Football Manager 2020 incontra la simulazione: guida ad un calciomercato "easy" grazie ad algoritmi di clusterizzazione

Andrea Lucini Paioni[1]

[1]*CdLM Data Science, Università degli studi di Milano Bicocca*

Ogni anno, il giocatore/manager appassionato di Football Manager, si getta a capofitto nel mondo creato dalla Sport Interactive, pronto a scoprire le principali novità del gioco e a condurre al successo la propria squadra. Ogni anno, tra le mille novità proposte, il giocatore/manager cerca di trovare risposta alla domanda che ogni appassionato si è posto, almeno una volta, giocando ad un titolo di questa saga di videogiochi manageriali: chi sono e quali caratteristiche hanno i *wonderkids*, ovvero i talenti grezzi nascosti tra la miriade di giocatori presenti nel gioco, pronti ad esplodere indossando la divisa della propria squadra?

Indice

Introduzione

Football Manager [1] [2] è una serie di videogiochi di simulazione, sviluppata con l'obiettivo primario di creare una riproduzione, sempre più verosimile per ogni nuova edizione del gioco, del ruolo di allenatore e gestore di una squadra di calcio professionistico o dilettantistico.

Sviluppata dall'azienda Sport Interactive a partire dal 2004, la serie è composta da ormai 20 versioni diverse uscite nell'arco degli ultimi 18 anni, in cui l'azienda ha perfezionato sempre di più il gioco dal punto di vista della cura dei dettagli, sviluppando un numero sempre maggiore di funzioni che rappresentino il più fedelmente possibile le realtà del calcio dilettantistico e professionistico, attirando così centinaia di migliaia di fan ogni anno provenienti da tutto il mondo.

Il gioco consiste nello scegliere quale squadra iniziare a gestire tra le migliaia presenti – ad esempio, la versione uscita nel 2019, ovvero Football Manager 2020, e che sarà quella considerata all'interno di questo progetto, permette di scegliere una squadra da ben 118 campionati diversi, situati in 54 paesi differenti – e di controllare tutti gli aspetti del gioco: dalla gestione della propria rosa (con allenamenti sia di squadra che mirati ai singoli calciatori, e tattiche personalizzabili sia a livello di formazione che di singoli giocatori per ogni partita e situazione di gioco all'interno della stessa), alla gestione della parte economica del club (rinnovi di contratto dei giocatori e dei dirigenti, gestione del budget e delle risorse finanziarie del club, scelta dello staff, gestione del settore giovanile, ricerca di sponsor, rispetto dei parametri economici imposti da organi come UEFA e FIFA e soprattutto gestione del calciomercato), fino ad arrivare alla simulazione delle partite vere e proprie, con possibilità di influire sull'esito delle partite con scelta di tattiche, formazione, consigli ai singoli giocatori, interviste pre e post-partita...

Di tutte queste componenti del gioco, uno dei compiti principali del manager/giocatore riguarda proprio la costruzione della propria rosa attraverso movimenti di calciomercato, col fine di portare la propria squadra a raggiungere gli obiettivi prefissati all'inizio di ogni stagione. E per riuscire nel proprio intento, è fondamentale creare una rosa equilibrata, con giocatori con caratteristiche precise in base al tipo di gioco ricercato e alle tattiche da applicare durante la stagione, mettendo calciatori in contesti adatti a

valorizzare e sviluppare le loro qualità, e contemporaneamente dando possibilità ai giovani di migliorare sia nel proprio club che con prestiti mirati a valorizzarli e a farli crescere.

Uno dei punti chiave sta nel capire quali siano i giocatori più adatti per raggiungere i propri obiettivi, dunque la fase di scouting è uno degli aspetti fondamentali che può portare la squadra al successo. A tal proposito, una prima fase di calciomercato mirata ai cosiddetti *wonderkids* – ovvero quei giovani talenti che possono migliorare velocemente e in modo molto significativo, ma che si possono ancora acquistare a cifre contenute perché non ancora sbocciati – può portare ad uno sviluppo molto veloce della propria squadra, dando un grosso boost sia ai risultati sul campo che al raggiungimento degli obiettivi prefissati dal club (in termini di valorizzazione dei giovani ed incremento del valore della rosa).

Tuttavia non è sempre semplice trovare giocatori dalle caratteristiche ideali per il proprio stile di gioco, e anzi spesso subentrano svariate problematiche che impediscono l'acquisto di un determinato giocatore (contratto appena rinnovato, richiesta troppo elevata per il budget del club, scarso interesse del giocatore nell'unirsi al proprio club…): questo progetto mira a fornire al manager/giocatore di FM20 uno strumento che permetta di trovare velocemente profili simili a quei calciatori "irraggiungibili", senza dover necessariamente impostare decine e decine di filtri.

2. Dataset e preprocessing

Il dataset utilizzato [3] in questo studio contiene informazioni riguardanti 144750 giocatori con 62 parametri differenti, da cui si otterranno 69 colonne:

1. *Name* (stringa): il nome del calciatore
2. *Position* (stringa): la/le posizioni in cui il calciatore può giocare
3. *Club* (stringa): il club del calciatore
4. *Division* (stringa): il campionato del club di appartenenza del calciatore
5. *Based* (stringa): la nazione in cui si trova il campionato di appartenenza del calciatore
6. *Nation* (stringa): la nazionalità del calciatore
7. *Height* (in cm): l'altezza del calciatore
8. *Weight* (in kg): il peso del calciatore
9. *Age*: l'età del calciatore
10. *Preferred.Foot* (stringa): il piede preferito
11. *Best.Pos* (stringa): la posizione in cui il calciatore si trova più a suo agio
12. *Best.Role* (stringa): il ruolo più adatto
13. *Value* (in Euro): il valore di mercato
14. *Wage* (in Euro): lo stipendio settimanale
15. *CA*: la cosiddetta CA (Current Ability), ovvero un valore intero da 0 a 200, non visibile all'interno del gioco, che indica il livello di abilità del calciatore ad inizio gioco; a valori alti corrispondono giocatori più forti
16. *PA*: la cosiddetta PA (Potential Ability), ovvero un valore intero da 0 a 200, non visibile all'interno del gioco, che indica il livello di abilità massima potenziale del calciatore; a valori alti corrispondono giocatori più forti
17. *Work_Rate* (valore intero da 0 a 20): la volontà, capacità del calciatore di lavorare al 100% delle sue potenzialità
18. *Vision* (valore intero da 0 a 20): la visione di gioco del calciatore
19. *GK_Throwing* (valore intero da 0 a 20): l'abilità del portiere di passare con le mani
20. *Technique* (valore intero da 0 a 20): la qualità tecnica del calciatore
21. *Teamwork* (valore intero da 0 a 20): l'abilità del calciatore di fare gioco di squadra
22. *Tackle* (valore intero da 0 a 20): l'abilità del calciatore di rubare palla senza fallo in tackle
23. *Strenght* (valore intero da 0 a 20): la forza fisica del calciatore
24. *Stamina* (valore intero da 0 a 20): la resistenza alla fatica del calciatore
25. *GK_TRO* (valore intero da 0 a 20): la tendenza del portiere ad uscire dai pali (Tendency Rushing Out)
26. *GK_Reflex* (valore intero da 0 a 20): i riflessi del portiere
27. *GK_Punching* (valore intero da 0 a 20): l'abilità del portiere nel respingere il pallone di pugno
28. *Positioning* (valore intero da 0 a 20): l'abilità del calciatore di posizionarsi correttamente e in modo intelligente in campo
29. *Penalty* (valore intero da 0 a 20): l'abilità del calciatore nel calciare un rigore
30. *Passing* (valore intero da 0 a 20): l'abilità nei passaggi del calciatore
31. *Pace* (valore intero da 0 a 20): la velocità massima del calciatore
32. *GK_1v1* (valore intero da 0 a 20): l'abilità del portiere negli 1 contro 1
33. *OTB* (valore intero da 0 a 20): l'abilità Off The Ball, ovvero la capacità di muoversi in modo intelligente senza palla
34. *Natural_Fitness* (valore intero da 0 a 20): quanto è probabile che un calciatore sia in salute/infortunato nel corso della stagione (valori alti indicano un calciatore sano).

35. *Marking* (valore intero da 0 a 20): l'abilità in marcatura del calciatore
36. *Long_Throw* (valore intero da 0 a 20): l'abilità di un calciatore di fare rimesse laterali lunghe
37. *Long_Shot* (valore intero da 0 a 20): l'abilità di un calciatore di tirare in modo preciso e potente dalla distanza
38. *Leadership* (valore intero da 0 a 20): la capacità di un calciatore di influenzare in modo positivo i giocatori vicini in campo
39. *GK_Kicking* (valore intero da 0 a 20): la distanza a cui il portiere riesce a calciare
40. *Jumping* (valore intero da 0 a 20): l'abilità del calciatore nel saltare per colpire di testa
41. *Heading* (valore intero da 0 a 20): l'abilità nel colpo di testa del calciatore
42. *GK_Handling* (valore intero da 0 a 20): l'abilità del portiere di parare e trattenere il pallone
43. *Free_Kick* (valore intero da 0 a 20): l'abilità del calciatore di essere pericoloso su calci di punizione, sia diretti in porta che diretti a compagni di squadra
44. *Flair* (valore intero da 0 a 20): la creatività e l'imprevedibilità del calciatore
45. *First_Touch* (valore intero da 0 a 20): l'abilità del calciatore nello stoppare un pallone
46. *Finishing* (valore intero da 0 a 20): l'abilità del calciatore nel segnare quando si presenta un'occasione da rete
47. *GK_Eccentricity* (valore intero da 0 a 20): la tendenza del portiere di essere imprevedibile nelle scelte e nei comportamenti
48. *Dribbling* (valore intero da 0 a 20): l'abilità nel dribbiling del calciatore
49. *Determination* (valore intero da 0 a 20): il livello di impegno e la determinazione di un calciatore, dentro e fuori dal campo
50. *Decisions* (valore intero da 0 a 20): la capacità di un calciatore di prendere decisioni corrette in tutte le circostanze, con o senza palla
51. *Crossing* (valore intero da 0 a 20): l'abilità nel crossare del calciatore
52. *Corner* (valore intero da 0 a 20): l'abilità del calciatore di crossare calci d'angolo pericolosi
53. *Concentration* (valore intero da 0 a 20): la capacità del calciatore di mantenere la concentrazione in ogni circostanza
54. *Composure* (valore intero da 0 a 20): l'abilità del calciatore di prendere decisioni corrette in situazioni delicate
55. *GK_Communication* (valore intero da 0 a 20): la capacità del portiere di comunicare in modo efficace con i propri compagni di squadra
56. *GK_COA* (valore intero da 0 a 20): la capacità del portiere di prendere il comando dell'area di rigore, ovvero Command Of Area
57. *Bravery* (valore intero da 0 a 20): quanto è disposto un calciatore a mettersi al servizio della squadra, anche a costo di rischiare infortuni o cartellini
58. *Balance* (valore intero da 0 a 20): l'equilibrio
59. *Anticipation* (valore intero da 0 a 20): l'abilità del calciatore di prevedere e reagire a situazioni di gioco
60. *Agility* (valore intero da 0 a 20): la reattività
61. *Aggression* (valore intero da 0 a 20): l'aggressività in campo del calciatore
62. *Aerial_Reach* (valore intero da 0 a 20): le capacità fisiche del calciatore nel gioco aereo
63. *Acceleration* (valore intero da 0 a 20): quanto velocemente un calciatore raggiunge la sua velocità massima.
64. *miglioramento*: differenza tra PA e CA.
65. *Position1* (stringa): 1° posizione in cui può giocare il calciatore
66. *Position2* (stringa): 2° posizione in cui può giocare il calciatore
67. *Position3* (stringa): 3° posizione in cui può giocare il calciatore
68. *Position4* (stringa): 4° posizione in cui può giocare il calciatore
69. *Position5* (stringa): 5° posizione in cui può giocare il calciatore

In particolare, rispetto al dataset originale, per meglio osservare lo sviluppo di un calciatore è stata creata la variabile *miglioramento*, che calcola la differenza tra *PA* e *CA*, in modo tale da avere valori molto elevati per giocatori destinati a migliorare molto col procedere delle stagioni all'interno del gioco.

Infine, si può notare che la variabile *Position* definisce uno o più ruoli in cui il calciatore può giocare: per questo motivo è stata effettuata, tramite il nodo Cell Splitter, una divisione della colonna *Position*. Sono state trovate al massimo cinque occorrenze nella stessa cella all'interno della colonna *Position*, quindi sono state create cinque colonne (*Position1, Position2, Position3, Position4 e Position5*), ognuna contenente una sola delle posizioni in cui il calciatore può essere schierato, e con valori uguali a *None* nelle rispettive celle quando un calciatore presentava meno di cinque posizioni in campo.

L'obiettivo dell'analisi, dunque, è di eseguire diversi metodi di clusterizzazione con il fine di trovare quello che individua i più precisi gruppi di calciatori; ovvero

cluster formati in modo tale che le caratteristiche dei calciatori appartenenti ad ogni gruppo siano le più uniformi possibile all'interno degli stessi, e il più possibile diverse rispetto a tutti gli altri cluster.

3. Ricerca dei wonderkids e metodi di clusterizzazione

La ricerca dei cosiddetti *wonderkids* è fondamentale all'interno del gioco, perché permette, quando si decide di iniziare con squadre con un budget di mercato basso, di migliorare velocemente nel gioco pur senza spendere cifre elevatissime.

Risulterebbe molto interessante ricercare profili di questi giocatori basandosi sulla posizione in campo, così da avere diverse opzioni a disposizione per poter comporre una squadra equilibrata, o anche solo completare la propria formazione con i profili adatti in base alle lacune presenti nella rosa: tuttavia, dato il numero di occorrenze diverse molto elevato sia per quanto riguarda la variabile *Position* (812 differenti), che per la variabile *Best.Role* (44 diversi), oltre alle grandi differenze presenti nelle diverse colonne *Position1, Position2, Position3, Position4 e Position5*, tutti questi sono criteri difficilmente utilizzabili per osservare gruppi differenti di calciatori.

Quindi, per classificare i calciatori in base alle loro caratteristiche, procediamo con un metodo di Cluster Analysis; per ottenere osservazioni interessanti per la ricerca di *wonderkids*, vengono filtrati i giocatori in base ad alcuni criteri: in ordine, vengono considerati solo i giocatori con valori consistenti, pari o superiori a 25, per la variabile *miglioramenti* (osservando il box plot relativo a questa variabile, i valori al di sopra di 49 vengono considerati tutti outliers, mentre il miglioramento medio di un giocatore si assesta attorno al valore 16, e il terzo quartile esattamente a 25); inoltre, dei 41168 calciatori rimanenti, vengono considerati solamente quelli con valore di abilità potenziale *PA* maggiore di 130, ovvero i futuri prospetti di livello alto o molto alto, portando il totale a 3883 calciatori; infine, trattandosi di *wonderkids*, escludiamo dalle osservazioni considerate quelle con calciatori di età superiore a 27 anni, ottenendo in questo modo un totale di 2859 calciatori su cui procedere con l'analisi. Un ultimo criterio utilizzato riguarda i portieri, che differenziandosi molto dai calciatori di movimento, e avendo una serie di caratteristiche peculiari della loro posizione, vengono esclusi dal dataset, portando ad un numero totale di 2597 calciatori (ovviamente, vengono escluse diverse variabili dal dataset relative puramente ai portieri, come *GK_Throwing, GK_TRO, GK_Reflex, GK_Punching, GK_1v1, GK_Kicking, GK_Handling, GK_Eccentricity, GK_Communication* e *GK_COA*).

Nel procedere con la Cluster Analysis su questo dataset ridotto, formato da 2597 calciatori con 59 colonne, sono state scelte differenti tecniche, tra cui:

- **Cluster gerarchici**
- **SOMs**
- **Metodo delle k-medie**
- **K-medoids**

Clusterizzazione gerarchica

I primi metodi utilizzati fanno parte dei metodi di clusterizzazione gerarchica: in particolare, si tratta di tipologie diverse di *agglomerative methods*, che si possono distinguere dai *divisive methods* in quanto partono da un numero di cluster pari al numero totale di osservazioni del dataset utilizzato, e procedono unendo a due a due i cluster fino ad ottenerne uno solo (i *divisive methods*, al contrario, partono da un unico cluster, che tenderà a dividersi con l'avanzare del processo iterativo fino ad ottenere un cluster per ogni osservazione).

Uno dei compiti principali di questi metodi risiede nell'individuare il numero di cluster più adatto a definire il dataset di interesse: poiché questi algoritmi non forniscono questo valore in modo automatico, si ricercherà tramite osservazione del relativo dendrogramma – ovvero osservando dove c'è un forte incremento della distanza tra un passaggio da X a X+1 gruppi – e dello *scree plot* corrispondente – dunque il grafico che permette di confrontare il numero di gruppi con la distanza di aggregazione tra di essi –.

In questo progetto, verranno utilizzati differenti metodi di aggregazione gerarchica, tra cui:

- Metodo del legame singolo (con distanza euclidea), che aggrega due cluster aventi minore distanza tra i due punti più vicini appartenenti ad uno ed all'altro cluster;
- Metodo del legame medio (con distanza euclidea): che aggrega due cluster aventi minore distanza media tra tutti i punti appartenenti ad uno ed all'altro cluster;
- Metodo del legame completo (con distanza euclidea): che aggrega due cluster aventi minore distanza tra i due punti più lontani appartenenti ad uno ed all'altro cluster;

Grazie al nodo Hierarchical Clustering [4], dunque, si è eseguita una clusterizzazione gerarchica per i tre metodi del legame singolo, del legame medio e del legame completo, ottenendo i tre dendrogrammi riportati sopra (*figura 1, dall'alto verso il basso metodo del legame singolo, del legame medio e del legame completo*).

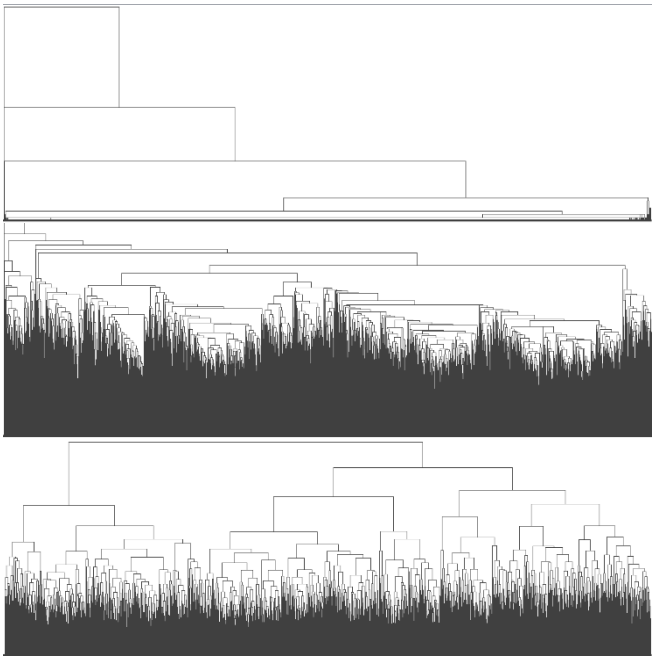

Figura 1

Sia in base ai dendrogrammi che in base agli *scree plot* (*figura 2, dall'alto verso il basso metodo del legame singolo, del legame medio e del legame completo*), risultano molto interessanti le divisioni ottenute suddividendo il dataset in 7 cluster (la divisione migliore, solo guardando la composizione dei gruppi, sembra quella ottenuta con legame completo; invece, sia legame singolo che legame medio tendono a raggruppare diverse osservazioni in un unico cluster, e ottengono in questo modo alcuni cluster molto piccoli, formati da una sola osservazione).

Tuttavia bisogna tenere conto che sia il metodo a legame singolo che quello a legame medio tendono a compiere errori, nel processo di clusterizzazione, in caso di presenza di numerosi outliers (come in questo dataset). Inoltre, osserviamo che, ad esempio, per il metodo del legame completo si potrebbe scegliere una divisione in due parti, se si volesse massimizzare la distanza tra i gruppi, però per lo scopo del progetto sembra essere una divisione troppo poco specifica: conviene invece creare più gruppi differenti per poter cercare profili di calciatori con caratteristiche affini.

Si potrebbe anche decidere di dividere in un numero superiore di cluster, ma come suggerisce lo scree plot corrispondente ai vari metodi, le distanze tra gruppi X

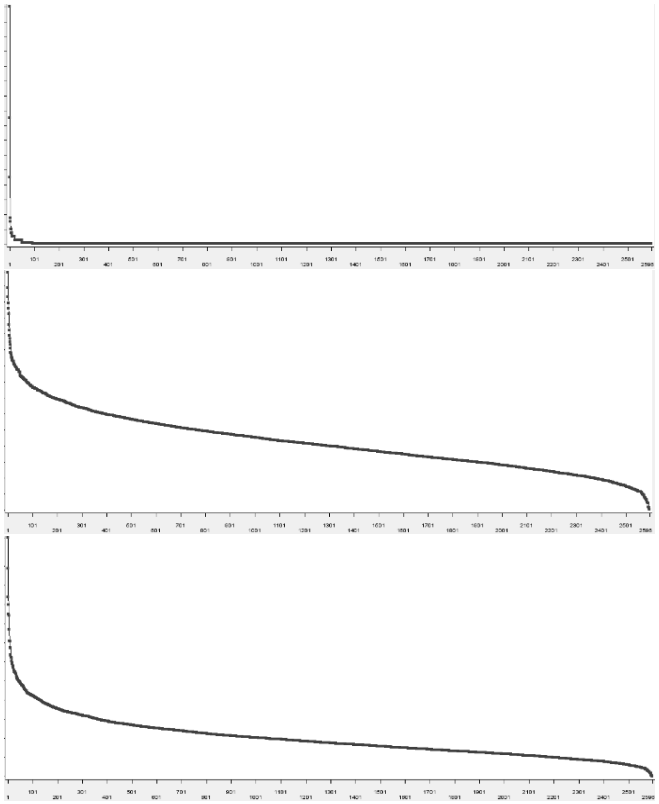

Figura 2

e X+1 sono inferiori all'aumentare del numero di cluster (ad esempio, negli *scree plot* del metodo del legame medio e del legame completo, dopo la divisione in 7 gruppi le distanze si riducono in modo considerevole), quindi rimane migliore la suddivisione in 7 cluster, che verrà approfondita in seguito in fase di valutazione del metodo di clusterizzazione migliore.

SOM

Le SOM, o Self Organizing Maps, sono un metodo non lineare di riduzione della dimensionalità, che permette di associare in cluster osservazioni con caratteristiche simili. In particolare, si tratta di un processo iterativo [7]: ad ogni osservazione corrisponde una prima fase di aggiornamento del centroide, definito "*codebook*", che più si avvicina per caratteristiche all'osservazione. A seguito di questa fase, per ogni osservazione viene calcolata la distanza euclidea tra questa ed ogni *codebook* – così da trovare il *codebook* vincente – tramite questa formula:

$$d(v - c_i) = ||v - c_i||_2$$

Quando è stato trovato il codebook vincente, si procede nella fase di aggiornamento dei centroidi, all'interno della visualizzazione topografica, che avviene in base a questa funzione:

$$h(s, i|i^*) = \alpha(s) e^{-\frac{||cell_i - cell_{i^*}||^2}{2\sigma(s)}}$$

con:
- $cell_i$* che è il centroide del codebook vincente;
- $cell_i$ che è la cella di appartenenza della singola osservazione;
- α(s) e σ(s) come funzioni decrescenti (infatti l'aggiornamento del centroide è maggiore se la distanza dell'osservazione dallo stesso è minore, da cui le funzioni decrescenti).

Il processo prosegue presentando le osservazioni ai codebook fino all'ultimo, e poi si ripete in base al numero di fasi, anche dette epoche, prestabilite nelle fasi preparatorie dell'algoritmo. I codebook si aggiorneranno sempre meno con l'avanzare della procedura iterativa, fino a quando la mappa topografica potrà essere considerata stabile (ovvero nuove epoche cambierebbero di pochissimo i centroidi dei codebook)

Dopo avere normalizzato i dati, utilizzando il nodo SelfOrganizingMap (3.7), e successivamente il nodo Weka Cluster Assigner (3.7), si ottengono 4 cluster, tutti di buone dimensioni (rispettivamente 918, 584, 742 e 353 osservazioni).

Metodo delle k-medie

Un altro metodo di Cluster Analysis molto utilizzato, soprattutto per l'efficienza e la semplicità dello stesso, è quello delle k-medie, ovvero un processo iterativo che seleziona un numero predefinito di centroidi, crea dei cluster associando ad ogni centroide tutte le osservazioni più vicine ad esso e ricalcola i centroidi basandosi su tutte le osservazioni appartenenti al cluster; il processo si ripete fino a quando arriva a produrre un risultato che non cambia ulteriormente il centroide di ogni cluster.

Uno dei problemi principali di questo algoritmo è la scelta del numero di cluster, e quindi di centroidi. Basandosi su quanto osservato coi metodi precedenti, dove non era necessario decidere a priori il numero di cluster (come la clusterizzazione gerarchica), si implementano le k-medie con 7 centroidi.

Dopo aver scelto un seed in modo tale da poter ripetere la procedura con gli stessi parametri, viene eseguito l'algoritmo tramite il nodo k-Means (configurato con massimo numero di interazioni pari a 5000). A partire da esso, otteniamo 7 cluster abbastanza uniformi per quanto riguarda la numerosità (come minimo 237 calciatori nel cluster 5, fino ad arrivare ad un massimo di 497 nel cluster 0), e tramite i nodi Interactive Pie Chart, GroupBy e Conditional Box Plot possiamo osservare sia la composizione vera e propria dei cluster in un diagramma a torta, sia i valori medi di tutte le variabili stratificate in base ai cluster, sia i boxplot delle singole variabili stratificate in base ai cluster ottenuti.

Vista la velocità di esecuzione del metodo, questo può essere tenuto in considerazione come uno degli algoritmi migliori se si volessero effettuare molti tentativi cambiando il numero di cluster richiesto. Tuttavia, è giusto tenere conto che l'algoritmo rischia di portare a risultati distorti sia per cardinalità differenti dei gruppi all'interno del dataset, che in caso di outliers numerosi: per questo motivo, si può ricorrere ad un altro metodo simile alle k-medie, ovvero il metodo dei k-Medoids.

k-Medoids

Il k-Medoids è un tipo di algoritmo che deriva dal metodo delle k-medie, cercando di risolvere alcuni dei problemi principali che affliggono quest'ultimo: infatti, la differenza principale tra i due risiede nel fatto che il k-Medoids non utilizza il centroide di ogni cluster, bensì utilizza l'osservazione che più si avvicina al centroide. In questo modo, nonostante il procedimento sia più lungo e dispendioso delle k-medie, non solo si risolve parzialmente il problema degli outliers, ma limita anche l'eventuale problema di avere un dataset con troppo rumore al suo interno.

Su Knime, dopo aver normalizzato i dati e trovato la matrice di distanze euclidee tra punti del dataset, si utilizza il nodo k-Medoids (considerando 7 cluster come nelle k-medie per poterli confrontare, e utilizzando *set_seed* = 12345 per poter ripetere la procedura ottenendo il medesimo risultato): otteniamo 7 cluster attorno ai calciatori Mechoso (riga n°5747, formato da 290 calciatori), Dobrev (riga n°1908, formato da 420 calciatori), Doyle-Hayes (riga n°2894, formato da 432 calciatori), Benedetti (riga n°4785, formato da 286 calciatori), De Marino (riga n°7545, formato da 507 calciatori), Pedro Pereira (riga n°2331, formato da 309 calciatori) e Tsagua (riga n°5453, formato da 353 calciatori).

4. Valutazione clusterizzazioni effettuate

Dopo aver utilizzato diversi metodi sia per la scelta del numero di cluster, sia per la vera e propria associazione di ogni osservazione al cluster corrispondente, vengono ora utilizzati alcuni indici esterni (anche definiti supervisionati) ed interni (o definiti non supervisionati) per valutare la bontà delle differenti clusterizzazioni.

Tra gli indici supervisionati utilizzati per osservare i differenti algoritmi di clusterizzazione, abbiamo:

1. L'indice di Rand $R = \frac{(a+d)}{M}$ con $R \in [0,1]$
2. L'indice di Jaccard $J = \frac{a}{a+b+c}$ con $J \in [0,1]$
3. L'indice di Fowlkes and Mallows $FM = \sqrt{\frac{a}{a+b} \times \frac{a}{a+c}}$ con $FM \in [0,1]$

Ora si possono osservare i valori degli indici sui tre metodi di cluster gerarchici, sulle k-medie e sui k-Medoids, come mostrato dalla *Tabella 1* qua sotto:

Row ID	single	average	complete	kmeans	pam
rand	0.985	0.772	0.622	0.84	0.783
jaccard	0.985	0.771	0.195	0.324	0.208
fow.mal	0.993	0.877	0.349	0.49	0.346

Tabella 1

Per quanto riguarda tutti e tre gli indici, il miglior algoritmo risulta essere il metodo del legame singolo, con valori molto vicini all'unità; segue il metodo del legame medio, con valori abbastanza elevati (attorno allo 0.8 per tutti e tre gli indici), mentre gli altri metodi hanno valori più elevati per quanto riguarda l'indice di Rand, ma valori molto più bassi sia per l'indice di Jaccard che per l'indice di Fowlkes and Mallows.

Anche nel caso degli indici non supervisionati, sono tre quelli che vengono utilizzati per la valutazione dei metodi di clusterizzazione:

1. Il coefficiente di Dunn, con valori compresi tra 0 e + ∞, per identificare cluster densi al loro interno e ben separati rispetto ai cluster esterni; sono preferibili metodi di clusterizzazione con valori elevati di questo indice.
2. Il coefficiente di Silhouette, con valori compresi tra -1 e +1, per affermare se gli elementi sono clusterizzati bene all'interno dei vari gruppi; sono preferibili valori vicini a +1 (ovvero valori inseriti nei cluster corretti), mentre valori vicini allo 0 mostrano dataset con molte osservazioni in conflitto tra diversi cluster, e valori vicini a -1 indicano osservazioni classificate erroneamente.
3. Il coefficiente di Connectivity, in cui a valori bassi corrispondono metodi di clusterizzazione che classificano meglio sul dataset fornito.
4. Il Cophenetic Correlation Coefficient, indice di correlazione tra la dissimilarità della matrice dei dati e la dissimilarità del dendrogramma ottenuto con i differenti metodi usati (legame singolo, legame medio e legame completo); valori elevati indicano metodi di clusterizzazione efficaci.

hierarchical	Connectivity	45.4063
	Dunn	0.2933
	Silhouette	0.2193
kmeans	Connectivity	1722.5992
	Dunn	0.1246
	Silhouette	0.0721
som	Connectivity	1824.5996
	Dunn	0.0698
	Silhouette	0.0797
pam	Connectivity	2531.8794
	Dunn	0.0670
	Silhouette	0.0484

Tabella 2

Come indicato in *tabella 2*, tutti i valori ottenuti ci portano a ritenere la clusterizzazione gerarchica come la scelta migliore, col dataset di calciatori e 7 cluster: infatti, abbiamo sia i valori più bassi di connettività, sia il valore più elevato dell'indice di Dunn (anche se comunque un valore basso, indice di difficoltà nell'ottenere cluster con calciatori molto differenti come caratteristiche rispetto a quelli degli altri gruppi), sia l'indice di Silhouette più elevato (ma anche per questo, si evidenzia presenza di osservazioni in conflitto tra differenti cluster).

È giusto chiedersi se il problema alla base di alcuni metodi di clusterizzazione non possa essere trovato nel numero di cluster: la risposta a tale quesito si può ricercare nei differenti valori degli indici di Dunn, di Silhouette e di Connectivity al variare del numero di cluster. Come riportato nella figura sopra (*figura 3*), gli algoritmi di clusterizzazione gerarchica (rappresentati dai numeri 1), le SOM (rappresentate dai numeri 2) e le k-medie (rappresentate dai numeri 3) non avrebbero presentato grossi miglioramenti in caso di scelte differenti come numero di cluster. Si possono notare generalmente valori leggermente migliori in corrispondenza di un numero minore di cluster (2/3 cluster al massimo conducono a valori migliori

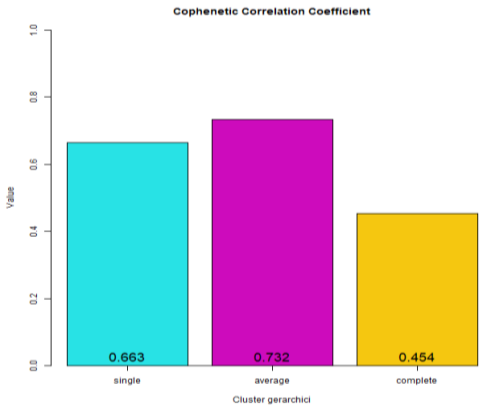

Figura 4

dell'indice di Silhouette e dell'indice di Connectivity), ma una così netta riduzione nel numero di cluster sarebbe in contrasto con l'obiettivo di questo progetto: infatti, avremmo pochissimi cluster formati da giocatori aventi ruoli e caratteristiche diverse, ma raggruppati insieme.

La *figura 4*, invece mostra il Coefficiente di Correlazione Cofenetica per quanto riguarda i metodi del legame singolo, medio e completo: in questo caso, con 7 cluster considerati, viene suggerito il metodo del legame medio, con un discreto valore di 0.73, subito seguito dal metodo del legame singolo a 0.63.

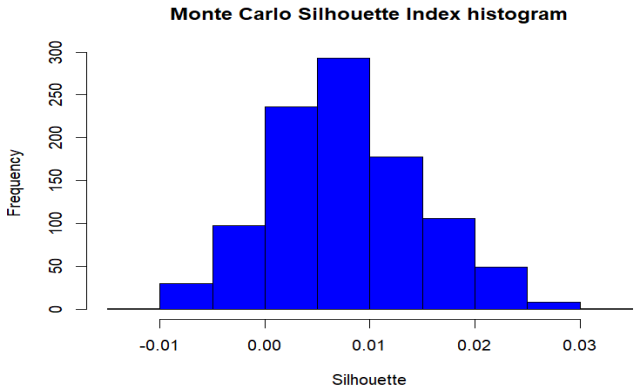

Figura 5

Tuttavia, per poter constatare la validità del metodo di clusterizzazione, dobbiamo verificare il paradigma di validità, che consiste nel determinare un indice di validazione che ci permetta di rifiutare l'ipotesi nulla per cui non esiste struttura all'interno del dataset. Viene utilizzato il coefficiente di Silhouette relativo alla clusterizzazione gerarchica, con distanza euclidea e metodo del legame medio, e su questo viene posta l'ipotesi H_0: applicando il metodo di Monte Carlo (*figura 5*), che simula una quantità finita di volte (stabilita pari a 1000) la distribuzione empirica, è possibile verificare l'ipotesi nulla. Avendo ottenuto un valore di 0.02, rispetto al valore del coefficiente di Silhouette trovato in precedenza (pari a 0.21), possiamo rifiutare l'ipotesi nulla, e quindi considerare valida la scelta di clusterizzazione gerarchica con metodo del legame medio.

5. Conclusioni

I diversi metodi di cluster analisi proposti all'interno di questo progetto, e applicati al database di calciatori di Football Manager 2020, hanno fatto emergere pregi e difetti degli stessi (facilità e velocità computazionale delle k-medie, ma difficoltà con outliers e cardinalità dei cluster; semplicità di utilizzo ma difficoltà nell'individuare il numero ideale di cluster per i modelli gerarchici; precisione nei risultati ma pesantezza computazionale per i k-Medoids; una buona via di mezzo a livello di prestazioni e qualche difficoltà nella fase di programmazione per le SOM), non portando ad un modello che risolvesse completamente il problema di ricerca dei wonderkids sul mercato di FM, ma conducendo comunque a qualche modello interessante che riesce a suddividere in modo abbastanza efficace i diversi calciatori.

Inoltre, un'analisi di questo tipo offre spunti interessanti per ulteriori ricerche future, sia riguardo ai portieri (ad esempio individuando tipologie diverse di portiere, come quello che aiuta in fase di primo possesso con "piedi educati" rispetto a quello che fatica ad impostare dal basso), sia scegliendo di dividere i giocatori in singoli ruoli e quindi trovando, nella stessa zona di campo, giocatori con attitudini e caratteristiche molto diverse (la punta bassa, veloce e tecnica rispetto al centravanti d'area alto e potente; il difensore centrale alto, lento e bravo nelle letture contrapposto al difensore più basso, ma veloce e aggressivo; il centrocampista di regia, tecnico e poco dinamico in confronto alla mezzala "box to box"…).

Comunque, è possibile utilizzare diversi dei metodi di clusterizzazione sperimentati per trovare figure interessanti sul calciomercato, sia ricercando in base a determinate caratteristiche dei singoli calciatori, sia ritrovando giocatori molto simili a qualche figura "inarrivabile" per questioni di costo elevato, disinteresse del giocatore, di nuovo contratto a lungo termine o di grave infortunio.

6. Bibliografia

[1] Football Manager, Wikipedia: https://it.wikipedia.org/wiki/Football_Manager_(serie)
[2] Football Manager: https://www.footballmanager.com/it
[3] Kaggle, Football Manager 2020 Dataset: https://www.kaggle.com/ktyptorio/football-manager-2020
[4] Knime: https://www.knime.com/
[5] R: https://www.r-project.org/
[6] clValid package: https://cran.r-project.org/web/packages/clv/clv.pdf
[7] Appunti corso di Data Mining e Statistica Computazionale, Pier Giorgio Lovaglio, corso di laurea triennale SGI, 2021