mostly on the type of information we consume; our worldview is shaped based on the information we digest. There is increasing evidence that consumers have reacted absurdly to news that later proved to be fake. One recent case is the spread of the novel coronavirus, where fake reports spread over the Internet about the origin, nature, and behavior of the virus. The situation worsened as more people read about the fake content online. Identifying such news online is a daunting task.

2. Contribution

In the current fake news corpus, there have been multiple instances where I used supervised learning algorithms are used to classify text. However, most of the literature focuses on specific datasets or domains, most prominently the domain of politics. Therefore, the algorithm trained works 2 Complexity best on a particular type of article's domain and does not achieve optimal results when exposed to articles from other domains. Since articles from different domains have a unique textual structure, it is difficult to train a generic algorithm that works best on all particular news domains. In this paper, we propose a solution to the fake news detection problem using the machine learning ensemble approach.

My study explores different textual properties that could be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various ensemble methods that are not thoroughly explored in the current literature. The ensemble learners have proven to be useful in a wide variety of applications, as the learning models tend to reduce the error rates by using techniques such as bagging and boosting.

These techniques facilitate the training of different machine learning algorithms effectively and efficiently. I also conducted extensive experiments on 2 real-world publicly available datasets. The results validate the improved performance of the proposed technique using the 4 commonly used performance metrics namely (accuracy, precision, recall, and F-1 score).

3. Data Pre-Processing

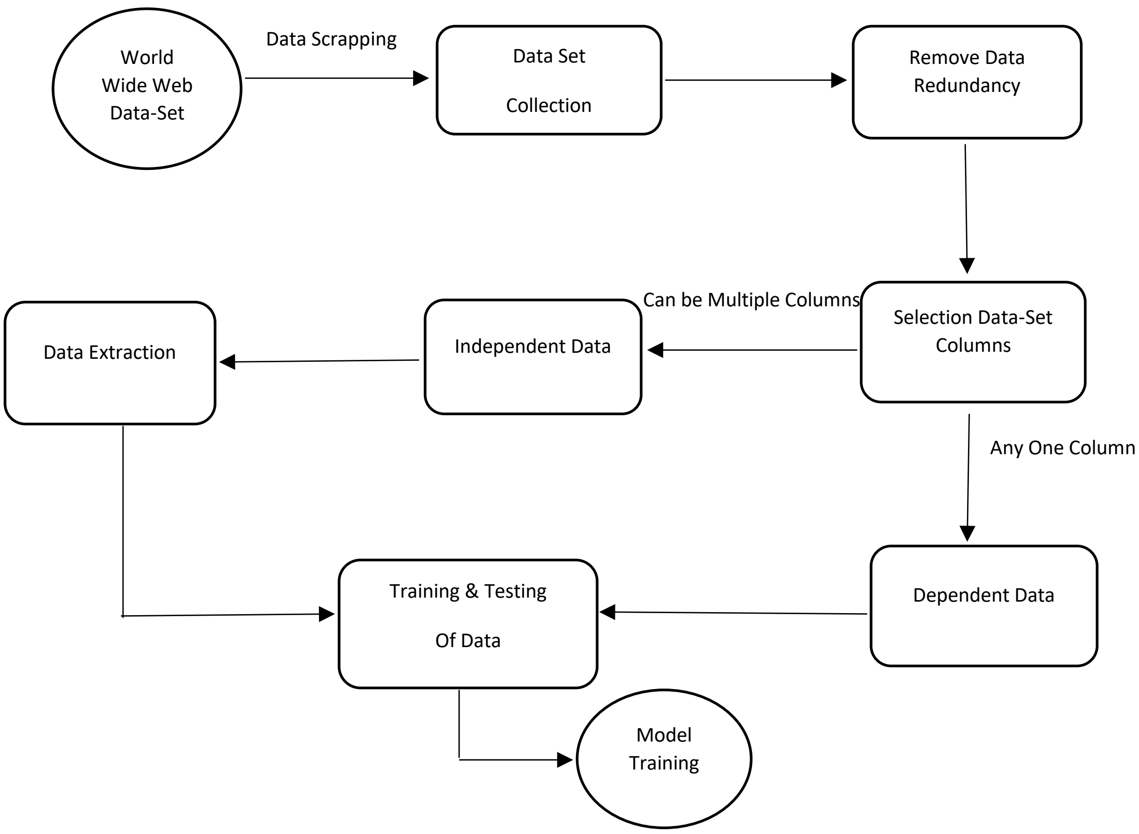
3.1 About Data

In this project, I used 2 different real-time dataset which is in CSV format and from which take only those parameters which play important role in giving information about those part from which the trained model can detect that particular news is fake or real.
So, according to model need I divided the data into two parts that are one part is independent which doesn't need any other parameters to get this, and the second part is dependent which depends on independent data. The dependent data is the only part that tells us the result.

3.2 About Given Data-Set

1. There are some missing values and also duplicate values so multiple operations are performed to remove redundancy.
2. As data is in string format so to fit and transform the data many features are used of the sklearn library to get a better format.

Fig. 3.1 Data Pre-Processing



4. Model Used

In this, a total of 4 models is used from which only the maximum accuracy model is taken into consideration for further process.

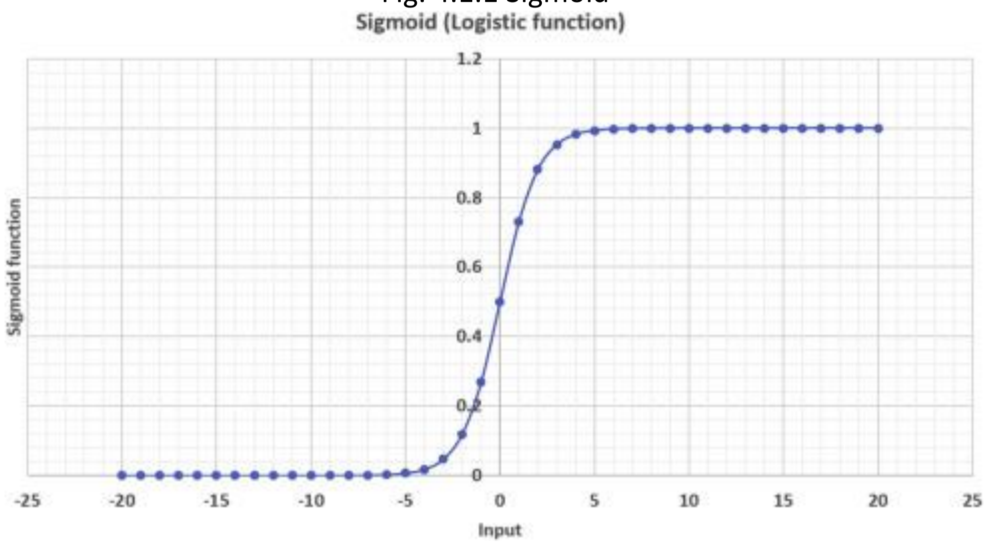
4.1 Logistic Regression

Logistic regression is another powerful supervised ML algorithm used for binary classification problems (when the target is categorical). The best way to think about logistic regression is that it is a linear regression but for classification problems. Logistic regression essentially uses a logistic function defined below to model a binary output variable. The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio (will be defined shortly).

Logisticfunction=11+e−x

In the logistic function equation, *x* is the input variable. Let's feed-in values −20 to 20 into the logistic function. The inputs have been transferred to between 0 and 1.

Fig. 4.1.1 Sigmoid



As opposed to linear regression where MSE or RMSE is used as the loss function, logistic regression uses a loss function referred to as "maximum likelihood estimation (MLE)" which is a conditional probability. If the probability is greater than 0.5, the predictions will be classified as class 0. Otherwise, class 1 will be assigned. Before going through logistic regression derivation, let's first define the logit function. Logit function is defined as the natural log of the odds. A probability of 0.5 corresponds to a logit of 0, probabilities smaller than 0.5 correspond to negative logit values, and probabilities greater than 0.5 correspond to positive logit values. It is important to note that as

illustrated in Fig. 5.17, logistic function ranges between 0 and 1 ($P \in [0,1]$) while logit function can be any real number from minus infinity to positive infinity ($P \in [-\infty, \infty]$).

odds=P1−P→logit(P)=ln(P1−P)

Let's set logit of *P* to be equal to *mx* + *b*, therefore:

logit(P)=mx+b→mx+b=ln(P1−P)(P1−P)=e(mx+b)→P=e(mx+b)1+e(mx+b)→P(x)=11+e−(mx+b)

Before using logistic regression in sci-kit-learn, let's review some very important classification metrics used for evaluating a classification model such as logistic regression.

4.2 Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Given training vectors xi∈Rn, i=1,…, l and a label vector y∈Rl, a decision tree recursively partitions the feature space such that the samples with the same labels or similar target values are grouped.

Let the data at node m be represented by Qm with Nm samples. For each candidate split θ=(j,tm) consisting of a feature j and threshold tm, partition the data into Qmleft(θ) and Qmright(θ) subsets

Qmleft(θ)={(x,y)|xj<=tm}Qmright(θ)=Qm\Qmleft(θ)

The quality of a candidate split of node m is then computed using an impurity function or loss function H(), the choice of which depends on the task being solved.

G(Qm,θ)=NmleftNmH(Qmleft(θ))+NmrightNmH(Qmright(θ))
Select the parameters that minimise the impurity

θ∗=argminθG(Qm,θ)

Recurse for subsets Qmleft(θ∗) and Qmright(θ∗) until the maximum allowable depth is reached, Nm<minsamples or Nm=1.

4.3 Random Forest

A random forest is a machine learning technique that’s used to solve regression and classification problems. It utilizes ensemble learning,

which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The ‘forest’ generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

When performing Random Forests based on classification data, you should know that you are often using the Gini_index, or the formula used to decide how nodes on a decision tree branch.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

This formula uses the class and probability to determine the Gini of each branch on a node, determining which of the branches is more likely to occur. Here, *pi* represents the relative frequency of the class you are observing in the dataset and *c* represents the number of classes.

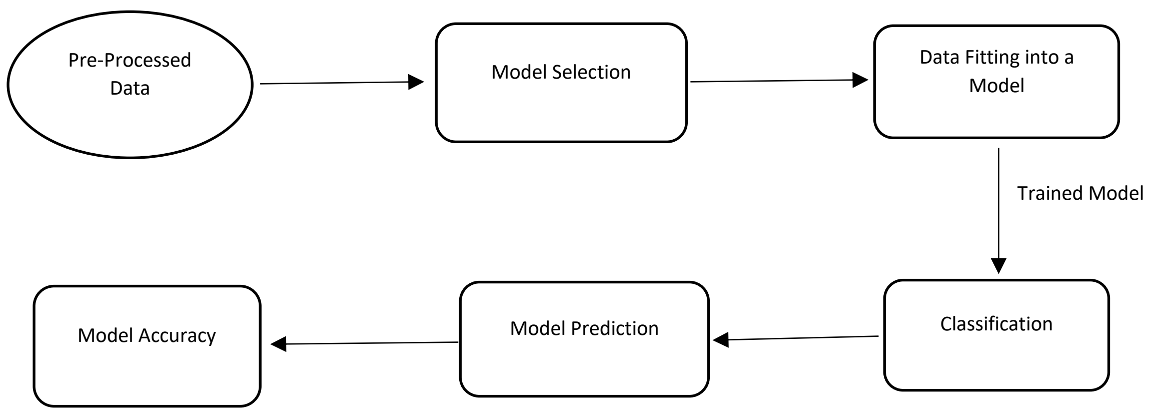
4.4 Passive Classifier

The Passive-Aggressive algorithms are a family of Machine learning algorithms that are not very well known by beginners and even intermediate Machine Learning enthusiasts. However, they can be very useful and efficient for certain applications.

Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.
Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

class sklearn.linear_model.PassiveAggressiveClassifier(*, C=1.0, fit_intercept=True, max_iter=1000, tol=0.001, early_stopping=False, validation_fraction=0.1, n_iter_no_change=5, shuffle=True, verbose=0, loss='hinge', n_jobs=None, random_state=None, warm_start=False, class_weight=None, average=False)

Fig 4.1 Data Fit & Transform by Model



5. Results And Discussion

- A **classification report** is a performance evaluation metric in machine learning. It is used to show the precision, recall, F1 Score, and support of your trained classification model.
- **Accuracy** is often the most used metric representing the percentage of correctly predicted observations, either true or false. To calculate the accuracy of model performance, the following equation can be used:

  Accuracy = (TP + TN) / (TP + TN + FP + FN)
- **Recall** represents the total number of positive classifications out of true class. In our case, it represents the number of articles predicted as true out of the total number of true articles.

  Recall =TP/ (TP + FN)
- Conversely, **precision score** represents the ratio of true positives to all events predicted as true. In our case, precision shows the number of articles that are marked as true out of all the positively predicted (true) articles:

  Precision = TP/ (TP + FP)
- **F1-score** represents the trade-off between precision and recall. It calculates the harmonic mean between each of the two. Thus, it takes both the false positive and the false-negative observations into account. F1-score can be calculated using the following formula:

  F1 − score =2 Precision × Recall / (Precision + Recall)

5.1 Classification Report

5.1.1 Logistic Regression

	precision	recall	f1-score	support
Fake	0.99	0.96	0.97	534
Real	0.95	0.99	0.97	463
accuracy			0.97	997
macro avg	0.97	0.97	0.97	997
weighted avg	0.97	0.97	0.97	997

5.1.2 Decision Tree

	precision	recall	f1-score	support
Fake	0.96	0.96	0.96	534
Real	0.96	0.95	0.96	463
accuracy			0.96	997
macro avg	0.96	0.96	0.96	997
weighted avg	0.96	0.96	0.96	997

5.1.3 Passive-Aggressive Classifier

	Precision	recall	f1-score	support
Fake	0.99	0.99	0.99	534
Real	0.98	0.99	0.99	463
accuracy			0.99	997
macro avg	0.99	0.99	0.99	997
weighted avg	0.99	0.99	0.99	997

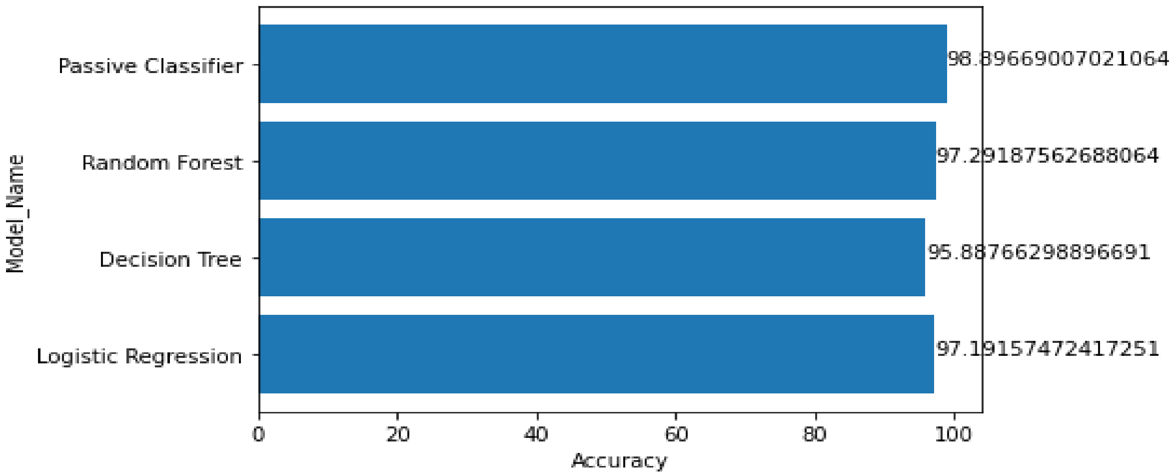
5.1.4 Random Forest

	precision	recall	f1-score	support
Fake	1.00	0.95	0.97	534
Real	0.94	1.00	0.97	463
accuracy			0.97	997
macro avg	0.97	0.97	0.97	997
weighted avg	0.97	0.97	0.97	997

6. Model Accuracy Graph

Try to Accuracies of the different models through graphs so that can compare which model gives better results and is used for deployment.

Fig 6.1 Accuracy Graph



The above graph, clearly shows that the passive classifier gives more accuracy and result also working properly.

7. Conclusion

The task of classifying news manually requires in-depth knowledge of the domain and expertise to identify anomalies in the text. In this research, we discussed the problem of classifying fake news articles using machine learning models and different techniques. The data we used in our work is collected from the World Wide Web and contains Logistic regression (LR) 0.97 0.97 0.97, Decision Tree (DTC) 0.96 0.96 0.96, Random Forest (RFC) 0.97 0.97 0.97, Passive Classifier (PAC) 0.99 0.99 0.99. Precision, recall, and F1-score over all datasets in the 5.1 column. Complexity 9 articles from various domains to cover most of the news rather than specifically classifying political news. The primary aim of the research is to identify patterns in text that differentiate fake articles from true news. We extracted different textual features from the articles using a feature-extraction library and used the feature set as an input to the models. The learning models were trained and parameter-tuned to obtain optimal accuracy. At least all models have achieved comparatively higher accuracy. I used multiple classification reports to compare the results for each algorithm. Fake news detection has many open issues that require the attention of researchers. For instance, to reduce the spread of fake news, identifying key elements involved in the spread of news is an important step. Graph theory and machine learning techniques can be employed to identify the key sources involved in the spread of fake news. Likewise, real-time fake news identification in videos can be another possible future direction.

8. References

URL:" https://www.kaggle.com/jruvika/fake-news-detection "

(i) License-: CC BY 4.0

(ii) Splash icon-: Icon by Freepik, available on Flaticon.

(iii) Splash banner-: Wallpaper by jcomp, available on Freepik.