Sarcasm Detection Using NLP

Group 29

Tejas Badgujar (tcb2145)
Sion Chun (sc3791)
Ritayan Patra (rp3247)
Meyhar Sharma (ms7070)

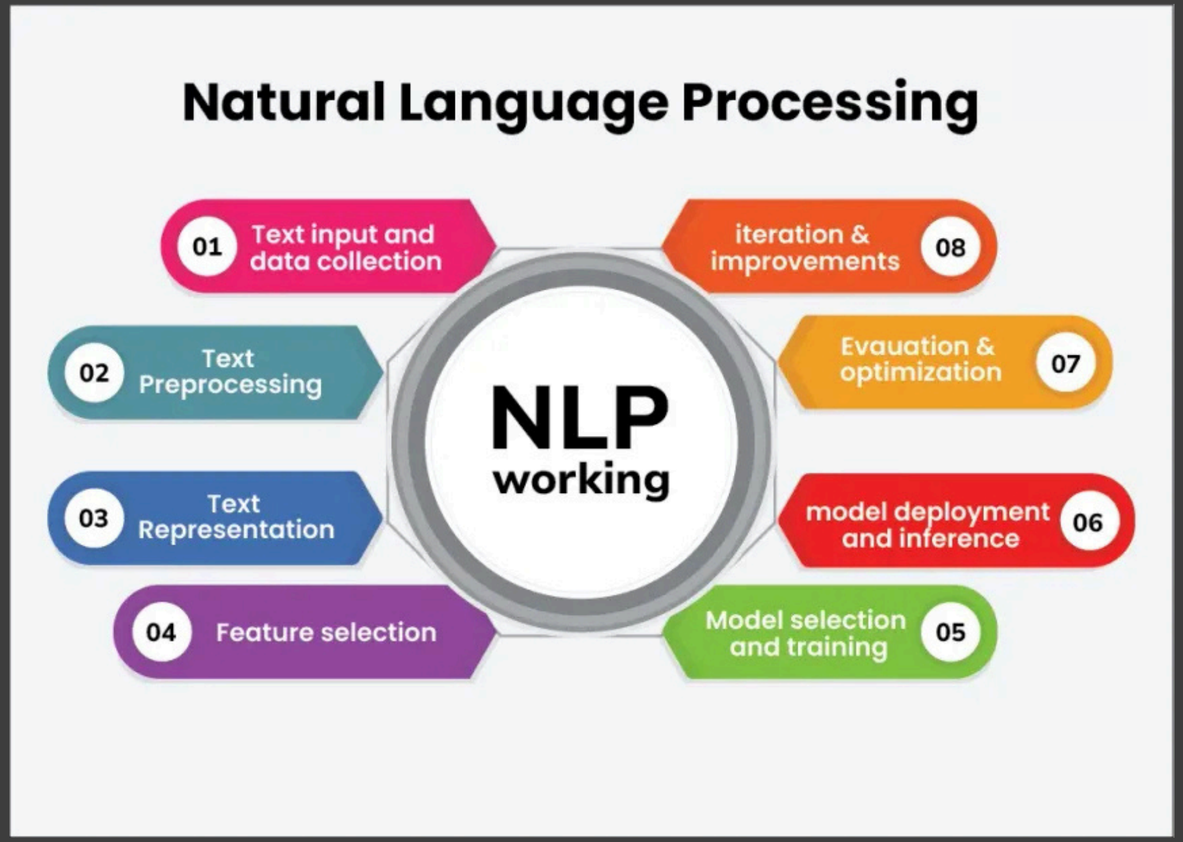
About the Project

Sarcasm is an interesting notion to capture and quantify in a Natural Language Processing setting, and implementing the same can be fairly difficult.

Doing so in settings such as headlines from the news is fairly important, as discerning between sarcastic and non-sarcastic headlines could greatly change the writer's overall intent and the likely content of the article itself. It impacts which and how many readers are interested in reading it based on the headline.

Objective

The goal of this machine learning project is to develop a NLP-based machine learning model that can detect whether or not (binary classification) a given headline for a news article is sarcastic.



Sneak peak into the data!

The data collected comes from an overly sarcastic, satirical news source (The Onion) and a straightforward, likely non-sarcastic news source (HuffPost) and has been taken from - Sarcasm Headlines Dataset Kaggle.

Each record consists of three attributes:

- is_sarcastic: 1 if the record is sarcastic otherwise 0.
- headline: the headline of the news article.
- article_link: link to the original news article.

```
data_df = pd.read_json('./data/Sarcasm_Headlines_Dataset.json', lines=True)

print(data_df.shape)
data_df.head()
```

```
(26709, 3)
```

	article_link	headline	is_sarcastic
0	https://www.huffingtonpost.com/entry/versace-b...	former versace store clerk sues over secret 'b...	0
1	https://www.huffingtonpost.com/entry/roseanne-...	the 'roseanne' revival catches up to our thorn...	0
2	https://local.theonion.com/mom-starting-to-fea...	mom starting to fear son's web series closest ...	1
3	https://politics.theonion.com/boehner-just-wan...	boehner just wants wife to listen, not come up...	1
4	https://www.huffingtonpost.com/entry/jk-rowlin...	j.k. rowling wishes snape happy birthday in th...	0


```
data_df = pd.read_json('./data/Sarcasm_Headlines_Dataset.json', lines=True)

print(data_df.shape)
data_df.head()
```

```
(26709, 3)
```

	article_link	headline	is_sarcastic
0	https://www.huffingtonpost.com/entry/versace-b...	former versace store clerk sues over secret 'b...	0
1	https://www.huffingtonpost.com/entry/roseanne-...	the 'roseanne' revival catches up to our thorn...	0
2	https://local.theonion.com/mom-starting-to-fea...	mom starting to fear son's web series closest ...	1
3	https://politics.theonion.com/boehner-just-wan...	boehner just wants wife to listen, not come up...	1
4	https://www.huffingtonpost.com/entry/jk-rowlin...	j.k. rowling wishes snape happy birthday in th...	0


```
data_df.isnull().sum()
```

```
article_link    0
headline        0
is_sarcastic    0
dtype: int64
```

```
X = data_df['headline']
y = data_df['is_sarcastic']
print(X.shape, y.shape)
```

```
(26709,) (26709,)
```

Pre-processing

The headlines have been preprocessed to handle the following

- Only alphanumeric characters should be present. The rest are replaced with a space.
- No numeric values such as integer and float should be present.
- The entire text of the headline is lower cases
- However, single quotation marks are preserved as is because some quoted words may indicate sarcasm hence relevant for the model to learn

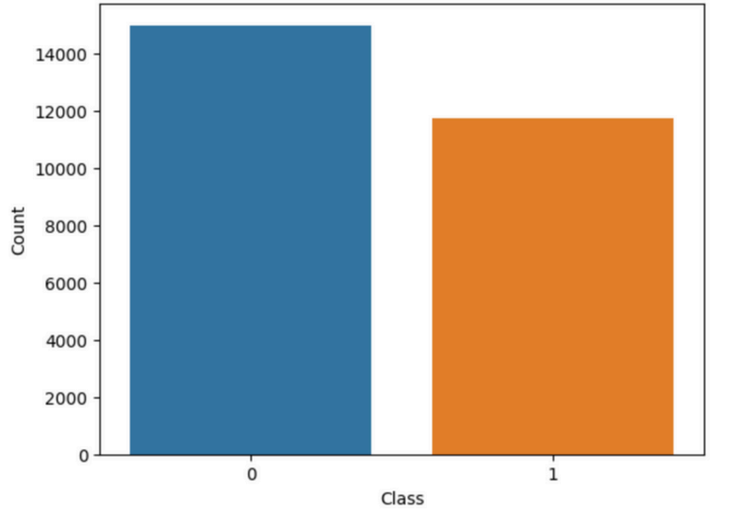
```python
def text_cleaning(text):
    text = re.sub(r"[^A-Za-z0-9']", " ", text) # keep single quote for sarcasm
    text = re.sub(r"\'s", " ", text)
    text = re.sub(r'\b\d+(?:\.\d+)?\s+', '', text)
    text = text.lower()
    return text
```

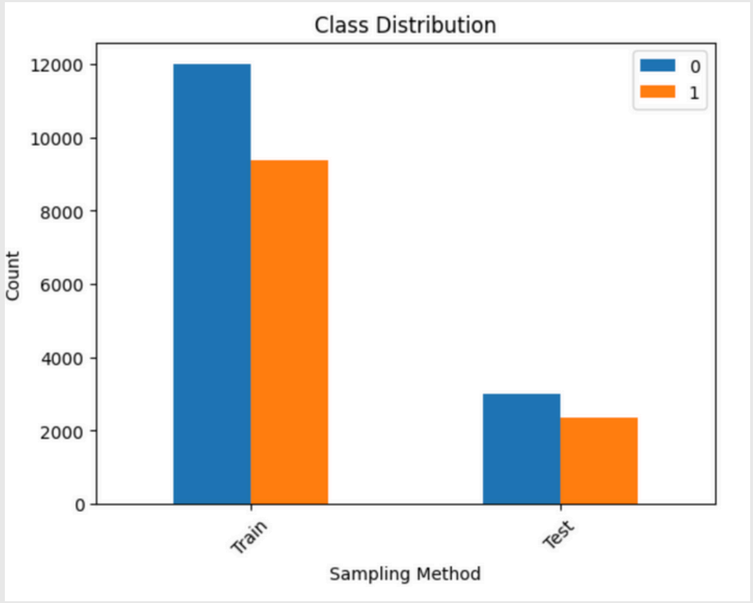
```python
count = 0
for i in range(len(X)):
    original_text = X[i]
    cleaned_text = text_cleaning(original_text)

    if original_text != cleaned_text:
        print("Original: ", original_text)
        print("Cleaned: ", cleaned_text)
        print("-" * 50)
        count +=1
    if count>=10: break
```

```
Original:  the 'roseanne' revival catches up to our thorny political mood, for better and worse
Cleaned:  the 'roseanne' revival catches up to our thorny political mood  for better and worse
--------------------------------------------------
Original:  mom starting to fear son's web series closest thing she will have to grandchild
Cleaned:  mom starting to fear son  web series closest thing she will have to grandchild
--------------------------------------------------
Original:  boehner just wants wife to listen, not come up with alternative debt-reduction ideas
Cleaned:  boehner just wants wife to listen  not come up with alternative debt reduction ideas
--------------------------------------------------
Original:  j.k. rowling wishes snape happy birthday in the most magical way
Cleaned:  j k  rowling wishes snape happy birthday in the most magical way
--------------------------------------------------
Original:  advancing the world's women
Cleaned:  advancing the world  women
--------------------------------------------------
Original:  the fascinating case for eating lab-grown meat
Cleaned:  the fascinating case for eating lab grown meat
--------------------------------------------------
Original:  this ceo will send your kids to school, if you work for his company
Cleaned:  this ceo will send your kids to school  if you work for his company
--------------------------------------------------
Original:  friday's morning email: inside trump's presser for the ages
Cleaned:  friday  morning email  inside trump  presser for the ages
```

EDA - Class Distribution

```
Class Distribution:
 0    14985
1    11724
Name: is_sarcastic, dtype: int64
```






```
cd = y.value_counts()
print("Class Distribution: \n", cd)

sns.barplot(x=cd.index, y=cd.values)
plt.xlabel('Class')
plt.ylabel('Count')
plt.show()
```

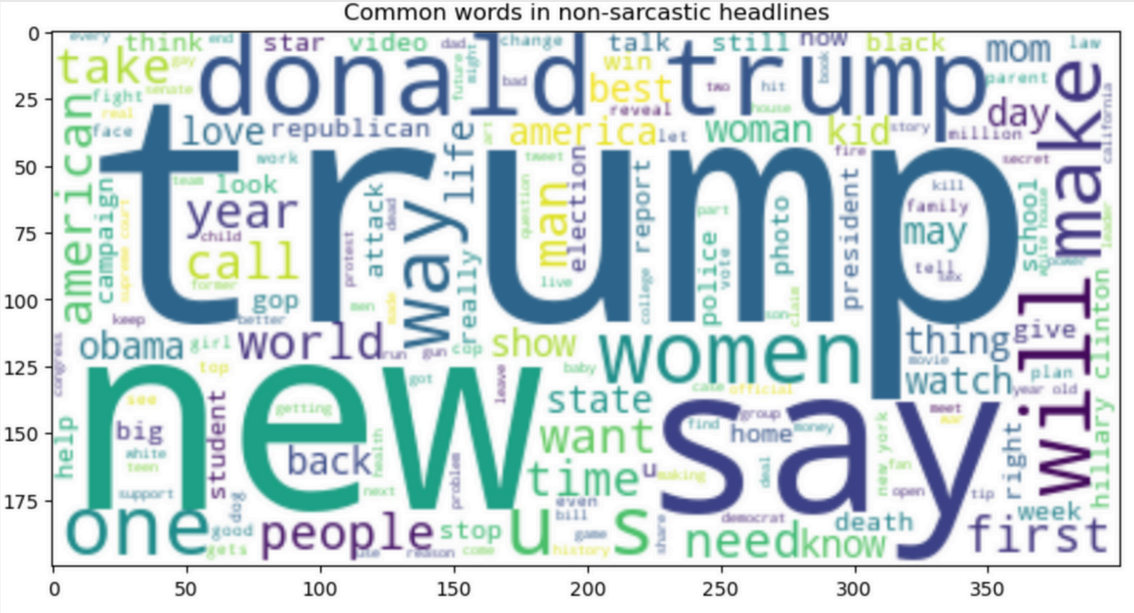
```
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2, random_state=42)
print(f"Train: {X_train.shape} {X_test.shape} | Test: {y_train.shape} {y_test.shape}")
```

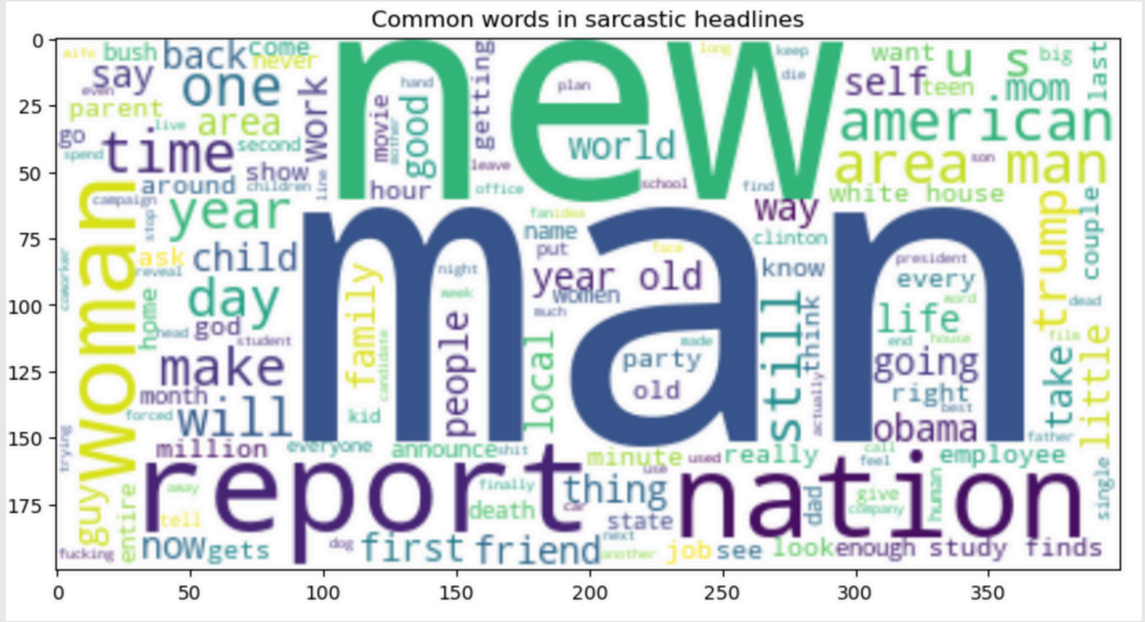
```
Train: (21367,) (5342,) | Test: (21367,) (5342,)
```

```
cd_train = y_train.value_counts()
cd_test = y_test.value_counts()

cd_df = pd.DataFrame({
    'Train': cd_train,
    'Test': cd_test,
})

cd_df.T.plot(kind='bar')
plt.title('Class Distribution')
plt.xlabel('Sampling Method')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

- The plots show that about 56% of the data is headlines that are not sarcastic while 44% headlines are sarcastic.
- This ratio has been maintained during the train-test split of the data.

EDA - Feature Extraction





- In sarcastic headlines the most common word is ‘man’ and ‘new’. Other common words include ‘report’, ‘nation’, ‘woman’. This gives an idea of the commonly used words used by the news outlets to show sarcasm.

- While the non-sarcastic headlines include words like ‘trump’, ‘donald’, ‘new’, ‘say’. From the above picture, it is possible to infer that non-sarcastic words include more proper nouns than sarcastic words. It contains names of indiviuals, country, date, etc.

Project Models (upcoming)

Logistic Regression

Tree Based Classifiers

Neural Networks

NLP

details.