



# StockPredictor\*

Petrillo Francesco

February 2025

---

\*GitHub Repository

## Contents

<b>1</b>	<b>Analisi del problema</b>	<b>4</b>
1.1	Specifiche PEAS . . . . .	4
1.2	Caratteristiche fondamentali del mercato azionario . . . . .	4
1.3	Il ruolo dei dati nel mercato azionario . . . . .	5
1.4	La sfida della previsione del mercato . . . . .	6
<b>2</b>	<b>Analisi dei dati</b>	<b>7</b>
2.1	Data Gathering . . . . .	7
2.2	Scelta dei dati più rilevanti . . . . .	8
<b>3</b>	<b>Preprocessing</b>	<b>9</b>
3.1	Gestione dei valori mancanti . . . . .	9
3.2	Riduzione del rumore . . . . .	10
3.2.1	Media mobile . . . . .	10
3.2.2	Individuazione del rumore . . . . .	10
3.2.3	Conclusione sulla riduzione del rumore . . . . .	13
3.3	Accorpamento dei Dati . . . . .	13
3.4	Normalizzazione dei dati . . . . .	14
3.4.1	Min-max scaling . . . . .	14
3.4.2	Implementazione . . . . .	14
<b>4</b>	<b>Definizione del modello</b>	<b>15</b>
4.1	Scelta del modello . . . . .	15
4.2	Criteri di Scelta del Modello . . . . .	15
4.3	Modelli Considerati . . . . .	15
4.4	Creazione di serie temporali . . . . .	16
4.5	Specifiche del modello . . . . .	16
<b>5</b>	<b>Valutazione del modello</b>	<b>17</b>
5.1	Fase di allenamento . . . . .	17
5.2	Fase di valutazione . . . . .	17
5.3	Verifica matematica dell'efficienza . . . . .	19
<b>6</b>	<b>Considerazioni finali</b>	<b>20</b>

## Introduzione

Negli ultimi decenni, il progresso delle tecnologie di machine learning ha aperto nuove opportunità nel settore finanziario, in particolare nel tentativo di prevedere l'andamento dei mercati azionari. La previsione dei prezzi delle azioni rappresenta una sfida complessa a causa della natura dinamica e influenzata da molteplici fattori, come notizie economiche, trend di mercato e comportamento degli investitori.

Lo scopo di questo progetto è sviluppare un modello di intelligenza artificiale basato su machine learning che possa essere sfruttato per prevedere l'andamento del mercato di un'azione in particolare.

In questa documentazione, saranno descritti i passaggi seguiti per lo sviluppo del progetto: dall'analisi dei dati, alla costruzione del modello, fino alla valutazione dei risultati e alle considerazioni finali.



# 1 Analisi del problema

## 1.1 Specifiche PEAS

Prima di iniziare con l'analisi, andiamo a definire l'ambiente in cui ci troviamo tramite le specifiche PEAS (Performance, Environment, Actuators, Sensors):

- **Performance Measure:** Gli indicatori principali per valutare il modello includono l'errore quadratico medio (MSE), l'errore assoluto medio (MAE), il punteggio  $R^2$ , la variazione percentuale delle previsioni.
- **Environment:** L'ambiente è costituito da dati storici e dati fondamentali di mercato.
- **Actuators:** Gli attuatori includono le previsioni future dei prezzi delle azioni.
- **Sensors:** I sensori sono costituiti dai dati storici del mercato utilizzati per allenare il modello e fare previsioni sul modello già allenato.

## 1.2 Caratteristiche fondamentali del mercato azionario

La prima fase nella creazione di un modello predittivo sta nella comprensione del dominio del problema; in questa sezione descriveremo le caratteristiche principali del mercato azionario al fine di capire quali feature potranno essere utilizzate per addestrare il nostro modello.

Il funzionamento del mercato azionario si basa su alcune caratteristiche chiave:

- **Volatilità:** I prezzi delle azioni possono subire variazioni significative anche in brevi periodi di tempo. La volatilità è influenzata da fattori macroeconomici (ad esempio, decisioni sui tassi di interesse, dati sull'occupazione), eventi geopolitici e risultati finanziari delle aziende.
- **Domanda e offerta:** Come in ogni mercato, il prezzo delle azioni è determinato dall'interazione tra domanda e offerta. Un aumento della domanda per un titolo spinge il prezzo verso l'alto, mentre un aumento dell'offerta tende a ridurlo.
- **Trend e stagionalità:** Il mercato azionario può mostrare trend a breve, medio o lungo termine. Alcuni settori possono essere influenzati da cicli economici o stagionali (ad esempio, il settore del turismo tende a crescere nei mesi estivi).
- **Sentiment degli investitori:** Gli aspetti psicologici giocano un ruolo importante. Eventi inattesi o notizie rilevanti possono provocare reazioni emotive, generando movimenti improvvisi dei prezzi (comportamento noto come *herding effect*, ovvero comportamento gregario).

- **Indicatori di performance:** Per valutare lo stato del mercato, gli investitori utilizzano una serie di indicatori, come gli indici di mercato (S&P 500, Nasdaq, Dow Jones) e dati finanziari specifici delle aziende (ricavi, utile netto, margine operativo).

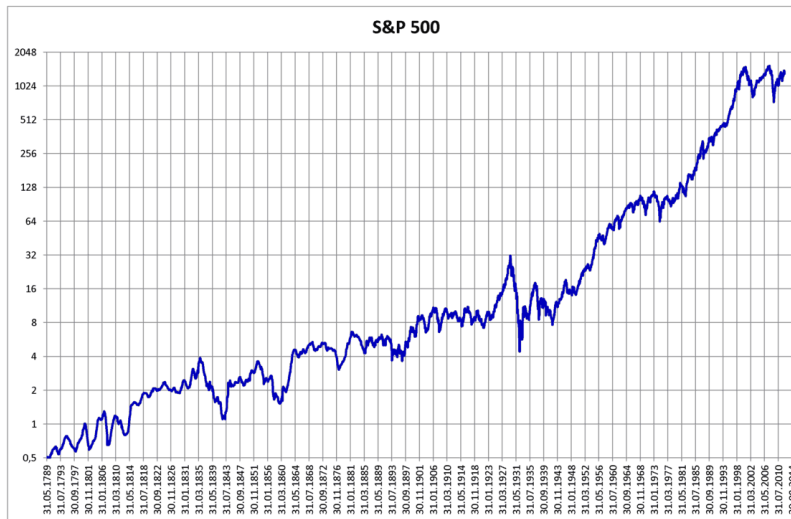


Figure 1: grafico di S&P500 fino al 2014

### 1.3 Il ruolo dei dati nel mercato azionario

Il mercato azionario genera una quantità enorme di dati in tempo reale. Questi dati sono principalmente di due tipi:

- **Dati storici:** Prezzi di apertura, chiusura, massimo, minimo e volume degli scambi di ciascun titolo. Sono fondamentali per analisi tecniche e modelli predittivi.
- **Dati fondamentali:** Informazioni relative alla performance finanziaria delle aziende (bilanci, utili, rapporti finanziari).

L'analisi dei dati storici è spesso utilizzata per identificare pattern e trend che possono aiutare a prevedere l'andamento futuro del mercato. Tuttavia, la natura complessa e dinamica del mercato rende difficile ottenere previsioni accurate con i modelli tradizionali.

## 1.4 La sfida della previsione del mercato

Prevedere il prezzo futuro di un'azione è una delle sfide più complesse del settore finanziario, poiché il mercato non segue una logica deterministica ma è influenzato da fattori casuali e non lineari. I principali ostacoli includono:

- **Rumore nei dati:** Il mercato è altamente rumoroso, con oscillazioni casuali che possono mascherare i pattern reali.
- **Eventi esterni:** Crisi economiche, cambiamenti politici o disastri naturali possono avere un impatto imprevedibile.
- **Causalità multipla:** Il prezzo delle azioni non dipende mai da un unico fattore, ma da una combinazione di variabili interconnesse.

Alla luce di queste (e molte altre) caratteristiche del mercato azionario, è evidente che non può esistere un unico “giusto” modello predittivo per lo stock market, ma si tratta di capire quali caratteristiche sono più rilevanti per l'addestramento della nostra IA.

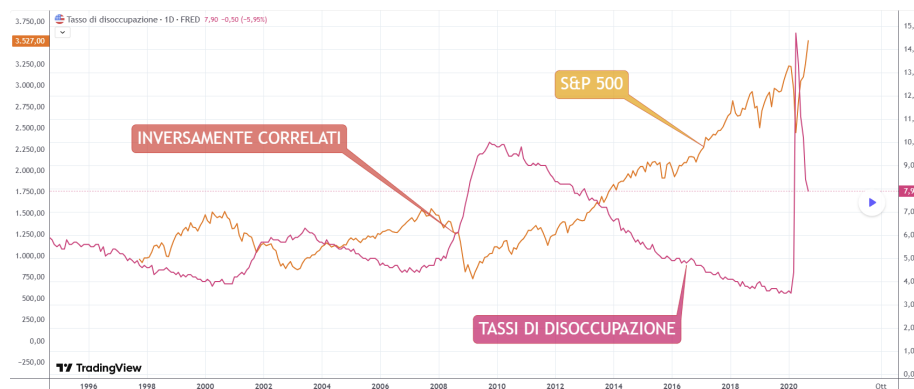


Figure 2: Correlazione tra S&P500 e tasso di disoccupazione negli USA

## 2 Analisi dei dati

In questa fase andremo a fare uno studio sui dati che il modello utilizzerà per l'allenamento, facendo una selezione delle feature più rilevanti.

### 2.1 Data Gathering

La raccolta dei dati rappresenta una fase cruciale per la costruzione di un modello di previsione accurato. Nel contesto del mercato azionario, i dati possono essere ottenuti da diverse fonti, sia gratuite che a pagamento. Tra i principali metodi disponibili ci sono l'utilizzo di database finanziari ufficiali, piattaforme di trading, servizi API offerti da provider finanziari e strumenti open-source.

Dopo un'attenta analisi delle opzioni disponibili, abbiamo scelto di utilizzare la libreria python *yfinance*, una soluzione open-source che consente di scaricare facilmente dati storici di mercato direttamente da Yahoo Finance. La scelta di *yfinance* è motivata da diversi fattori:

- **Facilità d'uso:** *yfinance* offre un'interfaccia semplice e intuitiva per scaricare dati relativi a prezzi di apertura, chiusura, massimo, minimo, volume degli scambi e molto altro.
- **Dati aggiornati:** La libreria fornisce dati aggiornati in tempo reale, rendendola una scelta ideale per analisi di mercato e costruzione di modelli predittivi.
- **Compatibilità:** I dati ottenuti tramite *yfinance* sono facilmente integrabili con strumenti di analisi come *pandas* e *numpy*, facilitando il processo di preprocessing.



Figure 3: Un esempio di grafico su yahoo finance

La scelta di utilizzare *yfinance* permette di semplificare il processo di aggiornamento dei dati, garantendo una maggiore flessibilità nel caso di modifiche al periodo di osservazione o ai titoli analizzati.

Per ottenere i dati, abbiamo utilizzato il seguente codice in Python:

```
import yfinance as yf

# Scarica i dati di Apple (AAPL) dal 2010 al 2025
data = yf.download("AAPL", start="2010-01-01", end="2025-01-31")

# Visualizza i primi cinque record
print(data.head())
```

Per l'analisi è stato scelto di utilizzare i dati storici di cinque titoli azionari rappresentativi di grandi aziende tecnologiche: Apple (AAPL), Microsoft (MSFT), Google (GOOGL), Tesla (TSLA) e Amazon (AMZN). Questi titoli sono stati selezionati per la loro rilevanza nel mercato azionario e per la disponibilità di dati storici completi.

## 2.2 Scelta dei dati più rilevanti

Come già evidenziato nella sezione di *analisi del problema* le caratteristiche che si potrebbero usare per allenare un modello sono tantissime, ed è probabile che molte correlazioni importanti non siano ancora state scoperte.

Ai fini di garantire semplicità e comprensione, nella fase di *feature selection* si è deciso di considerare per l'allenamento del modello poche caratteristiche e di tipo numerico.

Tali caratteristiche sono:

Feature	Descrizione
<b>Open</b>	Prezzo della prima transazione effettuata per un'azione in un determinato giorno di mercato.
<b>High</b>	Prezzo più alto raggiunto dall'azione durante la giornata di trading.
<b>Low</b>	Prezzo più basso a cui l'azione è stata scambiata durante la giornata.
<b>Close</b>	Prezzo dell'ultima transazione effettuata per l'azione nel giorno considerato. È il valore che vogliamo prevedere.
<b>Volume</b>	Numero totale di azioni scambiate durante la giornata.

Table 1: Feature selezionate e descrizione

Oltre alle specifiche, bisogna tenere conto della natura temporale di tali dati nella fase di scelta e valutazione del modello.

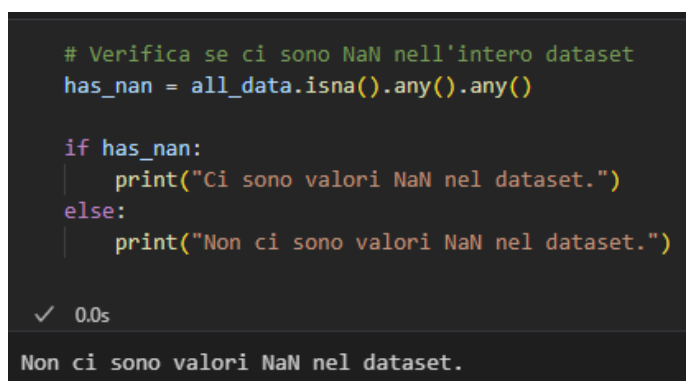


## 3 Preprocessing

Il *preprocessing* dei dati è una fase fondamentale per garantire l'efficacia del modello predittivo. I dati grezzi raccolti spesso contengono valori mancanti, anomalie e rumore che devono essere corretti o rimossi per evitare di compromettere la qualità delle previsioni. Questa sezione descrive i principali passaggi eseguiti durante il *preprocessing* dei dati.

### 3.1 Gestione dei valori mancanti

La presenza di valori mancanti nei dati storici potrebbe causare problemi durante l'addestramento del modello. Fortunatamente, almeno per gli stock che abbiamo deciso di utilizzare, le tabelle risultano complete.



```
# Verifica se ci sono NaN nell'intero dataset
has_nan = all_data.isna().any().any()

if has_nan:
    print("Ci sono valori NaN nel dataset.")
else:
    print("Non ci sono valori NaN nel dataset.")
```

✓ 0.0s

Non ci sono valori NaN nel dataset.

Figure 4: Controllo dei valori nan all'interno del dataset

C'è tuttavia un'osservazione interessante che si può fare, a differenza delle altre compagnie, TESLA non è stata quotata in borsa fino a luglio 2010. Nelle fasi future questo potrebbe comportare la necessità di gestire i dati mancanti, ma per ora ci limiteremo a tenerlo a mente.

## 3.2 Riduzione del rumore

I dati finanziari spesso presentano una notevole quantità di rumore dovuta a fluttuazioni casuali del mercato, eventi esterni e altre anomalie non rappresentative del trend generale. Per ridurre le imprecisioni che il rumore potrebbe introdurre, utilizzeremo una tecnica di *soothing* detta *media mobile*.

### 3.2.1 Media mobile

La media mobile (*Moving Average*) è una delle tecniche più semplici ed efficaci per ridurre il rumore nei dati di serie temporali. Calcolando la media di un certo numero di valori consecutivi, questa tecnica attenua le fluttuazioni locali senza modificare il trend generale:

$$MA_t = \frac{1}{n} \sum_{i=0}^{n-1} x_{t-i} \quad (1)$$

Dove  $MA_t$  rappresenta la media mobile al tempo  $t$ ,  $n$  è la finestra temporale utilizzata per calcolare la media, e  $x_t$  è il valore della serie temporale al tempo  $t$ . Nell'applicazione della formula, sarà necessario capire la finestra temporale migliore per i nostri dati.

### 3.2.2 Individuazione del rumore

Per capire se e quali feature dei dati necessitano di *soothing* useremo un approccio basato sull'osservazione delle *statistiche descrittive* dei dataset, in questo modo:

1. Controlliamo le statistiche sul dataset originale
2. Applichiamo la media mobile a tutte le feature
3. Confrontiamo le nuove statistiche con quelle vecchie

A seconda dei risultati ottenuti, decideremo se procedere o meno con la riduzione.

L'analisi verrà fatta su un solo dataset di riferimento (GOOGLE), assumendo che gli altri si comportino allo stesso modo.

	Open	High	Low	Close	Volume
count	3799.000000	3799.000000	3799.000000	3799.000000	3.799000e+03
mean	62.102565	62.726474	61.451896	62.079381	5.599243e+07
std	47.786109	48.305732	47.256177	47.756085	4.740387e+07
min	10.873247	11.028089	10.812405	10.929100	9.312000e+06
25%	22.870544	22.957443	22.609730	22.914805	2.728100e+07
50%	47.560089	47.783782	47.085312	47.392699	3.753400e+07
75%	94.861122	96.480250	93.660470	94.940829	7.292246e+07
max	206.380005	207.050003	202.809998	203.389999	5.923990e+08

Figure 5: Statistiche per GOOGLE prima di *moving average*

Una prima analisi su tutto il dataset rileva una varianza estremamente alta, confermando la non stazionarietà che ci si aspetterebbe dall'andamento di uno stock. Una varianza bassa avrebbe permesso di effettuare l'attenuazione del rumore su tutto il dataset, mentre ora sembra più corretto analizzare periodi di tempo più brevi, proviamo con i dati relativi a un solo anno.

	Open	High	Low	Close	Volume
count	252.000000	252.000000	252.000000	252.000000	2.520000e+02
mean	30.887441	31.171679	30.587030	30.890260	4.339637e+07
std	4.343829	4.397117	4.282566	4.346961	2.264566e+07
min	24.763233	24.923651	24.456843	24.871839	1.041200e+07
25%	27.233902	27.405029	27.043838	27.257441	3.138400e+07
50%	28.590358	28.918168	28.308380	28.578153	3.796200e+07
75%	33.959268	34.577779	33.675922	34.088924	4.767250e+07
max	39.554615	39.790261	39.217830	39.554610	2.571620e+08

Figure 6: Statistiche per GOOGLE nel 2015 prima di *moving average*

La varianza è sensibilmente diminuita, procediamo quindi con il confronto con i dati "derumorizzati", inizieremo con una finestra di 7 giorni.

	Open	High	Low	Close	Volume
count	252.000000	252.000000	252.000000	252.000000	2.520000e+02
mean	30.585838	30.872308	30.290240	30.594126	4.344942e+07
std	4.161919	4.216546	4.094634	4.158732	1.211131e+07
min	25.169392	25.428178	24.928002	25.218713	2.648000e+07
25%	27.295107	27.464294	27.104598	27.297547	3.625043e+07
50%	28.114029	28.353441	27.911436	28.130591	4.052547e+07
75%	33.407548	33.841271	33.042329	33.466589	4.668641e+07
max	38.530274	38.923755	38.152531	38.558467	1.020820e+08

Figure 7: Statistiche per GOOGLE nel 2015 dopo *moving average*

Confrontando media e varianza, notiamo come la riduzione del rumore non ha quasi per nulla intaccato il dataset, suggerendo una situazione poco rumorosa che quindi non necessita di *soothing*. L'unica eccezione è rappresentata dal campo **Volume** in cui invece la varianza è stata dimezzata. Possiamo notare bene l'efficacia che ha avuto l'algoritmo dai grafici relativi.

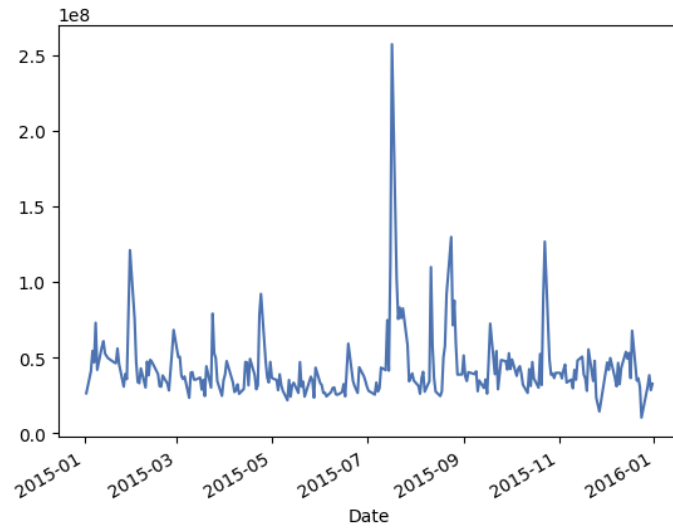


Figure 8: Grafico del volume prima di *Moving average*

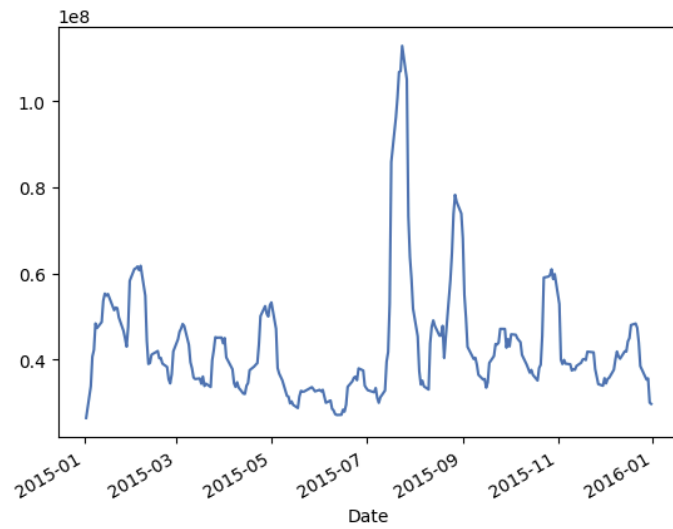


Figure 9: Grafico del volume dopo *Moving average*

Come previsto la differenza non è trascurabile, il rumore è stato notevolmente ridotto. Sono ancora presenti dei picchi "appuntiti", quindi si è deciso empiricamente di aumentare la finestra a 10; in questo modo ci si aspetta di avere il compromesso migliore tra riduzione del rumore e perdita di informazione.

### 3.2.3 Conclusione sulla riduzione del rumore

Alla luce delle informazioni emerse dallo studio del dataset, solo la feature **Volume** subirà *Moving Average* con una finestra di 10.

## 3.3 Accorpamento dei Dati

Una volta raccolti i dati, si è deciso di unire i dataset di ciascun titolo in un unico DataFrame per semplificare l'analisi successiva. Questo processo di unione è stato realizzato utilizzando la funzione di concatenazione di Pandas, che permette di combinare i dati di tutti gli stock in un singolo dataset, mantenendo la data come indice e le variabili come colonne. Il **Ticker**, che indica a quale titolo appartiene ciascun dato, è stato aggiunto come ultima colonna per identificare il titolo di riferimento per ogni record.

Il vantaggio di avere dati relativi a stock diversi durante la fase di addestramento del modello consiste nella possibilità di identificare trend comuni al mercato in generale piuttosto che a una singola azione. Questa caratteristica potrebbe però rendere il modello più efficiente a predire stock simili a quelli presenti nei dati di training (come in questo caso quelli del settore tecnologico). Un accorpamento del genere dei dati garantisce scalabilità e generalizzazione, consente infatti di aggiungere nuovi titoli azionari senza modificare il codice esistente.

Di seguito è riportato uno snippet di codice che mostra come i dati dei vari stock siano stati uniti utilizzando la funzione `concat` di Pandas:

```
import pandas as pd
import yfinance as yf

# Lista dei ticker da scaricare
tickers = ["AAPL", "MSFT", "GOOGL", "TSLA", "AMZN"]

# Funzione per scaricare i dati di ciascun ticker
def get_stock_data(ticker):
    data = yf.download(ticker, start="2010-01-01", end="2023-01-01")
    data['Ticker'] = ticker
    return data

# Creazione di una lista di dataset
dataset_list = [get_stock_data(ticker) for ticker in tickers]

# Unione dei dataset in un unico DataFrame
all_data = pd.concat(dataset_list, axis=0)

# Visualizza le prime righe del dataset unito
print(all_data.head())
```

## 3.4 Normalizzazione dei dati

### 3.4.1 Min-max scaling

Le variabili selezionate hanno scale di valori diverse, il che potrebbe influire negativamente sulle prestazioni del modello. Per garantire che tutte le variabili abbiano lo stesso peso durante l'addestramento, è stata applicata una normalizzazione *min-max*, che scala i dati tra 0 e 1:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$

Dove  $x$  rappresenta il valore originale,  $x'$  il valore normalizzato, e  $\min(x)$  e  $\max(x)$  indicano rispettivamente il minimo e il massimo della variabile.

### 3.4.2 Implementazione

Abbiamo applicato la normalizzazione separatamente per ogni ticker, utilizzando le seguenti caratteristiche:

- **Open:** Prezzo di apertura
- **High:** Prezzo massimo
- **Low:** Prezzo minimo
- **Close:** Prezzo di chiusura
- **Volume:** Volume di scambi

L'implementazione della normalizzazione è stata realizzata utilizzando la libreria `scikit-learn` e il metodo `MinMaxScaler`. Di seguito il codice utilizzato:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
all_data[columns_to_normalize] = scaler.fit_transform(all_data[columns_to_normalize])
```

## 4 Definizione del modello

### 4.1 Scelta del modello

La scelta del modello è una fase cruciale nello sviluppo di applicazioni di previsione finanziaria. Nel nostro caso, l'obiettivo è prevedere il prezzo futuro di un titolo azionario utilizzando dati storici su più variabili (*Open*, *High*, *Low*, *Close*, *Volume*). Dopo aver analizzato le caratteristiche dei dati e i requisiti del progetto, abbiamo deciso di adottare un approccio basato su reti neurali ricorrenti (RNN), in particolare le *Long Short-Term Memory* (LSTM). Questa scelta è motivata dalla capacità delle LSTM di catturare dipendenze a lungo termine nei dati sequenziali, come i trend e le stagionalità presenti nei mercati finanziari.

### 4.2 Criteri di Scelta del Modello

La selezione del modello è stata guidata dai seguenti criteri:

- **Natura sequenziale dei dati:** I dati storici dei prezzi delle azioni sono naturalmente organizzati in sequenza temporale. Modelli in grado di apprendere sequenze temporali, come le LSTM, sono particolarmente adatti a questo tipo di dati.
- **Rilevanza delle dipendenze temporali a lungo termine:** I movimenti di prezzo non sono influenzati solo dai valori recenti, ma anche da pattern che si manifestano su intervalli temporali più lunghi.
- **Robustezza agli outlier e al rumore:** Le LSTM, se opportunamente regolarizzate e addestrate su dati filtrati, possono gestire il rumore tipico dei dati finanziari meglio rispetto ai modelli tradizionali di regressione.
- **Generalizzazione su nuovi dati:** Un modello LSTM addestrato su dati di più titoli azionari può generalizzare meglio e fornire previsioni anche per titoli non presenti nel set di addestramento iniziale.

### 4.3 Modelli Considerati

Abbiamo confrontato diverse tipologie di modelli prima di scegliere definitivamente le LSTM:

- **Regressione Lineare e Modelli ARIMA:** Modelli tradizionali per dati temporali, adatti a serie stazionarie ma limitati nel catturare relazioni non lineari e dipendenze a lungo termine.
- **Random Forest e Modelli di Ensemble:** Buoni per problemi di classificazione e regressione, ma meno efficaci nei dati sequenziali rispetto ai modelli basati su reti neurali.
- **Reti Neurali Classiche (MLP):** Potenti ma incapaci di catturare informazioni temporali a lungo termine.

## 4.4 Creazione di serie temporali

Il modello *LSTM* richiede dati in forma di sequenze temporali. Per questo motivo, i dati normalizzati sono stati trasformati in sequenze di lunghezza predefinita (ad esempio, 60 giorni consecutivi), in modo che il modello possa apprendere le relazioni temporali tra le osservazioni:

- Ogni sequenza di 60 giorni è stata utilizzata per prevedere il valore dei giorni successivi.
- I dati sono stati divisi in set di **training** (80%) e **test** (20%).

## 4.5 Specifiche del modello

Diverse iterazioni sono state proposte per le specifiche del modello, fino ad arrivare a queste caratteristiche:

- Numero di neuroni - 512
- Tasso di dropout - 0.2
- Densità di previsione - 7 giorni

All'inizio era stata scelta una densità di previsione unitaria, e questo aveva portato a un modello estremamente efficiente, o per meglio dire *troppo* efficiente. Senza volerlo eravamo andati incontro a quella che viene definita "*lazy prediction*", un fenomeno per il quale il modello non fa altro che predire correttamente valori molto simili a quelli precedenti. Tra un giorno e l'altro, infatti, gli stock non tendono ad avere bruschi cambiamenti di valore, al punto che anche un essere umano potrebbe prevederne il cambiamento con un'accuratezza, sulla carta, altissima.

Aumentando la densità, si costringe il modello a cercare effettivamente dei pattern e quindi a "prevedere" a tutti gli effetti.



## 5 Valutazione del modello

### 5.1 Fase di allenamento

Per l'addestramento sono state previste 20 epoche, con l'errore quadratico medio come funzione di loss.

```
Epoch 1/20  
464/464 ————— 56s 118ms/step - loss: 0.0044 - val_loss: 7.2868e-04  
Epoch 2/20  
464/464 ————— 55s 118ms/step - loss: 5.5862e-04 - val_loss: 5.8665e-04  
Epoch 3/20  
464/464 ————— 56s 120ms/step - loss: 4.2703e-04 - val_loss: 5.7853e-04  
Epoch 4/20  
464/464 ————— 55s 118ms/step - loss: 4.2160e-04 - val_loss: 5.2211e-04  
Epoch 5/20  
464/464 ————— 54s 117ms/step - loss: 3.7816e-04 - val_loss: 5.1372e-04  
Epoch 6/20  
464/464 ————— 55s 118ms/step - loss: 3.6550e-04 - val_loss: 5.2990e-04  
Epoch 7/20  
464/464 ————— 56s 120ms/step - loss: 3.4939e-04 - val_loss: 4.9868e-04  
Epoch 8/20  
464/464 ————— 57s 123ms/step - loss: 3.1394e-04 - val_loss: 5.4409e-04  
Epoch 9/20  
464/464 ————— 57s 124ms/step - loss: 2.9579e-04 - val_loss: 4.6921e-04  
Epoch 10/20  
464/464 ————— 56s 121ms/step - loss: 2.9868e-04 - val_loss: 5.0081e-04  
Epoch 11/20  
464/464 ————— 55s 119ms/step - loss: 2.9554e-04 - val_loss: 4.9700e-04  
Epoch 12/20  
464/464 ————— 56s 120ms/step - loss: 2.9099e-04 - val_loss: 5.1386e-04  
Epoch 13/20  
...  
Epoch 19/20  
464/464 ————— 54s 117ms/step - loss: 2.6417e-04 - val_loss: 4.7437e-04  
Epoch 20/20  
464/464 ————— 56s 120ms/step - loss: 2.5842e-04 - val_loss: 4.6412e-04
```

Figure 10: Fase di addestramento

Il valore di *loss* è diminuito gradualmente durante l'allenamento, indicando che il modello è diventato progressivamente più bravo a fare previsioni sul training set. Nonostante lo stesso non si possa dire del *loss di validazione* (che non è diminuito gradualmente), il trend decrescente osservato fa comunque ben sperare.

### 5.2 Fase di valutazione

Per verificare l'efficienza del modello, come prima cosa confrontiamo i grafici dei valori reali e previsti per il training set.

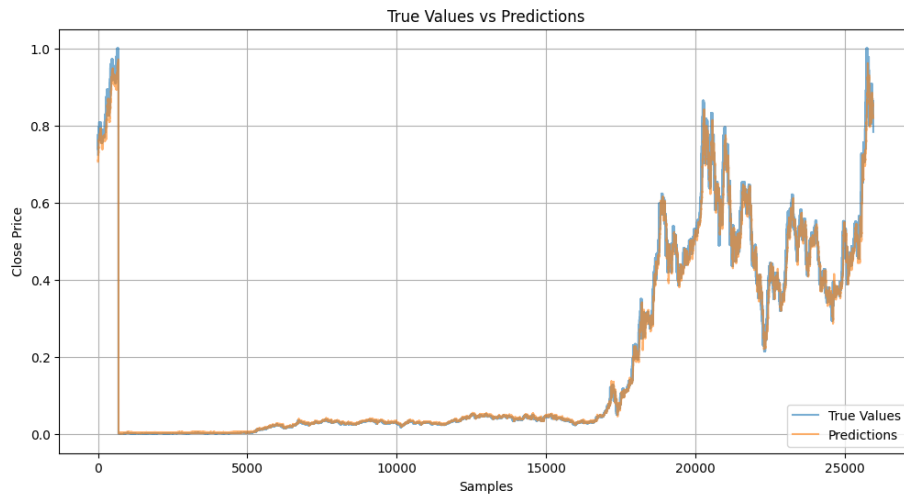


Figure 11: Confronto fra Valori Reali e Predizioni nel test set

A primo acchitto sembrerebbe che il modello faccia previsioni perfette, ma ancora una volta un comportamento del genere era più che aspettato: la densità di previsione, nonostante aumentata, è ancora troppo bassa per notare delle imperfezioni su un lasso di tempo così grande. Se andiamo a zoommare su un periodo temporale di sette giorni le differenze saranno evidenti.

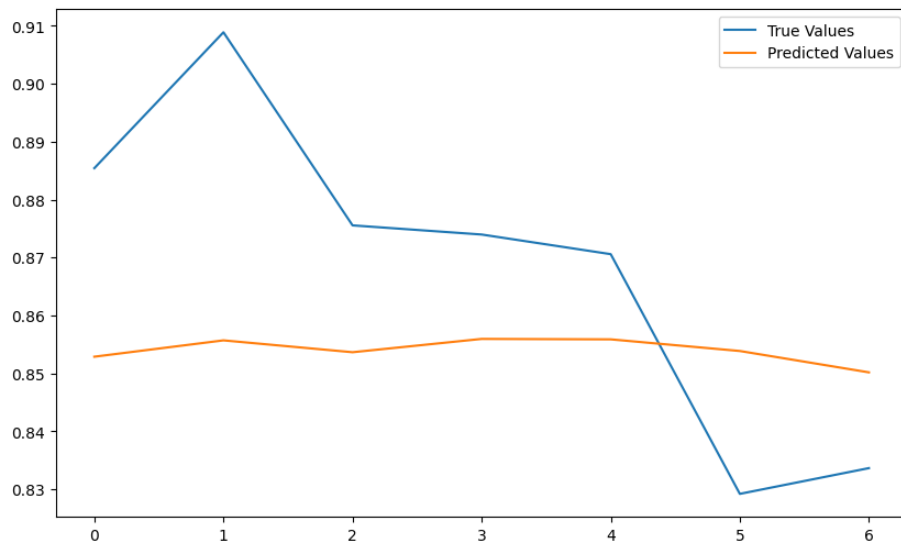


Figure 12: Confronto fra Valori Reali e Predizioni in una finestra di 7 giorni

Questo però non significa che il modello sia inefficiente, solo che bisogna fare ulteriori verifiche.

### 5.3 Verifica matematica dell'efficienza

Il grafico è utile per visualizzare la differenza tra valori predetti e reali, ma esistono indici appositi per calcolarla matematicamente.

Dato che i valori di *loss* individuati in fase di addestramento potrebbero indicare overfitting del modello e che gli stock appartenenti alla stessa categoria (tech) potrebbero seguire trend simili, è opportuno che i test siano effettuati su titoli di natura diversa da quelli presenti nel dataset di training. I valori successivi si riferiscono a *NFLX* (Netflix) e gli indici calcolati sono:

- **Errore quadratico medio (MSE):** 0.0002
- **Errore assoluto medio (MAE):** 0.0083
- **R<sup>2</sup> Score:** 0.9950
- **Variazione percentuale media:** 9.91%
- **Variazione percentuale massima:** 835.27%
- **Variazione percentuale minima:** 0.00%

I primi tre indicatori, per quanto promettenti, sono poco significativi a causa della bassa densità di previsione in relazione alla grandezza del dataset. Molto più interessante sono invece gli indici percentuali, che ci danno una visione più onesta del quadro generale. La **Variazione percentuale media** in particolare, con un valore che si aggira sul 10%, indica che il modello non gode di precisione impeccabile ma è ultimamente accettabile.

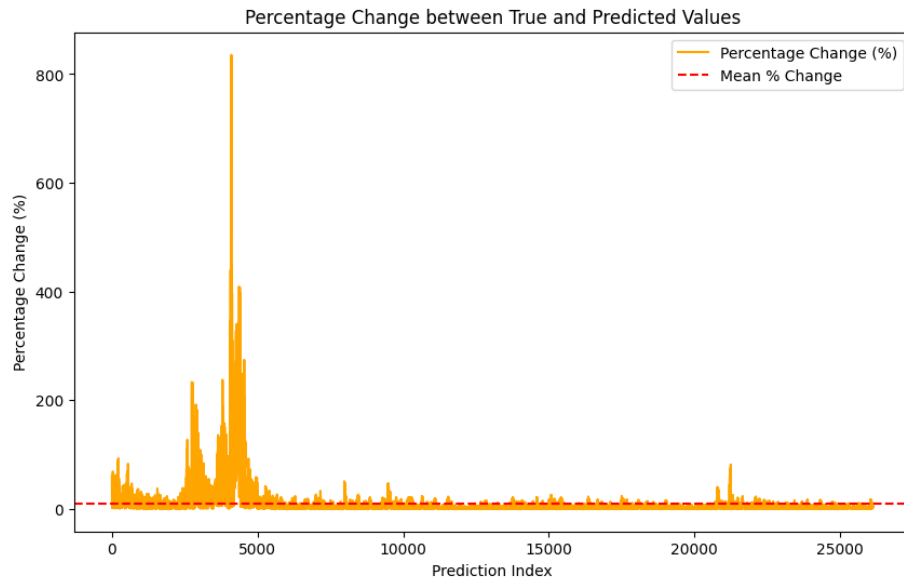


Figure 13: Percentage Change Graph

## 6 Considerazioni finali

Più che alla creazione di un modello predittivo dello *stock market* che funzioni bene, questo progetto mirava a scavare, seppur superficialmente, in quello che è il mondo del mercato azionario, per comprenderne la natura e il funzionamento, e capire come un'intelligenza artificiale potesse comportarsi nel cercare di prevederlo. Persone molto più preparate di me si sono gettate nella mia stessa impresa e hanno fallito, io mi ritengo soddisfatto di aver terminato con successo questo *esercizio di stile*.