Report task di tesi

Leonardo Poggiani

October 22, 2020

Abstract

In questo task veniva richiesto di confrontare visivamente una giornata anomala e una giornata predetta attraverso le giornate normali che la precedono. Inoltre viene anche individuata la *cosine distance* che servirà a predire una giornata anomala senza avvalersi del confronto visivo.

1 Motivazioni

Il task si articolava in quattro fasi distinte:

- Preparazione del dato
- Salvataggio del dato
- Recupero del dato e predizione
- Refactoring del codice e creazione di classi distinte

La prima fase è resa necessaria che il dataset su cui predire le giornate erano in forma eterogenea e non utilizzabili per effettuare la predizione. Infatti il formato su cui effettuare la predizione era del tipo [serie di giorni normali, giorno anomalo].
Quindi era necessario un unico giorno anomalo per ogni *slice* di dati e questo deve essere preceduto da tutti i giorni normali consecutivi fino al giorno anomalo precedente.

La riorganizzazione del dataset permetterà una predizione più efficiente in quanto basterà passare alla classe che si occupa della predizione il dataset target della predizione, che conterrà tutti e solamente i dati necessari alla predizione.
Il salvataggio del dato è un altro punto problematico, perchè viene reso necessario dall'esigenza di poterlo esportare in un formato recuperabile senza perdita di informazione da parte del predittore.
Altrettanto importante darà la riorganizzazione del codice volta alla creazione di varie classi che si occuperanno di generare i risultati voluti dopo aver ricevuto come input i dati nel nuovo formato.

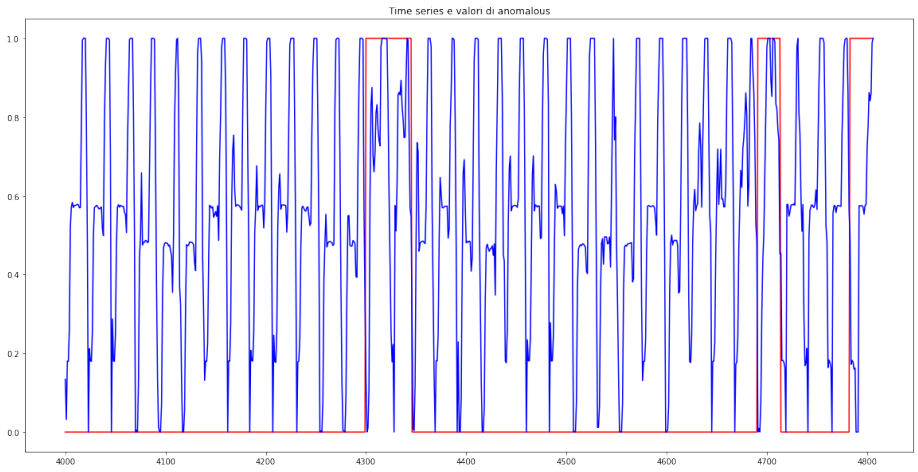

Figure 1: Come veniva raffigurato il dataset in precedenza.

2 Problema

Il problema in questione riguarda la struttura del dataset iniziale. Infatti la predizione che vogliamo andare a fare richiede un formato specifico che si può ottenere raggruppando un giorno anomalo e tutti i giorni normali precedenti a dato giorno anomalo.
Tutto questo viene fatto nell'ottica di scrivere una classe (*data prediction*) dotata di due metodi principali che si occuperanno di predire l'ultima giornata normale all'interno del dataset di riferimento per confrontarla con la time-series reale della stessa giornata, questo al fine di mostrare la bontà delle predizioni effettuate dalla rete, e di predire l'andamento dell'ultima giornata del dataset (quella anomala) a partire dalle precedenti giornate normali, per mostrare le differenze tra la time-series di una giornata normale (quella predetta) e una anomala (quella reale).
Dopo aver codificato tutte le richieste riportate, si dovrà collaudare il sistema per essere sicuri che mantenga un certo standard di affidabilità, misurabile attraverso la funzione *score*.

3 Risultati

Per risolvere i problemi sopra riportati, per prima cosa si è data importanza alla riorganizzazione del dato.
Inizialmente si è aggiunta una colonna al dataframe originale denominata "*Finestra*" che ha il compito di indicare il numero di righe appartenenti a giorni normali che precedono un giorno anomalo. Incidentalmente questa informazione fornisce anche l'indicazione dei giorni normali (consecutivi) che precedono dato giorno

anomalo in maniera molto semplice:

$$giorninormaliconsecutiviprecedenti = \frac{numerodirighe}{23}$$

	Index	Affluenza	Anomalous	Finestra
0	1	1.0	1	0
1	1	0.0	1	0
2	1	0.29989084300651303	1	0
3	1	0.195044014019326	1	0
4	1	0.198812234343099	1	0
5	1	0.194608629371612	1	0
6	1	0.195873221700534	1	0
7	1	0.0	1	0
8	1	0.5836151837018779	1	0
9	1	0.653947165851926	1	0

Figure 2: Colonna finestra.

	Index	Affluenza	Anomalous	Finestra
206	10	0.0	1	184
1862	82	0.37736370310840794	1	1265
3242	142	0.343491640559844	1	1357

Figure 3: Valori tipici della colonna finestra.

Questa colonna è quindi giustapposta al dataframe originale e si rivelerà molto utile per evitare di dover scorrere ogni volta il dataset, potendo agire direttamente sugli indici per filtrare le entrate che ci interessano sfruttando quindi le operazioni predefinite di *Pandas*, molto più efficienti dell'andare a definire operazioni da zero.
Dopo questo viene fatta un'iterazione sul dataset scartando eventuali giorni anomali consecutivi, per i quali si terrà in considerazione solo il primo giorno dei due. Durante l'iterazione vengono salvati in file .csv separati con nomi ricavati da un intero incrementale.
In questo modo ogni file contiene un dataframe nel formato che verrà usato per la predizione.

In seguito il codice è stato riorganizzato per renderlo più chiaro e per racchiudere ogni funzionalità distinta del generico "Predittore" in una classe con metodi e attributi.
Sono state quindi create 4 classi fondamentali con responsabilità diverse e ben definite:

anomalyDetector: Contiene il metodo che si interfaccia con l'utente, chiedendo quale è il path del dato da analizzare e il *max iter* che si vuole fornire al clas-

sificatore in seguito. Dopo aver ottenuto i dati che gli servono avvia il processo di predizione in automatico, richiamando tutte le altre classi definite.

dataPreprocessing: Prende i dati al path specificato dall'utente e li prepara, formattandoli nel modo stabilito per renderli utilizzabili dal Predittore. Salva quindi ogni dataframe portato nel formato corretto in un file separato.

hotspotFlowPrediction: Predice le time series partendo dai dati forniti dalla classe *data preparation*, mostrando graficamente le differenze tra le serie temporali e calcolandone la *cosine distance*-

dataClassifier: Attraverso le *cosine distance* calcolate, classifica i giorni in anomali o non anomali, fornendo come output il vettore dei giorni predetti e lo score del sistema.

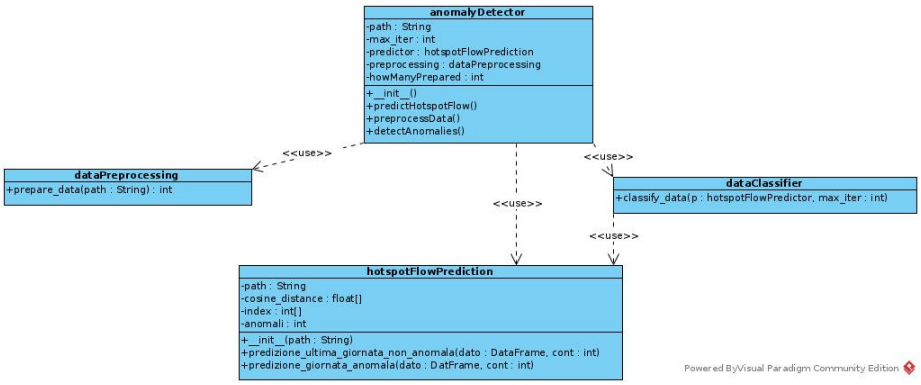

Figure 4: Diagramma delle classi.

A questo punto una semplice esecuzione del sistema prevede la creazione di un'istanza dell'interfaccia, che si occuperà di chiedere all'utente il path dei dati e il *max iter* da usare.

Durante la classificazione viene usata una soglia, ovvero come valore sotto il quale i dati non vengono considerati di interesse, pari a 10 giorni. Questa si riferisce alla colonna "Finestra", che è stata aggiunta in fase di preparazione dei dati e che rappresenta i giorni normali che precedono un dato giorno anomalo. Il valore 10 è stato scelto poichè non molto distante dalla media di giorni normali che compongono la finestra, pari a 22, e perchè garantisce di poter analizzare almeno 6 giorni anomali su 9.
Di seguito sono riportate le prove con valore di soglia pari a 22 e valore di soglia pari a 10.

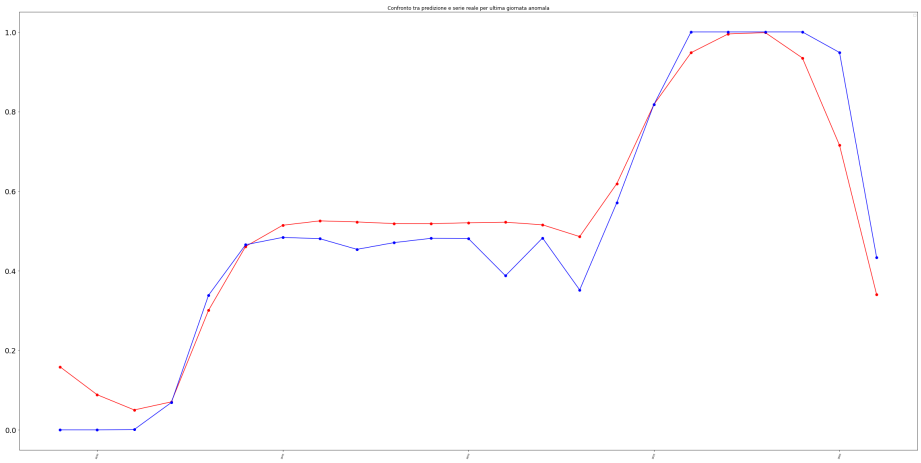

Figure 5: Soglia a 22, primo dataset: confronto tra giornata predetta e giornata normale

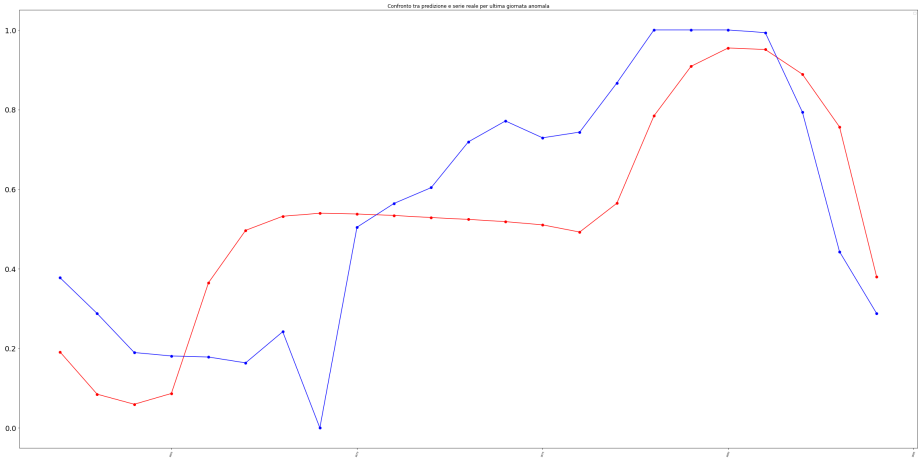

Figure 6: Soglia a 22, primo dataset: confronto tra giornata predetta e giornata anomala

4 Conclusioni

Osservando le immagini prodotte si nota l'evidente differenza tra la predizione e la giornata normale e la giornata anomala.
E' possibile anche notare la quasi identicità delle figure rappresentanti le predizioni con le time-series reali, tenuto conto che più la predizione può contare su un maggior numero di giorni normali da usare e più questa sarà precisa.
Per quanto riguardo la *cosine distance* possiamo osservare che nel caso del secondo dataframe la distanza è la minore rilevata.

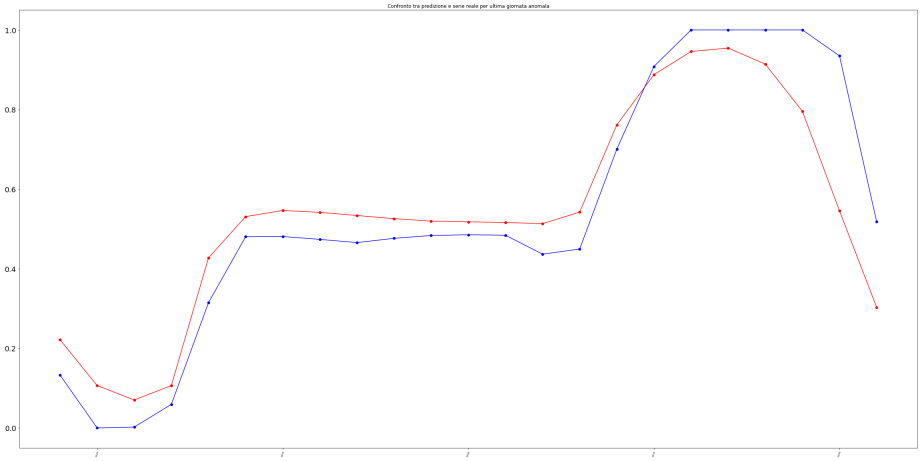

Figure 7: Soglia a 22, secondo dataset: confronto tra giornata predetta e giornata normale

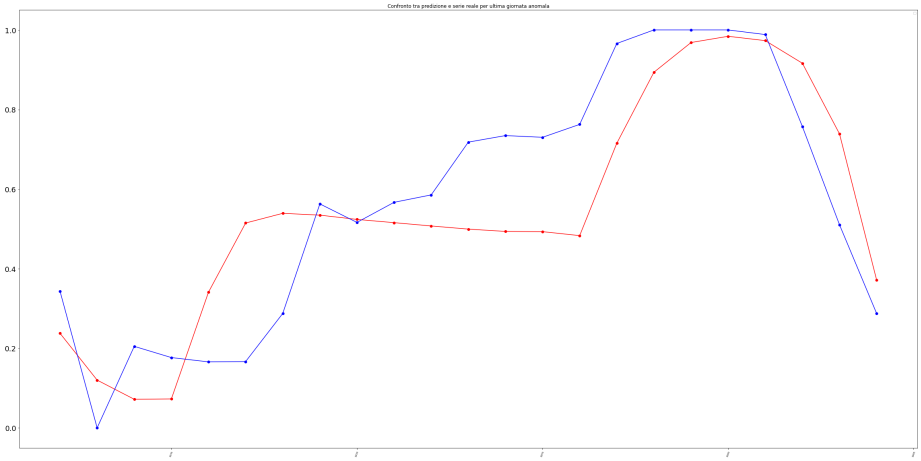

Figure 8: Soglia a 22, secondo dataset: confronto tra giornata predetta e giornata anomala

Questo è dovuto al fatto che il secondo dataframe è quello contenente le entrate relative alla massima finestra disponibile, ovvero composta dal massimo numero di giorni normali consecutivi.

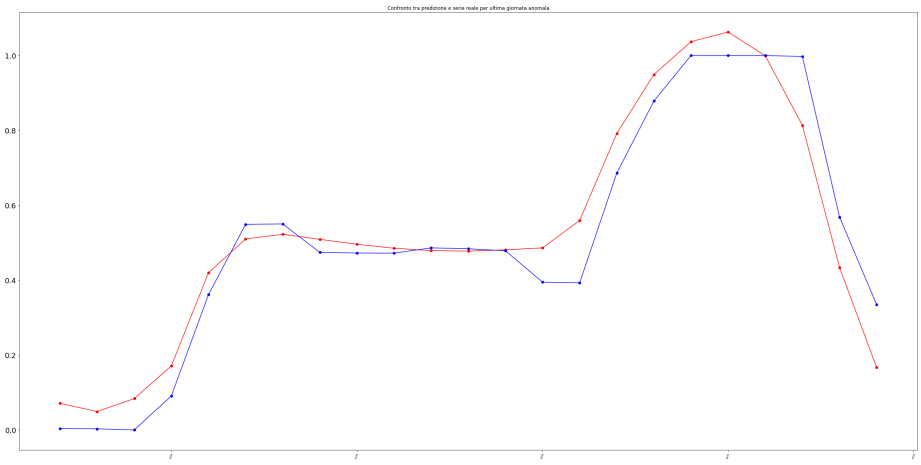

Figure 9: Soglia a 22, terzo dataset: confronto tra giornata predetta e giornata normale

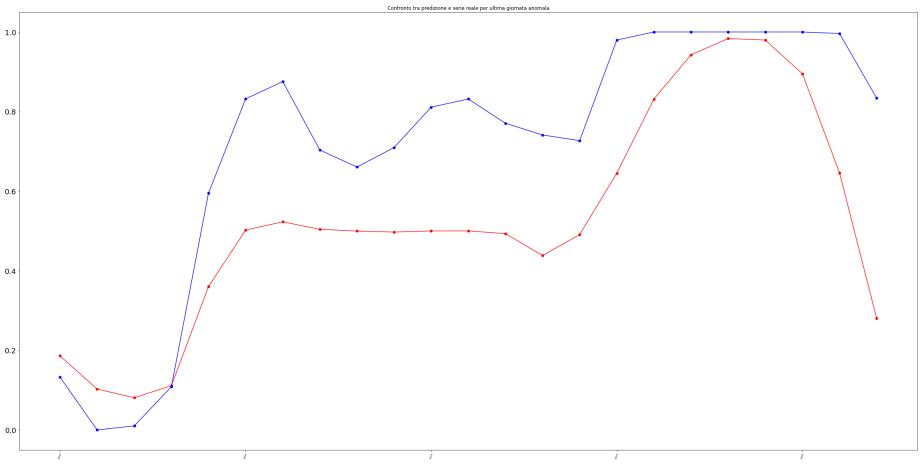

Figure 10: Soglia a 22, terzo dataset: confronto tra giornata predetta e giornata anomala

Data (Index)	Anomalia/Non anomalia	cosine distance
81	Non anomalo	0,00716183981648899
82	Anomalo	0,0707886920073656
141	Non anomalo	0,00565043322142455
142	Anomalo	0,0238151981270008
187	Non anomalo	0,0146362620007773
188	Anomalo	0,0193583450085392

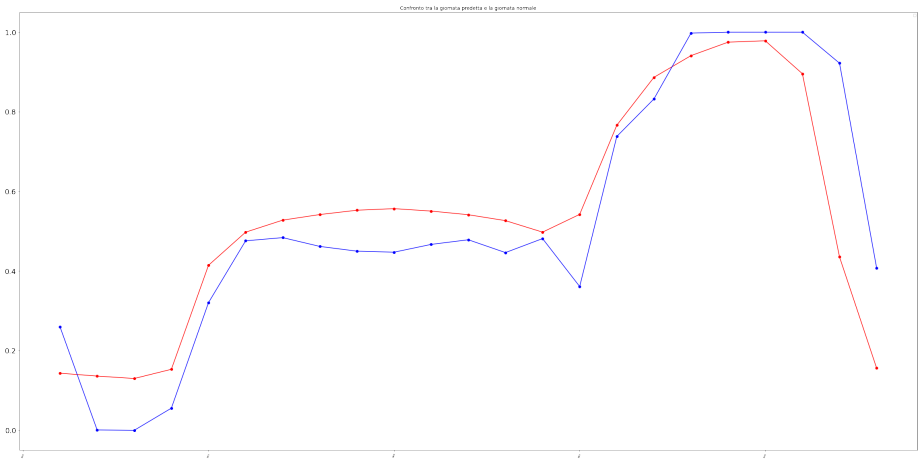

Figure 11: Soglia a 10, Giornata con index 25, non anomala.

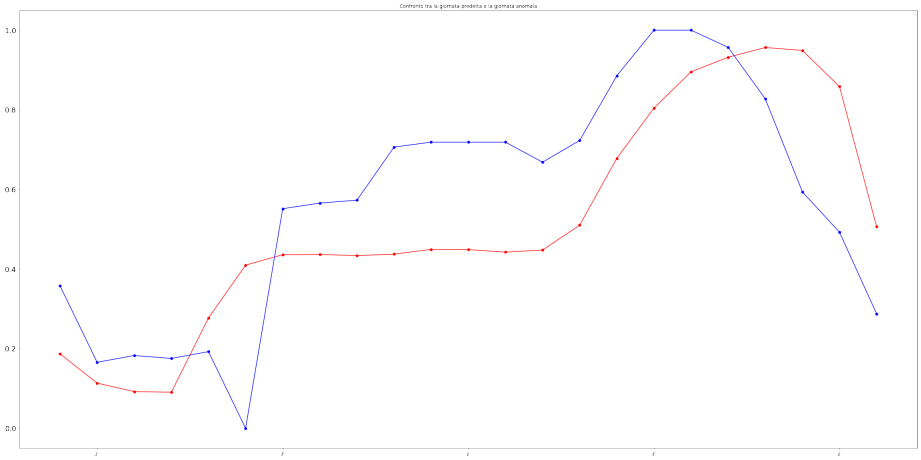

Figure 12: Soglia a 10, Giornata con index 26, anomala.

Possiamo inoltre notare che tutti i valori di *cosine distance* dei giorni anomali superano i valori relativi ai giorni normali.
Per avere più dati da analizzare potremmo provare ad abbassare la soglia da 23 giorni a 10. Questo ci permette di considerare 6 giorni anomali su 9 anzichè solamente 3 su 9.
La prima cosa che possiamo notare è che generalmente abbassando il valore della finestra di predizione otterremo predizioni meno precise. Infatti le migliori predizioni delle giornate anomale (valore di cosine distance tra giornata predetta e giornata anomala) vengono fornite dai giorni con indice 82 e 142, che corrispondono ai giorni con la finestra più ampia.

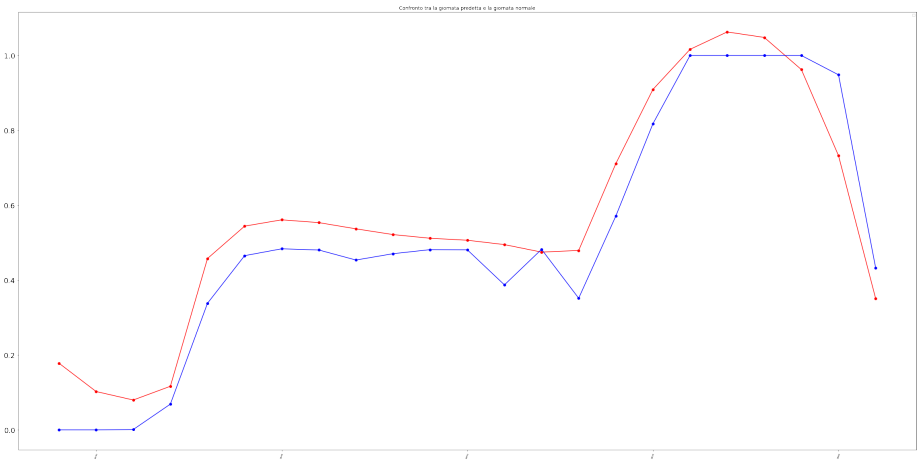

Figure 13: Soglia a 10, Giornata con index 81, non anomala.

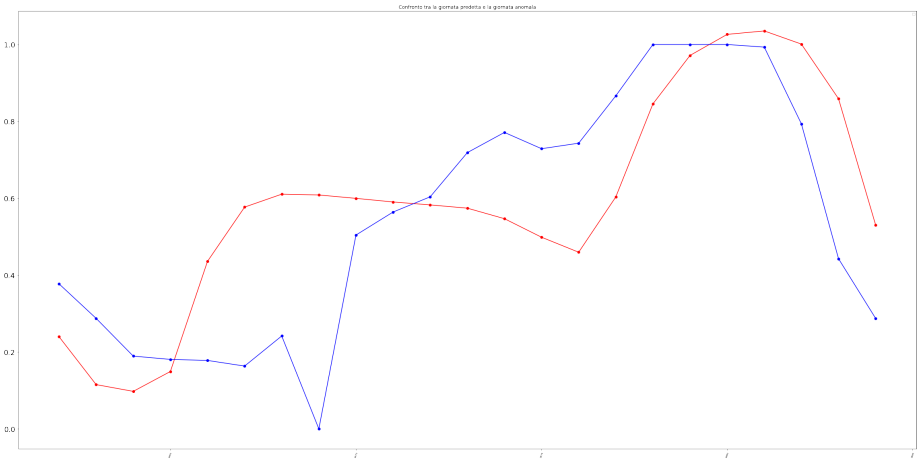

Figure 14: Soglia a 10, Giornata con index 82, anomala.

Questo se si esclude il dato relativo al giorno 26 che è uno di quelli a finestra minima (10 giorni) e quindi il dato relativo alla cosine distance potrebbe essere condizionato dalla predizione meno corretta della giornata normale, cosa che può essere confermata anche visivamente.

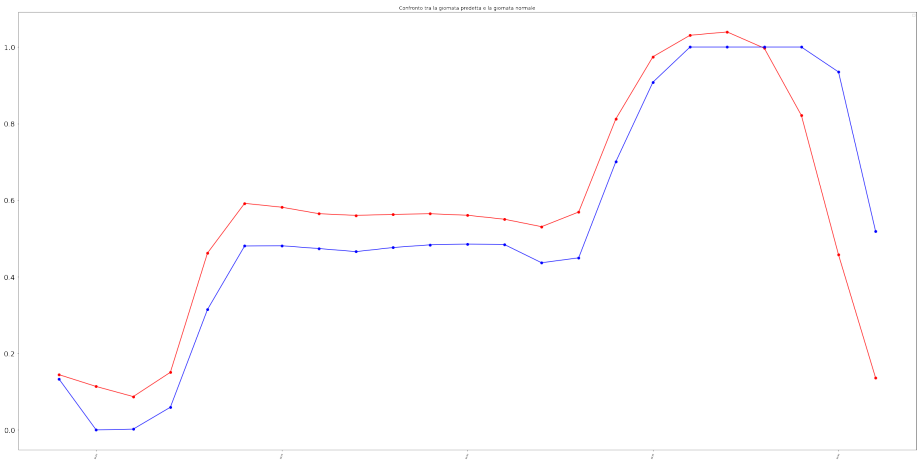

Figure 15: Soglia a 10, Giornata con index 141, non anomala.

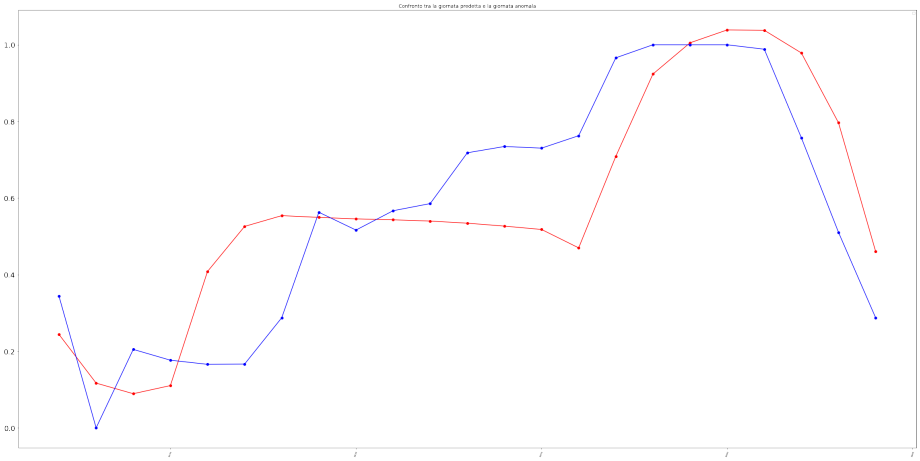

Figure 16: Soglia a 10, Giornata con index 142, anomala.

Data (Index)	Anomalia/Non anomalia	cosine distance
25	Non anomalo	0,027788794491170 11
26	Anomalo	0,059318941697522 72
81	Non anomalo	0,010871693489237777
82	Anomalo	0,07330551350792547
141	Non anomalo	0,03131250329642299
142	Anomalo	0,0393965106382177
152	Non anomalo	0,01938268993867709
153	Anomalo	0,02598689095459017
187	Non anomalo	0,009974987075688113
188	Anomalo	0,020252144437057584
204	Non anomalo	0,01848881759983234
205	Anomalo	0,017218302138751196

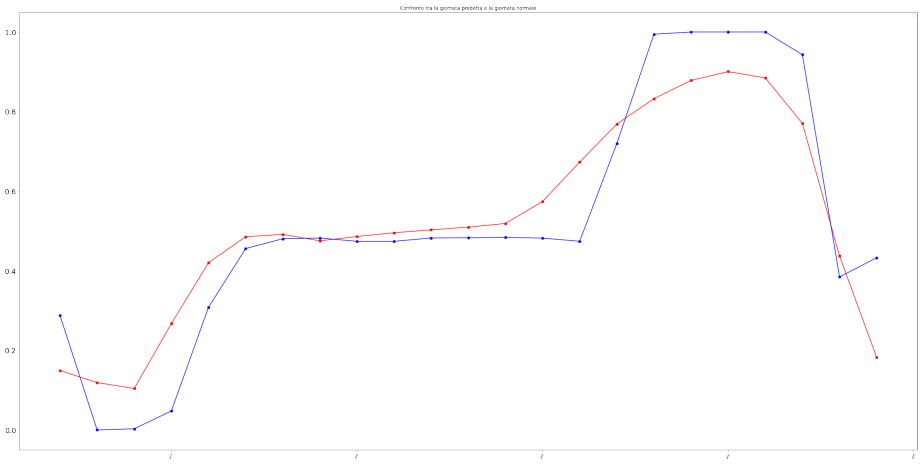

Figure 17: Soglia a 10, Giornata con index 152, non anomala.

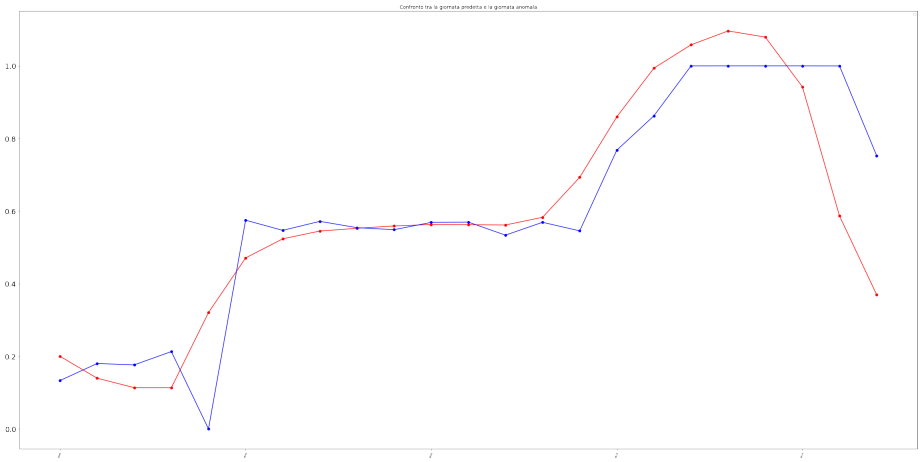

Figure 18: Soglia a 10, Giornata con index 153, anomala.

5 Risultati sperimentali

Dopo aver suddiviso il codice in classi e aggiunto tutte le funzionalità necessarie al predittore, possiamo passare a collaudare il sistema per controllare che dia risultati significativi anche al variare dei parametri in gioco e per un numero consistente di ripetizioni.
Sono state provate diverse configurazioni e di seguito viene riportata una breve spiegazione dei parametri usati:

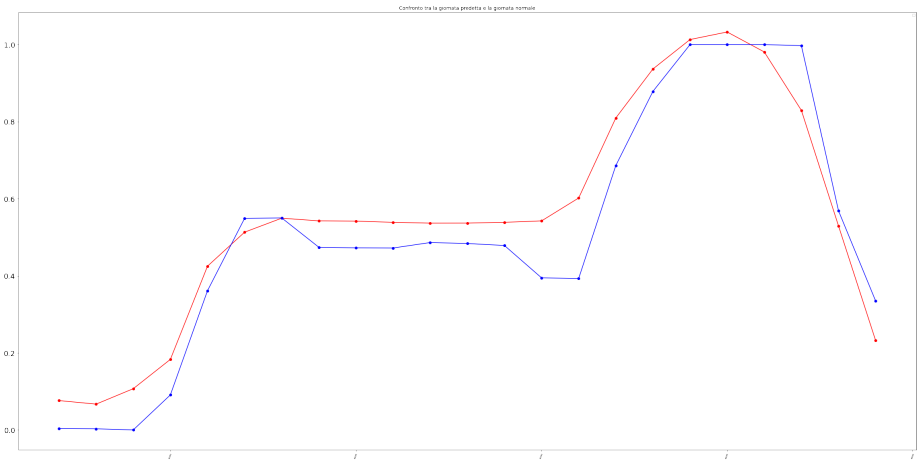

Figure 19: Soglia a 10, Giornata con index 187, non anomala.

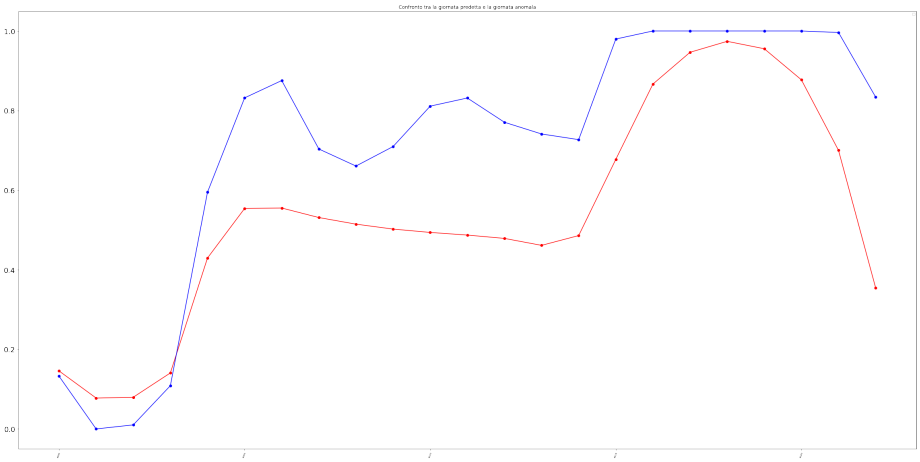

Figure 20: Soglia a 10, Giornata con index 188, anomala.

path: Si riferisce al percorso (all'interno di Google Colab) del file su cui si vuole effettuare la predizione.

max iter: Numero massimo di epoche di addestramento, di default pari a 200 ma nel nostro caso configurabile dall'utente.

hidden layer sizes: Indica la dimensione dello strato di neuroni nascosti. Di default pari a (100,) ma nel nostro caso è stato portato a (100,100) per aggiungere un ulteriore strato di neuroni nascosti che aumenta la precisione del sistema.

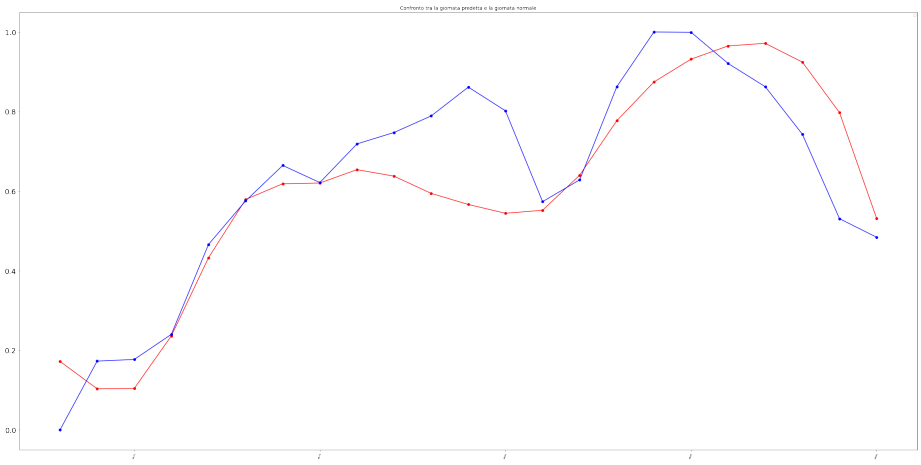

Figure 21: Soglia a 10, Giornata con index 204, non anomala.

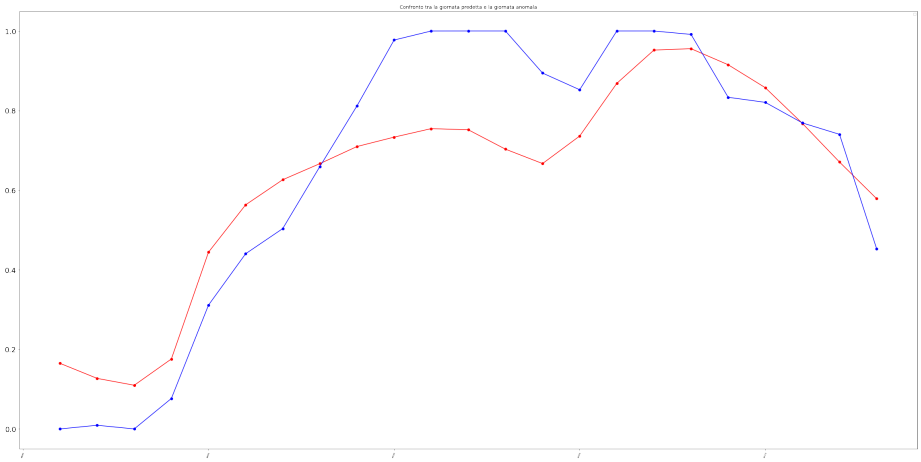

Figure 22: Soglia a 10, Giornata con index 205, anomala.

Il valore di *max iter* che si è dimostrato più efficiente è stato quello di 200, dopo di che si perde un po' di precisione forse per problemi che derivano dall'*overfitting* dei dati.
Restano validi i risultati ottenuti sulle cosine distance precedentemente. Infatti i giorni che sono stati classificati in modo migliore risultano essere quelli con una cosine distance più elevata.

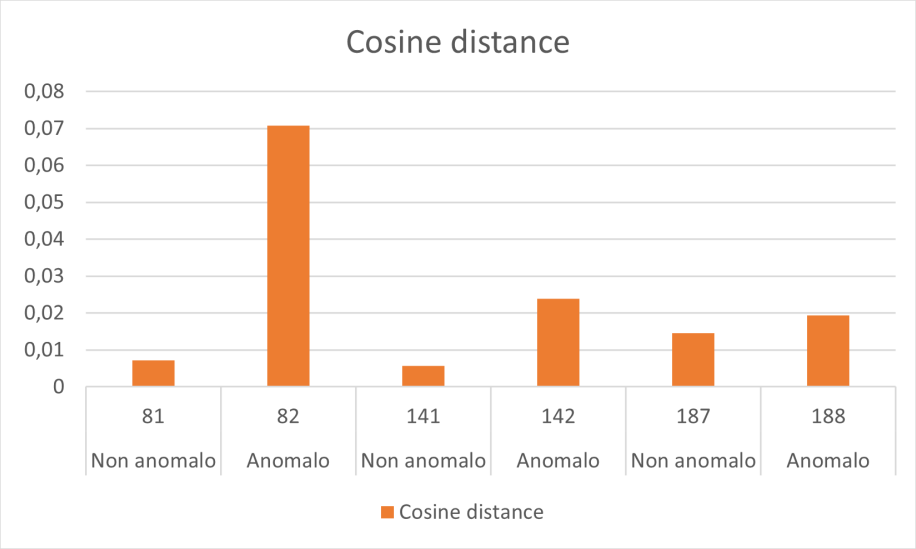

Figure 23: Istogramma che confronta i valori di cosine distance

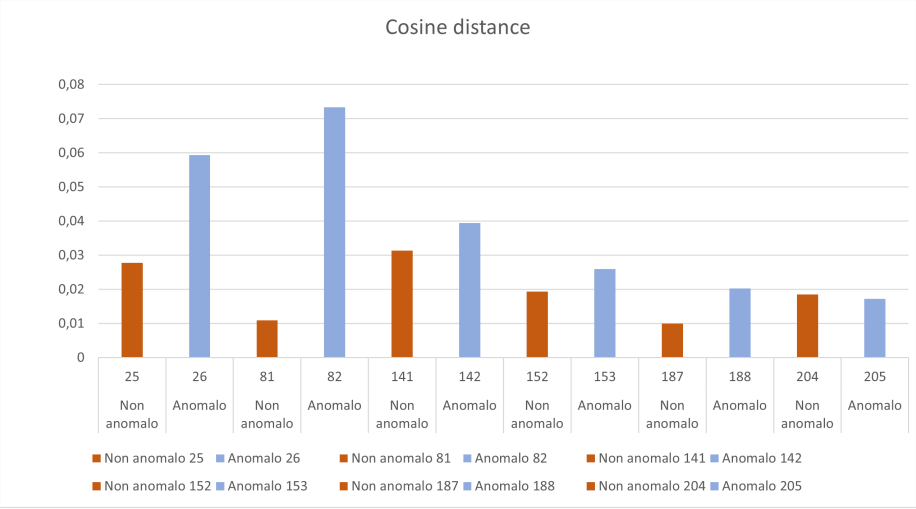

Figure 24: Istogramma che confronta i valori di cosine distance

Foglio1

Max_iter = 200	Predict	Score
1)	[0 1 0 1 0 1 0 0 0 0 0 0]	0.750000
2)	[0 1 0 1 0 1 0 0 0 0 0 0]	0.750000
3)	[0 1 0 1 0 1 0 0 0 0 0 0]	0.750000
4)	[0 1 0 1 0 1 0 0 0 0 0 0]	0.750000
5)	[0 1 0 1 0 1 0 0 0 0 0 0]	0.750000

Max_iter = 350	Predict	Score
1)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
2)	[1 1 0 1 0 1 0 0 0 0 0 0]	0.666667
3)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
4)	[1 1 0 1 0 1 0 0 0 0 0 0]	0.666667
5)	[1 1 0 1 0 1 0 0 0 0 0 0]	0.666667

Max_iter = 500	Predict	Score
1)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
2)	[1 1 0 1 0 1 0 0 0 0 0 0]	0.666667
3)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
4)	[1 1 0 1 0 1 0 0 0 0 0 0]	0.666667
5)	[1 1 0 1 0 1 0 0 0 0 0 0]	0.666667

Max_iter = 5	Predict	Score
1)	[0 0 0 0 0 0 0 0 0 0 0 0]	0.5
2)	[0 0 0 0 0 0 0 0 0 0 0 0]	0.5
3)	[1 1 1 1 1 1 1 1 1 1 1 1]	0.5
3)	[1 1 1 1 1 1 1 1 1 1 1 1]	0.5
5)	[0 0 0 0 0 0 0 0 0 0 0 0]	0.5

Max_iter = 1000	Predict	Score
1)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
2)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
3)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
4)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333
5)	[1 1 0 1 0 1 1 0 0 0 0 0]	0.583333

Pagina 1

Figure 25: Misurazioni effettuate.