UNIVERSITÀ DI PISA

Relazione progetto di Data Mining 1

IBM-HR Analytics Employee Attrition & Performance

Biviano Matteo
Graziani Alice
Segurini Gloria

Anno Accademico 2020/2021

Indice

1 Introduzione

IBM-HR è un dataset creato dagli operatori del Dipartimento delle Risorse Umane (HR) di IBM con lo scopo di indagare sui fattori che possono portare al logoramento dei dipendenti e quindi al loro possibile abbandono del posto di lavoro. Queste valutazioni, per esempio, possono essere basate sulle relazioni tra la distanza casa-lavoro o sul reddito mensile. In ambito aziendale è appunto il Dipartimento delle Risorse Umane che si occupa di queste analisi: attraverso opportuni modelli di Machine Learning, sarà possibile predire il logoramento dei dipendenti dell'organizzazione in modo tale da attrarre e trattenere i soggetti più talentuosi.
Questa analisi potrà infatti rappresentare una chiave per il successo aziendale.

2 Data Understanding & Data Preparation

Il dataset contiene 1470 record e 33 features, divisi in due gruppi **Train_HR_Employee_Attrition** e **Test_HR_Employee_Attrition**, i quali sono stati uniti in un unico dataset, ai fini della fase di Understanding. In questa sezione vengono riportate le analisi (qualitative e non) effettuate sulle features, le quali descrivono il background e le caratteristiche di ciascun dipendente.

2.1 Semantica dei dati

Al fine di familiarizzare con l'ambito di ricerca sono state analizzate le variabili presenti nel dataset, riportandone i risultati in Figura [1]. In essa è possibile notare come nel dataset siano presenti 3 tipi di dati: *continuo, categorico, ordinale.*

Nome	Tipologia	Descrizione	N° Distinct Values
Age	Numerico	Età del dipendente	43
Attrition	Booleano	Variabile target, indica se il dipendente ha lasciato l'azienda: { Yes, No}	2
BusinessTravel	Categorico	Con che frequenza viaggia il dipendentei: {Travel_Rarely, Travel_Frequently, Non-Travel}	3
DailyRate	Numerico	Tasso di retribuzione giornaliero del dipendente	886
Department	Categorico	Dipartimento del dipendente: {Research & Development, Sales, Human Resources}	3
DistanceFromHome	Numerico	Distanza casa-lavoro del dipendente	29
Education	Ordinale	Livello di istruzione del dipendente	5
EducationField	Categorico	Campo di istruzione del dipendente: {Medical, Life Sciences, Technical Degree, Other, Human Resources, Marketing}	6
EnvironmentSatisfaction	Ordinale	Livello di soddisfazione del dipendente rispetto all'ambiente di lavoro	4
Gender	Categorico	Sesso del dipendente: {Male, Female}	2
HourlyRate	Numerico	Tasso di retribuzione oraria del dipendente	71
JobInvolvement	Ordinale	Livello di coinvolgimento lavorativo del dipendente	4
JobLevel	Ordinale	Livello di esperienza professionale del dipendente	5
JobRole	Categorico	Ruolo lavorativo specifico del dipendente: {Research Director, Manager, Sales Executive, Laboratory Technician, Sales Representative, Manufacturing Director, Healthcare Representative, Human Resources}	9
JobSatisfaction	Ordinale	Livello di soddisfazione del dipendente riguardo il lavoro	4
MaritalStatus	Categorico	Stato civile dei dipendente: {Single, Divorced, Married}	3
MonthlyIncome	Numerico	Reddito mensile del dipendente	1112
MonthlyRate	Numerico	Tasso di retribuzione mensile del dipendente	1427
NumCompaniesWorked	Numerico	Numero di aziende per cui i dipendenti avevano lavorato in precedenza	10
Over18	Booleano	Indica se il dipendente ha raggiunto la maggiore età	1
OverTime	Booleano	Indica se il dipendente ha svolto lavoro straordinario	2
PercentSalaryHike	Numerico	Percentuale di aumento salariale del dipendente	15

PerformanceRating	Ordinale	Valutazione delle prestazioni del dipendente, condotte dal manager	2
RelationshipSatisfaction	Ordinale	Soddisfazione del dipendente per la relazione lavorativa	4
StandardHours	Numerico	Quante ore standard lavora il dipendente	1
StockOptionLevel	Ordinale	Livello di stock option del dipendente	4
TotalWorkingYears	Numerico	Anni di lavoro totali del dipendente	40
TrainingTimesLastYear	Numerico	Ore totali spese dal dipendente per l'aggiornamento lo scorso anno	7
WorkLifeBalance	Ordinale	Valutazione del dipendente per l'equilibrio tra lavoro e vita privata	4
YearsAtCompany	Numerico	Anni di lavoro del dipendente in azienda	36
YearsInCurrentRole	Numerico	Anno lavorativo del dipendente nel ruolo attuale	19
YearsSinceLastPromotion	Numerico	Anni dall'ultima promozione	16
YearsWithCurrManager	Numerico	Anni con l'attuale manager	18

Figura 1: Semantica delle features

2.2 Distribuzione delle variabili e statistiche

Allo scopo di esplorare completamente tutti i tipi di features e trovare eventuali relazioni, sono state eseguite tecniche di visualizzazione dei dati per ogni tipo di feature. Nel dataset, i record rispetto alla variabile target **Attrition** risultano essere sbilanciati, caratteristica che potrebbe avere effetti sull'accuratezza dei predittori. Come mostrato in Figura [2] (a sinistra), il numero di dipendenti che hanno abbandonato l'azienda (pari al 16.1%) è di fatto inferiore al numero di volte in cui ciò non è accaduto (pari all'83.9%). Anche altri attributi risultano distribuiti in modo sbilanciato, caratteristica che può influenzare la distribuzione di **Attrition=Yes** rispetto ai valori delle classi. Un esempio è il caso di **BusinessTravel** (a destra di Figura [2]) dove è possibile notare come vi è un numero elevato di dipedenti che viaggiano raramente, classe che è risultata appunto avere il più alto tasso di abbandono.

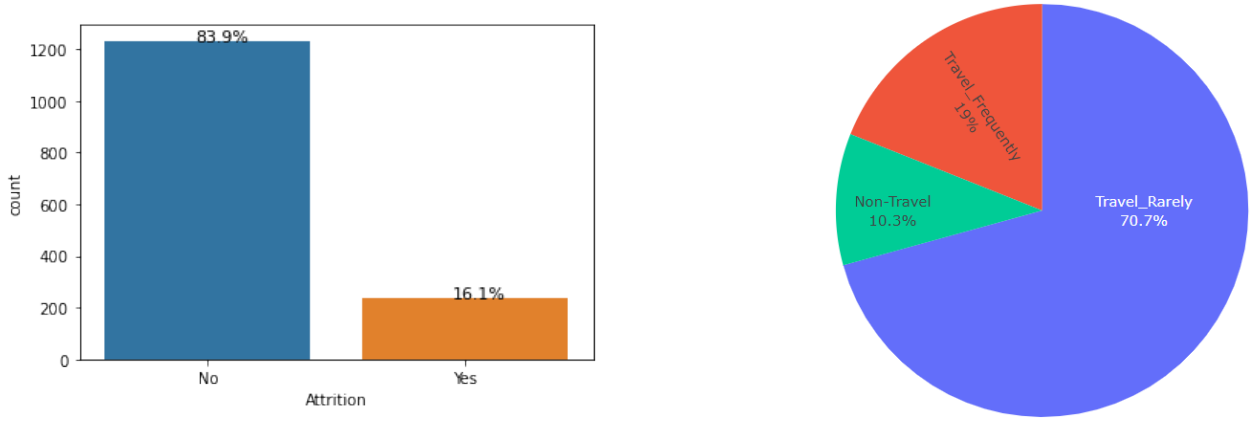


Figura 2: Distribuzione di Attrition e di BusinessTravel

Dalla Figura [3] (a sinistra) è possibile notare che, indipendentemente dallo stato civile, sono molti i dipendenti che non abbandonano l'azienda. Dal plot si evince che l'abbandono è una caratteristica più pronunciata per i dipendenti *Single* per tutte le fasce d'età, stessa cosa per i dipendenti sposati con uno stipendio più basso. Mentre le persone divorziate non presentano un alto tasso di abbandono.
Confrontando (a destra di Figura [3]) la distribuzione di **DistanceFromHome** per i due valori della variabile target è visibile come un maggior numero di dipendenti risiede vicino al luogo di lavoro, con conseguenti livelli di abbandono inferiori. Mentre, con l'aumento della distanza da casa, la curva di abbandono supera la curva di non abbandono, come ci si sarebbe aspettato. Come la distanza dal posto di lavoro, anche gli straordinari (definiti da **OverTime**), potrebbero essere un indice di abbandono: il relativo tasso è elevato per i dipendenti che hanno svolto straordinari, specialmente se con un reddito mensile basso.

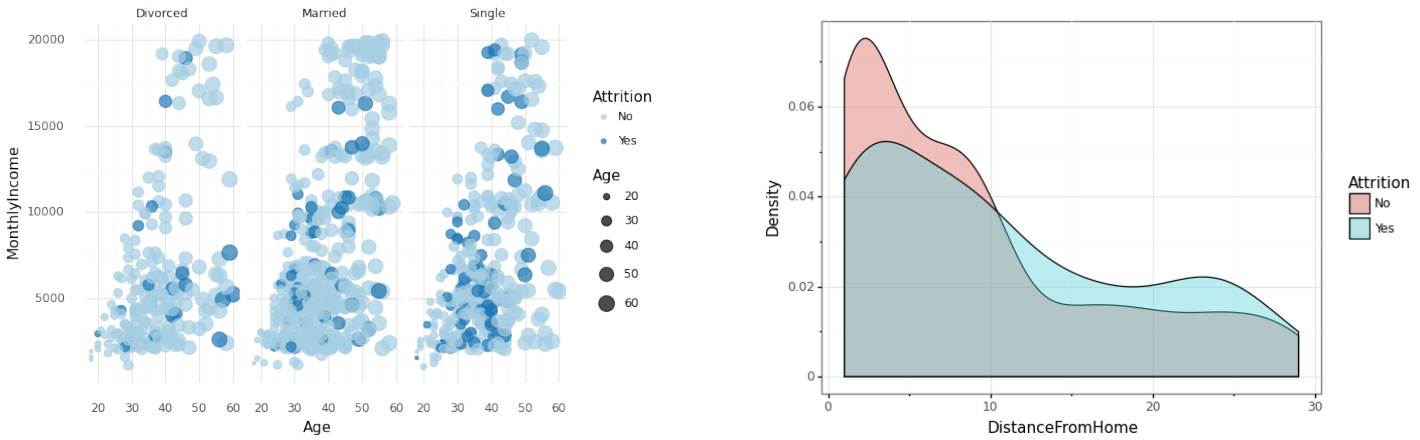


Figura 3: Livelli di Attrition rispetto a MaritalStatus - Distribuzione di Distance Home

La tabella (in Figura [4]) mostra le differenze di fondo tra chi lascia l'azienda e chi no. Dalle statistiche emerge che i dipendenti che abbandonano l'azienda hanno un reddito mensile ed una tariffa giornaliera inferiori rispetto agli altri. Inoltre è più probabile che un dipendente lasci il suo ruolo in azienda se abita lontano (come già esposto). Le features come *TotalWorkingYears*, *YearsAtCompany*, *YearsCurrentRole* e *YearsWithCurrManager*, mostrano che i dipendenti che lavorano al servizio dell'azienda per un periodo breve hanno maggiori probabilità di lasciare l'azienda, mentre i dipendenti che lavorano più a lungo hanno meno probabilità di abbandonare il proprio posto.

Nome	Age	DailyRate	Distance FromHome	HourlyRate	Total WorkingYears	Training TimesLastYear	Years AtCompany	YearsIn CurrentRole	MonthlyRate	NumCompanies Worked	PercentS alaryHike	YearsSince LastPromotion	YearsWith CurrManager	MonthlyIncome
Min	18	102	1	30	0	0	0	0	2.094	0	11	0	0	1.009
Max	60	1.499	29	100	40	6	40	18	26.999	9	25	15	17	19.999
Mean (Attrition = Yes)	36,813	750,362	10,633	65,573	8,245	2,717	7,291	2,903	14.559,308	2,941	15,097	1,945	2,852	6.542,410
Mean (Attrition = No)	37,173	812,504	8,916	65,952	11,863	2,827	6,876	4,484	14.265,779	2,646	15,231	2,234	4,368	6.550,137
Mean	37,115	802,486	9,193	65,891	11,280	2,811	6,942	4,229	14.313,103	2,693	15,210	2,188	4,123	6.548,916
Std (Attrition = Yes)	8,530	401,051	8,435	20,058	7,154	1,248	5,734	3,168	7.192,930	2,673	3,762	3,146	3,137	4.302,511
Std (Attrition = No)	9,163	403,045	8,009	20,372	7,758	1,310	6,084	3,648	7.099,380	2,459	3,638	3,233	3,593	4.806,891
Std	9,065	403,372	8,104	20,323	7,778	1,301	6,031	3,622	7.115,365	2,497	3,659	3,221	3,567	4.730,786

Figura 4: Statistiche variabili numeriche

2.3 Data Quality

In questa fase è stato analizzato il dataset al fine di controllare la presenza di dati duplicati e l'eventuale gestione di *Missing values* ed *Outliers*. In particolare, è stata presa in considerazione l'impossibilità di eliminare record a causa del dataset già di per sè ridotto e sbilanciato.

2.3.1 Missing Values

La distribuzione dei missing values presente nel dataset è rappresentata in Figura [5] (a sinistra), nella quale vengono mostrate le 9 features che presentano *Missing values*. Tra queste, **Over18** e **StandardHours** presentano il maggior numero di valori mancanti (468 e 717 rispettivamente). Inoltre, dalla Figura, è visibile che almeno 3 colonne su 9 non presentano mai contemporaneamente missing values, infatti è risultato che il massimo numero di attributi per cui simultaneamente mancano valori è 6, numero che giustifica ulteriormente la scelta di non eliminare record. I missing values sono stati trattati come appartenenti alla tipologia **MCAR** (Missing completely at random), infatti, come si evince a destra della Figura [5], la presenza o l'assenza del valore di un attributo non influisce sugli altri. La Heatmap mostra una leggera correlazione positiva tra **Gender** e **TrainingTimesLastYear**, che però potrebbe essere considerata casuale osservando il numero di valori mancanti non del tutto equilibrati (75 contro 292).

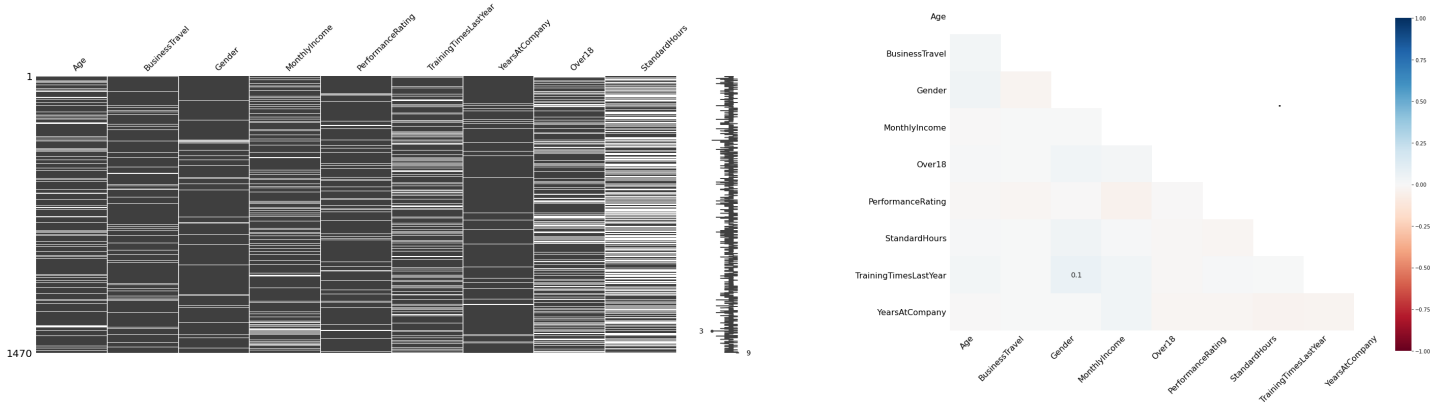


Figura 5: Distribuzione dei missing values - Nullity Correlation

Considerando il tipo di missing values per la loro gestione è stato scelto di adottare il metodo dell'imputazione semplice. Per individuare i valori mancanti è stato scelto di effettuare raggruppamenti con le features che meglio sembravano discriminare le variabili su cui effettuare l'imputazione.
Per gli attributi numerici (**Age**, **TrainingTimesLastYear**, **YearsAtCompany**) i valori mancanti sono stati quindi sostituiti con la media ricavata dal grouby di **Department** e **JobRole**; mentre per gli attributi categorici/ordinali (**PerformanceRating**, **Gender** e **BusinessTravel**) i valori mancanti sono stati ricavati tramite la moda ottenuta dal groupby di **Department**, **JobRole** e **WorkLifeBalance**.
Per quanto riguarda le features **Over18** e **StandardHours**, sono state eliminate come esplicitato in Sezione [2.5]

2.3.2 Gestione outliers ed errori

L'individuazione di valori *outliers* è stata effettuata attraverso la rappresentazione della distribuzione di ogni valore di tipo numerico in un *boxplot*. É emerso che gli attributi **MonthlyIncome**, **TotalWorkingYears**, **YearsAtCompany**, **YearsSinceLastPromotion** hanno un gran numero di istanze al di sopra del *valore adiacente superiore*. I valori riportati nel grafico in Figura [6] sono stati standardizzati attraverso il metodo *RobustScaler* (robusto agli outliers) affinché tutti gli attributi fossero rappresentati secondo uno stesso intervallo, per facilitarne la lettura. Dall'analisi di tali outliers (al fine di comprendere se trattarli come anomalie da eliminare o meno) è risultato che 655 records hanno soddisfatto una delle seguenti condizioni:
1) TotalWorkingYears < YearsAtCompany,
2) (Age - TotalWorkingYears) < 16.

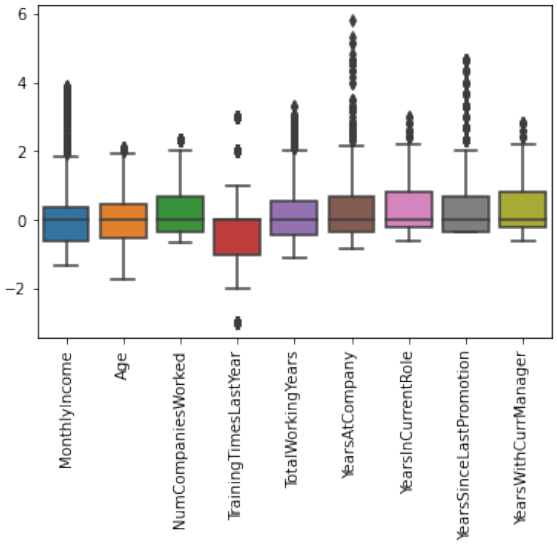


Figura 6: Outliers

Le condizioni identificano rispettivamente dipendenti che hanno lavorato per la compagnia un numero di anni maggiore del totale di anni lavorativi; dipendenti con un numero di anni di lavoro superiore all'età; dipendenti che hanno lavorato ad un'età inferiore a 16 anni. Dato il numero elevato di questi record è stato scelto di mantenerli all'interno del dataset, aggiungendo una marca identificativa racchiusa nel nuovo attributo **NotValid** (Se **NotValid = True** allora il record appartiene alla cerchia dei 655 precedenti). Non sono stati invece considerati in quest'ultima variabile errori *minori* quali:
1) YearsWithCurrManager > YearsAtCompany,
2) YearsSinceLastPromotion > YearsAtCompany,
3) YearsInCurrentRole > YearsAtCompany.
Identificati come *minori* sia perchè in quantità molto inferiore ai precedenti, sia perchè potrebbero non essere errori in base ad eventuali specifiche (non riportate dalla sorgente) sulle politiche aziendali IBM.

Ad esempio, a causa di una fusione d'aziende o di un consorzio un dipendente potrebbe essere stato sotto lo stesso Manager per un numero di anni superiori al numero di anni effettivi sotto un'azienda (formalmente diversa dalla precedente dopo la fusione).

2.4 Trasformazione delle variabili

Considerando le diverse tipologie di task che sarà necessario realizzare, è stato ritenuto utile in questa fase iniziale di Understanding, non eseguire trasformazioni preliminari, ma trasformare i dati a seconda dei diversi algoritmi che verranno applicati.

2.5 Correlazione attributi ed eliminazione variabili ridondanti

In prima istanza sono stati eliminati dal dataset gli attributi **Over18** e **StandardHours** perchè contengono un unico valore distinto e tanti missing values. Successivamente, è stata calcolata la correlazione tra le coppie di attributi del dataset. In Figura [7] (a sinistra) viene mostrata la *HeatMap* (calcolata con la regola di Spearman) dei soli attributi con una correlazione superiore al 30%. É possibile notare come molti attributi siano scarsamente correlati, mentre le coppie di attributi **(TotalWorkingYears, JobLevel)** e **(YearsWithCurrManager, YearsInCurrentRole)** siano altamente correlate (seppur inferiore all'80%). La lieve correlazione tra **Age**, **MonthlyIncome** ed **YearsAtCompany**, rispetto al target, è visibile nel grafico a destra in Figura [7].

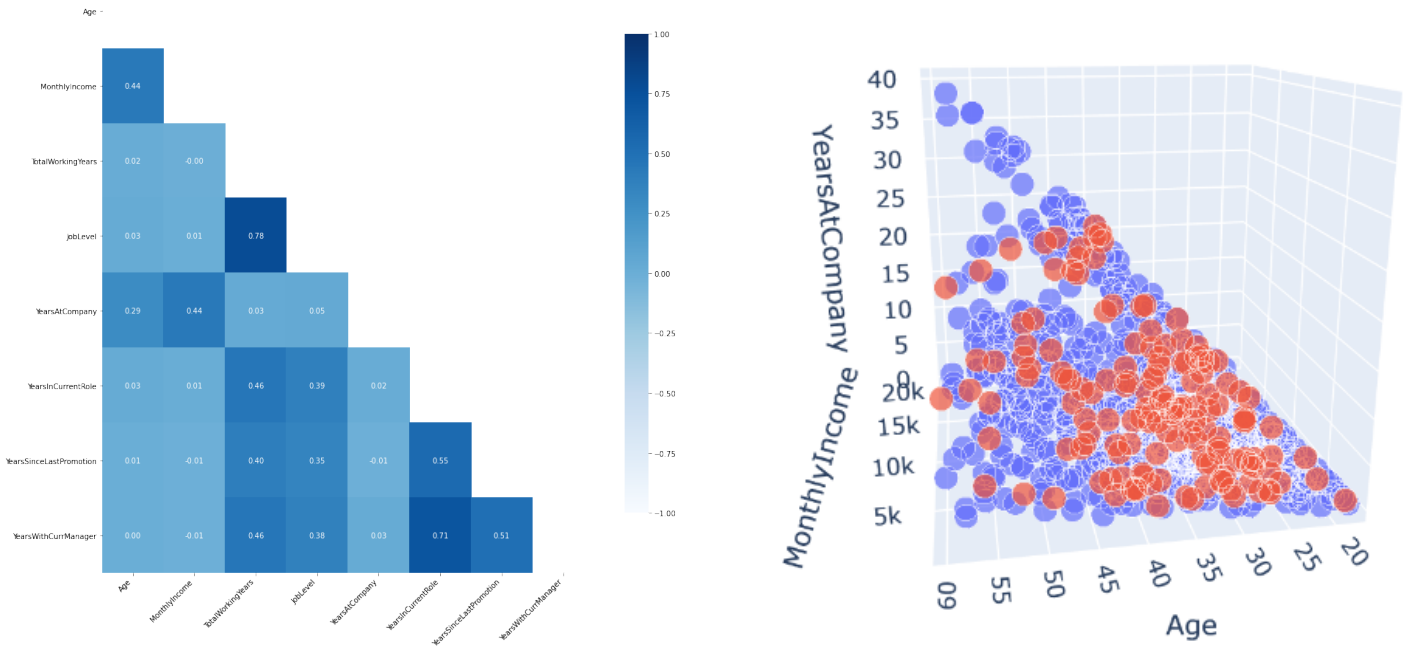


Figura 7: Correlation - Scatterplot (Age, MonthlyIncome, Company)

Dalle analisi precedenti è stato scelto di inserire due nuove features: **TotalSatisfaction** ed **ProbablyLeave**. La prima è determinata dalla somma dei quattro fattori di soddisfacimento **EnvironmentSatisfaction**, **JobInvolvement**, **JobSatisfaction**, **RelationshipSatisfaction**. Il valore massimo della variabile (pari a 16) indicherà *soddisfacimento massimo* del dipendente, mentre il valore minimo (pari a 5) indicherà *insoddisfazione*. Questo poichè, all'aumentare del grado di soddisfazione, la percentuale di abbandono del cliente diminuisce (fino ad annullarsi per soddisfazione pari a 16). La seconda feature invece assume valore 1 se il **MonthlyIncome** del dipendente è inferiore al **MonthlyIncome** medio del **Department** a cui appartiene quel dipendente e il suo **PercentSalaryHike** è inferiore alla media dei dipendenti (pari a 15). Quest'ultima variabile con il valore 1 indica che il dipendente è più probabile che abbandoni.

3 Clustering

In questa sezione vengono descritti i tre algoritmi di clustering (*KMeans*, *DBScan*, *Hierarchical*) applicati al dataset ed i rispettivi risultati. Per l'applicazione di tutti gli algoritmi, le features numeriche sono state normalizzate con *RobustScaler* in modo tale che gli "outliers" individuati in Sezione 2.3.2 influenzassero il meno possibile i risultati (specialmente nel caso del *KMeans*).

3.1 Analisi dei cluster attraverso l'algoritmo K-Means

3.1.1 Selezione attributi

Considerando che il dataset contiene informazioni riguardo le caratteristiche di un dipendente è stato scelto di utilizzare tra gli attributi numerici i 6 seguenti, i quali rappresentano un dipendente nei vari ambiti di analisi: **Age**, **DistanceFromHome**, **MonthlyIncome**, **TotalWorkingYears**, **YearsAtCompany**, **TotalSatisfaction**.

3.1.2 Identificazione del miglior valore di k

Al fine di identificare il miglior parametro **k** per il *K-Means*, è stato usato il metodo *Elbow* calcolando l'SSE per $k \in [2, 28]$. Il grafico in Figura [8] (a sinistra) mostra le variazioni dell'SSE rispetto al **k** considerato. Non potendo visualizzare dal solo grafico il valore per il quale si verifica una maggiore diminuzione del valore di SSE, è stato utilizzato il metodo *KneeLocator*, il quale ci ha permesso di identificare il valore $\mathbf{k = 9}$. A destra, Figura [8] è possibile invece visualizzare, per gli stessi valori di k, il confronto tra Silhouette e tempo di calcolo, ottenendo come miglior valore $\mathbf{k = 7}$. Per questi motivi è stato scelto come valore migliore per il KMeans, $\mathbf{k = 8}$ con cui si ottengono $\mathbf{SSE = 3293.917}$ e $\mathbf{silhouette = 0.154}$. Un valore di silhouette migliore (benchè sempre inferiore a 0.5) è stato ottenuto da altre combinazioni di attributi, ad esempio utilizzando le combinazioni *banali*: {YearsAtSinceLastPromotion, YearsInCurrentRole, YearsWithCurrentManager} oppure {JobLevel, JobInvolvement, JobSatisfaction}. Tuttavia, è stato deciso di mantenere questa configurazione (con bassa silhouette) in quanto ha permesso di trovare (come spiegato in Sezione [3.1.3]) relazioni tra il numero di clusters e gli errori presenti nel dataset.

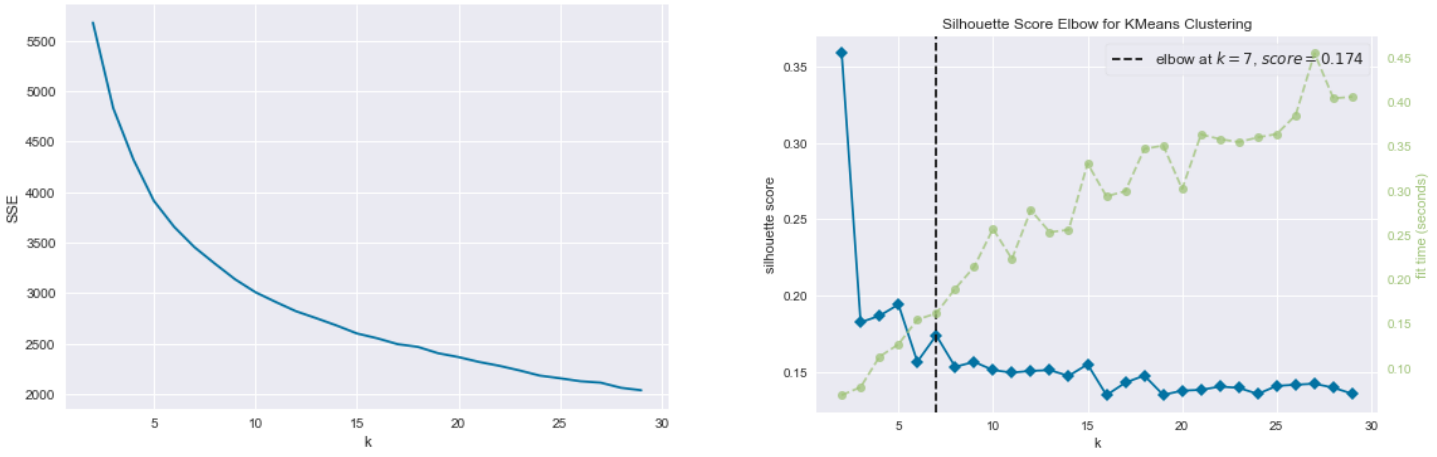


Figura 8: SSE - Silhouette

3.1.3 Caratterizzazione dei cluster ottenuti

I clusters sono stati analizzati sia in termini delle features sopra dichiarate sia in termini degli "errori" specificati in Sezione [2.3.2]. Si è notato infatti, che alcuni clusters, "simili" in termini di valori complessivi delle features si differenziano quasi del tutto per gli "errori" che vi appartengono. A tale scopo, vengono riportati (in Figura [9 - 10] i clusters 1 e 4, in cui è possibile ritrovare queste considerazioni. Per una visualizzazione più chiara sono stati utilizzati grafici a coordinate parallele, anche rispetto a tre variabili d'errore: **IllegalWorking** (dipendenti con

$YearsAtCompany > TotalWorkingYears$), **ChildWorking** (dipendenti con $0 < (Age - TotalWorkingYears) < 16$), **PreviousBirthWorking** (dipendenti con $(Age - TotalWorkingYears) <= 0$).

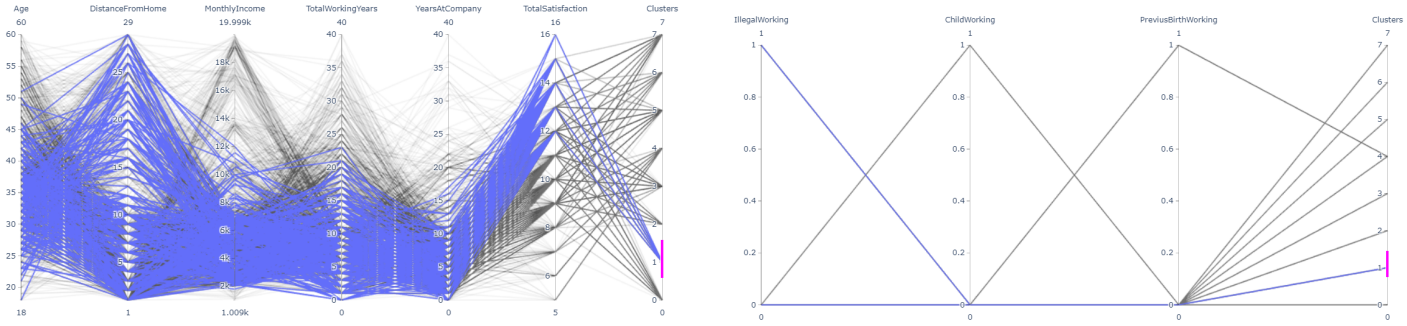


Figura 9: KMeans - Cluster 1

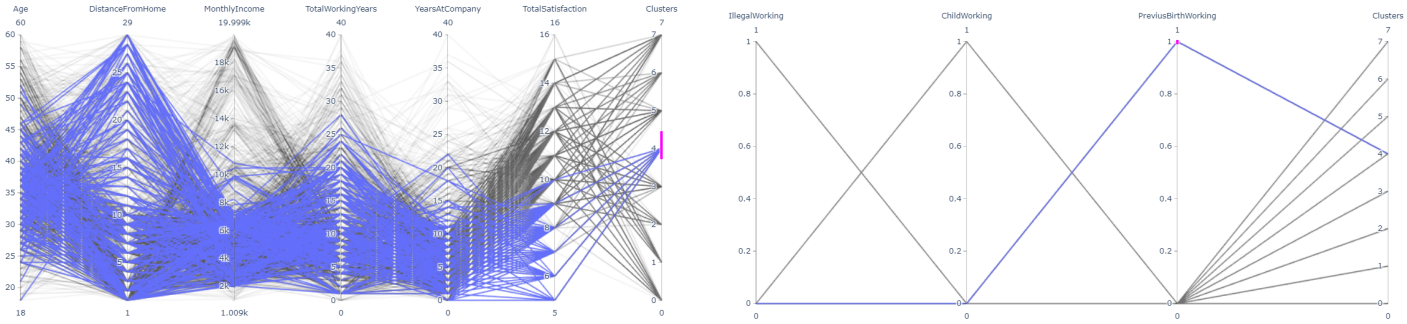


Figura 10: KMeans - Cluster 4

I due clusters, che rappresentano dipendenti con un reddito mensile medio-basso, si differenziano per **TotalSatisfaction** e per gli "errori" contenuti in essi. Il **cluster 1** contiene infatti dipendenti con un indice di soddisfazione complessiva alto (superiore a 12) e si caratterizza per il fatto di avere al suo interno dipendenti che hanno lavorato *in nero*, oltre a dipendenti regolari; mentre il **cluster 4** è rappresentato da dipendenti con una soddisfazione medio-bassa (al di sotto di 10), parte dei quali risulta aver lavorato *prima della nascita*. In particolare, il cluster 4 è anche l'unico a contenere questa caratteristica, probabilmente in quanto tra gli errori, è il meno frequente.

3.1.4 Osservazioni

Come è possibile notare sia dai grafici in Figura [11] che dal valore basso di Silhouette dell'algoritmo, i cluster non risultano ben separati. Questa proprietà è probabilmente dovuta alle caratteristiche intrinseche del dataset, discusse in Sezione [2].

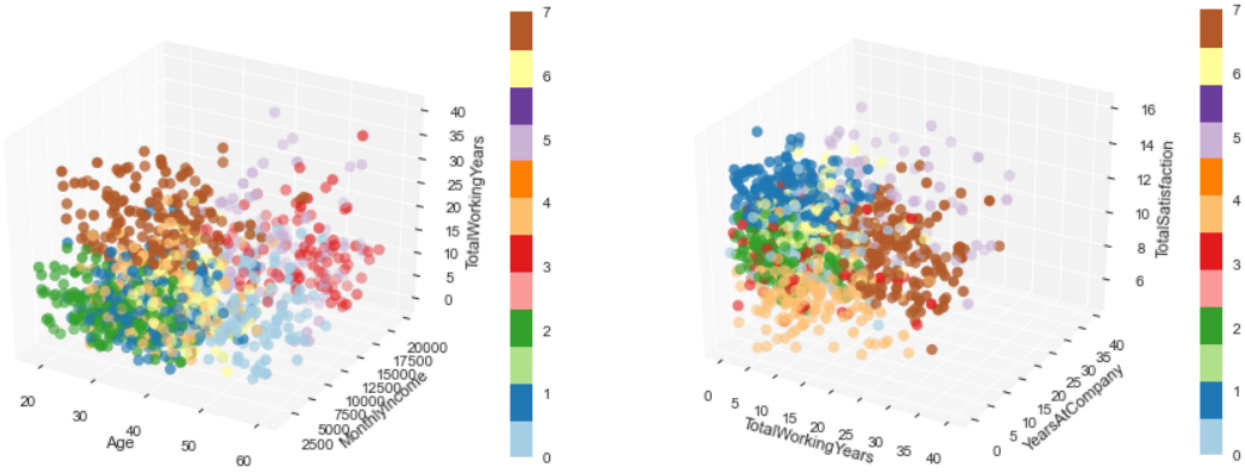


Figura 11: KMeans - 3DPlot

3.2 Analisi dei cluster attraverso algoritmi basati sulla densità

3.2.1 Scelta degli attributi e della funzione di distanza

In modo da ottenere un confronto migliore tra algoritmi, per effettuare il clustering è stato scelto di testare il dataset con gli stessi attributi della precedente sezione, utilizzando la funzione di distanza euclidea (non avendo comunque ottenuto risultati migliori con altre tipologie di distanze).

3.2.2 Studio dei parametri di clustering

Dato che il DBScan non si preoccupa molto della forma dei clusters, l'utilizzo di una metrica come la Silhouette (usata in Sezione [3.1.2] per scegliere il **k** che restituiva cluster più densi ed isolati possibili) potrebbe essere inefficiente per la scelta dei due parametri cruciali dell'algoritmo: **Epsilon** e **MinPoints**. Per risolvere questo problema sono stati utilizzati due tipi di metodi per la valutazione della clusterizzazione al variare degli iperparametri.
I due metodi sono:
- 5-NearestNeighbor: considera la distanza media tra i *noise points* e i 5 punti più vicini (provando anche diversi valori di k, non sono state riscontrate differenze sostanziali).
- Numero di clusters che vengono individuati.

Per queste metriche, in Figura [12] vengono mostrati i risultati per **Epsilon** $\in [0.1, 1.9]$ e **MinPoints** $\in [5, 45]$. É possibile notare come per valori molto bassi di ϵ, praticamente tutti i punti vengono identificati come *noise points* (0 clusters). Viceversa, per valori molto alti si tende ad avere un solo cluster. É stata scelta la combinazione (*Epsilon = 1.1 MinPoints = 10*) perchè produce non più di tre clusters (aumentandone il numero infatti si ottengono clusters sempre più piccoli e meno significativi) e mediamente restituisce dei noise points più distanti (a parità di N).

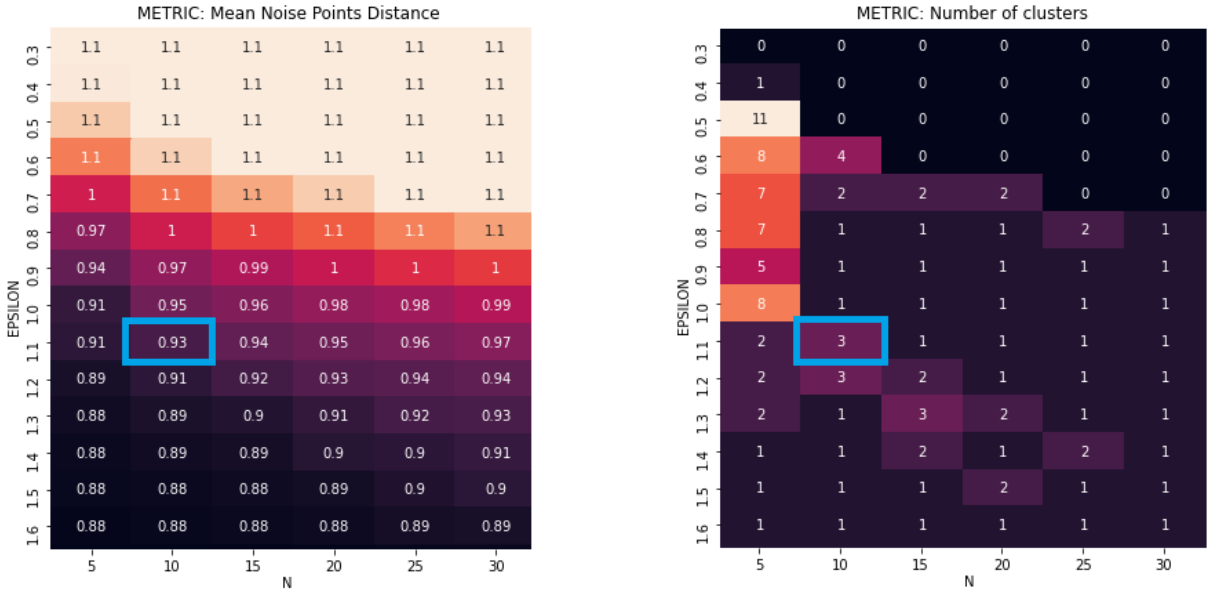


Figura 12: DBScan - Tuning Phase

3.2.3 Caratterizzazione e interpretazione dei cluster ottenuti

Nonostante sia stato scelto un numero di clusters inferiore rispetto al KMeans, l'algoritmo non ha restituito risultati particolarmente significativi. É emerso che il 20.8% dei dati è stato identificato come outliers, mentre i clusters sono risultati così distribuiti: {0: 77.6, 1: 0.88, 2: 0.72}.
Le differenze sostanziali dei due cluster più piccoli rispetto al cluster 0 sono le seguenti:

- Cluster 1: composto dai dipendenti con un'età superiore ai 35 anni e numero di anni in azienda maggiore di 20.
- Cluster 2: composto dai dipendenti che abitano ad una distanza superiore ai 10 km dal posto di lavoro.

Non sono stati ottenuti risultati più significativi provando altre combinazioni di features o escludendo i record *errati*. Per quest'ultimi, l'unica differenza che è possibile notare (rispetto all'algoritmo precedente) dal grafico in Figura [13] (a sinistra) è che il cluster 2, tra i possibili errori, contiene unicamente **IllegalWorking**.

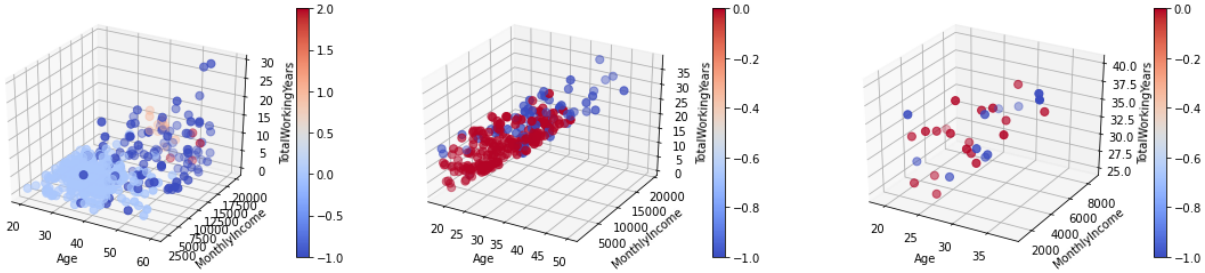


Figura 13: DBScan Error: IllegalWorking - ChildWorking - PreviusBirthWorking

3.3 Analisi mediante clustering gerarchico

3.3.1 Scelta degli attributi e della funzione di distanza

Per l'implementazione del modello sono stati utilizzati gli stessi attributi scelti per i modelli precedenti e come funzione di distanza quella euclidea, in modo tale da rendere più agevole il confronto dei risultati.

3.3.2 Mostrare e discutere diversi dendogrammi utilizzando algoritmi diversi

L'algoritmo è stato eseguito con diverse configurazioni, variando:
- Il criterio di connessione tra i clusters: single, ward, average e complete.
- L'uso di una matrice di connettività (derivata dal grafo ottenuto dai primi 20 vicini di ogni nodo) per il calcolo delle distanze.

La scelta della dimensione dei clusters è stata effettuata attraverso la visualizzazione dei dendogrammi presenti in Figura [14 - 15].

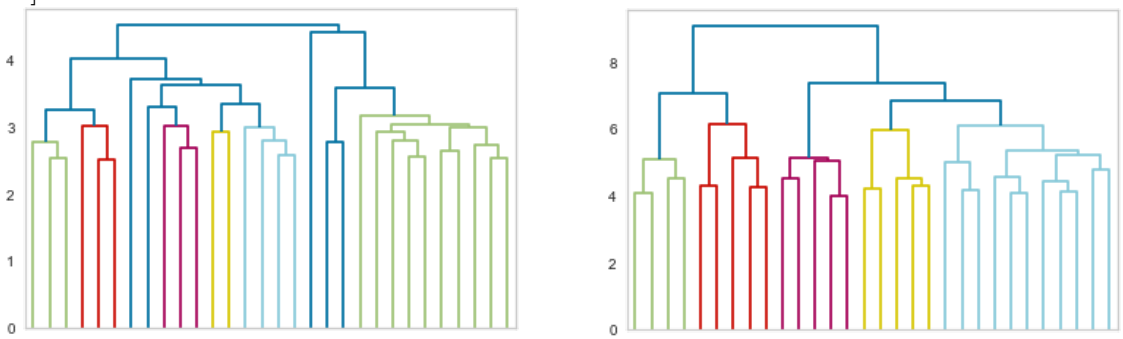


Figura 14: Dendogrammi: Average - Complete

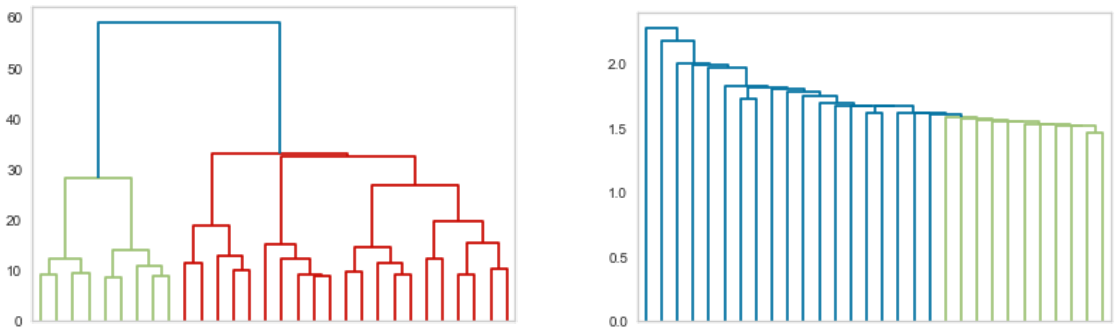


Figura 15: Dendogrammi: Ward - Single

Si può notare (come da aspettative) che utilizzando il criterio **Complete Linkage** i cluster vengono uniti a distanze doppie rispetto al criterio **Average** (che utilizza la media delle distanze invece del massimo), in quanto la media di qualsiasi distribuzione è sempre minore o al più uguale del massimo. Non è invece possibile confrontare la distanza a cui vengono uniti i clusters dal criterio **Ward**, con la distanza degli altri criteri, poichè quest'ultimo, utilizzando la varianza, assume valori su una scala diversa. Il **Single Link**, in particolare, dà risultati non soddisfacenti (come visibile dal dendogramma). Inoltre, i clusters ottenuti sono altamente sbilanciati, infatti per un taglio che genera 5 clusters è stata ottenuta la seguente configurazione di elementi: {0: 1466, 1: 1, 2: 1, 3: 1, 4: 1}. Per ogni configurazione provata è stato calcolato il coefficiente di silhouette per valutare la bontà dei parametri. In Figura [16] sono visibili i risultati ottenuti.

Metodo	Dimensioni dei clusters	Valore Silhouette
Average	75, 68, 12, 1293, 7, 1, 14	0.191
Average + Connectivity	5, 1452, 2, 1, 2, 3, 5	0.330
Complete	80, 919, 110, 18, 343	0.116
Complete + Connectivity	1450, 12, 1, 3, 4	0.398
Ward	330, 222, 352, 89, 123, 354	0.131
Ward + Connectivity	330, 222, 352, 89, 123, 354	0.131

Figura 16: Hierarchical - Confronto diversi criteri

É possibile notare come le configurazioni **Average** e **Complete Linkage** usate in combinazione con la matrice di connettività portano a pessime performance infatti, si ottiene un grande cluster con più del 98% dei dati. Questo problema è noto come "rich getting richer", secondo il quale è più facile che ai clusters grandi vengano aggregati altri clusters. Per il criterio **Ward**, invece, la performance osservata è identica usando o meno la matrice di connettività. Inoltre, vengono ottenuti clusters bilanciati, presentando quindi risultati migliori dei precedenti. Inoltre, i tre criteri (senza considerare il caso di matrice di connettività) hanno ottenuto, in termini di valori di silhouette, circa le stesse performance.

3.4 Confronto tra clustering ottenuti e valutazioni finali

In Figura [17] sono riassunti i risultati migliori degli algoritmi di clustering considerati. In particolare, non sono state riscontrate corrispondenze rilevanti fra i clusters ottenuti usando clustering gerarchico e i clusters ottenuti usando altri metodi. I risultati insoddisfacenti tratti dal DBScan sono coerenti con i risultati ottenuti con il criterio Single Link del clustering gerarchico. Il confronto finale è stato quindi effettuato tra il K-Means ed il clustering gerarchico. Dal contenuto di quest'ultimo, tuttavia, non sono state rilevate informazioni interessanti neanche riguardo la separazione degli errori inter-clusters. É stato quindi preferito come metodo il K-Means, in quanto metodo di clustering con il miglior trade-off tra valore di silhouette e significatività della distribuzione interna dei dati.

Algoritmo	Numero di Clusters	Valore Silhouette
K-Means	8	0.154
DBScan	3	0.235
Gerarchico	6	0.116

Figura 17: Riassunto Clustering

4 Classification

In questa sezione vengono illustrati due algoritmi di classificazione e le metodologie utilizzate per prevedere la variabile *Attrition* e i risultati ottenuti.

4.1 Pre-processing

Per poter eseguire la classificazione, il dataset è stato preprocessato trasformando gli attributi categorici con una codifica **One Hot Encoding**, mentre i dati numerici sono stati trattati con la stessa metodologia utilizzata nella sezione precedente.

4.2 Scelta degli iperparametri

É stata effettuata una fase preliminare di validation, in modo da scegliere gli iperparametri che massimizzassero le metriche **F1**, **Recall** e **ROC**. Questo perchè, a causa dello sbilanciamento del dataset (discusso in Sezione [2.2]), usare metriche come l'accuracy potrebbe comportare ottimi valori, ma modelli privi di informazioni. Inoltre, un valore più alto di **Recall** indica che il modello ottenuto potrebbe prevedere un abbandono dei dipendenti più realistico (scopo principale dell'analisi). In questa fase è stato utilizzato l'algoritmo "GridSearchCV", nel quale il numero di folds è ottenuto tramite una "Stratified 3-Fold Cross Validation". In Figura [18] vengono mostrati i gruppi di iperparametri utilizzati durante la fase di validazione dei classificatori.

DecisionTree	K-Nearest Neighbors
criterion : gini, entropy *max_depth* : None, 2, 5, 10, 15, 20 *min_samples_split* : 2, 5, 10, 15, 20 *min_samples_leaf* : 1, 5, 10, 15, 20	*n_neighbors* : range(1, 30) *weights* : uniform, distance *metric* : euclidean, manhattan, minkowski

Figura 18: Hyperparameter Tuning

4.3 Decision Tree

I risultati ottenuti dalla fase di validazione del classificatore, per le metriche specificate, sono mostrati attraverso le Heatmaps in Figura [19], nelle quali le configurazioni risultate migliori sono state evidenziate in rosso.

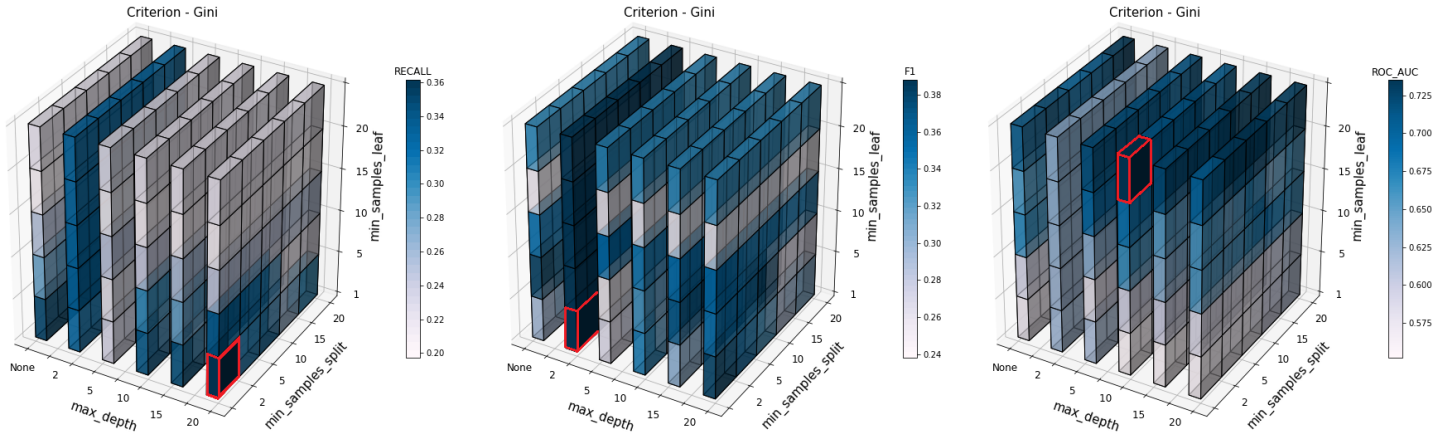


Figura 19: 3D Heatmaps risultati GridSearch

Per tali configurazioni degli iperparametri sono stati riportati in Figura [20] i valori delle tre metriche analizzate sia per l'insieme di training che per il validation. É possibile notare che per la configurazione di iperparametri: **Criterion = Gini**, **Max Depth = 20**, **Min Samples Split = 2**, **Min Samples Leaf = 1** il classificatore si adatta eccessivamente all'insieme di Training. L'**Overfitting** del classificatore per questa particolare configurazione è visibile graficamente in Figura [21]. Per le altre due configurazioni, invece, le curve d'apprendimento sul training e sul validation set tendono a coincidere (per l'insieme che ottimizza l'**F1** questo accade dopo aver visto i primi 200 esempi). Per questi problemi è stato deciso di eseguire il test del modello con la configurazione: **Criterion = Gini**, **Max Depth = 5**, **Min Samples Split = 2**, **Min Samples Leaf = 20**, poichè priva di Overfitting e con discrete performance tra **F1** e **ROC**.

		Optimal Recall	Optimal F1	Optimal ROC
Criterion		Gini	Gini	Gini
Max Depth		20	2	10
Min Samples Split		2	2	2
Min Samples Leaf		1	1	20
Recall	**Training**	1.000	0.319	0.244
	Validation	0.362	0.318	0.232
F1	**Training**	1.000	0.413	0.357
	Validation	0.349	0.388	0.318
ROC	**Training**	1.000	0.714	0.889
	Validation	0.590	0.666	0.735

Figura 20: DecisionTree: Risultati ottenuti per Training e Validation

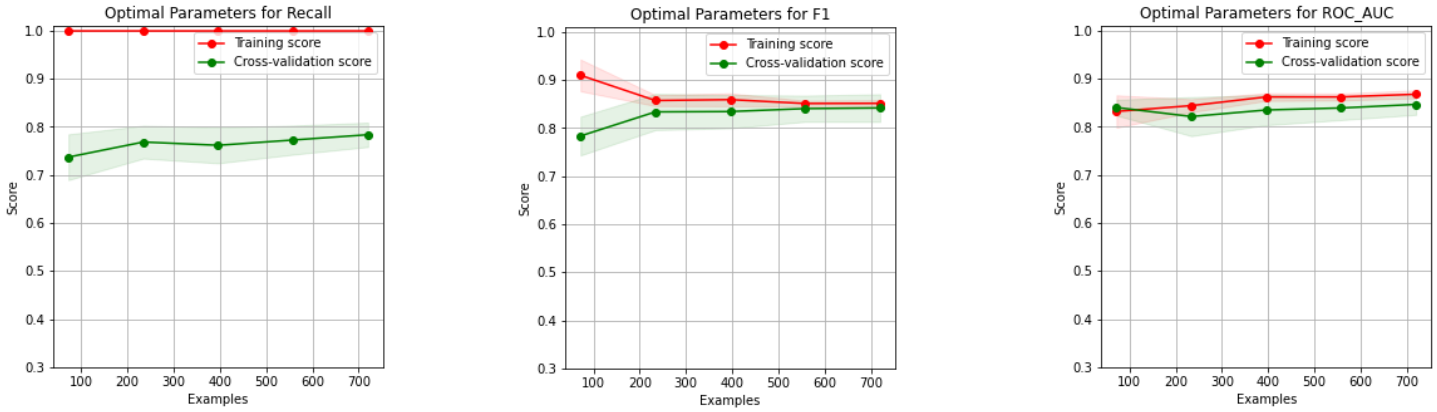


Figura 21: DT: Curve di apprendimento per le configurazioni ottenute

Per il modello scelto è stato ottenuto lo schema in Figura [22], nel quale è possibile notare che (come ipotizzato in Sezione [2.2]) l'aver fatto straordinari (**OverTime = Yes**) fornisce informazioni interessanti ai fini della classificazione. Inoltre, i dipendenti che hanno un basso numero di anni di lavoro in azienda o che sono *Single* hanno più probabilità di abbandonare il posto di lavoro.

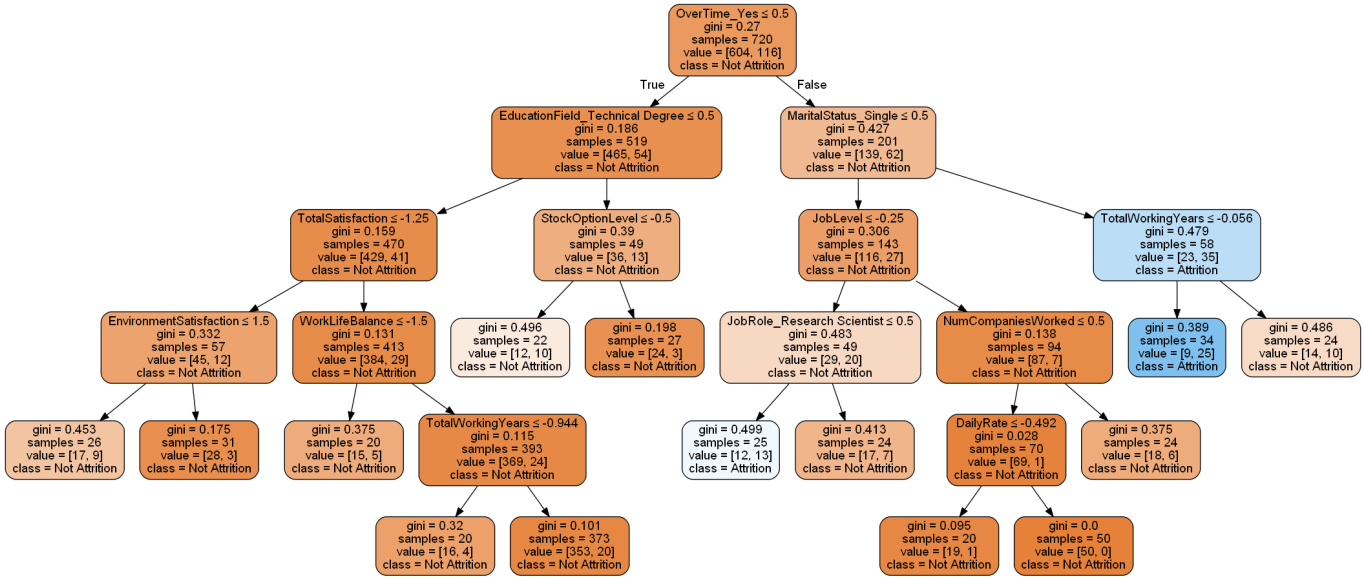
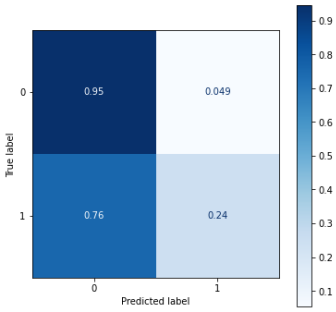


Figura 22: Schema del Decision Tree

In Figura [23] vengono mostrate (a sinistra) le performance sul test set del modello, la matrice di confusione corrispondente (al centro) e le prime 5 features in termini di importanza (riscontrabili nello schema precedente). Con il modello risultante si ottiene un'accuracy elevata, pari all'84%, corrispondente alla percentuale di record etichettati con la classe maggioritaria, come da aspettative. Per la classe minoritaria è stata ottenuta una F1 molto bassa, pari al 36%, rispetto al 91% della classe maggioritaria (visibile in Figura [24]). Il miglior modello ottenuto non è riuscito quindi a predire correttamente la classe minoritaria, come visibile anche dalla *Confusion Matrix*.

Accuracy	0.839
Precision	0.500
Recall	0.282
Specificity	0.956
F1	0.360
ROC	0.752



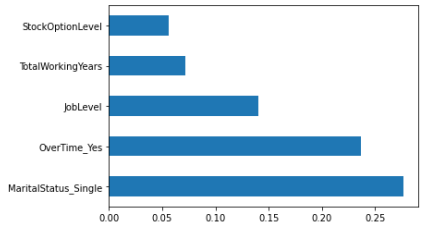


Figura 23: Risultati Finali - Feature Importance

	Precision	Recall	F1	Support
Attrition (1)	0.50	0.28	0.36	71
NotAttrition (0)	0.87	0.95	0.91	370

Figura 24: Risultati Finali per target

4.4 K-Nearest Neighbors

La fase preliminare di validazione ha prodotto per il classificatore in esame le tre configurazioni riportate in Figura [25], nella quale vengono anche mostrati i risultati ottenuti per le metriche prese in considerazione.

		Optimal Recall	Optimal F1	Optimal ROC
N_Neighbors		2	5	16
Metric		manhattan	euclidean	euclidean
Weights		distance	distance	uniform
Recall	**Training**	1.000	1.000	0
	Validation	0.189	0.172	0
F1	**Training**	1.000	1.000	0
	Validation	0.196	0.213	0
ROC	**Training**	1.000	1.000	0.714
	Validation	0.512	0.528	0.567

Figura 25: KNN: Risultati GridSearch

É possibile notare come le configurazioni ottime per *Recall* ed *F1* producano eccessivo overfitting sul training, rispetto al validation. Mentre la configurazione ottima per *ROC* non produce risultati significativi per le metriche *Recall* e *F1*, benchè non sia influenzata da overfitting (come visibile anche a destra in Figura [26]).

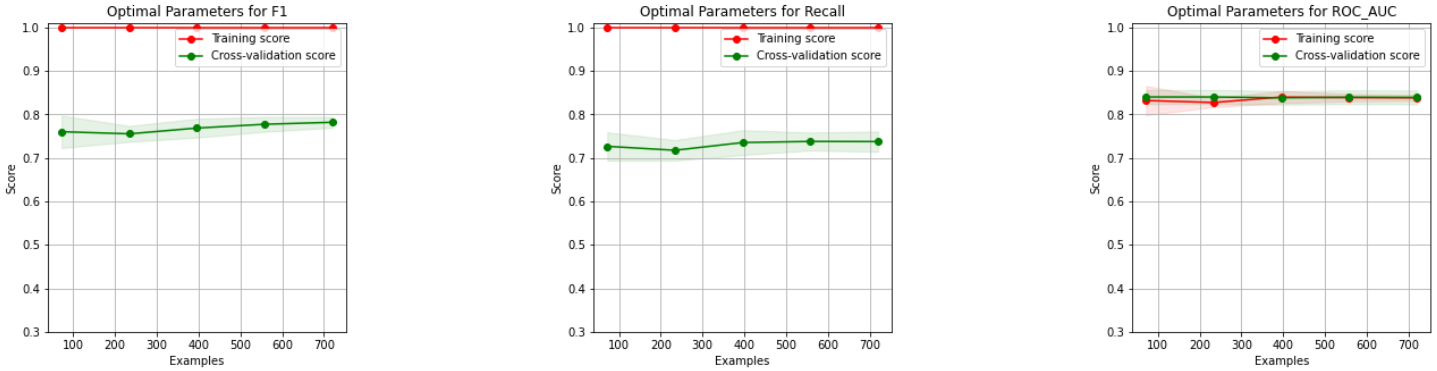


Figura 26: KNN: Curve di apprendimento per le configurazioni ottenute

Per risolvere il problema dell'Overfitting sarebbe possibile testare il modello dopo aver trattato il dataset attraverso la tecnica dell'*Oversamplig* della classe minoritaria. Quest'ultima tecnica risulta essere preferita rispetto all'*Undersampling* a causa della quantità insufficiente dei dati a disposizione. Nel caso in cui venisse adottato il sovracampionamento, sarebbe opportuno effettuarlo dopo la divisione in 70% training e 30% test. Eseguirlo prima, potrebbe portare a risultati distorti a causa della possibilità che la stessa istanza di classe di minoranza venga utilizzata sia per l'apprendimento del modello, che per il test dello stesso.

Tuttavia, in questo contesto è stato preferito mostrare gli effetti del classificatore utilizzando le 5 features risultate più importanti dal DecisionTree, con le quali sono state ottenute le migliori performances: **MaritalStatus_Single**, **OverTime_Yes**, **JobLevel**, **TotalWorkingYears**, **StockOptionLevel**. La GridSearch con questo insieme di features ha prodotto un'unica configurazione con la quale è stato ridotto notevolmente l'overfitting causato dalle configurazioni precedenti. I risultati ottenuti dalla configurazione e la curva d'apprendimento corrispondente sono riportati in Figura [27].

N_Neighbors		3
Metric		manhattan
Weights		uniform
Recall	**Training**	0.485
	Validation	0.363
F1	**Training**	0.566
	Validation	0.420
ROC	**Training**	0.892
	Validation	0.697

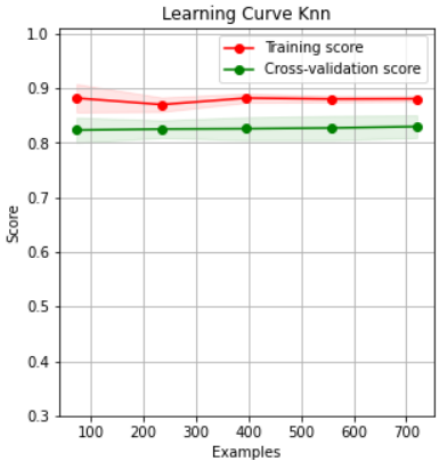


Figura 27: KNN: GridSearch - Learning Curve

Per questa configurazione in Figura [28] vengono visualizzati graficamente le medie d'errore di predizione per k $\in$ [1,24]. É possibile notare che k = 3 non è il valore che minimizza l'errore, tuttavia un valore maggiore porta il modello ad un eccessivo adattamento al training.

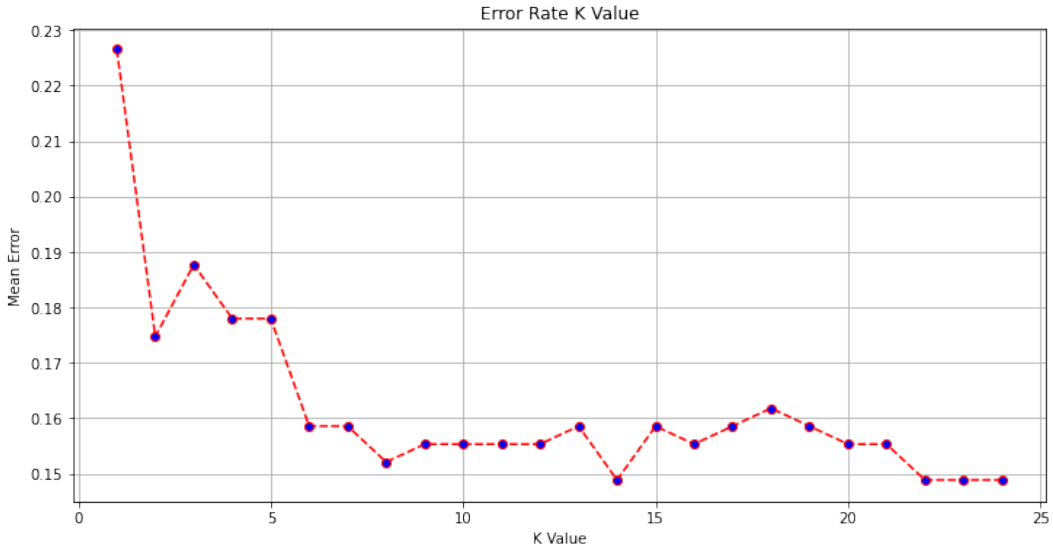


Figura 28: KNN: Error Rate

I risultati ottenuti dal classificatore sono riportati in Figura [29]. In particolare, la curva ROC mostra che il classificatore basato sul KNN migliora il classificatore casuale, presentando un AUC di 0.713.

Accuracy	0.825
Precision	0.456
Recall	0.423
Specificity	0.903
F1	0.438
ROC	0.713

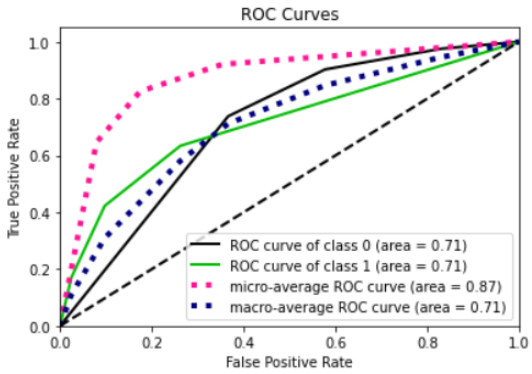


Figura 29: KNN - Risultati

4.5 Confronto

Tutti gli algoritmi di classificazione testati hanno chiaramente riportato un'accuracy elevata (superiore all'80%), ma poco interessante a causa dello sbilanciamento del dataset iniziale. Tuttavia, dalle analisi effettuate è possibile concludere che mantenendo le medesime features, il Decision tree risulta avere performance migliori, producendo un minor rischio di overfitting rispetto al K-Nearest Neighbors.

5 Pattern Mining

In questa sezione vengono presentati i risultati ottenuti dalla ricerca di regole interessanti nei dati, al fine di comprendere meglio le informazioni presenti nel dataset.

5.1 Pre-processing

Per poter applicare l'algoritmo **Apriori** i dati sono stati pre-processati nel seguente modo:

- I missing values degli attributi **BusinessTravel**, **PerformanceRating**, **Gender** sono stati rispettivamente imputati con i valori *Missing_BusinessTravel*, *Missing_PerformanceRating* e *Missing_Gender*. É stato quindi

eseguito un preprocessing differente da quello svolto in Sezione [2.3.1] per poter conseguire le analisi riportate in Sezione [5.4].

- Discretizzazione degli attributi numerici in 4 e 5 bins (a seconda della distribuzione dell'attributo). In particolare, l'attributo **MonthlyIncome** è stato discretizzato in 5 bins utilizzando le etichette *Very Low*, *Low*, *Medium*, *High*, *Very High*.
- Trasformazione in *valore_NomeAttributo* dei valori numerici e dei valori categorici conteneti valori ripetuti, al fine di evitare ambiguità nella lettura delle regole generate dall'algoritmo. I valori (*No*, *Yes*) della variabile target **Attrition** sono stati ad esempio sostituiti con i valori *Not_Attrition*, *Attrition*.
- Eliminazione delle variabili relative alla retribuzione **DailyRate**, **HourlyRate**, **MonthlyRate**, degli indici di soddisfazione **EnvironmentSatisfaction**, **JobSatisfaction**, **WorkLifeBalance**, **JobInvolvement** ritenuti poco interessanti dalle analisi effettuate nelle sezioni precedenti.
- Trasformazione dei dati in un *transactional dataset*.

Per l'estrazione di frequent patterns e association rules delle Sezioni [5.2 - 5.3] sono state, inoltre, escluse le variabili di annualità (eccetto **TotalWorkingYears**), poichè portavano ad un numero elevato di pattern/regole banali, come ad esempio il pattern: $<= 4\_YearsInCurrentRole$, $<= 4\_YearsWithCurrManager$, $<= 4\_YearsSinceLastPromotion$, $<= 10\_YearsAtCompany$.

5.2 Estrazione frequent patterns con diversi valori di support

Per trovare la soglia minima di support, adatta a non generare regole banali e non rilevanti, è stato analizzato il numero di itemsets (*frequent*, *closed* e *maximal*) trovati al variare del supporto in un intervallo [10%, 60%], rispetto a diverse soglie di lunghezza minima dei pattern. In Figura [30] è possibile notare come aumentando la lunghezza minima da 3 a 4 elementi, il numero di itemsets si riduce di circa il 50%. Per un valore di support superiore al 30% il numero di itemsets trovati è esiguo o nullo. Dato l'alto numero di attributi e i risultati appena ottenuti è stato deciso di considerare pattern con lunghezza minima di 4 elementi.

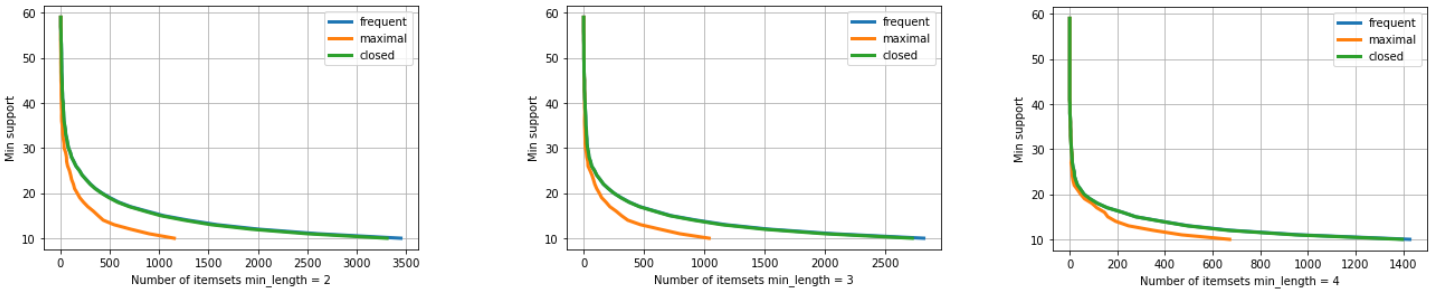


Figura 30: Numero itemsets al variare del support

In Figura [31] vengono riportati gli itemsets considerati più significativi per un support minimo del 30%. Da un'analisi più approfondita però è possibile notare come gli itemsets sono semplicemente composti dai valori più diffusi all'interno del dataset per i relativi attributi, ad esempio l'associazione di **3.0_PerformanceRating** e **Travel_Rarely** risulta essere un'informazione triviale.

Frequent Itemset	Support Count	Support
Male, Research & Development, Not_Attrition, 3.0_PerformanceRating	448	30.48
Research & Development, Not_OverTime, Travel_Rarely, Not_Attrition	463	31.50
Not_OverTime, Travel_Rarely, Not_Attrition, 3.0_PerformanceRating	602	40.95

Figura 31: Frequent itemset minsupp = 30, minlen = 4

Abbassando il min_sup al 5% si riscontrano nuove associazioni. In particolare, vengono associati i dipendenti di genere femminile con bassi livelli lavorativi e soddisfazione; oppure l'abbandono del dipendente con bassa soddisfazione e l'aver svolto straordinari.

Frequent Itemset	Support Count	Support
Attrition, OverTime, Travel_Rarely, 3.0_PerformanceRating	80	5.44
Attrition, Single, 1_JobLevel, 0_StockOptionLevel	78	5.21
Attrition, Low_TotalSatisfaction, Travel_Rarely, 3.0_PerformanceRating	83	5.65
>31 <=37_Age, 1_JobLevel, Research & Development, Not_Attrition	114	7.76
Medium_TotalSatisfaction, Male, Research & Development, Not_Attrition	202	13.74
2_JobLevel, Female, Low_TotalSatisfaction, 3.0_PerformanceRating	76	5.17
2_JobLevel, Low_MonthlyIncome, Life Sciences, Not_Attrition	77	5.24
High_TotalSatisfaction, Male, Not_Attrition, 3.0_PerformanceRating	158	10.75
Sales Executive, 2_JobLevel, Not_Attrition, Sales	197	13.40

Figura 32: Frequent itemset minsupp = 5, minlen = 4

5.3 Estrazione di regole di associazione con diversi valori di confidence

In questa fase è stato analizzato il numero di regole al variare della confidenza minima in [1, 100] per tre soglie di supporto minimo (10%, 15%, 20%), come mostrato in Figura [33] (a sinistra). A destra della Figura [33] viene mostrato invece il *Lift* delle regole generate al variare di *support* e *confidence*, potendo notare che un lift > 1.5 è ottenibile solo da regole con un support $< 20\%$.

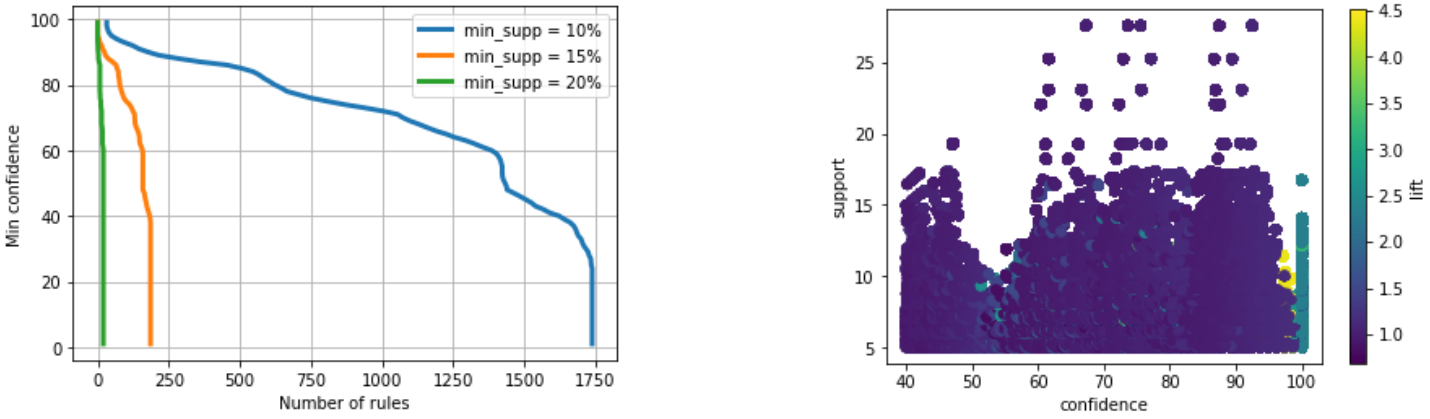


Figura 33: AR: Distribuzioni parametri

Considerando un min_support = 20% e una min_confidence = 70% vengono infatti ottenute solo 14 regole (di cui 2 con $lift < 2$), delle quali vengono mostrate in Figura [34] le due con lift più alto.

Rules	Support	Confidence	Lift
Male, Research & Development, Not_Attrition, 3.0_PerformanceRating => Not_OverTime	23.06	75.67	1.05
Research & Development, Not_OverTime, Travel_Rarely, 3.0_PerformanceRating => Not_Attrition	27.55	92.47	1.10

Figura 34: AR minsupp = 20% minconf = 70% minlenght = 4

Tuttavia, le regole ottenute contengono gli stessi difetti discussi in Sezione [5.2]. Sono state quindi esplorate le regole ottenibili con un min_supp più basso, pari a 10% e min_conf = 80%. La distribuzione di lift e confidence delle regole ottenute sono mostrate in Figura [35]. Dalla Figura [36], nella quale vengono riportate alcune delle regole ottenute, è possibile notare come le prime tre presentino un *lift* decisamente superiore ad 1, il quale indica una correlazione positiva, mentre le ultime due hanno un *lift* tendente ad 1, quindi antecedente e conseguente sono quasi indipendenti.

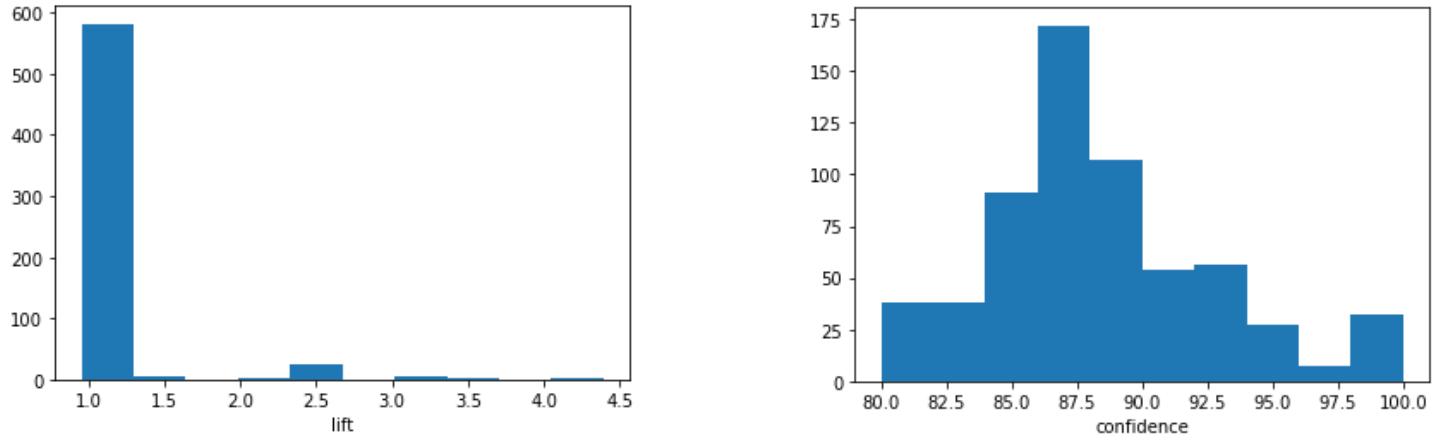


Figura 35: AR Istogrammi: lift - confidence

Rules	Support	Confidence	Lift
Research Scientist, Research & Development, Travel_Rarely, 3.0_PerformanceRating => 1_JobLevel	10.61	81.25	2.20
Sales, 2_JobLevel, Not_OverTime, Not_Attrition => Sales Executive	10.27	97.42	4.39
Single, Male, Not_OverTime, Not_Attrition, 3.0_PerformanceRating => 0_StockOptionLevel	10.54	98.32	2.33
Sales Executive, Sales, 2_JobLevel, Not_OverTime => Not_Attrition	10.27	90.96	1.08
<=2_DistanceFromHome, Research & Development, Travel_Rarely, 3.0_PerformanceRating => Not_Attrition	10.41	88.44	1.05

Figura 36: AR minsupp = 10% minconf = 80% minlenght = 4

5.4 Sostituzione dei Missing values

Durante l'analisi dei *Missing values* presenti nelle regole d'associazione è stata constatata la presenza dei valori mancanti in gran parte delle regole aventi **Not_Attrition** come conseguente, ma sempre presenti in regole con *support* basso (inferiore al 6%). Alcune di queste regole sono riportate in Figura [37].

Rules	Support	Confidence	Lift
Missing_BusinessTravel, Not_OverTime => Not_Attrition	5.44	87.91	1.05
Missing_BusinessTravel, Married, <=10_YearsAtCompany => Not_Attrition	3.06	84.91	1.01
Missing_PerformanceRating, Male, Not_OverTime => Not_Attrition	5.24	95.06	1.13
Missing_PerformanceRating, Not_OverTime, <=4_YearsSinceLastPromotion, <=10_YearsAtCompany => Not_Attrition	5.37	92.94	1.11
Missing_Gender, 3_Education => Not_Attrition	2.18	84.21	1.00
Missing_Gender, 3_JobInvolvement => Not_Attrition	2.52	84.10	1.00

Figura 37: Regole con Missing_NomeAttributo nell'antecedente

A causa dello sbilanciamento dei valori nei singoli attributi, molti valori (quelli che occorrono meno frequentemente) sono apparsi come conseguenti di sole regole con *support* e *confidence* molto bassi, benché *lift* maggiore di 1, come visibile in Figura [38].

Rules	Support	Confidence	Lift
>14 <=18_PercentSalaryHike, Very Low_MonthlyIncome, Male, Not_Attrition => Non-Travel_BusinessTravel	1.09	21.62	2.30
High_TotalSatisfaction_Range, Life Sciences_EducationField, 3_JobInvolvement => Travel_Frequently_BusinessTravel	2.04	30.61	1.77
Low_TotalSatisfaction_Range, <=4_YearsSinceLastPromotion => 4.0_PerformanceRating	5.44	15.87	1.17
Low_MonthlyIncome, <=10_YearsAtCompany => 4.0_PerformanceRating	5.31	15.45	1.14

Figura 38: Regole per i valori poco frequenti

Date le considerazioni fatte è stato scelto di utilizzare le regole ottenute con *min_sup = 2%*, *min_conf = 60%* e *lift > 1*, poichè per support maggiori del 5% non sarebbero state trovate regole con conseguenti adeguati allo scopo.

In Figura [39] sono riportati i risultati ottenuti sostituendo i valori mancanti con l'uso di tali regole, rispetto alla metodologia utilizzata in Sezione [2.3.1]. Come è possibile osservare, una buona percentuale di records non è stata ricoperta; per la parte restante sono stati, invece, ottenuti valori di accuratezza elevati. I due metodi hanno quindi imputato lo stesso valore per più del 70% dei dati mancanti, questo è spiegabile probabilmente (come già discusso) dallo sbilanciamento della distribuzione dei valori negli attributi.

Attributo	Percentuale non coperta	Accuracy (%)
BusinessTravel	36.12	82
Gender	25.37	72
PerformanceRating	43.06	94

Figura 39: AR: Risultati sostituzione Missing Values

In questa fase non sono stati trattati altri attributi contenenti *missing values*, poichè, a causa dell'alto numero di valori distinti, per avere regole utili allo scopo del task si sarebbero dovute cercare regole con *support* e *confidence* decisamente bassi, difficili da ricercare dal punto di vista computazionale.

5.5 Predizione della variabile target

Per stimare la variabile target, utilizzando le regole d'associazione, sono state estratte ed utilizzate regole sia dal dataset preprocessato in Sezione [5.1], sia da una sua versione bilanciata tramite *Random Undersampling*. La prova bilanciata è stata effettuata perchè, in relazione a quanto analizzato in Sezione [2.2], le regole d'associazione non avrebbero prodotto regole con **Attrition** (classe minoritaria) come conseguente. Nel primo caso, sono state considerate regole con **supporto** maggiore del **30%** e una **confidenza** superiore all'**80%**; mentre per la prova bilanciata sono stati utilizzati come limiti inferiori, rispettivamente **20%** e **65%**. In entrambi i casi, sono state considerate solo regole con almeno 4 items e **lift** minimo pari a 1. Dalle regole così ottenute sono state selezionate quelle contenenti la variabile target nella conseguenza. Nelle Figure [40 - 41] vengono riportate alcune delle

regole ottenute ritenute più significative. É possibile notare, come già ipotizzato in Sezione [2.2], che l'assenza di straordinario, presente in tutte le regole, è un indice di *non abbandono*, mentre la presenza di valori quali **Travel_Rarely** e **3.0_PerformanceRating** (come già analizzato) dipende dalla distribuzione sbilanciata dei valori per il corrispondente attributo. Invece, bassi valori di livello lavorativo, soddisfazione e anni di lavoro (ad esempio sotto lo stesso manager o nel ruolo corrente) determinano regole con *abbandono* come conseguente.

Antecedente	Support	Confidence	Lift
Research&Development, Not_OverTime, <=10_YearsAtCompany, 3.0_PerformanceRating	31.90	92.14	1.98
Not_OverTime, Travel_Rarely, <=4_YearsSinceLastPromotion, <=10_YearsAtCompany	32.79	90.43	1.08
Male, Not_OverTime, <=10_YearsAtCompany, 3.0_PerformanceRating	30.07	89.66	1.069

Figura 40: AR predizione target: conseguente Not_Attrition

Antecedente	Support	Confidence	Lift
0_StockOptionLevel, <=4_YearsWithCurrManager, <=4_YearsInCurrentRole, <=4_YearsSinceLastPromotion	23.89	69.32	1.387
1_JobLevel, <=4_YearsWithCurrManager, Travel_Rarely,<=4_YearsSinceLastPromotion	20.25	69.06	1.38
Low_TotalSatisfaction, <=4_YearsWithCurrManager, <=4_YearsInCurrentRole, <=4_YearsSinceLastPromotion	21.31	67.33	1.35

Figura 41: AR predizione target: conseguente Attrition

In Figura [42] vengono mostrati i risultati ottenuti per la previsione della variabile target usando il dataset sbilanciato (*ConfusionMatrix* centrale) e quello bilanciato (matrice di destra). Rispetto ai risultati ottenuti in Sezione [4] è possibile notare che le regole ottenute dal dataset bilanciato (i cui risultati complessivi sono i migliori per le prove effettuate in questa sezione) hanno conseguito un'*F1* superiore ai precedenti classificatori, pur con accuratezza inferiore. Inferiore è anche il numero di falsi negativi, con un aumento del 20% dei veri positivi.

	min_sup = 30% min_confidence = 80%	min_sup = 20% min_confidence = 65%
Accuracy	0.779	0.648
Precision	0.256	0.733
Recall	0.194	0.464
F1	0.256	0.733
TN - FN	1099 - 191	197 - 127
TP - FP	46 - 134	110 - 40

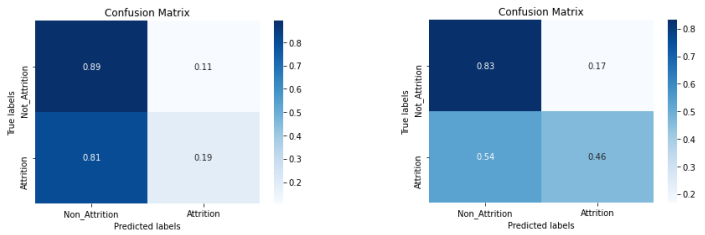


Figura 42: AR predizione target: risultati

Inoltre, è possibile notare che il classificatore sul dataset sbilanciato produce performance di gran lunga inferiori rispetto ai risultati ottenuti in Sezione [4].