Rozdział 5

Moduł analityczny

Adam Gudyś, Andrzej Janusz, Marek Sikora, Tomasz Stęclik,
Dominik Ślęzak, Łukasz Wróbel

5.1 Wstęp

Ważnym składnikiem systemu DISESOR (systemu opracowanego w ramach projektu) jest moduł analityczny. Za pomocą modułu analitycznego możliwe jest tworzenie różnego rodzaju systemów prognostycznych, realizujących zadania klasyfikacji i regresji oraz odkrywanie zależności w danych. Moduł analityczny w największym stopniu — w porównaniu z innymi modułami systemu DISESOR — wykorzystuje wbudowane funkcje środowiska RapidMiner [5.7]. Środowisko RapidMiner jest oprogramowaniem dedykowanym do analizy danych, a w szczególności ich eksploracji w celu odkrywania zależności nieznanych i interesujących dla użytkownika.

Środowisko RapidMiner zostało w czasie realizacji projektu całkowicie dostosowane do jego celów. Dostosowanie to obejmowało również opracowanie polskiej wersji tego oprogramowania. System DISESOR widoczny jest w środowisku RapidMiner jako dodatkowa grupa komponentów, które można wykorzystać i budować z nich złożone schematy analizy i wnioskowania.

W założeniu, moduł analityczny dedykowany jest dla dwóch grup użytkowników. Pierwszą z nich stanowią użytkownicy zaawansowani, którzy znakomicie orientują się w tematyce analizy i eksploracji danych oraz w środowisku RapidMiner. Ta grupa użytkowników w pełni wykorzystuje wszystkie możliwości RapidMiner'a: analiza danych odbywa się za pomocą interfejsu użytkownika udostępnianego przez to oprogramowanie lub realizowana jest na poziomie kodu źródłowego tworzonego przez analityków należących do tej grupy. Druga grupa to użytkownicy mniej zaawansowani, którzy podczas analizy danych potrzebują wsparcia w postaci różnego rodzaju funkcji ułatwiających realizację procesu analizy i weryfikacji uzyskanych wyników.

Zastosowanie środowiska RapidMiner jako jądra systemu DISESOR gwarantuje grupie użytkowników zaawansowanych dostęp do szerokiego spectrum metod oczyszczania danych, uzupełniania brakujących wartości, ekstrakcji i selekcji

cech, rozwiązywania problemów klasyfikacyjnych i regresyjnych, analizy asocjacji, grupowania danych oraz zaawansowanej walidacji uzyskanych wyników obejmującej m.in. różne techniki oceny eksperymentalnej i szerokie spectrum testów statystycznych. Podczas realizacji modułu analitycznego główny nacisk położono zatem na rozszerzenie funkcjonalności środowiska RapidMiner o nowe, dotychczas niedostępne funkcje analityczne oraz o zgrupowanie i opakowanie podstawowych operatorów z jakich budowane są zaawansowane schematy analizy, aby maksymalnie zautomatyzować proces analizy danych.

Dwie główne grupy nowych metod opracowanych i zaimplementowanych w ramach realizacji modułu analitycznego to: algorytmy bazujące na teorii zbiorów przybliżonych oraz algorytmy indukcji reguł bazujące na paradygmacie sekwencyjnego pokrywania. W przypadku indukcji reguł opracowano uniwersalny operator umożliwiający indukcję reguł klasyfikacyjnych, regresyjnych i tzw. wzorców przeżycia, które mogą być wykorzystane np. w analizie niezawodności. Dla grupy użytkowników niezaawansowanych opracowano swego rodzaju asystenta analizy, którego funkcjonalność obejmuje m.in. w pełni automatyczną realizację zadań klasyfikacyjnych, regresyjnych i analizy asocjacji. W module analitycznym dostępne są również dwa operatory związane z bazowym obszarem zastosowania systemu DISESOR — górnictwem węgla kamiennego. Oba operatory służą do rozwiązania zadań tworzenia modeli prognostycznych przeznaczonych do zadań prognozowania stężenia metanu i zagrożenia sejsmicznego. Metody analityczne realizowane przez te operatory (patrz rozdział 11) są efektem przeprowadzonych międzynarodowych konkursów. Konkursy zrealizowano w ramach dwóch międzynarodowych konferencji: International Joint Conference on Rough Sets 2015 oraz Federated Conference on Computer Science and Information Systems.

5.2 Metody bazujące na teorii zbiorów przybliżonych — RapidRoughSets

Operatory tworzące blok RapidRoughSets opracowane zostały w taki sposób, aby w maksymalnym stopniu wykorzystać i udostępnić szerszej grupie użytkowników najważniejsze metody analityczne oferowane przez Teorię Zbiorów Przybliżonych (Rough Set Theory — RST) [5.14]. W RapidRoughSets zawarto efektywne obliczeniowo implementacje metod będących kanonem RST. W szczególności w RapidRoughSets dostępne są operatory realizujące następujące metody i algorytmy:

- definiowanie przybliżeń zbioru (przybliżenie dolne, górne, obszar brzegowy, obszar negatywny): ta grupa operatorów przyjmuje na wejściu zbiór przykładów X i atrybutów B, a na wyjściu generuje zbiór przykładów na leżących do przybliżeń zbioru X definiowanych przez zbiór atrybutów B [5.14];
- definiowanie przybliżeń zbiorów: operatory podobne do poprzedniego, ale mogące operować na większej liczbie zbiorów (np. na wszystkich klasach decyzyjnych jednocześnie); operator umożliwia także wyznaczenie obszaru pozytywnego tablicy decyzyjnej [5.14];

– ilościowa ocena jakości przybliżeń: ta grupa operatorów służy do obliczania wartości znanych w RST wskaźników jakości przybliżeń (np. współczynnik dokładności przybliżenia α);
– selekcja cech za pomocą metod obliczania reduktów decyzyjnych: zaimplementowano wiele operatorów realizujących efektywne obliczeniowo metody takie jak algorytmy dokładne obejmujące m.in. wyznaczenie jednego reduktu zawierającego minimalna liczbę atrybutów (najkrótszy redukt) [5.6]; algorytmy aproksymacyjne poszukujące reduktów wg strategii heurystycznych (zachłannej, permutacyjnej iDAAR Dynamically Adjusted Approximate Reducts [5.9]);
– nienadzorowana dyskretyzacja atrybutów za pomocą metod wnioskowania boolowskiego: dostępne są dwa algorytmy, jeden bazujący na globalnej, drugi na lokalnej (w obrębie klasy decyzyjnej) rozróżnialności pomiędzy przykładami [5.6];
– selekcja cech: realizowana za pomocą metody MRMR (Minimal Redundancy Maximal Relewance); algorytm MRMR okazuje się efektywnym narzędziem selekcji cech w zbiorach danych złożonych z tysięcy atrybutów; rozwiązanie to jest stosunkowo nowe dlatego poniżej zaprezentowano główną ideę algorytmu.

Celem algorytmu, bazującego na podejściu *MRMR* (Minimal Redundancy Maximal Relevance), jest minimalizacja liczby podobnych atrybutów (tj. niosących podobną informację) jakie znajdą się w wynikowym zbiorze cech, przy zachowaniu maksymalnie dużej ilości informacji w danych. Pseudokod algorytmu przedstawiony jest poniżej. Algorytm ten został również zaimplementowany w języku R jako niezależna biblioteka *RmRMR*, dosępna pod adresem https://github.com/janusza/RmRMR.

Przedstawiony algorytm na wejściu potrzebuje zbioru dostępnych atrybutów warunkowych oraz atrybutu decyzyjnego, np. w postaci systemu decyzyjnego [5.14]. Jako parametr przyjmuje również funkcje opisującą zależność pomiędzy dwoma dowolnymi atrybutami oraz liczbę rzeczywistą określającą dopuszczalne prawdopodobieństwo wybrania nieistotnego atrybutu w pojedynczej iteracji algorytmu.

W pierwszym kroku wybierany jest atrybut, którego zależność z decyzją w danych jest największa. Następnie, w pętli typu *while* wybierane są kolejne atrybuty, tak by w każdej iteracji maksymalizować różnicę między zależnością (tj. wartością funkcji ϕ) wybieranej cechy i decyzji oraz maksymalną zależnością wybieranej cech i dowolnego z wcześniej wybranych atrybutów. Jako kryterium stopu w algorytmie wykorzystywany jest, tak zwany test próbek losowych (ang. random probe test) [5.3, 5.9]. Test ten polega na empirycznej estymacji prawdopodobieństwa, że losowo wygenerowany atrybut o rozkładzie wartości zgodnym z cechą $\bar{a}$ wybraną w danej iteracji pętli, może być co najmniej tak zależny z atrybutem decyzyjnym jak sama cecha $\bar{a}$.

W praktyce, opisany powyżej algorytm pozwala generować kompaktowe zbiory cech, które dobrze oddają najważniejsze aspekty danych na potrzeby wizualizacji, czy też tworzenia modeli predykcyjnych pozwalających na łatwą inter-

pretację wyników. Możliwe jest także wykorzystanie go do wygenerowania wielu różnych podzbiorów cech stanowiących dobrą bazę do budowy zespołów zróżnicowanych klasyfikatorów, np. poprzez uruchamianie algorytmu na wielu losowych podzbiorach atrybutów z danych [5.20, 5.23].

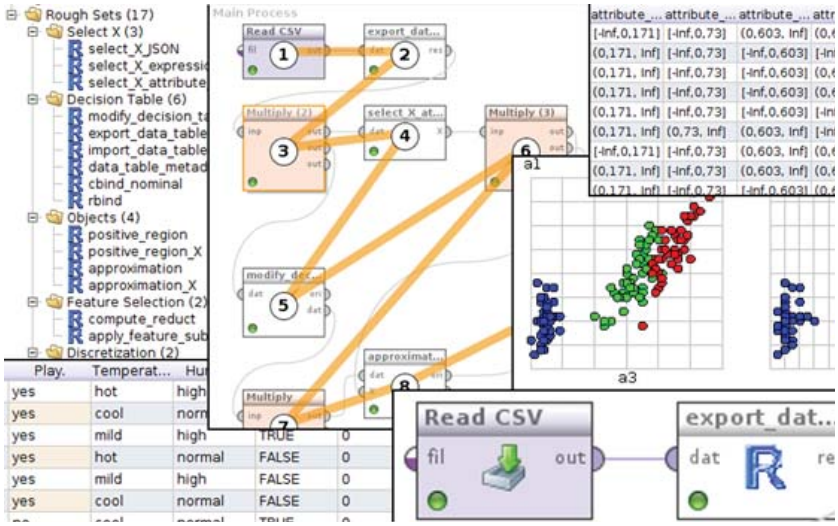
Input: zbiór atrybutów warunkowych A oraz atrybut decyzyjny d;
$\phi : A \times A \cup \{d\} \to R^+$ funkcja mierząca zależność atrybutów;
$N \in \mathbb{N}$; $\varepsilon \in [0, 1)$;
Output: podzbiór istotnych cech $A' \subseteq A$
begin
  $stopFlag \leftarrow FALSE$;
  $A' \leftarrow \arg\max_{a \in A} \ \phi\big(a, d\big)$;
  $A \leftarrow A \setminus A'$;
  while $stopFlag == FALSE$ **do**
    $\bar{a} \leftarrow \arg\max_{a \in A} \ \Big(\phi\big(a, d\big) - \max_{b \in A'} \ \phi\big(a, b\big)\Big)$;
    foreach $i \in 1, \ldots, N$ **do**
      $\bar{p}_i \leftarrow$ losowa permutacja $\bar{a}$;
    end
    if $\frac{|\{i : |\phi(\bar{p}_i, d)| > |\phi(\bar{a}, d)|\}| + 1}{N+2} > \varepsilon$ **then**
      $stopFlag \leftarrow TRUE$;
    else
      $A' \leftarrow A' \cup \bar{a}$
    end
  end
end

Algorytm 1: Pseudokod algorytmu wyboru istotnych cech bazującego na podejściu *mR-MR*, zaimplementowanego w systemie DISESOR

Każdy z algorytmów (operatorów) udostępniany przez RapidRoughSets może być łączony z innymi operatorami systemu DISESOR lub szerzej, środowiska RapidMiner (rys. 5.1).

5.3 Metody indukcji reguł — RuleInduction

W zależności od zastosowanej metody analitycznej uzyskujemy różne formy reprezentacji wiedzy, jaką udało się odkryć na podstawie danych. Drzewa i reguły stanowią reprezentację, która uważana jest za najbardziej zbliżoną do sposobu zapisu wiedzy przez człowieka. Najbardziej rozpowszechnionym rodzajem reguł są reguły decyzyjne zwane równie często regułami klasyfikacyjnymi [5.4, 5.5]. Są to wyrażenia warunkowe, reprezentujące lokalne zależności pomiędzy wartościami cech (atrybutów) charakteryzujących analizowany zbiór danych, przy czym cecha znajdująca się w konkluzji reguły decyzyjnej jest ustalona i nazywa się ją atrybutem decyzyjnym. Inny popularny rodzaj reguł to reguły regresyjne, które



Rys. 5.1. RapidRoughSets. Przykład procesu realizowanego w środowisku RapidMiner za pomocą operatorów RapidRoughSets

od reguł klasyfikacyjnych różnią się tym, że w konkluzji znajduje się albo wskazanie na konkretną wartość cechy decyzyjnej (która w tym przypadku jest cechą numeryczną) albo pewne wyrażenie matematyczne, które pozwala tę wartość wyliczyć. Trzeci typ reguł to tzw. wzorce przeżycia. W konkluzji wzorca przeżycia znajduje się estymator funkcji przeżycia (np. estymator Kaplana–Meiera [5.10]) wskazujący jak obliczyć prawdopodobieństwo przeżycia dla każdego z przykładów spełniających część warunkową reguły.

Załóżmy, że dostępny jest zbiór przykładów E, a każdy przykład opisany jest pewnym zbiorem atrybutów $A \cup \{d\}$. Każdy atrybut może być traktowany jako funkcja $a : E \to V_a$, gdzie V_a nazywany jest zakresem atrybutu a w zbiorze E. Elementy zbioru A nazywane są atrybutami warunkowymi, a atrybut d nazywany jest decyzyjnym. Atrybuty warunkowe są zazwyczaj typu dyskretnego, porządkowego lub rzeczywistego (liczbowego).

W przypadku reguł klasyfikacyjnych atrybut d jest atrybutem dyskretnym. W regułach regresyjnych atrybut decyzyjny jest typu rzeczywistego. W obu wymienionych typach reguł każdy przykład $x \in E$ może być zapisany jako wektor $x = (x_1, x_2, \ldots, x_{|A|}, y)$ gdzie $a_i(x) = x_i$ dla każdego $i \in \{1, 2, \ldots, |A|\}$ oraz $y = d(x)$.

W analizie przeżycia występują tzw. przykłady ucięte zwane również cenzurowanymi. Z tego powodu każdy przykład opisują dwa atrybuty decyzyjne T (zwany czasem obserwacji) oraz d (zwany statusem). Dla dowolnego przykładu x, czas $T(x) = t$ oznacza czas przeżycia jaki udało się do chwili obecnej zaobserwować dla przykładu x. Równocześnie przykład x opisany jest jedną z dwóch możliwych wartości atrybutu d. Jeśli $d(x) = 0$ oznacza to, że dla przy-

kładu x nie zaobserwowano w czasie t tzw. zdarzenia. Zdarzeniem nazywamy zajście pewnego zjawiska, które jest szczególnie interesujące z punktu widzenia celu analizy np. śmierć pacjenta (analiza przeżycia), awaria maszyny (analiza niezawodności), etc. Jeśli $d(x) = 1$ oznacza, to że dokładnie po upływie czasu t, dla przykładu x zaobserwowano zdarzenie. W analizie przeżycia każdy przykład reprezentowany jest zatem przez wektor $x = (x_1, x_2, \ldots, x_{|A|}, t, y)$.

Aby zadanie indukcji reguł przeżycia sprowadzić do indukcji reguł klasyfikacyjnych opracowano metodę przyporządkowywania przykładom, dla których $d(x) = 0$, wag i przekształcania zbioru przykładów z dwoma atrybutami decyzyjnymi T i d do zbioru zawierającego jeden atrybut D. Atrybut D może przyjmować dwie wartości ($\oplus$ — wystąpiło zdarzenie, $\otimes$ — nie wystąpiło zdarzenie), a do każdego przykładu przyporządkowana jest dodatkowo wartość liczbowa w zwana wagą przykładu.

Wartości $T(x) = t$ i $d(x) = 0$ oznaczają jedynie, że do chwili t nie wystąpiło zdarzenie, nie oznacza to, że zdarzenie nie może wystąpić po czasie dłuższym od t. W związku z tym, na bazie każdego przykładu x dla którego $d(x) = 0$ definiowane są dwa przykłady, których wagi obliczane są na podstawie ilorazu prawdopodobieństw obliczonych metodą estymaty Kaplana—Meiera. W liczniku znajduje się prawdopodobieństwo przeżycia obliczone dla czasu najdłuższego przeżycia jaki zaobserwowano w zbiorze treningowym. W mianowniku znajduje się prawdopodobieństwo obliczone dla przykłady x. Jeśli iloraz ten oznaczymy jako w, to w zbiorze treningowym pojawiają się dwa przykłady $(x, \oplus, w)$ oraz $(x, \otimes, 1-w)$. Przykłady dla których wystąpiło zdarzenie reprezentowane sa jako $(x, \oplus, 1)$.

Regułą decyzyjną r nazywamy wyrażenie (5.1):

$$\textbf{jeżeli } w_1 \ \wedge \ w_2 \ \wedge \ \ldots \ \wedge \ w_j \textbf{ to } decyzja \tag{5.1}$$

Przesłanka reguły składa się z koniunkcji warunków elementarnych. Aby przykład spełniał warunek elementarny, musi on spełniać zapisane w nim ograniczenia. Reguła (5.1) reprezentuje zależność mówiącą o tym, że dla przykładów spełniających jednocześnie wszystkie ograniczenia zapisane w warunkach elementarnych należy podjąć decyzję wyspecyfikowaną w jej konkluzji.

Warunek elementarny w definiowany jest jako wyrażenie o postaci $a\ op\ Z_a$. W wyrażeniu tym a jest atrybutem warunkowym, traktowanym tutaj jako zmienna mogąca przyjmować wartości należące do dziedziny atrybutu a. Operator relacyjny Op wskazuje na jeden z symboli relacyjnych należących do zbioru $\{=, \neq, \leqslant, \geqslant, >, <, \in\}$, a Z_a jest tzw. zakresem warunku i w zależności od użytego operatora relacyjnego jest wartością lub podzbiorem zbioru V_a (np. temperatura $> 36{,}6$; stan_zagrożenia = dopuszczalny, etc.).

O przykładach spełniających przesłankę reguły r mówi się, że pokrywają one r lub są przez r pokrywane. Zbiór przykładów pokrywanych przez r oznaczamy $[r]$.

Jeśli mamy do czynienia z regułą klasyfikacyjną jej konkluzja wskazuje na klasę decyzyjną jaką należy przyporządkować przykładom należącym do $[r]$. Konkluzje reguł regresyjnych rozważanych w niniejszym rozdziale składają się

z konkretnej wartości liczbowej v, którą należy przyporządkować przykładom należącym do $[r]$. Bardzo dobre wyniki (niski błąd prognozy) uzyskuje się jeśli wartość v jest medianą lub średnią z wartości atrybutów decyzyjnych przykładów należących do $[r]$ [5.18]. W konkluzji reguły przeżycia znajduje się funkcja będąca estymatą Kaplana–Meiera [5.10] obliczoną na podstawie przykładów z $[r]$.

Aby wyznaczyć reguły konieczne jest określenie wartości czterech liczb: p, n, P, N. Dla reguły klasyfikacyjnej r, $P = |Pos(r)|$, gdzie $Pos(r)$ jest zbiorem wszystkich przykładów treningowych których decyzje sa identyczne z decyzją reguły. Elementy zbioru $Pos(r)$ to tzw. przykłady pozytywne. Wartość N obliczana jest podobnie: $N = Neg(r)$, gdzie $Neg(r)$ jest zbiorem wszystkich przykładów ze zbioru treningowego o wartościach atrybutu decyzyjnego innych niż wskazano to w decyzji r. Wartości liczb p i n obliczane są na podstawie zależności $p = |Pos(r) \cap [r]|$, $n = |Neg(r) \cap [r]|$.

W przypadku reguł regresyjnych zbiory $Pos(r)$ i $Neg(r)$ zmieniają się podczas procesu indukcji konkretnej reguły. Załóżmy, że dana jest reguła regresyjna r o wartości konkluzji v, wówczas do zbioru $Pos(r)$ należą te przykłady treningowe, których wartość atrybutu decyzyjnego mieści się w przedziale $[v-\sigma, v+\sigma]$, gdzie σ jest wartością odchylenia standardowego wartości atrybutu decyzyjnego obliczoną dla przykładów z $[r]$. Elementy zbioru $Neg(r)$ to wszystkie elementy zbioru treningowego, które nie należą do $Pos(r)$.

W przypadku reguł przeżycia mamy do czynienia z dwoma wartościami atrybutu decyzyjnego $D(\oplus, \ominus)$, wartości P, N, p i n obliczane są identycznie jak w przypadku reguł klasyfikacyjnych. Dla dowolnej reguły r i danego zbioru treningowego możliwe jest wyznaczenie macierzy kontyngencji prezentowanej w Tablicy 5.1.

Tab. 5.1. Macierz kontyngencji wyznaczona dla reguły r, pokrywającej p przykładów pozytywnych i n przykładów negatywnych

p	n	$p+n$
$P-p$	$N-n$	$P+N-p-n$
P	N	$P+N$

Celem pokryciowego algorytmu indukcji reguł jest wyznaczenie reguł pokrywających jak najwięcej przykładów pozytywnych i jak najmniej negatywnych. Znalezienie minimalnego zbioru reguł pokrywającego wszystkie przykłady treningowe regułami o wysokich wartościach p i niskich n jest problemem NP–trudnym. W praktyce stosuje się heurystyczny algorytm sekwencyjnego pokrywania (Alg. 2).

Proces indukcji przebiega oddzielnie dla każdej klasy decyzyjnej. W każdej iteracji algorytmu tworzona jest jedna reguła. Następnie usuwane są wszystkie przykłady pozytywne pokrywane przez utworzoną regułę, a proces indukcji reguł trwa tak długo, dopóki nie pokryte zostaną wszystkie przykłady pozytyw-

RuleInduction(dataset, targetClass, minCov, q_gr, q_pr)
Input:
dataset: cały zbiór obserwacji
targetClass: zbiór obserwacji należących do targetClass
minCov: minimalne wymagane pokrycie klasy targetClass pojedynczą regułą
q_gr, q_pr: miary jakości reguł
Output:
zbiór reguł opisujących klasę targetClass
zbiór wygenerowanych reguł (początkowo pusty)
ruleSet := ∅
if *regression_rule_induction* **then**
| targetClass := dataset
end
while *targetClass* ≠ ∅ **do**
| rule := LearnRule(dataset, targetClass, minCov, q_gr, q_pr)
| rules := rules ∪ {rule}
| targetClass := targetClass \ Covered(rule, targetClass)
end
return *rules*

Algorytm 2: Zarys algorytmu indukcji reguł klasyfikacyjnych, regresyjnych i przeżycia

ne. Przesłanka reguły budowana jest zgodnie ze strategią wspinaczki, warunki elementarne dodawane są do przesłanki dopóki reguła nie pokrywa jedynie przykładów pozytywnych lub dodawanie kolejnych warunków elementarnych nie powoduje już eliminacji ze zbioru $[r]$ przykładów negatywnych.

Podczas wzrostu reguły testowane są wszystkie możliwe warunki elementarne jakie można dodać do przesłanki. Dla atrybutu symbolicznego a testowane są wszystkie warunki postaci $a = v$, gdzie $v \in V_a$. Dla atrybutu typu numerycznego testowane są wszystkie warunki postaci $a < v$, $a \geqslant v$, gdzie v jest wartością średniej arytmetycznej pomiędzy dwoma kolejnymi wartościami atrybutu a. Na stałe dodawany jest ten warunek, który zapewnia największą wartość miary jakości q_{gr} używanej podczas fazy wzrostu. Faza przycinania polega na usuwaniu warunków, które po fazie wzrostu mogą okazać się zbędne. Szablon procedury budowy przedstawia algorytm 2.

Do nadzorowania procesu indukcji używane są znane z literatury i najbardziej efektywne — z punktu widzenia uzyskanych dokładności prognoz — miary jakości reguł [5.2, 5.8, 5.16, 5.19]. Zadaniem miary jakości jest takie sterowanie procesem tworzenia warunków elementarnych, aby maksymalizować wartość p i jednocześnie minimalizować wartość n. Użytkownik na dowolnym etapie indukcji, może użyć dowolnej z miar zaprezentowanych w Tabeli 5.2.

W standardowym algorytmie indukcji reguł użytkownik nie ma wpływu na postać tworzonych warunków elementarnych. Pokryciowy algorytm indukcji reguł można jednak zmodyfikować tak, aby pod uwagę brał on preferencje użytkownika [5.17]. Preferencje te najczęściej wyrażane są przez:

```
LearnRule(dataset, targetClass, minCov, q_gr, q_pr)
Input:
zob. funkcja RuleInduction
Output:
nauczona reguła
faza wzrostu reguły
rule := ∅
repeat
    bestQuality := −∞
    bestCondition := ∅
    for c ∈ PossibleConditions(rule, dataset) do
        if |Covered(rule ∪ {c}, classUncovered)| <Minimum(|classUncovered|,
          minCov) then
            continue
        end
        quality := Evaluate(rule ∪ {c}, dataset, q_gr)
        if quality > bestQuality then
            bestQuality := quality
            bestCondition := c
        end
    end
    rule := rule ∪ {bestCondition}
until reguła jest dokładna or bestCondition = ∅;
faza przycinania reguły
ruleQuality := Evaluate(rule, dataset, q_pr)
repeat
    worstCondition := ∅
    for c ∈ Conditions(rule) do
        quality := Evaluate(rule  {c}, dataset, q_pr)
        if quality ⩾ ruleQuality then
            ruleQuality := quality
            worstCondition := c
        end
    end
    rule := rule \ {worstCondition}
until worstCondition = ∅;
return rules
```

Algorytm 3: Zarys algorytmu indukcji pojedynczej reguły

- wymuszanie (zabranianie) pojawiania się konkretnych atrybutów w przesłankach reguł,
- wymuszanie (zabranianie) pojawiania się konkretnych warunków elementarnych w przesłankach reguł,
- wymuszanie (zabranianie) pojawiania się konkretnych reguł w opisach klas decyzyjnych.

W zależności od tego, jakiego rodzaju preferencje zostaną zdefiniowane, mamy do czynienia z różnymi trybami działania algorytmu.

O1. Użytkownik wprowadza zbiór gotowych reguł:

Tab. 5.2. Zestawienie wybranych miar oceny jakości reguł

$$\mathbf{g} = \frac{p}{p+n+2}$$

$$\mathbf{wLap} = \frac{(p+1)(P+N)}{(p+n+2)P}$$

$$\mathbf{LS} = \frac{pN}{nP}$$

$$\mathbf{Rss} = \frac{p}{P} - \frac{n}{N}$$

$$\mathbf{C1} = Coleman \cdot \frac{2 + Cohen}{3}$$

$$\mathbf{C2} = Coleman \cdot 0.5\left(1 + \frac{p}{P}\right)$$

$$\mathbf{Corr} = \frac{pN - Pn}{\sqrt{PN(p+n)(P-p+N-n)}}$$

$$\mathbf{s} = \frac{p}{p+n} - \frac{P-p}{P-p+N-n}$$

$$Cohen = \frac{(P+N)(\frac{p}{p+n}) - P}{\frac{P+N}{2}\frac{p+n+P}{p+n} - P}$$

$$Coleman = \frac{(P+N)\frac{p}{p+n} - P}{N}$$

a) algorytm nie ingeruje w budowę reguł ani nie dodaje do nich nowych reguł; efektywność zdefiniowanego zbioru reguł weryfikowana jest na całym dostępnym zbiorze przykładów lub w trybie walidacji krzyżowej;
b) algorytm dokonuje redefinicji reguł wprowadzonych przez użytkownika; redefinicję rozumiemy jako dodanie do przesłanek nowych warunków elementarnych (faza wzrostu) i/lub usunięcie warunków istniejących (faza przycinania); redefinicja przeprowadzana jest po to, aby zwiększyć jakość reguł zdefiniowanych przez użytkownika;
c) algorytm dokonuje redefinicji reguł wprowadzonych przez użytkownika w sposób identyczny z opisanym w poprzednim punkcie; po zakończeniu redefinicji generowane są kolejne reguły, tak aby pokryć zbiór przykładów treningowych;

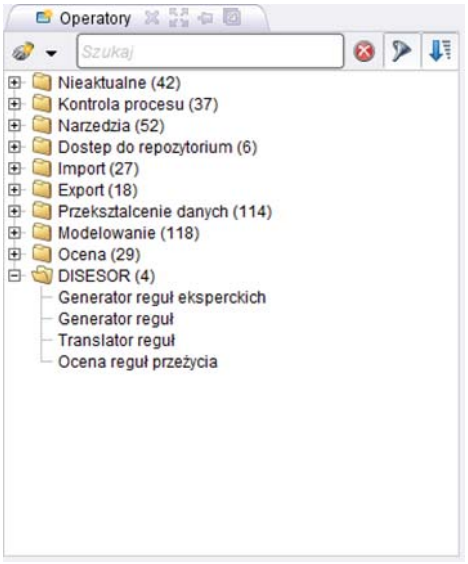
O2. Użytkownik definiuje dla każdej klasy decyzyjnej warunki elementarne, jakie muszą pojawić się w co najmniej jednej z reguł opisujących daną klasę decyzyjną. Dopuszczalne jest definiowanie pojedynczego warunku elementarnego lub ich koniunkcji.
O3. Użytkownik wskazuje atrybuty, które muszą się pojawić w co najmniej jednej regule opisującej daną klasę decyzyjną.

Opcja O1.a) pozwala na weryfikację hipotez definiowanych przez użytkownika. Hipotezy te wyrażane są w postaci reguł. Opcja O1.b) pozwala na doprecyzowanie reguł przedstawionych przez użytkownika tak, aby zmaksymalizować ich jakość.

W początkowej fazie działania algorytm stara się wygenerować reguły spełniające wymagania eksperta, a następnie kontynuuje indukcję dla pozostałych, niepokrytych jeszcze przykładów pozytywnych (oczywiście z wyjątkiem opcji O1.a) oraz O1.b)). Opracowana implementacja umożliwia łączenie opcji O1., O2., O3. Przykładowo możliwe jest połączenie opcji O1.c) z opcją O2. i/lub opcją O3. Algorytm rozpoczyna indukcję od doprecyzowania reguł zadanych przez użytkownika i jeśli po redefinicji nie spełniają one wymagań określonych opcjami O2., O3., to w kolejnym etapie generowane są reguły spełniające te wymagania. W przypadku jednoczesnego stosowania opcji O2. oraz O3. algorytm najpierw stara się wygenerować reguły spełniające warunki zdefiniowane w O2.

Efektywność przedstawionych metod indukcji reguł potwierdzono m.in. w następujących publikacjach: [5.19, 5.21, 5.22]. W pracach tych zamieszczono porównania z innymi algorytmami indukcji reguł. Porównania wykonano na kilkudziesięciu ogólnodostępnych zbiorach danych benchmarkowych, a do porównań użyto odpowiednich testów statystycznych.

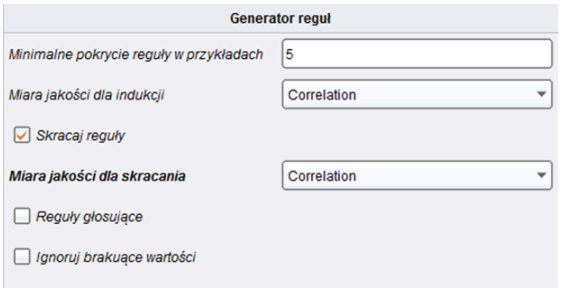
Na Rysunku 5.2 zaprezentowano widok operatorów za pomocą, który można dokonać indukcji reguł zgodnie z zasadami przedstawionymi powyżej.



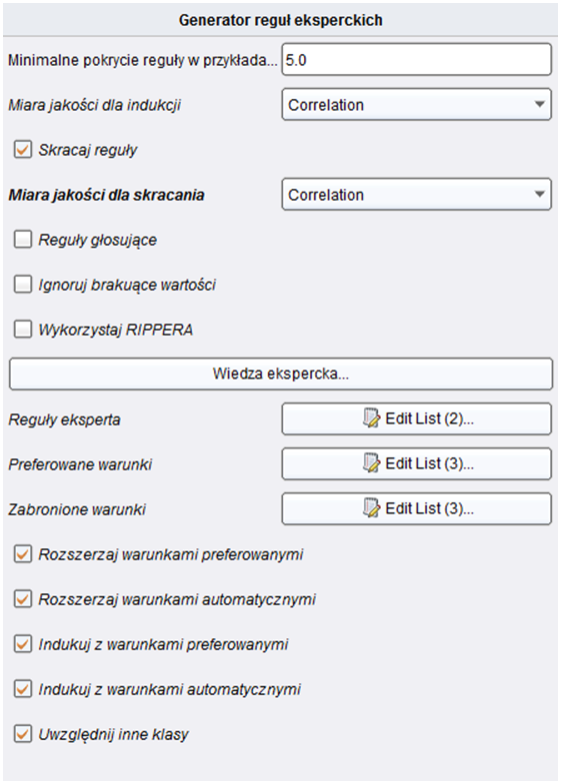
Rys. 5.2. DISESOR — Operatory indukcji reguł

Po wybraniu typu indukowanej reguły pokazuje się okno konfiguracyjne umożliwiające: wybór miar jakości sterujących procesem indukcji, zdefiniowanie

sposobu rozstrzyganiem konfliktów klasyfikacji, obsługę przykładów zawierających brakujące wartości atrybutów (rys. 5.3), czy wreszcie ustalenie minimalnego pokrycia generowanych reguł.

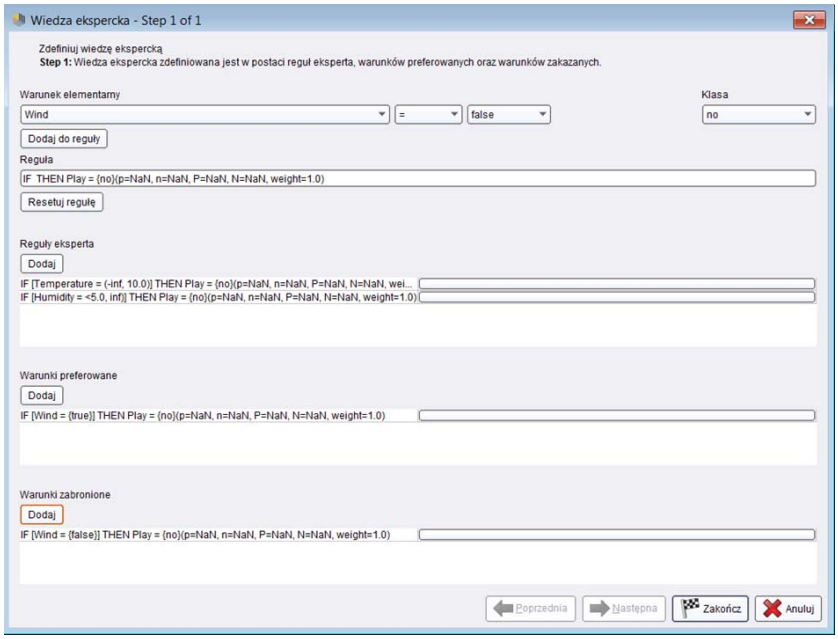


Rys. 5.3. Formularz konfiguracyjny procesu indukcji reguł



Rys. 5.4. Formularz konfiguracyjny indukcji reguł nadzorowanej przez eksperta

Formularz konfiguracyjny indukcji reguł nadzorowanej przez zaawansowanego użytkownika (eksperta) posiada więcej opcji konfiguracyjnych (rys. 5.4). Na Rysunku 5.5 zaprezentowano formularz umożliwiający ekspertowi na zdefiniowanie warunków nadzorowania indukcji.



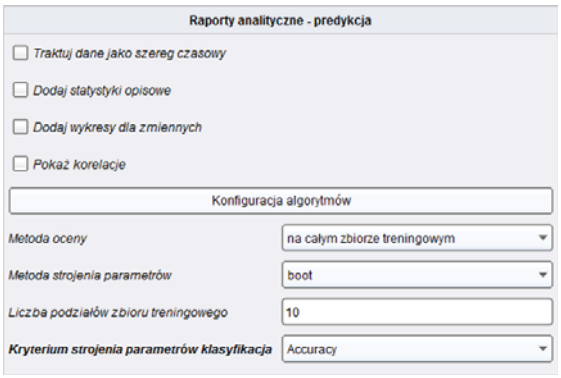
Rys. 5.5. Formularz definiowania preferencji eksperta dotyczących budowy wyznaczanych reguł

Prezentowane powyżej mechanizmy indukcji reguł mogą być także wykorzystane z poziomu języka programowania Java — bez pośrednictwa interfejsu użytkownika.

5.4 Asystent analizy — Analytics.Raports

Użycie modułu analitycznego dostępne jest również dla użytkowników niezaawansowanych. Dla tej grupy użytkowników opracowano tzw. asystenta analizy, za pośrednictwem którego proces analizy danych jest w dużej mierze zautomatyzowany. Asystent analizy umożliwia realizację zadań tworzenia modeli klasyfikacyjnych i regresyjnych oraz analizę asocjacji (rys. 5.6).

Zasadą działania asystenta jest testowanie różnych modeli analitycznych wraz z różnymi ustawianiami parametrów wymaganych przez metody będące podstawą do utworzenia tych modeli. Działanie użytkownika — opcja minimum — ogranicza się do wskazania zbioru danych i określanie jakiego rodzaju informacje



Rys. 5.6. Formularz konfiguracyjny asystenta analizy dla zadań prognostycznych

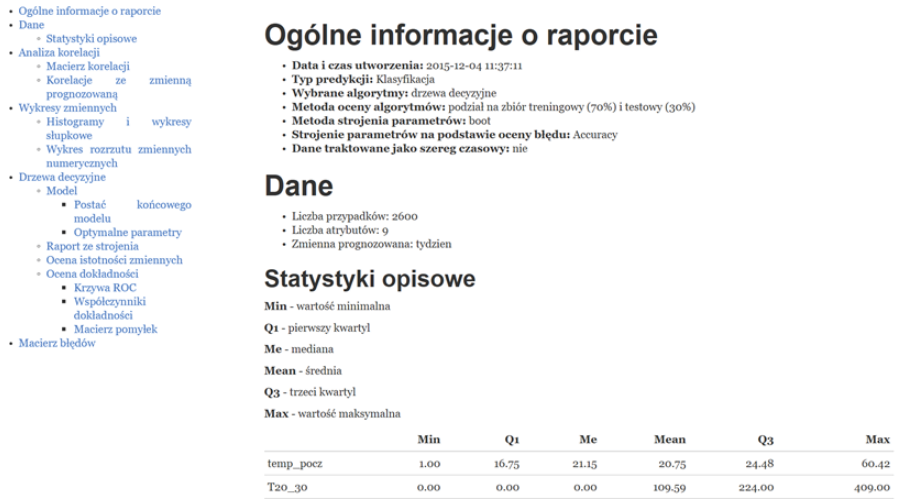
podsumowujące mają znaleźć się w końcowym raporcie. Nieco bardziej świadomy użytkownik posiada możliwość wyboru konkretnych metod analitycznych czy też ustawieniu pewnych parametrów tych metod.

Asystent analizy posiada zaimplementowaną grupę metod bazowych, które są podstawą do budowania modeli danych. Metody bazowe to:

- indukcja drzew decyzyjnych,
- indukcja reguł,
- SVM,
- kNN,
- naiveBayes,
- lasy losowe,
- sieci neuronowe,
- indukcja reguł asocjacyjnych.

W module zdefiniowane są również dwie metody poszukiwania optymalnych wartości parametrów metod bazowych dla wczytanego zbioru danych: metoda prostsza polegająca na sekwencyjnym poszukiwaniu optymalnych wartości kolejnych parametrów oraz metoda złożona (siatkowa) polegająca na poszukaniu optymalnej kombinacji wartości wszystkich parametrów. Dobór wartości parametów może odbywać się na podstawie zbioru treningowego, jego części lub w trybie wewnętrznej (niezależnej od zbioru testowego) walidacji krzyżowej.

Wynikiem działania asystenta jest raport podsumowujący. W raporcie znajdują się podstawowe informacje o danych. Przykładowo są to statystyki opisowe każdej zmiennej (rys. 5.7), macierz korelacji pomiędzy zmiennymi oraz wykresy (linowe, histogramy). Zasadnicza część raportu składa się z sekcji, każda sekcja opisuje wyniki uzyskane przez daną metodę analityczną (rys. 5.8), prezentowane są optymalne wartości parametrów metody oraz wyniki testowania modelu. Strojenie parametrów modelu odbywa się wg kryterium optymalności wybranego przez użytkownika (np. AUC — Area Under ROC). W raporcie prezentowane są jednak wartości wielu kryteriów (m.in. Accuracy, Balanced Accuracy, MAE, F–measure, G–measure, RMSE, rRMSE, etc.)



Rys. 5.7. Początek przykładowego raportu analitycznego – po lewej stronie widoczne jest menu pozwalające przejść do poszczególnych sekcji raportu

Użytkownik po zakończeniu analizy może wybrać najbardziej interesujący go model i przekazać do działania w module prognostycznym lub próbować dalszej, dokładniejszej (manualnej) analizy bazującej na wybranym modelu. Działania takie jest możliwe, gdyż asystent analizy bazuje na procesach zdefiniowanych w środowisku RapidMiner (rys. 5.9). Użytkownik niezaawansowany nie musi znać środowiska Rapid—Miner aby mógł korzystać z podstawowej funkcjonalności asystenta analizy.

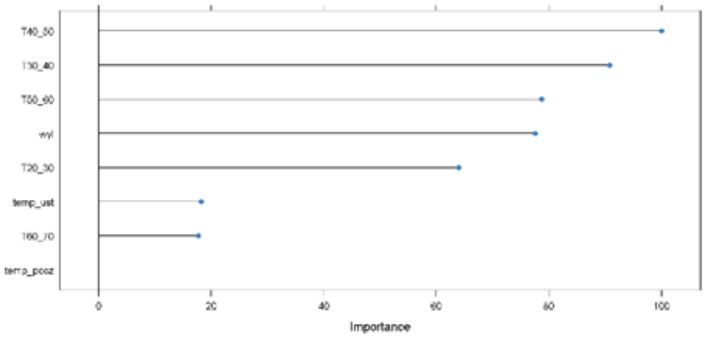
Procesy analityczne zostały opracowane na podstawie ontologii analizy danych opisane m.in. w pracach [5.11, 5.15].

W obecnej wersji systemu nie jest szacowany czas potrzebny na wykonanie obliczeń, użytkownik może jednak przerwać proces analizy, a wtedy w raporcie umieszczane są jedynie informacje dotyczące metod dla których zakończono obliczenia.

Raport ze strojenia

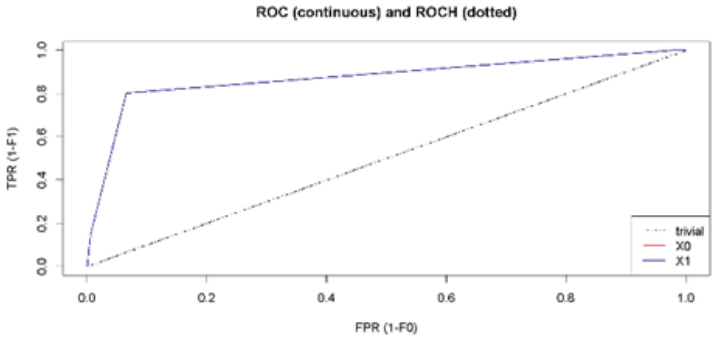
cp	Accuracy	Kappa	AccuracySD	KappaSD
0.0176991	0.8525188	0.7049071	0.0169729	0.0340698
0.0202276	0.8485749	0.6964022	0.0177047	0.0355681
0.1441214	0.8039530	0.6137747	0.0361386	0.0667589
0.5006321	0.6894447	0.3355389	0.1093923	0.2890983

Ocena istotności zmiennych



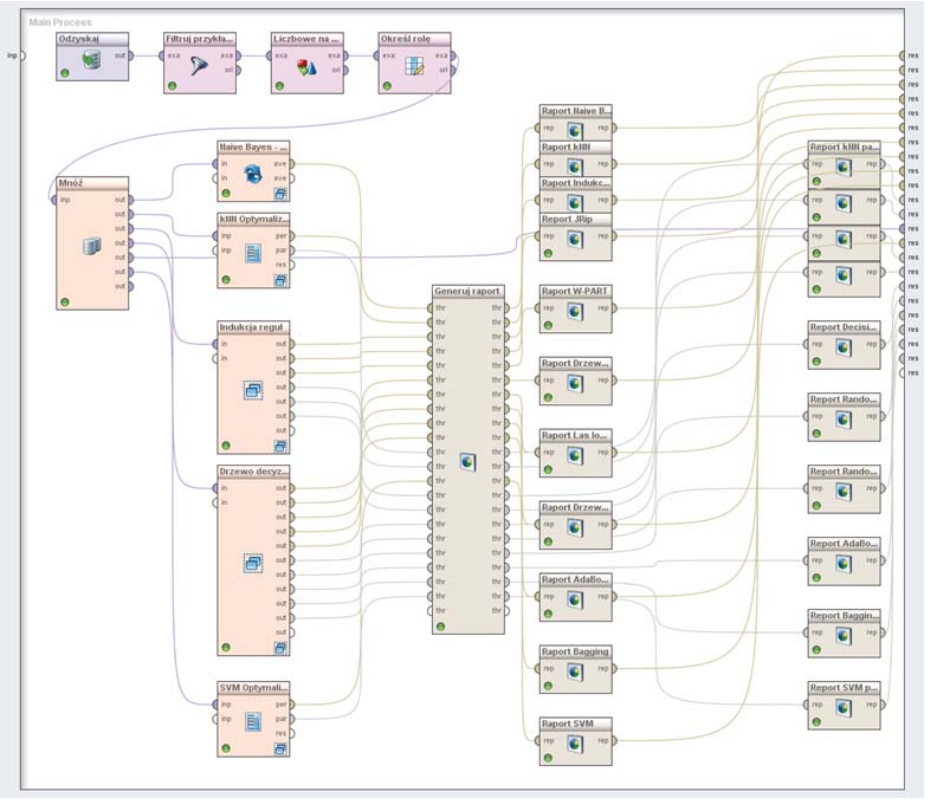
Ocena dokładności

Krzywa ROC



Klasa	AUC (pole pod krzywą ROC)
X0	0.873646500020126
X1	0.873646500020126

Rys. 5.8. Raport analityczny — widok fragmentu sekcji dotyczącej strojenia parametrów wybranej metody analitycznej

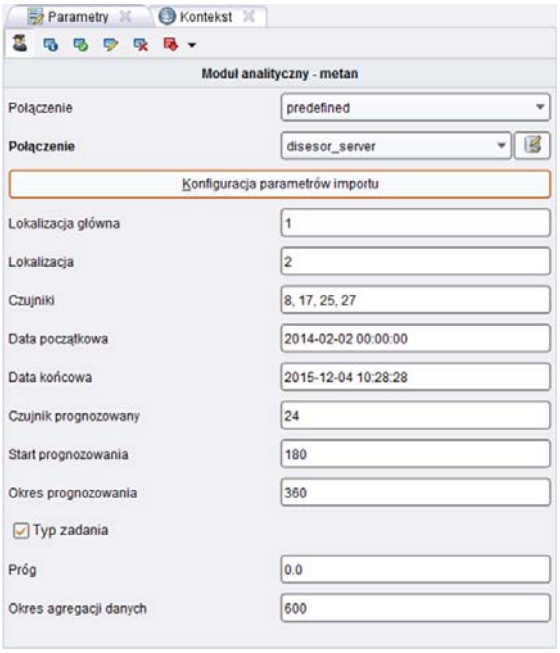


Rys. 5.9. Przykładowy widok procesu analizy danych będący podstawa działania asystenta analizy

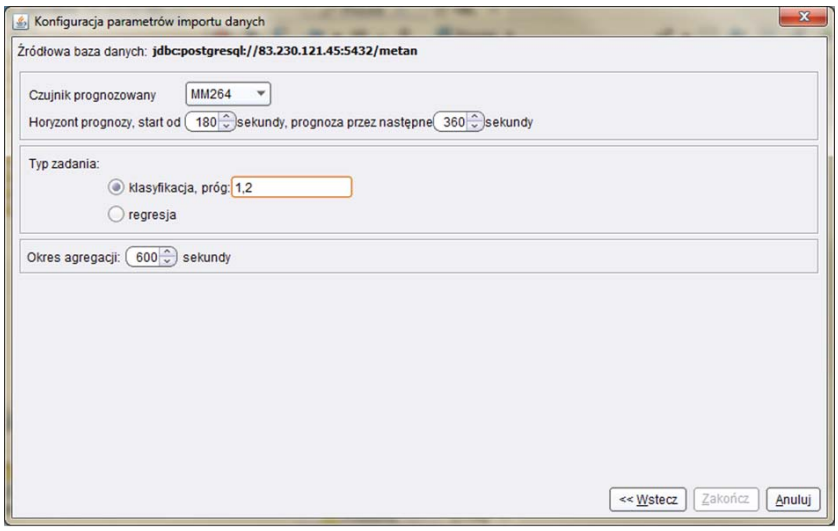
5.5 Operatory tematyczne — prognoza stężenia metanu i zagrożenia sejsmicznego

W module analitycznym opracowano również dwa komponenty związane stricte z pierwotnym obszarem zastosowania projektu — górnictwem węgla kamiennego. Komponenty te umożliwiają w sposób automatyczny utworzenie modeli prognozujących zagrożenia naturalne. Rozważano dwa typy zagrożeń: metanowe i sejsmiczne. W przypadku zagrożenia metanowego prognoza polega na wyznaczeniu maksymalnej wartości stężenia metanu (w zadanym horyzoncie prognozy) jaka zarejestrowana zostanie w pewnym obszarze wyrobiska. Prognoza może zatem dotyczyć pojedynczego metanomierza jak również grupy tych czujników. Prognoza zagrożenia sejsmicznego polega na wyznaczeniu wartości ilości energii jaka wydzieli się w wyrobisku w ciągu najbliższej godziny lub zmiany. Prognozowana jest wartość energii EPZ [5.12]. W literaturze przedmiotu [5.1, 5.13] wysokie wartości tej energii powiązane są bezpośrednio ze wzrostem zagrożenia.

Podczas realizacji prac badawczych związanych z poszukiwaniem najlepszej metody prognozowania zorganizowano dwa międzynarodowe konkursy mające na celu poszukiwanie jak najlepszych metodyk analizy danych związanych z opisanymi zagrożeniami. W rozdziale 11 opisano rezultaty tych konkursów. Metodyki zwycięskie zaimplementowano i udostępniono w module analitycznym. W chwili obecnej analiza danych związanych z przedstawionymi zagrożeniami może sprowadzić się do wskazania zbioru danych oraz konfiguracji wymagań (progi alarmowe, horyzont prognozy etc. — rys. 5.10, 5.11), a następnie, w przypadku uzyskania satysfakcjonujących wyników, wykorzystania uzyskanych rezultatów w module prognostycznym. W module prognostycznym istnieje możliwość śledzenia jakości tych modeli i w razie potrzeby ich automatycznego przeuczenia.



Rys. 5.10. Formularz konfiguracyjny zadania prognozowania zagrożeń metanowych — dane konfiguracyjne wprowadzane są za pomocą specjalnie przygotowanych formularzy, które otwierają się po przejściu kursora do danego pola



Rys. 5.11. Formularz konfiguracyjny zadania prognozowania zagrożeń metanowych — określanie parametrów dotyczących horyzontu prognozy i progów alarmowych

Literatura

[5.1] A. Barański, J. Drzewiecki, J. Kabiesz, W. Konopko, J. Kornowski, A. Krzyżowski, G. G. Mutke. *Rules of application of the comprehensive and detailed rockburst hazard assessment methods in hard-coal mines (in Polish)*, vol. 22 serii *Instructions of Central Mining Institute*. Central Mining Institue Press, 2010.

[5.2] I. Bruha, J. Tkadlec. Rule quality for multiple-rules classifier: Empirical expertise and theoretical methodology. *Intelligent Data Analysis*, 7(2):99–124, 2003.

[5.3] D. R. Cox, D. V. Hinkley. *Theoretical Statistics*. Chapman and Hall, London, UK, 1974.

[5.4] T. G. Dietterich, R. S. Michalski. Inductive learning of structural descriptions: Evaluation criteria and comparative review of selected methods. *Artificial Intelligence*, 16(3):257–294, 1981.

[5.5] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.

[5.6] N. S. Hoa. Some efficient algorithms for rough set methods. *In Proceedings IPMU'96 Granada (Spain)*, s. 1541–1457, 1996.

[5.7] M. Hofmann, R. Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman and Hall/CRC, 2013.

[5.8] F. Janssen, J. Fürnkranz. On the quest for optimal rule learning heuristics. *Machine Learning*, 78:343–379, 2010.

[5.9] A. Janusz, D. Ślezak. Computation of approximate reducts with dynamically adjusted approximation threshold. F. Esposito, O. Pivert, M. Hacid, Z. W. Ras, S. Ferilli, redaktorzy, *Proceedings of ISMIS 2015*, vol. 9384, s. 19–28. Springer, 2015.

[5.10] E. L. Kaplan, P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958.

[5.11] C. M. Keet, A. Ławrynowicz, C. d'Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens, M. Hilario. The data mining optimization ontology. *Web Semantics: Science, Services and Agents on the World Wide Web*, 32:43–53, 2015.

[5.12] J. Kornowski. Linear prediction of aggregated seismic and seismoacoustic energy emitted from a mining longwall. *Acta Montana*, 129:5–14, 2003.

[5.13] J. Kornowski, J. Kurzeja. Prediction of rockburst probability given seismic energy and factors defined by the expert method of hazard evaluation (MRG). *Acta Geophysica*, 60(2):472–486, 2012.

[5.14] Z. Pawlak, A. Skowron. Rudiments of rough sets. *Information Sciences*, 177(1):3–27, 2007.

[5.15] J. Potoniec, A. Ławrynowicz. Rmonto: ontological extension to rapidminer. *10th International Semantic Web Conference*. Citeseer, 2011.

[5.16] S. Sahar. *Data Mining and Knowledge Discovery Handbook*, rozdział: Interestingness measures - On determining what is interesting, s. 603–612. Springer-Verlag, 2010.