

# Conceptos y Aplicaciones de Big Data

---

MLLIB Y GRAPHX

Prof. Waldo Hasperué  
[whasperue@lidi.info.unlp.edu.ar](mailto:whasperue@lidi.info.unlp.edu.ar)

# Temario

---

MLlib

GraphX

# Inteligencia de datos

---

La inteligencia de datos es el proceso de analizar datos complejos y presentar información de tal manera que ayude a los usuarios a tomar decisiones de negocio basados en la información analizada y en tiempo real.

La inteligencia de negocios es la recolección de resultados arrojados por muchos programas y sistemas que analizan flujos de datos de gran tamaño y se utiliza para generar información esencial que ayude al proceso de toma de decisiones.

# Inteligencia de datos

---

La inteligencia de datos y de negocios puede agregar valor a cualquier proceso de negocios:

- ❖ Marketing digital
- ❖ Aumento de las tasas de membresía
- ❖ Maximización de la eficiencia operativa
- ❖ Vista unificada de operaciones
- ❖ Identificación a estudiantes en riesgo

# Machine learning

---

Un sistema inteligente es aquel sistema capaz de resolver problemas complejos y multidisciplinarios de una forma automática dando soporte a las decisiones de un experto:

- ❖ Algoritmos simbolistas. Razonamiento inductivo
- ❖ Redes neuronales artificiales
- ❖ Algoritmos genéticos y evolutivos
- ❖ Probabilísticos. Teorema de Bayes.
- ❖ Algoritmos de "similitudes". K-NN, SVM

# Sistemas inteligentes en Big Data

---

Los algoritmos que implementan los sistemas inteligentes son algoritmos iterativos, por lo tanto tienen que dar varias "pasadas" a los datos para llevar a cabo su tarea.

Se los conocen como algoritmos de aprendizaje de máquina (machine learning).

Deben estar optimizados para un óptimo rendimiento.

# MLlib

---

MLlib es la librería de algoritmos de machine learning para Spark.

Los algoritmos están diseñados e implementados para ejecutarse de manera eficiente en un ambiente distribuido

# MLlib - Algoritmos

---

- ❖ Logistic regression
- ❖ Naive Bayes
- ❖ Generalized linear regression
- ❖ Survival regression
- ❖ Decision trees
- ❖ Random forests
- ❖ Gradient-boosted trees
- ❖ Alternating least squares (ALS)
- ❖ K-means
- ❖ Gaussian mixtures
- ❖ Latent Dirichlet allocation (LDA)
- ❖ Frequent itemsets
- ❖ Association rules
- ❖ Sequential pattern mining

# MLlib – Ejemplo de uso

---

```
from pyspark.ml.regression import LinearRegression
# Carga de los datos
datos = sc.textFile("datos.txt")
lr = LinearRegression(maxIter=10, regParam=0.01,
                      elasticNetParam=0.01)
# Entrenamiento del modelo
lrModel = lr.fit(datos)
# Resultados
print("Coefficients: %s" % str(lrModel.coefficients))
print("Intercept: %s" % str(lrModel.intercept))
```

# MLlib – Ejemplo de uso

---

```
from pyspark.ml.classification import DecisionTreeClassifier
# Carga de los datos
datos = sc.textFile("datos.txt")
dt = DecisionTreeClassifier(labelCol="indexedLabel",
                           featuresCol="indexedFeatures")
# Entrenamiento del modelo
modelo = dt.fit(datos)
#####
# Predicciones
nuevos = sc.textFile("nuevos.txt")
predicciones = modelo.transform(nuevos)
```

# MLlib – Más ejemplos

---

Decenas de ejemplos listos para copiar, pegar y ejecutar en:

- Python
- Java
- Scala

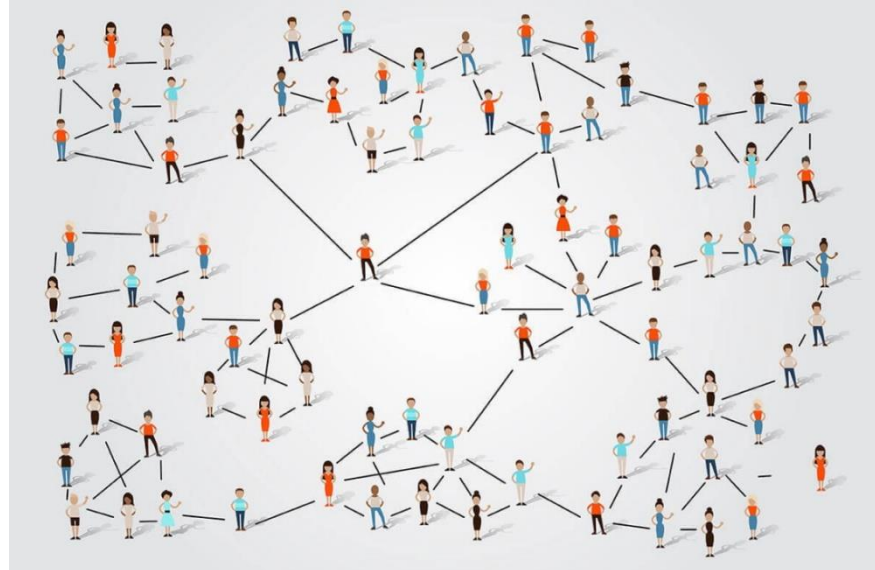
<https://spark.apache.org/docs/3.0.0-preview/ml-guide.html>

# GraphX

---

GraphX es la librería de algoritmos para problemas con grafos de Spark.

Los algoritmos están diseñados e implementados para ejecutarse de manera eficiente en un ambiente distribuido.



# GraphX - Algoritmos

---

PageRank

Connected components

Label propagation

SVD++

Strongly connected components

Triangle count

# GraphX - Ejemplos

---

```
from graphframes import GraphFrame
# Carga del grafo desde archivo
vertices = sc.textFile("vertices.txt")
edges = sc.textFile("edges.txt")
g = GraphFrame(vertices, edges)
print(g.inDegrees, g.outDegrees)
res = g.bfs( from, to )
pr = g.pageRank(resetProbability=0.15, tol=0.01)
cmc = g.shortestPaths(landmarks = ["a", "d"])
```

# Más ejemplos de GraphX

---

<https://spark.apache.org/docs/latest/graphx-programming-guide.html>

<https://github.com/graphframes/graphframes>