Assignment 1 - SML

- Mayank Chauhan, MT18008

Naive Bayes Classifier for 2 Classes

Confusion Matrix, Precision and Recall

```
[14] # In[116]:


     predicted_ls = NaiveBayes(test_images)
     print(cm_2)
     print('Precision: ',cm_2[0,0]/(cm_2[0,0]+cm_2[0,1]) )
     print('Recall: ', cm_2[0,0]/(cm_2[0,0] + cm_2[1,0]) )
     [[965.  97.]
      [ 35. 903.]]
     Precision:  0.908662900188324
     Recall:  0.965


[15] # In[117]:


     # Accuracy 2 class
     print(sum(np.diag(cm_2))/(sum(sum(cm_2))))
     0.934
```

Confusion Matrix

The rows represent predicted data. Columns represent the actual data. This is same for every Confusion matrix in this report.

Taking trouser as a positive class.

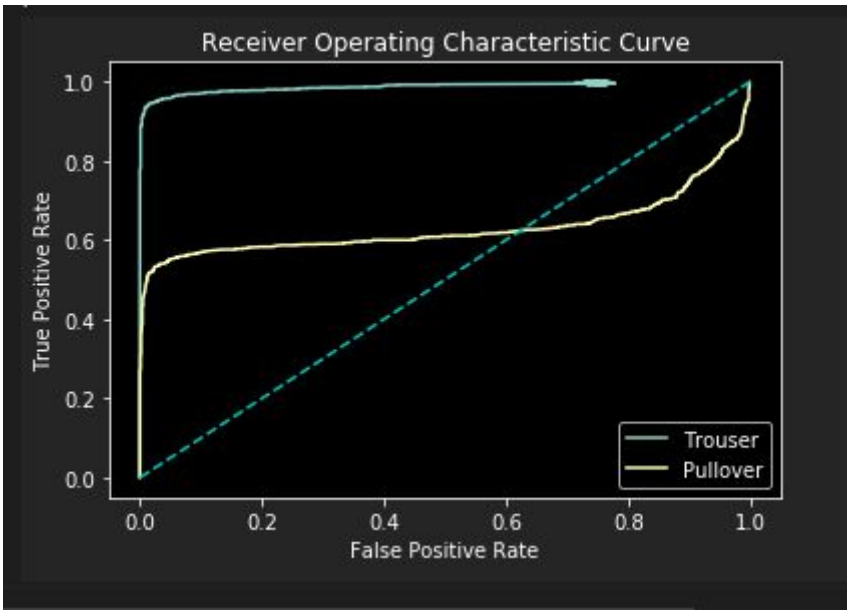
	Actual	
	Trouser	PullOver
Trouser	965	97
PullOver	35	903

Precision : 0.9086

Recall : 0.965

Accuracy : 0.934

ROC Curve for Trouser Vs Pullover



Case 1. Trouser is positive class. In this case, the tradeoff between True positives and false positives is worth it, the area under the curve is almost 1.

Case 2. When, Pullover is positive class. In this case, the tradeoff between True positives and false positives is not so worth it, the increase is false positives rate is much higher than the increase in true positive rate.

Analysis : From the confusion matrix of the Case 1, we can see the no. of false positives are very low making the false positive rate close to zero and the no. of true positives is very high, which makes the true positive rate close to 1, which is verified from
the ROC graph.

Naive Bayes Classifier for 10 Classes

Confusion Matrix:

```
print(cm_10)
[[602.  27.   4.  32.   2.   0. 167.   0.   3.   0.]
 [ 33. 874.   6.  17.   3.   0.   3.   0.   1.   0.]
 [ 26.   3. 279.   1.  60.   0.  77.   0.   9.   0.]
 [ 89.  54.  10. 731.  64.   1.  55.   0.  44.   1.]
 [ 29.  14. 351.  66. 709.   0. 270.   0.  11.   0.]
 [105.  16. 109.  96.  64. 737. 182. 133.  74.  69.]
 [103.  10. 219.  53.  86.   7. 210.   0.  61.  11.]
 [  0.   0.   0.   0.   0. 185.   0. 801.   9.  57.]
 [ 13.   2.  22.   4.  12.   5.  36.   0. 787.   3.]
 [  0.   0.   0.   0.   0.  65.   0.  66.   1. 859.]]
```

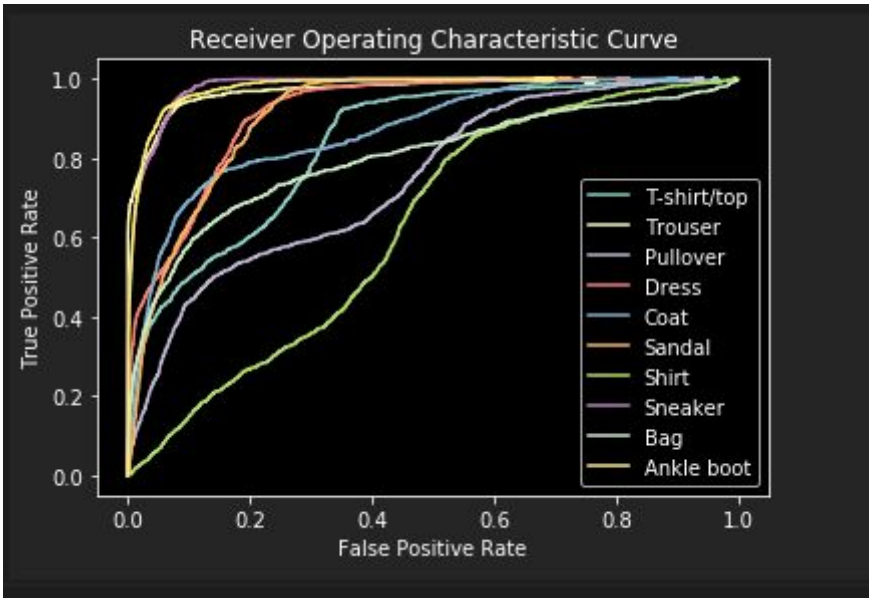
Actual Class

Predicted Class	T-shirt /Top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt /Top	**602**	27	4	32	2	0	167	0	3	0
Trouser	33	**874**	6	17	3	0	3	0	1	0
Pullover	26	3	279	1	60	0	77	0	9	0
Dress	89	54	10	**731**	64	1	55	0	44	1
Coat	29	14	351	66	**709**	0	270	0	11	0
Sandal	105	16	109	96	64	**737**	182	133	74	69
Shirt	103	10	219	53	86	7	210	0	61	11
Sneaker	0	0	0	0	0	185	0	**801**	9	57
Bag	13	2	22	4	12	5	36	0	**787**	3
Ankle boot	0	0	0	0	0	65	0	66	1	**859**

Observations

- Shirt is the least correctly identified, because it's values are spread over other classes.
- Sandal, Sneaker and Ankle boot are able to match with their classes.
  Because they are footwear, so they have almost zero False Positives with the rest of the classes.
- Coat has many False Positives with PullOver and Shirt. It suggests that they are very similar.

ROC Curve of all the 10 Classes



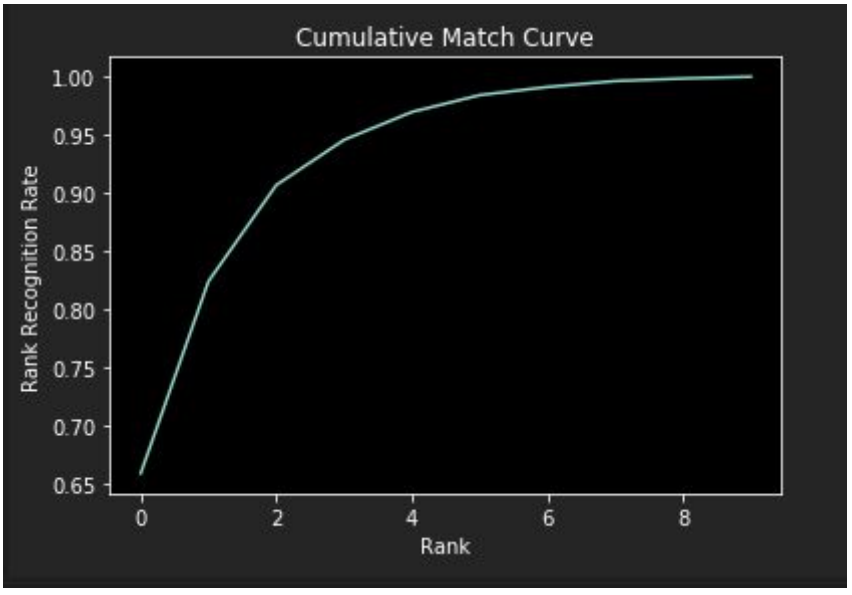
Precision and Recall of each Class (0 - 9)

```
print("Precision of 10 classes: ", precision_i)
print("Recall of 10 classes: ", recall_i)
```

```
Accuracy :  0.6589
Precision of 10 classes:  [0.7192353643966547,
0.9327641408751334, 0.6131868131868132, 0.6968541468064824,
0.4889655172413793, 0.4649842271293375, 0.27631578947368424,
0.7614068441064639, 0.8902714932126696, 0.8668012108980827]
Recall of 10 classes:  [0.602, 0.874, 0.279, 0.731, 0.709,
0.737, 0.21, 0.801, 0.787, 0.859]
```

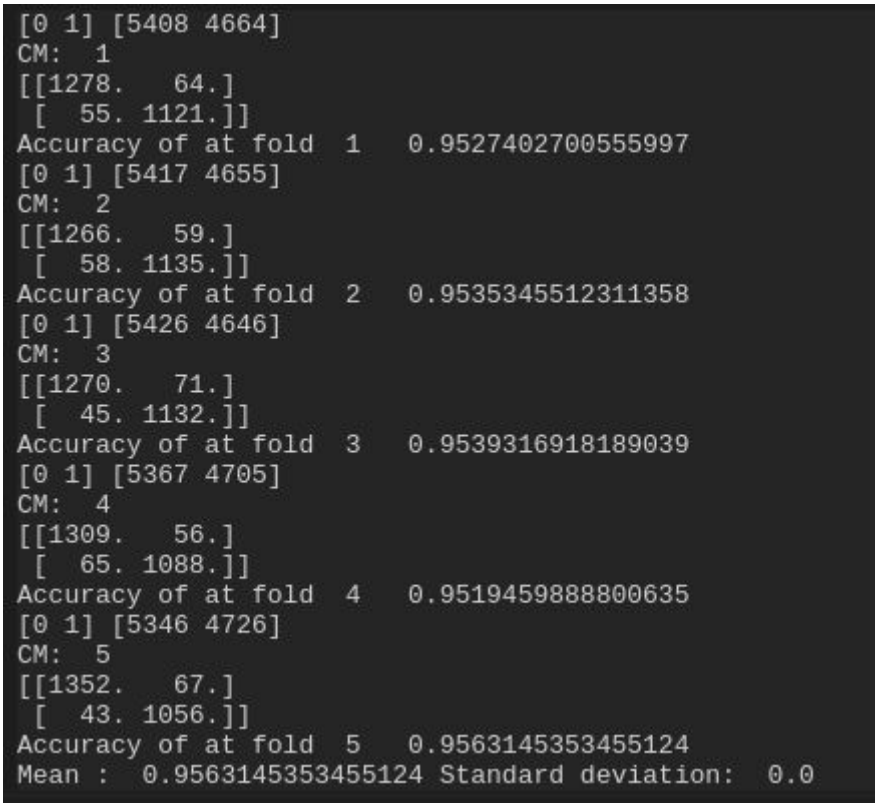
Accuracy : 0.6589

Cumulative Match curve



Random 5 Fold and Stratified Cross Validation (1,8)

Mean, Standard Deviation across folds



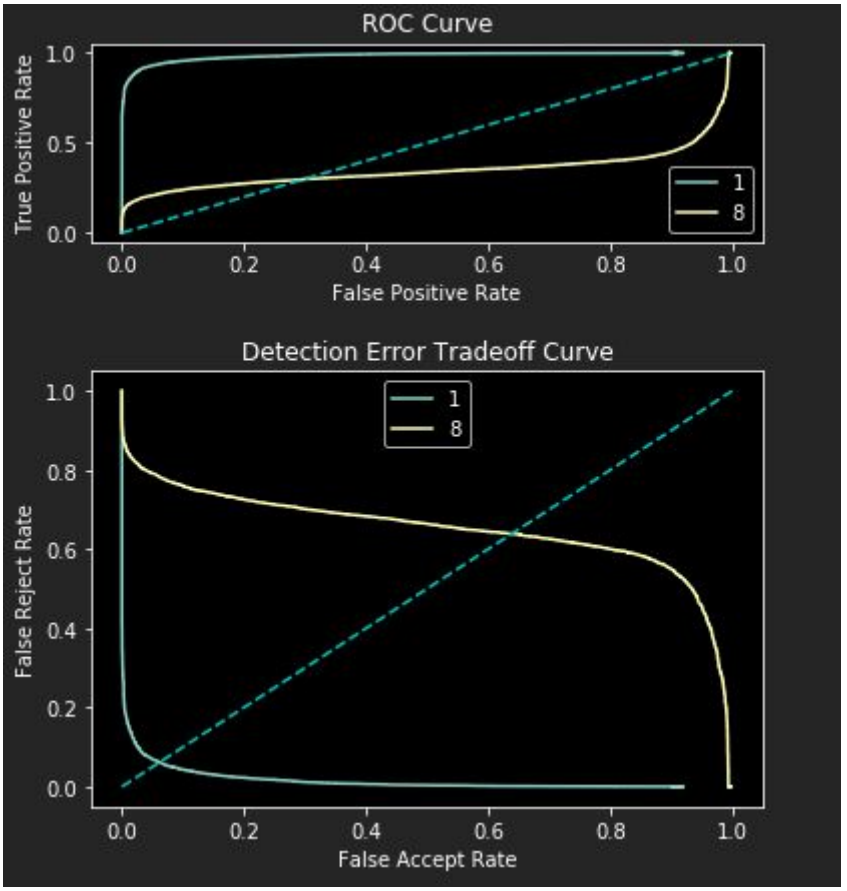
Mean of Accuracy : 0.95

Standard Deviation : 0

Stratification Cross Validation is also giving the same accuracy around 95%

As the data is already balanced, it doesn't improve much.

ROC and DET Curve for (1,8) on Training Set



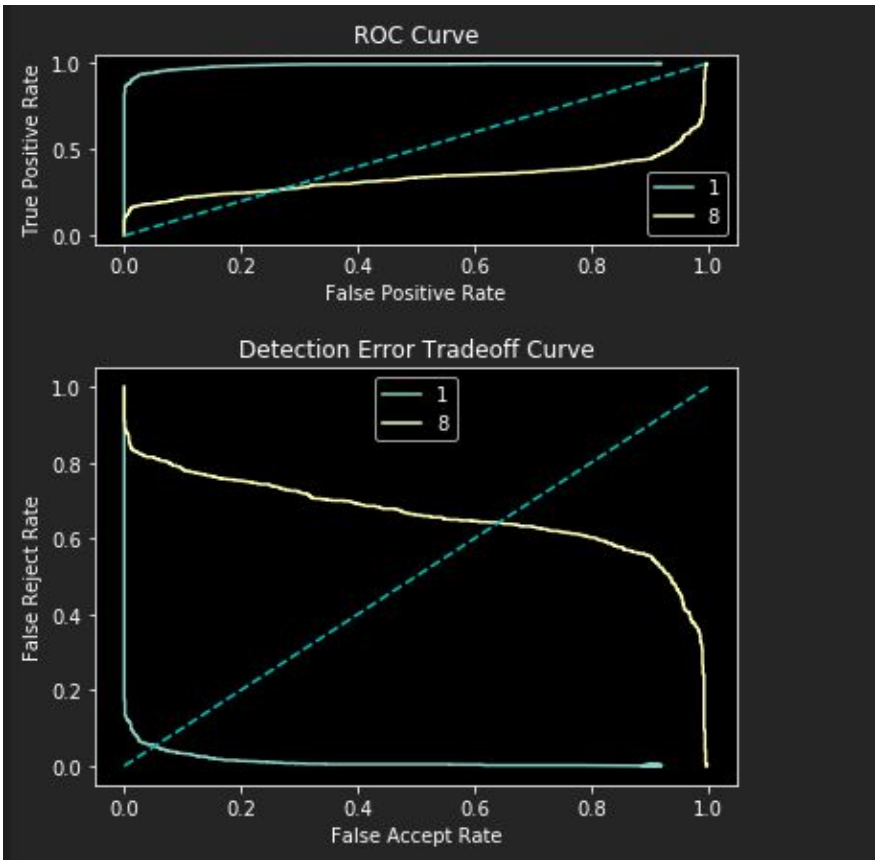
Equal error rate = 0.06 and 0.6

Confusion Matrix for (1,8)

```
Mean:  0.9531374106433678 Standard deviation:  0.0
[[5219.  234.]
 [ 210. 4409.]]
Precision:  0.9570878415551073
Recall:  0.9613188432492171
0.9559173947577443
```

Accuracy: 95%

ROC and DET Curve for (1,8) on Test Set:

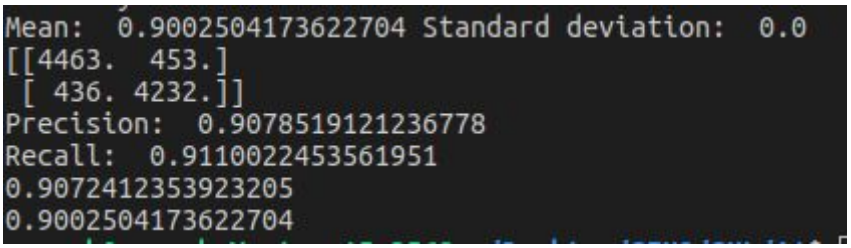


The curve is not smooth as the no. of data points is less now. The equal error rate is For both the test and train set.

ROC and DET Curve (3,8) with 5 Fold , Stratified Cross Validation

Mean and Standard Deviation

Confusion Matrix

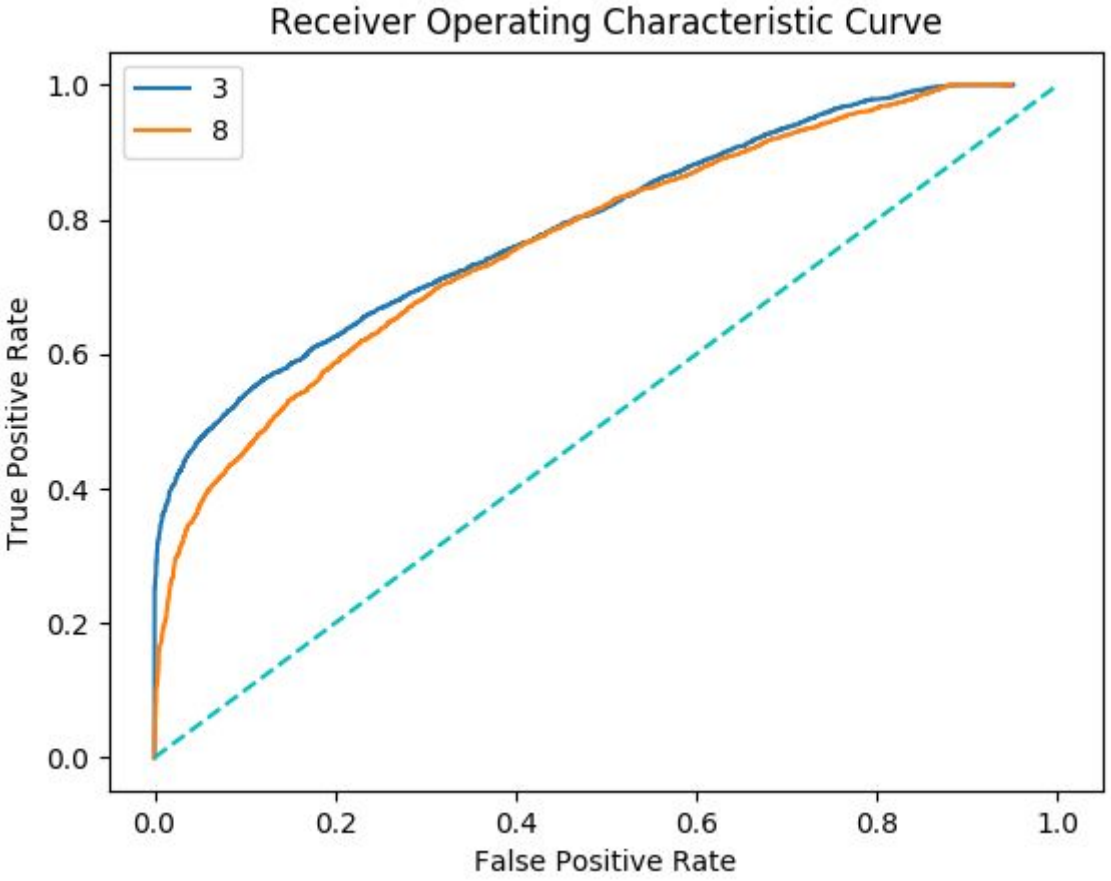


Accuracy : 90%

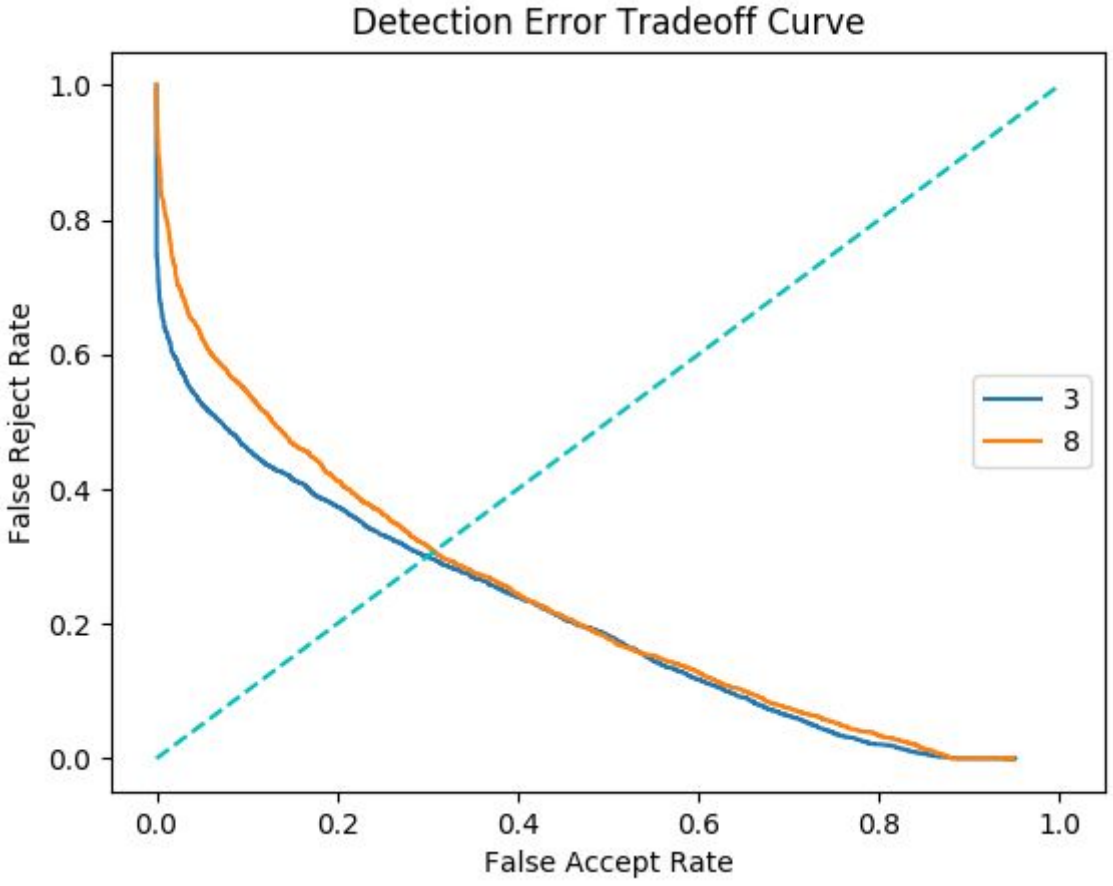
Observation :

The no. of false positives for digit 3, are high comparatively to the confusion matrix of (1,8). Because the pixels of 3 are subpixels in the pixels of 8. Maybe, because of that there are 453 false positives.

ROC (3,8)


Receiver Operating Characteristic Curve
3
8
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

DET Curve (3,8)



Equal Error rate = 0.33

Equal Error Rate = 0.30

Mean, Standard Deviation

```
print(stat.mean(Acc))
```

```
[0 1] [4900 4684]
CM:  1
[[1109.  102.]
 [ 122. 1063.]]
Accuracy of at fold  1   0.9065108514190318
[0 1] [4921 4663]
CM:  2
[[1114.  125.]
 [  96. 1061.]]
Accuracy of at fold  2   0.907762938230384
[0 1] [4894 4690]
CM:  3
[[1134.  115.]
 [ 103. 1044.]]
Accuracy of at fold  3   0.9090150250417363
[0 1] [4903 4681]
CM:  4
[[1132.  115.]
 [  96. 1053.]]
Accuracy of at fold  4   0.9119365609348915
[0 1] [4906 4678]
CM:  5
[[1107.  131.]
 [ 118. 1040.]]
Accuracy of at fold  5   0.8960767946577629
Mean:  0.8960767946577629 Standard deviation:  0.0
[[4464.  448.]
 [ 436. 4236.]]
Precision:  0.9087947882736156
Recall:  0.9110204081632653
0.907762938230384
```

Mean Accuracy : 89.6%

Standard Deviation : 0