PROGETTO SABD

Relazione di lunghezza 3-6 pagine

Lo scopo del progetto è rispondere a 3 query su un dataset che riguarda le vaccinazioni italiane per Covid-19, usando Apache Spark per il processamento dei dati.

I file csv al link <https://github.com/italia/covid19-opendata-vaccini/tree/master/dati> vengono aggiornati quotidianamente, e vengono caricati e manipolati nel codice tramite le opportune query.

* Per ogni query vedere il codice e scrivere tutto (poi toglierò le cose inutili)
* Con l’occasione segnare i commenti al codice che poi farò a casa
* Scrivere in modo che la relazione aiuti a ripetere
* Motivare ogni cosa

Il progetto è stato sviluppato da un gruppo di due persone, in linguaggio Java e in ambiente Linux.

Per l’organizzazione del progetto è stata inizializzata una repository su GitHub, che contiene il codice sorgente e gli script bash che fanno partire l’applicazione. **(+ hdfs, nifi, docker-compose.yml).**

È stato scelto di procedere alla realizzazione del progetto con Apache Spark piuttosto che Hadoop, per una maggiore flessibilità e per un insieme più ampio di primitive che semplificano la programmazione.

La prima query richiede di calcolare, per ciascun mese e ciascuna regione, il numero medio di vaccinazioni giornaliere in un generico centro vaccinale.

Partendo dai due file csv in input, vengono creati due dataset che considerano solo le colonne utili allo scopo della query.

I dataset sono poi trasformati in un JavaRDD, così da poter effettuare trasformazioni e operazioni di calcolo sulle collezioni di dati in maniera comoda e veloce.

Successivamente, poiché richiesto di non considerare i dati precedenti al giorno 01-01-2021, i dati ordinati temporalmente vengono filtrati per eliminare le istanze risalenti al 2020.

Con lo scopo di ottenere un JavaPairRDD che contenga la chiave *“area, mese”* e i valori delle somministrazioni, vengono applicate le operazioni mapToPair e reduceByKey all’RDD di partenza. In seguito, il numero di somministrazioni viene diviso per il numero di centri di una regione così da ottenere la media giornaliera di vaccinazioni in un generico centro vaccinale, cioè il valore richiesto dalla query.

Il risultato finale è un dataset con le colonne che indicano l’area, il mese e la media sopraccennata.

Per la seconda query si parte da un altro file csv, che fornisce informazioni sulle vaccinazioni specificando la data, il fornitore, l’area, la fascia anagrafica, il sesso delle persone vaccinate e altre informazioni che non verranno usate per lo svolgimento.

Ci si interessa infatti alle persone vaccinate di sesso femminile e, per ogni mese, viene chiesto di determinare la classifica delle cinque regioni per le quali è previsto il maggior numero di donne vaccinate nel primo giorno del mese successivo.

Per la predizione si importa la libreria di MLlib di Spark e, al fine di raggiungere il risultato, si usa la retta di regressione che approssima la tendenza di vaccinazioni giornaliere, che riceve in input solamente i dati di un mese per predire il mese seguente.

Quindi, partendo dal file csv, si crea anche questa volta un dataset che viene direttamente trasformato in JavaRDD dal quale viene costruito un JavaPairRDD con la chiave *“data, area, fascia”* e il valore del totale dei vaccinati. Anche per questa query vengono scartati i dati del 2020, ma in un modo diverso rispetto al caso precedente, cioè riferendosi al giorno 01-01-2021 con un formato *Date* invece che *String*.

Dai dati filtrati viene realizzato un dataset di training che chiama la regressione.

A seguire viene realizzato un JavaPairRDD la cui chiave contiene il mese successivo (mese della predizione) e il valore di somministrazioni predetto. L’RDD è fatto in modo che i dati siano ordinati per mese, fascia anagrafica e regione, in maniera decrescente, grazie a un Comparator.

Infine, come richiesto, si crea la classifica delle cinque regioni con il valore predetto più grande; questo avviene mediante una lista che prende solo i primi cinque elementi dell’insieme ordinato.

Il risultato finale è un dataset con le colonne che indicano la data (primo giorno del mese da predire), la fascia anagrafica, l’area e le somministrazioni previste.

Nella terza e ultima query si chiede di stimare il numero di vaccinazioni al primo giugno. Viene inoltre richiesto di applicare algoritmi di clustering sulle regioni, dividendole in base alla popolazione vaccinata, normalizzata rispetto alla popolazione totale.

Per il clustering vengono considerati due algoritmi: K-Means e Bisecting K-Means, entrambi implementati in MLlib, dei quali vengono poi confrontate le performance.

Nello svolgimento si analizzano due file csv: uno che specifica il totale della popolazione di ogni regione e un altro che è lo stesso usato nella prima query, da cui si estrapolano la data, l’area e il totale dei vaccinati in una regione in uno specifico giorno.

A differenza delle due query precedenti, vengono presi in considerazione anche i dati relativi al dicembre 2020, fino al 31 maggio 2021.

Inizialmente si crea un JavaPairRDD con area, data e vaccinazioni, per poi passare a una regressione lineare che stima le vaccinazioni previste per il primo giugno, per ogni regione. Dopodiché si calcola la percentuale di popolazione vaccinata, attraverso una reduceByKey che conta le vaccinazioni totali in una regione, per poi dividere questo valore per la popolazione totale presa dall’altro file csv.

--- Vector? ---

Nella fase di clustering verranno usati i due algoritmi che sono stati descritti in due classi distinte che estendono una classe astratta, la quale dichiara le caratteristiche comuni, che nel nostro caso sono i metodi necessari al completamento della query. I metodi non fanno altro che eseguire il train del modello, la predizione del cluster e il calcolo del training cost e del WSSSE.

Il clustering è eseguito, come da richiesta, al variare del valore di K da 2 a 5.

Il risultato prodotto è un dataset che indica l’algoritmo usato con lo specifico valore di K, il costo e il WSSSE legato a quell’esecuzione dell’algoritmo, l’area, la percentuale di vaccinati e il cluster di predizione a cui viene assegnata ogni regione.