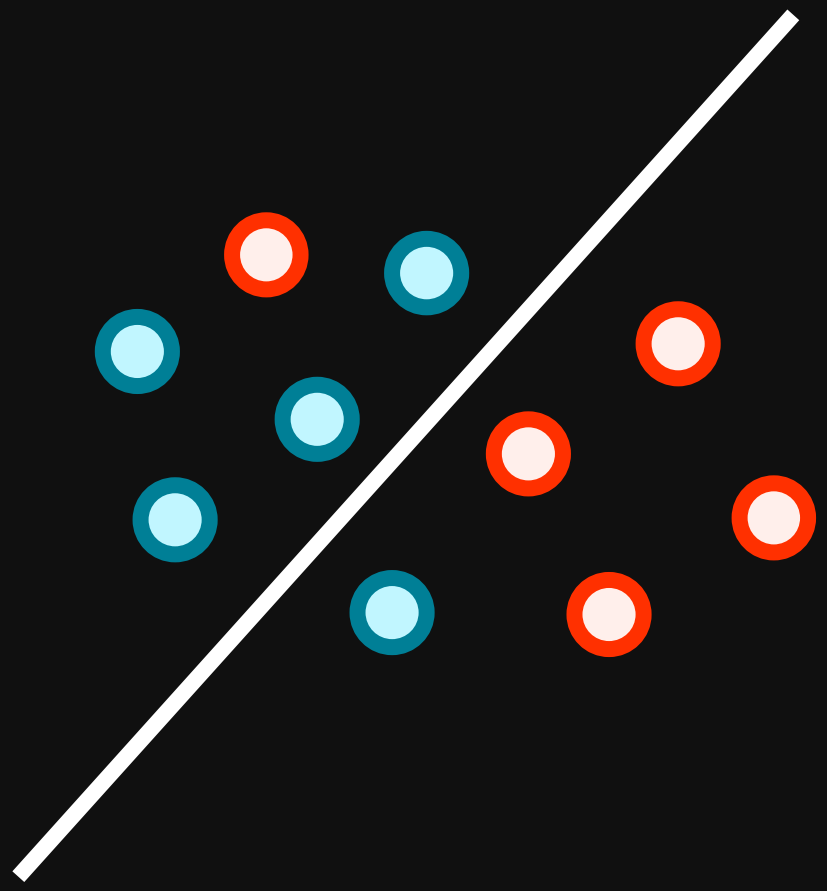
Basics of Machine Learning

Dmitry Ryabokon, github.com/dryabokon



Lesson 22
PySpark

Installation

Install Pyspark @ Conda

```
> wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
> chmod 777 Miniconda3-latest-Linux-x86_64.sh
> sudo ./Miniconda3-latest-Linux-x86_64.sh
> sudo bash

conda create -n p37 python=3.7
conda activate p37
conda install -c conda-forge pyspark


(p37) root@instance-test-dr:/opt/spark/spark-3.2.0-bin-hadoop2.7/bin# ./pyspark
```

Install Pyspark @ Conda

Spark Ubuntu

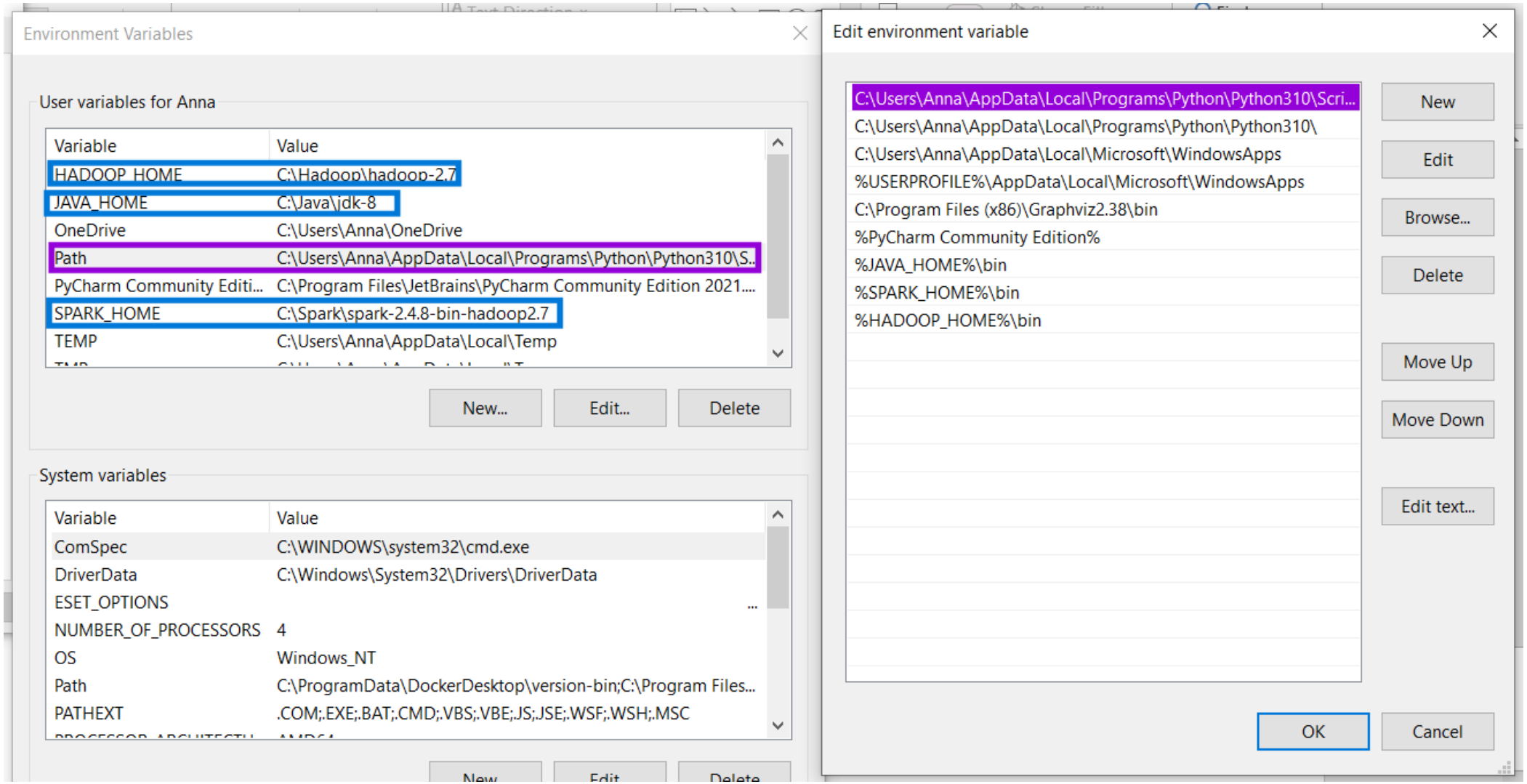
mc [root@instance-test-dr]:/opt/spark/spark-3.2.0-bin-hadoop2.7/bin

```
root@instance-test-dr:/opt/spark/spark-3.2.0-bin-hadoop2.7/bin# java -version; javac -version; scala -version; git --version
openjdk version "11.0.12" 2021-07-20
OpenJDK Runtime Environment (build 11.0.12+7-post-Debian-2deb10u1)
OpenJDK 64-Bit Server VM (build 11.0.12+7-post-Debian-2deb10u1, mixed mode, sharing)
javac 11.0.12
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
git version 2.20.1
root@instance-test-dr:/opt/spark/spark-3.2.0-bin-hadoop2.7/bin#

root@instance-test-dr:/opt/spark/spark-3.2.0-bin-hadoop2.7/bin# java -version; javac -version; scala -version; git --version
openjdk version "11.0.12" 2021-07-20
OpenJDK Runtime Environment (build 11.0.12+7-post-Debian-2deb10u1)
OpenJDK 64-Bit Server VM (build 11.0.12+7-post-Debian-2deb10u1, mixed mode, sharing)
javac 11.0.12
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
git version 2.20.1
root@instance-test-dr:/opt/spark/spark-3.2.0-bin-hadoop2.7/bin# ./pyspark
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/spark-3.2.0-bin-hadoop2.7/jars/spark-unsafe_2.12-3.2.0.jar) to constructor java.nio.DirectByteBuffer(
long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
21/10/28 19:38:04 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.2.0
      /_/

Using Python version 3.7.3 (default, Jan 22 2021 20:04:44)
Spark context Web UI available at http://instance-test-dr.c.ml-ops-poc-695.internal:4040
Spark context available as 'sc' (master = local[*], app id = local-1635449885599).
SparkSession available as 'spark'.
>>> nums = sc.parallelize([1, 2, 3, 4])
>>> print(nums.map(lambda x: x * x).collect())
[1, 4, 9, 16]
>>>
```

Spark Windows PC



Spark Windows PC

```
(base) C:\Users\Anna>java -version
java version "1.8.0_301"
Java(TM) SE Runtime Environment (build 1.8.0_301-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.301-b09, mixed mode)

(base) C:\Users\Anna>cd c:/

(base) c:\>cd Spark

(base) c:\Spark>cd spark-2.4.8-bin-hadoop2.7/bin

(base) c:\Spark\spark-2.4.8-bin-hadoop2.7\bin>pyspark
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.8
      /_/

Using Python version 3.7.4 (default, Aug  9 2019 18:34:13)
SparkSession available as 'spark'.
>>> nums = sc.parallelize([1, 2, 3, 4])
>>> print(nums.map(lambda x: x * x).collect())
[1, 4, 9, 16]
>>> 
```

RDD vs Dataframe vs Dataset

https://data-flair.training/blogs/apache-spark-rdd-vs-dataframe-vs-dataset/

https://www.analyticsvidhya.com/blog/2020/11/what-is-the-difference-between-rdds-dataframes-and-datasets/

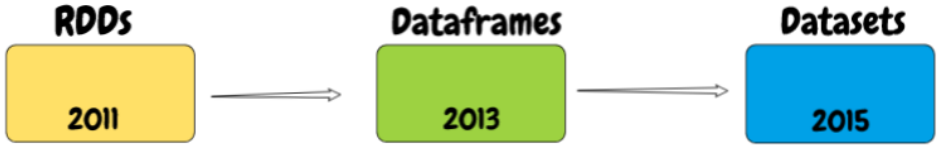
RDD vs Dataframe vs Dataset

RDD

	RDDs	Dataframes	Datasets
Data Representation	Distributed collection of data elements without any schema.	Distributed collection organized into the **named** columns	It is an extension of Dataframes with more features like type-safety and object-oriented interface.
Optimization	No in-built optimization engine for RDDs. Developers need to write the optimized code themselves.	It uses a catalyst optimizer for optimization.	It also uses a catalyst optimizer for optimization purposes.
Projection of Schema	Here, we need to define the schema manually.	It will automatically find out the schema of the dataset.	It will also automatically find out the schema of the dataset by using the SQL Engine.
Aggregation Operation	RDD is slower than both Dataframes and Datasets to perform simple operations like grouping the data.	It provides an easy API to perform aggregation operations. It performs aggregation faster than both RDDs and Datasets.	Dataset is faster than RDDs but a bit slower than Dataframes.

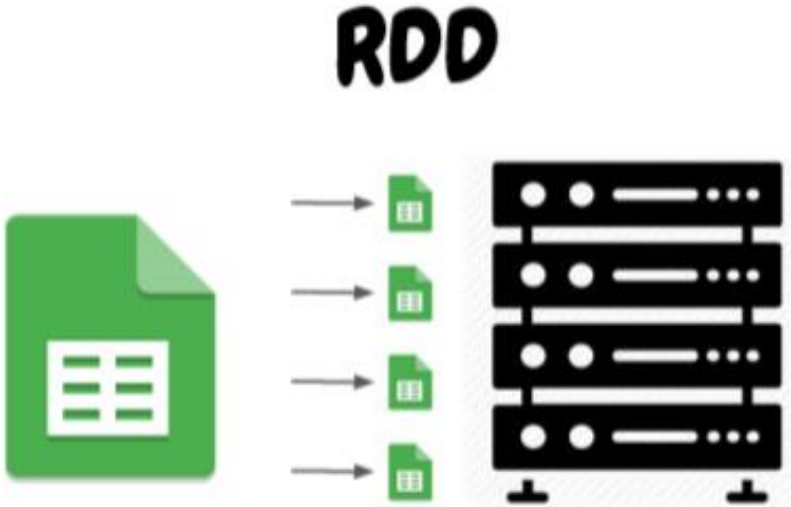
RDD vs Dataframe vs Dataset

RDD



RDDs or Resilient Distributed Datasets is the collection of objects which is capable of fault-tolerant storing the data partitioned across the multiple nodes of the cluster and also allows them to do processing in parallel.

There are 4 ways of creating an RDD:
1.Parallelizing an existing collection of data
2.Referencing to the external data file stored
3. From an existing RDD
4. from existing DataFrames and DataSets



RDDs usage:
Low-level transformations
Handling unstructured data (media / text streams)

RDD: split

```
>>> RDD_text = sc.textFile(name="file:///C://Users//Anna//source//digits//ML//data//ex_datasets//dataset_wine.csv")
>>> RDD_text.take(3)
['target\talcohol\tmalic_acid\tash\talcalinity_of_ash\tmagnesium\ttotal_phenols\tflavanoids\tnonflavanoid_phenols\tproanthocyanins\t
color_intensity\thue\tod280/od315_of_diluted_wines\tproline',
'0.0\t14.23\t1.71\t2.43\t15.6\t127.0\t2.8\t3.06\t0.28\t2.29\t5.64\t1.04\t3.92\t1065.0',
'0.0\t13.2\t1.78\t2.14\t11.2\t100.0\t2.65\t2.76\t0.26\t1.28\t4.38\t1.05\t3.4\t1050.0’]


>>> RDD_text_split = RDD_text.map(lambda line: line.split('\t'))
>>> RDD_text_split.take(3)
[['target', 'alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_phenols', 'flavanoids', 'nonflavanoid_phenols',
'proanthocyanins', 'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline'], ['0.0', '14.23', '1.71', '2.43', '15.6',
'127.0', '2.8', '3.06', '0.28', '2.29', '5.64', '1.04', '3.92', '1065.0'], ['0.0', '13.2', '1.78', '2.14', '11.2', '100.0', '2.65',
'2.76', '0.26', '1.28', '4.38', '1.05', '3.4', '1050.0']]
```

RDD vs Dataframe vs Dataset

Dataframes

DF is an immutable distributed collection of data.
Unlike an RDD, data is organized into named columns, similar to a table in a relational database or Python Pandas DataFrames

Usage:
- Rich semantics, high-level abstractions, and domain specific APIs
- High-level expressions, filters, maps, aggregation, averages, sum, SQL queries, columnar access and use of lambda functions on semi-structured data

RDD vs Dataframe vs Dataset

Dataframes

```
>>> DF_titanic = sqlContext.read.csv("file:///C://Users//Anna//source//digits//ML//data//ex_datasets//dataset_titanic.csv", header=True, sep='\t')
>>> DF_titanic.printSchema()
root
 |-- survived: string (nullable = true)
 |-- pclass: string (nullable = true)
 |-- sex: string (nullable = true)
 |-- age: string (nullable = true)
 |-- sibsp: string (nullable = true)
 |-- parch: string (nullable = true)
 |-- fare: string (nullable = true)
 |-- embarked: string (nullable = true)
 |-- class: string (nullable = true)
 |-- who: string (nullable = true)
 |-- adult_male: string (nullable = true)
 |-- deck: string (nullable = true)
 |-- embark_town: string (nullable = true)
 |-- alive: string (nullable = true)
 |-- alone: string (nullable = true)

>>> DF_titanic.take(3)
[Row(survived='0', pclass='3', sex='male', age='22.0', sibsp='1', parch='0', fare='7.25', embarked='S', class='Third', who='man',
adult_male='True', deck=None, embark_town='Southampton', alive='no', alone='False'), Row(survived='1', pclass='1', sex='female', age='38.0',
sibsp='1', parch='0', fare='71.2833', embarked='C', class='First', who='woman', adult_male='False', deck='C', embark_town='Cherbourg',
alive='yes', alone='False'), Row(survived='1', pclass='3', sex='female', age='26.0', sibsp='0', parch='0', fare='7.925', embarked='S',
class='Third', who='woman', adult_male='False', deck=None, embark_town='Southampton', alive='yes', alone='True')]
```

RDD vs Dataframe vs Dataset

Dataframes

```
>>> DF_titanic = sqlContext.read.csv("file:///C://Users//Anna//source//digits//ML//data//ex_datasets//dataset_titanic.csv", header=True, sep='\t’)

>>> DF_titanic.show()
+--------+------+------+----+-----+-----+-------+--------+------+-----+----------+----+-----------+-----+-----+
|survived|pclass|   sex| age|sibsp|parch|   fare|embarked| class|  who|adult_male|deck|embark_town|alive|alone|
+--------+------+------+----+-----+-----+-------+--------+------+-----+----------+----+-----------+-----+-----+
|       0|     3|  male|22.0|    1|    0|   7.25|       S| Third|  man|      True|null|Southampton|   no|False|
|       1|     1|female|38.0|    1|    0|71.2833|       C| First|woman|     False|   C|  Cherbourg|  yes|False|
|       1|     3|female|26.0|    0|    0|  7.925|       S| Third|woman|     False|null|Southampton|  yes| True|
|       1|     1|female|35.0|    1|    0|   53.1|       S| First|woman|     False|   C|Southampton|  yes|False|
|       0|     3|  male|35.0|    0|    0|   8.05|       S| Third|  man|      True|null|Southampton|   no| True|
|       0|     3|  male|null|    0|    0| 8.4583|       Q| Third|  man|      True|null| Queenstown|   no| True|
|       0|     1|  male|54.0|    0|    0|51.8625|       S| First|  man|      True|   E|Southampton|   no| True|
|       0|     3|  male| 2.0|    3|    1| 21.075|       S| Third|child|     False|null|Southampton|   no|False|
|       1|     3|female|27.0|    0|    2|11.1333|       S| Third|woman|     False|null|Southampton|  yes|False|
|       1|     2|female|14.0|    1|    0|30.0708|       C|Second|child|     False|null|  Cherbourg|  yes|False|
|       1|     3|female| 4.0|    1|    1|   16.7|       S| Third|child|     False|   G|Southampton|  yes|False|
|       1|     1|female|58.0|    0|    0|  26.55|       S| First|woman|     False|   C|Southampton|  yes| True|
|       0|     3|  male|20.0|    0|    0|   8.05|       S| Third|  man|      True|null|Southampton|   no| True|
|       0|     3|  male|39.0|    1|    5| 31.275|       S| Third|  man|      True|null|Southampton|   no|False|
|       0|     3|female|14.0|    0|    0| 7.8542|       S| Third|child|     False|null|Southampton|   no| True|
|       1|     2|female|55.0|    0|    0|   16.0|       S|Second|woman|     False|null|Southampton|  yes| True|
|       0|     3|  male| 2.0|    4|    1| 29.125|       Q| Third|child|     False|null| Queenstown|   no|False|
|       1|     2|  male|null|    0|    0|   13.0|       S|Second|  man|      True|null|Southampton|  yes| True|
|       0|     3|female|31.0|    1|    0|   18.0|       S| Third|woman|     False|null|Southampton|   no|False|
|       1|     3|female|null|    0|    0|  7.225|       C| Third|woman|     False|null|  Cherbourg|  yes| True|
+--------+------+------+----+-----+-----+-------+--------+------+-----+----------+----+-----------+-----+-----+
```

Datasets

In Spark 2.0, Dataset and DataFrame merge into one unit to reduce the complexity while learning Spark.

DataFrame – is an *alias* for a collection of generic objects *Dataset[Row]*, where a *Row* is a generic ***untyped*** JVM object.

Dataset, by contrast, is a collection of ***strongly-typed*** JVM objects, dictated by a case class you define in Scala or a class in Java