Data Analysis and Data Mining Project

Heart disease

Introduzione

Di tutte le applicazioni di **machine learning**, diagnosticare una qualsiasi malattia grave usando una **black box** sarà sempre difficile da vendere. Se l'output di un modello è il particolare corso del trattamento (potenzialmente con effetti collaterali), o la chirurgia, o l'assenza di trattamento, le persone vorranno sapere perché.

Questo set di dati fornisce un numero di variabili insieme a una condizione obiettivo di avere o non avere malattie cardiache. Di seguito, i dati vengono utilizzati per la prima volta in un modello di **random forest** semplice, quindi il modello viene esaminato utilizzando gli strumenti e le tecniche di spiegabilità di **machine learning**.

Vediamo nel dettaglio la struttura del data set:

* **age**: L'età della persona in anni
* **sex**: Il sesso della persona (1 = maschio, 0 = femmina)
* **cp:** Il dolore al petto sperimentato (Valore 1: angina tipica, Valore 2: angina atipica, Valore 3: dolore non anginoso, Valore 4: asintomatico)
* **trestbps:** Pressione sanguigna a riposo della persona (mm Hg al momento del ricovero in ospedale)
* **chol:** La misurazione del colesterolo della persona in mg / dl
* **fbs:** Glicemia a digiuno (> 120 mg / dl, 1 = vero; 0 = falso)
* **restecg:** Misurazione elettrocardiografica a riposo (0 = normale, 1 = con anormalità dell'onda ST-T, 2 = mostra ipertrofia ventricolare sinistra probabile o definita secondo i criteri di Estes)
* **thalach:** La frequenza cardiaca massima della persona raggiunta
* **exang:** Angina indotta dall'esercizio (1 = si; 0 = no)
* **oldpeak:** Depressione ST indotta dall'esercizio relativo al riposo
* **slope:** la pendenza del segmento di picco dell'esercizio ST (Valore 1: salita, Valore 2: piano, Valore 3: discesa)
* **ca:** Il numero di vasi principali (0-3)
* **thal:** Disturbo del sangue chiamato talassemia (3 = normale; 6 = difetto fisso; 7 = difetto reversibile)
* **target:** Cardiopatia (0 = no, 1 = si)

Obiettivo

L’obiettivo è di creare un modello per la predizione di malattie cardiache, utlizzando Random Forest Classifier il quale verrà configurato mediante la variazione degli Iper-parametri e ne verrà misurata la perfomance mediante la Cross Validation.

Procedimento

Analisi Dataset

Abbiamo utilizzato I tools della libreria sklearn per vedere:

1. Lo stato dei dati, cioè che i tipi di dati siano coerenti con il loro significato e rilevare la presenza di possibili valori nulli.
2. Sono stati fatti gli istogrammi di tutti i campi del dataset per valutarne la distribuzione.
3. Si trattano le categorical features (dati categorici) utilizzando le dummy features.
4. Guardiamo la Correlation Heatmap per vedere la correlazione tra le features

Preparazione del Dataset

Inizialmente andiamo a eseguire l’algoritmo della Features importance, il quale stabilisce quali “features” o campi sono più importanti al fine della rilevazione del target: nel nostro caso sono stati num\_major\_vessels e max\_hearth\_rate\_achieved. Questo ci aiuta a capire quali campi andrà a utilizzare l’algoritmo di predizione Random forest.

Successivamente abbiamo normalizzato i dati perchè ...

Ora possiamo dividere il Dataset in Training set e Validation set, abbiamo usato l’75% per il training e il 25% per il testing.

Prima prova con algoritmo Random Forest

Facendo girare l’algoritmo predisponendo il dataset come descritto sopra arriviamo a una accuratezza sulla predizione del 82,895% o 77,06%.

Tuning Iper-parametri

A questo punto vediamo se attuando il tuning degli Iperparametri migliora l’accuratezza, individuiamo come iperparametri:

* max\_features
* n\_estimators
* min\_samples\_leaf
* max\_depth
* bootstrap

A questo punto andiamo a eseguire il Random Forest variando un Iperparametro alla volta trovando qual’è il miglior valore (non la miglior combinazione) attraverso la verifica dell’accuratezza, ottenendo i seguenti risultati:

* Best n\_estimators: 8
* Best max\_features: 1
* Best min\_samples\_leaf: 1
* Best max\_depth: 50
* Best bootstrap: False

L’accuratezza dell’algoritmo Cross Validation migliora a 80,42%

Normalizzazione

Standardizza le funzionalità rimuovendo la media e il ridimensionamento alla varianza dell'unità

Il punteggio standard di un campione x è calcolato come:

z = (x - u) / s

dove u è la media dei campioni di addestramento o zero se with\_mean = False, e s è la deviazione standard dei campioni di addestramento o uno se with\_std = False.

Centratura e ridimensionamento avvengono indipendentemente su ciascuna caratteristica calcolando le statistiche pertinenti sui campioni nel set di training. La media e la deviazione standard vengono quindi memorizzate per essere utilizzate su dati successivi utilizzando il metodo di trasformazione.

La standardizzazione di un set di dati è un requisito comune per molti stimatori di apprendimento automatico: potrebbero comportarsi in modo negativo se le singole caratteristiche non assomigliano più o meno ai dati normalmente distribuiti (ad esempio gaussiana con media 0 e varianza dell'unità).

Ad esempio molti elementi utilizzati nella funzione obiettivo di un algoritmo di apprendimento (come il kernel RBF di Support Vector Machines oi regolatori L1 e L2 di modelli lineari) presumono che tutte le caratteristiche siano centrate attorno a 0 e abbiano varianza nello stesso ordine. Se una caratteristica ha una varianza che è di ordine di grandezza più grande di altre, potrebbe dominare la funzione obiettivo e rendere lo stimatore incapace di apprendere da altre caratteristiche correttamente come previsto.

Questo ridimensionatore può essere applicato anche a matrici CSR o CSC sparse passando con with\_mean = False per evitare di rompere la struttura di sparsità dei dati

Dummy features

**Cos'è una dummy variable?**

Una dummy variable o variabile indicatore è una variabile artificiale creata per rappresentare un attributo con due o più categorie / livelli distinti.

**Perché viene usato?**

L'analisi di regressione considera tutte le variabili indipendenti (X) nell'analisi come numeriche. Le variabili numeriche sono variabili di scala di intervallo o rapporto i cui valori sono direttamente confrontabili, ad es. '10 è il doppio di 5 'o' 3 meno 1 uguale a 2 '. Spesso, tuttavia, è possibile includere nel proprio studio un attributo o una variabile di scala nominale come "Marchio del prodotto" o "Tipo di difetto". Di 'che hai tre tipi di difetti, numerati' 1 ',' 2 'e' 3 '. In questo caso, "3 meno 1" non significa nulla ... non puoi sottrarre il difetto 1 dal difetto 3. I numeri qui sono usati per indicare o identificare i livelli di "Tipo di difetto" e non hanno il significato intrinseco del loro proprio. Le dummy variable vengono create in questa situazione per "ingannare" l'algoritmo di regressione per analizzare correttamente le variabili degli attributi.

**Cose da tenere a mente sulle variabili dummy**

Le dummy variable assegnano i numeri "0" e "1" per indicare l'appartenenza a una categoria reciprocamente esclusiva ed esauriente.

  1. Il numero di dummy variable necessarie per rappresentare una singola variabile di attributo è uguale al numero di livelli (categorie) in quella variabile meno uno.

  2. Per una determinata variabile di attributo, nessuna delle variabili dummy costruite può essere ridondante. Cioè, una variabile dummy non può essere una costante multipla o una semplice relazione lineare di un'altra.

3. L'interazione di due variabili di attributo (ad esempio Gender e Stato civile) è rappresentata da una terza variabile dummy che è semplicemente il prodotto delle due variabili dummy individuali.

Cross Validation

La Random forest, come suggerisce il nome, consiste in un gran numero di alberi decisionali individuali che operano come un insieme. Ogni singolo albero nella foresta casuale sputa una previsione di classe e la classe con il maggior numero di voti diventa la previsione del nostro modello (vedi figura sotto).



Il concetto fondamentale dietro la random forest è semplice ma potente: la saggezza delle folle. Nel campo della scienza dei dati, la ragione per cui il modello di random forest funziona così bene è:

**Un gran numero di modelli relativamente non correlati (alberi) che operano come un comitato supererà qualsiasi dei singoli modelli costituenti.**

La bassa correlazione tra i modelli è la chiave. Proprio come gli investimenti con basse correlazioni (come azioni e obbligazioni) si uniscono per formare un portafoglio che è maggiore della somma delle sue parti, i modelli non correlati possono produrre previsioni di insieme che sono più accurate di qualsiasi previsione individuale. La ragione di questo meraviglioso effetto è che gli alberi si proteggono a vicenda dai loro errori individuali (a patto che non si sbagliano costantemente tutti nella stessa direzione). Mentre alcuni alberi potrebbero sbagliare, molti altri alberi avranno ragione, quindi come gruppo gli alberi sono in grado di muoversi nella direzione corretta. Quindi i prerequisiti per la random forest per ottenere buoni risultati sono:

1. Deve esserci un segnale reale nelle nostre funzionalità in modo che i modelli costruiti usando queste caratteristiche facciano meglio delle ipotesi casuali.

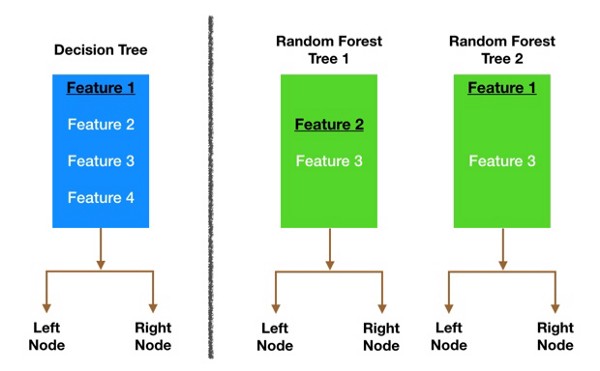
2. Le previsioni (e quindi gli errori) fatte dai singoli alberi devono avere una bassa correlazione l'una con l'altra.

Garantire che i modelli si diversificano a vicenda

In che modo la random forest garantisce che il comportamento di ogni singolo albero non sia troppo correlato al comportamento di uno degli altri alberi nel modello? Utilizza i seguenti due metodi:

Bagging (Bootstrap Aggregation) - Gli alberi delle decisioni sono molto sensibili ai dati sui quali sono addestrati - piccole modifiche al set di allenamento possono portare a strutture dell'albero significativamente diverse. La random forest si avvantaggia di ciò consentendo a ogni singolo albero di campionare casualmente dal set di dati con la sostituzione, risultando in alberi diversi. Questo processo è noto come bagging.

Si noti che con il bagging non stiamo inserendo i dati dell'allenamento in blocchi più piccoli e allenando ciascun albero su un altro blocco. Piuttosto, se abbiamo un campione di dimensione N, stiamo ancora dando da mangiare ad ogni albero un set di allenamento di taglia N (se non diversamente specificato). Ma invece dei dati di allenamento originali, prendiamo un campione casuale di taglia N con la sostituzione.

**Disponibilità casuale** - In un normale albero decisionale, quando è il momento di dividere un nodo, consideriamo ogni possibile caratteristica e selezioniamo quella che produce la maggior separazione tra le osservazioni nel nodo sinistro e quelle nel nodo destro. Al contrario, ogni albero in una random forest può selezionare solo da un sottoinsieme casuale di funzionalità. Ciò impone ancora più variazioni tra gli alberi nel modello e alla fine si traduce in una minore correlazione tra gli alberi e una maggiore diversificazione.

Conclusione

Cos'è un classificatore di random forest?

La random forest è un algoritmo di classificazione costituito da molti alberi di decisioni. Utilizza l'insaccamento e la casualità delle caratteristiche quando costruisce ogni singolo albero per cercare di creare una foresta di alberi non correlata, la cui previsione da parte del comitato è più accurata di quella di ogni singolo albero.

Di cosa abbiamo bisogno per fare in modo che la nostra random forest effettui previsioni di classe accurate?

1. Abbiamo bisogno di caratteristiche che abbiano almeno un po 'di potere predittivo. Dopotutto, se mettiamo la spazzatura allora otterremo la spazzatura.

2. Gli alberi della foresta e, soprattutto, le loro previsioni devono essere non correlate (o almeno avere correlazioni basse tra loro). Mentre l'algoritmo stesso attraverso la casualità delle feature cerca di ingegnerizzare queste basse correlazioni per noi, le caratteristiche che selezioniamo e gli iper-parametri che scegliamo influenzeranno anche le correlazioni finali.

**Come funziona l'algoritmo?**

Funziona in quattro passaggi:

1. Seleziona campioni casuali da un determinato set di dati.

2. Costruire un albero decisionale per ogni campione e ottenere un risultato di predizione da ciascun albero decisionale.

3. Esegui un voto per ogni risultato previsto.

4. Selezionare il risultato della previsione con il maggior numero di voti come previsione finale.

**Vantaggi:**

• Le random forest sono considerate un metodo estremamente preciso e robusto a causa del numero di alberi decisionali che partecipano al processo.

• Non soffre del problema di overfitting. La ragione principale è che ci vuole la media di tutte le previsioni, che annulla i pregiudizi.

• L'algoritmo può essere utilizzato in entrambi i problemi di classificazione e regressione.

• Le radom forest possono anche gestire i valori mancanti. Ci sono due modi per gestirli: usare valori mediani per sostituire le variabili continue e calcolare la media ponderata in prossimità dei valori mancanti.

• È possibile ottenere l'importanza relativa delle funzionalità, il che aiuta a selezionare le funzioni più contributive per il classificatore.

**Svantaggi:**

• Le random forest sono lente nel generare previsioni perché ha più alberi decisionali. Ogni volta che effettua una previsione, tutti gli alberi nella foresta devono fare una previsione per lo stesso input dato e quindi eseguire il voto su di essa. L'intero processo richiede molto tempo.

• Il modello è difficile da interpretare rispetto a un decision tree, in cui è possibile prendere facilmente una decisione seguendo il percorso nell'albero.

Trovare features importance

Le random forest offrono anche un buon indicatore di selezione delle funzioni. Scikit-learn fornisce una variabile extra con il modello, che mostra l'importanza relativa o il contributo di ciascuna feature nella previsione. Calcola automaticamente il punteggio di pertinenza di ciascuna feature nella fase di addestramento. Quindi ridimensiona la rilevanza in modo che la somma di tutti i punteggi sia 1.

Questo punteggio ti aiuterà a scegliere le caratteristiche più importanti e a eliminare quelle meno importanti per la costruzione del modello.

La random forest utilizza l'importanza di gini o la riduzione media nell'impurità (MDI) per calcolare l'importanza di ciascuna funzione. L'importanza di Gini è anche nota come la diminuzione totale dell'impurità dei nodi. Questo è quanto il modello si adatta o la precisione diminuisce quando si rilascia una variabile. Maggiore è la diminuzione, più significativa è la variabile. Qui, la diminuzione media è un parametro significativo per la selezione delle variabili. L'indice Gini può descrivere il potere esplicativo complessivo delle variabili.