Fake News Detection

Jeremy Goh, Adrian Ong, Nigel Chok [Team 4, SC12]
SC1015 - Intro to Data Science & Artificial Intelligence

BACKGROUND

Anyone can make up a story and go viral

1

Problem

Increasing trend of Fake-News

2

Effects

Distrust in Media
Spread of Harmful Conspiracy Theories

3

Measures in Place

Fact-Checking Organizations
POFMA Regulation

PROBLEM DEFINITION

How can we effectively identify fake news with the attributes of a news article?

DATASET

Train.csv: This is a full training dataset with the following attributes:

- id: unique id for a news article
- title: the title of a news article
- author: author of the news article
- text: the text of the article; could be incomplete
- label: a label that marks the article as potentially unreliable
    - 1: Fake
    - 0: Not Fake

	A	B	C	D	E
1	id	title	author	text	label
2	0	House Dem Aide: We Didn?€?t Even Se	Darrell Lucus	House Dem Aide: We	1
3	1	FLYNN: Hillary Clinton, Big Woman on C	Daniel J. Flynn	Ever get the feeling your life	0
4	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get	1
5	3	15 Civilians Killed In Single US Airstrike	Jessica Purkiss	Videos 15 Civilians Killed In	1
6	4	Iranian woman jailed for fictional unpu	Howard Portnoy	Print	1
7	5	Jackie Mason: Hollywood Would Love T	Daniel Nussbaum	In these trying times, Jackie	0
8	6	Life: Life Of Luxury: Elton John?€?s 6 Fa	nan	Ever wonder how	1
9	7	Beno??t Hamon Wins French Socialist F	Alissa J. Rubin	PARIS ?€? France chose an	0
10	8	Excerpts From a Draft Script for Donald	nan	Donald J. Trump is schedule	0
11	9	A Back-Channel Plan for Ukraine and Ru	Megan Twohey and Scott Shane	A week before Michael T. Fly	0
12	10	Obama?€?s Organizing for Action Partr	Aaron Klein	Organizing for Action, the ac	0
13	11	BBC Comedy Sketch "Real Housewives (	Chris Tomlinson	The BBC produced spoof on	0
14	12	Russian Researchers Discover Secret Na	Amando Flavio	The mystery surrounding	1
15	13	US Officials See No Link Between Trump	Jason Ditz	Clinton Campaign Demands	1
16	14	Re: Yes, There Are Paid Government Tr	AnotherAnnie	Yes, There Are Paid	
17	BART SIMPSONSON				
18	Hey	it?€?s just another means of getting th	channels	and programs fellating them daily?€??€?. James	
19	It?€?s not re	I imagine most governments do it. And	oil companies spreading disinformation about cl	difficult to know who to trust on the Internet these days.	
20	In any societ	most people do nothing. It?€?s up to the	minority to defend the naive majority. It?€?s	how things are done. Bob G	
21	If I read the article correctly the government is targeting conservative thought. I always wondered why liberals would deliberately read conservative web sites a				
22	The DNC is be	stupid and racist. (Not to say that there	but these j@ck@sses ramp it up to 11.) Tami Chapman		
23	I almost post	which was taken totally out of context.	especially the conservatives. It?€?s Independen	1	

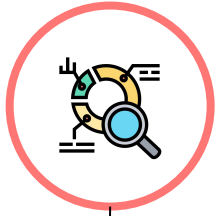
GENERAL FLOW

Data Cleaning

1

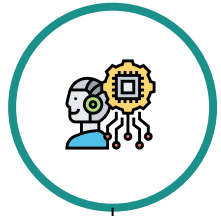
Data Preparation

2



Exploratory Data Analysis

3



Model Training

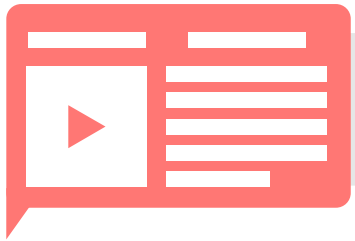
4

Insights & Recommendations

5

Data Cleaning

DATA CLEANING



NULL VALUES

Re-evaluate the worth of data and replace NULL values

1

Variables with NULL values

- Authors
  - To be replaced with “Unknown” to preserve row record which will aid in our further analysis.
- Title
  - To be replaced with the text of the article. Word count would be the median value for title word count.
- Text
  - To be replaced with the title of the article. Word count would be the median value for text word count.

Rows with NULL title and text will be completely dropped from the dataset.

	id	title	author	text	label
2	0	hous dem aid even see c	Darrell Lucus	hous dem aid even see comey letter	1
3	1	flynn hillari clinton big w	Daniel J. Flynn	ever get feel life circl roundabout rat	0
4	2	truth might get fire	Consortiumnews.com	truth might get fire octob tension int	1
5	3	civilian kill singl us airstri	Jessica Purkiss	video civilian kill singl us airstrik iden	1
6	4	iranian woman jail fictior	Howard Portnoy	print iranian woman sentenc six year	1
7	5	jacki mason hollywood lo	Daniel Nussbaum	tri time jacki mason voic reason weel	0
8	6	life life luxuri elton john f	Unknown	ever wonder britain icon pop pianist	1
9	7	beno hamon win french	Alissa J. Rubin	pari franc chose idealist tradit candid	0
10	8	excerpt draft script dona	Unknown	donald j trump schedul make highli a	0
11	9	back channel plan ukrain	Megan Twohey and Sc	week michael flynn resign nation sec	0
12	10	obama organ action part	Aaron Klein	organ action activist group morph ba	0
13	11	bbc comedi sketch real h	Chris Tomlinson	bbc produc spoof real housew tv pro	0
14	12	russian research discov s	Amando Flavio	mysteri surround third reich nazi ger	1
15	13	us offici see link trump ru	Jason Ditz	clinton campaign demand fbi affirm	1
16	14	ye paid govern troll socia	AnotherAnnie	ye paid govern troll social media blog	1
17	15	major leagu soccer argen	Jack Williams	guillermo barro schelotto first argent	0
18	16	well fargo chief abruptli s	Michael Corkery and S	scandal engulf well fargo toppl chairr	0
19	17	anonym donor pay millio	Starkman	caddo nation tribal leader freed sper	1

Data Preparation

DATA PREPARATION

1

Removal of Non Essential Text

Analysing of continuous sequences of words which computes the top used terms.

2

Part of Speech Tagging

A form of text analysis used to perform comparisons between terms at a large scale

REMOVAL OF NON-ESSENTIAL TEXT

Word stemming

```
# Function to perform stemming (Reducing words to their root form)
def stem(string):
    # Tokenize sentence into list
    char_arr = string.split()
    # Iterate list of words, stemming each word
    char_arr = [ps.stem(char) for char in char_arr]
    # Form the sentence back
    return " ".join(char_arr)
```

```
# Apply all functions to whole data
news_data['text'] = news_data['text'].apply(stem)
# Do the same for title
news_data['title'] = news_data['title'].apply(stem)
```

Removal of unnecessary data (Symbols & Numbers)

```
def remove_symbols_numbers(string):
    # Use regex to replace for anything that is not alphabets and punctuation marks
    string = re.sub('[^a-zA-Z!?\']', ' ', string)
    # Eliminate multiple spaces
    return " ".join(string.split())
```

```
# Apply all functions to whole data
news_data['text'] = news_data['text'].apply(remove_symbols_numbers)
# Do the same for title
news_data['title'] = news_data['title'].apply(remove_symbols_numbers)
```

Removal of stopwords

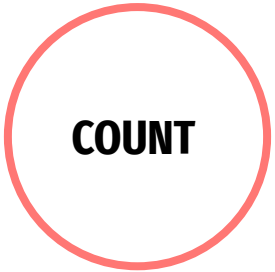
```
# Remove stopwords
nltk.download('stopwords')

# Print stopwords to see
my_stopwords = stopwords.words('english')

# Add some additional stopwords
my_stopwords.extend(["a", "about", "above", "after", "again", "against", "all", "am", "an", "and", "any", "are",
                     "aren't", "as", "at", "be", "because", "been", "before", "being", "below", "between", "both",
                     "but", "by", "can't", "cannot", "could", "couldn't", "did", "didn't", "do", "does", "doesn't",
                     "doing", "don't", "down", "during", "each", "few", "for", "from", "further", "had", "hadn't", "has",
                     "hasn't", "have", "haven't", "having", "he", "he'd", "he'll", "he's", "her", "here", "here's", "hers",
                     "herself", "him", "himself", "his", "how", "how's", "i", "i'd", "i'll", "i'm", "i've", "if", "in", "into",
                     "is", "isn't", "it", "it's", "its", "itself", "let's", "me", "more", "most", "mustn't", "my", "myself",
                     "no", "nor", "not", "of", "off", "on", "once", "only", "or", "other", "ought", "our", "ours", "ourselves",
                     "out", "over", "own", "same", "shan't", "she", "she'd", "she'll", "she's", "should", "shouldn't", "so", "so
                     "such", "than", "that", "that's", "the", "their", "theirs", "them", "themselves", "then", "there", "there's
                     "these", "they", "they'd", "they'll", "they're", "they've", "this", "those", "through", "to", "too", "under
                     "until", "up", "very", "was", "wasn't", "we", "we'd", "we'll", "we're", "we've", "were", "weren't", "what",
                     "what's", "when", "when's", "where", "where's", "which", "while", "who", "who's", "whom", "why", "why's",
                     "with", "won't", "would", "wouldn't", "you", "you'd", "you'll", "you're", "you've", "your", "yours",
                     "yourself", "yourselves"
                    ])
print(my_stopwords:=set(my_stopwords))
```

PART OF SPEECH TAGGING (POS)

id	title	author	text	label	title_pos_noun	title_pos_num	title_pos_pron	title_pos_verb	text_pos_adj	text_pos_adv	text_pos_noun	text_pos_num	text_pos_pron	text_pos_verb
0	hous dem aid even see comey letter jason chaff...	Darrell Lucus	hous dem aid even see comey letter jason chaff...	1	6	0	0	1	102	16	234	3	0	68
1	flynn hillari clinton big woman campu breitbart	Daniel J. Flynn	ever get feel life circl roundabout rather hea...	0	5	0	0	1	81	13	208	3	3	45
2	truth might get fire	Consortiumnews.com	truth might get fire octob tension intellig an...	1	2	0	0	2	155	33	385	4	3	92
3	civilian kill singl us airstrik identifi	Jessica Purkiss	video civilian kill singl us airstrik identifi...	1	2	0	1	1	74	5	158	6	12	44
4	iranian woman jail fiction unpublish stori wom...	Howard Portnoy	print iranian woman sentenc six year prison ir...	1	7	0	0	1	13	5	54	2	0	11

EXPLORATORY DATA ANALYSIS

1



Word & Char Count

A measure which computes the count of words and characters in a specified text column.

2

Corpus Analysis

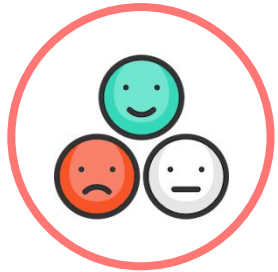
A form of text analysis used to perform comparisons between terms at a large scale

3

N-Gram Analysis

Analysing of continuous sequences of words which computes the top used terms.

4



Emotion & Sentiment

A statistical measure which computes the relevancy of a word to a document within a collection of documents

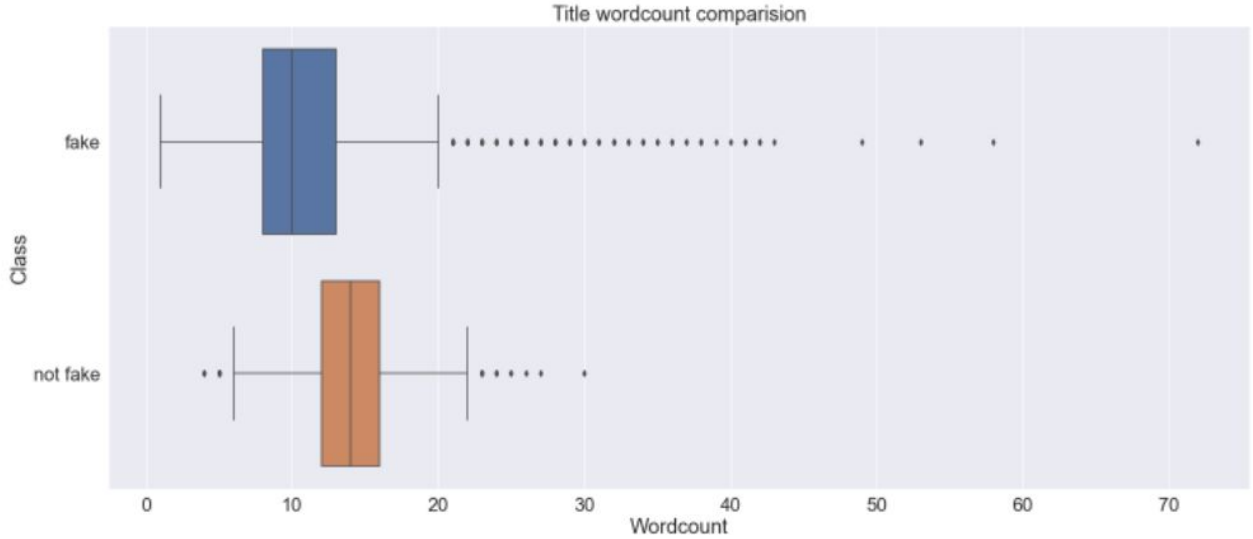
COUNT

WORD & CHAR COUNT

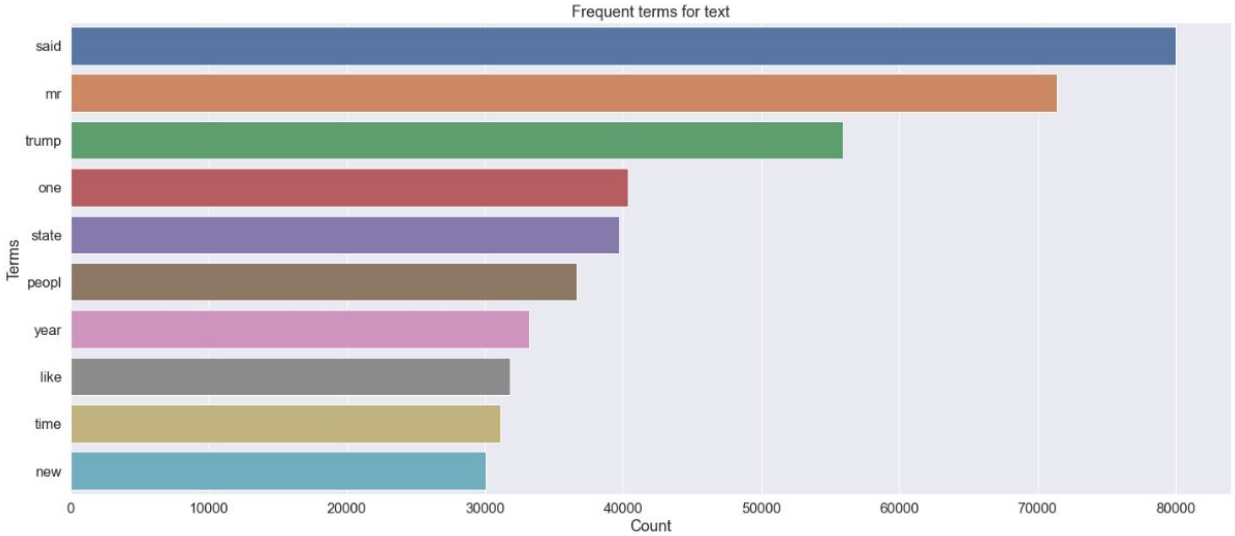
Fake vs not fake wordcount

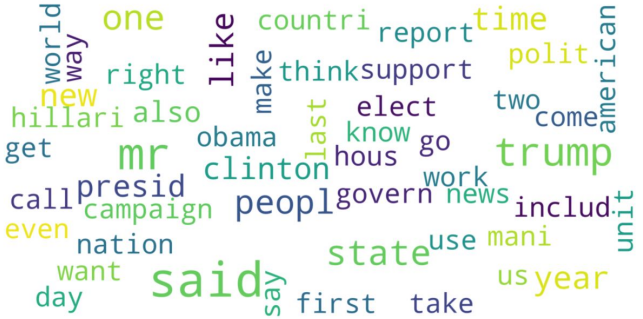
In [228]:
```
# Plot histogram for both real and fake news word count
f = plt.figure(figsize=(24,10))
sb.boxplot(x=news_data["title_wordcount"], y=news_data["label_translated"]).set(
    title="Title wordcount comparision",
    xlabel='Wordcount',
    ylabel='Class')
```

Out[228]:
```
[Text(0.5, 1.0, 'Title wordcount comparision'),
 Text(0.5, 0, 'Wordcount'),
 Text(0, 0.5, 'Class')]
```

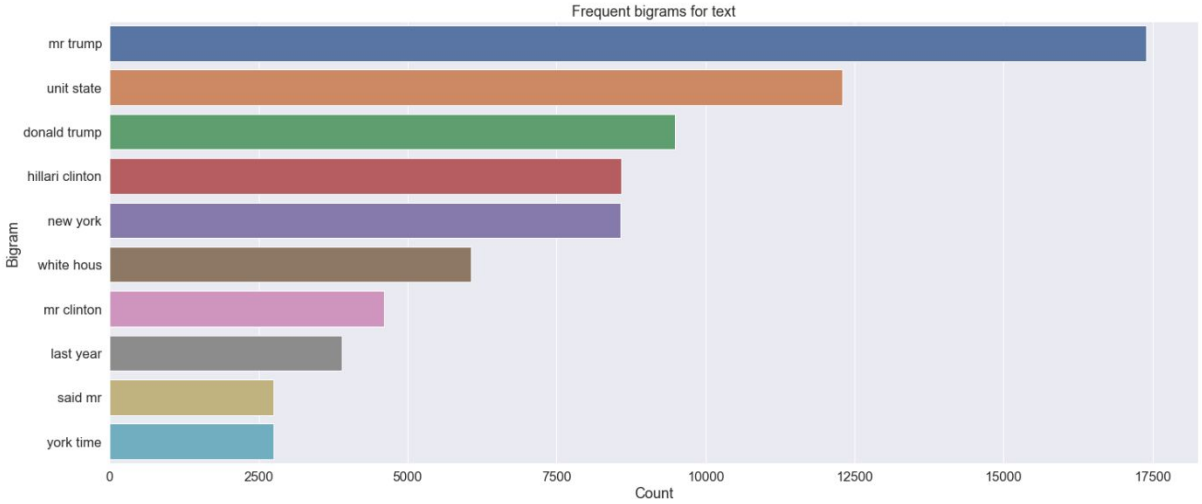


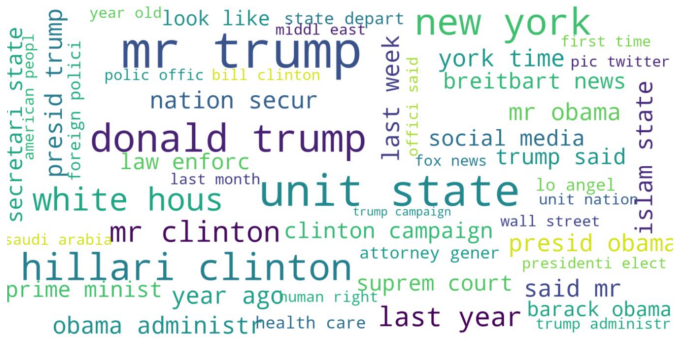
CORPUS ANALYSIS



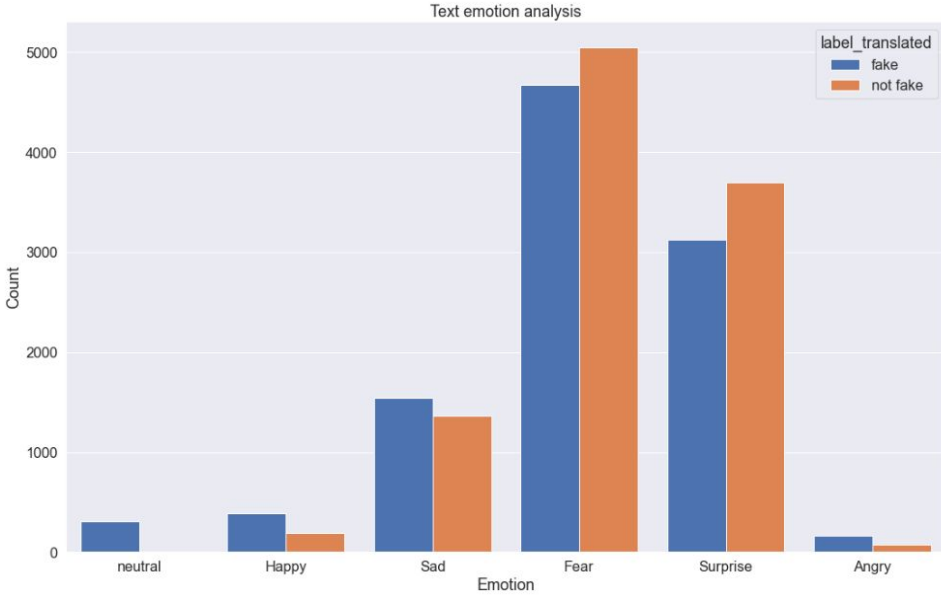


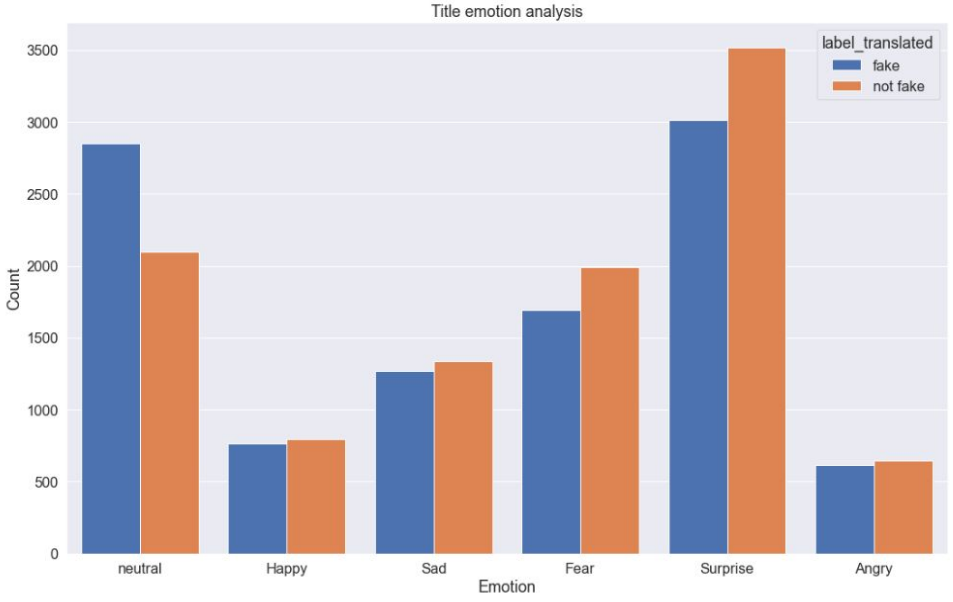
N-GRAM ANALYSIS



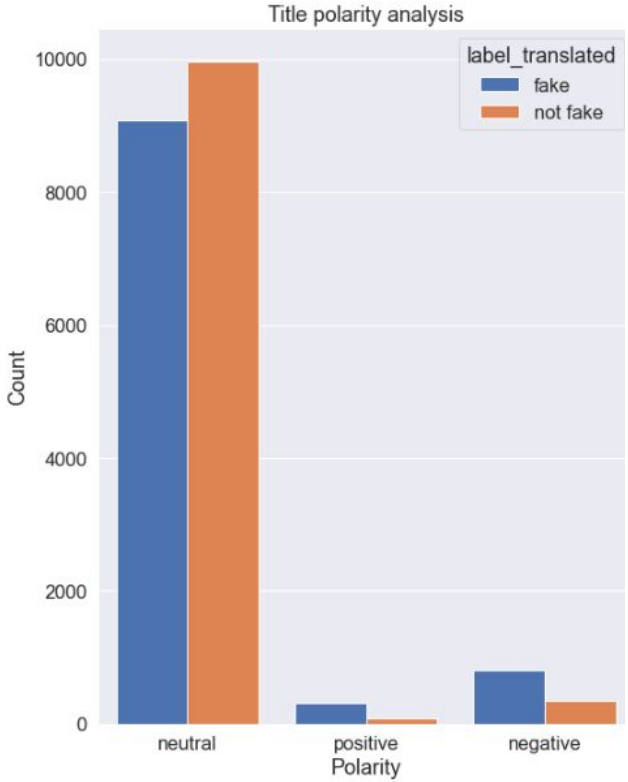


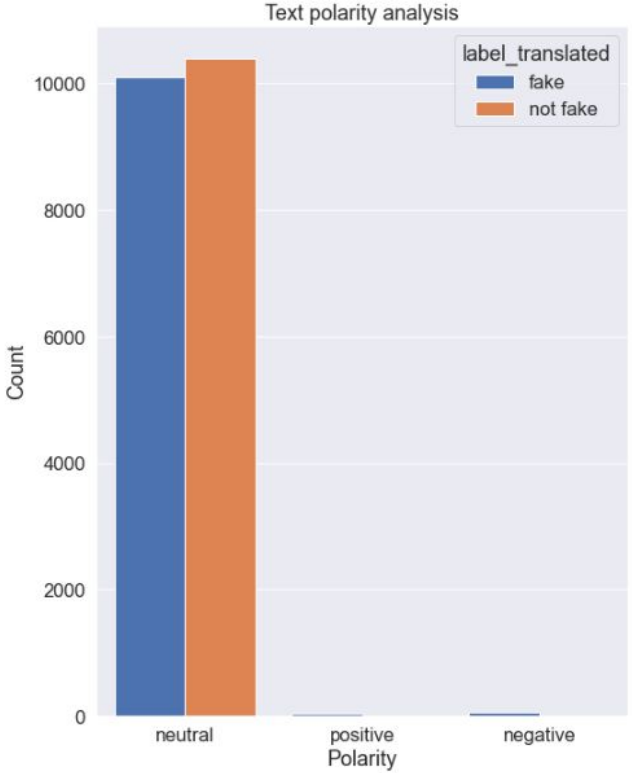
EMOTION ANALYSIS

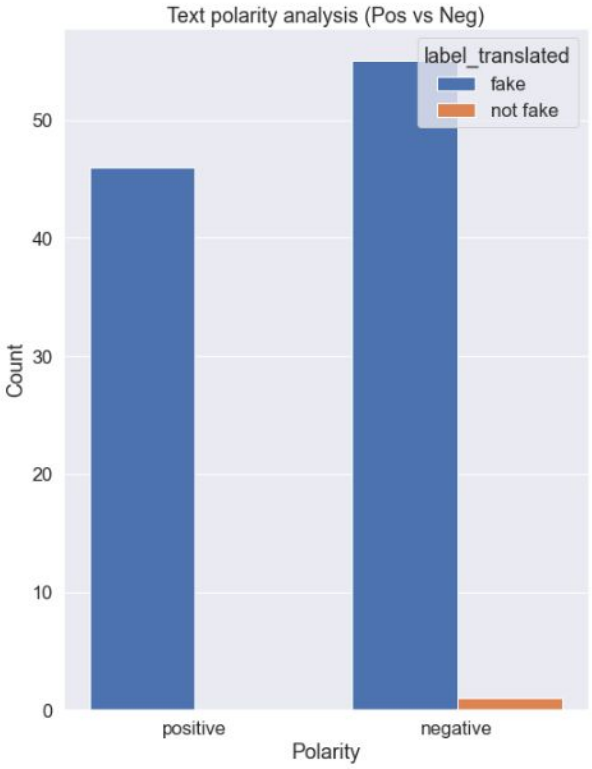




SENTIMENT ANALYSIS



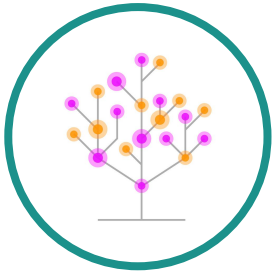




Model Training

MODEL TRAINING

1



Decision Tree

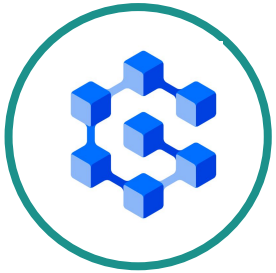
A classification algorithm that accounts for multiple variables

2

Random Forest

A classification algorithm that consists of many decision trees.

3



TF-IDF

A statistical measure which computes the relevancy of a word to a document within a collection of documents

4

Logistic Regression

Algorithm to predict the decision boundary for a binary classification problem.

USING STRATIFIED K-FOLD FOR DATA CONSISTENCY

Stratified k- fold for decision tree

In [175]:
```
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=17)
cv_results = cross_val_score(dectree, X_train, y_train['label_translated'], cv=skf)
```

In [176]:
```
cv_results, cv_results.mean()
```

Out[176]:
```
(array([0.72206801, 0.7241499 , 0.73074254, 0.73273169, 0.72162444]),
 0.7262633140962071)
```

Stratified k-fold verification for random forest classifier

In [183]:
```
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=17)
cv_results = cross_val_score(rforest, X_train, y_train['label_translated'], cv=skf)
```

In [184]:
```
cv_results, cv_results.mean()
```

Out[184]:
```
(array([0.73664122, 0.73317141, 0.73872311, 0.73030198, 0.73064908]),
 0.7338973595460555)
```

```python
# Init Stratified K fold cross validation object
sf = StratifiedKFold(n_splits=5, shuffle=True, random_state=2)

# Run k-fold using logistic regression
cv_results = cross_val_score(lr, tfidf_train, y_train, cv=sf)
```

```python
for score in cv_results:
    print(f"Score: {score :.5f}")

print(f"Mean score: {cv_results.mean() :.5f}")
```

```
Score: 0.94080
Score: 0.93746
Score: 0.93046
Score: 0.93562
Score: 0.92773
Mean score: 0.93441
```

Stratified K-fold for logistic regression

DECISION TREE

FINDING TOP 5 PREDICTORS USING STRATIFIED K-FOLD

```
Predictor: title_wordcount, score: 0.70
Predictor: title_pos_adj, score: 0.68
Predictor: title_charcount, score: 0.67
Predictor: title_pos_noun, score: 0.64
Predictor: text_pos_noun, score: 0.63
Predictor: text_pos_adj, score: 0.63
Predictor: stopwords_count_text, score: 0.63
Predictor: text_wordcount, score: 0.63
Predictor: text_pos_verb, score: 0.63
Predictor: text_charcount, score: 0.63
Predictor: text_pos_num, score: 0.62
Predictor: text_pos_adv, score: 0.62
Predictor: stopwords_count_title, score: 0.57
Predictor: title_pos_verb, score: 0.57
Predictor: text_pos_pron, score: 0.55
Predictor: title_emotion, score: 0.54
Predictor: text_emotion, score: 0.54
Predictor: title_polarity, score: 0.53
Predictor: title_pos_pron, score: 0.52
Predictor: title_pos_adv, score: 0.52
Predictor: text_polarity, score: 0.51
Predictor: title_pos_num, score: 0.50
```

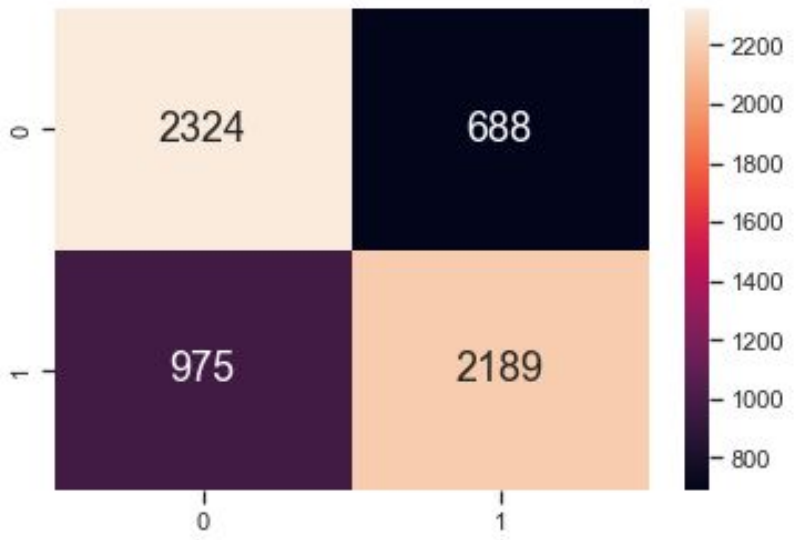
DECISION TREE

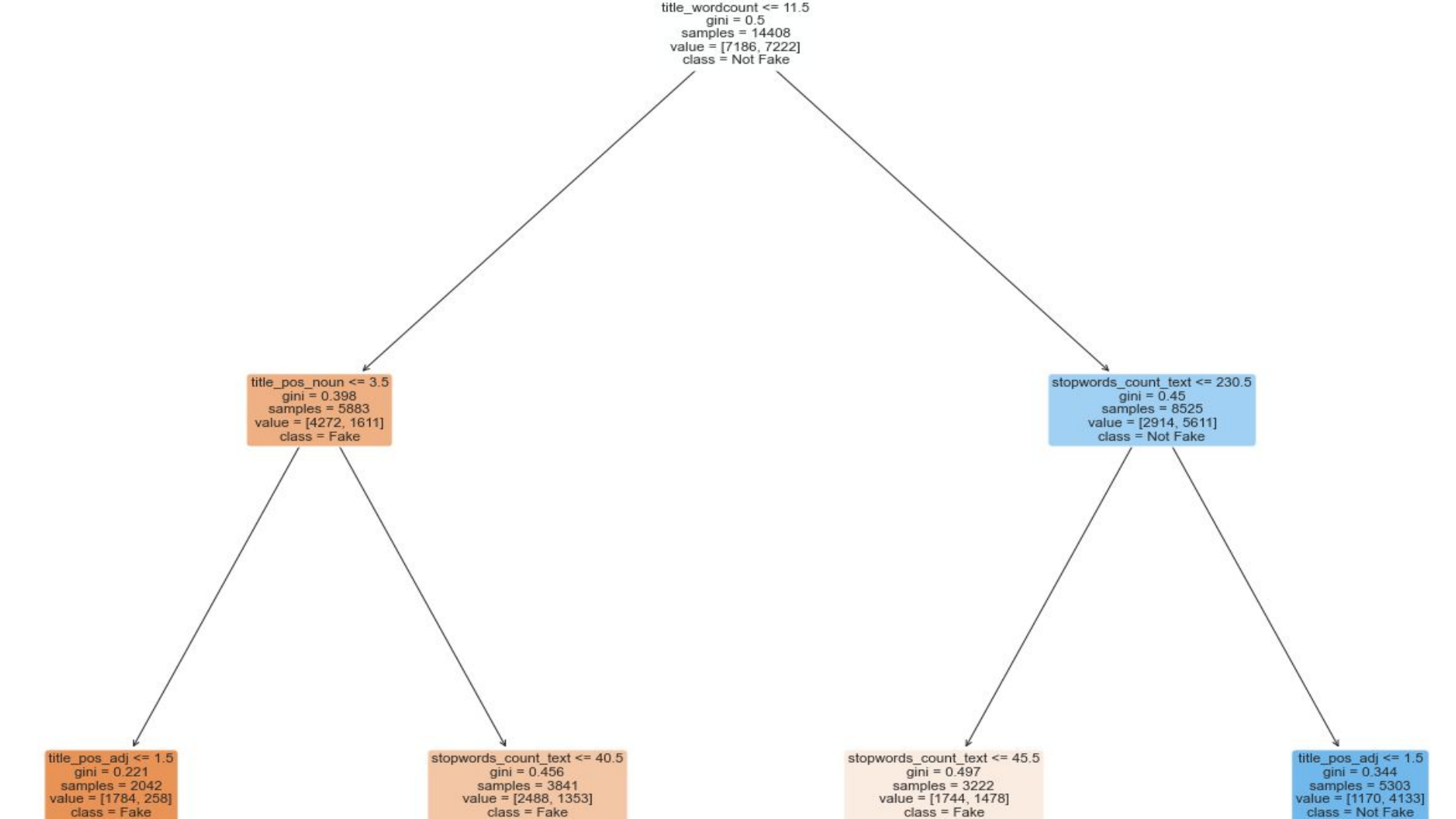
Variables used in Decision Tree

- title_wordcount
- title_pos_adj
- stopwords_count_text
- title_pos_noun
- text_wordcount

Accuracy Level:

- 0.7307318652849741




title_wordcount <= 11.5
gini = 0.5
samples = 14408
value = [7186, 7222]
class = Not Fake
title_pos_noun <= 3.5
gini = 0.398
samples = 5883
value = [4272, 1611]
class = Fake
stopwords_count_text <= 230.5
gini = 0.45
samples = 8525
value = [2914, 5611]
class = Not Fake
title_pos_adj <= 1.5
gini = 0.221
samples = 2042
value = [1784, 258]
class = Fake
stopwords_count_text <= 40.5
gini = 0.456
samples = 3841
value = [2488, 1353]
class = Fake
stopwords_count_text <= 45.5
gini = 0.497
samples = 3222
value = [1744, 1478]
class = Fake
title_pos_adj <= 1.5
gini = 0.344
samples = 5303
value = [1170, 4133]
class = Not Fake

RANDOM FOREST

A classification algorithm that consists of multiple decision trees
N estimator = 100 , Max depth = 4.

Variables used in Random Forest Tree

- title_wordcount
- title_pos_adj
- stopwords_count_text
- title_pos_noun
- text_wordcount

Accuracy Level:

- Accuracy : 0.7362370466321243

RANDOM FOREST

Finding best accuracy by hyperparameter tuning using Cross-Validation(CV)

In [189]:
```
# Define the Hyper-parameter Grid to search on, in case of Random Forest
param_grid = {'n_estimators': np.arange(100,1001,100),   # number of trees 100, 200, ..., 1000
              'max_depth': np.arange(2, 11)}             # depth of trees 2, 3, 4, 5, ..., 10

# Create the Hyper-parameter Grid
hpGrid = GridSearchCV(RandomForestClassifier(),   # the model family
                      param_grid,                 # the search grid
                      cv = 5,                     # 5-fold cross-validation
                      scoring = 'accuracy')       # score to evaluate

# Train the models using Cross-Validation
hpGrid.fit(X_train, y_train.label_translated.ravel())
```

Out[189]:
```
GridSearchCV(cv=5, estimator=RandomForestClassifier(),
             param_grid={'max_depth': array([ 2,  3,  4,  5,  6,  7,  8,  9, 10]),
                         'n_estimators': array([ 100,  200,  300,  400,  500,  600,  700,  800,  900, 1000])},
             scoring='accuracy')
```

In [190]:
```
# Fetch the best Model or the best set of Hyper-parameters
print(hpGrid.best_estimator_)

# Print the score (accuracy) of the best Model after CV
print(np.abs(hpGrid.best_score_))
```

```
RandomForestClassifier(max_depth=9, n_estimators=200)
0.7495843330673264
```

TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY (TF-IDF)

	feature	avg_tfidf
8470	new	0.039548
13075	time	0.037138
14887	york	0.036715
13412	trump	0.027451
1969	breitbart	0.018187
2903	clinton	0.013099
6676	hillari	0.011782
4559	donald	0.010311
4930	elect	0.008235
8604	obama	0.007209

Top 10 features (General - Title)

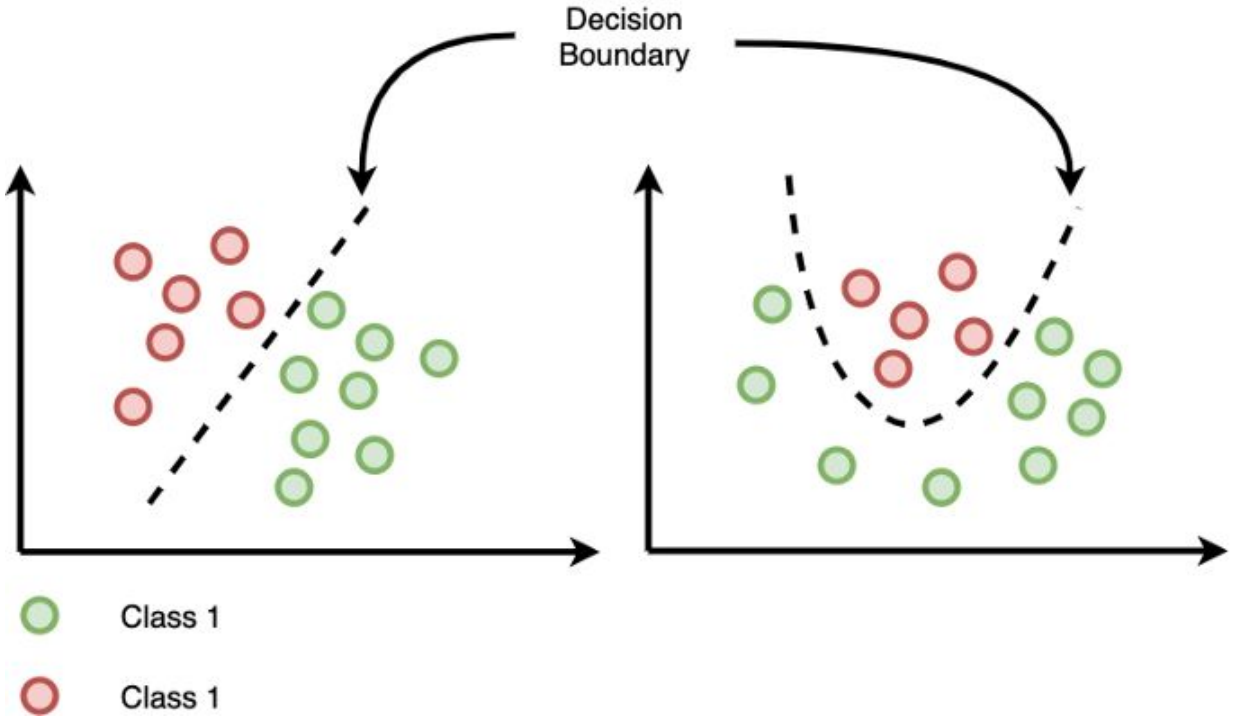
	feature	avg_tfidf
11025	trump	0.024648
1944	clinton	0.018348
4851	hillari	0.017184
3325	elect	0.012156
11635	war	0.009292
7219	new	0.009113
3819	fbi	0.008366
11490	video	0.008356
9212	russia	0.008178
11578	vote	0.007935

Top 10 features (Fake - Title)

	feature	avg_tfidf
6973	new	0.053014
11653	york	0.050753
10497	time	0.050647
10728	trump	0.030606
1289	breitbart	0.028589
3023	donald	0.013441
9057	say	0.008366
1913	clinton	0.007278
7124	obama	0.007020
1308	brief	0.006087

Top 10 features (Not Fake - Title)

LOGISTIC REGRESSION



LOGISTIC REGRESSION

CONVERTING UNSTRUCTURED TEXT (TITLE) INTO TERM FREQUENCY MATRIX

In [200]:

```
# Init tfidvectorizer again for training with different parameters
tfidf = TfidfVectorizer(strip_accents=None,
                        lowercase=False,
                        preprocessor=None,
                        use_idf=True,
                        norm='l2',
                        smooth_idf=True)

# Send all docs to fit
tfidf_train = tfidf.fit_transform(X_train)
tfidf_test = tfidf.transform(X_test)
```

In [201]:

```
# Init logistic regression for k-fold and training
lr = LogisticRegressionCV(cv=5, scoring='accuracy', random_state=0, n_jobs=-1, verbose=3, max_iter=300)
```

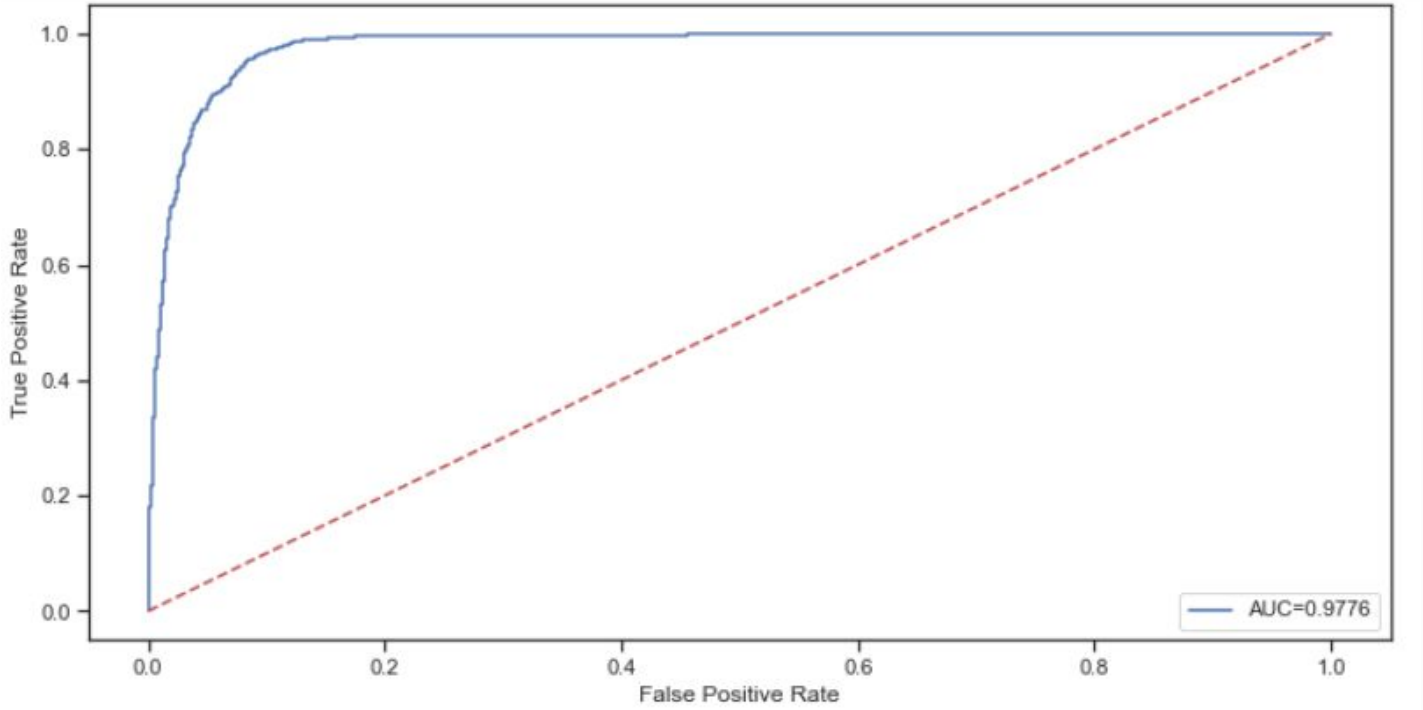
LOGISTIC REGRESSION

ACCURACY (TITLE)

```
Accuracy  :       0.9349040563517124

TPR Test :        0.9592233009708738
TNR Test :        0.9105493437044239

FPR Test :        0.08945065629557608
FNR Test :        0.04077669902912621
```

LOGISTIC REGRESSION

RECEIVER OPERATING CHARACTERISTIC (ROC) CURVE & AREA UNDER CURVE(AUC) FOR TITLE



LOGISTIC REGRESSION

VISUALISING WEIGHTS OF THE MODEL (TITLE)

Weight?	Feature
+9.632	apologist
+8.645	comment
+8.564	hoover
+8.311	hillari
+6.564	honestli
+6.424	hillary
+6.414	discrimin
+6.307	us
+5.811	aleppo
+5.799	deceiv

Top 10 features (Fake-Title)

-9.069	swedish
-9.079	virgil
-9.342	gorka
-9.416	cartel
-9.566	new
-9.718	exclus
-10.460	delingpol
-18.043	time
-46.819	breitbart
-47.347	york

Top 10 features (True-Title)

LOGISTIC REGRESSION

IS TITLE ALONE SUFFICIENT? TRYING OUT DIFFERENT COMBINATIONS OF VARIABLES

Predictor used: Text only
Train Data
Accuracy : 0.9567646344425552

TPR Test : 0.9583945178658835
TNR Test : 0.9551591128254581

FPR Test : 0.044840887174541946
FNR Test : 0.041605482134116495

	precision	recall	f1-score	support
0	0.96	0.96	0.96	2074
1	0.95	0.96	0.96	2043
accuracy			0.96	4117
macro avg	0.96	0.96	0.96	4117
weighted avg	0.96	0.96	0.96	4117

Text as Predictor

Predictor used: Title & text
Train Data
Accuracy : 0.9640514938061695

TPR Test : 0.9668316831683168
TNR Test : 0.9613733905579399

FPR Test : 0.03862660944206009
FNR Test : 0.03316831683168317

	precision	recall	f1-score	support
0	0.97	0.96	0.96	2097
1	0.96	0.97	0.96	2020
accuracy			0.96	4117
macro avg	0.96	0.96	0.96	4117
weighted avg	0.96	0.96	0.96	4117

Title & Text as Predictor

LOGISTIC REGRESSION


```
Predictor used: Author & title
Train Data
Accuracy  :      0.9941705125091086

TPR Test :       0.9932627526467758
TNR Test :       0.9950956351152526

FPR Test :       0.004904364884747425
FNR Test :       0.006737247353224254
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      2039
           1       1.00      0.99      0.99      2078

    accuracy                           0.99      4117
   macro avg       0.99      0.99      0.99      4117
weighted avg       0.99      0.99      0.99      4117
```

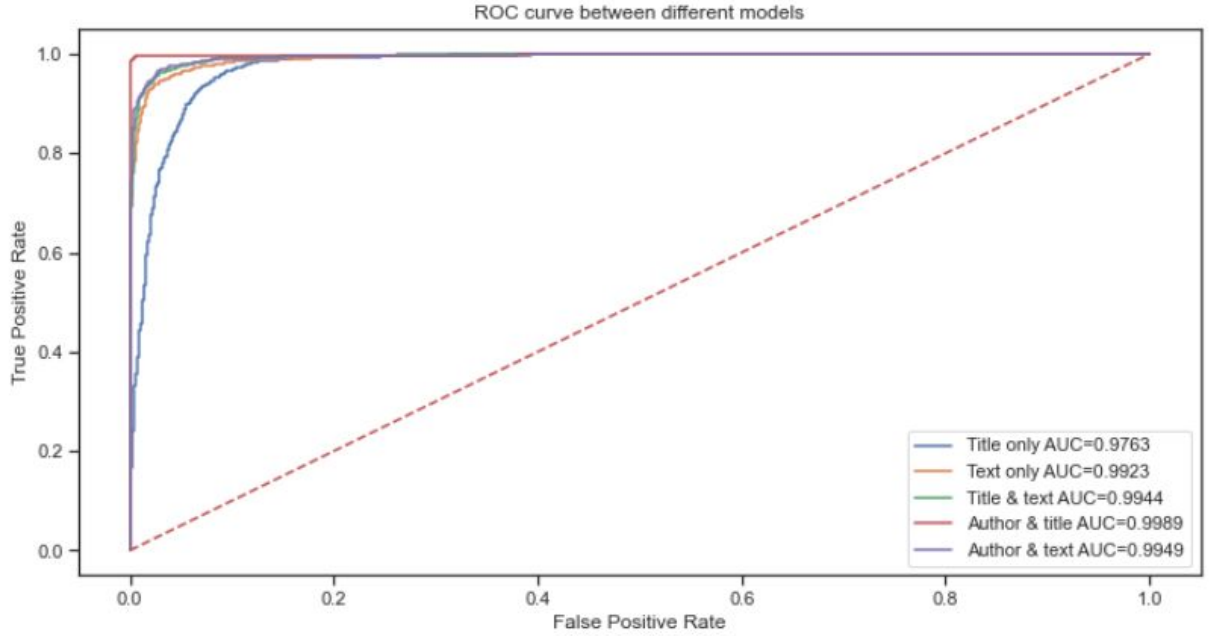
Author & Title as Predictors

```
Predictor used: Author & text
Train Data
Accuracy  :      0.96939519067282

TPR Test :       0.9665048543689321
TNR Test :       0.9722897423432183

FPR Test :       0.02771025765678172
FNR Test :       0.03349514563106796
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      2057
           1       0.97      0.97      0.97      2060

    accuracy                           0.97      4117
   macro avg       0.97      0.97      0.97      4117
weighted avg       0.97      0.97      0.97      4117
```

Author & Text as Predictors

LOGISTIC REGRESSION

ROC AND AUC CURVE FOR THE DIFFERENT MODELS



Insights & Recommendations

INSIGHTS & RECOMMENDATIONS

- Polarity & emotions does not have a strong relation to fake news. Therefore, it is not a good indicator of fake news.
- Instead, indicators such as title word count, title adjective count, and text stopwords count are the best indicators to fake news
- Based on Attempt 3, detection of fake news using title is sufficient. However, for the best results, author & title are required.
- Out of all the 3 models we implemented, decision tree performed the worst while logistic regression performed the best.
- We can suggest that from a reader’s perspective in identifying fake news, author is a quick and credible identifier.
- The title could further support a reader’s attempt in identifying fake news.

Thank You