MANGO SOLUTIONS

MLlib

MLlib Overview

“Apache Spark's scalable machine learning library”

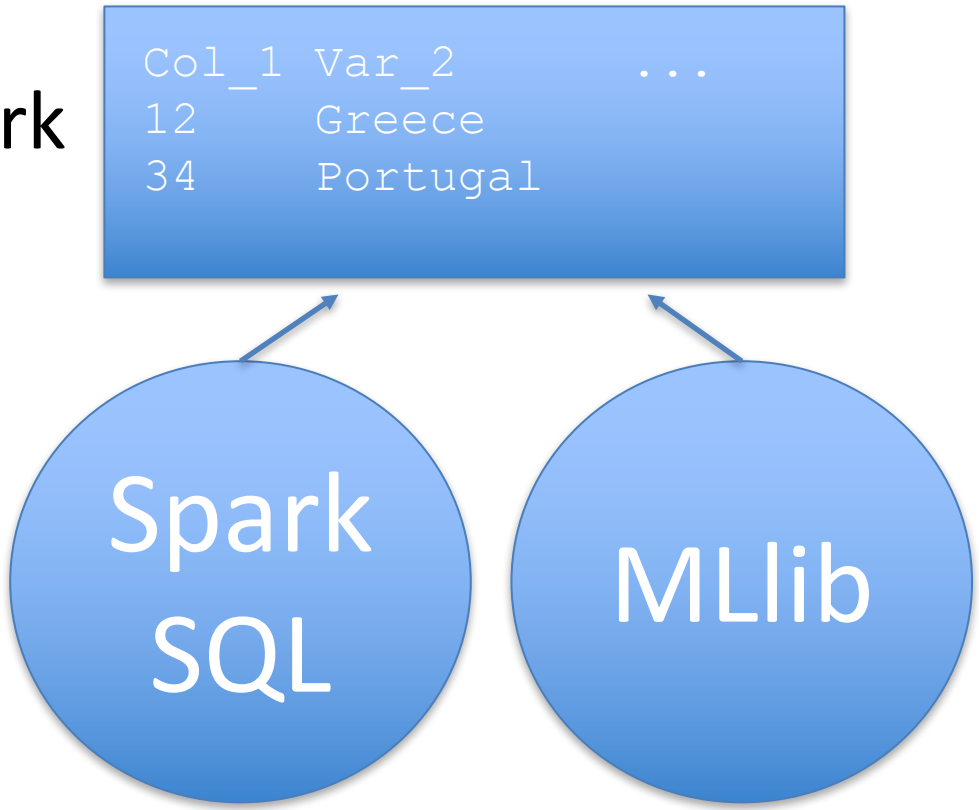
- Data Preparation
  - Feature transformations, data partition
- Machine Learning Algorithms:
  - glm, Random Forest, GBM, K-means, Naive Bayes...
- Model Evaluation

MLlib in Context

- One of the four main libraries on top of core Spark
  - **Spark SQL**
  - Spark Streaming
  - **MLlib**
  - GraphX
- Shipped with Spark 0.8 (2013)

MLlib & Spark SQL

- Completely separate way to interact with DataFrames from Spark SQL.
- We can use them together!



Aside: Spark Core Structures

- RDD: Resilent Distributed Datasets
  - The original Spark data structure
  - Immutable general purpose structure
- Dataset:
  - New in Spark 1.6
  - More performant than RDDs
- DataFrame: A Dataset of rows
  - The only structure we'll be using

MLlib in sparklyr

- Three families of functions
- Feature Transformers: `ft_`
- Spark DataFrame Manipulation: `sdf_`
- Machine Learning: `ml_`

MANGO SOLUTIONS

Data Preparation

Target Creation

- To demonstrate ft_ functions try the binarizer
- For classification can use threshold a continuous variable

```
iris_tbl <- copy_to(sc, iris)

ft_binarizer(iris_tbl,
             input.col = "Sepal_Length",
             output.col = "SL_Threshold",
             threshold = 6)
```

Using MLlib in dplyr chain

- Where `mutate` maps to Hive (Spark SQL) functions, `sdf_mutate` uses MLlib
- Easier on the eye than raw `ft_` functions

```
flights %>%
  mutate(ArrDelayDb = as.numeric(ArrDelay)) %>%
  sdf_mutate(Late = ft_binarizer(ArrDelayDb,
                                 threshold = 30)) %>%
  select(ArrDelay, Late)
```

Categorical (Factor) Data

- There is no “Factor” data type.
- Character data interpreted as factor
  - No need to convert for most ML algorithms
- Can converted using one hot encoding:
  - `ft_one_hot_encoder`
  - Or `ml_create_dummy_variables`

```
ml_create_dummy_variables(iris_tbl ,
                          input = "Species")
```

Note: Need to drop reference var

Cut (bucketizer) 1

- `ft_bucketizer` works like R's cut

```
# In R
iris %>%
  mutate(SL_Group = cut(Sepal.Length,
                        breaks = 5))
# In Spark
iris_tbl %>%
  sdf_mutate(SL_Group =
ft_quantile_discretizer(Sepal_Length,
                                n.buckets = 5))
```

Cut (bucketizer) 2

```
# In R
iris %>%
  mutate(SL_Group = cut(Sepal.Length,
                        breaks = 4:8))
# In Spark
iris_tbl %>%
  sdf_mutate(SL_Group =
ft_quantile_discretizer(Sepal_Length,
                                splits = 4:8))
```

Other `ft_` functions

- `ft_one_hot_encoder`: Encoding categoricals
- `ft_tokenizer`: Splitting text into words
- `ft_string_to_index`: Category -> index and back again
- `ft_quantile_discretizer`: Faster, non-deterministic bucketizer
- `ft_vector_assembler`: Bring various vector cols back together

Exercise

Create a categorical feature in the friday dataset for morning and afternoon departures.

(hint DepTime > 1200)

MANGO SOLUTIONS

Model Training

Available Algorithms

Function	Description
ml_kmeans	K-Means Clustering
ml_linear_regression	Linear Regression
ml_logistic_regression	Logistic Regression
ml_survival_regression	Survival Regression
ml_generalized_linear_regression	Generalized Linear Regression
ml_decision_tree	Decision Trees
ml_random_forest	Random Forests
ml_gradient_boosted_trees	Gradient-Boosted Trees
ml_pca	Principal Components Analysis
ml_naive_bayes	Naive-Bayes
ml_multilayer_perceptron	Multilayer Perceptron
ml_lda	Latent Dirichlet Allocation
ml_one_vs_rest	One vs Rest

Source: spark.rstudio.com

Final Data Prep

- Select **only** columns you want to use

```
select(friday , ArrDelay, DepTime)
```

- Last thing to do is partition into test and train

```
fl_part <- friday %>%
  select(ArrDelay, DepTime) %>%
  na.omit() %>%
  sdf_partition(train=0.8, test=0.2)
```

- Can have as many partitions as you like. Full cross validation not implement in R yet (cf H2O sparkling water).

Train the Model

- Similar to R. Can use formula interface.

```
model <- fl_part$train %>%
    ml_linear_regression(ArrDelay ~ DepTime)
```

- Can't use interaction (`y ~ x1:x2`) terms yet
- Alternatively, specify **Response** and **Features**

```
model <- fl_part$train %>%
 ml_linear_regression(response = "ArrDelay",
                      features = "DepTime")
```

Model Object

```
> class(model)
[1] "ml_model_linear_regression" "ml_model"
```

Works much like any other model

```
> summary(model)

Deviance Residuals::
     Min       1Q   Median       3Q      Max
 -81.260  -19.136   -8.682    5.231 1245.612

Coefficients:
               Estimate  Std. Error t value  Pr(>|t|)
(Intercept) -1.0703e+01  5.2566e-01 -20.361 < 2.2e-16 ***
DepTime      1.5007e-02  3.7059e-04  40.495 < 2.2e-16 ***
---
```

Scoring

- Use the `sdf_predict` function

```
scores <- sdf_predict(model,
                      newdata = fl_part$test) %>%
                      collect()
```

Elastic Net

- `ml_linear_regression` and `ml_logistic_regression` have lasso and ridge regression built in.
- Alpha: 0=ridge, 1=lasso
- Lambda: Sets strength of penalisation (use small numbers ~ 10^{-3} (0 for normal lm)
- `ml_generalized_linear_regression` works like glm

Exercise

Fit a ml_logisitic regression on the Friday dataset with a response of "Cancelled" predicted by departure time and destination state

MANGO SOLUTIONS

Model Evaluation

Cross Validation

- MLlib does have cross validation routines but not yet available from sparklyr
- Can use H2O's cross validation with rsparkling

Area Under Curve

```
pred <- sdf_predict(model_class,
                    fl_part$test)

ml_binary_classification_eval(pred,
   label = "Cancelled",
   score = "probability",
   metric = "areaUnderROC")
```

MANGO SOLUTIONS

Concluding

Workshop Takeaways

- sparklyr evolving rapidly, things may change!
- Great tool for exploratory analysis on big datasets without needing to learn Scala/Java
- Built into Rstudio and Cloudera
- Can mix MLlib and Hive commands through dplyr
- Expect lots of new features soon