Social and Behavioral analysis in a Smart City

Progetto per il corso di Big Data della Facoltà di Ingegneria Informatica di Roma Tre

https://github.com/MARMANGIONE/AnalysisOfASmartCity.git

Versione documento: 3.0

Data	Autore
24/07/2019	Martina Mangione

Indice

1. Introduzione

La crescente disponibilità di grandi quantità di dati ha dato luogo a sfide di ricerca ambiziose in molti campi, che vanno dal mondo finanziario e commerciale, alle persone e al monitoraggio ambientale. Mentre le tradizionali fonti di dati e il censimento falliscono nel catturare comportamenti attuali e aggiornati, i Big Data integrano le conoscenze mancanti fornendo informazioni utili e nascoste agli analisti e ai responsabili delle decisioni.
Con questo progetto, intendo concentrarmi sull'analisi dei dati dei dispositivi mobili ossia i cellulari (Call Detail Record) studiando e valutando l'impatto che le previsioni meteo hanno su di essi e sulle abitudini dei cittadini di Milano.

1.1 Obiettivo del progetto

Gli obiettivi che ho cercato di pormi con il seguente progetto sono i seguenti:

1. **Data Analysis con Hive e Spark:**
   - Mappatura delle nazionalità di chi utilizza dei telefoni cellulari
   - Osservazione dei periodi di congestione a livello di comunicazione
   - Monitoraggio di eventi su larga scala
2. **Elaborazione dei dati:** Correlazione tra le attività svolte dall'utente (telefonate) con le diverse condizioni metereologiche
3. **Machine Learning con MLlib:** Previsione della domanda di energia.

1.2 Asset Tecnico e Ambiente di Sviluppo

Per il progetto sono stati utilizzati i seguenti dispositivi:

1. MacBook Air:
- **Memoria:** 8 GB 1600 MHz DDR3
- **Processore:** 1,4 GHz Intel Core i5

2. MacBook Pro:
- **Memoria:** 16 GB 2133 MHz LPDDR3
- **Processore:** 2,7 GHz Intel Core i7

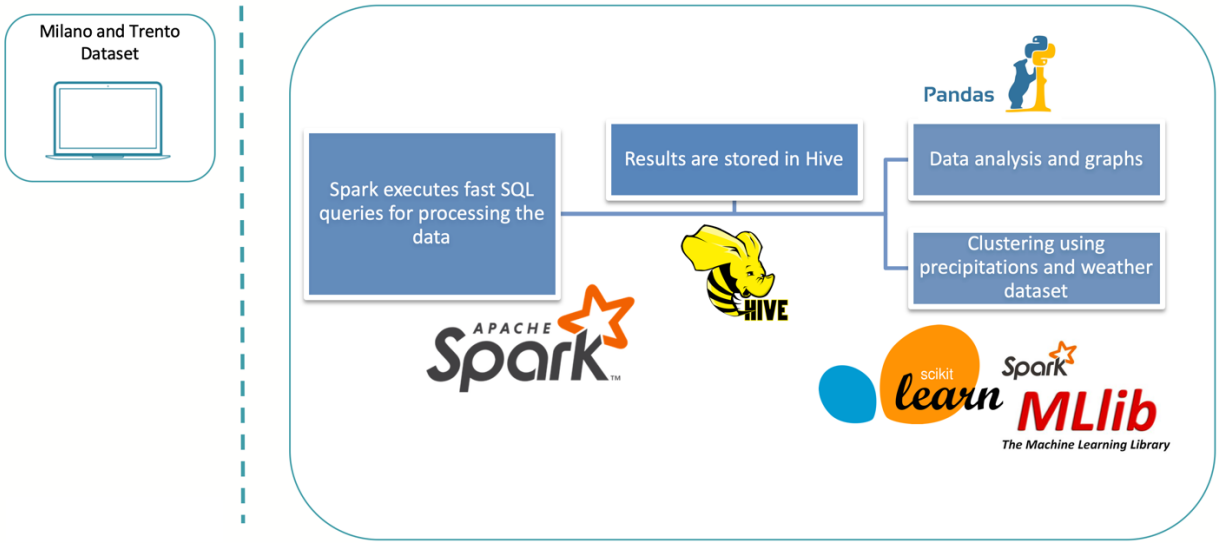
Gli IDE utilizzati sono invece i seguenti:

- PyCharm 1.1 con Python v3.7 e PySpark v2.4.3

Le alter dipendenze installate sono:

- matplotlib 3.1.1, numpy 1.16.4, pandas 0.24.2, geojson 2.4.1, seaborn 0.9.0

2. Architettura del sistema



Per questo progetto è stato deciso di operare in locale con l'asset descritto al paragrafo 1.2 utilizzando per la data ingestion Spark . I dati recuperati vengono filtrati e pre-elaborati attraverso delle query con HiveQL.
Grazie all'utilizzo di Pandas è stato possibile realizzare dei grafici a dimostrazione delle tesi portate in ogni capitolo.

Sono state adoperate, infine tecniche, statistiche non parametriche (Kruskal-Wallis) attraverso sci-kit learn e di machine learning grazie all'utilizzo di MLlib.

3. Il Dataset

Il dataset è composto da dati inerenti alle telecomunicazioni, il meteo, e il consumo di corrente elettrica nella città di Milano e nella provincia di Trento. La composizione multi-sorgente lo rende un banco di prova ideale per metodologie e approcci finalizzati ad affrontare una vasta gamma di problemi tra cui consumo di energia, strutture e interazioni urbane, rilevamento di eventi e molti altri… I dati sono accessibili da un'API pubblica fornita da Dandelion (http://dandelion.eu), Open Big Data (https://dandelion.eu/datamine/open-big-data/) utilizzando un account personale creato nel sito Web.

3.1. Composizione geografica dei dati

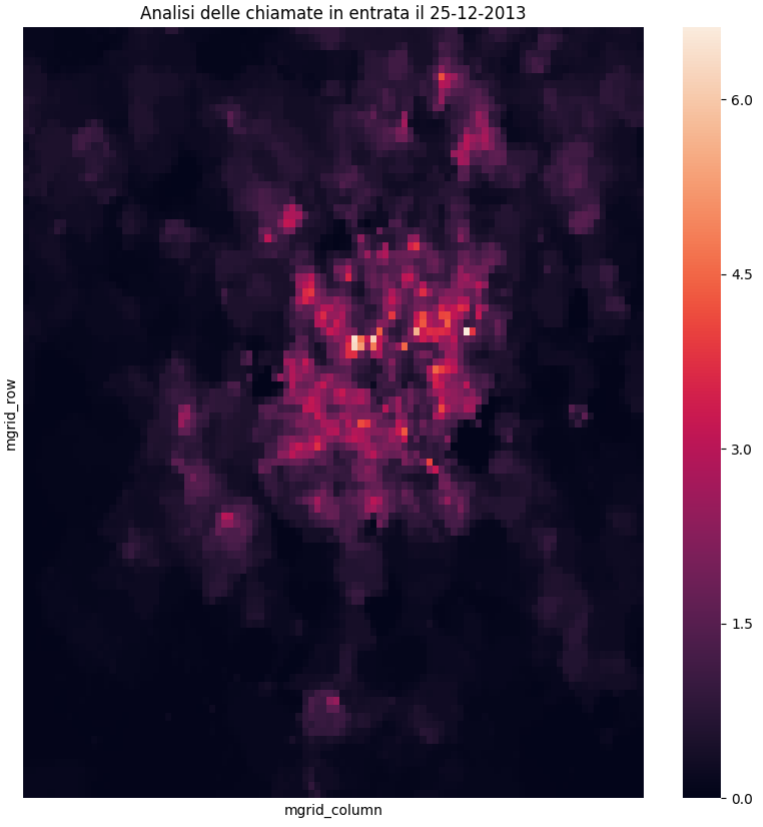
9901	9902	···	9999	10000
9801	···	···	9899	9900
···	···	···	···	···
101	102	···	···	200
1	2	3	···	100

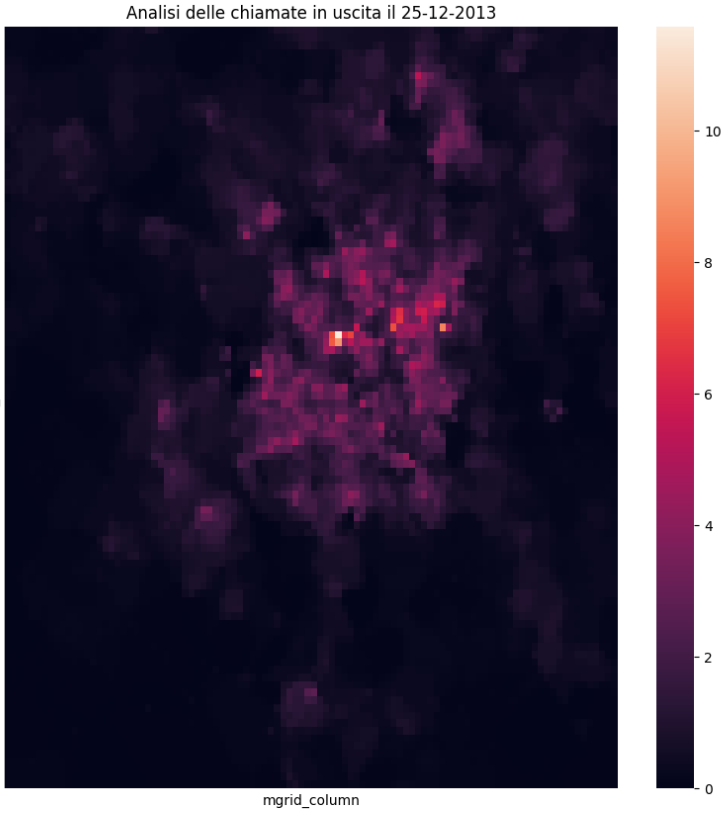
Griglia rappresentante l'area di Milano

Poiché i set di dati provengono da varie società che hanno adottato standard diversi, la loro irregolarità nella distribuzione spaziale è aggregata in una griglia con celle quadrate. Ciò consente confronti tra aree diverse e facilita la gestione geografica dei dati. L'area di Milano è composta da una griglia sovrapposta di 1.000 quadrati con dimensioni di circa 235 × 235 metri. Questa griglia è proiettata con lo standard WGS84 (EPSG: 4326). Per aggregare spazialmente i CDR all'interno della griglia, ogni interazione è associata all'area di copertura v della RBS(Radio Base Station) che la gestiva. Quindi, il numero di record s_i (t) in una griglia quadrata i al tempo t è calcolato come segue:

$$S_i(t) = \sum_{v \in C_{map}} R_v(t) \frac{A_{v \cap i}}{A_v}$$

Dove $R_v(t)$ è il numero di record nell'area di copertura v al tempo t, A_v la superficie dell'area di copertura v e $A_{v \cap i}$ la superficie data dall'intersezione spaziale tra v e il quadrato i. Un esempio di come viene composta questa griglia è rintracciabile nel file python **milanpo_grid_analysis_heatmap** in cui vengono analizzate le aree della città con il maggior numero di chiamate in entrata e in uscita il giorno di Natale 2013.
Utilizzando Pandas per la costruzione del dataframe e seaborn per la costruzione del grafico è stato possibile dimostrare come le aree con maggiore densità di attività sono quelle centrali rispetto a quelle periferiche.
Il risultato prodotto è visibile a seguire e corrisponde all'analisi delle chiamate in entrata.





3.2. I CDR

Un Call Detail Record (CDR) è un record prodotto da una centrale telefonica che documenta in dettaglio una chiamata o lo scambio di SMS attraverso una infrastruttura di telecomunicazioni.
Le reti dei telefoni cellulari sono costruite usando un set di Base Transceiver Stations (BTS) che si occupano della comunicazione con i dispositivi mobili. L'area coperta da una torre BTS è chiamata cella. La dimensione di una cella varia da poche centinaia di metri quadrati in un ambiente urbano fino a tre chilometri quadrati. In qualsiasi momento, uno o più BTS possono fornire copertura a un telefono cellulare. Ogni volta che un individuo fa una telefonata, la chiamata viene instradata attraverso un BTS nell'area di copertura. Il BTS viene assegnato in base al traffico della rete e alla posizione geografica dell'individuo.
Lo schema dei CDR è pertanto il seguente:

1. **Square id**: l'id del quadrato che fa parte della griglia in cui è suddivisa Milano
2. **Time Interval:** l'inizio dell'intervallo di tempo espresso come il numero di millisecondi trascorsi dil 1 ° gennaio 1970 della Unix Epoch. La fine dell'intervallo di tempo può essere ottenuta aggiungendo 600000 millisecondi (10 minuti) a questo valore.
3. **Country code**: il prefisso telefonico associato ad una nazione.
4. **SMS-in:** l'attività in termini di SMS ricevuti all'interno di un determinato square id, durante un certo time interval e inviati dalla nazione identificata dal country code.
5. **SMS-out:** l'attività in termini di SMS inviati all'interno di un determinato square id, durante un certo time interval e ricevuti dalla nazione identificata dal country code.
6. **Call-in:** l'attività in termini di chiamate ricevute all'interno di uno square id, durante il time interval e rilasciato dalla nazione identificata dal country code.
7. **Call-out:** l'attività in termini di chiamate emesse all'interno dello square id, durante l'intervallo di tempo e ricevute dalla nazione identificata dal country code.
8. **Internet: traffic** l'attività in termini di traffico internet eseguito all'interno dello square ID.

I file sono in formato tsv. Se non è stata registrata alcuna attività per un campo specificato nello schema precedente, il valore corrispondente non è presente nel file. Ad esempio, se per una data combinazione di square id, il time interval e il country code non è stato inviato alcun SMS, il record corrispondente appare come segue:

s \ t id \ t \ t \ t SMSout \ t Callin \ t Callout \ t Internettraffic

dove \ t corrisponde al carattere di tabulazione.
Inoltre, se per una data combinazione di square id s, il time interval i e il country code c non viene registrata nessuna attività, si avrà un record del tipo:
s \ t \ t \ t \ t \ t \ t \ t

3.3. Dataset sulle condizioni climatiche

La cartella **weather_phenomena** contiene il dataset che descrive vari tipi di fenomeni meteorologici e la loro intensità nella città di Milano misurati da degli appositi sensori.
Queste informazioni sono fornite direttamente dall'ARPA (Agenzia Regionale per la Protezione dell'Ambiente) sul seguente sito web. Ogni sensore ha un ID univoco, un tipo e una posizione e diversi sensori possono condividere la stessa posizione.
Il dataset è composto da due sotto-set di dati:
Legend dataset(mi_meteo_legend.csv): contiene informazioni sui sensori che raccolgono i dati:
- ID, posizione e tipo del sensore;
- unità di misura.

Weather Phenomen(mi_meteo_#): contiene le misure.
In particolare, i dataset hanno la seguente struttura:
Legend dataset:

- **Sensor ID:** è l'id del sensore.
- **Sensor Street name:** il nome della via in cui si trova il sensore.
- **Sensor lat**: la latitudine geografica che specifica la posizione del sensore.
- **Sensore long**: la longitudine geografica che specifica la posizione del sensore.
- **Sensor type**: il tipo di sensore.
- **UOM:** l'unità di misura del valore registrato dal sensore.

Weather Phenomena dataset:
Questo dataset contiene un file per ciascun sensore. Il nome dei file ha il seguente formato MI_Meteo_ <ID sensore> .csv.

- **Sensor ID:** l'id del sensore. TIPO: alfanumerico
- **Time istant:** l'istante temporale della misurazione espresso come data / ora con i seguenti formato AAAA / MM / GG HH24: MI.
- **Measurement**: il valore dell'intensità dei fenomeni meteorologici misurata nel Time Istant dall'ID sensore.
- **L'unità di misura (UOM)** del valore registrato dal sensor, specificata nel dataset Legend.

La direzione del vento viene misurata in gradi con il nord come piano di riferimento (il nord è specificato come valore 0 o 360 gradi).

3.4 Dataset con i luoghi di interesse

All'interno della cartella milano_pois è presente il file **pois_milano_tripadvisor.csv**.
Questo dataset non è incluso in quelli offerti da Open Big Data (vedere riferimento al punto 2 della bibliografia) ma è stato recuperato da Tripadvisor, per fornire un'analisi più approfondita sulla costituzione del territorio di Milano e sui suoi luoghi di interesse.
Sarà particolarmente utile quando si dovrà ad esempio analizzare la correlazione tra condizioni metereologiche e attività svolte dagli utenti.
Ogni riga del dataset è così formata:

- **name :** indica il nome del punto di interesse (musei, università, parchi,monumenti,chiese)
- **reviews:** il numero di recensioni che il punto di interesse ha ricevuto
- **lat:** la latitudine del punto di interesse
- **lon:** la longitudine del punto di interesse

3.5 Dataset relativo all'utilizzo della corrente elettrica nella regione del Trentino

Tra i dataset offerti dall'API Dandelion di Open Big Data troviamo anche i dati relativi all'uso della corrente elettrica nella regione del Trentino. Questo dataset è stato utilizzato per sperimentare gli algoritmi di machine learning e di MLlib che si trovano nel capitolo finale di questa documentazione.
SET Distribuzione SPA gestisce quasi tutta la rete elettrica sul territorio trentino. Utilizza circa 180 linee di distribuzione primaria (linee di media tensione) per portare energia dalla rete nazionale e distribuirla tra gli utenti. Il dataset fornisce informazioni sulla corrente che scorre attraverso le linee di distribuzione e dettagli su come le linee di distribuzione sono localizzate sul territorio.
Il dataset è composto da due sotto-componenti:
Customer site dataset:

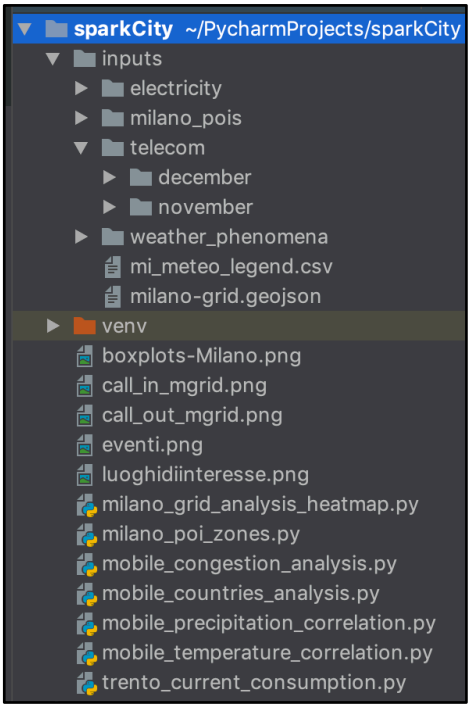
- **Square id**: l'id della griglia che fa parte della GRID del Trentino
- **Line id**: l'identificativo della linea di distribuzione della corrente elettrica associata ad una particolare square_id.
- **Number of costumer sites**: il numero di utenti che si trovano all'interno di una particolare square_id del Trentino

Line measurement dataset:

- **Line ID**: l'id della linea di distribuzione della corrente
- **Timestamp**: la data e l'orario della misura di corrente registrata attraverso una determinate linea di distribuzione. Il formato è il seguente: YYYY-MM-DD HH24: MI'
- **Value**: Il valore in Ampère di ciò che è passato attraverso una particolare linea di distribuzione in una determinata data. Il valore è positivo se il flusso di corrente passa da una linea nazionale a una locale. È negativo quando avviene il caso opposto.

Il primo dataset fornisce una descrizione delle linee di distribuzione primarie che servono il territorio Trentino. Si noti che i siti forniscono spesso energia a più di un cliente. In altre parole, possono fornire elettricità a un cliente (case unifamiliari), a molti clienti (condomini), alle attività commerciali e alle strutture pubbliche.
Nel secondo dataset si evidenzia il fatto che i siti di ciascuna linea sono raggruppati in base ai quadrati della GRID Trentino. Ciò significa che, dato un quadrato della GRID Trentino e una linea di distribuzione specifica, viene registrato il numero di siti che rientrano in tale gruppo.
La corrente che scorre attraverso le linee di distribuzione è stata registrata ogni 10 minuti.

4. Struttura del progetto



Il progetto è costituito da una cartella "inputs" che contiene, divisi per categoria, tutti i dataset descritti al capitolo 4.
A seguire:

- **milano_grid_analysis_heatmap.py**: effettua un'analisi della distribuzione delle attività di telecomunicazioni sul territorio di Milano. I risultati dell'esecuzione di questo file sono i seguenti png: call_in_mgrid, call_out_mgrid, situati nella medesima cartella
- **milano_poi_zones.py**: individua le zone della città di Milano che contengono punti di interesse (chiese, monumenti e musei) ed effettua operazioni di clustering.
- **mobile_congestion_analysis.py**: analizza le attività degli abbonati nei mesi di novembre e dicembre, considerando eventi e luoghi particolari di Milano. I grafici ottenuti sono:
  - boxplots-Milano.png che mostra la variazione delle attività durante la settimana,
  - eventi.png che effettua la medesima analisi su base oraria.
  - luoghidiinteresse.png che prende in considerazione quattro zone diverse di Milano mettendole a confronto per quanto riguarda l'utilizzo della rete internet.
- **mobile_countries_analysis.py** analizza la diversità nelle attività delle minoranze etniche all'interno della città di Milano
- **mobile_precipitation_correlation.py** verifica l'esistenza di una correlazione fra la variazione di intensità nelle precipitazioni e variazioni nella fruizione dei servizi da parte degli abbonati
- **mobile_temperature_correlation.py** trova il legame tra temperatura e attività di telecomunicazioni a seconda o meno della presenza di punti di interesse
- **trento_current_consumption.py** prende in esame la distribuzione della rete elettrica di Trento per utilizzare librerie di Machine Learning come MLlib e raggruppare utenti con finalità di marketing.

5. Istruzioni per l'installazione

Per eseguire il progetto è necessario avere installato Python, Hive e Spark.
Utilizzando un'IDE di sviluppo come lo stesso Pycharm è possibile, modificando lo script path, eseguire i file Python descritti all'interno di ogni paragrafo del progetto.
Una panoramica sui file e le loro funzionalità puo' essere trovata nel paragrafo precedente.

6. Mappatura della popolazione e delle nazionalità presenti grazie ai CDR

La disponibilità dell'enorme quantità di CDR permette di effettuare una vasta sperimentazione sulla città di Milano, indagando su come le persone utilizzano e vivono in una delle più grandi città italiane. Un efficiente strumento di analisi come Pandas è stato indispensabile per l'identificazione di comportamenti interessanti e nascosti che altrimenti non emergerebbero.

6.1 Data processing con Hive

I dati di censimento ci aiutano a capire i modelli di sviluppo e il movimento umano. Nonostante l'ampia rilevanza e l'importanza di tali dati l'acquisizione di stime del censimento locale è difficile in modo preciso e accurato perché i conteggi delle popolazioni possono cambiare rapidamente e soffrire di problemi logistici e amministrativi oltre ad essere molto costosi. Queste limitazioni richiedono lo sviluppo di approcci alternativi e complementari alla mappatura della popolazione. I dati offerti dal campo delle telecomunicazioni come telefonate, messaggi di testo e utilizzo di internet sono una promettente nuova fonte di misurazione della popolazione in tempo reale. Secondo l'Istituto Nazionale di Statistica la popolazione straniera è stata di circa il 15.5% della popolazione di Milano nel 2013. Molti immigrati formano delle comunità mentre alti sono distribuiti in modo più uniforme in tutta la città. Inoltre, alcuni sono irregolari e questo complica gli sforzi nella formazione dei dati sulla popolazione. Il numero crescente di immigrati in Italia, sia regolari che irregolari ha portato l'Italia a stringere le sue politiche di migrazione e integrazione nonché a istituire una serie di programmi.
Di conseguenza è importante sviluppare strumenti per capire meglio dove vivono i migranti regolari e irregolari e la concentrazione o dispersione delle loro comunità.
Con Hive è possibile analizzare e processare una grande quantità di dati come i CDR e formulare delle queries.
La composizione di queste deve tenere a mente che con il country-code 0 l'operatore di telecomunicazione non conosce il paese di origine/ destinazione oppure che l'utente ha chiesto di nascondere queste informazioni per motivi di privacy.

Assumeremo che 0 sia solo un altro countrycode.
I dataset hanno inoltre molte celle vuote. Bisogna quindi importare i file in modo tale che le celle vuote vengano trattate come NULL in Hive. Ai fini progettuali, a causa dell'elevata complessità computazionale dovuta alla grandezza dei file si è deciso di utilizzare un campione di dati che coprono cinque giorni che vanno dall' 1-12-2013 al 5-12-2013 (la scelta delle date è stata casuale).

```
drop table cdr;

create external table IF NOT EXISTS cdr (
square_id int,
time_interval bigint,
country_code int,
sms_in float,
sms_out float,
call_in float,
call_out float,
internet_traffic float)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
TBLPROPERTIES('serialization.null.format'='');
LOAD DATA LOCAL INPATH 'sparkCity/hive/inputs' OVERWRITE INTO TABLE cdr;
```

Analizziamo quindi le minoranze etniche presenti all'interno della città di Milano per descriverne il comportamento.
Da Wikipedia sappiamo che la percentuale più alta di stranieri è costituita da Filippini (40 474), Egiziani (35 884) e Cinesi (27 679). Il prefisso utilizzato nelle Filippine è il 63, quello utilizzato in Egitto 20, quello cinese 86.
Con Hive individuiamo le griglie (square_id) che sono popolate dalle diverse etnie per analizzare il fenomeno dei quartieri a lato indice straniero. Come prima cosa creiamo una nuova tabella in cui sono presenti solo i tre countrycode relativi ai popoli maggiormente presenti.

```
CREATE EXTERNAL TABLE IF NOT EXISTS cdr_threecountries(
square_id int,
time_interval bigint,
country_code int,
sms_in float,
sms_out float,
call_in float,
call_out float,
internet_traffic float);

SELECT square_id,time_interval,country_code,sms_in,sms_out,call_in,call_out,internet_traffic
FROM cdr
WHERE country_code IN (63,20,86);
```

Parte del risultato ottenuto è visibile nell'immagine a seguire:

```
7780    1386087600000   20      0.006464182399213314    NULL    NULL    NULL    NULL
7780    1386088200000   20      NULL    NULL    NULL    0.012928364798426628    NULL
7780    1386089400000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386091800000   20      NULL    NULL    NULL    0.012928364798426628    NULL
7780    1386092400000   20      NULL    NULL    NULL    0.019392548128962517    NULL
7780    1386093000000   20      NULL    NULL    NULL    0.012928364798426628    NULL
7780    1386093600000   20      NULL    NULL    NULL    0.16003909707069397     NULL
7780    1386094200000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386094800000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386095400000   20      0.006464182399213314    NULL    NULL    0.17296746373176575     NULL
7780    1386096000000   20      0.006464182399213314    NULL    NULL    0.006464182399213314    NULL
7780    1386096600000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386097200000   20      NULL    NULL    NULL    0.025856729596853256    NULL
7780    1386097800000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386099600000   20      NULL    NULL    NULL    0.012928364798426628    NULL
7780    1386100200000   20      NULL    NULL    NULL    0.012928364798426628    NULL
7780    1386100800000   20      NULL    NULL    NULL    0.025856729596853256    NULL
7780    1386102000000   20      0.006464182399213314    NULL    NULL    NULL    NULL
7780    1386103200000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386104400000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386105000000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386105600000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386108000000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7780    1386110400000   20      NULL    NULL    NULL    0.006464182399213314    NULL
7781    1386043200000   20      0.01147052925080061     NULL    NULL    NULL    NULL
7781    1386046200000   20      NULL    NULL    NULL    0.01147052925080061     NULL
7781    1386046200000   86      0.01147052925080061     NULL    NULL    NULL    NULL
7781    1386048000000   86      NULL    NULL    NULL    0.01147052925080061     NULL
7781    1386048600000   20      NULL    NULL    NULL    0.01147052925080061     NULL
7781    1386054000000   20      0.22386381030082703     NULL    NULL    NULL    NULL
```

Quante griglie (ossia square_id) racchiudono le seguenti minoranze?

```
SELECT count(DISTINCT(square_id)) as square_id from cdr_ threecountries;
```

Il risultato è 9998.
Ci chiediamo adesso quale siano le attività svolte: Quali tra queste etnie fa un utilizzo minore di internet?

```
SELECT sum(internet_traffic) as internet_activity ,country_code
FROM cdr_threecountries
GROUP BY country_code
ORDER BY internet_activity ASC
LIMIT 1;
```

Prendendo come campione I cinque giorni precedentemente indicati, la nazione che fa meno utilizzo della connessione di rete è quella delle Filippine. Quale nazione invece ha il maggior numero di attività? (Chiamate, invio e ricezione di sms, traffico di rete).

```
SELECT
SUM(SMSin_activity)+SUM(SMSout_activity)+SUM(Callin_activity)+SUM(Callout_activity)+SUM(Internet
traffic_activity) as total_activity, country_code
FROM cdr_threecountries
GROUP BY country_code
ORDER BY total_activity DESC
LIMIT 1,1;
```

La query porta alla luce il fatto che la nazione a compiere il maggior numero di attività è l'Egitto per un totale di 2000102.04191124724
Se vogliamo analizzare il solo invio di messaggi, la zona destinata a farne maggior utilizzo secondo la query sottostante è 5061 per un totale di 988.8287563584745

```
SELECT SUM(SMSin_activity)+SUM(SMSout_activity) as total_sms_activity ,Square_id
FROM cdr_threecountries
GROUP BY square_id
ORDER BY total_sms_activity desc
limit 1;
```

6.2 Data Analysis con Spark

Vista la complessità dei dati che fanno uso delle coordinate geografiche si è deciso di utilizzare PySpark per proseguire con la nostra analisi. In particolare, verrà adoperato Spark SQL. Spark SQL si integra perfettamente con il resto del sistema Spark, infatti i risultati delle query possono essere trasformati e analizzati usando l'API di Spark che funziona sugli RDD, così come è possibile analizzare i dati con HiveQL dopo avergli associato uno schema tabellare. Le operazioni anche in questo caso verranno effettuate su un campione di dati relativo a cinque giorni: dal 1° dicembre al 5 dicembre del 2013.
Si è deciso di individuare la griglia che ha il maggior numero di attività, vale a dire SMS, chiamate e internet.
Facendo riferimento al file **mobile_countries_analysis.py** , come prima operazione si procede con il caricamento del dataset e la creazione del dataframe, si veda punto [1] del file..
Si filtra poi il dataset eliminando tutte le righe in cui il countrycode è pari a 0 e quindi inutile ai fini della nostra analisi

```
cleanedData = df.filter(df['country_code'] > 0)
cleanedData.count()
```

Il dataframe che si ottiene (utilizzando la funzione printSchema()) avrà questi campi:

```
root
 |-- square_id: string (nullable = true)
 |-- time_interval: string (nullable = true)
 |-- country_code: string (nullable = true)
 |-- SMS_in: string (nullable = true)
 |-- SMS_out: string (nullable = true)
 |-- Call_in: string (nullable = true)
 |-- Call_out: string (nullable = true)
 |-- Internet_traffic: string (nullable = true)
```

Analizziamo la griglia che ha il maggior numero di attività ossia: chiamate, sms e utilizzo di internet. Per questo scopo si crea con Spark SQL una view che puo' essere usata come una tabella Hive. La view generata non è persistente in memoria e ed è rintracciabile al punto [2] nei commenti del file Python.
Ciò che la console restituirà saranno i seguenti risultati:

```
+---------+------------------+
|square_id|           SMS_out|
+---------+------------------+
|     5772|99.99004543563095|
+---------+------------------+

|square_id|          Call_out|
+---------+------------------+
|     4862|99.99975980836263|
+---------+------------------+

+---------+------------------+
|square_id| Internet_traffic|
+---------+------------------+
|     5956|999.9872433188533|
+---------+------------------+
```

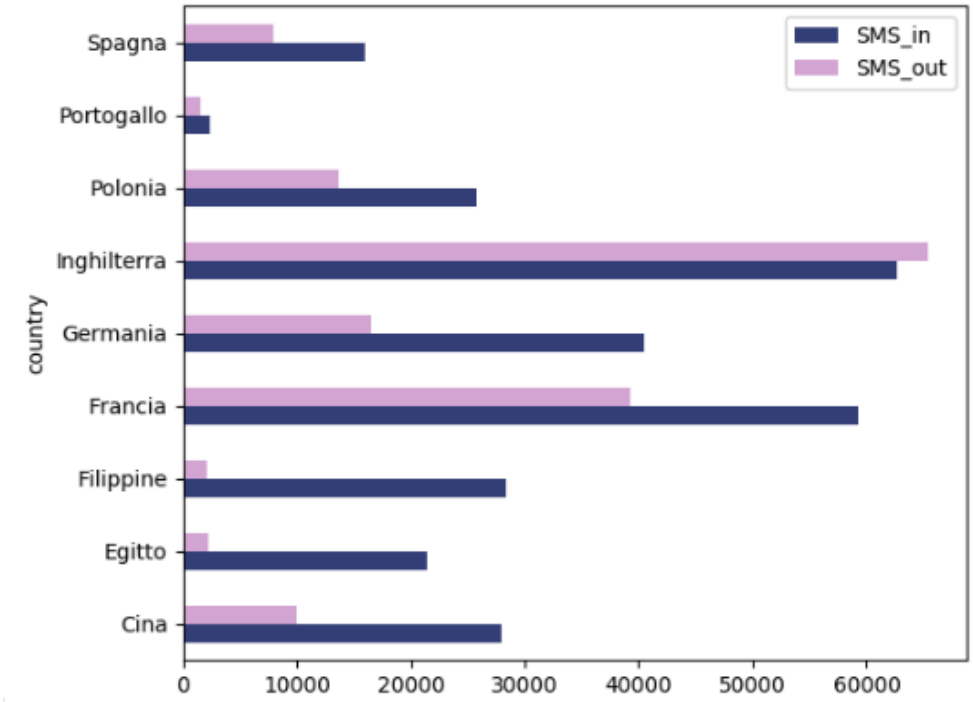
Desideriamo quindi ricercare i differenti utilizzi di GSM, SMS, Internet per nazionalità.
Si è scelto per questa query di adoperare oltre a Filippine, Egitto e Cina, già utilizzate per l'analisi fatta con Hive, anche altre nazioni come la stessa Italia, Francia, Spagna, Germania, Polonia, Portogallo, Inghilterra. Questo perché adesso abbiamo degli strumenti più complessi che ci permettono di restituire un'analisi visiva più dettagliata attraverso i plot.
La query attraverso cui distinguiamo le diverse nazionalità degli abbonati è la seguente:

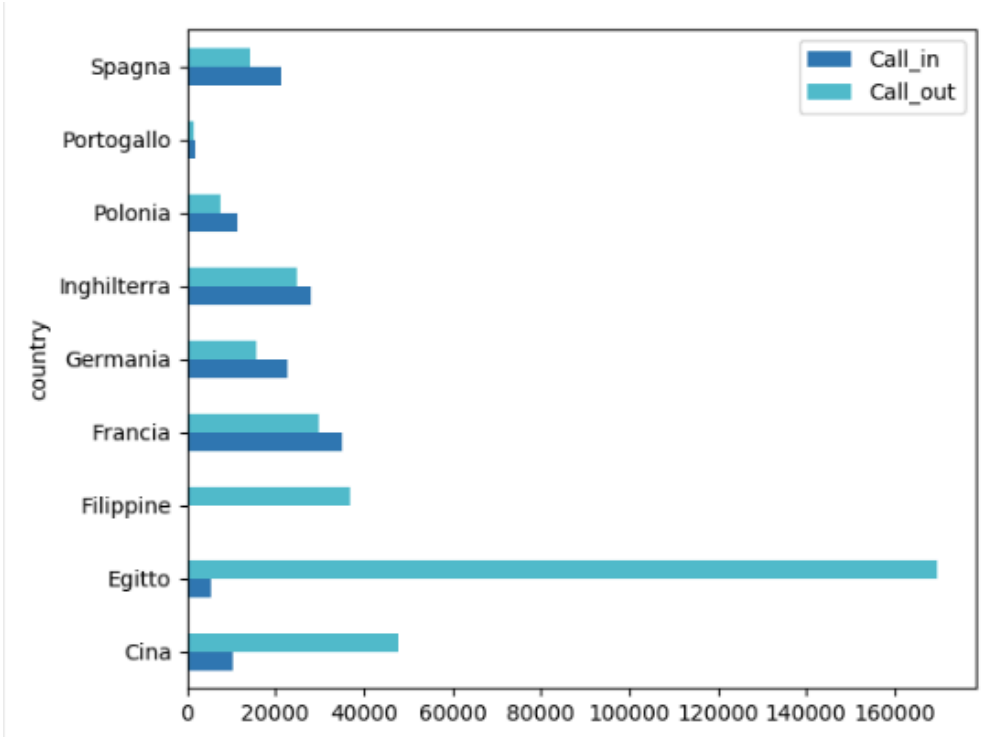
```
aggcountryDF = spark.sql("""select
                                CASE country_code
                                    WHEN 63 THEN "Filippine"
                                    WHEN 33 THEN "Francia"
                                    WHEN 34 THEN "Spagna"
                                    WHEN 39 THEN "Italia"
                                    WHEN 44 THEN "Inghilterra"
                                    WHEN 20 THEN "Egitto"
                                    WHEN 48 THEN "Polonia"
                                    WHEN 49 THEN "Germania"
                                    WHEN 351 THEN "Portogallo"
                                    WHEN 86 THEN "Cina"
                                    ELSE "Altre"
                                END as country,
                                round(sum(SMS_in),6) SMS_in,
                                round(sum(SMS_out),6) SMS_out,
                                round(sum(Call_in),6) Call_in,
                                round(sum(Call_out),6) Call_out,
                                round(sum(Internet_traffic),6) Internet_traffic
                            from telecommunicationData
                            where country_code != 0
                            group by country
                            order by 1""")
```

Che è visibile sotto forma tabellare in questo modo:

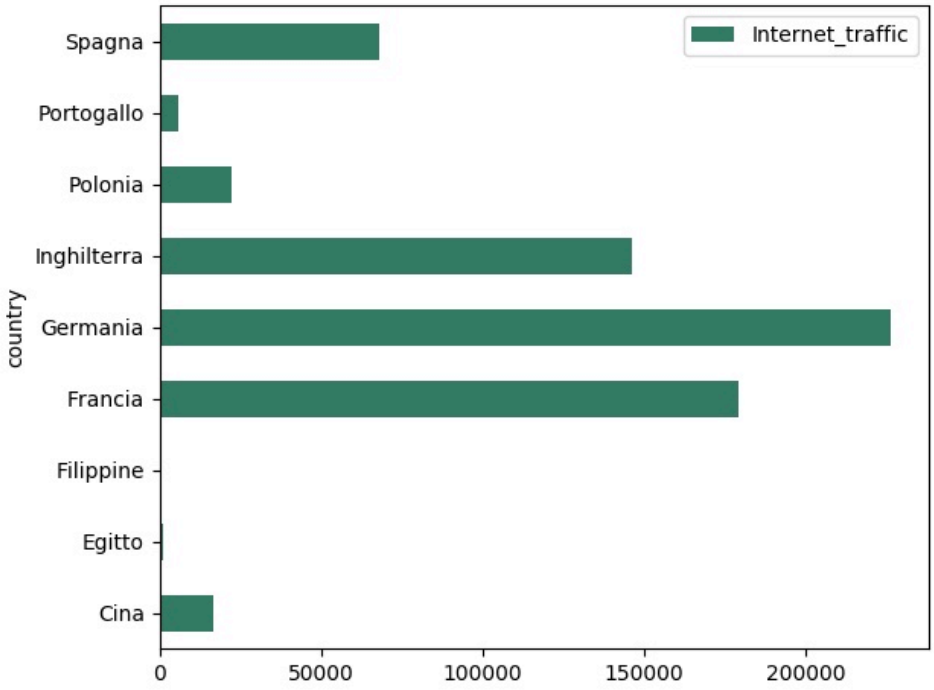
```
+----------+----------------+----------------+----------------+----------------+-----------------+
|   country|          SMS_in|         SMS_out|         Call_in|        Call_out| Internet_traffic|
+----------+----------------+----------------+----------------+----------------+-----------------+
|    Altre|  381368.272657|  186155.149414|  239657.529463|  498091.577531|   676191.023837|
|     Cina|   27909.714098|   10022.440945|    10331.24619|   47650.287358|    16875.560404|
|   Egitto|   21379.321702|    2294.659039|    5520.78595|  169741.796501|     1165.478634|
| Filippine|   28346.974974|    2126.521784|    598.531432|   36942.268813|            null|
|   Francia|   59208.872552|   39284.937754|  35205.296096|   29655.335169|   179177.143056|
|  Germania|   40498.230324|   16530.045565|  22730.967892|   15582.870831|   226534.361368|
|Inghilterra|   62683.440039|   65441.288504|   28008.208634|   24722.740587|    146372.48245|
|    Italia|2.2810714258679E7|1.6726538319501E7|2.2961608775101E7|2.5656826633084E7|4.59787080743861E8|
|   Polonia|   25719.256536|   13730.954473|   11542.897828|    7610.048337|    22401.70777|
|Portogallo|    2418.558668|    1522.691022|    1806.092069|     1555.01297|    5828.966016|
|    Spagna|   15949.253687|    7995.851581|   21388.340187|   14163.609537|    67992.440729|
+----------+----------------+----------------+----------------+----------------+-----------------+
```

Grazie a matplotlib e a pandas è stato possibile elaborare visivamente i seguenti dati utilizzando le istruzioni del punto [3] e ottenendo i seguenti risultati :





Nel primo grafico emerge come gli inglesi facciano grandissimo uso della messaggistica al contrario dei portoghesi.
Il secondo invece dimostra che la minoranza egiziana è quella che effettua il maggior numero di chiamate mentre i francesi sono coloro che ne ricevono di più. Emerge inoltre che in questo intervallo di cinque giorni nessuno della comunità filippina ha effettuato chiamate in uscita, risultato insolito, che trova tuttavia conferme nelle query effettuate nel capitolo precedente con Hive. Nel caso invece del traffico di rete, chi ne genera maggiormente è la Germania, seguita da Francia e Inghilterra.
Le Filippine e l'Egitto invece hanno valori prossimi allo zero come evidenziato dall'immagine nella pagina successiva.



7.Osservazione dei periodi di congestione delle reti ed eventi

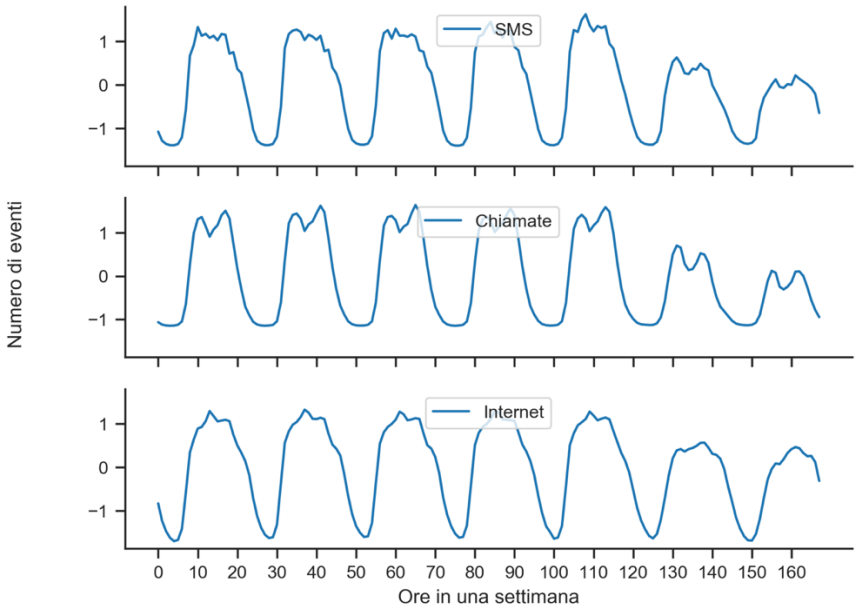
Le informazioni contenute nei Call Details Records (CDRs) delle reti mobili possono essere utilizzate per studiare l'efficacia operativa delle reti cellulari e il modello comportamentale degli abbonati. Nel file python **milano_poi_zones** è possibile recuperare attraverso il dataframe cell_position, servendosi di latitudine e longitudine, le coordinate di tutti i punti di interesse frequentati dagli abbonati:

	lat	lon
2031	9.10158	45.40106
2032	9.10458	45.40106
2033	9.10759	45.40106
2034	9.11059	45.40105
2035	9.11359	45.40105
2036	9.11659	45.40105
2037	9.11960	45.40105
2038	9.12260	45.40104
2039	9.12560	45.40104
2040	9.12860	45.40104
2041	9.13161	45.40103
2042	9.13461	45.40103
2043	9.13761	45.40103
2044	9.14061	45.40102
2045	9.14362	45.40102
2046	9.14662	45.40101
2047	9.14962	45.40101
2048	9.15263	45.40101
2049	9.15563	45.40100

7.1 Data Ingestion con Spark e analisi con Pandas

L'ingestion dei dati avviene in **mobile_congestion_analysis.py** grazie a Spark e Pandas.
Si osservi il punto [1] del file nella sezione "Graphic Analysis with Pandas" per quanto riguarda la realizzazione dei grafici.
Data l'elevata latenza nei calcoli, è stato scelto di focalizzarsi su un intervallo di giorni che vanno dal 4 novembre (lunedì) al 10 novembre (domenica). L'obiettivo è capire come i dati vengono distribuiti nelle ore del giorno e nei giorni della settimana. Si realizza quindi a tale scopo un nuovo dataframe in cui record sono raccolti tramite le colonne "weekday" e "hour" dove la variabile weekday corrisponde al giorno della settimana (è un valore che varia da 0 a 6 dove 0 è lunedì e 6 domenica).
Si sommano a questo punto i valori presenti nelle colonne SMS_in e SMS_out e Call_in e Call_out per ottenere il nostro grafico (punto [2]). Come analizzare i dati ottenuti dal punto di vista statistico? Lo Z score permette di prendere un campione di dati all'interno di un insieme più ampio e di determinare di quante deviazioni standard si trova sopra o sotto la media. Per trovare lo Z score, occorre quindi prima calcolare la media e la deviazione standard (funzione std() di Python). Ciò viene fatto nel punto [3].

Si procede quindi alla realizzazione del grafico con matplotlib (punto [4]). Come asse delle ascisse sono state inserite le ore in una settimana e come asse delle ordinate il numero di eventi (SMS, chiamate, internet).
Il risultato ottenuto è il seguente:



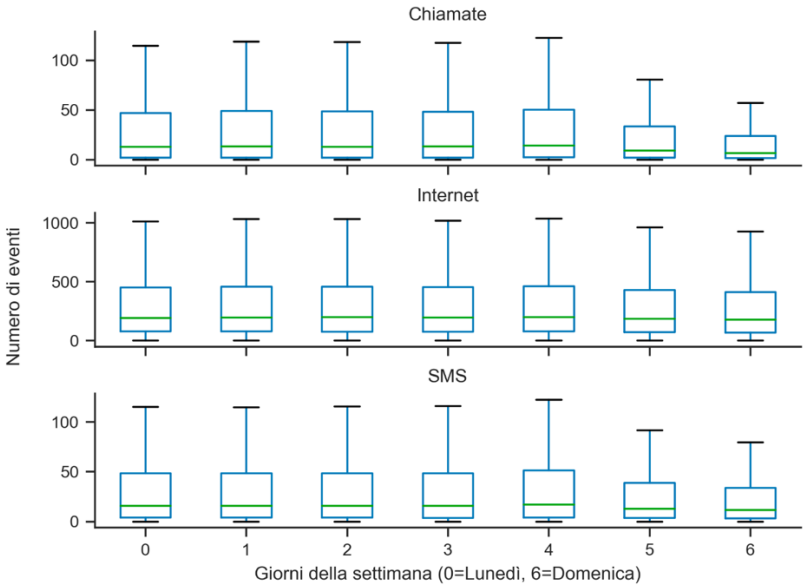
È possibile osservare una forte frequenza giornaliera che di solito inizia alle 7:00, quando le persone accendono i loro telefoni e probabilmente vanno al lavoro. Le attività diminuiscono lentamente di sera quando le persone tornano a casa e dormono. Inoltre, vi è anche una frequenza settimanale dovuta al comportamento dei cicli di lavoro delle persone (ad esempio, giorni lavorativi rispetto ai fine settimana). Possiamo trovare conferme a questi grafici effettuando delle query con Spark SQL, che si trovano all'inizio del file e sono contrassegnate dal commento #Analysis with Spark.
Grazie alla funzione weekday anche in questo caso è possibile recuperare il giorno della settimana con il picco di SMS, Chiamate e internet. Si ottengono i seguenti risultati che confermano quanto detto in precedenza

weekday	smsInUscita	smsInEntrata	total
1	3060250.1236862116	4736588.866935715	7796838.990621926
6	4340860.837271979	8444244.795913843	1.2785105633185823E7
3	4368527.1849885015	7931790.919737555	1.2300318104726057E7
5	4286852.189652543	8144553.503937074	1.2431405693589617E7
4	4351455.45901171	7947297.951739079	1.229875341075079E7
7	3211772.895494709	5878167.618909889	9089940.514404599
2	4288716.307735313	7742700.151768441	1.2031416459503755E7

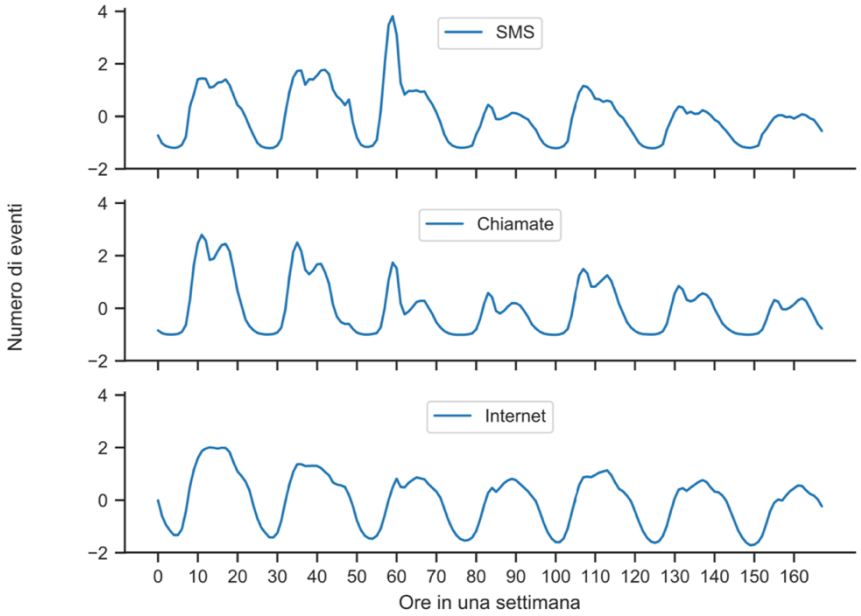
weekday	chiamateInUscita	chiamateInEntrata	total
1	3023323.8005409585	2519114.0247366857	5542437.825277644
6	6375751.505718112	5764061.544184897	1.213981304990301E7
3	6476254.529037363	5692466.008660497	1.216872053769786E7
5	6231773.568982978	5648969.902365169	1.1880743471348148E7
4	6340527.040137751	5702408.535075061	1.2042935575212812E7
7	4152390.4474694757	3616129.360533009	7768519.808002485
2	6166560.319346815	5409633.875191655	1.157619419453847E7

weekday	max(internet_traffic)
1	999.917814771181
6	999.8713934941288
3	999.9598361169271
5	999.9015505109251
4	999.9946722742834
7	999.7782154703342
2	999.8256473925418

Per rappresentare al meglio la distribuzione relativa e assoluta a livello temporale dei servizi di cui gli abbonati usufruiscono, sono stati utilizzati i boxplot (punto [6])
I boxplot ci permettono di visualizzare la mediana ed eventuali outliers.



Ciò che si evince è che durante il periodo preso in esame, il numero di attività svolte è maggiore a inizio settimana piuttosto che durante il weekend. Lo stesso studio può essere fatto su un campione diverso di dati. Ad esempio, durante la settimana del Natale, così da analizzare come variano le abitudini degli abbonati in prossimità di eventi. Per fare ciò basta sostituire NOVEMBER_PATH con DECEMBER_PATH alla riga 66, nella fase di ingestion di Pandas. L'attività relativa alle chiamate ha il suo picco Lunedì 23 dicembre e la Vigilia di Natale. Diverso è il comportamento degli SMS che hanno il loro picco tra la vigilia e la giornata di Natale. Da ciò possiamo evincere come gli abbonati Telecom diano maggiore risalto alla messaggistica durante alle festività. Questo comportamento è evidente anche nel seguente grafico:

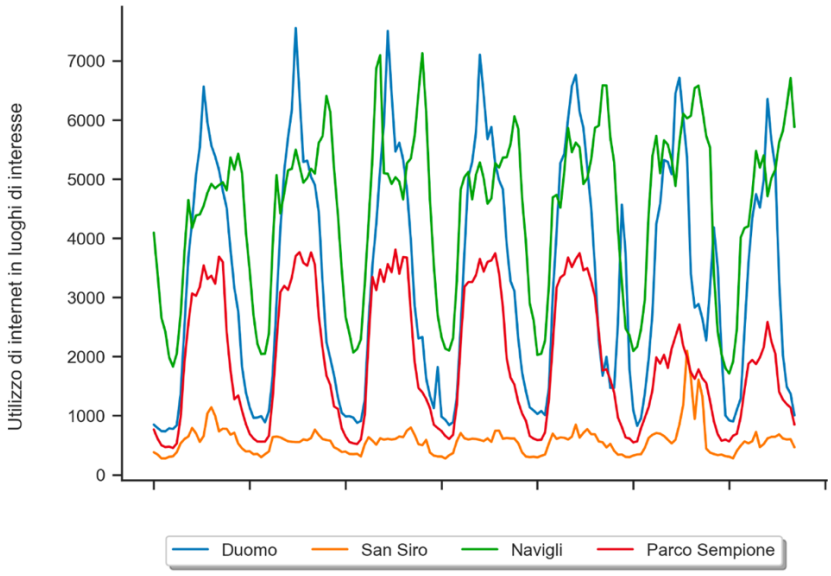


Un'ulteriore analisi è stata fatta per i luoghi di interesse di Milano (punto [5]). Sono stati presi in esame:

- Il Duomo (id **5060**)
- San Siro (id **5737)**
- Navigli (id **4456**)
- Parco Sempione (id **5356**)

Ognuno di questi luoghi ha una fetta di mercato completamente diversa, con abbonati che avranno quindi comportamenti diversi (turisti, tifosi, studenti) nell'utilizzo di sms, chiamate e internet.
Per effettuare questa analisi sul dataframe Pandas la colonna 'idx' deve essere filtrata in base ai diversi valori dello square_id dato in input.

Ricordando che ‘idx’ raccoglie il numero totale di ore e minuti di attività per i diversi giorni della settimana, avremo per la settimana di novembre:



Gli abbonati che fanno il maggior utilizzo di Internet sono quelli situati al Duomo.
Il motivo potrebbe essere legato al luogo di interesse, che si presta a ricerche ed approfondimenti da parte dei turisti ma anche il fatto che questo è uno snodo fondamentale della rete metropolitana di Milano.
Anche in prossimità dei Navigli si ha un utilizzo molto consistente della connessione. Questo è dovuto al fatto che chi frequenta tipicamente la zona sono studenti e giovani che fanno un maggiore utilizzo di questa tecnologia. Da segnalare il picco su San Siro causato dalla partita Inter – Livorno (2 -0) che vede il rientro in campo, dopo 200 giorni dall’infortunio, dell’ex capitano dell’Inter Zanetti.

8.La temperatura come strumento per prevedere il traffico telefonico

Con questo capitolo ci poniamo l’obiettivo di osservare la correlazione che c’è fra l’utilizzo dei mezzi di comunicazione e le differenti condizioni meteo. Il legame tra queste diverse grandezze è da ricercare nel file python **mobile_temperature_correlation.py**.
L’operazione di join di questi dati con i cdr precedentemente analizzati avviene sulla base della colonna time_interval (punto 5). Prima di analizzare i dati è necessario effettuare delle operazioni per aumentare l’efficienza e la potenza di calcolo: si è deciso di sommare il contenuto della colonna “SMS_in” con quello della colonna “SMS_out” e lo stesso vale per le colonne “Call_in”, “Call_out”(punto 3). Si è scelto di eliminare il country_code in quanto non è di interesse ai fini di questa analisi.
Si opera poi sulla colonna “time_interval” per rendere il suo formato omogeneo a quello della tabella delle condizioni meteo in modo da effettuare l’operazione di join.

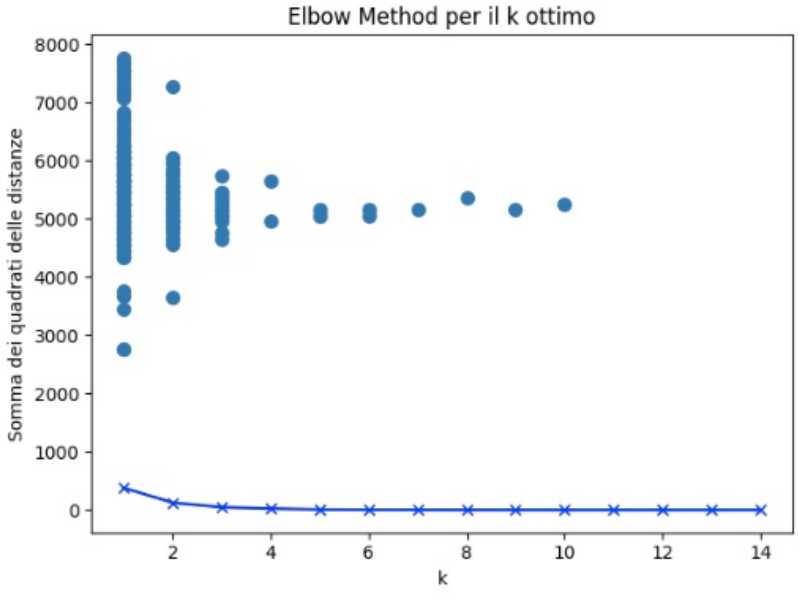
time_istant	SMS_total	Call_total	internet	sensor_id	measurement	street_name	lat	lon	sensor_type	unity_of_measure
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6502	999.9	Milano – via Fili...	45.473622	9.220392	Atmospheric Pressure	hPa
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	19021	38	Milano – viale Ma...	45.496067	9.193023	Wind Direction	degree
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	2001	12.9	Milano – via Lamb...	45.490051	9.225596	Temperature	Celsius degree
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	8162	12.8	Milano – via Feltre	45.49145	9.242386	Temperature	Celsius degree
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	5911	13	Milano – viale Ma...	45.496067	9.193023	Temperature	Celsius degree
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	5920	12.8	Milano – P.zza Z...	45.476089	9.143509	Temperature	Celsius degree
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	5909	13.6	Milano – via Fili...	45.473622	9.220392	Temperature	Celsius degree
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6129	1.7	Milano – via Fili...	45.473622	9.220392	Wind Speed	m/s
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6457	-7	Milano – via Fili...	45.473622	9.220392	Net Radiation	W/m^2
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	14121	0.6	Milano – via Ippo...	45.490043	9.194632	Precipitation	mm
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6045	41	Milano – via Fili...	45.473622	9.220392	Wind Direction	degree
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6597	99	Milano – viale Ma...	45.496067	9.193023	Relative Humidity	%
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	2002	93	Milano – via Lamb...	45.490051	9.225596	Relative Humidity	%
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6179	100	Milano – via Fili...	45.473622	9.220392	Relative Humidity	%
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6185	98	Milano – P.zza Z...	45.476089	9.143509	Relative Humidity	%
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	2008	0	Milano – via Lamb...	45.490051	9.225596	Global Radiation	W/m^2
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	2006	0.6	Milano – via Lamb...	45.490051	9.225596	Precipitation	mm
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	5908	0.2	Milano – via Fili...	45.473622	9.220392	Precipitation	mm
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6120	1	Milano – via Brera	45.471192	9.187616	Wind Speed	m/s
2013/11/08 19:00	728252.4195843171	803058.0606093591	5597472.112938035	6030	44	Milano – via Brera	45.471192	9.187616	Wind Direction	degree

only showing top 20 rows

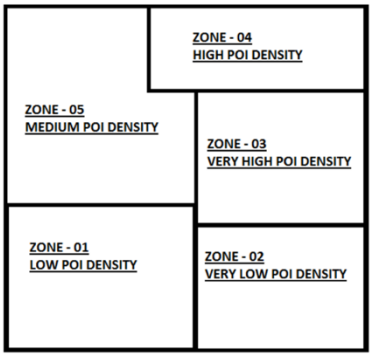
Continuiamo a prendere in esame la prima settimana di novembre in quanto sono presenti precipitazioni ed eventi metereologici rilevanti. L'analisi di una settimana non è tuttavia sufficiente per poter stabilire a priori se c'è o meno un legame tra queste variabili. Occorre avere un intervallo più ampio per avere dati concreti su cui trarre delle conclusioni.
Si è deciso di procedere con l'analisi di questi dati, prendendo innanzitutto gli estremi ovvero vedendo la correlazione che c'è fra la condizione meteo e il giorno in cui è stato mandato il maggiore e il minor numero di sms (nel file Python questa parte è commentata ed è rintracciabile al commento FIRST ACTIVITIES TO INVESTIGATE THE DATASET).
In questo caso, come si può vedere dalla stampa del dataframe, non sono stati riscontrati dei legami in quanto le precipitazioni sono nulle anche se l'attività di messaggistica è superiore alla media. Lo stesso è stata fatto per chiamate e collegamento ad internet. Si deduce quindi che l'impennata di questi eventi sia dovuta a fattori esterni che nulla centrano con le condizioni metereologiche.

time_istant	measurement	street_name	sensor_type	unity_of_measure	SMS_total
2013/11/08 12:00	1001.2	Milano - via Fili...	Atmospheric Pressure	hPa	972512.2887596646
2013/11/08 12:00	111	Milano - viale Ma...	Wind Direction	degree	972512.2887596646
2013/11/08 12:00	14.3	Milano - via Lamb...	Temperature	Celsius degree	972512.2887596646
2013/11/08 12:00	14	Milano - via Feltre	Temperature	Celsius degree	972512.2887596646
2013/11/08 12:00	14.4	Milano - viale Ma...	Temperature	Celsius degree	972512.2887596646
2013/11/08 12:00	13.6	Milano - P.zza Z...	Temperature	Celsius degree	972512.2887596646
2013/11/08 12:00	14.1	Milano - via Fili...	Temperature	Celsius degree	972512.2887596646
2013/11/08 12:00	1.1	Milano - via Fili...	Wind Speed	m/s	972512.2887596646
2013/11/08 12:00	40	Milano - via Fili...	Net Radiation	W/m^2	972512.2887596646
2013/11/08 12:00	0	Milano - via Ippo...	Precipitation	mm	972512.2887596646
2013/11/08 12:00	134	Milano - via Fili...	Wind Direction	degree	972512.2887596646
2013/11/08 12:00	91	Milano - viale Ma...	Relative Humidity	%	972512.2887596646
2013/11/08 12:00	88	Milano - via Lamb...	Relative Humidity	%	972512.2887596646
2013/11/08 12:00	98	Milano - via Fili...	Relative Humidity	%	972512.2887596646
2013/11/08 12:00	91	Milano - P.zza Z...	Relative Humidity	%	972512.2887596646
2013/11/08 12:00	54	Milano - via Lamb...	Global Radiation	W/m^2	972512.2887596646
2013/11/08 12:00	0	Milano - via Lamb...	Precipitation	mm	972512.2887596646
2013/11/08 12:00	0	Milano - via Fili...	Precipitation	mm	972512.2887596646
2013/11/08 12:00	1.3	Milano - via Brera	Wind Speed	m/s	972512.2887596646
2013/11/08 12:00	85	Milano - via Brera	Wind Direction	degree	972512.2887596646

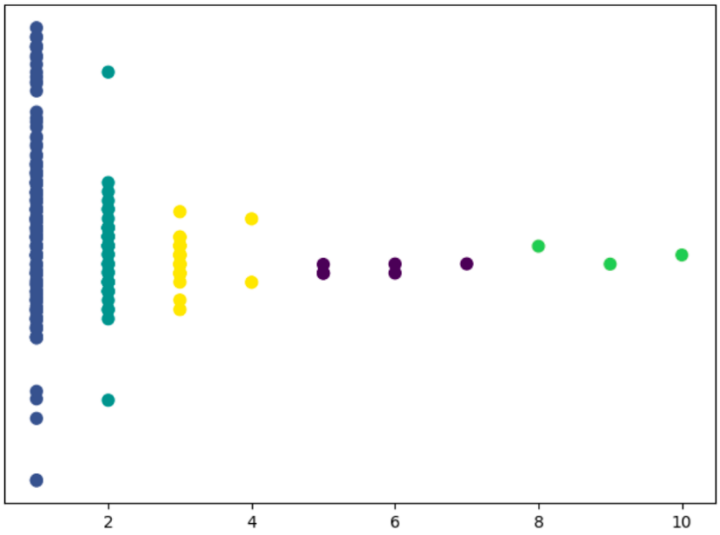
La funzione di correlazione restituisce il coefficiente di correlazione di due colonne, se vicino alla 0 significa che non vi è alcun legame fra questi valori. Tuttavia, è chiaro che la maggior parte della variazione del traffico generato dai dispositivi mobili è dovuto anche ad eventi ma anche soprattutto ai punti di interesse, come è stato reso evidente dalle heatmap a capitolo 3 pagina 4 e dal grafico di pagina 14. Proviamo adesso a considerare la presenza di questi punti di interesse, vale a dire quindi: musei, monumenti, chiese...
Per analizzare la correlazione che c'è fra i dati sul clima e le attività degli abbonati è importante definire un modo intuitivo per raggruppare i diecimila quadrati che costituiscono Milano. Si pensi quindi a un raggruppamento in zone in base alle principali attrazioni. Come cambia quindi l'analisi precedente se ci basiamo quindi sui POIs (Point of Interest) anziché sull'intera grid?
Come prima operazione occorre recuperare la lista dei punti di interesse, e questa può essere trovata facilmente su Tripadvisor. Il file di progetto che li raccoglie è **pois_milano_tripadvisor.csv**
La griglia di Milano viene divisa intuitivamente in cinque zone a seconda della densità del punto di interesse (POI) nella zona. La scelta del numero cinque viene effettuata dopo aver utilizzato l'Elbow Method, in un file apposito **milano_poi_zones.py** (punto [3]). Il grafico ottenuto è il seguente:



Il processo utilizzato è stato il seguente: iterativamente si è ipotizzato un numero di cluster (nel nostro caso è stato preso in esame un range tra 1 e 15 cluster) e per ogni simulazione è stata individuata la somma degli scarti dei centroidi che sono stati mappati sul grafico. Notare come la curva in basso, che ha un angolo a gomito scende progressivamente fino al livello 5. I miglioramenti successivi non sono consistenti e quindi questo è proprio il numero da prendere in esame. A seguito dell'individuazione di questi cluster la mappatura delle zone di Milano sarà la seguente:



A causa della natura di queste zone che possono essere urbane, suburbane e rurali, la correlazione tra la temperatura e i dati sarà diversa.
L'intuizione è che se la griglia è pesantemente popolata da edifici residenziali, il valore di correlazione tra temperatura e dati sarà probabilmente poco correlato in alcune fasce orarie. È immediato notare, tuttavia, che le celle con almeno un punto di interesse sono solo 219 su 10.000 e solo 6 celle hanno 6 o più punti di interesse nella loro area.
La temperatura ha una correlazione con le attività degli abbonati (sms, chiamate, internet), ma potrebbe essere significativa per alcuni o non esserlo per gli altri a seconda della natura del POI.
Anche queste categorie di correlazione cambiano e dipendono dall'ora del giorno e dal giorno stesso per qualsiasi punto di interesse. Una maggiore precisione si può ottenere utilizzando il Machine Learning e in particolare le tecniche di clustering (punto [2]). Si è scelto di impiegare a tale scopo, sci kit learn: la soluzione verrà poi confrontata con MLlib nei capitoli a seguire.
Il raggruppamento dei quadrati che compongono la griglia di Milano viene fatto in base al numero di POI nel metodo kMeans() del file **milano_poi_zone.py** che prende in input un oggetto di tipo Pandas Series costituito da quadrati e punti di interesse.
Il risultato ottenuto sarà il seguente grafico:



Ciò che a noi interessa in questo caso non è tanto il risultato visivo, quanto, piuttosto gli elementi che compongono i cluster.
L'obiettivo è quello di calcolare la correlazione con la temperatura per ognuno di questi.
Viene quindi chiamato il metodo filter(n) da **milano_poi_zones.py**. Questo ci permette di selezionare il cluster di interesse (punto [4]).
Il numero da assegnare ad ogni cluster lo recuperiamo dal dataframe df e dal grafico della pagina antecedente:

	poi	square_id	kmeans
69	3	5061	2
70	1	5065	0
71	1	5067	0
72	2	5070	3
73	1	5091	0
74	1	5151	0
75	1	5152	0
76	3	5153	2
77	5	5155	1
78	2	5156	3
79	3	5157	2
80	9	5158	4
81	6	5159	1
82	2	5160	3
83	7	5161	1
84	3	5162	2
85	2	5163	3

- Zone che ha un numero di POI pari 1 hanno il valore di kmeans pari a 0 (**Very Low**)
- Zone che hanno due POI corrispondono al cluster 3 (**Low**)
- Zone che hanno un numero di POI compreso fra 3 e 4 situtati nel cluster 2 (**Medium**)
- Zone che hanno un numero di POI compreso fra 5 e 7 hanno kmeans pari a 1 (**High**)
- Zone che hanno un numero di POI compreso 8 e 10 hanno kmeans pari a 4 (**Very High**)

Nel file **mobile_temperature_correlation.py** il risultato della chiamata filter(n) viene utilizzato per individuare i quadrati che compongono la griglia di Milano che fanno parte di un determinato cluster.
Ad esempio, se siamo interessati al cluster delle zone di Milano che hanno pochi punti di interesse:

```
lowDensityList = pois.filter(3)
cdr = cdr.filter(cdr.square_id.isin(lowDensityList))
```

Vengono infine effettuati tutti i calcoli per la correlazione. Avremo per ogni zona i seguenti coefficienti che segnalano il legame tra temperatura ed attività di telecomunicazioni:

	Very Low Poi	Low Poi	Medium Poi	High Poi	Very High Poi
SMS	**0.166746109135**	**0.166746109135**	**0.207706702437**	**0.250463196287**	**0.250463196287**
Chiamate	**0.203364241162**	**0.203364241162**	**0.206108058771**	**0.23404044233**	**0.23404044233**
Internet	**0.207180425008**	**0.207180425008**	**0.229573317949**	**0.268738889585**	**0.268738889585**

I seguenti valori sono stati ottenuti sostituendo ad ogni iterazione al metodo filter il valore n.
Si osservi il risultato per ogni riga: A mano a mano che i punti di interesse aumentano, progressivamente aumenta anche il coefficiente di correlazione. Notare come inoltre agli estremi la differenza sia minima.
La conclusione è che In prossimità di luoghi con elevato numero di punti di interesse, gli abbonati sono più propensi all'utilizzo della rete e questo è strettamente correlato con la temperatura.
Per quanto riguarda le precipitazioni vogliamo analizzare come varia il comportamento degli utenti in base alla loro intensità (alta, moderata, leggera, assenza totale).
Una domanda comune che si pone su due o più dataset è se sono diversi o meglio se la differenza tra la loro tendenza (ad esempio media o mediana) è statisticamente significativa.
È possibile rispondere a questa domanda per campioni di dati che non hanno una distribuzione gaussiana utilizzando test di statistica non parametrici. L'ipotesi nulla di questi test è spesso l'assunzione che entrambi i campioni sono stati prelevati da una popolazione con la stessa distribuzione e quindi essi abbiano gli stessi parametri di popolazione, come media o mediana.
Se dopo aver calcolato il test di significatività su due o più campioni l'ipotesi nulla viene rifiutata, vuol dire che ci sono prove che suggeriscono che i campioni sono stati prelevati da diverse popolazioni e, che a loro volta, la differenza tra le stime campionarie dei parametri della popolazione, come medie o mediane potrebbe essere significativa.
Per fare ciò è stato utilizzato il test di Kruskal-Wallis.
Il test restituisce anche un valore p che viene utilizzato per interpretare il risultato.
Il valore p può essere interpretato nel contesto di un livello di significatività scelto chiamato alfa. Un valore comune per alpha è 5% o 0.05. Se il valore p è inferiore al livello di significatività, il test ci dice che ci sono prove sufficienti per rifiutare l'ipotesi nulla e che i campioni sono stati probabilmente estratti da popolazioni con distribuzioni differenti.

- **valore-p ≤ alfa:** risultato significativo, rifiuto dell'ipotesi nulla (H0), le distribuzioni differiscono.
- **valore p> alfa:** risultato non significativo, non si riesce rifiutare l'ipotesi nulla (H0), le distribuzioni sono le stesse.

Come si procede:
• Si ordinano i dati dei campioni in un'unica serie, in ordine crescente (nel nostro caso per data).
• Si individua la mediana dei dati osservati.
• Per ogni campione si contano le osservazioni il cui valore supera quello della mediana comune.
Quando il test H di Kruskal-Wallis porta a risultati significativi, allora almeno uno dei campioni è diverso dagli altri campioni.
Tuttavia, il test non identifica dove si verificano le differenze. Inoltre, non identifica quante differenze si verificano.
L'H test di Kruskal-Wallis puo' essere implementato con SciPy e viene utilizzato per ogni quadrato della griglia di Milano.
C'è una relazione significativa tra internet e i livelli di intensità delle precipitazioni come si può vedere dal risultato prodotto dal file **mobile_precipitation_correlation.py**.
In questo file è stata, in primo luogo, impiegata una suddivisione dei valori di intensità delle piogge in intervalli a cui assegnare un valore per contraddistinguerli. Ne sono stati ipotizzati quattro:

- Piogge assenti, intensità nulla: **0**
- Piogge scarse, valori compresi fra 0 e 2.6 mm: **1**
- Piogge moderate, valori compresi fra 2.6 e 7.6 mm: **2**
- Piogge consistenti, valori superiori a 7.6 mm: **3**

Sono state fatte poi le seguenti ipotesi semplificative:

- Le precipitazioni influenzano la vita delle persone quando queste sono in attività. Difficilmente di notte, quando la maggior parte della popolazione dorme, si hanno delle variazioni. Per questo motivo è stato preso in esame l'intervallo di tempo (7-20)

- La popolazione si comporta diversamente durante la settimana e nel weekend. E' stato dimostrato nei capitoli precedenti e per questo i dati devono essere distinti questi due casi perché la distribuzione della variabile relativa alla connessione ad internet avrà una distribuzione diversa. Per effettuare questa distinzione è stata impiegata, come nei capitoli precedenti, la funzione "weekday".

Fatte queste ipotesi, avendo distinti degli intervalli in cui suddividere l'intensità delle piogge, occorre trovarne altri per la suddivisione dei valori di internet. Questa operazione deve essere fatta due volte perché questa variabile avrà comportamenti diversi durante la settimana e il weekend. Viene inoltre fatta un'ulteriori semplificazione:
Si considerano solo le prime due cifre un aumento consistente dei consumi solo per valori consistenti
Gli intervalli ipotizzati sono i seguenti:

- Utilizzo assente: **0**
- Utilizzo scarso, valori compresi fra 1 e 40: **1**
- Utilizzo moderato, valori compresi fra 40 e 50: **2**
- Utilizzo consistente, valori compresi fra 50 e 60: **3**

Per il nostro test, prendiamo in considerazione tutto il dataset relativo al mese di novembre.
Escludiamo i record in cui le piogge sono assenti e otteniamo i seguenti risultati:
Per i giorni lavorativi, il test di Krunstall-Wallis risulta superato. La distribuzione dei valori è molto simile e si dimostra come all'aumentare delle piogge corrisponde anche un aumento dell'utilizzo della rete

date_istant	rain_intensity	SMS_total	Call_total	internet	weekday	internet_level
2013-11-04 10:00:00	1.0	881524.1832279834	921824.7625742243	56.0	2.0	2
2013-11-04 12:00:00	1.0	832998.7312579572	860330.804398153	59.0	2.0	2
2013-11-04 13:00:00	1.0	803898.4771432871	772368.7569644863	62.0	2.0	3
2013-11-19 08:00:00	1.0	669085.7720874468	522760.39364341466	48.0	3.0	1
2013-11-19 09:00:00	2.0	798527.397467872	830039.1161616452	54.0	3.0	2
2013-11-19 10:00:00	2.0	868264.7786540166	921211.5800559886	55.0	3.0	2
2013-11-19 11:00:00	2.0	909907.8595760746	922789.3631899476	57.0	3.0	2
2013-11-19 12:00:00	2.0	888643.1094679004	874157.1249483847	59.0	3.0	2
2013-11-20 16:00:00	2.0	854987.8776848039	934282.1016954988	57.0	4.0	2

```
[Stage 33:>                                                         (0 + 4) / 85][1. 1. 1. 1. 2. 2. 2. 2. 2.]
[Stage 42:>                 (0 + 4) / 85][Stage 44:>                 (0 + 0) / 4]Kruskal Wallis H-test test:
('H-statistic:', 2.917378917378907)
('P-Value:', 0.08763010059774853)
Accept NULL hypothesis - No significant difference between groups.
```

Durante il weekend invece, sebbene il p-value sia molto vicino al valore alpha il test non risulta andare a buon fine.
Bisogna considerare tuttavia che record con precipitazioni sono piuttosto scarsi e per avere un test più preciso occorrerebbe disporre di dataset con un intervallo temporale più ampio.

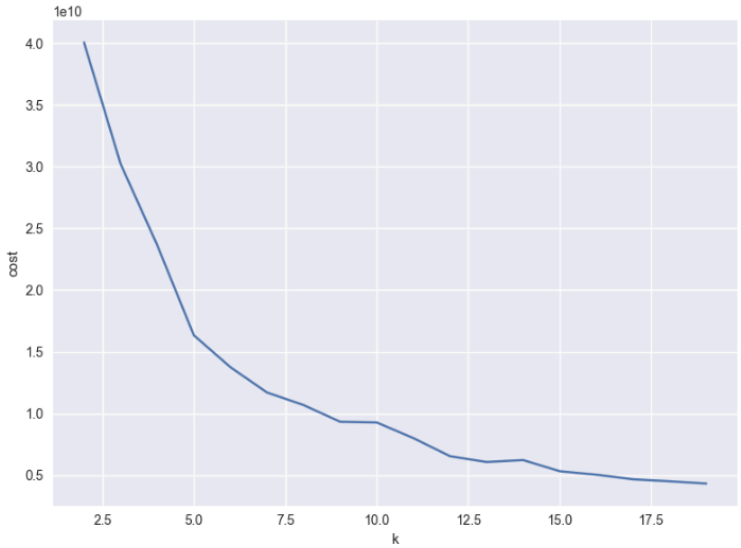
date_istant	rain_intensity	SMS_total	Call_total	internet	weekday	internet_level
2013-11-15 08:00:00	1.0	672005.4500927367	531329.9858295269	49.0	6.0	1
2013-11-15 09:00:00	2.0	810688.551019229	822500.4878010997	53.0	6.0	2
2013-11-15 10:00:00	1.0	859455.4603600646	913133.5226560896	55.0	6.0	2
2013-11-15 11:00:00	1.0	840295.1728096082	943449.1620407307	56.0	6.0	2
2013-11-15 12:00:00	1.0	904840.0608443596	894069.4554061163	58.0	6.0	2
2013-11-15 14:00:00	1.0	795208.4678138718	830941.0001479517	60.0	6.0	3
2013-11-30 12:00:00	1.0	586907.5328481345	623212.1772067926	43.0	7.0	1
2013-11-30 13:00:00	1.0	526754.0391553837	490327.1804623788	43.0	7.0	1

```
[Stage 57:>                 (0 + 4) / 85][Stage 59:>                 (0 + 0) / 4][1. 2. 1. 1. 1. 1. 1. 1.]
Kruskal Wallis H-test test:
('H-statistic:', 4.125000000000006)
('P-Value:', 0.04225402065442203)
Reject NULL hypothesis - Significant differences exist between groups.
```

Data l'elevata latenza dell'operazione (circa due ore solo per il dataset di novembre) si è scelto di non proseguire il test includendo anche il mese di dicembre.

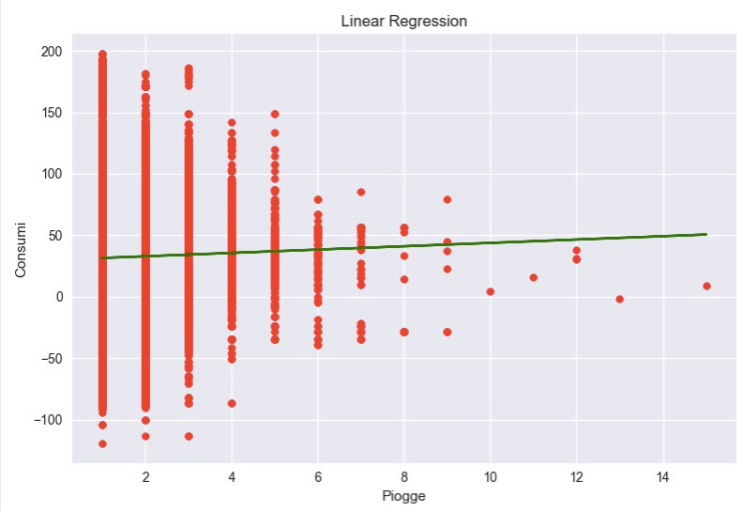
9. Analisi dell'utilizzo della corrente in Trentino Alto Adige attraverso Spark MLlib (Clustering e Linear Regression)

Il dataset presente all'interno della cartella electricity ci permette di ricavare il modello di consumo energetico di gruppi di persone con una data fase temporale (ad esempio, ogni ora) in un dato intervallo di tempo (ad esempio mensile).
Dobbiamo chiederci quale tipo di analisi sia possibile in tale contesto. Possiamo ad esempio cercare di comprendere le relazioni tra le osservazioni. Un approccio che possiamo usare in tali situazioni è quella della cluster analysis o clustering.
L'obiettivo del clustering è quello di verificare, date le features in input, se le osservazioni disponibili ricadono all'interno di gruppi relativamente distinti tra di loro.
Se il modello di consumo di ciascun gruppo è regolare durante tale intervallo di tempo per diverse ubicazioni in un quadrato della griglia di Trento, questo può essere definito con dei nuovi gruppi attraverso il clustering, grazie al quale è possibile indirizzare delle offerte ad-hoc.
Una Smart City può essere definita come un centro urbano efficiente e sostenibile che assicura un'elevata qualità della vita ottimizzando le sue risorse. La gestione energetica è uno dei problemi più impegnativi all'interno di questi centri urbani.
In genere k-means converge ad un ottimo locale. L'algoritmo è molto sensibile all'inizializzazione dei centroidi.
In un gruppo, tuttavia, in genere, anche lo stesso consumatore non ha un uso regolare degli apparecchi elettronici in giorni diversi. Per questo la caratterizzazione di gruppi di consumatori non è semplice. Pertanto, provare e prevedere il consumo orario esatto di un gruppo di persone su un orizzonte temporale di più giorni potrebbe essere un obiettivo mal formulato.
Il dataset fornito da Open Data purtroppo non fa riferimento alla città di Milano ma a quella di Trento. Si è scelto di operare una semplice analisi al fine di poter confrontare per le operazioni di Machine Learning MLlib e Sci kit learn,
In quanto gruppi quindi possiamo suddividere i consumatori in base al flusso di corrente e alla quantità di ubicazioni presenti in un quadrato della griglia?
Nel file **trento_current_consumption**, dopo aver elaborato il dataset **"line.csv"**, che descrive le linee di distribuzione di corrente, e **"SET-nov-2013.csv** "che descrive l'intensità di corrente ogni dieci minuti, viene effettuata la join e il raggruppamento per ogni ora dei record (punto 2).
Con poche righe è stato possibile riprodurre il lavoro fatto nel capitolo precedente con scikit learn utilizzando MLlib.
Attraverso l'Elbow Method, anche in questo caso, è stato individuato il numero k di cluster prendendo in considerazione come unica features l'intensità di corrente (punto 3).
La curva a gomito è rappresentata dal seguente grafico:



Viene aggiunta nel dataframe una nuova colonna che descrive per ogni record il cluster di appartenenza. Sono in totale 15 i raggruppamenti in cui è possibile suddividere la popolazione di Trento.
Attraverso MLlib è possibile inoltre utilizzare la regressione lineare per prevedere i consumi di corrente in corrispondenza degli aumenti dell'intensità delle piogge.
Matematicamente la regressione può essere intesa come il trovare la funzione che meglio approssima la relazione tra la variabile indipendente X (l'input) e la variabile dipendente Y (l'output). Nel caso di una regressione lineare questa funzione è un semplice polinomio: $f(x) = xw + b$
Nella cartella electricity è possibile individuare il file precipitation-trentino.csv che associa ad ogni quadrato, per ogni data con intervalli di dieci minuti, l'intensità delle piogge.
L'utilizzo di MLlib si è rivelato molto efficiente vista l'estensione dei dataset.

Precedentemente nel progetto si è scelto di operare scikit learn per un motivo ben preciso: ha il supporto per Pandas e Matplotlib, il che rende il processo di sviluppo di modelli di apprendimento automatico molto iterativo ma soprattutto permette di visualizzare i dati con maggiore facilità.
Dopo aver raggruppato per media oraria (punto 2 e 4) si procede nello stabilire il legame che c'è tra intensità di corrente e intensità delle piogge.
Ci sono due termini che trovi spesso in ML: "Feature" e "Label".
Nell'apprendimento automatico e nella pattern recognition, una feature è una proprietà misurabile caratteristica di un fenomeno osservato. La scelta delle features è un passaggio cruciale per algoritmi efficaci di riconoscimento, classificazione e regressione.
In termini semplici, le features sono tutte le variabili indipendenti ci possono aiutare a prevedere i valori della variabile dipendente (in questo caso la corrente detta label).
Data la nostra feature che corrisponde all'intensità delle piogge e data la label che corrisponde al flusso di corrente, otteniamo il seguente grafico:



RMSE e MSE stanno per root-mean square error e mean square error.
Un RMSE che si avvicina a 1 dimostra per la variazione dell'intensità delle piogge spiega la variabile dei consumi.
Nel nostro caso, il limite dei dati a disposizione con il solo mese di novembre, ma soprattutto la scarsità di esempi in cui l'intensità delle piogge è pressochè invariata, non ci permette di effettuare un'analisi precisa.
Scegliendo un insieme di test randomicamente infatti non è possibile allenare il modello in maniera corretta.
I dati a disposizione non rendono possibile prevedere l'aumento o meno dei consumi di corrente a fronte delle variazioni delle precipitazioni.
Il valore di RMSE ottenuto infatti è 19.7799876387 ben superiore al desiderato.
L'esperienza comune ci induce a pensare che la relazione tra le due variabili non sia proprio lineare.
In genere, all'aumentare dell'intensità della pioggia ci aspettiamo infatti che il flusso di corrente aumenti ma non in modo esattamente proporzionale.

9.Riferimenti Bibliografici

1. Standard Score: https://statistics.laerd.com/statistical-guides/standard-score-2.php

2. Open Big Data

3. TripAdvisor CSV Attraction

4. Barlacchi, G. et al. A multi-source dataset of urban life in the city of Milan and the Province of Trentino. Sci. Data 2:150055 doi: 10.1038/sdata.2015.55 (2015).

5. Can be temperature be used as a predictor of data traffic: a real network big data analysis – muhammad nauman rafiq, hasan farooq, ahmed zoha and ali imran

6. Bajardi, Paolo & Delfino, Matteo & Panisson, Andre & Petri, Giovanni & Tizzoni, Michele. (2015). Unveiling patterns of international communities in a global city using mobile phone data. EPJ Data Science. 4. 10.1140/epjds/s13688-015-0041-5.

7. Brdar S. et al. (2019) Big Data Processing, Analysis and Applications in Mobile Cellular Networks. In: Kołodziej J., González-Vélez H. (eds) High-Performance Modelling and Simulation for Big Data Applications. Lecture Notes in Computer Science, vol 11400. Springer, Cham

8. The effect of weather on user-generated big geo data in mobile phone networks Carolina Arias Muñoz Maria Antonia Brovelli

9. Discovering Electricity Consumption Over Time for Residential Consumers Through Cluster Analysis Tania Cerquitelli , Gianfranco Chicco, Evelina Di Corso , Francesco Ventura , Giuseppe Montesano, Anita Del Pizzo, Alicia Mateo González, Eduardo Martin Sobrino

10. Amri, Yasirli & Lailatul Fadhilah, Amanda & , Fatmawati & Setiani, Novi & Rani, Septia. (2016). Analysis Clustering of Electricity Usage Profile Using K-Means Algorithm. IOP Conference Series: Materials Science and Engineering. 105. 012020. 10.1088/1757-899X/105/1/012020.

11. Federico Plazzi – anova ANalysis of VAriance

12. Discover how to Transform Data into Knowledge with Python - Jason Brownlee