Data Science and Business Informatics

A.A. 2017/2018

Data Mining II

A cura di:
Affolter Riccardo (555563)
Fanton Marco (555200)
Vasile Ludovica (495765)

Indice

Time series

1.1 - IBM stocks

Il dataset utilizzato per le analisi, "*IBM stocks*", rappresenta i valori azionari dell'azienda statunitense, relativi agli ultimi 57 anni (dal 1962 al 2018).

Figura 1.1: Andamento di IBM Stocks

Ciascun record contiene i dati relativi ai prezzi storici espressi in dollari statunitensi ed è caratterizzato da 6 attributi: *open, high, low, close, adg* e *volume*.

1.2 - Preparazione dei dati

Prima di procedere con le analisi, dopo che il dataset è stato suddiviso per anni, sono state apportate alcune modifiche:

- ➢ A causa del numero eccessivo di giorni mancanti (294), quindi di dati reali, l'anno 2018 non è stato preso in considerazione.
- ➢ Al fine di ottenere prestazioni migliori nel clustering e nello studio della similarità, i valori sono stati normalizzati mediante *amplitude scaling* e *smoothing*.
- ➢ Per ottenere time series della stessa lunghezza, è stato effettuato il padding sostituendo al valore -3 iniziale, la media.

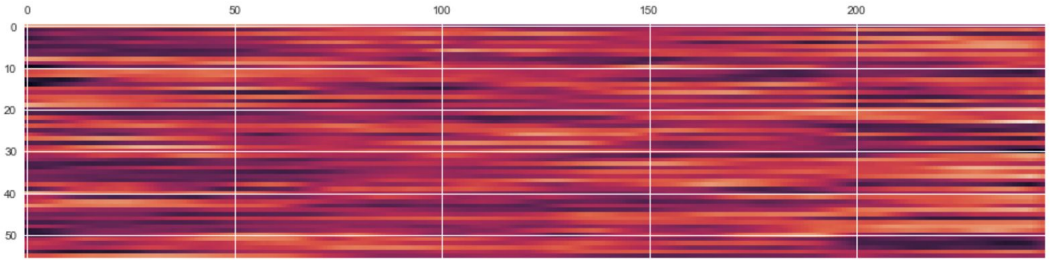


Figura 1.2: Heatmap delle time series annuali

Al termine delle trasformazioni, le 56 time series presentano tutte stessa lunghezza pari a 246.

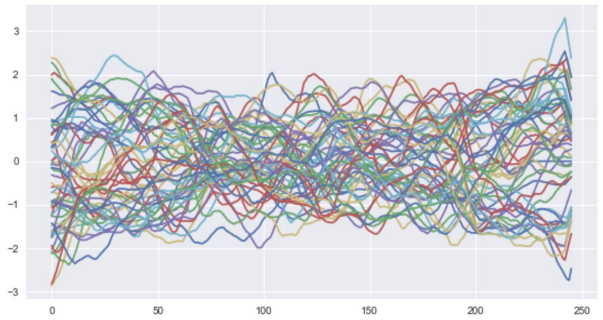


Figura 1.3: Plot delle time series risultanti dopo le operazioni di normalizzazione

Dall'analisi dell'autocorrelazione, inoltre, è possibile stimare che il dataset non presenta valori generati casualmente: dal grafico sottostante è possibile notare un gran numero di valori positivi di autocorrelazione.

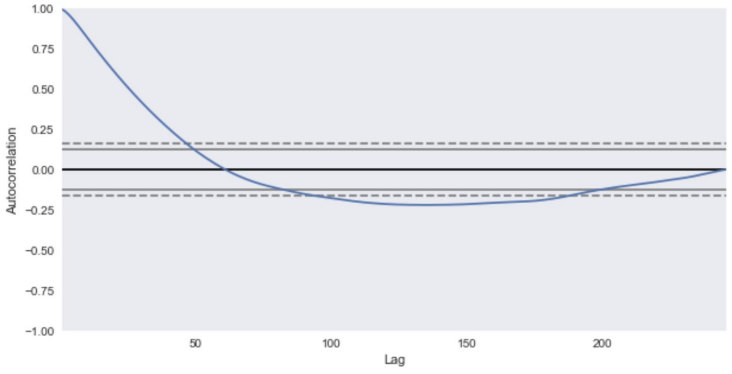


Figura 1.4: Autocorrelation plot

1.3 - Clustering e analisi della similarità

Per lo studio della similarità si è scelto di eseguire il clustering mediante *DBscan*, algoritmo di clustering basato sulla densità, confrontando i risultati ottenuti utilizzando come funzioni di distanza *Dynamic Time Warping* e distanza euclidea.

1.3.1 - DTW

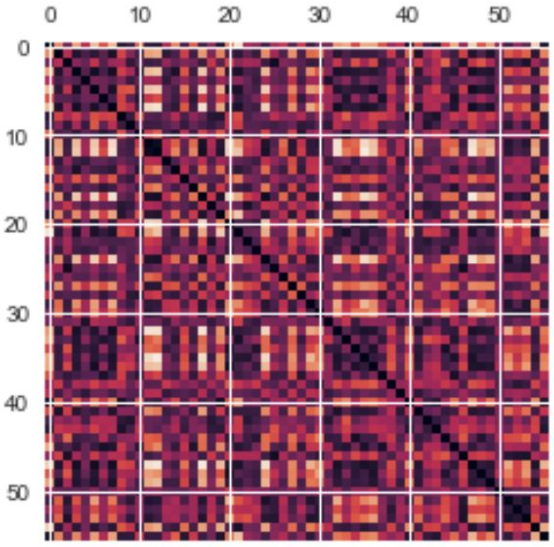


Figura 1.5: Matrice delle distanze DTW

Per la scelta del valore ottimale del parametro ε, necessario per l'implementazione dell'algoritmo di clustering, sono state effettuate alcune prove; il grafico sottostante suggerisce che il valore di ε che massimizza il numero di clustering è compreso nell'intervallo [30,40].

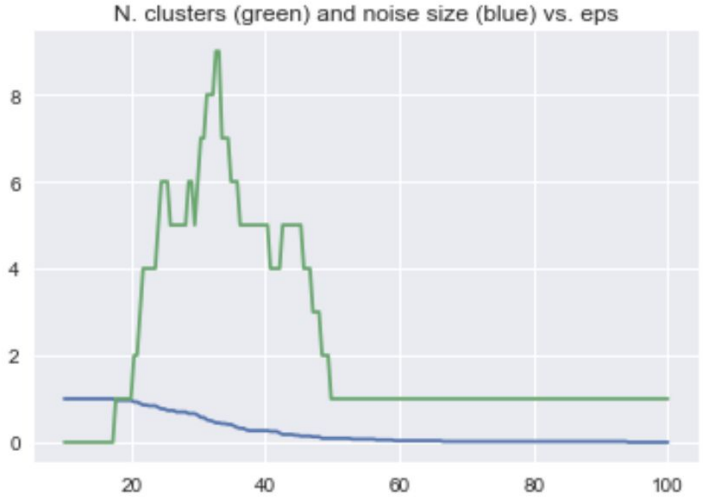


Figura 1.6: Grafico per la scelta del valore ottimale di ε

Impostando ε = 32.5 e minPts = 2, sono stati ottenuti 8 clusters:

- `Cluster 0 = {1962, 1970, 1981, 2002}`
- `Cluster 1 = {1964, 1987, 2008}`
- `Cluster 2 = {1965, 1967, 1969, 1982, 1994, 1997, 1998, 2009, 2010}`

- Cluster 3 = {1968, 1999, 2007}
- Cluster 4 = {1973, 1974, 1979, 1986}
- Cluster 5 = {1983, 1995}
- Cluster 6 = {1996, 2006}
- Cluster 7 = {2013, 2015}

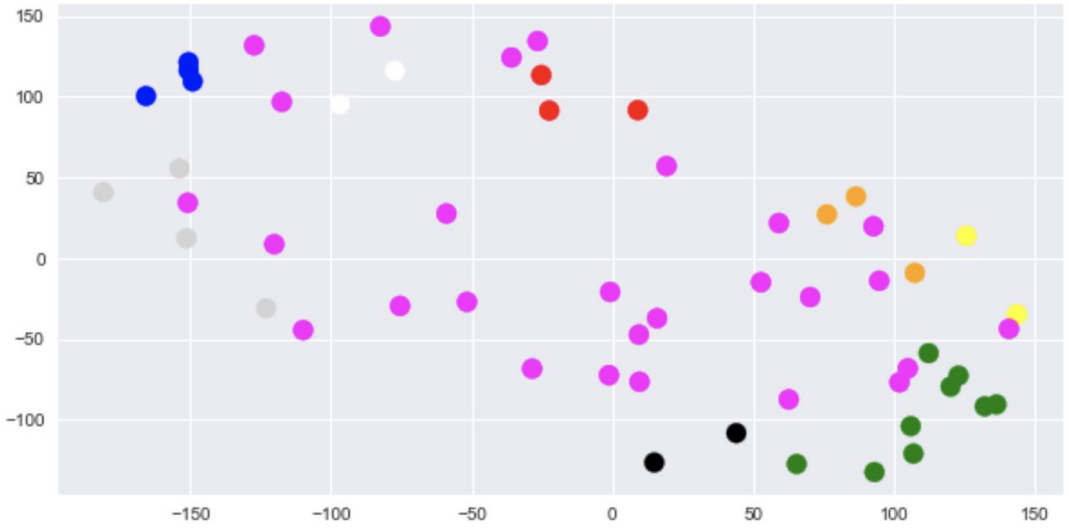


Figura 1.7: Scatter plot dei clusters individuati

Come emerge dallo scatter plot in Figura 1.7, tutti gli 8 clusters risultano essere abbastanza densi e ben separati; gli anni che non rientrano nei suddetti clusters sono stati classificati come *noise points*.

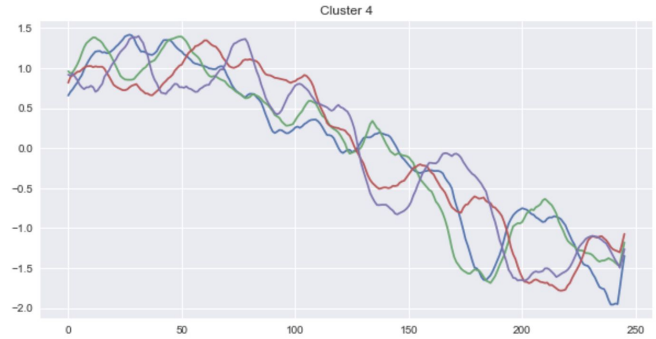
Dai grafici sottostanti è possibile notare, che le time series appartenenti ai singoli clusters presentano, in linea di massima, tutte lo stesso andamento; in alcuni casi si registra la presenza di picchi, anche piuttosto pronunciati.

Gli anni 1962, 1970, 1981, 2002, sono caratterizzati da un andamento abbastanza alto relativo ai primi mesi, che tende progressivamente a decrescere per poi risollevarsi leggermente in corrispondenza degli ultimi giorni.

Dopo un inizio non particolarmente eccellente, gli anni 1964, 1987, 2008 si caratterizzano per una crescita iniziale alla quale segue un rapido calo dell'andamento.

A differenza di tutti gli altri, gli anni appartenenti al Cluster 2, presentano un andamento pressoché lineare che tende progressivamente a crescere.

L'andamento degli anni 1964, 1987, 2008 potrebbe quasi ricordare quello degli anni 1968, 1999, 2007 appartenenti al Cluster 3. Questi ultimi si distinguono per la presenza dei numerosi picchi.



Per gli anni 1973, 1974, 1979, 1986, si registrano valori iniziali piuttosto elevati, ai quali segue un progressivo calo dell'andamento.

Una situazione pressoché opposta riguarda gli anni 1983 e 1995: i valori piuttosto bassi in corrispondenza dei primi mesi, tendono gradualmente a crescere. Anche in questo caso si nota la presenza di alcuni picchi.

Gli anni 1996 e 2006 dopo un inizio particolarmente negativo per uno e poco discreto per l'altro, presentano un andamento che prima decresce e poi cresce in corrispondenza degli ultimi giorni

Per gli anni 2013 e 2015 l'andamento è piuttosto simile a quello degli anni appartenenti al Cluster 4. Le differenze più rilevanti stanno nei valori iniziali poco più bassi e nella presenza di numerosi picchi.

1.3.2 - Distanza euclidea

In un secondo momento si è deciso di eseguire l'algoritmo di clustering, utilizzando come metrica la distanza euclidea (in fase di preparazione era stato generato un dataset contenente time series di pari lunghezza, utile ai fini del calcolo della matrice delle distanze euclidee).
Per la scelta del valore ottimale del parametro ε, è stata utilizzata la funzione in R *kNNdist*, con minPts = 2. Tale funzione calcola le distanze da ciascun punto al proprio k-esimo vicino, le ordina in maniera crescente e le plotta su un grafico (Figura 1.8). Il punto di flesso indica il valore ottimale di ε.

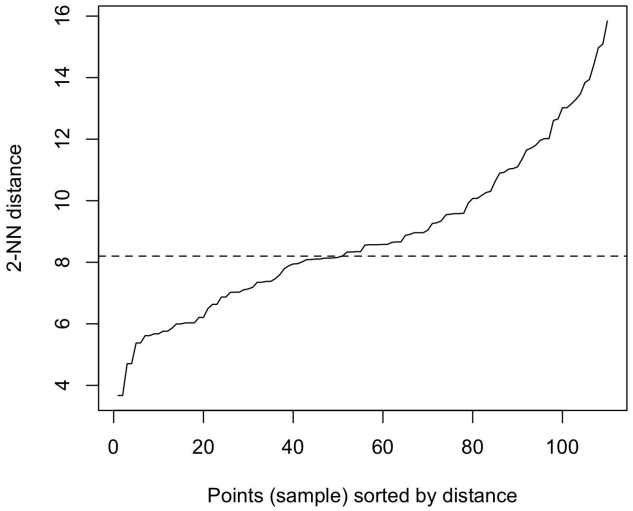
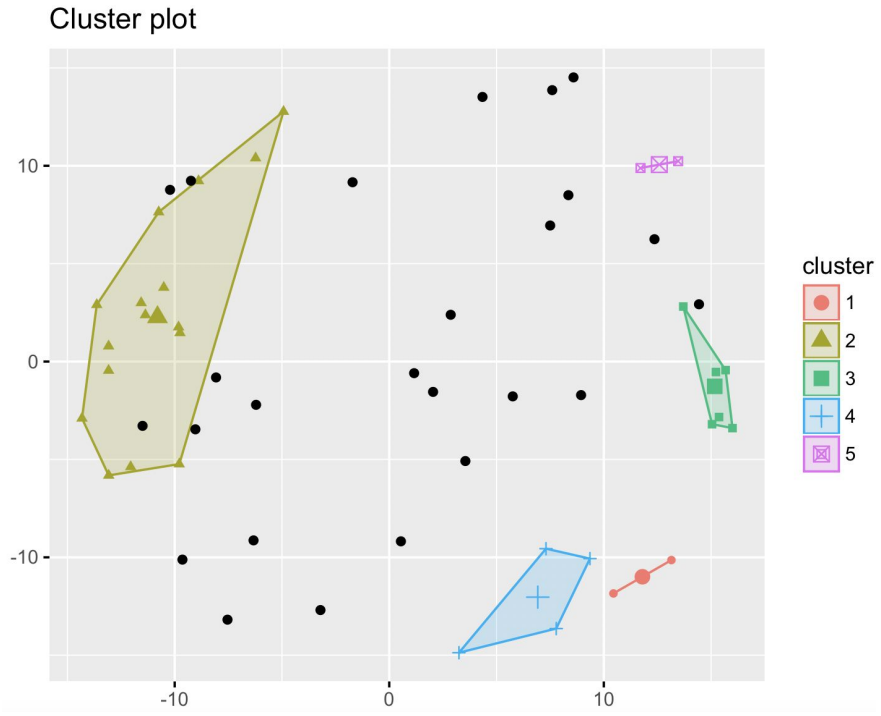


Figura 1.8: Line plot delle distanze

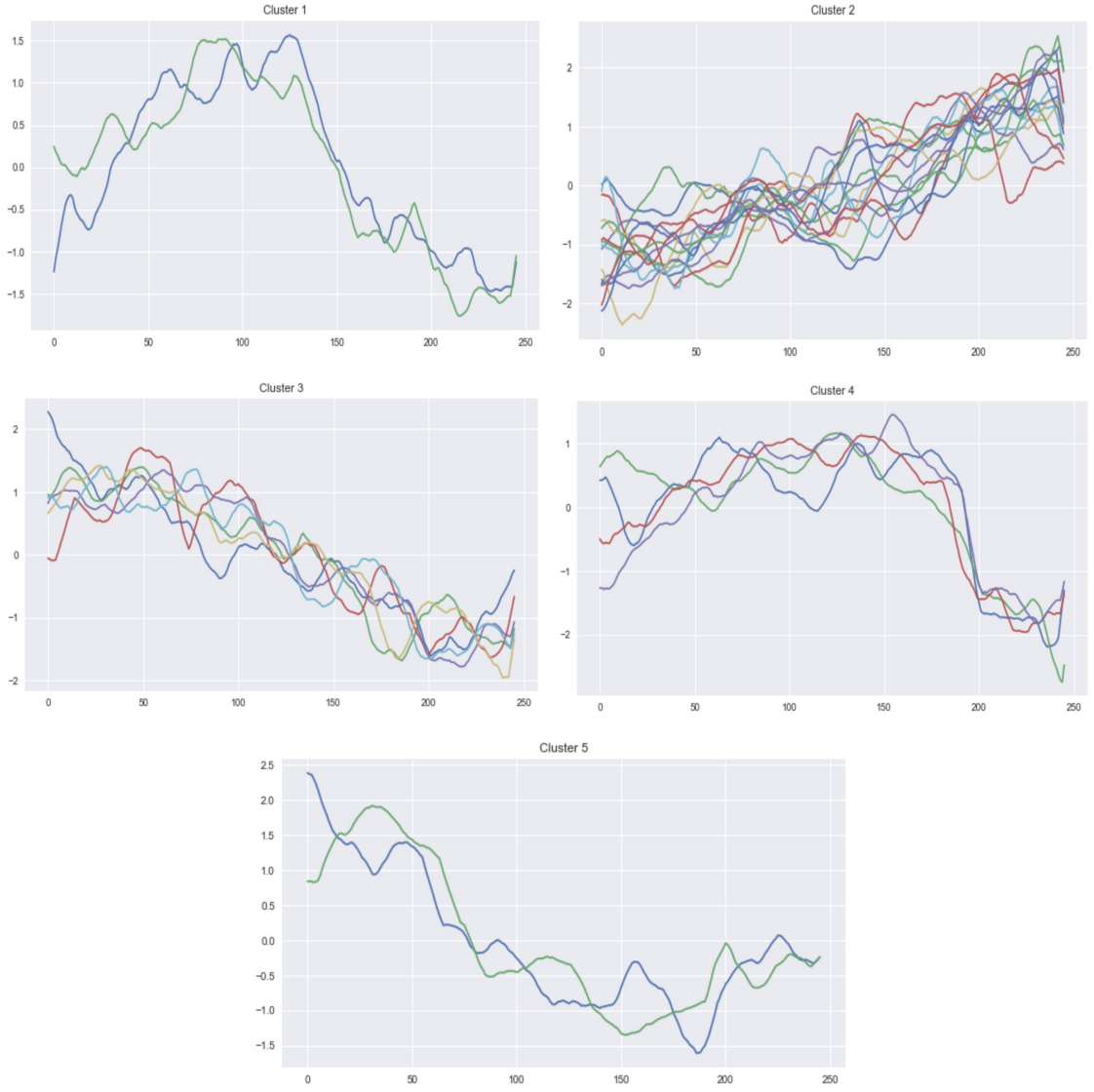
Impostando ε = 8.2 e minPts = 2 sono stati ottenuti 5 clusters:

- Cluster 1 = {1964, 2015}
- Cluster 2 = {1965, 1967, 1969, 1982, 1983, 1994, 1996, 1997, 1998, 2003, 2006, 2007, 2009, 2010, 2011, 2016}
- Cluster 3 = {1973, 1974, 1979, 1982, 1986, 2013}
- Cluster 4 = {1987, 1992, 2008, 2014}
- Cluster 5 = {2002, 2017}

Il grafico seguente riporta la rappresentazione bidimensionale dei clusters ottenuti.



Alcuni dei clusters individuati, tuttavia, non sembrano presentare un elevato grado di coesione interna. Se ne riporta una rappresentazione più dettagliata, nei grafici seguenti.



Il cluster 2 e il cluster 3, sembrano richiamare la struttura “generale” rispettivamente dei clusters 2 e 4 ottenuti con DTW.
I clusters 1 e 4 presentano entrambi un andamento che prima cresce e poi decresce (più in fretta nel cluster 4), richiamando la struttura del cluster 1 ottenuto con DTW.
All’ultimo cluster (cluster 5) appartengono due time series (2002 e 2017) che non sembrano poi così “vicine”: utilizzando l’approccio con DTW, il 2017 era stato classificato come noise point e il 2002 raggruppato con 1962, 1970 1981.

Nonostante la complessità più elevata per il calcolo delle distanze, l’approccio che utilizza DTW sembra definire performances migliori. Attraverso la “distorsione”

dell’asse temporale, tale metodo permette di ottenere infatti, allineamenti migliori; nel caso dei clusters ottenuti con DTW, in più occasioni, è possibile verificare che i picchi di due o più serie appartenenti allo stesso cluster tendono a “coincidere”, più di quanto succeda nel caso dei clusters ottenuti con la distanza euclidea.

Sequential patterns

2.1 - Dataset e preparazione dei dati

Per il secondo task, il dataset utilizzato è lo stesso del punto precedente. L'obiettivo adesso consiste nell'estrazione dei *sequential patterns*, cioè delle sottosequenze contigue, frequenti e di lunghezza pari ad almeno 4 giorni. Per tale operazione, dopo che i dati sono stati opportunamente preprocessati, è stato applicato l'algoritmo *SPAM* (*Sequential PAttern Mining*) di *SPMF*, libreria open source per il Data Mining, scritta in Java.

Nella fase di preprocessing, i dati sono stati prima suddivisi in time series mensili; le 676 serie ottenute sono state quindi normalizzate e "tagliate" tutte alla lunghezza di 23 giorni, eliminando gli ultimi 8 giorni (in media) per ciascun mese, perché contenenti valori tutti pari a -3.
Preliminare per l'applicazione dell'algoritmo SPAM è inoltre la discretizzazione della time series, per la quale si è scelto di implementare l'algoritmo *SAX* (*Symbolic Aggregate ApproXimation*). SAX riceve in input una time series di lunghezza *n* e la trasforma restituendo come output una stringa di lunghezza arbitraria *m*, dove m«n. L'algoritmo consiste di due passi:

1. *PAA* (*Piecewise Aggregate Approximation*): al fine di ridurre la dimensionalità, la time series viene suddivisa in *t* segmenti e per ciascuno è calcolata la media. La sequenza così assemblata a partire dai valori della media, costituisce la rappresentazione PAA della serie.
2. Conversione di PAA in sequenza di simboli: l'idea alla base è che i valori seguono una distribuzione normale; i simboli vengono scelti, in modo da rappresentare i diversi intervalli dei valori.

Per la seguente analisi, è stato scelto un alfabeto di 6 lettere (a, b, c, d, e, f); si è scelto quindi di suddividere i dati in 6 intervalli, ciascuno dei quali identificato da una lettera dell'alfabeto.
Per ottenere dei risultati più accurati, i valori pari a -3 contenuti ancora in alcune delle serie dopo l'eliminazione degli ultimi 8 giorni, al momento della realizzazione degli intervalli, non sono stati presi in considerazione.

2.2 - SPAM e analisi dei patterns

Oltre un file di input opportunamente preprocessato, l'algoritmo SPAM richiede il settaggio di alcuni parametri:

- *min pattern length* è stato impostato a 4, al fine di ottenere sottosequenze lunghe almeno 4 giorni;
- *max gap* è stato impostato a 1, al fine di ottenere sottosequenze contigue.

I patterns sono stati estratti impostando differenti livelli di *minsup*:

minsup = 20% (136)

patterns	**support**
‘ffff’	247
‘aaaa’	237
‘efff’	238
‘aaab’	230
‘fffe’	220
‘baaa’	206
‘bbaa’	168
‘eeff’	164
‘aabb’	160
‘ffee’	136

minsup = 25% (169) e **minsup = 30% (203)**

patterns	**support**
‘ffff’	247
‘aaaa’	237
‘efff’	238
‘aaab’	230
‘fffe’	220
‘baaa’	206

minsup = 35% (237)

patterns	**support**
‘ffff’	247
‘aaaa’	237
‘efff’	238

Variando il supporto minimo tra 0,20 e 0,35, non è stata trovata nessuna sottosequenza di lunghezza 5. I patterns più frequenti (minsup = 0,35) consistono in sottosequenze di 4 items dai valori molto alti (identificati dalla lettera ‘f’) o molto bassi (identificati dalla lettera ‘a’).

(Alternative) Classification methods

3.1 - Abalone dataset

Il dataset utilizzato per lo svolgimento del task di classificazione, contiene i dati provenienti da uno studio effettuato sugli abaloni, un genere di molluschi gasteropodi marini. Il dataset si compone di 4177 records, ognuno dei quali caratterizzato da 8 diversi attributi:

- *sex*: attributo categorico che indica il sesso dell'esemplare, assume i valori *M* (male), *F* (female) o *I* (infant);
- *length*: attributo continuo, che indica la lunghezza del guscio, in millimetri;
- *diameter*: attributo continuo che misura il diametro del guscio, in millimetri;
- *height*: attributo numerico continuo che indica l'altezza dell'esemplare, in millimetri;
- *whole*: attributo continuo che indica il peso dell'esemplare in grammi, includendo guscio e viscere;
- *shuked*: attributo continuo che indica il peso delle viscere, in grammi;
- *viscera*: attributo continuo che indica il peso delle viscere, in grammi, dopo il dissanguamento;
- *shell*: attributo continuo che indica il peso del guscio, in grammi, dopo l'essiccazione;
- *rings*: attributo numerico intero, che indica il numero di anelli sul guscio e assume valori compresi tra 1 e 29. Attraverso il numero di anelli è possibile inferire l'età dell'esemplare, che è pari al numero di anelli più uno. Tale attributo rappresenta inoltre, il target per il problema di classificazione.

L'analisi della matrice di correlazione evidenzia che solo l'attributo *height* presenta dei valori di correlazione leggermente bassi. Tra tutti gli altri attributi, invece, si registrano valori di correlazione piuttosto alti; si tratta di una situazione del tutto normale, che suggerisce che al crescere di lunghezza e diametro, cresce anche il peso dell'esemplare. La correlazione più forte è quella tra *length* e *diameter*.

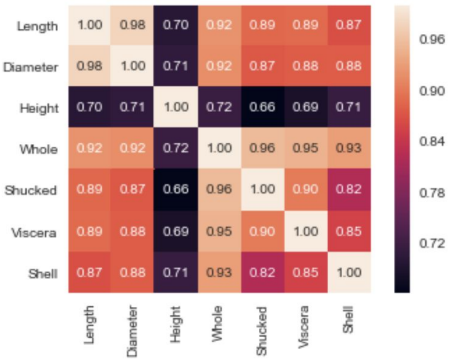


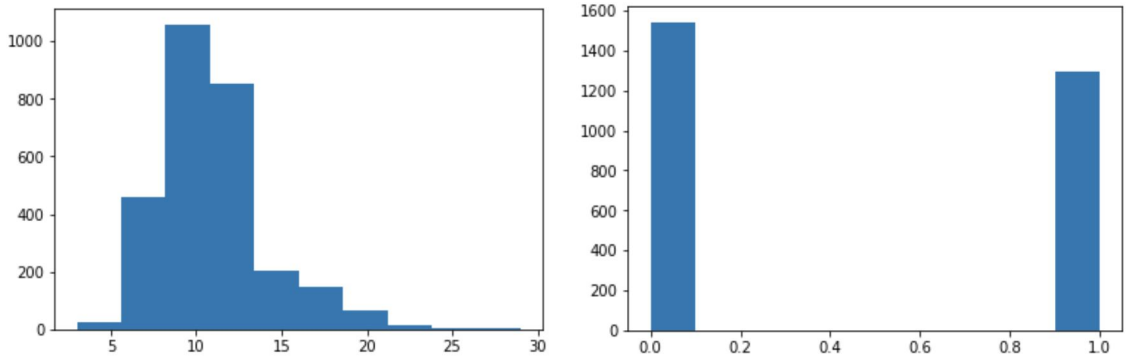
Figura 3.1: Matrice di correlazione

3.2 - Preparazione dei dati

Anche in questo caso, prima di procedere con le analisi, sono state apportate alcune modifiche al dataset:

- ➢ I records con *sex* = *I* sono stati scartati, in quanto poco interessanti rispetto quello che è l'obiettivo del presente task.
- ➢ L'attributo categorico *sex* è stato trasformato nel binario *sex_val* (valore 0 per *F* , valore 1 per *M*).
- ➢ Necessaria ai fini della classificazione è stata la discretizzazione dell'attributo *rings*: valori di *rings* minori di 11 (valore medio) sono contenuti nel bin "0", valori maggiori o uguali di 11 nel bin "1".

I grafici seguenti riportano la distribuzione dei valori dell'attributo *rings* prima e dopo la discretizzazione.



Il dataset finale, pronto per le analisi, conta 2835 records e 7 attributi; a seguito di alcune prove con diverse configurazioni di attributi, sono emerse prestazioni migliori non tenendo in considerazione l'attributo *sex_val*, che per questo è stato escluso.

3.3 - Metodi di classificazione e valutazione

L'obiettivo del task è quello di valutare le capacità predittive di diversi classificatori rispetto la classe *rings* e confrontare le performances.
I modelli analizzati sono stati 4, tutti validati attraverso *cross validation*, impostando k=10, quindi suddividendo il dataset in 10 parti ed utilizzando, ad ogni passo, la k-esima parte come test set e la restante come training set.
La valutazione dei modelli di classificazione è avvenuta mediante l'analisi della performance sul test set e in particolare dei valori di accuratezza e *AUC* (*area under curve*).

- *Decision Tree*: per la scelta dei parametri ottimali, sono state testate diverse configurazioni, valutando come variava l'accuratezza al variare dei valori dei

parametri. La profondità massima è stata impostata a 5 e il criterio di split scelto è stato *entropy*, seppur non si è notata una notevole differenza con *gini*.
Accuratezza del modello = 70%

- *K-nearest neighbors*: il valore ottimale di *k* è stato impostato dopo aver effettuato diverse prove; 13 è il valore che permette di minimizzare l'errore e ottenere risultati migliori.
  Accuratezza del modello = 71%

- *Neural Network*: a seguito di varie prove, la configurazione ottimale della rete neurale risulta caratterizzata da 5 *hidden layers* e 10 *hidden units*.
  Accuratezza del modello = 73%

- *Bagging*: si tratta di un metodo di classificazione di tipo *ensemble*, basato su *bootstrap*. Ciascun classificatore viene addestrato su una ridistribuzione casuale del training set; viene "estratta" la classe di maggioranza predetta. Il metodo di classificazione di base è stato un albero decisionale e *gini* il criterio di split scelto; il numero di classificatori costruiti è stato impostato a 25. Il modello è stato iterato 100 volte e i valori risultanti sono stati plottati nell'istogramma in figura (x).
  Accuratezza del modello = 72%

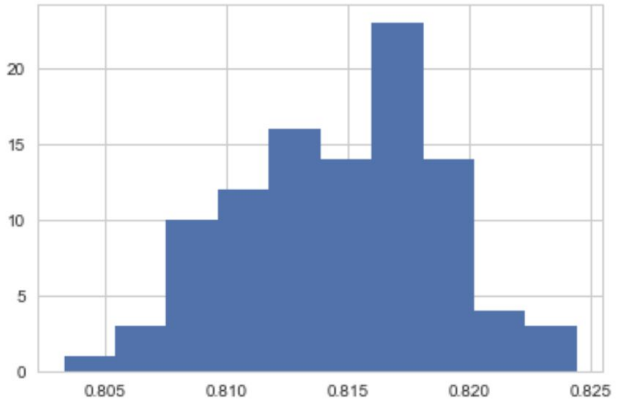


Figure 3.2: Andamento AUC per 100 iterazioni

3.4 - Roc curves e confronto dei modelli

Il plot delle curve ROC dei singoli modelli (figura(x)), permette di confrontare le performances.

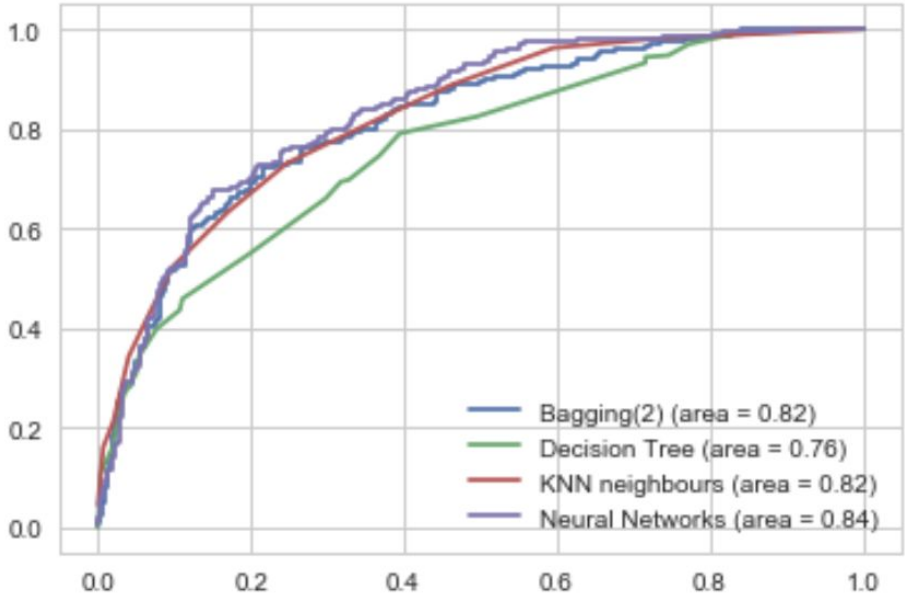


Figure 3.3: Roc curves e AUC

Dall'analisi delle curve ROC, il metodo più efficace risulta essere Neural Network; una performance discreta hanno realizzato invece gli algoritmi K-nn e Bagging. Poiché i risultati dei singoli modelli non si discostano in maniera netta e i valori di accuratezza e AUC non risultano eccezionali, si può pensare che nessuno dei classificatori esaminati risulti particolarmente adatto al presente problema di classificazione binaria.

Outlier Detection

Per lo studio degli outliers è stato utilizzato lo stesso dataset del punto precedente. L'obiettivo del task è identificare l'1% di record aventi un *outlier score* maggiore, che, considerando il dataset completo di 4177 records, equivale ai primi 41 outliers.
Per l'analisi sono stati applicati tre approcci: DBscan, *Local Outlier Factor* e *Convex Hull*.

4.1 – DBscan

È una tecnica *distance-based* che classifica un punto come outliers o meno, sulla base delle distanze dai suoi vicini. Dato ε, cioè il raggio della circonferenza con centro corrispondente al punto che si vuole classificare, se almeno π (%) records stanno entro ε, allora il punto non sarà classificato come outlier. Nel nostro caso, per ottenere l'1% di outliers, sono stati impostati i seguenti parametri:

- ε: 0.67
- minpts: 5 (rappresenta il corrispondente di π, in valore assoluto).

Il grafico in Figura 4.1 mostra la distribuzione dei dati; gli outliers sono colorati di rosso mentre gli inliers sono colorati di blu.

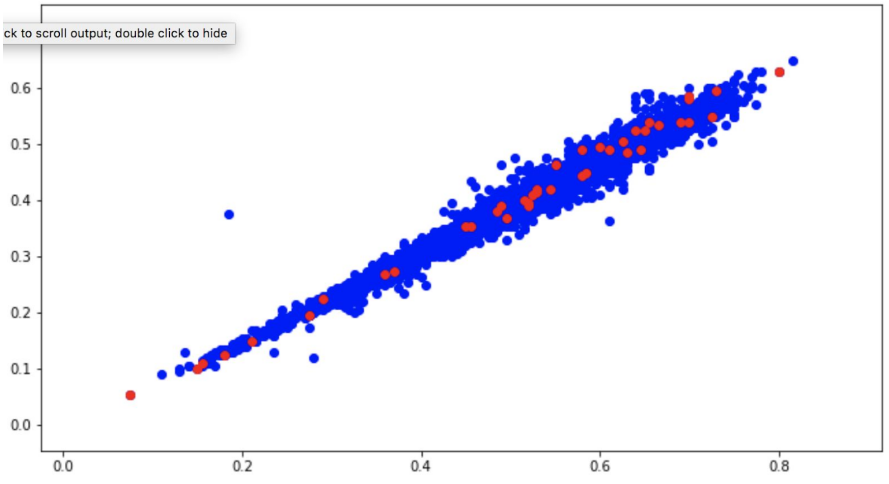


Figura 4.1: Inlier/Outlier con DBscan

I 41 punti classificati come outliers sono i seguenti:

[232, 236, 270, 275, 294, 314, 351, 355, 433, 480, 514, 520, 521, 526, 530, 592, 613, 621, 628, 658, 660, 664, 673, 675, 719, 891, 2051, 2108, 2167, 2201, 2206, 2209, 2334, 2335, 2371, 2436, 3149, 3280, 3359, 3914, 3928]

4.2 - Local Outlier Factor (LOF)

È una tecnica basata sul concetto di densità locale, che confronta la densità attorno a un punto con la densità intorno ai suoi vicini. Il valore della densità è stimato utilizzando la distanza del punto dai suoi K nearest neighbors. I punti la cui densità è considerevolmente minore di quella attorno ai propri vicini sono considerati outliers. Nel nostro caso, K è stato impostato pari a 20.

Il grafico mostra la distribuzione dei dati; gli outliers sono i punti evidenziati di rosso.

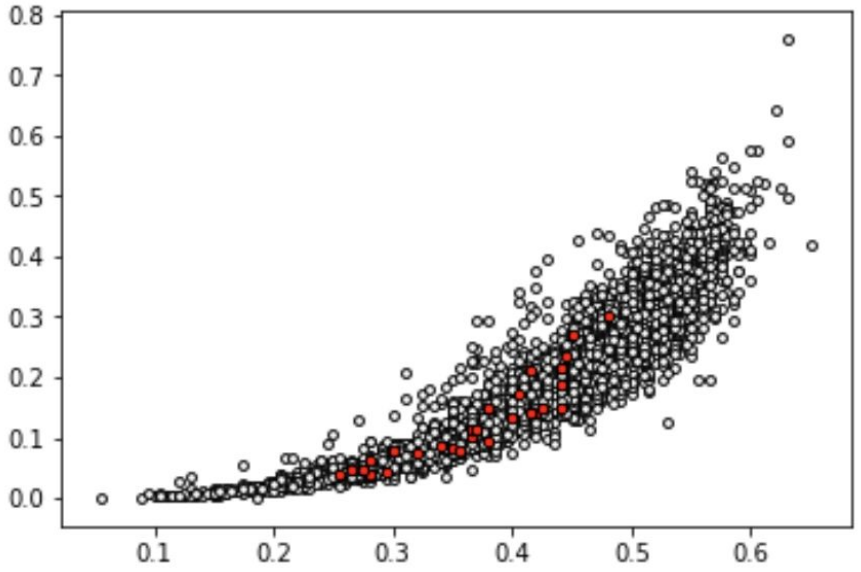


Figura 4.2: Inlier/Outlier con LOF

I 41 punti classificati come outliers sono i seguenti:
[514, 2206, 3366, 3358, 524, 650, 3906, 538, 517, 611, 697, 3161, 3308, 519, 3159, 521, 525, 3318, 2711, 2123, 894, 3801, 3521, 1054, 3996, 236, 3201, 2167, 2115, 523, 2051, 3472, 720, 306, 238, 237, 1429, 518, 2381, 2380, 2627]

4.3 - Convex Hull

È una tecnica basata sul concetto di profondità, che organizza gli oggetti del dataset in livelli convessi; dato k, un punto è definito outlier se ha profondità ≤ k.
Per l'applicazione di questa tecnica, a causa di problemi computazionali, si è deciso di utilizzare un dataset ridotto, rimuovendo:

- i records con *sex* = I
- gli attributi *sex* e *length*.

Come potenziali outliers, nel nostro caso, si sono considerati tutti i punti appartenenti al primo livello che nel grafico sono tutti i punti di colore blu.

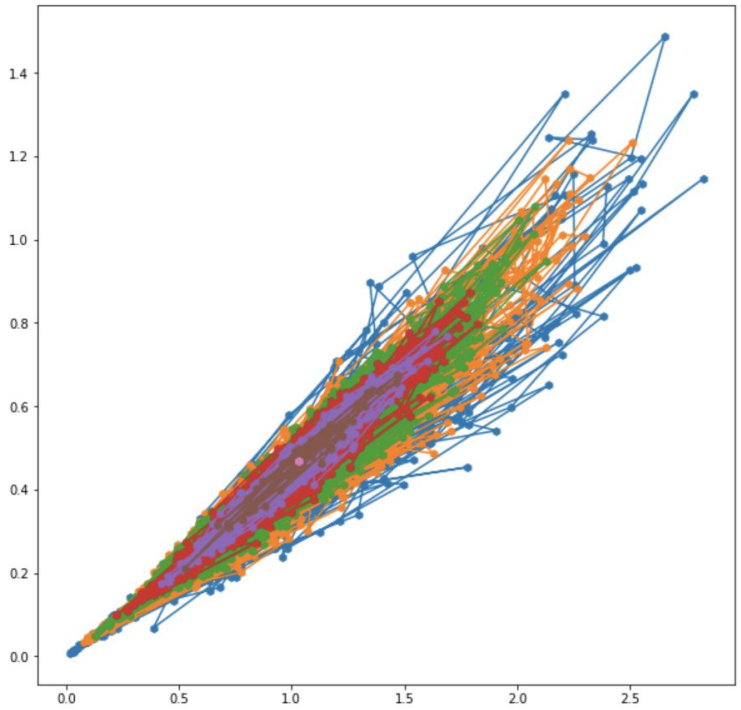


Figura 4.3: Distribuzione dei dati con approccio basato sulla profondità

4.4 - Conclusioni

Dopo aver applicato tre diversi metodi per il rilevamento degli outliers, possiamo dire che con DBscan e LOF si rilevano 5 punti in comune classificati come outliers: [236, 514, 521, 2051, 2167, 2206]. Un solo punto (2051) è classificato come outlier da tutte e tre le tecniche.