In [1]:

```
# Import all necessary modules
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import matplotlib
import sklearn.model_selection
%matplotlib inline
# sagemaker libraries
import boto3
import sagemaker
```

In [2]:

```
# Load the data and perform exploratory analysis
df = pd.read_csv('data.csv', header = 0, index_col = 0)
```

In [3]:

```
# Scale the data for normalization
from sklearn.preprocessing import MinMaxScaler
scaler  = MinMaxScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df.iloc[:,1:]), columns=df.iloc[:,1:].column
s)
```

In [4]:

```
dict_category = {'B':0, 'M':1}
target = df.iloc[:,0].apply(lambda x : dict_category[x]) # convert the class into 0-1 Bina
ry and store the target class
X_train,X_test,Y_train,Y_test = sklearn.model_selection.train_test_split(df_scaled, target
,test_size=0.2)
```

In [5]:

```
data_dir = '../Capstone Project/data_full/'
if not os.path.exists( data_dir ):
    os.makedirs(data_dir)
```

In [6]:

```
# create the training and test data and save locally
test = np.hstack([np.reshape(Y_test.values, (Y_test.shape[0], 1)), X_test.values])
train = np.hstack([np.reshape(Y_train.values, (Y_train.shape[0], 1)), X_train.values])
pd.DataFrame(test).to_csv(os.path.join(data_dir,'test.csv'),header=False,index=False)
pd.DataFrame(train).to_csv(os.path.join(data_dir,'train.csv'),header=False,index=False)
```

In [8]:

```
# How many samples in train test??
print(f'We have {X_test.shape[0]} number of test samples in our dataset')
print(f'We have {X_train.shape[0]} number of train samples in our dataset')
print(f'We have {X_train.shape[1]} features')
```

```
We have 114 number of test samples in our dataset
We have 455 number of train samples in our dataset
We have 30 features
```

In [9]:

```
from sagemaker import get_execution_role
session =sagemaker.Session()
# store the current SageMaker session
# get IAM role
role=get_execution_role()
print(role)

bucket_name=session.default_bucket()
```

```
arn:aws:iam::172268057478:role/service-role/AmazonSageMaker-ExecutionRole-202
10122T150167
```

In [10]:

```
# set prefix, a descriptive name for a directory for our train test data
data_dir = '../Capstone Project/data_full/'
prefix = 'cancer-class'
# upload all data to S3
test_location = session.upload_data(os.path.join(data_dir,'test.csv'),key_prefix=prefix)
train_location= session.upload_data(os.path.join(data_dir,'train.csv'),key_prefix=prefix)
```

In [11]:

```
from sagemaker.pytorch import PyTorch

# Create an estimtor
# your import and estimator code, here
output_path = 's3://{}/{}'.format(bucket_name, prefix)
estimator = PyTorch(entry_point="train.py",
                    source_dir="source_pytorch",
                    role=role,
                    framework_version='1.0',
                    py_version = 'py3',
                    sagemaker_session = session,
                    output_path = output_path,
                    train_instance_count=1,
                    train_instance_type='ml.p2.xlarge',
                    hyperparameters= {'input_features':30,
                                      'hidden_dim':6,
                                      'output_dim':1,
                                      'epochs':70
                                     }
                   )
```

```
train_instance_count has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
train_instance_type has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
```

In [12]:

```
%time

# Train your estimator on S3 training data
s3_input_train = sagemaker.TrainingInput(s3_data = train_location, content_type='csv')
estimator.fit({'train':s3_input_train})
```

```
CPU times: user 3 µs, sys: 1e+03 ns, total: 4 µs
Wall time: 8.34 µs
2021-03-17 21:25:37 Starting - Starting the training job...
2021-03-17 21:26:01 Starting - Launching requested ML instancesProfilerReport
-1616016337: InProgress
.........
2021-03-17 21:27:21 Starting - Preparing the instances for trainin
g............
2021-03-17 21:29:30 Downloading - Downloading input data
2021-03-17 21:29:30 Training - Downloading the training image.....bash: canno
t set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
2021-03-17 21:30:20,047 sagemaker-containers INFO     Imported framework sage
maker_pytorch_container.training
2021-03-17 21:30:20,078 sagemaker_pytorch_container.training INFO     Block u
ntil all host DNS lookups succeed.
2021-03-17 21:30:20,304 sagemaker_pytorch_container.training INFO     Invokin
g user training script.
2021-03-17 21:30:20,610 sagemaker-containers INFO     Module train does not p
rovide a setup.py.
Generating setup.py
2021-03-17 21:30:20,610 sagemaker-containers INFO     Generating setup.cfg
2021-03-17 21:30:20,611 sagemaker-containers INFO     Generating MANIFEST.in
2021-03-17 21:30:20,611 sagemaker-containers INFO     Installing module with
 the following command:
/usr/bin/python -m pip install -U .
Processing /opt/ml/code
Building wheels for collected packages: train
  Running setup.py bdist_wheel for train: started
  Running setup.py bdist_wheel for train: finished with status 'done'
  Stored in directory: /tmp/pip-ephem-wheel-cache-czwpc3xz/wheels/35/24/16/37
574d11bf9bde50616c67372a334f94fa8356bc7164af8ca3
Successfully built train
Installing collected packages: train
Successfully installed train-1.0.0
You are using pip version 18.1, however version 21.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
2021-03-17 21:30:22,968 sagemaker-containers INFO     Invoking user script

Training Env:

{
    "additional_framework_parameters": {},
    "channel_input_dirs": {
        "train": "/opt/ml/input/data/train"
    },
    "current_host": "algo-1",
    "framework_module": "sagemaker_pytorch_container.training:main",
    "hosts": [
        "algo-1"
    ],
    "hyperparameters": {
        "hidden_dim": 6,
        "input_features": 30,
        "epochs": 70,
        "output_dim": 1
    },
```

```
    "input_config_dir": "/opt/ml/input/config",
    "input_data_config": {
        "train": {
            "ContentType": "csv",
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        }
    },
    "input_dir": "/opt/ml/input",
    "is_master": true,
    "job_name": "sagemaker-pytorch-2021-03-17-21-25-37-062",
    "log_level": 20,
    "master_hostname": "algo-1",
    "model_dir": "/opt/ml/model",
    "module_dir": "s3://sagemaker-us-east-1-172268057478/sagemaker-pytorch-20
21-03-17-21-25-37-062/source/sourcedir.tar.gz",
    "module_name": "train",
    "network_interface_name": "eth0",
    "num_cpus": 4,
    "num_gpus": 1,
    "output_data_dir": "/opt/ml/output/data",
    "output_dir": "/opt/ml/output",
    "output_intermediate_dir": "/opt/ml/output/intermediate",
    "resource_config": {
        "current_host": "algo-1",
        "hosts": [
            "algo-1"
        ],
        "network_interface_name": "eth0"
    },
    "user_entry_point": "train.py"
}

Environment variables:

SM_HOSTS=["algo-1"]
SM_NETWORK_INTERFACE_NAME=eth0
SM_HPS={"epochs":70,"hidden_dim":6,"input_features":30,"output_dim":1}
SM_USER_ENTRY_POINT=train.py
SM_FRAMEWORK_PARAMS={}
SM_RESOURCE_CONFIG={"current_host":"algo-1","hosts":["algo-1"],"network_inter
face_name":"eth0"}
SM_INPUT_DATA_CONFIG={"train":{"ContentType":"csv","RecordWrapperType":"Non
e","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"}}
SM_OUTPUT_DATA_DIR=/opt/ml/output/data
SM_CHANNELS=["train"]
SM_CURRENT_HOST=algo-1
SM_MODULE_NAME=train
SM_LOG_LEVEL=20
SM_FRAMEWORK_MODULE=sagemaker_pytorch_container.training:main
SM_INPUT_DIR=/opt/ml/input
SM_INPUT_CONFIG_DIR=/opt/ml/input/config
SM_OUTPUT_DIR=/opt/ml/output
SM_NUM_CPUS=4
SM_NUM_GPUS=1
SM_MODEL_DIR=/opt/ml/model
```

```
SM_MODULE_DIR=s3://sagemaker-us-east-1-172268057478/sagemaker-pytorch-2021-03
-17-21-25-37-062/source/sourcedir.tar.gz
SM_TRAINING_ENV={"additional_framework_parameters":{},"channel_input_dirs":
{"train":"/opt/ml/input/data/train"},"current_host":"algo-1","framework_modul
e":"sagemaker_pytorch_container.training:main","hosts":["algo-1"],"hyperparam
eters":{"epochs":70,"hidden_dim":6,"input_features":30,"output_dim":1},"input
_config_dir":"/opt/ml/input/config","input_data_config":{"train":{"ContentTyp
e":"csv","RecordWrapperType":"None","S3DistributionType":"FullyReplicated","T
rainingInputMode":"File"}},"input_dir":"/opt/ml/input","is_master":true,"job_
name":"sagemaker-pytorch-2021-03-17-21-25-37-062","log_level":20,"master_host
name":"algo-1","model_dir":"/opt/ml/model","module_dir":"s3://sagemaker-us-ea
st-1-172268057478/sagemaker-pytorch-2021-03-17-21-25-37-062/source/sourcedir.
tar.gz","module_name":"train","network_interface_name":"eth0","num_cpus":4,"n
um_gpus":1,"output_data_dir":"/opt/ml/output/data","output_dir":"/opt/ml/outp
ut","output_intermediate_dir":"/opt/ml/output/intermediate","resource_confi
g":{"current_host":"algo-1","hosts":["algo-1"],"network_interface_name":"eth
0"},"user_entry_point":"train.py"}
SM_USER_ARGS=["--epochs","70","--hidden_dim","6","--input_features","30","--o
utput_dim","1"]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_TRAIN=/opt/ml/input/data/train
SM_HP_HIDDEN_DIM=6
SM_HP_INPUT_FEATURES=30
SM_HP_EPOCHS=70
SM_HP_OUTPUT_DIM=1
PYTHONPATH=/usr/local/bin:/usr/lib/python36.zip:/usr/lib/python3.6:/usr/lib/p
ython3.6/lib-dynload:/usr/local/lib/python3.6/dist-packages:/usr/lib/python3/
dist-packages

Invoking script with the following command:

/usr/bin/python -m train --epochs 70 --hidden_dim 6 --input_features 30 --out
put_dim 1


Using device cuda.
Get train data loader.

2021-03-17 21:30:38 Uploading - Uploading generated training modelEpoch: 1, L
oss: 0.587756317594777
Epoch: 2, Loss: 0.2899604067206383
Epoch: 3, Loss: 0.15904661895626265
Epoch: 4, Loss: 0.13384426279884318
Epoch: 5, Loss: 0.12015294727018994
Epoch: 6, Loss: 0.10480490249946066
Epoch: 7, Loss: 0.09329626022878548
Epoch: 8, Loss: 0.08592536075688574
Epoch: 9, Loss: 0.08105596011180592
Epoch: 10, Loss: 0.07780973190117789
Epoch: 11, Loss: 0.07549264685898695
Epoch: 12, Loss: 0.07370754845602356
Epoch: 13, Loss: 0.07224946690257639
Epoch: 14, Loss: 0.07100484377967761
Epoch: 15, Loss: 0.06994116757287765
Epoch: 16, Loss: 0.06899494991597274
Epoch: 17, Loss: 0.06814446243072819
Epoch: 18, Loss: 0.06737407528714318
```

```
Epoch: 19, Loss: 0.06666959552904186
Epoch: 20, Loss: 0.06601947152723922
Epoch: 21, Loss: 0.06541728906576401
Epoch: 22, Loss: 0.0648535487252166
Epoch: 23, Loss: 0.06432552868500352
Epoch: 24, Loss: 0.0638301123905441
Epoch: 25, Loss: 0.06336418713402489
Epoch: 26, Loss: 0.06292326672954242
Epoch: 27, Loss: 0.06250922459825549
Epoch: 28, Loss: 0.0621190032283978
Epoch: 29, Loss: 0.061750815816871495
Epoch: 30, Loss: 0.06140351407331131
Epoch: 31, Loss: 0.061076272304331804
Epoch: 32, Loss: 0.06076471590558472
Epoch: 33, Loss: 0.06046861530099388
Epoch: 34, Loss: 0.0601650090665435
Epoch: 35, Loss: 0.05990169172519413
Epoch: 36, Loss: 0.05965080116774239
Epoch: 37, Loss: 0.05941227730869761
Epoch: 38, Loss: 0.05918077094753718
Epoch: 39, Loss: 0.05896086298921348
Epoch: 40, Loss: 0.05873960530405621
Epoch: 41, Loss: 0.058530334271895496
Epoch: 42, Loss: 0.058320877144781305
Epoch: 43, Loss: 0.05812847951133533
Epoch: 44, Loss: 0.05792808673931691
Epoch: 45, Loss: 0.057731155306100845
Epoch: 46, Loss: 0.05755687313933816
Epoch: 47, Loss: 0.05729268945287913
Epoch: 48, Loss: 0.05709122971419002
Epoch: 49, Loss: 0.05691547229956916
Epoch: 50, Loss: 0.05673920998157447
Epoch: 51, Loss: 0.05660296061187577
Epoch: 52, Loss: 0.05645316853137363
Epoch: 53, Loss: 0.05627864126227391
Epoch: 54, Loss: 0.05614194484002402
Epoch: 55, Loss: 0.05598801289907778
Epoch: 56, Loss: 0.05584331855441079
Epoch: 57, Loss: 0.055669507851986134
Epoch: 58, Loss: 0.055541180360191705
Epoch: 59, Loss: 0.05543637007434407
Epoch: 60, Loss: 0.0553432156093945
Epoch: 61, Loss: 0.05522009437276131
Epoch: 62, Loss: 0.0550704578256599
Epoch: 63, Loss: 0.0549166586639805
Epoch: 64, Loss: 0.054772645009823304
Epoch: 65, Loss: 0.05463174149221943
Epoch: 66, Loss: 0.05448487985362906
Epoch: 67, Loss: 0.05439245556582413
Epoch: 68, Loss: 0.0542476438542666
Epoch: 69, Loss: 0.0540956300885781
Epoch: 70, Loss: 0.0539441496442558
2021-03-17 21:30:37,213 sagemaker-containers INFO     Reporting training SUCC
ESS

2021-03-17 21:31:02 Completed - Training job completed
```

```
Training seconds: 92
Billable seconds: 92
```

In [13]:

```
# deploy your model to create a predictor
%time
predictor = estimator.deploy(initial_instance_count=1,instance_type='ml.m4.xlarge')
```

```
CPU times: user 3 µs, sys: 1 µs, total: 4 µs
Wall time: 7.63 µs
-----------------!
```

In [14]:

```
# read in test data, assuming it is stored locally
test_data = pd.read_csv(os.path.join(data_dir, "test.csv"), header=None, names=None)
# labels are in the first column
test_y = test_data.iloc[:,0]
test_x = test_data.iloc[:,1:]
```

In [15]:

```
test_y_preds = predictor.predict(test_x.values.astype(np.float32))
test_y_preds = [round(num) for num in test_y_preds.squeeze()]
```

In [16]:

```
from sklearn.metrics import accuracy_score
train_data = pd.read_csv(os.path.join(data_dir, "train.csv"), header=None, names=None)
train_y = train_data.iloc[:,0]
train_x = train_data.iloc[:,1:]
train_y_preds = predictor.predict(train_x.values.astype(np.float32))
train_y_preds = [round(num) for num in train_y_preds.squeeze()]
accuracy = accuracy_score(train_y, train_y_preds)
print("Training accuracy %4.2f" % (100*accuracy), "%" )
```

```
Training accuracy 98.68 %
```

In [17]:

```
# Second: calculate the test accuracy
accuracy = accuracy_score(test_y, test_y_preds)

print("Test accuracy %4.2f %%" % (100*accuracy) )
```

```
Test accuracy 98.25 %
```

In [37]:

```
preds_train = np.reshape([train_y], (train_y.shape[0],1))
train       = np.reshape([train_y_preds], (train_y.shape[0],1) )
preds_test  = np.reshape([test_y], (test_y.shape[0],1))
test        = np.reshape([test_y_preds], (test_y.shape[0],1) )
pd.DataFrame(data = np.hstack((train, preds_train)),
             columns = ['original', 'model_output']).to_csv('modeloutput_train.csv')
pd.DataFrame(data = np.hstack((test, preds_test)),
             columns = ['original', 'model_output']).to_csv('modeloutput_test.csv')
```

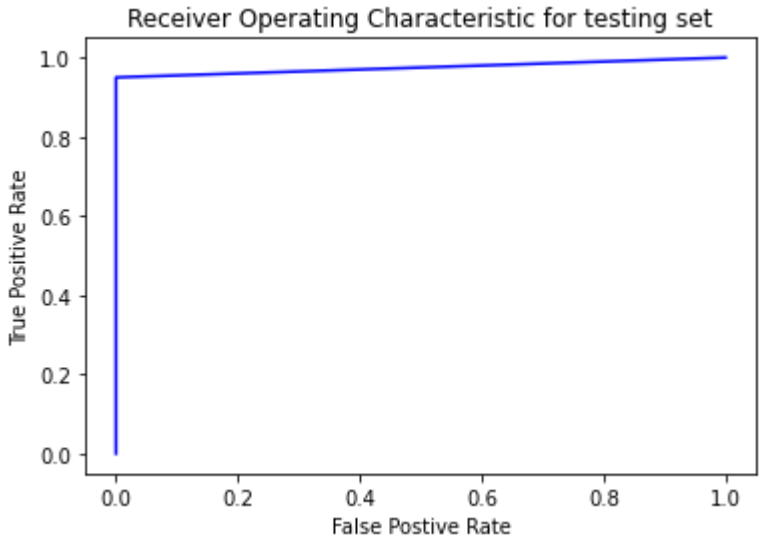
In [48]:

```
# Plot roc curve for testing set
import sklearn.metrics as metrics
fp, tp, threshold = metrics.roc_curve(test_y, test_y_preds)
roc = metrics.auc(fp, tp)

plt.title('Receiver Operating Characteristic for testing set')
plt.xlabel('False Postive Rate')
plt.ylabel('True Positive Rate')
plt.plot(fp, tp, 'b')
```

Out[48]:

```
[<matplotlib.lines.Line2D at 0x7f140d39d710>]
```



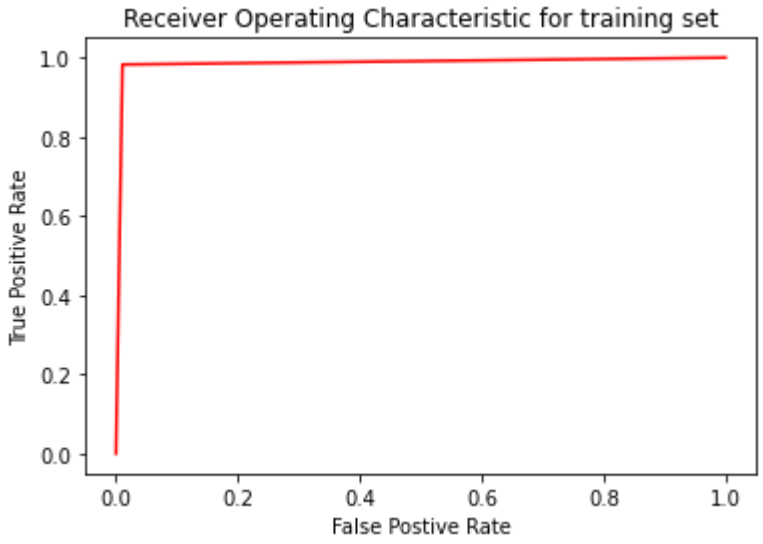
In [49]:

```
# Plot roc curve for training set
fp, tp, threshold = metrics.roc_curve(train_y, train_y_preds)
roc = metrics.auc(fp, tp)

plt.title('Receiver Operating Characteristic for training set')
plt.xlabel('False Postive Rate')
plt.ylabel('True Positive Rate')
plt.plot(fp, tp, 'r')
```

Out[49]:

```
[<matplotlib.lines.Line2D at 0x7f140d37a2e8>]
```



In [45]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(np.vstack([test, train]), np.vstack([preds_test, preds_train]))
```

Out[45]:

```
array([[354,   5],
       [  3, 207]])
```

In [ ]:

```

```