a.a. 2022/2023

PROGETTO MACHINE LEARNING

Classificazione basata su caratteristiche del volto

SOMMARIO

1 DESCRIZIONE DEL DATASET

Il dataset selezionato per questo progetto permette di suddividere un campione di soggetti in base in due generi (maschio o femmina), basandosi su alcune caratteristiche fisiche del volto.

Il dataset contiene 5000 istanze, ognuna delle quali è caratterizzata da 7 covariate e dalla classificazione in maschio o femmina. In particolare, il dataset è diviso equamente tra maschi e femmine.

Non è stato necessario effettuare alcuna operazione di pulizia dei dati, in quanto non manca alcun valore.

Il dataset utilizzato è il seguente: https://www.kaggle.com/datasets/elakiricoder/gender-classification-dataset.

Le covariate che descrivono il dataset sono le seguenti:

- *long_hair*: covariata booleana che indica se il soggetto ha i capelli lunghi (1) o corti (0)
- *forehead_width_cm*: covariata numerica che misura in cm la larghezza della fronte
- *forehead_height_cm*: covariata numerica che misura in cm l'altezza della fronte
- *nose_wide*: covariata booleana che indica se il soggetto ha il naso largo (1) o no (0)
- *nose_long*: covariata booleana che indica se il soggetto ha il naso lungo (1) o no (0)
- *lips_thin*: covariata booleana che indica se il soggetto ha le labbra sottili (1) o no (0)
- *distance_nose_to_lip_long*: covariata booleana che indica se il soggetto ha tanto spazio tra il naso e le labbra (1) o no (0)
- *gender*: covariata target categorica che indica se il soggetto è maschio (male) o femmina (female)

Le covariate booleane sono stabilite in basi a parametri non di nostra conoscenza (ad esempio non sappiamo quale lunghezza per i capelli è considerata il valore soglia per decidere se la covariata viene messa a 1 o a 0).

Per poter analizzare meglio il dataset decidiamo di lasciare le covariate di tipo booleano come attributi di tipo numerico all'interno del dataset utilizzato su RStudio.

Per questa relazione utilizzeremo la seguente codifica di colori per rappresentare maschi e femmine:

2 ANALISI DELLE COVARIATE

Per comprendere meglio il dataset utilizziamo due tipologie di analisi:

2.1 Univariate: per comprendere una singola covariata
2.2 Multivariate: per comprendere la relazione tra le covariate

2.1 UNIVARIATE

Effettuiamo un'analisi di ogni covariata per esplorare il dataset e stabilire quali variabili crediamo saranno più significative per classificare ogni istanza, quindi stabilire se il soggetto è maschio o femmina.

Per questo tipo di analisi lasciamo tutte le covariate booleane come variabili di tipo numerico all'interno di RStudio, senza renderle categoriche. In questo modo riusciamo a calcolarne gli indici e a rappresentare i grafici, tenendo sempre in considerazione che sono booleane.

2.1.1 INDICI DI STATISTICA DESCRITTIVA

Ricaviamo gli indici di statistica descrittiva, in particolare calcoliamo:

- Minimo
- Primo quartile
- Mediana
- Media
- Terzo quartile
- Massimo

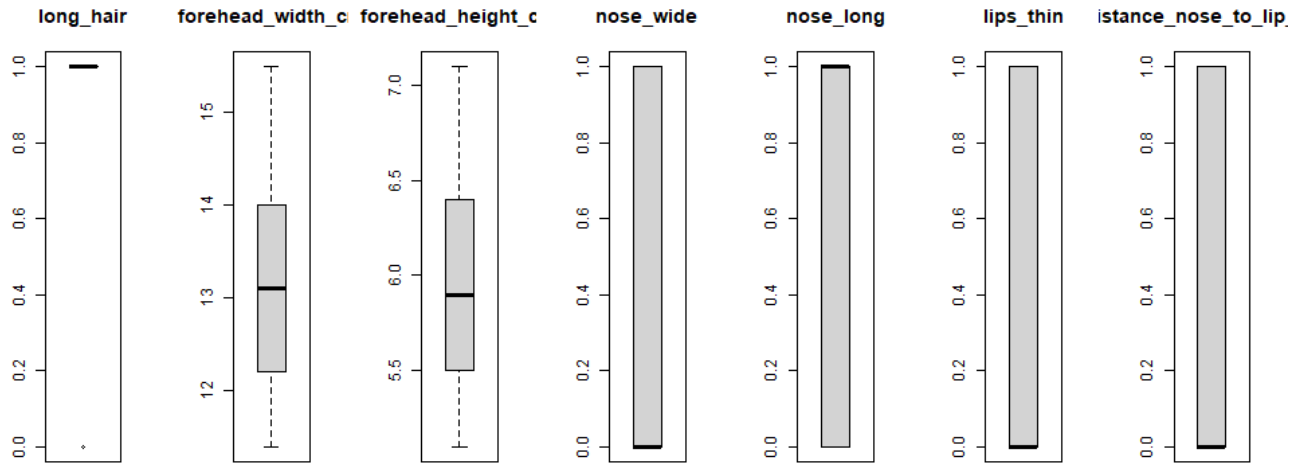
Sappiamo che nell'analisi dei valori booleani la media indicherà esattamente quale percentuale di istanze ha valore 1 per quella determinata covariata.

```
   long_hair      forehead_width_cm forehead_height_cm   nose_wide
Min.   :0.0000   Min.   :11.40     Min.   :5.100      Min.   :0.0000
1st Qu.:1.0000   1st Qu.:12.20     1st Qu.:5.500      1st Qu.:0.0000
Median :1.0000   Median :13.10     Median :5.900      Median :0.0000
Mean   :0.8696   Mean   :13.18     Mean   :5.946      Mean   :0.4939
3rd Qu.:1.0000   3rd Qu.:14.00     3rd Qu.:6.400      3rd Qu.:1.0000
Max.   :1.0000   Max.   :15.50     Max.   :7.100      Max.   :1.0000

   nose_long        lips_thin     distance_nose_to_lip_long
Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
Median :1.0000   Median :0.0000   Median :0.0000
Mean   :0.5079   Mean   :0.4931   Mean   :0.4989
3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

2.1.2 BOXPLOT

Per aiutarci ad analizzare meglio i dati appena calcolati decidiamo di creare i relativi grafici, ovvero i boxplot.

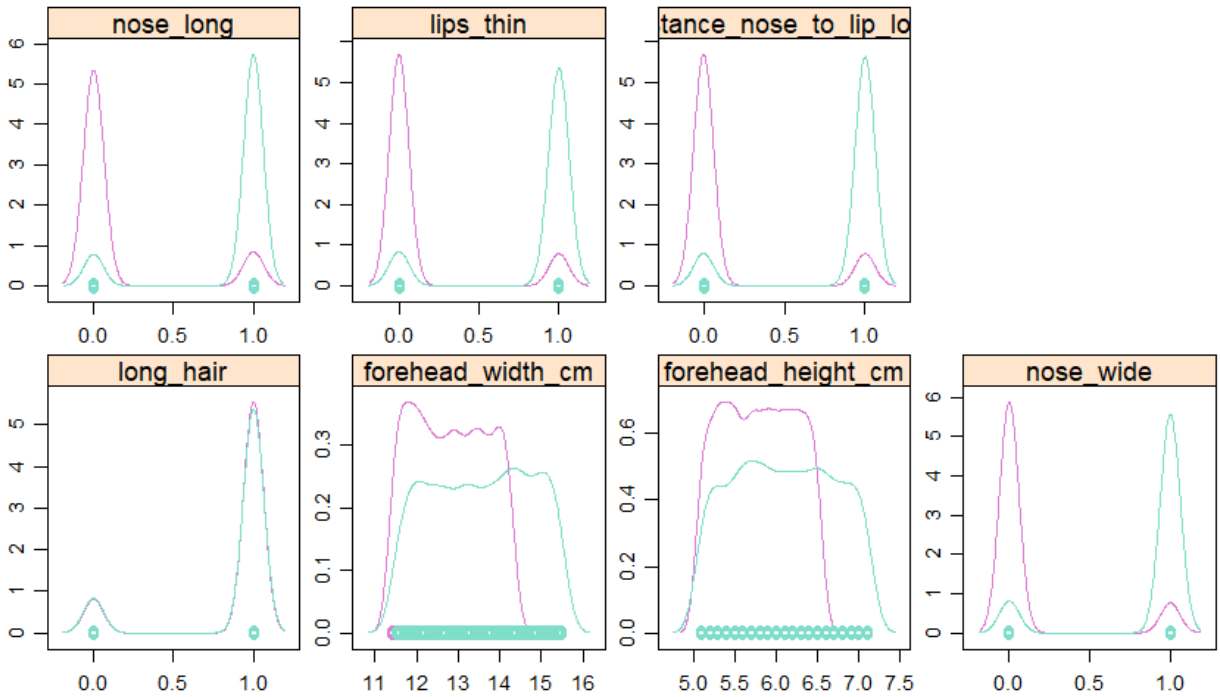


Da questi dati e con il supporto dei boxplot possiamo ricavare che:

- *long_hair* ha una maggioranza di istanze con valore 1, quindi la maggior parte dei soggetti ha i capelli lunghi, infatti la media è all'86,96%. Il boxplot, inoltre, mostra che il terzo quartile ha valore 1, quindi almeno il 75% dei dati hanno valore 1.
- I dati relativi a *forehead_width_cm* e *forehead_height_cm* sono equamente distribuiti tra i valori raccolti. Il boxplot, infatti, mostra che tutti i quartili coprono all'incirca lo stesso range di valori.
- Le altre covariate booleane sono equamente distribuite, come possiamo notare dalla media che è sempre vicina al 50%. Il boxplot mostra che ogni valore (0 o 1) ha una distribuzione compresa tra il 25% e il 75%, come anticipato dalla media.

2.1.3 DENSITY PLOT

Per avere un'idea più chiara di come le covariate influiscono sulla classificazione dell'istanza in maschio o femmina rappresentiamo ognuna di queste con un density plot.



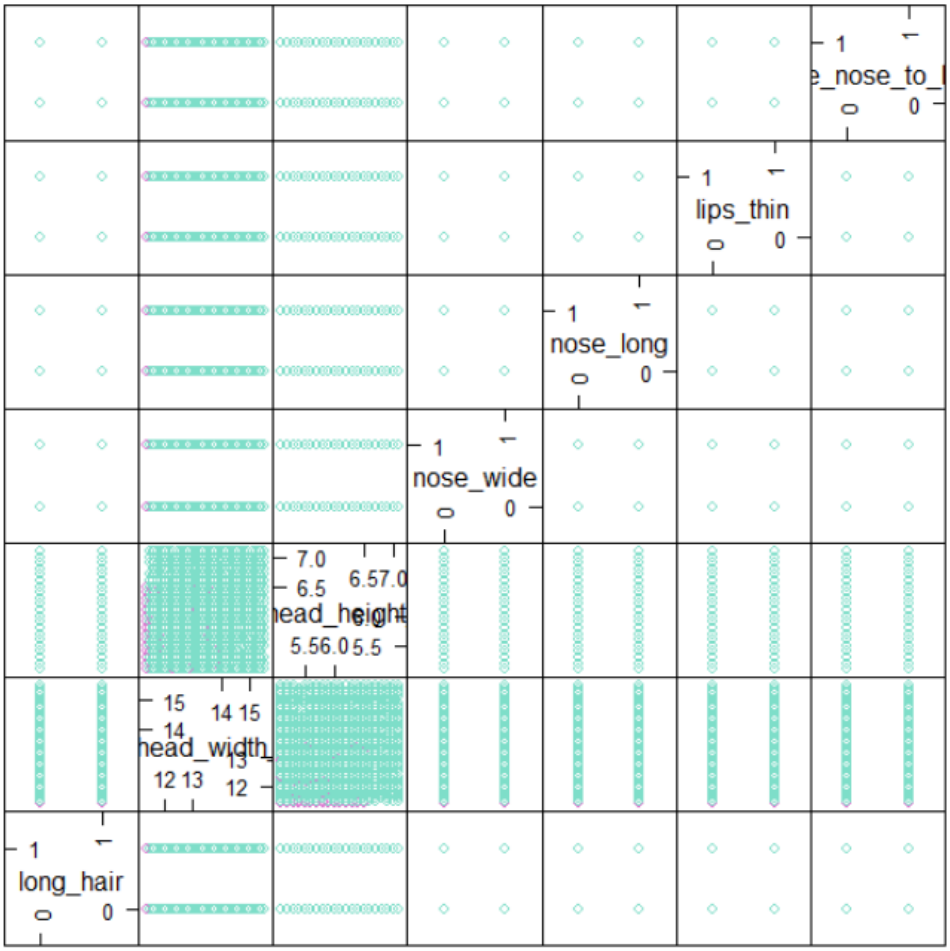
Da questi density plot possiamo ricavare che:

- La covariata *long_hair*, come già notato sopra, non può rappresentare bene la classificazione in maschio o femmina. Infatti, abbiamo un'alta percentuale di soggetti con i capelli lunghi, mentre la percentuale di maschi e femmine è esattamente a metà. Inoltre, grazie a questo grafico, possiamo anche notare che non c'è differenza tra il numero di maschi e di femmine con i capelli corti e neanche tra il numero di maschi e femmine con i capelli lunghi. Possiamo quindi dire che *long_hair* non sarà utile per classificare le istanze.
- Tutte le variabili booleane (*long_hair* esclusa) hanno una distribuzione molto simile tra loro, in particolare si nota che la maggioranza delle istanze di soggetti maschi ha tutte e quattro queste covariate con valore 1 (quindi true), viceversa per le femmine. Possiamo quindi dire che queste variabili ci serviranno sicuramente a classificare le istanze.
- Le covariate numeriche hanno invece una distribuzione simile sia per maschi che per femmine. Questa similitudine non è estrema come per *long_hair*, ma, rispetto alle altre covariate booleane, può darci poche informazioni sulla classificazione tra maschio e femmina.

2.2 MULTIVARIATE

Per verificare se alcune covariate hanno una stretta correlazione con altre creiamo una matrice di scatterplot (grafici di dispersione). Prima di effettuare questa operazione trasformiamo tutte le variabili booleane in variabili di tipo categorico all'interno di RStudio.

La matrice, come ci aspettavamo, risulta poco utile nel dividere i soggetti tra maschi e femmine. Nonostante questo, ne possiamo comunque ricavare alcune interessanti considerazioni.



In particolare, ci sono tre tipi di relazioni tra covariate:

- Boolena in relazione con booleana: i dati si possono distribuire solo nei quattro angoli. Tutti gli scatterplot che rappresentano questa relazione hanno esattamente tutti gli angoli coperti da almeno un'istanza, possiamo quindi dedurre che non c'è una correlazione stretta tra nessuna delle covariate booleane.
- Booleana in relazione con numerica: i dati si possono distribuire solo sull'asse che rappresenta la covariata booleana. Tutti i relativi scatterplot hanno i dati sull'asse interamente e uniformemente distribuiti, possiamo quindi dedurre che non c'è una correlazione stretta tra nessuna di queste covariate.
- Numerica in relazione con numerica: i dati si possono distribuire sull'intero grafico. Lo scatterplot relativo ha tutta la superficie uniformemente coperta, possiamo quindi dedurre che non c'è una correlazione stretta tra le due covariate.

Dall'analisi multivariata possiamo quindi dedurre che nessuna covariata presenta una stretta relazione con nessun'altra covariata.

3 MODELLI DI MACHINE LEARNING

Per poter analizzare il dataset lo suddividiamo in due sottogruppi:

- Training set: contiene il 70% delle istanze del dataset e viene utilizzato per l'addestramento dei modelli
- Test set: contiene le istanze rimanenti del dataset e viene utilizzato per testare l'accuratezza dei modelli

Verifichiamo che nel training set le istanze siano distribuite equamente tra maschi e femmine, ottenendo che la distribuzione rimane molto vicina a quella del dataset. Verifichiamo questo dato per evitare di avere una forte predominanza di maschi o femmine nel training set, che ridurrebbe l'efficacia dell'addestramento dei modelli.

All'interno del test set aggiungiamo una covariata, chiamata *prediction*, che permette di applicare ogni modello addestrato e verificarne l'accuratezza. Per confrontare più facilmente la covariata *prediction* con il valore del target gender decidiamo di rappresentare all'interno del test set i maschi con il valore 0 e le femmine con 1.

3.1 BASELINE MODEL

Utilizziamo il baseline model come primo modello perché, oltre ad essere molto semplice da costruire, fornisce molte informazioni che possono tornare utili nelle fasi successive.

Inizializziamo la covariata *prediction* per ogni istanza in modo casuale (tra 0 e 1). Decidiamo di farlo in modo completamente casuale, dato che il training set è equamente distribuito tra maschi e femmine.

L'accuracy del modello varia ogni volta che vengono assegnati i valori alla covariata *prediction*, dato che i valori vengono assegnati in modo casuale. In particolare, notiamo che l'accuracy varia approssimativamente tra **0.48** e **0.52**.

Otteniamo il risultato atteso, ovvero vicino al 50%. Non considerando altre covariate non potremmo ottenere un risultato migliore. Anche assegnando a tutte le istanze lo stesso valore della covariata *prediction*, otterremmo comunque il 50% di accuracy.

Selezioniamo poi una covariata che, dall'analisi delle univariate, ha dimostrato di spiegare bene il dataset. In particolare, decidiamo di utilizzare la variabile *distance_nose_to_lip_long*.

Per verificare che *distance_nose_to_lip_long* discrimini il training set in maniera efficiente, controlliamo la tabella che mostra la relazione tra questa variabile e il genere.

gender / distance_nose_to_lip_long	Femmina	Maschio
0	0.8803419	0.1196581
1	0.1191977	0.8808023

Basandoci su questi risultati, attribuiamo alla covariata *prediction* del test set il valore 0 (maschio) se *distance_nose_to_lip_long* è 1, femmina altrimenti. Da questa analisi ci aspettiamo di ottenere un'accuracy vicina all'88%. Verifichiamo quindi la matrice di confusione, la cui accuracy è **0.87**.

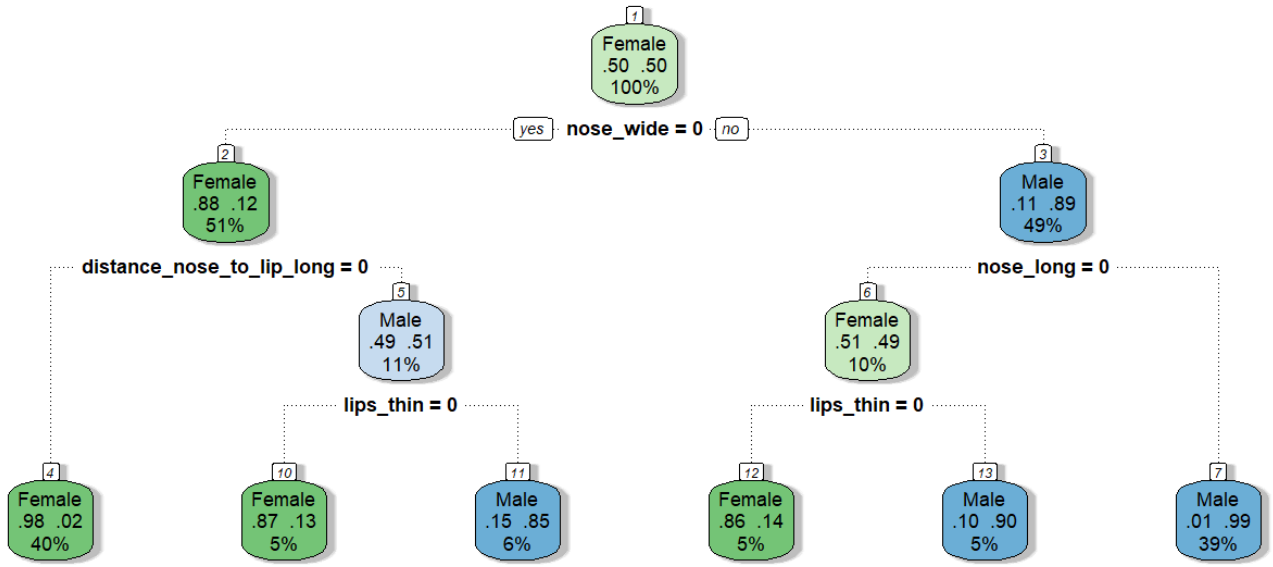
Target / Prediction	0 (maschio)	1 (femmina)
0 (maschio)	654	99
1 (femmina)	96	651

Riteniamo che il risultato di questo modello sia più che soddisfacente, considerando che è stato ottenuto utilizzando i valori di una sola covariata.

3.2 ALBERI DI DECISIONE

Scegliamo ora di utilizzare gli alberi di decisione, un modello semplice da comprendere e veloce sia nell'addestramento che nell'utilizzo.

3.2.1 ALBERO DI DECISIONE CON 6 FOGLIE

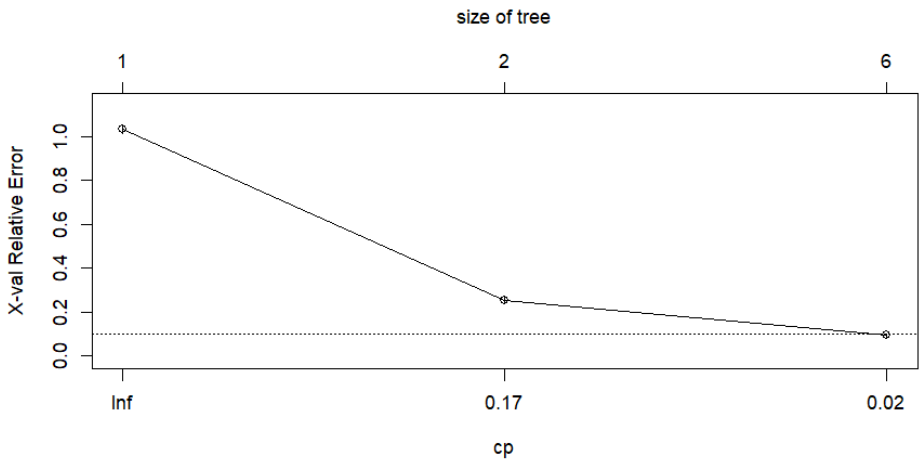


Come si può notare ed era stato già previsto, l'albero ha dato priorità alle covariate booleane, che ripartiscono molto meglio il dataset rispetto a quelle numeriche. Tra queste non è presente *long_hair*, che, come avevamo già detto, non divide bene il dataset in maschi e femmine.

Riportiamo la matrice di confusione relativa all'albero, la cui accuracy è di **0.9613**.

Target / Prediction	**0 (maschio)**	**1 (femmina)**
0 (maschio)	724	29
1 (femmina)	29	718

Ora analizziamo il grafico del complexity parameter (cp).



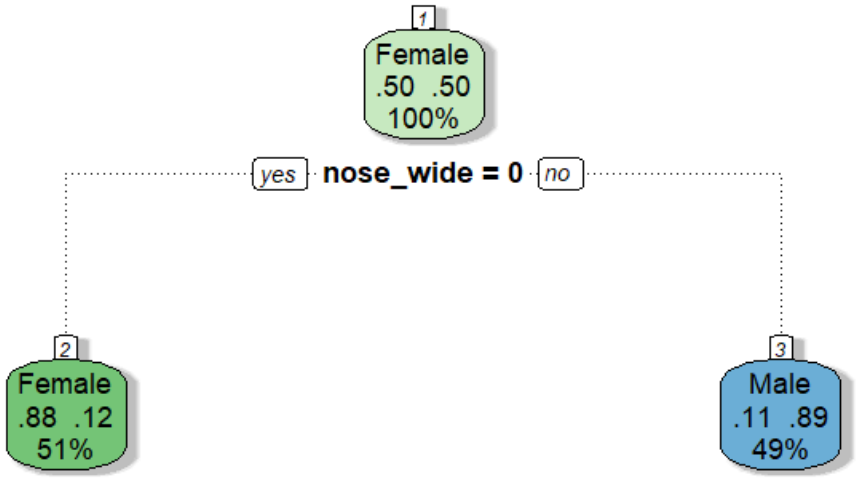
Notiamo che l'unica alternativa all'albero attuale è quella di farne uno con sole 2 foglie. Notiamo fin da subito che questo albero avrà un peggioramento del tasso di errore significativo, nonostante il guadagno in termini di dimensioni dell'albero non sia molto, dato che partiamo già da un albero piuttosto semplice.

Non riteniamo che l'alta accuracy sia dovuta a un caso di overfitting, dato che l'albero è comunque piccolo.

3.2.2 ALBERO DI DECISIONE CON 2 FOGLIE

Decidiamo comunque di costruire l'albero con 2 sole foglie per verificarne l'accuratezza.

Creiamo il grafico, lo mettiamo nel test set, e ne calcoliamo l'accuracy.



Otteniamo la seguente matrice di confusione, la cui accuracy è **0.8753**.

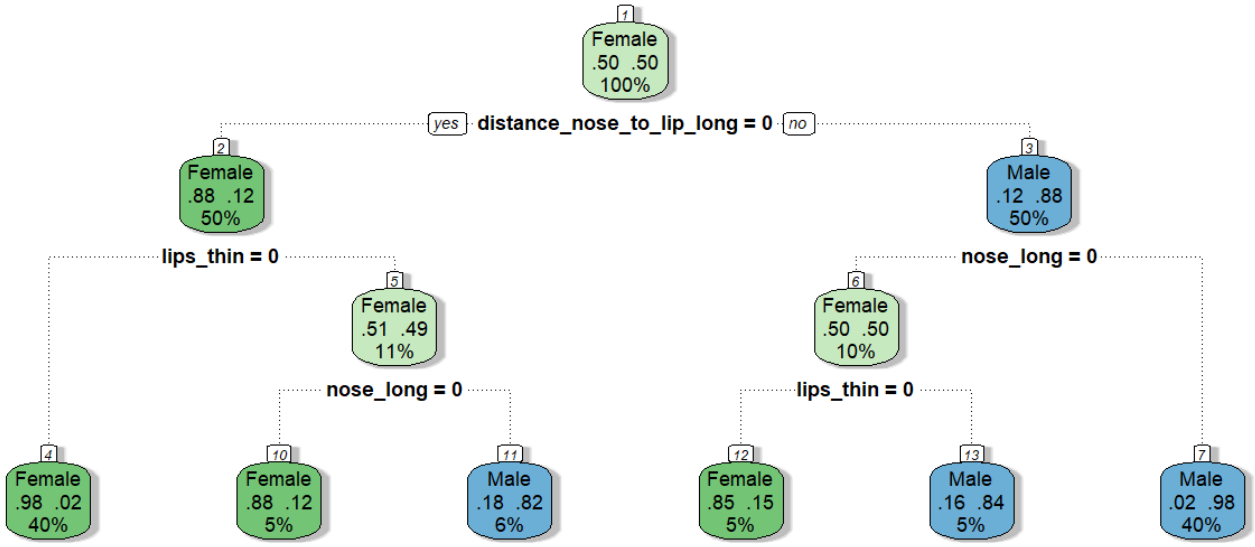
Target / Prediction	**0 (maschio)**	**1 (femmina)**
0 (maschio)	655	98
1 (femmina)	89	658

L'accuracy è sostanzialmente la stessa che abbiamo ottenuto dividendo il database sulla base della singola covariata *distance_nose_to_lip_long* (0.87). Infatti, stiamo ripetendo in pratica lo stesso processo, con l'unica differenza della covariata scelta per suddividere il dataset. In questo caso la variabile scelta (*nose_wide*) è quella che divide meglio il dataset, mentre precedentemente era stata scelta arbitrariamente. Nonostante ciò, la differenza risulta minima.

3.2.3 ALBERO DI DECISIONE CON INFORMATION GAIN

Proviamo ora a impostare come criterio di selezione dei nodi dell'albero quello con maggiore information gain. In questo modo viene ricavato un albero identico a quello iniziale, ottenendo una conferma del fatto che l'albero che ne risulta è stato selezionato con un buon criterio.

Proviamo anche a creare l'albero basandoci solo sulle covariate booleane *distance_nose_to_lip_long*, *nose_long* e *lips_thin*. Decidiamo di escludere la variabile *nose_wide*, che nell'albero precedente era stata selezionata come radice. In questo modo otteniamo il seguente albero.



Si ottiene la seguente matrice di confusione, la cui accuracy è di **0.9586667**.

Target / Prediction	**0 (maschio)**	**1 (femmina)**
0 (maschio)	721	32
1 (femmina)	30	717

Da questo albero possiamo dedurre che, anche riducendo il numero di covariate analizzate, otteniamo ottimi risultati. In particolare, notiamo che l'accuracy è peggiorata di meno dell'1%.

3.3 RETE NEURALE

Proviamo ora ad utilizzare una rete neurale, un modello di machine learning più difficile da comprendere rispetto agli alberi di decisione e più lento nell'addestramento, ma comunque rapido nell'utilizzo. A differenza degli alberi di decisione ha però una buona tolleranza delle istanze contraddittorie.

Per poter utilizzare la rete neurale è necessario trasformare nel training set tutte le variabili booleane in variabili numeriche, dove 0 rappresenta false e 1 rappresenta true.

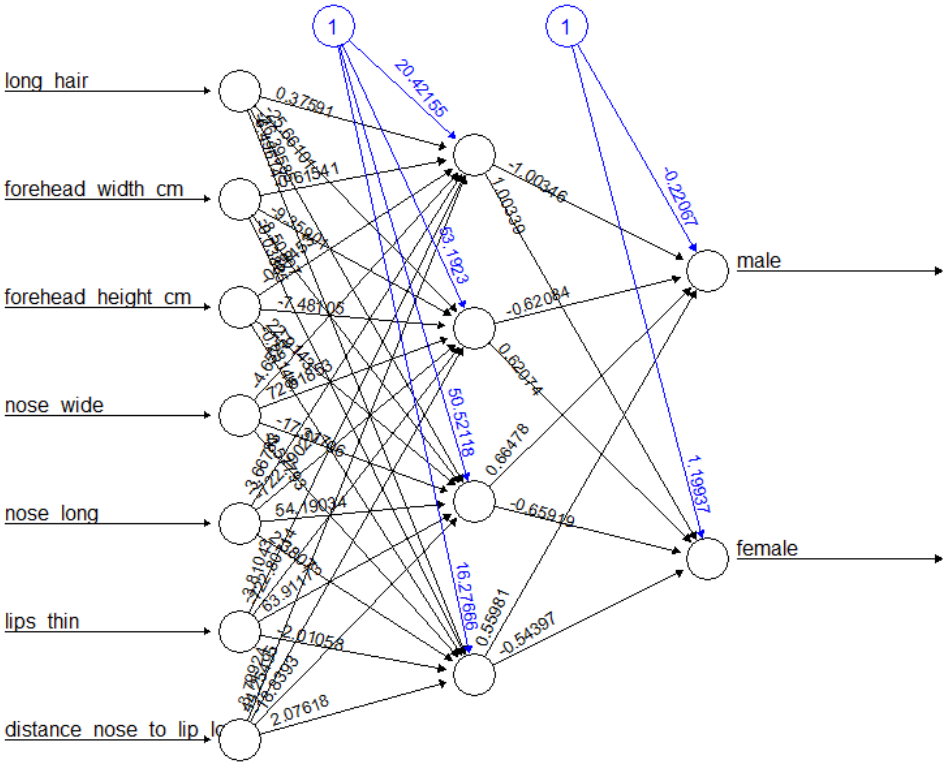
Aggiungiamo inoltre due colonne target al training set: *male* e *female*, dove il valore dell'attributo *male* per un'istanza è true se il soggetto è maschio, false altrimenti (viceversa per *female*).

La rete neurale che si ottiene ha in input un neurone per ogni covariata e in output un neurone per ogni colonna target (*male* o *female*).

Per decidere quanti neuroni nascosti utilizzare in una rete neurale non esiste un modo definito per calcolare il numero ottimale. Nonostante questo, ci sono comunque alcune indicazioni per stabilire un buon valore, in particolare la regola più generale è che il numero di neuroni nascosti è compreso tra il numero di neuroni in input e il numero di neuroni in output. Inoltre, abbiamo trovato una regola, citata in più fonti, che suggerisce di applicare la seguente formula:

$$numero\ di\ neuroni\ nascosti = \sqrt{numero\ di\ neuroni\ in\ input * numero\ di\ neuroni\ in\ output} = \sqrt{7 * 2} \sim 4$$

Utilizzando quattro neuroni nascosti, diamo quindi origine alla seguente rete neurale:

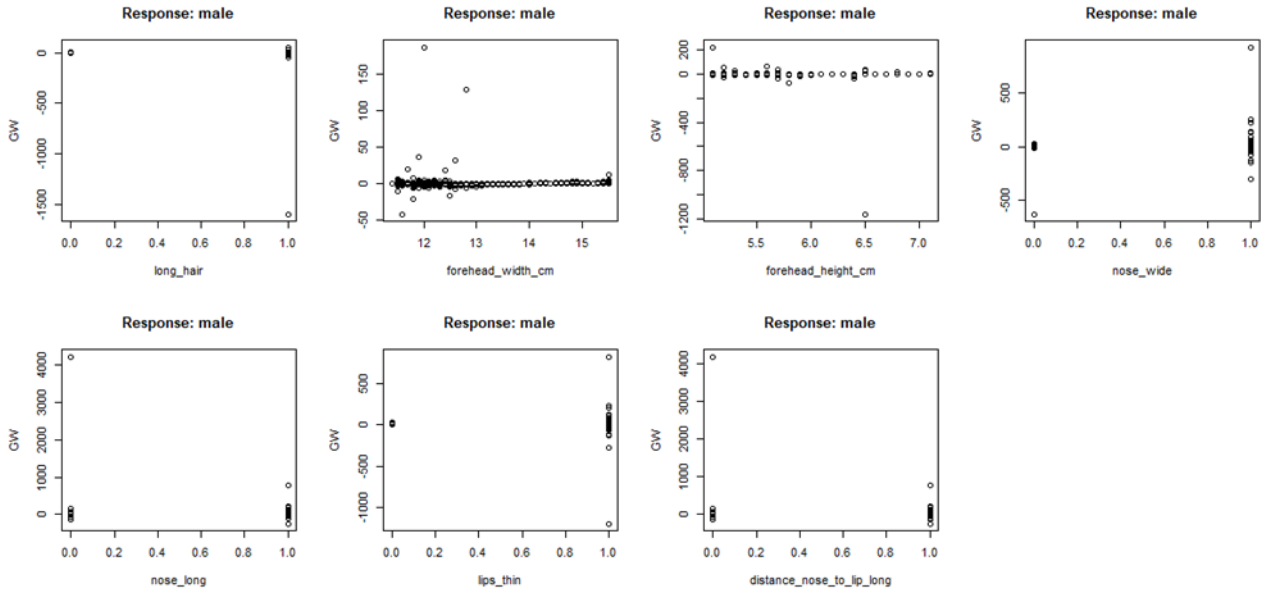


Error: 72.43714 Steps: 42786

Si deve considerare che la rete neurale ottenuta varia i pesi iniziali ad ogni esecuzione, quindi il numero di passi in cui converge e l'errore sono diversi ogni volta che viene addestrata la rete. I pesi finali ottenuti sono simili in ogni esecuzione, anche se non uguali.

Data la poca expleneability del modello, non possiamo trarre molte conclusioni interessanti dall'osservazione dei pesi finali. Sappiamo solo che i pesi positivi tendono ad attivare il neurone verso cui sono diretti, viceversa quelli negativi.

Per rendere più interpretabili questi risultati, pensiamo sia utile analizzare i pesi generalizzati. Questi infatti ci aiutano a capire l'importanza di un singolo attributo nella rete neurale, ovvero quali input spiegano meglio la classificazione dell'output. Viene riportato di seguito i grafici prodotti tramite l'analisi dei pesi generalizzati per i maschi.



Notiamo che vengono confermate le ipotesi ricavate dai density plot iniziali. Infatti, i grafici delle due covariate numeriche hanno pesi vicini allo zero, mostrando che le covariate hanno un effetto minimo. Anche la prima variabile booleana, *long_hair*, presenta un grafico di questo tipo, dal quale traiamo le medesime conclusioni. Per le rimanenti covariate booleane invece si nota come queste spieghino meglio la classificazione dell'output, avendo dei pesi maggiormente distribuiti sull'asse relativa al valore dei pesi generalizzati.

La funzione di attivazione della nostra rete neurale è la funzione logistica, il criterio di arresto per la rete neurale utilizzata è una devianza media rispetto ai pesi inferiore a una certa epsilon e la funzione di errore è la somma di quadrati.

In particolare, abbiamo deciso di utilizzare la funzione logistica in quanto particolarmente appropriata per variabili binarie come la nostra. Sarebbe più appropriato utilizzare la funzione sigmoide, ma dato che non è disponibile nel package utilizzato (neuralnet), abbiamo deciso di utilizzare la funzione logistica, che è comunque indicata per il dataset analizzato.

Ogni neurone in output ha come valore la probabilità che l'istanza considerata appartenga alla classe rappresentata da quel neurone. La colonna *prediction* sarà stabilita in base al valore più alto tra le due probabilità.

Con la rete neurale presentata in figura si ottiene un'accuracy di **0.968** e la matrice di confusione di seguito riportata.

Target / Prediction	**0 (maschio)**	**1 (femmina)**
0 (maschio)	721	32
1 (femmina)	16	731

È da notare però che ad ogni esecuzione del codice la rete neurale ottenuta avrà un'accuracy diversa, addirittura potrebbe anche non convergere, dato che inizializza i pesi iniziali in modo casuale.

3.4 NAIVE BAYES

Come ultimo modello di machine learning utilizziamo Naive Bayes. Questo modello è particolarmente adatto per il nostro dataset, in quanto la classe target è equamente divisa e le covariate sono quasi tutte categoriche. In particolare, decidiamo di rimuovere le due variate numeriche, che sappiamo dall'analisi esplorativa del dataset essere poco significative per stabilire se il soggetto è maschio o femmina.

Per poter applicare questo modello è stato necessario rendere le colonne fattoriali, sia del training set che del test set.

Come per la rete neurale si ha una bassa expleneability. Inoltre, non viene prodotto nessun artefatto di tipo grafico. Questo modello, infatti, si basa unicamente su calcoli di probabilità a priori e condizionate, che si ottengono come produttoria delle singole probabilità condizionate relative a ogni attributo. È possibile utilizzare questa produttoria poiché ipotizziamo che gli attributi siano indipendenti tra loro.

Otteniamo un'accuracy di **0.9586667** e la seguente matrice di confusione.

Target / Prediction	**0 (maschio)**	**1 (femmina)**
0 (maschio)	705	48
1 (femmina)	14	733

4 COMPARAZIONE DEI MODELLI

In questa fase conclusiva analizziamo la curva ROC e i tempi di training di ognuno dei tre modelli ottenuti e cerchiamo di scegliere il migliore.

Valutare un modello considerando solo la performance media non è significativo.

Si utilizza quindi la 10-fold cross validation, che consiste nel suddividere le istanze del dataset in 10 parti (chiamate appunto fold) e nell'eseguire la procedura di apprendimento utilizzando 9 di queste come training set e la rimanente come test set. Questa procedura viene ripetuta per dieci volte in tutto, in modo che ogni fold venga utilizzato come test set esattamente una volta.

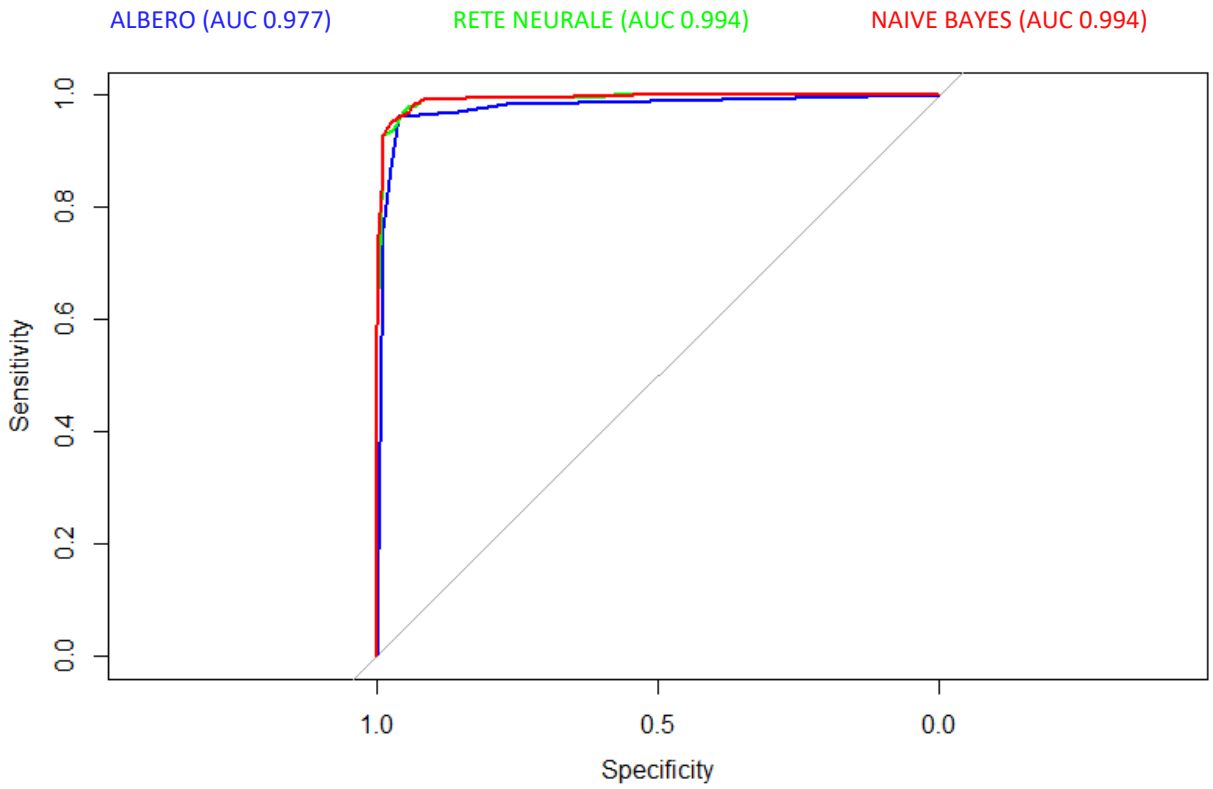
Si ottengono di fatto dieci modelli che possiamo confrontare per stabilire dei parametri di performance che forniscono una stima più realistica sulle performance del modello finale, rispetto alla classica divisione del dataset in due porzioni.

4.1 CURVE ROC

La curva ROC mostra come ogni modello si comporta al variare del cut off impostato. Il valore di soglia (o valore di cut off) per una determinata classe è un valore da superare per poter classificare un'istanza in quella classe.

Di default il valore di soglia di "male" è 50%, il che significa che un'istanza viene messa a "male" se la sua percentuale è >50%, con soglia 0.5.

Modificando questo valore si sbilancia la previsione in uno o nell'altro lato. Infatti, se per esempio si imposta il valore di cut off a 0.6, la colonna prediction viene messa a "male" solamente se la sua percentuale è >60%. Al crescere del cut off aumenta quindi il numero di istanze con attributo "female". Con un cut off di 1 si hanno tutte istanze "female", mentre con un cut off di 0 si hanno tutte "male".



Ascisse e ordinate sono i valori di sensitivity e specificity:

- Sensitivity: la sensitività indica il tasso di positivi reali (prima riga della confusion matrix)
$$\frac{TP}{TP + FN}$$
- Specificity: la specificità indica il tasso di negativi reali (seconda riga della confusion matrix)
$$\frac{TN}{TN + FP}$$

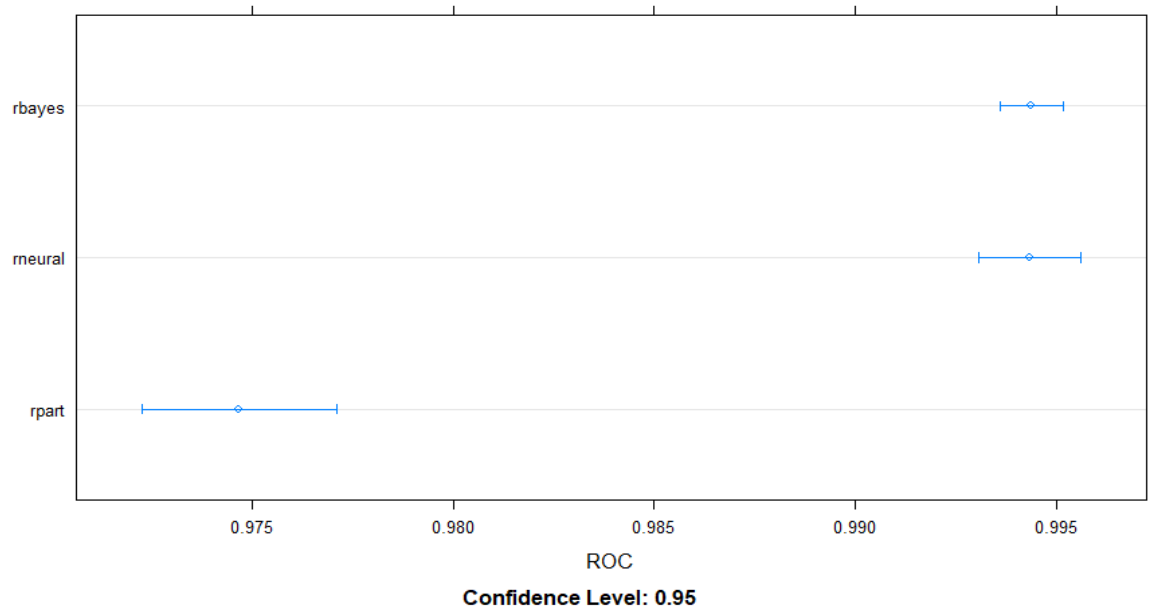
Per disegnare la curva di ogni modello, è stato istruito il modello ed è stato fatto variare il valore di cut off da 0 a 1, calcolando ogni volta sensitivity e specificity e ottenendo una curva.

La curva ROC rende possibile osservare, al fine di stabilire il modello migliore:

- Il punto dove sia sensitivity che specificity si avvicinano il più possibile al valore 1, che rappresenta il punto migliore, ovvero un modello che ha un valore massimo nel riconoscere le istanze positive e negative
- Area totale sottostante alla curva (valore AUC – Area Under Curve)

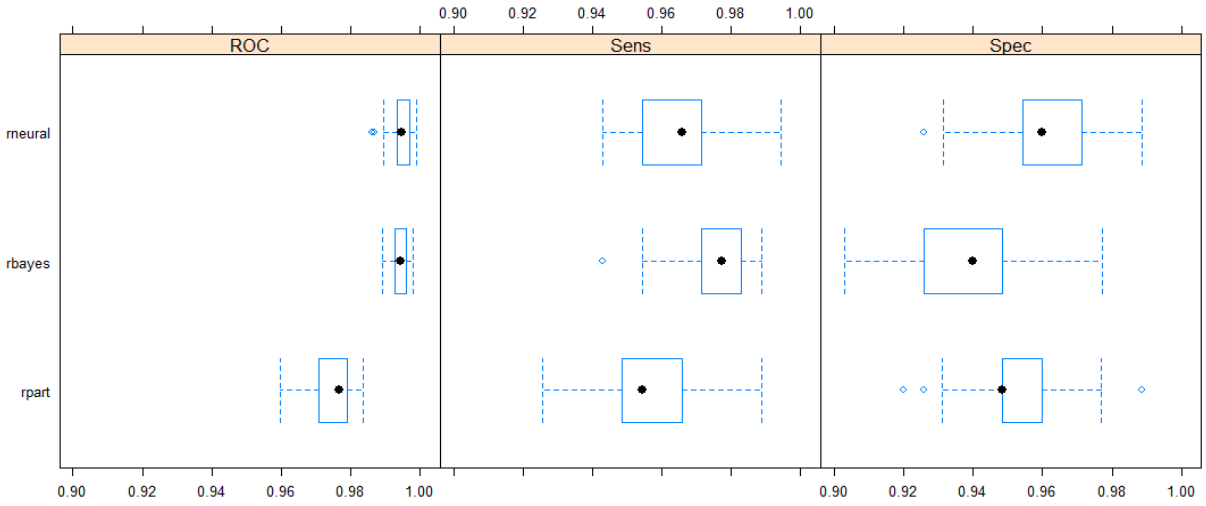
Notiamo, osservando questi due indicatori, che la curva ROC dell'albero ha prestazioni peggiori rispetto agli altri due modelli, i quali invece hanno un andamento simile e si sovrappongono in molti tratti, rendendo difficile scegliere uno dei due modelli.

Per scegliere il modello migliore tra la rete neurale e Naive Bayes, decidiamo quindi di guardare gli intervalli di confidenza dell'AUC (con livello di confidenza al 95%).



Questo grafico ci fa nuovamente escludere l'albero, mentre la sovrapposizione totale nel naive bayes rispetto alla rete neurale non ci fornisce informazioni utili. Sarebbe preferibile scegliere modelli con poca varianza, ma anche la differenza tra le varianze è minima.

Anche analizzando il grafico successivo, che mostra anche il confronto tra valori di sensitivity e specificity dei tre modelli, non permette di fornire informazioni sufficienti per preferire in maniera netta un modello all'altro.



4.2 MISURE DI PERFORMANCE

Analizziamo ora i seguenti indici per ogni modello, calcolati con la 10-fold cross validation, per ottenere ulteriori informazioni e fare un confronto più approfondito:

- Accuracy: indica quanto il modello è preciso nel classificare correttamente le istanze
$$\frac{TP + TN}{TP + TN + FP + FN}$$
- Precision: indica quanto il modello è preciso nel classificare positivamente le istanze
$$\frac{TP}{TP + FP}$$
- Recall: indica quanto il modello è preciso nel riconoscere le istanze positive
$$\frac{TP}{TP + FN}$$
- F-measure: combina precision e recall per avere un parametro che indica la performance del modello, facendo una combinazione pesata tra i due parametri precedenti
$$\frac{2 * Precision * Recall}{Precision + Recall}$$

Modello / Parametro	ALBERO	RETE NEURALE	NAIVE BAYES
Accuracy	0.961	0.959	0.959
Precision	0.961	0.959	0.981
Recall	0.961	0.959	0.936
F-measure	0.961	0.959	0.958

Analizzando i risultati per ogni singolo modello notiamo che:

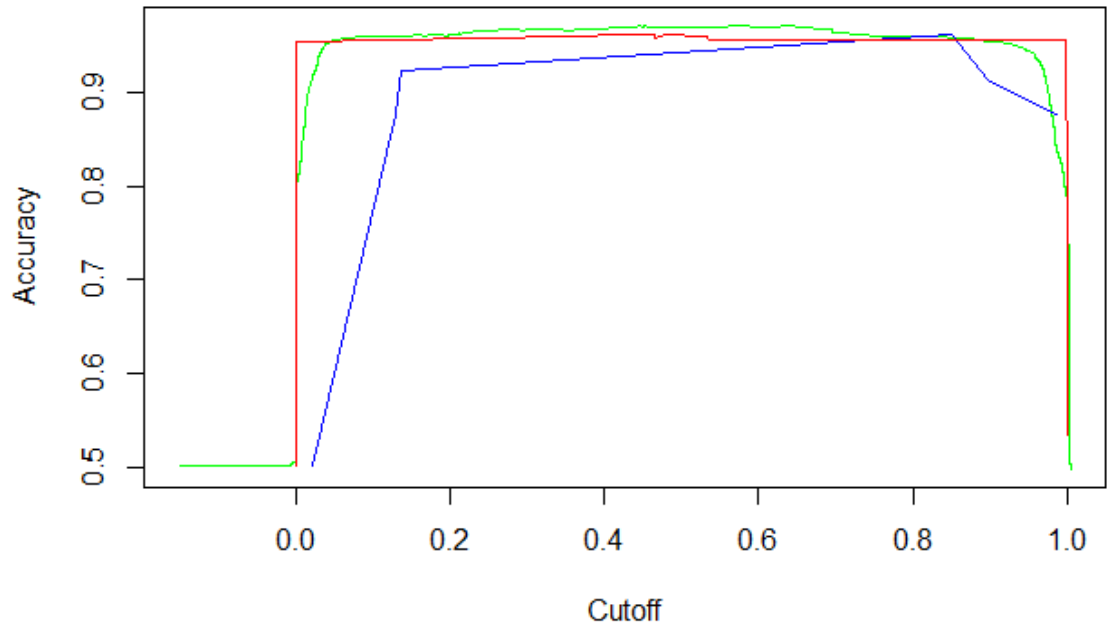
- Albero e rete neurale: sono bilanciati nel classificare istanze positive e negative, quindi si ottengono tutti gli indici con lo stesso valore
- Bayes: ha la tendenza a classificare le istanze come negative, infatti si ottiene un valore di precision superiore, ottenendo "di conseguenza" un valore di recall inferiore

Nonostante le considerazioni sui singoli modelli, notiamo che tutti i modelli hanno i valori di tutti gli indici molto alti.

4.3 MAXIMUM ACCURACY

Analizziamo ora l'andamento dell'accuracy al variare del cut off e riportiamo i valori di cut off per cui l'accuracy risulta massima:

Modello / Parametro	ALBERO	RETE NEURALE	NAIVE BAYES
Accuracy	0.961	0.971	0.961
Cutoff	0.851	0.572	0.476



Confrontando i soli valori di accuracy più elevata potremmo dire che l'albero e Bayes ottengono lo stesso risultato, mentre la rete neurale ha un risultato superiore. Sappiamo però che questa informazione non è sufficiente per trarre conclusioni, ma bisogna confrontare l'andamento dell'intera curva.

L'albero ha un accuracy che aumenta all'aumentare del cut off. La curva dell'albero è quasi sempre sottostante a quelle di rete neurale e Bayes, quindi, nonostante il valore di accuracy più elevata sia uguale a quello di Bayes, per tutti gli altri valori di cut off Bayes classifica meglio dell'albero.

Anche analizzando l'andamento delle curve abbiamo un ulteriore conferma che la rete neurale e Bayes sono molto simili, infatti le due curve hanno un andamento quasi sovrapponibile. Inoltre, notiamo nuovamente che hanno valori di accuracy molto elevati per ogni valore di cut off (ovviamente escludendo gli estremi), questo significa che non è necessario che il valore di cut off sia "preciso" per poter classificare correttamente le istanze.

4.4 TEMPI DI TRAINING

L'ultima caratteristica che analizziamo per stabilire il modello migliore sono i tempi di training dei singoli modelli.

Otteniamo così i seguenti risultati:

Modello / Parametro	ALBERO	RETE NEURALE	NAIVE BAYES
Everything	2.23	128.02	2.90
Final model	0.03	0.20	0.03

- Everything: tempo per fare 10 training e 10 test
- FinalModel: tempo per un solo train

Viene immediato notare che la rete neurale ha tempi di training di gran lunga superiori a quelli di naive Bayes, quindi, avendo prestazioni simili, scegliamo il modello che ha tempo di training inferiore.

CONCLUSIONI

Nella fase iniziale, in cui abbiamo analizzato le covariate del dataset, abbiamo ottenuto informazioni rilevanti riguardo la capacità delle covariate di spiegare le istanze.

I dati ottenuti in questa fase ci sono stati utili per trovare un criterio di classificazione delle istanze che, per quanto basilare, ha comunque fornito un buon risultato nella suddivisione del dataset.

I modelli di machine learning poi utilizzati sono stati:

1. ALBERO DI DECISIONE
   Essendo un modello semplice da interpretare, abbiamo potuto confermare ciò che era emerso dalle analisi precedenti, ovvero quali attributi spiegano bene il dataset. Nonostante i buoni risultati ottenuti, abbiamo notato che la rete neurale e Naive Bayes classificano meglio le istanze.
2. RETE NEURALE
   Nonostante abbia tempi di apprendimento molto elevati, classifica molto bene le istanze. Inoltre, grazie ai generalized weights abbiamo dato un'interpretazione ai risultati della rete, che hanno ulteriormente confermato le osservazioni derivate dalle prime analisi.
3. NAIVE BAYES
   Ottiene ottimi risultati nel classificare le istanze in tempo breve, ma non fornisce alcuna spiegazione su come avvenga la classificazione, rendendo difficile interpretare i risultati.

Modello / Caratteristica	ALBERO	RETE NEURALE	NAIVE BAYES
Classificazione	Media	Buona	Buona
Interpretabilità	Buona	Media	Scarsa
Velocità	Buona	Scarsa	Buona

Tutti i modelli scelti hanno vantaggi e svantaggi, ma classificano tutti molto bene le istanze.

L'albero è sicuramente il migliore a livello di interpretabilità, come già sapevamo. Naive Bayes classifica molto bene in tempi brevi. La rete neurale fornisce una via di mezzo tra i due modelli, a discapito della velocità di addestramento.

Come già si sapeva quindi è spesso difficile scegliere un modello nettamente migliore rispetto agli altri, dipende principalmente dalle caratteristiche che si ricercano.

Nel nostro caso abbiamo comunque un dataset con alcune covariate che aiutano molto la classificazione, rendendo qualunque modello una buona scelta.