Project: Dogs, Fried Chicken or Blueberry Muffins?

Team #4

Summary:

In this project, we created a classifier for images of puppies, fried chickens and blueberry muffins.

Install Packages

```
# packages.used=c("gbm", "caret","DMwR" ,"nnet","randomForest","EBImage","e1071","xgboost")
#
# # check packages that need to be installed.
# packages.needed=setdiff(packages.used,
#                          intersect(installed.packages()[,1],
#                                    packages.used))
# # install additional packages
# if(length(packages.needed)>0){
#   install.packages(packages.needed, dependencies = TRUE)
# }
```

Read in SIFT feature data

```
sift_train0 <- read.csv("../data/sift_train.csv", header=F)
label_train0 <- read.csv("../data/label_train.csv", header=F)
source("../lib/eco2121_train_gbm_baseline.r")
source("../lib/pca_features.r")
#source("../lib/new_xgboost_sift_pca100.r")

sift <- sift_train0[, -1]
```

Use PCA to reduce dimension

```
set.seed(500)
# data <- pca_features(sift, 100)
#
# # selected data with labels
# pca_train_data <- cbind(data, label_train0[,2])
# colnames(pca_train_data)[ncol(pca_train_data)] <- "label"
# pca_train_data<-as.data.frame(pca_train_data)

sift_pca<-read.csv("../data/feature_pca100.csv",header = T, as.is = T)
label<-read.csv("../data/label_train.csv",header = T,as.is = T)
dat<-cbind(label[,2],sift_pca[,-1])
colnames(dat)[1]<-"label"
```

Train and Validate set

```
set.seed(500)
# Train and test split
train_index<-sample(1:nrow(dat),0.7*nrow(dat))

xgb_variables<-as.matrix(dat[,-1]) # Full dataset
xgb_label<-dat[,1] # Full label

# Split train data
xgb_train<-xgb_variables[train_index,]
train_label<-xgb_label[train_index]
train_matrix<-xgb.DMatrix(data = xgb_train, label=train_label)

# Split test data
xgb_test<-xgb_variables[-train_index,]
test_label<-xgb_label[-train_index]
test_matrix<-xgb.DMatrix(data = xgb_test, label=test_label)
```

Baseline Model: GBM + SIFT

```
sift_train = read.csv("../data/sift_train.csv")
label = read.csv("../data/label_train.csv")
data = data.frame(label[,2], sift_train[,2:ncol(sift_train)])
colnames(data)[1] = "label"

set.seed(123)
index = sample(1:nrow(data), size=0.7*nrow(data))
train_data = data[index,]
test_data = data[-index,]

## To run the baseline model uncomment the following ##

# dat_train = training features
# label_train = labels
# K = number of folds
# d = a certain interaction depth
# system.time(result<-gbm_train(train_data[,2:ncol(train_data)],train_data$label))
# result

#
```

Our Model: XGBoost + PCA + SIFT

```
# Basic model
basic = xgboost(data = train_matrix,
                max.depth=3,eta=0.01,nthread=2,nround=50,
                objective = "multi:softprob",
                eval_metric = "mlogloss",
                num_class = 3,
                verbose = F)
```

```r
# Tune the model
xgb_params_3 = list(objective="multi:softprob",
                    eta = 0.01,
                    max.depth = 3,
                    eval_metric = "mlogloss",
                    num_class = 3)

# fit the model with arbitrary parameters
xgb_3 = xgboost(data = train_matrix,
                params = xgb_params_3,
                nrounds = 100,
                verbose = F)

# cross validation
xgb_cv_3 = xgb.cv(params = xgb_params_3,
                  data = train_matrix,
                  nrounds = 100,
                  nfold = 5,
                  showsd = T,
                  stratified = T,
                  verbose = F,
                  prediction = T)

# set up the cross validated hyper-parameter search
xgb_grid_3 = expand.grid(nrounds=c(100,250,500),
                         eta = c(1,0.1,0.01),
                         max_depth = c(2,4,6,8,10),
                         gamma=1,
                         colsample_bytree=0.5,
                         min_child_weight=2,
                         subsample = 1)

# pack the training control parameters
xgb_trcontrol_3 = trainControl(method = "cv",
                               number = 5,
                               verboseIter = T,
                               returnData = F,
                               returnResamp = "all",
                               allowParallel = T)

# train the model for each parameter combination in the grid

ptm <- proc.time() ## start the time
```

```r
xgb_train_3 = train(x=train_matrix, y=train_label,
                    trControl = xgb_trcontrol_3,
                    tuneGrid = xgb_grid_3,
                    method = "xgbTree")
```

```r
ptm2 <- proc.time()
ptm2- ptm ## stop the clock1
```

```
##    user  system elapsed
## 363.781   1.093 185.344
```

```
# ## Time for training: 350.92s
#
head(xgb_train_3$results[with(xgb_train_3$results,order(RMSE)),],5)
```

```
##     eta max_depth gamma colsample_bytree min_child_weight subsample
## 19 0.10         4     1              0.5                2         1
## 20 0.10         4     1              0.5                2         1
## 21 0.10         4     1              0.5                2         1
## 9  0.01         6     1              0.5                2         1
## 6  0.01         4     1              0.5                2         1
##    nrounds      RMSE  Rsquared       MAE      RMSESD RsquaredSD
## 19     100 0.5281338 0.5869854 0.4298434 0.009097552 0.01756472
## 20     250 0.5281342 0.5869853 0.4298453 0.009097221 0.01756473
## 21     500 0.5281342 0.5869853 0.4298453 0.009097218 0.01756473
## 9      500 0.5307063 0.5890263 0.4331808 0.012147287 0.02247521
## 6      500 0.5317416 0.5863206 0.4405082 0.009503730 0.01996117
##          MAESD
## 19 0.004850170
## 20 0.004850061
## 21 0.004850061
## 9  0.008111319
## 6  0.005830697
```

```
# get the best model's parameters
xgb_train_3$bestTune
```

```
##    nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 19     100         4 0.1     1              0.5                2         1
```

```
# # best model
bst = xgboost(data=train_matrix,max.depth=4,eta=0.1,nthread=2,nround=250,colsample_bytree=0.5,min_child
```

```
## [1]  train-mlogloss:1.046641
## [2]  train-mlogloss:1.001864
## [3]  train-mlogloss:0.961761
## [4]  train-mlogloss:0.926534
## [5]  train-mlogloss:0.893585
## [6]  train-mlogloss:0.863442
## [7]  train-mlogloss:0.835136
## [8]  train-mlogloss:0.810492
## [9]  train-mlogloss:0.784920
## [10] train-mlogloss:0.762907
## [11] train-mlogloss:0.741376
## [12] train-mlogloss:0.721307
## [13] train-mlogloss:0.703002
## [14] train-mlogloss:0.684435
## [15] train-mlogloss:0.666497
## [16] train-mlogloss:0.651517
## [17] train-mlogloss:0.636595
## [18] train-mlogloss:0.621898
## [19] train-mlogloss:0.608445
## [20] train-mlogloss:0.595521
## [21] train-mlogloss:0.582974
## [22] train-mlogloss:0.570615
## [23] train-mlogloss:0.559091
## [24] train-mlogloss:0.547633
## [25] train-mlogloss:0.536969
## [26] train-mlogloss:0.526722
## [27] train-mlogloss:0.517187
## [28] train-mlogloss:0.507662
## [29] train-mlogloss:0.499092
## [30] train-mlogloss:0.490774
## [31] train-mlogloss:0.482044
## [32] train-mlogloss:0.473782
## [33] train-mlogloss:0.466406
## [34] train-mlogloss:0.458688
## [35] train-mlogloss:0.450924
## [36] train-mlogloss:0.443706
## [37] train-mlogloss:0.436071
## [38] train-mlogloss:0.429466
## [39] train-mlogloss:0.422733
## [40] train-mlogloss:0.415989
## [41] train-mlogloss:0.409102
## [42] train-mlogloss:0.402623
## [43] train-mlogloss:0.396998
## [44] train-mlogloss:0.391213
## [45] train-mlogloss:0.386150
## [46] train-mlogloss:0.380720
## [47] train-mlogloss:0.375326
## [48] train-mlogloss:0.369987
## [49] train-mlogloss:0.364800
## [50] train-mlogloss:0.359715
## [51] train-mlogloss:0.354392
## [52] train-mlogloss:0.350055
## [53] train-mlogloss:0.345898
## [54] train-mlogloss:0.340898
## [55] train-mlogloss:0.336375
## [56] train-mlogloss:0.331864
## [57] train-mlogloss:0.327777
## [58] train-mlogloss:0.323460
## [59] train-mlogloss:0.319854
## [60] train-mlogloss:0.315955
## [61] train-mlogloss:0.311894
## [62] train-mlogloss:0.308605
## [63] train-mlogloss:0.305139
## [64] train-mlogloss:0.301682
## [65] train-mlogloss:0.298304
## [66] train-mlogloss:0.295097
## [67] train-mlogloss:0.291946
## [68] train-mlogloss:0.288274
## [69] train-mlogloss:0.285148
## [70] train-mlogloss:0.282024
## [71] train-mlogloss:0.278933
## [72] train-mlogloss:0.276255
## [73] train-mlogloss:0.273190
## [74] train-mlogloss:0.270343
## [75] train-mlogloss:0.267508
## [76] train-mlogloss:0.264329
## [77] train-mlogloss:0.261411
## [78] train-mlogloss:0.258454
## [79] train-mlogloss:0.255644
## [80] train-mlogloss:0.253290
## [81] train-mlogloss:0.250890
## [82] train-mlogloss:0.248649
## [83] train-mlogloss:0.246158
## [84] train-mlogloss:0.243372
## [85] train-mlogloss:0.240619
## [86] train-mlogloss:0.237953
## [87] train-mlogloss:0.234995
## [88] train-mlogloss:0.231851
## [89] train-mlogloss:0.229330
## [90] train-mlogloss:0.226797
## [91] train-mlogloss:0.224779
## [92] train-mlogloss:0.222427
## [93] train-mlogloss:0.220292
## [94] train-mlogloss:0.218295
## [95] train-mlogloss:0.216358
## [96] train-mlogloss:0.214260
## [97] train-mlogloss:0.212138
## [98] train-mlogloss:0.210369
## [99] train-mlogloss:0.208662
## [100]    train-mlogloss:0.206367
## [101]    train-mlogloss:0.203990
## [102]    train-mlogloss:0.202138
## [103]    train-mlogloss:0.199980
## [104]    train-mlogloss:0.198534
## [105]    train-mlogloss:0.196547
## [106]    train-mlogloss:0.194822
## [107]    train-mlogloss:0.192831
## [108]    train-mlogloss:0.191212
## [109]    train-mlogloss:0.189820
## [110]    train-mlogloss:0.187844
## [111]    train-mlogloss:0.185801
## [112]    train-mlogloss:0.183622
## [113]    train-mlogloss:0.182170
## [114]    train-mlogloss:0.180820
## [115]    train-mlogloss:0.179056
## [116]    train-mlogloss:0.177298
## [117]    train-mlogloss:0.175879
## [118]    train-mlogloss:0.174041
## [119]    train-mlogloss:0.172147
## [120]    train-mlogloss:0.170611
## [121]    train-mlogloss:0.169077
## [122]    train-mlogloss:0.167372
## [123]    train-mlogloss:0.165894
## [124]    train-mlogloss:0.164531
## [125]    train-mlogloss:0.162740
## [126]    train-mlogloss:0.161462
## [127]    train-mlogloss:0.160295
## [128]    train-mlogloss:0.158286
## [129]    train-mlogloss:0.157034
## [130]    train-mlogloss:0.155418
## [131]    train-mlogloss:0.153333
## [132]    train-mlogloss:0.152313
## [133]    train-mlogloss:0.150777
## [134]    train-mlogloss:0.149715
## [135]    train-mlogloss:0.148585
## [136]    train-mlogloss:0.147245
## [137]    train-mlogloss:0.145991
## [138]    train-mlogloss:0.144308
## [139]    train-mlogloss:0.143233
## [140]    train-mlogloss:0.142301
## [141]    train-mlogloss:0.140886
## [142]    train-mlogloss:0.139786
## [143]    train-mlogloss:0.138882
## [144]    train-mlogloss:0.137815
## [145]    train-mlogloss:0.136624
## [146]    train-mlogloss:0.135577
## [147]    train-mlogloss:0.134211
## [148]    train-mlogloss:0.132973
## [149]    train-mlogloss:0.131928
## [150]    train-mlogloss:0.130867
## [151]    train-mlogloss:0.129484
## [152]    train-mlogloss:0.128175
## [153]    train-mlogloss:0.126871
## [154]    train-mlogloss:0.125898
## [155]    train-mlogloss:0.125156
## [156]    train-mlogloss:0.124126
## [157]    train-mlogloss:0.123484
## [158]    train-mlogloss:0.122404
## [159]    train-mlogloss:0.121228
## [160]    train-mlogloss:0.119832
## [161]    train-mlogloss:0.118628
## [162]    train-mlogloss:0.117737
## [163]    train-mlogloss:0.116944
## [164]    train-mlogloss:0.115921
## [165]    train-mlogloss:0.115215
## [166]    train-mlogloss:0.114270
## [167]    train-mlogloss:0.113188
## [168]    train-mlogloss:0.112308
## [169]    train-mlogloss:0.111104
## [170]    train-mlogloss:0.110185
## [171]    train-mlogloss:0.109364
## [172]    train-mlogloss:0.108542
## [173]    train-mlogloss:0.107805
## [174]    train-mlogloss:0.107107
## [175]    train-mlogloss:0.106189
## [176]    train-mlogloss:0.105184
## [177]    train-mlogloss:0.104161
## [178]    train-mlogloss:0.103237
## [179]    train-mlogloss:0.102378
## [180]    train-mlogloss:0.101885
## [181]    train-mlogloss:0.101249
## [182]    train-mlogloss:0.100466
## [183]    train-mlogloss:0.099936
## [184]    train-mlogloss:0.099560
## [185]    train-mlogloss:0.098861
## [186]     train-mlogloss:0.097890
## [187]     train-mlogloss:0.097230
## [188]     train-mlogloss:0.096381
## [189]     train-mlogloss:0.095762
## [190]     train-mlogloss:0.094787
## [191]     train-mlogloss:0.094040
## [192]     train-mlogloss:0.093344
## [193]     train-mlogloss:0.092410
## [194]     train-mlogloss:0.091494
## [195]     train-mlogloss:0.090818
## [196]     train-mlogloss:0.090117
## [197]     train-mlogloss:0.089385
## [198]     train-mlogloss:0.088658
## [199]     train-mlogloss:0.087960
## [200]     train-mlogloss:0.087183
## [201]     train-mlogloss:0.086615
## [202]     train-mlogloss:0.085680
## [203]     train-mlogloss:0.085188
## [204]     train-mlogloss:0.084448
## [205]     train-mlogloss:0.083915
## [206]     train-mlogloss:0.083374
## [207]     train-mlogloss:0.082667
## [208]     train-mlogloss:0.082225
## [209]     train-mlogloss:0.081579
## [210]     train-mlogloss:0.080840
## [211]     train-mlogloss:0.080150
## [212]     train-mlogloss:0.079621
## [213]     train-mlogloss:0.078979
## [214]     train-mlogloss:0.078380
## [215]     train-mlogloss:0.077835
## [216]     train-mlogloss:0.077312
## [217]     train-mlogloss:0.076818
## [218]     train-mlogloss:0.076000
## [219]     train-mlogloss:0.075363
## [220]     train-mlogloss:0.074737
## [221]     train-mlogloss:0.074248
## [222]     train-mlogloss:0.073547
## [223]     train-mlogloss:0.073081
## [224]     train-mlogloss:0.072503
## [225]     train-mlogloss:0.071803
## [226]     train-mlogloss:0.071348
## [227]     train-mlogloss:0.070979
## [228]     train-mlogloss:0.070271
## [229]     train-mlogloss:0.069546
## [230]     train-mlogloss:0.069217
## [231]     train-mlogloss:0.068826
## [232]     train-mlogloss:0.068392
## [233]     train-mlogloss:0.067994
## [234]     train-mlogloss:0.067564
## [235]     train-mlogloss:0.067080
## [236]     train-mlogloss:0.066612
## [237]     train-mlogloss:0.066042
## [238]     train-mlogloss:0.065614
## [239]     train-mlogloss:0.065217
## [240]    train-mlogloss:0.064693
## [241]    train-mlogloss:0.064127
## [242]    train-mlogloss:0.063756
## [243]    train-mlogloss:0.063319
## [244]    train-mlogloss:0.062778
## [245]    train-mlogloss:0.062200
## [246]    train-mlogloss:0.061629
## [247]    train-mlogloss:0.061131
## [248]    train-mlogloss:0.060701
## [249]    train-mlogloss:0.060315
## [250]    train-mlogloss:0.059977
```

```r
pred = predict(bst, test_matrix)
prediction<-matrix(pred,nrow = 3,ncol = length(pred)/3) %>%
  t() %>%
  data.frame() %>%
  mutate(label=test_label+1,max_prob=max.col(.,"last"))

# ## confusion matrix of test set
confusionMatrix(factor(prediction$label),factor(prediction$max_prob),mode = "everything")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 283   7  16
##          2  14 234  60
##          3  16  70 200
##
## Overall Statistics
##
##                Accuracy : 0.7967
##                  95% CI : (0.7689, 0.8225)
##     No Information Rate : 0.3478
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6947
##  Mcnemar's Test P-Value : 0.3761
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9042   0.7524   0.7246
## Specificity            0.9608   0.8744   0.8622
## Pos Pred Value         0.9248   0.7597   0.6993
## Neg Pred Value         0.9495   0.8699   0.8762
## Precision              0.9248   0.7597   0.6993
## Recall                 0.9042   0.7524   0.7246
## F1                     0.9144   0.7561   0.7117
## Prevalence             0.3478   0.3456   0.3067
## Detection Rate         0.3144   0.2600   0.2222
## Detection Prevalence   0.3400   0.3422   0.3178
## Balanced Accuracy      0.9325   0.8134   0.7934
```

```
# ## Accuracy: 82.67%
# ## Parameters: max.depth=4, eta=0.1, nthread=2, nround=250
```