PROJET 8 :

Déployez un modèle dans le cloud

Livrable n° 2 :

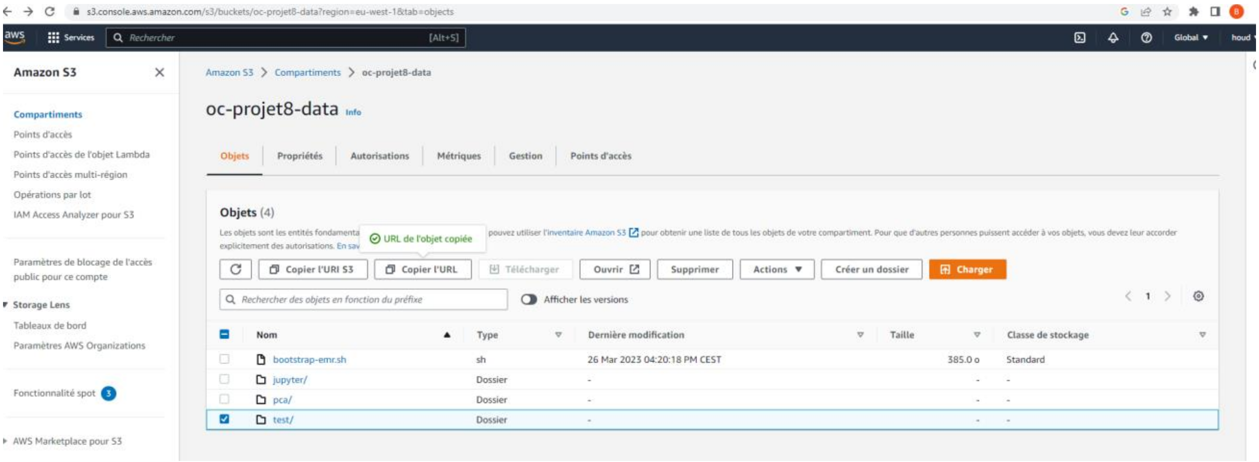
- images du jeu de données (dossier test)

https://oc-projet8-data.s3.eu-west-1.amazonaws.com/test/

- matrice de la réduction de dimension en format parquet (dossier pca) :

https://oc-projet8-data.s3.eu-west-1.amazonaws.com/pca/

Stockage dans le cloud dans le bucket aws : s3://oc-projet8-data



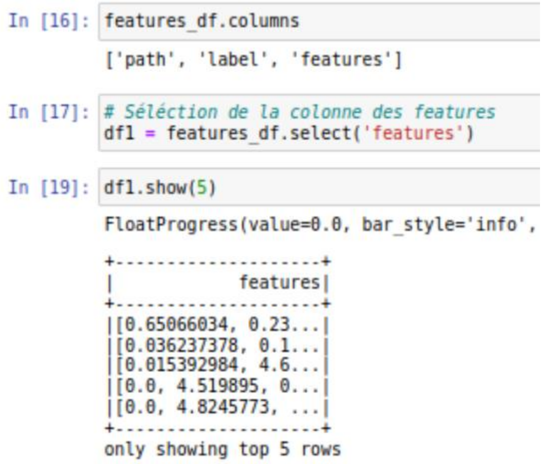
Objets (4)

Les objets sont les entités fondamenta 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 sav

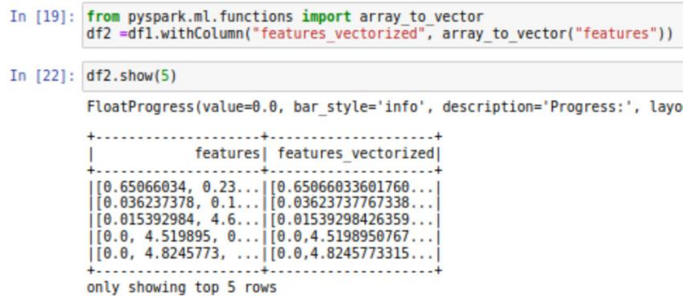
✓ URL de l'objet copiée

☐	Nom	Type	Dernière modification	Taille	Classe de stockage
☐	bootstrap-emr.sh	sh	26 Mar 2023 04:20:18 PM CEST	385.0 o	Standard
☐	jupyter/	Dossier	-	-	-
☐	pca/	Dossier	-	-	-
☑	test/	Dossier	-	-	-

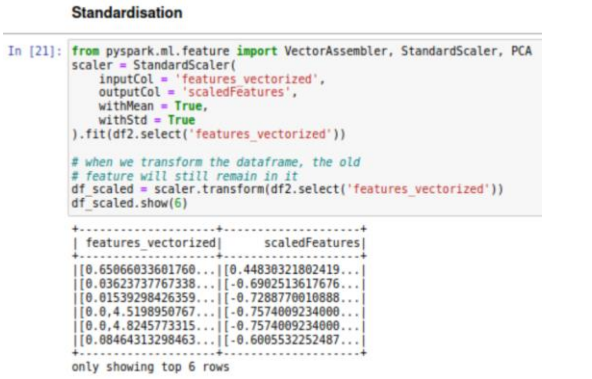
- Sélection de la colonne des features, standardisation des données, réduction de dimension puis enregistrement en format parquet dans le dossier pca du bucket aws : s3://oc-projet8-data:



```
In [16]: features_df.columns

         ['path', 'label', 'features']

In [17]: # Séléction de la colonne des features
         df1 = features_df.select('features')

In [19]: df1.show(5)

         FloatProgress(value=0.0, bar_style='info',

         +--------------------+
         |            features|
         +--------------------+
         |[0.65066034, 0.23...|
         |[0.036237378, 0.1...|
         |[0.015392984, 4.6...|
         |[0.0, 4.519895, 0...|
         |[0.0, 4.8245773, ...|
         +--------------------+
         only showing top 5 rows
```

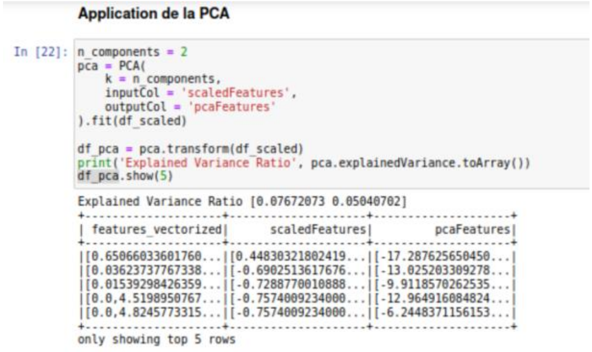


```
In [19]: from pyspark.ml.functions import array_to_vector
         df2 =df1.withColumn("features_vectorized", array_to_vector("features"))

In [22]: df2.show(5)

         FloatProgress(value=0.0, bar_style='info', description='Progress:', layo

         +--------------------+--------------------+
         |            features| features_vectorized|
         +--------------------+--------------------+
         |[0.65066034, 0.23...|[0.65066033601760...|
         |[0.036237378, 0.1...|[0.03623737767338...|
         |[0.015392984, 4.6...|[0.01539298426359...|
         |[0.0, 4.519895, 0...|[0.0,4.5198950767...|
         |[0.0, 4.8245773, ...|[0.0,4.8245773315...|
         +--------------------+--------------------+
         only showing top 5 rows
```



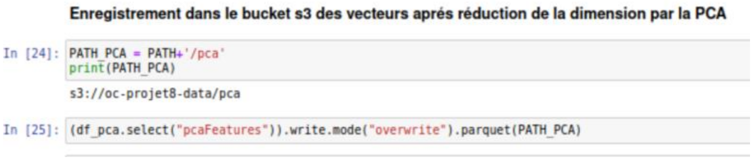
Standardisation

```
In [21]: from pyspark.ml.feature import VectorAssembler, StandardScaler, PCA
         scaler = StandardScaler(
             inputCol = 'features_vectorized',
             outputCol = 'scaledFeatures',
             withMean = True,
             withStd = True
         ).fit(df2.select('features_vectorized'))

         # when we transform the dataframe, the old
         # feature will still remain in it
         df_scaled = scaler.transform(df2.select('features_vectorized'))
         df_scaled.show(6)

         +--------------------+--------------------+
         | features_vectorized|      scaledFeatures|
         +--------------------+--------------------+
         |[0.65066033601760...|[0.44830321802419...|
         |[0.03623737767338...|[-0.6902513617676...|
         |[0.01539298426359...|[-0.7288770010888...|
         |[0.0,4.5198950767...|[-0.7574009234000...|
         |[0.0,4.8245773315...|[-0.7574009234000...|
         |[0.08464313298463...|[-0.6005532252487...|
         +--------------------+--------------------+
         only showing top 6 rows
```



Application de la PCA

```
In [22]: n_components = 2
         pca = PCA(
             k = n_components,
             inputCol = 'scaledFeatures',
             outputCol = 'pcaFeatures'
         ).fit(df_scaled)

         df_pca = pca.transform(df_scaled)
         print('Explained Variance Ratio', pca.explainedVariance.toArray())
         df_pca.show(5)

         Explained Variance Ratio [0.07672073 0.05040702]
         +--------------------+--------------------+--------------------+
         | features_vectorized|      scaledFeatures|         pcaFeatures|
         +--------------------+--------------------+--------------------+
         |[0.65066033601760...|[0.44830321802419...|[-17.287625650450...|
         |[0.03623737767338...|[-0.6902513617676...|[-13.025203309278...|
         |[0.01539298426359...|[-0.7288770010888...|[-9.9118570262535...|
         |[0.0,4.5198950767...|[-0.7574009234000...|[-12.964916084824...|
         |[0.0,4.8245773315...|[-0.7574009234000...|[-6.2448371156153...|
         +--------------------+--------------------+--------------------+
         only showing top 5 rows
```



Enregistrement dans le bucket s3 des vecteurs aprés réduction de la dimension par la PCA

```
In [24]: PATH_PCA = PATH+'/pca'
         print(PATH_PCA)

         s3://oc-projet8-data/pca

In [25]: (df_pca.select("pcaFeatures")).write.mode("overwrite").parquet(PATH_PCA)
```