Clustering studenti informatica

Tommaso Ceccarini, Filippo Mameli

17 agosto 2018

Introduzione

Il dataset

Il dataset che abbiamo analizzato contiene dati sulle carriere accademiche degli studenti del corso di laurea di informatica dell'università degli studi di Firenze e il loro voto conseguito al test di ingresso.

- Coorte: Anno di immatricolazione
- Crediti totali: Numero crediti complessivi dello studente
- Crediti con voto: Numero di crediti assegnati allo studente per esami con votazione in trentesimi (tutti tranne Inglese)
- Voto medio: Media pesata dei voti degli esami sostenuti

Il dataset

- Nome dell'esame
- Data in cui lo studente ha sostenuto l'esame

Gli esami sono Algoritmi e strutture dati (ASD), Programmazione (PRG), Architetture degli elaboratori (ARC), Analisi I (ANI), Matematica discreta e logica (MDL) e Inglese.

- Punteggio conseguito al test di ingresso.

La gestione dei dati

Le principali operazioni effettuate sul dataset sono:

- eliminare gli studenti che hanno sostenuto solo inglese
- riportare tutti gli attributi relativi alle date degli esami nel formato YYYY-MM-DD

La gestione dei dati

Le principali operazioni effettuate sul dataset sono:

- eliminare gli studenti che hanno sostenuto solo inglese
- riportare tutti gli attributi relativi alle date degli esami nel formato YYYY-MM-DD

Creazione table

```
CREATE TABLE 'studenti' (
  'coorte' int(11),
  'crediti_totali' int(11),
  'crediti_con_voto' int(11),
  'voto_medio' int(11),
  'ASD' int(11),
  'data_ASD' text,
    ...
  'data_INGLESE' text,
  'TEST' int(11)
) ENGINE=InnoDB

LOAD DATA INFILE 'studenti.csv' INTO TABLE studenti
  FIELDS TERMINATED BY ',' ENCLOSED BY '"'
  LINES TERMINATED BY '\r\n'
  IGNORE 1 LINES;
```

Update tabella

```
update dmo.studenti set data_ARC = '0000-00-00' where
    data_ARC='0';
update dmo.studenti set data_ASD = '0000-00-00' where
    data_ASD='0';
update dmo.studenti set data_PRG = '0000-00-00' where
    data_PRG='0';
update dmo.studenti set data_ANI = '0000-00-00' where
    data_ANI='0';
update dmo.studenti set data_MDL = '0000-00-00' where
    data_MDL='0';
update dmo.studenti set data_INGLESE = '0000-00-00' where
    data_INGLESE = '0';
```

Analisi dei dati

Valutazione del clustering e model selection

- Selezione del numero "ottimale" di cluster per il K-means
- Valutazione del K-means
- Valutazione DBSCAN

Selezione numero di cluster nel K-means

Viene effettuata tramite la seguente procedura

- Determinazione SSE in funzione di k
- Selezione del valore ottimale di k_{opt}

successivamente è possibile valutare e confrontare i risultati ottenuti dall'algoritmo con i diversi valori di k.

Example

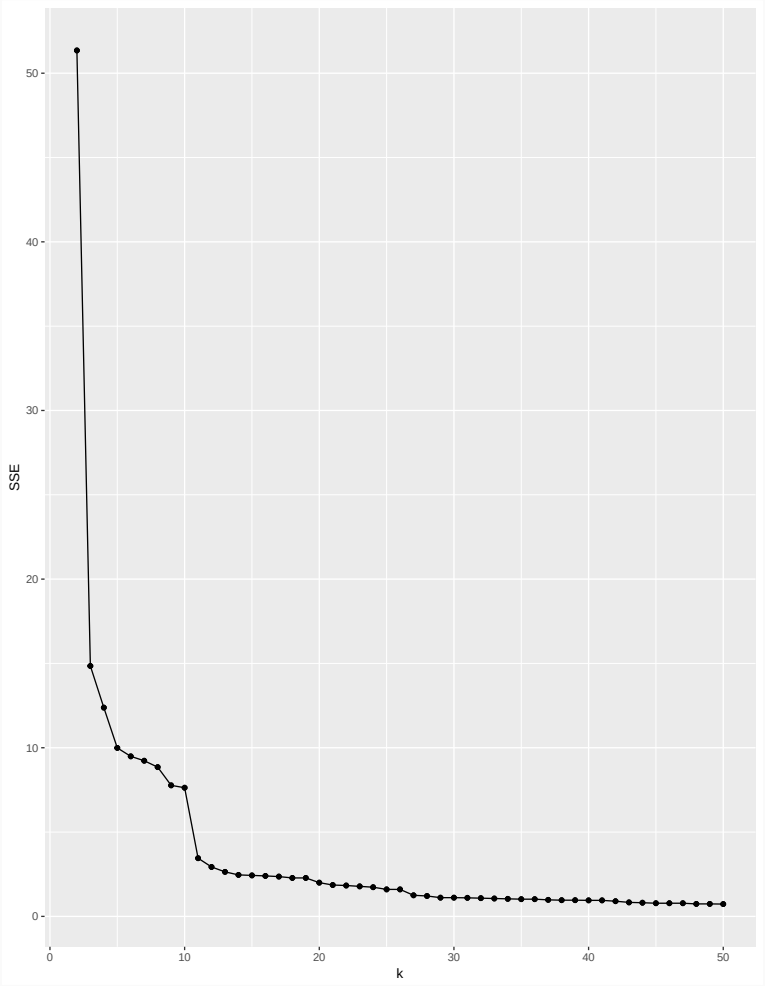

Figura 1: Dependency update

Example

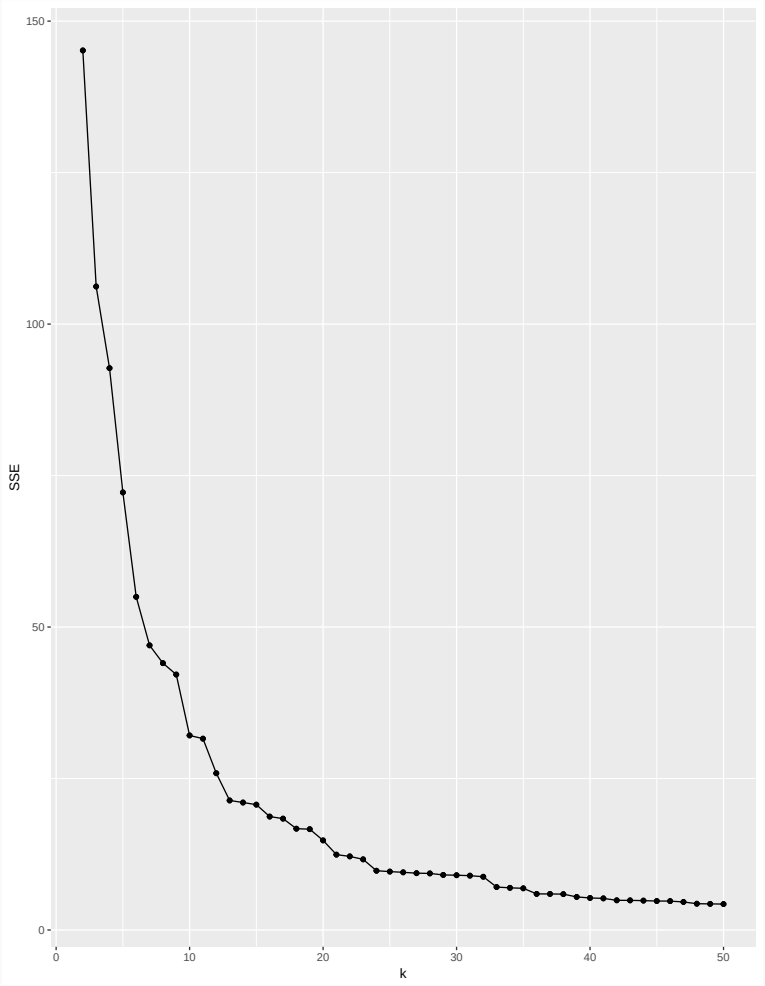

Figura 2: Dependency update

Example

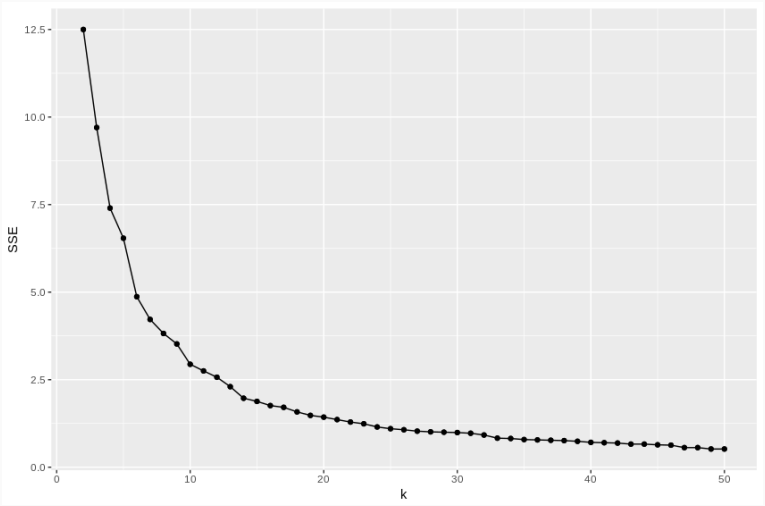

Figura 3: Dependency update

Selezione `Eps` fissato `MinPts` in DBSCAN

Viene effettuata tramite la seguente procedura

- Ordino i punti rispetto alla loro distanza dal loro k-esimo punto più vicino item pongo `MinPts`$=k$
- Determino un grafico con indici punti ordinati e distanze dal k-esimo più vicino
- Selezione come valore di `Eps` quello per cui c'è un picco.

Valutazione

La valutazione dei clustering ottenuti con K-means e DBSCAN è stata fatta con la seguente procedura

- Calcolo matrice distanze tra i punti
- Calcolo matrice di incidenza dei cluster
- "Serializzazione" e calcolo della correlazione

successivamente è possibile valutare e confrontare i risultati ottenuti dai clustering ottenuti con il K-means con i diversi valori di k e con il DBSCAN.

SQL

```
# Matrice di incidenza
matriceIncidenza <- function(data){
  nr = nrow(data)
  nc = ncol(data)
  C = matrix(nrow = nr, ncol = nr)
  for(i in 1:nr){
    for(j in 1:nr){
      if(data[i,nc] == data[j,nc])
        C[i,j] = 1
      else
        C[i,j] = 0
    }
  }
return(C)
```

SQL

```
# matrice distanza
matriceDistanza <- function(data){
  return(as.matrix(dist(data[,1:(ncol(data)-1)],method =
      'euclidean',diag = TRUE,upper = TRUE)))
}

calcoloCorrelazione <- function(data){
  MI <- matriceIncidenza(data)
  D <- matriceDistanza(data)
  mi = as.vector(t(MI))
  d = as.vector(t(D))

  return(cor(mi,d,method="pearson"))
}

calcoloCorrelazione(crediti_totali_prg_arc_clustered)
```