UNIVERSITÀ DEGLI STUDI DI PERUGIA

DIPARTIMENTO DI INGEGNERIA

Analisi di dati azionari storici attraverso l'utilizzo di Hadoop MapReduce

Docente

Fabrizio Montecchiani

Studente

Giacomo Amato

Anno accademico 2024/2025

Indice

1 Introduzione

Lo scopo di questo progetto è quello di andare ad effettuare un'analisi riguardante sia i dati storici dei prezzi azionari sia fare un'analisi delle informazioni aziendali che sono fornite nei dataset Daily historical stock prices. Abbiamo due dataset historical_stock_prices.csv in cui sono contenuti i dati riguardanti i ticker che rappresentano l'identificativo di un'azione, open e close che rappresentano i prezzi di apertura e chiusura, adj_close per indicare il prezzo di chiusura aggiustato, low e high che indicano il prezzo più alto e più basso raggiunto dall'azione, volume per il volume di azioni scambiate e date. Nel secondo dataset historical_stocks troviamo ticker, nome dell'azienda, l'exchange, il settore della compagnia e l'industria.

Andando a processare questi dati facendo uso di Hadoop MapReduce installato in un ambiente linux in modalità psudo-distribuita e con input e output da HDFS, possiamo ricavare un sottoinsieme di essi che ci servirà poi per le successive analisi, che sono la valutazione della volatilità di un titolo ovvero la misura delle fluttuazioni del suo prezzo nel tempo, il trend settoriale annuale di un titolo e a fare l'analisi di un singolo titolo.

Per fare ciò si utilizzano 4 job MapReduce il primo quali prende in input da HDFS i file csv precedentemente citati li elabora e ne ricaveremo poi un file csv, che andrà in input ai 3 job riguardanti le analisi dei dati.

Infine i dati risultanti da questi ultimi verrano prima convertiti da file txt a csv e poi mostrati attraverso heatmap e grafici con degli script python.

La macchina virtuale utilizzata per questa analisi utilizza il sistema operativo Linux Ubuntu 24.04 a cui sono stati assegnati 8 GB di RAM e 30 di storage, e viene sfruttato Hadoop 3.3.6 in modalità pseudo-distribuita.

2 DataFlow, tecnologie utilizzate e casi d'uso

Per poter realizzare il progetto è stato sfruttato **Apache Hadoop** nella versione 3.3.6, il quale è stato utilizzato per la costruzione del cluster e HDFS come tecnologia per il file system distribuito e YARN come resource negotiator.

il cluster è stato configurato in modalità pseudo-distribuita su singolo nodo in cui abbiamo il Namenode assegnato all'utente appositamente creato nella macchina virtuale con il nome di hadoop, il secondary datanode assegnato all'altro utente della macchina virtuale e il resource manager necessari per poter eseguire HDFS e YARN oltre ai datanode e al node manager.

Per qunato riguarda la configurazione si può vedere in maniera più approfondita nel file README presente su github.

Quello che andiamo a fare per poter analizzare i dati è un'iniziale fase di pre-processing in cui i dati che provengono dai due dataset historical_stock_prices che contiene i seguenti attributi:

- ticker: simbolo per l'azione;
- open: prezzo di apertura;
- close: prezzo di chiusura;
- adj_close: prezzo di chiusura corretto;
- low: prezzo più basso;
- high: prezzo più alto;
- volume: il volume di azioni scambiate;
- date: la data

e il dataset historical_stocks che contiene gli attributi:

- ticker: simbolo per l'azione;
- exchange: il nome dell'exchange;
- name: il nome dell'industria;
- sector: il settore della compagnia;
- industry: l'industria a cui appartiene la compagnia.

vengono uniti attraverso l'ausilio di un apposito job MapReduce che ne fa il join, in un singolo risultante file csv che ha come colonne d'interesse ticker, name, date, close, volume, sector. I dataset utilizzati vengono caricati su HDFS nella cartella stock_input, e il dataset historical_prices.csv verra caricato nella cache.

Una volta finita la fase di processing l'output risultante verrà unito e rinominato appropriatamente, e verrà quindi utilizzato succeccivamente come input per i successivi 3 job.

Per andare a fare le analisi d'interesse utilizziamo 3 job:

- volatilità: in questo job andiamo a prendere in input il file csv ottenuto dall'output del preprocessing, e andiamo ad utilizzare le colonne sector, date, year che viene ricavata dalla data e close per andare a stimare la volatilità (o deviazione standard) del settore in un anno e ne valutiamo anche il prezzo medio;

- trend settoriale: in questo caso andiamo ad utilizzare anche qui le colonne precedentemente citate e in più anche quella di ticker e volume per andare a stimare il volume totale di azioni scambiate, il prezzo medio giornaliero e la variazione percentuale dei vari settori per ogni anno;

- analisi singolo ticker: andiamo qui invece ad eseguire un analisi per singolo ticker la variazione percentuale, il prezzo iniziale e quello finale annuale.

Fatto ciò i dati ricavati dai precedenti job vengono salvati in un formato txt e poi successivamente resi in un formato csv attraverso l'utilizzo di uno script python in modo tale da renderli utilizzabili per una possibile analisi grafica, in quanto sennò si avrebbero in un formato che è del tipo "Volatilità(StdDev)=".

Con un ulteriore script python si vanno a mostrare delle heatmap e un grafico a barre dei dati ottenuti.

Prendiamo come esempio il job per il calcolo della volatilità, esso ha una fase iniziale in cui il mapper va a prendere in input i dati risultanti dal preprocessing andando a considerare gli attributi sector, date, year che viene ricavato e close, e poi va a formare la chiave sfruttando sector e year, e come valore prende close. Successivamente abbiamo la fase del reducer in cui inizialmente ci ricaviamo tutto ciò che ci serve per il calcolo della volatilità, che viene quindi calcolata e poi andrà a creare l'output che avrà stessa key che riceve in input ma i valori saranno volatilità e prezzo medio.

3 Limiti e possibili estensioni

Attualmente il progetto va ad utilizzare un dataset statico che non viene aggiornato in tempo reale e che quindi contiene i dati fino ad una certa data, si potrebbe andare quindi ad integrare un sistema che scarica quotidianamente i dati, si potrebbero fare analisi più avanzate e si potrebbero fare anche delle previsioni andando ad utilizzare degli algoritmi di machine learning sfruttando Apache Spark e pySpark.

In questo momento il cluster gira su nodo singolo quindi si potrebbe andare a estendere facendolo diventare un cluster multinodo in modo tale da poter gestire file di dimensioni maggiori, aumentando quindi la scalabilità, e avendo a disposizione più nodi avremo anche un miglioramento delle performace.