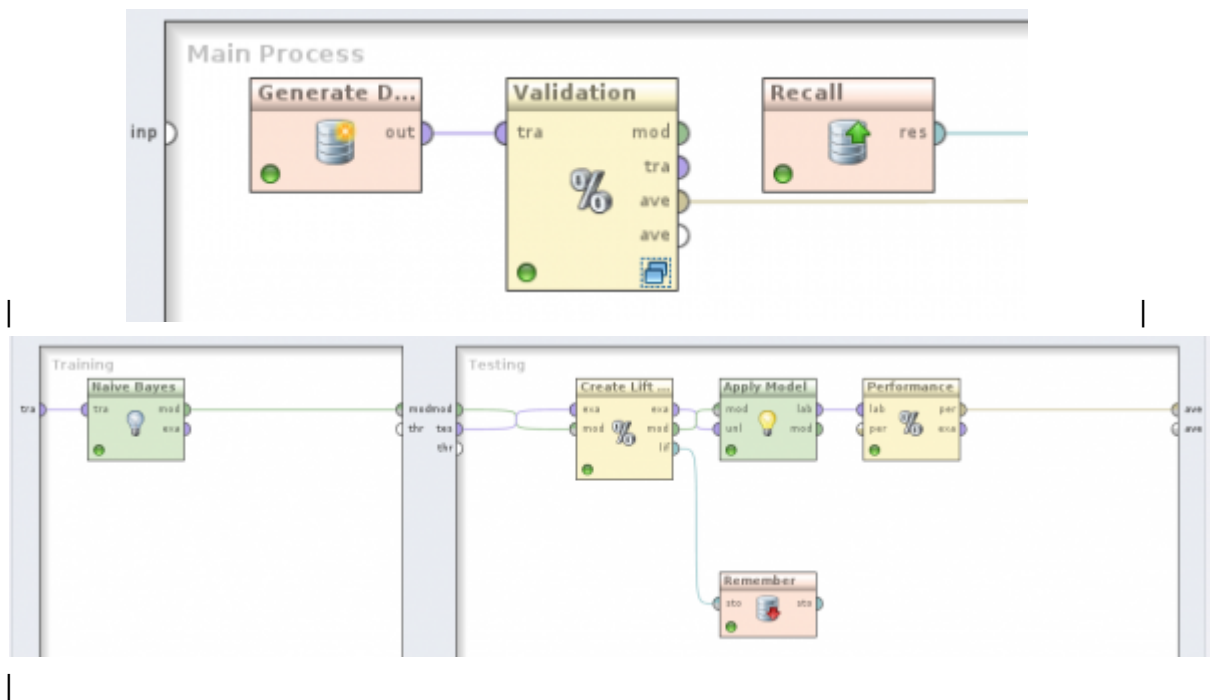


Support Vector Machines

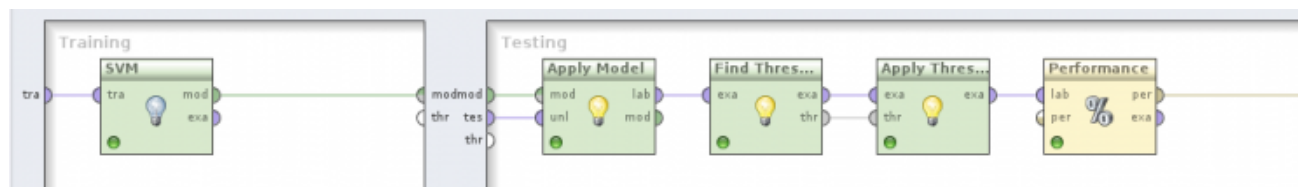
Zapoznaj się z krótką [wizualizacją głównej koncepcji](#) stojącej za rodziną algorytmów SVM.

Rapid Miner

- Uruchom narzędzie RapidMiner
- Utwórz przepływ polegający na wczytaniu pliku `bank.csv` i zbudowaniu klasyfikatora SVM. Zapoznaj się z [opisem zbioru danych](#). Wczytaj dane za pomocą operatora `Read CSV` i prześlij je do operatora `Nominal to Numerical`, zamieniając atrybuty nominalne na numeryczne zgodnie z metodą *unique integers*. Wewnątrz operatora `Split Validation` umieść operator `SVM` i wykorzystaj początkowo kernel *dot* (możesz też poeksperymentować z innymi rodzajami kerneli). Obejrzyj uzyskany model. Zwiększ wartość parametru `C` zezwalając na pewną elastyczność granicy decyzyjnej. Możesz manipulować modelem także za pomocą parametrów `L pos` i `L neg`, które działają analogicznie do funkcji kosztu.
- Co zrobić w przypadku gdy klasyfikowany zbiór danych ma więcej niż dwie klasy? Załaduj plik `letter-recognition.csv` i zapoznaj się z [jego opisem](#). Ponieważ plik jest dość duży, użyj operatora `Sample (Stratified)` do wylosowania 10% danych. Prześlij je do operatora `Split Validation`, wewnątrz którego umieść operator `Polynomial by Binomial Classification` wykorzystujący we wnętrzu operator `SVM`. Sprawdź, która z metod (*1-against-all*, *1-against-1*, *ECOC*) daje najlepsze rezultaty.
- Utwórz przepływ obrazujący wykorzystanie krzywej *lift*. Użyj operatora `Generate Direct Mailing Data` w celu wygenerowania 1000 syntetycznych rekordów. Prześlij wygenerowany zbiór uczący do operatora `Split Validation`. Wewnątrz operatora `Split Validation` wykorzystaj w fazie uczenia operator `Naive Bayes`, natomiast w fazie testowania umieść kolejno: operator `Create Lift Chart`, operator `Apply Model` oraz operator `Performance (Classification)`. Dodatkowo, port `lift` operatora `Create Lift Chart` prześlij do operatora `Remember` i zapisz w repozytorium pod dowolną nazwą (jako wartość `io object` wybierz *LiftParetoChart*). W procesie nadrzędnym dodaj operator `Recall` i odczytaj obiekt zapisany w trakcie walidacji modelu, wysyłając go jako końcowy wynik przepływu. Jeśli chcesz, możesz też wyświetlić zbiór uczący (port `tra` operatora `Validation`). Aby upewnić się, że operatory są wykonywane we właściwej kolejności, wybierz z menu opcję `Proces - Operation Execution Order - Order Execution` i w razie konieczności popraw przepływ: walidacja klasyfikatora musi się odbyć przed operatorem `Recall`.



- Utwórz podstawowy przepływ obrazujący wykorzystanie krzywej ROC. Odczytaj z repozytorium zbior danych Sonar i wyślij go do operatora Split Validation. Wewnątrz operatora do walidacji w fazie uczenia wykorzystaj operator Support Vector Machine (LibSVM), a w fazie testowania użyj kolejno: operatora Apply Model, operatora Find Threshold, operatora Apply Threshold, a w końcu operatora Performance.



- Ostatnim przepływem jest przepływ obrazujący możliwość jednoczesnego porównania wielu krzywych ROC pochodzących od różnych klasyfikatorów. Odczytaj zbiór danych Sonar, a następnie dodaj operator Add Noise. Nie dodawaj nowych atrybutów, ale wprowadź 5% szum we wszystkich atrybutach. Tak zmodyfikowany zbiór uczący wyślij do operatora Compare ROCs. Wewnątrz tego operatora umieść operatory Naive Bayes, Decision Tree oraz Support Vector Machines. Port roc operatora Compare ROCs wyślij jako wynik przepływu. Obejrzyj i porównaj krzywe ROC klasyfikatorów.