Using Decision Tree Classifier in Apache Spark-

Dataset used in LibSVM format, split it into training and test sets, train on the first dataset, and then evaluate on the held out test set. I used two feature transformers to prepare the data; these help index categories for the label and categorical features, adding metadata to the DataFrame which the Decision Tree algorithm can recognize.

Used the pipeline in Spark for simplifying the training process. MLlib standardizes APIs

for machine learning algorithms to make it easier to combine multiple algorithms into a single

pipeline, or workflow. The pipeline concept is mostly inspired by the scikit-learn project.

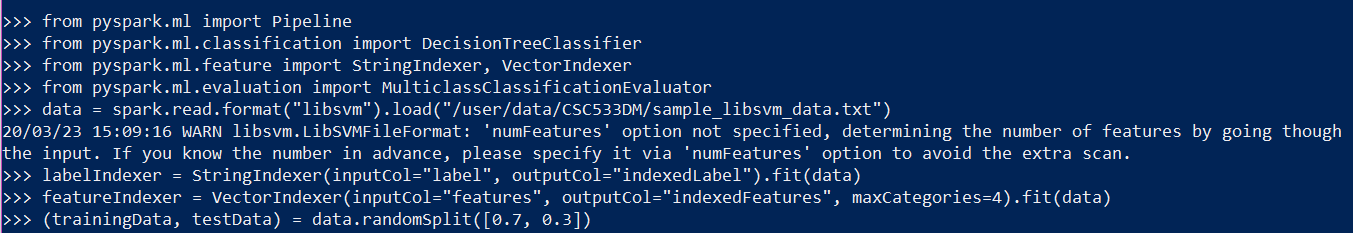
Evaluated the performance of trained model on MulticlassClassificationEvaluator. The accuracy of a classifier on a given test set is the percentage of test set tuples that are

correctly classified by the classifier. The associated class label of each test tuple is com- pared

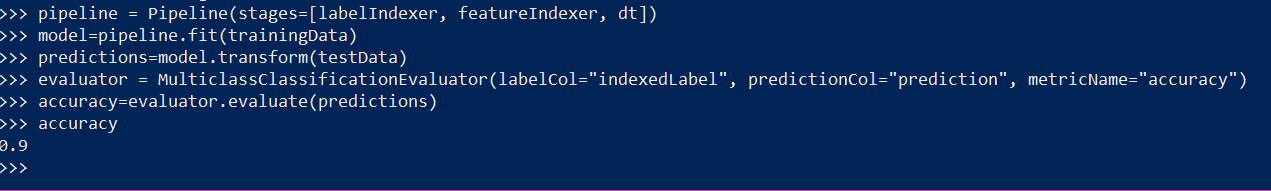
with the learned classifier’s class prediction for that tuple.

Lastly, I coded three evaluation metrics: weighted precision, weighted recall, and F-1 measure.

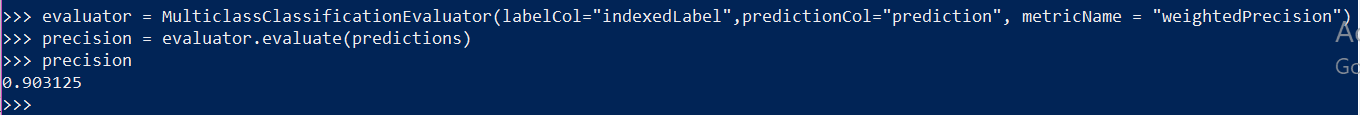
**Accuracy**



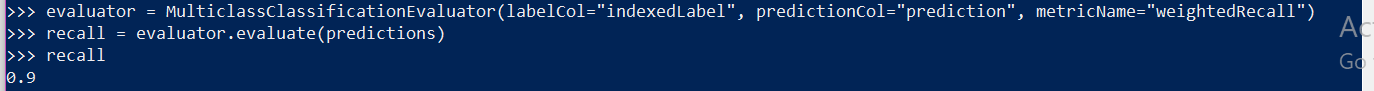




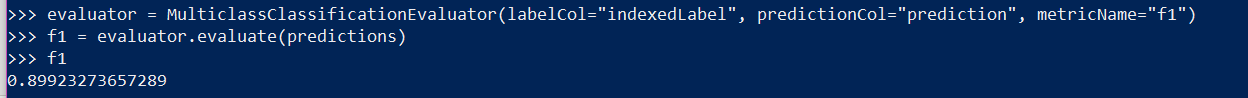
**Precision**



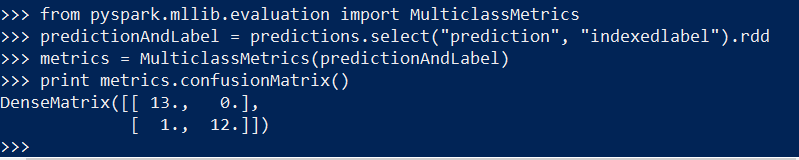
**Recall**



**F1 Score**



**OPTIONAL PART! Using an RDD based metric:**



confusionMatrix() returns a DenseMatrix with the columns representing the predicted class ordered by ascending class label, and each row represents the actual class ordered by ascending class label. The diagonal from top left to bottom right represents the observations that were predicted correctly.