Relazione Big Data

Corso di HPC

Università degli Studi di Salerno

Anno Accademico 2024/2025

Docente: Giuseppe D'aniello
gi.daniello@unisa.it

Studente: Giulia Minichiello
Matricola: 0622702127

Chapter 1

Descrizione del Dataset

Il dataset utilizzato per l'analisi di auto BMW è un file CSV denominato `BMW_Car_Sales.csv`. Include i seguenti campi:

- **Model**: Modello dell'auto BMW.
- **Year**: Anno di produzione dell'auto.
- **Region**: Regione geografica in cui è avvenuta la vendita.
- **Color**: Colore dell'auto.
- **Fuel_Type**: Tipo di carburante utilizzato (Petrol, Diesel, Electric, Hybrid).
- **Transmission**: Tipo di trasmissione (Manual o Automatic).
- **Engine_Size_L**: Cilindrata del motore.
- **Mileage_KM**: Chilometraggio dell'auto.
- **Price_USD**: Prezzo dell'auto in dollari.
- **Sales_Volume**: Volume delle vendite per il modello specifico.
- **Sales_Classification**: Classificazione delle vendite (High, Low).

Chapter 2

Introduzione ad hadoop MapReduce

Hadoop MapReduce è un framework open-source progettato per l'elaborazione distribuita di grandi quantità di dati. Il modello MapReduce si basa su due fasi principali: la fase *Map*, in cui i dati vengono suddivisi e processati in parallelo, e la fase *Reduce*, in cui i risultati intermedi vengono aggregati per produrre l'output finale.

2.1 Esercizio 1- MapReduce

Realizzare un programma con MapReduce che identifichi i topK modelli con il maggior numero di vendite (campo `Sales_Volume`). Il programma dovrà aggregare i volumi di vendita per ciascun modello e restituire in output i K modelli che hanno totalizzato il maggior numero di vendite complessive.

2.1.1 Soluzione

Il problema è stato risolto utilizzando due Job.

Pattern Utilizzati

Per risolvere questo problema utilizzando Hadoop MapReduce sono stati utilizzati i seguenti Pattern:

- **Job Chaining**: Utilizzato in modo tale che l'output di Job 1 diventi l'input di Job 2.
- **Numerical Summarization**: Applicato per aggregare i volumi di vendita per ciascun modello.
  - **Mapper**: Il mapper emette coppie (modello, sales_volume);
  - **Combiner**: Il primo reducer viene utilizzato come Combiner nel Job 1 , perché la somma è associativa e commutativa. Quindi somma localmente i valori con la stessa chiave.
  - **Reducer**: Il reducer usa la chiave come "group by" e somma le vendite per modello restituendo (modello,totale_vendite).

  Il Secondo Job è stato implementato con un filtering pattern per implementare topK
- **TopK**: Implementato per selezionare i K modelli con il maggior numero di vendite.
  - **Mapper**: ogni mapper crea la propria lista di topK modelli
  - **Reducer**: Il reducer combina le liste ricevute dai mapper per ottenere la lista finale.

Il numero di reducer scelto è 2 per il Job 1, per suddividere il carico tra più nodi, e 1 per il Job 2 per ottenere TopK globale.

2.1.2 Risultati

Di seguito sono riportati i risultati ottenuti dai due job.

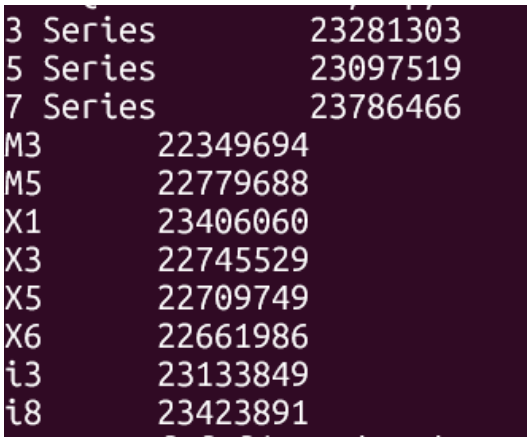

Figure 2.1: Figura1- output Job 1

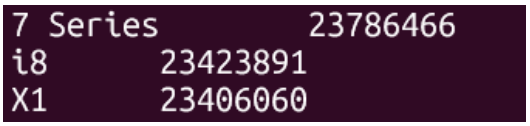

Figure 2.2: Figura2- output Job 2

Chapter 3

Apache Spark

Apache Spark è un framework open-source per l'elaborazione distribuita di dati su larga scala. Rispetto a Hadoop MapReduce, Spark offre prestazioni superiori grazie all'elaborazione in memoria.

3.1 Esercizio 2- Spark

La statistica che si è scelto di ricavare è un indice che stima quanto una regione sia "green" in relazione al numero di auto elettriche vendute dal 2015. Per la realizzazione dell'esercizio si è scelto di eliminare la riga di intestazione dal csv.

3.1.1 Soluzione

Per calcolare l'indice "green" per ciascuna regione, si è proceduto come segue:

- Utilizzo del primo filtro per selezionare solo le auto con anno dal 2015.
- Poi è stato applicato un secondo filtro per considerare solo le auto elettriche.
- Ogni riga è stata trasformata nella coppia (regione,vendite) con la trasformazione "mapToPair".
- Le vendite per regione sono state sommate utilizzando la trasformazione "reduceByKey".
- Le coppie sono state invertite per poter utilizzare la sort,
- È stata utilizzata la trasformazione "sortByKey" per ordinare le regioni in base alle vendite in ordine decrescente.

3.1.2 Risultati

Di seguito è riportato il risultato ottenuto dall'esecuzione del programma Spark.

Figure 3.1: Figura3- output Spark