INTRODUZIONE AL CLASSIFICATORE NAIVE BAYES E AL SUO UTILIZZO

Vorrei introdurvi ad un concetto fondamentale nel campo dell'Intelligenza Artificiale e del Machine Learning: il **classificatore Naive Bayes**.

Che cos'è il Classificatore Naive Bayes?

Per comprendere il classificatore (divide i dati in classi / gruppi) Naive Bayes, iniziamo con il **teorema di Bayes**, un principio fondamentale della probabilità che porta il nome del matematico inglese Thomas Bayes del XVIII secolo. Questo teorema ci aiuta a calcolare la probabilità che un evento accada, basandoci sul fatto che un altro evento correlato è già avvenuto.

La parola **"Naive"**, che in inglese significa "ingenuo", indica una semplificazione o un'assunzione che l'algoritmo fa: ritiene che tutte le caratteristiche o gli attributi, che stiamo considerando, siano **indipendenti** tra loro. Anche se nella realtà gli attributi possono essere correlati, questa assunzione semplifica enormemente i calcoli e, sorprendentemente, l'algoritmo funziona molto bene in molte applicazioni pratiche.

Come Funziona il Classificatore Naive Bayes?

Il classificatore Naive Bayes è un tipo di algoritmo di **apprendimento automatico supervisionato**. Il suo scopo è quello di assegnare un'etichetta o una categoria ad un oggetto basandosi sulle sue caratteristiche.
Ecco una panoramica di come funziona.

- **Fase di Addestramento**

L'algoritmo analizza un insieme di dati già etichettati (ad esempio, email già classificate come "spam" o "non spam") e calcola le probabilità delle caratteristiche presenti in ciascuna categoria.

- **Fase di Predizione**

Quando arriva un nuovo dato (ad esempio, una nuova email), l'algoritmo utilizza le probabilità calcolate per determinare a quale categoria è più probabile che appartenga il nuovo dato.

Motivi per l'Utilizzo

Il classificatore Naive Bayes è ampiamente utilizzato per diversi motivi.
• **Semplicità**: è facile da implementare e richiede un tempo di addestramento relativamente breve rispetto ad altri algoritmi più complessi.
• **Efficienza**: funziona bene con grandi quantità di dati, il che è essenziale nell'era dei "Big Data" e non richiede grandi capacità di calcolo.
• **Versatilità**: può essere applicato in vari campi e per diversi tipi di problemi.

Applicazioni Pratiche

Ecco alcuni esempi di come il classificatore Naive Bayes viene utilizzato nel mondo reale.

• **Filtraggio di Spam**
Molti servizi di posta elettronica utilizzano questo algoritmo, come anticipato, per identificare e filtrare le email indesiderate. Analizzando parole chiave e modelli comuni nelle email di spam, il sistema può classificare automaticamente i messaggi in arrivo.
• **Analisi del Sentiment**
Nelle recensioni online o nei social media, il classificatore può determinare se un commento è positivo, negativo o neutro, aiutando le aziende a comprendere l'opinione pubblica sui loro prodotti o servizi.
• **Diagnosi Medica**
In campo sanitario, può aiutare a prevedere la probabilità che un paziente abbia una certa malattia basandosi su sintomi e risultati di esami.
• **Sistemi di Raccomandazione**
Piattaforme come Netflix o Amazon possono utilizzare questo algoritmo per suggerire film, serie TV o prodotti che potrebbero interessare all'utente, basandosi sul suo comportamento precedente.

Vantaggi

• **Efficienza Computazionale**: richiede meno risorse computazionali rispetto ad altri algoritmi più complessi.
• **Buone Prestazioni**: nonostante l'assunzione di indipendenza tra le caratteristiche, offre spesso risultati accurati.
• **Facilità di Interpretazione**: essendo basato su probabilità, è più facile da interpretare rispetto ad algoritmi "black box" come le reti neurali profonde.

Limitazioni

• **Ipotesi di Indipendenza**: nella realtà, le caratteristiche possono essere

correlate. Ad esempio, in un testo, la presenza delle parole “neve” e “freddo” è correlata.
• **Dati Categorici**: funziona meglio con dati categorici piuttosto che con dati numerici continui, anche se esistono versioni modificate per gestire questo tipo di dati.

Alcune definizioni

Per rendere più concreta la comprensione di come funziona il classificatore Naive Bayes, faremo un semplice esempio pratico, ma prima ricapitoliamo i concetti chiave.

• **Probabilità a priori**
La probabilità iniziale di un evento prima di osservare qualsiasi evidenza. Ad esempio, la probabilità che un paziente abbia l’influenza senza conoscere i suoi sintomi.
• **Probabilità condizionata (a Posteriori)**
La probabilità che un evento accada dato che un altro evento è già accaduto. Ad esempio, la probabilità che un paziente abbia la febbre dato che ha l’influenza.
• **Enunciato del teorema**
L’enunciato in maniera semplice è: **La probabilità di A, dato B, è uguale alla probabilità di B, dato A, moltiplicato per la probabilità di A, il tutto diviso per la probabilità di B.**

Interpretazione della Formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B) Questa è la probabilità che l’evento A sia vero dato che B è vero. In altre parole, se sappiamo che B si è verificato (ad esempio, l’e-mail contiene la parola ‘offerta’), vogliamo sapere quanto è probabile che A sia vero (l’e-mail è spam).

P(B|A) Questa è la probabilità che B sia vero se A è vero. Quindi, se sappiamo che l’e-mail è spam (A), questa probabilità ci dice quanto è probabile che contenga la parola ‘offerta’ (B).
P(A) Questa rappresenta la probabilità a priori di A, cioè quanto è probabile che un’e-mail sia spam prima di considerare le caratteristiche specifiche (B).
P(B) Questa è la probabilità a priori di B, ovvero quanto è probabile osservare le caratteristiche B in generale (“offerta”), indipendentemente dalla classe A.

Con questi concetti facciamo un esempio pratico che ci aiuterà a vedere in azione il classificatore Naive Bayes, seguirà poi l’esempio con KNIME, da scaricare.

Applichiamo la formula comple-

Esempio di pura fantasia basato sulla Formula 1: supponiamo che Ferrari sia più forte nei circuiti più lenti e McLaren invece in quelli più veloci. Creiamo quella che si chiama” **tabella di contingenza**” (qui a fianco)

	Ferrari vince (A)	McLaren vince (A')	Totale vittorie
Circuito veloce (B)	4	9	13
Circuito lento (B')	5	2	7
Totale vittorie	9	11	20

Calcoliamo le distribuzioni di frequenza, dividendo sempre per il numero totale di osservazioni: 20 (si veda tabella a fianco)

	Ferrari vince (A)	McLaren vince (A')	Totale vittorie
Circuito veloce (B)	4/20 = **0,20**	9/20=**0,45**	13/20= **0,65**
Circuito lento (B')	5/20=**0,25**	2/20=**0,10**	7/20= **0,35**
Totale vittorie	9/20=**0,45**	11/20=**0,55**	20/20=**1,00**

Calcoliamo le varie probabilità (a fianco):

Probabilità singole:
p(A) = 0,45
p(A') = 0,55
p(B) = 0,65
p(B') = 0,35

Probabilità congiunte (A e B insieme):
p(A ∩ B) = 0,20
p(A' ∩ B) = 0,45
p(A ∩ B') = 0,25
p(A' ∩ B') = 0,10

Calcoliamo le probabilità condizionate (a fianco):

p(A|B) = p(A ∩ B) / p(B) = > 0,20/ 0,65= 0,30
p(A'|B) = p(A' ∩ B) / p(B) = > 0,45/ 0,65= 0,69
p(A|B') = p(A ∩ B') / p(B') = > 0,25/ 0,35= 0,71
p(A'|B') = p(A' ∩ B') / p(B') = > 0,10/ 0,35= 0,29

ta di Bayes, calcolando la p(B|A), quindi, data la vittoria della Ferrari, con quale probabilità si sia corso su di un circuito veloce:

p(B|A)=p(B)*p(A|B)/p(B)*p(A|B)+p(B')*p(A'|B') => 0,65*0,31 / (0,65*0,31) + (0,35*0,71) **=0,45**

Lascio a voi gli altri calcoli. In questo caso l'esempio è semplice e si poteva ottenere anche direttamente dalla tabella, ma se si hanno molte righe e colonne il tutto si complica parecchio.

Il machine learning ci viene però incontro e possiamo usare KNIME.
L'esercizio tratta il riconoscere se un vino è bianco o rosso partendo dalle caratteristiche chimiche.
Calcolare Naive Bayes è molto semplice. Per confronto ho aggiunto anche una Regressione Logistica e i risultati ottenuti sono molto simili con alta precisione.

Risultati Logistic Regression

Scorer View

Confusion Matrix

	red (Predicted)	white (Predicted)	
red (Actual)	319	1	99.69%
white (Actual)	5	975	99.49%
	98.46%	99.90%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
99.54%	0.46%	0.968	1294	6

Risultati Naive Bayes

Scorer View

Confusion Matrix

	red (Predicted)	white (Predicted)	
red (Actual)	313	7	97.81%
white (Actual)	16	964	98.37%
	95.14%	99.28%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
98.23%	1.77%	0.953	1277	23

L'esercizio si può scaricare a questo link:
https://hub.knime.com/zompazompa/spaces/Public/Esercizio_NaiveBayes_11_24~Yaw9XSMZIG4UMlyV/current-state