Università degli Studi di Milano-Bicocca

2022/2023

*Dipartimento di Informatica, Sistemistica e Comunicazione*

*Corso di Laurea Magistrale in Informatica*

Classificazione delle specie di bradipi

Progetto Machine Learning

**Membri del team:**

Matteo Rondena 847381

Luca Loddo 844529

Indice

[Introduzione 3](#_Toc127134422)

[Obiettivo 3](#_Toc127134423)

[Scelta del dataset 3](#_Toc127134424)

[Analisi e pre-process sui dati 4](#_Toc127134425)

[Preparazione del dataset 5](#_Toc127134426)

[Analisi delle features 6](#_Toc127134427)

[Modelli di Machine Learning 15](#_Toc127134428)

[Decision Tree 15](#_Toc127134429)

[Support Vector Machine (SVM) 18](#_Toc127134430)

[Stima delle misure di performance 20](#_Toc127134431)

[Performance Decision Tree 20](#_Toc127134432)

[Performance Support Vector Machine (SVM) 23](#_Toc127134433)

[Confronto tra modelli 25](#_Toc127134434)

[Conclusioni 27](#_Toc127134435)

# **Introduzione**

Per il progetto d’esame di Machine Learning è stato deciso di utilizzare un dataset composto dai dati di misurazione, rilevati in natura, su diversi esemplari di bradipi. In questo progetto sono stati utilizzati due modelli di machine learning, ovvero Support Vector Machine e Decision Tree, per poter determinare la precisa sottospecie di appartenenza dei bradipi.

# **Obiettivo**

Questo progetto ha come obiettivo quello di stimare la specie caratteristica di vari esemplari di bradipi in natura, basando la decisione sull’analisi di alcune feature degli animali. Le feature utilizzate per il dataset saranno poi descritte più specificamente al capitolo Analisi delle feature.

# **Scelta del dataset**

Il dataset scelto è denominato ***Species of sloths*** ed è disponibile sul sito kaggle.com al seguente link: <https://www.kaggle.com/datasets/bertiemackie/sloth-species>

Tale dataset è composto da 5000 istanze, ognuna delle quali è descritta da 8 features, che rappresentano le colonne del dataset in cui sono presenti i rilevamenti.

***Species of sloths*** è stato scelto perché il dataset, dopo essere stato analizzato, ha presentato le seguenti caratteristiche, che saranno analizzate nello specifico nel capitolo successivo:

* è costituito da un buon numero di istanze
* presenta una feature categorica (*specie*) utilizzabile come variabile target per la classificazione degli esemplari
* presenta due classi all’interno della variabile target molto ben bilanciate (53%, 47%)
* non presenta dati mancanti
* non presenta valori nulli

## Analisi e pre-process sui dati

Il dataset che è stato scelto presenta 5000 istanze descritte da 8 attributi, di seguito viene riportata una rappresentazione visiva dei primi record del dataframe, derivanti dalla funzione *head(dataset)*, per fornire contezza della struttura del medesimo.

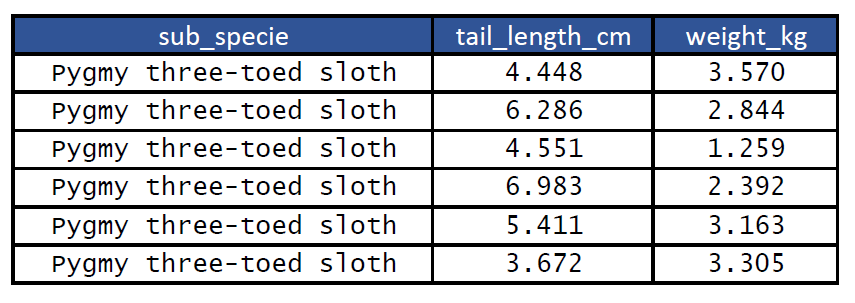
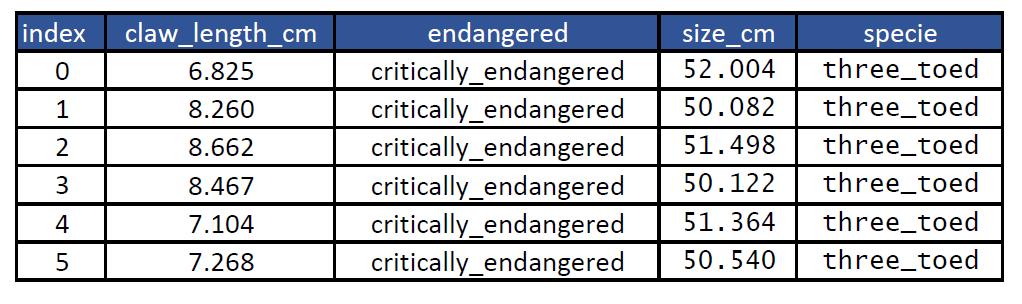


Tabella 1: Estratto del Dataset

Successivamente è esposta una breve descrizione delle feature, esplicativa del loro significato:

* **index:** identificatore numerico dell’istanza nel dataset
* **claw\_length\_cm:** variabile continua numerica della misurazione degli artigli espressa in centimetri
* **endangered:** variabile categorica discreta che descrive il grado di rischio della specie in via d’estinzione
* **size\_cm:** variabile continua rappresentante l’altezza dell’esemplare, comprendente la misurazione di testa e corpo, espressa in centimetri
* **specie:** variabile categorica binaria che denota la specie di appartenenza dell’esemplare (tridattilo o didattilo)
* **sub\_specie:** variabile categorica discreta che individua la sottospecie di appartenenza dell’animale
* **tail\_length\_cm:** variabile continua numerica della misurazione della coda espressa in centimetri
* **weight\_kg:** variabile continua numerica della misurazione del peso espresso in chilogrammi

## Preparazione del dataset

Prima di procedere all’ utilizzo del dataset per la modellazione, si sono svolti una serie di passaggi necessari alla preparazione dei dati, allo scopo di utilizzarli per i modelli di Machine Learning presi in esame.

I passaggi svolti sono stati:

1. **Controllo ed eventuale rimozione di valori nulli presenti nel dataset:**

tramite l’istruzione *round(colMeans(is.na(dataset))\*100, 2)* è stata verificata la presenza di valori NA all’interno del dataset, riscontrandone la totale assenza.

1. **Controllo e rimozione dei valori negativi presenti nel dataset e delle istanze ad essi collegate:**

tramite l’istruzione *colSums(dataset < 0)* è stata verificata la presenza di valori negativi presenti nel dataset. I valori negativi riscontrati sono stati 434, tutti presenti nella colonna della feature *tail\_length\_cm,* di conseguenza per mezzo dell’istruzione *dataset = dataset[dataset$tail\_length\_cm >= 0, ]* si è provveduto alla rimozione di ogni istanza ad essi collegata.

1. **Trasformazione in factor della variabile target:**

la variabile target identificata per la classificazione è *specie* ed i suoi valori sono stati convertiti in *factor* per mezzo dell’istruzione *dataset$specie = factor(dataset$specie)*

1. **Rimozione delle colonne inutili ai fini dello sviluppo dei modelli:**

si è proceduto alla rimozione di alcune features, le quali sono state considerate non necessarie ai fini del progetto, quindi, tramite l’istruzione *dataset = subset(dataset, select =c("claw\_length\_cm","size\_cm","tail\_length\_cm","weight\_kg","specie")),* sono state eliminate le colonne non necessarie, generando un nuovo dataset ridotto.

Successivamente a questi passaggi si è deciso di effettuare un’analisi preliminare della distribuzione delle osservazioni.

## Analisi delle features

È stata identificata nella variabile *specie* la target feature con la quale sarà eseguita la classificazione delle istanze del dataset.

Per la target feature sono stati individuati i valori unici che rappresentano la classificazione della specie e nella seguente pie chart viene rappresentata la suddivisione delle due classi.

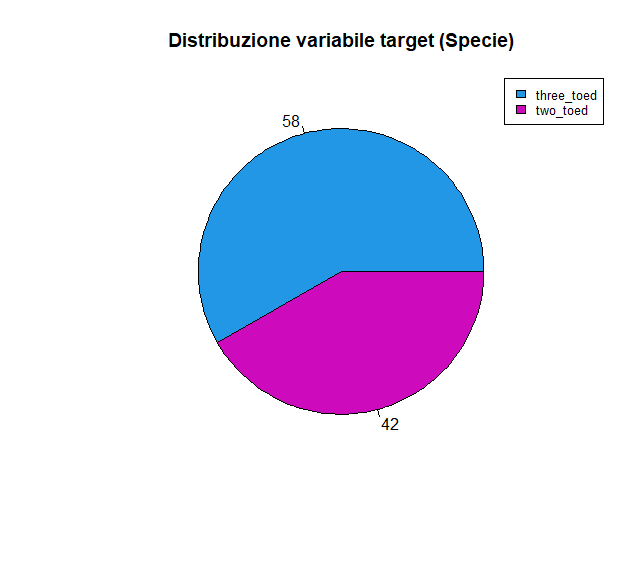


Figura 1: Pie Chart distribuzione di specie

In Figura 1 è stata presentata la Pie Chart che denota la percentuale di distribuzione dei valori della target feature del dataset. Possiamo notare come le due classi siano abbastanza equilibrate, la label three\_toed è presente per il 58%, invece la label two\_toed è presente per il 42% delle istanze.

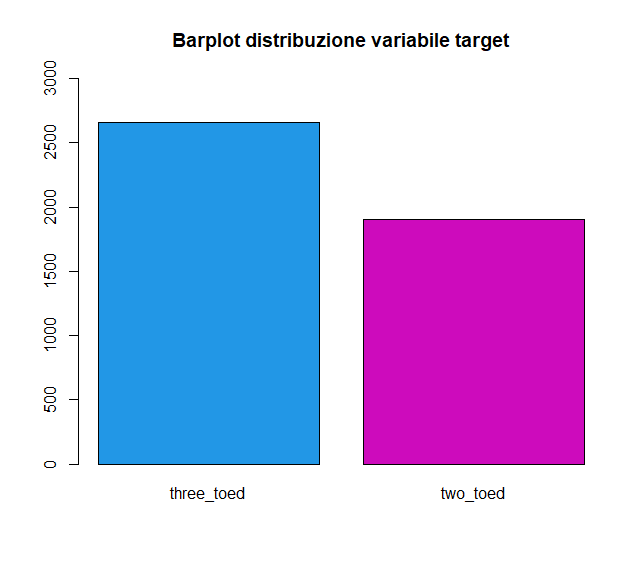


Figura 2: Barplot distribuzione della target feature (specie)

In Figura 2 è presentata la quantità di istanze a cui è stata assegnata una determinata label. La suddivisione, in valore assoluto, della target feature che è stata riscontrata è di 2662 istanze etichettate three\_toed e 1904 istanze per two\_toed.

Di seguito viene esposta un’analisi della distribuzione delle osservazioni per le singole feature costituite da valori continui, fornendo per ognuna di esse un istogramma della distribuzione delle osservazioni e un paired boxplot, dove sono riportate le mediane e i quartili delle distribuzioni ed eventuali *outliers,* che mostra la classificazione in three\_toed e two\_toed rispetto alla variabile considerata.

In questo modo abbiamo la possibilità di identificare come le misurazioni di queste feature, ad un primo esame, siano rappresentative della classificazione binaria, oppure si sovrappongano, impedendo di ricavarne particolari informazioni.

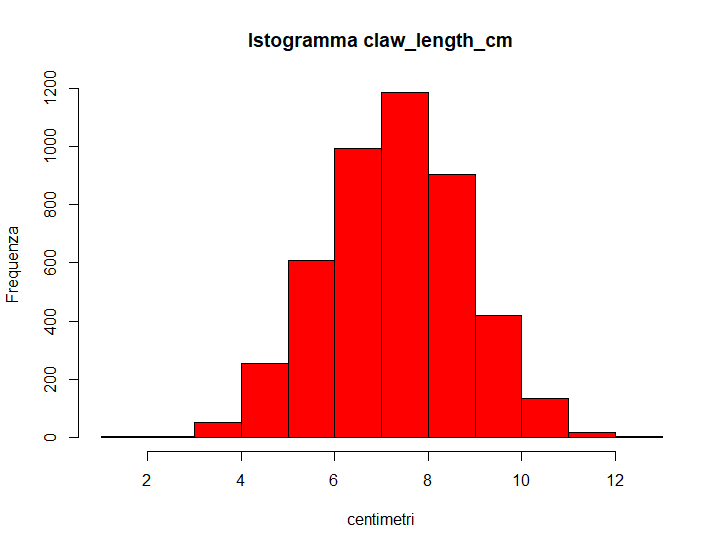


Figura 3.1: Istogramma di claw\_length\_cm

In Figura 3.1 è riportata la variabile *claw\_length\_cm* che rappresenta la lunghezza in centimetri degli artigli di ogni esemplare e mostra una distribuzione normale delle misurazioni osservate.

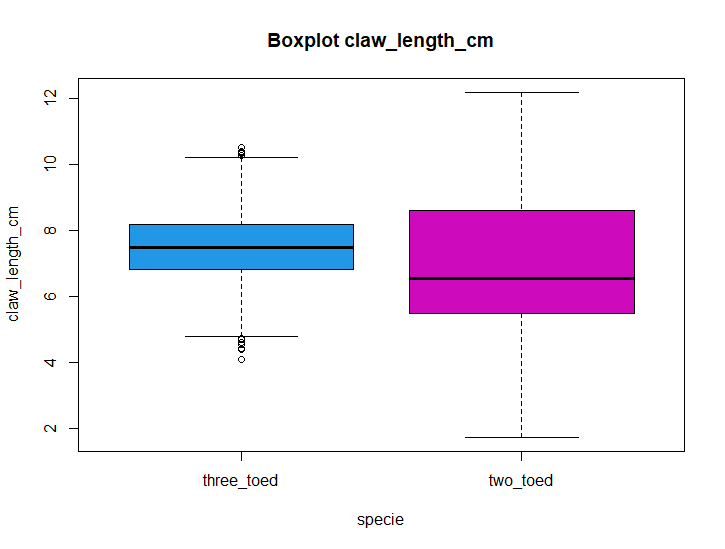


Figura 3.2: Paired Boxplot di claw\_length\_cm in relazione alla target feature

In Figura 3.2 è mostrato come la colonna *claw\_length\_cm* non sia significativa nella previsione della specie poiché le distribuzioni delle osservazioni si sovrappongono.

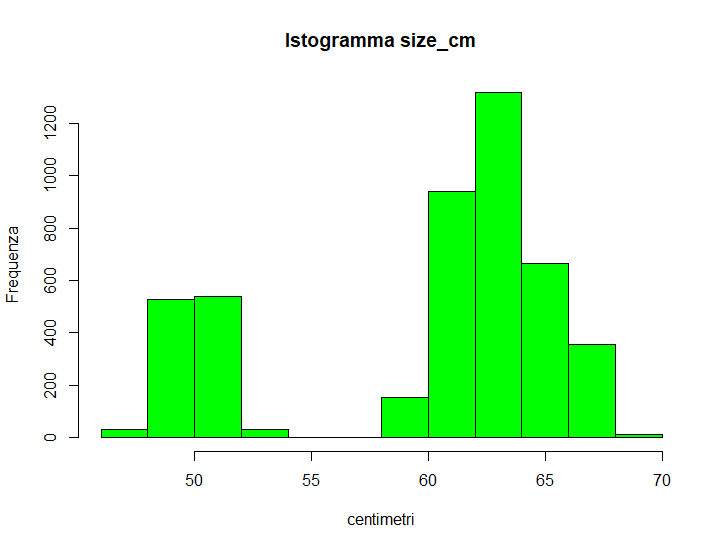


Figura 4.1: Istogramma di size\_cm

In Figura 4.1 è mostrata la variabile *size\_cm* che rappresenta la grandezza degli esemplari misurata in centimetri e fa riferimento alla misurazione di testa più la misurazione del corpo. La distribuzione della feature è particolarmente asimmetrica e sbilanciata verso il range di valori che vanno dai 60 ai 70 centimetri.

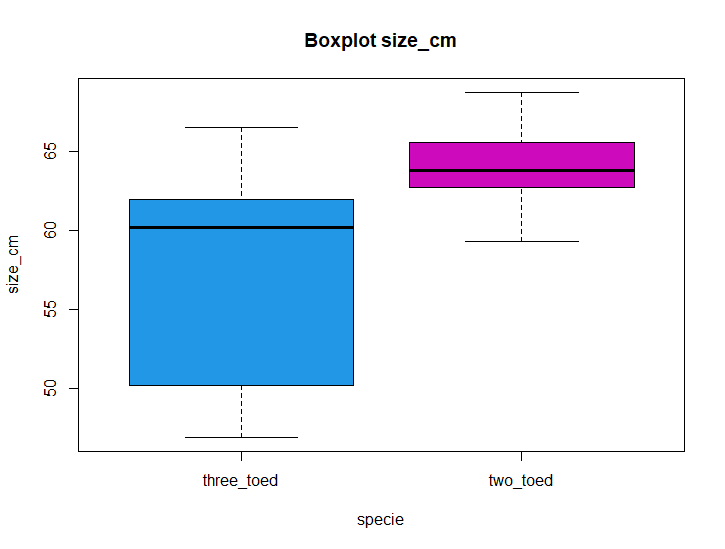


Figura 4.2: Paired Boxplot di size\_cm in relazione alla target feature

Nel boxplot in Figura 4.2 è possibile apprezzare come la colonna *size\_cm*, probabilmente, sia significativa nella previsione della specie in quanto le distribuzioni non si sovrappongono; la maggior parte degli esemplari più grandi di 60 cm, infatti, risultano appartenenti alla specie two\_toed.

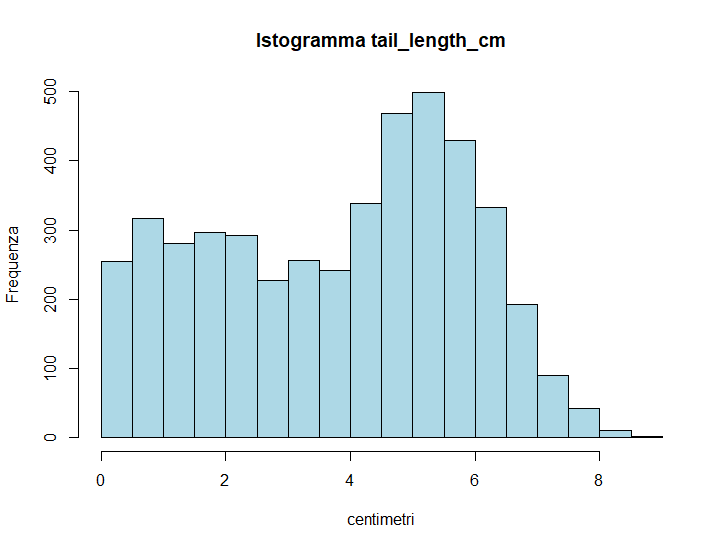


Figura 5.1: Istogramma di tail\_length\_cm

In Figura 5.1 è mostrata la variabile *tail\_length\_cm* che rappresenta la lunghezza misurata in centimetri della coda degli esemplari. La distribuzione, abbastanza asimmetrica, presenta una maggioranza di valori concentrati nel range tra i 4 e i 7 centimetri.

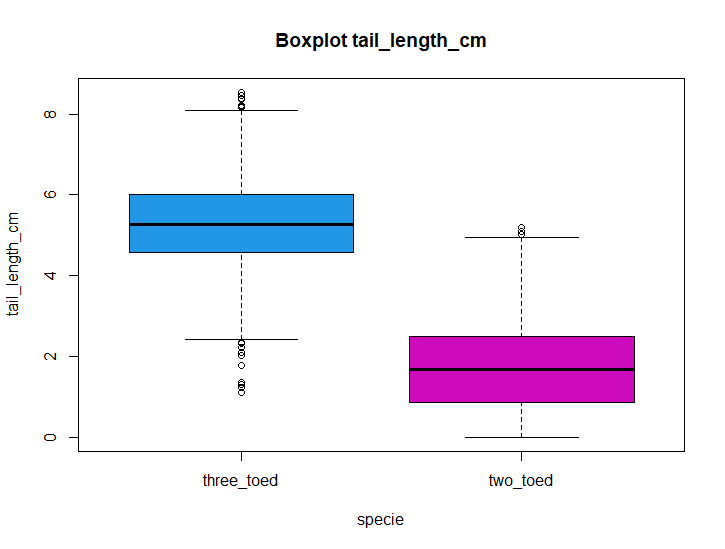


Figura 5.2: Paired Boxplot di tail\_length\_cm in relazione alla target feature

In Figura 5.2 è mostrato come la colonna *tail\_length\_cm*, probabilmente, è significativa nella previsione della specie in quanto le distribuzioni non si sovrappongono; la maggior parte degli esemplari con la coda più piccola di 3 cm, infatti, risultano appartenenti alla specie two\_toed.

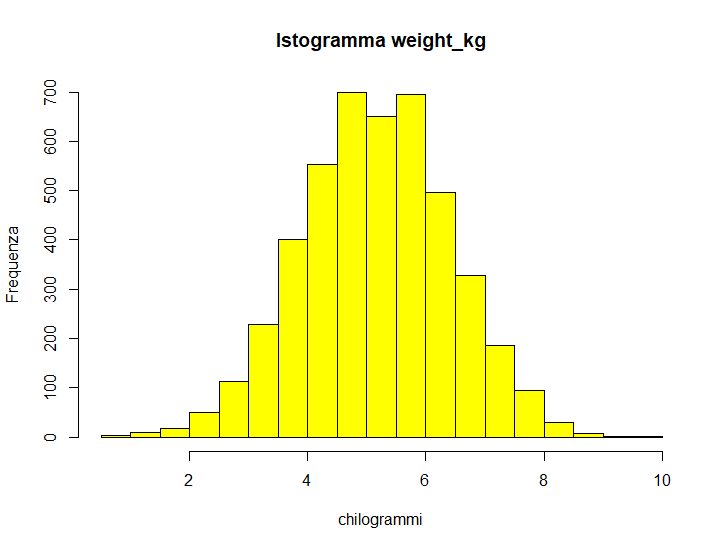


Figura 6.1: Istogramma di weight\_kg

Per quanto concerne la feature *weight\_kg*, la quale rappresenta la variabile che identifica il peso degli esemplari come mostrato in Figura 6.1, ha una distribuzione di tipo normale.

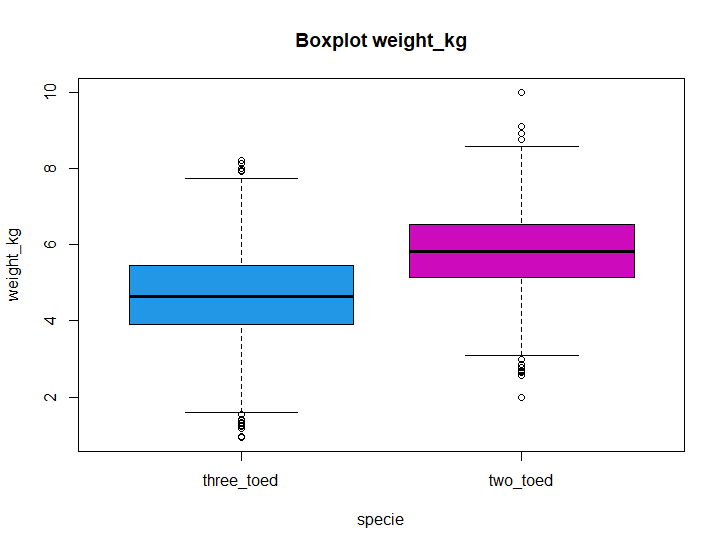


Figura 6.2: Paired Boxplot di weight\_kg in relazione alla target feature

In Figura 6.2 è osservabile come la colonna *weight\_kg*, probabilmente, è significativa nella previsione della specie in quanto le distribuzioni si sovrappongono solo in piccola parte; una parte consistente degli esemplari che pesano più di 5 kg, infatti, risultano appartenenti alla specie two\_toed.

Successivamente, si è deciso fornire una Data Visualization, tramite uno scatterplot comparativo di tutte le feature presenti nel dataset rimodellato, in modo da valutare quali degli attributi presi in considerazione rappresentasse al meglio la distribuzione dei dati per usarli nei modelli.

Per mezzo della funzione *featurePlot(x = x, y = y, main = "Scatter plot", plot = "pairs", auto.key = list(columns = 2))* sono state plottate tutte le feature del dataset, si veda Figura 7.

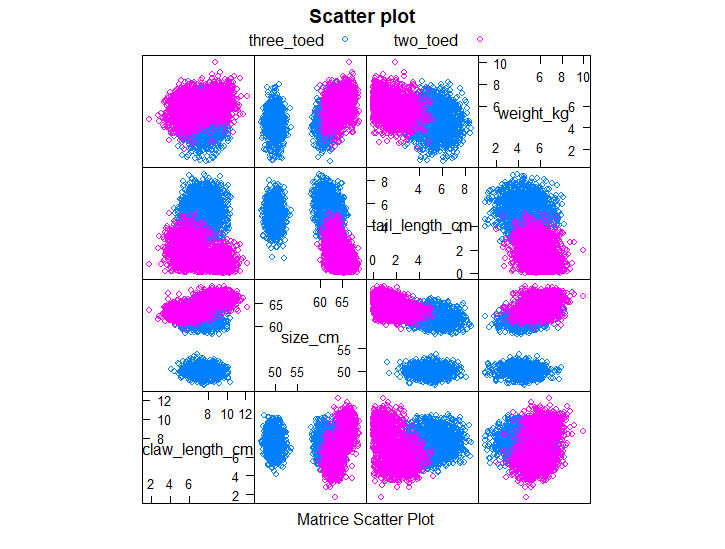


Figura 7: Scatterplot delle istanze rappresentate dalle 4 feature considerate

Come è possibile notare dalla Figura 7, osservando lo scatterplot, gli attributi *size\_cm* e *tail\_length\_cm* presentano una distribuzione abbastanza buona a livello di separabilità, il che ci permette di scegliere Support Vector Machine come modello di Machine Learning da implementare nel progetto.

È stata valutata inoltre la densità delle distribuzioni legata ad ogni feature con particolare attenzione riservata alla distribuzione delle densità suddivise per classificazione delle target label in *three\_toed* e *two\_toed*.

Per mezzo della funzione *featurePlot(x, y, plot = "density", main = "Feature plot", scales = list(x = list(relation = "free"), y = list(relation = "free")), auto.key = list(columns = 2))* sono stati generati 4 plot, uno per ogni feature, in relazione alla densità calcolata della classificazione delle istanze, si veda Figura 8.

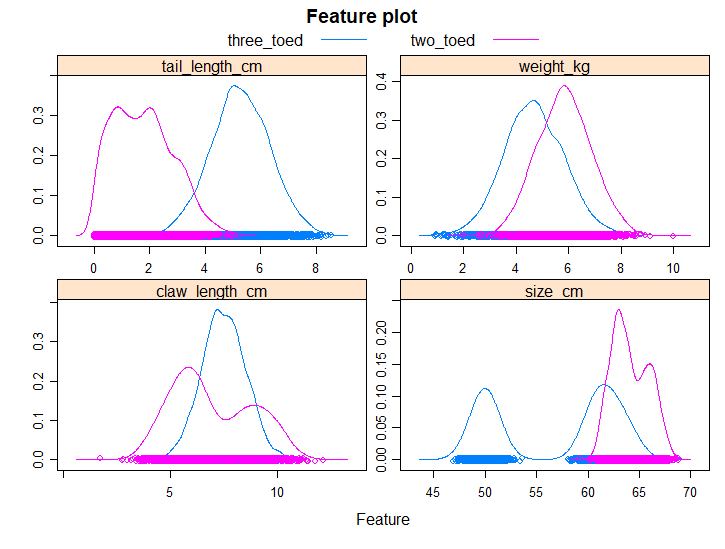


Figura 8: Feature plot della densità delle istanze

Come è possibile notare dalla distribuzione di *weight\_kg* e di *tail\_length\_cm*, esse denotano una sovrapposizione particolarmente accentuata in termini di misurazioni nelle osservazioni del dataset, questo ci porta a credere che tali feature siano poco significative per svolgere una classificazione delle istanze.

Invece, per quanto è possibile osservare riguardo a *tail\_length\_cm*, vediamo una distribuzione molto polarizzata delle osservazioni, dove la sovrapposizione delle istanze appartenenti alla classe *three\_toed* e *two\_toed* è veramente minima.

Per quanto riguarda *size\_cm*, come già riportato per la precedente feature, è possibile notare che la distribuzione delle classi *three\_toed* e *two\_toed* si sovrappone solo in parte. Infatti, vediamo come le istanze etichettate come *two\_toed* si concentrino nella parte più a destra del diagramma, per un range di valori tra 60 e 70, e seppur ci sia una certa parte di sovrapposizione con le istanze *three\_toed*, si è reputata come una buona feature, utile alla classificazione binaria della specie.

Quindi al termine dell’analisi, si è denotato come le feature *tail\_length\_cm* e *size\_cm* siano le covariate più rappresentative e che forniscono più informazioni riguardo la classificazione binaria delle istanze.

# **Modelli di Machine Learning**

In questo capitolo vengono presentati i modelli utilizzati ai fini dello svolgimento del progetto di Machine Learning, ovvero Decision Tree e Support Vector Machine (SVM).

## Decision Tree

È stato effettuato uno split del dataset in due dataset più piccoli nella misura del 70% e 30%, rispettivamente chiamati training set e test set, in modo da addestrare i modelli sul primo e, una volta “allenati”, effettuare delle previsioni sul secondo.  
  
Il primo è stato lanciato con l’istruzione *decisionTree = rpart(specie ~ ., data = trainset, method = "class”)* e ne viene effettuato il plot con l’istruzione *fancyRpartPlot(decisionTree)* che viene riportato di seguito, in Figura 9.1.

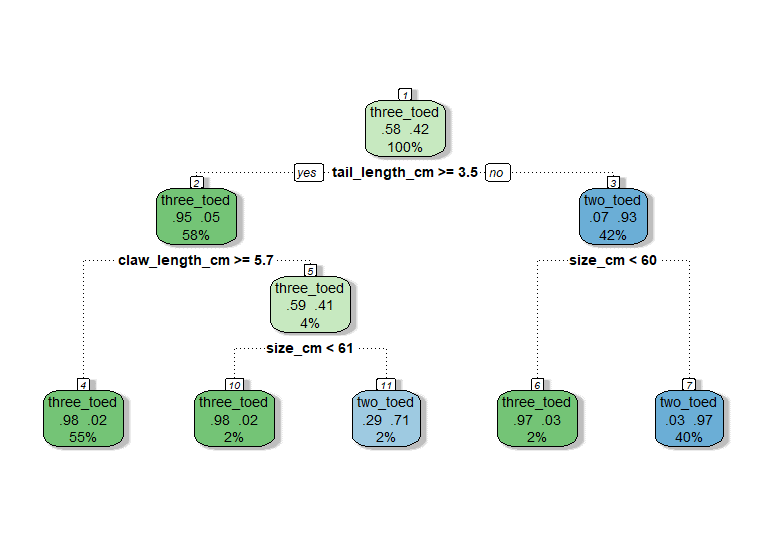


Figura 9.1: Decision Tree

Nel Decision Tree presentato in Figura 9.1, si è calcolata la misura di accuratezza che è risultata pari a 96.76%.

Controlliamo a quale parametro di complessità viene tagliato l’albero con l’istruzione *plotcp(decisionTree)*, che ci permette di effettuare il plot dell’intera tabella dei parametri e relativi livelli di split dell’albero, si veda Figura 9.2.

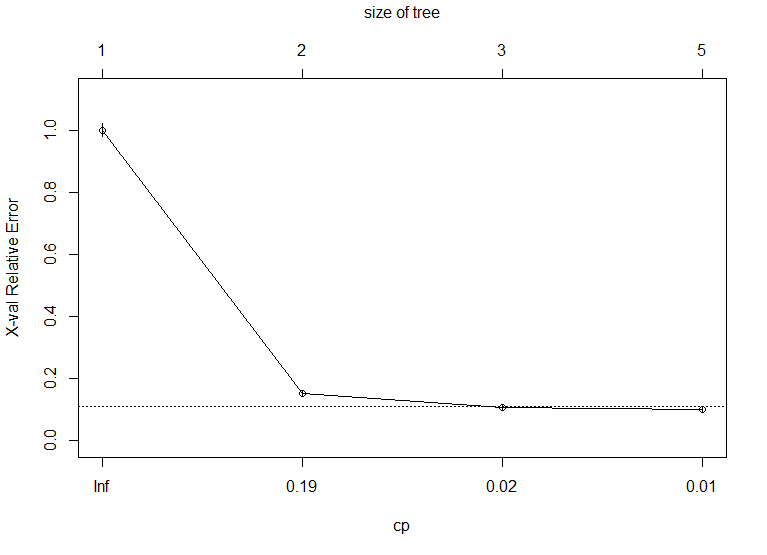


Figura 9.2: Plot dei parametri di complessità

Notiamo dal plotting dei parametri di complessità che si potrebbe effettuare il cut dell’albero ad una sua dimensione minore (3), in modo tale da evitare l’overfitting. Il cut dell’albero *pruned* è stato effettuato nella misura *cp = 0.02*.

Di seguito è presentato il plot del nuovo albero di decisione, che ovviamente presenterà una misura di accuratezza leggermente minore rispetto all’albero originale, seppur mantenendo altissimo tale livello di accuratezza.

Il nuovo pruned Decision Tree, che presenta due split, riferiti alla variabile *tail\_length\_cm* per valori maggiori o uguali a 3.5 centimetri e alla variabile size\_cm per valori inferiori a 60 centimetri, è stato plottato in Figura 9.3.

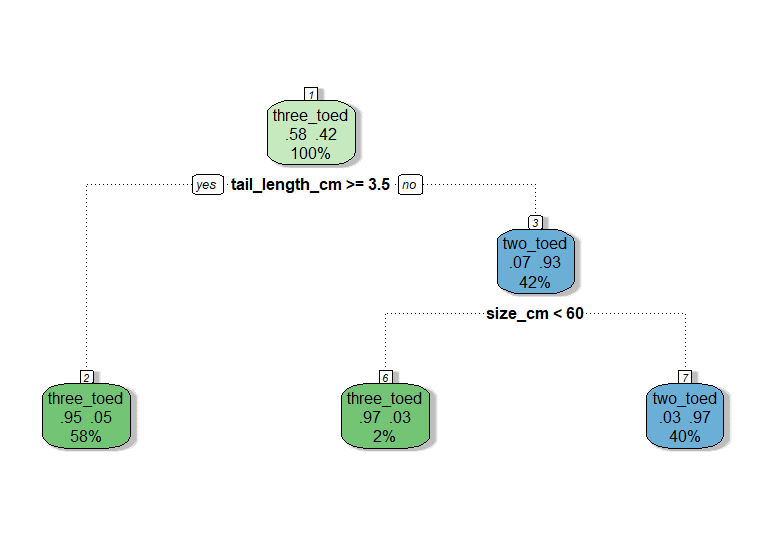


Figura 9.3: Nuovo Pruned Decision Tree

Per quanto riguarda questa nuova formulazione del Decision Tree è stato calcolato che la sua misura di accuratezza è pari al 96.47%.

Quindi possiamo notare come riducendo la complessità del modello, riusciamo a mantenere un altissimo grado di accuratezza, che in questo caso è stato riscontrato in una diminuzione percentuale di solo lo 0.29%.

## Support Vector Machine (SVM)

Il modello SVM è stato scelto per la sua elevata efficienza nel separare dati complessi inseriti in spazi ad alta dimensionalità.

Prima di addestrare il modello sul training set (*trainset*), è stato effettuato un tuning degli iperparametri quali costo, gamma e funzione kernel. Il tuning di questo modello è stato effettuato per un range di valori di seguito riportati:

* Il costo viene scelto tra una serie di valori: 0.1, 1, 10, 100, 1000.
* Gamma viene scelto tra una seconda serie di valori: 0.5, 1, 2, 3, 4.
* La funzione kernel scelta sarà lineare o radiale (sono state utilizzate solo queste due funzioni poiché, aggiungendone una terza, veniva generato un errore di *reached maximum number of iterations*)

Il tuning del modello viene effettuato dalla funzione *tune* del package *e1071* di R, che sceglie tra i valori descritti nel paragrafo precedente. Di seguito il blocco di istruzioni per effettuare le operazioni precedentemente descritte.

*cost\_range = c(0.1, 1, 10, 100, 1000)  
gamma\_range = c(0.5, 1, 2, 3, 4)  
kernel\_range = c("linear", "radial")  
tuned = tune(svm, specie ~ ., data = trainset, ranges = list(cost = cost\_range, gamma = gamma\_range, kernel = kernel\_range))*

In seguito al tuning, vengono scelti e impostati gli iperparametri migliori che, al momento della scrittura di questo documento, sono:

**Costo** = 10  
**Gamma** = 0.5  
**Funzione kernel** = Radiale

Una volta ottenuti questi valori, quali risultato della ricerca dei migliori iperparametri, essi sono stati passati al modello SVM ed il modello è stato lanciato.

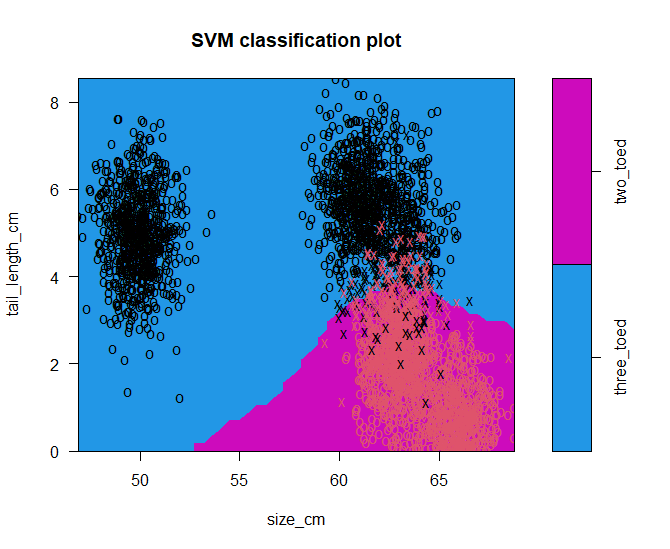


Figura 9.4: SVM plot

In Figura 9.4, viene mostrata la rappresentazione grafica del modello SVM, addestrato con una funzione kernel radiale, applicato alle features *tail\_length\_cm* e *size\_cm*.

# **Stima delle misure di performance**

In questo capitolo è presentato lo studio delle misure di performance relative ai modelli utilizzati nel progetto. Al termine dell’analisi sarà riportato, nel Capitolo Conclusioni, una riflessione finale su quale dei due modelli si è dimostrato essere più performante.

## Performance Decision Tree

In questo capitolo viene riportata la tabella che presenta la matrice di confusione complessiva calcolata per il Decision Tree. Successivamente vengono presentate le misure di performance da essa ricavate.

|  |  |  |
| --- | --- | --- |
|  | REFERENCE |  |
| PREDICTION | THREE\_TOED | TWO\_TOED |
| THREE\_TOED | 774 | 35 |
| TWO\_TOED | 13 | 538 |

Dopo aver allenato entrambi i modelli, si sono analizzate le metriche di performance ”Accuracy”, ”Precision”, ”Recall”, ”F1” ricavate dalla matrice di confusione e le curve ROC con la rispettiva AUC.

L’**Accuracy**, misura di accuratezza, come già enunciato nel capitolo relativo alle motivazioni che hanno portato alla scelta di questo modello, è del 96.47%. Essa è rappresentata dal rapporto tra il numero di previsioni corrette sia positive che negative e il totale delle previsioni, come segue:

*Accuracy = (T P + T N)/(T P + T N + F P + F N)*

La **Precision** è il rapporto tra il numero delle previsioni corrette (TP) di un evento sul totale delle volte che il modello lo prevede (TP+FP), per mezzo della formulazione:

*Precision = T P/(T P + F P)*

La **Recall** misura la sensibilità del modello. È il rapporto tra le previsioni corrette (TP) per una classe sul totale dei casi in cui si verifica effettivamente (TP + FN), espressa con la formula:

*Recall = T P/(T P + F N)*

L’ **F1-Measure** è una media armonica, più nello specificatamente è la media armonica della Precision e della Recall, che si esprime nella formula:

*F1-Measure = (2 ∗ P recision ∗ Recall)/(P recision + Recall)*

Relativamente alle misure di performance per ogni classe, di seguito vengono riportati Precision, Recall e F-Measure (F1) per la classe *three\_toed*.

**Precision** = 0.9567

**Recall** = 0.9835

**F1-Measure** = 0.9699

Di seguito vengono riportati Precision, Recall e F-Measure (F1) per la classe *two\_toed*.

**Precision** = 0.9764

**Recall** = 0.9389

**F1-Measure** = 0.9573

Calcoliamo ora i valori medi complessivi delle performance per il modello Support Vector Machine, sfruttando la Macro-average, dove ogni classe riveste un’importanza uguale alle altre nella valutazione del modello.

**Mean Accuracy** = 0.9647 = 96.47%

**Mean Precision** = 0.9665 = 96.65%

**Mean Recall** = 0.9612 = 96.12%

**Mean F1-Measure** = 0.9636 = 96.36%

Per valutare l’efficacia del modello, inoltre, sono state utilizzate le curve ROC (Receiver Operating Characteristic). Le curve ROC sono dei plot che illustrano le performance di un classificatore binario. Lungo i due assi vengono mostrati la sensibilità e la specificità, rispettivamente rappresentanti la proporzione di veri positivi e la proporzione di veri negativi. In altre parole queste curve permettono di studiare il rapporto fra i veri positivi e i veri negativi.

Comunemente questo plot è utilizzato per calcolare l’AUC (Area Under Curve), necessario a misurare la perfomance di un modello di classificazione. Più la curva ottenuta tende ad avvicinarsi alla curva perfetta (curva verde nell’esempio di Figura 10.1), dove il 100% delle istanze viene classificato correttamente, più si otterrà una buona performance del modello.

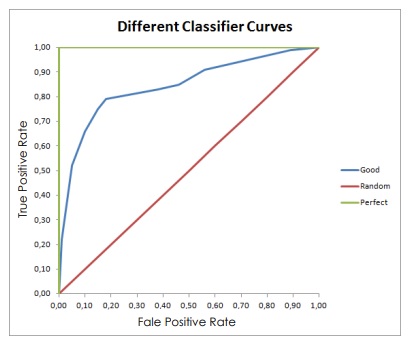


Figura 10.1: esempio di curva ROC

Ora riportiamo il plot della curva ROC per il modello Decision Tree, riportata in Figura 10.2, la quale, come si può visivamente notare, si avvicina molto all’esempio di curva perfetta, presentata nella Figura 10.1. Essa presenta una AUC (Area Under Curve) pari a 0.961. Ricordiamo che, essendo l’AUC una misura che varia in un range continuo di valori tra 0 e 1, tale risultato denota una performance molto buona del Decision Tree.

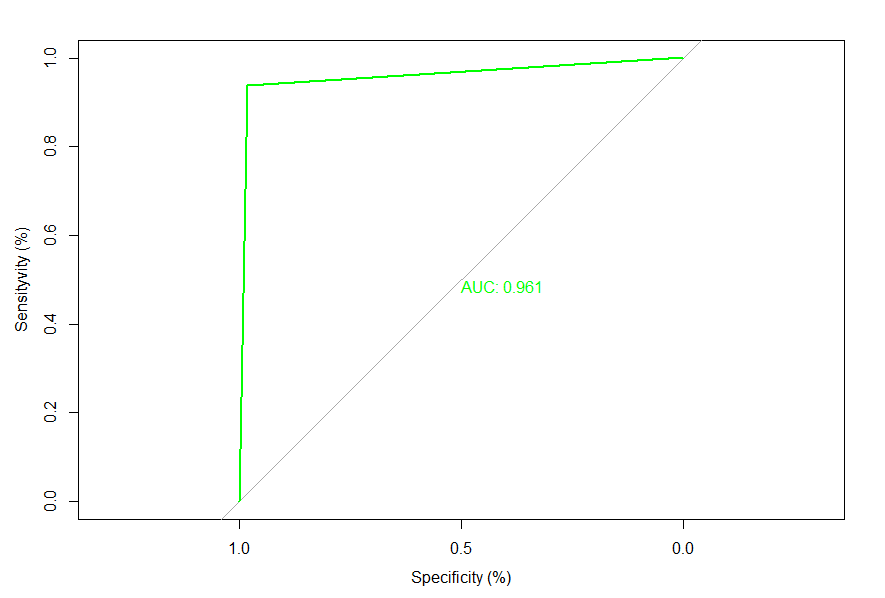
****

Figura 10.2: Curva ROC di Decision Tree con valore di AUC

## Performance Support Vector Machine (SVM)

In questo capitolo, viene riportata in primis, la tabella che presenta la matrice di confusione complessiva, calcolata per SVM. Successivamente vengono presentate le misure di performance da essa ricavate.

|  |  |  |
| --- | --- | --- |
|  | REFERENCE |  |
| PREDICTION | THREE\_TOED | TWO\_TOED |
| THREE\_TOED | 768 | 14 |
| TWO\_TOED | 19 | 559 |

La misura di **Accuracy** stimata è risultata pari al 97.57%.

Relativamente alle misure di performance per ogni classe, di seguito vengono riportati Precision, Recall e F-Measure (F1) per la classe *three\_toed*.

**Precision** = 0.9821

**Recall** = 0.9759

**F1-Measure** = 0.9790

Di seguito vengono riportati Precision, Recall e F-Measure (F1) per la classe *two\_toed*.

**Precision** = 0.9671

**Recall** = 0.9756

**F1-Measure** = 0.9713

Calcoliamo ora i valori medi complessivi delle performance per il modello Support Vector Machine, sfruttando la Macro-average, dove ogni classe riveste un’importanza uguale alle altre nella valutazione del modello.

**Mean Accuracy** = 0.9757 = 97.57%

**Mean Precision** = 0.9746 = 97.46%

**Mean Recall** = 0.9757 = 97.57%

**Mean F1-Measure** = 0.9751 = 97.51%

Ora riportiamo il plot della curva ROC per il modello Support Vector Machine, riportata in Figura 10.3, la quale, come si può visivamente notare, si avvicina ancora di più all’esempio di curva perfetta, presentata nella Figura 10.1, rispetto alla curva ROC del Decision Tree. Essa presenta una AUC (Area Under Curve) pari a 0.991, che supera di 0.03 punti la curva del Decision Tree. Ricordiamo che, essendo l’AUC una misura che varia in un range continuo di valori tra 0 e 1, tale risultato denota una performance estremamente elevata del modello SVM.

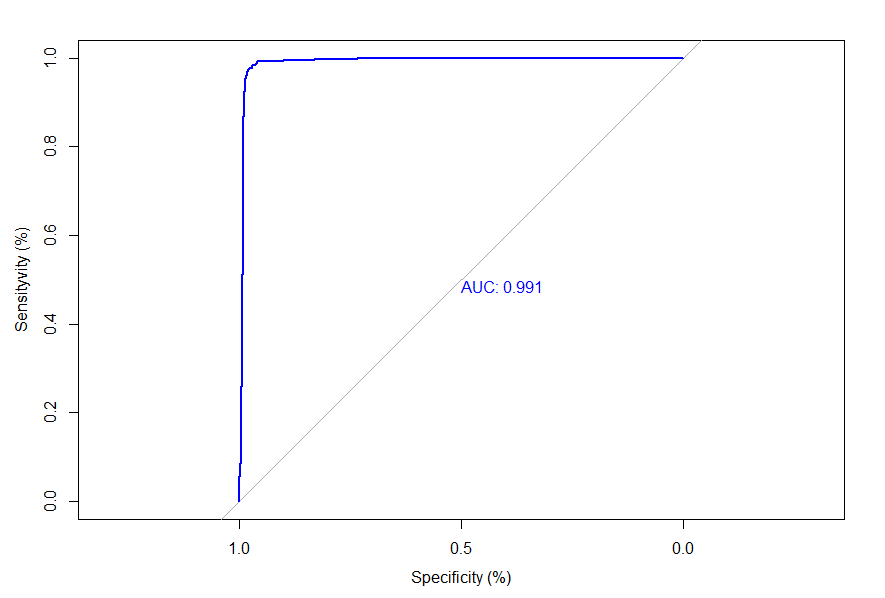


Figura 10.3: Curva ROC di SVM con valore di AUC

# **Confronto tra modelli**

Come possiamo notare dalle varie metriche analizzate nei capitoli precedenti, i due modelli performano in maniera molto simile. Denotiamo, però, come il Support Vector Machine, con funzione kernel radiale, sia in grado di performare in maniera più efficiente, ed in particolar modo:

Il modello SVM risulta più efficiente dell’1.1% per la metrica **Accuracy.**

Il modello SVM risulta più efficiente dello 0.81% per la metrica **Precision.**

Il modello SVM risulta più efficiente dell’1.45% per la metrica **Recall.**

Il modello SVM risulta più efficiente dell’1.15% per la metrica **F1-Measure.**

L’AUC di **Decision Tree** è pari a 96.1%.

L’AUC di **Support Vector Machine** è pari a 99.1%.

L’AUC, essendo la misura della capacità di un classificatore di distinguere tra le classi ed essendo utilizzata come sintesi della curva ROC, consente di dire, paragonando le due prestazioni dei modelli, che il modello SVM ha un AUC maggiore del 3% e quindi migliore nella classificazione delle istanze rispetto al Decision Tree.

Tale differenza è anche rappresentata graficamente in Figura 11.

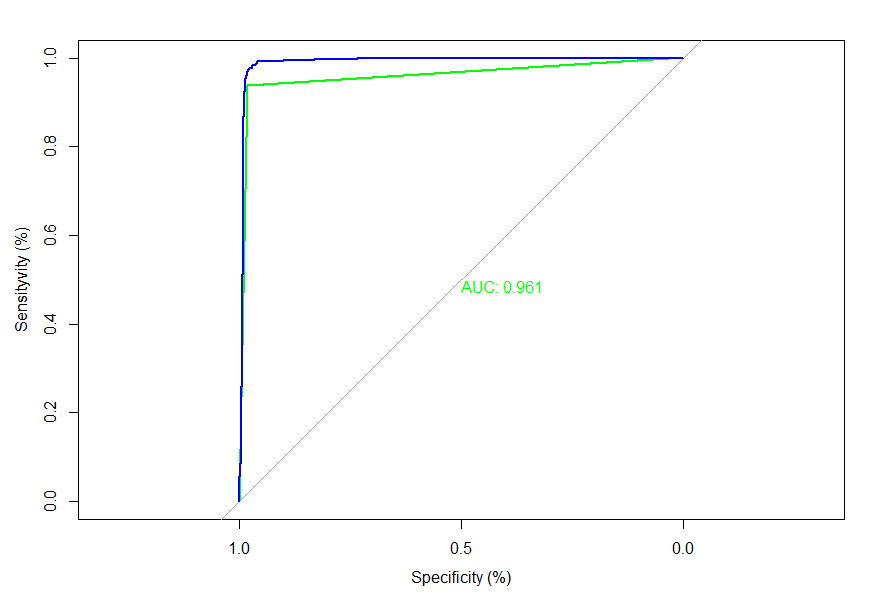


Figura 11: Curve ROC a confronto

# **Conclusioni**

In questo progetto abbiamo utilizzato un dataset composto da molte istanze (5000 osservazioni) e con un buon numero di feature, ovvero 8, che ci ha permesso, selezionando una target feature ben bilanciata, di effettuare una classificazione binaria molto buona per le due classi di appartenenza a cui abbiamo fatto riferimento. Possiamo affermare che, dall’analisi preliminare svolta sulle feature del dataset, è stato subito evidente come alcune delle covariate fossero molto significative per la classificazione della specie.

Dopo aver addestrato i modelli scelti, ovvero Decision Tree e Support Vector Machine, abbiamo deciso di svolgere il tuning degli iperparametri, per consentire ai nostri modelli di essere più performanti, senza andare ad intaccare significativamente l’accuratezza del modello stesso.

Siamo giunti così, nel primo caso, a ridurre la complessità del modello senza andare a perdere accuratezza nella classificazione, per quanto riguarda il Decision Tree. In quanto ricordiamo che passando a un pruned tree, con due split su tre livelli (con cp=0.02), abbiamo perso solamente lo 0.29% di accuratezza, con una misura finale di Accuracy del 96.47%.

Invece, nel secondo caso, abbiamo mostrato come il nostro modello di SVM, dopo il tuning, mantenga una accuratezza molto elevata, pari al 97.57%.

Aggiungiamo che a livello computazionale, però, Decision Tree è decisamente più efficiente in quanto il tuning di SVM risulta essere molto più oneroso.

Infine, possiamo dire, come mostrato nel capitolo precedente “Confronto tra modelli”, che abbiamo evidenziato come le prestazioni del modello SVM siano state sempre superiori al modello Decision Tree, seppur di pochi punti percentuali e come la feature per la classificazione siano state estremamente significative, consentendoci di mantenere alti valori di performance nelle predizioni delle classi.