> library(pROC)

Type 'citation("pROC")' for a citation.

Attaching package: ‘pROC’

The following objects are masked from ‘package:stats’

cov, smooth, var

> library(mlbench)

Error in library(mlbench) : there is no package called ‘mlbench’

> utils:::menuInstallPkgs()

trying URL 'https://cran.ma.imperial.ac.uk/bin/windows/contrib/3.6/mlbench\_2.1-1.zip'

Content type 'application/zip' length 1061047 bytes (1.0 MB)

downloaded 1.0 MB

package ‘mlbench’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\OKEDARAB\AppData\Local\Temp\RtmpOMA4vs\downloaded\_packages

> library(caret)

Loading required package: lattice

Loading required package: ggplot2

> library(mlbench)

# Example 1 student applications (classification)

> data <- read.csv(file.choose(), header = T)

Error in file.choose() : file choice cancelled

> data <- read.csv("~/downloads/binary.csv/")

Error in file(file, "rt") : cannot open the connection

In addition: Warning message:

In file(file, "rt") :

cannot open file 'C:\Users\OKEDARAB\Documents/downloads/binary.csv/': No such file or directory

> data <- read.csv(file.choose(), header = T)

> str(data)

'data.frame': 400 obs. of 4 variables:

$ admit: int 0 1 1 1 0 1 1 0 1 0 ...

$ gre: int 380 660 800 640 520 760 560 400 540 700 ...

$ gpa: num 3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...

$ rank: int 3 3 1 4 4 2 1 2 3 2 ...

> data$admit[data$admit ==0] <- 'No'

> data$admit[data$admit ==1] <- 'Yes'

> data$admit <- factor(data$admit)

> # Splitting the data

> set.seed(1234)

> ind <- sample(2, nrow(data), replace = T, prob = c(0.7, 0.3))

> training <- data[ind == 1,]

> test <- data[ind == 2,]

> # Building KNN model

> trControl <- trainControl(method = "repeatedcv",

+ number = 10,

+ repeats = 3)

> set.seed(222)

> fit <- train(admit ~ .,

+ data = training,

+ method = 'knn',

+ tuneLength = 2-,

Error: unexpected ',' in:

"method = 'knn',

tuneLength = 2-,"

> > fit <- train(admit ~ .,

Error: unexpected '>' in ">"

> + data = training,

Error: unexpected ',' in "+ data = training,"

> + method = 'knn',

Error: unexpected ',' in "+ method = 'knn',"

> + tuneLength = 2-,

Error: unexpected ',' in "+ tuneLength = 2-,"

> fit <- train(admit ~ .

+ data = training,

Error: unexpected symbol in:

"fit <- train(admit ~ .

data"

> fit <- train(admit ~ .,

+ data = training,

+ method = 'knn',

+ tuneLength = 20,

+ trControl = trControl,

+ preProc = c("center", "scale"))

Error: package e1071 is required

> utils:::menuInstallPkgs()

trying URL 'https://cran.ma.imperial.ac.uk/bin/windows/contrib/3.6/e1071\_1.7-3.zip'

Content type 'application/zip' length 1021796 bytes (997 KB)

downloaded 997 KB

package ‘e1071’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\OKEDARAB\AppData\Local\Temp\RtmpOMA4vs\downloaded\_packages

> fit <- train(admit ~ .,

+ + data = training,

Error: unexpected '=' in:

" fit <- train(admit ~ .,

+ data ="

> + method = 'knn',

Error: unexpected ',' in "+ method = 'knn',"

> + tuneLength = 20,

Error: unexpected ',' in "+ tuneLength = 20,"

> + trControl = trControl,

Error: unexpected ',' in "+ trControl = trControl,"

> + preProc = c("center", "scale"))

Error: unexpected ')' in "+ preProc = c("center", "scale"))"

> fit <- train(admit ~ .,

+ data = training,

+ method = 'knn',

+ tuneLength = 20,

+ trControl = trControl,

+ preProc = c("center", "scale"))

> # model performance

> fit

k-Nearest Neighbors

* 284 samples
* 3 predictor
* 2 classes: 'No', 'Yes'

Pre-processing: centered (3), scaled (3)

Resampling: Cross-Validated (10 fold, repeated 3 times)

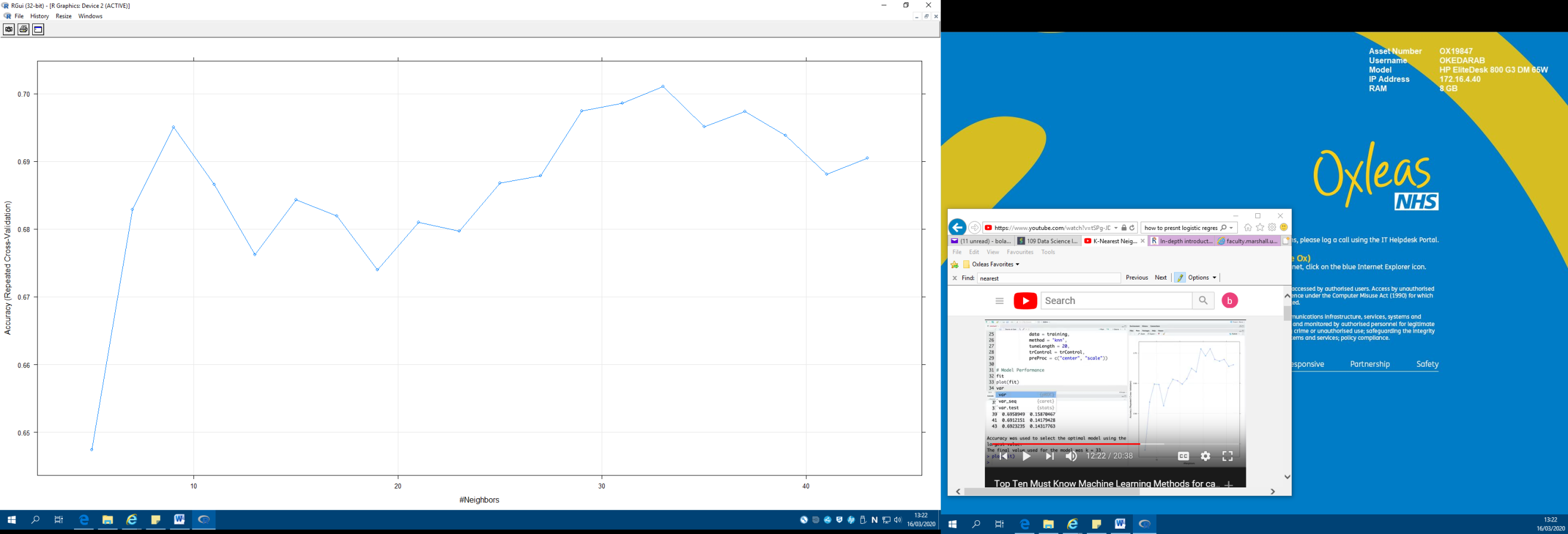
Summary of sample sizes: 256, 255, 256, 255, 256, 256, ...

Resampling results across tuning parameters:

|  |  |  |
| --- | --- | --- |
| k | Accuracy | Kappa |
| 5 | 0.6474083 | 0.1146828 |
| 7 | 0.6828763 | 0.1890827 |
| 9 | 0.6950219 | 0.2032152 |
| 11 | 0.6865709 | 0.1866015 |
| 13 | 0.6762562 | 0.1559336 |
| 15 | 0.6843487 | 0.1645697 |
| 17 | 0.6819677 | 0.1600165 |
| 19 | 0.6739546 | 0.1376984 |
| 21 | 0.6809743 | 0.1398042 |
| 23 | 0.6796634 | 0.1354138 |
| 25 | 0.6867679 | 0.1456241 |
| 27 | 0.6878790 | 0.1427854 |
| 29 | 0.6974028 | 0.1654528 |
| 31 | 0.6985933 | 0.1696574 |
| 33 | 0.7010536 | 0.1711829 |
| 35 | 0.6951013 | 0.1583319 |
| 37 | 0.6973618 | 0.1611921 |
| 39 | 0.6938314 | 0.1493096 |
| 41 | 0.6881199 | 0.1392181 |
| 43 | 0.6904598 | 0.1469002 |

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was k = 33.

> plot(fit)

> varImp(fit)

ROC curve variable importance

Importance

Gpa: 100.00

Rank: 25.18

Gre: 0.00

> pred <- predict(fit, newdata = test)

> pred

[1] No No No Yes No No No No No No No No No No No No Yes No

[19] No No Yes No No No No No No No No No No No No No Yes No

[37] No No No No Yes No No No No No No No No No No No No No

[55] No No No No No No No No No No No No No No No No No No

[73] No No No No No No No No No No No No No No No No No No

[91] No No No No No No No No No No No No Yes No No No Yes No

[109] No No No No No No No No

Levels: No Yes

> summary(pred)

No - 109

Yes - 7

> confusionMatrix(pred, test$admit)

Confusion Matrix and Statistics

|  |  |  |
| --- | --- | --- |
|  | Reference |  |
| Prediction | No | Yes |
| No | 79 | 30 |
| Yes | 3 | 4 |

Accuracy : 0.7155

95% CI : (0.6243, 0.7954)

No Information Rate : 0.7069

P-Value [Acc > NIR] : 0.465

Kappa : 0.1056

Mcnemar's Test P-Value : 6.011e-06

Sensitivity : 0.9634

Specificity : 0.1176

Pos Pred Value : 0.7248

Neg Pred Value : 0.5714

Prevalence : 0.7069

Detection Rate : 0.6810

Detection Prevalence : 0.9397

Balanced Accuracy : 0.5405

'Positive' Class : No

# Error

# model performance

trControl <- trainControl(method = "repeatedcv",

> number = 10,

Error: unexpected ',' in "number = 10,"

> repeats = 3)

Error: unexpected ')' in "repeats = 3)"

> trControl <- trainControl(method = "repeatedcv",

+ number = 10,

+ repeats = 3)

> trControl <- trainControl(method = "repeatedcv",

+ number = 10,

+ repeats = 3)

> > trControl <- trainControl(method = "repeatedcv",

Error: unexpected '>' in ">"

> + number = 10,

Error: unexpected ',' in "+ number = 10,"

> + repeats = 3)

Error: unexpected ')' in "+ repeats = 3)"

> trControl <- trainControl(method = "repeatedcv",

+ number = 10,

+ repeats = 3)

> trControl <- trainControl(method = "repeatedcv",

+ number = 10,

+ repeats = 3)

> trControl <- trainControl(method = "repeatedcv",

+ number = 10,

+ repeats = 3,

+ classProbs = TRUE,

+ summaryFunction = twoClassSummary())

Error in levels(data[, "pred"]) :

argument "data" is missing, with no default

> trControl <- trainControl(method = "repeatedcv",

+ number = 10,

+ repeats = 3,

+ classProbs = TRUE,

+ summaryFunction = twoClassSummary)

> fit <- train(admit ~ .,

+ data = training,

+ method = 'knn',

+ tuneLength = 20,

+ trControl = trControl,

+ preProc = c("center", "scale"),

+ metric = "ROC",

+ tuneGrid(k=1:60))

Error in { : task 1 failed - "could not find function "tuneGrid""

In addition: There were 50 or more warnings (use warnings() to see the first 50)

> fit <- train(admit ~ .,

+ data = training,

+ method = 'knn',

+ tuneLength = 20,

+ trControl = trControl,

+ preProc = c("center", "scale"),

+ metric = "ROC",

+ tuneGrid = expand.grid(k = 1:60))

> fit

k-Nearest Neighbors

284 samples

3 predictor

2 classes: 'No', 'Yes'

Pre-processing: centered (3), scaled (3)

Resampling: Cross-Validated (10 fold, repeated 3 times)

Summary of sample sizes: 256, 256, 256, 256, 256, 256, ...

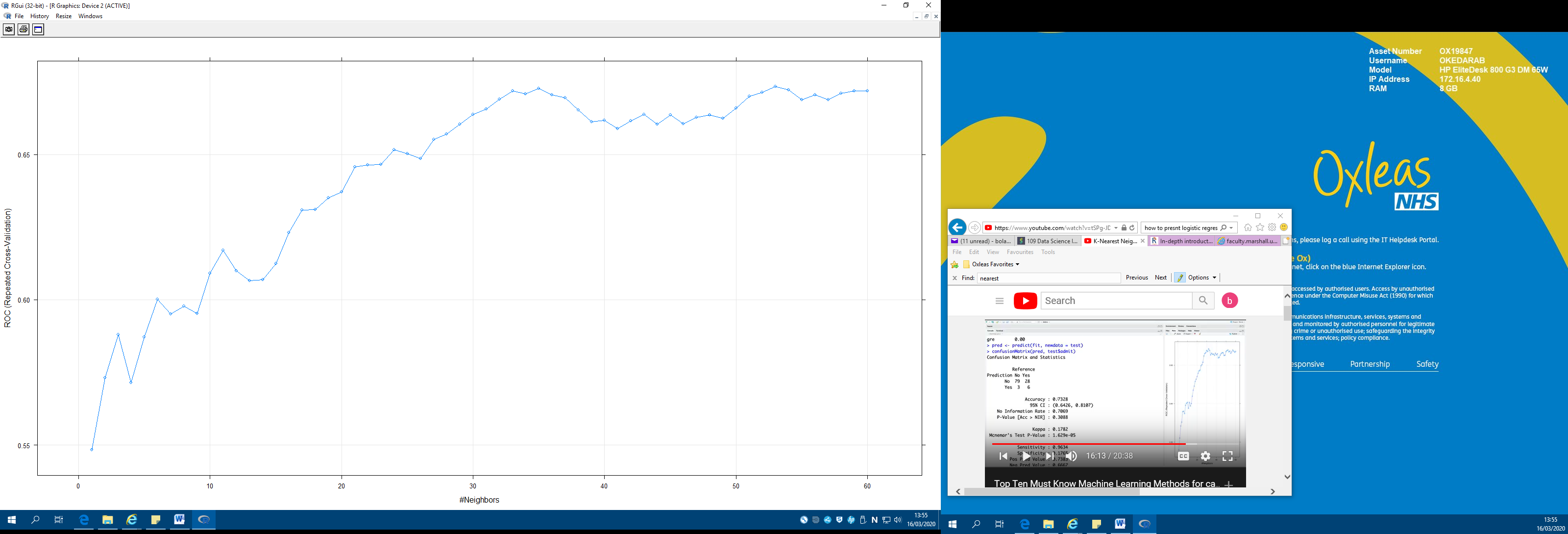
Resampling results across tuning parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| k | ROC | Sens | Spec |
| 1 | 0.5483499 | 0.6969298 | 0.3981481 |
| 2 | 0.5731238 | 0.7000877 | 0.3696296 |
| 3 | 0.5880673 | 0.8064035 | 0.3322222 |
| … |  |  |  |
| 58 | 0.6710931 | 0.9670175 | 0.1148148 |
| 59 | 0.6718826 | 0.9687719 | 0.1151852 |
| 60 | 0.6719074 | 0.9687719 | 0.1185185 |

ROC was used to select the optimal model using the largest value.

The final value used for the model was k = 53.

> plot(fit)



> varImp(fit)

ROC curve variable importance

|  |  |
| --- | --- |
| gpa | 100.00 |
| rank | 25.18 |
| gre | 0.00 |

> pred <- predict(fit, newdata = test)

> confusionMatrix(pred, test$admit)

Confusion Matrix and Statistics

|  |  |  |
| --- | --- | --- |
|  | Reference |  |
| Prediction | No | Yes |
| No | 81 | 32 |
| Yes | 1 | 2 |

Accuracy : 0.7155

95% CI : (0.6243, 0.7954)

No Information Rate : 0.7069

P-Value [Acc > NIR] : 0.465

Kappa : 0.0636

Mcnemar's Test P-Value : 1.767e-07

Sensitivity : 0.98780

Specificity : 0.05882

Pos Pred Value : 0.71681

Neg Pred Value : 0.66667

Prevalence : 0.70690

Detection Rate : 0.69828

Detection Prevalence : 0.97414

Balanced Accuracy : 0.52331

'Positive' Class : No

# Example 2 - Boston housing (regression)

> data("BostonHousing")

> data <- BostonHousing

> str (data)

'data.frame': 506 obs. of 14 variables:

$ crim: num 0.00632 0.02731 0.02729 0.03237 0.06905 ...

$ zn: num 18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...

$ indus: num 2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...

$ chas: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

$ nox: num 0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...

$ rm: num 6.58 6.42 7.18 7 7.15 ...

$ age: num 65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...

$ dis: num 4.09 4.97 4.97 6.06 6.06 ...

$ rad: num 1 2 2 3 3 3 5 5 5 5 ...

$ tax: num 296 242 242 222 222 222 311 311 311 311 ...

$ ptratio: num 15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...

$ b: num 397 397 393 395 397 ...

$ lstat: num 4.98 9.14 4.03 2.94 5.33 ...

$ medv: num 24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...

> # Splitting the data

> set.seed(1234)

> ind <- sample(2, nrow(data), replace = T, prob = c(0.7, 0.3))

> training <- data[ind == 1,]

> test <- data[ind == 2,]

> # Building the KNN model

> trControl <- trainControl(method = 'repeatedcv',

+ number = 10,

+ repeats = 3)

> set.seed(333)

> fit <- train(medv ~.,

+ data = training,

+ tuneGrid = expand.grid(k=1:70),

+ method = 'knn',

+ trcontrol = trControl,

+ preProc = c('center', 'scale'))

> # Model performance

> fit

k-Nearest Neighbors

355 samples

13 predictor

Pre-processing: centered (13), scaled (13)

Resampling: Bootstrapped (25 reps)

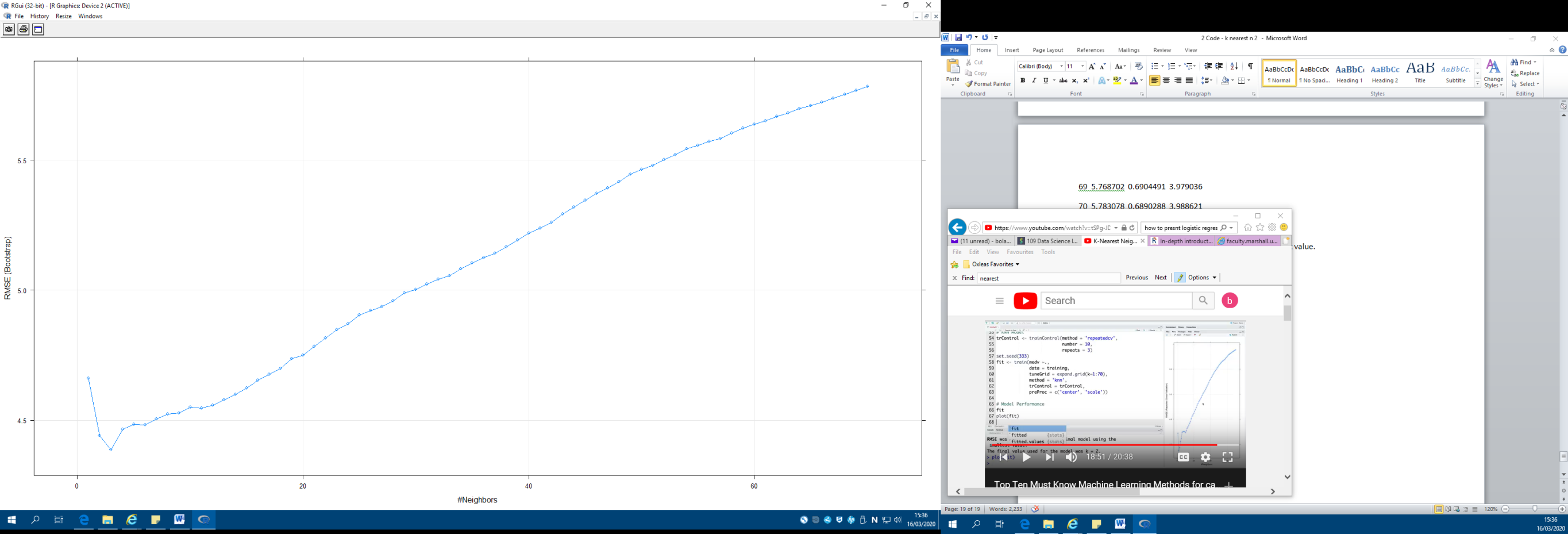
Summary of sample sizes: 355, 355, 355, 355, 355, 355, ...

Resampling results across tuning parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| k | RMSE | Rsquared | MAE |
| 1 | 4.661605 | 0.7394251 | 3.006525 |
| 2 | 4.441444 | 0.7551155 | 2.939341 |
| 3 | 4.386535 | 0.7593905 | 2.893563 |
| … |  |  |  |
| 68 | 5.753221 | 0.6921072 | 3.967463 |
| 69 | 5.768702 | 0.6904491 | 3.979036 |
| 70 | 5.783078 | 0.6890288 | 3.988621 |

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was k = 3.

> plot(fit)

> varImp(fit)

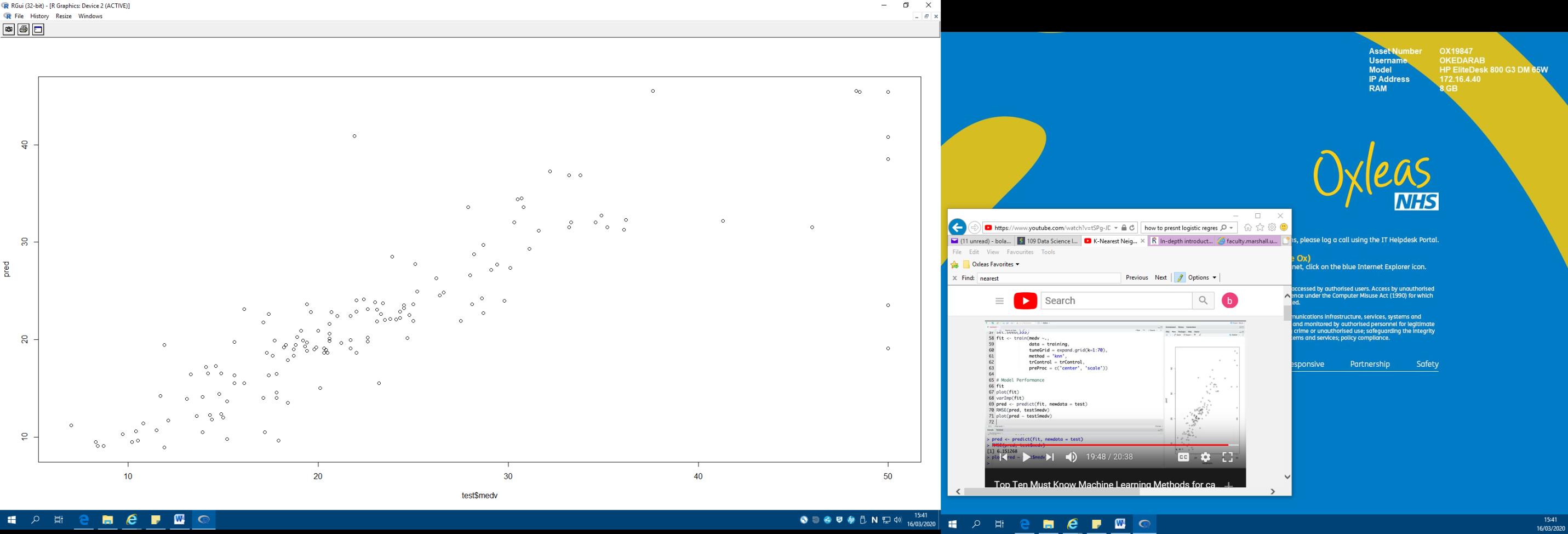
loess r-squared variable importance

|  |  |
| --- | --- |
| Overall |  |
| lstat | 100.00 |
| rm | 94.28 |
| indus | 88.94 |
| tax | 69.84 |
| ptratio | 69.20 |
| rad | 42.23 |
| zn | 38.95 |
| crim | 35.25 |
| nox | 31.91 |
| b | 24.68 |
| age | 22.83 |
| dis | 19.78 |
| chas | 0.00 |

> pred <- predict(fit, newdata = test)

> RMSE(pred, test$medv)

[1] 6.127333

> plot(pred ~ test$medv)