Saving and Reloading Models

Lesson Objectives

After completing this lesson, you should be able to:

- Explain some benefits of saving machine learning models
- Determine if a model can be saved
- Save and reload models, when possible
- Figure out which class was used when writing out a model
- Overwrite an existing model

When is saving models useful?

- When re-training would be painful, because it takes a long time
- When you want to train many models, to be evaluated later
- When transferring between clusters, e.g. deploying a model to production

What can be saved in Spark 1.6?

- Any Transformer or Estimator class that implements MLWritable, which includes
  - Pipeline, CrossValidator
  - Some feature indexers and vectorizers
  - LinearRegression, LogisticRegression, NaiveBayes
  - KMeans, LDA, ALS
- Notably missing
  - DecisionTree, RandomForest
- When in doubt, try an example first

What can be saved in Spark 2.0?

- Much more complete coverage in Spark 2.0
  - –All languages (Scala, Java, Python, and R)
  - –Nearly all dataframe-based algorithms
  - –Models and pipelines, unfitted and fitted
- Except
  - –Python does not yet support saving CrossValidator and TrainValidationSplit
  - –Cross-language support for R needs improvement

Example - setup

```
from pyspark.ml.linalg import Vectors

df = sqlc.createDataFrame([(1.0, Vectors.dense(1.0, 2.0, 3.0)),
                           (1.0, Vectors.dense(2.0, 3.0, 4.0)),
                           (0.0, Vectors.dense(-1.0, 1.0, 2.0)),
                           (0.0, Vectors.dense(-2.0, 3.0, 5.0))]).toDF("label", "features")
```

Example – fit a model

```
from pyspark.ml.classification import LogisticRegression

lr = LogisticRegression()

lrModel = lr.fit(df)
lrModel.transform(df).show()
```

```
+-----+--------------+--------------------+--------------------+----------+
|label|      features|       rawPrediction|         probability|prediction|
+-----+--------------+--------------------+--------------------+----------+
|  1.0| [1.0,2.0,3.0]|[-18.070405604445...|[1.41945802370848...|       1.0|
|  1.0| [2.0,3.0,4.0]|[-38.987081234651...|[1.16983808020729...|       1.0|
|  0.0|[-1.0,1.0,2.0]|[19.2085506510254...|[0.99999999545187...|       0.0|
|  0.0|[-2.0,3.0,5.0]|[29.1902958840818...|[0.99999999999978...|       0.0|
+-----+--------------+--------------------+--------------------+----------+
```

Example – save and load

```
lrModel.save("lrModel.parquet")
```

```
from pyspark.ml.classification import LogisticRegressionModel

sameModel = LogisticRegressionModel.load("lrModel.parquet")
sameModel.transform(df).show()
```

```
+-----+--------------+--------------------+--------------------+----------+
|label|      features|       rawPrediction|         probability|prediction|
+-----+--------------+--------------------+--------------------+----------+
|  1.0| [1.0,2.0,3.0]||[-18.070405604445...|[1.41945802370848...|       1.0|
|  1.0| [2.0,3.0,4.0]||[-38.987081234651...|[1.16983808020729...|       1.0|
|  0.0|[-1.0,1.0,2.0]||[19.2085506510254...|[0.99999999545187...|       0.0|
|  0.0|[-2.0,3.0,5.0]||[29.1902958840818...|[0.99999999999978...|       0.0|
+-----+--------------+--------------------+--------------------+----------+
```

Which class wrote this model?

```
!cat lrModel.parquet/metadata/part-00000
```

```
{"class":"org.apache.spark.ml.classification.LogisticRegressionModel","timestamp":1475089043445,"sparkVersion":"2.0.0","u
id":"LogisticRegression_43d69252fb4e6810e320","paramMap":{"regParam":0.0,"tol":1.0E-6,"fitIntercept":true,"maxIter":100,"
standardization":true,"elasticNetParam":0.0,"probabilityCol":"probability","rawPredictionCol":"rawPrediction","featuresCo
l":"features","labelCol":"label","predictionCol":"prediction","threshold":0.5}}
```

Lesson Summary

Having completed this lesson, you should be able to:

- Explain some benefits of saving machine learning models
- Determine if a model can be saved
- Save and reload models, when possible
- Figure out which class was used when writing out a model
- Overwrite an existing model