Machine Learning y Data Science con PySpark: cero a experto

Data Bootcamp
BEST DATA TRAINING

1

Introducción a Apache Spark

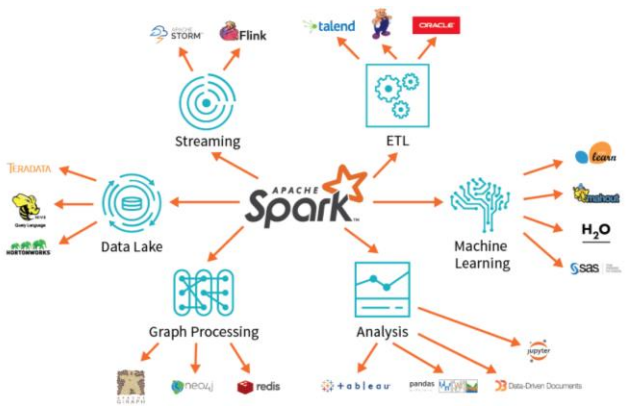
Data Bootcamp
BEST DATA TRAINING

2

Apache Spark

Spark es una solución **Big Data** de **código abierto**. Desarrollado por el laboratorio RAD de **UC Berkeley** (2009).

Se ha convertido en una **herramienta de referencia** en el campo del Big Data.



Data Bootcamp
BEST DATA TRAINING

3

Apache Spark vs MapReduce

Más fácil y rápida que Hadoop MapReduce.

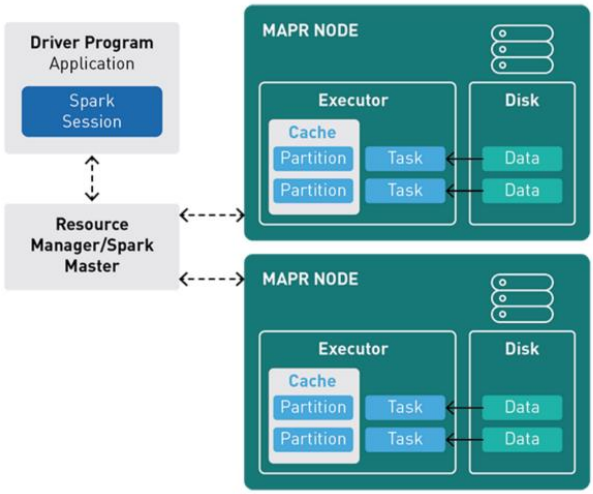
Diferencias:

- **Spark** mucho **más rápido** al almacenar en caché los datos en la **memoria** vs **MapReduce** en el **disco duro** (más lectura y escritura)
- Spark optimizado para un mejor **paralelismo**, utilización **CPU** e inicio más rápido
- Spark tiene modelo de **programación funcional** más rico
- Spark es especialmente útil para **algoritmos iterativos**

Data Bootcamp
BEST DATA TRAINING

4

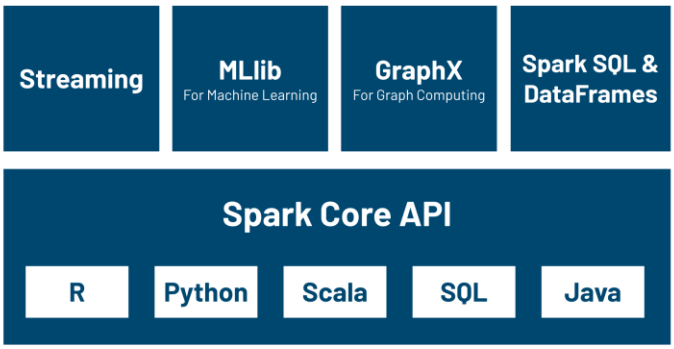
Cómo se Spark en un clúster



5

Componentes de Spark

Spark contiene un **ecosistema** de herramientas **muy completo.**



6

PySpark

PySpark es una biblioteca Spark **escrita en Python** para ejecutar la aplicación Python usando las **capacidades de Apache Spark**.

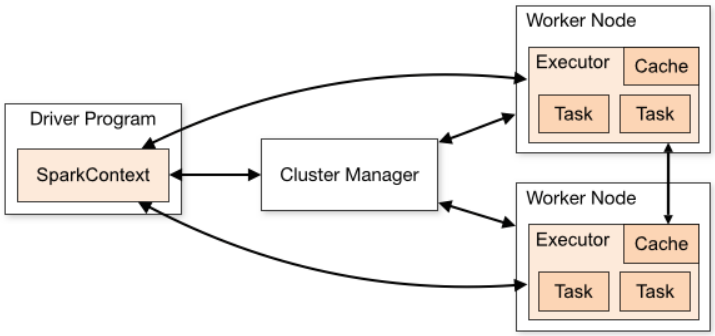
Ventajas de PySpark:

- **Fácil** de aprender
- Amplio conjunto de librerías para **ML y DS**
- Gran apoyo de la **comunidad**

7

Arquitectura de PySpark

Apache Spark funciona en una **arquitectura maestro-esclavo**. Las **operaciones** se ejecutan en los **trabajadores**, y el **Cluster Manager** administra los recursos.



8

Tipos de administradores de clústeres

Spark admite los siguientes administradores de clústeres:

- **Standalone** : administrador de clúster simple
- **Apache Mesos** : es un administrador de clústeres que puede ejecutar también Hadoop MapReduce y PySpark.
- **Hadoop YARN** : el administrador de recursos en Hadoop 2
- **Kubernetes**: para automatizar la implementación y administración de aplicaciones en contenedores.

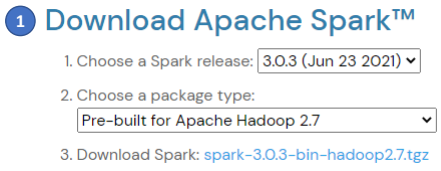
9

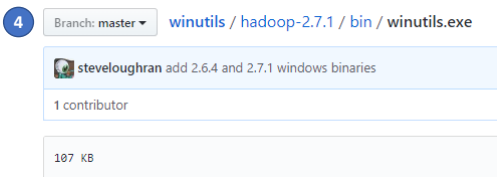
Instalación de Apache Spark

10

Pasos para instalar Spark (1)

1. Descarga **Spark** de https://spark.apache.org/downloads.html
2. Modifica el **log4j.properties.template** pon en log4j.rootCategory=**ERROR** en vez de INFO.
3. Instala **Anaconda** de https://www.anaconda.com/
4. Descarga **winutils.exe**. Es un binario de Hadoop para Windows - del repositorio de GitHub de https://github.com/steveloughran/winutils/ . Vaya a la versión de Hadoop correspondiente con la distribución de Spark y busque winutils.exe en **/bin**.



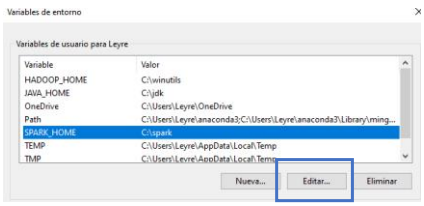


Data Bootcamp
BEST DATA TRAINING

11

Pasos para instalar Spark (2)

1. Si no tienes **Java** o la versión de Java es 7.x o menos, descargue e instale Java desde Oracle https://www.oracle.com/java/technologies/downloads/
2. Descomprime Spark en **C:\spark**
3. Añade el winutils.exe descargado a una carpeta de winutils en C:. Debe quedar así: **C:\winutils\bin\winutils.exe**.
4. Desde **cmd** ejecuta: "cd C:\winutils\bin" y después: winutils.exe chmod 777 \tmp\hive
5. Añade las variables de entorno:
    - HADOOP_HOME -> C:\winutils
    - SPARK_HOME -> C:\spark
    - JAVA_HOME -> C:\jdk
    - Path -> %SPARK_HOME%\bin
    - Path -> %JAVA_HOME%\bin



Data Bootcamp
BEST DATA TRAINING

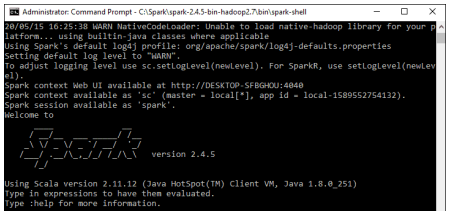
12

Validación de la instalación de Spark

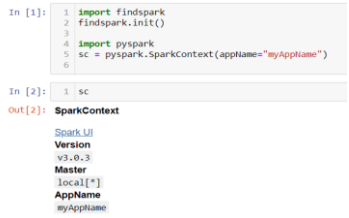
1. Desde el **prompt** de Anaconda ejecuta: “cd C:\spark” y después “pyspark”. Deberías ver algo como lo de la imagen 1.
2. Desde **jupyter notebook** instala findspark con “pip install findspark” y ejecuta el siguiente código.

```
import findspark
findspark.init()
import pyspark
sc = pyspark.SparkContext(appName="myAppName")
sc
```

1



2



Data Bootcamp
BEST DATA TRAINING

13

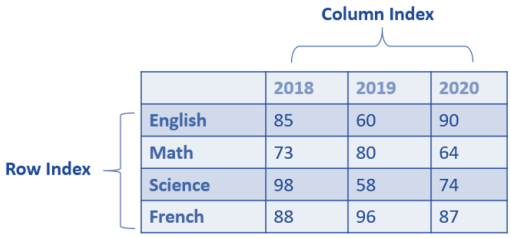
DataFrames en Apache Spark

Data Bootcamp
BEST DATA TRAINING

14

Introducción a DataFrames

Los **DataFrames** son de naturaleza **tabular**. Permiten varios formatos dentro de una misma tabla (**heterogéneos**), mientras que cada variable suele tener valores con un único formato (**homogéneos**). Similares a las tablas SQL o a las hojas de calculo.



	2018	2019	2020
English	85	60	90
Math	73	80	64
Science	98	58	74
French	88	96	87

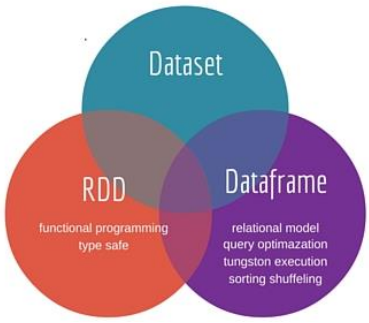
Data Bootcamp
BEST DATA TRAINING

15

Ventajas de los DataFrames

Algunas de las ventajas de trabajar con Dataframes en Spark son:

- Capacidad de procesar una **gran cantidad de datos** estructurados o semiestructurados
- Fácil **manejo de datos** e imputación de valores faltantes
- Múltiples formatos como **fuentes de datos**
- Compatibilidad con **múltiples lenguajes**

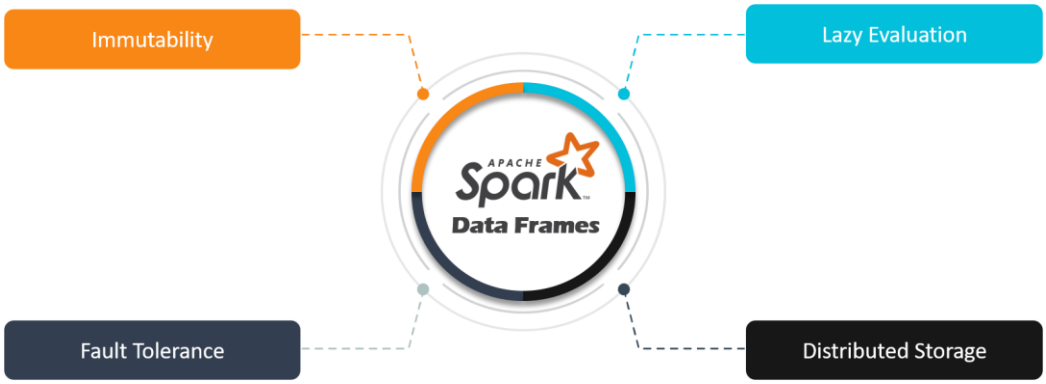


Data Bootcamp
BEST DATA TRAINING

16

Características de los DataFrames

Los **DataFrames** de Spark **se caracterizan** por: ser distribuidos, evaluación perezosa, inmutabilidad y tolerancia a fallos.

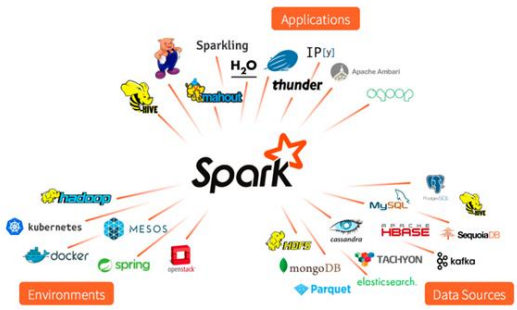


Data Bootcamp
BEST DATA TRAINING

17

Fuentes de datos de DataFrames

Los marcos de datos en Pyspark se pueden crear de varias formas: a través de archivos, utilizando RDDs o a través de bases de datos.



Data Bootcamp
BEST DATA TRAINING

18

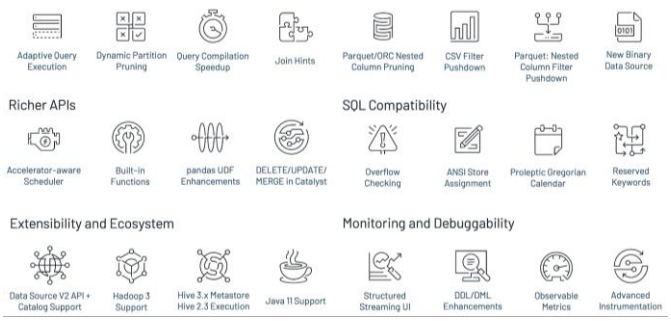
Funciones avanzadas de Spark

Data Bootcamp
BEST DATA TRAINING

19

Funciones avanzadas

Spark contiene numerosas **funciones avanzadas** para optimizar su rendimiento y realizar transformaciones complejas en los datos. Algunas de ellas son: las expresiones de selectExpr(), UDF, cache ( ), etc

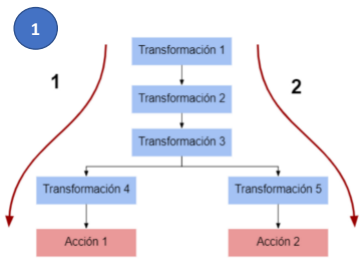


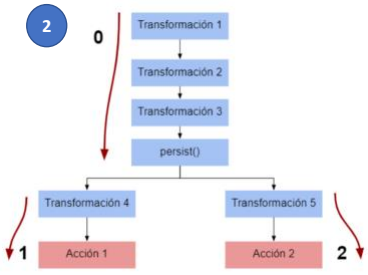
Data Bootcamp
BEST DATA TRAINING

20

Optimización del rendimiento

Una de las **técnicas de optimización** son los métodos **cache**() y **persist**(). Estos métodos se usan para **almacenar un calculo intermedio** de un RDD, DataFrame y Dataset para que puedan reutilizarse en acciones posteriores.





Data Bootcamp
BEST DATA TRAINING

21

Machine Learning con Spark

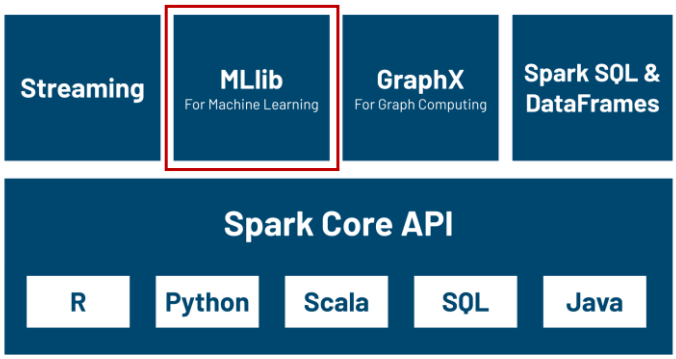
Data Bootcamp
BEST DATA TRAINING

22

Spark Machine Learning

Machine Learning: es la construcción de **algoritmos** que pueden aprender de los datos y hacer predicciones sobre ellos.

Spark MLlib se usa para realizar aprendizaje automático en Apache Spark. MLlib consta de algoritmos y funciones habituales.



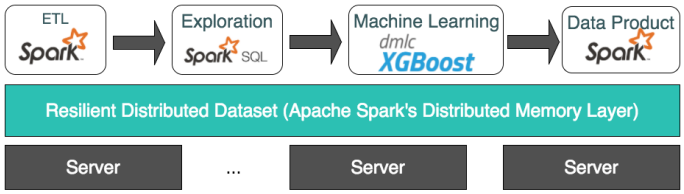
Data Bootcamp
BEST DATA TRAINING

23

Herramientas Spark Machine Learning

Herramientas de MLlib:

- **spark.mllib** contiene la API original construida sobre RDD
- **spark.ml** proporciona una API de nivel superior construida sobre DataFrames para construcción de pipelines de ML. La API de ML principal.



Fuente: https://www.r-bloggers.com/

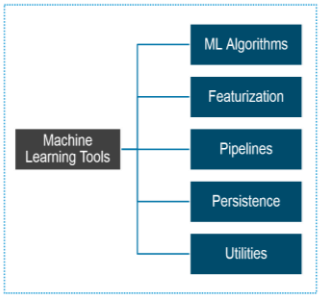
Data Bootcamp
BEST DATA TRAINING

24

Componentes Spark Machine Learning

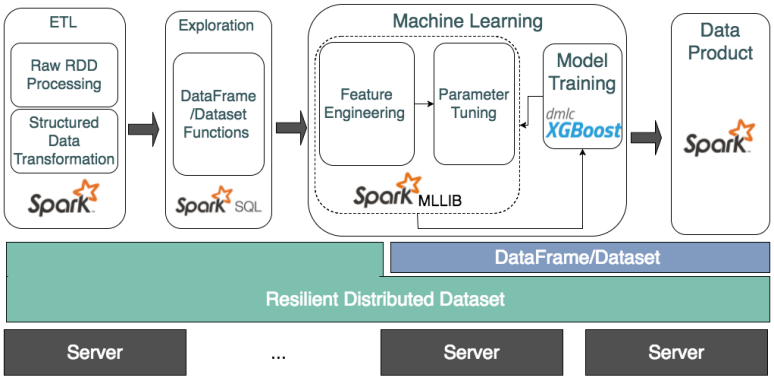
Spark MLlib proporciona las siguientes herramientas:

- **Algoritmos ML**: Incluyen algoritmos de aprendizaje comunes como clasificación, regresión, agrupamiento y filtrado colaborativo.
- **Caracterización**: Incluye: extracción, transformación, reducción de dimensionalidad y selección de características.
- **Pipelines**: son herramientas para construir modelos de ML en etapas.
- **Persistencia**: permite guardar y cargar algoritmos, modelos y pipelines.
- **Utilidades**: para álgebra lineal, estadística y manejo de datos.



25

Proceso de Machine Learning



Fuente: https://www.r-bloggers.com/

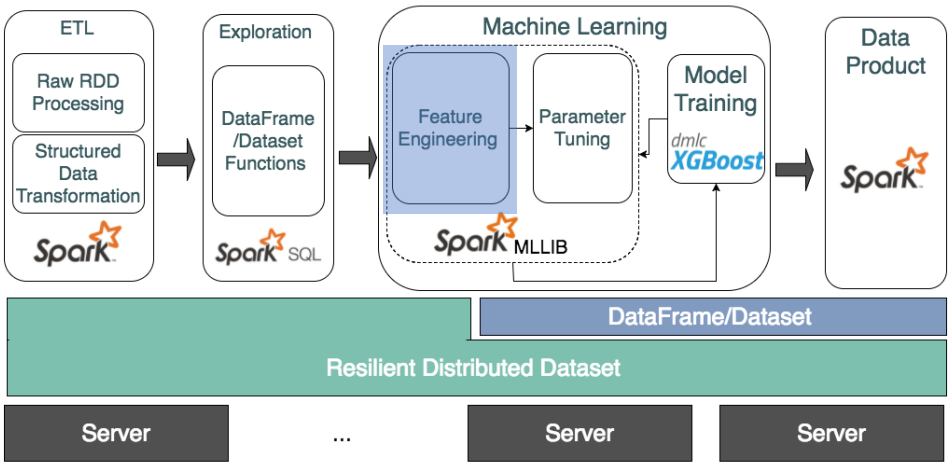
26

Ingeniería de características

Data Bootcamp
BEST DATA TRAINING

27

70% tiempo



Fuente: https://www.r-bloggers.com/

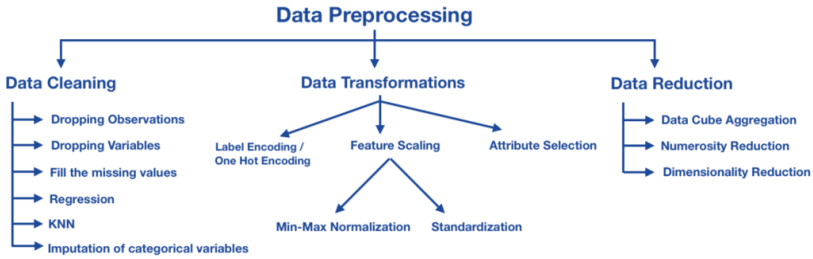
Data Bootcamp
BEST DATA TRAINING

28

Ingeniería de características con Spark

Las técnicas de preprocesamiento de datos más utilizadas en los enfoques de Spark son las siguientes

- VectorAssembler
- Agrupamiento
- Escalado y normalización
- Trabajar con características categóricas
- Transformadores de datos de texto
- Manipulación de funciones
- PCA



Data Bootcamp
BEST DATA TRAINING

29

Ingeniería de características con Spark

- **Vector Asembler:** Se utiliza básicamente para concatenar todas las características en un solo vector que se puede pasar al estimador o al algoritmo ML
- **Agrupamiento:** es el método más sencillo para convertir las variables continuas en variables categóricas. Se puede realizar con la clase Bucketizer.
- **Escalado y normalización:** es otra tarea común en variables continuas. Permite que los datos tengan una distribución normal.
- **MinMaxScaler y StandardScaler:** estandarizan las características con una media cero y una desviación estándar de 1.
- **StringIndexer :** para convertir características categóricas en numéricas.

```
+ ---- + ---- + ---- + ------------- +
| int1 | int2 | int3 | características |
+ ---- + ---- + ---- + ------------- +
| 7 | 8 | 9 | [7.0,8.0,9.0] |
| 1 | 2 | 3 | [1.0,2.0,3.0] |
| 4 | 5 | 6 | [4.0,5.0,6.0] |
+ ---- + ---- + ---- + ------------- +
```

Data Bootcamp
BEST DATA TRAINING

30

StringIndexer vs OneHot Encoder

No podemos aplicar OneHotEncoder a variables de texto directamente. Primero debemos convertir las columnas de texto en numericas. Para eso usaremos StringIndexer. Después de eso, podemos aplicar OneHotEncoder.

StringIndexer

id	name	qualification	age	gender	qualificationIndex
1	John	B.A.	20	Male	0.0
2	Martha	B.Com.	20	Female	1.0
3	Mona	B.Com.	21	Female	1.0
4	Harish	B.Sc.	22	Male	2.0
5	Jonny	B.A.	22	Male	0.0
6	Maria	B.A.	23	Female	0.0
7	Monalisa	B.A.	21	Female	0.0

OneHot Encoder

id	name	qualification	age	gender	qualificationIndex	qualification_vec
1	John	B.A.	20	Male	0.0	(2,[0],[1.0])
2	Martha	B.Com.	20	Female	1.0	(2,[1],[1.0])
3	Mona	B.Com.	21	Female	1.0	(2,[1],[1.0])
4	Harish	B.Sc.	22	Male	2.0	(2,[],[])
5	Jonny	B.A.	22	Male	0.0	(2,[0],[1.0])
6	Maria	B.A.	23	Female	0.0	(2,[0],[1.0])
7	Monalisa	B.A.	21	Female	0.0	(2,[0],[1.0])

Data Bootcamp
BEST DATA TRAINING

31

VarianceThresholdSelector

VarianceThresholdSelectores un selector que elimina las **características de baja variación**.

Ejemplo: La varianza para las 6 características son 16.67, **0.67**, 8.17, 10.17, **5.07** y 11.47. Si varianceThreshold = 8.0, entonces se eliminan las características con varianza <= 8.0:

id	features	selectedFeatures
1	[6.0, 7.0, 0.0, 7.0, 6.0, 0.0]	[6.0,0.0,7.0,0.0]
2	[0.0, 9.0, 6.0, 0.0, 5.0, 9.0]	[0.0,6.0,0.0,9.0]
3	[0.0, 9.0, 3.0, 0.0, 5.0, 5.0]	[0.0,3.0,0.0,5.0]
4	[0.0, 9.0, 8.0, 5.0, 6.0, 4.0]	[0.0,8.0,5.0,4.0]
5	[8.0, 9.0, 6.0, 5.0, 4.0, 4.0]	[8.0,6.0,5.0,4.0]
6	[8.0, 9.0, 6.0, 0.0, 0.0, 0.0]	[8.0,6.0,0.0,0.0]

Data Bootcamp
BEST DATA TRAINING

32

UnivariateFeatureSelector

Opera con características categóricas / continuas. El usuario puede establecer featureTypey labelType, y Spark seleccionará la función de puntuación que se utilizará según el featureTypey especificado labelType..

featureType	labelType	score function
categorical	categorical	chi-squared (chi2)
continuous	categorical	ANOVATest (f_classif)
continuous	continuous	F-value (f_regression)

id	features	label	selectedFeatures
1	[1.7, 4.4, 7.6, 5.8, 9.6, 2.3]	3.0	[2.3]
2	[8.8, 7.3, 5.7, 7.3, 2.2, 4.1]	2.0	[4.1]
3	[1.2, 9.5, 2.5, 3.1, 8.7, 2.5]	3.0	[2.5]
4	[3.7, 9.2, 6.1, 4.1, 7.5, 3.8]	2.0	[3.8]
5	[8.9, 5.2, 7.8, 8.3, 5.2, 3.0]	4.0	[3.0]
6	[7.9, 8.5, 9.2, 4.0, 9.4, 2.1]	4.0	[2.1]

33

Modelos de Clasificación de PySpark

34

Regresión logística

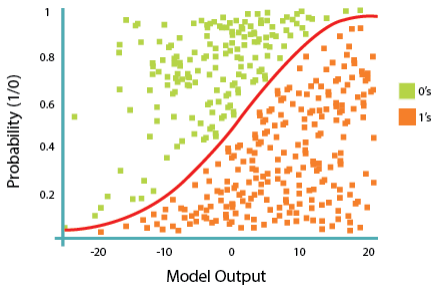
La regresión logística es un método popular para predecir una respuesta categórica. Es un caso especial de modelos lineales generalizados que predice la probabilidad de los resultados.

Ventajas

- Fácil de entender y explicar
- Rara vez existe sobreajuste
- Rápido para entrenar.
- Fácil de entrenar con grandes cantidades de datos

Desventajas

- Difícilmente ajusta con datos no lineales.
- Sensible a valores atípicos.



Árbol de decisión

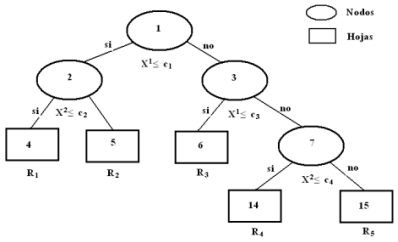
Es uno de los algoritmos de aprendizaje automático más antiguos y sencillos. Son populares debido a su fácil **interpretación visual**. Sin embargo, en la práctica, tienden a **sobreajustar** los datos de entrenamiento

Ventajas

- Fáciles de interpretar y visualizar.
- Puede capturar fácilmente patrones no lineales.
- Requiere menos preprocesamiento de datos

Desventajas

- Sensibles al ruido
- Una pequeña variación en los datos puede dar lugar a un árbol de decisión diferente.
- Con datos desbalanceados quedan sesgados

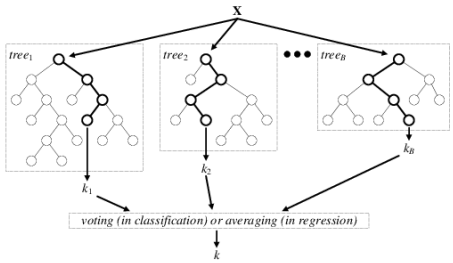


Random Forest

Estos combinan muchos (cientos o miles) de árboles de decisión. Los random forest reducen en gran medida la posibilidad de sobreajuste.

Ventajas
- Capacidad de trabajar con grandes cantidades de datos
- Capacidad de trabajar con datos faltantes

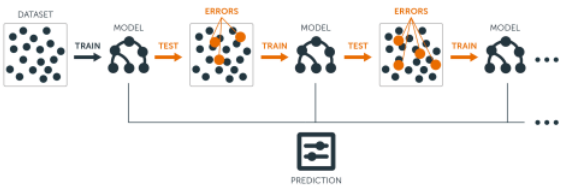
Desventajas
- Caja negra



37

Gradient Boosting

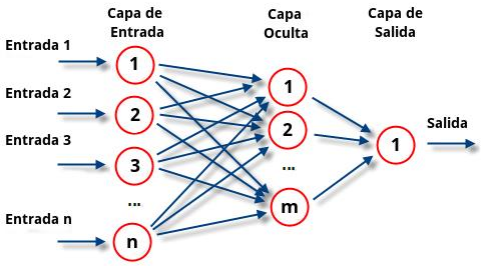
Un modelo Gradient Boosting está formado por un conjunto de **árboles de decisión individuales**, entrenados de forma secuencial, de forma que **cada nuevo** árbol trata de **mejorar los errores de los árboles anteriores**. La predicción se obtiene agregando las predicciones de todos los árboles individuales.



38

Perceptrón multicapa

El clasificador de perceptrones multicapa (MLPC) es un clasificador basado en la **red neuronal artificial feedforward** . MLPC consta de múltiples capas de nodos. Cada capa está **completamente conectada** a la siguiente capa de la red.



39

Suport Vector Machine

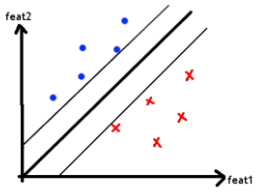
Las Máquinas Vectores de Soporte clasificación (SVM) construye un hiperplano en un espacio multidimensional para separar las diferentes clases. Trata de encontrar un hiperplano marginal máximo que mejor divida el conjunto de datos en clases.

Ventajas:

- Buena precisión y menor uso de memoria
- Funcionan bien con un espacio de dimensiones elevados

Desventajas

- Requieren mucho tiempo de entrenamiento

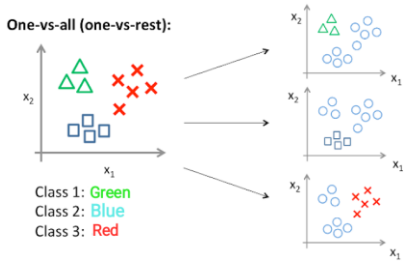


40

One-vs-Rest classifier

Los algoritmos como Perceptrón, Regresión logística y LSVC se diseñaron para la clasificación binaria y **no admiten** de forma nativa **tareas de clasificación con más de dos clases**.

Para usar algoritmos de clasificación binaria en **clasificación múltiple** es dividir el conjunto de datos en múltiples conjuntos de clasificación binaria y ajustar un modelo de clasificación binaria en cada uno, estrategia **One-vs-Rest classifier**



41

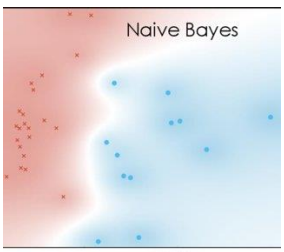
Naive Bayes

Basado en el Teorema de Bayes con una suposición de independencia entre los predictores. Naive Bayes es fácil de construir y particularmente útil para conjuntos de datos muy grandes.

Ventajas:

- Es fácil y rápido predecir la clase
- Funciona bien en la predicción multiclase

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$



42

Modelos de Regresión de PySpark

Data Bootcamp
BEST DATA TRAINING

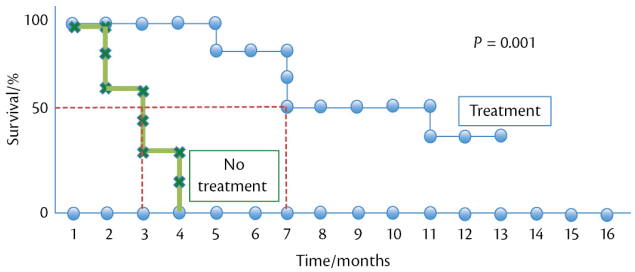
43

Análisis de supervivencia

El análisis de supervivencia (regresión) modela el **tiempo hasta un evento**. El análisis de supervivencia es un tipo especial de regresión y se caracteriza por:

- La **etiqueta** es **siempre positiva**
- Es posible que la **etiqueta no se conozca** por completo **o no esté**

Una de las primeras es modelar la mortalidad de una población determinada



Data Bootcamp
BEST DATA TRAINING

44

Machine Learning avanzado

Data Bootcamp
BEST DATA TRAINING

45

Análisis exploratorio de los datos (EDA)

El análisis de datos podemos hacerlo generando estadísticos, gráficos, etc.

Para **la visualización** mediante **gráficos**, PySpark es compatible con numerosas **librerías de visualización** de datos de Python como seaborn, matplotlib, bokehn, etc.

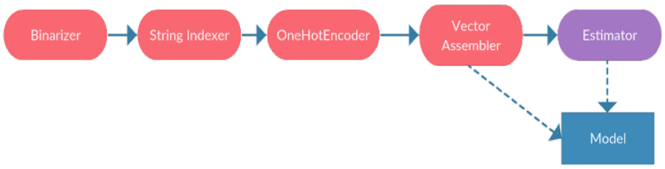
Solo si no tenemos demasiados datos o podemos agruparlos -> Pandas-Profiling puede ser útil. Realiza un análisis de la distribución de los datos, correlaciones, valores faltantes, etc.

Data Bootcamp
BEST DATA TRAINING

46

Pipelines en PySpark

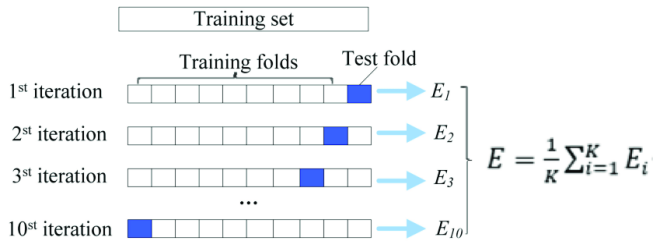
En los **Pipelines** (canalizaciones) las diferentes **etapas del trabajo** de aprendizaje automático se pueden agrupar como una sola entidad y se pueden considerar como un flujo de trabajo ininterrumpido.
Cada etapa es un **Transformador** . Se ejecutan en **secuencia** y los datos de entrada se transforman mientras pasan por cada etapa.



47

Validación cruzada

Divide el conjunto de datos en **folds** o particiones que se utilizan como conjuntos de datos de prueba y entrenamiento separados. Después calcula el **promedio de la métrica de evaluación** para todas las iteraciones, ajustando los conjuntos de datos utilizados (entrenamiento, prueba)

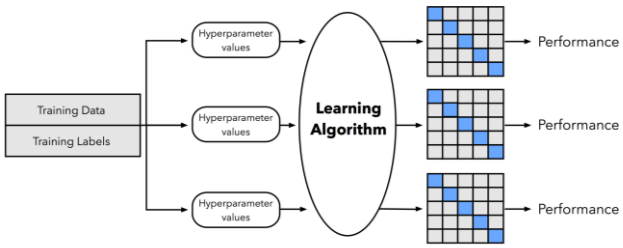


48

Ajuste de hiperparámetros

El ajuste de hiperparametros funciona de la siguiente manera:

- Para cada par de datos (entrenamiento, prueba), iteran a través del conjunto de **ParamMaps** (matriz de hiperparametros)
- Para cada ParamMap, se **ajusta el modelo** usando esos parámetros y **evalúan el desempeño**
- Seleccionan el conjunto de **parámetros** que producen un **mejor rendimiento**.



Data Bootcamp
BEST DATA TRAINING

49

Spark Streaming

Data Bootcamp
BEST DATA TRAINING

50

Fundamentos Spark Streaming

PySpark Streaming es un sistema escalable y tolerante a fallos que sigue el paradigma de **lotes RDD**.

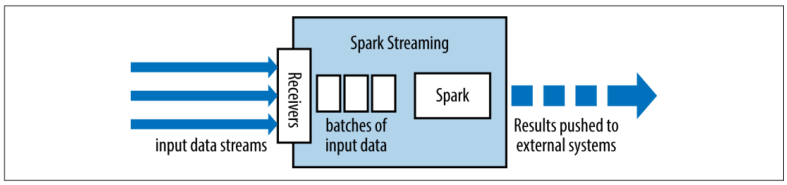
Opera en intervalos de lotes, recibiendo un **flujo de datos de entrada continuo** de fuentes como Apache Flume , Kinesis, Kafka, sockets TCP, etc.

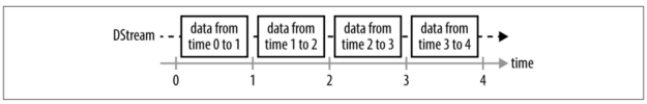
Spark Engine se encarga de procesarlos.

51

Funcionamiento Spark Streaming

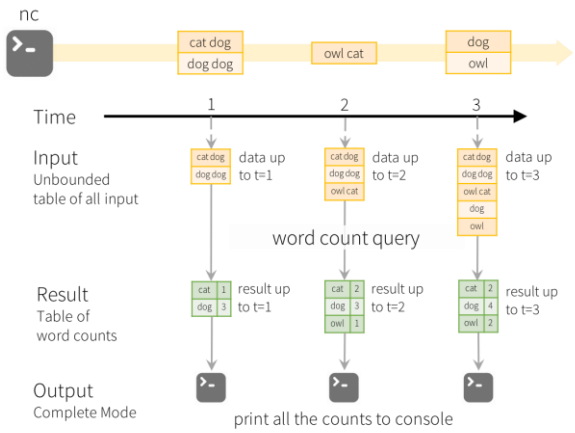
Spark Streaming recibe datos de varias fuentes y los agrupa en pequeños lotes (**Dstreams**) en un intervalo de tiempo. El usuario puede definir el **intervalo**. Cada lote de entrada forma un RDD y se procesa mediante trabajos de Spark para crear otros RDD.





52

Ejemplo: contar palabras



Modos de salida

Spark usa varios modos de salida para almacenar los datos:

- **Modo completo** (*Complete*): toda la tabla se almacenará
- **Modo de adición** (*Append*): solo las nuevas filas del último proceso se alamcenará. Solo para las consultas en las que no se espera que cambien las filas existentes.
- **Modo de actualización** (*Update*): solo las filas que se actualizaron desde el último proceso se almacenarán. Este modo solo genera las filas que han cambiado desde el último proceso. Si la consulta no contiene agregaciones, será equivalente al modo append.

Complete, Append, Update

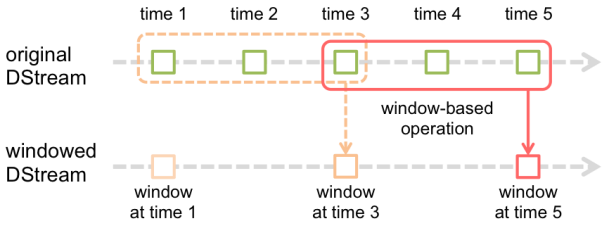
```
query = wordCounts \
    .writeStream \
    .outputMode("complete") \
    .format("console") \
    .start()
```

Tipos de transformaciones

Para **tolerancia a fallos** los datos recibidos se copian en dos nodos y hay también un mecanismo llamado **checkpointing**.

Las transformaciones se pueden agrupar en :

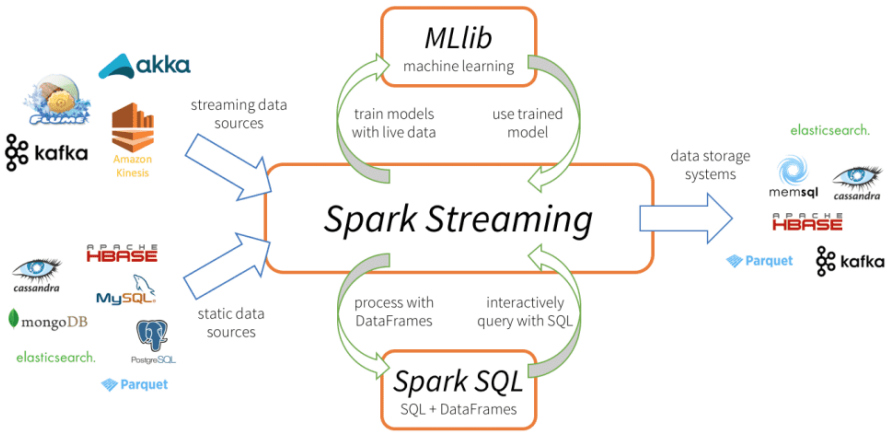
- **sin estado**: no depende de los datos de lotes anteriores.
- **con estado**: utilizan datos de lotes anteriores



Data Bootcamp
BEST DATA TRAINING

55

Capacidades de Spark Streaming



Data Bootcamp
BEST DATA TRAINING

56

Recursos

Data Bootcamp
BEST DATA TRAINING

57

Recursos:

- **https://spark.apache.org/docs/2.2.0/index.html** Documentación oficial de Spark
- **https://colab.research.google.com/** Google Colab para poder tener capacidad de computo adicional

Data Bootcamp
BEST DATA TRAINING

58