

**Università Politecnica delle Marche**  
**Facoltà di Ingegneria**

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

---



**Students performance analysis:  
Classificazione e clustering**

**Standard and Poor's 500 prediction:  
Forecasting**

**Docenti**

Prof. Ursino Domenico  
Dott. Marchetti Michele

**Componenti del gruppo**

Bellante Luca  
Coccia Giansimone  
Ferretti Laura

---

ANNO ACCADEMICO 2024-2025

# Contents

<b>1 Student performance analysis: Introduzione</b>	<b>3</b>
1.1 Obiettivi del progetto . . . . .	3
<b>2 Student performance analysis: Dataset</b>	<b>4</b>
2.1 ETL . . . . .	6
2.2 Analisi descrittiva . . . . .	7
2.2.1 Distribuzione delle età . . . . .	7
2.2.2 Distribuzione di genere . . . . .	8
2.2.3 Distribuzione delle etnie . . . . .	9
2.2.4 Distribuzione del tempo di studio settimanale . . . . .	10
2.2.5 Distribuzione delle classi di voto ( <i>GradeClass</i> ) . . . . .	10
2.2.6 Distribuzione delle assenze . . . . .	11
2.2.7 Distribuzione dell'educazione dei genitori . . . . .	12
2.2.8 Matrice di correlazione . . . . .	12
<b>3 Student performance analysis: Classificazione</b>	<b>14</b>
3.1 Preprocessing del Dataset . . . . .	14
3.1.1 Oversampling con SMOTE . . . . .	14
3.1.2 Undersampling con Clusterizzazione . . . . .	15
3.1.3 Dataset Finale . . . . .	15
3.1.4 Analisi della distribuzione degli attributi . . . . .	16
3.2 Scenari di Classificazione . . . . .	16
3.3 Processo di Classificazione . . . . .	17
3.3.1 Ottimizzazione dei Modelli . . . . .	18
3.3.2 Ensemble Learning . . . . .	20
3.3.3 Commento ai risultati . . . . .	25
<b>4 Student performance analysis: Clustering</b>	<b>26</b>
4.1 Fase di ETL per il clustering . . . . .	27
4.1.1 Operazioni preliminari sul dataset . . . . .	27
4.1.2 Riduzione di dimensionalità . . . . .	28
4.2 Kmeans . . . . .	30
4.2.1 Applicazione dell'algoritmo K-means . . . . .	31
4.2.2 Analisi dei risultati ottenuti . . . . .	31
4.3 DBSCAN . . . . .	35
4.3.1 Applicazione dell'algoritmo DBSCAN . . . . .	35
4.3.2 Analisi dei risultati ottenuti . . . . .	36
<b>5 S&amp;P 500 prediction: Introduzione</b>	<b>41</b>

<b>6 S&amp;P 500 prediction: Dataset</b>	<b>42</b>
6.1 ETL - Preparazione serie temporale . . . . .	43
6.1.1 Time series analysis parametro S&P500 . . . . .	43
6.1.2 Time series analysis parametro S&P500 - dati aggregati per mese . . . . .	49
6.2 Metriche finale e conclusioni . . . . .	54
<b>7 Close - Open price prediction</b>	<b>56</b>
7.1 Obiettivo del progetto . . . . .	56
7.2 ETL - Preparazione serie temporale . . . . .	56
7.3 Modello ARIMA . . . . .	59
7.3.1 Risultati . . . . .	61
7.4 Modello SARIMAX . . . . .	63
7.4.1 Risultati . . . . .	64
7.5 Conclusioni . . . . .	66

# 1 Student performance analysis: Introduzione



Il rendimento scolastico degli studenti è influenzato da una combinazione di fattori personali, sociali e accademici. Comprendere queste dinamiche può supportare docenti e istituzioni educative nell'adozione di strategie mirate per migliorare i risultati accademici e il benessere degli studenti. In questo contesto, il presente progetto si propone di analizzare un dataset contenente informazioni relative ai voti di studenti e ad altri fattori che potenzialmente influenzano l'andamento scolastico, come età, supporto genitoriale, attività extracurriculare e assenze.

## 1.1 Obiettivi del progetto

L'obiettivo principale del progetto è l'applicazione di tecniche di machine learning, in particolare di classificazione e clustering, per individuare schemi e relazioni nei dati. Nello specifico, il progetto si articola nei seguenti obiettivi:

- Classificazione: Creare modelli predittivi in grado di stimare la classe di rendimento degli studenti sulla base delle variabili disponibili.
- Clustering: Raggruppare gli studenti in cluster significativi, basati su caratteristiche condivise, per identificare profili comuni di rendimento e comportamento.
- Analisi interpretativa: Esplorare i fattori che contribuiscono maggiormente al successo scolastico, evidenziando il peso relativo delle variabili incluse nel dataset.

Attraverso l'impiego di queste tecniche, il progetto mira a fornire spunti utili per una migliore comprensione delle esigenze degli studenti e per la definizione di interventi educativi personalizzati.

## 2 Student performance analysis: Dataset

Il dataset utilizzato per l'analisi delle performance scolastiche degli studenti è stato reperito dalla piattaforma Kaggle, disponibile al seguente link: <https://www.kaggle.com/datasets/rabieelkharoua/students-performance-dataset>. Il dataset in questione contiene informazioni dettagliate su 2.392 studenti delle scuole superiori, includendo dati demografici, abitudini di studio, coinvolgimento dei genitori, attività extracurricolari e risultati accademici.

Nel dettaglio il dataset contiene le seguenti informazioni:

- **Informazioni sullo studente**

- **StudentID:** Identificativo univoco assegnato a ciascuno studente (da 1001 a 3392).

- **Dettagli demografici**

- **Age:** Età degli studenti, compresa tra 15 e 18 anni.
  - **Gender:** Genere degli studenti, dove 0 rappresenta Maschio e 1 rappresenta Femmina.
  - **Ethnicity:** Etnia degli studenti, codificata come segue:
    - \* 0: Caucasiche
    - \* 1: Afroamericana
    - \* 2: Asiatica
    - \* 3: Altro
  - **ParentalEducation:** Livello di istruzione dei genitori, codificato come segue:
    - \* 0: Nessuno
    - \* 1: Scuola superiore
    - \* 2: Alcuni studi universitari
    - \* 3: Laurea
    - \* 4: Livello superiore

- **Abitudini di studio**

- **StudyTimeWeekly:** Tempo settimanale dedicato allo studio (in ore), compreso tra 0 e 20.
  - **Absences:** Numero di assenze durante l'anno scolastico, compreso tra 0 e 30.
  - **Tutoring:** Stato di partecipazione al tutoring, dove 0 indica No e 1 indica Sì.

- **Coinvolgimento dei genitori**

- **ParentalSupport:** Livello di supporto genitoriale, codificato come segue:
  - \* 0: Nessuno
  - \* 1: Basso
  - \* 2: Moderato
  - \* 3: Alto
  - \* 4: Molto alto

- **Attività extracurricolari**

- **Extracurricular:** Partecipazione ad attività extracurricolari, dove 0 indica No e 1 indica Sì.
- **Sports:** Partecipazione a sport, dove 0 indica No e 1 indica Sì.
- **Music:** Partecipazione ad attività musicali, dove 0 indica No e 1 indica Sì.
- **Volunteering:** Partecipazione ad attività di volontariato, dove 0 indica No e 1 indica Sì.

- **Performance accademica**

- **GPA:** Media scolastica (Grade Point Average) su una scala da 2.0 a 4.0.
- **GradeClass:** Classificazione dei voti in base alla GPA:
  - \* 0: *A* ( $\text{GPA} \geq 3.5$ )
  - \* 1: *B* ( $3.0 \leq \text{GPA} < 3.5$ )
  - \* 2: *C* ( $2.5 \leq \text{GPA} < 3.0$ )
  - \* 3: *D* ( $2.0 \leq \text{GPA} < 2.5$ )
  - \* 4: *F* ( $\text{GPA} < 2.0$ )

Nella tabella 1 vediamo una rappresentazione delle colonne del dataset e il relativo formato dei dati.

Table 1: Colonne del dataset e relativo formato dei dati.

Colonna	Formato
<b>StudentID</b>	Intero
<b>Age</b>	Intero
<b>Gender</b>	Categoriale
<b>Ethnicity</b>	Categoriale
<b>ParentalEducation</b>	Categoriale
<b>StudyTimeWeekly</b>	Intero
<b>Absences</b>	Intero
<b>Tutoring</b>	Categoriale
<b>ParentalSupport</b>	Categoriale
<b>Extracurricular</b>	Categoriale
<b>Sports</b>	Categoriale
<b>Music</b>	Categoriale
<b>Volunteering</b>	Categoriale
<b>GPA</b>	Float (2.0 - 4.0)
<b>GradeClass</b>	Categoriale

## 2.1 ETL

Il processo di ETL è stato fondamentale per garantire la qualità dei dati e la loro adeguatezza per le analisi successive. Durante questa fase, sono state applicate diverse operazioni per preparare i dati:

- Eliminazione dello *StudentID*: La variabile *StudentID* è stata rimossa poiché rappresentava un identificativo univoco privo di rilevanza analitica o predittiva.
- Verifica della bontà dei dati: È stato effettuato un controllo dettagliato per identificare eventuali valori anomali o non coerenti.
- Gestione dei valori nulli: Il dataset è stato analizzato al fine di individuare l'eventuale presenza di valori nulli, che però non sono stati rilevati.

## 2.2 Analisi descrittiva

L'analisi descrittiva svolta ha avuto lo scopo di esplorare e sintetizzare le principali caratteristiche del dataset, fornendo una comprensione preliminare delle variabili e delle loro distribuzioni. Durante questa fase sono stati esaminati i dati demografici, le abitudini di studio, il coinvolgimento dei genitori e le attività extracurricolari degli studenti, con particolare attenzione alle relazioni tra queste variabili e il rendimento accademico.

### 2.2.1 Distribuzione delle età

La distribuzione delle età degli studenti è stata visualizzata tramite un grafico a barre riportato in Figura 1. I dati evidenziano un campione bilanciato, con una lieve prevalenza di studenti di 15 anni. Tuttavia, questa discrepanza non risulta statisticamente significativa.

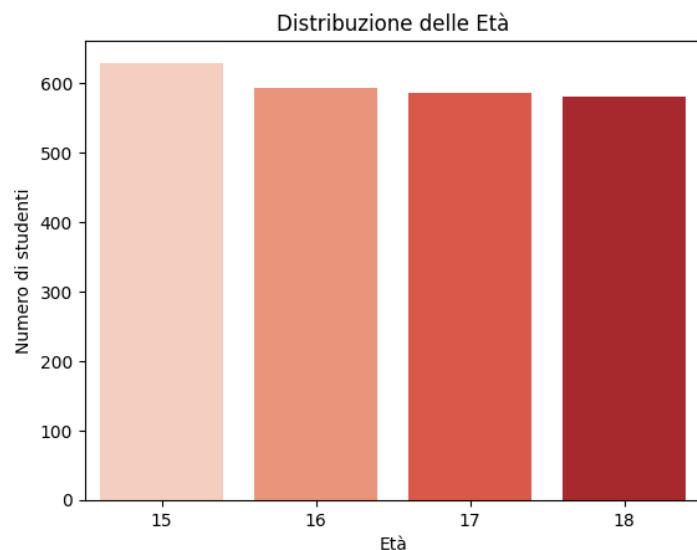


Figure 1: Distribuzione delle età degli studenti.

### 2.2.2 Distribuzione di genere

La composizione di genere è stata analizzata tramite un grafico a torta (vedi Figura 2), che evidenzia una divisione equilibrata tra maschi e femmine. Questo equilibrio garantisce una rappresentatività uniforme nel dataset, eliminando potenziali bias di genere.

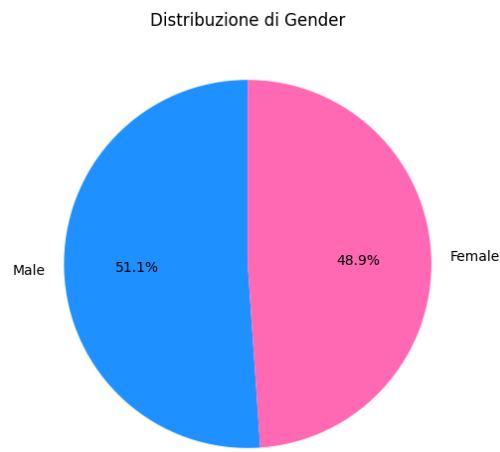


Figure 2: Distribuzione di genere tra gli studenti.

### 2.2.3 Distribuzione delle etnie

La composizione etnica degli studenti nel dataset è rappresentata in Figura 3 tramite un grafico a torta.

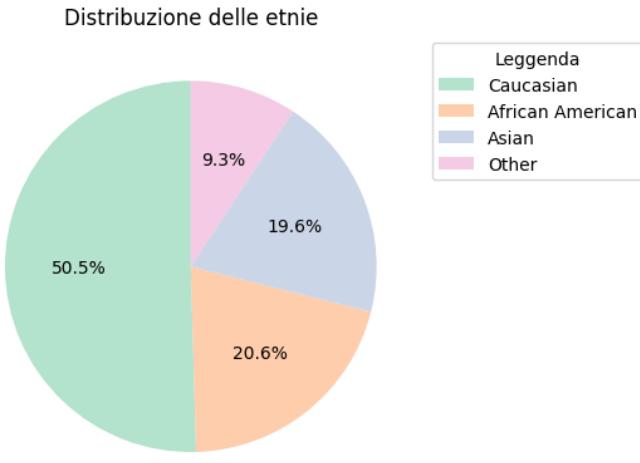


Figure 3: Distribuzione delle etnie degli studenti.

La maggior parte degli studenti appartiene al gruppo *Caucasian*, seguito da *African American*, *Asian*, e *Other*. Sebbene il dataset evidenzi una predominanza di un gruppo etnico, le differenze nelle performance accademiche tra i gruppi sono risultate non significative, suggerendo che l'etnia non rappresenti un fattore discriminante, come si può notare in Figura 4.

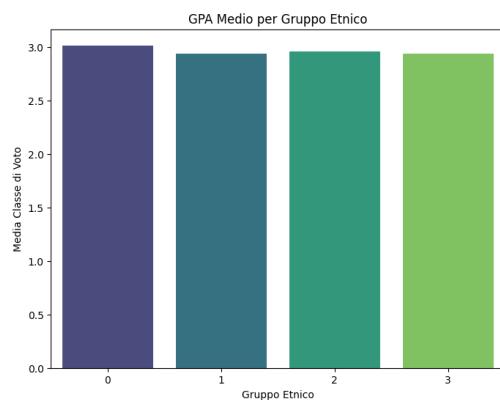


Figure 4: GPA medio per gruppo etnico.

#### 2.2.4 Distribuzione del tempo di studio settimanale

Per analizzare la variabile numerica *StudyTimeWeekly*, invece, è stato creato un istogramma con una curva di densità. La distribuzione mostra come il tempo dedicato allo studio varia tra gli studenti, evidenziando alcuni picchi, non particolarmente accentuati, in corrispondenza di valori specifici (si faccia riferimento alla Figura 5).

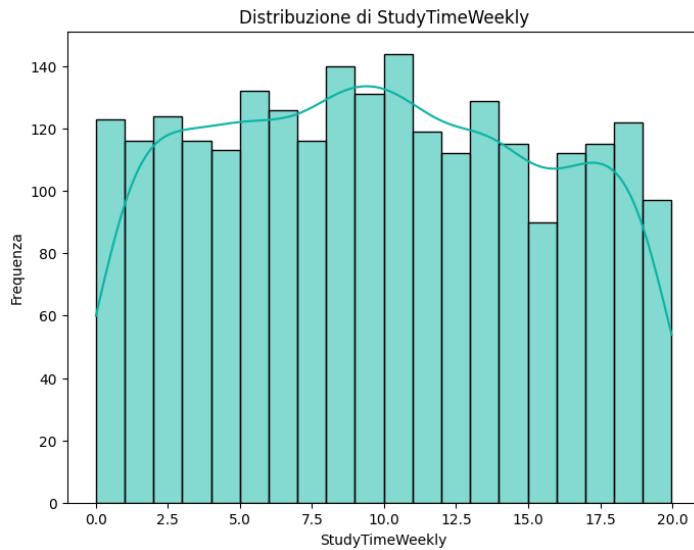


Figura 5: Distribuzione del tempo di studio settimanale degli studenti.

#### 2.2.5 Distribuzione delle classi di voto (*GradeClass*)

La variabile *GradeClass* è stata rappresentata attraverso un grafico a barre per osservare la frequenza di ciascuna classe di voto (da A a F). La variabile in questione mostra uno sbilanciamento significativo nella distribuzione dei voti, come si nota in Figura 6.

Questo rappresenta una potenziale criticità per i modelli di classificazione, che potrebbero essere influenzati dalla mancanza di esempi sufficienti per le classi meno rappresentate.

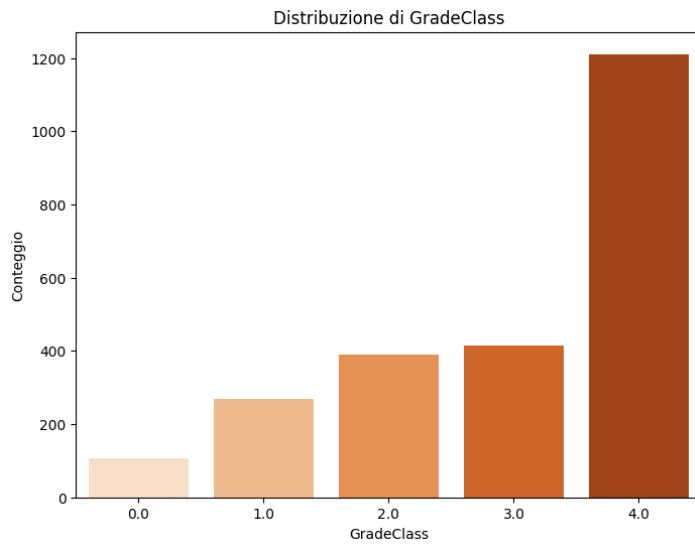


Figure 6: Distribuzione delle classi di voto (*GradeClass*).

#### 2.2.6 Distribuzione delle assenze

La variabile *Absences* è stata analizzata tramite un istogramma per osservare la distribuzione delle assenze tra gli studenti. È possibile notare in Figura 7 come la distribuzione media delle assenze sia abbastanza equilibrata.

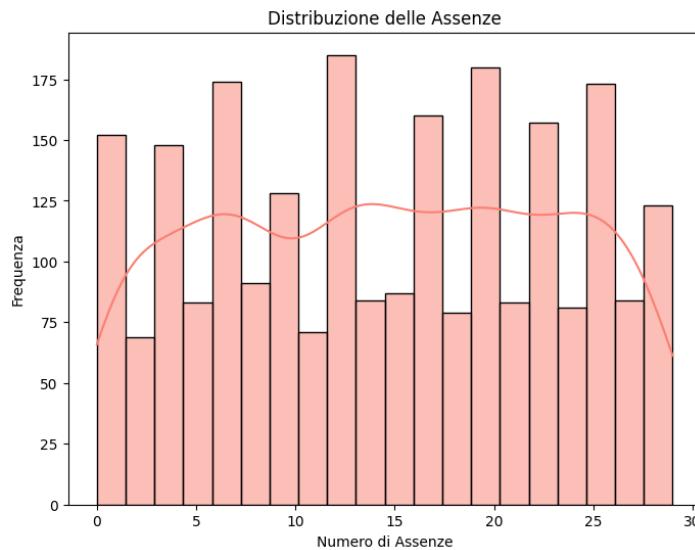


Figure 7: Distribuzione delle assenze degli studenti.

### 2.2.7 Distribuzione dell'educazione dei genitori

Un'altra analisi eseguita è relativa al rapporto tra il livello di educazione dei genitori e le performance scolastiche degli studenti. È stata utilizzata una visualizzazione a barre, visibile in Figura 8, che però non ha mostrato un impatto significativo sul voto degli studenti, mostrando una distribuzione piuttosto bilanciata.

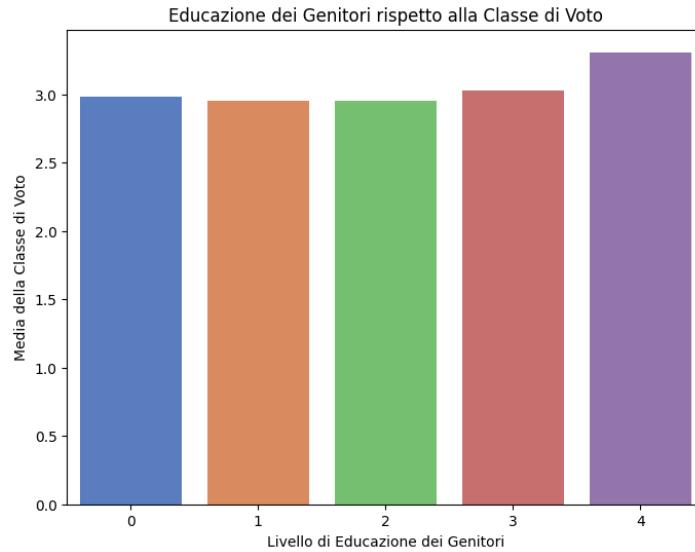


Figure 8: Distribuzione dell'educazione dei genitori rispetto alla classe di voto.

### 2.2.8 Matrice di correlazione

È stata calcolata, infine, una matrice di correlazione per le principali variabili numeriche e categoriali codificate. La matrice è stata visualizzata tramite una heatmap, evidenziando le correlazioni positive e negative tra le variabili. Ad esempio è stata evidenziata la forte relazione tra il numero delle assenze e il GradeClass, ma anche delle lievi relazioni tra il tempo di studio e il GradeClass, e tra il parentalSupport ed il GradeClass, tutto visualizzabile in Figura 9.

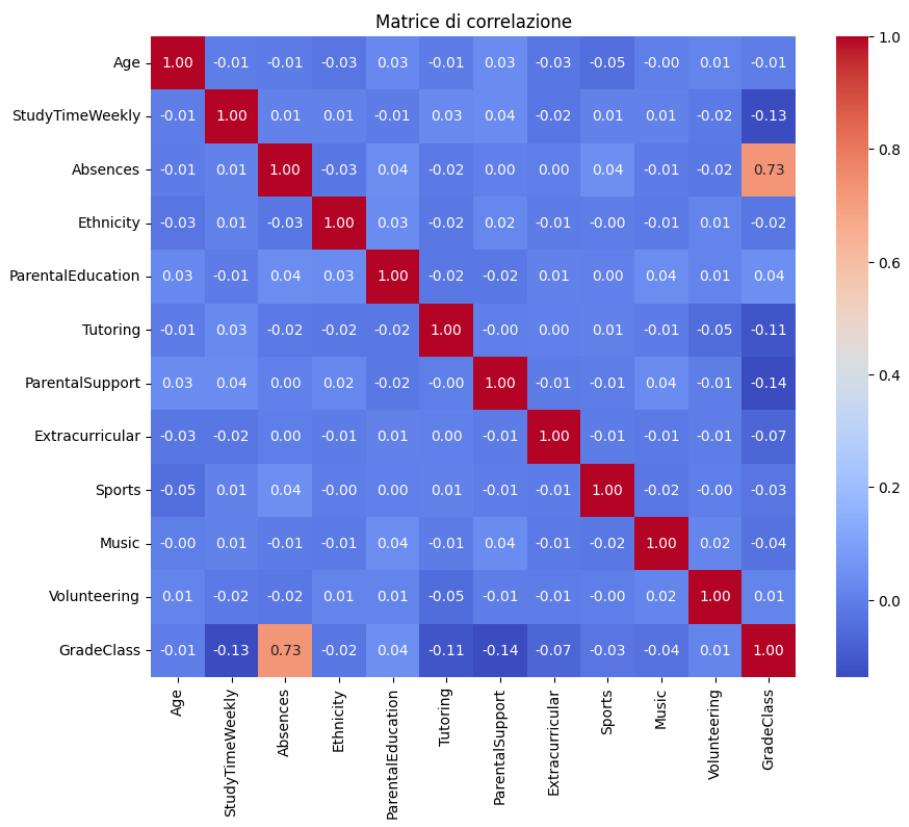


Figure 9: Matrice di correlazione tra le variabili numeriche e codificate.

### 3 Student performance analysis: Classificazione

La classificazione è stata condotta con l'obiettivo di sviluppare modelli predittivi in grado di stimare la classe di voto degli studenti sulla base degli altri attributi presenti nel dataset. Il processo ha incluso un'analisi esplorativa approfondita, la selezione delle caratteristiche più rilevanti, l'applicazione di diversi algoritmi di machine learning per identificare il modello con le migliori prestazioni e, infine, l'integrazione di modelli per migliorare ulteriormente l'accuratezza e la robustezza della classificazione.

#### 3.1 Preprocessing del Dataset

Per affrontare il problema dello sbilanciamento delle classi nella variabile target *GradeClass* (vedi Figura 10), è stata adottata una combinazione di tecniche di oversampling e undersampling. L'obiettivo era bilanciare il numero di campioni per ciascuna classe, riportando tutti i campioni a 500 elementi. Questa procedura è stata applicata esclusivamente al dataset di training, preservando il dataset di test per una valutazione non influenzata.

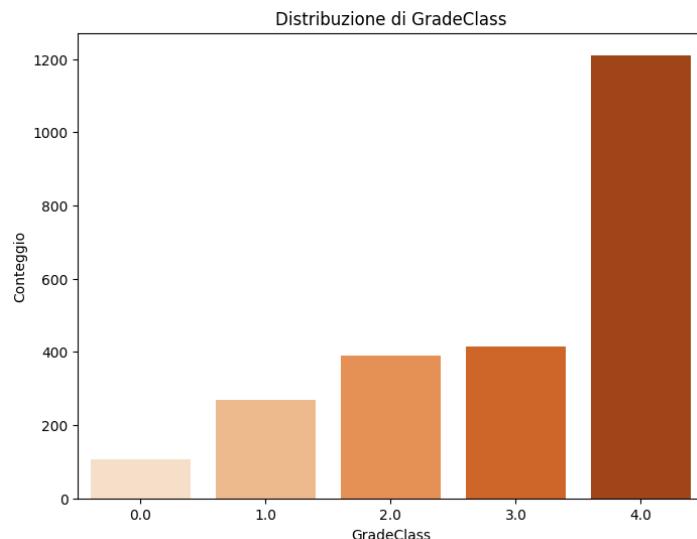


Figure 10: Distribuzione di GradeClass prima di applicare le tecniche di sampling

##### 3.1.1 Oversampling con SMOTE

Per aumentare il numero di campioni delle classi minoritarie (*GradeClass* 0, 1, 2 e 3), è stata utilizzata la tecnica *SMOTE* (*Synthetic Minority Oversampling Technique*). Questa tecnica genera nuovi campioni sintetici interpolando tra i campioni esistenti,

mantenendo inalterata la distribuzione delle feature. I risultati dell'oversampling sono riportati di seguito:

- **Classe 0:** 393 campioni sintetici e 107 campioni originali.
- **Classe 1:** 232 campioni sintetici e 268 campioni originali.
- **Classe 2:** 109 campioni sintetici e 391 campioni originali.
- **Classe 3:** 86 campioni sintetici e 414 campioni originali.

### 3.1.2 Undersampling con Clusterizzazione

Per ridurre il numero di campioni della classe maggioritaria (*Grade Class 4*) da 1211 a 500, è stata utilizzata la tecnica di *undersampling basata sulla clusterizzazione* (*ClusterCentroids*). Questo metodo identifica i cluster rappresentativi dei dati originali e seleziona i campioni centrali di ciascun cluster, mantenendo intatta la distribuzione delle feature.

### 3.1.3 Dataset Finale

Dopo l'applicazione di entrambe le tecniche, il dataset di training bilanciato presenta esattamente 500 campioni per ciascuna classe. Questo dataset è stato salvato in un file CSV per l'addestramento dei modelli, insieme al dataset di test non bilanciato. In Figura 11 è possibile vedere il risultato della distribuzione dei voti dopo il processo di bilanciamento.

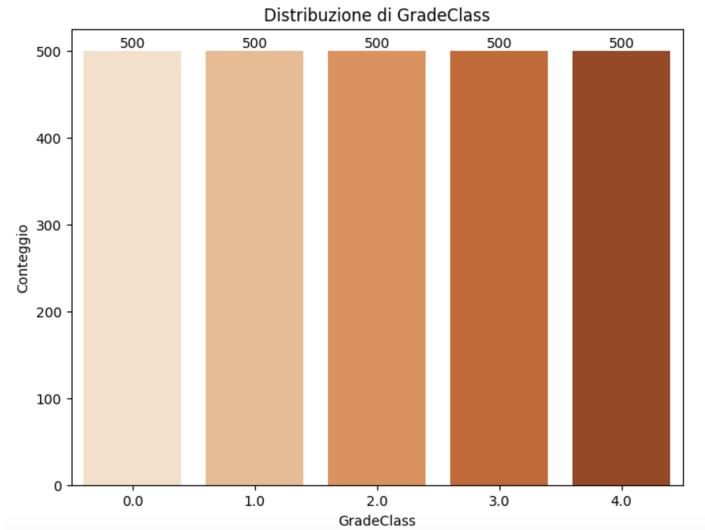


Figure 11: Distribuzione di GradeClass dopo il sampling

### 3.1.4 Analisi della distribuzione degli attributi

Al termine delle operazioni di oversampling e undersampling, è stata verificata la coerenza della distribuzione degli attributi nel dataset bilanciato. Per questo scopo è stata calcolata e visualizzata la matrice di correlazione delle feature. La matrice di correlazione fornisce una rappresentazione delle relazioni lineari tra le variabili, evidenziando eventuali cambiamenti nella struttura dei dati causati dalle operazioni di bilanciamento.

In Figura 12 è possibile vedere la matrice ottenuta dopo le operazioni di sampling.

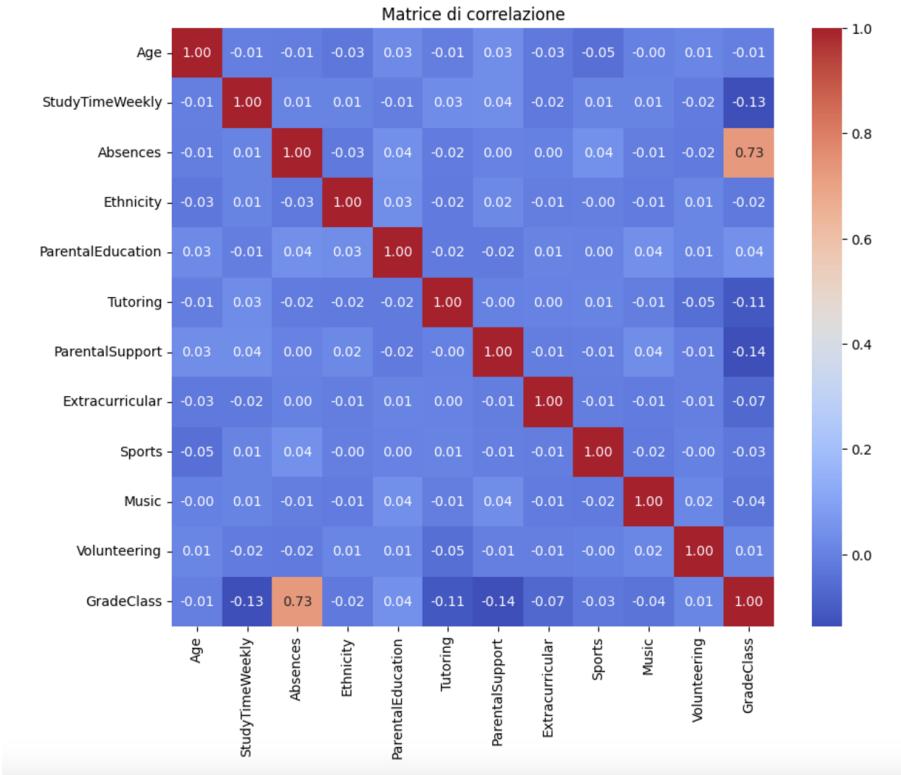


Figure 12: Matrice di correlazione dopo le operazioni di sampling

L'analisi della matrice di correlazione ha confermato che la distribuzione degli attributi nel dataset bilanciato è rimasta coerente con quella originale mostrata in Figura 9. Non sono emerse correlazioni spuriamente elevate o relazioni alterate, a conferma che il bilanciamento delle classi non ha compromesso l'integrità del dataset.

## 3.2 Scenari di Classificazione

Nel procedere con la classificazione della variabile *GradeClass*, sono stati considerati due scenari differenti, al fine di valutare l'impatto degli attributi sul modello e migliorare la capacità predittiva. Di seguito sono descritti i due scenari:

1. **Classificazione escludendo *GPA*:** L'attributo *GPA* rappresenta il valore continuo del voto, dal quale è direttamente derivata la variabile target *GradeClass*. Poiché *GPA* costituisce una fonte altamente predittiva ma ridondante di informazione, mantenere questa variabile nel dataset durante il task di classificazione della *GradeClass* è stato considerato inappropriato. La sua presenza, infatti, potrebbe introdurre un bias nei modelli predittivi, compromettendo l'integrità e l'utilità dell'analisi.
2. **Classificazione escludendo *Absences* (in aggiunta al già rimosso *GPA*):** Dall'analisi della matrice di correlazione (Figura 12), l'attributo *Absences* è risultato essere altamente correlato con *GradeClass*. Per verificare se questa correlazione fosse un fattore determinante per il modello, è stato escluso in questo scenario, al fine di analizzare l'impatto della sua assenza sulle prestazioni predittive.

### 3.3 Processo di Classificazione

Il processo di classificazione è stato applicato ad entrambi gli scenari descritti in precedenza, seguendo un approccio uniforme per garantire un confronto equo. I passi principali includono la selezione di modelli, l'ottimizzazione dei parametri e l'integrazione dei modelli attraverso tecniche di ensemble learning.

Gli algoritmi di machine learning che sono stati scelti per la classificazione sono:

- **Random Forest**
- **K-Nearest Neighbors**
- **Decision Tree**
- **Naive Bayes**
- **AdaBoost Classifier**
- **Gradient Boosting Classifier**

Si è proceduto addestrando i modelli con parametri di default, in modo da poter analizzare i miglioramenti con approcci più sofisticati. I risultati ottenuti negli scenari sono i seguenti<sup>1</sup>:

---

<sup>1</sup>I risultati con metriche più dettagliate sono presenti nel codice reperibile nella repository GitHub

- Classificazione escludendo *GPA*:

Index	Model	Mean Accuracy
0	Random Forest	0.6504
1	K-Nearest Neighbors	0.4303
2	Decision Tree	0.5473
3	Naive Bayes	0.4721
4	AdaBoost Classifier	0.5974
5	Gradient Boosting Classifier	0.6587

Table 2: Confronto delle performance dei modelli relativi al caso senza GPA: accuratezza media calcolata sul test set

- Classificazione escludendo *Absences* e *GPA*:

Index	Model	Mean Accuracy
0	Random Forest	0.2214
1	K-Nearest Neighbors	0.1671
2	Decision Tree	0.2214
3	Naive Bayes	0.2159
4	AdaBoost Classifier	0.2354
5	Gradient Boosting Classifier	0.2479

Table 3: Confronto delle performance dei modelli escludendo GPA e Absenses: accuratezza media (*Mean Accuracy*) calcolata sul test set

### 3.3.1 Ottimizzazione dei Modelli

Per migliorare le prestazioni, è stata applicata la *grid search* sui modelli per ottimizzarne i parametri. L'approccio Grid Search consiste nel testare diverse combinazioni di parametri per ogni modello, al fine di individuare quello migliore. In particolare, per individuare le migliori performance, si fa uso di una tecnica di valutazione avanzata: la *5-fold cross-validation*.

I migliori parametri individuati per il primo scenario (escludendo GPA) sono:

- **Random Forest**: `max_depth=30, n_estimators=200`.
- **KNN**: `metric='minkowski', n_neighbors=11, p=1, weights='distance'`.
- **Decision Tree**: Parametri default.
- **Naive Bayes**: `var_smoothing=1e-09`.
- **AdaBoost Classifier**: `n_estimators=50`, con base learner `DecisionTreeClassifier(max_depth=3)`.
- **Gradient Boosting Classifier**: Parametri default con `random_state=42`.

I risultati ottenuti in seguito al raffinamento dei parametri sono riportati in Tabella 4:

<b>Index</b>	<b>Model</b>	<b>Mean Accuracy</b>
0	Random Forest	0.6532
1	K-Nearest Neighbors	0.4721
2	Decision Tree	0.5584
3	Naive Bayes	0.4721
4	AdaBoost Classifier	0.5974
5	Gradient Boosting Classifier	0.6587

Table 4: Confronto delle performance dei modelli relativi al caso senza GPA: accuratezza media calcolata sul test set

Mentre nel secondo scenario (escludendo GPA e Absenses), i parametri migliori risultano essere:

- **Random Forest:** `max_depth = None, n_estimators=300.`
- **KNN:** `metric='minkowski', n_neighbors=3, p=1, weights='distance'.`
- **Decision Tree:** Parametri default.
- **Naive Bayes:** `var_smoothing=1e-09.`
- **AdaBoost Classifier:** `n_estimators=200`, con base learner `DecisionTreeClassifier(max_depth=3)`.
- **Gradient Boosting Classifier:** Parametri default con `random_state=42`.

I risultati ottenuti in seguito al raffinamento dei parametri sono riportati in Tabella 5:

<b>Index</b>	<b>Model</b>	<b>Mean Accuracy</b>
0	Random Forest	0.2353
1	K-Nearest Neighbors	0.1894
2	Decision Tree	0.2228
3	Naive Bayes	0.2158
4	AdaBoost Classifier	0.2089
5	Gradient Boosting Classifier	0.2465

Table 5: Confronto delle performance dei modelli escludendo GPA e Absenses: accuratezza media (*Mean Accuracy*) calcolata sul test set

Analizzando i risultati abbiamo notato che, l'assenza sia dell'attributo GPA che dell'attributo Absenses, porta a risultati poco ottimali. Pertanto nella prosecuzione delle analisi, si è deciso di considerare solamente il caso riconducibile al primo scenario, ossia quello privato della sola variabile GPA.

Nella sezione successiva viene presentato un approccio avanzato alla classificazione per cercare di migliorare i risultati ottenuti e mostrati precedentemente.

### 3.3.2 Ensemble Learning

Gli approcci di *Ensemble Learning* adottati sono due:

**Voting Classifier** Un *Voting Classifier* è un ensemble di modelli di apprendimento automatico che combina le previsioni di diversi classificatori per migliorare la performance rispetto ai singoli modelli. In questo caso, abbiamo implementato il Voting Classifier utilizzando il metodo di soft voting, che sfrutta le probabilità predette dai singoli modelli per determinare la classe finale. La previsione di ciascun modello è ponderata in

base alla sua probabilità di appartenenza alle diverse classi, e il modello finale predice la classe con la probabilità media più alta.

I modelli integrati per la realizzazione del voting classifier sono quelli esposti in precedenza con i parametri ottimizzati ottenuti mediante grid search. Le prestazioni sono state valutate tramite metriche standard, e i risultati sono stati visualizzati attraverso:

- **Classification Report:** Precisione, recall, F1-score e supporto (vedi Figura 13).

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	33
1.0	0.33	0.03	0.05	80
2.0	0.00	0.00	0.00	121
3.0	0.08	0.01	0.01	127
4.0	0.51	0.99	0.68	357
accuracy			0.50	718
macro avg	0.18	0.20	0.15	718
weighted avg	0.31	0.50	0.34	718

Figure 13: Classification report

- **Confusion Matrix:** Visualizzata come *heatmap* (riportata in Figura 14).

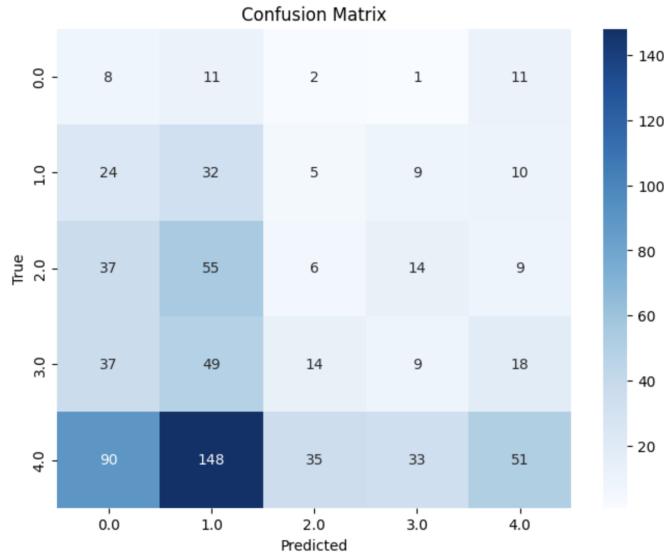


Figure 14: Matrice di confusione Voting Classifier

La matrice di confusione mostra molti problemi nella classificazione, si può vedere come il modello non riesca a predire correttamente le classi di voto.

- **Curva Precision-Recall:** Per valutare le prestazioni per ciascuna classe (vedi Figura 15).

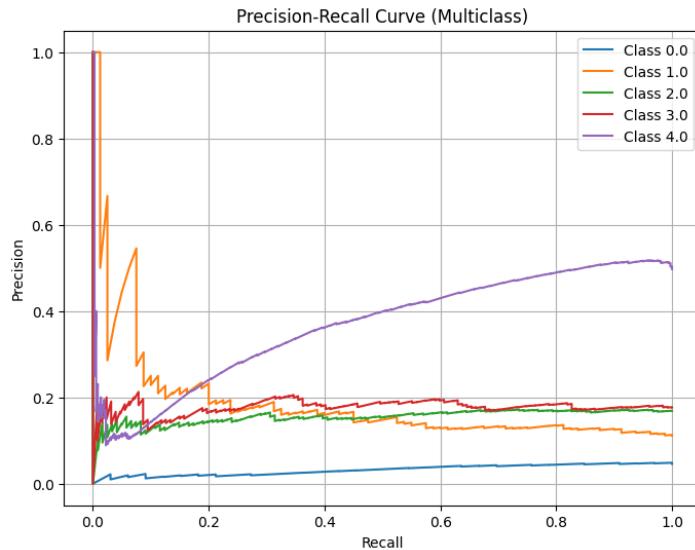


Figure 15: Curva Precision-Recall

Il grafico evidenzia la precisione rispetto al recall per ciascuna classe. I valori di precisione sono generalmente bassi per la maggior parte delle classi, il che indica che il modello spesso classifica erroneamente. Alcune classi (come la classe 4) mostrano una crescita graduale della precisione con l'aumentare del recall, ma non raggiungono mai valori particolarmente alti.

- **Curva ROC:** Per rappresentare graficamente la capacità del modello di distinguere tra le classi (Figura 16).

Le AUC per le varie classi variano notevolmente. La classe 1 ha l'AUC più alta (0.60), il che indica una discreta capacità di discriminazione. Le classi 0, 2, 3 e 4 hanno AUC inferiori a 0.51, con la classe 4 che ha il valore peggiore (0.29), suggerendo che il modello ha difficoltà a distinguere queste classi dagli altri. Il modello sembra avere prestazioni subottimali per molte classi, segnalando possibili problemi di separabilità nei dati.

- **Learning Curve:** Per visualizzare l'andamento dell'addestramento su training e test set (vedi Figura 17).

Il grafico mostra la differenza tra l'accuratezza su training e test al variare del numero di campioni di addestramento. L'accuratezza sul training è costantemente a 1.0, mentre l'accuratezza sul test migliora all'aumentare dei campioni, ma rimane relativamente bassa.

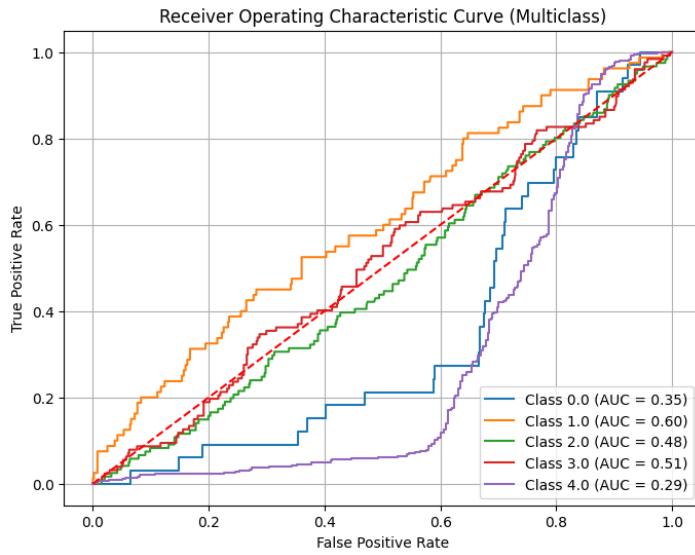


Figure 16: Curva ROC

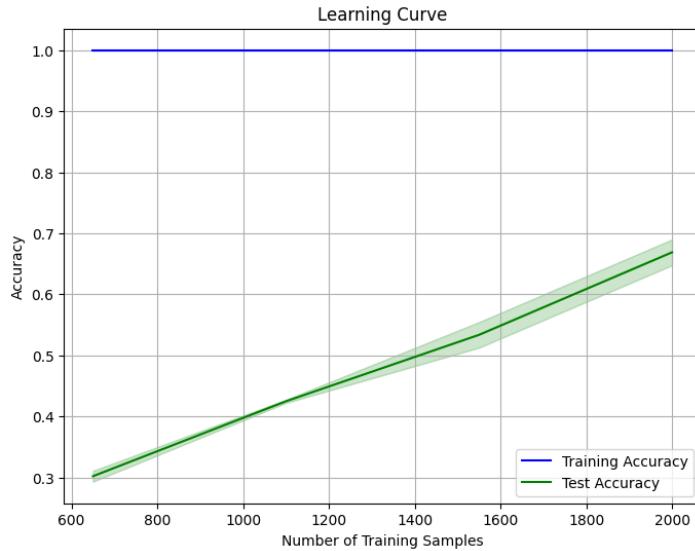


Figure 17: Learning curve

**Stacking Classifier** Un *Stacking Classifier* è una tecnica di ensemble che combina le previsioni di diversi modelli base (classificatori di base) utilizzando un meta-modello per effettuare la previsione finale. A differenza del Voting Classifier, che aggrega le previsioni in modo semplice (ad esempio con soft voting), lo stacking sfrutta un processo di apprendimento secondario per migliorare le performance complessive. In questo caso, è stato implementato un Stacking Classifier che combina gli stessi modelli di base utiliz-

zati nel Voting Classifier. Tuttavia, anziché fare una semplice media delle probabilità, le predizioni dei modelli base sono state utilizzate come input per un meta-modello (in questo caso una regressione logistica). Questo meta-modello apprende come combinare in modo ottimale le predizioni dei modelli base, cercando di ridurre gli errori e migliorare la performance finale.

Le prestazioni sono state valutate tramite metriche standard, e i risultati sono stati visualizzati attraverso:

- **Classification Report:** Precisione, recall, F1-score e supporto (vedi Figura 18).

	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
0.0	0.00	0.00	0.00	33
1.0	0.33	0.03	0.05	80
2.0	0.00	0.00	0.00	121
3.0	0.08	0.01	0.01	127
4.0	0.51	0.99	0.68	357
<b>accuracy</b>				0.50
<b>macro avg</b>	<b>0.18</b>	<b>0.20</b>	<b>0.15</b>	<b>718</b>
<b>weighted avg</b>	<b>0.31</b>	<b>0.50</b>	<b>0.34</b>	<b>718</b>

Figure 18: Classification report Stacking Classifier con Logistic Regression

- **Confusion Matrix:** Visualizzata come *heatmap* (Figura 19).

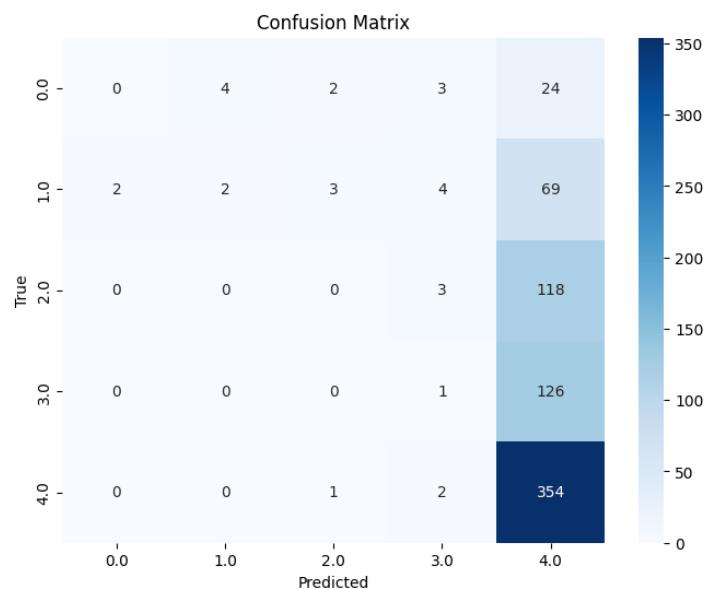


Figure 19: Matrice di confusione relativa allo Stacking Classifier con Logistic Regression

La matrice mostra una forte concentrazione di campioni nella classe 4 (357 casi), con numeri decrescenti nelle classi superiori. La matrice evidenzia come il modello non riesce a distinguere correttamente i dati e tende a predire con grande frequenza la classe 4.

### **3.3.3 Commento ai risultati**

I risultati ottenuti sono purtroppo deludenti e non rispecchiano le aspettative, nonostante l'adozione di tecniche avanzate come il Voting Classifier e il Stacking Classifier. La bassa performance potrebbe essere attribuita a diversi fattori, tra cui la qualità del dataset utilizzato e la possibile insufficienza delle feature per stimare correttamente il voto. Le variabili a disposizione potrebbero non essere significative o sufficientemente informative per il compito di classificazione, limitando così la capacità dei modelli di apprendere correttamente dai dati.

## 4 Student performance analysis: Clustering

Questo capitolo ha lo scopo di illustrare alcune tecniche di clusterizzazione con lo scopo di suddividere il dataset in un insieme di gruppi o cluster con caratteristiche simili. Quali siano queste caratteristiche e/o quante siano, dipende dalla specifica tecnica utilizzata e dal tuning degli iperparametri dell'algoritmo. In generale, le tecniche di clusterizzazione si suddividono in:

- **Clusterizzazione per partizione.** Suddivide i dati in  $k$  cluster, ottimizzando una funzione obiettivo (ad esempio, minimizzando la distanza intra-cluster cercando di massimizzare la distanza inter-cluster).
- **Clusterizzazione gerarchica.** Crea una struttura gerarchica dei cluster (un albero o dendrogramma) combinando o dividendo iterativamente i cluster.
- **Clusterizzazione basata su densità.** Identifica cluster come regioni dense nello spazio dati, separandole dalle aree a bassa densità.
- **Clusterizzazione basata su modello.** Assume che i dati siano generati da un modello statistico (ad esempio, una combinazione di distribuzioni gaussiane) e cerca di stimare i parametri del modello.
- **Clusterizzazione fuzzy.** Permette ai punti dati di appartenere a più cluster contemporaneamente, assegnando un grado di appartenenza (fuzzy membership).
- **Clusterizzazione basata su grafi.** Modella i dati come un grafo, dove i nodi rappresentano punti dati e gli spigoli rappresentano le somiglianze; i cluster sono identificati come sottogruppi con spigoli densi.
- **Clusterizzazione basata su griglie.** Suddivide lo spazio in una griglia finita di celle e identifica i cluster in base alla densità delle celle.
- **Clusterizzazione evolutiva.** Utilizza algoritmi evolutivi o tecniche di ottimizzazione per identificare cluster. È particolarmente utile in spazi di ricerca complessi.
- **Clusterizzazione temporale/spazio-temporale.** Analizza dati con componenti temporali o spaziotemporali per identificare pattern che si evolvono nel tempo.

Gli algoritmi esplorati nell'elaborato ricadono nelle tecniche basate su partizionamento e densità.

## 4.1 Fase di ETL per il clustering

### 4.1.1 Operazioni preliminari sul dataset

Il dataset, descritto accuratamente nel capitolo 2, ha subito alcuni passaggi fondamentali per ”agevolare” e rendere più efficace la fase di clusterizzazione. Gli attributi *StudentID* e *GradeClass* sono stati eliminati poichè non sono utili ai fini delle tecniche di clusterizzazione da noi utilizzate.

Come è possibile osservare dalla matrice di correlazione (2.2.8), *GradeClass* è un attributo ricavato da *GPA*, per questo motivo è inutile tenerlo in considerazione per il clustering, non farebbe altro che aumentare il problema della *Course of dimensionality*<sup>2</sup>. L’attributo *StudentId* non è utile nella clusterizzazione perché rappresenta semplicemente un identificatore univoco assegnato agli studenti, senza fornire alcuna informazione significativa sulle caratteristiche o sulla similarità tra gli studenti. Complessivamente le feature del dataset di partenza, descritte al punto 2, si riducono a:

1. **Age.**
2. **Gender.**
3. **Ethnicity.**
4. **ParentalEducation.**
5. **StudyTimeWeekly.**
6. **Absences.**
7. **Tutoring.**
8. **ParentalSupport.**
9. **Extracurricular.**
10. **Sports.**
11. **Music.**
12. **Volunteering.**
13. **GPA.**

---

<sup>2</sup>La curse of dimensionality nel machine learning si verifica quando l’aumento del numero di caratteristiche rende i dati più sparsi, aumentando la complessità dello spazio e causando difficoltà per gli algoritmi nel trovare pattern significativi.

Le 13 features rimanenti sono state successivamente standardizzate con lo **StandardScaler**. La standardizzazione è una trasformazione che ridimensiona le caratteristiche in modo che abbiano una media pari a 0 e una deviazione standard pari a 1:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

L'equazione 1, rappresenta il processo di standardizzazione applicato per ogni feature  $z$ , in cui  $x$  è il valore di una feature, mentre  $\mu$  e  $\sigma$  sono rispettivamente la media e la varianza della feature in esame.

La standardizzazione è un passo fondamentale per l'applicazione delle tecniche illustrate al punto 4, in quanto scale diverse tra le feature potrebbero influenzare e distorcere i risultati.

#### 4.1.2 Riduzione di dimensionalità

Pur effettuando le operazioni riportate al punto precedente, le dimensioni rimanenti non permettono una chiara visualizzazione del risultato, ed inoltre contribuiscono al problema della *Course of dimensionality*.

La soluzione proposta consiste nell'utilizzare algoritmi di *feature extraction* per generare, a partire dalle feature esistenti, un numero ridotto di nuove feature.

Dalla Figura 20 si può osservare che solo alcune variabili presentano una varianza significativa. In particolare, una grande variazione si riscontra nelle variabili *StudyTime Weekly* e *Absences*. Le altre variabili, invece, tendono ad assumere valori discreti compresi in un range tra 0 e 4, il che comporta una variabilità media molto bassa (meno del 5%).

Il metodo di *feature-extraction* maggiormente utilizzato per diminuire il problema della *Course of dimensionality* è la *Principal Component Analysis (PCA)*.

La *PCA* è una tecnica di riduzione della dimensionalità utilizzata per trasformare un dataset con molte variabili in un insieme più piccolo di variabili, dette componenti principali, che catturano la maggior parte della varianza dei dati originali. La Figura 21 mostra che per raccogliere il 70% di varianza cumulativa<sup>3</sup> occorrono 8 componenti, mentre scendiamo a 6 componenti se ci si accontenta di catturare il 50% della varianza cumulativa.

Siccome l'obiettivo è quello di, contemporaneamente, ridurre il problema della *course of dimensionality* e rendere visualizzabili i risultati, allora si è optato per l'utilizzo di un'altra tecnica, ossia la *t-SNE*.

La tecnica *t-SNE* (*t-Distributed Stochastic Neighbor Embedding*) è un metodo di riduzione della dimensionalità sviluppato per visualizzare dati complessi in spazi a due

---

<sup>3</sup>la varianza cumulativa rappresenta la quantità totale di varianza spiegata dalle prime  $K$  componenti principali considerate. È una misura che aiuta a determinare quante componenti principali sono necessarie per catturare una quota significativa della varianza originale dei dati.

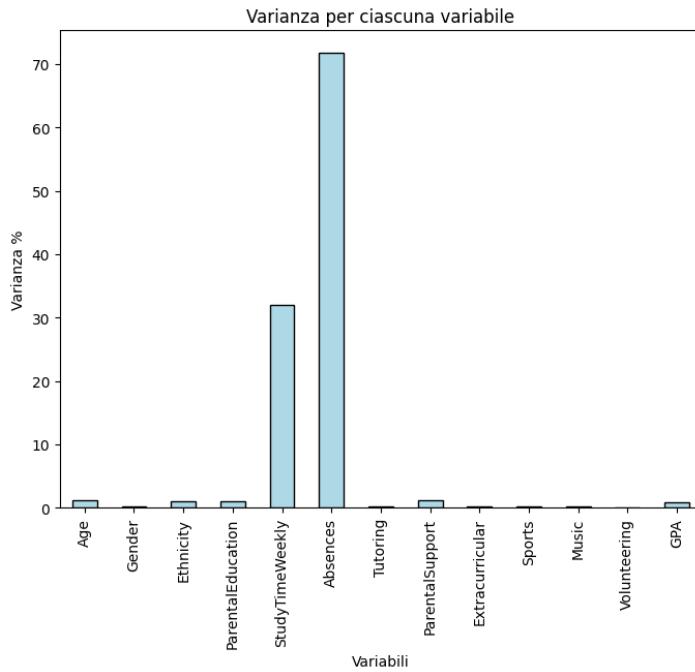


Figure 20: Varianza delle variabili del dataset.

o tre dimensioni. Si basa su un principio elementare; preservare le relazioni locali dei dati in uno spazio più semplice. In Python attraverso la libreria `scikit-learn`, possiamo utilizzare il metodo predefinito `TSNE`, al quale vengono dati in input i seguenti parametri:

- **n\_components.** Specifica il numero di dimensioni nello spazio di output (tipicamente 2 o 3 per la visualizzazione).
- **Random\_state.** Controlla il generatore di numeri casuali usato per l'inizializzazione. Abbiamo utilizzato il valore 42. Specificare il valore è importante per rendere questa operazione riproducibile.
- **Perplexity.** Determina il bilanciamento tra l'attenzione ai vicini locali e globali. È una sorta di misura del numero di vicini da considerare per ogni punto. Un valore troppo basso può causare un'eccessiva frammentazione, mentre uno troppo alto potrebbe perdere dettagli locali. Solitamente i valori oscillano tra 30 e 50 e il tutto dipende dalla dimensione del dataset in questione. Per le analisi dell'elaborato è stato utilizzato un valore di 40.

La Figura 22 mostra il risultato dell'applicazione della tecnica *T-sne* sul dataset di partenza, già parzialmente trasformato al punto 4.1.1. E' possibile notare come le dimensioni siano passate da 13 a 2 ed una distribuzione di punti dello spazio che tende a conservare le relazioni locali.

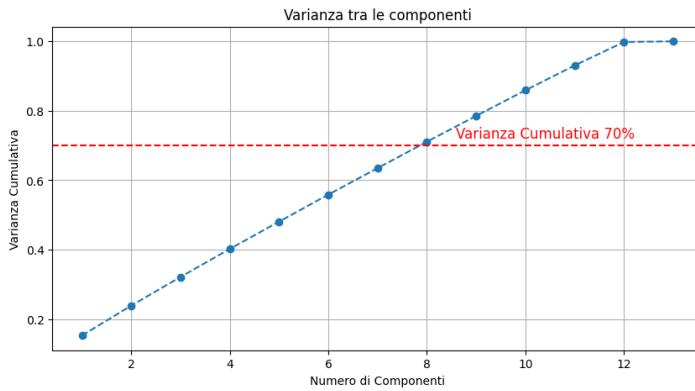


Figure 21: Varianza cumulativa all'aumentare del numero di componenti

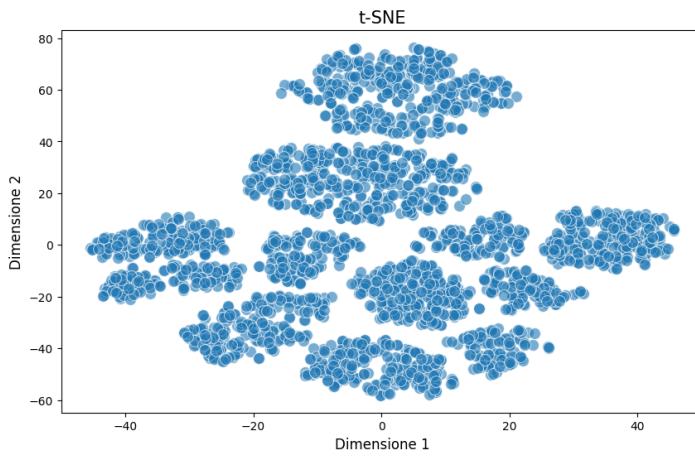


Figure 22: Visualizzazione dataset dopo l'applicazione *T-sne*

## 4.2 Kmeans

Il metodo *K-means* è una tecnica di clustering per partizionamento molto popolare, ampiamente utilizzata su diversi dataset per affrontare una varietà di compiti e analisi. La caratteristica principale di questo algoritmo è la necessità di specificare un unico parametro  $K$ , ovvero il numero di cluster. L'obiettivo principale del K-means è minimizzare la distanza intra-cluster (la distanza tra ciascun punto e il centroide del cluster, definito come il valore medio dei punti del cluster) e massimizzare la distanza inter-cluster (la distanza tra i centroidi dei diversi cluster).

La complessità computazionale dell'algoritmo dipende sia dalla dimensionalità che dalla grandezza del dataset su cui è applicato. Sebbene il metodo sia relativamente semplice, la sua scalabilità è limitata in presenza di dataset particolarmente ampi o complessi, rendendolo meno adatto per problemi che richiedono l'elaborazione di grandi volumi di

dati.

#### 4.2.1 Applicazione dell'algoritmo K-means

Come specificato all'inizio del paragrafo 4.2, occorre selezionare opportunamente il parametro  $K$ , un valore troppo basso o troppo alto potrebbe portare rispettivamente a problemi di underfitting ed overfitting.

Il *metodo di Elbow(gomito)* è un metodo empirico per scegliere il valore migliore di  $k$ , ovvero il corretto numero di cluster. Nel nostro caso vengono creati 19 modelli di k-means, con valori di  $k$  che variano da 1 a 19. Per ciascun modello viene calcolato il valore di *WCSS(Within Cluster Sum of Squares)*<sup>4</sup>.

Il metodo è stato opportunamente modificato per ampliarne la sua efficacia, considerando ulteriori parametri in aggiunta a quelli basilari:

- **max\_iter:** Corrisponde al numero di iterazioni, ovvero di calcoli di nuovi centroide, che al più possono essere fatti per ciascun modello creato. Il valore è stato impostato a 30000.
- **n\_init:** Corrisponde al numero di inizializzazioni dei centroidi prima di applicare il vero e proprio algoritmo del k-means. Fare più inizializzazioni significa avere maggior probabilità di ottenere una configurazione che ha buon valori per quanto riguarda la distanza intra-cluster e inter-cluster. Il valore assegnato è 3000.

Tali valori rappresentano il risultato di un trade-off tra efficacia di clusterizzazione ed efficienza computazionale per ciascun modello del metodo di Elbow.

Il valore di  $k$  si sceglie in corrispondenza del gomito della funzione di Elbow, perché rappresenta il punto in cui l'aggiunta di nuovi cluster non porta più a una significativa riduzione della varianza intra-cluster (WCSS). Questo punto indica un buon equilibrio tra la qualità del clustering e la semplicità del modello, evitando sia cluster troppo generici che sovraccaricati. Come è possibile evidenziare in Figura 23, un possibile valore ottimale si otterrebbe scegliendo un valore di  $K$  pari a 5.

Il modello risultante è quello rappresentato in Figura 24, con l'applicazione del metodo K-Means con valore di  $K$  impostato a 5.

#### 4.2.2 Analisi dei risultati ottenuti

I cluster ottenuti sono:

- **0.** Composto da 485 studenti.

---

<sup>4</sup>Il WCSS (Within-Cluster Sum of Squares) si calcola sommando le distanze quadratiche tra ogni punto di un cluster e il suo centroide, ripetendo il calcolo per tutti i cluster. Serve a misurare la compattezza interna dei cluster: valori più bassi indicano cluster meglio definiti. È utilizzato nel metodo del gomito per scegliere il numero ottimale di cluster  $k$ .

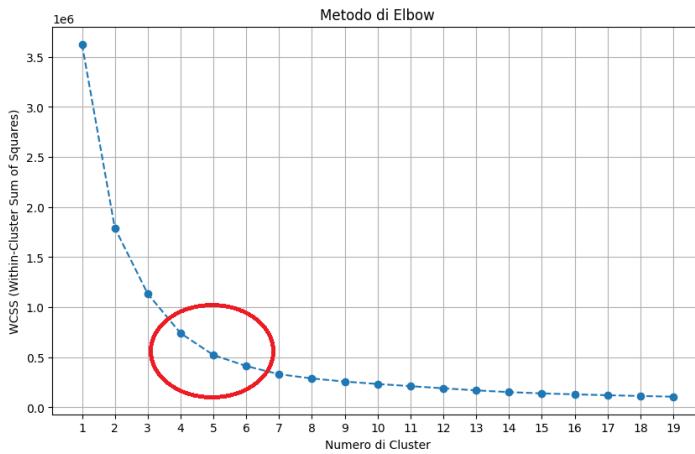


Figure 23: Metodo di Elbow per la scelta del parametro  $K$ .

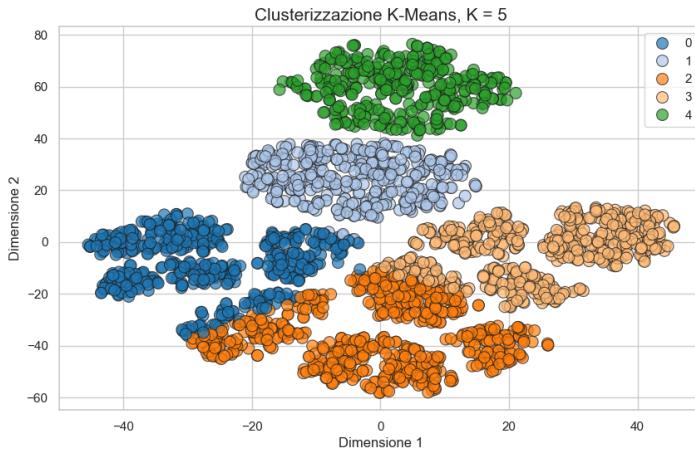


Figure 24: K-means con  $K = 5$ .

- 1. Composto da 396 studenti.
- 2. Composto da 598 studenti.
- 3. Composto da 537 studenti.
- 4. Composto da 376 studenti.

Dalla Figura 25 è possibile osservare come il cluster 0 sia composto da persone con voti mediamente più alti, mentre il cluster 2 è composto da persone che ottengono voti mediamente bassi. Inoltre, sempre dal grafico in Figura 25, è possibile notare come la varianza della variabile *GPA* nei diversi cluster sia molto simile.

Dalla Figura 26 è possibile osservare come il cluster 0, corrispondente agli studenti

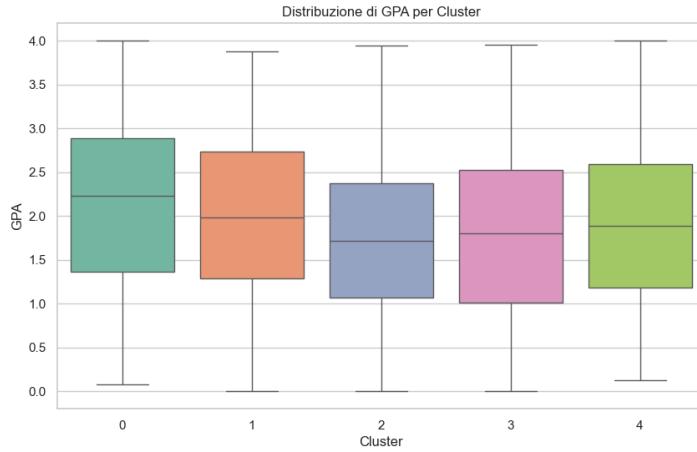


Figure 25: Distribuzione dei valori *GPA* nei cluster del *K-means*.

mediamente più bravi, è composto da un numero di donne leggermente superiore rispetto al numero di uomini. Viceversa, il cluster 2, corrispondente al gruppo di studenti mediamente meno bravi, è composto prevalentemente da studenti di sesso maschile. Da ciò non è possibile affermare che le donne siano sempre più brave degli uomini in ambito scolastico poiché, come dimostra il cluster 3, il quale ottiene un valore di GPA molto basso, esso risulta composto prevalentemente da donne. Nel nostro caso, quindi, la "bravura" non è determinata dal sesso dei singoli studenti.

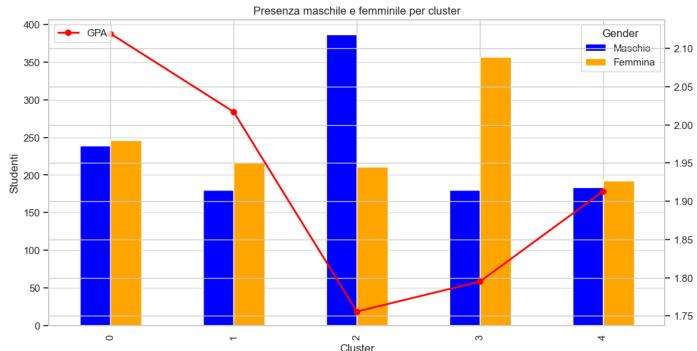


Figure 26: Presenza maschile/femminile e GPA per cluster.

Dalla Figura 27 è interessante osservare come i componenti del cluster 0, dedichino allo studio una quantità di ore settimanali minore rispetto a tutti gli altri componenti in altri cluster, andando contro una comune credenza secondo cui per ottenere voti più alti occorre dedicare maggior tempo allo studio.

I risultati precedenti rappresentano medie calcolate sui valori dei cluster e, pertanto, dipendono sia dalla variabilità all'interno di ciascun cluster sia dal numero di studenti

per cluster.

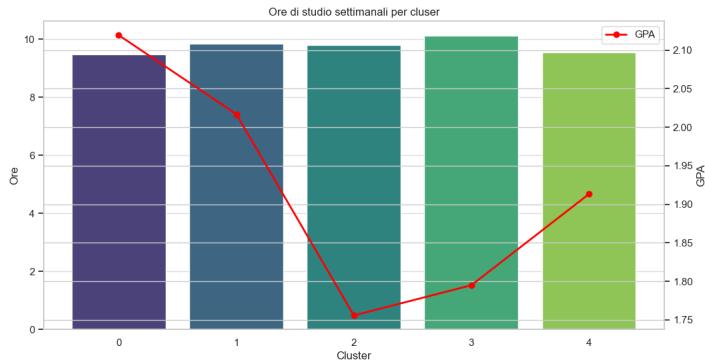


Figure 27: Ore di studio medie per cluster.

Per testare la qualità del modello, utilizziamo la metrica non supervisionata chiamata Silhouette<sup>5</sup>. La Figura 28 mostra un valore medio di circa 0.45. La maggior parte dei punti di ciascun cluster hanno un valore compreso tra 0.01 e 0.4, il che indica una distanza inter cluster non elevata, ma ciò era da aspettarselo dato che come metodo di riduzione della dimensionalità è stato utilizzato la tecnica *T-sne* che, a partire da punti vicini nello spazio originale, assicura che questi punti restino vicini anche nello spazio ridimensionato.

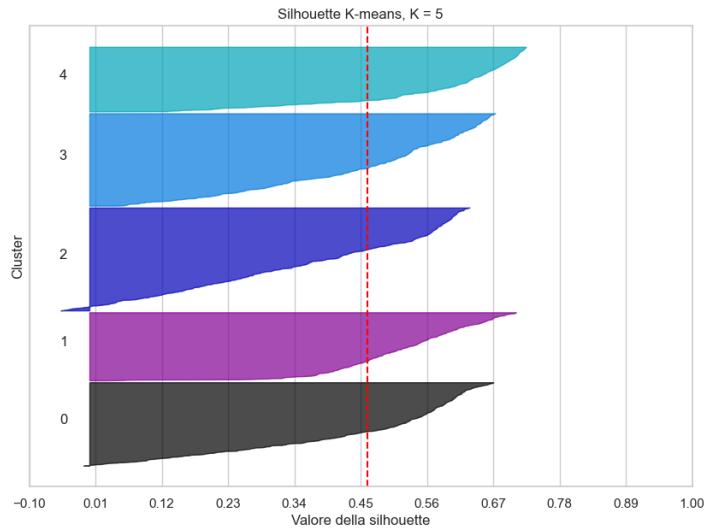


Figure 28: Grafico della Silhouette.

---

<sup>5</sup>La silhouette è una metrica unsupervised, in cui ad ogni punto del dataset viene assegnato un valore compreso tra -1 ed 1. I valori vicini a 1 indicano che i punti sono ben assegnati al proprio cluster, valori prossimi a 0 suggeriscono sovrapposizione tra cluster, e valori negativi indicano un'assegnazione errata.

## 4.3 DBSCAN

*DBSCAN*, acronimo di *Density-Based Spatial Clustering of Applications with Noise*, è un algoritmo di clustering non supervisionato che si basa sulla densità dei dati per individuare gruppi o cluster. È particolarmente utile quando i cluster hanno forme irregolari e i dati contengono elementi di disturbo o rumore.

Il funzionamento dell'algoritmo si basa su due parametri fondamentali: un raggio di ricerca, indicato con  $\epsilon$ , e un numero minimo di punti richiesti all'interno di quel raggio, noto come *minPts*. Partendo da un punto qualsiasi, DBSCAN verifica quanti altri punti si trovano nel raggio  $\epsilon$ . Se il numero è sufficiente a soddisfare il requisito di **minpts**, quel punto viene considerato un "punto core" e viene utilizzato per espandere un cluster, includendo tutti i punti vicini che rispettano la stessa condizione di densità. I punti che si trovano vicino ai cluster ma non soddisfano i requisiti di densità vengono classificati come "punti di bordo", mentre quelli troppo isolati, che non appartengono a nessun cluster, sono considerati rumore. Questa capacità di distinguere tra zone dense e zone più isolate rende DBSCAN adatto a gestire situazioni in cui sono presenti outlier. Uno dei principali vantaggi dell'algoritmo è che non richiede di specificare in anticipo il numero di cluster, a differenza di altri metodi come *k-means*, descritto nella sez. 4.2.

### 4.3.1 Applicazione dell'algoritmo DBSCAN

Come primo step, l'algoritmo del DBSCAN richiede l'inserimento dei parametri  $\epsilon$  e *minpts*. Per ottemperare a tale step, è stato utilizzato il *Metodo Nearest-Neighbors* che consiste nel graficare una funzione che rappresenta, punto per punto, la distanza tra il campione in esame ed il suo *k-esimo* vicino. L'obiettivo è quello di selezionare  $\epsilon$  in corrispondenza del gomito della funzione, che corrisponde al punto in cui la distanza del *k-esimo* vicino rispetto al punto in esame aumenta esponenzialmente, indicando che tale vicino è appartenente ad un altro cluster oppure è un punto rumoroso.

Una valore troppo grande o piccolo del parametro  $\epsilon$ , fissato il parametro *minpts*, potrebbe portare rispettivamente ad un numero basso o elevato di cluster. La Figura 29 mostra l'applicazione del *Metodo Nearest-Neighbors* per i seguenti valori di *k(minpts)*: 3, 4, 5, 6, 7.

Empiricamente, data la disposizione del dataset rappresentato in Figura 22, sono stati scelti i seguenti parametri:

- $\epsilon$ : 3
- *minpts*: 5 (curva marrone)

Come dimostra la Figura 30, tali valori forniscono un ottimo bilanciamento tra efficacia ed efficienza, ottenendo un valore di Silhouette relativamente elevato, con un basso numero di cluster e di media delle distanze dei punti rumorosi, rispetto a tutte le altre combinazioni di parametri presenti nelle matrici.

È interessante osservare come nella prima matrice in Figura 30 sia presente una cella vuota. Questo è dovuto al fatto che per la configurazione selezionata non sono stati individuati punti rumorosi.

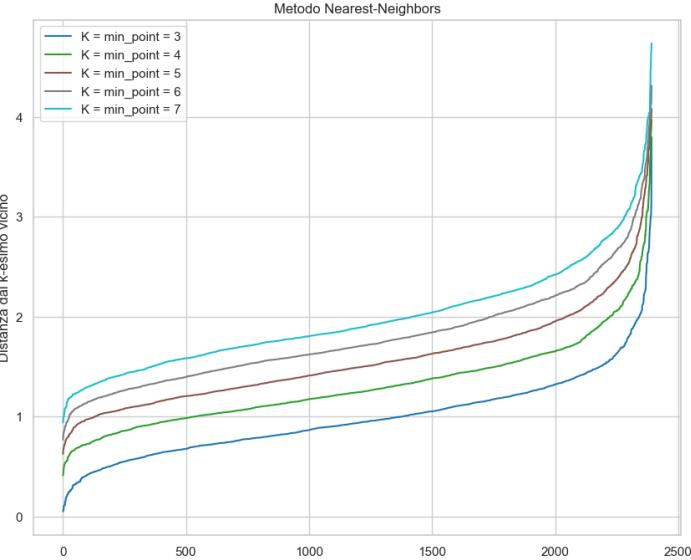


Figure 29: Metodo Nearest-Neighbors per diversi valori di  $K$

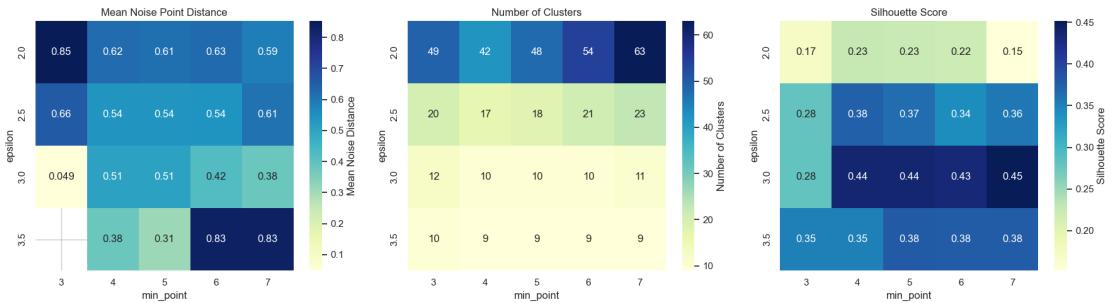


Figure 30: Media distanze punti rumorosi, numero di cluster e silhouette per diverse applicazioni del DBSCAN

In Figura 31 è presente il risultato dell'applicazione dell'algoritmo DBSCAN.

#### 4.3.2 Analisi dei risultati ottenuti

I cluster ottenuti in Figura 31 sono composti esattamente da:

- **0:** 387 studenti.
- **1:** 333 studenti.

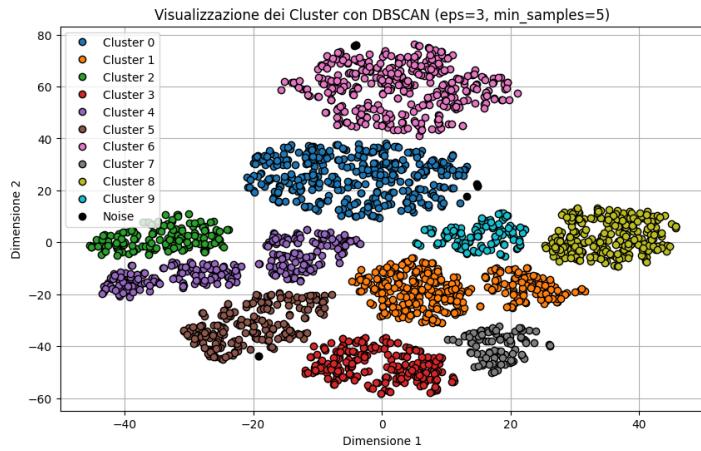


Figure 31: Clusterizzazione mediante DBSCAN.

- **2:** 158 studenti.
- **3:** 210 studenti.
- **4:** 264 studenti.
- **5:** 201 studenti.
- **6:** 373 studenti.
- **7:** 97 studenti.
- **8:** 240 studenti.
- **9:** 117 studenti.
- **Noise/Rumore/-1:** 9 studenti.

Nella seguente trattazione, l’analisi del cluster rumoroso sarà effettuata alla fine del paragrafo 4.3.2 Come possibile osservare dalla Figura 32, che valuta la distribuzione del GPA tra i vari gruppi, i cluster *0*, *5*, *7* sono composti da studenti che ottengono buoni voti a scuola. Intersecando il valor medio con la varianza e tenendo conto del numero di studenti per i cluster in esame, è possibile notare come il cluster *0* sia il migliore in assoluto.

D’altro canto, sempre tenendo conto di media, varianza e popolosità, e ad eccezione del cluster rumoroso, il cluster *1* è composto da studenti che ottengono punteggi scolastici mediamente peggiori rispetto a tutti gli altri gruppi.

Dalla Figura 33 si può osservare come il cluster *0*, ovvero il migliore, sia formato per la maggioranza da donne, mentre il cluster *1*, ovvero il peggiore, sia composto da soli uomini. Tale pattern non può definirsi di natura generale ed applicabile ad ognuno dei

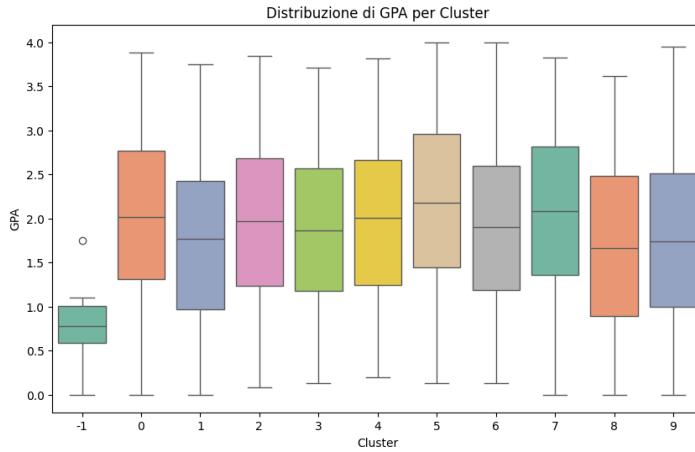


Figure 32: Distribuzione di GPA per cluster.

cluster di Figura 31. Ad esempio il cluster 4 è composto in prevalenza da uomini, ed ottiene un valore di GPA superiore rispetto al cluster 8, composto da sole donne.

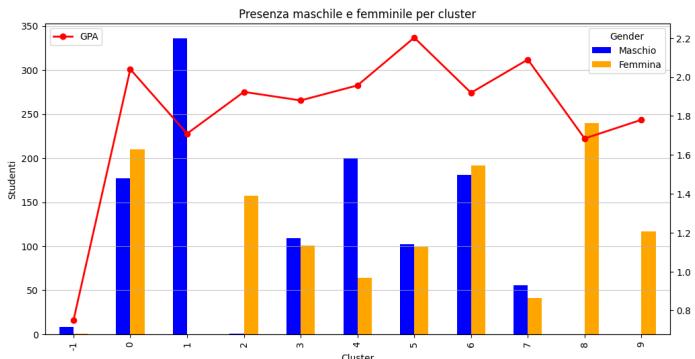


Figure 33: Presenza maschile/femminile e GPA per cluster.

La Figura 34 mostra la medie di ore settimanali dedicate allo studio per ciascun cluster, da cui è possibile osservare come gli studenti migliori (cluster 0) investano un tempo di poco superiore allo studio rispetto agli studenti peggiori (cluster 1).

La classe "Rumorosa", rappresentata in Figura 31, comprende solamente 9 studenti. Come mostrato nella Figura 35, questi punti si trovano ai margini dei cluster 0, 5 e 6. Questo suggerisce che il cluster rumoroso, in maniera controtuitiva, sia composto da studenti che si collocano mediamente in una posizione meno favorevole rispetto agli altri cluster, e da individui il cui indice GPA risulti fuori dai range di confidenza dei valori GPA degli altri gruppi.

La Figura 33 evidenzia che il cluster "rumoroso" è costituito prevalentemente da uomini, mentre dalla Figura 34 emerge che gli studenti appartenenti a questo gruppo

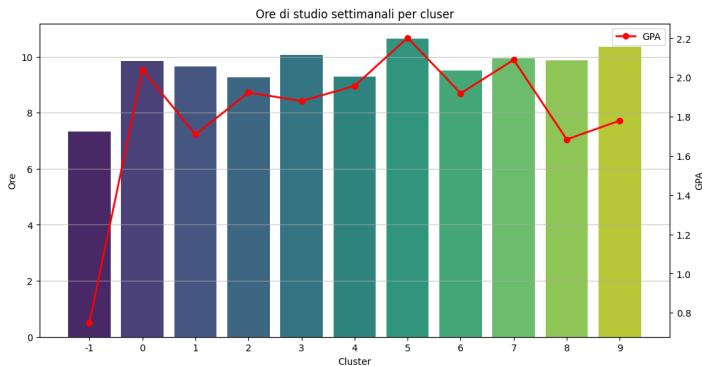


Figure 34: Ore di studio medie per cluster.

dedicano la maggior parte del loro tempo ad attività diverse dallo studio. È importante sottolineare che tali considerazioni si basano su indicatori statistici e, pertanto, sono influenzate dalla dimensione del campione analizzato. In questo caso specifico, il cluster "rumoroso" include un numero limitato di studenti.

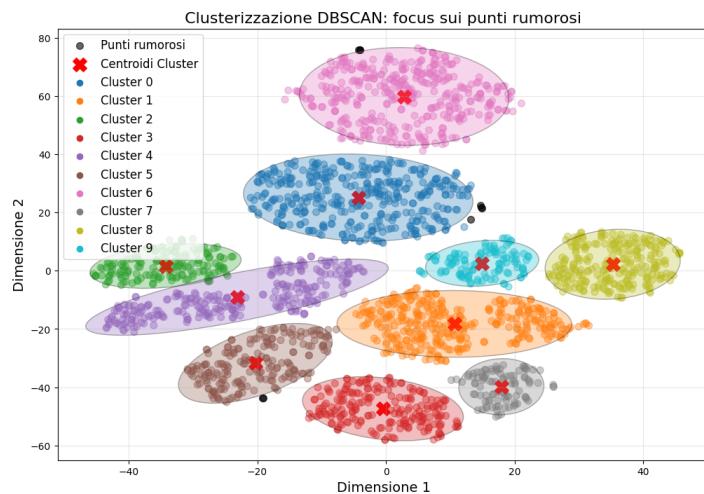


Figure 35: Applicazione dell'algoritmo DBSCAN: focus sui *punti rumorosi*.

Il modello DBSCAN realizzato al punto 4.3.1 ottiene un valor medio di Silhouette pari a 0.44, come mostra la Figura 36.

E' interessante osservare come alcuni studenti del cluster 0 (gruppo di studenti con voti mediamente migliori) misurino un valore negativo, significativo del fatto che tali studenti hanno probabilità maggiore di appartenere ai cluster 6, 4, 9.

Un'altra osservazione importante riguarda il cluster 4, in cui diversi punti potrebbero appartenere ai cluster 2, 5, 1 o 0. Inoltre, i punti effettivamente assegnati a questo gruppo (con silhouette maggiore di zero) risultano internamente distanti tra loro, indi-

cando che la distanza intra-cluster non sia stata minimizzata in modo ottimale. Come evidenziato nella Figura 31, il cluster 4 non presenta una struttura "densa" ma tende a espandersi verso altri cluster, risultando poco compatto. In altre parole, nel gruppo 4, siccome una buona parte di studenti è vicina agli studenti del gruppo 2, allora anche la media delle valutazioni (GPA) e la rispettiva varianza saranno simili (si faccia riferimento alla Figura 32).

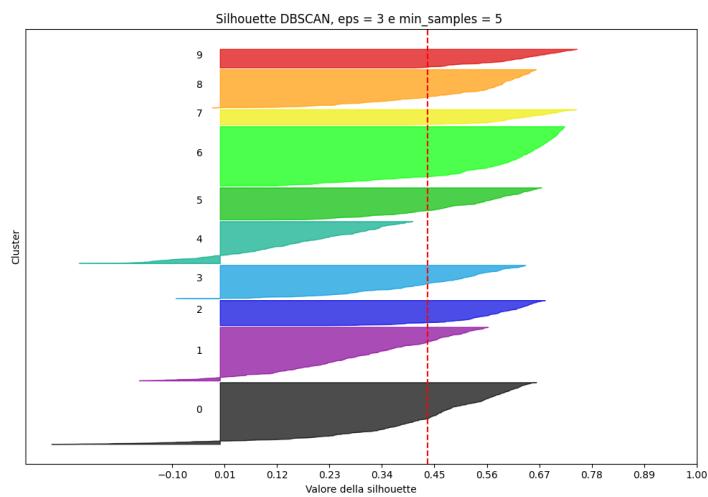


Figure 36: Grafico della Silhouette.

## 5 S&P 500 prediction: Introduzione

Il presente progetto si concentra sull'analisi di serie temporali applicata ai dati relativi all'indice Standard and Poor's 500 (S&P 500), uno dei benchmark finanziari più rilevanti a livello globale. L'S&P 500 rappresenta l'andamento delle 500 maggiori società quotate nelle borse statunitensi, ed è un indicatore fondamentale per comprendere la performance del mercato azionario. Al 31 dicembre 2020, più di 5,4 trilioni di dollari erano investiti in asset legati alla performance di questo indice.

In particolare, l'analisi condotta si è focalizzata su due aspetti principali:

- Un'analisi generale delle serie temporali relative all'indice S&P 500 a livello globale.
- Un'analisi specifica dell'andamento giornaliero delle azioni di NVIDIA, confrontando i valori di apertura e chiusura per valutare i trend e predire risultati futuri.

### Obiettivi del progetto

Gli obiettivi di questo progetto sono:

- Analizzare l'andamento storico dell'indice S&P 500 per individuare trend e comportamenti significativi e sviluppare previsioni basate sui dati storici.
- Studiare le variazioni giornaliere del titolo NVIDIA per comprendere l'andamento delle sue azioni, identificare eventuali pattern ricorrenti e sviluppare previsioni basate sui dati storici.

Questa relazione presenta i metodi utilizzati, i risultati ottenuti e le interpretazioni delle analisi svolte, offrendo una panoramica completa sulle dinamiche esplorate.

## 6 S&P 500 prediction: Dataset

Il dataset utilizzato per l'analisi delle serie temporali relative all'indice S&P500 è stato reperito dalla piattaforma Kaggle ed disponibile al seguente link: <https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks>.

Il dataset in questione è composto da tre file CSV che contengono informazioni relative all'indice S&P 500, alle aziende che lo compongono e ai dati storici delle loro azioni. Di seguito viene fornita una descrizione dettagliata dei file e delle colonne incluse:

- **sp500\_companies.csv** Questo file fornisce i metadati relativi alle aziende che compongono l'indice S&P 500. Le colonne incluse sono:
  - **Exchange**: La borsa valori presso cui i titoli dell'azienda sono negoziati.
  - **Symbol**: Il simbolo del titolo azionario.
  - **Longname**: Il nome completo dell'azienda.
  - **Sector**: Il settore di appartenenza dell'azienda.
  - **Industry**: L'industria, all'interno del settore, in cui opera l'azienda.
  - **Currentprice**: Il prezzo attuale del titolo azionario.
  - **Marketcap**: La capitalizzazione di mercato attuale.
  - **Ebitda**: Gli utili prima di interessi, imposte, deprezzamento e ammortamenti.
  - **Revenuegrowth**: La crescita dei ricavi dell'azienda.
- **sp500\_index.csv** Questo file contiene i dati storici relativi al valore dell'indice S&P 500. Le colonne incluse sono:
  - **Date**: La data della rilevazione.
  - **S&P500**: Il valore dell'indice S&P 500 nella data specificata.
- **sp500\_stocks.csv** Questo file include i dati storici delle azioni per ciascuna azienda presente nell'indice S&P 500. Le colonne incluse sono:
  - **Date**: La data della rilevazione.
  - **Symbol**: Il simbolo/ticker dell'azienda.
  - **Adj Close**: Il prezzo di chiusura rettificato, che tiene conto di azioni come dividendi e split.
  - **Close**: Il prezzo al momento della chiusura del mercato.
  - **High**: Il valore massimo registrato nel periodo considerato.
  - **Low**: Il valore minimo registrato nel periodo considerato.
  - **Open**: Il prezzo al momento dell'apertura del mercato.
  - **Volume**: Il volume di scambi registrato.

## 6.1 ETL - Preparazione serie temporale

Prima di iniziare le analisi effettive, abbiamo eseguito alcune operazioni di ETL. Nello specifico, per quanto riguarda il dataset *sp500\_index.csv*, abbiamo esplicitamente convertito la colonna *"Date"* in formato *datetime*, così da non riscontrare eventuali problemi futuri quando si andranno ad utilizzare grafici ecc.

Dopo aver fatto ciò, siamo andati a verificare la presenza di eventuali valori nulli. Fortunatamente questa analisi è stata negativa quindi non c'è stata necessità di effettuare la rimozione di alcun valore.

In seguito abbiamo suddiviso le analisi in base agli indici da noi utilizzati per effettuare le relative previsioni, ossia i valori di *S&P500* per quanto riguarda il dataset *sp500\_index.csv*, ed il corrispettivo prezzo, per le varie azioni di vendita-acquisto per l'azienda Nvidia, ottenuti dal dataset *sp500\_stocks.csv* su cui è stata basata la seconda analisi.

### 6.1.1 Time series analysis parametro S&P500

La prima sezione che affronteremo è quella riguardante il parametro *S&P500*, contenuto all'interno del dataset *sp500\_index.csv*.

Il primo passo è stato quello dell'Exploratory Data Analysis, ossia andare a studiare l'andamento dei dati in possesso. Nello specifico, l'andamento del nostro parametro *S&P500* è riportato nel grafico mostrato in Figura 37.

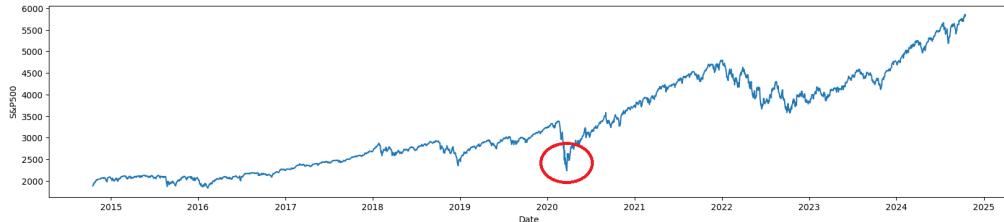


Figure 37: Andamento indice S&P500

Possiamo osservare un forte picco verso il basso in corrispondenza dell'inizio del 2020. Ciò può essere riconducibile alla difficile situazione che, in quel momento, ci vedeva lottare contro il COVID-19. Affinché potessimo comprendere meglio l'andamento in questo periodo così da individuare eventuali anomalie, siamo andati ad analizzare più nel dettaglio la nostra serie. L'andamento preciso viene mostrato in Figura 38, dove abbiamo considerato l'andamento relativo all'anno in questione, e lo zoom effettuato ha naturalmente confermato la nostra ipotesi in merito al periodo in questione.

Dopo aver analizzato la nostra serie e dopo aver accertato che non ci fossero anomalie o problemi, abbiamo provveduto con la suddivisione del dataset in training e test (rispettivamente 80% e 20%), per poi effettuare le analisi per la stazionarietà ed il seguente

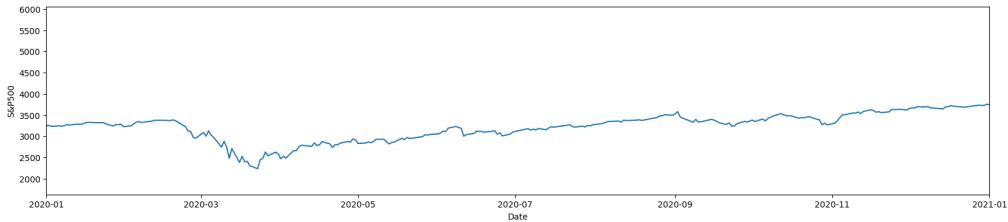


Figure 38: Zoom della situazione critica del 2020

addestramento del modello sui dati del test set. In Figura 39 si può vedere un esempio dei dati utilizzati.

```

Train DataFrame:
    Date      S&P500
0 2014-10-17  1886.76
1 2014-10-20  1904.01
2 2014-10-21  1941.28
3 2014-10-22  1927.11
4 2014-10-23  1950.82

Test DataFrame:
    Date      S&P500
2311 2023-12-22  4754.63
2312 2023-12-26  4774.75
2313 2023-12-27  4781.58
2314 2023-12-28  4783.35
2315 2023-12-29  4769.83

```

Figure 39: Descrizione dei dataframe di training e di test utilizzati

La *stazionarietà* è una proprietà fondamentale di una serie temporale, che implica che i suoi valori non dipendano dal tempo. In una serie stazionaria, media e varianza rimangono costanti nel tempo e la distribuzione di probabilità non cambia quando la serie viene traslata nel tempo. Per verificare la stazionarietà, si utilizza spesso il test di radice unitaria chiamato Augmented Dickey-Fuller (ADF test). Questo metodo assume come ipotesi nulla che la serie temporale non sia stazionaria. Dopo aver calcolato il p-value, se quest'ultimo risulta inferiore a 0,05, si rifiuta l'ipotesi nulla, suggerendo che la serie è stazionaria.

Difatti, avendo il test restituito un p-value pari a 0.91311, possiamo constatare come la serie non sia stazionaria, motivo per cui abbiamo eseguito un'operazione di differenziazione. A questo punto, il test ha restituito un valore di p-value pari a  $9.6524 \times 10^{-28}$ , concludendo dunque che la serie risulti ora stazionaria. In Figura 40 è riportata la nostra serie differenziata.

In seguito abbiamo cercato di determinare i parametri  $p$  e  $q$  attraverso i grafici di *Autocorrelation* e *Partial Autocorrelation* riportati in Figura 41.

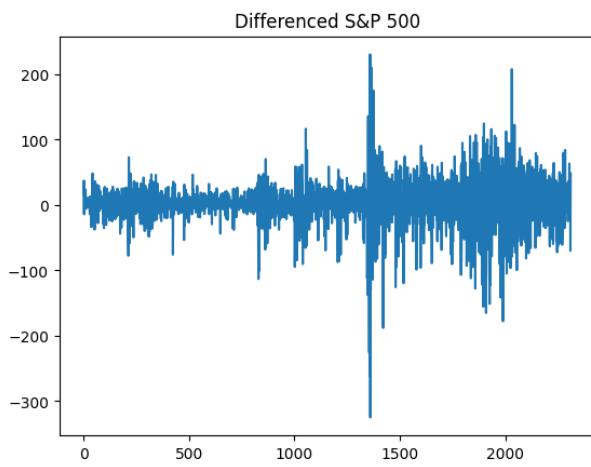


Figure 40: Serie differenziata

Analizzando i grafici in questione e grazie all'aiuto del tool automatico *auto\_arima*, abbiamo impostato un primo modello ARIMA con order pari a (2,0,2) sulla serie differenziata.

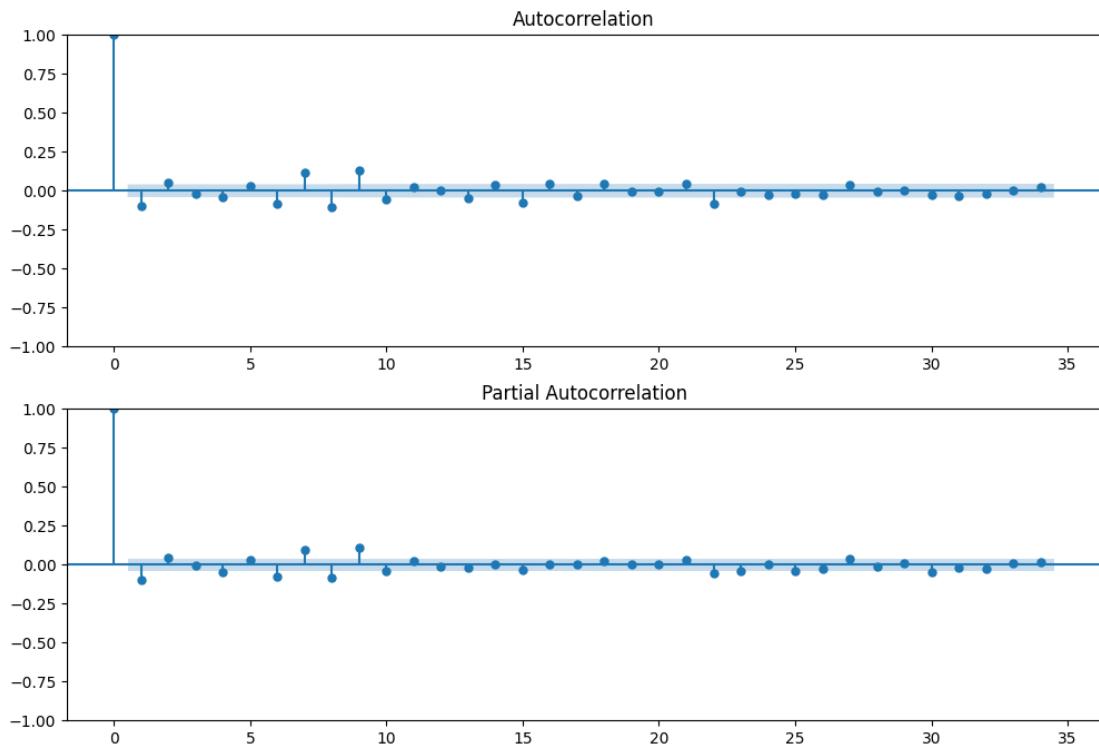


Figure 41: ACF e PACF della serie

Un primo possibile risultato del modello implementato viene mostrato in Figura 42, mentre alcuni grafici di supporto utili per comprendere i risultati ottenuti sono stati forniti in Figura 43.

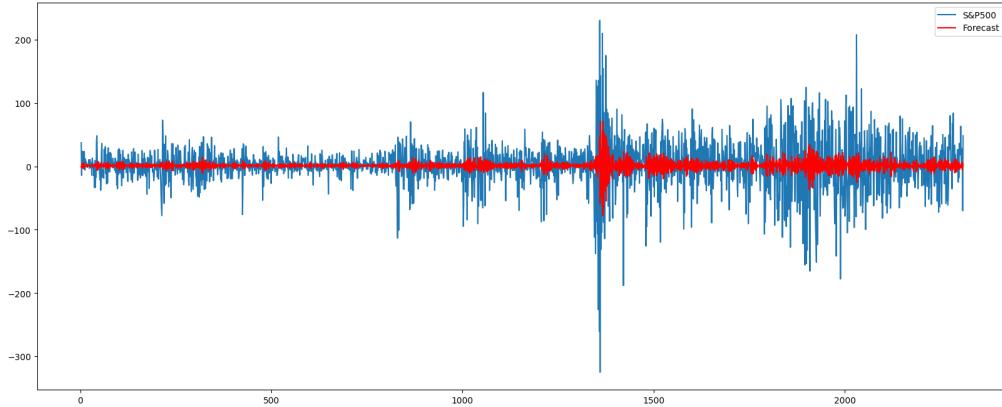


Figure 42: Andamento serie con modello ARIMA(2,0,2)

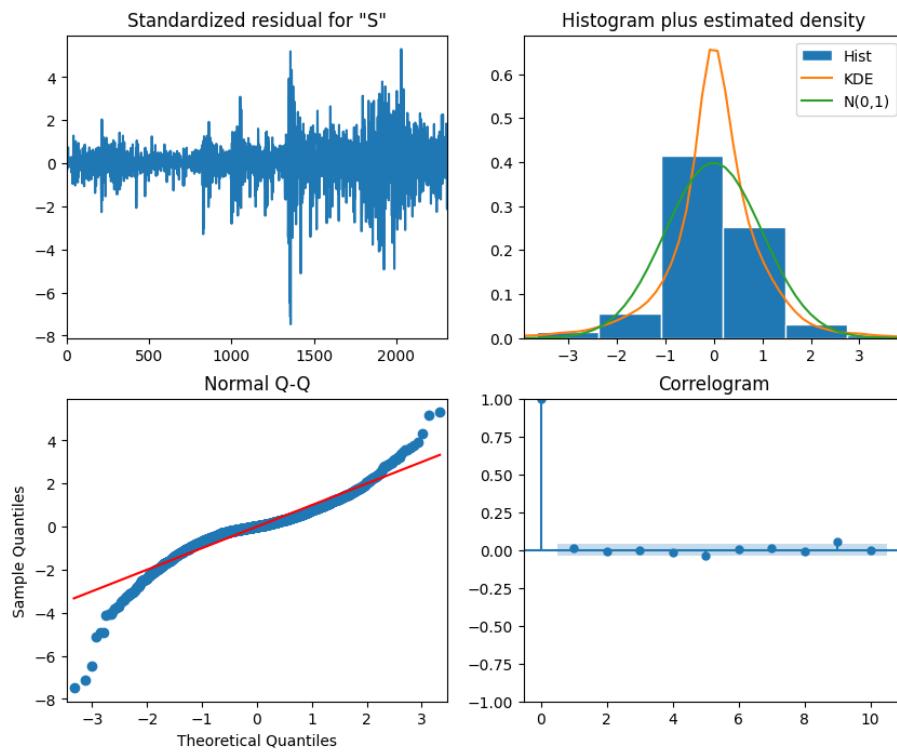


Figure 43: Grafici di supporto ARIMA(2,0,2)

Ovviamente, come risulta in Figura 43, si è ottenuto un buon correlogramma, ma

nel grafico in basso a sinistra non si è ottenuto un andamento ottimo e lineare come avremmo voluto.

Il test di Ljun-Box sui residui per i primi trenta lags sono i seguenti:

- lag 10: lb\_stat<sup>6</sup>: 11.382182, lb\_pvalue<sup>7</sup>: 0.328528
- lag 20: lb\_stat: 16.215106, lb\_pvalue: 0.703193
- lag 30: lb\_stat: 44.416037, lb\_pvalue: 0.043705

Non contenti dei risultati ottenuti e avendo notato una lieve stagionalità ogni venti giorni circa, attraverso un approccio a griglia abbiamo optato per un modello SARIMAX con ordine (10,0,10) e seasonal\_order (2,0,2,22).

Un primo andamento viene mostrato in Figura 44, mentre ulteriori dati statistici sono mostrati in Figura 45. I risultati ottenuti in quest'ultima figura sono pressoché simili al caso precedente, abbiamo un correlogramma più piatto ed una oscillazione meno marcata.

In aggiunta, il test di Ljung-Box risulta notevolmente migliorato, come dimostrano i seguenti valori:

- lag 10: lb\_stat: 0.086697, lb\_pvalue: 1.000000
- lag 20: lb\_stat: 1.104563, lb\_pvalue: 1.000000
- lag 30: lb\_stat: 14.577486, lb\_pvalue: 0.991918

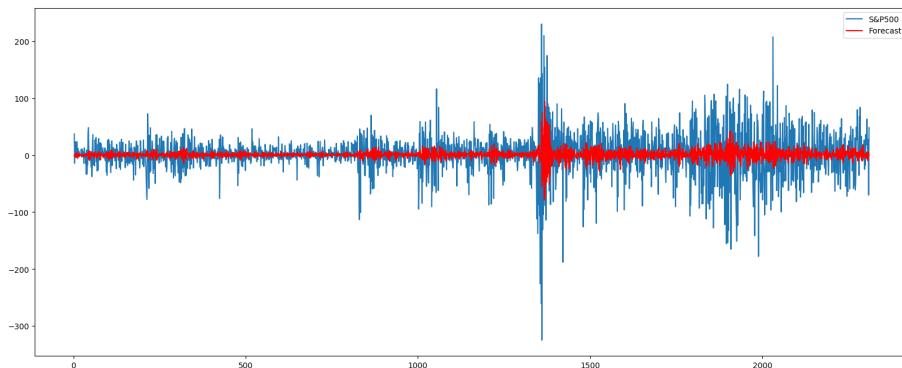


Figure 44: Andamento serie con SARIMAX(2,0,2), s.o (2,0,2,22)

In Figura 46 possiamo notare che le previsioni ottenute non sono delle migliori, ma l'andamento crescente rimarca una correlazione con i dati di test.

---

<sup>6</sup>lb\_stat (o Q): è il valore della statistica del test di Ljung-Box.

<sup>7</sup>lb\_pvalue: è il p-value associato alla statistica Q. Se il p-value è basso (tipicamente minore di 0.05), si rifiuta l'ipotesi nulla, suggerendo che i residui presentano autocorrelazione. Se il p-value è alto, non si può rifiutare l'ipotesi nulla, suggerendo assenza di autocorrelazione.

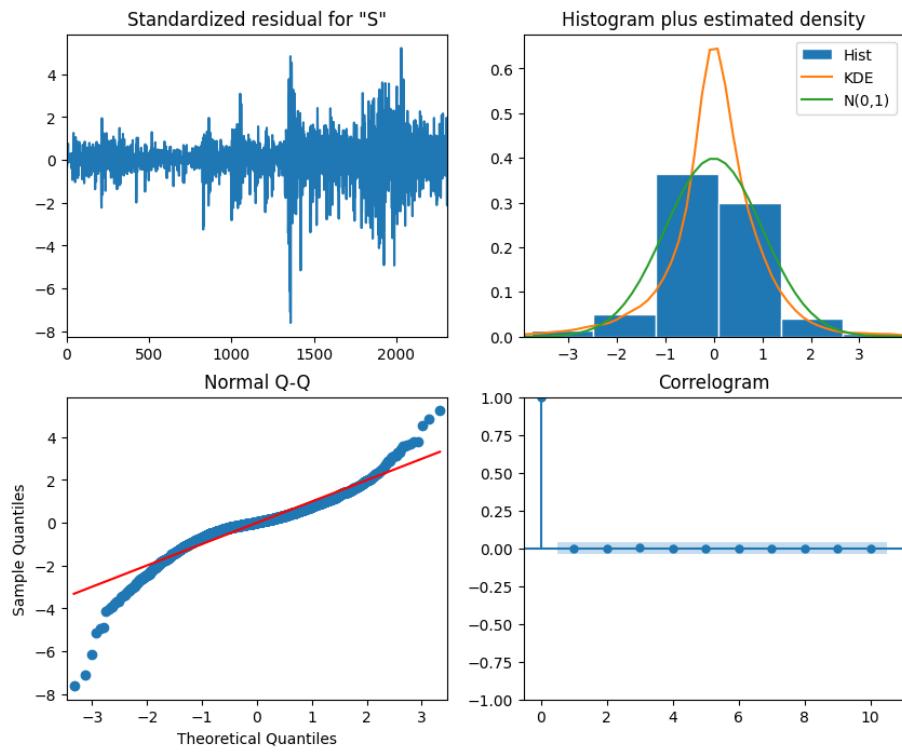


Figure 45: Grafici di supporto SARIMAX(2,0,2), s.o (2,0,2,22)

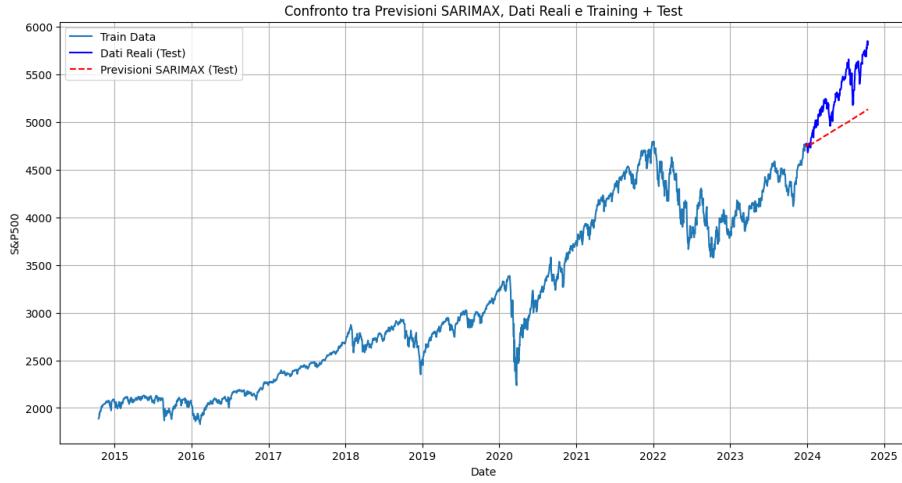


Figure 46: Previsione su dati di test con modello SARIMAX(2,0,2), s.o (2,0,2,22)

In seguito a varie valutazioni e confronti abbiamo provveduto con una aggregazione dei dati mensile, così da avere una previsione più semplice da parte del modello, nella speranza che potesse fornirci delle previsioni più accurate vista la restrizione dei dati su

cui lavorare.

### 6.1.2 Time series analysis parametro S&P500 - dati aggregati per mese

In questo caso, abbiamo effettuato una aggregazione dei dati per mese, ottenendo una struttura tabellare come quella mostrata in Figura 47 (i dati di S&P500 sono stati calcolati come valore medio dei giorni del mese).

	Date	S&P500
0	2014-10-31	1956.021818
1	2014-11-30	2044.572105
2	2014-12-31	2054.266364
3	2015-01-31	2028.178500
4	2015-02-28	2082.195789

Figure 47: Dati aggregati per mese

Successivamente abbiamo effettuato una veloce analisi esplorativa, andando ad analizzare il solito picco nell'anno 2020 riconducibile al periodo del Covid (vedi Figura 48) ed effettuando uno studio lineare dal 2021 in poi (vedi Figura 49).

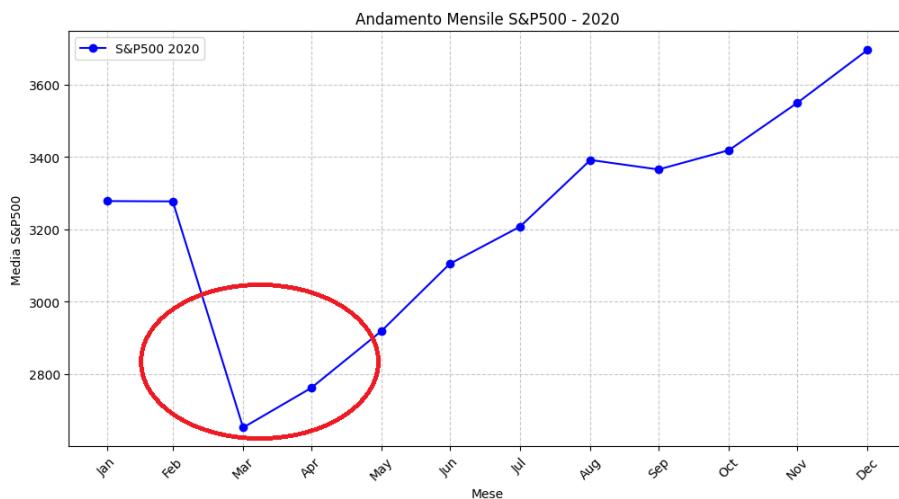


Figure 48: Analisi mensile ristretta all'anno 2020

Prima di iniziare a riflettere sul possibile modello da utilizzare, abbiamo effettuato la decomposizione STL (vedi Figura 50). Dal terzo dei quattro grafici possiamo notare una ovvia stagionalità, che verrà poi modellata attraverso un modello nella parte successiva riguardante quest'ultimo.

I dati aggregati sono stati anche qui suddivisi in training e test (80% e 20%), così

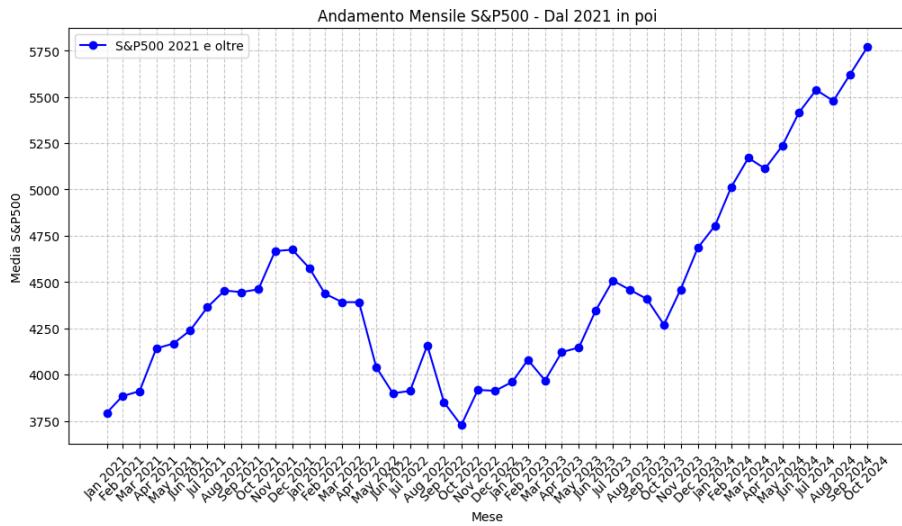


Figure 49: Analisi mensile dal 2021 in poi

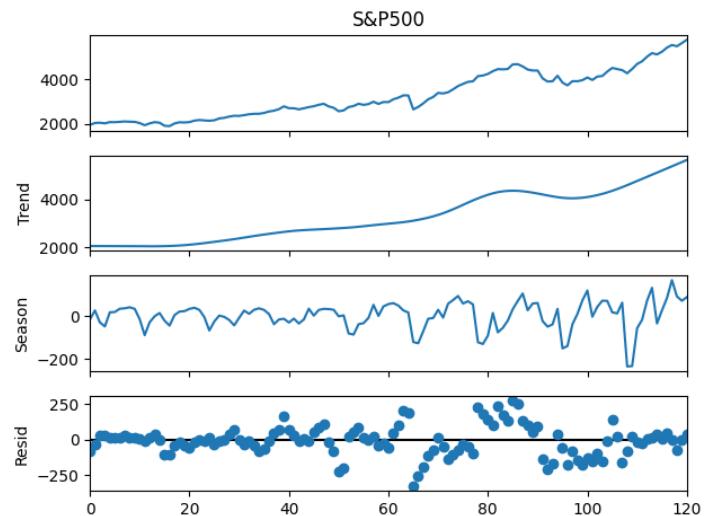


Figure 50: Decomposizione STL

da poterci consentire un'addestramento preciso e puntuale, con una piccola rimanenza di dati su cui testare il modello allenato.

In seguito al test tramite il metodo Augmented Dickey-Fuller (ADF test) abbiamo constatato che la serie non fosse stazionaria, avendo ottenuto un valore di p-value pari a 0.7974. Dopo aver eseguito una sola differenziazione, fortunatamente, il p-value è migliorato notevolmente raggiungendo il valore di  $1.2460 \times 10^{-13}$ , indice del fatto che la serie sia ora stazionaria.

In Figura 51 è riportata la serie differenziata, mentre in Figura 52 sono riportati i grafici di autocorrelazione e correlazione parziale utilizzati per stabilire i rimanenti parametri del modello ARIMA utilizzato.

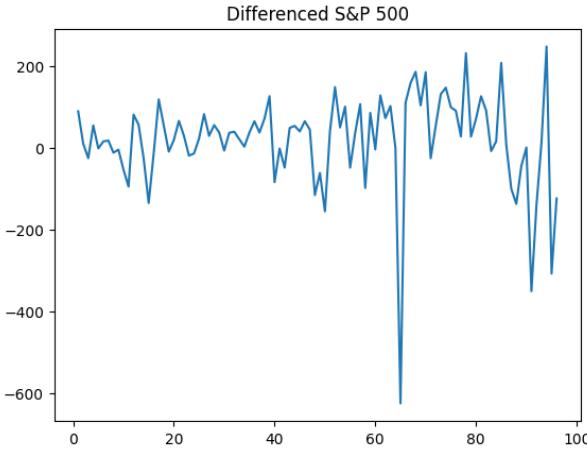


Figure 51: Serie differenziata con dati aggregati per mese

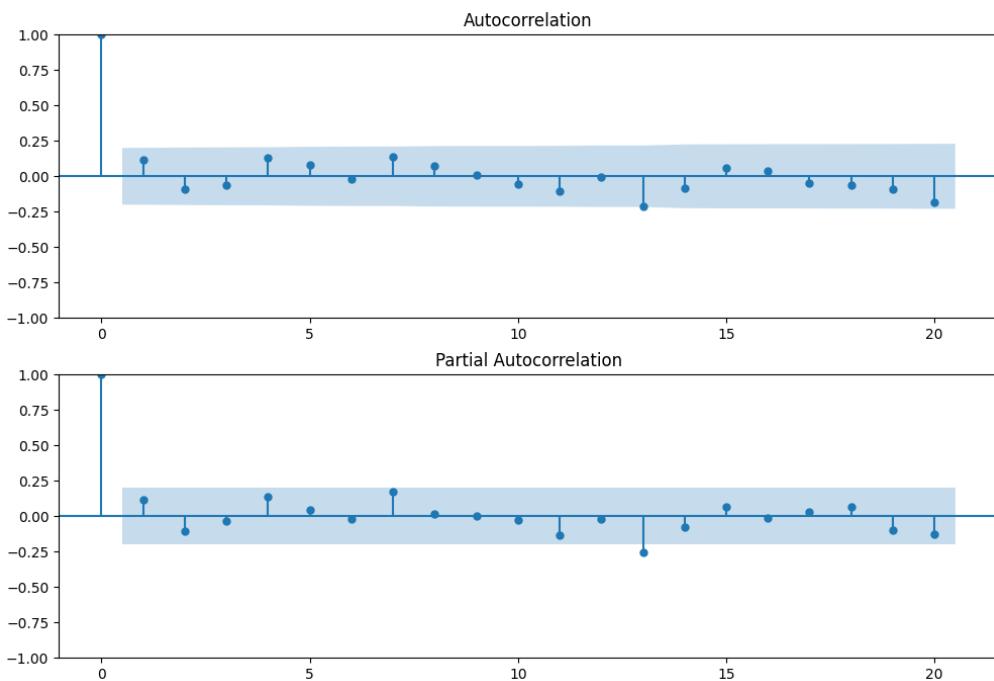


Figure 52: ACF e PACF con dati aggregati per mese

Non avendo valori al di fuori dell'intervallo di confidenza, abbiamo testato un primo modello con la serie differenziata che sfrutta ARIMA con order pari a  $(0,0,0)$ , che ha

prodotto i risultati forniti in Figura 53 ed in Figura 54.

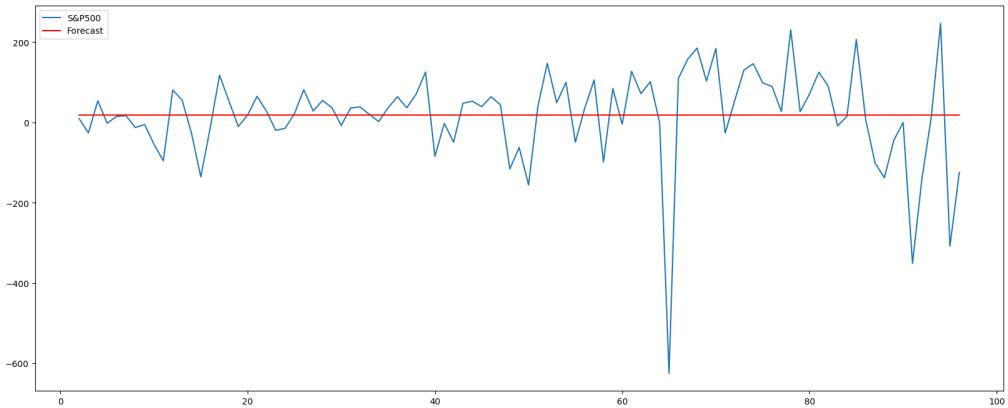


Figure 53: Andamento con modello ARIMA(0,0,0)

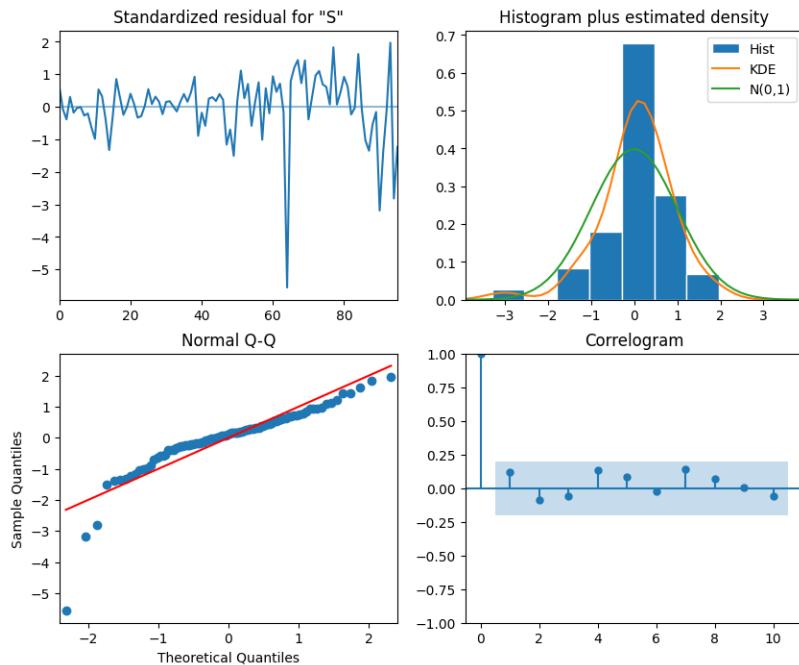


Figure 54: Grafici di supporto modello ARIMA(0,0,0)

I valori di Ljung-Box prodotti, invece, sono i seguenti:

- lag 10: lb\_stat: 7.933500, lb\_pvalue: 0.635333
- lag 20: lb\_stat: 21.353837, lb\_pvalue: 0.376575
- lag 30: lb\_stat: 32.223086, lb\_pvalue: 0.357225

Il modello dunque non performa al meglio, come è possibile notare anche dai valori di Ljung-Box che non sono prossimi al valore unitario. Di conseguenza, avendo individuato una stagionalità iniziale, abbiamo inserito un `seasonal_order` all'interno del modello ARIMA.

A conferma di ciò è stato utile osservare anche l'output restituito da `auto_arima`, che ci ha fornito un modello migliore di tipo ARIMA  $(0,0,0)(1,0,0)[13]$ ; Unendo le nostre conoscenze con i suggerimenti forniti dal tool, abbiamo pensato di sviluppare ed implementare un modello ARIMA  $(2,0,2)(1,0,1,13)$ , ossia con stagionalità di tredici giorni (il che è plausibile se andiamo ad osservare il grafico della decomposizione iniziale riportato in Figura 50).

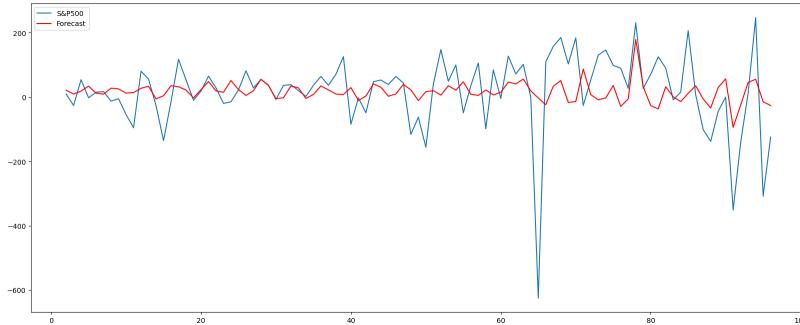


Figure 55: Andamento del modello ARIMA $(2,0,2)(1,0,1,13)$

I risultati di un primo andamento sono stati forniti in Figura 55, mentre dei grafici utilizzati da noi per l'analisi diagnostica sono riportati in Figura 56. Il correlogramma è migliorato, ma ciò che maggiormente notiamo è un quasi appiattimento dei valori sulla linea retta rossa.

Indice di miglioramento ulteriore sono i valori di Ljung-Box ottenuti, cioè:

- lag 10: `lb_stat: 3.730341, lb_pvalue: 0.958693`
- lag 20: `lb_stat: 14.938285, lb_pvalue: 0.809742`
- lag 30: `lb_stat: 19.926756, lb_pvalue: 0.918435`

Infine, le previsioni che abbiamo ottenuto su dati di test sono quelle riportate nella Figura 57. Come possiamo notare sono notevolmente migliori delle precedenti; l'andamento non segue linearmente i valori di test ma, analizzando attentamente, è possibile notare come per ogni andamento positivo dei dati di test si abbia un andamento positivo anche della previsione, mentre per ogni "discesa" e quindi picco verso il basso, si ha il corrispettivo anche per la previsione, dimostrando un adattamento ed una performance più modellabili ai dati rispetto al caso precedente senza aggregazione.

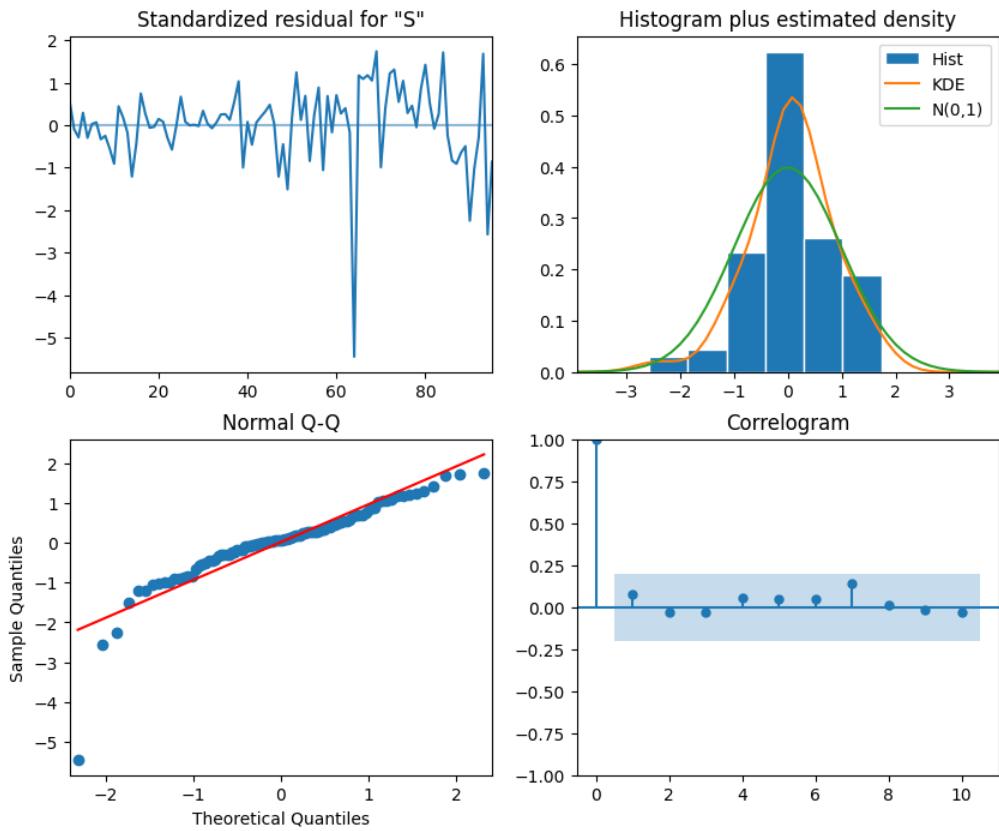


Figure 56: Grafici di supporto modello ARIMA(2,0,2)(1,0,1,13)

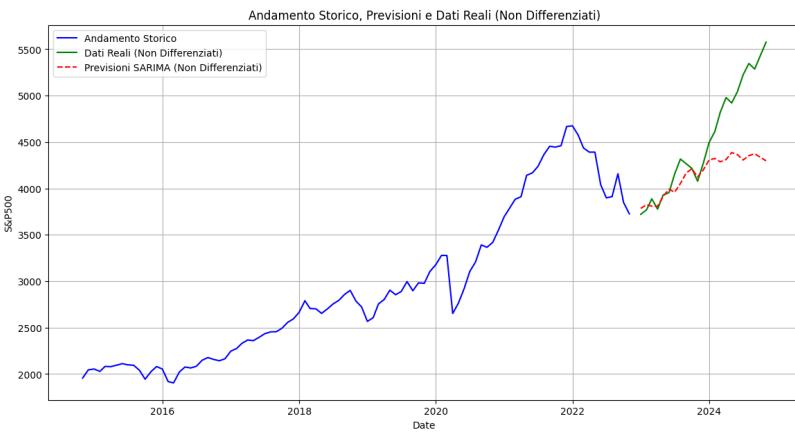


Figure 57: Previsioni su dati di test del modello ARIMA(2,0,2)(1,0,1,13)

## 6.2 Metriche finale e conclusioni

Le metriche finali ottenute da questo nostro ultimo modello sono le seguenti:

- *Mean Absolute Error (MAE)*: 120.5296
- *Mean Squared Error (MSE)*: 18471.7846
- *Root Mean Squared Error (RMSE)*: 135.9109
- *Mean Absolute Percentage Error (MAPE)*: 1.6317
- $R^2$ : -0.5420

Come è possibile notare, le metriche non sono delle migliori, nonostante i passi per lo sviluppo dei nostri modelli siano stati eseguiti in maniera precisa ed accurata. Abbiamo attribuito questi valori poco ottimali a molteplici motivazioni.

L'indice S&P 500 è altamente volatile e influenzato da fattori esterni come notizie economiche, decisioni della Federal Reserve, eventi geopolitici, ecc. I modelli SARIMA funzionano meglio con serie stazionarie, ma il mercato azionario ha tendenze a lungo termine e stagionalità irregolari che potrebbero rendere il modello meno efficace.

Potrebbero esserci effetti non lineari che il modello non riesce a catturare (ad es. crisi improvvise o rally di mercato). Il valore di  $R^2$  leggermente negativo indica che il modello non spiega bene la variabilità dei dati.

Sebbene il MAPE indichi un errore percentuale relativamente contenuto, il valore elevato di MAE e RMSE suggerisce che il modello non riesce a catturare accuratamente i movimenti dell'indice. Questo è tipico dei modelli lineari applicati a dati finanziari caratterizzati da forte volatilità.

Un possibile miglioramento sarebbe l'uso di modelli di deep learning come LSTM o metodi di regressione avanzati come XGBoost, che potrebbero meglio catturare la non linearità dell'indice.

## 7 Close - Open price prediction

### 7.1 Obiettivo del progetto

Il seguente progetto si propone come obiettivo quello di fare previsione sulla differenza tra il prezzo di chiusura e quello di apertura delle azioni della società *"Nvidia"*. Il dataset utilizzato è lo stesso di quello descritto al capitolo 6.

Questa previsione risulta essere molto importante soprattutto in ambito finanziario, in quanto aiuta l'azionista a capire il momento giusto per vendere e/o acquistare le azioni della azienda in questione.

Occorre tenere presente che, tali dati, coprono un periodo temporale compreso tra il 2010 e la fine del 2024.

### 7.2 ETL - Preparazione serie temporale

Prima di iniziare a lavorare sui modelli, è importante effettuare un analisi esplorativa seguita da una fase di ETL al fine di preparare i dati per essere utilizzati all'interno dei modelli statistici.

Come prima operazione è stato necessario ridimensionare le osservazioni della serie temporale, considerando solo ed esclusivamente le osservazioni successive al 2019. Questo perché le osservazioni precedenti avevano una media e una varianza molto basse, non in linea con gli andamenti misurati dal 2019 in poi, e avrebbero potuto portare a un abbassamento delle performance, indipendentemente dal modello statistico selezionato.

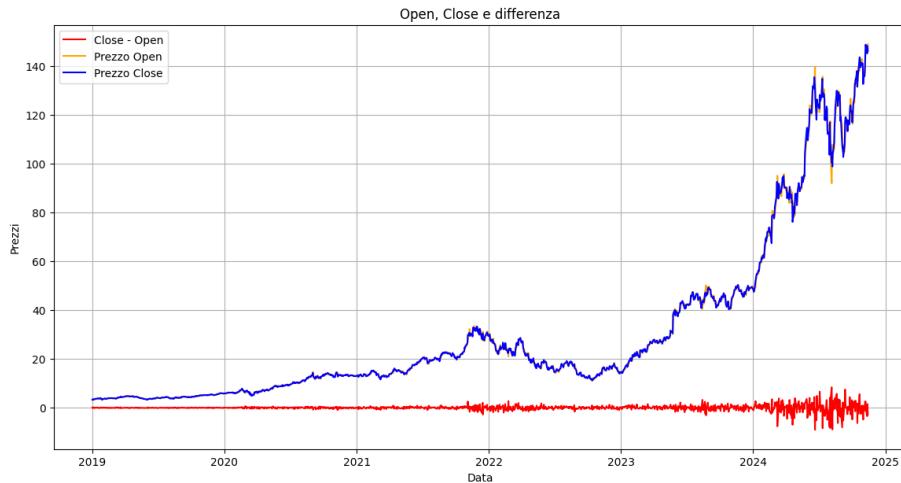


Figure 58: Andamento di Close, Open e della loro differenza

In Figura 58 è presente, in rosso, la variabile di interesse al fine dell'analisi. Come è possibile osservare sempre dalla Figura 58, i valori tendono ad essere molto bassi, con

una media vicino allo zero.

Successivamente, è stato necessario capire se ci fosse o meno la presenza di comportamenti ciclici ad intervalli lunghi e costanti. Il grafico di Figura 59 mostra l'andamento *close-open* nei 12 mesi dell'anno per gli anni dal 2019 al 2024. Non è possibile osservare nessuna esplicita correlazione tra i trend nei diversi anni, né tantomeno componenti stagionali con intervallo annuale.

E' però possibile evidenziare due situazioni particolari:

- **Anno 2020.** Nell'anno 2020 è scoppiata in Europa e nel resto del mondo la pandemia per *Covid-19*. Questo ha difatto abbassato il volume ed il valore delle azioni di *Nvidia*. Per questo motivo abbiamo delle osservazioni *close-open* mediamente più basse rispetto a quelle degli altri anni.
- **Luglio 2024.** In questo periodo, la differenza tra *close - open* ha raggiunto minimi mai visti primi. Questo a causa di andamenti generali del mercato azionario e dal fenomeno dello *Stock Split*.<sup>8</sup>

Una volta effettuate le analisi preliminari, è stato necessario suddividere il dataset in due componenti:

- **Trainig:** Contiene 1393 campioni.
- **Test:** Contiene 85 campioni.

In Figura 60 è mostrata una decomposizione *STL* applicata al dataset di training, considerando una componente stagionale con un periodo di 5 giorni, corrispondente a una settimana lavorativa.

La selezione di questo periodo è stata guidata da ricerche effettuate sul Web e da test con diverse finestre temporali, al fine di individuare la più appropriata. Sebbene la Figura 60 non evidensi in modo netto componenti stagionali o trend, l'utilizzo di un periodo di 5 giorni si è rivelato relativamente più adeguato rispetto a opzioni più lunghe o più brevi.

Un passo fondamentale per l'applicazione di modelli statistici è la valutazione della proprietà di stazionarietà<sup>9</sup>. Per fare ciò è stato utilizzato il test *ADF*<sup>10</sup>. Il p-value è risultato essere pari a  $3.3123 \times 10^{-12}$ , molto minore del 5%, quindi si può rifiutare l'ipotesi nulla<sup>11</sup> ed accettare l'ipotesi alternativa<sup>12</sup>, decretando la stazionarietà della serie temporale.

---

<sup>8</sup>Uno stock split è una divisione delle azioni di una società per aumentarne il numero e ridurne il prezzo unitario, senza cambiarne il valore totale di mercato.

<sup>9</sup>Una serie temporale si dice stazionaria se e soltanto se ha media e varianza costante.

<sup>10</sup>Augmented Dickey–Fuller è un test statistico per valutare la stazionarietà di una serie temporale.

<sup>11</sup> $H_0$ : I dati presentano una radice unitaria, quindi la serie non è stazionaria.

<sup>12</sup> $H_1$ : I dati non presentano una radice unitaria, quindi la serie è stazionaria.

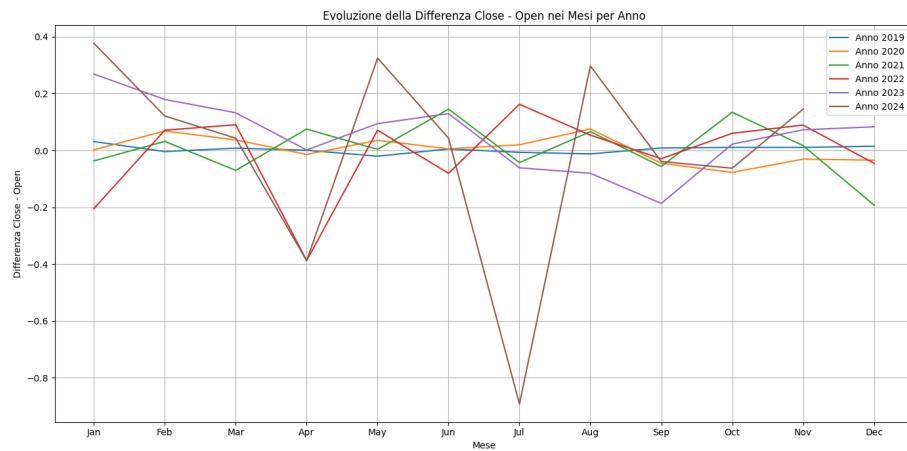


Figure 59: Evoluzione della differenza *close-open* nei mesi per anno.

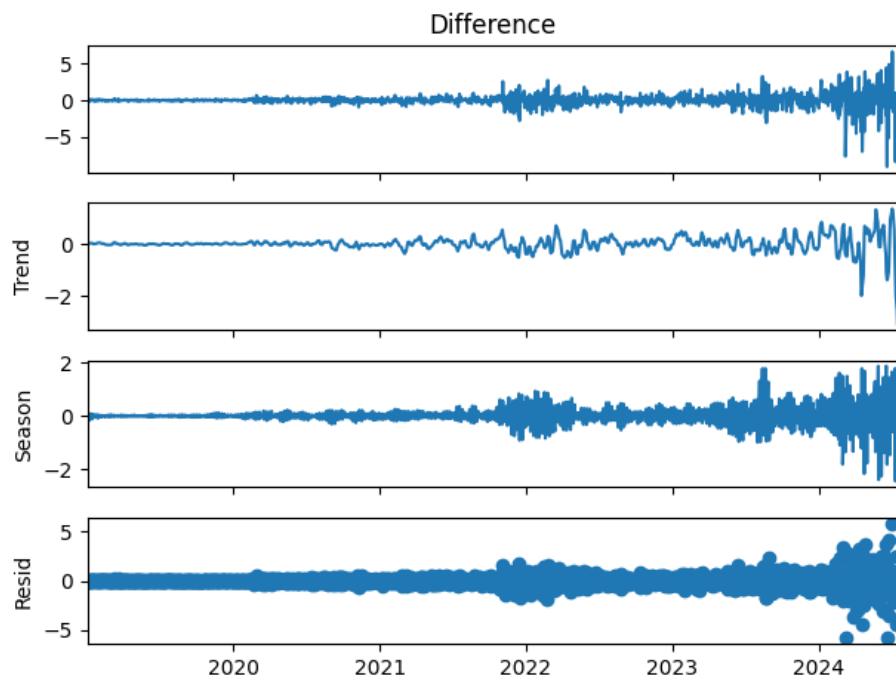


Figure 60: Decomposizione *STL* di *close-open*.

Data la complessità del fenomeno e la sua natura volubile, è stato deciso di partire da un modello ARIMA per poi aggiungere una componente stagionale ed una variabile esogena.

### 7.3 Modello ARIMA

La Figura 61 mostra rispettivamente i grafici di autocorrelazione (ACF) e autocorrelazione parziale (PACF), utili per individuare, rispettivamente, l'ordine della parte autoregressiva (AR) e della parte a media mobile (MA).

Con l'obiettivo di minimizzare la complessità del sistema massimizzando le performance del modello, sono stati scelti come ordine per la parte AR e MA, rispettivamente i valori  $p=2$  e  $q=2$ . Questo perché, in corrispondenza del secondo lag, i valori superano la soglia di confidenza, il che significa che i valori della serie ( $y(t)$ ) e degli errori ( $e(t)$ ), all'istante di tempo  $t$  dipendono fortemente dai valori, rispettivamente della serie e degli errori nei due istanti di tempo precedenti.

Utilizzando la funzione "auto\_arima", è stata individuata la stessa identica combinazione per gli ordini  $p$  e  $q$ , con un valore di AIC pari a 3661.922.

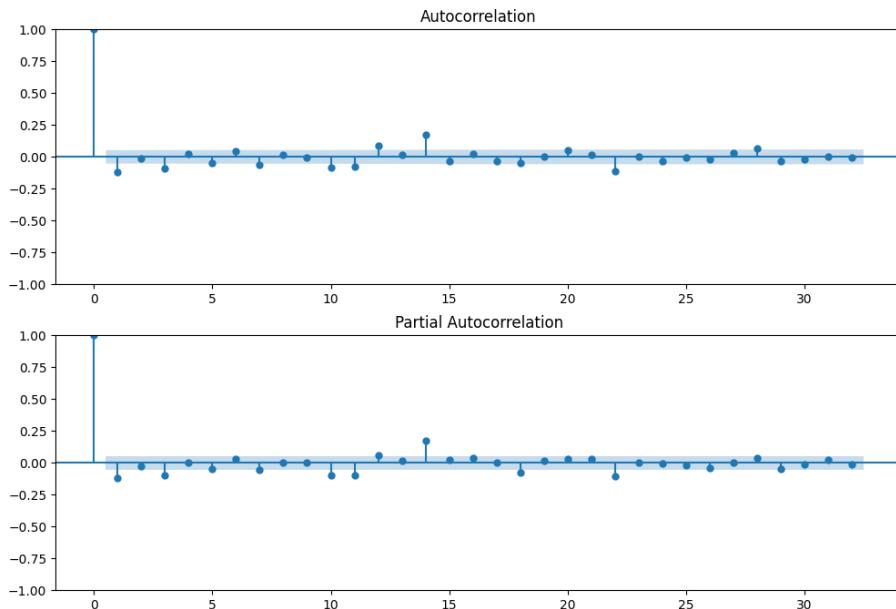


Figure 61: ACF e PACF per la stima dell'ordine AR ed MA.

Per validare il modello, è necessario osservare i residui e verificare che non ci sia correlazione. Infatti, una correlazione tra i residui indicherebbe una dinamica non correttamente modellata.

I risultati di tale analisi sono rappresentati in Figura 62, in cui da sinistra verso destra e dall'alto verso il basso troviamo:

- **Residui standardizzati:** Il grafico dei residui standardizzati permette di valutare l'adeguatezza del modello. Quanto più i residui sono piccoli, tanto più il loro valore medio è vicino allo zero, indicando che il modello è riuscito a rappresentare

correttamente la dinamica del fenomeno contenuta nella serie temporale.

- **Istogramma e densità dei residui:** In un modello ideale, in cui il residuo è assimilabile a un rumore bianco, esso dovrebbe avere media nulla e varianza costante. Questo grafico aiuta a valutare quanto il residuo si avvicina a questa condizione. In particolare, se le barre blu dell'istogramma sono centrate rispetto alla densità stimata ("KDE") e alla distribuzione normale teorica ("N"), allora il residuo è simile a un rumore bianco, suggerendo che il modello è in grado di spiegare efficacemente la dinamica della serie temporale.
- **Normal Q-Q:** Questo grafico confronta i quantili dei residui con quelli di una distribuzione normale teorica. Se i punti seguono la linea rossa, i residui sono normalmente distribuiti, suggerendo che essi sono prevalentemente costituiti da rumore e che il modello ha catturato correttamente la dinamica del sistema. Al contrario, deviazioni evidenti, come code pesanti o asimmetrie, indicano scostamenti dalla normalità.
- **Correlogramma:** L'autocorrelazione (ACF) dei residui permette di evidenziare eventuali strutture residue nei dati, che si manifestano quando il fenomeno non è stato modellato correttamente. In particolare, se il valore dell'ACF supera la soglia di confidenza a un certo lag  $L$ , significa che il residuo al tempo  $t$  è correlato a una combinazione di valori dei residui fino al tempo  $t - L$ , indicando una dinamica nascosta e non correttamente modellata nel modello.

Il modello ARIMA(2,0,2) non può essere ritenuto completamente valido a causa della presenza di un residuo significativo al *lag 9*.

Inoltre, effettuando un'analisi statistica tramite il test di Ljung-Box, i risultati riportati in Tabella 6 mostrano un p-value del 7% fino al lag 10, indicando una correlazione non significativa. Tuttavia, dal lag 10 fino ai lag 20 e 30, il p-value scende sotto il 5%, suggerendo la presenza di correlazione nei residui e quindi una dinamica non completamente modellata.

Si potrebbe quindi domandare perché, nel correlogramma in Figura 62, il valore dell'ACF dei residui risulti significativo, mentre il test in Tabella 6 indica assenza di autocorrelazione. Questo è dovuto alla natura statistica del test: tanto più è alto il valore del p-value, tanto più risulta essere vera l'ipotesi nulla<sup>13</sup>. Nel caso di Tabella 6, sebbene il p-value sia superiore al 5%, non è particolarmente elevato, suggerendo che l'ipotesi nulla venga accettata con una certa incertezza.

---

<sup>13</sup>  $H_0$ : I residui non sono correlati tra loro.

lag	p-value	lb_stat
10	0.0725	17.083779
20	$8.817412 \times 10^{-2}$	77.922524
30	$3.957635 \times 10^{-11}$	110.399574

Table 6: Risultati del test di Ljung-Box per il modello ARIMA(2,0,2)

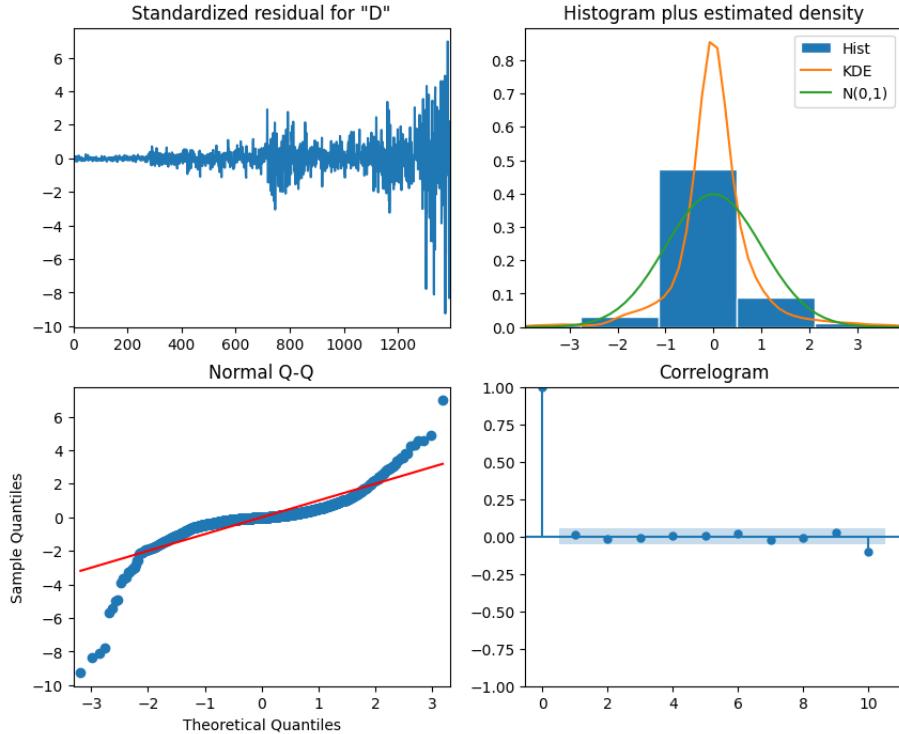


Figure 62: Grafici dei residui del modello ARIMA(2,0,2)

### 7.3.1 Risultati

Per valutare le prestazioni del modello sono state utilizzate le osservazioni contenute nel dataset di test, che coprono il periodo da luglio 2024 a novembre 2024. In Figura 63 si può osservare come il modello, almeno nelle prime osservazioni, segua l'andamento e il trend dei "Dati Reali", seppur smorzandone i valori.

Man mano che le previsioni si estendono nel tempo, si nota che il valore restituito dal modello tende ad "appiatirsi" intorno al valore medio della serie temporale.

Questo comportamento è attribuibile alle componenti autoregressive (AR) e di media mobile (MA), che, nelle previsioni a lungo termine, tendono rispettivamente a produrre valori che si avvicinano alla media della serie (AR) o che coincidono esattamente con essa (MA).

In Figura 64 è riportato il grafico delle osservazioni passate (dataset di training) e

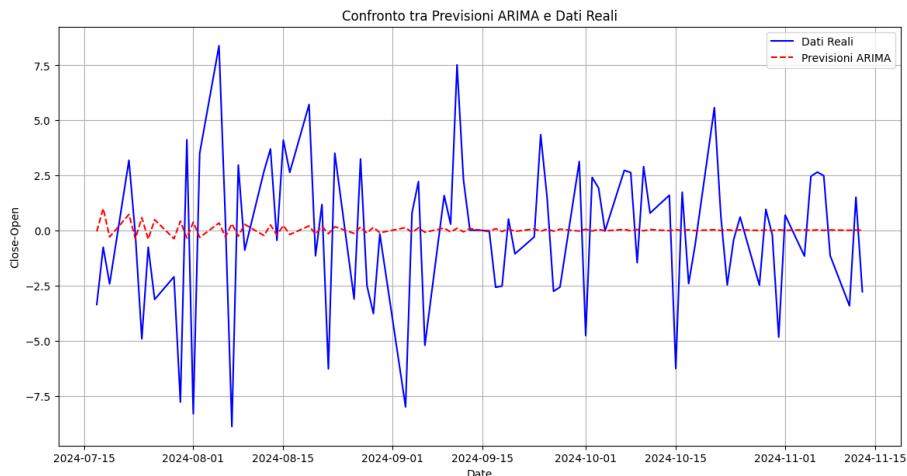


Figure 63: Grafico delle previsioni del modello ARIMA(2,0,2).

delle previsioni effettuate sul dataset di test. È interessante notare che l'intervallo di confidenza tende ad aumentare all'aumentare dell'orizzonte previsionale.

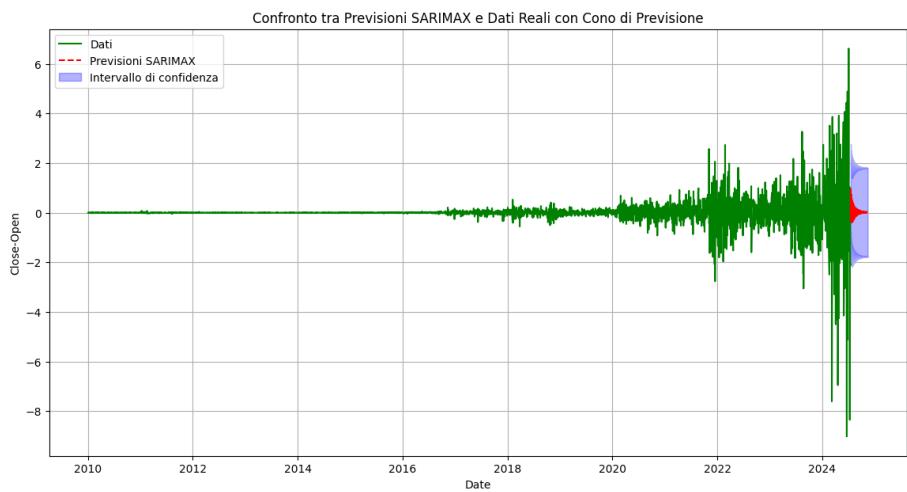


Figure 64: Grafico della confidenza del modello ARIMA(2,0,2).

I risultati ottenuti per le diverse metriche di valutazione sono:

- **Mean Absolute Error (MAE):** 2.7005. Indica l'errore medio assoluto tra i valori previsti e quelli reali, esprimendo l'errore medio in termini di unità della variabile analizzata (differenza close-open). Più basso è il MAE, migliore è la capacità del modello di adattarsi ai dati reali.
- **Mean Squared Error (MSE):** 11.9856. Penalizza maggiormente gli errori più

grandi rispetto al MAE, rendendo evidente la presenza di outlier. Un valore elevato suggerisce che il modello ha difficoltà a prevedere alcune variazioni significative della serie temporale.

- **Mean Absolute Percentage Error (MAPE):** 1.0673. Esprime l'errore percentuale medio rispetto ai valori reali. In questo caso, il valore del 106.73% indica che le previsioni del modello sono molto imprecise, suggerendo una scarsa capacità predittiva.
- $R^2$ : -0.0193. Il coefficiente di determinazione misura quanto il modello riesca a spiegare la variabilità della serie temporale. Un valore negativo indica che il modello è peggiore rispetto a una semplice previsione basata sulla media storica, suggerendo che l'ARIMA(2,0,2) potrebbe non essere adatto per catturare la dinamica della differenza close-open delle azioni NVIDIA.

## 7.4 Modello SARIMAX

A partire dei risultati raggiunti con il modello ARIMA(2,0,2) ampiamente descritto al punto 7.3, considerando la componente stagionale con periodo 5 e tenendo conto della complessità del task, si è optato per un modello SARIMAX.

La scelta dell'ordine autoregressivo (AR) e quello a media mobile (MA) è identico a quanto visto nella sezione 7.3.

Per bilanciare efficienza ed efficacia, la componente stagionale è stata modellata con  $p = 2$ ,  $d = 0$ ,  $q = 2$  e  $P = 5$ . Questi valori sono frutto di prove *trial & error*.

Per quanto riguarda la componente esogena<sup>14</sup> è stato scelto il "Volume"<sup>15</sup> delle vendite di *Nvidia*, poiché un'elevata attività di scambio può amplificare la volatilità e influenzare la differenza close-open. In particolare, volumi elevati possono generare gap tra chiusura e apertura, mentre volumi bassi tendono a stabilizzare questa differenza.

I grafici dei residui presenti in Figura 65 mostrano una media molto bassa e tendente allo zero, con valori nel correlogramma per i primi dieci lag tutti inferiori alla soglia di significatività.

Il test di Ljung-Box presente in tabella 7 evidenzia un p-value molto alto 65,8% per i primi dieci lag, evidenziando come l'ipotesi nulla sia sicuramente quella valida.

Tali risultati sono migliori di quelli presentati con il modello ARIMA(2,0,2) nella sezione 7.3 a causa di considerazioni su stagionalità e variabili esogene che migliorano la validità del modello, aumentandone purtroppo la complessità.

---

<sup>14</sup>Variabile esterna non direttamente parte del fenomeno analizzato, ma in grado di influenzarlo.

<sup>15</sup>Il volume delle vendite rappresenta il numero totale di azioni scambiate in un determinato periodo di tempo. Indica l'attività e la liquidità di un titolo, aiutando a identificare trend e confermare movimenti di prezzo.

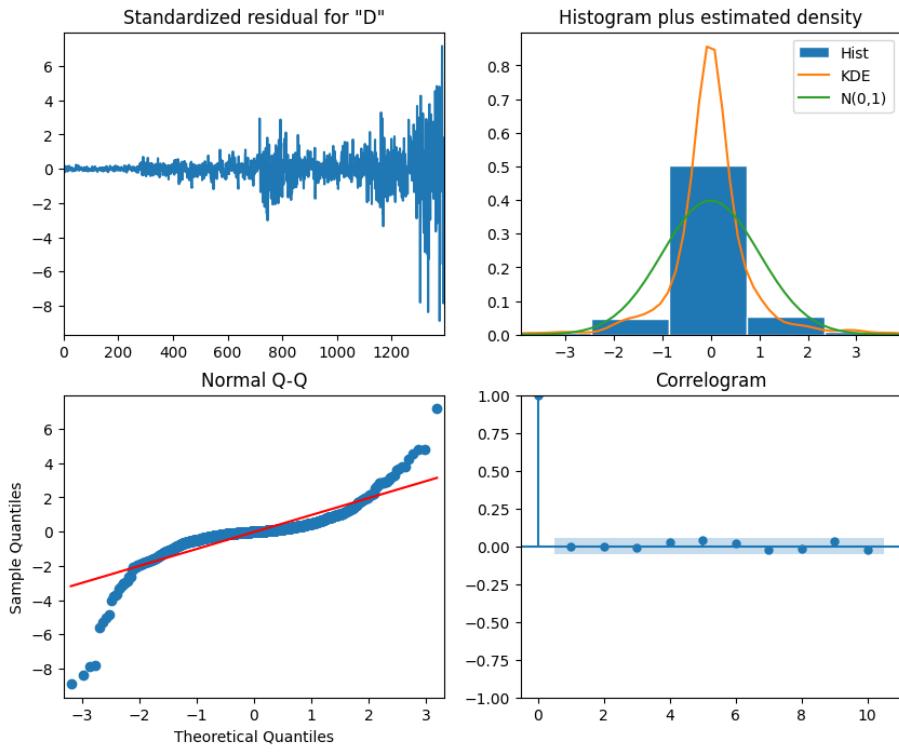


Figure 65: Grafici dei residui del modello SARIMAX(2,0,2)(2,0,2,5).

lag	p-value	lb_stat
10	$6.585188 \times 10^{-1}$	7.695843
20	$4.026686 \times 10^{-6}$	61.592220
30	$1.768444 \times 10^{-8}$	93.727485

Table 7: Risultati del test di Ljung-Box per il modello SARIMAX(2,0,2)(2,0,2,5)

#### 7.4.1 Risultati

La Figura 66 mostra le previsioni effettuate dal modello SARIMAX per il periodo da luglio 2024 a novembre 2024.

Con l'inclusione della variabile esogena e della componente stagionale, il modello SARIMAX riesce a seguire più accuratamente i trend a lungo termine, riducendo significativamente il fenomeno di "schiacciamento" verso la media che si manifesta nelle previsioni a lungo termine del modello ARIMA.

Un confronto interessante può essere fatto tra i risultati mostrati nella Figura 63 e quelli nella Figura 66. La differenza è significativa. Nel caso del modello ARIMA si osserva un rapido decremento dei valori che si appiattiscono verso la media della serie, soprattutto nelle previsioni più lunghe. Al contrario, nel modello SARIMAX,

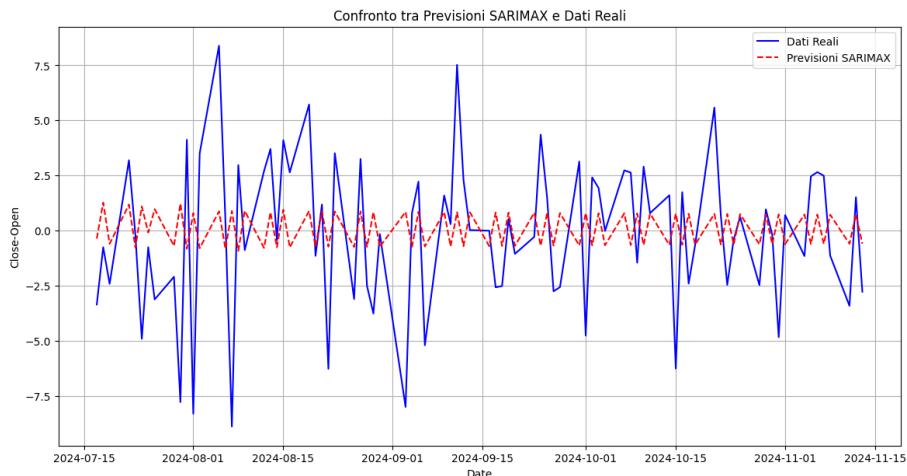


Figure 66: Grafico delle previsioni del modello SARIMAX(2,0,2)(2,0,2,5).

grazie alla variabile stagionale e alla componente esogena, questo fenomeno è attenuato, consentendo una previsione più aderente ai trend reali.

Inoltre l'intervallo di confidenza risulta essere più ristretto nel modello SARIMAX (Figura 67) rispetto a quello del modello ARIMA (Figura 64). Questo accade perché l'inclusione della variabile esogena e della componente stagionale consente al modello di adattarsi meglio ai pattern reali presenti nei dati, riducendo l'incertezza nelle previsioni a lungo termine. La variabile esogena, infatti, fornisce informazioni supplementari che migliorano la capacità predittiva del modello, mentre la componente stagionale permette di captare eventuali ciclicità o fluttuazioni periodiche, rafforzando la stima della confidenza.

I risultati delle metriche di valutazione ottenuti per il modello **SARIMAX(2,0,2)(2,0,2,5)**, applicato alla differenza *Close - Open* delle azioni NVIDIA dal 2019 al 2024, considerando i volumi di vendita come variabile esogena, sono i seguenti:

- **Mean Absolute Error (MAE):** 2.7788.

Il MAE misura l'errore medio in valore assoluto senza distinguere tra sovrastime e sottostime, un valore più basso è segno di una maggiore precisione delle previsioni. Tuttavia, da solo non fornisce informazioni sulla distribuzione degli errori.

- **Mean Squared Error (MSE):** 12.7825.

Il MSE, essendo l'errore medio quadratico, amplifica gli errori più grandi rispetto al MAE. Il valore relativamente alto suggerisce che il modello può avere difficoltà a catturare le variazioni più estreme della differenza *Close - Open*, generando errori più elevati nei momenti di maggiore volatilità del mercato.

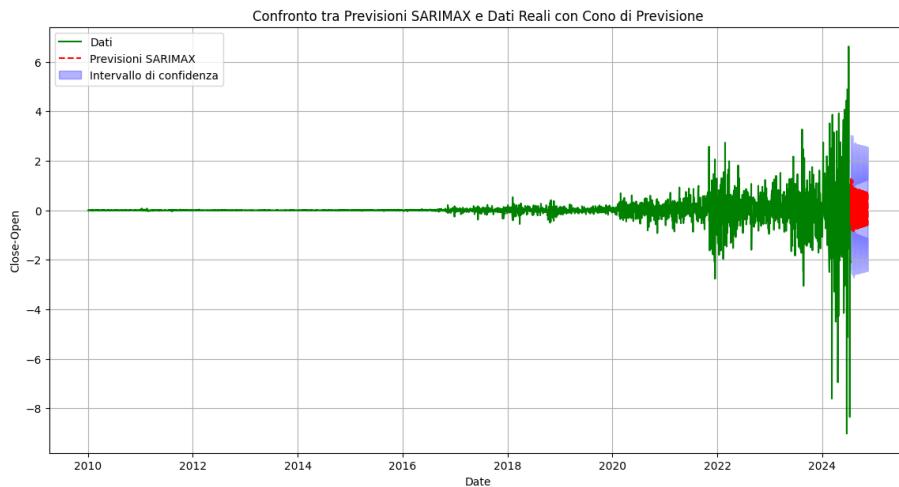


Figure 67: Grafico della confidenza del modello SARIMAX(2,0,2)(2,0,2,5).

- **Mean Absolute Percentage Error (MAPE):** 3.5753.

Il valore ottenuto, pari a 3.5753, indica che l'errore medio assoluto è circa 3.58 volte il valore reale. Questo suggerisce che il modello ha una capacità predittiva piuttosto limitata, che comporta un'alta imprecisione delle previsioni rispetto ai dati reali.

- **R-squared ( $R^2$ ):** -0.0871.

Il valore negativo di  $R^2$  è un segnale critico: indica che il modello non è in grado di spiegare la variabilità dei dati meglio di una semplice media storica. In altre parole, le previsioni ottenute dal modello SARIMAX, anche includendo i volumi di vendita come variabile esogena, non riescono a catturare in modo efficace il comportamento della differenza *Close - Open*.

## 7.5 Conclusioni

Dal punto di vista dell'analisi dei residui, il modello SARIMAX (7.4) si dimostra superiore rispetto al più semplice ARIMA (7.3), grazie alla capacità di incorporare sia la componente stagionale che una variabile esogena. Questo permette di ottenere previsioni potenzialmente più affidabili, poiché il modello è in grado di catturare meglio le dinamiche sottostanti della serie temporale.

I risultati ottenuti dai modelli ARIMA e SARIMAX, presentati rispettivamente nelle sezioni 7.3.1 e 7.4.1, sebbene non ottimali, devono essere interpretati nel contesto del fenomeno analizzato. La scelta di un modello statistico per la previsione dipende sempre dal bilanciamento tra semplicità e capacità predittiva, e in questo caso i modelli adottati forniscono un compromesso accettabile.

Optare per ARIMA o SARIMAX, piuttosto che per modelli più complessi come *ARCH* e/o *GARCH*, implica necessariamente una minore capacità di catturare l'intera complessità del fenomeno. Infatti, questi ultimi modelli sono progettati per gestire esplicitamente la volatilità variabile nel tempo, mentre ARIMA e SARIMAX non considerano direttamente gli effetti di shock economici o altri fattori esterni che possono influenzare la dinamica della serie.

Tuttavia, la scelta di modelli meno complessi può essere giustificata dalla necessità di una maggiore interpretabilità e da una minore richiesta computazionale, rendendoli più adatti in determinati contesti applicativi.

## List of Figures

1	Distribuzione delle età degli studenti. . . . .	7
2	Distribuzione di genere tra gli studenti. . . . .	8
3	Distribuzione delle etnie degli studenti. . . . .	9
4	GPA medio per gruppo etnico. . . . .	9
5	Distribuzione del tempo di studio settimanale degli studenti. . . . .	10
6	Distribuzione delle classi di voto ( <i>GradeClass</i> ). . . . .	11
7	Distribuzione delle assenze degli studenti. . . . .	11
8	Distribuzione dell'educazione dei genitori rispetto alla classe di voto. . . . .	12
9	Matrice di correlazione tra le variabili numeriche e codificate. . . . .	13
10	Distribuzione di GradeClass prima di applicare le tecniche di sampling . . . . .	14
11	Distribuzione di GradeClass dopo il sampling . . . . .	15
12	Matrice di correlazione dopo le operazioni di sampling . . . . .	16
13	Classification report . . . . .	21
14	Matrice di confusione Voting Classifier . . . . .	21
15	Curva Precision-Recall . . . . .	22
16	Curva ROC . . . . .	23
17	Learning curve . . . . .	23
18	Classification report Stacking Classifier con Logistic Regression . . . . .	24
19	Matrice di confusione relativa allo Stacking Classifier con Logistic Regression . . . . .	24
20	Varianza delle variabili del dataset. . . . .	29
21	Varianza cumulativa all'aumentare del numero di componenti . . . . .	30
22	Visualizzazione dataset dopo l'applicazione <i>T-sne</i> . . . . .	30
23	Metodo di Elbow per la scelta del parametro <i>K</i> . . . . .	32
24	K-means con <i>K</i> 5. . . . .	32
25	Distribuzione dei valori <i>GPA</i> nei cluster del <i>K-means</i> . . . . .	33
26	Presenza maschile/femminile e GPA per cluster. . . . .	33
27	Ore di studio medie per cluster. . . . .	34
28	Grafico della Silhouette. . . . .	34
29	Metodo Nearest-Neighbors per diversi valori di <i>K</i> . . . . .	36
30	Media distanze punti rumorosi, numero di cluster e silhouette per diverse applicazioni del DBSCAN . . . . .	36
31	Clusterizzazione mediante DBSCAN. . . . .	37
32	Distribuzione di GPA per cluster. . . . .	38
33	Presenza maschile/femminile e GPA per cluster. . . . .	38
34	Ore di studio medie per cluster. . . . .	39
35	Applicazione dell'algoritmo DBSCAN: focus sui <i>punti rumorosi</i> . . . . .	39
36	Grafico della Silhouette. . . . .	40

37	Andamento indice S&P500 . . . . .	43
38	Zoom della situazione critica del 2020 . . . . .	44
39	Descrizione dei dataframe di training e di test utilizzati . . . . .	44
40	Serie differenziata . . . . .	45
41	ACF e PACF della serie . . . . .	45
42	Andamento serie con modello ARIMA(2,0,2) . . . . .	46
43	Grafici di supporto ARIMA(2,0,2) . . . . .	46
44	Andamento serie con SARIMAX(2,0,2), s.o (2,0,2,22) . . . . .	47
45	Grafici di supporto SARIMAX(2,0,2), s.o (2,0,2,22) . . . . .	48
46	Previsione su dati di test con modello SARIMAX(2,0,2), s.o (2,0,2,22) . . . . .	48
47	Dati aggregati per mese . . . . .	49
48	Analisi mensile ristretta all'anno 2020 . . . . .	49
49	Analisi mensile dal 2021 in poi . . . . .	50
50	Decomposizione STL . . . . .	50
51	Serie differenziata con dati aggregati per mese . . . . .	51
52	ACF e PACF con dati aggregati per mese . . . . .	51
53	Andamento con modello ARIMA(0,0,0) . . . . .	52
54	Grafici di supporto modello ARIMA(0,0,0) . . . . .	52
55	Andamento del modello ARIMA(2,0,2)(1,0,1,13) . . . . .	53
56	Grafici di supporto modello ARIMA(2,0,2)(1,0,1,13) . . . . .	54
57	Previsioni su dati di test del modello ARIMA(2,0,2)(1,0,1,13) . . . . .	54
58	Andamento di Close, Open e della loro differenza . . . . .	56
59	Evoluzione della differenza <i>close-open</i> nei mesi per anno. . . . .	58
60	Decomposizione <i>STL</i> di <i>close-open</i> . . . . .	58
61	ACF e PACF per la stima dell'ordine AR ed MA. . . . .	59
62	Grafici dei residui del modello ARIMA(2,0,2) . . . . .	61
63	Grafico delle previsioni del modello ARIMA(2,0,2). . . . .	62
64	Grafico della confidenza del modello ARIMA(2,0,2). . . . .	62
65	Grafici dei residui del modello SARIMAX(2,0,2)(2,0,2,5). . . . .	64
66	Grafico delle previsioni del modello SARIMAX(2,0,2)(2,0,2,5). . . . .	65
67	Grafico della confidenza del modello SARIMAX(2,0,2)(2,0,2,5). . . . .	66