Capitolo 1

Data Understanding

Credit Card Default è il dataset utilizzato per questa analisi che si pone l'obiettivo di classificare ed identificare i clienti di una banca di Taiwan in base ai loro default payments o pagamenti in stato di insolvenza, relativi alla carta di credito personale. Dal punto di vista della gestione del rischio, il risultato della classificazione sarà prezioso per identificare clienti credibili o non credibili. Il periodo a cui si riferiscono i dati in possesso va da Aprile a Settembre 2005 e la valuta di riferimento è il dollaro taiwanese (NT$). Il datasetet contiene 10.000 records descritti da 24 colonne di attributi.

1.1 Analisi degli attributi

Di seguito (Tabella 1.1) vengono riportati e descritti gli attributi raggruppandoli in base alla loro tipologia e indicando per ciascuno di essi il loro dominio.

Tipologia		Attributo	Descrizione	Dominio
Numerici	Discreti	age	Indica l'età anagrafica del titolare della carta di credito.	[21, 75] ∩ $\mathbb{N}$
		limit	Indica l'importo limite in NT$ della carta di credito di ciascun titolare.	[10 000, 780 000] ∩ $\mathbb{N}$
		ba-X, X ∈ {apr, may, jun, jul, aug, sep}	Bill Amount: indica l'importo in NT$ da pagare relativo alle spese effettuate nel mese X. (Valori negativi indicano di avere un credito positivo sulla carta al momento del conto)	[-209 051, 616 836] ∩ $\mathbb{Z}$.
		ps-X, X ∈ {apr, may, jun, jul, aug, sep}	Payment Status: indica lo stato di pagamento relativo alle spese del mese X. Legenda dei valori: -2: nessun utilizzo -1: pagamento per intero 0: utilizzo del credito rotativo, rateizzando l'importo [1,9]: indica di quanti mesi è in ritardo un pagamento	[-2, 8] ∩ $\mathbb{Z}$.
		pa-X, X ∈ {apr, may, jun, jul, aug, sep}	Payment Amount: indica l'importo effettivo pagato nel mese X+1 relativo a ba-X.	[0, 1 227 082] ∩ $\mathbb{N}$
Categorici	Nominali	Status	Indica lo stato civile del titolare della carta di credito.	{single, married, others}
		Sex	Indica il sesso del titolare della carta di credito	{male, female}
	Ordinali	Education	Indica l'istruzione del titolare della carta di credito	{ others, graduate school, high school, university }
	Binari	Credit default	Indica se il titolare di carta di credito è in situazione di insolvenza del credito fornito dalla banca.	{yes, no}

Tabella 1.1: Attributi del dataset

1.2 Analisi della qualità dei dati

È stata analizzata la qualità dei dati all'interno del dataset fornito secondo i seguenti parametri:

Accuratezza Semantica
Dal punto di vista semantico sono stati analizzati nello specifico gli attributi `payment amount`, `payment status` e `billing amount`. L'attributo `payment amount` è stato confrontato con l'attributo `billing amount` e `payment status` per verificare a quale importo di pagamento si riferisse il valore indicato `pa-X`, ovvero se `pa-apr`, per esempio, indicasse un pagamento effettuato ad Aprile (quindi relativo al conto di Marzo) o un pagamento del conto di Aprile (ndicato da `ba-apr`) e quindi effettuato a Maggio. Attraverso tale valutazione si è concluso che il valore `pa-X` indica il pagamento della cifra indicata dal `billing amount` del mese precedente. Questo corrisponde alla realtà poichè utilizzando una carta di credito, il pagamento effettuato in un mese X è relativo all' importo da pagare del mese precedente. Riprendendo l'esempio precedente, il pagamento relativo al conto di Aprile (`ba-apr`) è indicato da `pa-may`. Da sottolineare la presenza di valori negativi nell'attributo `ba-X`, rappresentanti dal punto di vista semantico una situazione di credito e non di debito nei confronti della banca.

	limit	ps-sep	ps-aug	ps-jul	ps-jun	ps-may	ps-apr	ba-sep	ba-aug	ba-jul	ba-jun	ba-may	ba-apr	pa-sep	pa-aug	pa-jul	pa-jun	pa-may	pa-apr
157	460000	-1	-1	-1	-1	0	-1	1637	-196	4594	1517	1517	1306	0	4790	1517	0	1306	0

Figura 1.1: un esempio di record che ha permesso facilmente di riconoscere il valore semantico degli attributi relativi alle transazioni monetarie. Si può notare grazie ai valori di `ps-X`, come la cifra da pagare nel mese di Aprile, per esempio, sia equivalente alla cifra indicata nel pagamento di Maggio e lo stato di pagamento di aprile sia effettivamente -1, ovvero pagato completamente. Anche nei successivi mesi è possibile vedere un comportamento che corrisponde alla semantica degli attributi sopradescritta.

Da sottolineare però che per verificare il corretto valore degli attributi `pa-X`, `ba-X` e `ps-X` sarebbe necessario conoscere tutta la cronologia delle transazioni di una specifica carta di credito così da poter calcolare attraverso semplici operazioni aritmetiche i valori effettivi di spese e pagamenti. Pertanto si è deciso di assumere come corretti i dati considerando inoltre che essendo informazioni raccolte da un istituto bancario non dovrebbero essere presenti dati monetari errati.

Gestione di Outliers e Missing Values Per la ricerca di outliers è stato prodotto per ciascun attributo il relativo Boxplot, ma per la natura degli attributi del dataset, i valori rappresentati come outliers nel boxplot, non sono considerabili errati. Infatti per gli attributi numerici che rappresentano somme di denaro (`pa-X`, `ba-X` e `limit`), abbiamo considerato questi valori utili per le successive fasi di analisi sul dataset. Lo stesso ragionamento vale per gli attributi `ps-X`. Per quanto riguarda l'attributo `age` invece, abbiamo considerato non utile escludere i record per il fatto della loro bassa correlazione con qualsiasi altro attributo del dataset. L'utilizzo dei boxplot ha sollevato una problematica più facilmente osservabile del dataset, ovvero la presenza dei Missing Values:

attributi numerici nessun attributo numerico relativo al plafond (`limit`), allo stato del pagamento (`ps-X`) o alle transazioni della carta (`pa-X`, `ba-X`) presentano missing values.

Un attributo numerico che presenta missing value è l'attributo `age`, infatti all'interno del dataset sono presenti per tale attributo molteplici valori -1, valore non coerente con un attributo che indichi l'età anagrafica. È facilmente intuibili come tale valore indichi un valore mancante. Pertanto, tale valore è stato sostituito con la media dell'età anagrafica dei record del dataset, ovviamente non tenendo conto dei valori -1. La media corrisponde a 35.4 che (per coerenza con il tipo di dato intero dell'età) è stata troncata a 35 e settata in sostituzione del valore -1.

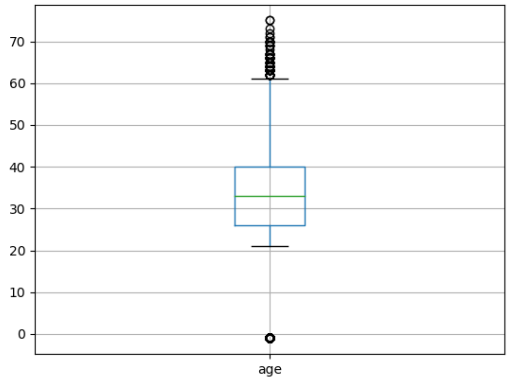

Figura 1.2: Boxplot dell'attributo age

attributi categorici : ad eccezione di `credit_default`, in ciascuno degli altri attributi categorici sono presenti missing values (nan), sostituiti dalla moda dell'attributo:

- `sex`: moda "female"
- `education`: moda "university"
- `status`: moda "single"

1.3 Normalizzazione delle variabili

Gli attributi `limit`, `ba-X` e `pa-X` rappresentando somme di denaro e pagamenti, hanno un ampio range di valori con massimo e minimo molto discostanti. Per utilizzare e visualizzare al meglio questi 3 attributi è stata utilizzata una normalizzazione **min-max**, trasformando il loro dominio in un range di valori continui [0,1].

1.4 Distribuzione delle variabili e analisi statistiche

Attributi Numerici
Per la visualizzazione della distribuzione di attributi numerici sono stati utilizzati degli istogrammi con curva gaussiana. Per gli attributi `limit`, `ba-X`, `pa-X` e `age` si è scelto un numero di bins ottimale pari a 15 utilizzando la regola di Sturge, mentre per gli attributi `ps-X`, avendo un ristretto range di valori [-2, 8] si è utilizzato un numero di bins pari al numero di valori dell'attributo.

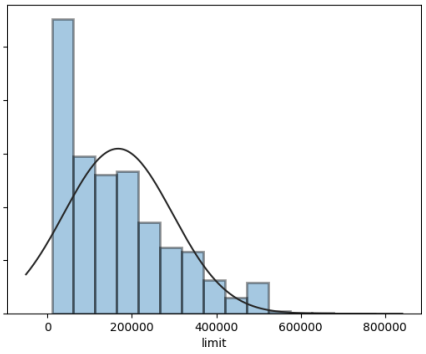

Figura 1.3: Distribuzione attributo `limit`

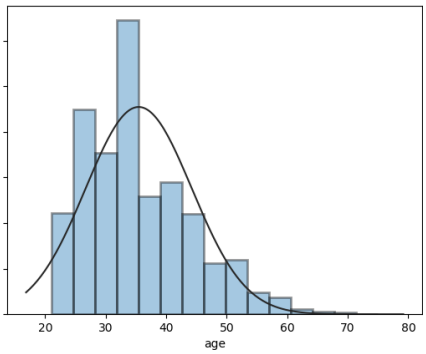

Figura 1.4: Distribuzione attributo `age`

Attributo	Media	Moda	Mediana	STD
limit	167197.0	50000	140000	128975.49
age	35.44	35	35	8.78

Tabella 1.2: valori statistici `limit e age`

Attraverso l'istogramma dell'attributo `limit` (Figura 1.3) è possibile vedere come vi sia una distribuzione inversamente proporzionale alla somma di denaro che la banca mette a disposizione del titolare di carta di credito. Un comportamento dei dati che rispecchia la realtà, evidenziando come le persone più facoltose e con limiti di carta di credito più alte siano anche le meno numerose. L'attributo `age` segue una distribuzione normale, infatti si può notare in Figura 1.4 come il picco dell'istogramma coincida con la curva distribuzione, esattamente sul valore 35.

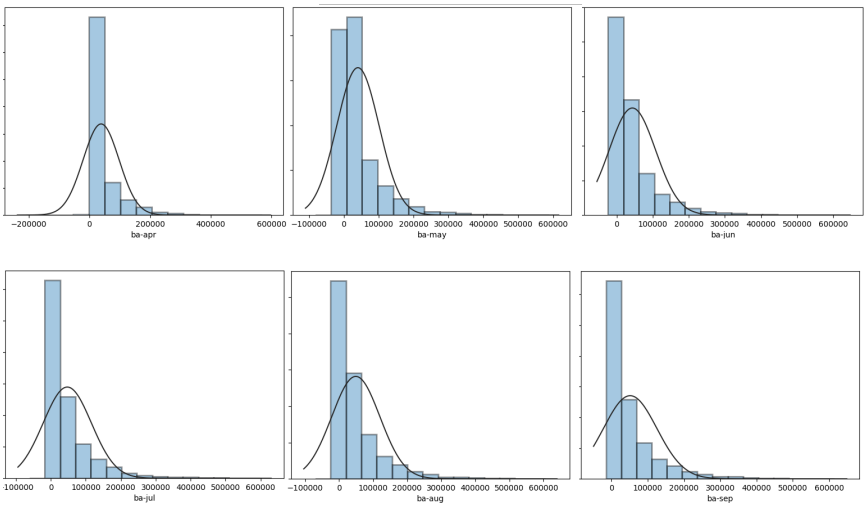

Figura 1.6: Distribuzione attributo `ba-X`

Attributo	Media	Moda	Mediana	STD
ba-apr	38621.58	0	16977	59325.34
ba-may	40182.13	0	18071	60732.33
ba-jun	43306.11	0	19072	64519.91
ba-jul	46957.47	0	19905.5	68948.63
ba-aug	49239.44	0	21202	70777.47
ba-sep	51490.70	0	22246	73740.38

Tabella 1.3: valori statistici `ba-X`

Per i gli attributi relativi al `bill amount` e al `payment amount` è possibile osservare picchi elevati per valori bassi. E' osservabile come il valore più ricorrente in entrambi i casi sia 0, ovvero che la maggior parte dei titolari di carta ogni mese non abbiano da pagare nessuna somma di denaro. Si può come nonostante siano presenti valori medi di `billing amount` piuttosto alti, per ogni mese il payment amount ha un valore medio di circa 5000. Osservando gli istogrammi in figura 1.10 e la Tabella 1.5 relativi ai payment status, si può giustificare la differenza di valori medi degli attributi `billing amount` e `payment amount` da un alto numero di titolari di carta che ricorrono al metodo di pagamento **revolving credit**, rappresentato dal valore 0 degli attributi `payment status`, valore più ricorrente. Inoltre è da sottolineare come gli attributi `payment status` seguano a grandi linee una distribuzione normale, con picco sul valore 0.

Attributi Categorici

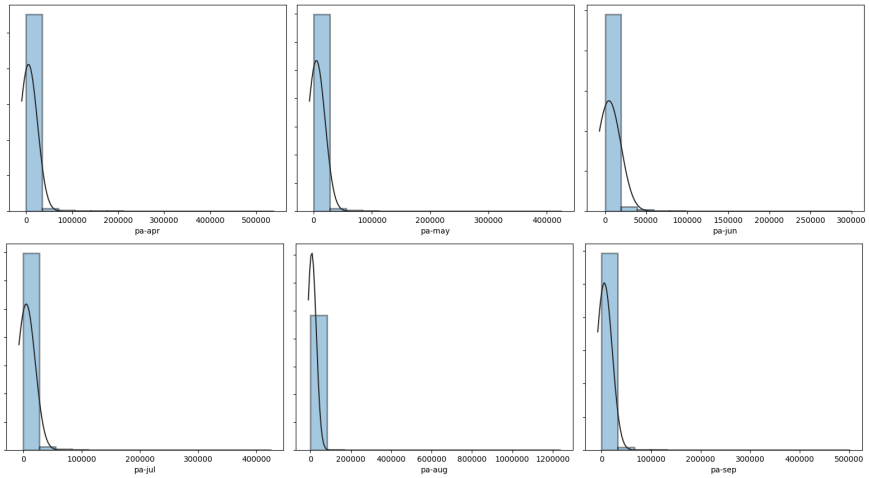

Figura 1.8: Distribuzione attributo `pa-X`

Attributo	Media	Moda	Mediana	STD
pa-apr	5480.15	0	1500	19361.41
pa-may	4734.70	0	1500	14912.37
pa-jun	4719.77	0	1500	14483.41
pa-jul	5131.90	0	1800	15416.40
pa-aug	5973.68	0	2000	22511.75
pa-sep	5651.34	0	2081.5	15835.84

Tabella 1.4: valori statistici `pa-X`

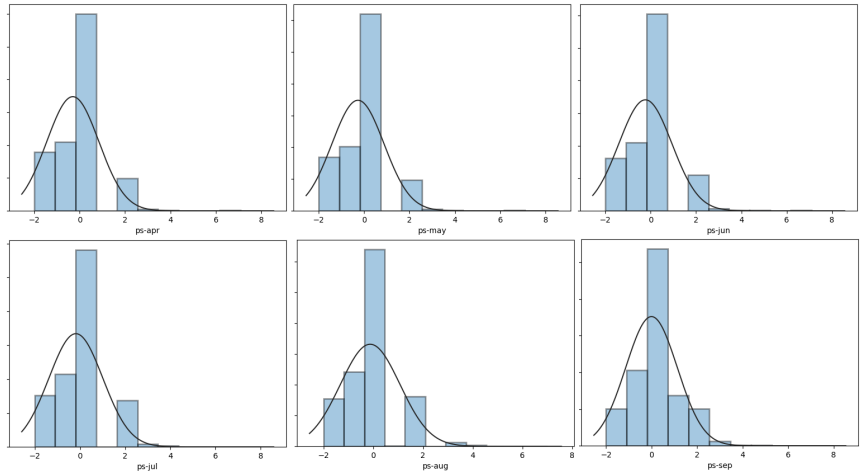

Figura 1.10: Distribuzione attributo `ps-X`

Attributo	Media	Moda	Mediana	STD
ps-apr	-0.30	0	0	1.15
ps-may	-0.26	0	0	1.15
ps-jun	-0.23	0	0	1.17
ps-jul	-0.17	0	0	1.19
ps-aug	-0.13	0	0	1.20
ps-sep	0.00	0	0	1.13

Tabella 1.5: valori statistici `ps-X`

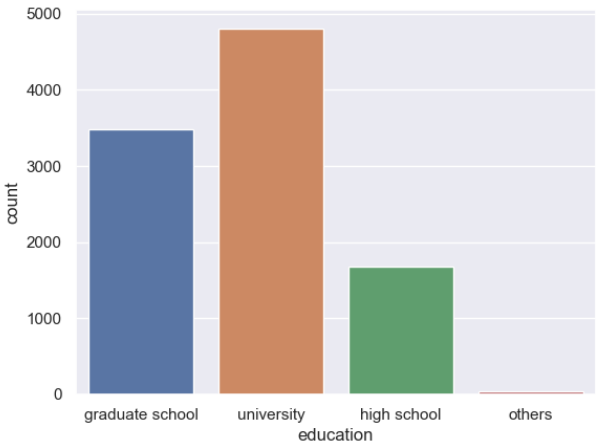

Figura 1.12: Istogramma attributo `education`

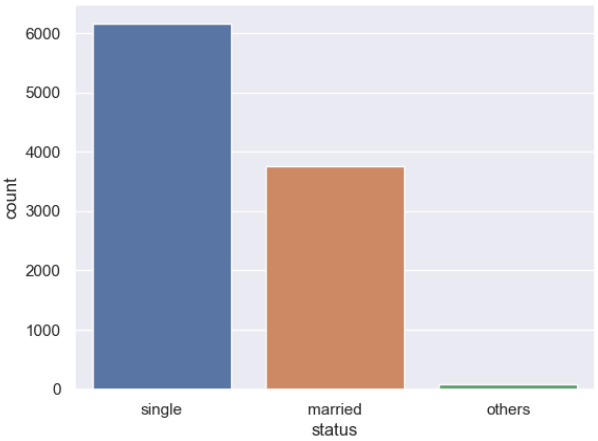

Figura 1.13: Istogramma attributo `status`

Per gli attributi categorici `education` e `status` sono stati generati gli istogrammi riportati in figura 1.12 e 1.13, mentre per l'attributo `sex`, dato che presentava solamente due valori, e l'attributo binario `credit_default` si è preferito l'utilizzo di grafici a torta.

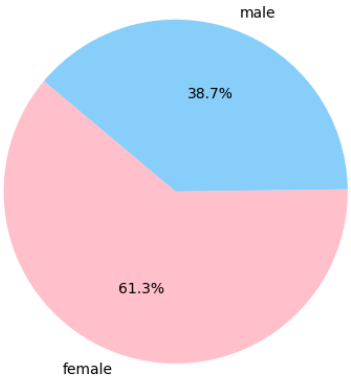

Figura 1.14: Istogramma attributo `sex`

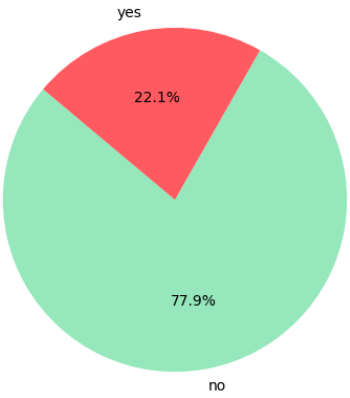

Figura 1.15: Istogramma attributo `credit_default`

1.5 Correlazione tra variabili

L'analisi della correlazione è stata fatta escludendo gli attributi categorici e binari. Con riferimento alla heatmap in figura 1.16, le correlazioni significative sono senza dubbio quelle che occorrono tra i vari `payment status` e `billing amount` di ogni mese, con tendenza decrescente per mesi temporalmente più lontani. I valori alti di correlazione indicano una tendenza nel mantenere il proprio comportamento nel tempo, sia esso positivo (i.e, pagamenti in orario), sia esso negativo nel caso dei `payment status`, o mantenere il proprio trend di spesa nel caso dei `billing amount`. Questo logicamente è del tutto normale in quanto se un individuo comincia a ritardare i pagamenti (equivalente ad alzare il valore dell'attributo), è più facile che i ritardi avvengano anche nei mesi successivi. Mentre per il trend di spesa, dipende dallo stile di vita della persona.

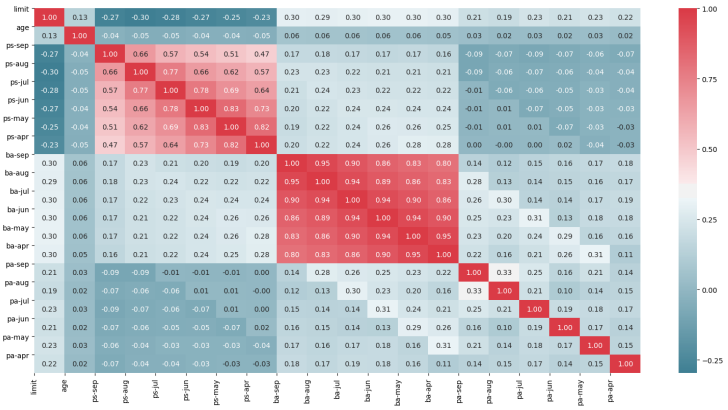

Figura 1.16: Heatmap delle correlazioni tra attributi

Capitolo 2

Clustering

In questo capitolo mostreremo il comportamento degli algoritmi di clustering KMeans, DBSCAN e Hierarchical applicati al nostro insieme di dati.

Il dataset utile per questa fase è stato ottenuto eliminando gli attributi categorici `education`, `sex` e `status` e l'attributo `credit_default` poichè il clustering rientra tra gli addestramenti di tipo non supervisionato.

2.1 KMeans

Il miglior parametro k con cui eseguire KMeans è stato stimato calcolando la *SSE* variando k su un range da 2 a 20, con metrica di distanza euclidea. In figura 2.1 sono riportati i risultati ottenuti. Si nota che il cambio di pendenza della curva si trova quando k vale 7. A partire da ciò abbiamo eseguito l'algoritmo per $k \in [3, 10]$, reiterando per ogni passo 50 volte KMeans per evitare che la scelta casuale dei centroidi influenzasse i risultati e calcolando in questo caso anche l'indice di Silhouette per i cluster trovati. In seguito a queste analisi abbiamo constatato come i cluster ottenuti per $k = 4$ siano i più significativi, la nostra considerazione è stata rafforzata dal fatto che il clustering ottenuto per $k = 4$ presenta il valore dell'indice di Silhouette più alto trovato.

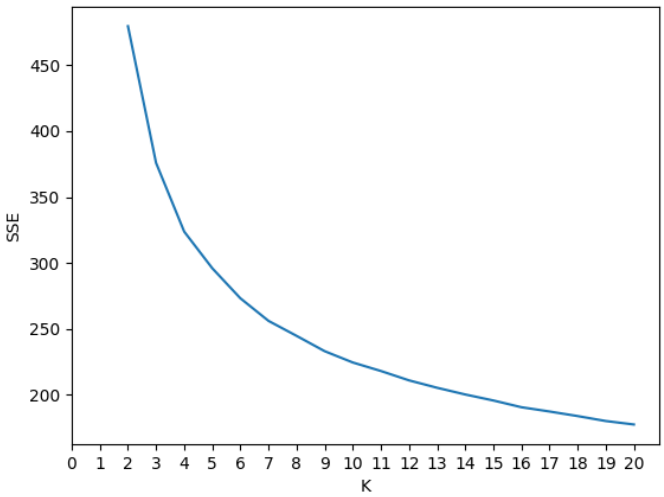

Figura 2.1: SSE

Di seguito analizziamo i cluster trovati, assegnando loro un nome che contraddistingua le caratteristiche di ogni cluster.

Senza rischio Cluster costituito da 3032 persone. Pagano in modo puntuale le loro spese ogni mese senza registrare ritardi e compiono solitamente spese di bassa entità. Non costituiscono alcuna minaccia per la banca.

Piccoli pagatori Gruppo di 4832 persone. Sono soliti fare un uso abbastanza intensivo del revolving credit per pagare le loro spese. Anch'essi registrano spese di bassa entità e non commettono gravi ritardi nel pagamento, per questo motivo rientrano tra clienti credibili per la banca.

Grandi pagatori Cluster di 1083 persone. Come per i *piccoli pagatori*, anch'essi sono soliti usare la modalità di pagamento rateizzata anche se le loro spese sono generalmente molto elevate. Non si registrano comunque grossi ritardi ed é per questo che rientrano comunque tra il gruppo di clienti credibili.

Ritardatari Gruppo formato da 1053 persone. Registrano spese di media entità ma al contrario degli altri tre gruppi si verificano gravi ritardi nei pagamenti. Sono il cluster di persone che, infine, finisce in credit default e non sono clienti credibili per la banca.

In figura 2.2 si sono plottate una selezione delle coordinate dei centroidi, mostrando chiaramente dove i quattro cluster trovati differiscono maggiormente.

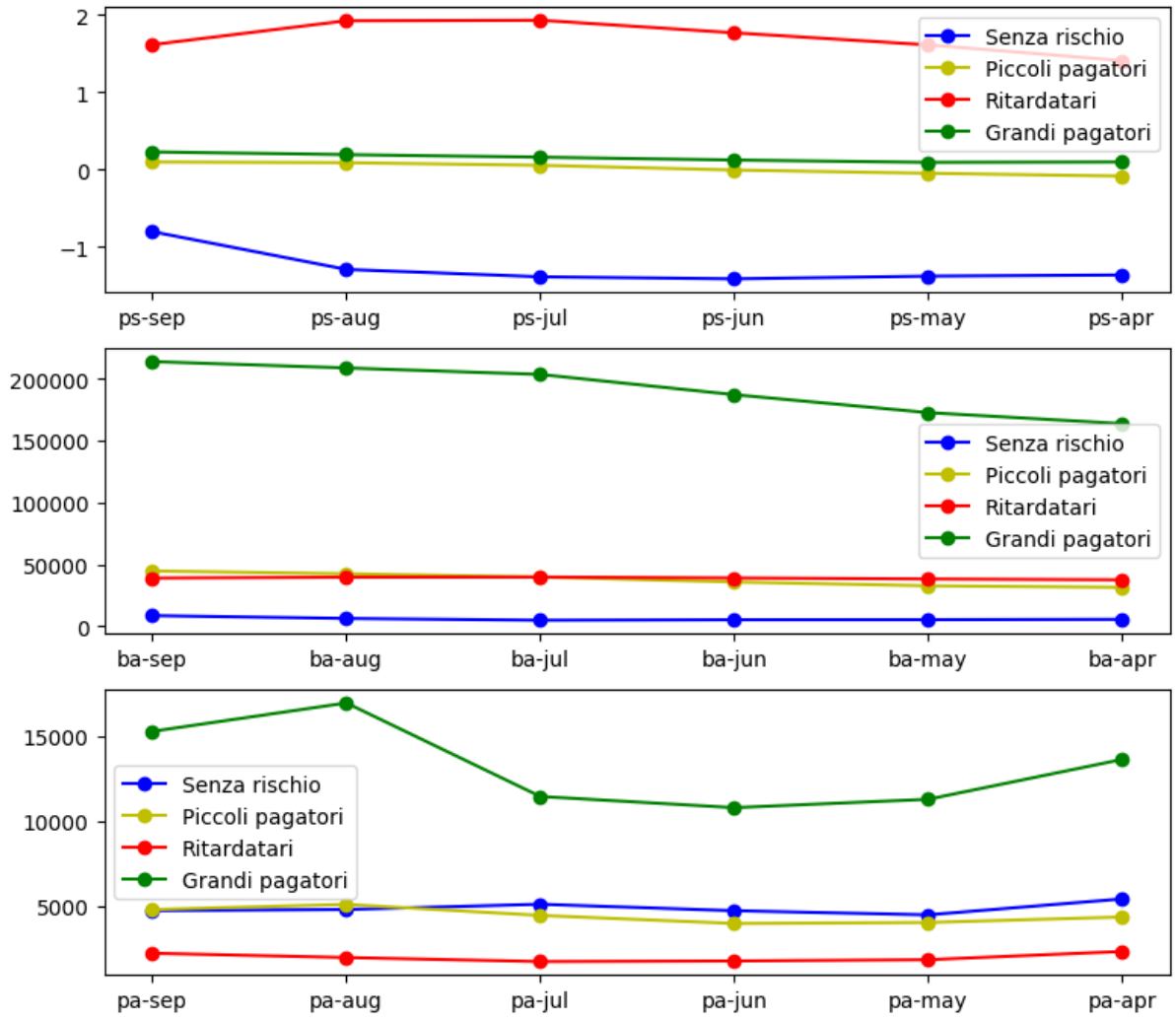

Figura 2.2: Caratteristiche dei cluster

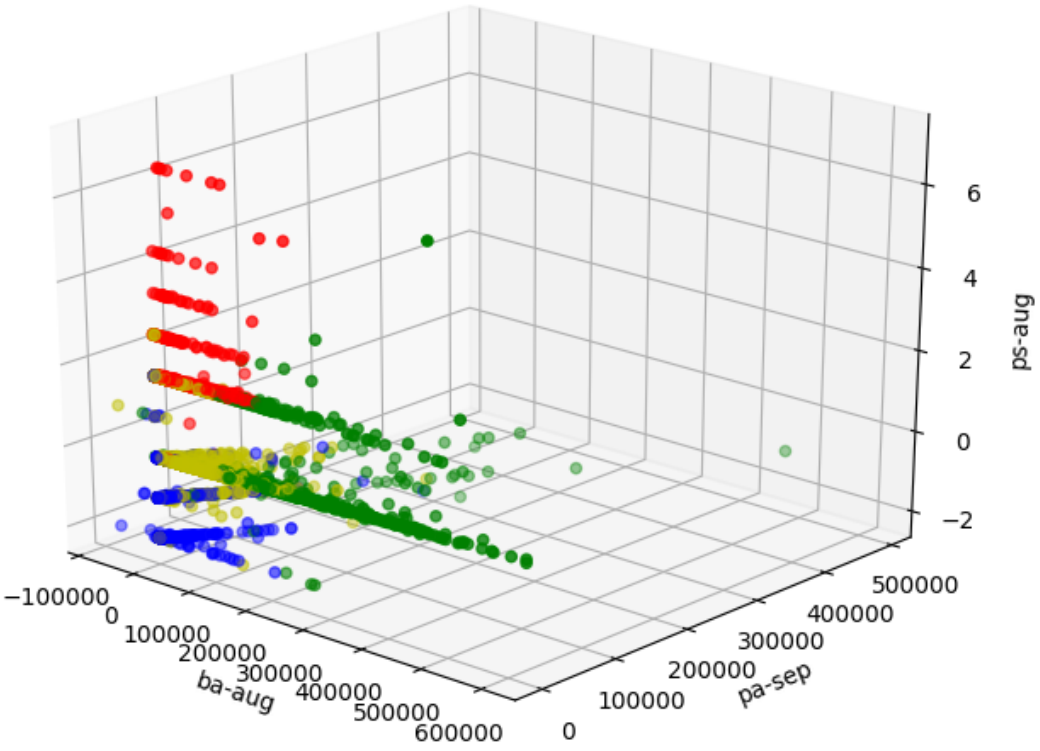

Figura 2.3: Distribuzione dei cluster su un pagamento mensile

Infine si è plottato su un un grafico 3D (figura 2.3 la distribuzione dei cluster su un pagamento mensile (in questo caso si e' preso il pagamento del mese di agosto) ed abbiamo notato che la distribuzione è rispettata per ogni terna di attributi validi costruibili sull'insieme dei mesi disponibili. In particolare il cluster dei *ritardari* si posiziona sempre a valori molto elevati di `payment status`, mentre vale esattamente il contrario per il cluster dei *senza rischio*. Le tre dimensioni scelte per l'esempio sono `billing amount august`, `payment status august` e `payment amount september`.

Infine riportiamo una tabella contenente la media e la deviazione standard di ogni cluster individuato. Per ragioni di spazio per gli attributi `payment status`, `payment amount` e `billing amount` mostriamo i valori degli ultimi due mesi.

Cluster	**ps-sep**	**ps-aug**	**pa-sep**	**pa-aug**
Senza rischio	$-0.8(\pm 1.0)$	$-1.3(\pm 0.7)$	$4.7k(\pm 12.0k)$	$4.8k(\pm 14.3)$
Piccoli pagatori	$0.09(\pm 0.71)$	$0.08(\pm 0.71)$	$4.8k(\pm 10k)$	$5k(\pm 13.3k)$
Grandi pagatori	$0.22(\pm 0.75)$	$0.19(\pm 0.71)$	$15k(\pm 35k)$	$16k(\pm 56k)$
Ritardatari	$1.60(\pm 1.18)$	$1.90(\pm 1.03)$	$2k(\pm 3.4k)$	$1.8k(\pm 3.0k)$
	ba-sep	**ba-aug**	**limit**	**age**
Senza rischio	$8.7k(\pm 21k)$	$6.4k(\pm 16k)$	$215k(\pm 126k)$	$36(\pm 8)$
Piccoli pagatori	$44k(\pm 39k)$	$42k(\pm 35k)$	$130k(\pm 113k)$	$35(\pm 9)$
Grandi pagatori	$213k(\pm 93k)$	$208k(\pm 88k)$	$281k(\pm 115k)$	$37(\pm 8)$
Ritardatari	$38k(\pm 35k)$	$39k(\pm 36k)$	$79k(\pm 68k)$	$34(\pm 8)$
	sex	**status**	**education**	**default**
Senza rischio	F	Single	University	17%
Piccoli pagatori	F	Single	University	18%
Grandi pagatori	F	Single	University	19%
Ritardatari	F	Single	University	61%

Si noti come il gruppo dei ritardatari ha un limite imposto dalla banca molto più basso rispetto agli altri cluster. Segno che la banca ha valutato ottimamente i profili nella scelta di concessione del credito.

2.2 DBSCAN

L'algoritmo DBSCAN è stato eseguito su un dataset modificato rispetto alla esecuzione del KMeans. Questo è stato dovuto per permettere all'algoritmo di funzionare al meglio. Dopo molteplici esperimenti infatti i migliori risultati per DBSCAN sono stati ottenuti su un dataset composto dai soli attributi `payment status` di ciascun mese. Ciò è in linea con la teoria in quanto DBSCAN ha problemi su dati con un numero troppo elevato di dimensioni. Per stimare i parametri ottimali dell'algoritmo si sono utilizzati i grafici del k-dist[2], utilizzando la distanza di Manhatthan. I valori che ci hanno permesso di ottenere un (primo) risultato soddisfacente sono stati $\varepsilon = 0.30$ e $minPoints = 350$. Il primo risultato è stato l'individuazione da parte di DBSCAN di due cluster di dimensioni molto diverse tra loro ma con un significato molto forte.

Senza rischio Composto da 8597 persone, formato da persone che non costituiscono alcun rischio in quanto i loro valori di payment status sono costantemente sotto lo 0.

Ritardatari Composto da sole 454 unità. Ciò che lo contraddistingue è un costante ritardo nei pagamenti dei propri debiti. Queste sono le persone che maggiormente rappresentano un rischio di perdita di credito per la banca.

Data la notevole disparità di dimensioni dei due cluster, abbiamo deciso di eseguire nuovamente DBSCAN sul cluster dei senza rischio al fine di verificare se anch'esso avrebbe trovato la conformazione dei tre cluster individuati da KMeans. Abbiamo rieseguito l'algoritmo con gli stessi parametri e il risultato è stato l'individuazione di tre cluster con una notevole quantità di rumore.

Rifinanziatori Cluster composto da 3271 persone. Sono contraddistinti da un uso intensivo del revolving credit. Le spese sono di entità media. Per questo motivo sospettiamo sia un merge tra i *Grandi pagatori* e *Piccoli pagatori* trovati da KMeans.

Senza rischio Cluster composto da 635 persone. Sono le persone che pagano sempre in orario, ogni mese rispettano la scadenza e ripagano in pieno il loro debito. Hanno caratteristiche molto simili all'omonimo di KMeans.

No consumption Cluster composto da 703 persone. Questa è la novità rispetto a KMeans. Questo cluster incorpora tutte le persone che non fanno uso del loro credito, evidenziato da un valore di *No consumption* molto ripetuto.

In conclusione, DBSCAN in qualche modo valida i risultati ottenuti da KMeans, in quanto due diversi algoritmi, su due selezioni diverse del dataset hanno trovato dei risultati molto simili, fatta eccezione per il cluster dei *No consumption*.

Di seguito riportiamo una tabella riassuntiva delle caratteristiche dei cluster trovati:

Cluster	**ps-sep**	**ps-aug**	**pa-sep**	**pa-aug**
Senza rischio	$-1.0(\pm 0.0)$	$-1.0(\pm 0.0)$	$6.8k(\pm 14.0k)$	$6.4k(\pm 12k)$
No consumption	$-2.0(\pm 0.71)$	$-2.0(\pm 0.71)$	$4.8k(\pm 14k)$	$5k(\pm 14k)$
Rifinanziatori	$0.0(\pm 0.0)$	$0.0(\pm 0.0)$	$6k(\pm 13k)$	$5.9k(\pm 16k)$
Ritardatari	$1.72(\pm 0.68)$	$2.10(\pm 0.33)$	$2.5k(\pm 3.9k)$	$2.5k(\pm 4.3k)$
	ba-sep	**ba-aug**	**limit**	**age**
Senza rischio	$6.3k(\pm 12k)$	$6.4k(\pm 12k)$	$221k(\pm 123k)$	$36(\pm 8)$
No consumption	$7k(\pm 23k)$	$6k(\pm 19k)$	$248k(\pm 122k)$	$36(\pm 7)$
Rifinanziatori	$93k(\pm 88k)$	$89k(\pm 85k)$	$161k(\pm 128k)$	$35(\pm 9)$
Ritardatari	$52k(\pm 56k)$	$53k(\pm 56k)$	$92k(\pm 68k)$	$35(\pm 9)$
	sex	**status**	**education**	**default**
Senza rischio	F	Single	Graduate school	14%
No consumption	F	Single	Graduate School	11%
Rifinanziatori	F	Single	University	9%
Ritardatari	F	Single	University	68%

2.3 Hierarchical Clustering

Per l'algoritmo di Hierarchical Clustering si è utilizzato un dataset privo delle varibili categoriche `education, status, sex` e della variabile `credit_default`, poichè come più volte sottolineato, l'addestramento è di tipo non supervisionato. Inoltre dopo una serie di test si è deciso di eliminare anche la variabile `age`, senza la quale si sono raggiunti buoni risultati. Per parametri del clustering gerarchico si è scelta la **formula euclidea** per la distanza dato che l'utilizzo delle metriche *supremum e manhattan* ha portato alla generazione di clusters di cardinalit troppo diverse. Per lo stesso motivo si è scelto come metodo aggregativo il **Metodo di Ward**, o della devianza minima, attraverso il quale si è raggiunto il miglior risultato in termini di ordine di grandezza delle dimensioni dei cluster. Gli altri metodi utilizzati (single, complete, average) non hanno portato a risultati accettabili.

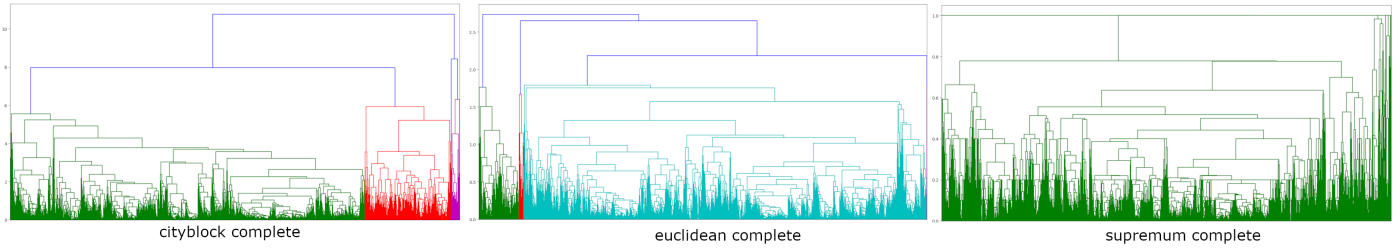

Figura 2.4: Esempio di dendrogrammi ricavati durante il test. I 3 casi in figura non sono stati considerati cluster accettabili.

Utilizzando la metrica euclidea e il metodo di Ward per il merging, si è deciso di settare la soglia a 15.2 ottenendo 4 cluster di dimensioni comparabili. In figura 2.5 sono visualizzabili i clusters ottenuti mentre in figura 2.6 è stato effettuato raggruppamento dei non singleton più profondi e sono stati aggiunti dei label per effettuare il troncamento a 15.2.

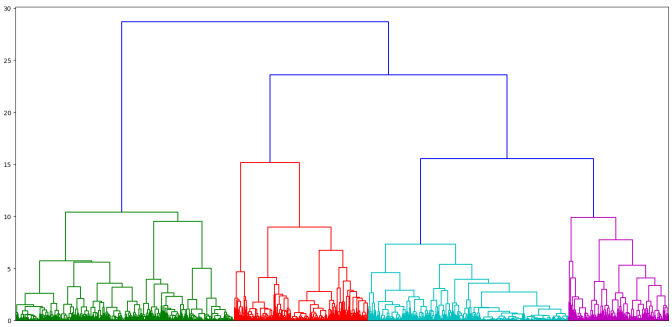

Figura 2.5

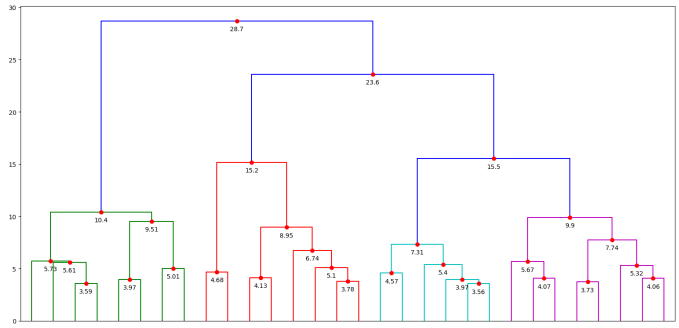

Figura 2.6

$$d(u,v) = \sqrt{\frac{|v|+|s|}{T}d(v,s)^2 + \frac{|v|+|t|}{T}d(v,t)^2 - \frac{|v|}{T}d(s,t)^2}$$

Figura 2.7: Formula metodo di Ward

La Figura 2.7 descrive il metodo di Ward, metodo che cerca di minimizzare la varianza e generare cluster più coesi possibili. Nella formula u è il cluster generato dai cluster s e t, v è un cluster inutilizzato e T è la somma delle cardinalit dei cluster tale che $T = |v| + |s| + |t|$.

I cluster trovati somigliano moltissimo agli stessi cluster trovati di KMeans. Anche in questo caso infatti, oltre ai cluster dei *senza rischio* e dei *ritardatari*, abbiamo una divisione sulla base dell'entità delle spese mensili, formando quindi altri due cluster di *piccoli e grandi pagatori*. Per la descrizione dei cluster rimandiamo alla descrizione dei cluster di KMeans.

Cluster	**ps-sep**	**ps-aug**	**pa-sep**	**pa-aug**
Cluster 1	$-0.01(\pm 0.5)$	$-0.05(\pm 0.46)$	$11.8k(\pm 28.1k)$	$14.2k(\pm 44.6)$
Cluster 2	$-0.68(\pm 1.08)$	$-1.18(\pm 0.72)$	$4.8k(\pm 11.6k)$	$4.8k(\pm 13.2k)$
Cluster 3	$1.7(\pm 0.99)$	$1.8(\pm 0.94)$	$1.8k(\pm 3.3k)$	$2.4k(\pm 3.9k)$
Cluster 4	$-0.10(\pm 0.48)$	$-0.02(\pm 0.56)$	$4.2k(\pm 9.9k)$	$3.4k(\pm 8.0k)$
	ba-sep	**ba-aug**	**limit**	**age**
Cluster 1	$152k(\pm 97k)$	$145k(\pm 94k)$	$246k(\pm 126k)$	$36(\pm 8)$
Cluster 2	$10.5k(\pm 21.8k)$	$9k(\pm 19k)$	$214.3k(\pm 125.3k)$	$36(\pm 8)$
Cluster 3	$48.9k(\pm 49k)$	$48k(\pm 48.1k)$	$85.8k(\pm 71.7k)$	$34(\pm 9)$
Cluster 4	$30.2k(\pm 24.3k)$	$29.4k(\pm 22.7k)$	$103.4k(\pm 85.4k)$	$34(\pm 9)$
	sex	**status**	**education**	**default**
Cluster 1	F	Single	University	12%
Cluster 2	F	Single	Graduate School	15%
Cluster 3	F	Single	University	60%
Cluster 4	F	Single	University	16%

2.4 Analisi del miglior clustering e precisazioni

Prima di passare alle analisi del miglior clustering ottenuto, teniamo a specificare che DBSCAN é stato eseguito su un dataset modificato ad hoc per il suo funzionamento. Questo stesso dataset, composto da soli `payment status` che a

nostro avviso é l'attributo con più carico semantico a disposizione per la nostra analisi, é stato usato successivamente anche con KMeans e Hierarchical. Rieseguendo tutto, essi con un taglio a quattro cluster suddividevano in modo anomalo il cluster dei *ritardatari* in base alla gravità dei ritardi stessi. Con tre cluster entrambi suddividevano il dataset in *senza rischio*, *ritardatari* e persone che utilizzavano intensivamente il revolving credit. Sebbene l'indice di Silhouette sia molto più alto, abbiamo preferito tenere per questi due il clustering sul dataset descritto precedentemente, poichè andava a suddividere in modo più specifico come il revolving credit veniva utilizzato, in modo da avere più informazioni su chi poteva essere a rischio default. In ogni caso, la scelta dipende dalla granularità a cui si vuole vedere il clustering.

Detto ciò analizzando DBSCAN notiamo che sullo stesso dataset esso differisce dal fatto che riesce ad individuare un cluster dei *no consumption*.

In conclusione, non pensiamo ci sia un algoritmo che funziona decisamente meglio degli altri. Tutto dipende da che *cosa* si vuole trovare. Ad esempio, se vogliamo fare un'analisi per trovare il modo di fare spendere di più ai nostri utenti, allora prenderemo in considerazione DBSCAN poichè abbiamo un cluster di persone che non utilizzano la carta ben separato dagli altri, il che rende possibile uno studio più specifico su queste persone. Se invece si vuole semplicemente evidenziare i clienti più a rischio qualsiasi delle tecniche riesce a ben evidenziare il comportamento.

Capitolo 3

Classification

Per la costruzione dei classificatori abbiamo adottato lo stesso dataset usato per DBSCAN, ovvero costituito dai soli attributi relativi ai `payment status`. Questo ci ha permesso di ottenere le migliori performance.

Si è operata una grid search con i seguenti range di valori degli attributi:

- `min_samples_split` $\in [2, 5, 10, 20, 100, 150, 250]$
- `min_samples_leaf` $\in [1, 5, 10, 20, 100]$
- `criterion` $\in [gini, entropy]$
- `max_features` $\in [None, log2, sqrt]$
- `min_impurity_decrease` $\in [1^{-6}, 1^{-7}, 1^{-8}, 0]$
- `max_depth` $\in [None, 2, 4, 7, 10]$

Il risultato è stata l'individuazione di diversi modelli di decision tree equivalenti tra loro. Tutti questi modelli, comunque, condividevano alcuni valori per gli attributi sopracitati. In particolare, il *min_samples_split* era sempre 150, e il *min_samples_leaf* sempre a 20, mentre per gli altri attributi occorrevano una diversa combinazione. Uno dei migliori decision tree trovati presentava un valore di *max_depth* uguale a 7, *min_impurity_decrease* = 7, *gini* come criterio e un numero di split arbitrario. Tutte i classificatori sotto elencati sono stati validati con una operazione di una cross validation con $n\_folds = 10$.

Misure	**Train**	**Test**
Accuracy	0.82	0.81
Precision	0.81	0.80
Recall	0.83	0.82
F1	0.81	0.80

Analizzando questi risultati si nota come la accuracy sia un dato abbastanza positivo, ma, dal punto di vista della banca secondo noi il dato che andava più tenuto sotto controllo era la *recall*. Questo perchè dal punto di vista di chi emette credito, e che si espone dunque al rischio di perderlo, in un dataset dove la classe positiva indica una situazione sfavorevole per l'istituto bancario, sarebbe meglio predirre meno falsi negativi possibili. Bisogna quindi trovare il giusto compromesso tra accuracy e recall, abbiamo dunque esplorato diversi classificatori al fine di trovare il più aderente ai nostri vincoli. Uno dei tentativi è stato fatto con il *Naive-Bayes classifier*, ma i risultati non sono stati abbastanza soddisfacenti.

Misure	**Train**	**Test**
Accuracy	0.80	0.79
Precision	0.80	0.79
Recall	0.80	0.80
F1	0.80	0.80

Come si può notare si è perso sia in accuracy che in recall.

Successivamente, abbiamo utilizzato un RandomForestClassifier, con una grid search molto simile a quella per l'albero e con un numero di estimatori che variava in un range da 5 a 40 il miglior classificatore trovato (avente 10 estimatori) presentava queste performance:

Misure	**Train**	**Test**
Accuracy	0.82	0.81
Precision	0.81	0.80
Recall	0.83	0.82
F1	0.81	0.80

Non si è quindi ottenuto un grande miglioramento rispetto agli ottimi risultati del decision tree. A questo punto abbiamo tentato di settare i pesi per ciascuna classe in modo da migliorare la recall, ma i pesi unitari ed uniformi standard sono quelli che hanno dato i migliori risultati.

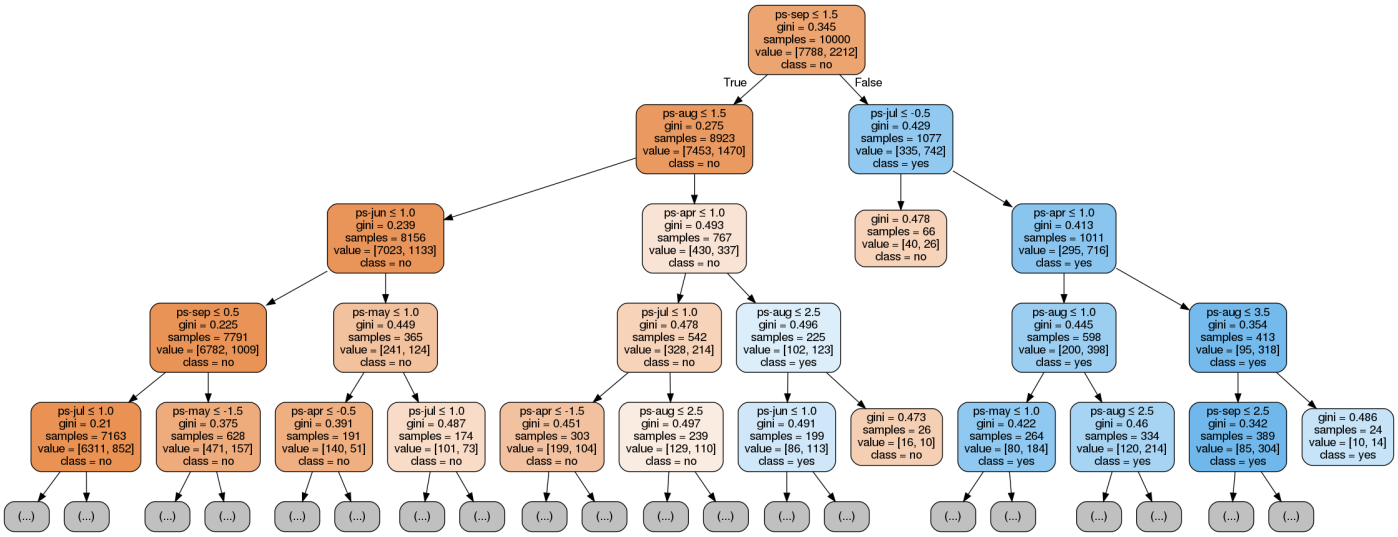

Figura 3.1: Primi quattro livelli dell'albero di decisione trovato

I nostri esperimenti sono continuati utilizzando altri due classificatori, una *neural network* e un *KNN classifier*. Sul primo abbiamo operato una grid search con i seguenti attributi e valori:

- `activation` $\in [logistic, relu]$
- `alpha` $\in [1^{-5}, 1^{-6}, 1^{-7}]$
- `solver` $\in [lbfgs, sgd, adam]$

La rete neurale risultante era caratterizzata da $activation = relu$, $alpha = 1^{-7}$ e $solver = sgd$.

Misure	Train	Test
Accuracy	0.82	0.81
Precision	0.81	0.79
Recall	0.82	0.81
F1	0.80	0.79

Infine, abbiamo tentato con *KNN Classifier* che ha dato i risultati più interessanti. In primo luogo, iterando con diversi valori di *n_neighbors* abbiamo trovato il valore ideale a 150 e con pesi uniformi.

Misure	Train	Test
Accuracy	0.82	0.81
Precision	0.81	0.79
Recall	0.82	0.81
F1	0.80	0.79

In questo caso il classificatore non differisce molto dagli altri, ma nel caso in cui si settino i pesi proporzionali alla distanza (ovvero i vicini più prossimi influenzano maggiormente l'unità presa in esame) si registrano i valori più alti di recall e con minima perdita di accuracy.

Misure	Train	Test
Accuracy	0.84	0.81
Precision	0.83	0.81
Recall	0.84	0.83
F1	0.82	0.80

3.1 Valutazione miglior modello

Come descritto in precedenza, il nostro obiettivo era mettere in risalto che la misura più importante da considerare in questo caso, a nostro avviso, è la recall. Per tanto il miglior classificatore trovato è stato il *KNN classifier* con numero di vicini uguale a 100 e con pesi proporzionali alla distanza dei vicini.

Capitolo 4

Pattern Mining

In questo capitolo analizzeremo la fase di Pattern Mining. Un punto importante di questa fase è che utilizzeremo un dataset che, differentemente dalle altre fasi, sarà privo di outlier. Ciò è stato fatto principalmente per due motivi, il primo è che gli outlier sono per definizione valori anomali, quindi ripetuti poche volte, di conseguenza sarebbero stati inutili nella generazione degli itemset e delle successive regole. Il secondo motivo, più importante, è che data la necessaria discretizzazione di alcuni attributi (i.e, `billing amount`), la presenza di essi produceva degli intervalli che rendevano più difficoltoso il pattern mining. La discretizzazione degli attributi è stata implementata con Sturges. Per quanto riguarda gli attributi `payment status`, è stato invece applicata una discretizzazione con bins asimmetrici. In particolare per i valori -2, -1, e 0 si sono tenuti i singoli valori, mentre si è creato un intervallo [1,9] per i restanti per mettere particolare enfasi sulla presenza o meno di un ritardo nel pagamento, mettendo in secondo piano la gravità del ritardo stesso.

4.1 Estrazione degli item più significativi

La prima estrazione di itemset è stata effettuata con un supporto del 30%. Si sono estratti circa 2200 item di cui tutti completi e circa 490 massimali. La prima cosa che si è notato è che tutte le regole erano accorpabili e riconducibili a pochi insiemi che descriviamo:

Gruppo 1 Il primo gruppo evidenzia una tendenza nel dataset a mantenere più o meno costante il trend di spesa durante tutti i mesi. Esempio:

(('ba-jun [-15910.0, 6717.7)', 'ba-jul [-15910.0, 7900.0)', 'ba-may [-7529.9, 15421.2)', 'ba-apr [-7616.5, 17105.2)'))

Gruppo 2 Questo gruppo di regole evidenzia le persone che utilizzano il revolving credit e che infine non vanno in credit default. Ha una somiglianza molto forte con i cluster trovati.

(('ps-sep 0', 'ps-jul 0', 'ps-aug 0', 'ps-jun 0', 'ps-may 0', 'default 0'))

Inoltre si è provato ad abbassare il support al 10%, sono risultati circa 12000 itemset, di cui 11000 completi e 9000 massimali. Si sono trovati anche qui gruppi interessanti:

Gruppo 3 Percentuale di persone che smette di usare la carta ma continua a pagare i propri debiti accumulati in precedenza, riuscendo a non finire in default.

(('ps-jun -2', 'ps-jul -2','default 0', 'pa-jul [0.0, 4636.4)', 'pa-jun [0.0, 4371.8)'))

Gruppo 4 Percentuale di persone che hanno un limite molto alto e non vanno in credit default. Questo può essere un segnale di come la valutazione del merito creditizio sia buona nella banca.

(('limit [157272.7, 206363.6)', 'default 0'))

Gruppo 5 Percentuale di persone che pagano puntualmente e come da previsioni non finiscono in credit default.

(('ps-sep [-1, 0)', 'ps-jul [-1, 0)', 'ps-aug [-1, 0)', 'default 0'))

Gruppo 6 Persone che non utilizzano il credito.

(('ps-aug -2', 'ps-jul -2', 'ps-jun -2', 'ps-may -2'))

Infine, data la percentuale molto sbilanciata di distribuzione di casi di default nel dataset si è deciso di eseguire *apriori* su un dataset composto da soli casi di default. Con un supporto del 30% si sono ottenuti 25 itemset, di cui 16 completi e 6 massimali.

Gruppo 7 Persone che utilizzano il revolving credit ma non riescono a pagare, finendo in default.

(('ps-aug 0', 'ps-sep 0', 'ps-jun 0' 'default 1'))

Mentre con una percentuale del 10% si ottengono 300 item, di cui circa 200 completi e solo 15 massimali, tutti riconducibili ad un unico gruppo.

Gruppo 8 Persone che pagano sempre in ritardo ed ovviamente finiscono col finire in credit default.

(('ps-apr [1, 10)', 'ps-may [1, 10)', 'ps-jun [1, 10)', 'ps-jul [1, 10)', 'ps-aug [1, 10)', 'ps-sep [1, 10)', 'default 1'), 312)

4.2 Estrazione delle association rules più significative

La prima estrazione delle association rules è stato effettuata con una percentuale di supporto del 30% e di confindenza del 90%. Questa estrazione ha portato soli due gruppi di regole:

Gruppo 1

$$baX \in [a, b] \rightarrow ba(X \in \{apr, may, jun, jul, aug, sep\}) \in [c * a, c * b]$$

Essa afferma che se il billing amount di un mese risulta entro un certo intervallo di spesa, allora il billing amount di tutti gli altri mesi risultera più o meno entro lo stesso intervallo o comunque non troppo distante. Questa regola afferma chiaramente che c'è una tendenza a mantenere costante il proprio stile di vita. Questa regola ha un lift di circa 2.

Gruppo 2

$$psX = 0 \rightarrow ps(X \pm 1) = 0$$

Ovvero che l'uso del revolving credit per un mese implica il suo uso anche nei mesi immediatamente precendenti e successivi. Questo indica che le difficoltà di un mese si ripercuotono anche successivamente (o sono stati indotte precedentemente). Il lift di questa regola è la più bassa registrata, circa 1.60.

Abbassando il supporto fino al 5% e con una confidenza del 60% si trova un insieme di regole (**Gruppo 3**) che descrive un tipo di comportamento prudente tenuto da una parte del dataset. Dopo una serie di mesi di spesa, non si utilizza più il credito per poter pagare con meno rischi, il debito accumulato precedentemente, rientrano dal rischio e non registrando credit default. Un'istanza di questa regola è riportata successivamente assieme alle misure di supporto, confidenza e lift.

('ps-sep -2', ('ps-aug -2', 'ba-jun [-15910.0, 6717.7)', 'ba-jul [-15910.0, 7900.0)', 'ba-may [-7529.9, 15421.2)', 'ba-aug [-9557.5, 20552.2)', 'default 0'), count=631, supp=0.07, conf=0.70, lift=5.43)

Data la necessità di costruire un classificatore rule based, abbiamo deciso di estrarre le regole su un dataset solamente composto dagli attributi `payment status`, per estrarre con più facilità le regole che legavano essi al caso di default. Tenendo il supporto al 5% e una confidenza del 70% si sono estratte le seguenti regole. Le prime regole estratte descrivevano dei comportamenti molto simili alla regola sui `billing amount`, ovvero:

$$psX \geq 1 \rightarrow psY, psZ \geq 1 \text{ con } lift \geq 5.9$$
$$psX = -1 \rightarrow psY, psZ = -1 \text{ con } lift \geq 4.5$$

Ovvero si registra anche sugli status una tendenza nel mantenere il proprio comportamento. Si sono inoltre estratte delle regole utili per predirre la classe *no default* con un lift abbastanza alto. Tutte le regole estratte si possono riassumere in due schemi di regole.

$$psX, psY = -1 \rightarrow default = 0 \text{ con } lift \geq 4.2$$
$$psX, psY, psZ = -1 \rightarrow default = 0 \text{ con } lift \geq 3.9$$

Ovvero il pagamento in orario per un numero di mesi maggiore o uguale a due implica un caso di non default nel dataset. Si noti che psX non significa che la regola valga per qualsiasi valore di X ma solo per un suo sottoinsieme. In altre parole le regole non valgono per qualsiasi tuple di mesi in quanto esistono dei mesi per i quali sembra che il pagamento in orario e in pieno sia considerato più che per altri. Si è inoltre ripetuta l'estrazione delle regole sul solo dataset di credit default e si è verificata l'esistenza di regole simmetriche seppur con valore di lift più basso.

$$psX, psY, psZ \geq 1 \rightarrow default = 1 \text{ con } lift \geq 2.7$$
$$psX, psY, psZ, psH \geq 1 \rightarrow default = 1 \text{ con } lift \geq 2.4$$

4.3 Rule based classifier

Avendo solo due valori nella classe target è possibile utilizzare solo uno dei due set di regole. La scelta è ricaduta sul set di regole per la classe *no default* in quanto i valori di lift sono più alti. Abbiamo quindi costruito un classificatore con le istanze trovate dello schema di regole descritto precedentemente. I clienti aderenti alle regole sono classificati come non default, il restante come default. Di seguito riportiamo i risultati:

Misure	**Performance**
Accuracy	0.79
Precision	0.68
Recall	0.16
F1	0.26

Notiamo come i risultanti non siano soddisfacenti, specialmente per quanto riguarda la recall che, per quanto spiegato nel capitolo 3, è la misura che consideriamo più importante di tutte. Anche per quanto riguarda la accuracy comunque, si tenga conto che il classificatore banale che risponde sempre *no default* avrebbe una accuracy dell'82%, per tanto non può essere tenuto come dato positivo. Ai fini di un'analisi più accurata abbiamo anche rieseguito l'estrazione delle association rules suddividendo il dataset in training e test set. L'estrazione delle regole (eseguita solo sul training set) ha portato a risultati molto simili ai precedenti, per questo abbiamo riscontrato risultati molto simili.

Misure	**Train**	**Test**
Accuracy	0.79	0.78
Precision	0.64	0.63
Recall	0.19	0.19
F1	0.34	0.33

Per quanto riguarda i classificatori per i missing values, abbiamo implementato un classificatore per l'attributo `sex`. Abbiamo rieseguto l'estrazione delle association rules sul dataset. Le regole con valore di lift più alte sono stati ottenute con un supporto del 5% e una confidenza del 50%. Non avendo ottenuto regole in cui le istanze di `sex` occorrevano da sole come secondo termine dell'implicazione, abbiamo analizzato solo le precondizioni delle regole per le quali una delle due istanze di `sex` rientrava (insieme ad altri) a destra dell'implicazione. Le precondizioni ottenute sono state:

- `ps-sep` $\in [0]$
- `ps-aug` $\in [-2, -1]$
- `ps-jul` $\in [-2, -1]$
- `ba-jun` $\in [-15k, 6.7k]$
- `ba-sep` $\in [-9.8k, 15.8k]$
- `ps-jun` $\in [-2]$
- `ba-may` $\in [15k, 38k]$
- `ba-jul` $\in [-15k, 7.9k]$
- `ba-may` $\in [15k, 38k]$
- `ps-apr` $\in [-2, -1]$
- `ps-may` $\in [-1]$
- `ba-apr` $\in [17k, 41k]$

Per questo classificatore è stata valutata solo l'accuracy che si è dimostrata abbastanza scarsa (intorno al 58%). L'unico classificatore per i missing values sensato costruibile su questo dataset è questo poichè è un attributo categorico che presenta solo due valori. Per altri attributi che presentano missing values, `status`, `education`, `age`, per la loro conformazione multi classe, per alti volare di supporto e confidenza non sono state trovare sufficienti regole, mentre per abbassando i valori dei parametri si trovavano molte regole ambigue. Per tanto costruire un classificatore su queste basi non è stato possibile.