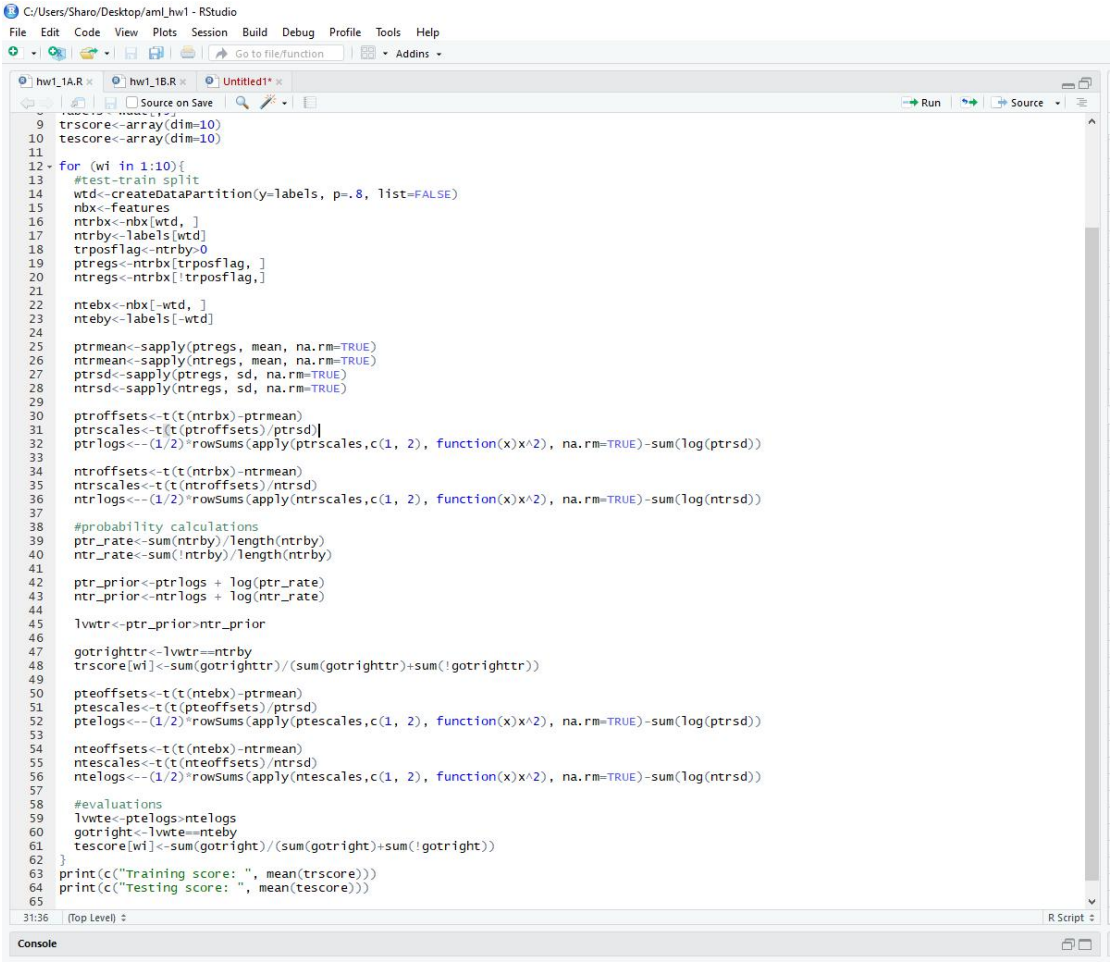
CS498 AML HW1

Huiyun Wu
hwu63

Part1 A
0.736601307189543

Part1 B
0.74640522875817

Part1 D
0.777777777777778


C:/Users/Sharo/Desktop/aml_hw1 - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
hw1_1A.R
hw1_1B.R
Untitled1*
Source on Save
Run
Source
trscore<-array(dim=10)
tescore<-array(dim=10)
for (wi in 1:10){
#test-train split
wtd<-createDataPartition(y=labels, p=.8, list=FALSE)
nbx<-features
ntrbx<-nbx[wtd,]
ntrby<-labels[wtd]
trposflag<-ntrby>0
ptregs<-ntrbx[trposflag,]
ntregs<-ntrbx[!trposflag,]
ntebx<-nbx[-wtd,]
nteby<-labels[-wtd]
ptrmean<-sapply(ptregs, mean, na.rm=TRUE)
ntrmean<-sapply(ntregs, mean, na.rm=TRUE)
ptrsd<-sapply(ptregs, sd, na.rm=TRUE)
ntrsd<-sapply(ntregs, sd, na.rm=TRUE)
ptroffsets<-t(t(ntrbx)-ptrmean)
ptrscales<-t(t(ptroffsets)/ptrsd)
ptrlogs<--(1/2)*rowSums(apply(ptrscales,c(1, 2), function(x)x^2), na.rm=TRUE)-sum(log(ptrsd))
ntroffsets<-t(t(ntrbx)-ntrmean)
ntrscales<-t(t(ntroffsets)/ntrsd)
ntrlogs<--(1/2)*rowSums(apply(ntrscales,c(1, 2), function(x)x^2), na.rm=TRUE)-sum(log(ntrsd))
#probability calculations
ptr_rate<-sum(ntrby)/length(ntrby)
ntr_rate<-sum(!ntrby)/length(ntrby)
ptr_prior<-ptrlogs + log(ptr_rate)
ntr_prior<-ntrlogs + log(ntr_rate)
lvwtr<-ptr_prior>ntr_prior
gotrighttr<-lvwtr==ntrby
trscore[wi]<-sum(gotrighttr)/(sum(gotrighttr)+sum(!gotrighttr))
pteoffsets<-t(t(ntebx)-ptrmean)
ptescales<-t(t(pteoffsets)/ptrsd)
ptelogs<--(1/2)*rowSums(apply(ptescales,c(1, 2), function(x)x^2), na.rm=TRUE)-sum(log(ptrsd))
nteoffsets<-t(t(ntebx)-ntrmean)
ntescales<-t(t(nteoffsets)/ntrsd)
ntelogs<--(1/2)*rowSums(apply(ntescales,c(1, 2), function(x)x^2), na.rm=TRUE)-sum(log(ntrsd))
#evaluations
lvwte<-ptelogs>ntelogs
gotright<-lvwte==nteby
tescore[wi]<-sum(gotright)/(sum(gotright)+sum(!gotright))
}
print(c("Training score: ", mean(trscore)))
print(c("Testing score: ", mean(tescore)))
31:36
(Top Level)
R Script
Console

Accuracy	depth = 4	depth = 8	depth = 16
#trees = 10	0.8481	0.933	0.933
#trees = 30	0.8744	0.9394	0.9649

Untounched (I was not able to write these test result to files so I just showed the confusionMatrix results...)

```
               Accuracy : 0.9624
                 95% CI : (0.9585, 0.966)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9582
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9367
                 95% CI : (0.9317, 0.9414)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9296
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9394
                 95% CI : (0.9345, 0.944)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9326
cnemar's Test P-Value : NA
```

```
               Accuracy : 0.8653
                 95% CI : (0.8585, 0.8719)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8502
 Mcnemar's Test P-Value : < 2.2e-16
```

```
               Accuracy : 0.8744
                 95% CI : (0.8677, 0.8808)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8603
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9649
                 95% CI : (0.9611, 0.9684)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.961
cnemar's Test P-Value : NA
```

Accuracy	depth = 4	depth = 8	depth = 16
#trees = 10	0.8011	0.9059	0.9385
#trees = 30	0.9385	0.9151	0.9484

Bounded

```
               Accuracy : 0.8011
                 95% CI : (0.7931, 0.8089)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7787
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9059
                 95% CI : (0.9, 0.9116)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8954
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9151
                 95% CI : (0.9095, 0.9205)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9056
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9385
                 95% CI : (0.9336, 0.9431)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9316
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9385
                 95% CI : (0.9336, 0.9431)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9316
 Mcnemar's Test P-Value : NA
```

```
               Accuracy : 0.9484
                 95% CI : (0.9439, 0.9527)
    No Information Rate : 0.1135
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9426
 Mcnemar's Test P-Value : < 2.2e-16
```

Accuracy	Gaussian	Bernoulli
Untouched images	0.5352	0.1135
Stretched bounding box	0.8289	0.8145

Gaussian is better in both untouched and stretched bounding box images. Because Bernoulli only recognize 0 or 1 but Gaussian is continuous.

class: 0	class: 1	class: 2	class: 3	class: 4	class: 5	class: 6	class: 7	class: 8	class: 9
0.8796	0.9524	0.2054	0.3069	0.1334	0.03924	0.9290	0.2218	0.6591	0.9524
0.9707	0.9791	0.9982	0.9859	0.9979	0.99704	0.9437	0.9970	0.8124	0.8019
0.7655	0.8539	0.9298	0.7094	0.8733	0.56452	0.6362	0.8941	0.2749	0.3505
0.9867	0.9938	0.9161	0.9268	0.9136	0.91377	0.9921	0.9179	0.9567	0.9934
0.0980	0.1135	0.1032	0.1010	0.0982	0.08920	0.0958	0.1028	0.0974	0.1009
0.0862	0.1081	0.0212	0.0310	0.0131	0.00350	0.0890	0.0228	0.0642	0.0961
0.1126	0.1266	0.0228	0.0437	0.0150	0.00620	0.1399	0.0255	0.2335	0.2742
0.9252	0.9658	0.6018	0.6464	0.5656	0.51814	0.9364	0.6094	0.7358	0.8772

class: 0	class: 1	class: 2	class: 3	class: 4	class: 5	class: 6	class: 7	class: 8	class: 9
0.000	1.0000	0.0000	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.000	0.0000	1.0000	1.000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
NaN	0.1135	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.902	NaN	0.8968	0.899	0.9018	0.9108	0.9042	0.8972	0.9026	0.8991
0.098	0.1135	0.1032	0.101	0.0982	0.0892	0.0958	0.1028	0.0974	0.1009
0.000	0.1135	0.0000	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.000	1.0000	0.0000	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.500	0.5000	0.5000	0.500	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000

class: 0	class: 1	class: 2	class: 3	class: 4	class: 5	class: 6	class: 7	class: 8	class: 9
0.9541	0.7956	0.8091	0.8446	0.8289	0.7881	0.8800	0.8074	0.7659	0.8196
0.9911	0.9877	0.9816	0.9889	0.9799	0.9847	0.9894	0.9777	0.9629	0.9660
0.9212	0.8923	0.8350	0.8951	0.8181	0.8349	0.8978	0.8058	0.6901	0.7299
0.9950	0.9742	0.9781	0.9826	0.9813	0.9794	0.9873	0.9779	0.9744	0.9795
0.0980	0.1135	0.1032	0.1010	0.0982	0.0892	0.0958	0.1028	0.0974	0.1009
0.0935	0.0903	0.0835	0.0853	0.0814	0.0703	0.0843	0.0830	0.0746	0.0827
0.1015	0.1012	0.1000	0.0953	0.0995	0.0842	0.0939	0.1030	0.1081	0.1133
0.9726	0.8916	0.8954	0.9167	0.9044	0.8864	0.9347	0.8926	0.8644	0.8928

class: 0	class: 1	class: 2	class: 3	class: 4	class: 5	class: 6	class: 7	class: 8	class: 9
0.9571	0.7463	0.7771	0.8347	0.8096	0.7814	0.8820	0.8054	0.7567	0.8057
0.9899	0.9873	0.9809	0.9878	0.9792	0.9852	0.9885	0.9737	0.9595	0.9621
0.9116	0.8823	0.8243	0.8846	0.8087	0.8377	0.8904	0.7782	0.6682	0.7045
0.9953	0.9681	0.9745	0.9815	0.9793	0.9787	0.9875	0.9776	0.9734	0.9778
0.0980	0.1135	0.1032	0.1010	0.0982	0.0892	0.0958	0.1028	0.0974	0.1009
0.0938	0.0847	0.0802	0.0843	0.0795	0.0697	0.0845	0.0828	0.0737	0.0813
0.1029	0.0960	0.0973	0.0953	0.0983	0.0832	0.0949	0.1064	0.1103	0.1154
0.9735	0.8668	0.8790	0.9112	0.8944	0.8833	0.9353	0.8896	0.8581	0.8839

Library:

```
library(readr)
source('Reader.R')
library(naivebayes)
library(caret)
```

```
source('Reader.R')

library(caret)
library(h2o)
library(klaR)
library(caret)
```

Evaluations:

```
#result
predictions<-as.data.frame(h2o.predict(rfut_4_10,h2o_df_te_ut))
confusionMatrix(predictions[,1], te_labels)
predictions<-as.data.frame(h2o.predict(rfut_4_30,h2o_df_te_ut))
confusionMatrix(predictions[,1], te_labels)

predictions<-as.data.frame(h2o.predict(rfut_8_10,h2o_df_te_ut))
confusionMatrix(predictions[,1], te_labels)
predictions<-as.data.frame(h2o.predict(rfut_8_30,h2o_df_te_ut))
confusionMatrix(predictions[,1], te_labels)

predictions<-as.data.frame(h2o.predict(rfut_16_10,h2o_df_te_ut))
confusionMatrix(predictions[,1], te_labels)
predictions<-as.data.frame(h2o.predict(rfut_16_30,h2o_df_te_ut))
confusionMatrix(predictions[,1], te_labels)

predictions<-as.data.frame(h2o.predict(rfBounded_4_10,h2o_df_te_bounded))
confusionMatrix(predictions[,1], te_labels)
predictions<-as.data.frame(h2o.predict(rfBounded_8_10,h2o_df_te_bounded))
confusionMatrix(predictions[,1], te_labels)
predictions<-as.data.frame(h2o.predict(rfBounded_16_10,h2o_df_te_bounded))
confusionMatrix(predictions[,1], te_labels)

predictions<-as.data.frame(h2o.predict(rfBounded_4_30,h2o_df_te_bounded))
confusionMatrix(predictions[,1], te_labels)
predictions<-as.data.frame(h2o.predict(rfBounded_8_30,h2o_df_te_bounded))
confusionMatrix(predictions[,1], te_labels)
predictions<-as.data.frame(h2o.predict(rfBounded_16_30,h2o_df_te_bounded))
confusionMatrix(predictions[,1], te_labels)
```