Università degli Studi di Padova
Corso di Ingegneria del Software 2019/2020

Predire in Grafana

Monitoraggio predittivo per DevOps

ZUCCHETTI
IL SOFTWARE CHE CREA SUCCESSO

Oggetto dell' appalto

Oggi molte applicazioni vengono erogate come servizi nel Cloud. La Zucchetti, primaria software house italiana, da tempo produce procedure gestionali (ERP, Paghe, Fiscale) in tecnologia Web che vengono installate dai clienti nei cloud pubblici, come Amazon AWS o Azure di Microsoft, nella server farm della Zucchetti stessa o in data center privati.

Dall'inizio di quest'anno tutte le fatture emesse in Italia devono transitare in formato elettronico dal Ministero dell'Economia e delle Finanze.
La Zucchetti ha attivato un servizio di "Fatturazione Elettronica" nel proprio data center. Il servizio è completamente gestito internamente e fa da tramite tra i propri clienti, i quali producono le fatture con i programmi gestionali, e il ministero che le processa e le archivia.
I volumi di traffico sono ingenti, alla fine di Giugno 2019 in tutta Italia erano state emesse 1 Miliardo di fatture e la Zucchetti ne ha processate circa il 10%, con un traffico medio di 500.000 documenti al giorno.

Questo scenario prende il nome di "DevOps", dall'unione delle parole "Development" e "Operations": la fabbrica del software e gli operatori che erogano il servizio sono a stretto contatto tra loro e la costante collaborazione tra i due attori porta ad un notevole miglioramento della qualità.
Perché la collaborazione sia efficace è necessario un pieno monitoraggio dei sistemi, in modo che gli erogatori del servizio possano segnalare alla fabbrica i punti in cui è possibile migliorare la procedura. Nella modalità tradizionale invece i sistemisti si preoccupano solo dello stato del sistema, dovendo garantire la continuità del servizio, non del suo costante miglioramento.
La linea di produzione del software può intervenire con cognizione di causa migliorando i punti critici che l'erogazione mette in evidenza. Ne consegue che i prodotti seguiti in "DevOps" hanno molti più rilasci, vicini nel tempo, e risultano di qualità superiore.

Per eseguire il monitoraggio dei propri sistemi la Zucchetti ha scelto Grafana, un prodotto Open Source, individuato come il miglior sistema di monitoraggio. Grafana è estendibile con plug-in in linguaggio JavaScript.

Gli elementi oggi presenti in Grafana sono sistemi di presentazione del dato, raccolti in Dashboard operative e sistemi di allarme al raggiungimento di determinate soglie.
Per dispiegare i vantaggi del "DevOps" la Zucchetti vorrebbe realizzare dei sistemi che possano effettuare delle previsioni al flusso dei dati raccolti, al fine non solo di monitorare la "liveliness" del sistema ma anche per consigliare gli interventi o quanto meno le zone di intervento alla linea di produzione del software.
Le previsioni che desideriamo ottenere possono essere di due tipi: "classificazioni" per stimare il gruppo di appartenenza degli eventi dai dati "predittori" o "regressioni" nel caso in cui il valore cercato sia numerico e con campo continuo. A questo fine le tecniche che proponiamo sono le Support Vector Machine per la classificazione e la Regressione Lineare per la previsione di valori numerici.

Il presente capitolato ha per oggetto l'affidamento della fornitura per la realizzazione di un plug-in per lo strumento di monitoraggio Grafana che applichi SVM e Regressione Lineare al flusso dei dati ricevuti per allarmi o segnalazioni tra gli operatori del servizio Cloud e la linea di produzione del software.

Caratteristiche e Requisiti Obbligatori

Il sistema che richiediamo saranno due plug-in di Grafana, scritti in linguaggio JavaScript, che leggeranno da un file json la definizione dei calcoli da applicare (SVM o RL) e quindi permetteranno di associarli ad alcuni nodi della rete del flusso del monitoraggio.
I plug-in quindi eseguiranno i calcoli previsti, producendo dei valori che potranno essere aggiunti al flusso del monitoraggio come se fossero stati rilevati dal campo.

L'addestramento delle SVM e della RL devono essere fatti in una applicazione apposita a cui verranno forniti i dati di test.
Alternativamente, se possibile, può essere fatta direttamente in Grafana quando non sono necessari dati aggiuntivi per l'addestramento, come nel caso della RL o di una seconda classificazione che si affianchi ad un dato di classificazione già presente nei dati osservati.

Il software richiesto dovrà svolgere almeno i seguenti compiti:

1. Produrre un file json dai dati di addestramento con i parametri per le previsioni con Support Vector Machine (SVM) per le classificazioni o la Regressione Lineare (RL o LR da Linear Regression in inglese)
2. Leggere la definizione del predittore dal file in formato json.
3. Associare i predittori letti dal file json al flusso di dati presente in Grafana.
4. Applicare la previsione e fornire i nuovi dati ottenuti dalla previsione al sistema di Grafana.
5. Rendere disponibili i dati al sistema di creazione di grafici e dashboard per la loro visualizzazione.

Requisiti Opzionali

Il software potrebbe avere le seguenti caratteristiche:

1. Possibilità di definire “alert” in base a livelli di soglia raggiunti dai nodi collegati alle previsioni.
2. Fornire i dati di bontà dei modelli di previsione. “Precision” e “Recall” per le SVM e “R^2” per la Regressione Lineare.
3. Possibilità di applicare delle trasformazioni alle misure lette dal campo per ottenere delle regressioni esponenziali o logaritmiche e non solo lineari.
4. Possibilità di addestrare la Support Vector Machine o la Regressione Lineare direttamente in Grafana.
5. Implementare dei meccanismi di apprendimento di flusso, in modo da poter disporre di sistemi di previsione in costante adattamento ai dati rilevati sul campo.
6. Utilizzare anche altri metodi di previsione, tra cui la versione delle SVM adattate alla Regressione o piccole Reti Neurali per la classificazione.

Suggerimenti

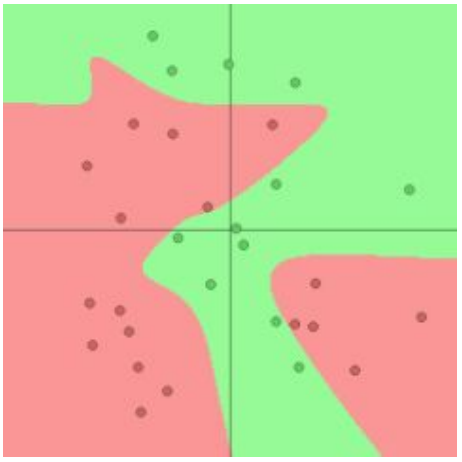
L’azienda proponente chiede al team di realizzare i plug-in di Grafana in linguaggio Javascript ed è ampiamente disponibile a fornire la formazione sugli algoritmi di Machine Learning, che non fanno parte del corso di studi della laurea triennale, e librerie che realizzano tali algoritmi.

Support Vector Machines

L’algoritmo di classificazione delle SVM è nato dalla ricerca di Vladimir Vapnik per risolvere la “maledizione della dimensionalità”. L’adozione dei computer aveva creato la speranza di poter utilizzare i metodi tradizionali della statistica parametrica su grandi volumi di dati, con molte variabili osservate, ottenendo delle previsioni più precise perché più informate.
Purtroppo questa speranza si è scontrata con l’effetto di “diradamento” dei dati osservati man mano che vengono aggiunte dimensioni attraverso la considerazione di più variabili.
Banalmente 100 punti osservati con due predittori si devono disporre su una superficie piana in due dimensioni, 100 punti osservati con tre predittori occupano il volume di un cubo tridimensionale e l’effetto è che sono più staccati tra loro perché hanno più “spazio” che li può ospitare.
Aggiungendo tante variabili ai predittori si creano spazi con molte dimensioni (100 predittori genera uno spazio di 100 dimensioni) e i punti osservati diventano sempre più punti isolati che non aiutano ad ottenere previsioni informate.

Le SVM cercano l’iperpiano che divide meglio i dati osservati in due classi, in pratica cercano la linea che separa meglio i punti di una superficie piana, il piano che taglia meglio il cubo e così via. In questo modo riesce a resistere bene anche all’aggiunta di dimensioni ed al diradarsi dei punti nello spazio corrispondente.

Poiché il monitoraggio può avvenire su tanti dati raccolti dal campo (es: CPU, memoria, utenti, query, tempi di rete,...) è importante fornire un algoritmo che non perda validità anche se viene associato a molti predittori.
E' importante considerare che tanti predittori danno la possibilità al sistema di adattarsi ai dati di addestramento, l'effetto di "overfitting", per cui le previsioni sono molto buone sui dati noti e molto meno valide sui dati che non sono stati incontrati durante la fase di addestramento. Vapnik è riuscito a legare la probabilità dell'errore di previsione al numero di dati dell'insieme di test e ai gradi di libertà determinati dal numero di predittori e dalla struttura dell'algoritmo di previsione (VC dimension).



Addestramento delle SVM

Per ottenere una previsione da una SVM per prima cosa bisogna "addestrarla" con dei dati noti. In particolare se la classe che si vuole prevedere (es: condizione normale / allarme) non è presente nei dati raccolti deve essere aggiunta a mano secondo il giudizio di chi prepara il sistema di previsione.
Il percorso è quindi raccogliere dei dati dal campo, valutare di aver coperto in modo esaustivo i casi attesi, associare ad ogni osservazione la classe a cui appartiene e quindi addestrare la SVM con questi dati.
Al termine dell'addestramento si ottiene una serie di parametri della SVM che saranno quelli da utilizzare per eseguire le previsioni.

Nelle SVM lineari il numero di parametri è il numero dei predittori e la previsione viene eseguita con una moltiplicazione ed una somma per ogni predittore, quindi con tempi estremamente veloci compatibili con il monitoraggio in tempo reale.

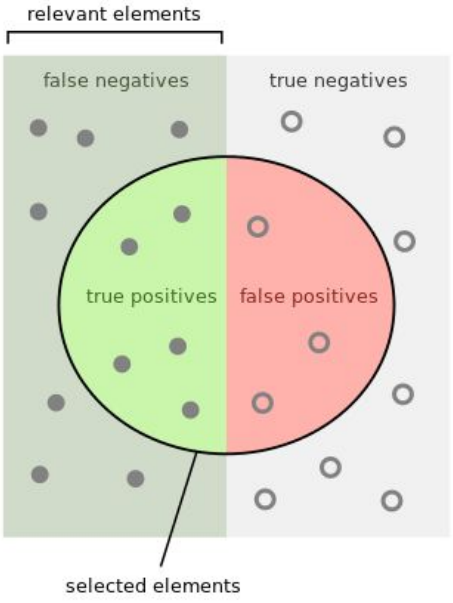
Nelle SVM con Kernel le osservazioni vengono prima trasformate con una opportuna funzione e quindi viene trovato il vettore di supporto nel nuovo spazio. In questo caso i calcoli per la previsione non dipende più dal numero di predittori ma dal numero di elementi del vettore di supporto. Se il vettore di supporto è di grandi dimensioni potrebbe non essere compatibile con il calcolo in tempo reale.

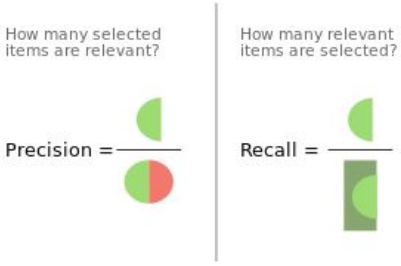
Per ottenere delle SVM con capacità di separare classi anche con forme diverse da un piano si può ricorrere alla possibilità di aggiungere dimensioni derivate da quelle dei predittori, utilizzando somme e moltiplicazioni tra i predittori stessi. In questo modo i calcoli restano molto contenuti e la capacità previsionale migliora per le forme considerate.

Una buona pagina introduttiva sulle SVM è lorenzogovoni, oppure la pagina SVM.

Validazione del modello predittivo delle SVM

A seguito dell'addestramento è importante disporre di una misura della bontà del sistema creato. A questo fine i dati di addestramento vengono di solito separati in due gruppi, con uno si addestra la SVM e poi si esegue la previsione sul secondo gruppo, controllando se la previsione coincide con il dato presente nell'insieme di test.





Per la classificazione le misure più note sono "Precision" e "Recall".

La misura "Precision" è il rapporto tra i "Veri Positivi" e "Veri Positivi + Falsi Positivi", cioè tutti quelli che la SVM ha considerato veri, anche se qualcuno non lo era.

La misura "Recall" è il rapporto tra i "Veri Positivi" trovati dal sistema e tutti i "Positivi" che esistono nei dati. Spesso il sistema non riesce a classificare come vero un dato che invece lo è.

Di solito modificando i parametri di addestramento si modificano queste due misure, ma migliorando una si peggiora l'altra. La cosa è evidente considerando il caso estremo della macchina attribuisce ad ogni punto sempre della stessa classe: avrà un "Recall" di 1, ma una "Precision" molto bassa perché di fatto sbaglia tutti i punti dell'altra classe.

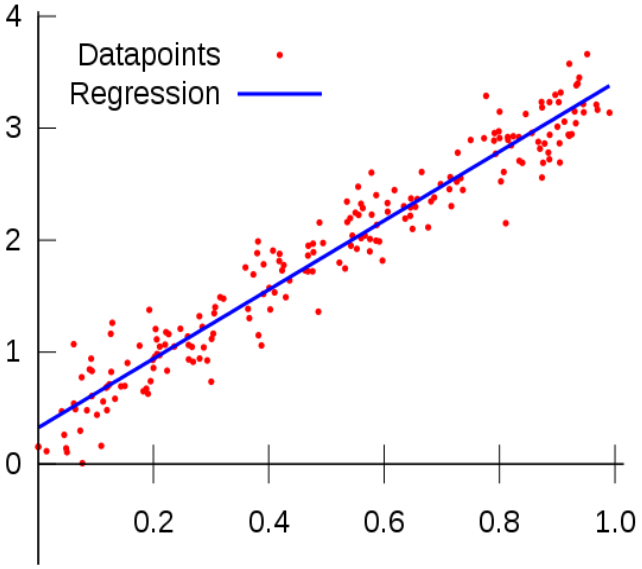
Per poter confrontare due sistemi tenendo conto di queste due misure normalmente viene create la F-Measure che è data da:

$$F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Un buon riferimento per comprendere le varie misure di bontà di un sistema di classificazione è la pagina Precision and Recall.

Regressione Lineare

La Regressione Lineare è uno dei metodi più antichi per attribuire un valore numerico come stima di una osservazione partendo da dei valori di riferimento. Il metodo dei “minimi quadrati”, che è tuttora tra i più utilizzati, è stato pubblicato da Legendre nel 1805 e reinventato indipendentemente da Gauss nel 1809, che però sostenne di averlo sviluppato nel 1795.



La Regressione Lineare, come dice il nome, immagina che la legge sottostante i dati osservati sia esprimibile con una retta. Ogni punto osservato viene posto in un sistema per determinare i coefficienti della retta, poiché sarebbe un sistema sovrastimato appena si hanno più di due punti (e quindi non risolvibile), viene considerata la somma del quadrato di di tutte le differenze tra i valori trovati e i valori stimati. Minimizzando questa somma si trova la retta migliore per approssimare i dati.

La pagina Regressione Lineare è una introduzione all’argomento, mentre la pagina Least Square spiega come valutare i risultati. Un testo completo sulla regressione è Ricci - Principali tecniche di Regressione con R

Addestramento della Regressione Lineare

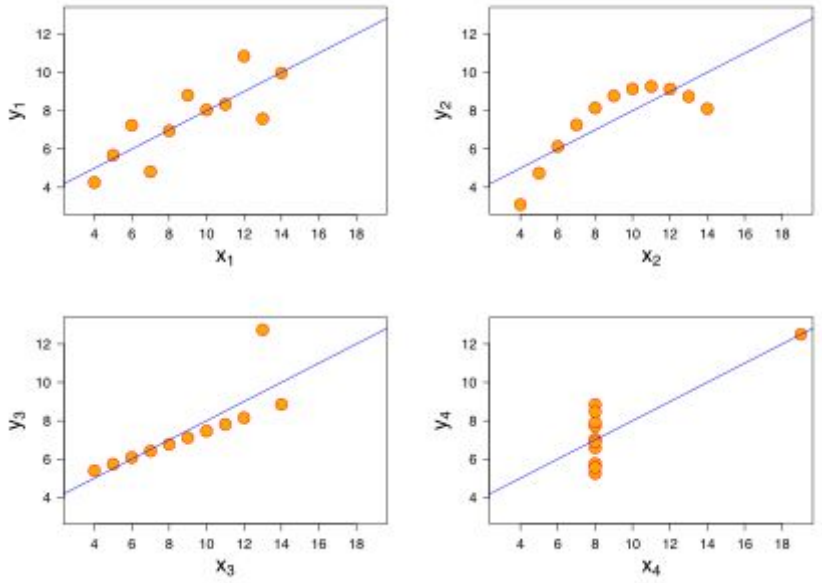
Anche nel caso della Regressione Lineare si devono avere dei dati di addestramento, cioè dei dati con cui si identificano i parametri della retta che servirà per le previsioni.

Mentre con la classificazione dei dati di monitoraggio di solito i dati raccolti non dispongono del valore della classe, che quindi deve essere attribuita a mano, i dati per la regressione hanno anche il valore di riferimento e la previsione serve più a valutare lo scostamento del dato rilevato dal valore

atteso che non il valore numerico di un dato non noto. Comunque se non fosse così anche in questo caso chi addestra deve fornire i valori attesi che non sono noti.

Se la legge sottostante non è una retta è possibile applicare delle trasformazioni per adattare ai dati altre forme, come parabole, esponenziali o logaritmi.
In particolare quando i fenomeni sottostanti sono moltiplicativi e non addittivi, una trasformazione logaritmica fornisce previsioni molto più accurate.

Non tutti i dati sono adatti alla analisi con la Regressione Lineare. Il famosissimo “Quartetto di Anscombe” propone quattro disposizioni dei dati con le stesse caratteristiche dal punto di vista statistico, ma con un risultato completamente diverso dal punto di vista della previsione eseguita con i metodi lineari.



La pagina Quartetto di Anscombe contiene informazioni sull’argomento.

Validazione del modello di Regressione Lineare

Proprio per la sua età i metodi per validare il modello che nasce dalla Regressione Lineare sono molto ricchi e sofisticati. Ad esempio se i residui, cioè la differenza tra il valore reale ed il valore previsto, non sono distribuiti come una curva Gaussiana, ci sono forti indizi che debba essere applicata una trasformazione ai predittori.

Vista la complessità di queste valutazioni prenderemo in considerazione solo l’indicatore principale: R^2.
Questo indicatore viene calcolato sottraendo a 1 il rapporto la la somma dei quadrati dei residui e la somma dei quadrati dei residui rispetto alla media dei valori osservati.
In pratica questo numero da il rapporto tra quanto bene prevede la linea trovata, che usa i predittori, e la semplice media dei valori osservati, che è il sistema più semplice e che non usa i predittori.

Un valore vicino ad 1 di questa misura indica una ottima previsione, spostandosi verso lo 0 la regressione lineare è poco più significativa della media e i predittori non forniscono di fatto alcuna informazione. Se diventa negativo vuol dire che la linea trovata è addirittura controproducente per l'attività di previsione.

Algoritmi di flusso

I sistemi fin qui descritti prevedono la fase di addestramento separata da quella di previsione. Spesso il flusso di dati contiene sia i predittori che il valore da prevedere, in questo caso questi sistemi possono essere utilizzati per identificare eventi con valori molto diversi da quelli attesi (outliers).

Ma in questo scenario è interessante allora poter svolgere un addestramento continuo dei sistemi mentre ricevono i dati stessi, in modo da poter adattare le previsioni al mutare delle condizioni.

Sia per le SVM che per la RL è possibile avere algoritmi di addestramento sul flusso, cioè algoritmi in grado di modificare i parametri del sistema aggiungendo un punto al sistema già addestrato.

Per il monitoraggio questa possibilità è molto interessante, perché permette di saltare la fase di addestramento, anche se la complessità diventa molto più elevata e la valutazione della bontà dei risultati è più problematica.

Applicazioni e Librerie

Una buona libreria in Javascript per le Support Vector Machine è disponibile all'indirizzo https://github.com/karpathy/svmjs
Questa libreria è stata ampiamente modificata da uno studente nel corso di uno stage presso la Zucchetti e può essere richiesta al committente.

La regressione lineare (ed altre forme di regressione) è ampiamente disponibile in Javascript in vari siti, uno è https://github.com/Tom-Alexander/regression-js.
Il committente dispone di una libreria in grado di modificare i parametri di regressione mentre riceve il flusso dei dati ed è disponibile a fornirla ai gruppi che realizzeranno il progetto proposto.

Le reti neurali in Javascrpt possono essere realizzate con la libreria https://cs.stanford.edu/people/karpathy/convnetjs/ dello stesso autore della libreria consigliata per le Support Vector Machine.

Il prodotto di monitoraggio Grafana è reperibile all'indirizzo: https://grafana.com/

Uno strumento di analisi dei dati che l'azienda consiglia per comprendere sia le SVM che la RL è Orange Canvas.

Variazioni ai requisiti

In corso d'opera non sarà possibile variare/modificare i requisiti minimi (obbligatori per accettare il prodotto). Sarà invece possibile variare i requisiti opzionali, in quanto saranno i gruppi vincitori dell'appalto a modificarli / eliminarli / aggiungerli.

Documentazione

Il progetto dovrà essere supportato dalla documentazione minima richiesta per il corso di Ingegneria del software e dovrà essere fornito un manuale per l'utilizzo ed un manuale per chiunque voglia estendere l'applicazione.

Garanzia e Manutenzione

L'azienda Zucchetti SPA è interessata a questo progetto come dimostrazione della fattibilità dell'obiettivo utilizzando le tecnologie web. Costituirà titolo preferenziale nella valutazione delle proposte la pubblicazione del progetto sul sito “github.com” o altri repository pubblici, in conformità con i relativi requisiti di natura open-source, per favorire la continuità del prodotto risultante.

Rinvio

Per tutto quanto non previsto nel presente capitolato, sono applicabili le disposizioni contenute nelle leggi e nei collegati per la gestione degli appalti pubblici.