In [ ]:
```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, auc, f1_score, roc_curve
from tensorflow.keras.models import load_model
import tensorflow as tf
from sklearn.metrics import average_precision_score
import matplotlib.pyplot as plt
import pickle
import torch
import os
import sys
import glob
import json
```

In [5]:
```
def read_files():
    file_names = glob.glob('data/tt*.pkl')
    return file_names
```

In [3]:
```
def pre_process():
    from pathlib import Path
    from sklearn.model_selection import train_test_split
    for file in read_files():
        print(file)
        with open(file, 'rb') as f:
            pickle_model = pickle.load(f)
        place = torch.as_tensor(pickle_model.get('place'))
        cast = torch.as_tensor(pickle_model.get('cast'))
        action = torch.as_tensor(pickle_model.get('action'))
        audio = torch.as_tensor(pickle_model.get('audio'))
        scene_transition_boundary_ground_truth = torch.as_tensor(pickle_model.get('scene_transition_bou
ndary_ground_truth'))
        shot_end_frame = torch.as_tensor(pickle_model.get('shot_end_frame'))
        imdb_ids = pickle_model.get('imdb_id')

        a = []
        for i in range(0, place.shape[0]):
            if i<place.shape[0]-1:
                temp = place[i+1]-place[i]
            a.append(temp)
        places = torch.stack(a, dim=0)

        b = []
        for i in range(0, cast.shape[0]):
            if i<cast.shape[0]-1:
                temp = cast[i+1]-cast[i]
            b.append(temp)
        casts = torch.stack(b, dim=0)

        c = []
        for i in range(0, action.shape[0]):
            if i<action.shape[0]-1:
                temp = action[i+1]-action[i]
            c.append(temp)
        actions = torch.stack(c, dim=0)

        d = []
        for i in range(0, audio.shape[0]):
            if i<audio.shape[0]-1:
                temp = audio[i+1]-audio[i]
            d.append(temp)
        audios = torch.stack(d, dim=0)

        e = []
        for i in range(0, shot_end_frame.shape[0]):
            if i<shot_end_frame.shape[0]-1:
                temp = shot_end_frame[i+1]-shot_end_frame[i]
            e.append(temp)
        shot_end_frames = torch.stack(e, dim=0)

        tensor_x = tf.concat([places, casts, actions, audios], 1)
        tensor_y = scene_transition_boundary_ground_truth

        l = []
        for i in tensor_y:
            if i == False:
                l.append(0)
            else:
                l.append(1)
#         label = []
        label = np.array(l)

        px = pd.DataFrame(tensor_x).astype("double")[0:len(tensor_x)-1]
        px["shot_end_frame"] = shot_end_frames[0:len(tensor_x)-1]
        py = pd.DataFrame(label).astype("int")
#         py = py.values.ravel()

        x_train, x_test, y_train, y_test = train_test_split(px, py, test_size=0.2, stratify=py, random_
state=42, shuffle=True)
        output_dir = Path(imdb_ids)
        output_dir.mkdir(parents=True, exist_ok=True)
        x_train.to_csv(output_dir/'x_train.csv')
        x_test.to_csv(output_dir/'x_test.csv')
        y_train.to_csv(output_dir/'y_train.csv')
        y_test.to_csv(output_dir/'y_test.csv')
        print('COMPLETE!')
```

In [4]:
```
pre_process()
```

```
data\tt0114746.pkl
COMPLETE!
data\tt0115759.pkl
COMPLETE!
data\tt0116282.pkl
COMPLETE!
data\tt0116922.pkl
COMPLETE!
data\tt0117060.pkl
COMPLETE!
data\tt0118715.pkl
COMPLETE!
data\tt0119488.pkl
COMPLETE!
data\tt0120382.pkl
COMPLETE!
data\tt0120689.pkl
COMPLETE!
data\tt0120890.pkl
COMPLETE!
data\tt0123755.pkl
COMPLETE!
data\tt0124315.pkl
COMPLETE!
data\tt0137523.pkl
COMPLETE!
data\tt0163025.pkl
COMPLETE!
data\tt0172495.pkl
COMPLETE!
data\tt0178868.pkl
COMPLETE!
data\tt0190332.pkl
COMPLETE!
data\tt0217505.pkl
COMPLETE!
data\tt0253474.pkl
COMPLETE!
data\tt0281358.pkl
COMPLETE!
data\tt0319061.pkl
COMPLETE!
data\tt0361748.pkl
COMPLETE!
data\tt0379786.pkl
COMPLETE!
data\tt0399201.pkl
COMPLETE!
data\tt0409459.pkl
COMPLETE!
data\tt0440963.pkl
COMPLETE!
data\tt0443272.pkl
COMPLETE!
data\tt0479884.pkl
COMPLETE!
data\tt0780571.pkl
COMPLETE!
data\tt0822832.pkl
COMPLETE!
data\tt0945513.pkl
COMPLETE!
data\tt0976051.pkl
COMPLETE!
data\tt1001508.pkl
COMPLETE!
data\tt1038919.pkl
COMPLETE!
data\tt1099212.pkl
COMPLETE!
data\tt1119646.pkl
COMPLETE!
data\tt1205489.pkl
COMPLETE!
data\tt1375666.pkl
COMPLETE!
data\tt1412386.pkl
COMPLETE!
data\tt1707386.pkl
COMPLETE!
data\tt2024544.pkl
COMPLETE!
data\tt2488496.pkl
COMPLETE!
data\tt2582846.pkl
COMPLETE!
```

In [ ]:
```
```