SupervisedLearning

July 15, 2020

1 Classification with KNN

using wine27 dataset from MBCbook package

```
Warning message:
"package 'MBCbook' is in use and will not be installed"
```

A data.frame: 6 × 29

	Alcohol <dbl>	Sugar.free_extract <dbl>	Fixed_acidity <dbl>	Tartaric_acid <dbl>	Malic_acid <dbl>	Uronic_a <dbl>
1	14.23	24.82	73.1	1.21	1.71	0.72
2	13.20	26.30	72.8	1.84	1.78	0.71
3	13.16	26.30	68.5	1.94	2.36	0.84
4	14.37	25.85	74.9	1.59	1.95	0.72
5	13.24	26.05	83.5	1.30	2.59	1.10
6	14.20	28.40	79.9	2.14	1.76	0.96

we will use the column Type in order to realize the classification

1.1 Training and prediction with split method

KNN train and train at the same time

Here we have choosen k=3 arbitrary

1. Barolo 2. Barolo 3. Barolo 4. Barolo 5. Barolo 6. Barolo 7. Barbera 8. Barolo 9. Barolo 10. Grignolino 11. Grignolino 12. Grignolino 13. Grignolino 14. Grignolino 15. Grignolino 16. Grignolino 17. Grignolino 18. Grignolino 19. Grignolino 20. Grignolino 21. Barbera 22. Barbera 23. Barbera 24. Barolo 25. Barbera 26. Barbera 27. Barbera 28. Grignolino

Levels: 1. 'Barbera' 2. 'Barolo' 3. 'Grignolino'

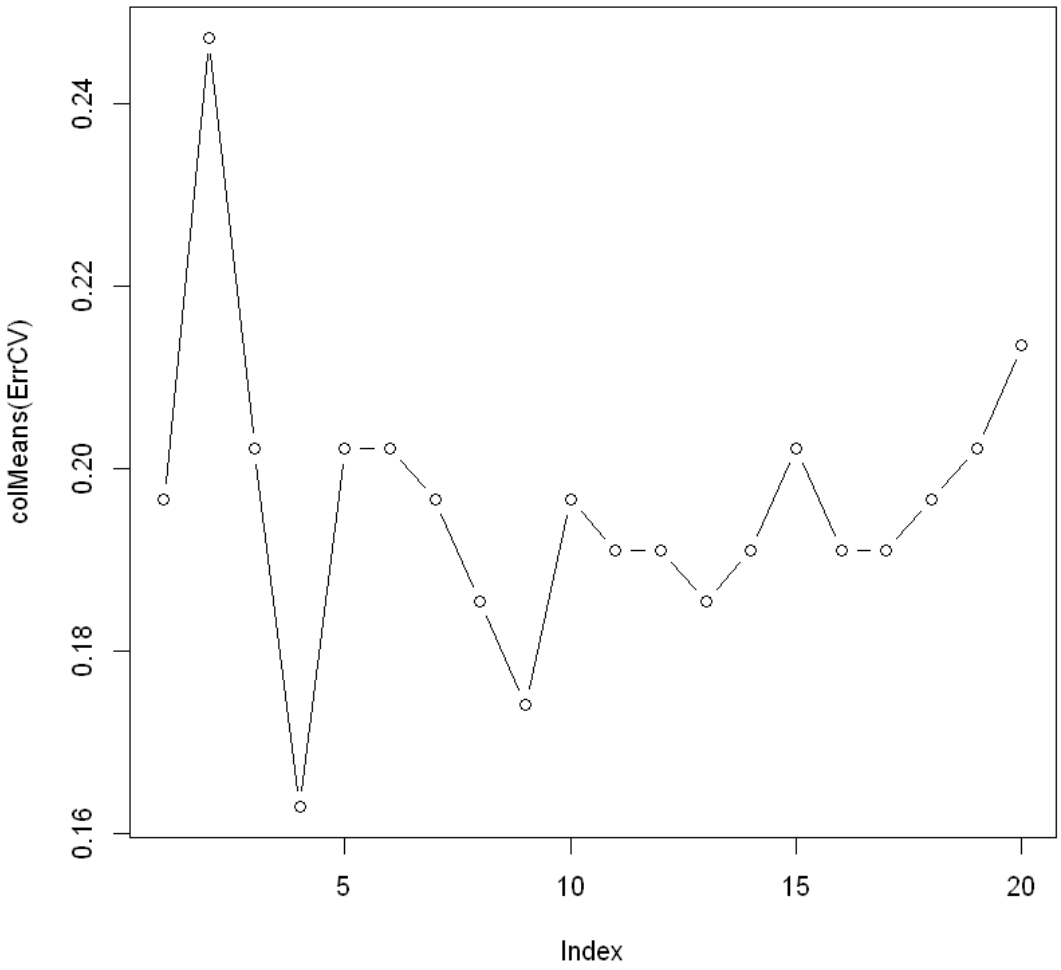
1.2 Calculation of the validation error

0.107142857142857

0.107142857142857

1.3 Training and prediction with "leave one out" - searching for best k

4



here the smallest error is for k=9

0.162921348314607

1.3.1 16% of error is quit high

Best solution seems to be 9

But there are random in the search of the neighboor done by R, so if the algo is run several times, the result changes

2 Classification with LDA algorythm

Learning, prediction and calculation erro value - with split method

0

- yhat$class : gives the result
- yhat$posterior : gives the probability for each class

Here LDA perfectly classify : err =0

2.1 LDA with 'leave one out' method

0.0112359550561798

1.1% really better then KNN

3 Classification with Logistic Regression

We will use banknote dataset with binary classification

A data.frame: 6 × 7

	Status <fct>	Length <dbl>	Left <dbl>	Right <dbl>	Bottom <dbl>	Top <dbl>	Diagonal <dbl>
1	genuine	214.8	131.0	131.1	9.0	9.7	141.0
2	genuine	214.6	129.7	129.7	8.1	9.5	141.7
3	genuine	214.8	129.7	129.7	8.7	9.6	142.2
4	genuine	214.8	129.7	129.6	7.5	10.4	142.0
5	genuine	215.0	129.6	129.7	10.4	7.7	141.8
6	genuine	215.7	130.8	130.5	9.0	10.1	141.4

Training, classification and calculation of the error

with split method

```
Warning message:
"glm.fit: algorithm did not converge"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
```

0.02

The error is 4%

4 Comparing Knn, LDA and Logistic Regression with leave one out method

0.005

0.025

0.005

ErrCV.LDA 0.005 **ErrCV.LReg** 0.025 **ErrCV.kNN** 0.005

4.0.1 Comparaison of the 3 methods

KNN and LDA are best

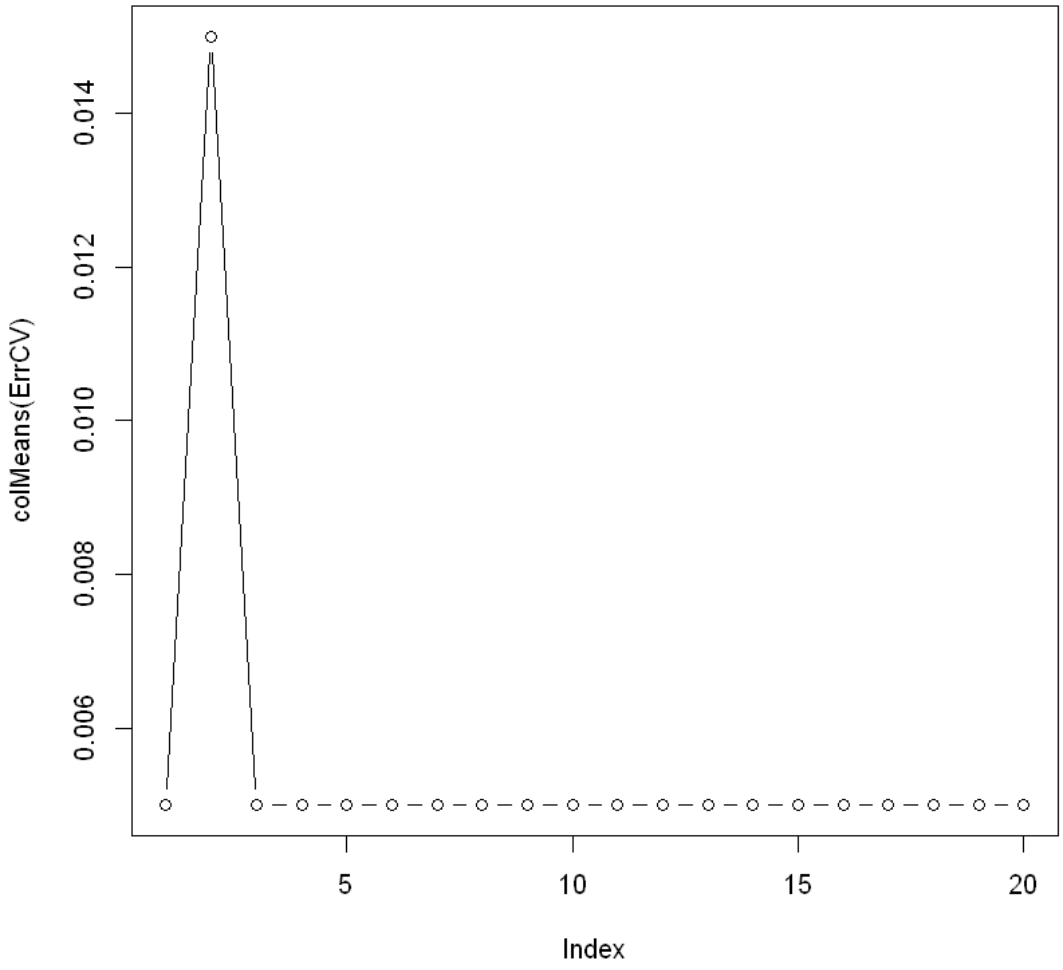
Logistic Regression is 3rd

We can even calculate standard deviation (see below) and even confidence interval (to be done) with the errors calculated with leave one out methos

ErrCV.LDA 0.0707106781186548 **ErrCV.LReg** 0.156516732131699 **ErrCV.kNN** 0.0707106781186548

4.0.2 Confirmation that the k=1 is the best parameter for Knn on banknote dataser

1



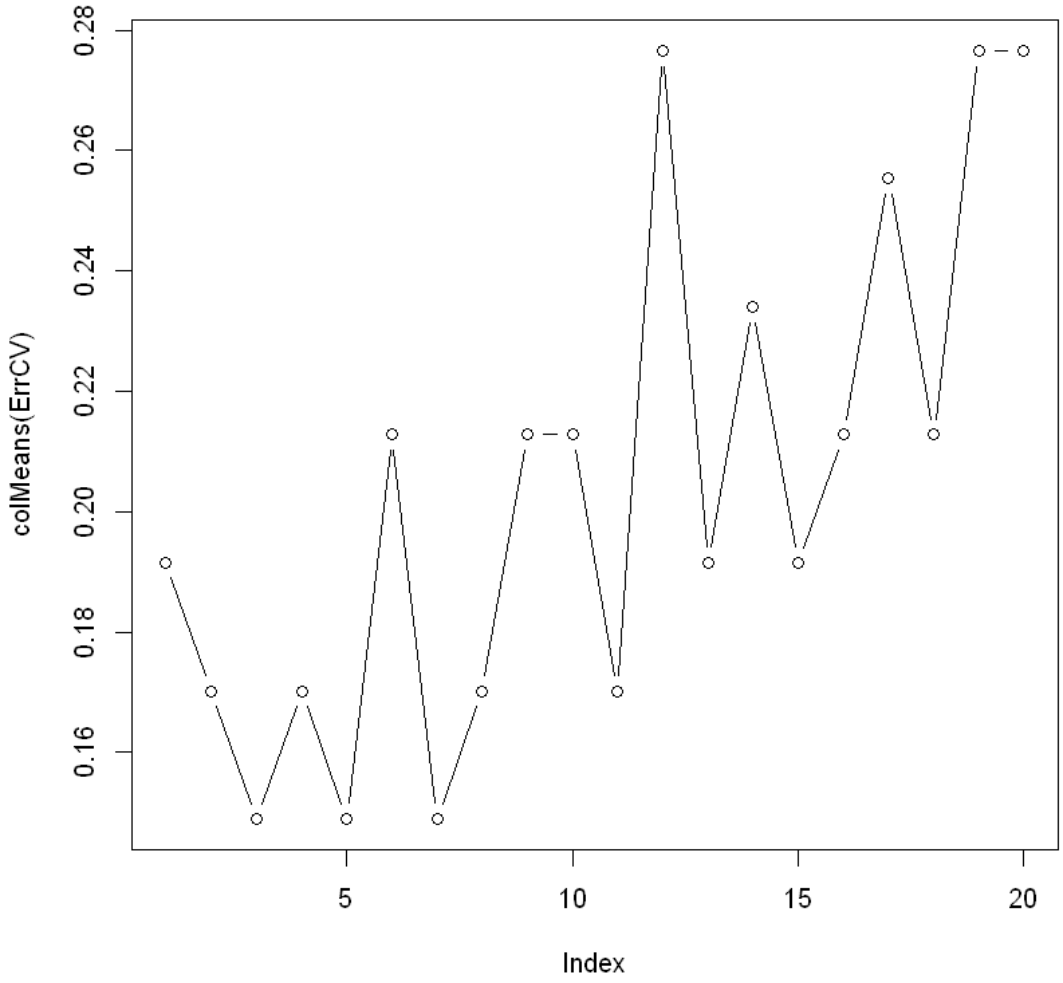
4.0.3 Comparison of Knn, Lda and Logistic Regression on swiss dataset

The Classification is done on $Catholic variable.

The numerical variable is trnasformed into a boolean

4.0.4 find best k for Knn - using leave one out method

3



K = 3

ErrCV.LDA 0.148936170212766 **ErrCV.LReg** 0.127659574468085 **ErrCV.kNN** 0.148936170212766

For swiss dataset, Logistic regression is best method.

5 Classification with SVM

```
Installing package into 'C:/Users/erick/R'
(as 'lib' is unspecified)


package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\erick\AppData\Local\Temp\RtmpyudMa2\downloaded_packages
Warning message:
"package 'e1071' was built under R version 4.0.2"
```

5.1 Test SVM with linear kernel on Iris Dataset and compare with LDA

ErrCV.LDA 0.140469013039037 **ErrCV.SVM** 0.19661566092457

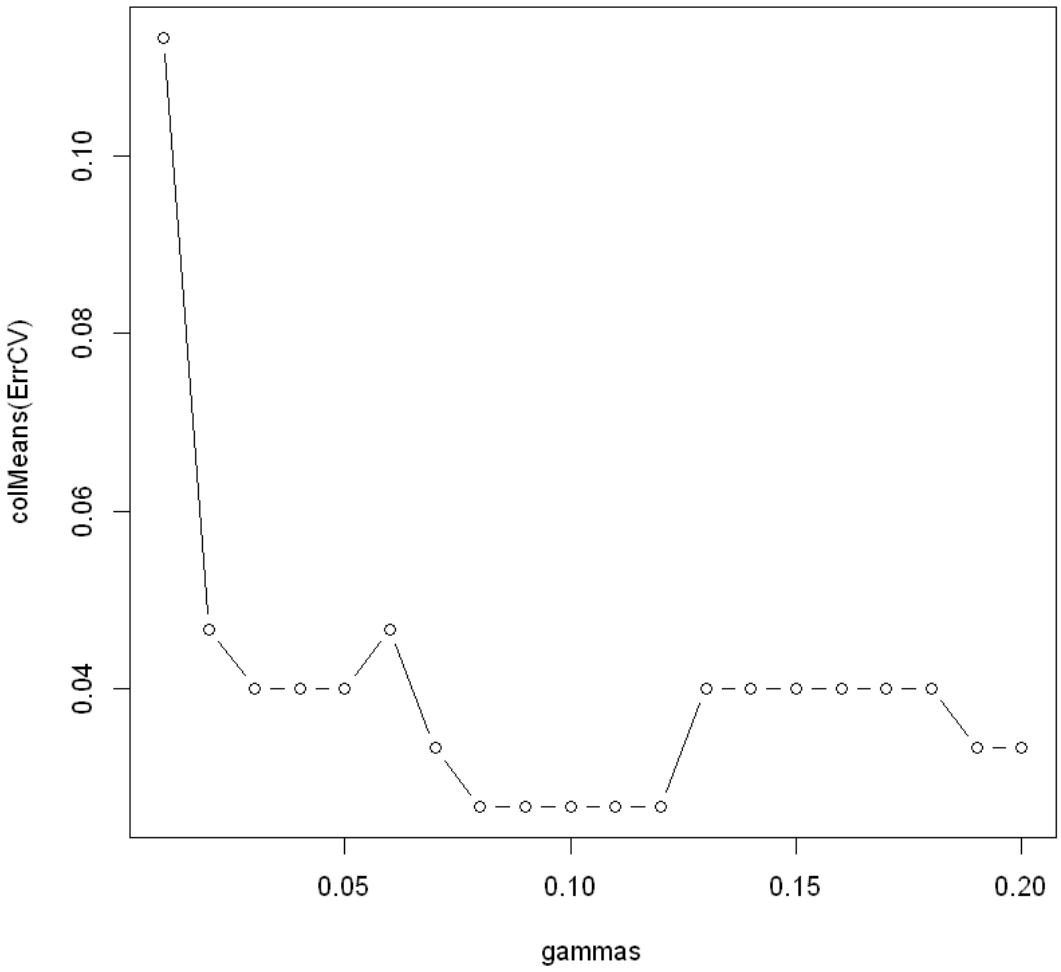
ErrCV.LDA 0.02 **ErrCV.SVM** 0.04

LDA is better then SVM

ErrCV.LDA 0.140469013039037 **ErrCV.SVM** 0.180106853693033

ErrCV.LDA 0.02 **ErrCV.SVM** 0.0333333333333333

8



We find a minimum for gamma : 0.1

We put this gamma = 0.1 in the comparaison with LDA

ErrCV.LDA 0.140469013039037 **ErrCV.SVM** 0.180106853693033

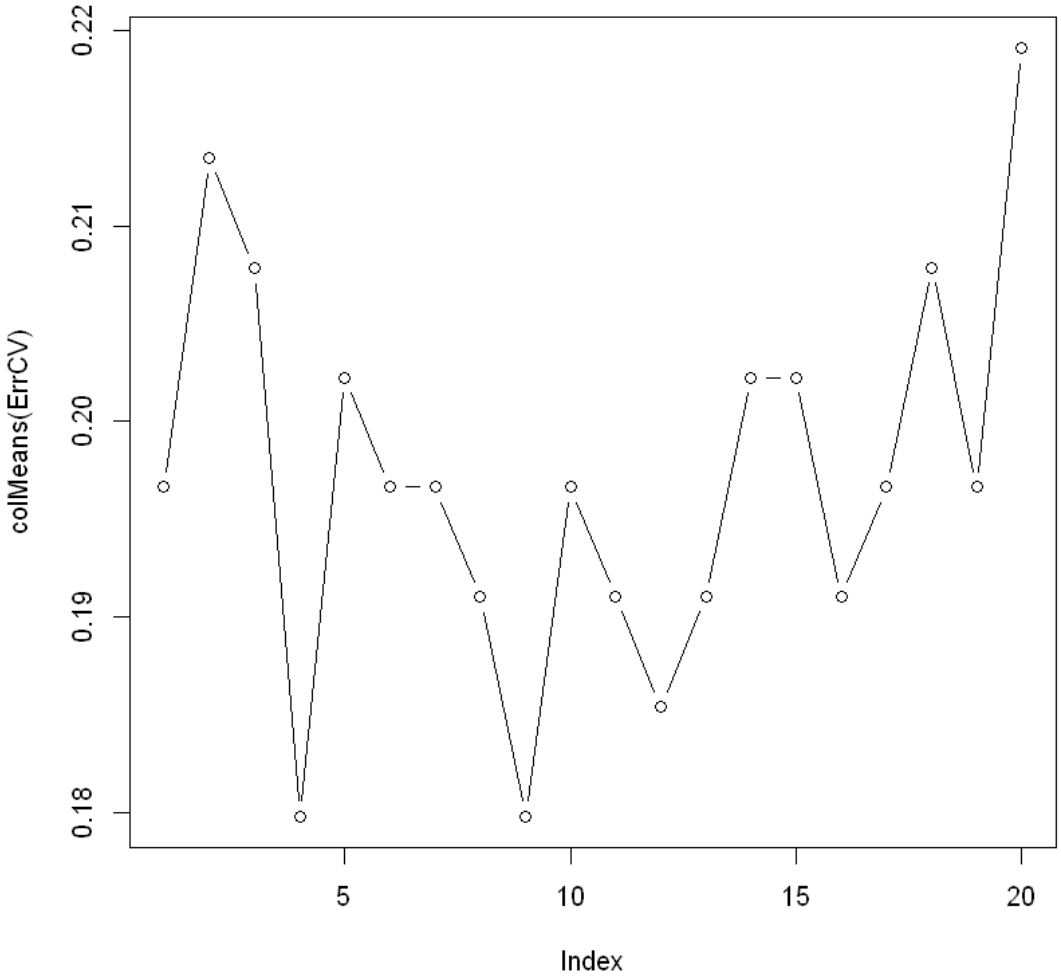
ErrCV.LDA 0.02 **ErrCV.SVM** 0.0333333333333333

LDA is still better then SVM even with RBF

6 Comparison of KNN , LDA, QDA, SVM on the wine dataset

6.0.1 Best model for KNN

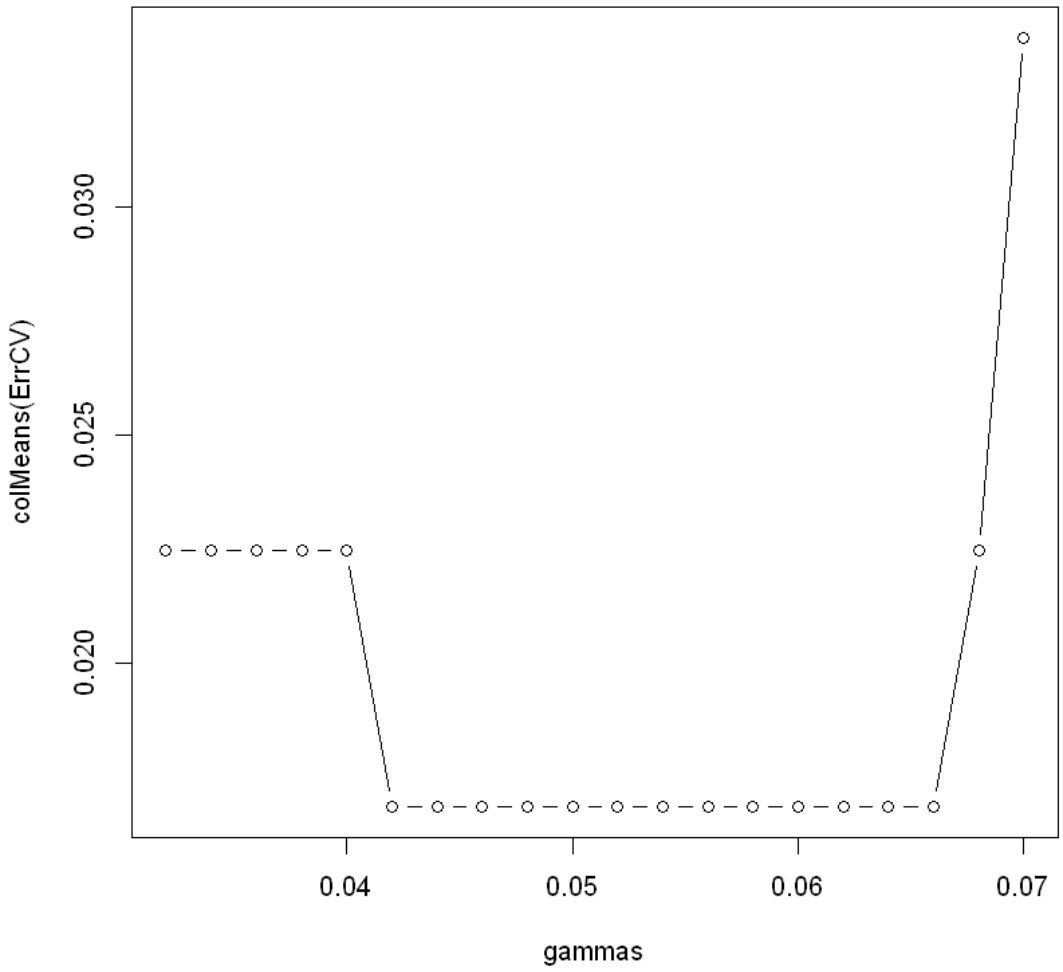
4



Best K=9

6.0.2 Best model for SVM

0.042



Gamma=0.05 is the best value

We can now compare the four methods

ErrCV.KNN 0.38031449664559 **ErrCV.LDA** 0.105699929442318 **ErrCV.QDA** 0.180985263453982 **ErrCV.SVM** 0.129087151444531

ErrCV.KNN 0.174157303370787 **ErrCV.LDA** 0.0112359550561798 **ErrCV.QDA** 0.0337078651685393 **ErrCV.SVM** 0.0168539325842697

For this dataset the quality order is :

- 1) LDA
- 2) SVM
- 3) QDA

- 4) KNN

2