**Progetto Data Mining II**

**“Air Quality UCI**”

**Introduzione**

“Air Qality UCI” è un dataset contenente informazioni sulla quantità di agenti inquinanti presenti nell’aria.

È composto da 9471 righe, ognuna contenente le seguenti informazioni:

* Data *Date*
* Orario della misurazione *Time*
* Concentrazione oraria di monossido di carbonio *CO(GT)*
* Quantità media oraria di ossido di stagno *PT08.S1(CO)*
* Quantità media oraria di idrocarburi diversi dal metano *NMHC(GT)*
* Concentrazione media oraria di benzene *C6H6(GT)*
* Quantità media oraria di ossido di tungsteno *PT08.S2(NMHC)*
* Quantità media oraria di tutti gli ossidi di azoto e le loro miscele *NOx (GT)*
* Quantità media oraria di ossido di tungsteno rispetto agli ossidi di azoto *PT08.S3(NOx)*
* Concentrazione oraria di biossido di azoto *NO2(GT)*
* Quantità media oraria di ossido di tungsteno rispetto al biossido di azoto *PT08.S4(NO2)*
* Quantità media oraria di ossido di indio *PT08.S5(O3)*
* Temperatura *T*
* Umidità relativa *RH*
* Umidità assoluta *AH*

Il dataset “grezzo” si presenta con le ultime 134 righe interamente di Nan, così come le due ultime due colonne chiamate “Unnamed:15” e “ ‘’’’ “.

Il lavoro di analisi di questo dataset si organizza in 4 tasks:

1. Time series: considerando solo l’attributo “PT08.S1(CO)” dividere il dataset in serie giornaliere di 24 ore, cancellando quelle contenti troppi missing values (valore= -200) e gestendo i restanti in qualche modo; assicurarsi che, alla fine, ogni time series abbia 24 valori. Effettuare, poi, il clustering con un algoritmo utilizzando prima la DTW e poi l’Euclidean Distance, confrontandone i risultati.
2. Sequential patterns: dopo aver discretizzato le serie, cercare sequential patterns contigui di lunghezza minima 4.
3. Classification methods: definire la variabile obiettivo “WE” per le serie, che assuma valore *true* se il giorno della misurazione è un giorno del week-end, *false* altrimenti. Utilizzare il metodo di classificazione k-NN usando la DTW come misura della distanza, e almeno un altro metodo di classificazione che usi i 24 valori come variabili distinte.
4. Outlier detection: utilizzando il dataset “grezzo” dopo aver eliminato ogni missing values, identificare l’1% dei records “più outlier” utilizzando due metodi differenti appartenenti a due famiglie diverse. Confrontare i risultati ottenuti.

**Task 1: Time series**

Il dataset da analizzare contiene registrazioni orarie di valori misuranti il livello di inquinamento nell’aria, per un arco temporale che va dalle ore 18:00 del 10/03/2004 alle ore 14:00 del 4/4/2005.

Per la prima parte del task, si è effettuato il drop di tutti i Nan e, una volta verificato che nell’operazione di drop non si fossero creati “buchi orari” e provveduto alla trasformazione degli orari in numeri interi per una più veloce gestione dei dati, si è provveduto al drop delle misurazioni riguardanti il primo e l’ultimo giorno, in quanto queste non coprivano per intero l’arco delle 24 ore. Si è così ottenuto un dataset contenete misurazioni per un totale di 389 giorni.

Le misurazioni riguardanti i 389 giorni sono state organizzate in time series giornaliere e si è proceduto al conteggio dei missing values contenuti in ogni singola time series.

Stabilita in due la quantità massima di missing values ammessa all’interno della stessa time series, è stato effettuato il drop di quelle che ne contenevano di più e sono stati sistemati quelli contenuti nelle restanti; il metodo di correzione applicato è stato quello di sostituire questi con la media aritmetica fra le misurazioni riferite all’orario immediatamente precedente e a quello immediatamente successivo.

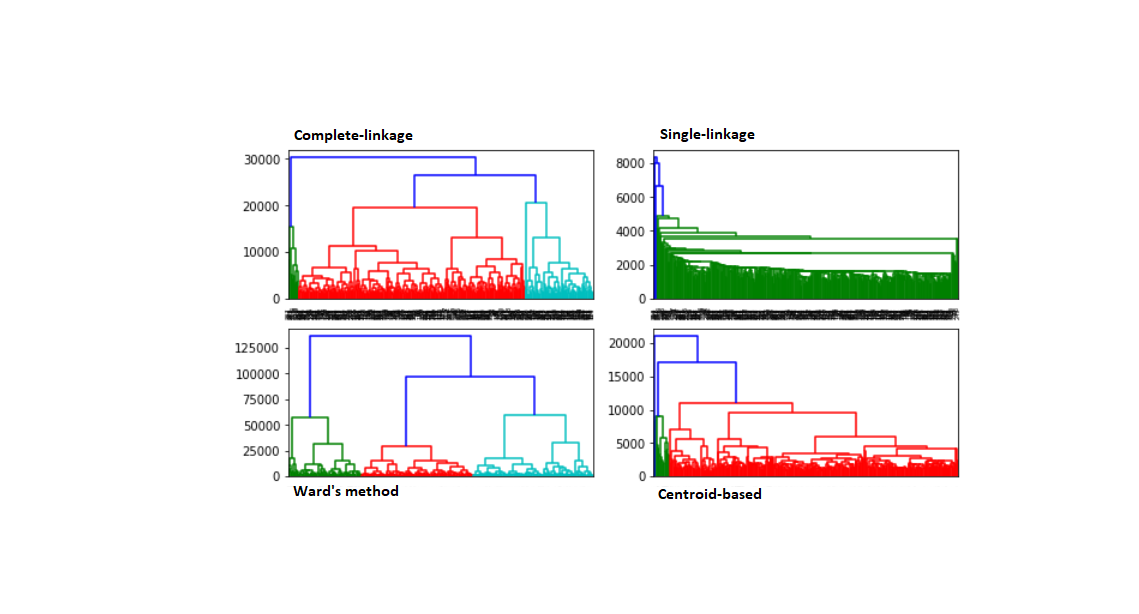
Il dataset si presenta alla fine composto da 364 time series, ognuna contenente 24 misurazioni.

È stato successivamente effettuato il clustering gerarchico basato dapprima sulla DTW e successivamente la distanza euclidea.

Per questa operazione sono stati utilizzati 4 metodi di clustering, al fine di confrontarne i risultati e poter scegliere quindi il metodo che offriva i cluster migliori.

I metodi sono: single-linkage, complete-linkage, Ward’s method e il centroid-based.

I dendogrammi ottenuti utilizzando la DTW sono:



Mentre, quelli ottenuti utilizzando l’euclidean distance sono:

Immagine che contiene screenshot

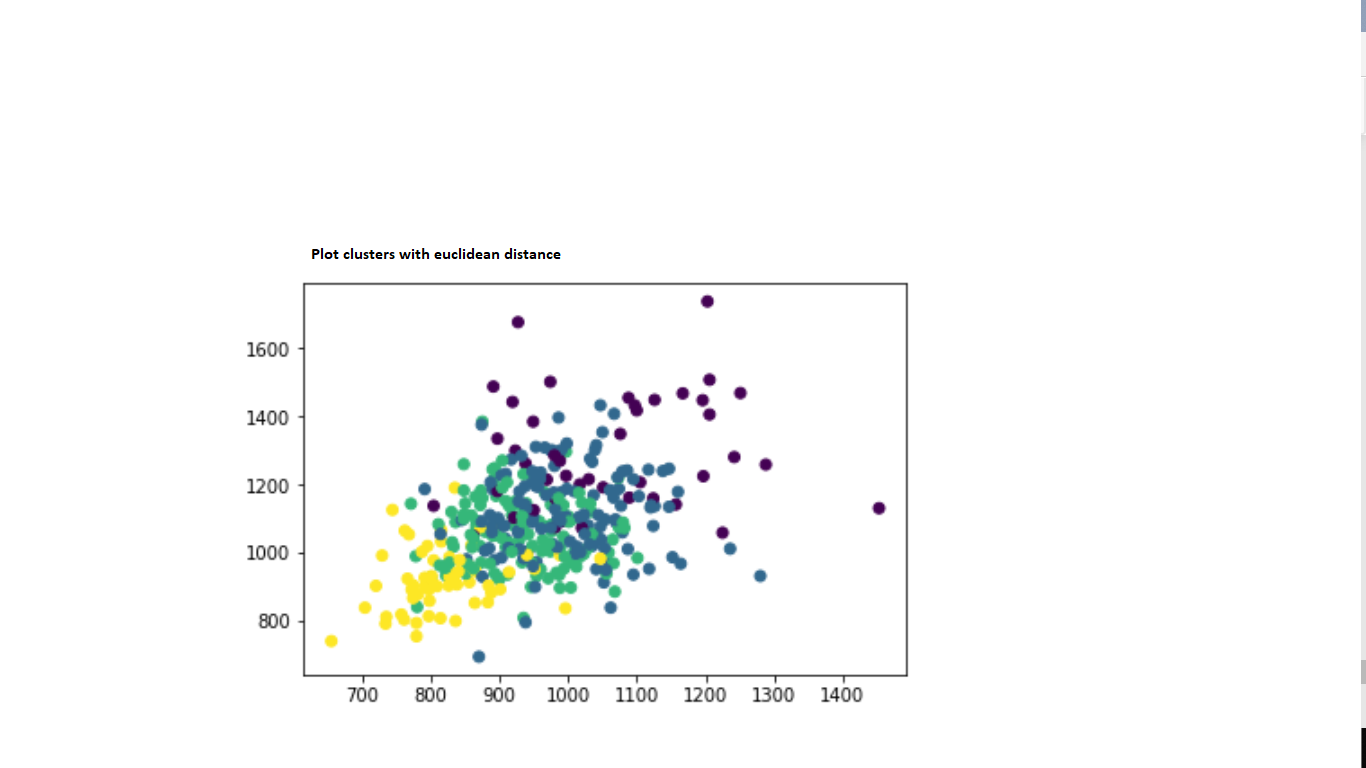
Descrizione generata automaticamente

Come si puo’ notare dai dendogrammi di entrambi i metodi, il metodo ward offre dei cluster meglio definiti, motivo per il quale si è scelto di ignorare gli altri 3 metodi e procedere al confronto fra i cluster così ottenuti.

Per il numero di cluster da considerare si è prima osservato dove il dendogramma fosse meglio definito e si è poi optato per il numero di cluster che, con entrambi i tipi di distanza, avesse la percentuale di coincidenza più alta.

Il risultato di questa operazione è stato lo scegliere 4 cluster, che offrivano una percentuale di coincidenza pari al 72.8%.

Immagine che contiene screenshot

Descrizione generata automaticamente

**Task 2: Sequential patterns**

Lo scopo di questa task è l’individuazione di frequent patterns contigui di lunghezza minima pari a 4.

Per l’obiettivo si è provveduto a discretizzare le time series con l’algoritmo SAX in 12 intervalli di ampiezza 116.083.

Le time series discretizzate sono state successivamente analizzate con l’algoritmo Fourier08/closed+time, che richiede l’inserimento dei seguenti constraints:

* Min sup = numero minimo di time series che devono contenere il sequential pattern,
* Min\_time interval = intervallo minimo di tempo che deve intercorrere tra due elementi del sequential pattern,
* Max\_time interval = intervallo massimo di tempo che deve intercorrere tra due elementi del sequential pattern,
* Min\_whole interval = intervallo minimo di tempo che deve intercorrere tra il primo e l’ultimo elemento del sequential patter,
* Max\_whole interval = intervallo massimo di tempo che deve intercorrere tra il primo e l’ultimo elemento del sequential patter

Il min\_time interval e il max\_time interval impostati a 1 garantiscono che gli elementi del sequential pattern trovato dall’algoritmo siano contigui fra loro, mentre il min\_whole interval impostato a 3 garantisce che che i sequential pattern trovati abbiano le caratteristiche richieste dal task.

L’algortimo è stato lanciato 3 volte con, rispettivamente, i seguenti constraints:

-Minsup: 20% - Minsup: 15% -Minsup:10%

-Min time int:1 - Min time: 1 - Min time int: 1

-Max time int:1 - Max time int: 1 -Max time int: 1

-Min Whole time int:3 -Min whole int:3 -Min whole int: 4

-Max Whole time int :24 -Max whole int:24 -Max whole int: 24

Gli ultimi settaggi sono stati impostati per cercare frequent patterns di lunghezza maggiore, approfittando del valore basso di support minimo richiesto.

I risultati dei tre settaggi sono, rispettivamente:

1. <0> 0 -1 <1> 0 -1 <2> 0 -1 <3> 1 -1 #SUP: 99

<0> 1 -1 <1> 0 -1 <2> 0 -1 <3> 0 -1 #SUP: 106

2) <0> 0 -1 <1> 0 -1 <2> 0 -1 <3> 1 -1 #SUP: 99

<0> 0 -1 <1> 0 -1 <2> 0 -1 <3> 0 -1 #SUP: 64

<0> 1 -1 <1> 1 -1 <2> 0 -1 <3> 0 -1 #SUP: 69

<0> 1 -1 <1> 0 -1 <2> 0 -1 <3> 0 -1 #SUP: 106

3) <0> 1 -1 <1> 1 -1 <2> 0 -1 <3> 0 -1 <4> 0 -1 #SUP: 38

<0> 1 -1 <1> 0 -1 <2> 0 -1 <3> 0 -1 <4> 1 -1 #SUP: 51

Gli elementi dei frequent pattern sono presentati con l’indice (fra <>) e il valore; ogni valore è separato dagli altri da -1.

Possiamo notare come i frequenti pattern riguardino sempre le prime 4 e 5 ore della giornata (quindi l’arco di tempo che va dalle 00:00 alle 04:00/05:00) e siano sempre molto simili fra di loro.

**Task 3: Classification methods**

**K-NEAREST NEIGHBORS (K-NN)**

Per lo svolgimento di questa task, si è partito dalla definizione di una variabile di tipo booleano WE che assume valore *true* per le time series che raccolgono le misurazioni nei giorni del week-end, *false* altrimenti. Successivamente si è suddiviso il dataset in train set e test set: il primo raccoglie le prime 300 time series, il secondo le restanti 64.

Il K-NN è un algoritmo distance-based per la classification la cui idea di base è: dato un record da classificare, si prendono i k punti più vicini a quello dato e si classifica questo con il label più presente fra i k presi in considerazione.

Impostato il K-NN, per la scelta del parametro k sono state effettuate le prove per tutti i valori di k compresi fra 1 e 10; la scelta è successivamente ricaduta sul valore

K = 5, in quanto garantiva prestazioni superiori agli altri casi, nello specifico la confusion matrix sul test set è risultata essere:

|  |  |  |
| --- | --- | --- |
|  | True  (actual class) | False  (actual class) |
| True  (predicted class) | 39 | 5 |
| False  (predicted class) | 4 | 16 |

Dal quale si ottengono i seguenti risultati:

* Accuracy = 85.9375%
* Precision = 88.6364%
* Recall = 90.6977%
* F-1 score = 89.6552%

**ARTIFICIAL NEURAL NETWORK (ANN)**

L’ANN è un metodo di classification che si basa su nodi interconessi fra loro e weighted links. I nodi sono organizzati generalmente in 3 layers: il primo raccoglie gli input, il secondo (detto hidden layer) i cosiddetti perceptron, “neuroni artificiali” che rielaborano gli input secondo una activation function, il terzo layer contiene gli output, ossia le informazioni contenute nell’hidden layer rielaborate secondo i weighted links.

Nel caso specifico di questa analisi sono stati utilizzati due algoritmi: il multi-layer perceptron classifier (MLPClassifier) per la costruzione del modello di apprendimento e il GridSearchCV che adatta il modello di apprendimento al test set e permette di visualizzare i risultati.

La confusion matrix e i risultati ottenuti dall’applicazione di questo metodo sul train set sono i seguenti:

|  |  |
| --- | --- |
| 170 | 25 |
| 6 | 72 |

Accuracy = 88.6446%

Precision = 87.1795%

Recall = 96.5909%

F-1 score = 91.6442%

I risultati ottenuti sul test set, invece, risultano essere:

|  |  |
| --- | --- |
| 52 | 9 |
| 6 | 24 |

Accuracy = 83.5164%

Precision = 85.2459%

Recall = 89.6552%

F-1 score = 87.3949%

Dai risultati ottenuti si può evincere come, nel caso specifico, il K-NN sia stato un classificatore migliore in quanto ottiene percentuali migliori qualsiasi risultati si osservi.

**Task 4: Outlier**

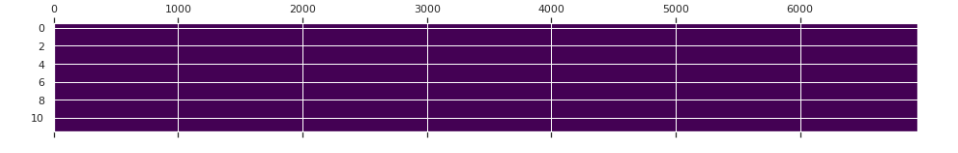
Per la ricerca degli outlier si è provveduto a ripulire il dataset da tutti i missing value presenti.

Prima della “pulizia”, il dataset presentava 9357 records e plottando il dataset evidenziando i missing values si otteneva:

Immagine che contiene dispositivo

Descrizione generata automaticamente

A seguito della “pulizia, il dataset contava 6941 records e il plot non evidenzia più missing values



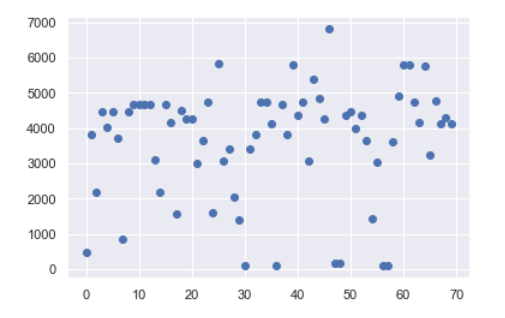
Il top 1% degli outliers saranno quindi 70 punti.

La ricerca degli outliers è avvenuta tramite l’utilizzo del DBScan e Nearest Neighbors (NN), di tipo distance- based, e il LOF, di tipo density -based.

**NN**

Per la ricerca degli outliers tramite NN si è calcolata la distanza per ogni punto da ogni punto, per poi plottare le distanze minime.

Si considereranno quindi nel top 1% degli outliers i 70 punti che risultano avere la distanza minima più alta.



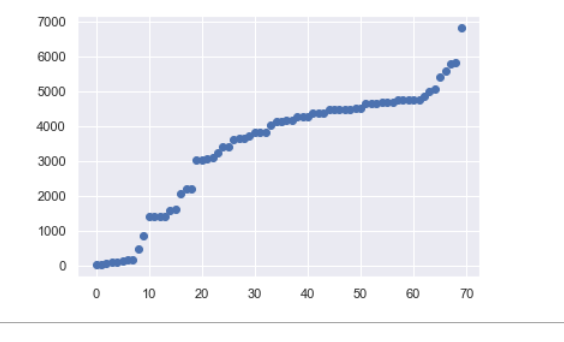
Dal plot si può osservare che gli outlier si concentrano nell’intervallo di misurazioni di indici compresi fra 4200 e 4800 e nell’intervallo 5400 e 5800; da un punto di vista temporale questi intervalli corrispondono al mese di settembre e novembre.

**DBScan**

Il DBScan è un algoritmo che, dato un raggio e un numero minimo di punti, mette in evidenza tutti i punti che, entro il raggio, non possiedono il numero minimo di punti richiesto.

Per la ricerca degli outliers tramite DBScan, si è lanciato l’algoritmo più volte per la ricerca dell’epsilon ottimale; impostato il min\_samples a 3, si è individuato in 192 il raggio ottimo in quanto, con questo valore, l’algoritmo etichettava come outliers i 70 punti da noi richiesto.

Plottando i punti trovati dall’algoritmo otteniamo:



**LOF**

Il LOF è un algoritmo che basa la ricerca degli outliers sul concetto di densità locale, calcolata sulle distanze relative dei k punti più vicini a quello preso in considerazione; i punti a più bassa densità locale sono considerati outliers.

Definito il k = 20 e plottati i punti individuati dall’algoritmo otteniamo:

Immagine che contiene esterni, cielo

Descrizione generata automaticamente

Notiamo come rispetto ai precedenti due metodi, i punti trovati sono molto più sparsi.

**Confronto fra i tre metodi**

Confrontando i risultati ottenuti dai 3 algoritmi notiamo che:

* I punti individuati da DBSCan e NN coincidono per il 75.71% dei punti (53/70)
* I punti individuati da DBSCan e LOF coincidono per il 48.57% dei punti (34/70)
* I punti individuati da LOF e NN coincidono per il 50% dei punti (35/70)

Possiamo inoltre notare che se per i due distance-based i punti sono molto simili (come ci si poteva aspettare), il LOF, nonostante la natura diversa dell’algoritmo, ha comunque evidenziato, in comune con gli altri due, una concentrazione di outliers nelle misurazioni comprese nell’intervallo 4200-4800, che coincidono con le misurazione prese lungo il mese di settembre, sottolineando come le emissioni di agenti inquinanti in quel periodo siano più intense rispetto agli altri mesi.