LINUX / Bash

pour la data

Les essentiels

PROGRAMME DATA UPSKILLING

Savoir manier linux et le terminal ne fera pas de vous un pro’ de la data.

Mais ne pas savoir s’en servir peut vous mettre en difficulté...

Installation de Linux

WSL vs. Dual Boot vs. VM

PROGRAMME DATA UPSKILLING

Manipulation fichiers/dossiers:

> créer, lister, copier, déplacer, supprimer

> naviguer dans l’arborescence Linux (/bin /root /dev etc...)

> autocompletions

> zipper un fichier

Modifier des fichiers avec vim

> ouvrir un fichier avec vim

> utiliser les raccourcis

> entrer/quitter le mode insertion

> quitter VIM avec et sans sauvegarde

Exécution de programmes

> rendre un script bash ou python exécutable

> créer et récupérer une variable d’environnement

> la variable PATH: utilité et modification

Les startup files

.bashrc

.bash_profile

.bash_login

+ utilité & ordre de précédence

+ différence entre sh, bash et zsh

Les Standard Streams:

> STDIN

> STDOUT

> STDERR

> redirection des streams (avec le mythique 2>&1)

Utilitaires

> man et tldr

> apt

> lister et killer les process en cours avec htop

> shutdown ;)

SQL pour la data

Manipuler la donnée dans tous les sens

PROGRAMME DATA UPSKILLING

“SQL est un indispensable dans la Data.

Voici ce que j’ai appris à faire en SQL pour être opérationnel en mission”

(…vous connaissiez “Qualify” ?)

PROGRAMME DATA UPSKILLING

Les Joins

Indispensables mais souvent mal maîtrisés

> Cross Joins

> Inner Joins

> Left Joins
(un Right ne devrait jamais partir en prod...)

> Full Outer Joins

> Self Joins
(pas vraiment un type de join mais très utile)

Les CTE

> ou “common table expressions”

> pour éviter les subqueries
(ou pire encore, les *nested subqueries*)

> indispensable pour créer des requêtes faciles à maintenir

Les agrégations simples

> sommes, moyennes, counts...

> avec et sans groupby

> le keyword “HAVING”

> pro level: le CASE WHEN dans une fonction d’agrégation

Les agrégations avancées

> GROUPING SETS

> FILTER

> le keyword “HAVING”

> ROLLUP et CUBE

Les window functions

> OVER()

> ROWS BETWEEN

> PARTITION BY

> LAG et LEAD

> ROW_NUMBER vs. RANK vs. DENSE_RANK

> QUALIFY !

GIT
pour la data

Utiliser la pleine puissance du versioning

PROGRAMME DATA UPSKILLING

“GIT fait peur à tout le monde”

Pourtant, bien maîtrisé il fait gagner beaucoup de temps.

Voici ce que j’ai appris sur GIT pour faciliter la collaboration en entreprise

PROGRAMME DATA UPSKILLING

Les bases

> Initialiser et cloner un repo

> La zone de staging: git add, rm et .gitignore

> Les commits

> Push / Pull

> git reset

Github

> créer un compte et un premier repo sur Github

> le fichier Readme (+ checkpoints d'une bonne doc)

> Mise en place d'une connexion SSH

> Collaborer sur Github: travailler à plusieurs sur le même fichier

Les branches

> master / develop

> créer/supprimer une branche

> Pull requests sur Github (sans conflit)

> résoudre les conflits (sur Github et en local)

> merge vs. rebase

Avancé

> Le dossier .git, ce qu'il contient et son fonctionnement

> Le hash commit

> git log (et options/flags)

> les alias

Python pour la data

Cleaner, traiter et exposer les données

PROGRAMME DATA UPSKILLING

“Python est le langage #1 de la data”

Voici ce que je peux faire avec Python pour créer des projets qui ont un vrai impact pour votre entreprise

(exit les POCs sans suite...)

PROGRAMME DATA UPSKILLING

Environnements virtuels

> Utilisation de venv pour isoler plusieurs projets

> Associer Pycharm au venv du projet

> gestion des dépendances

Manipulations de données

> Manipulations classiques: jointures, agrégations...

> Utilisation et traitement du format JSON

> Pandas vs. DuckDB

> Data cleaning

Python intermédiaire

> classes / POO

> modules

> exception-handling (try/except/finally)

> logging

> datetime

Exposition de la data

> créer une app avec Flask

> créer une app avec Streamlit

> créer et requêter une API avec FASTApi

> communiquer avec SQL: DuckDB, SQLite3

Utilitaires & code quality

> Pycharm et ses meilleurs raccourcis

> Black

> mypy

> pylint

Apache SPARK

Utiliser la puissance du calcul distribué pour le Big Data

PROGRAMME DATA UPSKILLING

Spark est le framework de référence pour le calcul distribué de grosses volumétries de données.

Jusqu’à 100x plus rapide que son “ancêtre” MapReduce

Voici ce que j’ai appris sur Spark...

L’éco-système Hadoop

> Les limites de MapReduce et pourquoi Spark a vu le jour

> l’horizontal scaling

> HDFS et Spark

Les clusters

> Master / worker Nodes

> La communication entre les nodes

> L’UI de monitoring du cluster

Spark APIs

> RDD vs. DataFrame vs. Dataset

> Les API high-level SQL et Pandas

> SparkContext et SparkSession

Spark Internals

> Lazy Execution

> Transformations vs. Actions

> “Stages” et “Tasks”

> Narrow vs. Wide dependancies et Shuffle