In [23]:
```
import pandas as pd
import numpy as np
import re
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

In [7]:
```
data = pd.read_csv('/Users/raunavsharma/Downloads/IMDB.csv')
print(data.shape)
data.head(10)
```

```
(50000, 2)
```

Out[7]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
5	Probably my all-time favorite movie, a story o...	positive
6	I sure would like to see a resurrection of a u...	positive
7	This show was an amazing, fresh & innovative i...	negative
8	Encouraged by the positive comments about this...	negative
9	If you like original gut wrenching laughter yo...	positive

In [8]:
```
data.describe()
```

Out[8]:

	review	sentiment
count	50000	50000
unique	49582	2

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	review	sentiment
top	Loved today's show!!! It was a variety and not...	negative
freq	5	25000

In [10]:
```
data["sentiment"].value_counts()
```

Out[10]:
```
negative    25000
positive    25000
Name: sentiment, dtype: int64
```

In [86]:
```
reviews_train = []
for line in open('/Users/raunavsharma/Downloads/movie_data/full_train.txt', 'r'):
    reviews_train.append(line.strip())

reviews_test = []
for line in open('/Users/raunavsharma/Downloads/movie_data/full_test.txt', 'r'):
    reviews_test.append(line.strip())
```

In [87]:
```
replace_no_space = re.compile("[.;:!\'?,\"()\[\]]")
replace_with_space = re.compile("(<br\s*/><br\s*/>)|(\-)|(\/)")

def preprocess_reviews(reviews):
    reviews = [replace_no_space.sub("", line.lower()) for line in reviews]
    reviews = [replace_with_space.sub(" ", line) for line in reviews]

    return reviews

reviews_train_clean = preprocess_reviews(reviews_train)
reviews_test_clean = preprocess_reviews(reviews_test)
```

In [88]:
```
p = CountVectorizer(binary=True)
p.fit(reviews_train_clean)
X = p.transform(reviews_train_clean)
X_test = p.transform(reviews_test_clean)
X_train = p.transform(reviews_test_clean)
```

In [122…
```
target = [1 if i < 12500 else 0 for i in range(25000)]
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```python
X_train, X_val, y_train, y_val = train_test_split(
    X, target, train_size = 0.75
)

for c in [0.01, 0.05, 0.25, 0.5, 1]:

    lr = LogisticRegression(C=c)
    lr.fit(X_train, y_train)
    print ("Accuracy for C=%s: %s"
           % (c, accuracy_score(y_val, lr.predict(X_val))))
```

```
Accuracy for C=0.01: 0.86736
/Users/raunavsharma/opt/anaconda3/lib/python3.8/site-packages/sklearn/linear_model/_logistic.py:762: ConvergenceWarni
ng: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy for C=0.05: 0.87744
/Users/raunavsharma/opt/anaconda3/lib/python3.8/site-packages/sklearn/linear_model/_logistic.py:762: ConvergenceWarni
ng: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy for C=0.25: 0.8752
/Users/raunavsharma/opt/anaconda3/lib/python3.8/site-packages/sklearn/linear_model/_logistic.py:762: ConvergenceWarni
ng: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy for C=0.5: 0.87312
Accuracy for C=1: 0.8696
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
/Users/raunavsharma/opt/anaconda3/lib/python3.8/site-packages/sklearn/linear_model/_logistic.py:762: ConvergenceWarni
ng: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [123…

```
final_model = LogisticRegression(C=0.05)
final_model.fit(X, target)
print ("Final Accuracy: %s"
       % accuracy_score(target, final_model.predict(X_test)))
```

```
Final Accuracy: 0.88156
```

```
/Users/raunavsharma/opt/anaconda3/lib/python3.8/site-packages/sklearn/linear_model/_logistic.py:762: ConvergenceWarni
ng: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [124…

```
feature_to_coef = {
    word: coef for word, coef in zip(
        p.get_feature_names(), final_model.coef_[0]
    )
}
for best_positive in sorted(
    feature_to_coef.items(),
    key=lambda x: x[1],
    reverse=True)[:5]:
    print (best_positive)



for best_negative in sorted(
    feature_to_coef.items(),
    key=lambda x: x[1])[:5]:
    print (best_negative)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
('excellent', 0.9287863620357822)
('perfect', 0.7916864122112646)
('great', 0.6740677270263481)
('amazing', 0.6131983883926218)
('superb', 0.6011322324407015)
('worst', -1.3645720549264835)
('waste', -1.166786439535322)
('awful', -1.0321101486702577)
('poorly', -0.8751951963515281)
('boring', -0.8567832150328295)
```

In [ ]:
```
## In this project, I first cleaned the data and then performed the Logistic Regression. I also got various accuracie
## References:
## https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
## https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD