

# Foot classifier

## Progetto IC

Luigi Lomasto, Marco Mecchia

29 Febbraio 2016

Prof. Roberto Tagliaferri

Dott. Michele Fratello

Dott. Paola Galdi

Dott. Angela Serra

# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori
- 6 Infrastruttura
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni

# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori
- 6 Infrastruttura
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni

# Analisi posturale statica

L'analisi posturale statica é un esame in grado di diagnosticare diversi disturbi della postura.

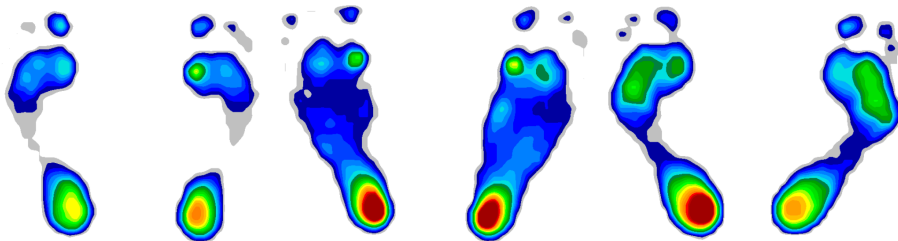
- Scoliosi.
- Cifosi.
- Patologie del piede.
- Corpo sbilanciato.
- etc.

# Il problema

Dato un insieme di immagini piedebarometriche, classificare automaticamente le seguenti patologie:

- Cavo
- Piatto
- Normale
- Valgo

# Patologie



# Struttura del dataset

Il dataset è costituito da 190 piedi (95 coppie) di cui:

- 121 Cavi
- 13 Piatti
- 56 Normali

per la prima classe di patologie, mentre:

- 88 Valghi
- 102 Normali

per la seconda classe di patologie.

# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori
- 6 Infrastruttura
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni

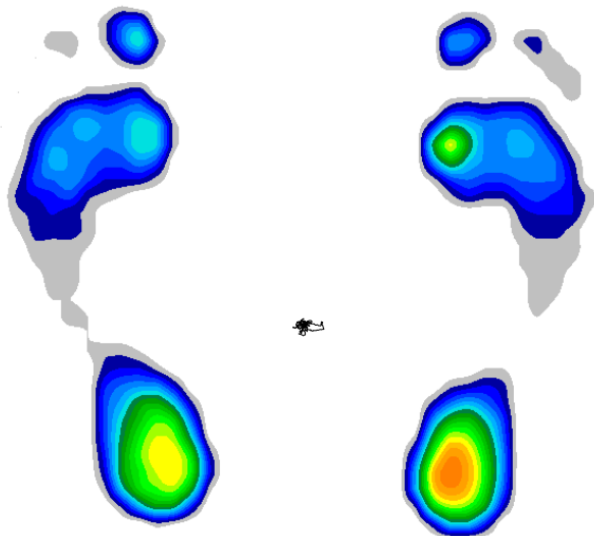


# Panoramica

Nella fase di preprocessing, abbiamo eseguito le seguenti operazioni.

- Conversione delle immagini .bmp in .png.
- Pulizia delle immagini.
- Trasformazione delle immagini in scala di grigio.
- Divisione, rotazione e cropping dei piedi.

# Rimozione del tracciato del baricentro



# Trasformazione delle immagini

- Trasformazione da uno spazio 3D (R,G,B) ad uno spazio 1D.
- Necessaria per valutare la pressione in un pixel dell'immagine.
- Problema: Come associare ad ogni tripla (R,G,B) il valore di intensità corretto?

# Prima soluzione

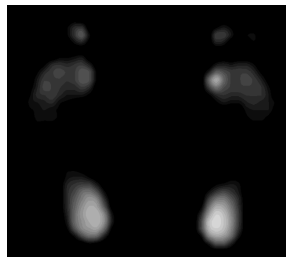
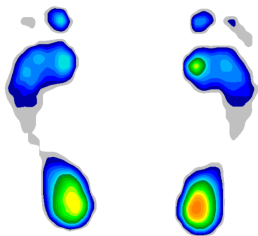
- Clustering sulle immagini.
- Trovati i centroidi, sostituire ogni pixel dell'immagine con il rappresentante del proprio cluster.
- Ordinare i centroidi per intensità crescente ed associare valori compresi tra 0 e 1.
- Problemi: Trovare il  $k$  adatto, trovare un ordinamento per i cluster.

# Soluzione definitiva

- Variante della prima soluzione.
- Possibile grazie alla scala colorata delle pressioni fornita dal medico.



# Effetto della conversione



# Rotazione e cropping delle immagini

- La divisione é stata necessaria per separare piede destro da piede sinistro.
- La rotazione non é strettamente necessaria.
- Il cropping rimuove la parte di sfondo superflua.

# Rotazione del piede

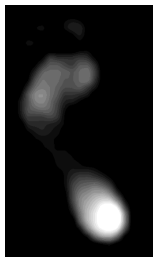
Abbiamo utilizzato l'algoritmo classico per la rotazione di una retta, cioè:

- 1 Trovare il punto dove applicare la rotazione.
- 2 Calcolare il coefficiente angolare della retta.
- 3 Applicare la rotazione.

Nel nostro caso, abbiamo usato la retta che congiunge i due punti di massima pressione, ed il punto più esterno in basso del tallone.



# Effetto della rotazione



# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori
- 6 Infrastruttura
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni

# Feature extraction

- Necessaria per via della natura delle immagini.
- Normalizzazione in uno spazio standard non nota in letteratura.

Dovendo lavorare con due classificatori:

- Cavo vs piatto vs normale.
- Valgo vs varo vs normale.

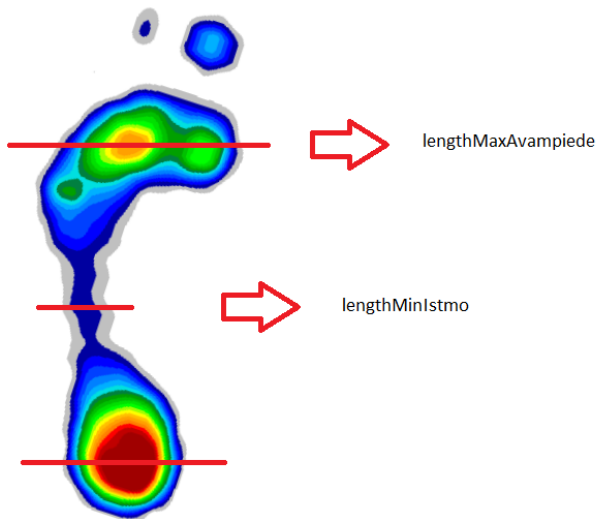
sono stati implementati due algoritmi ad hoc per estrarre le caratteristiche diverse.

## Features primo classificatore (1/2)

Le features estratte per le patologie appartenenti alla prima classe sono le seguenti:

- **lengthMinIstmo**: esprime la lunghezza minima che assume l'istmo.
- **lengthMedialstmo**: esprime la lunghezza media dell'istmo.
- **lengthMaxAvampiede**: esprime la massima lunghezza che assume l'avampiede.
- **indexPathology**: Si ottiene dal rapporto  $\frac{\text{lengthMaxAvampiede}}{\text{lengthMinIstmo}}$
- **mediumPressure**: Indica la pressione media esercitata dal piede.

## Features primo classificatore (2/2)

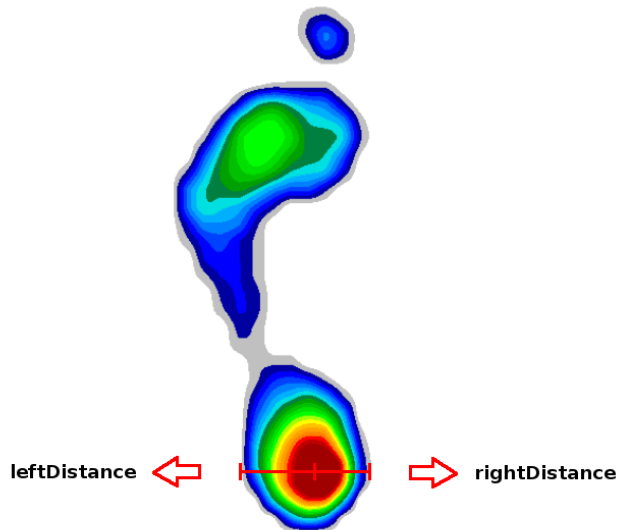


## Features secondo classificatore (1/2)

Le features estratte per le patologie appartenenti alla seconda classe sono le seguenti:

- **approssimated**: vale 0 se la differenza tra il punto di massima pressione del tallone ed il centro del tallone  $\leq 10$ .
- **leftDistance**: esprime la distanza del punto di massima pressione del tallone dall'estremo sinistro.
- **rightDistance**: esprime la distanza del punto di massima pressione del tallone dall'estremo destro.

# Features secondo classificatore (2/2)



# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection**
- 5 Scelta dei classificatori
- 6 Infrastruttura
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni



# Feature Selection

La features selection è stata fatta in modo esaustivo, utilizzando le performance del classificatore come metro di giudizio.

Motivi:

- Basso numero di features usate.
- Si valutano tutti i possibili sottoinsiemi di features.

Sottoinsiemi valutati:

- 31 per il primo classificatore.
- 7 per il secondo classificatore.

# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori**
- 6 Infrastruttura
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni

# Classificatori

## SVM Gaussiano (Radial-Basis-Function).

Scelta dovuta ai seguenti motivi :

- RBF ben si adatta a problemi di classificazione dove il dataset è significativamente più grande rispetto al numero di dimensioni.
- Reti Neurali troppo complesse per il problema affrontato.
- SVM lineare generalmente usato quando si ha alta dimensionalità.

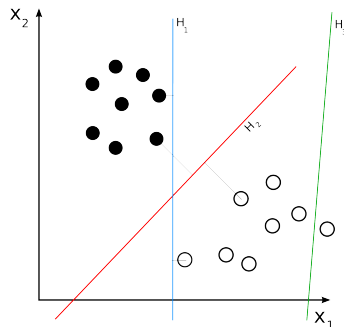
## SVM Lineare.

Motivo:

- Per la prima classe di problemi abbiamo individuato delle features con le quali è possibile generare un iperpiano di separazione.

# SVM Lineare

- Un classificatore SVM costruisce un iperpiano (o insieme di iperpiani) che può essere usato per fare classificazione.
- Una buona separazione è data dall'iperpiano che massimizza la distanza dal punto dei dati più vicino

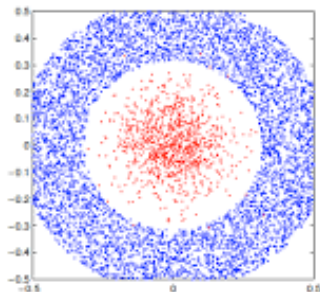


# Kernel

Spesso si ha a che fare con classi non separabili linearmente.

**Kernel:** Rendono utilizzabili modelli lineari con dati non linearmente separabili

- Mapping dei dati in dimensioni più alte
- Applicano un modello lineare nel nuovo spazio



# Kernel RBF (Radial-Basis-Function)

**Kernel RBF:**  $k(x, z) = \exp(-\gamma \|x_i - x\|^2)$

Per scegliere la configurazione ottimale abbiamo testato tutte le possibili combinazioni con:

- $10^{-3} < C < 10^3$
- $10^{-2} < \gamma < 10^0$

# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori
- 6 Infrastruttura**
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni

# Infrastruttura

L'infrastruttura progettata è la seguente:

- $\forall$  Sottoinsieme di features
  - Per N volte
    - Dividiamo il dataset in training set e test set.
    - Appliciamo cross validation sul training set per stimare la configurazione migliore.
    - Calcolo delle accuratezze sul test.
  - Calcolo delle accuratezze medie.

Il sottoinsieme scelto è quello con accuratezza totale media più alta.



# Scelta della migliore configurazione: T-test

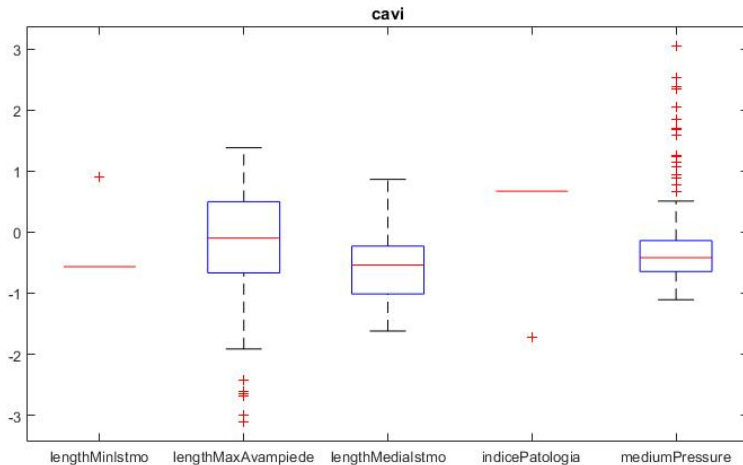
Per valutare la migliore configurazione ci siamo serviti del T-test.

- Per ogni configurazione abbiamo ottenuto una distribuzione di accuratezze.
- T-test tra ogni configurazione e la successiva per studiarne la significatività.
- Se c'è significatività, (differenza non casuale) si prende la configurazione con accuratezza media più alta.

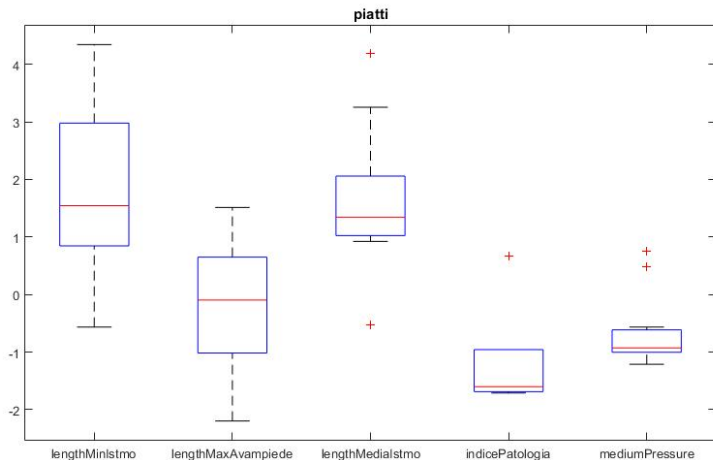
# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori
- 6 Infrastruttura
- 7 Risultati**
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni

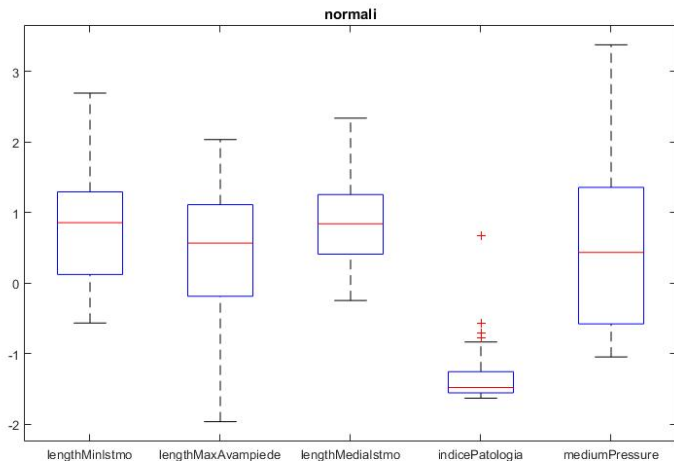
# BoxPlot delle features per i piedi Cavi



# BoxPlot delle features per i piedi Piatti



# BoxPlot delle features per i piedi Normali



# Risultati SVM lineare sottoinsieme migliore

## PRIMO CLASSIFICATORE

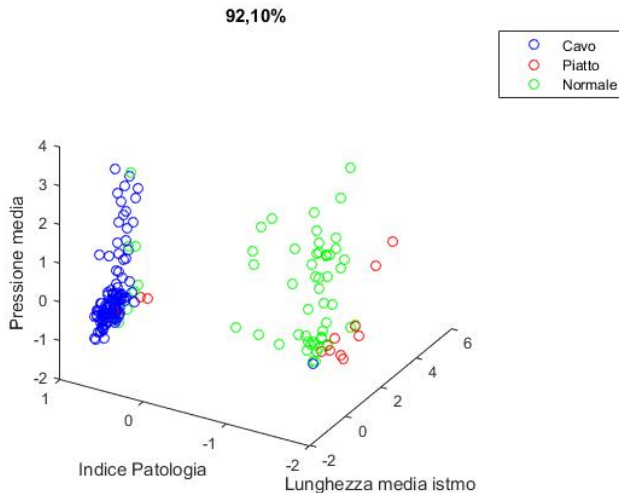
**Features:** lengthMedialstmo, indicePatologia, mediumPressure

**Numero fold:** 5

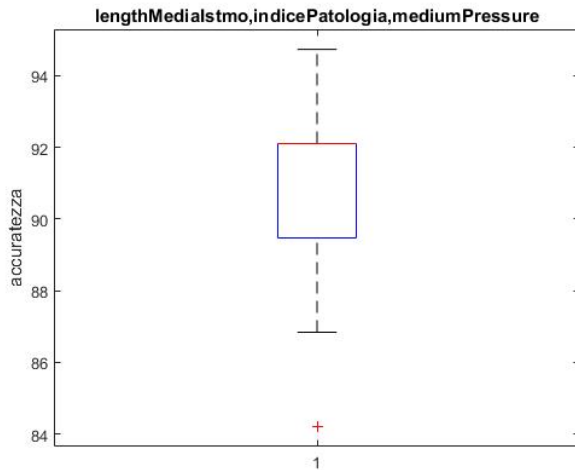
**Numero piedi per fold:** 24 cavi, 2 piatti, 12 normali.

- Accuratezza totale: 92.14%
- Accuratezza prima classe: 99.33%
- Accuratezza seconda classe: 2%
- Accuratezza terza classe: 86.50%
- C:0.01

# Risultati SVM lineare sottoinsieme migliore



# Risultati SVM lineare sottoinsieme migliore





# Risultati SVM lineare sottoinsieme peggiore

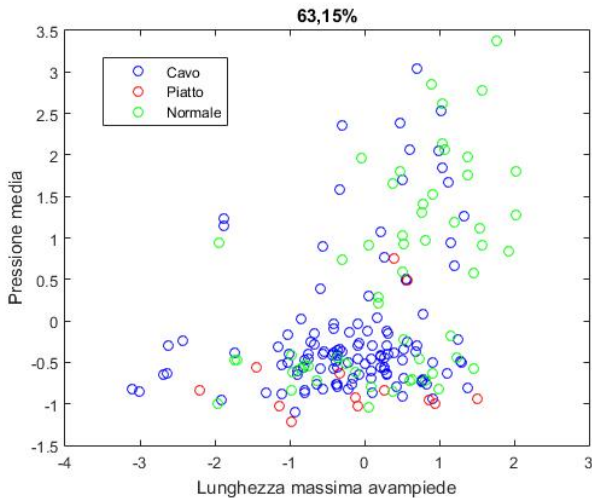
**Features:** lengthMaxAvampiede, mediumPressure

**Numero fold:** 5

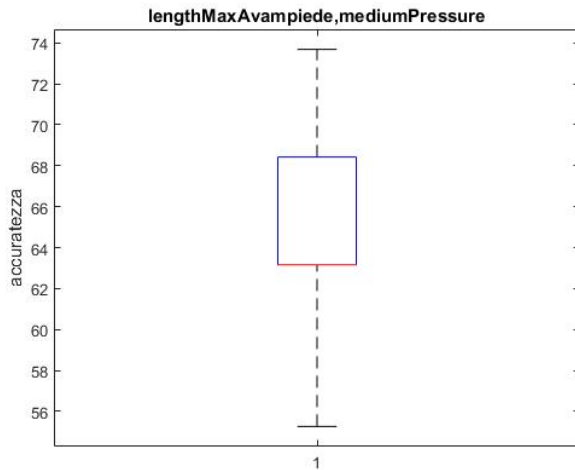
**Numero piedi per fold:** 24 cavi, 2 piatti, 12 normali.

- **Accuratezza totale:** 63,15
- **Accuratezza prima classe:** 91,5
- **Accuratezza seconda classe:** 0
- **Accuratezza terza classe:** 17,83
- **C:1**

# Risultati SVM lineare sottoinsieme peggiore



# Risultati SVM lineare sottoinsieme peggiore



# Risultati SVM lineare con tutte le features

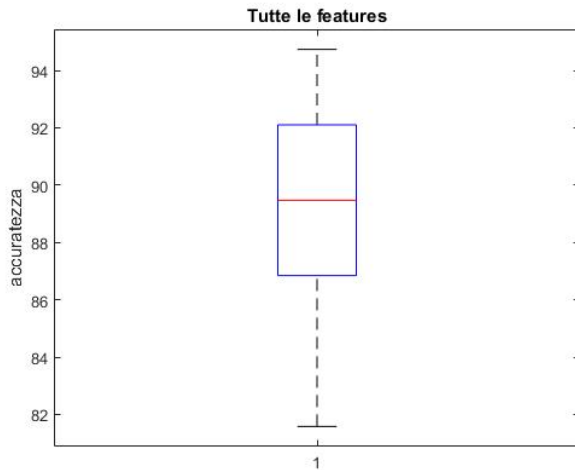
**Features:** tutte

**Numero fold:** 5

**Numero piedi per fold:** 24 cavi, 2 piatti, 12 normali.

- **Accuratezza totale:** 89,47%
- **Accuratezza prima classe:** 99,25%
- **Accuratezza seconda classe:** 0%
- **Accuratezza terza classe:** 84,5%
- **C:**0.01

# Risultati SVM lineare con tutte le features



# Risultati kernel RBF con features migliori

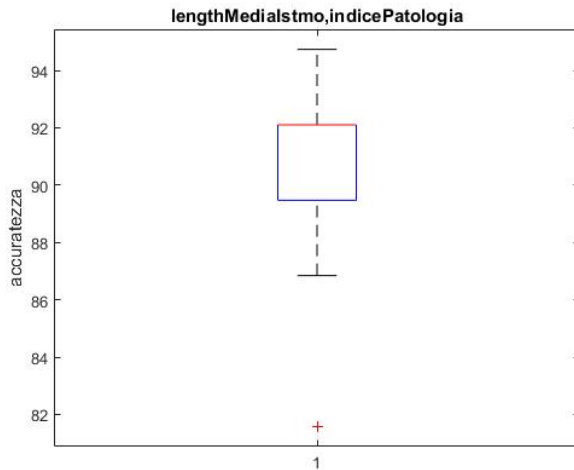
**Features:** lengthMedialstmo, indicePatologia

**Numero fold:** 5

**Numero piedi per fold:** 24 cavi, 2 piatti, 12 normali.

- **Accuratezza totale:** 92,63%
- **Accuratezza prima classe:** 100%
- **Accuratezza seconda classe:** 0%
- **Accuratezza terza classe:** 93,3%
- **C:**0.01
- **Gamma:**0.01

# Risultati kernel RBF con features migliori



# Risultati kernel RBF con tutte le features

**Features:** lengthMedialstmo, indicePatologia

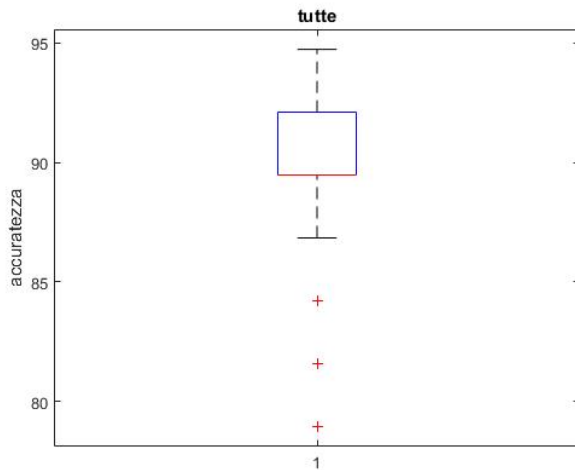
**Numero fold:** 5

**Numero piedi per fold:** 24 cavi, 2 piatti, 12 normali.

- **Accuratezza totale:** 89.47%
- **Accuratezza prima classe:** 99.16%
- **Accuratezza seconda classe:** 0%
- **Accuratezza terza classe:** 85%
- **C:**1
- **Gamma:**0.01



# Risultati kernel RBF con tutte le features



# Risultati SVM lineare con features migliori

## SECONDO CLASSIFICATORE

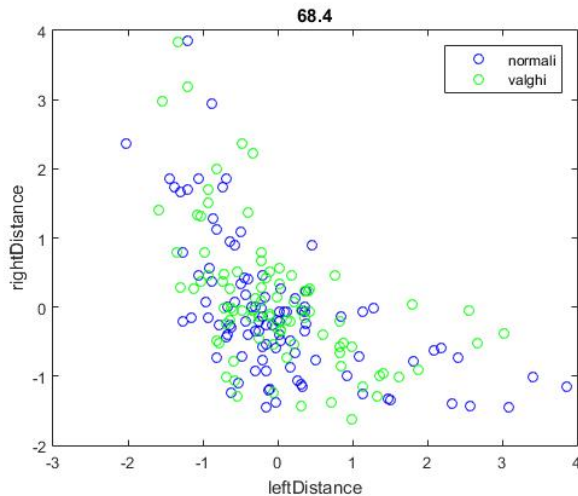
**Features:** leftDistance, rightDistance

**Numero fold:** 10

**Numero piedi per fold:** 17 Valghi, 21 Normali

- **Accuratezza totale:** 68.4%
- **Accuratezza prima classe:** 37.5%
- **Accuratezza seconda classe:** 90.9%
- **C:**10

# Risultati SVM lineare con features migliori



# Risultati kernel RBF con features migliori

## SECONDO CLASSIFICATORE

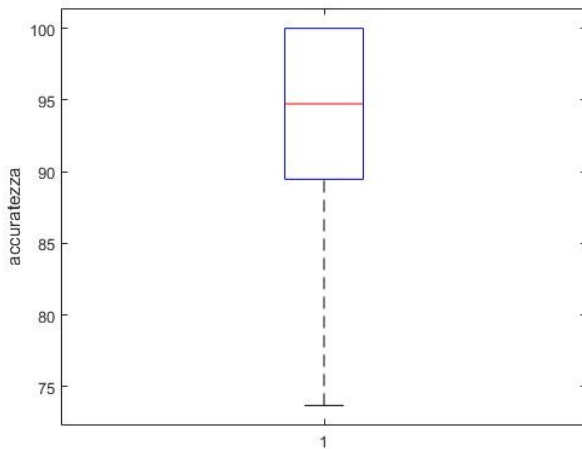
**Features:** leftDistance, rightDistance

**Numero fold:** 10

**Numero piedi per fold:** 17 Valghi, 21 Normali

- **Accuratezza totale:** 94.4%
- **Accuratezza prima classe:** 90.5%
- **Accuratezza seconda classe:** 97.45%
- **C:**100
- **Gamma:**0.1

# Risultati kernel RBF con features migliori



# Risultati kernel RBF con tutte le features

## SECONDO CLASSIFICATORE

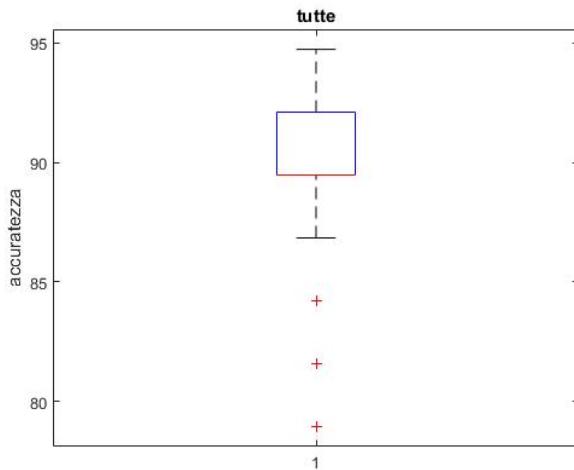
**Features:** tutte

**Numero fold:** 10

**Numero piedi per fold:** 17 Valghi, 21 Normali

- **Accuratezza totale:** 89.05%
- **Accuratezza prima classe:** 84.25%
- **Accuratezza seconda classe:** 95.81%
- **C:**1
- **Gamma:**1

# Risultati kernel RBF con tutte le features



# Outline

- 1 Introduzione al problema
- 2 Preprocessing
  - Pulizia delle immagini
  - Trasformazione delle immagini
  - Divisione, rotazione e cropping delle immagini
- 3 Feature extraction
  - Features primo classificatore
  - Features secondo classificatore
- 4 Feature Selection
- 5 Scelta dei classificatori
- 6 Infrastruttura
- 7 Risultati
  - Primo classificatore
  - Secondo classificatore
- 8 Conclusioni



# Conclusioni

- Le feature estratte hanno separato abbastanza bene tutte le tipologie di piedi (tranne i piatti).
- Il kernel RBF generalmente si é comportato meglio, in entrambi i problemi.
- Il kernel lineare ha avuto buone percentuali solo sul primo problema.

# Sviluppi futuri

Il classificatore ha avuto ottime percentuali di accuratezza in generale.  
Possibili miglioramenti includono:

- Normalizzazione del piede e uso delle reti neurali.
- Trovare nuove feature in grado di separare meglio i piedi piatti dai normali(?).
- Includere quanto fatto in un framework omnicomprensivo degli altri tipi di analisi.

Grazie per l'attenzione!