

FACOLTA’ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL’INFORMAZIONE

APPRENDIMENTO AUTOMATICO

TESI DI LAUREA

**ANALISI DEL DATASET MAMMOGRAPHIC MASSES**

Candidato: Cosimo Strusi

A.A. 2022/2023

Indice

[1 Introduzione 3](#_Toc114437249)

[1.1 Cancro al seno 3](#_Toc114437250)

[1.2 Problema 3](#_Toc114437251)

[1.3 Scopo elaborato 4](#_Toc114437252)

[2 Dataset 5](#_Toc114437253)

[2.1 Descrizione dataset 5](#_Toc114437254)

[2.2 Descrizione attributi 5](#_Toc114437255)

[2.3 Dati mancanti e pulizia dati 9](#_Toc114437256)

[2.3.1 Pulizia dati 9](#_Toc114437257)

[2.3.2 Eliminazione dell’unità statistica (primo approccio) 10](#_Toc114437258)

[2.3.3 Imputazione attraverso KNN (secondo approccio) 14](#_Toc114437259)

[2.4 Datasets finali 18](#_Toc114437260)

[3 Visualizzazione in 2 dimensioni dei dati 20](#_Toc114437261)

[4.1 Calcolo Decision Boundary con Regressione Logistica 22](#_Toc114437262)

[4.2 Calcolo Decision Boundary con Random Forest 24](#_Toc114437263)

[4.3 Commento sulla rappresentazione in due dimensioni 25](#_Toc114437264)

[5 Allenamento modelli 26](#_Toc114437265)

[5.1 Scelta classificatori 26](#_Toc114437266)

[5.2 Hold-out method 27](#_Toc114437267)

[5.3 K-FOLD Validazione 28](#_Toc114437268)

[5.4 Confronto classificatori 30](#_Toc114437269)

[6 Conclusione 32](#_Toc114437270)

# 1 Introduzione

## 1.1 Cancro al seno

Tra molti tumori, il cancro al seno è la seconda causa di morte più comune nelle donne. Secondo l'American Cancer Society, più di 40mila donne sono morte di cancro al seno e una donna su tre, fra quelle con il cancro, hanno il cancro al seno. La diagnosi e il trattamento precoce riducono la mortalità, in questo ambito la mammografia è il metodo più efficace disponibile oggi per lo screening del cancro al seno.

## 1.2 Problema

La mammografia è la tecnica più efficace e sicura in quanto supera i danni dell'esposizione alle radiazioni. Tuttavia, il basso valore predittivo positivo risultante dall'interpretazione della mammografia porta a circa il 70% di biopsie non necessarie con esiti benigni, che non solo causa stress fisico e mentale per i pazienti, ma anche un danno di tipo economico. Quindi, è auspicabile ridurre al minimo gli errori di interpretazione delle lesioni visibili alla mammografia digitale. Per aiutare il medico a prendere decisioni e ridurre il numero di biopsie mammarie non necessarie, si sviluppano sistemi di supporto alla decisione per affiancare il medico nelle sue interpretazioni.

## 1.3 Scopo elaborato

In questo elaborato utilizzeremo i dati del dataset “Mammographic Masses” per cercare di predire la diagnosi utilizzando gli attributi a disposizione. Per fare ciò proponiamo due diversi approcci che si differenziano nel trattamento dei dati:

* Uno che esclude completamente l’unità statistica con dati mancanti
* L’altro che imputa i dati con un metodo di KNN

Per ogni approccio è stato poi fatto un pre-processing del data-set. Con i due dataset risultanti sono stati allenati vari modelli studiati durante il corso. Oltre ad allenare dei classificatori, si è deciso di visualizzare i dati in uno spazio di dimensione ridotta utilizzando la PCA e in questo caso si sono applicati dei classificatori per mostrare a livello grafico il Decision Boundary differenziando fra caso lineare e non lineare.

# 2 Dataset

## 2.1 Descrizione dataset

Questo set di dati può essere utilizzato per prevedere la gravità (benigna o maligna) di una lesione di massa mammografica da vari attributi sulla massa e dall'età del paziente. Contiene una valutazione BI-RADS, l'età del paziente e tre attributi che descrivono la forma della massa insieme alla classe reale di appartenenza (il campo SEVERITY) per 516 benigni e 445 masse maligne che sono state identificate su mammografie digitali raccolti presso l'Istituto di Radiologia dell’ Università Erlangen-Norimberga tra il 2003 e il 2006. Ogni istanza ha una valutazione BI-RADS associata che va da 0 a 6 assegnato in un processo di doppia revisione effettuata da un gruppo di medici. Assumendo che tutti i casi con valutazioni BI-RADS siano maggiori o uguali un dato valore, ​​sono maligni e gli altri casi benigni, possono essere calcolate le sensibilità e le specificità associate. Questi possono essere un’ indicazione delle prestazioni di un sistema CAD rispetto ai risultati raggiunti dai radiologi.

## 2.2 Descrizione attributi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome variabile** | **Tipo variabile** | **Range** | **Unità di misura** |  |
| BI-RADS | ORDINALE | Da 1 a 5 | - | NON PREDITTIVA |
| AGE | CONTINUA | --- | anni | PREDITTIVA |
| SHAPE | NOMINALE | Da 1 a 4 | - | PREDITTIVA |
| MARGIN | NOMINALE | Da 1 a 5 | - | PREDITTIVA |
| DENSITY | ORDINALE | Da 1 a 4 | - | PREDITTIVA |
| SEVERITY | CATEGORICA | 0-1 | - | TARGET |

Come è possibile vedere dalla tabella il dataset presenta 6 attributi totali di cui uno è il campo target (SEVERITY), un campo non è predittivo BI-RADS e gli altri 4 attributi (AGE, SHAPE, MARGIN, DENSITY, SEVERITY) sono predettivi:

* BI-RADS: Si tratta della classificazione americana dei livelli della malattia maligna

del seno. I valori del BI-RADS sono i seguenti:

|  |  |  |
| --- | --- | --- |
| **VALORE BI-RADS** | **RISCHIO** | **DESCRIZIONE** |
| **0** |  | Incompleto |
| **1** | Basso | Negativo |
| **2** | Basso | Benigno |
| **3** | <2% | Risultato probabilmente benigno (suggerito follow-up) |
| **4** | 25-50% | Anomalia sospetta (suggerita biopsia) |
| **5** | 75%-99% | Risultato probabilmente maligno |
| **6** |  | Accertato da biopsia |

* AGE: E’ una variabile continua che indica l’età. Si può pensare di categorizzarla per approcciarsi a determinati modelli.
* SHAPE: E’ una variabile nominale che indica la forma della massa:

|  |  |
| --- | --- |
| **VALORE SHAPE** | **DESCRIZIONE** |
| **1** | Round |
| **2** | Oval |
| **3** | Lobular |
| **4** | Irregular |

* MARGIN: E’ una variabile nominale che indica i margini della massa:

|  |  |
| --- | --- |
| **VALORE MARGIN** | **DESCRIZIONE** |
| **1** | Circumscribed |
| **2** | Microbulated |
| **3** | Obscured |
| **4** | Ill-defined |
| **5** | Spiculated |

* DENSITY: E’ una variabile nominale che indica la densità della massa:

|  |  |
| --- | --- |
| **VALORE DENSITY** | **DESCRIZIONE** |
| **1** | High |
| **2** | Iso |
| **3** | Low |
| **4** | Fat-containing |

* SEVERITY: E’ una variabile binomiale che indica la classificazione della massa tuomorale:

|  |  |
| --- | --- |
| **VALORE SEVERITY** | **DESCRIZIONE** |
| **0** | Benign |
| **1** | Malignant |

Nonostante i campi MARGIN, DENSITY, SHAPE siano stati definiti come variabili nominali, in realtà possono anche essere viste come variabili ordinali perché intrinsecamente posso ammettere un ordinamento che va da quello meno pericoloso a quello più pericoloso. Essendo il BI-RADS un attributo attributo da più medici a titolo diagnostico è stato definito come non predittivo. Utilizzeremo invece gli altri attributi per definire i vari modelli.

## 2.3 Dati mancanti e pulizia dati

### 2.3.1 Pulizia dati

Analizzando ogni variabile categorica possiamo controllare che i valori presenti nel dataset rispettino i range dei rispettivi attributi. Controllando i vari conteggi dei valori per ogni variabile otteniamo:

|  |  |
| --- | --- |
| **VALORE BI-RADS** | **CONTEGGIO** |
| **0** | 5 |
| **1** | 0 |
| **2** | 14 |
| **3** | 36 |
| **4** | 547 |
| **5** | 345 |
| **6** | 11 |
| **?** | 2 |
| **55** | 1 |

|  |  |
| --- | --- |
| **VALORE MARGIN** | **CONTEGGIO** |
| **1** | 357 |
| **2** | 24 |
| **3** | 116 |
| **4** | 280 |
| **5** | 136 |
| **?** | 48 |

|  |  |
| --- | --- |
| **VALORE SHAPE** | **CONTEGGIO** |
| **1** | 224 |
| **2** | 211 |
| **3** | 95 |
| **4** | 400 |
| **?** | 31 |

|  |  |
| --- | --- |
| **VALORE DENSITY** | **CONTEGGIO** |
| **1** | 16 |
| **2** | 59 |
| **3** | 798 |
| **4** | 12 |
| **?** | 76 |

|  |  |
| --- | --- |
| **VALORE SEVERITY** | **CONTEGGIO** |
| **0** | 516 |
| **1** | 445 |

Solo nel BI-RADS troviamo dei problemi con un valore anomalo “55” che è molto probabile sia un errore di battitura dovuto all’errore di un operatore. Per questo si è deciso di sostituirlo con la classe “5”.

### 2.3.2 Eliminazione dell’unità statistica (primo approccio)

Sono stati inizialmente analizzati i dati mancanti codificati con il simbolo “?” all’interno del dataset (che inizialmente aveva 961 unità statistiche). Per ogni variabile è stato calcolato il conteggio dei dati mancanti ottenendo il seguente risultato:

|  |  |
| --- | --- |
| **ATTRIBUTO** | **DATI MANCANTI** |
| **AGE** | 2 |
| **BI-RADS** | 5 |
| **SHAPE** | 31 |
| **MARGIN** | 48 |
| **SEVERITY** | 76 |
| **DENSITY** | 0 |

In totale i valori mancanti sono 162. Intersecando gli indici corrispondenti di queste righe si ottengono 131 risultati che hanno almeno un valore mancante. Per la ridotta dimensione si è deciso inizialmente di eliminare completamente l’unità statistica. Così facendo le frequenze assolute calcolate precedentemente nelle tabelle cambiano in questo modo:

|  |  |  |
| --- | --- | --- |
| **VALORE BI-RADS** | **DATI VALIDI** | **DATI ELIMINATI** |
| **0** | 5 | 0 |
| **1** | 0 | 0 |
| **2** | 7 | 7 |
| **3** | 24 | 12 |
| **4** | 468 | 79 |
| **5** | 317 | 28 |
| **6** | 9 | 2 |

|  |  |  |
| --- | --- | --- |
| **VALORE MARGIN** | **DATI VALIDI** | **DATI ELIMINATI** |
| **1** | 320 | 37 |
| **2** | 23 | 1 |
| **3** | 106 | 10 |
| **4** | 254 | 26 |
| **5** | 127 | 9 |

|  |  |  |
| --- | --- | --- |
| **VALORE SHAPE** | **DATI VALIDI** | **DATI ELIMINATI** |
| **1** | 190 | 34 |
| **2** | 180 | 31 |
| **3** | 81 | 14 |
| **4** | 379 | 21 |

|  |  |  |
| --- | --- | --- |
| **VALORE DENSITY** | **DATI VALIDI** | **DATI ELIMINATI** |
| **1** | 11 | 1 |
| **2** | 56 | 3 |
| **3** | 755 | 43 |
| **4** | 8 | 4 |

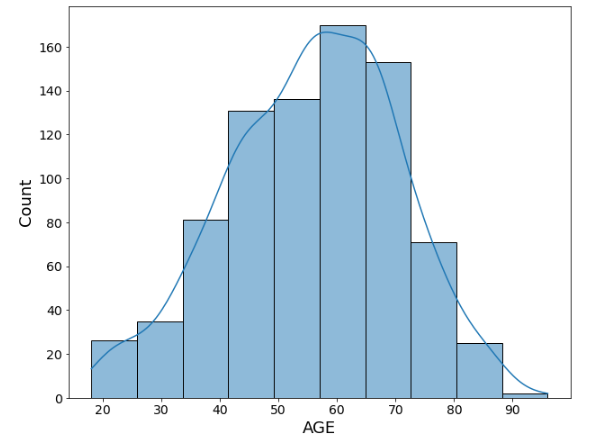
|  |  |  |
| --- | --- | --- |
| **VALORE SEVERITY** | **DATI VALIDI** | **DATI ELIMINATI** |
| **0** | 427 | 89 |
| **1** | 403 | 42 |

Dopo la pulizia delle varie tuple di dati, si ottiene una matrice di 830 elementi. Nella seguente tabella mostriamo un riassunto delle variabili. Da notare che per le variabili nominali la media non ha senso. Però siccome le stiamo trattando come variabili ordinali la media e altre statistiche possono dare un’idea iniziale della disposizione dei vari dati.

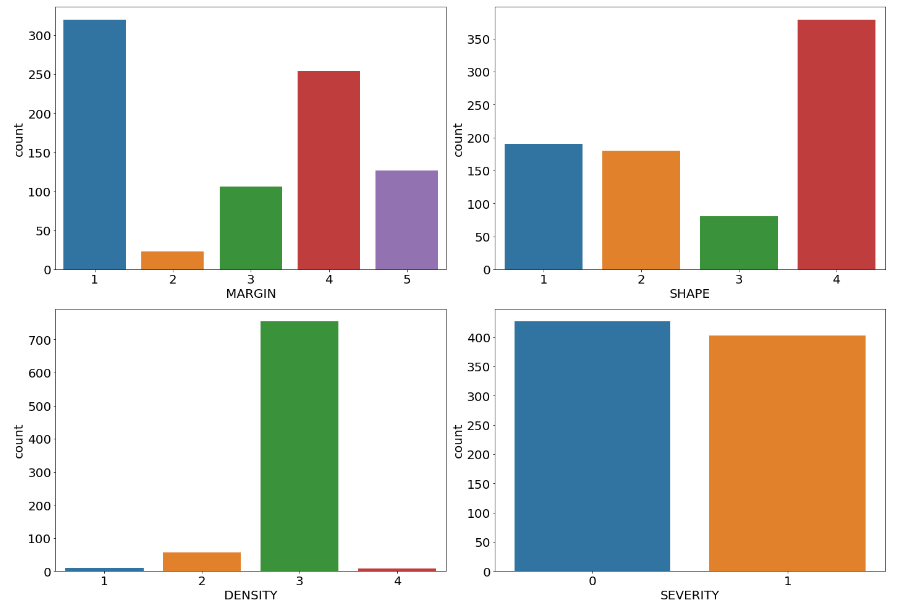
Immagine che contiene tavolo

Descrizione generata automaticamente

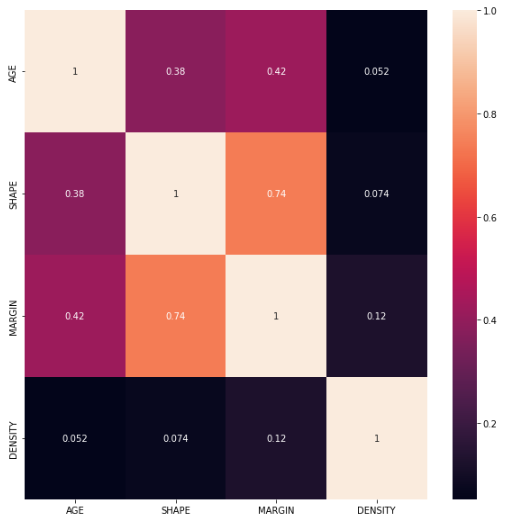
Per l’AGE mostriamo un istogramma costruito con 10 bin. Possiamo notare che la maggior parte dei casi in questo dataset aveva un’età intorno ai 60 anni, per il resto possiamo dire essere “abbastanza” simmetrica.



Per gli altri attributi invece, essendo variabili non continue si è optato per un diagramma a barre per mostrare la loro frequenza assoluta.



Con una heatmap controlliamo la correlazione degli attributi fra di loro. Come possiamo vedere dalla figura la Density è a quanto pare l’attributo meno correlato con gli altri. Tutti gli altri mostrano fra di loro correlazioni non completamente trascurabili come potevamo aspettarci essendo attributi che descrivono la massa tumorale. Nonostante ciò essendo pochi attributi si è deciso di lasciarli tutti per le prossime analisi.



### 2.3.3 Imputazione attraverso KNN (secondo approccio)

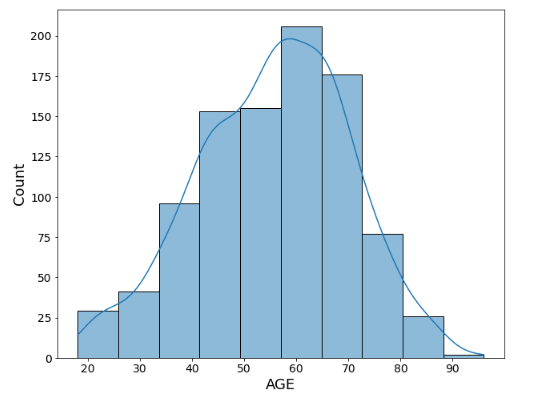
Un’altra possibilità è quella di imputare i dati mancanti. Ci troviamo davanti righe di dati in cui uno o più valori o colonne in quella riga non sono presenti. I valori potrebbero mancare completamente o essere contrassegnati da un carattere o valore speciale, ad esempio un punto interrogativo “?” come nel nostro caso. I valori potrebbero mancare per molte ragioni, a causa di un errore o per mancata raccolta di quest’ultimo. La maggior parte degli algoritmi di apprendimento automatico richiede valori di input numerici e un valore presente per ogni riga e colonna in un set di dati. Pertanto, la presenza di queste anomalie potrebbe causare problemi agli algoritmi di apprendimento automatico. In questo sotto-capitolo imputeremo i dati in modo da ottenere in ogni caso una matrice di dati completa. Utilizzeremo la libreria scikit-learn che fornisce la classe KNNImputer che supporta l’imputazione con i “vicini”. La classe KNNImputer ha bisogno in ingresso del numero di vicini da analizzare (di default è 5), la metrica che deve utilizzare per determinare la distanza e il peso che lasceremo uniforme fra le varie istanze. Sostituiamo i “?” con un NaN in modo che possa essere ignorato dall’algoritmo che calcola le varie distanze. Tutto questo procedimento viene fatto naturalmente eliminando l’attributo ‘SEVERITY’ che è il nostro target che non ha alcun valore mancante e che non deve assolutamente essere sottoposto a questo processo.

Dopo il lancio di questo algoritmo otteniamo una matrice che ha mantenuto le dimensioni originali di 961 righe. I dati mancanti erano comunque pochi soprattutto se prendiamo in considerazione le celle stesse e non la riga come nel metodo di prima. Questo approccio infatti potrebbe creare problemi in quanto cerca di aggiungere determinati dati che sono “artificiali”, ciò però come possiamo vedere dalla tabella seguente e per i motivi elencati precedentemente non distorce per niente la distribuzione dei vari attributi essendo un approccio molto solido. Infatti si basa sull’assunzione che i dati che mancano siano simili a quelli che abbiamo.

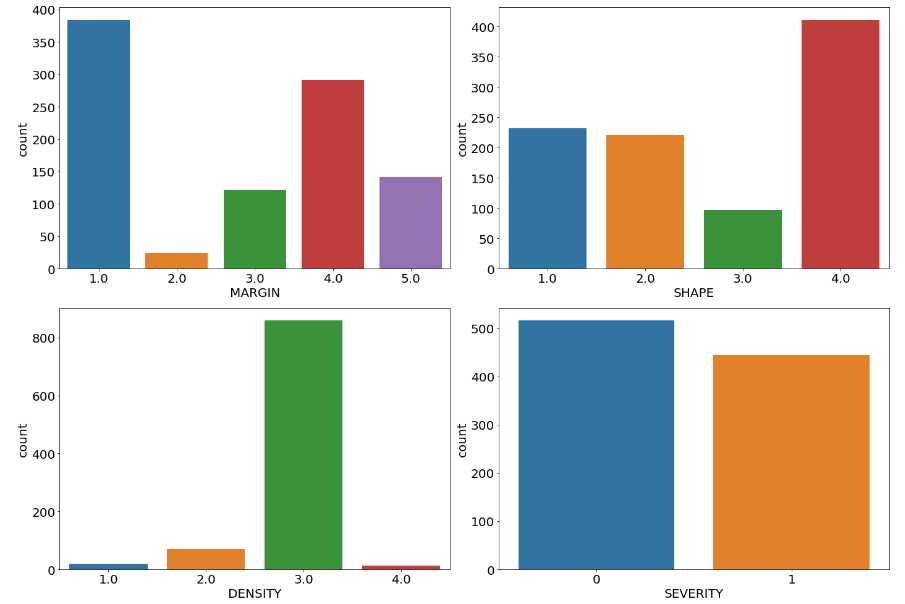
Immagine che contiene tavolo

Descrizione generata automaticamente

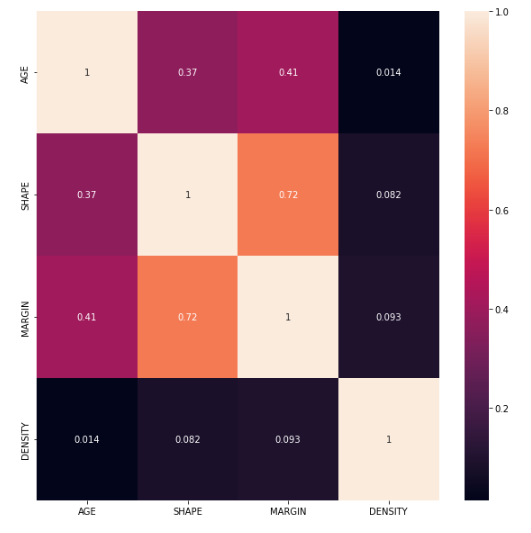
Con il nuovo approccio possiamo anche notare che la distribuzione dell’età ci porta in ogni caso alle stesse conclusioni di prima.



Ripetiamo l’analisi di prima per controllare i valori per ogni attributo. Adesso ci aspettiamo una frequenza assoluta diversa perché ci sono più dati che sono stati imputati e altri che sono stati in ogni caso recuperati perché abbiamo deciso di non cancellare interamente alcuna unità statistica.



Anche la heatmap del nuovo approccio conferma le stesse conclusione dell’approccio precedente.



## 2.4 Datasets finali

Otteniamo quindi due matrici di dati (una con un totale di 830 valori nella quale per evitare di mantenere dati non validi, indicati dal simbolo “?”, sono stati cancellati le intere unità statistiche che contenevano almeno un dato mancante. Ciò ha determinato la cancellazione anche di dati validi. La seconda matrice invece contiene 961 unità perché ogni valore mancante è stato imputato permettendo in questo modo di non cancellare eventuali tuple di dati con dati utili. Per entrambi i datasets sono stati presi in considerazione come features i 4 attributi (DENSITY, MARGIN, SHAPE, AGE) senza effettuare alcuna feature selection essendo già in numero limitato. In generale quando ci troviamo davanti tante features potremmo incorrere nel problema della dimensionalità, ma questo non è il nostro caso. Inoltre per entrambi i casi è stato eseguito uno split preliminare per permettere di definire un set di Training e un set di Test utilizzando il pacchetto **sklearn**.



Definendo il test\_size uguale a 0.2 si è deciso praticamente di fare una suddivisione in modo da lasciare al training l’80% dei dati e al test il 20%. Per evitare problemi con eventuali classificatori si è deciso di normalizzare le features di training e di applicare la stessa trasformazione (riutilizzando sempre i valori calcolati sul training) sul test. Per normalizzarli ho utilizzato il seguente codice: 

Uso il metodo fit() per stimare la media del campione e la deviazione standard dell'insieme di training (train\_features), in seguito con il metodo transform() standardizzo i dati dell'insieme di training (train\_features) usando i parametri appena calcolati dal metodo fit. Sovrascrivo le variabili delle features del training e test con le nuove variabili che hanno subito il processo di standardizzazione. Tutto questo è stato fatto perché le performance degli algoritmi sono migliori quando lavorano su dati standardizzati, in questo modo infatti assegniamo ad ogni attributo la stessa importanza (eventuali scale di misura differenti sui vari attributi può portare alla distorsione dell’allenamento).

Immagine che contiene testo

Descrizione generata automaticamenteImmagine che contiene testo

Descrizione generata automaticamente

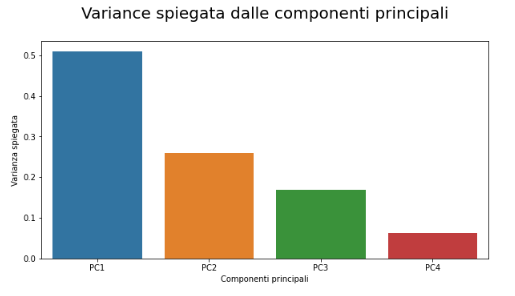
Questo tipo di approccio è stato eseguito sia per il dataset con i dati imputati che per il dataset dal quale sono stati cancellati i dati mancanti. Da questo momento in poi affronteremo tutti gli algoritmi su entrambi i dataset, non confrontando in modo quantitativo i diversi risultati essendo dataset diversi.

La normalizzazione delle caratteristiche (o standardizzazione dei dati) delle variabili esplicative (o predittive) è una tecnica utilizzata per centrare e normalizzare i dati sottraendo la media e dividendo per la varianza. Se avessi preso la media e la varianza dell'intero set di dati, avrei introdotto informazioni future nelle variabili esplicative dell'addestramento (cioè la media e la varianza). Pertanto, è necessario eseguire la normalizzazione delle funzionalità sui dati di addestramento. Quindi eseguire la normalizzazione anche sulle istanze di test, ma questa volta utilizzando la media e la varianza delle variabili esplicative dell'addestramento. In questo modo, possiamo testare e valutare se il nostro modello può generalizzarsi bene a nuovi dati mai visti prima.

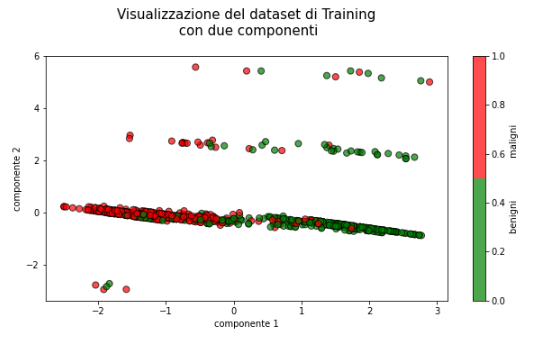
# 3 Visualizzazione in 2 dimensioni dei dati

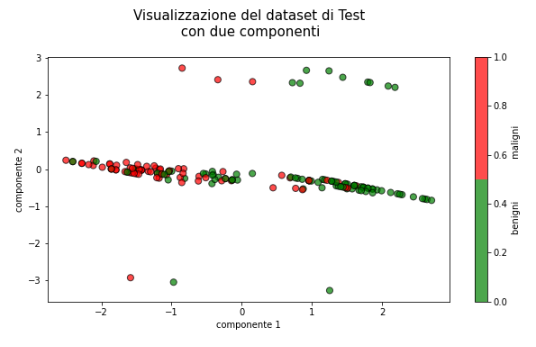
Ci troviamo davanti a un dataset con 4 features (DENSITY, MARGIN, SHAPE, AGE), per visualizzare i dati in 2 dimensioni è necessario fare una PCA (Principal Component Analysis) sfruttando la libreria “sklearn” di Python, più precisamente la classe PCA() del pacchetto “decomposition”. Come dataset utilizzeremo solo quello del primo approccio che non prevede l’imputazione di alcun elemento in modo da mostrare dati reali.





La maggior parte della varianza è definita dalle prime due componenti principali. Quindi consideriamo queste ultime per definire la visualizzazione 2d dei dati. Mostriamo con queste due componenti principali il dataset di training e il dataset di test.





## 

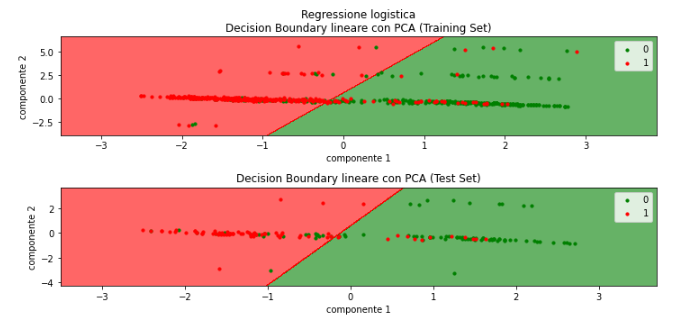
Proviamo a definire un decision boundary che possa permetterci di definire due spazi per separare nel miglior modo possibile i maligni dai benigni. Per far ciò proviamo a definire un decision boundary lineare con la Regressione Logistica e un decision boundary non lineare con le Random Forest. Per proiettare i dati in due dimensioni è necessario trasformare lo spazio delle features con il seguente codice:

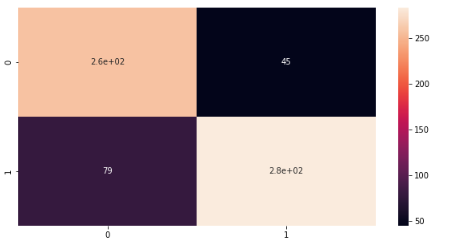


Naturalmente seguendo lo stesso tipo di ragionamento per il quale il modello va stimato con i dati di training e il test invece subisce la medesima trasformazione così come è stato fatto nel processo di standardizzazione precedente.

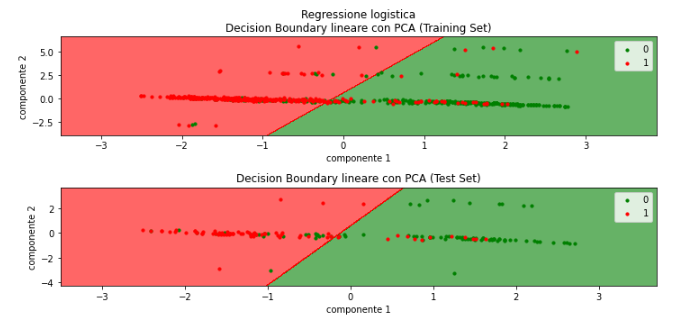
## 4.1 Calcolo Decision Boundary con Regressione Logistica

Con la seguente matrice “train\_features\_pca2” possiamo allenare e definire un decision boundary lineare con la Regressione Logistica. Di seguito mostro il nuovo spazio delle features con la matrice di confusione ottenuta classificando grazie al Decision Boundary.

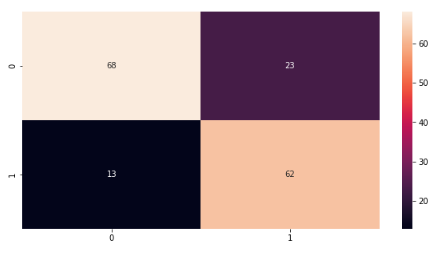




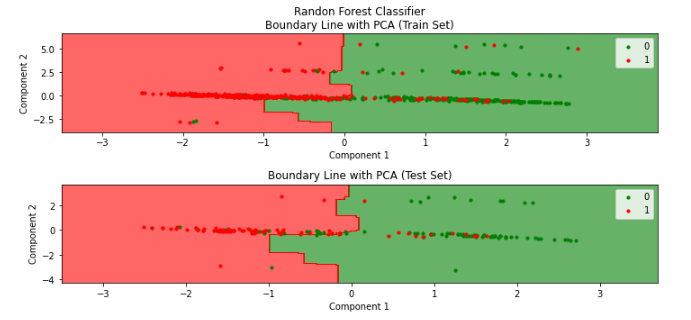
Applichiamo lo stesso classificatore che definiva questo Decision Boundary al test per capire se la classificazione generalizza bene per casi su cui non si è allenato.

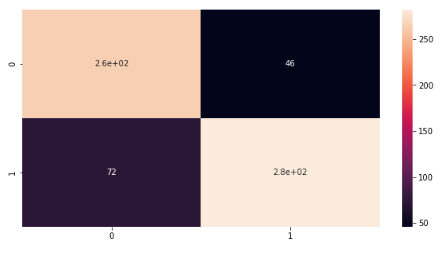


Come possiamo vedere sia dal plot che dalla matrice di confusione, il modello si comporta abbastanza bene anche per i dati di test

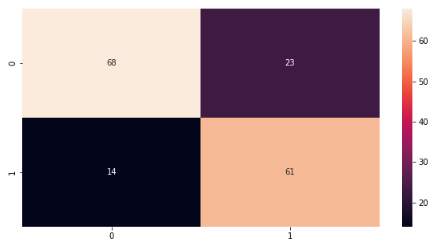
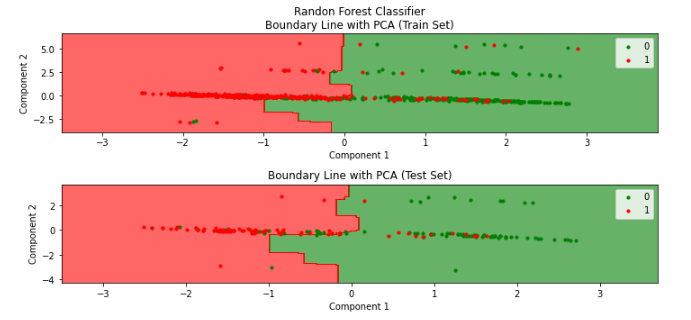


## 4.2 Calcolo Decision Boundary con Random Forest

Con la stessa matrice “train\_features\_pca2” utilizzata precedentemente per la regressione logistica possiamo allenare e definire un decision boundary non lineare con la Random Forest. Di seguito mostro il nuovo spazio delle features con la matrice di confusione ottenuta classificando grazie al Decision Boundary.



Di seguito mostro lo stesso modello applicati al test.



## 4.3 Commento sulla rappresentazione in due dimensioni

|  |  |  |
| --- | --- | --- |
|  | REGRESSIONE LOGISTICA | RANDOM FOREST |
| TRAINING | 0.8132 | 0.8222 |
| TEST | 0.7831 | 0.7771 |

Raccogliendo i risultati possiamo costruire una tabella nella quale mostriamo le accuratezze raggiunte nella fase di training e test sia per la Regressione Logistica, che per la Random Forest.

Commentando questi risultati possiamo confermare ciò che ci aspettavamo. Il Decision Boundary non lineare è più flessibile e quindi si adatta di più ai dati di training, adattandosi di più si comporta leggermente peggio rispetto al decision boundary lineare che invece sembra generalizzare meglio. In questo capitolo abbiamo solo approcciato una prima analisi, nel seguente alleneremo in modo approfondito una serie di modelli e confronteremo lo score in modo più quantitativo senza però preoccuparci della rappresentazione dei dati.

# 5 Allenamento modelli

## 5.1 Scelta classificatori

In questo capitolo affrontiamo invece la costruzione di modelli sullae matrici di features che avevamo definito prima della PCA. Si è deciso di allenare e confrontare i seguenti classificatori:

* **SUPPORT VECTOR MACHINE**

Kernel = RBF

C=1

Gamma=1

* **RANDOM FOREST**

Massima prodondità : 3 (La profondità massima dell’albero)

Criterio : Indice di Gini

* **REGRESSIONE LOGISTICA**

Algoritmo usato: liblinear che supporta le penalizzazione l1 e l2 ed è consigliato per piccoli set di dati.

* **K-NEAREST-NEIGHBORS**

Numero di vicini da considerare: 10

* **ALBERI DECISIONALI**

Numero minimo di campioni per effettuare lo split: 2

## 5.2 Hold-out method

Il metodo hold-out prevede la suddivisione del dataset in train e test. Il set di addestramento è ciò su cui viene eseguito il training del modello e il set di test viene utilizzato per valutare le prestazioni di quel modello su dati che sono “invisibili” al classificatore, cioè che non ha mai visto prima.

Una suddivisione comune quando si utilizza il metodo di controllo consiste nell'usare l'80% dei dati per l'addestramento e il restante 20% dei dati per testarlo.

Utilizzando una variabile models costruita come una dictionaries di python che contiene le varie istanze dei modelli scelti, il seguente codice ci ha permesso di allenare il modello sul training test (80%) e di testarlo sul test set (20%).



Ottenendo i seguenti risultati per il primo metodo:

|  |  |
| --- | --- |
| **MODELLI** | **ACCURATEZZA SUL TEST SET** |
| SUPPORT VECTOR MACHINE | 0.67 |
| RANDOM FOREST | 0.77 |
| REGRESSIONE LOGISTICA | **0.78** |
| K-NEAREST-NEIGHBORS | **0.78** |
| ALBERI DECISIONALI | 0.72 |

E i seguenti risultati per il secondo metodo:

|  |  |
| --- | --- |
| **MODELLI** | **ACCURATEZZA SUL TEST SET** |
| SUPPORT VECTOR MACHINE | 0.79 |
| RANDOM FOREST | 0.81 |
| REGRESSIONE LOGISTICA | **0.80** |
| K-NEAREST-NEIGHBORS | **0.78** |
| ALBERI DECISIONALI | 0.75 |

## 5.3 K-FOLD Validazione

La convalida incrociata o "convalida incrociata k-fold" è quando il set di dati viene suddiviso casualmente in "k" gruppi. Uno dei gruppi viene utilizzato come set di prova e gli altri come set di allenamento. Il modello viene addestrato sul set di allenamento e valutato sul set di prova. Quindi il processo viene ripetuto fino a quando ogni gruppo univoco è stato utilizzato come set di test.

In generale i risultati possono essere facilmente influenzati dalla particolare porzione di dati scelti per il training e per il test. Per rendere i risultati più solidi possibili si è deciso di approcciare una cross validation in modo da introdurre una variazione in fase di training del train-set. Ciò ci permetterà di ottenere un vettore di accuratezze per i vari modelli che ci permetterà di eseguire test di ipotesi per valutare la differenza dei modelli. Allo stesso modo per l’allenamento precedente si è sfruttata la variabile models per ciclare e all’interno del ciclo realizzare la cross-validation per ogni modello.



Mettendo insieme i risultati delle varie fold per ogni modello otteniamo un valore di accuratezza media per ogni modello:

|  |  |
| --- | --- |
| **MODELLI** | **ACCURATEZZA SUL TEST SET** |
| SUPPORT VECTOR MACHINE | 0.79 |
| RANDOM FOREST | 0.80 |
| REGRESSIONE LOGISTICA | **0.81** |
| K-NEAREST-NEIGHBORS | 0.79 |
| ALBERI DECISIONALI | 0.74 |

Per quanto riguarda il secondo approccio otteniamo:

|  |  |
| --- | --- |
| **MODELLI** | **ACCURATEZZA SUL TEST SET** |
| SUPPORT VECTOR MACHINE | 0.79 |
| RANDOM FOREST | **0.81** |
| REGRESSIONE LOGISTICA | 0.80 |
| K-NEAREST-NEIGHBORS | 0.78 |
| ALBERI DECISIONALI | 0.75 |

Con questa strategia le accuratezze finali sono decisamente migliorate, ciò è dovuto alla variazione intrinseca che avviene del dataset di training all’interno della cross-validazione. Ciò permette al modello di non vedere sempre i soliti dati e quindi ha maggiore potere di generalizzazione del problema rispetto all’allenamento precedente. Nel caso in cui ci fossero pochi dati questo problema si noterebbe di più, perché sicuramente andremmo in contro a un caso di overfitting nel quale il modello si adatta troppo ai dati di training causando una poca aderenza a quelli di test.

## 5.4 Confronto classificatori

Analizziamo adesso i vari valori di accuratezza per ogni iterazione della cross-validazione, estraendo il valore di accuratezza per ogni fold e per ogni modello possiamo confrontarli per capire se esistono differenze significative. Di seguito mostriamo un esempio di valori di accuratezza per ogni iterazione e ogni modello.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 fold | 2 fold | 3 fold | 4 fold | 5 fold | 6 fold | 7 fold | 8 fold | 9 fold | 10 fold |
| SVM | 0.746 | 0.783 | 0.807 | 0.795 | 0.855 | 0.746 | 0.831 | 0.795 | 0.855 | 0.722 |
| RL | 0.759 | 0.783 | 0.855 | 0.819 | 0.831 | 0.734 | 0.795 | 0.819 | 0.879 | 0.795 |
| KNN | 0.759 | 0.783 | 0.819 | 0.759 | 0.879 | 0.759 | 0.843 | 0.807 | 0.807 | 0.710 |
| RF | 0.746 | 0.807 | 0.843 | 0.807 | 0.855 | 0.734 | 0.819 | 0.783 | 0.843 | 0.759 |
| AD | 0.734 | 0.783 | 0.734 | 0.7469 | 0.7469 | 0.698 | 0.734 | 0.746 | 0.7469 | 0.687 |

Definendo come ipotesi nulla H0, l’ipotesi che i due classificatori siano simili. Se rifiuto l’ipotesi nulla allora le differenze dei due classificatori sono significative. Scegliendo un alfa uguale a 0.05 (cioè il livello di significatività) e realizzando un t-test per ogni coppia di modelli. Di seguito mostriamo una tabella con una riga per ogni confronto, indicando in verde i confronti per i quali è stata rifiutata l’ipotesi nulla, e quindi per i quali la differenza è statisticamente significativa. Per il primo approccio la seguente tabella mostra il valore della statistica e del p-value per i confronti a incrocio fra i vari modelli:

|  |  |  |
| --- | --- | --- |
| **CONFRONTI** | **STATISTICA** | **P-VALUE** |
| SVM vs RL | -0.5004 | 0.6228 |
| SVM vs KNN | 0.46929 | 0.64449 |
| SVM vs RF | -1.0549 | 0.305416 |
| SVM vs AD | 1.78497 | 0.0911 |
| RL vs RF | -0.5667 | 0.5779 |
| RL vs AD | 2.4834 | 0.0230 |
| RL vs KNN | 1.022 | 0.3199 |
| KNN vs RF | -1.6536 | 0.1155 |
| KNN vs AD | 1.3290 | 0.2004 |
| RF vs AD | 3.3251 | 0.00376 |

Per il secondo approccio:

|  |  |  |
| --- | --- | --- |
| **CONFRONTI** | **STATISTICA** | **P-VALUE** |
| SVM vs RL | -0.5004 | 0.6228 |
| SVM vs KNN | 0.4692 | 0.6444 |
| SVM vs RF | -1.0549 | 0.30541 |
| SVM vs AD | 1.7849 | 0.09112 |
| RL vs RF | -0.5667 | 0.5779 |
| RL vs AD | 2.483 | 0.0230 |
| RL vs KNN | 1.0228 | 0.3199 |
| KNN vs RF | -1.6536 | 0.11552 |
| KNN vs AD | 1.3290 | 0.2004 |
| RF vs AD | 4.0449 | 0.0007 |

Possiamo concludere che l’unico modello effettivamente diverso dagli altri è quello dell’albero decisionale che ha le performance peggiori. La difficoltà nell’avere tanta differenza fra i vari classificatori è dovuta alla quantità di dati ridotta del dataset. Se avessimo avuto un dataset più ampio le differenze delle applicazioni dei vari modelli si sarebbero evidenziate più facilmente.

# 6 Conclusione

Un’ approccio di questo tipo può essere affrontato in modo più accurato con una quantità di dati maggiore, infatti non possiamo pensare che questi pochi dati possano rappresentare con precisione questo tipo di problema. Nel caso in cui avessimo avuto più dati si sarebbe anche potuto utilizzare metodi di allenamento più solidi. Si poteva realizzare una ricerca di parametri per un dato modello validandolo grazie a un validation set, in questo caso una grid search con un cambiamento dei parametri avrebbe portato alla scelta dei parametri migliori. Solo a quel punto avremmo potuto usare il test set per capire come in questo elaborato se il modello definito si comportasse bene.