

UNIVERSITÀ DEGLI STUDI MILANO-BICOCCA

Corso di Laurea Magistrale in Informatica



Progetto Machine Learning

A.A. 2022/2023

Report di:

[Davide Grandesso 852078](#)

[Fabio Marini 851977](#)

Sommario

1	Introduzione	1
1.1	Dominio di riferimento	1
1.2	Design del dataset	1
1.2.1	Assunzioni e modifiche	2
1.3	Obiettivi elaborato	2
2	Exploratory Data Analysis (EDA)	3
2.1	Analisi del dataset	3
2.1.1	Distribuzioni e densità	4
2.1.2	Matrice delle correlazioni	5
2.1.3	Grafico a torta sulla sicurezza dell'acqua	5
2.1.4	Confronto degli attributi sulla variabile target	6
2.2	Analisi dei Training Set	7
2.2.1	Distribuzioni e densità	8
2.2.2	Matrice delle correlazioni	9
2.2.3	Grafico a torta sulla sicurezza dell'acqua	9
2.2.4	Confronto degli attributi sulla variabile target	10
2.3	PCA	11
2.3.1	Variabili	11
2.3.2	Autovalori	11
2.3.3	Individui	13
3	Alberi di decisione (DT)	14
3.1	Motivazioni della scelta del modello	14
3.2	Allenamento del modello	14
3.3	Misure di performance	15
3.3.1	Matrice di confusione e relative metriche di performance	15
3.3.2	Curva ROC & AUC	16
3.3.3	Cut-off	17
4	Support Vector Machine (SVM)	18
4.1	Motivazioni della scelta del modello	18
4.2	Allenamento del modello	18
4.3	Misure di performance	19
4.3.1	Matrice di confusione e relative misure di performance	19
4.3.2	Curva ROC & AUC	20

4.3.3	Cut-off	21
5	Naive Bayes	22
5.1	Motivazioni della scelta del modello	22
5.2	Allenamento del modello	22
5.3	Misure di performance	23
5.3.1	Matrice di confusione e relative misure di performance	23
5.3.2	Curva ROC & AUC	24
5.3.3	Cut-off	25
6	Comparazione dei modelli	26
7	10-Fold Cross-Validation	27
7.1	Dotplot	28
7.2	Bwplot	29
7.3	Tempi di esecuzione	29
8	Conclusioni	30

1 Introduzione

1.1 Dominio di riferimento

Il dataset scelto ([Water quality](#)) è stato creato con dati d'esempio sulla qualità dell'acqua in un ambiente urbano.



1.2 Design del dataset

Il nostro dataset è composto da 7999 osservazioni su 21 attributi dove i primi 20 attributi descrivono quanto è presente un elemento all'interno dell'acqua, mentre l'ultimo campo esprime se l'acqua presa in esame è sicura o meno. Tutti gli attributi sono variabili numeriche e sono elencati di seguito:

- **Alluminio** [aluminium] → *pericoloso se superiore a 2,8*
- **Ammoniaca** [ammonia] → *pericolosa se superiore a 32,5*
- **Arsenico** [arsenic] → *pericoloso se superiore allo 0,01*
- **Bario** [barium] → *pericoloso se superiore a 2*
- **Cadmio** [cadmium] → *pericoloso se superiore a 0,005*
- **Cloramina** [chloramine] → *pericolosa se superiore a 4*
- **Cromo** [chromium] → *pericoloso se superiore allo 0,1*
- **Rame** [copper] → *pericoloso se superiore a 1,3*
- **Flouride** [flouride] → *pericoloso se superiore a 1,5*
- **Batteri** [bacteria] → *pericolosi se superiori a 0*
- **Virus** [viruses] → *pericoloso se maggiore di 0*
- **Piombo** [lead] → *pericoloso se superiore a 0,015*
- **Nitrati** [nitrates] → *pericolosi se superiori a 10*
- **Nitriti** [nitrites] → *pericolosi se superiori a 1*
- **Mercurio** [mercury] → *pericoloso se superiore a 0,002*
- **Perclorato** [perchlorate] → *pericoloso se superiore a 56*
- **Radio** [radium] → *pericoloso se superiore al 5*
- **Selenio** [selenium] → *pericoloso se superiore a 0,5*
- **Argento** [silver] → *pericoloso se superiore allo 0,1*
- **Uranio** [uranium] → *pericoloso se maggiore di 0,3*
- **È sicura** [is_safe] → attributo di classe {0 → non sicuro - 1 → sicuro}

Le variabili che rappresentano gli elementi hanno valore continuo, mentre la variabile target (is_safe) ha valore binario [0,1].

Se un'osservazione di un elemento supera il valore di pericolosità, non implica che l'acqua sia non sicura. Se ad esempio dovessimo rilevare un valore di alluminio maggiore di 2,8, questo non ci porterebbe a dire con certezza che l'acqua osservata non sia sicura. Solamente analizzando l'intera riga di osservazioni siamo in grado di capire la qualità dell'acqua.

1.2.1 Assunzioni e modifiche

All'interno del dataset sono presenti 3 righe con valore nullo rappresentato da #NUM! all'interno della colonna `ammonia`, queste osservazioni portano ad avere valore nullo anche per quanto riguarda il valore della variabile target `is_safe`. Abbiamo quindi eliminato le suddette righe.

Abbiamo notato che la colonna `ammonia` conteneva valori negativi. Dato che queste osservazioni misurano delle quantità non le abbiamo ritenute valide; perciò, abbiamo eliminato le 10 righe che contenevano questi valori errati.

È stato effettuato un controllo di eventuali righe duplicate senza però riscontrarne alcune, di conseguenza non sono state apportate altre modifiche.

Finite le modifiche, le righe rimanenti nel dataset sono 7986.

1.3 Obiettivi elaborato

Dopo aver effettuato le opportune correzioni sul dataset ed un'analisi esplorativa preliminare, il nostro obiettivo è quello di prevedere se il campione di acqua osservato sia sicuro o meno tramite dei modelli di machine learning.

Per eseguire questa classificazione binaria useremo i seguenti modelli: "Alberi di Decisione (DT)", "Support Vector Machine (SVM)" e "Naive Bayes (NB)" di cui verranno fatte per ognuno di essi delle valutazioni delle performance.

2 Exploratory Data Analysis (EDA)

2.1 Analisi del dataset

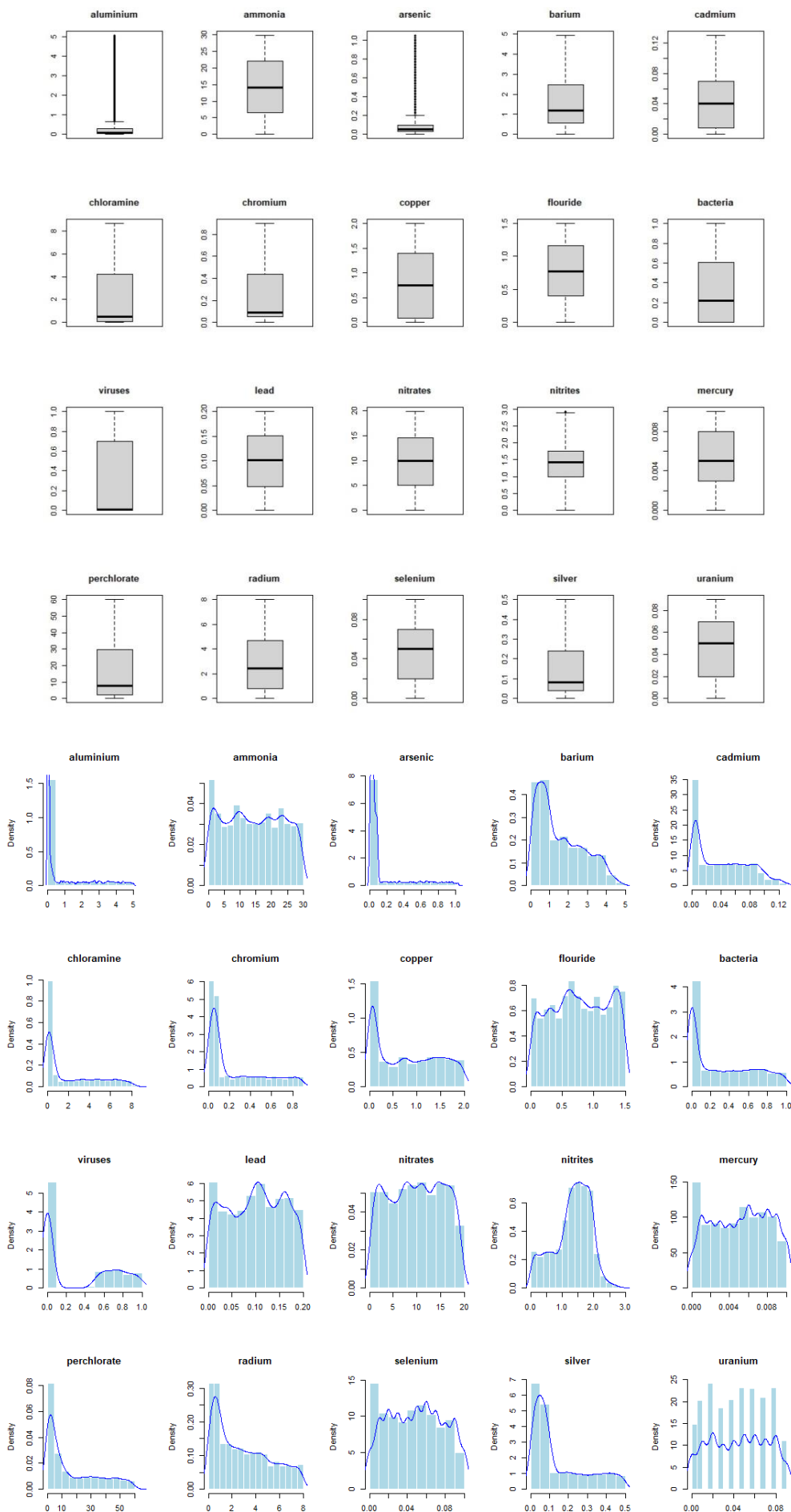
Come prima analisi riportiamo di seguito una tabella che rappresenta una breve descrizione delle variabili presenti nel dataset per quanto riguarda le principali statistiche descrittive:

	Aluminium	Ammonia	Arsenic	Barium	Cadmium	Chloramine	Chromium
Minimo	0	0	0	0	0	0	0
Media	0.6664	14.30	0.1615	1.568	0.0428	2.178	0.2473
Mediana	0.07	14.15	0.05	1.190	0.0400	0.530	0.0900
Massimo	5.05	29.84	1.05	4.940	0.1300	8.680	0.9000
Varianza	1.60081	78.67706	0.06363	1.4762	0.0012976	6.590255	0.07323

	Copper	Flouride	Bacteria	Viruses	Lead	Nitrates	Nitrites
Minimo	0	0	0	0	0	0	0
Media	0.8059	0.7716	0.3197	0.3287	0.09943	9.819	1.33
Mediana	0.7500	0.7700	0.2200	0.0080	0.10200	9.930	1.42
Massimo	2.0000	1.5000	1.0000	1.0000	0.20000	19.830	2.93
Varianza	0.4269283	0.1896933	0.108522	0.1429388	0.003380035	30.72451	0.3286881

	Mercury	Perchlorate	Radium	Selenium	Silver	Uranium
Minimo	0	0	0	0	0	0
Media	0.005193	16.465	2.92	0.04968	0.1478	0.04467
Mediana	0.005000	7.745	2.41	0.05000	0.0800	0.05000
Massimo	0.010000	60.010	7.99	0.10000	0.5000	0.09000
Varianza	8.8037e-06	313.0897	5.392496	0.0008283	0.02059859	0.0007237

2.1.1 Distribuzioni e densità



Report Machine Learning

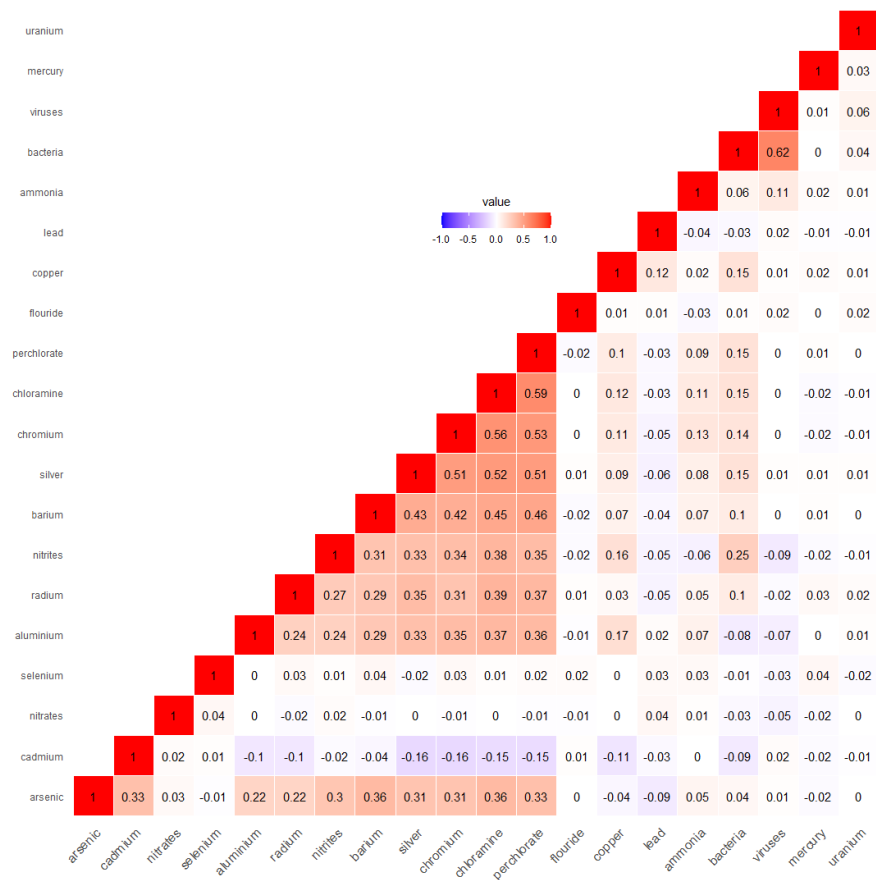
Dai grafici emerge che nessuna delle variabili segue una distribuzione normale.

Si nota che le colonne aluminium, arsenic, chloramine, chromium, bacteria, silver e perchlorate seguono distribuzioni a coda lunga verso destra. Ciò significa che ci sono pochi valori estremamente elevati rispetto ai valori più comuni.

Questa caratteristica era già possibile notarla nei rispettivi boxplot presentati precedentemente, in particolare si nota in quelli di arsenic e aluminium che presentano molti outliers sopra il 3° quartile.

2.1.2 Matrice delle correlazioni

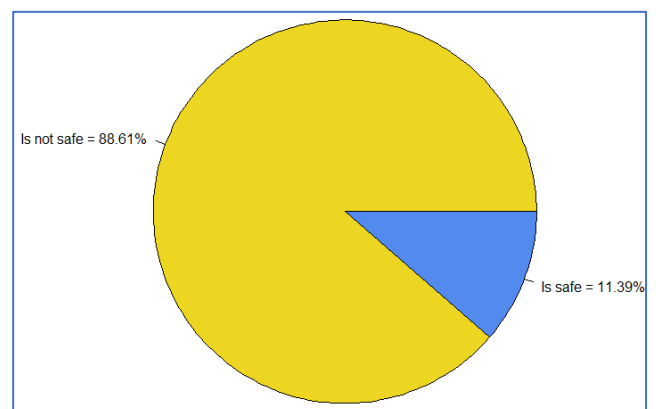
Andiamo adesso ad osservare la correlazione tra le colonne del dataset:



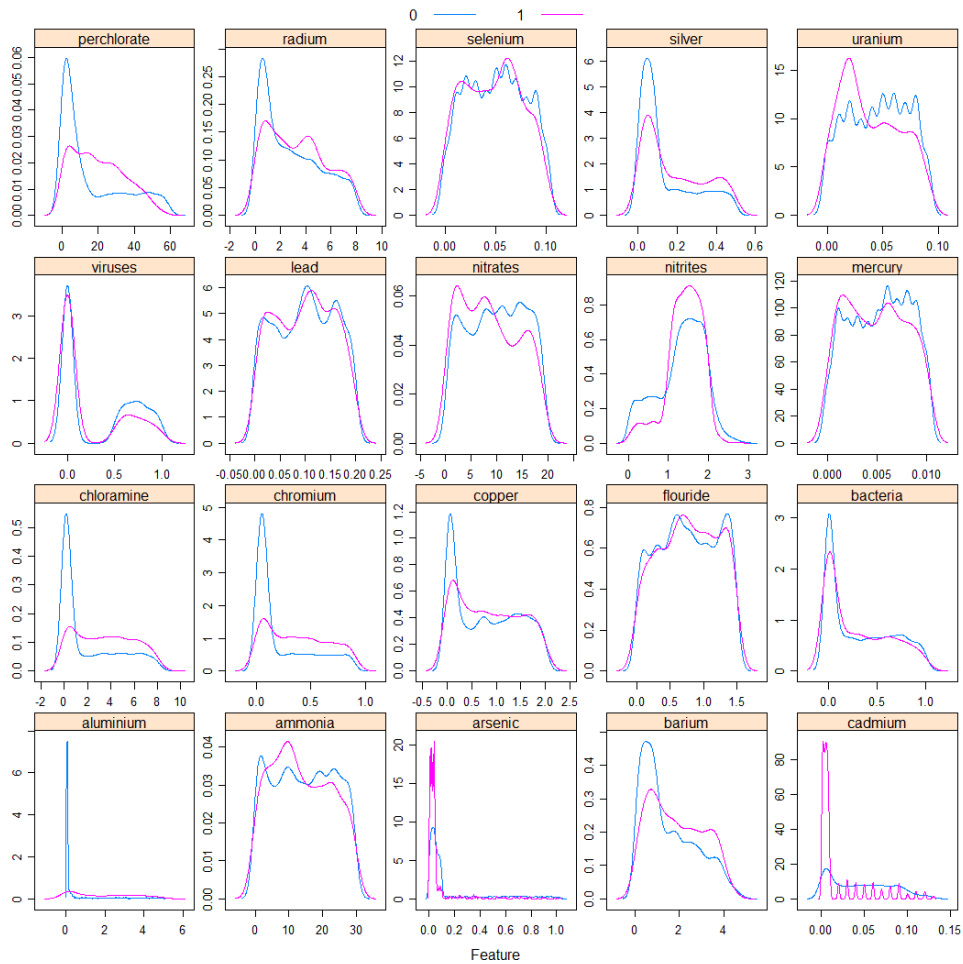
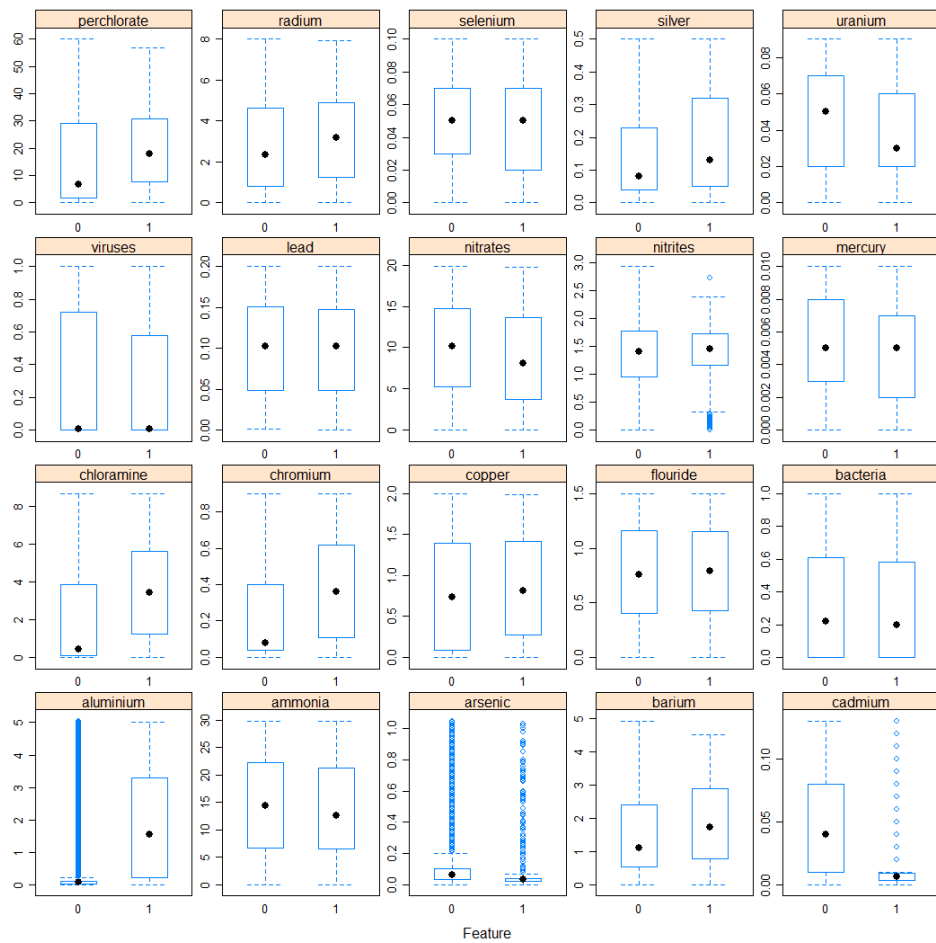
Solo pochi attributi sono altamente correlati, ma nessuno di essi è inversamente correlato ad un altro. È interessante notare che la correlazione più alta (0.62) è tra i virus ed i batteri.

2.1.3 Grafico a torta sulla sicurezza dell'acqua

Il dataset contiene 7076 osservazioni di campioni dell'acqua non sicuri e 910 sicuri, di seguito riportiamo questa distribuzione tramite un grafico a torta.



2.1.4 Confronto degli attributi sulla variabile target



Abbiamo riportato i grafici sulle distribuzioni, ma suddivise in base a se il campione osservato contiene acqua sicura o meno. Questi tipi di grafici ci permettono di capire meglio come sono fatte le distribuzioni sia nel caso in cui l'acqua è sicura sia nel caso in cui non lo fosse.

Non è possibile osservare una regola generale di distribuzione delle variabili in base al campione sicuro o meno, infatti si nota come nelle colonne *virus* e *bacteria* la distribuzione sia simile in entrambi i casi, nelle colonne *aluminium* e *cadmium* sembrano esserci valori minori quando l'acqua non è sicura e nelle colonne *uranium* e *nitrites* ci siano valori minori quando l'acqua è sicura.

2.2 Analisi dei Training Set

Una volta analizzato tutto il dataset andiamo ad effettuare un'analisi dei dati relativa al solo training set. Per dividere il dataset in training set e test set abbiamo scelto di suddividere i dati rispettivamente del 70% e del 30% avendo così nel training set un numero di righe pari a 5590.

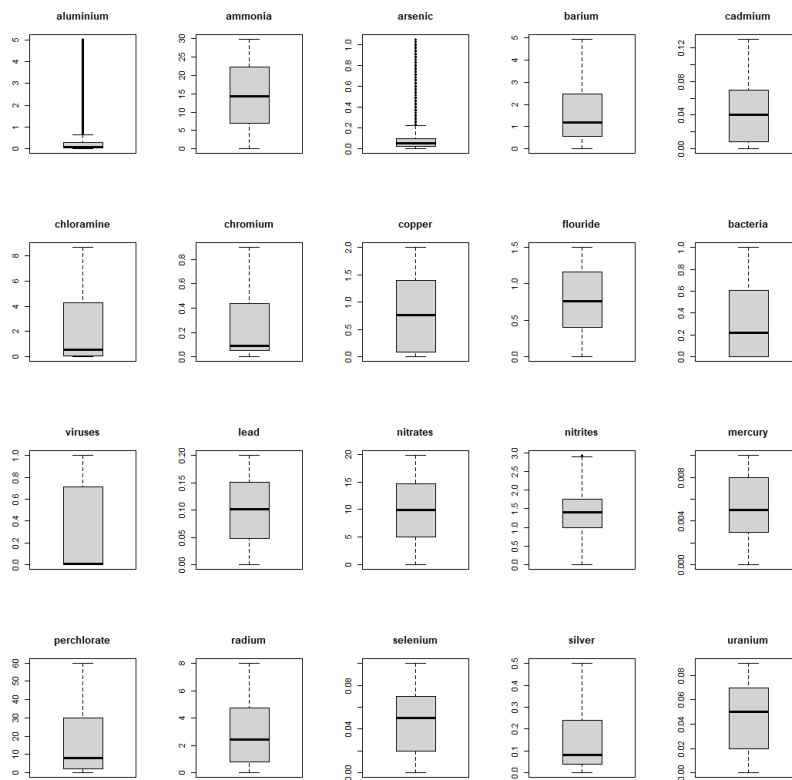
Quest'analisi ha lo scopo di capire se questo sottoinsieme di dati sia simile al dataset e quindi che possa generalizzare bene nuovi dati come quelli del test set.

	Aluminium	Ammonia	Arsenic	Barium	Cadmium	Chloramine	Chromium
Minimo	0	0	0	0	0	0	0
Media	0.6797	14.417	0.1618	1.563	0.04256	2.199	0.248
Mediana	0.0700	14.380	0.0500	1.190	0.04000	0.550	0.090
Massimo	5.0300	29.840	1.0500	4.940	0.13000	8.680	0.900
Varianza	1.640967	77.84203	0.06487521	1.46411	0.001293	6.6435	0.07348296

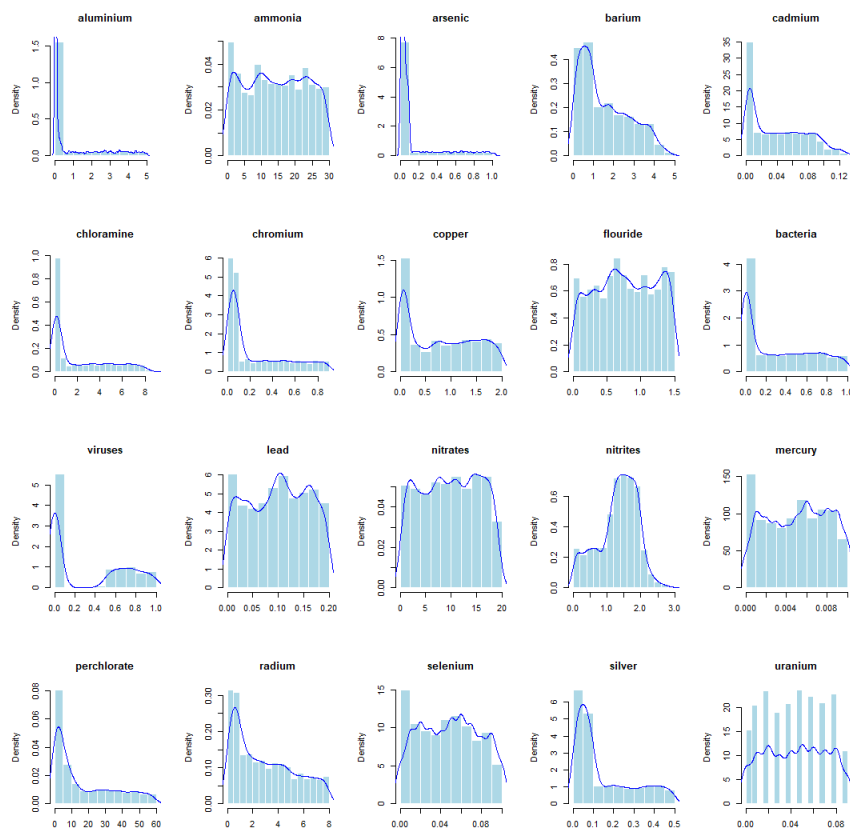
	Copper	Flouride	Bacteria	Viruses	Lead	Nitrates	Nitrites
Minimo	0	0	0	0	0	0	0
Media	0.8143	0.769	0.3229	0.3325	0.09955	9.834	1.329
Mediana	0.7700	0.760	0.2200	0.0080	0.10200	9.975	1.410
Massimo	2.0000	1.500	1.0000	1.0000	0.20000	19.800	2.930
Varianza	0.430441	0.1889227	0.1094197	0.1432171	0.003362503	30.80615	0.3299386

	Mercury	Perchlorate	Radium	Selenium	Silver	Uranium
Minimo	0	0	0	0	0	0
Media	0.005184	16.64	2.938	0.04956	0.1493	0.04448
Mediana	0.005000	8.06	2.430	0.05000	0.0800	0.05000
Massimo	0.010000	59.74	7.990	0.10000	0.5000	0.09000
Varianza	8.897647e-06	314.5753	5.445181	0.0008369833	0.02078726	0.0007227794

2.2.1 Distribuzioni e densità

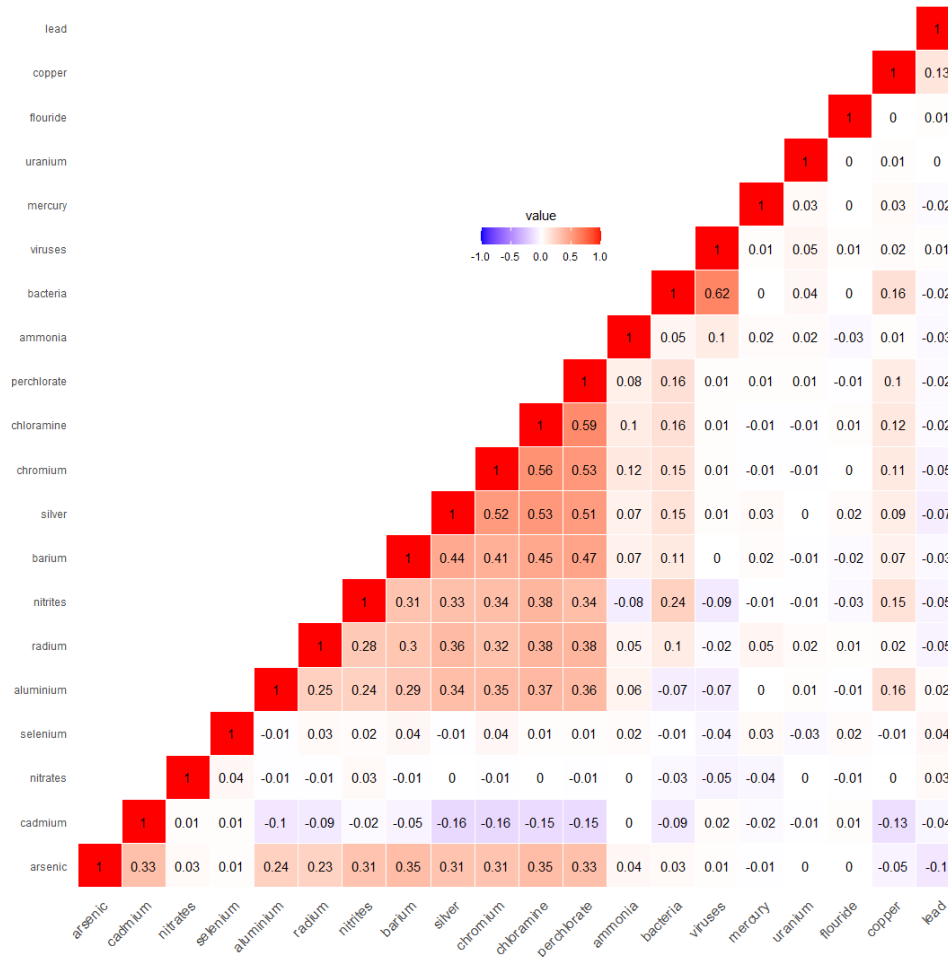


Confrontando le tabelle, e di conseguenza i boxplot, del training set e del dataset non emergono differenze significative. Perciò anche le seguenti distribuzioni risultano essere simili alle precedenti.



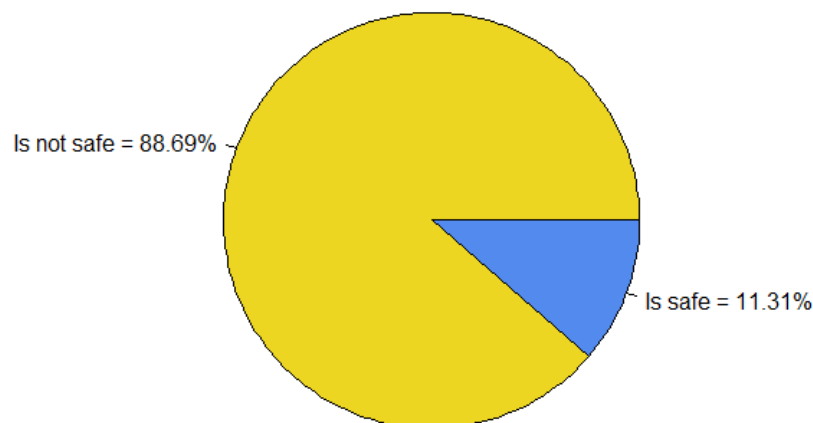
2.2.2 Matrice delle correlazioni

Anche dalla matrice delle correlazioni non emergono particolari differenze, tutte le correlazioni precedenti sono rispettate o cambiate di valori molto poco significativi.

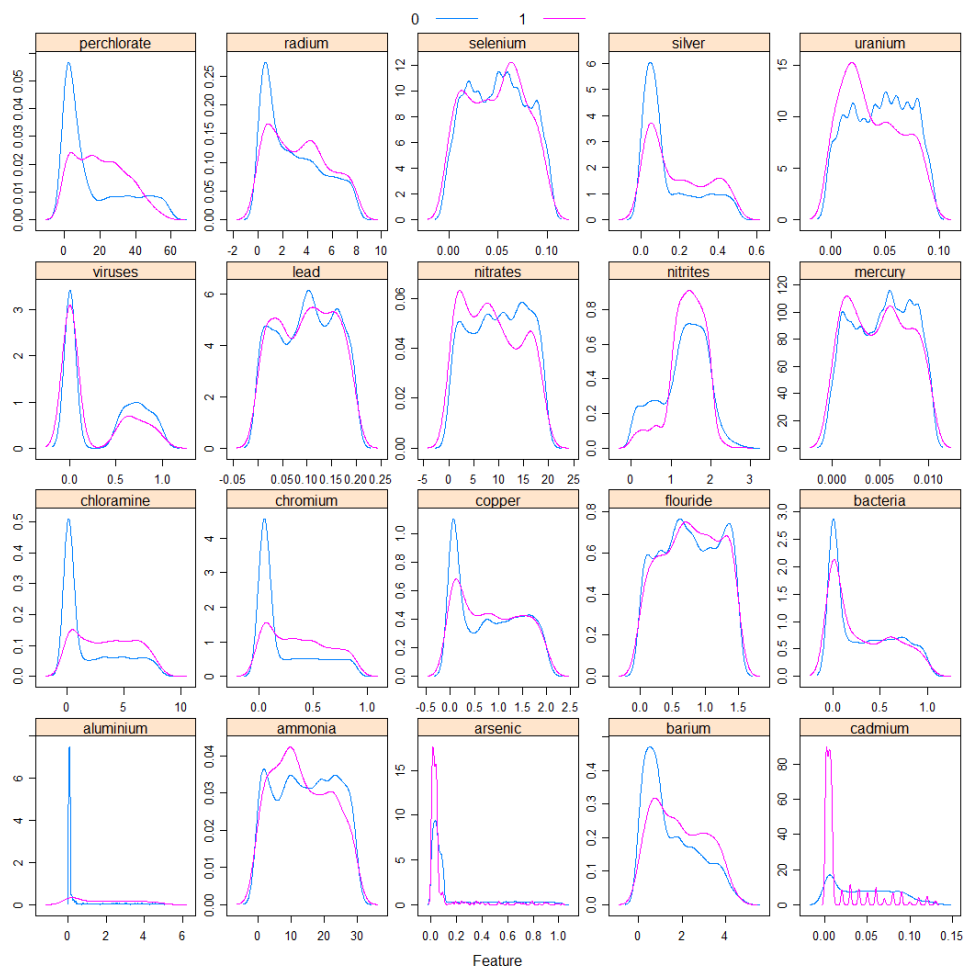
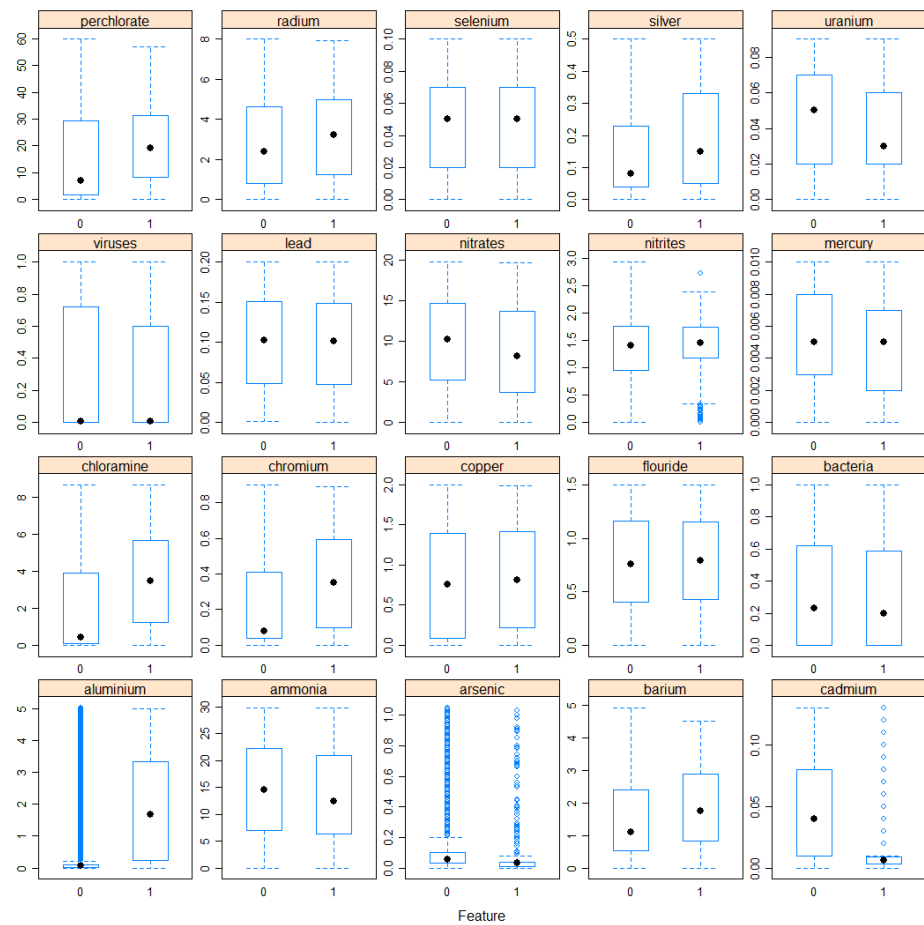


2.2.3 Grafico a torta sulla sicurezza dell'acqua

Il numero di osservazioni di acqua sicura nel training set è 632 e 4958 di acqua non sicura. Rispetto al dataset, la distribuzione di questa variabile cambia in maniera non significativa ($\sim 0,08\%$).



2.2.4 Confronto degli attributi sulla variabile target

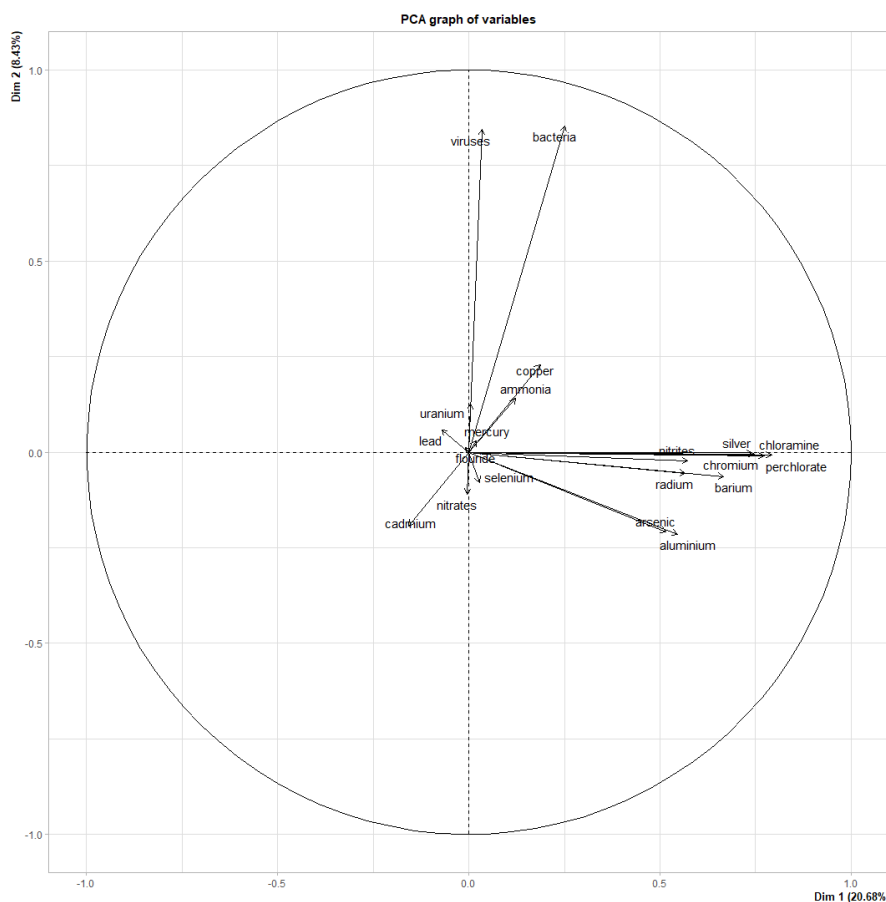


2.3 PCA

La Principal Component Analysis è un tipo di trasformazione lineare che adatta il dataset ad un nuovo spazio di coordinate riducendo lo spazio delle variabili di input tramite l'identificazione delle componenti principali, ovvero delle combinazioni lineari dei dati originali che catturano la massima varianza possibile.

Per effettuare questa operazione abbiamo standardizzato le variabili così da evitare che alcune di esse diventino dominanti a causa dei loro grandi valori rispetto ad altre variabili, rendendole così confrontabili tra di loro.

2.3.1 Variabili



Questo grafico mostra la relazione tra le variabili e le componenti più significative. La distanza tra variabili e origine misura la qualità delle variabili. Quelle più lontane dall'origine, come virus e bacteria e perchlorate, sono quelle che rappresentano meglio le dimensioni riportate. Le variabili positivamente correlate sono raggruppate insieme, quelle negativamente correlate sono posizionate in quadranti opposti.

2.3.2 Autovalori

Per determinare il numero di componenti principali da tenere dopo la PCA ci sono due criteri:

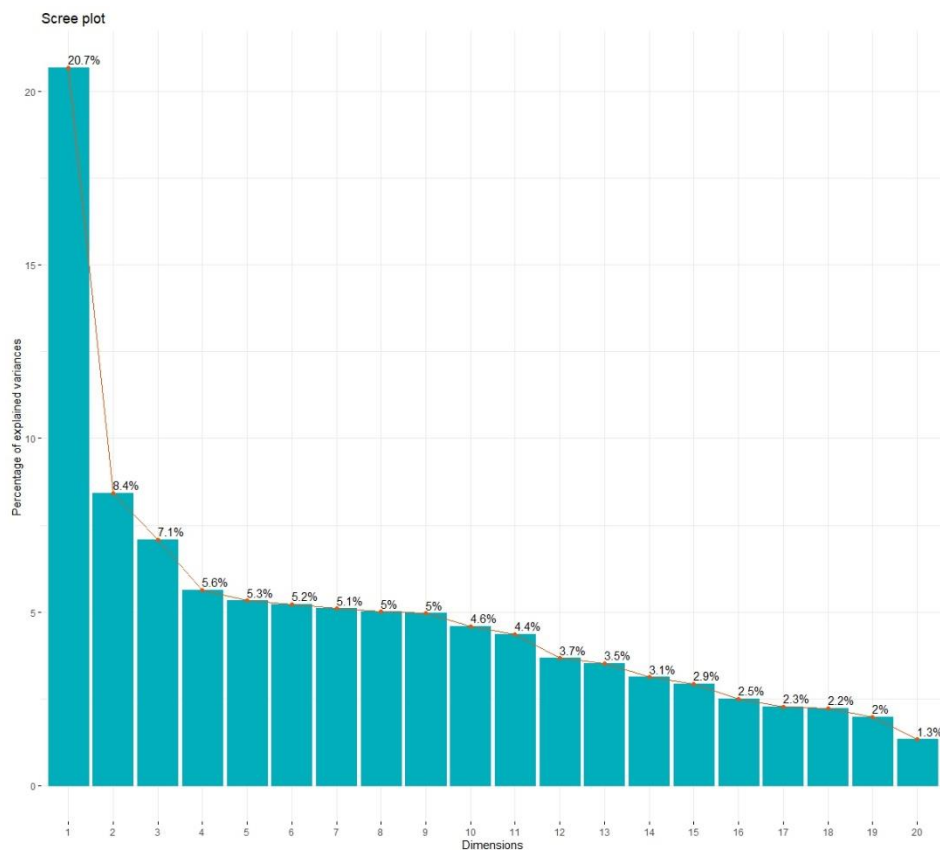
- **Autovalori > 1:** indica che le PC rappresentano una varianza maggiore di quella rappresentata da una delle variabili originali nei dati standardizzati. Questo valore viene comunemente utilizzato come punto di cutoff per la scelta delle PC da mantenere

- **Limitare il numero di componenti per varianza spiegata:** ad esempio, se si è soddisfatti del 70% della varianza totale spiegata, si può utilizzare il numero di componenti per raggiungere questo obiettivo.

Riportiamo la tabella che contiene gli autovalori e la varianza spiegata da ogni dimensione:

	Autovalore	Varianza (%)	Varianza Accumulata (%)
Dim.1	4.1352965	20.676482	20.67648
Dim.2	1.6853107	8.426553	29.10304
Dim.3	1.4161209	7.080604	36.18364
Dim.4	1.1275963	5.637982	41.82162
Dim.5	1.0662144	5.331072	47.15269
Dim.6	1.0454960	5.227480	52.38017
Dim.7	1.0201129	5.100564	57.48074
Dim.8	1.0023683	5.011842	62.49258
Dim.9	0.9937103	4.968552	67.46113
Dim.10	0.9171739	4.585870	72.04700
Dim.11	0.8714230	4.357115	76.40412
Dim.12	0.7346548	3.673274	80.07739
Dim.13	0.7053768	3.526884	83.60427
Dim.14	0.6257250	3.128625	86.73290
Dim.15	0.5876282	2.938141	89.67104
Dim.16	0.5016365	2.508183	92.17922
Dim.17	0.4557355	2.278677	94.45790
Dim.18	0.4450293	2.225147	96.68305
Dim.19	0.3956525	1.978262	98.66131
Dim.20	0.2677382	1.338691	100.00000

È possibile visualizzare questi dati con uno Scree plot.



Dal grafico possiamo notare come la varianza sia ben distribuita, questo è dovuto dalla presenza di molti attributi. In base al primo criterio di scelta avremmo dovuto tenere in considerazione solamente le prime 8 dimensioni, mentre in base al secondo sarebbero state prese in considerazione le prime 10.

2.3.3 Individui

Mostriamo ora il biplot, ovvero il grafico che mostra tutti i campioni rispetto alle componenti.

Quelli che meglio spiegano la varianza nel dataset sono quelli colorati in rosso ovvero quelli che hanno un alto valore di \cos^2 . Questo rappresenta la somma dei quadrati dei coefficienti della proiezione delle variabili originali sulla componente principale, ciascuno moltiplicato per il corrispondente valore singolare. Un basso valore di \cos^2 indica che un individuo non è ben rappresentato dai PC.

Come per le variabili, anche in questo caso gli individui migliori sono quelli che più si allontanano dall'origine. Avendo un alto numero di osservazioni non è possibile identificare i singoli campioni, ma il grafico mostra bene la loro distribuzione rispetto alle componenti principali e alle variabili.



3 Alberi di decisione (DT)

Come abbiamo già detto precedentemente, l'obiettivo del dataset scelto era quello di effettuare una classificazione binaria sulla sicurezza dell'acqua avendo a disposizione 20 attributi. Abbiamo quindi ritenuto che gli alberi di decisione fossero una buona scelta per questo tipo di compito.

3.1 Motivazioni della scelta del modello

Gli alberi di decisione sono un modello di machine learning supervisionato che si presta bene a lavori di classificazione.

Tra i suoi principali vantaggi che ci hanno portato a scegliere questo modello c'è la possibilità di fare una sua rappresentazione visuale, utile a capire le scelte prese dall'algoritmo quando effettua una previsione. Inoltre, questo modello si adatta bene a problemi di grandi dimensioni risultando comunque più veloce di altri in fase di training.

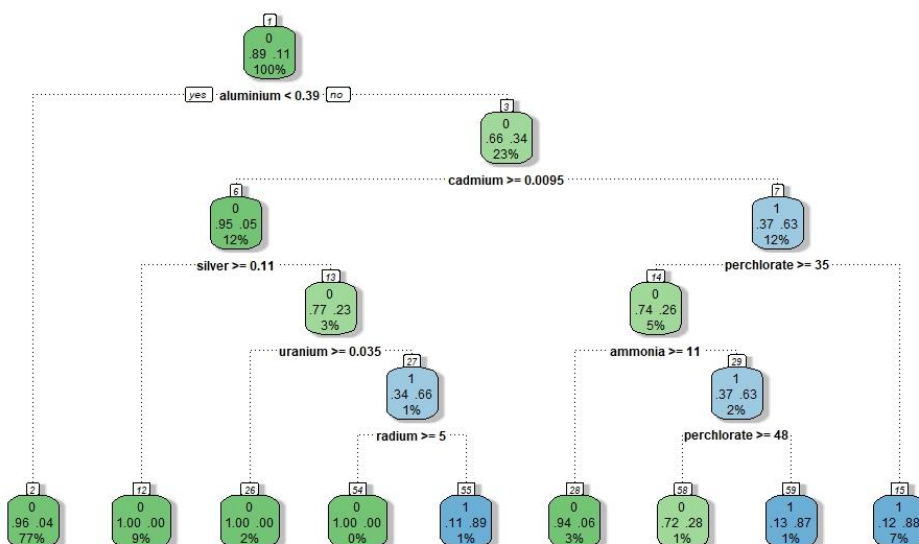
3.2 Allenamento del modello

Dopo una prima fase di allenamento del modello su tutti gli attributi abbiamo studiato il parametro di complessità (CP) per capire come alleggerire il modello.

Il parametro di complessità è il miglioramento minimo del modello necessario in ogni nodo. Si basa sulla complessità dei costi del modello ed esprime come al crescere del numero di split quanto migliora o peggiora l'ultimo split.

Dal grafico qui sopra è possibile notare come al variare del numero di split (e del CP) l'errore relativo cambia in modo più o meno significativo. Studiando questi dati abbiamo ricercato il valore CP che minimizza questo errore ricreando l'albero effettuando delle "potature" dei nodi, così da rendere il nostro modello meno sensibile all'overfitting. Nel nostro caso però l'errore si minimizza quando la dimensione dell'albero è maggiore, quindi non viene effettuato alcun cambiamento.

Di seguito mostriamo il grafico dell'albero allenato




3.3 Misure di performance

Una volta allenato il modello abbiamo effettuato delle previsioni sul Test Set così da poter valutare la bontà del modello

3.3.1 Matrice di confusione e relative metriche di performance

		Predicted class	
		Campioni sicuri	Campioni non sicuri
Actual class	Campioni sicuri	180	98
	Campioni non sicuri	34	2083


Accuratezza	0.9449
Precisione	0.8411
Recall	0.6475
F-measure	0.7317
Sensibilità	0.6475
Specificità	0.9839

 **Accuracy:** $\frac{TP+TN}{TP+TN+FP+FN}$


- Rapporto tra previsioni corrette rispetto al totale delle previsioni.

 **Precision:** $\frac{TP}{TP+FP}$


- Rapporto tra previsioni positive effettivamente corrette rispetto al totale delle previsioni positive

 **Recall:** $\frac{TP}{TP+FN}$

- Rapporto tra previsioni positive corrette rispetto al totale delle osservazioni positive.

 **F-measure:** $\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

- Media ponderata tra *Precision* e *Recall*

 **Sensitivity:** $\frac{TP}{TP+FN} \rightarrow \text{Recall}$

- Rapporto tra previsioni positive effettivamente corrette rispetto al totale delle previsioni positive $\rightarrow \text{Recall}$

 **Specificity:** $\frac{TN}{TN+FP}$

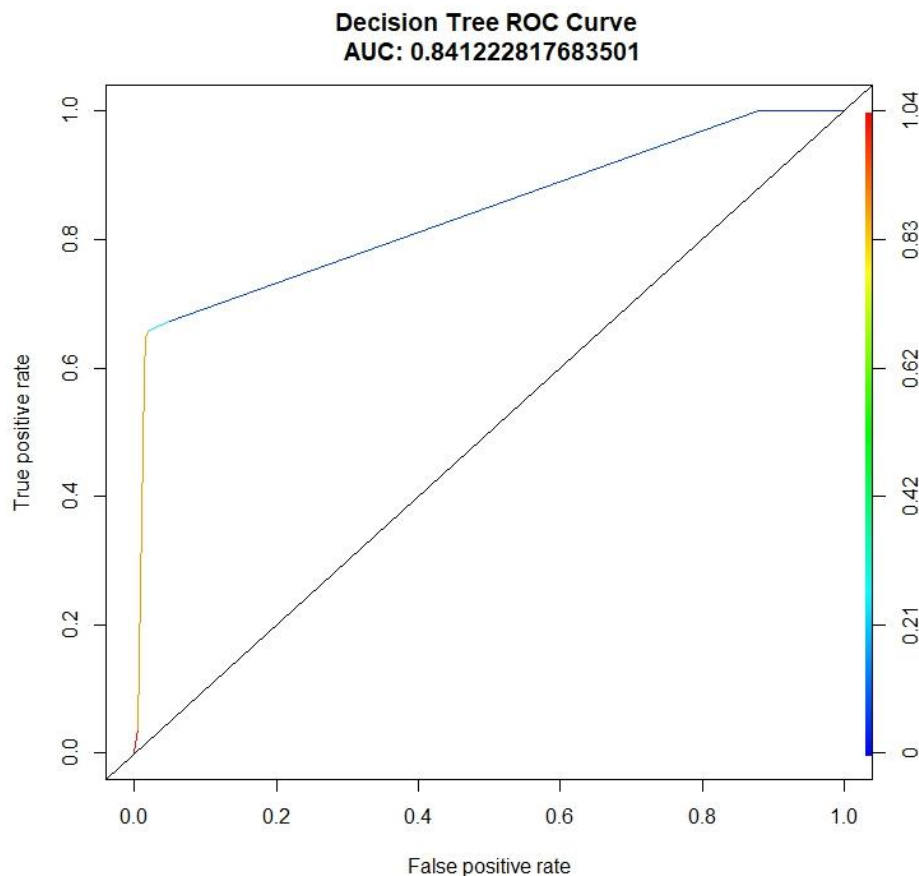
- Rapporto tra previsioni negative effettivamente corrette rispetto al totale delle previsioni negative

Queste metriche assumono tutte valore compreso tra 0 e 1, l'obiettivo è quello di cercare di avere delle valutazioni più vicine possibile ad 1.

Possiamo quindi dire che l'albero di decisione creato ha delle ottime performance su tutte le metriche, ad eccezione fatta sia per sensitivity che per recall dove abbiamo una valutazione decisamente più bassa, questo risultato molto probabilmente è dovuto ad una grande sproporzione tra campioni positivi e negativi.

3.3.2 Curva ROC & AUC

La curva ROC permette di capire le prestazioni di un sistema di classificazione binaria mettendo in evidenza il tasso di valori classificati come veri positivi rispetto al tasso di valori classificati come falsi positivi. Mostriamo di seguito il grafico della curva ROC per il modello di Decision Tree



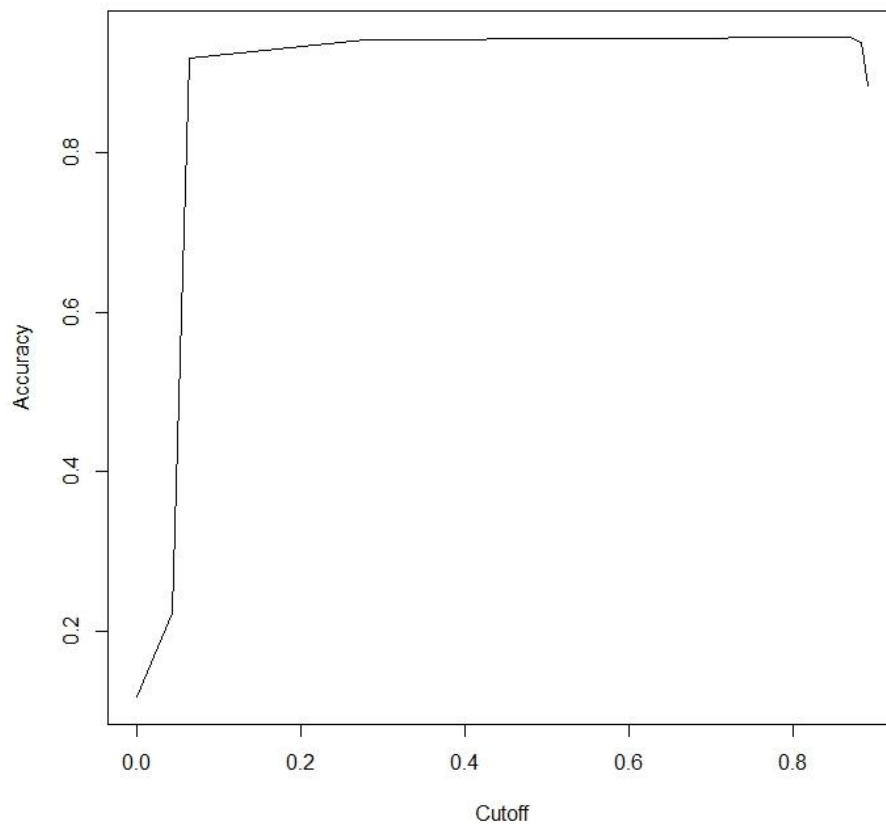
Il grafico restituisce anche il valore l'AUC ovvero area sotto la curva ROC, utile per misurare le prestazioni del modello. Dato che il valore di AUC calcolato è pari a 0.841 possiamo dire che la capacità del modello di distinguere le due classi è molto buona.

3.3.3 Cut-off

Andiamo ora a calcolare le misure di performance cambiando il cut-off ovvero il punto di decisione utilizzato dal modello per classificare un'osservazione come appartenente alla classe positiva o negativa. Esso viene trovato bilanciando i valori di sensibilità (tasso di veri positivi) e specificità (tasso di falsi positivi) che sono inversamente proporzionali.

I risultati ottenuti dalla ricerca del cut-off ottimale:

- **sensitivity:** 0.673
- **specificity:** 0.951
- **cut- off:** 0.063



Il valore massimo di accuracy trovato è di 0.945 per un indice di cut-off pari a 0.870

4 Support Vector Machine (SVM)

Support Vector Machine (SVM) è un algoritmo di classificazione supervisionato che utilizza una linea o un piano per separare i dati in classi. Sfrutta un processo per trovare la linea o il piano che ha la maggiore distanza tra i punti dei gruppi, chiamato "margine". In questo modo, SVM costruisce un modello che è in grado di classificare nuovi dati in base alla posizione rispetto alla linea o al piano di separazione.

4.1 Motivazioni della scelta del modello

Support Vector Machine è stato scelto come modello per questo progetto dato che si presta molto bene per problemi di classificazioni binaria in termini di accuratezza delle previsioni. Infatti, rispetto ad altri algoritmi, è meno suscettibile all'overfitting al netto però di una computazione più lenta. SVM, inoltre, è utilizzato per gestire grandi quantità di dati e grazie alle funzioni kernel è possibile utilizzarlo anche per dati non linearmente separabili.

4.2 Allenamento del modello

Il modello è stato allenato sul training set con la trasformazione kernel lineare ed è stato fatto in modo che l'allenamento venisse eseguito più volte, al fine di cercare il parametro di costo che restituisce l'errore minore.

Il parametro di costo permette di andare ad allargare il margine (soft-margin) o stringere il margine (hard-margin). Un parametro di costo basso porta il margine ad allargarsi per poter classificare più elementi, ma questo può portare ad errori dato che consente ai punti di stare all'interno del margine. Con un costo alto vale il contrario e serve per effettuare classificazioni più robuste.

Tra i costi analizzati, quello che ha restituito errore minore è stato il costo 5 ed è interessante notare come, all'aumentare del costo, la matrice di confusione rimanga invariata; un'ulteriore conferma sul fatto che non era possibile fare una predizione migliore di così anche stringendo sempre di più il margine. Con questo è possibile affermare che i punti non classificati correttamente siano molto vicini, o addirittura oltre, all'iperpiano separatore.

4.3 Misure di performance

Una volta allenato il modello abbiamo effettuato delle previsioni sul Test Set così da poter valutare la bontà del modello

4.3.1 Matrice di confusione e relative misure di performance

Sulla base della matrice di confusione mostriamo le misure di performance ottenute dal modello:

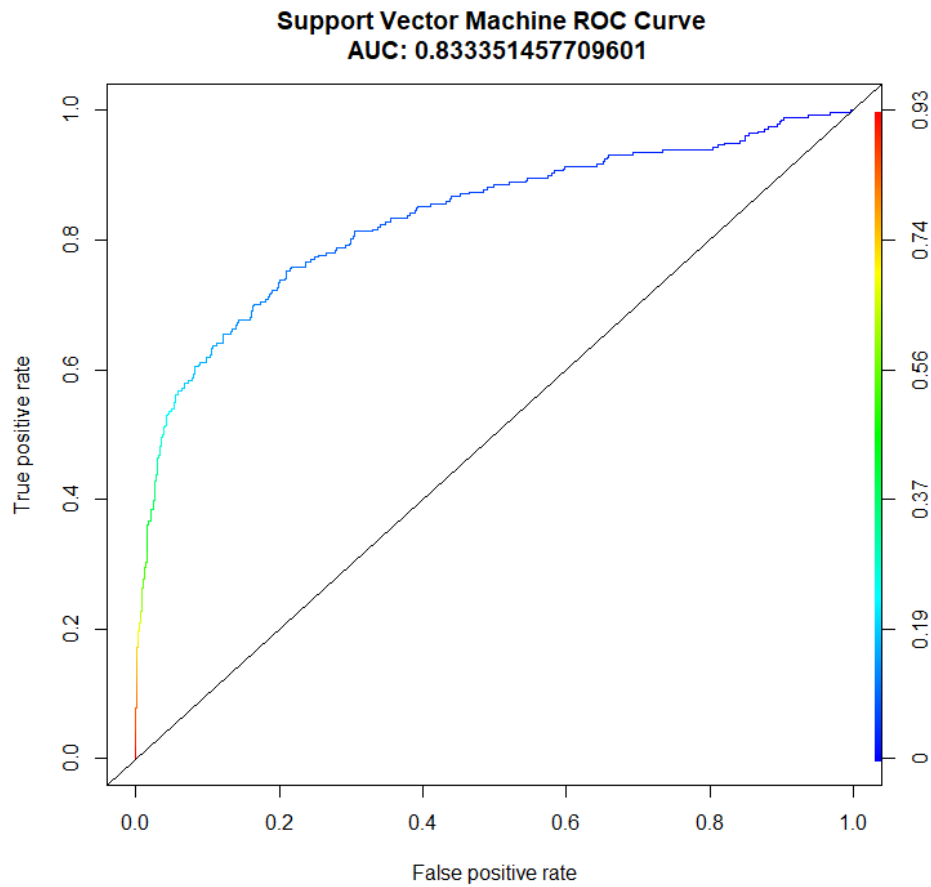
		Predicted class	
		Campioni sicuri	Campioni non sicuri
Actual class	Campioni sicuri	82	196
	Campioni non sicuri	26	2091

Accuratezza	0.9073
Precisione	0.7593
Recall	0.2950
F-measure	0.4249
Sensibilità	0.2950
Specificità	0.9877

Possiamo quindi dire che SVM ha ottime misure di performance soprattutto per quanto riguarda la classificazione dei negativi; infatti, abbiamo un valore alto per quanto riguarda la specificità. Questo è dovuto al fatto che è la classe più presente del dataset.

4.3.2 Curva ROC & AUC

Mostriamo di seguito il grafico della curva ROC per SVM.



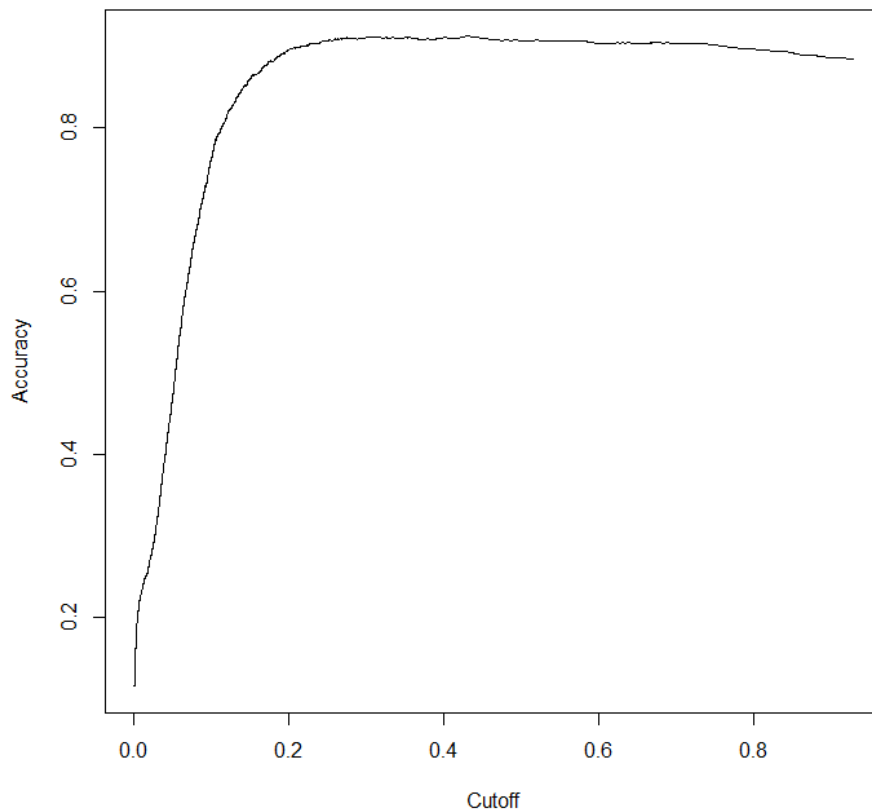
Il valore di AUC restituito è maggiore di 0.8, possiamo quindi dire che la capacità del modello di distinguere le due classi è molto buona.

4.3.3 Cut-off

Con il seguente grafico andiamo a cercare l'accuracy migliore cambiando il valore di cut-off.

I risultati ottenuti dalla ricerca del cut-off ottimale sono:

- **sensitivity:** 0.759
- **specificity:** 0.7827
- **cut-off:** 0.1033



L'accuracy migliore trovata è uguale a 0.9119 ed è stata trovata all'indice di cut-off uguale a 0.4368. L'accuratezza trovata grazie al cut-off migliora quindi di ~ 0.0047 .

5 Naive Bayes

Naive Bayes (NB) è un modello di classificazione supervisionato basato sulla teoria della probabilità. Il modello ipotizza che le variabili siano indipendenti tra loro calcolando le medie e le deviazioni standard. Quando si fa una previsione per una nuova istanza, il modello utilizza queste informazioni per calcolare la probabilità che essa appartenga a ciascuna classe così da classificarla nella classe con la probabilità più alta.

Questo processo utilizza la formula di Bayes per calcolare la probabilità a posteriori, ovvero la probabilità che un esempio appartenga ad una classe dati i valori delle sue caratteristiche.

Quindi Naive Bayes utilizza le probabilità per classificare nuovi esempi basandosi sulla distribuzione delle caratteristiche nei dati di addestramento, tutto questo utilizzando anche poche risorse computazionali.

5.1 Motivazioni della scelta del modello

Naive Bayes è stato scelto come modello per questo progetto dato che è rapido, semplice da addestrare e da comprendere, inoltre funziona bene con grandi quantità di variabili dato che ignora le relazioni tra esse che nel nostro caso sono indipendenti tra loro.

Inoltre, essendo un modello che lavora sull'ipotesi a posteriori, si sarebbe prestato bene a lavorare cercando di classificare meglio una specifica classe. Nel nostro caso sarebbe stato indicato per cercare di fare una previsione accurata sui campioni non sicuri dato che abbiamo un insieme di dati sproporzionato verso questa classe.

5.2 Allenamento del modello

Il modello è stato allenato sul training set facendo in modo che operasse su dati continui e non su variabili categoriche. Come citato in precedenza, Naive Bayes è semplice da comprendere e utilizzare infatti non sono state fatte ulteriori operazioni per l'addestramento del modello.

5.3 Misure di performance

Una volta allenato il modello abbiamo effettuato delle previsioni sul Test Set così da poter valutare la bontà del modello

5.3.1 Matrice di confusione e relative misure di performance

Sulla base della matrice di confusione mostriamo le misure di performance ottenute dal modello:

		Predicted class	
		Campioni sicuri	Campioni non sicuri
Actual class	Campioni sicuri	161	117
	Campioni non sicuri	244	1873

Accuratezza	0.8493
Precisione	0.39753
Recall	0.57914
F-measure	0.47145
Sensibilità	0.57914
Specificità	0.88474

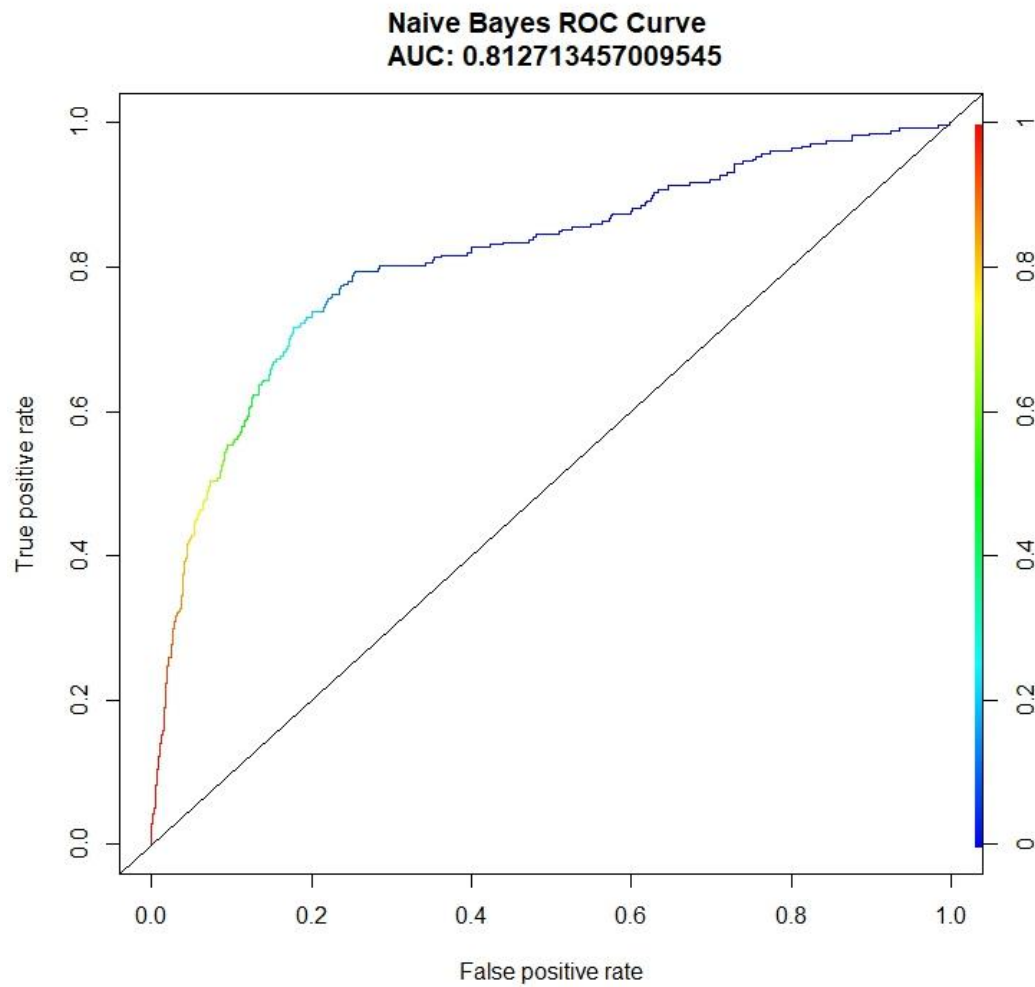
Possiamo quindi dire che NB ha discrete misure di performance.

I valori di accuratezza e specificità sono buoni. La prima metrica corrisponde al fatto che NB riesce a predire correttamente sia abbastanza valori positivi che negativi. La seconda invece è dovuta all'alto numero di valori negativi predetti correttamente.

Per quanto riguarda i valori positivi, NB ha misure di performance medie; infatti, i valori della classe positiva predetti correttamente sono all'incirca la metà.

5.3.2 Curva ROC & AUC

Mostriamo di seguito il grafico della curva ROC per NB.



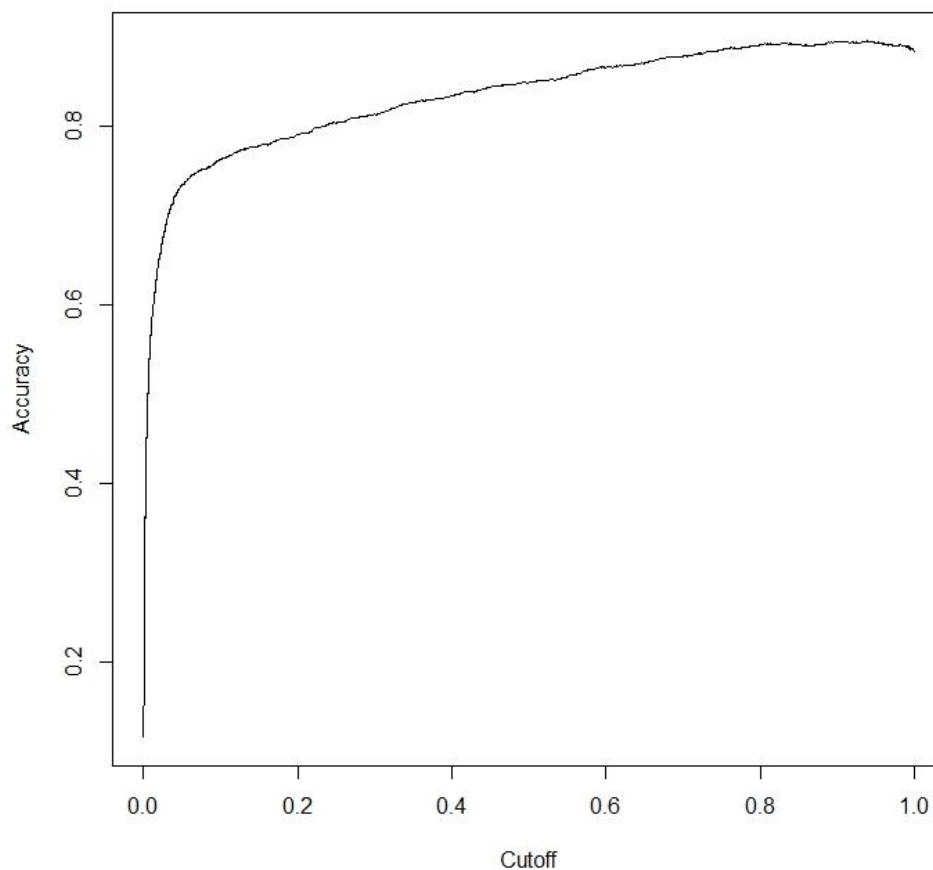
Il valore di AUC restituito è maggiore di 0.81, possiamo quindi dire che la capacità del modello di distinguere le due classi è molto buona.

5.3.3 Cut-off

Con il seguente grafico andiamo a cercare l'accuracy migliore cambiando il valore di cut-off.

I risultati ottenuti dalla ricerca del cut-off ottimale:

- **sensitivity:** 0.795
- **specificity:** 0.7454
- **cutoff:** 0.0738



L'accuracy migliore trovata è uguale a 0.896 ed è stata trovata all'indice di cut-off uguale a 0.939. L'accuratezza trovata grazie al cut-off migliora quindi di ~0.09 .

6 Comparazione dei modelli

Andiamo adesso a mettere a confronto le misure di performance dei tre modelli addestrati precedentemente.

	Alberi di decisione	Support Vector Machine	Naive Bayes
Accuratezza	0.9449	0.9073	0.8493
Precisione	0.8411	0.7593	0.39753
Recall	0.6475	0.2950	0.57914
F-measure	0.7317	0.4249	0.47145
Sensibilità	0.6475	0.2950	0.57914
Specificità	0.9839	0.9877	0.88474

Si osserva come tra i modelli usati, Decision Tree abbia performance migliori. Questo può essere dovuto al fatto che le SVM sono meno flessibili degli alberi di decisione e possono avere difficoltà a gestire dati non lineari o dati con una distribuzione complessa. Mentre Naive Bayes rispetto a Decision Tree è sensibile ai dati sbilanciati, per questo il modello sovrastima la probabilità delle osservazioni appartenenti alla classe dei negativi.

Mettiamo a confronto anche i valori di AUC per capire quale dei tre modelli distingue meglio tra la classe di campioni non sicuri e quella di campioni sicuri:

- **AUC DT:** 0.8412
- **AUC SVM:** 0.8333
- **AUC NB:** 0.8127

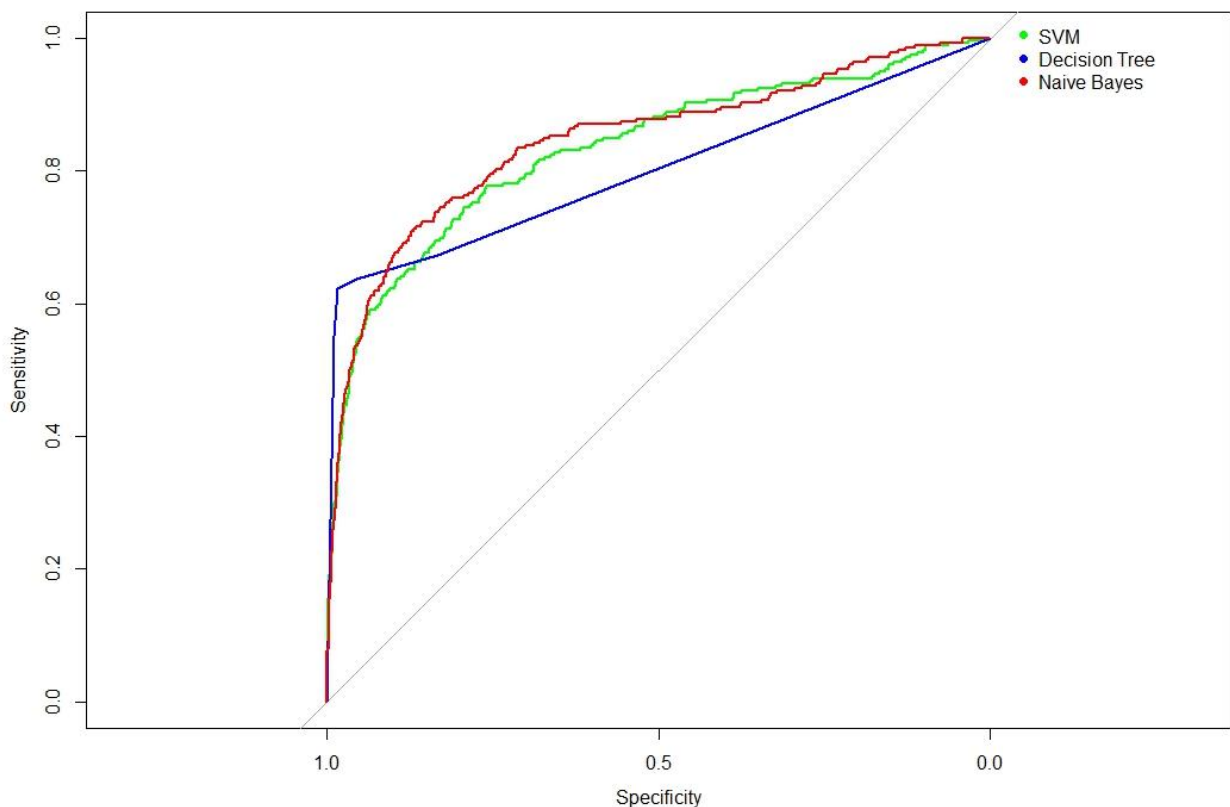
Tra i tre modelli, Decision Tree è quello con un AUC maggiore anche se di poco. Ciò significa che esso è in grado di distinguere meglio tra le due classi di interesse rispetto a SVM e NB. Questo viene confermato osservando le tre matrici di confusione, infatti nella matrice di Decision Tree sono predetti più valori positivi in maniera corretta.

7 10-Fold Cross-Validation

In questa fase abbiamo fatto un confronto tra i tre modelli precedenti addestrandoli però con il metodo 10-Fold Cross Validation.

Questo metodo consiste nel dividere il dataset in 10 parti (o “fold”) uguali e utilizzare 9 di esse come Training set, la rimanente come test. Questo processo viene ripetuto 3 volte utilizzando ogni volta un test set diverso.

Di seguito riportiamo le curve ROC di entrambi i modelli ottenute con questo metodo:

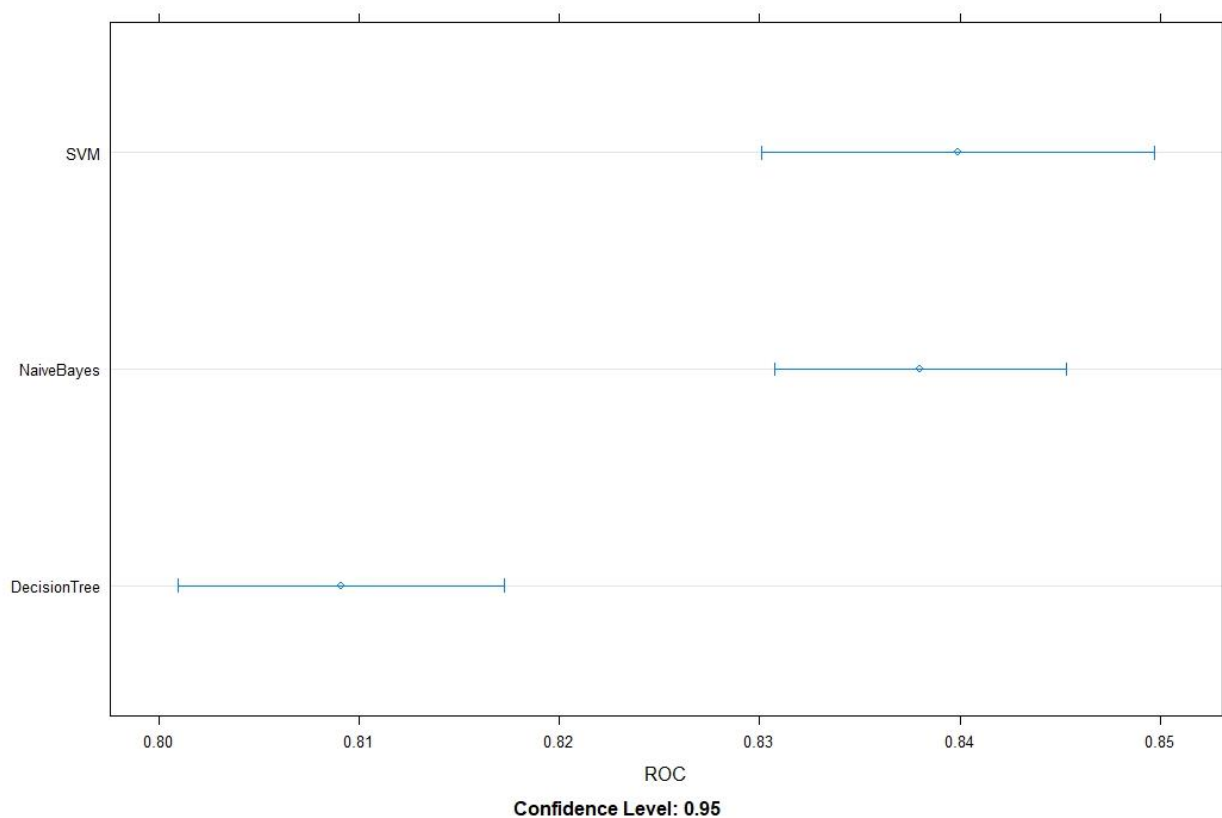


Decision Tree è rappresentato dalla curva di colore blu e ha AUC uguale a 0.801. La curva in verde invece appartiene a SVM e in questo caso abbiamo AUC uguale a 0.833. Infine, la curva rossa indica NB che ha AUC uguale a 0.846.

Con questo metodo quindi osserviamo che le situazioni sono inverse rispetto a prima; infatti, Decision Tree distingue peggio le due classi e NB è quello che le distingue meglio.

7.1 Dotplot

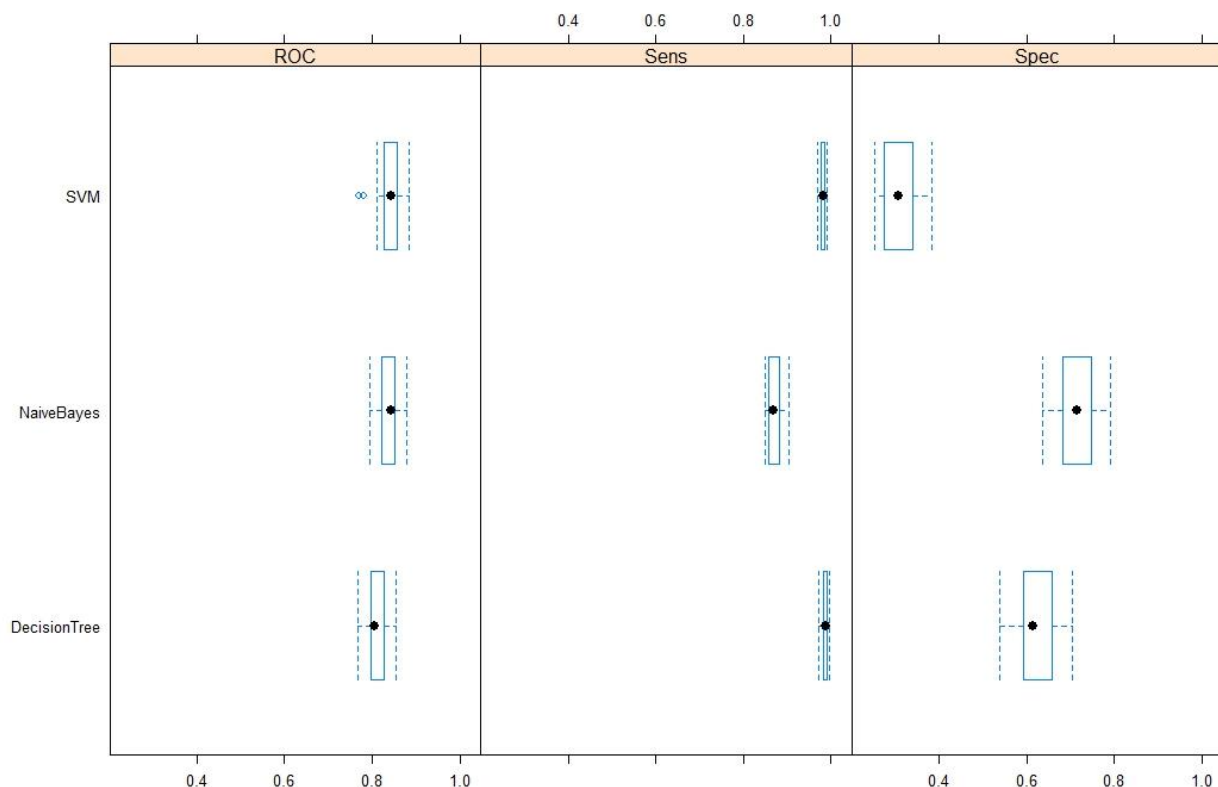
Questo grafico ci permette di capire come si comportano i modelli rispetto al valore di AUC, quando il modello è migliore esso avrà un intervallo di confidenza più piccolo oltre ad un valore medio maggiore.



In questo caso Decision Tree ha valori più bassi rispetto a SVM e NB permettendoci di dire sin da subito che DT in questo caso è il modello meno prestante tra i 3.

Gli altri due modelli hanno una sovrapposizione; Naive Bayes presenta un intervallo di confidenza più piccolo rispetto a SVM, a discapito però di un valore medio leggermente più basso.

7.2 Bwplot



In questo grafico vengono messe a confronto 3 metriche:

- ✚ La prima di queste è **ROC** che evidenzia i risultati descritti precedentemente nel dotplot
- ✚ Per quanto riguarda **sensitivity** non si riesce a notare alcuna differenza tra SVM e DT, ma si ha che NB ha sia un intervallo più ampio che dei valori minori rispetto agli altri due modelli. Questo ci permette di dire che NB ha prestazioni peggiori per quanto riguarda la predizione dei positivi.
- ✚ Infine, l'ultima metrica è **specificity** dove troviamo un intervallo maggiore per quanto riguarda DT e dei valori migliori per NB, ciò significa che quest'ultimo modello riesce a fare delle previsioni migliori per quanto riguarda la classe dei negativi

7.3 Tempi di esecuzione

Sfruttando i metodi a disposizione, è stato possibile effettuare un confronto dei tre modelli anche per quanto riguarda le tempistiche di calcolo. Da questo confronto si evince come tra i modelli appena allenati, l'albero di decisione e Naive Bayes abbiano delle tempistiche nettamente minori rispetto a SVM. In questo modo è stato confutato così uno dei pro presi in considerazione in fase di scelta di questi due modelli.

Da queste tempistiche è possibile notare anche come questa tecnica di training per SVM sia molto più efficiente rispetto alla grid search usata in precedenza per allenare questo tipo di modello.

8 Conclusioni

L'obiettivo di questo progetto è stato quello di trattare un dataset a nostra scelta con i modelli che abbiamo ritenuto più opportuni per questo tipo di classificazione.

In prima analisi è stata fatta un'esplorazione dei dati che ci ha mostrato come le variabili del dataset avessero tutte una distribuzione non normale e molto spesso diversa tra loro.

Abbiamo visto come la variabile target sia sproporzionata nei confronti della classe dei valori negativi, dandoci così un indizio sul fatto che i modelli allenati avrebbero restituito prestazioni migliori per quanto riguarda questa classe. Dopo l'allenamento dei modelli scelti e la misurazione delle loro performance abbiamo potuto verificare questa ipotesi.

Inoltre, dalle misurazioni delle performance dei modelli usati abbiamo ottenuto degli esiti in generale soddisfacenti. In particolare, Decision Tree risulta essere di poco migliore rispetto a SVM per quanto riguarda quasi tutte le metriche di classificazione valutate, mentre Naive Bayes presenta invece i risultati nel complesso peggiori.

In conclusione, il progetto ha dimostrato l'efficacia dell'utilizzo di modelli di machine learning come il support vector machine, gli alberi di decisione e Naive Bayes per effettuare classificazioni binarie. Inoltre, ci ha permesso di capire come anche il modo di addestrare un modello di machine learning possa influire sulle sue performance.