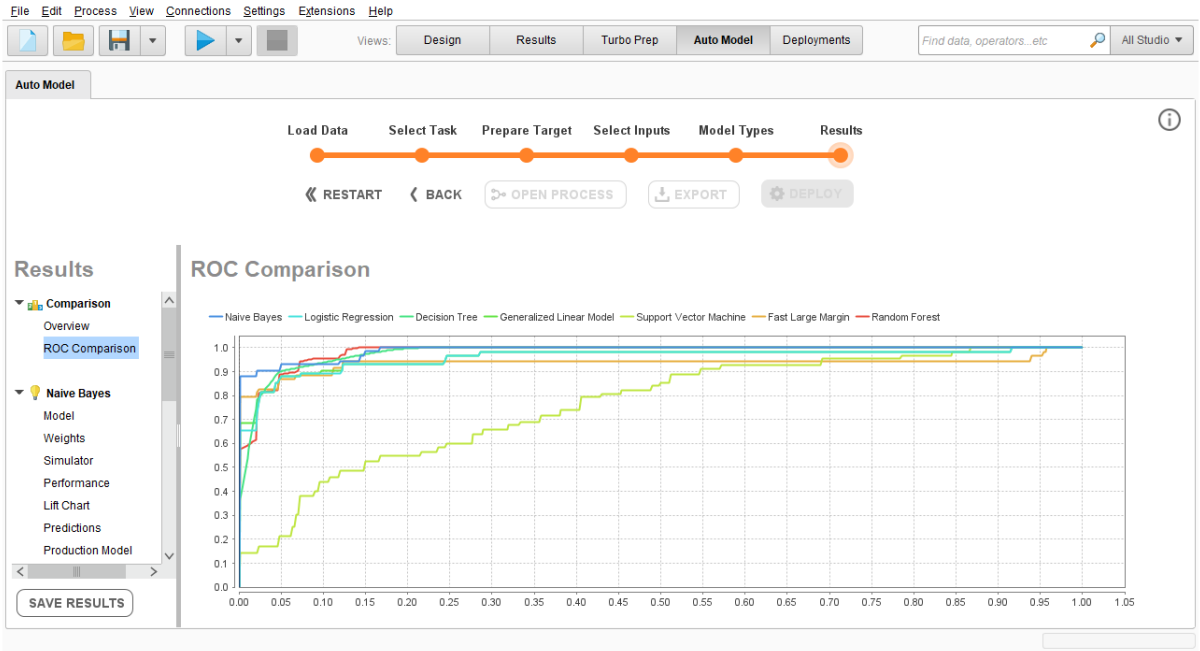
Mateusz Guściora, 228884, zadanie 3.1

a) Przykładowy proces-Direct Marketing

Po zapoznaniu się z strukturą procesu (Sample\Templates), który wykorzystuje dwa zbiory danych najpierw historyczne czyli Past o klientach a potem zbiór New o nowych klientach. Dla walidacji krzyżowej jest algorytm Naive Baes oraz zastosowanie tego algorytmu(apply model)

Po uruchomieniu procesu zapoznajemy się z wynikami w zakładce results. Widzimy, że pojawiła się zmienna prediction tzn. predykcja tej zmiennej którą wybraliśmy czy klient będzie chętny(czy zakupi) czy nie („yes", „no"). Pojawiły się też kolumny confidence. Zapoznajemy się również z wizualizacjami(zakładka results i visualizations). Interfejs jest czytelny i prosty. Możemy wybierać zmienne, które pokaża się na danej osi, typ wykresu, color innej zmiennej itd. Następnie w tym samym procesie zmieniamy algorytm w cross validation na decission tree. Zostały zmienione parametry algorytmu aby skrócić drzewo. Po uruchomieniu procesu możemy zapoznać się z results. Następnie w widoku results uruchamiamy automodel. W predykcji jako zmienną, do przewidzenia wybieramy email chcąc się dowiedzieć czy klient wykupi premium czy pozostanie na free (jako zmienną highest interest wybieramy premium). Wyniki zapisujemy do pliku csv wyniki accuracy(bo wybraliśmy dokładność jako wskaźnik). Zapisujemy też jako obraz (export image) wykres krzywych ROC danych algorytmów.



Algorytmy dają podobną jakość predykcji(chociaż suport vector machine odstaje na niekorzyść).

Następnie dla algorytmu drzewa decission tree uruchamiamy (open proces) proces. Łączymy proces, oraz dodajemy write as text na końcu procesu aby zapisać wyniki(plik unnamed2 wyniki.txt)

b) Nowy proces dla klienci

Przed przystąpieniem do tego zadania zmieniłem plik 228884_klienci_2 na 228884_klienci2 ze względów estetycznych. (Nazwa przedział na przedział kwotowy oraz nazwy poszczególnych grup dla przedział kwotowy). Klienci2 nie posiadają zmiennej kwota rozmowy tylko przedziały kwotowe. I to ta zmienna będzie zmienną wyjściową. Ustawimy to poprzez wstawienie read csv i wybraniu opcji

import configuration wizard. Dzielimy zbiór danych na 50/50 i używamy algorytmów rule induction oraz decission tree. Wyniki zapisano do pliku res.

Tworzymy drugi proces z rozszerzeniem z programu Weka używając algorytmu W-JRip. Widzimy, że algorytm W-JRip stworzył nam dużo mniej zasad (3) niż algorymt rule induction. Wyniki zapisano do pliku res.

c) Procesy dla danych glass i bank2

Tworzymy teraz proces dla danych glass i dla danych bank2. Zmienną do predykcji (label) w zbiorze glass wybrałem type (poprzez read csv) a zbiorze bank2 jest nią y. Dla zbioru glass i algorytmu rule induction w cross validation, wyniki są czytelne i zostały stworzone 13 reguł. Dla zbioru bank, który jest dużo większym zbiorem(ponad 40 000) nie udało mi się skrócić liczby reguł. Wyniki zapisano do plików res.

d) Procesy z rozszerzeniem Weka

Po zastosowaniu algorytmu W-JRip dla tych samych danych i porównaniu z algorytmem rule induction widzimy znowu, że W-JRip tworzy mniej reguł i daje czytelniejszy obraz predykcji. Dla danych bankowych jest to 14 reguł a dla danych glass 8. Wyniki zapisano do plików res.