UNIVERSITÀ DI PISA

DIPARTIMENTO DI INFORMATICA

Corso di Laurea Magistrale in Informatica

Clustering e Predizione nel Mercato Azionario: Un'Analisi Quantitativa

Relatori:

Anna MONREALE

Lorenzo MANNOCCI

Francesca NARETTO

Candidato:

Alessandro STEFANELLI

ANNO ACCADEMICO 2022/2023

1 Introduzione

Il progetto si concentra sull'analisi dei dati di circa 3500 delle più grandi aziende del mondo appartenenti al mercato azionario NASDAQ.

Per l'analisi dei dati, è stata utilizzata una serie di strumenti di data mining. Queste tecniche hanno permesso di raggruppare le aziende in base alle loro caratteristiche e di identificare pattern nei dati. Inoltre, sono state utilizzate tecniche di analisi predittiva per prevedere l'etichetta del settore di interesse per ciascuna azienda.

Per implementare queste tecniche, sono stati utilizzati vari pacchetti e librerie Python. Tra questi, pandas per la manipolazione dei dati, numpy per le operazioni matematiche, matplotlib e seaborn per la visualizzazione dei dati e scikit-learn per l'implementazione di algoritmi di machine learning, in particolare, è stato utilizzato per implementare gli algoritmi di clustering e di analisi predittiva. Questa libreria fornisce un'ampia gamma di algoritmi di apprendimento supervisionato e non supervisionato, rendendola uno strumento ideale per l'analisi dei dati. Inoltre, è stato utilizzato yfinance per scaricare i dati storici del mercato azionario, e tensorflow e keras per costruire e addestrare i modelli di deep learning come le reti neurali. Questi strumenti hanno permesso di eseguire un'analisi dettagliata e di ottenere intuizioni utili.

Il progetto ha due obiettivi principali. Il primo è esplorare e comprendere i dati attraverso l'analisi dei cluster. Il secondo è l'analisi predittiva, i.e. prevedere per ciascuna azienda l'etichetta che indica il settore di interesse dell'azienda.

Questo può essere utile per una serie di applicazioni, come l'identificazione di potenziali opportunità di investimento o la comprensione delle dinamiche del mercato e può quindi contribuire a migliorare la comprensione del mercato azionario fornendo intuizioni preziose per gli investitori, gli analisti e altre parti interessate.

Il progetto si articola in tre parti principali:

1. Comprensione e Preparazione dei dati: questo include l'esplorazione del dataset con gli strumenti analitici studiati, il miglioramento della qualità dei dati, ad esempio gestendo i valori mancanti e standardizzando i dati, ed infine l'estrazione delle caratteristiche statistiche che saranno utilizzate nella terza parte del progetto.

2. Analisi del clustering: questo compito prevede l'esplorazione del dataset utilizzando varie tecniche di clustering e la descrizione dettagliata delle decisioni prese per ciascun algoritmo. L'obiettivo è analizzare i risultati del clustering, osservando i pattern nei dati del cluster rispetto alle variabili di input, all'anno delle serie temporali, al settore, ecc.

3. Analisi predittiva: l'obiettivo è prevedere per ciascuna azienda l'etichetta del settore dell'azienda target. Questo compito può essere eseguito utilizzando modelli di machine learning basati su dati tabulari e modelli adatti per le serie temporali.

2 Comprensione e Preparazione dei dati

I dataset sono stati presi da Kaggle, un sito che permette agli utenti di trovare o pubblicare dataset. I dataset utilizzati in questo progetto sono circa 3500 dove ogni dataset rappresenta un'azienda, e all'interno di ogni dataset sono presenti 4 serie temporali dove ogni serie temporale rappresenta un

aspetto specifico del prezzo delle azioni dell'azienda nel corso del tempo. Queste quattro serie temporali sono 'open', 'high', 'low' e 'close', che sono rispettivamente il prezzo di apertura, il prezzo più alto, il prezzo più basso e il prezzo di chiusura delle azioni dell'azienda per ogni giorno di trading. I dataset includono dati di più di un anno.

Un aspetto fondamentale della comprensione dei nostri dati è stato determinare l'inizio di ogni dataset di ogni azienda:

```
1962-01-02 00:00:00
1962-01-02 00:00:00
...
2023-06-02 00:00:00
2023-06-02 00:00:00
```

Questo è stato particolarmente importante per decidere su quanti anni condurre la nostra analisi e per identificare quali aziende avevano una durata sufficiente per essere analizzate in modo significativo. Un altro passaggio cruciale della fase di comprensione dei dati è stato assicurarsi che tutti i dataset sotto forma di file CSV avessero gli stessi attributi. Questo è importante per garantire la combinazione o il confronto di dati tra le diverse aziende in modo coerente.

```
All CSV files have the same columns.
The following columns:
['ticker', 'date', 'open', 'high', 'low', 'close']
```

Una volta confermato che le colonne fossero le stesse, è stata presa la decisione di concentrarsi esclusivamente sulle colonne 'date' e 'close' dei dati. Questa scelta è stata guidata da diverse considerazioni:

- In primo luogo, la colonna 'date' è fondamentale per qualsiasi analisi di serie temporali. Fornisce il contesto temporale per ciascuna osservazione, permettendo di vedere come le variabili cambiano nel tempo. Senza questa colonna, non sarebbe possibile tracciare l'andamento delle azioni nel tempo o eseguire analisi categoriche su di esse.

- In secondo luogo, è stata scelta la colonna 'close' perché rappresenta il prezzo di chiusura di un'azione alla fine di ogni giornata di trading. Questo è spesso considerato come il dato più significativo da analizzare in una serie temporale di un mercato azionario. Il prezzo di chiusura è il valore "finalizzato" di un'azione per la giornata di trading e riflette tutte le notizie, gli eventi e le attività di trading che sono avvenute durante il giorno. Di conseguenza, è spesso utilizzato come indicatore chiave del valore di un'azione e può fornire intuizioni preziose sull'andamento generale del mercato azionario.

- Infine, concentrarsi su queste due colonne ha semplificato i dati e ha reso l'analisi più gestibile e meno onerosa dal punto di vista computazionale.

Un aspetto critico è stata la gestione dei valori mancanti, o NaN, nei dati. Questi valori possono causare problemi in molte tecniche di data mining e di analisi dei dati, quindi è stato importante identificarli e gestirli in modo appropriato.

Nel progetto, è stato adottato un approccio in due fasi per gestire i valori mancanti:

1. Per ogni dato mancante, è stata calcolata la media tra il valore che lo precedeva e quello che lo seguiva, e il valore mancante è stato sostituito con questa media. Questo metodo è basato sull'idea che i dati di mercato tendono a rimanere in un certo range da un giorno all'altro, quindi la media tra il giorno precedente e il giorno successivo è una stima ragionevole del valore mancante.

2. Tuttavia, in alcuni casi, un valore mancante poteva essere il primo o l'ultimo valore nella serie, quindi non ci sarebbe stato un valore precedente o successivo da utilizzare. In questi casi, il valore mancante è stato sostituito con il valore successivo o precedente disponibile. Questo ha assicurato che tutti i valori mancanti siano stati gestiti.

Dopodiché é stata effettuata la selezione dei dati rilevanti per l'analisi. In particolare, è stato deciso di concentrarsi sui dati dal 2 gennaio 2018 in poi. Questa decisione è stata presa per garantire che l'analisi fosse basata su dati recenti e pertinenti per le condizioni attuali del mercato.

Un altro motivo per questa scelta è stato quello di rendere le operazioni di elaborazione dei dati più efficienti e meno onerose poiché limitando l'analisi a un periodo di tempo più breve, è stato possibile ridurre la quantità di dati da elaborare, accelerando così i calcoli e riducendo l'onere computazionale. Tuttavia, è stato anche importante assicurarsi che la quantità di dati selezionati fosse ancora sufficientemente rappresentativa per fornire risultati affidabili.

Per assicurarsi che l'analisi fosse basata su dati recenti e pertinenti, è stato esaminato, in primo luogo, ogni dataset per determinare la data più antica presente. Se l'insieme di dati conteneva esclusivamente informazioni posteriori al 2 gennaio 2018, o se mancavano del tutto le date, veniva escluso dall'analisi. Questo processo di filtraggio è stato eseguito da un algoritmo specificamente progettato per questo scopo. L'output parziale di questo algoritmo, mostrato di seguito, evidenzia i file rimossi a causa della mancanza di date o della presenza di date di inizio troppo recenti:

```
2020-07-01 00:00:00
File ACCD.csv removed.
2022-05-13 00:00:00
File ACDC.csv removed.
2004-04-14 00:00:00
2018-01-26 00:00:00
File ACET.csv removed.
1995-09-14 00:00:00
1996-01-25 00:00:00
1994-03-04 00:00:00
2021-03-31 00:00:00
File ACHL.csv removed.
...
```

Successivamente, è stata effettuata una selezione dei dataset ritenuti idonei, mantenendo solo i dati a partire dal 2 gennaio 2018 fino al 9 giugno 2023. Questo ha permesso di isolare le informazioni più recenti e pertinenti per l'analisi. I dati selezionati sono stati archiviati in una nuova cartella, pronti per l'analisi nella fase successiva del progetto.

Per assicurare l'uniformità delle date, è stato realizzato un controllo incrociato su tutte le date dei diversi dataset e verificato l'allineamento delle stesse, garantendo un ulteriore livello di affidabilità.

Nel contesto di questo progetto, l'obiettivo principale è prevedere il settore industriale di una determinata azienda, tramite un approccio di classificazione supervisionata. L'importanza di tale previsione deriva dalla possibilità di identificare dinamiche di mercato e tendenze emergenti, utili per informare decisioni strategiche in vari settori, tra cui investimenti, marketing e strategia aziendale.

Per facilitare questo compito, è stato costruito un dataset che combina dati finanziari storici con informazioni aziendali aggiuntive, tra cui il settore industriale di appartenenza e la capitalizzazione di mercato, ottenute da Yahoo Finance, una risorsa online leader nel campo.

Se sono stati riscontrati problemi nel recupero di queste informazioni da Yahoo Finance, i file di dati corrispondenti sono stati esclusi dal dataset. Questa misura precauzionale assicura che tutti i dati utilizzati nell'analisi siano completi e accurati.

Dopo aver raccolto queste informazioni, sono state organizzate in un formato che facilitasse l'analisi dei dati. Un esempio di tale formato può essere visto nella seguente tabella:

```
company capitalization sector
0 ACCD 991849984 Healthcare
1 ACDC 2008131968 Energy
... ... ... ...
3484 ZYNE 17877014 Healthcare
3485 ZYXI 356356000 Healthcare
```

Inoltre, è stato messo in evidenza quanto sia importante garantire un equilibrio nel numero di aziende per ciascun settore nei dataset originali. Questo equilibrio è cruciale per evitare una predominanza di un settore specifico nei dati.

In conclusione, l'intero processo di acquisizione, verifica e organizzazione delle informazioni ha permesso la creazione di un dataset ottimizzato ed equilibrato, che presenta un numero uniforme di aziende per ciascun settore. Questa caratteristica è stata introdotta per assicurare l'integrità dell'analisi di classificazione supervisionata, evitando distorsioni dovute a una sovrarappresentazione di un settore particolare.

Infine, l'adozione di questa strategia di riduzione delle dimensioni dei dati ha offerto il vantaggio addizionale di ottimizzare l'efficienza computazionale, consentendo un risparmio significativo in termini di potenza di calcolo e tempo di elaborazione.

Ecco il dataset impiegato, che è il prodotto di un'operazione di fusione (merge) eseguita su tutti i file CSV relativi alle diverse aziende:

```
       date       ACER     ACGL   ACGN    ACHC   ACHV    ACIU  \
0     2018-01-02  14.8100  29.43  300.56  33.81  288.00  13.35
1     2018-01-03  15.8100  29.46  304.64  33.43  282.00  13.14
...          ...      ...    ...     ...    ...     ...    ...
1367  2023-06-08   0.9400  70.99    1.62  68.95    5.91   2.22
1368  2023-06-09   0.9622  71.44    1.63  68.30    5.95   2.20

     ...     USLM   VIVK     VNOM   VTNR  WDFC     WHLR    YTEN
     ...     77.60  11.1000  23.60  0.92  116.70   9.1400  82.000
     ...     79.70  11.1000  23.30  0.95  116.90   8.9000  77.600
     ...     ...      ...     ...   ...     ...     ...     ...
```

```
...    191.95   1.1499  26.01  6.18  192.02   0.6700   2.475
...    190.68   1.1500  25.96  6.08  190.92   0.6210   2.430
```

Sul dataset consolidato sono stati effettuati vari controlli, tra cui la gestione dei valori mancanti, per verificarne la validità.

2.1 Normalizzazione dei dati

Un passo fondamentale è stata la normalizzazione dei dati delle serie temporali. Questa procedura modifica i valori nelle serie temporali per ricondurli a una scala comune, risultando essenziale quando si lavora con dati di serie temporali con ampiezze o offset variabili.

La normalizzazione è cruciale anche nell'utilizzo di certi modelli di machine learning e di clustering, come ad esempio il Support Vector Machine (SVM) e il K-Means Clustering, che presuppongono che tutti i dati siano sulla stessa scala. In assenza di normalizzazione, variabili con valori più alti possono dominare quelle con valori più bassi, portando a risultati imprecisi.

Per normalizzare i dati, si è ricorso al metodo di scalatura media-varianza della libreria tslearn. Questo metodo non solo scala i dati in base alla loro varianza, ma consente anche la traslazione dell'offset (anche se in questo caso non è stato necessario dato che le date corrispondevano) e la scalatura dell'ampiezza.

Il processo di normalizzazione è iniziato con la conversione dei dati in un array numpy a 3 dimensioni, requisito per la funzione di trasformazione. In seguito, i dati sono stati scalati utilizzando il metodo di scalatura media-varianza. Infine, i dati scalati sono stati riconvertiti in un DataFrame per analisi successive. Qui di seguito è presentato il DataFrame.

```
             ACER      ACGL      ACGN      ACHC      ACHV      ...
date
2018-01-02 -0.250389 -0.155850  1.597381 -0.127527  1.516163  ...
2018-01-03 -0.246712 -0.159062  1.607931 -0.133570  1.462554  ...
2018-01-04 -0.241947 -0.157739  1.587059 -0.132941  1.387661  ...
2018-01-05 -0.241920 -0.159573  1.599972 -0.136513  1.330814  ...
2018-01-08 -0.246569 -0.160002  1.518660 -0.141582  1.264176  ...
```

Successivamente, è stata realizzata l'estrazione di caratteristiche statistiche dai dati, che saranno utilizzate nella successiva fase di classificazione. Queste caratteristiche offrono una sintesi dei dati, utilizzabile per identificare pattern e tendenze.

Per ogni serie temporale, sono state calcolate diverse statistiche, tra cui media, deviazione standard, volatilità, rendimento medio, massima variazione percentuale, asimmetria e Kurtosis. La media e la deviazione standard offrono informazioni fondamentali sulla posizione centrale e dispersione dei dati, mentre la volatilità - calcolata come deviazione standard delle variazioni percentuali - dà una misura della variabilità dei dati. Il rendimento medio e la massima variazione percentuale forniscono rispettivamente una misura della tendenza al rendimento e dell'ampiezza delle variazioni percentuali. Infine, l'asimmetria e la Kurtosis danno informazioni sulla forma della distribuzione dei dati, misurando rispettivamente la simmetria e la "pesantezza" delle code della distribuzione.

Queste caratteristiche statistiche offrono un riassunto compatto dei dati, utile per la classificazione delle serie temporali.

2.2 PCA

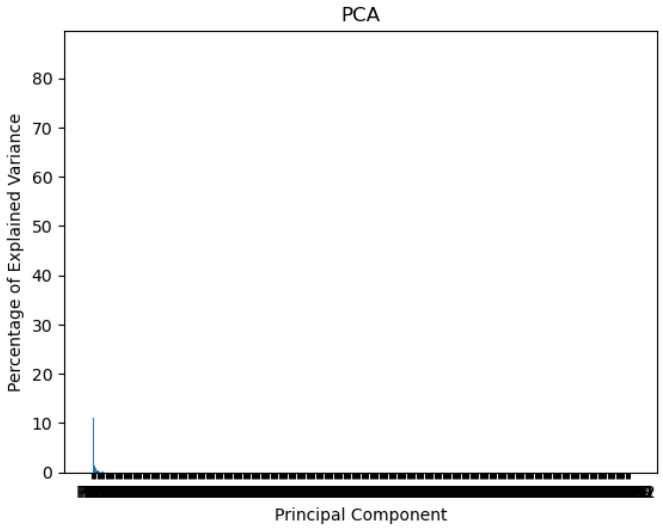
2.2.1 Riduzione della dimensionalità dei dati attraverso la PCA

L'Analisi delle Componenti Principali (PCA) costituisce un strumento efficace per semplificare la visualizzazione dei risultati del clustering. Questa tecnica di riduzione della dimensionalità si rivela particolarmente utile nel trattare dati multidimensionali, i quali possono risultare complessi da interpretare e rappresentare graficamente.

Spesso, i dati multidimensionali possono essere ridotti a due o tre componenti principali senza significative perdite di informazioni. Grazie alla PCA, è quindi possibile trasformare un insieme di variabili potenzialmente correlate in un set di variabili non correlate, denominate componenti principali.

Dopo aver ridotto i dati in questo modo, diventa molto più agevole creare un grafico che illustra la distribuzione dei cluster nello spazio definito dalle componenti principali. Ciò facilita non solo l'interpretazione dei cluster, ma anche l'identificazione di pattern complessi all'interno dei dati.

Come si può identificare nell'immagine, le prime due componenti principali derivanti dalla PCA rappresentano una quota significativa della varianza totale dei dati. Ciò indica che queste due componenti riescono a catturare e rappresentare da sole una grande parte della varietà e della complessità presenti nel dataset originale.



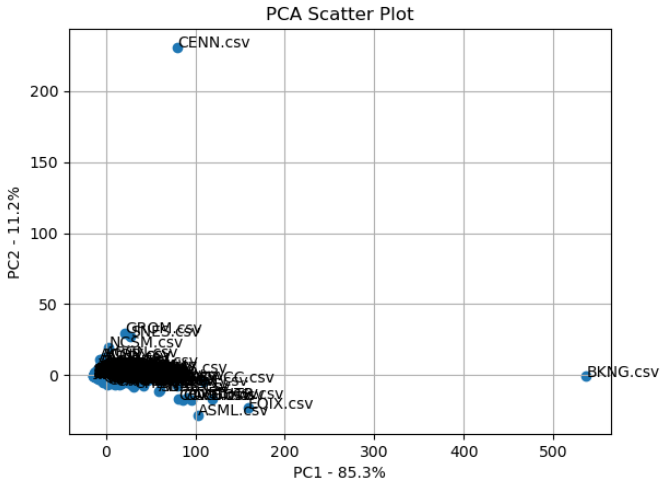
In termini più semplici, ciò implica che, nonostante la riduzione della dimensionalità dei dati, le prime due componenti principali conservano la maggior parte delle informazioni rilevanti. Ciò consente di semplificare l'analisi e la visualizzazione dei dati, senza compromettere troppo l'informazione contenuta.

2.2.2 Visualizzazione e clusterizzazione dei dati PCA

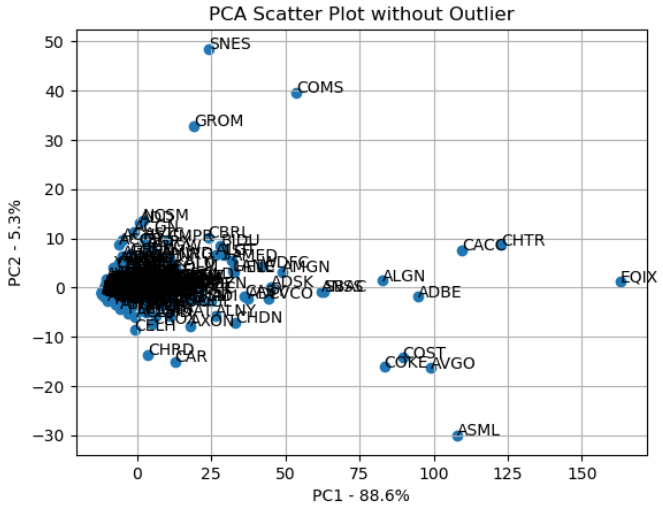
Dopo l'applicazione della PCA sui dati, è stato realizzato un grafico a dispersione per rappresentare i risultati. Questo grafico riporta i valori delle prime due componenti principali (PC1 e PC2) per ogni serie temporale, dove ogni punto rappresenta una specifica serie temporale.

Inoltre, la PCA può essere particolarmente utile per mettere in luce la presenza di outlier e valori anomali nel dataset. Poiché questi elementi tendono a distanziarsi dai cluster principali, la riduzione della dimensionalità può facilitare la visualizzazione e l'identificazione di questi punti anomali. Pertanto,

l'uso della PCA può contribuire a una migliore comprensione e interpretazione dei dati, facilitando l'individuazione di pattern, tendenze, e anomalie all'interno del dataset.



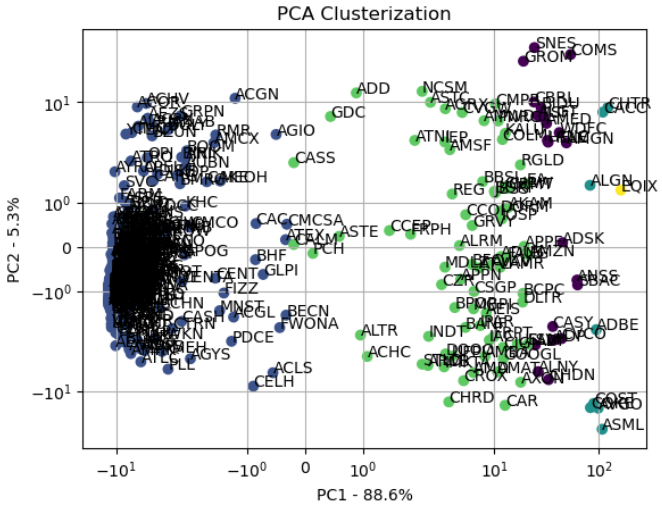
Come evidenziato dal grafico a dispersione PCA, alcune serie temporali si comportano come outlier, distinguendosi chiaramente dal resto dei dati. Per affrontare questo problema, si è deciso di eliminare queste serie temporali specifiche, identificate come outlier, e di eseguire nuovamente l'analisi PCA.



Come si può vedere nella figura sopra, questo passaggio ha permesso di esaminare l'effetto dell'esclusione di tali outlier sulla distribuzione generale dei dati.

Dalla visualizzazione del grafico a dispersione PCA aggiornato, risulta evidente una distribuzione dei dati più omogenea. Tuttavia, i dati appaiono ancora fortemente aggregati, indicando la necessità di una trasformazione per migliorarne la visualizzazione. Considerando la natura della distribuzione dei dati, l'applicazione di una scala logaritmica simmetrica potrebbe rappresentare un efficace compromesso.

Una volta effettuata la trasformazione, è possibile procedere con la clusterizzazione dei dati PCA utilizzando l'algoritmo di clustering K-means per raggruppare le serie temporali.



Dall'esame della figura e della distribuzione dei dati, si evince che l'implementazione della trasformazione ha reso la distribuzione dei dati più marcata e interpretabile. L'analisi del grafico evidenzia due gruppi principali, contraddistinti da due cluster distinti. Il cluster denso a sinistra suggerisce un insieme di dati con caratteristiche affini, indicativo di un trend comune. Al contrario, il cluster più disperso a destra riflette una variabilità maggiore, denotando una diversità di comportamenti all'interno di quel gruppo. Inoltre, i cluster minori potrebbero rappresentare sottogruppi con caratteristiche peculiari.

Grazie a questi passaggi di visualizzazione e raggruppamento, si è potuto esplorare i dati in maniera più approfondita, identificando pattern e tendenze non immediatamente evidenti dai soli dati grezzi.

3 Analisi di clustering

L'analisi di clustering è una tecnica di apprendimento non supervisionato che raggruppa insiemi di dati sulla base della loro somiglianza. Tale somiglianza può essere quantificata in vari modi, ma in questo contesto si basa sulla Dynamic Time Warping (DTW), una misura di distanza utilizzata per comparare serie temporali che possono variare in lunghezza e velocità. La DTW calcola la distanza minima tra due serie temporali, considerando le possibili deformazioni nel tempo.

In questa cornice, l'analisi di clustering serve a raggruppare le aziende in base alle loro similitudini, permettendo di identificare gruppi di aziende con comportamenti di mercato analoghi, che non sarebbero immediatamente evidenti osservando i dati grezzi.

Dopo aver raggruppato le aziende, queste informazioni possono essere sfruttate per fare previsioni o guidare decisioni di investimento. Ad esempio, le informazioni sui cluster potrebbero essere utilizzate per prevedere il comportamento futuro di una specifica azienda, basandosi sul comportamento passato di altre aziende nello stesso cluster (lead-lag behaviour).

Infine, l'analisi di clustering può aiutare ad identificare outlier o anomalie nei dati, che mostrano un comportamento di mercato atipico meritevole di ulteriori indagini.

Nel progetto, sono stati adottati tre metodi di clustering: K-means, DBSCAN e clustering gerarchico.

- Il K-means è un algoritmo partitivo che suddivide i dati in K gruppi.
- DBSCAN, un algoritmo basato sulla densità, identifica regioni di alta densità separate da regioni di bassa densità, utile per scoprire cluster di forma arbitraria e identificare anomalie.

- Il clustering gerarchico costruisce una gerarchia di cluster, fornendo una visualizzazione intuitiva della struttura dei dati e permettendo di esaminare i cluster a vari livelli di granularità.

L'applicazione combinata di diversi metodi fornisce una panoramica dettagliata della struttura dei dati, facilitando un confronto tra i risultati ottenuti.

Un'importante aggiunta a quest'analisi è l'utilizzo della matrice di distanza DTW, che permette la visualizzazione grafica dei risultati del clustering. Trasforma le serie temporali in un formato facilmente visualizzabile e analizzabile, permettendo la creazione di grafici a dispersione e dendrogrammi.

Inoltre, la matrice di distanza DTW agevola la valutazione della qualità dei cluster tramite il punteggio Silhouette, rendendo l'analisi dei dati più dettagliata ed esaustiva.

Infatti, nel calcolo dell'indice Silhouette con scikit-learn, possiamo utilizzare sia dati grezzi che una matrice di distanze precalcolata, a seconda del contesto. Per distanze standard come l'Euclidea, è sufficiente fornire direttamente il DataFrame al metodo silhouette_score, che calcolerà autonomamente le distanze. Per distanze personalizzate o non standard, come la Dynamic Time Warping, è necessario calcolare la matrice di distanza e passarla al metodo silhouette_score con il parametro metric='precomputed'.

3.1 K-Means

Nel progetto, è stato implementato K-Means++, una versione avanzata dell'algoritmo tradizionale K-Means. Il vantaggio di K-Means++ risiede nella scelta iniziale dei centroidi, studiata per minimizzare la possibilità di selezionare punti di partenza non ottimali e migliorare così la qualità del raggruppamento.

3.1.1 Determinazione del numero ottimale di cluster tramite il metodo del gomito

Un elemento chiave dell'analisi è stata l'individuazione del numero ottimale di cluster da impiegare per l'algoritmo di clustering K-means. Questo è stato ottenuto utilizzando il cosiddetto "metodo del gomito", che prevede l'applicazione dell'algoritmo di clustering per un range di numeri di cluster e la misurazione dell'"inerzia" per ognuno di essi. L'inerzia, o somma dei quadrati intra-cluster, è un indice della coesione dei cluster. Essa viene calcolata come la somma delle distanze quadrate tra ogni elemento del cluster e il suo centroide. Matematicamente, l'inerzia per un singolo cluster può essere espressa come:

$$I = \sum_{i=1}^{n} (x_i - \mu)^2$$

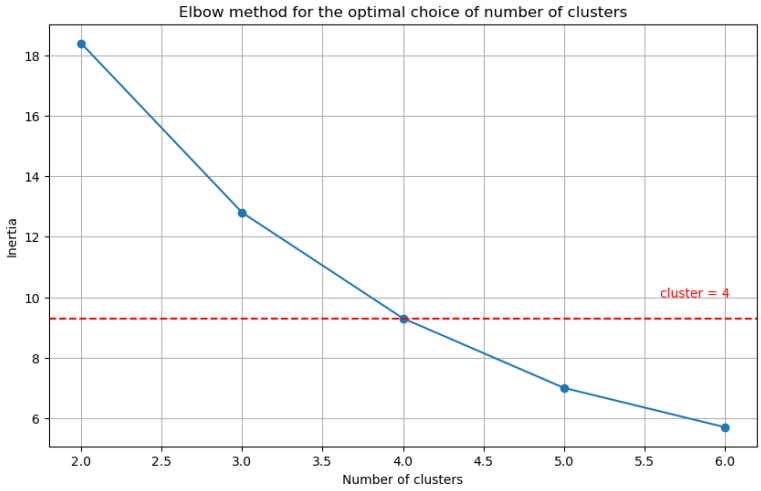
dove:

- I è l'inerzia,
- n è il numero di campioni nel cluster,
- x_i è il vettore di caratteristiche del campione i,
- μ è il centroide del cluster.

Per un insieme di cluster, l'inerzia totale è data dalla somma delle inerzie di ciascun cluster.

Il "metodo del gomito" identifica il punto in cui l'inerzia smette di diminuire in modo significativo all'aumentare del numero di cluster. Tale punto è detto "gomito" e il numero di cluster corrispondente è ritenuto il numero ottimale.

Nel progetto, è stato eseguito per un intervallo di 2 a 6 cluster e il risultato é il seguente:



Nonostante l'assenza di un evidente punto di "gomito", l'analisi del grafico suggerisce un cambiamento verso una pendenza più orizzontale a partire dai 4 cluster. Quest'osservazione, unitamente alla praticità di gestire 4 cluster, ha portato alla decisione di scegliere 4 cluster. Questa scelta rappresenta un equilibrio tra la necessità di una suddivisione significativa dei dati e la gestibilità della soluzione. In seguito, si evidenzierà come i 4 cluster forniscono una rappresentazione intuitiva e maneggevole dei dati, distinguendo i punti in gruppi distinti.

Per quanto riguarda il range di cluster esplorati, la scelta di un intervallo da 2 a 6 è dovuta al fatto che, ad esempio, 9 cluster avrebbero richiesto un tempo computazionale notevolmente più lungo, che non era disponibile per questo progetto. Questo range più ristretto permette un'analisi accurata, pur mantenendo un tempo di elaborazione ragionevole.

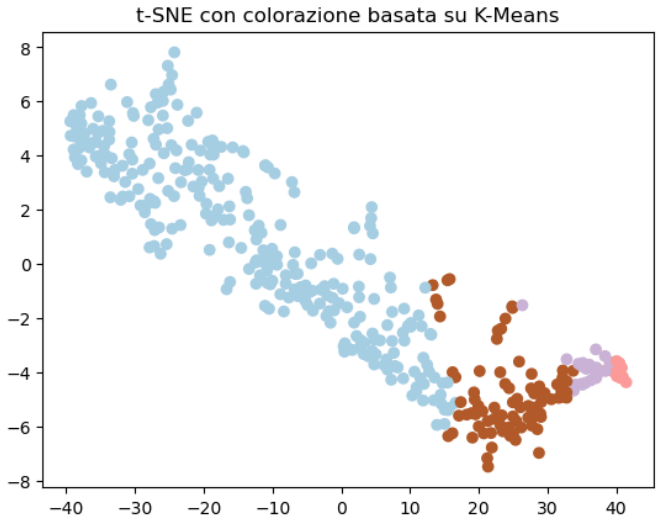
Questo processo ottimizza l'algoritmo di clustering K-means per i dati specifici del progetto, contribuendo a garantire che i risultati dell'analisi siano validi e significativi entro un tempo computazionale ragionevole.

3.1.2 Applicazione dell'algoritmo di clustering K-means e valutazione dei risultati

Dopo aver determinato il numero ottimale di cluster, l'algoritmo di clustering K-means è stato applicato ai dati. Questo algoritmo ha attribuito ogni serie temporale a uno dei quattro cluster, sulla base della sua somiglianza con gli altri dati all'interno dello stesso cluster.

Infine, è stata calcolata la media del punteggio silhouette per valutare la qualità del clustering. Il punteggio silhouette è una misura che varia tra -1 e 1, dove un valore più alto indica che i dati sono ben raggruppati all'interno dei loro cluster e ben separati dai dati negli altri cluster. Il punteggio silhouette medio ottenuto è 0.6551608254379132.

In conclusione, l'algoritmo di clustering K-means dimostra un'efficacia soddisfacente, come evidenziato dal punteggio silhouette. Indica una separazione ben definita tra i cluster e che ogni cluster ha un

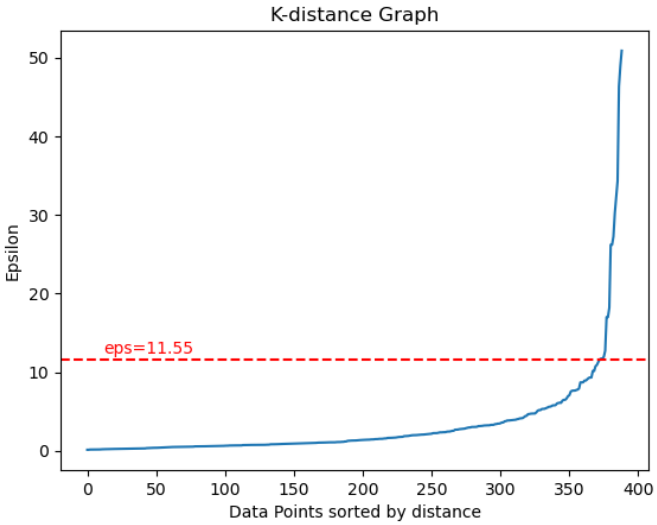


numero adeguato di elementi. Questa distribuzione uniforme dei dati nei cluster permette una lettura e interpretazione più facile e precisa delle dinamiche del nostro dataset. In effetti, una rappresentazione equilibrata e ben separata dei cluster favorisce una comprensione più intuitiva e una migliore caratterizzazione delle relazioni tra i dati.

Successivamente, è stato introdotto DBSCAN, un algoritmo di clustering basato sulla densità che raggruppa i punti in base alla loro vicinanza spaziale. Utilizza il parametro epsilon (EPS) per definire il raggio massimo all'interno del quale i punti sono considerati parte dello stesso cluster. DBSCAN è capace di identificare cluster di forme arbitrarie e rilevare punti di rumore.

Dopo l'applicazione del K-means, è stato impiegato un secondo algoritmo di clustering, DBSCAN. A differenza del K-means, DBSCAN non richiede di specificare il numero di cluster in anticipo e può scoprire cluster di forma arbitraria, il che è particolarmente utile per i dati di serie temporali.

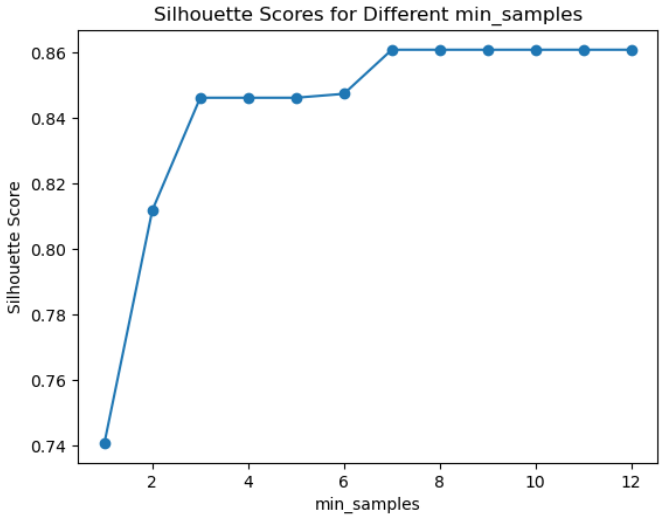
Prima di applicare DBSCAN, è necessario determinare un valore ottimale per EPS. Per fare ciò, è stato utilizzato un grafico di distanza k-nearest neighbors, che ordina i dati in base alla distanza dal loro k-esimo vicino più vicino. Il punto di massima curvatura nel grafico è stato scelto come valore ottimale di epsilon.



Come si può osservare, il valore ottimale risulta essere 11.55.

Successivamente, abbiamo calcolato e regolato il parametro 'min_samples', che determina il numero minimo di punti necessari per formare un cluster. Per affinare ulteriormente l'algoritmo, questo parametro è stato variato da 1 a 12. Per ogni valore assegnato a 'min_samples', abbiamo calcolato il corrispondente punteggio silhouette per valutare l'efficacia del clustering.

Questo processo può essere paragonato al metodo del gomito, ma con una logica invertita. Questo significa che stiamo cercando il punto in cui la curva mostra la crescita più rapida, ovvero la curva sale rapidamente fino a raggiungere un determinato punto, dopo il quale si stabilizza e non mostra più variazioni significative.



Come si può notare, il valore ottimale è 3.

Questo processo di ottimizzazione ha permesso un utilizzo più efficace dell'algoritmo DBSCAN nell'analisi dei dati del progetto.

3.1.3 Applicazione dell'algoritmo di clustering DBSCAN ottimizzato e valutazione dei risultati

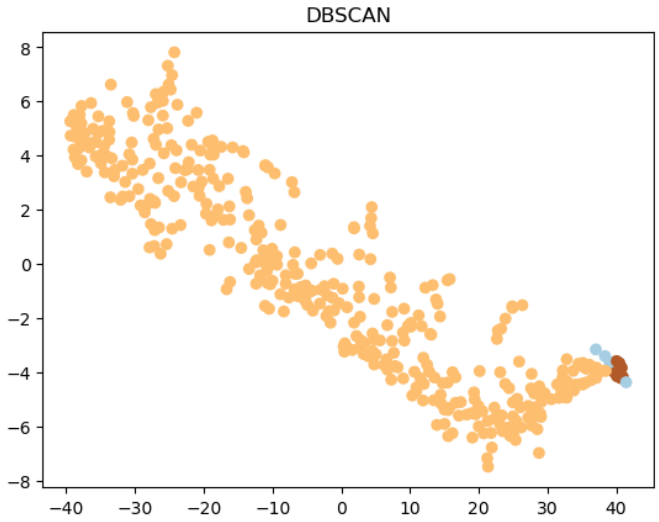
Una volta determinati i valori ottimali per epsilon e min_samples, l'algoritmo DBSCAN è stato implementato sui dati utilizzando tali parametri.

Si può notare che la distribuzione dei cluster non è ottimale. In particolare, si evidenzia la presenza di un unico cluster di dimensioni più ampie, seguito da due cluster di dimensioni ridotte. Questa distribuzione disomogenea suggerisce che il clustering potrebbe non rivelare efficacemente correlazioni latenti nei dati.

Successivamente, è stato calcolato il punteggio silhouette per valutare la qualità del clustering.

```
The Silhouette Score is: 0.8462330751809751
```

Il punteggio silhouette è piuttosto elevato, principalmente a causa della presenza di un grande cluster separato dagli altri. Tuttavia, questo non garantisce la similitudine tra gli elementi all'interno del cluster, in quanto, essendo un cluster molto ampio, potrebbe includere elementi piuttosto diversi tra loro. Questo implica che la coesione interna del cluster potrebbe essere bassa. Un punteggio silhouette alto potrebbe riflettere più l'isolamento del cluster rispetto alla somiglianza degli elementi al suo interno.



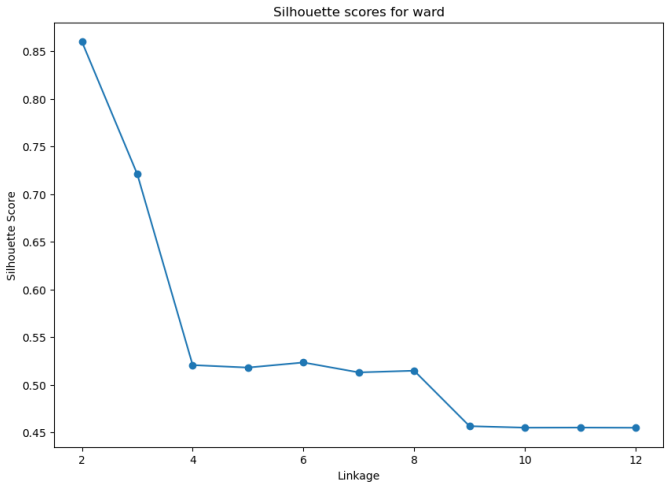
3.2 Clustering Gerarchico

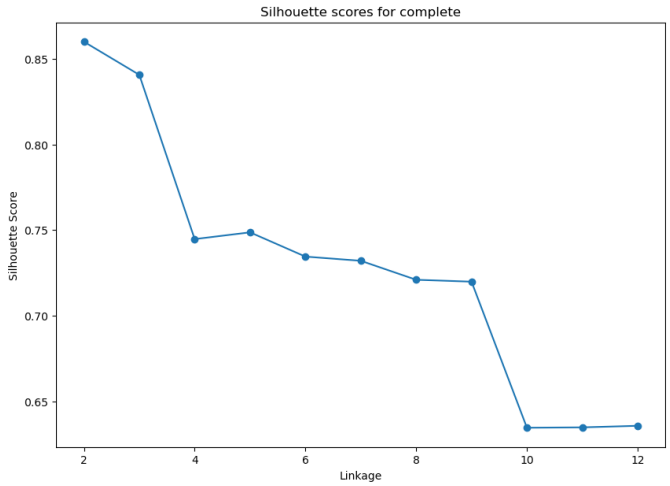
Il clustering gerarchico è un metodo statistico per raggruppare dati in una struttura ad albero di cluster, o dendrogramma. È chiamato "gerarchico" perché costruisce una gerarchia di cluster partendo dai singoli elementi fino all'intero dataset.

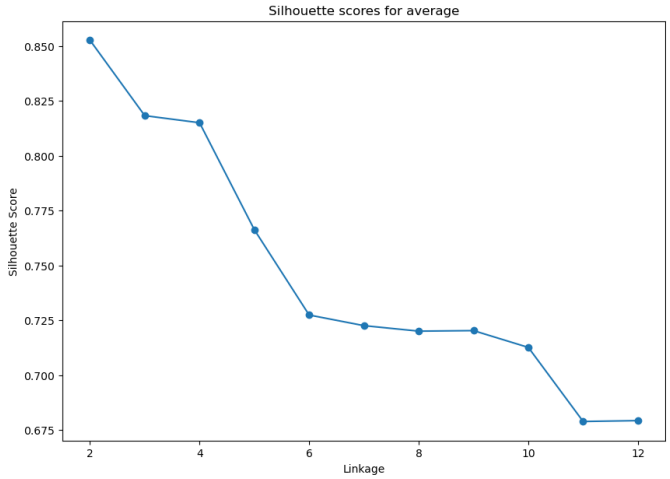
Il suo obiettivo principale è di creare una gerarchia di cluster facilmente visualizzabile e interpretabile, molto utile per comprendere la struttura dei dati.

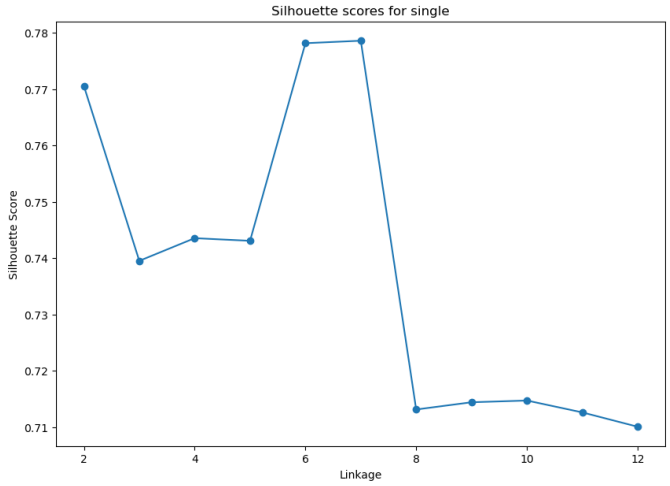
3.2.1 Applicazione dell'algoritmo di clustering gerarchico e valutazione dei risultati

Il clustering gerarchico è stato applicato ai dati utilizzando quattro diversi metodi di linkage: ward, complete, average e single. Per ciascun metodo, il numero di cluster è stato variato da 1 a 12 e il punteggio silhouette è stato calcolato per valutare la qualità del clustering, come illustrato nei seguenti grafici:





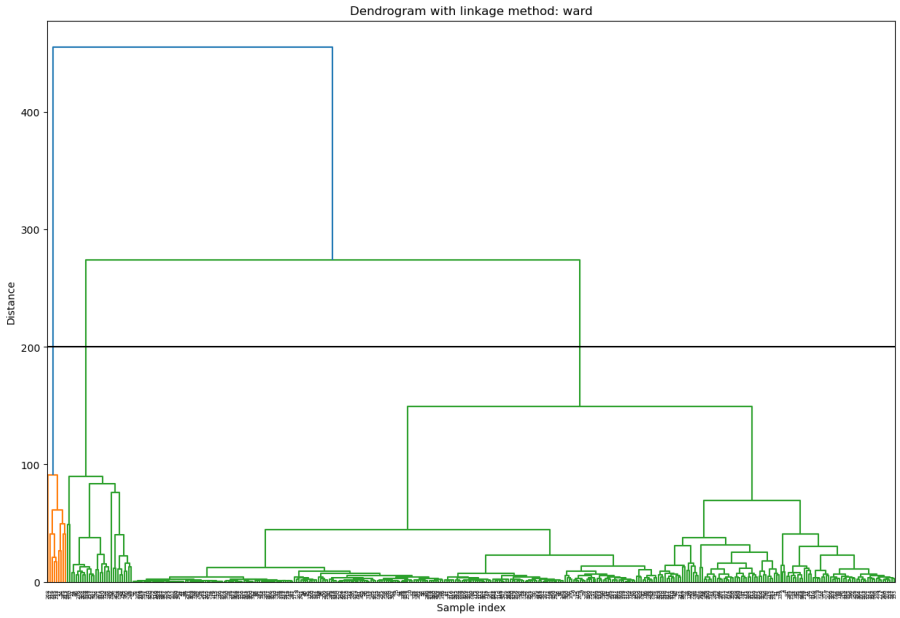




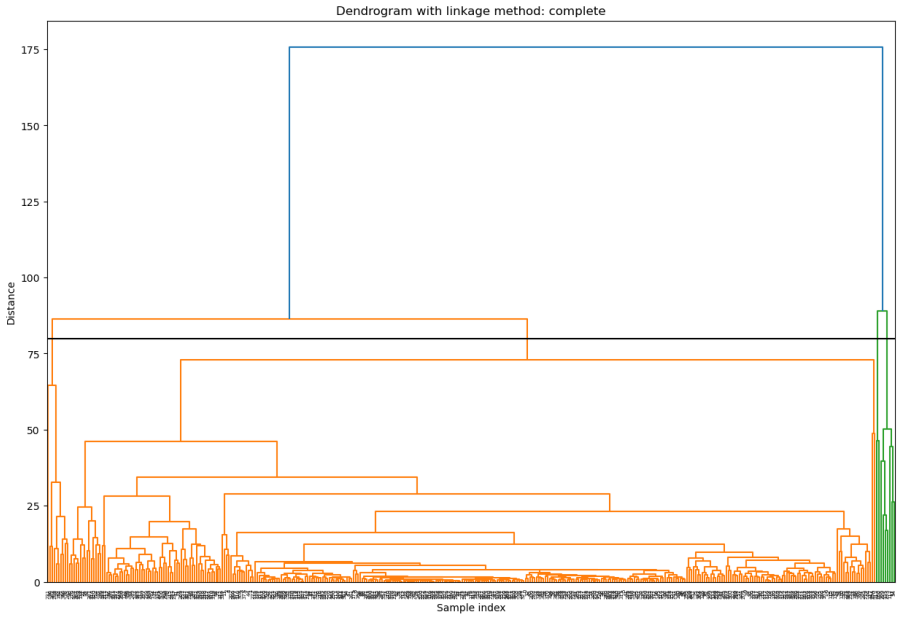
C'è un'importante compromesso da ricercare tra cluster dettagliati e specifici, che potrebbero essere molto numerosi e ricchi di informazioni ma con un basso punteggio silhouette, e cluster più generali, che potrebbero perdere informazioni ma vantare un alto punteggio silhouette, rischiando di non catturare dettagli cruciali. I trade-off per ogni tipo di linkage sono:

- Ward : 3 cluster
- Complete : 4 cluster
- Average : 4 cluster
- Silhouette : 7 cluster

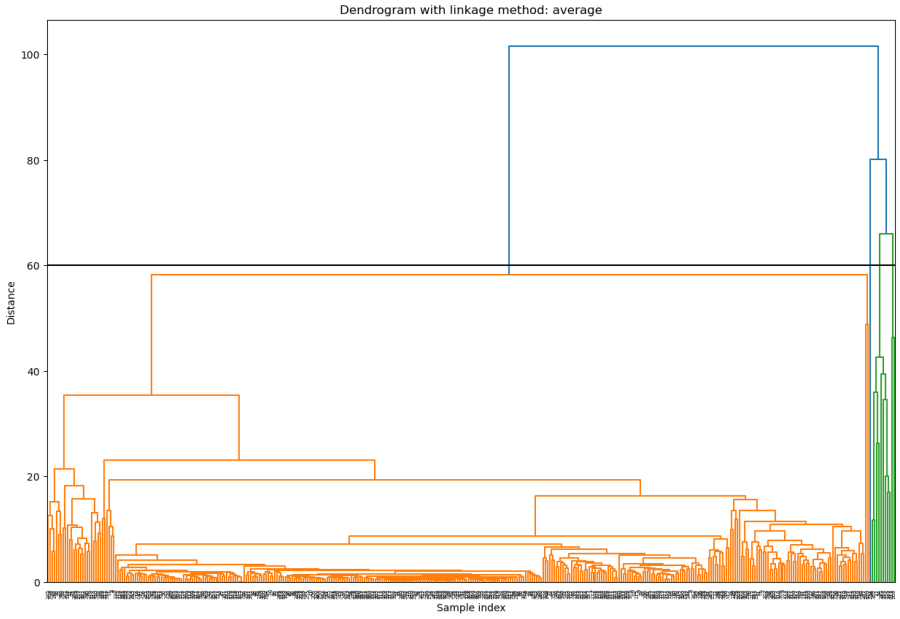
Nell'analisi del dendrogramma, si traccia una linea che taglia il grafico creando i cluster. I punti di sezione nei dendrogrammi correlati e i corrispondenti indici silhouette si presentano nel seguente modo:



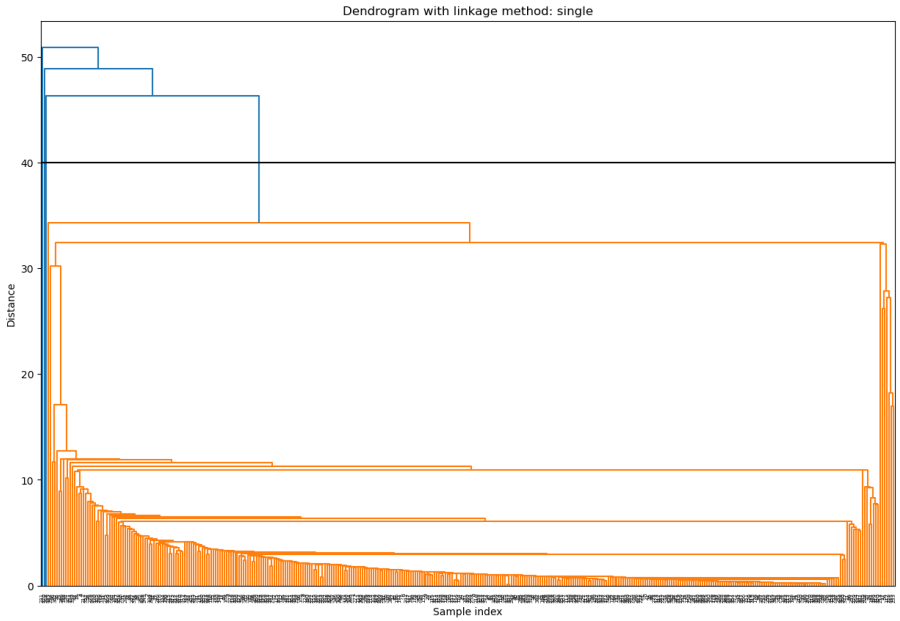

```
The silhouette score for linkage ward is: 0.7587352721929276
```



The silhouette score for linkage complete is: 0.7707167804313383



The silhouette score for linkage average is: 0.806566132796459




```
The silhouette score for linkage single is: 0.7663733398955381
```

Come si può notare, in questa specifica situazione, il "linkage average" risulta essere l'opzione più adatta per il linkage, in quanto più equilibrato e capace di fornire una rappresentazione più precisa della distanza tra i gruppi, creando cluster ben definiti. Si nota inoltre che questo metodo di linkage tende a minimizzare la distorsione introdotta dagli outlier e presenta un eccellente punteggio Silhouette.

3.3 Valutazione finale del miglior approccio di clustering

Tenendo conto di tutte le informazioni precedenti, risulta che il clustering K-means ha fornito i risultati più significativi in questo specifico contesto, per tre ragioni principali:

1. Il numero di cluster: dalle analisi risulta che le tecniche di clustering più efficaci sono il K-means e il clustering gerarchico. Per quanto riguarda il clustering gerarchico, i metodi di linkage 'complete' e 'average' si sono rivelati i più performanti, suggerendo entrambi un numero ottimale di 4 cluster. Si conferma quindi che il numero ideale di cluster per questo contesto specifico è 4, rappresentando il miglior compromesso tra la suddivisione dei dati e la loro gestibilità.

2. Il punteggio silhouette: nonostante non sia tra i più alti, è comunque accettabile, soprattutto considerando l'ottima separazione tra i cluster e il guadagno informativo derivante.

3. Qualità dei cluster: come già menzionato, la qualità dei cluster risulta superiore rispetto alle altre tecniche, portando a un incremento significativo dell'informazione. Questo miglioramento ha permesso di ottenere una comprensione più profonda dei dati.

4 Analisi predittiva

4.1 Preparazione dei dati per la classificazione e l'analisi

Per preparare i dati per la classificazione e l'analisi, sono state selezionate diverse caratteristiche finanziarie di ogni azienda, tra cui la media, la deviazione standard, la volatilità, il rendimento medio, il massimo cambiamento percentuale, l'asimmetria e la Kurtosis del prezzo. Queste caratteristiche sono state poi integrate in un unico DataFrame, come segue:

```
        company      mean   std_dev  volatility  avg_return  max_pct_change  \
0      ACER -0.319643  0.082463    0.029163    0.000666        0.552717
1      ACGL -0.081345  0.079631    0.794634   -0.033719        2.721415
..      ...       ...       ...         ...         ...             ...
387    WHLR -0.354564  0.049464    0.023821    0.000491        0.220087
388    YTEN -0.231839  0.165026   15.473101    0.083524      305.022081

     price_skewness  price_kurtosis              sector
0          1.209650        0.097929          Healthcare
1          1.170366        0.521442  Financial Services
..              ...             ...                 ...
387        2.384504        6.153125         Real Estate
388        0.935145       -0.702816     Basic Materials
```

I settori aziendali e i ticker delle aziende sono stati discretizzati, ossia trasformati in numeri, per la classificazione, dal momento che gli algoritmi utilizzati per tale scopo necessitano di variabili discrete per prevedere il settore. L'azienda stessa è stata codificata numericamente con lo stesso metodo, al fine di conservare le informazioni in un formato utilizzabile. Dopo la discretizzazione dei ticker, questi sono stati utilizzati come indice del DataFrame, mentre il settore è stato rimosso dal DataFrame e memorizzato in una variabile separata, nel modo seguente:

```
Dataframe Settore:
    company  sector
        0      6
        1      5
          ...
        387    8
        388    0
Resto Dataframe:
             mean   std_dev  volatility  avg_return  max_pct_change  \
company
0       -0.319643  0.082463    0.029163    0.000666        0.552717
1       -0.081345  0.079631    0.794634   -0.033719        2.721415
...           ...       ...         ...         ...             ...
387     -0.354564  0.049464    0.023821    0.000491        0.220087
388     -0.231839  0.165026   15.473101    0.083524      305.022081
```

company	price_skewness	price_kurtosis
0	1.209650	0.097929
1	1.170366	0.521442
...	...	...
387	2.384504	6.153125
388	0.935145	-0.702816

Una volta terminata la preparazione dei dati, si può procedere alla predizione della variabile 'settore'.

4.2 Implementazione di vari modelli di classificazione

Nel progetto, sono stati implementati e valutati vari modelli di classificazione per prevedere il settore di appartenenza di ciascuna azienda basandosi sulle sue caratteristiche finanziarie. I dati sono stati suddivisi in set di addestramento e di test, utilizzando l'80% dei dati per l'addestramento e il restante 20% per il test.

Sono stati addestrati e testati diversi modelli di classificazione, tra cui SVM con diverse funzioni kernel, AdaBoost, Random Forest, Naive Bayes, K-Nearest Neighbors (KNN) e una rete neurale. Per ciascun modello, è stata calcolata l'accuratezza delle previsioni sul set di test e generato un report di classificazione che mostra le prestazioni del modello per ciascuna classe (settore), e lo score RSME (Root Mean Squared Error), utile a valutare la precisione di un modello di previsione. Inoltre, per alcuni modelli, è stata utilizzata la ricerca randomizzata con validazione incrociata per ottimizzare i loro iperparametri.

4.2.1 SVM

Ci sono vari modi in cui le SVM possono fare previsioni, e queste variazioni sono determinate da quello che viene chiamato "funzione kernel". Nel progetto sono state testate quattro diversi tipi di funzioni kernel per determinare quale funzionasse meglio:

1. lineare,
2. polinomiale,
3. RBF (Radial Basis Function),
4. sigmoide.

Prima di tutto, questi modelli vengono addestrati utilizzando l'80% dei dati di addestramento, dopodiché i modelli vengono utilizzati per fare delle previsioni sul restante 20% dei dati di test. Successivamente, queste previsioni vengono confrontate con i risultati reali per valutare l'efficacia di ciascun modello. L'accuratezza di ogni modello sarà riportata di seguito, e sulla base di essa verrà scelta la funzione kernel più efficace:

```
{'linear': 0.07692307692307693, 'poly': 0.08974358974358974,
'rbf': 0.08974358974358974, 'sigmoid': 0.08974358974358974}
```

Come si può notare, tutte le funzioni kernel risultano valide ad eccezione di quella lineare. Nonostante ciò, è stata selezionata la funzione polinomiale, da cui è stato ottenuto il seguente report:

```
Classification report:
              precision    recall  f1-score   support

           0       0.09      0.33      0.14         6
           1       1.00      0.00      0.00         6
           2       0.00      0.00      0.00         8
           3       0.00      0.00      0.00        11
           4       1.00      0.00      0.00        10
           5       0.22      0.20      0.21        10
           6       0.00      0.00      0.00        10
           7       0.11      0.29      0.16         7
           8       0.00      0.00      0.00         3
           9       1.00      0.00      0.00         7

    accuracy                           0.08        78
   macro avg       0.34      0.08      0.05        78
weighted avg       0.34      0.08      0.05        78

RMSE:  4.238105527418879
```

Infine, il codice identifica e visualizza il modello con l'accuratezza più alta, indicando quindi quale funzione kernel ha fornito la previsione migliore.

4.2.2 AdaBoost

Per AdaBoost, è stata implementata una ricerca randomizzata con validazione incrociata, che ha prodotto il seguente report:

```
Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         6
           1       0.19      0.50      0.27         6
           2       0.00      0.00      0.00         8
           3       0.22      0.18      0.20        11
           4       0.75      0.30      0.43        10
           5       0.67      0.20      0.31        10
           6       0.43      0.30      0.35        10
           7       0.25      0.29      0.27         7
           8       0.14      0.33      0.20         3
```

9	0.15	0.29	0.20	7
accuracy			0.23	78
macro avg	0.28	0.24	0.22	78
weighted avg	0.32	0.23	0.24	78

4.2.3 Random Forest

La validazione incrociata è stata applicata anche per la Random Forest, ottenendo il seguente report:

Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	6
1	0.12	0.17	0.14	6
2	0.00	0.00	0.00	8
3	0.27	0.36	0.31	11
4	0.50	0.20	0.29	10
5	0.29	0.20	0.24	10
6	0.20	0.30	0.24	10
7	0.20	0.29	0.24	7
8	0.20	0.33	0.25	3
9	0.17	0.14	0.15	7
accuracy			0.21	78
macro avg	0.19	0.20	0.19	78
weighted avg	0.21	0.21	0.20	78

4.2.4 Naive Bayes

In questo caso, non è stata applicata la validazione incrociata a causa della semplicità del modello Naive Bayes. Di seguito si trova il report relativo:

Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	6
1	1.00	0.00	0.00	6
2	0.00	0.00	0.00	8
3	1.00	0.00	0.00	11
4	0.07	0.10	0.08	10
5	0.13	0.70	0.23	10
6	0.00	0.00	0.00	10
7	0.33	0.14	0.20	7
8	0.00	0.00	0.00	3

```
              9       1.00      0.00      0.00         7

       accuracy                           0.12        78
      macro avg       0.35      0.09      0.05        78
   weighted avg       0.36      0.12      0.06        78
```

4.2.5 K-Nearest Neighbors

Per quanto riguarda il KNN, invece, la validazione incrociata risulta necessaria. Il report ottenuto è il seguente:

```
    Report:
                  precision    recall  f1-score   support

               0       0.10      0.17      0.12         6
               1       0.20      0.17      0.18         6
               2       0.12      0.12      0.12         8
               3       0.00      0.00      0.00        11
               4       0.33      0.40      0.36        10
               5       0.10      0.10      0.10        10
               6       0.11      0.10      0.11        10
               7       0.14      0.14      0.14         7
               8       0.17      0.33      0.22         3
               9       0.00      0.00      0.00         7

        accuracy                           0.14        78
       macro avg       0.13      0.15      0.14        78
    weighted avg       0.12      0.14      0.13        78
```

4.2.6 Neural Network

Nel contesto della rete neurale, è stato definito un modello sequenziale con tre strati densi. I primi due strati hanno 256 unità ciascuno, mentre l'ultimo, il livello di output, ha 10 unità, corrispondenti alle 10 classi uniche da prevedere. La funzione di attivazione 'softmax', tipicamente utilizzata nel livello di output per problemi di classificazione multiclasse, produce un vettore di probabilità che somma a 1.

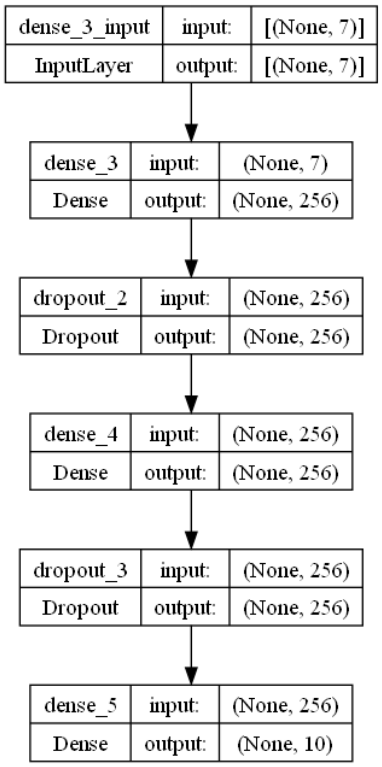
Tra ciascun livello denso sono stati inseriti i livelli di dropout, una tecnica di regolarizzazione impiegata per prevenire l'overfitting. Durante l'addestramento, il dropout "disattiva" casualmente alcune unità nel livello specificato, in questo caso con una probabilità del 10% (0.1). Questo contribuisce a rendere il modello più robusto e a prevenire l'overfitting.

Successivamente, il modello è stato compilato con la funzione di perdita 'sparse_categorical_crossentropy', appropriata per problemi di classificazione multiclasse con etichette intere. È stato utilizzato 'Adam' come algoritmo di ottimizzazione, popolare per la sua capacità di combinare i benefici di altri metodi di ottimizzazione.

Infine, il modello è stato addestrato per 80 epoche, il che indica il numero di volte in cui l'algoritmo di apprendimento attraversa l'intero dataset di addestramento. Si è optato per un batch size di 256, che

rappresenta il numero di esempi di addestramento utilizzati in una singola iterazione, e una divisione di validazione del 20%, che significa che il 20% del set di addestramento è stato utilizzato come set di validazione.

Qui di seguito è riportato il grafico della rete neurale costruita:



La rete neurale ha fornito il seguente report:

Report:

	precision	recall	f1-score	support
0	0.09	0.17	0.12	6
1	0.00	0.00	0.00	6
2	1.00	0.12	0.22	8
3	0.20	0.27	0.23	11
4	0.50	0.20	0.29	10
5	0.50	0.10	0.17	10
6	0.29	0.20	0.24	10
7	0.17	0.57	0.26	7
8	0.00	0.00	0.00	3
9	0.00	0.00	0.00	7
accuracy			0.18	78
macro avg	0.27	0.16	0.15	78
weighted avg	0.32	0.18	0.18	78

4.3 Valutazione dei vari modelli di classificazione

Dalle nostre analisi, è stato possibile confrontare le prestazioni di diversi modelli di classificazione. A giudicare dai report di classificazione ottenuti, nessuno dei modelli ha mostrato risultati eccellenti in termini di precisione, recall e F1-score. Tuttavia, è possibile analizzare i dati per determinare quale modello abbia superato gli altri in termini di prestazioni.

Tra tutti i modelli esaminati, AdaBoost ha raggiunto l'accuracy più alta (23%). Di conseguenza, potremmo preliminarmente concludere che AdaBoost sia il modello più performante.

Ciononostante, l'accuracy non è sempre un indicatore affidabile della qualità di un modello, specialmente quando le classi sono sbilanciate. In queste situazioni, metriche come la precisione, il recall e l'F1-score possono offrire una visione più dettagliata.

Osservando l'F1-score medio ponderato, AdaBoost emerge ancora come il migliore, con un punteggio di 0.24. L'F1-score è una metrica che unisce precisione e recall in un singolo valore.

5 Conclusioni

In conclusione, l'analisi di clustering ha portato a risultati rilevanti e in linea con le aspettative iniziali. L'Analisi delle Componenti Principali (PCA) ha suggerito l'esistenza di due principali cluster, conferma che è stata successivamente approfondita attraverso vari metodi di clustering. Questi hanno efficacemente identificato due macro-gruppi, oltre a diversi cluster minori, all'interno del dataset.

Questi cluster, rappresentando pattern distinti nei dati, possono fungere da base per ulteriori analisi, come l'analisi lead-lag, la quale postula che le variazioni in un gruppo possano anticipare e quindi prevedere le variazioni in un altro gruppo.

Nel contesto specifico, l'uso della Dynamic Time Warping (DTW) è stato particolarmente utile. La DTW è una tecnica che permette di allineare sequenze temporali che possono variare in velocità o in tempo, rendendola adatta per l'analisi lead-lag.

Tuttavia, l'analisi predittiva, nonostante l'impiego di tecniche avanzate di apprendimento automatico e analisi dei dati, non ha raggiunto l'obiettivo di costruire un modello che prevedesse accuratamente le classi e che garantisse una buona classificazione, poiché tutti i modelli mostravano bassa precisione e recall. Questo suggerisce che è estremamente difficile elaborare certi modelli in presenza di fattori esterni non quantificabili, come nel caso del nostro studio sui mercati finanziari.

I mercati finanziari sono influenzati da innumerevoli fattori, molti dei quali sono difficili da quantificare e includere in un modello. Questa complessità rende estremamente arduo prevedere i movimenti futuri dei prezzi.

I nostri modelli hanno inoltre avuto problemi nel distinguere tra diverse classi. Ciò potrebbe essere dovuto alla complessità dei dati finanziari. Ad esempio, due aziende di settori diversi potrebbero avere profili di rendimento simili a causa di fattori macroeconomici comuni, mentre due aziende dello stesso settore potrebbero avere profili di rendimento molto diversi a causa di specifici fattori aziendali.

Per questo motivo, la finanza si concentra prevalentemente sulla previsione di andamenti a brevissimo termine, dove i modelli di apprendimento automatico possono essere più efficaci nel catturare i pattern nei dati. Le previsioni a lungo termine, al contrario, sono estremamente incerte e influenzate da numerosi fattori imprevedibili.

In conclusione, pur fornendo intuizioni preziose e aiutando a identificare pattern nei dati finanziari, è fondamentale riconoscere i limiti dell'apprendimento automatico e dell'analisi dei dati, oltre all'incertezza intrinseca nella previsione dei mercati finanziari.