UNIVERSITA' DEGLI STUDI DI PERUGIA

Dipartimento di Ingegneria

Corso di laurea magistrale in Ingegneria Informatica e Robotica

curriculum Data-Science

Tesina di Big Data Management

Docente Fabrizio Montecchiani

Anno accademico 2022-2023

BENCHMARKS TRA ALGORITMI SPARK E RAY CON MODELLO XGBOOST

Tommaso Martinelli *350400*

Riccardo Rossi *346626*

1.Introduzione

Lo scopo del progetto è quello di fare predizione sul dataset di Kaggle, 'Rain in Australia' (https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package), utilizzando rispettivamente i due noti framework di calcolo, Ray e Spark, andandoli a confrontare per analizzarne le prestazioni ed i risultati migliori.

2. DataFlow e tecnologie utilizzate

2.1 Il Dataset

Il Dataset 'Rain in Australia', scaricato da Kaggle, è un dataset che contiene circa 10 anni di osservazioni meteorologiche giornaliere registrate da molte località dell'Australia. Le 23 features, infatti, raccontano le caratteristiche metereologiche di una tipica giornata australiana: percentuale di umidità, direzione del vento, velocità del vento, temperatura giornaliera in diversi istanti della giornata, ecc…

Lo scopo del dataset è quello di fare classificazione binaria sulla label 'Rain_Tomorrow', ovvero riuscire a prevedere se, nella giornata successiva a quella per la quale si sono raccolte le features, pioverà oppure no, dunque RainTomorrow=1 (pioverà), oppure RainTomorrow=0 (non pioverà).

2.2 Tecnologie utilizzate

Le tecnologie utilizzate nel progetto sono:

- Jupyter notebook
- PyCharm
- Python
- Spark
- MLlib
- Ray

2.3 Architettura e DataFlow

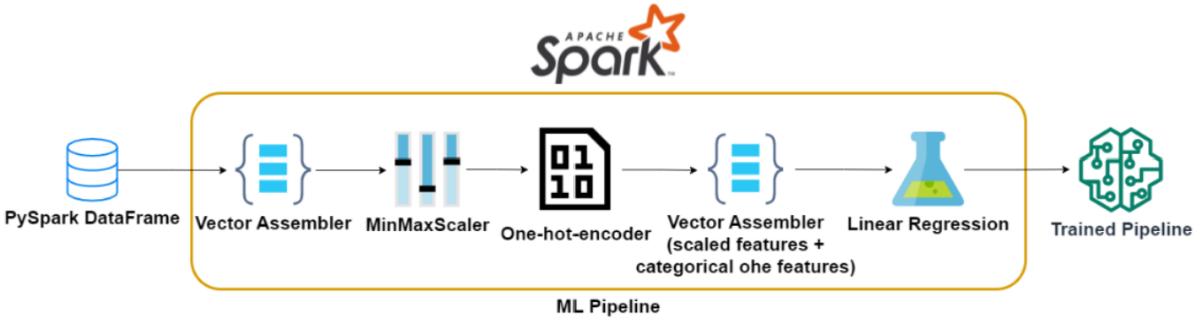
Una volta scaricato il dataset, si è deciso di fare EDA (Exploratory Data Analisys) separatamente dal resto del codice dedicato al calcolo distribuito, per una maggiore chiarezza di quest'ultimo. Una volta ottenuti i nuovi dati grazie all'EDA, è stato esportato il nuovo dataset ed è stato caricato quest'ultimo sia sullo script di Ray sia in quello di Python.

Ray è un framework di calcolo unificato open source che semplifica la scalabilità dei carichi di lavoro di intelligenza artificiale. Affronta queste sfide a testa alta consentendo agli ingegneri e agli sviluppatori di machine learning di scalare facilmente i propri carichi di lavoro dai laptop al cloud senza la necessità di creare complesse infrastrutture di calcolo. Ray include Ray AI Runtime (**AIR**), un set nativo di librerie ML scalabili best-in-class, come ad esempio:

- ML data pre-processing tasks via *Ray Data*
- Training large models via *Ray Train*
- Hyperparameter tuning via *Ray Tune*

Inoltre, Ray e le sue librerie si integrano perfettamente con il resto dell'ecosistema Python e ML. Con queste librerie, i non esperti possono facilmente sfruttare il calcolo distribuito utilizzando i loro strumenti ML e Python preferiti. Nel progetto sono state riportate tutte e tre le librerie sopra citate, a eccezione della prima, il cui utilizzo è stato ridotto a seguito della scelta di fare EDA separatamente tramite PyCharm.

Apache Spark è un framework di elaborazione dati in grado di eseguire rapidamente attività di elaborazione su set di dati molto grandi e può anche distribuire attività di elaborazione dati su più computer, da solo o in tandem con altri strumenti di calcolo distribuito. La libreria di Spark dedita al machine learning che si è utilizzata nel progetto è MLlib. Quest'ultima standardizza le API per gli algoritmi di machine learning per semplificare la combinazione di più algoritmi in un'unica pipeline o flusso di lavoro:



Nel progetto sono stati riportati solamente i seguenti elementi della pipeline:

- Assembler
- Scaler
- Model

Questo perché gli altri strumenti servivano per la gestione delle variabili categoriche, le quali sono già state gestite diversamente come già ribadito più volte nel documento.

XGBoost, che sta per Extreme Gradient Boosting, è una libreria di machine learning GBDT (Gradient Boosting Decision Tree) scalabile e distribuita. Fornisce ‘parallel tree boosting’ ed è la principale libreria di machine learning per problemi di regressione e classificazione. XGBoost viene normalmente utilizzato per addestrare alberi decisionali basati su gradient boosting. È possibile utilizzare XGBoost per addestrare una foresta casuale (Random Forest) autonoma.

3. Risultati

4. Casi d’uso

5. Limiti e possibili estensioni