A Thesis/Project/Dissertation Report

on

FAKE NEWS DETECTION

Submitted in partial fulfillment of the
requirement for the award of the degree of

B-Tech/CSE

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. Praveen Mishra

Submitted By

Rahul Kumar Gupta 19SCSE1010333
Anand Kasaudhan 19SCSE1010668

SUBMITTED TO
Mr. A. Midhun. Kumar

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA
NOV-2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"FAKE NEWS DETECTION"** in partial fulfillment of the requirements for the award of the B-tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Mr Praveen Mishra, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Rahul Kumar Gupta 19SCSE1010333

Anand Kasaudhan 19SCSE1010668

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name
Mr Praveen Mishra

Acknowledgement

An endeavor over a long period can be successful with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.

We wish to express our sincere thanks and gratitude to our project guide Mr Praveen Mishra Sir, Assistant Professor, Department of Computer Science and Engineering, for the simulating discussions, in analyzing problems associated with our project work and for guiding us throughout the project. Project meetings were highly informative. We express our warm and sincere thanks for the encouragement, untiring guidance and the confidence she had shown in us. We are immensely indebted for her valuable guidance throughout our project.

We also thank all the staff members of CSE department for their valuable advices. We also thank supporting staff for providing resources as and when required. .

PROJECT STUDENTS:
Rahul Kumar Gupta
Anand Kasaudhan

CERTIFICATE

The Final Project Viva-Voce examination of 1. RAHUL KUMAR GUPTA (19SCSE1010333) 2. ANAND KASAUDHAN (19SCSE1010668) has been held on _____ and their work is recommended for the award of Bachelor of Technology in the School of Computing Science and Engineering of Galgotias University, Greater Noida

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Abstract

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 the human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article. In this work, we propose to use machine learning ensemble approach for automated classification of news articles. Our study explores different textual properties that can 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 and evaluate their performance on 4 real world datasets. Experimental evaluation confirms the superior performance of our proposed ensemble learner approach in comparison to individual learners.

Contents

List of Table

List of Figures

CHAPTER-1

Introduction

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 the 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 its 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.

Our Contribution

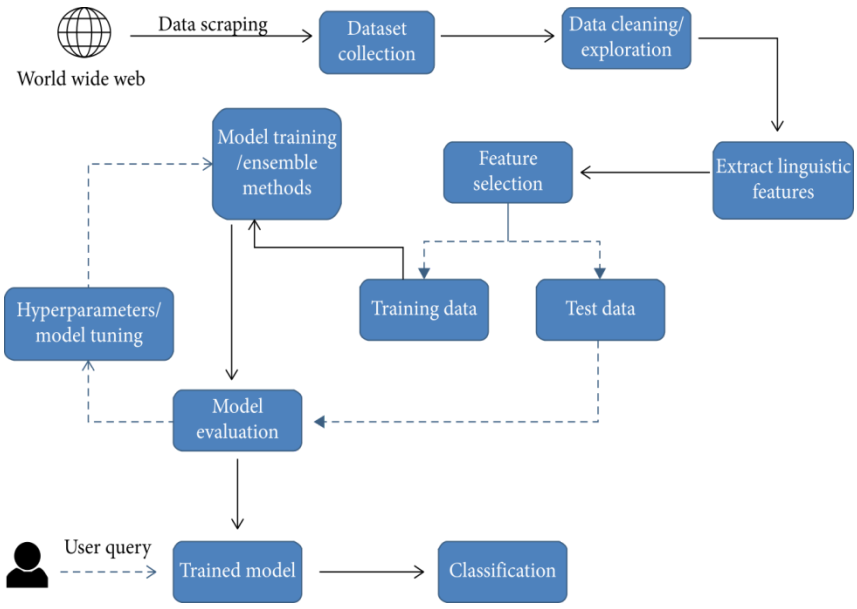
In the current fake news corpus, there have been multiple instances where both supervised and unsupervised learning algorithms are used to classify text. However, most of the literature focuses on specific datasets or domains, most prominently the politics domain. Therefore, the algorithm trained works 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. Our 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 have the tendency to reduce error rate by using techniques such as bagging and boosting. These techniques facilitate the training of different machine learning algorithms in an effective and efficient manner. We also conducted extensive experiments on 4 real world publicly available datasets.

Chapter 2

Project Design

we are expanding on the current literature by introducing ensemble techniques with various linguistic feature sets to classify news articles from multiple domains as true or fake. The ensemble techniques along with Linguistic Inquiry and Word Count (LIWC) feature set used in this research are the novelty of our proposed approach.



Algorithms : -

We used the following learning algorithms in conjunction with our proposed methodology to evaluate the performance of fake news detection classifiers.

As we are classifying text on the basis of a wide feature set, with a binary output (true/false or true article/fake article), a logistic regression (LR) model is used, since it provides the intuitive equation to classify problems into binary or multiple classes. We performed hyperparameters tuning to get the best result for all individual datasets, while multiple parameters are tested before acquiring the maximum accuracies from LR model.

Support Vector Machine :-

Support vector machine (SVM) is another model for binary classification problem and is available in various kernels functions. The objective of an SVM model is to estimate a hyperplane (or decision boundary) on the basis of feature set to classify data points. The dimension of hyperplane varies according to the number of features. As there could be multiple possibilities for a hyperplane to exist in an *N*-dimensional space, the task is to identify the plane that separates the data points of two classes with maximum margin

Multilayer Perceptron :-

A multilayer perceptron (MLP) is an artificial neural network, with an input layer, one or more hidden layers, and an output layer. MLP can be as simple as having each of the three layers; however, in our experiments we have fine-tuned the model with various parameters and number of layers to generate an optimum predicting model.

***K*-Nearest Neighbors (KNN) :-**

KNN is an unsupervised machine learning model where a dependent variable is not required to predict the outcome on a specific data. We provide enough training data to the model and let it decide to which particular neighborhood a data point

belongs. KNN model estimates the distance of a new data point to its nearest neighbors, and the value of *K* estimates the majority of its neighbors' votes; if the value of *K* is 1, then the new data point is assigned to a class which has the nearest distance.

Random Forest (RF) :-

Random forest (RF) is an advanced form of decision trees (DT) which is also a supervised learning model. RF consists of large number of decision trees working individually to predict an outcome of a class where the final prediction is based on a class that received majority votes. The error rate is low in random forest as compared to other models, due to low correlation among trees. Our random forest model was trained using different parameters; i.e., different numbers of estimators were used in a grid search to produce the best model that can predict the outcome with high accuracy. There are multiple algorithms to decide a split in a decision tree based on the problem of regression or classification. For the classification problem, we have used the Gini index as a cost function to estimate a split in the dataset.

Chapter 3

Functionality/Working of Project

A. Static Search Implementation :-

In static part, we have trained and used 3 out of 4 algorithms for classification. They are Naive Bayes , Random Forest and Logistic Regression.

Step 1: In first step, we have extracted features from the already pre-processed dataset. These features are; Bag-of-words, Tf-Idf Features and N-grams.

Step 2: Here, we have built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers. We have used Naive-bayes, Logistic Regression, and Random forest classifiers from sklearn. Each of the extracted features was used in all of the classifiers.

Step 3: Once fitting the model, we compared the f1 score and checked the confusion matrix.

Step 4: After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification.

Step 5: We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier.

Step 6: Finally selected model was used for fake news detection with the probability of truth.

Step 7: Our finally selected and best performing classifier was Logistic Regression which was then saved on disk. It will be used to classify the fake news.

B. Dynamic Search Implementation :-

Our dynamic implementation contains 3 search fields which are-
1) Search by article content.
2) Search using key terms.
3) Search for website in database.

In the first search field we have used Natural Language Processing for the first search field to come up with a proper solution for the problem, and hence we have attempted to create a model which can classify fake news according to the terms used in the newspaper articles. Our application uses NLP techniques like CountVectorization and TF-IDF Vectorization before passing it through a Passive Aggressive Classifier to output the authenticity as a percentage probability of an article. The second search field of the site asks for specific keywords to be searched on the net upon which it provides a suitable output for the percentage probability of that term actually being present in an article or a similar article with those keyword references in it.The third search field of the site accepts a specific

website domain name upon which the implementation looks for the site in our true sites database or the blacklisted sites database. The true sites database holds the domain names which regularly provide proper and authentic

news and vice versa. If the site isn’ t found in either of the databases then the implementation doesn’ t classify the domain i t simply states that the news aggregator does not exist.

Working :-

The problem can be broken down into 3 statements-

1) Use NLP to check the authenticity of a news article.

2) If the user has a query about the authenticity of a search query then we he/she can directly search on our platform and using our custom algorithm we output a confidence score.

3) Check the authenticity of a news source.

CODE AND IMPLEMENTATION

Python:-

```
from flask import Flask, render_template, request
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
import pickle
import pandas as pd
from sklearn.model_selection import train_test_split

app = Flask(_name_)
tfvect = TfidfVectorizer(stop_words='english', max_df=0.7)
loaded_model = pickle.load(open('model.pkl', 'rb'))
dataframe = pd.read_csv('news.csv')
x = dataframe['text']
y = dataframe['label']
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=0)

def fake_news_det(news):
      tfid_x_train = tfvect.fit_transform(x_train)
      tfid_x_test = tfvect.transform(x_test)
      input_data = [news]
      vectorized_input_data = tfvect.transform(input_data)
```

```
      prediction = loaded_model.predict(vectorized_input_data)
      return prediction

@app.route('/')
def home():
      return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
      if request.method == 'POST':
            message = request.form['message']
            pred = fake_news_det(message)
            print(pred)
            return render_template('index.html', prediction=pred)
      else:
            return render_template('index.html',
prediction="Something went wrong")

if _name_ == '_main_':
      app.run(debug=True)
```

Flask &Python:-

```
from sklearn.feature_extraction.text import TfidfVectorizer
text = ['Hello Kushal Bhavsar here, I love machine
learning','Welcome to the Machine learning hub' ]


vect = TfidfVectorizer()
vect.fit(text)

## TF will count the frequency of word in each document. and IDF
print(vect.idf_)
```

```
[1.40546511  1.40546511  1.40546511  1.40546511  1.40546511  1.
1.40546511 1.  1.40546511 1.40546511 1.40546511]


print(vect.vocabulary_)
example = text[0]
example
example = vect.transform([example])
print(example.toarray())


import os
os.chdir("D:/Fake News Detection")
import pandas as pd
dataframe = pd.read_csv('news.csv')
dataframe.head()


x = dataframe['text']
y = dataframe['label']

print(x)

print(y)

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix


x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)
y_train
y_train


tfvect = TfidfVectorizer(stop_words='english',max_df=0.7)
tfid_x_train = tfvect.fit_transform(x_train)
tfid_x_test = tfvect.transform(x_test)
```

```
classifier = PassiveAggressiveClassifier(max_iter=50)
classifier.fit(tfid_x_train,y_train)
y_pred = classifier.predict(tfid_x_test)
score = accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')


 cf = confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])
print(cf)

def fake_news_det(news):
      input_data = [news]
      vectorized_input_data = tfvect.transform(input_data)
      prediction = classifier.predict(vectorized_input_data)
      print(prediction)

fake_news_det('U.S. Secretary of State John F. Kerry said Monday
that he will stop in Paris later this week, amid criticism that no
top American officials attended Sundayâ€™s unity march against
terrorism.
fake_news_det("""Go to Article
President Barack Obama has been campaigning hard for the woman who
is supposedly going to extend his legacy four more years. The only
problem with stumping for Hillary Clinton, however, is sheâ€™s
not exactly a candidate easy to get too enthused about.

import pickle
pickle.dump(classifier,open('model.pkl', 'wb'))

# load the model from disk
loaded_model = pickle.load(open('model.pkl', 'rb'))

def fake_news_det1(news):
      input_data = [news]
      vectorized_input_data = tfvect.transform(input_data)
      prediction = loaded_model.predict(vectorized_input_data)
      print(prediction)
```

fake_news_det1("""Go to Article
President Barack Obama has been campaigning hard for the woman who is supposedly going to extend his legacy four more years. The only problem with stumping for Hillary Clinton, however, is sheâ€™s not exactly a candidate easy to get too enthused about. """)

fake_news_det1("""U.S. Secretary of State John F. Kerry said Monday that he will stop in Paris later this week, amid criticism that no top American officials attended Sundayâ€™s unity march against terrorism.""")

fake_news_det('''U.S. Secretary of State John F. Kerry said Monday that he will stop in Paris later this week, amid criticism that no top American officials attended Sundayâ€™s unity march against terrorism.''')

🔹 HTML

```
<!DOCTYPE html>
<html >
<head>
   <meta charset="UTF-8">
   <title>Fake News📰 Detection System</title>
   <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:
300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
```

```
</head>

<body>
  <div class="login">
      <h1>Fake News   Detector</h1>

      <form action="{{ url_for('predict')}}" method="POST">
            <textarea   name="message" rows="6" cols="50"
required="required" style="font-size: 18pt"></textarea>
            <br> </br>
            <button type="submit" class="btn btn-primary btn-block
btn-large">Predict</button>

                  <div class="results">

      {% if prediction == ['FAKE']%}
      <h2 style="color:red;">Looking Spam⚠️News📰 </h2>
      {% elif prediction == ['REAL']%}
                        <h2 style="color:green;"><b>Looking Real
News📰</b></h2>
      {% endif %}

      </div>

      </form>

</div>
```

🎨 CSS

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding:
4px 10px 4px; margin-bottom: 0; font-size: 13px; line-height:
18px; color: #333333; text-align: center;text-shadow: 0 1px 1px
rgba(255, 255, 255, 0.75); vertical-align: middle;
```

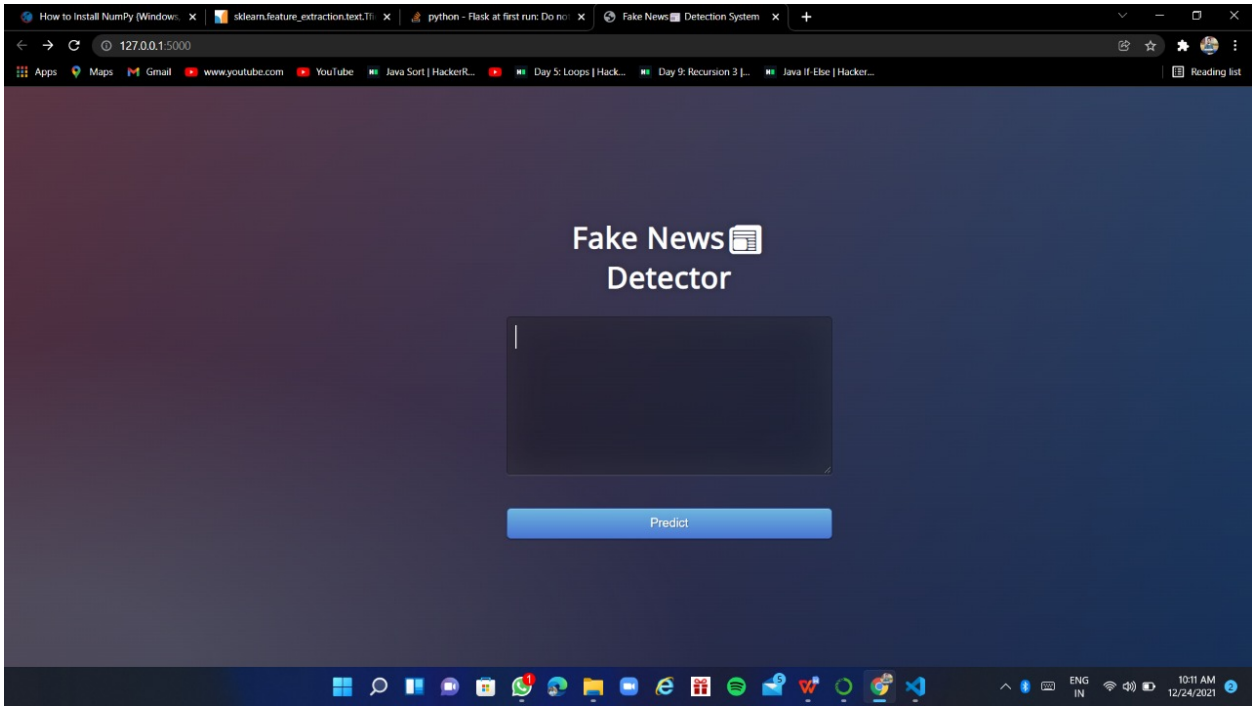
```
background-color: #f5f5f5; background-image:
-moz-linear-gradient(top, #ffffff, #e6e6e6); background-image:
-ms-linear-gradient(top, #ffffff, #e6e6e6); background-image:
-webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));
background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);
background-image: -o-linear-gradient(top, #ffffff, #e6e6e6);
background-image: linear-gradient(top, #ffffff, #e6e6e6);
background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff,
endColorstr=#e6e6e6, GradientType=0); border-color: #e6e6e6
#e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0,
0.1) rgba(0, 0, 0, 0.25); border: 1px solid #e6e6e6;
-webkit-border-radius: 4px; -moz-border-radius: 4px;
border-radius: 4px; -webkit-box-shadow: inset 0 1px 0 rgba(255,
255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); -moz-box-shadow:
inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0,
0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px
2px rgba(0, 0, 0, 0.05); cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled,
.btn[disabled] { background-color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height:
normal; -webkit-border-radius: 5px; -moz-border-radius: 5px;
border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none;
background-color: #e6e6e6; background-position: 0 -15px;
-webkit-transition: background-position 0.1s linear;
-moz-transition: background-position 0.1s linear; -ms-transition:
background-position 0.1s linear; -o-transition:
background-position 0.1s linear; transition: background-position
0.1s linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0,
0, 0, 0.25); color: #ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image:
-moz-linear-gradient(top, #6eb6de, #4a77d4); background-image:
-ms-linear-gradient(top, #6eb6de, #4a77d4); background-image:
-webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4));
background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -o-linear-gradient(top, #6eb6de, #4a77d4);
```

```
background-image: linear-gradient(top, #6eb6de, #4a77d4);
background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0);   border: 1px solid #3762bc;
text-shadow: 1px 1px 1px rgba(0,0,0,0.4); box-shadow: inset 0 1px
0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5); }
.btn-primary:hover, .btn-primary:active, .btn-primary.active,
.btn-primary.disabled, .btn-primary[disabled] { filter: none;
background-color: #4a77d4; }
.btn-block { width: 100%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box;
-ms-box-sizing:border-box; -o-box-sizing:border-box;
box-sizing:border-box; }

html { width: 100%; height:100%; overflow:hidden; }

body {
      width: 100%;
      height:100%;
      font-family: 'Open Sans', sans-serif;
      background: #092756;
      color: #fff;
      font-size: 18px;
      text-align:center;
      letter-spacing:1.2px;
      background: -moz-radial-gradient(0% 100%, ellipse cover,
rgba(104,128,138,.4) 10%,rgba(138,114,76,0)
40%),-moz-linear-gradient(top,    rgba(57,173,219,.25) 0%,
rgba(42,60,87,.4) 100%), -moz-linear-gradient(-45deg,    #670d10
0%, #092756 100%);
      background: -webkit-radial-gradient(0% 100%, ellipse cover,
rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%),
-webkit-linear-gradient(top,    rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -webkit-linear-gradient(-45deg,
  #670d10 0%,#092756 100%);
      background: -o-radial-gradient(0% 100%, ellipse cover,
rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%),
-o-linear-gradient(top,    rgba(57,173,219,.25)
```

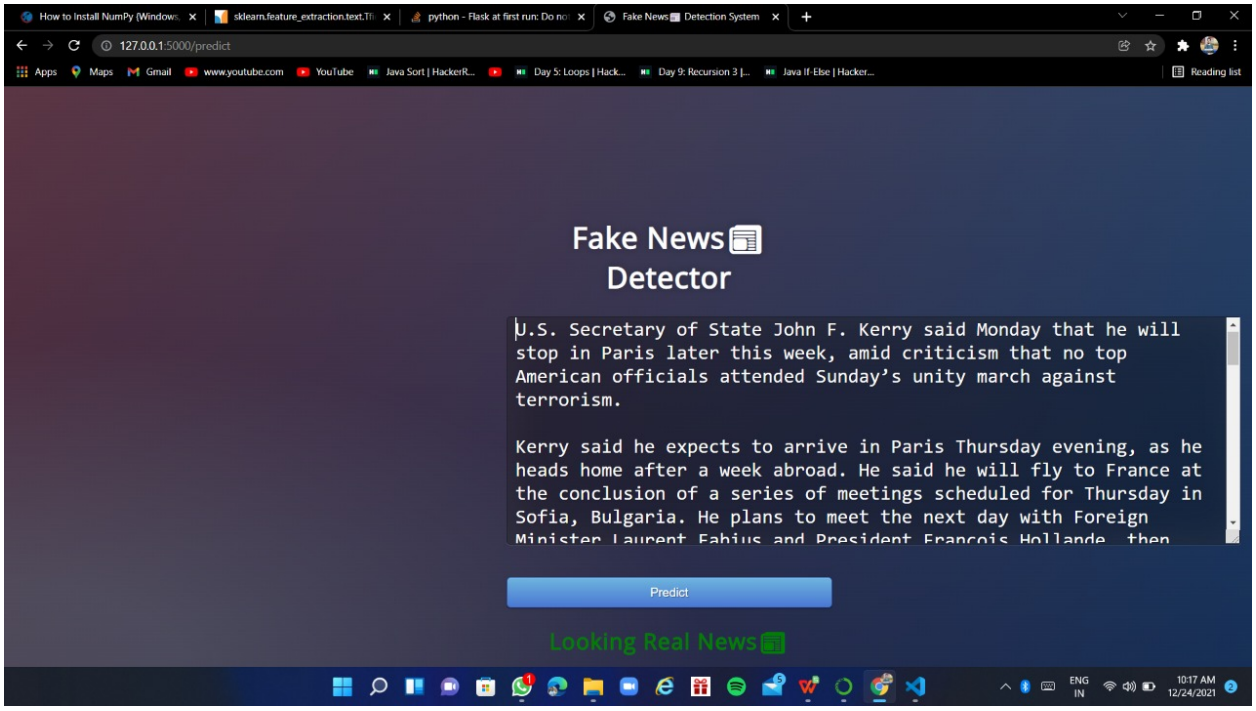
```
0%,rgba(42,60,87,.4) 100%), -o-linear-gradient(-45deg,    #670d10
0%,#092756 100%);
      background: -ms-radial-gradient(0% 100%, ellipse cover,
rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%),
-ms-linear-gradient(top,    rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -ms-linear-gradient(-45deg,    #670d10
0%,#092756 100%);
      background: -webkit-radial-gradient(0% 100%, ellipse cover,
rgba(104,128,138,.4) 10%,rgba(138,114,76,0) 40%),
linear-gradient(to bottom,    rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), linear-gradient(135deg,    #670d10
0%,#092756 100%);
      filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#3E1D6D', endColorstr='#092756',GradientType=1 );

}
.login {
      position: absolute;
      top: 40%;
      left: 50%;
      margin: -150px 0 0 -150px;
      width:400px;
      height:400px;
}

.login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3);
letter-spacing:1px; text-align:center; }

textarea {
      width: 100%;
      margin-bottom: 10px;
      background: rgba(0,0,0,0.3);
      border: none;
      outline: none;
      padding: 10px;
      font-size: 25px;
      color: #fff;
      text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
      border: 1px solid rgba(0,0,0,0.3);
```

```
      border-radius: 4px;
      box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px
1px rgba(255,255,255,0.2);
      -webkit-transition: box-shadow .5s ease;
      -moz-transition: box-shadow .5s ease;
      -o-transition: box-shadow .5s ease;
      -ms-transition: box-shadow .5s ease;
      transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4),
0 1px 1px rgba(255,255,255,0.2); }
```

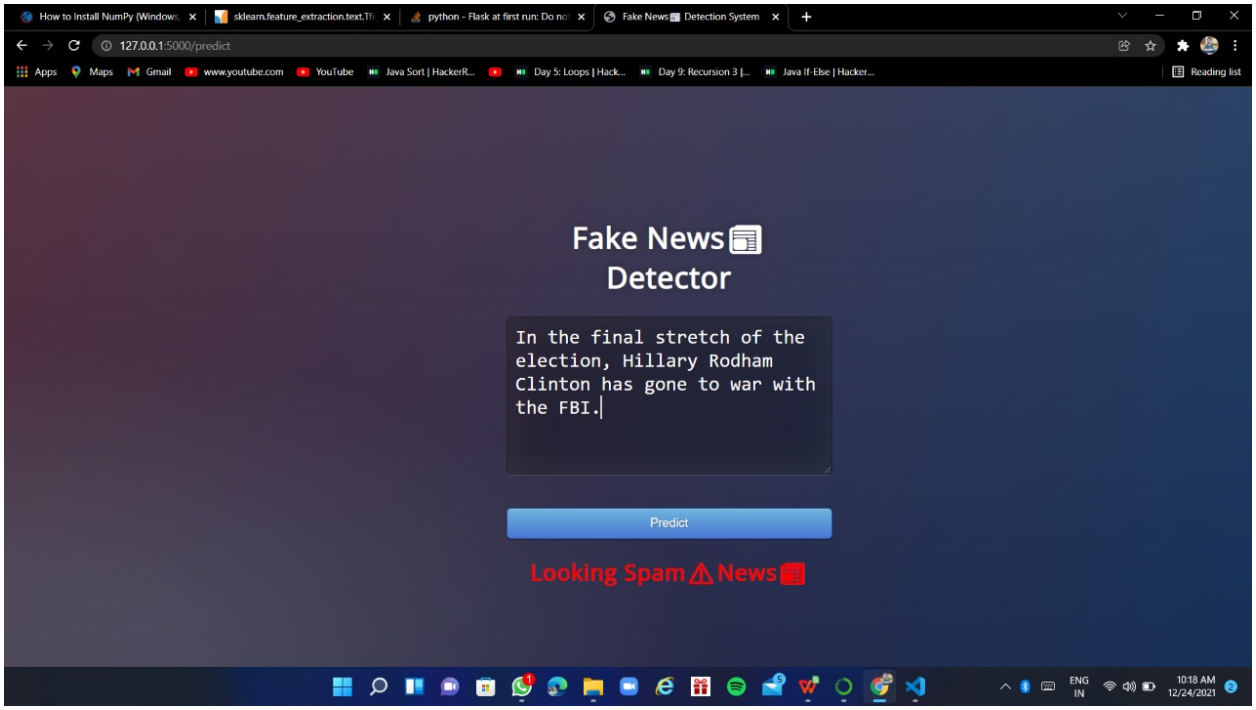
❖ OutPut



- Real News



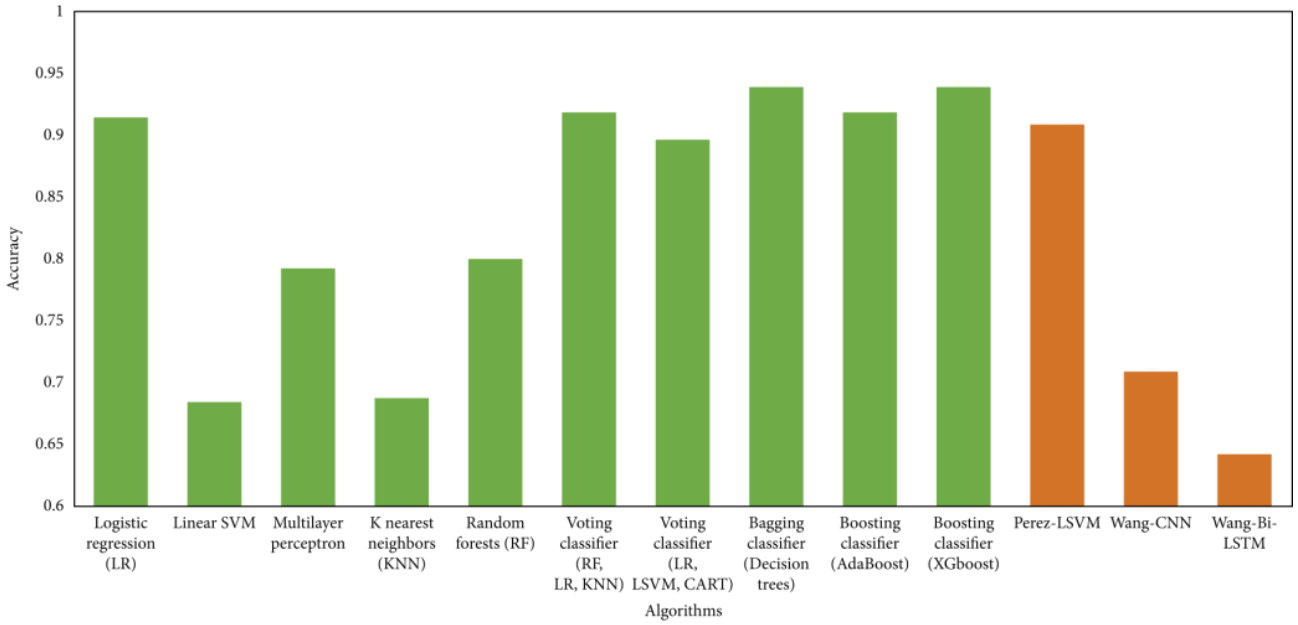
- Fake News



Results and Discussion

It is evident that the maximum accuracy achieved on DS1 (ISOT Fake News Dataset) is 99%, achieved by random forest algorithm and Perez-LSVM. Linear SVM, multilayer perceptron, bagging classifiers, and boosting classifiers achieved an accuracy of 98%. The average accuracy attained by ensemble learners is 97.67% on DS1, whereas the corresponding average for individual learners is 95.25%. The absolute difference between individual learners and ensemble learners is 2.42% which is not significant. Benchmark algorithms Wang-CNN and Wang-Bi-LSTM performed poorer than all other algorithms. On DS2, bagging classifier (decision trees) and boosting classifier (XGBoost) are the best performing algorithms, achieving an accuracy of 94%. Interestingly, linear SVM, random forest, and Perez-LSVM performed poorly on DS2. Individual learners reported an accuracy of 47.75%, whereas ensemble learners' accuracy is 81.5%. A similar trend is observed for DS3, where individual learners' accuracy is 80% whereas ensemble learners' accuracy is 93.5%. However, unlike DS2, the best performing algorithm on DS3 is Perez-LSVM which achieved an accuracy of 96%. On DS4 (DS1, DS2, and DS3 combined), the best performing algorithm is random forest (91% accuracy). On average, individual learners achieved an accuracy of 85%, whereas ensemble learners achieved an accuracy of 88.16%. The worst performing algorithm is Wang-Bi-LSTM which achieved an accuracy of 62%.

Figure summarizes the average accuracy of all algorithms over the 4 datasets. Overall, the best performing algorithm is bagging classifier (decision trees) (accuracy 94%), whereas the worst performing algorithm is Wang-Bi-LSTM (accuracy 64.25%). Individual learners' accuracy is 77.6% whereas the accuracy of ensemble learners is 92.25%. Random forest achieved better accuracy on all datasets except DS2. However, accuracy score alone is not a good measure to evaluate the performance of a model; therefore, we also evaluate performance of learning models on the basis of recall, precision, and F1-score.


1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
Accuracy
Logistic regression (LR)
Linear SVM
Multilayer perceptron
K nearest neighbors (KNN)
Random forests (RF)
Voting classifier (RF, LR, KNN)
Voting classifier (LR, LSVM, CART)
Bagging classifier (Decision trees)
Boosting classifier (AdaBoost)
Boosting classifier (XGboost)
Perez-LSVM
Wang-CNN
Wang-Bi-LSTM
Algorithms

Conclusion

The main contribution of this project is support for the idea that machine learning could be useful in a novel way for the task of classifying fake news. Our findings show that after much pre-processing of relatively small dataset, a simple CNN is able to pick up on a diverse set of potentially subtle language patterns that a human may (or may not) be able to detect. Many of these language patterns are intuitively useful in a humans manner of classifying fake news. Some such intuitive patterns that our model has found to indicate fake news include generalizations, colloquialisms and exaggerations. Likewise, our model looks for indefinite or inconclusive words, referential words, and evidence words as patterns that characterize real news. Even if a human could detect these patterns, they are not able to store as much information as a CNN model, and therefore, may not understand the complex relationships between the detection of these patterns and the decision for classification

Future Scope

Through the work done in this project, we have shown that machine learning certainly does have the capacity to pick up on sometimes subtle language patterns that may be difficult for humans to pick up on. The next steps involved in this project come in three different aspects. The first of aspect that could be improved in this project is augmenting and increasing the size of the dataset. We feel that more data would be beneficial in ridding the model of any bias based on specific patterns in the source. There is also question as to weather or not the size of our dataset is sufficient.

The second aspect in which this project could be expanded is by comparing it to humans performing the same task. Comparing the accuracies would be beneficial in deciding whether or not the dataset is representative of how difficult the task of separating fake from real news is.

Reference

1.A. Douglas, “News consumption and the new electronic media,” *The International Journal of Press/Politics*, vol. 11, no. 1, pp. 29–52, 2006.View at:Publisher Site |Google Scholar

2.J. Wong, “Almost all the traffic to fake news sites is from facebook, new data show,” 2016.View at: Google Scholer.

3.D. M. J. Lazer, M. A. Baum, Y. Benkler et al., “The science of fake news,” *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.View at:Publisher Site |Google Scholar

4.S. A. García, G. G. García, M. S. Prieto, A. J. M. Guerrero, and C. R. Jiménez, “The impact of term fake news on the scientific community scientific performance and mapping in web of science,” *Social Sciences*, vol. 9, no. 5, 2020.View at:Google Scholer

5.A. D. Holan, *2016 Lie of the Year: Fake News*, Politifact, Washington, DC, USA, 2016.

Copyright