STAT 665 Homework 2

Issac Li

1/30/2017

Part I

Impute the NAs in `capital_run_length_average` with k-nn algorithm.

```
require(FNN)

## Loading required package: FNN

require(data.table)

## Loading required package: data.table

train=fread("/Users/lizhuo/Documents/STAT665/HW2/spam_train.csv",header = T)
test=fread("/Users/lizhuo/Documents/STAT665/HW2/spam_test.csv",header = T)
paste(sum(is.na(train$capital_run_length_average))," missing values in Traini
ng data")

## [1] "661  missing values in Training data"

paste(sum(is.na(test$capital_run_length_average))," missing values in Testing
data")

## [1] "259  missing values in Testing data"
```

The missing values are approximately distributed with similar proportions so we can merge the two datasets and then do imputation to yield better imputation results.

```
# Combining datasets
train$group=1
test$group=2;test$spam=NA
merged=rbind(train,test)
# Rescaling and add group numbers for reverse mapping
merged=data.table(cbind(merged$group,merged$spam,merged$capital_run_length_av
erage,apply(merged[,!c("group","spam","capital_run_length_average"),with=F],2
,function(x) scale(x,center = min(x),scale=max(x)-min(x)))))

colnames(merged)[1:3]<-c("group","spam","capital_run_length_average")

# Prepare dataset for imputation
impute_test=merged[is.na(merged$capital_run_length_average)]
impute_train=merged[!is.na(merged$capital_run_length_average)]

impute_test_x=impute_test[,!c("capital_run_length_average"),with=F]
```

```
impute_train_x=impute_train[,!c("capital_run_length_average"),with=F]
impute_train_y=impute_train$capital_run_length_average

ans=knn.reg(train =impute_train_x[,!c("group","spam"),with=F],test = impute_t
est_x[,!c("group","spam"),with=F], y = impute_train_y,k=15)

impute_test$capital_run_length_average=ans$pred

merged=rbind(impute_test,impute_train)

ctrain=merged[group==1,!c("group"),with=F]
ctest=merged[group==2,!c("group","spam"),with=F]

paste(sum(is.na(ctrain$capital_run_length_average))," missing values in Train
ing data")

## [1] "0  missing values in Training data"

paste(sum(is.na(ctest$capital_run_length_average)),"  missing values in Testin
g data")

## [1] "0  missing values in Testing data"
```

After imputation we can see that there is no more NA's in the capital_run_length_average column in either the training or the test set

Part II

Please see HW2_knnclass.R

Part III

```
# Use KNN to Predict spam without ```capital_run_length_average```
knn_pred1 = knnclass(xtrain = ctrain[,!c("capital_run_length_average","spam")
,with=F],
                     xtest = ctest[,!c("capital_run_length_average"),with=F],
                     ytrain = ctrain$spam)


# Use KNN to Predict spam with ```capital_run_length_average```
knn_pred2 = knnclass(xtrain = ctrain[,!c("spam"),with=F],
                     xtest = ctest,
                     ytrain = ctrain$spam)

# Use logistic regression to predict spam without ```capital_run_length_avera
ge```
log_fit1=glm(data=ctrain,formula = spam ~ .-capital_run_length_average,family
= "binomial")
logm_pred1=predict(log_fit1,newdata = ctest,type = "response")
```

```r
logm_pred1=ifelse(logm_pred1>=0.5,1,0)

# Use logistic regression to predict spam with ```capital_run_length_average`
``
log_fit2=glm(data=ctrain,formula = spam ~ .,family = "binomial")
logm_pred2=predict(log_fit2,newdata = ctest,type = "response")
logm_pred2=ifelse(logm_pred2>=0.5,1,0)

summary.glm(log_fit2)

##
## Call:
## glm(formula = spam ~ ., family = "binomial", data = ctrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0883  -0.2136   0.0000   0.1273   4.6095
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.78266    0.18305  -9.739  < 2e-16 ***
## capital_run_length_average    0.07713    0.04553   1.694 0.090239 .
## word_freq_make               -1.26333    1.12283  -1.125 0.260535
## word_freq_address            -2.16722    1.21519  -1.783 0.074513 .
## word_freq_all                 0.93979    0.69408   1.354 0.175729
## word_freq_3d                 94.73299   81.58671   1.161 0.245588
## word_freq_our                 5.73798    1.25820   4.560 5.10e-06 ***
## word_freq_over                5.59950    1.89702   2.952 0.003160 **
## word_freq_remove             13.09388    2.44335   5.359 8.37e-08 ***
## word_freq_internet            6.73605    2.62947   2.562 0.010415 *
## word_freq_order               2.31141    1.63511   1.414 0.157475
## word_freq_mail                5.03290    1.98138   2.540 0.011082 *
## word_freq_receive            -0.74136    0.87286  -0.849 0.395686
## word_freq_will               -0.83257    0.83009  -1.003 0.315868
## word_freq_people             -0.67931    1.54250  -0.440 0.659651
## word_freq_report              1.49470    1.52999   0.977 0.328601
## word_freq_addresses           3.79242    3.15599   1.202 0.229497
## word_freq_free               16.97194    3.25216   5.219 1.80e-07 ***
## word_freq_business            6.62702    1.89473   3.498 0.000469 ***
## word_freq_email               2.04325    1.38103   1.480 0.139003
## word_freq_you                 1.33621    0.78861   1.694 0.090190 .
## word_freq_credit             19.37542   12.20030   1.588 0.112261
## word_freq_your                3.21989    0.72797   4.423 9.73e-06 ***
## word_freq_font                3.50716    3.22207   1.088 0.276384
## word_freq_000                12.73959    3.04291   4.187 2.83e-05 ***
## word_freq_money               9.52317    4.11568   2.314 0.020675 *
## word_freq_hp                -33.80919    6.53672  -5.172 2.31e-07 ***
## word_freq_hpl               -16.04721    7.68894  -2.087 0.036884 *
## word_freq_george           -285.73874   67.90933  -4.208 2.58e-05 ***
## word_freq_650                 2.79948    2.15137   1.301 0.193171
```

```
## word_freq_lab               -28.93802   18.64973  -1.552 0.120744
## word_freq_labs               -1.61675    1.90205  -0.850 0.395321
## word_freq_telnet             -2.27243   18.37563  -0.124 0.901580
## word_freq_857                13.82775   12.54547   1.102 0.270370
## word_freq_data              -10.72003    6.40100  -1.675 0.093984 .
## word_freq_415                 2.06551    6.94642   0.297 0.766201
## word_freq_85                -35.95809   17.10156  -2.103 0.035499 *
## word_freq_technology          6.85097    2.92242   2.344 0.019064 *
## word_freq_1999                0.30593    1.43795   0.213 0.831517
## word_freq_parts              -3.82580    3.34468  -1.144 0.252688
## word_freq_pm                -13.84814    6.01522  -2.302 0.021325 *
## word_freq_direct              1.64366    3.98998   0.412 0.680378
## word_freq_cs               -269.35562  226.75309  -1.188 0.234880
## word_freq_meeting           -36.73497   13.20450  -2.782 0.005402 **
## word_freq_original           -2.70881    2.77147  -0.977 0.328376
## word_freq_project           -26.51837   12.02840  -2.205 0.027479 *
## word_freq_re                -17.91064    3.89824  -4.595 4.34e-06 ***
## word_freq_edu               -34.11940    7.62493  -4.475 7.65e-06 ***
## word_freq_table              -2.68728    4.67553  -0.575 0.565458
## word_freq_conference        -38.94859   18.84014  -2.067 0.038704 *
## `char_freq_;`                -5.72536    2.49914  -2.291 0.021967 *
## `char_freq_(`                 0.94611    2.96111   0.320 0.749338
## `char_freq_[`                -4.20958    5.77397  -0.729 0.465964
## `char_freq_!`                 7.91974    2.12732   3.723 0.000197 ***
## `char_freq_$`                26.65787    4.49653   5.929 3.06e-09 ***
## `char_freq_#`                45.47778   23.37821   1.945 0.051738 .
## capital_run_length_longest   94.13178   26.46329   3.557 0.000375 ***
## capital_run_length_total      8.01102    3.79013   2.114 0.034545 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4316.6  on 3219  degrees of freedom
## Residual deviance: 1292.6  on 3162  degrees of freedom
## AIC: 1408.6
##
## Number of Fisher Scoring iterations: 13

results=cbind(ctest$capital_run_length_average,knn_pred1,knn_pred2,logm_pred1
,logm_pred2)
colnames(results)[1]<-"capital_run_length_average"
write.csv(results,file = "HW2_zl368_results.csv")
```

Based on the regression summary, we can see that emails with the following characteristics correlate with high probability of being spam:

1. High frequence of the following words:
   *our, over, remove, internet, mail, free, business, your, 000, money, hp, hpl,george, 85, technology, pm, meeting, confernce

2. High frequence of the following chars:
    * $, !

3. Long capital run length:
    * Length of longest capital run length > total capital run length > average capital run length

Some words such as “re” and “edu” negatively correlate with probability of being spam, which is very intuitive. “credit” is not significant and “money” is not one of the most significant; these observations are surprising. The most significant positive indicators are the word “remove” and the char “$”.