# Clustering e Classificazione

January 16, 2018

**Abstract**

## 1 Preprocessing e Database

Per l'analisi del *dataset* fornito basata sulle tecniche di clustering e di classificazione è stato predisposto un database MySQL per la gestione dei dati e per la creazione di alcune viste che facilitassero lo studio del dataset tramite tali tecniche. Il database utilizzato presenta le seguenti tabelle:

- **students**: che contiene i dati principali legati agli studenti (*student_id*, *cohort*, *test_grade*, *hs_diploma_grade*, *hs_diploma_title*). La tabella è stata popolata prelevando i dati dal file "*anagrafica studenti*" fornito (*students.csv*), selezionando tutti gli attributi presenti nel file.
- **courses**: che contiene i dati relativi ai corsi, prelevati dal file *exams_preproc.csv* ottenuto nella precedente fase di preprocessing per l'analisi dei pattern e selezionando dunque solo gli attributi *course_id*, *cfu*, *description*.
- **exams**: che contiene i dati relativi agli esami sostenuti da ogni studente. Anche questa tabella è stata popolata prelevando i dati dal file ottenuto nella precedente fase di preprocessing e in particolare, selezionando gli attributi *student_id*, *course_id*, *date*, *grade*, *semester*.

Tutte le tabelle sono state strutturate scegliendo le corrette chiave primarie e inserendo le chiavi esterne opportune. Di seguito viene riportato il codice SQL utilizzato per il caricamento dei dati nel database.

Listing 1: import_data.sql

```
1 LOAD DATA LOCAL INFILE 'students.csv'
2 INTO TABLE students
3 FIELDS TERMINATED BY ','
4 ENCLOSED BY '"'
5 LINES TERMINATED BY '\n'
6 IGNORE 1 ROWS
7
```

```
(cohort, student_id, test_grade, hs_diploma_grade, hs_diploma_title);
LOAD DATA LOCAL INFILE 'exams_preproc.csv'
INTO TABLE courses
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(@dummy, course_id, @dummy, @dummy, cfu, description, @dummy);

LOAD DATA LOCAL INFILE 'exams_preproc.csv'
INTO TABLE exams
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(student_id, course_id, date, grade, @dummy, @dummy, semester);
```

Una volta strutturato e popolato le tabelle del database si è provveduto ad effettuare la fase di preprocessing, ovvero, la gestione dei dati mancanti, in parte eseguita direttamente nel database costruito, la costruzione delle viste necessarie alle analisi, lo studio della correlazione fra coppie di attributi e la normalizzazione di alcuni di essi per il corretto utilizzo degli algoritmi di clustering e di classificazione.

Per gestire la mancanza di alcuni voti del test di ingresso nel dataset "anagrafica studenti" fornito, è stato deciso di integrare i voti mancanti con una media complessiva dei risultati del test di ingresso di tutti gli studenti, utilizzando una semplice query di UPDATE sulla tabella *students*. Di seguito viene riportata la query SQL utilizzata.

Listing 2: preprocessing_students_table.sql

```
UPDATE students s1 , (SELECT round(avg(test_grade)) as avarage FROM
    students) s2
SET s1.test_grade = s2.avarage
WHERE s1.test_grade=0;
```

Per quanto riguarda lo studio attraverso le tecniche di clustering si è deciso di fare due principali analisi:

1. Sulla carriera e il percorso di ogni studente considerando dati come il voto ottenuto al test di ingresso, il voto di diploma e la media pesata del voto degli esami sostenuti, per cercare di raggruppare fra loro tutti gli studenti con carriere simili.

2. Sull'andamento dei risultati di ogni esame degli studenti, per cercare di raggruppare fra loro gli studenti che hanno ottenuto delle votazioni simili rispetto agli esami.

Al fine di eseguire le analisi sopra descritte sono state create rispettivamente le due seguenti viste in modo tale da raggruppare i dati necessari. Le viste sono state create in modo seguente.

- **cluster_career**: la quale contiene per ogni studente il voto al test di ingresso (*test_grade*), il voto di diploma (*diploma_grade*), la media pesata (sui *cfu*) dei voti degli esami sostenuti e arrotondata (*grade_weighted_avg*), il numero di esami sostenuti (*exams_taken*), il numero totale di cfu acquisiti (*total_cfu*) e un'ultima colonna con la differenza di anni tra la data dell'ultimo esame sostenuto e la corte dello studente (*years*). Gli attributi precedentemente descritti sono stati ricavati tramite script SQL dagli attributi delle tabelle del database (*Feature Creation*). Di seguito viene riportata una porzione della vista creata.

| student_id | test_grade | hs_diploma_grade | grade_weighted_avg | exams_taken | total_cfu | years |
|---|---|---|---|---|---|---|
| A | 18 | 80 | 27.0 | 10 | 90 | 4 |
| B | 13 | 67 | 23.0 | 10 | 96 | 4 |
| C | 18 | 78 | 25.0 | 7 | 69 | 4 |
| D | 14 | 66 | 23.0 | 7 | 66 | 2 |
| E | 16 | 82 | 28.0 | 2 | 24 | 2 |

- **cluster_exams**: la quale contiene per ogni studente i voti ottenuti ad ogni esame. Per questa vista, dato che molti studenti non hanno ancora sostenuto alcuni esami, abbiamo deciso di integrare i valori mancanti utilizzando la media dei voti dello studente. Tale modifica è stata fatta esportando la vista in un file *.csv* (contenente dunque anche i dati mancanti) e modificando quest'ultimo con uno script python. Di seguito viene riportata una porzione della vista con i dati aggiornati.

| student_id | B006800 | B006801 | B006802 | B006803 | B006804 | B006807 |
|---|---|---|---|---|---|---|
| A | 29 | 30.0 | 25.0 | 25.9 | 30.0 | 24.0 |
| B | 26 | 20.0 | 21.0 | 18.0 | 26.0 | 22.0 |
| C | 28 | 24.7142 | 22.0 | 24.7142 | 26.0 | 24.7142 |
| D | 20 | 28.0 | 22.0 | 23.1428 | 22.0 | 21.0 |
| E | 28 | 27.0 | 27.5 | 27.5 | 27.5 | 27.5 |

Per la classificazione abbiamo invece deciso, come verrà spiegato meglio in seguito nella fase di postprocessing, di recuperare le classi dai risultati del clustering eseguito sulla prima vista *cluster_career*, essendo tali risultati i più significativi.
Per la creazione del database viene fornito in allegato alla relazione il codice SQL dell'intera creazione del database "database_creation.sql" che ricrea completamente il database con i dati e le viste utilizzate.

Prima di applicare gli algoritmi di clustering e di classificazione sui dataset sono state fatte alcune analisi sui essi per comprendere meglio i dati a disposizione. È stata quindi calcolata la matrice di correlazione usando l'indice di *Pearson* sui dati relativi alla vista *cluster_career* per capire quanto i vari attributi sono correlati tra loro.

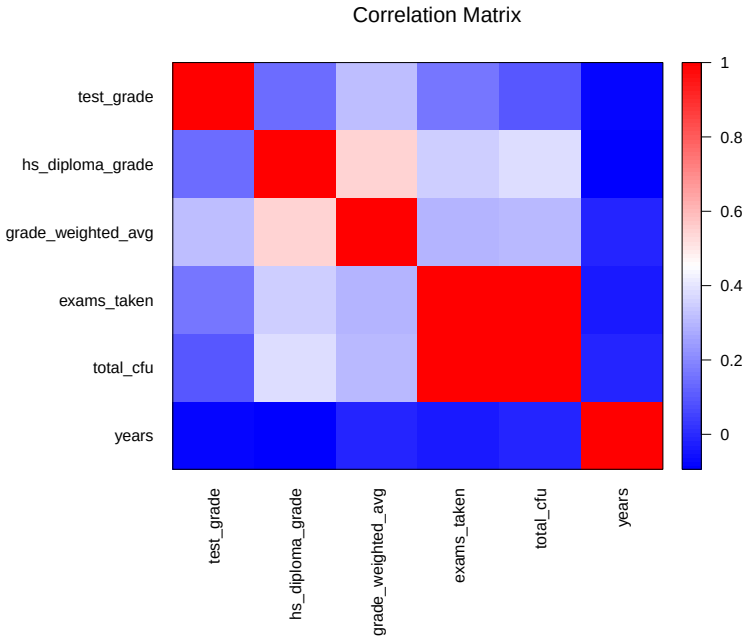


Figure 1: Matrice di correlazione per *cluster_career*.

| | test_grade | hs_diploma_grade | grade_weighted_avg | exams_taken | total_cfu | years |
|---|---|---|---|---|---|---|
| test_grade | 1 | 0.14486 | 0.32133 | 0.16350 | 0.098900 | -0.074509 |
| hs_diploma_grade | 0.14486 | 1 | 0.54464 | 0.35031 | 0.381964 | -0.094038 |
| grade_weighted_avg | 0.32133 | 0.54464 | 1 | 0.29749 | 0.303728 | -0.011494 |
| exams_taken | 0.16350 | 0.35031 | 0.29749 | 1 | 0.991599 | -0.038072 |
| total_cfu | 0.09890 | 0.38196 | 0.30372 | 0.99159 | 1 | -0.009638 |
| years | -0.0745 | 0.009638 | -0.0114 | -0.0380 | -0.00963 | 1 |

Come si può vedere dalla heatmap in Figura 1 ed in modo specifico dalla matrice di correlazione, è presente un'interessante discreta correlazione di circa $0,545$ tra la media pesata dei voti degli esami dello studente e il voto di diploma, lasciando dedurre che un buon risultato all'esame di maturità comporti in genere una buona media degli esami universitari. C'è inoltre una forte correlazione tra gli esami sostenuti e il numero di CFU ottenuti dallo studente, tale risultato risulta tuttavia abbastanza intuitivo, in quanto al crescere degli esami sostenuti crescerà anche il numero di CFU acquisiti dallo studente e viceversa. Infine, la

figura suggerisce come l'attributo *year* risulti quasi totalmente non correlato al resto degli attributi, probabilmente per il fatto che per la maggior parte degli studenti di questo dataset dall'anno di immatricolazione fino all'ultimo esame sostenuto sono passati due anni. Il resto degli attributi non presentano valori di correlazione significativi.

È stata successivamente calcolata la matrice di correlazione usando Pearson sui dati relativi alla vista *cluster_exams* per scoprire anche su di essa gli attributi più correlati.

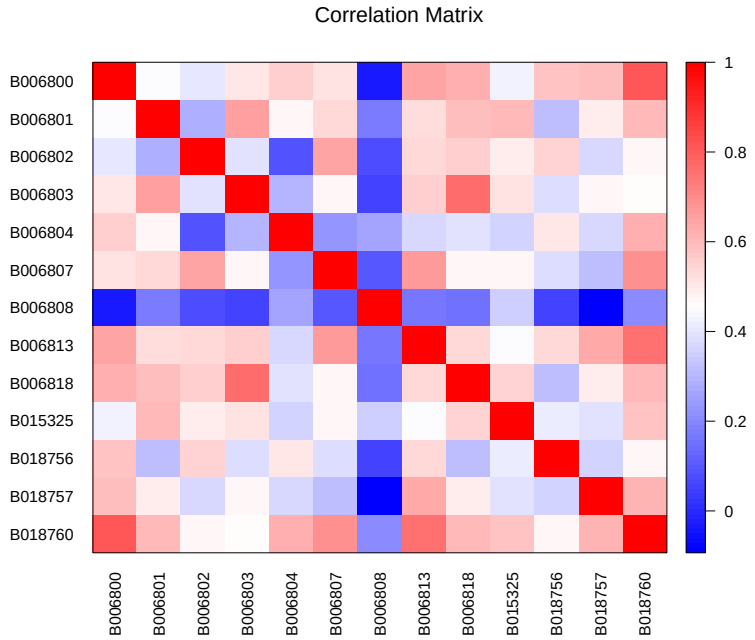


Figure 2: Matrice di correlazione per *cluster_exams*.

Come si può vedere dalla heatmap di Figura 2, abbiamo diversi esami che sembrano avere una correlazione nella distribuzione dei voti. Ad esempio esiste una forte correlazione dello 0.812 tra l'esame di Algoritmi e Strutture Dati (B006800) e l'esame di Calcolo delle Probabilità e Statistica (B018760), una correlazione dello 0.762 tra Sistemi Operativi (B006818) e Matematica Discreta e Logica (B006803) ed una di 0.759 tra BSDI (B006813) e CPS (B018760), le quali rappresentano le correlazioni più forti. La figura mostra inoltre come l'esame di Analisi II (B006808) risulti quasi totalmente incorrelato dal resto degli esami, in quanto la maggior parte degli studenti ha una votazione bassa oppure non hanno sostenuto l'esame. Queste correlazioni però potrebbero essere imprecise dato l'alto numero di dati mancanti che mostrava il *dataset* fornito,

ovvero molti studenti che non hanno ancora sostenuto tutti gli esami.

Per applicare gli algoritmi di clustering al data set è stato utilizzato il software Weka, collegandolo direttamente al database MySQL per estrarre facilmente le viste sulle quali applicare gli algoritmi.

Prima di applicare gli algoritmi è stato però eseguita un'ulteriore fase di preprocessing sui dati della vista *cluster_career* direttamente in Weka. Durante tale fase sono stati normalizzati tutti gli attributi della vista eccetto l'identificatore dello studente (*student_id*) in scala da 0 a 1 al fine di evitare problemi dovuti a scale di valori differenti per gli attributi. Per quanto riguarda invece la vista *cluster_exams* non è stata fatta nessuna normalizzazione sui dati in quanto ogni campo ha lo stesso range di valori, essendo tutti voti di esami da 18 a 31.

Tutte e due le viste predisposte per il clustering sono disponibili anche in due differenti file *.csv*, rispettivamente nel file *cluster_career.csv* e *cluster_exams.csv* allegati a questa relazione per facilitare la riproduzione dei risultati nella fase di applicazione degli algoritmi di clustering.

# 2 Clustering

Come prima analisi si mostra quella relativa alla carriera dello studente usando i dati estrapolati dalla vista *cluster_career* resi disponibili nel file *cluster_career.csv*.

Inizialmente è stata fatta un'analisi per capire il numero di cluster nascosti nel dataset. Per questo scopo è stato utilizzato l'algoritmo di clustering gerarchico agglomerativo con metodo *Complete Link*, al fine di usare opportunamente un metodo robusto per gestire gli *outliers* presenti ed eventuale rumore nei dati. L'algoritmo è stato eseguito utilizzando come funzione di distanza quella Euclidea, selezionando un numero di cluster pari ad 1 (per non ricercare un numero di cluster specifico), lasciando tutti gli altri parametri di default e selezionando i primi tre attributi della vista (test_grade, hs_diploma_grade, grade_weighted_avg), questo giustificato dal fatto che, come mostrato in precedenza, sono gli attributi più correlati tra loro e quindi più significativi.

Come si può vedere dal dendogramma risultante in Figura 3, l'algoritmo divide il set in due principali cluster. Per avere una conferma di questo raggruppamento è stato utlizzato sul set anche il metodo *Group Avarage* ottenendo il dendogramma in Figura 4.

Anche in questo caso si è avuta la conferma che i cluster nascosti sono principalmente due.

Prima di eseguire un algoritmo di clustering ricercando un numero di cluster specifico sono state fatte alcune prove anche utilizzando l'algoritmo *DB-Scan* non ottentendo però risultati soddisfacenti dovuti probabilmente al fatto di operare con un database molto ridotto e poco denso.

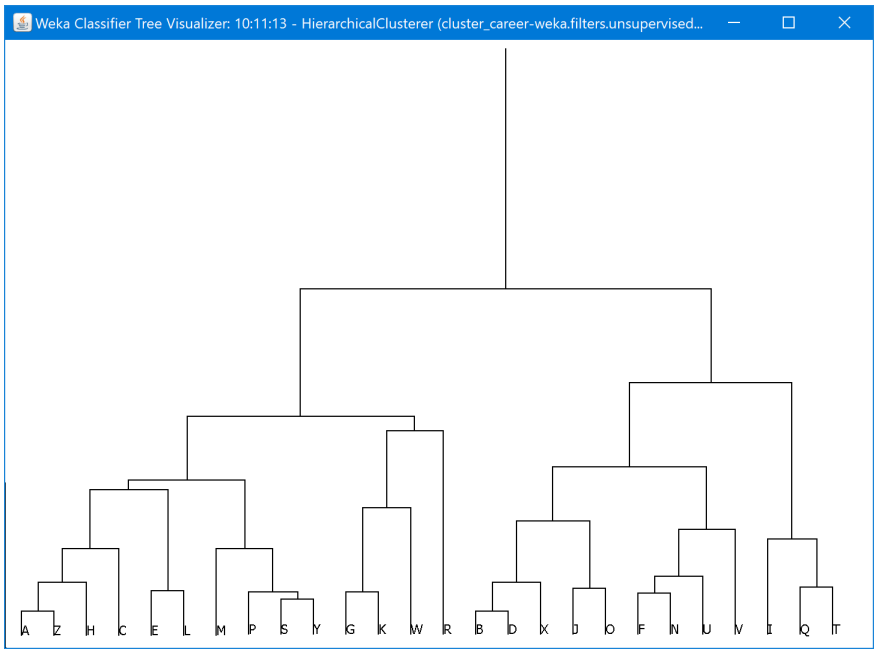


Figure 3: Clustering gerarchico Complete Link su *cluster_career*.

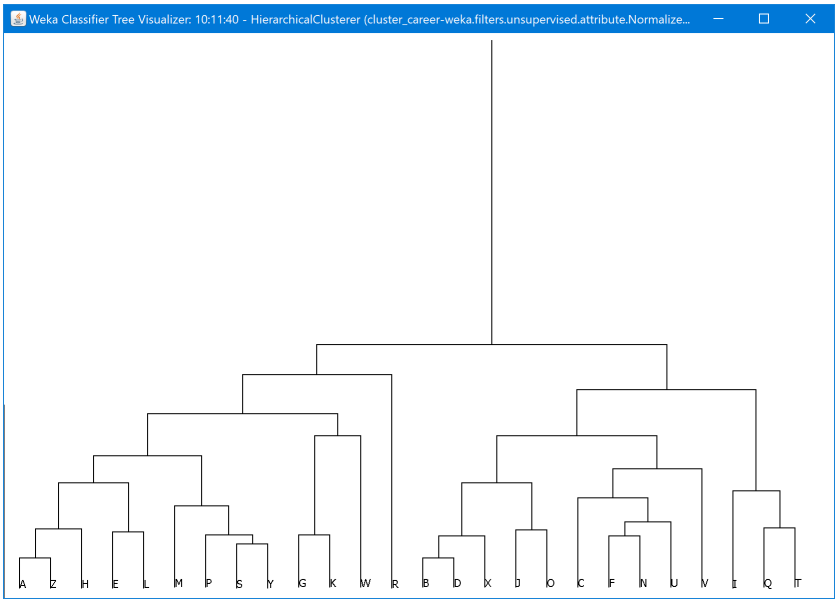


Figure 4: Clustering gerarchico Group Avarage su *cluster_career*.

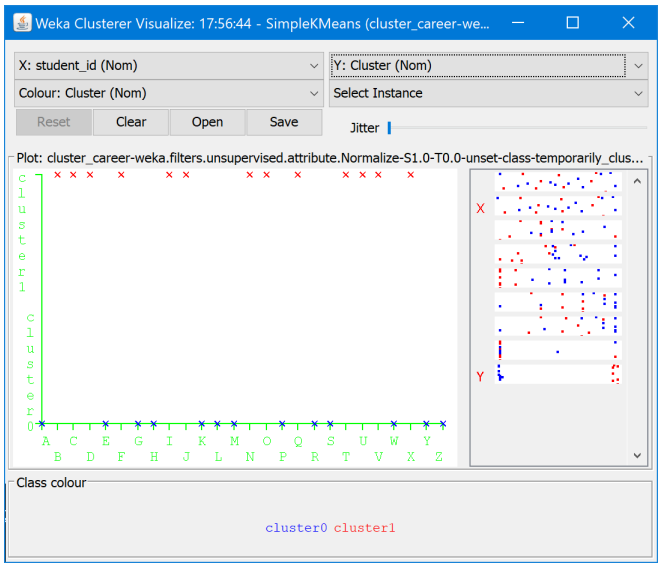


Figure 5: Assegnazioni k-means clustering (3 attributi) su *cluster_career*.

Appurato quindi che i cluster principali sono due, l'insieme dei dati è stato analizzato usando l'algoritmo *k-means* implementato in Weka, eseguito utilizzando la distanza Euclidea, specificando due cluster da ricercare e lasciando i valori di default per la generazione casuale dei centroidi (seed = 10).

L'algoritmo k-means è stato eseguito con due configurazioni di dati, la prima, come fatto per l'algoritmo di clustering gerarchico, considerando solo i primi tre attributi della vista (*test_grade*,*hs_diploma_grade*, *grade_weighted_avg*) ed ignorando tutti gli altri. I risultati ottenuti sono, una proporzione del 50% degli studenti assegnata ad ogni cluster (13 studenti in tutti e due i cluster su un totale di 26 studenti), un SSE pari a 3.19 ed i seguenti centroidi.

| Cluster | test_grade | hs_diploma_grade | grade_weighted_avg |
|---|---|---|---|
| 0 | 0.5315 | 0.659 | 0.6044 |
| 1 | 0.4406 | 0.3056 | 0.0659 |

In particolare nella Figura 5 viene mostrata l'assegnazione di ogni studente al suo cluster. Mentre nella Figura 6 viene mostrato il plot di incrocio tra i due attributi più correlati (*hs_diploma_grade* e *grade_weighted_avg*).

Successivamente, durante la seconda esecuzione del k-means, sono stati considerati tutti gli attributi della vista, escludendo solo lo *student_id*. I risultati ottenuti in questo caso sono, una proporzione del 46% (12 studenti) assegnata al cluster 0 ed una proporzione del 54% (14 studenti) assegnata al cluster 1, un SSE pari a 8.37 ed i seguenti centroidi:

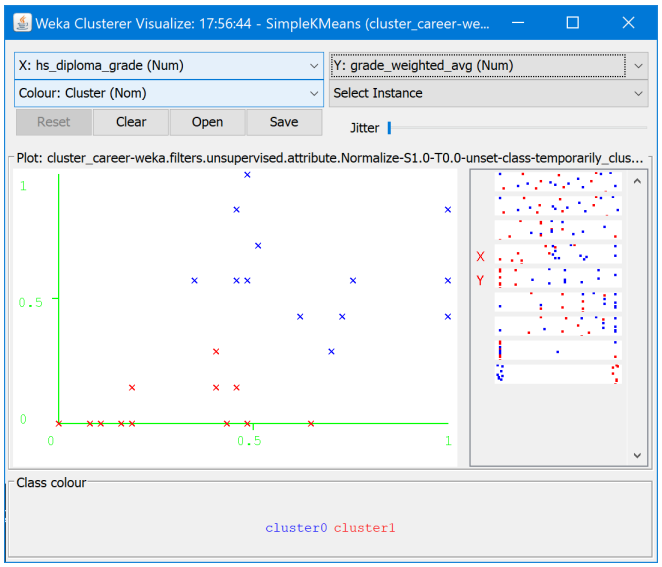


Figure 6: k-means clustering su *cluster_career* (3 attributi), incrocio fra due attributi.

| Cluster | test_grade | hs_diploma_grade | grade_weighted_avg | exams_taken | total_cfu | years |
|---|---|---|---|---|---|---|
| 0 | 0.553 | 0.6486 | 0.5476 | 0.9167 | 0.914 | 0.125 |
| 1 | 0.4286 | 0.3398 | 0.1531 | 0.513 | 0.5115 | 0.1429 |

Nella Figura 7 viene mostrata l'assegnazione di ogni studente al suo cluster. Mentre nella Figura 8 viene mostrato il plot di incrocio tra i due attributi più correlati.

Analizzando i risultati dei centroidi e degli assegnamenti, si può notare come in entrambi i casi, gli studenti siano divisi in due categorie, in base alla loro appartenenza ad uno o all'altro cluster, quella degli studenti con un miglior percorso, ovvero quelli con una miglior carriera universitaria, e quella relativa agli studenti con un percorso peggiore, evidenziato anche dalla Figura 9.
Come si può vedere nelle Figure 6 e 8, i due gruppi sono chiaramente visibili e separati in base ai risultati ottenuti dagli studenti durante la loro carriera universitaria. La differenza principale tra le due sta in alcuni studenti che, pur avendo ottenuto voti abbastanza alti durante la loro carriera universitaria, hanno sostenuto pochi esami e che quindi vengono penalizzati nel seconda operazione di clustering, in quanto viene tenuto conto anche del numero e dell'importanza (cfu) degli esami dati.

Mostriamo ora le seconda analisi relativa all'andamento dei risultati di ogni esame degli studenti, usando i dati estrapolati dalla vista *cluster_exams*, aggiornando i dati mancanti come spiegato nella fase di preprocessing, e resi disponibili nel file *cluster_exams.csv*.

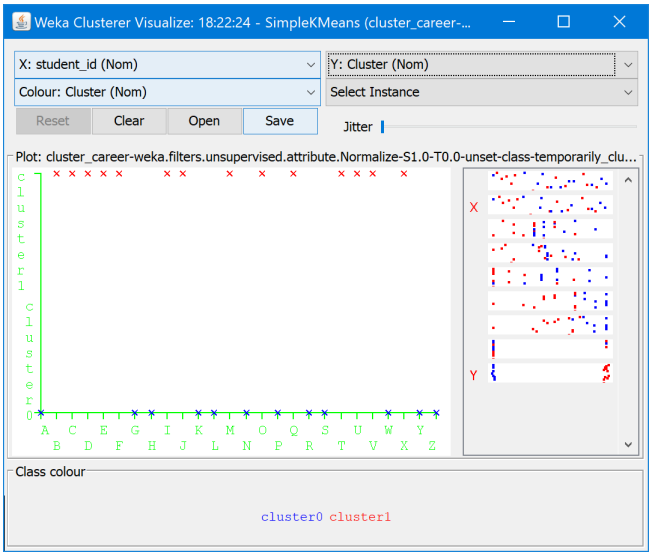


Figure 7: Assegnazioni k-means clustering su *cluster_career*.

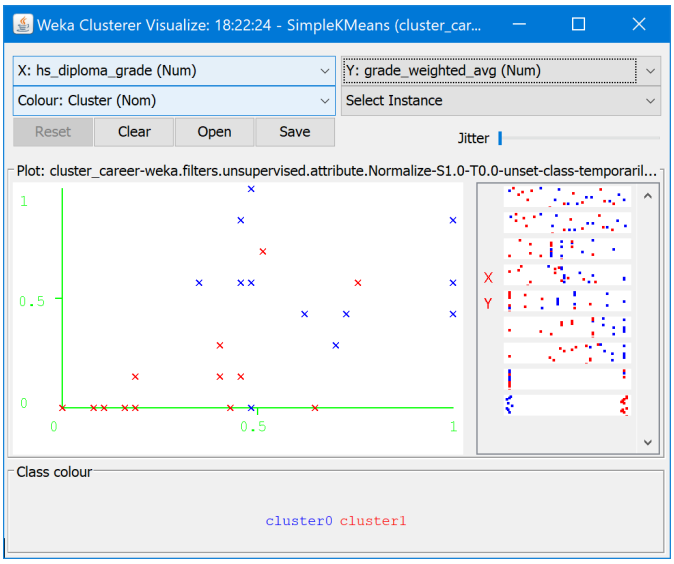


Figure 8: k-means clustering su *cluster_career*, incrocio fra due attributi.

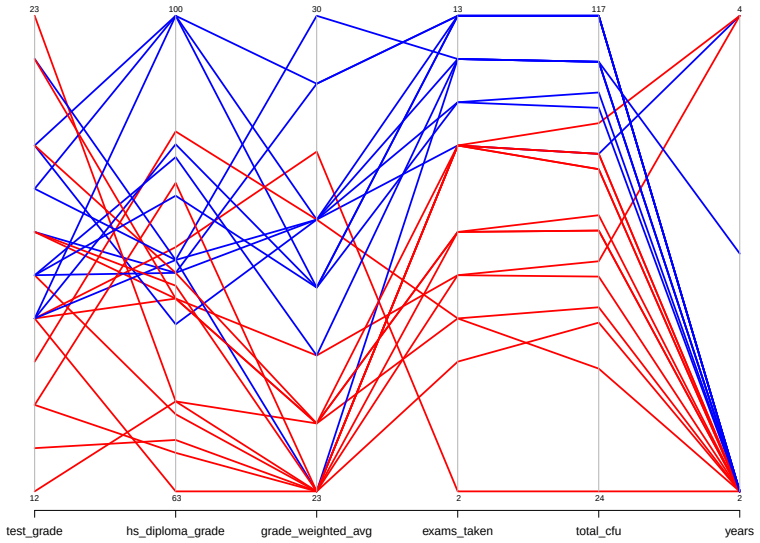


Figure 9: Coordiante Parallele per il k-means clustering su *cluster_career*.

Anche in questo caso inizialmente è stata fatta un'analisi per capire il numero di cluster nascosti nel dataset utilizzando alcuni algoritmi di clustering gerarchici ed ottenendo come nel caso precedente un numero di lcuster nascosti pari a due.

Appurato quindi che i cluster principali sono due, l'insieme dei dati è stato analizzato usando l'algoritmo *k-means*, eseguito utilizzando la distanza Euclidea, specificando due cluster da ricercare e lasciando i valori di default per la generazione casuale dei centroidi (seed = 10). I risultati ottenuti sono, una proporzione del 50% degli studenti assegnata ad ogni cluster (13 studenti in tutti e due i cluster su un totale di 26 studenti), un SSE pari a 17.96 ed i seguenti centroidi.

| Cluster | B006800 | B006801 | B006802 | B006803 | B006804 | B006807 | B006808 | B006813 | B006818 | B015325 | B018756 | B018757 | B018760 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 29.2308 | 28.3077 | 26.2436 | 25.7324 | 27.8846 | 27.859 | 23.3462 | 27.8077 | 26.0821 | 25.6875 | 28.0734 | 28.1667 | 28.1247 |
| 1 | 24.5385 | 24.1165 | 21.1282 | 22.5961 | 25.1667 | 22.578 | 23.7415 | 22.7584 | 23.0661 | 23.6839 | 25.0238 | 23.9648 | 22.028 |

In particolare nella Figura 10 viene mostrata l'assegnazione di ogni studente al suo cluster. Mentre nella Figura 11 viene mostrato il plot di incrocio tra i due attributi più correlati (ASD e CPS).

Analizzando i risultati dei centroidi e degli assegnamenti, si può notare come anche in questo caso gli studenti siano divisi in due categorie, quella degli studenti con una sequenza di voti più alti e quella relativa agli studenti con una sequenza di voti peggiore. Inoltre si nota, osservando la Figura 10 come gli assegnamenti ai cluster siano molto simili (si differenziano per alcuni studenti assegnati diversamente) a quelli ottenuti per il clustering sulla carriera degli


Figure 10: Assegnazioni k-means clustering su *cluster_exams*.

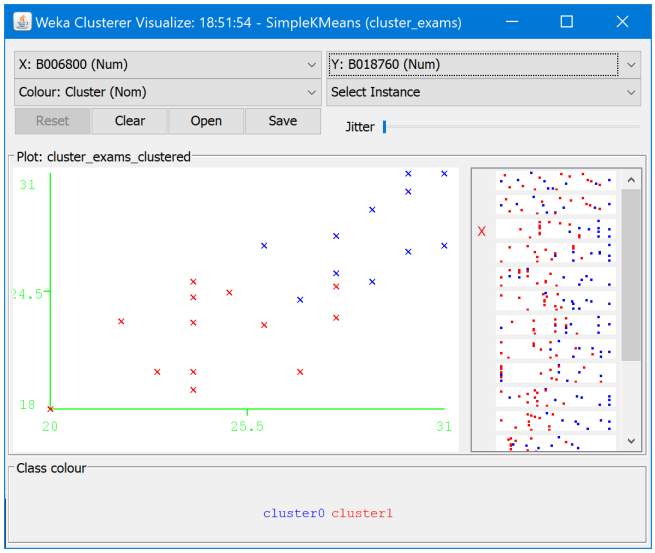


Figure 11: k-means clustering su *cluster_exams*, incrocio fra ASD e CPS.

studenti confermando dunque l'accuratezza del clustering precedente come raggruppamento delle carriere ottimali e di quelle meno positive. Guardando la Figure 11 possiamo vedere come sono stati formati i due gruppi, chiarimenti visibili, se confrontati su i due attributi più correlati.

# 3 Postprocessing e Classificazione

Come ultima analisi si è voluto cercare di assegnare una particolare classe agli studenti per poi applicare al dataset un algoritmo di classificazione per comprendere se ci sia un modo, attraverso il dataset fornito, di prevedere se la carriera e il percorso intrapreso da uno studente sia "positivo" o "negativo", ovvero se lo studente riuscirà ad ottenere una carriera soddisfacente o meno, basando tale previsione sugli attributi già visti per la vista *cluster_career*

Per eseguire quest'ultima analisi per il *training set* si è deciso di scegliere le classi basandoci sulle assegnazioni ai cluster ottenute per i risultati del clustering sulla carriera, ovvero, sulle assegnazioni mostrate in Figura 8 sulla vista *cluster_career*.

Una volta ottenuto le assegnazioni attraverso il software Weka è stato creato il file *classification_career.csv* a partire dal file *cluster_career.csv* inserendo una nuova colonna "*class*" dedicata appunto alle assegnazioni ottenute dai risultati del clustering sulla vista *cluster_career*, ottenendo una tabella simile a quella seguente (ne viene riportata solo una porzione).

| student_id | test_grade | hs_diploma_grade | grade_weighted_avg | exams_taken | total_cfu | years | class |
|---|---|---|---|---|---|---|---|
| A | 18 | 80 | 27.0 | 10 | 90 | 4 | positive |
| B | 13 | 67 | 23.0 | 10 | 96 | 4 | negative |
| C | 18 | 78 | 25.0 | 7 | 69 | 4 | negative |
| D | 14 | 66 | 23.0 | 7 | 66 | 2 | negative |
| E | 16 | 82 | 28.0 | 2 | 24 | 2 | negative |

Ad ogni studente è stata assegnata la classe "*positive*" nel caso in cui lo studente sia stato precedentemente assegnato al cluster 0 ottenuto nella fase di clustering oppure la classe "*negative*" nel caso in cui lo studente sia stato precedentemente assegnato al cluster 1, in modo tale da rispettare i diversi raggruppamenti basati sulla qualità della carriera e del percorso dello studente. In Figura 12 viene mostrato un grafico di riassunto sulle caratteristiche del training set *classification_career.csv* così creato.

A questo punto, il training set *classification_career.csv*, è stato importato in Weka e i vari attributi del set sono stati normalizzati direttamente attraverso tale software ad eccezione dello *student_id* e dell'attributo *class*. Come algoritmo di classificazione è stato scelto il J48, per classificare tramite alberi di decisione, utilizzando i parametri di default, scegliendo come classe l'attributo *class* del set e scegliendo come opzione per il test set il metodo *Cross-Validation*.

I risultati ottenuti sono, una percentuale di istanze correttamente classificate pari al 96.15% (*Accuracy*) e una matrice di confusione riportata qui di seguito. Viene inoltre riportato in Figura 13 l'albero di decisione creato dall'algoritmo J48.

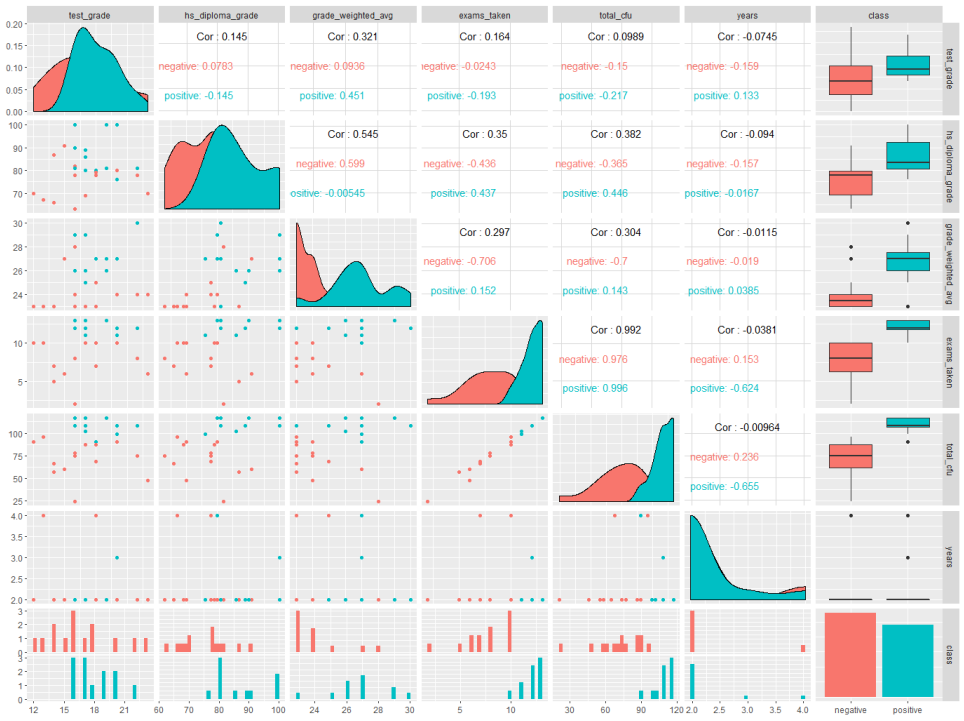


Figure 12: Grafico di riassunto del set *classification_career.csv*.

| | Predicted Class | | |
|---|---|---|---|
| Actual Class | | Class=Positive | Class=Negative |
| | Class=Positive | 11 | 1 |
| | Class=Negative | 0 | 14 |

Purtroppo come è facilmente osservabile dalla Figura 13 i risultati della classificazione non sono molto interessanti dato che l'albero di decisione finale ha un solo livello e lo split scelto risulta essere un semplice controllo sul numero totale di esami dati, se quest'ultimi superano il valore normalizzato di circa 0.73, che corrisponde ad un numero di esami pari a 10 allora la carriera dello studente verrà classificata come "positiva" altrimenti sarà classificata come "negativa". Anche utilizzando altri tipi di metodi per la generazione del test set i risultati sono circa gli stessi, ottenendo quasi sempre una classificazione con un accuratezza del 96.15% e dunque corrispondente ad un solo studente classificato in modo errato.

I risultati ottenuti sulla classificazione sono però sicuramente penalizzati dalla dimensione molto ridotta del dataset fornito e anche dal peso di molti datti mancanti, sopratutto relativi agli esami sostenuti.

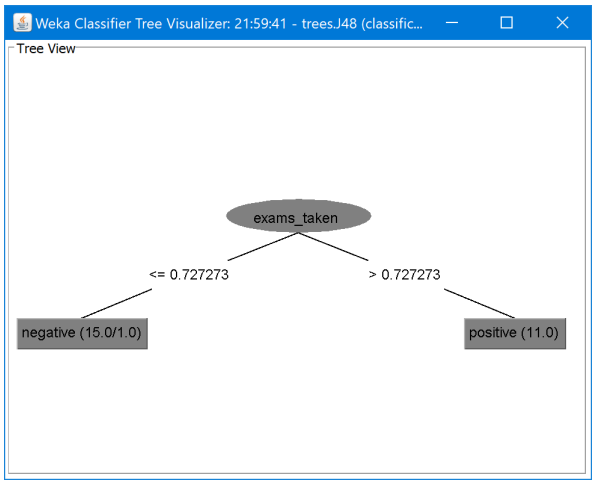


Figure 13: Albero di decisione.