**Bài thực hành số 9.**

**Phân lớp và Phân cụm**

**Link bài làm trên databricks:** [Bài làm](https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/345358637122346/4065519797493082/5017236833160125/latest.html)

**1. Dự đoán chất lượng rượu vang dựa vào các đặc trưng hóa học**

**1.1 Sử dụng lớp DecisionTreeClassifier**

**1.1.1 Dự đoán chất lượng rượu vang**

**Chuyển các cột đặc trưng thành vector**

from pyspark.ml.classification import DecisionTreeClassifier

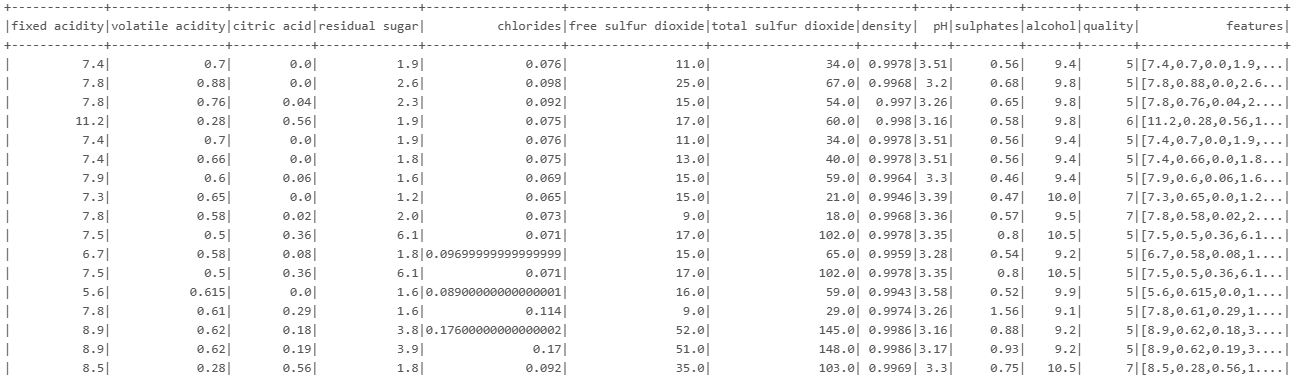
from pyspark.ml.feature import VectorAssembler

col\_assembler = df.columns[:-1]

assembler = VectorAssembler(inputCols = col\_assembler, outputCol = 'features')

df\_assembler = assembler.transform(df)

df\_assembler.show()

****

**Xây dựng mô hình Decision Tree**

data\_train, data\_test = df\_assembler.randomSplit([0.7, 0.3], seed=666)

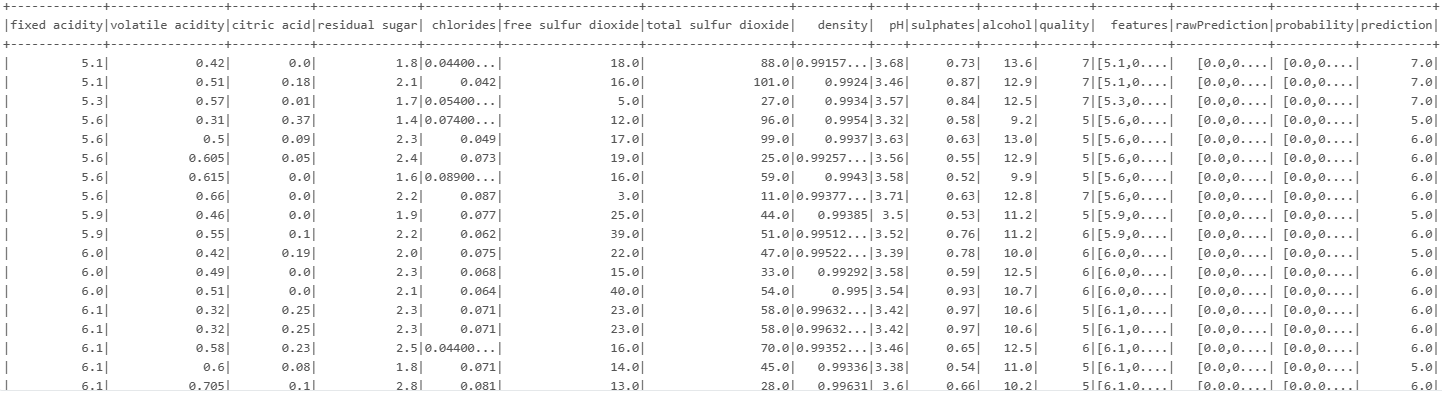
dtc = DecisionTreeClassifier(featuresCol='features', labelCol='quality')

model\_dtc = dtc.fit(data\_train)

**Thử nghiệm mô hình**

test\_model\_dtc = model\_dtc.transform(data\_test)

test\_model\_dtc.show(truncate=10)

****

**1.1.2 Đánh giá độ chính xác (accuracy) của mô hình.**

from pyspark.ml.evaluation import MulticlassClassificationEvaluator

from pyspark.ml.evaluation import BinaryClassificationEvaluator

evaluator = MulticlassClassificationEvaluator(predictionCol="prediction", labelCol="quality")

print('Độ chính xác mô hình: ', evaluator.evaluate(test\_model\_dtc)) #Độ chính xác mô hình: 0.5774

**1.1.3 Thực hiện điều chỉnh một số tham số của mô hình xem độ chính xác**

**thay đổi như thế nào?**

dtc = DecisionTreeClassifier(featuresCol='features', labelCol='quality', maxDepth=3, maxBins=5)

model\_dtc = dtc.fit(data\_train)

test\_model\_dtc = model\_dtc.transform(data\_test)

print('Độ chính xác mô hình: ', evaluator.evaluate(test\_model\_dtc)) #Độ chính xác mô hình: 0.519

=> Sau khi điều chỉnh tham số maxDepth và maxBins thì độ chính xác của mô hình giảm xuống

**1.2 Sử dụng lớp RandomForestClassifier thực hiện các thao tác tương tự trên.**

**1.2.1 Dự đoán chất lượng rượu vang**

**Xây dựng mô hình Random Forest**

from pyspark.ml.classification import RandomForestClassifier

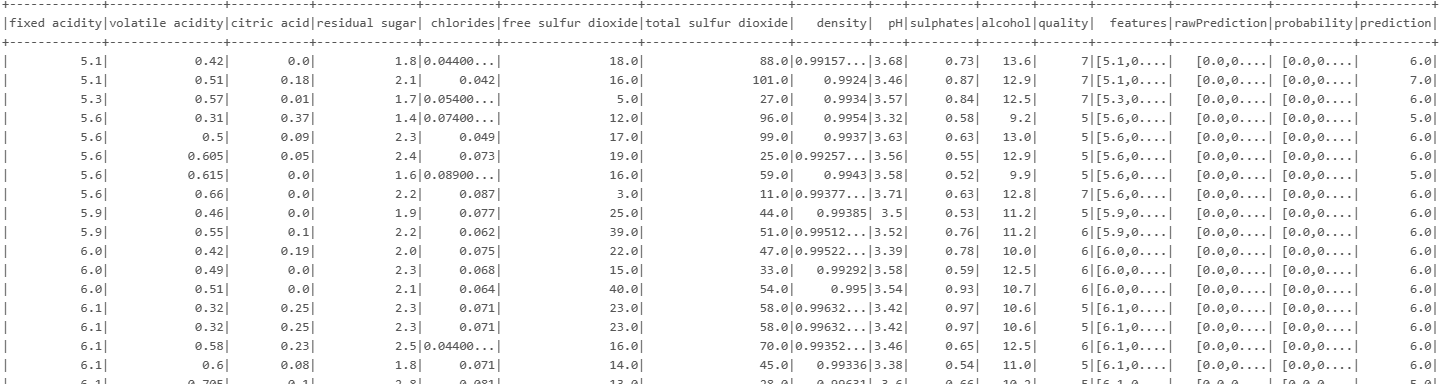
rfc = RandomForestClassifier(featuresCol='features', labelCol='quality')

model\_dtc = rfc.fit(data\_train)

**Thử nghiệm mô hình**

test\_model\_rfc = model\_rfc.transform(data\_test)

test\_model\_rfc.show(truncate = 10)

****

**1.2.2 Đánh giá độ chính xác (accuracy) của mô hình.**

print('Độ chính xác mô hình: ', evaluator.evaluate(test\_model\_rfc)) #Độ chính xác mô hình: 0.5922

**1.2.3 Thực hiện điều chỉnh một số tham số của mô hình xem độ chính xác thay đổi như thế nào?**

rfc = RandomForestClassifier(featuresCol='features', labelCol='quality', numTrees=35, maxDepth=3)

model\_rfc = rfc.fit(data\_train)

test\_model\_rfc = model\_rfc.transform(data\_test)

print('Độ chính xác mô hình: ', evaluator.evaluate(test\_model\_rfc)) #Độ chính xác mô hình: 0.530

**1.3 Thực hiện chọn 7 thuộc tính có độ ảnh hưởng cao nhất. Sau đó xây dựng mô hình phân lớp trên 7 thuộc tính này và đánh giá độ chính xác để so sánh với các phương án trên.**

**Trích chọn đặc trưng bằng ChSqSelector**

from pyspark.ml.feature import ChiSqSelector

selector = ChiSqSelector(numTopFeatures=7, featuresCol='features', outputCol='selectedFeatures', labelCol='quality')

chiSelector = selector.fit(df\_assembler)

print(chiSelector.selectedFeatures) #[1, 2, 3, 4, 7, 9, 10]

**Xây dựng mô hình Random Forest với các đặc trưng được chọn**

data\_selected = chiSelector.transform(df\_assembler)

data\_train\_s, data\_test\_s = data\_selected.randomSplit([0.8, 0.2], seed=666)

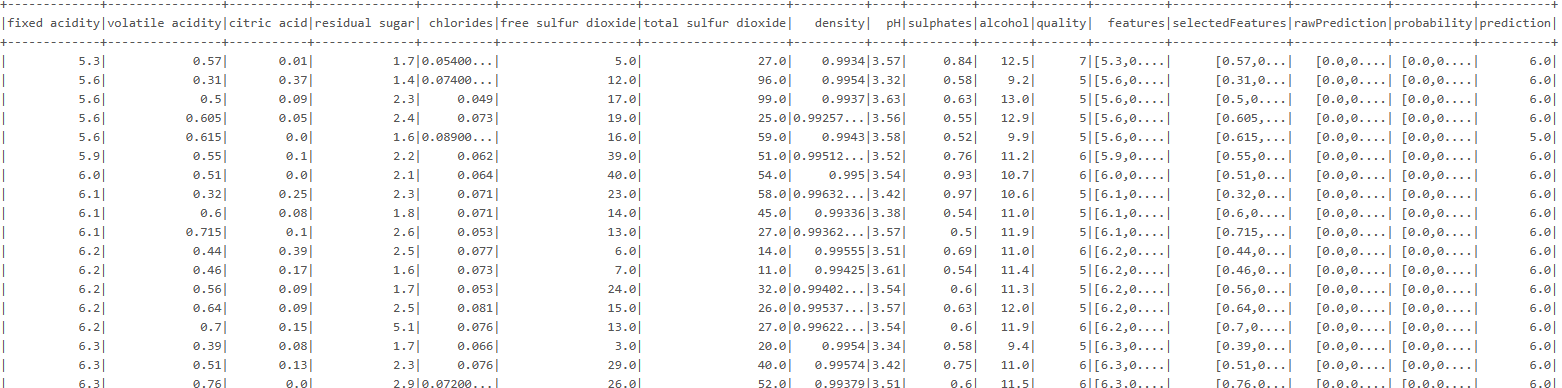
rfc\_s = RandomForestClassifier(featuresCol='selectedFeatures', labelCol='quality')

model\_rfc\_s = rfc\_s.fit(data\_train\_s)

**Thử nghiệm mô hình**

test\_model\_rfc\_s = model\_rfc\_s.transform(data\_test\_s)

test\_model\_rfc\_s.show(truncate=10)

****

print('Độ chính xác mô hình: ', evaluator.evaluate(test\_model\_rfc\_s)) #Độ chính xác mô hình:  0.58

**2. Phân cụm dữ liệu bằng lớp KMeans**

**2.1 Thực hiện phân cụm dữ liệu với số cụm là 7, 10.**

**Phân thành 7 cụm**

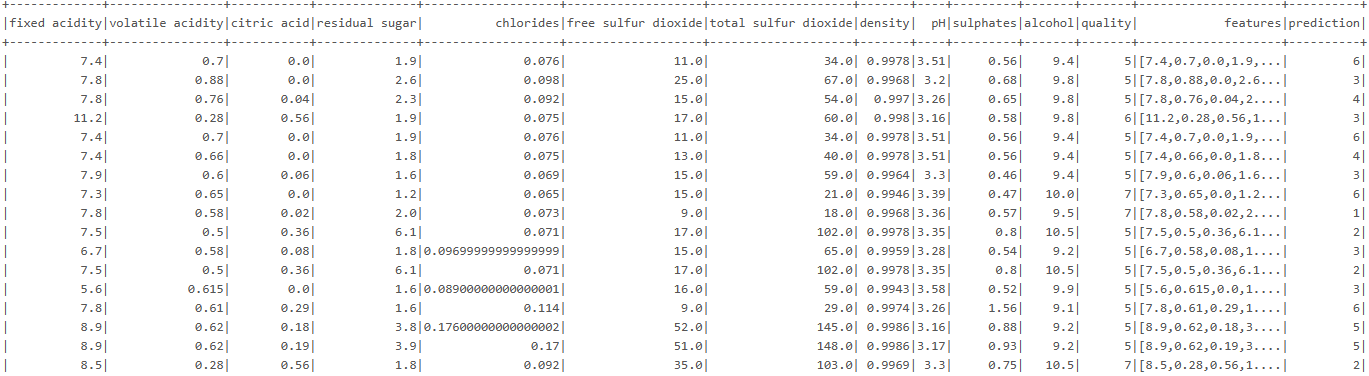
from pyspark.ml.clustering import KMeans

km = KMeans(featuresCol='features', k=7)

model\_km = km.fit(df\_assembler)

df\_cluster = model\_km.transform(df\_assembler)

df\_cluster.show()

****

**2.2 Thực hiện đánh giá chất lượng phân cụm dựa vào metric: Silhouette score và Sum of squared distances.**

from pyspark.ml.evaluation import ClusteringEvaluator

evaluator\_clus = ClusteringEvaluator(featuresCol='features', metricName='silhouette', distanceMeasure='squaredEuclidean')

print("Chất lượng khi phân 7 cụm: " , evaluator\_clus.evaluate(df\_cluster\_7))

print("Chất lượng khi phân 10 cụm: " , evaluator\_clus.evaluate(df\_cluster\_10))

****

**2.3 Xem mức độ tương đồng của phân cụm so với nhãn chất lượng đã gắn.**

Mức độ tương đồng của phân cụm so với nhãn đã gắn chỉ đạt hơn 50%.