

FAKE NEWS DETECTION



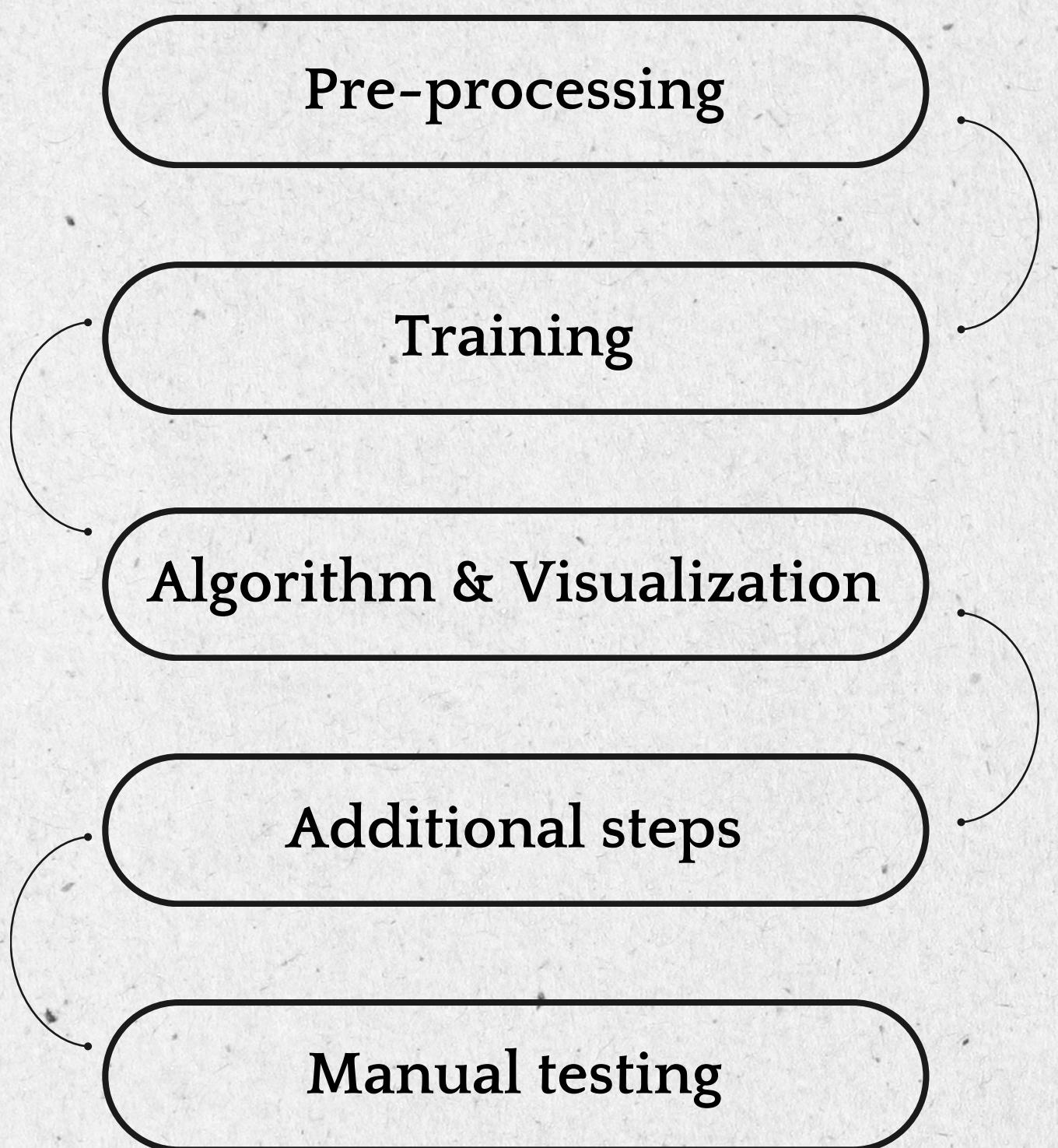
Problem Identification



- Exists in every society.
- Catalyst: Technology (social media) and irresponsible journalism parties.
- Reign over people's actions & perception
- Controlled by a number of people behind the screen.

DISCLAIMER

Work Flow



Libraries



[51] ✓ 0.0s

	title	text	subject	date
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016
23479	How to Blow \$700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016

[52] ✓ 0.0s

	title	text	subject	date
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of I...	worldnews	August 22, 2017
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017
21416	Indonesia to buy \$1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017



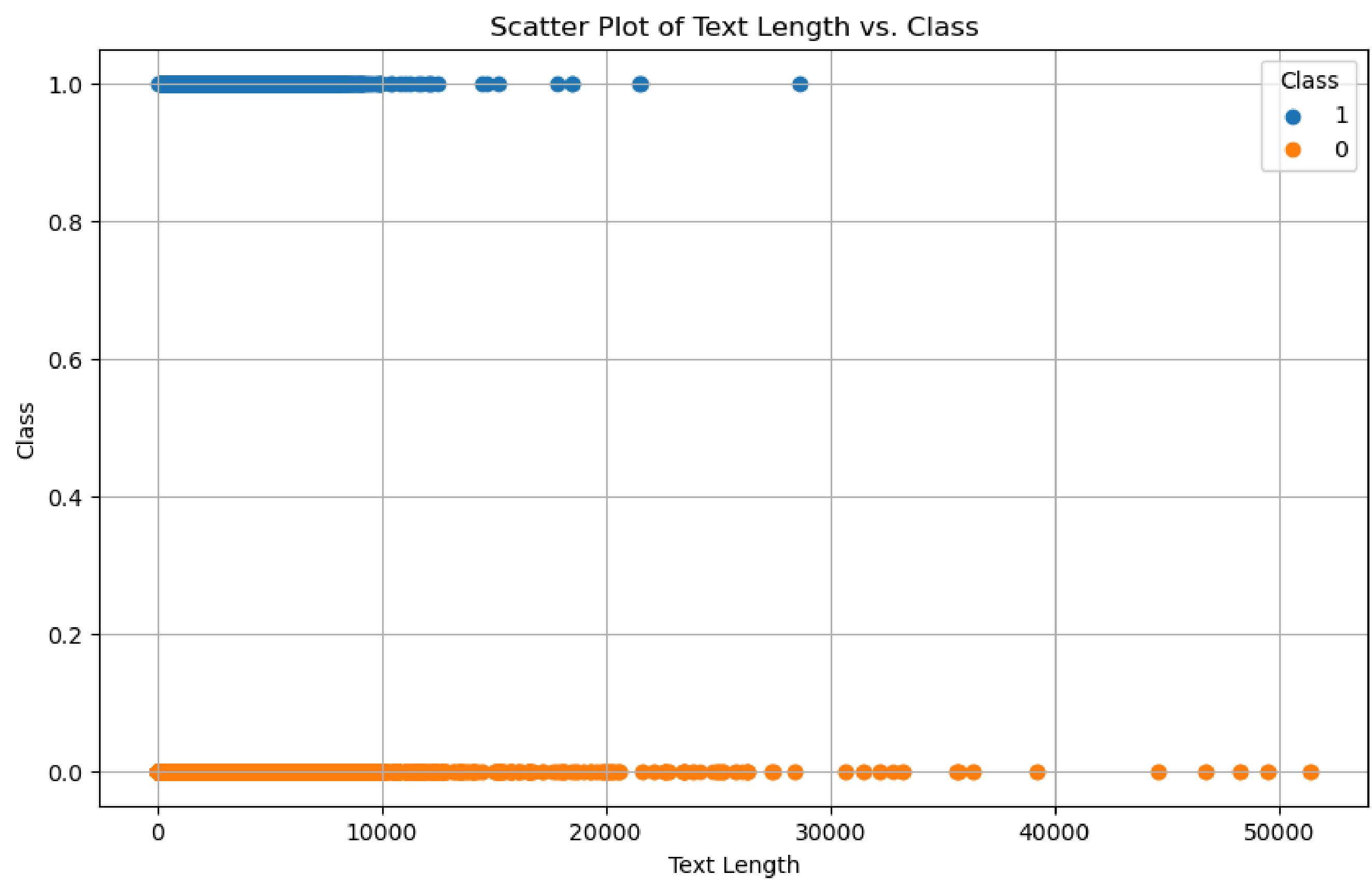
```
1 def clean_text(text):
2     text = text.lower()
3     text = re.sub(r"\[.*?\]", "", text)
4     text = "".join(char if char.isalnum() or char.isspace() else " " for char in text)
5     text = re.sub(r"https?://\S+|www\.\S+", "", text)
6     text = re.sub(r"<.*?>", "", text)
7     text = "".join(char for char in text if char not in string.punctuation)
8     text = re.sub(r"\w*\d\w*", "", text)
9
10    return text
```



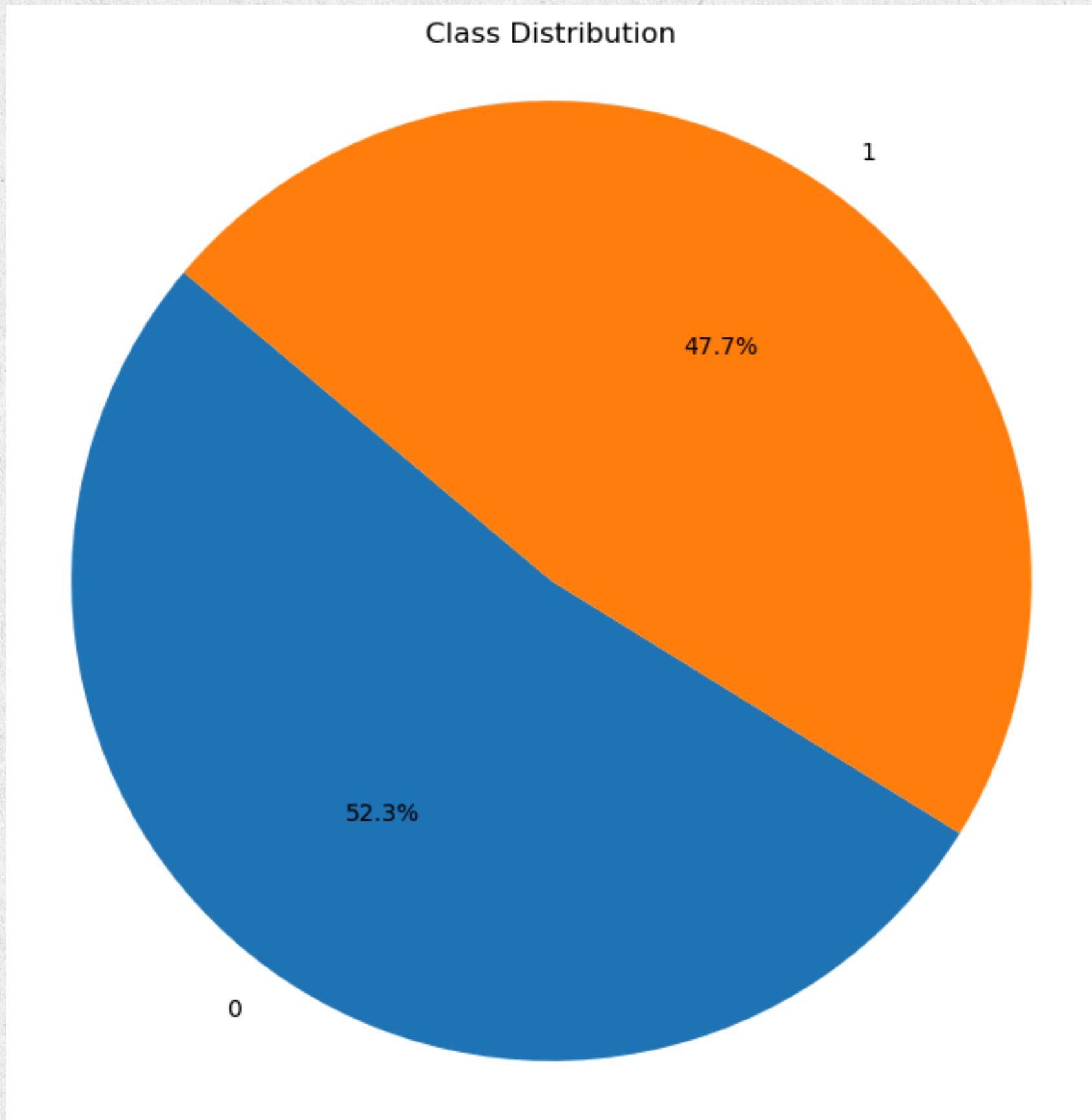
```
1 data_fake["class"] = 0
2 data_true["class"] = 1
3
4 data_merge = pd.concat([data_fake,data_true], axis = 0)
5 data = data_merge.drop(['title', 'subject', 'date'], axis = 1)
6
7 data = data.sample(frac=1).reset_index(drop=True)
8
9 data['text'] = data['text'].apply(clean_text)
```

	text	class
0	WASHINGTON (Reuters) - Lawmakers in the U.S. H...	1
1	Jay Dyer 21st Century WireIn the famous treati...	0
2	Dividing America will be Obama s legacy. Hilla...	0
3	DANANG, Vietnam (Reuters) - The remaining 11 c...	1
4	Liberals like to use the term right wing extr...	0
5	Here is the FULL Episode #9 of this podcast ...	0
6	BAMAKO (Reuters) - Rival Tuareg groups said on...	1
7	Apparently breaking the law and scamming the g...	0
8	WASHINGTON (Reuters) - President Donald Trump ...	1
9	WASHINGTON (Reuters) - The heads of the U.S. a...	1
10	If there s anyone who is absolutely tired of D...	0
11	On Tuesday night, Ted Cruz dropped out of the ...	0
12	In September 2015, the Sterling Heights, MI Ci...	0
13	RIYADH (Reuters) - Britain and Saudi Arabia si...	1
14	LONDON (Reuters) - Britain's ambassador to the...	1

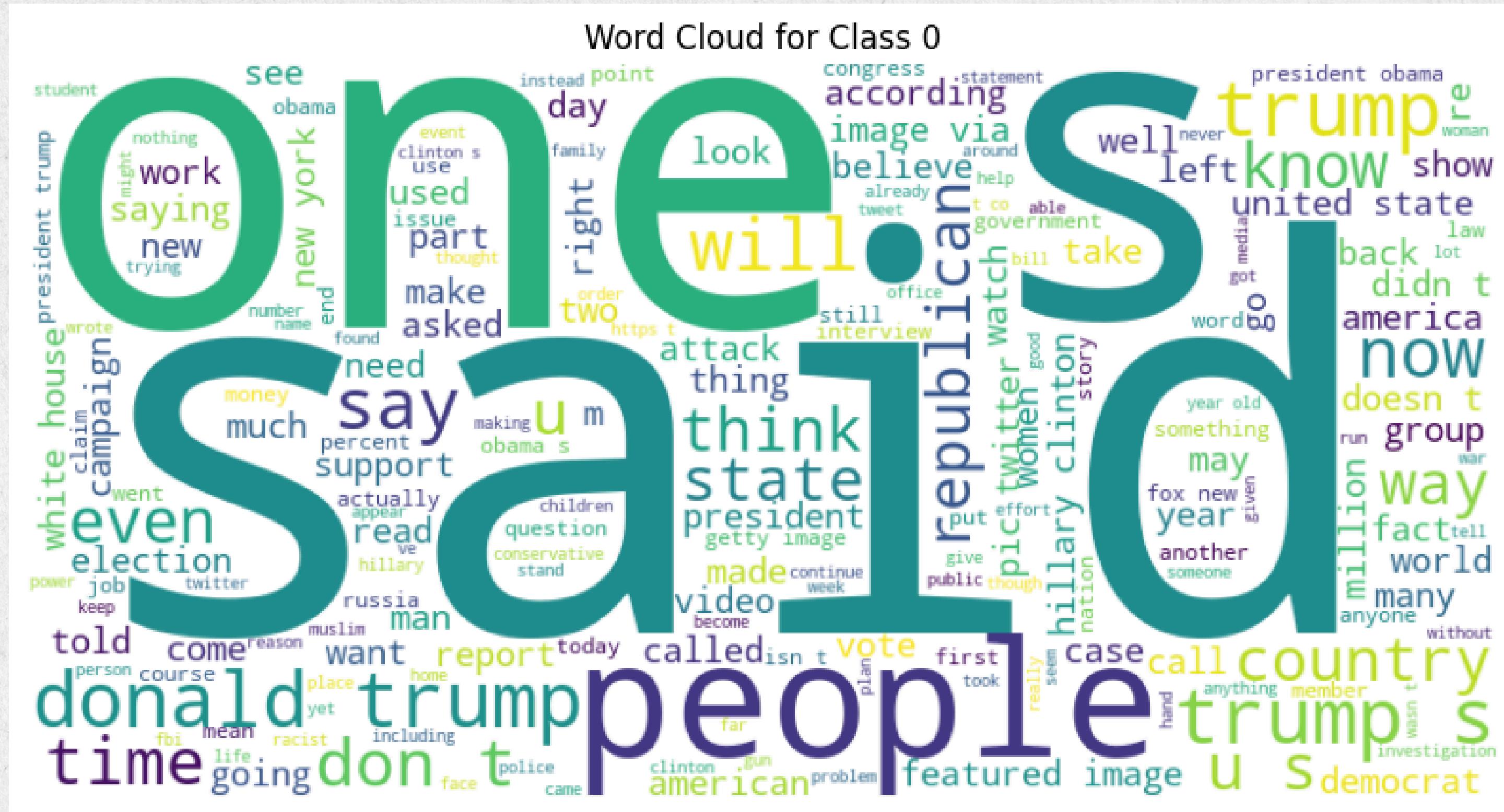
Exploratory Data Analysis



Exploratory Data Analysis



Exploratory Data Analysis



Exploratory Data Analysis

```
print(xv_test)
```

```
(0, 93651) 0.024460258258940725
(0, 92675) 0.01997646157663912
(0, 92583) 0.03912656826239789
(0, 92569) 0.03389733781614474
(0, 92567) 0.04512913168156637
(0, 92298) 0.07111958417021776
...
(0, 100000) 0.015805302478440773
```

```
print(xv_train)
```

```
(0, 81097) 0.041548848569729416
(0, 1469) 0.019343159125811377
(0, 26017) 0.020407452899510542
(0, 18424) 0.020758269858817173
(0, 597) 0.05533549320376825
(0, 24244) 0.049297667186769104
(0, 44091) 0.015805302478440773
```



```
1 X = data['text']
2 y = data['class']
3
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=101)
5
6 vectorization = TfidfVectorizer()
7 xv_train = vectorization.fit_transform(X_train)
8 xv_test = vectorization.transform(X_test)
```



```
1 LR = LogisticRegression()  
2 LR.fit(xv_train, y_train)  
3 pred_lr = LR.predict(xv_test)  
4 LR.score(xv_test, y_test)
```



```
1 DT = DecisionTreeClassifier()  
2 DT.fit(xv_train, y_train)  
3 pred_dt = DT.predict(xv_test)  
4 DT.score(xv_test, y_test)
```

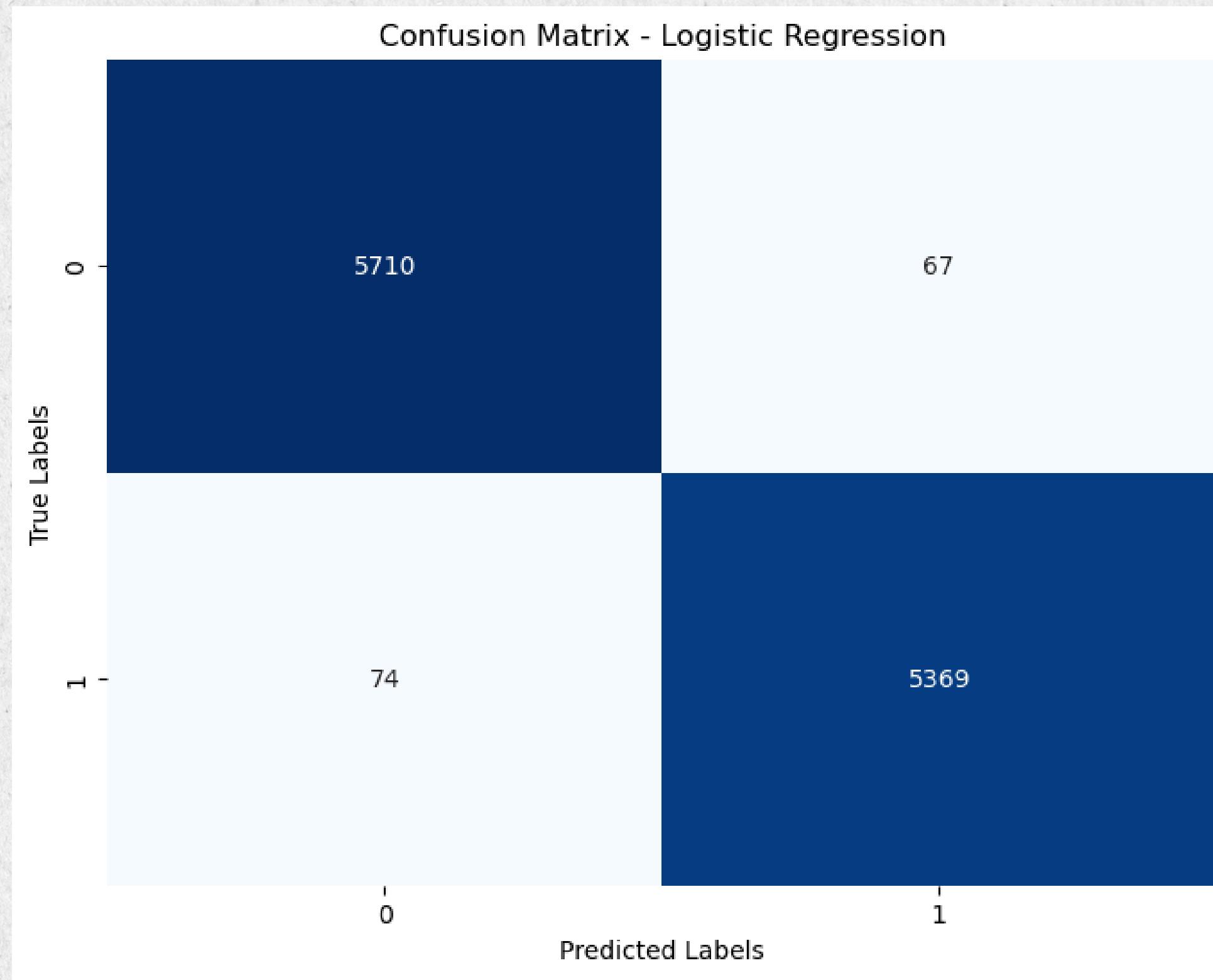


```
1 GB = GradientBoostingClassifier(  
    random_state=0)  
2 GB.fit(xv_train, y_train)  
3 pred_gb = GB.predict(xv_test)  
4 GB.score(xv_test, y_test)
```



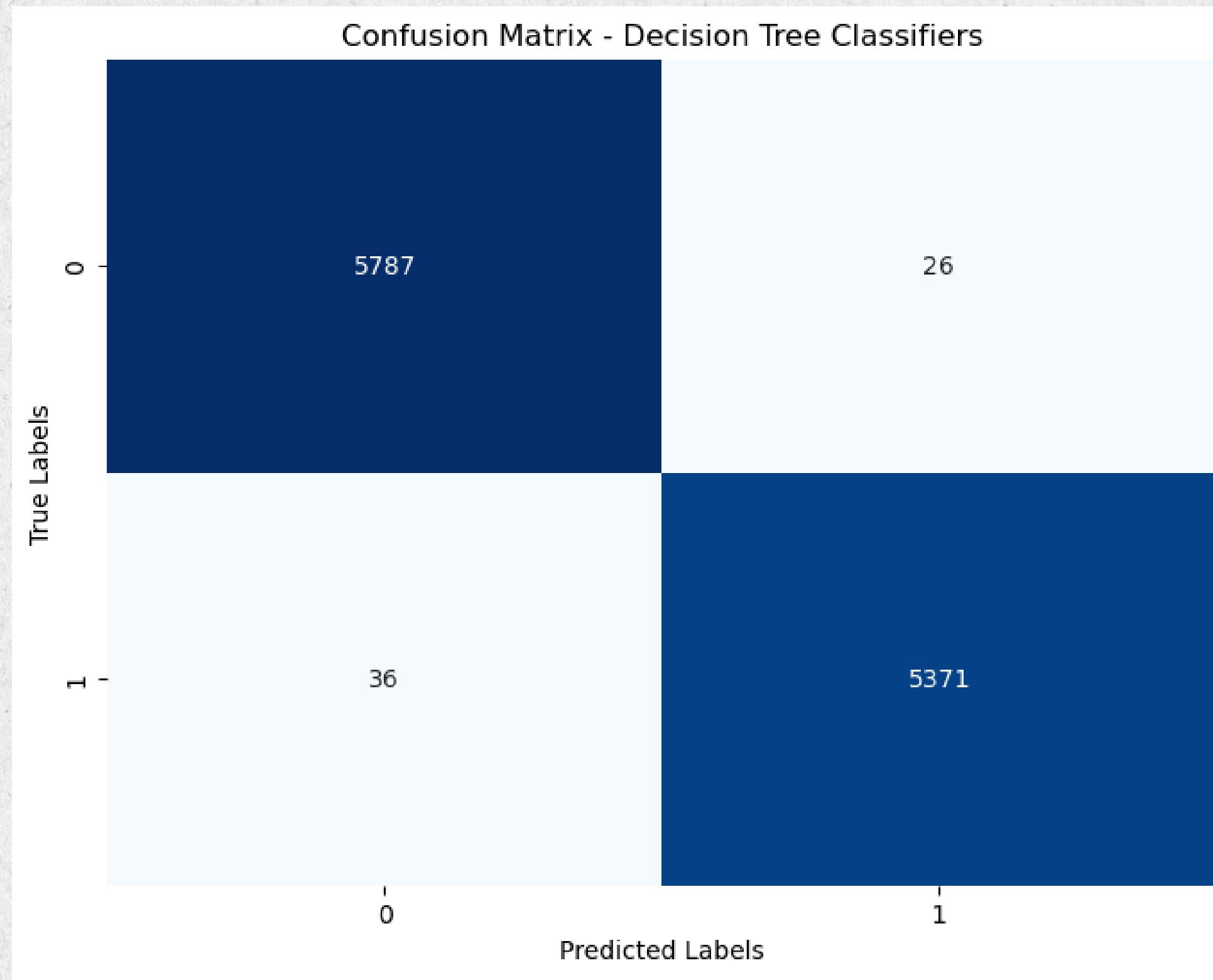
```
1 RF = RandomForestClassifier(random  
    _state=0)  
2 RF.fit(xv_train, y_train)  
3 pred_rf = RF.predict(xv_test)  
4 RF.score(xv_test, y_test)
```

Visualization



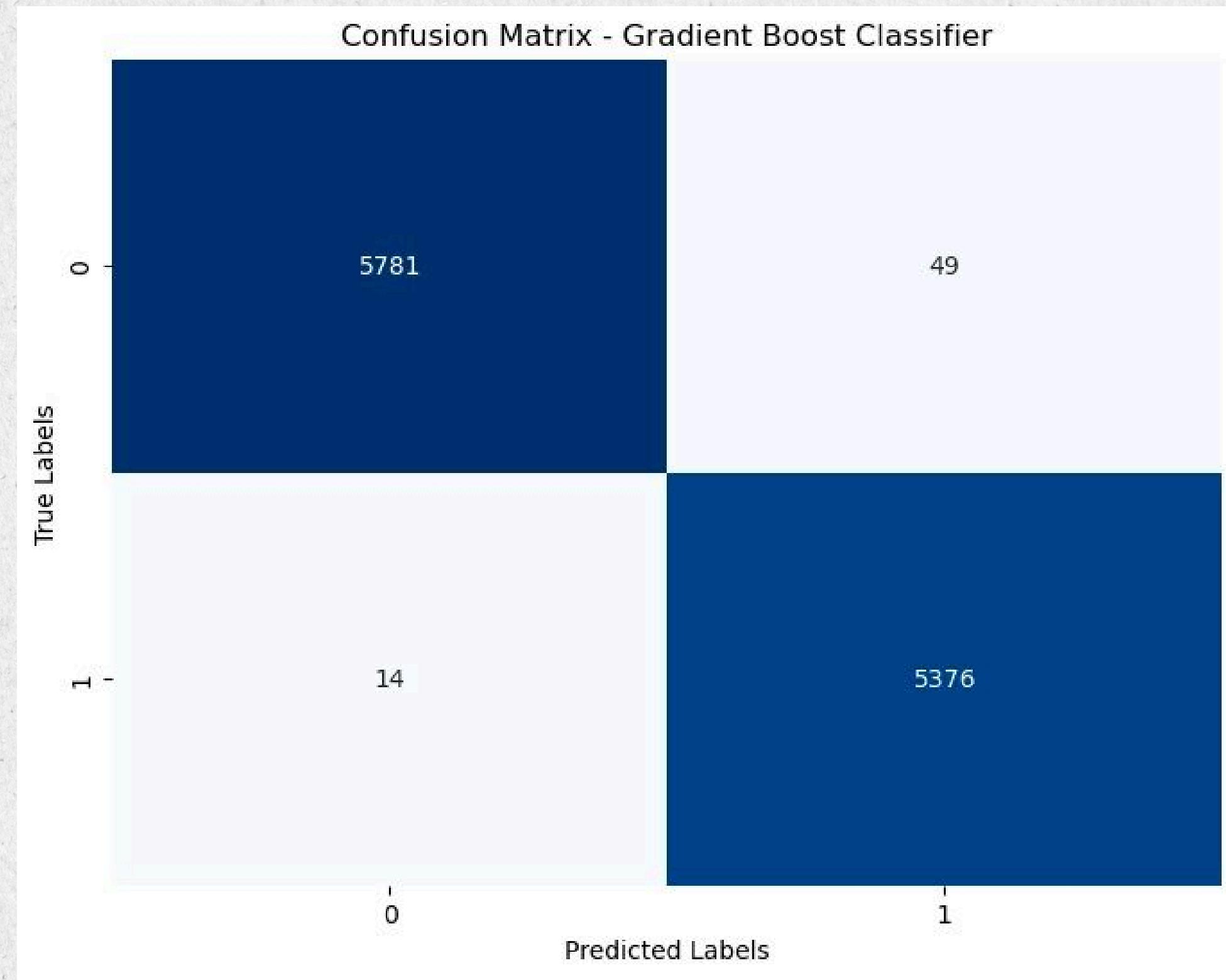
Accuracy: 0.9857397504456328

Visualization



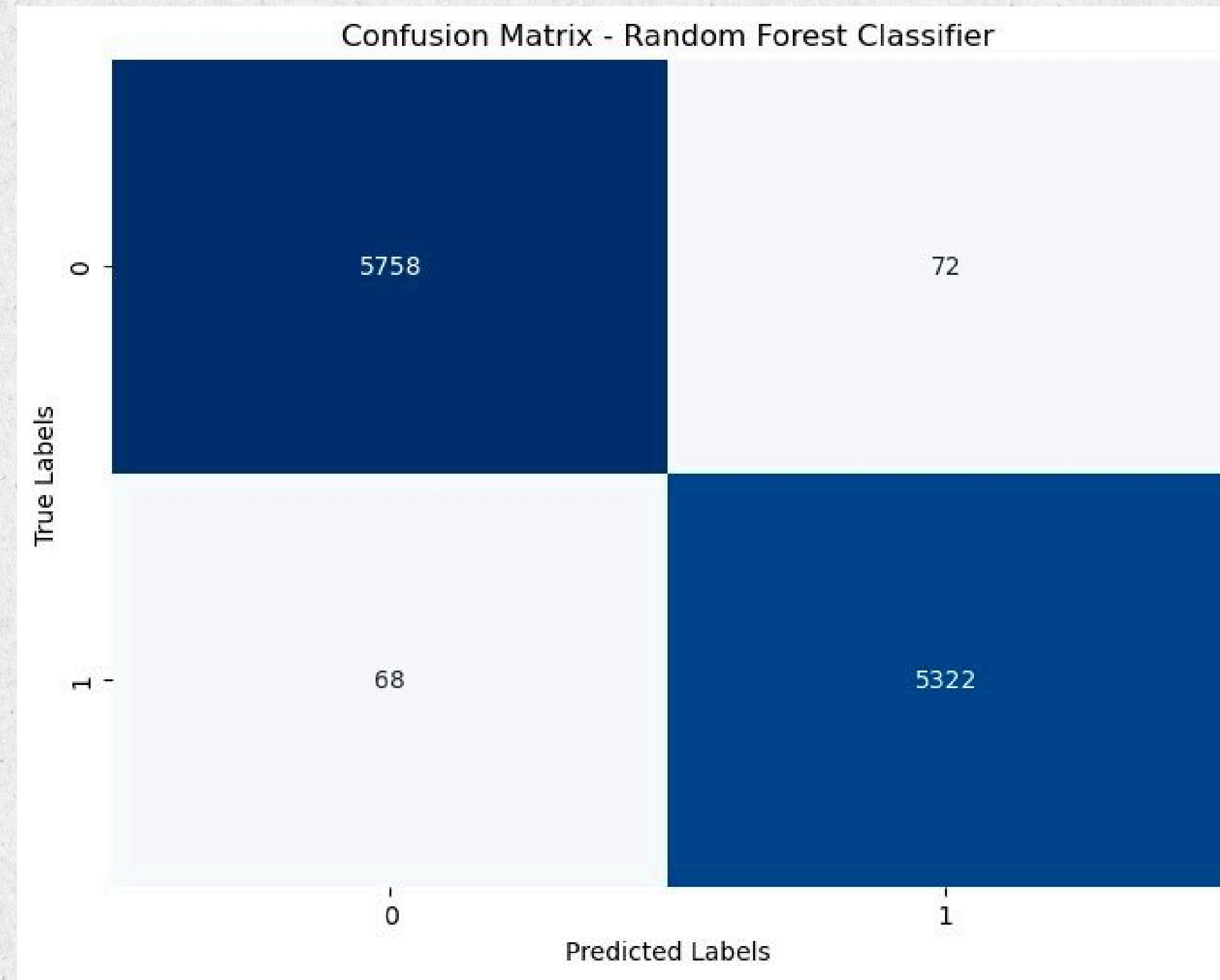
Accuracy: 0.9961675579322639

Visualization



Accuracy: 0.9942959001782531

Visualization



Accuracy: 0.9852941176470589



```
1 def evaluate_model(model, X_train, y_train, k=5, stratified=False):
2     if stratified:
3         kf = StratifiedKFold(n_splits=k)
4     else:
5         kf = KFold(n_splits=k)
6
7     accuracies = []
8
9     for train_index, val_index in kf.split(X_train, y_train):
10        X_train_fold, X_val_fold = X_train[train_index], X_train[val_index]
11        y_train_fold, y_val_fold = y_train.iloc[train_index], y_train.iloc[val_index]
12
13        model.fit(X_train_fold, y_train_fold)
14        y_pred = model.predict(X_val_fold)
15        accuracy = accuracy_score(y_val_fold, y_pred)
16        accuracies.append(accuracy)
17
18    avg_accuracy = sum(accuracies) / len(accuracies)
19    return avg_accuracy
```

Model: LogisticRegression

Average accuracy: 0.9852634510132303

Accuracy: 0.9857397504456328

Model: DecisionTreeClassifier

Average accuracy: 0.9948006902729126

Accuracy: 0.9961675579322639



```
1 pipeline = Pipeline([("tfidf", TfidfVectorizer()), ("clf", LogisticRegression())])
2 cv_scores = cross_val_score(pipeline, X, y, cv=5)
3
4 print("Average Accuracy:", cv_scores.mean())
5
6 pipeline.fit(X, y)
7 pred_lr = pipeline.predict(X_test)
8 cm = confusion_matrix(y_test, pred_lr)
9
```

Classification Report:					
	precision	recall	f1-score	support	
0	0.99	0.98	0.99	58	
1	0.98	0.99	0.99	53	
accuracy			0.99	112	
macro avg	0.99	0.99	0.99	112	
weighted avg	0.99	0.99	0.99	112	

Without pipeline

With pipeline

Classification Report:					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	5777	
1	0.99	0.99	0.99	5443	
accuracy			0.99	11220	
macro avg	0.99	0.99	0.99	11220	
weighted avg	0.99	0.99	0.99	11220	

Testing



```
news = str(  
    "21st Century Wire says This week, the historic international  
)  
manual_test(news)
```

[]

Python

...

LR Prediction: True
DT Prediction: True
GBC Prediction: True
RFC Prediction:True

THANK YOU FOR LISTENING!
