Nicholas Montenegro - Data Science and Business
Francesca Sbolgi - Computer Science
Elisabetta Triolo - Digital Humanities

Data Mining Project on

Human Resources Analytics

Why are our best and most experienced employees leaving prematurely?

aa 2017/2018

Indice

Introduzione

Questo progetto consiste nell'analisi del database Human Resources (HR), scaricabile dalla piattaforma Kaggle e contenente informazione sugli impiegati di una compagnia, allo scopo di predire quali impiegati lasceranno l'azienda prematuramente e di comprendere quali sono i principali motivi del loro licenziamento.
La realizzazione di un modello accurato che permetta di prevedere se un impiegato lascerà o no il proprio lavoro costituisce un vantaggio per l'azienda, poiché le permette di individuare le caratteristiche principali degli impiegati che lasciano il posto di lavoro e, soprattutto in caso di dipendenti qualificati e con esperienza, di servirsene per evitarne la perdita, attuando dei provvedimenti interni indirizzati proprio verso la categoria di impiegati interessati. Ad esempio, se dal modello si evidenzia che a lasciare il posto di lavoro prematuramente sono gli impiegati con un salario più basso, l'azienda potrà, se lo ritiene opportuno, intervenire in maniera preventiva proponendo degli aumenti di stipendio agli impiegati interessati.
L'analisi dei dati è suddivisa in quattro fasi principali:

1. Data Understanding: è formato da due fasi, la comprensione dei dati tramite statistiche, valutazione della qualità e misura della correlazione e la preparazione dei dati per le analisi successive (gestione di eventuali missing values e outliers, normalizzazione, eliminazione di variabili ridondanti);
2. Clustering: consiste nell'applicazione degli algoritmi di clustering K-means, DBSCAN e Agglomerative Hierarchical, al fine di trovare raggruppamenti ottimali dei dati;
3. Association Rules Mining: ha il compito di individuare i pattern più frequenti e valutare quali siano i più interessanti;
4. Classification: consiste nello sviluppo di un modello di predizione accurato e affidabile.

1 Data Understanding

Per poter essere trattati dalle analisi successive, i dati sono stati in primo luogo sottoposti ad un processo di comprensione e preparazione. In particolare, infatti, si è fissato il significato di ogni attributo, sono state effettuate analisi statistiche che mettessero in evidenza la distribuzione e l'eventuale presenza di outliers e missing values, ed è stata misurata la correlazione tra gli attributi numerici. I dati, inoltre, sono stati preparati per le analisi successive attraverso processi di normalizzazione, discretizzazione e sostituzione. I principali metodi adottati sono descritti in questo capitolo.

1.1 Descrizione del database

Lo Human Resources (HR) è un database in formato csv che descrive gli impiegati di una compagnia in base ad alcune loro caratteristiche. Il database contiene 14999 righe, ognuna delle quali rappresenta un impiegato della compagnia, e 10 colonne che descrivono ciascuno di essi.
Nella seguente tabella per ogni colonna del database è riportata una breve descrizione per spiegarne il significato.

Nome	Descrizione
satisfaction_level	livello di soddisfazione dell'impiegato (a valori prossimi allo 0 corrisponde una minore soddisfazione, a valori prossimi a 1 una maggiore)
last_evaluation	più recente punteggio di valutazione dell'impiegato (a valori prossimi allo 0 corrisponde una minore valutazione, a valori prossimi a 1 una maggiore)
number_project	numero di progetti completati dall'impiegato nel periodo di lavoro
average_montly_hours	numero medio mensile di ore di lavoro dell'impiegato
time_spent_company	numero di anni di lavoro presso la compagnia
Work_accident	indica se l'impiegato ha avuto (1) o no (0) incidenti sul lavoro
left	indica se l'impiegato ha lasciato (1) o no (0) la compagnia
promotion_last_5years	indica se l'impiegato ha avuto (1) o no (0) una promozione negli ultimi cinque anni
sales	dipartimento della compagnia presso il quale l'impiegato lavora o ha lavorato
salary	livello di retribuzione dell'impiegato rispetto alle retribuzioni della compagnia

Tabella 1: Descrizione degli attributi presenti nel dataset.

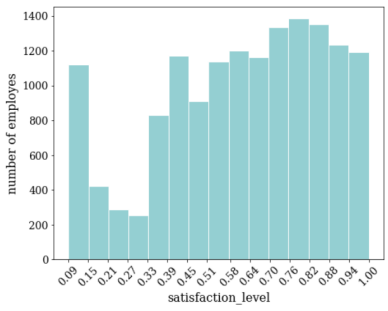
Al fine di una maggiore chiarezza nella lettura dei dati abbiamo ritenuto opportuno rinominare alcuni attributi. Nello specifico, per omogeneità e convenzione è stato modificato *Work_accident* in *work_accident*. Le proprietà *average_montly_hours* e *time_spend_company* presentavano errori grammaticali e sono stati modificati rispettivamente in *average_monthly_hours* e *time_spent_company*.
Alla proprietà *sales* era stato originariamente dato il nome di un valore contenuto al suo interno, quindi, per chiarezza (e correttezza) si è deciso di rinominarla *departments*.
Su Kaggle, da cui il dataset è stato estratto, l'attributo *last_evaluation* è descritto come "Time since last performance evaluation (in Years)".
Nelle discussioni relative al dataset presenti sempre su Kaggle, tuttavia, è possibile notare che molti interpretano l'attributo come riferito al punteggio ottenuto dall'impiegato nell'ultima valutazione fatta dalla compagnia. [1]
A questo punto, considerando anche il fatto che l'attributo ha valori continui, i casi potrebbero essere tre:

1. esso si riferisce al numero di anni dall'ultima valutazione del *satisfation_level*, attributo immediatamente precedente;
2. esso si riferisce al numero di anni trascorsi dall'ultima valutazione dell'impiegato da parte della compagnia;
3. esso si riferisce, come suggerito dalle discussioni, al punteggio ottenuto dall'impiegato nell'ultima valutazione fatta dalla compagnia e la descrizione dell'attributo fatta su Kaggle è errata.

Nella nostra analisi, abbiamo deciso di escludere il primo caso, visto che la tabella è riferita agli impiegati e quindi l'attributo dovrebbe indicare una loro diretta caratteristica e se fosse riferito a *satisfation_level* sarebbe inserito probabilmente in un'altra tabella dedicata alle valutazioni sul livello di soddisfazione.
Abbiamo inoltre escluso anche il secondo caso, sulla base del fatto che l'indicazione del numero di anni trascorsi dall'ultima valutazione sarebbe meno significativa in quanto priva dell'indicazione del punteggio conseguito nella valutazione stessa.
Per questi motivi, nella nostra analisi abbiamo scelto il terzo caso, considerando i valori di *last_evaluation* come riferiti al punteggio ottenuto dall'impiegato nell'ultima valutazione fatta dalla compagnia.

1.2 Distribuzione delle variabili

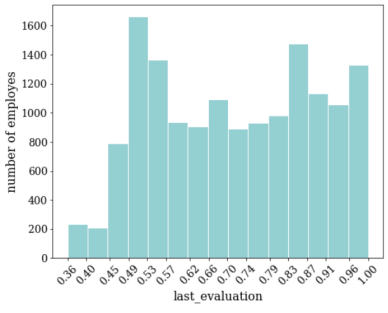
Di seguito sono state riportate per ogni variabile la distribuzione, qualche statistica che ci permetta di comprendere meglio il grafico e il bilanciamento dei valori.
Le variabili continue sono state discretizzate utilizzando la regola di Sturges in 15 bins. Infatti, sebbene le diverse distribuzioni non siano propriamente gaussiane possono essere approssimate come tali. Sono stati comunque sperimentati valori diversi da quello suggerito dalla formula ma non sono stati osservati cambiamenti rilevanti.



satisfaction_level

Type: Numerical Continuous
Range: [0.09, 1]
Mean: 0.613
Std: 0.249

La distribuzione è vicina ad una gaussiana sebbene sia leggermente sbilanciata per livelli di soddisfazione alti, come si nota anche dal valore della media, e abbia un picco iniziale. La variabile è stata discretizzata in 15 bins, calcolati tramite la regola di Sturges.

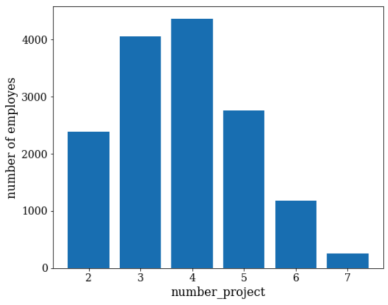


last_evaluation

Type: Numerical Continuous
Range: [0.36, 1]
Mean: 0.716
Std: 0.171

Anche in questo caso possiamo assumere una distribuzione gaussiana ed utilizzare la regola di Sturges per suddividere la variabile in 15 bins. La distribuzione è piuttosto uniforme, sebbene vi siano un numero ristretto di impiegati che hanno ricevuto una valutazione molto bassa.

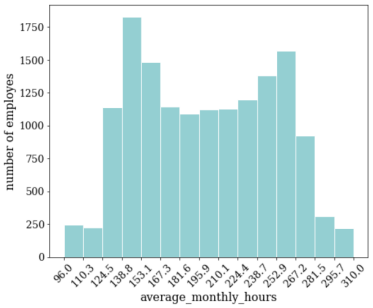
[1] ad esempio in questa discussione: `https://www.kaggle.com/ludobenistant/hr-analytics/discussion/39300`



number_project

Type: Numerical Discrete
Range: [2, 7]
Mean: 3.803
Std: 1.233

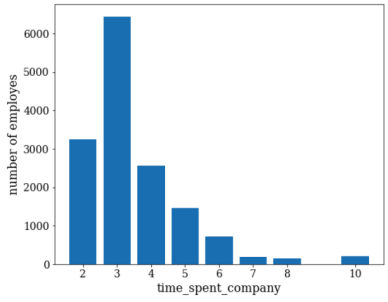
Il valore più frequente è 4, quello meno frequente è 7. Possiamo quindi notare che la maggior parte degli impiegati ha portato a termine dai 3 ai 5 progetti, mentre non vi sono impiegati che hanno svolti 0 o 1 progetto oppure più di 7.



average_monthly_hours

Type: Numerical Discrete
Range: [96, 310]
Mean: 201.50
Std: 49.94

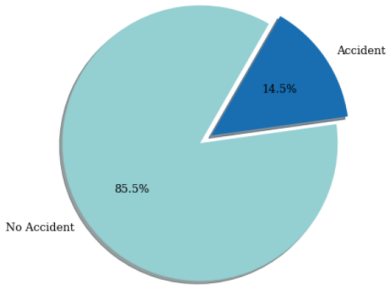
La distribuzione di questa variabile è molto ben bilanciata in quanto è abbastanza simmetrica rispetto alla media. Poiché presentava troppi valori distinti si è deciso di discretezzare anche questa in 15 bins.



time_spent_company

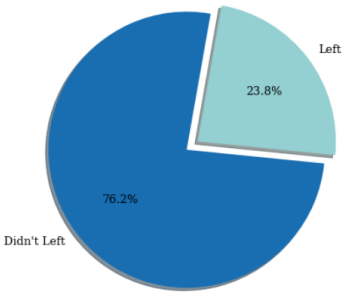
Type: Numerical Discrete
Range: [2,10]
Mean: 3.498
Std: 1.460

time_spent_company è una variabile fortemente sbilanciata che mostra la propensione dei dipendenti a rimanere nella compagnia per un periodo che va dai 2 ai 4 anni formando l'81.64% del totale; coloro che invece rimangono dai 5 ai 7 anni sono il 15.86%; il restante, che va da 8 a 10 anni è il 2,51%.



work_accident
Type: Numerical Binary
Values: {0,1}
Mean: 0.145
Std: 0.351

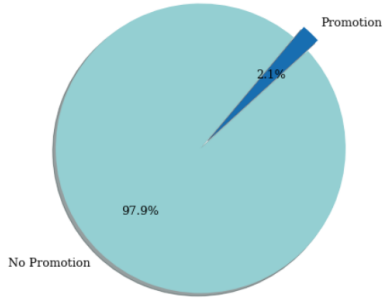
La percentuale di incidenti sul lavoro è più alta di quanto si potrebbe pensare: il 14.5% degli impiegati hanno avuto almeno un incidente. In ogni caso però l'attributo *work_accidents* risulta sbilanciato verso coloro che non hanno avuto incidenti con l'87.5% (12830 istanze) contro il 14.5% del restante (2169 istanze) .



left

Type: Numerical Binary
Values: {0, 1}
Mean: 0.238
Std: 0.426

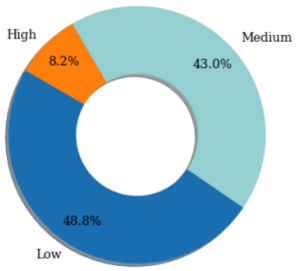
left sarà la nostra variabile di riferimento nel corso delle prossime analisi. La distribuzione non è equilibrata in quanto il 76.2% (11428 istanze) delle persone ha deciso di restare nella compagnia a fronte del 23.8% (3571 istanze) che ha invece deciso di lasciare.



promotion_last_5years

Type: Numerical Binary
Values: {0, 1}
Mean: 0.021
Std: 0.144

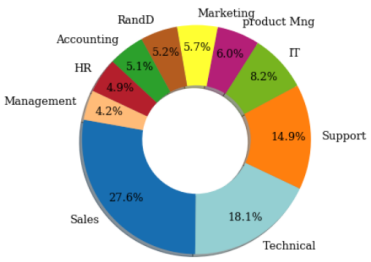
La variabile *promotion_last_5years* è senza alcun dubbio la proprietà più sbilanciata con il 97.9% (14680 istanze) di personale che non ha avuto una promozione negli ultimi 5 anni e con il restante 2.1% (319 istanze) che è stato invece promosso.



salary

Type: Categorical
Values: {Low, Medium, High}

La variabile *salary* può assumere 3 valori che indicano la fascia di stipendio che gli impiegati percepiscono. La gran parte riceve uno stipendio medio-basso. In particolare il valore low racchiude il 48.8% della popolazione, mentre medium il 43.0%. Solo l'8.2% degli impiegati ha uno stipendio che è considerato alto.



departments

Type: Categorical
Values: {sales, accounting, hr, technical, support, management, IT, product_mng, marketing, RandD}

I dipartimenti con maggior numero di impiegati sono in ordine Sales, Technical, Support e IT. Gli altri sono approssimativamente distribuiti in modo uniforme. Il valore RandD indica il reparto di Research and Development.

1.3 Missing values e outliers

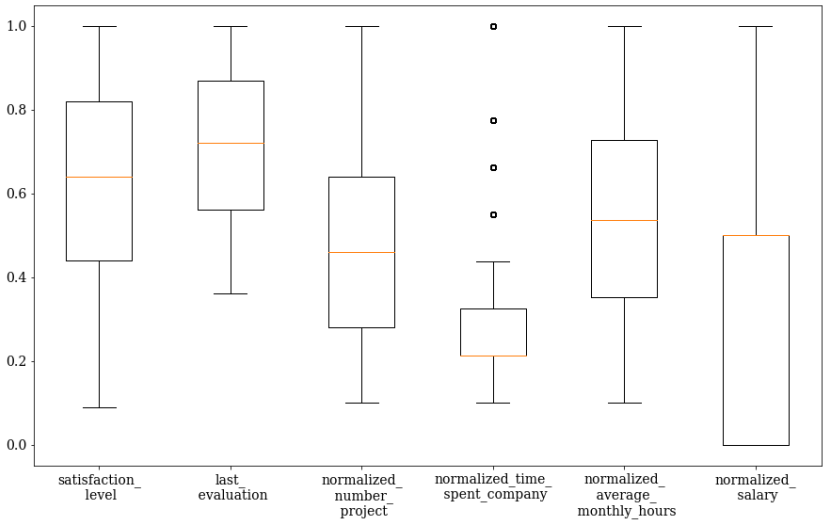


Figura 1: Box Plots.

Nel dataset non sono presenti missing values. Non è stata infatti rilevata la presenza esplicita di valori mancanti, né sono stati individuati valori di default che possano nascondere missing values.
Come mostrato dal grafico, è possibile individuare la presenza di alcuni outliers relativamente all'attributo *time_spent_company*, che indica, in anni, da quanto tempo un impiegato lavora presso la compagnia. Gli outliers coincidono con i valori 6, 7, 8 e 10 che si allontanano significativamente dal resto dei valori.
Non è pensabile che tali valori costituiscano dati errati, ma si tratta semplicemente di dati che rispecchiano la situazione aziendale. Infatti è del tutto logico supporre che in un'azienda ci siano persone che lavorano da più tempo di altre, anche se in numero molto minore.
I record contenenti outliers sono in totale 1282. Si tratta circa dell'11,7% dei record, che non sono pochi, ed inoltre, essi non incidono quasi per nulla sulla media e sulla standard deviation degli altri attributi. Per questi motivi, si è deciso di mantenerli all'interno del dataset.

1.4 Discretizzazione e normalizzazione

Sono state applicate alle variabili del dataset alcune trasformazioni per poter processare e leggere più facilmente i dati.
Per poter applicare alcuni algoritmi che richiedono in input dati numerici, i valori degli attributi *departments* e *salary* sono stati trasformati da stringa a numero. In particolare i dieci valori distinti di *departments* (IT, RandD, accounting, hr, management, marketing, product_mng, sales, support, technical) sono stati mappati rispettivamente con un valore numerico da 0 a 9, attraverso la creazione della nuova variabile *departments_val*.
I tre valori di *salary*, low, medium e high, sono stati mappati rispettivamente nei valori numerici 0, 1 e 2 all'interno della nuova variabile *salary_val*.
Per migliorare la leggibilità dei dati e ridurne la sparsità, è stata effettuata una discretizzazione delle variabili numeriche *satisfaction_level*, *last_evaluation* e *average_monthly_hours*. Si è eseguito un natural binning

per ottenere 15 bins con la stessa ampiezza. Il numero k di bins è stato scelto usando la formula di Sturges $k = \lceil 1 + \frac{10}{3} log_{10}(N) \rceil$ in cui N è il numero totale dei valori dell'attributo (ovvero 14999).
In alcuni casi, per permettere un confronto più accurato tra variabili che hanno una scala diversa tra loro, sono stati normalizzati gli attributi *number_project*, *time_spent_company* e *average_monthly_hours* con una normalizzazione di tipo min-max in modo da trasformare i valori in una scala da 0 a 1, creando le nuove variabili *normalized_number_project*, *normalized_time_spent_company*, *normalized_average_monthly_hours*.
Al fine di comprendere meglio il quantitativo medio di lavoro degli impiegati la variabile *average_monthly_hours* è stata sostituita con *average_daily_hours* , ottenuta dalla precedente tramite la seguente operazione:

$$average\_daily\_hours = \frac{average\_monthly\_hours}{21.5}$$

in cui possiamo assumere che in media vi siano 21.5 giorni lavorativi al mese. Tramite questa trasformazione otteniamo che gli impiegati lavorano in media 9.35 ore al giorno con una standard deviation di 2.32 ore. In questo modo quindi è stato più semplice comprendere che i turni di lavoro sono piuttosto lunghi. Anche questa variabile, inoltre, è stata normalizzata con una normalizzazione di tipo min-max, ottenendo la nuova variabile *normalized_average_daily_hours*.

1.5 Correlazioni tra le variabili

Nel database non sono presenti correlazioni sufficientemente alte che potrebbero indicare la presenza di variabili ridondanti.
Utilizzando la formula di Pearson si può notare che c'è una leggera correlazione lineare positiva tra gli attributi *last_evaluation*, *number_project* e *average_daily_hours*. Questo indica che al crescere di una anche le altre due tendono ad aumentare. È invece curioso notare che il livello di soddisfazione non è correlato a nessuno degli altri parametri. Oppure che il numero di progetti non aumenta in relazione al numero di anni passati nella compagnia, come ci si potrebbe aspettare.
Sono stati testati anche altri tipi di indici, tra cui Spearman, per valutare se la correlazione tra le variabili non fosse lineare ma non hanno portato a risultati significativamente diversi.
Il discorso invece cambia se eseguiamo la stessa analisi separando gli impiegati che se ne sono andati da quelli che sono rimasti (vedi figura 3) . Infatti, mentre nel primo gruppo le correlazioni diventano più evidenti e ne appaiono anche di nuove, nel secondo si annullano completamente.

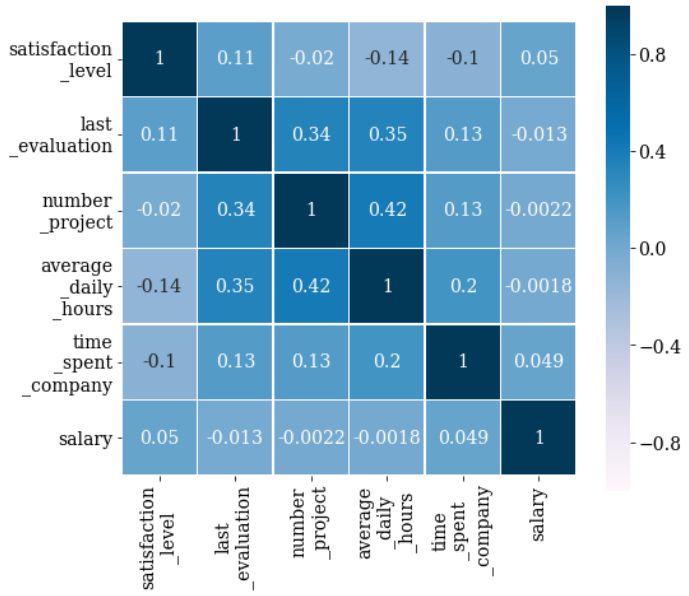
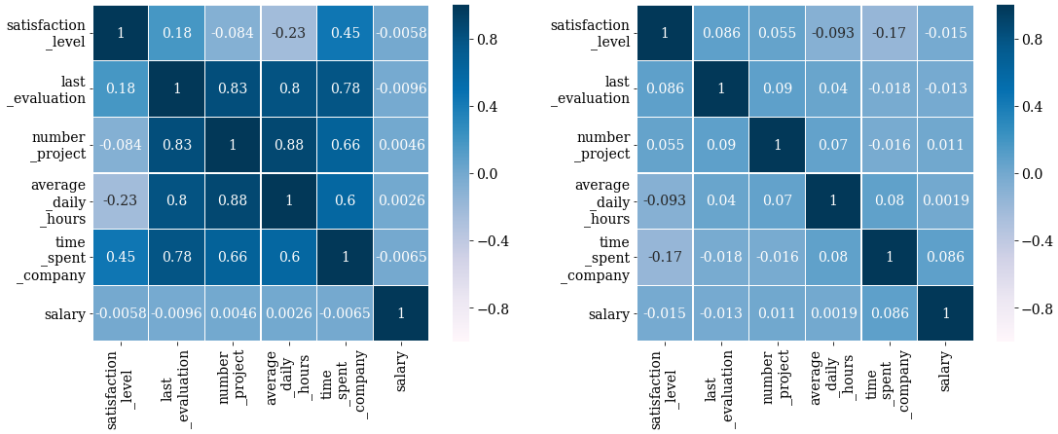


Figura 2: Heatmap of correlation matrix.

Studiando il database così suddiviso si comprende che la leggera correlazione tra *last_evaluation*, *number_project* e *average_monthly_hours* che si era notata in precedenza è in realtà formata da una forte correlazione in quelli che hanno lasciato e una nulla negli altri. Inoltre, tra i lavoratori che hanno lasciato, è presente anche una correlazione positiva tra *time_spent_company* e tutti gli altri attributi tranne salary. Questi schemi che si creano ci potrebbero aiutare anche in seguito per identificare quali impiegati se ne andranno. Infatti, se troviamo un record che presenta la stessa relazione tra le variabili che abbiamo appena descritto, potrebbe essere una prima indicazione che quella persona se ne potrebbe andare.
La popolazione di dipendenti ancora in azienda invece è più eterogenea, le variabili sono tutte indipendenti tra loro.



a) b)

Figura 3: Heatmap di chi ha lasciato (a) e di chi è rimasto (b).

2 Clustering

Al fine di trovare eventuali differenze o similitudini significative tra i record del dataset, dopo le operazioni di comprensione e preparazione dei dati, sono stati applicati su di essi tre algoritmi di clustering differenti, ovvero K-means, DBSCAN e Agglomerative Hierarchical.
In questo capitolo si descrivono i metodi usati e i risultati ottenuti.

2.1 Clustering con K-means

Il primo algoritmo applicato sui dati è il k-means. Esso permette di ottenere un center-based clustering, in cui cioè, ogni cluster è individuato da un punto centrale detto centroide e ogni altro punto è assegnato al cluster con il più vicino centroide. Tale algoritmo richiede che sia fissato in precedenza il numero k di clusters che si vogliono ottenere.
Per la misura della distanza tra i punti si è scelto di usare la distanza euclidea poiché è su di essa che si basa il Lloyd's algorithm, ovvero l'algoritmo standard originario da cui deriva il k-means a cui ancora oggi ci si può riferire con entrambi i termini. Per poter applicare l'algoritmo k-means è stato creato un dataset di training che contiene tutti i record e gli attributi del dataset originario, fatta eccezione per gli attributi categorici e binari e per l'attributo target left, anch'esso, tra l'altro, binario. Infatti, non è possibile ottenere una misura adeguata della distanza tra valori binari e/o categorici usando la distanza euclidea.
Gli attributi, quindi, sui quali è stato applicato l'algoritmo sono *satisfaction_level*, *last_evaluation*, *normalized_number_project*, *normalized_time_spent_company* e *normalized_average_daily_hours*.

2.1.1 Identificazione del miglior numero di centroidi

Il k-means richiede che il numero k di clusters che si vogliono ottenere sia fornito in precedenza, prima dell'esecuzione. Per scegliere tale valore è possibile utilizzare la misura del SSE (Sum of Squared Errors):

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x).$$

Essa consiste nel sommare per ogni cluster C_i, la distanza al quadrato di ogni punto x dal centroide m_i e permette di misurare la coesione interna di ogni cluster, ovvero di misurare quanto sono vicini tra loro i punti appartenenti allo stesso cluster. A un valore maggiore di SSE corrisponde una minore coesione interna dei clusters. L'algoritmo di k-means è stato iterato per un numero di k che va da 2 a 30 e per ogni iterazione è stato memorizzato il SSE. La seguente figura mostra i risultati ottenuti:

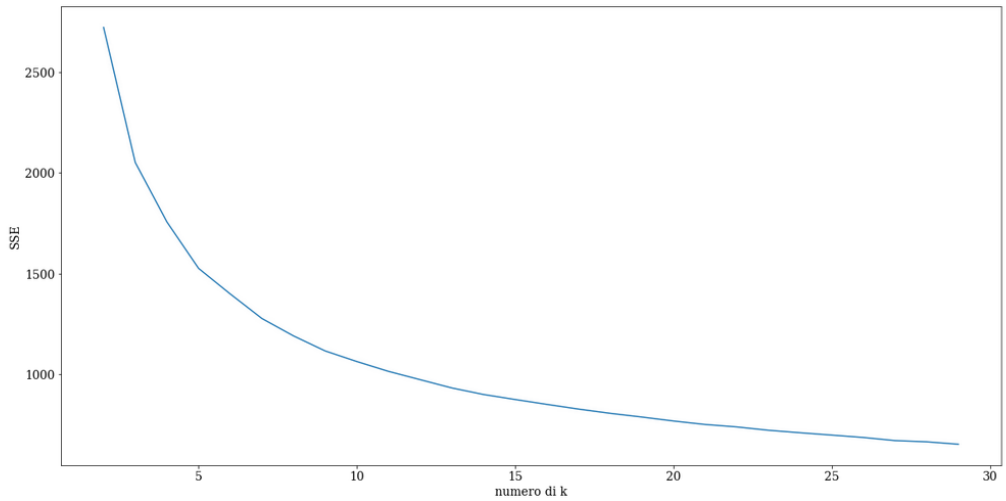


Figura 4: K-Means: SSE analisi.

In base a questi risultati, il valore k di cluster da scegliere è 11, in quanto a partire da tale valore il SSE, che è più alto per un numero di clusters inferiore, si stabilizza intorno a 1000 e non subisce significative variazioni.
Per avere un ulteriore riscontro sul numero k di clusters da scegliere, oltre che con la misura del SSE, il risultato è stato messo in relazione alla misura della silhouette che permette di valutare il clustering sulla base non solo della coesione interna ad ogni cluster ma anche della separazione tra di essi. Essa assume in genere valori che vanno da -1 a 1 e, più è vicina a 1, migliore è il clustering.
L'algoritmo di k-means è stato dunque iterato per un numero di k che va da 5 a 15 e, per ogni iterazione, è stata memorizzata la silhouette, come mostra la seguente figura:

Figura 5: K-Means: Silhouette analisi.

Il valore della silhouette è in tutti i casi molto basso, infatti non supera lo 0.28. Tuttavia, si ha un picco per un numero di clusters pari a 7. Per 7 clusters, infatti, la silhouette ha un valore (arrotondato per eccesso) di 0.28, mentre per 11 clusters essa scende a 0.27. Per un numero di clusters pari a 7, però, come mostra la figura del SSE, si registra un aumento del SSE che si aggira intorno a 1200.
Premesso che si tratta di oscillazioni relativamente poco significative, è stato deciso di dare la priorità alla misura della silhouette per la scelta del numero k di clusters, in quanto essa fornisce una misura anche della separazione tra i clusters e in relazione a una silhouette di 0.28 il valore di SSE non subisce un aumento eccessivamente grande. Per questo motivo, è stato scelto di usare 7 clusters.

2.1.2 Caratterizzazione dei clusters ottenuti

Nella figura 6 sono mostrati i 7 clusters ottenuti applicando l'algoritmo k-means.

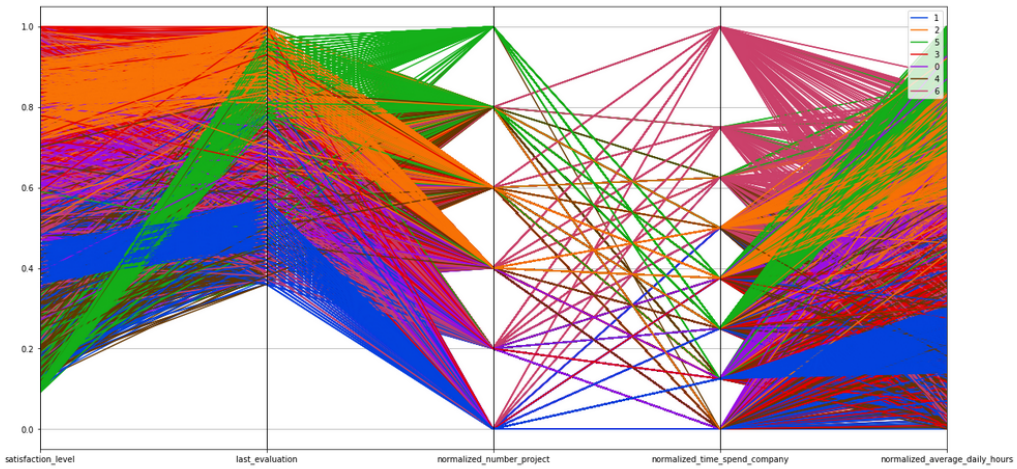


Figura 6: Parallel coordinates dei cluster risultanti dal k-means.

Come si può notare, il cluster 1, rappresentato dal colore blu, appare quello più nettamente distinto dagli altri, seguito dal 5, rappresentato dal verde, e dal 2, rappresentato dall'arancione. Per gli altri clusters invece, sono più evidenti sovrapposizioni ed essi appaiono, dunque, meno distinti gli uni dagli altri.

- Il cluster 0 contiene 3226 impiegati. La media di *normalized_average_daily_hours* è significativamente superiore a quella dell'intero dataset, essendo 11.13 contro 9.34. La maggior parte delle persone appartenenti a questo cluster non ha lasciato il lavoro (97.5%).
- Il cluster 1 contiene 2562 impiegati. La media del numero di progetti è molto più bassa rispetto a quella calcolata sul dataset, infatti è 2.25 contro 3.8. Anche il numero di ore lavorative giornaliere è basso rispetto alla media totale del dataset, infatti è 6.95 contro 9.34. Più bassi sono anche i valori di *satisfaction_level* e *last_evaluation*, che sono rispettivamente 0.43 e 0.54 contro 0.61 e 0.72. Esso contiene inoltre, il maggior numero di impiegati, rispetto a tutti gli altri clusters, che hanno lasciato il lavoro (60.2%).
- Il cluster 2 contiene 2323 impiegati. Le medie del numero di progetti e il numero di ore lavorative sono più alte rispetto a quelle calcolate sull'intero dataset, ovvero sono rispettivamente 4.7 e 11.23 contro 3.8 e 9.34. Anche le medie di *satisfaction_level* e *last_evaluation* superano significativamente quelle calcolate sull'intero dataset, essendo rispettivamente 0.80 e 0.85 contro 0.61 e 0.72. Questo cluster è formato per la maggior parte da impiegati che non hanno lasciato il lavoro (61.3%).
- Al cluster 3 appartengono 3433 impiegati. La media di *satisfaction_level* è più alta di quella calcolata sull'intero numero di record, infatti è 0.80 contro 0.61. Per quanto riguarda il numero di ore lavorative giornaliere inoltre, la media è 7.65, al di sotto di quella dell'intero dataset, cioè 9.34. Rispetto agli altri clusters, questo è quello che contiene il maggior numero di impiegati che non hanno lasciato il lavoro (98.7%), che non hanno avuto una promozione negli ultimi cinque anni (98%), che non hanno avuto incidenti sul lavoro (83.4%) e che hanno il salario più basso (47.1%).
- Il cluster 4 contiene 1480 impiegati. Le medie di *satisfaction_level* e del numero di ore di lavoro giornaliere sono inferiori rispetto a quella dell'intero dataset, infatti sono rispettivamente 0.40 e 7.95 contro 0.61 e 9.34. La media del numero di progetti, invece, supera quella calcolata sull'intero dataset ed è 4.65 contro 3.8. La maggior parte degli impiegati di questo cluster non ha lasciato il lavoro (94.7%).
- Al cluster 5 appartengono 1283 impiegati. La media del livello di soddisfazione è molto più bassa rispetto a quella dell'intero dataset essendo 0.14 contro 0.61. Significativamente più alte sono invece, le medie del numero di progetti realizzati e del numero di ore giornaliere lavorative, che sono rispettivamente 5.95 e 12.53 contro 3.8 e 9.34. La maggior parte degli impiegati appartenenti a questo cluster ha lasciato il lavoro (71.9%). Questo cluster, inoltre, contiene il minor numero di persone che hanno avuto promozioni negli ultimi cinque anni (solamente lo 0.9%) e che hanno un salario alto (5.1%).
- Il cluster 6 contiene 692 impiegati. Le medie di tutti gli attributi sono pressappoco allineate con quelle dell'intero dataset, eccetto per il numero di anni trascorsi presso la compagnia, per cui la media, che è 7.92, supera significativamente quella del dataset, ovvero 3.5. La maggior parte degli impiegati appartenenti a questo cluster ha un salario medio (51.3%). Inoltre, a questo cluster appartengono il minor numero di impiegati (solo lo 0.9%), rispetto a tutti gli altri clusters, che hanno lasciato il lavoro.

Il cluster 1 e il cluster 5, che sono gli unici due in cui il maggior numero di impiegati ha lasciato il lavoro, possono essere usati per delineare due tipi principali di impiegati che lasciano il lavoro. Nel primo caso, si tratta di persone che hanno un livello di soddisfazione, una valutazione, un numero di progetti ed anche una quantità di ore lavorative giornaliere inferiori rispetto alla media. Nel secondo caso, quello delineato dal cluster 5, si tratta di impiegati che hanno un numero di progetti e di ore giornaliere lavorative superiori rispetto alla media, ma tra i quali si ha il minor numero di persone, rispetto a tutti gli altri impiegati, che ha avuto una promozione negli ultimi cinque anni e che percepisce un salario alto. Appare ovvio, dunque, come per questo cluster il livello di soddisfazione sia molto più basso rispetto alla media.
I cluster 2 e 3 invece, formati per la maggior parte da impiegati che non hanno lasciato il lavoro, mostrano chiaramente come questa scelta corrisponda a livelli di soddisfazione e valutazione più alti rispetto alla media.
Il cluster 6, infine, contenente il minor numero di impiegati, rispetto agli clusters, che hanno lasciato il lavoro, delinea un'altra caratteristica di questi impiegati, infatti la maggior parte di quelli appartenenti a questo cluster ha un salario alto e ovviamente, lavora da più anni nella compagnia rispetto ad altri.

2.2 Clustering con DBscan

Il secondo algoritmo di clustering usato è il DBSCAN. Esso individua i cluster come regioni ad alta densità di punti, permettendo di separarli da aree a bassa densità che costituiscono il noise. L'algoritmo non richiede che sia fissato in precedenza il numero di cluster che si vogliono ottenere, ma è necessario fissare un raggio epsilon (*eps*), e un numero minimo di punti *minPts* che devono essere contenuti entro tale raggio. In base a questi valori i punti vengono suddivisi in core, border e noise.
Il DBSCAN non è stato effettuato utilizzando l'intero dataset ma solamente su un gruppo ristretto di features: *satisfaction_level*, *last_evaluation*, *normalized_number_project*, *normalized_time_spent_company* e *normalized_average_daily_hours*.
Sono stati quindi eliminati gli attributi categorici, quelli binari e l'attributo left in quanto target dell'analisi.
Le proprietà *satisfaction_level* e *last_evaluation*, come già descritto, hanno un range di valori che va da 0 a 1 e si è reso quindi necessario normalizzare *number_project*, *time_spent_company* e *average_daily_hours*.

2.2.1 Studio dei parametri

Per il calcolo ottimale dell'epsilon ci si è avvalsi dell'algoritmo k-nearest neighbor; sono state eseguite molteplici prove per valori di *minPts* che vanno da 5 a 100 aumentando ad ogni prova successiva il valore di 5.
Ad ogni prova è stato individuato il miglior *eps* analizzando la curva ricavata da k-nearest neighbor per quel valore di *minPts* e ne è stato calcolato il DBSCAN e la corrispondente silhouette.
Infine sono stati scelti i valori di *eps* e *minPts* che hanno ottenuto il miglior punteggio di silhouette della serie.
Nel nostro caso abbiamo ottenuto per *eps* 0.2 e *minPts* 75 con una silhouette di 0.243.

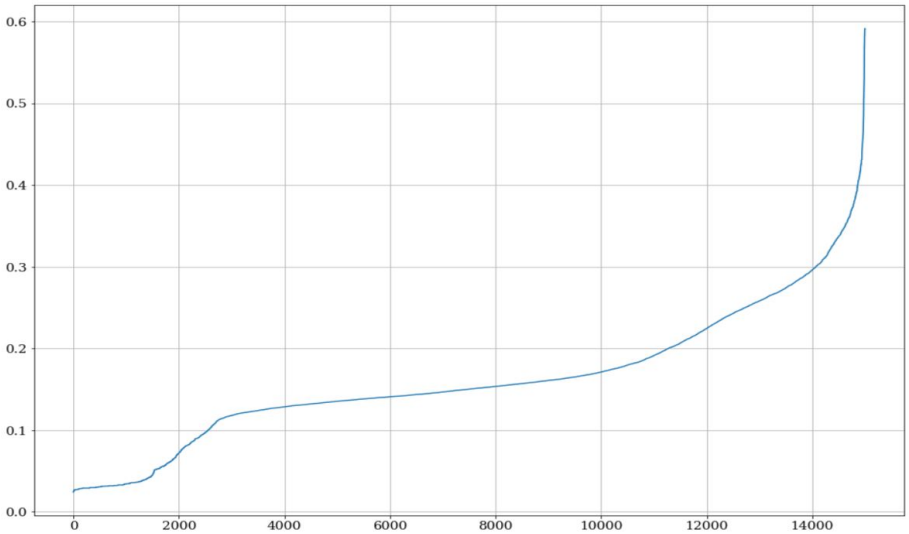
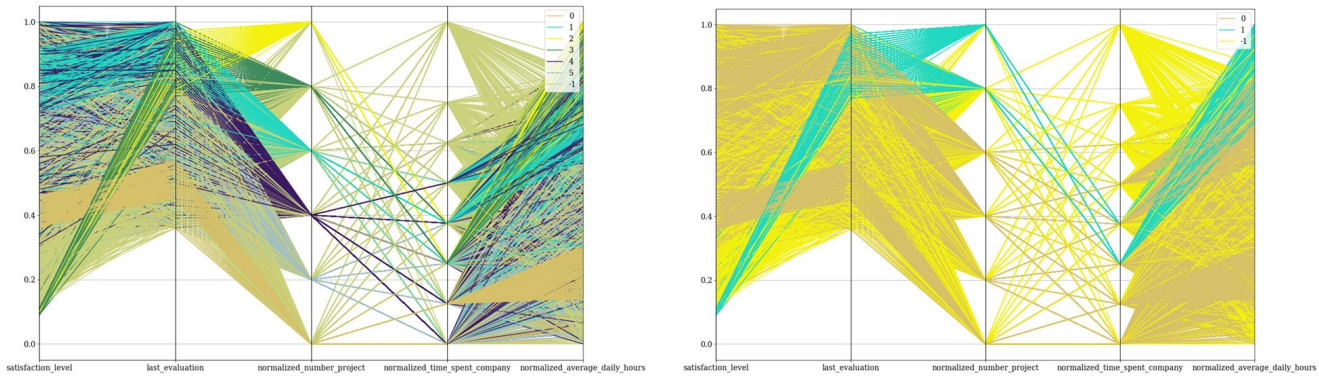


Figura 7: DBSCAN. K-nearest neighbor con minPts: 75.

2.2.2 Caratterizzazione dei clusters ottenuti

Questi valori hanno fornito una clusterizzazione del dataset in un cluster maggiore formato da 11545 record ed uno minore di 959 record oltre a 2495 nodi identificati come noise.
Prendendo invece *eps* pari a 0.19 il dataset viene suddiviso in 6 cluster (oltre al noise), che tuttavia risultano sparsi e frammentari, al contrario invece dei valori scelti in cui i 959 record ottenuti nel cluster minore hanno valori di features molto simili fra loro con uno scostamento di questi ultimi molto ridotto in confronto agli altri cluster.



a) b)

Figura 8: DBSCAN. Confronto dei risultati: a sinistra (a) si è utilizzato un *eps* di 0.19, a destra (b) un *eps* di 0.2, in entrambi i casi *minPts* è fissato a 75.

In figura 9 sono stati riportati i nodi del cluster in questione rapportati con l'attributo left, dove risulta che su un totale di 959 persone, 850 di questi hanno lasciato l'azienda e soltanto 109 hanno deciso di rimanere.
Come si nota dal grafico, coloro che hanno deciso di lasciare hanno valori fortemente simili tra loro per tutte le features prese in esame, ovvero *satisfaction_level* tendenzialmente basso, *last_evaluation* elevato, 6 e 7 progetti (i più alti del dataset), *normalized_time_spent_company* ristretto al range 0.25-0.37 e soltanto i valori più alti delle ore lavorative medie giornaliere.

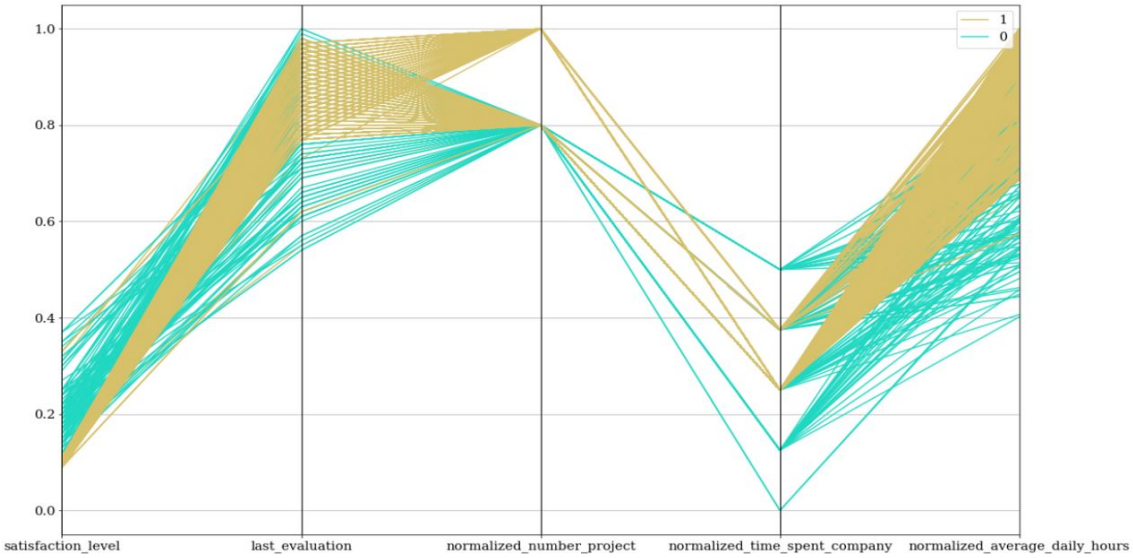


Figura 9: DBSCAN. Dettaglio del cluster formato da 959 record, in celeste sono evidenziati i dipendenti che sono rimasti, in oro quelli che hanno deciso di lasciare.

2.3 Clustering con Hierarchical

Il terzo algoritmo di clustering applicato sui dati è lo hierarchical di tipo agglomerative. Esso considera inizialmente ogni punto come un singolo cluster e ad ogni iterazione riunisce i due clusters più vicini (cioè tra i quali vi è la minore distanza o la maggiore similarità), in modo da ottenere un insieme di clusters annidati organizzati in una struttura ad albero. A differenza di altri algoritmi di clustering, come il k-means, esso non richiede di specificare a priori il numero di clusters che si vogliono ottenere.
Le variabili scelte per eseguire questa analisi sono quelle numeriche non binarie. I valori binari infatti sono per natura suddivisi in 2 cluster, mentre per quelli categorici non è possibile stabilire un criterio per calcolare la distanza.
Gli attributi numerici sono stati normalizzati poiché valori appartenenti a scale diverse potrebbero sbilanciare la matrice delle distanze verso un particolare attributo. I valori quindi utilizzati sono: *satisfaction_level*, *last_evaluation*, *normalized_average_daily_hours*, *normalized_number_project*, *normalized_time_spent_company*.
Per scegliere quale funzione utilizzare per calcolare la matrice delle distanze si sono svolti esperimenti con la distanza euclidea, la manhattan e la cosine. Per testare quale fosse la più adatta si è eseguita un'analisi della silhouette. Da questa si è compreso che con la cosine si ottengono i valori più bassi anche per un numero minore di cluster. La manhattan e la euclidea invece sono paragonabili, i risultati variano leggermente a seconda del criterio di collegamento utilizzato.
Alla fine si è scelto di utilizzare la distanza euclidea poiché è il metodo di misurazione più standard e ci permette di testare anche il criterio di Ward per calcolare i cluster.

2.3.1 Scelta tra i diversi algoritmi

Per testare quale funzione di collegamento utilizzare anche in questo caso abbiamo eseguito diversi esperimenti e osservato i cluster che si formavano al variare dell'algoritmo e del numero di cluster scelto.
Il complete linkage restituisce per qualunque numero di cluster dei valori della silhouette vicini allo zero, il che indica che molti record erano associati a cluster sbagliati. L'average linkage d'altra parte crea cluster poco uniformi. Già con valori bassi di *n_cluster* si formano cluster di dimensione irrisoria. Il single linkage infine presenta entrambe le problematiche.
Queste analisi sono confermate anche dai relativi dendrogrammi, infatti nel complete si nota che la divisione tra i cluster non è ben definita, nell'average è presente un cluster molto ampio e alcuni di dimensione talmente ridotta che non sono neanche visibili nel grafico e nel single vi sono tantissimi cluster minuscoli e le distanze tra cluster diversi sono molto ridotte.

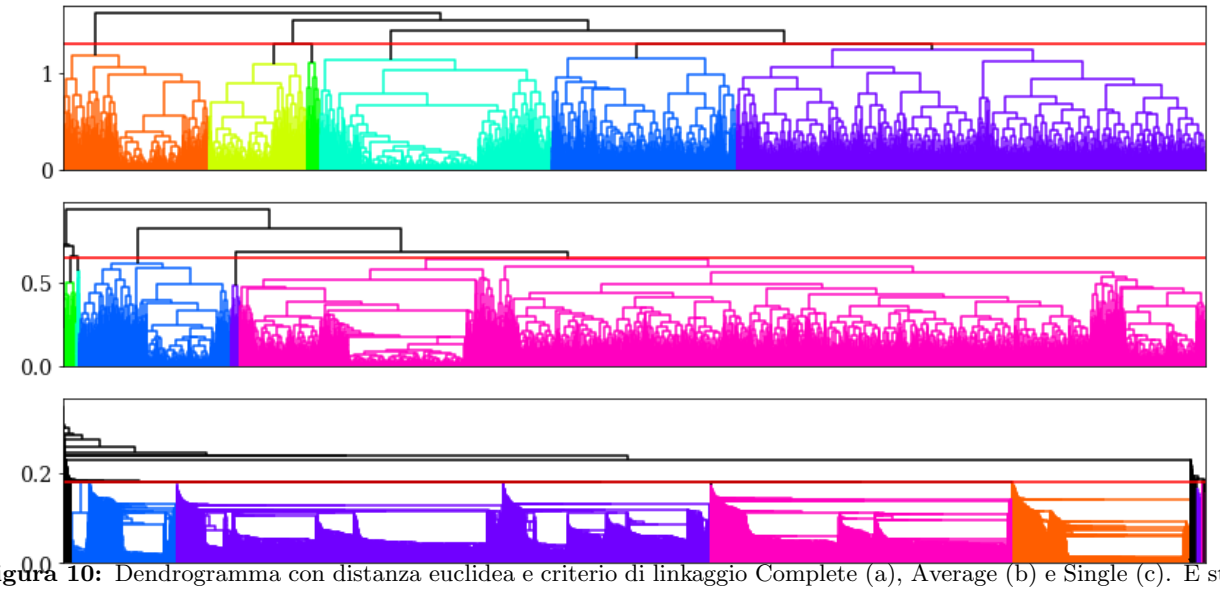


Figura 10: Dendrogramma con distanza euclidea e criterio di linkaggio Complete (a), Average (b) e Single (c). È stata fissata una soglia di threshold solo per rendere più chiara la lettura dei grafici.

Infine si è sperimentato l'algoritmo con il criterio di Ward. Questo metodo, per scegliere quali coppie di cluster

unire ad ogni passo, si basa sul valore ottimale di una funzione obiettiva. Le funzioni utilizzate possono essere diverse, quella più standard è la somma al quadrato degli errori, infatti il criterio di Ward è anche chiamato metodo della minima varianza.
Questo metodo ha restituito i risultati migliori. Infatti, anche verificando visivamente tramite dedrogramma, si può notare che in questo caso le distanze tra cluster diversi sono maggiori rispetto agli altri metodi, mentre sono minimizzate quelle interne. Inoltre le dimensioni dei vari cluster sono piuttosto omogenee.

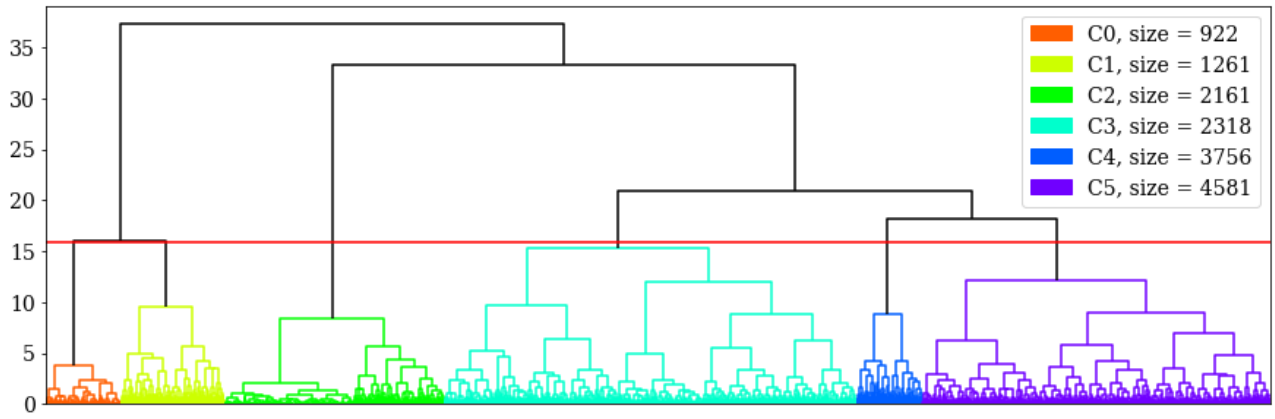


Figura 11: Dendrogramma con distanza euclidea e criterio di Ward.

Per scegliere quale sia il numero di cluster ideale si possono sfruttare diversi criteri.
Innanzi tutto si è osservato il dendrogramma e si è cercato di dividere i cluster aventi maggiore distanza. Questo ci ha portato a suddividere il nostro database dai 3 ai 7 cluster. Come ulteriore conferma di questa ipotesi abbiamo sfruttato le analisi sulla silhouette fatte in precedenza. Siccome per $n\_cluster = 5, 6, 7, 11, 14$ o 15 la silhouette presenta un jump positivo, il che implica che tali valori sono buoni candidati, restringiamo la nostra scelta tra 5, 6 o 7.
Infine abbiamo deciso che il numero ottimale di cluster sarebbe stato 6 osservando la distribuzione dei cluster in una scatter matrix.

2.4 Valutazione finale del miglior metodo di clustering

Mettendo i vari metodi di clustering a confronto si può notare come tutti e tre hanno raggruppato i record in un numero che va dai 6 ai 7 cluster. Analizzando i risultati più nello specifico, tuttavia, si nota che:

- Il K-means ci ha fornito 7 cluster, tra i quali 2 risultano avere un consistente numero di persone che hanno lasciato. Il primo denota una tipologia di persone con livelli di soddisfazione, valutazione, numero di progetti e ore giornaliere lavorative inferiore alla media. L'altro gruppo, al contrario, raggruppa persone con numero di progetti e ore giornaliere lavorative tra le più alte del dataset.

- Nel caso del DBSCAN, i 6 cluster creati (più i noise) risultano frammentari e sparsi senza nette distinzioni. Questo ci ha portato ad optare (come confermato dal k-nearest neighbor) per un *eps* maggiore ed un risultato di due soli cluster che hanno messo in evidenza un raggruppamento (seppur ristretto) molto omogeneo di persone che hanno abbandonato l'azienda. Questo raggruppamento somiglia, per caratteristiche, ad uno di quelli ricavati dal K-means.

- L'algoritmo di Hierarchical Clustering, con l'ausilio del criterio di Ward, è quello che ha dato i risultati migliori. Questo metodo ci ha fornito 6 cluster dove le distanze interne sono minimizzate fornendo raggruppamenti più equilibrati, mentre sono accentuate quelle fra cluster diversi. Dal dendrogramma si può notare anche come questi ultimi risultino avere dimensioni più omogenee rispetto agli altri risultati.

3 Association Rules

In questo capitolo cercheremo di comprendere se vi sono delle relazioni nascoste tra i dati, ossia ad esempio se vi sono delle caratteristiche di un impiegato che si riscontrano più frequentemente insieme. In particolare utilizzeremo l'algoritmo Apriori per generare gli itemsets frequenti e ricavarne le relative regole di associazione.
Nell'analisi il passaggio fondamentale è la scelta dei valori di *minimum support* e *minimum confidence*. Infatti al variare di *min_sup* potremmo passare da pochi itemsets molto frequenti a tanti che compaiono raramente. Allo stesso modo ponendo un *min_conf* alto otterremmo poche regole che si verificano quasi sempre, mentre con valori più bassi ne avremmo un numero più alto ma con meno probabilità che siano veritiere.
I valori tipici scelti sono 2-10% per *min_sup* e 70-90% per *min_conf*.
Una volta identificate le regole di associazione più significative possiamo utilizzarle per diversi scopi. Ad esempio possono servire per predire il valore di una variabile oppure per sostituire dei valori mancanti.
Nel nostro caso il database non presenta missing values per cui non è necessario sfruttare le regole di associazione per gestirli. Useremo quindi le regole estratte per capire quali siano le caratteristiche di un impiegato che influenzano positivamente la probabilità che questo lasci il lavoro.

3.1 Preparazione dei dati

Prima di poter procedere con l'analisi è necessario trasformare il nostro database relazionale in uno transazionale. Ogni record quindi equivale ad una transazione e i valori che assume per ogni attributo rappresentano gli items. Inoltre, poiché l'algoritmo Apriori lavora su attributi categorici, quelli numerici sono stati discretizzati e trasformati in stringhe.
Nel primo capitolo, relativo alla data understanding, avevamo suddiviso le variabili continue in 15 bins, seguendo la regola di Sturges. Tuttavia, in questo capitolo, abbiamo ritenuto migliore diminuire la quantità di bins poiché un numero troppo alto di valori distinti per una variabile poteva portare a pochi itemsets frequenti e di conseguenza poche regole di associazione. I due valori che sono stati presi in considerazione quindi sono 3 e 5 in quanto consentono di mantenere la suddivisione in 15 bins fatta precedentemente ed eseguire un semplice raggruppamento. Dopo alcuni esperimenti si è deciso di utilizzare 5 bins per variabile in quanto questo permetteva una maggiore distinzione tra i valori ma allo stesso tempo forniva regole interessanti.
Un fattore da tenere in considerazione che potrebbe portare a risultati errati è lo sbilanciamento dei dati. Infatti se una variabile assume spesso un particolare valore è possibile che gli itemsets più frequenti lo contengano tutti. Per ovviare a questo problema è necessario prendere in considerazione anche valori di support minori ed eseguire una scrematura tra i risultati ottenuti.

3.2 Estrazione degli itemsets più frequenti

Il valore ottimale per il supporto minimo è molto influenzato dal contesto in cui stiamo lavorando e dalla distribuzione dei dati. Per questo motivo non esiste una regola o un indicatore per scegliere il valore migliore ma è necessario procedere tramite esperimenti.
Abbiamo quindi eseguito l'algoritmo diverse volte per valori decrescenti di support. Nella seguente tabella possiamo vedere i 20 itemsets più frequenti di lunghezza almeno due.

Itemsets	Support
('WA0', 'PL5Y0')	0.839
('L0', 'PL5Y0')	0.742
('L0', 'WA0')	0.629
('L0', 'WA0', 'PL5Y0')	0.613
('low', 'PL5Y0')	0.483
('TSC3', 'PL5Y0')	0.421
('low', 'WA0')	0.418
('medium', 'PL5Y0')	0.418
('low', 'WA0', 'PL5Y0')	0.415
('TSC3', 'WA0')	0.370

Itemsets	Support
('medium', 'WA0')	0.367
('TSC3', 'WA0', 'PL5Y0')	0.363
('medium', 'WA0', 'PL5Y0')	0.358
('low', 'L0')	0.343
('medium', 'L0')	0.342
('low', 'L0', 'PL5Y0')	0.339
('medium', 'L0', 'PL5Y0')	0.330
('TSC3', 'L0')	0.324
('TSC3', 'L0', 'PL5Y0')	0.316
('ADH2', 'PL5Y0')	0.291

Tabella 2: I primi 20 itemsets più frequenti.

Abbiamo quindi ottenuto 20 itemsets di dimensione 2 o 3 con support compreso tra 0.291 e 0.839. Questi valori così alti sono giustificati dal fatto che il nostro database è sbilanciato per diverse variabili. Infatti gli item presenti sono:

- 'PL5Y0' : *promotion_last_5years* = 0, gli impiegati che non hanno ricevuto promozioni negli ultimi 5 anni comprendono il 97.9% di tutti i dati;
- 'WA0': *work_accident* = 0, quelli che non hanno avuto incidenti sul lavoro rappresentano l'87.5% del database;
- 'L0' : *left* = 0, il 76.2% dei lavoratori dell'azienda non hanno lasciato il lavoro;
- 'low' e 'medium' : si riferiscono al salario, sono rispettivamente il 48.8% e il 43.0% dell'intera popolazione;
- 'TSC3' : *time_spent_company* = 3, più del 40% ha lavorato nell'azienda per esattamente 3 anni.

Di conseguenza possiamo concludere che gli itemsets più frequenti che abbiamo ottenuto derivano dal fatto che i singoli item sono estremamente frequenti e non che la presenza di alcuni valori insieme implichi una correlazione. Si è deciso quindi di non considerare gli itemsets contenenti 'PL5Y0' e 'WA0' in quanto sono i due valori più sbilanciati e non forniscono informazioni ulteriori sui dati. Come ulteriore riprova abbiamo generato gli itemsets massimali per min_sup = 0.10. Quelli risultanti contenevano tutti sia 'PL5Y0' che 'WA0', confermando l'ipotesi che tali valori sono troppo comuni per essere utili. Eliminando gli itemsets relativi si ottiene:

Itemsets	**Support**	**Itemsets**	**Support**
('low', 'L0')	0.343	('TSC2', 'L0')	0.213
('medium', 'L0')	0.342	('sales', 'L0')	0.208
('TSC3', 'L0')	0.324	('SL3', 'L0')	0.206
('NP3', 'L0')	0.266	('ADH4', 'L0')	0.206
('NP4', 'L0')	0.264	('LE2', 'L0')	0.189
('SL4', 'L0')	0.228	('LE3', 'L0')	0.188
('ADH2', 'L0')	0.226	('TSC3', 'medium')	0.181
('SL5', 'L0')	0.219	('LE4', 'L0')	0.174
('TSC3', 'low')	0.214	('ADH2', 'TSC3')	0.166
('ADH3', 'L0')	0.212	('LE5', 'L0')	0.161

Tabella 3: I 20 itemsets più frequenti che non contengono 'PL5Y0' e 'WA0'.

Notiamo quindi che gli itemsets contengono quasi tutti il valore 'L0', tuttavia poiché *left* è l'attributo centrale nella nostra analisi cancellare anche questo valore potrebbe portare a previsioni scorrette.
Per ottenere anche associazioni interessanti è stato necessario diminuire il valore del support, infatti se *min_sup* è maggiore della frequenza di un valore singolo non è possibile che vengano generati itemsets che lo contengano. D'altra parte fissare un valore troppo basso di support genera una mole di risultati difficilmente gestibile e regole valide solo per una porzione molto ristretta della popolazione.
Alla fine si è deciso di procedere all'estrazione delle regole di associazione fissando *min_sup* = 0.07. Questo dà origine a 731 itemsets frequenti che contengono una varietà sufficientemente ampia di items.

3.3 Estrazione delle regole di associazione

In questa analisi, per ridurre il numero di regole prodotte, abbiamo considerato solo regole di associazione aventi un unico elemento nella testa della regola.
Questa semplificazione tuttavia non comporta particolari restrizioni nell'analisi. Infatti, le regole che più ci interessano sono quelle che hanno come conseguenza l'eventuale abbandono dell'azienda da parte dell'impiegato. Inoltre, regole aventi più elementi nella testa non aggiungono informazioni più significative rispetto alle altre.
Ad esempio, per comprendere questo concetto, possiamo pensare ad una regola del tipo $\{a, b\} \to \{c, d\}$ e alle sue semplificazioni $\{a, b\} \to c$ e $\{a, b\} \to d$. Se la regola più complessa presenta sufficiente support e confidence allora sarà lo stesso anche per le due più semplici. Infatti se abbiamo due itemsets X, Y con $X \subseteq Y$ allora per il support avremo che $s(X) \geq s(Y)$ e lo stesso vale per la confidence $c(X) \geq c(Y)$.
Abbiamo quindi preso i pattern frequenti estratti nel paragrafo precedente, che erano solo quelli di dimensione maggiore di 2 in quanto è necessario la presenza di almeno due item per non formare una regola banale, ed applicato l'algoritmo Apriori.

Nel valutare le regole risultanti abbiamo dato una forte importanza al valore del *lift* perché questo indica l'accuratezza della regola. Non abbiamo tenuto in considerazione quindi le regole aventi *lift* minore di 1.1. Essendo infatti vicino ad 1 significa che le occorrenze della testa e del corpo della regola sono quasi indipendenti l'una dall'altra.
Anche in questo caso non esiste un metodo per stabilire il livello di confidence più adeguato quindi procederemo in modo sperimentale. Abbiamo quindi testato valori decrescenti di *min_conf* a partire da 0.90 ed osservato i risultati ottenuti. Nella seguente tabella sono riportate in ordine di *lift* le regole aventi *min_sup* maggiore di 0.7 e *min_conf* maggiore di 0.9.

Rule	Conf	Lift
(NP2, L1, LE2, TSC3) → 'SL2'	0.95	6.32
('NP2', 'L1', 'TSC3') → 'SL2'	0.94	6.30
('NP2', 'L1', 'LE2') → 'SL2'	0.94	6.28
('SL2', 'L1', 'TSC3') → 'NP2'	0.99	6.23
('L1', 'LE2', 'TSC3') → 'SL2'	0.94	6.23
('NP2', 'L1') → 'SL2'	0.93	6.20
('L1', 'LE2', 'TSC3') → 'NP2'	0.98	6.17
('L1', 'TSC3') → 'SL2'	0.92	6.12
('L1', 'TSC3') → 'NP2'	0.96	6.05
('SL2', 'LE2', 'TSC3') → 'NP2'	0.96	6.04

Rule	Conf	Lift
('SL2', 'L1') → 'NP2'	0.96	6.03
('NP2', 'LE2', 'TSC3') → 'SL2'	0.91	6.02
('L1', 'LE2') → 'NP2'	0.94	5.89
('L1', 'AMH2') → 'NP2'	0.93	5.86
('SL2', 'NP2', 'LE2') → 'L1'	0.96	4.02
('SL2', 'NP2', 'TSC3') → 'L1'	0.95	3.99
('SL2', 'LE2', 'TSC3') → 'L1'	0.94	3.95
('NP2', 'LE2', 'TSC3') → 'L1'	0.93	3.91
('SL2', 'NP2') → 'L1'	0.90	3.78
('SL2', 'NP2') → 'TSC3'	0.94	2.19

Tabella 4: Le 20 regole aventi lift più alto.

Si osserva facilmente che le regole risultanti contengono una combinazione dei valori 'NP2', 'LE2', 'SL2', 'TSC3' e 'L1'. Il che indica una forte correlazione tra la presenza di tali valori in contemporanea in un record. Questi valori rappresentano un impiegato che ha portato a termine 2 progetti, ha un livello di soddisfazione ed una valutazione da parte dell'azienda piuttosto basse (rispettivamente tra 0.272-0.454 e tra 0.488-0.616) e ha passato 3 anni nella compagnia prima di licenziarsi.
Questo significa che se riscontriamo un lavoratore che presenta almeno 2 di queste caratteristiche è fortemente probabile che possegga anche le altre 3.

3.4 Predire tramite le regole di associazione se un impiegato lascerà l'azienda

Per predire in modo più accurato possibile l'eventuale licenziamento di un impiegato abbiamo preso in considerazione solo le regole aventi come conseguenza *left* uguale a 0 o ad 1. Infatti abbiamo ritenuto necessario creare anche una caratterizzazione del lavoratore che decide di non licenziarsi per controllare che le regole siano relative solo a coloro che lasciano e non a qualunque impiegato. Le regole che abbiamo ottenuto sono le seguenti:

Rule	Conf	Lift
('SL2', 'NP2', 'LE2') → 'L1'	0.96	4.02
('SL2', 'NP2', 'TSC3') → 'L1'	0.95	3.99
('SL2', 'LE2', 'TSC3') → 'L1'	0.94	3.95
('NP2', 'LE2', 'TSC3') → 'L1'	0.93	3.91
('SL2', 'NP2') → 'L1'	0.90	3.78
('NP2', 'ADH2', 'TSC3') → 'L1'	0.89	3.75
('SL2', 'LE2') → 'L1'	0.87	3.65
('NP2', 'TSC3', 'low') → 'L1'	0.86	3.62
('SL2', 'TSC3') → 'L1'	0.85	3.57
('NP2', 'LE2') → 'L1'	0.84	3.52

Rule	Conf	Lift
(ADH3', 'TSC3') → 'L0'	0.99	1.30
('TSC2', 'NP3') → 'L0'	0.99	1.30
('SL3', 'NP3') → 'L0'	0.98	1.30
('TSC2', 'low') → 'L0'	0.99	1.29
('TSC2', 'NP4') → 'L0'	0.99	1.29
('LE3', 'TSC3') → 'L0'	0.98	1.29
('ADH3', 'NP3') → 'L0'	0.98	1.29
('LE3', 'medium') → 'L0'	0.98	1.29
('high') → 'L0'	0.93	1.23
('WA1',) → 'L0'	0.92	1.21

Tabella 5: Le 10 regole aventi lift più alto per *left* = 1 a sinistra e *left* = 0 a destra.

Come avevamo notato nei capitoli precedenti, gli impiegati che hanno lasciato sono molto più caratterizzati di quelli che rimangono, infatti nel primo caso i livelli di *lift* sono molto più alti.
Tra i valori nella tabella osserviamo che 'TSC3' e 'low' sono presenti sia per *left* = 1 che per *left* = 0. Questo significa che non sono caratteristiche specifiche di una categoria di impiegati ma che, come avevamo osservato

per gli itemsets, sono semplicemente dei valori frequenti.
La presenza in un record di almeno una coppia tra i valori 'SL2', 'NP2', 'LE2' e 'ADH2' incrementa in modo positivo la probabilità che l'impiegato lascerà l'azienda. Possiamo quindi classificare quelli che si licenziano come impiegati che solitamente hanno un livello di soddisfazione basso, compreso tra 0.272 e 0.454, hanno svolto solo 2 progetti, hanno ricevuto una valutazione inferiore alla media, tra 0.488 e 0.616, e hanno lavorato in media tra le 6.5 e 8.4 ore al giorno.
Gli impiegati che invece hanno deciso di restare sono caratterizzati da un numero di ore giornaliere più alte, tra le 8.4 e le 10.4, hanno passato 2 anni nell'azienda, lavorato a 3 o 4 progetti ed hanno ricevuto una valutazione nella media, ossia tra 0.616 e 0.744. Non è chiaro se ricevono uno stipendio leggermente più alto oppure semplicemente questa regola non è presente tra quelli che hanno lasciato poichè il support è troppo basso visto lo sbilanciamento del database. Lo stesso ragionamento vale per la presenza della regola che *work_accident* = 1 implica che l'impiegato è rimasto.

4 Classification

L'ultima fase di analisi riguarda lo sviluppo di modelli di classificazione costruiti allo scopo di predire se e quali impiegati lasceranno la compagnia, utilizzando l'attributo target *left*. Per fare questo, sono stati usati due principali modelli di classificazione, ovvero Decision Tree e Random Forest. Inoltre, si è deciso di costruire i differenti modelli usando sia il dataset originario, sbilanciato rispetto all'attributo *left* (ci sono 3571 'left' contro 11428 'didn't left'), sia un dataset bilanciato rispetto all'attributo target. Per il bilanciamento del dataset sono state effettuate prove con SMOTE (Synthetic Minority Over-sampling Technique), random oversampling e random undersampling. Poiché non si è osservata una significativa variazione nei risultati, si è deciso di costruire il dataset bilanciato con random undersampling. In esso si ha lo stesso numero di *left*=0 ("didn't left") e di *left*=1, per un totale di 7142 record.

4.1 Generazione dei decision trees

Prima di procedere con la costruzione dei decision trees è stato necessario trasformare le variabili categoriche *salary* e *departments* in numeriche di tipo binario, in quanto il modello usato, fornito dalla libreria python Scikit-learn, funziona con valori numerici e permette di effettuare solo binary splits. Per *salary*, ad esempio, sono stati ottenuti i tre attributi *salary_val_high*, *salary_val_low*, *salary_val_medium*, ognuno dei quali può assumere valore 0 o 1.
Per la generazione e la valutazione dei diversi decision trees sono stati usati il metodo holdout, che usa due terzi del dataset per il training e un terzo per il testing, e la cross validation, suddividendo il dataset in dieci parti di cui alternativamente nove vengono usate come training set e una come test set.
Sia per il dataset sbilanciato che per quello bilanciato sono stati generati diversi decision trees ottenuti variando i parametri riguardanti la profondità dell'albero, il minimo numero di record necessari per lo splitting di un nodo, il numero massimo e il tipo di features considerate per la costruzione dell'albero e il criterio usato per misurare la qualità di uno split, ovvero gini o entropy.
Dai diversi alberi generati si è potuto notare come gini ed entropy dessero risultati molto simili tra loro, alcune volte leggermente migliori per gini e altre usando entropy. In alcuni casi i risultati erano leggermente più significativi se venivano escluse dalla fase di apprendimento alcune features, ovvero quelle riguardanti il dipartimento di appartenenza di ogni impiegato. È stato deciso, dunque, di usare gini come misura della qualità degli splits e di escludere dalle features quelle riguardanti il dipartimento di appartenenza di ogni impiegato, mantenendo invece, *satisfaction_level*, *last_evaluation*, *number_project*, *promotion_last_5years*, *average_daily_hours*, *time_spent_company*, *work_accident*, *salary_val_high*, *salary_val_low*, *salary_val_medium*, le quali tra l'altro, erano risultate già da analisi precedenti come il k-means legate alla decisione di un impiegato di lasciare la compagnia.

4.1.1 Validazione dei decision trees tramite test e training set

Riguardo al dataset non bilanciato, osservando in particolare le misure di *accuracy*, *precision*, *recall* e *f1-measure* ottenute attraverso una valutazione holdout per i diversi alberi generati, si può notare che si parte fin da subito, anche con modelli di profondità bassa (2 o 3) con valori di accuratezza elevati, tra l'85% e il 95% sia per training che per test set.
Per quanto riguarda la *precision* e la *recall*, la selezione delle features da usare nell'apprendimento del modello

ha permesso in qualche caso di colmare le differenze tra i valori per *left*=0 e *left*=1 ottenuti con la valutazione sul test set. Senza la selezione delle features che ha portato all'eliminazione di quelle che riguardano il dipartimento di appartenenza, infatti, ad esempio, un decision tree di profondità 3 presenta sul test set una *precision* del 98% per *left*=0 e del 77% per *left*=1 e una *recall* del 92% per *left*=0 e del 94% per *left*=1. La differenza notevole tra il valore di *precision* per *left*=0 e quello *left*=1 è data proprio dal fatto che il dataset è sbilanciato verso coloro che non hanno lasciato il lavoro, quindi per *left*=0. Sempre per un albero di profondità 3, selezionando solo dieci features ed eliminando quelle relative al dipartimento di appartenenza, si riesce a far diminuire la differenza tra *precision* per *left*=0, che rimane 98%, e quella per *left*=1, che sale all'88%, mentre i valori della *recall* si stabilizzano al 96% per *left*=0 e 92% per *left*=1. Si registra in generale un aumento del valore di *f1-measure* sia per *left*=0 che per *left*=1.
Sempre sul dataset sbilanciato, inoltre, è possibile notare che la differenza tra *precision* per *left*=0 e per *left*=1 diminuisce non solo attraverso la selezione delle features, ma anche aumentando la profondità dell'albero. In questo ultimo caso, è necessario tuttavia, controllare sempre che non si tratti di overfitting.
Per quanto riguarda le misure di *accuracy*, *precision*, *recall* e *f1-measure* ottenute a partire dal dataset bilanciato utilizzando sempre una valutazione holdout e selezionando le dieci features indicate in precedenza (4.1), invece, si può notare fin da subito che a una diminuzione non eccessiva dell'accuratezza dei modelli corrisponde, già a partire dai decision trees con profondità più bassa, un maggiore equilibrio dei valori della *precision* per *left*=0 e per *left*=1. Per un decision tree di profondità 3, ad esempio, si ha un'accuratezza sia sul training che sul test del 92%, una *precision* sul test del 93% per *left*=0 e del 91% per *left*=1 e una *recall* del 91% per *left*=0 e del 93% per *left*=1.
I valori di accuratezza ottenuti sia per il training che per il test attraverso cross validation sono molto simili e rispecchiano quelli forniti da holdout sia per il dataset non bilanciato, sia per quello bilanciato.

4.1.2 Decision tree su dataset non bilanciato

I valori di *accuracy* sul test set riguardanti il decision tree selezionato per il dataset non bilanciato sono rispettivamente 98% con validazione holdout e 98% con cross validation.
La seguente tabella mostra i valori di *precision*, *recall* e *f1-measure* relativi al test set per l'attributo target *left*=0 e *left*=1 ottenuti con holdout.

ValidationTarget	left	Precision	Recall	F1-measure	Support
Holdout	0	98%	99%	98%	3818
	1	98%	92%	95%	1132

Tabella 6: Statistiche Decision tree con dataset non bilanciato.

Anche se il valore dell'*accuracy* risulta molto alto sul training set (98%) e potrebbe far pensare che si tratti di overfitting, costruendo un grafico che mostra gli errori del training e del test set a confronto all'aumentare della profondità dell'albero, tale fenomeno è stato escluso in quanto non si registra un aumento degli errori sul test al diminuire di quelli sul training per il modello preso in considerazione.
È possibile notare, come evidenziato anche in precedenza (4.1.1) che pur trattandosi di un dataset sbilanciato verso *left*=0 sono stati raggiunti dei valori di *precision* e *recall* elevati anche per *left*=1.
Il modello usato individua due principali categorie di impiegati che lasciano il lavoro. Da un lato vi sono 1025 impiegati che hanno un livello di soddisfazione, un numero di progetti, una valutazione e un numero di ore lavorative giornaliere inferiori rispetto alla media (1.2). Questi impiegati, infatti, hanno un livello di soddisfazione minore o uguale a 0.5, un numero di progetti minore o uguale a 2.5, una valutazione compresa tra 0.4 e 0.6 e un numero di ore lavorative giornaliere compreso tra 5.8 e 7.5. Dall'altro vi sono 581 impiegati che hanno valutazione, numero di ore lavorative giornaliere, numero di anni trascorsi nella compagnia e livello di soddisfazione superiori rispetto alla media e numero di progetti vicino alla media (1.2) o superiore ad essa. Essi hanno, infatti, valutazione maggiore di 0.8, numero di ore lavorative maggiore di 10, numero di anni trascorsi nella compagnia compreso tra 4.5 e 6.5, soddisfazione maggiore di 0.7 e numero di progetti maggiore di 3.5.

4.1.3 Decision tree su dataset bilanciato

I valori di *accuracy* sul test set riguardanti il decision tree selezionato per il dataset bilanciato sono rispettivamente 94% con validazione holdout e 94% con cross validation. La seguente tabella mostra i valori di *precision*,

recall e *f1-measure* relativi al test set per l'attributo target *left*=0 e *left*=1 ottenuti con holdout.

ValidationTarget	left	Precision	Recall	F1-measure	Support
Holdout	0	94%	94%	94%	1171
	1	94%	94%	94%	1186

Tabella 7: Statistiche Decision tree con dataset bilanciato.

I valori di *accuracy*, *precision*, *recall* e *f1-measure* risultano più bassi di quelli ottenuti sul dataset non bilanciato, ma questo modello permette di raggiungere un equilibrio tra *precision* e *recall* per *left*=0 e *left*=1, infatti rispetto al modello precedente, si abbassa la *recall* per *left*=0 ma si alza quella per *left*=1.
Anche in questo caso, il valore di *accuracy* elevato sul training set (95%) potrebbe far pensare che vi sia overfitting, ma il fenomeno è stato escluso in quanto non si registra un aumento degli errori sul test al diminuire di quelli sul training per il modello preso in considerazione.
Anche questo modello individua due principali categorie di impiegati che lasciano il lavoro, molto simili a quelle individuate dal modello allenato su dataset non bilanciato. Da un lato vi sono 1023 impiegati che hanno un livello di soddisfazione e un numero di progetti inferiori rispetto alla media (1.2), una valutazione e un numero di ore lavorative giornaliere che partono da valori inferiori rispetto alla media e un numero di anni trascorsi nella compagnia di poco al di sotto o al di sopra della media. Questi impiegati, infatti, hanno un livello di soddisfazione minore o uguale a 0.5, un numero di progetti minore o uguale a 2.5, una valutazione maggiore di 0.4, un numero di ore lavorative giornaliere maggiore di 5.8 e hanno trascorso nella compagnia da 2.5 a 4.5 anni. Dall'altro vi sono 595 impiegati che hanno valutazione, numero di ore lavorative giornaliere, numero di anni trascorsi nella compagnia e livello di soddisfazione superiori rispetto alla media. Essi hanno, infatti, valutazione maggiore di 0.8, numero di ore lavorative maggiore di 9.9, numero di anni trascorsi nella compagnia compreso tra 4.5 e 6.5, soddisfazione maggiore di 0.7.

4.2 Random Forest

Il secondo modello usato per la classificazione è Random Forest che combinato con RandomizedSearchCV (nome del metodo usato da python) permette di addestrare diversi decision trees usando per ognuno una combinazione di parametri tra quelli forniti e di restituire il modello che ha fornito i migliori risultati.
Anche in questo caso, prima di procedere con l'addestramento del modello è stato necessario trasformare le variabili categoriche in numeriche di tipo binario e si è scelto di escludere dalle features quelle riguardanti il dipartimento di appartenenza di ogni impiegato, mantenendo invece, *satisfaction_level*, *last_evaluation*, *number_project*, *promotion_last_5years*, *average_daily_hours*, *time_spent_company*, *work_accident*, *salary_val_high*, *salary_val_low*, *salary_val_medium*.
La valutazione dei modelli restituiti dalla ricerca è stata fatta con holdout e cross validation.

4.2.1 Validazione tramite test e training set su dataset non bilanciato e bilanciato

I valori di *accuracy* sul test set riguardanti il decision tree selezionato attraverso Random Forest per il dataset sbilanciato sono rispettivamente 98% con validazione holdout e 98% con cross validation.
La seguente tabella mostra i valori di *precision*, *recall* e *f1-measure* relativi al test set per l'attributo target *left*=0 e *left*=1 ottenuti con holdout.

ValidationTarget	left	Precision	Recall	F1-measure	Support
Holdout	0	98%	100%	99%	3818
	1	99%	92%	95%	1132

Tabella 8: Statistiche Random Forest con dataset non bilanciato.

I risultati del modello selezionato non sono molto diversi da quelli ottenuti direttamente con decision tree (4.1.2).
I valori di *accuracy* sul test set riguardanti il decision tree selezionato attraverso Random Forest per il dataset bilanciato sono rispettivamente 95% con validazione holdout e 96% con cross validation.

La seguente tabella mostra i valori di *precision*, *recall* e *f1-measure* relativi al test set per l'attributo target *left*=0 e *left*=1 ottenuti con holdout.

ValidationTarget	left	Precision	Recall	F1-measure	Support
Holdout	0	92%	99%	95%	1171
	1	99%	92%	95%	1186

Tabella 9: Statistiche Random Forest con dataset bilanciato.

Rispetto ai risultati ottenuti con decision tree (4.1.3) si può notare, oltre che una lievissima crescita dell'accuratezza sul test (dal 94% al 95%), una crescita della *precision* per *left*=1 a cui corrisponde, tuttavia, la diminuzione di *precision* per *left*=0, mentre la *recall* subisce un andamento opposto.

4.3 Discussione sul miglior modello di predizione

Possiamo concludere che relativamente al tipo di modello da usare per la classificazione, per il dataset sbilanciato non si sono registrate significative variazioni nelle misure di valutazione con l'uso di Random Forest invece che di Decision tree.
Un lieve innalzamento dell'accuratezza si è, invece registrato con l'uso di Random Forest per il dataset bilanciato. A esso è corrisposto un aumento della *precision* per *left*=1.

Conclusioni

In conclusione, in questo progetto abbiamo compreso come è formato il database di Human Resources e quali siano le caratteristiche più importanti per classificare un impiegato.
Fin dall'analisi delle correlazioni tra le variabili, passando per gli algoritmi di clustering k-means e DBscan ed ottenendo ulteriori conferme dalle regole di associazione e dai decision trees, siamo riusciti ad estrarre un marcato gruppo di attributi che descrivono una categoria di lavoratori che lasciano il lavoro. In particolare questi si distinguono per un basso livello di soddisfazione, minore di 0.5, un numero minimo di progetti, ossia 2 o massimo 3, una valutazione inferiore alla media, con valori che oscillano tra 0.4 e 0.6, ed hanno passato tra i 2 e i 4 anni nell'azienda lavorando in media tra le 6 e le 7.5 ore al giorno.
Se quindi l'azienda riscontra un impiegato che presenta queste caratteristiche e desidera che mantenga il proprio posto di lavoro dovrà prendere provvedimenti per cercare, ad esempio, di capire come mai il livello di soddisfazione sia basso oppure perché dopo pochi anni nell'azienda una persona decida di cambiare lavoro.
I decision trees, inoltre, hanno evidenziato la presenza di un'altra categoria di impiegati che lasciano la compagnia. Si tratta di un gruppo di persone che hanno una valutazione superiore alla media, ovvero maggiore di 0.8, che hanno passato nella compagnia un numero di anni maggiore della media, tra i 4 e i 7, lavorando più della media, 10 o più ore al giorno, e realizzando un numero di progetti molto vicino alla media o superiore ad essa, più di 3.5. Per questa categoria, inoltre, anche il livello di soddisfazione, maggiore di 0.7, è superiore alla media.
Lo scopo dell'analisi era di individuare perchè gli impiegati migliori decidessero di lasciare l'azienda. Nel corso della nostra analisi abbiamo individuato un gruppo di persone che lasciano aventi valutazioni mediocri, e quindi che magari l'azienda non è interessata a tenere, ed uno aventi valutazioni molto alte, che potrebbe coincidere con il target iniziale.