Université Cadi Ayyad

Faculté des Sciences Semlalia - Marrakech

Report:

Models for Fake News Classification

Master science de données
NLP and Text Mining

Created by:

El Arachi Mounia

Maghrane Wail

Requested by:
Dr. ZAHIR JIHAD
Professor at the Faculty of Sciences
Semlalia, Marrakech (FSSM)

Contents

Business Understanding

The goal of our project

Our project aims to develop a comprehensive system for detecting fake news with a particular focus on events related to Palestine and Israel, by integrating multiple forms of media, including text and images. We employ advanced machine learning techniques for accurate and real-time analysis across various platforms, enhancing detection precision and reliability. Additionally, our project focuses on user education for better awareness of fake news and addresses ethical considerations and privacy concerns in the process. The ultimate goal is to create a more informed and truthful digital environment, effectively mitigating the spread of misinformation.

Data Preparation

Data Collection (Scraping)

In the course of our project, we have implemented a comprehensive data collection strategy by scraping data from the internet. Specifically, we have successfully gathered a substantial dataset of 2,000 articles, sourced from a diverse range of websites. This dataset includes articles from notable sites such as The Monitor, Israel Today, and Intifada, ensuring a broad spectrum of perspectives and content styles. This diverse collection of data forms the foundation of our analysis, enabling our system to learn from a wide array of news sources for effective fake news detection.

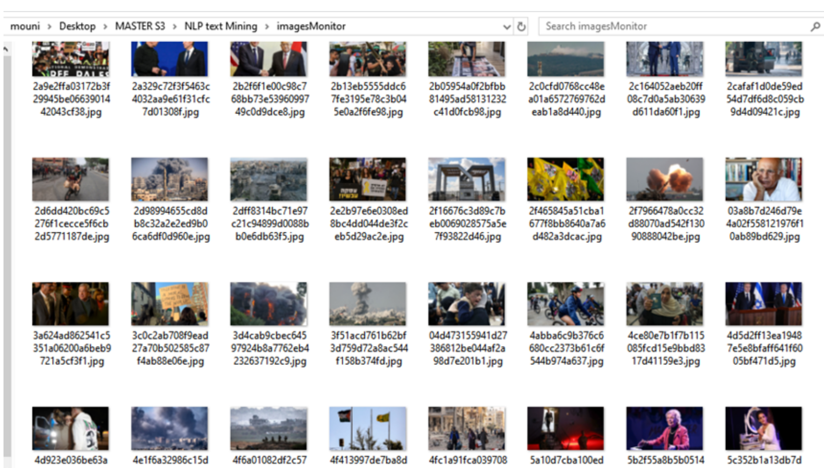
In our web scraping process, our primary objective is to extract specific information from each article, ensuring we capture the following key data points:

1. Article ID
2. URL
3. Title
4. Date
5. Content

6. Summary
7. Source
8. Caption
9. Image URL
10. Image Path
11. Class (FAKE or REEL)

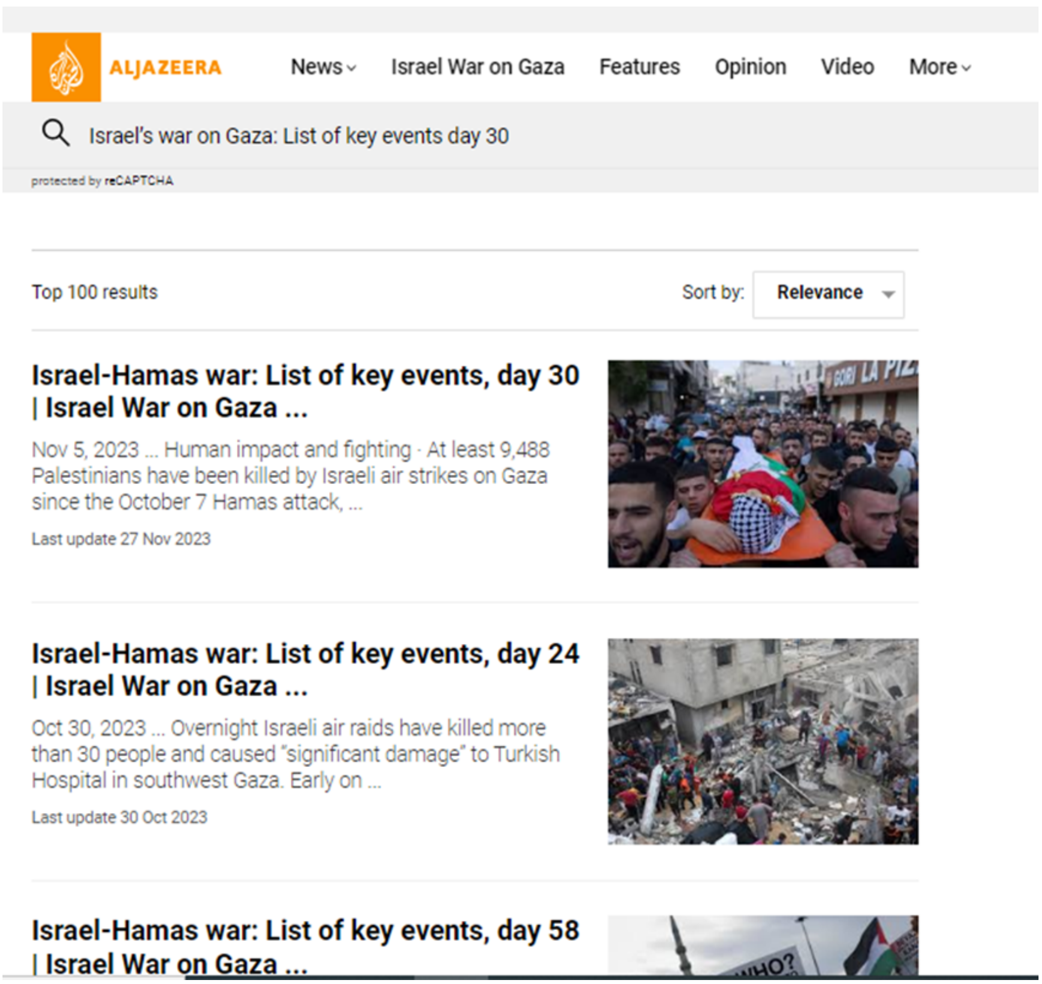
id	url	title	date	content	sum	source	caption	image_url	image_path	Class
art_1	https://www.timesofi	Visiting Isra	18 Decemb	US Defens	Aus	Time of	In this govern	https://static.	timesofisrael\IMG-20231	TRUE
art_4	https://www.timesofi	Hamas rele	18 Decemb	The Hama	Cha	Time of	Screen captu	https://static.	timesofisrael\Elderly-Hos	FAKE
art_8	https://www.timesofi	President H	18 Decemb	President	The	Time of	A home in As	https://static.	timesofisrael\F231007OI	TRUE
art_10	https://www.timesofi	Amid report	18 Decemb	Finance M	Harc	Time of	Finance minis	https://static.	timesofisrael\F231218C(	TRUE
art_12	https://www.timesofi	South Africa	18 Decemb	JOHANNE	Pret	Time of	Four IDF sold	https://static.	timesofisrael\3soldiersG	TRUE
art_14	https://www.timesofi	'You wouldr	18 Decemb	Yotam Ha	Hea	Time of	Iris, Raviv anc	https://static.	timesofisrael\image-43-e	TRUE
art_15	https://www.timesofi	Twenty rele	18 Decemb	During a s	Soc	Time of	Illustrative: Pa	https://static.	timesofisrael\schoolyard	FAKE
art_16	https://www.timesofi	IDF jets hit	18 Decemb	Israeli war	Roc	Time of	A picture take	https://static.	timesofisrael\34839LX-hi	TRUE
art_17	https://www.timesofi	Mossad chi	18 Decemb	Mossad c	Ren	Time of	Mossad chief	https://static.	timesofisrael\F221212A\	TRUE
art_18	https://www.timesofi	Oil giant BF	18 Decemb	Oil giant E	Dec	Time of	The USS Can	https://static.	timesofisrael\AP233407!	TRUE
art_19	https://www.timesofi	Freed hosta	18 Decemb	Yarden Rc	Yarc	Time of	Yarden Roma	https://static.	timesofisrael\Screensho	FAKE
art_20	https://www.timesofi	Jails runnin	18 Decemb	The drama	Pris	Time of	A prison guar	https://static.	timesofisrael\AP212492!	FAKE
art_21	https://www.timesofi	Woman inju	18 Decemb	A 27-year-	Six-	Time of	An Israeli veh	https://static.	timesofisrael\WhatsApp-	FAKE
art_26	https://www.timesofi	Israel-linkec	18 Decemb	A hacking	Tehi	Time of	People wait a	https://static.	timesofisrael\34833FQ-h	TRUE
art_27	https://www.timesofi	Human Rigl	18 Decemb	Human Ri	In re	Time of	Workers and	https://static.	timesofisrael\347E8U3-h	TRUE
art_29	https://www.timesofi	Unsatisfied	18 Decemb	Prime Min	PM	Time of	Prime Ministe	https://static.	timesofisrael\F231217M.	TRUE
art_30	https://www.timesofi	'Financially	18 Decemb	Hamas's r	A cc	Time of	Demonstrator	https://static.	timesofisrael\347Z23L-hi	FAKE
art_31	https://www.timesofi	Slain hosta	18 Decemb	The father	Fatl	Time of	Relatives and	https://static.	timesofisrael\34839M6-h	TRUE
art_32	https://www.timesofi	Citing Gaza	18 Decemb	Citing Isra	Criti	Time of	A choir direct	https://static.	timesofisrael\rias-640x4(	TRUE
art_34	https://www.timesofi	Israel may l	18 Decemb	Israel and	Mos	Time of	A neon sign r	https://static.	timesofisrael\347F633-hi	FAKE
art_35	https://www.timesofi	IDF refutes	18 Decemb	The Israel	Arm	Time of	Nahida Khalil	https://static.	timesofisrael\75251-640x	FAKE
art_36	https://www.timesofi	Houthi attac	18 Decemb	As Yemer	Due	Time of	Iran-backed H	https://static.	timesofisrael\Screensho	FAKE

- Scraped images:



Data Preparation (Annotation)

In our annotation process, we rely heavily on key events and data points extracted from the Al Jazeera website, which we consider to be our trusted and reliable source. Our methodology involves a meticulous examination of both the provided information and the publication date. We then cross-reference this information with the events to assess their authenticity. By comparing the extracted data with the corresponding dates and events, we make informed judgments to determine whether the events are genuine or fabricated.

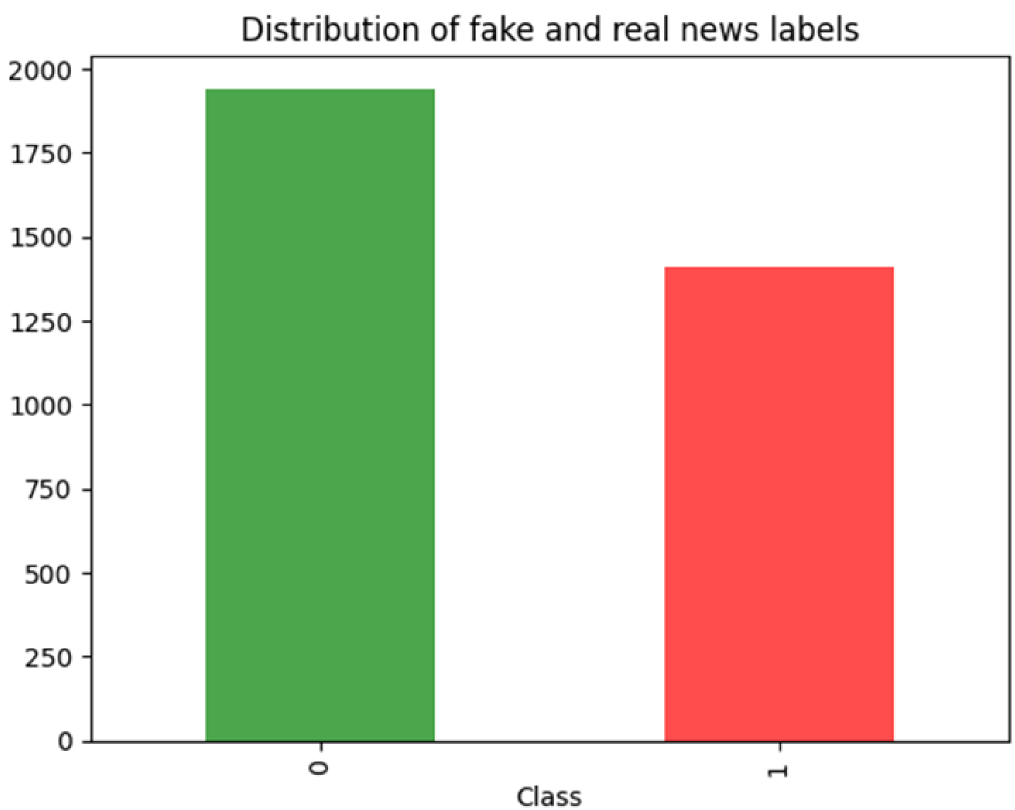


Data Visualization

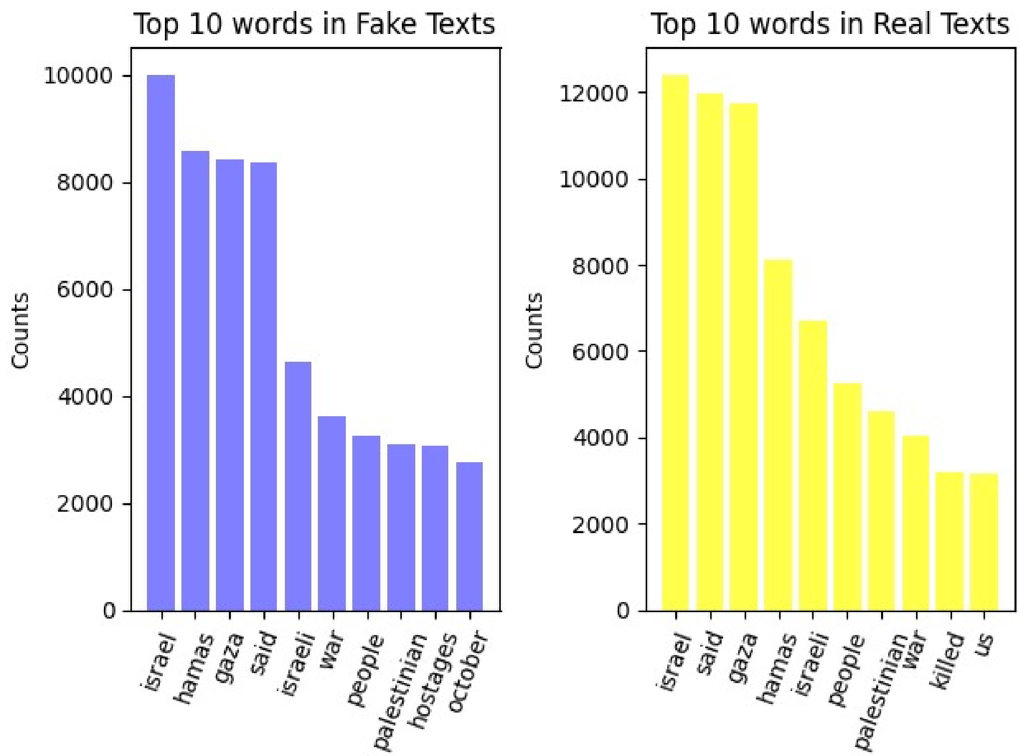
- There are 1940 occurrences of the value 'REAL' and 1409 occurrences of the value 'FAKE' within the 'Class' column of our Data.

```
print(df['Class'].value_counts())

✓ 0.0s

Class
REAL    1940
FAKE    1409
Name: count, dtype: int64
```

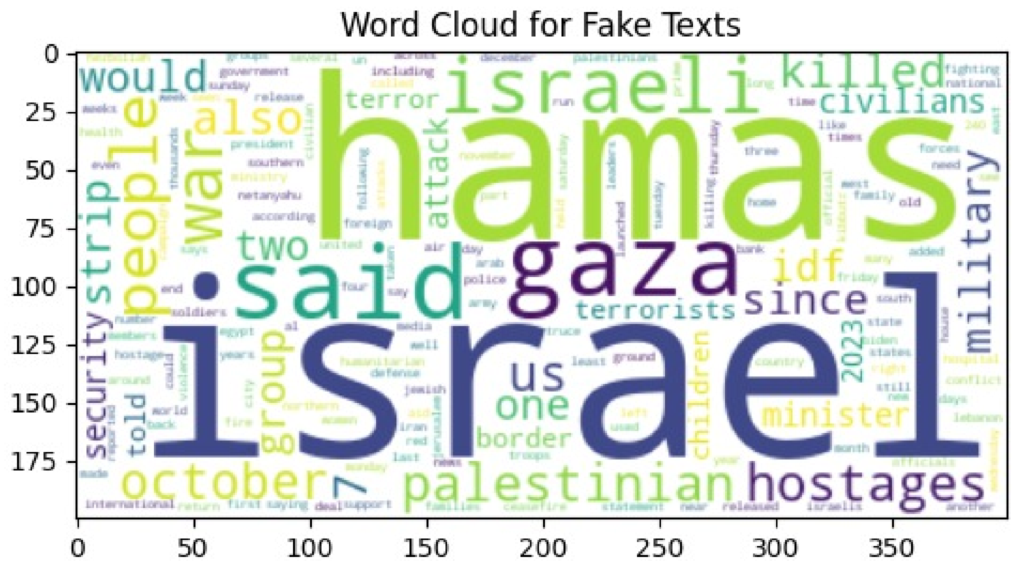
Distribution of fake and real news labels:



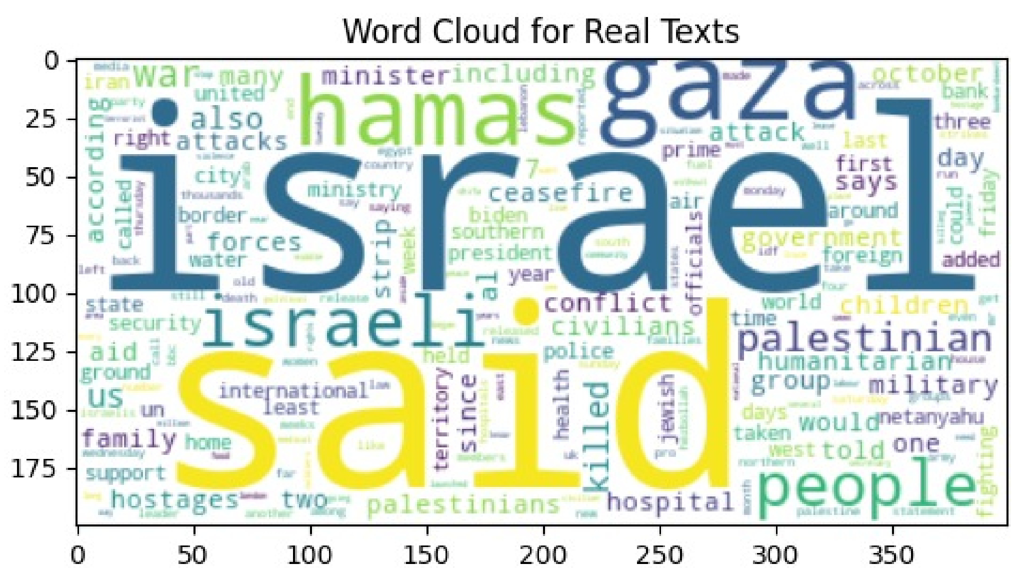
Visualize the Top Words in Fake and Real Texts:



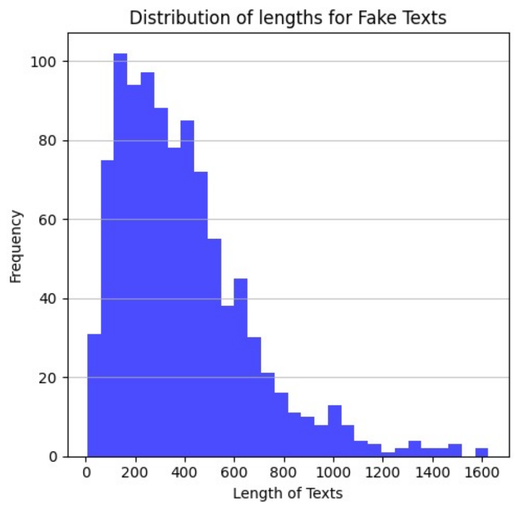
Word Cloud for Fake Texts:

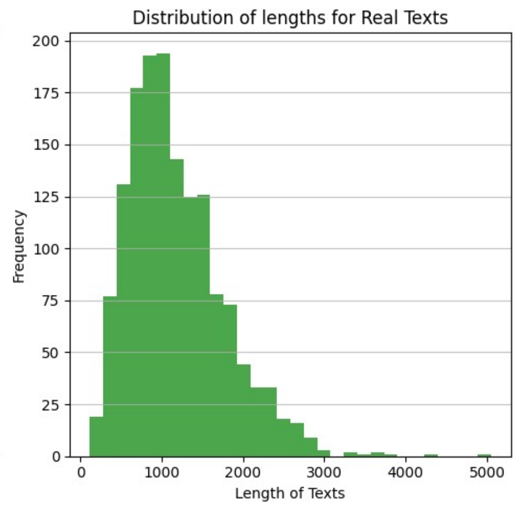


Word Cloud for Real Texts:



When comparing both histograms, it seems that fake texts are generally shorter than real texts:





1 Modeling: Text Preprocessing Methods

1.1 Bags of Words model (BoW)

NLP models cannot take in raw text, as machine learning models only understand numbers and vectors. To feed our real and fake news data into a model, we must first convert the text into something vectorial, and one option is the **Bags of Words (BoW)** model.

The **BoW** model treats text as an unordered set of words, where the sequence doesn't matter.

1.2 TF-IDF

TF-IDF stands for "Term Frequency-Inverse Document Frequency." It is a numerical statistic used in natural language processing (NLP) and information retrieval to evaluate the importance of a term (word or phrase) within a document relative to a collection of documents (corpus). TF-IDF is commonly used for text analysis and document retrieval tasks.

$$TF(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

$$IDF(t,D) = \log\left(\frac{\text{Total number of documents in the corpus } D}{\text{Number of documents containing term } t{+}1}\right)$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

1.3 GloVe

GloVe, which stands for "Global Vectors for Word Representation," is a word embedding technique used in natural language processing (NLP) and machine learning. It allows words to be represented as numerical vectors by leveraging co-occurrence statistics in large text corpora. These vectors capture semantic relationships between words and are used in various NLP applications to enhance word understanding and representation.

2 Techniques and Methods

2.1 Logistic Regression using Bags of Words (BoW)

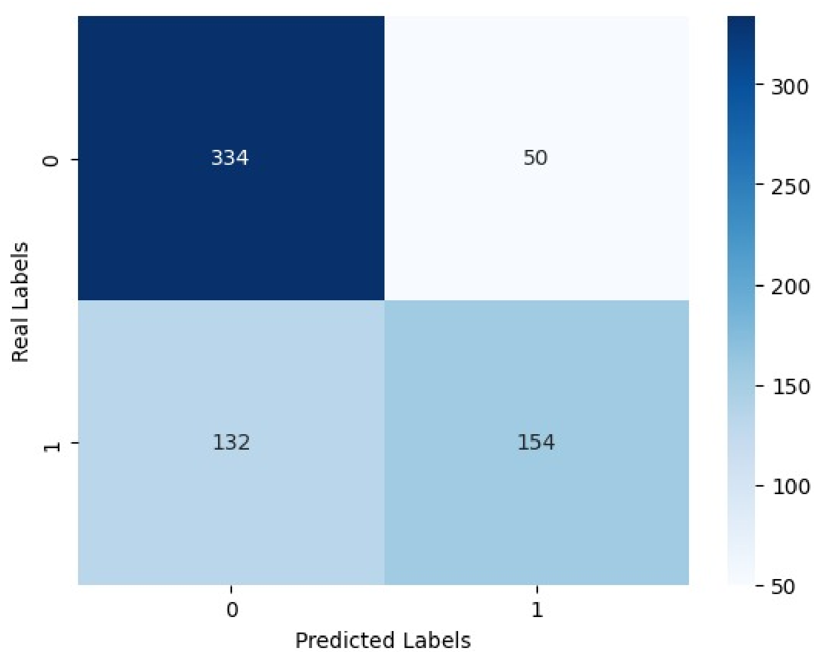
2.1.1 Classification Report:

0.1s

	precision	recall	f1-score	support
0	0.71	0.85	0.77	384
1	0.73	0.54	0.62	286
accuracy			0.72	670
macro avg	0.72	0.69	0.70	670
weighted avg	0.72	0.72	0.71	670

0.7164179104477612

2.1.2 Evaluate with Confusion Matrix:

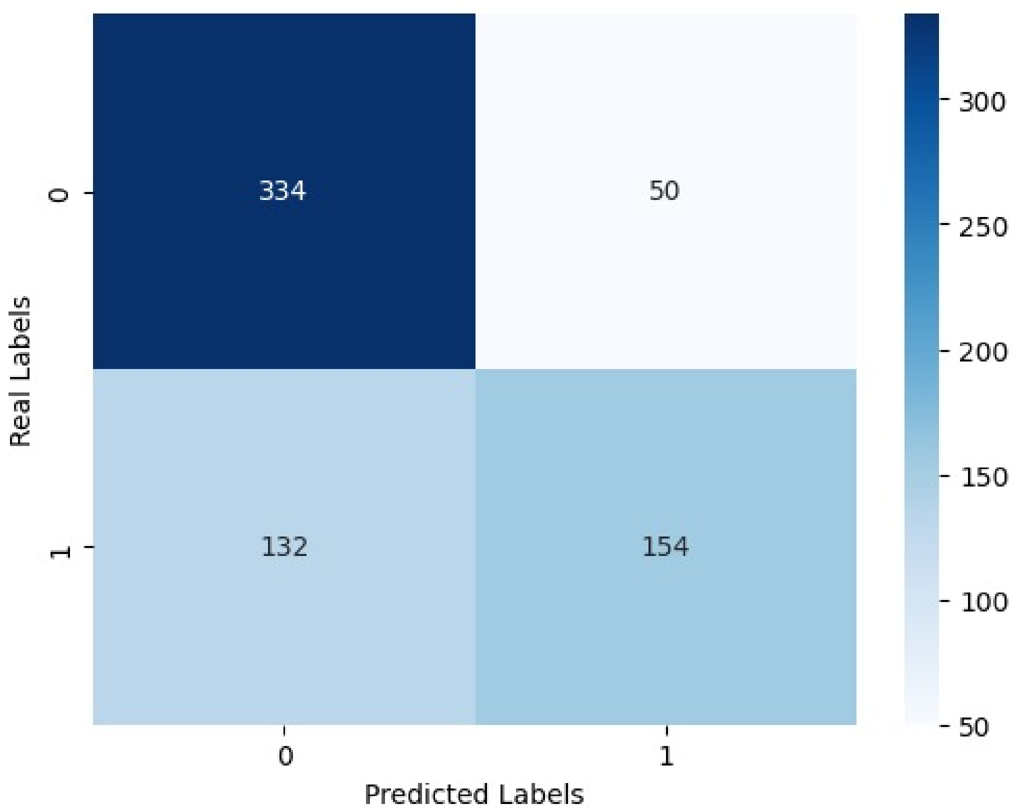


2.2 SVM using Bags of Words (BoW)

2.2.1 Classification Report:

```
              precision    recall  f1-score   support

           0       0.72      0.87      0.79       384
           1       0.75      0.54      0.63       286

    accuracy                           0.73       670
   macro avg       0.74      0.70      0.71       670
weighted avg       0.73      0.73      0.72       670

0.7283582089552239
```

2.2.2 Evaluate with Confusion Matrix:

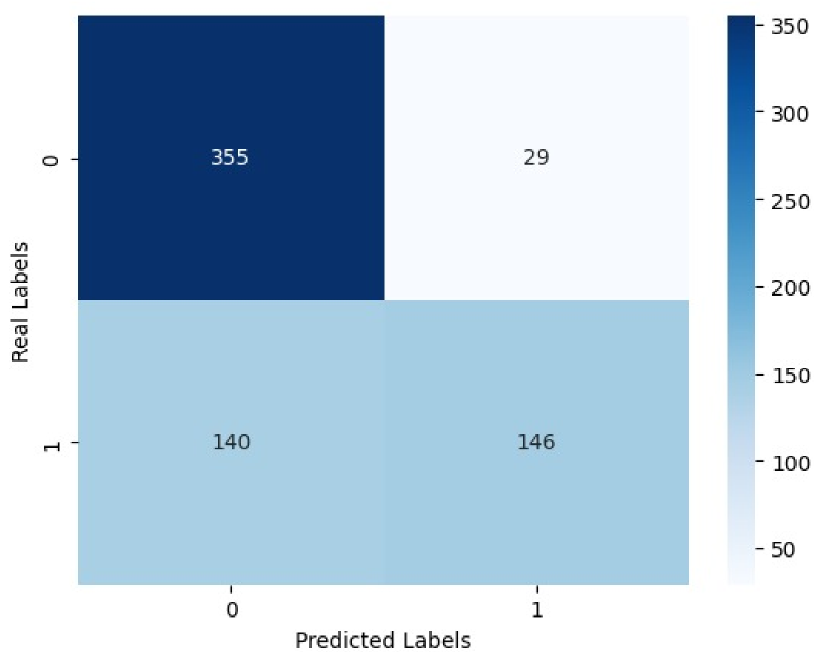


2.3 Random Forest (Untuned) using Bags of Words (BoW)

2.3.1 Classification Report:

```
0.2s
              precision    recall  f1-score   support

           0       0.72      0.92      0.81       384
           1       0.83      0.51      0.63       286

    accuracy                           0.75       670
   macro avg       0.78      0.72      0.72       670
weighted avg       0.77      0.75      0.73       670

0.7477611940298508
```

2.3.2 Evaluate with Confusion Matrix:



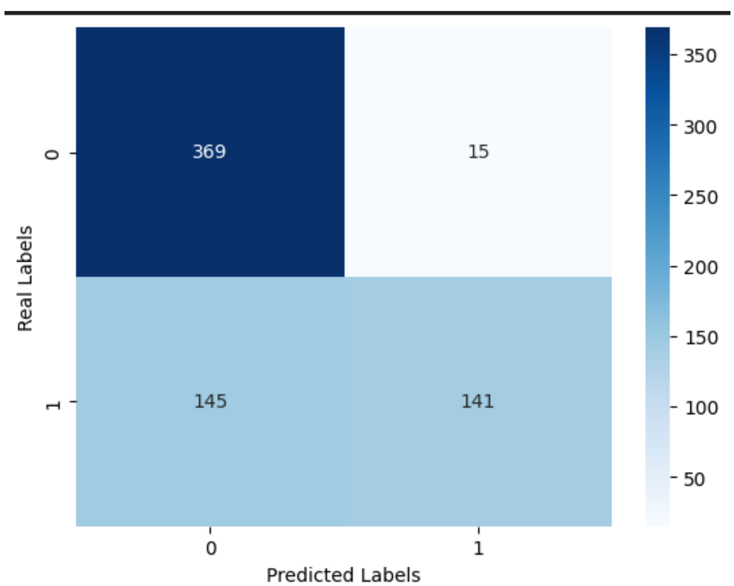
2.4 Random Forest (Tuned) using Bags of Words (BoW)

There are many hyperparameters to be tuned with random forests. We will use random search (cross-validation) instead of grid search, which may be faster and more efficient.

2.4.1 Classification Report :

```
Random Forest Results
              precision    recall  f1-score   support

           0       0.72      0.96      0.82       384
           1       0.90      0.49      0.64       286

    accuracy                           0.76       670
   macro avg       0.81      0.73      0.73       670
weighted avg       0.80      0.76      0.74       670

0.7611940298507462
```

2.4.2 Evaluate with Confusion Matrix :



3 RF,TRF,LG and SVM with TF-IDF:

```
                 Logistic Regression Results
                 --------------------------
               precision    recall  f1-score   support

           0       0.72      0.86      0.78       384
           1       0.75      0.54      0.63       286

    accuracy                           0.73       670
   macro avg       0.73      0.70      0.70       670
weighted avg       0.73      0.73      0.72       670

*******************Accuracy: 0.7253731343283583 ***************
                      SVM Linear Results
                 --------------------------
               precision    recall  f1-score   support

           0       0.71      0.83      0.76       384
           1       0.70      0.54      0.61       286

    accuracy                           0.71       670
   macro avg       0.71      0.69      0.69       670
weighted avg       0.71      0.71      0.70       670

*******************Accuracy: 0.7074626865671642 ***************
```

```
                    Random Forest Results
                ---------------------------
              precision    recall  f1-score   support

           0       0.72      0.92      0.81       384
           1       0.83      0.51      0.63       286

    accuracy                           0.75       670
   macro avg       0.77      0.72      0.72       670
weighted avg       0.76      0.75      0.73       670

*******************Accuracy: 0.746268656716418 ***************
                   Tunned  Random Forest Results
                ---------------------------
              precision    recall  f1-score   support

           0       0.72      0.93      0.81       384
           1       0.85      0.52      0.65       286

    accuracy                           0.76       670
   macro avg       0.79      0.73      0.73       670
weighted avg       0.78      0.76      0.74       670

*******************Accuracy: 0.7567164179104477 ***************
```

4 RF,TRF,LG and SVM with Glove model:

	Vector_type	Model	Accuracy_score
0	GloVe	Logistic Regression	0.656716
1	GloVe	Random Forest	0.697015
2	GloVe	SVM	0.637313

5 Models Evaluation

The tuned RF model with BOW vectorization has the highest accuracy score (0.76), which might lead one to interpret it as the best performing model among those listed. This is despite it not being the fastest model to fit. If the primary concern is prediction accuracy, and the time to train the model is not a limiting factor, then the RF with BOW would indeed be the best choice among these options.

6 Sequential Models (RNNs, LSTMs)

So far, we have been dealing with models that treat predictors (words) independently and have not been able to incorporate any sequentiality, which is obviously critical in speech and text. Words in a sentence have a particular order that otherwise would not make sense to another human being.

So far, we have been using models that could hopefully detect words that are commonly used in real vs. fake articles, instead of properly understanding the text, which could lead to better results.

In this case, we are doing a special form of prediction that is **'many to one'**, where we take a sequence of words but only produce a single output, but where the decision is only made at the last word.

6.1 Simple RNNs

6.1.1 RNN architecture :

```
from tensorflow.keras.layers import SimpleRNN, Embedding, Dropout
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.callbacks import EarlyStopping


from keras.layers import Dropout

model_rnn = Sequential([
    Embedding(input_dim = len(word_index), output_dim = 8,
              input_length=avg_length_text), ## recall that we set the post-padding length to be this value
    SimpleRNN(16), ## THIS IS THE RECURRENT LAYER
    Dropout(0.2),
    Dense(8, activation='relu'),
    Dropout(0.2),
    Dense(1, activation= 'sigmoid') ## final layer for prediction, hence only one node
])

## compile -- add optim, add loss, add metrics
model_rnn.compile(optimizer = 'adam', loss = ['binary_crossentropy'], metrics = ['accuracy'])

early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

history_rnn = model_rnn.fit(X_train_pad, y_train, epochs=100, validation_data=(X_test_pad, y_test), callbacks=[early_stopping])
```

6.1.2 Result:

```
## Test set
loss, accuracy = model_rnn.evaluate(X_test_pad, y_test)
print("Test Loss:", loss)
print("Test Accuracy:", accuracy)
✓ 1.1s

21/21 [==============================] - 1s 37ms/step - loss: 0.6771 -
Test Loss: 0.6770897507667542
Test Accuracy: 0.5686567425727844
```

6.2 LSTM:

6.2.1 LSTM architecture :

```
from tensorflow.keras.layers import LSTM


model_lstm = Sequential([
    Embedding(input_dim=len(word_index) + 1, output_dim=8, input_length=avg_length_text),
    LSTM(16),  # 16 LSTM units
    Dropout(0.2),
    Dense(8, activation='relu'),
    Dropout(0.2),  # Adding dropout
    Dense(1, activation='sigmoid')
])

model_lstm.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

history_lstm = model_lstm.fit(X_train_pad, y_train, epochs=50,
                              validation_data=(X_test_pad, y_test), callbacks=[early_stopping])
```

6.2.2 Result :

```
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense

## Test set
loss, accuracy = model_lstm.evaluate(X_test_pad, y_test)
print("Test Loss:", loss)
print("Test Accuracy:", accuracy)
```

✓ 2.2s

```
 3/21 [===>..........................] - ETA: 1s - loss: 0.5483 - accur
21/21 [==============================] - 2s 76ms/step - loss: 0.6038 - a
Test Loss: 0.6038157939910889
Test Accuracy: 0.6910447478294373
```

6.3 BERT:

In the process of building a fake news classifier, we must first set several crucial parameters. We designate a `train_fraction` of 0.8, allocating 80% of our dataset for training, with the remaining 20% reserved for validation. Over the course of `num_train_epochs`, which is set to 3, our model will undergo training.

During this training, we employ a `train_batch_size` of 8, specifying the number of data samples processed in each training batch, while the `eval_batch_size` is set to 16 for the validation phase.

To optimize the training process, we incorporate `warmup_steps` of 50 and introduce a `weight_decay` of 0.02. Utilizing the `BERT_MODEL` `"bert-base-cased,"` or the option `"distilbert-base-cased"` if preferred, we steer the architecture towards our chosen model configuration. Lastly, all the generated model files will be stored in the output directory named `"fake-news-classifier."` These parameter settings lay the foundation for an effective fake news detection system.

6.3.1 BERT result :

Epoch	Training Loss	Validation Loss	Accuracy
1	0.431900	0.538157	0.738806
2	0.506900	0.511132	0.746269
3	0.311800	0.550451	0.731343

7 Deployment:

7.0.1 True News:

7.0.2 Fake News:

8 Conclusion

In conclusion, this project on fake news classification involving a diverse range of algorithms and models, such as Random Forest, SVM, Logistic Regression, various vectorization techniques including Bag of Words (BoW), TF-IDF, and GloVe, as well as more advanced approaches like RNN, LSTM, and BERT, has yielded valuable insights. Despite the versatility and power of deep learning models like BERT, the project's data size played a crucial role in determining the most effective solution. Tuned

Random Forest emerged as the best-performing model, highlighting the importance of selecting the appropriate algorithm for the specific dataset and problem at hand. This outcome reinforces the idea that machine learning solutions need not always rely on complex deep learning models, especially when working with limited data resources.

Furthermore, deploying the project with a user-friendly interface using Streamlit to detect fake news during the Palestino-Israel War demonstrates the practicality and real-world applicability of the developed solution. This project not only showcases the importance of tailoring machine learning approaches to the task's requirements but also demonstrates the potential to address critical issues such as fake news detection in sensitive and dynamic contexts.