Projet Spark
Analyse du trafic des taxis à New York City

Mastering Big Data avec Apache Spark

Introduction

Ce projet est une occasion pour vous familiariser avec la conception et la mise en œuvre de pipelines de traitement et d'analyse de données à grande échelle à l'aide d'**Apache Spark**.

Objectif du projet

L'objectif de ce projet est de concevoir un pipeline de traitement et d'analyse de données à l'aide d'Apache Spark afin d'explorer le trafic des taxis dans la ville de New York.

Vous devez effectuer les étapes suivantes : *ingestion*, *nettoyage*, *transformation* et *analyse exploratoire* des données afin d'identifier des tendances spatio-temporelles, des comportements de paiement et des opportunités d'optimisation (ex. covoiturage urbain).

Jeu de données : NYC Taxi Trip Records

Les données proviennent du site officiel de la **New York City Taxi and Limousine Commission (TLC)** :

`https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page`

Chaque enregistrement correspond à un trajet individuel effectué par un taxi.

Consultez le fichier annexe **DataDictionary–YellowTaxiTripRecords.pdf** contenant la description des données.

Nom de la colonne	Description
tpep_pickup_datetime	Date et heure de prise en charge du passager
tpep_dropoff_datetime	Date et heure de dépot du passager
passenger_count	Nombre de passagers transportés
trip_distance	Distance totale du trajet (en miles)
pickup_longitude / latitude	Coordonnées géographiques du point de départ
dropoff_longitude / latitude	Coordonnées géographiques du point d'arrivée
fare_amount	Montant de la course (hors taxes et suppléments)
extra, mta_tax, tip_amount, tolls_amount, total_amount	Différentes composantes du tarif total payé
payment_type	Type de paiement (1 = carte, 2 = espèces, etc.)
RatecodeID	Code tarifaire (tarif standard, aéroport, etc.)
VendorID	Identifiant du fournisseur de données (taxi jaune, vert, etc.)

Fichiers de zones TLC

Afin de relier les identifiants `PULocationID` et `DOLocationID` à leurs noms réels de zones géographiques, il est nécessaire d'utiliser les fichiers CSV complémentaires fournis par la TLC :

— `taxi_zone_lookup.csv` : fichier contenant la correspondance entre les identifiants de zones et leur description.

Contenu du fichier `taxi_zone_lookup.csv` :

Nom de la colonne	Description
LocationID	Identifiant unique de la zone (correspond à PULocationID / DOLocationID)
Borough	Nom du borough (district administratif, ex. Manhattan, Brooklyn, Queens...)
Zone	Nom spécifique de la zone ou du quartier
service_zone	Type de service (Yellow, Green, etc.)

Lien officiel de téléchargement :

`https://www1.nyc.gov/assets/tlc/downloads/pdf/taxi_zone_lookup.csv`

Utilisation attendue :

— Charger ce fichier dans Spark.

— Joindre les zones via :

  — `PULocationID = LocationID` pour obtenir la zone de départ.

  — `DOLocationID = LocationID` pour obtenir la zone d'arrivée.

— Remplacer les identifiants numériques par les noms de zones dans les analyses et visualisations.

Phase 1 : Ingestion et exploration initiale

Objectifs :

— Charger le jeu de données dans un *DataFrame Spark*

— Examiner le schéma et les types de colonnes

— Identifier les valeurs manquantes ou aberrantes

Questions à traiter :

— Combien de trajets sont disponibles dans l'échantillon ?

— Quelle est la période couverte ?

— Y a-t-il des valeurs nulles ou incohérentes (distance = 0, tarif négatif, etc.) ?

Phase 2 : Nettoyage et transformation des données

Tâches :

— Supprimer ou corriger les lignes contenant des erreurs.

— Convertir les colonnes de dates en format `timestamp`.

— Créer des colonnes dérivées :

  — `trip_duration` : durée du trajet (en minutes)

  — `average_speed` : vitesse moyenne (km/h)

  — `hour` : heure de la journée

  — `day_of_week` : jour de la semaine

— Catégoriser les trajets :

  — `short_trip` : trajets de moins de 10 km

  — `long_trip` : trajets de 10 km ou plus

Questions à traiter :

— Quelle est la distribution des distances et durées ?

— Quelle proportion de trajets courts vs longs observe-t-on ?

— Quelles sont les vitesses moyennes selon les heures ou les jours ?

Phase 3 : Analyse spatio-temporelle

Questions à traiter :

— Quelles zones présentent le plus de départs et d'arrivées ?
— Quelles sont les heures de pointe pour les taxis ?
— Quels sont les 3 principaux points de départ et d'arrivée pour les trajets courts et longs ?

Phase 4 : Analyse des modes de paiement

Questions à traiter :

— Comment les passagers paient-ils les trajets courts et longs ?
— L'utilisation des modes de paiement évolue-t-elle dans le temps ?
— Y a-t-il une relation entre le montant total et le mode de paiement ?

Phase 5 : Exploration du covoiturage (Ride-Sharing)

Questions à traiter :

— Peut-on regrouper des trajets courts ayant des départs proches dans le temps et l'espace ?
— Quelle économie de temps ou d'argent cela pourrait-elle générer ?

Extension à choisir

Vous pouvez aller au-delà des analyses de base et proposer des extensions permettant de valoriser davantage les données. L'objectif est de choisir au moins une extension parmi celles listées ci-dessous et de l'implémenter en utilisant les fonctionnalités avancées d'**Apache Spark** et/ou des bibliothèques associées.

Extensions proposées

— **Analyse avancée et transformations :**
  — Calcul des tendances horaires ou journalières des trajets.
  — Détection des anomalies (trajets très longs ou très courts).
  — Catégorisation des trajets par distance, tarif ou durée.
— **Feature Engineering pour Machine Learning :**
  — Création de nouvelles colonnes telles que `average_speed`, `tip_percentage`, fréquence de trajets par zone.
  — Encodage des informations temporelles (jour de la semaine, heure, périodes de pointe) pour des modèles prédictifs.

— **Modélisation prédictive :**
  — Prédiction du tarif ou de la durée d'un trajet.
  — Prédiction de la demande par zone et par période.
  — Segmentation des trajets ou clients via clustering (ex. KMeans).
— **Optimisation et ride-sharing :**
  — Identification de trajets courts proches dans le temps et l'espace pour proposer du covoiturage.
  — Calcul des gains potentiels en temps et en coût.
— **Visualisations avancées :**
  — Cartes *heatmaps* des zones de départ et d'arrivée.
  — Graphiques temporels pour la demande, les revenus ou les pourboires.
  — Dashboard interactif avec `Databricks`, `Plotly` ou `Folium`.
— **Extension Big Data / temps réel :**
  — Intégration de données externes (météo, trafic, événements).
  — Transformation du pipeline batch en un pipeline *streaming* pour analyser la demande dynamique en quasi temps réel.

Remarque : Il faut choisir au moins une extension et justifier les choix techniques et méthodologiques dans le rapport final.

Livrables attendus

— Code complet (Scala Spark)
— Notebook d'analyse commenté
— Rapport synthétique (PDF ou Markdown)
— Visualisations graphiques