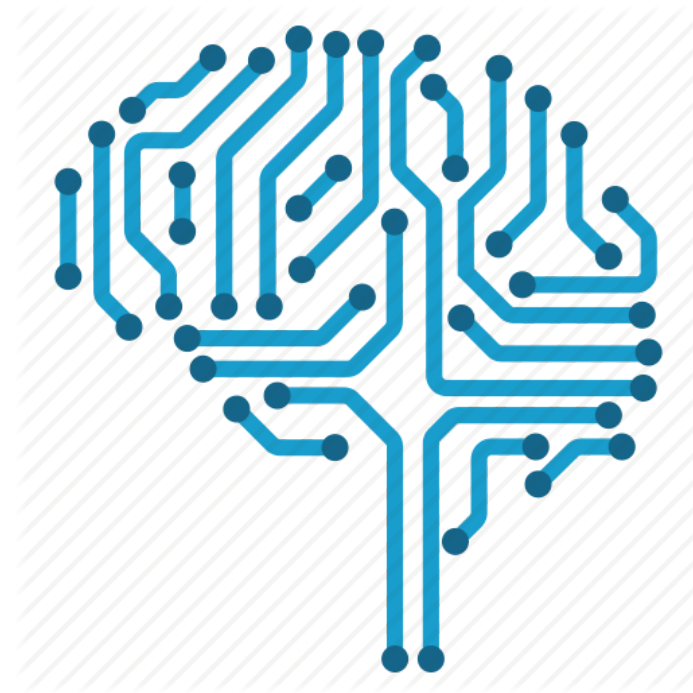




Big Data e Business Intelligence



Machine Learning

Giulio Angiani - UniPr



Machine Learning - primi passi

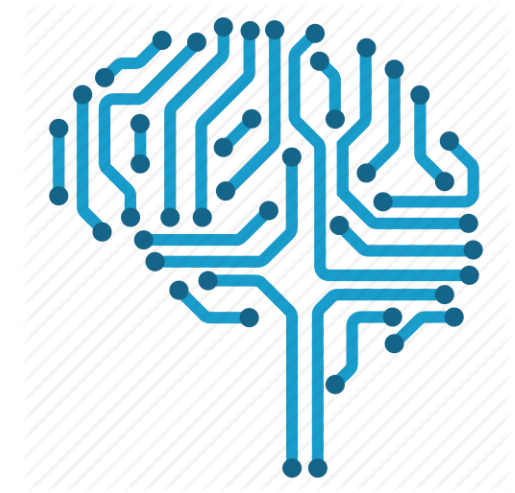
Machine Learning

Cos'è ML

- Capacità di un algoritmo di **prendere decisioni** sulla base di una knowledge-base (base di conoscenza) e di **apprendere** nuove informazioni sulla base dell'esperienza (decisioni prese precedentemente)

3 tipologie fondamentali

- Supervised learning (apprendimento supervisionato)
- UnSupervised learning (apprendimento non supervisionato)
- Semi-Supervised learning (apprendimento semi-supervisionato)



Machine Learning

*L'**apprendimento supervisionato** è una tecnica di apprendimento automatico che mira a istruire un sistema informatico in modo da consentirgli di elaborare automaticamente previsioni sui valori di uscita di un sistema rispetto ad un input sulla base di una serie di esempi ideali, costituiti da coppie di input e di output, che vengono inizialmente forniti¹*

- insieme di dati di addestramento (**training set**)
- insieme di dati di confronto (**test set**)
- **labeling** (annotazione) e **classi**

training set contiene informazioni **labeled** per permettere all'algoritmo di trovare il modo **migliore** per indovinare più casi possibile.

il **labeling** permette all'algoritmo di imparare a **discernere** un esempio dagli altri

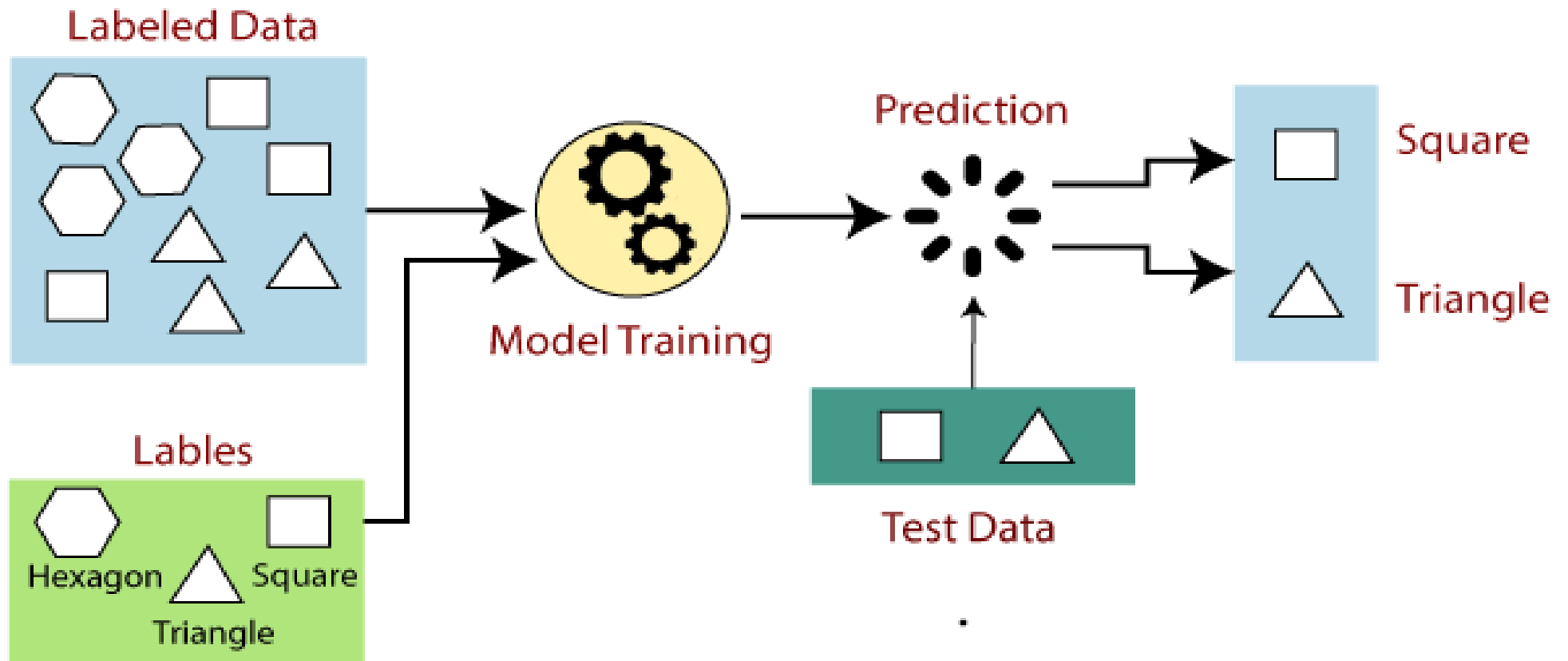
test set contiene informazioni del tutto **simili** per poter calcolare l'**accuratezza** dell'algoritmo addestrato



Machine Learning

Un esempio di **apprendimento supervisionato**

- dati forniti con labels => il modello può prevedere/assegnare una classe

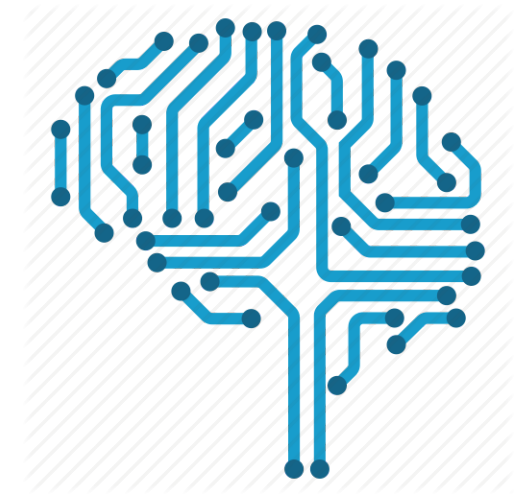


Machine Learning

*L'**apprendimento non supervisionato** è una tecnica di apprendimento automatico che consiste nel fornire al sistema informatico una serie di input (esperienza del sistema) che egli riclassificherà ed organizzerà sulla base di caratteristiche comuni per cercare di effettuare ragionamenti e previsioni sugli input successivi.*

*Al contrario dell'apprendimento supervisionato, durante l'apprendimento vengono forniti all'apprendista solo esempi **non annotati**, in quanto le classi non sono note a priori ma devono essere apprese automaticamente.¹*

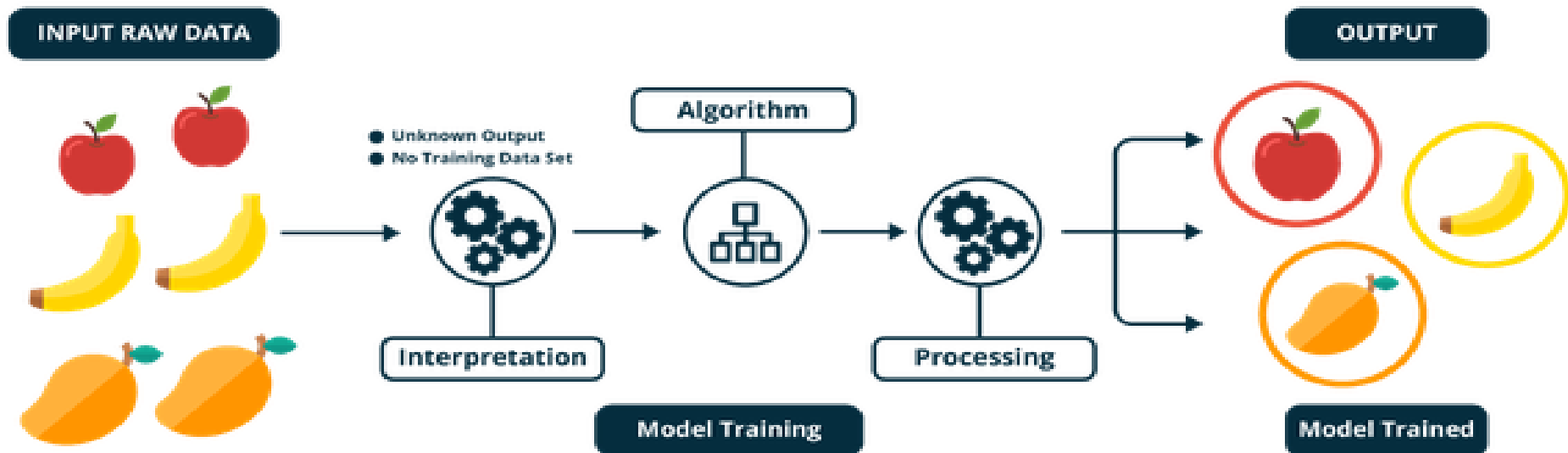
- insieme di dati di analisi (**data set**)
- **clustering** (raggruppamento)



Machine Learning

Un esempio di **apprendimento non supervisionato**

- dati raw, no labels => il modello può raggruppare gli items per similitudine



Machine Learning

*L'**apprendimento semi-supervisionato** è una tecnica di apprendimento automatico che combina una grande quantità di dati **non etichettati** con una piccola quantità di dati **etichettati**.¹*

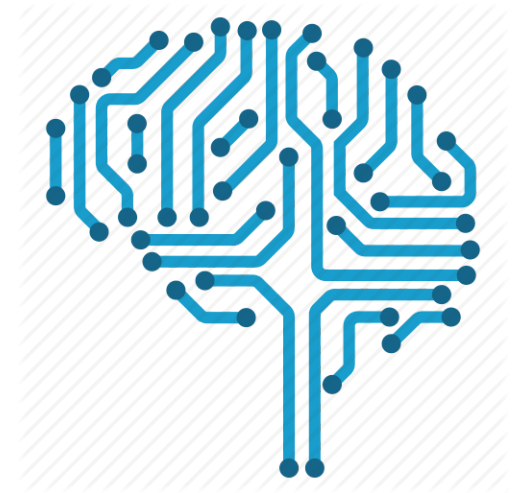
L'uso dell'apprendimento **non supervisionato** insieme a quello **supervisionato** permette all'algoritmo di clusterizzare gli esempi e poi di assegnare a **tutti gli elementi** di un certo gruppo, la **label** di quelli etichettati presenti nel gruppo

pros

- permette il labeling automatico di grandi quantità di dati altrimenti non etichettabili

cons

- i dati al **confine** fra due gruppi potrebbero avere etichette di entrambi
- introduce un po' di **bias** nel training



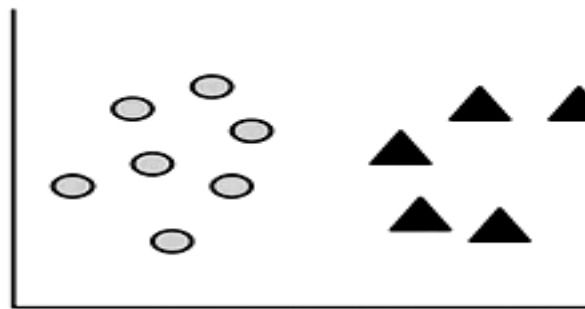
1) https://en.wikipedia.org/wiki/Semi-supervised_learning

Machine Learning

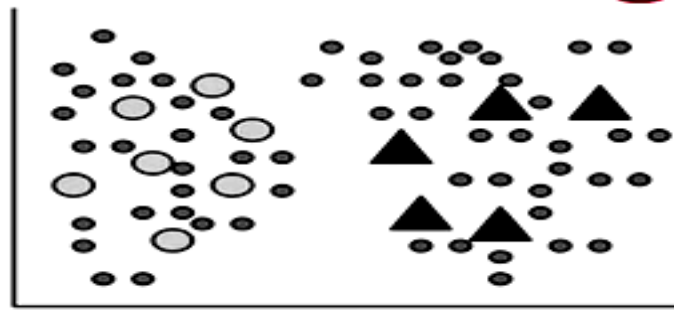
Un esempio di **apprendimento non supervisionato**

- dati raw, no labels => il modello può raggruppare gli items per similitudine

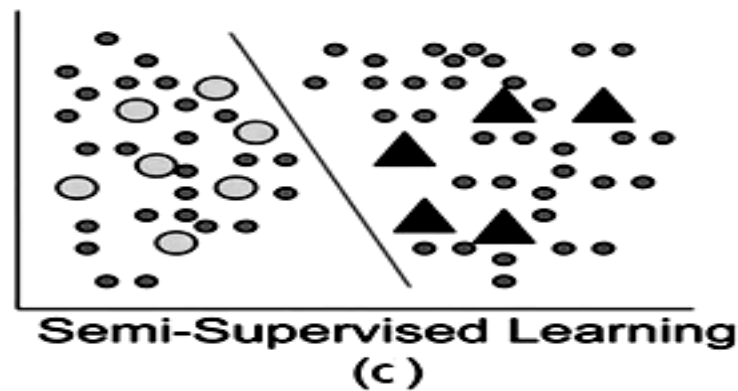
Semi-supervised learning



Labeled Data
(a)



Labeled and Unlabeled Data
(b)

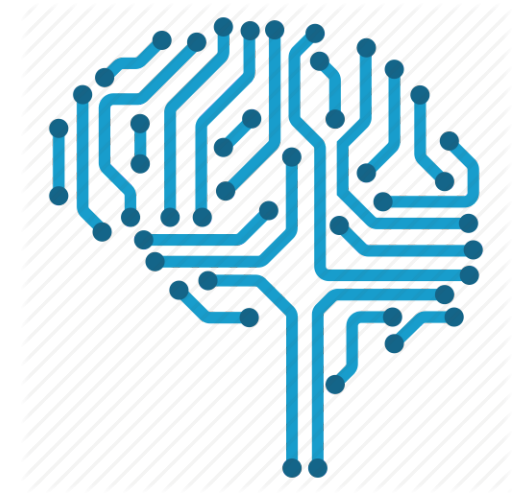


Semi-Supervised Learning
(c)

Machine Learning

Cos'è ML

- Vari tipi di Apprendimento automatico
 - Classificazione
 - Regressione
 - Clustering
- Tecniche
 - Alberi decisionali
 - Reti neurali
 - Tecniche statistiche
 - Regole di induzione



Machine Learning

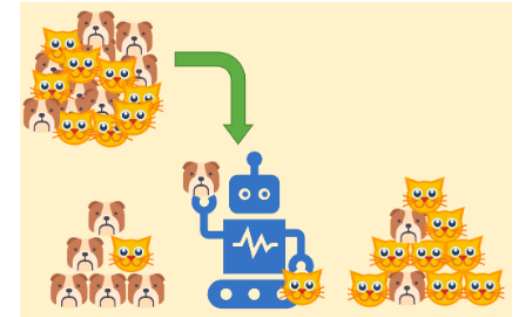
Classificazione

Nel machine learning, la **classificazione** è un concetto di **supervised learning** che sostanzialmente permette di **categorizzare** un insieme di dati in **classi**

I problemi più noti affrontabili con algoritmi di classificazione sono

- riconoscimento vocale
- riconoscimento facciale
- interpretazione della scrittura a mano
- classificazione dei documenti
- riconoscimento di immagini
- etc...

Esistono numerosi algoritmi di classificazione nella pratica del ML

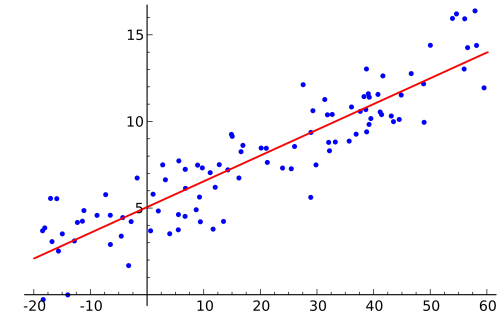


Machine Learning

Regressione

Nel machine learning, la **regressione** è un modello di calcolo **statistico** che, a differenza della classificazione, non assegna una classe ad ogni item esaminato ma assegna un **valore reale** stimato

E' quindi possibile, e quasi sempre vero, che items diversi vengano associati a **valori** reali diversi.



Il calcolo statistico è tipicamente il risultato di un algoritmo di **minimizzazione di errore**

la **regressione** fa sempre riferimento all'apprendimento **supervisionato**

I problemi più noti affrontabili con algoritmi di classificazione sono

- previsioni temperature meteo
- previsioni andamento azioni di borsa
- stima della capacità di spesa di clienti
- etc...

Esistono numerosi algoritmi di classificazione nella pratica del ML

Machine Learning

Clustering

Nel machine learning, il **clustering** è un modello di calcolo che ha lo scopo di **raggruppare** gli items analizzati in gruppi con caratteristiche simili.

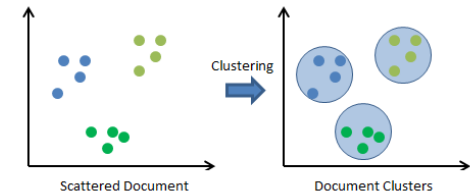
Il calcolo effettuato per determinare le **similitudini** fra items è spesso la **distanza** in qualche spazio n-dimensionale

Il modo con il quale è calcolata tale distanza va sotto il nome di **metrica**

La scelta della metrica influenza fortemente il risultato della cluterizzazione

Il **clustering** fa riferimento ad analisi di apprendimento **non supervisionato**

E' di supporto in algoritmi **semi-supervised**



Machine Learning

Tecniche per la classificazione

- Alberi decisionali
- Reti neurali
- Tecniche statistiche
- Regole di induzione

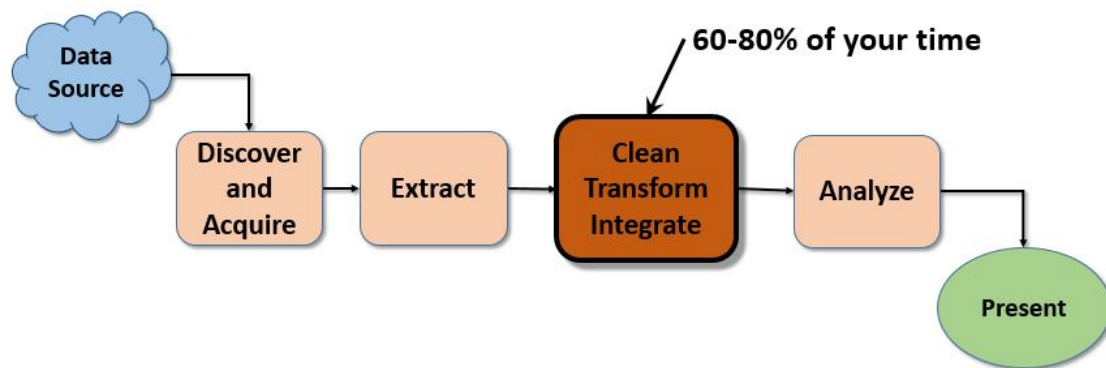


Machine Learning

I cinque passi della scienza dei dati

- Porre una domanda interessante
- **Ottenere i dati**
- **Esplorare i dati**
- **Creare un modello per i dati**
- Comunicare e presentare i risultati

Data Processing Pipeline



Martin Magdinier, RefinePro

Machine Learning

Strumenti software

Linguaggio di riferimento : **Python3.x**

Librerie Python :

- Scikit-learn
- Pandas
- Numpy
- Matplotlib

Jupyter Notebook

Machine Learning

I' **Hello world!** del ML

Problema di classificazione

- **apprendimento supervisionato**

Iris Data Set

Petal length, **Petal** width, **Class**

1.4, 0.2, **Iris**-setosa

1.4, 0.2, **Iris**-setosa

...

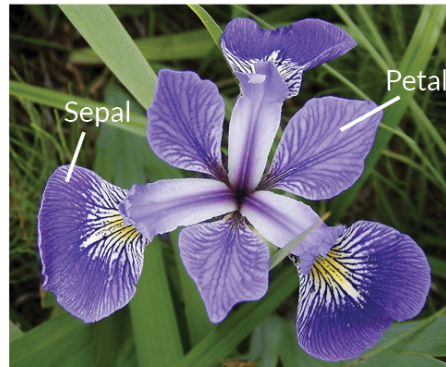
5.0, 1.7, **Iris**-versicolor

4.5, 1.5, **Iris**-versicolor

...

5.4, 2.3, **Iris**-virginica

5.1, 1.8, **Iris**-virginica



Iris Versicolor



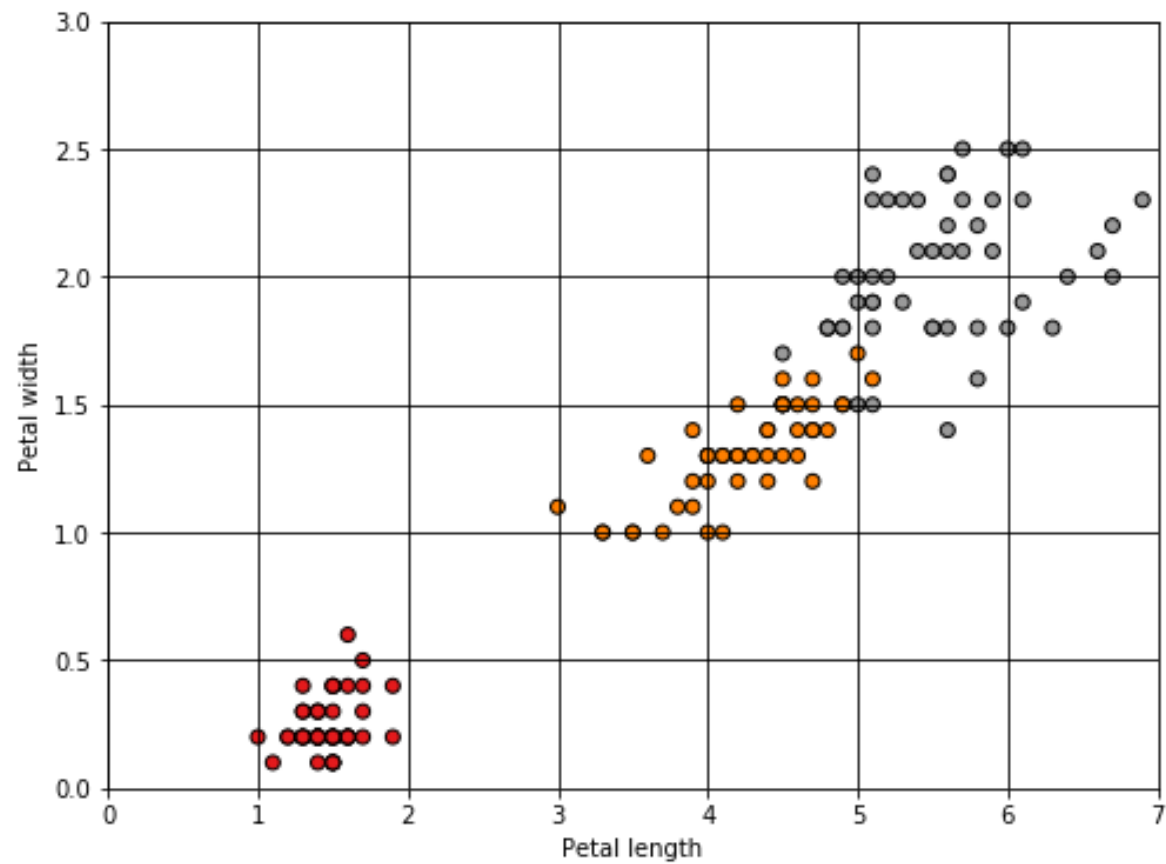
Iris Setosa



Iris Virginica

Iris 2D - distribuzione

distribuzione degli items come punti su un piano



Iris-setosa Iris-versicolor Iris-virginica

Iris 2D - Addestramento

formalismo standard

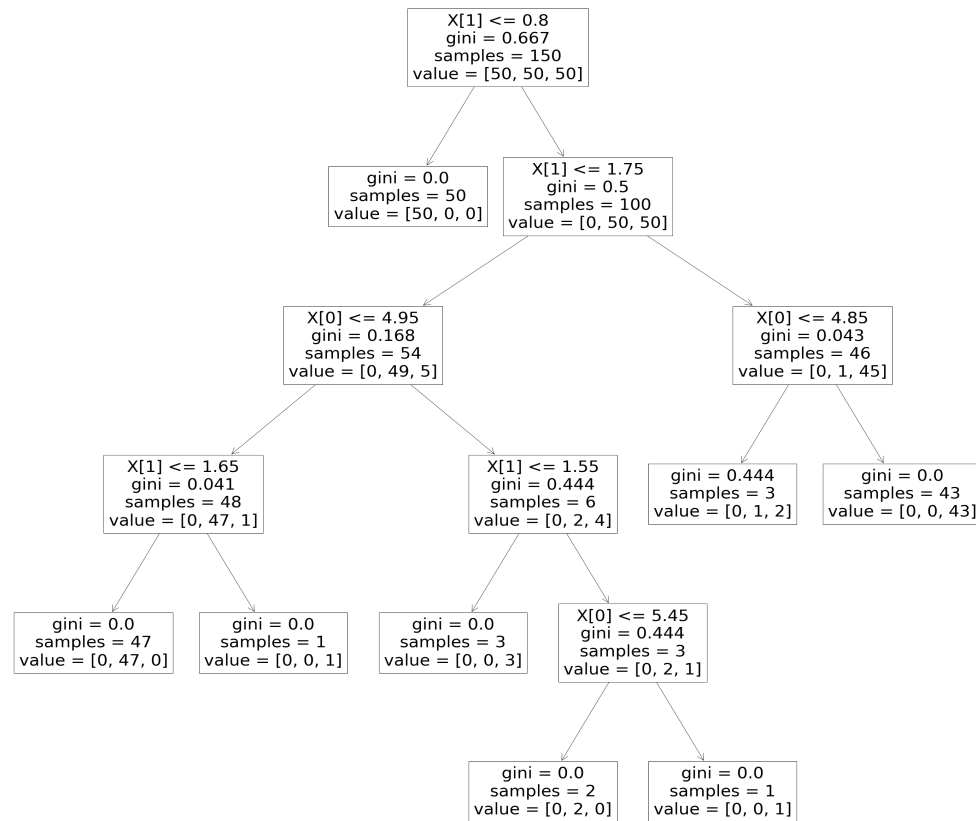
Si indica con **X** l'insieme dei valori delle colonne delle features Si indica con **y** la colonna delle classi

```
# importo il modulo tree da sklearn (scikit-learn) libreria per ML  
from sklearn import tree  
# fra gli alberi scelgo un albero di decisione  
clf = tree.DecisionTreeClassifier()  
# l'operazione di fit addestra il classificatore coi dati presenti in X e y  
clf = clf.fit(X, y)
```

PYTHON

Iris 2D - Albero decisionale

Cosa costruisce l'operazione di **fit** di un **DecisionTreeClassifier**

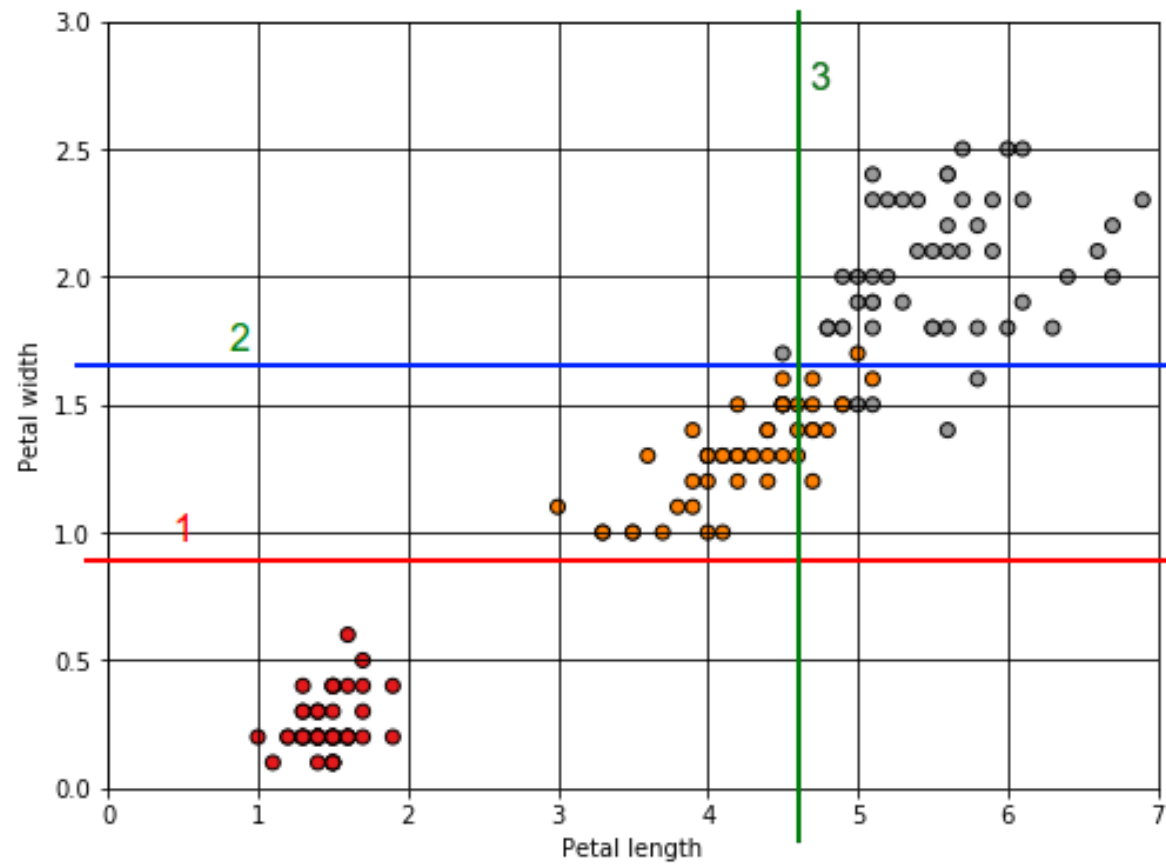


Iris 2D - Albero decisionale (testo)

```
|--- petal_width <= 0.80',
| |--- class: 0',
|--- petal_width > 0.80',
| |--- petal_width <= 1.75',
| | |--- petal_length <= 4.95',
| | | |--- petal_width <= 1.65',
| | | | |--- class: 1',
| | | |--- petal_width > 1.65',
| | | | |--- class: 2',
| | |--- petal_length > 4.95',
| | | |--- petal_width <= 1.55',
| | | | |--- class: 2',
| | | |--- petal_width > 1.55',
| | | | |--- petal_length <= 5.45',
| | | | | |--- class: 1',
| | | | |--- petal_length > 5.45',
| | | | | |--- class: 2',
| |--- petal_width > 1.75',
| | |--- petal_length <= 4.85',
| | | |--- class: 2',
| | |--- petal_length > 4.85',
| | | |--- class: 2',
```

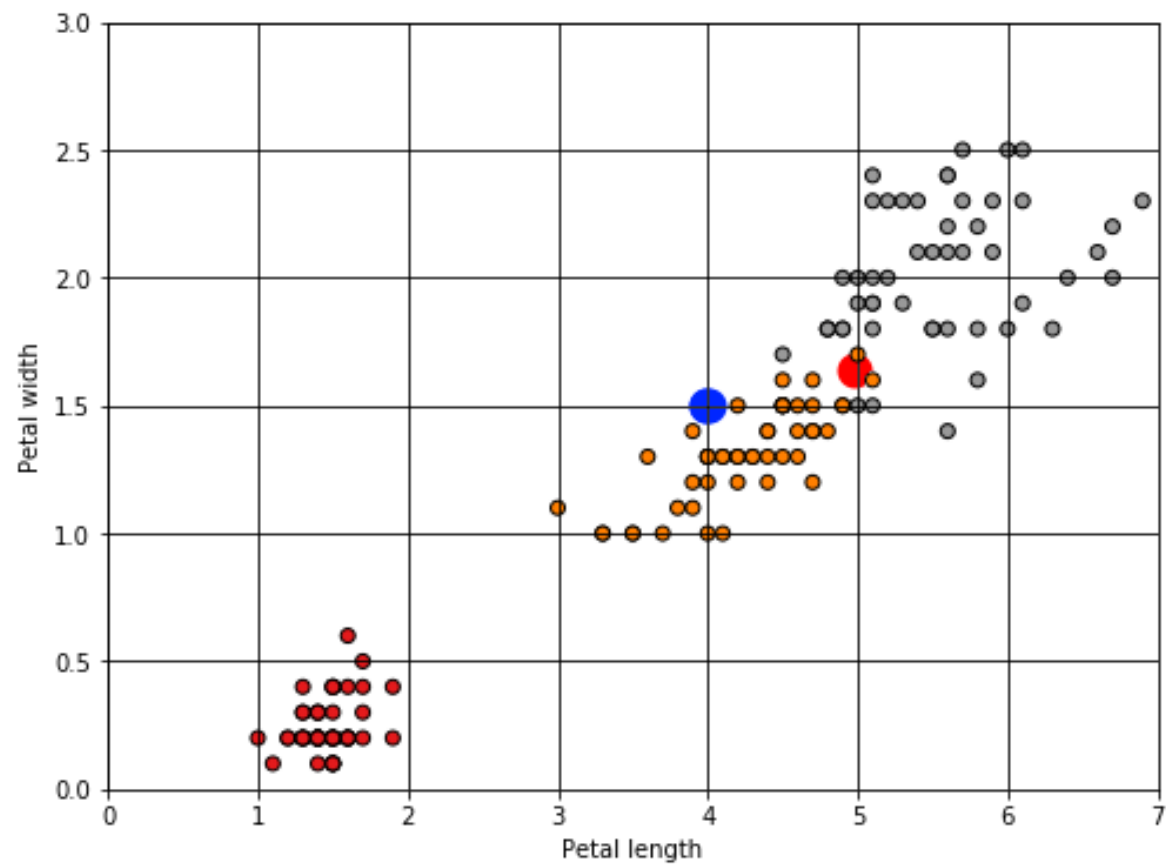
Iris 2D - Albero decisionale (grafica)

tagli decisionali dell'algoritmo J48



Iris 2D - predizione

- prendiamo altri items
- per esempio: (4, 1.5) e (5, 1.6)
- dove si posizionano?



Iris 2D - predizione

- usiamo sklearn

```
clf.predict([[4, 1.5]])
```

PYTHON

```
array([1])
```

OUTPUT

- Predizione sulla classe 2 (Iris-versicolor)

```
clf.predict([[5, 1.6]])
```

```
clf.predict_proba([[5, 1.6]])
```

PYTHON

```
array([1]) # classe 2
```

```
array([[ 0.,  1.,  0.]]) # 100% confidenza
```

OUTPUT

Iris 2D - predizione

- un esempio più distante dal dataset (outlier)

```
clf.predict([[2, 2]])  
clf.predict_proba([[2, 2]])
```

PYTHON

```
array([2]) # classe 2  
array([[ 0,  0.33333333, 0.66666667]]) # 33% e 66% di confidenza
```

OUTPUT

Iris 2D - SVM

PYTHON

```
# cambiamo classificatore
# Support Vector Machine
from sklearn.svm import SVC
clf = SVC(gamma='auto', probability=True)
clf.fit(X, y)
print(clf.predict([[4, 1.5]]))
print(clf.predict_proba([[4, 1.5]]))
```

OUTPUT

```
array([1])
array([[ 0.00839442,  0.97857937,  0.01302621]])
```

- la **confidence** è spalmata sulle tre classi
- confidence = livello di affidabilità della predizione
- valore reale in $[0,1]$

Iris 2D - Valutazione del classificatore

Le modalità di valutazione sono diverse a seconda del tipo di algoritmo

i modelli di classificazione

i modelli di regressione

i modelli di clustering

[cit. slides Mordonini]

- Dataset Split
- Cross Validation

Iris 2D - Valutazione del classificatore

Dataset Split

- tipicamente
 - 66% train dataset
 - 33% test dataset
- nel nostro caso: 150 elementi
 - 100 training
 - 50 test
- **classi bilanciate**



Iris 2D - Valutazione del classificatore

- Dividiamo il dataset (X,y) in [(X_train, y_train) e (X_test, y_test)]

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
print(len(X_train), len(y_train))
print(len(X_test), len(y_test))
```

PYTHON

```
100 100
50 50
```

OUTPUT

- addestriamo sul **solo** training set

```
clf.fit(X_train, y_train)
```

PYTHON

Iris 2D - Valutazione del classificatore

- Testiamo il classificatore addestrato sul **solo** training set
- sia sullo stesso **training set** che sul **test set**

```
print("TRAIN SET", clf.score(X_train, y_train))  
print("TEST SET", clf.score(X_test, y_test))
```

PYTHON

```
TRAIN SET 0.99 # errore di 1%  
TEST SET 0.98 # errore di 2%
```

OUTPUT

Iris 2D - Valutazione del classificatore

- quali sono gli errori?
- concetto di **accuracy** - accuratezza della classificazione

```
print("Errori in training set")
predictions = clf.predict(X_train)
for elem, prediction, label in zip(X_train, predictions, y_train):
    if prediction != label:
        print(elem, 'has been classified as ', prediction, 'and should be ', label)
# similmente per test set...
```

PYTHON

```
Errori in training set
[ 4.8  1.8] has been classified as  2 and should be  1

Errori in test set
[ 5.1  1.5] has been classified as  1 and should be  2
```

OUTPUT

Iris 2D - Valutazione del classificatore

- Matrice di confusione
 - quanti elementi di una certa classe sono associati alle altre classi

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_train, clf.predict(X_train))
print("CM per Train set\n", cm)
# similmente per test set...
```

PYTHON

CM per Train set

```
[[31  0  0]
 [ 0 34  1]
 [ 0  0 34]]
```

CM per Test set

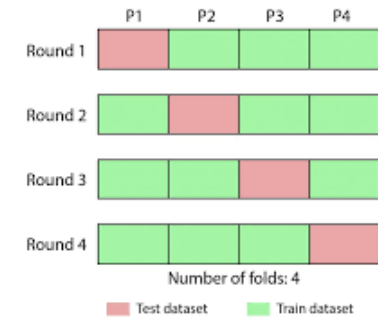
```
[[19  0  0]
 [ 0 15  0]
 [ 0  1 15]]
```

OUTPUT

Iris 2D - Valutazione del classificatore

Cross Validation

- Quando? Se abbiamo pochi dati e non possiamo permetterci di splittare fra train e test set
- parametri
 - folds (numero di blocchi in cui suddivido il mio dataset)



```
from sklearn.model_selection import cross_val_score
print(cross_val_score(clf, X, y, cv=4))
```

PYTHON

```
[ 0.97435897  0.94871795  0.91666667  0.97222222]
```

OUTPUT

- media di accuracy: **0.95**

Iris 2D - Valutazione del classificatore

- si lascia per esercizio al lettore
 - addestramento di due classificatore SVM e RF (Random Forest)
 - valutazione dell'accuratezza con CV
 - valutazione dell'accuratezza con Dataset Split
 - visualizzazione matrici di confusione
 - confronto prestazioni fra SVM e RF



Giulio Angiani
Universita' degli Studi di Parma