



UNIVERSITÀ DEGLI STUDI DI SALERNO

**Dipartimento di Scienze Economiche e Statistiche**

Corso di Laurea Triennale in Statistica per i Big Data

Tesi in

*Progettazione del Software*

**ANALISI DEI DATI FINANZIARI  
IN PYTHON:  
LE AZIENDE PETROLIFERE  
DURANTE IL CONFLITTO  
RUSSO-UCRAINO**

Relatore:

Ch.mo Prof. Francesco Orciuoli

Candidato:

Francesco Landi

matr. 0212800525

**ANNO ACCADEMICO 2021/2022**



# INDICE

<b>INTRODUZIONE</b>	<b>5</b>
<b>CAPITOLO 1 - L'ANALISI TECNICA</b>	<b>6</b>
1.1 LE BASI	6
1.1.1 IL PREZZO	6
1.1.2 IL VOLUME	6
1.1.3 L'OPEN INTEREST	7
1.1.4 IL TREND	7
1.2 ANALISI TECNICA VS. ANALISI FONDAMENTALE	8
1.3 LA TEORIA DI DOW	8
<b>CAPITOLO 2 – L'ANALISI GRAFICA</b>	<b>10</b>
2.1 INTRODUZIONE	10
2.2 LE FORMAZIONI GRAFICHE	10
2.2.1 TRENDLINES	11
2.2.2 SUPPORTI E RESISTENZE	12
2.3 LE MEDIE MOBILI	13
2.3.1 MEDIE MOBILI SEMPLICI	13
2.3.2 MEDIE MOBILI ESPONENZIALI	14
2.4 INDICATORI E OSCILLATORI	16
2.4.1 LA DIFFERENZA TRA INDICATORE E OSCILLATORE	16
2.4.2 IL MACD	16
2.4.3 LE BANDE DI BOLLINGER E IL BANDWIDTH	17
2.3.4 IL RELATIVE STRENGTH INDEX (RSI)	18
2.3.5 LO STOCASTICO	19
2.5 DOPPIO TOP E DOPPIO BOTTOM	20
2.6 TESTA E SPALLE	21
2.7 TRIANGOLI	22
<b>CAPITOLO 3 - L'ANALISI ALGORITMICA IN PYTHON</b>	<b>23</b>
3.1 INTRODUZIONE	23
3.2 LE BASI DI PYTHON	23
3.3 LE API E I DATI FINANZIARI	25
3.4 LE LIBRERIE NECESSARIE	26
3.4.1 NUMPY	27
3.4.2 PANDAS	27
3.4.3 YFINANCE	29
3.4.4 MATPLOTLIB	29
3.5 ANALISI DI SERIE TEMPORALI	31
3.5.1 LE COMPONENTI DELLE SERIE TEMPORALI	31

3.5.2 LA STAZIONARIETÀ	31
3.6 VISUALIZZAZIONE DELLE SERIE TEMPORALI	32
<b>CAPITOLO 4 – ESEMPIO PRATICO SU DIVERSI TITOLI AZIONARI</b>	<b>34</b>
4.1 INTRODUZIONE	34
4.2 ANALISI DI MEDIO PERIODO	35
<b>CONCLUSIONE</b>	<b>44</b>
<b>APPENDICE A – CODICE PYTHON PER CREARE I GRAFICI</b>	<b>45</b>
A.1 LE CLASSI DA CUI EREDITARE LE FUNZIONI	45
A.2 BANDE DI BOLLINGER	47
A.3 MEDIE MOBILI	49
A.4 FASCIO DI MEDIE MOBILI	50
A.5 MACD	52
A.6 RSI	54
A.7 STOCASTICO	56
A.8 SUPPORTI E RESISTENZE	58
A.9 TRENDLINES	61
<b>BIBLIOGRAFIA</b>	<b>63</b>
<b>SITOGRAFIA</b>	<b>64</b>
<b>RINGRAZIAMENTI</b>	<b>65</b>

## INTRODUZIONE

In questa tesi vengono introdotti alcuni aspetti dell'analisi tecnica, cioè lo studio dell'andamento dei prezzi nei mercati finanziari utilizzando metodi grafici e statistico-matematici. Questo ramo della finanza è immenso e in continuo sviluppo, non si vuole quindi ridurla a quanto detto in questa tesi anzi si vuole far notare a chi legge quanto può essere potente e interessante. Con l'avvento dei calcolatori moderni e di internet si è creato un mondo nuovo di possibilità, una di queste è quella del *trading* online. Operazioni accessibili solo ai facoltosi attraverso i servizi bancari finalmente lo diventano a tutti in ogni angolo del globo. Grazie a ciò, materie come la finanza e quindi l'analisi tecnica sono diventate sempre più diffuse e studiate.

Nel capitolo primo si introducono *la teoria di Dow* che è uno dei pilastri cardine di questa materia, e si introducono concetti basilari di finanza. Questi concetti saranno utili per la lettura dei grafici e per gli studi successivi. Inoltre viene spiegata la differenza con un altro tipo di analisi, quella fondamentale. Questa differenza non è da sottovalutare, però c'è anche da tenere a mente che per riuscire ad interpretare tutti i possibili esiti non può bastare solo l'analisi tecnica e si dovrà ricorrere a quella fondamentale.

Il capitolo secondo descrive l'analisi grafica e cioè quella parte di questa materia che si basa su ciò che viene visto e rappresentato sul grafico, vengono quindi introdotte le basi come ad esempio i supporti e le resistenze, le trendlines, i massimi e i minimi. Questi strumenti sono fondamentali all'analisi grafica, tutte le formazioni successive viste nella tesi si baseranno sempre su di essi. Analizzerò anche famose formazioni grafiche come i *triangoli* o alcune più sporadiche come il *testa e spalle*, queste formazioni di tanto in tanto si vengono a formare e danno a chi le vede un'idea chiara sul possibile esito futuro del prezzo.

Nel capitolo terzo si introduce praticamente tutto quello che si è visto nei capitoli precedenti, utilizzando *Python*.

Questo è un linguaggio di programmazione relativamente recente, molto utilizzato nella scienza dei dati e nei calcoli matematici. Viene scelto da molti per delle fondamentali caratteristiche e cioè una curva di apprendimento non troppo inclinata e una comunità di programmatori molto attiva che mette a disposizione dei sotto-programmi chiamati moduli che permettono di fare praticamente tutto. Sono mostrate le basi di questo linguaggio e sono spiegate le maggiori librerie utilizzate in questa tesi: *NumPy*, *Pandas*, *yFinance* e *Matplotlib*.

Nel capitolo quarto si vuole applicare, quanto detto e visto nei capitoli precedenti, ad alcuni titoli azioni di aziende petrolifere di diversi paesi. Lo studio è mirato a vedere l'impatto del conflitto Russo-Ucraino sui diversi mercati scelti, precisamente il mercato italiano, quello europeo, quello russo e infine quello statunitense. Inoltre si vuole vedere se davvero l'analisi tecnica permetta di prevedere qualche risultato oppure affermare la sua soggettività come già fatto da molti.

Nell'appendice sono indicati i codici sorgenti dei programmi creati per generare i grafici nel capitolo quarto, questi programmi sono stati creati avendo in mente i punti forti di *Python* e della programmazione ad oggetti. Viene creata infatti una super classe da cui ereditano tutte le altre classi chiamata *DownloadDati*. Questa permetterà il download da Yahoo Finance oppure la lettura da file in formato tabulare. Inoltre tutte le classi create da essa avranno una funzione chiamata *plot* che permetterà la stampa dei grafici specifici.

## CAPITOLO 1 - L'analisi tecnica

### 1.1 Le basi

L'analisi tecnica è lo studio dei movimenti dei prezzi di mercato attraverso metodi grafici e statistici con lo scopo di prevederne le tendenze future.

Gli assunti dell'analisi tecnica sono tre:

1. **Il mercato sconta tutto:** nei prezzi sono incorporati già tutti i possibili fattori esterni quali politici, psicologici, macro-economici;
2. **I prezzi si muovono dentro un trend:** il trend è destinato a proseguire fino a segni di inversione;
3. **La storia si ripete:** per capire il futuro bisogna studiare il passato, perché è probabile che si ripeta.

#### 1.1.1 Il prezzo

Il prezzo indica la quotazione attuale del titolo, può essere rappresentato con diversi grafici. Il grafico più comune, ed anche quello più rappresentativo, è quello a *candele giapponesi*. Questa rappresentazione riassume i 4 livelli di prezzo più significativi cioè l'apertura, la chiusura, il massimo e il minimo. Quando il prezzo di apertura è minore del prezzo di chiusura allora la candela sarà di colore verde, sarà rossa per il contrario. Inoltre le candele avranno un *body* (corpo) e delle *shadow* (ombre) differenti che rappresenteranno l'indecisione nel mercato, maggiore saranno le *shadow* più c'è oscillazione di prezzo e quindi incertezza. Ci sono due prezzi, di apertura e di chiusura, perché il mercato ha degli orari di contrattazione ben precisi, la Borsa Italiana apre alle ore 09:00 e chiude alle 20:30.<sup>1</sup>

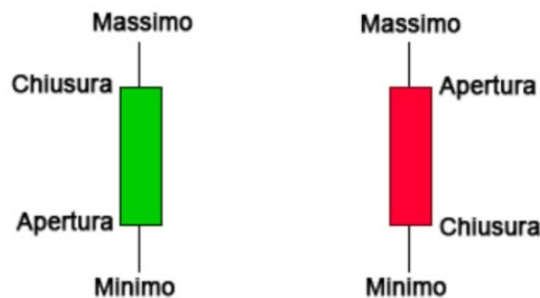


Figura 1: Candele giapponesi

#### 1.1.2 Il volume

Il volume è un altro fattore molto importante, esso rappresenta il numero totale dei contratti scambiati durante la giornata. Anch'esso viene rappresentato con un grafico a barre, inoltre come il prezzo può assumere colorazione verde se il volume di acquisto è maggiore di quello di vendita, mentre assumerà colorazione rossa se il volume di vendita sarà maggiore di quello di acquisto.

Il volume è un indicatore del sentimento del mercato molto importante, non tanto da solo, ma soprattutto se accompagnato ad altri fattori e/o figure grafiche. Come si vedrà nel seguito uno dei cardini dell'analisi tecnica, data dalla *teoria di Dow*, è proprio la conferma di un segnale grazie al volume.

---

<sup>1</sup> Da Money.it, <https://www.money.it/calendario-2023-borsa-italiana>, (ultima visita il 20 gennaio 2023)

### 1.1.3 L'open interest

L'*open interest* sui *futures*<sup>2</sup> rappresenta il totale dei contratti detenuti in un determinato mercato alla fine di ogni giorno. È il numero totale dei contratti di acquisto (*long*) o di vendita (*short*), non può essere la somma dei due perché ad ogni contratto di acquisto corrisponde uno di vendita e viceversa.

Rappresenta, quindi, la partecipazione al mercato da parte degli investitori.

### 1.1.4 Il trend

Per l'analisi tecnica e quindi quella grafica, il concetto fondamentale è quello del *trend*. Per trend si intende la direzione (non lineare) del mercato. Essendo il mercato "altalenante" e quindi formato da massimi e minimi, proprio questi indicano la direzione del trend:

- **Rialzista:** se ha massimi e minimi crescenti
- **Ribassista:** se ha massimi e minimi decrescenti
- **Laterale:** se ha massimi e minimi orizzontali

Il trend ha anche tre classificazioni diverse, è:

- **Primario** (lungo periodo): per estensioni superiori all'anno.
- **Secondario** (medio periodo): per estensioni da tre settimane fino all'anno.
- **Minore** (breve periodo): per estensioni inferiori alle tre settimane.

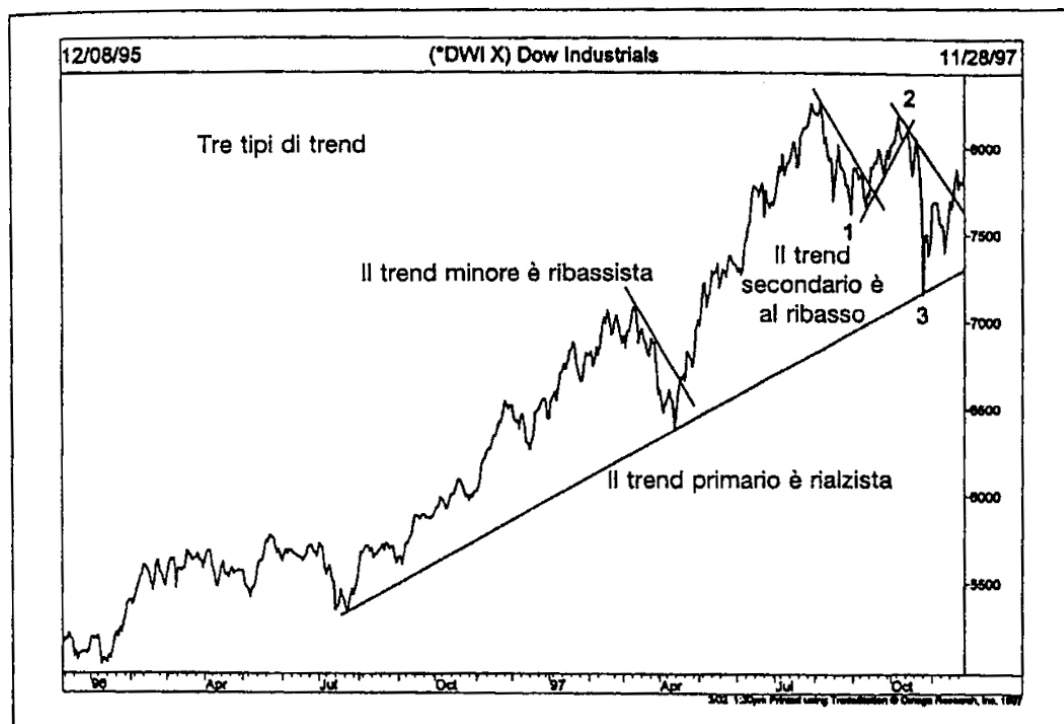


Figura 2: I tipi di trend, da John J. Murphy, *Analisi tecnica dei mercati finanziari*, (Milano: Hoepli, 2002), pagina 41. Stampa.

<sup>2</sup> Il *future* è un contratto a termine di controvalore standard, scambiato su mercati regolamentati, in cui l'acquirente s'impegna alla scadenza ad acquistare l'attività sottostante.

## 1.2 Analisi tecnica vs. analisi fondamentale

L'analisi fondamentale è differente da quella tecnica. La prima studia i motivi per il quale il mercato si comporta in un certo modo, la seconda ne studia gli effetti.

Quando ci sono grandi movimenti di mercato gli analisti fondamentali non spiegano cosa sta per succedere, coloro che usano l'analisi tecnica invece provano ad analizzare i possibili scenari che avverranno e le relative probabilità. Si può affermare che i movimenti dei prezzi fungono quindi da indicatore fondamentale e quindi che l'analisi tecnica in realtà per concetto include l'analisi fondamentale.<sup>3</sup>

L'analisi tecnica si può adattare a diverse finestre temporali, breve, medio e lungo periodo. Questa materia è duttile, infatti può essere adattata a tutti i tipi di *trading*, cioè l'investimento nei mercati finanziari, a partire dai *futures*, passando per le *commodities*<sup>4</sup> fino ai titoli azionari.

Una è però la critica mossa da accademici e scettici e cioè l'esistenza di un fattore di autoalimentazione<sup>5</sup>, un fattore quindi che faccia sì che l'analisi tecnica funzioni solo perché la gente creda esista.

Questa critica è posta in due modi diversi, personalmente credo che queste due definizioni si annullino a vicenda.

- a. Essendo ormai conosciuta, vengono fatti acquisti e vendite in base alle nozioni della materia.
- b. Le figure grafiche sono del tutto soggettive senza nessuna base matematica dietro;

La comunità accademica ha sviluppato la teoria del *random walk* che dice che il movimento dei prezzi è casuale e imprevedibile, andando in contrasto con il secondo assunto dell'analisi tecnica.

Questa teoria va in contrasto con la filosofia dell'analisi tecnica.

## 1.3 La teoria di Dow

Fu sviluppata dal giornalista, creatore del The Wall Street Journal, Charles Henry Dow. Egli creò anche il famoso *indice*<sup>6</sup> *Dow Jones* insieme allo statista finanziario Edward Jones<sup>7</sup>.

Questa teoria si basa su sei cardini:

- **Un trend persiste fino ad un segnale definitivo di inversione:** il cambiamento di trend può essere individuato solo al persistere di determinati movimenti in controtendenza.
- **La conferma del trend è definitiva solo se lo è quella del volume:** in un mercato rialzista, il volume dovrà essere in espansione durante i movimenti rialzisti e in contrazione durante movimenti ribassisti. Nel caso contrario in un mercato ribassista, il volume dovrà essere in espansione in movimenti al ribasso e in contrazione al rialzo.
- **Le medie devono confermarsi:** nella teoria iniziale Dow intendeva proprio l'indice Dow Jones, estendendo il discorso al giorno d'oggi il trend di un settore deve essere confermato anche dai trend dei settori a lui collegati.

---

<sup>3</sup> John J. Murphy, *Analisi tecnica dei mercati finanziari*, (Milano: Hoepli, 2002), p. 2

<sup>4</sup> Con il termine *commodity* si intende una materia prima (oro, argento, petrolio ecc.) o un'attività reale.

<sup>5</sup> *Ivi*, p. 11

<sup>6</sup> È la sintesi del valore del paniere di titoli azionari che esso rappresenta.

<sup>7</sup> Teoria di Dow, [https://it.wikipedia.org/wiki/Teoria\\_di\\_Dow](https://it.wikipedia.org/wiki/Teoria_di_Dow), (ultima visita il 20 novembre 2022)



- **Le medie scontano tutte le notizie:** le notizie una volta uscite sono già state incluse nelle quotazioni attuali degli indici.
- **Il mercato è composto da tre fasi:** la fase di accumulo, dove solo chi davvero crede nel mercato compra, la fase di partecipazione del pubblico e della massa chiamata *trend-following*<sup>8</sup>, la fase di distribuzione fatta da un'iniziale euforia e poi dalla vendita anche da parte di chi ha partecipato alla fase di accumulo.
- **Esistono tre tipi di trend:** quello primario, di lungo periodo da uno a più anni, quello secondario, di medio periodo da tre settimane a un anno, ed infine quello minore, di breve periodo fino a tre settimane.

---

<sup>8</sup> Tradotto letteralmente: che segue il trend.

## CAPITOLO 2 – L’analisi grafica

### 2.1 Introduzione

L’analisi tecnica è quindi formata da una parte grafica e da una parte statistico-matematica.

In questo capitolo ci si soffermerà sulla prima parte, sullo studio dei dati di apertura/chiusura del mercato per l’individuazione di figure o pattern<sup>9</sup>. Le figure sono “delle forme” che indicano la continuazione o l’inversione del trend, queste possono essere individuate in diversi periodi di tempo in base alla scelta del *timeframe*<sup>10</sup> e alla bravura di chi guarda il grafico.

La validità di una figura è data dalla sua durata, maggiore è meglio.

Inoltre le figure che si verificano in finestre temporali maggiori, come quelle mensili, e che successivamente vengono confermate in finestre temporali minori, hanno una validità ancora più forte da tenere in considerazione.

Con l’osservazione del comportamento nel tempo di azioni, *futures* e *commodities* si è capito che vengono a crearsi le medesime figure, perché dettate dalla psicologia umana.

Le figure si basano quindi sulla *teoria di Dow* ed in particolare sul terzo assunto e cioè la storia si ripete, questo permette agli analisti di provare a prevedere il futuro conoscendo il passato.

### 2.2 Le formazioni grafiche

Determinate formazioni grafiche vengono a determinarsi a seguito di alcuni movimenti di prezzo, con gli anni lo studio di questi ha portato alla conclusione che spesso determinate formazioni grafiche, sotto determinate condizioni, portano ad un certo risultato.

Quindi le figure sono state classificate in diverse categorie, in linea di massima si può dire che esistono formazioni rialziste, d’inversione e di continuazione.

Quelle rialziste solito vengono seguite da una tendenza che farà aumentare il prezzo.

Le formazioni di inversione di solito vengono seguite da un cambio di rotta dell’attuale tendenza.

Ed infine le formazioni di continuazione ci permettono di capire che la tendenza in atto continuerà a vivere.

Spesso le formazioni grafiche vengono individuate su una scala aritmetica, questa però attribuisce lo stesso valore all’incremento di un titolo da 50€ a 100€ e ad uno che passa da 500€ a 550€, benché la cifra sia la stessa la portata dell’incremento non è uguale. Viene usata quindi anche la scala logaritmica, che permette a chi individua il grafico di ottenere l’incremento percentuale e non più assoluto come con l’altra scala.<sup>11</sup>

Le formazioni grafiche fondamentali sono indicate nel seguito del capitolo.

---

<sup>9</sup> Un modello, uno schema che si ripete.

<sup>10</sup> La finestra temporale scelta per osservare i grafici, può essere ad esempio giornaliera, settimanale, mensile, annuale.

<sup>11</sup> Grafici analisi tecnica, <https://www.borsamercato.com/grafici-nellanalisi-tecnica/> (ultima visita il 20 novembre 2022)

### 2.2.1 Trendlines

Per creare una trendline si utilizzano i valori minimi e massimi raggiunti dal prezzo, si collegano tra di loro per indicare appunto delle linee “di sostegno” dei prezzi. Nel caso ci siano due trendlines parallele si può parlare di canale, se è rivolto verso l’alto allora parliamo di canale rialzista altrimenti di canale ribassista.



Figura 3: Trendlines (Titolo azionario: Eni S.p.A)

Nel grafico creato in Python si nota che le trendlines hanno formato un canale leggermente rialzista. Per quanto riguarda le fasi di rialzo però, oltre le trendline bisogna sempre confermare con il volume sempre seguendo la *teoria di Dow*, questo non vale sempre per le fasi di ribasso in quanto anche per mancanza di acquisto il mercato tende a calare. Il canale quindi è una figura importante, questo se rotto al rialzo ci permette di ottenere un primo *target price*<sup>12</sup> proiettando l’altezza di esso verso l’alto. Infatti analizzando sempre la figura generata in Python si nota che il prezzo tra ottobre e novembre è ritornato a toccare la trendline superiore del canale. Di tutti gli strumenti, benché basilari, le trendlines sono fondamentali per chiunque voglia studiare questa materia.

---

<sup>12</sup>Il prezzo a cui si prevede possa arrivare la quotazione in questione.

### 2.2.2 Supporti e resistenze

Come già detto i movimenti di prezzo vanno su e giù così da formare massimi e minimi, questi se classificati divengono supporti o resistenze. In effetti un supporto o una resistenza non è altro che una trendline orizzontale.

Per supporto si intende un minimo (o più minimi dello stesso livello) dove il prezzo rimbalza verso su e cioè dove la domanda di acquisto è più forte di quella di vendita.

Per resistenza invece si intendono un massimo (o più massimi dello stesso livello) dove il prezzo rimbalza verso giù e cioè dove la domanda di vendita è più forte di quella di acquisto.

Supporti e resistenze vengono “provate” ogni volta che il prezzo rimbalza su di esse, e cioè viene visto se ancora la domanda è più forte dell’offerta o viceversa. In caso la prova fallisca e cioè quando il supporto o la resistenza vengono rotti questi diventano l’opposto; una resistenza diviene supporto, un supporto diviene una resistenza.

Spesso le resistenze e i supporti si trovano in corrispondenza di prezzi a cifre tonde, questo è riconducibile alla psicologia umana che da sempre predilige la “precisione” del numero tondo rispetto ad altri.



*Figura 4: Supporti e resistenze (Titolo azionario: Petróleo Brasileiro S.A.)*

## 2.3 Le medie mobili

Una media mobile, semplicemente, è un filtro dei prezzi. Spesso capita che le quotazioni varino in maniera elevata, utilizzando una media mobile si prende in considerazione un solo elemento per ciascuna candela (di solito la chiusura) e lo si relaziona con i prezzi passati; il risultato è una linea piuttosto morbida che indica in maniera meno erratica l'andamento dei prezzi.

Spesso le medie mobili sono sopravvalutate e vengono utilizzate per scopi che vanno oltre la loro reale potenzialità, i tipi principali sono due: semplici ed esponenziali.

### 2.3.1 Medie mobili semplici

La media mobile semplice è la media aritmetica di una serie di prezzi in un determinato periodo scelto da chi la usa. Questa si muoverà:

- al di sotto dei prezzi in caso di tendenza rialzista;
- al di sopra dei prezzi in caso di tendenza ribassista;

*Tabella 1: Corrispondenza tra periodi di una media mobile semplice e arco temporale di una seduta borsistica (questa si svolge dal lunedì al venerdì)*

Periodo	Arco temporale
5 periodi	1 settimana
20 periodi	1 mese
60 periodi	3 mesi
120 periodi	6 mesi
240 periodi	1 anno

La formula di una media mobile semplice non è altro che una media aritmetica, quindi considerando un periodo di borsa come segue:

Seduta 1	Seduta 2	Seduta 3	Seduta 4	Seduta 5
15,00	15,10	15,03	14,87	15,20

La media mobile semplice a 5 periodi sarà calcolata come segue:

$$SMA = \frac{15.00 + 15.10 + 15.03 + 14.87 + 15.20}{5} = 15.04$$

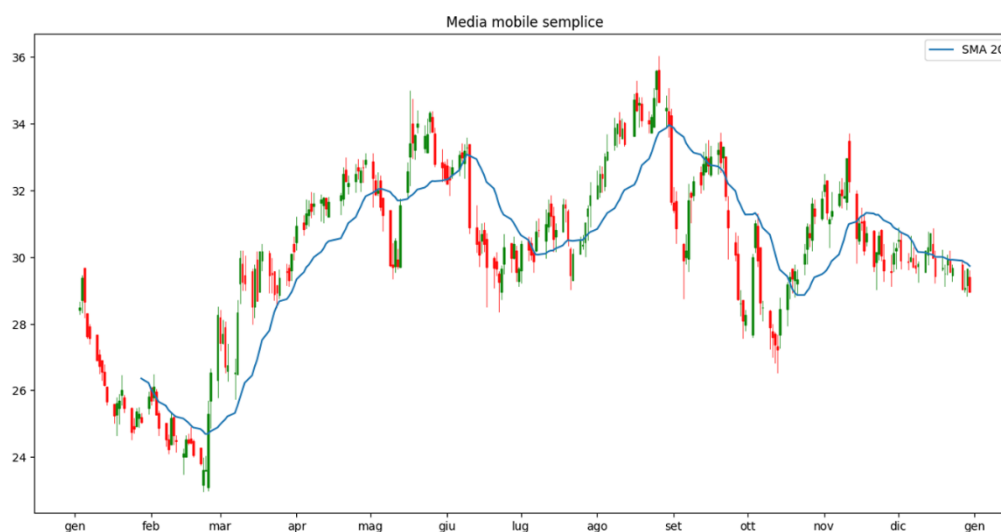


Figura 5: Media mobile semplice a 20 periodi (Titolo azionario: TotalErg S.p.A)

Le medie mobili semplici sono caratteristiche perché:

- filtrano i movimenti dei prezzi, rappresentando una linea con un andamento più morbido;
- reagiscono con ritardo a cambi di direzione;
- nelle inversioni di tendenza attraversano i prezzi;

### 2.3.2 Medie mobili esponenziali

La media mobile esponenziale riprende il funzionamento di quella semplice, l'unica differenza è che l'esponenziale attribuisce dei pesi maggiori cioè una maggiore importanza ai valori di prezzo più recenti rispetto a quelli passati.

La formula per il calcolo di una media mobile esponenziale è:

$$EMA = \alpha \times P_0 + (1 - \alpha) \times EMA_1$$

Dalla formula si evince che la media mobile è separata in due parti, la prima che è il valore del prezzo più recente ( $P_0$ ) moltiplicato per un fattore  $\alpha$ ; la seconda che è il valore della media mobile esponenziale precedente ( $EMA_1$ ) moltiplicato per il complemento ad uno del fattore  $\alpha$ , cioè la differenza tra uno e il fattore  $\alpha$ .

Il fattore  $\alpha$  è quindi fondamentale alla creazione della media mobile esponenziale e dipende dalla scelta del periodo di quest'ultima.

Tabella 2: Corrispondenza tra periodi di una media mobile esponenziale e fattore di una seduta borsistica (questa si svolge dal lunedì al venerdì)

Periodi media mobile	Fattore $\alpha$
5 periodi	0,333
20 periodi	0,095
50 periodi	0,039
100 periodi	0,020
200 periodi	0,010

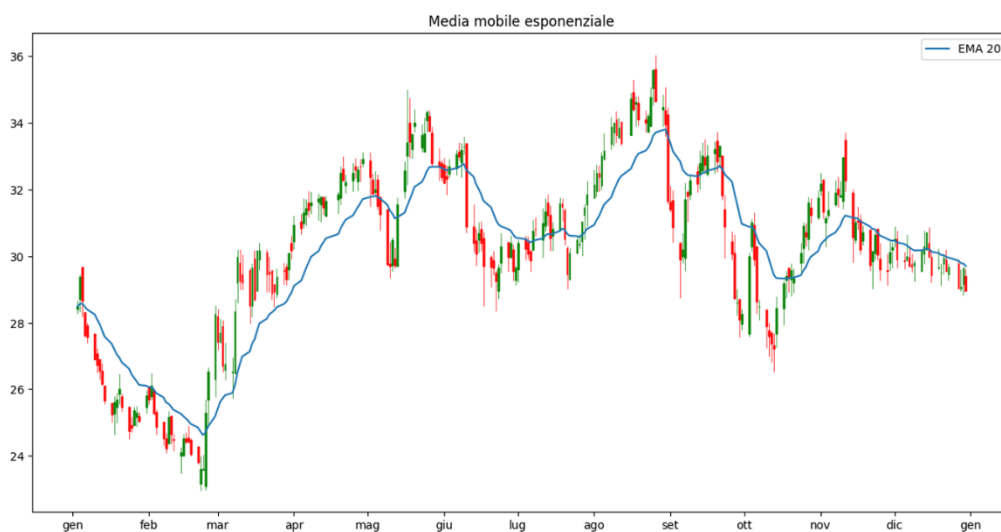


Figura 6: Media mobile esponenziale a 20 periodi, quindi con fattore  $\alpha = 0.095$  (Titolo azionario: TotalErg S.p.A.)

Con le medie mobili esponenziali si ha anche la possibilità di generare un fascio formato da medie mobili di lungo e di breve periodo. Il loro incrocio indicherà quindi un segnale di acquisto o di vendita.

Precisamente si ha un segnale di acquisto quando il fascio di medie mobili veloci incrocia, dal basso verso l'alto, quello delle medie mobili lente.

Un segnale di vendita invece si ha quando il fascio delle medie mobili veloci incrocia, dall'alto verso il basso, quello delle medie mobili lente.

Si possono fare tutte le configurazioni che colui che vede il grafico preferisce, nell'esempio sottostante i periodi delle medie mobili veloci sono rispettivamente 3, 5, 8, 10, 12, 15 mentre i periodi delle medie mobili lente sono rispettivamente 30, 35, 40, 45, 50, 60

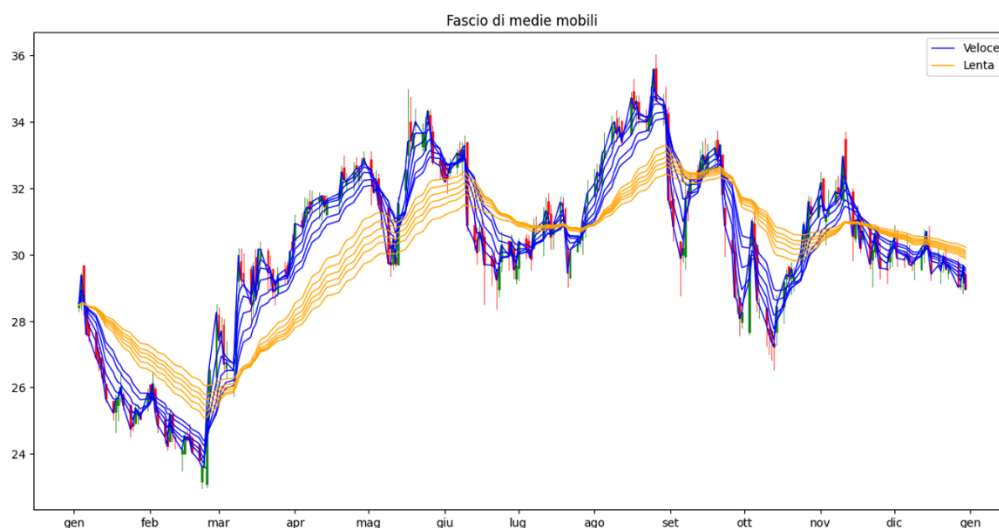


Figura 7: Fascio di medie mobili (Titolo azionario: TotalErg S.p.A)

## 2.4 Indicatori e oscillatori

### 2.4.1 La differenza tra indicatore e oscillatore

Benché spesso vengano identificati come uguali, indicatori e oscillatori posseggono alcune differenze che spesso comportano un utilizzo errato di essi.

Gli indicatori si muovono attraversando una linea di equilibrio che rappresenta da separatore per una fase rialzista e una ribassista, quindi non hanno alcun limite. Per quanto riguarda gli oscillatori hanno un massimo, un minimo, un centro e sono quindi limitati.

La categoria degli indicatori è quindi utilizzata per rilevare delle divergenze tra il prezzo e l'indicatore stesso, il risultato quindi è l'evidenziazione di zone in cui ci potrebbe essere un'inversione di trend.

La categoria degli oscillatori invece è utilizzata per rilevare le così dette zone di *ipercomprato* e *ipervenduto*, cioè zone estreme dove il prezzo assume valori anomali ottenuti in maniera repentina e questo potrebbe indicare una inversione di tendenza.<sup>13</sup>

### 2.4.2 Il MACD

Creato da Gerald Appel nel 1970 è l'acronimo di *Moving Average Convergence-Divergence* che in italiano si può tradurre come "convergenza-divergenza di medie mobili". Il funzionamento di questo indicatore si basa sulle medie mobili esponenziali, il loro incrocio indica che la differenza del loro valore sarà nulla e quindi anche il MACD avrà valore pari a zero.

Quando l'incrocio delle medie mobili avverrà dal basso verso l'alto, il MACD da negativo passerà a positivo, viceversa passerà da positivo a negativo.

Il MACD quindi viene utilizzato per comprare quando la linea del segnale incrocia quella del MACD al rialzo in zona negativa, viceversa per vendere quando la linea del segnale incrocia quella del MACD al ribasso in una zona positiva.



Figura 8: Grafico dei prezzi che include il MACD (Titolo azionario: Eni S.p.A)

---

<sup>13</sup> Da Bruno Moltrasio, <https://www.brunomoltrasio.eu/oscillatori-e-indicatori-qual-e-la-differenza/>, (ultima visita il 29 dicembre 2022)



### 2.4.3 Le bande di Bollinger e il Bandwidth

Ideate da John Bollinger dal quale prendono il nome, sono un canale dinamico che segue il movimento dei prezzi, l'ampiezza delle bande si adatta alla volatilità dei prezzi. Queste infatti sono composte da una linea superiore, una inferiore e una linea mediana. Le bande sono ottenute utilizzando la *deviazione standard*<sup>14</sup>, così facendo si è in grado di misurare il grado di volatilità dei dati. L'utilizzo delle bande di Bollinger è ottimale in condizioni di lateralizzazione o di tendenze di breve periodo, benché lo stesso Bollinger non abbia descritto situazioni assolute che indichino un segnale preciso ma si consiglia di badare alla così detta *camminata sulle Bande* cioè quando il prezzo tocca le bande e le supera o rimbalza. Questo strumento infatti non verrà utilizzato nel capitolo quarto proprio perché da solo è di difficile comprensione.

Un indicatore derivato, da utilizzare in combinazione, è il Bandwidth che misura l'ampiezza delle bande in termini relativi.

La formula del Bandwidth è la seguente:

$$\text{Bandwidth} = \frac{(\text{Banda superiore} - \text{Banda inferiore})}{\text{Linea mediana}}$$

Dove:

$$\text{Linea mediana} = \text{SMA20}^{15}$$

$$\text{Banda superiore} = \text{Linea mediana} + 2 * \text{Deviazione Standard}$$

$$\text{Banda inferiore} = \text{Linea mediana} - 2 * \text{Deviazione Standard}$$

Il Bandwidth è utile quindi per individuare i momenti di bassa volatilità, per individuare l'inizio e la fine di un trend e inoltre intercettare i momenti di espansione della volatilità.

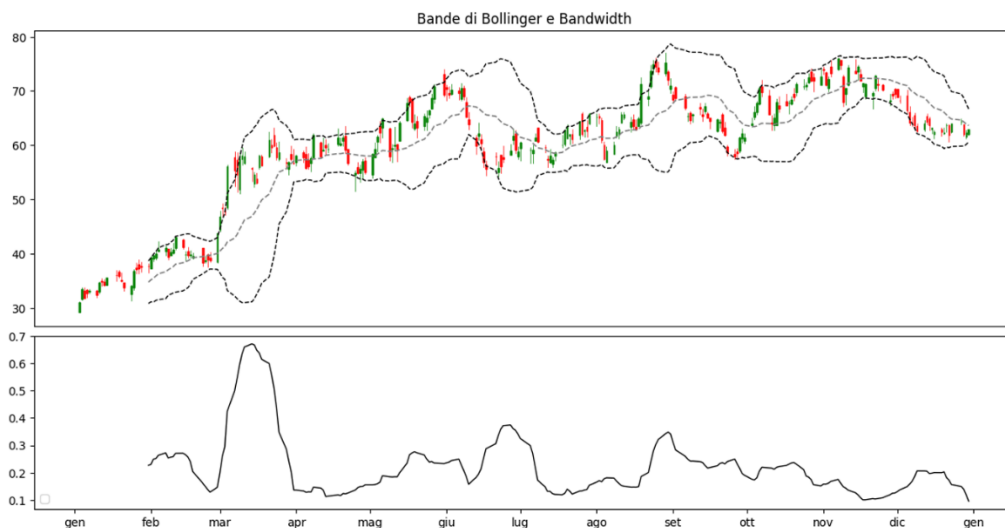


Figura 9: Grafico dei prezzi che include bande di Bollinger e Bandwidth (Titolo azionario: Occidental Petroleum Corporation)

<sup>14</sup> È la radice quadrata del tasso di variabilità (varianza).

<sup>15</sup> Media mobile semplice a 20 periodi.

### 2.3.4 Il Relative Strength Index (RSI)

Ideato da Welles Wilder, differente dall'indicatore di forza relativa chiamato *Relative Strength*, questo oscillatore misura la forza relativa delle chiusure rialziste rispetto a quelle ribassiste.

Secondo la concezione originale questo oscillatore prevede una finestra temporale a 14 periodi e si calcola utilizzando la media delle variazioni al rialzo e la media in valore assoluto delle variazioni al ribasso, ha la seguente formula:

$$RSI = \left( \frac{U}{U + |D|} \right) * 100$$

Dove:

$U$  = media delle variazioni di prezzo al rialzo

$D$  = media delle variazioni di prezzo al ribasso

La moltiplicazione per 100 è utile per normalizzare il risultato, questo infatti sarà compreso tra 0 e 100. L'oscillatore ha delle così dette zone di *ipercomprato* e *ipervenduto*, è sconsigliabile usare dei livelli comuni a tutti i titoli ma si può dire che la zona di ipercomprato è a 70, quella di ipervenduto è a 30. Conviene sempre fare qualche prova e adattare queste zone al titolo azionario specifico, in particolare nell'esempio successivo queste zone saranno a 75 e 35. Per ipercomprato si intende una zona che viene raggiunta dopo una serie di prezzi al rialzo, per ipervenduto invece è una zona che viene raggiunta dopo una serie di prezzi al ribasso.

Il funzionamento è semplice, ci indica di comprare quando l'oscillatore sta per uscire dalla zona di ipervenduto e di vendere quando sta per uscire da quella di ipercomprato.

È un oscillatore che mostra molti falsi positivi.



Figura 10: Grafico dei prezzi che include il RSI (Titolo azionario: British Petroleum plc)

### 2.3.5 Lo Stocastico

È un oscillatore creato da George Lane, il suo valore dipende dalla chiusura di prezzo più recente rispetto a un massimo e a un minimo dei prezzi che risiedono nella finestra temporale scelta. In una tendenza in salita lo Stocastico si avvicinerà al limite massimo 100, in presenza di prezzi in discesa tenderà ad avvicinarsi al valore minimo 0. Rispetto al RSI è molto simile, ma leggermente più complesso.

Questo oscillatore è formato dallo stocastico veloce e da quello lento, e il suo utilizzo è per lo più legato all'applicazione di strategie di *trend following* ma anche di inversione di tendenza. Funziona meglio in situazioni di assenza di trend.

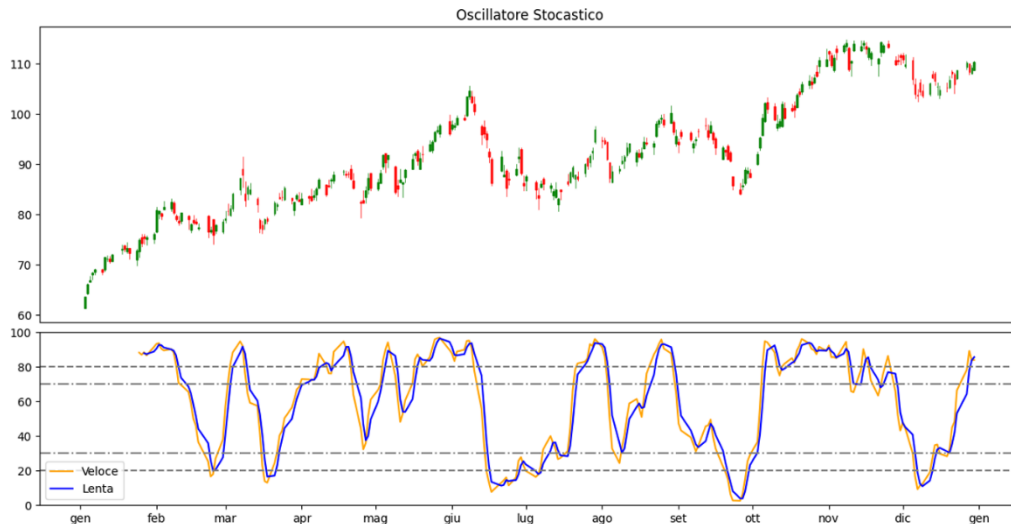


Figura 11: Grafico dei prezzi che include lo Stocastico (Titolo azionario: Exxon Mobil)

Come si può notare dal grafico, questo oscillatore come l'RSI, può creare molti falsi positivi. Infatti nel capitolo quarto è usato di meno rispetto ad altri strumenti, anch'esso infatti deve essere accompagnato da altri segnali e strumenti per evitare errori di valutazione.

Le zone di ipercomprato e ipervenduto sono due, quelle per lo stocastico veloce è a 80-20 mentre quello per lo stocastico lento è a 70-30.

Questo oscillatore quindi ci indica di comprare quando lo stocastico veloce incrocia quello lento nella zona di *ipervenduto* ed infine esce dalla zona stessa.

Ci indica di vendere quando lo stocastico veloce incrocia quello lento nella zona di *ipercomprato* ed esce dalla zona estrema.

## 2.5 Doppio top e doppio bottom

Il doppio top è una formazione grafica di tipo ribassista, questa come dal nome è formata da due top o massimi allo stesso livello di prezzo. Il volume che accompagna questa formazione è quasi speculare al prezzo.

Il doppio bottom invece è una formazione rialzista, è l'opposto del doppio top ed è quindi formata da due minimi.

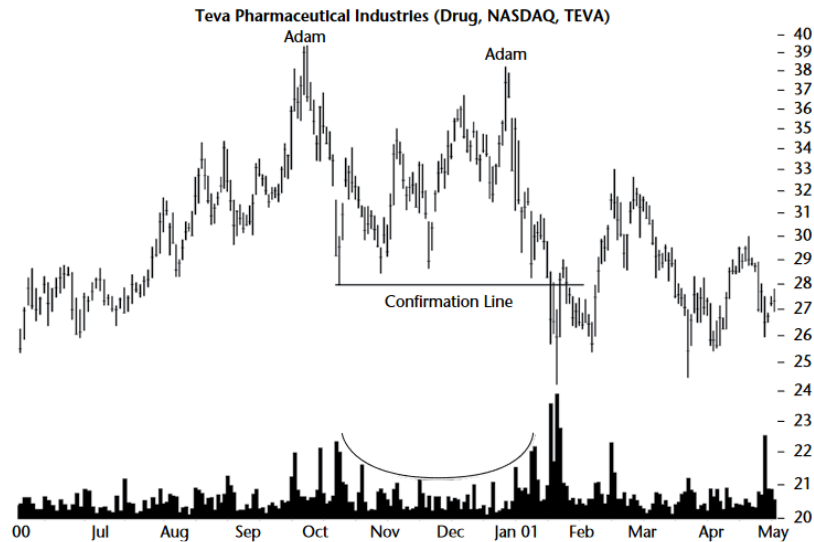


Figura 12: Doppio top, da Thomas N. Bulkowski, *Encyclopedia of Chart Patterns*, (Hoboken: John Wiley & Sons, Inc., 2021), p. 380. Ebook.

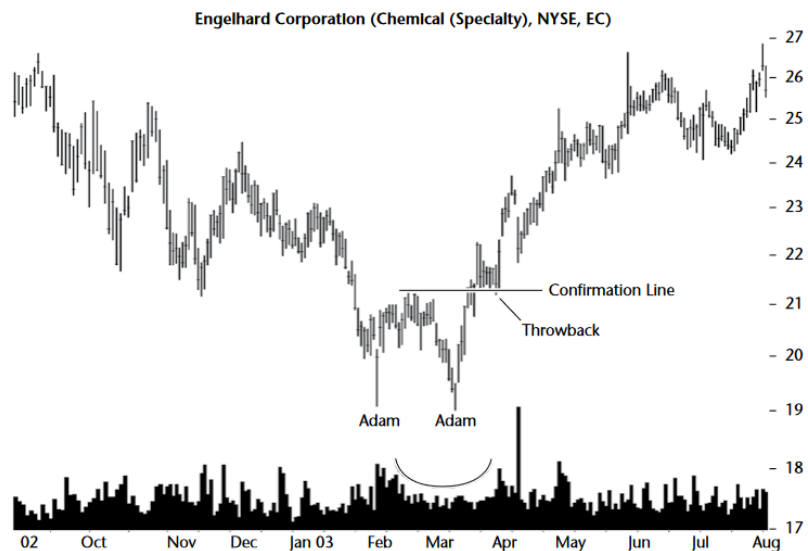


Figura 13: Doppio bottom, da Thomas N. Bulkowski, *Encyclopedia of Chart Patterns*, (Hoboken: John Wiley & Sons, Inc., 2021), p. 380. Ebook.

Spesso capita che esistano molte accezioni di ogni formazione. Essendo solo una breve introduzione a questa materia negli esempi di doppio top e doppio bottom sono stati rappresentati solo i così detti *Eve & Eve* perché questa variante viene considerata dalla letteratura come “classica”. Infatti questo è il pattern ideale, purtroppo però nel mondo reale spesso ciò non accade. Esistono quindi altre varianti e riguardano le combinazioni di “Adamo ed Eva” quindi vengono chiamate *Adam & Adam*, *Eve & Adam*, *Adam & Eve*. Queste sono più complesse.

## 2.6 Testa e spalle

Una delle formazioni grafiche più conosciute è il così detto *testa e spalle*, questa formazione infatti viene spesso “vista” da analisti anche se la sua forma non è proprio come dovrebbe essere.

È una formazione di inversione e rispetto al doppio top o doppio bottom è meno comune, soprattutto nella sua forma teorica.

La sua individuazione dipende quindi se aspettiamo un ribasso o un rialzo.



Figura 12: Testa e spalle da Investopedia, <https://www.investopedia.com/terms/h/head-shoulders.asp> (ultima visita il 24 novembre 2022)

Le fasi di sviluppo di un *testa e spalle* sono le seguenti<sup>16</sup>:

1. Lo sviluppo della spalla sinistra deve essere su un trend rialzista, ci dovrà essere un minimo che completa la spalla che si appoggia sulla trendline della crescita dei prezzi.
2. La formazione della testa segue quello della spalla, dovrà esserci un massimo maggiore rispetto alla spalla.
3. Fino ad ora nulla può far notare la presenza di un testa e spalle, il primo segnale sarà la rottura della trendline della crescita dei prezzi. Si ha bisogno però di una conferma.
4. La conferma di inversione del trend avverrà quando si avrà un movimento concorde ma insufficiente, e cioè alla creazione della spalla destra che avrà un massimo minore della spalla sinistra.
5. La così detta *neckline* letteralmente “linea del collo”, sarà la linea tracciata tra i minimi della testa e delle spalle. La rottura di essa è un segno definitivo dell’inversione del trend.

<sup>16</sup> Maurizio Mazziero, Guida all’analisi tecnica, (Milano: Hoepli, 2014), p. 191

## 2.7 Triangoli

I triangoli sono un tipo di formazione di continuazione ma secondo alcuni criteri anche di inversione, ne esistono diverse sfaccettature infatti i triangoli possono essere simmetrici, ascendenti o discendenti. Ognuno di essi ha caratteristiche diverse.

- *Triangolo simmetrico*: si sviluppa simmetricamente, i lati di esso sono della stessa inclinazione e convergono fra loro formando un triangolo isoscele o simmetrico



Figura 13: Triangolo simmetrico da IG Italia, <https://www.ig.com/it/il-triangolo> (ultima visita il 25 novembre 2022)

- *Triangolo ascendente*: si sviluppa in maniera asimmetrica con una tendenza ad avere quotazioni che rimangono al di sotto di una resistenza per lo più orizzontale, il lato obliquo inferiore fa da supporto e idealmente viene a formarsi un triangolo rettangolo.
- *Triangolo discendente*: è lo speculare del triangolo ascendente, anch'esso si sviluppa asimmetricamente ma con una tendenza ad avere quotazioni che rimangono sopra ad un supporto orizzontale, il lato obliquo superiore fa da resistenza e idealmente viene a formarsi un triangolo rettangolo.

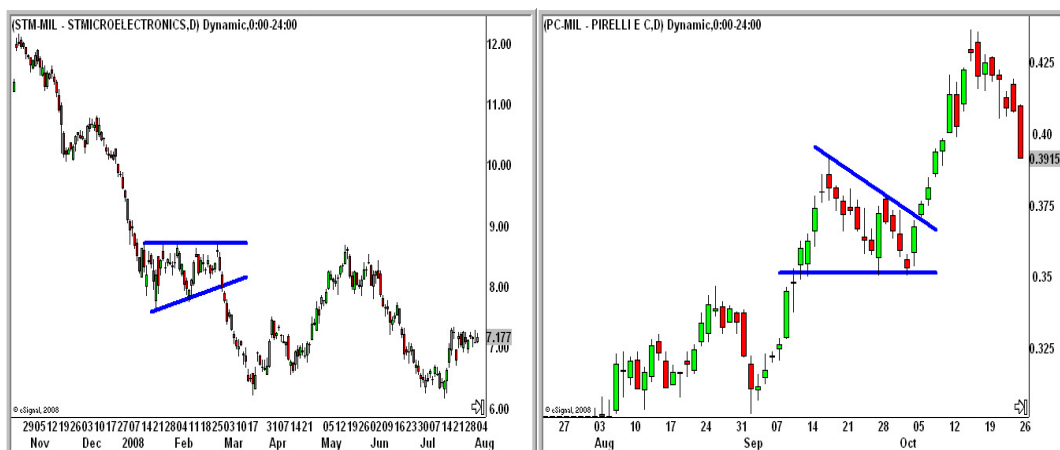


Figura 14: Triangolo ascendente a sinistra, triangolo discendente a destra da IG Italia, <https://www.ig.com/it/il-triangolo> (ultima visita il 25 novembre 2022)

Il segnale da ricercare è la rottura di uno dei lati del triangolo.

## CAPITOLO 3 - L'analisi algoritmica in Python

### 3.1 Introduzione

Python è un linguaggio di programmazione ad alto livello, cioè scritto in un linguaggio “naturale” facilmente comprensibile per l'uomo e tradotto da un altro programma per essere compreso dai computer. Nasce il 20 febbraio del 1991.

Benché possa sembrare vecchio in realtà rispetto ad altri è abbastanza recente e negli ultimi anni soprattutto sta avendo un “boom” di nuovi programmatori, questo perché è molto potente e ha una linea di apprendimento poco inclinata. Python ha tre principali obiettivi: dinamicità, semplicità e flessibilità.<sup>17</sup>

Le strategie algoritmiche non sono nate come le conosciamo oggi ma hanno avuto ben tre diverse fasi<sup>18</sup>:

1. Tra il 1980 e 1990, si utilizzavano modelli e segnali matematici derivanti dalla ricerca accademica.
2. Agli inizi del 2000, si utilizzavano invece lavori pionieristici come quelli di Eugene Fama<sup>19</sup> e Kenneth French<sup>20</sup> che includevano algoritmi utili ad identificare fattori di rischio per cercare opportunità di arbitraggio nei mercati<sup>21</sup>.
3. Ad oggi si utilizza il Machine Learning, che fa parte dell'intelligenza artificiale, per prevedere i possibili esiti del mercato.

### 3.2 Le basi di Python

Python come già introdotto è un linguaggio ad alto livello, quindi quando eseguiamo un programma, anche chiamato *script*, con esso il codice scritto viene scansionato “parola per parola” e ognuna di essa viene analizzata dentro una struttura logica ad albero che rappresenta il programma. Questa struttura viene poi trasformata in *bytecode*, un linguaggio intermedio che si interpone tra il linguaggio macchina e il linguaggio Python, ed infine viene eseguita nella *PVM* cioè la *macchina virtuale Python*.<sup>22</sup>

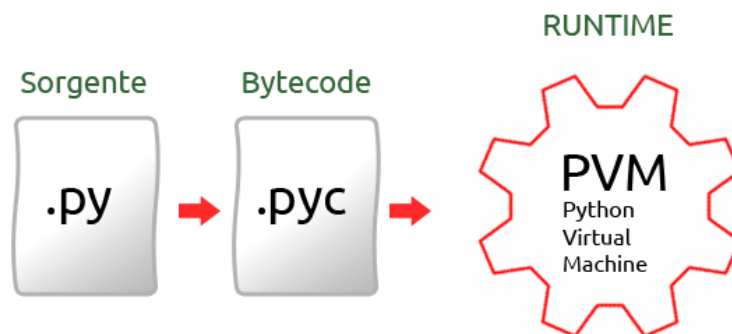


Figura 15: Esecuzione di un file Python

<sup>17</sup> Python, <https://it.wikipedia.org/wiki/Python>, (ultima visita il 5 dicembre 2022)

<sup>18</sup> Jansen Stefan, Machine Learning for Algorithmic Trading, (Birmingham: Packt, 2020), p. 7

<sup>19</sup> Premio Nobel per l'economia 2013, noto per il suo lavoro sulla teoria del portafoglio e l'asset pricing.

<sup>20</sup> Professore di Finanza al Dartmouth College, noto per la sua collaborazione con Eugene Fama sull'asset pricing.

<sup>21</sup> È un'operazione di acquisto di un bene o attività su un mercato e di rivendita su un altro, sfrutta quindi le differenze di prezzo per ottenere un profitto.

<sup>22</sup> Da HTML.it, <https://www.html.it/guide/guida-python/>, (ultima visita il 5 dicembre 2022)

Una peculiarità di Python è la possibilità di ottenere risultati in tempo reale senza dover aspettare alcuna compilazione come succede ad esempio con altri linguaggi quali C o C++. L'esecuzione di un programma in Python avviene in maniera del tutto simile, è ovvio però che quest'ultima opzione è quella da scegliere in caso di molteplici calcoli da fare.

Un esempio di codice Python scritto nell'interprete interattivo è il seguente:

```
>>> 5 * 3
15
>>>
>>> a = 5
>>> b = 6
>>> 2 * (a+b) + 3*a
37
```

Python lavora con un input e un output, queste sono due funzioni basilari.

Come si nota nell'esempio precedente in realtà basta scrivere la variabile o il numero che si desidera nell'interprete per ottenere la stampa di esso, utilizzando però la funzione *print()* si possono stampare diversi tipi di dati concatenandoli.

Un breve esempio:

```
>>> a = 12
>>> b = 3
>>> print(a, b, a-b)
12 3 9
>>>
>>> c = 'Hello'
>>> print(c, 'World!')
Hello World!
```

Per quanto riguarda l'input invece dei dati si utilizza la funzione *input()*, questa funziona accetta un singolo argomento che viene salvato come stringa. Ad esempio:

```
>>> nome = input('Inserisci il tuo nome: ')
Inserisci il tuo nome: Francesco
>>> print(nome)
'Francesco'
```

Un'altra caratteristica fondamentale in Python, che deriva da linguaggi come C, C++ o Java, è quella dell'indentazione del codice. Benché questa pratica sia comune a tutti i linguaggi in Python diviene fondamentale e quindi non più di natura puramente estetica, una mancata indentazione condurrà inevitabilmente a degli errori o malfunzionamenti. I costrutti come *if*, *for*, *while* ecc. devono essere tutti indentati:

```
print("eseguito sempre all'inizio")
if condizione:
    print('eseguito in mezzo solo se la condizione è vera')
print('eseguito sempre alla fine')
```

La comunità di Python per avere una maggiore consistenza ha stilato un documento chiamato *PEP8*, acronimo di *Python Enhancement Proposal*, che raccoglie diverse linee guida incluse quella per l'indentazione.<sup>23</sup>

---

<sup>23</sup> PEP 8, <https://www.python.org/dev/peps/pep-0008/>, (ultima visita il 5 dicembre 2022)



Python è un linguaggio a tipizzazione dinamica, cioè le variabili non sono altro che dei puntatori ad un oggetto<sup>24</sup>, infatti questo linguaggio permette la *Object Oriented Programming* (OOP) cioè la programmazione orientata agli oggetti.

Durante le fasi iniziali di programmazione è necessario identificare gli oggetti, identificare i dati e i comportamenti comuni a gruppi di oggetti e definire una classe<sup>25</sup> per ognuno di questi gruppi.

Le caratteristiche dell'OOP sono molteplici, alcune verranno utilizzate per la creazione dei grafici nel capitolo quarto, e sono l'ereditarietà, il polimorfismo, l'astrazione e il duck typing.

- **Ereditarietà:** le classi ereditano dati e comportamenti da un'altra classe, chiamata super-classe. Chi eredita è chiamata sotto-classe. Nel capitolo quarto, tutti i programmi creati ereditano dalla classe *DownloadDati*.
- **Polimorfismo:** le classi utilizzano in maniera intercambiabile determinati comportamenti che vengono ereditati dalla stessa super-classe.
- **Astrazione:** definisce un insieme di dati e comportamenti che deve essere per forza ereditato dalla sotto-classe.
- **Duck Typing:** le classi utilizzano in maniera intercambiabile determinati comportamenti, anche se non ereditano dalla stessa classe. È utile rispetto all'astrazione se non si vuole ereditare tutto il comportamento di una super-classe. Nel capitolo quarto, tutti i programmi oltre a ereditare dalla classe *DownloadDati* avranno a loro volta un comportamento chiamato *plot()*, necessario a tutte le classi per stampare il grafico.

### 3.3 Le API e i dati finanziari

L'accesso ai dati negli anni è diventato sempre più semplice e veloce. Ad oggi produciamo una quantità immensa di dati, questi a volte vengono messi a disposizione da chi li gestisce ne sono un esempio i prezzi dei titoli azionari gestiti da aziende tecnologiche come Yahoo e Google, o stesso dai grandi della finanza come il gruppo Nasdaq, Inc. fondatore del NASDAQ<sup>26</sup>.

Come fanno Yahoo, Google e il NASDAQ a rendere disponibili questi dati? Con le API, acronimo di *Application Programming Interface*, non sono altro che un insieme di procedure che permette le comunicazioni tra diversi software o computer. Una buona API fornisce una “scatola nera” cioè un livello di astrazione che permette al programmatore di non conoscere il funzionamento specifico che c'è nell'API in questione ma di ottenere direttamente un risultato. Python, e la community, hanno messo a disposizione diverse librerie che permettono di accedere a questi dati in maniera semplicissima, alcune delle più famose sono *yfinance* (Yahoo Finanza), *googlefinance* (Google Finanza) e *nasdaq-data-link* (NASDAQ).

In particolare le API di Yahoo servono ad ottenere i dati proprio come dal sito <https://finance.yahoo.com/>, incluse quindi tutte le criptovalute, valute, azioni, analisi del mercato e notizie. Le API un tempo erano accessibili a tutti, ma purtroppo per un utilizzo

---

<sup>24</sup> Sono “entità” che hanno sia dati che comportamenti.

<sup>25</sup> È un modello utilizzato per creare oggetti, comprende dati chiamati attributi e comportamenti chiamati metodi.

<sup>26</sup> Il NASDAQ è uno degli indici più famosi al mondo formato dalle maggiori aziende tecnologiche americane, fu il primo mercato borsistico elettronico. Nasce il 5 febbraio del 1971 a New York.

malevolo da parte di diversi individui nel 2017 Yahoo ha deciso di non mantenerle più. Le API utilizzate in questo studio sono quindi non ufficiali, queste purtroppo hanno lo svantaggio di poter essere rotte o bloccate in qualsiasi momento da chi eroga i dati. Se però la soluzione riportata va bene al lettore, allora per accedere a queste API esistono diverse librerie in Python come *RapidAPI*, *yFinance*, *Yahoo\_fin*. Esistono una moltitudine di dataset<sup>27</sup> finanziari, molto utili al machine-learning<sup>28</sup> e quindi alla previsione delle tendenze future:

- **Quandl:** utilizzato da più di 250000 analisti, banche e gestori di fondi. I dati inclusi sono precisi e accurati. Di proprietà della NASDAQ, Inc.
- **EU Open Data Portal:** include dati pubblicati a partire dal 2012 da agenzie ed istituzioni, contengono dati sull'ambiente, sul lavoro e sull'educazione
- **World Bank Open Data:** include dati che coprono la demografia della popolazione mondiale, oltre che diversi dati di varia entità economica e di sviluppo.
- **IMF Data:** include dati con informazioni su finanze internazionali, riserve di valuta estera, prezzi delle materie prime, tassi di debito. I dati sono stati raccolti dal Fondo Monetario Internazionale.
- **Financial Times Market Data:** include dati costantemente aggiornati su indici dei prezzi azionari, materie prime e cambi di valuta.

### 3.4 Le librerie necessarie

Una libreria o modulo è un file che contiene non altro che codice Python, un pacchetto invece è una cartella che contiene diversi moduli.

L'utilizzo di moduli e pacchetti consente un'organizzazione e gestione migliore soprattutto di programmi molto grandi, inoltre gli sviluppatori di Python hanno messo a disposizione i così detti moduli *built-in*, cioè moduli già pronti all'uso dopo l'installazione.

Gli esperti prediligono i dati economici e finanziari in forme tabulari, ad esempio sotto forma di array/vettore o matrice. Viene scelto questo formato perché è molto comodo e perché permette l'accesso e la modifica del singolo o per gruppi di dati cioè per riga e per colonna. Nel capitolo quarto verranno fatti alcuni esempi con file in formato *csv*.

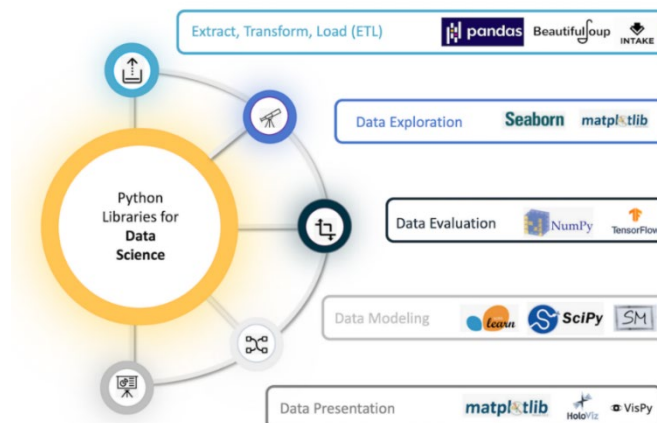


Figura 16: Le librerie più famose per la manipolazione dei dati, sono incluse Pandas e NumPy.

<sup>27</sup> Una collezione di dati.

<sup>28</sup> Sono dei metodi che utilizzano la statistica per individuare, attraverso algoritmi, delle somiglianze nei dati.

### 3.4.1 NumPy

Vettori, matrici e array<sup>29</sup> di dimensioni superiori sono strumenti fondamentali per la computazione di numeri. Quando un calcolo deve essere ripetuto per un insieme di valori è vantaggioso rappresentare i dati come matrici, per utilizzare le operazioni matematiche tra matrici. Questa scelta permette l'esclusione dell'utilizzo di loop che in Python sono molto onerosi in termini di calcolo, inoltre permette la delegazione di queste operazioni a librerie di più basso livello molto più efficienti.

Strutture efficienti per lavorare con array sono gestite da *NumPy*, acronimo di *Numeric Python*, il cuore di questa libreria è implementata nel linguaggio C che per la sua struttura è efficiente per la manipolazione e la gestione degli array. La differenza fondamentale tra un array di *NumPy* e una lista di Python è che il primo è un oggetto omogeneo, cioè con gli stessi tipi di dati al suo interno, di lunghezza predefinita mentre l'altro permette l'inserimento di diversi tipi di dati. Per questo e i motivi elencati sopra, operazioni e funzioni su array di *NumPy* sono molto più efficienti rispetto alle liste di Python. Le operazioni che *NumPy* permette sugli array sono molteplici, operazioni di algebra lineare fino alle trasformate di Fourier.<sup>30</sup>

Tabella 3: Attributi dell'array di NumPy

Attributo	Descrizione
shape	Numero di righe e colonne dell'array
size	Il numero di elementi nell'array
ndim	Numero di dimensioni (assi)
nbytes	Numero di byte utilizzati
dtype	Tipo di elementi nell'array

### 3.4.2 Pandas

È chiaro quindi che NumPy è la base per le operazioni matematiche in Python.

Un'altra libreria molto importante che permette la rappresentazione, in maniera conveniente, di strutture dati più complesse come serie e tabelle inoltre permette la loro trasformazione, separazione, conversione. La libreria in questione è *Pandas*, questa infatti estende la precedente libreria *NumPy*, con funzioni che permettono ad esempio l'indicizzazione con etichette, la gerarchia degli indici, allineamento dei dati per il loro confronto e la loro fusione, la gestione di dati mancanti<sup>31</sup>. In campo statistico *Pandas* si è affermata come la libreria fondamentale, utilizzata al pari del linguaggio R<sup>32</sup>.

Uno dei tipi di dati più famosi di *Pandas* è la *Series* (serie in italiano), questa non solo permette l'indicizzazione dei dati con numeri ma anche l'indicizzazione attraverso etichette, quest'ultima in alcune circostanze risulta più funzionale. Di seguito un esempio di codice

```
In [6]: s = pd.Series([909976, 8615246, 2872086, 2273305])
In [7]: s
Out[7]: 0 909976
        1 8615246
        2 2872086
        3 2273305
        dtype: int64
```

<sup>29</sup> Una sorta di contenitore, le cui caselle sono dette celle.

<sup>30</sup> Robert Johansson, Numerical Python, (California: Apress Media, 2019), p. 44

<sup>31</sup> *Ivi*, p. 405

<sup>32</sup> R è un linguaggio di programmazione specifico per la statistica.

con indici interi:

Altro esempio però che utilizza indici etichettati:

```
In [10]: s.index = ["Stockholm", "London", "Rome", "Paris"]
In [11]: s.name = "Population"
In [12]: s
Out[12]: Stockholm 909976
          London 8615246
          Rome 2872086
          Paris 2273305
          Name: Population, dtype: int64
```

Un'altra struttura dati fondamentale, che rende multi-dimensionale la *Series*, è il *DataFrame*. Inoltre questa struttura permette l'utilizzo di etichette descrittive e può essere vista come una collezione di oggetti di tipo *Series*.

Esistono diversi modi per inizializzare un *DataFrame*, ad esempio:

```
In [22]: df = pd.DataFrame([[909976, "Sweden"],
...: [8615246, "United Kingdom"],
...: [2872086, "Italy"],
...: [2273305, "France"]])
In [23]: df
Out[23]:
```

	0	1
0	909976	Sweden
1	8615246	United Kingdom
2	2872086	Italy
3	2273305	France

Mentre utilizzando la forza del *DataFrame* e quindi assegnando delle etichette alle colonne e alle righe si otterrebbe un risultato più “funzionale”:

	Population	State
Stockholm	909976	Sweden
London	8615246	United Kingdom
Rome	2872086	Italy
Paris	2273305	France

Per quanto riguarda invece strutture dati come Serie Storiche, in *Pandas* si possono implementare sia con la struttura *Series* che *DataFrame*, utilizzando la funzione *date\_range()* si possono generare le date che permettono quindi la creazione della Serie Storica:

```
In [61]: pd.date_range("2015-1-1 00:00", "2015-1-1 12:00",
...: freq="H")
Out[61]: <class 'pandas.tseries.index.DatetimeIndex'>
          [2015-01-01 00:00:00, ..., 2015-01-01 12:00:00]
          Length: 13, Freq: H, Timezone: None
```

### 3.4.3 yFinance

Questa libreria è fondamentale per il download dei dati finanziari, essa è open source<sup>33</sup> e sviluppata da Ran Aroussi. Ne esistono anche altre ma per la stesura del codice della tesi ho deciso di utilizzare questa, perché molto semplice e aggiornata. Come si intuisce la libreria scarica i dati da Yahoo Finance, un sito che offre molti dettagli sul mercato, sui bond, sulle valute, sulle criptovalute eccetera.

Prima della nascita di questa libreria non ufficiale, Yahoo supportava delle API ufficiali, purtroppo però per utilizzi “sbagliati” l’azienda il 15 maggio 2017 ha smesso di supportarle. I tre moduli fondamentali della libreria sono 3 e sono *Tickers*, *download*, *pandas\_datareader*.

Una volta scaricato il così detto *Ticker*<sup>34</sup> bisogna utilizzare la funzione *history()*, per ottenere una serie storica. yFinance, oltre il download del prezzo dei titoli azionari, permette il download anche del *rapporto prezzo-utili* e i *dividendi*.

```
import yfinance as yf

aapl = yf.Ticker("AAPL")
aapl
```

### 3.4.4 Matplotlib

La visualizzazione è uno strumento universale per investigare e comunicare risultati, è il prodotto finale di tutti gli studi computazionali.

In Python esistono diverse librerie per la visualizzazione, una delle più famose è *Matplotlib* questa supporta la pubblicazione di grafici in 2D e 3D, per una moltitudine di dati. Prima di creare grafici con *Matplotlib*, la libreria creerà in automatico un’area adibita al disegno formata da 2 assi, la x e la y.<sup>35</sup>

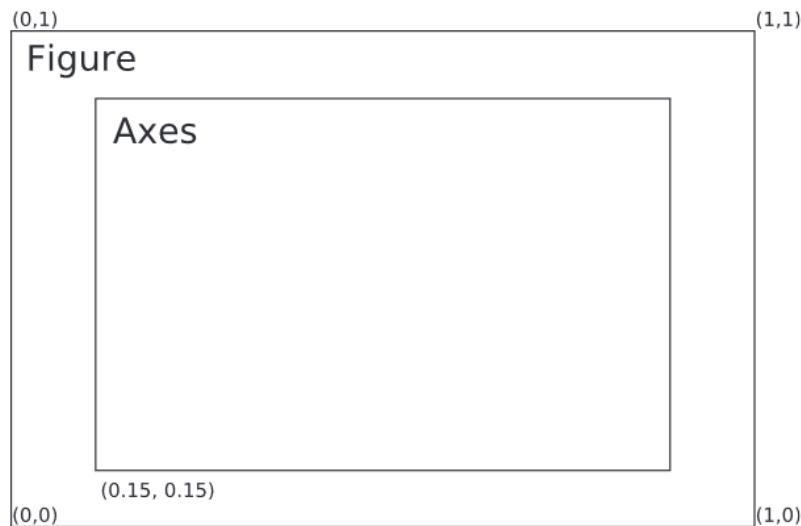


Figura 17: La figura base che Matplotlib stampa è formata da coordinate che vanno da (0,0) fino a (1,1)

<sup>33</sup> Si indica un software distribuito sotto i termini di una licenza che ne concede lo studio, l'utilizzo, la modifica e la redistribuzione.

<sup>34</sup> Il ticker è un'abbreviazione che identifica le società che vengono quotate su un mercato finanziario.

<sup>35</sup> Robert Johansson, Numerical Python, (California: Apress Media, 2019), p. 135

Gli assi forniscono un sistema di coordinate che permettono la stampa di grafici lineari, a barre o a punti. Per stampare ad esempio una funzione e le sue derivate, il primo passo da fare è creare un array con *NumPy* per i valori della  $x$  e successivamente sempre con *NumPy* si computano le derivate.

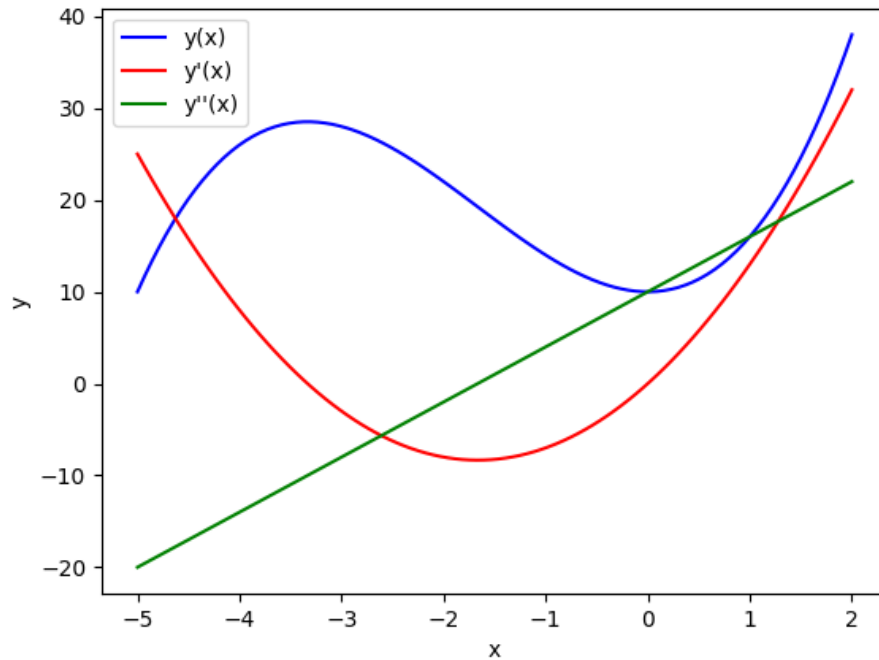


Figura 18: Stampa con Matplotlib della funzione  $y(x) = x^3 + 5x^2 + 10$  con la derivata prima e seconda

Di seguito il codice per riprodurre il grafico appena visto:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-5, 2, 100)
y1 = x**3 + 5*x**2 + 10
y2 = 3*x**2 + 10*x
y3 = 6*x + 10

fig, ax = plt.subplots()
ax.plot(x, y1, color="blue", label="y(x)")
ax.plot(x, y2, color="red", label="y'(x)")
ax.plot(x, y3, color="green", label="y''(x)")
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.legend()

plt.show()
```

### 3.5 Analisi di serie temporali

Una serie temporale quindi non è altro che una sequenza di dati che si susseguono seguendo un ordine, in questo caso la data.

L'analisi di serie temporali permette quindi di capire cosa influenza una determinata variabile della serie, anche in diversi periodi di tempo, inoltre l'analisi di serie temporali permette anche la previsione dei valori futuri.<sup>36</sup>

Il mercato azionario permette la vendita e l'acquisto di azioni, le previsioni quindi di eventuali tendenze possono fare la differenza e soprattutto possono portare ad un gran profitto; queste però sono molto complesse, se non impossibili per singoli individui con strumentazione inadeguata. Maggiore è la complessità dell'ambiente che si vuole prevedere maggiore sarà la difficoltà di previsione, o meglio, minore sarà l'accuratezza di previsione soprattutto di lungo periodo.

Se però, secondo Dow, il prezzo di un'azione non è un valore casuale e può essere considerato come un valore discreto e quindi come una serie temporale allora è possibile analizzare il titolo azionario e identificarne uno o più modelli. Uno dei più buoni è il modello *ARIMA* acronimo di *Autoregressive Integrated Moving Average*, è spesso utilizzato in economia e finanza ed è un modello robusto ed efficiente con un'elevata potenzialità di previsione nel breve periodo.

#### 3.5.1 Le componenti delle serie temporali

Le serie tempotrli sono formate da diverse componenti<sup>37</sup>:

- **Trend**: una tendenza in un intervallo di tempo fisso. Può essere di lungo o breve periodo; è un movimento graduale. Il trend può essere di due tipi, rialzista o ribassista ed è quindi prevedibile.
- **Stagionalità**: come il trend, è in un intervallo di tempo fisso. È di breve periodo, a differenza del trend non è graduale. La stagionalità ha un pattern che si ripete nel tempo, è quindi prevedibile.
- **Ciclicità**: non è in un intervallo di tempo fisso. Può essere di lungo o breve periodo; non è un movimento graduale. Non ha un pattern preciso e la sua previsione può risultare complessa.
- **Irregolarità**: non è in un intervallo di tempo fisso, la sua durata è variabile e il suo pattern non è ripetibile. È quindi di difficile previsione.

#### 3.5.2 La stazionarietà

Le serie temporali possono essere stazionarie o non stazionarie<sup>38</sup>, se è stazionaria si avranno diverse proprietà:

1. Mancheranno trend, stagionalità, ciclicità o irregolarità.
2. La media della serie sarà costante.
3. La varianza sarà anch'essa costante rispetto al tempo.

---

<sup>36</sup> Stock Market Forecasting using TSA, <https://www.kdnuggets.com/2020/01/stock-market-forecasting-time-series-analysis.html>, (ultima visita il 28 novembre 2022)

<sup>37</sup> A Comprehensive Guide to Time Series Analysis, <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/>, (ultima visita il 28 novembre 2022)

<sup>38</sup> *Ibidem*.

Se una serie dovesse essere non stazionaria, le soprantanti proprietà verranno a mancare. Esistono diversi test statistici per vedere se la serie temporale è stazionaria o meno, sono famosi il test di *Dickey-Fuller aumentato* ed il test di *Kwiatkowski-Phillips-Schmidt-Shin*.

Nel caso in cui una serie non dovesse essere stazionaria esistono metodi di rimozione del trend, di differenziazione e di trasformazione che permettono la trasformazione della serie non stazionaria, in una stazionaria.

La stazionarietà è fondamentale per lo studio corretto della serie stessa.

### 3.6 Visualizzazione delle serie temporali

Dopo aver introdotto le API, le librerie e le caratteristiche delle serie temporali utilizzando tutte queste nozioni insieme si riesce a visualizzare un grafico in Python.

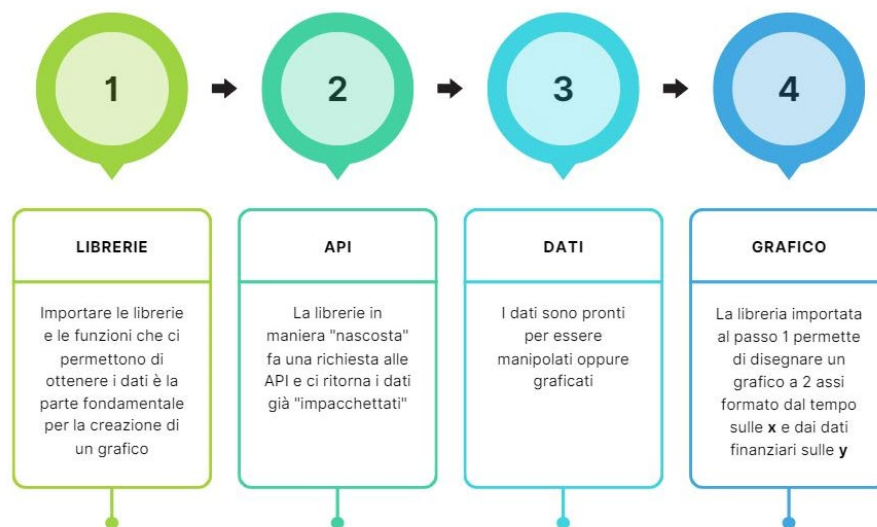


Figura 19: I passi da seguire per creare un grafico.

Le librerie fondamentali per il plot<sup>39</sup> dei dati sono diverse ma una delle maggiori è *Matplotlib* in particolare il modulo *PyPlot*.

Le librerie scelta invece per il reperimento dei dati veri e proprio è *yfinance*, analizzando i file della libreria, in particolare il file *ticker.py*, si evince che la libreria per ogni azione che vogliamo scaricare crea un oggetto di classe *Ticker*.

Questa classe include la funzione *download\_options()* nella quale sono inclusi i richiami all'API di Yahoo Finance.

---

<sup>39</sup>Termine inglese per indicare la creazione del grafico.



Il risultato ottenuto è un *DataFrame*, infatti prima della creazione della classe viene richiamato *Pandas*. L'oggetto ottenuto sarà così composto:

```
<class 'pandas.core.frame.DataFrame'>
Data columns (total 5 columns):
#   Column  Non-Null Count  Dtype
---  -
0    Date    234 non-null    float64
1   Open    234 non-null    float64
2   High    234 non-null    float64
3   Low     234 non-null    float64
4   Close   234 non-null    float64
```

La classe *Ticker* eredita gli attributi e i metodi dalla classe astratta *TickerBase*, questi permettono non solo di vedere il prezzo dell'azione ma anche di vedere i dati dell'azienda proprio come sul sito di Yahoo Finance.

## CAPITOLO 4 – Esempio pratico su diversi titoli azionari

### 4.1 Introduzione

In questo capitolo si vuole studiare un paniere di titoli azionari che trattano petroli o derivati, sono stati scelti dei titoli azionari rappresentativi per paesi come Gran Bretagna, Francia, Italia, Russia e Stati Uniti. Precisamente per la Gran Bretagna si analizzerà la **Shell plc**, per la Francia si analizzerà la **TotalEnergies SE**, per l'Italia invece si analizzeranno la **Eni S.p.A** e la **Saras S.p.A**. Per la Russia invece si analizzeranno i due colossi **PJSC Gazprom** e **PJSC Lukoil**, per gli Stati Uniti invece la **Chevron Corporation** e la **Occidental Petroleum Corporation**.

La scelta dei paesi non è casuale, si vuole rappresentare il più possibile la situazione prima e dopo il conflitto Russo-Ucraino e soprattutto il suo impatto economico e finanziario.

Questo impatto ovviamente è limitato ai titoli azionari, questi non per forza riflettono la situazione economica dei paesi ma è possibile capire il sentimento degli investitori, piccoli e grandi, nei confronti di un determinato mercato.

Larry Fink, CEO di *BlackRock*, la società di gestione del risparmio più grande del mondo ha affrontato proprio questo tema.<sup>40</sup>

Secondo lui esistono tre fenomeni, alcuni dei quali partiti dalla pandemia, che cambieranno il mondo finanziario.

**Fine alla globalizzazione:** La guerra, con le sanzioni alla Russia e il congelamento delle sue riserve valutarie, ha peggiorato una situazione già avviata dal Covid 19 e i lockdown.

Alcune grandi compagnie di auto quali Volkswagen e BMW hanno dovuto interrompere i loro cicli di produzione perché avvenivano in Ucraina.

Altre conseguenze dell'invasione riguardano la crisi dei chip che rischia di essere ancora più sostenuta, il gas *neon* utilizzato per “scrivere” sui chip dei microprocessori viene prodotto per più della metà della domanda in Ucraina e Russia.

**Le valute vanno verso il digitale:** La guerra ha fatto notare a molti Stati la loro dipendenza valutaria, un ruolo più attivo nella creazione di una valuta digitale potrebbe portare a questi un vantaggio non indifferente. È molto avanti nella ricerca la Cina, che con il suo Yuan digitale (e-CNY) il 31 dicembre 2021, vantava già 261 milioni di portafogli personali aperti e le transazioni erano pari a oltre 87 miliardi di yuan.

**Transizione ecologica:** La guerra è stato un catalizzatore per la transizione ecologica, questa, già avviata da qualche anno ma interrotta per la pandemia, è inevitabile. L'umanità finalmente sta iniziando a capire quanto è importante il nostro pianeta e la sua salute.

Il 24 febbraio 2022 le truppe russe invadono l'Ucraina, il presidente Putin parla di un'operazione militare speciale per demilitarizzare l'Ucraina.

Alcune conseguenze saranno immediate, altre le avremo viste col tempo.

Le borse mondiali di questo atto ne hanno risentito poco, quelle europee ne hanno risentito di più proprio perché l'Europa ha, ormai si può dire aveva, maggiori relazioni col mondo russo.

Il dollaro, usato come moneta rifugio, si è rafforzato nei confronti di tutte le altre monete, l'Euro infatti ha perso circa il 30% del valore.

---

<sup>40</sup> Da Il Sole 24 Ore, <https://www.ilssole24ore.com/art/larry-fink-ecco-tre-conseguenze-guerra-mercati-ed-economia-AEyWXMMB>, (ultima visita il 20 dicembre 2022)

Il rublo è una valuta che sta per crollare, per arginare queste conseguenze la borsa russa chiude per diversi giorni. L' Europa inizia a sanzionare la Russia. Le materie prime, non solo quelle energetiche, iniziano a subire degli aumenti repentini di prezzo e questo influisce anche sull'inflazione.

## 4.2 Analisi di medio periodo

Si vuole considerare come periodo solamente l'anno 2022, questo perché si vuole tenere in considerazione il conflitto Russo-Ucraina. L'analisi verrà strutturata per paesi, partendo dalla Russia e dai relativi titoli.

Si vuole analizzare il titolo *PJSC Gazprom* e i suoi supporti e resistenze.

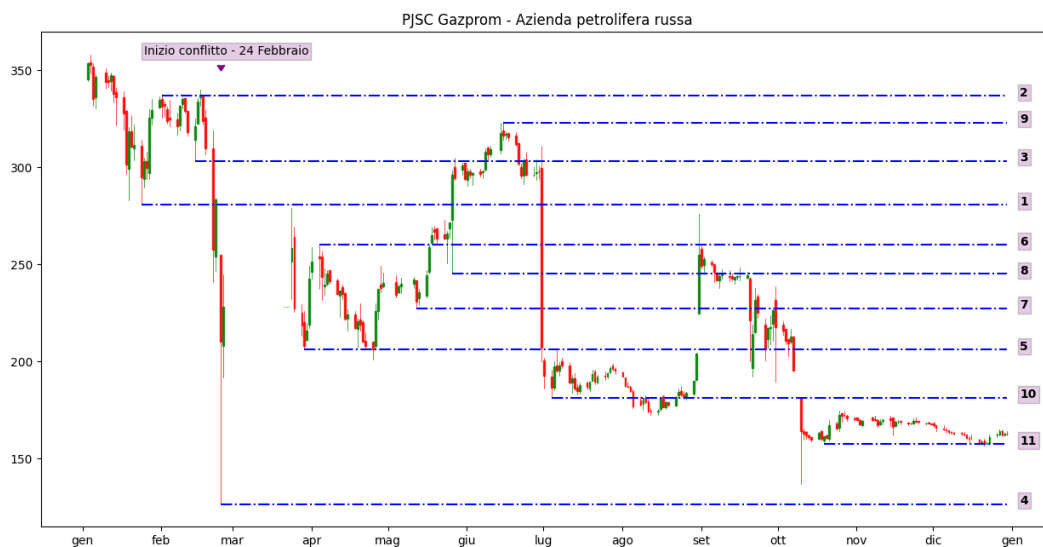


Figura 20: Supporti e resistenze - Codice Python in Appendice A.8

Per una maggiore comprensione i supporti e le resistenze sono stati contrassegnati con dei numeri.

La tendenza primaria del titolo è ribassista, con minimi e massimi decrescenti. Ad inizio gennaio il titolo azionario valeva circa 350 rubli<sup>41</sup>, fino a febbraio il titolo ha avuto una tendenza laterale fino ad arrivare al giorno 24 febbraio cioè lo scoppio del conflitto. Il titolo azionario è sceso in due/tre giorni fino a 150 rubli circa, un crollo del 40%.

Fino al 21 marzo il MOEX<sup>42</sup> non ha scambiato alcun titolo, per tutelarsi dalle conseguenze di questo avvenimento. La resistenza più importante, che se dovesse essere recuperata, comporterebbe quindi il ritorno a quota 350 rubli è la *numero 2*.

Durante quei giorni neanche il supporto (ora resistenza come dice la teoria) *numero 1* ha retto il forte tonfo.

Durante i mesi di aprile fino a giugno il titolo ha ripreso quota fino ad arrivare ad un massimo locale di circa 325 rubli e costruendo in questi mesi diversi supporti, più o meno, importati.

A fine giugno però il titolo subisce un altro crollo e buca tutte i supporti fino ad arrivare alla *numero 10* cioè quota 180 rubli circa, questa proprio per questo motivo diviene molto importante. Benché tra settembre e ottobre il prezzo sia salito fino a quota 250 rubli, ad oggi

<sup>41</sup> Valuta russa, al 2 gennaio 2022 il tasso di cambio di 1 euro è di 84 rubli.

<sup>42</sup> Borsa di Mosca.

rimane sotto quota 180 rubli. Il titolo dall'inizio del conflitto a perso quindi circa il 40% del suo valore iniziale.

L'analisi tecnica non è utile solo per capire cosa è successo ma anche per aiutarci a prevedere alcuni possibili esiti, tenendo a mente però che non è una palla magica e che a volte non funziona.

Utilizzando infatti l'indicatore MACD intorno ai primi di giugno le due medie mobili si sono incrociate (MACD e SEGNALE), da positivo il MACD diviene negativo. Questo è un segnale di vendita, infatti un mese dopo il titolo crolla da 300 rubli a quasi 220, senza apparente motivo.

Successivamente accade la stessa cosa intorno alla metà di settembre, infatti dopo un lieve recupero del prezzo fino a quota 280 rubli le medie mobili si sono incrociate e nel mese successivo il prezzo crolla.

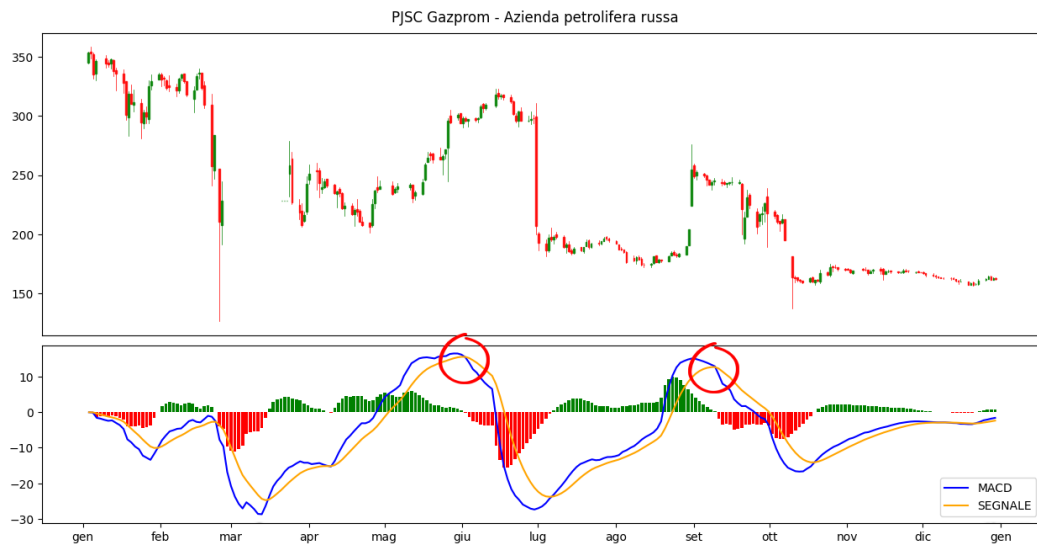


Figura 21: MACD - Codice Python in Appendice A.5

Situazione abbastanza analoga succede per il titolo azionario di un'altra grande azienda petrolifera russa, la *PJSC Lukoil*.



*Figura 22: Supporti e resistenze - Codice Python in Appendice A.8*

Analogamente il titolo azionario ha una tendenza ribassista, che da maggio in poi diviene laterale. Il titolo valeva circa 7000 rubli fino a metà febbraio, prima infatti il massimo locale è proprio la resistenza *numero 1* a quota 7000 rubli. In data 24 febbraio, con l'inizio del conflitto, il titolo inizia a crollare e buca il supporto *numero 2* fino a toccare un minimo di circa 3000 rubli; un crollo superiore al 50%.

Dopo la pausa del MOEX dalle contrattazioni il titolo riprende quota intorno ai 6000 rubli ma verso aprile rimane sotto il supporto *numero 4* che quindi diviene resistenza, la più importante da bucare.

Da maggio ad oggi il prezzo del titolo azionario è oscillato da un massimo di 5000 a un minimo di 3500, seguendo una tendenza laterale. Quindi il prezzo del titolo è crollato del 50%.

Rappresentando i titoli azionari russi con questo paniere la loro tendenza è, inevitabilmente, ribassista.

Si vogliono analizzare dei titoli che rappresentano l'Italia, partendo con la *Eni S.p.A.*



Figura 23: Supporti e resistenze - Codice Python in Appendice A.8

La tendenza prima del titolo azionario *Eni* è una tendenza laterale, ad inizio 2022 aveva un valore che è oscillato tra i 13 e i 14 euro. Il titolo azionario il 24 febbraio, allo scoppio della guerra, non ha subito nessun'alterazione anzi è salito fino a quota 14.5 euro, venendo a creare la resistenza *numero 5*.

Fino a giugno il prezzo ha tenuto un andamento laterale, successivamente però c'è stato un crollo del prezzo fino a quota 11 euro fino al supporto *numero 9*. Il prezzo da luglio a ottobre è rimbalzato fino al vecchio supporto, ora resistenza, *numero 2*, per poi ritornare a toccare il supporto *numero 9*.

Viene a crearsi la formazione doppio minimo, infatti essendo una formazione di inversione il prezzo dopo aver superato la resistenza *numero 8* riesce a recuperare quota fino alla resistenza *numero 6*. Ad oggi i movimenti di prezzo sono rimasti sotto questa resistenza fino ad arrivare ad un prezzo di 13.5 euro. In conclusione, il prezzo durante tutto il 2022 ha fluttuato da un minimo di 11 euro fino ad un massimo di 15 circa ma la differenza di prezzo tra inizio e fine anno è pari a zero.

Si vuole analizzare con fasci di medie mobili la *Saras S.p.A.*

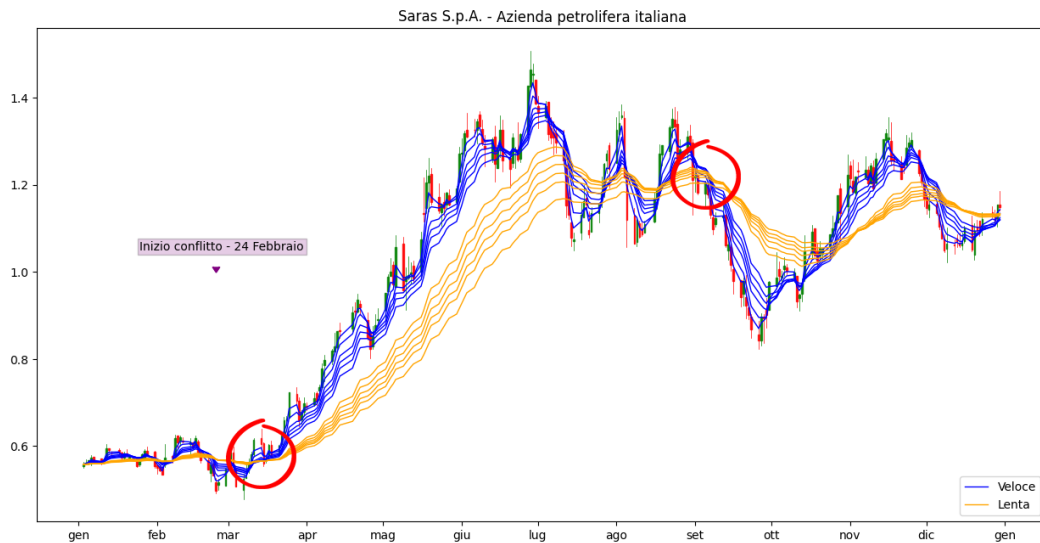


Figura 24: Fascio di medie mobili - Codice Python in Appendice A.4

La tendenza primaria del titolo *Saras* per tutto il 2022 è una tendenza fortemente rialzista. Fino a marzo ha avuto una media di prezzo intorno ai 0.6 euro, non ha subito alcuna variazione dovuta all'inizio del conflitto della guerra Russo-Ucraina. Intorno a metà marzo però c'è stato un segnale di acquisto molto importante, cioè quando il fascio delle medie mobili veloci incrocia, dal basso verso l'alto, quello delle medie mobili lente.

Infatti poi il trend è stato fortemente rialzista fino ad arrivare ad un massimo di 1.4 euro ad inizio luglio raggiungendo un incremento quindi del 80%.

Successivamente il titolo ha mantenuto una tendenza laterale fino ad arrivare ad inizio settembre dove si è venuto a creare un altro segnale, questa volta di vendita cioè quando il fascio delle medie mobili veloci incrocia, dall'alto verso il basso, quello dalle medie mobili lente. Infatti nel breve periodo il titolo azionario ha subito un crollo fino a quota 1 euro circa. Il titolo termina il 2022 con il recupero del prezzo fino a quota 1.2 euro.

Nel grafico si notano anche altri incroci, questi però non possono essere considerati rilevanti perché hanno creato un segnale troppo in ritardo.

Si può concludere che nel breve periodo, l'inizio del conflitto non ha influito sui titoli azionari italiani, alcuni titoli però ne hanno beneficiato nel lungo periodo; infatti quelli analizzati hanno chiuso l'anno con un prezzo maggiore o uguale rispetto ad inizio 2022.

Ora si vogliono analizzare un'azienda francese e una del Regno Unito, per vedere la differenza con l'Italia. Benché ad oggi il Regno Unito non faccia più parte dell'Unione Europea l'azienda analizzata è la *Shell plc*, una delle più grandi aziende petrolifere con impianti in tutta Europa e nel mondo. Queste aziende rappresenteranno quindi i titoli europei.

Partendo dalla *Total Energies SE*, si decide di utilizzare l'oscillatore Stocastico.

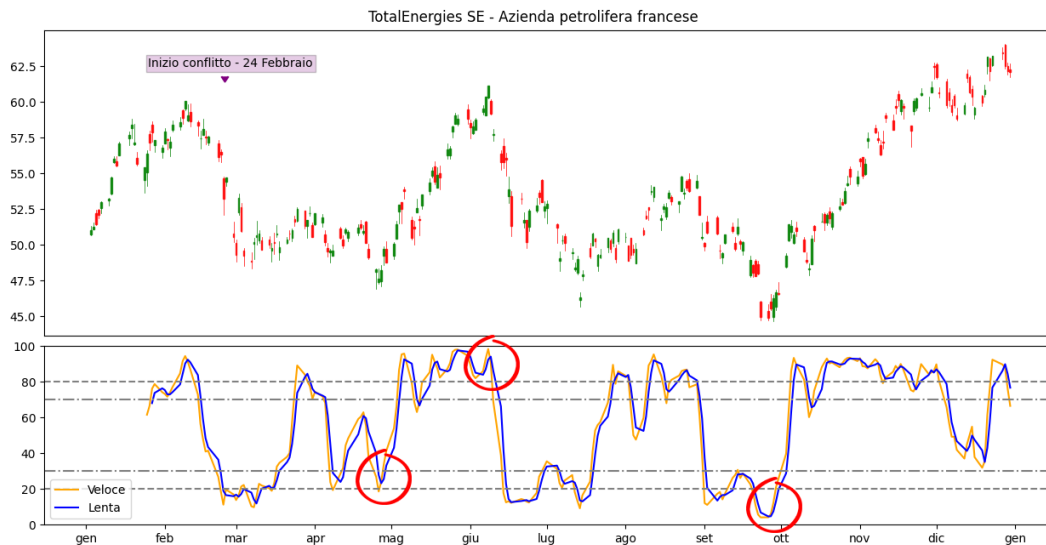


Figura 25: Stocastico - Codice Python in Appendice A.7

La tendenza primaria del titolo per tutto il 2022 è una tendenza laterale.

Il titolo apre l'anno con un prezzo intorno ai 50 euro, fino a febbraio arriva ad un massimo locale di 60 euro. Dopo l'invasione russa dell'Ucraina, iniziata il 24 febbraio, molte società internazionali si sono ritirate dalla Russia.

Il primo marzo, *TotalEnergies* ha annunciato che "non fornirà più capitale per nuovi progetti in Russia", ma ha mantenuto la proprietà della sua partecipazione del 19,4% in Novatek, del 20% nel progetto Yamal e del 10% in Arctic LNG 2 (tutte società russe). Ciò ha portato a pesanti critiche, in particolare alla luce del completo disinvestimento di altre importanti società energetiche occidentali e dell'annuncio dell'Unione europea di diventare più indipendente dal punto di vista energetico dalla Russia.

Un'indagine di Global Witness ha dimostrato che un giacimento di gas siberiano di proprietà di *TotalEnergies* ha fornito una raffineria, che produce carburante per aerei da guerra russi.<sup>43</sup>

Il prezzo quindi è sceso di nuovo in zona 50 euro, è rimasto lì tra oscillazioni fino a fine aprile quando finalmente si ha il primo segnale di acquisto importante (si può anche notare una formazione di doppio minimo, che rafforza il segnale) e cioè le medie dello Stocastico si intersecano uscendo poi dalla zona di ipervenduto. Il prezzo sale di nuovo a quota 60 euro e supera il precedente massimo.

A metà giugno però si ha un altro segnale, questa volta di vendita: le medie dello stocastico si incrociano ed escono dalla zona di ipercomprato; il prezzo crolla questa volta ad una resistenza intorno ai 52.5 euro per poi scendere sotto i 50 euro. Dopo una leggera ripresa nei mesi di agosto e settembre il prezzo non riesce a superare la resistenza che si è venuta a creare nei mesi precedenti intorno ai 52.5 euro, il prezzo crolla intorno ai 45 euro. Si viene a formare il minimo del 2022 e anche un segnale di acquisto, infatti le medie dell'oscillatore stocastico si intersecano e poi escono dalla zona di ipervenduto. Il prezzo da ottobre fino alla fine dell'anno recupera quota e inizia un trend rialzista che lo porta a superare addirittura il massimo che si è venuto a creare a metà giugno.

L'oscillatore Stocastico, come l'RSI, è uno strumento sicuramente potente ma anche molto

---

<sup>43</sup> Da global witness, <https://www.globalwitness.org/en/campaigns/stop-russian-oil/total-russian-jet-fuel/>, (ultima visita il 10 gennaio 2023)



pericoloso; questo infatti crea molti falsi positivi, è facile leggerlo con tutta la serie storica *a posteriori* ma è molto difficile da leggere in “tempo reale”. Questi due oscillatori vanno accompagnati quindi da altri strumenti per far sì di non commettere errori di valutazione.

Si vuole analizzare il titolo azionario dell’azienda *Shell*, gestita nel Regno Unito.

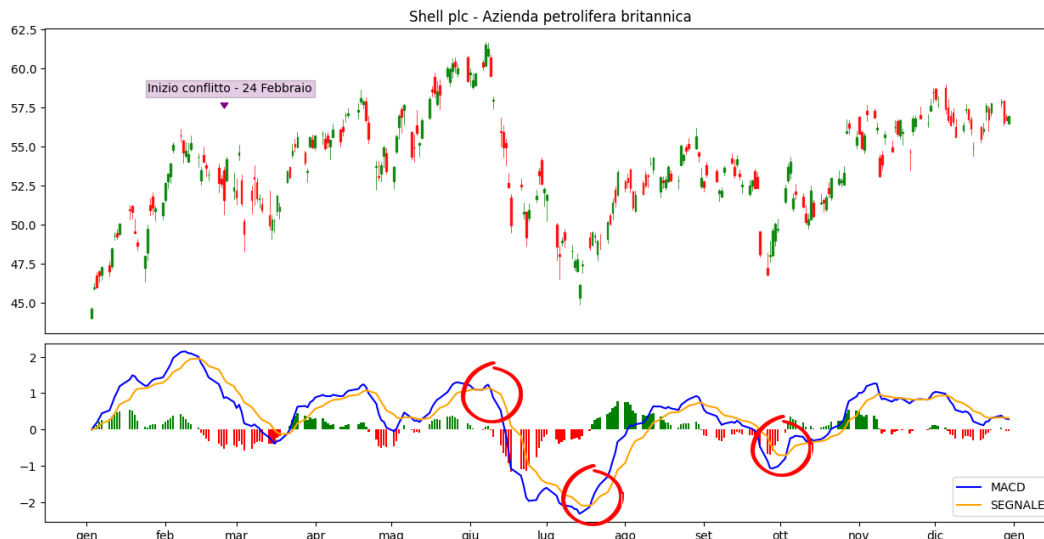


Figura 26: MACD - Codice Python in Appendice A.5

Il titolo azionario ha un trend leggermente rialzista. Inizia il 2022 con una quota a 45 dollari fino ad arrivare ad inizio febbraio ad un massimo locale di 55 dollari.

Benché il 24 febbraio il prezzo non subisca variazioni, successivamente inizia un trend rialzista che dura fino a giugno, verso la metà di esso infatti si viene a creare il primo segnale importate, il MACD incrocia la media SEGNALE al ribasso, questo è un segnale di vendita che è anche confermato dal nuovo massimo che si è formato nei giorni precedenti.

Infatti successivamente il prezzo scende da circa 62.5 dollari fino ad arrivare prima a quota 52.5 dollari per poi toccare quota 47.5 dollari. Arrivati qui il MACD ci dà un altro segnale, questa volta di acquisto, le medie infatti si intersecano al rialzo. La conseguenza di ciò è che il prezzo riprende quota fino a circa 52.5 dollari e durante i primi di ottobre dopo un leggero crollo il MACD incrocia di nuovo il SEGNALE al rialzo; il trend rialzista continua fino al termine dell’anno.

Il MACD rispetto allo Stocastico e al RSI produce meno falsi positivi, in questo breve studio infatti viene usato spesso.

Il risultato che si ottiene analizzando questi due titoli è simile a quello che è avvenuto analizzando quelli italiani, e cioè l’apprezzamento del valore azionario nel tempo, eccezion fatta per il titolo *TotalEnergies* che in un primo momento si è dimostrato molto vicino alle aziende russe e quindi con l’inizio della guerra il titolo ha perso fiducia e valore.

Si vuole vedere due titoli degli Stati Uniti.  
Il primo in analisi è quello della *Chevron Corporation*.

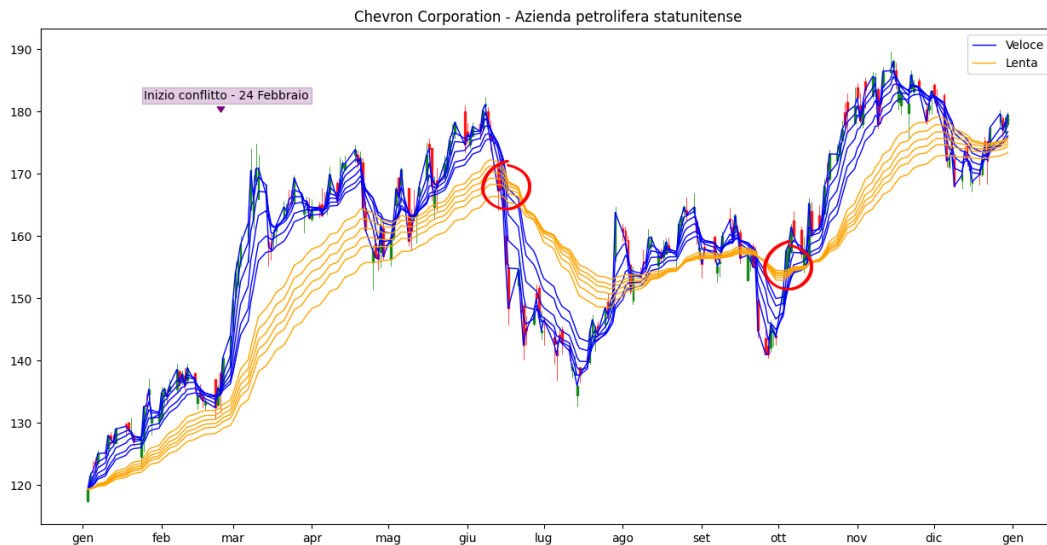


Figura 27: Fascio di medie mobili - Codice Python in Appendice A.4

Il trend primario è un trend rialzista, il prezzo del titolo inizia l'anno intorno ai 120 dollari, tra gennaio e febbraio il prezzo riesce ad arrivare 140 dollari, creano un massimo locale. Dopo il 24 febbraio, data dell'inizio del conflitto, i prezzi prendono quota vertiginosamente, talmente tanto che i fasci di medie mobili non riescono a segnalare la repentinità del cambiamento di tendenza. Per movimenti di prezzo molto forti i fasci di medie mobili non riescono a cogliere il segnale. Il prezzo durante i primi giorni di marzo arriva ad un massimo di circa 175 dollari. In poco meno di due settimane il prezzo è salito di circa il 30%.

Fino a maggio il prezzo è rimasto più o meno stabile con oscillazioni poco importanti, intorno a maggio però il prezzo crolla fino a 160 dollari dove c'è un'importante resistenza. Da maggio a giugno il prezzo sale di nuovo fino ad arrivare ad un massimo che si trova intorno ai 180 dollari, questo è un nuovo massimo.

Verso metà giugno però i fasci di medie mobili si incrociano al ribasso, è un forte segnale di vendita, infatti, nel mese successivo fino a metà luglio il prezzo è sceso fino ai 140 dollari. Da metà luglio in poi però il prezzo riprende quota per poi riscendere di nuovo a circa 140 dollari. Durante la prima settimana di ottobre le medie mobili si incrociano di nuovo, questa volta al rialzo.

È un segnale di acquisto importante infatti il prezzo supera dapprima quota 160 dollari, che era diventata una zona di resistenza, per poi arrivare durante il mese di dicembre ad un nuovo massimo intorno a quota 190 dollari. Il prezzo termina l'anno 2022 intorno ai 180 dollari, decretando quindi un incremento rispetto ad inizio anno di circa il 40%

L'ultimo titolo azionario in analisi è quello della *Occidental Petroleum Corporation*.



Figura 28: Trendlines - Codice Python in Appendice A.9

Grazie alle trendline si può subito notare il trend, molto forte, al rialzo. Infatti il titolo *Occidental Petroleum Corporation* inizia l'anno ad un prezzo di circa 30 dollari, durante febbraio continua ad aumentare il prezzo rimbalzando anche sulla trendline che poi si verrà a formare. Il 24 febbraio il prezzo sale da circa 38 dollari fino circa 58 dollari. In soli quattro giorni un incremento del 50%.

Nei mesi successivi fino a giugno il prezzo riesce a salire fino al massimo locale di 70 dollari, poi ha una breve correzione verso il basso fino a ritoccare quota 58 dollari che ormai è un supporto importante. Da lì il prezzo fino ad agosto è salito fino a 62 dollari, successivamente riesce a salire ancora di più e a creare un nuovo massimo in settembre arrivando oltre quota 70 dollari. Durante il mese di settembre però il prezzo cala di nuovo e ritocca il supporto intorno ai 58 dollari, come si vede il prezzo rimbalza sempre lì; è un supporto molto importante. Infatti il prezzo a fine 2022 chiude proprio in quelle zone, dopo aver provato di nuovo a superare il precedente massimo senza successo. L'incremento del prezzo da inizio anno passa da 30 dollari a circa 60, un incremento del 50%.

## CONCLUSIONE

In conclusione, l'obiettivo di questa tesi è quello di presentare la materia in questione, l'analisi tecnica applicata con *Python*, frequentemente discussa e criticata dal mondo accademico perché considerata troppo soggettiva e senza alcuna base scientifica.

Dopo aver spiegato ed esposto i concetti fondamentali, su cui essa si basa, si è passato a studiarne gli strumenti sia in modo teorico che pratico. Infatti tutto ciò che è precedentemente descritto viene messo in pratica e viene commentato in un breve studio. Utilizzando infatti *Python* si è potuto constatare come il mercato ha reagito a determinate circostanze. Per ottenere il massimo dall'analisi tecnica è necessario uno studio congiunto di argomenti prettamente economici e finanziari, che purtroppo per forza di cose vengono escluse da questa tesi.

Benché personalmente sia convinto non sia l'unica soluzione per ottenere profitto in operazioni in borsa, è innegabile la sua sempre maggiore importanza nel mondo finanziario. Veritiera o meno, l'analisi tecnica con gli anni diviene sempre più diffusa e con la sua maggiore diffusione a sua volta il mercato reagirà positivamente ad essa. La forza di questa disciplina infatti è proprio la sua semplicità.

Nel capitolo quarto si è compreso che dall'inizio del conflitto gli investitori di azioni russe che trattano petroli hanno perso fiducia e ciò ha portato ad una vendita istantanea che però è continuata nel tempo. Sia la *PJSC Gazprom* che la *PJSC Lukoil* durante il 2022 rispetto a inizio anno hanno perso un valore azionario di circa il 50%.

I titoli europei, benché all'inizio hanno avuto una tendenza ribassista successivamente si nota che il conflitto non ha danneggiato economicamente le azioni in questione. Infatti per quanto riguarda i titoli analizzati hanno acquisito tutti un po' di valore rispetto ad inizio anno.

Le azioni italiane riprendono un po' il comportamento di quelli europei. I titoli analizzati infatti non hanno sofferto dell'inizio del conflitto e chi più chi meno, hanno acquisito valore nel tempo chiudendo il 2022 a un prezzo maggiore rispetto ad inizio anno.

I titoli statunitensi invece sono quelli che hanno avuto un maggiore incremento. Si sono innescati degli ordini di acquisto molto importanti, che hanno fatto vedere aumenti del prezzo anche del 30% in pochi giorni. Dei titoli azionari analizzati alcuni hanno acquisito fino al 50% di valore rispetto all'inizio del 2022.

In conclusione vedendo l'instabilità in Russia gli investitori hanno preferito rivolgersi ad un altro mercato, quello statunitense preferito a quello europeo.

La domanda a cui si vuole rispondere è se l'analisi tecnica possa aiutare a prevedere le tendenze future dei prezzi.

È una domanda difficile da rispondere, soprattutto considerando che il materiale esposto ed analizzato non è sufficiente. Nonostante ciò si riescono a notare, grazie a grafici e strumenti, alcune tendenze e prevedere alcuni esiti.

Per riuscire al meglio ed escludere errori di valutazione l'analisi tecnica deve essere affiancata però da basi ancor più solide su temi finanziari ed economici.

## APPENDICE A – Codice Python per creare i grafici

Il seguente codice, con altri esempi, è anche reperibile online al link seguente:  
<https://github.com/landifrancesco/Analisi-Dati-Finanziari-in-Python>

### A.1 Le classi da cui ereditare le funzioni

```
import locale
from mplfinance.original_flavor import candlestick_ohlc
import matplotlib.dates as mpl_dates
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import pandas_ta as ta
from scipy.stats import linregress
import yfinance as yf

locale.setlocale(locale.LC_ALL, 'it_IT')

class DownloadDati:
    def __init__(self, titolo_azionario):
        self.df = None
        self._marker = None
        self._ylim = None
        self._ymarker = None
        self._ytext = None
        self._title = None
        if ".csv" in titolo_azionario:
            self.df = self.csv(titolo_azionario)
        else:
            self.df = self.yahoo(titolo_azionario)

    def yahoo(self, ticker):
        df = yf.download(ticker, start='2022-01-01', end='2022-12-31', threads=True)
        df['Date'] = pd.to_datetime(df.index)
        df['Date'] = df['Date'].apply(mpl_dates.date2num)
        df = df.loc[:, ['Date', 'Open', 'High', 'Low', 'Close']]
        return df

    def csv(self, file):
        df = pd.read_csv(file, delimiter=";", index_col=0, keep_date_col=True)
        df['Date'] = pd.to_datetime(df.index)
        df['Date'] = df['Date'].apply(mpl_dates.date2num)
        df = df.loc[:, ['Date', 'Open', 'High', 'Low', 'Close']]
        return df

    @property
    def marker(self):
        return self._marker

    @marker.setter
    def marker(self, value):
        self._marker = value

    @property
```

```

def yaxis_limit(self):
    return self._ylim

@yaxis_limit.setter
def yaxis_limit(self, value):
    self._ylim = value

@property
def yaxis_marker(self):
    return self._ymarker

@yaxis_marker.setter
def yaxis_marker(self, value):
    self._ymarker = value

@property
def yaxis_text(self):
    return self._ytext

@yaxis_text.setter
def yaxis_text(self, value):
    self._ytext = value

@property
def title(self):
    return self._title

@title.setter
def title(self, text):
    self._title = text

```

## A.2 Bande di Bollinger

```
from libs import *

class BollingerBands(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)
        self._get_bb(20)

    def _get_bb(self, periodo):
        sma_str = 'SMA_' + str(periodo)
        self.df[sma_str] =
self.df['Close'].rolling(window=periodo).mean()
        std = self.df['Close'].rolling(window = periodo).std()
        self.df['UpperBB'] = self.df[sma_str] + std * 2
        self.df['LowerBB'] = self.df[sma_str] - std * 2
        self.df['Bandwidth'] = (4*std)/self.df[sma_str]

    def plot(self, upper_bb=True, sma=True, lower_bb=True):
        ax1 = plt.subplot2grid((8, 1), (0, 0), rowspan=5,
colspan=1)
        ax2 = plt.subplot2grid((8, 1), (5, 0), rowspan=3,
colspan=1)

        candlestick_ohlc(ax1, self.df.values, width=0.6,
colorup='green',
                        colordown='red', alpha=0.8)

        if upper_bb:
            ax1.plot(self.df['UpperBB'], linestyle='--',
linewidth=1, color='black')

        if sma:
            ax1.plot(self.df.filter(regex='^SMA', axis=1),
linewidth='--', linewidth=1.2, color='grey')

        if lower_bb:
            ax1.plot(self.df['LowerBB'], linestyle='--',
linewidth=1, color='black')

        ax2.set_xlim(ax1.get_xlim())
        ax2.plot(self.df['Bandwidth'], linewidth=1, color='black')

        date_format = mpl_dates.DateFormatter('%b')
        ax2.xaxis.set_major_formatter(date_format)
        ax2.xaxis.set_major_locator(mpl_dates.MonthLocator())
        ax1.set(xticklabels=[]) # remove the tick labels
        ax1.tick_params(bottom=False) # remove the ticks
        plt.legend(loc='lower left')

        if self.title is not None:
            ax1.set_title(self.title)

        plt.show()
```

```
from bollinger_bands import BollingerBands  
  
oxy = BollingerBands("OXY")  
oxy.title = "Bande di Bollinger e Bandwidth"  
oxy.plot()
```



### A.3 Medie mobili

```
from libs import *

class MovingAverage(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)

    def sma(self, periodo, etichetta=None):
        return [self.df['Close'].rolling(window=periodo).mean(),
etichetta] if etichetta is not None else
self.df['Close'].rolling(window=periodo).mean()

    def ema(self, periodo, etichetta=None):
        return [self.df['Close'].ewm(span=periodo,
adjust=False).mean(), etichetta] if etichetta is not None else
self.df['Close'].ewm(span=periodo, adjust=False).mean()

    def plot(self, *args):
        fig, ax = plt.subplots(figsize=(16, 9))
        candlestick_ohlc(ax, self.df.values, width=0.6,
colorup='green',
                                colordown='red', alpha=0.8)

        for arg in args:
            plt.plot(arg[0], label=arg[1])

        date_format = mpl_dates.DateFormatter('%b')
        ax.xaxis.set_major_formatter(date_format)
        ax.xaxis.set_major_locator(mpl_dates.MonthLocator())
        plt.legend()

        if self.title is not None:
            plt.title(self.title)

        plt.show()

from moving_average import MovingAverage

erg = MovingAverage("ERG.MI")
erg.title = "Media mobile esponenziale"
sma = erg.sma(20, "SMA 20")
ema = erg.ema(20, "EMA 20")

erg.plot(ema)
```

## A.4 Fascio di medie mobili

```
from libs import *
from moving_average import MovingAverage

class FascioEma(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)
        self.ma = MovingAverage(titolo_azionario)
        self.ema_fast = self._get_ema_fast()
        self.ema_slow = self._get_ema_slow()

    def _get_ema_fast(self):
        ema_fast_periodi = [3, 5, 8, 1, 12, 15]
        ema_fast = []
        for periodo in ema_fast_periodi:
            ema_fast.append(self.ma.ema(periodo))
        return ema_fast

    def _get_ema_slow(self):
        ema_fast_periodi = [30, 35, 40, 45, 50, 60]
        ema_slow = []
        for periodo in ema_fast_periodi:
            ema_slow.append(self.ma.ema(periodo))
        return ema_slow

    def plot(self, colore_fast, colore_slow):
        fig, ax = plt.subplots(figsize=(16, 9))
        candlestick_ohlc(ax, self.df.values, width=0.6,
colorup="green",
                                colordown="red", alpha=0.8)
        width = 1

        for ema in self.ema_fast[:len(self.ema_fast)-1]:
            plt.plot(ema, linewidth=width, color=colore_fast)
        plt.plot(self.ema_fast[-1], linewidth=width,
color=colore_fast, label="Veloce")

        for ema in self.ema_slow[:len(self.ema_slow)-1]:
            plt.plot(ema, linewidth=width, color=colore_slow)
        plt.plot(self.ema_slow[-1], linewidth=width,
color=colore_slow, label="Lenta")

        date_format = mpl_dates.DateFormatter('%b')
        ax.xaxis.set_major_formatter(date_format)
        ax.xaxis.set_major_locator(mpl_dates.MonthLocator())
        plt.legend()

        if self.title is not None:
            plt.title(self.title)

        if self.marker is not None:
            plt.scatter(x=19047.166666666668, y=self.yaxis_marker,
marker=self.marker, color="purple")
            plt.annotate('Inizio conflitto - 24 Febbraio',
xy=(19047.166666666668, self.yaxis_marker),
xytext=(19017.166666666668,
```

```

self.yaxis_text),
                                bbox={'facecolor': 'purple', 'alpha':
0.2, 'pad': 2})

plt.show()

```

```

from fascio_ema import FascioEma

srs = FascioEma("SRS.MI")
srs.title = "Fascio di medie mobili - Saras S.p.A"
srs.marker = 7
srs.yaxis_marker = 1
srs.yaxis_text = 1.05
srs.plot("blue", "orange")

```

```

from fascio_ema import FascioEma

cvx = FascioEma("CVX")
cvx.title = "Chevron Corporation - Azienda petrolifera
statunitense"
cvx.marker = 7
cvx.yaxis_marker = 180
cvx.yaxis_text = 182
cvx.plot("blue", "orange")

```

## A.5 MACD

```
from libs import *

class MACD(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)
        self.df_macd = self._get_macd(26, 12, 9)

    def _get_macd(self, slow, fast, smooth):
        price = self.df['Close']
        exp1 = price.ewm(span=fast, adjust=False).mean()
        exp2 = price.ewm(span=slow, adjust=False).mean()
        macd = pd.DataFrame(exp1 - exp2).rename(columns={'Close':
'Macd'})
        signal = pd.DataFrame(macd.ewm(span=smooth,
adjust=False).mean()).rename(columns={'Macd': 'Signal'})
        hist = pd.DataFrame(macd['Macd'] -
signal['Signal']).rename(columns={0: 'Hist'})
        frames = [macd, signal, hist]
        return pd.concat(frames, join='inner', axis=1)

    def plot(self):
        prices = self.df
        macd = self.df_macd['Macd']
        signal = self.df_macd['Signal']
        hist = self.df_macd['Hist']

        ax1 = plt.subplot2grid((8, 1), (0, 0), rowspan=5,
colspan=1)
        ax2 = plt.subplot2grid((8, 1), (5, 0), rowspan=3,
colspan=1)

        candlestick_ohlc(ax1, prices.values, width=0.6,
colorup='green', colordown='red', alpha=0.8)

        ax2.plot(macd, color='blue', linewidth=1.5, label='MACD')
        ax2.plot(signal, color='orange', linewidth=1.5,
label='SEGNALE')

        for i in range(len(prices)):
            if str(hist[i])[0] == '-':
                ax2.bar(prices.index[i], hist[i], color='red')
                # ef5350 rosso
            else:
                ax2.bar(prices.index[i], hist[i], color='green')
                # 26a69a verde

        date_format = mpl_dates.DateFormatter('%b')
        ax2.xaxis.set_major_formatter(date_format)
        ax2.xaxis.set_major_locator(mpl_dates.MonthLocator())
        ax1.set(xticklabels=[]) # remove the tick labels
        ax1.tick_params(bottom=False) # remove the ticks
        plt.legend(loc='lower left')

        if self.title is not None:
            ax1.set_title(self.title)
```

```

        if self.marker is not None:
            ax1.scatter(x=19047.166666666668, y=self.yaxis_marker,
marker=self.marker, color="purple")
            ax1.annotate('Inizio conflitto - 24 Febbraio',
xy=(19047.166666666668, self.yaxis_marker),
                        xytext=(19017.166666666668,
self.yaxis_text),
                        bbox={'facecolor': 'purple', 'alpha':
0.2, 'pad': 2})

plt.show()

```

```

from macd import MACD

gzp = MACD("data_tradingview/gazprom.csv")
gzp.title = "PJSC Gazprom - Azienda petrolifera russa"
gzp.marker = 7
gzp.yaxis_marker = 350
gzp.yaxis_text = 358
gzp.plot()

```

```

from macd import MACD

shl = MACD("SHEL")
shl.title = "Shell plc - Azienda petrolifera britannica"
shl.marker = 7
shl.yaxis_marker = 57.5
shl.yaxis_text = 58.5
shl.plot()

```

## A.6 RSI

```
from libs import *

class RSI(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)
        self.df_rsi = self._get_rsi(14)

    def _get_rsi(self, periodo):
        df = self.df['Close'].copy().to_frame()

        df['Diff'] = df.diff(1)
        df['Gain'] = df['Diff'].clip(lower=0).round(2)
        df['Loss'] = df['Diff'].clip(upper=0).abs().round(2)
        df['AvgGain'] = df['Gain'].rolling(window=periodo,
min_periods=periodo).mean()[:periodo + 1]
        df['AvgLoss'] = df['Loss'].rolling(window=periodo,
min_periods=periodo).mean()[:periodo + 1]

        for i, row in enumerate(df['AvgGain'].iloc[periodo + 1:]):
            df['AvgGain'].iloc[i + periodo + 1] = \
                (df['AvgGain'].iloc[i + periodo] *
                 (periodo - 1) +
                 df['Gain'].iloc[i + periodo + 1]) \
                / periodo

        for i, row in enumerate(df['AvgLoss'].iloc[periodo + 1:]):
            df['AvgLoss'].iloc[i + periodo + 1] = \
                (df['AvgLoss'].iloc[i + periodo] *
                 (periodo - 1) +
                 df['Loss'].iloc[i + periodo + 1]) \
                / periodo

        df['Rs'] = df['AvgGain'] / df['AvgLoss']
        df['Rsi'] = 100 - (100 / (1.0 + df['Rs']))

        return df

    def plot(self, livelli):
        ax1 = plt.subplot2grid((8, 1), (0, 0), rowspan=5,
colspan=1)
        ax2 = plt.subplot2grid((8, 1), (5, 0), rowspan=3,
colspan=1)

        candlestick_ohlc(ax1, self.df.values, width=0.6,
colorup='green', colordown='red', alpha=0.8)
        ax2.plot(self.df_rsi['Rsi'], linewidth=1, color='black')
        ax2.set_xlim(ax1.get_xlim())
        x1, y1 = [ax2.get_xlim()[0], ax2.get_xlim()[1]],
[livelli[0], livelli[0]]
        x2, y2 = [ax2.get_xlim()[0], ax2.get_xlim()[1]],
[livelli[1], livelli[1]]
        ax2.plot(x1, y1, color='grey', linestyle='--')
        ax2.plot(x2, y2, color='grey', linestyle='--')

        ax1.set(xticklabels=[]) # remove the tick labels
```

```
ax1.tick_params(bottom=False) # remove the ticks

if self.title is not None:
    ax1.set_title(self.title)

date_format = mpl_dates.DateFormatter('%b')
ax2.xaxis.set_major_formatter(date_format)
ax2.xaxis.set_major_locator(mpl_dates.MonthLocator())

plt.show()
```

```
from rsi import RSI

bp = RSI("BP")
bp.title = "Oscillatore RSI"
bp.plot([35, 65])
```

## A.7 Stocastico

```
from libs import *

class Stocastico(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)
        self.df2 = self._get_stocastico(14, 3)

    def _get_stocastico(self, periodo_k, periodo_d):
        df = self.df.copy()
        df['n_high'] = df['High'].rolling(periodo_k).max()
        # Adds an "n_low" column with min value of previous 14
        # periods
        df['n_low'] = df['Low'].rolling(periodo_k).min()
        # Uses the min/max values to calculate the %k (as a
        # percentage)
        df['%K'] = (df['Close'] - df['n_low']) * 100 /
        (df['n_high'] - df['n_low'])
        # Uses the %k to calculates a SMA over the past 3 values
        # of %k
        df['%D'] = df['%K'].rolling(periodo_d).mean()

        # Add some indicators
        df.ta.stoch(high='High', low='Low', close='Close',
        k=periodo_k, d=periodo_d, append=True)

        # Avoid case-sensitive issues for accessing data.
        # Optional if using pandas_ta
        df.columns = [x.lower() for x in df.columns]

        return df

    def plot(self):
        ax1 = plt.subplot2grid((8, 1), (0, 0), rowspan=5,
        colspan=1)
        ax2 = plt.subplot2grid((8, 1), (5, 0), rowspan=3,
        colspan=1)

        candlestick_ohlc(ax1, self.df.values, width=0.6,
        colorup='green', colordown='red', alpha=0.8)
        ax2.plot(self.df2['stochk_14_3_3'], color='orange',
        label="Veloce")
        ax2.plot(self.df2['stochd_14_3_3'], color='blue',
        label="Lenta")

        ax2.set_ylim([0, 100])
        ax2.set_xlim(ax1.get_xlim())
        livelli = [20, 80]
        x1, y1 = [ax2.get_xlim()[0], ax2.get_xlim()[1]],
        [livelli[0], livelli[0]]
        x2, y2 = [ax2.get_xlim()[0], ax2.get_xlim()[1]],
        [livelli[1], livelli[1]]
        ax2.plot(x1, y1, color='grey', linestyle='--')
        ax2.plot(x2, y2, color='grey', linestyle='--')
        livelli = [30, 70]
        x1, y1 = [ax2.get_xlim()[0], ax2.get_xlim()[1]],
```



```

[livelli[0], livelli[0]]
    x2, y2 = [ax2.get_xlim()[0], ax2.get_xlim()[1]],
[livelli[1], livelli[1]]
    ax2.plot(x1, y1, color='grey', linestyle='-.')
    ax2.plot(x2, y2, color='grey', linestyle='-.')

    ax1.set(xticklabels=[]) # remove the tick labels
    ax1.tick_params(bottom=False) # remove the ticks
    date_format = mpl_dates.DateFormatter('%b')
    ax2.xaxis.set_major_formatter(date_format)
    ax2.xaxis.set_major_locator(mpl_dates.MonthLocator())
    plt.legend(loc='lower left')

    if self.title is not None:
        ax1.set_title(self.title)

    if self.marker is not None:
        ax1.scatter(x=19047.166666666668, y=self.yaxis_marker,
marker=self.marker, color="purple")
        ax1.annotate('Inizio conflitto - 24 Febbraio',
xy=(19047.166666666668, self.yaxis_marker),
xytext=(19017.166666666668,
self.yaxis_text),
bbox={'facecolor': 'purple', 'alpha':
0.2, 'pad': 2})

    plt.show()

```

```

from stocastico import Stocastico

tte = Stocastico("TTE")
tte.title = "TotalEnergies SE - Azienda petrolifera francese"
tte.marker = 7
tte.yaxis_marker = 61.5
tte.yaxis_text = 62.5
tte.plot()

```

## A.8 Supporti e resistenze

```
from libs import *

class SupportiResistenze(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)

    def _is_support(self, i):
        cond1 = self.df['Low'][i] < self.df['Low'][i - 1]
        cond2 = self.df['Low'][i] < self.df['Low'][i + 1]
        cond3 = self.df['Low'][i + 1] < self.df['Low'][i + 2]
        cond4 = self.df['Low'][i - 1] < self.df['Low'][i - 2]
        return (cond1 and cond2 and cond3 and cond4) # determine
bearish fractal

    def _is_resistance(self, i):
        cond1 = self.df['High'][i] > self.df['High'][i - 1]
        cond2 = self.df['High'][i] > self.df['High'][i + 1]
        cond3 = self.df['High'][i + 1] > self.df['High'][i + 2]
        cond4 = self.df['High'][i - 1] > self.df['High'][i - 2]
        return (cond1 and cond2 and cond3 and cond4) # to make
sure the new level area does not exist already

    def _is_far_from_level(self, value, levels):
        ave = np.mean(self.df['High'] - self.df['Low'])
        return np.sum(
            [abs(value - level) < ave for _, level in levels]) ==
0 # a list to store resistance and support levels

    def plot(self):
        levels = []

        for i in range(2, self.df.shape[0] - 2):
            if self._is_support(i):
                low = self.df['Low'][i]
                if self._is_far_from_level(low, levels):
                    levels.append((i, low))
            elif self._is_resistance(i):
                high = self.df['High'][i]
                if self._is_far_from_level(high, levels):
                    levels.append((i, high))

        pivots = []
        max_list = []
        min_list = []
        for i in range(5, len(self.df) - 5):
            # taking a window of 9 candles
            high_range = self.df['High'][i - 5:i + 4]
            current_max = high_range.max()
            # if we find a new maximum value, empty the max_list
            if current_max not in max_list:
                max_list = []
            max_list.append(current_max)
            # if the maximum value remains the same after shifting
5 times

            if len(max_list) == 5 and
```

```

self._is_far_from_level(current_max, pivots):
    pivots.append((high_range.idxmax(), current_max))

    low_range = self.df['Low'][i - 5:i + 5]
    current_min = low_range.min()
    if current_min not in min_list:
        min_list = []
    min_list.append(current_min)
    if len(min_list) == 5 and
self._is_far_from_level(current_min, pivots):
    pivots.append((low_range.idxmin(), current_min))

    fig, ax = plt.subplots(figsize=(16, 9))
    candlestick_ohlc(ax, self.df.values, width=0.6,
colorup='green',
                        colordown='red', alpha=0.8)

    i = 0

    for level in pivots:
        i += 1
        plt.hlines(level[1], xmin=self.df['Date'][level[0]],
xmax=
                    max(self.df['Date']), colors='blue', linestyle='-.')
        plt.annotate(i, xy=(self.df['Date'].max() + 5,
level[1]), xytext=(self.df['Date'].max() + 5, level[1]),
                    weight='bold',
                    bbox={'facecolor': 'purple', 'alpha':
0.2, 'pad': 2})

        if self.title is not None:
            plt.title(self.title)

        if self.marker is not None:
            plt.scatter(x=19047.166666666668, y=self.yaxis_marker,
marker=self.marker, color="purple")
            plt.annotate('Inizio conflitto - 24 Febbraio',
xy=(19047.166666666668, self.yaxis_marker),
                        xytext=(19017.166666666668,
self.yaxis_text),
                        bbox={'facecolor': 'purple', 'alpha':
0.2, 'pad': 2})

        if self.yaxis_limit is not None:
            ax.set_ylim([self.yaxis_limit, ax.get_ylim()[1]])

    date_format = mpl_dates.DateFormatter('%b')
    ax.xaxis.set_major_formatter(date_format)
    ax.xaxis.set_major_locator(mpl_dates.MonthLocator())

    plt.show()

```

```
from supporti_resistenze import SupportiResistenze

gzp = SupportiResistenze("data_tradingview/gazprom.csv")
gzp.title = "PJSC Gazprom - Azienda petrolifera russa"
gzp.marker = 7
gzp.yaxis_marker = 350
gzp.yaxis_text = 358
gzp.plot()
```

```
from supporti_resistenze import SupportiResistenze

luk = SupportiResistenze("data_tradingview/lukoil.csv")
luk.title = "PJSC Lukoil - Azienda petrolifera russa"
luk.marker = 6
luk.yaxis_limit = 2000
luk.yaxis_marker = 2700
luk.yaxis_text = 2450
luk.plot()
```

## A.9 Trendlines

```
from libs import *

class Trendlines(DownloadDati):
    def __init__(self, titolo_azionario):
        super().__init__(titolo_azionario)
        self._get_channel()

    def _get_channel(self):
        df_len = len(self.df)
        self.df['Number'] = np.arange(df_len) + 1
        df_high = self.df.copy()
        df_low = self.df.copy()

        while len(df_low) > 2:
            slope, intercept, r_value, p_value, std_err =
linregress(x=df_low['Number'], y=df_low['Low'])
            df_low = df_low.loc[df_low['Low'] < slope *
df_low['Number'] + intercept]

        while len(df_high) > 2:
            slope, intercept, r_value, p_value, std_err =
linregress(x=df_high['Number'], y=df_high['High'])
            df_high = df_high.loc[df_high['High'] > slope *
df_high['Number'] + intercept]

        slope, intercept, r_value, p_value, std_err =
linregress(x=df_high['Number'], y=df_high['Close'])
        self.df['Downtrend'] = slope * self.df['Number'] +
intercept

        slope, intercept, r_value, p_value, std_err =
linregress(x=df_low['Number'], y=df_low['Close'])
        self.df['Uptrend'] = slope * self.df['Number'] + intercept

    def plot(self, downtrend=True, uptrend=True):
        fig, ax1 = plt.subplots(figsize=(16, 9))
        candlestick_ohlc(ax1, self.df.values, width=0.6,
colorup='green',
                                colordown='red', alpha=0.8)

        ax2 = ax1.twinx() # ax2 and ax1 will have common y axis
and different x axis, twiny

        if downtrend:
            ax2.plot(self.df['Number'], self.df['Downtrend'],
label="Downtrend", color="orange")

        if uptrend:
            ax2.plot(self.df['Number'], self.df['Uptrend'],
label="Uptrend", color="blue")

        plt.legend()
        plt.tick_params(left=False, top=False, labelleft=False,
labeltop=False)
        date_format = mpl_dates.DateFormatter('%b')
```

```

        ax1.xaxis.set_major_formatter(date_format)
        ax1.xaxis.set_major_locator(mpl_dates.MonthLocator())

        if self.title is not None:
            ax1.set_title(self.title)

        if self.marker is not None:
            ax1.scatter(x=19047.166666666668, y=self.yaxis_marker,
marker=self.marker, color="purple")
            ax1.annotate('Inizio conflitto - 24 Febbraio',
xy=(19047.166666666668, self.yaxis_marker),
                        xytext=(19017.166666666668,
self.yaxis_text),
                        bbox={'facecolor': 'purple', 'alpha':
0.2, 'pad': 2})

        plt.show()

from trendlines import Trendlines

oxy = Trendlines("OXY")
oxy.title = "Occidental Petroleum Corporation - Azienda
petrolifera statunitense"
oxy.marker = 7
oxy.yaxis_marker = 63
oxy.yaxis_text = 64.5
oxy.plot(downtrend=False)

```

## **BIBLIOGRAFIA**

Dmitry Zinoviev, Data Science con Python, (Milano: Apogeo, 2016)

Jansen Stefan, Machine Learning for Algorithmic Trading, (Birmingham: Packt, 2020)

John J. Murphy, Analisi tecnica dei mercati finanziari, (Milano: Hoepli, 2002)

Maurizio Mazziero, Guida all'analisi tecnica, (Milano: Hoepli, 2014)

Robert Johansson, Numerical Python, (California: Apress Media, 2019)

Thomas N. Bulkowski, Encyclopedia of Chart Patterns, (Hoboken: John Wiley & Sons, Inc., 2021)

## SITOGRAFIA

Analytics Vidhya, <https://analyticsvidhya.com/>, (ultima visita il 28 novembre 2022)

BorsaMercato, <https://borsamercato.com/>, (ultima visita il 20 novembre 2022)

Bruno Moltrasio, <https://brunomoltrasio.eu/>, (ultima visita il 29 dicembre 2022)

global witness, <https://www.globalwitness.org>, (ultima visita il 10 gennaio 2023)

HTML.it, <https://html.it>, (ultima visita il 5 dicembre 2022)

IG Italia, <https://ig.com/it/>, (ultima visita il 28 novembre 2022)

Il Sole 24 Ore, <https://ilsole24ore.com/>, (ultima visita il 20 dicembre 2022)

iMerit, <https://imerit.net/>, (ultima visita il 14 dicembre 2022)

Investopedia, <https://investopedia.com/>, (ultima visita il 24 novembre 2022)

KDnuggets, <https://kdnuggets.com/>, (ultima visita il 28 novembre 2022)

Money.it, <https://money.it>, (ultima visita il 20 gennaio 2023)

Python.org, <https://python.org/>, (ultima visita il 5 dicembre 2022)

Wikipedia, <https://it.wikipedia.org/>, (ultima visita il 20 novembre 2022)



## **RINGRAZIAMENTI**

Un ringraziamento speciale va al mio relatore Francesco Orciuoli, per avermi insegnato molto e per avermi permesso di studiare una materia affascinante e interessante, una delle migliori del corso.

Un ringraziamento a mio padre, per avermi sempre spronato e per avermi insegnato a pensare positivo ed essere lungimirante.

Un ringraziamento a mia madre, per avermi aiutato in qualsiasi circostanza e per avermi insegnato a rispettare e amare il prossimo.

Un ringraziamento a mia sorella Alessia, per avermi dimostrato cosa significhi davvero volersi bene ed essere fratelli.

Un ringraziamento alla mia splendida fidanzata Concetta, per avermi sostenuto durante tutto il percorso universitario e per continuare a farlo ogni giorno, comunque vada. Senza di te tutti i piccoli successi, come quello di oggi, non sarebbero stati possibili. Grazie di cuore.

Un ringraziamento a mia nonna Annamaria, per il suo affetto mai mancato.

Un ringraziamento ai miei nonni Raffaello, Francesco e Raffaella per il loro affetto e per, ne sono certo, la loro protezione da lì su.

Un ringraziamento alle mie amiche Clara, Claudia, Manuela e a Pietro per aver allietato le mie giornate, da un bel po' di anni, con la vostra fedele compagnia.

Un ringraziamento ai miei colleghi di corso Angelo, Antonio, Emanuele e Sara per avermi aiutato a finire questo percorso e per la vostra compagnia durante questi tre fantastici anni.

Un ringraziamento ai volontari della Misericordia di Salerno per avermi permesso di studiare con serenità e per avermi permesso, insieme e in allegria, di rendere un utile servizio alla società.