KNN_ForensicGlass_241026

Ahmad Mudrik

2024-10-26

Forensic Glass KNN Model

Introduction

The data set presents measurement of the refractive index (RI), and the chemical composition (in weight percentages) of oxide elements Na, Mg, Al, Si, K, Ca, Ba, and Fe. This information is used for as input for the task of predicting six possible glass types: (a) WinF: Float glass window (b) WinNF: Non-float glass window (c) Veh: Vehicle window (d) Con: Container (bottles) (e) Tabl: Tableware (f) Head: Vehicle headlamp

Hypothesis

1. The predicted y_f changes with different K-values

Objective

1. To run **1-NN and 5-NN* algorithms using random sample of "test" observations

Install packages, import and inspect data set

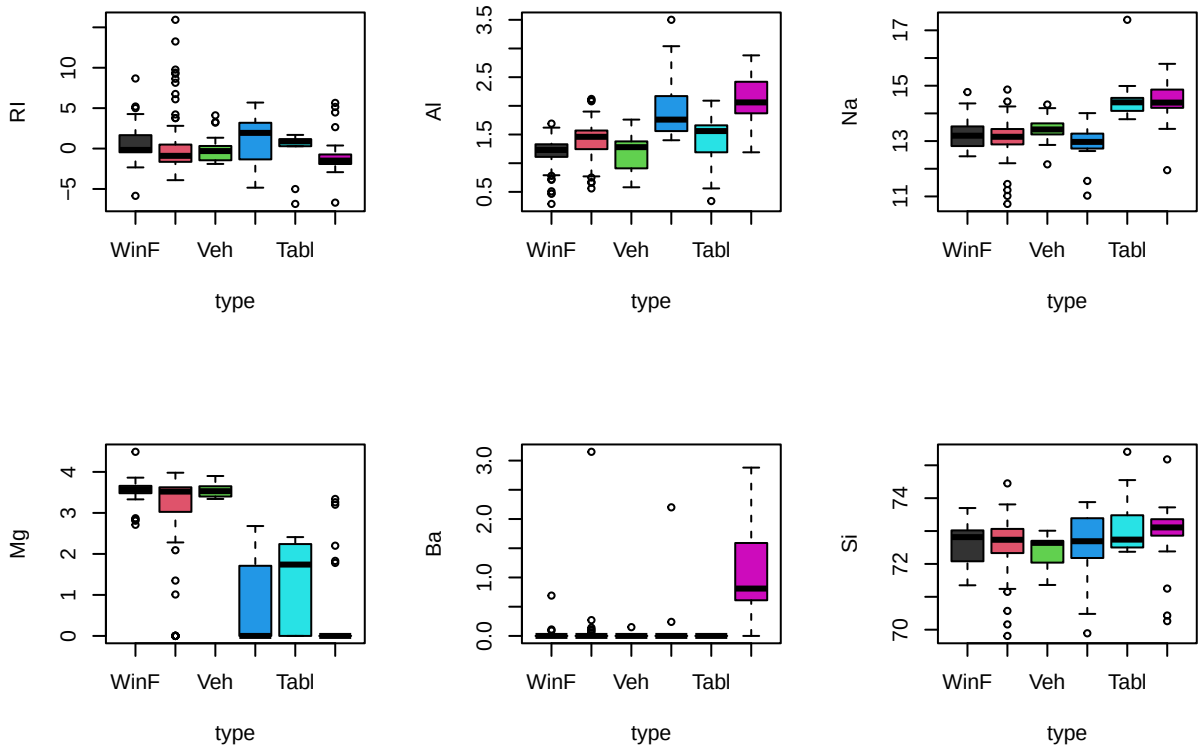
Import dataset

```
library(MASS) ## a library of example data sets
data(fgl) ## loads the data into R
head(fgl)
```

```
##      RI    Na   Mg   Al    Si    K   Ca Ba   Fe type
## 1  3.01 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00 WinF
## 2 -0.39 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00 WinF
## 3 -1.82 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00 WinF
## 4 -0.34 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00 WinF
## 5 -0.58 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00 WinF
## 6 -2.04 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26 WinF
```

- the *forensic glass* - fgl data set was obtained from the "MASS" library
- The data includes measurements for each of the 214 glass shards

Inspect data



- Some of the inputs are clear discriminators, e.g Barium (Ba) is generally present in small amounts except for headlamps, where it is relatively abundant. - Magnsium (Mg) is common for windows of all types - in houses and in vehicles. - Other inputs are more subtle discriminators, or may matter only in interaction.

Standardize data

```
x <- scale(fgl[,1:9]) # convert columns to mean-zero sd-one
apply(x,2,sd) # validate standardization
```

```
## RI Na Mg Al Si  K Ca Ba Fe
##  1  1  1  1  1  1  1  1  1
```

- Before KNN algorithm is applied, the data must be standardized.
- Standardization is a scaling method, where the values are centered around mean with a unit standard deviation.
- "scale()" function is used to scale the data set to obtain standard deviation = 1.
- validation can be made using the "apply()" function, using data set "x", margin = 2 indicating field, and apply function "sd" for standard deviation.

Study 1: Find the nearest neighbours

```
library(class)
test <- sample(1:214,10) # picks 10 samples out of 214 rows
nearest1 <- knn(train=x[-test,], test=x[test,], cl=fgl$type[-test], k=1)
nearest5 <- knn(train=x[-test,], test=x[test,], cl=fgl$type[-test], k=5)
```

```r
data.frame(fgl$type[test],nearest1,nearest5)
```

```
##    fgl.type.test. nearest1 nearest5
## 1            Head    WinNF     WinF
## 2            WinF      Veh     WinF
## 3             Con     WinF    WinNF
## 4             Veh      Veh     WinF
## 5            Tabl     Tabl     Tabl
## 6           WinNF    WinNF    WinNF
## 7             Con    WinNF    WinNF
## 8            WinF     WinF     WinF
## 9            Tabl     Tabl     Tabl
## 10          WinNF    WinNF     WinF
```

Findings from Study 1

- 1-NN and 5-NN algorithms were ran, where a random sample of *test* observation are left out for prediction.
- 1-NN obtained 80% accuracy, 5-NN obtains 70%
- However, if both algorithm is re-run on random test sets, the numbers constantly shifts,
- It can deduced that K-NN predictions will be unstable as a function of K. The predicted y_f also changes with each of these different K.
- The instability of prediction makes it hard to choose the optimal K, and hence cross validation does not work well for KNN.