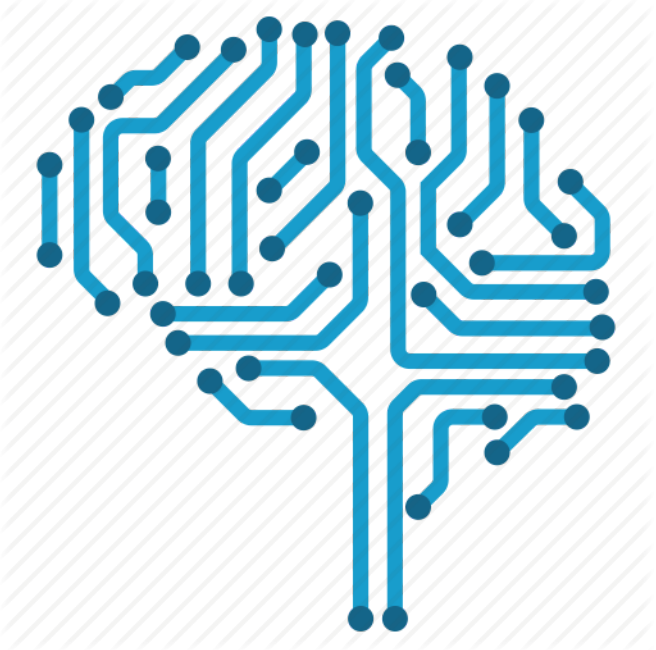
Big Data e Business Intelligence



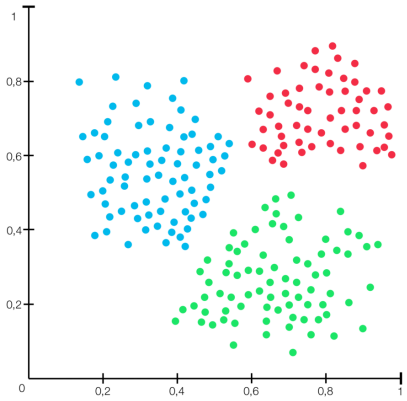
Machine Learning

Giulio Angiani - UniPr

Machine Learning
Clustering e applicazioni

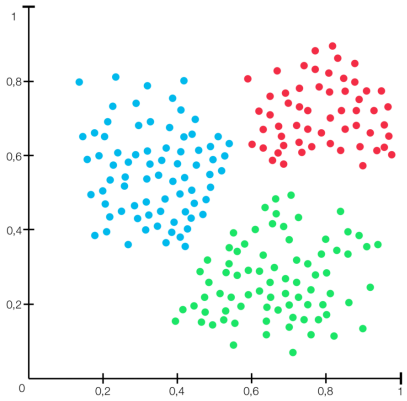
Clustering

- Cos'è: insieme di tecniche di **analisi** dei dati volte alla individuazione di elementi comuni nella popolazione in esame
- Obiettivo: selezione e **raggruppamento** di elementi omogenei in un insieme di dati
- Metodologia: Misure di **somiglianza** tra gli elementi
  - spesso distanza in uno spazio multidimensionale
- Molto dipendente dalla scelta della **metrica**
- Appartenenza ad un insieme è funzione della **distanza** da elementi dell'insieme



Clustering - Algoritmi

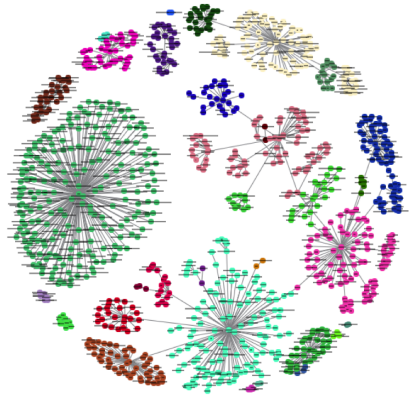
Vari algoritmi di clustering



- Partition-based clustering
  - Dato k, partiziona gli esempi in k cluster di almeno un elemento; ogni esempio può appartenere solo ad un elemento.
- Hierarchical clustering
  - Scompone l'insieme degli esempi in una gerarchia di partizioni di diversa complessità.
- Density-based clustering
  - Gli esempi vengono suddivisi in cluster via via sempre più numerosi fino a quando la "densità" di ogni cluster rimane accettabile.

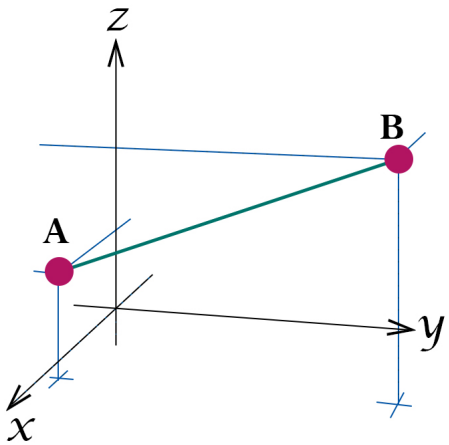
Clustering - Utilizzi

- Ricerche di mercato.
- Riconoscimento di pattern.
- Raggruppamento di clienti in base ai comportamenti d'acquisto (segmentazione del mercato).
- Posizionamento dei prodotti.
- Analisi dei social network, per il riconoscimento di community di utenti.
- Identificazione degli outliers



Clustering - Misure di similarità

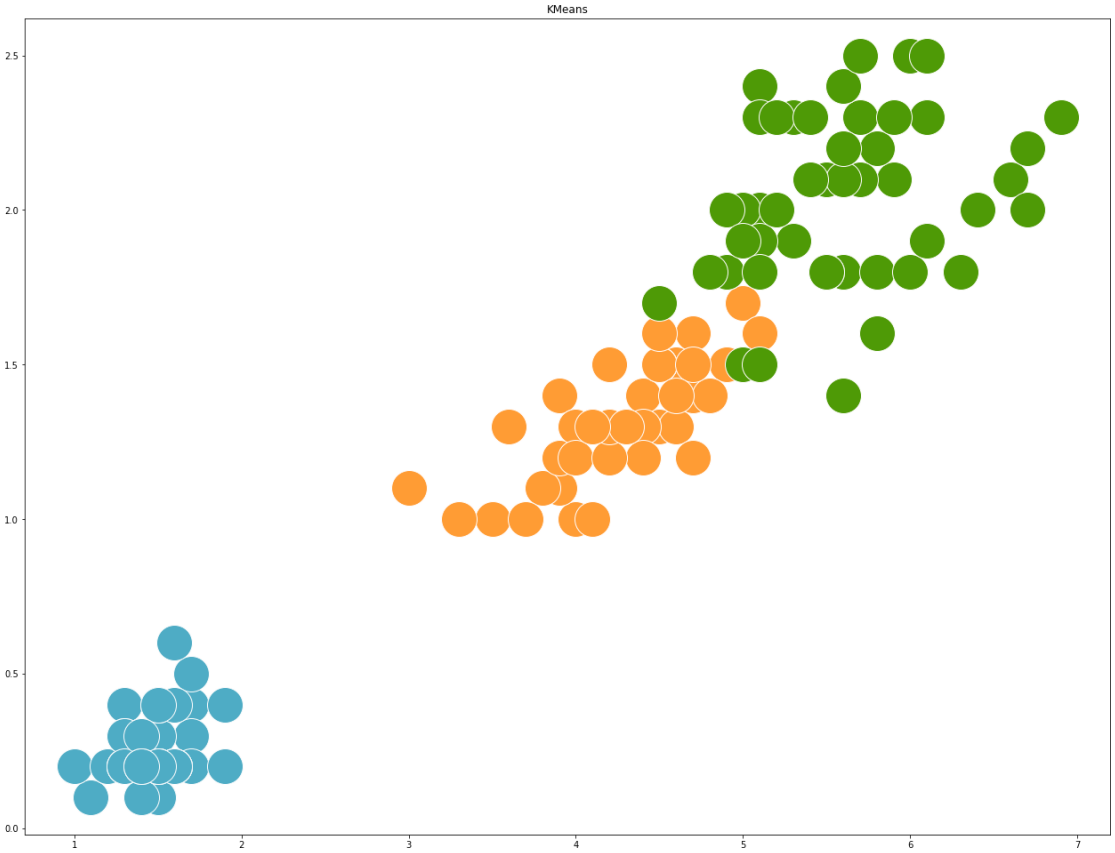
- Una delle misure più semplici è la **distanza euclidea** tra due punti in uno spazio n-dimensionale
- Distanza di Minkowski (simile a euclidea o manhattan)
- Simple Matching Coefficient
- Coefficiente di Jaccard
- Correlazione di Pearson



$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{i\,k} - x_{j\,k})^2}$$

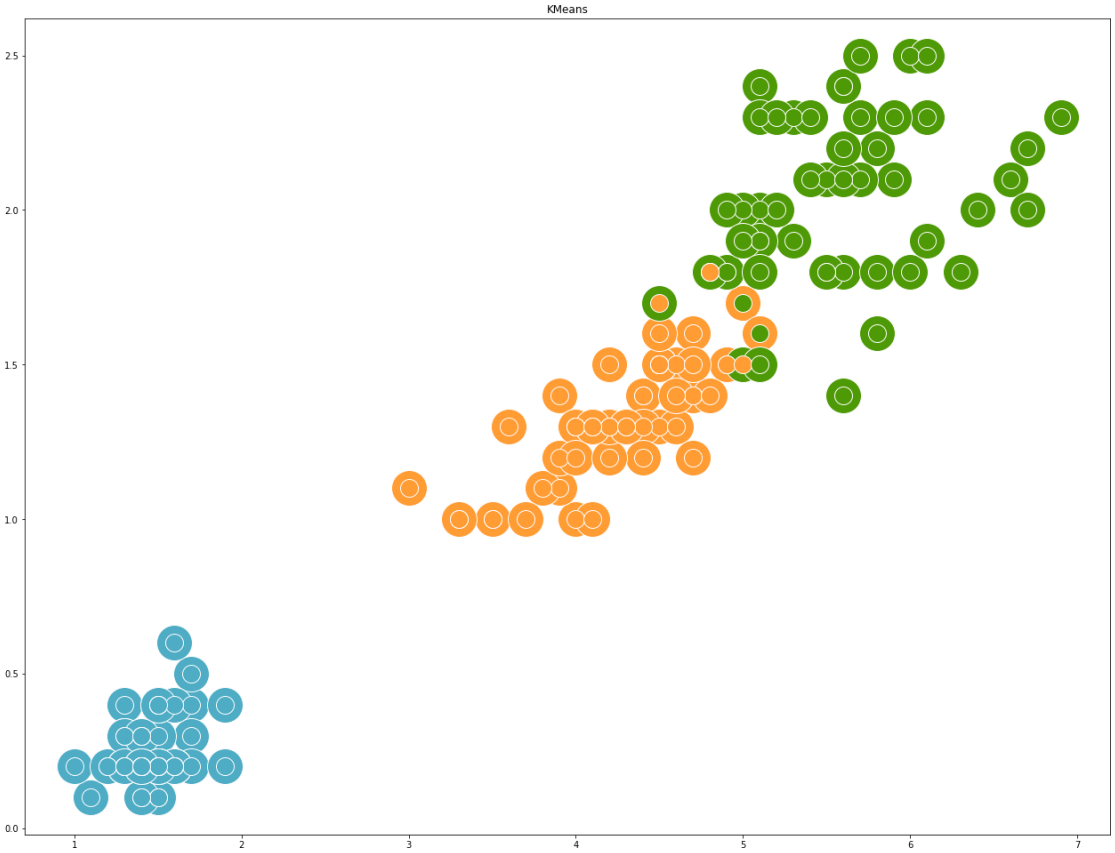
Clustering - Esempio

- riprendiamo IRIS Data Set



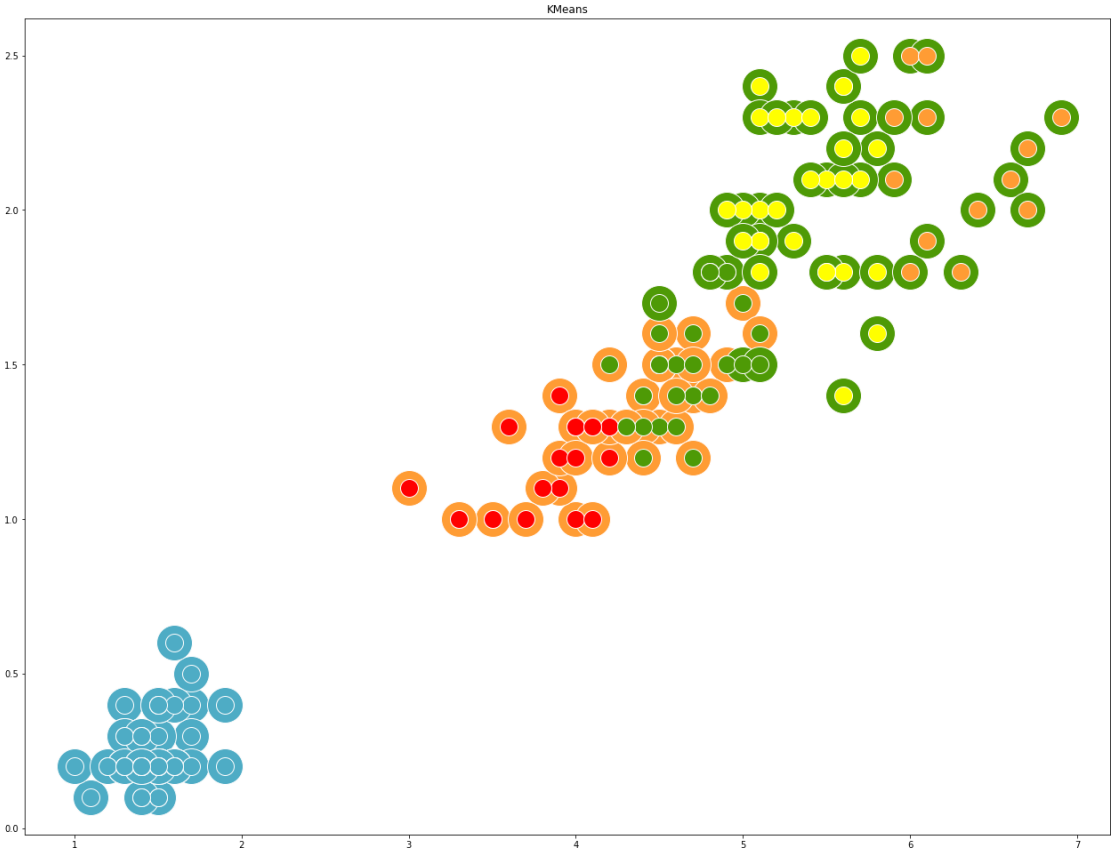
Clustering - Esempio

- 3 Cluster in IRIS Data Set



Clustering - Esempio

- 5 Cluster in IRIS Data Set



Machine Learning
Valutazione

Valutazione

Obiettivo

- Valutare la bontà di un classificatore
- Conoscere le features più significative
- Testare la validità del classificatore con meno features

Valutazione

- riprendiamo IRIS dataset (4 features)

```
df = pd.DataFrame(iris.data)
df.columns = iris.feature_names
df.head()
```

```
   sepalL  sepalW  petalL  petalW
0   5.1     3.5     1.4     0.2
1   4.9     3.0     1.4     0.2
2   4.7     3.2     1.3     0.2
3   4.6     3.1     1.5     0.2
4   5.0     3.6     1.4     0.2
```

Valutazione

- features selection

PYTHON

```
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)
clf.fit(X_train, y_train)
clf.feature_importances_    # contributo nella decisione di ogni feature
```

OUTPUT

```
array([ 0.01256535,  0.04005207,  0.06894128,  0.87844129])
```

PYTHON

```
from sklearn.feature_selection import SelectKBest, f_classif
select = SelectKBest(f_classif,k=2)
select.fit(X,y)
mask = select.get_support()
print(mask) # presente o non presente nelle 2 più significative
```

OUTPUT

```
[False False  True  True]
```

Ridimensionamento

- proiezione del data set sulle features più significative

PYTHON

```
X_new = iris.data[:, :2]  # we only take the first two columns.
y_new = iris.target
```

DATASET

	sepalL	sepalW
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Valutazione

- **Stessa configurazione** del classificatore su dataset ridotto

PYTHON

```
X_train_new, X_test_new, y_train_new, y_test_new =
    train_test_split(X_new, y_new, test_size=0.33, random_state=42)
clf_new = tree.DecisionTreeClassifier()
clf_new.fit(X_train_new, y_train_new)
print("TEST SET (2  feats)", clf_new.score(X_test_new, y_test_new))
```

OUTPUT

```
TEST SET (2  feats) 0.66
```

Giulio Angiani
Universita' degli Studi di Parma