Modele predykcyjne – przykładowe techniki umożliwiające predykcję wartości pewnej zmiennej losowej[1]

I. Drzewa decyzyjne

Drzewa decyzyjne są jedną z najpopularniejszych metod prezentacji wiedzy zdobytej w procesie eksploracji danych. Algorytmy tworzenia drzew należą do najczęściej używanych metod tworzenia modelu danych. Jeden z najpopularniejszych to algorytm C4.5 (podobny do opisanego na wykładzie), na którym opiera się także działanie odpowiedniego komponentu narzędzia KNIME oraz algorytm J48 z narzędzia WEKA.

W celu wykonania zadań z następnych punktów należy zainstalować rozszerzenie WEKA:

1. Po uruchomieniu systemu, sprawdź, czy zainstalowano rozszerzenie WEKA (wpisać słowo *Weka* do okienka tekstowego w górnej części okna *Node Repository* i zobaczyć, czy zostały zwrócone jakieś rezultaty).

2. Jeśli nie zainstalowano rozszerzenia WEKA, z menu *File* wybierz opcję *Install KNIME Extensions*. W oknie, które się pojawi, rozwiń *KNIME & Extensions*, a następnie wybierz *KNIME WEKA Data Mining Integration (3.7).*

3. Zaczekaj na zakończenie instalacji, po czym zrestartuj narzędzie.

1. Budowa drzewa decyzyjnego w KNIME

1. Po uruchomieniu systemu i wybraniu odpowiedniej perspektywy (KNIME) przenieś na przestrzeń roboczą węzeł *ARFF Reader*. Skonfiguruj go do odczytu pliku *Hepatitis.arff*.

2. Obejrzyj zawartość pliku *Hepatitis.arff*.

Analizowany zbiór danych obejmuje 155 udokumentowanych przypadków żółtaczki. Wyróżnione atrybuty obejmują:
ID: liczba całkowita
    Identyfikator przykładu.
AGE: liczba całkowita
    Wiek pacjenta.
SEX: atrybut nominalny (M/K)
    Płeć pacjenta.
STEROID: atrybut nominalny (T/N)
    Czy zastosowano leki hormonalne?
ANTIVIRALS: atrybut nominalny (T/N)
    Czy zastosowano leki antywirusowe?
FATIGUE: atrybut nominalny (T/N)

[1] Przykłady zostały opracowane na podstawie: Kursu IBM: Introduction to IBM SPSS Modele rand Data Mining (Student Guide) oraz książki Daniela T. Larose „Odkrywanie wiedzy z danych" Wprowadzenie do eksploracji danych. Wydawnictwo Naukowe PWN, Warszawa 2006.

Czy pacjent skarżył się na zmęczenie?
MALAISE: atrybut nominalny (T/N)
Czy pacjent skarżył się na złe samopoczucie?
ANOREXIA: atrybut nominalny (T/N)
Czy u pacjenta występowała anoreksja?
LIVER_BIG: atrybut nominalny (T/N)
Czy u pacjenta wykryto powiększoną wątrobę?
LIVER_FIRM: atrybut nominalny (T/N)
Czy pacjent miał twardą wątrobę?
SPLEEN_PALPABLE: atrybut nominalny (T/N)
Czy pacjent miał wyczuwalną śledzionę?
SPIDERS: atrybut nominalny (T/N)
Czy u pacjenta wystąpiły gwiaździste naczyniaki (małe wybroczyny na skórze)?
ASCITES: atrybut nominalny (T/N)
Czy u pacjenta wystąpiło wodobrzusze?
VARICES: atrybut nominalny (T/N)
Czy u pacjenta wystąpiły żylaki?
BILIRUBIN: liczba rzeczywista
Wynik badania poziomu bilirubiny (żółty barwnik).
ALK_PHOSPHATE: liczba całkowita
Wynik badania poziomu fosforanów zasadowych.
SGOT: liczba całkowita
Wynik badania poziomu enzymu SGOT (uwalniany przy uszkodzeniu wątroby).
ALBUMIN: liczba rzeczywista
Wynik badania poziomu albumin.
PROTIME: liczba całkowita
Czas protrombinowy.
HISTOLOGY: atrybut nominalny (T/N)
Czy zastosowano badania histologiczne?
Class: atrybut nominalny (LIVE/DIE)
Końcowy efekt terapii.

3. Wstaw na przestrzeń roboczą węzeł *Analytics→Mining→Decision Tree→Decision Tree Learner*. Na jego wejście podaj dane odczytane z pliku *Hepatitis.arff*.

4. Przeanalizuj uzyskane drzewo decyzyjne, które próbuje przewidzieć końcowy efekt terapii.

5. Wstaw na przestrzeń roboczą drugi węzeł *Decision Tree Learner*. Na jego wejście podaj dane odczytane z pliku *Hepatitis.arff*.

6. Zapoznaj się z opcjami konfiguracyjnymi węzła *Decision Tree Learner*. W drugim węźle włącz przycinanie (*Pruning method* na *MDL*). Obejrzyj utworzone drzewo.

7. Użyjemy teraz utworzonych modeli do klasyfikacji. W tym celu należy wstawić na przestrzeń roboczą dwa węzły *Decision Tree Predictor* (z *Analytics→Mining→Decision*

Tree). Ich wejścia należy połączyć z odpowiednimi wyjściami węzłów *Decision Tree Learner* oraz z węzłem *ARFF Reader*.

8. Uruchom węzły *Decision Tree Predictor* za pomocą opcji *Execute*.

9. Obejrzyj dane wygenerowane przez każdy z węzłów (prawy przycisk, *Classified Data*).

10. Do wyjścia każdego z węzłów *Decision Tree Predictor* dołącz węzły *Scatter Plot (Local)* i *Interactive Table (Local)*. Wykresy punktowe skonfiguruj tak, aby pokazywały wartość *Class* oraz wartość przewidzianą (*Prediction* (*Class*)).

11. Na jednym z wykresów punktowych oznacz błędne decyzje dla jednego z klasyfikatorów (poprzez zaznaczenie obszaru i wybranie z menu kontekstowego *Hilite selected*).

12. Spróbuj oszacować liczbę błędnie sklasyfikowanych przykładów przez oba klasyfikatory. Zaznacz w tym celu odpowiednie przykłady na wykresie punktowym i posłuż się węzłem *Interactive table*. Dla ułatwienia przy podglądzie danych (w widoku danych węzła *Interactive table*) wybierz z menu opcję *Hilite* → *Filter* → *Show hilited only*.

2. Wykorzystanie mechanizmów WEKI

1. Na przestrzeń roboczą wstaw węzeł *J48* (*Analytics* → *Integrations* → *Weka* → *Weka (3.7)* → *Classification Algorithms* → *trees*). Na jego wejście podaj dane odczytane z pliku *Hepatitis.arff*.

2. Obejrzyj wygenerowane drzewo decyzyjne.

3. W konfiguracji węzła zmień *Confidence factor* na 0.15. Obejrzyj wygenerowane drzewo.

4. Zmień sposób przycinania drzewa, zmieniając w konfiguracji wartość opcji *reducedErrorPruning* na *tru*e. Obejrzyj wynik.

5. Przywróć poprzednią wartość atrybutu *reducedErrorPruning*. Na przestrzeń roboczą wstaw węzeł *Weka Predictor* (*Analytics* → *Integrations* → *Weka* → *Weka (3.7)* → *Predictors*). Jego wejścia połącz odpowiednio z wyjściami węzłów *J48* i *ARFF Reader*. Do wyjścia węzła dołącz węzeł *Interactive Table*.

6. Obejrzyj sklasyfikowane dane.

3. Budowa reguł decyzyjnych za pomocą mechanizmów WEKI

1. Po uruchomieniu systemu i wybraniu odpowiedniej perspektywy (KNIME) przenieś na przestrzeń roboczą węzeł *ARFF Reader*. Skonfiguruj go do odczytu pliku *Hepatitis.arff*

2. Wstaw na przestrzeń roboczą węzeł *Analytics* → *Integrations* → *Weka* → *Weka (3.7)* → *Classification Algorithms* → *rules* → *Prism*. Ten komponent implementuje uproszczoną

wersję algorytmu budowy reguł prezentowanego na wykładzie z wymaganą pełną poprawności reguł.

3. Komponent *Prism* nie potrafi obsłużyć atrybutów numerycznych ani brakujących wartości atrybutów. Dlatego dane trzeba najpierw poddać obróbce.

4. Wykorzystaj węzeł *Column Filter* do usunięcia atrybutów numerycznych. Wykorzystaj węzeł *Missing Values* do usunięcia przykładów z wartościami brakującymi (w tej kolejności).

5. Obejrzyj wygenerowane reguły.

6. Na przestrzeń roboczą wstaw węzeł *Weka Predictor* (*Analytics* → *Integrations* → *Weka* → *Weka (3.7)* → *Predictors*). Jego wejścia połącz odpowiednio z wyjściami węzłów *Prism* i *Missing Values*. Do wyjścia węzła *Weka Predictor* dołącz węzły *Scatter Plot* i *Interactive Table*.

7. Wykres punktowy skonfiguruj tak, aby pokazywały wartość *Class* oraz wartość przewidzianą *Prediction (Class).*

8. Oznacz dane błędnie zaklasyfikowane (poprzez zaznaczenie obszaru i wybranie z menu kontekstowego *Hilite selected*) i obejrzyj za pomocą węzła *Interactive Table*.

9. Wstaw na przestrzeń roboczą węzeł *Analytics* → *Mining* → *Weka* → *Weka (3.7)* → *Classification Algorithms* → *rules* → *PART*. Ten komponent buduje reguły, generując częściowe drzewa decyzyjne.

10. Do wejścia węzła *PART* podłącz węzeł *ARFF Reader*. Obejrzyj wygenerowane reguły.

11. Obejrzyj opcje konfiguracyjne węzła *PART*.