Liste des projets Big data et Cloud Computing

par Serge NDOUMIN

2024 - 2025

N°	Description du projet	Objectif recherché
1.	Analyse de données de e-Commerce avec PySpark	- Identifier 10 indicateurs pertinents à analyser - Mettre en application le traitement en parallèle - Utiliser le framework pyspark - Maitriser des transformations et des actions
2.		
3.	Analyse de données du secteur santé avec PySpark	
4.		
5.	Analyse de données log d'un serveur avec PySpark	
6.		
7.	Analyse de données du secteur Finance avec pyspark	
8.		

Ressources : Lien vers mon drive contenant les données

Deadline : 28 février 2025 à envoyer à l'adresse mailto:ndoumins@gmail.com?subject=Projet Big Data

Livrables :

- Rapport détaillé de l'analyse contenant les graphes
- Notebook jupyter

Projet 1 : Analyse de données de e-Commerce avec PySpark

Description du jeu de données

- **Géographie :** Pakistan
- **Période de temps :** 03/2016 - 08/2018
- **Unité d'analyse :** Commandes de commerce électronique

L'ensemble de données contient des informations détaillées sur un demi-million de commandes de commerce électronique au Pakistan entre mars 2016 et août 2018. Il contient les détails de l'article, le mode d'expédition, le mode de paiement comme la carte de crédit, Easy-Paisa, Jazz-Cash, le paiement à la livraison, les catégories de produits comme la mode, le mobile, l'électronique, l'électroménager etc..., la date de la commande, l'UGS, le prix, la quantité, le total et l'identifiant du client. Il s'agit de l'ensemble de données le plus détaillé sur le commerce électronique au Pakistan que l'on puisse trouver dans le domaine public.

Variables : L'ensemble de données contient l'ID de l'article, le statut de la commande (terminée, annulée, remboursée), la date de la commande, l'UGS, le prix, la quantité, le total général, la catégorie, le mode de paiement et l'ID du client.

Exemples d'analyses :

- Le best-seller par catégorie
- Best-seller par an
- Nombre de commandes annulées par an
- Meilleure vente en termes d'iphone en 2018
- Identifier les articles les moins vendus ou invendus pour optimiser les stocks

- Compter le nombre de commandes pour chaque mode de paiement
- Compter le nombre de commandes pour chaque mode de paiement
- Calculer le chiffre d'affaires par jour, semaine ou mois
- Identifier les plages horaires où les commandes sont les plus nombreuses
- Etc.

Projet 2 : Analyse de données du secteur santé avec PySpark

Description des données :

FitLife360 est un ensemble de données synthétiques qui simule les données de suivi de la santé et de la condition physique de 3 000 participants sur une période d'un an. L'ensemble de données saisit les activités quotidiennes, les mesures vitales de la santé et les facteurs liés au mode de vie, ce qui le rend précieux pour l'analyse de la santé et la modélisation prédictive.

- **Description des caractéristiques Informations démographiques**

participant_id : Identifiant unique pour chaque participant
age : Âge du participant (18-65 ans)
gender : Sexe (M/F/Autre)
height_cm : Taille en centimètres
weight_kg : Poids en kilogrammes
bmi : Indice de masse corporelle calculé à partir de la taille et du poids

- **Paramètres d'activité**

activity_type : Type d'exercice (course à pied, natation, vélo, etc.)
duration_minutes : Durée de la séance d'activité
intensity (intensité) : intensité de l'exercice (faible/moyen/élevé) : Intensité de l'exercice (faible/moyenne/élevée)
calories_burned : Estimation des calories brûlées pendant l'activité
daily_steps : Nombre de pas quotidiens

- **Indicateurs de santé**

avg_heart_rate : Fréquence cardiaque moyenne pendant l'activité
resting_heart_rate : Fréquence cardiaque au repos
blood_pressure_systolic : Tension artérielle systolique
blood_pressure_diastolic : Tension artérielle diastolique
health_condition: Présence de problèmes de santé
smoking_status : Antécédents de tabagisme (jamais/ancien/actuel)

- **Paramètres du mode de vie**

hours_sleep : Heures de sommeil par nuit
stress_level : Niveau de stress quotidien (1-10)
hydration_level : Consommation quotidienne d'eau en litres
fitness_level : Score de forme calculé sur la base de l'activité cumulée

Exemples d'indicateurs :

- Nombre et pourcentage de participants par sexe (gender).
- Groupement des participants par tranche d'âge (e.g., 18-25, 26-35, etc.) et comptage.
- Analyse du BMI moyen pour chaque tranche d'âge afin de détecter des tendances dans la population
- Calculer la durée moyenne (duration_minutes) pour chaque type d'activité (activity_type).
- Moyenne et somme des calories brûlées pour chaque activité.
- Répartition des participants selon l'intensité de leurs exercices (low, medium, high)
- Moyenne des pas quotidiens sur tous les participants
- Fréquence cardiaque moyenne (avg_heart_rate) pour chaque niveau d'intensité
- Calcul de l'écart moyen entre la fréquence cardiaque au repos et pendant l'activité

- Moyennes de la pression artérielle systolique et diastolique
- Comparer les fréquences cardiaques, les niveaux de stress et d'autres métriques entre les groupes "jamais", "ancien" et "actuel" fumeurs
- Répartition des participants ayant ou non des problèmes de santé (health_condition).
- Moyenne des heures de sommeil par nuit pour chaque tranche d'âge
- Moyenne du niveau de stress (stress_level) pour chaque type et intensité d'activité
- Analyse de la corrélation entre les heures de sommeil (hours_sleep) et le score de forme physique (fitness_level)

Projet 3 : Analyse de données log d'un serveur avec PySpark

Données :

```
Last login: Fri Jan 31 09:23:12 on ttys022
serge-nd@MacBook-Pro-de-Serge ~ % ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=105 time=548.917 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=105 time=350.260 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=105 time=530.040 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=105 time=370.254 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=105 time=424.650 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=105 time=142.222 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=105 time=364.631 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=105 time=303.672 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=105 time=315.577 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=105 time=258.308 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=105 time=656.045 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=105 time=377.271 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=105 time=702.369 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=105 time=402.506 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=105 time=537.176 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=105 time=452.802 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=105 time=481.603 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=105 time=335.816 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=105 time=636.983 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=105 time=257.781 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=105 time=662.877 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=105 time=480.569 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=105 time=708.056 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=105 time=626.371 ms
64 bytes from 8.8.8.8: icmp_seq=24 ttl=105 time=397.067 ms
64 bytes from 8.8.8.8: icmp_seq=25 ttl=105 time=667.939 ms
64 bytes from 8.8.8.8: icmp_seq=26 ttl=105 time=383.339 ms
```

Ces données proviennent d'une commande ping exécutée pour mesurer la latence d'accès au serveur DNS public de Google (8.8.8.8). Chaque ligne représente une réponse ICMP recueillie à un intervalle d'une seconde. Voici la structure d'une ligne typique, avec sa signification :

- 64 bytes from 8.8.8.8 : Le paquet reçu contient 64 octets de données et provient de l'adresse IP 8.8.8.8.
- icmp_seq=XX : Le numéro de séquence ICMP, incrémenté à chaque requête envoyée.
- ttl=105 : Le Time-To-Live (TTL) du paquet, indiquant le nombre maximal de sauts restants avant que le paquet ne soit rejeté.
- time=XXX ms : La durée (en millisecondes) que le paquet a mise pour aller et revenir entre l'ordinateur source et l'IP cible.

La date de départ est indiquée dans la première ligne de la capture : Fri Jan 31 09:23:12. Cela correspond au moment où l'utilisateur a démarré la session sur le terminal.

Voici un résumé des informations importantes :

- Date et heure de départ : Vendredi 31 janvier à 09:23:12.
- Adresse cible : 8.8.8.8 (serveur DNS public de Google).
- Intervalle de collecte : Chaque seconde.

Exemple d'indicateurs :

- Temps moyen (mean) de latence : Moyenne des valeurs time=XXX ms
- Temps minimum (min) de latence : Le plus faible temps de réponse observé
- Temps maximum (max) de latence : Le plus grand temps de réponse observé
- Écart-type : Pour analyser la variabilité de la latence
- Nombre total de requêtes envoyées : Nombre de lignes dans les données (basé sur icmp_seq)
- Nombre total de réponses reçues : Lignes où une réponse est enregistrée
- Taux de perte de paquets (Packet Loss Rate) : Pourcentage de requêtes sans réponse

- Calculer la latence moyenne par intervalle de 10 secondes pour détecter les fluctuations temporelles
- Identifier les anomalies où le temps de latence dépasse un certain seuil (par exemple, 500 ms)
- Compter le nombre d'occurrences où la latence est anormalement élevée

- **Projet 4 : Analyse de données du secteur Finance avec pyspark**

Il s'agit d'un échantillon d'une ligne avec l'explication des en-têtes :

1,PAYMENT,1060.31,C429214117,1089.0,28.69,M1591654462,0.0,0.0,0,0

step - représente une unité de temps dans le monde réel. Dans ce cas, 1 pas correspond à 1 heure de temps. Le nombre total d'étapes est de 744 (simulation de 30 jours).
type - ENTRÉE, SORTIE, DÉBIT, PAIEMENT et TRANSFERT.
montant -montant de la transaction en monnaie locale.
nameOrig - client à l'origine de la transaction
oldbalanceOrg - solde initial avant la transaction
newbalanceOrig - nouveau solde après la transaction.
nameDest - client qui est le destinataire de la transaction
oldbalanceDest - solde initial du destinataire avant la transaction. Notez qu'il n'y a pas d'informations pour les clients qui commencent par M (Merchants).
newbalanceDest - destinataire du nouveau solde après la transaction. Notez qu'il n'y a pas d'information pour les clients qui commencent par M (Merchants).
isFraud - Il s'agit des transactions effectuées par les agents frauduleux dans le cadre de la simulation. Dans cet ensemble de données spécifique, le comportement frauduleux des agents vise à faire du profit en prenant le contrôle des comptes des clients et en essayant de vider les fonds en les transférant sur un autre compte, puis en les retirant du système.

isFlaggedFraud - Le modèle économique vise à contrôler les transferts massifs d'un compte à l'autre et signale les tentatives illégales. Dans cet ensemble de données, une tentative illégale est une tentative de transfert de plus de 200 000 euros en une seule transaction.

Exemples d'indicateurs :

- Nombre unique de clients à l'origine des transactions (nameOrig)
- Calculer le nombre total et le montant des transactions par heure ou jour pour détecter des pics d'activité
- Nombre unique de destinataires (nameDest)
- Identifier les clients les plus actifs en termes de nombre ou de montant total des transactions.
- Top 10 des destinataires ayant reçu le plus grand montant
- Calculer la moyenne des soldes avant et après les transactions
- Nombre de transactions où le newbalanceOrig est inférieur à zéro
- Identifier les heures ou jours où le nombre ou le montant des transactions est le plus élevé.
- Nombre total de fraudes détectées (isFraud)
- Identifier les types de transaction les plus associés à des fraudes
- Transactions où nameDest commence par "M"
- etc