Fake News detection on Twitter

-KARAN MANKAR

Introduction

- Social media is a big part of our daily life.
- Heavy reliance on microblogging sites
  - Present our views or opinions.
  - Follow influential personalities.
- Problem of fake news to shape the public discourse is of growing concern
  - Exaggerated claims over the Hillary emails.
  - NHS bus that was roaming the streets of London (heavily publicized by the pro-Brexit camp using Twitter).

Business Understanding

- Information quality over social media is of increasing concern as our reliance on the web platforms grows.
- Widely used to undertake crucial tasks such as coordinating relief effort in times of disaster.
  - Influencers coordinating medicine and oxygen information in India during the recent Second wave.
- Fake news during such trying times can lead to irreparable losses and so detecting Fake news before being widely disseminated is of paramount importance.
- A system to detect fake news is presented in this project
  - The system will analyze:
    - Tweet content
    - Retweet data
    - Tweet source information
    - Sentiment of the tweet.

Literature Review

1. **FakeBERT: Fake news detection in social media with a BERT-based deep learning approach**
   - The FakeBERT system uses Deep Learning techniques like:
     - LSTM
     - Convolution Neural Networks
     - BERT transformer
   - Accuracy gain of approximately 90% using data acknowledged as fake news and from reputable tabloids like the New York Times and Washington Post.

2. **Fake News Detection Using Machine Learning Ensemble Methods**
   - Highlights the grave issue of lack of enough reliable open-sourced data
   - Has created a dataset MisInfoText.
   - Contribution for collecting data from various fact checking websites required.

Literature Review

3. **Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM)**
   - AI-powered analytic tools such as stance-classification
   - Determine whether the headline of the news matches the body
   - Text processing to analyze the author’s writing style
   - Image forensics to detect photoshop use.
   - Real time anomaly detection to detect anomalies in text indicating a pre-determined text that the owner of the handle just posted after being told to.

Data Understanding

- Datasets used:
  - Politifact Dataset
  - Pheme RumourDataset

- **Politifact Dataset:**
  - Consists of tweets, urls and Retweet ids
  - Divided into two folders
    - Real
    - Fake

```
df.head()
```

id	news_url	title	tweet_ids
politifact15014	speedtalk.com/forum/viewtopic.php?t=51650	BREAKING: First NFL Team Declares Bankruptcy O...	937349434668498944\t937379378006282240\t937380...
politifact15156	politics2020.info/index.php/2018/03/13/court-o...	Court Orders Obama To Pay $400 Million In Rest...	972666281441878016\t972678396575559680\t972827...
politifact14745	www.nscdscamps.org/blog/category/parenting/467...	UPDATE: Second Roy Moore Accuser Works For Mic...	929405740732870656\t929439450400264192\t929439...
politifact14355	https://howafrica.com/oscar-pistorius-attempts...	Oscar Pistorius Attempts To Commit Suicide	886941526458347521\t887011300278194176\t887023...
politifact15371	http://washingtonsources.org/trump-votes-for-d...	Trump Votes For Death Penalty For Being Gay	915205698212040704\t915242076681506816\t915249...

Data Understanding

Pheme Rumour Dataset:

Dataset Structure

The downloaded dataset will have the following folder structure,

```bash
├── charliehebdo
│   ├── non-rumours
│   │   ├── 552785249420447745
│   │   │   ├── reactions
│   │   │   │   ├──552785249420447745.json
│   │   │   │   └── ....
│   │   │   └── source-tweet
│   │   │       └── 552785249420447745.json
│   │   └── ....
│   │
│   └── rumours
│       ├── 552783238415265792
│       │   ├── reactions
│       │   │   ├──552787794503143424.json
│       │   │   └── ....
│       │   └── source-tweet
│       │       └── 552783238415265792.json
│       └── ....
├── ferguson
│   ├── non-rumours
│   │   ├── 498235547685756928
│   │   │   ├── reactions
│   │   │   │   ├──498243332204949504.json
│   │   │   │   └── ....
│   │   │   └── source-tweet
│   │   │       └── 498235547685756928.json
│   │   └── ....
│   │
│   └── rumours
│       ├── 552783238415265792
│       │   ├── reactions
│       │   │   ├──552787794503143424.json
│       │   │   └── ....
│       │   └── source-tweet
│       │       └── 552783238415265792.json
│       └── ....
├── germanwings-crash
│   ├── non-rumours
│   │   ├── 498235547685756928
```

Data Preparation

- Data preparation was done separately for both the datasets.
- Python Script for Pheme Rumour Dataset
- Sentiment Analysis using VaderSentiment Library
- Removal of stopwords, URLs, emojis, @-mentions, hashtags from tweets
- Conversion of tweet_ids to number of retweets

```
df.head()
```

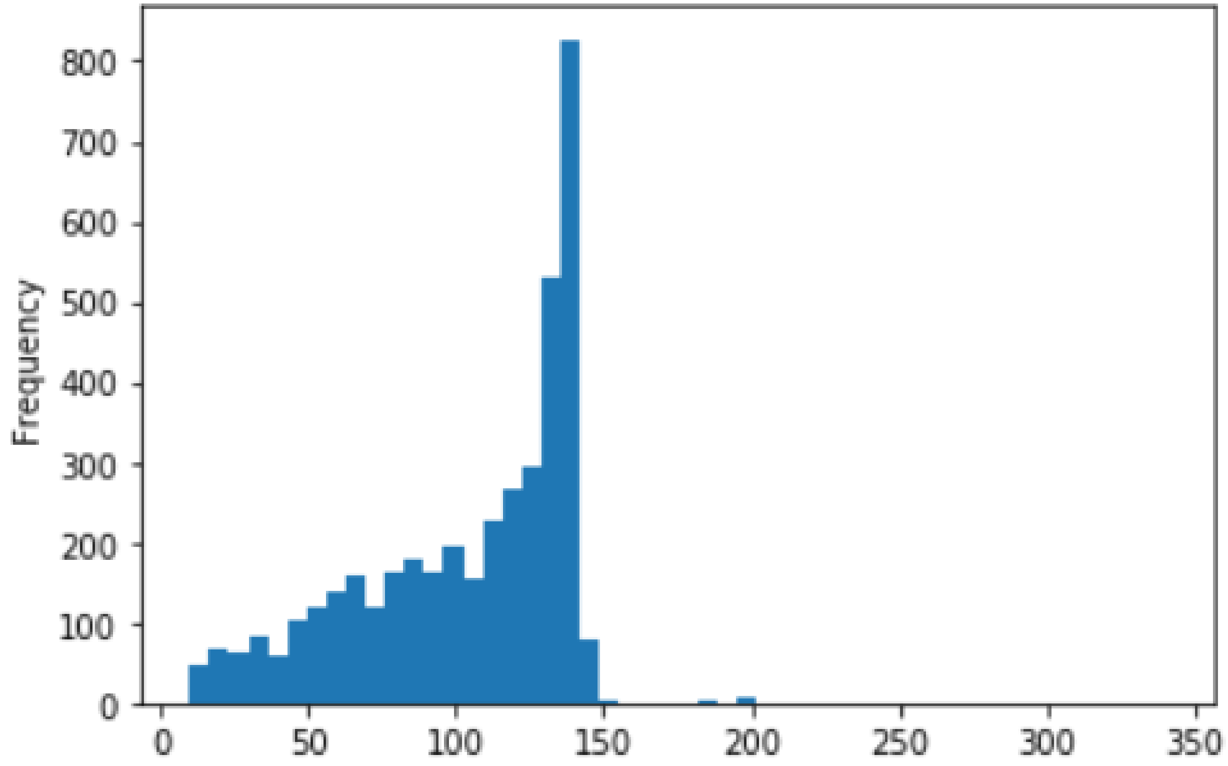
	news_url	title	no_of_retweets	label	length
0	speedtalk.com	BREAKING: First NFL Team Declares Bankruptcy O...	163	1	64
1	politics2020.info	Court Orders Obama To Pay $400 Million In Rest...	102	1	53
2	nscdscamps.org	UPDATE: Second Roy Moore Accuser Works For Mic...	220	1	69
3	howafrica.com	Oscar Pistorius Attempts To Commit Suicide	22	1	42
4	washingtonsources.org	Trump Votes For Death Penalty For Being Gay	550	1	43

Exploratory Data Analysis

Length of all the tweets

```
df['length'].plot(bins=50, kind='hist')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21dcb125788>
```

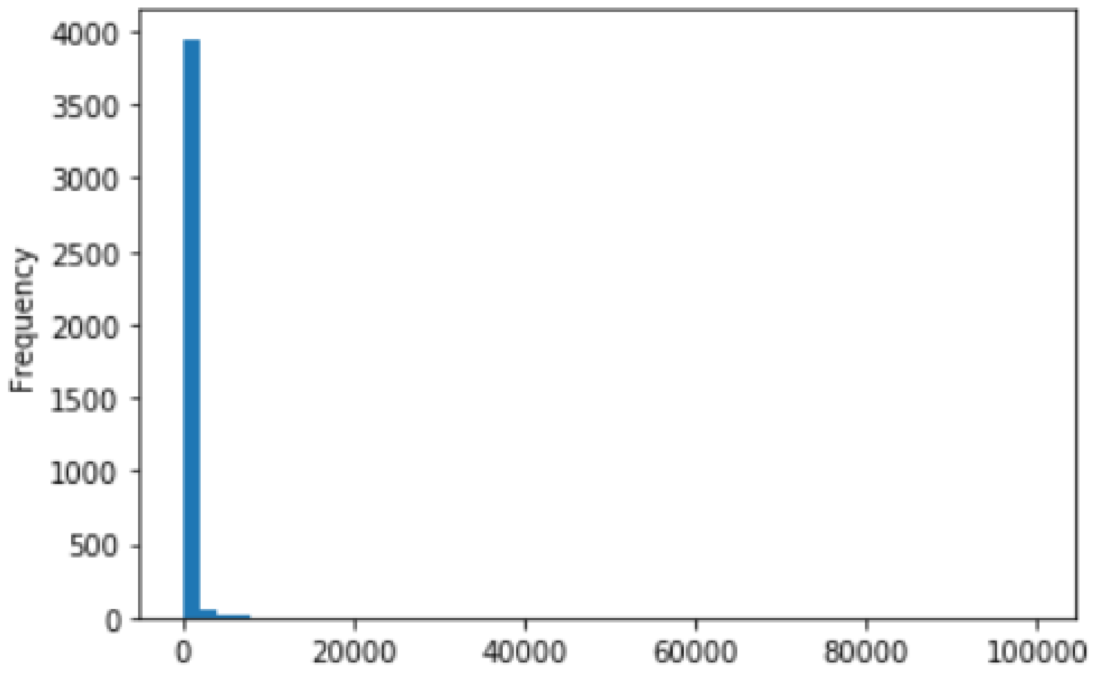


Exploratory Data Analysis

Total number of retweets per news article

```
df['no_of_retweets'].plot(bins=50, kind='hist')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21dcc4cd688>
```

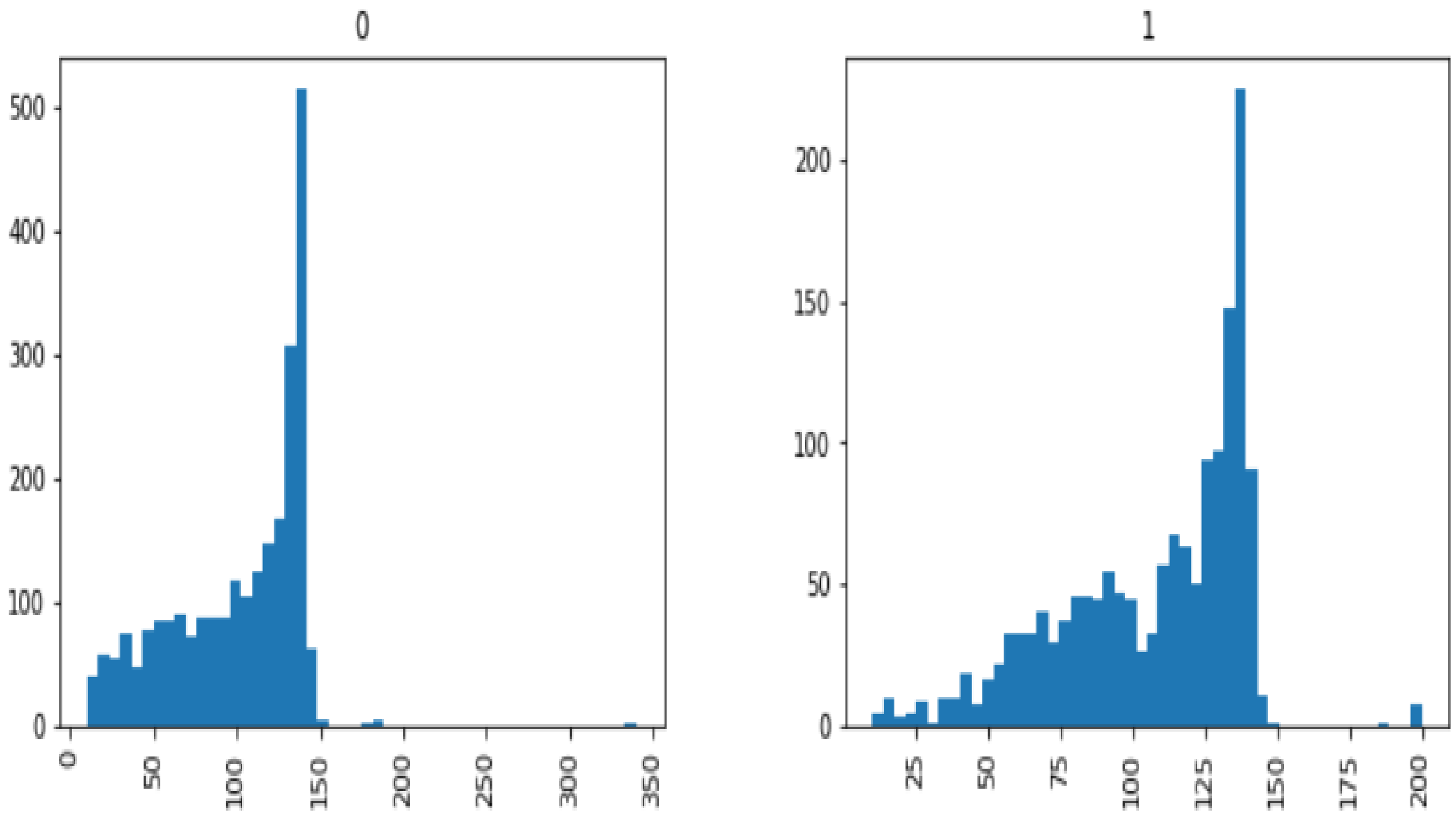


Exploratory Data Analysis

Length vs Label

```
df.hist(column='length', by='label', bins=50,figsize=(12,4))
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x0000021DCC6AC448>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x0000021DCC6E4748>],
      dtype=object)
```

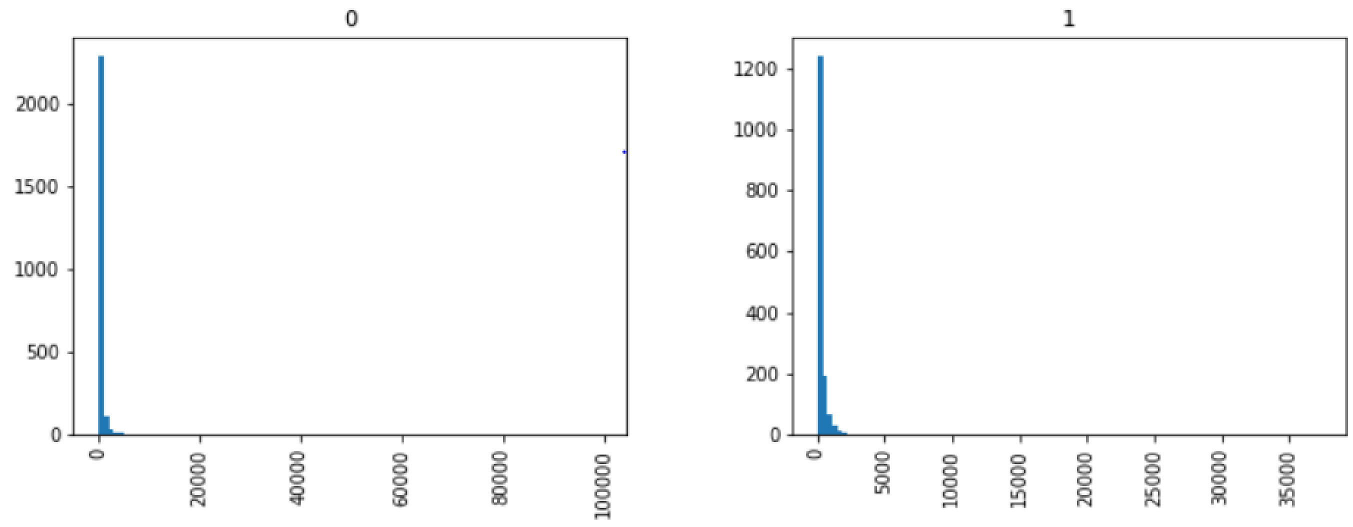


Exploratory Data Analysis

Number of retweets vs label

```
df.hist(column='no_of_retweets', by='label', bins=100,figsize=(12,4))
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x0000021DCD24DC48>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x0000021DCD20C048>],
      dtype=object)
```

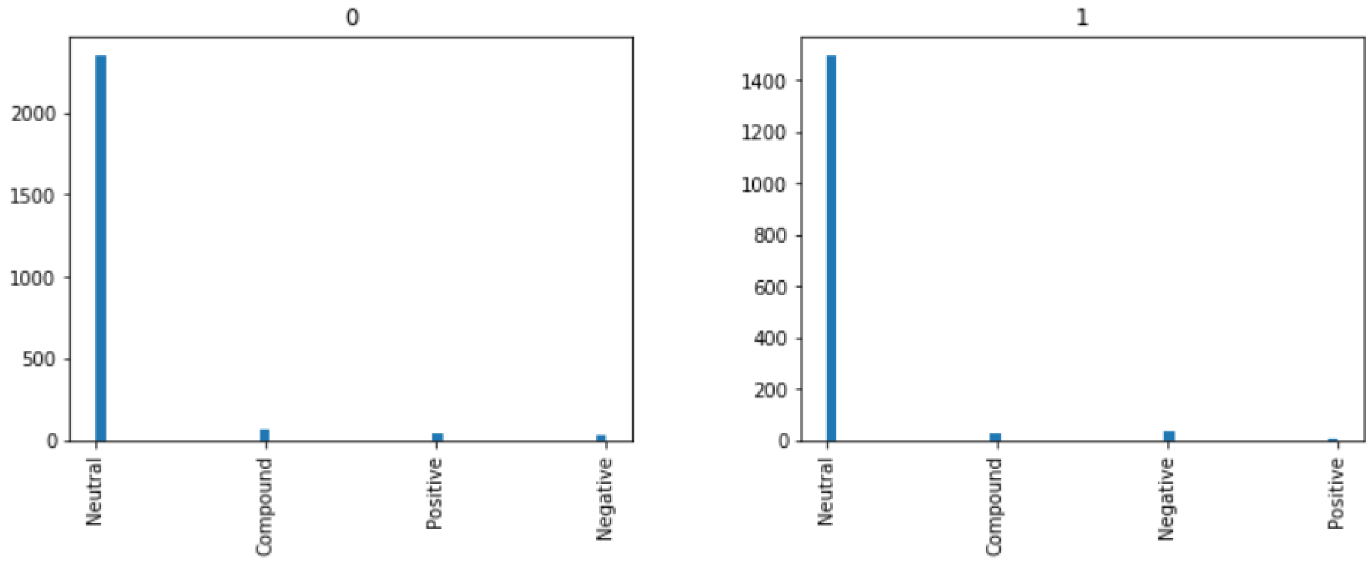


Exploratory Data Analysis

Sentiment vs Label

```
df.hist(column='sentiment', by='label', bins=50,figsize=(12,4))
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x0000021DCCB31BC8>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x0000021DCCB9B3C8>],
      dtype=object)
```



Vectorization

- Bag of words transformation
- Convert raw text data into vectors for modelling
- Sparse Matrix of both tweet text and source URL
- TF-IDF vectorization on tweet text and source URL

```
print('Shape of Sparse Matrix: ', tweets_bow.shape)
print('Amount of Non-Zero occurences: ', tweets_bow.nnz)
```

```
Shape of Sparse Matrix:  (4057, 9479)
Amount of Non-Zero occurences:  37214
```

```
print('Shape of Sparse Matrix: ', news_url_bow.shape)
print('Amount of Non-Zero occurences: ', news_url_bow.nnz)
```

```
Shape of Sparse Matrix:  (4057, 1356)
Amount of Non-Zero occurences:  4057
```

Modeling

Three models used:

- Naïve Bayes
- Support Vector Machines
- K Means Clustering

Naïve Bayes

	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.42	0.47	0.48	0.40

```
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
```

```
[[148 584]
 [117 369]]
              precision    recall  f1-score   support

           0       0.56      0.20      0.30       732
           1       0.39      0.76      0.51       486

    accuracy                           0.42      1218
   macro avg       0.47      0.48      0.40      1218
weighted avg       0.49      0.42      0.38      1218
```

Support Vector Machines

	Accuracy	Precision	Recall	F1-Score
SVM	0.60	0.60	0.51	0.39

```
print(confusion_matrix(y_test,predictions))
print(classification_report(y_test, predictions))
```

```
[[726   6]
 [477   9]]
              precision    recall  f1-score   support

           0       0.60      0.99      0.75       732
           1       0.60      0.02      0.04       486

    accuracy                           0.60      1218
   macro avg       0.60      0.51      0.39      1218
weighted avg       0.60      0.60      0.47      1218
```

K Means Clustering

	Accuracy	Precision	Recall	F1-Score
K Means	0.60	0.30	0.50	0.38

```
print(confusion_matrix(y_test,predictions))
print(classification_report(y_test, predictions))
```

```
[[732   0]
 [486   0]]
              precision    recall  f1-score   support

           0       0.60      1.00      0.75       732
           1       0.00      0.00      0.00       486

    accuracy                           0.60      1218
   macro avg       0.30      0.50      0.38      1218
weighted avg       0.36      0.60      0.45      1218
```

Evaluation

	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.42	0.47	0.48	0.40
SVM	0.60	0.60	0.51	0.39
K Means	0.60	0.30	0.50	0.38

The End