TUTTO QUELLO CHE TI SERVE PER LAVORARE
E STUDIARE L'APPRENDIMENTO AUTOMATICO

MACHINE LEARNING

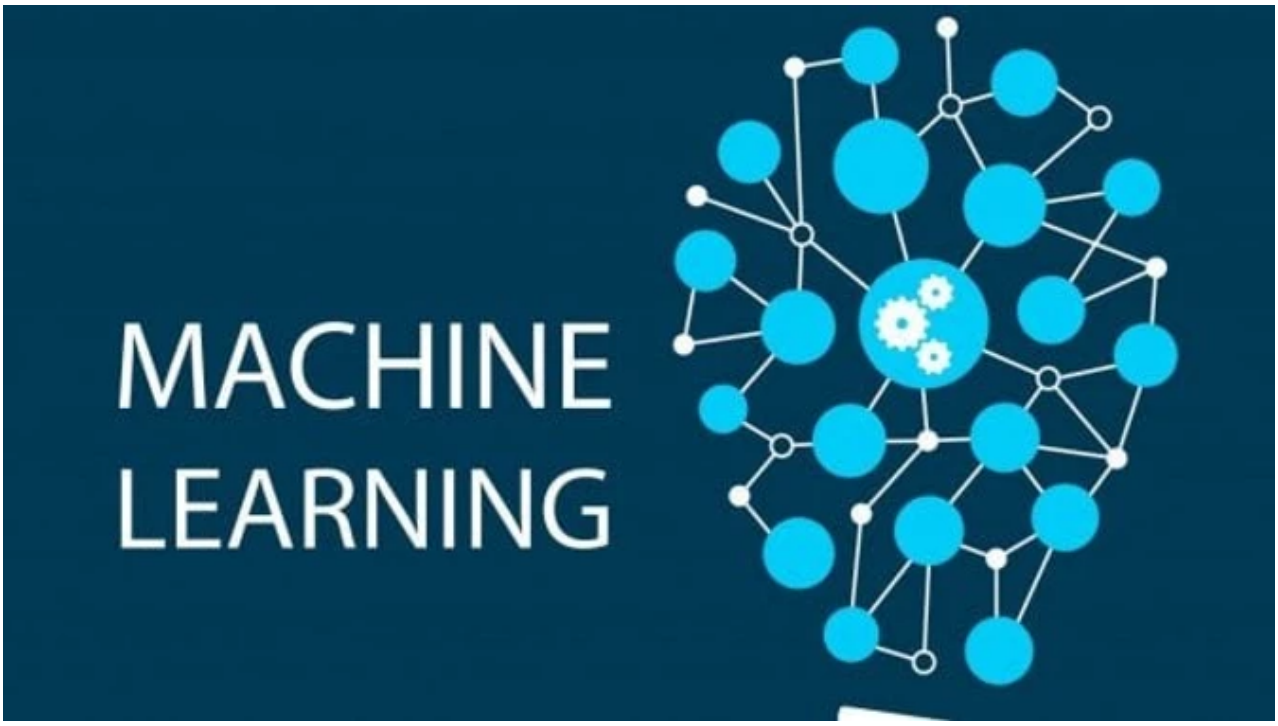
LA GUIDA UFFICIALE COMPLETA

INTELLIGENZAARTIFICIALEITALIA.NET

Che cosa è il Machine Learning (ML)

Premessa per chi già conosce, studia o lavora nel settore dell'intelligenza artificiale : Questa definizione di machine learning è stata semplificata per chi è ai primi approcci con la materia.
Successivamente verranno pubblicati molti approfondimenti ed esempi pratici,

Buona Lettura.



Machine Learning

Il **Machine Learning** è una branca dell'intelligenza artificiale.

La traduzione in italiano è "apprendimento delle macchine", il che letto così è anche un po' inquietante.

Questo termine venne usato per la prima volta nel 1959 da un ricercatore americano, il quale aveva supposto che se ad un computer è possibile fornirgli degli esempi accurati e ben lavorati riguardanti un determinato campo (ad esempio dati riguardanti le case, dati riguardanti cartelle cliniche) allora questo sarebbe stato in grado, tramite algoritmi complessi, di fare delle stime o delle classificazioni su dati che non aveva mai visto, basandosi solo sui dati degli esempi iniziali.

Effettivamente ciò, in parte, è risultato vero.

Una seconda definizione è stata data da **Tom Michael Mitchell**, direttore del dipartimento Machine Learning della Carnegie Mellon University:

> *Si dice che un programma apprende dall' esperienza **E** con riferimento a alcune classi di compiti **T** e con misurazione della performance **P**, se le sue performance nel compito **T**, come misurato da **P**, migliorano con l'esperienza **E***

Questa definizione più "**matematica**" in altre parole dice che:

se un programma migliora i suoi ***risultati/performance*** (ad esempio la stima del prezzo di una casa) in base alla quantità di ***esempi*** su cui si è allenato allora questo programma è in grado di ***apprendere.***

Ma vediamo come sia possibile una tale diavoleria.

Alla base di queste stime e classificazioni basate su grandi datasets (grandi quantità di esempi, composti da una serie di attributi e un target) abbiamo la STATISTICA E PROBABILITÀ.

Come appena detto il computer si basa su una serie di esempi, composti da una serie di etichette "**indipendenti**" che descrivono o classificano una variabile di **target**, in statistica chiamata variabile "**dipendente**"

Variabili dipendenti e indipendenti
Successivamente grazie a complesse formule di probabilità e statistica sugli esempi passati vennero creati modelli matematici dove ad ogni variabile indipendente sarà corrisposto un "**peso**" per la stima finale.

Ad esempio per quanto riguarda il prezzo di una casa la variabile che ha un peso maggiore, e cioè incide maggiormente, è la Dimensione della casa.

Supponiamo adesso di voler "passare" ad un **algoritmo di machine learning** due colonne di dati (le dimensioni della casa e il suo valore) aventi il seguente formato :

Dimensioni | **Prezzo**

40 m^3 | 50.000$

60 m³ | 70.000$

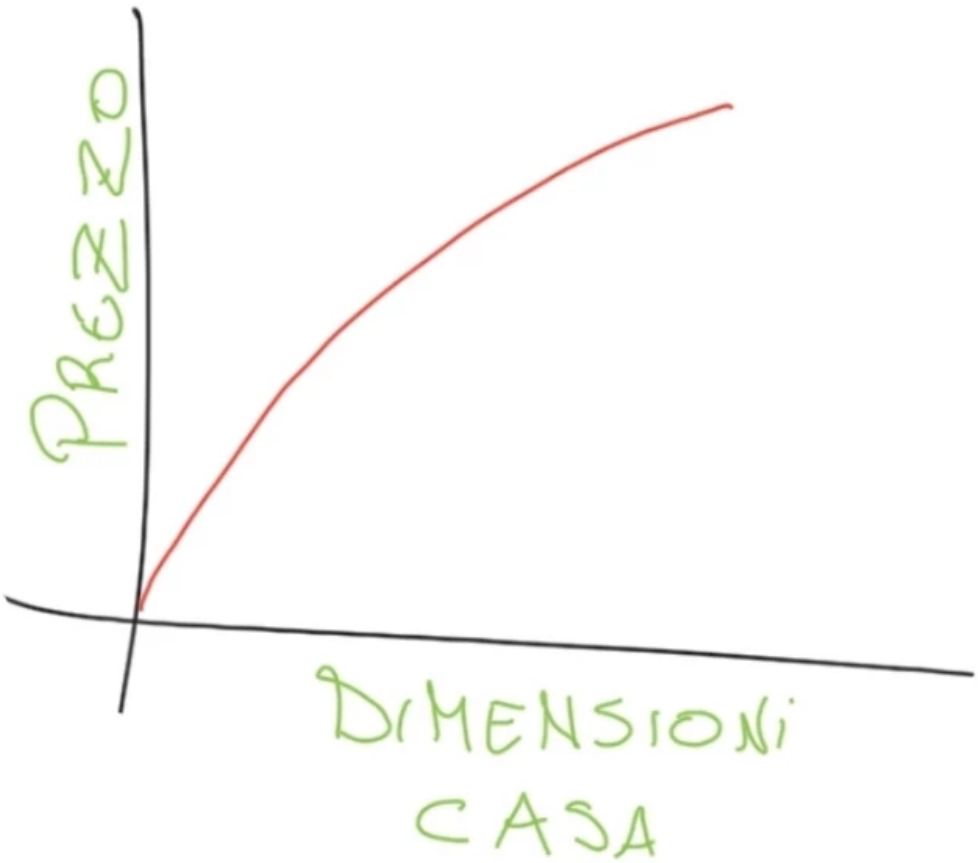
80 m³ | 100.000$

100 m³ | 120.000$

140 m³ | 250.000$

Dando ora questi dati in pasto al nostro algoritmo

questo genererà una tale correlazione tra la variabile indipendente "**dimensioni**" e la variabile dipendente "**prezzo**"



modello
La linea rossa che potete vedere indica la crescita dei prezzi delle case sulla base delle loro dimensioni.

Per arrivare a creare questo grafico il computer è partito rappresentando dei punti con coordinate **x=le dimensioni** della casa e **y=il prezzo** delle case, per poi cercare una funzione che descrive l'andamento nel grafico di tutti questi punti.

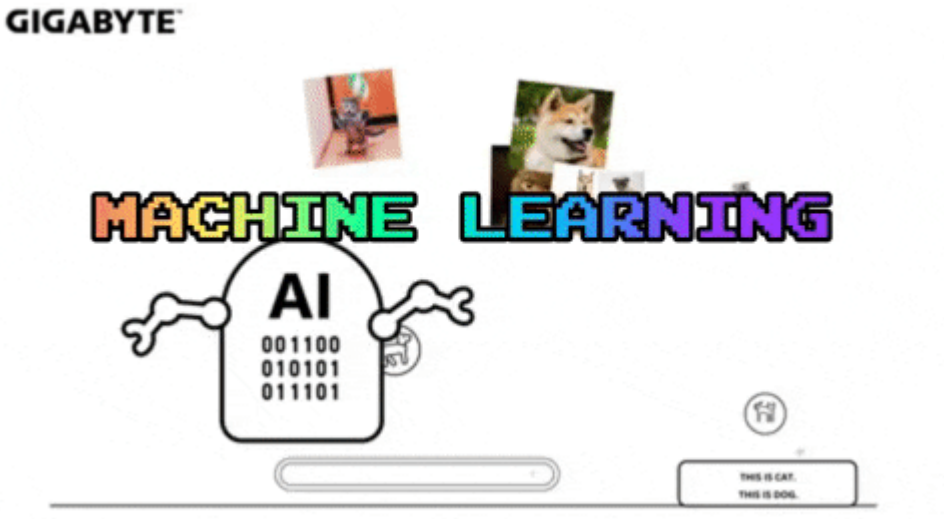
Dopo aver creato tale Modello e data in input una nuova dimensione della casa questo sarà in grado, basandosi sugli esempi precedenti, di fare una stima del prezzo.

Machine Learning Esempi di Utilizzo nella Vita di tutti i Giorni - Esempi Pratici Machine Learning

Il machine learning è un'innovazione moderna che ha aiutato l'uomo a migliorare non solo molti processi industriali e professionali, ma anche a far progredire la vita di tutti i giorni.

Ma cos'è l'apprendimento automatico?

È un sottoinsieme dell'intelligenza artificiale, che si concentra sull'utilizzo di tecniche statistiche per costruire sistemi informatici intelligenti al fine di apprendere dai database a sua disposizione. Attualmente, l'apprendimento automatico è stato utilizzato in più campi e settori. Ad esempio, diagnosi medica, elaborazione di immagini, previsione, classificazione, associazione di apprendimento, regressione, ecc.

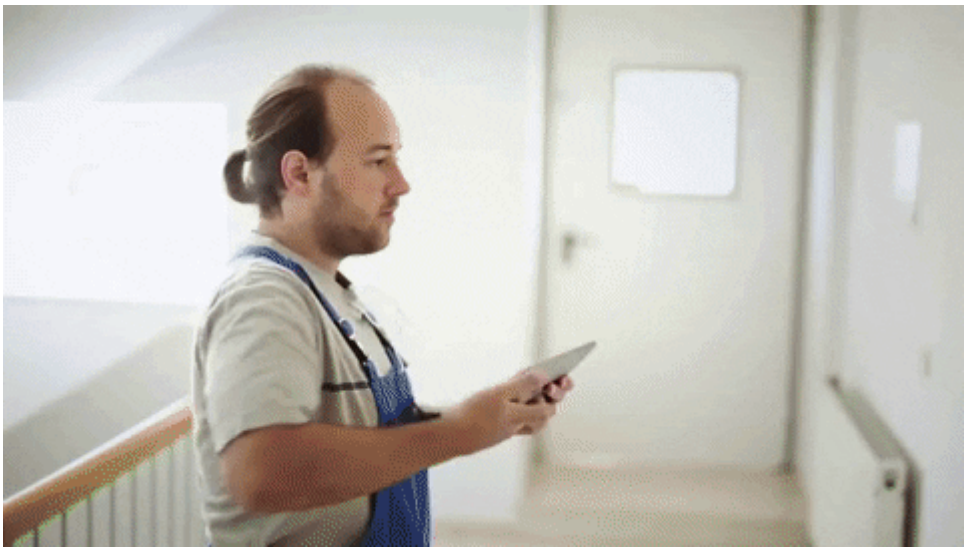


I sistemi intelligenti basati su algoritmi di apprendimento automatico hanno la capacità di apprendere dall'esperienza passata o dai dati storici. Le applicazioni di machine learning forniscono risultati sulla base dell'esperienza passata. In questo articolo, discuteremo 10 esempi di vita reale di come l'apprendimento automatico sta aiutando a creare una tecnologia migliore per alimentare le idee di oggi.

Esempio di Machine Learning 1) Riconoscimento delle immagini

Il riconoscimento delle immagini è uno degli usi più comuni del machine learning. Esistono molte situazioni in cui è possibile classificare l'oggetto come immagine digitale . Ad esempio, nel caso di un'immagine in bianco e nero, l'intensità di ciascun pixel viene utilizzata come una delle misurazioni. Nelle immagini a colori, ogni pixel fornisce 3 misurazioni di intensità in tre diversi colori: rosso, verde e blu (RGB).

L'apprendimento automatico può essere utilizzato anche per il rilevamento dei volti in un'immagine . C'è una categoria separata per ogni persona in un database di più persone. L'apprendimento automatico viene utilizzato anche per il riconoscimento dei caratteri per distinguere le lettere scritte a mano e quelle stampate. Possiamo segmentare un pezzo di scrittura in immagini più piccole, ciascuna contenente un singolo carattere.



Esempio di Machine Learning 2) Riconoscimento vocale

Il riconoscimento vocale è la traduzione delle parole pronunciate nel testo. È anche noto come riconoscimento vocale del computer o riconoscimento vocale automatico. Qui, un'applicazione software può riconoscere le parole pronunciate in una clip o file audio e quindi convertire l'audio in un file di testo. La misura in questa applicazione può essere un insieme di numeri che rappresentano il segnale vocale. Possiamo anche segmentare il segnale vocale per intensità in diverse bande tempo-frequenza.

Il riconoscimento vocale viene utilizzato in applicazioni come l'interfaccia utente vocale, le ricerche vocali e altro ancora. Le interfacce utente vocali includono la composizione vocale, l'instradamento delle chiamate e il controllo dell'appliance. Può essere utilizzato anche un semplice inserimento dati e la preparazione di documenti strutturati.

Esempio di Machine Learning 3) Diagnosi medica

L'apprendimento automatico può essere utilizzato nelle tecniche e negli strumenti che possono aiutare nella diagnosi delle malattie . Viene utilizzato per l'analisi dei parametri clinici e la loro combinazione per la previsione di esempio della prognosi della progressione della malattia per l'estrazione di conoscenze mediche per la ricerca dell'esito, per la pianificazione della terapia e il monitoraggio del paziente. Queste sono le implementazioni di successo dei metodi di apprendimento automatico. Può aiutare nell'integrazione di sistemi informatici nel settore sanitario.

Esempio di Machine Learning 4) Arbitraggio statistico

In finanza, l'arbitraggio si riferisce alle strategie di trading automatizzato che sono a breve termine e coinvolgono un gran numero di titoli. In queste strategie, l'utente si concentra sull'implementazione dell'algoritmo di negoziazione per un insieme di titoli sulla base di quantità come le correlazioni storiche e le variabili economiche generali. I metodi di apprendimento automatico vengono applicati per ottenere una strategia di arbitraggio dell'indice. Applichiamo la regressione lineare e la Support Vector Machine ai prezzi di un flusso di azioni.

Esempio di Machine Learning 5) Associazioni di apprendimento

Le associazioni di apprendimento sono il processo di sviluppo di intuizioni sulle varie associazioni tra i prodotti. Un buon esempio è come i prodotti non correlati possono essere associati tra loro. Una delle applicazioni del machine learning è studiare le associazioni tra i prodotti che le persone acquistano. Se una persona acquista un prodotto, gli verranno mostrati prodotti simili perché c'è una relazione tra i due prodotti. Quando nuovi prodotti vengono lanciati sul mercato, vengono associati a quelli vecchi per aumentarne le vendite.

Esempio di Machine Learning 6) Classificazione

Una classificazione è un processo di collocamento di ogni individuo sotto studio in molte classi. La classificazione aiuta ad analizzare le misurazioni di un oggetto per identificare la categoria a cui appartiene quell'oggetto. Per stabilire una relazione efficiente, gli analisti

utilizzano i dati. Ad esempio, prima che una banca decida di erogare prestiti, valuta i clienti sulla loro capacità di pagare i prestiti. Considerando fattori come i guadagni, i risparmi e la storia finanziaria del cliente, possiamo farlo. Queste informazioni sono tratte dai dati passati sul prestito.

Esempio di Machine Learning 7) Predizione

L'apprendimento automatico può essere utilizzato anche nei sistemi di previsione. Considerando l'esempio del prestito, per calcolare la probabilità di un guasto, il sistema dovrà classificare i dati disponibili in gruppi. È definito da un insieme di regole prescritte dagli analisti. Una volta effettuata la classificazione, possiamo calcolare la probabilità del guasto. Questi calcoli possono essere eseguiti in tutti i settori per vari scopi. Fare previsioni è una delle migliori applicazioni di machine learning.

Esempio di Machine Learning 8) Estrazione

L'estrazione di informazioni è una delle migliori applicazioni del machine learning . È il processo di estrazione di informazioni strutturate dai dati non strutturati. Ad esempio, le pagine Web, gli articoli, i blog, i rapporti aziendali e le e-mail. Il database relazionale mantiene l'output prodotto dall'estrazione delle informazioni. Il processo di estrazione prende un insieme di documenti come input ed emette i dati strutturati.

Esempio di Machine Learning 9) Regressione

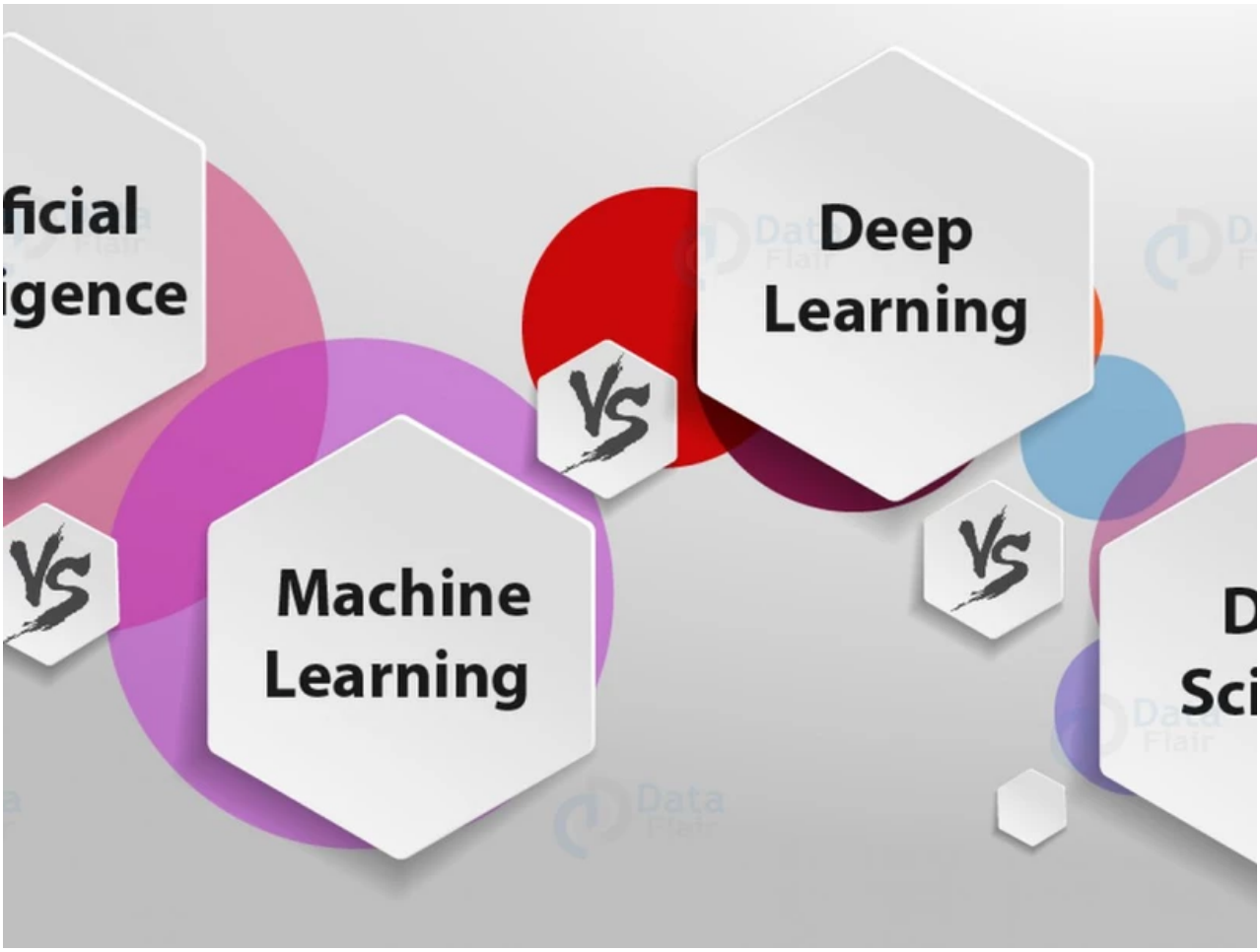
Possiamo anche implementare l'apprendimento automatico anche nella regressione. Nella regressione, possiamo utilizzare il principio dell'apprendimento automatico per ottimizzare i parametri. Può anche essere usato per diminuire l'errore di approssimazione e calcolare il risultato più vicino possibile. Possiamo anche utilizzare l'apprendimento automatico per l'ottimizzazione della funzione. Possiamo anche scegliere di modificare gli input per ottenere il risultato più vicino possibile.

Esempio di Machine Learning 10) Servizi finanziari

L'apprendimento automatico ha un grande potenziale nel settore finanziario e bancario . È la forza trainante della popolarità dei servizi finanziari. L'apprendimento automatico può aiutare le banche e gli istituti finanziari a prendere decisioni più intelligenti. L'apprendimento automatico può aiutare i servizi finanziari a individuare la chiusura di un conto prima che si verifichi. Può anche monitorare il modello di spesa dei clienti. L'apprendimento automatico può anche eseguire l'analisi di mercato. Le macchine intelligenti possono essere addestrate per monitorare i modelli di spesa. Gli algoritmi possono identificare facilmente le tende e possono reagire in tempo reale.

Le Differenze tra Machine Learning (ML) e Deep Learning (DL) e Intelligenza Artificiale

Questi termini sono spesso usati in modo intercambiabile, ma quali sono le differenze che li rendono ciascuno una tecnologia unica?



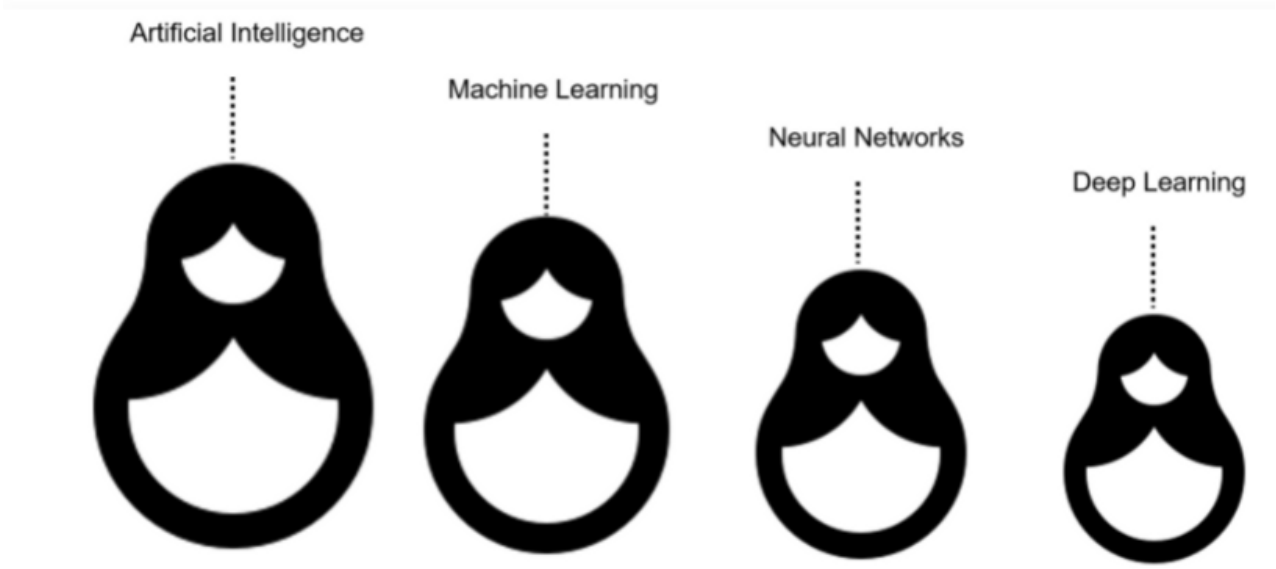
Le Differenze tra Machine Learning (ML) e Deep Learning (DL) e Intelligenza Artificiale

La tecnologia sta diventando sempre più incorporata nella nostra vita quotidiana di minuto in minuto e, per stare al passo con il ritmo delle aspettative dei consumatori, le aziende fanno sempre più affidamento su algoritmi di apprendimento per rendere le cose più facili. Puoi vedere la sua applicazione nei social media (tramite il riconoscimento degli oggetti nelle foto) o parlando direttamente ai dispositivi (come Alexa o Siri).

Queste tecnologie sono comunemente associati con l'intelligenza artificiale , machine learning , apprendimento profondo , e le reti neurali, e mentre lo fanno tutto il gioco un ruolo, questi termini tendono ad essere utilizzati in modo intercambiabile nella conversazione, che porta a una certa confusione intorno alle sfumature tra di loro. Si spera di poter utilizzare questo post del blog per chiarire alcune delle ambiguità qui.

Come si relazionano intelligenza artificiale, apprendimento automatico, reti neurali e apprendimento profondo?

Forse il modo più semplice per pensare all'intelligenza artificiale, all'apprendimento automatico, alle reti neurali e all'apprendimento profondo è pensarli come bambole russe che nidificano. Ciascuno è essenzialmente un componente del termine precedente.



Come si relazionano intelligenza artificiale, apprendimento automatico, reti neurali e apprendimento profondo?

Cioè, l'apprendimento automatico è un sottocampo dell'intelligenza artificiale. Il deep learning è un sottocampo del machine learning e le reti neurali costituiscono la spina dorsale degli algoritmi di deep learning. In effetti, è il numero di strati di nodi, o profondità, di reti neurali che distingue una singola rete neurale da un algoritmo di apprendimento profondo, che deve averne più di tre.

Cos'è una rete neurale?

Le reti neurali, e più specificamente le reti neurali artificiali (ANN), **imitano** il cervello umano attraverso una serie di algoritmi. A livello di base, una rete neurale è composta da quattro componenti principali: **input**, **pesi**, un **bias** o soglia e un **output**. Simile alla regressione lineare , la formula algebrica sarebbe simile a questa:

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

Formula algebrica

Da lì, appliciamolo a un esempio più tangibile, ad esempio se dovresti ordinare o meno una pizza per cena. Questo sarà il nostro risultato previsto, o y-capello. Supponiamo che ci siano tre fattori principali che influenzeranno la tua decisione:

1. Se risparmierai tempo ordinando (Sì: 1; No: 0)
2. Se perdi peso ordinando una pizza (Sì: 1; No: 0)
3. Se risparmierai denaro (Sì: 1; No: 0)

Quindi, supponiamo quanto segue, dandoci i seguenti input:

- $X1 = 1$, dal momento che non stai preparando la cena
- $X2 = 0$, poiché stiamo ottenendo TUTTI i condimenti
- $X3 = 1$, poiché stiamo ottenendo solo 2 fette

Per semplicità, i nostri input avranno un valore binario di 0 o 1. Questo tecnicamente lo definisce come un perceptron poiché le reti neurali sfruttano principalmente i neuroni sigmoide, che rappresentano valori dall'infinito negativo all'infinito positivo. Questa distinzione è importante poiché la maggior parte dei problemi del mondo reale non sono lineari, quindi abbiamo bisogno di valori che riducano l'influenza che ogni singolo input può avere sul risultato. Tuttavia, riassumere in questo modo ti aiuterà a capire la matematica sottostante in gioco qui.

Andando avanti, ora dobbiamo assegnare alcuni pesi per determinare l'importanza. Pesi maggiori rendono il contributo di un singolo input all'output più significativo rispetto ad altri input.

- $W1 = 5$, poiché dai valore al tempo
- $W2 = 3$, poiché apprezzi restare in forma
- $W3 = 2$, visto che hai soldi in banca

Infine, assumeremo anche un valore di soglia di 5, che si tradurrebbe in un valore di bias di –5.

Poiché abbiamo stabilito tutti i valori rilevanti per la nostra somma, ora possiamo inserirli in questa formula.

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

Formula

Utilizzando la seguente funzione di attivazione, possiamo ora calcolare l'output (ovvero la nostra decisione di ordinare la pizza):

$$\text{output} = f(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases}$$

Possibile funzione di Attivazione di una rete neurale
In sintesi:

1. Y-cappello(il nostro risultato previsto) = Decidi se ordinare o meno la pizza
2. Y-cappello = (1 * 5) + (0 * 3) + (1 * 2) - 5
3. Y-cappello = 5 + 0 + 2-5
4. Y-cappello = 2, che è maggiore di zero.

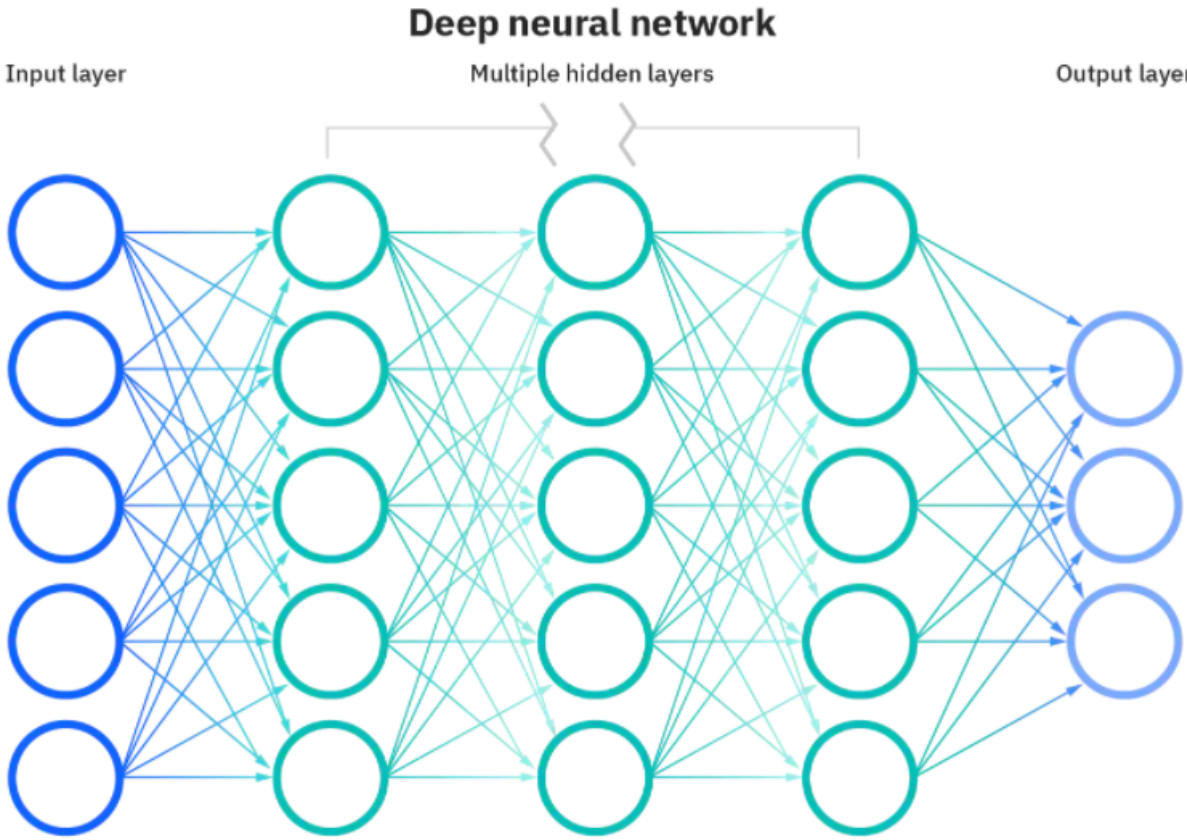
Poiché Y-cappello è 2, l' **uscita dalla funzione di attivazione** sarà **1,** il che significa che ci ***sarà da*** **ordinare la pizza** (Cioè, che non fa la pizza amore).

Se l'uscita di un singolo nodo è superiore al valore di soglia specificato, quel nodo viene attivato, inviando i dati al livello successivo della rete. In caso contrario, nessun dato viene passato al livello successivo della rete. Ora, immagina che il processo di cui sopra venga ripetuto più volte per una singola decisione poiché le reti neurali tendono ad avere più livelli "nascosti" come parte degli algoritmi di apprendimento profondo. Ogni livello nascosto ha una propria funzione di attivazione, che potenzialmente passa le informazioni dal livello precedente a quello successivo. Una volta generati tutti gli output dagli strati nascosti, vengono utilizzati come input per calcolare l'output finale della rete neurale. Ancora una volta, l'esempio sopra è solo l'esempio più semplice di una rete neurale; la maggior parte degli esempi del mondo reale sono non lineari e molto più complessi.

La principale differenza tra la regressione e una rete neurale è l'impatto del cambiamento su un singolo peso. Nella regressione, è possibile modificare un peso senza influire sugli altri input in una funzione. Tuttavia, questo non è il caso delle reti neurali. Poiché l'output di uno strato viene passato allo strato successivo della rete, una singola modifica può avere un effetto a cascata sugli altri neuroni della rete.

La differenza tra Reti Neurali e Deep Learning?

Sebbene fosse implicito nella spiegazione delle reti neurali, vale la pena sottolinearlo in modo più esplicito. Il "profondo" nell'apprendimento profondo si riferisce alla profondità degli strati in una rete neurale. Una rete neurale composta da più di tre livelli, che includerebbero gli input e l'output, può essere considerata un algoritmo di deep learning. Questo è generalmente rappresentato utilizzando il diagramma seguente:



In che modo il deep learning è diverso dalle reti neurali?
La maggior parte delle reti neurali profonde sono feed-forward, nel senso che fluiscono in una sola direzione dall'input all'output. Tuttavia, puoi anche addestrare il tuo modello tramite backpropagation; ovvero, spostarsi in direzione opposta dall'output all'input. La backpropagation ci consente di calcolare e attribuire l'errore associato a ciascun neurone, permettendoci di regolare e adattare l'algoritmo in modo appropriato.

Differenze tra Machine Learning e Deep Learning ?

Come spieghiamo nel nostro articolo sull'hub di apprendimento sul deep learning , il deep learning è semplicemente un sottoinsieme del machine learning. Il modo principale in cui differiscono è nel modo in cui ogni algoritmo apprende e nella quantità di dati utilizzati da ciascun tipo di algoritmo. Il deep learning automatizza gran parte del processo di estrazione delle funzionalità, eliminando parte dell'intervento umano manuale richiesto. Consente inoltre l'uso di grandi set di dati, guadagnandosi il titolo di " **apprendimento automatico scalabile** " in questa conferenza del MIT. Questa capacità sarà particolarmente interessante quando inizieremo a esplorare maggiormente l'uso di dati non strutturati, in particolare perché si stima che l'80-90% dei dati di un'organizzazione non sia strutturato .

L'apprendimento automatico classico o "non profondo" dipende maggiormente dall'intervento umano per l'apprendimento. Gli esperti umani determinano la gerarchia delle funzionalità per comprendere le differenze tra gli input di dati, che di solito richiedono dati più strutturati per l'apprendimento. Ad esempio, supponiamo che ti mostri una serie di immagini di diversi tipi di fast food, "pizza", "hamburger" o "taco". L'esperto umano di queste immagini determinerebbe le caratteristiche che distinguono ogni immagine come il tipo specifico di fast food. Ad esempio, il pane di ogni tipo di cibo potrebbe essere una caratteristica distintiva in ogni immagine. In alternativa, potresti semplicemente utilizzare etichette, come "pizza", "hamburger" o "taco", per semplificare il processo di apprendimento attraverso l'apprendimento supervisionato.

L'apprendimento automatico "profondo" può sfruttare set di dati etichettati, noti anche come apprendimento supervisionato, per informare il suo algoritmo, ma non richiede necessariamente un set di dati etichettato. Può importare dati non strutturati nella sua forma grezza (ad esempio testo, immagini) e può determinare automaticamente l'insieme di caratteristiche che distinguono "pizza", "hamburger" e "taco" l'una dall'altra.

Osservando i modelli nei dati, un modello di apprendimento profondo può raggruppare gli input in modo appropriato. Prendendo lo stesso esempio di prima, potremmo raggruppare le immagini di pizze, hamburger e tacos nelle rispettive categorie in base alle somiglianze o alle differenze identificate nelle immagini. Detto questo, un modello di apprendimento profondo richiederebbe più punti dati per migliorarne l'accuratezza, mentre un modello di apprendimento automatico si basa su meno dati data la struttura dei dati sottostante. Il deep learning viene utilizzato principalmente per casi d'uso più complessi, come assistenti virtuali o rilevamento di frodi.

Cos'è l'intelligenza artificiale AI o Artificial Intelligence AI ?

Infine, l'intelligenza artificiale è il termine più ampio utilizzato per classificare le macchine che imitano l'intelligenza umana. Viene utilizzato per prevedere, automatizzare e ottimizzare le attività che gli esseri umani hanno svolto storicamente, come il riconoscimento vocale e facciale, il processo decisionale e la traduzione.

Esistono tre categorie principali di AI:

- Artificial Narrow Intelligence (ANI)
- Intelligenza generale artificiale (AGI)
- Super Intelligenza Artificiale (ASI)

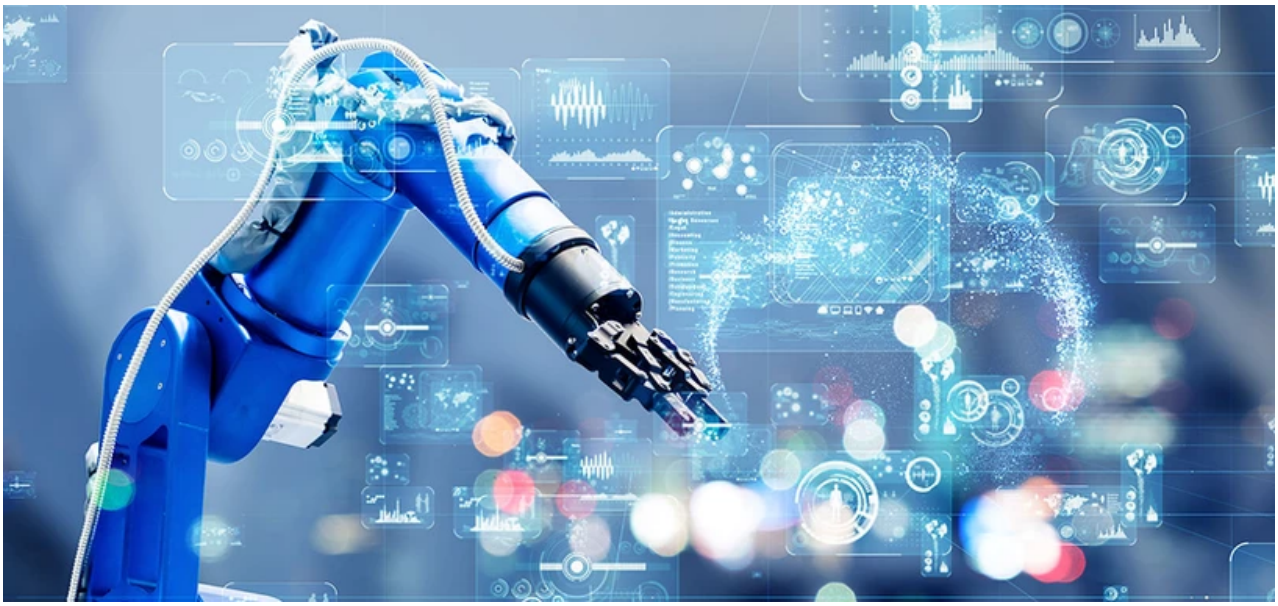
L'ANI è considerata AI "debole", mentre gli altri due tipi sono classificati come AI "forti". L'intelligenza artificiale debole è definita dalla sua capacità di completare un'attività molto specifica, come vincere una partita a scacchi o identificare un individuo specifico in una serie di foto. Man mano che ci spostiamo verso forme più forti di IA, come AGI e ASI, l'incorporazione di più comportamenti umani diventa più prominente, come la capacità di interpretare il tono e le emozioni. Chatbot e assistenti virtuali, come Siri, stanno grattando la superficie di questo, ma sono ancora esempi di ANI.

L'intelligenza artificiale forte è definita dalla sua capacità rispetto agli esseri umani.

L'Intelligenza Generale Artificiale (AGI) si esibirebbe alla pari di un altro essere umano mentre la **Super Intelligenza Artificiale (ASI)**, nota anche come iperintelligenza, supererebbe l'intelligenza e l'abilità di un essere umano. Nessuna delle due forme di IA forte esiste ancora, ma la ricerca in corso in questo campo continua. Poiché quest'area dell'IA è ancora in rapida evoluzione, il miglior esempio che posso offrire su come potrebbe apparire è il personaggio Dolores nello show della HBO ***Westworld*** .

Prerequisiti per le carriere di Intelligenza Artificiale - Machine Learning - Deep Learning

intelligenzaartificialeitalia.net/post/prerequisiti-per-le-carriere-di-intelligenza-artificiale-machine-learning-deep-learning



Quali sono i prerequisiti per le carriere di Intelligenza artificiale e Machine learning

Hai intenzione di iniziare una carriera nell'intelligenza artificiale e nell'apprendimento automatico?

Bene, allora questo articolo è per te. Costruire una carriera in AI e ML non è né facile né difficile. Ma richiede un approccio dedicato. A volte, quando provieni da un background IT, potresti aver voglia di scambiare anche le opzioni di carriera, a causa delle diverse opportunità. Per prima cosa comprendiamo i prerequisiti per accedere alle carriere di AI e ML.

Quali sono i prerequisiti per lavorare con l'Intelligenza Artificiale ?

L'intelligenza artificiale e l'apprendimento automatico hanno requisiti e qualifiche propri. Per costruire una carriera in AI e ML è necessario possedere set di competenze . Di seguito abbiamo elencato alcune abilità richieste.

1) Abilità statistiche

In qualità di aspirante AI , devi avere una conoscenza approfondita delle statistiche e delle probabilità per comprendere e analizzare algoritmi complessi. Poiché la maggior parte dei modelli di intelligenza artificiale dipende dalla ricerca di modelli in grandi quantità di informazioni, è fondamentale conoscere bene i metodi statistici utilizzati per ottenere informazioni dai dati.

2) Abilità matematiche e algebriche/geometriche

È necessaria una conoscenza completa delle abilità matematiche e di probabilità poiché l' intelligenza artificiale è un campo che presenta molti concetti matematici per creare l'intelligenza artificiale. La probabilità aiuta a determinare una varietà di risultati nell'intelligenza artificiale, con una comprensione più profonda dell'argomento che è parte integrante della creazione di modelli di intelligenza artificiale.

3) Abilità di programmazione

Se le abilità matematiche sono uno dei prerequisiti, le abilità di programmazione sono l'altra parte. Gli aspiranti all'intelligenza artificiale e all'apprendimento automatico richiedono linguaggi di programmazione Java, C++, Python e R.

Poiché il C++ aiuta gli ingegneri ad aumentare la velocità del loro processo di codifica, Python aiuterà a comprendere e creare algoritmi complessi. E quindi questi programmi sono importanti considerando tutti i ruoli nel settore AI e ML .

4) tecniche avanzate di elaborazione dati

Quando si tratta di machine learning, l'estrazione delle funzionalità è una caratteristica integrale. Per comprendere la prossima funzionalità e come distribuire i modelli, gli ingegneri di Intelligenza Artificiale e Machine Learning dovrebbero avere familiarità con varie tecniche avanzate di elaborazione dei dati.

5) Calcolo distribuito

Poiché tutti i lavori di intelligenza artificiale richiedono che i professionisti si occupino di set di dati grandi e complessi, è necessario distribuirli equamente su un intero cluster e quindi è obbligatorio disporre di competenze informatiche distribuite. Ciò include competenze in applicazioni, come MongoDB, oltre alla creazione e al funzionamento di ambienti cloud.

Come diventare un professionista di Intelligenza Artificiale ?

Inizia a prepararti

Ora che sai quali sono i prerequisiti per entrare a far parte del settore sei idoneo, ma ciò richiede anche set di abilità di prestazione lavorativa. Il prossimo passo della carriera è iniziare a lavorare sulle abilità che percepisci più ostili. La cosa migliore da fare è

acquistare libri sulla probabilità o sulle statistiche e rispolverare le abilità di programmazione(NON INIZIARE CON FRAMEWORK CHE LAVORANO AL POSTO TUO!).

O l'altro modo potrebbe anche partecipare a corsi di Intelligenza Artificiale e Machine Learning molto richiesti che possono aiutarti a potenziare le tue abilità a livelli avanzati. Quando si arriva a comprendere le funzioni e il lavoro del settore su base giornaliera, un esperto può aiutarti a navigare facilmente.

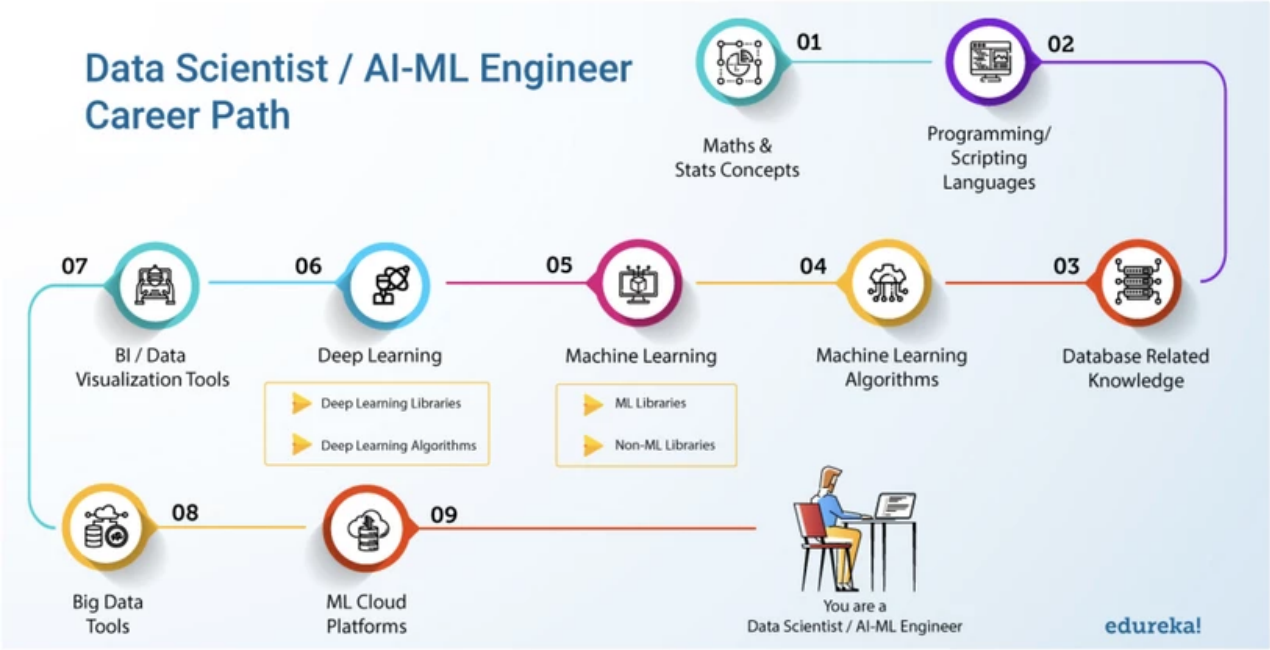
Lavora su progetti di Intelligenza Artificiale

Lavorare su progetti diversi ti offre un'ampia esperienza pratica sul campo e aiuta a mettere in evidenza il tuo curriculum. Quindi, lavora su tanti progetti e collabora con altri aspiranti ai progetti, questo può aiutarti a migliorare le tue capacità per soddisfare i requisiti del settore. La conoscenza della teoria è apprezzata quando viene applicata nell'ambiente reale. Quindi, è obbligatorio cimentarsi nell'applicazione delle proprie capacità per ottenere applicazioni pratiche.

Conclusioni

Ora che hai un'idea di come iniziare una carriera in AI e ML , è tempo per te di iniziare ad agire in tal senso. È anche importante analizzare i tuoi punti di forza e lavorare sui tuoi punti deboli e iniziare a lavorarci sopra. Questo può aiutarti a costruire una carriera di successo nell'intelligenza artificiale e nell'apprendimento automatico.

Nel caso potesse tornarti utile, ti lascio qui sotto un percorso che racchiude molto grossolanamente gli step formativi da seguire per avere successo in questo campo.


Data Scientist / AI-ML Engineer Career Path
01
Maths & Stats Concepts
02
Programming/ Scripting Languages
03
Database Related Knowledge
04
Machine Learning Algorithms
05
Machine Learning
ML Libraries
Non-ML Libraries
06
Deep Learning
Deep Learning Libraries
Deep Learning Algorithms
07
BI / Data Visualization Tools
08
Big Data Tools
09
ML Cloud Platforms
You are a Data Scientist / AI-ML Engineer
edureka!

Intelligenza Artificiale (IA) e Python Come si Relazionano?

Intelligenza Artificiale (IA) e Python Come si Relazionano?

Python è uno dei linguaggi di programmazione più popolari utilizzati dagli sviluppatori oggi. Guido Van Rossum lo ha creato nel 1991 e sin dal suo inizio è stato uno dei linguaggi più utilizzati insieme a C++, Java, ecc.

Nel nostro tentativo di identificare quale sia il miglior linguaggio di programmazione per l'intelligenza artificiale e la rete neurale, Python ha preso un grande vantaggio.

Caratteristiche e vantaggi di Python per l'intelligenza artificiale

Python è un linguaggio interpretato che in parole povere significa che non ha bisogno di essere compilato in istruzioni in linguaggio macchina prima dell'esecuzione e può essere utilizzato direttamente dallo sviluppatore per eseguire il programma. Ciò lo rende

abbastanza completo da consentire l'interpretazione del linguaggio da parte di un emulatore o di una macchina virtuale oltre al linguaggio macchina nativo, che è ciò che l'hardware comprende.

È un linguaggio di programmazione di alto livello e può essere utilizzato per scenari complicati. I linguaggi di alto livello trattano variabili, array, oggetti, complesse espressioni aritmetiche o booleane e altri concetti astratti di informatica per renderlo più completo, aumentando così in modo esponenziale la sua usabilità.

Python è anche un linguaggio di programmazione generico, il che significa che può essere utilizzato in più domini e tecnologie.

Python dispone anche di un sistema di tipi dinamici e di una gestione automatica della memoria che supporta un'ampia varietà di paradigmi di programmazione, inclusi quelli orientati agli oggetti, imperativi, funzionali e procedurali, per citarne alcuni.

Python è disponibile per tutti i sistemi operativi e ha anche un'offerta open source intitolata CPython che sta ottenendo una popolarità diffusa.

Diamo ora un'occhiata a come l'uso di Python per l'intelligenza Artificiale ci offre un vantaggio rispetto ad altri linguaggi di programmazione popolari.

Intelligenza Artificiale e Python Perché?

L'ovvia domanda che dobbiamo affrontare a questo punto è perché dovremmo scegliere Python per l'intelligenza artificiale rispetto ad altri .

Python offre maggior risultato con minor codice, infatti è stimato che serva solo 1/5 del codice per eseguire stesse operazioni rispetto ad altri linguaggi OOP. **Non c'è da stupirsi che sia uno dei più popolari oggi sul mercato.**

- Python ha librerie predefinite come Numpy per il calcolo scientifico, Scipy per il calcolo avanzato e Pybrain per l'apprendimento automatico (Python Machine Learning) che lo rendono uno dei migliori linguaggi per l'intelligenza artificiale.
- Gli sviluppatori Python di tutto il mondo forniscono supporto e assistenza completi tramite forum e tutorial, rendendo il lavoro del programmatore più semplice di qualsiasi altro linguaggio popolare.

- Python è indipendente dalla piattaforma ed è quindi una delle scelte più flessibili e popolari per l'utilizzo su diverse piattaforme e tecnologie con il minimo ritocco nella codifica di base.
- Python è il più flessibile di tutti gli altri con opzioni per scegliere tra l'approccio OOP e lo scripting. Puoi anche utilizzare l'IDE stesso per verificare la maggior parte dei codici ed è un vantaggio per gli sviluppatori alle prese con algoritmi diversi.

Python insieme all'intelligenza Artificiale

Python insieme a pacchetti come NumPy, scikit-learn, iPython Notebook e matplotlib costituiscono la base per avviare il tuo progetto di intelligenza artificiale.

NumPy viene utilizzato come contenitore per dati generici comprendenti un oggetto array N-dimensionale, strumenti per l'integrazione di codice C/C++, trasformata di Fourier, capacità di numeri casuali e altre funzioni.

Un'altra libreria utile è pandas, una libreria open source che fornisce agli utenti strutture dati di facile utilizzo e strumenti analitici per Python.

Matplotlib è un altro servizio che è una libreria di plottaggio 2D che crea figure di qualità di pubblicazione. È possibile utilizzare matplotlib fino a 6 toolkit di interfaccia utente grafica, server di applicazioni Web e script Python.

Il tuo prossimo passo sarà esplorare il clustering di k-means e anche raccogliere conoscenze su alberi decisionali, previsione numerica continua, regressione logistica, ecc.

Alcune delle librerie AI Python più comunemente utilizzate sono AIMA, pyDatalog, SimpleAI, EasyAi, ecc. Esistono anche librerie Python per l'apprendimento automatico come PyBrain, MDP, scikit, PyML.

Diamo un'occhiata un po' più in dettaglio alle varie librerie Python nell'IA e perché questo linguaggio di programmazione viene utilizzato per l'IA.

Librerie Python per l' Intelligenza Artificiale

- **AIMA** – Implementazione in Python di algoritmi da "Artificial Intelligence: A Modern Approach" di Russell e Norvig.
- **pyDatalog** – Motore di programmazione logica in Python
- **SimpleAI** – Implementazione in Python di molti degli algoritmi di intelligenza artificiale descritti nel libro “Artificial Intelligence, a Modern Approach”. Si concentra sulla fornitura di una libreria facile da usare, ben documentata e testata.
- **EasyAI** – Semplice motore Python per giochi a due giocatori con AI (Negamax, tabelle di trasposizione, risoluzione di giochi).

Python per l' Apprendimento Automatico (ML)

Diamo un'occhiata al motivo per cui Python viene utilizzato per l'apprendimento automatico e le varie librerie che offre allo scopo.

- **PyBrain** : un algoritmo flessibile, semplice ma efficace per le attività di **machine learning** . È anche una libreria modulare di Machine Learning per Python che fornisce una varietà di ambienti predefiniti per testare e confrontare algoritmi.
- **PyML** – Un framework bilaterale scritto in Python che si concentra su SVM e altri metodi del kernel. È supportato su Linux e Mac OS X.
- **Scikit-learn** – Scikit-learn è uno strumento efficiente per l'analisi dei dati durante l'utilizzo di Python. È open source e la libreria di machine learning generica più popolare.
- **MDP-Toolkit** – Un altro framework di elaborazione dati Python che può essere facilmente ampliato, ha anche una raccolta di algoritmi di apprendimento supervisionati e non supervisionati e altre unità di elaborazione dati che possono essere combinate in sequenze di elaborazione dati e architetture di rete feed-forward più complesse. L'implementazione di nuovi algoritmi è facile ed intuitiva. La base di algoritmi disponibili è in costante aumento e include metodi di elaborazione del segnale (analisi dei componenti principali, analisi dei componenti indipendenti e analisi delle caratteristiche lente), metodi di apprendimento multipli ([Hessian] Locally Linear Embedding), diversi classificatori, metodi probabilistici (analisi fattoriale, RBM), metodi di pre-trattamento dei dati e molti altri.

Librerie Python per il linguaggio naturale e l'elaborazione del testo

NLTK – Moduli Python open source, dati linguistici e documentazione per ricerca e sviluppo nell'elaborazione del linguaggio naturale e analisi del testo con distribuzioni per Windows, Mac OSX e Linux.

Python contro altri linguaggi popolari

Vediamo ora dove si trova Python con un altro linguaggio informatico per l'intelligenza artificiale come C++ e Java.

Python contro C++ per l'intelligenza Artificiale

- Python è un linguaggio più popolare rispetto a C++ per l'intelligenza artificiale e guida con un voto del 57% tra gli sviluppatori. Questo perché Python è facile da imparare e implementare. Con le sue numerose librerie, possono essere utilizzati anche per l'analisi dei dati.
- Per quanto riguarda le prestazioni, il C++ supera Python. Questo perché C++ ha il vantaggio di essere un linguaggio tipizzato staticamente e quindi non ci sono errori di digitazione durante il runtime. C++ crea anche codice runtime più compatto e veloce.
- Python è un linguaggio dinamico (al contrario di statico) e riduce la complessità quando si tratta di collaborare, il che significa che è possibile implementare funzionalità con meno codice. A differenza di C++, dove tutti i compilatori significativi tendono a eseguire ottimizzazioni specifiche e possono essere specifici della piattaforma, il codice Python può essere eseguito praticamente su qualsiasi piattaforma senza perdere tempo su configurazioni specifiche.
- Con l'aumento delle capacità di offerta di elaborazione accelerata da GPU per il parallelismo che ha portato alla creazione di librerie come CUDA Python e cuDNN, Python ha il vantaggio su C++. Ciò significa che una parte sempre maggiore del calcolo effettivo per i carichi di lavoro di machine learning viene scaricato sulle GPU e il risultato è che qualsiasi vantaggio prestazionale che il C++ può avere sta diventando sempre più irrilevante.
- Python vince su C++ per quanto riguarda la semplicità del codice, specialmente tra i nuovi sviluppatori. Il C++ essendo un linguaggio di livello inferiore richiede più esperienza e abilità da padroneggiare.
- La semplice sintassi di Python consente anche un processo ETL (Estrai, Trasforma, Carica) più naturale e intuitivo e significa che è più veloce per lo sviluppo rispetto a C++, consentendo agli sviluppatori di testare algoritmi di apprendimento automatico senza doverli implementare rapidamente.

Tra C++ e Python, quest'ultimo ha più margine ed è più adatto per l'IA. Con la sua semplice sintassi e leggibilità che promuovono il test rapido di complessi algoritmi di apprendimento automatico e una fiorente comunità supportata da strumenti collaborativi come Jupyter Notebooks e Google Colab, Python vince la corona.

Usare Java per programmare IA

Per capire come programmare l'intelligenza artificiale in Java, è essenziale sapere dove si trova rispetto a Python.

- Java è un linguaggio compilato mentre Python è un linguaggio interpretato.
- Le due lingue sono anche scritte in modo diverso. Una struttura in Java è racchiusa tra parentesi graffe. Python usa il rientro per eseguire le stesse attività.
- Java è anche più lento dal punto di vista delle prestazioni e per lo sviluppo di applicazioni di fascia alta in AI, Python è più preferito dagli sviluppatori.

Java Artificial Intelligence Library è la risposta di Java a Python, ma è ancora meno accessibile agli sviluppatori per evidenti ragioni. L'approccio moderno all'IA di Java Norvig Russell ha spianato la strada a molti per sedersi e notare perché potrebbe essere il linguaggio migliore per una rete neurale.

Piccolo Caso Studio

È stato condotto un esperimento per utilizzare l'intelligenza artificiale con un Internet of Things per creare un'applicazione IoT per l'analisi comportamentale dei dipendenti. Il software fornisce feedback utili ai dipendenti attraverso le emozioni dei dipendenti e l'analisi del comportamento, migliorando così i cambiamenti positivi nella gestione e nelle abitudini di lavoro.

Utilizzando le librerie di apprendimento automatico **Python**, **opencv** e **haarcascading** per la formazione delle applicazioni, è stato creato un **POC** di esempio per rilevare emozioni di base come felicità, rabbia, tristezza, disgusto, sospetto, disprezzo, sarcasmo e sorpresa attraverso telecamere wireless collegate in vari punti della baia.

I dati raccolti sono stati inviati a un database di cloud computing centralizzato in cui è possibile recuperare il quoziente emotivo giornaliero all'interno della baia o anche l'intero ufficio con un clic di un pulsante tramite un dispositivo Android o desktop.

Gli sviluppatori stanno facendo progressi graduali nell'analisi di ulteriori punti complessi sulle emozioni facciali e estrae maggiori dettagli con l'aiuto di algoritmi di deep learning e apprendimento automatico che possono aiutare ad analizzare le prestazioni dei singoli dipendenti e supportare il corretto feedback dei dipendenti/team.

Conclusione

Python svolge un ruolo fondamentale nel linguaggio di codifica AI fornendogli buoni framework come scikit-learn: machine learning in Python, che soddisfa quasi tutte le esigenze in questo campo e D3.js – Data-Driven Documents in JS, che è uno dei strumenti di visualizzazione più potenti e facili da usare.

Oltre ai framework, la sua rapida prototipazione lo rende un linguaggio importante da non ignorare. L'intelligenza artificiale ha bisogno di molte ricerche, e quindi è necessario non richiedere un codice standard di 500 KB in Java per testare una nuova ipotesi, che non porterà mai a termine il progetto. In Python, quasi ogni idea può essere rapidamente convalidata attraverso 20-30 righe di codice (lo stesso per JS con librerie). Pertanto, è un linguaggio piuttosto utile per il bene dell'intelligenza artificiale.

Quindi è abbastanza evidente che Python è il miglior linguaggio di programmazione per l'intelligenza Artificiale. Oltre ad essere il miglior linguaggio per l'intelligenza artificiale, Python è utile per molti altri obiettivi.

Algoritmi di Machine Learning (ML) usati nella Data Science (con Esempi Pratici di ML in Python)

Probabilmente stiamo vivendo nel periodo più decisivo della storia umana. Il periodo in cui l'informatica è passata dai grandi mainframe ai PC al cloud.

Ma ciò che lo rende determinante non è ciò che è successo, ma ciò che accadrà negli anni a venire.

Ciò che rende questo periodo emozionante e avvincente per uno come me è la democratizzazione dei vari strumenti e tecniche, che ha seguito l'impulso nell'informatica.

Benvenuto nel mondo della scienza dei dati !

Algoritmi di Machine Learning (ML) usati nella DataScience (con Esempi Pratici di ML in Python)

Oggi, come scienziato dei dati, posso costruire macchine per l'elaborazione dei dati con algoritmi complessi per pochi dollari all'ora. Ma arrivare qui non è stato facile! Ho avuto i miei giorni e le mie notti buie.

In linea di massima, ci sono 3 tipi di algoritmi di apprendimento automatico

1. Apprendimento supervisionato

Come funziona: questo algoritmo consiste in una variabile obiettivo/risultato (o variabile dipendente) che deve essere prevista da un determinato insieme di predittori (variabili indipendenti). Usando questo insieme di variabili, generiamo una funzione che mappa gli input agli output desiderati. Il processo di addestramento continua finché il modello non raggiunge il livello di accuratezza desiderato sui dati di addestramento. Esempi di apprendimento supervisionato: regressione, albero decisionale , foresta casuale , KNN, regressione logistica ecc.

2. Apprendimento senza supervisione

Come funziona: in questo algoritmo non abbiamo alcun obiettivo o variabile di risultato da prevedere/stimare. Viene utilizzato per raggruppare la popolazione in diversi gruppi, che è ampiamente utilizzato per segmentare i clienti in diversi gruppi per interventi specifici. Esempi di apprendimento non supervisionato: algoritmo Apriori, K-means.

3. Apprendimento per rinforzo

Come funziona: utilizzando questo algoritmo, la macchina viene addestrata a prendere decisioni specifiche. Funziona così: la macchina è esposta a un ambiente in cui si allena continuamente per tentativi ed errori. Questa macchina impara dall'esperienza passata e

cerca di acquisire la migliore conoscenza possibile per prendere decisioni aziendali accurate. Esempio di apprendimento per rinforzo: processo decisionale di Markov

Elenco Principali Algoritmi di apprendimento automatico o Machine Learning

Ecco l'elenco degli algoritmi di apprendimento automatico comunemente usati. Questi algoritmi possono essere applicati a quasi tutti i problemi di dati:

1. Regressione lineare
2. Regressione logistica
3. Albero decisionale
4. SVM
5. Naive Bayes
6. kNN
7. K-Means
8. Foresta casuale
9. Algoritmi di riduzione della dimensionalità
10. Algoritmi di aumento del gradiente (XGboost)

Partiamo e vediamoli uno ad uno con la relativa implementazione in Python.

1. Spiegazione e Implementazione Algoritmo Regressione lineare

Viene utilizzato per stimare i valori reali (costo delle case, numero di chiamate, vendite totali, ecc.) in base a variabili continue. Qui, stabiliamo una relazione tra variabili indipendenti e dipendenti adattando una linea migliore. Questa linea di miglior adattamento è nota come linea di regressione ed è rappresentata da un'equazione lineare **Y= a *X + b.**

Diciamo che chiedi a un bambino di quinta elementare di sistemare le persone nella sua classe aumentando l'ordine di peso, senza chiedere loro il peso! Cosa pensi che farà il bambino? Probabilmente guarderebbe (analizzerebbe visivamente) l'altezza e la corporatura delle persone e le disporrebbe utilizzando una combinazione di questi parametri visibili. Questa è regressione lineare nella vita reale! Il bambino ha effettivamente capito che altezza e corporatura sarebbero correlate al peso da una relazione, che assomiglia all'equazione sopra.

In questa equazione:

- Y – Variabile dipendente
- a - Pendenza retta
- X – Variabile indipendente
- b – Bias

Questi coefficienti a e b sono derivati sulla base della riduzione al minimo della differenza al quadrato della distanza tra i punti dati e la linea di regressione.

La regressione lineare è principalmente di due tipi: regressione lineare semplice e regressione lineare multipla. La regressione lineare semplice è caratterizzata da una variabile indipendente. Inoltre, la regressione lineare multipla (come suggerisce il nome) è caratterizzata da più (più di 1) variabili indipendenti. Mentre trovi la linea più adatta, puoi adattare una regressione polinomiale o curvilinea. E questi sono noti come regressione polinomiale o curvilinea.

Ecco una finestra per metterti alla prova e costruire il tuo modello di regressione lineare in Python:

2. Spiegazione e Implementazione Algoritmo Regressione logistica

Non farti confondere dal suo nome! È una classificazione, non un algoritmo di regressione. Viene utilizzato per stimare valori discreti (valori binari come 0/1, sì/no, vero/falso) in base a un determinato insieme di variabili indipendenti. In parole semplici, prevede la probabilità di occorrenza di un evento adattando i dati a una funzione logistica . Quindi, è anche nota come **regressione** logistica . Poiché prevede la probabilità, i suoi valori di output sono compresi tra 0 e 1 (come previsto).

Ancora una volta, cerchiamo di capirlo attraverso un semplice esempio.

Supponiamo che il tuo amico ti dia un puzzle da risolvere.

Ci sono solo 2 scenari di risultato:

1. **o lo risolvi**
2. **o non lo fai.**

Ora immagina che ti venga data un'ampia gamma di enigmi / quiz nel tentativo di capire in quali materie sei bravo. Il risultato di questo studio sarebbe qualcosa del genere: se ti viene assegnato un problema di terza media basato sulla trignometria, hai il 70% di probabilità di risolverlo. D'altra parte, se si tratta di una domanda di storia di quinta elementare, la probabilità di ottenere una risposta è solo del 30%. Questo è ciò che ti offre la regressione logistica.

Venendo alla matematica, le probabilità logaritmiche del risultato sono modellate come una combinazione lineare delle variabili predittive.

```
probabilità= p/ (1-p) = probabilità che si verifichi l'evento / (1-probabilità che
non si verifichi l'evento)
ln(odds) = ln(p/(1-p))
logit(p) = ln(p/(1-p)) = b0+b1X1+b2X2+b3X3....+bkXk
```

Sopra, p è la probabilità di presenza della caratteristica di interesse. Sceglie parametri che massimizzano la probabilità di osservare i valori del campione piuttosto che minimizzare la somma degli errori al quadrato (come nella regressione ordinaria).

Costruisci il tuo modello di regressione logistica in Python qui e controlla l'accuratezza:

3. Spiegazione e Implementazione Algoritmo Albero decisionale

Questo è uno dei miei algoritmi preferiti e lo uso abbastanza frequentemente. È un tipo di algoritmo di apprendimento supervisionato utilizzato principalmente per problemi di classificazione. Sorprendentemente, funziona sia per variabili dipendenti categoriali che continue. In questo algoritmo, dividiamo la popolazione in due o più insiemi omogenei. Questo viene fatto in base agli attributi/variabili indipendenti più significativi per creare gruppi il più distinti possibile.

Sporchiamoci le mani e codifichiamo il nostro albero decisionale in Python!

4. Spiegazione e Implementazione Algoritmo SVM (macchina vettoriale di supporto)

È un metodo di classificazione. In questo algoritmo, tracciamo ogni elemento di dati come un punto nello spazio n-dimensionale (dove n è il numero di caratteristiche che hai) con il valore di ciascuna caratteristica che è il valore di una particolare coordinata.

Ad esempio, se avessimo solo due caratteristiche come l'altezza e la lunghezza dei capelli di un individuo, per prima cosa tracciaremmo queste due variabili in uno spazio bidimensionale in cui ogni punto ha due coordinate (queste coordinate sono note come **vettori di supporto**)

Ora troveremo una *linea* che divide i dati tra i due gruppi di dati diversamente classificati.

Questa sarà la linea tale che le distanze dal punto più vicino in ciascuno dei due gruppi saranno più lontane.

Questo è il codice di una possibile implementazione.

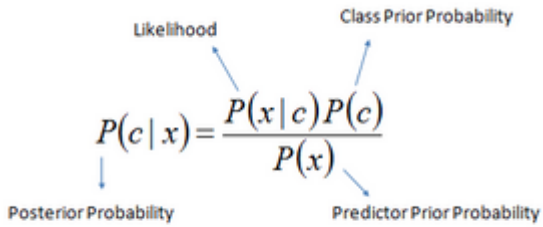
5. Spiegazione e Implementazione Algortimo Naive Bayes

È una tecnica di classificazione basata sul teorema di Bayes con un'assunzione di indipendenza tra predittori. In parole povere, un classificatore Naive Bayes presuppone che la presenza di una particolare caratteristica in una classe non sia correlata alla presenza di qualsiasi altra caratteristica. Ad esempio, un frutto può essere considerato una mela se è rosso, rotondo e di circa 3 pollici di diametro. Anche se queste

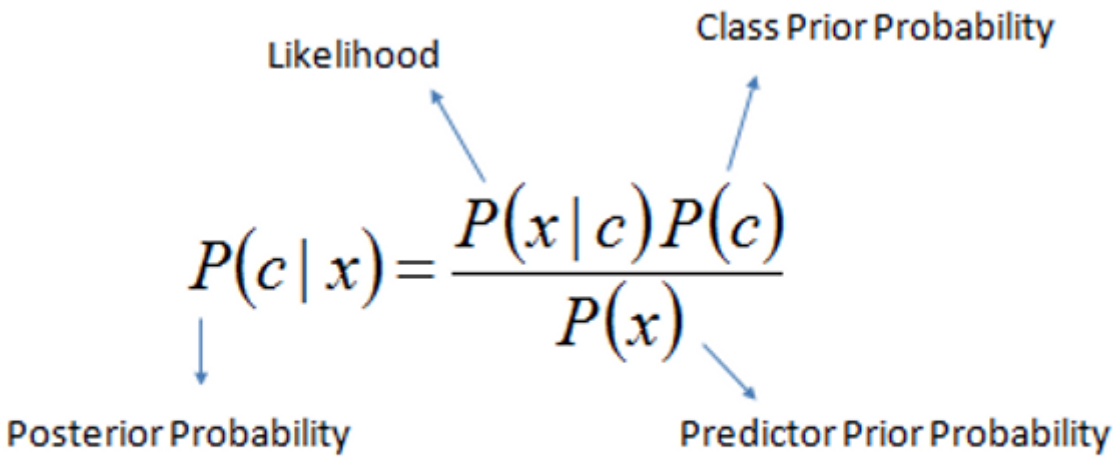
caratteristiche dipendono l'una dall'altra o dall'esistenza delle altre caratteristiche, un ingenuo classificatore di Bayes considererebbe tutte queste proprietà come un contributo indipendente alla probabilità che questo frutto sia una mela.

Il modello bayesiano ingenuo è facile da costruire e particolarmente utile per set di dati molto grandi. Insieme alla semplicità, Naive Bayes è noto per superare anche i metodi di classificazione altamente sofisticati.

Il teorema di Bayes fornisce un modo per calcolare la probabilità a posteriori P(c|x) da P(c), P(x) e P(x|c). Guarda l'equazione qui sotto:



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Equazione di Bayes

Qui,

- $P(c|x)$ è la probabilità a posteriori della *classe* (*obiettivo*) dato il *predittore* (*attributo*).
- $P(c)$ è la probabilità a priori della *classe* .
- $P(x|c)$ è la verosimiglianza che è la probabilità del *predittore* data la *classe* .
- $P(x)$ è la probabilità a priori del *predittore* .

Naive Bayes utilizza un metodo simile per prevedere la probabilità di classi diverse in base a vari attributi. Questo algoritmo è utilizzato principalmente nella classificazione del testo e con problemi con più classi.

Programma un modello di classificazione Naive Bayes in Python:

6. Spiegazione e Implementazione Algoritmo kNN

Può essere utilizzato sia per problemi di classificazione che di regressione. Tuttavia, è più ampiamente utilizzato nei problemi di classificazione nell'industria.

KNN è un semplice algoritmo che memorizza tutti i casi disponibili e classifica i nuovi casi con un voto di maggioranza dei suoi k vicini

Queste funzioni di distanza possono essere la distanza Euclidea, Manhattan, Minkowski e Hamming. Le prime tre funzioni sono utilizzate per la funzione continua e la quarta (Hamming) per le variabili categoriali.

Il caso viene semplicemente assegnato alla classe del suo vicino più prossimo. A volte, la scelta di K risulta essere una sfida durante l'esecuzione della modellazione kNN.

KNN può essere facilmente usato nelle nostre vite. Se vuoi conoscere una persona di cui non hai informazioni, ti potrebbe piacere conoscere i suoi amici intimi e le cerchie in cui si muove e avere accesso alle sue informazioni!

Aspetti da considerare prima di selezionare kNN:

- KNN è computazionalmente costoso
- Le variabili dovrebbero essere normalizzate, altrimenti le variabili di intervallo più alto possono distorcerlo
- Lavora di più sulla fase di pre-elaborazione prima di utilizzare kNN

Programma un modello di clusterizzazione in Python:

7. Spiegazione e Implementazione Algoritmo K-Means

È un tipo di algoritmo non supervisionato che risolve il problema del clustering. La sua procedura segue un modo semplice e facile per classificare un dato set di dati attraverso un certo numero di cluster (assumere k cluster). I punti dati all'interno di un cluster sono omogenei ed eterogenei rispetto ai gruppi di pari.

Ricordi di aver capito le forme dalle macchie d'inchiostro? k significa che è in qualche modo simile a questa attività. Guardi la forma e diffondi per decifrare quanti diversi cluster/popolazioni sono presenti!

Come K-mezzi forma il cluster:

1. K-means seleziona k numero di punti per ogni cluster noto come centroidi.
2. Ciascun punto dati forma un cluster con il centroidi più vicini, ovvero k cluster.
3. Trova il centroide di ogni cluster in base ai membri del cluster esistenti. Qui abbiamo nuovi centroidi.

4. Poiché abbiamo nuovi centroidi, ripeti i passaggi 2 e 3. Trova la distanza più vicina per ogni punto dati dai nuovi centroidi e associali ai nuovi k-cluster. Ripetere questo processo finché non si verifica la convergenza, ovvero i centroidi non cambiano.

Come determinare il valore di K:

In K-means, abbiamo cluster e ogni cluster ha il suo centroide. La somma dei quadrati della differenza tra il centroide ei punti dati all'interno di un cluster costituisce il valore della somma dei quadrati per quel cluster. Inoltre, quando vengono aggiunti i valori della somma dei quadrati per tutti i cluster, diventa totale all'interno del valore della somma dei quadrati per la soluzione del cluster.

Sappiamo che all'aumentare del numero di cluster, questo valore continua a diminuire, ma se tracci il risultato potresti vedere che la somma della distanza al quadrato diminuisce bruscamente fino a un certo valore di k, e poi molto più lentamente dopo.

Implementazione in Python dell'algoritmo K-Means

8. Spiegazione e Implementazione Algoritmi Foresta casuale

Random Forest è un termine caratteristico per un insieme di alberi decisionali.

Abbiamo una raccolta di alberi decisionali (conosciuti come "Foresta"). Per classificare un nuovo oggetto in base agli attributi, ogni albero fornisce una classificazione e diciamo che l'albero "vota" per quella classe. La foresta sceglie la classifica con il maggior numero di voti (su tutti gli alberi della foresta).

Ogni albero viene piantato e cresciuto come segue:

1. Se il numero di casi nel training set è N, il campione di N casi viene preso a caso ma *con sostituzione* . Questo esempio sarà il training set per far crescere l'albero.

2. Se ci sono M variabili di input, viene specificato un numero m<<M tale che ad ogni nodo vengono selezionate a caso m variabili dalle M e la migliore suddivisione su queste m viene utilizzata per dividere il nodo. Il valore di m è mantenuto costante durante la crescita della foresta.

3. Ogni albero è cresciuto nella misura più ampia possibile. Non c'è potatura.

Implementiamo l'algoritmo foresta casuale con Python :

9. Spiegazione e Implementazione Algoritmi di riduzione della dimensionalità

Negli ultimi 4-5 anni, c'è stato un aumento esponenziale nell'acquisizione dei dati in tutte le fasi possibili. Le aziende/ le agenzie governative/ le organizzazioni di ricerca non solo stanno arrivando con nuove fonti, ma stanno anche catturando i dati in grande dettaglio.

Ad esempio: le aziende di e-commerce stanno acquisendo più dettagli sui clienti come i loro dati demografici, la cronologia di scansione del web, ciò che gli piace o non gli piace, la cronologia degli acquisti, il feedback e molti altri per dare loro un'attenzione personalizzata più del tuo negoziante di alimentari più vicino.

Come data scientist, i dati che ci vengono offerti consistono anche di molte funzionalità, questo suona bene per costruire un buon modello robusto, ma c'è una sfida. Come hai identificato una o più variabili altamente significative su 1000 o 2000? In tali casi, l'algoritmo di riduzione della dimensionalità ci aiuta insieme a vari altri algoritmi come Decision Tree, Random Forest, PCA, Analisi fattoriale, Identificazione basata su matrice di correlazione, rapporto di valori mancanti e altri.

Implementiamo un Algoritmo di riduzione della dimensionalità sui dei dati e vediamo le differenze

10. Spiegazione e Implementazione Algoritmi di aumento del gradiente (XGboost)

XGBoost ha un potere predittivo immensamente elevato che lo rende la scelta migliore per la precisione negli eventi in quanto possiede sia il modello lineare che l'algoritmo di apprendimento ad albero, rendendo l'algoritmo quasi 10 volte più veloce rispetto alle tecniche di booster gradiente esistenti.

Il supporto include varie funzioni oggettive, tra cui regressione, classificazione e ranking.

Una delle cose più interessanti di XGBoost è che è anche chiamata una tecnica di potenziamento regolarizzata. Questo aiuta a ridurre la modellazione overfit e ha un enorme supporto per una vasta gamma di linguaggi come Scala, Java, R, Python, Julia e C++.

Supporta la formazione distribuita e diffusa su molte macchine che comprendono cluster GCE, AWS, Azure e Yarn. XGBoost può anche essere integrato con Spark, Flink e altri sistemi di flusso di dati cloud con una convalida incrociata integrata ad ogni iterazione del processo di potenziamento.

Implementiamo con Python l'algoritmo XGBoost

10 Migliori Librerie Python Che i DataScientist (Scienziati dei dati) dovrebbero conoscere nel 2021

Prerequisiti :

- Se non sai perchè utilizzeremo python, clicca qui
- Se non hai ancora installato Python, clicca qui
- Se non sai come scaricare e gestire le librerie, clicca qui
- Se non sai chi è un DataScientist, clicca qui
- Se non sai cosa è l'Apprendimento Automatico, clicca qui

Esistono così tante librerie Python che offrono basi potenti ed efficienti per supportare il tuo lavoro di data science e lo sviluppo di modelli di machine learning. Sebbene l'elenco possa sembrare travolgente, ci sono alcune librerie su cui dovresti concentrare il tuo tempo, poiché sono alcune delle più comunemente utilizzate oggi.

Conoscere e studiare queste librerie ti darà molti benefici, tra i quali :

1. **Creare applicazioni di Analisi dati**
2. **Creare modelli per fare predizioni**
3. **Creare grafici 2D e 3D con i tuoi dati**
4. **Poter mettere mano su programmi già scritti con queste librerie**
5. **Poter creare API o una libreria tua usando le librerie che vedremo tra poco**
6. **Creare applicazioni web con Python**

L'analisi dei Dati e l'apprendimento automatico sono due "**ARMI**" che hanno letteralmente stravolto la nostra concezione di Lavoro e vita privata. Questa loro potenza abbinata ad uno strumento di programmazione semplice come Python ci permette di creare progetti che possono farsi invidiare da Google e Facebook. Però nonostante la semplicità di Python, potremmo scontrarci con un altro problema quando vogliamo creare il nostro progetto:

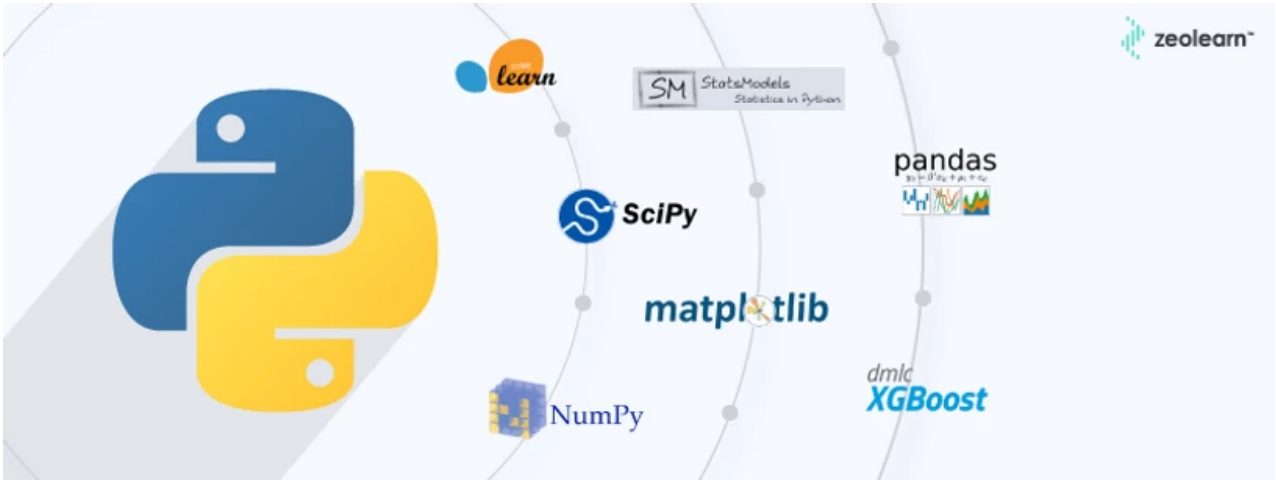
> Ci sono migliaia di strumenti, risorse e librerie là fuori e non è sempre ovvio su quali strumenti o librerie dovresti concentrarti o cosa dovresti imparare.

La risposta breve è che dovresti imparare **ciò che ti piace** perché la scienza dei dati offre **una vasta gamma di competenze e strumenti**. Detto questo, volevo condividere con voi quelle che credo siano le prime **10 librerie Python più comunemente utilizzate nella scienza dei dati**.

Ecco le 10 migliori librerie Python per la scienza dei dati.

1. **Pandas**
2. **Numpy**
3. **Scikit-learn**
4. **Gradio**
5. **Tensorflow**
6. **Keras**
7. **SciPy**
8. **StatsModels**
9. **Plotly**
10. **Seaborn**

Buona Lettura



10 Migliori Librerie Python Che i DataScientist (Scienziati dei dati) dovrebbero conoscere nel 2021

1. Pandas

Hai sentito il detto. Dal 70 all'80% del lavoro di un data scientist è comprendere e ripulire i dati, ovvero esplorazione dei dati e data munging.

Pandas viene utilizzato principalmente per l'analisi dei dati ed è una delle librerie Python più comunemente utilizzate. Ti fornisce alcuni dei set di strumenti più utili per esplorare, pulire e analizzare i tuoi dati. Con Pandas puoi caricare, preparare, manipolare e analizzare tutti i tipi di dati strutturati. Le librerie di machine learning ruotano anche attorno a Pandas DataFrames come input.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install pandas #se hai installato python 2
pip3 install pandas #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import pandas
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

2. NumPy

NumPy viene utilizzato principalmente per il supporto di array N-dimensionali. Questi array multidimensionali sono 50 volte più robusti rispetto alle liste Python, rendendo NumPy uno dei preferiti per i data scientist.

NumPy viene utilizzato anche da altre librerie come TensorFlow per il loro calcolo interno sui tensori. NumPy fornisce anche funzioni precompilate veloci per routine numeriche, che possono essere difficili da risolvere manualmente. Per ottenere una migliore efficienza, NumPy utilizza calcoli orientati agli array, quindi lavorare con più classi diventa facile.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install numpy #se hai installato python 2
pip3 install numpy  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import numpy
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

3. Scikit-learn

Scikit-learn è probabilmente la libreria più importante in Python per l'apprendimento automatico. Dopo aver pulito e manipolato i dati con Pandas o NumPy, scikit-learn viene utilizzato per creare modelli di apprendimento automatico in quanto dispone di tonnellate di strumenti utilizzati per la modellazione e l'analisi predittiva.

Ci sono molte ragioni per usare scikit-learn. Per citarne alcuni, è possibile utilizzare scikit-learn per creare diversi tipi di modelli di apprendimento automatico, supervisionati e non supervisionati, convalidare in modo incrociato l'accuratezza dei modelli e condurre l'importanza delle funzionalità.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install scikit-learn #se hai installato python 2
pip3 install scikit-learn  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import scikit-learn
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

4. Gradio

Gradio ti consente di creare e distribuire app Web per i tuoi modelli di machine learning in sole tre righe di codice. Ha lo stesso scopo di Streamlit o Flask, ma ho trovato molto più veloce e più facile ottenere un modello distribuito.

Gradio è utile per i seguenti motivi:

1. Consente un'ulteriore convalida del modello. In particolare, consente di testare in modo interattivo diversi input nel modello.
2. È un buon modo per condurre demo.
3. È facile da implementare e distribuire perché l'app Web è accessibile da chiunque tramite un collegamento pubblico.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install gradio #se hai installato python 2
pip3 install gradio #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import gradio
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

5. TensorFlow

TensorFlow è una delle librerie più popolari di Python per l'implementazione di reti neurali. Utilizza array multidimensionali, noti anche come tensori, che gli consentono di eseguire diverse operazioni su un particolare input.

Poiché è di natura altamente parallela, può addestrare più reti neurali e GPU per modelli altamente efficienti e scalabili. Questa funzionalità di TensorFlow è anche chiamata pipelining.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install tensorflow #se hai installato python 2
pip3 install tensorflow  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import tensorflow
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

6. Keras

Keras viene utilizzato principalmente per creare modelli di apprendimento profondo, in particolare reti neurali. È basato su TensorFlow e Theano e ti consente di creare reti neurali in modo molto semplice. Poiché Keras genera un grafico computazionale utilizzando l'infrastruttura back-end, è relativamente lento rispetto ad altre librerie.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install keras #se hai installato python 2
pip3 install keras  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import keras
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

7. SciPy

Come suggerisce il nome, SciPy è utilizzato principalmente per le sue funzioni scientifiche e le funzioni matematiche derivate da NumPy. Alcune funzioni utili fornite da questa libreria sono le funzioni di statistica, le funzioni di ottimizzazione e le funzioni di

elaborazione del segnale. Per risolvere equazioni differenziali e fornire l'ottimizzazione, include funzioni per il calcolo numerico degli integrali. Alcune delle applicazioni che rendono importante SciPy sono:

- Elaborazione di immagini multidimensionali
- Capacità di risolvere trasformate di Fourier ed equazioni differenziali
- Grazie ai suoi algoritmi ottimizzati, può eseguire calcoli di algebra lineare in modo molto robusto ed efficiente

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install scipy #se hai installato python 2
pip3 install scipy  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import scipy
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

---

8. Statsmodels

Statsmodels è un'ottima libreria per fare statistiche hardcore. Questa libreria multifunzionale è una miscela di diverse librerie Python, che prende le sue caratteristiche grafiche e funzioni da Matplotlib, per la gestione dei dati, usa Pandas, per la gestione di formule R-like, usa Pasty ed è costruita su NumPy e SciPy.

In particolare, è utile per creare modelli statistici, come OLS, e anche per eseguire test statistici.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install statsmodels #se hai installato python 2
pip3 install tensstatsmodelsorflow  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import statsmodels
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

9. Plotly

Plotly è sicuramente uno strumento indispensabile per la creazione di visualizzazioni poiché è estremamente potente, facile da usare e ha un grande vantaggio di essere in grado di interagire con le visualizzazioni.

Insieme a plotly c'è Dash, uno strumento che ti consente di creare dashboard dinamici utilizzando visualizzazioni Plotly. Dash è un'interfaccia Python basata sul Web che elimina la necessità di JavaScript in questi tipi di applicazioni Web analitiche e consente di eseguire questi grafici online e offline.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install plotly #se hai installato python 2
pip3 install plotly  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import plotly
```

E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

10. Seaborn

Costruito sulla parte superiore di Matplotlib, seaborn è una libreria efficace per la creazione di diverse visualizzazioni.

Una delle caratteristiche più importanti di Seaborn è la creazione di dati visivi amplificati. Alcune delle correlazioni che inizialmente non sono ovvie possono essere visualizzate in un contesto visivo, consentendo ai Data Scientist di comprendere i modelli in modo più appropriato.

Per installare questa Libreria, apri il terminale del tuo PC e digita :

```
pip install seaborn #se hai installato python 2
pip3 install seaborn  #se hai installato python 3
```

Per verificare la corretta installazione, sempre dal tuo terminale digita :

```
python #se hai installato python 2
python3 #se hai installato python 3
```

Una volta premuto Invio, e aperto l'interprete Python, digita :

```
import seaborn
```

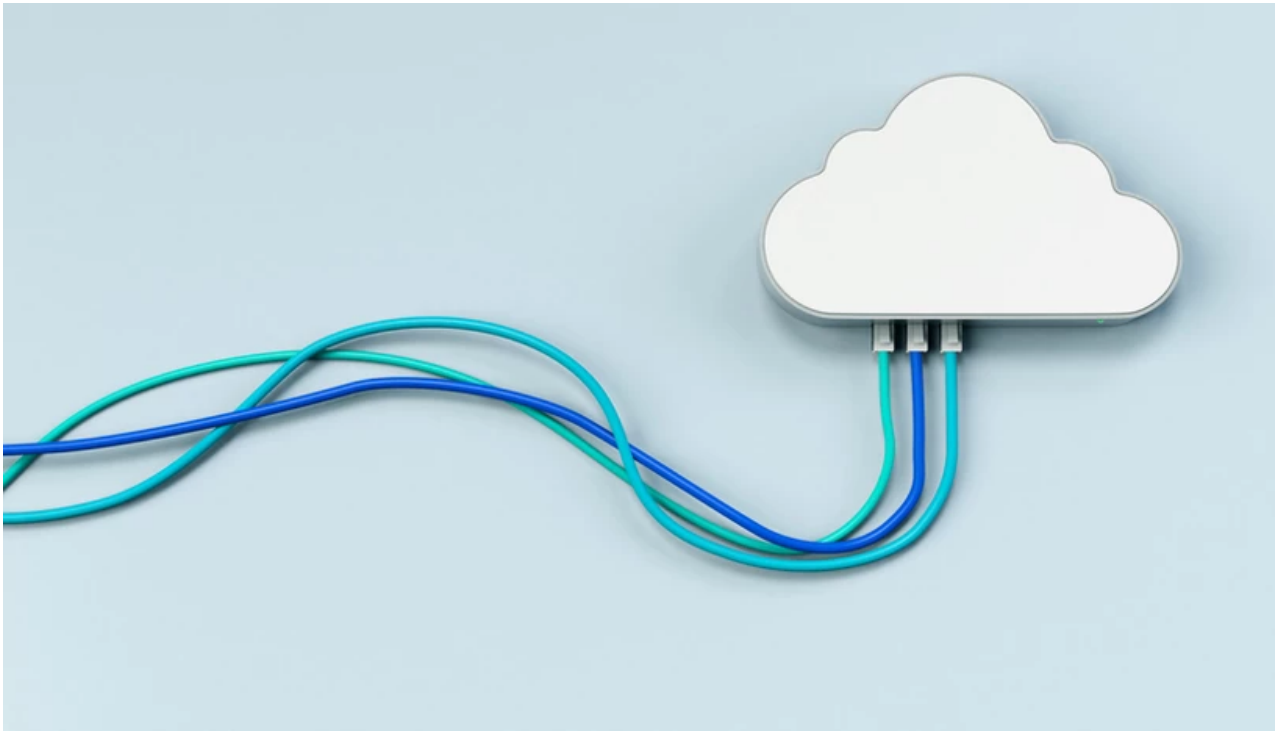
E premi invio, se non compare nessun messaggio di errore l'installazione è andata a buon fine

Grazie ai suoi temi personalizzabili e alle interfacce di alto livello, fornisce visualizzazioni di dati straordinarie e ben progettate, rendendo quindi le trame molto attraenti, che possono, in seguito, essere mostrate alle parti interessate.

I 60 migliori set di dati gratuiti per l'apprendimento automatico e profondo

Indice Tipologie di Dataset:

1. **I cinque principali strumenti di ricerca dataset gratis**
2. **Set di dati governativi per l'apprendimento automatico**
3. **Set di dati finanziari ed economici per l'apprendimento automatico**
4. **Set di dati di immagini per la visione artificiale**
5. **Set di dati per la sentiment analysis**
6. **Set di dati per l'elaborazione del linguaggio naturale**
7. **Set di dati per veicoli autonomi**



I 60 migliori set di dati gratuiti per l'apprendimento automatico e profondo

In questo articolo abbiamo raccolto 60 set di dati per l'apprendimento automatico, che vanno da dati altamente specifici a set di dati di prodotti Amazon. Prima di iniziare ad aggregare questi dati, è importante verificare alcune cose. Innanzitutto, assicurati che i set

di dati non siano "pompati", poiché probabilmente non vorrai perdere tempo a setacciare e ripulire i dati da solo. In secondo luogo, tieni presente che i set di dati con meno righe e colonne richiedono meno tempo in generale e sono anche più facili da utilizzare .

I cinque principali strumenti di ricerca dataset gratis

Quando si padroneggia l'apprendimento automatico, fare pratica con diversi set di dati è un ottimo punto di partenza. Fortunatamente, trovarli è facile.

Kaggle : questo sito di data science contiene un insieme diversificato di set di dati interessanti e forniti in modo indipendente per l'apprendimento automatico. Se stai cercando set di dati di nicchia, il motore di ricerca di Kaggle ti consente di specificare le categorie per assicurarti che i set di dati che trovi si adattino alla tua fattura.

UCI Machine Learning Repository : questo pilastro dei set di dati aperti è stato un punto di riferimento per decenni. Poiché molti dei set di dati sono forniti dall'utente, è imperativo controllarne la qualità poiché i livelli di pulizia possono variare. Vale la pena notare, tuttavia, che la maggior parte dei set di dati sono puliti, il che rende questo repository un punto di riferimento. Gli utenti possono anche scaricare i dati senza bisogno di registrarsi.

Ricerca di set di dati di Google : la ricerca di set di dati contiene oltre 25 milioni di set di dati provenienti da tutto il Web. Che siano ospitati sul sito di un editore, su un dominio governativo o sul blog di un ricercatore, Dataset Search può trovarlo.

AWS Open Data Registry : ovviamente anche Amazon ha le mani nel barattolo dei cookie del set di dati aperto. Il colosso dello shopping porta la sua intraprendenza caratteristica nel gioco di ricerca di set di dati. Un vantaggio chiave che differenzia AWS Open Data Registry è la sua funzione di feedback degli utenti, che consente agli utenti di aggiungere e modificare i set di dati. L'esperienza con AWS è anche altamente preferita nel mercato del lavoro.

Set di dati ML di Wikipedia : questa pagina di Wikipedia presenta diversi set di dati per l'apprendimento automatico, inclusi segnali, immagini, suoni e testo, solo per citarne alcuni.

Set di dati governativi per l'apprendimento automatico

Se stai cercando dati demografici per i tuoi algoritmi di machine learning, non cercare oltre questi portali di dati governativi. I modelli ML formati tramite i dati del governo pubblico possono consentire ai responsabili delle politiche di riconoscere e anticipare le tendenze che informano le decisioni politiche preventive.

Data USA : Data USA offre una fantastica gamma di dati pubblici statunitensi visualizzati in modo potente. Le informazioni sono digeribili e facilmente accessibili, rendendo facile vagliare e selezionare se è giusto per te.

Portale Open Data dell'UE : questo portale di dati aperti offre oltre un milione di set di dati in 36 paesi europei pubblicati da rinomate istituzioni dell'UE. Il sito ha un'interfaccia facile da usare che ti consente di cercare set di dati specifici in una varietà di categorie tra cui energia, sport, scienza ed economia.

Data.gov : questo sito è fantastico per chiunque cerchi di scaricare una moltitudine di fonti di dati disponibili pubblicamente dalle agenzie governative degli Stati Uniti. I dati sono diversi e vanno dai dati di bilancio ai punteggi delle prestazioni scolastiche. Le informazioni spesso richiedono ulteriori ricerche, che è qualcosa da tenere a mente.

Dati sanitari statunitensi : un ricco repository che presenta naturalmente tonnellate di set di dati sui dati sanitari statunitensi.

Il servizio dati del Regno Unito : questo archivio di dati presenta la più grande raccolta di dati sociali, economici e demografici del Regno Unito.

Finanze del sistema scolastico : un archivio favoloso per chiunque sia interessato ai dati finanziari dell'istruzione come entrate, spese, debito e risorse dei sistemi scolastici pubblici elementari e secondari. Le statistiche su questo sito coprono anche i sistemi scolastici negli Stati Uniti, incluso il Distretto di Columbia.

Il National Center for Education Statistics degli Stati Uniti: questo archivio contiene informazioni sulle istituzioni educative e sui dati demografici non solo dagli Stati Uniti, ma anche da tutto il mondo.

Set di dati finanziari ed economici per l'apprendimento automatico

Naturalmente il settore finanziario sta abbracciando il Machine Learning a braccia aperte. Poiché i record quantitativi finanziari ed economici sono in genere tenuti meticolosamente, la finanza e l'economia sono un ottimo argomento per implementare un modello AI o ML. Sta già accadendo, poiché molte società di investimento utilizzano algoritmi per guidare le loro scelte di azioni, previsioni e operazioni. L'apprendimento automatico viene utilizzato anche nel campo dell'economia per cose come testare modelli economici o analizzare e prevedere il comportamento delle popolazioni.

American Economic Association (AEA) : L'AEA è una fonte fantastica per i dati macroeconomici statunitensi.

Quandl : Un'altra grande fonte di dati economici e finanziari, in particolare per costruire modelli predittivi su azioni e indicatori economici.

Dati del FMI : il Fondo monetario internazionale tiene traccia e conserva meticolosamente i registri relativi alle riserve valutarie, ai risultati degli investimenti, ai prezzi delle materie prime, ai tassi di debito e alle finanze internazionali.

Dati aperti della Banca mondiale : i set di dati della Banca mondiale coprono la demografia della popolazione insieme a un numero elevato di indicatori economici e di sviluppo in tutto il mondo.

Dati di mercato del Financial Times : ottimo per informazioni aggiornate su materie prime, cambi e altri mercati finanziari mondiali.

Google Trends : Google Trends ti dà la libertà di esaminare e analizzare tutte le attività di ricerca su Internet e offre anche scorci su quali storie sono di tendenza in tutto il mondo.

Set di dati di immagini per la visione artificiale

Chiunque desideri addestrare applicazioni di visione artificiale come veicoli autonomi, riconoscimento facciale e tecnologia di imaging medico avrà bisogno di un database di immagini. Questo elenco contiene una serie diversificata di applicazioni che si riveleranno utili.

VisualQA : se hai una comprensione della visione e del linguaggio, questo set di dati è utile in quanto contiene domande complesse relative a oltre 265.000 immagini.

Labelme : questo set di dati per l'apprendimento automatico è già annotato, il che lo rende pronto e pronto per qualsiasi applicazione di visione artificiale.

ImageNet : il set di dati di apprendimento automatico per i nuovi algoritmi, questo set di dati è organizzato secondo la gerarchia di WordNet, il che significa che ogni nodo è in realtà solo tonnellate di immagini.

Riconoscimento scena interna : questo set di dati altamente specificato contiene immagini utili per i modelli di riconoscimento scena.

Genoma visivo : oltre 100.000 immagini altamente dettagliate e didascalie.

Stanford Dogs Dataset : ottimo per gli amanti dei cani tra noi, questo set di dati contiene oltre 20.000 immagini di oltre 120 diverse razze di cani.

Immagini aperte di Google : oltre 9 milioni di URL di immagini annotate in 6.000 categorie.

Facce etichettate nella casa selvaggia : set di dati particolarmente utile per le applicazioni che coinvolgono il riconoscimento facciale.

COIL-100 : contiene 100 oggetti che vengono ripresi su più angolazioni per una vista completa a 360 gradi.

CIFAR-10 : il set di dati CIFAR-10 è composto da 60000 immagini a colori 32×32 in 10 classi, con 6000 immagini per classe. Ci sono immagini di allenamento da 50K e immagini di prova da 10K.

Cityscapes : Cityscapes contiene annotazioni a livello di pixel di alta qualità di 5.000 fotogrammi oltre a un set più ampio di 20.000 fotogrammi con annotazioni scadenti.

IMDB-Wiki : in questo set di dati sono presenti oltre 500K+ immagini di volti che sono state raccolte sia su IMDB che su Wikipedia.

Fashion MNIST : Questo è un set di dati delle immagini degli articoli di Zalando. Contiene un training set di 60.000 esempi e un test set di 10.000 esempi.

MS COCO : questo set di dati contiene foto di vari oggetti e contiene oltre 2 milioni di istanze etichettate su oltre 300K immagini.

MPII Human Pose Dataset : questo set di dati include 25K immagini contenenti oltre 40K persone con articolazioni del corpo annotate. È perfetto per la valutazione della stima articolata della posa umana.

Set di dati per la sentiment analysis

Esistono innumerevoli modi per migliorare qualsiasi algoritmo di analisi del sentiment. Questi grandi set di dati altamente specializzati possono essere d'aiuto.

Set di dati di analisi del sentiment multidominio : un tesoro di recensioni di prodotti Amazon positive e negative (da 1 a 5 stelle) per i prodotti più vecchi.

Dati sui prodotti Amazon : con 142,8 milioni di set di dati di recensioni Amazon, questo set di dati SA include recensioni aggregate su Amazon tra il 1996 e il 2014.

Twitter US Airline Sentiment : dati Twitter sulle compagnie aeree statunitensi risalenti a febbraio 2015 che sono già stati classificati in base alla classe di sentiment (positivo, neutro, negativo).

IMDB Sentiment : questo set di dati più piccolo (e più vecchio) è perfetto per la classificazione binaria del sentimento e presenta oltre 25.000 recensioni di film.

Sentiment140 : uno dei set di dati più popolari che contiene oltre 160.000 tweet che sono stati controllati per le emoticon (che sono stati successivamente rimossi).

Stanford Sentiment Treebank : set di dati contenente oltre 10.000 file HTML Rotten Tomatoes con annotazioni sui sentimenti basate su una scala 1 (negativa) e 25 (positiva).

Recensioni cartacee : questo set di dati è composto da recensioni in lingua inglese e spagnola su informatica e informatica. Il set di dati viene valutato utilizzando una scala a cinque punti, dove -2 è il più negativo e 2 il più positivo.

Lexicoder Sentiment Dictionary : questo dizionario è progettato per essere utilizzato in conformità con Lexicoder, che aiuta nella codifica automatica del sentimento della copertura delle notizie, del discorso legislativo e di altri testi.

Lessici dei sentimenti per 81 lingue : questo set di dati contiene oltre 81 lingue esotiche con lessici dei sentimenti positivi e negativi, con i sentimenti analizzati e basati sui lessici dei sentimenti inglesi.

Opin-Rank Review Dataset : questo dataset di auto contiene una serie di recensioni sui modelli prodotti tra il 2007 e il 2009. Contiene anche i dati sulle recensioni degli hotel.

Set di dati per l'elaborazione del linguaggio naturale

L'elenco seguente contiene diversi set di dati per varie attività di elaborazione della PNL, inclusi il riconoscimento vocale e i chatbot.

Enron Dataset : dati e-mail di gestione senior organizzati in cartelle da Enron.

UCI's Spambase : un succoso set di dati sullo spam perfetto per il filtraggio dello spam.

Recensioni su Amazon : ancora un altro tesoro contenente 35 milioni di recensioni su Amazon in 18 anni con recensioni di prodotti, informazioni sugli utenti e persino la visualizzazione del testo in chiaro.

Recensioni di Yelp : 5 milioni di recensioni di Yelp in un set di dati aperto.

Google Books Ngrams : questa libreria di parole è abbondante per qualsiasi algoritmo di PNL.

SMS Spam Collection in inglese : oltre 5500 messaggi SMS di spam (in inglese).

Rischio : oltre 200.000 domande dal classico quiz show.

Elenco eBook Gutenberg : un elenco annotato degli ebook del Progetto Gutenberg.

Blogger Corpus : uno stuolo di blog (600K+) con un minimo di 200 occorrenze in ciascuna delle parole inglesi più comunemente usate.

Wikipedia Links Data : oltre 1,9 miliardi di parole su 4 milioni di articoli, questo set di dati contiene l'intero testo di Wikipedia.

Set di dati per veicoli autonomi

I veicoli autonomi richiedono grandi quantità di set di dati di alta qualità per interpretare l'ambiente circostante e reagire di conseguenza.

Berkeley DeepDrive BDD100K : questo set di dati AI a guida autonoma è considerato il più grande del suo genere. Presenta oltre 100.000 video di 1.100 ore di guida in diversi orari, condizioni meteorologiche e di guida.

L'auto robotica di Oxford: set di dati di Oxford, Regno Unito con 100 ripetizioni di un singolo percorso in diverse ore del giorno, condizioni meteorologiche e di guida (traffico, condizioni meteorologiche, pedoni).

Cityscapes Dataset : un insieme diversificato di dati di scene di strada in 50 città diverse.

Baidu Apolloscapes : questo set di dati include 26 diversi elementi semantici tra cui lampioni, pedoni, edifici, biciclette, automobili e altro ancora.

Landmarks : set di dati di Google open source progettato per distinguere tra formazioni naturali e punti di riferimento creati dall'uomo. Questo set di dati include oltre due milioni di immagini in 30 mila punti di riferimento in tutto il mondo.

Landmarks-v2 : con il miglioramento della tecnologia di classificazione delle immagini, Google ha deciso di rilasciare un altro set di dati per aiutare con i punti di riferimento. Questo set di dati ancora più grande include cinque milioni di immagini con più di 200 mila punti di riferimento in tutto il mondo.

PandaSet : PandaSet sta lavorando per promuovere e far progredire la guida autonoma e la ricerca e sviluppo ML. Questo set di dati include oltre 48.000 immagini della fotocamera, oltre 16.000 scansioni LiDar, oltre 100 scene di 8 secondi ciascuna, 28 classi di annotazioni, 37 etichette di segmentazione semantica e si estende all'intera suite di sensori.

nuScenes : questo set di dati su larga scala per veicoli autonomi utilizza l'intera suite di sensori di una vera auto a guida autonoma su strada. Questo vasto set di dati include immagini della fotocamera da 1,4 milioni, scansioni LiDar da 390K, informazioni cartografiche intime e altro ancora.

OpenImageV5 : questo set di dati è costituito da oltre 9 milioni di immagini annotate ed etichettate in migliaia di categorie di oggetti.

Waymo Open Dataset : questo set di dati di sensori multimodali open source e di alta qualità viene estratto dai veicoli a guida autonoma Waymo in una serie diversificata di ambienti.

Sfruttiamo il potere della condivisione

Il tuo Primo Programma di Machine Learning con Python e Google Colab

Indice

Perchè è importante saper creare un programma di Machine Learning ?

Perchè hai difficoltà a seguire i tutorial degli altri ?

Cosa Utilizzeremo per creare un programma con solo il browser?

Passaggio 1. Creazione di un taccuino python con Google Colab

Passaggio 2. Importare le librerie su Google colab

Passaggio 3. Come importare Dataset su Google Colab

Passaggio 4. Addestrare un Modello di Machine Learning su Google Colab

Passaggio 5. Fare previsioni con Python utilizzando Google Colab

Conclusioni

Perchè è importante saper creare un programma di Machine Learning ?

L'apprendimento automatico (ML) è di tendenza e ogni azienda vuole sfruttare il machine learning per aiutarla a migliorare i propri prodotti o servizi. Pertanto, abbiamo osservato una crescente domanda di ingegneri ML e tale richiesta ha attirato l'attenzione di molte persone. Tuttavia, il machine learning può sembrare scoraggiante per molti, specialmente per coloro che hanno poca **esperienza di programmazione o di lavoro relativa ai dati.**

Perchè hai difficoltà a seguire i tutorial degli altri ?

Una ragione probabile è che ci vuole una grande quantità di sforzi per impostare il computer, permettendo loro di sviluppare tutti i modelli ML.

> **Ci Pensiamo noi...Ti faremo creare il tuo primo script di machine learning con python, utilizzando solo un browser**

Cosa Utilizzeremo ?

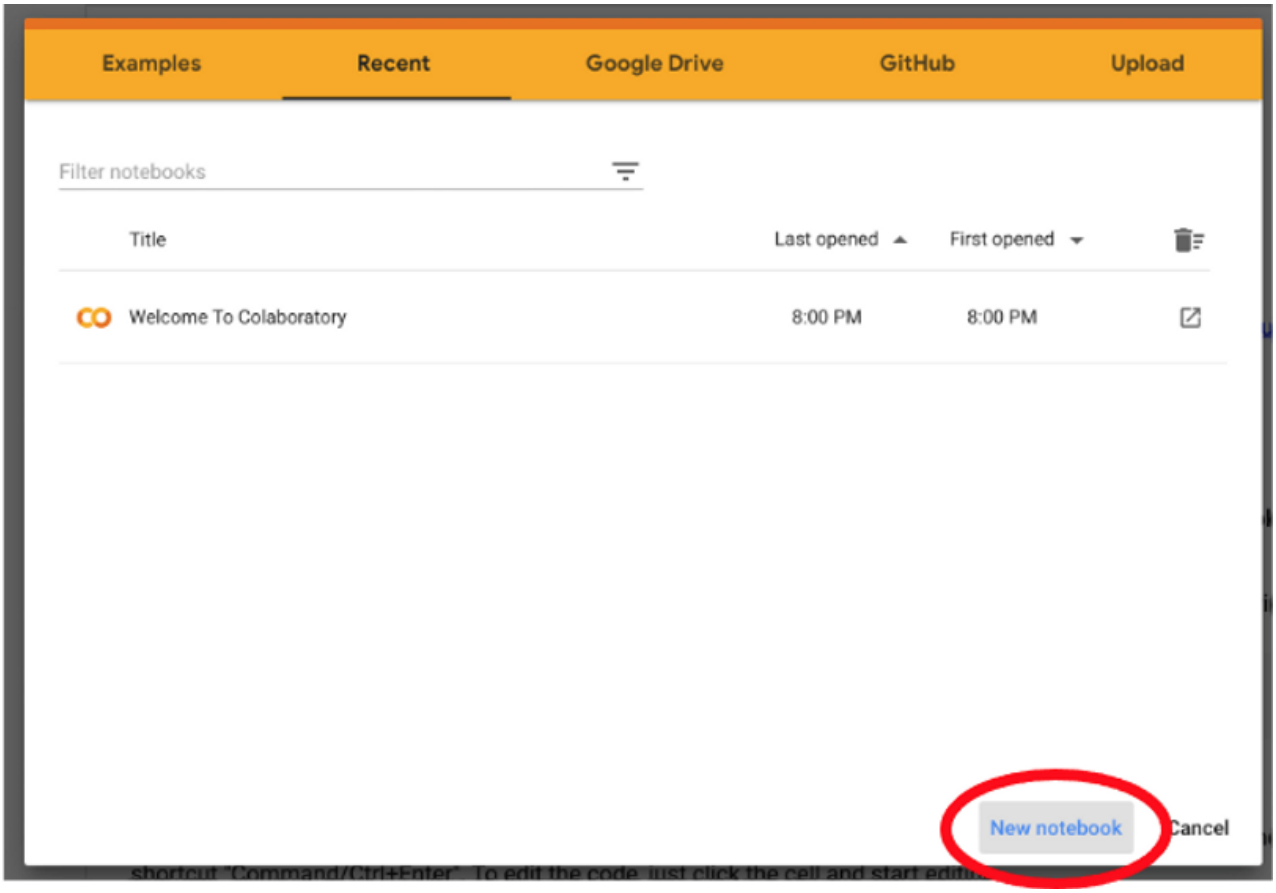
In questo articolo vorrei presentare Google Colab, uno strumento gratuito (con opzioni di aggiornamento a pagamento, però) per l'apprendimento e la creazione di modelli ML. Ancora più importante, come scoprirai, non ha alcuna configurazione per te - è pronto per l'uso ora - con l'unico requisito di avere un account Google. Se non ne hai uno, registrati in modo da poter seguire il tutorial.

Suppongo che tu non sappia molto di ML, ma sei molto entusiasta di apprendere il ML. Non importa quanto Python conosci. Spiegherò i passaggi principali usando il più possibile termini comprensibili anche per chi è alle prime armi.

Senza ulteriori indugi, iniziamo. Se vuoi vedere il codice sorgente, puoi accedere al Notebook passando nell'area progetti . Troverai il Link alla fine dell articolo.

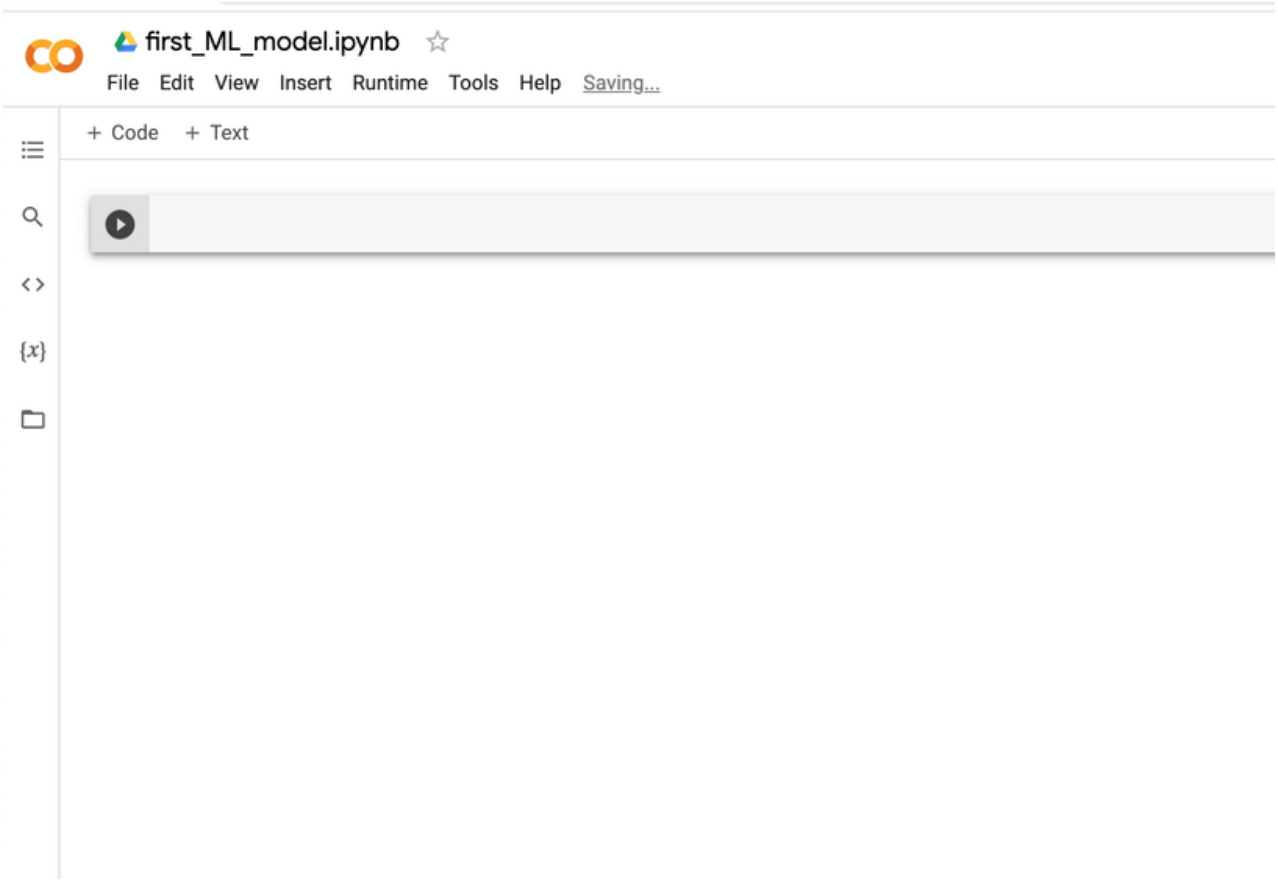
Passaggio 1. Creazione di un taccuino in Colab

In Colab, lavori con i taccuini (*.ipynb), proprio come lavori con i documenti (*.docx) in Microsoft Word. Quindi, il primo passo per usare Colab è creare un nuovo Notebook andando su: https://colab.research.google.com/ .



Creare un File python su Colab

Dopo aver fatto clic sul pulsante "New notebook", vedrai che Colab crea un nuovo taccuino con un nome predefinito di Untitled1.ipynb.



Creare rinominare un file su Google Colab

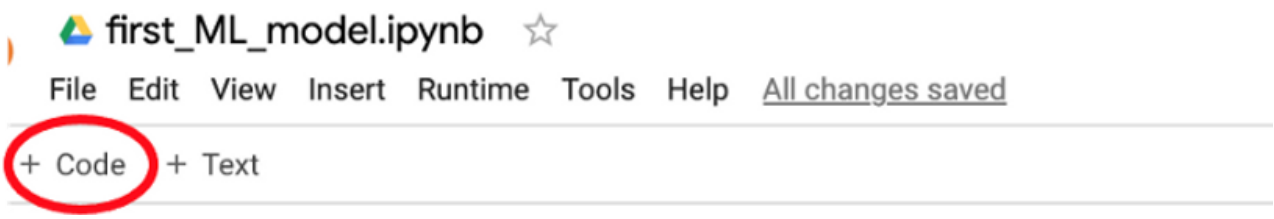
Per il bene dell'attuale tutorial, chiamiamolo first_ml_model.ipynb o come vuoi. D'ora in poi lavoreremo su questo file.

Passaggio 2. Importare le librerie su Google colab

Quando creiamo il nostro modello, abbiamo bisogno di librerie di codici sviluppate da sviluppatori esperti di machine learning. In sostanza, queste librerie fungono da set di strumenti fornendo funzionalità predefinite per l'elaborazione dei dati e la costruzione del modello. In questo tutorial, utilizzeremo principalmente le seguenti librerie.

- scikit-learn : una libreria ML che consiste in una varietà di funzioni di elaborazione dati e algoritmi ML (ad es. regressione, classificazione e clustering). Questa libreria è anche conosciuta come sklearn.
- pandas : una libreria di data science specializzata principalmente nella pre-elaborazione di dati simili a fogli di calcolo prima di creare modelli ML.

Nel taccuino di Google Colab, ogni unità di lavoro è nota come cella e usiamo una serie di celle per svolgere i nostri lavori di machine learning. In ogni cella, di solito svolgiamo un compito specifico. Per aggiungere una cella, fai semplicemente clic + ***Code*** su in alto, come mostrato di seguito. Puoi aggiungere le tue note sul codice facendo clic su + ***Text***.



Aggiungere codice o testo su Colab

Con il codice creato, eseguendo la cella seguente, puoi importare le librerie necessarie per il presente tutorial.

```
from sklearn import datasets, model_selection, metrics, ensemble
import pandas as pd
```

Importiamo le librerie necessarie su Google Colab

Una piccola Nota, se stai cercando di configurare un computer per il ML, devi installare tutte queste librerie oltre alla configurazione di Python. Mentre su Google Colab per creare il tuo primo programma di Machine learning avrai solo bisogno del nostro semplice tutorial.

Passaggio 3. Come importare Dataset su Colab

Per l'attuale tutorial, utilizzeremo il set di dati sulla qualità del vino rosso. Puoi trovare maggiori informazioni su questo set di dati su kaggle.com , un popolare sito Web di data science e ML che presenta una serie di concorsi. Puoi anche trovare le informazioni del set di dati su UCI , che è il principale repository di dati ML.

Tranquillo il Dataset sarà incluso nella cartella file del relativo progetto

Il dataset del vino è spesso usato come esempio per mostrare i modelli ML e, quindi, è comodamente disponibile nella libreria sklearn. Tieni presente che il set di dati in sklearn è stato modificato per servire meglio come set di dati giocattolo per l'addestramento e l'apprendimento ML. I dati sono riportati di seguito.

Creare Il tuo Primo Programma di Machine Learning sul Web

```
[2]  from sklearn import datasets, model_selection, metrics, ensemble
     import pandas as pd
```

Importiamo i dati

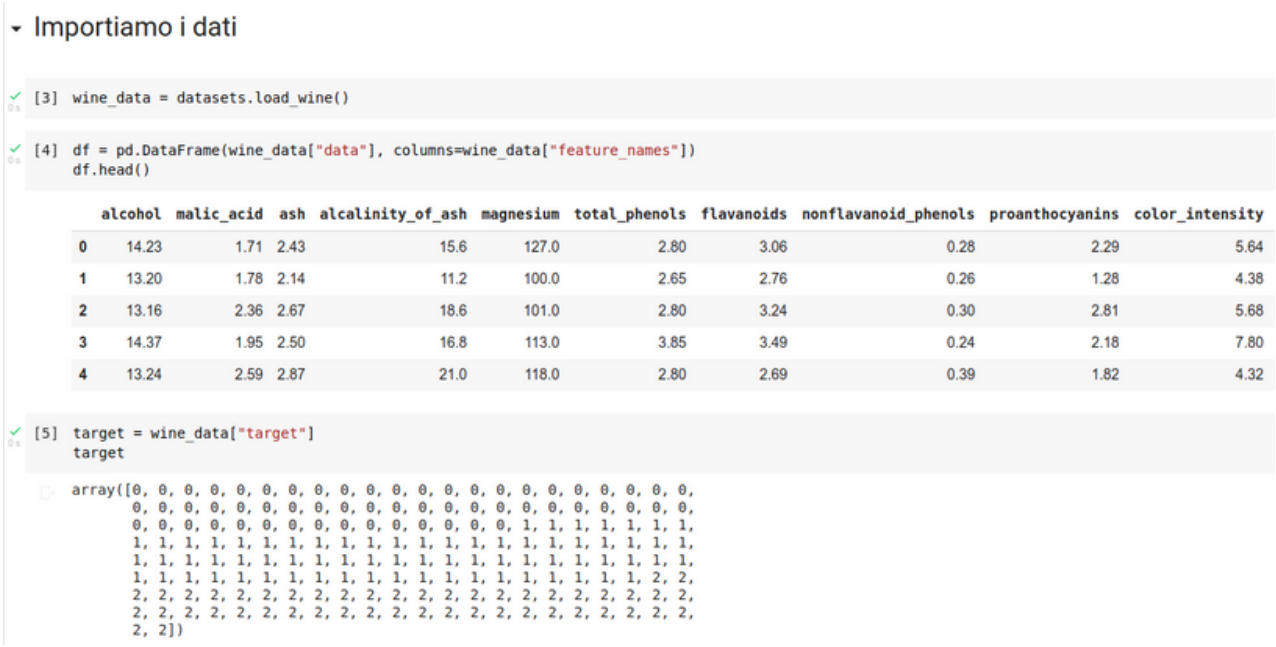
```
[3]  wine_data = datasets.load_wine()
```

```
[4]  df = pd.DataFrame(wine_data["data"], columns=wine_data["feature_names"])
     df.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69

Importare dati su Google Colab e stamparli

Lo screenshot sopra mostra le caratteristiche dei dati. In ML, utilizziamo le " **caratteristiche** " per studiare quali fattori possono essere importanti per la previsione corretta. Come puoi vedere, ci sono 12 funzioni disponibili e sono potenzialmente importanti per la qualità del vino rosso, come l'alcol e l'acido malico. Un ML specifico è preoccupato per la classificazione. Ogni record di dati ha un'etichetta che mostra la sua classe e le classi di tutti i record sono conosciute come "destinazione" del set di dati. Nel set di dati del vino rosso, ci sono tre classi per le etichette e possiamo controllare le etichette, come mostrato di seguito:



Importamo i dati

```
[3] wine_data = datasets.load_wine()
```

```
[4] df = pd.DataFrame(wine_data["data"], columns=wine_data["feature_names"])
    df.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32

```
[5] target = wine_data["target"]
    target
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])
```

Importare Dati e selezionare Variabile Target Google Colab

Tieni presente che in una tipica pipeline, di solito è necessario dedicare un sacco di tempo alla preparazione del set di dati. Alcune preparazioni comuni includono l'identificazione e la rimozione/ricodifica di valori anomali, la gestione dei dati mancanti, la ricodifica one-hot (necessaria per alcuni modelli), la riduzione della dimensionalità, la selezione delle funzionalità, il ridimensionamento e molti altri. Poiché il set di dati è stato ripulito come un set di dati giocattolo in sklearn , non dobbiamo preoccuparci di questi preparativi.

Passaggio 4. Addestrare un Modello di Machine Learning su Google Colab

Il passaggio successivo consiste nell'addestrare il modello ML. Forse ti starai chiedendo qual è il punto di addestrare un modello ML. Bene, per diversi casi d'uso, ci sono scopi diversi. Ma in generale, lo scopo dell'addestramento di un modello ML è più o meno quello di fare previsioni su cose che non hanno mai visto. Il modello riguarda come fare buone previsioni. Il modo per creare un modello si chiama training, utilizzando i dati esistenti per identificare un modo corretto per fare previsioni.

Esistono molti modi diversi per costruire un modello, come K-nearest neighbors, SVC, random forest, e gradient boosting, solo per citarne alcuni. Ai fini del presente tutorial che mostra come creare un modello ML utilizzando Google Colab, utilizziamo un modello prontamente disponibile in sklearn: il classificatore di foreste casuali.

Una cosa da notare è che abbiamo un solo set di dati. Per testare le prestazioni del modello, divideremo il set di dati in due parti, una per l'addestramento e l'altra per il test. Possiamo semplicemente usare il ***train_test_split*** , come mostrato di seguito. Il set di

dati di addestramento ha 142 record, mentre il set di dati di test ha 36 record, approssimativamente in un rapporto di 4:1 **(si noti che test_size=0.2 per i test viene utilizzato il 20% con arrotondamenti se necessario del set di dati originale)**.

```
[ ] target = wine_data["target"]
    target

    array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
           2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
           2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
           2, 2])
```

Dividiamo i dati in DatiAllenamento(train) e Datitest(test)

```
[ ] X_train, X_test, y_train, y_test = model_selection.train_test_split(df, target, test_size=0.2, random_state=0)
    print("Training Dataset:", X_train.shape)
    print("Test Dataset:", X_test.shape)

    Training Dataset: (142, 13)
    Test Dataset: (36, 13)
```

Train e Test split su Google Colab

La cosa bella di sklearn è che fa un sacco di lavoro pesante per noi rendendo molti classificatori preconfigurati in modo tale da poterli usare con poche righe di codice. Nello screenshot qui sotto, creiamo prima un classificatore forestale casuale. In sostanza, crea il framework in cui inserire i nostri dati per costruire il modello.

Ora facciamo quella cosa figa che fanno i DatScientist,

Alleniamo il Modello

```
[ ] classifier = ensemble.RandomForestClassifier()
    model = classifier.fit(X_train, y_train)
    model

    RandomForestClassifier()
```

Creazione di un classificatore forestale casuale

Utilizzando ***classifier.fit***, stiamo addestrando il modello per generare i parametri del modello, in modo tale che il modello possa essere utilizzato per previsioni future.

Passaggio 5. Fare previsioni con Python utilizzando Google Colab

Con il modello addestrato di sklearn, possiamo testare le prestazioni del modello sul set di dati di test che abbiamo creato in precedenza. Come mostrato di seguito, abbiamo ottenuto una previsione di accuratezza del 97,2%. Tieni presente che raggiungere un livello elevato come questo in un set di dati giocattolo non è atipico, ma è considerato molto alto nei progetti reali.

Ora facciamo quella cosa figa che fanno i DatScientist, Alleniamo il Modello

```
[ ] classifier = ensemble.RandomForestClassifier()
    model = classifier.fit(X_train, y_train)
    model

    RandomForestClassifier()
```

Ora facciamo quell'altra cosa figa che fanno i DatScientist, Faciamo le predizioni con il nostro modello

```
[ ] model.score(X_test, y_test)

    0.9722222222222222
```

```
[ ] y_pred = model.predict(X_test)
    report = metrics.classification_report(y_test, y_pred)
    print(report)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        14
           1       1.00      0.94      0.97        16
           2       0.86      1.00      0.92         6

    accuracy                           0.97        36
   macro avg       0.95      0.98      0.96        36
weighted avg       0.98      0.97      0.97        36
```

Verifica Precisione Modello classificatore forestale casuale

Se vuoi dare un'occhiata più da vicino alla previsione del nostro modello, **questo è il link**al progetto. All'interno troverai una cartella.zip contenente il file .ipynb per aprirlo con Colab, il file python per farlo girare in locale, il set di dati in formato csv e il pdf del progetto.

Conclusioni creazione primo programma di machine learning con Python e Google Colab

In questo articolo, abbiamo utilizzato Google Colab come editor di codice per mostrarti come creare un modello ML per fare previsioni su un set di dati giocattolo.

Tuttavia, ti mostra che Google Colab è uno strumento facile da usare che richiede configurazioni minime per iniziare il tuo percorso di apprendimento del machine learning. Quando ti senti a tuo agio con le terminologie e i concetti relativi a Google Colab, Python e ML. Puoi esplorare altri IDE Python, come PyCharm, per un lavoro ML più avanzato con una migliore esperienza di codifica.

Scarica Gratis il progetto