

# Capitolo 1

## Progetto e Obiettivo

### 1.1 Sommario esecutivo

Il nostro obiettivo principale è sviluppare un modello di Recurrent Neural Network (RNN) in Python che analizzi le correlazioni all'interno del NASDAQ e predica i futuri prezzi di chiusura delle azioni. Utilizzeremo il linguaggio di programmazione Python e la libreria di machine learning Keras per implementare il nostro modello. L'obiettivo è fornire previsioni dettagliate che contribuiscano a una presa di decisioni informata nel dinamico panorama degli investimenti di borsa.

### 1.2 Output atteso

La visione del progetto è presentare un modello completo entro la metà di giugno. Inoltre, esploreremo diverse applicazioni del modello, tra cui la possibilità di implementare un sito web informativo, la redazione di un articolo scientifico, la creazione di un corso introduttivo, e la prospettiva di collaborazioni future con altri club, come ad esempio il club di informatica del POLIMI.

### 1.3 Motivazioni

Il progetto costituisce un'opportunità eccellente per consolidare le conoscenze e le competenze dei membri della Quantitative Finance Division. Sarà un punto di riferimento organizzativo per i prossimi progetti, fornendo una conclusione che aggiunge notevole valore ai curriculum dei membri, arricchendo il loro bagaglio personale. Inoltre, ci consentirà di inserirci nel contesto della finanza quantitativa.

## Capitolo 2

# Tasks e Risorse

### 2.1 Individuazione del metodo

Per iniziare, ho adottato una strategia bottom-up per identificare i compiti chiave del progetto. I passaggi chiave includono l'introduzione all'economia (per fornire a tutti la possibilità di avere delle conoscenze di partenza uguali), la selezione delle librerie e la preparazione del dataframe, la pulizia dei dati, la costruzione del modello, i test e l'implementazione utilizzando Streamlit o la stesura di un paper.

Nel capitolo 3, è possibile leggere ogni step in modo più dettagliato, e con le rispettive tempistiche.

### 2.2 Prerequisiti

E' essenziale consultare la letteratura già esistente sull'argomento al fine di posizionarci rispetto alla presenza di questo tipo di modello di ML (machine learning) nel mondo accademico. Esistono numerose risorse dove accedere ai papers già scritti, anche quelli a pagamento, in modo gratuito. L'ideale, assieme a un'introduzione ai concetti matematici ed economici dietro al progetto, sarebbe dedicare del tempo per leggere e approfondire sulle pratiche già utilizzate per ottenere l'output desiderato da noi, ed evitare difficoltà inutili.

Papers e articles già scritti e di *scope* simile al nostro sono i seguenti:

- [The forecasting of consumer exchange-traded funds \(ETFs\) via grey relational analysis \(GRA\) and artificial neural network \(ANN\) by M. Malinda e J. Chen](#)
- [Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model by H. K. Choi.](#)
- [Time Series Forecasting with RNNs by Marek Galovic](#)
- [Deep Learning for ETF Price Prediction by Dickson Chan, Michael Hsieh, Sophie Pan](#)
- [ARIMA Model – Complete Guide to Time Series Forecasting in Python by Selva Prabhakaran.](#)
- [Neural Networks for Financial Time Series Forecasting by Kady Sako, Berthine Nyunga Mpinda, and Paulo Canas Rodrigues.](#)

Questo periodo di ricerca e studio sarebbe il primo step del progetto.

Per quanto riguardano le skills necessarie in modo da concludere il progetto, ho individuato le seguenti:

- **Matematici:** conoscenza base di Python, conoscenza  $\text{\LaTeX}$  Overleaf e Zotero, conoscenza inglese almeno B2 (il paper verrebbe scritto in inglese).
- **Programmatori:** conoscenza media di Python, saper utilizzare **bene** Numpy, Pandas, Matplotlib, Seaborn. Conoscenza base con Keras. Conoscere bene le basi dietro alle RNN, e in generale al ML. Saper utilizzare [Jupyter](#) notebooks, GitHub, e Git. Conoscenza inglese almeno B1.

Ci sono risorse online attraverso le quali imparare le skills necessarie.

## 2.3 Frameworks, API e linguaggi scelti per lo sviluppo del modello

In seguito a un semestre di ricerca personale, e dopo aver contattato sia professionisti nel settore della programmazione, che PhD's in Machine Learning e finanza, ho individuato i seguenti software che risultano essere i migliori, tenendo conto delle risorse disponibili (discusse in modo più dettagliato nel prossimo capitolo).

- Linguaggio di programmazione: Python.
- Source code editor: [Vscod](#).
- Librerie: [pandas](#), [Numpy](#), [Seaborn](#), [matplotlib](#), [Keras](#), [Streamlit](#), [yfinance](#) <sup>1</sup>.
- Paper writing platform: Overleaf & Zotero.
- Literature: [Hands-on Machine Learning with Keras](#), libri inviati dal team di matematici, risorse trovate online.

## 2.4 Risorse a disposizione e tempo

Il nostro team è composto da due matematici, due programmatori e un programmatore/manager. Con 75 giorni lavorativi dal 10 marzo al 1 giugno, e considerando un impegno di 4 ore a settimana per ciascun membro, disponiamo di un totale di 48 ore. L'intenzione è di lavorare simultaneamente su diverse attività, in modo da avere un workforce di circa 200 ore.

Tutti dovranno essere disponibili almeno tre volte al mese per i meetings, e tutti devono essere muniti di un computer abbastanza potente da sopportare i softwares discussi nel Capitolo 1.

## 2.5 Possibili sfide

Affronteremo anche le possibili sfide, come l'utilizzo di un database non ufficiale e quindi non affidabile. La fase del progetto relativa alla selezione e pulizia dei dati è cruciale per il successo dell'intero lavoro.

---

<sup>1</sup>La libreria da cui prendere il database verrà scelta insieme dopo l'inizio del progetto, anche se attualmente [yfinance](#) viene molto utilizzata.

Un'ulteriore sfida potrebbe derivare dalla potenziale mancanza di omogeneità nelle competenze tra i membri del team, troppo poco tempo, e la complessità del lavoro.

I meeting saranno cruciali per superare qualsiasi difficoltà e mantenere coerenza tra tutte le parti coinvolte. Il head dovrà essere consapevole delle prestazioni di ogni membro e possedere una solida base di conoscenze in modo da poter consigliare tutti i team e fornire soluzioni. Sarà responsabile di guidare la gestione delle tempistiche, organizzare i ruoli e assegnare compiti in modo efficiente.

## Capitolo 3

# Timeline

Ho utilizzato il Critical Path Method (CPM) per creare un diagramma di Gantt, fornendo dettagli su ogni passo del progetto. Questo strumento ci aiuterà a pianificare e monitorare le attività nel corso del tempo. Ogni membro del team può consultare il diagramma per comprendere in dettaglio le fasi del progetto. Inoltre, discuteremo la distribuzione delle risorse e del personale in ogni fase, identificando le dipendenze tra le attività e le possibili zone di rischio.

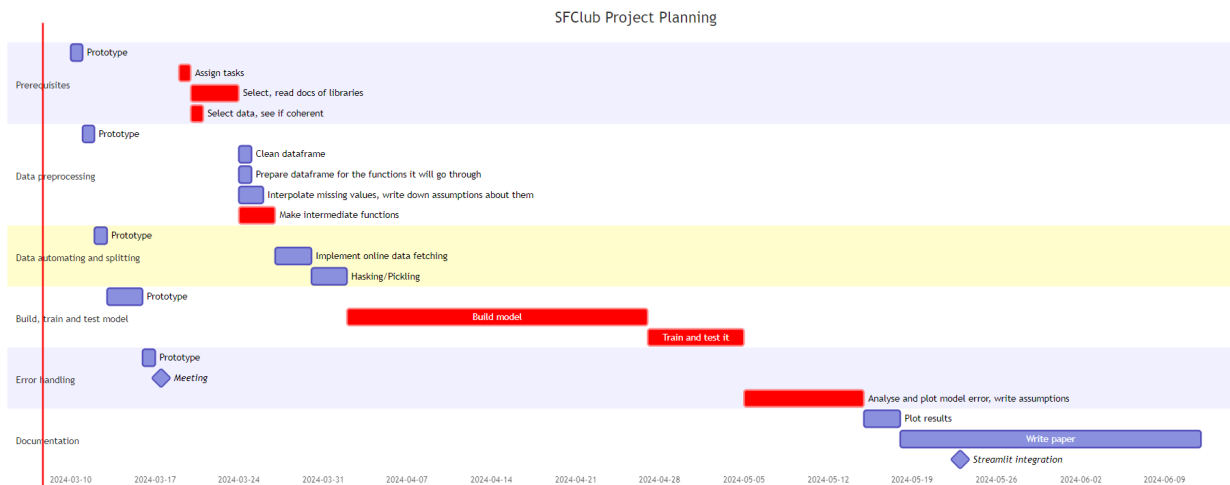


Figura 3.1: Gantt chart with CPM.

Realisticamente, il progetto richiederà tre mesi per essere concluso, e per evitare di sovrapporsi con la preparazione individuale alla sessione d'esami, è essenziale che tutti i membri partecipino attivamente e indipendentemente in modo efficace. Questo approccio mira a minimizzare la necessità di revisioni o correzioni che potrebbero rallentarci e non garantire una progressione fluida durante il periodo di lavoro.

Un'incontro a settimana sarebbe consigliabile per sostenere un andamento costante e fluido del progetto.

### 3.1 Roadmap

Per sviluppare il modello, seguendo una roadmap ben definita, inizierei con una (F1) fase introduttiva dedicata all'approfondimento dei concetti matematici ed economici alla base del progetto.

La proposta sarebbe quella di avviare successivamente una (F2) fase di prototipazione, in cui si svilupperebbe un progetto di dimensioni più contenute, ma sempre pertinenti e rappresentative delle funzionalità previste dal modello ufficiale. La decisione di dedicare del tempo a questa fase è principalmente motivata da due convinzioni: questa settimana fungerebbe da fase di valutazione, in cui i ruoli verrebbero definiti, i concetti studiati ufficialmente. Ciò consentirebbe a tutti di decidere se sono effettivamente interessati al progetto o meno, apprendere le sinergie tra i membri e fornire un tutorial sui metodi e modelli; la seconda convinzione è, soprattutto, perché nella realtà del *software e product development*, la strategia che prevede l'omissione di prototipizzazione e iterazioni (nota come "[Waterfall](#)") si è dimostrata inefficace. Attualmente, la strategia più diffusa è l'[Agile](#), ma risulta spesso troppo impegnativa data la limitata disponibilità temporale dei membri. Pertanto, la fase di prototipizzazione sarebbe un compromesso, con l'aspettativa di emergere più consapevoli delle fasi di sviluppo, più esperti degli argomenti e con la capacità di evitare errori iniziali, prevenendo così il cosiddetto "*technical debt*".

Successivamente, procederei alla (F3) selezione delle librerie e alla preparazione del dataframe, focalizzandomi sull'efficienza e sulla gestione dei dati. La (F4) pulizia dei dati rappresenterebbe una tappa fondamentale per garantire l'integrità e la qualità dell'input.

Dopo la selezione e la preparazione del dataframe, prevederei una fase dedicata (F5) all'hashing o al web scraping, a seconda delle esigenze, per garantire un accesso affidabile e aggiornato ai dati necessari. L'utilizzo di tecniche di hashing potrebbe contribuire alla sicurezza e all'integrità dei dati, mentre il web scraping potrebbe essere implementato per l'estrazione diretta di informazioni rilevanti da fonti online. Inoltre, considererei l'utilizzo di Pickle per la serializzazione dei dati, semplificando il processo di memorizzazione e caricamento dei dati pre-processati, migliorando così l'efficienza complessiva del flusso di lavoro.

Dopo questa fase, mi concentrerei sulla (F6) costruzione del modello RNN utilizzando la libreria Keras, applicando le migliori pratiche nel machine learning. Successivamente, (F7) testerei il modello per valutarne l'efficacia e l'accuratezza, adattandolo di conseguenza.

Dopo la fase di testing del modello, introdurrei una sezione dedicata (F8) all'analisi dei risultati e allo studio degli errori. Inoltre, si potrebbero formulare assunzioni riguardo agli errori osservati, sottolineando la necessità di future ottimizzazioni e ulteriori ricerche nel contesto del progetto.

Infine, (F9) implementerei il modello utilizzando Streamlit o procederei con la stesura del paper, assicurandomi di documentare accuratamente ogni passo e risultato ottenuto durante il processo di sviluppo.

## 3.2 Report: Example of prototype

Successivamente, troverai il link al Gist dove è possibile visualizzare un possibile prototipo. Ho realizzato questo prototipo da sola e, naturalmente, serve solo come esempio: [Prototype example](#).