

19/05/2020

Gender Classification

Università degli Studi di Salerno



**Fondamenti di Visione Artificiale e
Biometria**



| Team Member | |
|--------------------|------------|
| Nome | Matricola |
| Paolo Cantarella | 0522500793 |
| Silvio Corso | 0522500792 |
| Carminc Tramontano | 0522500865 |



Sommario

| | |
|---|----|
| 1. Introduzione..... | 3 |
| 2. Metodi utilizzati | 4 |
| 2.1 Radial Basis Function Support Vector Machine..... | 4 |
| 2.2 Decision Tree..... | 4 |
| 2.3 K-Nearest Neighbors | 6 |
| 3. Esecuzione e risultati ottenuti | 7 |
| 3.1 RBF SVM..... | 7 |
| 3.2 Decision Tree..... | 9 |
| 3.3 K-Nearest Neighbors | 10 |
| 4. Conclusioni..... | 12 |
| 5. Appendice..... | 13 |
| 5.1 Tabella riassuntiva risultati RBF SVM | 13 |
| 5.2 Tabella riassuntiva risultati Decision Tree..... | 16 |
| 5.3 Tabella riassuntiva risultati K-Nearest Neighbors | 23 |
| 6. Bibliografia | 30 |

1. Introduzione

Al giorno d'oggi la possibilità di poter riconoscere il sesso di una persona a partire da una sua foto o da un frame video è alla base di molti sistemi di sicurezza, quali le telecamere degli aeroporti o delle banche. Per questo motivo si è deciso di affrontare il problema andando a sperimentare nuove tecniche, in modo tale da ottenere un'accuratezza che possa superare quella degli attuali sistemi riguardanti la gender classification. Numerose sono le tecniche presenti in letteratura per ottenere risultati positivi su un problema del genere, tuttavia il nostro studio si è basato sull'utilizzo di una tecnica non ancora consolidata. La tecnica in questione è quella della "pose estimation", ampiamente spiegata nel paper "Web-Shaped Model for Head Pose Estimation: An Approach for Best Exemplar Selection" [1].

La pose estimation ci ha permesso di elaborare le immagini fornite in input, in nuove immagini, costruite con una "ragnatela virtuale" che suddivide le immagini in cerchi e settori. Ciascun settore contiene un numero variabile di landmark, dipendente da quanti ne vengono individuati in quella precisa zona del viso. Ciò che abbiamo fatto è stato quindi passare ai nostri classificatori le immagini elaborate con diverse configurazioni e valutare l'accuratezza con cui veniva stabilito il sesso dell'individuo.

È possibile elaborare le immagini con diverse configurazioni basate sul numero di cerchi e il numero di "fette", ovvero, è possibile ottenere un numero diverso di settori in cui viene suddiviso il volto a seconda della configurazione scelta. Sono state scelte le tre configurazioni che hanno dato i risultati migliori per la pose estimation:

- 5 cerchi 4 fette (80 settori)
- 4 cerchi 4 fette (64 settori)
- 4 cerchi 3 fette (48 settori)

Il dataset scelto per l'allenamento dei classificatori è stato "CelebA": un dataset contenente 202599 foto di personaggi famosi, uomini e donne e in pose differenti. Al dataset era anche già associato un file .csv contenente diverse features per ciascuna foto, ma l'unica di nostro interesse è stata "Male" ovvero la colonna relativo al sesso del soggetto. Un dettaglio importante al fine test è che il dataset è risultato sbilanciato, ovvero vi erano più volti femminili che maschili, per questo motivo si è passati per una fase di "postprocessing" in cui si è rielaborato il dataset per ottenerne uno con pari numero di uomini e donne.

Per quanto riguarda la scelta dei classificatori, è stata fatta al fine di massimizzare i risultati del problema. Chiaramente, sono numerosi i classificatori e non è detto che i tre scelti da noi siano quelli che possano dare i migliori risultati in assoluto, ma in base ai nostri studi dovrebbero essere quelli che quanto più si avvicinano ai risultati ottimali. Data questa premessa la scelta è ricaduta quindi su:

- Radial Basis Function Support Vector Machine
- Decision Tree
- K-Nearest Neighbors

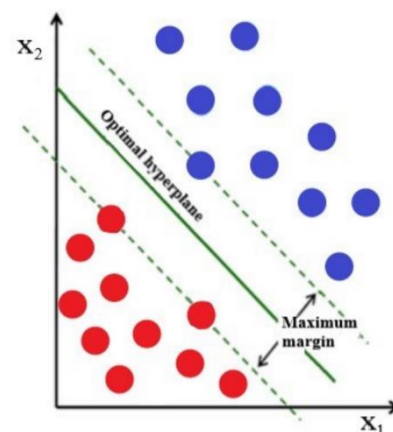
Nel capitolo successivo, verranno descritti nel dettaglio e spiegato il loro funzionamento affinché si possa comprendere in che modo abbiamo variato i parametri durante i nostri test.

2. Metodi utilizzati

2.1 Radial Basis Function Support Vector Machine

Un primo metodo di apprendimento supervisionato, per trattare l'argomento in questione, è stato il Support Vector Machine. In generale, con esso si fa riferimento ad un insieme di metodi utilizzati per la classificazione, regressione e outlier detection, ed inoltre sono alla base di una classificazione binaria e multi-classe su un insieme di dati in input. [2] Dato un insieme di esempi per l'addestramento, ognuno dei quali etichettato con la classe di appartenenza fra le due possibili classi, un algoritmo di addestramento per le SVM costruisce un modello che assegna i nuovi esempi a una delle due classi. Un modello SVM è una rappresentazione degli esempi come punti nello spazio, mappati in modo tale che gli esempi appartenenti alle due diverse categorie siano chiaramente separati da uno spazio il più possibile ampio.

Formalmente, una SVM costruisce un iperpiano o un insieme di iperpiani [3] in uno spazio a più dimensioni o a infinite dimensioni. Intuitivamente, una buona separazione si può ottenere dall'iperpiano che ha la distanza maggiore dal punto (del *training set*) più vicino di ognuna delle classi; in generale maggiore è il margine fra questi punti, minore è l'errore di generalizzazione commesso dal classificatore.



I nuovi esempi sono quindi mappati nello stesso spazio e la predizione della categoria alla quale appartengono viene fatta sulla base del lato nel quale ricade. Alla base della scelta di tale metodo vi è stata l'analisi dei vari vantaggi dell'SVM, tra i quali la sua grande versatilità dovuta dalla possibilità di specificare diversi valori del kernel, il quale rappresenta una classe di algoritmi di analisi, per la funzione decisionale come "linear", "sigmoid", "polynomial" e "rbf". [4] Quest'ultimo infatti è stato il valore utilizzato per il parametro kernel del metodo SVM per creare un modello di addestramento dei dati e della successiva predizione. Quindi è stato possibile adottare la potenza di tale metodo di classificazione per poter addestrare il modello, su una grande quantità di dati, ben distribuiti, come il dataset in questione. La grandissima robustezza delle SVM su quest'ultimo, combinata alla configurazione dei parametri della SVM stessa, ha evitato la peculiarità di un addestramento scorretto. Il numero di feature prese in considerazione in relazione al numero delle samples, non ha comportato un addestramento inadeguato, ma viceversa, SVM ha prodotto in output un risultato molto soddisfacente con una percentuale di accuratezza e affidabilità relativamente alta, che hanno confermato la grandissima versatilità di questo metodo.

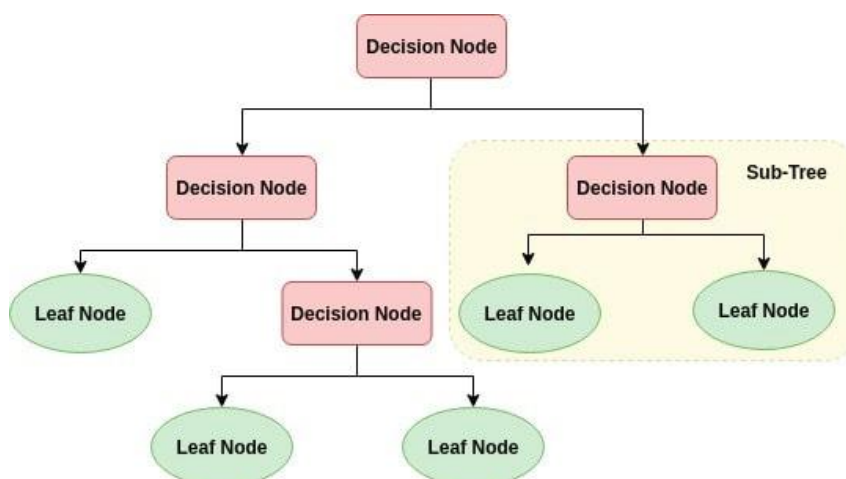
2.2 Decision Tree

L'algoritmo "Decision Tree" è un algoritmo con una struttura ad albero, [5] dove ciascun nodo interno rappresenta una feature, un arco rappresenta una regola di decisione e ogni nodo foglia rappresenta l'output. Il nodo radice, ovvero quello da cui ha origine l'intero albero, è dove a seconda della feature scelta avviene il partitioning e in maniera ricorsiva i nodi figli svolgono lo stesso tipo di partizione.

Questo tipo di algoritmo è molto utilizzato nel campo del machine learning in quanto emula in maniera facile ed intuitiva il comportamento del cervello umano. Oltre a questo, altri motivi ci hanno spinti ad utilizzare il Decision Tree:

- Facile da capire ed interpretare
- Training time più veloce rispetto ad una rete neurale
- Può gestire un grande numero di dati garantendo un'alta accuracy

In figura è mostrata la tipica struttura di un Decision Tree:



Nello specifico l'algoritmo opera come segue:

1. Seleziona il miglior attributo usando Attribute Selection Measure (ASM) per partizionare i record del dataset.
2. Rende tale attributo un "decision node" e divide il dataset in sottoinsiemi più piccoli
3. Inizia a costruire l'albero ripetendo i due punti precedenti ricorsivamente finché non viene raggiunta una delle seguenti condizioni:
 - Tutte le tuple appartengono allo stesso attributo
 - Non ci sono più attributi rimanenti
 - Non ci sono più istanze

L'Attribute Selection Measure è un metodo secondo il quale vengono stabiliti i criteri di partizione dei dati nel miglior modo possibile. Esso non fa altro che attribuire uno score a ciascuna feature, per poi selezionare quella con lo score più alto come attributo di riferimento per il partitioning (nodo radice). Diversi sono i Selection Measure esistenti e per il nostro studio ne sono stati utilizzati due: Information Gain e Gini Index. [6]

L'**Information Gain** viene utilizzato per determinare quale feature ci fornisce il massimo delle informazioni su una classe. Si basa sul concetto di "entropia" che è il grado di incertezza/impurità/disordine del dataset di input. Mira a ridurre il livello di entropia a partire dal nodo radice fino ai nodi figli. Di seguito viene mostrata la formula dell'entropia:

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

"p", indica la probabilità ed E(S) indica l'entropia. L'entropia non è preferita a causa della funzione "log" in quanto aumenta la complessità computazionale.

Il **Gini Index** misura il grado o la probabilità che una particolare feature venga erroneamente classificata quando viene scelta casualmente. Se tutti gli elementi appartengono a una singola classe, allora può essere chiamato puro. Il grado di Gini Index varia tra 0 e 1, dove 0 indica che tutti gli elementi appartengono a una determinata classe o se esiste solo una classe e 1 indica che gli elementi sono distribuiti in modo casuale tra varie classi. Un indice Gini di 0,5 indica elementi equamente distribuiti in alcune classi. Di seguito viene mostrata la formula per calcolare il Gini Index:

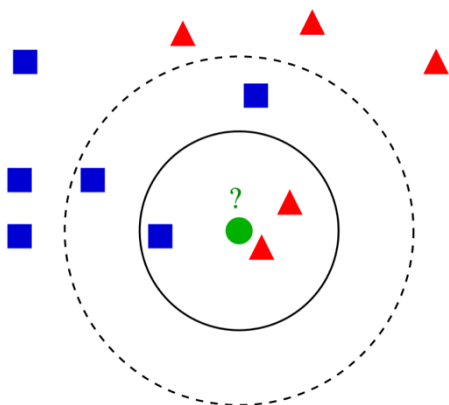
$$\text{Gini} = 1 - \sum_{i=1}^n (p_i)^2$$

dove p_i è la probabilità che un oggetto venga classificato in una particolare classe. Durante la creazione del Decision Tree, si preferisce scegliere la feature con il minimo Gini Index come nodo radice.

2.3 K-Nearest Neighbors

L'algoritmo k-Nearest Neighbor si basa sul concetto di classificare un campione incognito considerando la classe dei k campioni più vicini dell'insieme di addestramento. Il nuovo campione verrà assegnato alla classe a cui appartengono la maggior parte dei k campioni più vicini. La scelta di k è quindi molto importante affinché il campione venga assegnato alla classe corretta. Se k è troppo piccolo, la classificazione può essere sensibile al rumore, se k è troppo grande invece la classificazione può essere computazionalmente costosa e l'intorno può includere campioni appartenenti ad altre classi. Per realizzare un classificatore di tipo k-NN bisogna calcolare le distanze tra il campione e tutti i campioni di addestramento, identificare i k campioni di training più vicini e la loro rispettiva etichetta e scegliere infine la classe a cui appartiene la maggioranza dei campioni di training più vicini.

I principali vantaggi di questo metodo sono che esso non richiede l'apprendimento né la costruzione di un modello, può adattare i propri confini di decisione in modo arbitrario, producendo una rappresentazione del modello più flessibile e inoltre garantisce la possibilità di incrementare l'insieme di addestramento. Questo algoritmo presenta però anche molti svantaggi tra cui il fatto di essere suscettibile al rumore dei dati, di essere sensibile alla presenza di caratteristiche irrilevanti e di richiedere una misura di similarità per valutare la vicinanza.



[7] In figura è rappresentato un esempio di classificazione mediante kNN. Il punto sotto osservazione è il pallino verde. Se $k = 3$ (cioè vengono considerati i 3 oggetti più vicini), allora il pallino verde viene inserito nella stessa classe dei triangolini rossi perché sono presenti 2 triangolini e 1 quadratino. Se $k = 5$ allora viene inserito nella stessa classe dei quadratini blu perché sono presenti 3 quadratini e 2 triangolini.

3. Esecuzione e risultati ottenuti

Prima di passare all'esecuzione e ai risultati ottenuti da ciascun classificatore è bene fare una premessa: come anticipato nell'introduzione, è stata necessaria la creazione di un dataset bilanciato affinché i nostri test potessero risultare il più validi possibile.

Il dataset è stato creato unicamente sulla configurazione 4c4s, in quanto, come vedremo, è stata la configurazione che ha dato i risultati migliori su tutti e tre i classificatori. Il dataset bilanciato è composto da 168868 elementi (84434 donne, 84434 uomini).

3.1 RBF SVM

L'esecuzione del metodo di classificazione SVM con kernel RBF, come detto in precedenza, ha prodotto risultati molto soddisfacenti, in relazione alla modifica dei numerosi parametri che esso mette a disposizione, tra i quali i più importanti sono C e gamma. C corrisponde al parametro di regolarizzazione che con l'aumento del suo valore esso può comportare un eccesso di dati di allenamento, ovvero overfitting. Il parametro gamma definisce il limite di un singolo esempio di allenamento, all'aumentare di gamma si incorre in un overfitting dei dati. Impostando il parametro su "auto", il classificatore si adatta perfettamente ai dati di allenamento. Si è raggiunti un'accuratezza differente per i tre dataset dati in input a suddetto classificatore. Il primo dataset sottoposto (4 cerchi e 3 fette), avente una dimensione di 48 elementi per ogni array corrispondente ad una foto, ha prodotto i seguenti risultati:

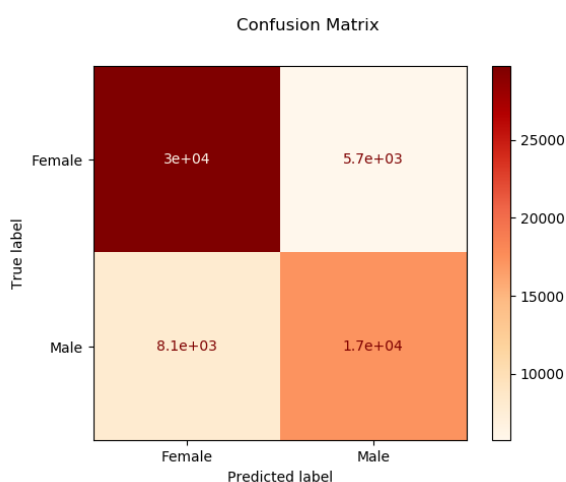
```
Classification report for classifier SVC(C=10, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
max_iter=-1, probability=False, random_state=1, shrinking=True, tol=0.001,
verbose=False):
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| -1 | 0.79 | 0.84 | 0.81 | 35436 |
| 1 | 0.75 | 0.68 | 0.72 | 25344 |
| accuracy | | | 0.77 | 60780 |
| macro avg | 0.77 | 0.76 | 0.76 | 60780 |
| weighted avg | 0.77 | 0.77 | 0.77 | 60780 |

```
Confusion matrix:
[[29716 5720]
 [ 8059 17285]]
```

Tempo trascorso: 134.47m

```
Accuratezza:
0.7732971372161895
```



Da come si evince in figura, si è raggiunti un'accuratezza del 77% nel classificare al meglio, su l'insieme di elementi (Testing Set) di 60.780 elementi, un volto maschile da quello femminile e viceversa. Nella figura viene mostrata la configurazione migliore raggiunta attraverso la modifica dei vari parametri; inoltre

vengono mostrate due versioni della matrice di confusione dove, in una si può notare il numero di campioni analizzati sottoforma di array bidimensionale, e nell'altra rappresentazione, sviluppata in una modalità graficamente più intuitiva, vengono mostrati i dati correttamente predetti in funzione dell'accentuarsi del colore, come mostra l'indicatore di riferimento verticale.

Successivamente, mantenendo sempre la configurazione ottimale, è stato dato in input un altro dataset (5 cerchi e 4 fette). I risultati sono stati i seguenti:

```
Classification report for classifier SVC(C=10, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
max_iter=-1, probability=False, random_state=1, shrinking=True, tol=0.001,
verbose=False):
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| -1 | 0.58 | 0.93 | 0.72 | 35436 |
| 1 | 0.41 | 0.07 | 0.12 | 25344 |
| accuracy | | | 0.57 | 60780 |
| macro avg | 0.50 | 0.50 | 0.42 | 60780 |
| weighted avg | 0.51 | 0.57 | 0.47 | 60780 |

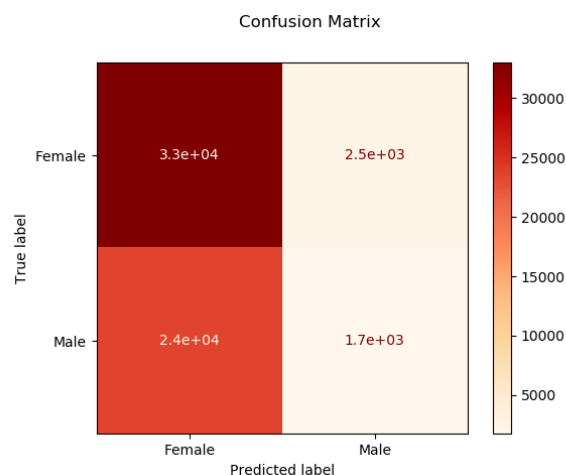
Confusion matrix:

```
[[32979 2457]
 [23619 1725]]
```

Tempo trascorso: 226.10m

Accuratezza:

0.5709772951628825



Il dataset non ha prodotto risultati confortanti, sfiorando la soglia del 57% di accuratezza, rilevandosi non adeguato per questa tipologia di problematica, in quanto con un'accuratezza simile i risultati saranno, per la maggior parte dei casi, casuali. Le due matrici di confusione indicano un basso equilibrio di predizione. Un cambio dei parametri del metodo non ha prodotto ugualmente dei risultati migliori. Infine, è stato posto come input un terzo dataset, quello corrispondente alla configurazione migliore per il problema della pose estimation (4 cerchi e 4 fette). Esso ha prodotto i seguenti risultati:

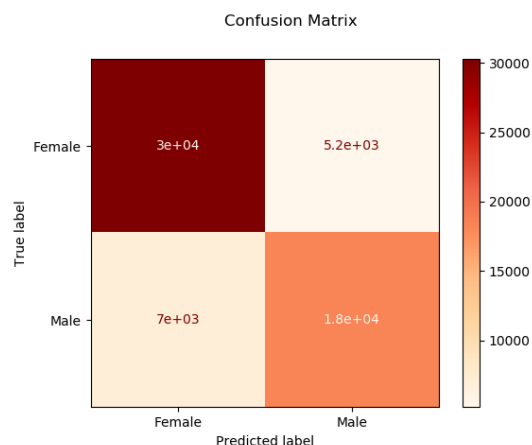
```
Classification report for classifier SVC(C=10, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
max_iter=-1, probability=False, random_state=1, shrinking=True, tol=0.001,
verbose=False):
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| -1 | 0.81 | 0.85 | 0.83 | 35436 |
| 1 | 0.78 | 0.72 | 0.75 | 25344 |
| accuracy | | | 0.80 | 60780 |
| macro avg | 0.80 | 0.79 | 0.79 | 60780 |
| weighted avg | 0.80 | 0.80 | 0.80 | 60780 |

```
Confusion matrix:
[[30266 5170]
 [ 7029 18315]]
```

Tempo trascorso: 131.85m

```
Accuratezza:
0.7992925304376439
```



Tale dataset, associato alla configurazione migliore del modello SVM in questione ha prodotto un'accuratezza pari all'80%. Infatti, vengono ben distinti gli elementi corrispondenti al sesso maschile rispetto a quelli del sesso femminile, raggiungendo un risultato molto soddisfacente. Le due matrici di confusione mostrano come il numero di elementi predetti correttamente sia molto elevato.

Infine, data la problematica del dataset preso in questione, circa lo sbilanciamento in favore del sesso femminile, è stato effettuato un processo di bilanciamento, il quale ha portato allo stesso numero di elementi di sesso maschile e femminile, al fine di migliorare ulteriormente il processo di training e successivamente di test. Nell'appendice sono stati inseriti i risultati circa questo nuovo test, con la configurazione migliore del metodo di classificazione SVM kernel RBF.

3.2 Decision Tree

L'esecuzione del classificatore basato sul "Decision Tree" è stata effettuata facendo variare due parametri: "criterion" ovvero l'attribute selection measure e "max_depth" ovvero la massima profondità del decision tree che veniva prodotto in output. Questi due parametri sono stati fatti variare in modo tale da ottenere il miglior risultato possibile di accuracy su ciascuna configurazione del dataset CelebA, come detto nel capitolo 1 i classificatori sono stati testati sulle tre configurazioni migliori dell'algoritmo di pose estimation.

Per quanto riguarda il dataset ottenuto dalla configurazione 4 cerchi e 3 fette, l'accuracy migliore è stata pari al 71.9%. I parametri utilizzati sono stati criterion: "gini" e max_depth: 13.

Con criterion: "entropy" (Information Gain) la massima accuracy è stata 71.5% molto vicina a quella con Gini Index ed è stata raggiunta con max_depth: 12.

I decision tree ottenuti in output non è possibile mostrarli in quanto di dimensioni troppo elevate e non risulterebbero leggibili nel paper.

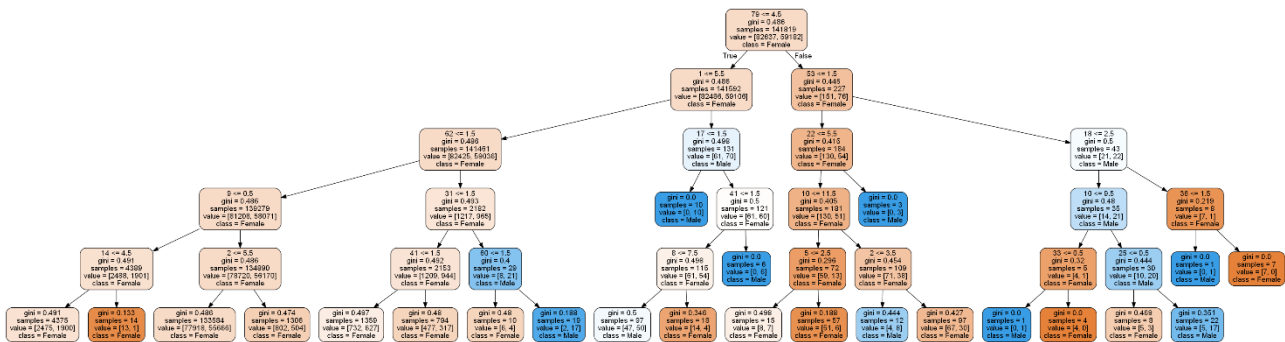
Con il dataset ottenuto dalla configurazione 4 cerchi e 4 fette, si sono avuti i risultati migliori.

L'accuracy massima è pari al 73.8%, sempre con criterion: "gini" e max_depth: 13. Anche in questo

caso la differenza rispetto al criterion: “entropy” è stata minima, con max_depth:12 è stato raggiunto il massimo di accuracy pari a 73.7%. Anche in questo caso i decision tree ottenuti in output, non è possibile mostrarli.

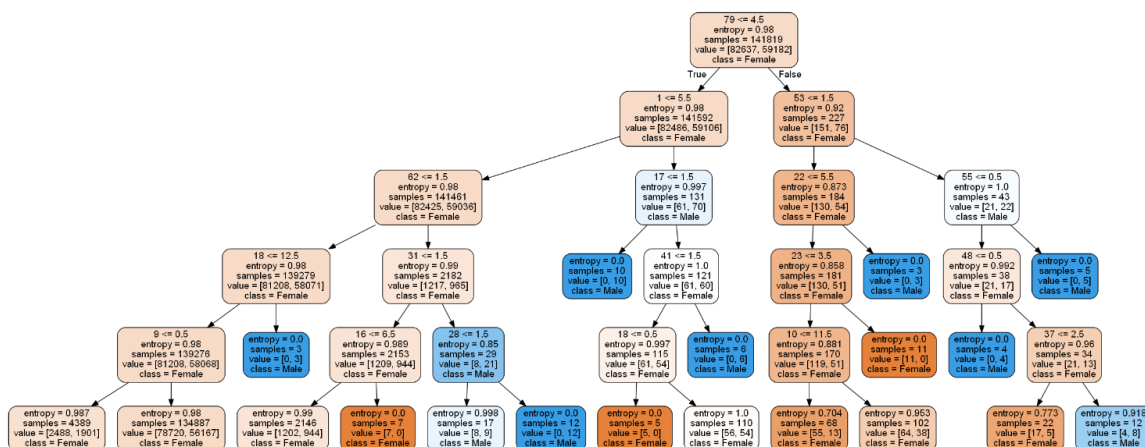
I risultati peggiori in assoluto si sono avuti sul dataset con configurazione 5 cerchi e 3 fette. L’accuracy massima è stata del 58.4% con criterion: “gini” e max_depth: 5.

Di seguito il decision tree ottenuto in output:



Anche con criterion: “entropy” e max_depth: 5 si è ottenuta la stessa accuracy.

Di seguito il decision tree ottenuto in output:

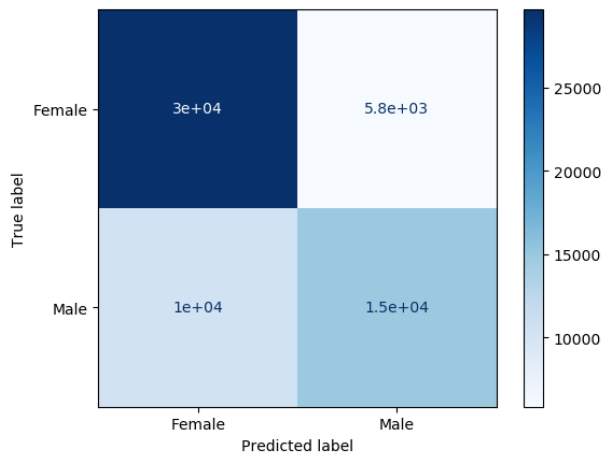


Infine, sul dataset bilanciato si è raggiunto il 72.3% di accuracy con criterion: “gini” e max_depth:13 e 72.1% con criterion: “entropy” e max_depth: 14. Da ciò si deduce che nel caso del Decision Tree operare su un dataset bilanciato non influenza particolarmente l’accuracy dei risultati ottenuti.

3.3 K-Nearest Neighbors

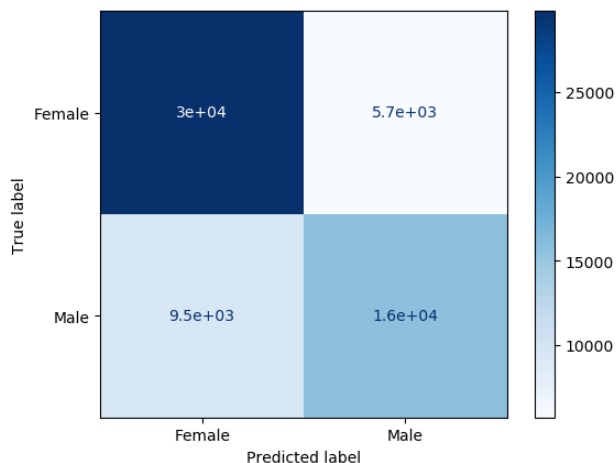
Come detto nel paragrafo 2.3, la scelta del valore k risulta importante per la corretta assegnazione del campione. K non deve essere né troppo grande, per evitare una costosa computazione, né troppo piccolo per evitare che il campione sia sensibile al rumore. Per questi motivi è stato scelto di eseguire il k-NN variando il valore k all’interno dell’intervallo [3,10].

Confusion Matrix



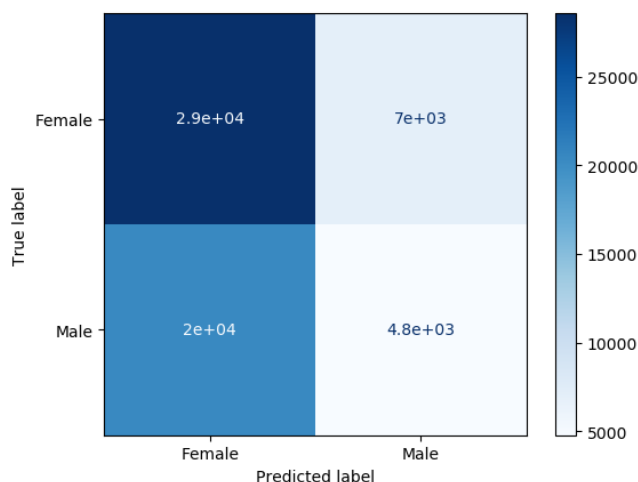
La prima esecuzione è avvenuta sul dataset composto da 4 cerchi e 3 fette, che associa un array di 48 elementi per ogni immagine. L'algoritmo al variare di k ha prodotto un'accuratezza media del 71.7%. La figura mostra la matrice di confusione con i risultati per il valore $k=9$, ovvero il miglior risultato ottenuto con un'accuratezza del 73.2%.

Confusion Matrix



La seconda esecuzione è avvenuta sul dataset 4 cerchi 4 fette. Tale dataset risulta essere il migliore in termini di accuratezza. L'accuratezza media è stata del 73.2%, trovando il migliore risultato ancora una volta per il $k=9$ con un'accuratezza del 74.9%.

Confusion Matrix



La terza esecuzione è avvenuta sul dataset 5 cerchi 4 fette. Per tale dataset i valori calcolati risultano i peggiori in termini di accuratezza, con una media del 53%. In questo caso il migliore risultato è stato ottenuto per $k=10$ con un'accuratezza del 54.8%.

Infine, l'ultima esecuzione è avvenuta sul dataset 4 cerchi 4 fette bilanciato. L'accuratezza ottenuta è stata del 74.3% per $k=9$. Lo stesso parametro utilizzato sul dataset completo aveva prodotto un'accuratezza leggermente più alta. Possiamo dedurre che nel caso di tale algoritmo l'utilizzo di un dataset bilanciato non migliora l'accuracy ottenuta.



4. Conclusioni

Come mostrato nel capitolo precedente, il dataset basato sulla configurazione che ha dato i migliori risultati, a prescindere dal classificatore utilizzato, è stato quello 4 cerchi 4 fette composto quindi da un totale di 64 settori, nonostante fosse basato su un dataset non bilanciato. L'utilizzo del dataset bilanciato ha dato dei buoni risultati ma non i migliori, probabilmente perché per poterlo creare, il numero di elementi nel test e nel train set sono diminuiti, facendo così allenare i classificatori su un numero ridotto di immagini rispetto al dataset intero. Si evince quindi, che così come la configurazione 4 cerchi 4 fette sia stata la migliore per la pose estimation, è risultata esserla anche per la gender classification.

Per quanto riguarda i classificatori, il migliore, come ci si aspettava, è stato RBF SVM. È stato in grado di raggiungere un'accuracy del 80% contro il 73.8% raggiunto dal Decision Tree e il 74.9% raggiunto dal K Nearest Neighbors. Ci si aspettava un risultato del genere proprio perché numerosi studi in letteratura dimostrano la forza di questo classificatore rispetto agli altri, seppur restino comunque dei classificatori validi per la gender classification.

5. Appendice

5.1 Tabella riassuntiva risultati RBF SVM

| #Test | Configurazione | Accuratezza | Output | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---|-------------|--|---------|--|--|--|--|-----------|--------|----------|---------|----|------|------|------|-------|---|------|------|------|-------|----------|--|--|------|-------|-----------|------|------|------|-------|--------------|------|------|------|-------|
| 1 | 80.000 elementi del dataset(4c3s), con 80% Training-Set e 20% Test-Set. C = 1 Gamma = auto Random state = 1 | 76.4% | <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>-1</td><td>0.78</td><td>0.82</td><td>0.80</td><td>9353</td></tr><tr><td>1</td><td>0.73</td><td>0.68</td><td>0.71</td><td>6647</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.76</td><td>16000</td></tr><tr><td>macro avg</td><td>0.76</td><td>0.75</td><td>0.75</td><td>16000</td></tr><tr><td>weighted avg</td><td>0.76</td><td>0.76</td><td>0.76</td><td>16000</td></tr></tbody></table> Confusion matrix: [[7708 1645] [2126 4521]] | | | | | | precision | recall | f1-score | support | -1 | 0.78 | 0.82 | 0.80 | 9353 | 1 | 0.73 | 0.68 | 0.71 | 6647 | accuracy | | | 0.76 | 16000 | macro avg | 0.76 | 0.75 | 0.75 | 16000 | weighted avg | 0.76 | 0.76 | 0.76 | 16000 |
| | precision | recall | f1-score | support | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0.78 | 0.82 | 0.80 | 9353 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0.73 | 0.68 | 0.71 | 6647 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| accuracy | | | 0.76 | 16000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| macro avg | 0.76 | 0.75 | 0.75 | 16000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| weighted avg | 0.76 | 0.76 | 0.76 | 16000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 150.000 elementi del dataset (4c3s), con 80% Training-Set e 20% Test-Set. C = 1 Gamma = 0.1 Random state = 1 | 76.9% | <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>-1</td><td>0.78</td><td>0.83</td><td>0.81</td><td>17362</td></tr><tr><td>1</td><td>0.75</td><td>0.68</td><td>0.71</td><td>12638</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.77</td><td>30000</td></tr><tr><td>macro avg</td><td>0.77</td><td>0.76</td><td>0.76</td><td>30000</td></tr><tr><td>weighted avg</td><td>0.77</td><td>0.77</td><td>0.77</td><td>30000</td></tr></tbody></table> Confusion matrix: [[14430 2932] [3988 8650]] | | | | | | precision | recall | f1-score | support | -1 | 0.78 | 0.83 | 0.81 | 17362 | 1 | 0.75 | 0.68 | 0.71 | 12638 | accuracy | | | 0.77 | 30000 | macro avg | 0.77 | 0.76 | 0.76 | 30000 | weighted avg | 0.77 | 0.77 | 0.77 | 30000 |
| | precision | recall | f1-score | support | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0.78 | 0.83 | 0.81 | 17362 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0.75 | 0.68 | 0.71 | 12638 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| accuracy | | | 0.77 | 30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| macro avg | 0.77 | 0.76 | 0.76 | 30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| weighted avg | 0.77 | 0.77 | 0.77 | 30000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Intero dataset (4c3s), con 80% Training-Set e 20% Test-Set. C = 1 Gamma = 0.1 Random state = 1 | 77.2% | <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>-1</td><td>0.79</td><td>0.83</td><td>0.81</td><td>23596</td></tr><tr><td>1</td><td>0.75</td><td>0.69</td><td>0.72</td><td>16924</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.77</td><td>40520</td></tr><tr><td>macro avg</td><td>0.77</td><td>0.76</td><td>0.76</td><td>40520</td></tr><tr><td>weighted avg</td><td>0.77</td><td>0.77</td><td>0.77</td><td>40520</td></tr></tbody></table> Confusion matrix: [[19628 3968] [5269 11655]] | | | | | | precision | recall | f1-score | support | -1 | 0.79 | 0.83 | 0.81 | 23596 | 1 | 0.75 | 0.69 | 0.72 | 16924 | accuracy | | | 0.77 | 40520 | macro avg | 0.77 | 0.76 | 0.76 | 40520 | weighted avg | 0.77 | 0.77 | 0.77 | 40520 |
| | precision | recall | f1-score | support | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -1 | 0.79 | 0.83 | 0.81 | 23596 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0.75 | 0.69 | 0.72 | 16924 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| accuracy | | | 0.77 | 40520 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| macro avg | 0.77 | 0.76 | 0.76 | 40520 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| weighted avg | 0.77 | 0.77 | 0.77 | 40520 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | |
|---|--|-------|---|
| 4 | Intero dataset (4c3s), con 70% Training-Set e 30% Test-Set. C = 1 Gamma = 0.1 Random state = 1 | 77.1% | <pre> precision recall f1-score support -1 0.79 0.83 0.81 35436 1 0.75 0.69 0.71 25344 accuracy 0.77 60780 macro avg 0.77 0.76 0.76 60780 weighted avg 0.77 0.77 0.77 60780 Confusion matrix: [[29509 5927] [7974 17370]] </pre> |
| 5 | Intero dataset (4c3s), con 70% Training-Set e 30% Test-Set. C = 10 Gamma = auto Random state = 1 | 77.3% | <pre> precision recall f1-score support -1 0.79 0.84 0.81 35436 1 0.75 0.68 0.72 25344 accuracy 0.77 60780 macro avg 0.77 0.76 0.76 60780 weighted avg 0.77 0.77 0.77 60780 Confusion matrix: [[29716 5720] [8059 17285]] </pre> |
| 6 | Intero dataset (4c3s), con 70% Training-Set e 30% Test-Set. C = 10 Gamma = scale Random state = 1 | 77.1% | <pre> precision recall f1-score support -1 0.78 0.85 0.81 35436 1 0.76 0.66 0.71 25344 accuracy 0.77 60780 macro avg 0.77 0.76 0.76 60780 weighted avg 0.77 0.77 0.77 60780 Confusion matrix: [[30108 5328] [8550 16794]] </pre> |
| 7 | Intero dataset (4c3s), con 70% Training-Set e 30% Test-Set. C = 10 Gamma = scale Random state = 1 | 77.1% | <pre> precision recall f1-score support -1 0.78 0.85 0.81 35436 1 0.76 0.66 0.71 25344 accuracy 0.77 60780 macro avg 0.77 0.76 0.76 60780 weighted avg 0.77 0.77 0.77 60780 Confusion matrix: [[30108 5328] [8550 16794]] </pre> |

| | | | |
|----|---|-------|---|
| 8 | Intero dataset (4c4s), con 70% Training-Set e 30% Test-Set. C = 10 Gamma = auto Random state = 1 | 79.9% | <pre> precision recall f1-score support -1 0.81 0.85 0.83 35436 1 0.78 0.72 0.75 25344 accuracy macro avg 0.80 0.79 0.79 60780 weighted avg 0.80 0.80 0.80 60780 Confusion matrix: [[30266 5170] [7029 18315]] </pre> |
| 9 | Intero dataset (4c4s), con 80% Training-Set e 20% Test-Set. C = 10 Gamma = auto Random state = 1 | 80.1% | <pre> precision recall f1-score support -1 0.81 0.85 0.83 35436 1 0.78 0.72 0.75 25344 accuracy macro avg 0.80 0.79 0.79 60780 weighted avg 0.80 0.80 0.80 60780 Confusion matrix: [[30266 5170] [7029 18315]] </pre> |
| 10 | Intero dataset (5c4s), con 70% Training-Set e 30% Test-Set. C = 10 Gamma = auto Random state = 1 | 57% | <pre> precision recall f1-score support -1 0.58 0.93 0.72 35436 1 0.41 0.07 0.12 25344 accuracy macro avg 0.50 0.50 0.42 60780 weighted avg 0.51 0.57 0.47 60780 Confusion matrix: [[32979 2457] [23619 1725]] </pre> |
| 11 | Intero dataset (5c4s), con 80% Training-Set e 20% Test-Set. C = 10 Gamma = auto Random state = 1 | 57.2% | <pre> precision recall f1-score support -1 0.58 0.94 0.72 23596 1 0.42 0.06 0.11 16924 accuracy macro avg 0.50 0.50 0.41 40520 weighted avg 0.51 0.57 0.46 40520 Confusion matrix: [[22171 1425] [15900 1024]] </pre> |

| | | | | | | | |
|----|---|-------|--|------|------|------|----------------------|
| 12 | Dataset bilanciato (4c4s), con 70% Training-Set e 30% Test-Set. C = 10 Gamma = auto Random state = 1 | 79.2% | precision recall f1-score support | | | | |
| | | | -1 | 0.79 | 0.80 | 0.79 | 25183 |
| | | | 1 | 0.80 | 0.78 | 0.79 | 25478 |
| | | | accuracy | | | | 0.79 50661 |
| | | | macro avg | | | | 0.79 0.79 0.79 50661 |
| | | | weighted avg | | | | 0.79 0.79 0.79 50661 |
| | | | Confusion matrix: [[20167 5016] [5514 19964]] | | | | |

5.2 Tabella riassuntiva risultati Decision Tree

| #Test | Configurazione | Accuratezza |
|-------|--|-------------|
| 1 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = DEFAULT (ottimale) | 67.8% |
| 2 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 5 | 66.0% |
| 3 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 6 | 67.2% |
| 4 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 7 | 68.4% |
| 5 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 8 | 69.2% |
| 6 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 9 | 70.0% |
| 7 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 10 | 70.9% |

| | | |
|----|---|-------|
| 8 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 11 | 71.3% |
| 9 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 12 | 71.6% |
| 10 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 13 | 71.9% |
| 11 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 14 | 71.9% |
| 12 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = DEFAULT (ottimale) | 67.8% |
| 13 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 5 | 66.0% |
| 14 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 6 | 67.2% |
| 15 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 7 | 68.4% |
| 16 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 8 | 69.2% |
| 17 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 9 | 70.1% |
| 18 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 10 | 70.8% |

| | | |
|----|--|-------|
| 19 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 11 | 71.3% |
| 20 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 12 | 71.5% |
| 21 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 13 | 71.5% |
| 22 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = DEFAULT (ottimale) | 69.0% |
| 23 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 5 | 67.1% |
| 24 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 6 | 68.7% |
| 25 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 7 | 69.8% |
| 26 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 8 | 70.5% |
| 27 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 9 | 71.8% |
| 28 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 10 | 72.6% |
| 29 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 11 | 73.3% |

| | | |
|-----------|---|-------|
| 30 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 12 | 73.5% |
| 31 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 13 | 73.8% |
| 32 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 14 | 73.6% |
| 33 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = DEFAULT (ottimale) | 69.7% |
| 34 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 5 | 67.9% |
| 35 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 6 | 69.3% |
| 36 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 7 | 69.9% |
| 37 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 8 | 71.3% |
| 38 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 9 | 72.1% |
| 39 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 10 | 73.0% |
| 40 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 11 | 73.3% |

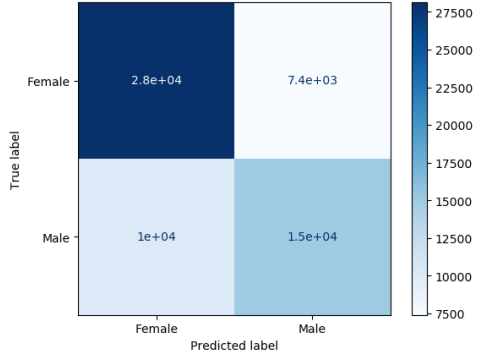
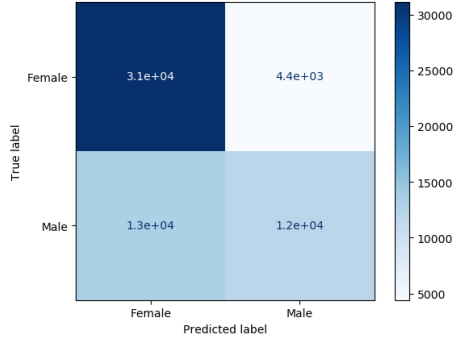
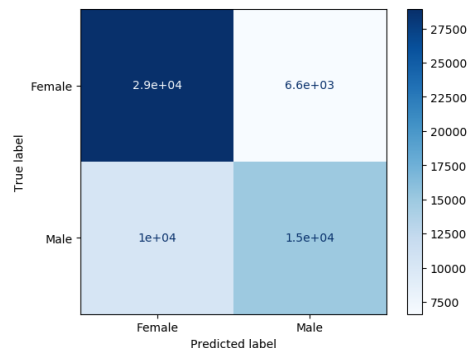
| | | |
|----|---|-------|
| 41 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 12 | 73.7% |
| 42 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 13 | 73.7% |
| 43 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = DEFAULT (ottimale) | 51.3% |
| 44 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 5 | 58.4% |
| 45 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 6 | 58.4% |
| 46 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 7 | 58.4% |
| 47 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = Gini max_depth = 8 | 58.3% |
| 48 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = DEFAULT (ottimale) | 51.4% |
| 49 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 5 | 58.4% |
| 50 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 6 | 58.4% |
| 51 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 7 | 58.4% |

| | | |
|----|--|-------|
| 52 | Intero dataset (5c3s) 70% Training-Set e 30% Test-Set criterion = entropy max_depth = 8 | 58.3% |
| 53 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set criterion = gini max_depth = DEFAULT (ottimale) | 68.8% |
| 54 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set criterion = gini max_depth = 5 | 65.5% |
| 55 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set criterion = gini max_depth = 6 | 67.7% |
| 56 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set criterion = gini max_depth = 7 | 68.7% |
| 57 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set criterion = gini max_depth = 8 | 69.9% |
| 58 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set criterion = gini max_depth = 9 | 70.6% |
| 59 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = gini max_depth = 10 | 71.4% |
| 60 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = gini max_depth = 11 | 72.0% |
| 61 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = gini max_depth = 12 | 72.3% |
| 62 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = gini max_depth = 13 | 72.3% |

| | | |
|-----------|---|-------|
| 63 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = DEFAULT (ottimale) | 68.9% |
| 64 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 5 | 64.8% |
| 65 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 6 | 66.5% |
| 66 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 7 | 67.9% |
| 67 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 8 | 69.3% |
| 68 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 9 | 70.1% |
| 69 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 10 | 70.8% |
| 70 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 11 | 71.3% |
| 71 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 12 | 71.7% |
| 72 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 13 | 71.9% |
| 73 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 14 | 72.1% |

| | | |
|----|---|-------|
| 74 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set Criterion = entropy max_depth = 15 | 71.9% |
|----|---|-------|

5.3 Tabella riassuntiva risultati K-Nearest Neighbors

| #Test | Configurazione | Accuratezza | Output |
|-------|--|-------------|--|
| 1 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=3 | 71.0% | <p>Confusion Matrix</p>  |
| 2 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=4 | 70.9% | <p>Confusion Matrix</p>  |
| 3 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=5 | 72.2% | <p>Confusion Matrix</p>  |

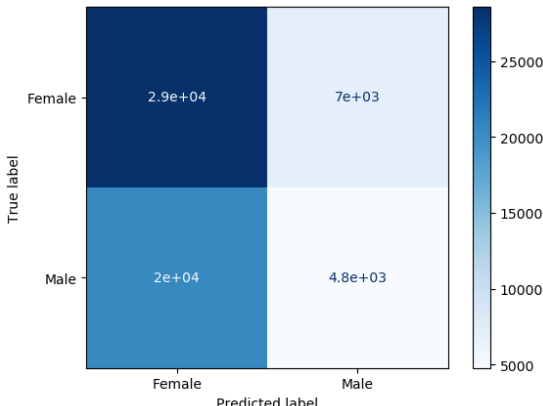
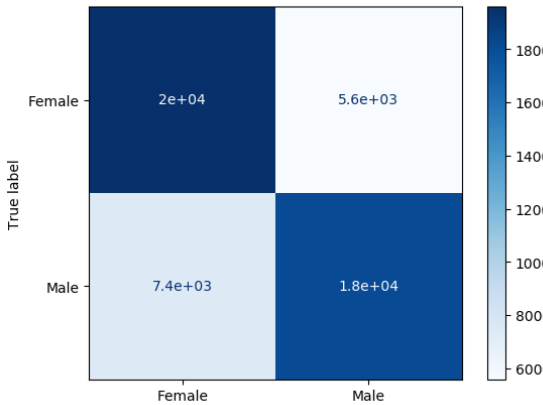
| 4 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=6 | 72.0% | <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p> <table><tr><th></th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3.1e+04</td><td>4.5e+03</td></tr><tr><th>Male</th><td>1.2e+04</td><td>1.3e+04</td></tr></table> | | Female | Male | Female | 3.1e+04 | 4.5e+03 | Male | 1.2e+04 | 1.3e+04 |
|--------|--|---------|--|--|--------|------|--------|---------|---------|------|---------|---------|
| | Female | Male | | | | | | | | | | |
| Female | 3.1e+04 | 4.5e+03 | | | | | | | | | | |
| Male | 1.2e+04 | 1.3e+04 | | | | | | | | | | |
| 5 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=7 | 72.9% | <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p> <table><tr><th></th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.9e+04</td><td>6.2e+03</td></tr><tr><th>Male</th><td>1e+04</td><td>1.5e+04</td></tr></table> | | Female | Male | Female | 2.9e+04 | 6.2e+03 | Male | 1e+04 | 1.5e+04 |
| | Female | Male | | | | | | | | | | |
| Female | 2.9e+04 | 6.2e+03 | | | | | | | | | | |
| Male | 1e+04 | 1.5e+04 | | | | | | | | | | |
| 6 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=8 | 72.5% | <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p> <table><tr><th></th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3.1e+04</td><td>4.5e+03</td></tr><tr><th>Male</th><td>1.2e+04</td><td>1.3e+04</td></tr></table> | | Female | Male | Female | 3.1e+04 | 4.5e+03 | Male | 1.2e+04 | 1.3e+04 |
| | Female | Male | | | | | | | | | | |
| Female | 3.1e+04 | 4.5e+03 | | | | | | | | | | |
| Male | 1.2e+04 | 1.3e+04 | | | | | | | | | | |
| 7 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=9 | 73.2% | <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p> <table><tr><th></th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3e+04</td><td>5.8e+03</td></tr><tr><th>Male</th><td>1e+04</td><td>1.5e+04</td></tr></table> | | Female | Male | Female | 3e+04 | 5.8e+03 | Male | 1e+04 | 1.5e+04 |
| | Female | Male | | | | | | | | | | |
| Female | 3e+04 | 5.8e+03 | | | | | | | | | | |
| Male | 1e+04 | 1.5e+04 | | | | | | | | | | |

| 8 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=10 | 72.9% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3.1e+04</td><td>4.6e+03</td></tr><tr><th>Male</th><td>1.2e+04</td><td>1.3e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 3.1e+04 | 4.6e+03 | Male | 1.2e+04 | 1.3e+04 |
|------------------------------|---|---------|---|------------------------------|--------|------|--------|---------|---------|------|---------|---------|
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 3.1e+04 | 4.6e+03 | | | | | | | | | | |
| Male | 1.2e+04 | 1.3e+04 | | | | | | | | | | |
| 9 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=3 | 72.7% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.8e+04</td><td>7.1e+03</td></tr><tr><th>Male</th><td>9.4e+03</td><td>1.6e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.8e+04 | 7.1e+03 | Male | 9.4e+03 | 1.6e+04 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.8e+04 | 7.1e+03 | | | | | | | | | | |
| Male | 9.4e+03 | 1.6e+04 | | | | | | | | | | |
| 10 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=4 | 72.5% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3.1e+04</td><td>4.3e+03</td></tr><tr><th>Male</th><td>1.2e+04</td><td>1.3e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 3.1e+04 | 4.3e+03 | Male | 1.2e+04 | 1.3e+04 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 3.1e+04 | 4.3e+03 | | | | | | | | | | |
| Male | 1.2e+04 | 1.3e+04 | | | | | | | | | | |
| 11 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=5 | 73.8% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.9e+04</td><td>6.5e+03</td></tr><tr><th>Male</th><td>9.4e+03</td><td>1.6e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.9e+04 | 6.5e+03 | Male | 9.4e+03 | 1.6e+04 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.9e+04 | 6.5e+03 | | | | | | | | | | |
| Male | 9.4e+03 | 1.6e+04 | | | | | | | | | | |

| 12 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=6 | 73.7% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3.1e+04</td><td>4.4e+03</td></tr><tr><th>Male</th><td>1.2e+04</td><td>1.4e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 3.1e+04 | 4.4e+03 | Male | 1.2e+04 | 1.4e+04 |
|------------------------------|--|---------|---|------------------------------|--------|------|--------|---------|---------|------|---------|---------|
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 3.1e+04 | 4.4e+03 | | | | | | | | | | |
| Male | 1.2e+04 | 1.4e+04 | | | | | | | | | | |
| 13 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=7 | 74.5% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3e+04</td><td>6e+03</td></tr><tr><th>Male</th><td>9.5e+03</td><td>1.6e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 3e+04 | 6e+03 | Male | 9.5e+03 | 1.6e+04 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 3e+04 | 6e+03 | | | | | | | | | | |
| Male | 9.5e+03 | 1.6e+04 | | | | | | | | | | |
| 14 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=8 | 74.2% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3.1e+04</td><td>4.4e+03</td></tr><tr><th>Male</th><td>1.1e+04</td><td>1.4e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 3.1e+04 | 4.4e+03 | Male | 1.1e+04 | 1.4e+04 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 3.1e+04 | 4.4e+03 | | | | | | | | | | |
| Male | 1.1e+04 | 1.4e+04 | | | | | | | | | | |
| 15 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=9 | 74.9% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3e+04</td><td>5.7e+03</td></tr><tr><th>Male</th><td>9.5e+03</td><td>1.6e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 3e+04 | 5.7e+03 | Male | 9.5e+03 | 1.6e+04 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 3e+04 | 5.7e+03 | | | | | | | | | | |
| Male | 9.5e+03 | 1.6e+04 | | | | | | | | | | |

| 16 | Intero dataset (4c4s) 70% Training-Set e 30% Test-Set n_neighbors=10 | 74.6% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>3.1e+04</td><td>4.4e+03</td></tr><tr><th>Male</th><td>1.1e+04</td><td>1.4e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 3.1e+04 | 4.4e+03 | Male | 1.1e+04 | 1.4e+04 |
|------------------------------|---|---------|---|------------------------------|--------|------|--------|---------|---------|------|---------|---------|
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 3.1e+04 | 4.4e+03 | | | | | | | | | | |
| Male | 1.1e+04 | 1.4e+04 | | | | | | | | | | |
| 17 | Intero dataset (4c3s) 70% Training-Set e 30% Test-Set n_neighbors=3 | 54.6% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.2e+04</td><td>1.3e+04</td></tr><tr><th>Male</th><td>1.6e+04</td><td>9.3e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.2e+04 | 1.3e+04 | Male | 1.6e+04 | 9.3e+03 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.2e+04 | 1.3e+04 | | | | | | | | | | |
| Male | 1.6e+04 | 9.3e+03 | | | | | | | | | | |
| 18 | Intero dataset (5c4s) 70% Training-Set e 30% Test-Set n_neighbors=4 | 51.8% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.8e+04</td><td>7.2e+03</td></tr><tr><th>Male</th><td>2e+04</td><td>4.9e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.8e+04 | 7.2e+03 | Male | 2e+04 | 4.9e+03 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.8e+04 | 7.2e+03 | | | | | | | | | | |
| Male | 2e+04 | 4.9e+03 | | | | | | | | | | |
| 19 | Intero dataset (5c4s) 70% Training-Set e 30% Test-Set n_neighbors=5 | 52.0% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.2e+04</td><td>1.3e+04</td></tr><tr><th>Male</th><td>1.6e+04</td><td>9.1e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.2e+04 | 1.3e+04 | Male | 1.6e+04 | 9.1e+03 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.2e+04 | 1.3e+04 | | | | | | | | | | |
| Male | 1.6e+04 | 9.1e+03 | | | | | | | | | | |

| 20 | Intero dataset (5c4s) 70% Training-Set e 30% Test-Set n_neighbors=6 | 54.7% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.8e+04</td><td>7.3e+03</td></tr><tr><th>Male</th><td>2e+04</td><td>5e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.8e+04 | 7.3e+03 | Male | 2e+04 | 5e+03 |
|------------------------------|--|---------|---|------------------------------|--------|------|--------|---------|---------|------|---------|---------|
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.8e+04 | 7.3e+03 | | | | | | | | | | |
| Male | 2e+04 | 5e+03 | | | | | | | | | | |
| 21 | Intero dataset (5c4s) 70% Training-Set e 30% Test-Set n_neighbors=7 | 52.2% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.3e+04</td><td>1.2e+04</td></tr><tr><th>Male</th><td>1.7e+04</td><td>8.5e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.3e+04 | 1.2e+04 | Male | 1.7e+04 | 8.5e+03 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.3e+04 | 1.2e+04 | | | | | | | | | | |
| Male | 1.7e+04 | 8.5e+03 | | | | | | | | | | |
| 22 | Intero dataset (5c4s) 70% Training-Set e 30% Test-Set n_neighbors=8 | 54.6% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.8e+04</td><td>7.2e+03</td></tr><tr><th>Male</th><td>2e+04</td><td>4.9e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.8e+04 | 7.2e+03 | Male | 2e+04 | 4.9e+03 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.8e+04 | 7.2e+03 | | | | | | | | | | |
| Male | 2e+04 | 4.9e+03 | | | | | | | | | | |
| 23 | Intero dataset (5c4s) 70% Training-Set e 30% Test-Set n_neighbors=9 | 53.1% | <p>Confusion Matrix</p> <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.5e+04</td><td>1.1e+04</td></tr><tr><th>Male</th><td>1.8e+04</td><td>7.5e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.5e+04 | 1.1e+04 | Male | 1.8e+04 | 7.5e+03 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.5e+04 | 1.1e+04 | | | | | | | | | | |
| Male | 1.8e+04 | 7.5e+03 | | | | | | | | | | |

| 24 | Intero dataset (5c4s) 70% Training-Set e 30% Test-Set n_neighbors=10 | 54.8% | <p>Confusion Matrix</p>  <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2.9e+04</td><td>7e+03</td></tr><tr><th>Male</th><td>2e+04</td><td>4.8e+03</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2.9e+04 | 7e+03 | Male | 2e+04 | 4.8e+03 |
|------------------------------|--|---------|---|------------------------------|--------|------|--------|---------|---------|------|---------|---------|
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2.9e+04 | 7e+03 | | | | | | | | | | |
| Male | 2e+04 | 4.8e+03 | | | | | | | | | | |
| 24 | Dataset bilanciato (4c4s) 70% Training-Set e 30% Test-Set n_neighbors = 9 | 74.3% | <p>Confusion Matrix</p>  <table><tr><th>True label \ Predicted label</th><th>Female</th><th>Male</th></tr><tr><th>Female</th><td>2e+04</td><td>5.6e+03</td></tr><tr><th>Male</th><td>7.4e+03</td><td>1.8e+04</td></tr></table> | True label \ Predicted label | Female | Male | Female | 2e+04 | 5.6e+03 | Male | 7.4e+03 | 1.8e+04 |
| True label \ Predicted label | Female | Male | | | | | | | | | | |
| Female | 2e+04 | 5.6e+03 | | | | | | | | | | |
| Male | 7.4e+03 | 1.8e+04 | | | | | | | | | | |



6. Bibliografia

- [1] P. Barra, S. Barra, C. Bisogni, M. De Marsico, M. Nappi. “Web-Shaped Model for Head Pose Estimation: An Approach for Best Exemplar Selection”, pp. 5457 – 5468.
- [2] https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- [3] Appunti del seminario sui classificatori nel corso di FVAB 2019/2020 - SeminarioClassificatori.pdf
- [4] <https://dl.acm.org/doi/abs/10.1145/1068009.1068160> - Evolutionary strategies for multi-scale radial basis function kernels in support vector machines
- [5] <https://www.datacamp.com/community/tutorials/decision-tree-classification-python>
- [6] <https://blog.quantinsti.com/gini-index/>
- [7] https://it.wikipedia.org/wiki/K-nearest_neighbors