Fake News Detection

Summary	Fake new detection using NLP and performing sentiment analysis
URL	
Category	Machine Learning, Web
Environment	NA
Status	Version 1
Feedback Link	
Team No	7
Team Members	Maneendar Sorupaka (sorupaka.m@husky.neu.edu) Pratiksha Sawant (sawant.prat@husky.neu.edu) Rajsharavan Senthilvelan (senthilvelan.r@husky.neu.edu) Reema Mehta (mehta.r@husky.neu.edu)

Team 7

Maneendar Sorupaka
Pratiksha Sawant
Rajsharavan Senthilvelan
Reema Mehta

Project goal

Our goal is to explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem. We believe by using these technologies we can automate and help the users to determine if a story is real or a hoax. Our purpose is to use different machine learning models and identify the accurate model to determine if any news is fake or real. If a user comes across any random news site with some misleading content before actually believing it and spreading the news, he can use our model to identify the content. Our motive is to take a step to prevent the casualties occurring due to the amount of fake information widespread in society by delivering a model having the highest accuracy of detecting the legit information.

Implementation details

Web scraping:

We first collect data by web scraping 3 newspaper sites WSJ, NYTimes, and TampaBay for legitimate news content. In order to collect fake news data we used an API and gathered fake news content. The data is collected in a standard format of id, title, publication, author, published date, category, URL, content.

Data merging and Data Preprocessing:

All the data collected is merged using pandas to form a combined dataset under Final_Data.csv. At the time of merging we label the data as "Real" and "Fake". Data is cleaned by removing all the null values, special characters, Unicode characters which are identified by performing exploratory data analysis on the merged data. This cleaned data is stored under a separate CSV named CleanData.

Data Cleansing

- Removed characters from the content
- Updated empty rows with values
- Dropped Unwanted rows
- Removed stop words from content
- Punctuation has been handled

Exploratory Data Analysis:

Exploratory data analysis usually performed to get an overview of the dataset gave us some insights about the data we had after merging. We realized a lot of data needs to be processed as shown in the below snapshot to generate a clean dataset.

	ID	Title	Publication	Author	Published	Year	Month	Category	URL	Content
0	1	What's News: Business & Finance	WSJ	[]	01/02/18	2018	1	\r\r\n Whats News Busines...	https://www.wsj.com/articles/whats-news-busine...	The unemployment rate in some metro areas stan...
1	2	A Browser You've Never Heard of Is Dethroning ...	WSJ	['Newley Purnell', 'Newley.Purnell Wsj.Com']	01/01/18	2018	1	Tech	https://www.wsj.com/articles/a-browser-youve-n...	JAKARTA, Indonesia—A mobile browser rarely use...
2	3	2017 Marked Safest Year in Commercial Aviation...	WSJ	['Andy Pasztor', 'Andy.Pasztor Wsj.Com']	01/02/18	2018	1	U.S.	https://www.wsj.com/articles/2017-marked-safes...	The global airline industry achieved a previou...
3	4	Five Things to Know About the Iranian Protests	WSJ	['Farnaz Fassihi', 'Farnaz.Fassihi Wsj.Com']	12/31/17	2017	12	World	https://www.wsj.com/articles/economics-dissati...	Protests erupted across Iran for a third day o...
4	5	Photos of the Day: Jan. 1	WSJ	[]	01/01/18	2018	1	None	https://www.wsj.com/articles/photos-of-the-day...	Photos of the Day: Jan. 1 Children warm up by ...

We added a feature named 'Label' to denote the fake and real news. The data set looks like below after cleaning

Null Values

Quantity of non-empty rows

In [33]:

```python
#Calcualing percetage of empty rows
def NaN_percent(df, column_name):
    row_count = df[column_name].shape[0]
    empty_values = row_count - df[column_name].count()
    return (100.0*empty_values)/row_count
for i in list(df):
    print(i +': ' + str(NaN_percent(df,i))+'%')
```

```
ID: 0.0%
Title: 24.89328836363304%
Publication: 0.0%
Author: 20.924620910910637%
Published: 0.0%
Year: 0.0%
Month: 0.0%
Category: 0.0%
URL: 0.0%
Content: 0.0%
```

Null Values

In [4]:

```python
#Count of Null values for each columns
df = pd.read_csv('Final_Data.csv')
df
null_columns=df.columns[df.isnull().any()]
df[null_columns].isnull().sum()
```

Out[4]:

```
Title     27235
Author    22893
dtype: int64
```

Null Values Handling

Null Values

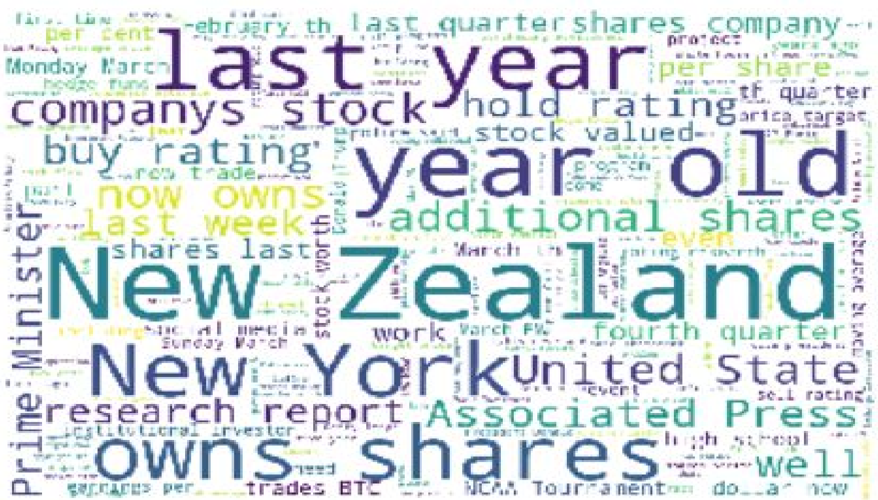
In [10]:

```python
df = pd.read_csv('CleanData.csv')
df
null_columns=df.columns[df.isnull().any()]
df[null_columns].isnull().sum()
```

Out[10]:

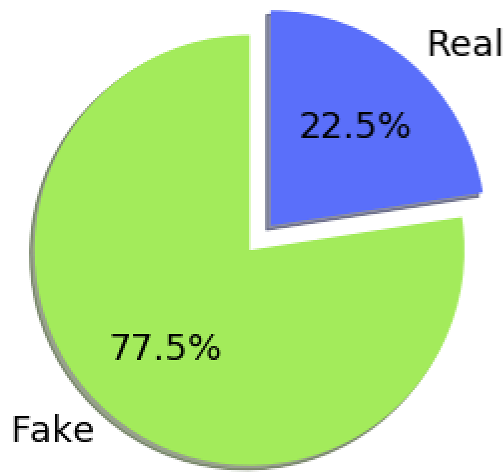
```
Series([], dtype: float64)
```

Fake Data Word Cloud



Real Data Word Cloud

So after data processing, we could come up to the conclusion that our dataset has a collection of real and fake news as shown in the pie chart below



Vectorization:
It is a process of collecting discriminative information from a set of given samples. We used two approaches to extract our features Count Vectorization and TfIdf Vectorization. The training data is transformed to learn vocabulary dictionary and return term-document matrix and the test data is transformed to the document-term matrix for model analysis.

Count vectorizer: It counts the number of times a token shows up in the document and uses this value as its weight.

TF-IDF Vectorizer : TF-IDF stands for “term frequency-inverse document frequency”, meaning the weight assigned to each token not only depends on its frequency in a document but also how recurrent that term is in the entire corpus.

Generating models:
The approach to implementing our idea is pretty simple, our ultimate goal is to classify a given piece of information. To achieve this we tried 5 models for classification and evaluated them to get the best one

- Multinomial Naive Bayes
- Random Forest
- Support Vector Machine
- Logistic Regression
- XGBoost

Analysis of models

Multinomial Naive Bayes:
MNB is a commonly used machine-learning probabilistic classifier which uses a set of features and classes to determine the probability of features occurring in each class and returns the most

likely class in return.The probability is calculated using features generated from count vectorizer and tf-idf vectorizer and labels(fake and real). We train our model on these features and test the model on transformed data. For evaluation, we are computing metrics as follows

Random Forest

Random forests create decision trees on randomly selected data samples, gets a prediction from each tree and selects the best solution by means of voting. The best solution is computed using features generated from count vectorizer and tf-idf vectorizer and labels(fake and real).

XGBoost

Boosting is a sequential technique which works on the principle of an ensemble. It combines a set of weak learners and delivers improved prediction accuracy. At any instant t, the model outcomes are weighed based on the outcomes of previous instant t-1. The outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher. Using features generated from count vectorizer and tf-idf vectorizer and labels(fake and real) for training the model

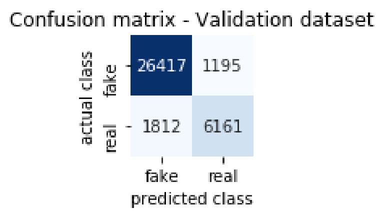
Logistic Regression

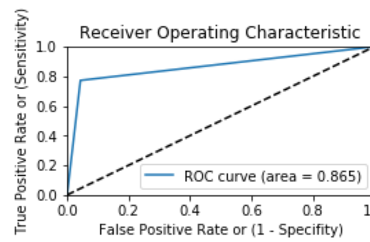
It is the go-to method for binary classification problems (problems with two class values).Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

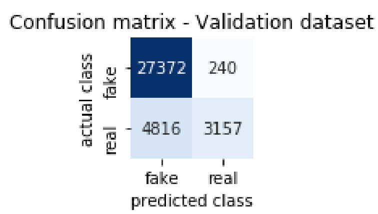
Support Vector Machine

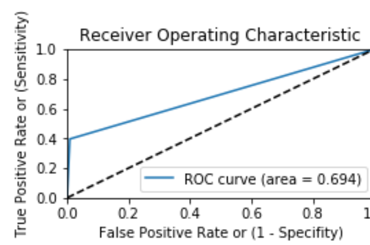
SVM is an algorithm to find a hyperplane in an n-dimensional space which distinctly classifies the data points.The model finds a place that has the maximum margin i.e. maximum distance between the data points of both the classes. The data points falling on either side of the hyperplane can be attributed to different classes Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane.

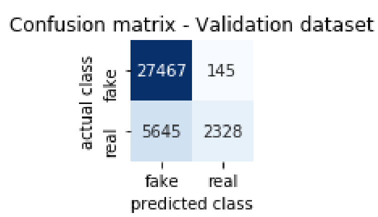
Confusion Matrix and ROC Curve using Count Vectorizer

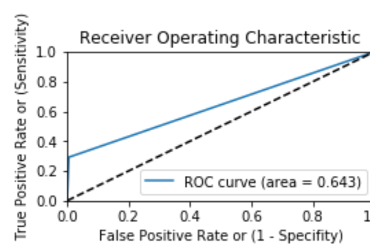
<table>
<tr><th>Models</th><th>Confusion Matrix</th><th>ROC Curve</th></tr>
<tr><td>Multinomial Naive Bayes</td><td>


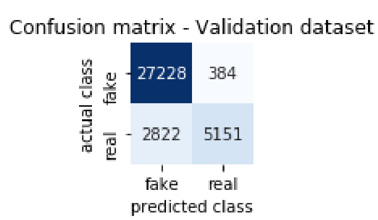
</td><td>


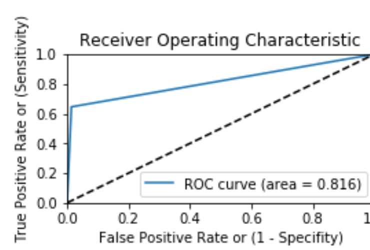
</td></tr>
<tr><td>Random Forest</td><td>


</td><td>


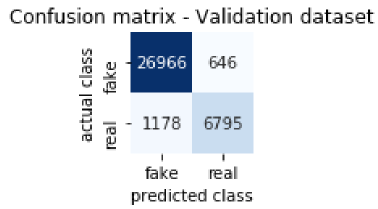
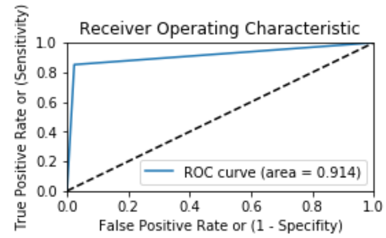
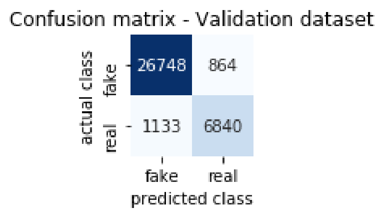
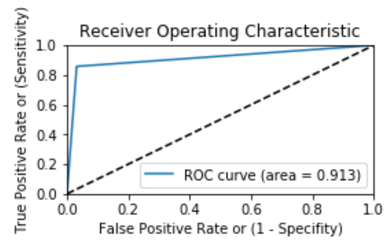
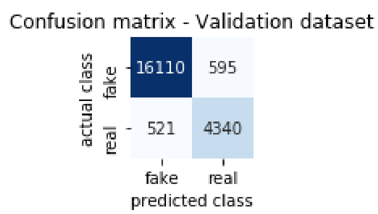
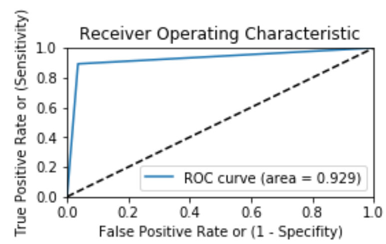
</td></tr>
<tr><td>XGBoost</td><td>


</td><td>


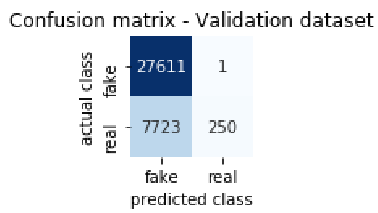
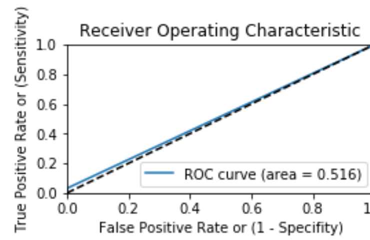
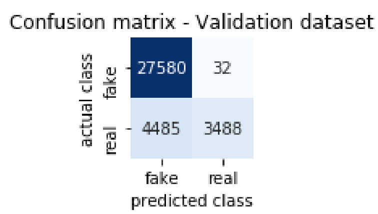
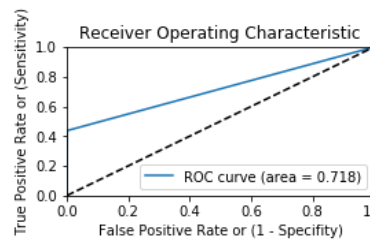
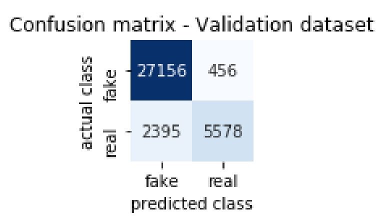
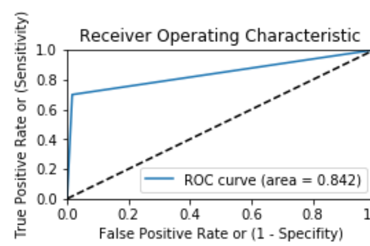
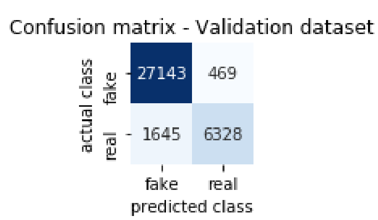
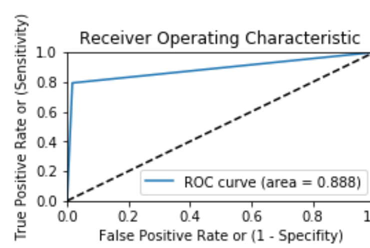
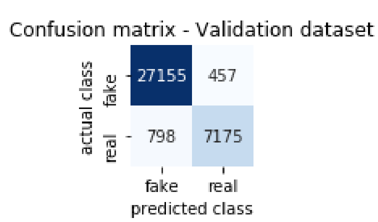
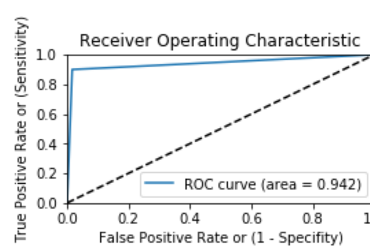
</td></tr>
<tr><td>XGBoost(HPT)</td><td>


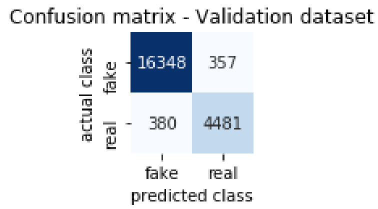
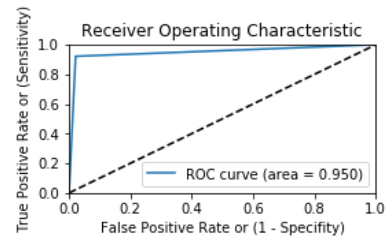
</td><td>


</td></tr>
</table>

<table>
<tr>
<td>Logistic Regression</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>SVM</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>SVM(KFold)</td>
<td>
</td>
<td>
</td>
</tr>
</table>

Confusion Matrix and ROC Curve using TF-IDF Vectorizer

Models	Confusion Matrix	ROC Curve
Multinomial Naive Bayes		
Random Forest		
XGBoost		
Logistic Regression		
SVM		

SVM(KFold)		

Other metrics for models using Count Vectorizer

Models	F1 Score	F1 %	Accuracy	ROC-AUC
Multinomial Naive Bayes	0.946	94.615	0.915	0.864
Random Forest	0.907	90.737	0.842	0.655
XGBoost	0.944	94.440	0.909	0.816
XGBoost(HPT)	0.904	90.465	0.837	0.643
Logistic Regression	0.967	96.728	0.948	0.914
SVM	0.964	96.401	0.943	0.913
SVM(KFold)	**0.966**	**96.652**	**0.948**	**0.928**

Other metrics for models using TF-IDF Vectorizer

Models	F1 Score	F1 %	Accuracy	ROC-AUC
Multinomial Naive Bayes	0.877	87.729	0.782	0.515
Random Forest	0.915	91.545	0.857	0.693
Random Forest(HPT)	0.924	92.430	0.873	0.718
XGBoost	0.950	95.012	0.919	0.841
Logistic Regression	0.962	96.251	0.940	0.888
SVM	0.977	97.741	0.964	0.941
SVM(KFold)	**0.977**	**97.795**	**0.965**	**0.950**

Evaluation Metrics:

We have implemented Random Forest, Multinomial Naive Bayes, Logistic Regression, Support Vector Machine(with 5 fold cross-validation) and XGBoost Models for both Count and TFIDF Vectorization.

Output for Support Vector Machine(with 5 fold cross validation) using TFIDF Vectorization is providing below values.

Confusion Matrix

True Positive - We predicted that 16348 news as fake and they are fake.

False Positive - We predicted 357 news as fake but they are real

False Negative - We predicted 380 news as real but they are fake

True Negative - We predicted 4481 news are real and they are real

Precision - Precision is the ratio of correctly predicted positive observations of the total predicted positive observations. High precision relates to the low false positive rate.

0.9786291529482191

Recall - Recall is the ratio of correctly predicted positive observations to all observations in the actual class

0.9772835963653754

F1 Score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

0.9779559118236474

F1 % - 97.79559118236475

Accuracy - Accuracy (ACC) is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.0, whereas the worst is 0.0. It can also be calculated by 1 – ERR.

0.965825836965594

Error Rate - 0.034174163034406006

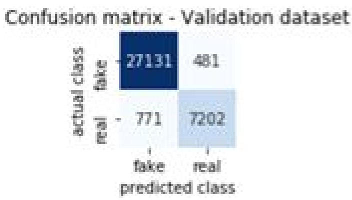
ROC-AUC - ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.

0.9502279687802193

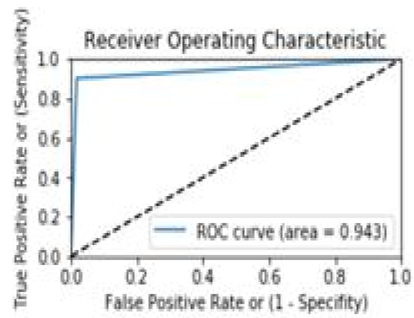
Which is higher than any other values, and we are using the SVM model for our final prediction.

Pipeline Design

Luigi is a Python tool for workflow management. It has been developed at Spotify, to help building complex data pipelines of batch jobs
There are two core concepts to understand how we can apply Luigi to our own data pipeline: Tasks and Targets. A task is a unit of work, designed by extending the class luigi.Task and overriding some basic methods. The output of a task is a target, which can be a file on the local filesystem, a file on Amazon's S3, some piece of data in a database etc.
Dependencies are defined in terms of inputs and outputs, i.e. if TaskB depends on TaskA, it means that the output of TaskA will be the input of TaskB.
Our code does the Data Merging for all the newspapers csv's that we have and the MergeAllDataSingleFile() depends on four tasks for its input and to merge all of them into a single csv file. Later on the DataCleaning() depends on MergeAllDataSingle() and cleans the csv that is provided by it and stores into a csv. This acts as an input to our Model() where the metrics are calculated providing the accuracy of the model used for classification




```
Precision =  0.9825800376647834
Recall =  0.9723675722170454
F1 Score =  0.9774471304535793
F1 % =  97.74471304535794
Error Rate =  0.03518336377687228
Accuracy =  0.9648166362231277
ROC Curve:
```




```
ROC-AUC 0.9429393352753868
```

Details on how to run the Model

```
tfidf_vectorizer = TfidfVectorizer(stop_words='english',
                                   encoding='utf-8',
                                   decode_error='replace',
                                   strip_accents='unicode',
                                   analyzer='word',
                                   tokenizer=porter_tokenizer,
                                   ngram_range=(1,2),
                                   binary=False)

#testing the implementation
tfcon = df['Content'].loc[1]
tfcon = [tfcon]

tfvect = tfidf_vectorizer.fit(tfcon)
print(tfvect)

# Fit and transform the training data
tfidf_train = tfidf_vectorizer.fit_transform(X_train)

# Transform the test set
tfidf_test = tfidf_vectorizer.transform(X_test)
```

SVM Model using KFold Cross Validation

```python
#K fold Cross Validation

#Defining the split into 5 Fold
from sklearn.model_selection import StratifiedKFold

le = LabelEncoder()
le.fit(df['Label'])

df_labels = pd.DataFrame(np.array(le.transform(df['Label'])))

skf = StratifiedKFold(n_splits = 5)

for trn_indx, tst_indx in skf.split(df['Content'],df_labels):
    skf_X_train, skf_X_test = df['Content'].iloc[trn_indx], df['Content'].iloc[tst_indx]
    skf_Y_train, skf_Y_test = df_labels.iloc[trn_indx], df_labels.iloc[tst_indx]
```

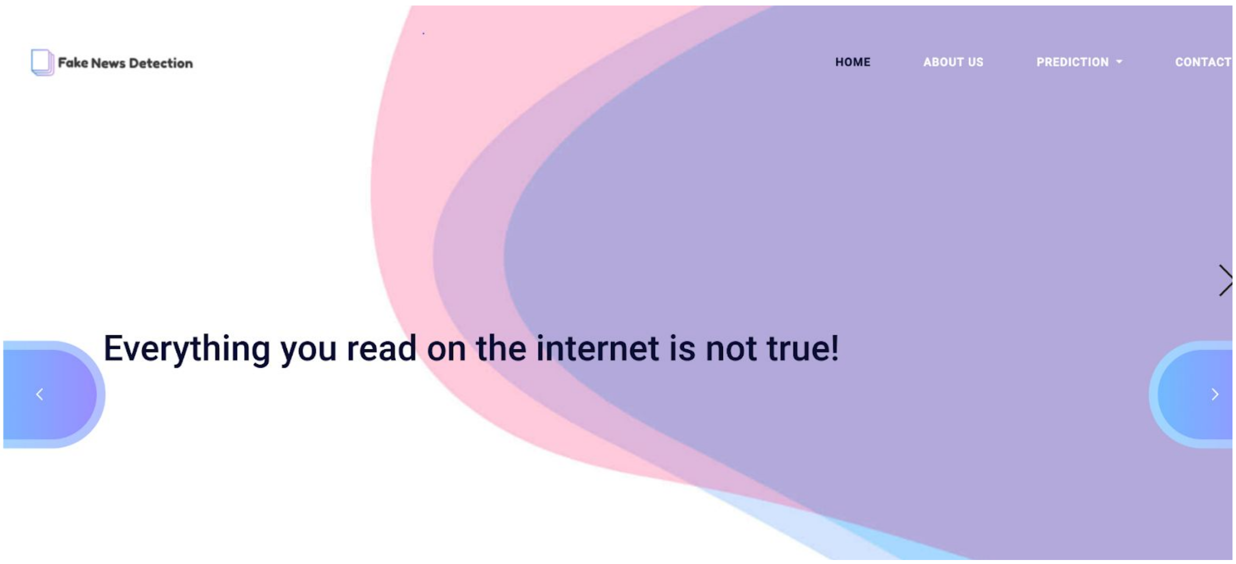
```python
# Fit and transform the training data for count vector
skf_count_train = count_vectorizer.fit_transform(skf_X_train)

# Transform the test set
skf_count_test = count_vectorizer.transform(skf_X_test)

# Fit and transform the training data for tfidf
skf_tfidf_train = tfidf_vectorizer.fit_transform(skf_X_train)

# Transform the test set
skf_tfidf_test = tfidf_vectorizer.transform(skf_X_test)
```

```python
# print("TfIdf Vectorization")
skf_clf2 = svm.LinearSVC()

skf_clf2.fit(skf_tfidf_train, skf_Y_train)                  # Fit SVM classifier according to X, y

skf_predtf = skf_clf2.predict(skf_tfidf_test)                # Perform classification on an array of test vectors X

skf_tf_roc_auc = roc_auc_score(skf_Y_test, skf_predtf, average= 'micro')

#confusion matrix for count vector
skf_tf_cm = metrics.confusion_matrix(skf_Y_test, skf_predtf, labels=[0,1])

#plot the confusion Matrix
plotConfusionMatrix(skf_tf_cm)

rocCurve(skf_Y_test, skf_predtf, skf_tf_roc_auc)

#Evaluation
evaluation(skf_tf_cm)

print("ROC-AUC", skf_tf_roc_auc)
```

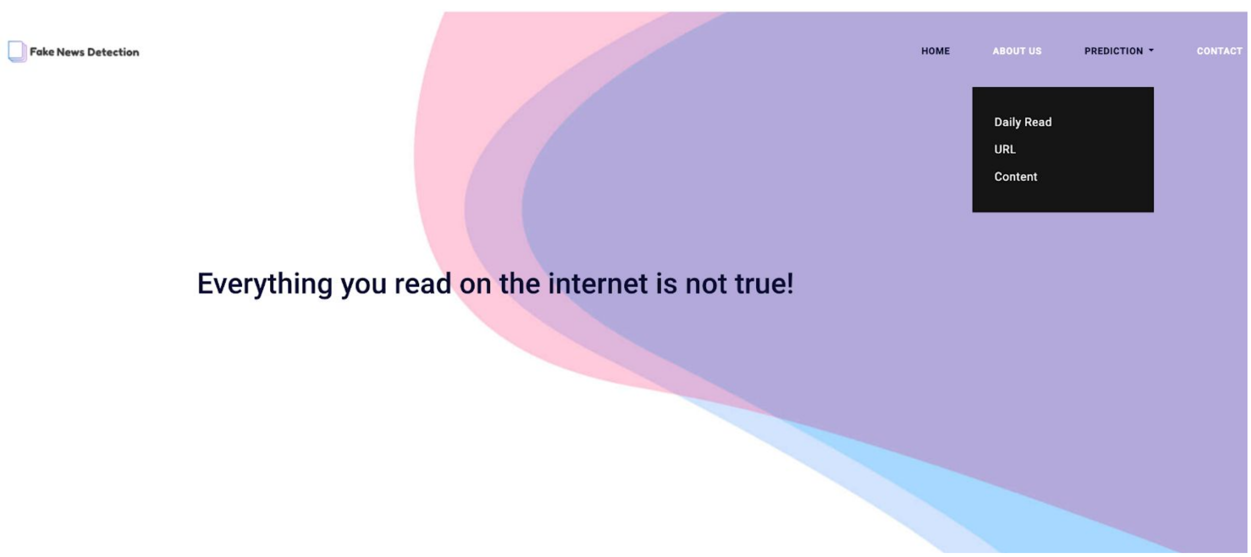
Details on how to run the Website

Docker Image - docker run -p 5000:5000 sharavan27/ads_fake_news_final1
Demo link - https://youtu.be/abbaa6n0Mpg

Step 1: Click on the website link.

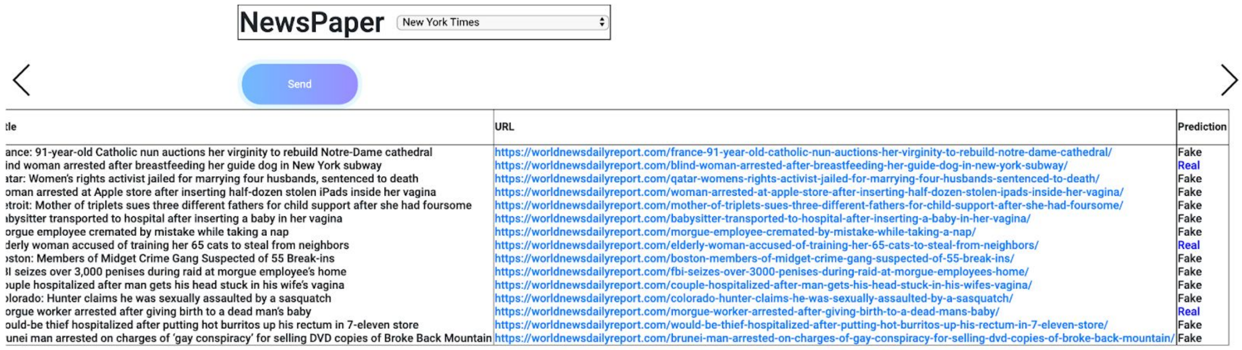


Step 2: Website is for predicting the news. Click on prediction drop down, three options are given.



Step 3: Select “Daily Read”.This option will take you to the drop down of New papers list.

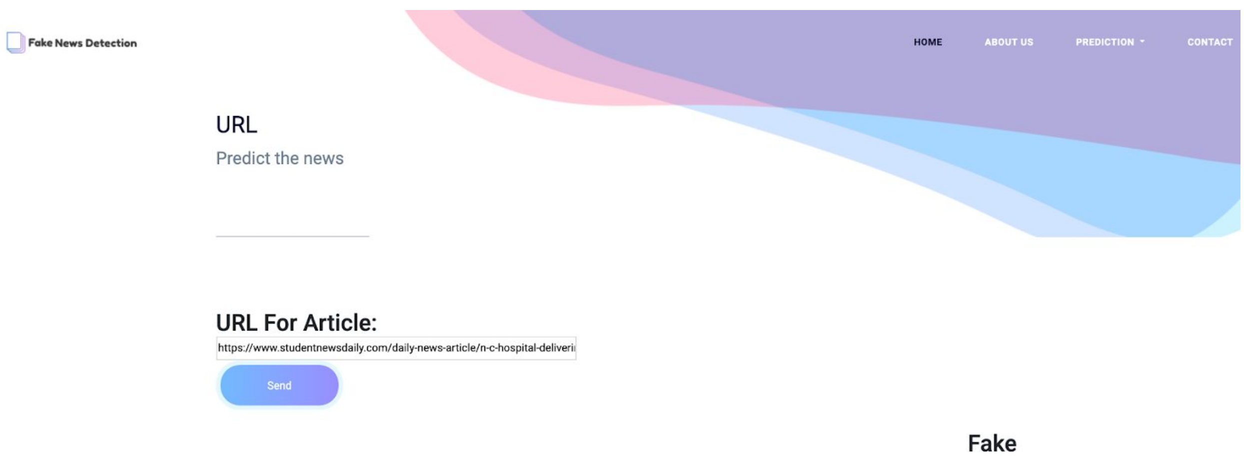
Select any one of Newspaper name.



tle	URL	Prediction
ance: 91-year-old Catholic nun auctions her virginity to rebuild Notre-Dame cathedral	https://worldnewsdailyreport.com/france-91-year-old-catholic-nun-auctions-her-virginity-to-rebuild-notre-dame-cathedral/	Fake
ind woman arrested after breastfeeding her guide dog in New York subway	https://worldnewsdailyreport.com/blind-woman-arrested-after-breastfeeding-her-guide-dog-in-new-york-subway/	Real
ıtar: Women's rights activist jailed for marrying four husbands, sentenced to death	https://worldnewsdailyreport.com/qatar-womens-rights-activist-jailed-for-marrying-four-husbands-sentenced-to-death/	Fake
oman arrested at Apple store after inserting half-dozen stolen iPads inside her vagina	https://worldnewsdailyreport.com/woman-arrested-at-apple-store-after-inserting-half-dozen-stolen-ipads-inside-her-vagina/	Fake
etroit: Mother of triplets sues three different fathers for child support after she had foursome	https://worldnewsdailyreport.com/mother-of-triplets-sues-three-different-fathers-for-child-support-after-she-had-foursome/	Fake
ıbysitter transported to hospital after inserting a baby in her vagina	https://worldnewsdailyreport.com/babysitter-transported-to-hospital-after-inserting-a-baby-in-her-vagina/	Fake
orgue employee cremated by mistake while taking a nap	https://worldnewsdailyreport.com/morgue-employee-cremated-by-mistake-while-taking-a-nap/	Fake
derly woman accused of training her 65 cats to steal from neighbors	https://worldnewsdailyreport.com/elderly-woman-accused-of-training-her-65-cats-to-steal-from-neighbors/	Real
›ston: Members of Midget Crime Gang Suspected of 55 Break-ins	https://worldnewsdailyreport.com/boston-members-of-midget-crime-gang-suspected-of-55-break-ins/	Fake
3I seizes over 3,000 penises during raid at morgue employee's home	https://worldnewsdailyreport.com/fbi-seizes-over-3000-penises-during-raid-at-morgue-employees-home/	Fake
›uple hospitalized after man gets his head stuck in his wife's vagina	https://worldnewsdailyreport.com/couple-hospitalized-after-man-gets-his-head-stuck-in-his-wifes-vagina/	Fake
›lorado: Hunter claims he was sexually assaulted by a sasquatch	https://worldnewsdailyreport.com/colorado-hunter-claims-he-was-sexually-assaulted-by-a-sasquatch/	Fake
orgue worker arrested after giving birth to a dead man's baby	https://worldnewsdailyreport.com/morgue-worker-arrested-after-giving-birth-to-a-dead-mans-baby/	Real
ould-be thief hospitalized after putting hot burritos up his rectum in 7-eleven store	https://worldnewsdailyreport.com/would-be-thief-hospitalized-after-putting-hot-burritos-up-his-rectum-in-7-eleven-store/	Fake
unei man arrested on charges of 'gay conspiracy' for selling DVD copies of Broke Back Mountain	https://worldnewsdailyreport.com/brunei-man-arrested-on-charges-of-gay-conspiracy-for-selling-dvd-copies-of-broke-back-mountain/	Fake

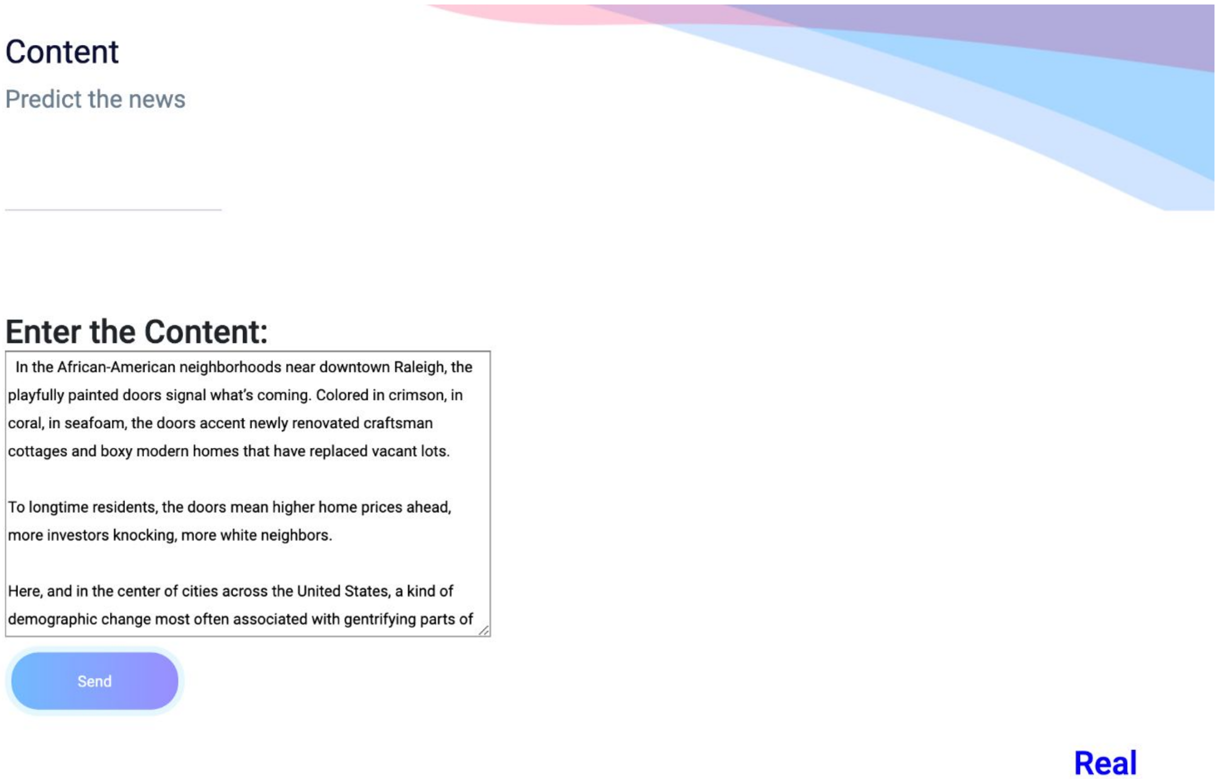
It will provide a list of articles of the websites and whether they are Fake or Real along with URL list of the article.

Step 4: Again go back to Prediction, dropdown. Select the URL option.In the given textbox, provide URL of Newspaper articles, which you want to check whether legitimate or not.



This will provide whether the article's data is real or not.

Step 5: From the dropdown, select Content option.On the page, in the given text box provide content from any news and check whether is it Fake or Real.



Test Cases

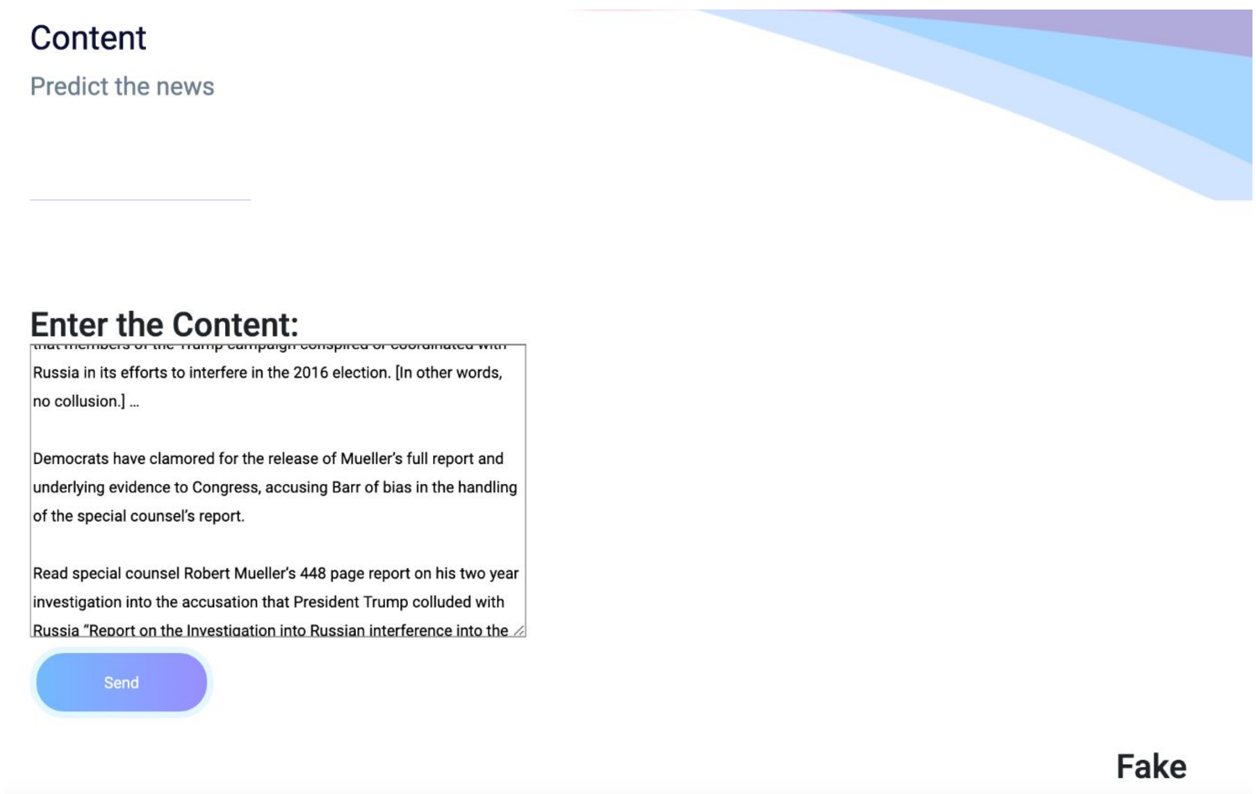
A reader is browsing through some site which has some hot news and wants to know whether the information is legitimate or not.

Input - From dropdown select Huffpost(https://www.huffpost.com/)
Output - List of articles with Label of Real or Fake

A reader has read news content and wants to know if the content is fake or real

Input - Provide content to the textbox

Output - User will get the Label of the content.



A reader has come across a news article which he finds suspicious and can be unreal

Input - Provide URL of the article.
Output - User will get the Label of the content.

Citations

https://github.com/rasbt/musicmood/blob/master/code/classify_lyrics/nb_init_model.ipynb

https://github.com/rockash/Fake-news-Detection/blob/master/getEmbeddings.py

https://github.com/rastogi-s/Fake-News-Detection/blob/master/ModelRun/Driver.py

https://github.com/tommartensen/fake-news-detector/tree/master/feature_generation

https://github.com/bedarkarpriyanka/NLP-Project-Fake-News-Detection/blob/master/NLPproject_6.ipynb

https://github.com/nishitpatel01/Fake_News_Detection

https://github.com/ajayjindal/Fake-News-Detection/blob/master/final.py

https://www.kaggle.com/mrisdal/fake-news

https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184

https://github.com/chbrown/liwc-python