Fondamenti di Scienza dei Dati
Classificazione del dataset Iris
mediante Support Vector Machine

Gruppo 0: Gabriele Lo Cascio, Alessia Bonì, Samuele Ales

27/05/2024

# Introduzione

Questo progetto riguarda l'analisi e la classificazione del famoso dataset Iris. Le informazioni raccolte includono la lunghezza e la larghezza sia del sepalo che del petalo di diversi fiori. L'obiettivo è utilizzare il modello Support Vector Machine (SVM), con diverse configurazioni, per classificare i dati. Di fondamentale importanza è l'analisi preliminare di statistica descrittiva, che consente di acquisire maggior consapevolezza dei dati al fine di valutare successivamente la bontà del modello.

# Cenni teorici

*Support Vector Machine* (SVM) è un modello di supervised machine learning potente e versatile in grado di effettuare classificazione binaria. Anche se può essere esteso per affrontare problemi di classificazione multi-classe (tramite tecniche come *One-vs-All* oppure *One-vs-One*), regressione e novelty detection ossia l'identificazione di dati o pattern nuovi in un dataset, ai quali il modello non è stato esposto durante la fase di addestramento. É importante notare che con questo tipo di modello si riescono ad ottenere buone performance solo per dataset nell'ordine di centinaia o al massimo migliaia di istanze. Il fatto che sia un modello di tipo supervisionato vuol dire che richiede dati di addestramento etichettati. Ciò significa che per effettuare operazioni di classificazione si deve addestrare il modello su dati di cui si conosce già la classe di appartenenza. Quindi sapere a priori l'etichetta è fondamentale e giustifica l'idea alla base del modello ossia il concetto di *classi linearmente separabili*. Supponendo di avere dei punti su un piano cartesiano che, sebbene siano sparpagliati, formano due agglomerati distinti, ecco che il modello SVM ricerca una linea che riesca a separare le due zone senza che vadano in *conflitto*. La classificazione consiste nel

capire in quale delle due zone, individuate durante la fase di addestramento, si trovano i nuovi dati. La parola *conflitto*, usata precedentemente, non è un caso, infatti non sempre è possibile riuscire a separare due classi con una linea. In questi casi un possibile approccio è aumentare la dimensionalità del dataset, per far in modo che si riesca a trovare un piano (o in casi più complessi un iperpiano) che separi le due classi. Da un punto di vista geometrico, questa operazione consiste nel proiettare in uno spazio vettoriale, ad una più alta dimensionalità, i punti del dataset. Difatto, la nuova feature aggiunta al dataset viene calcolata a partire da quelle già esistenti applicando una particolare funzione detta *kernel*. In machine learning una funzione di *kernel* è in grado di mappare vettori da un sotto spazio ad un altro, senza effettivamente applicare la formula di mapping e senza addirittura conoscerla, basta sapere che esiste. Gli unici ingredienti sono semplicemente i vettori del sotto spazio iniziale. Solitamente questa proprietà è detta *kernel trick* ed è garantita dal Teorema di Mercer.

# Statistica descrittiva

Il dataset è composto da 150 istanze per un totale di 5 features:

- *sepal_length*: la lunghezza del sepalo (ossia una particolare foglia);
- *sepal_width*: la larghezza del sepalo;
- *petal_length*: la lunghezza del petalo;
- *petal_width*: la larghezza del petalo;
- *species*: rappresenta il target ossia le diverse specie a cui appartengono i fiori. Nello specifico si hanno: setosa, versicolor e virginica.

Inoltre non sono presenti valori mancanti e la numerosità di ogni classe è la stessa.

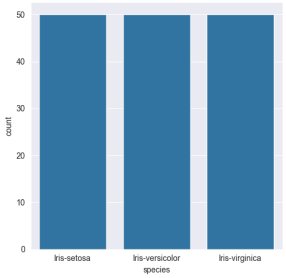



*Figura 1 - Grafico a barre relativo alla numerosità di ogni specie.*

Per avere contezza delle relazioni che intercorrono tra le features si è realizzato un *pairplot* mediante la libreria *seaborn*. Si impostano sulla diagonale principale della griglia dei boxplot al fine di individuare immediatamente eventuali asimmetrie e la presenza di outliers. Nei restanti grafici della griglia vengono effettuati degli scatterplot, in questo modo si notano eventuali dipendenze funzionali.

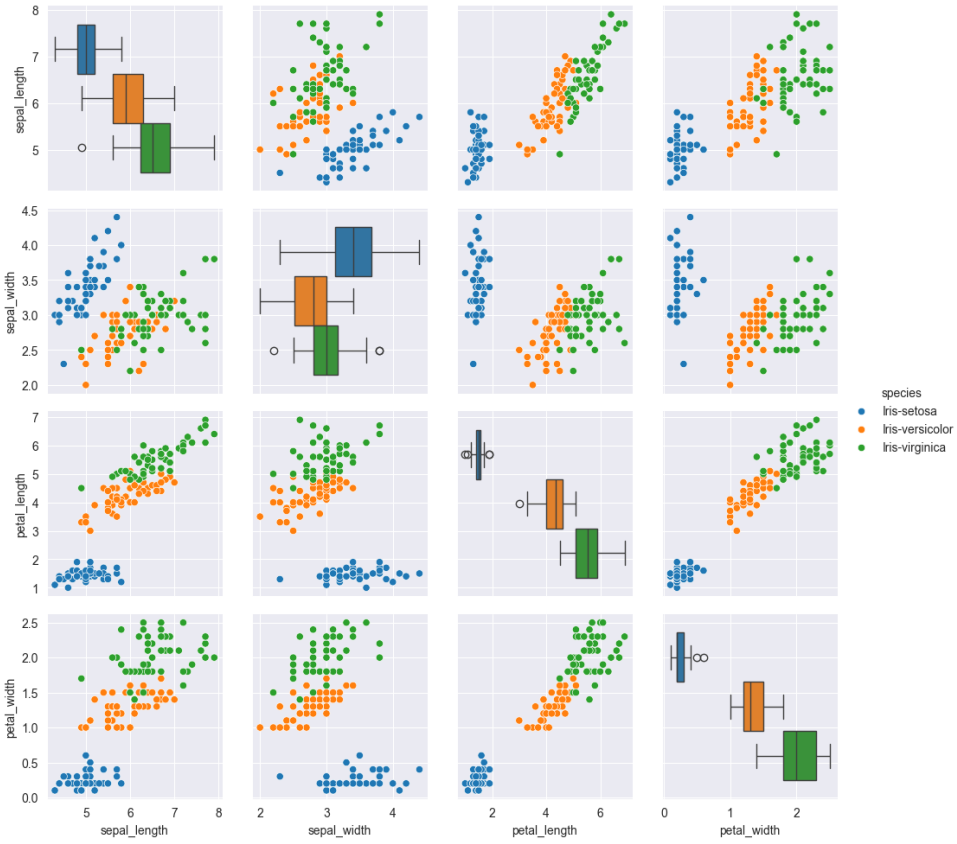



*Figura 2 - Griglia riassuntiva con scatterplot e boxplot sulla diagonale principale.*

Dagli scatterplot si nota che vi è una correlazione positiva tra le seguenti features:

- petal_lenght $\sim$ sepal_length;
- petal_width $\sim$ sepal_length;
- petal_width $\sim$ petal_length.

Mentre sui boxplot si nota la presenza di alcuni outliers, di seguito si riportano i soli boxplot per un'analisi di maggior dettaglio.

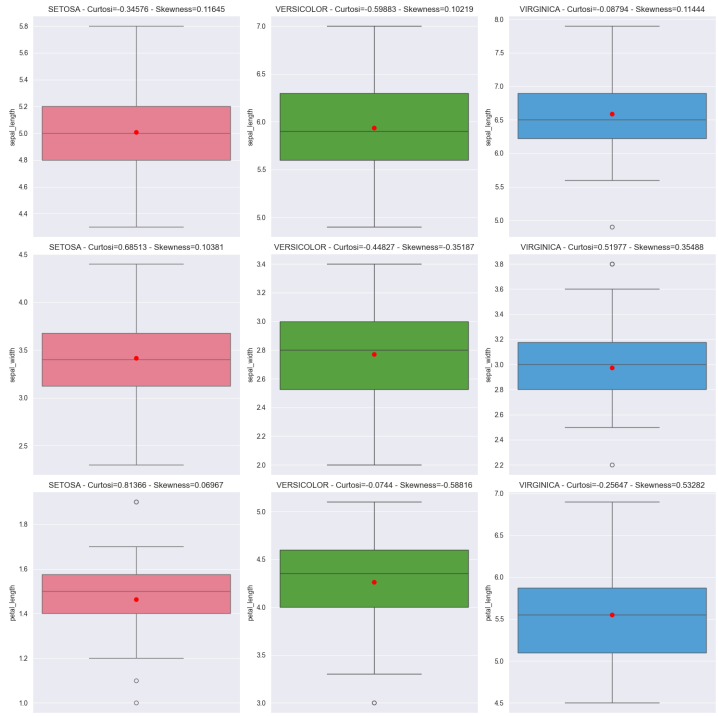



*Figura 3 - Boxplot delle varie features (righe) separate per specie (colonne). Si riporta anche l'indice di Curtosi e l'indice di Asimmetria di Pearson per ogni grafico. Il puntino rosso rappresenta la media aritmetica dei valori.*

Nel complesso le distribuzioni mostrano un andamento pressocché normale, si ricorda infatti che l'indice di Curtosi di Pearson è nullo nel caso in cui la distribuzione sia normale mentre è positivo nel caso di picchi appuntiti (o code pesanti) oppure negativo nel caso di picchi piatti (o code leggere). Inoltre non sono presenti evidenti asimmetrie. Infine si nota la presenza di probabili outliers in alcuni boxplots.

# Correlation Matrix

Una misura quantitativa di quanto ciascun attributo sia correlato ad un'altro si ottiene calcolando la matrice di correlazione. Di fatto si determina l'indice di correlazione lineare $\left(\frac{\sigma_{xy}}{\sigma_x \sigma_y}\right)$ per ogni coppia di attributi e si crea una matrice per visualizzare comodamente ogni valore.

Tenendo conto che una buona correlazione si ha se l'entrata della matrice è in modulo pressocché pari ad 1.

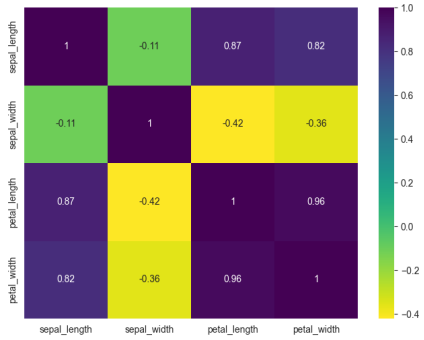



*Figura 4 - Heatmap di correlazione, si riconfermano le correlazioni tra le features notate nel pariplot in Figura 1.*

Si realizza infine un grafico in cui vengono rappresentate le tre rette di regressione.

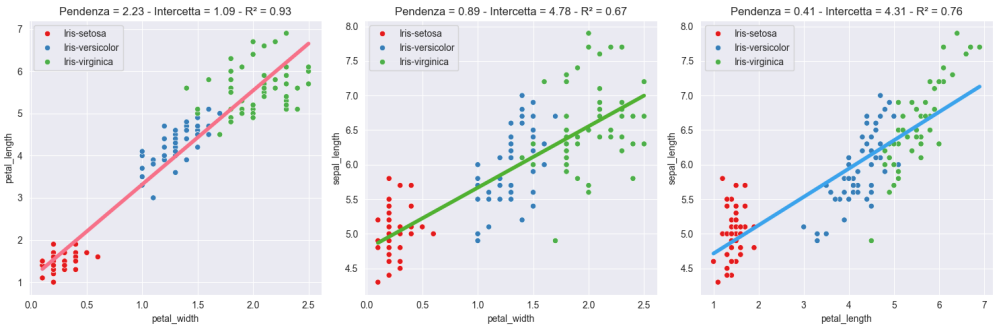



*Figura 5 - Grafici delle tre coppie di features. Si riporta la pendenza, l'intercetta e $R^2$ score.*

Si osserva dunque una correlazione positiva per ogni coppia di attributi. Si ricorda che la pendenza rappresenta la variazione media subita da Y quando X aumenta di un'unità. Si riporta inoltre lo score $R^2$ che fornisce una misura di quanto sia forte la correlazione tra gli attributi (che è perfetta se pari ad 1).

I risultati ottenuti durante questa analisi mostrano come non siano presenti valori anomali significativi, quindi non occore effettuare alcuna scrematura del dataset. Le correlazioni tra alcuni campi potrebbero risultare utili successivamente per un eventuale *PCA*, quindi rimuovendo ridondanza. Inoltre le features che presentano una correlazione con altre sono quelle che hanno maggior rilevanza all'interno del dataset, costituiscono un metodo di misura per classificare i fori. In *Figura 5*, infatti, si notano delle regioni differenti per ogni specie ed utilizzando una combinazione delle tre rette di regressione si potrebbere risalire alla classe di appartenenza del fiore.

# Costruzione del modello

## Pre-processing dei dati

Al fine di valutare il modello a seguito della fase di addestramento, si procede dapprima alla suddivisione del dataset in due parti: una dedicata al training ed una al testing. Questo perchè valutare la bontà del modello sugli stessi dati di addestramento produrrebbe chiaramente risultati soddisfacenti. Si è scelto di utilizzare l'80% dei dati per il training e la restante parte per il testing. Infine è stata operata una standardizzazione dei dati per cui avranno media nulla e varianza unitaria.

## Principal Component Analysis

Come anticipato precedentemente nella parte di regressione lineare, è utile effettuare una riduzione della dimensionalità del dataset rimuovendo ridondanza. Inoltre, visto che si sta utilizzando il modello SVM, è utile ridurre la dimensionalità per poter rappresentare graficamente le regioni di decisione del modello. Di seguito si riporta il grafico della varianza cumulata per comprendere quanto ogni feature è in grado di rappresentare il dataset.

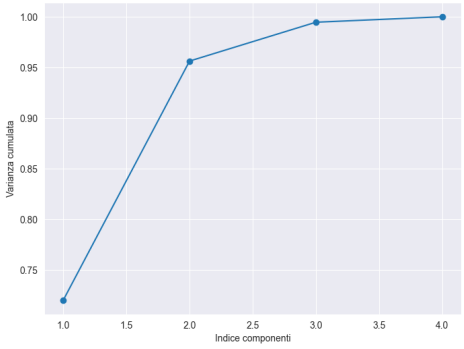



*Figura 6 - Grafico della varianza spiegata da ogni componente. Si nota che la maggior parte, poco più del* 95%*, è spiegata dalle prime due componenti.*

Per avere maggior contezza del legame tra le componenti principali trovate e le features originali, si procede a realizzare un *biplot* ossia un grafico in cui vengono rappresentati contemporaneamente gli *scores* ed i *loadings*. Dove gli scores rappresentano i punti del dataset originale ma proiettati nello sottospazio delle componenti principali. Mentre i loadings sono i coefficienti, relativi ai vettori della base del sottospazio delle PCs, che determinano la combinazione lineare per cui avviene il mapping da un sottospazio ad un altro. Indicando, inoltre, quanto ogni feature originale contribuisce a ciascuna componente principale. Il *biplot* permette quindi di visualizzare sia come le osservazioni si distribuiscono secondo le componenti principali, sia come le variabili originali influenzano queste componenti, facilitando l'interpretazione delle relazioni tra le variabili e le componenti principali.

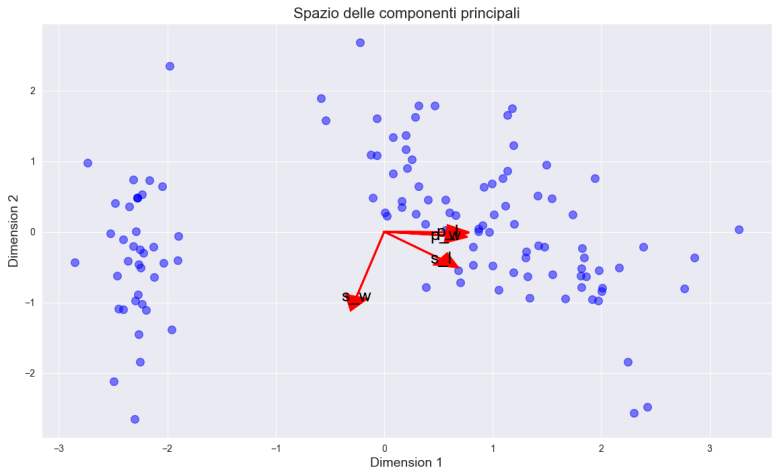
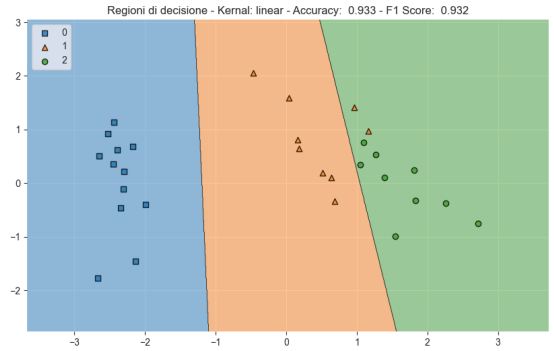
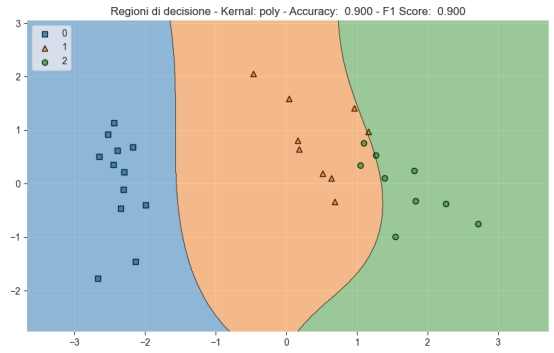


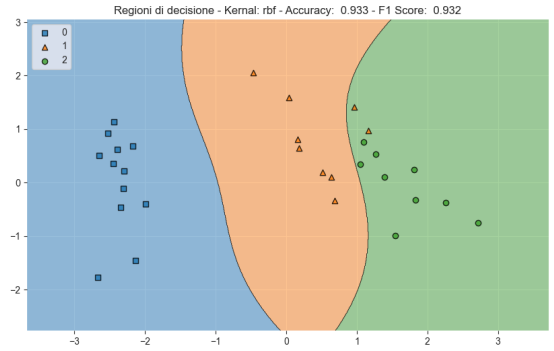

*Figura 7 - Tanto più un vettore è parallelo ad una componente principale, quanto più quest'ultima spiega la relativa feature. Anche il modulo del vettore fornisce un'indicazione circa l'entità della varianza spiegata: tanto più è maggiore quanto più spiega la relativa varianza. Infine gli angoli tra ogni coppia di vettori permettono di capire se le relative features sono correlate.*

## Addestramento

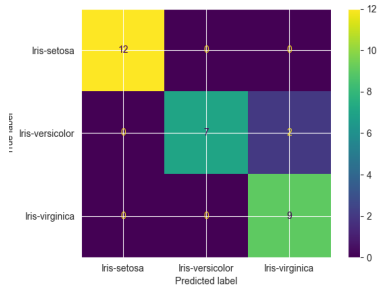
Per il training del modello si considerano solo le prime due PCs. Successivamente viene convertita la colonna target in valori di tipo numerico, assegnando ad ogni specie un numero intero. Infine si realizza un ciclo for che per ogni *kernel* effettua l'addestramento del modello, inoltre vengono anche calcolati gli indici di performance: accuratezza ed F1 score. Il primo misura il rate di classificazione ossia se un fiore viene correttamente classificato valutando l'output con il target conosciuto. Il secondo valuta la sensibilità e la precisione del modello che può anche essere analizzata graficamente con una matrice di confusione ossia una matrice in cui vengono mostrati i falsi/veri positivi ed i falsi/veri negativi. Di seguito si riportano i grafici delle regioni di decisione relativi ad ogni *kernel*.

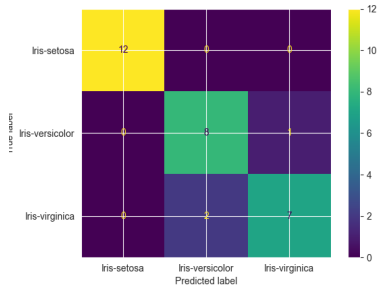


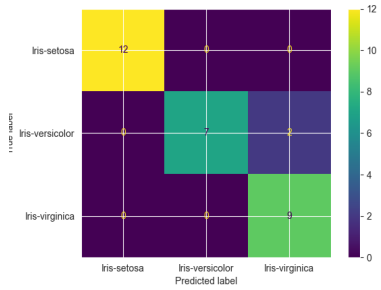




Di seguito invece si riportano le matrici di confusione relative.




# Conclusioni

La riduzione della dimensionalità non ha compromesso i risultati del modello. Inoltre, come ci si poteva aspettare dai risultati ottenuti durante la parte di statistica descrittiva, le distribuzioni pressocchè normali dei dati consentono di ottenere modelli di ottime performance, persino senza effettuare fine tuning degli iperparametri del modello.