# **Drills With R On K-NN Models**

Suraj Eswaran

University Of The Cumberlands

2024 Spring – Statistics For Data Science (MSDS – 531 – A01) – First Bi-Term

Professor Dr. Danny T. Barnes

02/12/2024

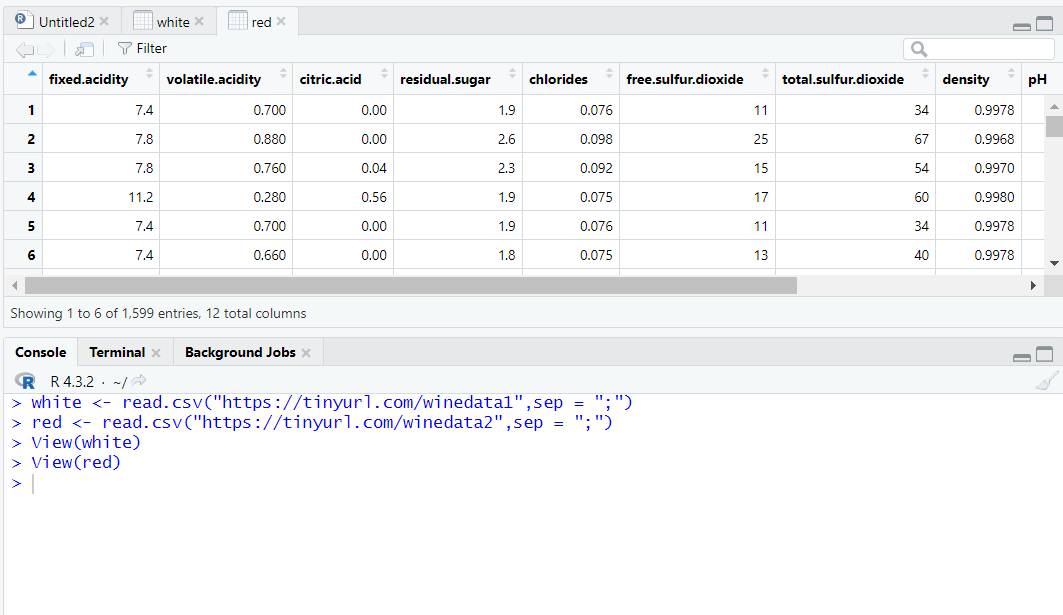
## This problem is related to Nearest neighbors classifiers described in section 9.5 in “Modern Statistics with R” - https://modernstatisticswithr.com: Fit a kNN classification model to the wine data, using pH, alcohol, fixed. Acidity, and residual. Sugar as explanatory variables. Evaluate its performance using 10-fold cross-validation, using AUC to choose the best k.

Before we start working on KNN classification model, we have to first import data with the help of read.csv function in R where we are importing data for both red and white wines.

> # Import data about white and red wines:

> white <- read.csv("https://tinyurl.com/winedata1",sep = ";")

> red <- read.csv("https://tinyurl.com/winedata2",sep = ";")

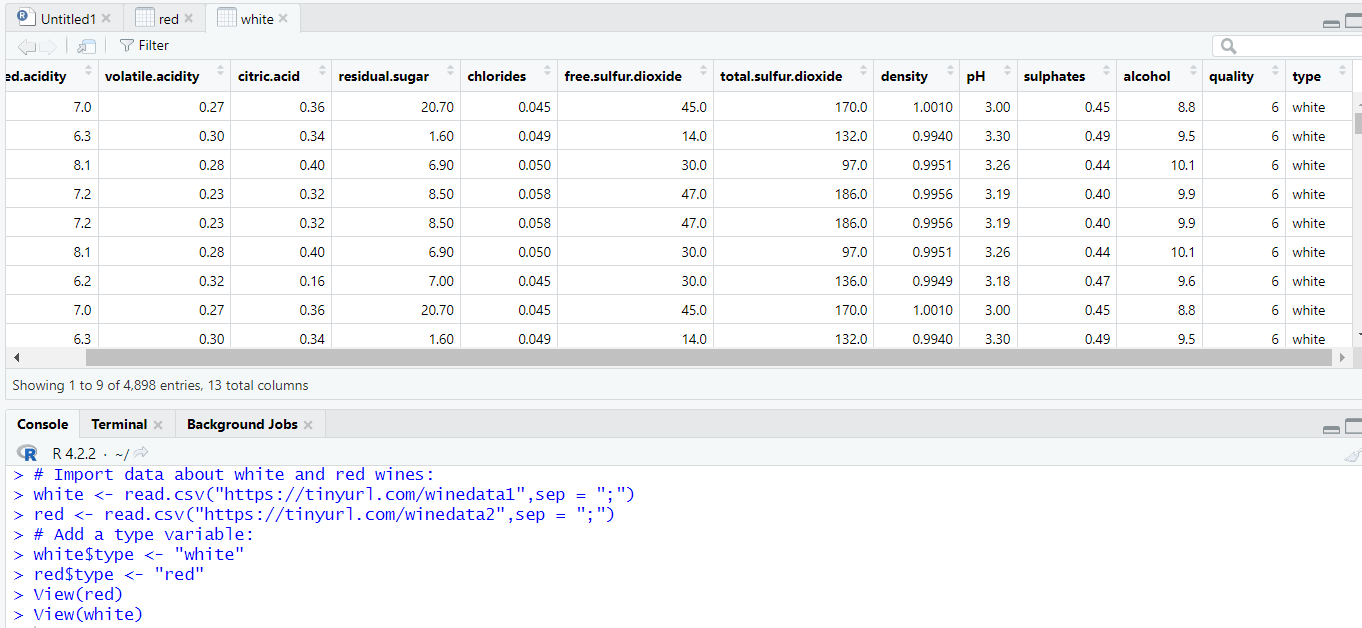


Now, we are adding a new variable name “type” for both the data, where red is value for red data and white for white data.

> # Add a type variable:

> white$type <- "white"

> red$type <- "red"

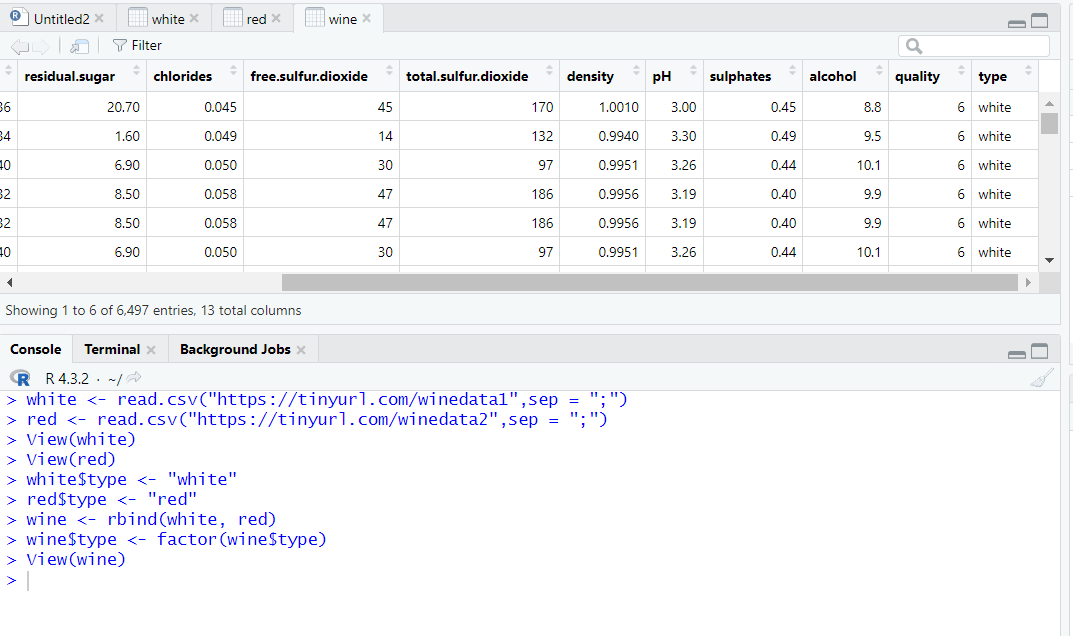


Now, we are merging the data of red wine and white wine, where we are using “rbind” function which is basically rows binding that joins multiple groups of rows together to a single whole data.

> # Merge the datasets:

> wine <- rbind(white, red)

> wine$type <- factor(wine$type)



In order to perform machine learning model and evaluating it, we need to load packages like “caret”(Classification And REgression Training) and “MLeval”(Machine Learning Evaluation).

> install.packages("caret", dependencies = TRUE)

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/

Installing package into ‘C:/Users/Suraj/AppData/Local/R/win-library/4.2’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caret\_6.0-94.zip'

Content type 'application/zip' length 3579176 bytes (3.4 MB)

downloaded 3.4 MB

package ‘caret’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Suraj\AppData\Local\Temp\RtmpWmmdbJ\downloaded\_packages

> library(caret)

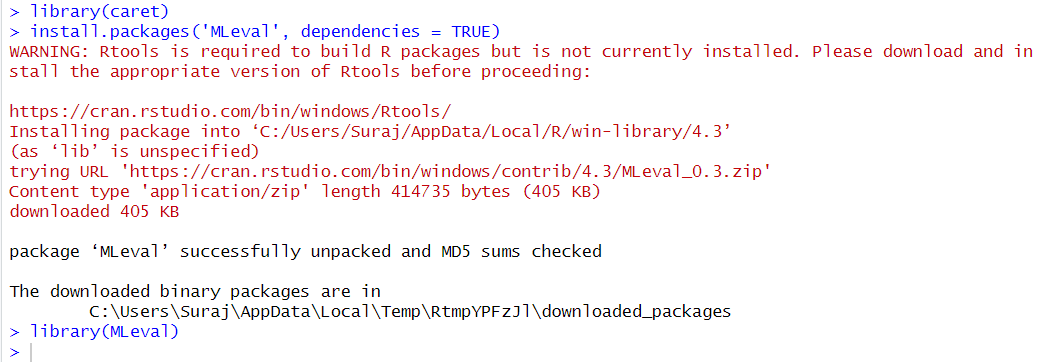
Loading required package: ggplot2

Loading required package: lattice

Warning messages:

1: package ‘caret’ was built under R version 4.2.3

2: package ‘ggplot2’ was built under R version 4.2.3



> # to visualize results you need the following

> install.packages('MLeval', dependencies = TRUE)

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/

Installing package into ‘C:/Users/Suraj/AppData/Local/R/win-library/4.2’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/MLeval\_0.3.zip'

Content type 'application/zip' length 415000 bytes (405 KB)

downloaded 405 KB

package ‘MLeval’ successfully unpacked and MD5 sums checked

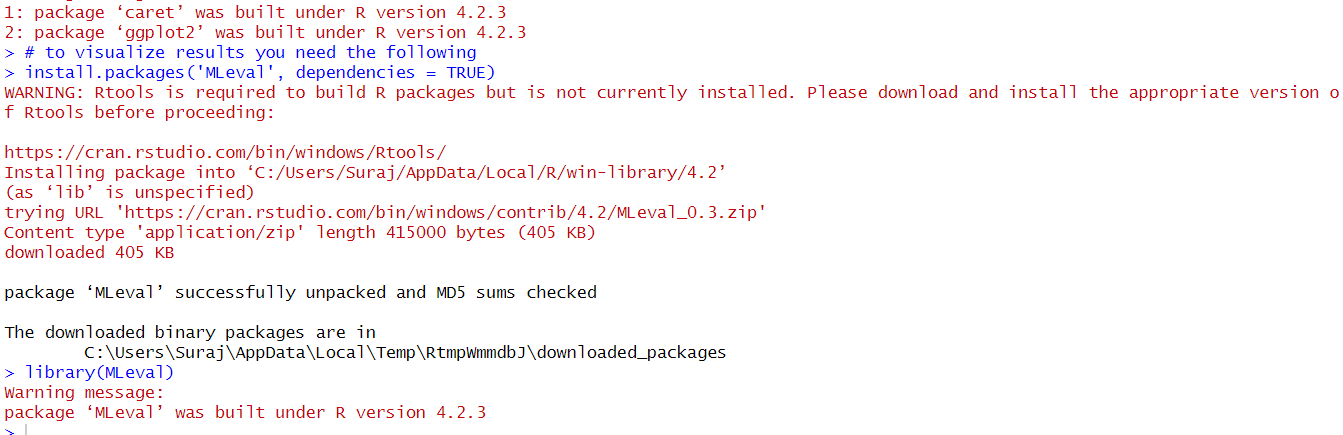
The downloaded binary packages are in

C:\Users\Suraj\AppData\Local\Temp\RtmpWmmdbJ\downloaded\_packages

> library(MLeval)

Warning message:

package ‘MLeval’ was built under R version 4.2.3



**1. Provide the commands in plain text that you used to solve the problem.**

Explanatory variables are those variables that are utilized to explain or predict the outcome of interest. In the context of machine learning, these are the input features used to make predictions. The values of these variables are manipulated or controlled to observe their impact on the response variable. Here, we are using pH, alcohol, fixed.acidity and residual sugar for explanatory variables. Response variables are those variables that is being predicted or explained with the help of explanatory variables. In the context of machine learning, this is the variable that the model aims to predict based on the explanatory variables. Wine type is considered as response variable for model.

For utilizing resampling method, we are using ‘trainControl’ which is widely used for machine learning tasks. It allows for fine-tuning of the control parameters for the training process, providing flexibility and control over various aspects of model training. In R, we have trainControl function which has parameters like method(here we are using repeated cross-validation (repeatedcv) for model evaluation, number which is given as 10(number of folds for repeated cross-validation), summary function(calculates performance metrics suitable for binary classification tasks) equal to ‘twoClassSummary’, savePredictions as TRUE indicates that predictions will be saved during the cross-validation process. This can be useful for further analysis or diagnostics and classProbs as TRUE which specifies that class probabilities will be computed for classification models which means that the model will output probabilities for each class label, which can be useful for evaluating confidence in predictions and for certain evaluation metrics.

> fit\_Control <- trainControl(method = "repeatedcv",

+ number = 10,

+ summaryFunction = twoClassSummary,

+ savePredictions = TRUE,

+ classProbs = TRUE)

Using the train function from the caret package in R, we were training a classification model on the wine dataset using k-Nearest Neighbors (kNN) algorithm with specified control parameters like formula “type ~ pH + alcohol + fixed.acidity + residual.sugar” specifies the dependent variable (type) and the independent variables (pH, alcohol, fixed.acidity, and residual.sugar) that are used in the model, wine data that is used for training the model, tc as trainControl which helps in specifying the control parameters for the training process, method as “knn” where we are using k-Nearest Neighbors (kNN), metric as “ROC” Specifies the performance metric to optimize during the tuning process ,as ROC is commonly used for binary classification tasks to evaluate the trade-off between true positive rate and false positive rate, tuneLength equal to 15 where it describes the number of different values of the tuning parameter to be evaluated during the model tuning process, preProcess parameter shows the pre-processing steps which is applied to the data before modeling them.

> fitted\_model <- train(type ~ pH + alcohol + fixed.acidity + residual.sugar,

+ data = wine,

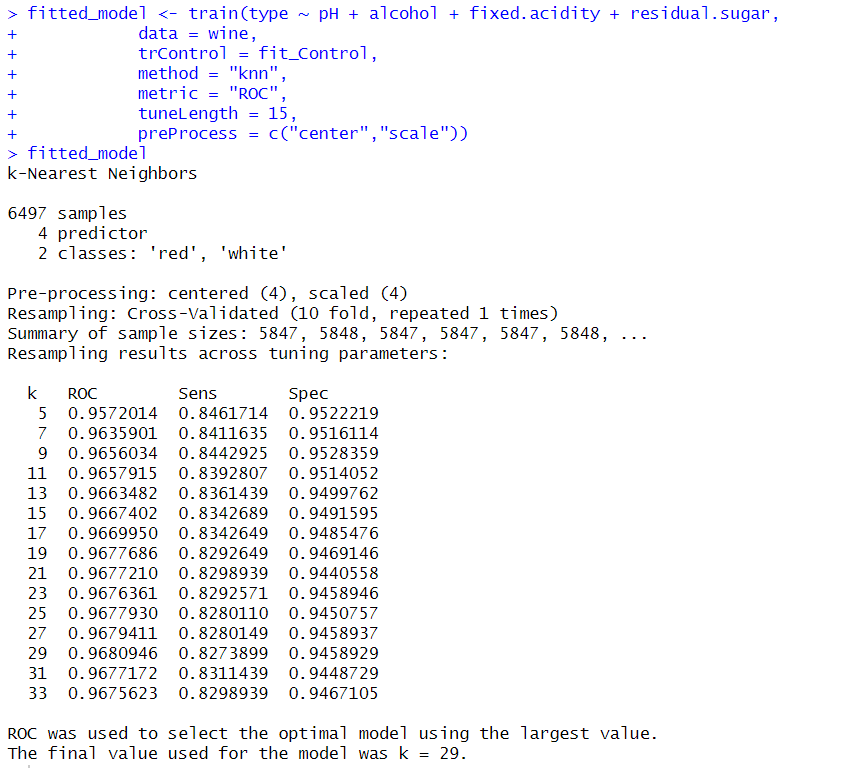
+ trControl = fit\_Control,

+ method = "knn",

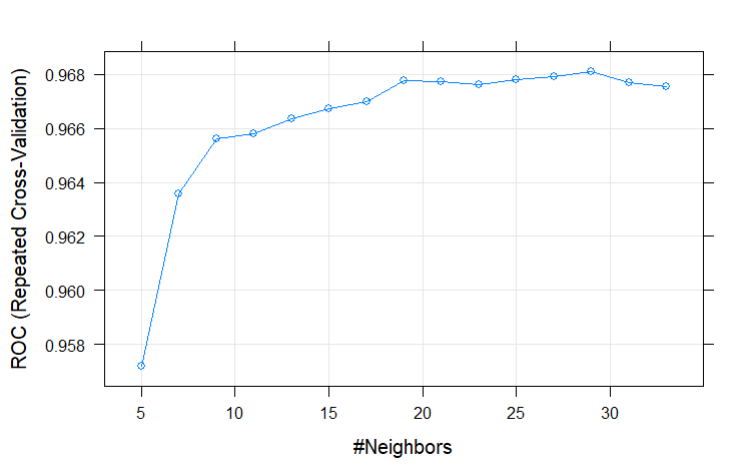
+ metric = "ROC",

+ tuneLength = 15,

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



> plot(fitted\_model)



From the model result, with the help of ROC, we have found out the highest value was obtained at k=29. It appears that the optimal value of the k for the kNN model was determined to be 29 based on maximizing the ROC metric. This means that after evaluating the performance of the kNN model with different values of k (the number of nearest neighbors), the model with k=29 achieved the highest ROC value among the tested values. The ROC metric is commonly used to assess the classification performance of a model, particularly in binary classification tasks.

**Attach the figure that resulted after command: plots$roc**

Before we start plotting, we need to load the “MLeval” library in R, which provides various functions to assess the performance of machine learning models, including metrics calculation, visualization, and comparison. After loading the library, we are using “evalm”, which is used for generating evaluation plots for one or more models. Here, we will generate a set of evaluation plots for the m model (which is our designed kNN model) and store them in the plots object. You can then further explore and analyze these plots to gain insights into the performance of your kNN model. Informedness is one such kind of measurement of a model's ability to make correct predictions across all classes, taking into account both sensitivity and specificity. If the value is close to 1, then it indicates that the model has good overall predictive performance. Here, we have obtained a kNN Optimal Infomedness to be equal to 0.8184. Overall, it seems that kNN model has achieved good performance, with an AUC-ROC of 0.97 indicating high discrimination ability between the classes in your dataset.

> library(MLeval)

> plots <-evalm(fitted\_model,gnames="kNN", plots='r')

\*\*\*MLeval: Machine Learning Model Evaluation\*\*\*

Input: caret train function object

Averaging probs.

Group 1 type: repeatedcv

Observations: 6497

Number of groups: 1

Observations per group: 6497

Positive: white

Negative: red

Group: kNN

Positive: 4898

Negative: 1599

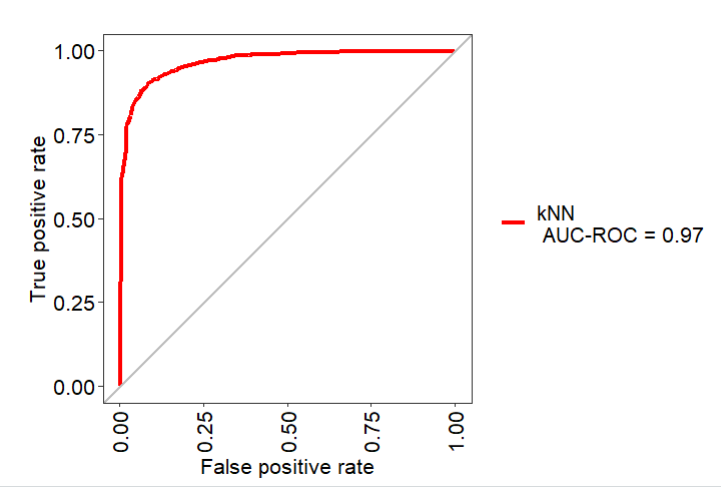
\*\*\*Performance Metrics\*\*\*

kNN Optimal Informedness = 0.81840260513985

kNN AUC-ROC = 0.97

After using “evalm” command, we will use “plots$roc”, where it will return the data associated with the ROC curve, including the false positive rates (FPR) and true positive rates (TPR) at various thresholds.

> plots$roc



**Output after executed command: plots$optres[[1]][13,]**

The command “**plots$optres[[1]][13,]**” displays the part that accesses the optimization results from plots. From the result, the confidence interval(CI) for the AUC-ROC score is reported to be in range of 0.97 to -0.97 which implies that the true AUC-ROC score of the kNN model lies within this interval with a certain level of confidence.

> plots$optres[[1]][13,]

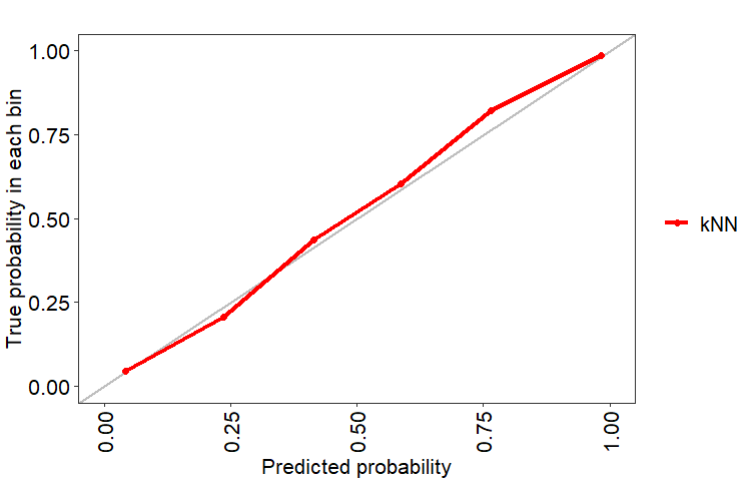
Score CI

AUC-ROC 0.97 0.97-0.97

**Attach the figure that resulted after command: plots$cc**

We are plotting a graph between Predicted Probability and Time Probability in each bin, where it shows that our model lying between range 0.01 to 1 value.

> plots$cc



## References

Thulin, M. (2021). Modern Statistics with R: From wrangling and exploring data to inference and predictive modelling. BoD-Books on Demand.

Wine-quality/winequality-white.csv. (n.d.-c). <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv>

Mleval: Machine learning model evaluation. (n.d.-a). <https://cran.r-project.org/web/packages/MLeval/MLeval.pdf>

Zach. (2023, April 19). R: How to use traincontrol to control training parameters. Statology. <https://www.statology.org/traincontrol-r/>