

Machine Learning en Big Data Pyspark



AGENDA

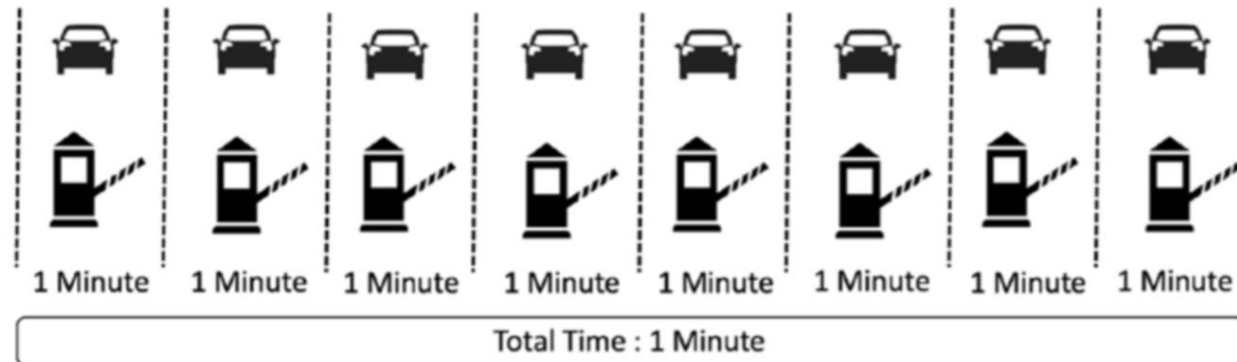
- 1. Acerca de Pyspark
- 2. Instalación Pyspark en Google colab
- 3. Importando archive de datos y adaptando a format Spark
- 4. Data Wrangling / Manipulación de datos con Pyspark
 - Lectura, Formato
 - Agrupamiento
 - EDA / Analisis exploratorio de datos
 - Identificación de valores Nulos
 - Tratamiento de Outliers
- 5. RDD vs DataFrame en Spark. Paralelizacion de operaciones
- Etapas de modelamiento de datos en Colab con Pyspark

1 ACERCA DE PYSPARK

PROCESAMIENTO DE COMPUTO PARALELO FRENTE A BIG DATA



- Lenguajes transaccionales como SQL responden a tareas en secuencia

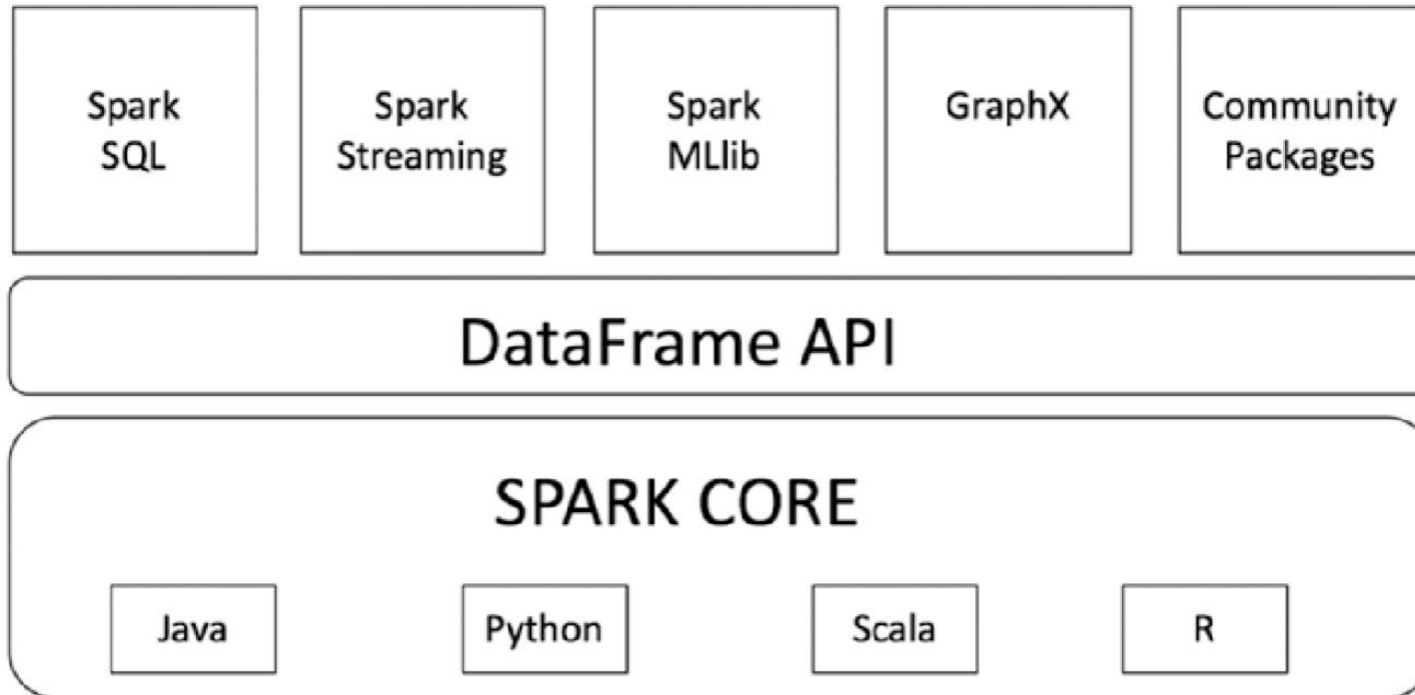


- Spark responde a procesamiento en paralelo de operaciones distribuyendo tareas y respondiendo rápidamente a operaciones frente a grandes volúmenes de datos

ACERCA DE SPARK



- Spark es un plataforma de entorno de trabajo distribuida que trabaja con nodos maestros y nodos trabajadores donde se almacenan los datos
- Spark utiliza diferentes estructuras de data como RDD o Dataframe

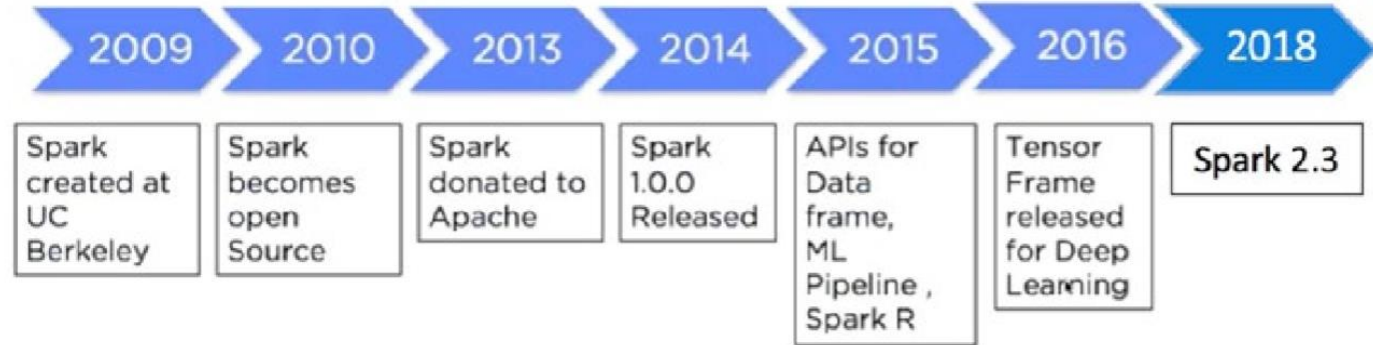


- **Spark SQL:** Busca optimizar el procesamiento de datos. Considerado SQL query distribuido
- **Spark Streaming:** Procesa en tiempo real los datos con micro procesos batch
- **Spark Mlib:** Machine Learning distribuido
- **Spark Graph:** Utilizado para comprender relaciones y visualizar insight

ACERCA DE PYSPARK



- Plataforma computacional diseñada para la **rapidez**, propósito general y uso sencillo.
- Propósito general ejecuta procesos: batch, iterativas y streaming
- Sistema de **alta integración** con API's: Scala, Python, Java, R y SQL



- Es el entorno de trabajo más utilizado para resolver problemas reales de Machine Learning de manera rápida y distribuida
- A diferencia de Python esta Pyspark esta preparado para procesar los algoritmos de manera distribuida

2 INSTALACION DE PYSPARK

GOOGLE COLAB

The screenshot shows the Google Colaboratory interface in a web browser. The browser's address bar displays the URL `colab.research.google.com/notebooks/welcome.ipynb`. The page title is "Te damos la bienvenida a Colaboratory". The interface includes a top navigation bar with options like "Archivo", "Editar", "Ver", "Insertar", "Entorno de ejecución", "Herramientas", and "Ayuda". A sidebar on the left contains a table of contents with links to "Introducción a Colaboratory", "Primeros pasos", "Más recursos", and "Ejemplos de aprendizaje automático: Seedbank". The main content area features a welcome message in Spanish, explaining that Colaboratory is a free Jupyter Notebook environment. Below this, there is a section titled "Introducción a Colaboratory" which includes a video player. The video player shows a thumbnail for a video titled "Intro to Google Colab" by Coding TensorFlow, featuring a man smiling. The video player controls include a play button, a progress bar, and a "Compartir" (Share) button. The bottom of the image shows the Windows taskbar with various application icons and the system clock indicating 1:00 on 19/10/2019.

Te damos la bienvenida a Colaboratory

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ Código + Texto Copiar en Drive

Conectar Editar

Te damos la bienvenida a Colaboratory

Colaboratory es un entorno gratuito de Jupyter Notebook que no requiere configuración y que se ejecuta completamente en la nube. Colaboratory te permite escribir y ejecutar código, guardar y compartir tus análisis y tener acceso a recursos informáticos muy potentes, todo de forma gratuita desde el navegador.

Introducción a Colaboratory

En este vídeo de 3 minutos puedes ver una descripción general de las funciones principales de Colaboratory:

Get started with Google Colaboratory (C...)

Ver más tarde Compartir

Intro to Google Colab

Coding TensorFlow

INSTALACION INICIALES



Paso1 BIC_Manipulacion_de_datos_Pyspark_.ipynb ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se han guardado todos los cambios](#)

+ Código + Texto

```
[1] !apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q https://www-us.apache.org/dist/spark/spark-2.3.4/spark-2.3.4-bin-hadoop2.7.tgz
!tar xf spark-2.3.4-bin-hadoop2.7.tgz
```

```
!wget -q https://www-us.apache.org/dist/spark/spark-2.4.4/spark-2.4.4-bin-hadoop2.7.tgz
!tar xf spark-2.4.4-bin-hadoop2.7.tgz
```

```
[2] !pip install -q findspark
```

```
[3] import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-2.4.4-bin-hadoop2.7"
```

```
[4] !pip install pyspark
```

```
[5] from pyspark.sql import SparkSession
from pyspark import SparkContext
spark = SparkSession.builder.master("local").getOrCreate()
sc = SparkContext.getOrCreate()
```

3 IMPORTANDO ARCHIVOS DE DATOS Y FORMATO SPARK

IMPORTANDO ARCHIVOS DE DATOS DESDE CSV HASTA DRIVE



1. Lectura de datos

```
[ ] from google.colab import files
    uploaded = files.upload()
```

Elegir archivos 0. DS_Seguros_Salud.csv

- 0. DS_Seguros_Salud.csv(application/vnd.ms-excel) - 2690973 bytes, last modified: 10/10/2019 - 100% done

Saving 0. DS_Seguros_Salud.csv to 0. DS_Seguros_Salud.csv

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=https://colab.research.google.com/&response_type=code

Enter your authorization code:
.....
Mounted at /content/drive

```
[ ] !ls '/content/drive/My Drive/PySpark Machine Learning en plataforma Big Data/ML Sesión 3 Training R ML en Casos de Negocios/2_caso negocio'
```

'0. Caso negocio endeudamiento credito.pptx' 2_DS_creditos.csv
'1_Diccionario credito.xls'

```
[ ] ls
```

'0. DS_Seguros_Salud.csv' spark-2.3.4-bin-hadoop2.7.tgz
drive/ spark-2.4.4-bin-hadoop2.7/
sample_data/ spark-2.4.4-bin-hadoop2.7.tgz
spark-2.3.4-bin-hadoop2.7/



Google Drive



IMPORTANDO ARCHIVOS DE DATOS DESDE CSV HASTA DRIVE



```
DS_Salud = spark.read.csv('0. DS_Seguros_Salud.csv',  
    sep=';', header=True, inferSchema=True)  
  
# 2.1 Revision de formatos  
DS_Salud.printSchema()
```



Google Drive



```
DS_cred = spark.read.csv('/content/drive/My Drive/P  
ySpark Machine Learning en plataforma Big Data/ML S  
esion 3 Training R ML en Casos de Negocios/2_caso n  
egocio 2 riesgo crediticio/2_DS_creditos.csv', sep=  
,',', header=True, inferSchema=True)  
  
# 2.1 Revision de formatos  
DS_cred.printSchema()
```



Data Pre-processing & Data Wrangling in Machine learning & Deep Learning

4 DATA WRANGLING

4.1 EXPLORACION DATAFRAME

4.1 EXPLORACION DE DATAFRAME

Explora el format de las variables



```
[In]: df.printSchema()  
[Out]: root  
|-- MES_T0: integer (nullable = true)
```

```
[In]: df.columns
```

```
[Out]: ['ratings', 'age', 'experience', 'family', 'mobile']
```

```
[In]: len(df.columns)
```

```
[Out]: 5
```

```
[In]: df.count
```

```
[Out]: ['ratings', 'age', 'experience', 'family', 'mobile']
```

4.1 EXPLORACION DE DATAFRAME

Revision de filas de una tabla

```
[In]: print((df.count) , (len(df.columns))
```

```
[Out]: ( 33,5)
```

```
[In]: df.show(3)
```

```
[Out]:
```

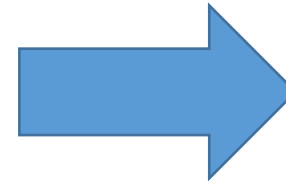
```
+-----+-----+-----+-----+-----+
| ratings | age | experience | family | mobile |
+-----+-----+-----+-----+-----+
|         | 3  | 32         | 9.0    | 3      | Vivo   |
|         | 3  | 27         | 13.0   | 3      | Apple  |
|         | 4  | 22         | 2.5    | 0      | Samsung|
|         | 4  | 37         | 16.5   | 4      | Apple  |
|         | 5  | 27         | 9.0    | 1      | MI     |
+-----+-----+-----+-----+-----+
```

only showing top 5 rows

```
[In]: df.select('age','mobile').show(5)
```

```
[Out]:
```

```
+---+-----+
| age | mobile |
+---+-----+
| 32  | Vivo   |
| 27  | Apple  |
| 22  | Samsung|
| 37  | Apple  |
| 27  | MI     |
+---+-----+
```



4.1 EXPLORACION DE DATAFRAME

```
[In]: df.describe().show()
```

```
[Out]:
```

summary	ratings	age	experience	family	mobile
count	33	33	33	33	33
mean	3.5757575757575757	30.484848484848484	10.303030303030303	1.8181818181818181	null
stddev	1.1188806636071336	6.18527087180309	6.770731351213326	1.8448330794164254	null
min	1	22	2.5	0	Apple
max	5	42	23.0	5	Vivo

4.2

PRE-PROCESAMIENTO DE DATOS

4.2 AGREGANDO UNA COLUMNA

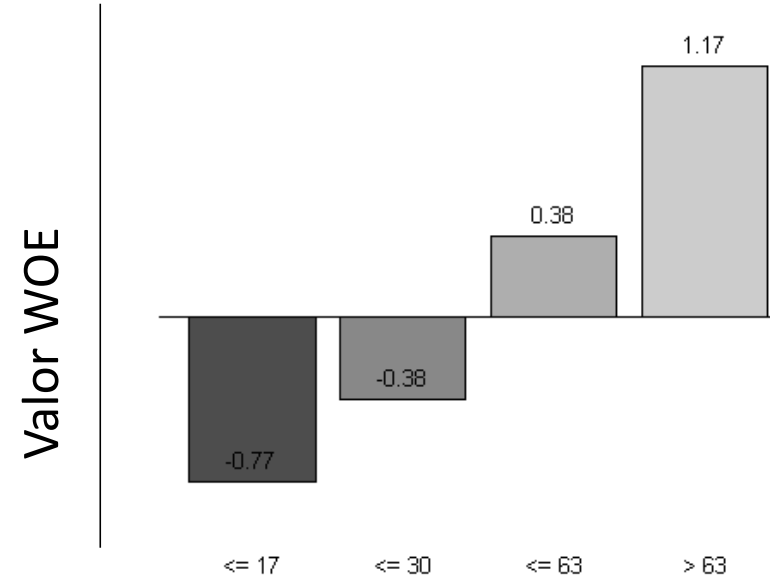
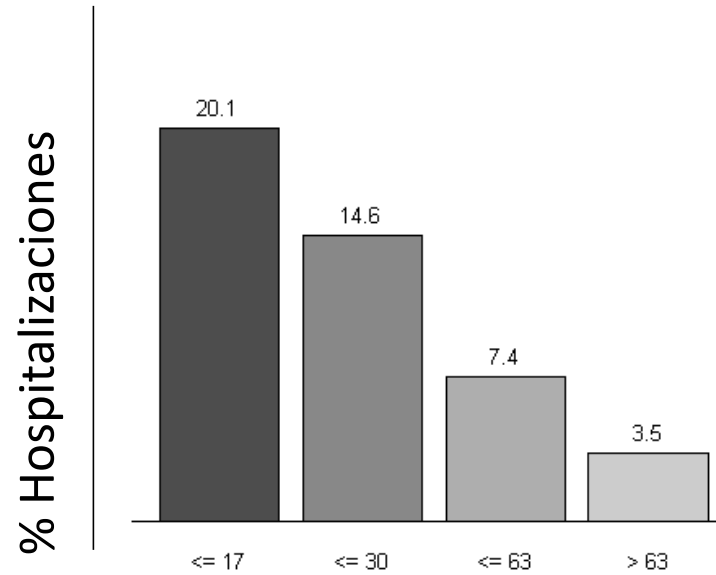
```
[In]: df.withColumn("age_after_10_yrs", (df["age"]+10)).  
      show(10, False)
```

[Out]:

ratings	age	experience	family	mobile	age_after_10_yrs
3	32	9.0	3	Vivo	42
3	27	13.0	3	Apple	37
4	22	2.5	0	Samsung	32
4	37	16.5	4	Apple	47
5	27	9.0	1	MI	37
4	27	9.0	0	Oppo	37
5	37	23.0	5	Vivo	47
5	37	23.0	5	Samsung	47
3	22	2.5	0	Apple	32
3	27	6.0	0	MI	37

only showing top 10 rows

Transformación WOE



Hemoglobina

Hemoglobina

<u>Paciente</u>	<u>Trans Mean enode</u>	<u>Trans WOE</u>
<17 Hemog	20.1%	-0.77%
>63 Hemog	3.5%	1.17

$$WOE = \ln \left(\frac{\% \text{ of non-events}}{\% \text{ of events}} \right)$$

Weight of Evidence Formula

lib Scorecardpy