```python
import optparse
install_requirements()
```

Feature Extraction

```python
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(tweets)
```

```python
print(vectorizer.get_feature_names())
```

```python
print(X.shape)
```

Lets see how can bigrams and trigrams can be included here

```python
vectorizer_ngram_range = CountVectorizer(analyzer='word', ngram_range=(1,3))
bow_matrix_ngram = vectorizer_ngram_range.fit_transform(tweets)
```

```python
print(vectorizer_ngram_range.get_feature_names())
```

```python
print(bow_matrix_ngram.shape)
```

Tf-idf vectorizer

```python
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer_n_gram_max_features = TfidfVectorizer(norm="l2", analyzer='word', ngram_range=(1,3), max_features=6000)
tf_idf_matrix_n_gram_max_features = vectorizer_n_gram_max_features.fit_transform(tweets)
```

```python
print(vectorizer_n_gram_max_features.get_feature_names())
```

```python
print(tf_idf_matrix_n_gram_max_features.shape)
```

Generate wordcloud

```python
# import wordcloud
from wordcloud import WordCloud
import matplotlib.pyplot as plt

from collections import Counter
wordcount = Counter(" ".join(tweets).split())

plt.figure(figsize=(20,20))
wordcloud = WordCloud(width=3000, height=2000, background_color='black', max_words=500, colormap='Set2').generate(text)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```