



Università di Pisa

DIPARTIMENTO DI FISICA
Corso di Laurea Magistrale in Fisica

Relazione progetto di Data Mining

Gruppo 33: Daniele Maria Di Nosse, Angelo Lasala, Raffaele Paradiso

Indice

1	Introduzione	2
2	Data Understanding	2
2.1	Data Semantics	2
2.2	Analisi statistica	2
2.3	Data Quality : Outliers e Missing values	5
3	Data Preparation	6
4	Clustering	8
4.1	K-Means	8
4.1.1	Validazione	10
4.2	DB-Scan	11
4.2.1	Validazione	12
4.3	Hierarchical	13
4.3.1	Validazione	14
4.4	Confronto fra gli algoritmi	14
5	Classification	15
5.1	Decision Tree	15
5.2	K-Nearest Neighbors	17
6	Association Rules	18
7	Conclusioni	21

1 Introduzione

Determinare le possibili relazioni che intercorrono fra le caratteristiche dei dipendenti di un'azienda può risultare di grande utilità per predire i possibili scenari lavorativi che posso verificarsi e gestire di conseguenza l'organizzazione del personale in maniera ottimale. Nel presente progetto ci si è posto l'obiettivo di valutare tali legami tramite un approccio di data mining. Le informazioni che si sono utilizzate sono relative ad un data frame fittizio (leggermente modificato) generato da IBM e presente sul portale Kaggle(URL <https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset>). Non ci si è posto un obiettivo principale, ovvero la determinazione di legami, correlazioni e classificazioni relativi ad un singolo attributo rispetto a tutti gli altri, ma si è proceduto in maniera più generale ricoprendo uno spettro più ampio di possibili relazioni fra tutte le variabili.

2 Data Understanding

2.1 Data Semantics

Nella prima fase dell'elaborazione si è studiato il data frame, valutando il numero degli attributi, la loro natura e dominio.

Il numero di attributi è pari a 33. Si dividono in attributi numerici e categorici, ma ad uno sguardo più attento si nota che alcuni di essi, come Education o EnviromentSatisfaction, presentano valori numerici che poco si adattano al loro significato. Si ha infatti che sussistono le seguenti relazioni:

Education	EnvironmentSatisfaction	JobInvolvement	JobSatisfaction
1 : 'Below College'	1 : 'Low'	1 : 'Low'	1 : 'Low'
2 : 'College'	2 : 'Medium'	2 : 'Medium'	2 : 'Medium'
3 : 'Bachelor'	3 : 'High'	3 : 'High'	3 : 'High'
4 : 'Master'	4 : 'Very High'	4 : 'Very High'	4 : 'Very High'
5 : 'Doctor'			

PerformanceRating	RelationshipSatisfaction	WorkLifeBalance
1 : 'Low'	1 : 'Low'	1 : 'Bad'
2 : 'Good'	2 : 'Medium'	2 : 'Good'
3 : 'Excellent'	3 : 'High'	3 : 'Better'
4 : 'Outstanding'	4 : 'Very High'	4 : 'Best'

Di conseguenza, il dominio di tali attributi è di tipo categorico od ordinale e non numerico (un attributo ordinale è effettivamente una sottocategoria categorica, si è scelto comunque di elencarli separatamente). Inoltre, sebbene non si abbiano informazioni dettagliate sulle classi relative agli attributi JobLevel e StockOptionLevel, per la loro stessa natura si è deciso di trattarli come attributi ordinali. Organizzando tutte le variabili per la loro tipologia, si ottiene quindi la seguente suddivisione:

Categorici: 8	Ordinali: 10	Numerici: 15
Attriction	Business Travel	Age
Department	Education	Daily Rate
Education Filed	Enviroment Satisfaction	Distance From Home
Gender	Job Involvement	Monthly Income
Job Role	Job Level	Monthly Rate
Marital Status	Job Satisfaction	Montly Rate
Over 18	Performance Rating	Num Company Worked
Over Time	Relationship Satisfaction	Persent Salary Hike
	Stock Option Level	Standard Hours
	Work Life Balance	Total Working Years
		Training Time Last Year
		Years At Company
		Years In Current Role
		Years Since Last Promotion
		Yeaer With Current Manager

Per quanto riguarda il range di valori degli attributi risulta essere molto più discretizzato per gli attributi ordinali che per gli attributi numerici. Inoltre differisce molto da attributo ad attributo (anche di 4 ordini di grandezza), cosa che sottolinea sin da ora l'importanza di una trasformazione delle variabili.

2.2 Analisi statistica

Le frequenze degli attributi categorici e le relative mode sono riportate nelle seguenti tabelle.

Attriction	Educational Field	Departement	Gender	Over Time
'No': 83.9%	'Life Science': 41.2%	'Research and Development': 65.4%	'Male': 57.2%	'No': 71.7%
'Yes': 16.1%	'Medical': 31.6%	'Sales': 30.3%	'Female': 37.7%	'Yes': 28.3%
	'Marketing': 10.8%	'Human Resources': 4.3%	'MISSING': 5.1%	
	'Technical Degree': 9.0%			
	'Other': 5.6%			
	'Human Resources': 1.8%			
Busniss Travel	Job Rule	Marital Status	Over 18	
'Travel Rarely': 64.4%	'Sales Executive': 22.2%	'Married': 45.8%	'Yes': 68.2%	
'Travel Frequently': 17.3%	'Research Scientist': 19.9%	'Single': 32.0%	'MISSING': 31.8%	
'Non Travel': 9.4%	'Laboratory Technician': 17.6%	'Divorced': 3.2%		
'MISSING': 9.0%	'Manufacturing Derector': 9.9%			
	'Healthcare Representative': 8.9%			
	'Manager': 6.9%			
	'Sales Representative': 5.6%			
	'Research Derector': 5.4%			
	'Human Resources': 3.5%			

Figura 1: Frequenze degli attributi categorici

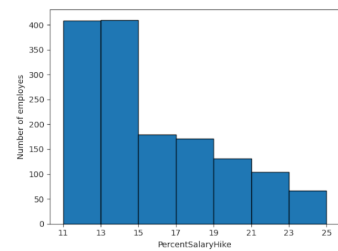
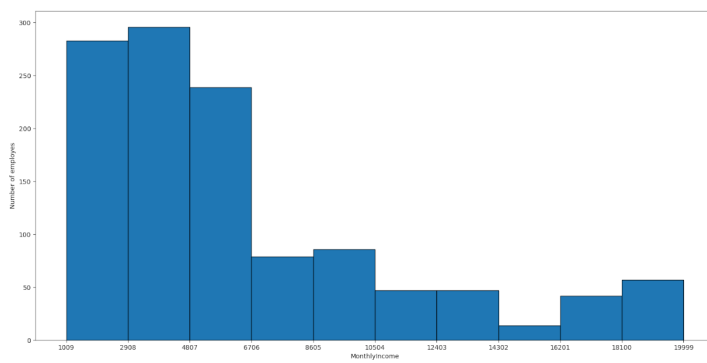
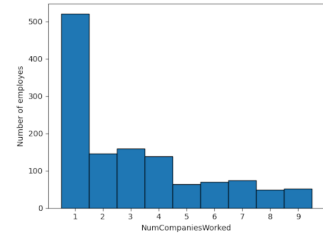
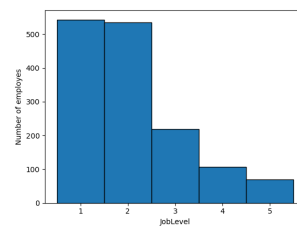
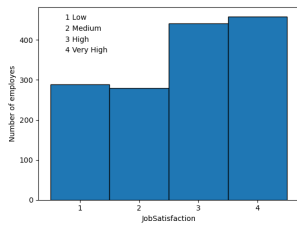
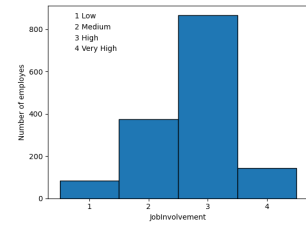
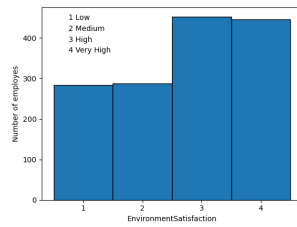
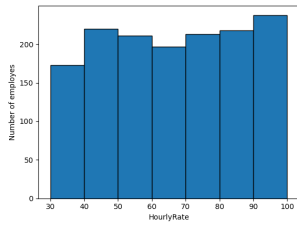
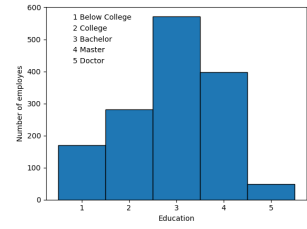
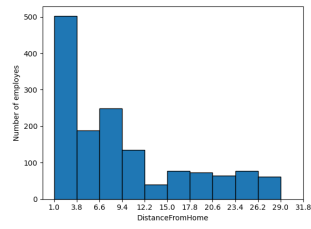
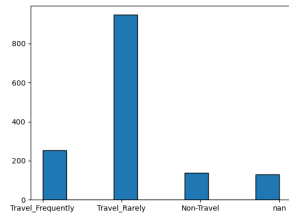
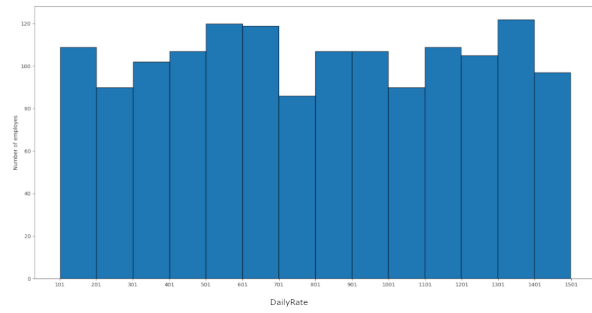
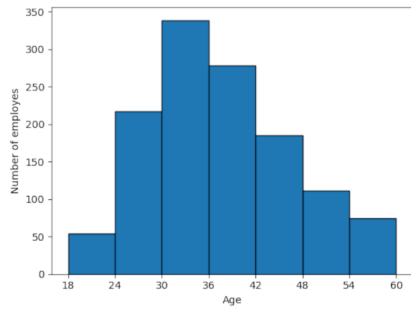
Moda	
Attriction	No
Educational Field	Life Science
Departement	Research and Development
Gender	Male
Over Time	No
Business Travel	Travel Rarely
Job Rule	Sales Executive
Marital Status	Married
Over 18	Yes
Education	Bachelor
Enviroment Satisfaction	High
Job Involvement	High
Job Satisfaction	Very High
Performance Rating	Excellent
Relationship Satisfaction	High
Job Level	1
Work Life Balace	Better
Stock Option Level	0

Figura 2: Mode

Le distribuzioni degli attributi ordinali e numerici con alcuni indici statistici sono rappresentate di seguito. Si può notare la forte asimmetria di molte distribuzioni ed un varianza molto grande in alcuni attributi. Tali problematiche dovranno essere sanate con opportune trasformazioni.

	Age	DailyRate	DistanceFromHome	HourlyRate	MonthlyIncome
count	1258	1470	1470	1470	1190
mean	37.1	802.5	9.2	65.9	6549.0
std	9.1	403.5	8.1	20.3	4732.8
min	18	102	1	30	1009
25%	30	465	2	48	2973.3
50%	36	802	7	66	4907.5
75%	43	1157	14	83.7	8437.5
max	60	1499	29	100	19999
	MonthlyRate	NumCompaniesWorked	PerentSalaryHikee	TotalWorkingYear	TraningTimeLast Years
count	1470	1470	1470	1470	1178
mean	14313.1	2.7	15.2	11.3	2.8
std	7117.8	2.5	3.6	7.8	1.3
min	2094	0	11	0	0
25%	8047	1	12	6	2
50%	14235.5	2	14	10	3
75%	20461.5	4	18	15	3
max	26999	9	25	40	6
	YearsAtCompany	YearsInCurrentRole	YearSinceLastPromotion	YarsWithCurrManager	StandardHours
count	1396	1470	1470	1470	753
mean	6.9	4.2	2.2	4.1	80
std	6.0	3.6	3.2	3.6	0
min	0	0	11	0	80
25%	3	2	0	2	80
50%	5	3	1	3	80
75%	9	7	3	7	80
max	40	18	15	17	80

Figura 3: Indici statistici per gli attributi numerici



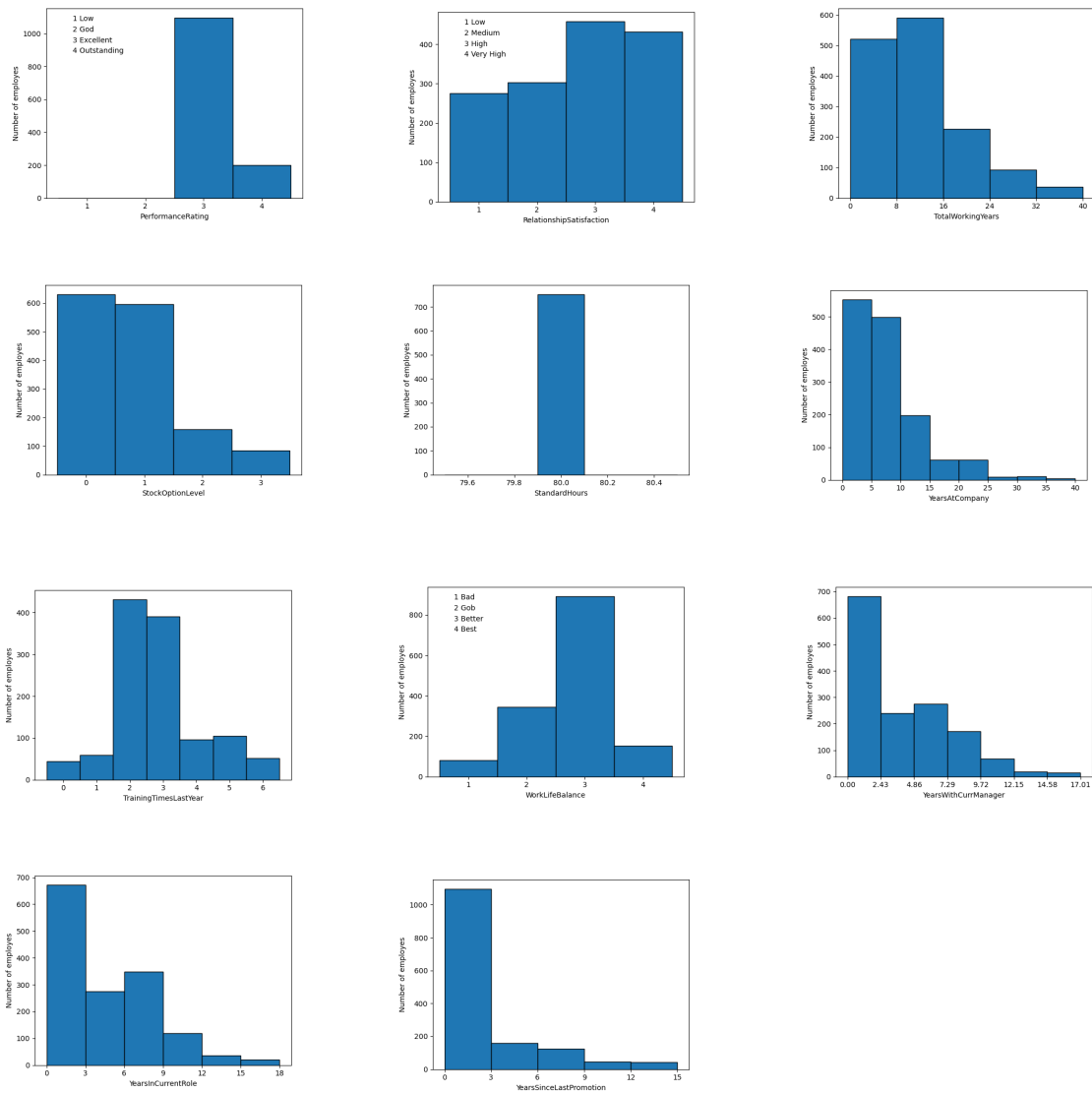


Figura 4: Istogrammi attributi numerici ed ordinali

2.3 Data Quality : Outliers e Missing values

La qualità dei dati è fortemente influenzata negativamente dalla presenza di outliers e di missing values. Algoritmi di clustering e correlazioni fra gli attributi possono restituire risultati falsificati se non si gestiscono in maniera appropriata tali valori. Nel data frame utilizzato la loro presenza è evidente.

Missing Values	
Age	212
Business Travel	131
Gender	75
MonthlyIncome	280
Over 18	468
PerformanceRating	172
Standard Hours	717
Traning Time Last Years	292
Years at Company	74

Figura 5: Count dei missing values

Gli attributi StandardHours ed Over18 presentano una qualità molto scarsa: nel primo circa la metà dei records sono mancanti e la restante parte ha un unico valore, mentre il secondo, oltre a contenere anch'esso una quantità significativa di missing values, non rappresenta in ogni caso un attributo di grande importanza, considerando che la stragrande maggioranza dei dipendenti di un'azienda sono maggiorenni. Per tali motivi, si è deciso di eliminare questi due attributi.

Per la determinazione degli outliers sono stati utilizzati sia test puramente statistici (Grubbs's test) che metodi di visualizzazione (Box Plot, Principal Component Analysis e scatter plot). Come è noto, per utilizzare approcci del primo tipo

bisogna fare delle assunzioni sulla distribuzione sottostante dei valori esaminati. In particolare, il Grubbs's test, applicabile singolarmente agli attributi, richiede che i dati siano distribuiti normalmente, cosa non vera in questo caso. Di conseguenza, tale metodo è stato scartato. La Principal Component Analysis, d'altro canto, è uno dei metodi maggiormente utilizzati nella ricerca di outliers in situazioni alto-dimensionali. Proiettando lo spazio n -dimensionale in uno spazio q -dimensionale ($q < n$), costruito tramite i vettori normalizzati della matrice di correlazione, si cerca di mantenere il più intatta possibile la varianza negli attributi. Nel caso in esame, la frazione di varianza conservata non risulta essere significativa (circa 0.4), inficiando inevitabilmente i risultati ottenibili. Anche la visualizzazione degli scatter plot confrontati con gli attributi categorici non ha evidenziato alcun punto identificabile come outlier. L'unico metodo che ha avuto successo per la loro determinazione è stata la visualizzazione dei Box Plot per i singoli attributi. Si è proceduto quindi alla loro rimozione tramite eliminazione delle righe corrispondenti.

3 Data Preparation

In questa fase del lavoro ci si è posto l'obiettivo di trasformare e preparare il set di dati all'analisi successiva. I problemi precedentemente evidenziati sono stati qui risolti.

Come primo task sono stati gestiti i missing values. L'attributo BusinessTravel presenta una frequenza di NaN pari al circa 9%, confrontabile con le frequenze degli altri valori. Siccome la granulosità dell'attributo ricopre in maniera completa lo spettro delle classi plausibilmente ad esso associabili, si è deciso di valutare se ci fosse dipendenza con gli altri attributi nel data frame. Per quanto riguarda quella con i numerici, sono stati utilizzati gli scatter plot, mentre per quelli nominali è stato eseguito il test di indipendenza del chi quadro. In entrambi i casi non si sono evinte dipendenze significative ($p - value > 0.05$ sempre). Di conseguenza tale attributo è stato scartato.

Per quanto riguarda PerformanceRating, si è aggiunta una nuova classe 'MISSING', poiché si è notato che la granulosità dell'attributo in questo caso non ricopre tutto lo spettro plausibile. Si presuppone che i valori 'MISSING' possano appartenere ad una classe di ordine inferiore ad 'Excellent'.

Queste due considerazioni non sono applicabili all'attributo Gender per il quale si è scelto semplicemente di sostituire ai missing values valori estratti dalla distribuzione del campione.

Procedimento analogo è stato applicato a tutti gli attributi numerici che presentano valori mancanti, l'unica differenza è che in questo caso i valori sostitutivi sono le medie degli intervalli dei bins degli istogrammi.

Come secondo task sono state applicate funzioni di trasformazione ad attributi numerici con lo scopo di rimediare ad alcune caratteristiche delle loro distribuzioni, quali l'asimmetrie e un valore spropositato della deviazione standar. In particolare è stata applicata la radice quadrata a DistanceFromHome (skew da 0.95 a 0.40), NumCompaniesWorked (skew da 1.03 a 0.03), PercentSalaryHike (skew da 0.82 a 0.65), TotalWorkingYears (skew da 1.12 a 0.18), YearsAtCompany (skew da 1.76 a 0.43), YearsInCurrentRole(skew da 0.92 a -0.25), YearsSinceLastPromotion (skew da 1.98 a 0.74) e YearsWithCurrManager (skew da 0.83 a -0.25); invece a MonthlyIncome è stato applicato il logaritmo naturale (varianza da 4710 a 0.67). In Figura 7 sono riportate alcune distribuzioni delle variabili trasformate.

Sono stati poi trattati gli outliers tramite il metodo di visualizzazione grafica dei Box Plot, che evidenzia la presenza di outliers solo in tre attributi numerici: TrainingTimeLastYear, TotalWorkingYears, YearsAtCompany

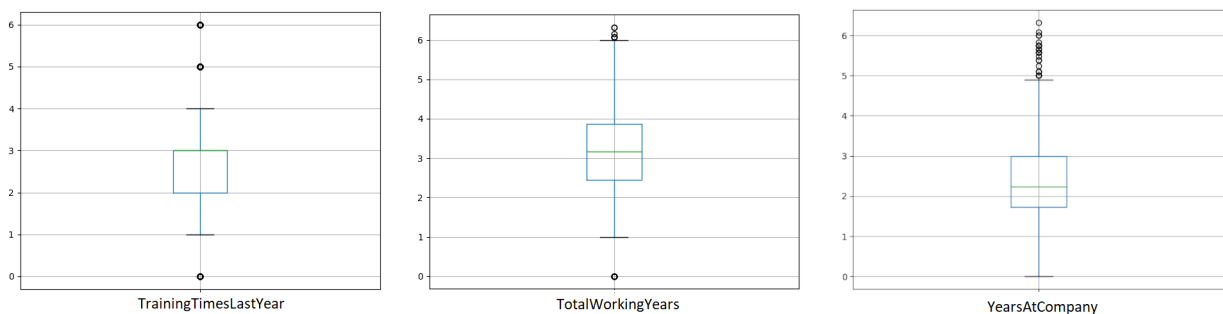


Figura 6: Box Plot degli attributi che presentano outliers

Il data frame dopo questa prima preparazione risulta contenere il 36% di dati in meno rispetto a quello di partenza.

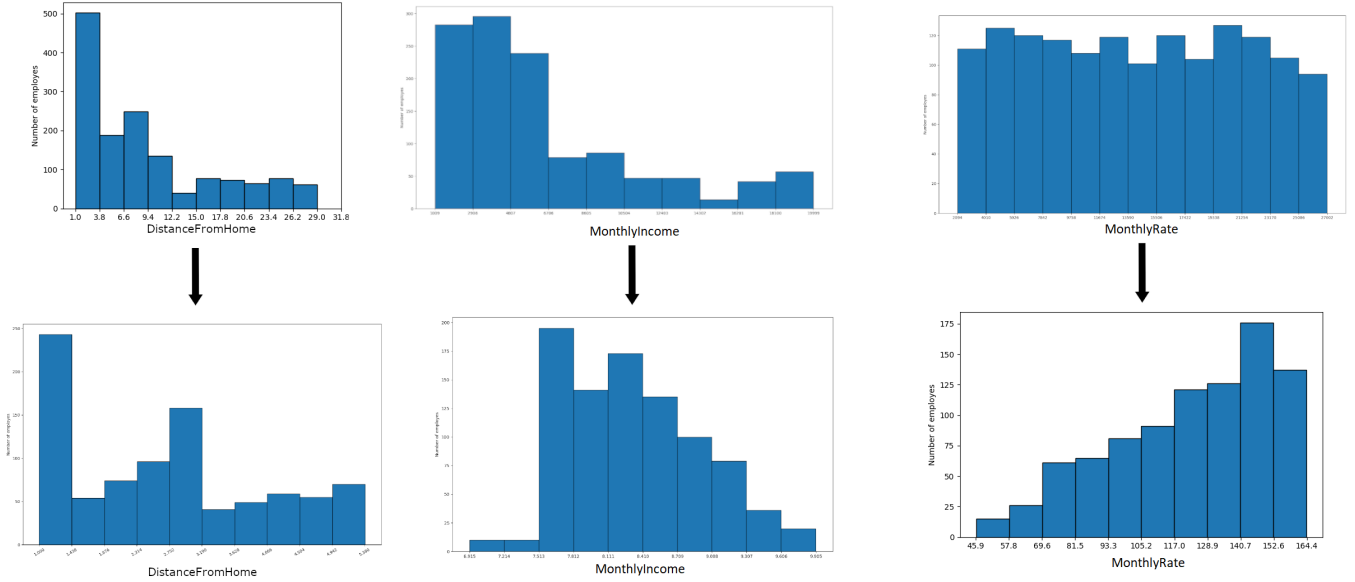


Figura 7: Distribuzione di alcune variabili trasformate

Successivamente sono stati eliminati ed aggiunti nuovi attributi.

In luogo di TotalWorkingYears e YearsAtCompany si è scelto di introdurre il loro rapporto, denominato FractionAtCompany che rappresenta la frazione di anni lavorativi del singolo dipendente nell'azienda. Tramite l'introduzione di tale attributo è stato possibile notare l'inconsistenza di alcuni records per cui tale rapporto risultava essere maggiore di 1. Tali valori sono stati eliminati. Analogamente si è proceduto per MonthlyIncome e MonthlyRate sostituiti da RateIncome, indice di quanto l'azienda spende per un impiegato in rapporto al suo stipendio. Anche in questo caso sono stati eliminati i valori inconsistenti. DailyRate e HourlyRate contengono la stessa informazione di MonthlyRate, quindi sono stati eliminati.

Inoltre, YearsInCurrentRole, YearInCurrManager e YearsSinceLastPromotion sono caratterizzati da una correlazione significativa e quindi si è deciso di mantenere solamente YearsInCurrentRole nell'analisi a seguire.

Infine è stata calcolata la matrice di correlazione lineare fra gli attributi numerici e i valori del p -value ottenuti tramite test del chi quadro per l'interdipendenza fra gli attributi categorici.

Si può notare come il p -value per la maggioranza dei casi è maggiore di 0.05, valore scelto come soglia; questo significa che l'ipotesi nulla (non c'è relazione tra i due attributi) non può essere scartata. Nell'altra situazione invece l'evidenza empirica è fortemente contraria all'ipotesi nulla, ciò significa che la presenza di una dipendenza tra gli attributi è statisticamente plausibile, come ad esempio avviene tra Attrition (quanto un impiegato è "logorato") e MaritalStatus (condizione sentimentale dell'impiegato).

Per gli attributi numerici, in seguito alle trasformazione effettuate, non sono presenti correlazioni significative.

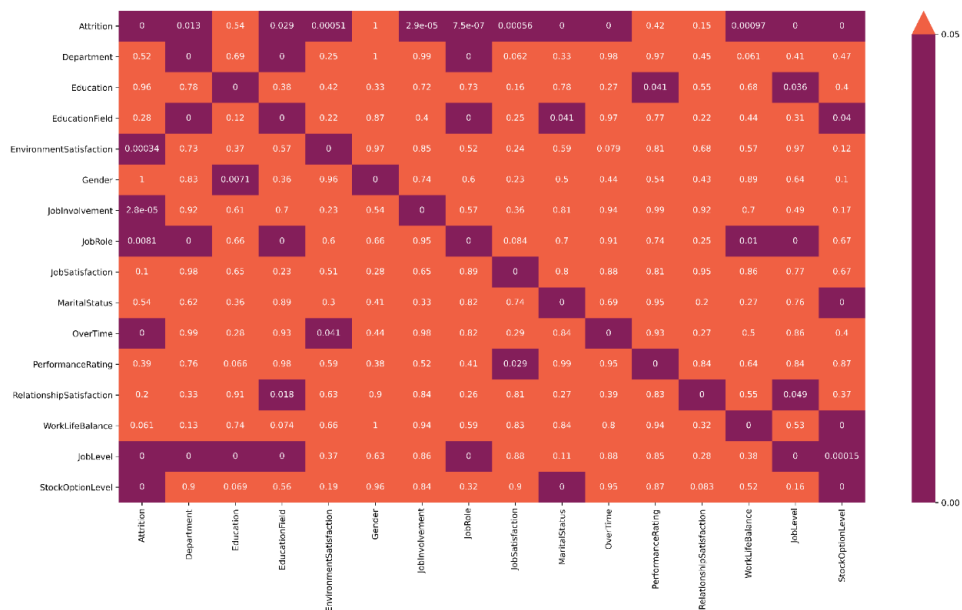


Figura 8: Matrice dei p -value per gli attributi categorici



Figura 9: Matrice di correlazione per gli attributi numerici

4 Clustering

Preparato il data frame si è proceduto all'analisi degli algoritmi di clustering: K-Means, DB Scan e Hierarchical. La dimensionalità del data frame (10) è stata considerata troppo elevata per ottenere risultati consistenti, quindi sono stati indagati sottoinsiemi 3-5 dimensionali alla ricerca di un qualche tipo di clusterizzazione. Come metrica è stata usata la distanza euclidea ed i dati sono stati inoltre normalizzati tra 0 e 1. Si vuole precisare che per la visualizzazione dei clusters ottenuti tramite K-Means e DB Scan è stato utilizzato uno spazio tridimensionale poiché, soprattutto nel primo caso, una visualizzazione bidimensionale portava ad un mixing eccessivo dei clusters stessi.

4.1 K-Means

Per quanto riguarda il K-Means i sottoinsiemi che hanno mostrato i risultati migliori sono:

1. PercentSalaryHike, FractionYearsAtCompany, YearsInCurrentRole, RateIncome, NumCompaniesWorked
2. DistanceFromHome, FractionYearsAtCompany, RateIncome, YearsInCurrentRole, TrainingTimesLastYear
3. DistanceFromHome, RateIncome, Age, FractionYearsAtCompany
4. DistanceFromHome, FractionYearsAtCompany, TrainingTimesLastYear, PercentSalaryHike, YearsInCurrentRole

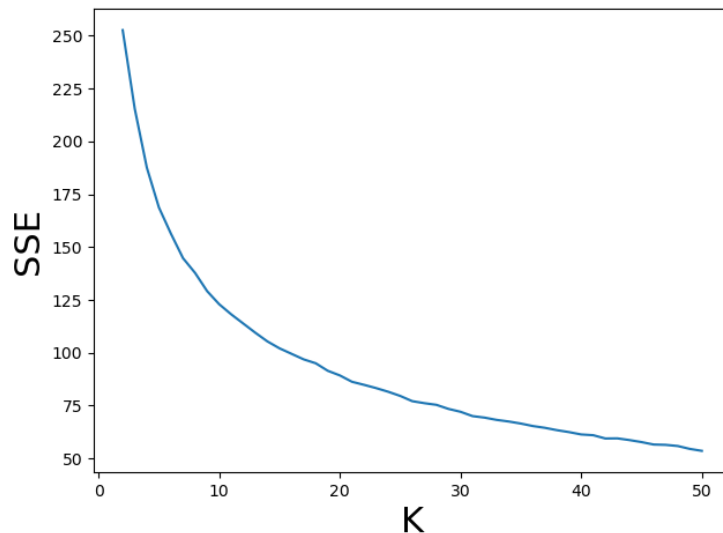


Figura 10: SSE in funzione di K

La scelta del numero di cluster K è stata presa osservando l'andamento del SSE in funzione di K (Figura 10), simile in tutti e quattro i casi esaminati; con lo scopo di aver un buon compromesso fra i due l'algoritmo è stato eseguito per K uguale a 3, 4 e 5. Dai risultati ottenuti si evince che, sebbene con $K = 5$ il valore delle SSE è minore rispetto agli altri due casi, non si apprezzano cluster evidenti: ve ne sono sempre due eccessivamente mescolati. Con $K = 3$ la divisione fra i clusters è sicuramente ben evidente ma, con $K = 4$, si ottengono comunque buoni risultati con il vantaggio di un SSE minore. Nei seguenti grafici sono mostrati i risultati ottenuti e per rendere più chiara la posizione dei centroidi, sono riportate anche le loro coordinate organizzate in parallelo per ciascun sottoinsieme usato.

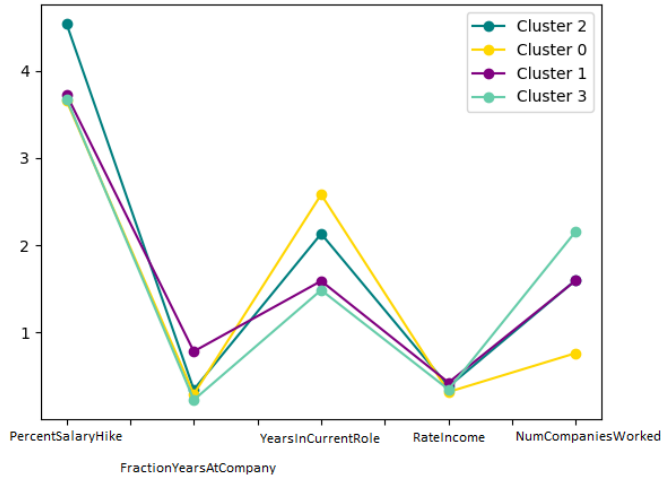


Figura 11: Parallel coordinates dei centroidi

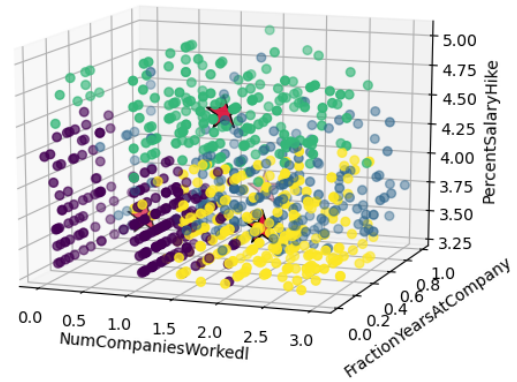


Figura 12: Cluster sottoinsieme 1: SSE= 187; silhouette= 0.18

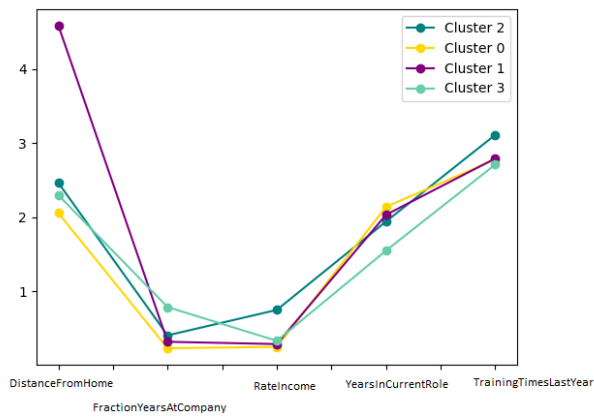


Figura 13: Parallel coordinates dei centroidi

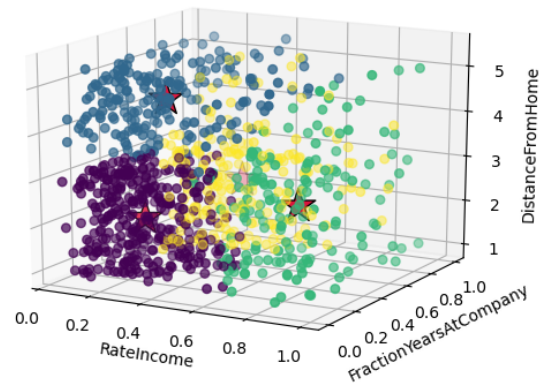


Figura 14: Cluster sottoinsieme 2: SSE= 177; silhouette= 0.21

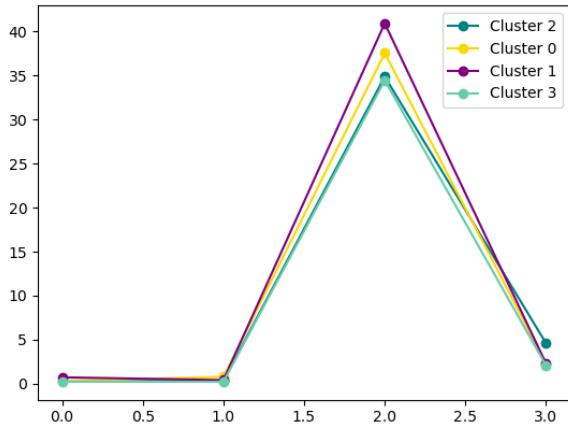


Figura 15: Parallel coordinates dei centroidi

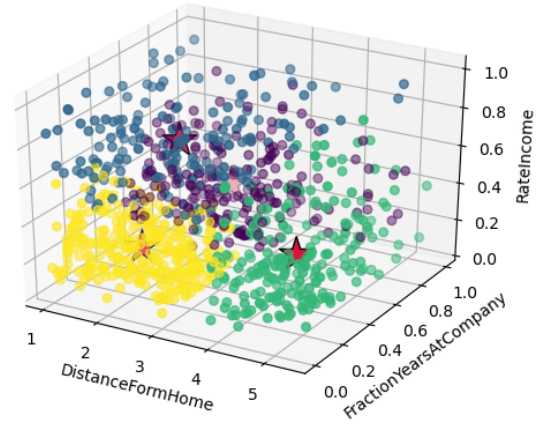


Figura 16: Cluster sottoinsieme 3: SSE= 129; silhouette= 0.25

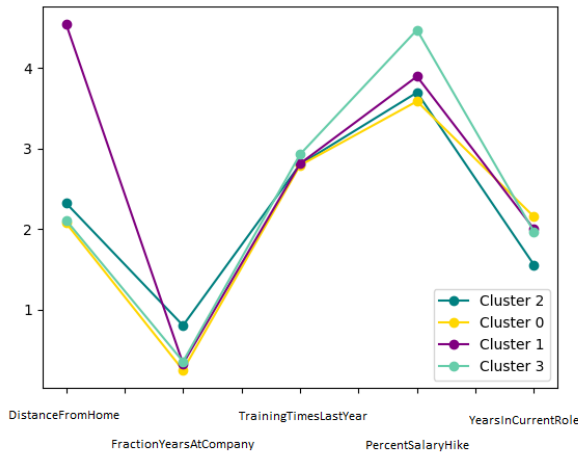


Figura 17: Parallel coordinates dei centroidi

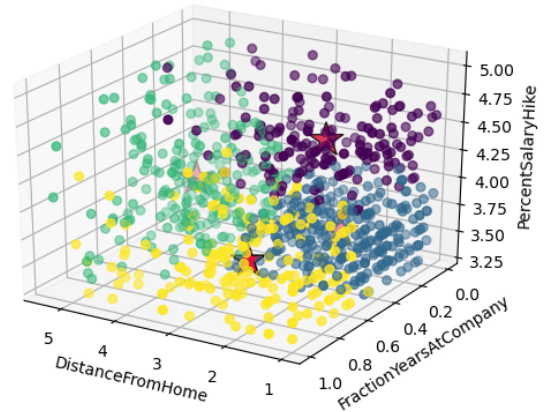


Figura 18: Cluster sottoinsieme 4: SSE= 182; silhouette= 0.20

Dalla divisione in cluster è possibile classificare gli impiegati in base ad alcune caratteristiche comuni. Ad esempio il cluster giallo nella Figura 12 esprime l'irrelevanza dell'esperienza lavorativa al di fuori dell'azienda relativamente all'aumento di stipendio. Infatti chi è stato assunto da poco (rispetto alla vita lavorativa), sebbene abbia maturato esperienza in altre aziende, ha un PercentSalaryHike basso. D'altro canto, al cluster viola è plausibile associare i lavoratori la cui esperienza è costituita quasi esclusivamente da ruoli indipendenti da compagnie. Anche in questo caso, il PercentSalaryHike è relativamente basso. Il cluster blu può essere considerato il gruppo di lavoratori la cui esperienza non è stata influenzata da altri impieghi nel passato. Infatti si nota che in questo caso l'aumento percentuale dello stipendio è indipendente dal numero di compagnie precedenti. Infine il cluster verde può essere visto come il gruppo di persone che hanno acquisito competenze utili all'azienda (ciò è evidente dalla percentuale di aumento di stipendio molto alta rispetto a tutti gli altri gruppi) in altri ambienti. La classificazione fatta non è certamente da considerarsi assoluta, poiché dai valori della silhouette non elevati, si capisce che la distinzione tra i cluster non è netta: mescolamenti tra di essi sono molto plausibili. Ragionamenti simili possono essere applicati agli altri grafici.

4.1.1 Validazione

Per quanto riguarda la validazione dei cluster ottenuti, si è valutata la distribuzione dell'SSE e della silhouette generati dal K-Means su un numero $N = 500$ di set di dati random estratti dallo stesso dominio degli attributi usati. La clusterizzazione è quindi ritenuta non random se l'SSE e la silhouette sono al di fuori di quattro deviazioni standard dal valor medio, ciò assicura che i dati ottenuti sono esterni al 99.994% delle relative distribuzioni. Dalle immagini seguenti si evince che tutti i valori ottenuti sono compatibili con una clusterizzazione non random.

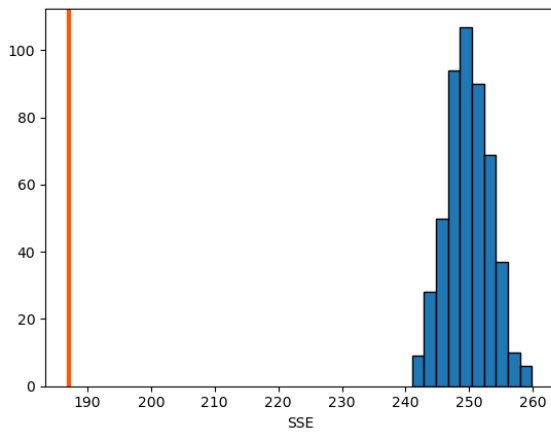


Figura 19: Distribuzione dell'SSE su 500 set di dati random per il sottoinsieme 1. Media: 250; Deviazione Standard: 3. La linea arancione rappresenta l'SSE sui dati reali.

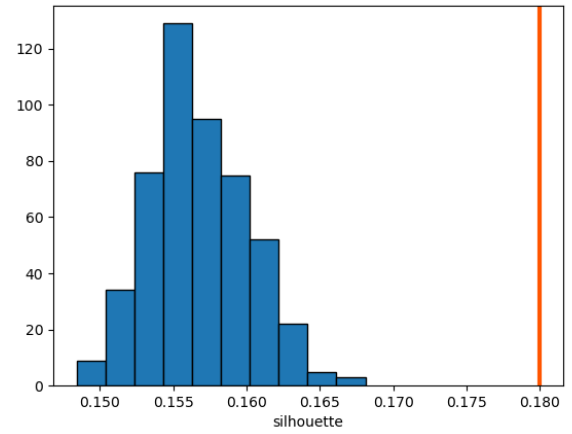


Figura 20: Distribuzione della silhouette su 500 set di dati random per il sottoinsieme 1. Media: 0.156; Deviazione Standard: 0.003. La linea arancione rappresenta la silhouette sui dati reali.

4.2 DB-Scan

Per quanto riguarda il DB-Scan i sottoinsiemi che sono stati utilizzati :

1. PercentSalaryHike, FractionYearsAtCompany, YearsInCurrentRole, RateIncome
2. PercentSalaryHike, FractionYearsAtCompany, RateIncome, YearsInCurrentRole, NumCompaniesWorked
3. DistanceFromHome, FractionYearsAtCompany, RateIncome, YearsInCurrentRole, TrainingTimesLastYear

La sensibilità dell'algoritmo alla scelta dei parametri è molto più elevata rispetto al K-Means. L'approccio utilizzato è stato quello di far variare il numero di min samples tra 5 e 20 e, attraverso il grafico dell'elbow curve, è stato scelto il valore di *eps* in prossimità del gomito. Successivamente è stato fatto variare il valore di *eps* su una scala molto fine in un suo intorno.

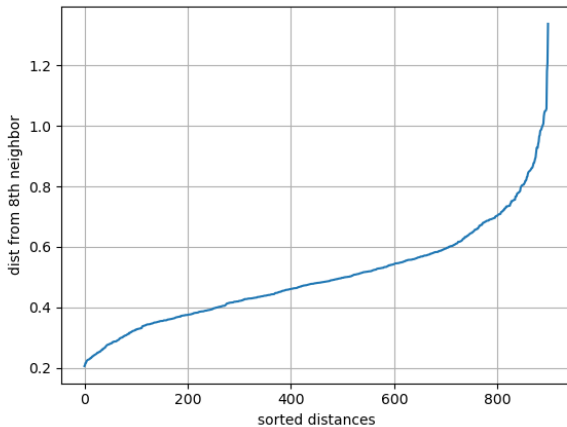


Figura 21: Sottoinsieme 1. Elbow curve con min samples = 8

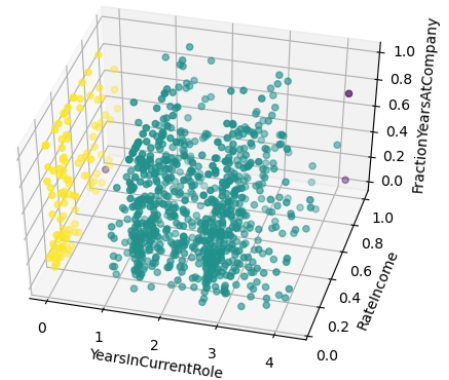


Figura 22: Sottoinsieme 1. $\epsilon = 0.750$, silhouette = 0.26

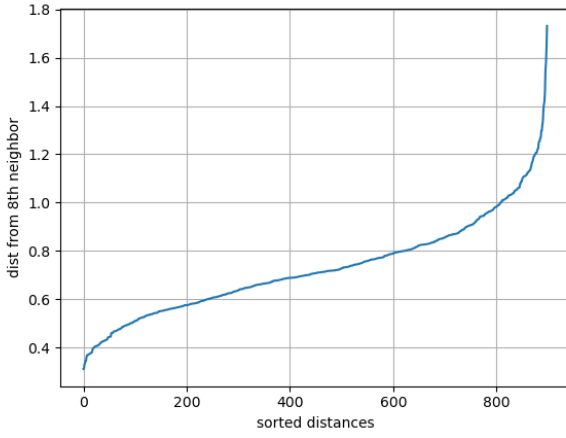


Figura 23: Sottinsieme 2. Elbow curve con min samples = 8

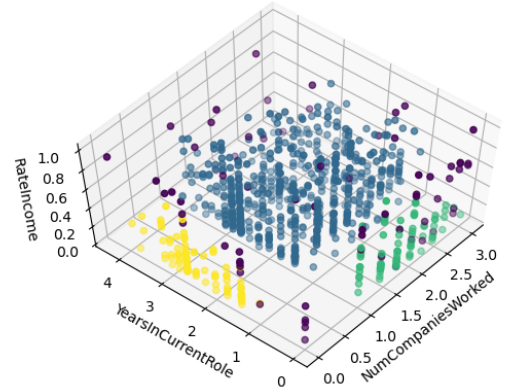


Figura 25: Sottinsieme 2. eps = 0.950, silhouette = 0.12

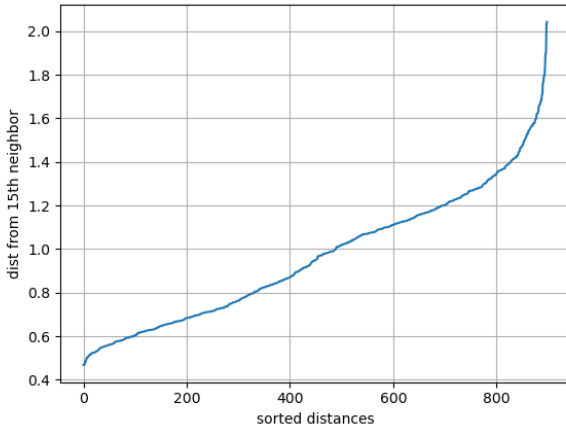


Figura 24: Sottinsieme 3. Elbow curve con min samples = 15

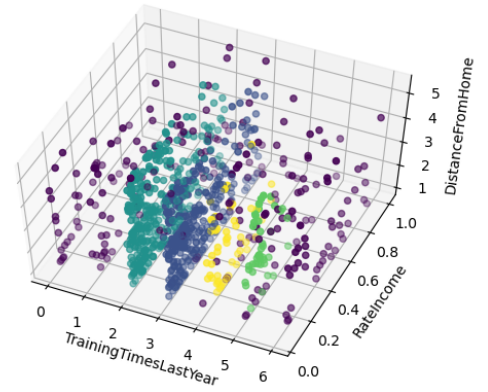


Figura 26: Sottinsieme 3. eps = 1.20, silhouette = 0.02

Come si può notare dai grafici l'algoritmo è riuscito ad evidenziare un diverso numero di cluster ed un certo numero di noise points (punti viola). Questo è essenzialmente dovuto alla distribuzione dei valori nello spazio tridimensionale: essi formano infatti delle slices molto dense a valori fissati di uno dei tre attributi, come si può ad esempio osservare nella Figura 26. In questo caso inoltre si nota che il valore della silhouette è praticamente nullo e ciò implica che la divisione ottenuta è totalmente inconsistente. Per tali motivi non è stato possibile associare ai cluster alcun significato.

4.2.1 Validazione

La validazione è stata eseguita anche in questo caso tramite la costruzione di $N_1 = 3.5 \times 10^4$ e $N_2 = 10^6$ distribuzioni random su cui si è valutata la silhouette. La clusterizzazione è stata considerata non random se il valore ottenuto della silhouette è al di fuori di quattro deviazioni standard dal valor medio, quindi esterno al 99.994% della distribuzione. Come si può notare dal secondo grafico i risultati ottenuti sono da scartare.

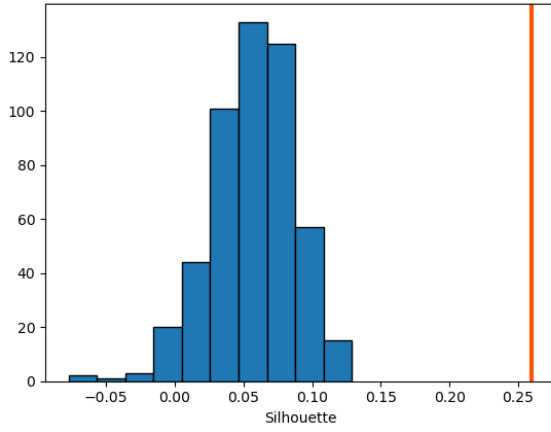


Figura 27: Distribuzione della silhouette su N_1 per il sottoinsieme 1. Media: 0.06; Deviazione Standard: 0.03. La linea arancione rappresenta la silhouette sui dati reali.

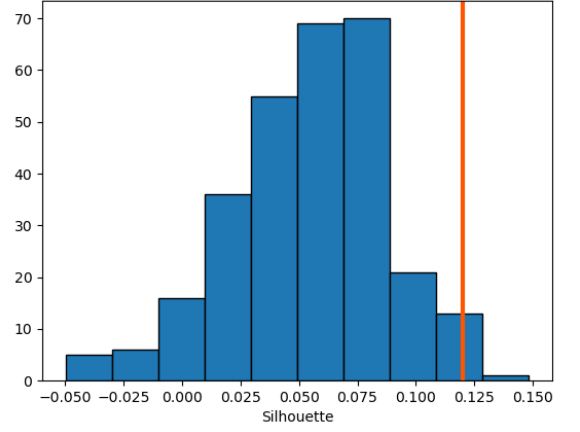


Figura 28: Distribuzione della silhouette su N_2 per il sottoinsieme 2. Media: 0.06; Deviazione Standard: 0.03. La linea arancione rappresenta la silhouette sui dati reali.

4.3 Hierarchical

Per quanto riguarda il Hierarchical i sottoinsiemi che hanno mostrato i risultati migliori sono:

1. DistanceFromHome, Age, MonthlyIncome
2. PercentSalaryHike, TrainingTimeLastYear, MonthlyIncome, Age

Per entrambi sono stati usati algoritmi di tipo agglomerativo con diverse definizioni della distanza inter clusters. Ai fini della scelta del metodo da utilizzare, come primo approccio, si è deciso di visualizzare lo scatter plot dei data frame alla ricerca di strutture che possano suggerire una scelta appropriata. Dall'analisi non sono emerse particolari evidenze e la decisione è ricaduta sui metodi che si sono rivelati più performanti. Di seguito sono riportati i metodi usati per ciascun sottoinsieme e i rispettivi dendrogrammi.

- Sottoinsieme 1: Group Average, Ward.
- Sottoinsieme 2: Median, Ward.

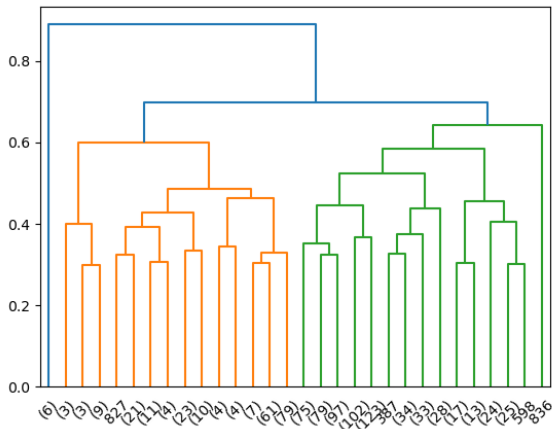


Figura 29: Dendrogramma sottoinsieme 1: metodo = Group Average; silhouette = 0.35

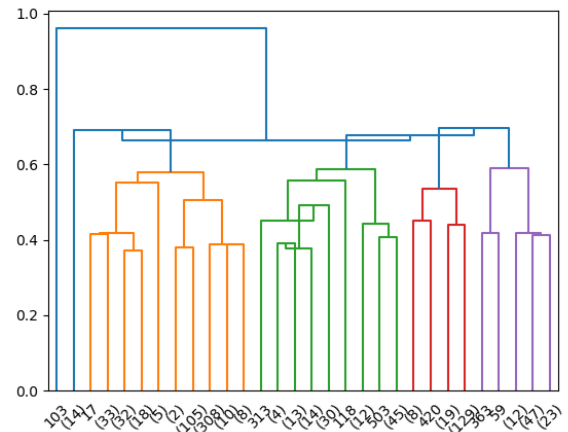


Figura 30: Dendrogramma sottoinsieme 2: metodo = Median; silhouette = 0.37

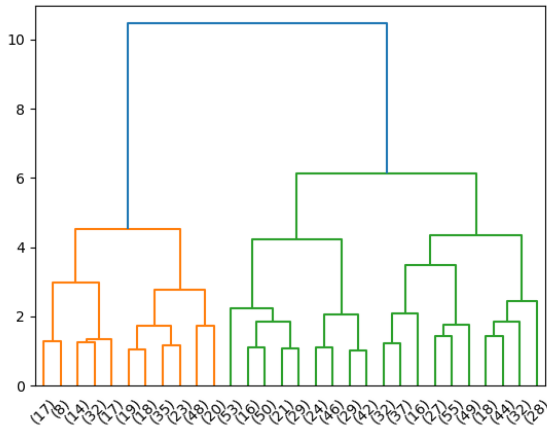


Figura 31: Dendrogramma sottoinsieme 1: metodo = Ward; silhouette = 0.35

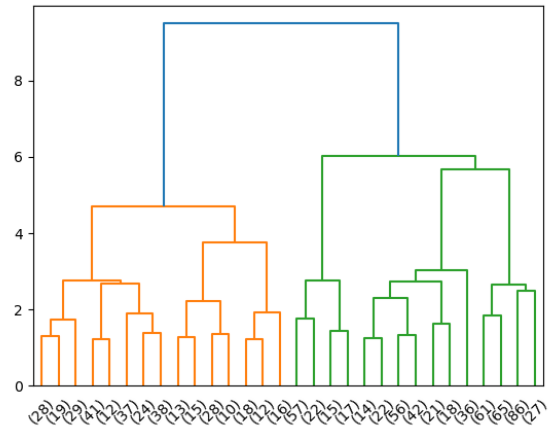


Figura 32: Dendrogramma sottoinsieme 2: metodo = Ward; silhouette = 0.24

Nei dendogrammi mostrati è possibile notare le differenze dei vari metodi. In generale con Ward la robustezza risultante è maggiore, sebbene nel secondo sottoinsieme la silhouette calcolata è significativamente più bassa rispetto all'altro metodo.

4.3.1 Validazione

Anche in questo caso per validare i risultati ottenuti, sono stati generati $N = 500$ set di dati random su cui è stato eseguito l'algoritmo Hierarchical e calcolata la silhouette. Se quella valutata sui dati reali cade al di fuori della distribuzione (media \pm quattro deviazioni standard) allora il risultato è stato considerato attendibile.

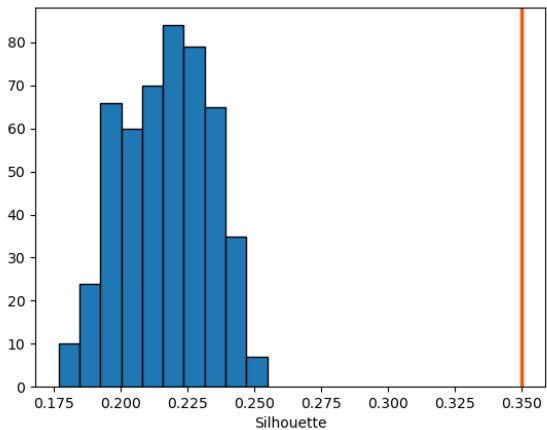


Figura 33: Distribuzione della silhouette su 500 set di dati random per il sottoinsieme 1. Media: 0.22; Deviazione Standard: 0.02. La linea arancione rappresenta la silhouette sui dati reali.

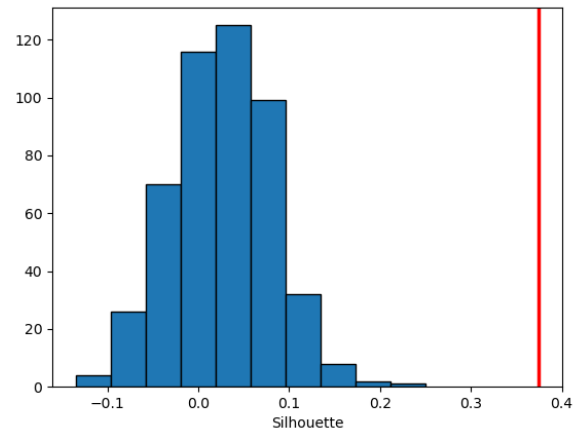


Figura 34: Distribuzione della silhouette su 500 set di dati random per il sottoinsieme 2. Media: 0.03; Deviazione Standard: 0.08. La linea arancione rappresenta la silhouette sui dati reali.

4.4 Confronto fra gli algoritmi

Alla luce di quanto ottenuto dai diversi algoritmi è possibile confrontare la bontà dei risultati ottenuti. Sicuramente il DB-Scan è quello che ha avuto la resa peggiore, in quanto, come già precisato, soffre a causa della distribuzione dei dati all'interno dello spazio m -dimensionale.

I risultati migliori in termini della silhouette sono relativi al K-Means e al Hierarchical. Inoltre, tramite quest'ultimo, è possibile scegliere il numero di clusters da utilizzare nel primo in modo da ottimizzarne i risultati. Questo metodo è stato applicato per il sottoinsieme 2 usato nel Hierarchical. Ne è riportato lo scatter plot ricavato dal K-Means con $K = 4$.

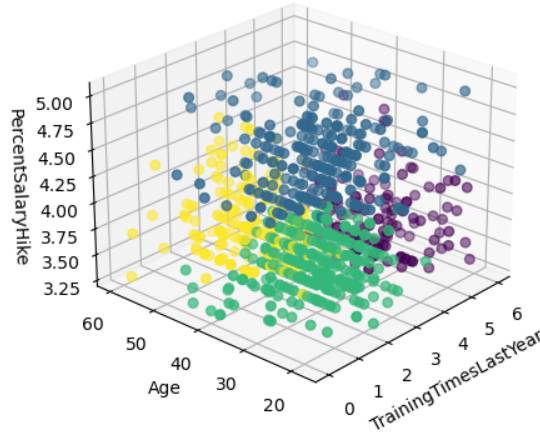


Figura 35: Cluster sottoinsieme 2(par 4.3): SSE=96; silhouette=0.24

I risultati sono nettamente migliori in termini dell'SSE.

5 Classification

Lo scopo in questo caso è quello di determinare modelli capaci di classificare i records presenti nel data frame in funzione dell'attributo Attrition. Sono stati usati due algoritmi differenti: Decision Tree e K-Nearest Neighbors. Per il primo il data frame considerato è stato quello precedente alla trasformazioni delle variabili, quindi senza missing values e con l'eliminazione del maggior numero di incongruenze possibili fra i dati. Ciò è stato fatto poiché la classificazione tramite Decision Tree non soffre se le distribuzioni dei dati sono sbilanciate e/o molto sparse. Per il secondo, invece, si è utilizzato il data frame con le variabili trasformate. In entrambi gli algoritmi i valori degli attributi categorici ordinali maggiormente correlati con Attrition (Figura 8) sono stati sostituiti con valori numerici in maniera tale da ottenere una divisione binaria mantenendo l'ordinamento:

- Gender: 'Female'=0, 'Male'=1.
- OverTime: 'No'=0, 'Yes'=1.
- EnvironmentSatisfaction: 'Low'=0, 'Medium'=0, 'High'=1, 'Very High'=1.
- WorkLifeBalance: 'Bad'=0, 'Good'=0, 'Better'=1, 'Best'=1.
- JobInvolvement: 'Low'=0, 'Medium'=0, 'High'=1, 'Very High'=1.
- MaritalStatus: 'Single'=0, 'Divorced'=0, 'Married'=1.

Siccome i valori dell'attributo Attrition sono fortemente sbilanciati a favore dei 'No' (84%), si è deciso di effettuare un random oversampling delle istanze relative al valore 'Yes' in modo da ridurre il bias presente nei dati. In particolare la frequenza del valore 'Yes' è passata dal 16% al 23% nel Decision Tree, mentre nel KNN dal 16% al 20%. Il data frame è stato suddiviso in train set (70%) e test set (30%). Inoltre ai fini della selezione del modello migliore è stata utilizzata la cross validation, in cui il train set è ulteriormente diviso in tre blocchi di cui due vengono usati per l'effettivo training ed uno, ciclicamente, per la validazione.

5.1 Decision Tree

Un classificatore di tipo *DecisionTree* è caratterizzato da un diverso numero di parametri. Per la loro scelta è stata utilizzata la cross-validation in modo da ottenere la combinazione più performante degli stessi. I parametri scelti e il relativo range sono i seguenti:

- min impurity decrease $\in [0, 0.2]$: il minimo gain dell'impurità affinché il nodo venga splittato;
- max depth $\in [2, 8]$: massima profondità dell'albero;
- min samples split $\in [2, 8]$: numero minimo di valori di un nodo affinché esso venga splittato;
- min samples leaf $\in [2, 8]$: numero minimo di valori di un nodo affinché esso sia un nodo leaf;

Le migliori combinazioni che ottimizzano rispettivamente la *Precision* e la *Recall* risultano essere:

- Precision: min impurity decrease= 0.0024; max depth= 7; min samples split= 5; min samples leaf= 1; $\alpha=0$.
- Recall: min impurity decrease= 0.0; max depth= 7; min samples split= 3; min samples leaf= 1; $\alpha=0$.

- Tali valori sono stati ulteriormente modificati per ridurre la complessità degli alberi generati sempre con lo scopo di migliorare le performance. Infatti, per come è stata implementata la cross-validation, Precision e Recall vengono ottimizzati singolarmente e quindi è possibile avere set di parametri maggiormente adatti agli scopi in esame che non erano stati indagati. I modelli ottenuti sono:

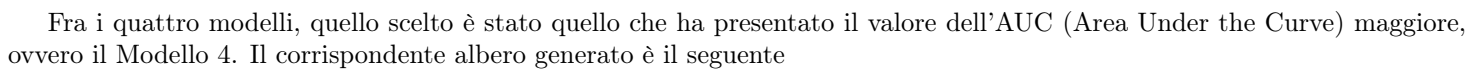
- Modello 1: min impurity decrease= 0.0; max depth= 8; min samples split= 5; min samples leaf= 2;
- Modello 2: min impurity decrease= 0.0; max depth= 3; min samples split= 2; min samples leaf= 1;
- Modello 3: min impurity decrease= 0.001; max depth= 12; min samples split= 5; min samples leaf= 1;
- Modello 4: min impurity decrease= 0.003; max depth= 6; min samples split= 2; min samples leaf= 1;



ROC curve showing True Positive Rate (Y-axis) versus False Positive Rate (X-axis) for four models. The dashed diagonal line represents random performance. The legend indicates the area under each curve:

- Modello 1 (area = 0.72)
- Modello 2 (area = 0.59)
- Modello 3 (area = 0.72)
- Modello 4 (area = 0.76)

Figura 36: ROC curve dei quattro modelli selezionati.



```

graph TD
    Root["NumCompaniesWorked <= 4.5  
gini = 0.329  
samples = 669  
value = [530, 139]  
class = No"]
    Root -- True --> Node1["FractionYearsAtCompany <= 0.894  
gini = 0.27  
samples = 497  
value = [417, 80]  
class = No"]
    Root -- False --> Node2["JobInvolvement <= 0.5  
gini = 0.451  
samples = 172  
value = [113, 59]  
class = No"]
    
    Node1 --> Node3["JobLevel <= 1.5  
gini = 0.23  
samples = 459  
value = [388, 51]  
class = No"]
    Node1 --> Node4["RateIncome <= 0.181  
gini = 0.5  
samples = 38  
value = [19, 19]  
class = No"]
    
    Node2 --> Node5["WorkLifeBalance <= 0.5  
gini = 0.496  
samples = 64  
value = [29, 35]  
class = Yes"]
    Node2 --> Node6["EnvironmentSatisfaction <= 0.5  
gini = 0.346  
samples = 108  
value = [84, 24]  
class = No"]
    
    Node3 --> Node7["OverTime <= 0.5  
gini = 0.377  
samples = 111  
value = [83, 28]  
class = No"]
    Node3 --> Node8["DistanceFromHome <= 11.5  
gini = 0.172  
samples = 348  
value = [315, 33]  
class = No"]
    
    Node4 --> Node9["gini = 0.0  
samples = 6  
value = [8, 0]  
class = No"]
    Node4 --> Node10["DistanceFromHome <= 5.5  
gini = 0.482  
samples = 32  
value = [13, 19]  
class = Yes"]
    
    Node5 --> Node11["PercentSalaryHike <= 18.0  
gini = 0.366  
samples = 29  
value = [7, 22]  
class = Yes"]
    Node5 --> Node12["StockOptionLevel <= 0.5  
gini = 0.461  
samples = 35  
value = [22, 13]  
class = No"]
    
    Node6 --> Node13["StockOptionLevel <= 1.5  
gini = 0.476  
samples = 46  
value = [28, 18]  
class = No"]
    Node6 --> Node14["gini = 0.175  
samples = 62  
value = [56, 6]  
class = No"]
    
    Node7 --> Node15["YearsInCurrentRole <= 12.5  
gini = 0.24  
samples = 79  
value = [68, 11]  
class = No"]
    Node7 --> Node16["MaritalStatus <= 0.5  
gini = 0.498  
samples = 32  
value = [15, 17]  
class = Yes"]
    
    Node8 --> Node17["gini = 0.093  
samples = 246  
value = [234, 12]  
class = No"]
    Node8 --> Node18["EnvironmentSatisfaction <= 0.5  
gini = 0.327  
samples = 102  
value = [81, 21]  
class = No"]
    
    Node9 --> Node19["gini = 0.0  
samples = 77  
value = [68, 9]  
class = No"]
    Node9 --> Node20["gini = 0.0  
samples = 2  
value = [0, 0]  
class = Yes"]
    
    Node10 --> Node21["gini = 0.346  
samples = 9  
value = [7, 2]  
class = No"]
    Node10 --> Node22["gini = 0.386  
samples = 23  
value = [6, 17]  
class = Yes"]
    
    Node11 --> Node23["gini = 0.26  
samples = 28  
value = [4, 22]  
class = Yes"]
    Node11 --> Node24["gini = 0.0  
samples = 3  
value = [3, 0]  
class = No"]
    
    Node12 --> Node25["FractionYearsAtCompany <= 0.339  
gini = 0.49  
samples = 21  
value = [9, 12]  
class = Yes"]
    Node12 --> Node26["gini = 0.133  
samples = 14  
value = [13, 1]  
class = No"]
    
    Node13 --> Node27["RateIncome <= 0.401  
gini = 0.375  
samples = 32  
value = [24, 8]  
class = No"]
    Node13 --> Node28["Gender <= 0.5  
gini = 0.408  
samples = 14  
value = [4, 10]  
class = Yes"]
    
    Node14 --> Node29["gini = 0.0  
samples = 4  
value = [0, 0]  
class = No"]
    Node14 --> Node30["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node15 --> Node31["gini = 0.206  
samples = 7  
value = [3, 1]  
class = No"]
    Node15 --> Node32["gini = 0.375  
samples = 4  
value = [3, 1]  
class = No"]
    
    Node16 --> Node33["gini = 0.375  
samples = 16  
value = [2, 14]  
class = Yes"]
    Node16 --> Node34["gini = 0.219  
samples = 16  
value = [9, 12]  
class = Yes"]
    
    Node17 --> Node35["gini = 0.0  
samples = 12  
value = [10, 2]  
class = No"]
    Node17 --> Node36["gini = 0.278  
samples = 20  
value = [5, 15]  
class = Yes"]
    
    Node18 --> Node37["StockOptionLevel <= 0.5  
gini = 0.444  
samples = 48  
value = [32, 16]  
class = No"]
    Node18 --> Node38["gini = 0.168  
samples = 54  
value = [46, 5]  
class = No"]
    
    Node19 --> Node39["gini = 0.278  
samples = 6  
value = [5, 1]  
class = No"]
    Node19 --> Node40["RateIncome <= 0.526  
gini = 0.391  
samples = 15  
value = [4, 11]  
class = Yes"]
    
    Node20 --> Node41["gini = 0.0  
samples = 13  
value = [0, 0]  
class = No"]
    Node20 --> Node42["gini = 0.0  
samples = 4  
value = [0, 4]  
class = Yes"]
    
    Node21 --> Node43["gini = 0.278  
samples = 13  
value = [2, 11]  
class = Yes"]
    Node21 --> Node44["gini = 0.0  
samples = 2  
value = [0, 0]  
class = No"]
    
    Node22 --> Node45["gini = 0.391  
samples = 15  
value = [11, 4]  
class = No"]
    Node22 --> Node46["gini = 0.0  
samples = 4  
value = [0, 4]  
class = Yes"]
    
    Node23 --> Node47["gini = 0.0  
samples = 19  
value = [11, 8]  
class = No"]
    Node23 --> Node48["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    
    Node24 --> Node49["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node24 --> Node50["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node25 --> Node51["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node25 --> Node52["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node26 --> Node53["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node26 --> Node54["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node27 --> Node55["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node27 --> Node56["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node28 --> Node57["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node28 --> Node58["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node29 --> Node59["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node29 --> Node60["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node30 --> Node61["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node30 --> Node62["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node31 --> Node63["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node31 --> Node64["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node32 --> Node65["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node32 --> Node66["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node33 --> Node67["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node33 --> Node68["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node34 --> Node69["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node34 --> Node70["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node35 --> Node71["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node35 --> Node72["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node36 --> Node73["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node36 --> Node74["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node37 --> Node75["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node37 --> Node76["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node38 --> Node77["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node38 --> Node78["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node39 --> Node79["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node39 --> Node80["gini = 0.0  
samples = 10  
value = [0, 10]  
class = Yes"]
    
    Node40 --> Node81["gini = 0.0  
samples = 13  
value = [0, 10]  
class = No"]
    Node40 --> Node82["gini = 0.0  
samples = 10  
value = [0, 10]
```

Si può vedere come gli attributi che maggiormente contribuiscono alla divisione dello spazio n -dimensionale sono quelli relativi vita lavorativa dei dipendenti: molto probabilmente aver lavorato per diverse aziende in passato influenza marcatamente lo stato motivazionale degli impiegati, così come la quantità di tempo passata nell'azienda rispetto agli anni totali di lavoro. Ancor più evidente è il ruolo di JobInvolvement, al quale consegue la divisione fra i dipendenti più e meno coinvolti nel proprio lavoro.

Applicando tale modello sul test set si sono ottenuti ricavati risultati che suggeriscono una buona performance del *DecisionTree* selezionato.

		Predicted class	
		No	Yes
Actual Class	No	237	25
	Yes	31	37

Figura 38: Confusion Matrix del test set. 'No': Precision= 0.88, Recall= 0.90; 'Yes': Precision= 0.60, Recall= 0.54

Come precedentemente detto, l'approccio dell'oversampling è stato utilizzato per ridurre il bias che il set di dati presentava a favore dei 'No' nell'attributo Attrition. Si verifica, infatti, che applicando le stesse procedure sul data frame originale i risultati che si ottengono sono nettamente peggiori, come si può notare dalla Confusion Matrix in Figura 39.

Si vuole precisare che in un primo momento, in luogo dell'oversampling, è stato applicato un peso alla classe 'Yes' ad ogni nodo dell'albero, sempre con l'idea di bilanciare i dati. Seguendo questa strada, comunque, non sono stati ottenuti grandi miglioramenti: l'oversampling è risultato quindi il metodo migliore ai fini dell'ottimizzazione.

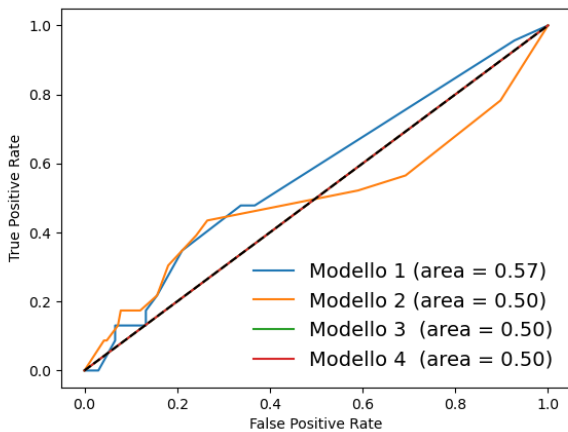


Figura 39: ROC curve relativi a quattro modelli selezionati con il data frame originale.

		Predicted class	
		No	Yes
Actual Class	No	231	7
	Yes	30	2

Figura 40: Confusion Matrix del test set. 'No': Precision= 0.89, Recall= 0.97; 'Yes': Precision= 0.22, Recall= 0.06

5.2 K-Nearest Neighbors

A differenza del caso precedente i dati sono stati normalizzati conservando la deviazione standard dei valori di partenza. Gli unici parametri da dover scegliere in maniera opportuna sono il numero di primi vicini k ed il peso da assegnare a ciascuno di essi relativamente alla loro distanza. Con lo scopo di ridurre l'impatto di k sui risultati si è deciso di utilizzare come peso l'inverso del quadrato della distanza. Infatti, in tal modo, si dà maggior importanza ai primi vicini più vicini. Il range in cui è stato fatto variare k ai fini della massimizzazione delle performance del modello è stato $k \in [1, 150]$. In seguito alla cross-validation, il miglior valore è risultato essere, come si può intuire dai grafici che seguono, $k = 9$.

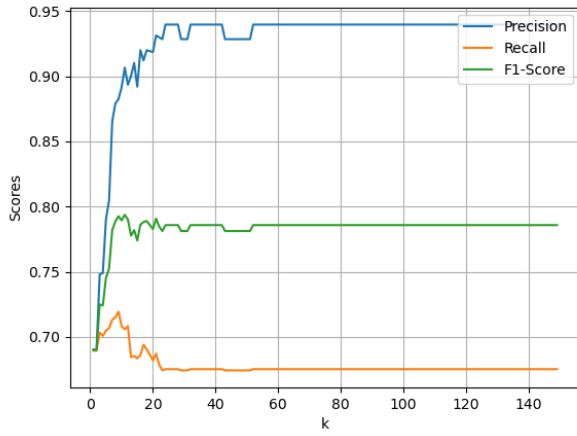


Figura 41: Risultati della cross-validation al variare del parametro k .

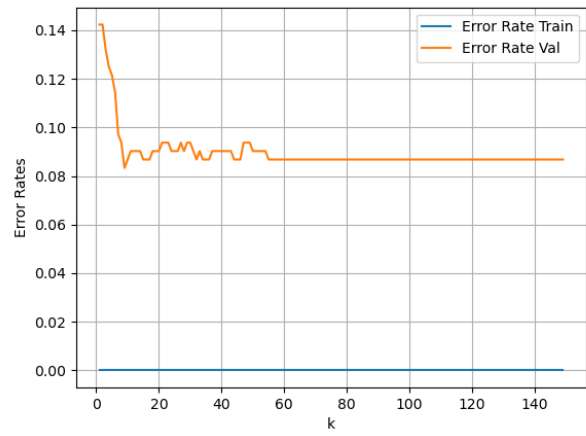


Figura 42: Andamento dell'error rate sul train set e sul validation set. Si può notare l'overfitting del modello per piccoli valori di k .

In Figura 41 si evince come il compromesso migliore fra i tre scores risulta essere $k = 8$, dove si nota il massimo dell'F1-Score che rappresenta quindi un buon bilanciamento fra Precision e Recall. Questo valore corrisponde anche al minimo del validation error rate in Figura 42. Di seguito è riportato il risultato del modello selezionato sul test set, tramite la Confusion Matrix.

		Predicted class	
		No	Yes
Actual Class	No	235	3
	Yes	21	29

Figura 43: Confusion Matrix del test set. 'No': Precision=0.92, Recall=0.98; 'Yes': Precision=0.91, Recall=0.58

Si può notare come, sebbene l'approccio dell'oversampling abbia sanato parzialmente lo squilibrio fra i due valori di Attrition, resta comunque un bias in favore dei 'No'.

6 Association Rules

Lo scopo è quello di ricavare le *Association Rules* più interessanti nei dati disponibili. Poiché il numero degli itemsets e di conseguenza il numero delle regole cresce esponenzialmente al variare dei valori, sono stati scartati gli attributi considerati meno importanti, ovvero: 'PerformanceRating', 'TrainingTimesLastYear', 'StockOptionLevel', 'YearsInCurrentRole', 'NumCompaniesWorked', 'PercentSalaryHike', 'OverTime', 'RateIncome', 'FractionYearsAtCompany'.

L'unico attributo numerico continuo sopravvissuto, 'DistanceFromHome', è stato categorizzato in 10 valori tramite la suddivisione in intervalli uguali. Ai fini dell'utilizzo delle regole di associazione che verranno determinate per la predizione di determinati target class e della conseguenza valutazione della loro bontà, è stato suddiviso il data frame in due sottoinsiemi: uno di train (630 elementi) ed uno di test (269 elementi). Come prima cosa sono stati determinati gli itemsets più frequenti, i massimali ed i chiusi al variare del valore minimo del supporto. Come si può osservare dal grafico in Figura 44 all'aumentare del supporto minimo il numero degli itemsets decresce velocemente, in particolare seguono un andamento esponenziale decrescente.

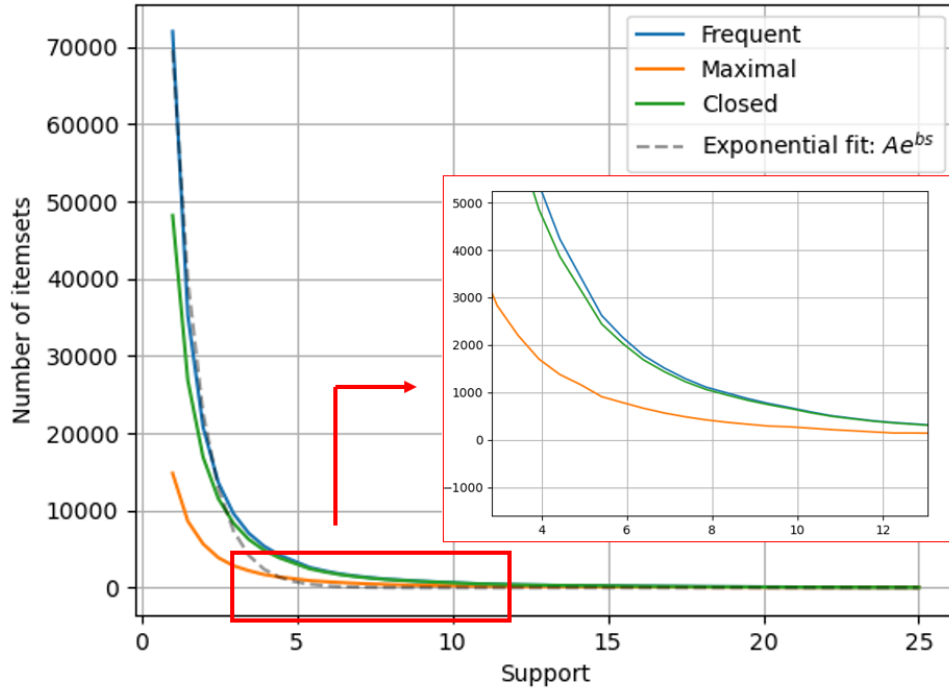


Figura 44: Numero dei diversi tipi di itemsets in funzione del supporto minimo.
Valori del fit: $A = 219226$; $b = -1.15$.

Successivamente si è valutato il numero di rules in funzione della confidence per diversi valori del supporto minimo. Come ci si poteva aspettare il numero di regole trovate diminuisce all'aumentare della confidence e del supporto minimo.

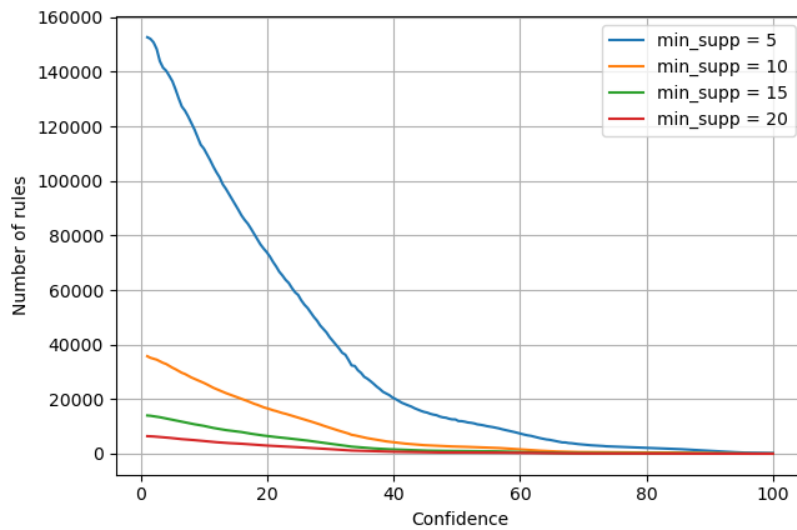


Figura 45: Numero delle regole in funzione della confidence per 4 diversi valori del supporto minimo.

Per le fasi seguenti è stato scelto come supporto il valore 10 per avere un numero consono (633) di itemsets da analizzare. Come misura di interesse per ciascuna regola è stato utilizzato l'*InterestFactor* ('*lift*'), di seguito è presentato l'istogramma del numero delle regole al variare del lift.

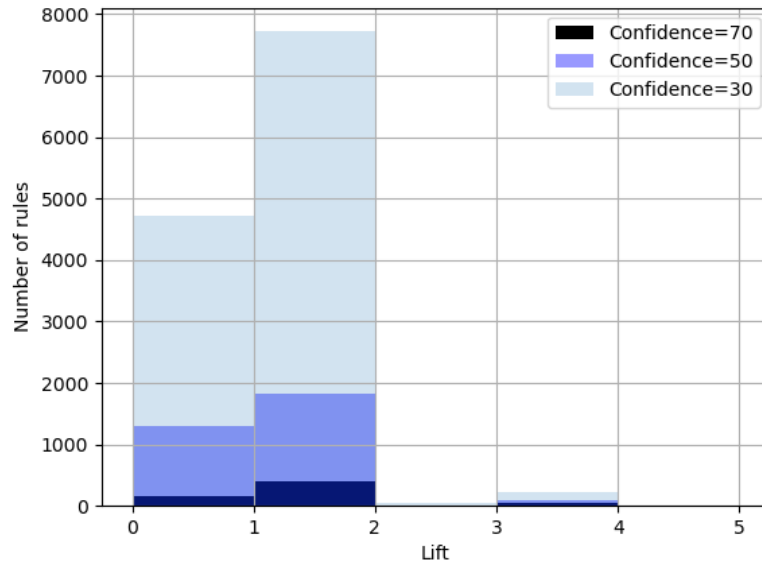


Figura 46: Istogramma del numero delle regole al variare del *lift* per 3 valori della confidence: 30%; 50%; 70%

Di conseguenza le regole più interessanti scelte risultano essere quelle con lift maggiore di 3, siccome il numero di regole significativamente minore di 1 è nullo. Le regole selezionate risultano essere:

1. (3.0, 'Research and Development', 'No') \rightarrow 'Research Director'
2. (1.0, 'Life Science', 'Research and Development') \rightarrow 'Research Scientist'
3. ('Sales', 2.0, 'Better_WorkLifeBalance') \rightarrow 'Sales Executive'
4. ('Sales', 2.0, 'No') \rightarrow 'Sales Executive'

Fra queste, quelle ulteriormente indagate sono state la 1 e la 3 aventi i lift più alti, utilizzate per predire la class label dell'attributo Department. Il tutto è stato applicato sul test set ed i risultati sono mostrati nelle seguenti tabelle di contingenza in cui **A** (**\bar{A}**) e **B** (**\bar{B}**) rappresentano precedente (precedente negato) ed antecedente (antecedente negato).

	B	\bar{B}	
A	3	27	30
\bar{A}	0	10	10
	3	37	40

	B	\bar{B}	
A	28	2	30
\bar{A}	0	56	56
	28	58	86

Figura 47: Tabella di contingenza regola 1. Accuratezza= 10% Figura 48: Tabella di contingenza regola 3. Accuratezza= 93.3%

Dal valore delle accuratze è evidente che la regola numero 3 è più performante della 1.

Utilizzando un procedimento analogo è possibile determinare regole per selezionare il valore di 'Attriction', sfruttando in questo caso non più gli attributi numerici, ma attributi categorici (non ordinali) non utilizzati nella costruzione dei modelli di classificazione. Le regole determinate, aventi i lift più alti, sono:

1. ('Low_EnvironmentSatisfaction', 'Better_WorkLifeBalance') \rightarrow 'Yes'; Lift=1.86
2. ('Low_EnvironmentSatisfaction', 'Male') \rightarrow 'Yes'; Lift=1.96
3. (1.0, 'Life Sciences') \rightarrow 'Yes'; Lift=1.84
4. (1.0, 'High_JobEnvolvement', 'Research and Development') \rightarrow 'Yes'; Lift=1.86
5. ('High_RelationshipSatisfaction', 'Sales') \rightarrow 'Yes'; Lift=1.96

La prima regola suggerisce che questi impiegati, sebbene abbiano un buon equilibrio tra la vita privata e quella lavorativa, la bassa soddisfazione dell'ambiente di lavoro, aumenta la probabilità di essere logorati. Discorso analogo per l'ultima, dove si evince che il dipartimento 'Sales' è particolarmente stressante per indipendenti nonostante esistano buoni rapporti tra colleghi. Per le predizioni sono state usate le regole 1 e 5, di seguito sono riportate le tabelle di contingenza.

	B	$\bar{\mathbf{B}}$	
A	4	27	31
$\bar{\mathbf{A}}$	9	78	87
	13	105	118

Figura 49: Tabella di contingenza regola 1. Accuratezza= 13.5%

	B	$\bar{\mathbf{B}}$	
A	2	17	19
$\bar{\mathbf{A}}$	12	126	138
	14	143	157

Figura 50: Tabella di contingenza regola 5. Accuratezza= 10.5%

L'accuratezza delle regole ricavate non è molto significativa (rispettivamente 13.5% e 10.5%), il che comporta quindi che le prestazioni ottenute sul test set sono basse: utilizzare tali associations rules non porta a predizioni soddisfacenti.

7 Conclusioni

L'obiettivo iniziale della trattazione era studiare in maniera generale i dati disponibili al fine di ricavare informazioni interessanti da essi. Nella prima fase è stato studiato il dataframe in generale, analizzando la tipologia degli attributi ed il loro significato. Valutate poi le distribuzioni dei dati, si è proceduto alla loro manipolazione ai fini di migliorare le successive indagini effettuate. Dagli algoritmi di clustering utilizzati sono emerse, sebbene non in maniera ottimale, partizionamenti dei dati che hanno suggerito prime relazioni fra gli attributi. I risultati maggiormente incoraggianti sono stati quelli ottenuti tramite un approccio gerarchico, soprattutto quando sono stati poi utilizzati per l'inizializzazione dell'algoritmo K-Means. Ulteriori relazioni sono stante evinte per mezzo di classificatori eager learn (Decision Tree) e lazy learn (KNN), che hanno indagato più dettagliatamente i valori degli attributi, con lo scopo di predire ora lo stato di logoramento dei dipendenti descritti dai dati: il KNN è risultato il più capace fra i due, ottenendo quindi risultati migliori su dati esterni al training. In conclusione, attraverso le Association Rules e con il medesimo scopo, patterns frequenti e regole interessanti sono risultate sensibilmente meno opportune.