Data Set 1 (ds1) Despcription:

```
My first data set is a collection of data gathered about a bike sharing program in Portugal. It
has data abou the number of
casual and registered users each day for two years, it also has data about the weather, temperat
ure, humidity, windspeed,
and whether that day was a holiday. There are 16 columns and all are described below.

My target variable is cnt which is total number of bikes used which includes both registered and
casual users.
I want to predict the number of users depending on various attributes, this prediction can be us
eful for the company
to calculate the number of bikes they need on any particular day.

I have performed linear regression and kNN classification on the data set.

Number of Instances: 731
Number of Attributes: 16 (including target)

Attribute Description:

  1. - instant: record index
    2. - dteday : date
    3. - season : season (1:spring, 2:summer, 3:fall, 4:winter)
    4. - yr : year (0: 2011, 1:2012)
    5. - mnth : month ( 1 to 12)
    6. - holiday : weather day is holiday or not (extracted from http://dchr.dc.gov/page/holiday
-schedule)
    7. - weekday : day of the week
    8. - workingday : if day is neither weekend nor holiday is 1, otherwise is 0.
    9. + weathersit :
         - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
         - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
         - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered c
louds
         - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
    10. - temp : Normalized temperature in Celsius. The values are divided to 41 (max)
    11. - atemp: Normalized feeling temperature in Celsius. The values are divided to 50 (max)
    12. - hum: Normalized humidity. The values are divided to 100 (max)
    13. - windspeed: Normalized wind speed. The values are divided to 67 (max)
    14. - casual: count of casual users
    15. - registered: count of registered users
    16. - cnt: count of total rental bikes including both casual and registered

    The data set is obtained from "https://archive.ics.uci.edu/ml/machine-learning-databases/002
75/"
    You have to download the zip file and use the day.csv file.
  For full citation see the citation section at the end.
```

```
ds1 <- read.table("C:/Users/ADMIN/Desktop/Academia 2.0/Junior/Spring 2018/Machine Learning/HW/da
y.csv", header=TRUE, sep=",")
```

Using R functions on the data set.

```
names(ds1)
```

```
##  [1] "instant"    "dteday"     "season"     "yr"         "mnth"
##  [6] "holiday"    "weekday"    "workingday" "weathersit" "temp"
## [11] "atemp"      "hum"        "windspeed"  "casual"     "registered"
## [16] "cnt"
```

```
head(ds1)
```

```
##   instant     dteday season yr mnth holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1       0       6          0          2
## 2       2 2011-01-02      1  0    1       0       0          0          2
## 3       3 2011-01-03      1  0    1       0       1          1          1
## 4       4 2011-01-04      1  0    1       0       2          1          1
## 5       5 2011-01-05      1  0    1       0       3          1          1
## 6       6 2011-01-06      1  0    1       0       4          1          1
##       temp    atemp      hum windspeed casual registered  cnt
## 1 0.344167 0.363625 0.805833 0.1604460    331        654  985
## 2 0.363478 0.353739 0.696087 0.2485390    131        670  801
## 3 0.196364 0.189405 0.437273 0.2483090    120       1229 1349
## 4 0.200000 0.212122 0.590435 0.1602960    108       1454 1562
## 5 0.226957 0.229270 0.436957 0.1869000     82       1518 1600
## 6 0.204348 0.233209 0.518261 0.0895652     88       1518 1606
```

```
str(ds1)
```

```
## 'data.frame':    731 obs. of  16 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
##  $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
##  $ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
##  $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
##  $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

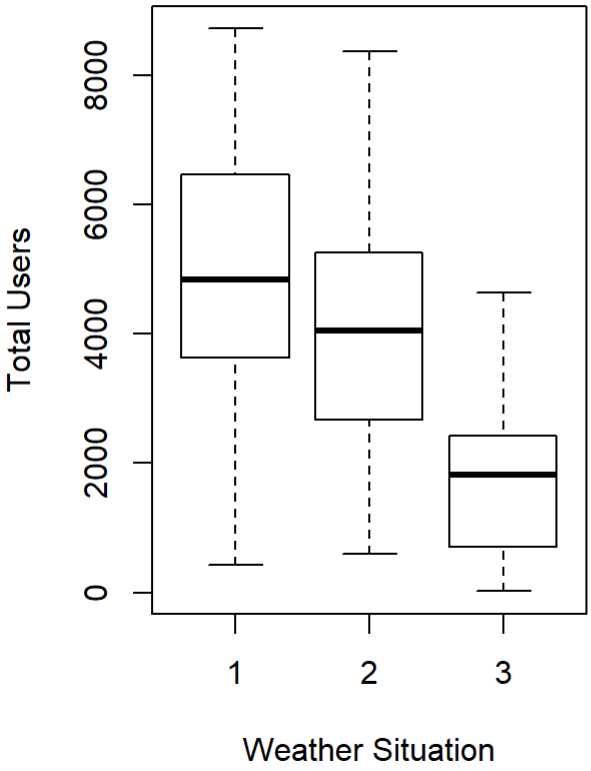
```
dim(ds1)
```

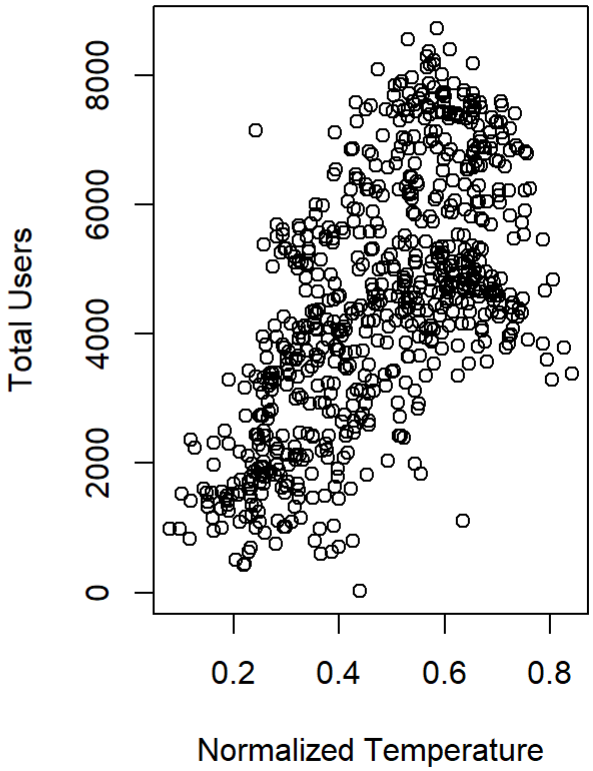
```
## [1] 731  16
```

```
summary(ds1)
```

```
##     instant             dteday        season            yr
##  Min.   :  1.0   2011-01-01:  1   Min.   :1.000   Min.   :0.0000
##  1st Qu.:183.5   2011-01-02:  1   1st Qu.:2.000   1st Qu.:0.0000
##  Median :366.0   2011-01-03:  1   Median :3.000   Median :1.0000
##  Mean   :366.0   2011-01-04:  1   Mean   :2.497   Mean   :0.5007
##  3rd Qu.:548.5   2011-01-05:  1   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :731.0   2011-01-06:  1   Max.   :4.000   Max.   :1.0000
##                  (Other)   :725
##       mnth          holiday           weekday        workingday
##  Min.   : 1.00   Min.   :0.00000   Min.   :0.000   Min.   :0.000
##  1st Qu.: 4.00   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000
##  Median : 7.00   Median :0.00000   Median :3.000   Median :1.000
##  Mean   : 6.52   Mean   :0.02873   Mean   :2.997   Mean   :0.684
##  3rd Qu.:10.00   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000
##  Max.   :12.00   Max.   :1.00000   Max.   :6.000   Max.   :1.000
##
##    weathersit         temp             atemp              hum
##  Min.   :1.000   Min.   :0.05913   Min.   :0.07907   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200
##  Median :1.000   Median :0.49833   Median :0.48673   Median :0.6267
##  Mean   :1.395   Mean   :0.49538   Mean   :0.47435   Mean   :0.6279
##  3rd Qu.:2.000   3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302
##  Max.   :3.000   Max.   :0.86167   Max.   :0.84090   Max.   :0.9725
##
##    windspeed           casual         registered        cnt
##  Min.   :0.02239   Min.   :   2.0   Min.   :  20   Min.   :  22
##  1st Qu.:0.13495   1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152
##  Median :0.18097   Median : 713.0   Median :3662   Median :4548
##  Mean   :0.19049   Mean   : 848.2   Mean   :3656   Mean   :4504
##  3rd Qu.:0.23321   3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956
##  Max.   :0.50746   Max.   :3410.0   Max.   :6946   Max.   :8714
##
```

```
par(mfrow=c(1,2))
plot(as.factor(ds1$weathersit),ds1$cnt, ylab = "Total Users", xlab = "Weather Situation")
plot(ds1$atemp,ds1$cnt, ylab = "Total Users", xlab = "Normalized Temperature")
```





From the first plot we can see that more users on a clear day. From the second plot we see that as the normalized temperature rises the count of users tends to rise.

Converting to factors, because all these variable have integer values which represent characters.

```
ds1$season <- as.factor(ds1$season)
ds1$mnth <- as.factor(ds1$mnth)
ds1$weekday <- as.factor(ds1$weekday)
ds1$workingday <- as.factor(ds1$workingday)
ds1$weathersit <- as.factor(ds1$weathersit)
```

Splitting the data randomly into 75% train and 25% test.

```
set.seed(1234)
i <- sample(1:nrow(ds1), nrow(ds1)*0.75, replace=FALSE)
train <- ds1[i,]
test <- ds1[-i,]
```

Building a linear model, where the target is total number of casual users and the predictors are humidity, average temperature, weather situation, season and wind speed.

```
lm1 <- lm(cnt~atemp*hum*temp+atemp+temp+hum+mnth+weathersit+season, data=train)
pred <- predict(lm1, newdata=test)
cor(pred, test$cnt)
```

```
## [1] 0.7462853
```

```
summary(lm1)
```

```
##
## Call:
## lm(formula = cnt ~ atemp * hum * temp + atemp + temp + hum +
##     mnth + weathersit + season, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3902.6  -992.2   -43.0   986.0  3018.4
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2988.18    1848.47  -1.617 0.106573
## atemp             131.36   14899.29   0.009 0.992969
## hum              2297.16    3309.91   0.694 0.487974
## temp            36166.35   15632.62   2.314 0.021080 *
## mnth2            -391.37     276.95  -1.413 0.158213
## mnth3            -492.00     313.52  -1.569 0.117183
## mnth4            -445.82     494.49  -0.902 0.367696
## mnth5              25.79     509.73   0.051 0.959660
## mnth6            -286.07     536.50  -0.533 0.594114
## mnth7            -377.02     585.61  -0.644 0.519985
## mnth8            -119.67     571.43  -0.209 0.834194
## mnth9             240.12     513.55   0.468 0.640286
## mnth10           -544.21     450.25  -1.209 0.227332
## mnth11           -941.68     434.99  -2.165 0.030853 *
## mnth12           -826.40     343.00  -2.409 0.016324 *
## weathersit2      -285.48     138.07  -2.068 0.039157 *
## weathersit3     -2383.48     349.58  -6.818 2.55e-11 ***
## season2           791.14     359.70   2.199 0.028283 *
## season3          1102.74     405.21   2.721 0.006717 **
## season4          1701.12     329.10   5.169 3.35e-07 ***
## atemp:hum       18281.99   24917.07   0.734 0.463451
## atemp:temp     -34140.18    9653.38  -3.537 0.000441 ***
## hum:temp       -43341.77   26078.44  -1.662 0.097115 .
## atemp:hum:temp  28879.20   16554.54   1.744 0.081660 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1217 on 524 degrees of freedom
## Multiple R-squared:  0.613,  Adjusted R-squared:  0.5961
## F-statistic: 36.09 on 23 and 524 DF,  p-value: < 2.2e-16
```

Calculating MSE and RSE.

```
mse <- mean(lm1$residuals^2)
mse
```

```
## [1] 1416148
```

```
rse <- sqrt(mse)
rse
```

```
## [1] 1190.02
```

Concverting the following attributes to numeric so that they can be used for kNN regression.

```
ds1$weathersit <- as.integer(ds1$weathersit)
ds1$mnth <- as.integer(ds1$mnth)
ds1$season <- as.integer(ds1$season)
```

Perfoming kNN regression on scaled data. The predictors are the same as the ones in the linear model.

```
library('caret')
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
df <- data.frame(scale(ds1[,c(3,9,10,11,12,13,16)])) # scaling total number of casual users, hum
idity, average temperature,                                                    # weather si
tuation, season, mnth and wind speed.
train2 <- df[i,]
test2 <- df[-i,]

fit <- knnreg(train2[,1:6],train2[,7],k=12)
predictions <- predict(fit, test2[,1:6])
cor(predictions, test2$cnt)                          # calculating accuracy
```

```
## [1] 0.7902023
```

```
summary(fit)
```

```
##         Length Class  Mode
## learn   2      -none- list
## k       1      -none- numeric
## theDots 0      -none- list
```

```
r_squared <- 1 - sum((test2$cnt-predictions)^2)/sum((test2$cnt-mean(test2$cnt))^2)
r_squared
```

```
## [1] 0.621735
```

Summary doesn't tell us much, but R squared will.

Analysing Models/Algorithm Results:

```
Both linear model and kNN regression have good accuracy and good R squared.

The linear model has an accuracy of 0.7462853 and R squared of 0.613. This is high accuracy and
the model is able to
explain 61.3% of the variabiity. It also has a low p value which tells us that the predictors ar
e predicting well.

kNN regression gives an accuracy of 0.7902023 and R squared of 0.621735. This is high accuracy a
nd the model is able to
explain 62.1735% of the variabiity.

kNN regression just edges the linear model in both accuracy and explaining variability, and is t
he best for this data set.
```

Data Set 2 (ds2) Despcription:

```
The data are MC generated (see below) to simulate registration of high energy gamma particles an
d hadron particles in a        ground-based atmospheric Cherenkov gamma telescope using an imagi
ng technique.

I'm trying to predict whether the particles are gamma or hadron based on predictors 1-10 describ
ed in the attribute            description.

I have performed logistic regression and kNN classification on the data.

Number of Instances: 19020
Number of Attributes: 11 (including target)

Attribute Description:

    1.   fLength:  continuous  # major axis of ellipse [mm]
    2.   fWidth:   continuous  # minor axis of ellipse [mm]
    3.   fSize:    continuous  # 10-log of sum of content of all pixels [in #phot]
    4.   fConc:    continuous  # ratio of sum of two highest pixels over fSize  [ratio]
    5.   fConc1:   continuous  # ratio of highest pixel over fSize  [ratio]
    6.   fAsym:    continuous  # distance from highest pixel to center, projected onto major axi
s [mm]
    7.   fM3Long:  continuous  # 3rd root of third moment along major axis  [mm]
    8.   fM3Trans: continuous  # 3rd root of third moment along minor axis  [mm]
    9.   fAlpha:   continuous  # angle of major axis with vector to origin [deg]
    10.  fDist:    continuous  # distance from origin to center of ellipse [mm]
    11.  class:    g,h         # gamma (signal), hadron (background)

Target Distribution:
    g = gamma (signal):     12332
    h = hadron (background): 6688

The data set is obtained from "http://archive.ics.uci.edu/ml/machine-learning-databases/magic/"
The data is in magic04.data "http://archive.ics.uci.edu/ml/machine-learning-databases/magic/magi
c04.data"
For full citation see the citation section at the end.
```

```
ds2 <- read.table("C:/Users/ADMIN/Desktop/Academia 2.0/Junior/Spring 2018/Machine Learning/HW/ma
gic04.data", header=FALSE, sep=",")
```

Adding column names for easy interpretation.

```
colnames(ds2) <- c("fLength","fWidth","fSize","fConc","fConc1","fAsym","fM3Long","fM3Trans","fAl
pha","fDist","class")
```

Using R functions to gauge data set.

```
dim(ds2)
```

```
## [1] 19020    11
```

```
summary(ds2)
```

```
##     fLength            fWidth           fSize           fConc        
##  Min.   :  4.284   Min.   :  0.00   Min.   :1.941   Min.   :0.0131  
##  1st Qu.: 24.336   1st Qu.: 11.86   1st Qu.:2.477   1st Qu.:0.2358  
##  Median : 37.148   Median : 17.14   Median :2.740   Median :0.3542  
##  Mean   : 53.250   Mean   : 22.18   Mean   :2.825   Mean   :0.3803  
##  3rd Qu.: 70.122   3rd Qu.: 24.74   3rd Qu.:3.102   3rd Qu.:0.5037  
##  Max.   :334.177   Max.   :256.38   Max.   :5.323   Max.   :0.8930  
##      fConc1           fAsym             fM3Long           fM3Trans       
##  Min.   :0.0003   Min.   :-457.916   Min.   :-331.78   Min.   :-205.8947  
##  1st Qu.:0.1285   1st Qu.: -20.587   1st Qu.: -12.84   1st Qu.: -10.8494  
##  Median :0.1965   Median :   4.013   Median :  15.31   Median :   0.6662  
##  Mean   :0.2147   Mean   :  -4.332   Mean   :  10.55   Mean   :   0.2497  
##  3rd Qu.:0.2852   3rd Qu.:  24.064   3rd Qu.:  35.84   3rd Qu.:  10.9464  
##  Max.   :0.6752   Max.   : 575.241   Max.   : 238.32   Max.   : 179.8510  
##      fAlpha           fDist         class      
##  Min.   : 0.000   Min.   :  1.283   g:12332  
##  1st Qu.: 5.548   1st Qu.:142.492   h: 6688  
##  Median :17.680   Median :191.851            
##  Mean   :27.646   Mean   :193.818            
##  3rd Qu.:45.884   3rd Qu.:240.564            
##  Max.   :90.000   Max.   :495.561
```

```
names(ds2)
```

```
##  [1] "fLength"  "fWidth"   "fSize"    "fConc"    "fConc1"   "fAsym"   
##  [7] "fM3Long"  "fM3Trans" "fAlpha"   "fDist"    "class"
```

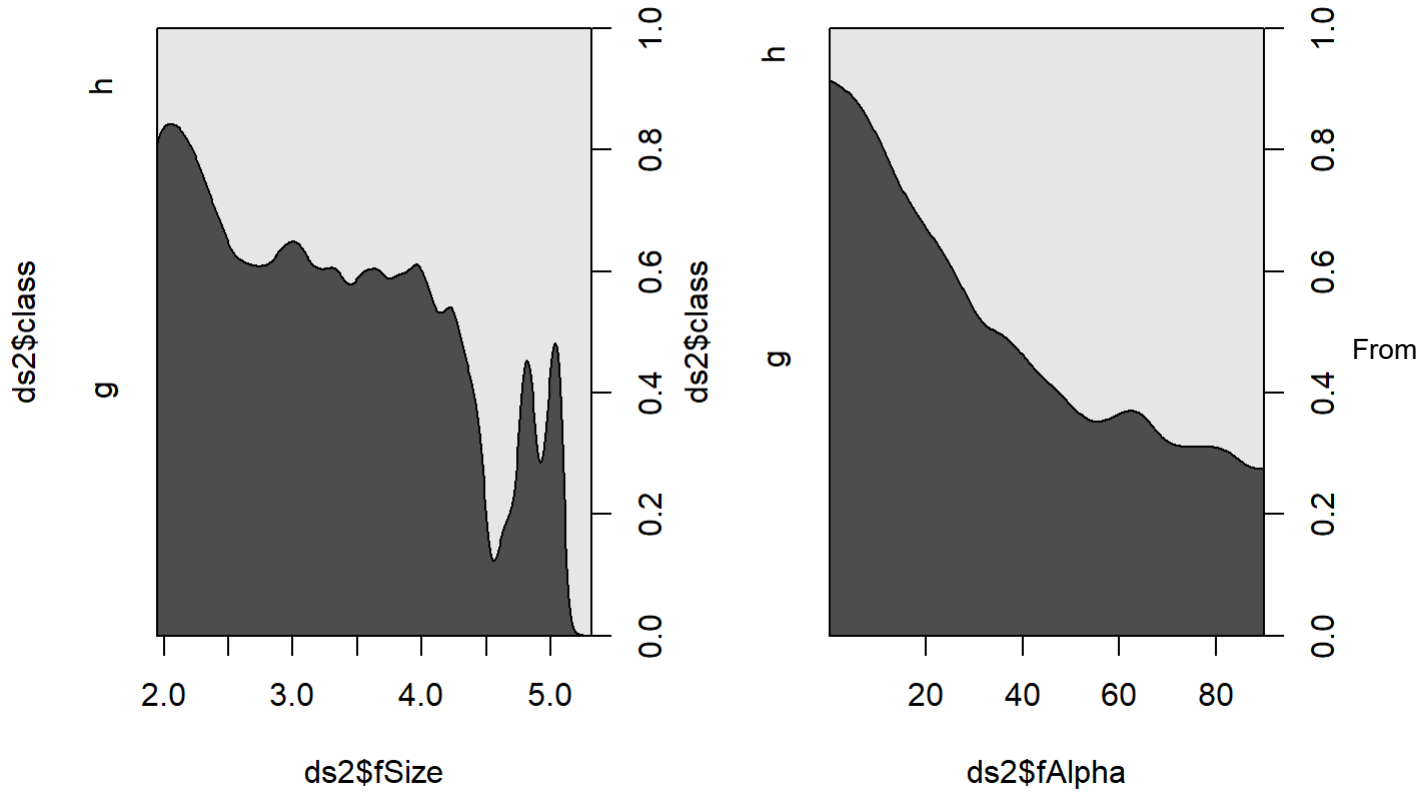
```
head(ds2)
```

```
##    fLength   fWidth  fSize  fConc fConc1    fAsym  fM3Long fM3Trans
## 1  28.7967  16.0021 2.6449 0.3918 0.1982  27.7004  22.0110  -8.2027
## 2  31.6036  11.7235 2.5185 0.5303 0.3773  26.2722  23.8238  -9.9574
## 3 162.0520 136.0310 4.0612 0.0374 0.0187 116.7410 -64.8580 -45.2160
## 4  23.8172   9.5728 2.3385 0.6147 0.3922  27.2107  -6.4633  -7.1513
## 5  75.1362  30.9205 3.1611 0.3168 0.1832  -5.5277  28.5525  21.8393
## 6  51.6240  21.1502 2.9085 0.2420 0.1340  50.8761  43.1887   9.8145
##    fAlpha    fDist class
## 1 40.0920  81.8828     g
## 2  6.3609 205.2610     g
## 3 76.9600 256.7880     g
## 4 10.4490 116.7370     g
## 5  4.6480 356.4620     g
## 6  3.6130 238.0980     g
```

```
str(ds2)
```

```
## 'data.frame':    19020 obs. of  11 variables:
##  $ fLength : num  28.8 31.6 162.1 23.8 75.1 ...
##  $ fWidth  : num  16 11.72 136.03 9.57 30.92 ...
##  $ fSize   : num  2.64 2.52 4.06 2.34 3.16 ...
##  $ fConc   : num  0.3918 0.5303 0.0374 0.6147 0.3168 ...
##  $ fConc1  : num  0.1982 0.3773 0.0187 0.3922 0.1832 ...
##  $ fAsym   : num  27.7 26.27 116.74 27.21 -5.53 ...
##  $ fM3Long : num  22.01 23.82 -64.86 -6.46 28.55 ...
##  $ fM3Trans: num  -8.2 -9.96 -45.22 -7.15 21.84 ...
##  $ fAlpha  : num  40.09 6.36 76.96 10.45 4.65 ...
##  $ fDist   : num  81.9 205.3 256.8 116.7 356.5 ...
##  $ class   : Factor w/ 2 levels "g","h": 1 1 1 1 1 1 1 1 1 1 ...
```

```
par(mfrow=c(1,2))
cdplot(ds2$class~ds2$fSize)
cdplot(ds2$class~ds2$fAlpha)
```



the first cdplot we can see that gamma particles tend to have smaller size. From the second cdplot we can see that gamma particles tend to have lower alpha.

Splitting the data randomly into 75% train and 25% test.

```
set.seed(1234)
i <- sample(nrow(ds2), nrow(ds2)*0.75, replace=FALSE)
train <- ds2[i,]
test <- ds2[-i,]
```

Creating logistic model with class as the target and fLength, fWidth, fSize, fConc, fConc1, fAsym, fM3Long, fM3Trans, fAlpha and fDist, as predictors

```
glm1 = glm(class~., data=train, family=binomial)
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, "h", "g")
table(pred, test$class)
```

```
##
## pred    g    h
##    g 2803  648
##    h  307  997
```

```
summary(glm1)
```

```
##
## Call:
## glm(formula = class ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9536  -0.6636  -0.4644   0.6586   2.4908
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.442e+00  3.571e-01 -18.041  < 2e-16 ***
## fLength      2.922e-02  1.215e-03  24.048  < 2e-16 ***
## fWidth       5.794e-03  2.842e-03   2.039   0.0415 *
## fSize        6.359e-01  1.104e-01   5.762 8.30e-09 ***
## fConc       -7.612e-01  6.011e-01  -1.266   0.2054
## fConc1       6.523e+00  8.697e-01   7.501 6.35e-14 ***
## fAsym        9.901e-05  4.957e-04   0.200   0.8417
## fM3Long     -7.092e-03  6.134e-04 -11.561  < 2e-16 ***
## fM3Trans    -8.433e-04  1.328e-03  -0.635   0.5255
## fAlpha       4.516e-02  9.827e-04  45.959  < 2e-16 ***
## fDist        2.351e-04  3.445e-04   0.682   0.4950
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 18533  on 14264  degrees of freedom
## Residual deviance: 13126  on 14254  degrees of freedom
## AIC: 13148
##
## Number of Fisher Scoring iterations: 5
```

Calculating te accuracy of the model.

```
paste("accuracy = ", sum(pred==test$class)/NROW(test$class)*100, "%")
```

```
## [1] "accuracy =  79.9158780231335 %"
```

Splitting the data randomly into 75% train and 25% test.

```
set.seed(1234)  # setting a seed gets the same results every time
ind <- sample(2, nrow(ds2), replace=TRUE, prob=c(0.75, 0.25))
ds2.train <- ds2[ind==1, 1:10]
ds2.test <- ds2[ind==2, 1:10]
ds2.trainLabels <- ds2[ind==1, 11]
ds2.testLabels <- ds2[ind==2, 11]
```

Performing kNN classification on the data. The predictors are the same as the ones for the logistic model.

```
library(class)
ds2_pred <- knn(train=ds2.train, test=ds2.test, cl=ds2.trainLabels, k=10)
```

```
summary(ds2_pred)
```

```
##    g    h
## 3594 1140
```

Summary doesn't tell us much.

Calculating accuracy.

```
results <- ds2_pred == ds2.testLabels
acc <- length(which(results==TRUE)) / length(results)
acc
```

```
## [1] 0.8073511
```

Analysing Models/Algorithm Results:

```
Both logistic model and kNN classification have good accuracy.

The logistic model has an accuracy of 79.9158780231335%.

kNN classification gives us an accuracy of 80.62949%.

kNN classifiaction gives is slightly better for this data set.
```

Citations:


```
[1] Contains link for data sets.

1. Data Set 1 (Bike Sharing in Portugal)

    [1] Lichman, M. (2013). UCI Machine Learning Repository
                                        [https://archive.ics.uci.edu/ml/machine-learning-database
s/00275/]. Irvine, CA: University of California, School of             Information and Computer
Science.

    [2] Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and backgro
und knowledge", Progress in             Artificial Intelligence (2013): pp. 1-15, Springer Berli
n Heidelberg, doi:10.1007/s13748-013-0040-3.

2. Data Set 2 (Gamma and hadron particle data set)

    [1] Lichman, M. (2013). UCI Machine Learning Repository
                                        [http://archive.ics.uci.edu/ml/machine-learning-database
s/magic/]. Irvine, CA: University of California, School of             Information and Computer
Science.
```