Helwan University

كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

Faculty of Computers and Artificial Intelligence

Computer Science Department

2021/2022

CS 395 Selected Topics in CS-1

Research Project

Report Submitted for Fulfillment of the Requirements and ILO's for Selected Topics in CS-1 course for Fall 2021

Team No. 58

	ID	Name	Grade
1.	201900997	يوسف مصطفى صابر عيد	Level 3
2.	201900818	مصطفى سيد مصطفى سيد	Level 3
3.	201900563	فاطمة رضا عبد المنعم عبد العزيز	Level 3
4.	201900203	ايمان أشرف احمد عبد العزيز	Level 3
5.	201900159	آلاء إبراهيم عبد العزيز محمد	Level 3
6.	201900751	محمد ياسر على محمد ابراهيم	Level 3
7.			

Delivered to:

Dr. Wessam El-Behaidy

Eng. Islam Gamal

Eng. Muhammed Kamal

I. NUMERICAL DATASET

1. Project Introduction

a. Dataset Name

Diabetes Dataset

b. Number of classes and their labels

The dataset has **9 classes**

(Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigree, Age, Outcome)

c. Dataset Samples Numbers

The total number of samples in dataset is **768**

d. Training, Validation and Testing

We split dataset to:

80% Training (614 samples)

20% validation from training (123 sample)

20% testing (154 sample)

2. Implementation Details

a. Extracted Features

Principal component analysis (PCA) in SVM model
This Extracted feature reduces the accuracy but allowed us to make the eight features only two and represent the result in plot

b. Artificial Neural Network (ANN)

℘ Hyper-parameters

I use activation functions for :

- input layer **linear**
- hidden layer **relu**
- output layer **sigmoid**

mean-squared error to loss & **adam** to optimizer

epochs=1000 & batch size=32

c. Support Vector Machine (SVM)

℘ Hyper-parameters

1-kernel = 'linear':

2-C=1:

-we used it to specify the class weight to 1; which is the optimal weight that gives the best accuracy for the model.

-in visualizing the training & test set results we used :

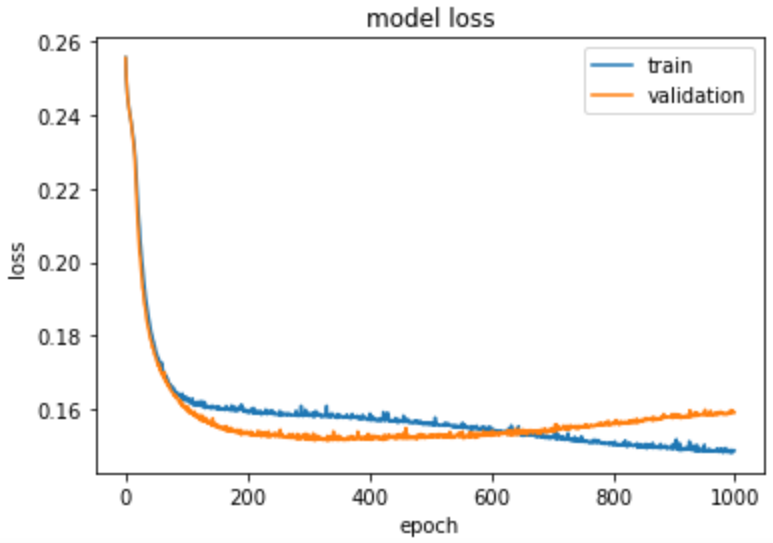
1-alpha = 0.75:

-it's used to set the learning rate (α) to the value we wanted which is .75.

3. Models Results

For each model you should show all these results for your model on testing data (loss curve, accuracy, confusion matrix, ROC curve)

a. ANN Results

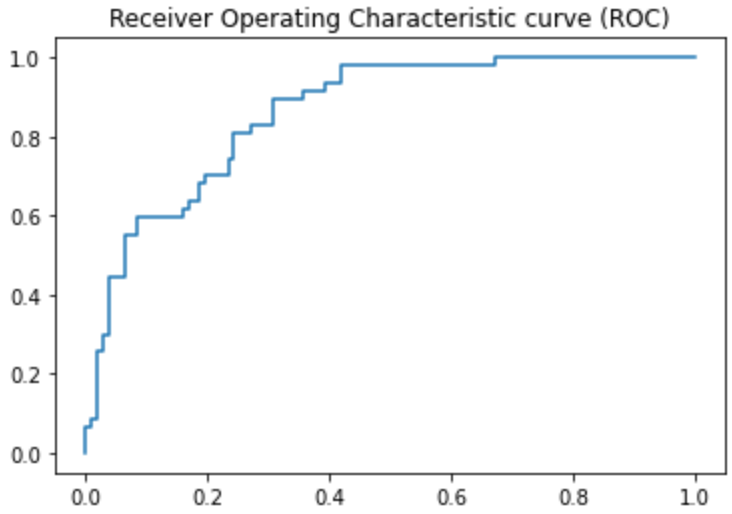


In [15]:
```
# loss
loss = ann.evaluate(X_test, Y_test)
print(loss , 'loss')
```

```
5/5 [==============================] - 0s 0s/step - loss: 0.1389
0.13887009024620056 loss
```

In [48]:
```
Y_pred = ann.predict(X_test)
Y_pred = (Y_pred > 0.5)

#confusion matrix & accuracy
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(Y_test, Y_pred)
print(cm)
print(accuracy_score(Y_test, Y_pred) ,'accuracy' )
```

```
[[95 12]
 [19 28]]
0.7987012987012987 accuracy
```

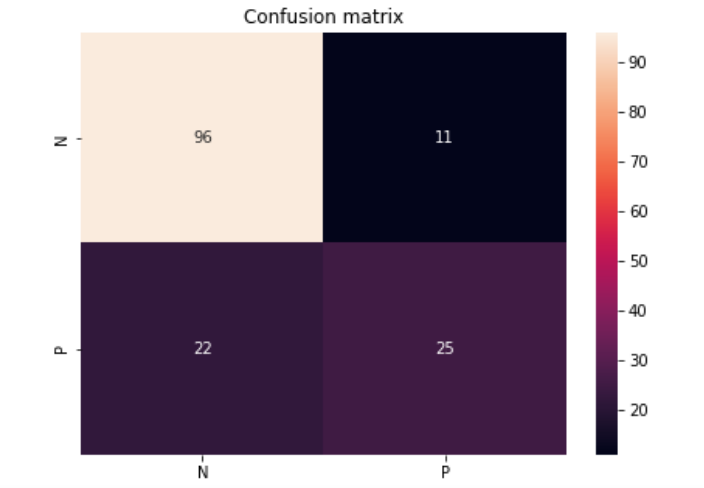


b.SVM Results

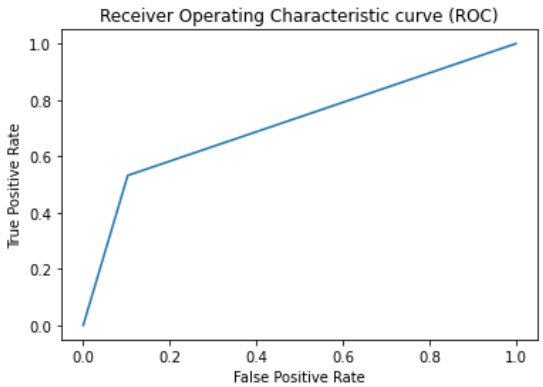
1-accuracy score:

```
In [19]: # Making the Confusion Matrix
         from sklearn.metrics import confusion_matrix, accuracy_score
         cm = confusion_matrix(y_test, y_pred)
         print(cm)
         accuracy_score(y_test, y_pred)

         [[96 11]
          [22 25]]

Out[19]: 0.7857142857142857
```

2-confusion matrix:



3-ROC:



II. IMAGE DATASET

1. Project Introduction

a. Dataset Name

Four Shapes

b. Number of classes and their labels

four shapes: square, star, circle, and triangle

c. Dataset Images Numbers and size

dataset contains 16,000 images
Each image is 200x200 pixels

d. Training, Validation and Testing

(The number of images used in training, validation and testing.)
We split dataset to:
80% Training
20% validation from training
20% testing

2. Implementation Details

a. Extracted Features

b. Artificial Neural Network (ANN)

℘ Hyper-parameters

(Specify all the hyper-parameters (initial learning rate, optimizer, regularization, batch size, no. of epochs…) with their specified value in implementation)

Activation: **relu, softmax**
optimizer = **adam**
loss = **sparse categorical crossentropy**
epochs=**5**
batch size=**32**

c. Support Vector Machine (SVM)

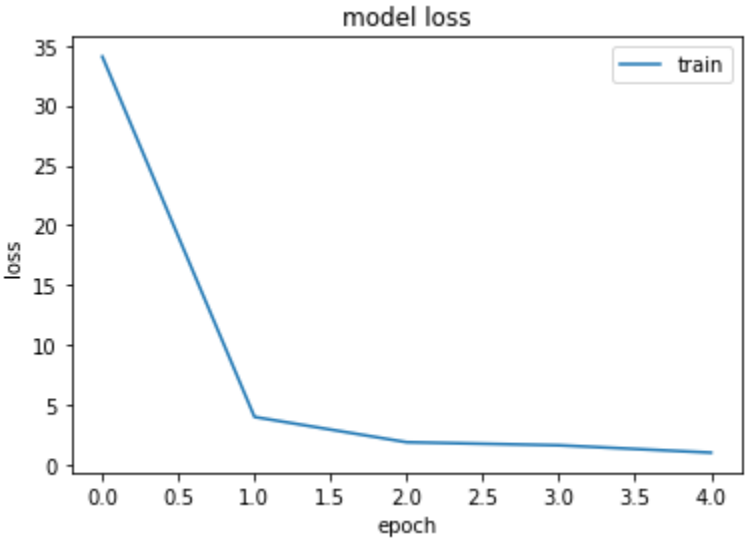
℘ Hyper-parameters

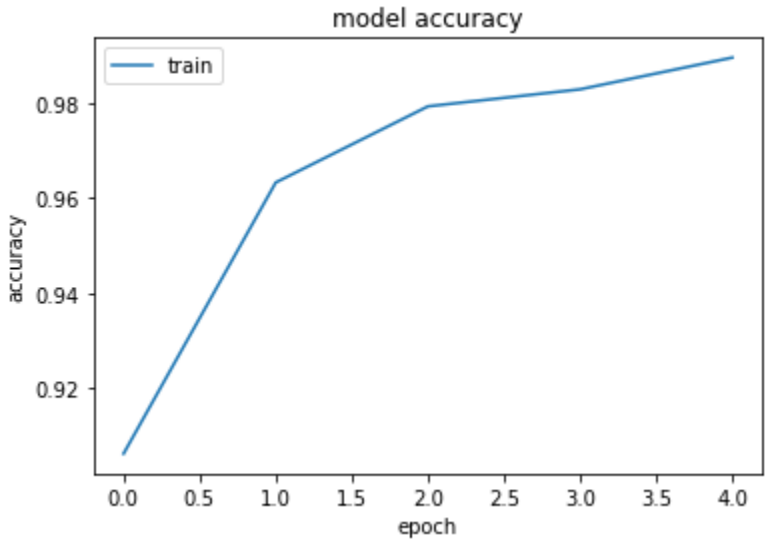
kernel= **linear**

3. Models Results

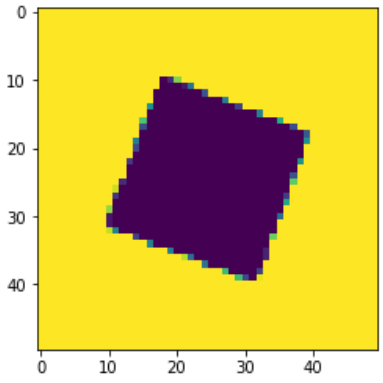
For each model you should show all these results for your model on testing data (loss curve, accuracy, confusion matrix, ROC curve)

a. ANN Results



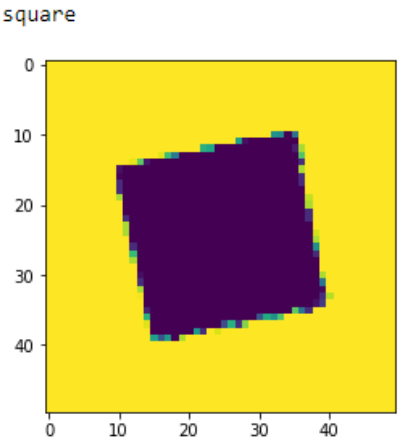



```
94/94 [==============================] - 0s 835us/step - loss: 0.1684 - accuracy: 0.9990
0.16840305924415588 0.9989979863166809
Accuracy 0.9989979863166809
prediction is :  square
```



b.SVM Results

1-accuracy score:




```
0.9993319973279893 accuracy
```

2-confusion matrix:

```
[[724   1   0   0]
 [  1 782   0   0]
 [  0   0 748   0]
 [  0   0   0 738]]
0.9993319973279893 accuracy
```