```
import pandas as pd

df = pd.read_csv('/content/IMDB Dataset.csv')

print(df)
```

```
                                                  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
...                                                  ...       ...
49995  I thought this movie did a down right good job...  positive
49996  Bad plot, bad dialogue, bad acting, idiotic di...  negative
49997  I am a Catholic taught in parochial elementary...  negative
49998  I'm going to have to disagree with the previou...  negative
49999  No one expects the Star Trek movies to be high...  negative

[50000 rows x 2 columns]
```

```
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer()
docs = np.array(['I am Bindu, studying in GLB'
                 'I wanna pet a husky'
                 'They are adorable'])
bag = vect.fit_transform(docs)
```

```
print(vect.vocabulary_)
```

```
{'am': 1, 'bindu': 3, 'studying': 8, 'in': 6, 'glbi': 4, 'wanna': 9, 'pet': 7, 'huskythey': 5, 'are': 2
```

```
print(bag.toarray())
```

```
[[1 1 1 1 1 1 1 1 1 1]]
```

```
from sklearn.feature_extraction.text import TfidfTransformer
np.set_printoptions(precision =2)
tfidf = TfidfTransformer(use_idf=True,norm='l2',smooth_idf=True )
print(tfidf.fit_transform(bag).toarray())
```

```
[[0.32 0.32 0.32 0.32 0.32 0.32 0.32 0.32 0.32 0.32]]
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(
                        use_idf = True,
                        norm = 'l2',
                        smooth_idf=True)

y = df.sentiment.values
x = tfidf.fit_transform(df['review'].values.astype('U'))
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test = train_test_split(x ,y,random_state=1,test_size=0.5,shuffle=False)
```

```
import pickle
from sklearn.linear_model import LogisticRegressionCV
clf = LogisticRegressionCV(cv = 5,
                           scoring = 'accuracy',
                           random_state = 0,
                           n_jobs = -1,
                           verbose = 3,
                           max_iter = 300).fit(x_train,y_train)

saved_model = open('saved_model.sav','wb')
pickle.dump(clf,saved_model)
saved_model.close()
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:  4.3min finished
```

```
filename = 'saved_model.sav'
saved_clf = pickle.load(open(filename,'rb'))

saved_clf.score(x_test,y_test)
```

```
0.89712
```