Feature Selection

Karen Mazidi

Look for correlations in Pima data

The findCorrelation() function suggests that we could remove column 6, mass, because it correlates with triceps. And that we could remove column 2, glucose, because it correlates with insulin.

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(mlbench)
data("PimaIndiansDiabetes2")
df <- PimaIndiansDiabetes2[complete.cases(PimaIndiansDiabetes2[]),]
corMatrix <- cor(df[,1:7])
findCorrelation(corMatrix, cutoff=0.5, verbose=TRUE)
```

```
## Compare row 6  and column  4 with corr  0.664
##   Means:  0.265 vs 0.187 so flagging column 6
## Compare row 2  and column  5 with corr  0.581
##   Means:  0.266 vs 0.161 so flagging column 2
## All correlations <= 0.5

## [1] 6 2
```

Remove the highly correlated columns

```
df <- df[,-c(2,6)]
```

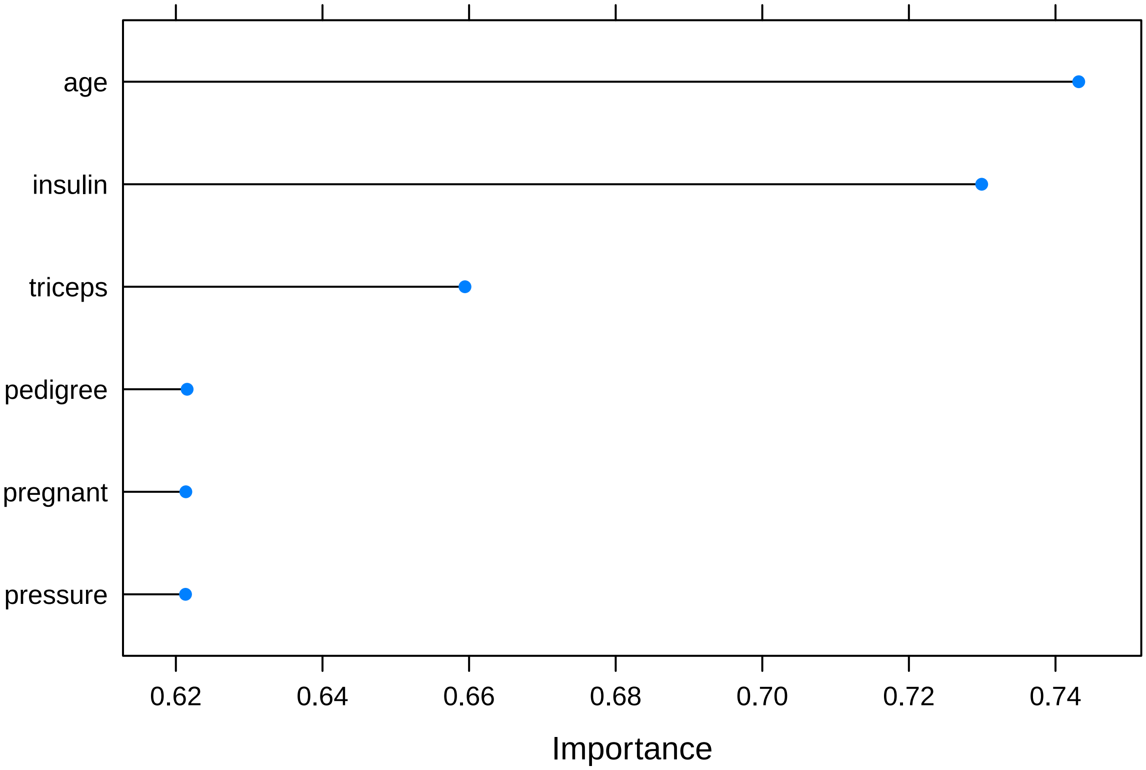
Rank features

The varImp() function ranks variables by importance. It requires a model which we trained on method knn, using control parameters stored in variable ctrl.

```
ctrl <- trainControl(method="repeatedcv", repeats=5)
model <- train(diabetes~., data=df, method="knn", preProcess="scale", trControl=ctrl)
importance <- varImp(model, scale=FALSE)
importance
```

```
## ROC curve variable importance
##
##          Importance
## age          0.7432
## insulin      0.7299
## triceps      0.6594
## pedigree     0.6215
## pregnant     0.6214
## pressure     0.6213
```

```
plot(importance)
```



###

Recursive feature selection

We start with the data set including all columns.

```
df <- PimaIndiansDiabetes2[complete.cases(PimaIndiansDiabetes2[]),]
ctrl <- rfeControl(functions=rfFuncs, method="cv", number=10)
rfe_out <- rfe(df[,1:7], df[,8], sizes=c(1:7), rfeControl=ctrl)
rfe_out
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold)
##
## Resampling performance over subset size:
##
##  Variables  RMSE Rsquared   MAE RMSESD RsquaredSD  MAESD Selected
##          1 7.297   0.4945 5.202  1.541    0.13628 0.8784
##          2 7.502   0.4700 5.332  1.244    0.09706 0.8018
##          3 7.342   0.4858 5.321  1.394    0.10811 0.8853
##          4 7.235   0.5051 5.293  1.460    0.11068 0.8894
##          5 7.176   0.5199 5.272  1.500    0.11619 0.9819
##          6 7.039   0.5313 5.113  1.332    0.08983 0.8708
##          7 6.964   0.5453 5.079  1.316    0.08578 0.9140        *
##
## The top 5 variables (out of 7):
##    pregnant, glucose, insulin, triceps, pressure
```

FSelector

```
library(FSelector)
var_scores <- random.forest.importance(diabetes~., df)
var_scores
```

```
##          attr_importance
## pregnant       14.359129
## glucose        50.145825
## pressure        1.419350
## triceps         7.829199
## insulin        22.248548
## mass           11.511762
## pedigree        8.202778
## age            26.711850
```