Paper
Data Management & Data Visualization

Davide Lagano *matricola 838358*
Davide Sangalli *matricola 848013*
Diana Tenca *matricola 789651*

Anno Accademico 2018-2019

Estendendosi per oltre 6000 chilometri, le autostrade costituiscono la principale via di comunicazione su ruote di tutto il nostro paese, collegandolo da Nord a Sud. Vista l'importanza del loro ruolo sia per i trasporti che per l'economia, monitorarne il traffico è un fattore cruciale.
Per i motivi sopracitati, questo progetto è incentrato sulla raccolta e la preparazione di dati relativi al ritardo medio di percorrenza delle principali autostrade italiane.
Lo scopo è quello di fornire all'utente una visione complessiva della situazione e di preparare i dati per poter, eventualmente, applicare dei modelli predittivi.
Dopo aver osservato le autostrade di maggior interesse si è proceduto alla selezione dei caselli da monitorare, scelti in modo progressivo ad una distanza di circa 50 chilometri, tali da permettere una visione complessiva soddisfaciente ma che nella visualizzazione grafica non risultassero troppi e confusionari. In seguito, utilizzando un'architettura *producer-consumer*, sono stati raccolti i dati in tempo reale durante tutto il mese di maggio 2019 e, per migliorare la rappresentazione grafica, anche le coordinate geografiche dei caselli selezionati.
Al termine della fase di raccolta dati, dopo averli puliti ed integrati, sono state effettuate una serie di *query* per mettere in luce aspetti interessanti che permettano all'utente di spaziare da una visione più generale ad una più dettagliata, osservando i ritardi per ogni tratta in relazione al mese, al giorno della settimana, al giorno specifico del mese e alla fascia oraria.

Indice

1 Data Management

La prima parte del progetto si è focalizzata sulla raccolta e sulla preparazione dei dati. Come è stato precedentemente detto, si è deciso di utilizzare la *velocity* e, per questa ragione, è stata utilizzata un'architettura lambda(kafka) per la raccolta dei dati relativi ai ritardi. In aggiunta, per rappresentare graficamente i caselli ne sono state raccolte le coordinate dal sito openstreetmap.org così da ottenere la *variety* richiesta. Di seguito verranno descritte le diverse fasi di raccolta, di pulizia ed integrazione dei dati che sono state condotte.
I dati sono stati successivamente caricati nel database Mongo DB dove sono state effettuate le opportune query per le rappresentazioni grafiche.

1.1 Raccolta dei Dati in Tempo Reale

Al fine di permettere la raccolta dei dati in tempo reale, oltre ad utilizzare l'architettura lambda, *Kafka*, è stato necessario utilizzare una macchina virtuale che permettesse allo script di raccolta di mantenersi sempre connesso.
Inizialmente è stata utilizzata la macchina virtuale data in concessione dall'università ma, avendo la necessità di rimanere connessi ventiquattro ore su ventiquattro, si è deciso di cercarne una alternativa. La scelta è ricaduta su quella di *Amazon AWS* che si è dimostrata particolarmente adatta per le esigenze del progetto. Sono state utilizzate tre macchine virtuali, una per ogni componente del gruppo ma, essendo la memoria RAM in dotazione di un 1GB, è stato comunque necessario aggiungere una partizione di swap usata come memoria.
Per terminare la preparazione della macchina, è stato installato Kafka e aggiunte alcune librerie Python necessarie per lo script di raccolta.

Per la raccolta in tempo reale era necessario trovare un sito che fornisse un'API tale da ottenere i dati con la frequenza richiesta. Valutando le diverse possibilità presenti online si è deciso di utilizzare l'API di *www.here.com*. Si tratta di un'impresa di coproprietà delle azienda automobilistiche Audi, BMW e Daimer che fornisce servizi e tecnologie di dati di tipo geografico e di mappatura per il settore automobilistico, consumer e aziendale. I dati raccolti si poggiano sul sistema GPS integrato in molti modelli di automobile che permette di monitorare gli spostamenti e i tempi di percorrenza dei veicoli.
Il sito forniva un ID ed un Codice che permettevano, una volta integrati nello script, di inviare le richieste ed ottenere le risposte.
Terminata la ricerca degli strumenti preliminari per l'attuazione della raccolta, si è proceduto all'implementazione dello script vero e proprio. Lo script Python è stato implementato in modo da inviare, utilizzando le opportune chiavi, le richeste al sito ogni cinque minuti.
Mediante l'utilizzo dell'architettura lambda, la cui costruzione verrà descritta successivamente, la funzione principale del main, *run_ script*(Figura 3), stabilisce una connessione con il sito dell'API e

```
HERE_APP_ID="B93QbhRRNtRuEoD24lKT"
HERE_CODE="4iQZzY41qxJ0la1nTs6HAg"
```

Figura 1: Esempio di ID e CODE forniti dall'API

invia le richieste per la raccolta. Se tale connessione risulta attiva, i dati vengono scaricati, aggiunti ad una lista precedentemente definita ed infine inviati al consumer. Nel caso di mal funzionamento ossia di mancata connessione con il sito, la funzione restituisce errore ed interrompe la raccolta uscendo dal ciclo while. Come ultimo passaggio i dati vengono salvati in file csv temporanei che sono stati successivamente caricati in Mongo DB e utilizzati come base per la costruzione del database conclusivo.

```
def run_script(origin,destination,highway):
    producer,consumer,topic_name=setup_kafka(highway)
    while(True):
        send_list=[]
        status,timestamp,distance,base_time,traffic_time=get_traffic_real_time(origin,destination)
        if(status=="Connection Established"):
            print("Everything went fine :)")
            send_list.append(timestamp)
            send_list.append(distance)
            send_list.append(base_time)
            send_list.append(traffic_time)
            producer.send(highway,str.encode(str(send_list)))
        else:
            print("Something went wrong...please retry")
            break

        write_file(origin,destination,highway,response.value)

        time.sleep(300)
```

Figura 2: Funzione *run_ script* richiamata nel main

Il punto cruciale della fase di raccolta è stato l'utilizzo di un'architettura di tipo *producer-consumer*. La funzione *setup_ kafka*, presentata in figura 4, definisce la struttura dell'architettura stessa: preso in input il nome della tratta, per ognuna è stato creato un producer con il relativo consumer a cui vengono inviati i dati dopo l'interrogazione.
I dati ottenuti dal sito dell'API fornivano molteplici campi, tuttavia è stato deciso di focalizzarsi esclusivamente su quelli di interesse per l'analisi, ossia:

Timestamp

Distance

Base Time

```
def setup_kafka(topic_name):
    producer=KafkaProducer(bootstrap_servers="localhost:9092",api_version=(0, 10, 1))
    consumer=KafkaConsumer(bootstrap_servers="localhost:9092",api_version=(0, 10, 1),
                           auto_offset_reset="latest",consumer_timeout_ms=1000)
    consumer.subscribe(topic_name)
    return producer,consumer,topic_name
```

Figura 3: Definizione del setup di Kafka

Traffic Time

Il *timestamp* è una semplice indicazione temporale relativa alla rilevazione dei dati, la *distance* è la lunghezza, espressa in metri, della tratta osservata ed il *base time* e il *traffic time* sono i tempi di percorrenza, entrambi espressi in secondi, della tratta, uno in condizioni normali e l'altro in presenza di traffico. Più precisamente, il *base time* viene calcolato supponendo di poter percorrere la tratta alla massima velocità consentita mentre il *traffic time* è una media delle velocità di percorrenza della tratta rilevate dai dispositivi dei diversi utenti.

Volendo lanciare parallelamente gli script relativi alle diverse tratte, sono stati scritti dei file *bash* che, presi in input quattro parametri(nome del file da cui leggere le informazioni, luogo di origine, destinazione e nome della cartella in cui salvare), permettessero di eseguire le raccolte simultaneamente. Ovviamente, sono stati caricati sulla macchina virtuale e resi eseguibili i documenti contenenti le informazioni necessarie alla corretta esecuzione dei file *bash*. La raccolta dei dati è durata un mese: dal **1 Maggio 2019 al 31 Maggio 2019**.

```
def get_traffic_real_time(origin,destination):

    origin_lat,origin_long = get_coordinates_origin(origin)
    dest_lat,dest_long = get_coordinates_destination(destination)

    file_date_2=datetime.datetime.now()
    response=requests.get("https://route.api.here.com/routing/7.2/calculateroute.json?waypoint0={}%2C{}&waypoint1={}%2C{}&mode=shortest%3Bcar%3Btraffic%3Aenabled&app_id={}&app_code={}&de
                format(origin_lat,origin_long,dest_lat,dest_long,HERE_APP_ID,HERE_CODE))

    if response.status_code==200:
        status="Connection Established"
    else:
        status="Connection Failed"

    test=response.json()
    timestamp,distance,base_time,traffic_time=parse_json(test)

    return status,timestamp,distance,base_time,traffic_time
```

Figura 4: Esempio di richiesta inviata al sito

Terminata la prima fase di raccolta dati, ci si è focalizzati su quella di preprocessing al fine di permettere una buona visualizzazione delle informazioni raccolte.
I dati, salvati in file csv, dovevano essere integrati e sistemati poichè presentavano alcune problematiche che avrebbero potuto inficiare il lavoro successivo. Per prima cosa le unità di misura dei campi *distance*, *base time* e *traffic time* dovevano essere cambiate rispettivamente in chilometri ed

in minuti. Inoltre il *timestamp*, automaticamente sincronizzato con il meridiano di Greenwich, è stato modificato, in linea con il fuso orario italiano.
La parte più onerosa della fase di preprocessing è stata quella di integrazione. Nonostante la macchina fosse sempre connessa, all'interno dei dati erano presenti dei *missing value*. Inizialmente si è cercato di effettuare l'integrazione utilizzando lo storico ma, poichè il sito non forniva accesso ai dati, si è deciso di utilizzare la metotodologia sostitutiva del *most-frequent*. Il valore mancante è stato sostituito con la moda dei valori, in riferimento alla tratta e alla fascia oraria, in questo modo le successive analisi circa la media non sono state compromesse.

Per quanto concerne i dati relativi alle coordinate geografiche, sono stati collezionati con due modalità differenti.
Volendo rappresentare le tratte analizzate e quindi necessitando esclusivamente delle coordinate relative ai caselli monitorati, è stato utilizzato *geopy*. I dati, in formato csv, sono stati utilizzati nella fase di rappresentazione per permettere all'utente di visualizzare la tratta presa in analisi.
Tuttavia, poichè tali informazioni consentono solo di disegnare l'autostrada in modo approssimato, si è deciso di raccogliere i dati di più caselli effettuando le query opportune sul sito *openstreetmap.org*. Anche in questo caso i dati, questa volta salvati in formato *json*, sono stati utilizzati nelle visualizzazioni.

1.2 Dati in Mongo DB

I dati, puliti ed integrati, dovevano essere caricati su un database.
Inizialmente è stato preso in considerazione l'utilizzo di *Neo4j*: impostando i nodi come caselli e le tratte come archi sarebbe stato possibile costruire la struttura a grafo, ma non sarebbero state sfruttate le caratteristiche del database visto l'unicità delle relazioni. Piuttosto che optare per questa soluzione si è deciso di valutare altri database come *Mongo DB*.
La scelta di *Mongo DB* come database è dovuta alla struttura dei dati raccolti: i file relativi alle tratte e ai giorni erano raggruppati per autostrada di appartenza. Tale conformazione poteva facilmente essere riportata in *Mongo DB*.

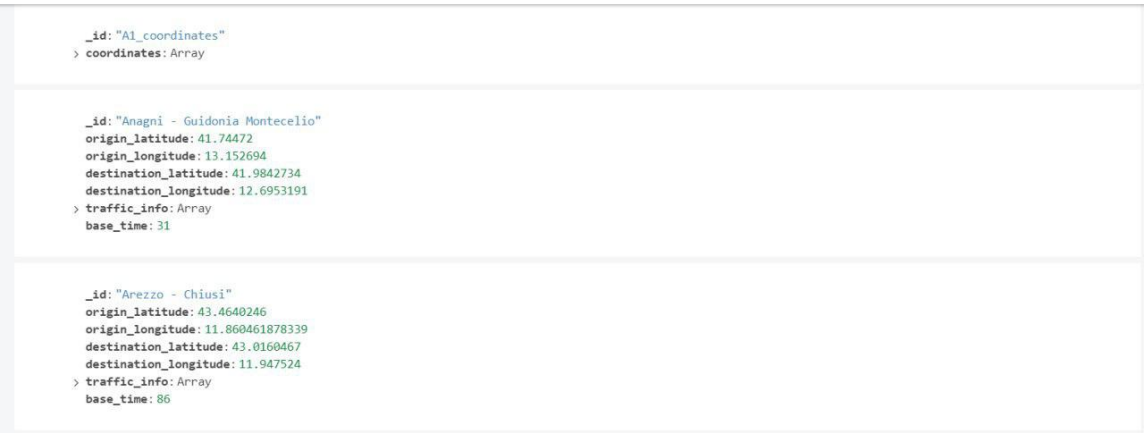


Figura 5: Esempio di documento salvato in *Mongo DB*

Si è infatti deciso di definire le autostrade come collezioni di documenti, ognugno dei quali così strutturato: nome della tratta, coordinate geografiche dell'origine della tratta, quelle del luogo di destinazione,il *base time* e un array denominato *traffic info* contenente tutti i dati relativi alla tratta ossia, nello specifico, la data, l'ora, il giorno della settimana, la distanza espressa in chilometri e i minuti di ritardo.

Una volta ottenuto il *dump* del database sono state effettute una serie di query per estrarre delle informazioni aggiuntive.

```
def ritardo_medio_per_hour(filename,collection,time_start,time_end,month=True):
    mycol=db[collection]
    time_start=datetime.datetime.strptime(time_start,"%H:%M:%S")
    time_end=datetime.datetime.strptime(time_end,"%H:%M:%S")

    if(month==False):
        query= [{"$unwind":"$traffic_info"},
                {"$match": {
                    "$and": [
                    { "traffic_info.time":{"$gte":time_start}},
                    { "traffic_info.time":{"$lte":time_end}}
                   ]
                }},
                    {"$group": { "_id":{"path":"$_id", "from-to":datetime.datetime.strftime(time_start,'
                                        +"-"+datetime.datetime.strftime(time_end,"%H:%M:%S"),
                                        "day":"$traffic_info.date"},"avgDelay": {"$avg":"$traffic_info.(

    else:
        query= [{"$unwind":"$traffic_info"},
                {"$match": {
                    "$and": [
                    { "traffic_info.time":{"$gte":time_start}},
                    { "traffic_info.time":{"$lte":time_end}}
                   ]
                }},
                    {"$group": { "_id":{"path":"$_id", "from-to":datetime.datetime.strftime(time_start,'
                                +"-"+datetime.datetime.strftime(time_end,"%H:%M:%S")},
                                "avgDelay": {"$avg":"$traffic_info.delay_minutes"}}}]

    if not (list(mycol.aggregate(query))):
        print("Data not available")
    else:
        df=pd.DataFrame(list(mycol.aggregate(query)))
        with open("C:/Users/dasan/Desktop/query_new/{}.json".format(filename),'a+') as json_file:
            df.to_json(json_file,orient='table')
```

Figura 6: Esempio di query effettuata in *Mongo DB*

In particolare sono state calcolate:

- Medie dei ritardi relativi all'autostrada e alle singole tratte su tutto l'arco temporale
- Medie dei ritardi relativi all'autostrada e alle tratte calcolate rispetto alla fascia oraria su tutto l'arco temporale

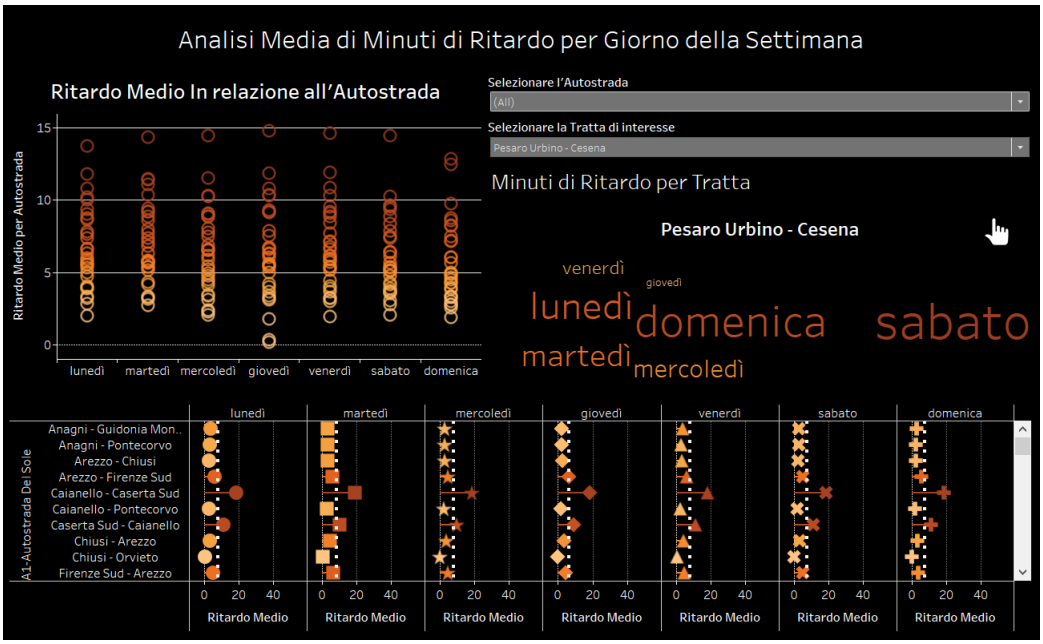
- Media dei ritardi relativi all'autostrada e alle singole tratte calcolate rispetto al giorno della settimana
- Media dei ritardi relativi all'autostrada e alle singole tratte calcolate sia rispetto al giorno specifico del mese che alla fascia oraria.

In figura 6 è presentato un esempio di query effettuata su *Mongo DB* utilizzando la libreria Python *PyMongo*.

2 Data Visualization

La seconda parte del progetto è consistita nella rappresentazione grafica dei dati. Lo scopo era quello di fornire all'utente la visione del problema da angolazioni fornendo una visione più complessiva ed una più dettagliata della situazione.
In riferimento alle query effettuate sul dump di *Mongo DB* sono state realizzate quattro visualizzazioni in *Tableau*.

2.1 Prima Visualizzazione

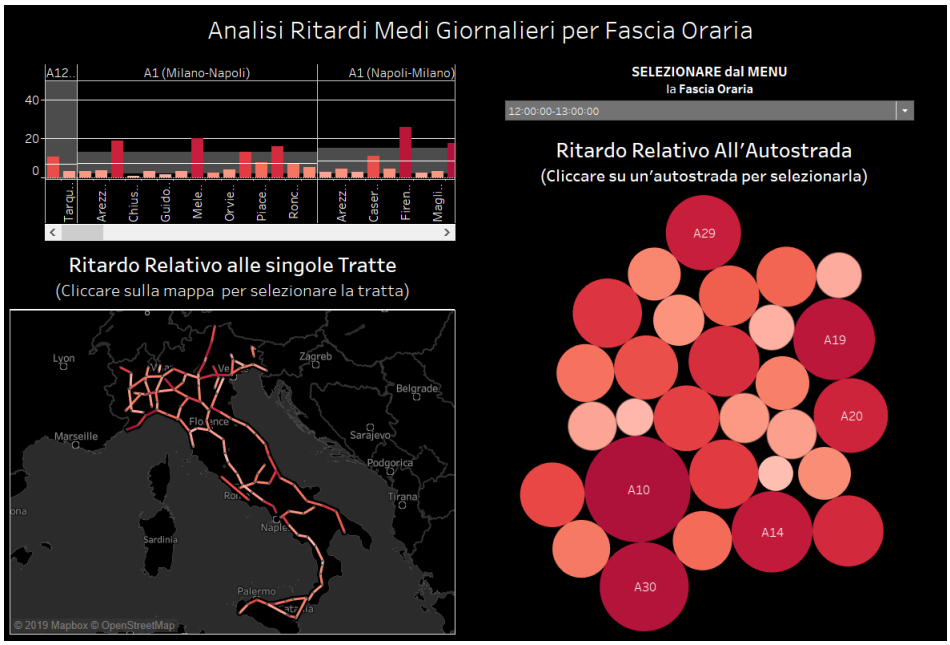


La prima rappresentazione, basata sui ritardi calcolati rispetto al giorno della settimana, fornisce all'utente una panoramica sulla settimana. Lo scatterplot in alto rappresenta il ritardo relativo all'autostrada, mentre gli altri due grafici permettono di analizzare le singole tratte. In particolare, si è avvertita l'esigenza di inserire sia il lollipop chart che il wordchart per permettere all'utente sia di confrontare le tratte di una stessa autostrada che i giorni rispetto ad una sola tratta.

Sottoponendo la visualizzazione agli utenti sono state riscontrate alcune problematiche, non sempre completamente risolte.

- Il filtro sulle tratte è applicato solo al wordchart: per come è strutturato il grafico è fondamentale che sia selezionata sempre una e una sola tratta. Applicando il filtro anche al lollipop chart si precludeva la possibilità di selezionare tutte le tratte di un'autostrada per poterle confrontare, inficiando l'utilità del grafico.
  Per ovviare al problema il wordchart è stato impostato come filtro dell'intera dashboard così da poter, cliccando sul nome della tratta, selezionarla anche nel grafico sottostante. Temendo che potesse non essere chiaro all'utente è stata inserita un'immagine aggiuntiva per suggerire l'azione.
- L'utilizzo del wordchart come filtro all'interno della dashboard fa sì che, selezionando uno dei giorni della settimana nel grafico, anche gli altri due mostrino solo i dati relativi al giorno selezionato, impedendo all'utente di avere la visione complessiva.
- Il wordchart non sempre è ben centrato: per alcune tratte, specialmente quelle associate a ritardi molto vicini e quindi a grandezze simili dei giorni della settimana, il wordchart non risulta completamente visibile nella dashboard. Modificando le dimensioni nello sheet e impostando l'opzione *"entire view"* sia nello sheet che nella dashboard si è limitato il problema.

2.2 Seconda Visualizzazione



In questo secondo caso si vogliono analizzare le diverse fasce orarie sia per l'autostrada che in relazione alle varie tratte. La domanda e l'indagine che si vuole compiere è la seguente: nell'immaginario collettivo le fasce orarie della mattina e della sera sono quelle associate al ritardo maggiore,

ma questo trova corrispondenza nei dati?
Anche in questo caso la dashboard è in parte incentrata sui dati relativi alle autostrade ed in parte su quelli delle singole tratte, così da permettere un'analisi generale ed una più dettagliata. Si è deciso di inserire una mappa per fornire all'utente una visione approssimata dell'autostrada e della tratta studiata.
Questa è stata la visualizzazione più criticata dagli utenti, e che in seguito ha subito il maggior numero di cambiamenti:

- In primo luogo, gli utenti si aspettavano di poter selezionare l'autostrada di interesse cliccando sul bubble chart, opzione non possibile nella precedente versione della visualizzione. Si è deciso di introdurre questa possibilità, ma eliminando il menù di selezione. Nonostante sarebbe stato possibile lasciarli entrambi, si è preferito eliminarlo, poichè nell'esecuzione dei task si è notata la tendenza degli utenti a scegliere un'autostrada dalla lista senza prima deselezionare quella evidenziata con il bubble chart, causando il crash della dashboard.

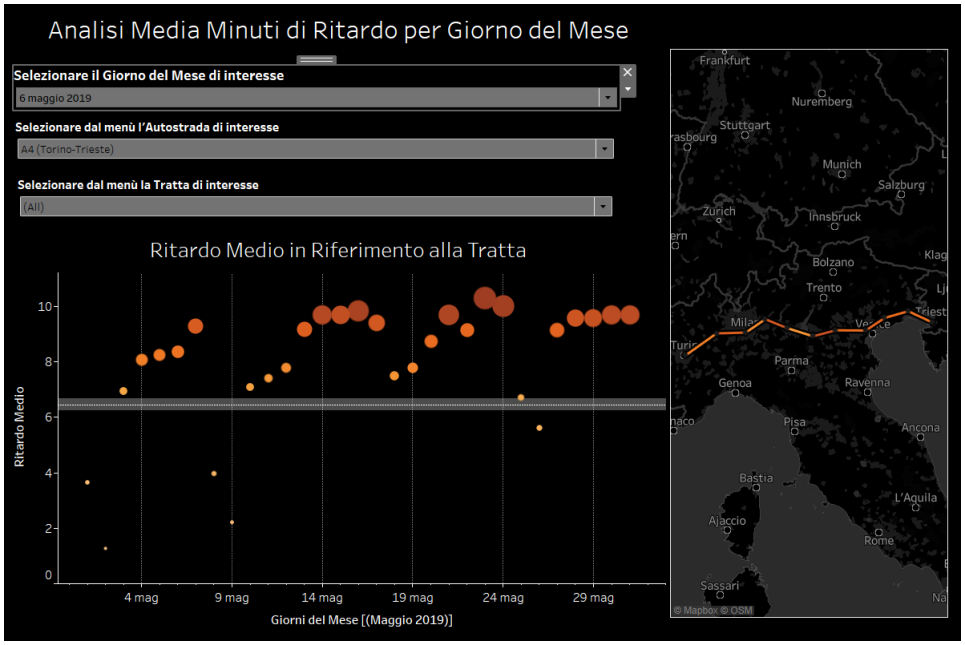
- Strettamente legata alla problematica precedente è la seguente: non tutti i cerchi del bubble chart presentano la label con il nome dell'autostrada corrispondente, quindi l'utente è costretto a cliccare o a passare col cursore per conoscere quest'informazione. Per cercare di ovviare al problema è stato dedicato più spazio al grafico nella dashboard permettendo la visione delle label associate alle autostrade più interessanti.

- Avendo eliminato il menù per l'autostrada, si è deciso di non inserire neanche quello relativo alle diverse tratte, poichè si sarebbe riscontrato un problema analogo a quello descritto in precedenza. Si è quindi deciso di permettere all'utente di selezionare la tratta mediante la mappa. Tuttavia, questo tipo di selezione risulta difficoltosa se prima non si è selezionata un'autostrada e quindi ingrandito la mappa.

- La mappa non sempre risulta di semplice consultazione per gli utenti. In molti casi l'ingrandimento involotario della mappa e la difficoltà nel ritornare alla visualizzazione iniziale hanno causato problemi durante la consultazione delle infografiche e nello svolgimento del task.

2.3 Terza Visualizzazione

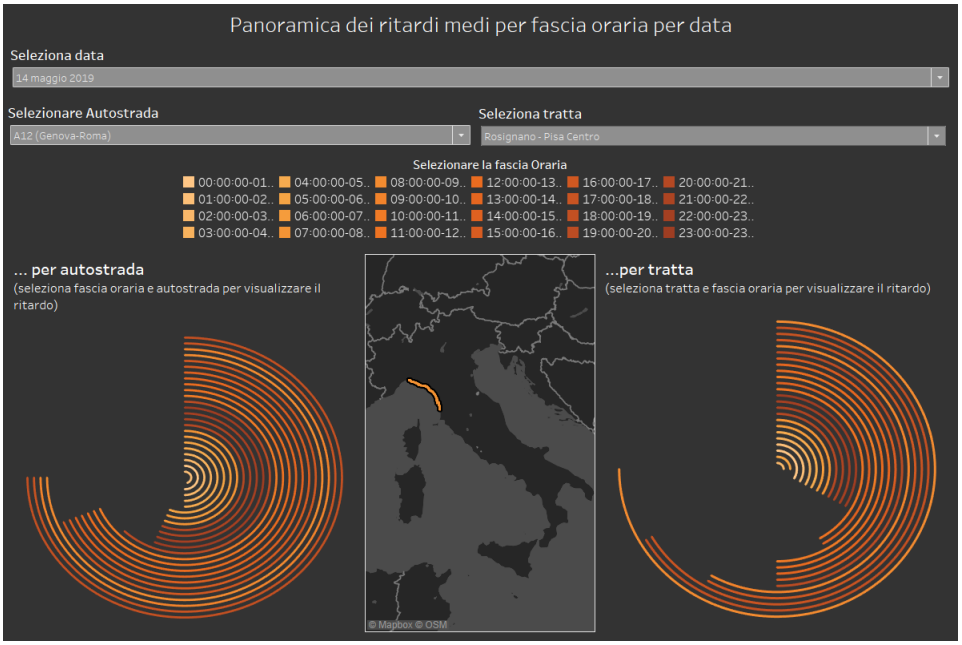
Tra tutte le visualizzazioni presentate questa risulta essere la più semplice ed intuitiva, confermato anche dai questionari e dai tempi di esecuzione dei task.
Lo scopo della visualizzazione è quello di fornire una visione complessiva giorno per giorno sulle singole tratte e di visualizzare l'oggetto dell'analisi graficamente nella mappa. Nonostante una mappa sia stata già presentata nella precedente visualizzazione è stato constatato che gli utenti preferiscono sempre aver un riferimento spaziale non avendo un'eccessiva familiarità con la localizzazione delle autostrade.
Inizialmente la dashboard presentava anche un linechart relativo all'andamento dei ritardi per l'autostrada. Si è deciso di eliminarlo poichè gli utenti hanno ritenuto l'informazione ridondante in quanto nella maggior parte dei casi l'andamento delle singole tratte e quello dell'intera autostrada risultavano essere molto simili. Al posto del grafico è stata inserita la media dell'autostrada nel grafico relativo alle tratte e il relativo intervallo di confidenza del 95%. In questo modo l'utente può osservare se il ritardo della tratta è in relazione a quello medio dell'autostrada.
In conclusione, gli utenti hanno suggerito di applicare il filtro sul giorno del mese anche sul grafico e non solo sulla mappa, così da permettere una miglior comprensione.



La principale problematica riscontrata è la sovrapposizione dei pallini rappresentanti i ritardi maggiori. Si è preferito associare anche la dimensione al ritardo oltre che al colore,così da rendere la visualizzazione più attrattiva e veloce da comprendere. Si è preferito lasciare una leggera sovrapposizione perchè, diminuendo ulteriormente la dimensione, i ritardi minori non sarebbero stati apprezzabili.

2.4 Quarta Visualizzazione

Tra tutte le visualizzazioni queste è risultata essere la meno intruitiva sia per i grafici in sè che per i dati a cui si riferisce. Si concentra sull'analisi dei ritardi medi per giorno del mese e per fascia oraria sia per l'autostrada che per le singole tratte.
La mappa, che non consente di essere zoomata, serve anche in questo caso per fornire all'utente una indicazione geografica tuttavia, al contrario delle precedenti visualizzazione essa utilzza le coordinate ottenute dall'interrogazione del sito *openstreetmap.org*.
Al contrario delle precedenti infografiche, dove il colore era associato al ritardo, qui è utilizzato per indicare la fascia oraria. Al fine di evitare fraintendimenti è stata inserita la legenda corrispondente anche utilizzabile come *highlighter*.
Non essendo il ritardo direttamente leggibile sul grafico, è stato inserito nelle label così da permettere all'utente di leggerlo facilmente una volta selezionata la fascia oraria di interesse.



2.5 Questionari Psicometrici e Valutazione degli utenti

Per comprendere come gli utenti valutassero le infografiche sono state condotte delle indagini mediante la somministrazione di un questionario psicometrico, come riportato in Figura 7. Si è scelto

	1	2	3	4	5	6
Utile	○	○	○	○	○	○
Intuitiva	○	○	○	○	○	○
Chiara	○	○	○	○	○	○
Informativa	○	○	○	○	○	○
Bella	○	○	○	○	○	○

Figura 7: Esempio di questionari psicometrici

questo questionario poichè permette di avere una visione complessiva riguardo l'opinione degli utenti, indagando le caratteristiche fondamentali che una visualizzazione dovrebbe avere. Inoltre, dando la possibilità all'utente di selezionare, per ogni caratteristica, un valore da 1 a 6, dove 1 è il minimo e 6 il massimo, gli si impedisce di evitare di esprimere una propensione positiva o negativa.
Di seguito vengono riportati,in figura 8, gli *stacked bar* relativi alle varie visualizzazioni. Come si legge anche nelle legende il blu indica i voti compresi tra 1 e 3 mentre, il rosso quelli da 4 a 6.

Dopo aver somministrato i questionari di valutazione, sono stati presentati dei task, uno per ciascuna visualizzazione, a 12 utenti, per valutare l'effettiva qualità dell'infografica.
Nello specifico i task presentati sono stati i seguenti

A Per la visualizzazione 1: Indicare l'autostrada e la tratta della stessa con il ritardo maggiore il lunedì.

B Per la visualizzazione 2: Indicare, rispetto alla fascia oraria 15.00:16.00, l'autostrada e la tratta corrispondente aventi il ritardo massimo.

C Per la visualizzazione 3: Indicare il giorno corrispondente al ritardo minimo per la tratta "Biandrate-Milano" dell'autostrada A4

D Per la visualizzazione 4: Indicare, in relazione al giorno 30 Maggio 2019 quale tratta dell'autostrada A10(Genova-Ventimiglia) presenta un ritardo di 8,17 minuti.

I risultati sono stati successivamente rappresentati attraverso dei violin plot.
Essi presentano una distribuzione normale e non evidenziano particolari problematiche nella consultazione della visualizzazione.

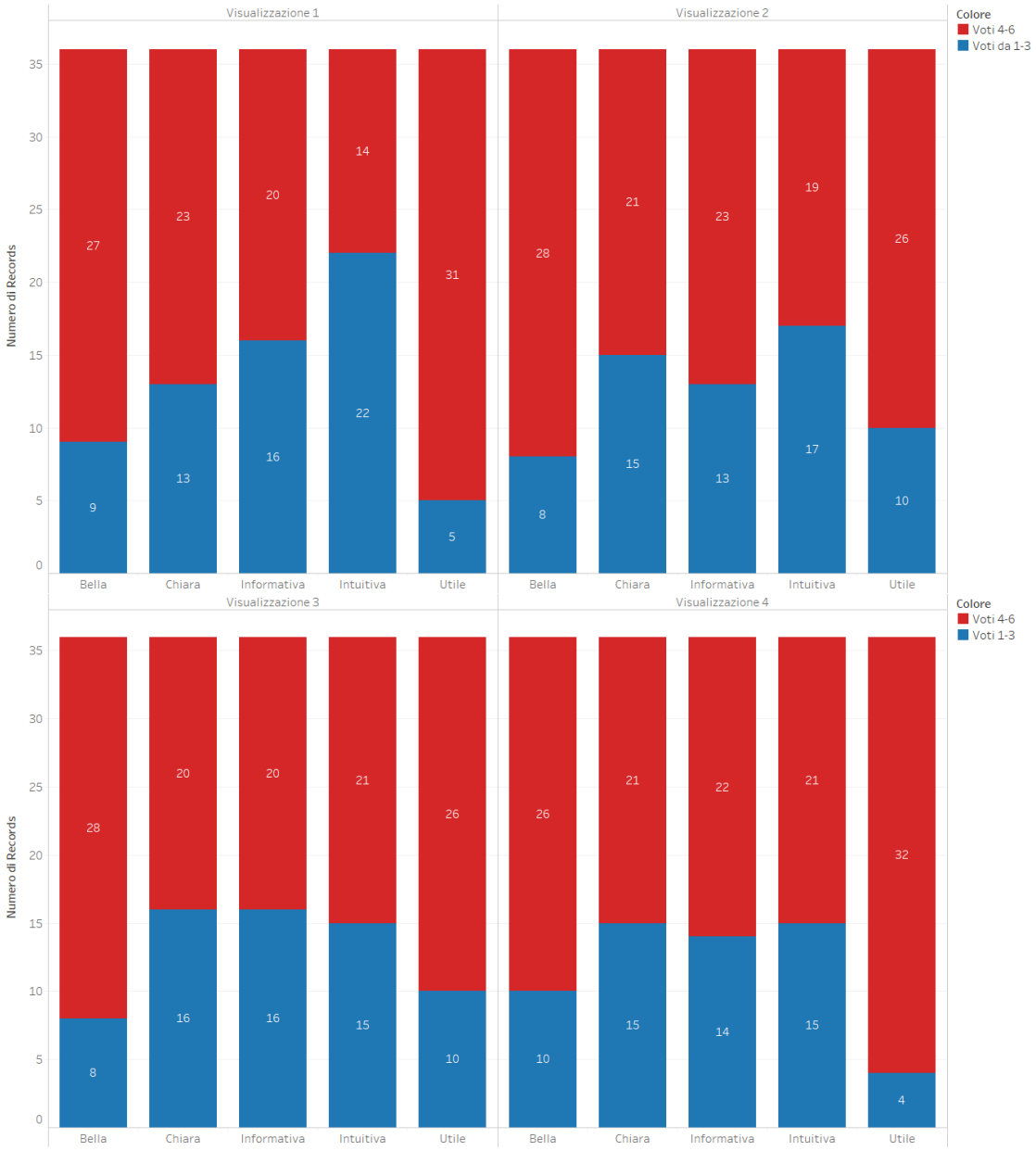


Figura 8: Grafici Questionari di valutazione

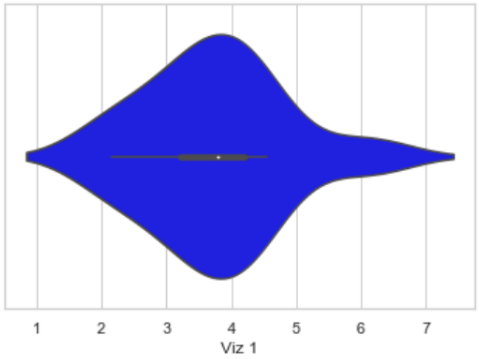


Figura 9: Visualizzazione 1

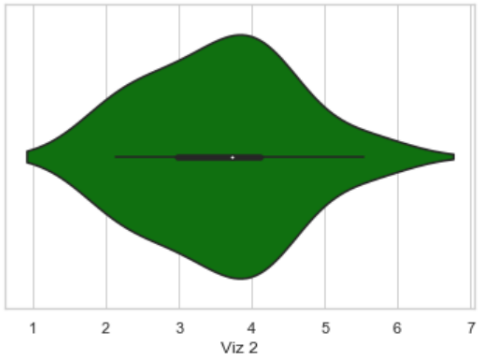


Figura 10: Visualizzazione 2

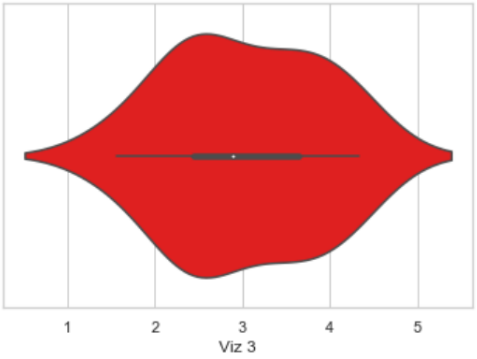


Figura 11: Visualizzazione 3

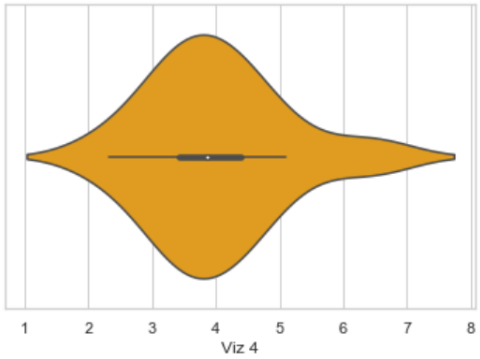


Figura 12: Visualizzazione 4