Use Case – Klassifikation mit RapidMiner

Wir haben bereits einige Fragestellungen kennengelernt, welche mit einer Klassifikation gelöst werden können beispielsweise die Bestimmung von Kunden, welche wahrscheinlich ihren Vertrag kündigen werden. In RapidMiner können Sie dies ähnlich leicht wie eine Regression durchführen.

Laden des Datasets

1. Ziehen Sie das „iris“-Dataset in den Prozessbereich.
2. Verbinden Sie das Dataset mit dem „res“-Knoten und schauen Sie es sich genauer an.

Welche Schritte sollten wohl vor einer Klassifikation noch getroffen werden und welches Attribut wird Ihre Zielvariable?

Daten vorbereiten

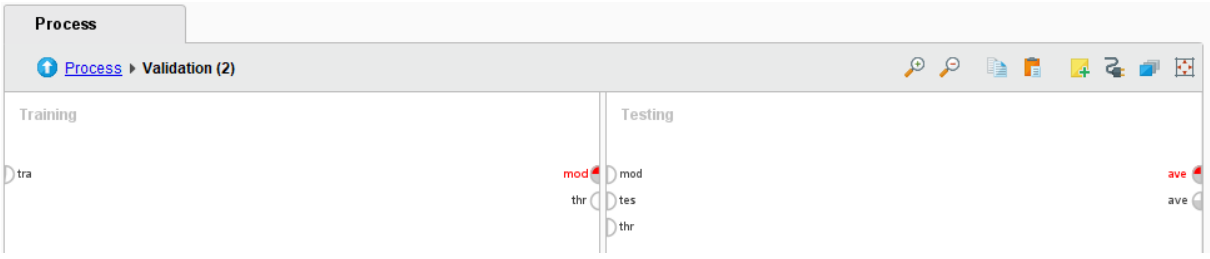
Wie Sie sicher schon vermutet haben sollten Sie vor dem Training eine Normalisierung durchführen, damit die unterschiedliche Dimensionierung bei der Klassifizierung keinen Einfluss auf die Gewichtungen hat.

1. Ziehen Sie den „Normalize“-Operator in den Prozessbereich.
2. Verbinden Sie das Dataset mit dem Operator.
3. Wählen Sie mithilfe der Parameter „attribute filter type“ und „attributes“ die umzuformenden Attribute aus.
4. Stellen Sie als „methode“ → Z-transformation ein.
5. Ziehen Sie den Operator „Set Role“ in den Prozessbereich
6. Verbinden Sie den „Set Role“-Operator mit dem vorherigen Operator.
7. Wählen Sie als „target role“ → label und wählen Sie bei „attribute name“ Ihre Zielvariable aus.

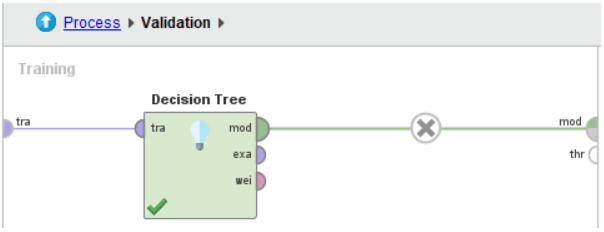
Klassifikation mit zusätzlicher Split-Validierung

Ebenso wie bei der Regression wird hier wieder mit einer Split-Validierung gearbeitet.

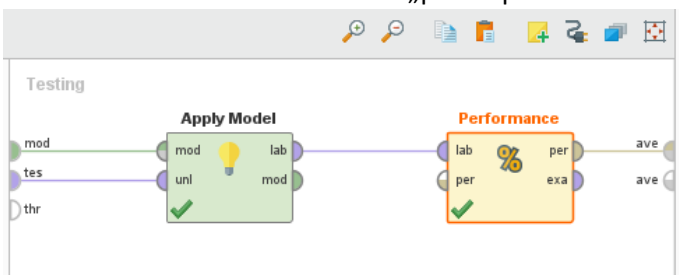
1. Suchen Sie bei den Operatoren den „Split Validation“-Operator und ziehen diesen in den Prozessbereich.
2. Verbinden Sie den „exa“-Knoten von „Set Role“ mit dem „tra“-Knoten des „Split Validation“-Operators.
3. Klicken Sie nun mit einem Doppelklick auf den Prozess. Sie sollten folgenden Prozess sehen.



4. Ziehen Sie nun den „Decision Tree“-Operator in den Trainingsteil des Prozesses und verbinden Sie die „tra“ und „mod“-Knoten miteinander.
5. Wählen Sie als Parameter „max-depth“ → 10 und als „main criterion“ den bekannten „information_gain“.



6. Wählen Sie den „Apply Model“-Operator aus und ziehen diesen in den Testteil des Prozesses. Verbinden Sie hier das zuvor trainierte Model „mod“ mit dem „mod“-Knoten des „Apply Model“-Operators. Nun müssen Sie noch die Daten auswählen, welche vorhergesagt werden sollen, in diesem Fall die Testdaten. Verbinden Sie hierfür den „tes“-Knoten und den „unl“-Knoten.
7. Ziehen Sie nun den „Performance(Classification)“-Operator in den Prozessbereich und verbinden den „lab“-Knoten mit dem „lab“-Knoten des Operators.
8. Verbinden Sie zum Abschluss den „per“-Operator mit dem „ave“-Operator.



9. Verlassen Sie den Prozess über den blauen Pfeil neben dem Prozess Symbol in der oberen linken Ecke des Prozessbereichs.
10. Passen Sie nun die „split ratio“ des „Validation“-Operators auf 0.8 an und geben Sie die Knoten „mod“ und „ave“ aus.

Auswertung

accuracy: 96.67%

	true Iris-setosa	true Iris-versicolor	true Iris-virginica	class precision
pred. Iris-setosa	10	0	0	100.00%
pred. Iris-versicolor	0	10	1	90.91%
pred. Iris-virginica	0	0	9	100.00%
class recall	100.00%	100.00%	90.00%	

Das was Sie hier sehen wird „Confusion-Matrix“ genannt und verdient seinen Namen für Neulinge auf jeden Fall 😉

Es werden sogenannte true-positives und false-positives dargestellt werden. Was dies genau bedeutet geht über diese Schulung hinaus. Für das Erste reicht es wenn Sie wissen, dass die Grafik oben zeigt, dass 1 sample welches eigentlich zur Klasse Iris-virginica gehört als Iris-versicolor vorhergesagt wurde.

Beeinflussen Sie nun die „split-ratio“ Ihres „Performance“-Operators. Was stellen Sie fest?

Was passiert wenn Sie als max_depth → 100 oder nur 1 einstellen?

Schauen Sie sich den Tree näher an, welches Label lässt sich wohl am einfachsten vorhersagen?