CSC311H1 Assignmnet 3

Xiaoyu Zhou

September 10, 2020

1. (a) The dimension of $W^{(1)}$ is $d \times d$
   The dimension of $W^{(2)}$ is $1 \times d$
   The dimension of z_1 is $d \times 1$
   The dimension of z_2 is $d \times 1$

   (b) The number of parameters is $d^2 + d$.

   (c) $\overline{y} = \frac{\partial L}{\partial y} = \frac{\partial}{\partial y}\frac{1}{2}(y-t)^2 = \frac{\partial}{\partial y}\frac{1}{2}(y^2 - 2yt + t^2) = \frac{1}{2}(2y - 2t) = y - t$

   $\overline{W}^{(2)} = \frac{\partial L}{\partial W^2} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial W^2} = \overline{y}\frac{\partial y}{\partial W^2} = \overline{y}z_2 = (y-t)z_2$

   $\overline{z}_2 = \frac{\partial L}{\partial z_2} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_2} = \overline{y}\frac{\partial y}{\partial z_2} = \overline{y}W^{(2)} = (y-t)W^{(2)}$

   $\overline{h} = \frac{\partial L}{\partial h} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_2}\frac{\partial z_2}{\partial h} = \overline{z}_2\frac{\partial z_2}{\partial h} = \overline{z}_2 \cdot 1 = \overline{z}_2 = (y-t)W^{(2)}$

   $\overline{z}_1 = \overline{h}\frac{\partial h}{\partial z_1} = \overline{h} \cdot \frac{e^{-z_1}}{(1+e^{-z_1})^2} = (y-t)W^{(2)} \cdot \frac{e^{-z_1}}{(1+e^{-z_1})^2}$

   $\overline{W}^{(1)} = \frac{\partial L}{\partial W^{(1)}} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial W^1} = \overline{z}_1\frac{\partial z_1}{\partial W^1} = \overline{z}_1 x = (y-t)W^{(2)} \cdot \frac{e^{-z_1}}{(1+e^{-z_1})^2}x$

   $\overline{x} = \frac{\partial L}{\partial x} = \frac{\partial L}{\partial z_1}\frac{\partial z1}{\partial x} + \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial x} = \overline{z}_1\frac{\partial z_1}{\partial x} + \overline{z}_2\frac{\partial z_2}{\partial x} = \overline{z}_1 W^{(1)} + \overline{z}_2 = (y-t)W^{(2)} \cdot \frac{e^{-z_1}}{(1+e^{-z_1})^2}W^{(1)} + (y-t)W^{(2)}$

2. (a) case 1: when $k = k'$

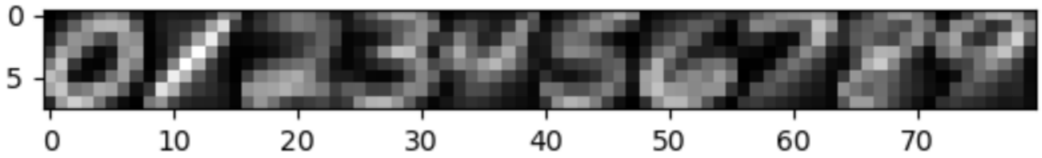
$\frac{\partial y_k}{\partial z_{k'}}$
$= \frac{e^{z_{k'}} \sum_{k'=1}^{K} e^{z_{k'}} - e^{z_{k'}} e^{z_{k'}}}{(\sum_{k'=1}^{K} e^{z_{k'}})^2}$
$= \frac{e^{z_k}}{\sum_{k'=1}^{K} e^{z_{k'}}} \cdot \frac{\sum_{k'=1}^{K} e^{z_{k'}} - e^{z_{k'}}}{\sum_{k'=1}^{K} e^{z_{k'}}}$
$= \frac{e^{z_k}}{\sum_{k'=1}^{K} e^{z_{k'}}} \cdot (1 - \frac{e^{z_k}}{\sum_{k'=1}^{K} e^{z_{k'}}})$
$= y_k(1 - y_k)$
$= y_k - y_k^2$

case 2: when $k \neq k'$

$\frac{\partial y_k}{\partial z_{k'}}$
$= -e^{z_k} (\sum_{k'=1}^{K} e^{z_{k'}})^{-2} (e^{z_{k'}})$
$= -\frac{e^{z_k} e^{z_{k'}}}{(\sum_{k'=1}^{K} e^{z_{k'}})^2}$
$= -y_k y_{k'}$

(b) $\frac{\partial L_{CE}(t, y(x; W))}{\partial W_k}$
$= \frac{\partial L_{CE}}{\partial y_k} \frac{\partial y_k}{\partial z_{k'}} \frac{\partial z_{k'}}{\partial w_k}$
$= \frac{\partial}{\partial y_k}(-t^T \cdot logy) \cdot (y_k - y_k^2) \cdot \frac{\partial}{\partial w_k}(W_{k'}^T \cdot x_k + b)$
$= (\frac{-t_k}{y_k} + \frac{1-t_k}{1-y_k}) \cdot (y_k - y_k^2) \cdot x$
$= \frac{-t_k + t_k y_k + y_k - t_k y_k}{y_k(1-y_k)} \cdot y_k(1 - y_k) \cdot x$
$= \frac{y_k - t_k}{y_k(1-y_k)} \cdot y_k(1 - y_k) \cdot x$
$= (y_k - t_k) \cdot x$

3. **3.0**



3.1.1

```
K == 1
Train accuracy:  1.0
Test accuracy:  0.96875

K == 15
Train accuracy:  0.9637142857142857
Test accuracy:  0.961
```

3.1.2 If there exists any encounter tie, the model will choose the smaller k value.

3.1.3

```
Best K:  4
Cross validation accuracy:  0.9655714285714284
Train accuracy:  0.9864285714285714
Test accuracy:  0.97275
```

3.2.1

The model please see the code of q_3_2_1_MLP.py.

3.2.2

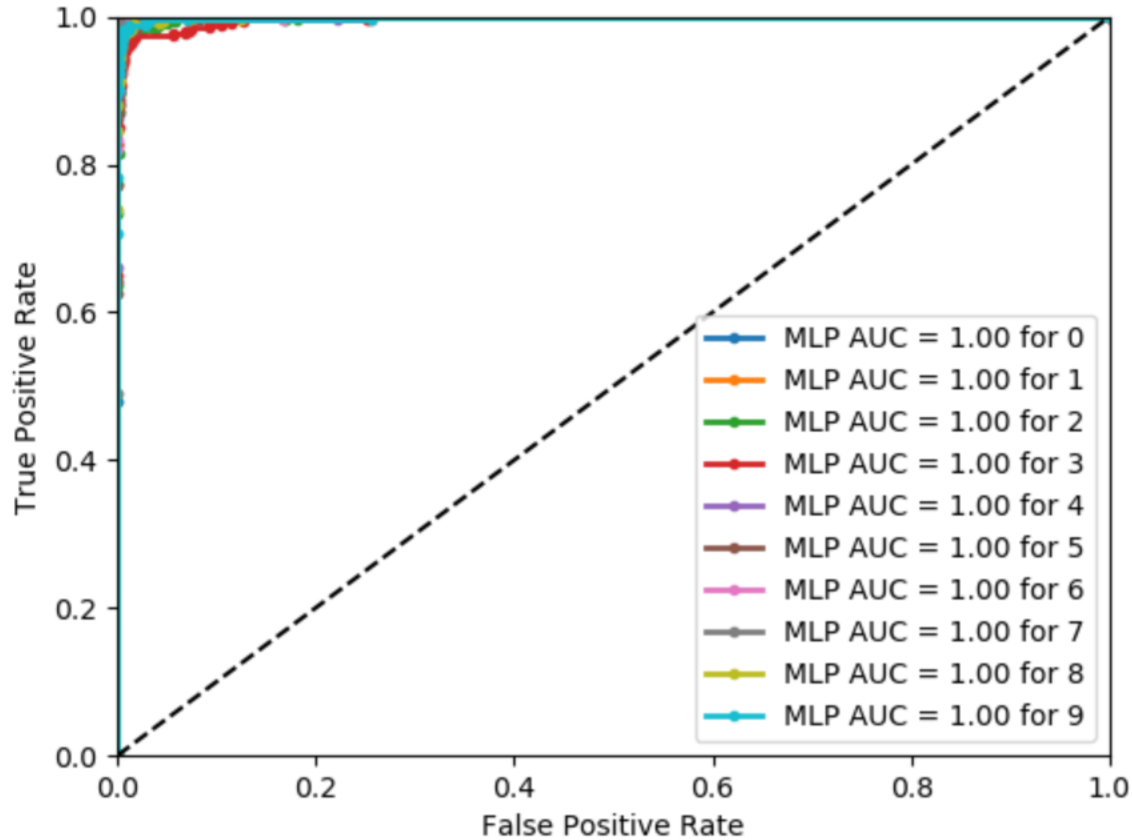
The model please see the code of svm.py.

3.2.3

The model please see the code of ada.py.

3.3

The data for MLP:

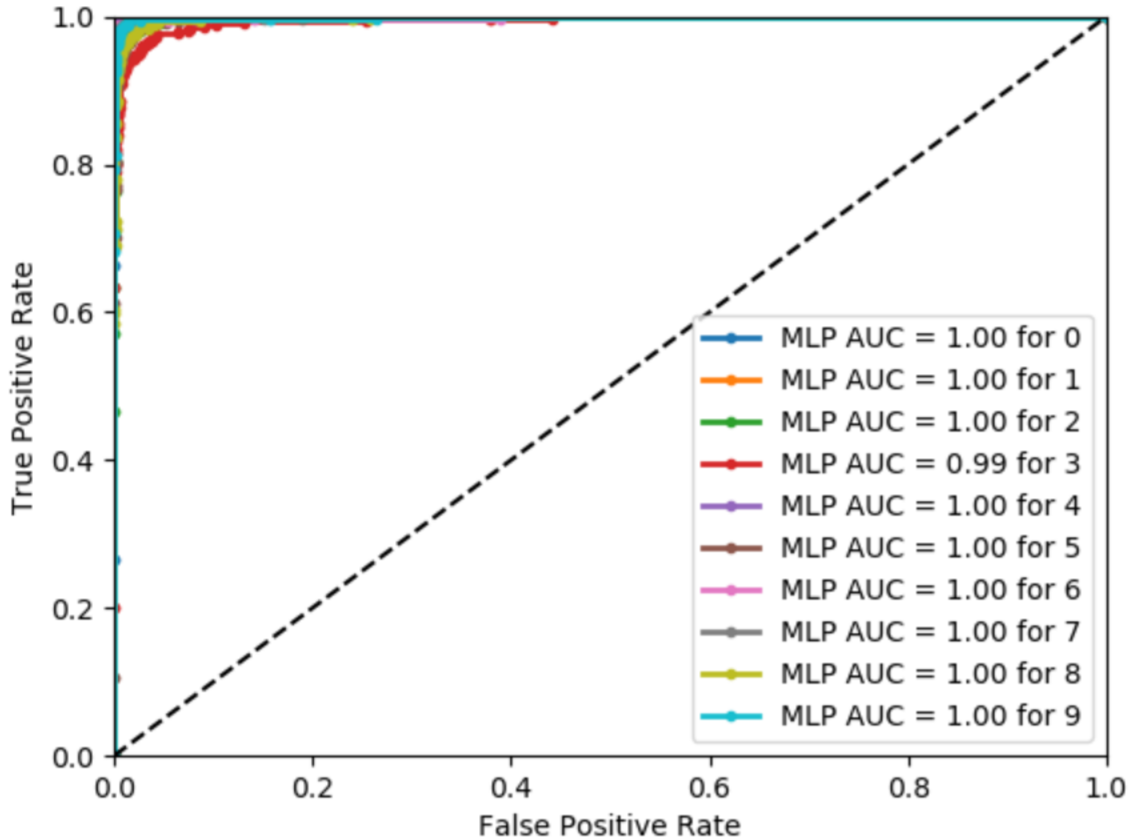


1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
MLP AUC = 1.00 for 0
MLP AUC = 1.00 for 1
MLP AUC = 1.00 for 2
MLP AUC = 1.00 for 3
MLP AUC = 1.00 for 4
MLP AUC = 1.00 for 5
MLP AUC = 1.00 for 6
MLP AUC = 1.00 for 7
MLP AUC = 1.00 for 8
MLP AUC = 1.00 for 9

```
precision for 0:  0.9829418225242457

recall for 0:  0.5802291666666667

confusion matrix for 0 [[3595    5]
 [   2  398]]

precision for 1:  0.9997270003745484

recall for 1:  0.5059790640394088

confusion matrix for 1 [[3595    5]
 [   0  400]]

precision for 2:  0.7409511254016866

recall for 2:  0.8034484389782403

confusion matrix for 2 [[3586   14]
 [  19  381]]

precision for 3:  0.6565035578152791

recall for 3:  0.8386094339622642

confusion matrix for 3 [[3584   16]
 [  25  375]]

precision for 4:  0.6989667217860346

recall for 4:  0.8304987479131889

confusion matrix for 4 [[3594    6]
 [   2  398]]

precision for 5:  0.7623310086295725

recall for 5:  0.7943675000000001

confusion matrix for 5 [[3583   17]
 [  13  387]]
```

```
precision for 6:  0.762336799140295

recall for 6:  0.7978205765407557

confusion matrix for 6 [[3589   11]
 [   9  391]]

precision for 7:  0.9744219102986995

recall for 7:  0.5972858565737051

confusion matrix for 7 [[3593    7]
 [  10  390]]

precision for 8:  0.822019401068258

recall for 8:  0.7591254416961132

confusion matrix for 8 [[3587   13]
 [  12  388]]

precision for 9:  0.66223955540305

recall for 9:  0.844314393939394

confusion matrix for 9 [[3588   12]
 [  14  386]]

accuracy: 0.9735
```

The data for SVM classifier:

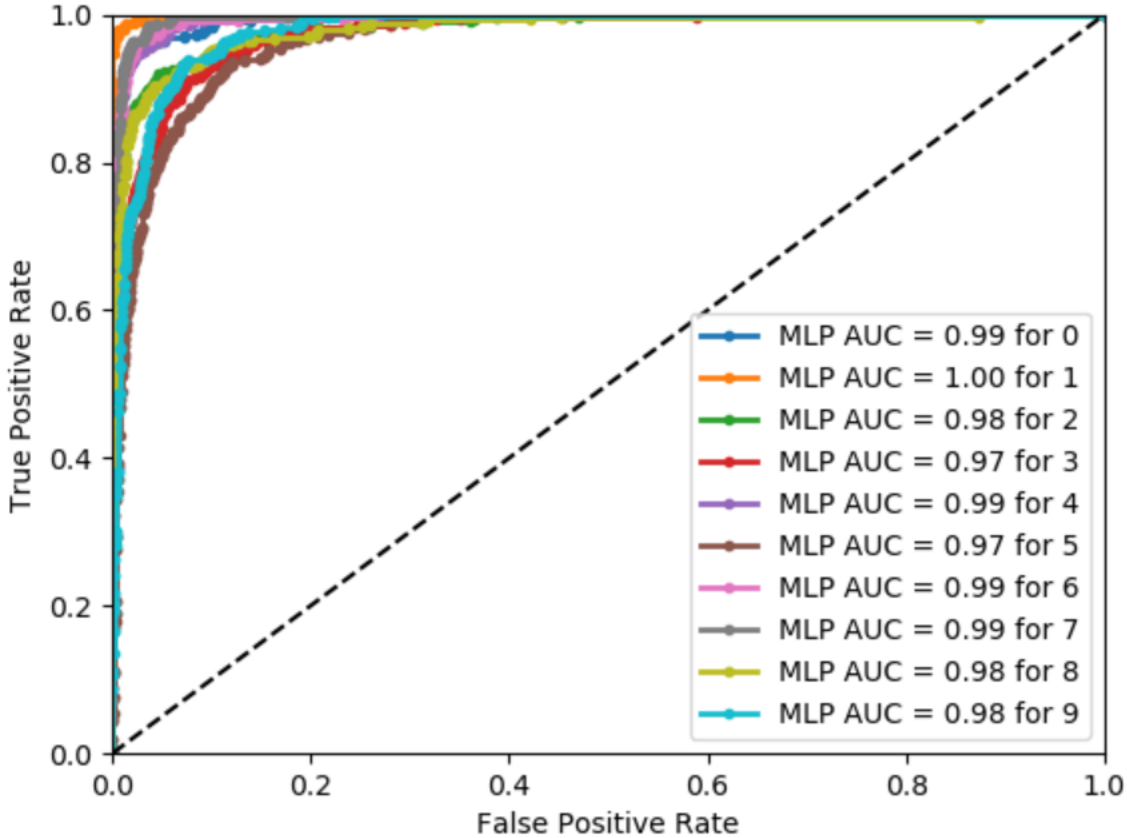


1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
MLP AUC = 1.00 for 0
MLP AUC = 1.00 for 1
MLP AUC = 1.00 for 2
MLP AUC = 0.99 for 3
MLP AUC = 1.00 for 4
MLP AUC = 1.00 for 5
MLP AUC = 1.00 for 6
MLP AUC = 1.00 for 7
MLP AUC = 1.00 for 8
MLP AUC = 1.00 for 9

```
precision for 0:  0.9942292440258093

recall for 0:  0.5219964454976304

confusion matrix for 0 [[3594    6]
 [   4  396]]

precision for 1:  0.9421230392253723

recall for 1:  0.654152397260274

confusion matrix for 1 [[3582   18]
 [   5  395]]

precision for 2:  0.913558395770426

recall for 2:  0.6727139461172742

confusion matrix for 2 [[3578   22]
 [  21  379]]

precision for 3:  0.5153300646638895

recall for 3:  0.8882881526104417

confusion matrix for 3 [[3574   26]
 [  36  364]]

precision for 4:  0.8933315928136542

recall for 4:  0.7081502890173411

confusion matrix for 4 [[3591    9]
 [   7  393]]

precision for 5:  0.7283466356320877

recall for 5:  0.8071382488479264

confusion matrix for 5 [[3578   22]
 [  23  377]]
```

```
precision for 6:  0.5542863807105123

recall for 6:  0.8852913429522754

confusion matrix for 6 [[3591    9]
 [   9  391]]

precision for 7:  0.7853540563238982

recall for 7:  0.7845365466101697

confusion matrix for 7 [[3589   11]
 [   7  393]]

precision for 8:  0.6554884890411131

recall for 8:  0.8412227648384674

confusion matrix for 8 [[3581   19]
 [  25  375]]

precision for 9:  0.6541276293569214

recall for 9:  0.8471015567086733

confusion matrix for 9 [[3589   11]
 [  16  384]]

accuracy: 0.96175
```

The data for AdaBoost classifier:

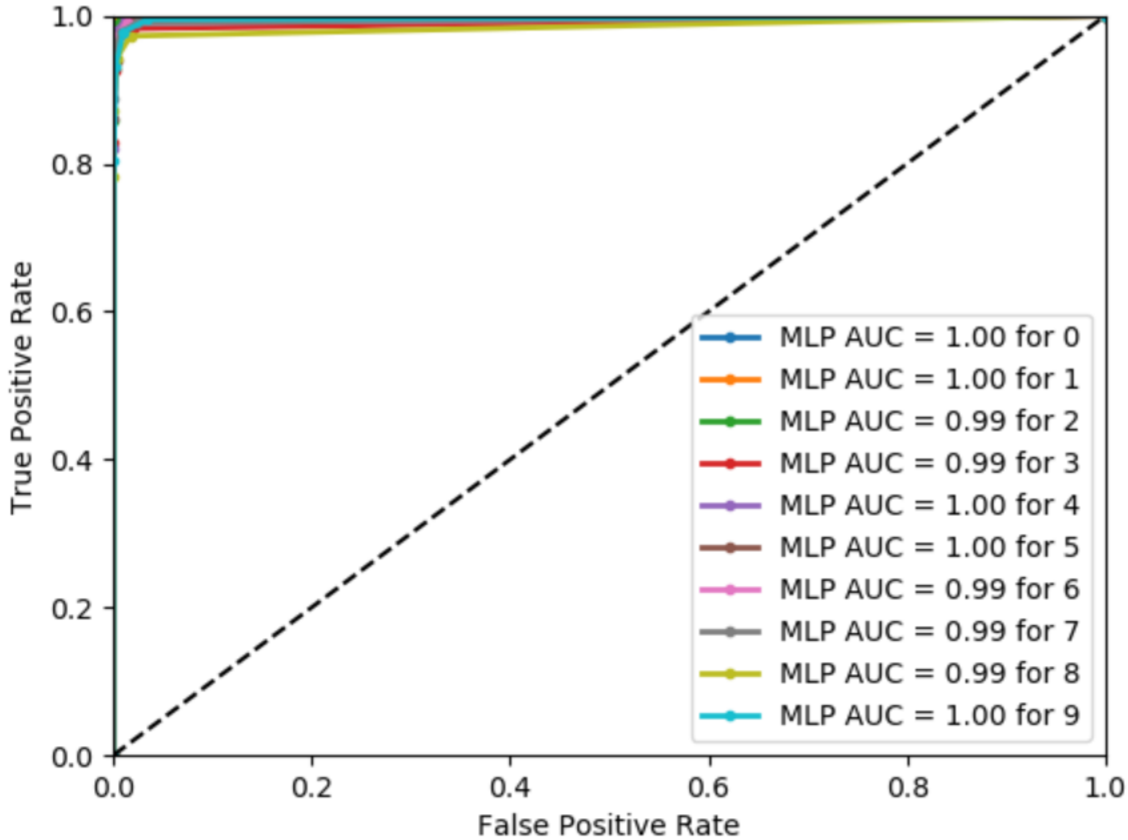

1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
MLP AUC = 0.99 for 0
MLP AUC = 1.00 for 1
MLP AUC = 0.98 for 2
MLP AUC = 0.97 for 3
MLP AUC = 0.99 for 4
MLP AUC = 0.97 for 5
MLP AUC = 0.99 for 6
MLP AUC = 0.99 for 7
MLP AUC = 0.98 for 8
MLP AUC = 0.98 for 9

```
precision for 0:  0.5984058976927893

recall for 0:  0.8550373376623376

confusion matrix for 0 [[3592    8]
 [  39  361]]

precision for 1:  0.8055374682997654

recall for 1:  0.7709652076318745

confusion matrix for 1 [[3597    3]
 [  33  367]]

precision for 2:  0.4872108437203294

recall for 2:  0.8747774054571567

confusion matrix for 2 [[3564   36]
 [  46  354]]
```

```
precision for 3:  0.42038130227357395

recall for 3:  0.8834415971394519

confusion matrix for 3 [[3490  110]
 [  45  355]]

precision for 4:  0.6373867400770861

recall for 4:  0.8368471104608632

confusion matrix for 4 [[3576   24]
 [  29  371]]

precision for 5:  0.4602613397256764

recall for 5:  0.8415915805022157

confusion matrix for 5 [[3501   99]
 [  94  306]]
```

```
precision for 6:  0.6430374115438604

recall for 6:  0.834805122494432

confusion matrix for 6 [[3573   27]
 [  59  341]]

precision for 7:  0.6992370840647648

recall for 7:  0.8050708591674047

confusion matrix for 7 [[3585   15]
 [  80  320]]

precision for 8:  0.3447375315540176

recall for 8:  0.9228230812641085

confusion matrix for 8 [[3517   83]
 [  29  371]]

precision for 9:  0.631158944294017

recall for 9:  0.7705165289256198

confusion matrix for 9 [[3516   84]
 [  35  365]]

accuracy: 0.87775
```

The data for KNN:



1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
MLP AUC = 1.00 for 0
MLP AUC = 1.00 for 1
MLP AUC = 0.99 for 2
MLP AUC = 0.99 for 3
MLP AUC = 1.00 for 4
MLP AUC = 1.00 for 5
MLP AUC = 0.99 for 6
MLP AUC = 0.99 for 7
MLP AUC = 0.99 for 8
MLP AUC = 1.00 for 9

```
precision for 0:  0.9808136021798262

recall for 0:  0.7925000000000001

confusion matrix for 0 [[3592    8]
 [   0  400]]

precision for 1:  0.984091745691322

recall for 1:  0.7462500000000001

confusion matrix for 1 [[3586   14]
 [   0  400]]

precision for 2:  0.8359975671006626

recall for 2:  0.7916666666666666

confusion matrix for 2 [[3595    5]
 [  10  390]]

precision for 3:  0.8144832665864149

recall for 3:  0.7833333333333332

confusion matrix for 3 [[3582   18]
 [  14  386]]

precision for 4:  0.8258124109538288

recall for 4:  0.7891666666666667

confusion matrix for 4 [[3585   15]
 [   9  391]]

precision for 5:  0.8039074839314694

recall for 5:  0.7958333333333334

confusion matrix for 5 [[3577   23]
 [  14  386]]
```

```
precision for 6:  0.824882862196295

recall for 6:  0.8112499999999999

confusion matrix for 6 [[3592    8]
 [  13  387]]

precision for 7:  0.8054294100733729

recall for 7:  0.8016666666666667

confusion matrix for 7 [[3577   23]
 [  11  389]]

precision for 8:  0.8207381654726787

recall for 8:  0.7616666666666667

confusion matrix for 8 [[3595    5]
 [  38  362]]

precision for 9:  0.7898940233139685

recall for 9:  0.785

confusion matrix for 9 [[3585   15]
 [  25  375]]

accuracy:  0.9865714285714285
```

Report

From the ROC curve, we can clearly see that the AdaBoost classifier's area that below the curve is smaller than other classifiers' area. That means that AdaBoost has lower accuracy than other model. Also, the accuracy and most of precision and recall are lower than other model. Thus, AdaBoost performs worst.

KNN model and SVM classfier have large area that under the curve, also, they all have large accuracy. At the same time, their precision and recall are higher than other two models. That means that those two models have lower error rate. So I think KNN model and SVM perform better.

The result kind of satisfied my prediction. In my prediction, KNN will perform best since it uses the a large number of data to analyze and we have already gotten a best k value before we predict.