Fruits!

DÉPLOYEZ UN MODÈLE DANS LE CLOUD

Alexandre Delaguillaumie

OBJECTIFS

- Développer des robots cueilleurs intelligents
- Se faire connaître en créant une application mobile pour prendre en photo un fruit et en obtenir des informations
- Mettre en place un première version du moteur de classification des images de fruits

PROBLÉMATIQUES

- LE VOLUME DE DONNÉES VA AUGMENTER TRÈS RAPIDEMENT APRÈS LA LIVRAISON

  ↳ Utiliser des scripts en PySpark et déployer le modèle sur le cloud

- LE TRAVAIL DE L'ALTERNANT EST À COMPLÉTER

  ↳ Réaliser une PCA

- IL FAUT RESPECTER LES CONTRAINTES DU RGPD

  ↳ Serveurs européens

PRÉSENTATION DES DONNÉES

TRAIN SET

Dimensions

67 692 images

UTILISÉ

TEST SET

Dimensions

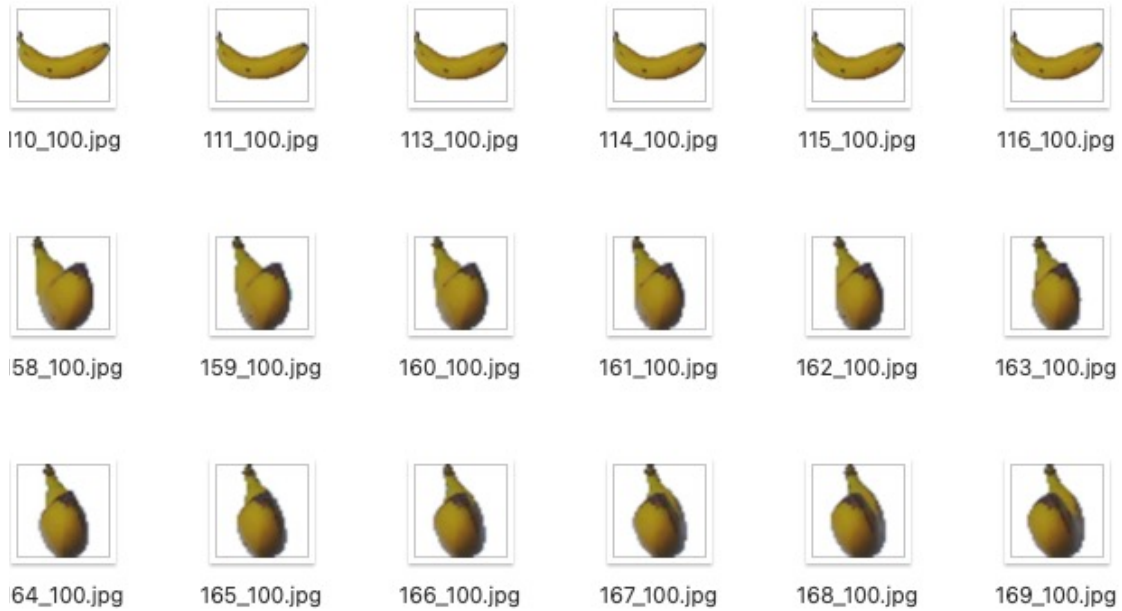
22 688 images (103,2 Mo)

PRÉSENTATION DES DONNÉES

features

- - 131 dossiers (catégories)
- - Images de 100x100 px
- - Format : jpeg
- - Colorimétrie : RGB

Exemple d'un ensemble de prises de vue 3D pour la classe 'Banane'



QUAND FAIRE DU BIG DATA ?

PASSAGE AUX SOLUTIONS CLOUD...

..Si le temps de calcul sur une machine en local dépasse 24h, ou encore si les données sont trop grosses pour être stockées en RAM (ce qui sera le cas d'après le brief)

3V DU BIG DATA

- Volume (stockage)
- Vélocité (temps de traitement)
- Variété (Structure des données)

PRINCIPE DU CALCUL DISTRIBUÉ

Les calculs sont réalisé en parallèle sur des machines distantes, autonomes et qui ne partagent pas de ressources, ce qui permet :

- D'éviter les pannes
- D'augmenter la puissance en augmentant les noeuds

MISE EN PLACE DE L'ENVIRONNEMENT

AWS S3

Les données stockées sous la forme d'objet auxquels on associe des données, des métadonnées et un identifiant

- Simple Storage Service (3S)
- Possibilité de stockage illimitée
- Accès privé grâce au tunnel SSH et aux paires de clés

1. Téléchargement d'AWS Cli

Les données stockées sous la forme d'objet auxquels on associe des données, des métadonnées et un identifiant

2. Création d'un bucket

Les données stockées sous la forme d'objet auxquels on associe des données, des métadonnées et un identifiant

3. Téléchargement des données

À partir du jeu de données disponible sur Kaggle et que l'on télécharge avec Kaggle Cli

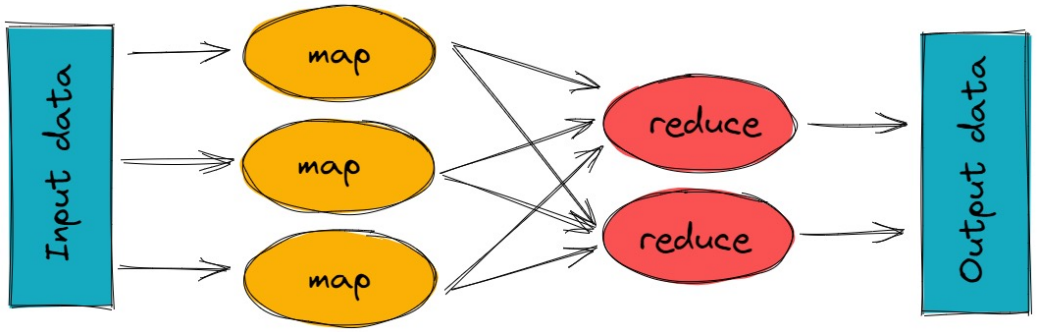
4. Création d'une clé d'accès

Pour effectuer des appels par programmation vers AWS à partir d'AWS CLI

5. Pousser les données

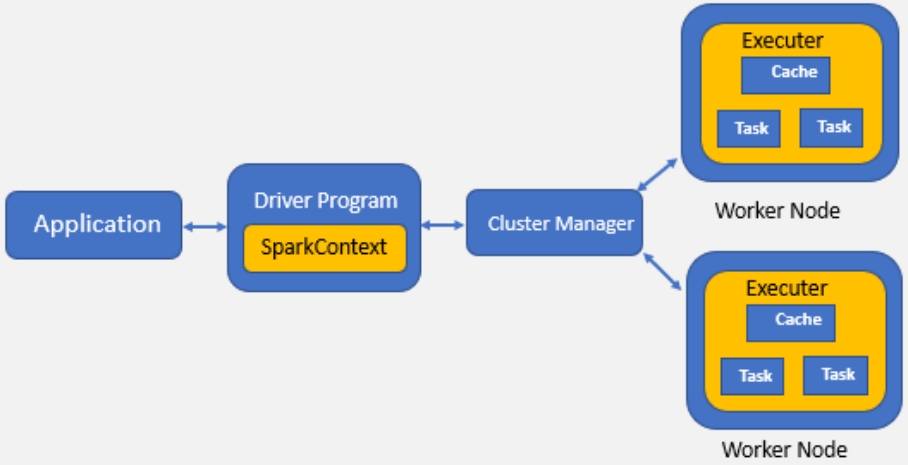
Pour effectuer des appels par programmation vers AWS à partir d'AWS CLI

MISE EN PLACE DE L'ENVIRONNEMENT



AWS EMR

- Elastic Map Reduce (concept fondamental Big Data)
- Permet de simplifier l'exécution des infrastructures Big Data
- Traitement de données distribuées à grande échelle
- Permet de faire fonctionner SPARK



API permettant d'utiliser Apache Spark avec le langage de programmation Python, à partir du langage Scala

VS

Plateforme de calcul open-source s'appuyant sur le concept de clusters, stockage sur RAM

ACCÉDER AU NOTEBOOK SUR LE CLOUD

1. Configuration des logiciels

On sélectionne emr-6.9.0, JupyterHub 1.4.1, TensorFlow 2.10.0 et Spark 3.3.0

2. Générateur JSON de stratégie

On sélectionne le bucket créé sur S3 en persistant

3. Sélection des nœuds

Ici 1 instance maître et 2 instances esclaves en m5.xlarge suffiront pour couvrir les besoins futurs sans trop dépenser, avec un périphérique par défaut de 15go

4. Créer un dossier de journalisation

Dans le bucket existant pour voir l'historique Spark

5. Ajouter une action d'amorçage

Pour installer automatiquement sur toutes les instances les bons packages pour faire tourner le code

```
#!/bin/bash
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install -U pip
sudo python3 -m pip install wheel
sudo python3 -m pip install pillow
sudo python3 -m pip install pandas
sudo python3 -m pip install matplotlib
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
sudo python3 -m pip install keras==2.10.0
```

6. Ajouter une paire de clé

Privée à garder sur son ordinateur pour pouvoir se connecter plus tard en SSH

7. Création d’un tunnel SSH

Qui permet d’accéder aux applications installées sur AWS

7.1. Création des autorisations sur les connexions entrantes

En ajoutant une règle de type SSH en port dans les groupes de sécurité des instances

7.2. Création du tunnel ssh vers le Driver

À l’aide de commandes de type ssh –i PATH/nom_clé hadoop@nom_du_dns

7.3. Configuration et activation d’un proxy (extension FoxyProxy)

En créant un proxy EMR avec l’option SOCKS proxy sur localhost:8157

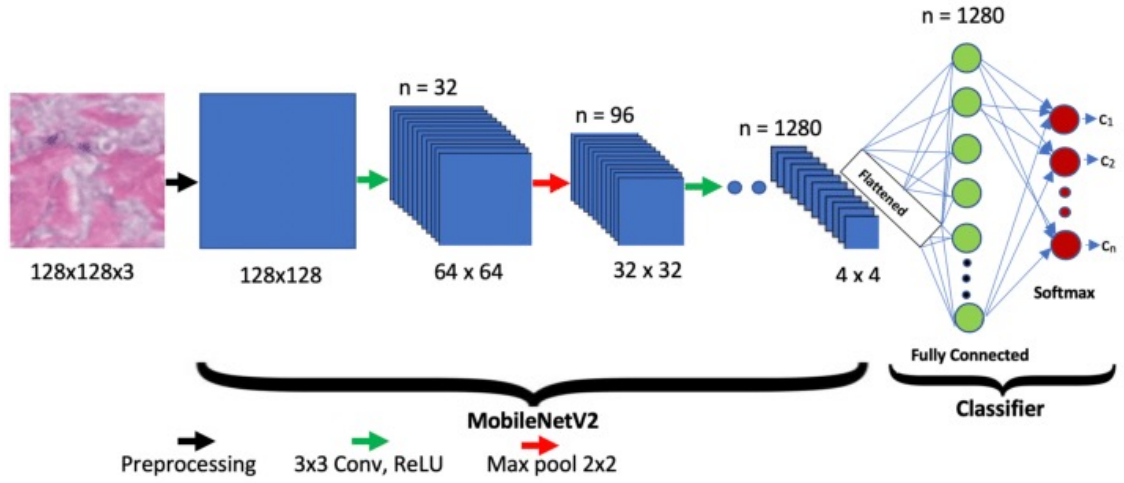
7.4. Accéder à l’url voulue disponible dans l’Historique de l’application

Puis valider l’accès non sécurisé au serveur

ALGORITHME UTILISÉ

MobileNetV2

- Réseau de neurones CNN (Convolution Neural Network)
- Spécialisé en computer vision pour les systèmes embarqués
- On enlève l'avant dernière couche pour fitter sur nos données



Bonnes performances en :
- détection d'objets
- segmentation d'images

« Bottlenecks » :
- réduit le nombre de canaux de données
- réduit la quantité de calcul nécéssaire
- Économise de la mémoire

« inverted residuals » :
- augmenter le nombre de canaux
- complexité globale faible

Pandas UDF

(User-Defined Function) dans Spark sont des fonctions définies par l'utilisateur qui utilisent la bibliothèque Pandas pour effectuer des opérations sur des données dans Spark.

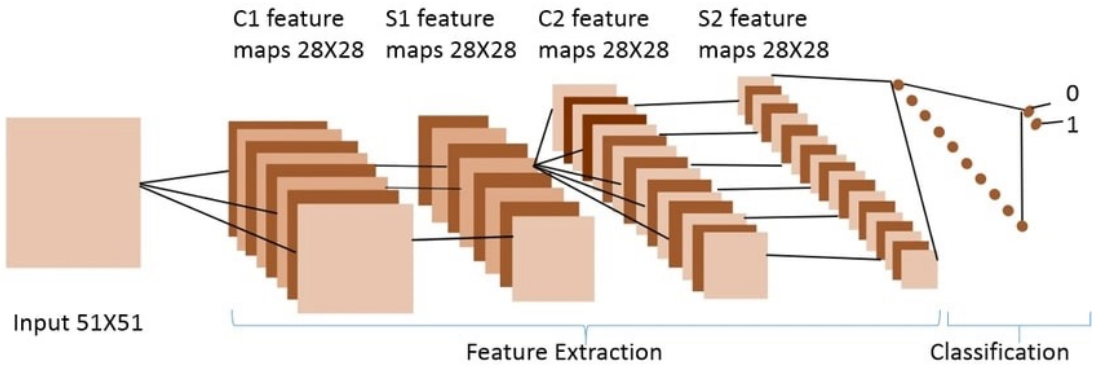
1. Preprocess des images

Mini-Batch learning par descente de gradient
Taille de images influe négativement sur la phase de training et d'inférence

224x224

2. Extraction de features

Utilisation de couches cachées pour détecter les motifs et features importantes
Couches simples et profondes pour capturer les caractéristiques simples (motifs) ou plus complexes (ici un fruit)
Ici on récupère la sortie des couches cachées afin de faire de la segmentation



PCA

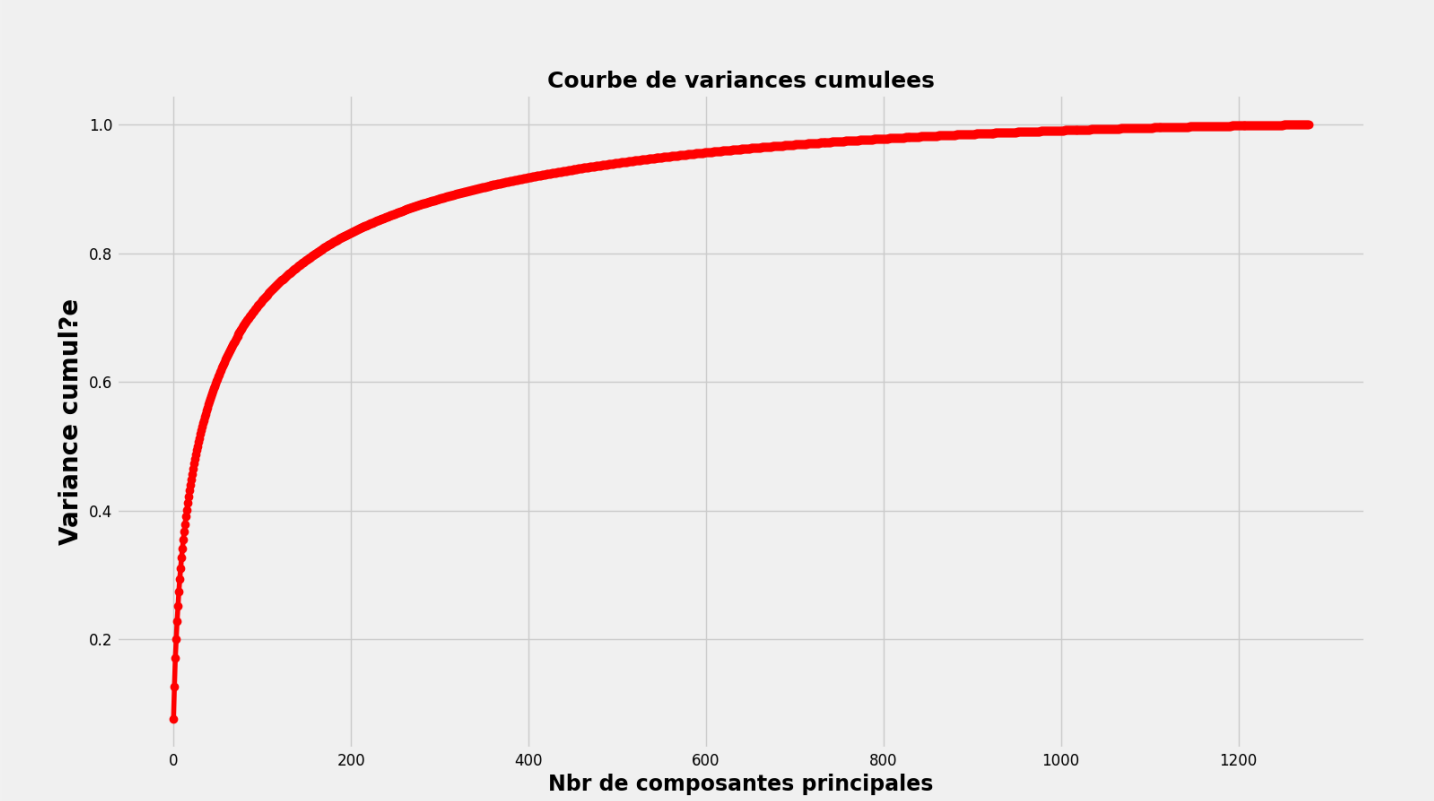
Méthode de réduction de dimension pour transformer un ensemble de variables corrélées en un ensemble de variables (composantes) non corrélées. Chaque axe supplémentaire représente un axe orthogonal au précédent qui maximise la variance

1. Preprocess des features

Standardisation :
- éliminer l'effet d'ordre des grandeurs
- Faire ressortir les relations entre variables

Transformation en vecteur dense :
- Accélerer le traitement des données
- Réduire les coûts de calcul et de mémoire

2. Réduction de dimension



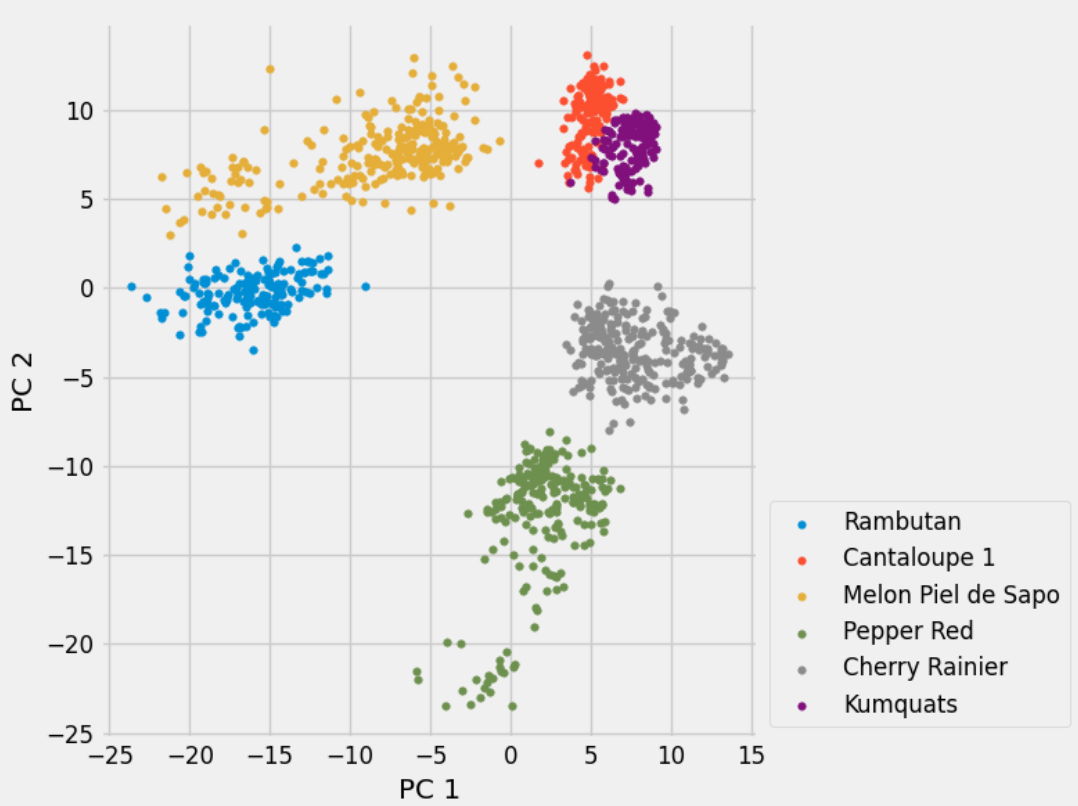
557 composantes

Expliquent 95% de la variance. On supprime donc le reste l'enregistre dans un nouveau DataFrame

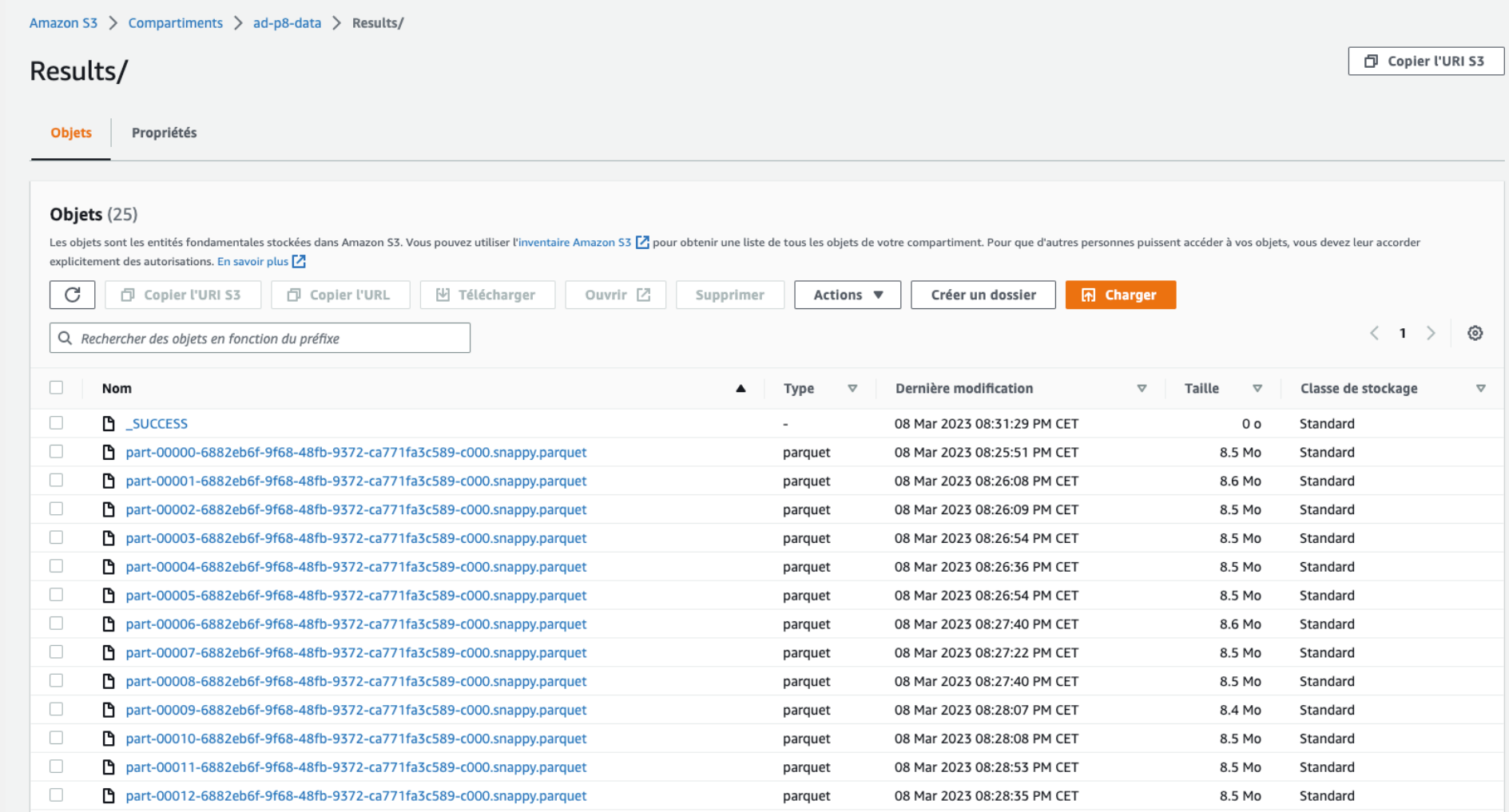
3. Affichage des catégories en 2 dimensions

Sélection aléatoire de 6 catégories
Sur les 2 composantes principales

On voit que le travail de classification fonctionne, les catégories sont relativement disctintes



Export des fichiers au format parquet



Amazon S3 > Compartiments > ad-p8-data > Results/

Results/

Copier l'URI S3

Objets | Propriétés

Objets (25)

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'inventaire Amazon S3 pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. En savoir plus

Copier l'URI S3 | Copier l'URL | Télécharger | Ouvrir | Supprimer | Actions | Créer un dossier | Charger

Rechercher des objets en fonction du préfixe

1

Nom	Type	Dernière modification	Taille	Classe de stockage
_SUCCESS	-	08 Mar 2023 08:31:29 PM CET	0 o	Standard
part-00000-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:25:51 PM CET	8.5 Mo	Standard
part-00001-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:26:08 PM CET	8.6 Mo	Standard
part-00002-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:26:09 PM CET	8.5 Mo	Standard
part-00003-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:26:54 PM CET	8.5 Mo	Standard
part-00004-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:26:36 PM CET	8.5 Mo	Standard
part-00005-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:26:54 PM CET	8.5 Mo	Standard
part-00006-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:27:40 PM CET	8.6 Mo	Standard
part-00007-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:27:22 PM CET	8.5 Mo	Standard
part-00008-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:27:40 PM CET	8.5 Mo	Standard
part-00009-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:28:07 PM CET	8.4 Mo	Standard
part-00010-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:28:08 PM CET	8.5 Mo	Standard
part-00011-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:28:53 PM CET	8.5 Mo	Standard
part-00012-6882eb6f-9f68-48fb-9372-ca771fa3c589-c000.snappy.parquet	parquet	08 Mar 2023 08:28:35 PM CET	8.5 Mo	Standard

Démonstration d'exécution du script PYSpark sur le Cloud

Cluster : p8-fruits En attente Cluster ready to run steps.

Récapitulatif | Historique de l'application | Surveillance | Matériel | Configurations | Événements | Étapes | Actions d'amorçage

Récapitulatif

ID : j-3O27JSCILAU5X
Date de création : 09-03-2023 05:56 (UTC+1)
Temps écoulé : 9 minutes
Résiliation automatique : Cluster waits
Protection de la résiliation : Désactivé Modification
Balises : -- Afficher tout/Modifier
DNS public principal :
ec2-34-240-57-228.eu-west-1.compute.amazonaws.com
Connect to the Master Node Using SSH

Détails de configuration

Étiquette de version : emr-6.9.0
Distribution Hadoop : Amazon
Applications : JupyterHub 1.4.1, Spark 3.3.0, TensorFlow 2.10.0
URI de connexion : s3://ad-p8-data/journal/
Vue cohérente EMRFS : Désactivé
ID d'AMI personnalisée : --
Version d'Amazon Linux : 2.0.20221103.3 En savoir plus

Application user interfaces

Service d'historique : Spark history server, YARN timeline server
Connexions : Not Enabled Activer la connexion Web

Réseau et matériel

Zone de disponibilité : eu-west-1a
ID de sous-réseau (subnet) : subnet-027145824c6edb083
Maître : Action d'amorçage 1 m5.xlarge
Principal : Action d'amorçage 2 m5.xlarge
Tâche : --
Cluster scaling: Not enabled
Résiliation automatique : Not enabled

Historique de session Spark

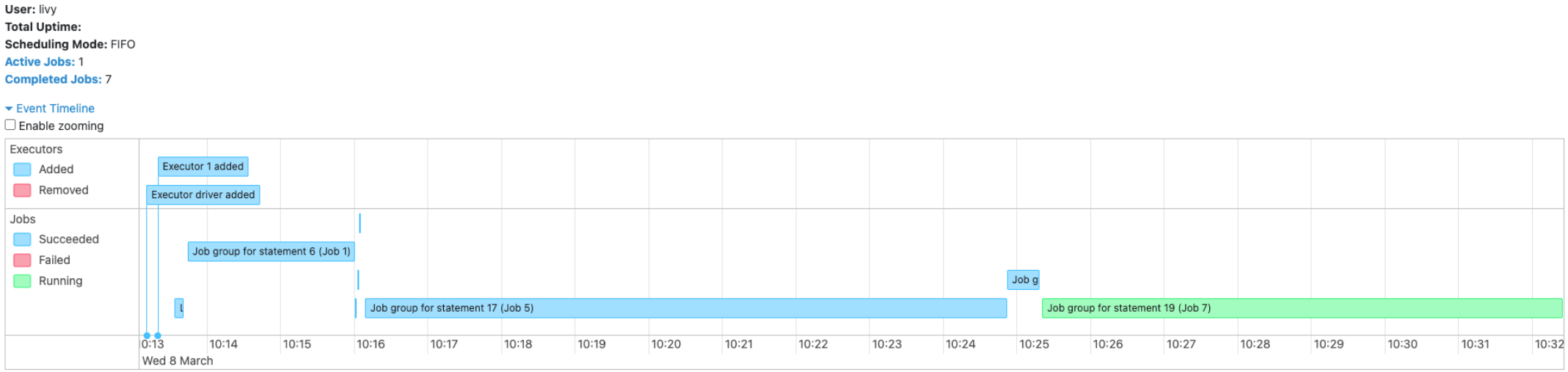
▾ Completed Jobs (54)

Page: 1

1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id (Job Group) ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
57 (159)	Job group for statement 159 toPandas at <stdin>:2	2023/03/08 19:44:47	11 s	1/1	3/3
56 (159)	Job group for statement 159 parquet at NativeMethodAccessorImpl.java:0	2023/03/08 19:44:40	7 s	1/1	1/1
55 (150)	Job group for statement 150 parquet at NativeMethodAccessorImpl.java:0	2023/03/08 19:25:13	6.3 min	1/1 (1 skipped)	24/24 (709 skipped)
54 (150)	Job group for statement 150 parquet at NativeMethodAccessorImpl.java:0	2023/03/08 19:15:41	9.5 min	1/1	709/709
52 (144)	Job group for statement 144 parquet at NativeMethodAccessorImpl.java:0	2023/03/08 18:29:57	6.6 min	1/1 (1 skipped)	24/24 (709 skipped)
51 (144)	Job group for statement 144 parquet at NativeMethodAccessorImpl.java:0	2023/03/08 18:20:03	9.9 min	1/1	709/709
50 (141)	Job group for statement 141 collect at <stdin>:1	2023/03/08 17:54:43	6.2 min	1/1 (1 skipped)	24/24 (709 skipped)
49 (141)	Job group for statement 141 javaToPython at NativeMethodAccessorImpl.java:0	2023/03/08 17:45:13	9.5 min	1/1	709/709
48 (140)	Job group for statement 140 showString at NativeMethodAccessorImpl.java:0	2023/03/08 17:44:37	24 s	1/1 (1 skipped)	1/1 (709 skipped)
47 (140)	Job group for statement 140 showString at NativeMethodAccessorImpl.java:0	2023/03/08 17:34:52	9.8 min	1/1	709/709
46 (140)	Job group for statement 140 treeAggregate at RowMatrix.scala:171	2023/03/08 17:28:20	6.4 min	2/2 (1 skipped)	28/28 (709 skipped)
45 (140)	Job group for statement 140 isEmpty at RowMatrix.scala:441	2023/03/08 17:28:03	18 s	1/1 (1 skipped)	1/1 (709 skipped)
44 (140)	Job group for statement 140 treeAggregate at Statistics.scala:58	2023/03/08 17:21:38	6.4 min	2/2 (1 skipped)	28/28 (709 skipped)
43 (140)	Job group for statement 140 first at RowMatrix.scala:62	2023/03/08 17:21:14	23 s	1/1 (1 skipped)	1/1 (709 skipped)
42 (140)	Job group for statement 140 first at PCA.scala:44	2023/03/08 17:10:53	10 min	2/2	710/710
39 (128)	Job group for statement 128 treeAggregate at RowMatrix.scala:171	2023/03/08 16:49:20	6.5 min	2/2 (1 skipped)	28/28 (709 skipped)
38 (128)	Job group for statement 128	2023/03/08 16:49:02	18 s	1/1 (1 skipped)	1/1 (709 skipped)

Historique de session Spark



Conclusion

1. Nous avons utilisé EMR, S3 et Spark pour anticiper les besoins de l'entreprise
2. Le modèle est privé de bout en bout grâce au tunnel SSH
3. Nous avons répondu aux exigences RGPD avec une localisation en Irlande
4. Nous avons ajouté une PCA pour optimiser le modèle
5. Les données ont été stockées au format .parquet sur S3

MERCI POUR VOTRE ÉCOUTE