



NahumFGz /  
TareaLayla



<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

TareaLayla / clase\_02 / practica\_c\_solucion.ipynb



**NahumFGz** feat: ✨ Semana 4 terminado

af1d94d · last week



456 lines (456 loc) · 11 KB

Preview

Code

Blame



Raw



```
In [0]: from pyspark.sql.functions import udf
from pyspark.sql.types import IntegerType, StringType, BooleanType, DoubleType
import math
```

```
In [0]: # Definir la función cuadrado
def cuadrado(num):
    return num * num

# Registrar la UDF
cuadrado_udf = udf(cuadrado, IntegerType())

# Crear un DataFrame de ejemplo
data = [(2,), (3,), (4,)]
columns = ["numero"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("cuadrado", cuadrado_udf(df["numero"])).show()
```

numero	cuadrado
2	4
3	9
4	16

```
In [0]: # Definir la función longitud
def longitud(cadena):
    return len(cadena)

# Registrar la UDF
longitud_udf = udf(longitud, IntegerType())

# Crear un DataFrame de ejemplo
data = [("Hola",), ("PySpark",), ("Mundo",)]
columns = ["cadena"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("longitud", longitud_udf(df["cadena"])).show()
```

cadena	longitud
Hola	4
PySpark	7
Mundo	5

```
In [0]: # Definir la función es_par
def es_par(num):
    return num % 2 == 0

# Registrar la UDF
es_par_udf = udf(es_par, BooleanType())
```

```
# Crear un DataFrame de ejemplo
data = [(2,), (3,), (4,)]
columns = ["numero"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("es_par", es_par_udf(df["numero"])).show()
```

```
+-----+-----+
|numero|es_par|
+-----+-----+
|      2|  true|
|      3| false|
|      4|  true|
+-----+-----+
```

In [0]:

```
# Definir la función revertir
def revertir(cadena):
    return cadena[::-1]

# Registrar la UDF
revertir_udf = udf(revertir, StringType())

# Crear un DataFrame de ejemplo
data = [("Hola",), ("PySpark",), ("Mundo",)]
columns = ["cadena"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("revertida", revertir_udf(df["cadena"])).show()
```

```
+-----+-----+
|cadena|revertida|
+-----+-----+
|  Hola|    aloH|
|PySpark| krapSyP|
|  Mundo|   odnuM|
+-----+-----+
```

In [0]:

```
# Definir la función calcular_area_circulo
def calcular_area_circulo(radio):
    return math.pi * radio**2

# Registrar la UDF
calcular_area_circulo_udf = udf(calcular_area_circulo, DoubleType())

# Crear un DataFrame de ejemplo
data = [(1,), (2,), (3,)]
columns = ["radio"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("area", calcular_area_circulo_udf(df["radio"])).show()
```

```
+-----+-----+
|radio|          area|
+-----+-----+
```

```
+-----+
| 1| 3.141592653589793|
| 2|12.566370614359172|
| 3|28.274333882308138|
+-----+
```

```
In [0]: # Definir la función convertir_mayusculas
def convertir_mayusculas(cadena):
    return cadena.upper()

# Registrar la UDF
convertir_mayusculas_udf = udf(convertir_mayusculas, StringType())

# Crear un DataFrame de ejemplo
data = [("hola",), ("pyspark",), ("mundo",)]
columns = ["cadena"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("mayusculas", convertir_mayusculas_udf(df["cadena"])).show()
```

```
+-----+
| cadena|mayusculas|
+-----+
| hola| HOLA|
| pyspark| PYSPARK|
| mundo| MUNDO|
+-----+
```

```
In [0]: # Definir la función calcular_edad
def calcular_edad(anio_nacimiento):
    return 2023 - anio_nacimiento

# Registrar la UDF
calcular_edad_udf = udf(calcular_edad, IntegerType())

# Crear un DataFrame de ejemplo
data = [(2000,), (1990,), (1985,)]
columns = ["anio_nacimiento"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("edad", calcular_edad_udf(df["anio_nacimiento"])).show()
```

```
+-----+
|anio_nacimiento|edad|
+-----+
| 2000| 23|
| 1990| 33|
| 1985| 38|
+-----+
```

```
In [0]: # Definir la función es_vocal
def es_vocal(letra):
    return letra.lower() in "aeiou"

# Registrar la UDF
```

```
es_vocal_udf = udf(es_vocal, BooleanType())

# Crear un DataFrame de ejemplo
data = [("a",), ("b",), ("e",)]
columns = ["letra"]
df = spark.createDataFrame(data, columns)

# Usar la UDF en el DataFrame
df.withColumn("es_vocal", es_vocal_udf(df["letra"])).show()
```

```
+-----+-----+
|letra|es_vocal|
+-----+-----+
|    a|    true|
|    b|   false|
|    e|    true|
+-----+-----+
```

In [0]: