ارزیابی متقابل (Cross Validation)

برای بهبود پارامترهای مدل و انتخاب بهترین پارامتر، از روش ارزیابی cross validation استفاده می‌کنیم؛ یعنی یک قسمتی از داده آموزش را به عنوان مجموعه داده ارزیابی(Validation set) جدا کرده و هربار با تغییر پارامترها، سنجه مورد نظر را بر روی مجموعه ارزیابی می‌سنجیم. سپس پارامترهایی که بالاترین مقدار سنجه را بدست آوردند انتخاب می‌کنیم. دقت کنید که در این قسمت، با مجموعه داده آزمون کاری نداریم و به اصطلاح مدل نباید داده‌های آزمون را ببیند.

برای بهبود پارامترها از روش GridSearch و برای مرحله ساخت مدل، از pipeline استفاده می‌کنیم. به عنوان نمونه برای روش رگرسیون منطقی، کد بهبود پارامتر به صورت زیر است:

model\_cs = current\_model()

pipeline = Pipeline(stages=[model\_cs])

grid = ParamGridBuilder().addGrid(model\_cs.regParam, [0.0,0.1]) \

    .addGrid(model\_cs.elasticNetParam, [0.0, 1.0])\

    .build()

evaluator = BinaryClassificationEvaluator()

cv = CrossValidator(estimator=pipeline,

                    estimatorParamMaps=grid,

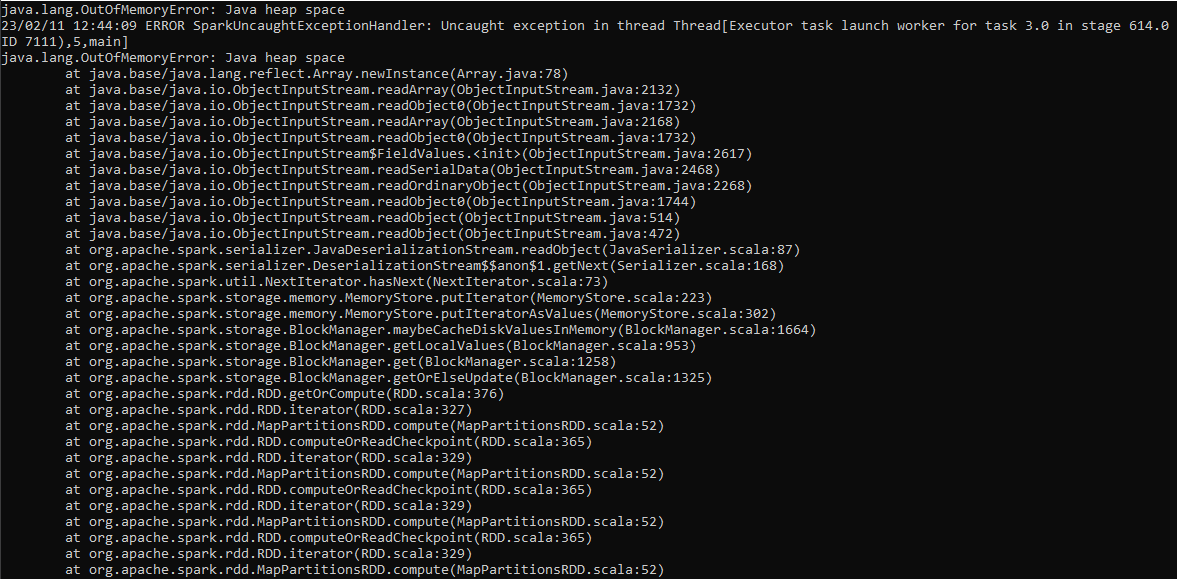
                    evaluator=evaluator,

                    numFolds=3)

cvModel = cv.fit(train)

همانطور که در کد بالا مشاهده می‌کنید، برای رگرسیون منطقی دو پارامتر regParam و elasticNetParam با مقادیر مختلف تعریف شده است. وظیفه CrossValidator، آزمایش تمام حالت‌های مختلف این پارامترها در کنار یکدیگر و ارزیابی آن‌هاست. آرگومان numFolds به معنای این است که هر بار مجموعه آموزش را به 3 قسمت تقسیم و یک قسمت آن را برای ارزیابی در نظر می‌گیرد.

با توجه به منابع محدود local، هنگام پردازش این قسمت از کد، با خطای حافظه مواجه شدیم.



کد این بخش در فایل cross\_valiation\_test.py موجود است.