

Stock Prediction

SCELTA DATASET

Grazie alla "Stock Market Prediction" riusciremo a predire il valore futuro di un'azione. Le predizioni sono molto difficili da accurare dato che non possono essere determinate solamente con l'andamento dei prezzi della stock perché la loro sbilanciamento è causato da numerosi fattori come fisici, psicologici, razionali ed irrazionali. Questi fattori rendono il prezzo delle azioni volatili ed imprevedibili con la conseguenza dell'impossibile predizione dei loro valori con un alto livello di accuratezza.

MODELLO

Per creare un sistema di "Stock Prediction" andremo ad utilizzare un modello di regressione lineare.

In una stock prediction, la regressione lineare può essere utilizzata per prevedere il prezzo futuro di una determinata azione sulla base delle caratteristiche dell'azione e dei dati storici relativi al suo prezzo e alle variabili indipendenti. Ad esempio, si potrebbe utilizzare la regressione lineare per valutare la relazione tra il prezzo di un'azione e il volume degli scambi, il tasso di crescita dei guadagni, la capitalizzazione di mercato, e altri fattori rilevanti per il mercato.

La regressione lineare è spesso utilizzata in ambito finanziario perché fornisce una semplice e intuitiva analisi della relazione tra le variabili e il target da prevedere, consentendo inoltre di identificare le variabili che influiscono maggiormente sul prezzo delle azioni. Inoltre, la regressione lineare può essere facilmente implementata e calcolata con software statistici come Excel, R o Python.

PERCENTUALE ACCURATEZZA MINIMA

Considerando che non si ha il quadro generale di tutte le variabili che modificano il valore delle azioni ritengo che il valore di percentuale di accuratezza minima sia del 70%.

PERCENTUALE DATASET

Abbiamo deciso di suddividere il dataset in 2 categorie:

- un dataset di addestramento (training dataset) : esso comprenderà i valori di apertura (Open), i valori più alti (High), i valori più bassi (Low), i valori di chiusura (Close), i valori di chiusura modificati (Adjusted Close) e il volume delle azioni comprate in quella giornata. Essendo i dati di training essi saranno presenti in quantità maggiori rispetto ai dati di testing. Abbiamo preso di conseguenza tutti i dati dal 29-06-2010 fino al 15-03-2019 cioè esattamente 2194 dati tabellari contenuti all'interno di un file csv estratto da una comune piattaforma di trading relativo alle azioni Tesla. Verranno riservati il 70% dei dati.
- Un dataset di prova (testing dataset) : questo dataset invece ha le stesse colonne di valori del primo con la differenza che i dati saranno relativi a dopo il 15-03-2019 così che il programma non si sia allenato precedentemente su quei dati. Verranno riservati il 30% dei dati.

SVOLGIMENTO

Abbiamo innanzitutto importato le librerie necessarie per la creazione della nostra rete neurale, manipolazione dei dati e visualizzazione dei grafici ovvero pandas, numi matplotlib, chart_studio e plotly.

Successivamente abbiamo importato il dataset delle azioni tesla e visualizzato le informazioni attraverso il comando `head()` per verificare l'importo effettivo dei dati.

Con il secondo comando ovvero `info()` andiamo a verificare il numero di colonne importate e se sono presenti dati nulli e il loro relativo datatype.

Il passo successivo è quello di trasformare le nostre date con valore di stringhe in veri valori datetime utilizzando quindi una classica manipolazione dei dati con pandas.

Con la funzione `describe()` ci restituisce il count, mean, std, min e % che ci servono per capire come sono organizzati i nostri dati.

Count = numero totale di colonne

Mean = valore medio

Std = la deviazione standard

Max = valore massimo

Ora si andrà a sviluppare il modello di regressione lineare con l'importo iniziale di alcune librerie necessarie per il preprocessing e model evaluation per trovare l'accuratezza.

Prima di creare il modello bisogna ovviamente slittare i dati in dati di training e dati di test assegnando ad ogni dataset una certa percentuale di divisione.

Per scalare il dataset usiamo una funzione pre fatta utile per le performance di rendimento del dataset e successivamente dopo aver importato la funzione di regressione lineare si andrà a creare un modello lineare al quale verranno passati i dati di training x-y training.

Creo quindi il modello andremo a visualizzarlo su un ulteriore grafico dove si andrà a comparare le azione effettive e quelle predette dal modello lineare.

La linea raffigura la predizione del modello.

Ultimo step è quello di calcolare lo score del nostro modello in particolare abbiamo calcolato il MSE ovvero la funzione di rischio che misura il quadrato degli errori e il r square che identifica quanto si adattano i dati al modello utilizzato.

Abbiamo deciso di sviluppare il nostro applicativo all'interno di Android Studio utilizzando il linguaggio di programmazione Java. Le ragioni che ci hanno spinto ad utilizzare questo software con quel determinato linguaggio sono la nostra familiarità con esso.

Le varie fasi di sviluppo si possono dividere in queste seguenti:

- A : sviluppo schermata iniziale con breve introduzione all'applicativo parte front-end
- B : sviluppo schermata secondaria con la possibilità di predire una cerca azione tesla con un determinato periodo parte front-end
- C : sviluppo schermata secondaria parte back-end. La quale consiste nel ricevere le API precedentemente create. La parte C consiste nel inviare API relative ai risultati ottenuti dall'esperimento.
- D : filtrare le API in base alla data da predire.
- E : setup di AWS per la creazione delle API
- F : inserimento del codice precedentemente creato su Python per la stock prediction.
- G : aggiunta di un return per evidenziare quali saranno i parametri json che l'API restituirà.

ATTIVITA'	COSTO (h)	PREDECESSORE
A	1	-
B	3	A
C	4	F
D	2	C
E	3	A
F	5	E
G	1	F

[illegible]