

# Fruits!

Pré-traitement des données  
Structure Big Data dans le cloud

Manuel MARTIN - 06-2023



# Fruits !

- Problématique et jeu de données
- Création de l'environnement Big Data
- Chaine de traitement des images
- Démonstration
- Conclusion

# Problématique et jeu de données



# Problématique et jeu de données

## Contexte

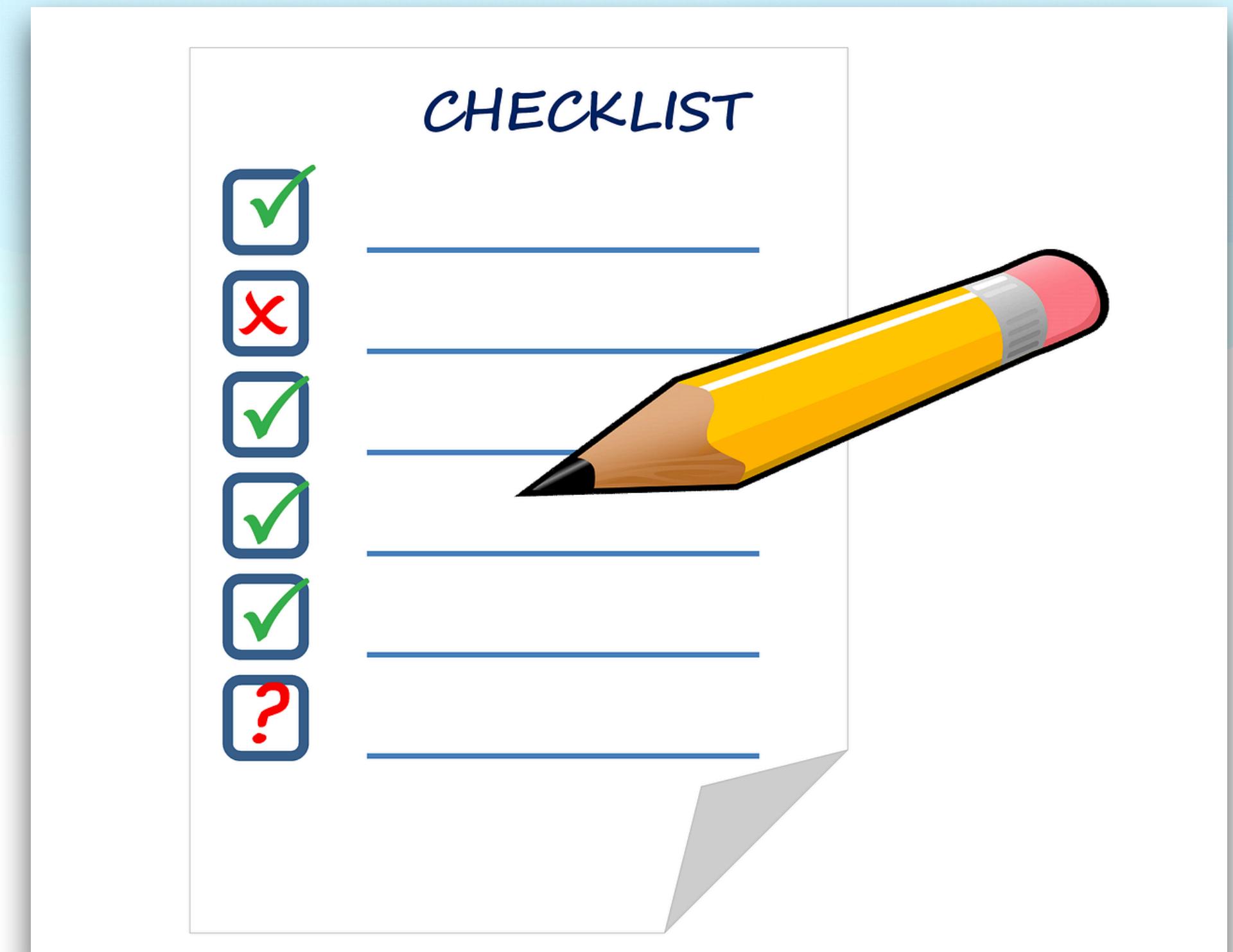
- Application mobile de reconnaissance de fruits
- Moteur de classification des images de fruits
- Architecture Big Data



# Problématique et jeu de données

## Mission

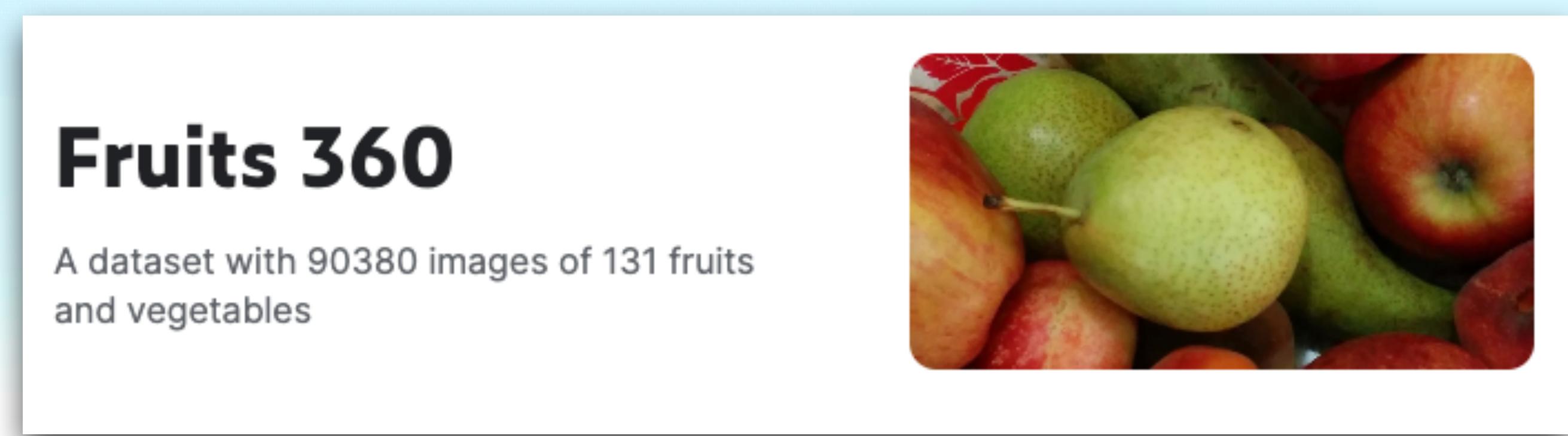
- S'approprier les travaux de l'alternant
- Compléter la chaîne de traitement :
  - Faire une démonstration instance EMR opérationnelle
  - Diffusion des poids du modèle Tensorflow
  - Ajout ACP
- Utiliser le cloud AWS et respecter les contraintes RGPD



# Problématique et jeu de données

## Dataset Kaggle

- Fruits 360
- 90483 images
- 131 fruits et légumes
- 2 versions d'images :
  - Taille 100x100 pixels
  - Taille d'origine
- Plusieurs dossiers : Train, Test, Validation, test-multiple\_fruits



# Création de l'environnement Big Data



## Création de l'environnement Big Data

### Architecture AWS

- Stockage des images
- Stockage des features
- Serveur de modélisation
- AWS
  - S3
  - EMR



## Création de l'environnement Big Data S3

- Stockage des images
  - Stockage des features
- 
- Irlande
    - RGPD
    - Moins cher que Paris



Amazon  
**S3**

# Création de l'environnement Big Data

## S3

- Accessible publiquement pour permettre la revue:
  - du dossier « fruits » contenant les images du dataset original
  - des fichiers parquet et du fichier .csv pour visualiser la réduction de dimension
  - du notebook

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "LecturePublique",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": [  
                "arn:aws:s3:::manuelmartin67-projet8/*",  
                "arn:aws:s3:::manuelmartin67-projet8/fruits/*"  
            ]  
        }  
    ]  
}
```

# Création de l'environnement Big Data S3

- Stockage des images :
  - URL du bucket : <https://manuelmartin67-projet8.s3.eu-west-1.amazonaws.com>
  - URI S3 du dossier « fruits » : <s3://manuelmartin67-projet8/fruits/>
  - URI S3 du dossier « Test\_light » : [s3://manuelmartin67-projet8/Test\\_light/](s3://manuelmartin67-projet8/Test_light/)

	Nom	Type
	bootstrap-emr.sh	sh
	features_reduction_ACP.csv	csv
	fruits/	Dossier
	jupyter/	Dossier
	Results/	Dossier
	Test_light/	Dossier

# Création de l'environnement Big Data S3

- Stockage de la sortie de l'ACP :
  - URL du bucket : <https://manuelmartin67-projet8.s3.eu-west-1.amazonaws.com>
  - URI S3 du dossier « Results » : <s3://manuelmartin67-projet8/Results/>
  - URL du « csv » : [https://manuelmartin67-projet8.s3.eu-west-1.amazonaws.com/features\\_reduction\\_ACP.csv](https://manuelmartin67-projet8.s3.eu-west-1.amazonaws.com/features_reduction_ACP.csv)

	Nom	Type
	bootstrap-emr.sh	sh
	features_reduction_ACP.csv	csv
	fruits/	Dossier
	jupyter/	Dossier
	Results/	Dossier
	Test_light/	Dossier

# Création de l'environnement Big Data S3

- Stockage du notebook :
  - URL : [https://manuelmartin67-projet8.s3.eu-west-1.amazonaws.com/jupyter/joyan/Martin\\_Manuel\\_1\\_notebook\\_062023.ipynb](https://manuelmartin67-projet8.s3.eu-west-1.amazonaws.com/jupyter/joyan/Martin_Manuel_1_notebook_062023.ipynb)

	Nom	Type
	.s3keep	s3keep
	Martin_Manuel_1_notebook_062023.ipynb	ipynb

# Création de l'environnement Big Data EMR

- Serveur de modélisation
- Irlande
  - Même serveur que S3
  - RGPD
  - Moins cher que Paris



Amazon  
EMR

# Création de l'environnement Big Data EMR

- Version Amazon EMR :

- emr-6.11.0

- Groupes d'instances m5.xlarge
    - 1 instance maître
    - 2 instances principales

Applications incluses dans l'offre		
<input type="checkbox"/> Flink 1.16.0	<input type="checkbox"/> Ganglia 3.7.2	<input type="checkbox"/> HBase 2.4.15
<input type="checkbox"/> HCatalog 3.1.3	<input checked="" type="checkbox"/> Hadoop 3.3.3	<input type="checkbox"/> Hive 3.1.3
<input type="checkbox"/> Hue 4.11.0	<input type="checkbox"/> JupyterEnterpriseGateway 2.6.0	<input checked="" type="checkbox"/> JupyterHub 1.4.1
<input type="checkbox"/> Livy 0.7.1	<input type="checkbox"/> MXNet 1.9.1	<input type="checkbox"/> Oozie 5.2.1
<input type="checkbox"/> Phoenix 5.1.2	<input type="checkbox"/> Pig 0.17.0	<input type="checkbox"/> Presto 0.279
<input checked="" type="checkbox"/> Spark 3.3.2	<input type="checkbox"/> Sqoop 1.4.7	<input checked="" type="checkbox"/> TensorFlow 2.11.0
<input type="checkbox"/> Tez 0.10.2	<input type="checkbox"/> Trino 410	<input type="checkbox"/> Zeppelin 0.10.1
<input type="checkbox"/> ZooKeeper 3.5.10		

m5.xlarge
4 vCore 16 GiB mémoire EBS uniquement stockage
Prix à la demande : 0.214 USD par instance/heure
Prix Spot le plus bas : \$0.082 (eu-west-1b)

# Création de l'environnement Big Data EMR

- Actions d'amorçage  
(Bootstraping)
- Librairies à ajouter
- MàJ de Tensorflow
- Paramètres logiciels

```
#!/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==1.2.5
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 tensorflow --upgrade
```

```
1 ▾ []
2 ▾ {
3   "Classification": "jupyter-s3-conf",
4   "Properties": {
5     "s3.persistence.bucket": "manuelmartin67-projet8",
6     "s3.persistence.enabled": "true"
7   }
8 }
9 ]
```

# Création de l'environnement Big Data

## Supervision

- Gestionnaire de ressources  
Hadoop
- Spark History Server



# Chaine de traitement des images



# Chaine de traitement des images

## Process

- Chargement des images
- Préparation du modèle
- Diffusion des poids
- Extraction des features
- Réduction de dimension via ACP
- Export au formats « parquet » et « CSV »



# Chaine de traitement des images

## Chargement des images

- Lecture du dossier contenant les images
- Ajout d'une colonne « label » en fonction du dossier contenant les images

```
root
|-- path: string (nullable = true)
|-- modificationTime: timestamp (nullable = true)
|-- length: long (nullable = true)
|-- content: binary (nullable = true)
|-- label: string (nullable = true)
```

# Chaine de traitement des images

## Préparation du modèle

- Import d'un modèle MobileNetV2
- Chargement des poids « imagenet »
- Taille des images en entrée :  
224x224x3
- Suppression de la dernière couche du réseau
- Récupération pour chaque image de 1280 features

Layer (type) nected to	Output Shape
=====	=====
input_1 (InputLayer)	[ (None, 224, 224, 3) ]
...	

```
global_average_pooling2d ( (None, 1280)
ut_relu[0][0]')
GlobalAveragePooling2D)

=====
=====
Total params: 2257984 (8.61 MB)
Trainable params: 2223872 (8.48 MB)
Non-trainable params: 34112 (133.25 KB)
```

# Chaine de traitement des images

## Diffusion des poids

- Diffusion des poids « imagenet » aux différents workers
- Chaque worker stockera ces informations en mémoire
- Economie de bande passante
- Gain de temps
- Optimisation mémoire

```
sc.broadcast(new_model.get_weights())
```

# Chaine de traitement des images

## Extraction des features

- Pour chaque image :
  - On redimensionne l'image
  - On transforme l'image en array
  - On applique le modèle
  - On récupère une liste de float

path	label	features
file:/Users/manue...	Watermelon	[1.2892096, 0.432...
file:/Users/manue...	Watermelon	[1.190253, 0.3241...
file:/Users/manue...	Watermelon	[0.69916594, 0.22...
file:/Users/manue...	Watermelon	[0.07922348, 0.08...
file:/Users/manue...	Watermelon	[0.17196739, 0.32...
file:/Users/manue...	Watermelon	[0.2004811, 0.061...
file:/Users/manue...	Watermelon	[0.2911226, 0.757...
file:/Users/manue...	Watermelon	[0.9548017, 0.114...
file:/Users/manue...	Watermelon	[0.0, 0.9271429, ...
file:/Users/manue...	Watermelon	[0.12645327, 0.13...
file:/Users/manue...	Watermelon	[0.038168285, 0.5...
file:/Users/manue...	Watermelon	[0.03105981, 1.23...
file:/Users/manue...	Watermelon	[0.26282156, 0.07...
file:/Users/manue...	Watermelon	[0.16239028, 0.29...
file:/Users/manue...	Watermelon	[0.09537591, 0.32...
file:/Users/manue...	Watermelon	[0.47870547, 0.15...
file:/Users/manue...	Watermelon	[0.5650965, 0.163...
file:/Users/manue...	Watermelon	[1.0002056, 0.089...
file:/Users/manue...	Pineapple Mini	[0.0, 5.0259542, ...
file:/Users/manue...	Pineapple Mini	[0.02445907, 4.81...

only showing top 20 rows

# Chaine de traitement des images

## Réduction de dimension via ACP

- Transformation des listes de float en vecteurs et Standard Scaling
- Entrainement d'une première ACP pour récupérer les variances expliquées
- Comptage du nombre de features donnant 100% du total de la variance expliquée
- Réduction via une nouvelle ACP des vecteurs avec le nombre de features retenu
- Retransformation des vecteurs en listes de float

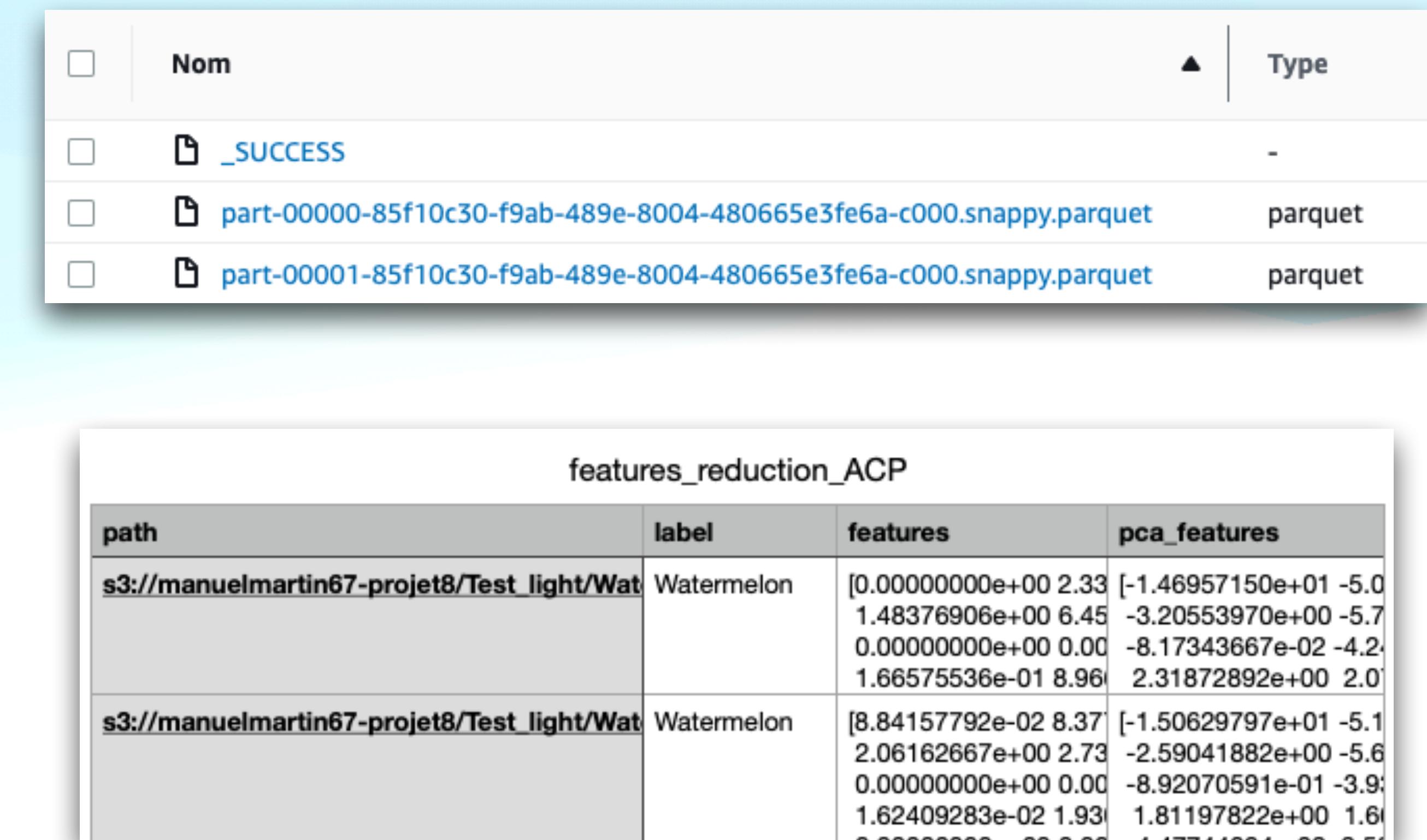
	path	label	features	pca_features
1	file:/Users/manue...	Watermelon	[1.2892096, 0.432...	[-13.651372, -11....
2	file:/Users/manue...	Watermelon	[1.190253, 0.3241...	[-14.176429, -12....
3	file:/Users/manue...	Watermelon	[0.69916594, 0.22...	[-12.516232, -13....
4	file:/Users/manue...	Watermelon	[0.07922348, 0.08...	[-14.515128, -11....
5	file:/Users/manue...	Watermelon	[0.17196739, 0.32...	[-14.429481, -8.5...
6	file:/Users/manue...	Watermelon	[0.2004811, 0.061...	[-14.6891775, -12...
7	file:/Users/manue...	Watermelon	[0.2911226, 0.757...	[-16.33754, -11.7...
8	file:/Users/manue...	Watermelon	[0.9548017, 0.114...	[-10.649744, -11....
9	file:/Users/manue...	Watermelon	[0.0, 0.9271429, ...	[-14.720228, -5.6...
10	file:/Users/manue...	Watermelon	[0.12645327, 0.13...	[-13.13125, -10.4...
11	file:/Users/manue...	Watermelon	[0.038168285, 0.5...	[-15.04185, -6.17...
12	file:/Users/manue...	Watermelon	[0.03105981, 1.23...	[-15.648656, -5.9...
13	file:/Users/manue...	Watermelon	[0.26282156, 0.07...	[-14.650167, -12....
14	file:/Users/manue...	Watermelon	[0.16239028, 0.29...	[-15.264589, -9.9...
15	file:/Users/manue...	Watermelon	[0.09537591, 0.32...	[-15.078143, -10....
16	file:/Users/manue...	Watermelon	[0.47870547, 0.15...	[-13.308122, -13....
17	file:/Users/manue...	Watermelon	[0.5650965, 0.163...	[-12.329149, -13....
18	file:/Users/manue...	Watermelon	[1.0002056, 0.089...	[-12.128314, -13....
19	file:/Users/manue...	Pineapple Mini	[0.0, 5.0259542, ...	[9.177064, 1.1799...
20	file:/Users/manue...	Pineapple Mini	[0.02445907, 4.81...	[9.14801, 0.12183...

only showing top 20 rows

# Chaine de traitement des images

Export au format « parquet » et « csv »

- Export au format « parquet » du dataframe PySpark
- Vérification du résultat en important les fichiers parquet en dataframe Pandas
- Export au format « csv » du dataframe Pandas



The image shows two screenshots illustrating the export process. The top screenshot is a file browser interface with columns for 'Nom' (Name) and 'Type'. It lists '\_SUCCESS', 'part-00000-85f10c30-f9ab-489e-8004-480665e3fe6a-c000.snappy.parquet', and 'part-00001-85f10c30-f9ab-489e-8004-480665e3fe6a-c000.snappy.parquet', all categorized as 'parquet'. The bottom screenshot is a Pandas DataFrame titled 'features\_reduction\_ACP' with columns 'path', 'label', 'features', and 'pca\_features'. It contains two rows corresponding to the Parquet files, both labeled 'Watermelon'.

	Nom	Type
<input type="checkbox"/>	_SUCCESS	-
<input type="checkbox"/>	part-00000-85f10c30-f9ab-489e-8004-480665e3fe6a-c000.snappy.parquet	parquet
<input type="checkbox"/>	part-00001-85f10c30-f9ab-489e-8004-480665e3fe6a-c000.snappy.parquet	parquet

features_reduction_ACP				
path	label	features	pca_features	
s3://manuelmartin67-projet8/Test_light/Wat	Watermelon	[0.0000000e+00 2.33 1.48376906e+00 6.45 0.0000000e+00 0.00 1.66575536e-01 8.96]	[-1.46957150e+01 -5.0 -3.20553970e+00 -5.7 -8.17343667e-02 -4.2 2.31872892e+00 2.0]	
s3://manuelmartin67-projet8/Test_light/Wat	Watermelon	[8.84157792e-02 8.37 2.06162667e+00 2.73 0.0000000e+00 0.00 1.62409283e-02 1.93]	[-1.50629797e+01 -5.1 -2.59041882e+00 -5.6 -8.92070591e-01 -3.9 1.81197822e+00 1.6]	

# Démonstration



# Conclusion



# Conclusion

## Rappels

- S'approprier les travaux de l'alternant
- Compléter la chaîne de traitement :
  - Faire une démonstration instance EMR opérationnelle
  - Diffusion des poids du modèle Tensorflow
  - Ajout ACP
- Utiliser le cloud AWS et respecter les contraintes RGPD



# Conclusion

## Mise en perspective

- Architecture Big Data opérationnelle
- Prête pour accueillir un moteur de classification des images de fruits
- Possibilité de modifier le nombre et le type de workers pour plus de rapidité
- Possibilité d'optimiser les coûts



Merci

