

Report task di tesi

Riconoscimento delle settimane anomale mediante una finestra
statica

Leonardo Poggiani

October 2, 2020

Contents

1	Motivazione	1
2	Problema	2
3	Risultati	3
4	Conclusioni	9

Abstract

Nel task proposto veniva richiesto di giustapporre i dati forniti con indicazione giornaliera al fine di ottenere un unico array come dataset. Inoltre veniva richiesto di individuare le settimane anomale, caratterizzate dall'avere almeno un giorno con il valore di "Anomalous" settato a 1.

Chapter 1

Motivazione

Al fine di rilevare la presenza di settimane anomale, definite come settimane che hanno almeno un giorno in cui si è rilevato un traffico anomalo, può essere utile pensare di vettorizzare i dati per poi analizzare i valori di Anomalous cos ricavati.

Questo può essere particolarmente utile perchè i dati così aggregati non sono molto utilizzabili.

Si rende quindi necessaria un'operazione di preprocessing molto semplice, che consiste nel portare i dati in un formato standard e più leggibile.

Chapter 2

Problema

Il problema risiede nella natura dei dati di cui disponiamo. Il dataset iniziale, infatti, è in un formato che prevede una riga per giorno dell'anno (tutti relativi all'anno 2015) e per ogni giorno una serie di colonne. Queste colonne contengono delle informazioni scomposte, come la data (espansa in 2 colonne, giorno e mese), il valore che indica il valore di "Anomalous" e infine una colonna per ogni fascia oraria (da h1 a h23). I dati così composti non possono essere usati in modo efficace per il nostro scopo.

	A	B	C	D	E	F
1	Anomalous	Cluster	Day	Month	h1	h2
2	1	0	1	1	1	0
3	0	1	2	1	0.174002633779448	0.181794500058363
4	0	1	3	1	0.440282061270347	0.244073007665684
5	0	2	4	1	0.52200292633957	0.314627518815994
6	0	0	5	1	0.0111238823171337	0.0104769275911869
7	0	0	6	1	0.180894763647734	0.179121350642964
8	0	0	7	1	0.0132592658560402	0.0117340681323592

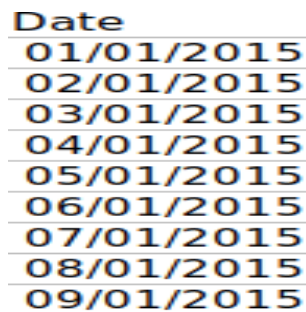
Figure 2.1: Formato della colonna "Data"

Chapter 3

Risultati

Per raggiungere il doppio obiettivo di individuare le settimane anomale e rendere più manipolabile il dataset fornito dobbiamo innanzitutto procedere ad aggregare il dato relativo alla data.

Ho pensato infatti che fosse più opportuno ricongiungere i dati inizialmente separati in un unico campo "Data" che farà da indice al dataset in seguito.



Date
01/01/2015
02/01/2015
03/01/2015
04/01/2015
05/01/2015
06/01/2015
07/01/2015
08/01/2015
09/01/2015

Figure 3.1: Colonna relativa alla data

La colonna viene riempita scorrendo le tuple del dataset e creando un oggetto di tipo *datetime* passando come argomenti "2015", il campo *Month* e *Day*. Fatto questo si passa a scomporre il dato relativo alla riga in varie tuple, precisamente una per ogni dato relativo all'affluenza oraria. In questo modo si avrebbe un dataset composto dal *timestamp* della rilevazione e il relativo valore di affluenza, formato abbastanza standard per i dataset.

Questa scomposizione viene fatta giustapponendo le colonne in verticale, iterando sulle tuple e creando un nuovo dataframe indicizzato con il valore del campo *Data*.

Data	Affluenza
2015-01-01 01:00:00	1.0
2015-01-01 02:00:00	0.0
2015-01-01 03:00:00	0.29989084300651303
2015-01-01 04:00:00	0.195044014019326
2015-01-01 05:00:00	0.198812234343099
2015-01-01 06:00:00	0.194608629371612
2015-01-01 07:00:00	0.195873221700534
2015-01-01 08:00:00	0.0
2015-01-01 09:00:00	0.5836151837018779
2015-01-01 10:00:00	0.653947165851926
2015-01-01 11:00:00	0.659252614296578
2015-01-01 12:00:00	0.658227682832466
2015-01-01 13:00:00	0.657949626242911
2015-01-01 14:00:00	0.6581539727875649
2015-01-01 15:00:00	0.659845219149165
2015-01-01 16:00:00	0.653915703787933
2015-01-01 17:00:00	0.659571343055467
2015-01-01 18:00:00	0.662061245988125
2015-01-01 19:00:00	0.662021517169433
2015-01-01 20:00:00	0.623036198460715
2015-01-01 21:00:00	0.660963150445381
2015-01-01 22:00:00	0.330982539053282
2015-01-01 23:00:00	0.23114175879834697

Figure 3.2: Dataset nel formato prodotto

Per comodità viene anche salvato un dataframe uguale al precedente a cui stata giustapposta una colonna ridondante che rappresenta il valore di "Anomalous" per ogni ora del giorno. Viene ripetuto per comodità per ogni ora del giorno definito come *anomalo*.

Vengono poi normalizzati i dati attraverso il *MinMaxScaler* per farne successivamente il *fit* e addestrare la rete *LSTM* già vista nei task precedenti. Questo viene fatto per mostrare come saranno visualizzati i dati avendo portato in dataset nella forma mostrata sopra.

Come si può ricavare dal grafico 3.4, la computazione risulta avere una precisione abbastanza buona. Si sono è predetto l'andamento dell'affluenza di 24 ore scelte in mode casuale all'interno del dataset.

La curva ottenuta (rosso) non si discosta molto dall'originale (blu) riuscendo anche a prevedere con una buona precisione un picco anomalo.

Per ricavare i dati che sono stati plottati in figura 3.4 si è usata una rete neurale LSTM con un settaggio abbastanza standard, che prevede un numero di *epoch* pari a 100 e una *batch size* pari a 32. Dato che la computazione con un numero così elevato di dati inizia ad essere abbastanza *time consuming* e poichè questo non era il focus del task, non sono state provate molte configurazioni diverse. È stata mostrata solo la configurazione riportata per fornire una prova di come adesso si presenta il dataset.

Infatti la parte principale del task era automatizzare il processo di riconosci-

2015-01-01 10:00:00	0.653947165851926	1
2015-01-01 11:00:00	0.659252614296578	1
2015-01-01 12:00:00	0.658227682832466	1
2015-01-01 13:00:00	0.657949626242911	1
2015-01-01 14:00:00	0.6581539727875649	1
2015-01-01 15:00:00	0.659845219149165	1
2015-01-01 16:00:00	0.653915703787933	1
2015-01-01 17:00:00	0.659571343055467	1
2015-01-01 18:00:00	0.662061245988125	1
2015-01-01 19:00:00	0.662021517169433	1
2015-01-01 20:00:00	0.623036198460715	1
2015-01-01 21:00:00	0.660963150445381	1
2015-01-01 22:00:00	0.330982539053282	1
2015-01-01 23:00:00	0.23114175879834697	1
2015-01-02 01:00:00	0.17400263377944802	0
2015-01-02 02:00:00	0.181794500058363	0
2015-01-02 03:00:00	0.17888933563972897	0
2015-01-02 04:00:00	0.155451962389229	0
2015-01-02 05:00:00	0.287481127848009	0
2015-01-02 06:00:00	0.0	0

Figure 3.3: Dataset con la colonna "Anomalous"

mento delle settimane anomale.

Questo è stato fatto creando inizialmente un dataframe composto da due colonne, una relativa al numero della settimana e un campo Anomalous, con lo stesso significato del campo Anomalous del dataset precedente ma riferito alla settimana anzichè al giorno.

Dopo si è fatta un'iterazione del dataset vettorizzato creato in precedenza, operazione che potrebbe essere resa più efficiente in quanto si cicla su un insieme di dati dell'ordine delle migliaia di righe.

Dopo aver recuperato il dato relativo alla data, si considera il primo lunedì presente nel dataset. Infatti è dalla prima settimana **completa** che parte la nostra analisi.

Avendo mantenuto il dato ridondante (ripetuto il valore di Anomalous per ogni ora di un giorno anomalo) si può considerare anche solo la prima ora del giorno. A questo punto viene creata la window statica di 7 giorni su cui verrà effettuato un ciclo, controllando di volta in volta se il valore di Anomalous è settato oppure no. Se il valore di Anomalous settato viene settato anche il corrispondente valore nel dataframe di appoggio, altrimenti viene lasciato invariato.

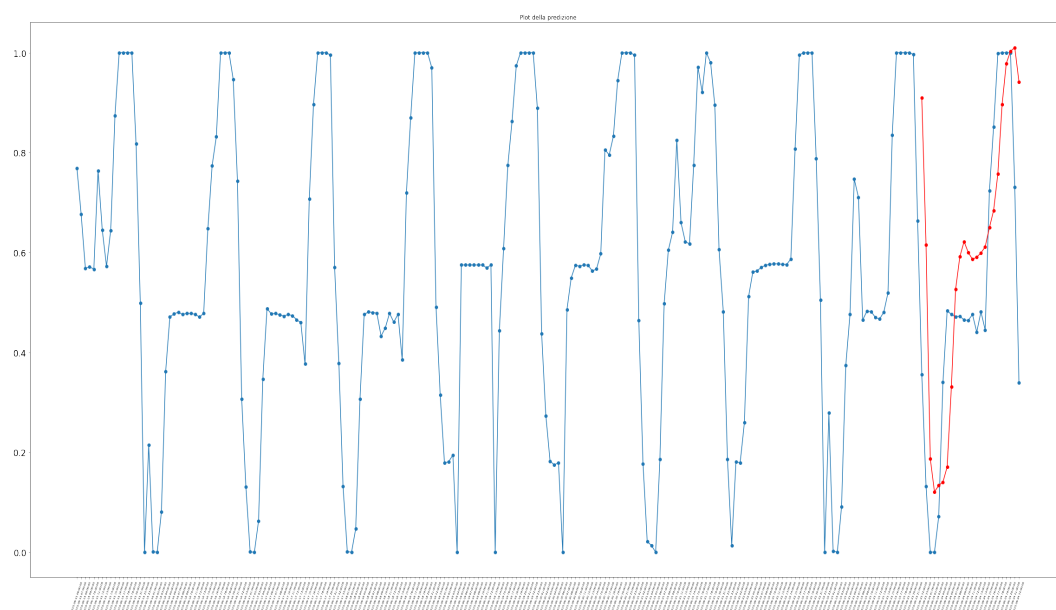


Figure 3.4: Plot del dataset nel nuovo formato. In blu i dati di train, in rosso la predizione.

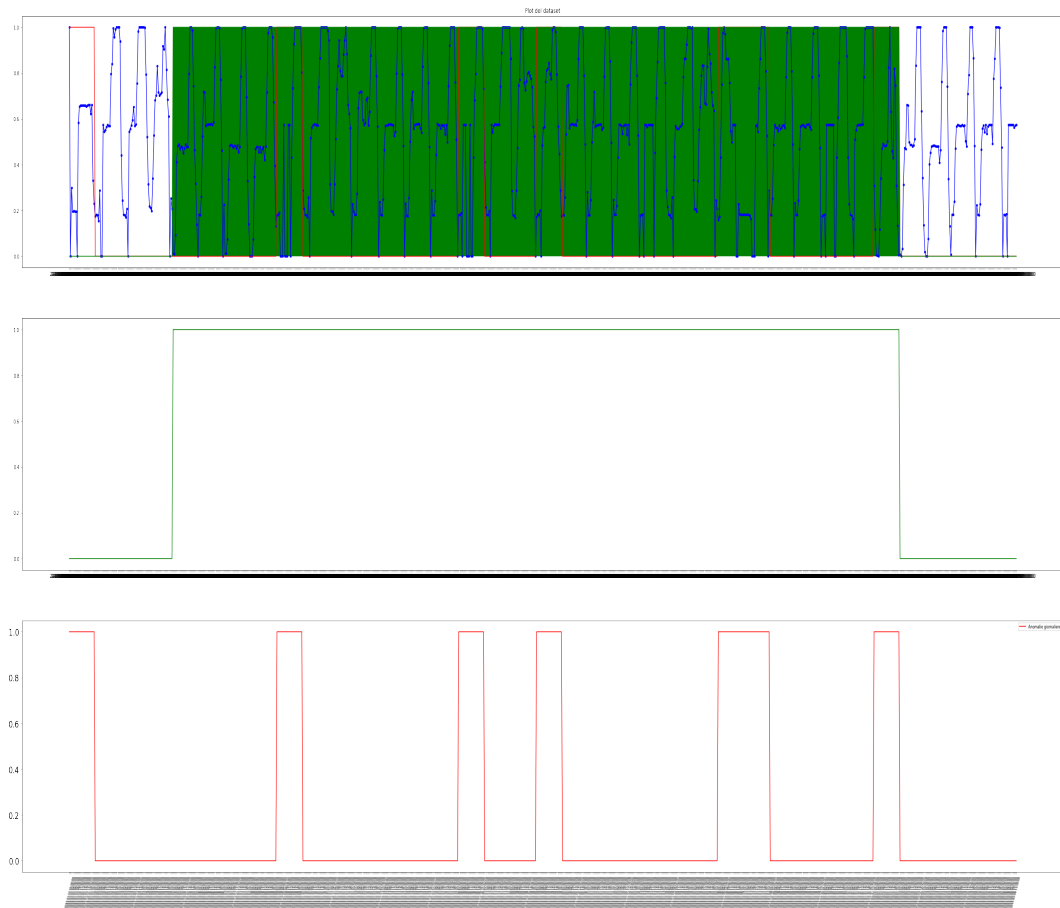


Figure 3.5: Nella prima figura in verde l'indicazione la settimana anomala, in blue la serie temporale normalizzata e in rosso le anomalie giornaliere. Nella seconda figura solo l'indicazione di settimana anomala e nell'ultima le anomalie giornaliere. Da notare che la prima anomalia non è stata considerata in quanto non facente parte di una settimana intera.

Settimana	Anomalous
0	0
1	0
2	1
3	1
4	1
5	1
6	0
7	0
8	1
9	1

Figure 3.6: Dataset risultante.

Chapter 4

Conclusioni

L'attività si è dimostrata impegnativa soprattutto per quanto riguarda la creazione della finestra di 7 giorni e lo scorrimento del dataset. Questo però è stato reso più semplice dal preprocessing fatto inizialmente.

Infatti l'operazione sarebbe stata molto difficile utilizzando i dati come erano stati presentati inizialmente e questo fa capire l'importanza di fare un buon preprocessing dei dati anzichè usarli così come sono stati raccolti.