UNIVERSITÀ DEGLI STUDI DI FIRENZE

CURRICULUM DATA SCIENCE

DATA MINING AND ORGANISATION

Analisi carriere studenti iscritti al I anno del corso di laurea in Informatica

Studenti
TOMMASO CECCARINI
tommaso.ceccarini1@stud.unifi.it
FILIPPO MAMELI
filippo.mameli@stud.unifi.it

13 settembre 2018

Indice

1 Descrizione dati iniziali

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. In particolare, le informazioni presenti sono:

- 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

In seguito ci sono sei coppie di attributi che indicano:

- 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.

Per quanto riguarda gli attributi relativi ai voti degli esami i valori che questi assumono sono: il voto conseguito dallo studente nel caso in cui abbia sostenuto l'esame oppure zero se lo studente non ha sostenuto l'esame durante la sessione estiva del proprio anno di immatricolazione.

2 Gestione dei dati

Abbiamo deciso di effetture le seguenti operazioni sul dataset:

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

Il motivo per cui abbiamo deciso di eseguire l'operazione del primo punto è che gli studenti in questione non avendo voti in trentesimi non ci sono parsi particolarmente significativi per l'analisi.

Per quanto riguarda il secondo punto invece, nel caso in cui uno studete non avesse sostenuto un particolare esame, erano presenti valori pari a zero in alcuni casi e valori pari a "0000-00-00" in altri. Abbiamo quindi deciso di trattare questa incosisistenza dei dati ponendo i valori zero a "0000-00-00". Per effetturare queste due operazioni abbiamo importato il dataset in un database creando per prima una tabella, che è stata in seguito popolata importando il file CSV con i comandi riportati in Codice 1.

```
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,
  'ARC' int(11),
  'data_ARC' text,
  'PRG' int(11),
  'data_PRG' text,
  'ANI' int(11),
  'data_ANI' text,
  'MDL' int(11),
  'data_MDL' text,
  'INGLESE' int(11),
  'data_INGLESE' text,
  'TEST' int(11)
) ENGINE=InnoDB

LOAD DATA INFILE 'studenti.csv' INTO TABLE studenti
```

```
  FIELDS TERMINATED BY ',' ENCLOSED BY '"'
23  LINES TERMINATED BY '\r\n'
  IGNORE 1 LINES;
```

Codice 1: Creazione della table

Per quanto riguarda la gestione dell'incosisistenza relativa alle date degli esami abbiamo risolto il problema specificando le query riportate nel Codice 2.

```
update dmo.studenti set data_ARC = '0000-00-00' where data_ARC='0';
2 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';
4 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';
6 update dmo.studenti set data_INGLESE = '0000-00-00' where
    data_INGLESE = '0';
```

Codice 2: Update della tabella

3 Analisi dei dati

Per analizzare i dati a disposizione abbiamo utilizzato il linguaggio R per determinare la correlazione di Pearson tra i diversi attributi. In Tabella 1 sono riportati i valori di correlazione.

	coorte	crediti totali	crediti con voto	voto medio	ASD	ARC	PRG	ANI	MDL	ING	TEST
coorte	1	0.013343	0.01821	0.03655	0.03581	-0.01609	-0.0822	0.13386	-0.04033	NA	0.04126
crediti_totali	0.01334	1	0.99522	0.44571	0.52984	0.72508	0.69882	0.61015	0.62789	NA	0.38433
crediti_con_voto	0.01821	0.99522	1	0.44838	0.52957	0.71955	0.70879	0.61593	0.62654	NA	0.39025
voto_medio	0.03655	0.44571	0.44838	1	0.36900	0.36427	0.43085	0.39777	0.31828	NA	0.39428
ASD	0.03581	0.52984	0.52957	0.36900	1	0.29321	0.31192	0.10116	0.23775	NA	0.16149
ARC	-0.0160	0.72508	0.71955	0.36427	0.29321	1	0.43166	0.27541	0.39622	NA	0.29979
PRG	-0.0822	0.69882	0.70879	0.43085	0.31192	0.43166	1	0.19585	0.27295	NA	0.24356
ANI	0.13386	0.61015	0.61593	0.39777	0.10116	0.27541	0.19585	1	0.36333	NA	0.32378
MDL	-0.0403	0.62789	0.62654	0.31828	0.23775	0.39622	0.27295	0.36333	1	NA	0.38777
ING	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA
TEST	0.04126	0.384332	0.39025	0.39428	0.16149	0.29979	0.2435	0.32378	0.38777	NA	1

Tabella 1: Correlazione di Pearson

Come è possibile notare dalla tabella, ci sono alcune ovvie correlazioni, come quella che coinvolge l'attributo crediti totali e crediti con voto. Per quanto riguarda l'attributo coorte si può notare come questo abbia una scarsa correlazione con tutti gli altri attributi. Le correlazioni più evidenti e significative sono quelle che coinvolgono l'attributo crediti totali (e quindi anche crediti con voto) con il voto che gli studenti hanno ottenuto nei diversi esami sostenuti. Infatti il valore di queste correlazioni è compreso tra 0.53 per quanto riguarda l'attributo di crediti totali e quello di Algoritmi e strutture dati e 0.73 per quanto riguarda Crediti totali e il voto di Architetture degli elaboratori. Tale risultato può essere interpretato in maniera abbastanza intuitiva, infatti mostra che se uno studente ha sostenuto più esami (e quindi ha più crediti) ha in generale ottenuto dei voti migliori. Ciò è vero soprattutto per gli esami di Architetture e Programmazione che hanno una correlazione con l'attributo Crediti Totali di 0.73 e 0.7 rispettivamente e in maniera meno significativa per gli esami di Matematica Discreta e Logica e Analisi 1 che hanno una correlazione con l'attributo Crediti Totali di 0.63 e 0.61 rispettivamente. Invece, per quanto riguarda Algoritmi e strutture dati, la correlazione è significativamente inferiore rispetto agli altri esami (in particolar modo con Architetture e Programmazione). Analizzando questi valori è quindi possibile dedurre che una significativa porzione degli studenti che ha fatto l'esame di ASD non ha sostenuto altri esami o pochi altri (attributo Crediti Totali basso). Mentre

gli studenti che hanno sostenuto con profitto Programmazione e Architetture hanno superato anche tutti gli altri esami o la maggior parte di essi (attributo Crediti Totali alto). Infine i voti di Analisi 1 e Matematica rappresentano delle situzioni intermedie.

Nelle Figure 1 e 2 sono riportati gli scatterplot degli attributi relativi a crediti totali con i voto di Architetture degli elaboratori e Algoritmi e strutture dati.

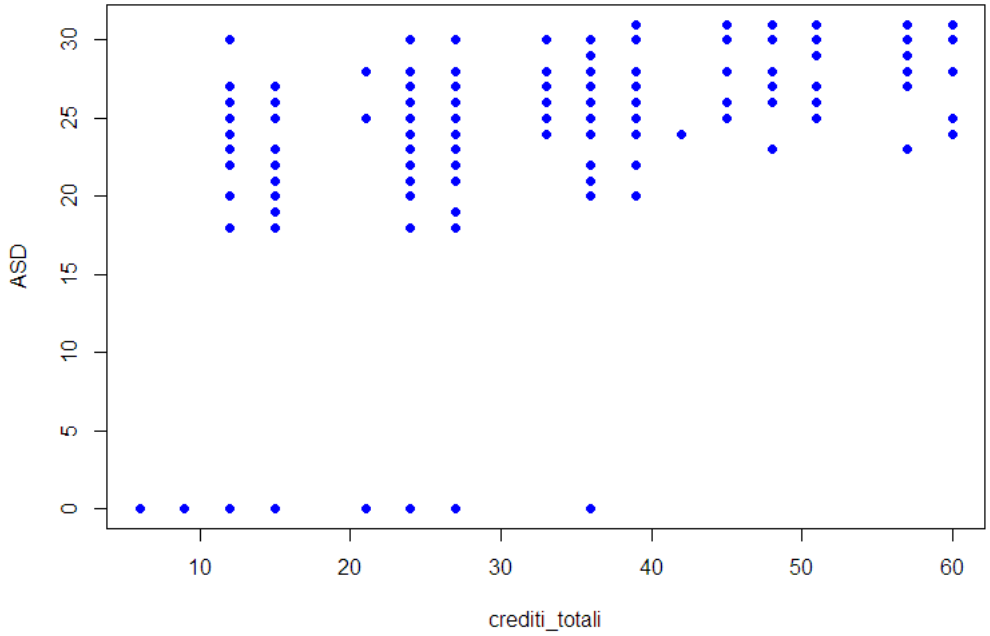

Figura 1: Scatterplot tra Crediti totali e Algoritmi e strutture dati

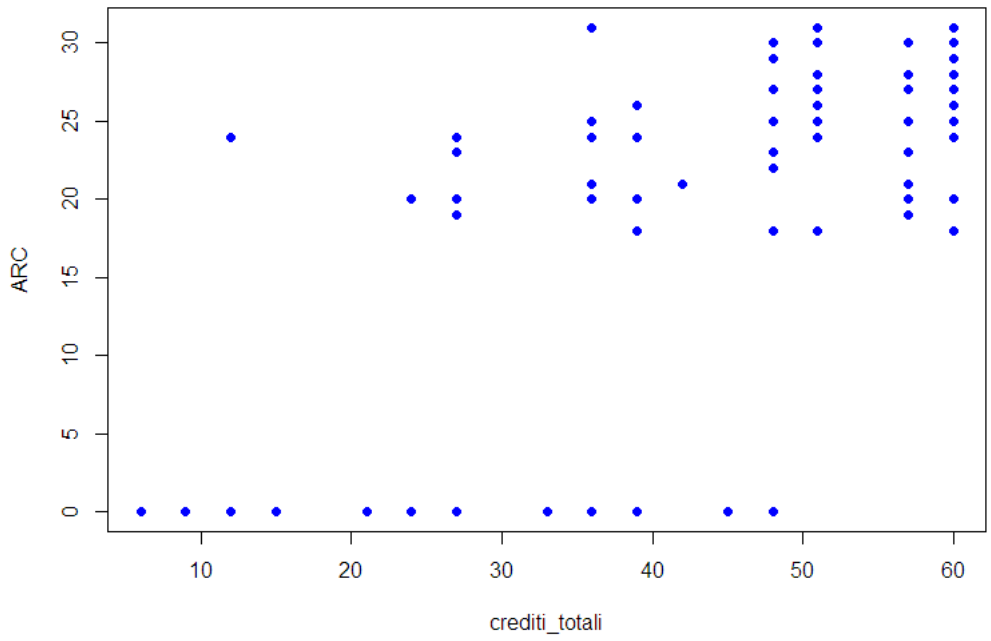

Figura 2: Scatterplot tra Crediti totali e Architetture degli elaboratori

L'attributo test è correlato maggiormente con l'attributo voto medio. Si può notare inoltre che il valore di correlazione è significativamente più alto di quello che l'attributo test ha con i voti dei singoli esami, con l'unica eccezione di MDL. Questi valori lasciano suggerire che il punteggio conseguito al test d'ingresso di ogni studente sia un buon parametro per valutarne l'andamento generale piuttosto che i voti di un singolo esame. Inoltre l'attributo test mostra una discreta correlazione con i crediti totali (e quindi anche con i crediti con voto). Per quanto riguarda le correlazioni tra i voti dei diversi esami è possibile notare che il valore più alto è quello tra Architetture degli elaboratori e Programmazione che è circa pari a 0.43. Mentre si ha una correlazione quasi nulla, ossia circa 0.10, tra il voto di Algoritmi e strutture dati e il voto di Analisi 1.

In Figura 3 sono riportati gli scatterplot tra Architetture degli elaboratori e Programmazione e tra Algoritmi e strutture dati e Analisi I. Contrariamente a quanto indicato dai valori delle correlazioni tra le due coppie

di attributi, gli scatterplot mostrati in Figura 3, sembrano suggerire una correlazione simile (se non migliore) tra Algoritmi e Strutture dati e Analisi 1, rispetto a quella tra Architetture e Programmazione. Questo fenomeno è dovuto al fatto che un notevole numero di studenti non ha sostenuto ne architteture ne programmazione. Tuttavia, in uno scatterplot non è possibile distinguere oggetti con gli stessi valori per gli attributi considerati e quindi i dati che influenzano maggiormente la correlazione tra Architetture e Programmazione non vengono essenzialmente mostrati (rappresentati tramite singolo punto) nel grafico in questione. D'altra parte, gli studenti che non hanno sostenuto ne Algoritmi ne Analisi 1 sono molti meno e quindi influenzano in mondo meno significativo la correlazione tra tali attributi. In Figura 4 sono mostrati gli scatterplot realizzati con Weka applicando piccole variazioni casuali ai punti per distinguere i dati che hanno valori identici (Jitter). Utilizzando questa tecnica è quindi possibile notare i seguenti fatti:

1. Il numero di studenti che non ha sostenuto ne Architteture ne Programmazione è, come abbiamo già detto, notevolmente superiore al numero di studenti che non hanno sostenuto ne algoritmi ne analisi 1. La presenza di studenti con queste caratteristiche influenza quindi maggiormente la correlazione complessiva tra Architetture e Programmazione rispetto a quella tra Algoritmi e Analisi1;

2. Gli studenti che hanno sostenuto Programmazione ma non hanno sostenuto Architetture sono molti di più rispetto a quelli che hanno sostenuto Analisi 1 ma non hanno sostenuto Algoritmi. Viceversa, gli studenti che hanno sostenuto Architetture ma non Programmazione sono molti meno di quelli che hanno sostenuto Algoritmi ma non Analisi 1. Gli studenti con queste caratteristiche (110 per Architetture e Programmazione, 127 per Algoritmi e Analisi1) non evidenziano correlazioni tra le due diverse coppie di attributi ed essendo in quantità paragonabili influenzeranno le correlazione complessive tra le due diverse coppie di attributi circa in egual misura;

3. Confrontando gli studenti che hanno sostenuto sia Architetture che Programmazione con quelli che hanno sostenuto sia Algoritmi che Analisi 1 si nota una migliore correlazione per la seconda coppia di attributi su tali sottoinsiemi del dataset. Tuttavia, essendo i due

sottoinsieme (studenti che hanno sostenuto sia Architetture che Programmazione e studenti che hanno sostenuto sia Algoritmi che Analisi 1) di dimensioni paragonabili influenzeranno in modo minore la correlazione complessiva rispetto a quanto lo fanno gli oggetti discussi al punto 1.

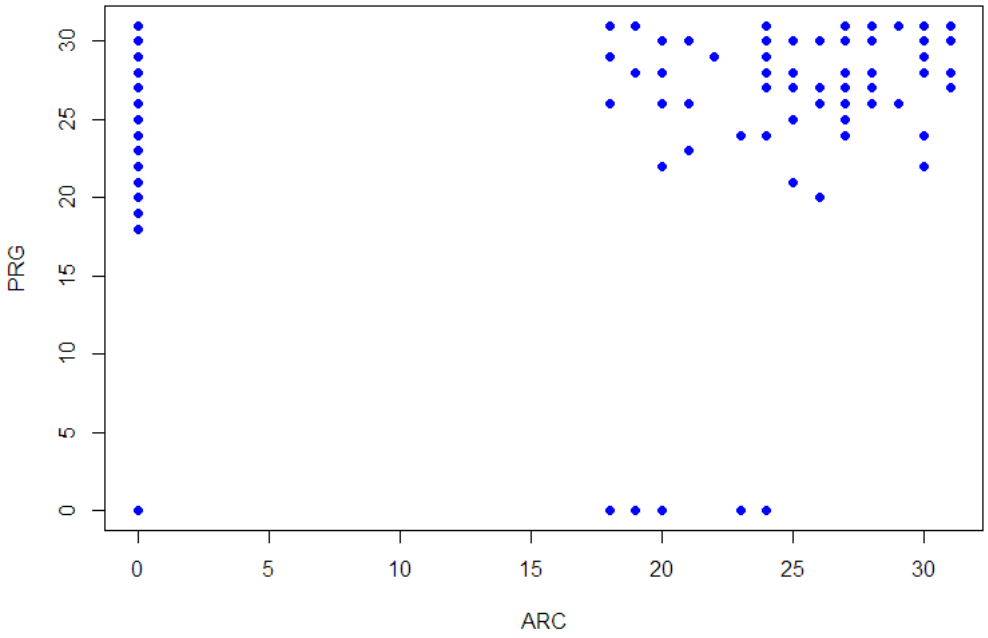

Figura 3: Scatterplot tra Architetture degli elaboratori e Programmazione

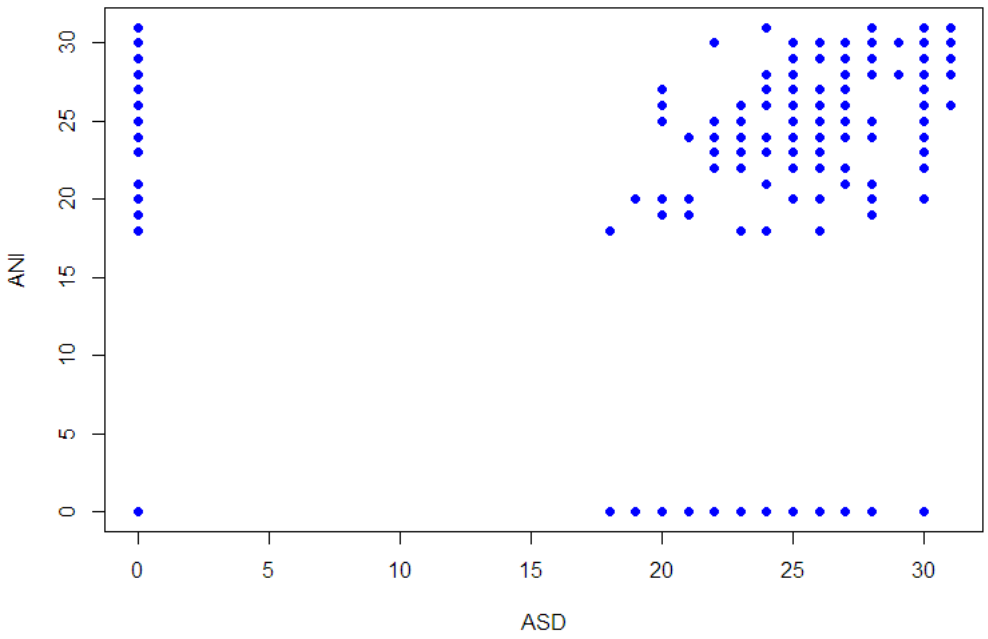

Figura 4: Scatterplot tra Algoritmi e strutture dati e Analisi I

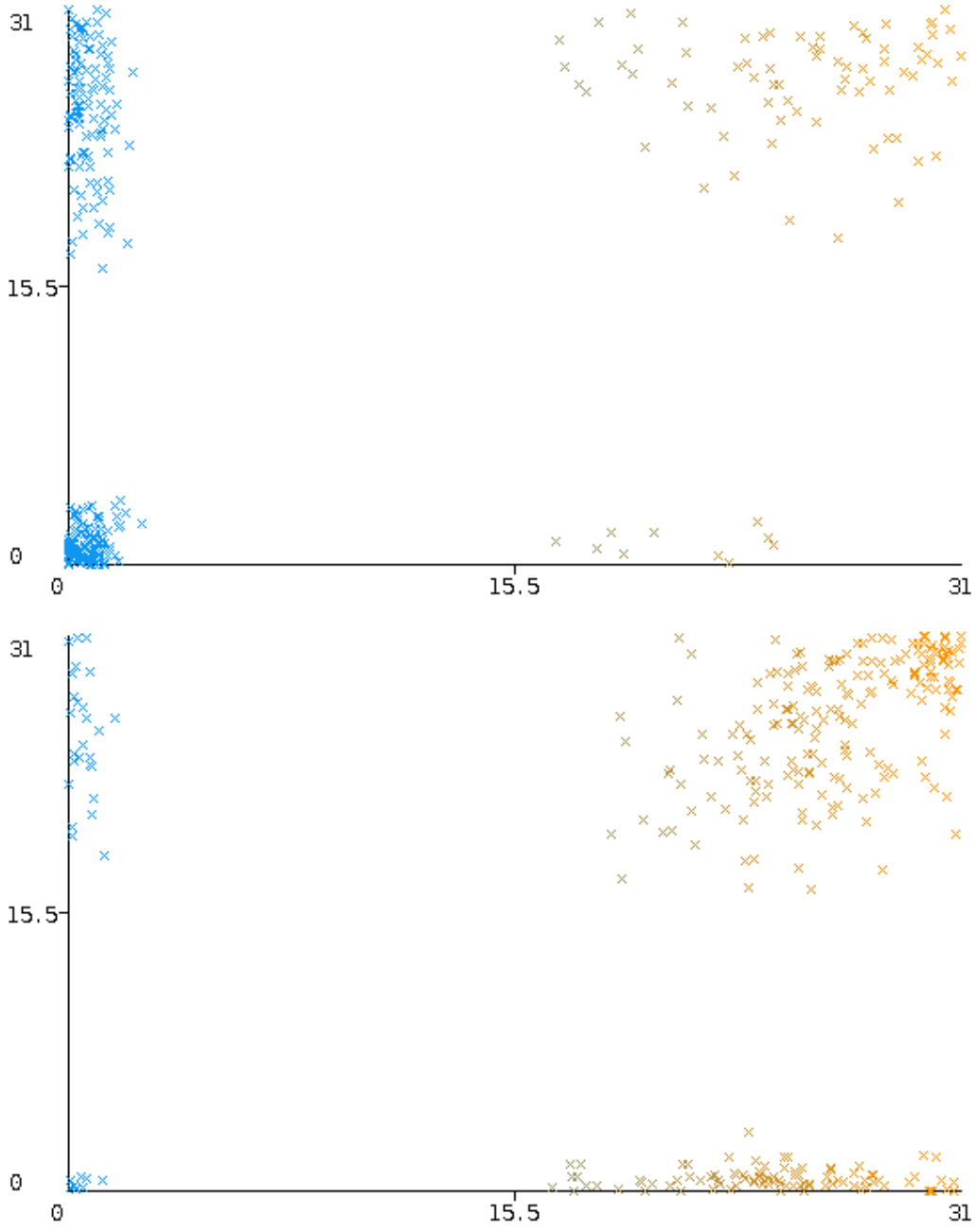

Figura 5: Scatterplot tra Architetture degli elaboratori e Programmazione e tra Algoritmi e strutture dati e Analisi I con Jitter

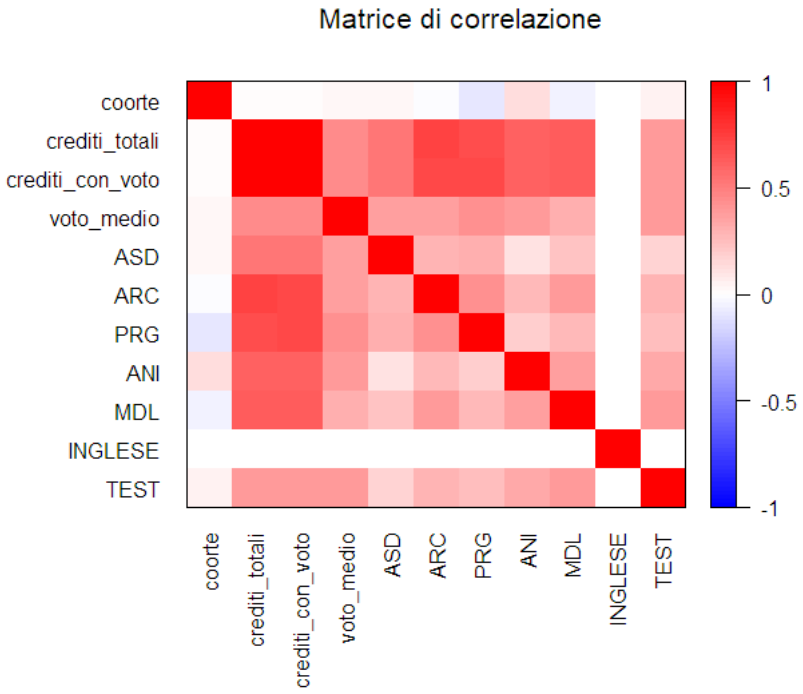

Figura 6: Matrice di correlazione

In Figura 6 viene riportata la matrice di correlazione. Per applicare gli algoritmi di clustering è stato utilizzato il software Weka. Tuttavia prima di applicare tali algoritmi è stato necessaria un ulteriore fase di preprocessing nella quale sono stati normalizzati tutti gli attributi del dataset in una scala di valori compresi tra zero e uno in modo da evitare problemi dovuti alle diverse scale di valori degli attributi.

4 Clustering

In questo capitolo verranno effettuate alcune analisi di clustering sui seguenti attributi:

- crediti totali, architetture, programmazione;
- algoritmi e strutture dati, architetture, programmazione, analisi 1 e matematica discreta e logica;
- voto medio e test.

Le tecniche di clustering utilizzate sono il clustering gerarchico, l'algoritmo di Kmeans e infine l'algoritmo DBSCAN. In particolare:

- l'analisi effettuata con tecniche di clustering gerarchico è stata effettuata su un sottoinsieme dei dati a disposizione selezionato in base alla coorte dello studente (anno 2010);
- nel caso dell'algoritmo di Kmeans viene stabilito preventivamente il numero dei cluster possibili utilizzando valori ritenuti sensati di volta in volta;
- l'algoritmo DBSCAN è stato utilizzato per l'analisi relativa ai voti dei diversi esami scegliendo preventivamente i valori di `MinPts` e `eps` ritenuti sensati di volta in volta.

Fatta eccezione per l'algoritmo DBSCAN tutte le analisi condotte sono state effettuate dopo la normalizzazione degli attributi coinvolti. Per la valutazione dei risultati ottenuti si rimanda al capitolo 5 dove viene analizzata la validità dei risultati ottenuti e sono determinati: numero di cluster ottimali per l'algoritmo di Kmeans e valore di `eps` ottimale per DBSCAN. Come prima analisi viene mostrata quella relativa ai tre attributi del dataset maggiormente correlati (a meno di correlazioni ovvie) ossia crediti totali, architetture e programmazione. È stato utilizzato l'algoritmo di Kmeans implementato in Weka specificando inizialmente un numero di cluster pari a due, lasciando i valori di default per la generazione dei centroidi. In Tabella 2 sono riportate le coordinate dei centroidi relativi all'esecuzione dell'algoritmo con un valore di k pari a 2. Come si può notare dalle coordinate dei centroidi questa prima esecuzione dell'algoritmo suddivide il dataset in due gruppi piuttosto distinti. Il valore della somma

	Crediti totali	ARC	PRG	Istanze
0	0.65	0.32	0.85	183 (58%)
1	0.27	0.05	0	133 (42%)

Tabella 2: Cluster con ARC e PRG con k = 2 SSE 51.35

degli errori al quadrato (SSE) in questa esecuzione è risultata pari a 51.35. In Tabella 3 sono riportate le coordinate dei centroidi relativi all'esecuzione dell'algoritmo con un valore di k pari a 3. In questo caso il valore di SSE

	Crediti totali	ARC	PRG	Istanze
0	0.88	0.82	0.89	73 (23%)
1	0.27	0.05	0	133 (42%)
2	0.50	0	0.81	110 (35%)

Tabella 3: Cluster con ARC e PRG con k = 3 SSE 14.85

è pari a 14.85. È inoltre possibile constatare come l'algoritmo di K-means abbia messo in evidenza tre tipologie ben distinte di studenti:

- Gli studenti appartenenti al cluster 0 sono gli studenti "migliori" avendo sostenuto la quasi totalità degli esami del primo anno alla fine della sessione estiva e riportando delle ottime valutazioni per quanto riguarda gli esami di Architetture e di Programmazione;
- La seconda categoria di studenti (cluster 1) sono gli studenti "peggiori" che hanno sostenuto pochi esami e nel caso specifico delle materie considerate hanno conseguito valutazioni basse o non hanno sostenuto l'esame;
- Infine gli studenti appartenenti all'ultimo cluster sono gli studenti che hanno sostenuto Programmazione con un buon voto ma non hanno fatto l'esame di Architetture.

Come si evince analizzando i tre cluster in particolare notando le diverse combinazioni dei valori assunti dai centroidi dei voti, si capisce come sia assente la categoria di studenti che ha sostenuto con profitto l'esame di architetture, ma non ha sostenuto l'esame di programmazione, lasciando

quindi intendere che se uno studente ha sostenuto l'esame di architetture allora generalmente ha sostenuto con una buona valutazione l'esame di programmazione.

In Figura 7 è mostrato il dendogramma relativo al clustering gerarchico con metodo complete mentre in Figura 8 viene mostrato il dendogramma relativo al clustering gerarchico con metodo average per questa prima analisi.

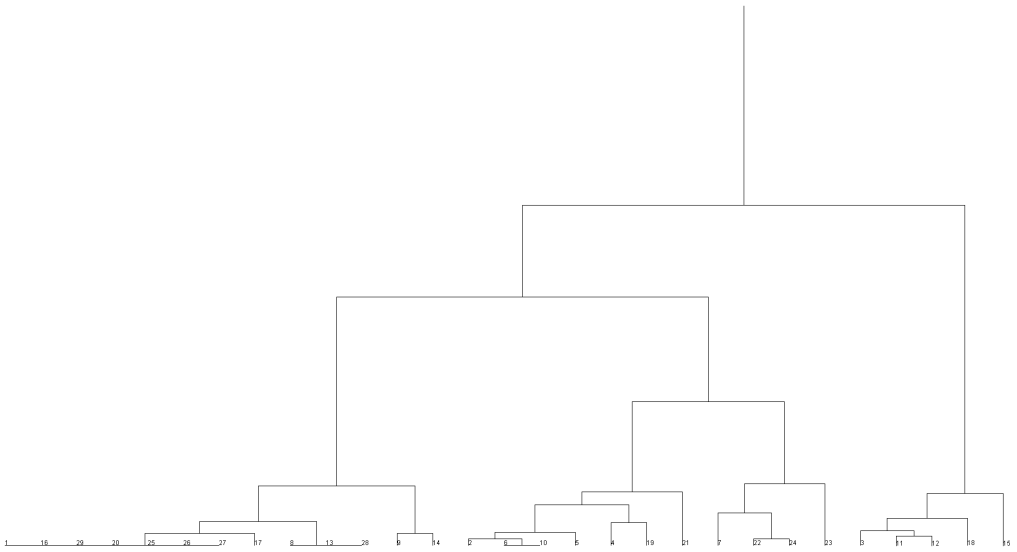

Figura 7: Dendogramma relativo al clustering gerarchico con metodo complete.

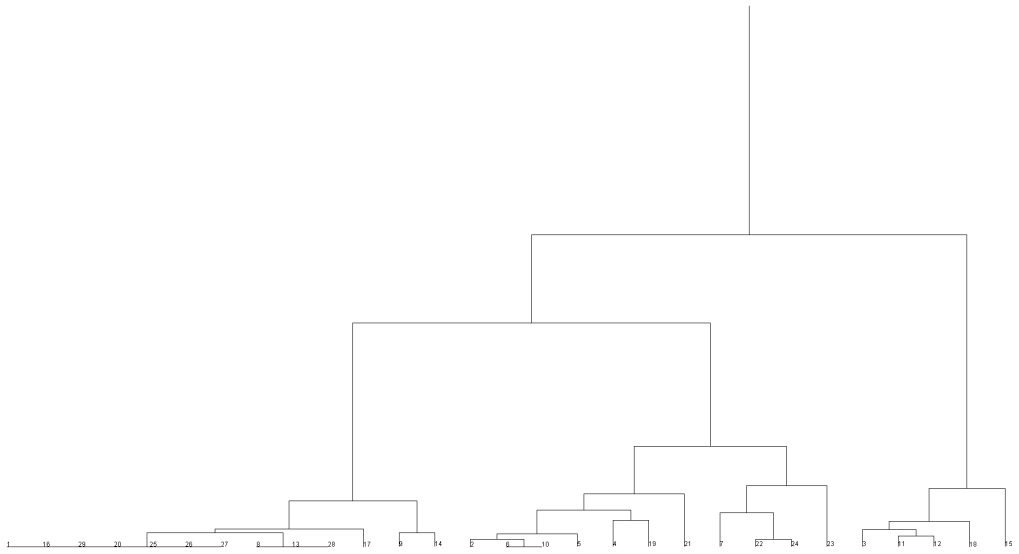

Figura 8: Dendogramma relativo al clustering gerarchico con metodo average.

La seconda analisi che è stata condotta è quella relativa ai voti conseguiti dagli studenti del primo anno durante la sessione estiva. In questo caso l'algoritmo di K-means è stato inizializzato con un valore di $k = 3$. In Tabella 4 sono riportate le coordinate dei centroidi al termine dell'algoritmo. In questo caso sono quindi stati determinati i profili di tre diversi

	ASD	ARC	PRG	ANI	MDL	Istanze
0	0.73	0.05	0.81	0.43	0.02	100 (32%)
1	0.65	0.05	0	0.50	0.12	133 (42%)
2	0.91	0.65	0.89	0.88	0.60	83 (26%)

Tabella 4: Cluster di tutti i voti con k = 3 SSE 106.19

gruppi di studenti:

- gli studenti che hanno conseguito una buona votazione negli esami di Algoritmi e Strutture Dati e Programmazione, una votazione discreta all'esame di Analisi I e che non hanno sostenuto Matematica discreta e Logica e Architetture degli elaboratori (Cluster 0);

- gli studenti con le stesse caratteristiche del cluster precedente, ma che non hanno sostenuto Programmazione (Cluster 1);
- gli studenti che hanno sostenuto tutti gli esami e con un buona votazione (Cluster 2).

In Figura 9 è riportato lo scatter plot relativo ai voti di Architetture degli Elaboratori e Programmazione che sono maggiormente correlati. Il valore di SSE in questo caso è pari a 106.19. Come è stato specificato all'inizio del capitolo, in questo caso è stato utilizzato anche l'algoritmo DBSCAN per effettuare il clustering degli attributi scelti. Inoltre, poiché gli attributi scelti per l'analisi in questo caso hanno valori nello stesso intervallo $[0, 31]$ non è stato necessario procedere alla normalizzazione. Quindi, analogamente a quanto fatto con l'algoritmo di K-means, i parametri con cui eseguire l'algoritmo sono stati scelti basandosi sull'intuizione e sulla conoscenza del problema. Per la valutazione dei risultati ottenuti e la scelta del valore ottimale di `eps` in funzione di `MinPts` si rimanda al capitolo 5.

In Tabella 5 sono riportati i cluster ottenuti e la proporzione di istanze al loro interno eseguendo l'algoritmo DBSCAN con `MinPts=6` e `eps=0.5`. In questo caso 38 record dei 316 totali sono stati marcati come rumore dall'algoritmo. Inoltre, è possibile notare che alcuni dei cluster prodotti hanno delle dimensioni decisamente ridotte. Infatti, il cluster 6 contiene solo 6 record dei 316 totali, mentre il cluster 3, il cluster 7 e il cluster 9 contengono solo, rispettivamente 13, 18 e 14 record. Il cluster 8 sembra rappresentare un caso limite con 22 record complessivi. In ogni caso, un numero così elevato di cluster di dimensioni ridotte sembra suggerire che il valore di `MinPts` sia troppo basso e che i gruppi di oggetti riconosciuti come cluster che hanno dimensione ridotta siano in realtà gruppi di outliers. In Tabella 6 sono riportati i medesimi risultati eseguendo l'algoritmo DBSCAN con `MinPts=10` e `eps=0.4`. In questo caso, i risultati ottenuti sembrano confermare le nostre aspettative poiché i record marcati come rumore dall' algoritmo diventano 44. Inoltre, il numero di cluster determinati dall'algoritmo è diminuito di uno e i cluster di dimensioni significativamente ridotte sono solamente il cluster 3 e il cluster 8. L'esecuzione dell'algoritmo con un numero maggiore di `MinPts` sembrerebbe quindi produrre un clustering migliore. Andando a incrementare ulteriormente il valore di `MinPts` ponendo `MinPts` a 20 si ottiene il clustering mostrato in Tabella 7. In questo caso, il numero di cluster scende ulteriormente di tre e inoltre la distribuzione dei record nei cluster risulta decisamente più uniforme e non sono presenti

dei cluster di dimensioni significativamente piccole. Inoltre, in questo caso 89 dei 316 record totali sono stati marcati come rumore dall'algoritmo.

In Figura 10 è mostrato il dendogramma relativo al clustering gerarchico con metodo complete mentre in Figura 11 viene mostrato il dendogramma relativo al clustering gerarchico con metodo average per la seconda analisi.

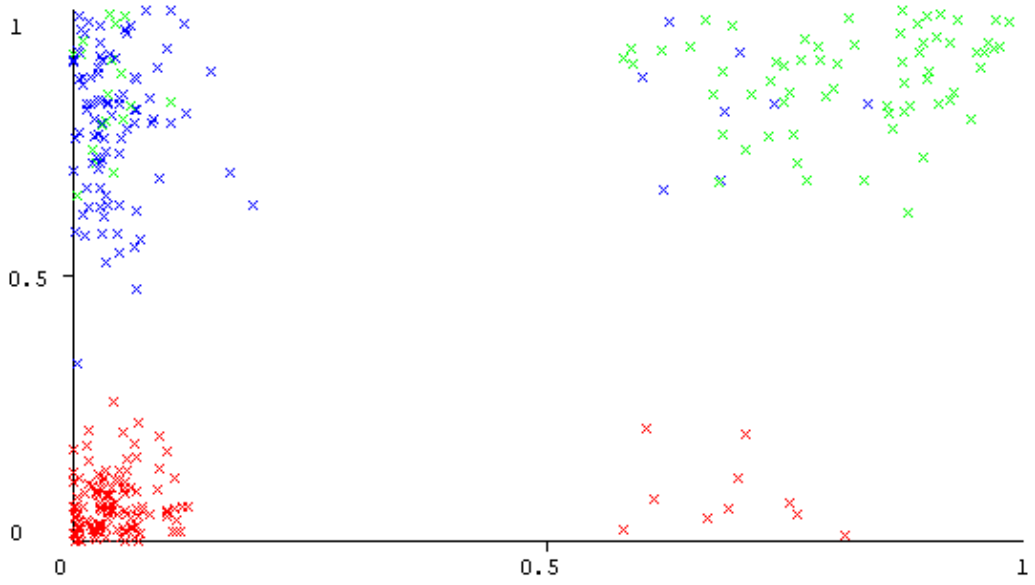

Figura 9: Scatter plot relativo ai cluster dei voti di Architetture degli Elaboratori e Programmazione

	Istanze
0	40 (14%)
1	46 (17%)
2	41 (15%)
3	13 (5%)
4	45 (16%)
5	33 (12%)
6	6 (2%)
7	18 (6%)
8	22 (8%)
9	14 (5%)

Tabella 5: Cluster ottenuti con DBSCAN eseguito con `MinPts=6` e `eps=0.5`. Molti dei cluster prodotti hanno dimensioni notevolmente ridotte.

	Istanze
0	40 (15%)
1	46 (17%)
2	41 (15%)
3	13 (5%)
4	45 (17%)
5	33 (12%)
6	18 (7%)
7	22 (8%)
8	14 (5%)

Tabella 6: Cluster ottenuti con DBSCAN eseguito con `MinPts=10` e `eps=0.4`. In questo caso di ottengono meno cluster e si riduce il numero di cluster di dimensioni ridotte.

	Istanze
0	40 (18%)
1	46 (20%)
2	41 (18%)
3	45 (20%)
4	33 (15%)
5	22 (10%)

Tabella 7: Cluster ottenuti con DBSCAN eseguito con `MinPts=20` e `eps=0.4`. In questo caso la distribuzione dei record nei cluster risulta decisamente più uniforme.

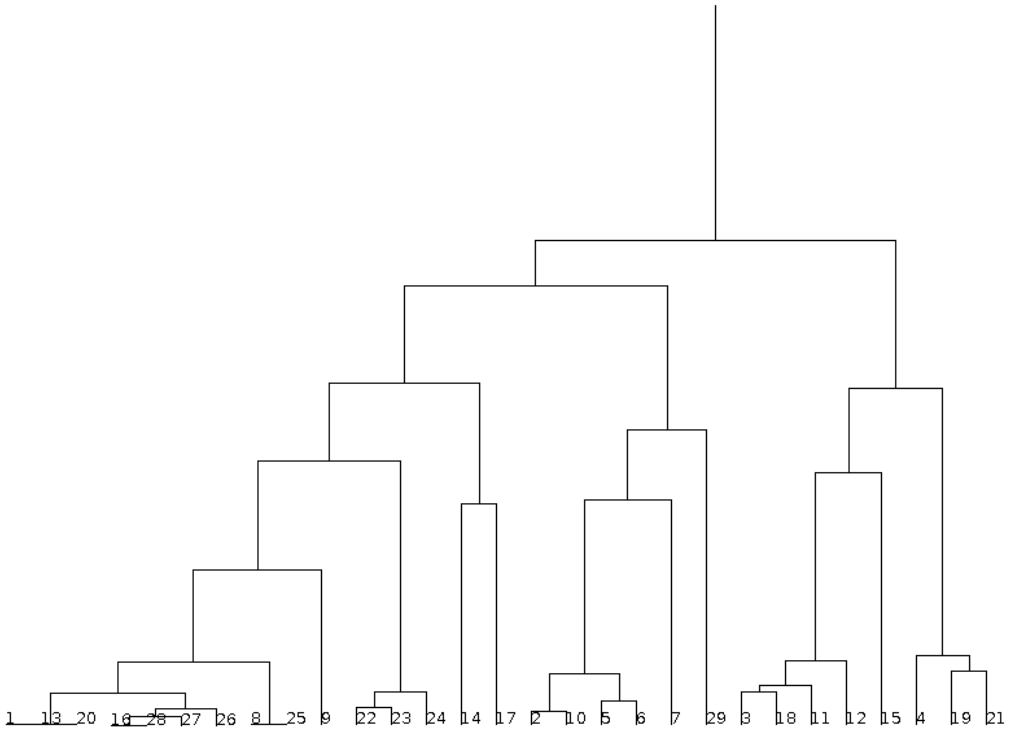

Figura 10: Dendogramma relativo al clustering gerarchico con metodo complete.

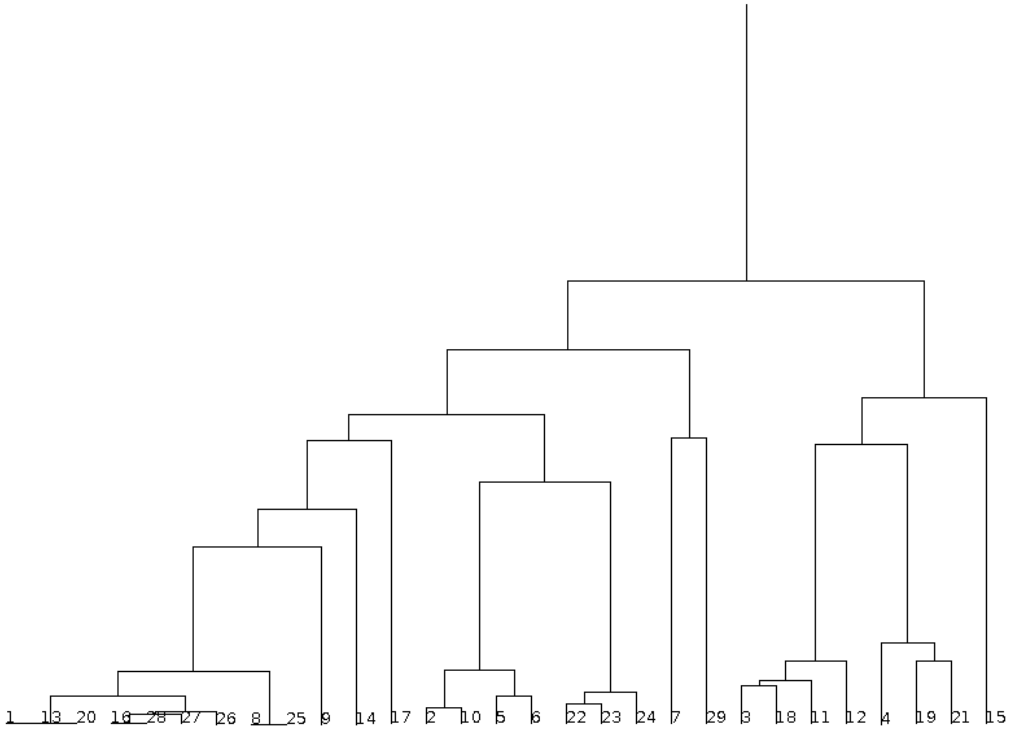

Figura 11: Dendogramma relativo al clustering gerarchico con metodo average.

Infine, l'ultima analisi che è stata condotta riguarda gli attributi test e voto medio. È stato scelto di analizzare questi due attributi congiuntamente in quanto, come è stato detto nel capitolo precedente, l'attributo voto medio presenta una buona correlazione con l'attributo test. In Tabella 8 sono riportate le coordinate dei centroidi relativi all'esecuzione dell'algoritmo con un numero di cluster pari a 3. In questo caso è possibile notare

	voto medio	Test	Istanze
0	0.36	0.41	85 (27%)
1	0.75	0.66	146 (42%)
2	0.45	0.67	85 (27%)

Tabella 8: Cluster con Voto_medio e Test con k = 3 SSE 9.6

come l'algoritmo di K-means determini tre cluster ben definiti che suddi-

vidono il dataset tra gli studenti che hanno una media complessiva maggiore e un voto al test d'ingresso alto e quelli che invece hanno una media più bassa e hanno conseguito punteggio basso al test di ingresso. Questi gruppi determinati sono coerenti con la correlazione che esiste tra i due attributi che tuttavia non è particolarmente elevata (diversamente dagli attributi presi in considerazione nell'analisi precedente). Infatti, oltre ai primi due cluster che identificano gli studenti "migliori" e quelli "peggiori" esiste un terzo cluster di studenti che hanno conseguito un punteggio al test d'ingresso decisamente positivo, ma non hanno mantenuto una media dei voti altrettanto buona. Il valore del SSE in questo caso è 9.6.

In Figura 12 è mostrato il dendogramma relativo al clustering gerarchico con metodo complete mentre in Figura 13 viene mostrato il dendogramma relativo al clustering gerarchico con metodo average per la terza e ultima analisi condotta.

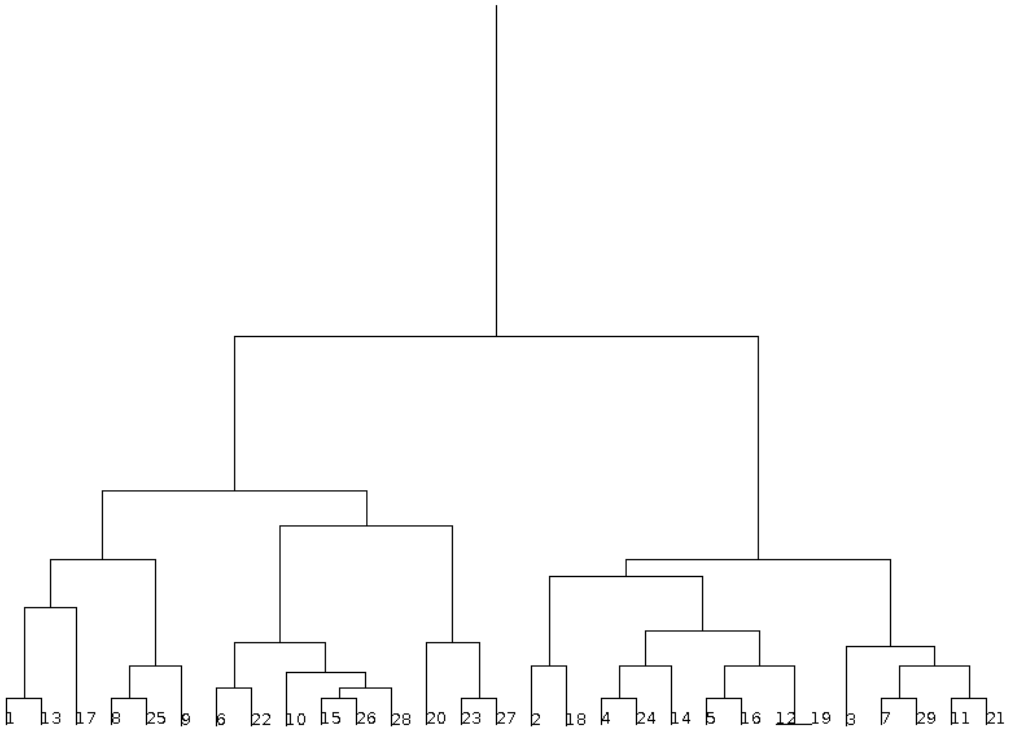

Figura 12: Dendogramma relativo al clustering gerarchico con metodo complete.

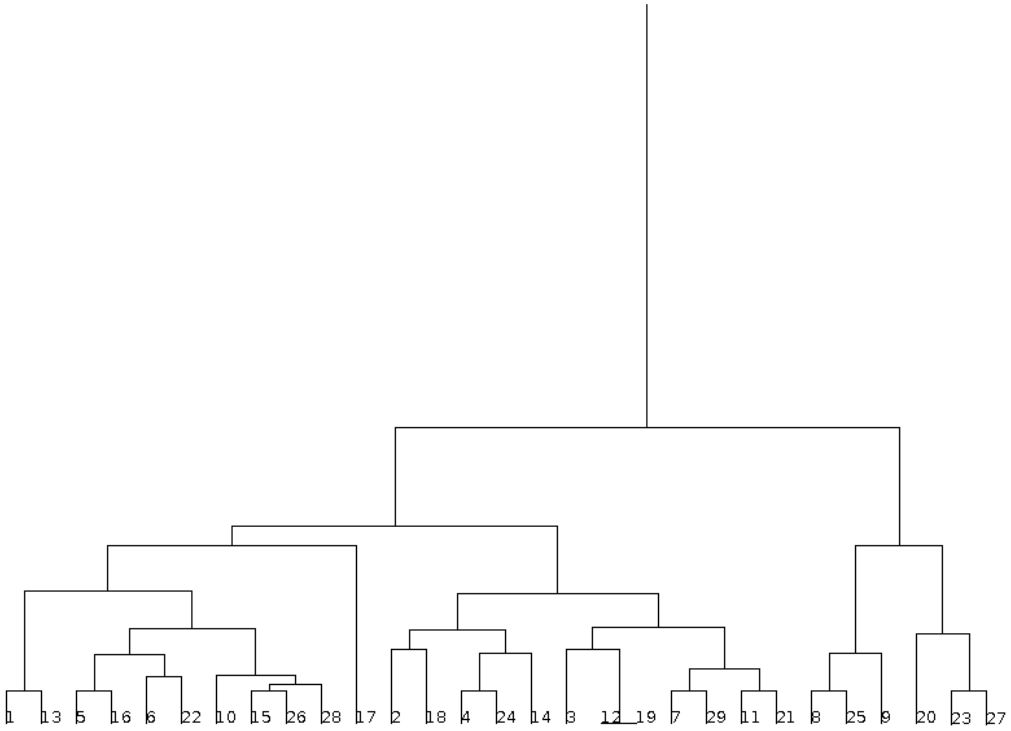

Figura 13: Dendogramma relativo al clustering gerarchico con metodo average.

5 Valutazione del clustering e model selection

In questo capitolo vengono presi in considerazione alcuni metodi per scegliere i parametri con cui inizializzare gli algoritmi di K-means e DBSCAN e vengono analizzati i risultati ottenuti nel capitolo 4. Nel capitolo precedente, infatti, l'algoritmo di K-means è stato eseguito scegliendo preventivamente il numero di cluster possibili (tipicamente 2 o 3) basandosi esclusivamente sull'intuizione e quindi senza avere garanzie circa la bontà e correttezza dei risultati ottenuti. In questo capitolo viene valutata la validità delle analisi di clustering effettuate con l'algoritmo K-means e viene utilizzata una procedura basata sul SSE che tenta di determinare il valore ottimale di k con cui inizializzare l'algoritmo di K-means in modo da migliorarne la validità.

Per ciascuno degli aspetti analizzati vengono quindi eseguite le seguenti operazioni:

1. determinazione dei valori del SSE in funzione di k;
2. scelta di k_{opt} come il più piccolo k per cui il valore del SSE "smette di decrescere";
3. confronto del valore di correlazione ottenuta tra la matrice di incidenza e quella delle distanze per i valori di $k = 2$, $k = 3$ e k_{opt}.

Per quanto riguarda il Punto 2 è necessario approssimare il valore di k_{opt} scegliendo, ad esempio, il primo valore di k per cui si verifica una variazione nel SSE minore di una quantità fissata ε. Per quanto concerne il Punto 3 si ha che un valore inferiore della correlazione (sperabilmente negativo) indica un miglior risultato di clustering poiché, idealmente, punti appartenenti allo stesso cluster (quindi con valore di incidenza 1) dovrebbero trovarsi a una distanza minore. Quindi, per i tre attributi crediti totali, architetture e programmazione su cui è stata fatta la prima analisi con l'algoritmo di K-means si ottiene il grafico $k - SSE$ riportato in Figura 14.

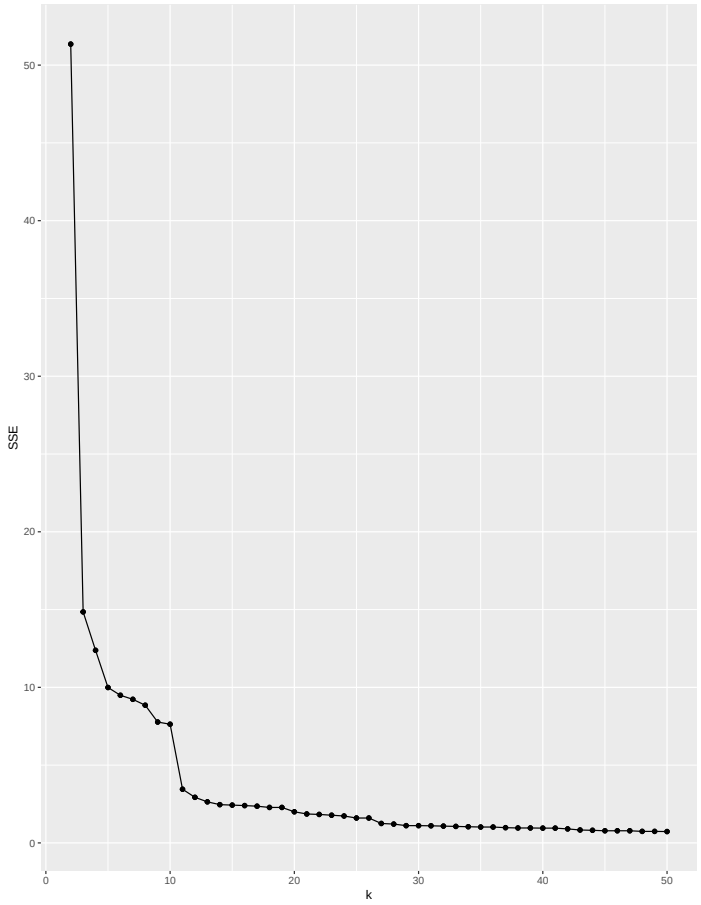

Figura 14: Andamento del valore del SSE in funzione del valore di k.

Scegliendo il valore minimo di $\varepsilon = 0.01$ (dato che è stato scelto di memorizzare i valori di SSE fino alla seconda cifra decimale) si ottiene un valore di $k_{opt} = 18$. Quindi, per determinare il valore di correlazione tra matrice di incidenza e matrice delle distanze è necessario esportare preventivamente da Weka il dataset munito di un attributo aggiuntivo che indichi il cluster di appartenenza di ciascun record. In Figura 15 viene mostrato come creare ed esportare un dataset con Weka aggiungendo per ogni record il riferimento al cluster di appartenenza a seguito dell'esecuzione dell'algoritmo di Kmeans per gli attributi crediti totali, architetture e programmazione con $k = 2$.

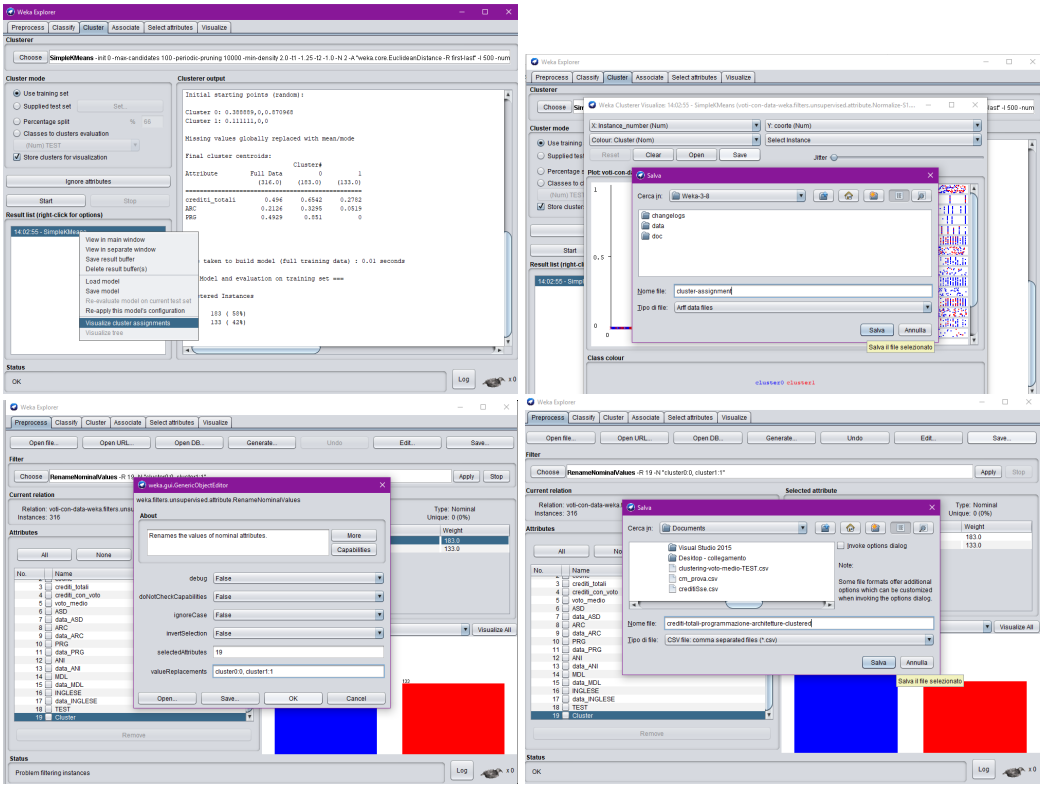

Figura 15: creazione ed esportazione del dataset con indicazione del cluster di appartenenza in Weka.

Nel Codice 3 viene mostrato come importare il dataset comprensivo degli attributi crediti totali, architetture, programmazione e cluster con R, mentre il Codice 4 calcola la Matrice di incidenza dei cluster, la Matrice delle distanze per i punti e, infine, la correlazione tra le due matrici. Come si evince dal codice, per poter calcolare il valore della correlazione è stato necessario linearizzare preventivamente le due matrici utilizzando l'istruzione `as.vector` di R.

```
library(readr)
crediti_totali_prg_arc_clustered <-
    read_csv("dmo/crediti_totali-prg-arc-clustered.csv",
    col_types = cols(ANI = col_skip(), ASD = col_skip(),
        INGLESE = col_skip(), Instance_number = col_skip(),
        MDL = col_skip(), TEST = col_skip(),
        coorte = col_skip(), crediti_con_voto = col_skip(),
        data_ANI = col_skip(), data_ARC = col_skip(),
        data_ASD = col_skip(), data_INGLESE = col_skip(),
        data_MDL = col_skip(), data_PRG = col_skip(),
        voto_medio = col_skip()))
View(crediti_totali_prg_arc_clustered)
```

Codice 3: Importazione degli attributi crediti totali, architetture, programmazione e cluster.

```
# 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)
}
```

```
# 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)
```

Codice 4: Calcolo Matrice di incidenza dei cluster, delle distanze e correlazione tra le due matrici.

```
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;
import weka.filters.Filter;
import weka.filters.unsupervised.attribute.Normalize;
import weka.clusterers.SimpleKMeans;
import weka.filters.unsupervised.attribute.Remove;

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class SseWeka {

   public static void main(String[] args) throws Exception {
      DataSource source = new DataSource("./voti-con-data.arff");
      Instances data = source.getDataSet();
      data.setClassIndex(-1);

      Normalize normalize = new Normalize();
      normalize.setInputFormat(data);
```

```
21      Instances nData = Filter.useFilter(data, normalize);

23      Remove remove = new Remove();
        // 0 coorte,1 crediti_totali,2 crediti_con_voto,3 voto_medio,4
            ASD,5 data_ASD,6 ARC,7 data_ARC,
25      // 8 PRG,9 data_PRG,10 ANI,11 data_ANI,12 MDL,13 data_MDL,14
            INGLESE,15 data_INGLESE,16 TEST
        int[] attributeIndexesToRemove = {1,6,8};
27      remove.setInvertSelection(true);
        remove.setAttributeIndicesArray(attributeIndexesToRemove);
29      remove.setInputFormat(nData);
        Instances creditiArcPrg = Filter.useFilter(nData, remove);
31
        SimpleKMeans kMeans = new SimpleKMeans();
33      kMeans.setPreserveInstancesOrder(true);

35      int maxK = 50;
        int minK = 2;
37      ArrayList<double[]> resultSet = new ArrayList<double[]>(maxK);
        double[] a;
39      for (int i = minK; i < maxK; i++) {
           kMeans.setNumClusters(i);
41         kMeans.buildClusterer(creditiArcPrg);
           a = new double[2];
43         a[0] = i; a[1] = kMeans.getSquaredError();
           resultSet.add(a);
45      }
        for (int i = 0; i < maxK - minK; i++) {
47         System.out.println(resultSet.get(i)[1]);
        }
49
        try {
51         FileWriter writer = new FileWriter("creditiArcSse.txt",
               false);

53         for (int i = minK; i < maxK; i++)
              writer.write(i + "," + resultSet.get(i - minK)[1] +
                  "\r\n");
55         writer.close();
        } catch (IOException e) {
```

```
57          e.printStackTrace();
        }
59
    }
61 }
```

Codice 5: Codice Java per il calcolo di SSE al variare di K

Il valore di correlazione ottenuto in questo caso è pari a -0.687. Ripetendo il clustering per gli stessi attributi con $k = 3$ e $k = 18$ si ottiene, rispettivamente un valore della correlazione tra le due matrici di -0.854 e -0.489.

Diversamente da quanto atteso, il valore calcolato per k_{opt} non presenta un valore di correlazione inferiore rispetto agli altri due clustering, bensì risulta essere il valore peggiore tra quelli verificati con il metodo della correlazione. In Figura 16 e Figura 17 sono riportati i grafici dell'andamento del SSE in funzione di k per, rispettivamente, l'analisi condotta sui voti conseguiti dagli studenti nelle cinque materie del primo anno e l'analisi relativa al voto medio e al risultato del test. Scegliendo quindi nuovamente $\varepsilon = 0.01$ i valori di k_{opt} sono, rispettivamente 36 e 43.

Attributi analizzati	**k**	**Correlazione**
crediti totali, ARC, PRG	2	-0.687
	3	-0.854
	18	-0.489
ARC, ASD, PRG, MDL, AN1	2	-0.520
	3	-0.618
	36	-0.424
voto medio, test	2	-0.476
	3	-0.465
	43	-0.273

Tabella 9: Valori correlazione tra la Matrice di incidenza dei cluster e la matrice delle distanze in funzione di k.

In Tabella 9 sono riportate sinteticamente le correlazioni ottenute per ciascuna delle tre analisi condotte in funzione dei diversi valori di k scelti. Come è possibile notare consultando la Tabella 1, anche nella seconda e terza analisi il valore di k scelto consultando il grafico $k - SSE$ risulta essere il peggiore contrariamente alle aspettative.

È stata ripetuta la procedura con valori maggiori di ε ($\varepsilon = 0.01$ e $\varepsilon = 1$), tuttavia le correlazioni ottenute sono risultate essere, anche in questo caso, decisamente peggiori rispetto alle correlazioni ottenute con i valori di k scelti inizialmente.

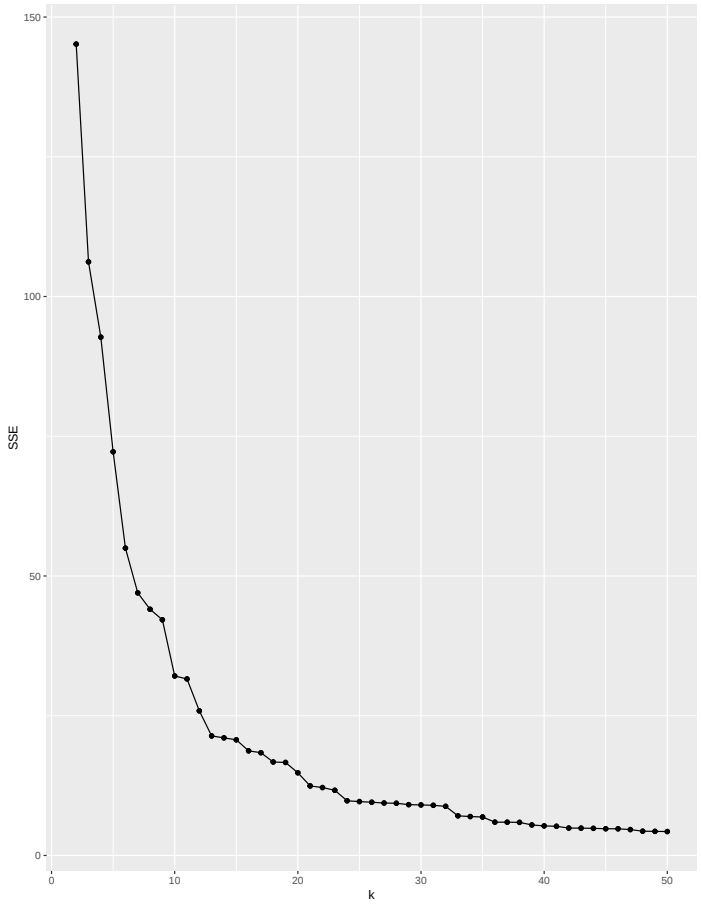

Figura 16: Andamento del valore del SSE in funzione del valore di k per i voti di architetture, programmazione, algoritmi e strutture dati, analisi matematica 1 e matematica discreta e logica.

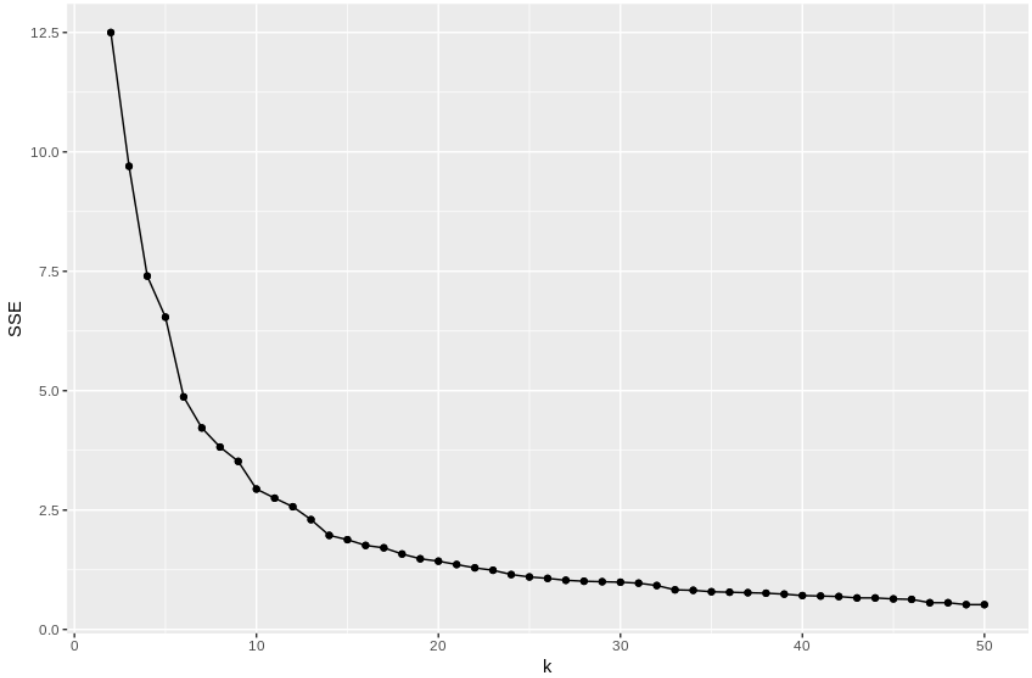

Figura 17: Andamento del valore del SSE in funzione del valore di k per il voto medio e il voto del test.

Una possibile spiegazione di questo fenomeno è la seguente: aumentando notevolmente il numero di cluster k è probabile che record vicini vengano posti in cluster differenti soprattutto se i centroidi di quest'ultimi sono anch'essi molto vicini, ossia se i cluster non sono ben separati e risultano essere particolarmente irregolare o sovrapposti. Quando si verifica questa circostanza il valore di incidenza nella matrice dei cluster è 0 anche se la corrispettiva distanza tra due record è molto piccola andando così a ridurre il modulo della correlazione (negativa) che esiste tra i valori delle due matrici. È possibile calcolare il valore di correlazione (e quindi valutare la bontà del clustering) tra matrice di incidenza dei cluster e matrice delle distanze anche nel clustering calcolato dall'algoritmo DBSCAN. Tale metodo è stato utilizzato nell'analisi relativa ai voti conseguiti dagli studenti nelle cinque materie del primo anno. Tuttavia, in questa circostanza è necessario provvedere all'eliminazione dal dataset dei record marcati come rumore da DBSCAN e quindi sarà necessario tener conto non solo del valore di correlazione ottenuto, ma considerare anche quanti dati vengono

esclusi eseguendo l'algoritmo con certi parametri.

La Figura 18 mostra con che valori inizializzare i parametri del filtro di Weka `RemoveWithValues` al fine di eliminare dal dataset i record marcati come rumore. La Tabella 10 riporta sinteticamente i valori di correlazione e numero di record etichettati come rumore ottenuti per ogni esecuzione dell'algoritmo DBSCAN con i diversi parametri utilizzati. Come si evince dalla Tabella 10 l'esecuzione ottimale del DBSCAN è ottenuta con `MinPts = 20` e `eps=0.4`. Tale risultato ha tuttavia un prezzo: i record etichettati in questo caso sono 89 ossia più del doppio rispetto alle altre due esecuzioni prese in considerazione. Inoltre, considerando la Tabella 1 è possibile concludere che, l'algoritmo DBSCAN determina dei clustering migliori rispetto a K-means per l'analisi relativa ai voti conseguiti dagli studenti.

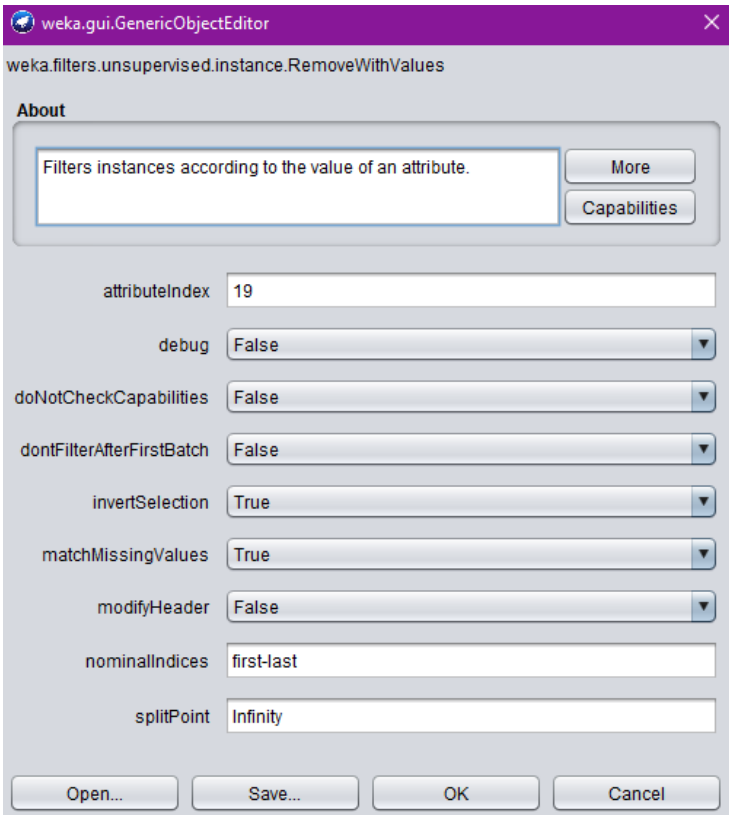

Figura 18: Valori da impostare nei parametri del filtro `RemoveWithValues` di Weka.

MinPts, eps	Correlazione	Rumore
6,0.5	-0.725	38
10,0.4	-0.728	44
20,0.4	-0.758	89

Tabella 10: Valori riepilogativi di correlazione e rumore per DBSCAN

Fino a questo momento è stata analizzata la bontà del clustering fornito da DBSCAN con i parametri scelti in maniera arbitraria basandosi sull'intuito e la conoscenza del problema. In maniera del tutto analoga a quanto fatto per l'algoritmo di K-means è possibile utilizzare una procedura di model selection che cerchi di stabilire il modello (e quindi i parametri) ottimale. Nel caso di DBSCAN la procedura che verrà utilizzata cerca di determinare il valore ottimale di `eps` fissando il valore di `MinPts` basandosi sull'idea che i k-esimi vicini dei punti contenuti in un cluster sono più o meno posti alla stessa distanza mentre i punti etichettati come rumore hanno il k-esimo vicino a una distanza maggiore. Basandosi su questo fatto è possibile determinare un grafico che sulle ascisse contiene gli indici dei record ordinati in modo non decrescente rispetto alla distanza dal loro k-esimo record più vicino e sulle ordinate mostra il valore di tale distanza. Come valore di `eps` è quindi possibile scegliere la prima distanza che è "sufficientemente" maggiore dalle precedenti.

Nel Codice 6 è mostrata la funzione in R che determina il grafico in questione e la riga di codice per calcolare tale grafico nel caso in cui `MinPts=6`. In Figura 19 è riportato il grafico dei valori delle distanze dei punti dal k-6-esimo più vicino. Come si può notare dalla figura, il valore ottimale indicato per `eps` è maggiore di 0.4 e minore di 0.6 e quindi il valore di `eps` risulta essere una buona scelta. La Figura 20 e la Figura 21 riportano, rispettivamente, le distanze dei punti dal loro k-10-esimo più vicino e dal loro k-20-esimo più vicino. Nel primo caso, il valore `eps` scelto inizialmente è confermato da quanto riportato dalla Figura 20, mentre la Figura 21 suggerisce di scegliere un valore di `eps` maggiore di 0.5 per l'esecuzione del DBSCAN con `MinPts=20`. Tuttavia, l'esecuzione di DBSCAN con la combinazione di tali parametri non ha prodotto risultati significativi.

```r
kDBScan <- function(data,k){
  library(ggplot2)
  D = as.matrix(dist(data[,1:ncol(data)-1],method = 'euclidean',diag
      = TRUE,upper = TRUE))
  D_1 = D
  for(i in 1:nrow(data)){
    D_1[i,] = sort(D[i,])
  }
  p = 1:nrow(data)
  dist = sort(D_1[, k])
  data = data.frame(p,dist)
  ggplot(data, aes(x=p, y=dist)) +geom_point(shape=1) + geom_line()
      + geom_point(color = 'black')
}

kDBScan(crediti_totali_prg_arc_clustered, 6)
```

Codice 6: Codice R per il calcolo del grafico della k-esima distanza da ogni punto del dataset.

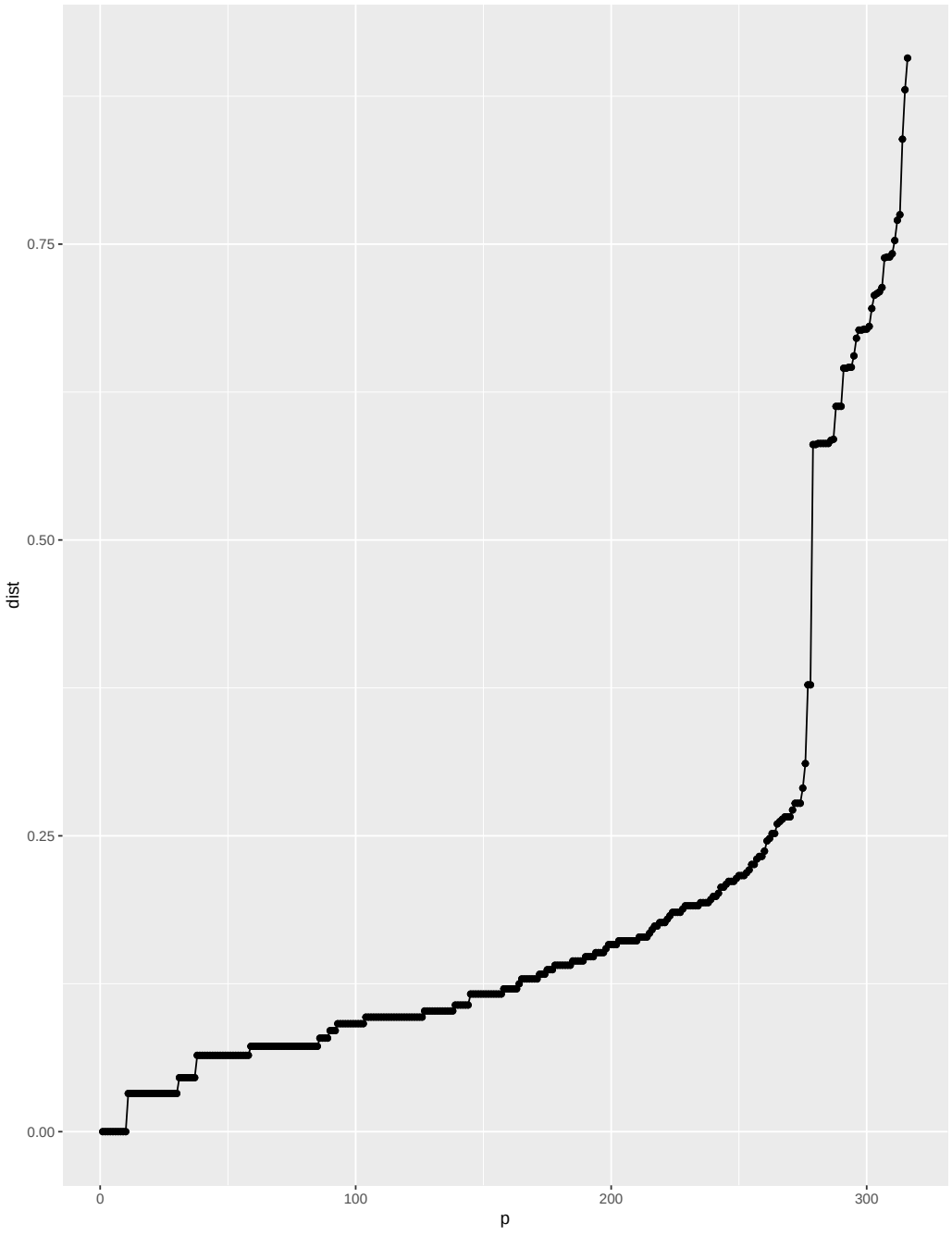

Figura 19: Distanze dei punti dal $k = 6$-esimo più vicino.

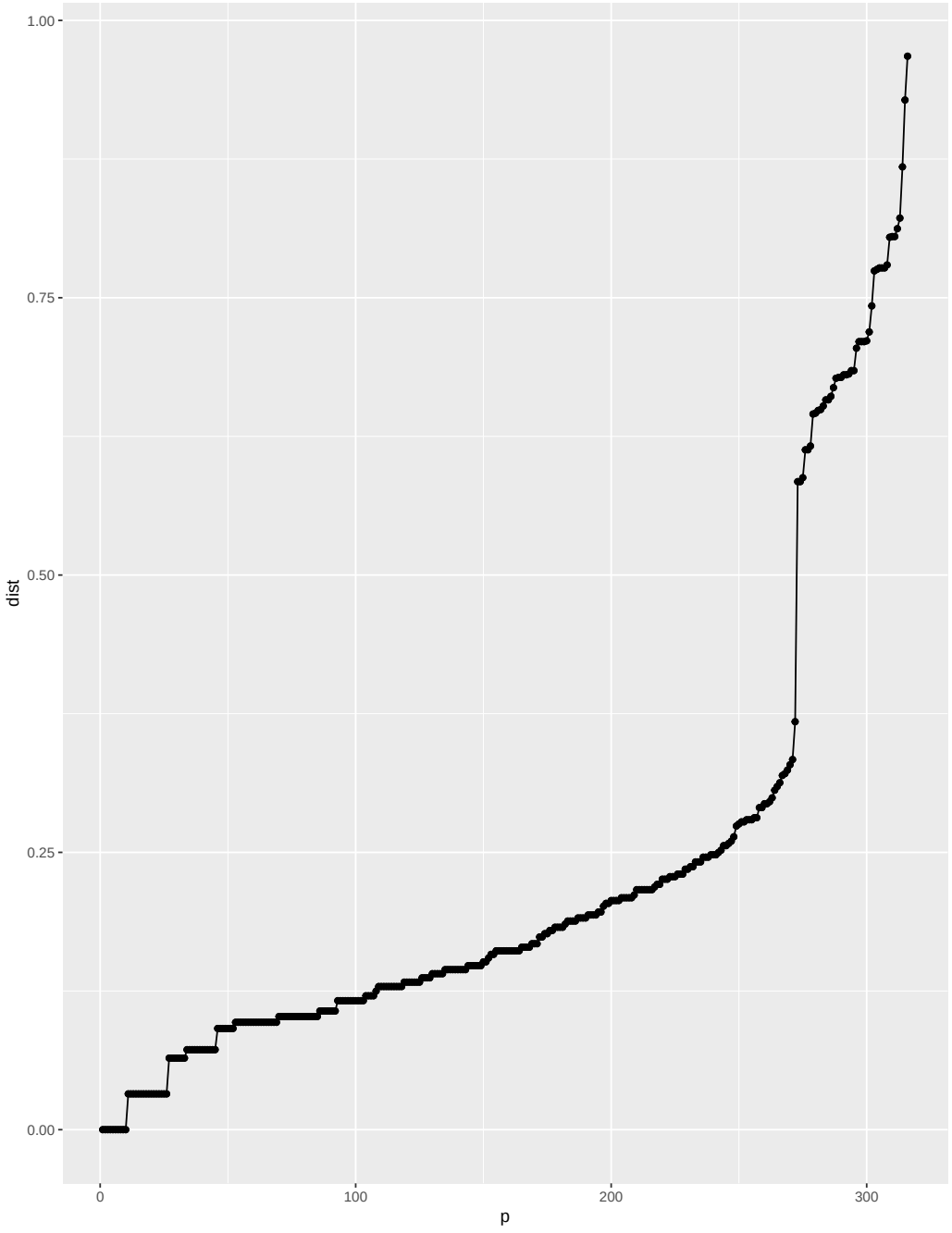

Figura 20: Distanze dei punti dal $k = 10$-esimo più vicino.

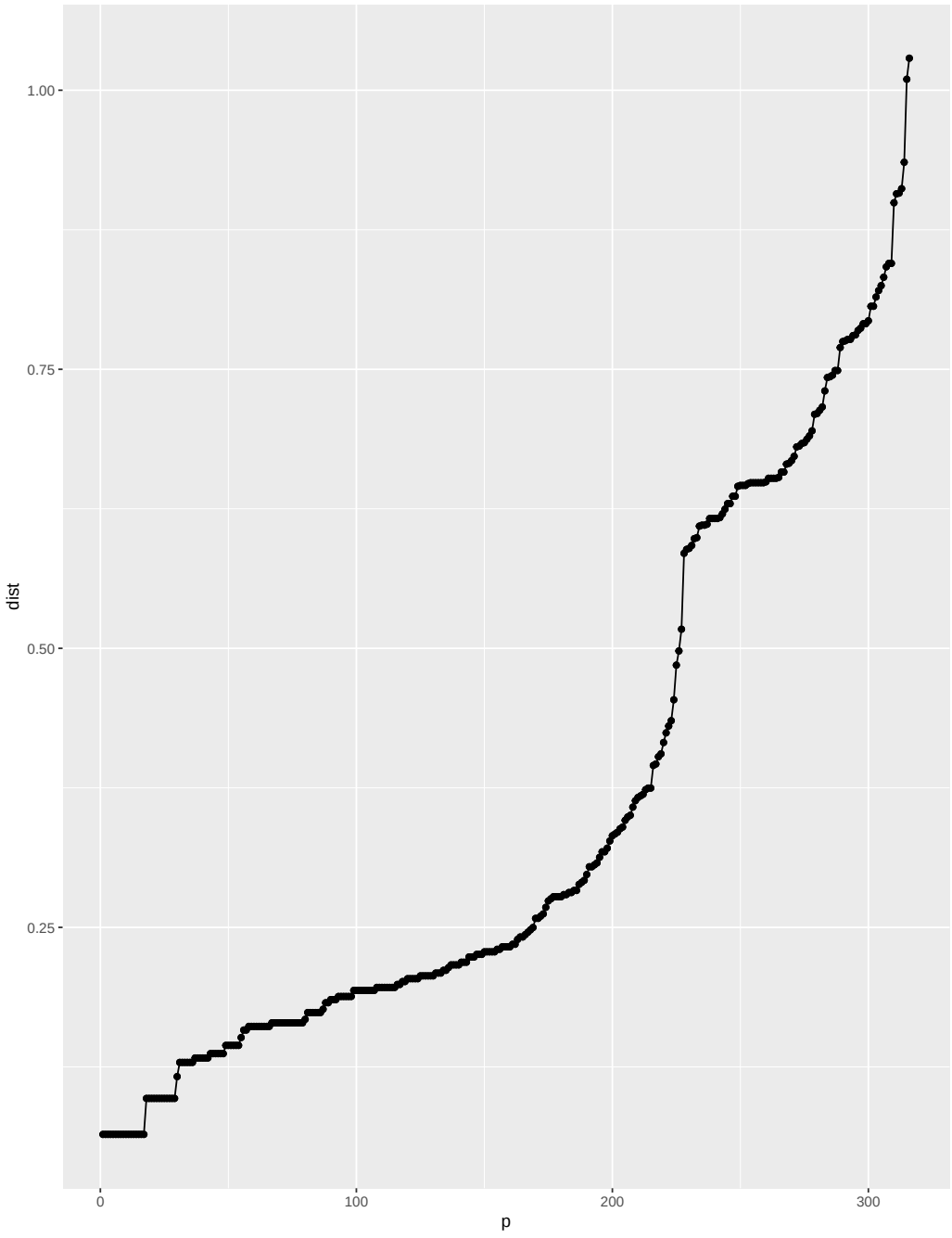

Figura 21: Distanze dei punti dal $k = 20$-esimo più vicino.

In Tabella 11 sono riportati i modelli ottimali determinati per i tre diversi aspetti del dataset che sono stati analizzati. Tali modelli sono stati scelti basandosi esclusivamente sul valore della correlazione tra la matrice di incidenza dei cluster e la matrice delle distanze e quindi nel caso di DBSCAN non viene tenuto conto della maggior perdita di dati che vengono etichettati come rumore nel modello ottimale riportato in Figura 21. In ogni caso, tenendo conto anche di questo aspetto, DBSCAN si è rilevato essere un algoritmo affidabile per l'analisi su cui è stato eseguito e anche l'esecuzione con gli altri parametri ha portato a risultati paragonabili a quelli del modello ottimale e decisamente migliori di quelli ottenuti con il K-means.

Attributi analizzati	Algoritmo Migliore	Parametri ottimali	Correlazione
crediti totali, ARC, PRG	K-means	$k = 3$	-0.854
ARC, ASD, PRG, MDL, AN1	DBSCAN	`MinPts=20,eps=0.4`	-0.758
voto medio, test	K-means	$k = 2$	-0.476

Tabella 11: Modelli ottimali.

6 Conclusioni

In riferimento ai risultati ottenuti con il K-means è possibile trarre le seguenti conclusioni riguardo le carriere degli studenti:

- l'esame di Architetture degli Elaboratori risulta essere l'esame più difficile per gli studenti del primo anno, infatti la maggioranza non riesce a sostenerlo nel corso della sessione estiva. Tuttavia, generalmente gli studenti che sostengono con profitto tale esame riescono a sostenere con un buon voto anche gli altri;

- la maggior parte degli studenti che ottengono un buon punteggio al test d'ingresso mantengono una buona media mentre quelli che hanno ottenuto un punteggio più basso hanno anche una media più bassa. Tuttavia è presente un significativo gruppo di studenti che pur avendo ottenuto un buon punteggio al test di ingresso non riescono ad avere una media altrettanto buona;

- La maggior parte degli studenti riesce a sostenere nel corso della sessione estiva gli esami di Algoritmi e Strutture Dati, Analisi I e in alcuni casi l'esame di Programmazione con risultati altalenanti. Mentre generalmente gli esami di Architetture degli Elaborati e Matematica Discreta e Logica non vengono sostenuti dagli studenti al termine del loro primo anno.

Inoltre, in ciascuna delle analisi eseguite risultavano esserci almeno 3 gruppi di studenti che presentavano caratteristiche significativamente differenti.

La tecnica del DBSCAN è risultata essere efficace e ha consentito di determinare dei clustering migliori rispetto a quelli determinati dal K-means nell'analisi dei voti degli studenti. Diversamente dai clustering ottimali determinati dal K-means, i clustering del DBSCAN presentano un alto numero di cluster diversi ed è quindi difficile (e inconcludente) descrivere il profilo dello studente medio appartente a ciascuno gruppo. Tuttavia, è necessario tenere presente che i clustering determinati con questa tecnica etichettano una buona parte dei dati totali come rumore (quasi un terzo nel clustering ottimale) e quindi l'analisi viene condotta necessariamente su un sottoinsieme del dataset.

Infine, per quanto riguarda il clustering gerarchico, è stato scelto di condurre delle analisi sulla proiezione del dataset relativa all'anno di immatricolazione del 2010. Tale scelta è motivata dal fatto che gli studenti

immatricolati nei diversi anni non presentano caratteristiche (voti ai test, agli esami, crediti totali, etc.) significativamente differenti e dal fatto che i dendogrammi dell'intero dataset risultavano incomprensibili. In ogni caso, l'analisi condotta con tali tecniche è limitata poiché si è scelto di valutare il numero di cluster con le tecniche presentate nel capitolo 5 e inoltre i relativi clustering non potevano essere valutati con il metodo della correlazione utilizzato nel capitolo 5 per K-means e DBSCAN.

Elenco delle figure

Elenco delle tabelle