POLITECNICO DI TORINO

POLITECNICO DI TORINO

Master Degree course in Ingegneria Matematica 2022/2023

Apprendimento statistico - Machine Learning

Tesina Machine Learning

Racca Riccardo s315163 Filippo Scaramozzino s312856

Gennaio 2023

Contents

Chapter 1

Introduzione

1.1 Dataset

Questo elaborato si occuperà di analizzare il dataset Glass Identification attraverso i metodi di supervised e unsupervised learning.
Questo dataset nasce con l'idea di identificare i pezzi di vetro trovati sulla scena del crimine.
Il dataset fu creato da B. German presso il Central Research Establishment,Home Office Forensic Science Service situato in Aldermaston, Reading, Berkshire RG7 4PN, risalente al settembre 1987.
Ed è accessibile all: http://archive.ics.uci.edu/ml/datasets/Glass+Identification
Dal punto di vista strutturale esso è formato da un numero di istanze raccolte pari a 214 e numero di attributi pari a 11, gli attributi scelti per lo studio dei frammenti di vetro includono sia proprietà chimiche che fisiche; di seguito gli elenchiamo nel dettaglio.

ID number	number from 1 to 2014
RI	refractive index
Na	Sodium
Mg	Magnesium
Al	Aluminium
Si	Silicon
K	Potassium
Ca	Calcium
Ba	Barium
Fe	Iron
Type	(*)

(*)building windows float processed, building windows non float processed, vehicle windows float processed, containers, tableware, headlamps.

1.2 Analisi preliminare

L'analisi preliminare sarà formata dalla rappresentazione di alcuni grafici indicativi dello stato del dataset; tutto questo sarà effettuato prima della suddivisione del dataset nella parte di training set e di test set. Escluderemo dalla rappresentazione la covariata che indica l'ID number, perchè consideriamo ragionevole ipotizzare che essa non dia informazioni di alcun tipo sulle altre covariate essendo solamente un codice identificativo; manteniamo invece la variabile di risposta.

I primi grafici che verranno rappresentati sono i box plot.

In breve essi forniscono un riepilogo visivo della variabilità dei valori assunti dalla feature in analisi, mostrando la mediana, il primo quartile e il terzo quartile, e il minimo e massimo valore che non sono da considerarsi come outliers.

I primi che verranno mostrati avranno la funzione di indicare i valori assunti dalla variabile.

A seguire, invece, verranno mostrati dei box plot con le classi sull'asse x e sull'asse y i valori assunti dalla variabile; all'interno di queste figure vi saranno tanti box plot quante sono le classi e nello specifico per ciascuna classe verrà quindi indicata l'intervallo di valori che la variabile assume.

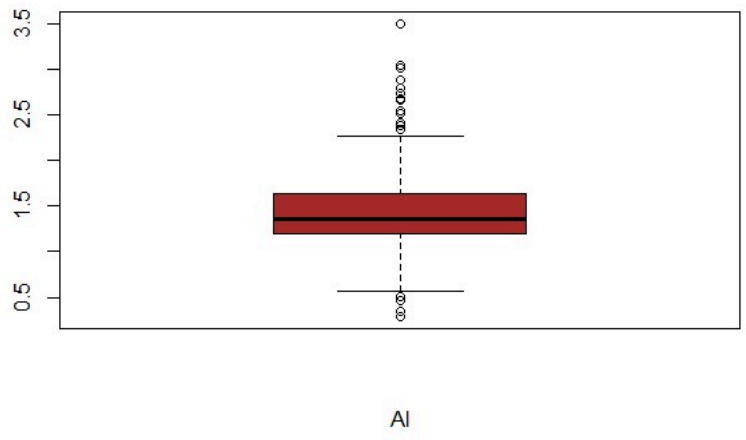

Figure 1.1. Boxplot di Al

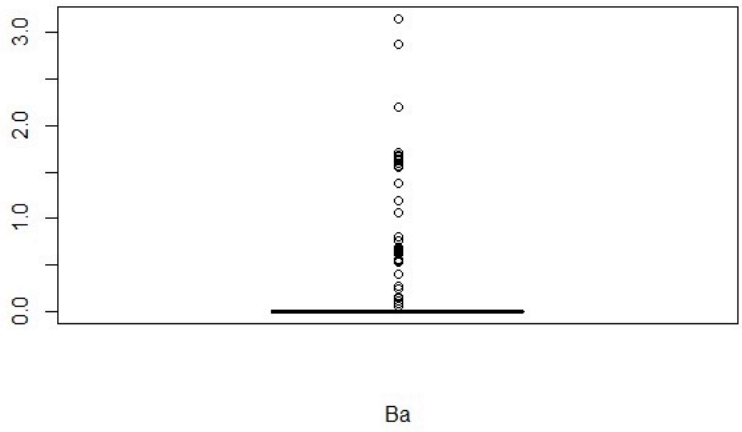

Figure 1.2. Boxplot di Ba

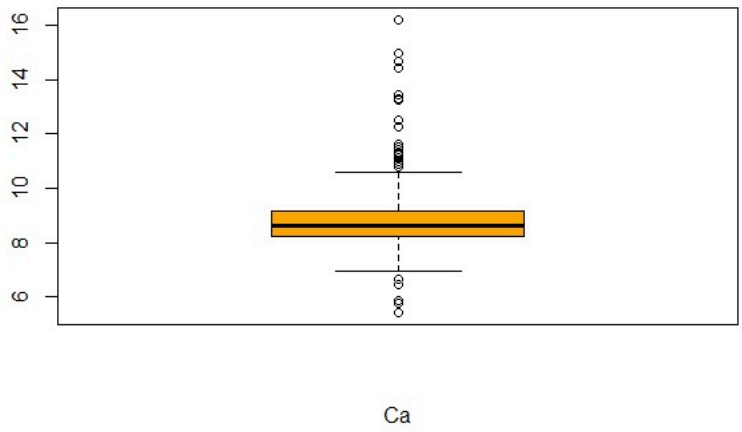

Figure 1.3. Boxplot di Ca

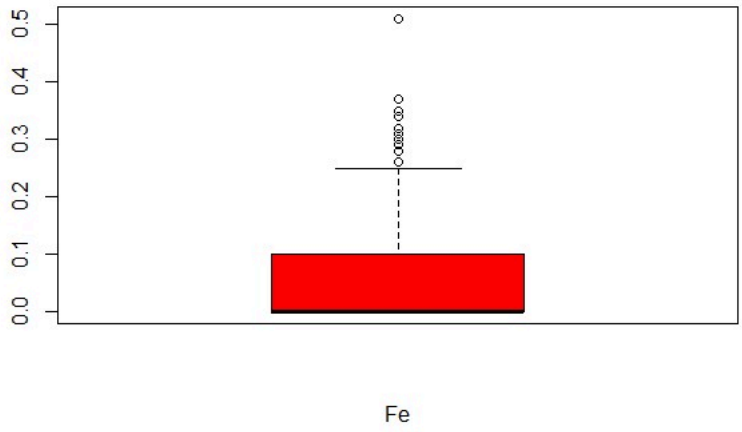

Figure 1.4. Boxplot di Fe

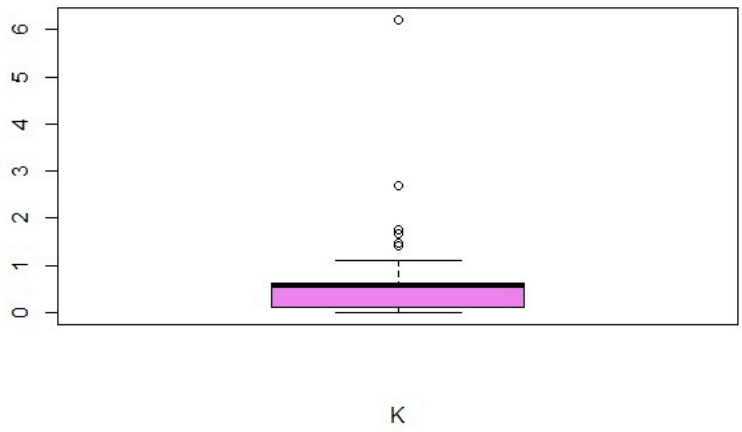

Figure 1.5. Boxplot di K

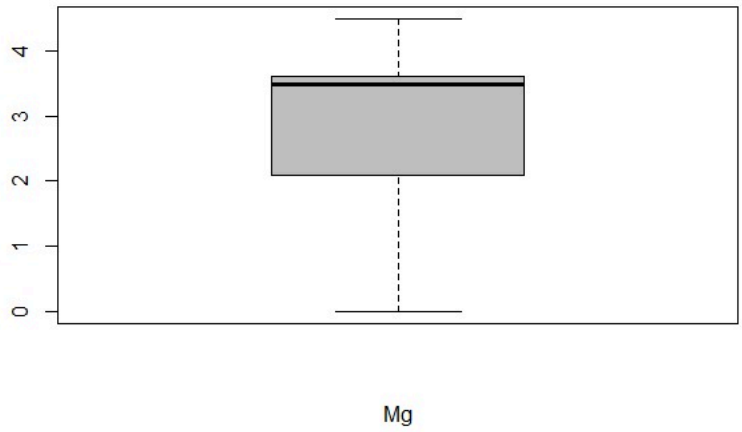

Figure 1.6. Boxplot di Mg

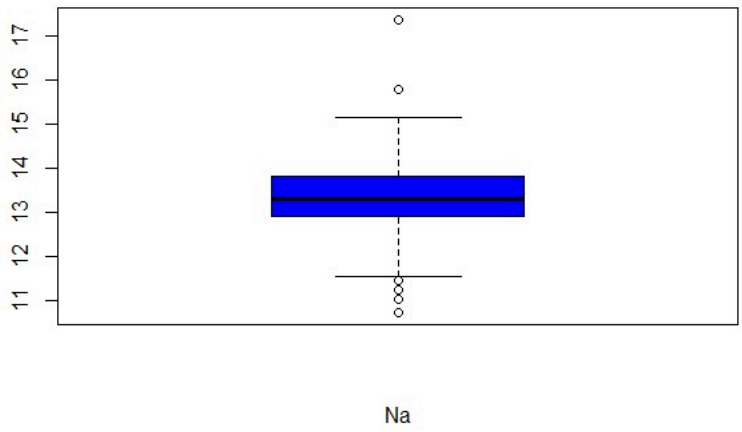

Figure 1.7. Boxplot di Na

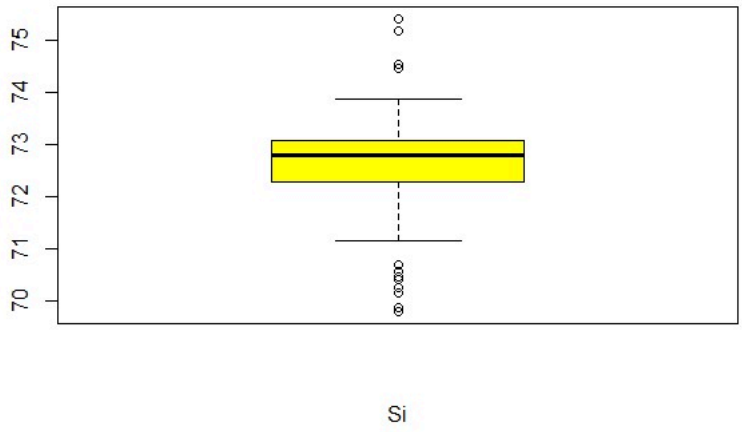

Figure 1.8. Boxplot di Si

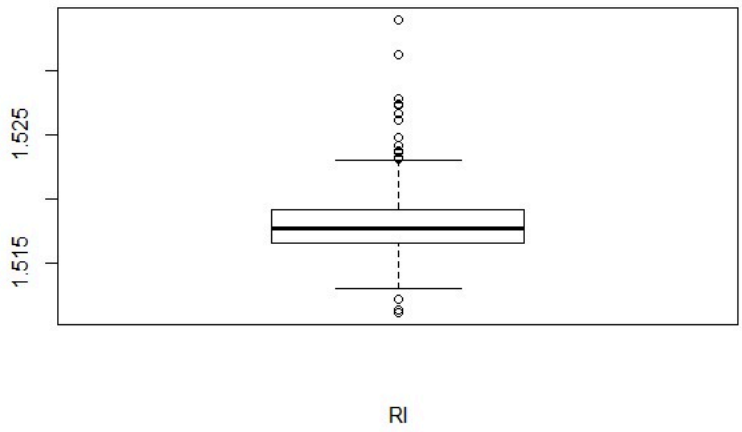

Figure 1.9. Boxplot di RI

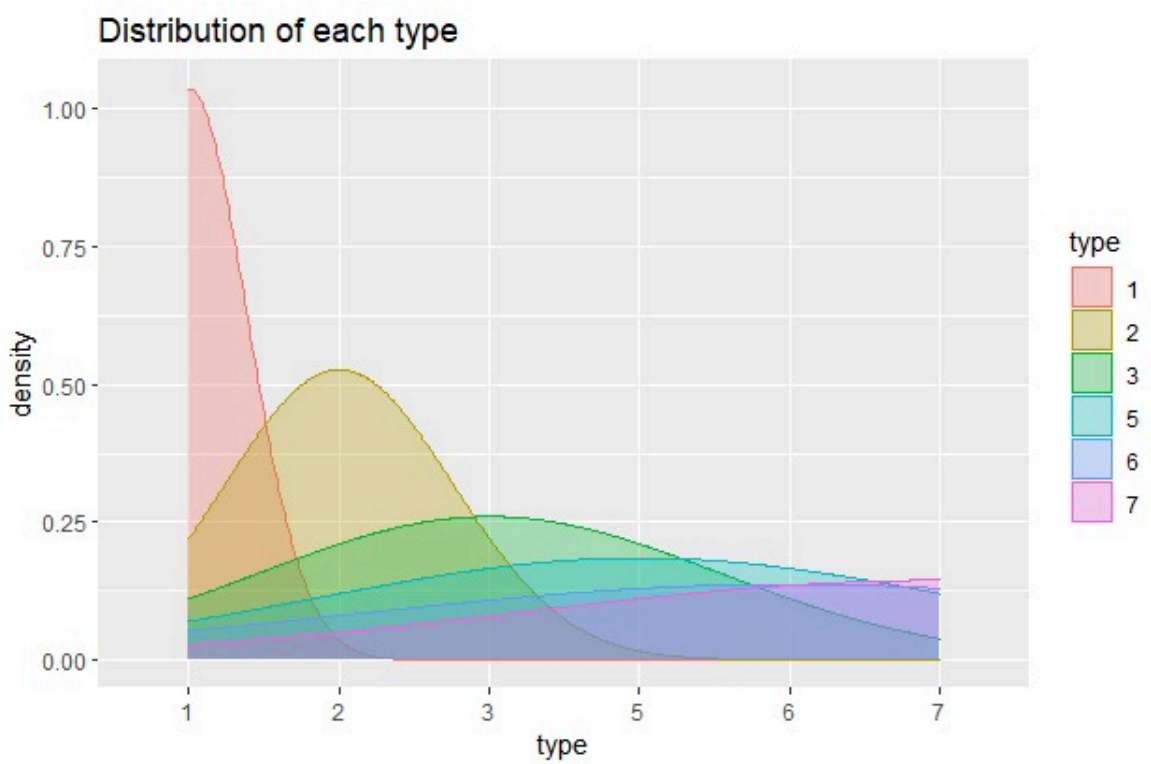

Figure 1.10. Distribuzione delle classi nel dataset

Adesso si andrà ad indagare che genere di correlazione intercorre tra le variabili del dataset, attraverso il correlation plot e la heatmap.

L'ultima figura mostrerà invece una tabella rappresentativa di tutte le quantità mostrate fin ora insieme ad uno scatter plot.

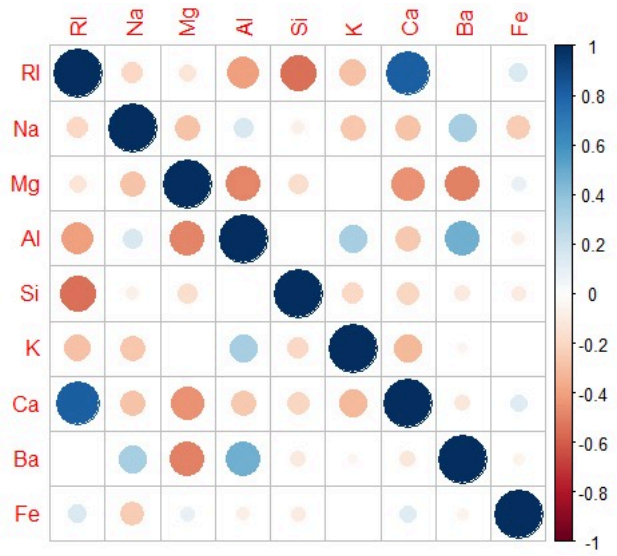

Figure 1.11. Correlation plot del dataset

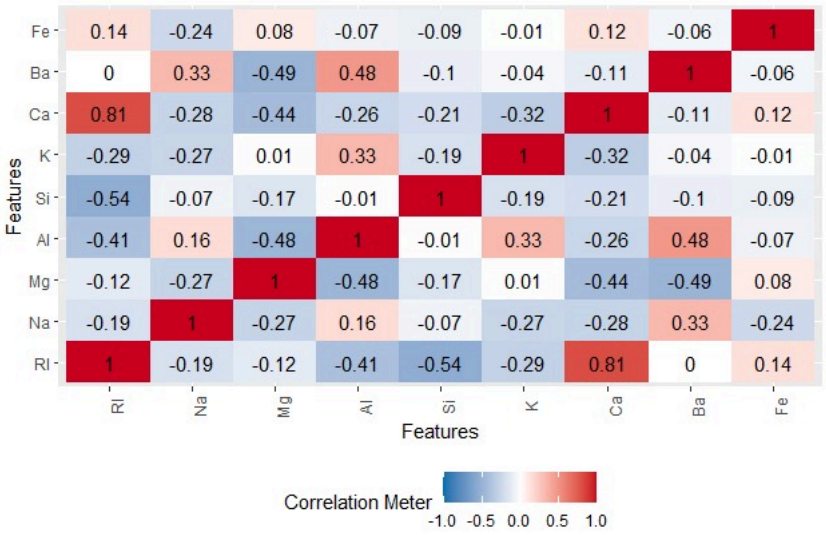

Figure 1.12. Heatmap del dataset

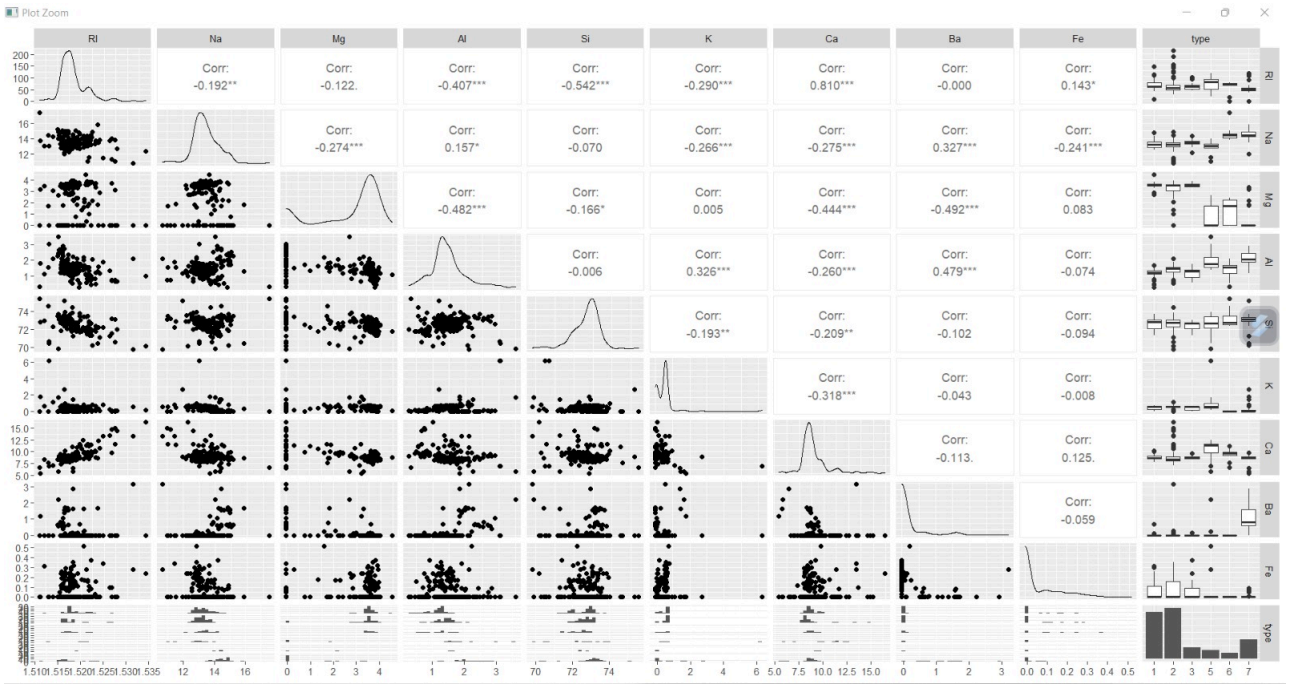

Figure 1.13. Correlazioni tra le variabili e scatterplot

Una volta terminata la descrizione procediamo a suddividere il data set in due parti training set e test set, nello specifico riserveremo l'80 % dei dati per il training set e il restante 20 % per il test set.

1.2.1 Statistical Learning

In questa sezione verranno ripercorsi alcuni concetti teorici che saranno utitilizzati in seguito.

- Prediction function : è una funzione che preso in ingresso un vettore di features restituisce in uscita una stima della variabile di risposta

- Loss : è tipicamente una funzione che prende in ingresso due valori quali valore vero della variabile di risposta e il valore stimato (e. g. squared error loss e zero-one loss)

- Risk : è definito come il valore atteso della funzione di loss.

- Training set : campioni di coppie osservate. Sarà l'insieme di coppie su cui alleneremo la nostra funzione predittiva.

- Test set : campioni di coppie su cui verrà testata la nostra funzione predittiva

- Training loss : siccome non è tipicamente possibile calcolare il valore della funzione di loss effettuiamo una media campionaria sul training set questo valore è chiamato training

loss.

- Learner : a seguito di una scelta opportuna di una collezione di funzioni G il learner è quella funzione appartenente a G che minimizza la training loss.

- Generalization risk : è il valore atteso della funzione di loss, a training set fissato, calcolato utilizzando il learner fornito dal training set.

- Test loss : media campionaria della funzione di loss calcolata con il learner del training set valutata nei punti del test set.

- Overfitting : fenomeno che avviene tipicamente quando la minimizzazione della training loss raggiunge i valori minimi, questo combacia con la ridotta abilità nel predire nuovi dati.

- In-sampe risk : siccome la minimizzazione della training loss non è un indicatore ottimale del generalization risk (visto che si vuole evitare overfitting) un'alternativa è quella di valutare il valore atteso della funzione di loss in nuovi punti ottenuti dalla densità condizionata $f(y|x_i)$, che si traduce operativamente nel calcolare la media campionaria.

1.2.2 K-Fold Cross Validation

Quando vi è una abbondanza dei dati il modo più semplice per stimare il generalization risk fissato un training set τ è quello di ottenere un test set τ' e valutare la test loss.
Quando non è possibile avere un test set sufficientemente grande/indicativo si può ottenre comunque una buona conoscenze dell'expected generalization risk attraverso un metodo chiamato Cross Validation.

1. Effettuare più copie identiche del data set, partizionare ciascuno di essi in diversi training set e data set.

2. Per ciascuno di questi si stimano i parametri del modello attraverso il training set e lo si valuti sul test set attraverso la minimizzazione della training loss.

3. La media pesata della loss tra risposte osservate e previste è quindi una misura della potenza predittiva del metodo.

Viene indicato con K il numero di partizioni del training set.
La k cross validation loss è la media campionaria della funzione di loss calcolata sui learners dei diversi training set.

Questo è una stima del valore atteso del generalization risk.

Chapter 2

Unsupervised Learning

Come già anticipato procederemo nella analisi di due metodi descrittivi dei dati unsupervised e supervised learning. Entrando più nello specifico la differenza tra i due metodi si basa sul fatto che nell'unsupervised learning siamo focalizzati caratteristiche dei dati come ad esempio il ragruppamento dei dati, lo studio della relazione tra le variabili e tecniche di pre processing. Mentre invece vedremo che nel supervised learning vi sarà una distinzione tra variabili esplicative e la variabile di risposta.

2.1 Analisi della varianza

Si procede nell'analizzare la varianza e la correlazione delle variabili, cercando di individuare le varibili che variano poco, intuitivamente si comportano approssimativamente in modo costante (varianza intorno allo zero). Si analizzeranno inoltre anche quelle variabili che sono correlate fortemente (correlazione prossima a uno) questo perchè forniscono approssimativamente la stessa informazione.

Di seguito viene riportata la tabella delle varianze per le rispettive variabili, mentre la correlazione si può osservare nella figura 1.12 (Heatmap).

Varianza di	valore
RI	1.02e-5
Na	0.66
Mg	2.10
Al	0.22
Si	0.58
K	0.29
Ca	2.1
Ba	0.25
Fe	0.008

2.2 Pca

La PCA(principal component analysis) è un metodo di pre-processing che viene effettuato prima di applicare le tecniche di supervised learning. Nello specifico essa costruisce, come suggerito dal nome, delle componenti principali ovvero quelle direzioni che meglio descrivono la varianza dei dati.

Il fondamento matematico su cui si basa è la decomposizione SVD. Riassumendo in poche parole il meccanismo di riduzione di dimensione offerto da questo metodo, si può dire che le componenti che meno descrivono la varianza dei dati, sotto un'opportuna soglia possono essere rimosse senza perdere troppa informazione sul dataset di partenza, si tratta di un trade-off tra facilità di gestione dei dati e accuratezza rispetto al dataset di partenza.

La rimozione di queste componenti avviene tramite prodotto matriciale.
Di seguito vengono riportati, nell'ordine, la tabella con la deviazione standard delle varie componenti, la proporzione di varianza descritta da ciascuna componente e la proporzione cumulata (per vedere da quale componente in poi possiamo escludere data una soglia fissata).

2.2.1 Scaling

Un passaggio critico nell'applicazione della PCA è la standardizzazione dei dati o scaling, questo perchè la PCA ha l'obiettivo di valutare la varianza dei dati, perciò una variabile con intervallo ampio (esempio [1,100]) dominerà una variabile con intervallo ridotto (esempio [0,1]). Questo comporterà una distorsione dei risultati. Quindi prima di applicare la PCA verrà effettuato uno scaling dei dati.

2.3 Data reduction

A seguito delle analisi condotte sulla varianza, correlazione e con il supporto della PCA, concludiamo che la dimensione del dataset può essere ridotta rimuovendo delle variabili. Nonostante la PCA ci suggerisca che è possibile ridurre la dimensione del dataset di tre componenti principali, mantenendo comunque una proporzione della varianza cumulata circa il 96 %, si decide di escludere solamente due variabili. Questa scelta verrà fatta basandosi sui risultati ottenuti unicamente dalla varianza siccome la correlazione tra variabili non è mai sufficientemente vicina a 1 per trarre conclusioni certe. Concludiamo rimuovendo quindi la variabile RI e la variabile Fe dal dataset, il criterio di scelta è stata la varianza nell'ordine dei millesimi o inferiore.

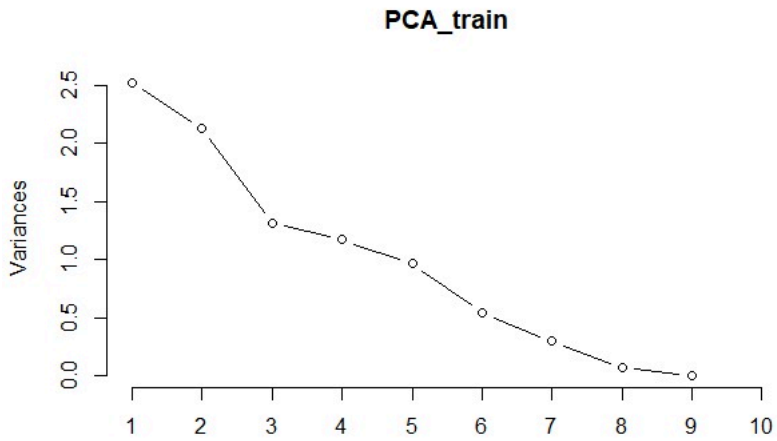

Figure 2.1. Plot della varianza delle componenti principali

```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7     PC8     PC9
Standard deviation     1.5877 1.4588 1.1469 1.0814 0.9803 0.73615 0.54265 0.25933 0.04136
Proportion of Variance 0.2801 0.2364 0.1462 0.1299 0.1068 0.06021 0.03272 0.00747 0.00019
Cumulative Proportion  0.2801 0.5165 0.6627 0.7926 0.8994 0.95962 0.99234 0.99981 1.00000
```

Figure 2.2. Proporzione di varianza raccolta dalle componenti principali calcolata tramite R

Chapter 3

Supervised Learning

Per supervised learning si intende un approccio di indagine del dataset attraverso l'analisi della relazione che intercorre tra le variabili esplicative (features) e la variabile di risposta. Uno degli scopi principali delle tecniche che andremo a vedere sarà quello di inferire qualcosa sulla variabile di risposta. I metodi che verrano descritti riguarderanno la trattazione di problemi di classificazione.

3.1 Classificazione

Per classificazione si intende la situazione in cui la variabile di risposta è categoriale, ovvero può assumere solamente un numero finito di valori.

L'obiettivo è quello di assegnare al vettore di features in input una classe.

I metodi che verranno analizzati sono :

- Classificazione Bayesiana
- Linear and quadratic discriminant analysis
- Logistic classification
- K-nearest neighbors
- Support vector machines
- Classification Trees
- Random forests

Per valutare la bontà dei risultati del modello verranno analizzati i seguenti parametri

- Confusion matrix : a seguito dell'immissione del test set nel learner la confusion matrix M è la matrice che ha come entrata (j, k) il numero di volte che la classe osservata era j ma è stata stimata la classe k .
- True positive : somma degli elementi sulla diagonale principale di M
- True negative(j) : somma degli elementi della sotto-matrice di M, formata dalla rimozione della riga j-esima e della colonna j-esima.
- False positive(j) : somma degli elementi presenti sulla colonna j-esima privata dell'elemento sulla diagonale principale.
- False negative(j) : somma degli elementi presenti sulla riga j-esima privata dell'elemento sulla diagonale principale.
- accuracy : somma della diagonale maggiore di M diviso la somma di tutte le entrate di M, verrà calcolata in percentuale.
- precision(j) : $\frac{truepositive(j)}{truepositive(j)+truenegative(j)}$
- recall(j) : $\frac{truepositive(j)}{truepositive(j)+falsenegative(j)}$
- specificity(j) : $\frac{truenegative(j)}{truenegative(j)+falsepositive(j)}$
- F1 score(j) : $\frac{2truepositive(j)}{2truepositive(j)+falsenegative(j)+falsepositive(j)}$

Per alcuni modelli verrà confrontata l'accuracy del modello con la versione stimata dalla K Fold cross validation.

3.2 Naive Bayes

La classificazione bayesiana si basa sull'utilizzo della formula di Bayes

$$f(y|x) \propto f(x|y)f(y)$$

In questo contesto chiameremo posterior $f(y|x)$ e prior $f(y)$ e infine $f(x|y)$ la verosimiglianza di ottenere il vettore delle features x dato y

Imponendo varie ipotesi riguardo la prior (ad esempio l'equiprobabilità delle classi) e riguardo alla funzione di verosimiglianza sarà possibile ottenere la posterior pdf attraverso la fomula di Bayes.

La classe stimata $\hat{y}$ verrà poi assegnata al vettore x in accordo alla più alta posterior probability, ovvero si classifica secondo la bayes optimal decision rule :

$$\hat{y} = argmax_y f(y|x)$$

Introduciamo ora il metodo naive bayes cominciando con la descrizione delle ipotesi supposte dal modello, ricordando che le densità non sono note e che quindi si deve procedere con delle approssimazioni, e infatti le ipotesi saranno proprio su queste.

Innanzitutto la classe di funzioni, approssimanti, G è scelta in modo tale che rispetti la seguente condizione:

$$g(x|y) = g(x_1|y) * (x_2|y) * ... * g(x_p|y)$$

ovvero ipotizziamo che, condizionatamente alla classe, tutte le features del vettore sono indipendenti.

Come seconda ipotesi, si assume inoltre una distribuzione uniforme sulle classi, ovvero una uniform-prior probability, che comporta:

$$g(y|x) \propto \Pi_j g(x_j|y)$$

Se si fa un ulteriore assunzione sulla distribuzione delle variabili aleatorie $X_i|y$, in cui si richiede che siano distribuite come una normale e che siano omoschedastiche (ovvero a varianza uguale) giungiamo, tralasciando la riscrittura della densità della della distribuzione normale seguita dalla variabile aleatoria $Y|x$, alla seguente Bayes decision rule:

$$\hat{y} = argmin_y |||x - \mu_y|||$$

dove μ_y è la media della variabile aleatoria $X_i|y$, ipotizzata normale.

```
Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        1         2         3         5         6         7
0.33333333 0.36158192 0.08474576 0.05084746 0.03389831 0.13559322

Conditional probabilities:
   Na
Y         [,1]      [,2]
  1 -0.1726391 0.6515257
  2 -0.3725816 0.8464028
  3  0.1523060 0.5363769
  5 -0.7963286 1.2328413
  6  1.2727338 0.5548701
  7  1.3032038 0.9633086

   Mg
Y         [,1]      [,2]
  1  0.5703226 0.1882951
  2  0.1919093 0.8499514
  3  0.5690485 0.1241048
  5 -1.3362632 0.7708520
  6 -0.8240144 0.7094745
  7 -1.5623544 0.8291659

   Al
Y          [,1]      [,2]
  1 -0.58509931 0.5785897
  2 -0.02062907 0.6423855
  3 -0.44001629 0.7242601
  5  0.94968557 1.3353495
  6 -0.13450356 0.9072915
  7  1.44588396 1.0071908
```

Figure 3.1. Applicazione del metodo naive bayes sul dataset

```
   Si
Y          [,1]      [,2]
  1 -0.05279011 0.7890319
  2 -0.08919651 0.9843733
  3 -0.40966670 0.7169516
  5 -0.06167838 1.6499902
  6  0.47788540 1.1768233
  7  0.52733278 1.1735204

   K
Y          [,1]      [,2]
  1 -0.03810045 0.6169325
  2  0.20819603 0.5798754
  3 -0.12934860 0.6619458
  5  0.58265280 1.2435560
  6 -1.30168484 0.0000000
  7 -0.27375652 2.0441209

   Ca
Y          [,1]      [,2]
  1 -0.10736172 0.4089514
  2  0.05562774 1.3410539
  3 -0.11677491 0.2835153
  5  1.24139494 1.4334223
  6  0.42772300 0.8349646
  7 -0.38387929 0.7331833

   Ba
Y          [,1]      [,2]
  1 -0.3247532 0.1800485
  2 -0.2367400 0.7786962
  3 -0.3545195 0.0000000
  5  0.1804556 1.4358987
  6 -0.3545195 0.0000000
  7  1.6721920 1.3117025
```

Figure 3.2. Applicazione del metodo naive bayes sul dataset

```
   y_pred
     1  2  3  5  6  7
  1 11  0  0  0  0  0
  2  9  2  0  1  0  0
  3  2  0  0  0  0  0
  5  0  2  0  1  0  1
  6  0  2  0  0  0  1
  7  0  1  0  0  0  4

Overall Statistics

               Accuracy : 0.4865
                 95% CI : (0.3192, 0.656)
    No Information Rate : 0.5946
    P-Value [Acc > NIR] : 0.9329

                  Kappa : 0.3005

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            0.5000  0.28571       NA  0.50000       NA   0.6667
Specificity            1.0000  0.66667  0.94595  0.91429  0.91892   0.9677
Pos Pred Value         1.0000  0.16667       NA  0.25000       NA   0.8000
Neg Pred Value         0.5769  0.80000       NA  0.96970       NA   0.9375
Precision              1.0000  0.16667  0.00000  0.25000  0.00000   0.8000
Recall                 0.5000  0.28571       NA  0.50000       NA   0.6667
F1                     0.6667  0.21053       NA  0.33333       NA   0.7273
Prevalence             0.5946  0.18919  0.00000  0.05405  0.00000   0.1622
Detection Rate         0.2973  0.05405  0.00000  0.02703  0.00000   0.1081
Detection Prevalence   0.2973  0.32432  0.05405  0.10811  0.08108   0.1351
Balanced Accuracy      0.7500  0.47619       NA  0.70714       NA   0.8172
```

Figure 3.3. Confusion matrix relativa al metodo

Applichiamo K-fold cross validation con K=20.

```
> fit
Naive Bayes

177 samples
  7 predictor
  6 classes: '1', '2', '3', '5', '6', '7'

No pre-processing
Resampling: Cross-Validated (20 fold)
Summary of sample sizes: 166, 169, 167, 167, 167, 169, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE            NaN        NaN
   TRUE      0.6556385  0.5215237
```

Figure 3.4. Applicazione di k-fold cv sul dataset

3.3 Linear Discriminant Analysis(LDA)

Riprendendo parte del discorso terminato nella descrizione della classificazione bayesiana, e concentrandoci al caso di classificazione binaria $y \in 0{,}1$ si può introdurre la discriminant analysis.

Dopo aver ipotizzato che le variabili aleatorie $X_i|y$ sono distribuite come delle normali di media μ_y e varianza σ_y, quindi non assumendo l'omoschedasticitÃ , e non ipotizzando niente sulla prior density delle classi, che verrà indicata α_i per la classe i-esima, è possibile scrivere, attravero la formula di Bayes:

$$g(y|\theta, x) \propto \alpha_i g(x|\theta y)$$

dove θ indica solamente il vettore dei parametri di definizione della distribuzione normale.

Attraverso la Bayes decision rule classifichiamo x nella classe 0 quando $\alpha_0 g(x|\theta, 0) > \alpha_1 g(x|\theta, 1)$, e rendendo tutto esplicito, ed usando le proprietà del logaritmo è possibile riscrivere il tutto come:

$$\log \alpha_0 - \frac{1}{2}\log|\Sigma_0| - \frac{1}{2}(x-\mu_0)\Sigma_0^{-}1(x-\mu_0) > \log \alpha_1 - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)\Sigma_1^{-}1(x-\mu_1)$$

Definiamo quindi ora la funzione discriminante:

$$\delta_y = \log \alpha_y - \frac{1}{2}\log|\Sigma_y| - \frac{1}{2}(x-\mu_1)\Sigma_y^{-}1(x-\mu_1)$$

L'ultimo passo per giungere alla linear discriminant analysis sta in un ulteriore assunzione, infatti, da come si può notare, la funzione discriminante è una funzione quadratica (che definisce la quadratic discriminant analysis) per rimuovere i termini quadratici è sufficiente ipotizzare l'omoschedasticità, ovvero $\Sigma_0 = \Sigma_1 = \Sigma$.

Effettuando questa ulteriore ipotesi si definiscono dei decision boundaries di tipo lineare, e da questo quindi la Linear Discriminant Analysis (LDA).

```
> #Linear discriminant analysis
> lda_model <- lda(train_s_noid$type~., data = train_s_noid)
> lda_model
Call:
lda(train_s_noid$type ~ ., data = train_s_noid)

Prior probabilities of groups:
         1          2          3          5          6          7
0.33333333 0.36158192 0.08474576 0.05084746 0.03389831 0.13559322

Group means:
          Na         Mg          Al          Si          K          Ca         Ba
1 -0.1726391  0.5703226 -0.58509931 -0.05279011 -0.03810045 -0.10736172 -0.3247532
2 -0.3725816  0.1919093 -0.02062907 -0.08919651  0.20819603  0.05562774 -0.2367400
3  0.1523060  0.5690485 -0.44001629 -0.40966670 -0.12934860 -0.11677491 -0.3545195
5 -0.7963286 -1.3362632  0.94968557 -0.06167838  0.58265280  1.24139494  0.1804556
6  1.2727338 -0.8240144 -0.13450356  0.47788540 -1.30168484  0.42772300 -0.3545195
7  1.3032038 -1.5623544  1.44588396  0.52733278 -0.27375652 -0.38387929  1.6721920

Coefficients of linear discriminants:
         LD1       LD2         LD3        LD4        LD5
Na -2.718926 -3.728909 -0.79824552  -6.600846 -1.6664926
Mg -2.496941 -6.710983  0.08186118 -12.905262 -2.3078029
Al -1.660876 -1.451733 -0.07008537  -4.315553 -1.1458291
Si -2.370753 -3.576659 -0.21154829  -6.648600 -0.3074044
K  -1.088776 -1.120773 -0.08004381  -3.646548 -0.8808350
Ca -3.528562 -5.877100 -0.76092576 -13.365537 -2.2415352
Ba -1.945836 -2.719682  0.78420242  -4.733320 -0.4350769

Proportion of trace:
   LD1    LD2    LD3    LD4    LD5
0.8469 0.1106 0.0231 0.0160 0.0033
```

Figure 3.5. Applicazione del metodo LDA al dataset

```
> # Model Evaluation
> confusionMatrix(cm, mode = "everything")
Confusion Matrix and Statistics


    1 2 3 5 6 7
  1 9 2 0 0 0 0
  2 6 5 0 0 1 0
  3 1 1 0 0 0 0
  5 1 2 0 1 0 0
  6 1 0 0 0 1 1
  7 0 1 0 0 0 4

Overall Statistics

               Accuracy : 0.5405
                 95% CI : (0.3692, 0.7051)
    No Information Rate : 0.4865
    P-Value [Acc > NIR] : 0.3108

                  Kappa : 0.3735

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            0.5000   0.4545       NA  1.00000  0.50000   0.8000
Specificity            0.8947   0.7308  0.94595  0.91667  0.94286   0.9688
Pos Pred Value         0.8182   0.4167       NA  0.25000  0.33333   0.8000
Neg Pred Value         0.6538   0.7600       NA  1.00000  0.97059   0.9688
Precision              0.8182   0.4167  0.00000  0.25000  0.33333   0.8000
Recall                 0.5000   0.4545       NA  1.00000  0.50000   0.8000
F1                     0.6207   0.4348       NA  0.40000  0.40000   0.8000
Prevalence             0.4865   0.2973  0.00000  0.02703  0.05405   0.1351
Detection Rate         0.2432   0.1351  0.00000  0.02703  0.02703   0.1081
Detection Prevalence   0.2973   0.3243  0.05405  0.10811  0.08108   0.1351
Balanced Accuracy      0.6974   0.5927       NA  0.95833  0.72143   0.8844
```

Figure 3.6. Confusion matrix relativa al metodo

Applichiamo K-fold cross validation, con K=20.

```
> fit
Linear Discriminant Analysis

177 samples
  7 predictor
  6 classes: '1', '2', '3', '5', '6', '7'

No pre-processing
Resampling: Cross-Validated (20 fold)
Summary of sample sizes: 167, 170, 169, 168, 168, 170, ...
Resampling results:

  Accuracy   Kappa
  0.6675577  0.5265622
```

Figure 3.7. Applicazione del K-fold cv

3.4 Logistic regression

Introduciamo la logistic (logit) regression come un modello lineare generalizzato, dove condizionatamente ad un vettore di input p-dimensionale x, la variabile di risposta Y è distribuita secondo una Bernoulli di parametro $h(x'\beta)$ dove h è la funzione sigmoidale così definita:

$$h(u) = \frac{1}{1 + \exp -u}$$

mentre, per quanto riguarda il parametro β è possibile stimarlo attraverso la minimizzazione della cross-entropy loss.

Attraverso l'indagine del decision boundary, ristretto alla classificazione binaria, $\{x : g(y = 0|\beta, x)\} = \{x : g(y = 1|\beta, x)\}$ otteniamo che il decision boundary è definito da un iperpiano $x'\beta$.

Senza approfondire ulteriormente la trattazione nel caso di classificazioni con numeri di classi maggiore di 2, si accenna che è possibile estendere quanto detto precedentemente attraverso la definizione della funzione softmax per la trattazione del esponenziale.

```
> # Summarize the model
> summary(logit_model)
Call:
nnet::multinom(formula = train_s_noid$type ~ ., data = train_s_noid)

Coefficients:
  (Intercept)         Na         Mg         Al         Si          K         Ca         Ba
2    1.216519 -6.1165370 -13.658689 -1.8880251 -6.3969108  -2.999602 -11.8222767  -4.907865
3  -11.555765  0.5204964  -1.419874  0.8638708 -0.9400636  -0.312891  -0.8547797 -31.625262
5   -2.996941 -2.0760118  -7.574900  3.6226558 -1.4076633  -0.965555  -3.5084965  -2.546079
6  -51.411198 20.5449584  20.291671 18.3466683 14.2790032 -24.135270  26.0480108 -30.297033
7   -3.163714 19.5585674  27.670858 13.7528866 18.3331157   7.679316  30.6484236  12.151117

Std. Errors:
  (Intercept)         Na         Mg         Al         Si          K         Ca         Ba
2   0.5936374   2.057949   3.822302  1.355934   1.979497   1.222309   3.790300   1.783742
3  37.6226371   2.829258   5.280404  1.946298   2.679872   1.620561   5.193136 106.124659
5   1.3707755   4.346328   8.080825  3.289057   4.270473   2.328319   8.142014   3.160246
6 125.5928631 158.277298 275.565517 91.815854 162.578583 187.010551 269.288375  78.111617
7   1.4923447  11.163135  19.187884  6.754901  10.442337   4.844796  19.603369   7.473138

Residual Deviance: 237.5416
AIC: 317.5416
```

Figure 3.8. Applicazione del metodo Logistic Regression al dataset

```
> # Model Evaluation
> confusionMatrix(cm, mode = "everything")
Confusion Matrix and Statistics

   y_pred_logit
    1 2 3 5 6 7
  1 7 2 0 0 0 2
  2 5 4 0 0 0 3
  3 1 1 0 0 0 0
  5 0 2 0 0 0 2
  6 0 1 0 0 0 2
  7 0 1 0 0 0 4

Overall Statistics

               Accuracy : 0.4054
                 95% CI : (0.2475, 0.579)
    No Information Rate : 0.3514
    P-Value [Acc > NIR] : 0.2986

                  Kappa : 0.2089

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            0.5385   0.3636       NA       NA       NA   0.3077
Specificity            0.8333   0.6923  0.94595   0.8919  0.91892   0.9583
Pos Pred Value         0.6364   0.3333       NA       NA       NA   0.8000
Neg Pred Value         0.7692   0.7200       NA       NA       NA   0.7188
Precision              0.6364   0.3333  0.00000   0.0000  0.00000   0.8000
Recall                 0.5385   0.3636       NA       NA       NA   0.3077
F1                     0.5833   0.3478       NA       NA       NA   0.4444
Prevalence             0.3514   0.2973  0.00000   0.0000  0.00000   0.3514
Detection Rate         0.1892   0.1081  0.00000   0.0000  0.00000   0.1081
Detection Prevalence   0.2973   0.3243  0.05405   0.1081  0.08108   0.1351
Balanced Accuracy      0.6859   0.5280       NA       NA       NA   0.6330
```

Figure 3.9. Confusion matrix relativa al metodo

3.5 K-Nearest Neighbors

La classificazione K-Nearest Neighbors (K-NN) è caratterizzata da una procedura molto intuitiva, che sfrutta il concetto di distanza.
Entrando più nel vivo del metodo, sia τ il training set a disposizione, con $y_i = \{0, ..., c-1\}$ classi possibili, e sia x un nuovo vettore di features in ingresso.

Definiamo ora $x_1, x_2, ..., x_n$ i vettori di features ordinati secondo il concetto di 'vicinanza '(relativa alla metrica scelta).

Sia ora τ_{x_k} il sottoinsieme di τ che contiene i primi 'k 'vettori di features, che sono più 'vicini 'a x.

Allora la K-nearest neighbors classification rule, classifica il vettore in ingresso x nella classe che compare più frequentemente nell'insieme τ_{x_k}.

```
> # Model Evaluation
> confusionMatrix(cm, mode = "everything")
Confusion Matrix and Statistics


y_pred_knn  1  2  3  5  6  7
         1 11  1  0  0  0  0
         2  0 11  0  0  0  0
         3  0  0  2  2  0  0
         5  0  0  0  2  1  0
         6  0  0  0  0  2  0
         7  0  0  0  0  0  5

Overall Statistics

               Accuracy : 0.8919
                 95% CI : (0.7458, 0.9697)
    No Information Rate : 0.3243
    P-Value [Acc > NIR] : 1.062e-12

                  Kappa : 0.8596

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            1.0000   0.9167  1.00000  0.50000  0.66667   1.0000
Specificity            0.9615   1.0000  0.94286  0.96970  1.00000   1.0000
Pos Pred Value         0.9167   1.0000  0.50000  0.66667  1.00000   1.0000
Neg Pred Value         1.0000   0.9615  1.00000  0.94118  0.97143   1.0000
Precision              0.9167   1.0000  0.50000  0.66667  1.00000   1.0000
Recall                 1.0000   0.9167  1.00000  0.50000  0.66667   1.0000
F1                     0.9565   0.9565  0.66667  0.57143  0.80000   1.0000
Prevalence             0.2973   0.3243  0.05405  0.10811  0.08108   0.1351
Detection Rate         0.2973   0.2973  0.05405  0.05405  0.05405   0.1351
Detection Prevalence   0.3243   0.2973  0.10811  0.08108  0.05405   0.1351
Balanced Accuracy      0.9808   0.9583  0.97143  0.73485  0.83333   1.0000
```

Figure 3.10. Applicazione del metodo KNN al dataset

3.6 SVM

Restringendo ora la classificazione nuovamente al caso binario $\{-, +\}$, introduciamo un nuovo metodo di classificazione denominato Support Vector Machines (SVM).

Sommariamente, questo metodo si pone l'obiettivo di separare i maniera lineare i dati attraverso una retta, quest'ultima scelta in modo "ottimale", dove, in questo contesto, la parola ottimale indica che la retta scelta sarà quella che avrà direzione più conforme alla separazione dei dati, lasciando più spazio possibile tra essa e i dati (più vicini) nelle due diverse classificazione.

La regola di classificazione di questo metodo viene trovata passando attraverso un probelma di massimizzazione vincolata che dopo alcuni conti ci porterà al seguente risultato

$$\Sigma_i \alpha_i x_i \cdot u + b \geq 0$$

dove x_i sono i vari dati presenti nel training set mentre u, che sta per unknown, indica appunto il dato in input che deve essere classifcato. Nello specifico, un dato che rispetta la condizione sopra viene categorizzato come $+$.

L'ultima questione da trattare per questo metodo è la casistica in cui i dati non possono essere separati da una linea (come magari può capitare con dati a spirale), in questo caso si applica una mappa ai dati per studiarli in dimensione maggiore, dove magari possono essere divisi da un iperpiano. Questo metodo è possibile nella pratica grazie al cosiddetto kernel trick, ovvero, un'idea che evita il calcolo esplicito della mappa. Questo trick è possibile dal fatto che la decision rule sia solamente in funzione del prodotto scalare tra vettori, quindi basta mappare i dati in spazi in cui il prodotto scalare è ben definito, come gli spazi di Hilbert, ed è possibile riproporre il metodo.

Per estendere il metodo al caso di classificazioni non binarie si adottano tipicamente strategie one-vs-one o one-vs-all. Nell'applicazione del metodo abbiamo utilizzato il kernel=radial.

```
> # Model Evaluation
> confusionMatrix(cm, mode = "everything")
Confusion Matrix and Statistics


y_pred_svm 1 2 3 5 6 7
         1 9 4 2 0 0 0
         2 2 8 0 3 2 1
         3 0 0 0 0 0 0
         5 0 0 0 1 1 0
         6 0 0 0 0 0 0
         7 0 0 0 0 0 4

Overall Statistics

               Accuracy : 0.5946
                 95% CI : (0.421, 0.7525)
    No Information Rate : 0.3243
    P-Value [Acc > NIR] : 0.0006516

                  Kappa : 0.436

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            0.8182   0.6667  0.00000  0.25000  0.00000   0.8000
Specificity            0.7692   0.6800  1.00000  0.96970  1.00000   1.0000
Pos Pred Value         0.6000   0.5000      NaN  0.50000      NaN   1.0000
Neg Pred Value         0.9091   0.8095  0.94595  0.91429  0.91892   0.9697
Precision              0.6000   0.5000       NA  0.50000       NA   1.0000
Recall                 0.8182   0.6667  0.00000  0.25000  0.00000   0.8000
F1                     0.6923   0.5714       NA  0.33333       NA   0.8889
Prevalence             0.2973   0.3243  0.05405  0.10811  0.08108   0.1351
Detection Rate         0.2432   0.2162  0.00000  0.02703  0.00000   0.1081
Detection Prevalence   0.4054   0.4324  0.00000  0.05405  0.00000   0.1081
Balanced Accuracy      0.7937   0.6733  0.50000  0.60985  0.50000   0.9000
```

Figure 3.11. Applicazione di SVM al dataset

Applichiamo K-fold cross validation, con K=20.

```
> fit
Support Vector Machines with Radial Basis Function Kernel

177 samples
  7 predictor
  6 classes: '1', '2', '3', '5', '6', '7'

No pre-processing
Resampling: Cross-Validated (20 fold)
Summary of sample sizes: 170, 168, 169, 169, 168, 169, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.6581602  0.4817704
  0.50  0.6628229  0.4897921
  1.00  0.6947294  0.5417307

Tuning parameter 'sigma' was held constant at a value of 0.6643791
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.6643791 and C = 1.
```

Figure 3.12. Applicazione di k fold cv al dataset

3.7 Decision tree

Il decision tree è un metodo di classificazione semplice da visualizzare e semplice da interpretare.
Il meccanismo di classficazione consiste nella suddivisione dei dati di input attraverso degli step (che saranno i nodi dell'albero) ciascuno caratterizzato da un criterio di suddivisione. Per ogni nodo dell'albero verrà selezionata un'entrata del vettore di input, una variabile, su cui applicare un criterio di suddivsione/scelta in modo tale da minimizzare, in modo miope, il generalization risk(tipicamente questo metod viene definito greedy - top - down).
Un fattore che sarà importante e che quindi è opportuno specificare, è che ad ogni step sono disponibili tutte le entarte del vettore, e quindi tutte le variabili, questo potrebbe comportare un utilizzo molto frequente di una variabile per la classificazione accurata dei dati.
I lati negativi di questo metodo sono dovuti alla poca adattabilità verso i dati di input a causa della tendenza intrinseca del metodo al fenomeno di overfitting. Per arginare questo lato negativo è possibile ridurre la profondità dell'albero a discapito della minimizzazione della training loss.

```
> # Model Evaluation
> confusionMatrix(cm, mode = "everything")
Confusion Matrix and Statistics


y_pred_tree  1  2  3  5  6  7
          1  5  1  1  0  0  0
          2  6 11  1  1  2  1
          3  0  0  0  0  0  0
          5  0  0  0  3  1  0
          6  0  0  0  0  0  0
          7  0  0  0  0  0  4

Overall Statistics

               Accuracy : 0.6216
                 95% CI : (0.4476, 0.7754)
    No Information Rate : 0.3243
    P-Value [Acc > NIR] : 0.0001955

                  Kappa : 0.4778

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            0.4545   0.9167  0.00000  0.75000  0.00000   0.8000
Specificity            0.9231   0.5600  1.00000  0.96970  1.00000   1.0000
Pos Pred Value         0.7143   0.5000      NaN  0.75000      NaN   1.0000
Neg Pred Value         0.8000   0.9333  0.94595  0.96970  0.91892   0.9697
Precision              0.7143   0.5000       NA  0.75000       NA   1.0000
Recall                 0.4545   0.9167  0.00000  0.75000  0.00000   0.8000
F1                     0.5556   0.6471       NA  0.75000       NA   0.8889
Prevalence             0.2973   0.3243  0.05405  0.10811  0.08108   0.1351
Detection Rate         0.1351   0.2973  0.00000  0.08108  0.00000   0.1081
Detection Prevalence   0.1892   0.5946  0.00000  0.10811  0.00000   0.1081
Balanced Accuracy      0.6888   0.7383  0.50000  0.85985  0.50000   0.9000
```

Figure 3.13. Applicazione del metodo Decision Tree al dataset

3.8 Random forest

In questa sezione verrà trattato il metodo di classificazione random forest. Questo metodo viene costruito partendo dall'idea di aggregazione di prediction functions allenati da diversi dataset, con l'obiettivo di migliorare l'accuracy. Denominiamo questa classe di metodi Bagging methods. Questa metodologia è particolarmente efficace per i metodi che soffrono del fenomeno di overfitting.

I risultati teorici danno importanti miglioramenti quando si hanno a disposizione copie indipendenti ed identicamente distribuite di training sets, ottenendo risultati importanti sulla riduzione dell'expected generalization risk quando si utilizza la squared-error loss.

Sfortunatamente non si è quasi mai in possesso di datasets iid, ma essi possono venir sostituiti dalle loro controparti ottenute col boostrap, da cui otteniamo il bagged estimator.

L'impostazione del problema e dei risultai appena seguita è adeguata ad un problema di tipo regressivo ma può essere anche applicata alla classificazione.

Malgrado i risultati ottenuti siano molto importanti essi sono validi a pieno in un contesto quasi ideale come quello iid descritto, e quando si utiilizzano campionamenti boostrap si perde parte di questi risultati, e la causa principale risiede nel fatto che, nonostante i dataset continuino ad essere identicamente distribuiti, non sono più indipendenti e questo comporta una correlazione tra i learners generati, che comporta un aumento della varianza dovuta ad un fattore di correlazione.

Per evitare questo fenomeno è possibile applicare delle tecniche per rendere meno correlati i learners e qui entra in gioco il random forest.

Come già anticipato questo metodo favorisce le classificazioni che tendono a soffrire di overfitting, proprio come il decision tree.

Per cercare di applicare il metodo in maniera efficiente si effettua la seguente procedura:

1. Si generano copie del dataset attraverso il metodo boostrap. 2. Per ciascun dataset creato si applica un decision tree, con la variazione che prima di applicarlo, vengano "nascoste" casualmente delle entrete del vettore di input, cosicchè, come già anticipato nel decision tree, si evtino ripetizioni frequenti di una variabile sopra le altre.

Il metodo appena descritto viene chiamato Random Forest.

```
> # Model Evaluation
> confusionMatrix(cm, mode = "everything")
Confusion Matrix and Statistics


y_pred_rf  1  2  3  5  6  7
        1  8  1  2  0  0  0
        2  3 11  0  3  2  1
        3  0  0  0  0  0  0
        5  0  0  0  1  0  0
        6  0  0  0  0  1  0
        7  0  0  0  0  0  4

Overall Statistics

               Accuracy : 0.6757
                 95% CI : (0.5021, 0.8199)
    No Information Rate : 0.3243
    P-Value [Acc > NIR] : 1.271e-05

                  Kappa : 0.5474

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            0.7273   0.9167  0.00000  0.25000  0.33333   0.8000
Specificity            0.8846   0.6400  1.00000  1.00000  1.00000   1.0000
Pos Pred Value         0.7273   0.5500      NaN  1.00000  1.00000   1.0000
Neg Pred Value         0.8846   0.9412  0.94595  0.91667  0.94444   0.9697
Precision              0.7273   0.5500       NA  1.00000  1.00000   1.0000
Recall                 0.7273   0.9167  0.00000  0.25000  0.33333   0.8000
F1                     0.7273   0.6875       NA  0.40000  0.50000   0.8889
Prevalence             0.2973   0.3243  0.05405  0.10811  0.08108   0.1351
Detection Rate         0.2162   0.2973  0.00000  0.02703  0.02703   0.1081
Detection Prevalence   0.2973   0.5405  0.00000  0.02703  0.02703   0.1081
Balanced Accuracy      0.8059   0.7783  0.50000  0.62500  0.66667   0.9000
```

Figure 3.14. Applicazione del metodo random forest sul dataset

Applichiamo K-fold cross validation con K=20.

```
> fit
Random Forest

177 samples
  7 predictor
  6 classes: '1', '2', '3', '5', '6', '7'

No pre-processing
Resampling: Cross-Validated (20 fold)
Summary of sample sizes: 169, 168, 168, 168, 168, 167, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.7394192  0.6272248
  4     0.7101136  0.5877759
  7     0.7020581  0.5753600

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Figure 3.15. Applicazione di k-fold cv

Chapter 4

Conclusioni

In questo capitolo verranno tratte alcune conclusioni sui risultati ottenuti dall'applicazione dei metodi.

Preliminarmente notiamo come la ridotta disponibilità di dati delle classi 3 e 6 abbia portato ad avere squilibri nei vari metodi, infatti la precision per queste classi assume raramente valori alti.
Notiamo inoltre come i metodi meno accurati sono stati la classificazione bayesiana e la logistic regression.
Mentre, invece, i risultati migliori sono stati ottenuti con la classificazione KNN e con il random forest.

Come si può constatare dai risultati notiamo come in generale i metodi oscillino tra valori di 0.48 e 0.67 dell'accuracy, con un picco superiore dato proprio dal metodo KNN, con accuracy pari a 0.89 ed un picco inferiore dato dalla Logistic Regression con valore pari a 0.40. Ipotizziamo che i dati trattati siano particolarmente adeguati per un'analisi metrica, come quella offerta da KNN.

Si possono giustificare i risultati dicendo che le ipotesi dei vari metodi potrebbero non essere adeguate per il data set in esame, ricordando anche che una causa della bassa efficienza dei metodi possa essere dovuta alla piuttosto ridotta disponibilità dei dati.

Non escludiamo la possibilità che la scelta di escludere delle features possa essere stata un fattore nel compromettere l'efficienza dei metodi, ricordando comunque che le evidenze delle analisi di unsupervised learning sembrano confermare la poca rilevanza delle variabili escluse.

Lo scaling effettuato in via preliminare immaginiamo abbia aiutato l'applicazione dei metodi, è noto infatti che per tutti i metodi che utilizzano i concetti di metrica è necessario uno scaling preliminare, nel nostro caso KNN e SVM sono i metodi che ne giovano maggiormente, e questo viene rispecchiato anche dai risultati ottenuti, in un caso eccellenti e nell'altro accettabili. Per gli altri metodi non è strettamente necessario lo scaling

ma è comunque consigliato.

Per alcuni metodi è stata riportata anche la valutazione effettuata dall'applicazione del K-folds cross validation, che restituisce una valutazione dell'accuracy per i vari metodi.

Concludiamo dicendo che il metodo più adeguato per il tipo di dataset analizzato sembra essere, per distacco, il KNN, e questo termina l'analisi.

Bibliography

G. James, D. Witten, T. Hastie, R.Tibshirani âAn Introduction To Statistical Learningâ edited by Springer Science+ Business Media New York 2013,