

In [239...

```
import numpy as np
from keras.models import Sequential
from keras.utils import np_utils
from keras.layers import Dense, Activation
import pandas as pd

import matplotlib.pyplot as plt
```

In [240...

```
df=pd.read_csv('EEGdata.csv')
df.head()
```

Out[240...

	A	X1	X2	X3	X4	X5	X6	X7	X8	X9	...	X170	X171	X172	X173	X174	X175	X'
0	X21.V1.791	135	190	229	223	192	125	55	-9	-33	...	-17	-15	-31	-77	-103	-127	-
1	X15.V1.924	386	382	356	331	320	315	307	272	244	...	164	150	146	152	157	156	
2	X8.V1.1	-32	-39	-47	-37	-32	-36	-57	-73	-85	...	57	64	48	19	-12	-30	
3	X16.V1.60	-105	-101	-96	-92	-89	-95	-102	-100	-87	...	-82	-81	-80	-77	-85	-77	
4	X20.V1.54	-9	-65	-98	-102	-78	-48	-16	0	-21	...	4	2	-12	-32	-41	-65	

5 rows × 180 columns



In [241...

```
X=df.values
```

In [242...

```
X=X[:,1:-1]
```

In [243...

```
X=df.values
X=X[:,1:-1]
X = np.asarray(X).astype(np.float32)
from sklearn.model_selection import train_test_split

y=np.array(df['y'])
Y=np_utils.to_categorical(y)
Y.shape
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20, random_state=1)
X_train = X_train.reshape(-1,178,1)
X_test = X_test.reshape(-1,178,1)

X_train.shape
X_test.shape
```

Out[243...] (2300, 178, 1)

In [244...

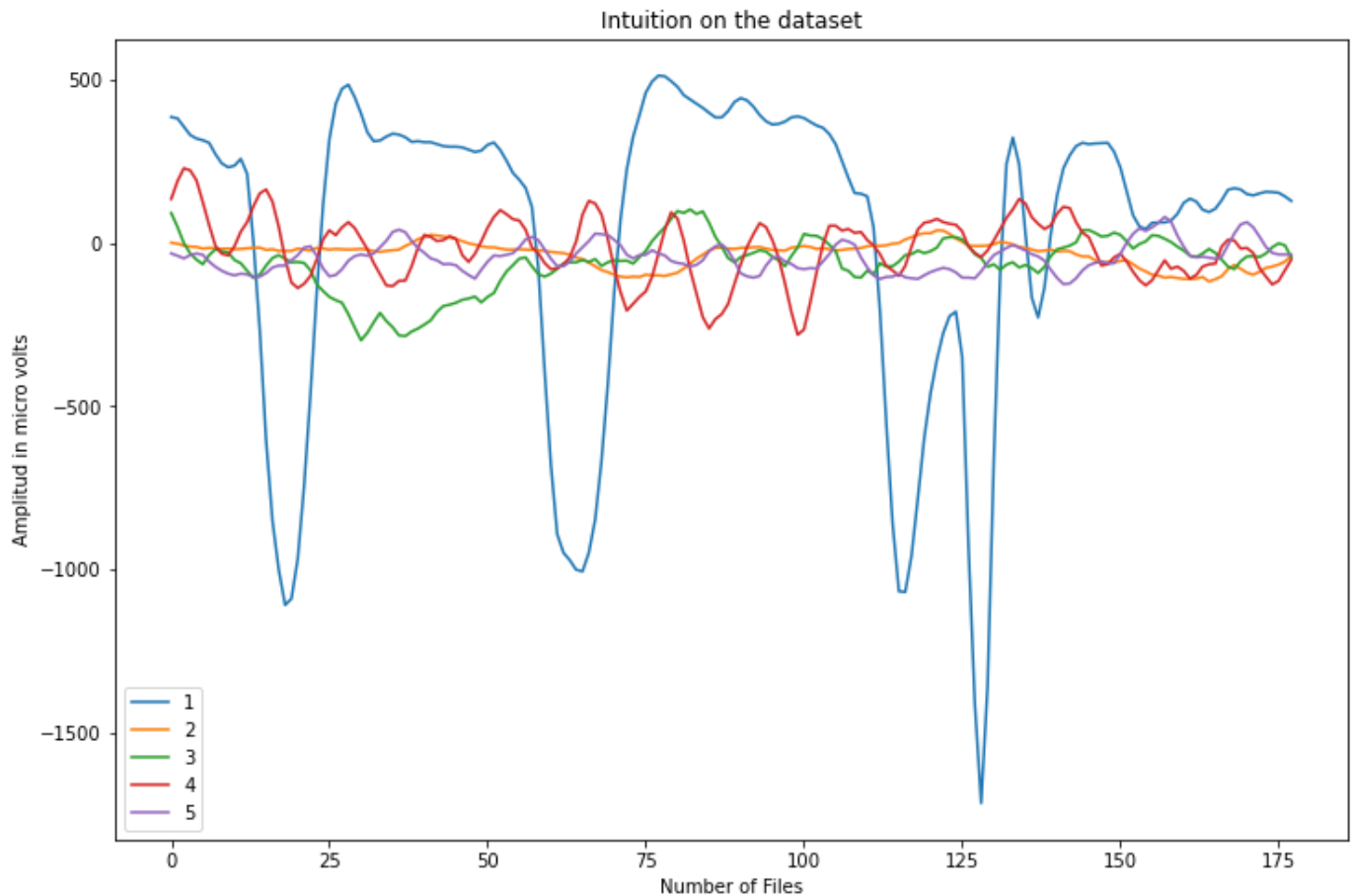
```
print(X.shape,y.shape)
```

(11500, 178) (11500,)

In [245...

```
plt.figure(figsize=(12,8))
plt.plot(X[1,:],label='1')
plt.plot(X[7,:],label='2')
plt.plot(X[12,:],label='3')
plt.plot(X[0,:],label='4')
plt.plot(X[2,:],label='5')
```

```
plt.legend()
plt.xlabel("Number of Files")
plt.ylabel("Amplitud in micro volts")
plt.title("Intuition on the dataset")
plt.show()
```



In [246...

```
model=Sequential()

model.add(Dense(256,input_shape=(45,)))
model.add(Activation('relu'))
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dense(32))
model.add(Activation('relu'))
model.add(Dense(32))
model.add(Activation('relu'))
model.add(Dense(32))
model.add(Activation('relu'))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dense(8))
model.add(Activation('relu'))
model.add(Dense(5))
model.add(Activation('softmax'))
model.summary()
```

Model: "sequential_7"

Layer (type)	Output Shape	Param #
dense_77 (Dense)	(None, 256)	11776

activation_77 (Activation)	(None, 256)	0
dense_78 (Dense)	(None, 128)	32896
activation_78 (Activation)	(None, 128)	0
dense_79 (Dense)	(None, 128)	16512
activation_79 (Activation)	(None, 128)	0
dense_80 (Dense)	(None, 64)	8256
activation_80 (Activation)	(None, 64)	0
dense_81 (Dense)	(None, 32)	2080
activation_81 (Activation)	(None, 32)	0
dense_82 (Dense)	(None, 32)	1056
activation_82 (Activation)	(None, 32)	0
dense_83 (Dense)	(None, 32)	1056
activation_83 (Activation)	(None, 32)	0
dense_84 (Dense)	(None, 16)	528
activation_84 (Activation)	(None, 16)	0
dense_85 (Dense)	(None, 16)	272
activation_85 (Activation)	(None, 16)	0
dense_86 (Dense)	(None, 8)	136
activation_86 (Activation)	(None, 8)	0
dense_87 (Dense)	(None, 5)	45
activation_87 (Activation)	(None, 5)	0

```

=====
Total params: 74,613
Trainable params: 74,613
Non-trainable params: 0

```

In [247...

```

#model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy', 'AUC', 'Recall',

```

In [248...

```

fits=model.fit(((X_train[:,::4]-X_train.mean())/X_train.std()),Y_train[:,1:],10,100,verbose=2,\

```

```

Epoch 1/100
920/920 - 4s - loss: 0.4089 - accuracy: 0.4513 - auc: 0.7840 - recall: 0.2084 - precision: 0.78
31 - val_loss: 0.3257 - val_accuracy: 0.5613 - val_auc: 0.8702 - val_recall: 0.3287 - val_preci
sion: 0.7762 - 4s/epoch - 5ms/step
Epoch 2/100
920/920 - 2s - loss: 0.2895 - accuracy: 0.6245 - auc: 0.9046 - recall: 0.4900 - precision: 0.71
71 - val_loss: 0.2722 - val_accuracy: 0.6387 - val_auc: 0.9173 - val_recall: 0.5952 - val_preci
sion: 0.6787 - 2s/epoch - 3ms/step
Epoch 3/100
920/920 - 2s - loss: 0.2483 - accuracy: 0.6642 - auc: 0.9304 - recall: 0.6140 - precision: 0.70
34 - val_loss: 0.2587 - val_accuracy: 0.6617 - val_auc: 0.9237 - val_recall: 0.6387 - val_preci
sion: 0.6782 - 2s/epoch - 3ms/step
Epoch 4/100
920/920 - 3s - loss: 0.2309 - accuracy: 0.6934 - auc: 0.9406 - recall: 0.6537 - precision: 0.72
42 - val_loss: 0.2634 - val_accuracy: 0.6583 - val_auc: 0.9219 - val_recall: 0.6065 - val_preci
sion: 0.7014 - 3s/epoch - 3ms/step
Epoch 5/100
920/920 - 2s - loss: 0.2192 - accuracy: 0.7040 - auc: 0.9463 - recall: 0.6698 - precision: 0.73
32 - val_loss: 0.2711 - val_accuracy: 0.6443 - val_auc: 0.9206 - val_recall: 0.5857 - val_preci

```

tion: 0.6824 - 2s/epoch - 3ms/step
Epoch 6/100
920/920 - 2s - loss: 0.2085 - accuracy: 0.7221 - auc: 0.9509 - recall: 0.6905 - precision: 0.7455 - val_loss: 0.2589 - val_accuracy: 0.6700 - val_auc: 0.9257 - val_recall: 0.6565 - val_precision: 0.6839 - 2s/epoch - 3ms/step
Epoch 7/100
920/920 - 3s - loss: 0.1961 - accuracy: 0.7411 - auc: 0.9569 - recall: 0.7179 - precision: 0.7593 - val_loss: 0.2527 - val_accuracy: 0.6696 - val_auc: 0.9257 - val_recall: 0.6478 - val_precision: 0.6882 - 3s/epoch - 3ms/step
Epoch 8/100
920/920 - 3s - loss: 0.1932 - accuracy: 0.7491 - auc: 0.9590 - recall: 0.7255 - precision: 0.7718 - val_loss: 0.2846 - val_accuracy: 0.6413 - val_auc: 0.9085 - val_recall: 0.6117 - val_precision: 0.6618 - 3s/epoch - 3ms/step
Epoch 9/100
920/920 - 2s - loss: 0.1786 - accuracy: 0.7659 - auc: 0.9628 - recall: 0.7472 - precision: 0.7838 - val_loss: 0.2748 - val_accuracy: 0.6583 - val_auc: 0.9145 - val_recall: 0.6413 - val_precision: 0.6829 - 2s/epoch - 3ms/step
Epoch 10/100
920/920 - 2s - loss: 0.1705 - accuracy: 0.7813 - auc: 0.9658 - recall: 0.7673 - precision: 0.7934 - val_loss: 0.2868 - val_accuracy: 0.6674 - val_auc: 0.9101 - val_recall: 0.6457 - val_precision: 0.6853 - 2s/epoch - 3ms/step
Epoch 11/100
920/920 - 2s - loss: 0.1622 - accuracy: 0.7941 - auc: 0.9686 - recall: 0.7807 - precision: 0.8047 - val_loss: 0.2802 - val_accuracy: 0.6839 - val_auc: 0.9148 - val_recall: 0.6722 - val_precision: 0.6977 - 2s/epoch - 3ms/step
Epoch 12/100
920/920 - 2s - loss: 0.1596 - accuracy: 0.8029 - auc: 0.9708 - recall: 0.7929 - precision: 0.8132 - val_loss: 0.3032 - val_accuracy: 0.6765 - val_auc: 0.9047 - val_recall: 0.6639 - val_precision: 0.6872 - 2s/epoch - 3ms/step
Epoch 13/100
920/920 - 2s - loss: 0.1484 - accuracy: 0.8170 - auc: 0.9737 - recall: 0.8080 - precision: 0.8254 - val_loss: 0.3002 - val_accuracy: 0.6852 - val_auc: 0.9089 - val_recall: 0.6752 - val_precision: 0.6949 - 2s/epoch - 3ms/step
Epoch 14/100
920/920 - 2s - loss: 0.1364 - accuracy: 0.8337 - auc: 0.9777 - recall: 0.8263 - precision: 0.8406 - val_loss: 0.3267 - val_accuracy: 0.6757 - val_auc: 0.8972 - val_recall: 0.6683 - val_precision: 0.6877 - 2s/epoch - 3ms/step
Epoch 15/100
920/920 - 2s - loss: 0.1322 - accuracy: 0.8426 - auc: 0.9800 - recall: 0.8347 - precision: 0.8502 - val_loss: 0.3221 - val_accuracy: 0.6948 - val_auc: 0.9048 - val_recall: 0.6835 - val_precision: 0.7052 - 2s/epoch - 3ms/step
Epoch 16/100
920/920 - 2s - loss: 0.1244 - accuracy: 0.8583 - auc: 0.9816 - recall: 0.8520 - precision: 0.8660 - val_loss: 0.3283 - val_accuracy: 0.6852 - val_auc: 0.8975 - val_recall: 0.6761 - val_precision: 0.6936 - 2s/epoch - 3ms/step
Epoch 17/100
920/920 - 2s - loss: 0.1154 - accuracy: 0.8709 - auc: 0.9839 - recall: 0.8679 - precision: 0.8762 - val_loss: 0.3402 - val_accuracy: 0.6870 - val_auc: 0.9005 - val_recall: 0.6774 - val_precision: 0.6931 - 2s/epoch - 3ms/step
Epoch 18/100
920/920 - 2s - loss: 0.1113 - accuracy: 0.8826 - auc: 0.9849 - recall: 0.8779 - precision: 0.8876 - val_loss: 0.3423 - val_accuracy: 0.6904 - val_auc: 0.8967 - val_recall: 0.6765 - val_precision: 0.7022 - 2s/epoch - 3ms/step
Epoch 19/100
920/920 - 2s - loss: 0.1060 - accuracy: 0.8870 - auc: 0.9855 - recall: 0.8817 - precision: 0.8930 - val_loss: 0.3508 - val_accuracy: 0.7009 - val_auc: 0.8968 - val_recall: 0.6943 - val_precision: 0.7073 - 2s/epoch - 3ms/step
Epoch 20/100
920/920 - 2s - loss: 0.0926 - accuracy: 0.9079 - auc: 0.9875 - recall: 0.9049 - precision: 0.9107 - val_loss: 0.3793 - val_accuracy: 0.6909 - val_auc: 0.8886 - val_recall: 0.6852 - val_precision: 0.6937 - 2s/epoch - 3ms/step
Epoch 21/100
920/920 - 2s - loss: 0.0948 - accuracy: 0.9038 - auc: 0.9877 - recall: 0.9001 - precision: 0.9092 - val_loss: 0.3900 - val_accuracy: 0.6878 - val_auc: 0.8826 - val_recall: 0.6830 - val_precision: 0.6936 - 2s/epoch - 3ms/step
Epoch 22/100
920/920 - 2s - loss: 0.0789 - accuracy: 0.9215 - auc: 0.9898 - recall: 0.9186 - precision: 0.9255 - val_loss: 0.4031 - val_accuracy: 0.7052 - val_auc: 0.8808 - val_recall: 0.7009 - val_precision: 0.7083 - 2s/epoch - 3ms/step
Epoch 23/100
920/920 - 2s - loss: 0.0701 - accuracy: 0.9341 - auc: 0.9902 - recall: 0.9323 - precision: 0.9368 - val_loss: 0.3879 - val_accuracy: 0.7143 - val_auc: 0.8895 - val_recall: 0.7096 - val_precision: 0.7183 - 2s/epoch - 3ms/step
Epoch 24/100

920/920 - 2s - loss: 0.0678 - accuracy: 0.9347 - auc: 0.9912 - recall: 0.9322 - precision: 0.9381 - val_loss: 0.3866 - val_accuracy: 0.6830 - val_auc: 0.8835 - val_recall: 0.6800 - val_precision: 0.6884 - 2s/epoch - 3ms/step
Epoch 25/100
920/920 - 3s - loss: 0.0598 - accuracy: 0.9413 - auc: 0.9925 - recall: 0.9402 - precision: 0.9434 - val_loss: 0.4818 - val_accuracy: 0.6857 - val_auc: 0.8671 - val_recall: 0.6804 - val_precision: 0.6897 - 3s/epoch - 3ms/step
Epoch 26/100
920/920 - 2s - loss: 0.0645 - accuracy: 0.9404 - auc: 0.9913 - recall: 0.9391 - precision: 0.9430 - val_loss: 0.4192 - val_accuracy: 0.6887 - val_auc: 0.8820 - val_recall: 0.6796 - val_precision: 0.6940 - 2s/epoch - 3ms/step
Epoch 27/100
920/920 - 3s - loss: 0.0546 - accuracy: 0.9492 - auc: 0.9928 - recall: 0.9480 - precision: 0.9516 - val_loss: 0.4415 - val_accuracy: 0.6991 - val_auc: 0.8743 - val_recall: 0.6974 - val_precision: 0.7020 - 3s/epoch - 3ms/step
Epoch 28/100
920/920 - 2s - loss: 0.0551 - accuracy: 0.9508 - auc: 0.9936 - recall: 0.9496 - precision: 0.9533 - val_loss: 0.4469 - val_accuracy: 0.7017 - val_auc: 0.8716 - val_recall: 0.6987 - val_precision: 0.7045 - 2s/epoch - 3ms/step
Epoch 29/100
920/920 - 2s - loss: 0.0488 - accuracy: 0.9547 - auc: 0.9936 - recall: 0.9528 - precision: 0.9565 - val_loss: 0.5015 - val_accuracy: 0.7009 - val_auc: 0.8682 - val_recall: 0.6991 - val_precision: 0.7046 - 2s/epoch - 3ms/step
Epoch 30/100
920/920 - 2s - loss: 0.0467 - accuracy: 0.9586 - auc: 0.9939 - recall: 0.9574 - precision: 0.9613 - val_loss: 0.5454 - val_accuracy: 0.6917 - val_auc: 0.8583 - val_recall: 0.6900 - val_precision: 0.6942 - 2s/epoch - 3ms/step
Epoch 31/100
920/920 - 2s - loss: 0.0442 - accuracy: 0.9622 - auc: 0.9938 - recall: 0.9610 - precision: 0.9648 - val_loss: 0.5170 - val_accuracy: 0.6887 - val_auc: 0.8601 - val_recall: 0.6883 - val_precision: 0.6916 - 2s/epoch - 3ms/step
Epoch 32/100
920/920 - 2s - loss: 0.0462 - accuracy: 0.9599 - auc: 0.9935 - recall: 0.9582 - precision: 0.9618 - val_loss: 0.4620 - val_accuracy: 0.6939 - val_auc: 0.8776 - val_recall: 0.6900 - val_precision: 0.6997 - 2s/epoch - 3ms/step
Epoch 33/100
920/920 - 2s - loss: 0.0455 - accuracy: 0.9608 - auc: 0.9937 - recall: 0.9591 - precision: 0.9625 - val_loss: 0.4299 - val_accuracy: 0.7122 - val_auc: 0.8787 - val_recall: 0.7091 - val_precision: 0.7172 - 2s/epoch - 3ms/step
Epoch 34/100
920/920 - 2s - loss: 0.0340 - accuracy: 0.9698 - auc: 0.9955 - recall: 0.9690 - precision: 0.9708 - val_loss: 0.5079 - val_accuracy: 0.7074 - val_auc: 0.8675 - val_recall: 0.7052 - val_precision: 0.7102 - 2s/epoch - 3ms/step
Epoch 35/100
920/920 - 2s - loss: 0.0344 - accuracy: 0.9700 - auc: 0.9954 - recall: 0.9691 - precision: 0.9716 - val_loss: 0.5258 - val_accuracy: 0.7057 - val_auc: 0.8652 - val_recall: 0.7048 - val_precision: 0.7088 - 2s/epoch - 3ms/step
Epoch 36/100
920/920 - 2s - loss: 0.0315 - accuracy: 0.9715 - auc: 0.9959 - recall: 0.9710 - precision: 0.9727 - val_loss: 0.5562 - val_accuracy: 0.7074 - val_auc: 0.8630 - val_recall: 0.7070 - val_precision: 0.7116 - 2s/epoch - 3ms/step
Epoch 37/100
920/920 - 2s - loss: 0.0358 - accuracy: 0.9691 - auc: 0.9954 - recall: 0.9684 - precision: 0.9710 - val_loss: 0.5564 - val_accuracy: 0.7196 - val_auc: 0.8664 - val_recall: 0.7157 - val_precision: 0.7219 - 2s/epoch - 3ms/step
Epoch 38/100
920/920 - 2s - loss: 0.0346 - accuracy: 0.9747 - auc: 0.9963 - recall: 0.9735 - precision: 0.9765 - val_loss: 0.5153 - val_accuracy: 0.7057 - val_auc: 0.8703 - val_recall: 0.7017 - val_precision: 0.7095 - 2s/epoch - 3ms/step
Epoch 39/100
920/920 - 2s - loss: 0.0319 - accuracy: 0.9741 - auc: 0.9956 - recall: 0.9729 - precision: 0.9748 - val_loss: 0.5493 - val_accuracy: 0.7109 - val_auc: 0.8597 - val_recall: 0.7096 - val_precision: 0.7133 - 2s/epoch - 3ms/step
Epoch 40/100
920/920 - 2s - loss: 0.0274 - accuracy: 0.9770 - auc: 0.9963 - recall: 0.9760 - precision: 0.9780 - val_loss: 0.5048 - val_accuracy: 0.7061 - val_auc: 0.8642 - val_recall: 0.7043 - val_precision: 0.7115 - 2s/epoch - 3ms/step
Epoch 41/100
920/920 - 2s - loss: 0.0280 - accuracy: 0.9764 - auc: 0.9958 - recall: 0.9757 - precision: 0.9770 - val_loss: 0.5286 - val_accuracy: 0.7070 - val_auc: 0.8592 - val_recall: 0.7026 - val_precision: 0.7069 - 2s/epoch - 3ms/step
Epoch 42/100
920/920 - 2s - loss: 0.0262 - accuracy: 0.9767 - auc: 0.9968 - recall: 0.9762 - precision: 0.9774 - val_loss: 0.5658 - val_accuracy: 0.7126 - val_auc: 0.8680 - val_recall: 0.7117 - val_precision:

sion: 0.7164 - 2s/epoch - 3ms/step
Epoch 43/100
920/920 - 2s - loss: 0.0322 - accuracy: 0.9743 - auc: 0.9956 - recall: 0.9736 - precision: 0.9752 - val_loss: 0.4839 - val_accuracy: 0.7122 - val_auc: 0.8652 - val_recall: 0.7087 - val_precision: 0.7159 - 2s/epoch - 3ms/step
Epoch 44/100
920/920 - 2s - loss: 0.0263 - accuracy: 0.9782 - auc: 0.9960 - recall: 0.9776 - precision: 0.9789 - val_loss: 0.5340 - val_accuracy: 0.7022 - val_auc: 0.8603 - val_recall: 0.6996 - val_precision: 0.7035 - 2s/epoch - 3ms/step
Epoch 45/100
920/920 - 2s - loss: 0.0235 - accuracy: 0.9811 - auc: 0.9967 - recall: 0.9808 - precision: 0.9817 - val_loss: 0.5518 - val_accuracy: 0.7017 - val_auc: 0.8651 - val_recall: 0.7000 - val_precision: 0.7043 - 2s/epoch - 3ms/step
Epoch 46/100
920/920 - 2s - loss: 0.0272 - accuracy: 0.9791 - auc: 0.9958 - recall: 0.9788 - precision: 0.9795 - val_loss: 0.4963 - val_accuracy: 0.7083 - val_auc: 0.8671 - val_recall: 0.7070 - val_precision: 0.7100 - 2s/epoch - 3ms/step
Epoch 47/100
920/920 - 3s - loss: 0.0239 - accuracy: 0.9801 - auc: 0.9970 - recall: 0.9798 - precision: 0.9804 - val_loss: 0.5221 - val_accuracy: 0.7000 - val_auc: 0.8624 - val_recall: 0.6965 - val_precision: 0.7039 - 3s/epoch - 3ms/step
Epoch 48/100
920/920 - 2s - loss: 0.0252 - accuracy: 0.9811 - auc: 0.9963 - recall: 0.9805 - precision: 0.9817 - val_loss: 0.5424 - val_accuracy: 0.6957 - val_auc: 0.8584 - val_recall: 0.6952 - val_precision: 0.6989 - 2s/epoch - 3ms/step
Epoch 49/100
920/920 - 2s - loss: 0.0277 - accuracy: 0.9761 - auc: 0.9964 - recall: 0.9757 - precision: 0.9768 - val_loss: 0.5433 - val_accuracy: 0.7035 - val_auc: 0.8615 - val_recall: 0.7013 - val_precision: 0.7044 - 2s/epoch - 3ms/step
Epoch 50/100
920/920 - 2s - loss: 0.0225 - accuracy: 0.9824 - auc: 0.9969 - recall: 0.9823 - precision: 0.9832 - val_loss: 0.5604 - val_accuracy: 0.7083 - val_auc: 0.8679 - val_recall: 0.7070 - val_precision: 0.7104 - 2s/epoch - 3ms/step
Epoch 51/100
920/920 - 2s - loss: 0.0231 - accuracy: 0.9821 - auc: 0.9970 - recall: 0.9815 - precision: 0.9830 - val_loss: 0.5977 - val_accuracy: 0.7104 - val_auc: 0.8600 - val_recall: 0.7083 - val_precision: 0.7117 - 2s/epoch - 3ms/step
Epoch 52/100
920/920 - 2s - loss: 0.0227 - accuracy: 0.9813 - auc: 0.9968 - recall: 0.9809 - precision: 0.9816 - val_loss: 0.4914 - val_accuracy: 0.7143 - val_auc: 0.8732 - val_recall: 0.7126 - val_precision: 0.7185 - 2s/epoch - 3ms/step
Epoch 53/100
920/920 - 2s - loss: 0.0197 - accuracy: 0.9828 - auc: 0.9979 - recall: 0.9823 - precision: 0.9837 - val_loss: 0.5781 - val_accuracy: 0.7048 - val_auc: 0.8592 - val_recall: 0.7013 - val_precision: 0.7071 - 2s/epoch - 3ms/step
Epoch 54/100
920/920 - 2s - loss: 0.0200 - accuracy: 0.9835 - auc: 0.9971 - recall: 0.9835 - precision: 0.9839 - val_loss: 0.5778 - val_accuracy: 0.7170 - val_auc: 0.8638 - val_recall: 0.7152 - val_precision: 0.7205 - 2s/epoch - 3ms/step
Epoch 55/100
920/920 - 2s - loss: 0.0212 - accuracy: 0.9826 - auc: 0.9973 - recall: 0.9821 - precision: 0.9829 - val_loss: 0.5578 - val_accuracy: 0.7074 - val_auc: 0.8567 - val_recall: 0.7039 - val_precision: 0.7107 - 2s/epoch - 3ms/step
Epoch 56/100
920/920 - 2s - loss: 0.0207 - accuracy: 0.9836 - auc: 0.9974 - recall: 0.9828 - precision: 0.9845 - val_loss: 0.6108 - val_accuracy: 0.7283 - val_auc: 0.8613 - val_recall: 0.7270 - val_precision: 0.7317 - 2s/epoch - 3ms/step
Epoch 57/100
920/920 - 3s - loss: 0.0209 - accuracy: 0.9823 - auc: 0.9971 - recall: 0.9818 - precision: 0.9828 - val_loss: 0.4830 - val_accuracy: 0.7165 - val_auc: 0.8738 - val_recall: 0.7126 - val_precision: 0.7198 - 3s/epoch - 3ms/step
Epoch 58/100
920/920 - 3s - loss: 0.0221 - accuracy: 0.9816 - auc: 0.9972 - recall: 0.9808 - precision: 0.9833 - val_loss: 0.5774 - val_accuracy: 0.7161 - val_auc: 0.8601 - val_recall: 0.7143 - val_precision: 0.7190 - 3s/epoch - 3ms/step
Epoch 59/100
920/920 - 3s - loss: 0.0192 - accuracy: 0.9841 - auc: 0.9978 - recall: 0.9836 - precision: 0.9850 - val_loss: 0.6630 - val_accuracy: 0.7017 - val_auc: 0.8562 - val_recall: 0.6983 - val_precision: 0.7032 - 3s/epoch - 3ms/step
Epoch 60/100
920/920 - 3s - loss: 0.0158 - accuracy: 0.9886 - auc: 0.9975 - recall: 0.9884 - precision: 0.9891 - val_loss: 0.5846 - val_accuracy: 0.7117 - val_auc: 0.8564 - val_recall: 0.7104 - val_precision: 0.7135 - 3s/epoch - 3ms/step
Epoch 61/100

920/920 - 3s - loss: 0.0236 - accuracy: 0.9824 - auc: 0.9964 - recall: 0.9821 - precision: 0.9828 - val_loss: 0.6338 - val_accuracy: 0.7043 - val_auc: 0.8573 - val_recall: 0.7030 - val_precision: 0.7070 - 3s/epoch - 3ms/step
Epoch 62/100
920/920 - 3s - loss: 0.0161 - accuracy: 0.9868 - auc: 0.9980 - recall: 0.9864 - precision: 0.9875 - val_loss: 0.5573 - val_accuracy: 0.7035 - val_auc: 0.8534 - val_recall: 0.7026 - val_precision: 0.7063 - 3s/epoch - 3ms/step
Epoch 63/100
920/920 - 3s - loss: 0.0212 - accuracy: 0.9839 - auc: 0.9973 - recall: 0.9834 - precision: 0.9847 - val_loss: 0.6208 - val_accuracy: 0.7022 - val_auc: 0.8549 - val_recall: 0.6983 - val_precision: 0.7038 - 3s/epoch - 3ms/step
Epoch 64/100
920/920 - 3s - loss: 0.0141 - accuracy: 0.9877 - auc: 0.9982 - recall: 0.9876 - precision: 0.9880 - val_loss: 0.6596 - val_accuracy: 0.7104 - val_auc: 0.8570 - val_recall: 0.7091 - val_precision: 0.7135 - 3s/epoch - 3ms/step
Epoch 65/100
920/920 - 3s - loss: 0.0261 - accuracy: 0.9809 - auc: 0.9963 - recall: 0.9804 - precision: 0.9816 - val_loss: 0.5594 - val_accuracy: 0.7039 - val_auc: 0.8587 - val_recall: 0.7013 - val_precision: 0.7071 - 3s/epoch - 3ms/step
Epoch 66/100
920/920 - 3s - loss: 0.0173 - accuracy: 0.9873 - auc: 0.9979 - recall: 0.9866 - precision: 0.9883 - val_loss: 0.6227 - val_accuracy: 0.7148 - val_auc: 0.8568 - val_recall: 0.7130 - val_precision: 0.7187 - 3s/epoch - 3ms/step
Epoch 67/100
920/920 - 3s - loss: 0.0205 - accuracy: 0.9849 - auc: 0.9972 - recall: 0.9849 - precision: 0.9854 - val_loss: 0.5761 - val_accuracy: 0.7183 - val_auc: 0.8652 - val_recall: 0.7174 - val_precision: 0.7205 - 3s/epoch - 3ms/step
Epoch 68/100
920/920 - 3s - loss: 0.0106 - accuracy: 0.9920 - auc: 0.9987 - recall: 0.9917 - precision: 0.9921 - val_loss: 0.7098 - val_accuracy: 0.7065 - val_auc: 0.8491 - val_recall: 0.7061 - val_precision: 0.7076 - 3s/epoch - 3ms/step
Epoch 69/100
920/920 - 3s - loss: 0.0187 - accuracy: 0.9858 - auc: 0.9979 - recall: 0.9853 - precision: 0.9862 - val_loss: 0.7470 - val_accuracy: 0.7096 - val_auc: 0.8518 - val_recall: 0.7087 - val_precision: 0.7118 - 3s/epoch - 3ms/step
Epoch 70/100
920/920 - 3s - loss: 0.0323 - accuracy: 0.9832 - auc: 0.9964 - recall: 0.9815 - precision: 0.9838 - val_loss: 0.5107 - val_accuracy: 0.7074 - val_auc: 0.8632 - val_recall: 0.7035 - val_precision: 0.7109 - 3s/epoch - 3ms/step
Epoch 71/100
920/920 - 3s - loss: 0.0177 - accuracy: 0.9861 - auc: 0.9977 - recall: 0.9852 - precision: 0.9874 - val_loss: 0.6395 - val_accuracy: 0.7239 - val_auc: 0.8623 - val_recall: 0.7217 - val_precision: 0.7281 - 3s/epoch - 3ms/step
Epoch 72/100
920/920 - 3s - loss: 0.0172 - accuracy: 0.9867 - auc: 0.9972 - recall: 0.9865 - precision: 0.9871 - val_loss: 0.5717 - val_accuracy: 0.7122 - val_auc: 0.8609 - val_recall: 0.7083 - val_precision: 0.7154 - 3s/epoch - 3ms/step
Epoch 73/100
920/920 - 3s - loss: 0.0116 - accuracy: 0.9908 - auc: 0.9984 - recall: 0.9907 - precision: 0.9913 - val_loss: 0.6087 - val_accuracy: 0.7200 - val_auc: 0.8646 - val_recall: 0.7187 - val_precision: 0.7231 - 3s/epoch - 3ms/step
Epoch 74/100
920/920 - 3s - loss: 0.0188 - accuracy: 0.9851 - auc: 0.9976 - recall: 0.9846 - precision: 0.9855 - val_loss: 0.6539 - val_accuracy: 0.7213 - val_auc: 0.8603 - val_recall: 0.7191 - val_precision: 0.7248 - 3s/epoch - 3ms/step
Epoch 75/100
920/920 - 3s - loss: 0.0141 - accuracy: 0.9888 - auc: 0.9980 - recall: 0.9883 - precision: 0.9891 - val_loss: 0.6433 - val_accuracy: 0.7143 - val_auc: 0.8584 - val_recall: 0.7139 - val_precision: 0.7155 - 3s/epoch - 4ms/step
Epoch 76/100
920/920 - 3s - loss: 0.0150 - accuracy: 0.9882 - auc: 0.9982 - recall: 0.9879 - precision: 0.9890 - val_loss: 0.5979 - val_accuracy: 0.7139 - val_auc: 0.8589 - val_recall: 0.7117 - val_precision: 0.7167 - 3s/epoch - 4ms/step
Epoch 77/100
920/920 - 3s - loss: 0.0137 - accuracy: 0.9883 - auc: 0.9983 - recall: 0.9872 - precision: 0.9889 - val_loss: 0.7168 - val_accuracy: 0.7143 - val_auc: 0.8518 - val_recall: 0.7139 - val_precision: 0.7155 - 3s/epoch - 3ms/step
Epoch 78/100
920/920 - 2s - loss: 0.0141 - accuracy: 0.9878 - auc: 0.9979 - recall: 0.9876 - precision: 0.9881 - val_loss: 0.7788 - val_accuracy: 0.7043 - val_auc: 0.8463 - val_recall: 0.7035 - val_precision: 0.7053 - 2s/epoch - 3ms/step
Epoch 79/100
920/920 - 2s - loss: 0.0243 - accuracy: 0.9812 - auc: 0.9968 - recall: 0.9810 - precision: 0.9824 - val_loss: 0.5528 - val_accuracy: 0.7126 - val_auc: 0.8626 - val_recall: 0.7074 - val_precision:

920/920 - 2s - loss: 0.7171 - accuracy: 0.9895 - auc: 0.9984 - recall: 0.9886 - precision: 0.9898 - val_loss: 0.6242 - val_accuracy: 0.7035 - val_auc: 0.8572 - val_recall: 0.7013 - val_precision: 0.7059 - 2s/epoch - 3ms/step
Epoch 80/100
920/920 - 2s - loss: 0.0128 - accuracy: 0.9895 - auc: 0.9984 - recall: 0.9886 - precision: 0.9898 - val_loss: 0.6242 - val_accuracy: 0.7035 - val_auc: 0.8572 - val_recall: 0.7013 - val_precision: 0.7059 - 2s/epoch - 3ms/step
Epoch 81/100
920/920 - 2s - loss: 0.0149 - accuracy: 0.9896 - auc: 0.9979 - recall: 0.9896 - precision: 0.9899 - val_loss: 0.6389 - val_accuracy: 0.7152 - val_auc: 0.8539 - val_recall: 0.7117 - val_precision: 0.7170 - 2s/epoch - 3ms/step
Epoch 82/100
920/920 - 2s - loss: 0.0150 - accuracy: 0.9895 - auc: 0.9979 - recall: 0.9893 - precision: 0.9896 - val_loss: 0.7452 - val_accuracy: 0.7183 - val_auc: 0.8533 - val_recall: 0.7165 - val_precision: 0.7209 - 2s/epoch - 3ms/step
Epoch 83/100
920/920 - 2s - loss: 0.0131 - accuracy: 0.9900 - auc: 0.9983 - recall: 0.9899 - precision: 0.9901 - val_loss: 0.6388 - val_accuracy: 0.7130 - val_auc: 0.8580 - val_recall: 0.7113 - val_precision: 0.7160 - 2s/epoch - 3ms/step
Epoch 84/100
920/920 - 3s - loss: 0.0185 - accuracy: 0.9855 - auc: 0.9976 - recall: 0.9846 - precision: 0.9863 - val_loss: 0.6221 - val_accuracy: 0.7052 - val_auc: 0.8594 - val_recall: 0.7026 - val_precision: 0.7094 - 3s/epoch - 3ms/step
Epoch 85/100
920/920 - 2s - loss: 0.0116 - accuracy: 0.9903 - auc: 0.9983 - recall: 0.9902 - precision: 0.9908 - val_loss: 0.7014 - val_accuracy: 0.7109 - val_auc: 0.8520 - val_recall: 0.7104 - val_precision: 0.7138 - 2s/epoch - 3ms/step
Epoch 86/100
920/920 - 2s - loss: 0.0161 - accuracy: 0.9875 - auc: 0.9975 - recall: 0.9871 - precision: 0.9877 - val_loss: 0.6121 - val_accuracy: 0.7165 - val_auc: 0.8585 - val_recall: 0.7143 - val_precision: 0.7184 - 2s/epoch - 3ms/step
Epoch 87/100
920/920 - 3s - loss: 0.0140 - accuracy: 0.9907 - auc: 0.9981 - recall: 0.9904 - precision: 0.9909 - val_loss: 0.6906 - val_accuracy: 0.7000 - val_auc: 0.8550 - val_recall: 0.6983 - val_precision: 0.7010 - 3s/epoch - 3ms/step
Epoch 88/100
920/920 - 3s - loss: 0.0163 - accuracy: 0.9885 - auc: 0.9978 - recall: 0.9882 - precision: 0.9890 - val_loss: 0.6046 - val_accuracy: 0.7074 - val_auc: 0.8496 - val_recall: 0.7048 - val_precision: 0.7116 - 3s/epoch - 4ms/step
Epoch 89/100
920/920 - 2s - loss: 0.0153 - accuracy: 0.9893 - auc: 0.9981 - recall: 0.9888 - precision: 0.9901 - val_loss: 0.6373 - val_accuracy: 0.7213 - val_auc: 0.8606 - val_recall: 0.7187 - val_precision: 0.7237 - 2s/epoch - 3ms/step
Epoch 90/100
920/920 - 3s - loss: 0.0152 - accuracy: 0.9896 - auc: 0.9985 - recall: 0.9892 - precision: 0.9898 - val_loss: 0.6620 - val_accuracy: 0.7065 - val_auc: 0.8556 - val_recall: 0.7043 - val_precision: 0.7074 - 3s/epoch - 3ms/step
Epoch 91/100
920/920 - 2s - loss: 0.0128 - accuracy: 0.9888 - auc: 0.9982 - recall: 0.9887 - precision: 0.9891 - val_loss: 0.7428 - val_accuracy: 0.7209 - val_auc: 0.8572 - val_recall: 0.7191 - val_precision: 0.7213 - 2s/epoch - 3ms/step
Epoch 92/100
920/920 - 3s - loss: 0.0081 - accuracy: 0.9948 - auc: 0.9988 - recall: 0.9946 - precision: 0.9950 - val_loss: 0.8174 - val_accuracy: 0.7170 - val_auc: 0.8459 - val_recall: 0.7152 - val_precision: 0.7190 - 3s/epoch - 3ms/step
Epoch 93/100
920/920 - 2s - loss: 0.0084 - accuracy: 0.9943 - auc: 0.9984 - recall: 0.9941 - precision: 0.9947 - val_loss: 0.6210 - val_accuracy: 0.7109 - val_auc: 0.8530 - val_recall: 0.7070 - val_precision: 0.7154 - 2s/epoch - 3ms/step
Epoch 94/100
920/920 - 2s - loss: 0.0169 - accuracy: 0.9876 - auc: 0.9977 - recall: 0.9867 - precision: 0.9879 - val_loss: 0.6040 - val_accuracy: 0.7104 - val_auc: 0.8596 - val_recall: 0.7078 - val_precision: 0.7122 - 2s/epoch - 3ms/step
Epoch 95/100
920/920 - 2s - loss: 0.0159 - accuracy: 0.9883 - auc: 0.9974 - recall: 0.9880 - precision: 0.9891 - val_loss: 0.7407 - val_accuracy: 0.7113 - val_auc: 0.8521 - val_recall: 0.7057 - val_precision: 0.7125 - 2s/epoch - 3ms/step
Epoch 96/100
920/920 - 2s - loss: 0.0224 - accuracy: 0.9853 - auc: 0.9975 - recall: 0.9843 - precision: 0.9869 - val_loss: 0.5628 - val_accuracy: 0.7183 - val_auc: 0.8608 - val_recall: 0.7130 - val_precision: 0.7206 - 2s/epoch - 3ms/step
Epoch 97/100
920/920 - 2s - loss: 0.0152 - accuracy: 0.9892 - auc: 0.9982 - recall: 0.9886 - precision: 0.9897 - val_loss: 0.6936 - val_accuracy: 0.7139 - val_auc: 0.8512 - val_recall: 0.7109 - val_precision: 0.7158 - 2s/epoch - 3ms/step
Epoch 98/100


```
920/920 - 2s - loss: 0.0097 - accuracy: 0.9930 - auc: 0.9986 - recall: 0.9928 - precision: 0.9934 - val_loss: 0.6512 - val_accuracy: 0.7091 - val_auc: 0.8554 - val_recall: 0.7074 - val_precision: 0.7123 - 2s/epoch - 3ms/step
Epoch 99/100
920/920 - 2s - loss: 0.0120 - accuracy: 0.9902 - auc: 0.9983 - recall: 0.9900 - precision: 0.9906 - val_loss: 0.6530 - val_accuracy: 0.7117 - val_auc: 0.8564 - val_recall: 0.7109 - val_precision: 0.7127 - 2s/epoch - 3ms/step
Epoch 100/100
920/920 - 2s - loss: 0.0072 - accuracy: 0.9941 - auc: 0.9990 - recall: 0.9940 - precision: 0.9945 - val_loss: 0.8528 - val_accuracy: 0.7043 - val_auc: 0.8460 - val_recall: 0.7030 - val_precision: 0.7052 - 2s/epoch - 3ms/step
```

```
In [249... ypred=model.predict((X_test[:,::4]-X_test.mean())/X_test.std())
ypred.shape
```

```
Out[249... (2300, 5)
```

```
In [250... yp=np.zeros((Y_test.shape[0]))
yo=np.ones((Y_test.shape[0]))
```

```
In [251... for i in range(Y_test.shape[0]):
    yp[i]=np.argmax(ypred[i])+1
    yo[i]=np.argmax(Y_test[i])
```

```
In [252... yp.shape
```

```
Out[252... (2300,)
```

```
In [253... np.unique(yo)
```

```
Out[253... array([1., 2., 3., 4., 5.])
```

```
In [254... np.unique(Y_test)
```

```
Out[254... array([0., 1.], dtype=float32)
```

```
In [255... #conversion of classes
for i in range(Y_test.shape[0]):
    if yo[i]!=1:
        yo[i]=0
    if yp[i]!=1:
        yp[i]=0
```

```
In [256... np.unique(yo)
```

```
Out[256... array([0., 1.])
```

```
In [257... from sklearn.metrics import accuracy_score
```

```
In [258... accuracy_score(yo,yp)
```

```
Out[258... 0.9765217391304348
```

```
In [274... from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score
```

```

from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
#from sklearn.metrics import sensitivity
#from sklearn.metrics import specificity
accuracy = accuracy_score(yo,yp)
print("Accuracy: %f" % accuracy)
precision = precision_score(yo,yp)
print("Precision: %f" % precision)
recall = recall_score(yo,yp)
print("Recall: %f" % recall)
f1 = f1_score(yo,yp)
print("F1_score: %f" % f1)
#sensitivity = sensitivity(yo,yp)
#print("F1_score: %f" % f1)
#specificity = specificity(yo,yp)
#print("F1_score: %f" % f1)

```

Accuracy: 0.976522
Precision: 0.962963
Recall: 0.916300
F1_score: 0.939052

In [260...

```

cm=confusion_matrix(yo,yp)
cm

```

Out[260...

```

array([[1830, 16],
       [ 38, 416]], dtype=int64)

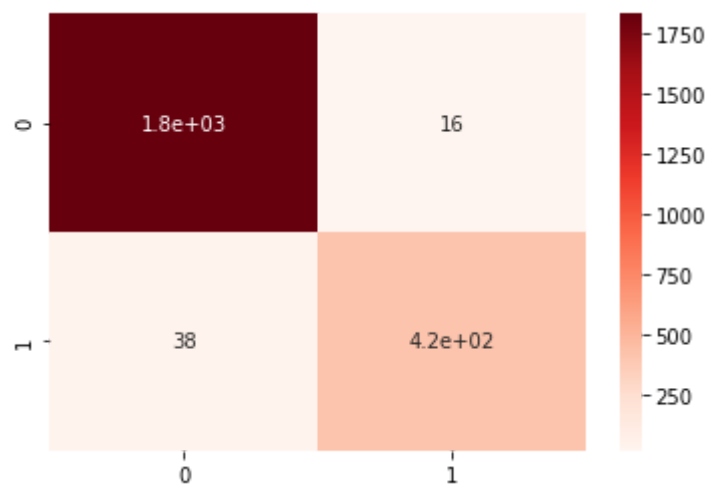
```

In [261...

```

import seaborn as sns
#sns.heatmap(cm)
sns.heatmap(cm,cmap="Reds",annot=True)
plt.show()

```

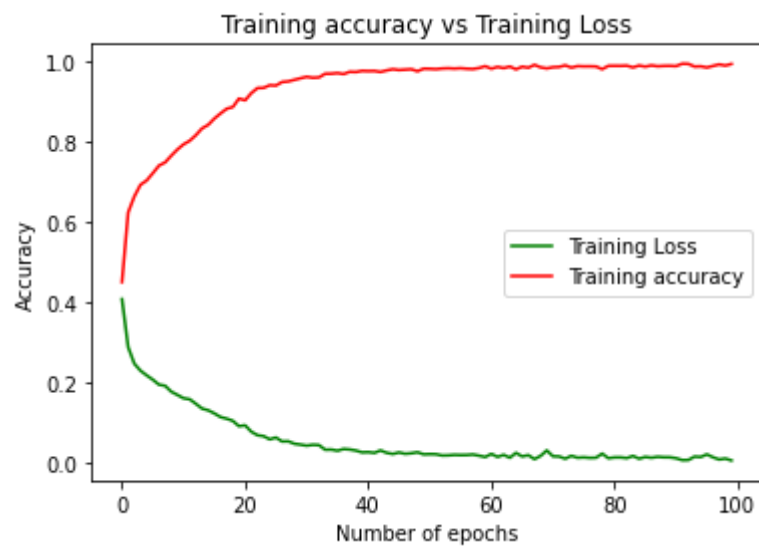


In [262...

```

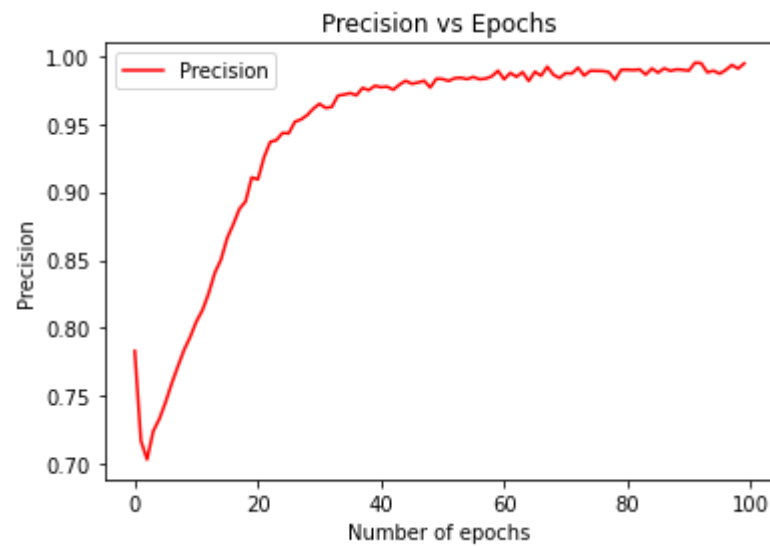
plt.figure(0)
plt.plot(fits.history['loss'],'g',label='Training Loss')
plt.plot(fits.history['accuracy'],'r',label='Training accuracy')
plt.xlabel("Number of epochs")
plt.ylabel("Accuracy")
plt.title("Training accuracy vs Training Loss")
plt.legend()
plt.show()

```



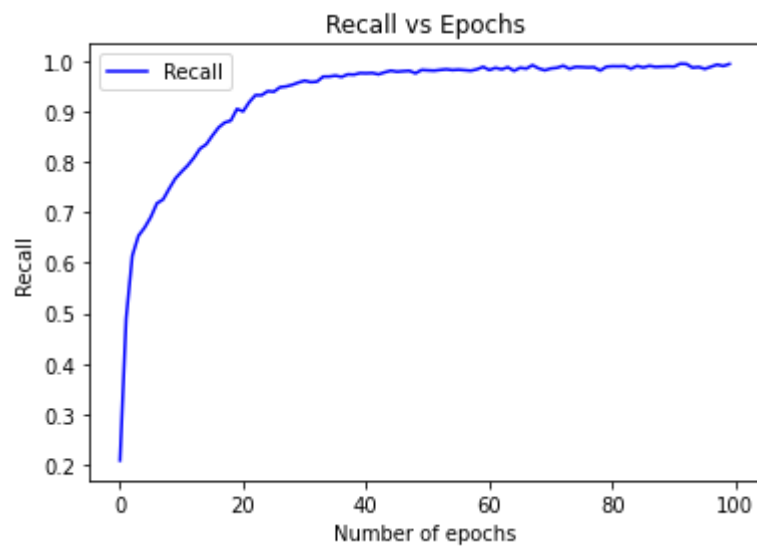
In [263...

```
#plt.figure(0)
plt.plot(fits.history['precision'], 'r', label='Precision')
plt.xlabel("Number of epochs")
plt.ylabel("Precision")
plt.title("Precision vs Epochs")
plt.legend()
plt.show()
```



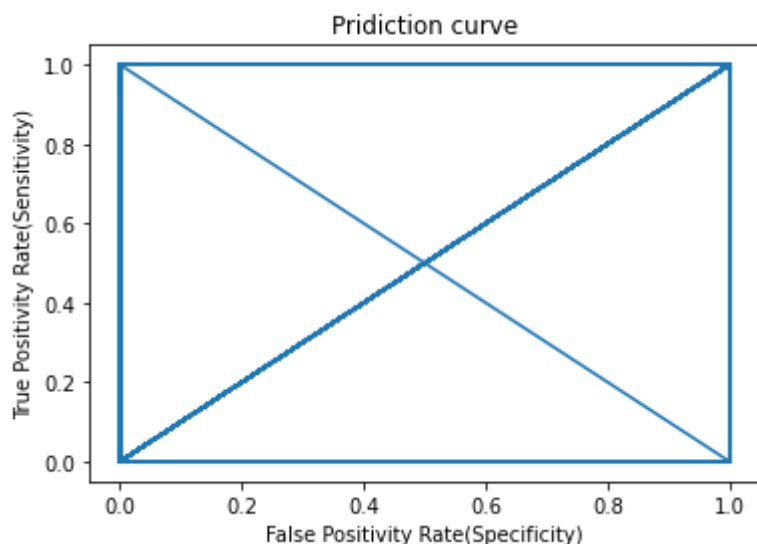
In [264...

```
plt.plot(fits.history['recall'], 'b', label='Recall')
plt.xlabel("Number of epochs")
plt.ylabel("Recall")
plt.title("Recall vs Epochs")
plt.legend()
plt.show()
```



In [178...

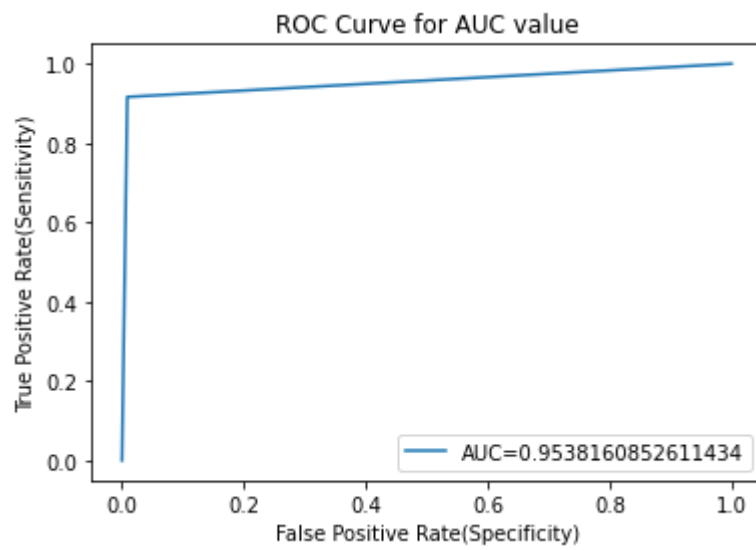
```
#fpr, tpr, _ = metrics.roc_curve(yo, yp)
plt.plot(yo, yp)
plt.xlabel('False Positivity Rate(Specificity)')
plt.ylabel('True Positivity Rate(Sensitivity)')
plt.title("Pridiction curve")
plt.show()
```



In [265...

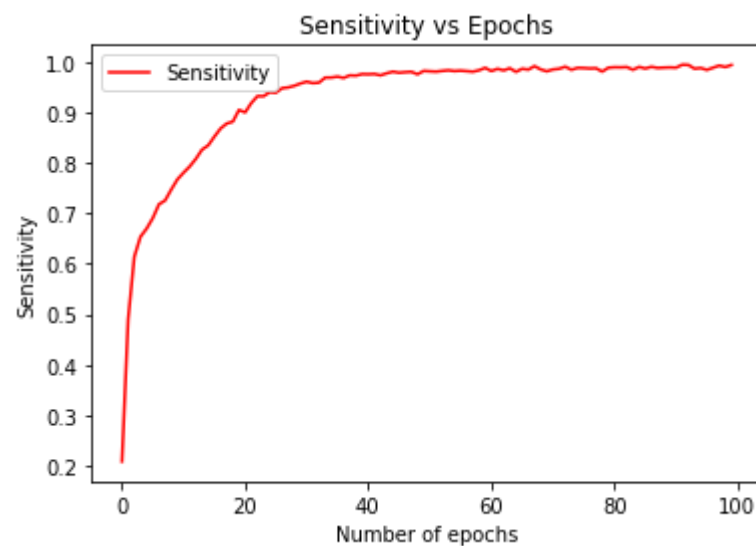
```
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score
import sklearn.metrics as metrics
fpr, tpr, thresholds = roc_curve(yo, yp)
fpr, tpr, _ = metrics.roc_curve(yo, yp)
auc = metrics.roc_auc_score(yo, yp)

#create ROC curve
plt.plot(fpr, tpr, label="AUC="+str(auc))
plt.ylabel('True Positive Rate(Sensitivity)')
plt.xlabel('False Positive Rate(Specificity)')
plt.legend(loc=4)
plt.title("ROC Curve for AUC value")
plt.show()
```



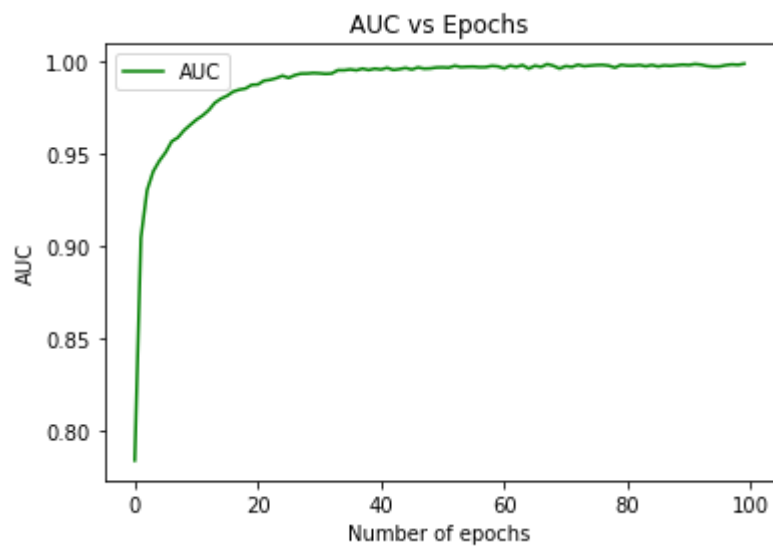
In [266...

```
plt.plot(fits.history['recall'],'r',label='Sensitivity')
plt.xlabel("Number of epochs")
plt.ylabel("Sensitivity")
plt.title("Sensitivity vs Epochs")
plt.legend()
plt.show()
```



In [271...

```
plt.plot(fits.history['auc'],'g',label='AUC')
plt.xlabel("Number of epochs")
plt.ylabel("AUC")
plt.title("AUC vs Epochs")
plt.legend()
plt.show()
```



In []: