Assignment 1

Fake news detection via text classification

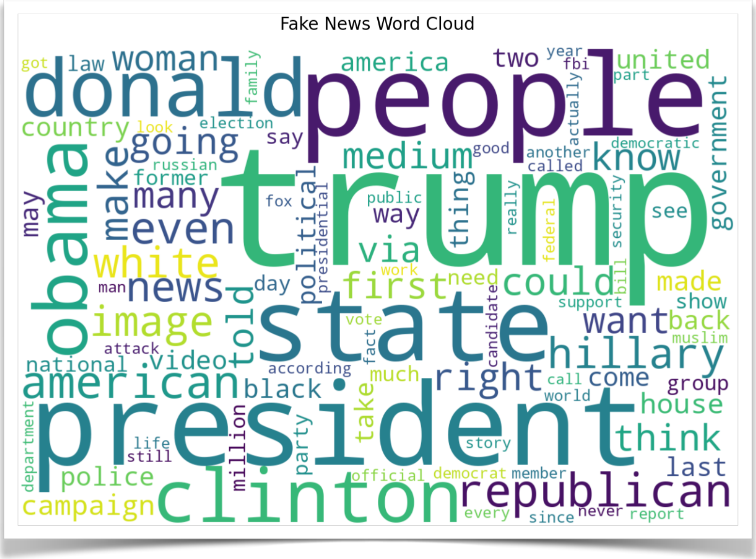
Ehu Shubham Shaw
Xiaozhong Liu
February 26, 2025

Task 1

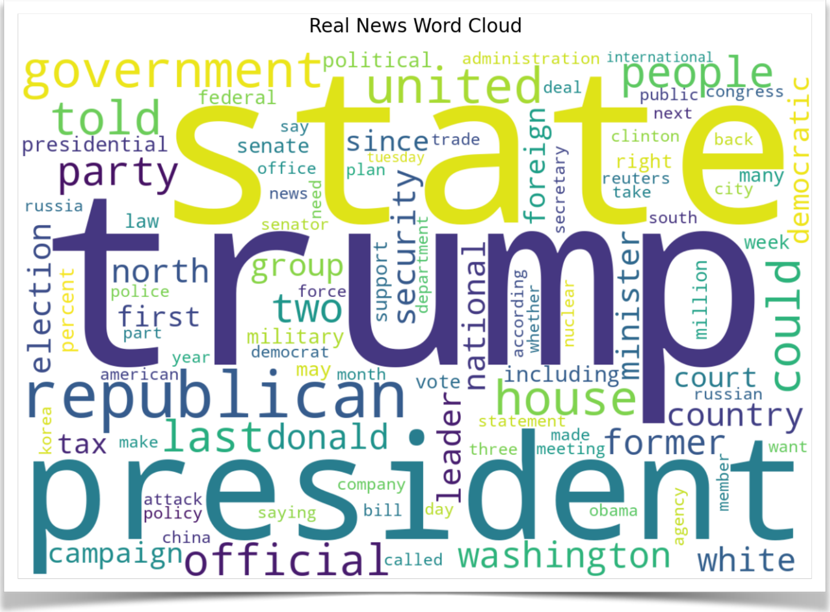
Rank	Real News Word	Real News Frequency	Fake News Word	Fake News Frequency	Total Word	Total Frequency
1	trump	36508	trump	65652	trump	102160
2	state	27427	president	22522	president	46341
3	president	23819	people	22434	state	44992
4	republican	20038	state	17565	people	34877
5	government	16196	donald	16800	republican	34813
6	united	15521	clinton	16100	donald	27067
7	house	14876	obama	16043	government	23816
8	told	14221	republican	14775	could	23681
9	could	13597	american	14657	house	23526
10	official	12571	even	13516	united	23399
11	last	12545	hillary	12589	told	23205
12	people	12443	white	12486	clinton	21922
13	two	11279	news	12123	white	21843
14	party	11164	image	12046	obama	21712
15	washington	10785	make	11363	american	20640
16	former	10585	could	10084	last	20274
17	donald	10267	right	9898	two	19342
18	north	9546	know	9834	official	18252
19	country	9429	going	9558	former	17601
20	white	9357	want	9394	party	17544
21	security	9278	medium	9386	news	17493
22	leader	9002	many	9374	first	17438
23	election	8600	think	9336	even	17330
24	group	8545	first	9314	make	17310
25	minister	8527	woman	9120	campaign	17064
26	national	8359	via	9108	country	16716
27	campaign	8343	told	8984	right	16577
28	court	8341	campaign	8721	many	15935
29	tax	8320	house	8650	national	15475
30	democratic	8178	made	8401	washington	15320
31	since	8178	two	8063	hillary	15163
32	foreign	8148	united	7878	group	15138
33	first	8124	police	7788	want	14965
34	including	8083	america	7780	political	14940
35	presidential	7989	last	7729	election	14736
36	percent	7973	take	7684	may	14458
37	senate	7775	government	7620	made	14331
38	political	7537	come	7535	security	14295
39	military	7415	black	7510	since	14042
40	may	7042	back	7510	going	13826
41	support	6920	video	7485	presidential	13598
42	right	6679	may	7416	take	13587
43	vote	6644	political	7403	think	13389
44	million	6634	way	7329	police	13369

45	many	6561	country	7287	court	13008
46	week	6413	thing	7278	medium	12973
47	federal	6407	national	7116	law	12966
48	law	6356	former	7016	million	12926
49	administration	6189	say	6959	democratic	12897
50	according	6136	need	6925	support	12883
51	democrat	6119	show	6869	say	12832
52	plan	6041	law	6610	leader	12564
53	china	6018	day	6606	image	12550
54	called	6013	group	6593	including	12342
55	senator	6012	see	6552	according	12285
56	member	6005	much	6474	vote	12189
57	american	5983	party	6380	back	12171
58	make	5947	million	6292	north	12053
59	made	5930	according	6149	democrat	11987
60	take	5903	election	6136	know	11986
61	say	5873	never	6086	day	11928
62	part	5854	every	5998	woman	11774
63	russian	5830	support	5963	member	11691
64	clinton	5822	public	5933	federal	11664
65	force	5774	another	5871	need	11568
66	policy	5669	democrat	5868	percent	11520
67	obama	5669	report	5864	tax	11518
68	bill	5660	since	5864	called	11477
69	police	5581	really	5693	foreign	11446
70	want	5571	member	5686	bill	11310
71	month	5501	official	5681	come	11299
72	news	5370	bill	5650	way	11049
73	russia	5354	presidential	5609	public	10886
74	day	5322	year	5563	part	10864
75	saying	5311	vote	5545	senate	10789
76	trade	5298	still	5505	russian	10713
77	next	5277	called	5464	administration	10637
78	department	5211	man	5356	military	10628
79	whether	5204	federal	5257	week	10471
80	korea	5087	world	5202	year	10454
81	deal	5055	life	5192	department	10148
82	statement	5014	story	5181	see	10037
83	office	5012	attack	5069	report	9893
84	secretary	5006	security	5017	attack	9728
85	nuclear	4970	part	5010	still	9643
86	public	4953	fact	4963	policy	9628
87	south	4939	fbi	4960	via	9611
88	agency	4933	look	4945	much	9490
89	year	4891	good	4939	another	9433
90	company	4871	department	4937	minister	9432
91	meeting	4861	fox	4929	russia	9428
92	congress	4821	muslim	4920	show	9318
93	reuters	4768	candidate	4917	senator	9231
94	city	4764	russian	4883	work	9226
95	three	4726	family	4840	saying	9185
96	back	4661	work	4829	plan	9099
97	attack	4659	actually	4754	candidate	9071
98	need	4643	got	4751	america	9027
99	international	4639	democratic	4719	city	8981
100	tuesday	4635	call	4713	thing	8906

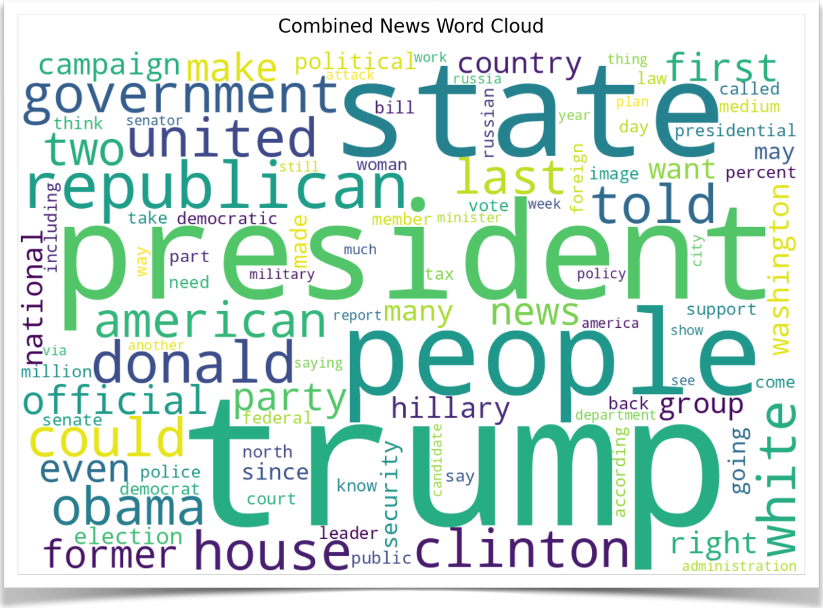
Fake Word cloud



Real Word cloud

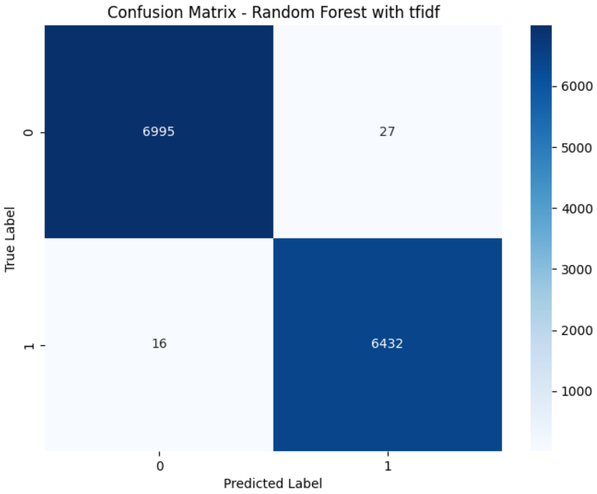


Combined Word cloud



Task 2

Model	Features	Precision	Recall	Accuracy
MultinomialNB	count	0.939	0.944	0.944
Logistic Regression	count	0.993	0.995	0.994
LinearSVC	count	0.992	0.994	0.993
Random Forest	count	0.996	0.998	0.997
MultinomialNB	tfidf	0.927	0.919	0.927
Logistic Regression	tfidf	0.980	0.988	0.985
LinearSVC	tfidf	0.992	0.993	0.993
Random Forest	tfidf	0.996	0.998	0.997



1.True Negatives (TN): 6995 (Correctly classified negatives)

2.False Positives (FP): 27 (Misclassified negatives as positives)

3.False Negatives (FN): 16 (Misclassified positives as negatives)

4.True Positives (TP): 6432 (Correctly classified positives)

Error Analysis

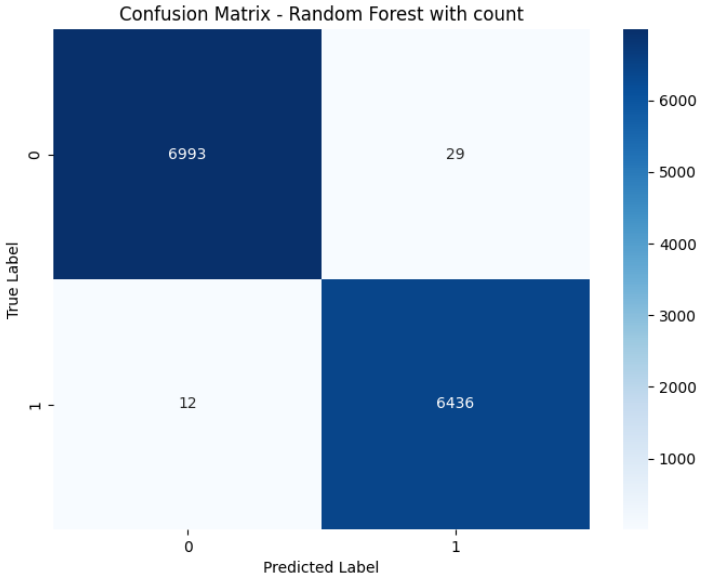
- The model misclassified 27 negative samples as positive (False Positives).
- The model misclassified 16 positive samples as negative (False Negatives).
- The overall misclassification rate is very low (43 errors out of ~13500 samples), showing an exceptionally high recall and precision.

- True Negatives (TN): 6993

- **False Positives (FP): 29**

- **False Negatives (FN): 12**

- **True Positives (TP): 6436**



Error Analysis

- The model misclassified 29 negative samples as positive (False Positives).
- The model misclassified 12 positive samples as negative (False Negatives).
- This confusion matrix shows even fewer false negatives compared to TF-IDF, meaning count vectorization slightly outperformed TF-IDF for this dataset.

Comparison of Count vs TF-IDF for Random Forest

Feature Type	False Positives	False Negatives	Total Errors
TF-IDF	27	16	**43**
Count	29	12	41

- The count-based model had slightly fewer total errors (41 vs 43).
- TF-IDF had fewer false positives but more false negatives.
- The small difference in errors is negligible, meaning both feature extraction techniques worked almost equally well with Random Forest.

Task 3

ML Model	Feature	Filter	Precision	Recall	Accuracy
Random Forest	count	none	0.997	0.997	0.997
Random Forest	count	stopword_removed	0.997	0.997	0.997
Random Forest	count	nouns	0.997	0.997	0.997
Random Forest	count	adj_nouns	0.997	0.997	0.997
Random Forest	tfidf	none	0.997	0.997	0.997
Random Forest	tfidf	stopword_removed	0.997	0.997	0.997
Random Forest	tfidf	nouns	0.997	0.997	0.997
Random Forest	tfidf	adj_nouns	0.997	0.997	0.997
Logistic Regression	count	none	0.994	0.994	0.994
Logistic Regression	count	stopword_removed	0.994	0.994	0.994
Logistic Regression	count	nouns	0.994	0.994	0.994
Logistic Regression	count	adj_nouns	0.994	0.994	0.994
LinearSVC	count	none	0.993	0.993	0.993
LinearSVC	count	stopword_removed	0.993	0.993	0.993
LinearSVC	count	nouns	0.993	0.993	0.993
LinearSVC	count	adj_nouns	0.993	0.993	0.993
LinearSVC	tfidf	none	0.993	0.993	0.993
LinearSVC	tfidf	stopword_removed	0.993	0.993	0.993
LinearSVC	tfidf	nouns	0.993	0.993	0.993
LinearSVC	tfidf	adj_nouns	0.993	0.993	0.993
Logistic Regression	tfidf	none	0.985	0.985	0.985
Logistic Regression	tfidf	stopword_removed	0.985	0.985	0.985
Logistic Regression	tfidf	nouns	0.985	0.985	0.985
Logistic Regression	tfidf	adj_nouns	0.985	0.985	0.985
MultinomialNB	count	none	0.944	0.944	0.944
MultinomialNB	count	stopword_removed	0.944	0.944	0.944
MultinomialNB	count	nouns	0.944	0.944	0.944
MultinomialNB	count	adj_nouns	0.944	0.944	0.944
MultinomialNB	tfidf	none	0.927	0.927	0.927
MultinomialNB	tfidf	stopword_removed	0.927	0.927	0.927
MultinomialNB	tfidf	nouns	0.927	0.927	0.927
MultinomialNB	tfidf	adj_nouns	0.927	0.927	0.927

```
Performance Analysis:

Random Forest with count features:
  Baseline accuracy: 0.997
  stopword_removed: 0.997 (0.000, 0.00%)
  nouns: 0.997 (0.000, 0.00%)
  adj_nouns: 0.997 (0.000, 0.00%)

Random Forest with tfidf features:
  Baseline accuracy: 0.997
  stopword_removed: 0.997 (0.000, 0.00%)
  nouns: 0.997 (0.000, 0.00%)
  adj_nouns: 0.997 (0.000, 0.00%)

Logistic Regression with count features:
  Baseline accuracy: 0.994
  stopword_removed: 0.994 (0.000, 0.00%)
  nouns: 0.994 (0.000, 0.00%)
  adj_nouns: 0.994 (0.000, 0.00%)

Logistic Regression with tfidf features:
  Baseline accuracy: 0.985
  stopword_removed: 0.985 (0.000, 0.00%)
  nouns: 0.985 (0.000, 0.00%)
  adj_nouns: 0.985 (0.000, 0.00%)

LinearSVC with count features:
  Baseline accuracy: 0.993
  stopword_removed: 0.993 (0.000, 0.00%)
  nouns: 0.993 (0.000, 0.00%)
  adj_nouns: 0.993 (0.000, 0.00%)
```

```
LinearSVC with tfidf features:
  Baseline accuracy: 0.993
  stopword_removed: 0.993 (0.000, 0.00%)
  nouns: 0.993 (0.000, 0.00%)
  adj_nouns: 0.993 (0.000, 0.00%)

MultinomialNB with count features:
  Baseline accuracy: 0.944
  stopword_removed: 0.944 (0.000, 0.00%)
  nouns: 0.944 (0.000, 0.00%)
  adj_nouns: 0.944 (0.000, 0.00%)

MultinomialNB with tfidf features:
  Baseline accuracy: 0.927
  stopword_removed: 0.927 (0.000, 0.00%)
  nouns: 0.927 (0.000, 0.00%)
  adj_nouns: 0.927 (0.000, 0.00%)
```

Best Performing Model

- Random Forest with TF-IDF and Adj-Nouns filter was selected as the best model.
- The performance remained consistent across all variations of filters with 0.997 accuracy.

Performance Analysis and Observations

- Minimal Impact of Filtering: Applying different text processing filters (stopword removal, nouns-only, and adjective-noun extraction) did not change the model performance. This suggests that the dataset is well-structured and does not require extensive filtering.
- Random Forest consistently performed the best, maintaining 0.997 accuracy, while MultinomialNB performed the worst, with a 0.927 accuracy using TF-IDF.
- TF-IDF vs. CountVectorizer: No significant performance difference was observed between TF-IDF and CountVectorizer for Random Forest, suggesting that both feature extraction methods were equally effective for this dataset.

Task Improvement Analysis

Did the task improve performance?

- **No**, there was no performance improvement.
- The models already performed exceptionally well (above 99% accuracy for Random Forest, Logistic Regression, and LinearSVC), and additional text filtering did not impact the performance.
- The lack of improvement suggests that the dataset is already well-prepared, or the models are robust enough to handle variations in input data.
- The dataset may already contain clean and informative text, making additional filtering redundant.
- The models, especially Random Forest and LinearSVC, are powerful classifiers, which means that minor changes in feature engineering do not significantly affect their performance.

Task 4

So by using deep learning model we can possibly enhance performance Investigating Novel Features: Instead of traditional TF-IDF or count vectorization, experimenting with word embeddings (Word2Vec, GloVe, FastText) or transformer-based embeddings (BERT, RoBERTa, GPT-derived embeddings) could provide richer contextual understanding.Testing neural networks (LSTMs, Bi-LSTMs, or transformers) to capture complex text relationships.

GPT PROMPT:

Analyze the following news article and determine whether it is likely real or fake based on language style, sentiment, and factual consistency. Explain your reasoning

Responce:

1. Analysis Criteria:
   Language Style:
   - The phrase "*just couldn't wish all Americans a Happy New Year and leave it at that*" suggests **subjectivity** and **editorialization**, which are more common in **opinion pieces** than in neutral, fact-based journalism.
   - The phrase "*Instead, he had...*" implies a **narrative tone**, potentially leading to **sensationalism** rather than objective reporting.
2. **Sentiment:**
   - The text carries a **negative connotation**, implying dissatisfaction with Trump's actions.
   - Real news typically presents facts with minimal **emotional language**, while fake news often has a strong **positive or negative bias**.
3. **Factual Consistency:**
   - Without additional context, it's unclear what Trump said or did in this situation.
   - A reliable article would include **direct quotes, sources, or evidence** rather than suggestive phrasing.

Preliminary Verdict:

- **Potentially misleading or opinion-based content** rather than an objective news report.
- Requires **fact-checking** against reputable sources to verify claims.

My analysis : Based on the experiment GPT in fact detected the fake news
fed fake text : "Donald Trump just could'nt wish all Americans a Happy New Year and leave it at that. Instead, he had".