HOW TO TRANSLATE PANDAS KNOWLEDGE TO PYSPARK

	PANDAS	PYSPARK
LOAD CSV	➔ df = pd.read_csv('/…/…/document.csv')	➔ df = spark.read \ .options (header=True, inferSchema=True) \ .csv('/…/…/document.csv')
VIEW DF	➔ df ➔ df.head(n=10)	➔ df.show() ➔ df.show(20)
COLUMNS & TYPES	➔ df.columns ➔ df.types	➔ df.columns ➔ df.types
RENAME COLUMN	➔ df.columns = ['a', 'b', 'c'] ➔ df.rename (columns= {'old':'new'})	➔ df.toDF ('a', 'b', 'c') ➔ df.withColumnRenamed ('old':'new')
DROP	➔ df.drop ('name', axis=1)	➔ df.drop ('name')
FILTERING	➔ df [df.colA >10] ➔ df [(df.colA >10) & (df.colB == 'dog')]	➔ df [df.colA >10] ➔ df [(df.colA >10) & (df.colB == 'dog')]
ADD COLUMN	➔ df['colB'] = 1/ df.mpg	➔ df.withColumn ('colB' , 1/ df.mpg)
FILL NA	➔ df.fillna(0)	➔ df.fillna(0)
AGGREGATE	➔ df.groupby (['colA', 'colB']).agg({'colC':'cat', 'colD':'dog'})	➔ df.groupby (['colA', 'colB']).agg({'colC':'cat', 'colD':'dog'})
STANDARD TRANSFORMATIONS	➔ import numpy as np ➔ df['logdisp']= np.log(df.disp)	➔ import pyspark.sql.functions as F ➔ df.withColumn('logdisp', F.log(df.disp))
CONDITIONAL STATEMENTS	➔ df['cond']=df.apply(lambda r: 1 if r.mpg >20 else 2 if r.cyl == 3 else 3, axis=1)	➔ import pyspark.sql.functions as F ➔ df.withColumn ('cond', F.when(df.mpg>20, 1).when(df.cyl

		==3,2).otherwise(3))
PYTHON WHEN REQUIRED	➔ df[‘disp1’] = df.disp.apply(lambda x: x+1)	➔ from pyspark.sql.types import DoubleType ➔ fn = F.udf.(lambda x: x+1, DoubleType()) ➔ df.withColumn(‘disp1’, fn(df.disp)
MERGE / JOIN	➔ left.merge(right, on= ‘key’) ➔ left.merge (right, left_on=’a’, right_on=’b’)	➔ left.join(right, on= ‘key’) ➔ left.join(right, left.a == right.b)
PIVOT	➔ pd.pivot_table(df, values= ‘D’ , index= [‘A’,’B’’], columns= [‘C’], aggfunc= np.sum)	➔ df.groupBy(“A”, “B”).pivot(“C”).sum(“ D”)
STATS	➔ df.describe()	➔ df.describe().show() ➔ df.selectExpr (“percentile_approx(mpg, array(.25, .5, .75)) as mpg”).show()
HISTOGRAM	➔ df.hist()	➔ df.sample(False, 0.1).toPandas().hist()
SQL	➔ NOT	➔ df.createOrReplaceT empView(‘foo’) ➔ df2= spark.sql (‘select * from foo’)

TAKE CARE OF:

- Use pyspark.sql.functions and others
- Use same version of python and packages on cluster as driver
- Check out the UI at http://localhost:4040/
- Learn about SSH port forwarding
- Check out Spark MLib

Source: Databricks
https://www.youtube.com/watch?v=XrpSRCwISdk

Sergio R.L. @saybyetogurus