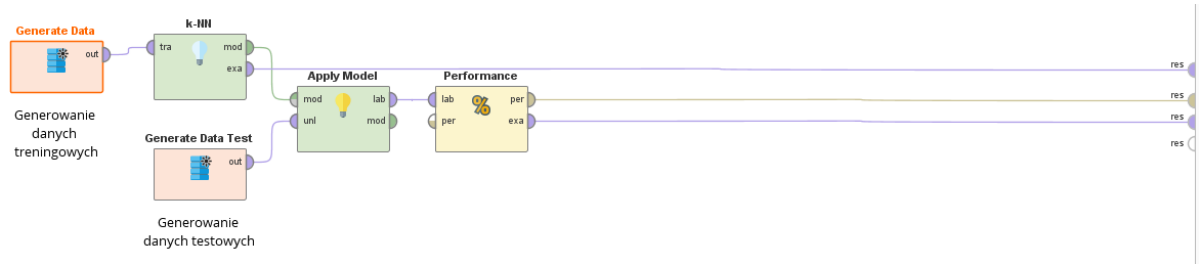
Wpływ danych na jakość modeli predykcyjnych

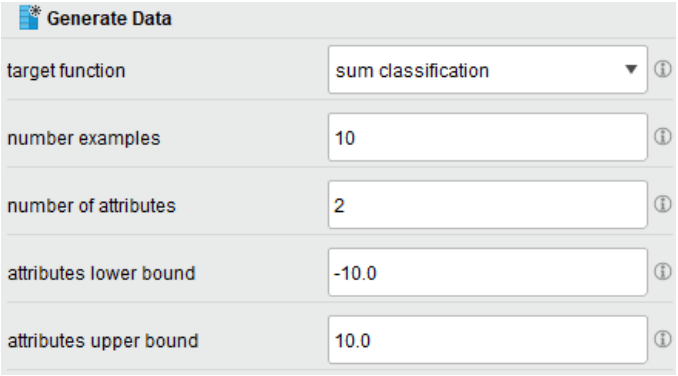
Modele predykcyjne to zbiór algorytmów których pracę można podzielić na dwa etapy. W pierwszym następuje uczenie modelu, czyli na podstawie znanych par <X,y> następuje uczenie modelu zapisanego jako y=M(X) gdzie M( ) jest modelem. W procesie uczenia dany algorytm dokonuje adaptacji parametrów modelu M( ), tak aby ten jak najdokładniej mógł dokonać predykcji. W drugim etapie, dysponując nauczonym modelem M( ) oraz znając X będziemy chcieli przewidywać wartości y.

Na proces ten istotnie wpływa rozkład danych oraz ich liczebność – tj. iloma danymi dysponujemy na potrzeby uczenia modelu.

Aby zobaczyć jak powyższe wpływa zbuduj proces jak poniżej



Dla operatora ***Generate Data*** ustaw wartości jak poniżej

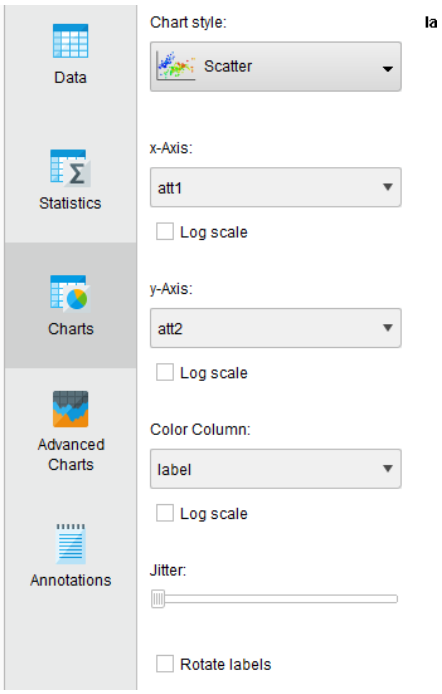


Dla operatora ***Generate Data test*** ustaw jego parametry tak jak powyżej, z tą różnicę iż liczbę *numer of examples* ustaw na 9000

W operatorze *k-NN* ustaw k=1

Wykonaj powyższy proces stopniowo zwiększając *numer of examples* w operatorze ***Generate Data*** do wartości [10, 30, 50, 100, 300, 1000]

Zarejestruj dokładność klasyfikacji oraz rozkład danych w postaci obrazka. Zwróć uwagę jak oryginalne dane wyglądały. Oryginalne dane możesz zobaczyć wstawiając pole Color Column na label, a wyniki predykcji poprzez ustawienie Color Column = prediction(label)

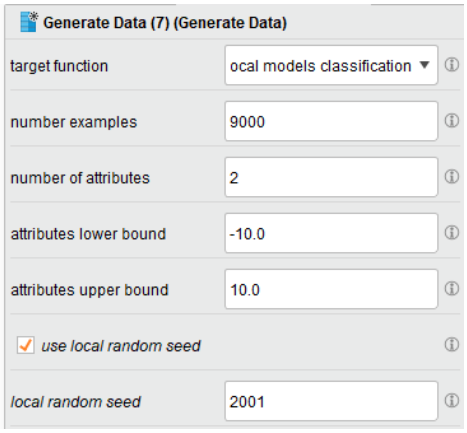


Jak ilość danych wpływa na dokładność modelu?

Wpływ stabilności rozkładu

Należy pamiętać iż w przypadku systemów uczących się przyjmuje się założenie, iż dane na podstawie których uczymy model i dla których stosujemy nasz model pochodzą z tego samego rozkładu prawdopodobieństwa oraz poszczególne wektory są względem siebie niezależne tj. wektor x_i nie zależy od x_{i-1}

Sprawdź co się dzieje gdy dane ulegają zmianie. W procedurze powyżej zmień ustawienia bloczka Generate Data i ustaw wartość ***numer of examples*** na tą która pozwalała ci uzyskać najlepszy wynik w poprzednich eksperymentach. Ponadto dokonaj zmiany w ustawieniach operatora Generate Data Test na:



Zanotuj dokładność i porównaj z dokładnością uzyskaną na wcześniej używanych danych.

Podaj przykłady realnego problemu klasyfikacyjnego i dla podanych przykładów (podaj dwa) wskaż co może spowodować zmianę rozkładu danych podczas użytkowania systemu.

Zaproponuj procedurę która pozwoli na weryfikację czy dane na których system był uczony ulegają zmianie czy też nie?

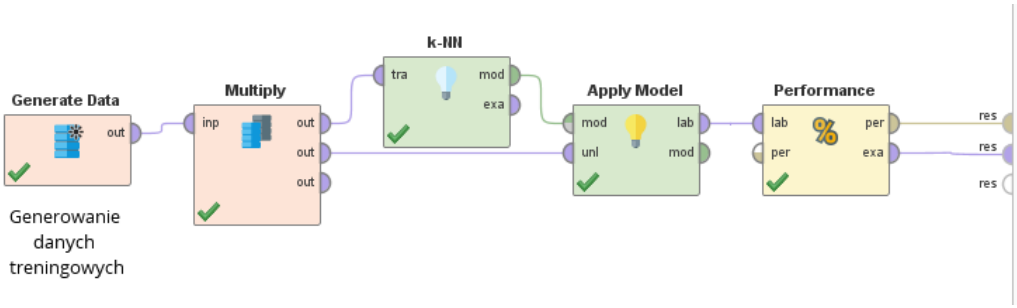
Zaproponuje co należałoby zrobić gdy zaobserwujesz zmianę rozkładu danych.

Szacowanie dokładności modeli predykcyjnych

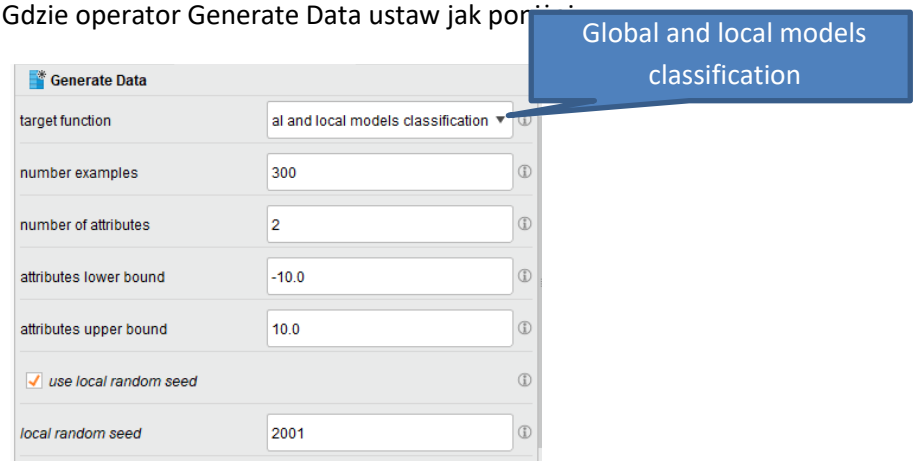
Problem, z którym zwykle spotykamy się budując model predykcyjny to niedobór danych, gdyż często pozyskanie dobrej jakości etykiet wiąże się z dużym kosztem. Dlatego też pojawia się problem oceny jakości modeli predykcyjnych, gdyż z jednej strony chcemy wiedzieć czy nasz model czegokolwiek się nauczył czy też nie, a z drugiej mamy do dyspozycji jeden zbiór danych. Zachodzi więc pytanie jak weryfikować to, czego nasz model się nauczył. Jak więc oceniać model:

1) Możemy nauczyć model na tych samych danych i na tych samych danych poddać go ocenie.

W tym celu zbuduj proces jak poniżej



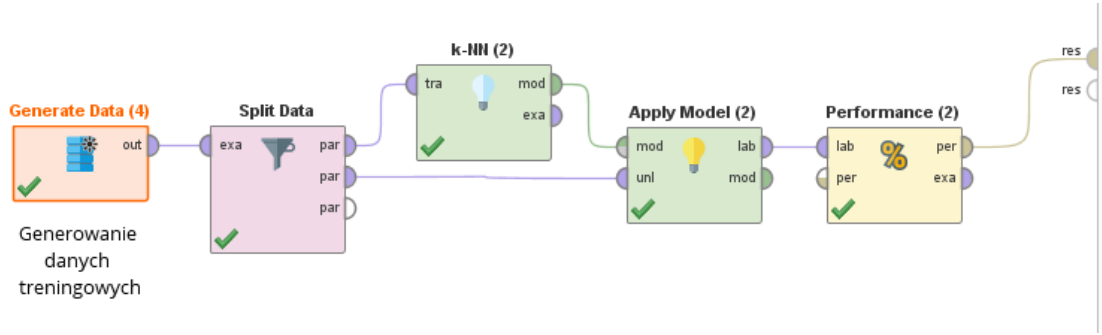
Gdzie operator Generate Data ustaw jak poni



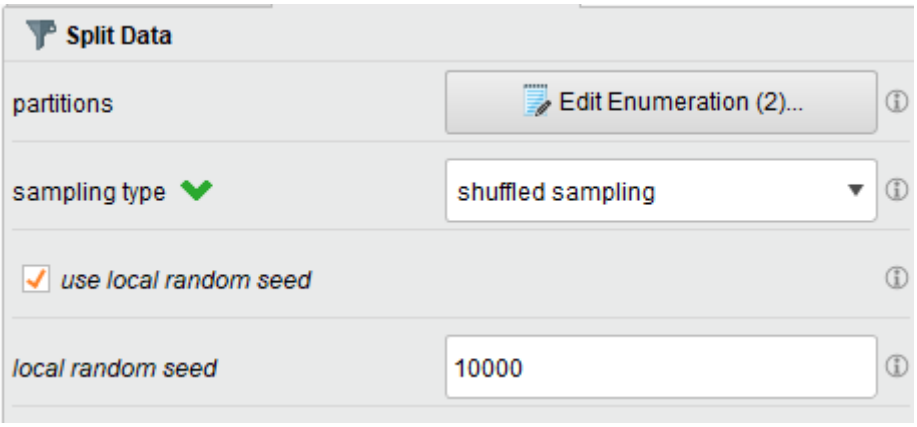
Zanotuj dokładność predykcji.

Jak myślisz dlaczego wyszła taka dokładność?

2) Innym sposobem jest podział danych na dwa podzbiory – jeden na którym model jest uczony, drugi na którym model jest testowany, a podział odbywa się w sposób losowy. W celu zbadania takiego rozwiązania zbuduj proces jak poniżej:



Gdzie operator **Generate Data** ma identyczną konfigurację jak powyżej, natomiast operator **Split Data** służy do podziału zbioru danych na dwa podzbiory. Podziel je w stosunku 70% (0.7) do 30% (0.3). Jednocześnie ustaw **sampling type** na ***Shuffled sampling***. Patrz poniżej



Dokonaj 10'cio krotnego powtórzenia obliczeń każdorazowo zmieniając ustawienia *local random seed*

Czy wyniki z poszczególnych prób różnią się między sobą?

Jak więc wybrać właściwą wartość (najlepszy/najgorszy/?) (zastanów się i zaproponuj rozwiązanie które pozwoli ocenić dokładność predykcji)

3) Kolejnym rozwiązaniem stosowanym do oceny jest identyczne jak powyżej, z tą różnicą iż zmianie ulega tyb próbkowania z *shuffled sampling* na *stratified sampling*.

Podobnie jak powyżej 10'ci krotnie powtórz obliczenia każdorazowo notując jaki jest wynik, przy czym każdorazowo zmieniaj wartość *local random seed* Możesz użyć tych samych wartości jak w podpunkcie 2.

Czy wyniki się różnią w stosunku do zadania 2?

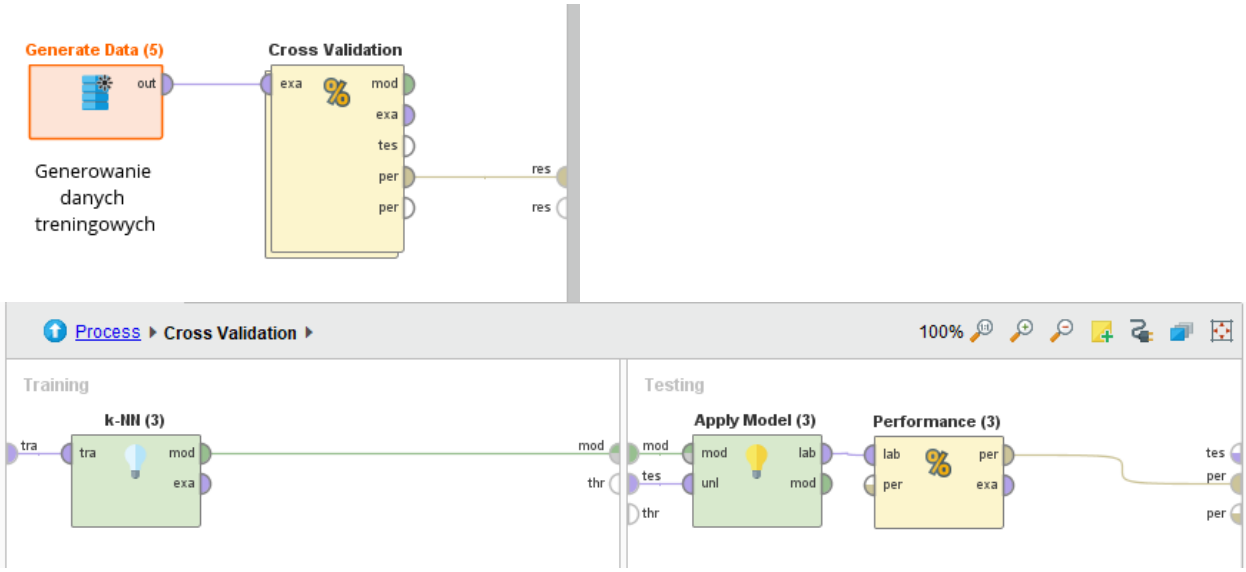
Skąd się bierze różnica?

Jaka jest więc dokładność naszego modelu na podstawie dotychczasowych obliczeń – podaj ją.

4) Rozwiązanie to polega na wykorzystaniu testu krzyżowego (ang. Cross Validation). Procedura ta dzieli zbiór danych na k (zwykle *k*=10) równych i rozłącznych kawałków, a następnie w pętli *k* razy

przeprowadza proces uczenia i wybierając do tego *k*-1 kawałków oryginalnego zbioru danych i testując na jednym (pozostałym). W każdej iteracji pętli wybierany jest inny kawałek do testowania.

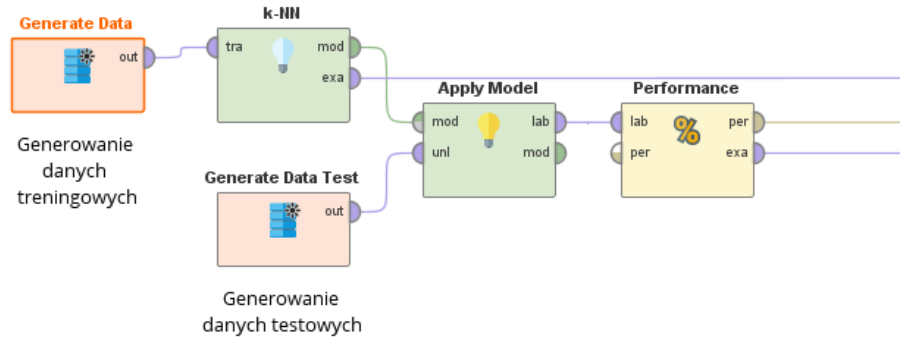
W celu weryfikacji tego rozwiązania zbuduj proces jak pokazano poniżej:



Uruchom proces. Dodatkowo możesz zmienić parametr *use local random seed* i sprawdzić jak zachowują się wyniki dla różnych wartości parametru *local random seed*

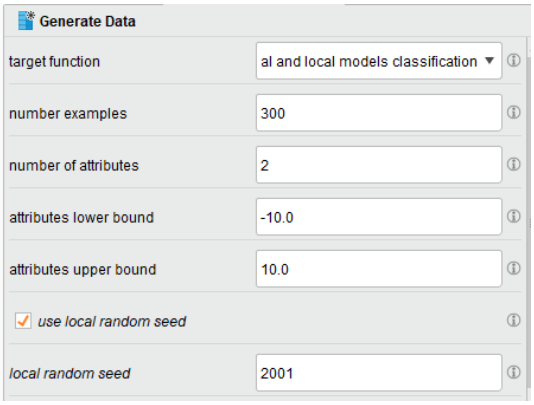
W celu weryfikacji która z metod oceny dała najbardziej wiarygodny wynik potrzebujemy prawdziwego zbioru testowego, który będziemy mogli wykorzystać do oceny jakości naszego modelu.

W tym celu zbuduj proces:



Gdzie:

Bloczek Generate Data jest identyczny we wszystkich eksperymentach,



A operator Generate Data Test w pozycji *numer of examles* ma wartość 9000

Ponieważ rozważamy problem sztuczny, więc możemy wygenerować sobie dowolnie duży zbiór testowy z tego samego rozkładu danych i podać go ocenie przez nasz klasyfikator. Najlepsza będzie ta metoda, która uzyska wynik najbardziej zbliżony do wyników z tego doświadczenia.