11.1.
Wzorcowy eksperyment *data science*

W eksperymencie będącym podsumowaniem książki raz jeszcze dokonamy klasyfikacji irysów na podstawie rozmiarów ich płatków i kielichów. Dane źródłowe do tego eksperymentu znamy już z rozdziałów 5 i 7, wiemy więc, że zawierają one informacje o wielkości płatków i kielichów trzech typów irysów, po 50 przypadków każdego typu, i że tylko dwa typy irysów są od siebie liniowo separowalne.

Po wstępnym przygotowaniu danych polegającym na nadaniu zmiennym opisowych nazw i usunięciu pustego wiersza dane są gotowe do dalszej analizy. Na tym etapie możemy stwierdzić, że o ile wielkości płatków mają dużą siłę predykcyjną, o tyle wielkości kielichów są słabiej skorelowane ze zmienną wyjściową (typem kwiatu).

Zanim jednak, kierując się zasadą, że prostsze modele są lepsze (wiemy, że usuwając mniej przydatne zmienne wejściowe, zapobiegamy przetrenowaniu modelu), usuniemy zmienne *SepalLength* i *SepalWidth*, sprawdzimy, czy nie można użyć ukrytych w nich informacji. Ocenimy na przykład siłę predykcyjną ilorazu długości i szerokości kielichów względem długości i szerokości płatków. W tym celu do eksperymentu należy:

- dodać moduł *Apply Math Operation*;
- skonfigurować dodany moduł, ustawiając parametry: *Category* na *Operation*, *Basic operation* na *Divide*, *Operation argument* na *PetalLength, PetalWidth*, *Column set* na *SepalLength, SepalWidth* i *Output mode* na *Append*;
- dodać do eksperymentu moduł *Filter Based Feature Selection* i połączyć go z wyjściem modułu *Apply Math Operation*;
- skonfigurować dodany moduł, ustawiając parametry: *Feature scoring method* na *Chi Squared*, *Target column* na *Type*.

Po uruchomieniu eksperymentu na drugim wyjściu modułu *Filter Based Feature Selection* będzie dostępny raport zawierający zmierzone miarą chi-kwadrat zależności między zmiennymi wejściowymi a wyjściową (rys. 11.1).

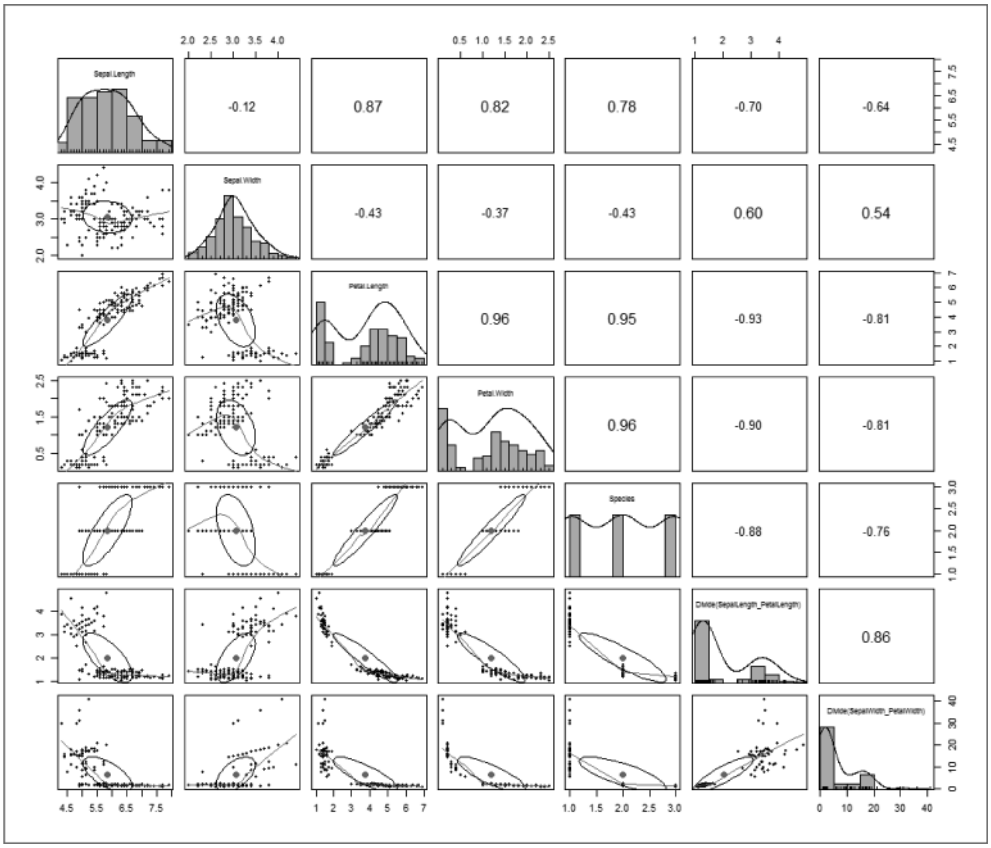
Iris wersja finalna › Filter Based Feature Selection › Features

rows 1 columns 7

	Type	PetalWidth	PetalLength	Divide(SepalWidth_PetalWidth)	Divide(SepalLength_PetalLength)	SepalLength	SepalWidth
view as							
	1	253.033333	252.079487	247.089286	228.436538	129.462524	68.450918

Rysunek 11.1. Z rozdziału 3 wiemy, że większe wartości miary chi-kwadrat oznaczają silniejszą zależność zmiennej wyjściowej od wejściowej

W rozdziale 2 dowiedzieliśmy się, że najlepszym sposobem sprawdzenia zależności między zmiennymi jest ich wizualizacja – w tym przypadku użyjemy funkcji `pairs.panels` wchodzącej w skład biblioteki *psych*. Wiemy, że funkcja ta wywołana dla zbioru danych źródłowych zwraca macierz wykresów zawierających rozkład poszczególnych zmiennych, współczynniki korelacji Pearsona między zmiennymi oraz krzywe reprezentujące kształt zależności między zmiennymi (rys. 11.2).



Rysunek 11.2. Obie zmienne wyliczeniowe nie tylko są silniej skorelowane ze zmienną wyjściową, ale przede wszystkim lepiej rozróżniają typy *Iris-Virginica* od *Iris-Versicolor*

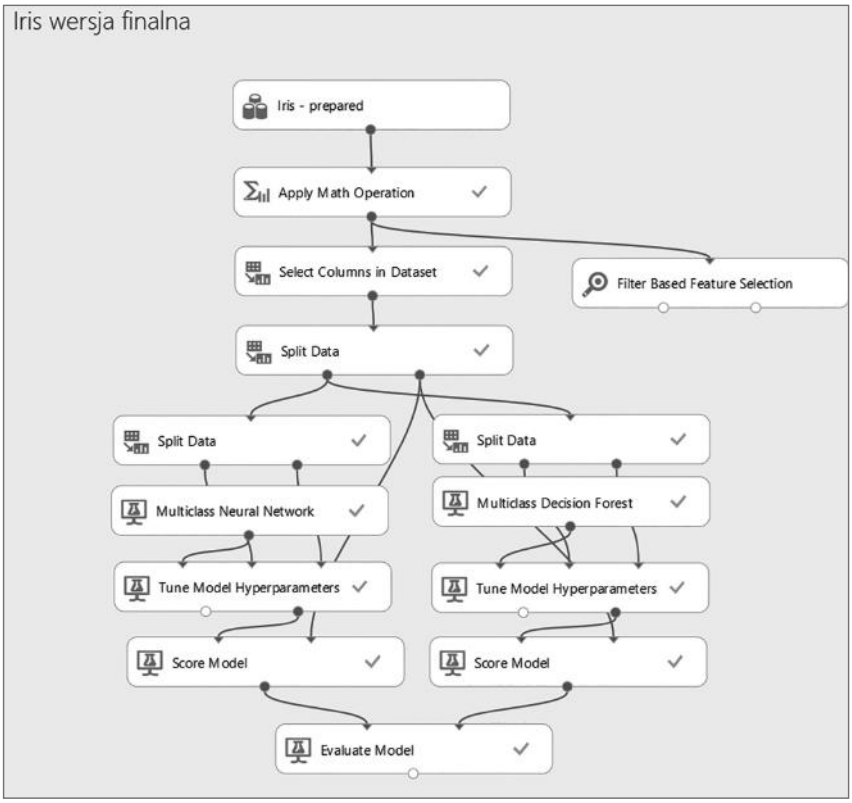
►► Zanim usuniemy zmienne wejściowe, należy sprawdzić, czy ukrytej w nich informacji nie można wydobyć, tworząc na ich podstawie zmiennych pochodnych.

Ponieważ ta sama informacja nie powinna być prezentowana przez różne zmienne wejściowe, następnym krokiem jest usunięcie ze zbioru danych źródłowych zmiennych *SepalLength* i *SepalWidth* – do tego celu posłuży moduł *Select Columns in Dataset*.

Jako że mamy do czynienia z problem klasyfikacyjnym, dane źródłowe podzielimy metodą doboru warstwowego na zbiór treningowy, testowy i kontrolny – w ten sposób będziemy mieć pewność, że proporcje liczby kwiatów poszczególnych typów w każdym z trzech zbiorów będą takie same. Do tego celu zastosujemy dwa podobnie skonfigurowane moduły *Split Data*, w których parametry ustawimy następująco:

- *Splitting mode* na *Split Rows*;
- *Fraction of rows in the first output dataset* na `0,7` dla pierwszego i na `0,6` dla drugiego modułu;
- *Random seed* na `1234`;
- *Stratification key column* na *Type*.

Uwzględnimy w nich też opcje *Randomized split* oraz *Stratified split*, a wejście drugiego modułu *Split Data* połączymy z pierwszym wyjściem poprzedzającego go modułu *Split Data*.



Rysunek 11.3. Wzorzec eksperymentu *data science* – dane źródłowe zostały w nim wstępnie przygotowane, wzbogacone i podzielone, a dwa wybrane modele eksploracji danych zoptymalizowane, ocenione i porównane ze sobą

W ten sposób otrzymamy:

- na pierwszym wyjściu drugiego modułu zbiór danych treningowych liczący 63 przypadki (150 · 0,7 · 0,6);
- na drugim wyjściu drugiego modułu zbiór danych treningowych liczący 42 przypadki (150 · 0,7 · 0,4);
- na drugim wyjściu pierwszego modułu zbiór danych kontrolnych liczący 45 przypadków (150 · 0,3).

Do rozwiązania problemu wybieramy sztuczną sieć neuronową (moduł *Multi-Class Neural Network*) oraz las drzew decyzyjnych (moduł *Multi-Class Decision Forest*). Parametry obu algorytmów uczenia maszynowego zostaną automatycznie dobrane za pomocą modułów *Tune Model Hyperparameter*. Ich konfiguracja jest następująca:

- parametr *Specify parameter sweeping* – ustawiony na *Entire Grid*;
- *Label column* – na *Type*;
- *Metric for measuring performance for classification* – na *F-score*.

Danych kontrolnych użyjemy do predykcji, którą wykonamy przy pomocy modułu *Score Model*, a jakość obu modeli zmierzymy za pomocą modułu *Evaluate Model* (rys. 11.3).

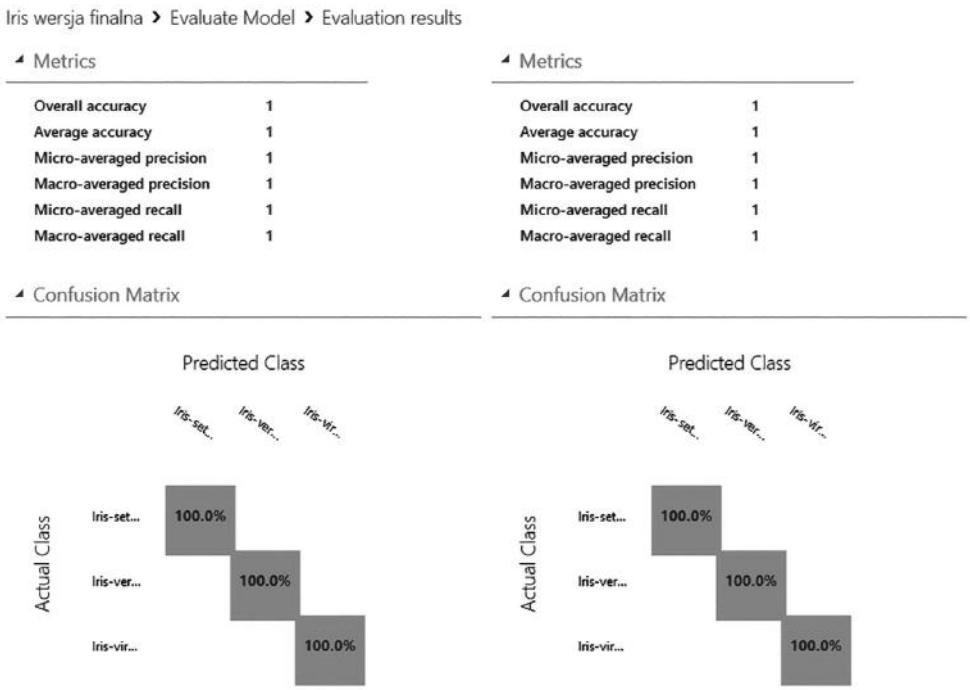
W wyniku przeprowadzonego eksperymentu okazało się, że oba modele eksploracji danych są idealne – żaden z nich nie popełnił ani jednego błędu, klasyfikując przypadki kontrolne (rys. 11.4).

Modele klasyfikacyjne zwracają nie tylko wyniki predykcji, czyli klasę, do której zaklasyfikowany został dany przypadek, ale również prawdopodobieństwa, z jakimi ten przypadek został oceniony jako należący do poszczególnych klas. Analizując te prawdopodobieństwa, możemy wybrać lepszy model, tj. **bardziej pewny swoich decyzji**. W najprostszym przypadku możemy obliczyć średnie prawdopodobieństwo poprawnych klasyfikacji. Oprócz tego ta sama metoda może być użyta do obliczenia średniego prawdopodobieństwa błędnych klasyfikacji.

Prawdopodobieństwo poprawnych klasyfikacji powinno być jak największe – dobry model to taki, który nie tylko się nie myli, ale w dodatku jest pewny swoich decyzji. Analogicznie średnie prawdopodobieństwo błędnych predykcji powinno być jak najmniejsze (skoro model popełnił błąd, to nie powinien być pewny tych decyzji). Ponieważ prawdopodobieństwa, z jakimi przypadek został zaklasyfikowany do poszczególnych klas, są zwracane w kolejnych kolumnach na wyjściu modułu Score Model, do obliczenia średniego prawdopodobieństwa prawidłowych predykcji, czyli miary pewności modelu, zastosujemy moduł *Execute R Script* (listing 11.1).

Listing 11.1. Do trzech wektorów są przypisywane prawdopodobieństwa prawidłowych predykcji odczytane z odpowiednich kolumn, a następnie wektory te są łączone i jest obliczana ich średnia

```
dataset1 <- maml.mapInputPort(1) # class: data.frame
virginica <- dataset1[dataset1[,9] == "Iris-virginica",8]
versicolor <- dataset1[dataset1[,9] == "Iris-versicolor",7]
setosa <- dataset1[dataset1[,9] == "Iris-setosa",6]
data.set <- append(virginica,versicolor)
data.set <- append(data.set,setosa)
data.set <- mean(data.set)
data.set <- as.data.frame(data.set)
maml.mapOutputPort("data.set");
```



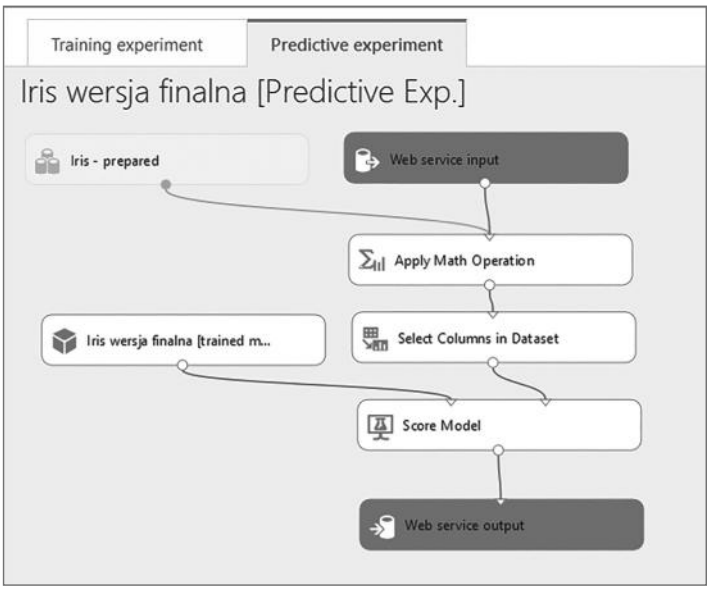
Rysunek 11.4. Oba modele bezbłędnie sklasyfikowały wszystkie 45 przypadków kontrolnych

Ponieważ las drzew decyzyjnych okazał się pewniejszy swoich decyzji (ze średnim prawdopodobieństwem prawidłowych predykcji 0,887827) od sieci neuronowej (dla której średnie prawdopodobieństwo prawidłowych predykcji wyniosło 0,803067), w dodatku był od niej szybszy, ten model eksploracji danych został też użyty w dalszej części rozdziału do utworzenia predykcyjnych usług WWW.

11.2. Predykcyjne usługi WWW

Pierwszy krok procesu publikacji predykcyjnej usługi WWW polega na utworzeniu finalnej wersji modelu eksploracji danych. W tym celu należy uruchomić cały eksperyment – jeżeli wszystkie moduły zostaną wykonane bezbłędnie, uaktywni się znajdujący się na dolnym pasku Studio Azure ML przycisk *SETUP WEB SERVICE*.

Jako że nasz eksperyment zawiera wiele modeli eksploracji danych, musimy wybrać ten, który zostanie wdrożony do produkcji. W tym przypadku będzie to model lasu drzew decyzyjnych sparametryzowany przy użyciu modułu *Tune Model Hyperparameter*, a więc należy zaznaczyć ten moduł. Po kliknięciu przycisku *SETUP WEB SERVICE* wystarczy wybrać opcję *Predictive Web Services (Recommended)*, żeby został utworzony eksperyment predykcyjny (rys. 11.5).



Rysunek 11.5. Automatycznie utworzona predykcyjna wersja eksperymentu

Ta wersja eksperymentu zawiera tylko niezbędne do wykonywania zapytań predykcyjnych moduły:

- *Web service input* – pozwala wysłać do modelu zapytanie predykcyjne; jedynym parametrem tego modułu jest nazwa usługi WWW;
- *Web service output* – umożliwia odebranie od modelu wyniku zapytania predykcyjnego; jedynym parametrem tego modułu jest nazwa usługi WWW;

- *Trained model* – zawiera nauczony, wybrany model eksploracji danych; moduł ten został automatycznie utworzony i zapisany w Studiu Azure ML, a więc jest on dostępny również w ramach innych eksperymentów.
- *Score Model* – pozwalający wykonać zapytanie predykcyjne wysłane do modułu Web service input.

Źródło danych jest dodane do eksperymentu tylko w celu ustalenia metadanych i nie jest ono odczytywane podczas wykonywania zapytań predykcyjnych. Jeżeli dane są w jakiś sposób przygotowane, predykcyjna wersja eksperymentu będzie zawierała moduły, które pozwolą automatycznie przekształcić użyte w zapytaniu predykcyjnym wartości.

Dostępność metadanych (informacji o nazwach i typach zmiennych) ułatwia dostosowywanie predykcyjnej wersji eksperymentu do własnych potrzeb. Choć eksperymenty tego typu można dowolnie modyfikować, najczęściej ich modyfikacje sprowadzają się do określenia formatu obu usług WWW (wejściowej i wyjściowej). W najprostszym przypadku polega to na ograniczeniu liczby kolumn: dane wejściowe nie powinny zawierać kolumny wyjściowej, a dane wyjściowe często ogranicza się do wyników predykcji. Obie te operacje można przeprowadzić przy użyciu modułu *Select Columns in Dataset*.

Żeby usunąć z listy danych wejściowych zmienną objaśnianą, należy podłączyć moduł *Web service input* do wyjścia odpowiednio skonfigurowanego modułu *Select Columns in Dataset*. Ograniczenie zwracanych danych do wyników predykcji polega zaś na podłączeniu modułu *Web service output* do wyjścia odpowiednio skonfigurowanego modułu *Select Columns in Dataset* (rys. 11.6).

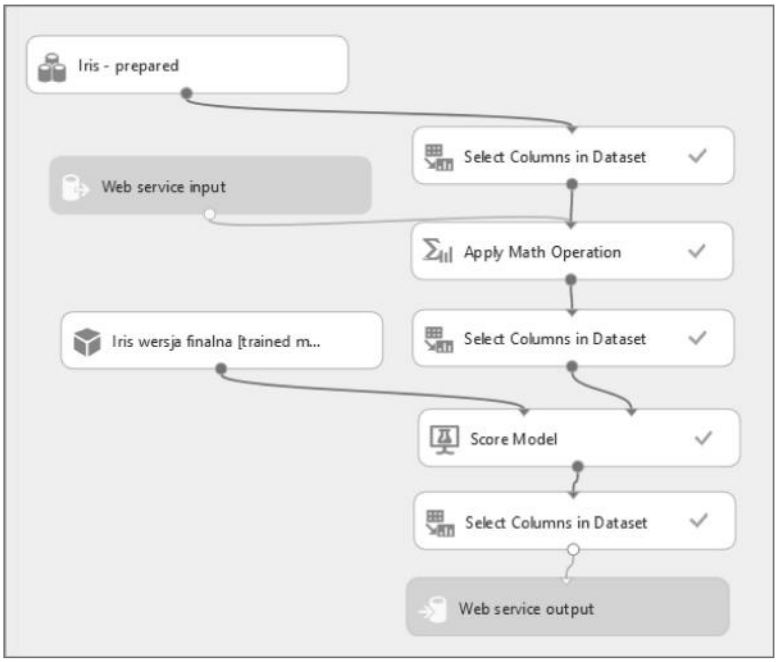
Po uruchomieniu całego eksperymentu predykcyjnego uaktywni się znajdujący się na dolnym pasku Studia Azure ML przycisk *DEPLOY WEB SERVICE*. Jego kliknięcie spowoduje utworzenie i opublikowanie predykcyjnej usługi WWW.

Do tej pory pracowaliśmy głównie z dostępnymi w sekcji *EXPERIMENTS* eksperymentami *data science* oraz wgranymi do Studia Azure ML, dostępnymi w sekcji *DATASET* zbiorami danych i bibliotekami *R*. Usługi WWW są dostępne w sekcji *WEB SERVICES*. Po kliknięciu znajdującej się w niej usługi WWW zostanie wyświetlona główna strona usługi.

W zakładce *CONFIGURATION* można podać nazwę i opis usługi WWW oraz opisy jej poszczególnych parametrów. Znacznie więcej danych i opcji znajdziemy w zakładce *DASHBOARD* pokazanej na rysunku 11.7. Można tam:

- wyświetlić zapisaną i ostatnią wersję eksperymentu predykcyjnego (odnośniki *View snapshot* i *View latest*);
- skopiować do schowka klucz API (klucz potrzebny, żeby programowo wywołać usługę);
- wyświetlić w osobnej zakładce dokumentację usługi umożliwiającą wykonywanie zapytań ad-hoc i wsadowych (odnośniki *REQUEST/RESPONSE* i *BATCH EXECUTION*);

- przetestować usługę, wysyłając do niej zapytanie ad-hoc (przycisk *Test*);
- przetestować usługę za pomocą lokalnie zainstalowanego arkusza Excel.



Rysunek 11.6. Zmodyfikowany eksperyment predykcyjny – od teraz użytkownik będzie musiał podać jedynie wielkości płatka i kielicha, a otrzyma wynik predykcji i wartości prawdopodobieństwa, z jakimi dany kwiat należy do poszczególnych klas irysów

Rysunek 11.7. Po kliknięciu przycisku *Test* zostanie wyświetlony formularz, w którym można podać wartości zmiennych *SepalLength*, *SepalWidth*, *PetalLength* i *PetalWidth* oraz wysłać zapytanie predykcyjne do modelu eksploracji danych