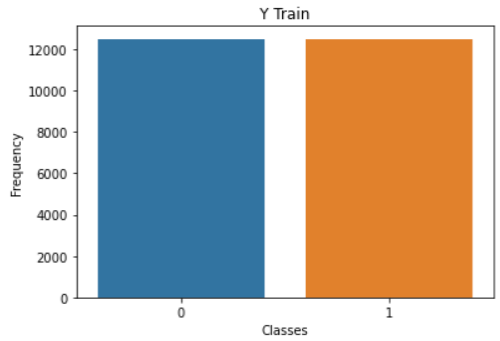
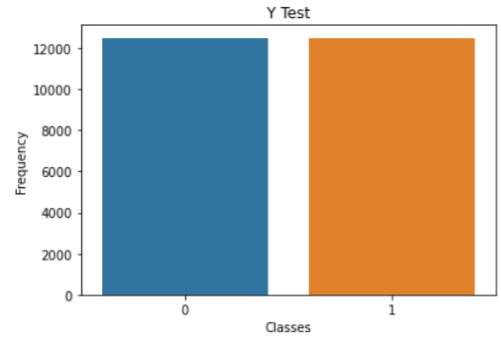
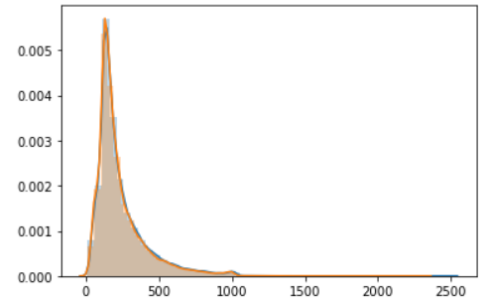
```
plt.figure();
sns.countplot(Y_train);
plt.xlabel("Classes");
plt.ylabel("Frequency");
plt.title("Y Train");
```




```
plt.figure();
sns.countplot(Y_test);
plt.xlabel("Classes");
plt.ylabel("Frequency");
plt.title("Y Test");
```




```
sns.distplot(review_len_train,hist_kws={"alpha":0.3});
sns.distplot(review_len_test,hist_kws={"alpha":0.3});
```



```
rnn = Sequential()

rnn.add(Embedding(num_words,32,input_length =len(X_train[0]))) # num_words=15000
rnn.add(SimpleRNN(16,input_shape = (num_words,maxlen), return_sequences=False,activation="relu"))
rnn.add(Dense(1)) #flatten
rnn.add(Activation("sigmoid")) #using sigmoid for binary classification

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

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 130, 32)           480000
_________________________________________________________________
simple_rnn (SimpleRNN)       (None, 16)                784
_________________________________________________________________
dense (Dense)                (None, 1)                 17
_________________________________________________________________
activation (Activation)      (None, 1)                 0
=================================================================
Total params: 480,801
Trainable params: 480,801
Non-trainable params: 0
_________________________________________________________________
None
```

```
history = rnn.fit(X_train,Y_train,validation_data = (X_test,Y_test),epochs = 5,batch_size=128,verbose = 1)
```

```
Epoch 1/5
196/196 [==============================] - 10s 50ms/step - loss: 0.6520 - accuracy: 0.6322 - val_loss: 0.5588 - val_accuracy:
0.7799
Epoch 2/5
196/196 [==============================] - 9s 48ms/step - loss: 0.4777 - accuracy: 0.7967 - val_loss: 0.4407 - val_accuracy: 0.
7997
Epoch 3/5
196/196 [==============================] - 10s 50ms/step - loss: 0.3459 - accuracy: 0.8548 - val_loss: 0.3921 - val_accuracy:
0.8346
Epoch 4/5
196/196 [==============================] - 10s 50ms/step - loss: 0.2680 - accuracy: 0.8928 - val_loss: 0.3208 - val_accuracy:
0.8628
Epoch 5/5
196/196 [==============================] - 9s 46ms/step - loss: 0.2191 - accuracy: 0.9154 - val_loss: 0.3230 - val_accuracy: 0.
8642
```

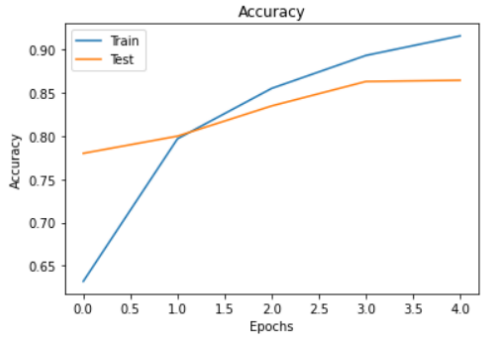
```
score = rnn.evaluate(X_test,Y_test)
```

```
782/782 [==============================] - 8s 10ms/step - loss: 0.3230 - accuracy: 0.8642
```

```
print("accuracy:", score[1]*100)
```

```
accuracy: 86.41600012779236
```

```
plt.figure()
plt.plot(history.history["accuracy"],label="Train");
plt.plot(history.history["val_accuracy"],label="Test");
plt.title("Accuracy")
plt.ylabel("Accuracy")
plt.xlabel("Epochs")
plt.legend()
plt.show();
```




```
plt.figure()
plt.plot(history.history["loss"],label="Train");
plt.plot(history.history["val_loss"],label="Test");
plt.title("Loss")
plt.ylabel("Loss")
plt.xlabel("Epochs")
plt.legend()
plt.show();
```