IMPORTING REQUIRED LIBRARIES

In [1]:
```
import sagemaker
import boto3
import os
import numpy as np
import pandas as pd
from sagemaker.serializers import CSVSerializer
from sagemaker.predictor import Predictor
from sagemaker.session import s3_input, Session
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
```

DOWNLOADING DATA FROM S3 BUCKET

In [2]:
```
bucket = 'mohan-cogintern-bucket'
subfolder = 'Assignment2.1/DATA'
df1=pd.read_csv("s3://mohan-cogintern-bucket/Assignment2.1/DATA/diab1.csv")
df2=pd.read_csv("s3://mohan-cogintern-bucket/Assignment2.1/DATA/diab2.csv")
```

COMBAINING DATA WHICH IS IN MULTIPLE FILES INTO A SINGLE DATAFRAME

In [3]:
```
df=pd.concat([df1,df2])
df.head(5)
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

SPLITTING DATA INTO TRAIN AND TEST

In [4]:
```
train_data, test_data = np.split(df.sample(frac=1, random_state=1729), [int(0.7 * len(df))])
print(train_data.shape, test_data.shape)
```

```
(537, 9) (231, 9)
```

UPLOADING TRAIN DATA AS CSV FILE INTO S3 BUCKET

In [5]:
```
train_data.to_csv('traindiab.csv',index=False,header=True)
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(subfolder, 'traindiab.csv')).upload_file('tra
```

UPLOADING TEST DATA AS CSV FILE INTO S3 BUCKET

In [6]:
```
test_data.to_csv('testdiab.csv',index=False,header=True)
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(subfolder, 'testdiab.csv')).upload_file('test
```

I HAVE CREATED AN AUTOML EXPERIMENT AND RUN THE EXP WITH TRAIN DATA. THE AUTOPIOLET HAS TESTED THE DATA WITH DIFFERENT MODELS AND FINALLY IT PRIDECTED THE BEST MODEL AS XGBOOST AND ACCURACY WAS 0.78. I DEPLOYED THE BEST MODEL WITH THE NAME "assignmnet2endpoint" AND TESTED THE MODEL WITH TEST DATA IN BELOW STEPS

TESTING THE DEPLOYED MODEL WITH TEST DATA

In [7]:
```
test_data_array = test_data.drop(['Outcome'], axis=1).values
y_true=test_data['Outcome']
plist=[]
predictor=Predictor('assignmnet2endpoint',sagemaker_session=sagemaker.Session(),serializer=CSVSerializer())
for i in range(len(test_data_array)):
    predictions=predictor.predict(test_data_array[i]).decode('utf-8')
    plist.append(int(predictions[0]))
print(plist)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1,
0, 0, 1, 0, 0, 1, 0, 0, 0]
```

CONFUSION MATRIX

In [8]:
```
CM = confusion_matrix(y_true, plist)
CM
```

Out[8]:
```
array([[132,  17],
       [ 38,  44]])
```

ACCURACY

In [9]:
```
print("ACCURACY : ",accuracy_score(y_true,plist))
```

```
ACCURACY :  0.7619047619047619
```

TESTING THE DEPLOYED MODEL WITH TRAIN DATA

In [10]:
```
train_data_array = train_data.drop(['Outcome'], axis=1).values
x_true=train_data['Outcome']
plist=[]
predictor=Predictor('assignmnet2endpoint',sagemaker_session=sagemaker.Session(),serializer=CSVSerializer())
for i in range(len(train_data_array)):
    predictions=predictor.predict(train_data_array[i]).decode('utf-8')
    plist.append(int(predictions[0]))
print(plist)
```

```
[0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0,
0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,
0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0,
0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1,
1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1]
```

CONFUSION MATRIX

In [11]:
```
CM = confusion_matrix(x_true, plist)
CM
```

Out[11]:
```
array([[324,  27],
       [ 32, 154]])
```

ACCURACY

In [12]:
```
print("ACCURACY : ",accuracy_score(x_true,plist))
```

```
ACCURACY :  0.8901303538175046
```