Spark: Repartition vs Coalesce

repartition()

* Changes the number of partitions by shuffling the data across the cluster.
* Can increase or decrease the number of partitions.
* Ensures evenly distributed data across all partitions.
* Always triggers a full shuffle (expensive operation).

When to use

* When you want to increase the number of partitions (e.g. for parallelism).
* When you need to rebalance data across partitions (e.g. before a join or groupBy).
* When data is skewed and you want uniform partition sizes.

coalesce()

* Changes the number of partitions by merging existing partitions.
* Typically used to reduce the number of partitions.
* Avoids shuffling (unless shuffle=True is specified).
* Faster and more efficient for reducing partitions.

When to use

* When you want to reduce partitions efficiently (e.g. from many small partitions to fewer large ones).
* Before writing to disk (reduce small files).
* When you don’t need to rebalance data across partitions.