Università degli Studi di Milano Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea Triennale in Informatica

Progetto Machine Learning

Contraceptive Method Choice

Autori:

Giacomo Savazzi – 845372

Raffaele Cerizza – 845512

Anno Accademico 2021 – 2022

Sommario

Indice delle Figure

Indice delle Tabelle

Descrizione del Dominio di riferimento e Obbiettivi dell'elaborato

Il dataset analizzato in questo elaborato, ***Contraceptive Method Choice*** (***cmc***), è tratto dai dati raccolti dal National Indonesia Contraceptive Prevalence Survey del 1987. In particolare questi dati sono stati raccolti al fine di eseguire una indagine sulla diffusione dei contraccettivi. I campioni sono donne sposate che non erano incinta, oppure non sapevano di esserlo, al momento dell'intervista. Il dataset di partenza è disponibile presso il sito ***UCI***.

Il problema che cerchiamo di risolvere applicando due diversi modelli predittivi è quello di prevedere se attualmente viene utilizzato oppure no un metodo contraccettivo, che sia a lungo o breve termine, in base alle caratteristiche demografiche e socioeconomiche della famiglia di cui la donna fa parte.

Scelte di Design per la creazione del Dataset

Per quanto riguarda l'organizzazione del progetto, si è deciso di dividere il codice in più scripts in modo tale da implementare il concetto di ***Separation of Concerns***: ogni script ha una sua responsabilità (caricare il dataset, eseguire l'analisi esplorativa, istallare i packages necessari etc.); agendo in questo modo si evita la duplicazione di codice, e risulta più semplice identificare la sezione da modificare. Inoltre, è anche possibile sfruttare una stessa sezione in progetti diversi.

In particolare, gli scripts sviluppati sono i seguenti:

- ***0_PackageInstaller.R***: questo script si occupa dell'istallazione di tutti i package necessari per quanto riguarda il progetto. Per ogni package, prima di istallarlo, viene controllato se già presente nell'ambiente corrente;
- **1*_LoadDataset.R***: script che si occupa del caricamento del dataset nell'ambiente R corrente, e del preprocessing dei dati. In particolare, tutte le colonne categoriche vengono trasformate da numeriche a fattoriali, in modo da poter svolgere successivamente un'analisi più precisa. Questo script include lo script ***0_PackageInstaller.R***, per l'istallazione di tutti i package necessari;
- ***2_1_ExplorationAnalysis.R***: script in cui vengono prodotte tutte le metriche e i grafici necessari per quanto riguarda l'analisi esplorativa univariata e multivariata del dataset. I dati ottenuti vengono analizzati dettagliatamente nella sezione successiva. Questo script include al suo interno ***1_LoadDataset.R***, per l'istallazione delle librerie necessarie e il caricamento del dataset;
- ***2_2_Pca.R***: script in cui viene eseguita la Principal Component Analysis sugli attributi numerici del dataset, con lo scopo di andare a semplificarlo. I risultati ottenuti e le modifiche al dataset che ne sono scaturite vengono spiegate nella sezione successiva. Questo script include al suo interno ***1_LoadDataset.R***, per l'istallazione delle librerie necessarie e il caricamento del dataset;

- ***3_1_1_DecisionTree_NoPca.R***: script in cui vengono creati, addestrati e testati alcuni modelli di Machine Learning noti come alberi di decisione a partire dal dataset originale. Inoltre questo script si occupa di calcolare diverse metriche utili per misurare la qualità dei modelli predittivi addestrati. Questo script utilizza i dati restituiti dallo script ***2_2_Pca.R***;
- ***3_1_2_DecisionTree_Pca.R***: script in cui vengono creati, addestrati e testati alcuni modelli di Machine Learning noti come alberi di decisione a partire dal dataset modificato a seguito della PCA. Inoltre questo script si occupa di calcolare diverse metriche utili per misurare la qualità dei modelli predittivi addestrati. Questo script utilizza i dati restituiti dallo script ***2_2_Pca.R***;
- ***3_2_1_NeuralNet_NoPca.R***: script in cui vengono creati, addestrati e testati alcuni modelli di Machine Learning noti come reti neurali a partire dal dataset originale. Inoltre questo script si occupa di calcolare diverse metriche utili per misurare la qualità dei modelli predittivi addestrati. Anche questo script utilizza i dati restituiti dallo script ***2_2_Pca.R***;
- ***3_2_2_NeuralNet_Pca.R***: script in cui vengono creati, addestrati e testati alcuni modelli di Machine Learning noti come reti neurali a partire dal dataset modificato a seguito della PCA. Inoltre questo script si occupa di calcolare diverse metriche utili per misurare la qualità dei modelli predittivi addestrati. Anche questo script utilizza i dati restituiti dallo script ***2_2_Pca.R***;
- ***4_1_ModelComparison_NoPca.R***: script in cui vengono messi a confronto i modelli addestrati tramite alberi decisionali con quelli addestrati tramite reti neurali, entrambi a partire dal dataset originale, per capire quale modello è il migliore nel rappresentare il dataset. Questo script utilizza i dati restituiti dagli scripts ***3_2_1_NeuralNet_NoPca.R*** e ***3_1_1_DecisionTree_NoPca.R***;
- ***4_2_ModelComparison_Pca.R***: script in cui vengono messi a confronto i modelli addestrati tramite alberi decisionali con quelli addestrati tramite reti neurali, entrambi a partire dal dataset modificato a seguito della PCA, per capire quale modello è il migliore nel rappresentare il dataset. Questo script utilizza i dati restituiti dagli scripts ***3_2_2_NeuralNet_Pca.R*** e ***3_1_2_DecisionTree_Pca.R***;
- ***5_Utilities.R***: script che contiene le definizioni di alcune funzioni utili in modo da permetterne il riutilizzo in altri script;

Anche se l'istallazione dei package necessari avviene in automatico, è necessario specificare esplicitamente all'interno di ogni script quali package si vuole usare, importandoli tramite comando ***library***. Questo perché non è detto che tutti gli scripts debbano fare uso di tutti i package istallati.

Descrizione dei dati (Analisi Esplorativa e PCA)

Il dataset, nella sua versione raw, si compone di dieci attributi:

- ***Wife's Age***: attributo di tipologia intera che rappresenta l'età della moglie;
- ***Wife's Education***: attributo di tipologia intera che rappresenta il livello di educazione della moglie. Questo attributo può assumere valore tra 1 (basso livello di educazione) e 4 (alto livello di educazione);
- ***Husband's Education***: attributo di tipologia intera che rappresenta il livello di educazione del marito. Questo attributo può assumere valore tra 1 (basso livello di educazione) e 4 (alto livello di educazione);
- ***Number of Children even born***: attributo di tipologia intera che rappresenta il numero di figli già nati nella famiglia considerata;
- ***Wife's Religion***: attributo di tipologia intera che rappresenta il tipo di religione della moglie. Questo attributo può assumere valore 1 (moglie di religione islamica) oppure 0 (moglie non islamica);
- ***Wife's working now***: attributo di tipologia intera che indica se la moglie lavora oppure no al momento. Questo attributo può assumere valore 0 (la moglie lavora al momento) oppure 1 (la moglie non lavora al momento);
- ***Husband's Occupation***: attributo di tipologia intera che rappresenta il livello di importanza del lavoro del marito. Questo attributo può assumere valore da 1 (lavoro di basso livello) a 4 (lavoro di alto livello);
- ***Standard-of-Living Index***: attributo di tipologia intera che rappresenta il livello generale della qualità di vita della famiglia. Questo attributo può assumere valore tra 1 (livello basso) e 4 (livello alto);
- ***Media Exposure***: attributo di tipologia intera che rappresenta il livello di esposizione mediatica della famiglia. Questo attributo può assumere valore 0 (buona esposizione) oppure 1 (cattiva esposizione);
- ***Contraceptive method used***: attributo di tipologia intera che rappresenta il tipo di contraccettivo usato. Questo attributo può assumere valore 1 (contraccettivo non usato), 2 (usato contraccettivo a lungo termine) oppure 3 (usato contraccettivo a breve termine);

In ***Figura 1*** possiamo vedere la tipologia dei diversi attributi subito dopo aver caricato il dataset nell'ambiente R, senza aver ancora eseguito su di esso alcuna operazione:

```
> sapply(cmc, class)
          Wife_Age      Wife_Education     Husband_Education       Number_Children
         "integer"           "integer"             "integer"             "integer"
     Wife_Religion     Wife_Is_Working   Husband_Occupation          Living_Index
         "integer"           "integer"             "integer"             "integer"
   Media_Exposure Contraceptive_Is_Used
         "integer"           "integer"
```

Figura 1 - Tipologia degli attributi raw

Prima cosa che si è deciso di fare è stato un ***Refactoring*** del dataset, con lo scopo principale di trasformare le variabili attualmente intere, ma che in realtà assumono valore all'interno di un range molto ristretto, in variabili categoriche. Questa operazione è stata eseguita per le seguenti variabili:

- ***Wife's Education***: la variabile è stata modificata dal tipo intero al tipo categorico con quattro livelli. L'associazione fatta è del seguente tipo:
  - 1 -> Low;
  - 2 -> Mid-Low;
  - 3 -> Mid-High;
  - 4 -> High;

  Stessa cosa è stata fatta anche per le variabili ***Husband's Education***, ***Husband's Occupation*** e ***Standard-of-Living Index***;
- ***Wife's Religion***: la variabile è stata modificata dal tipo intero al tipo categorico con due livelli. L'associazione fatta è la seguente:
  - 0 -> Non-Islam;
  - 1 -> Islam;
- ***Wife is Working***: la variabile è stata modificata dal tipo intero al tipo categorico con due livelli. L'associazione fatta è la seguente:
  - 0 -> Yes;
  - 1 -> No;
- ***Media Exposure***: la variabile è stata modificata dal tipo intero al tipo categorico con due livelli. L'associazione fatta è la seguente:
  - 0 -> Good;
  - 1 -> Not-Good;
- ***Contraceptive is used***: la variabile è stata modificata dal tipo intero al categorico con due livelli. L'associazione fatta è la seguente:
  - 1 -> No;
  - 2 o 3 -> Yes;

  Si è deciso di lavorare su due livelli (utilizzo o meno del contraccettivo) per semplificare l'analisi e i modelli. Questa decisione verrà approfondita meglio nel capitolo dedicato ai modelli utilizzati;

La tipologia degli attributi dopo il refactoring viene mostrata in ***Figura 2***.

```
> sapply(cmc, class)
             Wife_Age        Wife_Education     Husband_Education       Number_Children
            "integer"              "factor"              "factor"             "integer"
       Wife_Religion       Wife_Is_Working    Husband_Occupation          Living_Index
            "factor"              "factor"              "factor"              "factor"
     Media_Exposure Contraceptive_Is_Used
            "factor"              "factor"
```

Figura 2 - Tipologia degli attributi dopo il refactoring

Analisi Esplorativa

Una volta eseguito il refactoring del dataset, si è passati all'analisi esplorativa. Prima di tutto si è osservato il dataset nell'insieme: tramite comando ***str*** si è potuto analizzare la numerosità e la tipologia delle diverse variabili, ottenendo il risultato mostrato in ***Figura 3***.

```
> str(cmc)
'data.frame':   1473 obs. of  10 variables:
 $ Wife_Age             : int  24 45 43 42 36 19 38 21 27 45 ...
 $ Wife_Education       : Factor w/ 4 levels "Low","Mid-Low",..: 2 1 2 3 3 4 2 3 2 1 ...
 $ Husband_Education    : Factor w/ 4 levels "Low","Mid-Low",..: 3 3 3 2 3 4 3 3 3 1 ...
 $ Number_Children      : int  3 10 7 9 8 0 6 1 3 8 ...
 $ Wife_Religion        : Factor w/ 2 levels "Non-Islam","Islam": 2 2 2 2 2 2 2 2 2 2 ...
 $ Wife_Is_Working      : Factor w/ 2 levels "Yes","No": 2 2 2 2 2 2 2 1 2 2 ...
 $ Husband_Occupation   : Factor w/ 4 levels "Low","Mid-Low",..: 2 3 3 3 3 3 3 3 3 2 ...
 $ Living_Index         : Factor w/ 4 levels "Low","Mid-Low",..: 3 4 4 3 2 3 2 2 4 2 ...
 $ Media_Exposure       : Factor w/ 2 levels "Good","Not-Good": 1 1 1 1 1 1 1 1 1 2 ...
 $ Contraceptive_Is_Used: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

Figura 3 - Risultato del comando str eseguito sul dataset cmc

Si nota che il dataset si compone di 1473 istanze, un numero non indifferente ma neanche troppo elevato. Questa informazione sarà utile per capire come eseguire successivamente l'addestramento e il test dei modelli predittivi, cioè decidere se dividere il dataset in train e test, oppure eseguire una cross-validation.

1. Wife's Age

Per quanto riguarda ***Wife's Age***, si è eseguito un cut del dataset in quattro categorie sulla base del valore assunto da questa variabile:

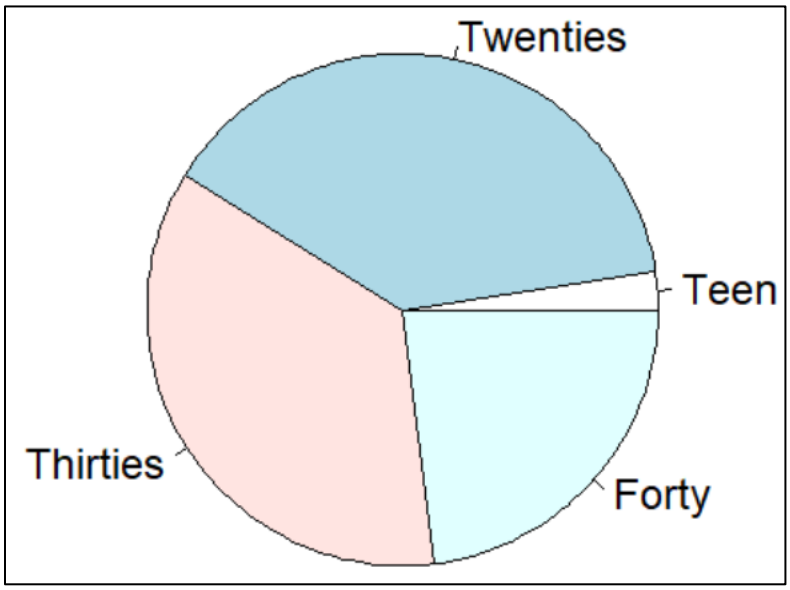
- ***Teen***: età minore o uguale a 19;
- ***Twenties***: età tra i 20 e i 29, estremi compresi;
- ***Thirty***: età tra i 30 e i 39, estremi compresi;
- ***Forty***: età tra i 40 e i 49, estremi compresi;

Tramite comando ***summary*** si è osservato che il valore minimo per l'età è 16, quello massimo 49, e la media è 32. Si osservi in ***Figura 4*** il risultato grezzo ottenuto.

```
> summary(cmc$Wife_Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  16.00   26.00   32.00   32.54   39.00   49.00 
```

Figura 4 - Risultato del comando summary su Wife's Age

In ***Figura 5*** possiamo vedere come si distribuiscono le istanze in base alla categoria di età.



Possiamo notare che il numero di istanze con età della moglie tra i 16 e 19 anni è molto basso, mentre, le altre tre categorie si distribuiscono abbastanza equamente.

Si potrebbe pensare, nelle fasi successive, di non considerare le istanze con età tra i 16 e 19 anni nell'addestramento dei modelli, perché potrebbero essere forvianti visto il loro basso impatto.

Figura 5 - Grafico a torta relativo alle categorie di età

Successivamente si è eseguito un boxplot relativo a Wife's Age, per vedere la distribuzione dei valori assunti dalle diverse istanze su questa variabile. In ***Figura 6*** vediamo il risultato ottenuto.

Figura 6 - Boxplot per Wife's Age

Da questo plot notiamo non essere presenti outliers, cioè valori eccessivamente distanti dalla media.

In ultimo, abbiamo sviluppato un feature plot per vedere come cambia il valore del target al variare del valore di Wife's Age. In ***Figura 7*** vediamo il risultato ottenuto.

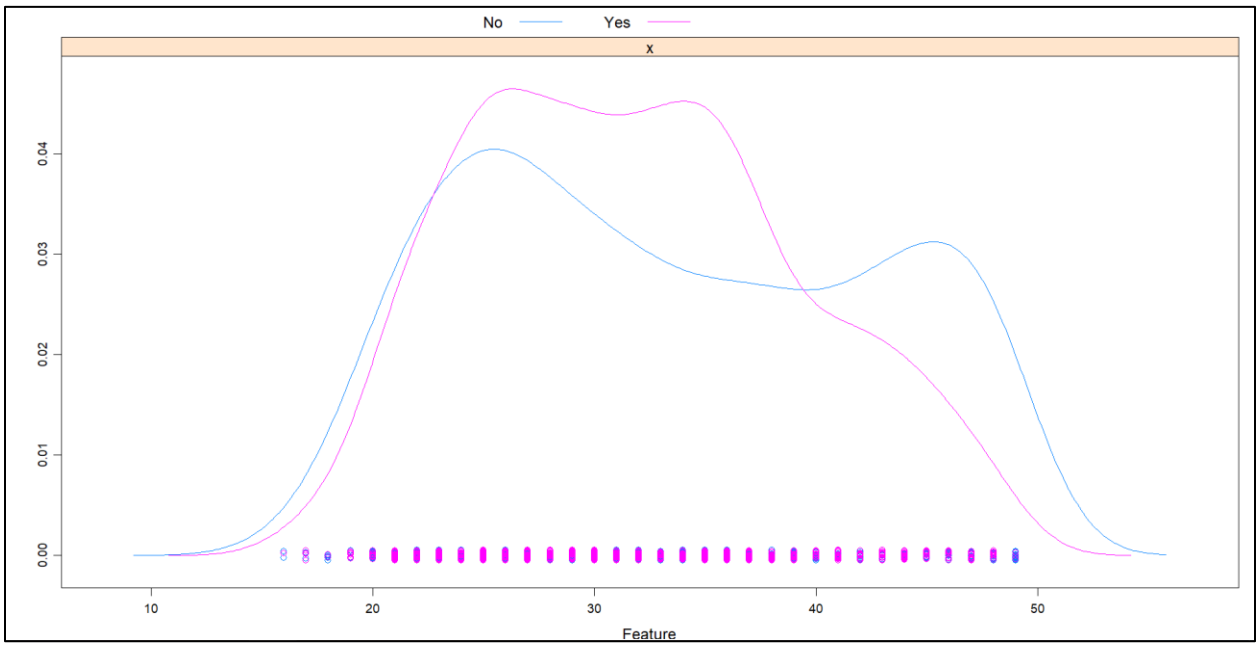


Figura 7 - Feature plot per Wife's Age in relazione al target

Da questo grafico notiamo due curve abbastanza sovrapposte. Questo vuol dire che è difficile capire se il contraccettivo viene usato oppure no solo sulla base di Wife's Age.

2. Number of Children

Tramite esecuzione del comando ***summary*** su Number of Children si è scoperto che il valore minimo assunto da questa variabile è 0, il massimo 16, e la media 3. In ***Figura 8*** possiamo vedere il risultato grezzo ottenuto dall'esecuzione del comando.

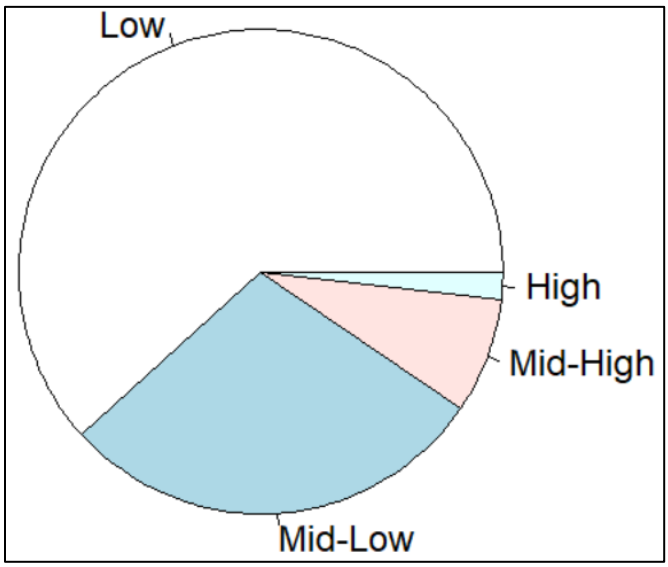
```
> summary(cmc$Number_Children)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   3.000   3.261   4.000  16.000
```

Figura 8 - Risultato del comando summary eseguito su Number of Children

Si è successivamente eseguito un cut del dataset sulla base del valore assunto dalla variabile Number of Children. Le quattro categorie considerate sono:

- ***Low***: valore tra 0 e 3, estremi compresi;
- ***Mid-Low***: valore tra 4 e 6, estremi compresi;
- ***Mid-High***: valore tra 7 e 9, estremi compresi;
- ***High***: valore tra 10 e 16, estremi compresi;

È stato poi costruito un grafico a torta per analizzare graficamente la numerosità delle istanze associate alle diverse categorie sviluppate. In ***Figura 9*** possiamo vedere il risultato ottenuto.



Come possiamo vedere, si ha che la maggior parte delle istanze presenta un numero di figli tra gli 0 e i 3, mentre, le istanze con un numero di figli tra 7 e 16 sono molto poche, e particolarmente poche sono le istanze con numeri di figli tra 9 e 16.

Figura 9 - Grafico a torta relativo a Number of Children

Eseguiamo poi un boxplot, per vedere la distribuzione dei valori. In ***Figura 10*** vediamo il risultato ottenuto. Si nota la presenza di outliers verso l'alto, cioè valori molto distanti dalla media. Questo risultato è consistente con quanto osservato dal grafico a torta, infatti sono presenti molte istanze con pochi figli, che abbassano la media, e poche istanze con tanti figli, estranee alla media.

Figura 10 - Boxplot per Number of Children

Anche per Number of Children è stato infine sviluppato un feature plot, per vedere come varia il valore della variabile target al variare del numero di figli. In ***Figura 11*** vediamo il risultato ottenuto.

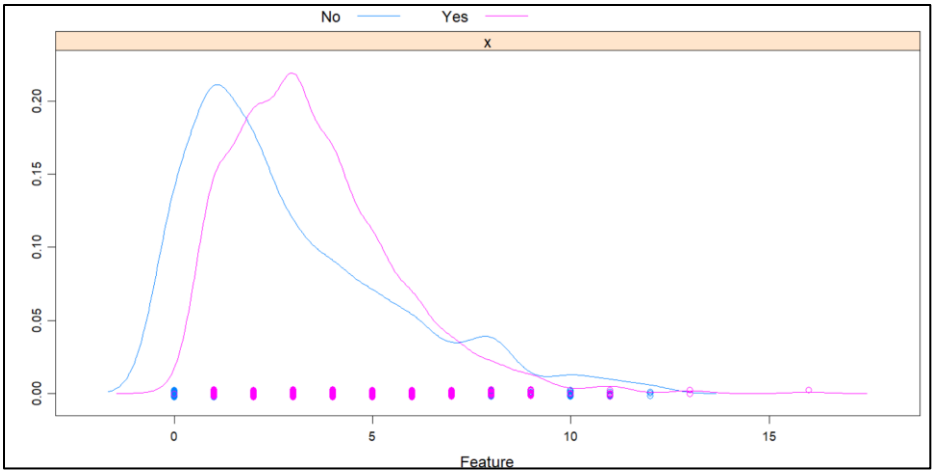


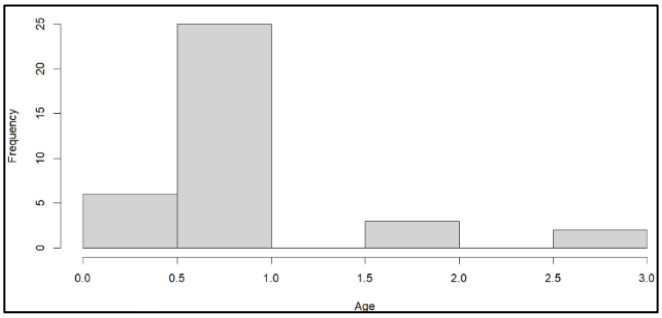
Figura 11 - Feature plot per Number of Children

Anche in questo caso, come osservato per Wife's Age, abbiamo una certa sovrapposizione tra le due curve, quindi non è possibile capire se il contraccettivo viene usato oppure no solo sulla base di Number of Children.

3. Number of Children e Wife's Age

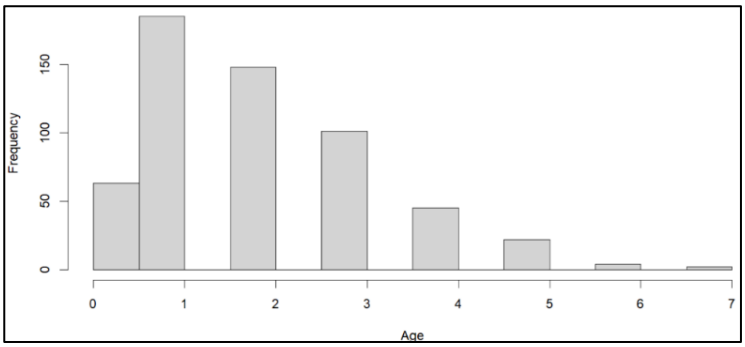
Concludendo l'analisi delle variabili numeriche, abbiamo analizzato il numero di figli sulla base dell'età della moglie, ottenendo i seguenti risultati:

- Per quanto riguarda le istanze con età tra i 16 e i 19, estremi compresi, abbiamo ottenuto il seguente risultato:



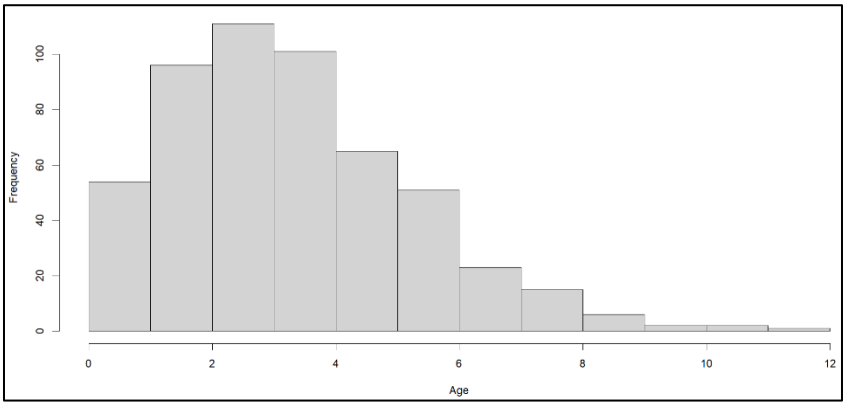
Si osserva che la maggior parte delle coppie con età della moglie tra i 16 e i 19 hanno un solo figlio;

- Per quanto riguarda le istanze con età tra i 20 e i 29, estremi compresi, abbiamo ottenuto il seguente risultato:



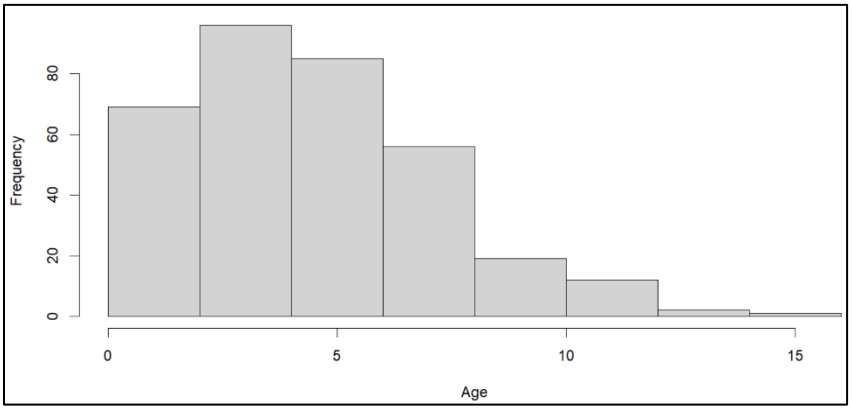
Si nota che ancora il numero di figli più frequente è pari ad uno, ma iniziano ad aumentare le coppie con più figli;

- Per quanto riguarda le istanze con età tra i 30 e i 39, estremi compresi, abbiamo ottenuto il seguente risultato:



Il numero di figli più presente è pari a 3, con un considerevole aumento delle coppie con più di un figlio;

- Per quanto riguarda le istanze con età tra i 40 e i 49, estremi compresi, abbiamo ottenuto il seguente risultato:



Si nota ancora un aumento del numero di figli per coppia;

In definitiva, si è osservata una correlazione tra le due dimensioni numeriche, visto che all'aumentare dell'età aumenta in genere anche il numero di figli. Tramite comando ***cor*** si è calcolata questa correlazione. In ***Figura 12*** vediamo il risultato ottenuto.

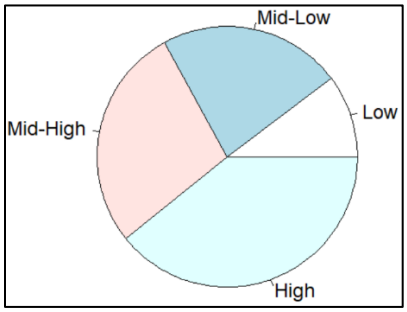
```
> cor(cmc[, c(1,4)]) # 0.5401259
                 Wife_Age Number_Children
Wife_Age        1.0000000       0.5401259
Number_Children 0.5401259       1.0000000
```

Figura 12 - Correlazione tra Wife's Age e Number of Children

Si nota una correlazione tra le due variabili più vicina all'uno anziché allo zero quindi, anche se di poco, possiamo dire che in generale all'aumentare di Wife's Age aumenta anche Number of Children.

4. Wife's Education e Husband's Education

Per quanto riguarda le variabili categoriche relative al livello di educazione di moglie e marito, si è prima di tutto analizzato visivamente i relativi grafici a torta, mostrati in ***Figura 13*** e ***Figura 14***.



Si nota dai grafici che la maggior parte degli individui, sia moglie che marito, hanno un alto livello di educazione. Vediamo con i prossimi grafici se in qualche modo questi valori sono legati tra loro.

Figura 13 - Grafico a torta relativo a Wife's Education

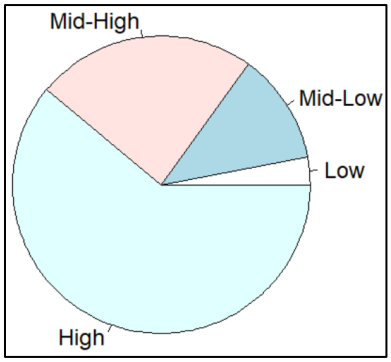
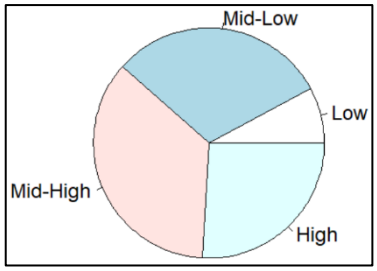


Figura 14 - Grafico a torta relativo a Husband's Education

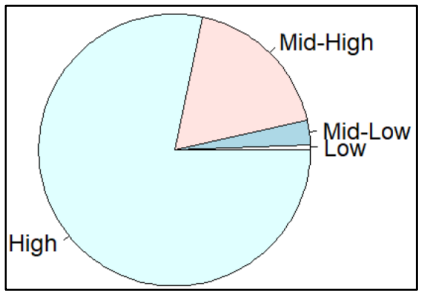
Consideriamo diversi casi:

- Vediamo il grafico a torta relativo alla distribuzione del livello di educazione dei mariti di donne con basso o medio basso livello di educazione:



Si nota che i mariti, a differenza delle mogli, presentano un livello di educazione variabile, con prevalenza di mariti con livello di educazione alto o medio alto;

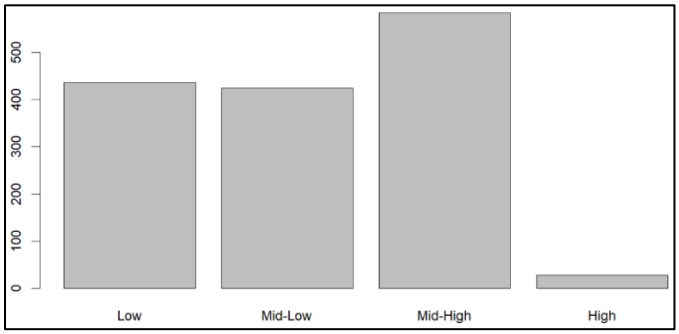
- Vediamo il grafico a torta relativo alla distribuzione del livello di educazione dei mariti di donne con alto o medio alto livello di educazione:



Si nota che le donne con alto o medio alto livello di educazione prediligono uomini con simile livello di educazione, a differenza di quanto visto per le donne con basso livello di educazione;

4. Husband's Occupation e Husband's Education

Analizziamo prima di tutto il grafico a barre relativo alla distribuzione del livello di occupazione del marito. Il risultato ottenuto è visibile in ***Figura 15***.

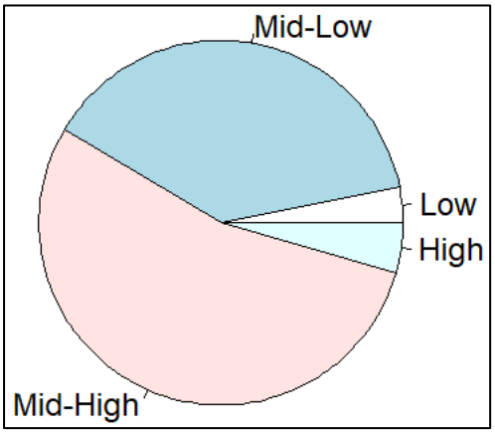


Dal risultato ottenuto si nota che la maggior parte dei mariti hanno un lavoro di fascia medio alta ma, la somma dei mariti con lavoro di fascia bassa o medio bassa è molto superiore.

Figura 15 - Bar plot relativo a Husband's Occupation

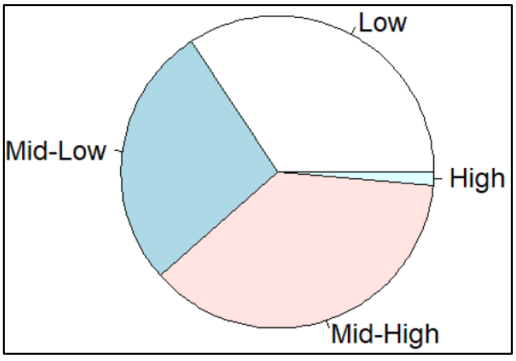
Vediamo la relazione tra il livello di educazione dei mariti e il relativo livello di occupazione:

- Consideriamo il grafico a torta relativo al livello di occupazione di mariti con basso o medio basso livello di educazione:



A differenza di quanto ci potessimo aspettare, la maggior parte dei mariti con basso o medio basso livello di educazione hanno una occupazione di medio alto livello;

- Consideriamo il grafico a torta relativo al livello di occupazione di mariti con alto o medio alto livello di educazione:

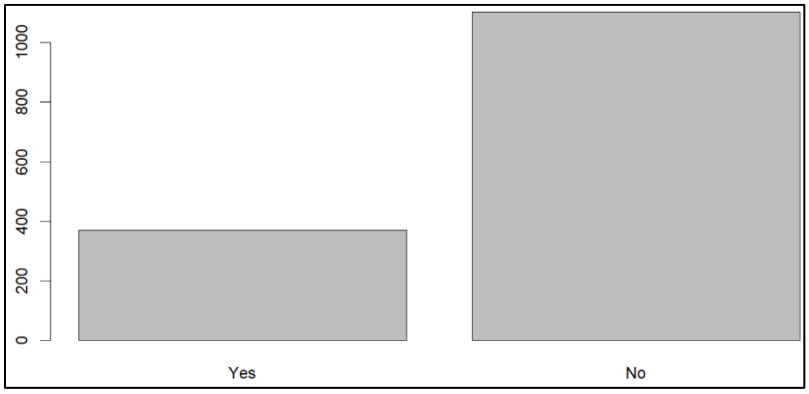


Anche in questo caso, il risultato ottenuto è diverso da quello atteso, infatti la maggior parte dei mariti con alto o medio alto livello di educazione hanno un lavoro di basso o medio basso livello;

In definitiva, si scopre che la relazione tra queste due variabili non è molto forte (quasi inversa).

5. Wife's Working, Wife's Education e Number of Children

Analizziamo il grafico a barre relativo alla distribuzione della variabile Wife's Working nel dataset. Il risultato ottenuto è visibile in ***Figura 16***.

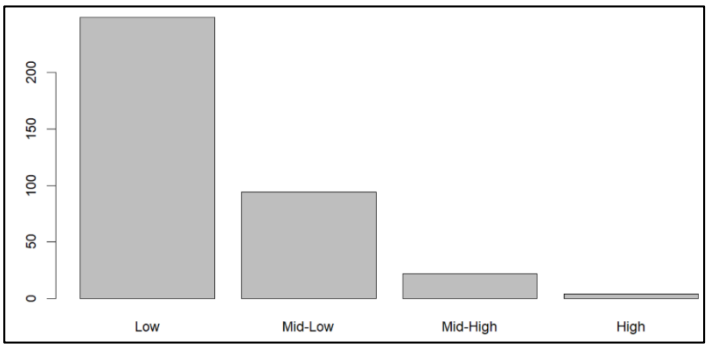


Si nota che la maggior parte delle mogli non lavora. Vediamo come questo dato si correla con il numero di figli nella famiglia e il livello di educazione della donna.

Figura 16 - Bar plot relativo a Wife's Working

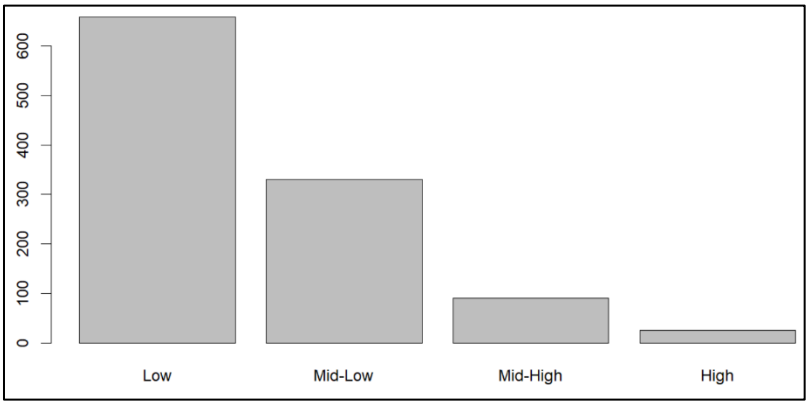
Consideriamo i seguenti grafici:

- Grafico a barre relativo alla distribuzione della variabile supplementare Children's Range per quanto riguarda le famiglie in cui la moglie non lavora. Il risultato ottenuto è il seguente:



Si noti che, come era sospettabile, il numero di figli prevalente è basso, cioè tra 0 e 3, estremi inclusi;

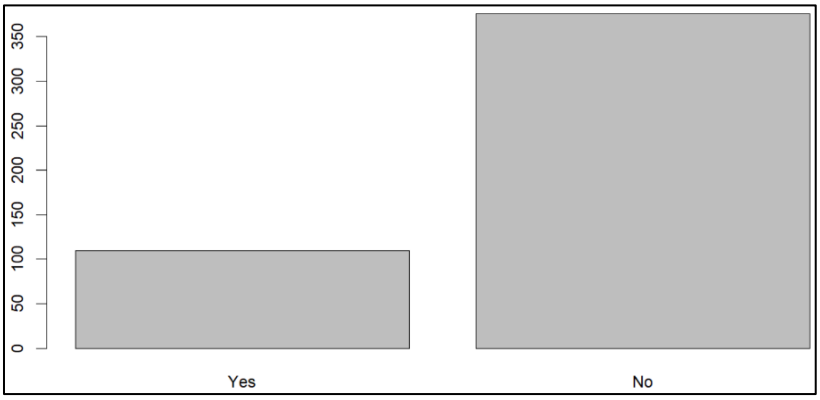
- Grafico a barre relativo alla distribuzione della variabile supplementare Children's Range per quanto riguarda le famiglie in cui la moglie lavora. Il risultato ottenuto è il seguente:



I valori sono leggermente migliorati, ma rimane prevalente la classe Low, che è anche la più numerosa nel dataset;

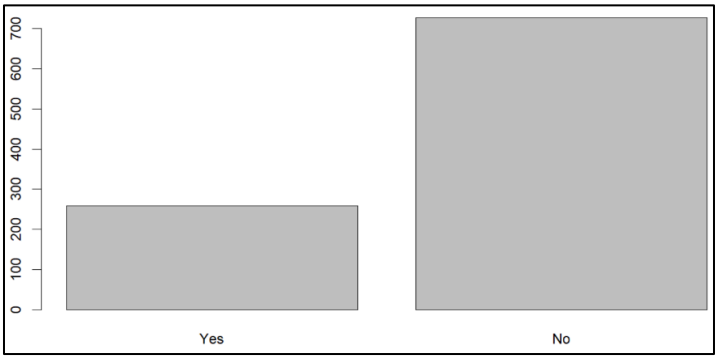
Vediamo ora le relazioni tra la variabile Wife's Working e Wife's Education, per capire se a lavorare sono donne per lo più con un certo livello di educazione:

- Consideriamo il grafico a barre relativo al livello di occupazione delle donne con basso o medio basso livello di educazione:



Si nota che la maggior parte di loro non lavora;

- Consideriamo il grafico a barre relativo al livello di occupazione delle donne con alto o medio alto livello di educazione:

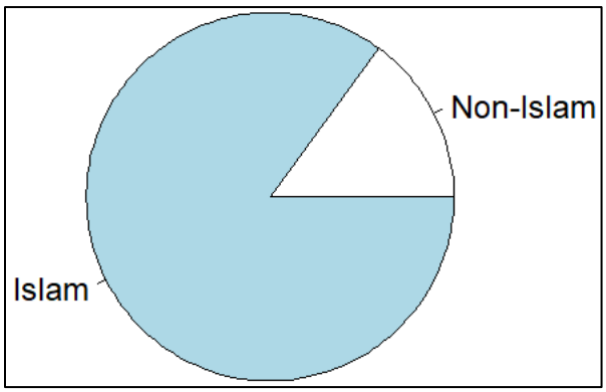


I livelli sono leggermente cambiati, infatti, in questo caso sono presenti più donne lavoratrici;

Giungiamo alla conclusione che nelle donne con alto o medio alto livello di educazione c'è più tendenza a lavorare.

6. Wife's Religion

Analizziamo prima di tutto il grafico a torta relativo alla distribuzione di Wife's Religion nel dataset. Il risultato ottenuto è visibile in ***Figura 17***.



Si nota che la maggior parte delle donne è di religione islamica. Vedremo poi la relazione tra questa variabile e quella target.

Figura 17 - Bar plot relativo a Wife's Religion

7. Media Exposure

Analizziamo il grafico a torta rappresentante la distribuzione della variabile Media Exposure nel dataset. Il risultato ottenuto è visibile in ***Figura 18***.

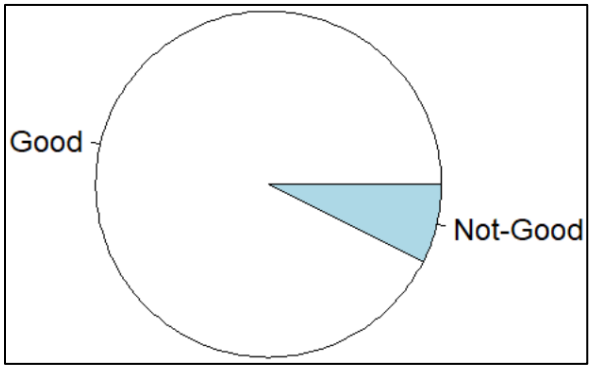


Figura 18 - Pie chart relativo a Media Exposure

Si nota che la maggior parte delle istanze hanno valore Good. Se non si osserveranno buone relazioni con la variabile target potremmo anche non considerare questa variabile.

8. Living Index

Analizziamo il grafico a barre relativo alla distribuzione della variabile Living Index nel dataset. Il risultato ottenuto è visibile in ***Figura 19***.

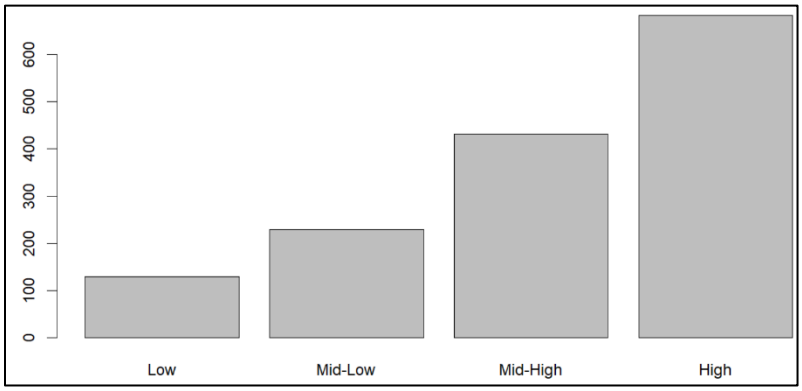


Figura 19 - Bar plot relativo a Living Index

Non si notano classi troppo prevalenti rispetto ad altre.

9. Utilizzo del Contraccettivo, e sue relazioni con altre variabili

Analizziamo prima di tutto il grafico a torta relativo alla variabile Contraceptive Is Used isolatamente. Il risultato ottenuto è visibile in ***Figura 20***.

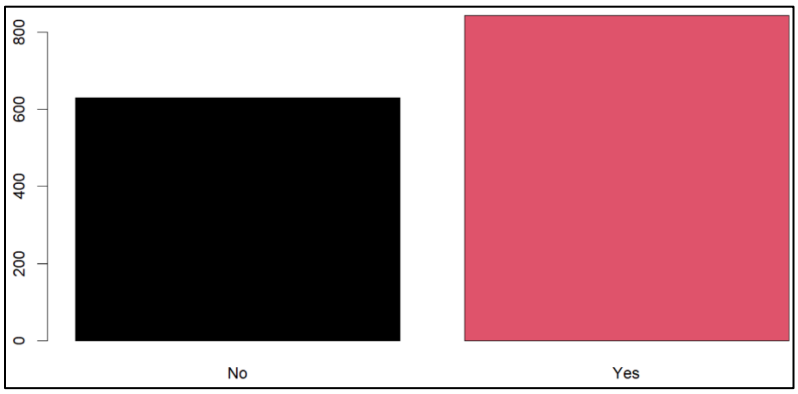
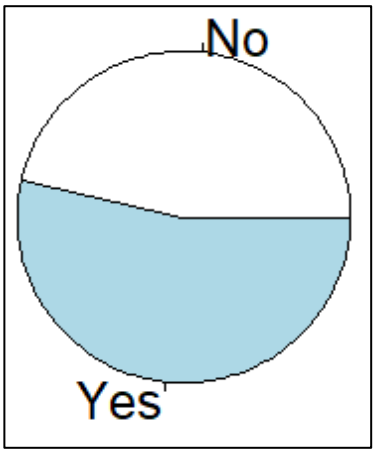


Figura 20 - Bar plot relativo a Contraceptive is Used

Si nota la prevalenza, seppur di poco, di famiglie che fanno uso del contraccettivo. Abbiamo, quindi, molte istanze per entrambe le categorie.

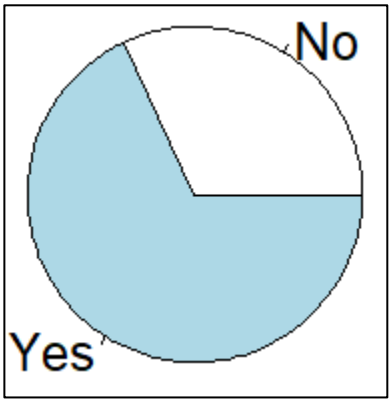
Consideriamo i seguenti grafici:

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di mogli con un numero di figli tra gli 0 e i 3 estremi compresi. Il risultato ottenuto è il seguente:



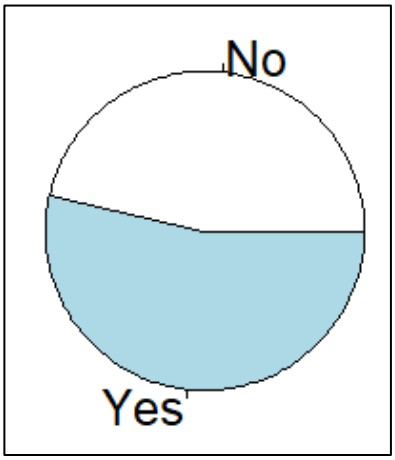
Si nota che nelle famiglie con pochi figli c'è la prevalenza nell'utilizzare il contraccettivo, anche se di poco;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di mogli con un numero di figli tra gli 4 e i 6, estremi compresi. Il risultato ottenuto è il seguente:



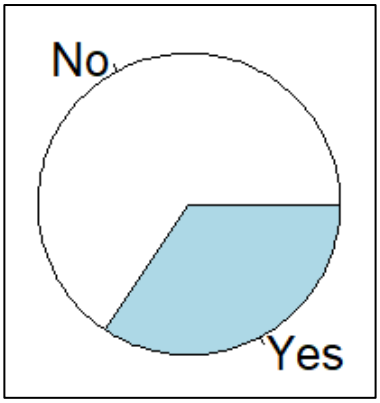
In questo caso c'è grande prevalenza nell'utilizzo del contraccettivo;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di mogli con un numero di figli tra gli 7 e i 9, estremi compresi. Il risultato ottenuto è il seguente:



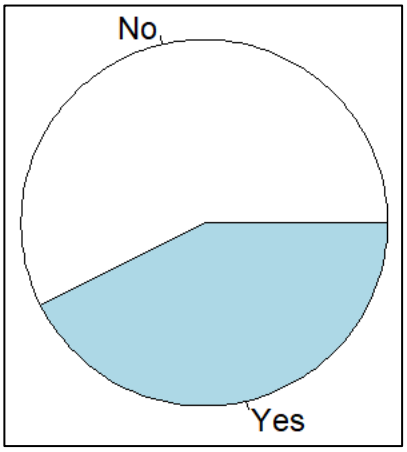
Le percentuali sono variate, adesso meno mogli utilizzano il contraccettivo;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di mogli con un numero di figli tra gli 10 e i 16, estremi compresi. Il risultato ottenuto è il seguente:



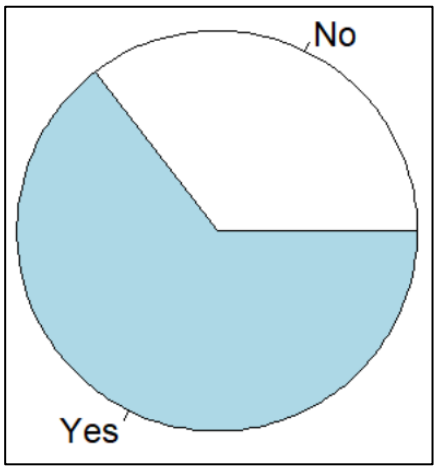
Come presumibile, le coppie con tanti figli fanno poco uso del contraccettivo.
Le conclusioni che possiamo trarre da questi dati sono che le coppie con pochi figli sono ben distribuite tra uso e non uso di un contraccettivo: questo può essere dovuto al fatto che certe coppie non vogliono figli, mentre altre ne sono alla ricerca. Si nota, invece, che le coppie con un numero medio basso di figli fanno tanto uso del contraccettivo: questo può essere dovuto al fatto che, avendo già un certo numero di figli, non ne vogliono altri. Infine, si nota che all'aumentare del numero di figli diminuisce l'uso del contraccettivo, forse perché quelle coppie vogliono proprio molti figli;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di mogli con basso o medio basso livello di educazione. Il risultato ottenuto è il seguente:



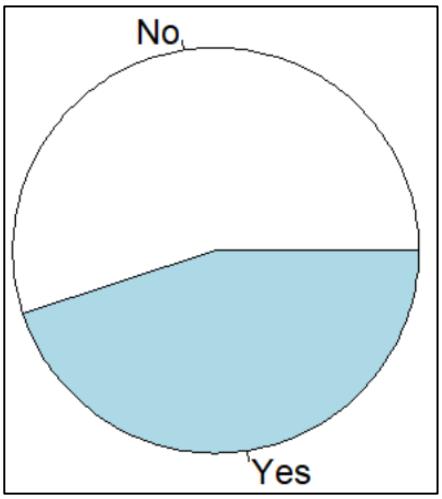
Si nota che le donne con basso o medio basso livello di educazione tendono a non utilizzare il contraccettivo;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di mogli con alto o medio alto livello di educazione. Il risultato ottenuto è il seguente:



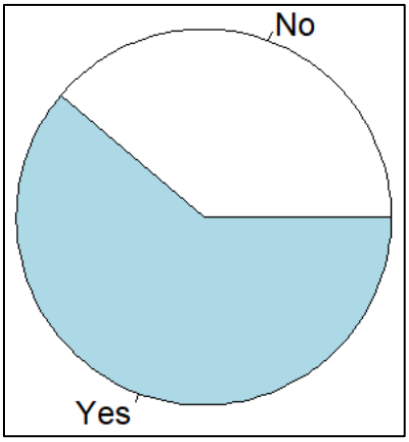
Si nota che le donne con alto o medio alto livello di educazione tendono molto di più ad utilizzare il contraccettivo. Questo risultato, unito al precedente, ci fa capire che il livello di educazione è molto legato all'utilizzo o meno del contraccettivo;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di coppie con basso o medio basso standard di vita. Il risultato ottenuto è il seguente:



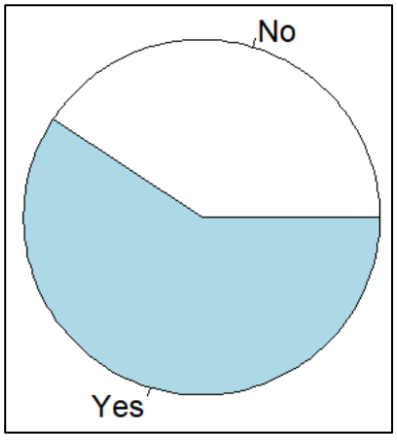
Si nota che le coppie con standard di vita basso o medio basso tendono a non utilizzare il contraccettivo;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di coppie con alto o medio alto standard di vita. Il risultato ottenuto è il seguente:



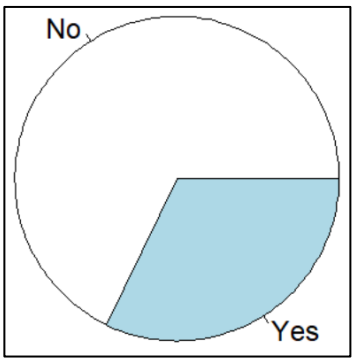
Si nota che le coppie con standard di vita alto o medio alto tendono a utilizzare il contraccettivo. Dal risultato precedente e il corrente capiamo che la variabile relativa allo standard di vita potrebbe essere utile nella predizione della variabile target;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di coppie con buona esposizione mediatica. Il risultato ottenuto è il seguente:



Si osserva che le coppie con buona esposizione mediatica tendono ad utilizzare il contraccettivo, ma con poco distacco;

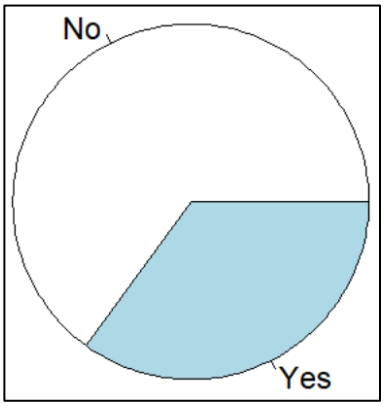
- Grafico a torta relativo all'utilizzo del contraccettivo da parte di coppie con cattiva esposizione mediatica. Il risultato ottenuto è il seguente:



Si osserva che gran parte delle coppie con cattiva esposizione mediatica non fa uso del contraccettivo. Dal risultato precedente e dal corrente capiamo che la variabile Media Exposure potrebbe essere utile nella predizione della variabile target, quindi si decide di tenerla;

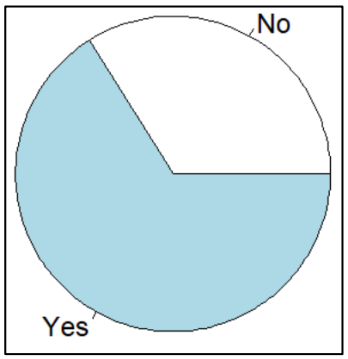
Avendo visto che le variabili Living Index e Wife's Education descrivono abbastanza bene la variabile target, vediamo cosa succede a considerarle entrambe per la predizione del contraccettivo utilizzato:

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di coppie con basso o medio basso livello di educazione della moglie, e basso o medio basso standard di vita. Il risultato ottenuto è il seguente:



Si nota che queste coppie tendono a non utilizzare il contraccettivo con abbastanza regolarità;

- Grafico a torta relativo all'utilizzo del contraccettivo da parte di coppie con alto o medio alto livello di educazione della moglie, e alto o medio alto standard di vita. Il risultato ottenuto è il seguente:



Si nota che queste coppie, a differenza delle precedenti, hanno molto di più la tendenza ad utilizzare il contraccettivo;

Dai risultati ottenuti, a maggior ragione queste due variabili verranno considerate nella definizione successiva dei modelli predittivi.

Principal Component Analysis (PCA)

Il processo di PCA è stato svolto con il principale obbiettivo di semplificare il dataset, in modo da cercare di ridurre il numero di attributi, mantenendo comunque una buona quantità di informazione.

Innanzitutto, sono stati calcolati due Principal Components facendo riferimento alle due uniche variabili numeriche presenti nel dataset: Wife's Age e Number of Children. Si è invece ritenuto opportuno non coinvolgere nella PCA le variabili categoriche. Infatti la PCA trasforma linearmente le variabili originali in un nuovo spazio (detto spazio delle componenti) nel quale le componenti sono ordinate in ordine decrescente di varianza e sono tra loro incorrelate. Poiché le variabili categoriche non sono numeriche, la determinazione della varianza tra queste variabili potrebbe dar luogo a errori nell'individuazione delle componenti. E questo suggerisce di non applicare la PCA alle variabili categoriche. In ogni caso il risultato della PCA svolta è visibile in ***Figura 21***.

```
> summary(cmc.pca)

Call:
PCA(X = cmc[, c(1, 4)], graph = FALSE)


Eigenvalues
                       Dim.1   Dim.2
Variance               1.540   0.460
% of var.             77.006  22.994
Cumulative % of var.  77.006 100.000

Individuals (the 10 first)
                  Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
1                |  1.044 | -0.812  0.029  0.606 | -0.656  0.063  0.394 |
2                |  3.235 |  3.092  0.422  0.914 | -0.950  0.133  0.086 |
3                |  2.033 |  2.021  0.180  0.988 | -0.222  0.007  0.012 |
4                |  2.692 |  2.535  0.283  0.886 | -0.908  0.122  0.114 |
5                |  2.053 |  1.719  0.130  0.701 | -1.124  0.186  0.299 |
6                |  2.150 | -2.142  0.202  0.993 | -0.186  0.005  0.007 |
7                |  1.338 |  1.291  0.073  0.931 | -0.352  0.018  0.069 |
8                |  1.699 | -1.670  0.123  0.966 | -0.314  0.015  0.034 |
9                |  0.682 | -0.555  0.014  0.660 | -0.398  0.023  0.340 |
10               |  2.517 |  2.493  0.274  0.981 | -0.350  0.018  0.019 |

Variables
                  Dim.1    ctr   cos2    Dim.2    ctr   cos2
Wife_Age        |  0.878 50.000  0.770 |  0.480 50.000  0.230 |
Number_Children |  0.878 50.000  0.770 | -0.480 50.000  0.230 |
```

Si notino gli autovalori ottenuti per le due dimensioni calcolate: si ha che l'autovalore della prima dimensione è molto maggiore di uno, a differenza dell'autovalore della seconda dimensione. Infatti, si nota che la sola prima dimensione descrive il 77% della varianza.

Si potrebbe pensare di non considerare la seconda dimensione, ma solo la prima, e andare a sostituirla nel dataset alle variabili Wife's Age e Number of Children.

Figura 21 - Summary della PCA eseguita sul dataset

Analizziamo graficamente gli autovalori tramite esecuzione del comando ***fviz_eig***. Il risultato ottenuto è visibile in ***Figura 22***.

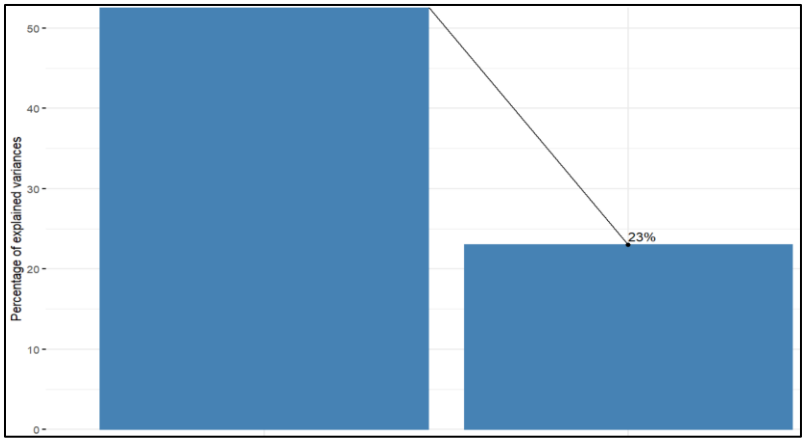


Figura 22 - Plot degli autovalori per le due PC

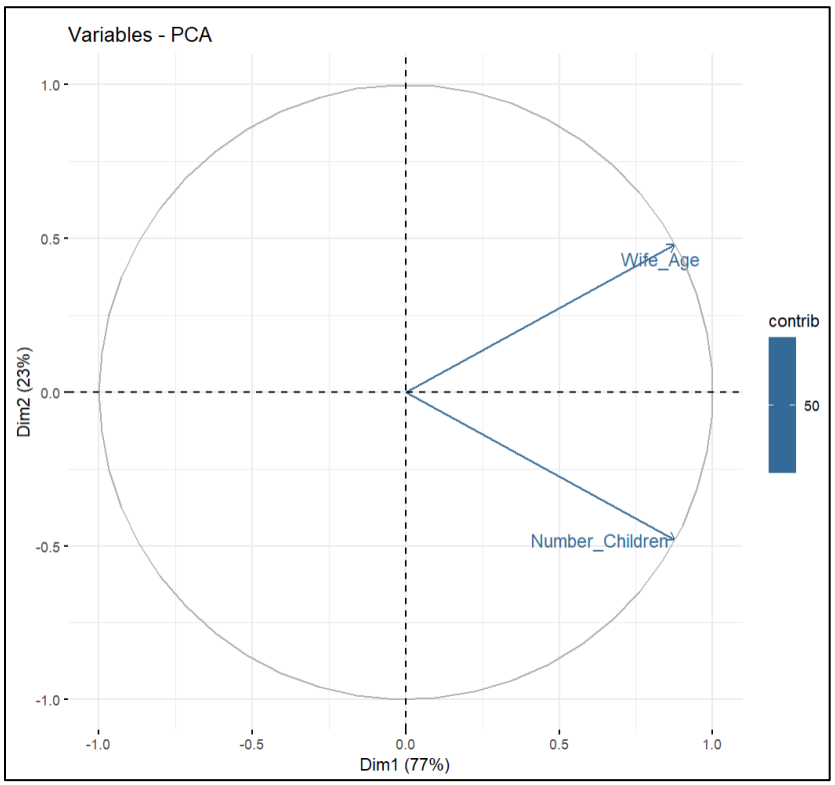
Abbiamo inoltre analizzato il grado di correlazione tra le variabili di partenza, Wife's Age e Number of Children, e le due dimensioni ottenute. Il risultato ottenuto è visibile in ***Figura 23***.

```
> cmc.pca$var$cor
                    Dim.1      Dim.2
Wife_Age        0.8775323  0.4795175
Number_Children 0.8775323 -0.4795175
```

Figura 23 - Correlazione tra variabili e PC

Si nota una alta correlazione tra le due variabili di partenza e la prima dimensione, e una bassa correlazione con la seconda dimensione.

Successivamente si è analizzato graficamente il contributo delle due variabili nella definizione dei due Principal Components. In ***Figura 24*** possiamo vedere il risultato ottenuto.



Si nota che le due variabili contribuiscono in egual modo nella definizione dei due Principal Components.

Figura 24 - Plot relativo al contributo delle due variabili nella definizione dei PC

Ora, la PCA svolta conduce alla seguente conclusione: è possibile trasformare le variabili Wife's Age e Number of Children in un'unica componente conservando il 77% della varianza complessiva delle variabili originali. A questo punto ci si è chiesti se la perdita di informazioni derivante dalla PCA potesse giustificare la riduzione della dimensione del problema di una sola variabile. Per rispondere a questa domanda si è proceduto a confrontare le performance di alcuni modelli di Machine Learning (che verranno descritti in seguito) sul dataset originale e sul dataset trasformato a seguito della PCA. Questo confronto ha mostrato che i modelli addestrati sul dataset originale hanno conseguito performance nettamente migliori rispetto a quelle raggiunte dai modelli addestrati sul dataset trasformato con la PCA. In questo quadro per non appesantire la presente relazione si è deciso di svolgere analisi più approfondite solo sui modelli addestrati sul dataset originale; mentre al confronto dei modelli sul dataset originale e sul dataset trasformato con la PCA verrà dedicato l'***Approfondimento A***.

Modelli utilizzati

Sin qui è stata fornita un'analisi esplorativa del dataset ed è stata descritta la PCA eseguita su di esso. Ora si procederà a illustrare i modelli di Machine Learning utilizzati per predire l'utilizzo o meno del contraccettivo da parte delle donne oggetto di studio.

Si procederà con il seguente ordine. In primo luogo verranno descritti i modelli di alberi decisionali addestrati. In secondo luogo verranno descritte le reti neurali utilizzate. In terzo luogo verranno presentate alcune considerazioni in merito alla distinzione tra problema binario e multi-classe. Successivamente verrà motivata la scelta di adottare una tecnica di Cross-Validation, e verranno forniti alcuni dettagli in merito all'implementazione dei modelli nel linguaggio R. Infine verrà fornita una sintesi dei modelli implementati. I risultati degli esperimenti condotti utilizzando questi modelli verranno esposti nel capitolo successivo.

Alberi decisionali

1. Motivazioni della scelta del modello

La prima tipologia di modello predittivo utilizzato è stata quella degli alberi decisionali. In particolare, gli alberi decisionali sono stati scelti per le seguenti ragioni:

- Il dataset utilizzato presenta diverse variabili categoriche e due variabili numeriche. Gli alberi decisionali sono in grado di gestire sia variabili categoriche che numeriche. Pertanto rappresentano dei buoni candidati per la scelta del modello da addestrare sul dataset utilizzato;
- Gli alberi decisionali sono poco sensibili alla presenza di outliers e valori mancanti[1]. Ora, il dataset utilizzato non presenta valori mancanti. Tuttavia presenta alcuni outliers, come evidenziato quando si è discusso della variabile Number of Children. Pertanto anche per questo motivo gli alberi decisionali sono dei buoni modelli candidati;
- Gli alberi decisionali permettono di classificare anche istanze non linearmente separabili. E questo suggerisce di preferire questi modelli ad altri che invece richiedono istanze linearmente separabili, come per esempio il percettrone semplice (non multistrato);
- Gli alberi decisionali sono modelli di apprendimento supervisionato. Questi modelli possono quindi sfruttare la variabile target del dataset che specifica l'utilizzo o meno del contraccettivo da parte delle donne. Pertanto anche questa circostanza porta a considerare gli alberi di decisione come buoni modelli candidati;
- Gli alberi decisionali sono facilmente interpretabili. Questo consente di inferire conclusioni sostanziali che vadano oltre le mere misure di performance del modello.

[1] Questo si verifica in ragione della costruzione degli alberi. Infatti i nodi degli alberi determinano uno split delle istanze in base al valore dell'attributo considerato. In questo modo si prendono in considerazione regioni di valori, invece di valori assoluti. E questo permette agli alberi di sopportare bene l'eventuale presenza di outliers nel dataset.

2. Descrizione degli alberi decisionali

Già si è specificato che gli alberi di decisione sono modelli di apprendimento supervisionato. In particolare, gli alberi di decisione sono grafi costruiti come segue:

- Ogni nodo interno dell'albero rappresenta una variabile;
- Un arco verso un nodo figlio determina un valore (o un insieme di valori) per quella variabile;
- Una foglia dell'albero identifica il valore predetto per la variabile target. Questo avviene sulla base dei valori delle variabili considerate nel cammino dalla radice alla foglia.

Ora, esistono diversi modelli di Machine Learning che appartengono alla famiglia degli alberi di decisione. Per questo progetto è stato considerato il modello CART (acronimo di Classification And Regression Tree). Questo modello genera un albero di decisione binario nel quale la scelta dello split (e quindi della variabile e dei valori da considerare per i nodi figli) viene fatta sulla base dell'indice di diversità di *Gini*. In particolare, questo indice è definito come:

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2$$

dove *J* è il numero di valori di una variabile, *i* è uno di questi valori e p_i è la frazione di elementi dell'insieme che assumono il valore *i*. Più precisamente, viene scelto lo split che minimizza l'indice di Gini.

Una caratteristica dell'albero generato dall'algoritmo CART è la possibilità di avere più nodi etichettati con la stessa variabile.

Reti neurali

1. Motivazioni della scelta del modello

La seconda tipologia di modello predittivo utilizzato è stata quella delle Rete Neurali. In particolare, le reti neurali sono state scelte per le seguenti ragioni:

- Le reti neurali sono molto tolleranti alla presenza di errori oppure di rumore. Anche se il nostro dataset non presenta valori mancanti, contiene outliers per certe variabili, come ad esempio Number of Children, quindi, si è pensato che questo modello fosse efficace perché tollerante a queste situazioni;
- Le reti neurali, come anche gli alberi decisionali, sono in grado di classificare anche pattern complessi non linearmente separabili, per questo si è preferito un modello del genere al posto del percettrone semplice;
- Le reti neurali sono modelli di apprendimento supervisionato. Questi sfruttano la variabile target del dataset che specifica l'utilizzo o meno del contraccettivo da parte delle donne. Anche questa circostanza porta a considerare le reti neurali come buoni modelli candidati;
- Le reti neurali sono facilmente aggiornabili con eventuali nuove osservazioni;

- Si sono preferite le reti neurali rispetto alle Support Vector Machine perché questo modello, nella sua forma nativa, non supporta la classificazione multi-classe. Volendo noi confrontare le prestazioni del modello sia sul problema binario che su quello multi-classe, abbiamo preferito la soluzione più semplice, seguendo il principio del rasoio di Occam.

Svantaggio delle reti neurali è che non sono in grado di trattare variabili categoriche, prevalenti nel nostro dataset. Comunque, a seguito di una adeguata rappresentazione di tali variabili in tante variabili binarie quanti sono i livelli della covariata fattoriale, si è stati in grado di gestire anche questo inconveniente. Questo è stato anche possibile perché le nostre variabili categoriche presentano al più quattro livelli quindi, anche dopo la rappresentazione di tali variabili categoriche in più variabili binarie, il numero di covariate non è diventato troppo elevato.

Ad occuparsi della rappresentazione del dataset in forma adeguata all'apprendimento tramite rete neurale ci ha pensato in automatico la funzione ***train*** del package ***caret***, funzione utilizzata per l'addestramento della rete.

2. Descrizione delle reti neurali

Le reti neurali, come specificato in precedenza, sono modelli di apprendimento supervisionato che, a differenza del percettrone semplice dove è presente un unico neurone, e permette la classificazione solo di insiemi linearmente indipendenti, permettono la definizione di reti di neuroni (come dice il nome) organizzate in diversi livelli. Questo consente alle reti neurali di lavorare anche su insiemi complessi non linearmente separabili.

Andando nello specifico, le reti neurali vengono rappresentate graficamente come dei grafi, dove ogni nodo è un neurone, e i nodi sono connessi tra di loro da archi orientati e pesati:

- Sono presenti dei neuroni di input, uno per ogni covariata da considerare per il calcolo del valore della variabile target;
- I neuroni di input sono collegati tramite archi pesati a tutti i neuroni definiti per il primo livello della rete. Inoltre, ogni neurone dei diversi livelli riceve in input anche un valore da un neurone aggiuntivo. Questo valore è detto soglia, e viene considerato per capire se tale neurone è da considerarsi attivo oppure no;
- I neuroni del primo livello sono collegati con archi pesati ai neuroni del secondo livello, e così via fino ad arrivare all'ultimo livello, dove sono presenti i neuroni di output;
- Per capire se un neurone è da considerarsi attivo oppure no si fa riferimento ad una specifica funzione di attivazione. Questa funzione, presi in input i segnali ricevuti dal neurone, calcola il suo valore. Se questo valore supera una certa soglia, allora il neurone sarà considerato attivo, altrimenti disattivo;
- L'ultimo livello della rete presenta tanti neuroni quanti sono i possibili valori assumibili dalla variabile target, nel caso questa fosse una variabile categorica come nel nostro caso. Ognuno di questi neuroni di output, data in input alla rete una istanza del problema, fornirà la probabilità che l'istanza sia classificata in quel tal modo;

Per il calcolo dei pesi ottimali da associare ai diversi archi si fa uso della strategia di back propagation. Questa strategia prevede l'inizializzazione di tutti i pesi in modo casuale; successivamente, viene considerata ogni istanza del training set, e per questa istanza viene calcolato tramite la rete il possibile valore per la variabile target: se il valore calcolato dalla rete coincide con quello effettivo, allora non avvengono modifiche; se, invece, il valore calcolato è diverso da quello atteso, si calcola l'errore, e si propaga il calcolo all'indietro a partire dai neuroni di output, aggiustando a mano a mano i pesi dei diversi archi.

Per quanto riguarda la funzione di attivazione considerata nei nostri esperimenti, si è fatto riferimento alla funzione logistica, così definita:

$$\mathbf{f(x)} = \frac{1}{1 + \mathbf{e}^{-x}}$$

ove $x = \Sigma w_i s_i$

Il valore ***x*** viene calcolato come sommatoria pesata dei valori forniti dai neuroni in input. Si noti che questa sommatoria considera anche il valore di soglia, perché viene fornito anch'esso in input da un neurone definito appositamente.

Problema binario e multi-classe

Nel paragrafo dedicato alla descrizione dei dati si è precisato che il dataset originale presentava una variabile target con tre valori possibili: 1 (contraccettivo non usato), 2 (usato contraccettivo a lungo termine) oppure 3 (usato contraccettivo a breve termine). Alcune motivazioni hanno tuttavia suggerito di trasformare il problema multi-classe in problema binario. In particolare:

- Il problema multi-classe si presenta sbilanciato. Infatti le istanze del dataset originale sono così suddivise: 629 istanze per l'assenza di contraccettivi, 333 istanze per i contraccettivi a lungo termine e 511 istanze per i contraccettivi a breve termine. Questo sbilanciamento e in particolare il ridotto numero di istanze per la classe relativa ai contraccettivi a lungo termine paventa il rischio di una difficile classificazione quantomeno di quest'ultima classe. Il problema binario invece risulta più bilanciato, come mostrato in precedenza;
- L'algoritmo CART genera un albero decisionale binario. Pertanto questo albero risulta naturalmente più adatto a trattare un problema binario, sebbene possa trattare anche problemi multi-classe con una maggiore complessità e profondità dell'albero. Le reti neurali non hanno in ogni caso problemi a trattare problemi binari.

Ora, queste considerazioni depongono a favore della trasformazione del problema multi-classe in problema binario. Tuttavia per mantenere un maggior rigore scientifico e per confermare (o smentire) le precedenti considerazioni, si è ritenuto più opportuno trattare separatamente sia il problema binario che il problema multi-classe. Pertanto si è deciso di addestrare i modelli sia sul problema binario che sul problema multi-classe.

Cross-Validation

Il dataset utilizzato presenta 1473 istanze. Si è ritenuto che la dimensione del dataset non fosse abbastanza grande da consentirne una semplice divisione in train set e test set. Per questo motivo si è pensato di ricorrere a una tecnica di Cross-Validation che prevede la suddivisione casuale del dataset in sottoinsiemi di uguale dimensione e considera a ogni passo un sottoinsieme diverso per il testing e il resto per il training.

Ora, tra le tecniche di Cross-Validation si è esclusa la tecnica Leave-One-Out Cross-Validation. Questa tecnica prevede di considerare a ogni iterazione una sola istanza come test set. E solitamente si ricorre a questa tecnica quando il dataset è molto piccolo in quanto onerosa dal punto di vista computazionale. Si è ritenuto che il dataset oggetto di studio non fosse tanto piccolo da giustificare il ricorso a questa tecnica di Cross-Validation.

Si è deciso di utilizzare invece la tecnica 10-fold Cross-Validation. In questo caso si prevede la suddivisione casuale del dataset in 10 fold di uguale dimensione. A ogni passo della Cross-Validation uno dei fold viene utilizzato per la validazione (test) e i restanti 9 fold vengono utilizzati per il training. La matrice di confusione del modello è calcolata come somma delle matrici di confusione delle iterazioni.

In particolare, si è deciso di utilizzare la 10-fold Cross-Validation in due modi:

- Una prima modalità prevede l'utilizzo della 10-fold Cross-Validation sull'intero dataset;
- Una seconda modalità prevede anzitutto (*i*) la suddivisione casuale del dataset in train set e test set rappresentanti rispettivamente il 70% e il 30% delle istanze; poi (*ii*) l'esecuzione di una 10-fold Cross-Validation ripetuta 3 volte sul train set; e infine (*iii*) il test del modello addestrato sul test set. In questo caso si ritiene che la ripetizione della 10-fold Cross-Validation consenta una buona stima della performance del modello nonostante venga eseguita su un train set non particolarmente grande.

Implementazione in R

In questo paragrafo verranno forniti alcuni dettagli relativi all'implementazione dei modelli in R. In particolare:

1) Alberi decisionali. Per l'implementazione degli alberi decisionali in R è stato sfruttato il package ***caret***. Questo package comprende un insieme di funzioni utili per la creazione di modelli di classificazione e regressione. In particolare, questo package presenta le funzioni ***train*** e ***predict*** che consentono rispettivamente di addestrare e testare un modello predittivo. Inoltre questo package permette di definire una serie di opzioni di addestramento attraverso la funzione ***trainControl***, che definisce i parametri utili per la funzione ***train***. Tra queste opzioni vi è in particolare la possibilità di richiedere l'esecuzione di una 10-fold Cross-Validation (eventualmente ripetuta) e di memorizzare le predizioni e le probabilità restituite da ogni sua iterazione.

Il package ***caret*** sfrutta a sua volta il package ***rpart***. Il nome del package ***rpart*** è un'abbreviazione di Recursive Partitioning and Regression Trees. Questo suggerisce che il package ***rpart*** serve per la creazione di alberi di classificazione e regressione. Il package ***rpart*** utilizza l'algoritmo CART come algoritmo standard per la creazione di un albero decisionale. Pertanto risulta particolarmente appropriato alla costruzione dei modelli di alberi decisionali descritti nei paragrafi precedenti.

2) Reti neurali. Per l'implementazione delle reti neurali in R, come fatto anche per gli alberi decisionali, si è fatto riferimento alle funzionalità offerte dal package ***caret***, in particolare alle funzioni ***train***, per l'addestramento della rete, e ***predict***, per la predizione della variabile target sul test set. A differenza degli alberi decisionali, si è utilizzato come metodo di training il metodo ***nnet***, definito proprio per l'addestramento di una rete neurale. Infatti, il package ***caret*** sfrutta a sua volta il package ***nnet*** per quanto riguarda le reti neurali.

Sintesi

Sin qui è stata fornita una descrizione dei modelli costruiti. In questo paragrafo si procederà a fornirne una sintesi e a mostrarne alcune prime caratteristiche.

1. Alberi decisionali

Cominciamo con i modelli per gli alberi decisionali.

1) Primo albero decisionale (in seguito: DT totale binario). Un primo modello è stato addestrato tramite 10-fold Cross-Validation sull'intero dataset relativo al problema binario. L'albero originato è mostrato in ***Figura 25***.

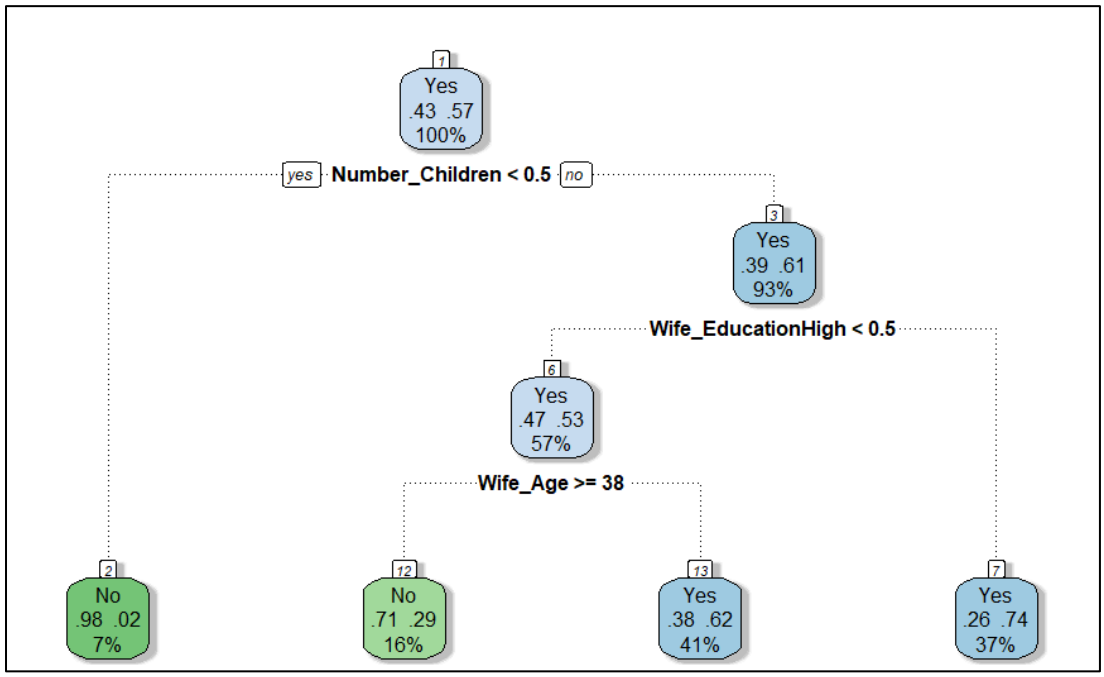


Figura 25 – Albero decisionale realizzato con 10-fold Cross-Validation sull'intero dataset del problema binario

Tramite la funzione ***train*** di ***caret***, l'albero è stato generato in modo da massimizzare la metrica ROC (e conseguentemente il valore AUC). Pertanto si è ritenuto di non procedere a un'ulteriore potatura dell'albero (*pruning*) accettando il Complexity Parameter (in seguito: CP) determinato dalla funzione ***train***. Per completezza si riporta comunque il valore del CP.

```
ROC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01589825.
```

2) <u>Secondo albero decisionale</u> (in seguito: DT split binario). Un secondo modello è stato addestrato tramite 10-fold Cross-Validation (ripetuta 3 volte) sul 70% delle istanze del dataset (train set) relativo al problema binario. L'albero originato è mostrato in ***Figura 26***.

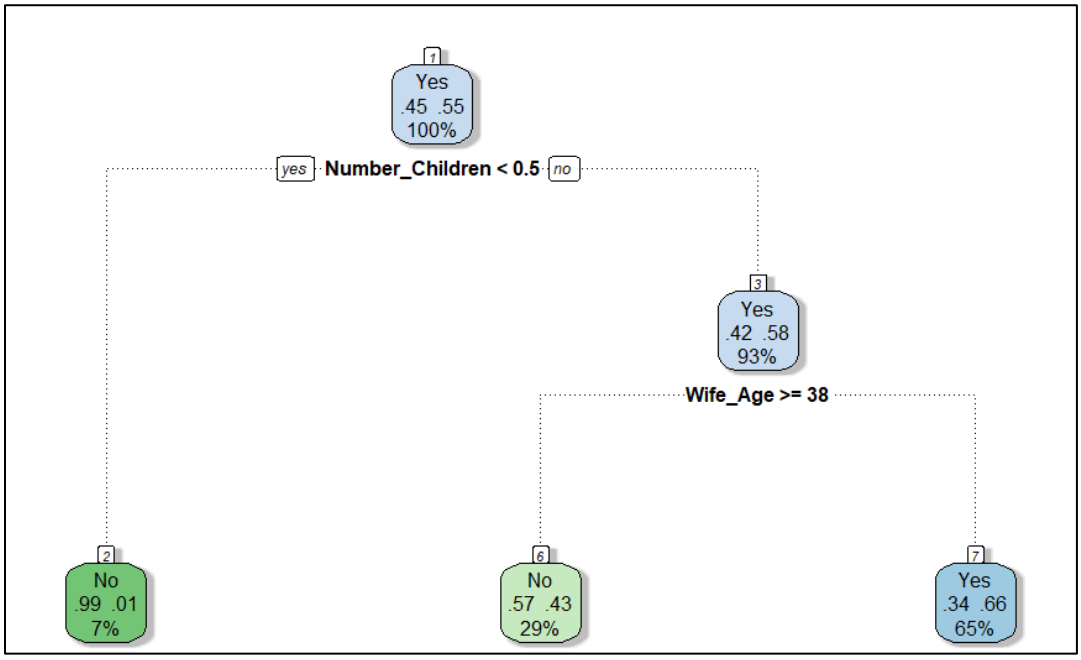


Figura 26 – Albero decisionale realizzato con 10-fold Cross-Validation su dataset diviso per il problema binario

Anche in questo caso è stata massimizzata la metrica ROC (e conseguentemente il valore AUC). Pertanto anche in questo caso non si è proceduto a una potatura dell'albero. Il valore del CP è il seguente.

```
ROC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.08119658.
```

3) Terzo albero decisionale (in seguito: DT totale multi). Un terzo modello è stato addestrato tramite 10-fold Cross-Validation sull'intero dataset relativo al problema multi-classe. L'albero originato è mostrato in ***Figura 27***.

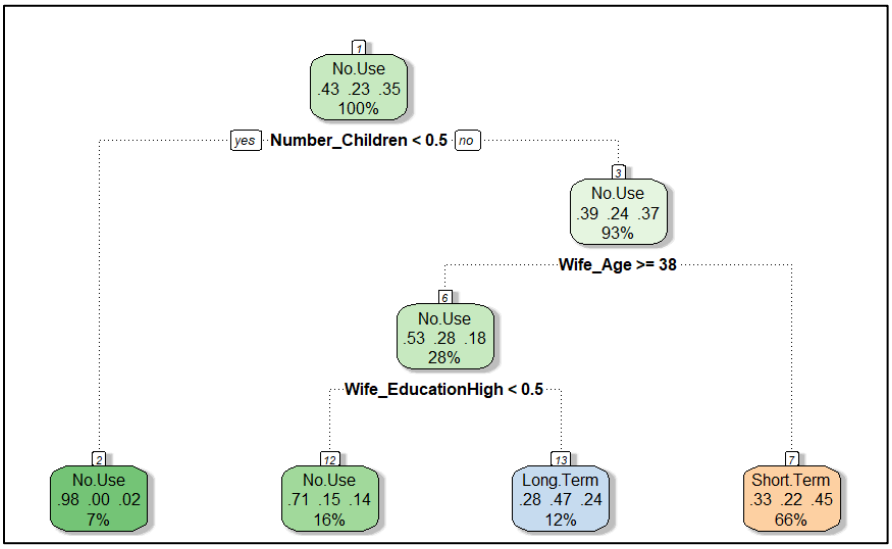


Figura 27 – Albero decisionale realizzato con 10-fold Cross-Validation sull'intero dataset del problema multi-classe

Tramite la funzione ***train*** di ***caret***, l'albero è stato generato in modo da massimizzare il valore AUC. Pertanto anche in questo caso si è ritenuto di non procedere a un'ulteriore potatura dell'albero. Il valore del CP è il seguente.

```
AUC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.02310427.
```

4) Quarto albero decisionale (in seguito: DT split multi). Un quarto modello è stato addestrato tramite 10-fold Cross-Validation (ripetuta 3 volte) sul 70% delle istanze del dataset (train set) relativo al problema multi-classe. L'albero originato è mostrato in ***Figura 28***.

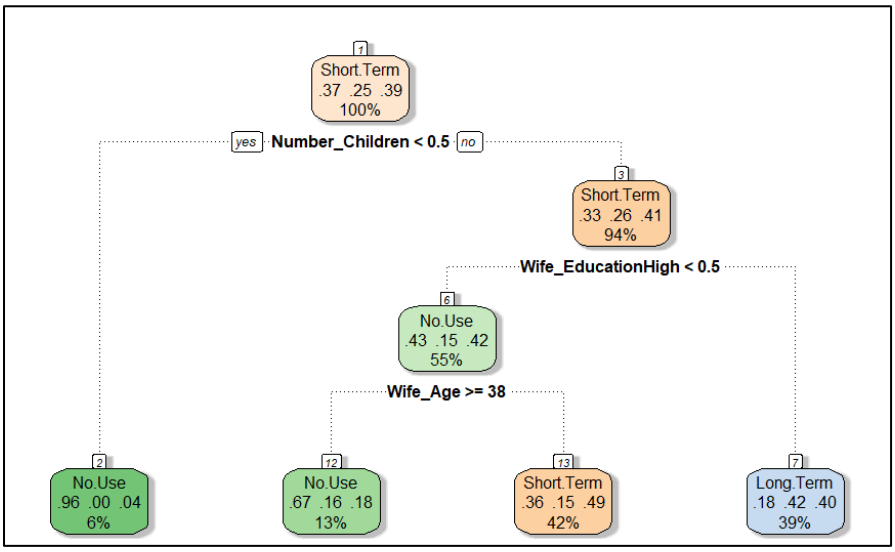


Figura 28 – Albero decisionale realizzato con 10-fold Cross-Validation su dataset diviso per il problema multi-classe

Anche in questo caso è stato massimizzato il valore AUC. Pertanto anche in questo caso non si è proceduto a una potatura dell'albero. Il valore del CP è il seguente.

```
AUC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.05555556.
```

Da questi alberi è possibile trarre alcune considerazioni:

- Tutti gli alberi partono dalla stessa radice: Number_Children < 0.5. A questo proposito si è già evidenziato come il numero di figli influenzi l'utilizzo o meno di un contraccettivo. Pertanto è ragionevole utilizzare questa informazione per classificare le istanze;
- Tutti gli alberi usano principalmente come nodi interni i valori delle variabili Wife_EducationHigh e Wife_Age. A questo proposito si è già evidenziato come le donne con un elevato livello di educazione siano più portate a usare contraccettivi. Pertanto è ragionevole utilizzare questa variabile nella classificazione delle istanze. L'analisi della covariata Wife_Age aveva invece portato a ritenere che fosse poco discriminante rispetto alle classi. Tuttavia questa variabile risulta avere un'importanza molto elevata nella costruzione degli alberi decisionali. Per mostrare ciò si riportano a titolo esemplificativo i valori di importanza delle variabili per l'albero decisionale mostrato in ***Figura 25***;

```
rpart variable importance

                              Overall
Wife_Age                      100.000
Number_Children                93.536
Wife_EducationHigh             68.201
Media_ExposureNot-Good         49.427
Husband_EducationHigh          31.579
Wife_EducationMid-High         12.130
Husband_OccupationMid-Low       2.744
Living_IndexHigh                0.000
`Husband_OccupationMid-Low`     0.000
Wife_Is_WorkingNo               0.000
`Husband_EducationMid-Low`      0.000
`Wife_EducationMid-High`        0.000
`Media_ExposureNot-Good`        0.000
Husband_OccupationHigh          0.000
`Husband_EducationMid-High`     0.000
`Living_IndexMid-High`          0.000
`Husband_OccupationMid-High`    0.000
`Living_IndexMid-Low`           0.000
Wife_ReligionIslam              0.000
`Wife_EducationMid-Low`         0.000
```

Figura 29 – Importanza variabili albero decisionale realizzato con 10-fold Cross-Validation sull'intero dataset per il problema binario

- In questo caso si osserva come solo un ristretto sottoinsieme delle variabili del dataset contribuisca a determinare gli alberi decisionali. E questo potrebbe influire negativamente sull'accuratezza dei modelli nella misura in cui non vengono prese in considerazione variabili che potrebbero contribuire a migliorare la capacità di predizione.

2. Reti neurali

Passiamo ora ai modelli di reti neurali.

1) Prima rete neurale (in seguito: NN totale binario). Un primo modello è stato addestrato tramite 10-fold Cross-Validation sull'intero dataset relativo al problema binario. La rete ottenuta è mostrata in ***Figura 30***.

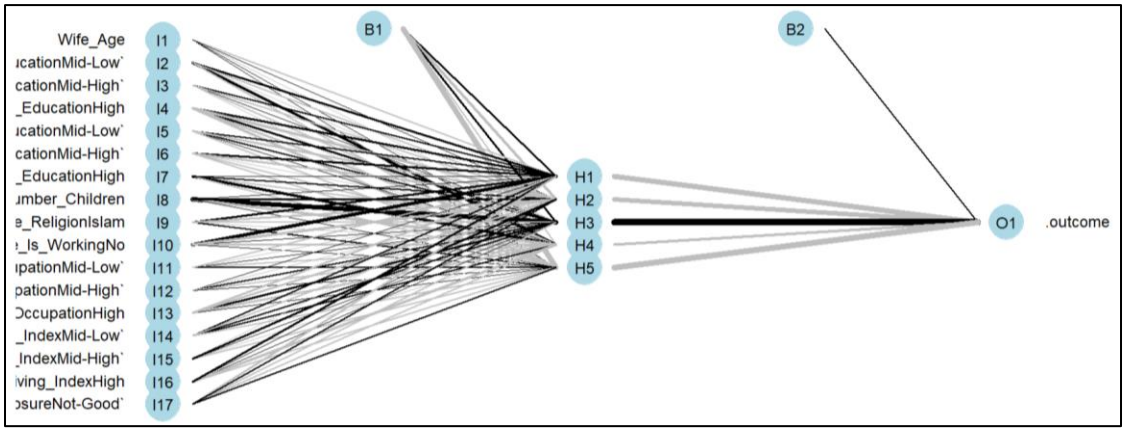


Figura 30 - Rete neurale ottenuta da intero dataset binario

Il ***train*** della rete è stato fatto con l'omonima funzione offerta da package ***caret***, andando a massimizzare la metrica ROC. Per quanto riguarda l'importanza delle variabili nella determinazione dell'outcome, in ***Figura 31*** possiamo vedere il risultato ottenuto. Si noti come mediamente tutte le variabili abbiano una importanza abbastanza alta nella determinazione della variabile target.

	Overall
Wife_EducationHigh	100.000
Number_Children	98.833
Wife_Is_WorkingNo	67.284
Husband_OccupationHigh	58.881
Wife_EducationMid-Low	47.182
Media_ExposureNot-Good	46.861
Wife_ReligionIslam	46.595
Living_IndexHigh	46.295
Husband_EducationMid-High	36.264
Husband_EducationMid-Low	34.676
Husband_EducationHigh	34.204
Living_IndexMid-Low	21.578
Living_IndexMid-High	19.622
Wife_EducationMid-High	16.687
Husband_OccupationMid-Low	12.809
Husband_OccupationMid-High	3.824
Wife_Age	0.000

Figura 31 - Output della funzione varImp su nn_total_bin

In particolare, valori di Wife Education pari ad High sono stati molto importanti. Come avevamo infatti osservato anche tramite l'analisi esplorativa, le donne con alto livello di education sono più portate a utilizzare il contraccettivo.

Si noti, inoltre, l'importanza del numero di figli: come osservato anche durante l'analisi esplorativa, si ha che all'aumentare del numero di figli si tende di più a non utilizzare il contraccettivo.

2) Seconda rete neurale (in seguito: NN split binario). Il secondo modello è stato addestrato tramite 10-fold Cross-Validation (ripetuta 3 volte) sul 70% delle istanze del dataset (train set) relativo al problema binario. La rete ottenuta è mostrata in ***Figura 32***.

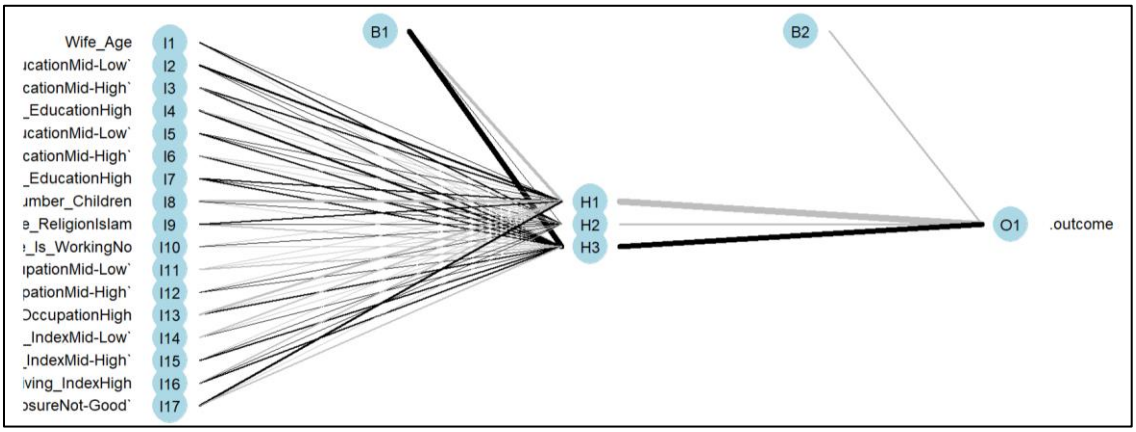


Figura 32 - Rete neurale ottenuta da train set binario

Il ***train*** della rete è stato fatto con l'omonima funzione del package ***caret***, andando a massimizzare la metrica ROC. A differenza del modello precedente, addestrato sull'intero dataset, essendo questo modello addestrato su un insieme ridotto dei dati, si è deciso di ripetere la procedura di 10-Fold Cross Validation per tre volte, in modo da ottenere un risultato più consistente, a scapito del tempo necessario per il training. Per quanto riguarda l'importanza delle covariate nella determinazione dell'outcome, in ***Figura 33*** possiamo vedere il risultato ottenuto.

	Overall
Wife_Age	100.000
Number_Children	35.769
Wife_EducationMid-Low	31.003
Media_ExposureNot-Good	30.333
Wife_EducationHigh	28.948
Husband_OccupationHigh	25.763
Husband_EducationHigh	25.169
Living_IndexHigh	23.029
Wife_ReligionIslam	22.639
Living_IndexMid-High	15.607
Wife_EducationMid-High	13.908
Husband_EducationMid-High	7.688
Husband_EducationMid-Low	7.464
Husband_OccupationMid-Low	7.363
Husband_OccupationMid-High	5.012
Wife_Is_WorkingNo	2.580
Living_IndexMid-Low	0.000

Figura 33 - Output della funzione varImp su nn_split_bin

Si noti come, a differenza del modello precedente, ora sia l'età della moglie la variabile più importante, seguita dal numero di figli;

3) Terza rete neurale (in seguito: NN totale multi). Il terzo modello è stato addestrato tramite 10-fold Cross-Validation sull'intero dataset relativo al problema multi-classe. La rete ottenuta è mostrata in ***Figura 34***.

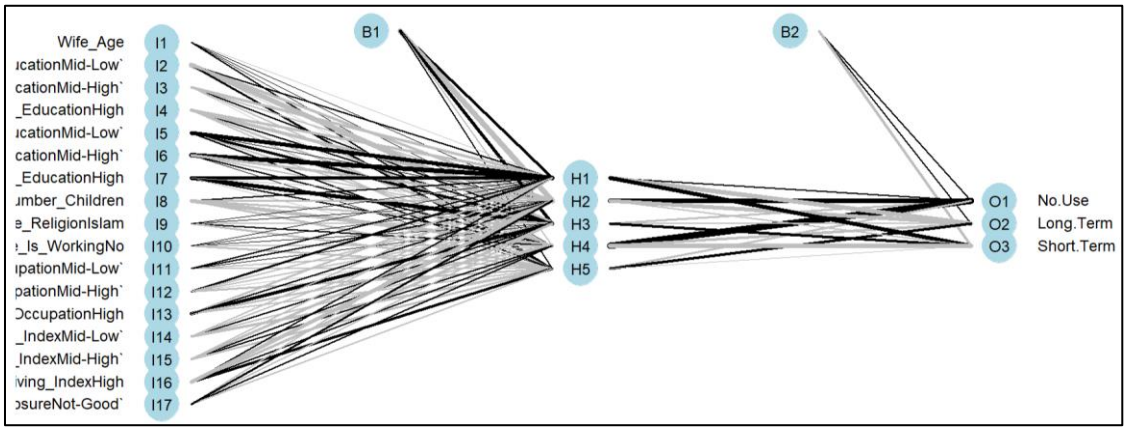


Figura 34 - Rete neurale ottenuta dal dataset originale per il problema multi-classe

Il ***train*** è stato fatto con l'omonima funzione del package ***caret*** applicata all'intero dataset, andando ad ottimizzare la misura AUC. Per quanto riguarda l'importanza delle covariate nella selezione del miglior valore per la variabile target, il risultato ottenuto è mostrato in ***Figura 35***.

Si noti come ancora sia molto importante l'educazione della moglie e il numero di figli per determinare la variabile target. A differenza di ciò che si è visto in precedenza, viene data più importanza ad alti valori di Living Index: questo è ragionevole visto che anche nell'analisi esplorativa si era notato che alti livelli di benessere sociale portano ad un maggiore utilizzo del contraccettivo.

Si noti che, ancora una volta, l'età della moglie non viene considerata;

variables are sorted by maximum importance across the classe

	Overall	No.Use	Long.Term	Short.Term
Living_IndexHigh	100.00	100.00	100.00	100.00
Number_Children	95.51	95.51	95.51	95.51
Wife_EducationMid-Low	94.75	94.75	94.75	94.75
Husband_OccupationHigh	78.24	78.24	78.24	78.24
Wife_EducationHigh	73.58	73.58	73.58	73.58
Husband_EducationMid-High	71.30	71.30	71.30	71.30
Husband_EducationHigh	67.64	67.64	67.64	67.64
Husband_EducationMid-Low	66.22	66.22	66.22	66.22
Living_IndexMid-High	59.30	59.30	59.30	59.30
Living_IndexMid-Low	44.14	44.14	44.14	44.14
Wife_EducationMid-High	34.16	34.16	34.16	34.16
Husband_OccupationMid-High	32.09	32.09	32.09	32.09
Media_ExposureNot-Good	30.05	30.05	30.05	30.05
Wife_ReligionIslam	28.00	28.00	28.00	28.00
Wife_Is_WorkingNo	21.60	21.60	21.60	21.60
Husband_OccupationMid-Low	11.08	11.08	11.08	11.08
Wife_Age	0.00	0.00	0.00	0.00

Figura 35 - Output della funzione varImp su nn_total_multi

4) Quarta rete neurale (in seguito: NN split multi). Il quarto modello è stato addestrato tramite 10-fold Cross-Validation (ripetuta 3 volte) sul 70% delle istanze del dataset (train set) relativo al problema multi-classe. La rete ottenuta è mostrata in ***Figura 36***.

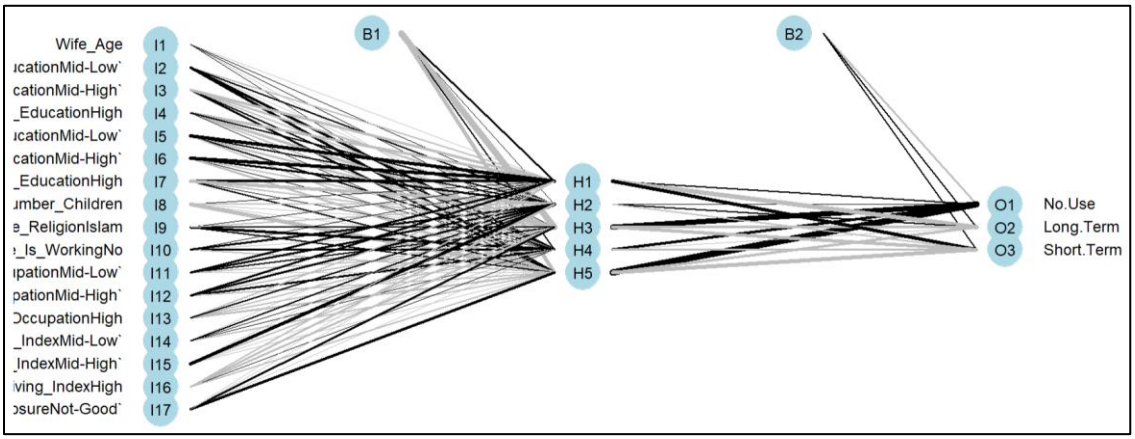


Figura 36 - Rete neurale ottenuta da training set multi-classe

Il ***train*** è stato fatto con l'omonima funzione offerta dal package ***caret***, andando a massimizzare la misura di AUC, a differenza del corrispettivo binario, in cui si è massimizzato ROC. Anche in questo caso si è adottata una strategia di training del tipo 10-Fold CV ripetuta tre volte, in modo da ottenere risultati consistenti con pochi dati. Per quanto riguarda l'importanza delle variabili nella determinazione dell'outcome, il risultato ottenuto viene mostrato in ***Figura 37***.

variables are sorted by maximum importance across the classes

	Overall	No.Use	Long.Term	Short.Term
Husband_EducationMid-Low	100.00	100.00	100.00	100.00
Husband_EducationHigh	91.91	91.91	91.91	91.91
Wife_EducationMid-Low	87.87	87.87	87.87	87.87
Husband_EducationMid-High	86.35	86.35	86.35	86.35
Husband_OccupationMid-High	80.69	80.69	80.69	80.69
Wife_EducationHigh	77.41	77.41	77.41	77.41
Wife_EducationMid-High	74.91	74.91	74.91	74.91
Wife_ReligionIslam	70.32	70.32	70.32	70.32
Living_IndexHigh	68.40	68.40	68.40	68.40
Husband_OccupationHigh	63.97	63.97	63.97	63.97
Media_ExposureNot-Good	59.05	59.05	59.05	59.05
Number_Children	51.37	51.37	51.37	51.37
Living_IndexMid-High	50.82	50.82	50.82	50.82
Wife_Is_WorkingNo	44.91	44.91	44.91	44.91
Husband_OccupationMid-Low	39.65	39.65	39.65	39.65
Living_IndexMid-Low	11.81	11.81	11.81	11.81
Wife_Age	0.00	0.00	0.00	0.00

Figura 37 - Output della funzione varImp su nn_split_multi

Si noti come in media tutte le variabili siano importanti per la determinazione del target, a meno di Wife Age;

Esperimenti eseguiti

Nei paragrafi precedenti sono stati descritti i modelli implementati. Ora si procederà a presentare i risultati degli esperimenti eseguiti con essi. In particolare, verrà seguito questo ordine. In primo luogo verranno analizzate le matrici di confusione generate dai modelli. In secondo luogo verrà misurata l'accuratezza dei modelli. In terzo luogo verranno descritte alcune ulteriori misure di performance relative alla capacità di classificazione dei modelli. In quarto luogo verranno rappresentate le curve ROC e i relativi valori AUC. In quinto luogo verranno dettagliati i tempi di computazione dei modelli. Infine verrà fornito un confronto generale tra alberi decisionali e reti neurali.

Matrici di confusione

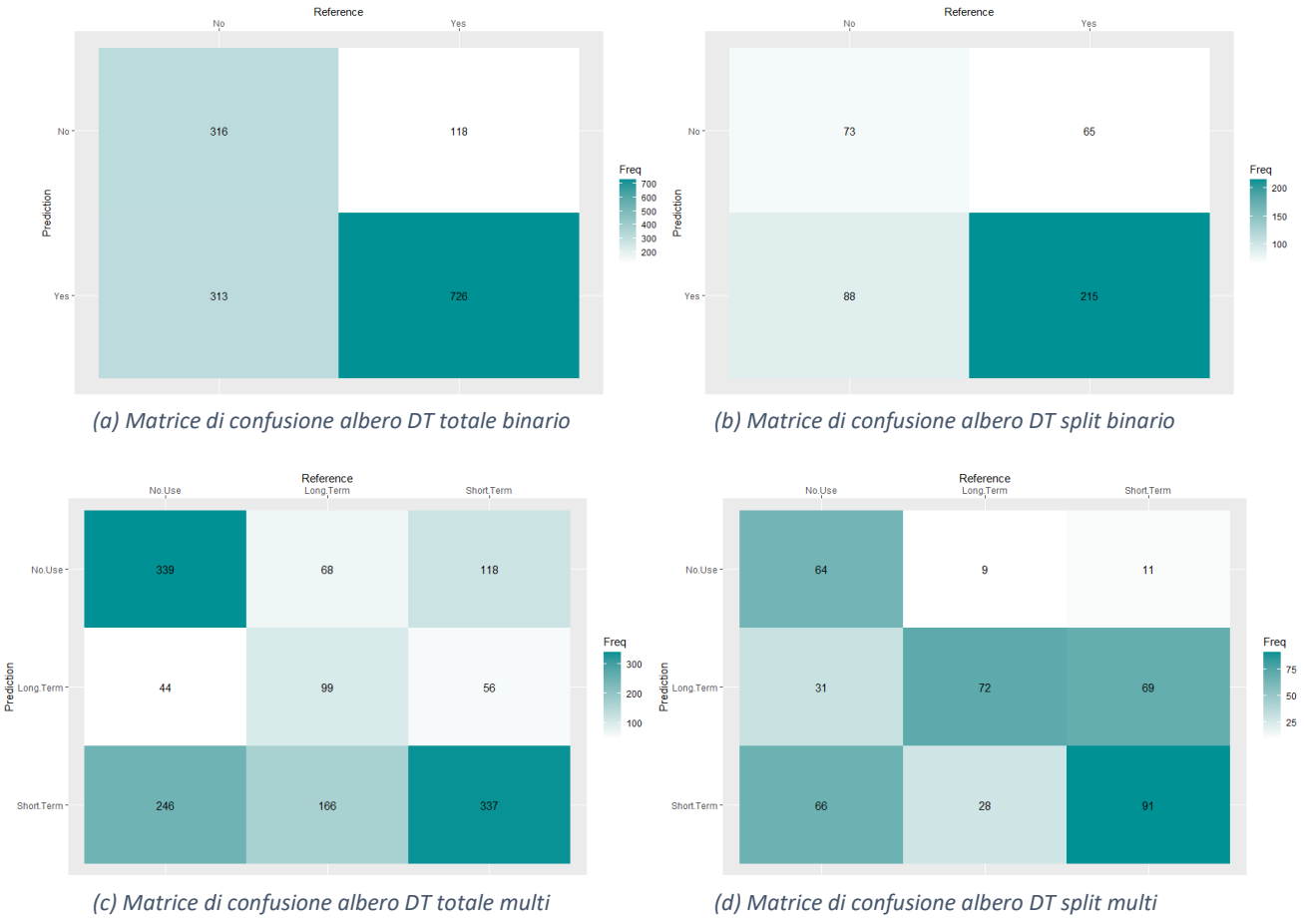
Anzitutto vengono presentate le matrici di confusione derivanti dalle predizioni effettuate dai modelli. Le matrici di confusione sono state calcolate come segue:

- Le matrici di confusione relative ai modelli addestrati sull'intero dataset sono state calcolate come somma delle matrici di confusione ottenute per ogni iterazione della 10-fold Cross-Validation;
- Le matrici di confusione relative ai modelli addestrati su una porzione del dataset sono state calcolate con riferimento alle predizioni effettuate sul test set.

Vediamo ora separatamente le matrici di confusione generate (*i*) dagli alberi decisionali e (*ii*) dalle reti neurali.

1. Alberi decisionali

Le matrici ottenute sono mostrate in ***Figura 38***.



(a) Matrice di confusione albero DT totale binario

(b) Matrice di confusione albero DT split binario

(c) Matrice di confusione albero DT totale multi

(d) Matrice di confusione albero DT split multi

Figura 38 – Matrici di confusione alberi decisionali

Da queste matrici di confusione si possono trarre alcune prime considerazioni. Anzitutto si può notare come in generale i modelli commettano molti errori di misclassificazione. Questo suggerisce che (*i*) l'accuratezza dei modelli non sarà particolarmente elevata e che (*ii*) in generale la qualità dei modelli non sarà molto buona. Inoltre si può osservare come sia per il problema binario che per il problema multi-classe i modelli predicano molto spesso l'uso di contraccettivo. Questo è ragionevole, dato che le istanze che ricadono nella classe relativa all'uso del contraccettivo sono la maggioranza. Tuttavia il problema multi-classe richiede anche la discriminazione tra i tipi di contraccettivi eventualmente utilizzati. E a questo proposito i modelli hanno difficoltà a definire correttamente questa discriminazione, commettendo molti errori di misclassificazione. In questo quadro si può già intuire come le performance dei modelli addestrati sul problema multi-classe risultino inferiori rispetto a quelle dei modelli addestrati sul problema binario.

2. Reti neurali

Le matrici ottenute sono mostrate in ***Figura 39*** e ***Figura 40***.

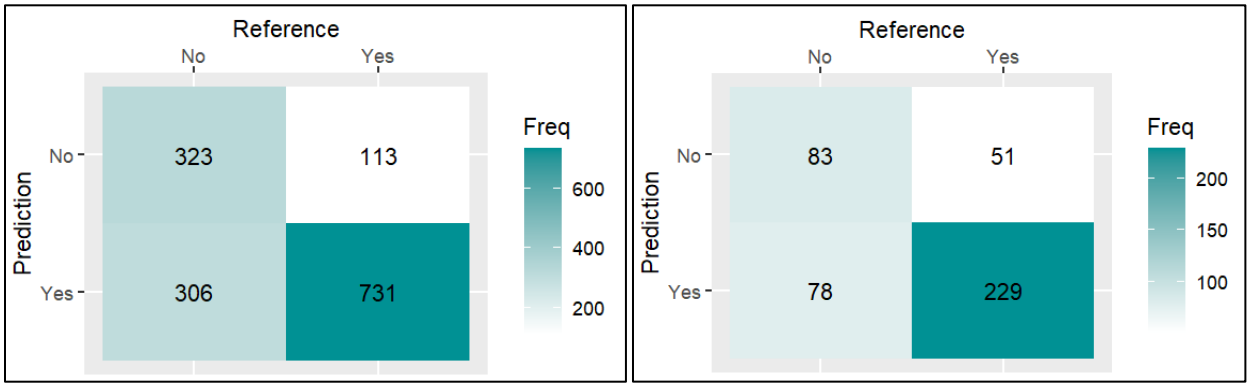


Figura 39 - Confusion Matrix per nn_total_bin (a sinistra) e nn_split_bin (a destra)

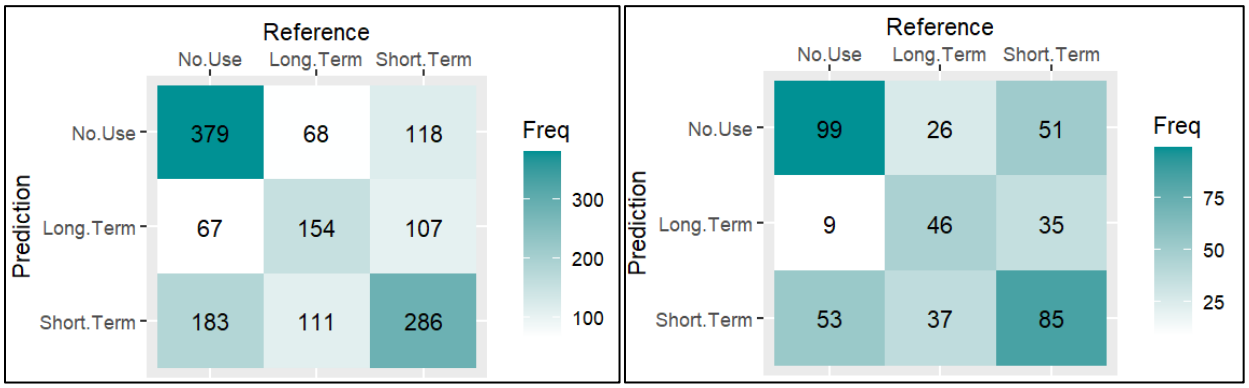


Figura 40 - Confusion Matrix per nn_total_multi (a sinistra) e nn_split_multi (a destra)

Da queste matrici di confusione si possono trarre alcune prime considerazioni. Anzitutto si può notare come i modelli definiti sul problema binario risultino più accurati di quelli definiti sul problema multi-classe. In particolare, i modelli definiti sul problema binario sembra riescano a riconoscere bene le istanze che fanno uso del contraccettivo, a differenza di quelle che non ne fanno uso. Questo è coerente con il fatto che nel dataset le istanze che fanno uso del contraccettivo sono in numero maggiore rispetto a quelle che non ne fanno uso. Si noti invece che, per quanto riguarda il problema multi-classe, i modelli addestrati riescono a discriminare bene soprattutto sul non utilizzo del contraccettivo: questo sarà dovuto anche al fatto che nel dataset, considerando le tre classi, quella più numerosa è proprio quella che non fa uso del contraccettivo.

In definitiva, anche con le reti neurali si ottengono risultati simili a quelli osservati con gli alberi decisionali, infatti, i modelli addestrati predicono bene l'utilizzo del contraccettivo sul problema binario, mentre, non hanno buone prestazioni sul problema multi-classe.

Accuratezza

L'accuratezza misura la proporzione di istanze classificate correttamente rispetto al totale delle istanze. L'accuratezza può essere calcolata agevolmente a partire dalla matrice di confusione del modello. In particolare nel caso di un problema binario l'accuratezza è calcolata come:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Dove TP e TN sono gli elementi classificati correttamente dal modello rispettivamente per la classe dei positivi e per la classe dei negativi, mentre FP e FN sono gli elementi classificati non correttamente. L'accuratezza può essere calcolata tramite la matrice di confusione sommando gli elementi della diagonale principale e dividendo questa somma per il numero totale degli elementi della matrice. Quest'ultimo procedimento può essere utilizzato anche per calcolare l'accuratezza di un modello nel caso di un problema multi-classe.

1. Alberi decisionali

L'accuratezza registrata dagli alberi decisionali è la seguente:

- Accuratezza DT totale binario: 0.7074 con intervallo di confidenza (0.6834, 0.7305);
- Accuratezza DT split binario: 0.6531 con intervallo di confidenza (0.6066, 0.6975);
- Accuratezza DT totale multi: 0.5261 con intervallo di confidenza (0.5003, 0.5519);
- Accuratezza DT split multi: 0.5147 con intervallo di confidenza (0.467, 0.5623);

Gli intervalli di confidenza sono stati calcolati con una confidenza pari al 95%. Anche dall'accuratezza registrata dai modelli si possono trarre alcune considerazioni. In primo luogo si può osservare come i modelli addestrati sul problema binario ottengano risultati nettamente migliori rispetto a quelli registrati per il problema multi-classe. In secondo luogo si può osservare come sia per il problema binario che per il problema multi-classe non si registrino grandi differenze tra l'addestramento sull'intero dataset e l'addestramento su una sola porzione. Infatti si nota come gli intervalli di confidenza siano (almeno parzialmente) sovrapposti, con un leggero vantaggio dei modelli addestrati sull'intero dataset.

In merito all'accuratezza rimane da fare un'ulteriore considerazione. Il problema oggetto di studio è noto. E come tale è già stato affrontato da diversi studiosi e ricercatori. A questo proposito si evidenzia come in un articolo del 2000 il problema in questione sia stato considerato particolarmente difficile e tale da generare un "tasso di errore minimo maggiore di 0.4"[2]. Ora, il tasso di errore è definito come: $\boldsymbol{error\ rate = 1 - accuracy}$. In questo quadro i risultati ottenuti dai modelli addestrati in termini di accuratezza sono compatibili con quanto riscontrato in articoli scientifici e possono pertanto considerarsi ragionevoli.

[2] LIM, LOH, SHIH, *A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms*, in *Machine Learning*, 40, 2000, pp. 203-228.

2. Reti neurali

L'accuratezza registrata dalle reti neurali è la seguente:

- Accuratezza DT totale binario: 0.7101 con intervallo di confidenza (0.6862, 0.7332);
- Accuratezza DT split binario: 0.7075 con intervallo di confidenza (0.6626, 0.7496);
- Accuratezza DT totale multi: 0.5628 con intervallo di confidenza (0.537, 0.5883);
- Accuratezza DT split multi: 0.5215 con intervallo di confidenza (0.4738, 0.569);

Gli intervalli di confidenza sono stati calcolati con una confidenza pari al 95%.

Alcune considerazioni che possiamo trarre analizzando questi dati sono:

- Si può osservare come i modelli addestrati sul problema binario ottengano risultati nettamente migliori rispetto a quelli registrati per il problema multi-classe.
- Si può osservare come sia per il problema binario che per il problema multi-classe non si registrino grandi differenze tra l'addestramento sull'intero dataset e l'addestramento su una sola porzione. Comunque, l'addestramento sull'intero dataset fornisce un'accuratezza lievemente migliore, probabilmente dovuta al fatto che vengono considerate più istanze. Quindi, a parità di tempi computazione, si sceglierà il modello addestrato sull'intero dataset;

Rispetto agli alberi decisionali, le reti neurali hanno ottenuto risultati lievemente migliori. Verranno successivamente analizzati i tempi computazionali relativi ai due modelli per capire quale effettivamente è il migliore per il problema analizzato.

Precision, Recall e F1-Measure

Le matrici di confusione possono essere utilizzate per calcolare alcune misure di performance (ulteriori rispetto all'accuratezza). In particolare:

- Precision. La Precision è una metrica che comunica quanto il modello è affidabile nel classificare correttamente le istanze di una determinata classe. Pertanto esiste una misura Precision per ogni classe. La Precision di un problema binario può essere calcolata come:

$$Precision = \frac{TP}{TP + FP}$$

  I valori di TP e FP sono chiaramente disponibili nella matrice di confusione. In questo quadro la matrice di confusione consente il calcolo della Precision di ogni classe. E questo vale *mutatis mutandis* anche per i problemi multi-classe;

- Recall. La Recall è una metrica che misura la capacità del modello di trovare tutte le istanze di una determinata classe all'interno di un dataset[3]. Infatti la Recall di un problema binario è calcolata come:

$$Recall = \frac{TP}{TP + FN}$$

  Chiaramente anche questa metrica può essere calcolata tramite la matrice di confusione. E questo vale anche per i problemi multi-classe.

- F1-Measure. La metrica F1-Measure per un problema binario è calcolata come:

$$F1\_Measure = 2 \cdot \left(\frac{precision \cdot recall}{precision + recall}\right)$$

  La metrica F1-Measure è la media armonica di Precision e Recall. Il valore di questa metrica risulta alto quando i valori di Precision e Recall del modello per una classe sono simili. Naturalmente questa metrica può essere calcolata anche per problemi multi-classe, richiedendo solo la conoscenza dei valori di Precision e Recall.

[3] GRANDINI, BAGLI, VISANI, *Metrics for Multi-Class Classification: an Overview*, in arXiv:2008.05756 [stat.ML].

1. Alberi decisionali

I valori registrati dagli alberi decisionali per queste misure di performance sono i seguenti:

	DT totale binario	DT split binario
Precision No	0.7281106	0.5289855
Precision Yes	0.6987488	0.709571
Precision Macro Average	0.7134297	0.6192782
Recall No	0.5023847	0.4534161
Recall Yes	0.8601896	0.7678571
Recall Macro Average	0.6812872	0.6106366
F1-Measure No	0.5945437	0.4882943
F1-Measure Yes	0.7711099	0.7375643
F1-Measure Macro Average	0.6828268	0.6129293

Tabella 1 - Misure di performance degli alberi decisionali per il problema binario

	DT totale multi	DT split multi
Precision No Use	0.6457143	0.7619048
Precision Long Term	0.4974874	0.4186047
Precision Short Term	0.4499332	0.4918919
Precision Macro Average	0.531045	0.5574671
Recall No Use	0.5389507	0.3975155
Recall Long Term	0.2972973	0.6605505
Recall Short Term	0.6594912	0.5321637
Recall Macro Average	0.4985797	0.5300766
F1-Measure No Use	0.5875217	0.522449
F1-Measure Long Term	0.3721805	0.5124555
F1-Measure Short Term	0.5349206	0.511236
F1-Measure Macro Average	0.4982076	0.5153802

Tabella 2 - Misure di performance degli alberi decisionali per il problema multi-classe

L'aggregazione dei valori di performance è stata effettuata tramite Macro Average. Infatti non si è ritenuto di dover attribuire maggiore importanza alle classi più numerose. Pertanto non è stata utilizzata l'aggregazione Micro Average[4].

Complessivamente i valori registrati sono mediocri. Tuttavia questi valori di performance dimostrano che:

(i) per il problema binario i modelli trovano più spesso le istanze che appartengono alla classe relativa all'uso del contraccettivo. Infatti per il problema binario il valore di Recall per la classe Yes è molto alto. Questo avviene al prezzo di un maggior numero di falsi positivi per questa classe;

(ii) per il problema multi-classe i modelli hanno difficoltà a classificare correttamente le istanze appartenenti alle classi Long Term e Short Term. Questo è testimoniato dai bassi valori di Precision per queste classi.

Questi dati sono coerenti con le considerazioni fatte analizzando *ictu oculi* le matrici di confusione. E giustificano in parte la diversa accuratezza registrata dai modelli.

[4] In particolare si calcola la Macro Average quando si vuole attribuire ugual peso a tutte le classi. Si calcola invece la Micro Average quando si vuole attribuire un peso maggiore alle classi più numerose.

2. Reti neurali

I valori registrati dalle reti neurali per queste misure di performance sono i seguenti:

	NN totale binario	NN split binario
Precision No	0.7186147	0.619403
Precision Yes	0.7062315	0.7459283
Precision Macro Average	0.7124231	0.6826657
Recall No	0.5278219	0.515528
Recall Yes	0.8459716	0.8178571
Recall Macro Average	0.6868968	0.6666925
F1-Measure No	0.6086159	0.5627119
F1-Measure Yes	0.7698113	0.7802385
F1-Measure Macro Average	0.6892136	0.6714752

Tabella 3 - Misure di performance delle reti neurali per il problema binario

	NN totale multi	NN split multi
Precision No Use	0.6677909	0.5625
Precision Long Term	0.4460641	0.5111111
Precision Short Term	0.5214153	0.4857143
Precision Macro Average	0.5450901	0.5197751
Recall No Use	0.6295707	0.6149068
Recall Long Term	0.4594595	0.4220183
Recall Short Term	0.5479452	0.497076
Recall Macro Average	0.5456585	0.5113337
F1-Measure No Use	0.6481178	0.5875371
F1-Measure Long Term	0.4526627	0.4623116
F1-Measure Short Term	0.5343511	0.4913295
F1-Measure Macro Average	0.5450439	0.513726

Tabella 4 - Misure di performance delle reti neurali per il problema multi-classe

L'aggregazione dei valori di performance è stata effettuata tramite Macro Average, come fatto per gli alberi decisionali. Complessivamente i valori registrati sono molto simili a quelli osservati con gli alberi decisionali. In particolare, per quanto riguarda il problema binario, i risultati ottenuti

sul problema totale sono molto simili, mentre, i risultati ottenuti sul problema splittato risultano migliori con le reti neurali, anche se di poco.

Questi valori di performance dimostrano che:

(i) Per il problema binario i modelli presentano un valore di Recall per lo Yes molto buono, a scapito del valore di Recall per No. Questo vuol dire che il modello identifica bene le istanze che fanno uso del contraccettivo, e peggio quello che non ne fanno uso. Questo potrebbe essere dovuto al fatto che il dataset presenta molte istanze che fanno uso del contraccettivo (Short Term o Long Term), a scapito di quelle che non lo utilizzano;

(ii) Per il problema multi-classe i modelli non riesco a discriminare molto bene le tre classi, in particolare, la classe meglio riconosciuta è quella relativa al non utilizzo del contraccettivo, che presenta il valore più alto sia di Precision che di Recall, anche se non molto buono, seguita dalla classe Short Term, e infine Long Term. Questi risultati, molto simili a quelli analizzati con gli alberi decisionali, potrebbero essere dovuti al fatto che la maggior parte delle istanze non fanno uso del contraccettivo.

Curve ROC e AUC

In questo paragrafo verranno analizzate le curve ROC e i valori AUC dei modelli. Prima di procedere pare opportuno fornire una definizione (almeno intuitiva) di cosa siano le curve ROC e i valori AUC. In particolare:

- La curva ROC mostra la performance di un classificatore binario (in cui la popolazione è divisa tra la classe dei positivi e quella dei negativi). Più precisamente, mostra il valore di TPR (relativo alla frazione di veri positivi di una classe) rispetto al valore di FPR (relativo alla frazione di falsi positivi di una classe) al variare di un tasso di soglia. Il valore di TPR viene anche detto Sensitivity, mentre il valore di FPR può essere anche calcolato come 1 – Specificity;
- L'area sottesa alla curva ROC prende il nome di AUC (acronimo di Area Under Curve). Il valore di AUC può essere interpretato come la probabilità che un'istanza estratta casualmente dalla popolazione dei positivi sia classificata con un valore superiore rispetto a quello ottenuto estraendo casualmente un'istanza dalla popolazione dei negativi.

La rappresentazione delle curve ROC in R è stata ottenuta sfruttando i package ***ROCR*** e ***pROC***. Occorre fare un'ultima considerazione. Si è specificato che le curve ROC sono utilizzate per problemi binari. Tuttavia il package ***multiROC*** in R permette anche la rappresentazione di curve ROC per problemi multi-classe. In questo quadro si è deciso di utilizzare questo package per rappresentare i risultati riguardanti il problema multi-classe.

1. Alberi decisionali

Si presentano ora i risultati ottenuti con gli alberi decisionali. Cominciamo con le curve ROC e i valori AUC dei modelli relativi al problema binario. In questo caso si è considerata la classe relativa all'uso del contraccettivo come classe positiva.

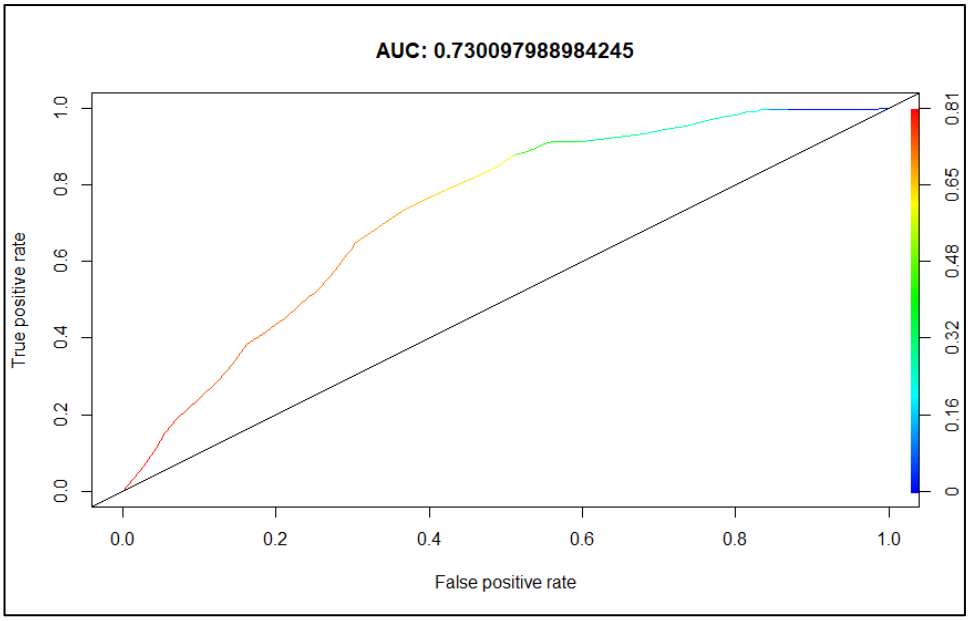


Figura 41 – Curva ROC modello DT totale binario

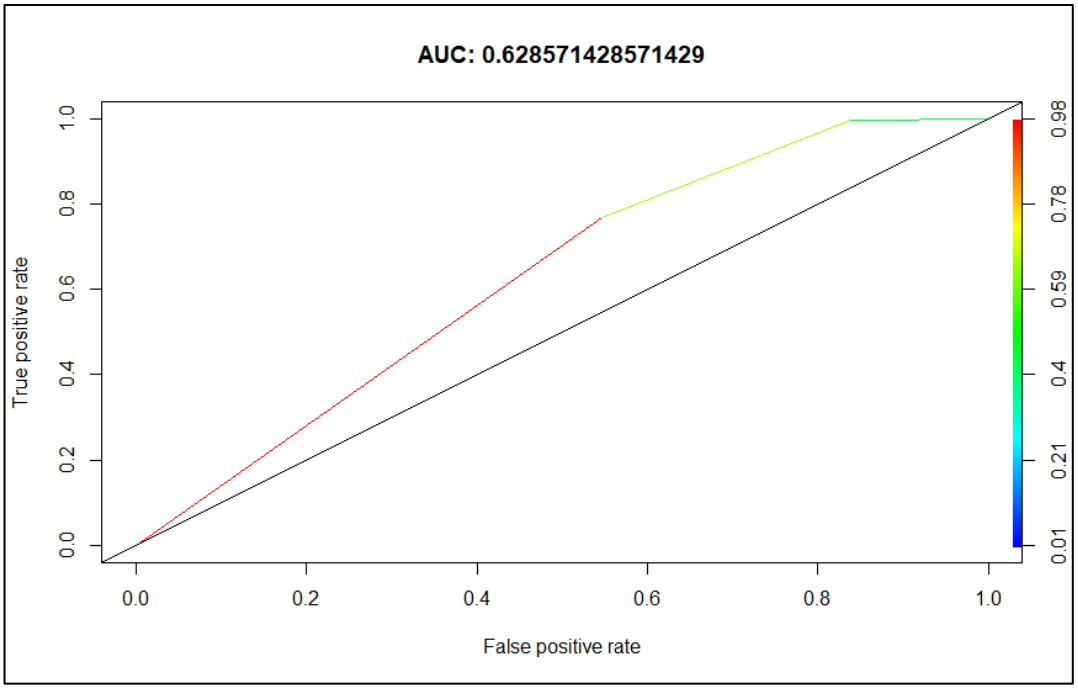


Figura 42 – Curva ROC modello DT split binario

Si può notare come la curva ROC del modello addestrato sull'intero dataset copra un'area maggiore (e sia quindi migliore) rispetto a quella del modello addestrato su una porzione del dataset. In ogni caso le curve ROC risultano migliori del classificatore random rappresentato nei grafici tramite la diagonale. Ma le curve ROC sono ancora lontane dal classificatore ottimale. I valori AUC di queste curve riflettono queste considerazioni.

Passiamo ora alle curve ROC relative al problema multi-classe.

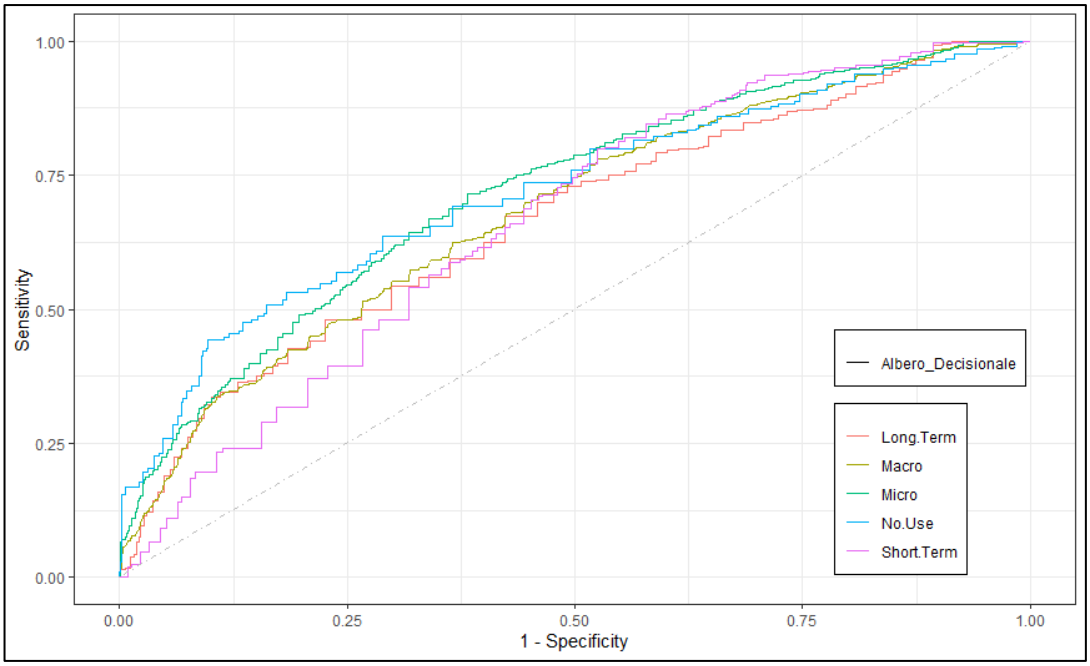


Figura 43 – Curve ROC modello DT totale multi

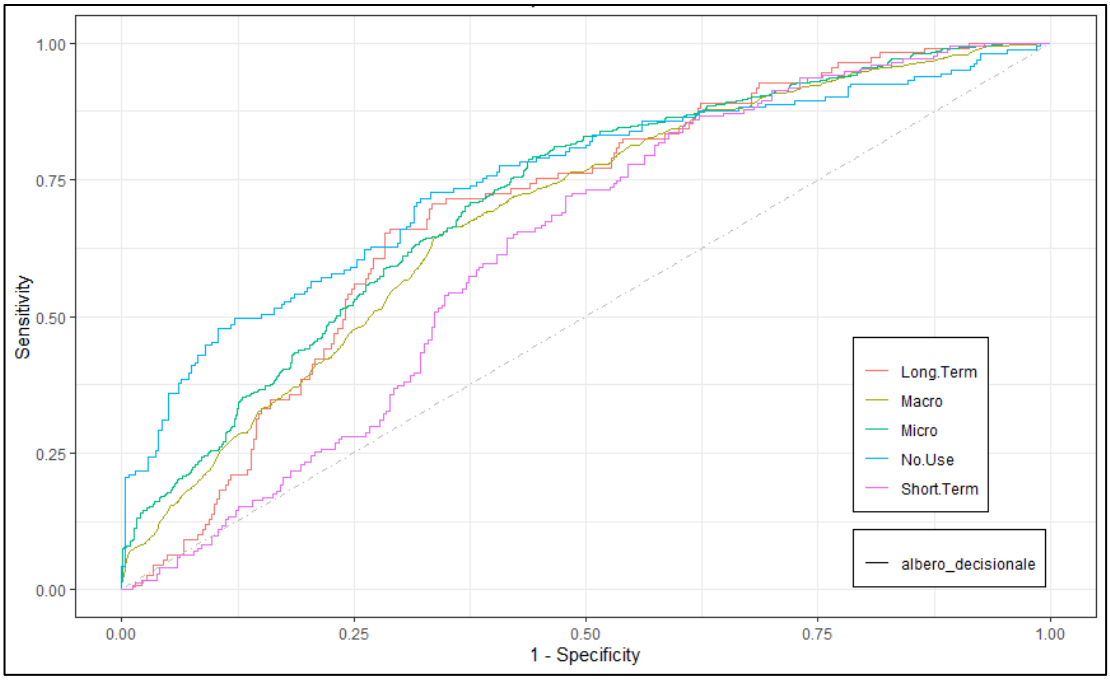


Figura 44 – Curve ROC modello DT split multi

I valori AUC delle curve ROC relative al problema multi-classe sono i seguenti.

	DT totale multi	**DT split multi**
AUC No Use	0.717548	0.7457631
AUC Long Term	0.6654286	0.6941804
AUC Short Term	0.6614156	0.621789
AUC Macro Average	0.6814631	0.6872312
AUC Micro Average	0.7169151	0.7149336

Tabella 3 – Valori AUC delle curve ROC relative agli alberi decisionali per il problema multi-classe

In questo caso le curve ROC relative alle diverse classi risultano molto simili tra loro, fatta eccezione per la curva relativa alla classe Short Term che risulta sensibilmente inferiore. Questo significa che i modelli hanno difficoltà a classificare correttamente le istanze di questa classe. E questo si riflette naturalmente anche nei valori AUC.

2. Reti neurali

Si presentano ora i risultati ottenuti con le reti neurali. Cominciamo con le curve ROC e i valori AUC dei modelli relativi al problema binario. In questo caso si è considerata la classe relativa all'uso del contraccettivo come classe positiva.

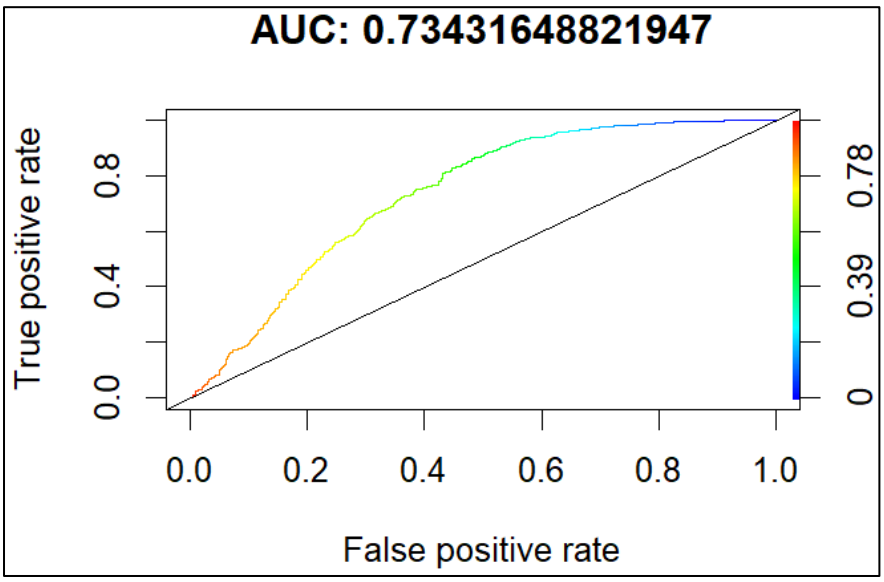


Figura 45 – Curva ROC per NN totale binario

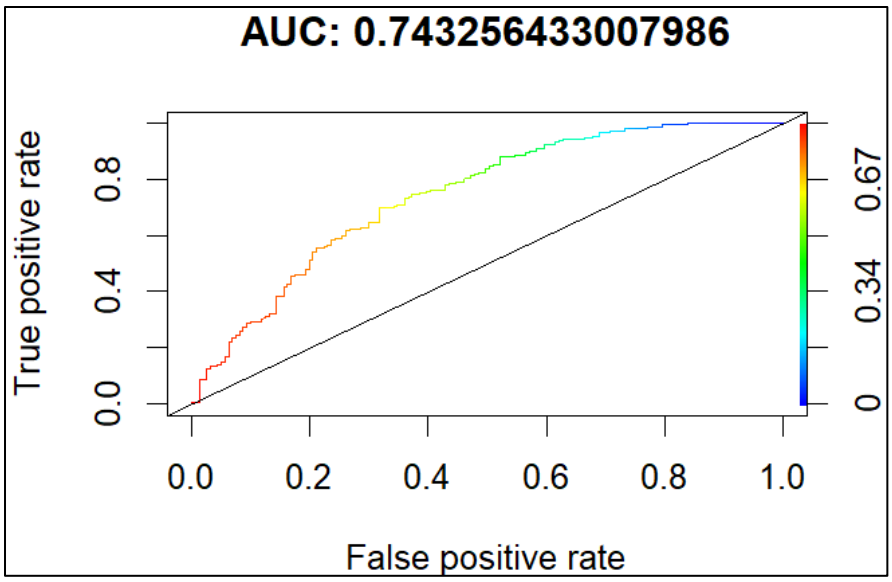


Figura 46 - Curva ROC per NN split binario

Si nota che i risultati ottenuti addestrando il modello sull'intero dataset, piuttosto che addestrandolo solo su una sua porzione con una Cross-Validation ripetuta, sono praticamente identici. In entrambi i casi, il classificatore ottenuto risulta migliore di quello random, ma ben lontano da quello ottimale.

Vediamo ora le curve ROC relative al problema multi-classe:

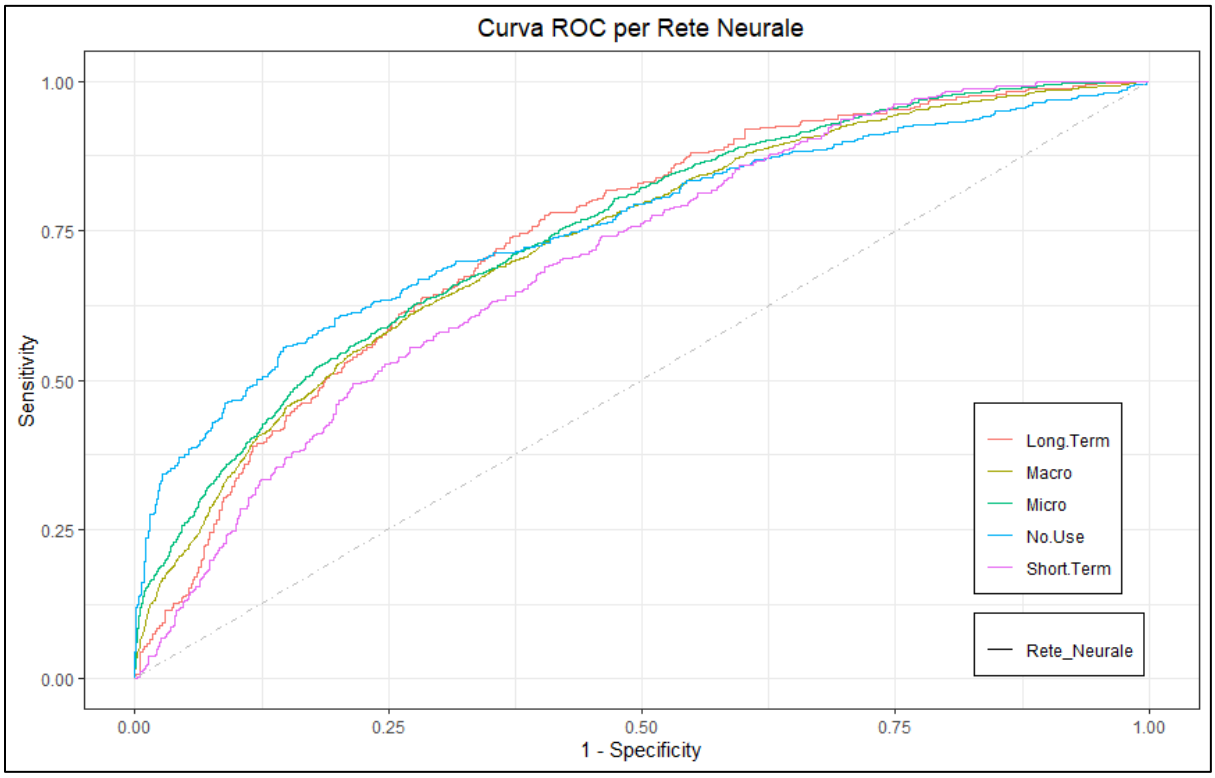


Figura 47 – Curve ROC per NN totale multi-classe

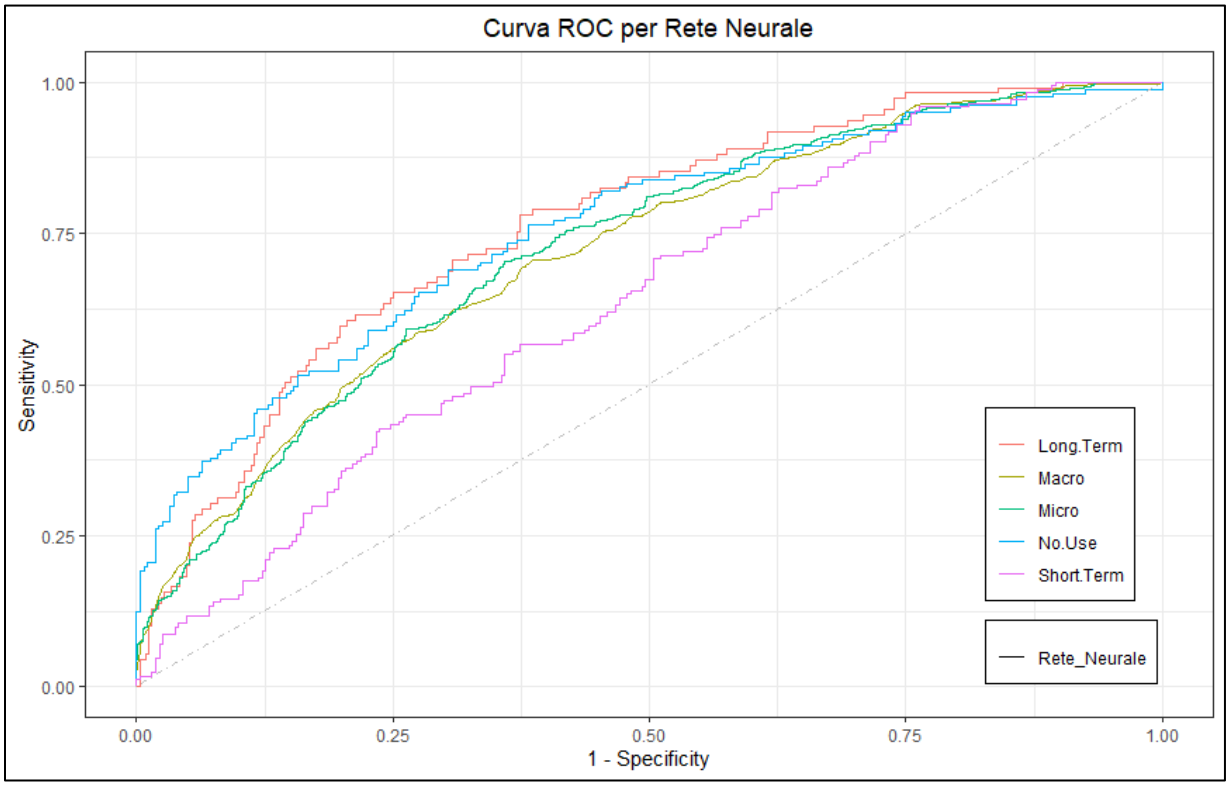


Figura 48 – Curve ROC per NN split multi-classe

I valori AUC delle curve ROC relative al problema multi-classe sono i seguenti.

	NN totale multi	NN split multi
AUC No Use	0.7549032	0.755457
AUC Long Term	0.7409646	0.7622416
AUC Short Term	0.7004345	0.6358241
AUC Macro Average	0.7320958	0.717804
AUC Micro Average	0.7480347	0.7232429

Tabella 5 - Valori AUC delle curve ROC relative alle reti neurali per il problema multi-classe

Si nota che le curve ROC relative alle diverse classi risultano molto simili, soprattutto per quanto visto riguardo al modello addestrato sull'intero dataset. Si nota in ***Figura 48***, però, che il modello addestrato su una parte del dataset presenta una curva ROC (e corrispettivo valore di AUC) più vicino al classificatore random, quindi peggiore: si tratta della curva relativa alla classe Short Term. Comunque, rispetto agli alberi decisionali si sono ottenuti valori in generale migliori.

Tempi di computazione

Infine si riportano i tempi di computazione dei modelli.

1. Alberi decisionali

I tempi di calcolo degli alberi decisionali sono i seguenti.

	Everything (s)	Final (s)	Prediction (s)
DT totale binario	0.76	0.01	NA
DT split binario	0.99	0.02	NA
DT totale multi	1.02	0.02	NA
DT split multi	2.09	0.04	NA

Tabella 6 - Tempi di computazione degli alberi decisionali

I dati riportati mostrano come i modelli richiedano tempi di computazione leggermente maggiori per il problema multi-classe rispetto al problema binario. Al di là di questa leggera differenza, i tempi di computazione risultano sostanzialmente equipollenti.

2. Reti neurali

I tempi di calcolo delle reti neurali sono i seguenti.

	Everything (s)	Final (s)	Prediction (s)
NN totale binario	8.76	0.17	NA
NN split binario	18.69	0.11	NA
NN totale multi	14.89	0.59	NA
NN split multi	28.71	0.22	NA

Tabella 7 - Tempi di computazione delle reti neurali

Si noti come i tempi di addestramento siano maggiori per quanto riguarda il problema multi-classe rispetto a quello binario. Inoltre, i tempi di addestramento dei modelli definiti sullo split del dataset sono molto maggiori dei corrispettivi definiti sull'intero dataset. Questo è dovuto al fatto che, per questi modelli, è stata utilizzata una Cross-Validation ripetuta tre volte.

Confronto tra alberi decisionali e reti neurali

Sin qui sono stati presentati gli esperimenti eseguiti sui modelli di alberi decisionali e reti neurali addestrati. Si procede ora a un confronto tra alberi decisionali e reti neurali. Per questo confronto verranno considerati solo i modelli addestrati sull'intero dataset. Infatti si è già riscontrato come le differenze tra i modelli addestrati sull'intero dataset e quelli addestrati sul 70% del dataset siano minime e comunque a favore dei primi. Il confronto si articolerà nei seguenti punti: (*i*) le principali misure di performance; (*ii*) le curve ROC e i valori AUC; e (*iii*) i tempi di computazione. Le conclusioni verranno invece rassegnate nel prossimo capitolo.

1. Principali misure di performance

Cominciamo con le principali misure di performance. Il confronto è mostrato tramite la ***Tabella 8*** e la ***Tabella 9*** riguardanti rispettivamente le misure di performance sul problema binario e sul problema multi-classe.

	DT binario	NN binario
Accuracy	0.7074 ± 0.0231	0.7101 ± 0.0231
Precision No	0.7281106	0.7186147
Precision Yes	0.6987488	0.7062315
Precision Macro Average	0.7134297	0.7124231
Recall No	0.5023847	0.5278219
Recall Yes	0.8601896	0.8459716
Recall Macro Average	0.6812872	0.6868968
F1-Measure No	0.5945437	0.6086159
F1-Measure Yes	0.7711099	0.7698113

F1-Measure Macro Average	0.6828268	0.6892136

Tabella 8 - Confronto delle performance tra albero decisionale e rete neurale sul problema binario

	DT multi-classe	**NN multi-classe**
Accuracy	0.5261 ± 0.0258	0.5628 ± 0.0255
Precision No Use	0.6457143	0.6677909
Precision Long Term	0.4974874	0.4460641
Precision Short Term	0.4499332	0.5214153
Precision Macro Average	0.531045	0.5450901
Recall No Use	0.5389507	0.6295707
Recall Long Term	0.2972973	0.4594595
Recall Short Term	0.6594912	0.5479452
Recall Macro Average	0.4985797	0.5456585
F1-Measure No Use	0.5875217	0.6481178
F1-Measure Long Term	0.3721805	0.4526627
F1-Measure Short Term	0.5349206	0.5343511
F1-Measure Macro Average	0.4982076	0.5450439

Tabella 9 - Confronto delle performance tra albero decisionale e rete neurale sul problema multi-classe

Già si è evidenziato come le performance di questi modelli non siano particolarmente buone. Tuttavia i dati registrati rivelano come le reti neurali raggiungano misure di performance leggermente migliori rispetto agli alberi decisionali. Questo vantaggio delle reti neurali è quasi impercettibile sul problema binario, mentre diventa leggermente più significativo sul problema multi-classe. Infatti si può notare come le performance delle reti neurali sulle classi di minoranza (Long Term e Short Term) siano complessivamente migliori rispetto a quelle raggiunte con gli alberi decisionali. E ciò fornisce un contributo significativo al vantaggio complessivo della performance delle reti neurali rispetto agli alberi decisionali.

2. Curve ROC e valori AUC

Vediamo ora le curve ROC e i valori AUC. In particolare:

1) Curve ROC relative alle classi del problema binario.

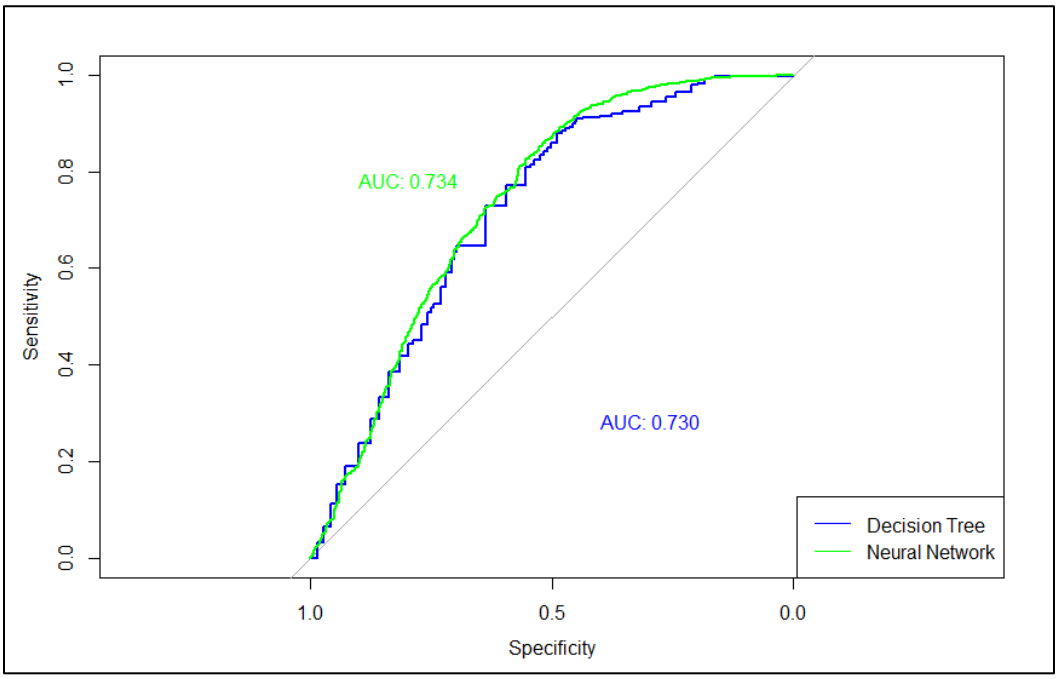


Figura 49 – Confronto curve ROC classe Yes del problema binario

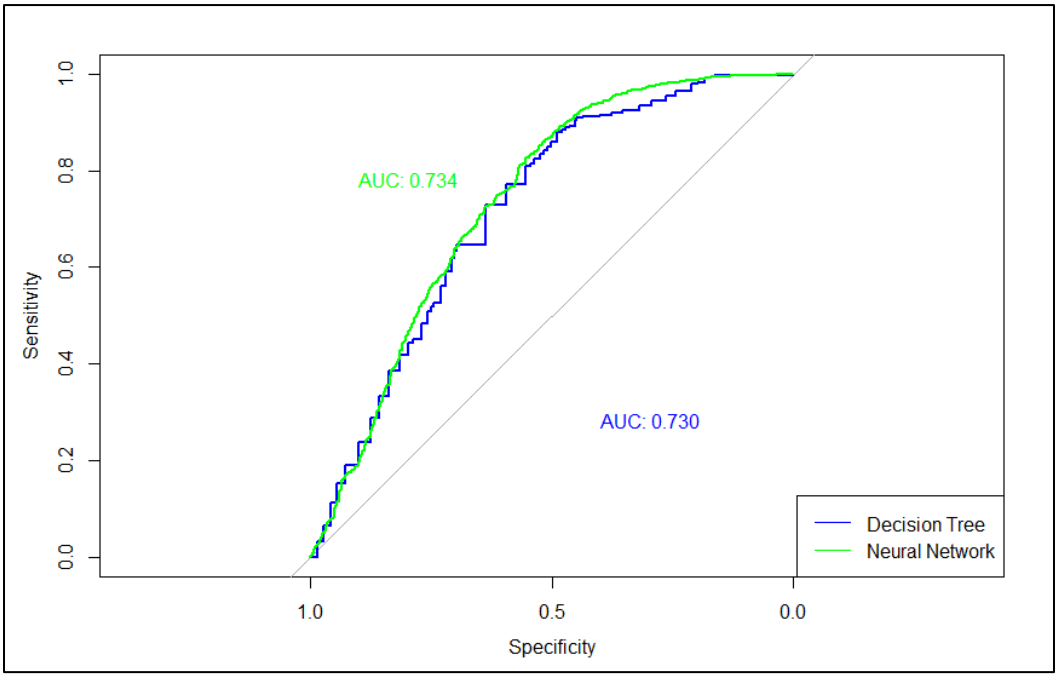


Figura 50 – Confronto curve ROC classe No del problema binario

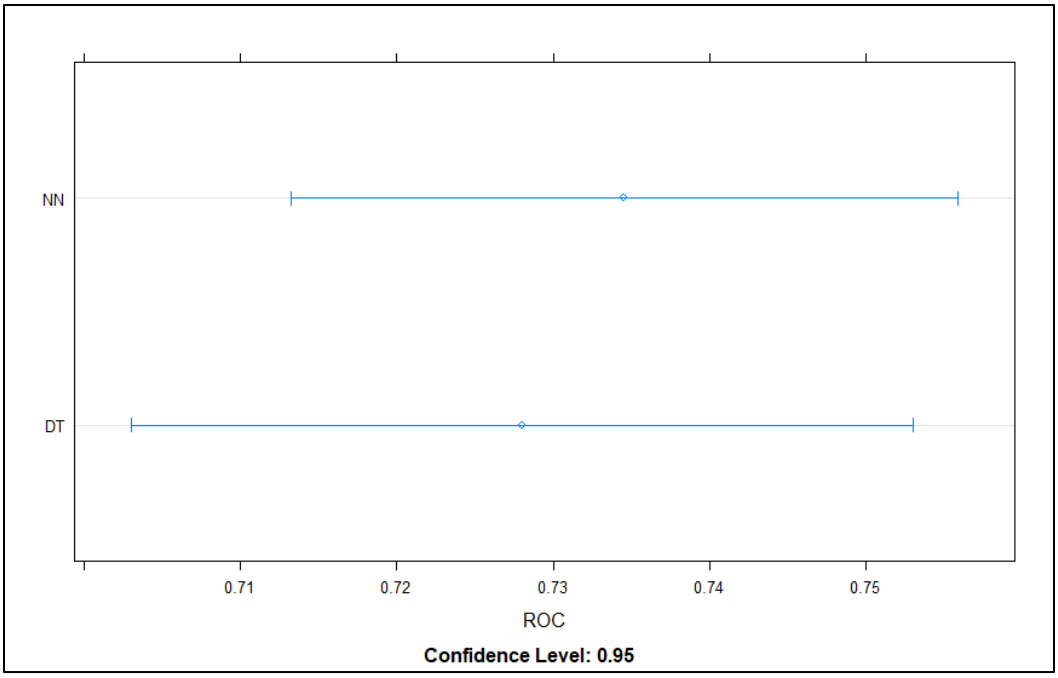


Figura 51 – Confronto intervalli di confidenza delle curve ROC del problema binario

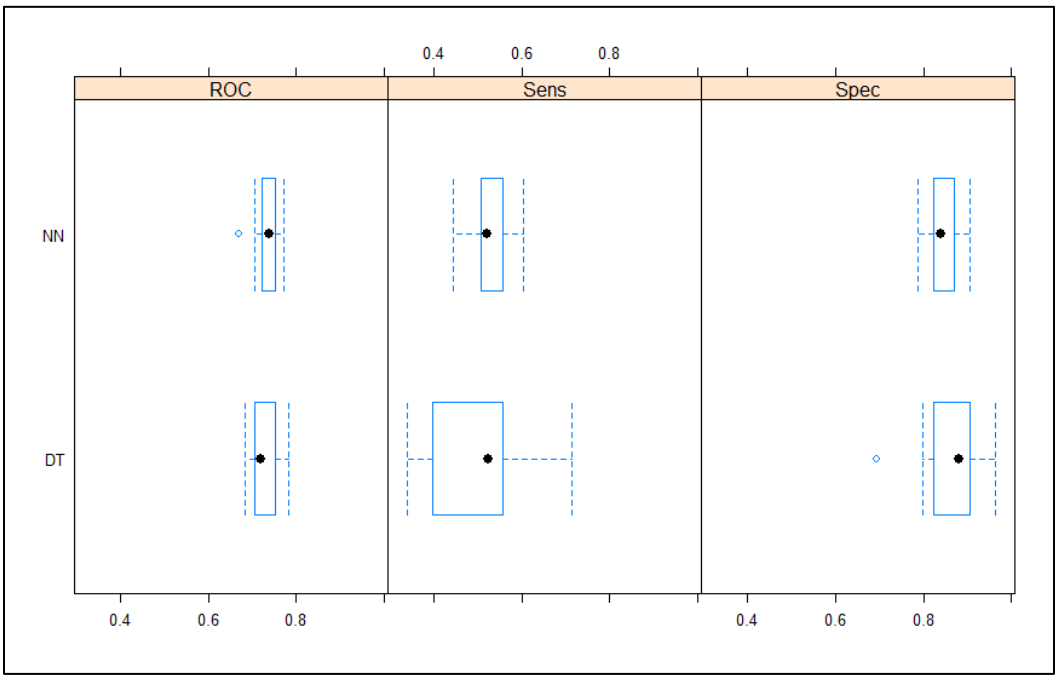


Figura 52 – Confronto ROC, Sensitivity e Specificity del problema binario

2) Curve ROC relative alle classi e alle medie del problema multi-classe.

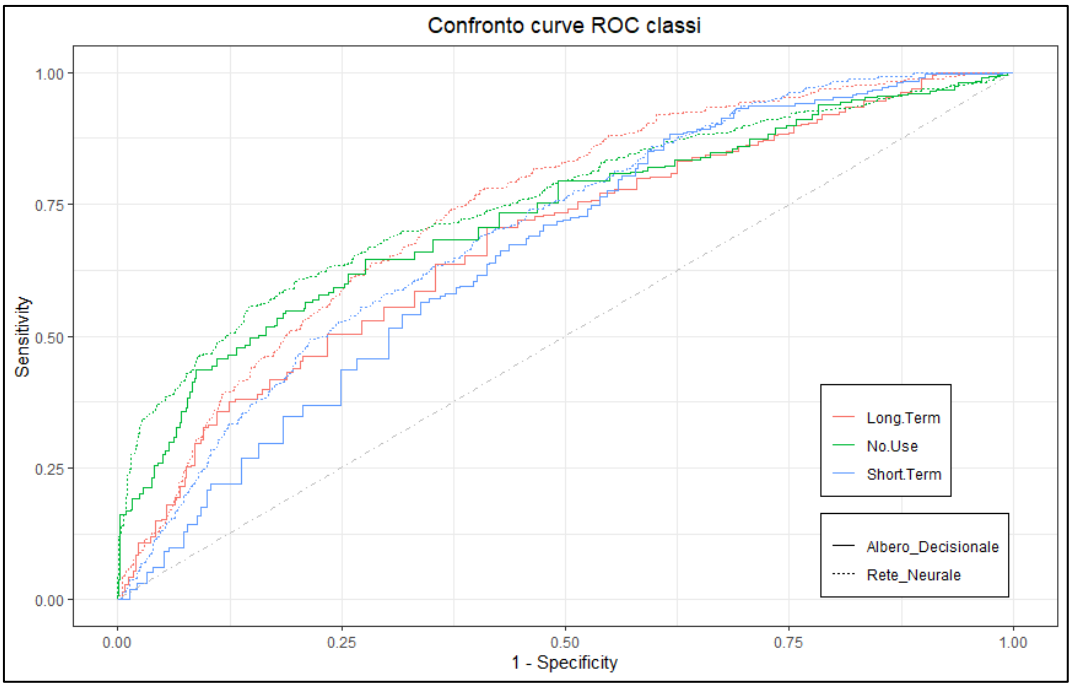


Figura 53 – Confronto curve ROC delle classi del problema multi-classe

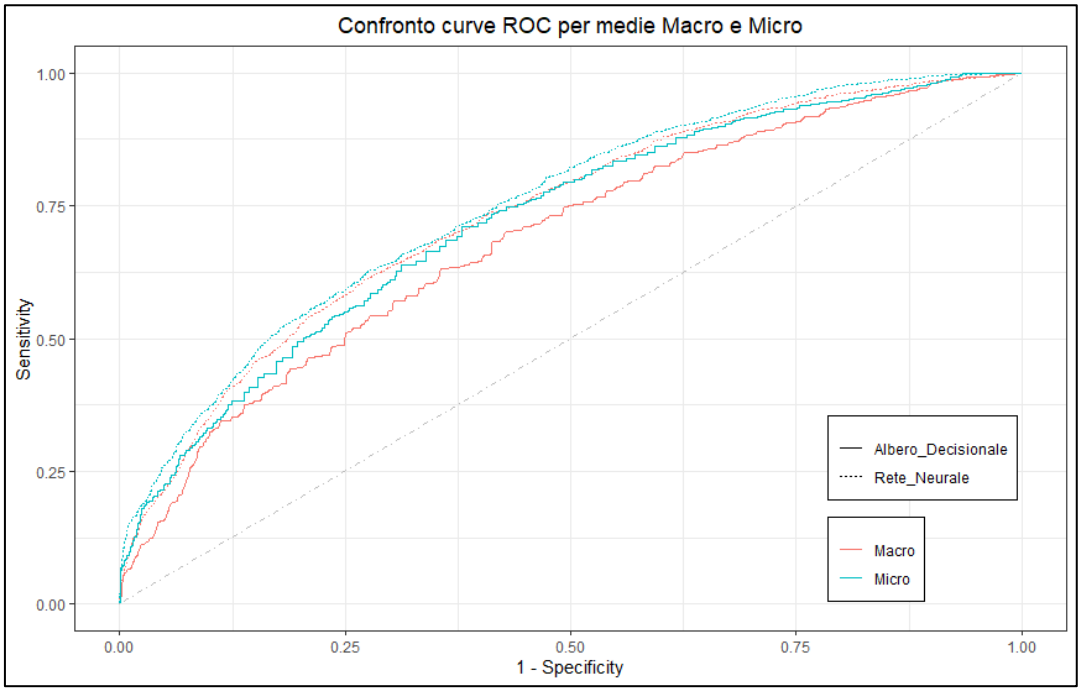


Figura 54 – Confronto curve ROC delle medie Macro e Micro del problema multi-classe

3) Riepilogo valori AUC.

	DT binario	NN binario
AUC No	0.7301	0.7343
AUC Yes	0.7301	0.7343

Tabella 7 – Confronto valori AUC delle curve ROC per il problema binario

	DT multi-classe	NN multi-classe
AUC No Use	0.717548	0.7549032
AUC Long Term	0.6654286	0.7409646
AUC Short Term	0.6614156	0.7004345
AUC Macro Average	0.6814631	0.7320958
AUC Micro Average	0.7169151	0.7480347

Tabella 8 – Confronto valori AUC delle curve ROC per il problema multi-classe

I grafici e le tabelle sopra riportate confermano la migliore efficacia delle reti neurali rispetto agli alberi decisionali. Infatti:

- Il valore AUC raggiunto dalle reti neurali è complessivamente maggiore rispetto a quello raggiunto dagli alberi decisionali. E ciò sia per il problema binario che per il problema multi-classe.
- Gli intervalli di confidenza del valore ROC dei modelli sul problema binario sono parzialmente sovrapposti. Tuttavia la sovrapposizione intercorre fra i valori superiori dell'intervallo dell'albero decisionale e i valori inferiori dell'intervallo della rete neurale. Pertanto anche questo dato conferma un leggero vantaggio per le reti neurali.
- Gli intervalli di confidenza delle reti neurali per i valori ROC, Sensitivity e Specificity sono più compatti rispetto a quelli degli alberi decisionali. Questo comporta una minore variabilità delle performance delle reti neurali. Pertanto anche questo dato favorisce le reti neurali.
- In merito al problema multi-classe le reti neurali presentano curve ROC decisamente migliori rispetto agli alberi decisionali. In particolare, per le reti neurali solo la classe Short Term risulta sensibilmente inferiore alle altre, mentre per gli alberi decisionali entrambe le classi di minoranza (Short Term e Long Term) presentano curve sensibilmente inferiori alla classe No Use. E questa differenza risulta significativa nel calcolo delle medie Macro e Micro.

In sintesi: le reti neurali conseguono performance migliori e più stabili rispetto agli alberi decisionali.

3. Tempi di computazione

Infine è possibile confrontare anche i tempi di computazione dei modelli. I risultati sono mostrati nella ***Tabella 9***.

	Everything (s)	Final (s)	Prediction (s)
DT binario	0.76	0.01	NA
NN binario	8.76	0.17	NA
DT multi-classe	1.02	0.02	NA
NN multi-classe	14.89	0.59	NA

Tabella 9 – Confronto dei tempi di computazione tra albero decisionale e rete neurale

La differenza tra i modelli in merito ai tempi di computazione è netta. In particolare, le reti neurali registrano tempi di calcolo superiori rispetto a quelli degli alberi decisionali di un fattore maggiore di 10. Questo dato non può essere trascurato, soprattutto se i modelli vengono impiegati all'interno di sistemi per i quali il tempo di computazione è critico.

Conclusioni

Il presente lavoro ha avuto per oggetto la predizione dell'uso o meno di contraccettivi (ed eventualmente di quale tipo) da parte di donne sposate sulla base di un dataset indonesiano del 1987. Per raggiungere questo obiettivo è stata effettuata un'analisi esplorativa del dataset che ha portato a scartare la sua trasformazione operata tramite PCA. I modelli di Machine Learning utilizzati per la predizione sono stati gli alberi decisionali di tipo CART e le reti neurali.

Per quanto riguarda il confronto tra i modelli adottati, gli esperimenti condotti hanno portato alle seguenti conclusioni. Le reti neurali hanno registrato performance predittive superiori rispetto agli alberi decisionali. Questo vantaggio è risultato lieve per il problema binario e più significativo per il problema multi-classe. D'altro canto gli alberi decisionali hanno beneficiato di tempi di computazione decisamente migliori rispetto alle reti neurali.

In ogni caso occorre precisare che le performance predittive dei modelli addestrati sono risultate complessivamente modeste. In particolare, l'accuratezza dei modelli non ha superato il valore di 0.75; e i valori AUC sono rimasti inferiori al valore di 0.8. Tuttavia questi risultati non sorprendono. Infatti nell'ambito della ricerca scientifica il presente problema è stato considerato difficile e tale da generare un "tasso di errore minimo maggiore di 0.4"[5].

I motivi che rendono difficile il problema oggetto di questo lavoro sono diversi. Se ne citano alcuni. Anzitutto i dati raccolti sono sbilanciati rispetto alle classi da predire. Questo ha reso più difficile la corretta identificazione delle classi meno popolate. Inoltre i dati raccolti per alcune classi sono esigui e anche questo ha contribuito a renderne più difficile l'identificazione. Questi

[5] LIM, LOH, SHIH, *A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms*, in *Machine Learning*, 40, 2000, pp. 203-228.

difetti sono stati solo parzialmente attenuati dalla trasformazione del problema originario da multi-classe a binario.

Si possono ipotizzare alcuni miglioramenti che si potrebbero apportare al dataset al fine di migliorare le predizioni dei modelli. In particolare, il dataset potrebbe essere arricchito da ulteriori informazioni, sia quantitative che qualitative. Tra le informazioni quantitative si possono citare: il reddito delle coppie, in luogo del generico riferimento al livello di occupazione; e il costo dell'accesso ai contraccettivi. Tra le informazioni qualitative si possono menzionare: l'accessibilità all'assistenza sanitaria; e l'accessibilità ai contraccettivi. Inoltre sarebbe naturalmente opportuno raccogliere una maggiore quantità di dati rispetto alle 1473 istanze del dataset utilizzato.

Approfondimento A: confronto dei modelli con PCA e senza PCA

Nel paragrafo relativo alla PCA si è specificato di aver preferito il dataset originale rispetto a quello trasformato con la PCA. Questa scelta è stata dettata dal fatto che i modelli addestrati sul dataset originale hanno registrato performance migliori rispetto a quelle dei modelli addestrati sul dataset trasformato con la PCA. In questo approfondimento verranno mostrate e confrontate sinteticamente le performance registrate.

Accuratezza

Anzitutto si procede a confrontare l'accuratezza registrata dai modelli.

1) Accuratezza degli alberi decisionali:

	DT totale binario (no PCA)	**DT totale binario (con PCA)**	**DT split binario (no PCA)**	**DT split binario (con PCA)**
Accuratezza	0.7074 ± 0.0231	0.6280 ± 0.0247	0.6531 ± 0.0444	0.6644 ± 0.0440

Tabella 10 – Confronto accuratezza alberi decisionali con PCA e senza sul problema binario

	DT totale multi (no PCA)	**DT totale multi (con PCA)**	**DT split multi (no PCA)**	**DT split multi (con PCA)**
Accuratezza	0.5261 ± 0.0258	0.4718 ± 0.0259	0.5147 ± 0.0476	0.4671 ± 0.0478

Tabella 11 - Confronto accuratezza alberi decisionali con PCA e senza sul problema multi-classe

2) Accuratezza delle reti neurali:

	NN totale binario (no PCA)	**NN totale binario (con PCA)**	**NN split binario (no PCA)**	**NN split binario (con PCA)**
Accuratezza	0.7101 ± 0.0231	0.6327 ± 0.0247	0.7075 ± 0.0421	0.6553 ± 0.0443

Tabella 12 – Confronto accuratezza alberi decisionali con PCA e senza sul problema binario

	NN totale multi (no PCA)	**NN totale multi (con PCA)**	**NN split multi (no PCA)**	**NN split multi (con PCA)**
Accuratezza	0.5628 ± 0.0255	0.5037 ± 0.0259	0.5215 ± 0.0475	0.4762 ± 0.0478

Tabella 13 - Confronto accuratezza alberi decisionali con PCA e senza sul problema multi-classe

Gli intervalli di confidenza sono stati calcolati con una confidenza pari al 95%.

Le tabelle riportate mostrano che l'accuratezza registrata dai modelli addestrati sul dataset trasformato con la PCA è quasi sempre inferiore rispetto a quella registrata dai modelli addestrati sul dataset originale. E questa conclusione vale anche tenendo in considerazione gli intervalli di confidenza.

Precision, Recall e F1-Measure

Si procede ora a mostrare i risultati dei modelli in termini di Precision, Recall e F1-Measure.

1) Confronto relativo agli alberi decisionali addestrati sull'intero dataset relativamente al problema binario:

	DT totale binario (no PCA)	DT totale binario (con PCA)
Precision No	0.7281106	0.5985401
Precision Yes	0.6987488	0.6393597
Precision Macro Average	0.7134297	0.6189499
Recall No	0.5023847	0.391097
Recall Yes	0.8601896	0.8045024
Recall Macro Average	0.6812872	0.5977997
F1-Measure No	0.5945437	0.4730769
F1-Measure Yes	0.7711099	0.7124869
F1-Measure Macro Average	0.6828268	0.5927819

Tabella 14 – Confronto performance alberi decisionali con PCA e senza sul dataset intero per il problema binario

2) Confronto relativo agli alberi decisionali addestrati sul dataset diviso relativamente al problema binario:

	DT split binario (no PCA)	DT split binario (con PCA)
Precision No	0.5289855	0.5565217
Precision Yes	0.709571	0.702454
Precision Macro Average	0.6192782	0.6294879
Recall No	0.4534161	0.3975155
Recall Yes	0.7678571	0.8178571
Recall Macro Average	0.6106366	0.6076863
F1-Measure No	0.4882943	0.4637681
F1-Measure Yes	0.7375643	0.7557756
F1-Measure Macro Average	0.6129293	0.6097718

Tabella 15 – Confronto performance alberi decisionali con PCA e senza sul dataset diviso per il problema binario

3) Confronto relativo agli alberi decisionali addestrati sull'intero dataset relativamente al problema multi-classe:

	DT totale multi (no PCA)	DT totale multi (con PCA)
Precision No Use	0.6457143	0.5345081
Precision Long Term	0.4974874	0.4857143
Precision Short Term	0.4499332	0.3934708
Precision Macro Average	0.531045	0.4712311
Recall No Use	0.5389507	0.5786963
Recall Long Term	0.2972973	0.3063063
Recall Short Term	0.6594912	0.4481409
Recall Macro Average	0.4985797	0.4443812
F1-Measure No Use	0.5875217	0.5557252
F1-Measure Long Term	0.3721805	0.3756906
F1-Measure Short Term	0.5349206	0.4190302
F1-Measure Macro Average	0.4982076	0.4501487

Tabella 16 – Confronto performance alberi decisionali con PCA e senza sul dataset intero per il problema multi-classe

4) Confronto relativo agli alberi decisionali addestrati sul dataset diviso relativamente al problema multi-classe:

	DT split multi (no PCA)	DT split multi (con PCA)
Precision No Use	0.7619048	0.459144
Precision Long Term	0.4186047	0.5584416
Precision Short Term	0.4918919	0.4205607
Precision Macro Average	0.5574671	0.4793821
Recall No Use	0.3975155	0.7329193
Recall Long Term	0.6605505	0.3944954
Recall Short Term	0.5321637	0.2631579
Recall Macro Average	0.5300766	0.4635242
F1-Measure No Use	0.522449	0.5645933
F1-Measure Long Term	0.5124555	0.4623656
F1-Measure Short Term	0.511236	0.323741
F1-Measure Macro Average	0.5153802	0.4502333

Tabella 17 – Confronto performance alberi decisionali con PCA e senza sul dataset diviso per il problema multi-classe

5) Confronto relativo alle reti neurali addestrate sull'intero dataset relativamente al problema binario:

	NN totale binario (no PCA)	NN totale binario (con PCA)
Precision No	0.7186147	0.5956522
Precision Yes	0.7062315	0.6495558
Precision Macro Average	0.7124231	0.622604
Recall No	0.5278219	0.4356121
Recall Yes	0.8459716	0.7796209
Recall Macro Average	0.6868968	0.6076165
F1-Measure No	0.6086159	0.503214
F1-Measure Yes	0.7698113	0.7086699
F1-Measure Macro Average	0.6892136	0.6059419

Tabella 18 - Confronto performance reti neurali con PCA e senza sul dataset intero per il problema binario

6) Confronto relativo alle reti neurali addestrate sul dataset diviso relativamente al problema binario:

	NN split binario (no PCA)	NN split binario (con PCA)
Precision No	0.619403	0.5319149
Precision Yes	0.7459283	0.7133333
Precision Macro Average	0.6826657	0.6226241
Recall No	0.515528	0.4658385
Recall Yes	0.8178571	0.7642857
Recall Macro Average	0.6666925	0.6150621
F1-Measure No	0.5627119	0.4966887
F1-Measure Yes	0.7802385	0.737931
F1-Measure Macro Average	0.6714752	0.6173099

Tabella 19 - Confronto performance reti neurali con PCA e senza sul dataset diviso per il problema binario

7) Confronto relativo alle reti neurali addestrate sull'intero dataset relativamente al problema multi-classe:

	NN totale multi (no PCA)	NN totale multi (con PCA)
Precision No Use	0.6677909	0.5566434
Precision Long Term	0.4460641	0.4166667
Precision Short Term	0.5214153	0.4781659
Precision Macro Average	0.5450901	0.4838253
Recall No Use	0.6295707	0.6327504
Recall Long Term	0.4594595	0.3753754
Recall Short Term	0.5479452	0.4285714
Recall Macro Average	0.5456585	0.4788991
F1-Measure No Use	0.6481178	0.5922619
F1-Measure Long Term	0.4526627	0.3949447
F1-Measure Short Term	0.5343511	0.4520124
F1-Measure Macro Average	0.5450439	0.4797397

Tabella 20 - Confronto performance reti neurali con PCA e senza sul dataset intero per il problema multi-classe

8) Confronto relativo alle reti neurali addestrate sul dataset diviso relativamente al problema multi-classe:

	NN split multi (no PCA)	NN split multi (con PCA)
Precision No Use	0.5625	0.4561404
Precision Long Term	0.5111111	0.5373134
Precision Short Term	0.4857143	0.4794521
Precision Macro Average	0.5197751	0.4909686
Recall No Use	0.6149068	0.6459627
Recall Long Term	0.4220183	0.3302752
Recall Short Term	0.497076	0.4093567
Recall Macro Average	0.5113337	0.4618649
F1-Measure No Use	0.5875371	0.5347044
F1-Measure Long Term	0.4623116	0.4090909
F1-Measure Short Term	0.4913295	0.4416404
F1-Measure Macro Average	0.513726	0.4797397

Tabella 21 - Confronto performance reti neurali con PCA e senza sul dataset diviso per il problema multi-classe

Dai dati riportati in queste tabelle emerge chiaramente che i modelli addestrati sul dataset originale ottengono quasi sempre valori di Precision, Recall e F1-Measure superiori rispetto ai valori ottenuti dai modelli addestrati sul dataset trasformato con PCA. E questo vale sia per gli alberi decisionali che per le reti neurali.

Valori AUC

Passiamo ora ai valori AUC delle curve ROC.

1) Valori AUC delle curve ROC sul problema binario (considerando la classe Yes):

	DT totale binario (no PCA)	**DT totale binario (con PCA)**	**DT split binario (no PCA)**	**DT split binario (con PCA)**
AUC	0.7301	0.6235	0.6286	0.6485

Tabella 22 – Confronto AUC alberi decisionali con PCA e senza PCA sul problema binario

	NN totale binario (no PCA)	**NN totale binario (con PCA)**	**NN split binario (no PCA)**	**NN split binario (con PCA)**
AUC	0.7343	0.6483	0.7433	0.6553

Tabella 23 - Confronto AUC reti neurali con PCA e senza PCA sul problema binario

2) Valori AUC delle curve ROC sul problema multi-classe:

	DT totale multi (no PCA)	**DT totale multi (con PCA)**	**DT split multi (no PCA)**	**DT split multi (con PCA)**
AUC No Use	0.717548	0.6490103	0.7457631	0.6545031
AUC Long Term	0.6654286	0.6730309	0.6941804	0.7079971
AUC Short Term	0.6614156	0.5579151	0.621789	0.50483
AUC Macro Average	0.6814631	0.626652	0.6872312	0.6224416
AUC Micro Average	0.7169151	0.6689667	0.7149336	0.6644171

Tabella 24 - Confronto AUC alberi decisionali con PCA e senza PCA sul problema multi-classe

	NN totale multi (no PCA)	**NN totale multi (con PCA)**	**NN split multi (no PCA)**	**NN split multi (con PCA)**
AUC No Use	0.7549032	0.6570819	0.755457	0.6631988
AUC Long Term	0.7409646	0.7052948	0.7622416	0.7354924
AUC Short Term	0.7004345	0.6124187	0.6358241	0.5916613
AUC Macro Average	0.7320958	0.6582649	0.717804	0.6634527
AUC Micro Average	0.7480347	0.6813222	0.7232429	0.6641163

Tabella 25 - Confronto AUC reti neurali con PCA e senza PCA sul problema multi-classe

I valori AUC delle curve ROC confermano le migliori performance dei modelli addestrati sul dataset originale.

Tempi di computazione

Infine si riportano i tempi di computazione registrati dai modelli.

	Everything (s)	Final (s)	Prediction (s)
DT totale binario (no PCA)	0.76	0.01	NA
DT totale binario (con PCA)	0.58	0.01	NA
DT split binario (no PCA)	0.99	0.02	NA
DT split binario (con PCA)	1.05	0.02	NA
DT totale multi (no PCA)	1.02	0.02	NA
DT totale multi (con PCA)	0.99	0.02	NA
DT split multi (no PCA)	2.09	0.04	NA
DT split multi (con PCA)	0.91	0.02	NA
NN totale binario (no PCA)	8.76	0.17	NA
NN totale binario (con PCA)	10.45	0.11	NA
NN split binario (no PCA)	18.69	0.11	NA
NN split binario (con PCA)	20.30	0.09	NA
NN totale multi (no PCA)	14.89	0.59	NA
NN totale multi (con PCA)	14.80	0.16	NA
NN split multi (no PCA)	28.71	0.22	NA
NN split multi (con PCA)	32.83	0.16	NA

Tabella 26 – Confronto tempi di computazione modelli con PCA e senza PCA

I dati riportati consentono di affermare come non vi siano sostanziali differenze tra i tempi di computazione dei modelli addestrati sul dataset originale e quelli dei modelli addestrati sul dataset trasformato con la PCA.

Sintesi

In sintesi: i modelli addestrati sul dataset originale hanno registrato performance nettamente migliori rispetto ai modelli addestrati sul dataset trasformato con la PCA; non vi sono invece sostanziali differenze a livello di tempi computazionali.