

# 1 Analisi dei cluster

L'analisi dei clusters è stata svolta normalizzando i valori degli attributi tra 0 e 1, con lo scopo di standardizzare il contributo del valore degli attributi nel calcolo della distanza.

## 1.1 clustering via K-means

### 1.1.1 Scelta degli attributi e della funzione distanza

Si sono scelti gli attributi quantitativi (*satisfaction\_level*, *last\_evaluation*, *average\_monthly\_hours*, *time\_spend\_company*) e l'attributo numerico ordinale *number\_project*. La scelta degli attributi quantitativi è giustificata dal fatto che l'algoritmo K-means richiede di calcolare la media, la quale è definita solo per gli attributi quantitativi. L'unica eccezione è stata fatta per l'attributo *number\_project* essendo ordinale. L'interpretazione è stata che se *number\_project* ha valore frazionario *d.f* (con *d* parte intera, *f* parte float) per un dato centroide, allora il cluster da lui rappresentato contiene gli impiegati che in media hanno fatto tra *d* e *d + 1* progetti.

La funzione distanza scelta è stata la distanza Euclidea perché i centroidi sono medie nello spazio Euclideo in  $\mathbb{R}^n$ , dove *n* è il numero di attributi. Inoltre il valore della distanza tra 2 punti è di più facile interpretazione rispetto ad altre metriche.

### 1.1.2 Identificazione del miglior valore di k

Identificare il numero di clusters è importante per trovare un compromesso tra pochi grandi clusters e molti piccoli, e spesso insignificanti, clusters. Per indentificare il miglior valore di *k* per l'algoritmo K-means si è monitorato l'andamento della *Sum of Squared Error (SSE)* e la *silhouette score* al variare di *k*, il cui grafico è riportato in Figura 1. Il grafico della *SSE* ha un andamento non-crescente e diminuisce in maniera smooth, mentre la silhouette presenta molti picchi.

Successivamente si è scelto il punto di gomito della *SSE* combinando un approccio visivo ad uno quantitativo, analizzando i punti *k* in cui era presente un maggior calo di *SSE*. La scelta finale per il valore



Figura 1: Andamento della SSE e silhouette al variare di *k*. Il punto di gomito è scelto per *k* = 8, cui corrisponde *SSE* = 1474 e *silhouette* = 0.27.

di *k* ha preso in considerazione anche il corrispondente valore della *silhouette*. La *silhouette* assumeva valori nel range [0.18, 0.28] con minimo per *k* = 47 e massimo per *k* = 3, mentre la *SSE* assumeva valori tra [640, 3315] con minimo per *k* = 50 e massimo per *k* = 2. Inoltre, sono stati presi in considerazione i top 10 valori di *k* corrispondenti a un maggiore calo della *SSE* (*top\_diffs*). Analogamente sono stati presi i top 10 *k* con maggiore *silhouette* (*top\_silho*). Si sono ottenuti i seguenti insiemi di valori di *k*, candidati ad essere punti di gomito: *top\_diffs* = {3, 4, 5, 6, 7, 8, 9, 10, 11, 12} and *top\_silho* = {3, 8, 14, 7, 6, 10, 5, 9, 4, 18}. Mettendo insieme tutte le precedenti informazioni e considerando l'insieme *top\_diffs* ∩ *top\_silho*, si è scelto il punto di gomito *k* = 8, corrispondente a *SSE* = 1474 and *silhouette* = 0.274.

Alternativamente, scegliendo il punto del grafico della *SSE* più vicino all'origine in norma Euclidea, si era ottenuto il punto *k* = 12, con *SSE* = 1210 e *silhouette* = 0.257.

### 1.1.3 Caratterizzazione dei clusters ottenuti

La caratterizzazione dei cluster ottenuti è stata svolta per mezzo dell'analisi dei centroidi, e confrontando le distribuzioni degli attributi dei singoli cluster con quelle dell'intero dataset.

La Figura 2 riporta l'analisi per centroidi. Gli attributi dei centroidi dei clusters 1 e 3 hanno relativamente lo stesso andamento. In particolare i valori di *average\_monthly\_hours* differiscono di poco. Questo significa che per valori più piccoli di  $k$  i due clusters potrebbero unirsi. I clusters 4, 5 e 7 sono caratterizzati da un basso valore di *satisfaction\_level*, mentre lo stesso attributo assume valori elevati nei clusters 0, 1, 3 e 6. Il cluster 6 è l'unico con un alto valore di *time\_spend\_company*, mentre per il resto dei clusters l'attributo assume valori bassi. Questo ci dice che i due attributi non sono correlati, altrimenti anche i clusters 0, 1 e 3 avrebbero avuto un alto valore di *time\_spend\_company*. Il cluster 4 ha il più basso valore di *satisfaction\_level* e il più alto valore di *average\_monthly\_hours*. Inoltre, il cluster 4 ha il più alto valore di *number\_projects* e un relativamente basso valore di *time\_spend\_company*. Questo significa che il sovraccarico di lavoro dovuto ad una grande quantità di progetti, svolti in poco tempo, porta gli impiegati ad essere infelici.

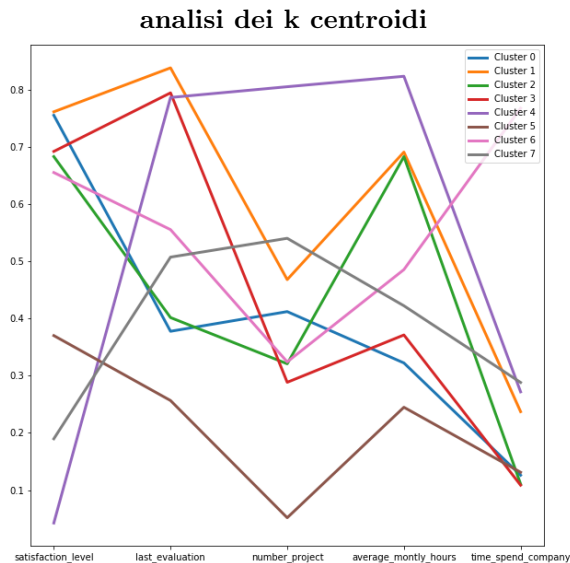


Figura 2: Analisi dei valori degli attributi dei centroidi ottenuti. Ogni centroide è rappresentato da un insieme di segmenti, i cui estremi marcano i valori degli attributi. Gli attributi dei clusters 1 e 3 seguono lo stesso andamento.

In Figura 3 viene fatto un confronto tra le distribuzioni degli attributi dell'intero dataset e quelle dei singoli clusters, in particolare i cluster 4 e 7. L'attributo *number\_project* assume una distribuzione multi-modale per tutte le kernel density estimation, essendo un attributo intero. Gli attributi *last\_evaluation* e *average\_monthly\_hours* hanno una distribuzione simile alla normale, ma leggermente schiacciata. Gli stessi attributi assumono una simile forma per il cluster 4. Invece per l'intero dataset i due attributi hanno una distribuzione bi-modale con picchi poco pronunciati, divisi da un lungo plateau. In entrambe le figure 3b e 3c è presente un picco molto acuto per l'attributo *satisfaction\_level*. Entrambi i picchi sono quasi simmetrici, se non per lo scalino nella parte finale a destra del punto medio. Questo comportamento non è ripetuto per l'intero dataset. L'attributo *time\_spend\_company* assume una distribuzione multi-modale per il cluster 7, con 5 picchi nell'intervallo  $[0, 0.5]$ . Mentre per il cluster 4, *time\_spend\_company* ha meno variabilità riportando un picco relativamente acuto centrato in 0.25, e altri due picchi meno pronunciati.

#### 1.1.4 Visualizzazione del clustering via Principal Component Analysis

I risultati ottenuti dal clustering via K-means sono stati combinati con la PCA per visualizzare la proiezione di ogni punto di dati in due dimensioni. Il dataset ha una forma globulare allungata, simile ad un ellisse. I punti di dati sono molto vicini gli uni agli altri formando un dataset molto denso, in cui i cluster 4 e 5 sono molto definiti, mentre i punti degli altri clusters tendono a mischiarsi. In basso a sinistra è presente una zona di bassa densità di punti appartenenti al cluster 0, mentre in alto a destra vi è un chiaro outlier assegnato al cluster 7. La natura dell'outlier in termini di attributi non è stata approfondita.

## 1.2 Clustering via DBSCAN

L'analisi dei cluster con il metodo DBSCAN è stata svolta per identificare clusters con forma arbitraria. Con lo scopo di confrontare i risultati ottenuti con K-means, sono stati scelti gli attributi quantitativi e ordinali, e la distanza euclidea per funzione distanza.

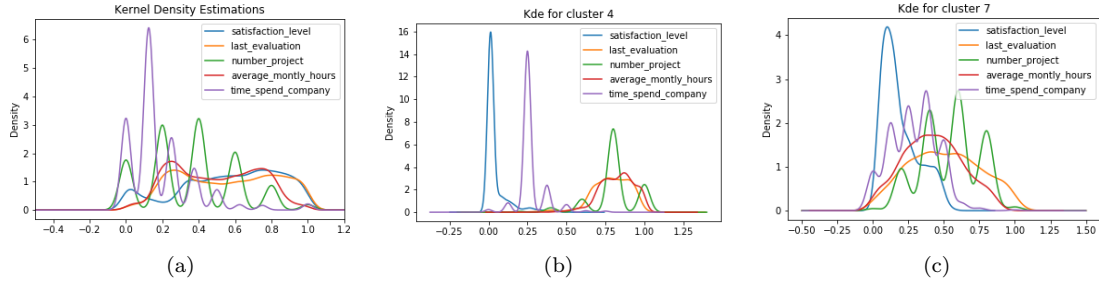


Figura 3: Kernel density estimation degli attributi (a) per l'intero dataset, (b) per il cluster 4, e (c) per il cluster 7. Gli acuti picchi dell'attributo *satisfaction\_level* per i cluster 4 e 7 non sono così pronunciati anche nell'intero dataset.

### 1.2.1 Studio dei parametri *minPoints* ed *epsilon*

La scelta dei parametri *minPoints* ed  $\epsilon$  è stata ispirata dal metodo proposto da Ester et al<sup>1</sup>. Figura 5 riporta il grafico delle distanze di ogni punto dati al  $k$ -esimo punto dati più vicino, ordinate in modo decrescente.<sup>2</sup> I grafici, per valori diversi di  $k$ , avevano la stessa forma, e i valori delle ascisse erano uguali per almeno 7 punti decimali. Inoltre, dal momento che i valori di *minPoints* ed  $\epsilon$  giocano sullo stesso compromesso, l'analisi è stata portata avanti fissando  $minPoints = k + 1$ , escludendo il punto dati stesso dal calcolo dei punti vicini.

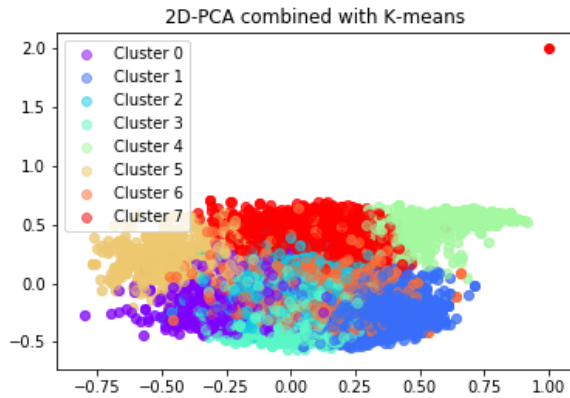


Figura 4: Visualizzazione del clustering in 2D. Ad ogni punto di dati è associata una label. Ogni label è rappresentata con un colore. Dataset denso a forma di ellisse. Un outlier è chiaramente presente in alto a destra.

Per la scelta di  $\epsilon$ , l'idea era quella di trovare il punto di gomito in Figura 5. Dalla figura si vede chiaramente che il punto di gomito giace nell'intervallo di valori  $\epsilon \in [0.1, 0.3]$ . Usando un metodo analogo utilizzato in sottosezione 1.1.2, si è ottenuto come risultato  $\epsilon = 0.297$  e  $silhouette = 0.301$ . Dal momento che il valore  $\epsilon$  ottenuto era ai margini dell'intervallo di valori predetto visualmente, è stata svolta un'indagine più approfondita per scoprire come variava il valore della *silhouette* al variare di  $\epsilon \in [0.1, 0.3]$ . L'intervallo è stato diviso arbitrariamente in 30 punti. L'andamento della *silhouette* è mostrato in Figura 6. Il grafico assume un plateau a partire da  $\epsilon \cong 0.26$ , in cui la *silhouette* rimane qua-

<sup>1</sup>Ester, Martin, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd. Vol. 96. No. 34. 1996.

<sup>2</sup>Per visibilità sono riportate soltanto le curve per  $k = 4$  e  $k = 8$ . Le curve per valori intermedi di  $k$  giacciono tra le due curve mostrate in figura.

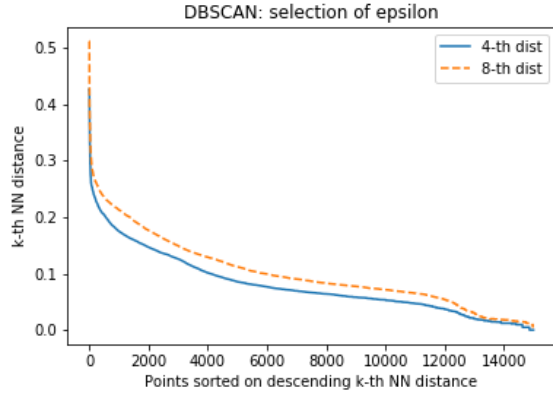


Figura 5: Distanze tra ogni punto dati e il suo  $k$ -th punto più vicino ordinate in maniera non-crescente.

si costante. Selezionando  $\epsilon = 0.26$  si è ottenuto un insieme di due clusters rispettivamente di 14958 e 6 elementi, con valore di  $silhouette = 0.344$ . I rimanenti 35 punti dati sono stati classificati come noise-points.

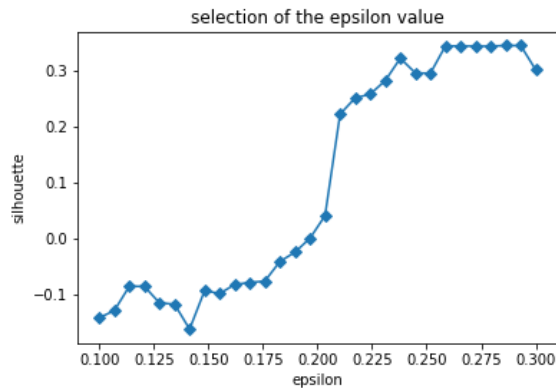


Figura 6: Andamento della silhouette per epsilon che varia nell'intervallo  $[0.1, 0.3]$ . Prima di  $\epsilon = 0.2$  la silhouette è negativa, mentre è presente un grande incremento per  $\epsilon \cong 0.225$ . Dopo  $\epsilon \cong 0.26$  è presente un plateau. Si è scelto  $\epsilon = 0.26$ , ottenendo  $silhouette = 0.344$ .

### 1.2.2 Caratterizzazione ed interpretazione dei clusters

I clustering ottenuti per diversi valori di  $\epsilon$  sono stati caratterizzati analizzando la curva in Figura 6. Nell'intervallo di valori  $epsilon \in [0.1, 0.2]$  (primi

15 punti sulla curva) il numero di clusters formati varia da 105 fino a 12, in cui sono presenti circa 6 clusters di medie dimensioni, e il resto di dimensioni molto piccole. I noise-points variano da 3278 fino a 281. In questo intervallo si ha  $silhouette \leq 0$ , il che significa che mediamente i punti di un cluster sono più vicini ai punti di un altro cluster, piuttosto che ai punti del cluster di appartenenza, o che sono presenti troppi clusters. Un'ulteriore interpretazione è che alcuni clusters si sovrappongono. Nella seconda metà del grafico, per  $\epsilon \in (0.2, 0.3]$  si inizia a formare un unico grande cluster contenente almeno il 97% dei dati, con il rimanente 3% diviso tra pochi piccoli clusters e noise-points. In particolare, nell'intervallo  $[0.26, 0.3)$  la  $silhouette$  forma un plateau, e sono presenti solo 2 clusters. Investigando i clusters ottenuti nei punti formanti il plateau si è scoperto che i noise-points gradualmente si uniscono ad uno dei 2 clusters. Infine, per  $\epsilon = 0.3$  alcuni noise-points si uniscono per formare un piccolo cluster, diminuendo la  $silhouette$ .

## 1.3 Clustering Gerarchico

L'analisi dei cluster con approccio gerarchico è stata fatta con i metodi complete, average e di Ward, perché sono meno suscettibili al rumore dei dati e a gli outliers. Sono stati scelti gli attributi quantitativi e ordinali, e la distanza Euclidea, come in K-means e DBSCAN.<sup>3</sup>

**Complete linkage.** Figura 7 mostra il dendrogramma ottenuto con il metodo complete linkage. A distanza minore di 25 si otterrebbero 16 clusters. I primi due clusters nella parte alta della figura (rispettivamente di 3300 e 4617 elementi) si uniscono con incremento di distanza relativamente alto, la quale diventa 1.6. Successive unificazioni di clusters incrementano lentamente la distanza fino a 1.91. Il dendrogramma suggerisce la presenza di 2 clusters ottenuti con una linea di taglio a distanza 1.85.

**Average Linkage.** Figura 8 mostra il dendrogramma ottenuto con il metodo average linkage. Nella parte bassa della figura, a distanza 1.04, un singleton (con label 7492) si unisce ad un grande

<sup>3</sup>Sono visualizzati soltanto i primi 3 livelli dei dendrogrammi per facilitarne la comprensione.

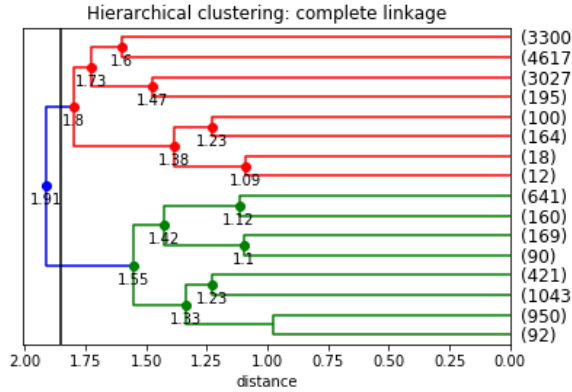


Figura 7: Dendrogramma per il metodo complete linkage. La riga verticale nera rappresenta il taglio del dendrogramma a distanza 1.85 risultante in 2 clusters.

agglomerato di clusters di 14966 elementi. Questa unificazione avviene relativamente tardi, il che suggerisce che il singleton sia un outlier. A metà figura due coppie di singleton si uniscono con un piccolo agglomerato di clusters, facendo un incremento di distanza relativamente alto, che arriva a 1.05. Dato questo grande salto in distanza e le dimensioni ridotte dei clusters coinvolti nell'unificazione, si è deciso di tagliare il dendrogramma a distanza 1, ottenendo 4 clusters.

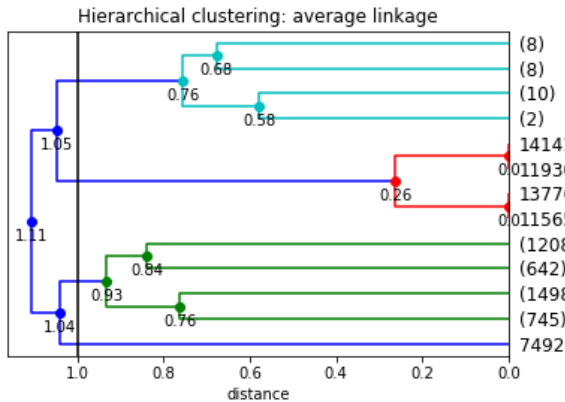


Figura 8: Dendrogramma per il metodo average linkage. La riga che taglia il dendrogramma a distanza 1 risulta in 4 clusters.

**Ward's method Linkage.** Figura 9 mostra il dendrogramma ottenuto con il metodo di Ward. Il clustering ottenuto con questo metodo è il più equilibrato, in quanto nei livelli mostrati in figura, non sono presenti clusters di dimensioni molto piccole. Tuttavia la distanza assume valori molto più elevati rispetto ai precedenti metodi. Per una distanza minore di 15 si hanno unificazioni con incremento graduale della distanza, mentre negli ultimi 6 collegamenti la distanza fa lunghi salti raggiungendo 41.61. Per questo motivo si è deciso di tagliare il dendrogramma a distanza 15 ottenendo 7 clusters.

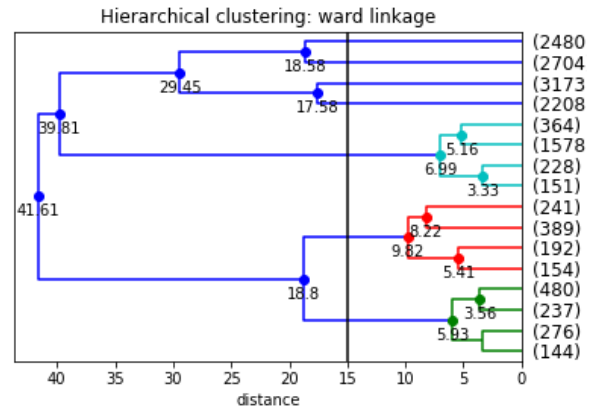


Figura 9: Dendrogramma per il metodo Ward. Il taglio a distanza 15 fa ottenere 7 clusters.

**Calcolo della silhouette.** Il calcolo della silhouette è stato fatto considerando un algoritmo di clustering non strutturato, e uno di tipo strutturato considerando  $n = 100$  punti vicini<sup>4</sup>. La silhouette è stata calcolata per i tre metodi impostando un numero di clusters da 2 a 9. Il range di valori assunti dalla silhouette è mostrato in Tabella 1, in cui si vede che i migliori risultati si ottengono per  $C=2$  clusters e  $n=100$  per i metodi complete e average, e con algoritmo non strutturato per il metodo Ward. In particolare, con i primi due metodi si ottengono risultati identici in termini di silhouette (solo per  $C=2$ ). Anche i clusters ottenuti hanno le stesse dimensioni (solo per  $C=2$ ) di 14995 e 4 elementi<sup>5</sup>.

<sup>4</sup>Dato il costo computazionale, valori maggiori di  $n$  non sono stati considerati.

<sup>5</sup>Non è stato controllato che anche i singoli elementi fossero assegnati a gli stessi clusters.

Tabella 1: Intervallo di valori assunti dalla silhouette, per un numero di cluster C da 2 a 9, per i metodi complete, average e Ward. Risultati ottenuti con un algoritmo non strutturato (U) e strutturato considerando n=100 vicini. Per ogni metodo il miglior risultato è mostrato in grassetto. I numeri sottolineati sono risultati altrettanto buoni.

Metodo	Silhouette C=2 - U	Silhouette C=9 - U	Silhouette C=2 - n=100	Silhouette C=9 - n=100
Complete	0.197	0.143	<b>0.348</b>	- 0.061
Average	<u>0.339</u>	0.209	<b>0.348</b>	0.149
Ward	<b>0.306</b>	0.215	<u>0.295</u>	0.202

Tabella 2: Confronto delle prestazioni degli algoritmi di clustering K-means, DBSCAN e gerarchico. Risultati ottenuti impostando i seguenti parametri: K=8 per K-means. MinPoints=4,  $\epsilon = 0.26$  per DBSCAN ottenendo 2 clusters. Metodo average linkage con n=100 vicini per clustering gerarchico, ottenendo 2 clusters. In grassetto il miglior risultato.

Algoritmo	Silhouette
K-means	0.274
DBSCAN	0.344
Gerarchico	<b>0.348</b>

## 1.4 Valutazione del miglior metodo di clustering

Tabella 2 riassume le prestazioni degli algoritmi di clustering usati. L'algoritmo che ha dato risultati migliori, in termini di silhouette, è il clustering gerarchico usando il metodo average linkage<sup>6</sup>, il quale rileva che nel dataset sono presenti 2 clusters, rispettivamente di 14955 e 4 elementi. DBSCAN ha avuto risultati simili, sia in termini di silhouette che per i clusters rilevati, ottenendo 2 clusters di dimensioni 14958 e 6 elementi, e 35 noise-points. Mentre l'algoritmo con prestazioni minori è K-means, il quale rileva 7 clusters di dimensioni quasi omogenee.

<sup>6</sup>A pari merito con il metodo complete linkage.