In [107]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import torch
```

In [108]:

```
df = pd.read_csv("Desktop/IMDB.csv")
df.head()
```

Out[108]:

	review	length of text	sentiment
0	Match 1: Tag Team Table Match Bubba Ray and Sp...	13704	positive
1	There's a sign on The Lost Highway that says:<...	12988	positive
2	(Some spoilers included:)<br /><br />Although,...	12930	positive
3	Back in the mid/late 80s, an OAV anime by titl...	12129	positive
4	**Attention Spoilers**<br /><br />First of all...	10363	positive

In [109]:

```
df['sentiment'] = df['sentiment'].replace('positive',1)
df['sentiment'] = df['sentiment'].replace('negative',0)
df['sentiment'].value_counts()
```

Out[109]:

```
1    25000
0    25000
Name: sentiment, dtype: int64
```

In [110]:

```
df.columns = ('text', 'length', 'label')

# computationally restricted model
#df_1 = df[:1000]

# full model
df_1 = df
```

In [111]:

```
X_train, X_test, y_train, y_test = train_test_split(df_1['text'], df_1[
'label'], test_size=0.2)
```

In [112]:

```
train = pd.concat([X_train, y_train], axis = 1)
test = pd.concat([X_test,y_test], axis = 1)
```

In [113]:

```
from simpletransformers.classification import ClassificationModel


# load in roberta, create a transformermodel
model = ClassificationModel('roberta', 'roberta-base', use_cuda = False,
args = {'overwrite_output_dir': True})
```

In [114]:

```
# train the model
model.train_model(train)
```

```
/opt/anaconda3/envs/transformers/lib/python3.8/site-packages/simpletran
sformers/classification/classification_model.py:262: UserWarning: Dataf
rame headers not specified. Falling back to using column 0 as text and
column 1 as labels.
  warnings.warn(
```

```
Running loss: 0.865722
```

```
/opt/anaconda3/envs/transformers/lib/python3.8/site-packages/torch/opti
m/lr_scheduler.py:231: UserWarning: To get the last learning rate compu
ted by the scheduler, please use `get_last_lr()`.
  warnings.warn("To get the last learning rate computed by the schedule
r, "
```

```
Running loss: 0.050782
```

```
/opt/anaconda3/envs/transformers/lib/python3.8/site-packages/torch/opti
m/lr_scheduler.py:200: UserWarning: Please also save or load the state
of the optimzer when saving or loading the scheduler.
  warnings.warn(SAVE_STATE_WARNING, UserWarning)
```

```
Running loss: 0.535551
```

In [115]:

```
# evaluate the model
result, model_outputs, wrong_predictions = model.eval_model(test)
```

```
/opt/anaconda3/envs/transformers/lib/python3.8/site-packages/simpletran
sformers/classification/classification_model.py:682: UserWarning: Dataf
rame headers not specified. Falling back to using column 0 as text and
column 1 as labels.
  warnings.warn(
```

In [116]:

```
print(result)
```

```
{'mcc': 0.8016350776166998, 'tp': 4612, 'tn': 4395, 'fp': 564, 'fn': 42
9, 'eval_loss': 0.32207771530747414}
```

In [117]:

```
from sklearn.metrics import roc_curve

# convert series to list for matplotlib
rocx = list(X_test)
rocy = list(y_test)
predictions, raw_outputs = model.predict(rocx)

fpr, tpr, threshold = roc_curve(rocy, predictions)
```

In [118]:

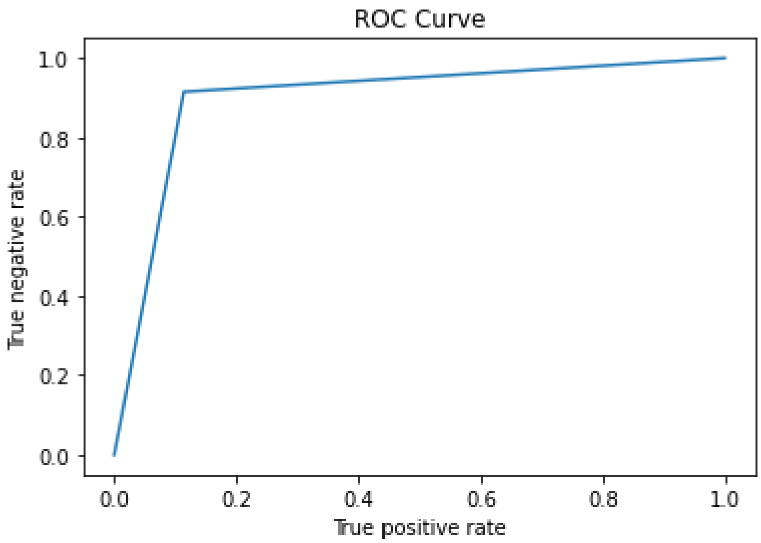
```
import matplotlib.pyplot as plt
```

In [119]:

```
plt.plot(fpr,tpr)
plt.title("ROC Curve")
plt.xlabel('True positive rate')
plt.ylabel('True negative rate')
```

Out[119]:

```
Text(0, 0.5, 'True negative rate')
```


ROC Curve
True negative rate
True positive rate