İbrahim Can Gençel
2166429

CENG 790 BIG DATA ANALYTICS
ASSIGNMENT III - Random Forest Classifier

Part 1:

CODE =

```
// PART 1 //
// Vector Assembler
val vector_assembler = new VectorAssembler().
 setInputCols(
   Array("balance", "duration", "history", "purpose", "amount", "savings",
"employment", "instPercent", "sexMarried", "guarantors", "residenceDuration",
"assets", "age", "concCredit", "apartment", "credits", "occupation",
"dependents", "hasPhone", "foreign")).setOutputCol("all_features")
```

Vector Assembler

Part 2:

CODE =

I do not think this part was critical for this assignment, because creditability column already were in integers.

```
// PART 2 //
val creditability_indexer = new StringIndexer()
 .setInputCol("creditability")
 .setOutputCol("creditabilityIndex")
```

String Indexer

Part 3:

CODE =

```
// PART 3 //
// Train test split with ratio %90 to %10 and seed was set
// in order to get same test and train set in each run
val Array(train_set, test_set) = creditDF.randomSplit(Array[Double](0.9, 0.1),
seed = 18)
```

Train Test Random Split

İbrahim Can Gençel
2166429

Part 4:

CODE =

```
// PART 4 //
// train Random Forest model with training data set
val rfc = new RandomForestClassifier()
 .setSeed(1234)
 .setFeaturesCol("all_features")
 .setLabelCol("creditabilityIndex")
 .setFeatureSubsetStrategy("auto")

// grid search parameters are given
val grid_search = new ParamGridBuilder()
 .addGrid(rfc.maxDepth, Array(4, 6, 8))
 .addGrid(rfc.maxBins, Array(25, 28, 31))
 .addGrid(rfc.impurity, Array("entropy", "gini"))
 .build()
```

Model Description and Grid Search Parameters

Part 5:

CODE = I could not find the best model's parameters because I have used pipeline.

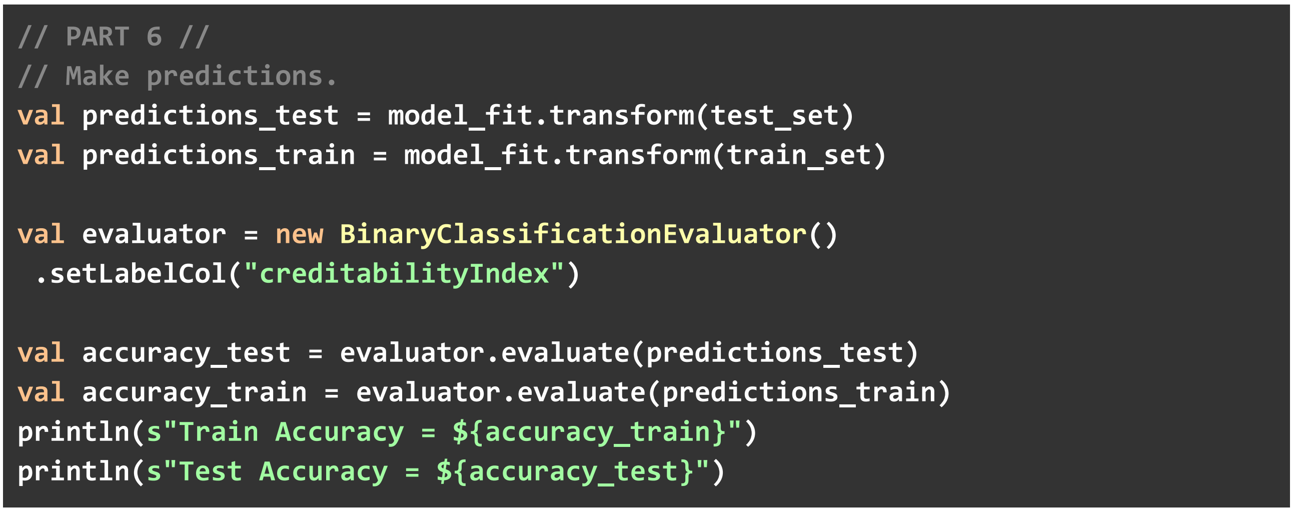
```
// PART 5 //
// Pipeline object is created
val pipeline = new Pipeline()
 .setStages(Array(vector_assembler, creditability_indexer, rfc))

// Creating train validation split
 // BinaryClassificationEvaluator is used
 // 0.75 ratio means 3x -> train set x -> validation set
val train_val_split = new TrainValidationSplit()
 .setEstimator(pipeline)
 .setEstimatorParamMaps(grid_search)
 .setEvaluator(new
BinaryClassificationEvaluator().setLabelCol("creditabilityIndex"))
 .setTrainRatio(0.75)

// Training the model
val model_fit = train_val_split.fit(train_set)

// Getting the best parameters
// I could not find the best parameters because I have used Pipeline
println(model_fit.bestModel.extractParamMap())
```

Pipeline and Training Model

İbrahim Can Gençel
2166429

Part 6:

CODE = I could not find the best model’s parameters because I have used pipeline.

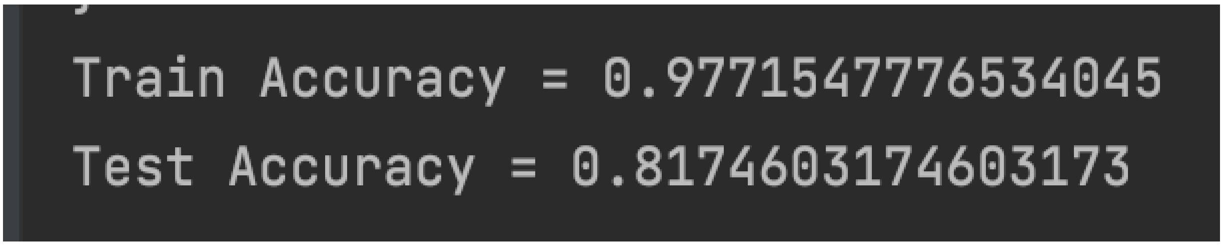


```
// PART 6 //
// Make predictions.
val predictions_test = model_fit.transform(test_set)
val predictions_train = model_fit.transform(train_set)

val evaluator = new BinaryClassificationEvaluator()
 .setLabelCol("creditabilityIndex")

val accuracy_test = evaluator.evaluate(predictions_test)
val accuracy_train = evaluator.evaluate(predictions_train)
println(s"Train Accuracy = ${accuracy_train}")
println(s"Test Accuracy = ${accuracy_test}")
```

Making Predictions

RESULTS =

Train accuracy was more than 97% where test accuracy was more than 81%; both of them are good therefore, I am satisfied with the model and its outputs.



```
Train Accuracy = 0.9771547776534045
Test Accuracy = 0.8174603174603173
```