Projet Fruits!

Support de présentation :

Déploiement d’un modèle dans le cloud

Décembre 2021

Sommaire

1. Rappel de la problématique
2. Exploration des données utilisées
3. Modélisation du process en local
4. Déploiement: Cloud & Big Data
5. Mise à l'échelle
6. Conclusion

1. Rappel de la problématique

Mission :

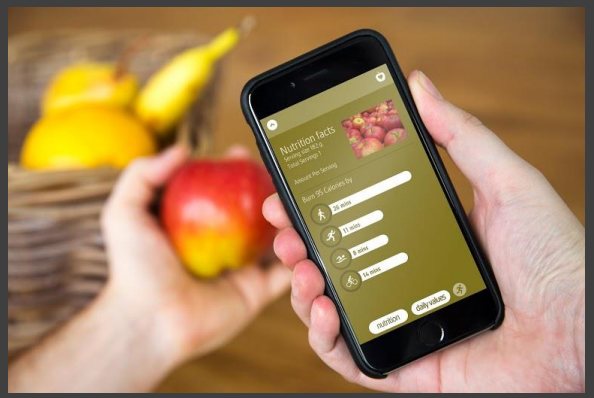
Mettre en place une 1ère architecture Big Data (pré-processing + réduction de dimension) pour la start-up Fruits! pour qu'elle puisse déployer la 1ère version de son moteur de classification d'images de fruits.

- Fruits! Souhaite se faire connaître en proposant :

  > une application mobile grand public
- Prendre une photo de fruits pour l'identifier et obtenir des informations
- Taille des données qui va augmenter rapidement :

  > besoin d'une solution adaptable.

Fruits!



Projet Fruits!

2. Exploration des données utilisées

Jeu de données à disposition -> Fruits 360

https://www.kaggle.com/moltean/fruits

```
Localisation de l'images: /watermelon/r_99_100.jpg
Dimensions de l'image:    (100, 100, 3)
```

0	/raspberry/242_100.jpg	raspberry
1	/raspberry/108_100.jpg	raspberry
2	/raspberry/r_320_100.jpg	raspberry
3	/raspberry/238_100.jpg	raspberry
4	/raspberry/208_100.jpg	raspberry

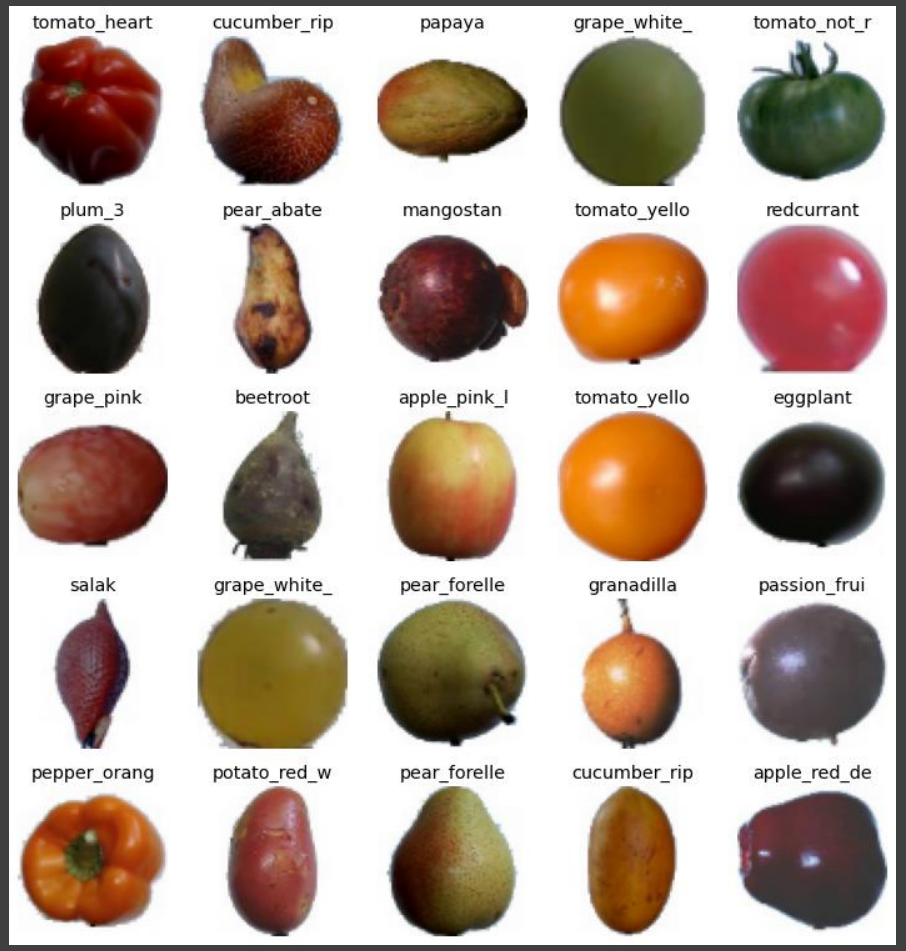
```
y_train.shape (67692,) <class 'numpy.ndarray'>
y_test.shape (22688,) <class 'numpy.ndarray'>
```

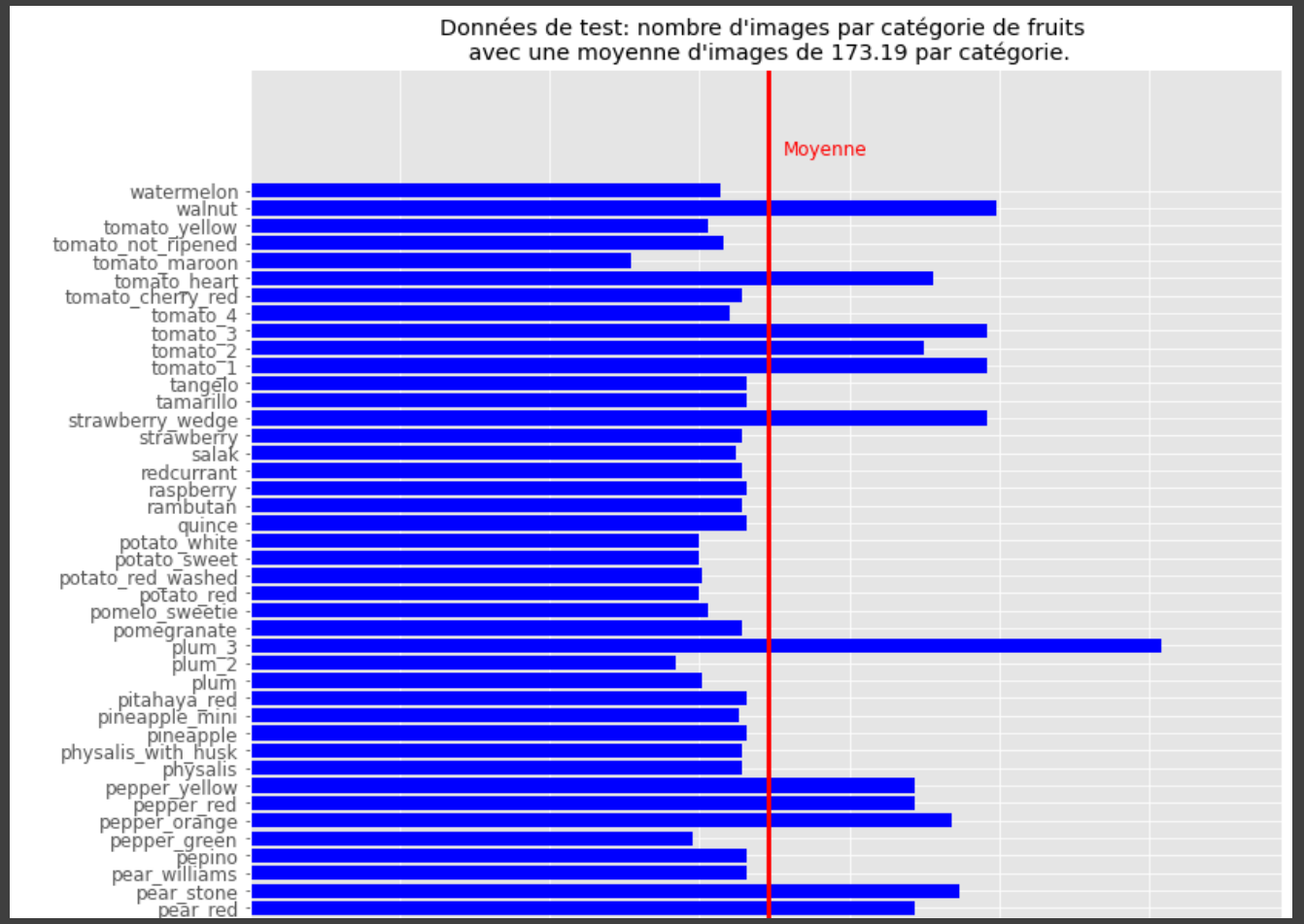
- 131 classes différentes de fruits et légumes
- Présence de plusieurs variétés d'un même fruit / légume.
- Fruits photographiés sur 360 -> intéressant pour l'entraînement d'un modèle
- Un volume d'images de départ conséquent (entraînement / test)
- Format d'image standard couleurs: JPEG en 100x100 pixels, 24 bits, 96 dpi
- Données étiquetées: Labels présents sur l'ensemble du jeu (train / test)
- Jeu d'entraînement = 67 692 images (moy. de 517 fruits / catégorie)
- Jeu de test = 22 688 images (moy. de 174 fruits / catégorie)

Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

2. Exploration des données utilisées





Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

3. Modélisation du process en local

But : Process de modélisation en local -> préparation des images -> baseline de référence
Différentes étapes nécessaires :

1

Sklearn

Skimage

2

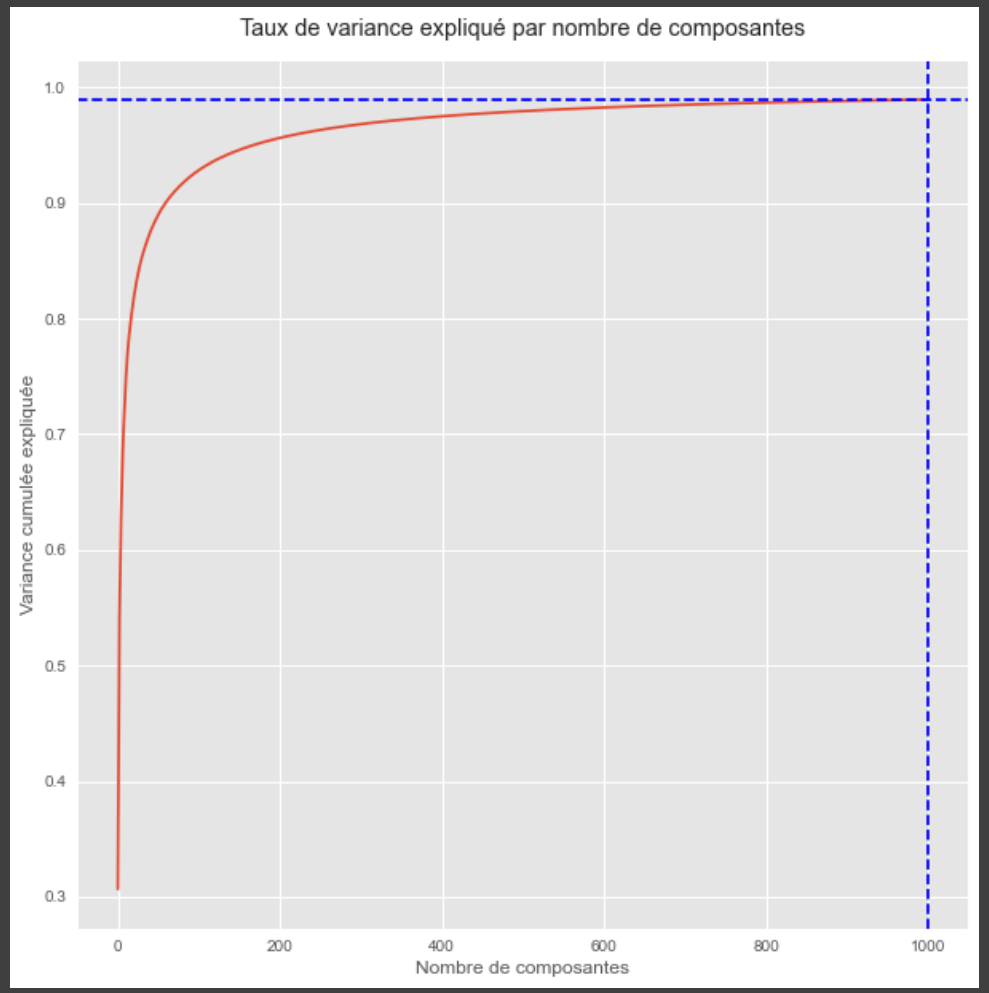
Pyspark

+ Keras

- Création d'un dataframe avec 3 colonnes (label, matrice image, url)
- Réduction des images en 64x64 -> PC peu puissant
- Réduction de dimensions (12 288 -> 1000 features) avec PCA et 99% de variance conservée
- Utilisation d'un classifier "simple" SGDClassifier et RandomForestClassifier (GridSearch allégée)
- Obtention sur le jeu de test de : 70% d'accuracy avec le SGDClassifier et ~83% avec le RFC
- Utilisation d'un modèle pré-entraîné avec ImageNet (Transfer Learning: InceptionV3 extraction de features)
- Transposition du process à Pyspark avec cluster en mode local (windows) et pseudo-distribué (linux)
- Process Hadoop / Pyspark transposable vers cluster Amazon EMR + Amazon S3

Projet Fruits!

3. Modélisation du process en local




```
# rapport de classification
print(classification_report(y_test, y_pred_rfc_std_pca))
```

116	0.93	0.87	0.90	164
117	0.88	0.53	0.66	246
118	0.91	1.00	0.95	166
119	0.93	0.97	0.95	166
120	0.94	0.99	0.97	246
121	0.90	1.00	0.95	225
122	0.97	0.83	0.89	246
123	0.87	1.00	0.93	160
124	0.78	1.00	0.88	164
125	0.90	0.76	0.83	228
126	0.93	1.00	0.97	127
127	0.78	0.87	0.82	158
128	0.94	1.00	0.97	153
129	0.47	1.00	0.64	249
130	0.73	0.97	0.83	157
accuracy			0.83	22688
macro avg	0.84	0.82	0.82	22688
weighted avg	0.84	0.83	0.82	22688

Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

3. Modélisation du process en local

Modélisation du process avec Pyspark + Keras :

Fruits 360

Images en

100x100x3

DataFrame

url | label | matrice

Transformation

Pandas UDF (matrice)

Extraction de features

avec Inception V3

path	features	label
file:/G:/p8_jc_im...	[0.23239073, 0.0,...	raspberry
file:/G:/p8_jc_im...	[0.0, 0.0, 0.4001...	raspberry
file:/G:/p8_jc_im...	[0.0, 1.1064178, ...	pineapple_mini
file:/G:/p8_jc_im...	[0.01444225, 0.0,...	raspberry
file:/G:/p8_jc_im...	[1.0919065, 0.0, ...	raspberry
file:/G:/p8_jc_im...	[0.6957115, 0.0, ...	raspberry
file:/G:/p8_jc_im...	[0.0, 2.9302442, ...	pineapple_mini
file:/G:/p8_jc_im...	[0.59812486, 0.0,...	raspberry
file:/G:/p8_jc_im...	[1.128924, 0.0, 1...	raspberry

Sauvegarde

Parquet

Données

prêtes pour la

classification

Spark ML Pipeline:

(indexer, scaler, pca

+ sélection Modèle)

Tuning

Hyperparamètres

Entraînement modèle

Évaluation sur les

données de Test

1ère

Version

de moteur

de

classification

Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

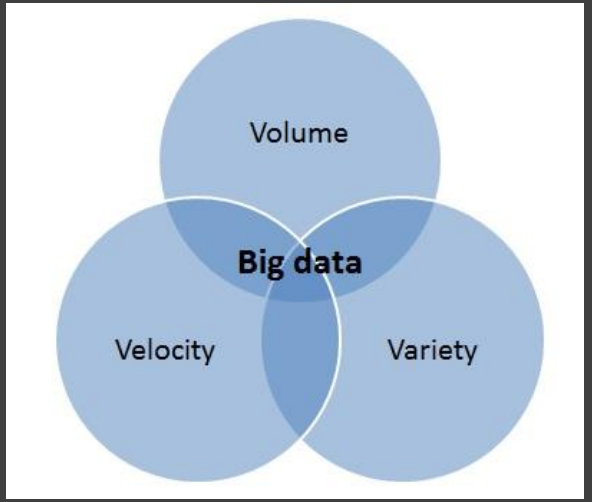
4. Déploiement: Cloud & Big Data

Explications sur les solutions technologiques employées (local & cloud):

Pourquoi ? Pyspark (Hadoop), Inception V3 (Keras), Amazon EMR & Amazon S3

Impératif de l'enjeu : Application publique à fort taux d'augmentation des données

Une prise en charge par une structure « finie » classique n'est plus possible



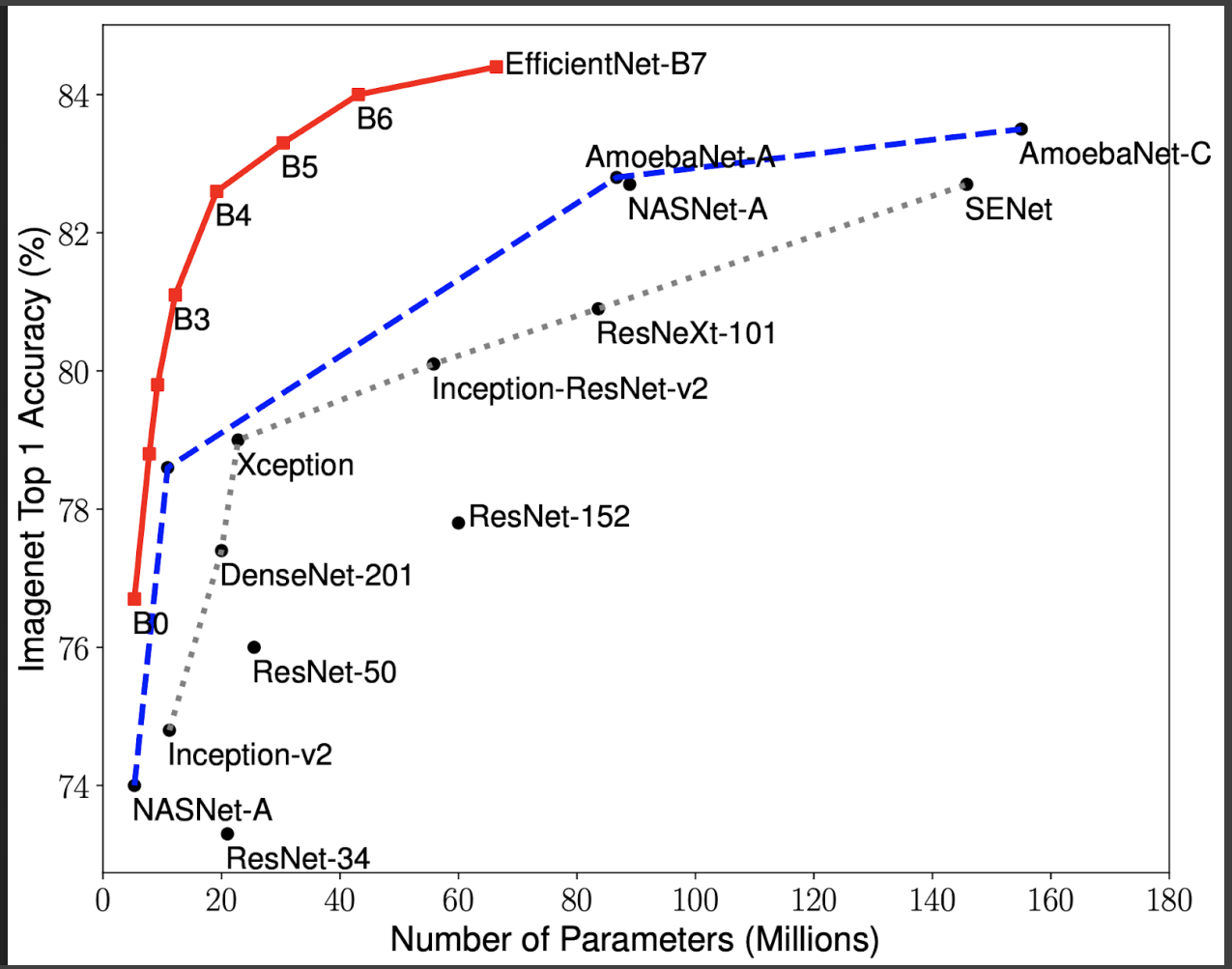
- Pourquoi un déploiement Cloud ? Nécessité : « Loi » des 3 V
- Avantages du cluster computing -> Travail en parallèle & distribué avec (Batch processing = travail fractionné en petites parties) + se prémunir contre pannes
- Hadoop: Stockage des données distribuées (HDFS)
- Spark (Pyspark): Calculs avec les données distribuées (RAM, API, Opensource)
- Keras: utilisation modèle pré-entraîné avec ImageNET
  - > InceptionV3 (Extraction de features) + possibilité Fine tuning
- Services Amazon: facilité au déploiement -> solution clef en main

Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

4. Déploiement: Cloud & Big Data

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	0.790	0.945	22,910,480	126	109.42	8.06
VGG16	528	0.713	0.901	138,357,544	23	69.50	4.16
VGG19	549	0.713	0.900	143,667,240	26	84.75	4.38
ResNet50	98	0.749	0.921	25,636,712	-	58.20	4.55
ResNet101	171	0.764	0.928	44,707,176	-	89.59	5.19
ResNet152	232	0.766	0.931	60,419,944	-	127.43	6.54
ResNet50V2	98	0.760	0.930	25,613,800	-	45.63	4.42
ResNet101V2	171	0.772	0.938	44,675,560	-	72.73	5.43
ResNet152V2	232	0.780	0.942	60,380,648	-	107.50	6.64
InceptionV3	92	0.779	0.937	23,851,784	159	42.25	6.86
InceptionResNetV2	215	0.803	0.953	55,873,736	572	130.19	10.02
MobileNet	16	0.704	0.895	4,253,864	88	22.60	3.44
MobileNetV2	14	0.713	0.901	3,538,984	88	25.90	3.83
DenseNet121	33	0.750	0.923	8,062,504	121	77.14	5.38
DenseNet169	57	0.762	0.932	14,307,880	169	96.40	6.28
DenseNet201	80	0.773	0.936	20,242,984	201	127.24	6.67
NASNetMobile	23	0.744	0.919	5,326,716	-	27.04	6.70
NASNetLarge	343	0.825	0.960	88,949,818	-	344.51	19.96
EfficientNetB0	29	-	-	5,330,571	-	46.00	4.91

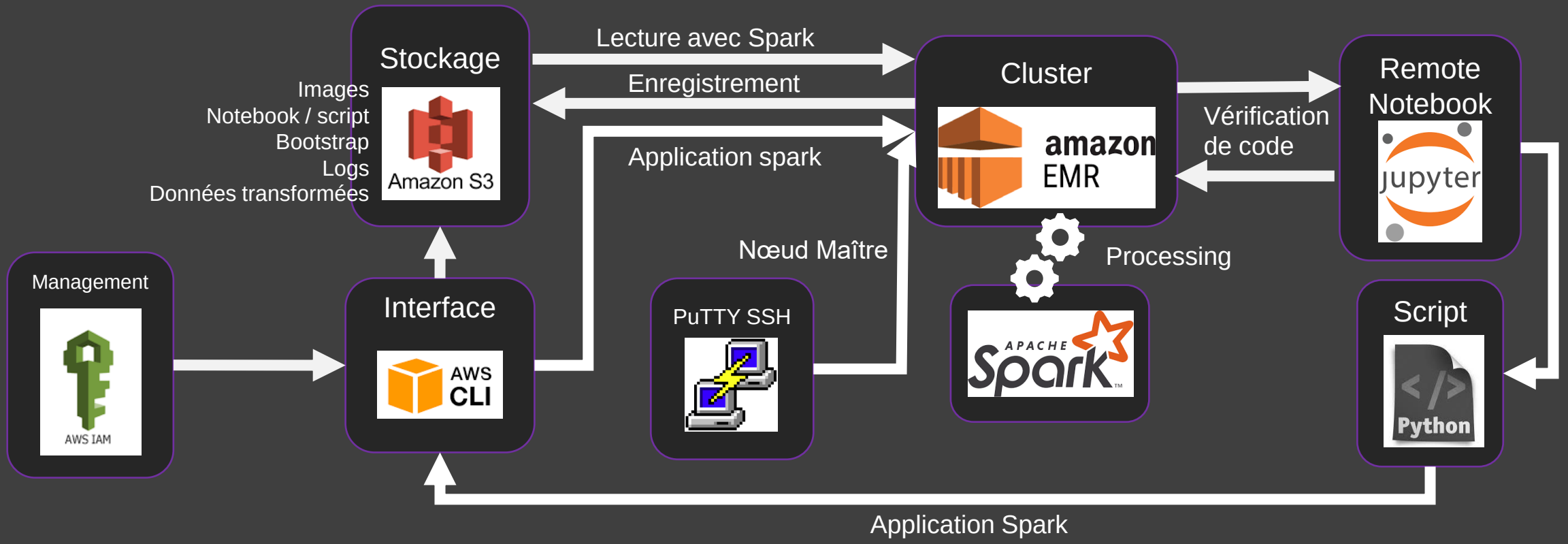


Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

4. Déploiement: Cloud & Big Data

Schématisation de l'architecture cloud:



Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

5. Mise à l'échelle

- Pas de limitation !
- Amazon EMR : choix de départ -> cluster c4.xlarge ou c4.2xlarge (4 Core avec 8/16 Go RAM) -> 1x MasterNode + 2x CoreNode
- Cluster peut être redimensionné à la demande: rajout d'instances EC2 (scaling auto programmable)
- Puissance de calcul : RAM (GPU) possible -> classification
- Amazon S3 : Taille automatique
- Pas de modification de code à prévoir sauf éventuellement si « Scale In » pour réduction de coût -> prévoir de ne pas surcharger le driver (augmenter le nombre de partitions par exemple)

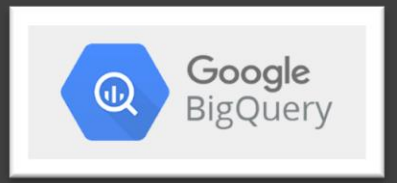
Rajout d'instances EC2

Projet Fruits!

6. Conclusion & axe(s) de réflexion(s)

- Entraînement différents modèles et choisir le meilleur pour le moteur de classification
- Utiliser le Fine Tuning avec le modèle pour comparaison avec le Transfer Learning
- Réentraînement du modèle avec des images de qualité différentes (sans fond blanc, etc)
- + même type de fruits avec des degrés de maturité différents (pas la même couleur/forme)
- Tester et comparer avec d'autres services (Microsoft Azure Hdinsight, Google, Snowflake, etc)

-> But comparer le meilleur rapport coût / performances / facilité de déploiement & maintenance



Projet Fruits!

Support présentation: Déploiement d'un modèle dans le cloud

Annexe: Listing des éléments fournis

1. Données images utilisées pour le projet: https://www.kaggle.com/moltean/fruits
2. Données finales préparées (format parquet): converties en fichiers csv compressés (gzip).
3. Notebook général de prototypage avec l'ensemble des étapes de réflexion
4. Notebook Pyspark utilisé sur le cluster AWS EMR 6.5.0 pour l'obtention des résultats (transformable en script)
5. Dossier de captures d'écrans AWS montrant l'organisation et les étapes de réalisation du projet
6. Powerpoint de présentation au format Microsoft
7. Dossier images de références ou d'illustrations utilisées dans le notebook et la présentation
8. Capture d'écrans des fichiers volumineux intermédiaires générés pour le prototypage