Pratique Spark ML

In [0]:

```
# Charger le data
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
data = spark.read.csv('/FileStore/tables/boston_housing.csv',
                      header=True, inferSchema=True)
```

In [0]:

```
# Creation des features
features = data.columns[:-1]
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=features,outputCol="features")
df = assembler.transform(data)
```

In [0]:

```
# train/test split
train, test = df.randomSplit([0.8, 0.2])
```

In [0]:

```
#Definir le modele
from pyspark.ml.regression import LinearRegression
lr_model = LinearRegression(featuresCol="features", labelCol="medv")
# training du modele
model = lr_model.fit(train)
```

In [0]:

```
# Evaluation
evaluation_summary = model.evaluate(test)
print("MAE:{}".format(evaluation_summary.meanAbsoluteError))
print("RMSE:{}".format(evaluation_summary.rootMeanSquaredError))
print("R-squared:{}".format(evaluation_summary.r2))
```

```
MAE:3.6573400871928583 RMSE:5.767411451028 R-squared:0.6121244586872179
```

In [0]:

```
# Prediction
predictions = model.transform(test)
predictions.select(predictions.columns[13::2]).show()
```

```
+----+------------------+ medv| prediction| +----+------------------+ 32.2| 32.28395328409807|
29.1|31.673097181961214| 30.1| 24.86547469497102| 20.1| 20.53018133272951| 31.1|
32.4449742471191| 25.0|29.324301150313488| 26.6| 21.36302103031064|
19.4|22.965646656553403| 22.0| 29.28001294710709| 22.9|24.947776006272484| 27.9|
32.57656433881376| 35.4| 34.49903093177059| 21.1| 20.21818976688882|
20.6|27.438999392431942| 18.2|13.539048796399086| 22.3| 27.15042372324784|
11.9|21.926578693945732| 28.2|33.497262984841754| 27.1|26.902335127155006|
50.0|36.375954349869744| +----+------------------+ only showing top 20 rows
```