UNIVERSITÀ DI PISA

SCUOLA DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica

Rilevamento di anomalie nelle dinamiche di popolamento di hotspot urbani attraverso One class Support Vector Machine

CANDIDATO

Francesco CAMPILONGO

RELATORI

Prof. Mario G.C.A. CIMINO

Prof. Gigliola VAGLINI

Ing. Antonio Luca ALFEO

ANNO ACCADEMICO 2019-2020

ABSTRACT

La possibilità di trovare anomalie nella mobilità urbana è un fattore molto importante per una serie di servizi che possono essere messi a disposizione del cittadino. Come ad esempio individuare una strada chiusa per lavori prima che questa scaturisca in un ingorgo. Ed è per questo motivo che l'obiettivo di questa tesi è, scovare queste anomalie, in maniera rapida automatica ed affidabile, sfruttando l'algoritmo di machine learning One Class Support Vector Machine.

In questa tesi sono partito da un dataset che raccoglieva i dati dei viaggi svolti dai taxi muniti di GPS in alcuni quartieri di New York i.e. Murray Hill, Gramercy Park ed East Village.

Quindi ho cercato manualmente le anomalie calcolando la distanza euclidea di ogni giorno dalla media e cercando quei giorni che più si distanziavano dal media, sfruttando grafici ed il 95° percentile, le anomalie trovate sono state confermate tramite ricerca sul web.

Il pool di anomalie trovate è stato utilizzato per "allenare" il One Class Support Vector Machine a trovare quante più anomalie possibili. Il OneClassSVM è stato provato variando alcuni parametri per ottenere la migliore configurazione possibile.

La scelta di questo algoritmo è stata fatta perché il OneClassSVM è conosciuto per essere un algoritmo efficace nell'individuare il confine tra anomalie ed istanze normali di una classe.

Per avere una panoramica migliore delle performance dell'algoritmo scelto, sono stati utilizzati accuratezza e precisione.

Gli esiti risultano essere molto incoraggianti.

Indice

1. INTRODUZIONE

La ricerca delle anomalie nella mobilità urbana è molto importante perché permette di ottenere una serie di servizi indiscutibilmente utili per i cittadini, come la possibilità di avere informazioni circa la chiusura di una strada prima che si crei un ingorgo, oppure la possibilità di scovare eventi guardando le aree vicine i luoghi in cui l'evento ha maggiore possibilità di aver luogo, oppure ancora cercare di sfruttare al meglio i dati sulle anomalie urbane per ottimizzare l'utilizzo di mezzi di trasporto pubblico ed ottenere vantaggi anche dal punto di vista dell'inquinamento urbano.

Lo studio qui riportato analizza i dati dei viaggi svolti dai taxi muniti di GPS in alcuni quartieri di New York, forse una delle città che fa maggiore affidamento nei caratteristici taxi gialli per la mobilità urbana quotidiana. I quartieri presi in considerazione in questa tesi sono Gramercy Park, Murray Hill e East Village. I dati risalgono all'anno 2015.

Lo scopo di questa tesi è capire come si comporta il One Class Support Vector Machine su un dataset come quello a disposizione, caratterizzato praticamente da vettori in ventitré dimensioni, ore di attività dei taxi, con trecentosessantacinque elementi, corrispondenti ai giorni dell'anno.

La tesi è suddivisa in due parti principali, ossia:

- *Ricerca manuale delle anomalie*: partendo dal dataset iniziale si cercano le anomalie manualmente;
- *Utilizzo del One Class Support Vector Machine*: partendo dal nuovo dataset ottenuto al punto precedente si cercano le anomalie sfruttando il OC-SVM.

Le motivazioni per la scelta del OC-SVM sono molteplici, la riconosciuta efficacia di questo algoritmo nell'individuare il confine tra istanze normali di una classe e anomalie, anche una migliore scalabilità rispetto ad un approccio statistico, che ha bisogno di un esperto per ogni città e la stessa città varia nel tempo.

2. SUPPORT VECTOR MACHINE

Sono dei modelli di supervised learning associati ad algoritmi di apprendimento per la regressione e la classificazione. In generale gli algoritmi di SVM costruiscono un iperpiano, o un insieme di iperpiani in uno spazio a più dimensioni o ad infinite dimensioni e questo (o questi) viene utilizzato per la classificazione per la regressione o per la ricerca delle anomalie.

2.1. One Class Support Vector Machine

Fa parte della famiglia degli algoritmi di support vector machine anche il One Class Support Vector Machine, preso in considerazione in questo scritto, e qui approfondito.

Supponiamo di avere un dataset di punti ad n-dimensioni alcuni "normali", presi da una distribuzione, ed altri anomali cioè presi da una diversa distribuzione. L'algoritmo cerca di trovare le anomalie senza esplicitamente definire punti anomali e punti normali. Prima mappa i dati all'interno di uno spazio delle funzioni, il kernel; Poi invece di separare il dataset in base alle sue etichette, li separa dall'origine O (con origine vettore di n zeri) attraverso un iperpiano. L'algoritmo cerca di massimizzare la distanza dall'iperpiano e l'origine. Vengono introdotte delle variabili 'slack' che tollerano che alcuni punti anomali ricadano all'interno dell'iperpiano venendo così marcati come normali, quelli esterni all'iperpiano sono definiti anomali. La distanza del punto dall'iperpiano può essere usata come misura del grado di anomalia.

OC-SVM e SVM differiscono tra loro per il tipo di dati che possono essere usati per allenare l'algoritmo (solo positivi nel primo, positivi e negativi nel secondo caso), il numero di "macchine" (una nel primo caso, molteplici nel secondo) e la necessita di un post-processing (nessuna nel primo mentre necessaria nel secondo per unire i vari output).

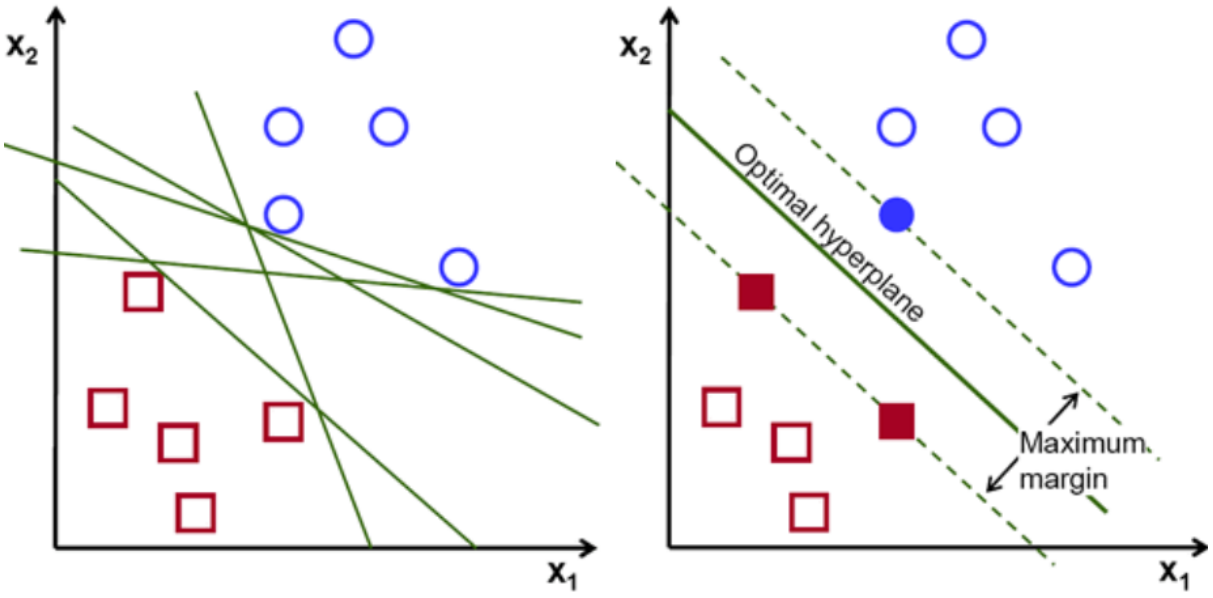


Figura 1 Rappresentazione grafica del comportamento del One Class SVM

In *Figura 1* è possibile notare come si comporta un algoritmo basato su Support Vector Machine, come vengono utilizzati più iperpiani per distaccare ciò che sono punti anomali da punti normali, è quello che viene chiamato iperpiano ottimale, equidistante dai punti anomali estremi e quelli normali estremi, che divide a metà il margine massimo.

2.2. Design ed implementazione

Il lavoro svolto su python per utilizzare l'algoritmo One Class Support Vector Machine è riassumibile nel seguente diagramma delle classi

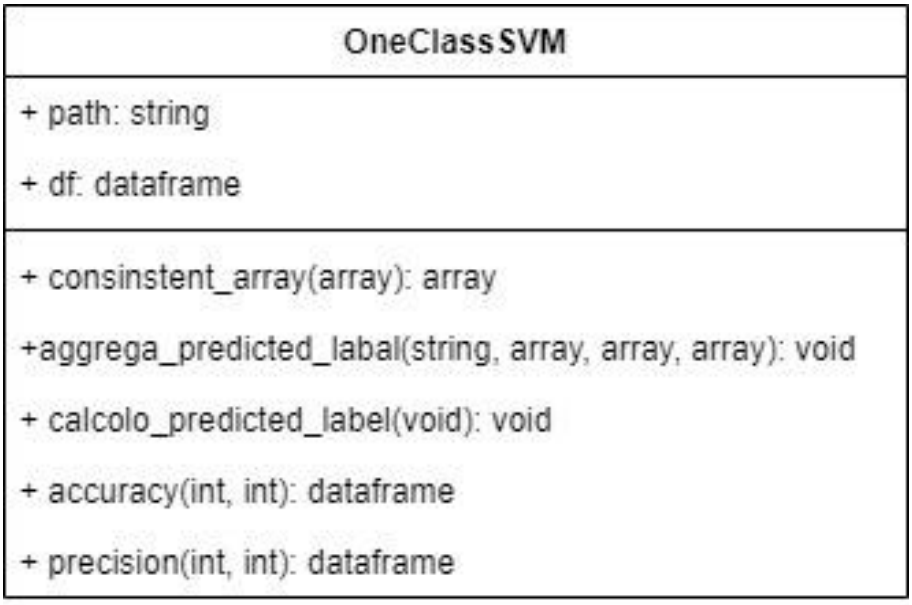


Figura 2 Diagramma delle classi per OneClassSVM

- path: è la variabile interna che mantiene il percorso ed il nome del file .csv che contiene i dati con le anomalie aggiornate.
- df: è il dataframe che otteniamo leggendo il file csv con la funzione della libreria pandas.

I metodi scritti per il conseguimento del risultato sono:

- consinstent_array: è una funzione alla quale viene passato un array contenente -1, 1 e restituisce un array contenente rispettivamente 1 e 0, è stata ideata per rendere i dati più consistenti con i dati dei file a disposizione, perché la funzione della libreria sklearn OneClassSVM.predict(array) restituisce come valori -1 se il giorno è anomalo e 1 per il giorno normale, mentre il ground truth, cioè la colonna di anomalie trovate manualmente, è riempita con 0 e 1 (normale e anomalia), quindi per permettere l'apprendimento all'algoritmo le colonne create devono essere consistenti.
- aggrega_predicted_label: è una funzione che crea una colonna nel file .csv con il nome che viene passato (l'argomento della funzione string), ed i tre array che vengono passati come argomenti, è necessaria questa funzione perché il dataset a disposizione è diviso in tre cluster (sarà più chiaro dopo), ed ogni

array corrisponde al risultato della funzione predict di OneClassSVM, resi compatibili con i dati a disposizione, quindi convertiti con la funzione consistent_array, suddivisi per i vari cluster.

- calcolo_predicted_label: è una funzione che ha lo scopo di aggiungere al dataset a disposizione le colonne che abbiamo creato con l'algoritmo del OneClassSVM, le colonne create sono consistenti con i dati ottenuti ed ordinate come i giorni dell'anno, facendo attenzione al cluster d'appartenenza dei giorni, le colonne create sono molteplici per motivazioni che verranno chiarite nel capitolo della sperimentazione.
- accuracy: una funzione che restituisce un dataframe, per semplicità logistiche nella creazione di tabelle, questa calcola l'accuratezza dei risultati ottenuti, ha due argomenti interi, che indicano il cluster sul quale si sta lavorando e un offset per raccogliere le colonne corrette create precedentemente con la funzione calcolo_predicted_label.
- precision: è una funzione gemella accuracy, ma come è facile immaginare non calcola l'accuratezza ma la precisione, anche gli argomenti hanno la stessa funzione della funzione accuracy, sono state create due funzioni diverse, e non una sola, per facilità di lettura e per libertà di utilizzo, nel caso in cui non fosse stato necessario calcolare entrambi gli indici.

Ulteriori dettagli sulle metriche scelte per calcolare l'accuratezza del risultato ottenuto con One Class SVM si possono trovare nel capitolo 4.1.

Nella figura successiva (*Figura 3*) vengono mostrate le funzioni scritte per il calcolo di accuratezza e precisione.

```python
def accuracy(index, cluster):
    dfr = pa.DataFrame(columns=['gamma=scale', 'gamma=auto', 'gamma=0.05'])
    dff = pa.read_csv('prova.csv')
    dff = dff.loc[dff['Cluster'] == cluster]
    xA = np.array(dff['Anomalous'])

    # seleziono solo le colonne che ho creato con le varie configurazioni di OneClassSVM
    dff = dff.iloc[:,3:30]
    for i in range(0, 3):
        xP0 = np.array(dff.iloc[:, (index + i)])
        xP1 = np.array(dff.iloc[:, (index + i + 3)])
        xP2 = np.array(dff.iloc[:, (index + i + 6)])
        r = np.array([metr.accuracy_score(xA, xP0), metr.accuracy_score(xA, xP1), metr.accuracy_score(xA, xP2)])
        dfr.loc[i] = r
    dfr = dfr.rename(index={0:'max_iter=100', 1:'max_iter=200', 2:'max_iter=500'})
    return dfr

def precision(index, cluster):
    dfr = pa.DataFrame(columns=['gamma=scale', 'gamma=auto', 'gamma=0.05'])
    dff = pa.read_csv('prova.csv')
    dff = dff.loc[dff['Cluster'] == cluster]
    xA = np.array(dff['Anomalous'])
    dff = dff.iloc[:,3:30]
    for i in range(0, 3):
        xP0 = np.array(dff.iloc[:,(index + i)])
        xP1 = np.array(dff.iloc[:,(index + i + 3)])
        xP2 = np.array(dff.iloc[:,(index + i + 6)])
        r = np.array([metr.precision_score(xA, xP0), metr.precision_score(xA, xP1), metr.precision_score(xA, xP2)])
        dfr.loc[i] = r
    dfr = dfr.rename(index={0: 'max_iter=100', 1: 'max_iter=200', 2: 'max_iter=500'})
    return dfr
```

Figura 3 Implementazione delle funzioni accuracy e precision

2.2.1. *Ambiente di sviluppo*

In questa tesi l'ambiente di sviluppo scelto è stato pycharm professional, ambiente di sviluppo per python (e non solo), con funzioni di programmazione avanzate.

Le librerie importate per il completamento degli studi svolti sono:

- pandas: per la gestione dei file .csv
- matlib.pyplot: per il disegno dei grafici
- scypy: per il calcolo delle distanze euclidee
- numpy: per la manipolazione di array
- sklearn.svm: per l'implementazione del OneClassSVM
- sklearn.metrics: per il calcolo di accuratezza e precisione

2.2.2. *Vocabolario*

- OneClassSVM.fit(array): questa funzione permette all'algoritmo OneClassSVM di apprendere dall'array passato come argomento, viene utilizzata insieme alla funzione predict.

- OneClassSVM.predict(array): una volta che è stata lanciata la funzione fit, l'algoritmo ha la possibilità adesso di dare dei risultati, in base all'array passato come argomento, questa funzione restituisce -1 e 1 (anomalo, normale), questo risultato viene modificato per renderlo compatibile con ground truth, ossia 1 e 0 (rispettivamente anomalia e giorno normale).

3. CASO DI STUDIO

3.1. Analisi dei dati

Come detto precedentemente i dati utilizzati in questa tesi sono quelli dell'hotspot D (*Figura 4*) di Manhattan, New York, precisamente sono i dati delle persone che salgono e scendono dai taxi nei quartieri di Murray Hill, Gramercy Park ed East Village. I dati sono organizzati in un file .csv contenente i giorni dell'anno. I giorni sono suddivisi in gruppi o *cluster* con i possibili valori {0,1,2} che nell'ordine indicano giorni lavorativi, giorni prefestivi e giorni festivi. È stata fatta questa suddivisione per rendere maggiormente effettivi i dati dei taxi, in quanto il comportamento dei clienti varia in base al tipo di giorno e.g. in un giorno lavorativo nei pressi di uffici ci sarà sicuramente più mobilità rispetto ad un giorno festivo nella stessa zona.

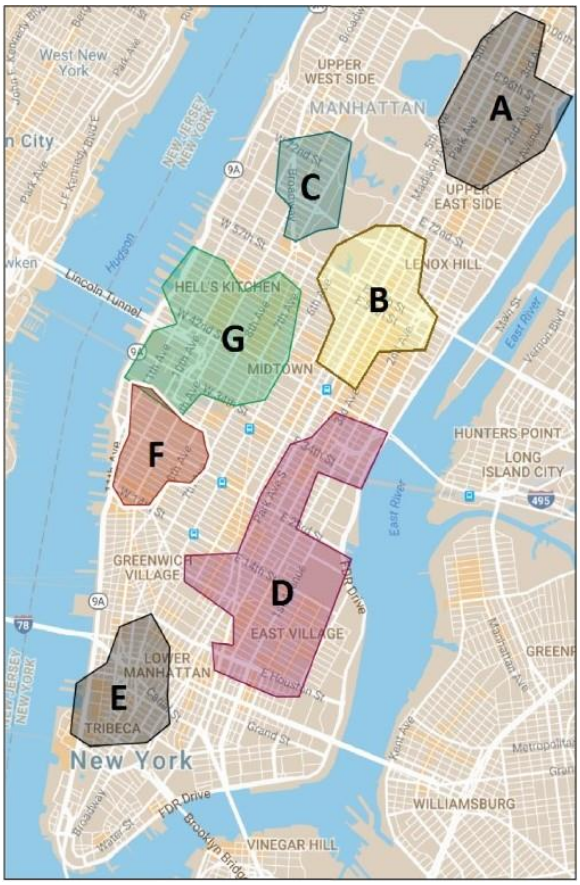


Figura 4 Isola di Manhattan, New York. Locazione hotspot D

3.2. Ricerca delle anomalie

Vista l'organizzazione dei dati a disposizione, il passo successivo è stato quello di ricercare le anomalie.

Per ricercare le anomalie sono stati seguiti due approcci:

1. Creato l'istogramma, e cercato di escludere quei giorni che graficamente erano più distanti dalla maggior parte degli altri, producendo una soglia.
2. Utilizzato il 95° percentile.

Il motivo di questi due approcci è stato per ottenere dati statisticamente validi e cercare di ottenere un buon numero di anomalie. Il 95° percentile viene utilizzato come soglia effettiva per la ricerca delle anomalie (secondo approccio) mentre l'ispezione visiva dell'istogramma (primo approccio) serve ad integrare ulteriormente il pool iniziale di anomalie, che verranno successivamente sfoltite con ricerche web.

In seguito la descrizione della fase preliminare per affrontare i due approcci:

• selezionare le colonne interessate (affluenza oraria giornaliera [h1:h23])

• selezionare le righe con Cluster = {0,1,2}

• calcolare la media per ogni colonna (funzione mean messa a disposizione dalla libreria pandas, utilizzata per manipolare i file .csv)

• calcolare la distanza euclidea tra due vettori a 23 dimensioni (media e valori giornalieri orari in base ai vari cluster)

• disegnare l'istogramma utilizzando la libreria matplotlib.pyplot

Per quanto riguarda il primo approccio, tramite un ispezione visiva dell'istogramma, è facile vedere quelli che potenzialmente sono anomalie, cioè quei dati con una distanza euclidea maggiore rispetto alla maggioranza, quindi è stata scelta una soglia visiva, ed i giorni con

distanza euclidea maggiore della soglia visiva trattati come potenziali anomalie.

Per il secondo approccio invece, viene calcolato il 95° percentile (tramite la funzione della libreria numpy, nanpercentile) sull'array delle distanze euclidee trovato, per ogni cluster. Ottenendo un numero che indica la soglia oltre la quale i giorni hanno una distanza euclidea dalla media maggiore rispetto al 95% dei giorni considerati.

I successivi istogrammi (*Figura 5, Figura 6, Figura 7)* sono organizzati con valori delle distanze euclidee sull'asse delle ascisse, posizionati coerentemente in base alle distanze riscontrate, mentre sull'asse delle ordinate vengono riportati delle quantità che rappresentano il numero di ricorrenze in funzione delle distanze euclidee.

In seguito i risultati ottenuti per i vari cluster.

3.2.1. *Cluster 0 [Giorni lavorativi]*

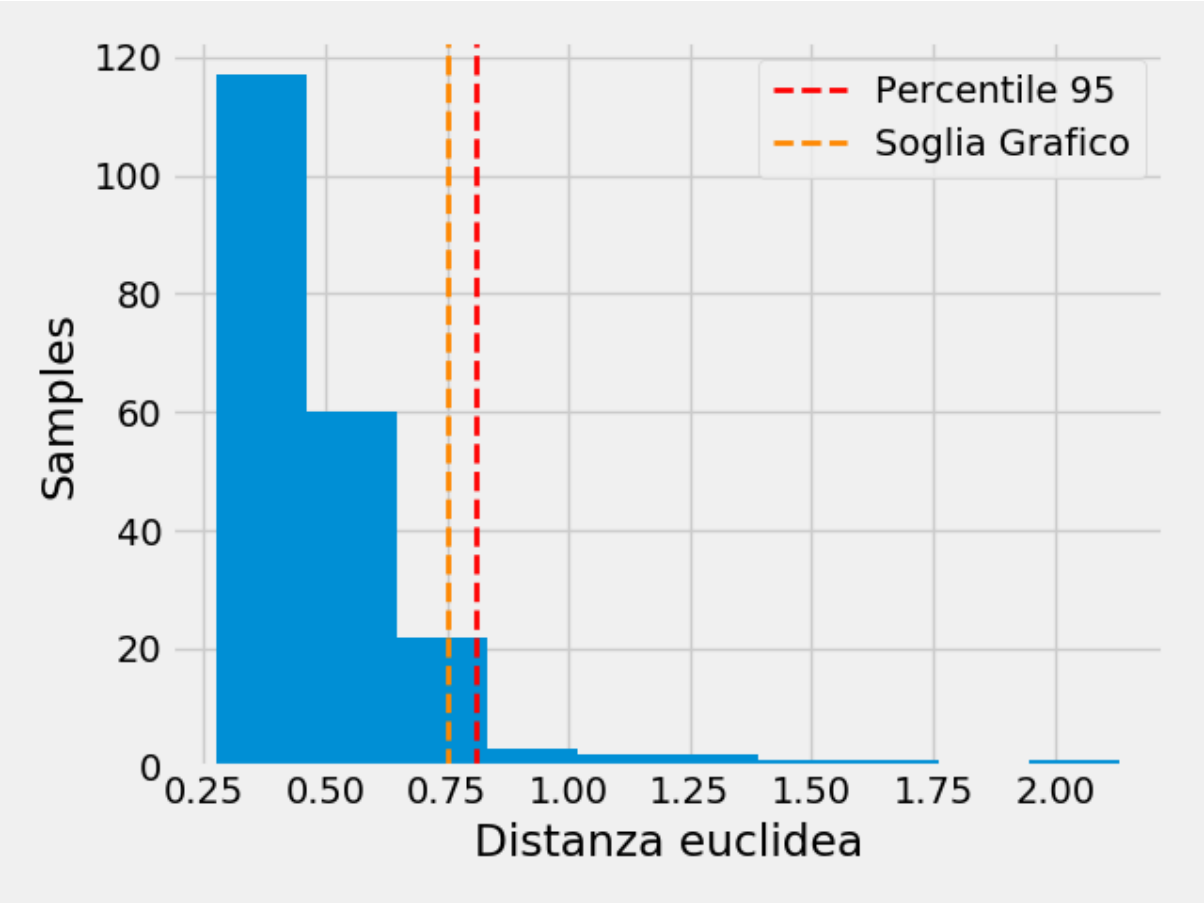


Figura 5 Istogramma delle distanze euclidee dalla media [cluster 0]

L'istogramma di *Figura 5* rappresenta come la maggior parte dei giorni, del cluster 0, siano distribuiti vicino una distanza euclidea che va dai 0.25 ai 0.75, motivo per il quale la soglia scelta graficamente è stata 0.75 (primo approccio, linea tratteggiata arancione), mentre il calcolo del 95° percentile ha restituito una soglia di 0.8117 (secondo approccio, linea tratteggiata rossa). Il pool di anomalie trovate tramite le due soglie è stato sfoltito tramite ricerca sul web, restituendo le anomalie per il primo cluster che sono riassunte nella tabella successiva.

Data	Motivazione
01/01	Capodanno
19/01	https://www.cs.ny.gov/attendance_leave/2015_legal_holidays.cfm
26/01	https://en.wikipedia.org/wiki/January_2015_North_American_blizzard
27/01	https://en.wikipedia.org/wiki/January_2015_North_American_blizzard
16/02	https://www.cs.ny.gov/attendance_leave/2015_legal_holidays.cfm
25/05	https://www.cs.ny.gov/attendance_leave/2015_legal_holidays.cfm
07/09	https://www.cs.ny.gov/attendance_leave/2015_legal_holidays.cfm
24/09	Papa in visita a New York
25/11	Vigilia giorno del ringraziamento
26/11	Giorno del Ringraziamento
24/12	Vigilia di Natale
31/12	Vigilia di Capodanno

Le anomalie sopra descritte sono state confermate tramite uno scrupoloso lavoro di ricerca svolto dopo aver ottenuto i giorni possibilmente anomali con gli approcci sopra descritti.

Segue la disamina approfondita delle query che sono state fatte nel motore di ricerca per ottenere risultati risalenti a cinque anni fa (vedere capitolo 3.3 Criterio di ricerca).

3.2.2. *Cluster 1[Giorni prefestivi]*

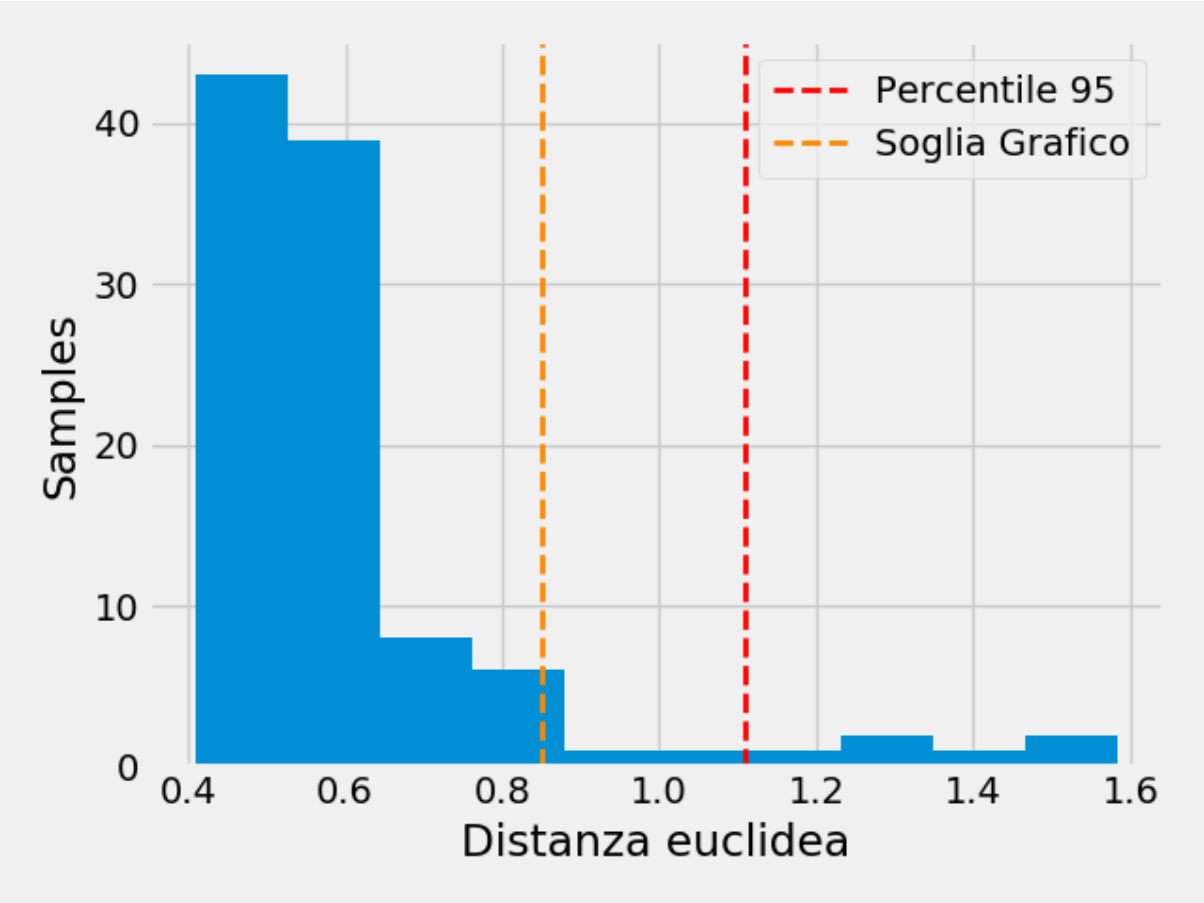


Figura 6 Istogramma delle distanze euclidee dalla media [cluster 1]

Dalla *Figura 6* è possibile notare come la maggior parte dei giorni del cluster 1 ha una distanza euclidea dalla media (del cluster) compresa tra 0.4 e circa 0.9, motivo per il quale la scelta della soglia grafico è stata fatta a 0.85 (linea tratteggiata arancione, primo approccio) mentre il calcolo del 95° percentile ha restituito una soglia di 1.1097 (linea tratteggiata rossa, secondo approccio). Il lavoro di filtraggio svolto da queste due soglie ha permesso di ottenere le anomalie seguenti, raccolte in tabella.

Data	Motivazione
09/01	https://nypost.com/2015/01/09/motorists-angry-over-alternative-side-parking-rules-during-snowfall/
16/01	https://www.dnainfo.com/new-york/20150116/midtown-east/worlds-top-squash-players-battle-grand-central-tournament/
06/03	https://www.wunderground.com/article/storms/winter/news/winter-storm-thor-snow-ice-reports
03/07	Vigilia dichiarazione indipendenza
05/09	Inizio week-end lungo, Labor day (07/09)
25/09	https://en.wikipedia.org/wiki/Pope_Francis%27s_2015_visit_to_North_America#25_September_(New_York_City,_United_Nations)
31/10	Halloween
13/11	https://www.dnainfo.com/new-york/20151113/midtown/penn-station-shooter-wanted-be-drug-kingpin-at-mcdonalds-sources/
12/12	https://www.amny.com/news/santacon-nyc-2015-see-photos-1-11221044/
25/12	Natale

Anche in questo caso le anomalie sono state confermate tramite una ricerca approfondita sul web, dopo un iniziale filtraggio tramite le soglie ottenute dai due approcci su menzionati. Maggiori info sui criteri di ricerca per confermare anomalie di oltre cinque anni fa nel capitolo 3.3. Criteri di ricerca.

3.2.3 *Cluster 2 [Giorni festivi]*

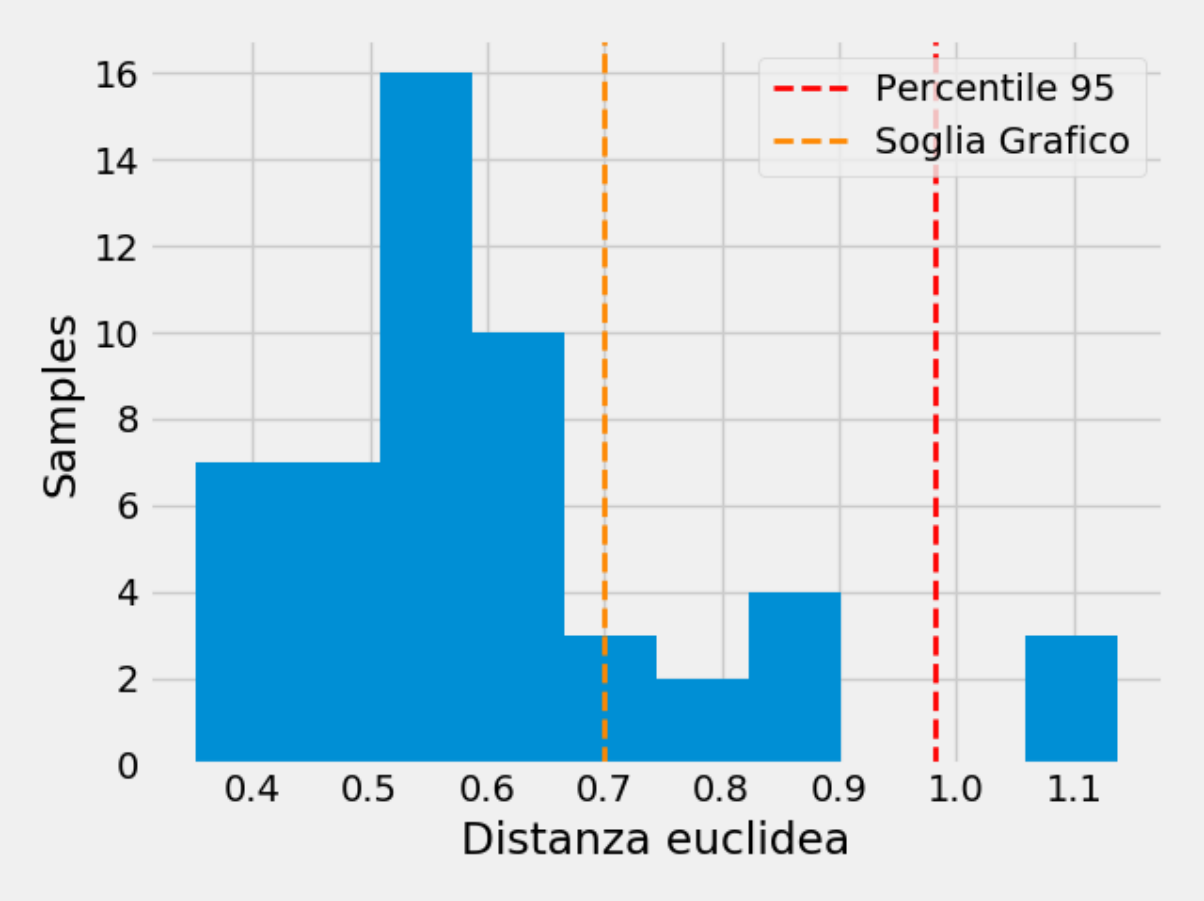


Figura 7 Istogramma delle distanze euclidee dalla media [cluster 2]

In *Figura 7* è rappresentato l'istogramma della distribuzione delle distanze euclidee dalla media per il cluster 2. È facile notare che la maggior parte dei giorni sono distanti dalla media in un intervallo compreso da 0.4 e 0.7, in questo cluster effettivamente è stata scelta graficamente la soglia a 0.7 (linea tratteggiata arancione, primo approccio) mentre il calcolo del 95° percentile ha restituito come seconda soglia 0.9815 (linea tratteggiata rossa, secondo approccio). Nel secondo e terzo cluster è possibile notare una distanza tra linea tratteggiata arancione (primo approccio) e linea tratteggiata rossa (secondo approccio) leggermente maggiore perché a differenza del primo cluster, gli ultimi due sono meno popolati. Il filtraggio ottenuto con gli approcci esaminati ci ha restituito le seguenti anomalie.

Data	Motivazione
01/02	https://nypost.com/2015/02/01/man-critically-injured-after-jumping-from-manhattan-bridge/
22/02	https://www.timeanddate.com/weather/usa/new-york/historic?month=2&year=2015
01/03	https://en.wikipedia.org/wiki/Early_March_2015_North_American_winter_storm
08/03	Giornata mondiale della Donna
15/03	https://newyork.cbslocal.com/top-lists/best-parades-and-celebrations-on-st-patricks-day-in-nyc/
24/05	Vigilia Memorial Day
31/05	https://www.dailymail.co.uk/news/article-3104812/NYPD-Material-lifted-crane-building-falls.html
28/06	https://www.washingtonpost.com/news/post-nation/wp/2015/06/28/millions-flood-new-york-city-and-san-francisco-streets-to-celebrate-gay-pride/
06/09	Vigilia Labor day

Queste sono tutte le anomalie trovate e che sono state confermate con ricerche approfondite sul web.

E con queste anomalie inizia lo studio del comportamento del One Class Support Vector Machine.

3.3. Criterio di ricerca

Cercare notizie dal web di fatti ed eventi accaduti più di cinque anni fa non è semplice, fortunatamente ci viene incontro Google con il suo potentissimo motore di ricerca. Per cercare notizie riguardanti i luoghi dell'hotspot preso in considerazione, sono state svolte cinque ricerche con cinque parole chiavi i.e. Murray Hill, Gramercy Park, East Village, New York e Manhattan, le prime tre rappresentano i quartieri dell'hotspot, mentre gli ultimi due sono ricerche un po' più generali per allargare i risultati ottenuti ed investigare più a fondo, e di volta in volta sfruttando gli strumenti del famoso motore di ricerca sono state modificati gli intervalli di tempo in cui le notizie erano state pubblicate, andando a scegliere come giorno iniziale della ricerca il giorno presumibilmente anomalo e come giorno finale della ricerca il giorno successivo al giorno presumibilmente anomalo. Credo sia stato il miglior compromesso per ottenere risultati nella ricerca in maniera veloce ed affidabile.

4. SPERIMENTAZIONE

Trovate le anomalie, viste nel capitolo precedente, si passa alla sperimentazione con l'algoritmo One Class Support Vector Machine, per farlo si è prodotto del codice (come anticipato nel capitolo 2.2) con dei metodi. Data un file .csv i metodi elaborati permettono di ottenere i risultati del OneClassSVM e calcolarne l'accuratezza e la precisione.

Il OneClassSVM è implementato nella libreria sklearn.svm e permette la modifica di alcuni parametri – kernel, degree, gamma, coef0, tol, nu, max_iter, shrinking, cache_size, verbose. I parametri che sono stati modificati per cercare la miglior configurazione possibile sono, kernel, max_iter, gamma e nu.

- Kernel:
  - poly
  - rbf (radial basis function)
  - sigmoid
- max_iter:
  - 100
  - 200
  - 500
- gamma:
  - Scale
  - Auto
  - 0.05
- nu calcolato in maniera automatica come:

$$nu = \frac{\text{numero giorni anomali}}{\text{numero giorni normali}}$$

Come già anticipato lo scopo è quello di ottenere i migliori risultati possibili, ecco perché sono state prese in considerazione tutte queste combinazioni di parametri, i kernel scelti sono quelli che davano risultati più promettenti, la possibilità di modificare il numero di iterazione (*max_iter*) è stata fatta per eventualmente migliorare i risultati permettendo all'algoritmo di "studiare" più o meno affondo i

dati passati, il parametro *gamma* ha due possibili scelte automatiche, scale (1 / n_features * x.var()) e auto (1 / n_features), ma questo può anche avere valori numerici e dopo varie prove è stato incluso tra le opzioni anche 0.05 per i risultati ottenuti, infine il parametro *nu* può avere valori compresi tra [0,1) ma la configurazione automatica (vista sopra) è risultata ottimale.

4.1. Risultati ottenuti

Di seguito vengono mostrati i risultati ottenuti suddivisi per cluster.

Per comprendere meglio le performance dell'algoritmo nelle sue diverse configurazioni sono state scelte accuratezza e precisione come metriche.

Accuratezza: grado di corrispondenza del dato teorico, desumibile da una serie di valori misurati, con il dato reale o di riferimento, ovvero la differenza tra il valore medio campionario ed il valore di riferimento.

La formula che descrive l'accuratezza è:

$$\text{accuratezza} = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

Precisione: grado di convergenza di dati rilevati individualmente rispetto al valore medio nella serie cui appartengono, ovvero la varianza rispetto alla media campionaria.

La formula che descrive la precisione è:

$$\text{precisione} = \frac{TruePositive}{TruePositive + FalsePositive}$$

Per rendere più chiare le formule sopra descritte:

Ground Truth	Predicted with OneClassSVM	
	Negative	**Positive**
Negative	TrueNegative	FalsePositive
Positive	FalseNegative	TruePositive

Con la tabella sopra descritta, si vuole spiegare cosa si intende per TrueNegative, TruePositive, FalseNegative, FalsePositive. Nomenclature utilizzate per spiegare i risultati ottenuti tramite One Class Support Vector Machine, quando Ground Truth, e.g. le anomalie trovate manualmente nel capitolo precedente, è negative (0) ed il OneClassSVM restituisce 0, allora si tratta di un TrueNegative, ossia il valore iniziale era 0 ed il valore trovato è 0, mentre nel caso in cui in valore iniziale fosse 0 ed il risultato è 1 si parlerebbe di FalsePositive; FalseNegative nel caso in cui il valore iniziale fosse 1 ed il valore ottenuto con OneClassSVM sia 0 ed infine TruePositive quando valore iniziale e valore trovato con l'algoritmo di machine learning siano entrambi 1.

L'utilizzo di entrambi i parametri permette di avere una buona stima del comportamento delle configurazioni provate.

4.1.1. Cluster 0 [Giorni lavorativi]

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.91866	0.923445	0.91866
max_iter=200	0.91866	0.923445	0.91866
max_iter=500	0.91866	0.923445	0.91866

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.307692	0.333333	0.307692
max_iter=200	0.307692	0.333333	0.307692
max_iter=500	0.307692	0.333333	0.307692

Figura 8 Accuracy e Precision – kernel = poly

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.966507	0.956938	0.966507
max_iter=200	0.966507	0.956938	0.966507
max_iter=500	0.966507	0.956938	0.966507

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.727273	0.615385	0.727273
max_iter=200	0.727273	0.615385	0.727273
max_iter=500	0.727273	0.615385	0.727273

Figura 9 Accuracy e Precision – kernel = rbf

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.91866	0.923445	0.923445
max_iter=200	0.91866	0.923445	0.923445
max_iter=500	0.91866	0.923445	0.923445

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.307692	0.333333	0.333333
max_iter=200	0.307692	0.333333	0.333333
max_iter=500	0.307692	0.333333	0.333333

Figura 10 Accuracy e Precision – kernel = sigmoid

4.1.2. Cluster 1 [Giorni prefestivi]

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.923077	0.903846	0.923077
max_iter=200	0.923077	0.903846	0.923077
max_iter=500	0.923077	0.903846	0.923077

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.583333	0.5	0.583333
max_iter=200	0.583333	0.5	0.583333
max_iter=500	0.583333	0.5	0.583333

Figura 11 Accuracy e Precision – kernel = poly

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.913462	0.971154	0.971154
max_iter=200	0.913462	0.971154	0.971154
max_iter=500	0.913462	0.971154	0.971154

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.545455	0.888889	0.888889
max_iter=200	0.545455	0.888889	0.888889
max_iter=500	0.545455	0.888889	0.888889

Figura 12 Accuracy e Precision – kernel = rbf

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.913462	0.923077	0.903846
max_iter=200	0.913462	0.923077	0.903846
max_iter=500	0.913462	0.923077	0.903846

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.545455	0.583333	0.5
max_iter=200	0.545455	0.583333	0.5
max_iter=500	0.545455	0.583333	0.5

Figura 13 Accuracy e Precision – kernel = sigmoid

4.1.3. Cluster 2 [Giorni festivi]

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.75	0.75	0.769231
max_iter=200	0.75	0.75	0.769231
max_iter=500	0.75	0.75	0.769231

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.333333	0.333333	0.363636
max_iter=200	0.333333	0.333333	0.363636
max_iter=500	0.333333	0.333333	0.363636

Figura 14 Accuracy e Precision – kernel = poly

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.903846	0.865385	0.884615
max_iter=200	0.903846	0.865385	0.884615
max_iter=500	0.903846	0.865385	0.884615

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.666667	0.6	0.636364
max_iter=200	0.666667	0.6	0.636364
max_iter=500	0.666667	0.6	0.636364

Figura 15 Accuracy e Precision – kernel = rbf

Accuracy:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.788462	0.788462	0.788462
max_iter=200	0.788462	0.788462	0.788462
max_iter=500	0.788462	0.788462	0.788462

Precision:

	gamma=scale	gamma=auto	gamma=0.05
max_iter=100	0.4	0.4	0.4
max_iter=200	0.4	0.4	0.4
max_iter=500	0.4	0.4	0.4

Figura 16 Accuracy e Precision – kernel = sigmoid

5. CONCLUSIONI

Alla luce dei dati sopra riportati è indubbia la bontà del kernel rbf, si ottengono valori molto soddisfacenti in termini di accuratezza e precisione su tutti i cluster presenti, l'invarianza dei dati ottenuti al variare del parametro max_iter è segno di un problema non molto complesso e quindi aumentando il numero di iterazioni non otteniamo effettivi vantaggi nell'apprendimento del problema da parte dell'algoritmo. Il parametro gamma, coefficiente del kernel, in alcuni cluster si ottengono risultati migliori con il valore *scale*, in altri casi con il valore *auto*, spesso questi si sovrappongono con gamma settato a 0.05, in ogni caso i risultati sono buoni con ogni parametro gamma preso in considerazioni, con accuratezza che raggiunge picchi di 0.97 e precisione 0.88 (nel cluster 1).

È interessante notare che l'accuratezza, con il parametro gamma corretto, non scende mai al disotto di 0.9038 questo ci fa intuire che il la scelta di sfruttare un algoritmo come quello del One Class Support Vector Machine che sfrutta iperpiani per separare dati anomali da dati normali è stata una scelta vincente per risolvere il problema della ricerca delle anomalie.

6. APPENDICE A

Manuali utilizzati:

- sklearn: https://scikit-learn.org/stable/user_guide.html
- pandas: https://pandas.pydata.org/pandas-docs/stable/reference/index.html
- nunpay: https://numpy.org/doc/stable/
- matplotlib: https://matplotlib.org/contents.html

Sitografia:

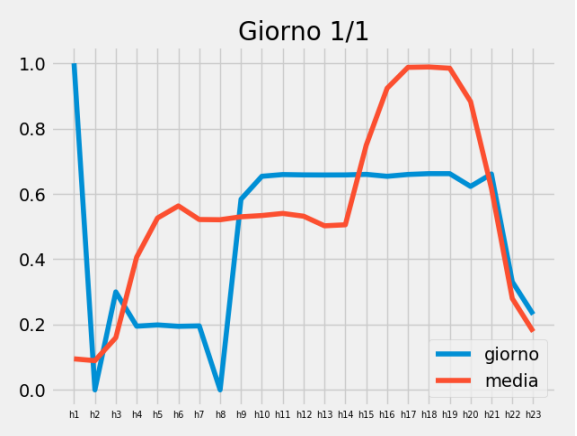
- accuracy – precision: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
- https://en.wikipedia.org/wiki/Accuracy_and_precision

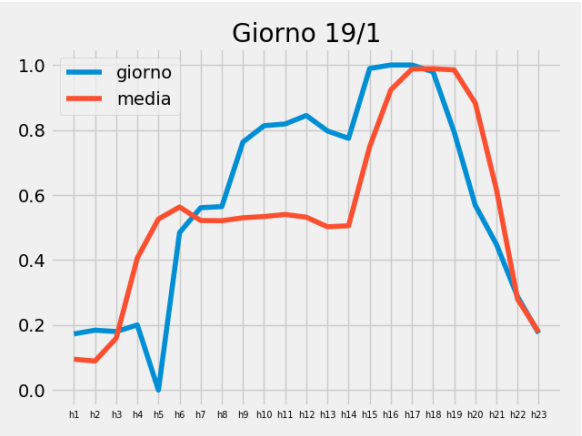
7. APPENDICE B

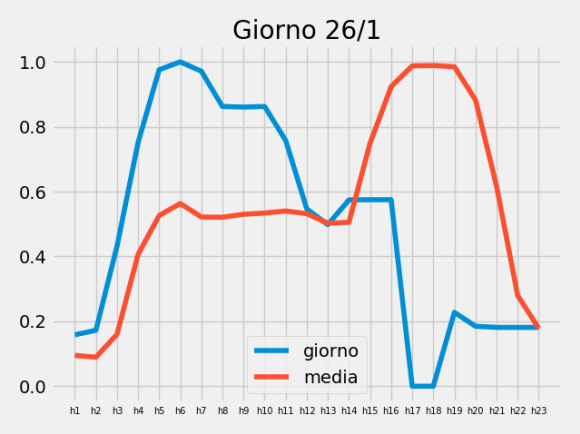
Vengono allegati in questa appendice i grafici delle time series dei giorni anomali trovati nel capitolo 3, per completezza di informazioni.

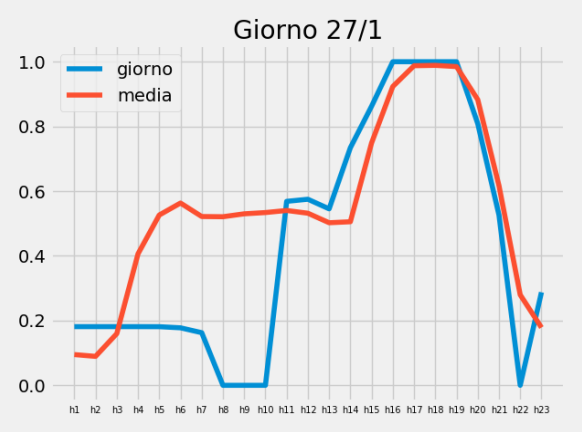
I seguenti grafici, mostrano sulle ascisse la suddivisione dei dati a nostra disposizione, cioè il vettore di ventitré dimensioni che rappresenta le ore di lavoro della società di taxi di New York presa in esame, mentre sulle ordinate i suoi rispettivi valori.

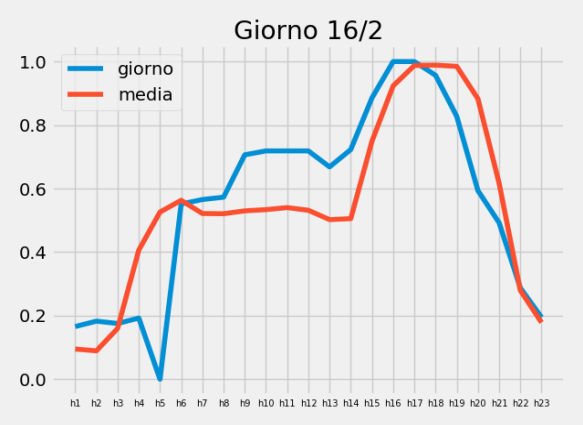
Cluster 0:

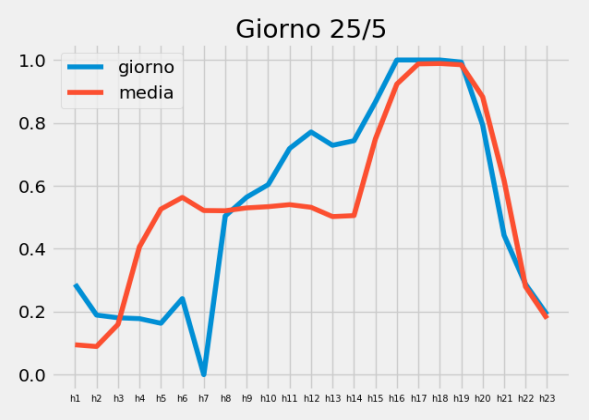


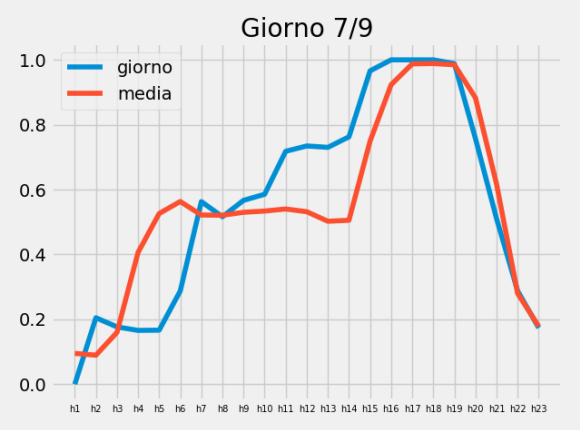


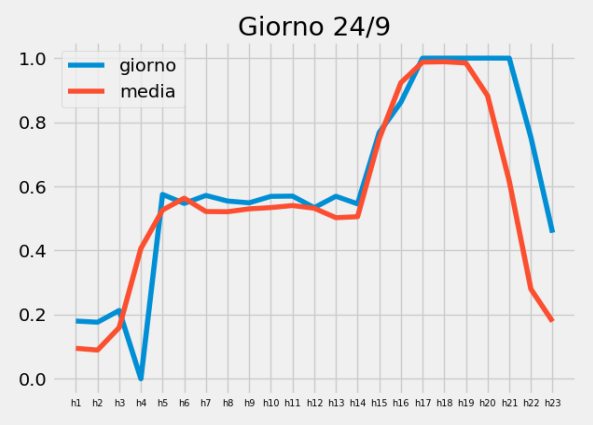

Giorno 26/1
giorno
media

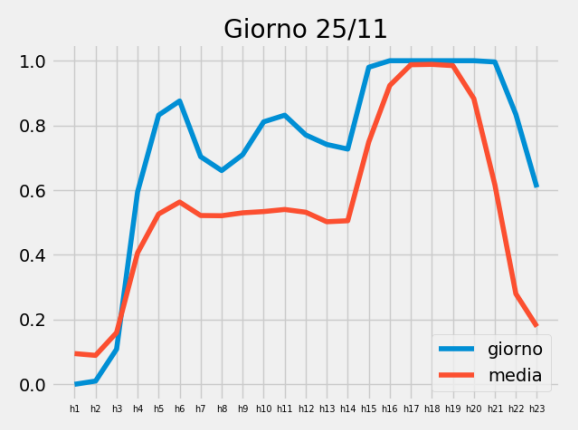

Giorno 27/1
giorno
media

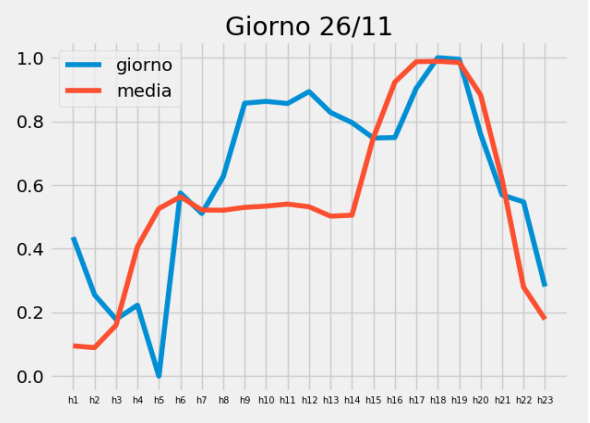

Giorno 16/2
giorno
media

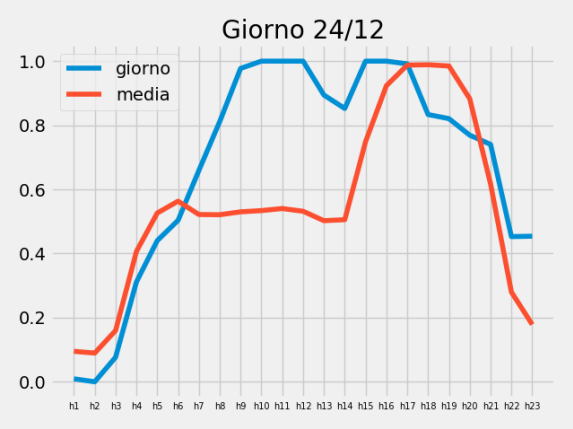

Giorno 25/5
giorno
media

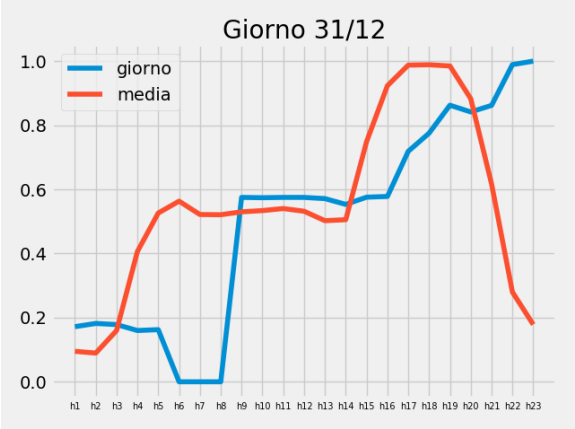

Giorno 7/9
giorno
media


Giorno 24/9
giorno
media

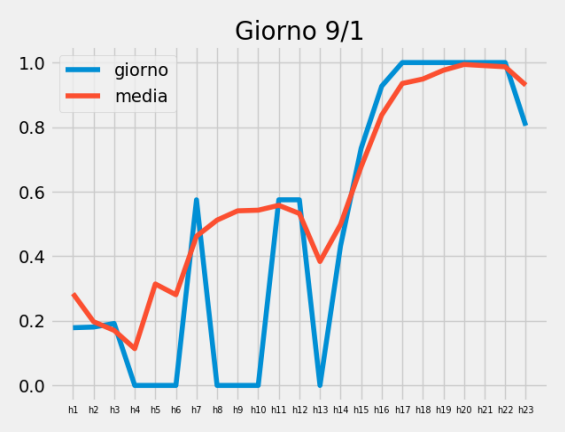

Giorno 25/11
giorno
media

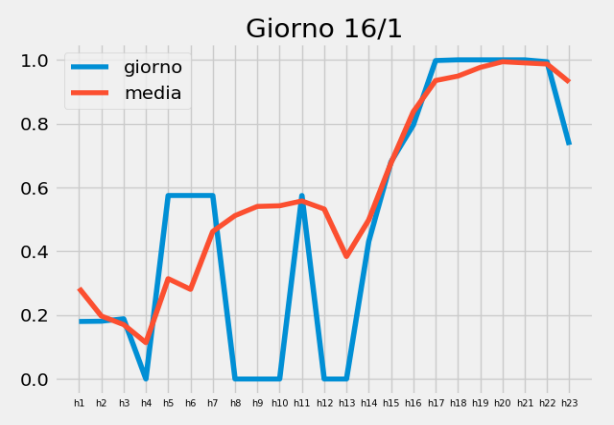

Giorno 26/11
giorno
media

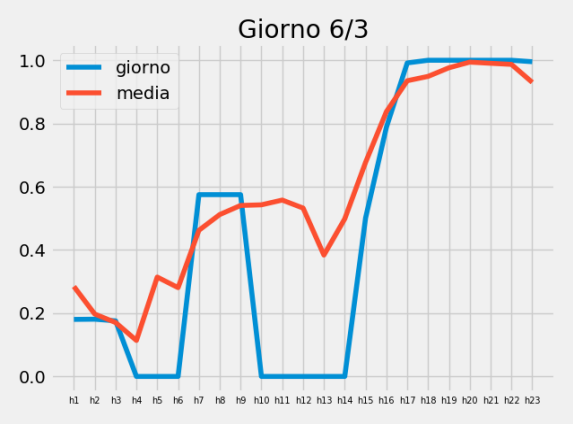

Giorno 24/12
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

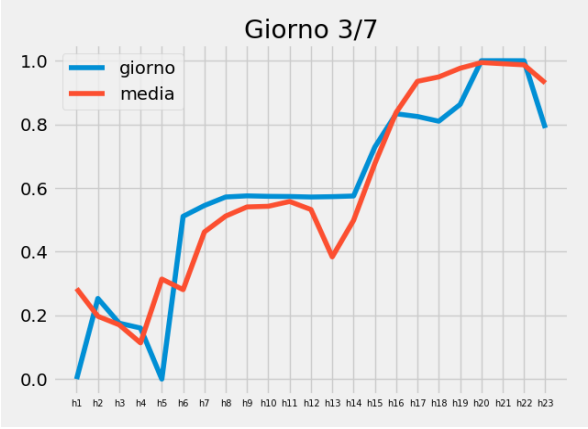

Giorno 31/12
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

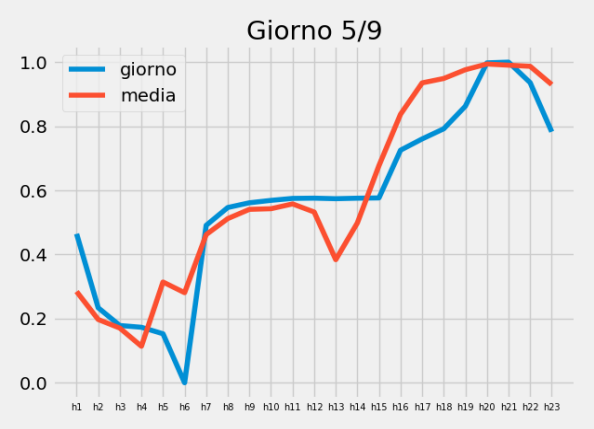
Cluster 1:

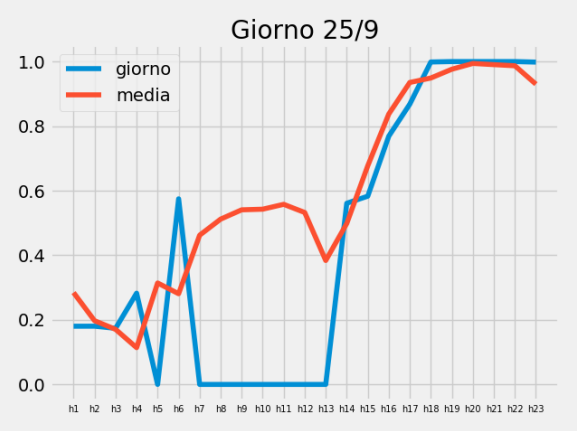

Giorno 9/1
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

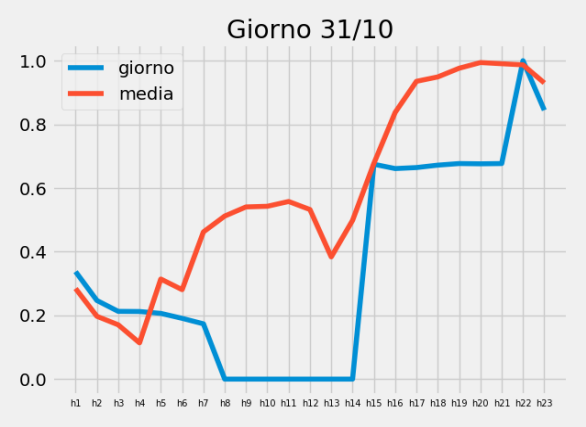

Giorno 16/1
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

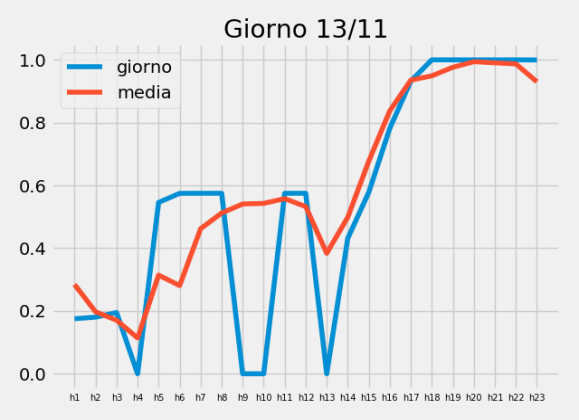

Giorno 6/3
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

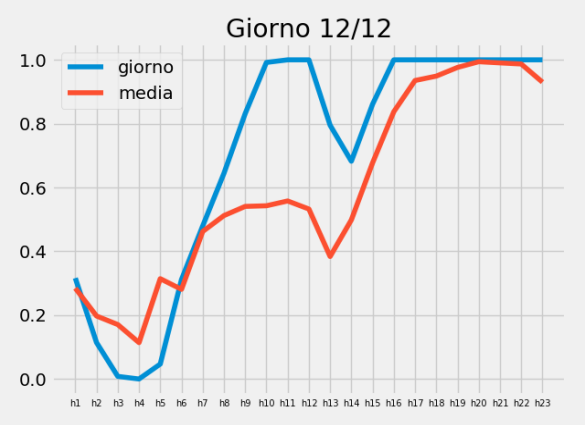

Giorno 3/7
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

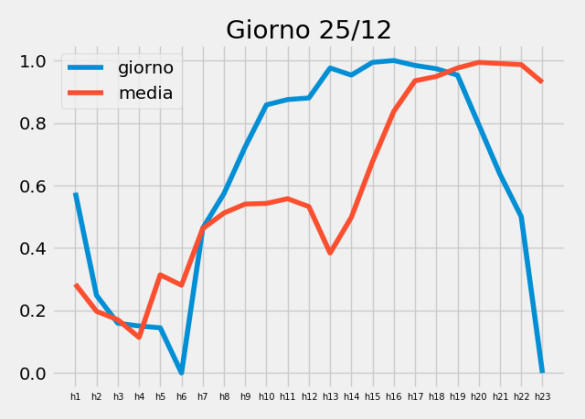

Giorno 5/9
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23


Giorno 25/9
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

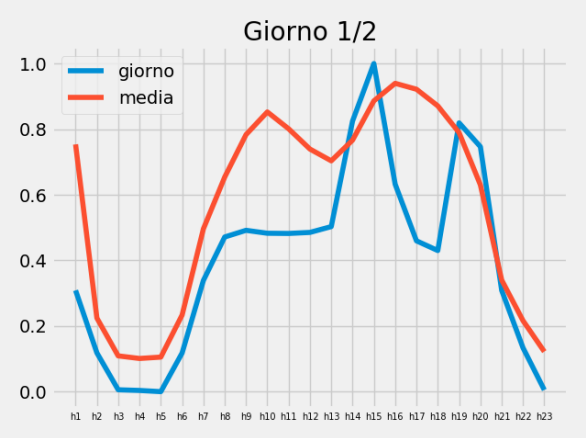

Giorno 31/10
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

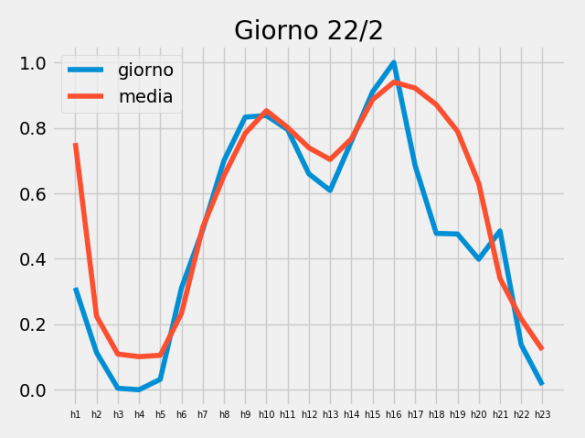

Giorno 13/11
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

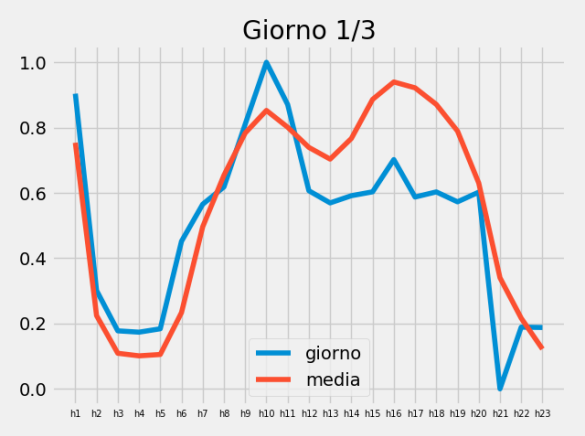

Giorno 12/12
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

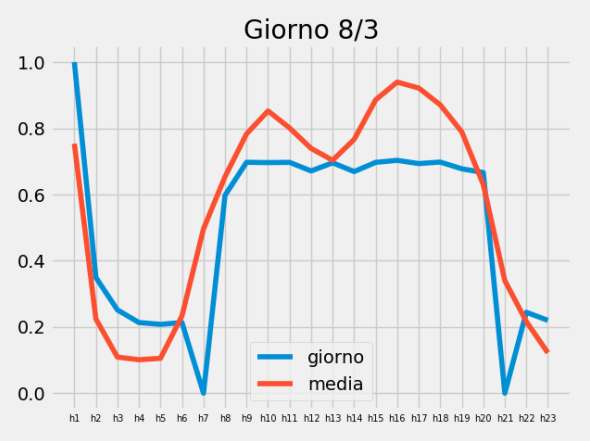

Giorno 25/12
giorno
media
1.0
0.8
0.6
0.4
0.2
0.0
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23

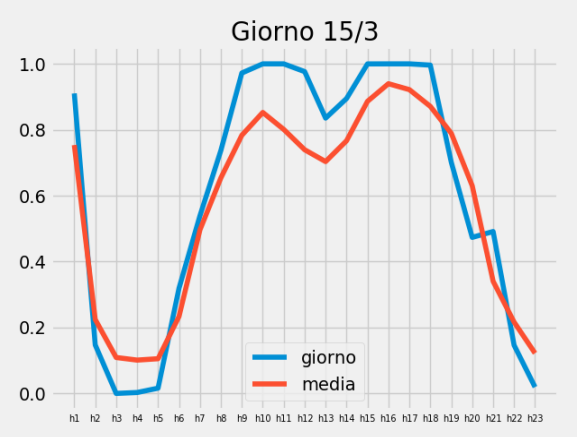
Cluster 2:

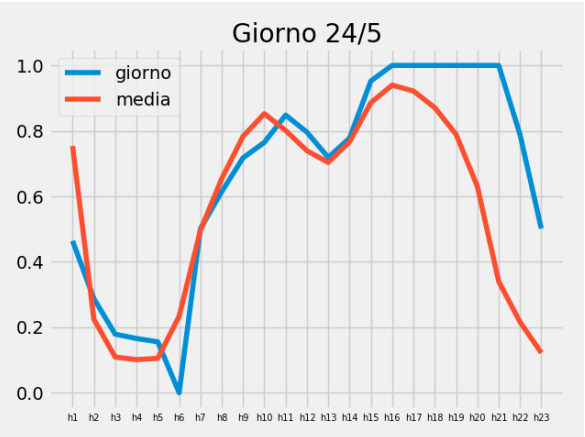

Giorno 1/2
giorno
media

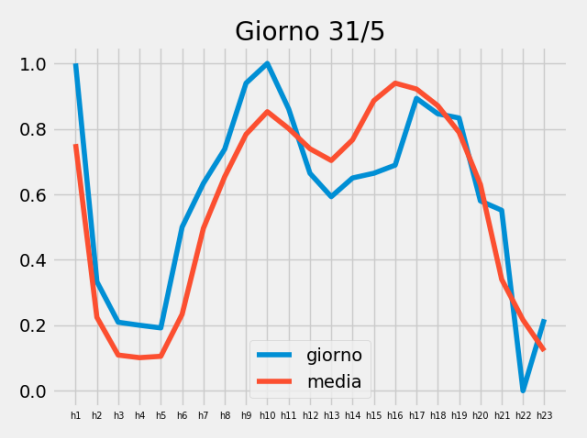

Giorno 22/2
giorno
media

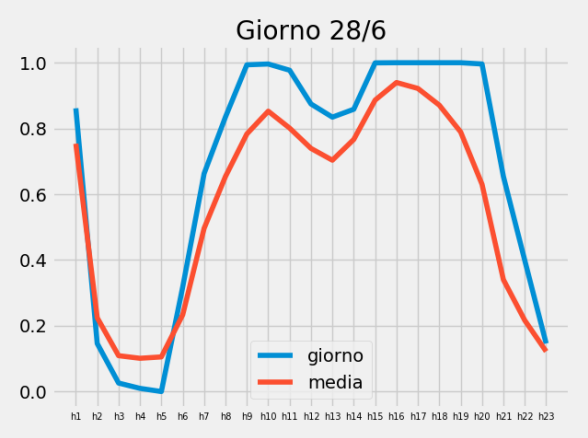

Giorno 1/3
giorno
media

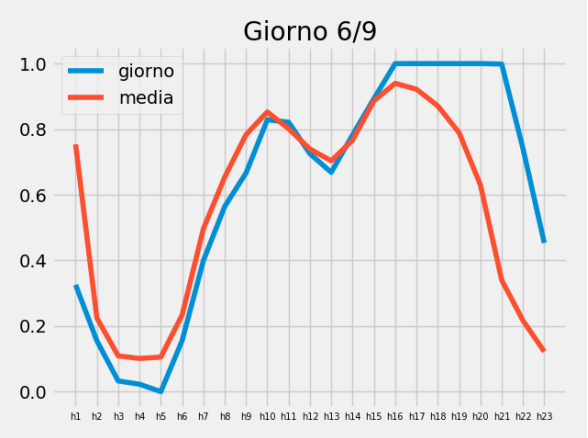

Giorno 8/3
giorno
media


Giorno 15/3
giorno
media


Giorno 24/5
giorno
media







Facile vedere come tutte queste time series (linee tratteggiate blu) sono distanti per comportamento e/o per valori dalla media (linea tratteggiata rossa).

8. BIBLIOGRAFIA

- Antonio L. Alfeo, Mario G.C.A. Cimino, Sara Egidi, Bruno Lepri, Gigliola Vaglini: A stigmergy-based analysis of city hotspots to discover trends and anomalies in urban transportation usage.

- Huichu Zhang, Yu Zheng, Yong Yu: Detecting urban anomalies using Multiple Spatio-Temporal Data sources

- Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, John Platt: Support Vector Machine for Novelty Detection

9. RINGRAZIAMENTI

In questo capitolo voglio ringraziare chi mi ha aiutato e mi è stato vicino in questo periodo.

I primi ringraziamenti sono per i miei relatori, tutti estremamente pronti ad aiutarmi, veloci ed efficaci nelle risposte, un privilegio poter svolgere quest'esperienza con loro.

Devo ringraziare adesso i miei amici tutti, quelli che conosco da sempre e quelli che ho conosciuto a Pisa, perché nonostante tutto ci siamo sempre fatti compagnia senza mai sentirci soli.

Voglio ringraziare Roberta, la mia compagna, che mi è stata vicina sempre, dall'inizio alla fine, che mi ha aiutato nei momenti peggiori ed era lì nei momenti migliori, ed è stata un supporto sul quale ho sempre potuto fare affidamento.

Infine ma non per ordine di importanza voglio ringraziare la mia famiglia, in particolar modo i miei genitori e mia sorella, che mi hanno supportato in ogni modo ed in ogni momento, e mi hanno permesso di realizzare il mio primo grande obiettivo e so che potrò contare su di loro anche per i prossimi.

GRAZIE.