S6 INTRODUCCIÓN AL MACHINE LEARNING ESCALABLE

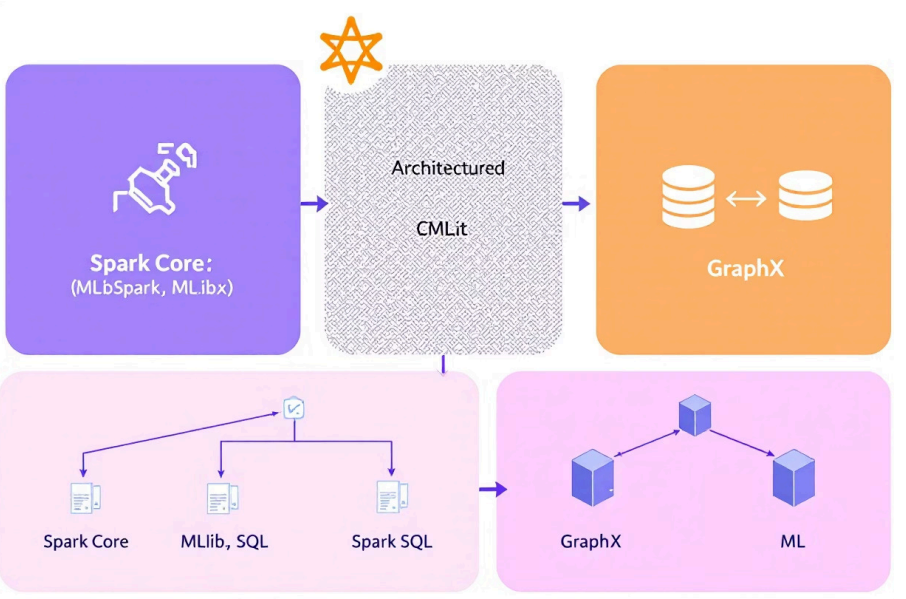
MLlib es una librería de aprendizaje automático (machine learning) de Apache Spark que proporciona una variedad de algoritmos y herramientas para facilitar la construcción y ejecución de modelos de aprendizaje automático en grandes volúmenes de datos. Está diseñada para aprovechar el procesamiento distribuido de Spark, lo que le permite escalar de manera eficiente en clústeres de computadoras.

por Kibernum Capacitación S.A.

MLLIB: Características y Herramientas

Características de MLlib

- Escalabilidad: Al estar integrada con Apache Spark, MLlib puede manejar grandes volúmenes de datos distribuidos y realizar cálculos de manera eficiente en un entorno de clústeres. Esto es ideal para tareas de big data.
- Optimización y Sintonización de Modelos: MLlib incluye herramientas para la selección de modelos y ajuste de hiperparámetros (como cross-validation y grid search) para obtener el mejor rendimiento en las tareas de machine learning.
- Facilidad de Uso: La librería está diseñada para ser fácil de usar, proporcionando una API en lenguajes como Scala, Python, Java, y R, lo que permite a los desarrolladores implementar modelos de machine learning de manera más sencilla.



Estructuras de Dato de MLlib

RDDs y DataFrames son las estructuras de datos fundamentales para trabajar con Spark en general, pero MLlib también introduce estructuras específicas como LabeledPoint, DenseVector, y SparseVector.

LabeledPoint es la estructura clave para representar los datos etiquetados en tareas de aprendizaje supervisado.

DenseVector y SparseVector son tipos especializados de vectores que se utilizan para representar las características de las observaciones de manera eficiente, dependiendo de si los datos son densos o dispersos.

Algoritmos de Machine Learning Soportados por MLlib

MLlib de Apache Spark ofrece una amplia gama de algoritmos de aprendizaje automático (machine learning) diseñados para ser escalables y eficientes, aprovechando el procesamiento distribuido de Spark. A continuación, se presenta un resumen de los algoritmos soportados por MLlib:

Algoritmos de Clasificación

- Regresión Logística: Utilizado para clasificación binaria, predice la probabilidad de pertenecer a una clase.
- Árboles de Decisión: Dividen las observaciones en grupos utilizando reglas basadas en características, y se utilizan para clasificación.
- Máquinas de Soporte Vectorial (SVM): Buscan el hiperplano que maximiza el margen entre clases, útil para datos no lineales.
- K-Vecinos más Cercanos (KNN): Clasificación basada en la similitud de los datos con sus vecinos más cercanos.

Algoritmos de Regresión

- Regresión Lineal: Modela la relación entre variables dependientes e independientes con una ecuación lineal.
- Regresión de Lasso y Ridge: Variantes de la regresión lineal que aplican regularización para evitar el sobreajuste.

Algoritmos de Clustering

- K-Means: Algoritmo de clustering que agrupa los datos en K clústeres, minimizando la variación dentro de cada grupo.
- Clustering Jerárquico: Crea una jerarquía de clústeres, que permite el agrupamiento a diferentes niveles de granularidad.

Otros Algoritmos

- PCA (Análisis de Componentes Principales): Reduce la dimensionalidad del conjunto de datos, conservando las características más importantes.
- ALS (Alternating Least Squares): Algoritmo utilizado en sistemas de recomendación, basado en la factorización de matrices para predecir las preferencias de los usuarios.

Implementación de Algoritmos de Machine Learning

Algoritmos Supervisados

Apache Spark MLlib soporta una variedad de algoritmos de aprendizaje supervisado. Los algoritmos supervisados son aquellos en los que el modelo se entrena con datos etiquetados, es decir, con ejemplos en los que tanto las características de entrada como la etiqueta de salida están disponibles.

Los algoritmos más comunes para aprendizaje supervisado incluyen:

- Regresión logística (Logistic Regression)
- Árboles de decisión (Decision Trees)
- Máquinas de soporte vectorial (SVM - Support Vector Machines)
- Regresión lineal (Linear Regression)

A continuación, vamos a implementar un ejemplo simple de regresión logística usando MLlib para clasificar datos en dos categorías. Vamos a usar un conjunto de datos sintéticos para ilustrar cómo entrenar un modelo y hacer predicciones.

Logigic Regression with Logisic Regression aith usta Apache Spark MLliib

```
data preparation  {
    which a dedi-nubic llogistions by to wmigin pand repeticd;

    data preparation

    mnodel {
       waial iof moust in logic Spark, "MLiib

       cale at fra(nuiapuction.() };;
           oubiatl patiar,(;
           gdata regrtion enciple data wedking.:(-apruet(;));
           maget in mtril-cilves, and indluction mome
           model.om data regrisne("M1"i-wipt wi".logisc regression"; );

           evaluatin evaatiion {
              epauts {;
              model = 9%,"1081""lent 11X}
              eabulatied_attrirdefils and fo": f(;(1U8"); }
           }.
```

Este ejemplo demuestra cómo usar regresión logística en MLlib para un problema de clasificación binaria. El modelo es entrenado en un conjunto de datos pequeño y luego se utiliza para hacer predicciones, evaluando su precisión con un evaluador de clasificación binaria. Esta es una forma eficiente de implementar y probar modelos de aprendizaje supervisado en un entorno distribuido utilizando Apache Spark.

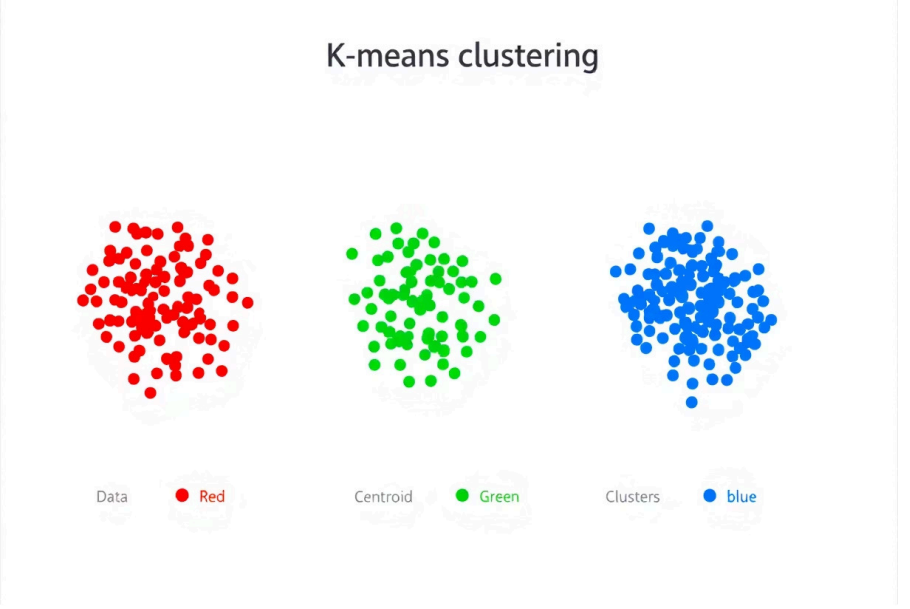
Algoritmos No Supervisados

Los algoritmos no supervisados son aquellos que no requieren etiquetas en los datos. El objetivo de estos algoritmos es encontrar patrones o estructuras subyacentes en los datos sin información previa sobre las salidas. Los algoritmos de clustering (agrupamiento) y reducción de dimensionalidad son algunos de los más comunes en aprendizaje no supervisado.

En MLlib de Apache Spark, se proporcionan varios algoritmos de machine learning no supervisado. Los más comunes incluyen:

- K-Means (clustering)
- Clustering Jerárquico (agglomerative clustering)
- PCA (Análisis de Componentes Principales) (reducción de dimensionalidad)

En este ejemplo, implementaremos el algoritmo K-Means para el clustering de datos.

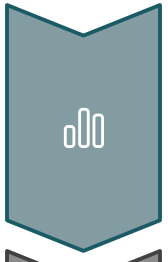


Este ejemplo ilustra cómo usar el algoritmo de K-Means de MLlib para realizar clustering en un conjunto de datos. Usamos K-Means para agrupar puntos de datos en clústeres, visualizamos los resultados y evaluamos la calidad del modelo. El uso de MLlib permite realizar estos procedimientos de forma eficiente, incluso con grandes volúmenes de datos distribuidos.

Este tipo de análisis es comúnmente utilizado en tareas de segmentación de clientes, análisis de patrones y detección de anomalías.

Actividad Práctica Guiada

OBJETIVO: En esta actividad, desarrollaremos 2 algoritmos de aprendizaje supervisado y 2 de aprendizaje no supervisado. Para cada algoritmo, elaboraremos un caso de uso práctico en el que sea aplicable y explicaremos por qué el algoritmo es adecuado para ese caso.



Regresión Lineal

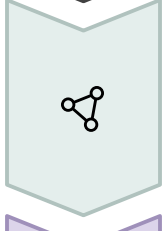
Caso: Predicción del precio de una casa basada en sus características.

Razón: Es un modelo adecuado para predecir variables continuas, como el precio de una casa, basándose en una relación lineal entre las características.

Árbol de Decisión

Caso: Clasificación de usuarios en compradores y no compradores en un sitio web.

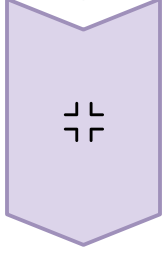
Razón: Ideal para clasificación binaria, proporcionando una visualización clara y fácil de entender de las decisiones.



K-Means (Clustering)

Caso: Segmentación de clientes en un supermercado para personalizar ofertas.

Razón: Útil para identificar grupos naturales en los datos y ayudar a personalizar estrategias de marketing.



PCA (Análisis de Componentes Principales)

Caso: Reducción de dimensionalidad para reconocimiento facial.

Razón: Permite reducir la cantidad de características de un conjunto de datos de imágenes, facilitando el procesamiento y mejora de la eficiencia del sistema.