Assignment 4

Paraskumar, Karthigeyan, Shanmuga Priya, Raju, Niranjan

14/08/2019

Data preparataion

Here we take the "take offer" as the positive outcome.

```
mort <- read.csv("sagedat2.csv",stringsAsFactors = FALSE)
mort$resp[mort$takeoffer == "take offer"] <- 1
mort$resp[mort$takeoffer == "decline offer"] <- 0
```

Question 1 : Comparing Logit and Probit

Logit equation

The logit fuction uses the following link equation :

$$f(\mu_Y) = \ln\left(\frac{P}{1-P}\right)$$

It can transform $\log(odds)$ to odds ratio. ## Probit Equation The probit function uses the following link equation as given below:

$$f(\mu_Y) = \Phi^{-1}(P)$$

It can be interpreted as the inverse normal CDF.

Logit model

```
m1.logit <- glm(data=mort, resp~Mortgage+Famsize,
                family=binomial(link = "logit"))
exp(m1.logit$coefficients)
```

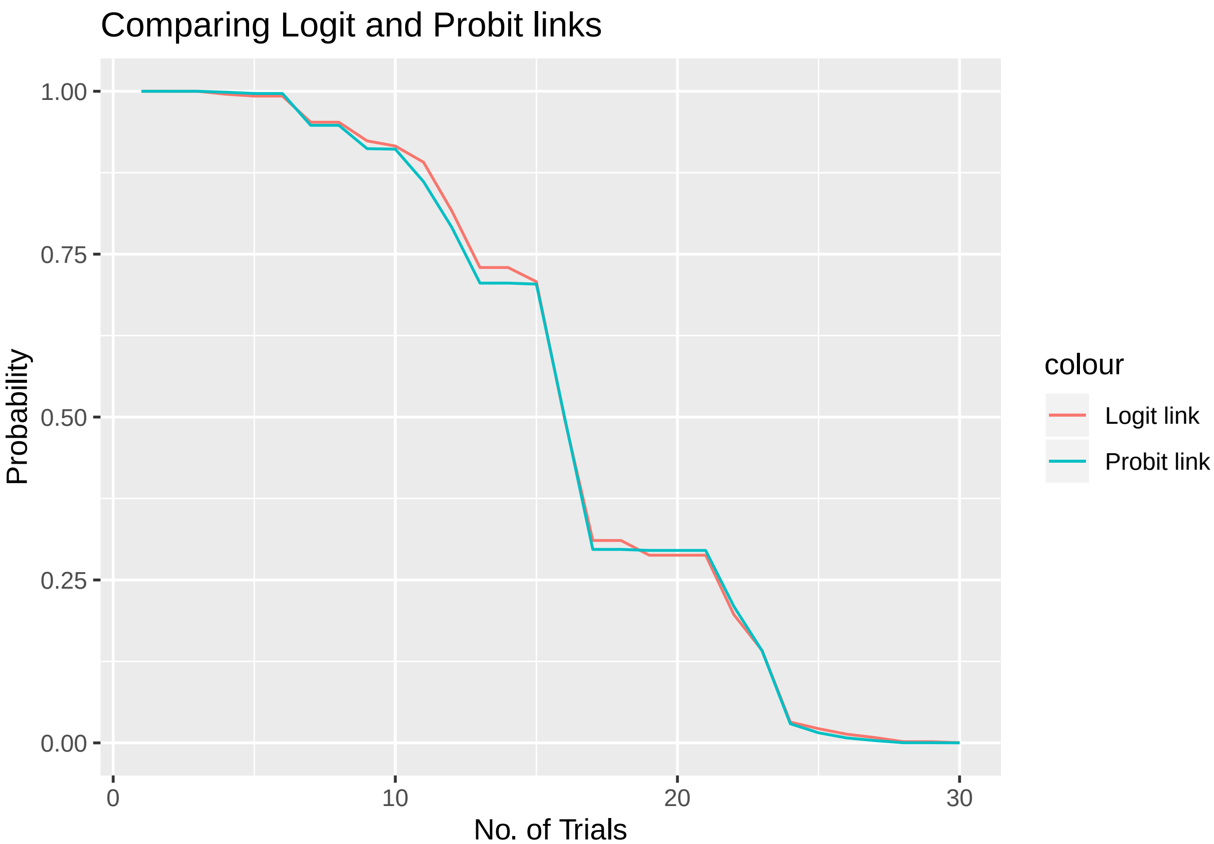
```
##  (Intercept)     Mortgage      Famsize
## 8.133346e-09 1.005025e+00 1.100703e+01
```

Probit model

```
m1.probit <- glm(data=mort, resp~Mortgage+Famsize,
                 family = binomial(link = "probit"))
exp(m1.probit$coefficients)
```

```
##  (Intercept)     Mortgage      Famsize
## 3.467139e-05 1.002711e+00 3.853620e+00
```

Difference between the two models



As can be noticed there's not much of a difference between the logit and probit probabilities.

Question 2: Switching the response variable

Data Preparation

Here we take the "decline offer" as the response variable.

```
mort2 <- read.csv("sagedat2.csv",stringsAsFactors = FALSE)
mort2$resp[mort$takeoffer == "take offer"] <- 0
mort2$resp[mort$takeoffer == "decline offer"] <- 1
```

Data Model

This model gives us teh probability of decling the offer.

```
m2.logit <- glm(data=mort2, resp~Mortgage+Famsize, family=binomial(link = "logit"))
exp(m2.logit$coefficients)
```

```
##  (Intercept)     Mortgage      Famsize
## 1.229506e+08 9.949999e-01 9.085102e-02
```

```
predict(m2.logit,type = "r")
```

```
##            1            2            3            4            5
## 2.704824e-01 7.119946e-01 9.981875e-01 5.025892e-01 9.866942e-01
##            6            7            8            9           10
## 9.680597e-01 9.998351e-01 4.515215e-03 1.088532e-01 6.893950e-01
##           11           12           13           14           15
## 4.755059e-02 7.431927e-03 9.918973e-01 2.704824e-01 2.741568e-06
##           16           17           18           19           20
## 7.119946e-01 4.755059e-02 7.431927e-03 9.782232e-01 7.119946e-01
##           21           22           23           24           25
## 7.614074e-02 3.361030e-05 8.031909e-01 8.581277e-01 2.922715e-01
##           26           27           28           29           30
## 8.407870e-02 9.981875e-01 6.893950e-01 2.741568e-06 1.834055e-01
```

Getting back original probabilities

We can get the original model probabilities for "take offer" by \$ 1-P("decline offer")\$ as the sum of the two probabilities is 1.

```
(predict(m1.logit,type = "r"))+(predict(m2.logit, type="r"))
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
## 26 27 28 29 30
##  1  1  1  1  1
```

Relation between the coefficients

The product of the exponentials of the coefficients between the two models is equal to one.

```
#Product of coefficients
exp(m1.logit$coefficients)*exp(m2.logit$coefficients)
```

```
## (Intercept)    Mortgage     Famsize
##           1           1           1
```

Question : German Credit Data

Data preparation

First we remove unwanted columns and code the correct attibutes so that we can undersstan the data. Here if the response is "1" mean good credit risk.

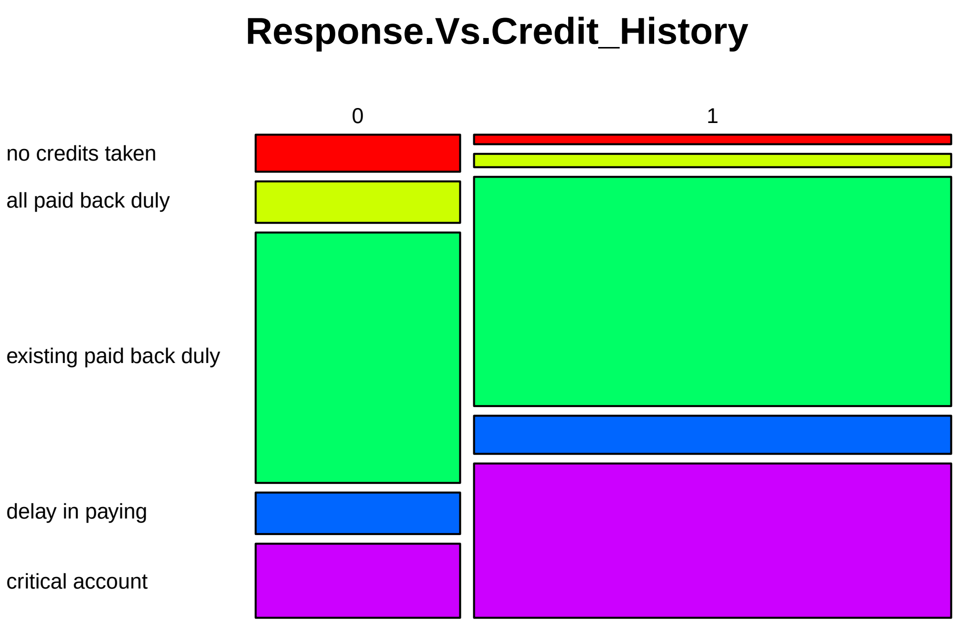
```
german<-read.table("german.data",header = FALSE)
german[,6:20]<-NULL
names(german)<-c("Account Status","Duration in month","Credit history","Purpose","Credit Amount","Respo
german$Response[german$Response == 2] <- 0 #Bad
german$Response[german$Response == 1] <- 1 #Good
levels(german$`Account Status`)<-c("< 0 DM","Between 0-200 DM",">= 200 DM","no checking")
levels(german$`Credit history`)<-c("no credits taken","all paid back duly","existing paid back duly",
                                   "delay in paying","critical account")
levels(german$Purpose)<-c("new car","used car","others","furniture","radio/television","domestic applia
                          "repairs","education","retraining","business")
```

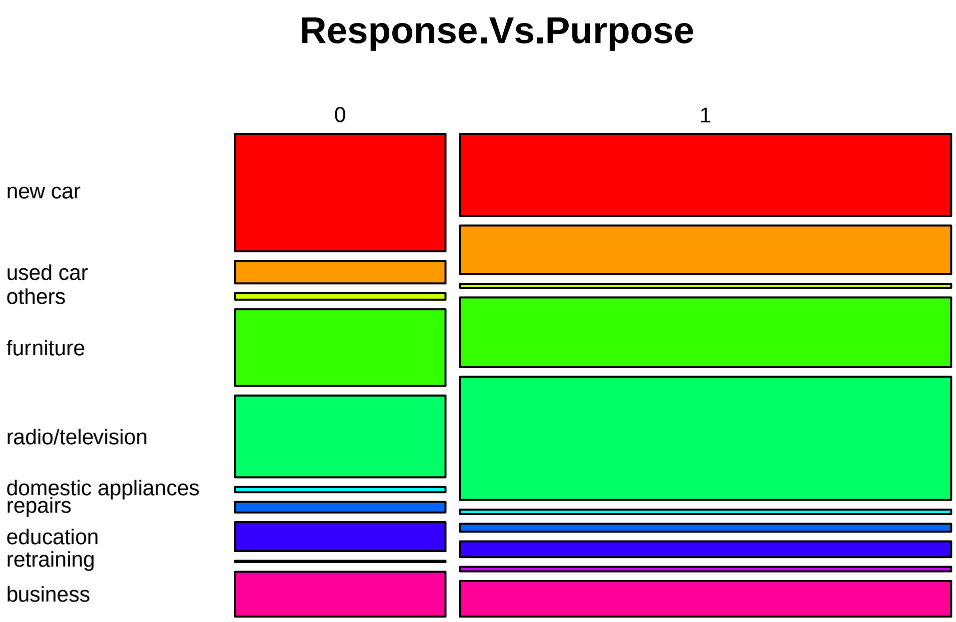
Univariate Analysis

```
table(german$Response)
```

```
##
##   0   1
## 300 700
```

Bivariate Analysis





Data Splitting

we use a 70:30 split.

```
train.index <- sample(1:nrow(german), nrow(german)*.7)
train.german <- german[train.index,]
test.german <- german[-train.index,]
```

Data Modeling

We use the step function to get the parsimonioous model with lowest AIC score. A summary of the model is given below.

```
fullmodel <- glm(data=train.german, Response~., family=binomial(link = "logit"))
zeromodel <- glm(data=train.german, Response~1, family=binomial(link = "logit"))
stepmodel <-  step(zeromodel, list(lower=formula(zeromodel),
                                   upper=formula(fullmodel)),
                                   direction="both",trace=0)

summary(stepmodel)
```

```
##
## Call:
## glm(formula = Response ~ `Account Status` + `Duration in month` +
##     `Credit history` + Purpose, family = binomial(link = "logit"),
##     data = train.german)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5246  -0.8295   0.4777   0.7793   2.0432
##
## Coefficients:
##                                              Estimate Std. Error z value
```

```
## (Intercept)                             -0.387144   0.546005  -0.709
## `Account Status`Between 0-200 DM          0.417562   0.231434   1.804
## `Account Status`>= 200 DM                 0.965857   0.425658   2.269
## `Account Status`no checking               1.710389   0.253562   6.745
## `Duration in month`                      -0.040885   0.008041  -5.084
## `Credit history`all paid back duly       -0.022758   0.601534  -0.038
## `Credit history`existing paid back duly   0.966211   0.478875   2.018
## `Credit history`delay in paying           0.949074   0.540335   1.756
## `Credit history`critical account          1.576459   0.500944   3.147
## Purposeused car                           1.256565   0.383133   3.280
## Purposeothers                             1.435284   0.873476   1.643
## Purposefurniture                          0.307773   0.279930   1.099
## Purposeradio/television                   0.735535   0.269826   2.726
## Purposedomestic appliances               -0.437418   0.912130  -0.480
## Purposerepairs                            0.566025   0.634333   0.892
## Purposeeducation                         -0.117566   0.446682  -0.263
## Purposeretraining                        15.474373 600.275082   0.026
## Purposebusiness                           0.559491   0.359239   1.557
##                                         Pr(>|z|)
## (Intercept)                              0.47829
## `Account Status`Between 0-200 DM         0.07119 .
## `Account Status`>= 200 DM                0.02326 *
## `Account Status`no checking             1.53e-11 ***
## `Duration in month`                     3.69e-07 ***
## `Credit history`all paid back duly       0.96982
## `Credit history`existing paid back duly  0.04363 *
## `Credit history`delay in paying          0.07901 .
## `Credit history`critical account         0.00165 **
## Purposeused car                          0.00104 **
## Purposeothers                            0.10034
## Purposefurniture                         0.27157
## Purposeradio/television                  0.00641 **
## Purposedomestic appliances               0.63154
## Purposerepairs                           0.37222
## Purposeeducation                         0.79240
## Purposeretraining                        0.97943
## Purposebusiness                          0.11937
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 835.74  on 699  degrees of freedom
## Residual deviance: 688.74  on 682  degrees of freedom
## AIC: 724.74
##
## Number of Fisher Scoring iterations: 14
```

```r
m3<-glm(formula = Response ~ `Account Status` + `Duration in month` +
             `Credit history` + Purpose, family = binomial(link = "logit"),
           data = train.german)
```

Model Interpretation

The probability of a customer being a good credit risk is a function of "Account Status", "Duration in Month", "Credit History" and "Purpose". The multiplicative factor for each of the Xs is given below.

```
exp(m3$coefficients)
```

```
##                                (Intercept)
##                               6.789930e-01
##         `Account Status`Between 0-200 DM
##                               1.518255e+00
##                `Account Status`>= 200 DM
##                               2.627039e+00
##              `Account Status`no checking
##                               5.531110e+00
##                      `Duration in month`
##                               9.599395e-01
##       `Credit history`all paid back duly
##                               9.774991e-01
## `Credit history`existing paid back duly
##                               2.627968e+00
##           `Credit history`delay in paying
##                               2.583316e+00
##          `Credit history`critical account
##                               4.837796e+00
##                           Purposeused car
##                               3.513332e+00
##                             Purposeothers
##                               4.200838e+00
##                          Purposefurniture
##                               1.360392e+00
##                   Purposeradio/television
##                               2.086598e+00
##               Purposedomestic appliances
##                               6.457014e-01
##                            Purposerepairs
##                               1.761253e+00
##                          Purposeeducation
##                               8.890818e-01
##                         Purposeretraining
##                               5.253333e+06
##                           Purposebusiness
##                               1.749782e+00
```

Model Validation

Confusion Matrix

```
train.german$prob <- predict(m3,type="r")
CUTOFF <- quantile(train.german$prob,.65)

train.german$pred <- ifelse(train.german$prob > CUTOFF,1,0)
table(train.german$pred, train.german$Response)
```

```
##
##       0   1
##   0 177 279
##   1  22 222
```

```r
#Conf(x = train.german$pred,ref=train.german$Response)

test.german$pred <- ifelse(predict(m3, test.german)>CUTOFF,1,0)
table(test.german$pred, test.german$Response)
```

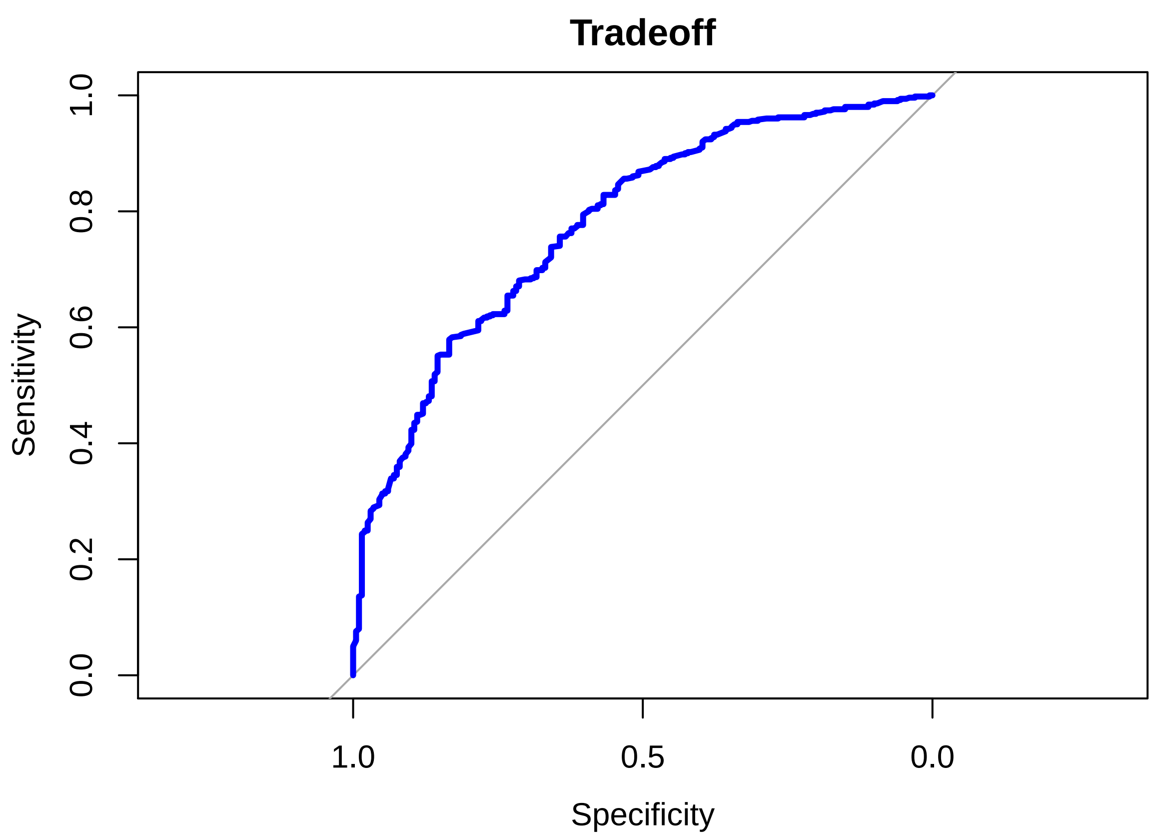
```
##
##       0   1
##   0  76  54
##   1  25 145
```

```r
#Conf(x = test.german$pred,ref=test.german$Response)
```

ROC chart

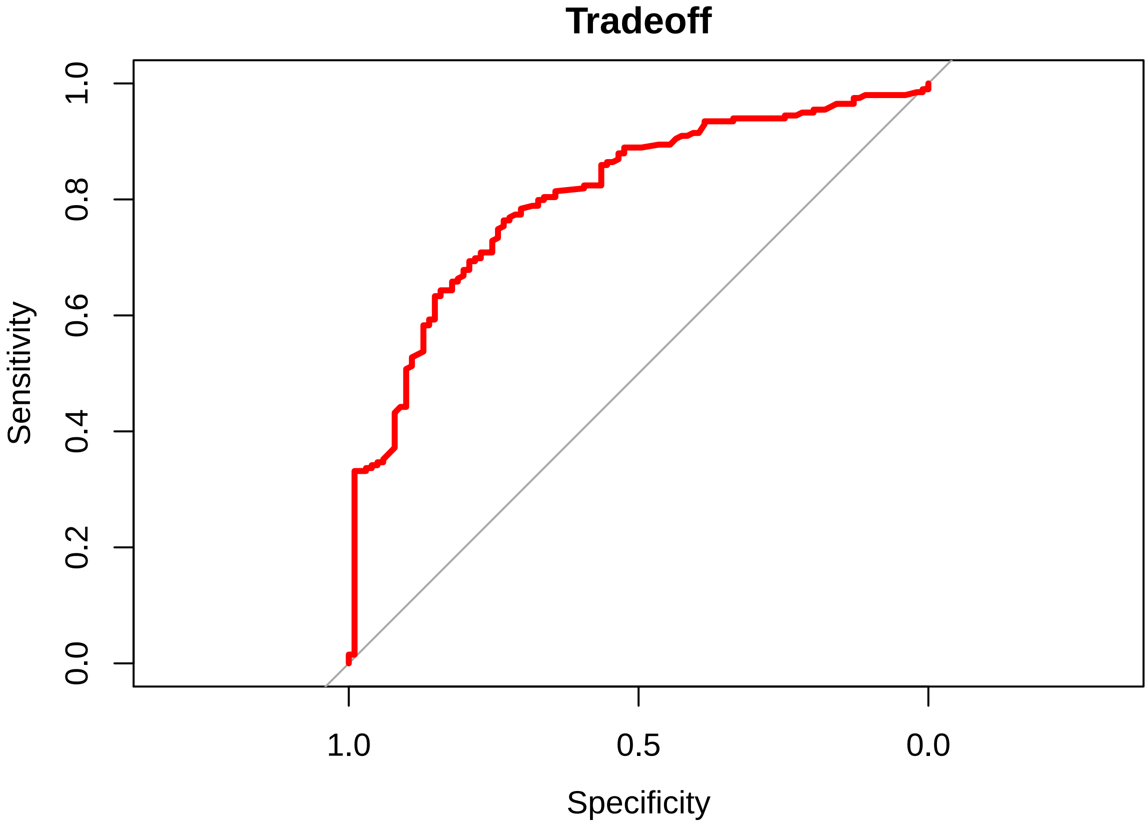
```r
train.german$prob <- predict(m3,type="r")
plot(roc(train.german$Response, train.german$prob, direction="<"),
     col="blue", lwd=3, main="Tradeoff")
```

```
## Setting levels: control = 0, case = 1
```




```r
test.german$prob <- predict(m3,test.german, type="r")
plot(roc(test.german$Response, test.german$prob, direction="<"),
     col="red", lwd=3, main="Tradeoff")
```

```
## Setting levels: control = 0, case = 1
```



Lift and Gain Chart

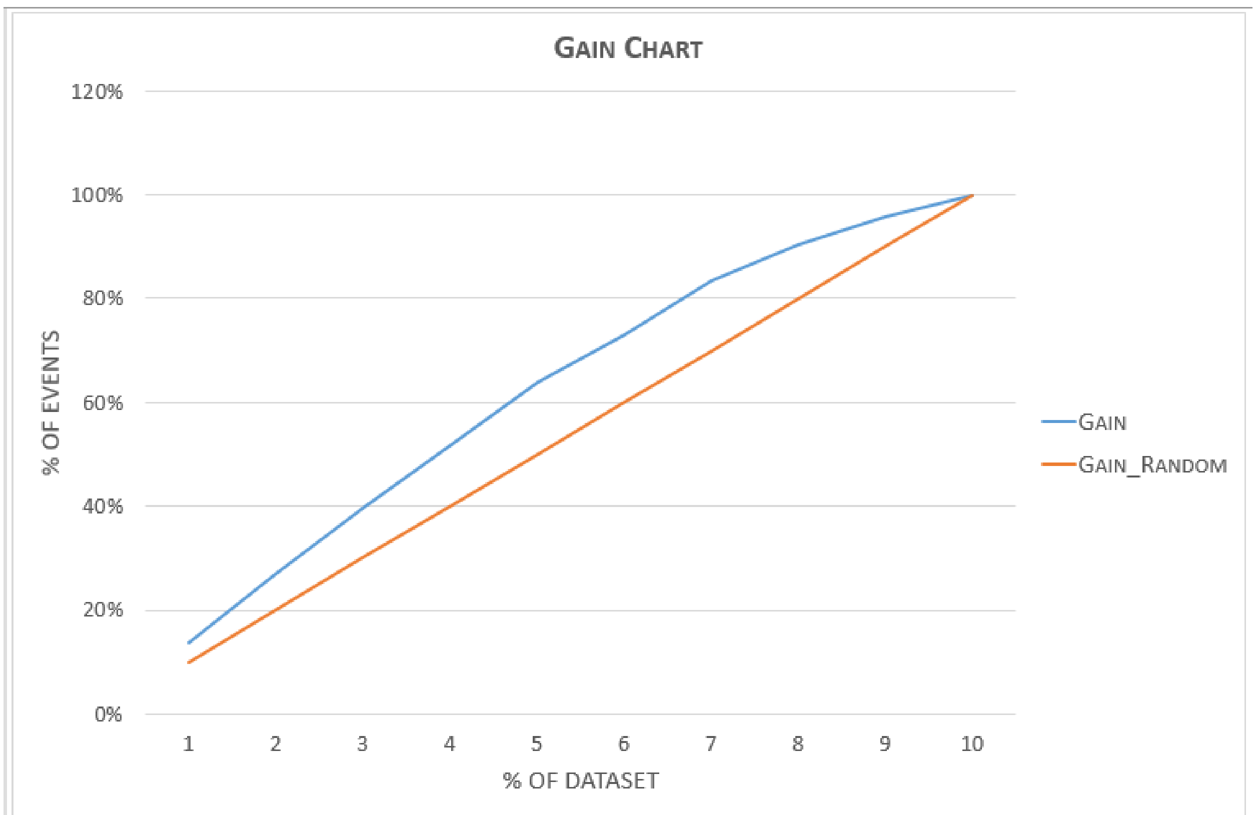


Figure 1: Gain Chart

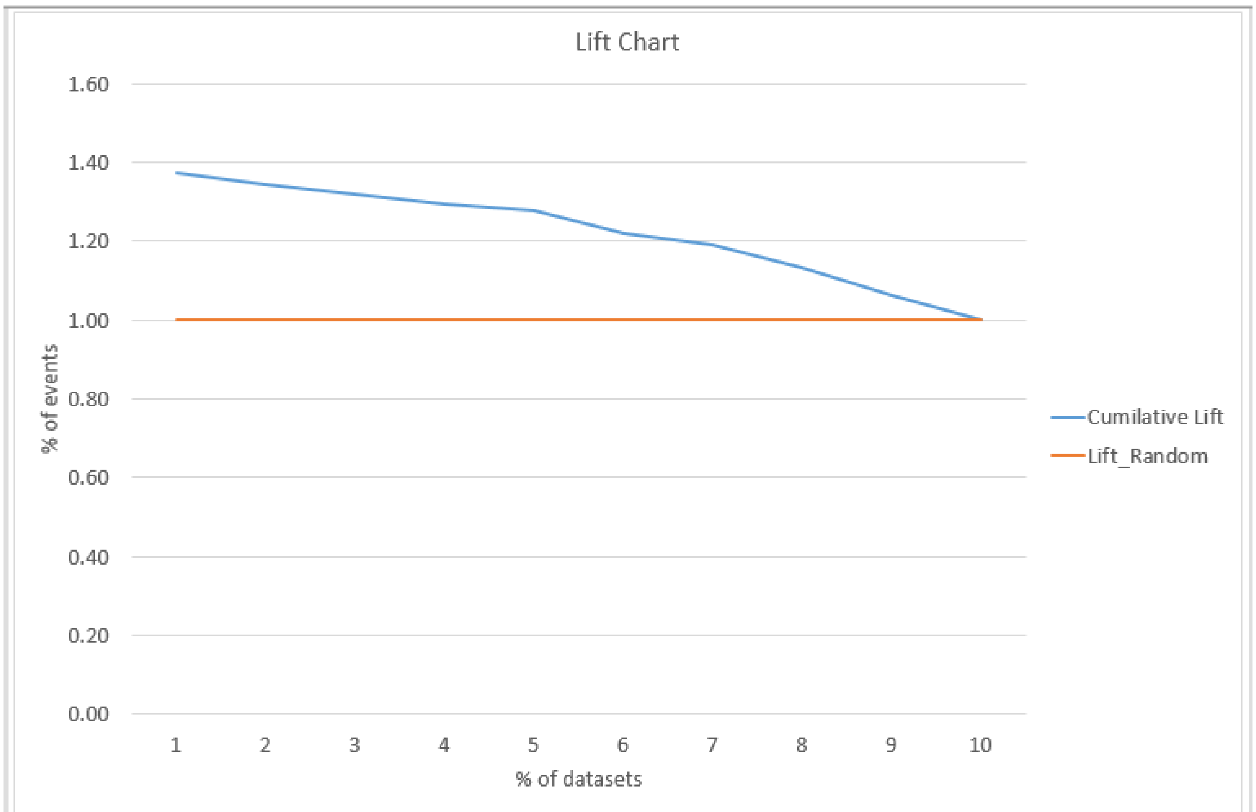


Figure 2: Lift Chart