Logistic/Lasso/Ridge/Decision Tree

Team 03

26/02/2021

Setup

```
library(data.table)
library(ggplot2)
library(ggthemes)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1
```

```
library(caret)
```

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

```
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

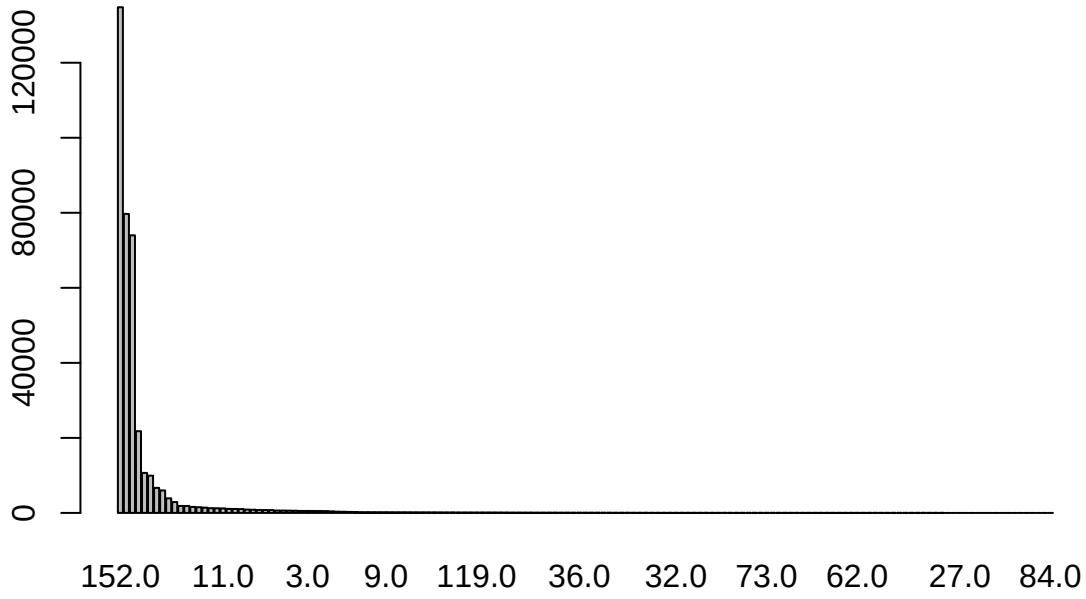
```
theme_set(theme_bw())
```

Load the dataset

```
## read into the data set
db <- fread("train.csv",
            colClasses = c(Driving_License='character', Region_Code='character',
                           Previously_Insured='character', Policy_Sales_Channel='character', Response='
            stringsAsFactors = T)

# Dropping the 'id' column and storing it into a new data table
dbm <- db[,-"id"]
str(dbm)
```

```
## Classes 'data.table' and 'data.frame':   381109 obs. of  11 variables:
##  $ Gender              : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 2 1 1 1 ...
##  $ Age                 : int  44 76 47 21 29 24 23 56 24 32 ...
##  $ Driving_License     : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Region_Code         : Factor w/ 53 levels "0.0","1.0","10.0",..: 22 24 22 4 37 28 4 22 24 50 ...
##  $ Previously_Insured  : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 1 1 2 2 ...
##  $ Vehicle_Age         : Factor w/ 3 levels "1-2 Year","< 1 Year",..: 3 1 3 2 2 2 2 1 2 2 ...
##  $ Vehicle_Damage      : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 2 2 2 1 1 ...
##  $ Annual_Premium      : num  40454 33536 38294 28619 27496 ...
##  $ Policy_Sales_Channel: Factor w/ 155 levels "1.0","10.0","100.0",..: 79 79 79 58 58 67 58 79 58 58
##  $ Vintage             : int  217 183 27 203 39 176 249 72 28 80 ...
##  $ Response            : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 2 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
# with this we have a dataset of 11 columns (1 response and 10 predictors) and 381109 rows
# Our numeric predictors are: Age, Annual_Premium, Vintage
# Our categorical predictors are: Gender, Driving_License, Region_Code, Previously_Insured, Vehicle_Age


#merge some levels of Policy_Sales_Channel, because many levels just have less than 10 observations
dbm$Policy_Sales_Channel <- as.character(dbm$Policy_Sales_Channel) #convert into character
n.policy <- sort(table(dbm$Policy_Sales_Channel), decreasing = T)  #counts for policy sales channels
barplot(n.policy)     #show the count for each channel
```



```r
sum(n.policy[1:9]) / nrow(dbm)   #the first 9 levels take account over 91%
```

```
## [1] 0.9115791
```

```r
dbm$Policy_Sales_Channel[!(dbm$Policy_Sales_Channel %in% names(n.policy)[1:9])] <- "other"
dbm$Policy_Sales_Channel <- as.factor(dbm$Policy_Sales_Channel)  #convert back into factor
```

Splitting the dataset into training (80%) and test (20%)

```r
set.seed(810)
i.train <- sample(nrow(dbm), round(0.8*nrow(dbm)))  #index of rows for training

#split for Logistic regression
dbm.train <- dbm[i.train, ]
dbm.test <- dbm[-i.train, ]


#split for Lasso and Ridge
x_data <- model.matrix( ~ -1 + Gender + Age + Driving_License + Region_Code + Previously_Insured +
                          Vehicle_Age + Vehicle_Damage + Annual_Premium + Policy_Sales_Channel + Vintage
# outcome is Response
y_data <- as.numeric(as.character(db$Response))
x_train <- x_data[i.train, ]
y_train <- y_data[i.train]
x_test <- x_data[-i.train, ]
y_test <- y_data[-i.train]
```

Run Logistic regression

```r
mylogit <- glm(Response ~., data = dbm.train, family = "binomial")
summary(mylogit)  #result of logistic regression
```

```
##
## Call:
## glm(formula = Response ~ ., family = "binomial", data = dbm.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3339  -0.6214  -0.0484  -0.0284   4.1877
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -3.748e+00  2.051e-01 -18.271  < 2e-16 ***
## GenderMale                 6.780e-02  1.262e-02   5.373 7.72e-08 ***
## Age                       -2.604e-02  6.138e-04 -42.427  < 2e-16 ***
## Driving_License1           1.065e+00  1.733e-01   6.147 7.88e-10 ***
## Region_Code1.0             3.026e-01  1.490e-01   2.030 0.042323 *
## Region_Code10.0            7.489e-01  1.210e-01   6.189 6.04e-10 ***
```

```
## Region_Code11.0            1.148e+00  1.023e-01  11.223  < 2e-16 ***
## Region_Code12.0            6.829e-01  1.202e-01   5.682 1.33e-08 ***
## Region_Code13.0            6.814e-01  1.149e-01   5.929 3.04e-09 ***
## Region_Code14.0            8.227e-01  1.134e-01   7.257 3.97e-13 ***
## Region_Code15.0            5.959e-01  1.015e-01   5.869 4.40e-09 ***
## Region_Code16.0            5.748e-01  1.440e-01   3.992 6.56e-05 ***
## Region_Code17.0            5.436e-01  1.305e-01   4.165 3.11e-05 ***
## Region_Code18.0            1.079e+00  1.069e-01  10.095  < 2e-16 ***
## Region_Code19.0            8.042e-01  1.252e-01   6.425 1.32e-10 ***
## Region_Code2.0             6.034e-01  1.185e-01   5.092 3.55e-07 ***
## Region_Code20.0            3.130e-01  1.245e-01   2.514 0.011947 *
## Region_Code21.0            9.369e-01  1.198e-01   7.822 5.19e-15 ***
## Region_Code22.0            2.672e-01  1.694e-01   1.577 0.114738
## Region_Code23.0            1.034e+00  1.207e-01   8.565  < 2e-16 ***
## Region_Code24.0            8.231e-01  1.178e-01   6.985 2.85e-12 ***
## Region_Code25.0            2.747e-01  1.489e-01   1.845 0.065012 .
## Region_Code26.0            2.580e-01  1.294e-01   1.994 0.046136 *
## Region_Code27.0            5.566e-01  1.279e-01   4.351 1.36e-05 ***
## Region_Code28.0            8.602e-01  9.371e-02   9.180  < 2e-16 ***
## Region_Code29.0            1.080e+00  1.004e-01  10.762  < 2e-16 ***
## Region_Code3.0             9.791e-01  1.008e-01   9.709  < 2e-16 ***
## Region_Code30.0            9.443e-01  1.030e-01   9.164  < 2e-16 ***
## Region_Code31.0            3.745e-01  1.284e-01   2.916 0.003543 **
## Region_Code32.0            8.479e-01  1.246e-01   6.803 1.03e-11 ***
## Region_Code33.0            7.355e-01  1.043e-01   7.050 1.78e-12 ***
## Region_Code34.0            4.206e-01  1.435e-01   2.932 0.003370 **
## Region_Code35.0            1.138e+00  1.037e-01  10.974  < 2e-16 ***
## Region_Code36.0            7.107e-01  1.047e-01   6.791 1.11e-11 ***
## Region_Code37.0            5.884e-01  1.115e-01   5.276 1.32e-07 ***
## Region_Code38.0            9.641e-01  1.153e-01   8.364  < 2e-16 ***
## Region_Code39.0            6.396e-01  1.074e-01   5.958 2.56e-09 ***
## Region_Code4.0             9.370e-01  1.224e-01   7.653 1.96e-14 ***
## Region_Code40.0            6.259e-01  1.394e-01   4.491 7.09e-06 ***
## Region_Code41.0            1.062e+00  9.741e-02  10.899  < 2e-16 ***
## Region_Code42.0            3.080e-01  2.113e-01   1.458 0.144860
## Region_Code43.0            4.302e-01  1.233e-01   3.488 0.000486 ***
## Region_Code44.0            1.223e-01  2.316e-01   0.528 0.597526
## Region_Code45.0            7.959e-01  1.076e-01   7.398 1.39e-13 ***
## Region_Code46.0            7.513e-01  9.728e-02   7.723 1.14e-14 ***
## Region_Code47.0            5.183e-01  1.052e-01   4.928 8.30e-07 ***
## Region_Code48.0            2.523e-01  1.082e-01   2.332 0.019710 *
## Region_Code49.0            4.564e-01  1.408e-01   3.242 0.001187 **
## Region_Code5.0             7.936e-01  1.411e-01   5.623 1.88e-08 ***
## Region_Code50.0            3.538e-01  1.050e-01   3.371 0.000749 ***
## Region_Code51.0            8.868e-01  2.651e-01   3.345 0.000821 ***
## Region_Code52.0            7.364e-01  2.478e-01   2.972 0.002962 **
## Region_Code6.0             9.570e-01  1.124e-01   8.517  < 2e-16 ***
## Region_Code7.0             7.120e-01  1.122e-01   6.348 2.18e-10 ***
## Region_Code8.0             6.442e-01  9.588e-02   6.719 1.84e-11 ***
## Region_Code9.0             4.874e-01  1.211e-01   4.023 5.74e-05 ***
## Previously_Insured1       -3.887e+00  9.061e-02 -42.902  < 2e-16 ***
## Vehicle_Age< 1 Year       -3.684e-01  2.656e-02 -13.872  < 2e-16 ***
## Vehicle_Age> 2 Years       2.178e-01  2.151e-02  10.126  < 2e-16 ***
## Vehicle_DamageYes          2.000e+00  3.864e-02  51.755  < 2e-16 ***
```

```
## Annual_Premium             7.352e-07  3.803e-07   1.933 0.053211 .
## Policy_Sales_Channel124.0  8.151e-02  3.414e-02   2.388 0.016957 *
## Policy_Sales_Channel151.0 -1.162e+00  1.141e-01 -10.186  < 2e-16 ***
## Policy_Sales_Channel152.0 -9.932e-01  4.376e-02 -22.696  < 2e-16 ***
## Policy_Sales_Channel154.0  1.165e-01  4.830e-02   2.413 0.015826 *
## Policy_Sales_Channel156.0 -9.948e-02  4.378e-02  -2.272 0.023062 *
## Policy_Sales_Channel157.0  1.521e-01  4.673e-02   3.255 0.001135 **
## Policy_Sales_Channel160.0 -1.953e+00  6.635e-02 -29.440  < 2e-16 ***
## Policy_Sales_Channel26.0   2.757e-01  3.359e-02   8.208 2.26e-16 ***
## Policy_Sales_Channelother  3.809e-02  3.693e-02   1.031 0.302430
## Vintage                   -1.293e-05  7.308e-05  -0.177 0.859547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 227152  on 304886  degrees of freedom
## Residual deviance: 164084  on 304816  degrees of freedom
## AIC: 164226
##
## Number of Fisher Scoring iterations: 9
```

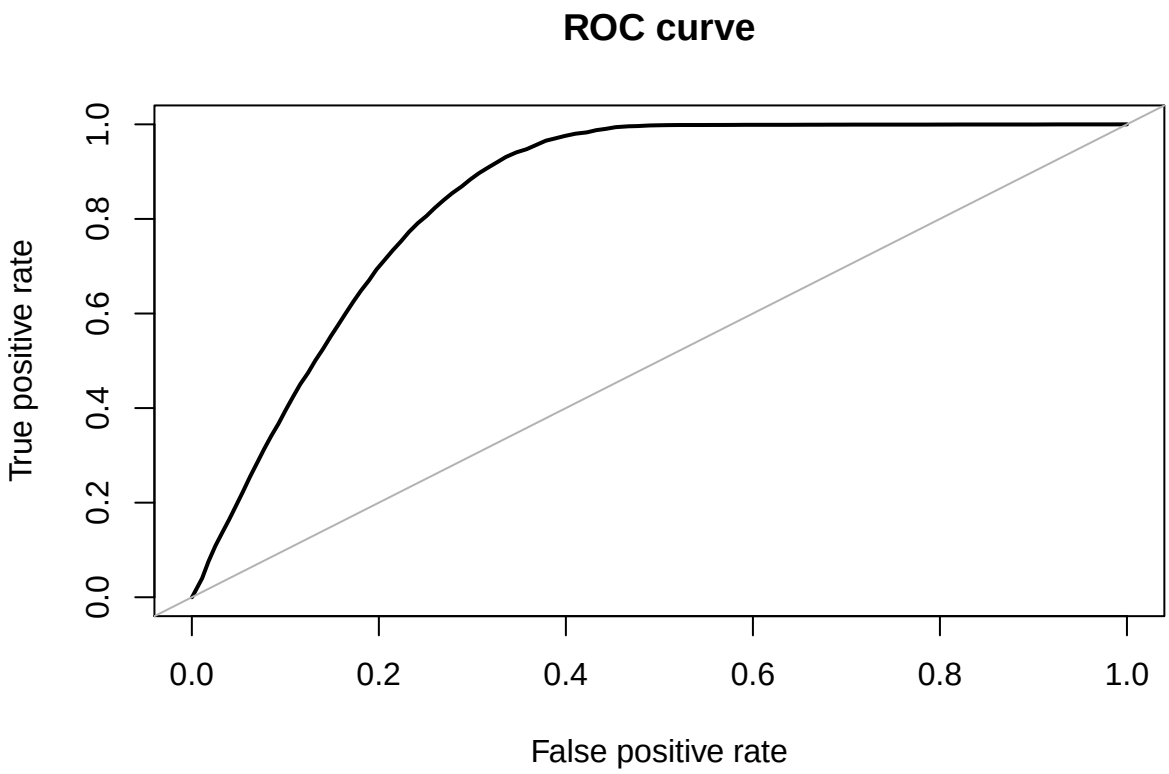
```
pred.lgt <- predict(mylogit, newdata=dbm.test, type="response")
yhat <- as.factor(ifelse(pred.lgt<0.5, 0, 1))  #get predicted response
CM.logit <- confusionMatrix(yhat, as.factor(y_test), positive ='1')     #create confusion matrix
CM.logit$table     #show the matrix
```

```
##           Reference
## Prediction     0     1
##          0 66929  9256
##          1    31     6
```

```
CM.logit$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   0.8781585369   0.0003237891   0.8758154160   0.8804728320   0.8784865262
## AccuracyPValue  McnemarPValue
##   0.6117881206   0.0000000000
```

```
mse_logistic_test <- mean((y_test - pred.lgt)^2)
roc.curve(as.numeric(y_test),as.numeric(pred.lgt),plotit = TRUE)
```




```
## Area under the curve (AUC): 0.848
```

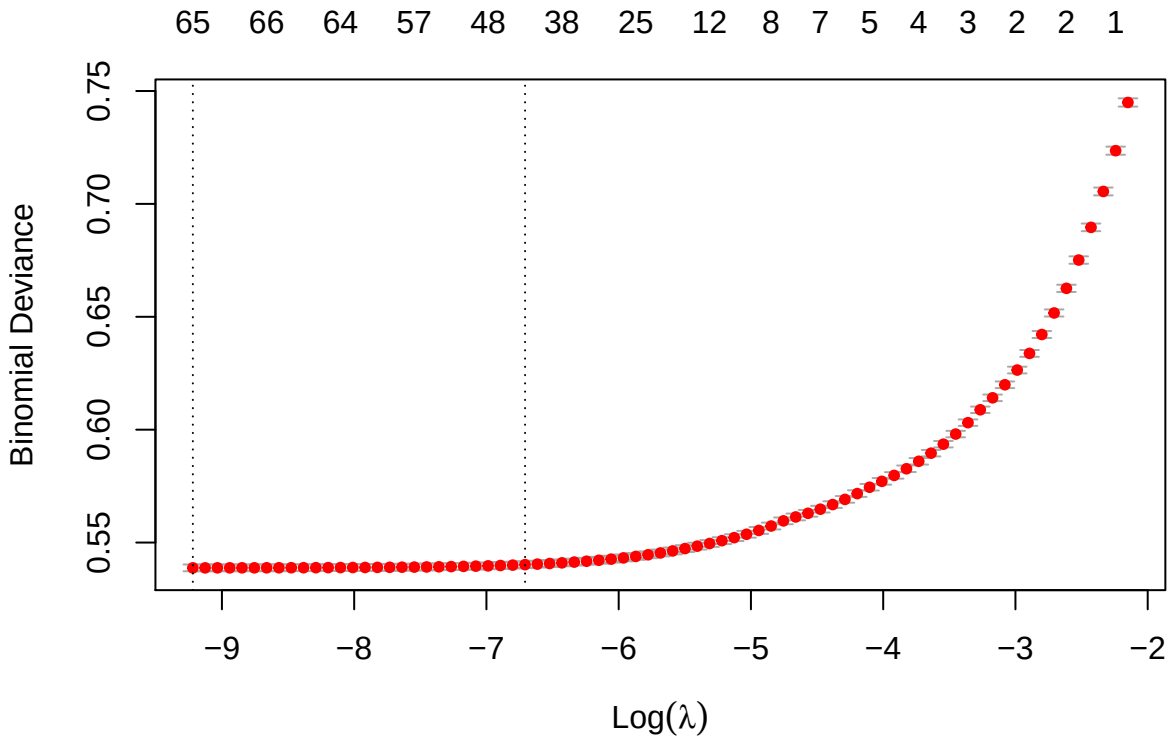
Run Lasso regressions

```
#Create formula
formula <- as.formula(Response ~ .)

#Set model and fit
train.matrix <- model.matrix(formula,dbm.train)[,-1]
laso.fit <- cv.glmnet(train.matrix,dbm.train$Response, family="binomial",alpha=1, nfolds = 10)
test.matrix <- model.matrix(formula, dbm.test)[,-1]

#predict test dataset
laso.pred <- predict(laso.fit, test.matrix, lambda = cv.laso.fit$lambda.min, type="response")
laso.yhat <- ifelse(laso.pred > 0.5, 1, 0)

confusionMatrix(as.factor(laso.yhat),as.factor(y_test),positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 66959  9262
```

```
##          1     1     0
##
##                Accuracy : 0.8785
##                  95% CI : (0.8761, 0.8808)
##     No Information Rate : 0.8785
##     P-Value [Acc > NIR] : 0.5072
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.000e+00
##             Specificity : 1.000e+00
##          Pos Pred Value : 0.000e+00
##          Neg Pred Value : 8.785e-01
##              Prevalence : 1.215e-01
##          Detection Rate : 0.000e+00
##    Detection Prevalence : 1.312e-05
##       Balanced Accuracy : 5.000e-01
##
##        'Positive' Class : 1
##
```

```
mse.laso <- mean((y_test - laso.pred)^2)

##plot
plot(laso.fit)
```




```
### Extract the results of the best fitting model

print(laso.fit$lambda.min)
```

```
## [1] 9.905024e-05
```

```
coef(laso.fit, s=laso.fit$lambda.min)
```

```
## 71 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)                -3.135348e+00
## GenderMale                  6.547266e-02
## Age                        -2.569465e-02
## Driving_License1            1.027077e+00
## Region_Code1.0             -2.618724e-01
## Region_Code10.0             1.337345e-01
## Region_Code11.0             5.459761e-01
## Region_Code12.0             7.381981e-02
## Region_Code13.0             7.301550e-02
## Region_Code14.0             2.150993e-01
## Region_Code15.0             .
## Region_Code16.0             .
## Region_Code17.0            -3.053628e-02
## Region_Code18.0             4.757798e-01
## Region_Code19.0             1.991334e-01
```

```
## Region_Code2.0             .
## Region_Code20.0           -2.567313e-01
## Region_Code21.0            3.235748e-01
## Region_Code22.0           -2.921070e-01
## Region_Code23.0            4.293598e-01
## Region_Code24.0            2.172950e-01
## Region_Code25.0           -2.881037e-01
## Region_Code26.0           -3.164213e-01
## Region_Code27.0           -2.028761e-02
## Region_Code28.0            2.658891e-01
## Region_Code29.0            4.798019e-01
## Region_Code3.0             3.792497e-01
## Region_Code30.0            3.394594e-01
## Region_Code31.0           -1.926629e-01
## Region_Code32.0            2.344637e-01
## Region_Code33.0            1.336100e-01
## Region_Code34.0           -1.505638e-01
## Region_Code35.0            5.362759e-01
## Region_Code36.0            1.047492e-01
## Region_Code37.0            .
## Region_Code38.0            3.621398e-01
## Region_Code39.0            3.798660e-02
## Region_Code4.0             3.321287e-01
## Region_Code40.0            1.405962e-02
## Region_Code41.0            4.630521e-01
## Region_Code42.0           -2.436108e-01
## Region_Code43.0           -1.460933e-01
## Region_Code44.0           -4.117773e-01
## Region_Code45.0            1.923038e-01
## Region_Code46.0            1.535043e-01
## Region_Code47.0           -6.612223e-02
## Region_Code48.0           -3.205063e-01
## Region_Code49.0           -1.160337e-01
## Region_Code5.0             1.799811e-01
## Region_Code50.0           -2.277680e-01
## Region_Code51.0            2.548082e-01
## Region_Code52.0            1.011759e-01
## Region_Code6.0             3.474683e-01
## Region_Code7.0             1.092050e-01
## Region_Code8.0             4.558589e-02
## Region_Code9.0            -9.007445e-02
## Previously_Insured1       -3.743450e+00
## Vehicle_Age< 1 Year       -3.628078e-01
## Vehicle_Age> 2 Years       2.127958e-01
## Vehicle_DamageYes          1.988646e+00
## Annual_Premium             1.016414e-06
## Policy_Sales_Channel124.0  8.404407e-02
## Policy_Sales_Channel151.0 -1.123362e+00
## Policy_Sales_Channel152.0 -9.803189e-01
## Policy_Sales_Channel154.0  1.207643e-01
## Policy_Sales_Channel156.0 -8.133376e-02
## Policy_Sales_Channel157.0  1.615419e-01
## Policy_Sales_Channel160.0 -1.923097e+00
## Policy_Sales_Channel26.0   2.733945e-01
```

```
## Policy_Sales_Channelother  3.486025e-02
## Vintage                    .
```

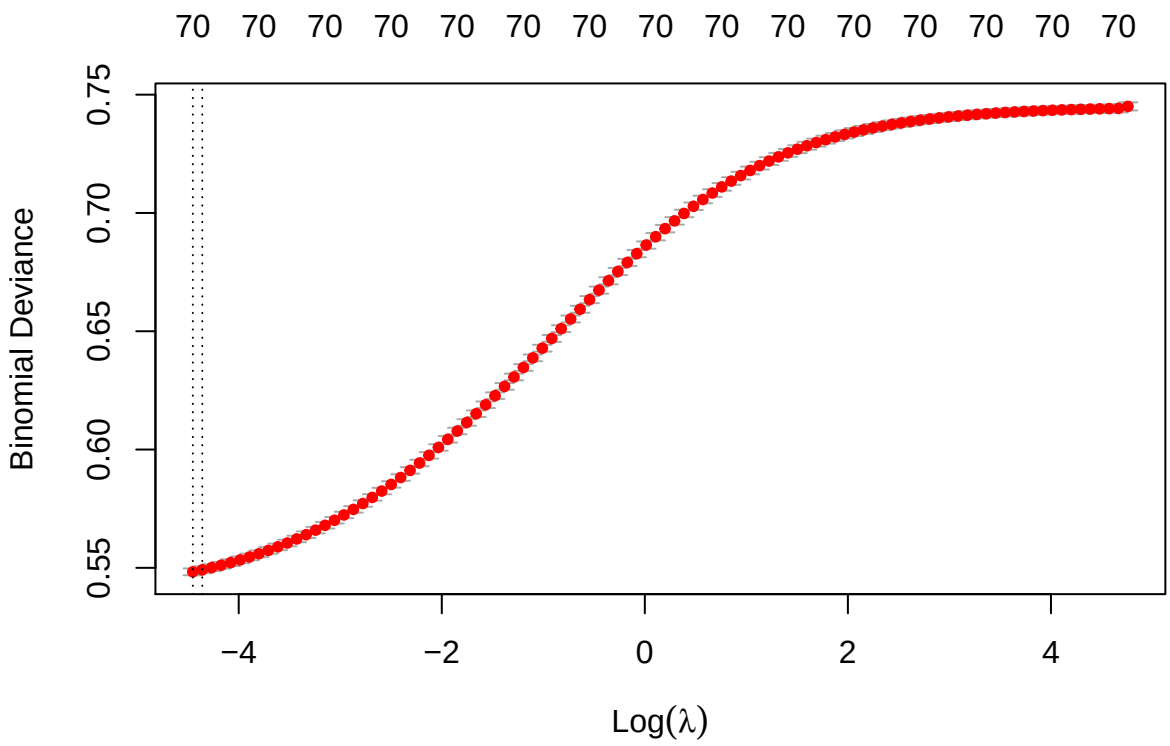
Run Ridge regressions

```
#Model and fit
Ridge.fit <- cv.glmnet(train.matrix,dbm.train$Response, family="binomial",alpha=0, nfolds = 10)

#Predict
Ridge.pred <- predict(Ridge.fit, test.matrix, lambda = cv.Ridge.fit$lambda.min, type="response")
Ridge.yhat <- ifelse(Ridge.pred > 0.5, 1, 0)
confusionMatrix(as.factor(Ridge.yhat),as.factor(y_test),positive = "1")
```

```
## Warning in confusionMatrix.default(as.factor(Ridge.yhat), as.factor(y_test), :
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 66960  9262
##          1     0     0
##
##                Accuracy : 0.8785
##                  95% CI : (0.8761, 0.8808)
##     No Information Rate : 0.8785
##     P-Value [Acc > NIR] : 0.5028
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.8785
##              Prevalence : 0.1215
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 1
##
```

```
##plot
plot(Ridge.fit)
```




```
### Extract the results of the best fitting model


mse.Ridge <- mean((y_test - Ridge.pred)^2)
coef(Ridge.fit, s=Ridge.fit$lambda.min)
```

```
## 71 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)               -3.130003e+00
## GenderMale                 6.560309e-02
## Age                       -1.739822e-02
## Driving_License1           9.482205e-01
## Region_Code1.0            -3.547485e-01
## Region_Code10.0           -2.775119e-02
## Region_Code11.0            3.256565e-01
## Region_Code12.0           -5.761012e-02
## Region_Code13.0           -4.784353e-02
## Region_Code14.0            5.191794e-02
## Region_Code15.0           -1.452122e-01
## Region_Code16.0           -1.631225e-01
## Region_Code17.0           -1.817913e-01
## Region_Code18.0            2.901575e-01
## Region_Code19.0            8.815432e-02
## Region_Code2.0            -1.333373e-01
## Region_Code20.0           -3.484121e-01
## Region_Code21.0            1.132501e-01
```

```
## Region_Code22.0           -4.060388e-01
## Region_Code23.0            2.805808e-01
## Region_Code24.0            8.431498e-02
## Region_Code25.0           -3.972218e-01
## Region_Code26.0           -4.037683e-01
## Region_Code27.0           -1.777588e-01
## Region_Code28.0            1.546660e-01
## Region_Code29.0            2.794427e-01
## Region_Code3.0             2.004810e-01
## Region_Code30.0            1.220783e-01
## Region_Code31.0           -2.972460e-01
## Region_Code32.0            6.662093e-02
## Region_Code33.0           -7.826677e-03
## Region_Code34.0           -2.676859e-01
## Region_Code35.0            3.277767e-01
## Region_Code36.0           -4.685035e-02
## Region_Code37.0           -1.509230e-01
## Region_Code38.0            2.156497e-01
## Region_Code39.0           -6.794617e-02
## Region_Code4.0             1.972911e-01
## Region_Code40.0           -7.746415e-02
## Region_Code41.0            2.711811e-01
## Region_Code42.0           -3.578311e-01
## Region_Code43.0           -2.690488e-01
## Region_Code44.0           -4.914009e-01
## Region_Code45.0            4.854203e-02
## Region_Code46.0            3.209103e-03
## Region_Code47.0           -1.875395e-01
## Region_Code48.0           -4.049002e-01
## Region_Code49.0           -2.432066e-01
## Region_Code5.0             5.118218e-02
## Region_Code50.0           -3.376885e-01
## Region_Code51.0            1.437951e-01
## Region_Code52.0           -3.196482e-02
## Region_Code6.0             1.337330e-01
## Region_Code7.0            -1.506530e-02
## Region_Code8.0            -9.459728e-02
## Region_Code9.0            -2.155181e-01
## Previously_Insured1       -1.629958e+00
## Vehicle_Age< 1 Year       -3.699739e-01
## Vehicle_Age> 2 Years       2.196081e-01
## Vehicle_DamageYes          1.551165e+00
## Annual_Premium             1.113400e-06
## Policy_Sales_Channel124.0  2.901058e-01
## Policy_Sales_Channel151.0 -6.146252e-01
## Policy_Sales_Channel152.0 -5.699607e-01
## Policy_Sales_Channel154.0  3.848329e-01
## Policy_Sales_Channel156.0  2.022363e-01
## Policy_Sales_Channel157.0  4.192303e-01
## Policy_Sales_Channel160.0 -1.095435e+00
## Policy_Sales_Channel26.0   4.236164e-01
## Policy_Sales_Channelother  2.049714e-01
## Vintage                   -1.274543e-05
```

```r
### Linear Regression
fit.lm <- lm(Response ~.,dbm.train)
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a factor
## response will be ignored
```

```
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors
```

```r
lm.pred <- predict(fit.lm,dbm.test,type="response")

mse.lm <- mean((y_test - lm.pred)^2)


coef(fit.lm)
```

```
##               (Intercept)                GenderMale                       Age
##              1.085910e+00              4.428507e-03             -3.124293e-03
##          Driving_License1             Region_Code1.0            Region_Code10.0
##              8.516848e-02              2.837304e-02              8.311343e-02
##           Region_Code11.0            Region_Code12.0            Region_Code13.0
##              1.133781e-01              7.470242e-02              6.865740e-02
##           Region_Code14.0            Region_Code15.0            Region_Code16.0
##              8.787627e-02              5.993741e-02              6.332897e-02
##           Region_Code17.0            Region_Code18.0            Region_Code19.0
##              6.715973e-02              1.094379e-01              8.529825e-02
##            Region_Code2.0            Region_Code20.0            Region_Code21.0
##              6.017286e-02              3.065727e-02              9.572760e-02
##           Region_Code22.0            Region_Code23.0            Region_Code24.0
##              4.519655e-02              1.092959e-01              8.102977e-02
##           Region_Code25.0            Region_Code26.0            Region_Code27.0
##              5.504651e-02              3.225235e-02              4.952676e-02
##           Region_Code28.0            Region_Code29.0             Region_Code3.0
##              9.283636e-02              1.096433e-01              9.613053e-02
##           Region_Code30.0            Region_Code31.0            Region_Code32.0
##              9.523535e-02              4.193380e-02              7.478145e-02
##           Region_Code33.0            Region_Code34.0            Region_Code35.0
##              8.188436e-02              3.996507e-02              1.137208e-01
##           Region_Code36.0            Region_Code37.0            Region_Code38.0
##              6.848107e-02              5.231059e-02              1.010579e-01
##           Region_Code39.0             Region_Code4.0            Region_Code40.0
##              6.642371e-02              9.954399e-02              5.601336e-02
##           Region_Code41.0            Region_Code42.0            Region_Code43.0
##              1.065278e-01              4.370475e-02              4.822764e-02
##           Region_Code44.0            Region_Code45.0            Region_Code46.0
##              4.960179e-02              7.935859e-02              8.018759e-02
##           Region_Code47.0            Region_Code48.0            Region_Code49.0
##              5.034694e-02              2.681628e-02              4.258738e-02
##            Region_Code5.0            Region_Code50.0            Region_Code51.0
##              6.678101e-02              5.442033e-02              9.109484e-02
##           Region_Code52.0             Region_Code6.0             Region_Code7.0
##              6.841885e-02              9.738988e-02              7.190784e-02
##            Region_Code8.0             Region_Code9.0      Previously_Insured1
```

```
##              6.900863e-02              5.462223e-02             -8.742157e-02
##       Vehicle_Age< 1 Year      Vehicle_Age> 2 Years         Vehicle_DamageYes
##             -5.124922e-02              5.619110e-02              1.277929e-01
##            Annual_Premium Policy_Sales_Channel124.0 Policy_Sales_Channel151.0
##              1.472973e-07              1.229416e-02             -7.305550e-02
## Policy_Sales_Channel152.0 Policy_Sales_Channel154.0 Policy_Sales_Channel156.0
##             -6.490761e-02              2.688640e-02             -9.094533e-03
## Policy_Sales_Channel157.0 Policy_Sales_Channel160.0  Policy_Sales_Channel26.0
##              3.589837e-02             -1.155436e-01              3.475024e-02
## Policy_Sales_Channelother                   Vintage
##              3.377273e-03             -1.029350e-06
```

Compare these three models

```
#Compare MSE and select the best one
c(MSE_Logistic=mse_logistic_test, MSE_Lasso=mse.laso, MSE_Ridge=mse.Ridge, MSE_Linear=mse.lm)
```

```
## MSE_Logistic    MSE_Lasso    MSE_Ridge   MSE_Linear
##   0.08760303   0.08771065   0.08803651   1.09376311
```

```
### Decision Tree
library(rpart)

ctrl <- trainControl(method='cv', number = 5)
dt <- train(Response ~ ., data = dbm.train,
            method = 'rpart',
            trControl = ctrl)


tree.pred = predict(dt, dbm.test)
confusionMatrix(tree.pred,as.factor(y_test), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 66960  9262
##          1     0     0
##
##                Accuracy : 0.8785
##                  95% CI : (0.8761, 0.8808)
##     No Information Rate : 0.8785
##     P-Value [Acc > NIR] : 0.5028
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
```

```
##          Neg Pred Value : 0.8785
##              Prevalence : 0.1215
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 1
##
```

```
mse_tree <- (mean(y_test - as.numeric(tree.pred))^2)
```