

Capitolo 1

Orchestrazione di Flussi di lavoro



Figura 1.1: MLops

Dopo aver visto che tra i vari casi di studio troviamo una linea comune nella pipeline di utilizzo del Machine Learning, proviamo automatizzare le invarianti trovate nel capitolo precedente.

Negli ultimi anni i ricercatori stanno implementando diversi framework automatici che ci danno la possibilità di automatizzare il flusso di lavoro per la costruzione del modello. Come visto in precedenza intendiamo: raccolta dati, creazione e implementazione del modello e successivamente alla distribuzione permettono la riproduzione e il monitoraggio.

Le pipeline ML, inoltre, aiutano a migliorare le prestazioni e la gestione dell'intero mo-

dello. Possiamo semplicemente semplificare lo schema visto in precedenza come segue, lasciando la gestione del modello assegnata al framework.

Gli strumenti per l'orchestrazione ML sono usati per automatizzare e gestire i workflow e le infrastrutture delle pipeline con semplici interfacce, aiutando i data scientists e al team di ML di concentrarsi solo sul necessario.

Orchestrare un flusso di lavoro è fondamentale per un'azienda che investe nel ML, perciò è necessario capire come automatizzare il maggior numero di risorse, tra cui l'estrazione dell'output del modello con dei monitoraggi costanti durante la fase di produzione.

Questi campi che stiamo analizzando sono un sottoinsieme di MLOps: un insieme di pratiche per la collaborazione e la comunicazione tra data scientist e professionisti delle operazioni. Applicando queste tecniche pratiche aumenta la qualità finale, semplifica il processo di gestione e automatizza l'implementazione di modelli di ML e DL in ambienti di produzione su larga scala(schema sottostante)

1.1 Framework

Prima di elencare una lista di tools di gestori di risorse di ML, diventa necessario anticipare una distinzione tra un software in fase di ricerca e un software già a livello di produzione ovvero già utilizzato a livello aziendale.

Un software ancora in fase di ricerca, spesso utilizzato a livello accademico, offre delle funzionalità magari non ancora rilasciate nel mondo "Aziendale" avendo delle "particolarità" che altri software magari *"in produzione"* non possiedono.

Questi ultimi, a loro volta, si differenziano grazie a caratteristiche quali l'affidabilità.

1.1.1 ZenML

ZenML è uno strumento open-source per operazione di ML. Lo strumento si focalizza sui problemi di riproduzioni basati sulla produzione, come le difficoltà di versioning e modelli, organizzazione di workflow di ML e la distribuzione. Può lavorare a fianco a un altro strumento di orchestrazione dei flussi di lavoro per fornire un semplice percorso per rilasciare il modello ML in produzione.

È possibile verificare con precisione dati, modelli e configurazioni. Ti consente di valutare in maniera semi-automatica il modello, confrontare le pipeline di addestramento e distribuire la preelaborazione nel cloud.

Come vediamo dalla figura 4.2 ogni metodo implementa uno specifico passaggio del workflow, vincolando i passaggi da eseguire sul modello a discrezione del Framework stesso; se l'addestramento (*add_trainer*) appare prima del processamento dei dati (*add_preprocessing*), sarà compito del framework nel metodo (*run*) organizzare l'ordine delle operazioni da eseguire nel modello.



```
python
>> import zenml
>> p = zenml.TrainingPipeline(name='My Awesome Pipeline')
>> p.add_split(RandomSplit())
>> p.add_preprocessing(StandardPreprocessor(features='all'))
>> p.add_trainer(FeedForwardTrainer())
>> (...)
>> p.run(orchestration='gcp', processing='dataflow')
```

Figura 1.2: ZenML

1.1.2 Kedro

Kedro é uno strumento open-source di orchestrazione basato su Python, che permette di creare workflow per ML riproducibili, mantenibili e modulari, semplificando i processi e rendendoli più accurati.

Kedro integra l'ingegneria del software in un ambiente di apprendimento automatico, con concetti quali: modularità, divisione degli interessi e versioning.

Astraendo la pipeline, è possibile automatizzare le dipendenze tra il codice Python e la visualizzazione del Workflow. L'obiettivo principale è la creazione di codice di data science gestibile per affrontare le carenze di Jupiter (applicazione web open-source, utile per condividere documenti che contengono codice live, equazioni e grafi ecc.)

Questo strumento crea un lavoro di squadra più semplice a vari livelli, e fornisce un efficiente ambiente di coding con codice modulare e riusabile.

1.1.3 Flyte

Flyte è uno strumento open-source di alta fascia che permette di facilitare la creazione di flussi di lavoro del ML. È una piattaforma di programmazione ad elaborazione distribuita e strutturata, con flussi di lavoro concorrenti, scalabili e mantenibili per l'apprendimento automatico e l'elaborazione dei dati.

Flyte gestisce già più di dieci mila workflow, è basato su Kubernetes e offre portabilità, scalabilità e affidabilità.

L'interfaccia è elastica, intuitiva e facile da usare; offre inoltre parametri, linee di dati e caching per organizzare i workflow.

L'intera piattaforma è dinamica ed estensibile attraverso vari plug-in per assistere la creazione e l'implementazione dei vari workflow. Tali Workflow, a loro volta, possono

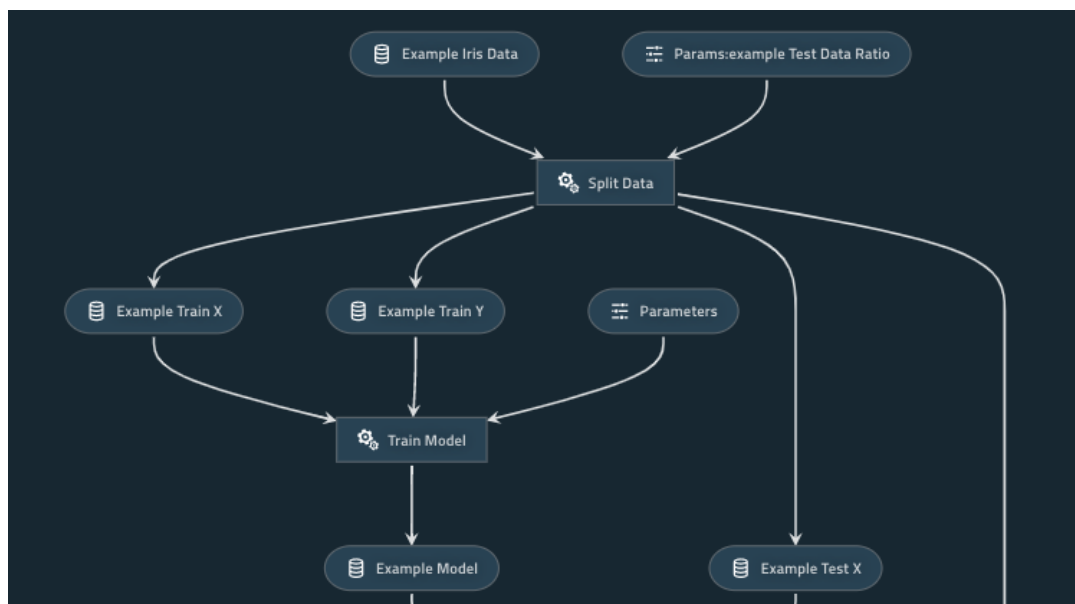


Figura 1.3: Kedro

essere reiterati, annullati, sperimentati e condivisi per accelerare il processo di sviluppo dell'intero team.

Poiché ogni entità in Flyte è immutabile, insieme a ogni modifica esplicita assunta come nuova versione, diventa semplice ed efficiente iterare, sperimentare e tornare indietro nei Workflow.

1.1.4 MLRun

MLRun è uno strumento open-source di orchestrazione di workflow/pipeline. Ha integrato un approccio per organizzare pipeline di ML, a partire dallo sviluppo iniziale, costruzione del modello verso tutti processi che portano allo sviluppo in produzione. Esso integra un vasto reparto di strumenti e plugin per lavorare con caratteristiche e modelli insieme alla distribuzione del flusso di lavoro.

MLRun dispone di un archivio di funzionalità e artefatti per controllare l'inserimento, l'elaborazione e l'archiviazione dei dati su più repository e con tecnologie diverse.

È un elastico servizio “Server-Less” per convertire semplice codice in scalabili e organizzati “Microservizi”; diventa così più semplice e leggero fare esperimenti, allenare i modelli e testarli, e implementare dei flussi di lavoro della pipeline in tempo reale.

L'interfaccia utente complessiva ha una struttura centralizzata per gestire flussi di lavoro di ML. Le caratteristiche principali includono una rapida implementazione, elastica scalabilità, gestione delle funzionalità e un'usabilità flessibile.

Executions			
<div> <div>Status ▾</div> <div>Version ▾</div> <div>Start Time ▾</div> <div>Duration ▾</div> </div>			
EXECUTION ID	STATUS	START TIME	DURATION
dklpfcdw6i Last run 13 hours ago	SUCCEEDED	10/12/2020 5:15:32 AM UTC 10/11/2020 10:15:32 PM PDT	3m 47s View Inputs & Outputs
gn2ay2vmqw Last run 13 hours ago	SUCCEEDED	10/12/2020 5:00:32 AM UTC 10/11/2020 10:00:32 PM PDT	3m 50s View Inputs & Outputs
pum4486wd8 Last run 13 hours ago	SUCCEEDED	10/12/2020 4:45:32 AM UTC 10/11/2020 9:45:32 PM PDT	4m 16s View Inputs & Outputs
th0eykgtp3 Last run 13 hours ago	SUCCEEDED	10/12/2020 4:30:32 AM UTC 10/11/2020 9:30:32 PM PDT	14m 51s View Inputs & Outputs
lsl4ks493v Last run 14 hours ago	SUCCEEDED	10/12/2020 4:15:32 AM UTC 10/11/2020 9:15:32 PM PDT	4m 7s View Inputs & Outputs
t7d9pyv9p3 Last run 14 hours ago	SUCCEEDED	10/12/2020 4:00:32 AM UTC 10/11/2020 9:00:32 PM PDT	3m 30s View Inputs & Outputs
g4h7xini7a Last run 14 hours ago	SUCCEEDED	10/12/2020 3:45:32 AM UTC 10/11/2020 8:45:32 PM PDT	3m 30s View Inputs & Outputs

Figura 1.4: Flyte Console

1.1.5 Couler

Couler ha un'interfaccia unificata per il codice e la gestione dei flussi di lavoro con differenti motori e framework di workflow.

L'unico strumento di orchestrazione del flusso di lavoro per la gestione di altri strumenti di orchestrazione.

Anch'esso è open-source e a differenza di altri motori, che hanno complessi livelli di astrazione, l'interfaccia di Couler rende estremamente semplice la gestione di questi livelli.

Couler ha uno stile di programmazione imperativo per la definizione del workflow integrato però con un supporto per la costruzione automatica di un Grafo Aciclico.

I Servizi di Couler sono altamente estendibili, supportando altri motori di flussi di lavoro. Facilita l'addestramento dei modelli di ML, aumentando la modularità e la riusabilità.

1.2 Conclusioni