Uso del paquete CARET

```
[ ]
#Loading caret package
library("caret")

#Loading training data
train<-read.csv("train_u6lujuX_CVtuZ9i.csv",stringsAsFactors = T)

#Looking at the structure of caret package.
str(train)
#'data.frame':         614 obs. of  13 variables:
#$ Loan_ID          : Factor w/ 614 levels "LP001002","LP001003",..: 1 2
#$ Gender           : Factor w/ 3 levels "","Female","Male": 3 3 3 3 3 3
#$ Married          : Factor w/ 3 levels "","No","Yes": 2 3 3 3 2 3 3 3 3
#$ Dependents       : Factor w/ 5 levels "","0","1","2",..: 2 3 2 2 2 4 2
#$ Education        : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 1
#$ Self_Employed    : Factor w/ 3 levels "","No","Yes": 2 2 3 2 2 3 2 2 2
#$ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 1
#$ CoapplicantIncome: num  0 1508 0 2358 0 ...
#$ LoanAmount       : int  NA 128 66 120 141 267 95 158 168 349 ...
#$ Loan_Amount_Term : int  360 360 360 360 360 360 360 360 360 360 ...
#$ Credit_History   : int  1 1 1 1 1 1 1 0 1 1 ...
#$ Property_Area    : Factor w/ 3 levels "Rural","Semiurban",..: 3 1 3 3
#$ Loan_Status      : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 1 2 1 ...
```

```
[ ]
#Imputing missing values using KNN.Also centering and scaling numerical c
preProcValues <- preProcess(train, method = c("knnImpute","center","scale

library('RANN')
train_processed <- predict(preProcValues, train)
sum(is.na(train_processed))
#[1] 0
```

```
[ ]
#Converting outcome variable to numeric
train_processed$Loan_Status<-ifelse(train_processed$Loan_Status=='N',0,1)

id<-train_processed$Loan_ID
train_processed$Loan_ID<-NULL

#Checking the structure of processed train file
str(train_processed)
#'data.frame':         614 obs. of  12 variables:
#$ Gender           : Factor w/ 3 levels "","Female","Male": 3 3 3 3 3 3
#$ Married          : Factor w/ 3 levels "","No","Yes": 2 3 3 3 2 3 3 3 3
```

```
#$ Dependents       : Factor w/ 5 levels "","0","1","2",..: 2 3 2 2 2 4 2
#$ Education        : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 1
#$ Self_Employed    : Factor w/ 3 levels "","No","Yes": 2 2 3 2 2 3 2 2 2
#$ ApplicantIncome  : num  0.0729 -0.1343 -0.3934 -0.4617 0.0976 ...
#$ CoapplicantIncome: num  -0.554 -0.0387 -0.554 0.2518 -0.554 ...
#$ LoanAmount       : num  0.0162 -0.2151 -0.9395 -0.3086 -0.0632 ...
#$ Loan_Amount_Term : num  0.276 0.276 0.276 0.276 0.276 ...
#$ Credit_History   : num  0.432 0.432 0.432 0.432 0.432 ...
#$ Property_Area    : Factor w/ 3 levels "Rural","Semiurban",..: 3 1 3 3
#$ Loan_Status      : num  1 0 1 1 1 1 1 0 1 0 ...
```

```
[ ]
#Converting every categorical variable to numerical using dummy variables
dmy <- dummyVars(" ~ .", data = train_processed,fullRank = T)
train_transformed <- data.frame(predict(dmy, newdata = train_processed))

#Checking the structure of transformed train file
str(train_transformed)
#'data.frame':        614 obs. of  19 variables:
#$ Gender.Female           : num  0 0 0 0 0 0 0 0 0 0 ...
#$ Gender.Male             : num  1 1 1 1 1 1 1 1 1 1 ...
#$ Married.No              : num  1 0 0 0 1 0 0 0 0 0 ...
#$ Married.Yes             : num  0 1 1 1 0 1 1 1 1 1 ...
#$ Dependents.0            : num  1 0 1 1 1 0 1 0 0 0 ...
#$ Dependents.1            : num  0 1 0 0 0 0 0 0 0 1 ...
#$ Dependents.2            : num  0 0 0 0 0 1 0 0 1 0 ...
#$ Dependents.3.           : num  0 0 0 0 0 0 0 1 0 0 ...
#$ Education.Not.Graduate : num  0 0 0 1 0 0 1 0 0 0 ...
#$ Self_Employed.No        : num  1 1 0 1 1 0 1 1 1 1 ...
#$ Self_Employed.Yes       : num  0 0 1 0 0 1 0 0 0 0 ...
#$ ApplicantIncome         : num  0.0729 -0.1343 -0.3934 -0.4617 0.0976 ..
#$ CoapplicantIncome       : num  -0.554 -0.0387 -0.554 0.2518 -0.554 ...
#$ LoanAmount              : num  0.0162 -0.2151 -0.9395 -0.3086 -0.0632 .
#$ Loan_Amount_Term        : num  0.276 0.276 0.276 0.276 0.276 ...
#$ Credit_History          : num  0.432 0.432 0.432 0.432 0.432 ...
#$ Property_Area.Semiurban: num  0 0 0 0 0 0 0 1 0 1 ...
#$ Property_Area.Urban     : num  1 0 1 1 1 1 1 0 1 0 ...
#$ Loan_Status             : num  1 0 1 1 1 1 1 0 1 0 ...

#Converting the dependent variable back to categorical
train_transformed$Loan_Status<-as.factor(train_transformed$Loan_Status)
```

```
[ ]
#Spliting training set into two parts based on outcome: 75% and 25%
index <- createDataPartition(train_transformed$Loan_Status, p=0.75, list=
trainSet <- train_transformed[ index,]
testSet <- train_transformed[-index,]

#Checking the structure of trainSet
str(trainSet)
#'data.frame':        461 obs. of  19 variables:
```

```
#$ Gender.Female          : num  0 0 0 0 0 0 0 0 0 0 ...
#$ Gender.Male            : num  1 1 1 1 1 1 1 1 1 1 ...
#$ Married.No             : num  1 0 0 0 1 0 0 0 0 0 ...
#$ Married.Yes            : num  0 1 1 1 0 1 1 1 1 1 ...
#$ Dependents.0           : num  1 0 1 1 1 0 1 0 0 0 ...
#$ Dependents.1           : num  0 1 0 0 0 0 0 0 1 0 ...
#$ Dependents.2           : num  0 0 0 0 0 1 0 0 0 1 ...
#$ Dependents.3.          : num  0 0 0 0 0 0 0 1 0 0 ...
#$ Education.Not.Graduate : num  0 0 0 1 0 0 1 0 0 0 ...
#$ Self_Employed.No       : num  1 1 0 1 1 0 1 1 1 1 ...
#$ Self_Employed.Yes      : num  0 0 1 0 0 1 0 0 0 0 ...
#$ ApplicantIncome        : num  0.0729 -0.1343 -0.3934 -0.4617 0.0976 ..
#$ CoapplicantIncome      : num  -0.554 -0.0387 -0.554 0.2518 -0.554 ...
#$ LoanAmount             : num  0.0162 -0.2151 -0.9395 -0.3086 -0.0632 .
#$ Loan_Amount_Term       : num  0.276 0.276 0.276 0.276 0.276 ...
#$ Credit_History         : num  0.432 0.432 0.432 0.432 0.432 ...
#$ Property_Area.Semiurban: num  0 0 0 0 0 0 0 1 1 0 ...
#$ Property_Area.Urban    : num  1 0 1 1 1 1 1 0 0 1 ...
#$ Loan_Status            : Factor w/ 2 levels "0","1": 2 1 2 2 2 2 2 1 1
```

[ ]

```
#Feature selection using rfe in caret
control <- rfeControl(functions = rfFuncs,
                      method = "repeatedcv",
                      repeats = 3,
                      verbose = FALSE)
outcomeName<-'Loan_Status'
predictors<-names(trainSet)[!names(trainSet) %in% outcomeName]
Loan_Pred_Profile <- rfe(trainSet[,predictors], trainSet[,outcomeName],
                         rfeControl = control)
Loan_Pred_Profile
#Recursive feature selection
#Outer resampling method: Cross-Validated (10 fold, repeated 3 times)
#Resampling performance over subset size:
#  Variables Accuracy  Kappa AccuracySD KappaSD Selected
#4   0.7737 0.4127    0.03707 0.09962
#8   0.7874 0.4317    0.03833 0.11168
#16   0.7903 0.4527    0.04159 0.11526
#18   0.7882 0.4431    0.03615 0.10812
#The top 5 variables (out of 16):
#  Credit_History, LoanAmount, Loan_Amount_Term, ApplicantIncome, Coappli
#Taking only the top 5 predictors
predictors<-c("Credit_History", "LoanAmount", "Loan_Amount_Term", "Applic
```

[ ]

```
model_gbm<-train(trainSet[,predictors],trainSet[,outcomeName],method='gbm
model_rf<-train(trainSet[,predictors],trainSet[,outcomeName],method='rf')
model_nnet<-train(trainSet[,predictors],trainSet[,outcomeName],method='nn
model_glm<-train(trainSet[,predictors],trainSet[,outcomeName],method='glm
```

```
[ ]  fitControl <- trainControl(
       method = "repeatedcv",
       number = 5,
       repeats = 5)
```

```
[ ]  modelLookup(model='gbm')

     #model         parameter                         label forReg forClass probMode
     #1    gbm           n.trees    # Boosting Iterations   TRUE     TRUE        T
     #2    gbm interaction.depth           Max Tree Depth   TRUE     TRUE        T
     #3    gbm         shrinkage                Shrinkage   TRUE     TRUE        T
     #4    gbm    n.minobsinnode Min. Terminal Node Size   TRUE     TRUE        T
     #using grid search

     #Creating grid
     grid <- expand.grid(n.trees=c(10,20,50,100,500,1000),shrinkage=c(0.01,0.0

     # training the model
     model_gbm<-train(trainSet[,predictors],trainSet[,outcomeName],method='gbm

     # summarizing the model
     print(model_gbm)

     #Stochastic Gradient Boosting
     #461 samples
     #5 predictor
     #2 classes: '0', '1'

     #No pre-processing
     #Resampling: Cross-Validated (5 fold, repeated 5 times)
     #Summary of sample sizes: 368, 370, 369, 369, 368, 369, ...
     #Resampling results across tuning parameters:

     #  shrinkage  interaction.depth  n.minobsinnode  n.trees  Accuracy   Kapp
     #0.01        1                   3                 10     0.6876416  0.0000
     #0.01        1                   3                 20     0.6876416  0.0000
     #0.01        1                   3                 50     0.7982345  0.4423
     #0.01        1                   3                100     0.7952190  0.4364
     #0.01        1                   3                500     0.7904882  0.4342
     #0.01        1                   3               1000     0.7913627  0.4421
     #0.01        1                   5                 10     0.6876416  0.0000
     #0.01        1                   5                 20     0.6876416  0.0000
     #0.01        1                   5                 50     0.7982345  0.4423
     #0.01        1                   5                100     0.7943635  0.4351
     #0.01        1                   5                500     0.7930783  0.4411
     #0.01        1                   5               1000     0.7913720  0.4417
     #0.01        1                  10                 10     0.6876416  0.0000
     #0.01        1                  10                 20     0.6876416  0.0000
     #0.01        1                  10                 50     0.7982345  0.4423
     #0.01        1                  10                100     0.7943635  0.4351
     #0.01        1                  10                500     0.7939525  0.4426
     #0.01        1                  10               1000     0.7948362  0.4476
```

```
#0.01         5                  3                 10      0.6876416  0.0000
#0.01         5                  3                 20      0.6876416  0.0000
#0.01         5                  3                 50      0.7960556  0.4349
#0.01         5                  3                100      0.7934987  0.4345
#0.01         5                  3                500      0.7775055  0.4147
#...
#0.50         5                 10                100      0.7045617  0.2834
#0.50         5                 10                500      0.6924480  0.2650
#0.50         5                 10               1000      0.7115234  0.3050
#0.50        10                  3                 10      0.7389117  0.3681
#0.50        10                  3                 20      0.7228519  0.3317
#0.50        10                  3                 50      0.7180833  0.3159
#0.50        10                  3                100      0.7172417  0.3189
#0.50        10                  3                500      0.7058472  0.3098
#0.50        10                  3               1000      0.7001852  0.2967
#0.50        10                  5                 10      0.7266895  0.3378
#0.50        10                  5                 20      0.7154746  0.3197
#0.50        10                  5                 50      0.7063535  0.2984
#0.50        10                  5                100      0.7151012  0.3141
#0.50        10                  5                500      0.7108516  0.3146
#0.50        10                  5               1000      0.7147320  0.3225
#0.50        10                 10                 10      0.7314871  0.3327
#0.50        10                 10                 20      0.7150814  0.3081
#0.50        10                 10                 50      0.6993723  0.2815
#0.50        10                 10                100      0.6977416  0.2719
#0.50        10                 10                500      0.7037864  0.2854
#0.50        10                 10               1000      0.6995610  0.2869
```

[ ]

```
#using tune length
model_gbm<-train(trainSet[,predictors],trainSet[,outcomeName],method='gbm
print(model_gbm)

#Stochastic Gradient Boosting
#461 samples
#5 predictor
#2 classes: '0', '1'

#No pre-processing
#Resampling: Cross-Validated (5 fold, repeated 5 times)
#Summary of sample sizes: 368, 369, 369, 370, 368, 369, ...
#Resampling results across tuning parameters:

#  interaction.depth  n.trees  Accuracy   Kappa
#1                  50       0.7978084  0.4541008
#1                 100       0.7978177  0.4566764
#1                 150       0.7934792  0.4472347
#1                 200       0.7904310  0.4424091
#1                 250       0.7869714  0.4342797
#1                 300       0.7830488  0.4262414
...
```

```
#10                   100      0.7575230  0.3860319
#10                   150      0.7479757  0.3719707
#10                   200      0.7397290  0.3566972
#10                   250      0.7397285  0.3561990
#10                   300      0.7362552  0.3513413
#10                   350      0.7340812  0.3453415
#10                   400      0.7336416  0.3453117
#10                   450      0.7306027  0.3415153
#10                   500      0.7253854  0.3295929
```

```
#Checking variable importance for GLM
varImp(object=model_glm)
#glm variable importance

#Overall
#Credit_History    100.000
#CoapplicantIncome  17.218
#Loan_Amount_Term   12.988
#LoanAmount          5.632
#ApplicantIncome     0.000

#Plotting Variable importance for GLM
plot(varImp(object=model_glm),main="GLM - Variable Importance")
```

```
#Predictions
predictions<-predict.train(object=model_gbm,testSet[,predictors],type="ra
table(predictions)
#predictions
#0   1
#28 125
```

```
confusionMatrix(predictions,testSet[,outcomeName])
#Confusion Matrix and Statistics
#Reference
#Prediction   0   1
#0  25   3
#1  23 102

#Accuracy : 0.8301
#95% CI : (0.761, 0.8859)
#No Information Rate : 0.6863
#P-Value [Acc > NIR] : 4.049e-05
#Kappa : 0.555
#Mcnemar's Test P-Value : 0.0001944
#Sensitivity : 0.5208
#Specificity : 0.9714
#Pos Pred Value : 0.8929
#Neg Pred Value : 0.8160
```

```
#Prevalence : 0.3137
#Detection Rate : 0.1634
#Detection Prevalence : 0.1830
#Balanced Accuracy : 0.7461
#'Positive' Class : 0
```

Preguntas para regresión

Supongamos que tenemos un conjunto de datos con 5 variables en el espacio $\mathcal{X}$:
$X_1 = GPA,\ X_2 = IQ,\ X_3 = Genero,\ X_4 = Interacción\ X_1 * X_3,\ X_5 = Interacción$
.

La variable dependiente es el primer salario después de graduarse.

Sopongamos que ajustamos un modelo de regresión lineal y obtenemos:

- $\beta_0 = 50$
- $\beta_1 = 20$
- $\beta_2 = 0.07$
- $\beta_3 = 35$
- $\beta_4 = 0.01$
- $\beta_5 = -10$

1. ¿Cuál de las siguientes es correcta y por qué?

   1.1 Para valores fijos de IQ y GPA, los hombres ganan, en promedio, más que las mujeres.

   1.2 Para valores fijos de IQ y GPA, las mujeres ganan, en promedio, más que los hombres.

   1.3 Para valores fijos de IQ y GPA, los hombres ganan, en promedio, más que las mujeres siempre que el GPA sea suficientemente alto.

   1.4 Para valores fijos de IQ y GPA, las mujeres ganan, en promedio, más que los hombres siempre que el GPA sea suficientemente alto.

2. Prediga el salario de una mujer con IQ de 110 y GPA de 4.0

Considere una regresión lineal sin intercepto, es decir

$$y_i = x_i\beta$$

con

$$\beta = \sum_{i=1}^{n} x_i y_i / (\sum_{i'=1}^{n} x_{i'}^2)$$

Muestres que podemos escribir:

$$y_i = \sum_{i'=1}^{n} a_{i'} y_{i'}$$

¿Quién es $a_{i'}$

Pruebe que en el caso de regresión lineal simple, la R^2 es igual al cuadrado de la correlación entre x y y

La siguiente tabla corresponde a la salida de un modelo de regresión con el cual se busca explicar ventas con inversiones en marketing en TV, radio y periódicos.

	Coefficient	**Std. error**	**t-statistic**	**p-value**
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	−0.001	0.0059	−0.18	0.8599

Describa la hipótesis nula que se realiza. Explique que conclusiones puede obtener basado en la tabla (la explicación no debe ser técnica).

Para el modelo de regresión logística pruebe que si:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

entonces:

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}$$

Suponga que recolectamos datos para un grupo de estudiantes de una clase del seminario de estadística y medimos $X_1 = horas\ de\ estudio,\ X_2 = promedio,\ Y = sacará\ 10.$ Ajustamos un modelo de regresión logística y obtenemos:

- $\beta_0 = -6$
- $\beta_1 = 0.05$
- $\beta_2 = 1$

1. Estime la probabilidad de que un estudiante que estudia 40horas y tiene promedio de 9 obtenga 10 en la clase

2. ¿Cuántas horas necesita estudiar el alumno anterior para tener buena probabilidad de sacar 10 en la clase?

Este ejercicio debe hacerse con los datos *Weekly* del paquete `ISLR`.

1. Haga descriptivos, comente.
2. Ajuste una regresión logística con $y = Direction$ y las 5 variables *lag* + Volume como el espacio $\mathcal{X}$, comente
3. Ajuste un modelo de regresión logística usando el periodo 1990-2008 como conjunto de entrenamiento y usando Lag2 como la única variable del espacio $\mathcal{X}$, prediga y evalue los resultados para el periodo 2009-2010, comente