```
import numpy as np
import pandas as pd
```

```
dataset = pd.read_csv("IMDB Dataset.csv")
df = pd.read_csv("IMDB Dataset.csv")
```

```
dataset.head()
```

	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

```
dataset["review"]
```

```
0        One of the other reviewers has mentioned that ...
1        A wonderful little production. <br /><br />The...
2        I thought this was a wonderful way to spend ti...
3        Basically there's a family where a little boy ...
4        Petter Mattei's "Love in the Time of Money" is...
                               ...
49995    I thought this movie did a down right good job...
49996    Bad plot, bad dialogue, bad acting, idiotic di...
49997    I am a Catholic taught in parochial elementary...
49998    I'm going to have to disagree with the previou...
49999    No one expects the Star Trek movies to be high...
Name: review, Length: 50000, dtype: object
```

```
dataset.shape
```

```
(50000, 2)
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset["sentiment"] = le.fit_transform(dataset["sentiment"])
```

```
dataset.head(10)
```

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

```
import re
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

```
data = []
data = []
```

```
for i in range(0,50000):
    review = dataset["review"][i]
    review = re.sub('[^a-zA-Z]', ' ',review)
    review = review.lower()
    review = review.split()

    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    data.append(review)
```

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 2000)
x = cv.fit_transform(data).toarray()
y = dataset.iloc[:,1:2].values
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2, random_state = 0)
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
model =  Sequential()
```

```
model.add(Dense(units = 4000,kernel_initializer = "random_uniform",activation= "relu"))
model.add(Dense(units = 8000,kernel_initializer = "random_uniform",activation= "relu"))
model.add(Dense(units = 8000,kernel_initializer = "random_uniform",activation= "relu"))
model.add(Dense(units = 1,kernel_initializer = "random_uniform",activation= "sigmoid"))
```

```
model.compile(optimizer = "rmsprop",loss = "binary_crossentropy",metrics = ["accuracy"])
```

```
model.fit(x_train,y_train, epochs = 3)
```

```
Epoch 1/3
1250/1250 [==============================] - 2645s 2s/step - loss: 0.2944 - accuracy: 0.9051
Epoch 2/3
1250/1250 [==============================] - 2399s 2s/step - loss: 0.2434 - accuracy: 0.9315
Epoch 3/3
1250/1250 [==============================] - 1688s 1s/step - loss: 0.1932 - accuracy: 0.9560
<tensorflow.python.keras.callbacks.History at 0x216250cd7f0>
```

```
pred = model.predict(x_test)
pred = pred>0.5
#comp pred y_test
```

```
pred[:10]
```

```
array([[ True],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [ True],
       [False],
       [False],
       [False]])
```

```
y_test[:10]
```

```
array([[1],
       [0],
       [0],
       [1],
       [1],
       [0],
       [1],
       [0],
       [1],
       [0]])
```