NAME OF THE PROJECT

Fake-news Project

Submitted by:

MANISH SINGH

INTRODUCTION

- Business Problem Framing

  Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse

  Fake news's simple meaning is to incorporate information that leads people to the wrong path. Nowadays fake news spreading like water and people share this information without verifying it. This is often done to further or impose certain ideas and is often achieved with political agendas.

- Data Sources and their formats

There are two datasets one for fake news and one for true news. In true news, there is 21417 news, and in fake news, there is 23481 news. You have to insert one label column zero for fake news and one for true news. We are combined both datasets using pandas built-in function.

- Data Preprocessing Done

  By using NLTK ,regex library and pandas to clean the text data.

  we done text_vectorization for converting text data into numerical format through sklearn .

  we also used some functions of tensorflow to build deep learning model

- Hardware and Software Requirements and Tools Used

  NLTK = removing stopwords and for lemmatization

  regex = removing punctuation marks and unnecessary character from text data

  pandas = for data manipulation

  matplotlib and WordCloud = for data visualization

  sklearn = for Text_vectorization and building machine learning model like Randomforest, MultinomialNB

  Tensorflow = for building deep learning model

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  For ML model building

  First we done cleaned and preprocessed the text data for making to used in building ML model

I used three ML algorithms for model building such as RandomForest, MultinomialNB and LogisticRegression and after that built a deep learning model through tensorflow

- Testing of Identified Approaches (Algorithms)

RandomForestClassifier, MultinomialNB and LogisticRegression by using sklearn

Deep learning model with LSTM by using Tensorflow

- Run and Evaluate selected models

1. this is RandomForestClassifier model with code and evaluation metrics

```
]: # RANDOMFOREST_CLASSIFER MODEL
pipe = Pipeline([('tf_idf', TfidfVectorizer()),('rf_c', RandomForestClassifier())])

pipe.fit(xtrain, ytrain)
pred = pipe.predict(xtest)

print('accuracy_score=', accuracy_score(ytest, pred))
print('precision_score=', precision_score(ytest, pred))
print('recall_score=', recall_score(ytest, pred))
print('f1_score=', f1_score(ytest, pred))
print('confusion matrix \n', confusion_matrix(ytest, pred))
```

```
accuracy_score= 0.9932103655086568
precision_score= 0.9934287725885942
recall_score= 0.9924970691676436
f1_score= 0.9929627023223082
confusion matrix
 [[4544   28]
 [  32 4233]]
```

2. this is MultinomialNB model with evalaution metrics

```
# MULTINOMIAL_NB MODEL

pipe_nb = Pipeline([('tf_idf', TfidfVectorizer()),('mnb', MultinomialNB())])

pipe_nb.fit(xtrain, ytrain)
pred2 = pipe_nb.predict(xtest)

print('accuracy_score=', accuracy_score(ytest, pred2))
print('precision_score=', precision_score(ytest, pred2))
print('recall_score=', recall_score(ytest, pred2))
print('f1_score=', f1_score(ytest, pred2))
print('confusion matrix \n', confusion_matrix(ytest, pred2))
```

```
accuracy_score= 0.9378748444042095
precision_score= 0.9390359168241966
recall_score= 0.931770222743259
f1_score= 0.9353889608096975
confusion matrix
 [[4314  258]
 [ 291 3974]]
```

3. This is LogisticRegression model

```
# LOGISTIC REGRESSION MODEL

pipe_LR = Pipeline([('tf_idf', TfidfVectorizer()),('LR', LogisticRegression())])

pipe_LR.fit(xtrain, ytrain)
pred3 = pipe_LR.predict(xtest)

print('accuracy_score=', accuracy_score(ytest, pred3))
print('precision_score=', precision_score(ytest, pred3))
print('recall_score=', recall_score(ytest, pred3))
print('f1_score=', f1_score(ytest, pred3))
print('confusion matrix \n', confusion_matrix(ytest, pred3))
```

```
accuracy_score= 0.9864207310173135
precision_score= 0.9838897968713518
recall_score= 0.988042203985932
f1_score= 0.9859616284510996
confusion matrix
 [[4503   69]
 [  51 4214]]
```

- Key Metrics for success in solving problem under consideration

I used accuracy_score, precision_score, recall_score,

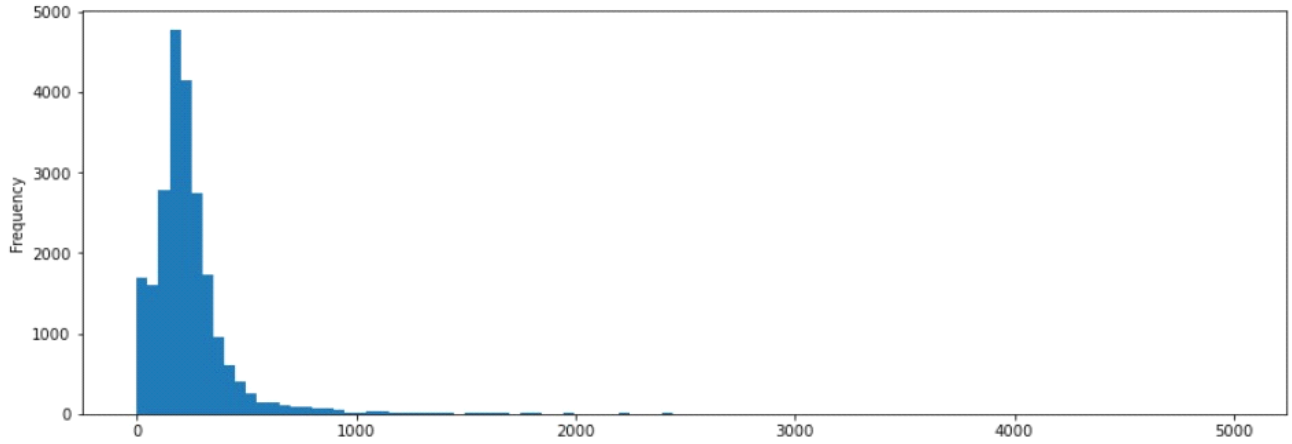
f1_csore and confusion_matrix over all models for evaluation.

- Visualizations

Fake-news words length distribution of each sentence

```
# Fake-news words length distribution
df['text_length'][df['label']==0].plot.hist(bins=100, figsize=(14,5))
```

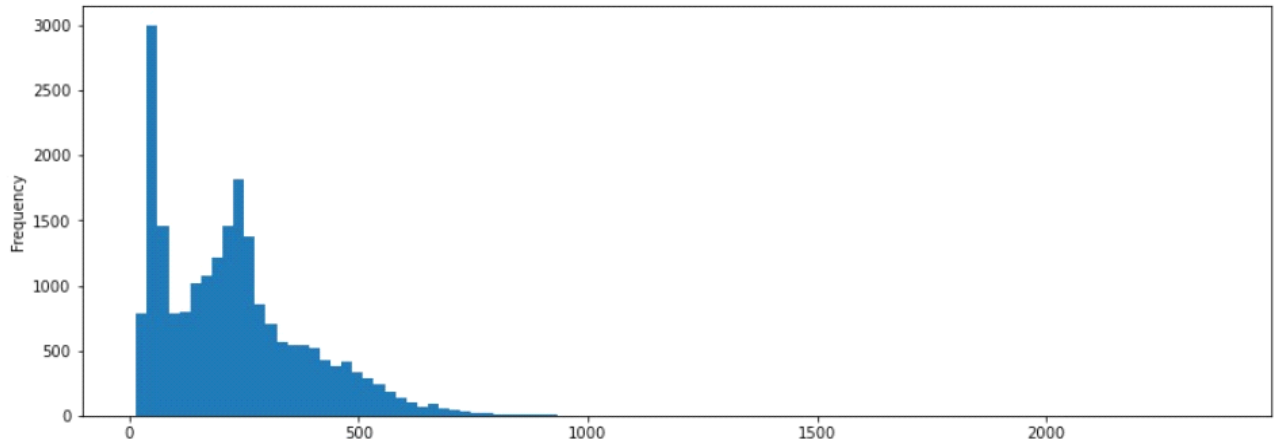
```
<AxesSubplot:ylabel='Frequency'>
```



True-news words length distribution of each sentence

```
# Plot the True-news words length distribution
df['text_length'][df['label']==1].plot.hist(bins=100, figsize=(14,5))
```

```
<AxesSubplot:ylabel='Frequency'>
```

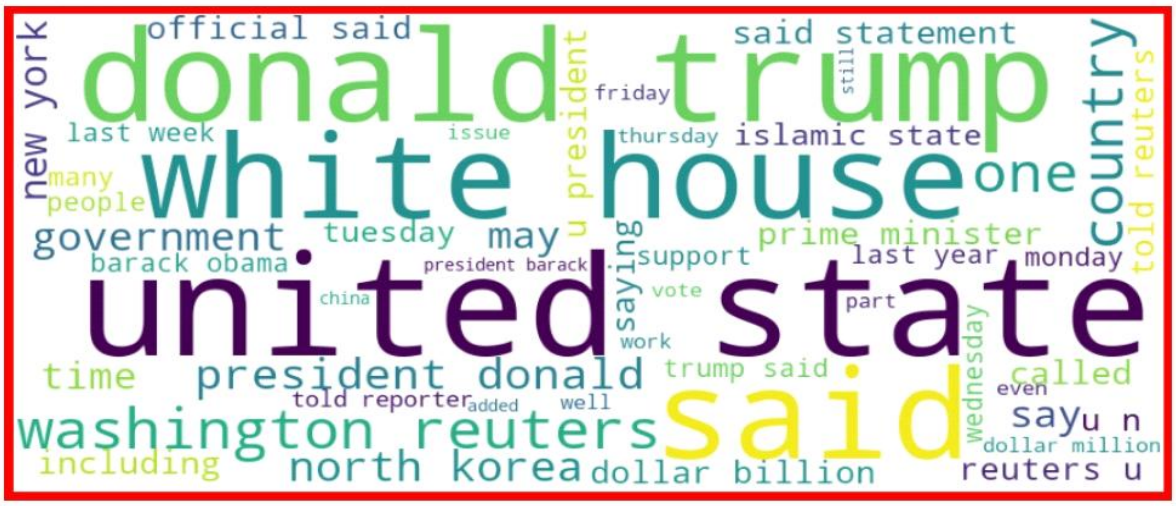


GETTING SENSE OF CLOUD WORDS OF fake_news

```
# GETTING SENSE OF CLOUD WORDS OF fake_news
from wordcloud import WordCloud

fake_news= df['text'][df['label']==1]
fake_news_cloud= WordCloud(width=900,height=400, background_color='white', max_words=50).generate(' '.join(fake_news))

plt.figure(figsize=(12,7), facecolor='r')
plt.imshow(fake_news_cloud)
plt.axis('off')
plt.tight_layout()
plt.show()
```



- Interpretation of the Results

  From visualization part, we get which words are occured more in fake and true news and these words are also important to make sentences fake or true news.

CONCLUSION

- Key Findings and Conclusions of the Study

Important point of this project is built a model in such way to get good accuracy of between Fake-news and True-news.

we get best precision_score, recall_score, f1_score from RandomForest_classifier, LogisticRegression and deep-learning LSTM model