escola britânica de artes criativas & tecnologia

Profissão: Analista de dados

escola britânica de artes criativas & tecnologia

BIG DATA I - PROCESSAMENTO

GUIA DA AULA 3

Conheça o Data Wrangling

- Exploração
- Limpeza
- Junção
- Escrita

Acompanhe aqui os temas que serão tratados na videoaula

Exploração

Vamos utilizar a API Python do Spark, o pacote PySpark, para limpar os dados da aula 2.

Nota: Atente-se sempre à natureza distribuída das operações.

In [ ]:
```
data.count()
```

In [ ]:
```
data.columns
```

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

Limpeza

O método `select` seleciona colunas do DataFrame. Já o método `withColumnRenamed` renomeia colunas.

In [ ]:

```
data = data.select([
    "Description",
    "Population
    (GB+NI)",
    "Unemployment rate"
])
```

In [ ]:

```
data = data.\
  withColumnRenamed
  (
    "Description", 'year'
).\
  withColumnRenamed(
    "Population (GB+NI)", "population"
).\
  withColumnRenamed(
    "Unemployment rate", "unemployment_rate"
)
```

In [ ]:

```
data.show(n=10)
```

O método `filter` seleciona linhas do `DataFrame` baseado no conteúdo de uma coluna

In [ ]:
```
data_description = data.filter(data['year'] == 'Units')
```

In [ ]:
```
data_description.show(n=10)
```

Junção

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

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

O método `join` faz a junção de dois `DataFrames`. Já o método `broadcast` marca" um `DataFrame` como "pequeno" e força o Spark a trafegá-lo pela rede.

In [ ]:
```
from pyspark.sql.functions import broadcast
```

In [ ]:
```
data = data.join(
    other=broadcast(data_description)
    , on=['year'],
    how='left_anti'
)
```

In [ ]:
```
data.show(n=10)
```

O método `dropna` remove todas as linhas que apresentarem ao menos um valor nulo.

In [ ]:
```
data = data.dropna()
```

In [ ]:
```
data.show(n=10)
```

O método `withColumn` ajuda a criar novas colunas.

In [ ]:
```
data =
    data.withColumn(
    'century',
    1 + (data['year']/100).cast('int')
)
```

In [ ]:
```
data.\
    select(['century', 'year']).\
    groupBy('century').\
    agg({'year':
    'count'}).show()\
```

O método `colletc` é uma ação que coleta os resultados dos nós e retorna para o Python.

In [ ]:
```
timing = data.\
    select(['century',
    'year']).\
    groupBy('century').\
    agg({'year': 'count'}).\
    collect()
```

In [ ]:
```
timing
```

In [ ]:
```
timing[0].asDict()
```

Escrita

O método `write.csv` persiste o `DataFrame` em formato `csv`. Já o método `repartition` controla o número de partições da escrita.

In [ ]:

```
data.\
    repartition('century').
    \ write.\
    csv(
        path="uk-macroeconomic-data-clean"
        , sep=",",
        header=True,
        mode="overwrite
        "
    )
```