APACHE SPARK
SESIÓN 3

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL
GENERALITAT VALENCIANA
Conselleria d'Educació, Investigació, Cultura i Esport
cefire
UNIÓ EUROPEA
Fons Social Europeu
L'FSE inverteix en el teu futur

¿QUÉ VEREMOS EN ESTA UNIDAD?

Repaso sesión anterior

Spark SQL

Introducción a MLLIB

REPASO SESIÓN ANTERIOR

RDD CLAVE_VALOR

PERSISTENCIA

- Cache()
- Persist()

PARTICIONADO

REPASO SESIÓN ANTERIOR

CORRECCIÓN DE EJERCICIOS

SPARK SQL

- Datos estructurados
- DataFrame
- Filas y columnas
- Mejor rendimiento que RDD

SPARK SQL

SPARKSESION

Inicialización:

```
from spark.sql import SparkSession

spark = SparkSesion.builder.appName('dataframe').getOrCreate()
```

SPARK SQL

SparkSession: es el objeto principal o la base a partir de la cual cuelga toda la funcionalidad de Apache Spark.

Es similar al **SparkContext** de los RDD, pero para trabajar con SparkSQL, los DataFrame y DataSet.

DataFrame es un DataSet organizado en columnas.

SPARK SQL

¿Que es un dataset?

Un DataSet es una colección de datos distribuidos que tienen ya una estructura, a diferencia de los RDD, que son conjuntos de datos desestructurados.

Sus características y ventajas principales son:

- Aparecen a partir de la versión de Spark 1.6.
- Poseen los beneficios de los RDD.
- Nos proporciona una API tanto para trabajar con Java como con Scala.
- No nos proporciona API para Python porque Python pasa directamente a los DataFrame.

SPARK SQL

¿Que es un dataframe?

Son conjuntos de datos organizados en estructuras rectangulares en forma de tabla o matriz, que almacenan sus datos en filas y columnas

¿Diferencia entre dataset y dataframe?

Un dataframe es un dataset que a la vez está organizado en columnas, de modo que en el dataframe tendremos los datos estructurados y cada columna con su nombre correspondiente.

SPARK SQL

Primeros pasos en la práctica

- Inicializar SparkSession
- read()
- printSchema()
- columns
- describe()
- show()

SPARK SQL

SCHEMA

from pyspark.sql.types import

- StructType
- StringType
- IntegerType
- ...

SPARK SQL

Selección de datos

- Select()
- head()

Modificación dataframe

- withColumn()
- withColumnRenamed()

EJERCICIO PROPUESTO

A partir del Dataframe que hemos utilizado en los ejercicios anteriores

a)Crea una nueva columna que indique si la persona es mayor de 30

b)¿Sabrías cómo borrar la columna Apellidos y Edad_2?

c)Podrías crear una columna con el año de nacimiento de cada persona?

(Puedes crear más columnas de apoyo)

d) Añade un id incremental para cada fila y haz que al hacer un show se vea en primer lugar

SPARK SQL

Filtrado de elementos

- Filter()
- Where()

Filtrar Valores nulos: **IS NULL - NOT IS NULL**

EJERCICIO PROPUESTO

Ejercicio propuesto

Para practicar los filtros vamos a:

Partiendo del mismo excel de nombres

a) Saca el nombre con su total de mujeres de Sevilla

b) ¿Qué nombre se repite 7.957 veces ?

c) ¿Qué nombres se repiten menos de 4 veces por cada 100?

d) ¿Tenemos alguna persona con sus datos sin informar?

SPARK SQL

Agrupaciones

GroupBy()

Funciones de agregación

- count() - Devuelve el recuento de filas de cada grupo.
- mean() - Devuelve la media de los valores de cada grupo.
- max() - Devuelve el máximo de valores para cada grupo.
- min() - Devuelve el mínimo de valores para cada grupo.
- sum() - Devuelve el total de los valores de cada grupo.
- avg() - Devuelve el promedio de los valores de cada grupo.
- agg() -Usando la función agg () , podemos calcular más de un agregado a la vez.

EJERCICIO PROPUESTO

Ejercicio propuesto

a) ¿Cuántos Joses hay en sevilla y Alicante juntos?

b) El total de mujeres en Sevilla

c) ¿Sabrías crear una columna llamada 'Total de Joses', donde muestre el total de nombres que contienen "Jose"? Por ejemplo Jose antonio, Jose Maria, Josefa...

nota: al igual que en sql e, spark sql también existe la función like

SPARK SQL

Otras acciones

from pyspark.sql.functions import [.....]

Alias()

CountDistinct()

Stdev()

formatNumber()

OrderBy() .desc() .asc()

...

SPARK SQL

Trabajando con nulos

pyspark.sql.DataFrameNaFunctions

- na.drop()
  - thresh
  - how
    - any
    - all
  - subset
- Fill()

EJERCICIO PROPUESTO

Ejercicio propuesto

Dado el csv de ventas, cambia los nulos por:

a) Nombre nulo -> ‘Empleado’

b) Venta nula -> La media de las ventas de los compañeros(redondeado a entero)

c) Euros nulos -> El valor del compañero que menos € ha ganado

d) Ciudad nula -> ‘En C.V’

e) Identificador nulo -> ‘ABC’

SPARK SQL

Trabajando con fechas

librería -> pyspark.sql.functions import

- dayofmonth
- hour
- dayofyear
- month
- year
- ....

SPARK SQL

DATE FUNCTION SIGNATURE	DATE FUNCTION DESCRIPTION
current_date () : Column	Returns the current date as a date column.
date_format(dateExpr: Column, format: String): Column	Converts a date/timestamp/string to a value of string in the format specified by the date format given by the second argument.
to_date(e: Column): Column	Converts the column into `DateType` by casting rules to `DateType`.
to_date(e: Column, fmt: String): Column	Converts the column into a `DateType` with a specified format
add_months(startDate: Column, numMonths: Int): Column	Returns the date that is `numMonths` after `startDate`.
date_add(start: Column, days: Int): Column date_sub(start: Column, days: Int): Column	Returns the date that is `days` days after `start`
datediff(end: Column, start: Column): Column	Returns the number of days from `start` to `end`.
months_between(end: Column, start: Column): Column	Returns number of months between dates `start` and `end`. A whole number is returned if both inputs have the same day of month or both are the last day of their respective months. Otherwise, the difference is calculated assuming 31 days per month.

SPARK SQL

next_day(date: Column, dayOfWeek: String): Column	Returns the first date which is later than the value of the `date` column that is on the specified day of the week. For example, `next_day('2015-07-27', "Sunday")` returns 2015-08-02 because that is the first Sunday after 2015-07-27.
trunc(date: Column, format: String): Column	Returns date truncated to the unit specified by the format. For example, `trunc("2018-11-19 12:01:19", "year")` returns 2018-01-01 format: 'year', 'yyyy', 'yy' to truncate by year, 'month', 'mon', 'mm' to truncate by month
date_trunc(format: String, timestamp: Column): Column	Returns timestamp truncated to the unit specified by the format. For example, `date_trunc("year", "2018-11-19 12:01:19")` returns 2018-01-01 00:00:00 format: 'year', 'yyyy', 'yy' to truncate by year, 'month', 'mon', 'mm' to truncate by month, 'day', 'dd' to truncate by day, Other options are: 'second', 'minute', 'hour', 'week', 'month', 'quarter'
year(e: Column): Column	Extracts the year as an integer from a given date/timestamp/string
quarter(e: Column): Column	Extracts the quarter as an integer from a given date/timestamp/string.
month(e: Column): Column	Extracts the month as an integer from a given date/timestamp/string

EJERCICIO PROPUESTO

Ejercicio propuesto

A partir del excel de compraventas, crea un nuevo dataframe y calcula:

a) La media de Venta y compra por año (dos decimales)

b) Calcula una nueva columna con la diferencia de la media anual de compra y venta(dos decimales)

c) Calcula del 2011 la suma de ventas por semana

d) ¿Que semana se ha vendido más?

SPARK SQL

Funciones UDF:

Las UDF de PySpark son similares a las UDF en las bases de datos tradicionales. En PySpark, podemos crear una función con una sintaxis de Python y convertirla como udf,esta función la usaremos en DataFrame.

Necesitamos:

from pyspark.sql.functions import udf

EJERCICIO PROPUESTO

Crea un dataframe a partir de estos datos:

ID NOMBRE

1 jorge lopez

2 belen sanchez

3 jose ruiz

4 maria garcia

5 joan bernabeu

Añade otra columna ‘Apellido’ copiando la de ‘Nombre’ y consigue mediante UDF que en la columna ’Nombre’ solo salga el nombre y la ‘Apellido’ solo el apellido, ambos con la inicial en mayúscula

SPARK SQL

Join String	Equivalent SQL Join
inner	INNER JOIN
outer, full, fullouter, full_outer	FULL OUTER JOIN
left, leftouter, left_outer	LEFT JOIN
right, rightouter, right_outer	RIGHT JOIN
cross	
anti, leftanti, left_anti	
semi, leftsemi, left_semi	

SPARK SQL

Formato de un join:

tabla1.join(tabla2,clave1 = clave 2,tipo join).show()

SPARK SQL

TIPOS DE JOIN

INNER: Esto une dos conjuntos de datos en columnas segun una clave

OUTER,FULL,FULLOUTER: devuelve todas las filas de ambos conjuntos de datos, donde la expresión de unión no coincide, devuelve nulo en las columnas de registro respectivas.

LEFT, LEFTOUTER devuelve todas las filas del conjunto de datos izquierdo independientemente de la coincidencia encontrada en el conjunto de datos derecho

RIGHT RIGHTOUTER : opuesto al left

OTROS TIPOS DE JOIN

LEFTSEMI: devuelve todas las columnas del conjunto de datos izquierdo e ignora todas las columnas del conjunto de datos derecho

LEFTANTI: Es el anticruce, devuelve todos los registros de la izquiera que NO crucen con los de la derecha

SPARK SQL

JOIN MULTIPLE

Podemos unir muchos dataframes concatenando joins:

```
df1.join(df2,df1.id1 == df2.id2,"inner") .join(df3,df1.id1 == df3.id3,"inner")
```

SPARK SQL

VISTAS

Crear vistas:

- spark.sql()
- createOrRreplaceTempView()

SPARK SQL

VISTAS

- Facilitan las consultas
- Vistas Temporales y globales
- Lenguaje SQL

SPARK SQL

VISTAS TEMPORALES

- Disponible en la sesion que la creamos
- Se le asigna un nombre a la vista

df.createOrRreplaceTempView(‘Nombre’)

SPARK SQL

VISTAS GLOBALES

- Disponible en todas las sesiones
- Se le asigna un nombre
- Se consulta añadiendo global_temp antes del nombre

df.createGlobalTempView(‘nombre’)

select * from global_temp.nombre

Práctica Repaso

Crea los dataframes siguientes: df: Empleados

emp_id	nombre	superior_emp_id	fecha_inicio	emp_dept_id	genero	sueldo
1	Jaime	-1	2018	10	M	1300
2	Rosa	1	2010	20	F	1400
3	Maria	1	2010	10	F	1000
4	Vanesa	2	null	10	F	1800
5	Leonardo	2	2010	40	M	1300
6	Paula	2	2013	20	F	1600
7	Jose Luis	2	2014	30	M	1200

Práctica Repaso

df: departamentos

dept_nombre	dept_id
Financiero	10
Marketing	20
Ventas	30
Desarrollo	40

Práctica Repaso

Calcula:

1. Crea los dataframes necesarios con la información anterior
2. Cuál es el sueldo medio de cada departamento?
3. Elimina las filas que tengan el campo fecha_inicio a nulo
4. A los empleados que tienen más de 10 años de antigüedad se les va a subir el sueldo al doble calcula con una función UDF el nuevo sueldo de los empleados y añadelo a una nueva columna Sueldo_nuevo
5. De los trabajadores cuya responsable es Maria ¿Quien es la persona que más cobra?
6. Crea una vista que se pueda acceder a ella desde otra sesion que muestre el total que se gasta en sueldos por departamentos
7. Se acaba de contratar una jefa de ventas, llamada Elena Martinez, va a cobrar 1700€, necesitamos añadirla a nuestra base de datos

SPARK SQL

REPASO SPARK SQL

1. Spark Sesion
2. Crear dataframes
3. filtros
4. agrupaciones
5. joins
6. UDF
7. Fechas
8. Nulos
9. ...

SPARK MLLIB

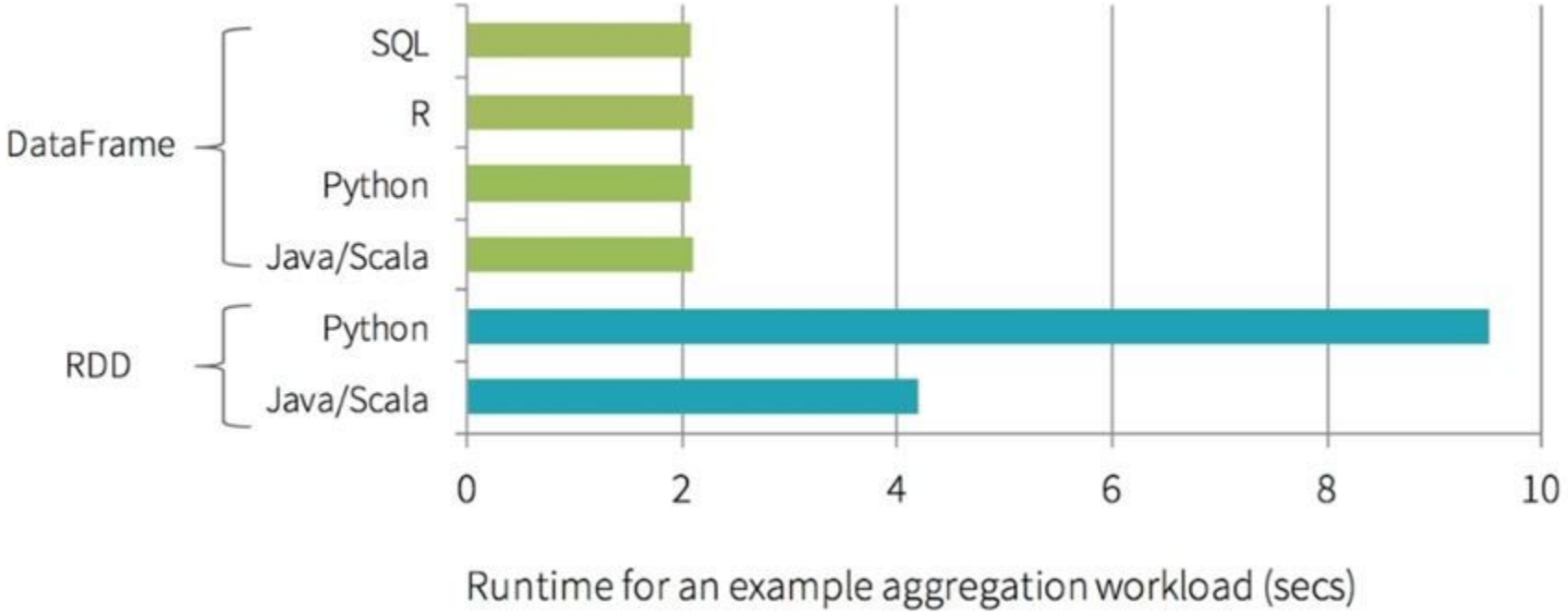
http://spark.apache.org/docs/latest/ml-guide.html

SPARK MLLIB

MLLIB NOS OFRECE DOS APIS

- La API principal o Spark ML, basada en DataFrames y que es esta dentro del paquete ml
- Y la API original o Spark MLlib, que hace uso de RDDs y esta dentro del paquete mllib. Actualmente se encuentra en modo mantenimiento y muy probable que deje de existir en Spark 3.0

SPARK MLLIB


DataFrame
SQL
R
Python
Java/Scala
RDD
Python
Java/Scala
0
2
4
6
8
10
Runtime for an example aggregation workload (secs)

SPARK MLLIB

Spark MLlib dispone de los siguientes herramientas:

- Algoritmos de ML: son el core de MLlib. Están incluidos los algoritmos de machine learning más comunes: Clasificadores, Regresores, Clustering y Filtros Colaborativos.
- Featurization: herramienta para extraer, seleccionar y transformar variables.
- Pipelines: otra herramienta para construir, evaluar y optimizar las pipelines de ML.
- Persistence: para guardar y cargar algoritmos, modelos y pipelines.
- Utilities: utilidades de álgebra lineal, estadística y manejo de datos.

SPARK MLLIB

DESTACAMOS de MLLIB

- Estadística Básica
- Algoritmos de Clasificación y Regresión
- Sistemas de Recomendación
- Clustering
- Gestión de Features
- Optimización

SPARK MLLIB

Estadística Básica

- Cálculo de estadísticos descriptivos
- Se puede ver el grado de correlación existente entre las variables.
- Muestreo estratificado
- Contraste de hipótesis.
- Generación de datos aleatorios siguiendo una determinada distribución, Normal o Poisson...

SPARK MLLIB

Algoritmos de Clasificación y Regresión

- **Clasificación**. La variable a predecir es una categoría o clase (valor categórico o discreto),
  Por ejemplo: rubio/moreno/pelirrojo, fuga/no-fuga de cliente, software malicioso/no-malicioso, etc.
  En estos, el modelo da como resultado un valor de probabilidad para cada una de las categorías disponibles (distribución de probabilidad).

- **Regresión**. La variable a predecir es numérica (normalmente continua pero no es estrictamente necesario)
  Por ejemplo: edad, altura, valor de una casa, número de Gb que va a gastar un cliente, etc.

SPARK MLLIB

Algoritmos de Clasificación y Regresión

Problem Type	Supported Methods
Binary Classification	linear SVMs, logistic regression, decision trees, random forests, gradient-boosted trees, naive Bayes
Multiclass Classification	logistic regression, decision trees, random forests, naive Bayes
Regression	linear least squares, Lasso, ridge regression, decision trees, random forests, gradient-boosted trees, isotonic regression

SPARK MLLIB

Sistemas de Recomendación

- **Los filtros colaborativos** :Este modelo luego se usa para predecir los artículos en los que el usuario puede estar interesado.
- **Filtrado basado en contenido** utilizan una serie de características discretas de un artículo para recomendar elementos adicionales, con propiedades similares.
- Estos dos enfoques a menudo se combinan como **sistemas de recomendación híbridos.**

Spark MLlib permite el uso de filtros colaborativos, a través del **algoritmo ALS** (alternating least squares).

SPARK MLLIB

Clustering

- **Basados en la distancia:** utilizan una métrica de distancia, como puede ser la euclídea, Manhattan o Mahalanobis, para separar los distintos grupos o clusters.
- **Basados en la densidad:** computan los grupos teniendo cuenta la distancia entre los puntos y las “zonas” donde la densidad de puntos/observaciones es mayor

  Spark MLlib, permite de los siguientes algoritmos de clustering: K-means, Latent Dirichlet allocation (LDA), Bisecting k-means, Gaussian Mixture Model (GMM)

SPARK MLLIB

Gestión de Features

Extracción: extracción de características de datos "sin procesar"

Transformación: escalado, conversión o modificación de características

Selección: seleccionar un subconjunto de un conjunto más grande de características

Hashing sensible a la localidad (LSH): esta clase de algoritmos combina aspectos de la transformación de características con otros algoritmos.

+info:
https://people.apache.org/~pwendell/spark-nightly/spark-master-docs/latest/ml-features.html

SPARK MLLIB

Optimization

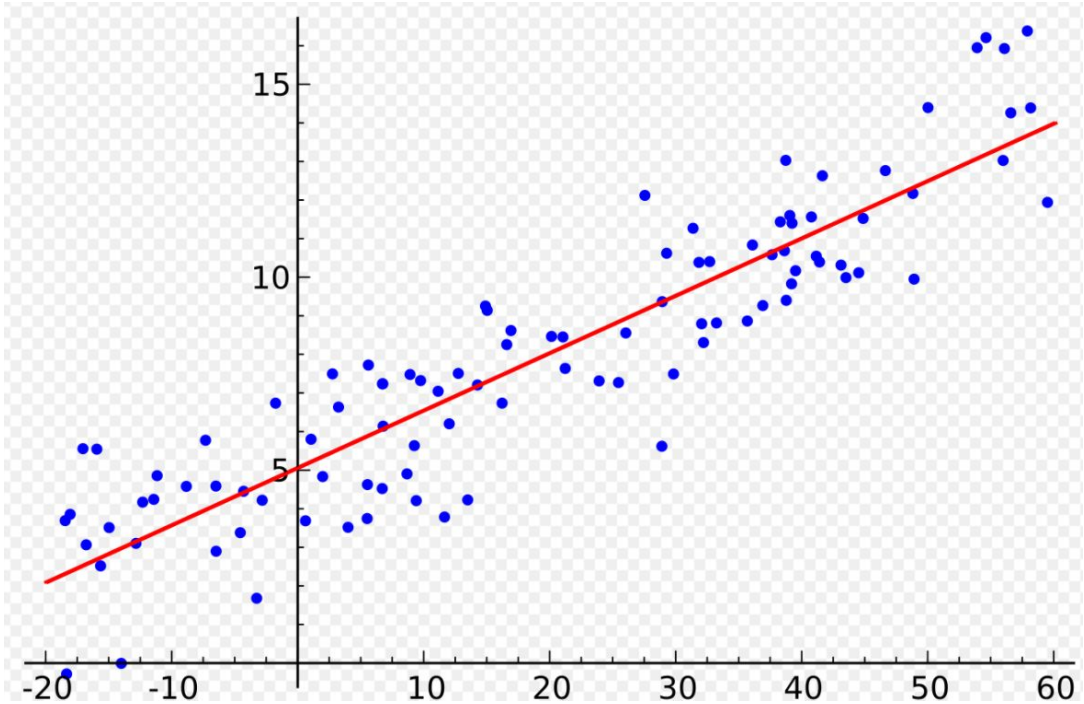
- Optimización de métodos lineales (desarrollador)
- BFGS de memoria limitada (L-BFGS)
- Solucionador de ecuaciones normal para mínimos cuadrados ponderados
- Mínimos cuadrados repetidos iterativamente (IRLS)

https://people.apache.org/~pwendell/spark-nightly/spark-master-docs/latest/ml-advanced.html

SPARK MLLIB

Regresión lineal con Spark

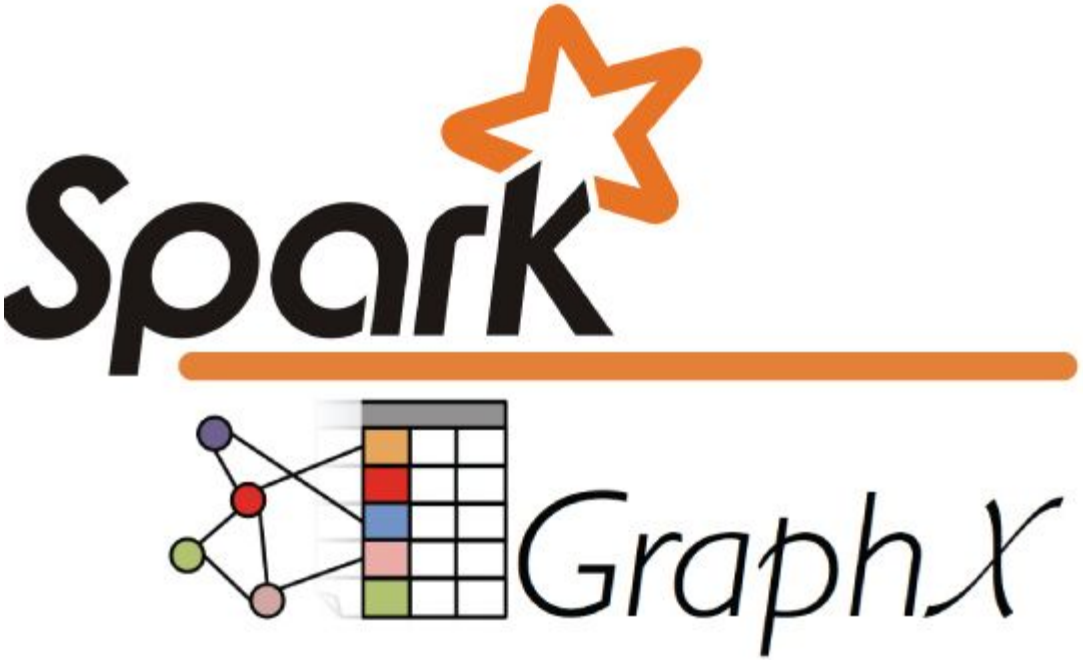
$$Y = \beta_0 + \beta_1 X$$



SPARK MLLIB

	A	B	C	D	E	F	G	H
1	Email	Address	Avatar	Avg Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spen
2	mstephenson@fernandez.com	835 Frank TunnelWrightmouth, MI 82180-9605	Violet	34.49726772511229	12.65565114916675	39.57766801952616	4.0826206329529615	587.9510539684005
3	hduke@hotmail.com	4547 Archer CommonDiazchester, CA 06566-8576	DarkGreen	31.92627202636016	11.109460728682564	37.268958868297744	2.66403418213262	392.2049334443264
4	pallen@yahoo.com	24645 Valerie Unions Suite 582Cobbborough, DC 99414-7564	Bisque	33.000914755642675	11.330278057777512	37.110597442120856	4.104543202376424	487.5475048674720
5	riverarebecca@gmail.com	1414 David ThroughwayPort Jason, OH 22070-1220	SaddleBrown	34.30555662975554	13.717513665142507	36.72128267790313	3.120178782748092	581.8523440352177
6	mstephens@davidson-herman.com	14023 Rodriguez PassagePort Jacobville, PR 37242-1057	MediumAquaMarine	33.33067252364639	12.795188551078114	37.53665330059473	4.446308318351434	599.4060920457634
7	alvareznancy@lucas.biz	645 Martha Park Apt. 611Jeffreychester, MN 67218-7250	FloralWhite	33.871037879341976	12.026925339755056	34.47687762925054	5.493507201364199	637.102447915074
8	katherine20@yahoo.com	68388 Reyes Lights Suite 692Josephbury, WV 92213-0247	DarkSlateBlue	32.02159550138701	11.366348309710526	36.68377615286961	4.685017246570912	521.5721747578274
9	awatkins@yahoo.com	Unit 6538 Box 8980DPO AP 09026-4941	Aqua	32.739142938380326	12.35195897300293	37.37335888554755	4.434273434899375	549.9041461052942
10	vchurch@walter-martinez.com	860 Lee KeyWest Debra, SD 97450-0495	Salmon	33.98777289568564	13.386235275676436	37.534497341555735	3.2734335777477144	570.2004089636196
11	bonnie69@lin.biz	PSC 2734, Box 5255APO AA 98456-7482	Brown	31.936548618448917	11.814128294972196	37.14516822352819	3.202806071553459	427.1993848953282
12	andrew06@peterson.com	26104 Alexander GrovesAlexandriaport, WY 28244-9149	Tomato	33.992572774953736	13.338975447662113	37.22580613162114	2.482607770510596	492.6060127179966
13	ryanwerner@freeman.biz	Unit 2413 Box 0347DPO AA 07580-2652	Tomato	33.87936082480498	11.584782999535266	37.08792607098381	3.71320920294043	522.3374046069357
14	knelson@gmail.com	6705 Miller Orchard Suite 186Lake Shanestad, MO 75696-5051	RoyalBlue	29.532428967057943	10.961298400154098	37.42021557502538	4.046423164299585	408.6403510726275
15	wrightpeter@yahoo.com	05302 Dunlap FerryNew Stephaniehaven, MP 42268	Bisque	33.19033404372265	12.959226091609382	36.144666700041924	3.918541839158999	573.4158673313865
16	taylormason@gmail.com	7773 Powell Springs Suite 190Samanthaland, ND 44358	DarkBlue	32.387975853153876	13.148725692056516	36.61995708279922	2.494543646659249	470.4527333009554
17	jstark@anderson.com	49558 Ramirez Road Suite 399Phillipstad, OH 35641-3238	Peru	30.737720372628182	12.636606052000127	36.213763093698624	3.3578468423262944	461.7807421962299
18	wjennings@gmail.com	6362 Wilson MountainJohnsonfurt, GA 15169	PowderBlue	32.12538689728784	11.733861690857394	34.8940927514398	3.1361327164897803	457.8476959449485
19	rebecca45@hale-bauer.biz	8982 Burton RowWilsonton, PW 88606	OliveDrab	32.338899323067196	12.013194694014402	38.38513659413844	2.420806160901484	407.7045475495441
20	alejandro75@hotmail.com	64475 Andre Club Apt. 795Port Dannytown, PW 63227	Cyan	33.187812045932155	14.715387544156 5	38.24411459434252	1.516575589831944	452.31567548003254

SPARK GRAPHX



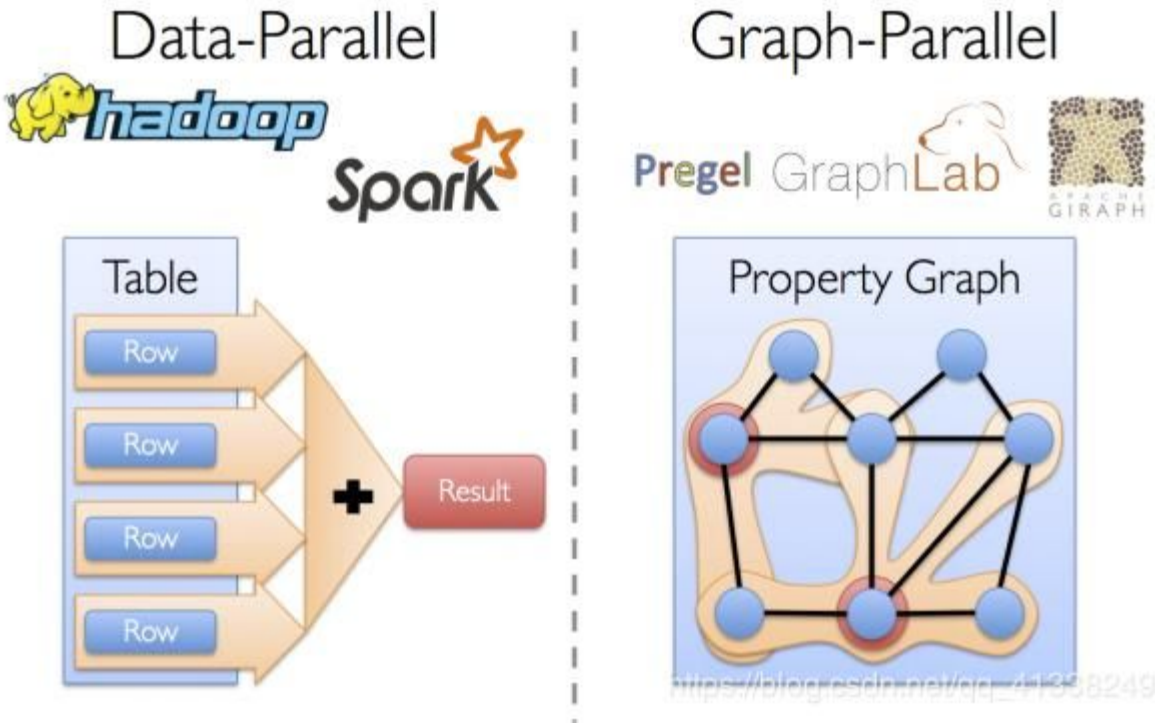
SPARK GRAPHX

¿Qué es Spark GraphX?

- GraphX es un componente para gráficos y cálculo de gráficos en Spark.
- Tiene una estructura de datos abstractos basada en RDD
- Ofrece una serie de operaciones básicas como InDegress, OutDegress, subgraph, etc.
- Dispone de algoritmos de cálculo de gráficos PageRank, TriangleCount, ConnectedComponents, etc.

https://spark.apache.org/docs/latest/graphx-programming-guide.html#overview

SPARK GRAPHX



APACHE SPARK
SESIÓN 3

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL
GENERALITAT VALENCIANA
Conselleria d'Educació, Investigació, Cultura i Esport
cefire
UNIÓ EUROPEA
Fons Social Europeu
L'FSE inverteix en el teu futur