MOW

Kacper Kamiński

Przygotowanie danych

Pierwszą czynnością którą należy wykonać przed przystąpieniem do analizy jest scalenie tabeli zawierających informacje na temat studentów kursu matematyki oraz studentów kursu języka portugalskiego bez duplikowania informacji o studentach uczęszczających na oba kursy.

Podział danych na zbiór treningowy i testowy

Następnie dzielimy nasze dane na zbiór treningowy oraz zbiór testowy. 80% danych zostanie używa do wytrenowania naszych modeli, natomiast 20% do ich testowania.

Przewidywanie spożycia alkoholu w dni robocze

Na początek zajmiemy się przewidywaniem spożycia alkoholu wśród studentów w dni robocze. W tym celu użyjemy funkcji `polr` z pakietu `MASS` do wytrenowania modelu regresji porządkowej. Uważamy że model ten jest bardziej adekwatny do predykcji tego atrybutu, niż przykładowo one-vs-all, ponieważ jego wartości nie są oddalone od siebie w równych odległościach. Nie możemy stwierdzić że między 1 a 2 jest taka sama różnica jak między 4 a 5. Początkowo do regresji atrybutu Dalc użyjemy wszystkich pozostałych atrybutów, z wyjątkiem atrybutu Walc, którego predykcją zajmować się będziemy potem. Pozwoli nam to w pewnym stopniu porównać ze sobą spożycie alkoholu w weekend oraz w tygodniu.

Trenowanie modelu i interpretacja statystyk

Poniżej widnieje wywołanie funkcji tworzącej nasz model.

```
m <- polr(Dalc ~ sex + age + address + famsize + Pstatus + Medu + Fedu + Mjob + Fjob
          + reason + guardian + traveltime + studytime + failures + schoolsup + famsup
          + activities + nursery + higher + internet + romantic + famrel + freetime
          + goout + health + absences + G1 + G2 + G3, data=train, Hess=TRUE,
          method='logistic')
```

Po wytrenowaniu naszego modelu wyświetlimy jego statystyki.

```
## Call:
## polr(formula = Dalc ~ sex + age + address + famsize + Pstatus +
##     Medu + Fedu + Mjob + Fjob + reason + guardian + traveltime +
##     studytime + failures + schoolsup + famsup + activities +
##     nursery + higher + internet + romantic + famrel + freetime +
##     goout + health + absences + G1 + G2 + G3, data = train, Hess = TRUE,
##     method = "logistic")
##
## Coefficients:
##                     Value Std. Error  t value
## sexM              1.08363    0.23312  4.64842
## age               0.11640    0.09911  1.17446
## addressU         -0.50798    0.25543 -1.98873
```

```
## famsizeLE3        0.64607    0.23189  2.78609
## PstatusT          0.54070    0.37230  1.45233
## Medu              0.09498    0.14536  0.65342
## Fedu              0.11307    0.13295  0.85047
## Mjobhealth       -1.13776    0.58577 -1.94235
## Mjobother         0.10807    0.30962  0.34905
## Mjobservices      0.26067    0.36477  0.71460
## Mjobteacher       0.21432    0.49042  0.43702
## Fjobhealth        0.14633    0.73236  0.19980
## Fjobother        -0.16908    0.40981 -0.41258
## Fjobservices      0.23400    0.43347  0.53983
## Fjobteacher      -0.82375    0.66396 -1.24067
## reasonhome        0.21078    0.28140  0.74903
## reasonother       0.97204    0.32386  3.00148
## reasonreputation -0.39542    0.30621 -1.29132
## guardianmother   -0.56048    0.25875 -2.16611
## guardianother     0.23789    0.44392  0.53588
## traveltime        0.01504    0.15741  0.09553
## studytime        -0.15336    0.14278 -1.07412
## failures          0.05779    0.15441  0.37429
## schoolsupyes      0.11482    0.34959  0.32843
## famsupyes         0.09513    0.22278  0.42700
## activitiesyes    -0.28723    0.21802 -1.31746
## nurseryyes       -0.39043    0.25334 -1.54113
## higheryes         0.09433    0.35223  0.26782
## internetyes       0.11766    0.27263  0.43156
## romanticyes      -0.01707    0.22397 -0.07622
## famrel           -0.46433    0.11112 -4.17858
## freetime          0.10806    0.11156  0.96866
## goout             0.58748    0.10303  5.70221
## health            0.18470    0.07967  2.31823
## absences          0.03421    0.01419  2.41131
## G1               -0.04243    0.06631 -0.63989
## G2               -0.01382    0.08348 -0.16558
## G3                0.02356    0.06609  0.35643
##
## Intercepts:
##     Value   Std. Error t value
## 1|2  4.3747  2.0132     2.1730
## 2|3  5.7846  2.0227     2.8598
## 3|4  6.8898  2.0386     3.3796
## 4|5  7.7702  2.0571     3.7773
##
## Residual Deviance: 850.9856
## AIC: 934.9856
```

W dostępnych tabelkach widać odpowiednio:

- Tabelkę współczynników zawierającą ich wartości, odchylenie standardowe oraz t-wartość.
- Tabelkę punktów przecięcia, także zawierającą ich wartości, odchylenie standardowe oraz t-wartość. Punkty przecięcia mówią nam w których miejscach następuje zmiana klas, jednakże nie będą one przydatne w naszej analizie.
- Residual Deviance mówiące jak dobrze nasz model dopasowany jest do naszych danych.
- AIC(Akaike information criterion) - jest to wartość określająca względną jakość modelu dla określonych danych. Jest ona przydatna przy podejmowaniu decyzji, który z naszych modeli wytrenowanych na tym

samym zbiorze danych wybrać. Zwyczajowo powinniśmy wybrać model z najmniejszą wartością AIC.

Interesującym faktem wartym zaznaczenia jest także to, że funkcja `polr` w celu wytrenowania modelu rozbiła automatycznie nasze niebinarne, dyskretne atrybuty na wiele binarnych atrybutów zamiennych. Przykładowo atrybut `reason` określający powód dla którego student wybrał szkołę w której przeprowadzono ankietę, został rozbity na atrybuty `reasonhome`, `reasonother` oraz `reasonreputation`.

Badanie dopasowania modelu do danych treningowych

Jak widać ze statystyk podana jest także wartość `Residual Deviance`. Możemy na podstawie tej wartości oraz wartości stopni swobody naszego modelu wykonać test chi-kwadrat i przekonać się czy nasza hipoteza zerowa zakładająca, że nasz model dobrze oddaje nasze dane, jest prawdziwa.

W tym celu znajdujemy ilość stopni swobody.

```
## [1] "Ilość stopni swobody modelu: 487"
```

A następnie wykonujemy test chi-kwadrat.

```
## [1] "Wynik testu chi-kwadrat wartości residual deviance: p-value = 0.00"
```

W wyniku naszego testu otrzymaliśmy p-wartość równą zero. Wynik ten oznacza że musimy odrzucić naszą hipotezę, że nasz model jest dobrze dopasowany do danych treningowych. Nie oznacza to jednakże, że jest on zły. W przypadku naszych danych pokazuje on, że przy dużej ilości atrybutów ciężko jest dopasować linię regresji tak, aby była ona zbliżona do wszystkich naszych przykładów.

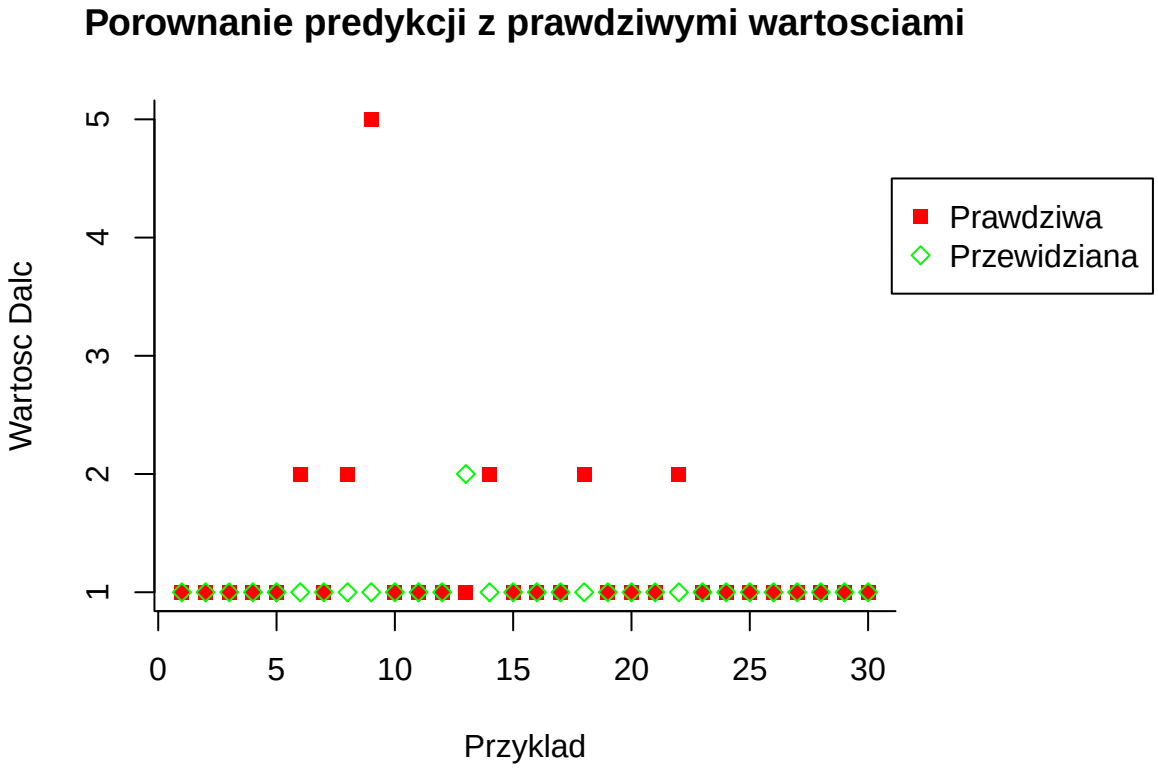
Dla potwierdzenia założenia, że pomimo powyższego wyniku testu chi-kwadrat nasz model regresji może być użyteczny obliczymy jego dokładność na zbiorze trenującym oraz na zbiorze testowym.

```
## [1] "Dokładność modelu na zbiorze treningowym wynosi: 70.51%"
```

```
## [1] "Dokładność modelu na zbiorze testowym wynosi: 68.42%"
```

Jak widać po wynikach nasz model okazał się być dość dokładny i dla zbioru treningowego jak i dla testowego. Pokazuje to że pomimo jego niedopasowania jest on w stanie dokonać dokładnych predykcji. Dodatkowo możemy zauważyć że dokładność naszego modelu na zbiorze testowym jest zbliżona do dokładności na zbiorze treningowym. Jest to korzystny fakt, ponieważ oznacza on, że nie doszło do sytuacji nadmiernego dopasowania do zbioru treningowego, kosztem dokładności na zbiorze testowym.

Poniżej pokazano wykres przewidywanych oraz prawdziwych wartości dla pierwszych 30 przykładów na zbiorze testowym.



Oglądając wykres możemy zacząć podejrzewać, że nasz model ma problem z przewidywaniem wyższych poziomów spożycia alkoholu. Żeby się przekonać czy jest to prawda wyświetlimy tablicę pomyłek.

```
##     predicted
## real  1  2  3  4  5
##    1 87  5  0  0  1
##    2 24  4  0  0  1
##    3  4  2  0  0  0
##    4  2  0  0  0  0
##    5  2  1  0  0  0
```

Z tablicy pomyłek wynika że nasz model stosunkowo dobrze radzi sobie z predykcją niskiego spożycia alkoholu, jednakże nieco gorzej idzie mu z predykcją wysokiego spożycia. Na 133 przykłady gdzie poziom spożycia alkoholu wynosił 2 lub więcej nasz model pomylił się aż w ponad 20 przypadkach przydzielając je do klasy 1 - niskiego spożycia alkoholu. Jak widać ma on tendencję do zaniżania rzeczywistych wartości.

Analiza p-wartości współczynników

Aby zbadać poziom istotności naszych zmiennych niezależnych obliczymy ich p-wartości używając z-testu (możemy go użyć w praktyce pomimo niespełnienia wszystkich formalnych wymagań z powodu dużej liczby przykładów w zbiorze trenującym) a następnie dołączamy je do tabeli podsumowującej nasz model.

P-wartość mówi nam czy hipoteza zerowa mówiąca że atrybut jest nieskorelowany z naszym przewidywanym atrybutem jest prawdziwa czy nie. Często przed wyliczeniem p-wartości dla każdego atrybutu, ustala się poziom istotności α powyżej którego atrybut jest uznawany za nieistotny statystycznie i odrzucany z modelu. Możemy ten sam eksperyment wykonać także i dla naszego modelu. Przyjmiemy w tym celu standardową wartość $\alpha = 5\%$

```
##                       Value Std. Error     t value      p value
## sexM             1.08362830 0.23311750  4.64842105 3.344856e-06
## age              0.11639783 0.09910767  1.17445829 2.402115e-01
```

```
## addressU         -0.50797701 0.25542743 -1.98873321 4.673066e-02
## famsizeLE3        0.64606781 0.23189024  2.78609311 5.334755e-03
## PstatusT          0.54069863 0.37229685  1.45233203 1.464093e-01
## Medu              0.09497953 0.14535808  0.65341760 5.134871e-01
## Fedu              0.11307423 0.13295468  0.85047192 3.950628e-01
## Mjobhealth       -1.13776059 0.58576541 -1.94234855 5.209492e-02
## Mjobother         0.10807363 0.30961992  0.34905259 7.270498e-01
## Mjobservices      0.26066575 0.36477365  0.71459588 4.748588e-01
## Mjobteacher       0.21432426 0.49041655  0.43702494 6.620933e-01
## Fjobhealth        0.14632789 0.73235543  0.19980448 8.416335e-01
## Fjobother        -0.16907837 0.40980518 -0.41258231 6.799127e-01
## Fjobservices      0.23400129 0.43347413  0.53982757 5.893159e-01
## Fjobteacher      -0.82375217 0.66395883 -1.24066753 2.147286e-01
## reasonhome        0.21077678 0.28139785  0.74903481 4.538362e-01
## reasonother       0.97204485 0.32385500  3.00148166 2.686692e-03
## reasonreputation -0.39541617 0.30621092 -1.29131965 1.965929e-01
## guardianmother   -0.56048011 0.25874924 -2.16611302 3.030255e-02
## guardianother     0.23788693 0.44392066  0.53587712 5.920435e-01
## traveltime        0.01503723 0.15740501  0.09553211 9.238922e-01
## studytime        -0.15336322 0.14277992 -1.07412316 2.827675e-01
## failures          0.05779436 0.15441122  0.37428864 7.081896e-01
## schoolsupyes      0.11481528 0.34959082  0.32842763 7.425884e-01
## famsupyes         0.09512881 0.22278150  0.42700499 6.693757e-01
## activitiesyes    -0.28722908 0.21801683 -1.31746288 1.876835e-01
## nurseryyes       -0.39042678 0.25333863 -1.54112613 1.232861e-01
## higheryes         0.09433364 0.35222506  0.26782206 7.888363e-01
## internetyes       0.11765548 0.27262578  0.43156403 6.660583e-01
## romanticyes      -0.01707085 0.22396744 -0.07622021 9.392439e-01
## famrel           -0.46433184 0.11112182 -4.17858395 2.933297e-05
## freetime          0.10806499 0.11156119  0.96866111 3.327143e-01
## goout             0.58748125 0.10302691  5.70221183 1.182627e-08
## health            0.18470468 0.07967486  2.31823017 2.043681e-02
## absences          0.03420632 0.01418577  2.41131207 1.589524e-02
## G1               -0.04242847 0.06630587 -0.63989003 5.222441e-01
## G2               -0.01382227 0.08348029 -0.16557525 8.684912e-01
## G3                0.02355813 0.06609398  0.35643389 7.215157e-01
## 1|2               4.37471875 2.01324591  2.17296790 2.978273e-02
## 2|3               5.78458300 2.02270105  2.85983092 4.238669e-03
## 3|4               6.88978242 2.03864395  3.37959085 7.259381e-04
## 4|5               7.77023158 2.05707600  3.77731868 1.585258e-04
```

Po obliczeniu p-wartości dla naszych współczynników, usuwamy z tabelki te które przekraczają naszą ustaloną wartość $\alpha = 5\%$ i wyświetlamy pozostałe współczynniki.

```
##                     Value Std. Error   t value      p value
## sexM           1.08362830 0.23311750  4.648421 3.344856e-06
## addressU      -0.50797701 0.25542743 -1.988733 4.673066e-02
## famsizeLE3     0.64606781 0.23189024  2.786093 5.334755e-03
## reasonother    0.97204485 0.32385500  3.001482 2.686692e-03
## guardianmother -0.56048011 0.25874924 -2.166113 3.030255e-02
## famrel        -0.46433184 0.11112182 -4.178584 2.933297e-05
## goout          0.58748125 0.10302691  5.702212 1.182627e-08
## health         0.18470468 0.07967486  2.318230 2.043681e-02
## absences       0.03420632 0.01418577  2.411312 1.589524e-02
## 1|2            4.37471875 2.01324591  2.172968 2.978273e-02
```

```
## 2|3             5.78458300 2.02270105  2.859831 4.238669e-03
## 3|4             6.88978242 2.03864395  3.379591 7.259381e-04
## 4|5             7.77023158 2.05707600  3.777319 1.585258e-04
```

Dla eksperymentu możemy utworzyć nowy model składający się jedynie z pozostałych współczynników i przekonać się czy powstały model straci na dokładności z powodu mniejszej ilości atrybutów czy też nie, a także czy będzie on lepiej dopasowany do naszych danych czy nie.

Poniżej pokazane jest wywołanie funkcji tworzącej drugi model.

```
m2 <- polr(Dalc ~ sex + famsize + reason + guardian + famrel + goout
           + absences, data=train, Hess=TRUE, method='logistic')
```

Najpierw wykonujemy test chi-kwadrat aby sprawdzić jak dobrze dopasowany jest on do naszych danych a następnie wyświetlimy p-wartość naszego pierwszego modelu oraz p-wartość naszego drugiego modelu.

```
## [1] "P-wartość pierwszego modelu: 0.00"
```

```
## [1] "P-wartość drugiego modelu: 0.00"
```

Jak widać nawet nasz drugi model nie przeszedł testu dopasowania chi-kwadrat. Tak jak i poprzednio nie znaczy to jednakże że jest on błędny, a jedynie to że przy takiej ilości zmiennych niezależnych i rozrzuceniu wartości, dopasowanie naszej krzywej regresji do danych jest bardzo ciężkie.

Następnie obliczymy dokładność naszego drugiego modelu i porównamy ją z dokładnością naszego pierwszego modelu którą obliczyliśmy wcześniej.

```
## [1] "Dokładność pierwszego modelu na zbiorze treningowym wynosi: 70.51%"
```

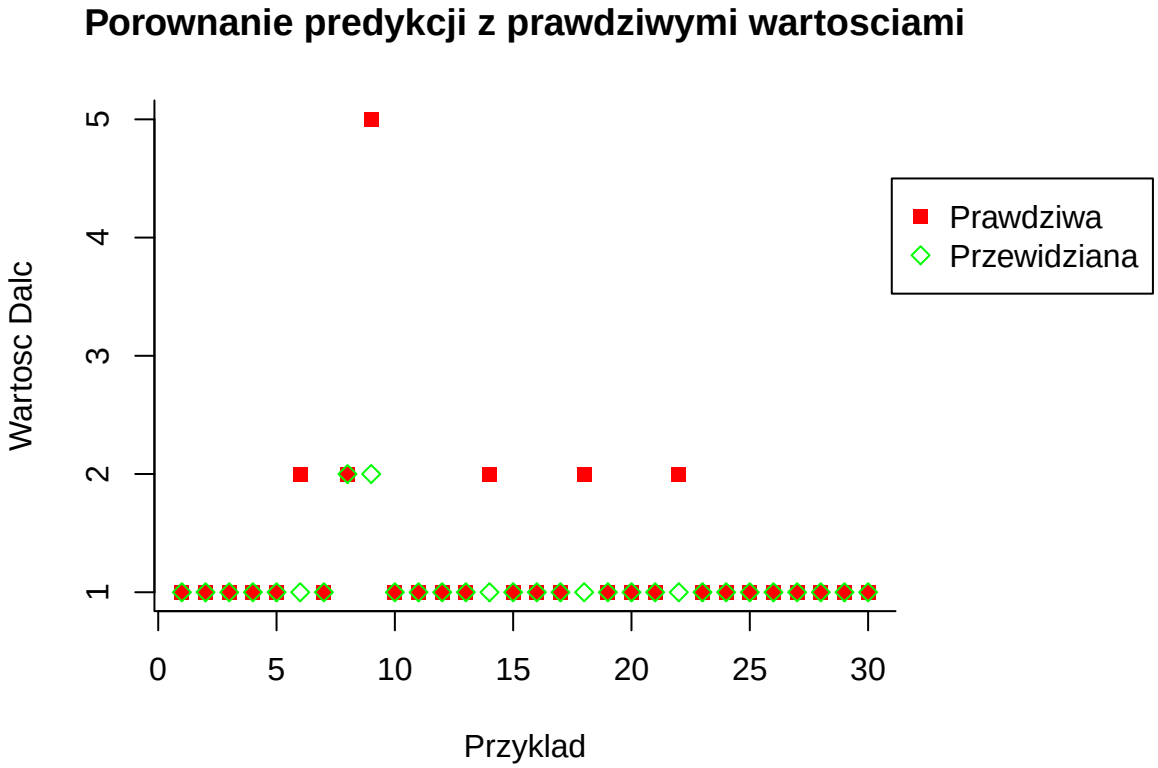
```
## [1] "Dokładność pierwszego modelu na zbiorze testowym wynosi: 68.42%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze treningowym wynosi: 70.51%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze testowym wynosi: 71.43%"
```

Okazuje się że po zmniejszeniu ilości zmiennych niezależnych w naszym modelu, nasz model różni się nieznacznie dokładnością na obu zbiorach.

Także i w tym modelu wyświetlimy wykres przewidywanych oraz prawdziwych wartości dla pierwszych 30 przykładów.



Także i dla tego modelu wyświetlimy tablicę pomyłek, aby przekonać się czy nie zmieniła się tendencja do przewidywania niższego niż w rzeczywistości spożycia alkoholu.

```
##     predicted
## real  1  2  3  4  5
##    1 91  2  0  0  0
##    2 26  3  0  0  0
##    3  4  1  1  0  0
##    4  2  0  0  0  0
##    5  1  2  0  0  0
```

Z tablicy pomyłek wynika że także i ten model posiada ten sam problem.

StepAIC

Jednym z problemów który poruszyliśmy przed chwilą jest wybór których atrybutów używać do tworzenia modelu. Nasze wcześniejsze podejście z pominięciem atrybutów których p-wartość przekracza z góry ustaloną wartość $\alpha = 5\%$ jest nieco naiwna. Lepszym sposobem może być wykorzystanie funkcji stepAIC która tworzy wiele modeli wykorzystując różne kombinacje atrybutów i wyświetla ich wartości AIC. Możemy następnie wybrać model z najniższą wartością AIC który powinien być potencjalnie najlepszy. Należy zauważyć że wartość AIC nie mówi nam jak ogólnie dobry jest nasz model, lecz jak dobry jest względem innych modeli. Przez to jeżeli wszystkie potencjalne modele są słabo dopasowane, będziemy mieli jedynie informację który z nich jest najlepiej dopasowany względem innych modeli, a nie czy jest dobrze dopasowany.

Na podstawie analizy funkcją stepAIC otrzymujemy formułę o najmniejszej wartości AIC. Utworzymy nowy model używając jej.

```
m3 <- polr(formula = Dalc ~ sex + address + famsize + reason + guardian +
    famrel + goout + absences + G3, data = train, Hess = TRUE,
    method = "logistic")
```

Wyświetlimy statystyki.

```
## Call:
## polr(formula = Dalc ~ sex + address + famsize + reason + guardian +
##     famrel + goout + absences + G3, data = train, Hess = TRUE,
##     method = "logistic")
##
## Coefficients:
##                     Value Std. Error t value
## sexM              1.23340    0.20635  5.9772
## addressU         -0.40569    0.22083 -1.8371
## famsizeLE3        0.52191    0.21570  2.4196
## reasonhome        0.10112    0.26563  0.3807
## reasonother       0.90432    0.30198  2.9947
## reasonreputation -0.53152    0.28623 -1.8570
## guardianmother   -0.55934    0.23632 -2.3669
## guardianother     0.41041    0.37142  1.1050
## famrel           -0.35903    0.10241 -3.5059
## goout             0.57363    0.09353  6.1332
## absences          0.03721    0.01302  2.8585
## G3               -0.02554    0.02458 -1.0390
##
## Intercepts:
##     Value   Std. Error t value
## 1|2  1.3861  0.6144     2.2558
## 2|3  2.7148  0.6263     4.3346
## 3|4  3.7656  0.6485     5.8066
## 4|5  4.6120  0.6812     6.7705
##
## Residual Deviance: 882.857
## AIC: 914.857
```

Tak jak uprzednio dokonamy testu chi-kwadrat dopasowania do danych aby przekonać się, czy model jest dobrze dopasowany do danych.

```
## [1] "p-wartość modelu: 0.00"
```

Oraz obliczymy dokładność na zbiorze treningowym i testowym i zestawimy z wartościami poprzednich modeli.

```
## [1] "Dokładność pierwszego modelu na zbiorze treningowym wynosi: 70.51%"
```

```
## [1] "Dokładność pierwszego modelu na zbiorze testowym wynosi: 68.42%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze treningowym wynosi: 70.51%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze testowym wynosi: 71.43%"
```

```
## [1] "Dokładność trzeciego modelu na zbiorze treningowym wynosi: 70.32%"
```

```
## [1] "Dokładność trzeciego modelu na zbiorze testowym wynosi: 72.18%"
```

Jak widać dokładność naszego trzeciego modelu wybranego z użyciem funkcji stepAIC jest zbliżona do dokładności poprzednich dwóch modeli. Wynika z tego że równie dobrą decyzją może być utworzenie modelu używając wszystkich dostępnych atrybutów, tych których p-wartość jest wystarczająco niska albo tych wybranych z użyciem funkcji stepAIC

Przewidywanie spożycia alkoholu w weekend

Analogicznie postaramy się przeprowadzić regresję drugiego atrybutu. Tym razem będzie to regresja spożycia alkoholu w weekend. Tak samo jak i wcześniej zaczniemy od utworzenia modelu z użyciem wszystkich(tym razem z wyjątkiem Dalc) atrybutów i wyświetlimy jego statystyki.

Trenowanie modelu i interpretacja statystyk

Poniżej widnieje wywołanie funkcji tworzącej nasz model.

```
m4 <- polr(Walc ~ sex + age + address + famsize + Pstatus + Medu + Fedu + Mjob + Fjob
          + reason + guardian + traveltime + studytime + failures + schoolsup + famsup
          + activities + nursery + higher + internet + romantic + famrel + freetime
          + goout + health + absences + G1 + G2 + G3, data=train, Hess=TRUE,
          method='logistic')
```

Po wytrenowaniu naszego modelu wyświetlimy jego statystyki.

```
## Call:
## polr(formula = Walc ~ sex + age + address + famsize + Pstatus +
##     Medu + Fedu + Mjob + Fjob + reason + guardian + traveltime +
##     studytime + failures + schoolsup + famsup + activities +
##     nursery + higher + internet + romantic + famrel + freetime +
##     goout + health + absences + G1 + G2 + G3, data = train, Hess = TRUE,
##     method = "logistic")
##
## Coefficients:
##                     Value Std. Error t value
## sexM              1.24884    0.19651  6.3552
## age               0.02529    0.08155  0.3101
## addressU         -0.26802    0.20750 -1.2917
## famsizeLE3        0.42077    0.19231  2.1879
## PstatusT          0.38738    0.27920  1.3875
## Medu             -0.17286    0.12047 -1.4349
## Fedu              0.26488    0.10804  2.4517
## Mjobhealth       -0.22556    0.41573 -0.5426
## Mjobother        -0.33983    0.24892 -1.3652
## Mjobservices     -0.05334    0.30095 -0.1772
## Mjobteacher       0.22556    0.40899  0.5515
## Fjobhealth       -0.29491    0.64907 -0.4544
## Fjobother         0.12092    0.34665  0.3488
## Fjobservices      0.53980    0.36667  1.4722
## Fjobteacher      -0.91760    0.52687 -1.7416
## reasonhome        0.04813    0.23652  0.2035
## reasonother       0.61101    0.28124  2.1725
## reasonreputation  0.18125    0.23046  0.7865
## guardianmother   -0.07419    0.21720 -0.3416
## guardianother    -0.20663    0.39085 -0.5287
## traveltime       -0.05071    0.12775 -0.3969
## studytime        -0.25215    0.11639 -2.1663
## failures          0.06081    0.13706  0.4436
## schoolsupyes     -0.43595    0.30621 -1.4237
## famsupyes        -0.14927    0.18037 -0.8275
## activitiesyes    -0.33538    0.17970 -1.8663
```

```
## nurseryyes       -0.51313    0.21320 -2.4068
## higheryes         0.11784    0.30320  0.3886
## internetyes       0.08878    0.21853  0.4063
## romanticyes      -0.23092    0.18375 -1.2567
## famrel           -0.42692    0.09290 -4.5957
## freetime         -0.17777    0.09067 -1.9606
## goout             0.90370    0.09061  9.9737
## health            0.17103    0.06303  2.7134
## absences          0.04237    0.01335  3.1729
## G1               -0.02734    0.05385 -0.5078
## G2               -0.04620    0.06966 -0.6632
## G3                0.04761    0.05545  0.8585
##
## Intercepts:
##     Value   Std. Error t value
## 1|2  0.7429  1.6540     0.4491
## 2|3  1.9899  1.6557     1.2019
## 3|4  3.2946  1.6598     1.9850
## 4|5  4.9179  1.6705     2.9440
##
## Residual Deviance: 1314.049
## AIC: 1398.049
```

Badanie dopasowania modelu do danych treningowych

Także teraz możemy na podstawie wartości Residual Deviance oraz wartości stopni swobody naszego modelu wykonać test chi-kwadrat i przekonać się czy nasza hipoteza zerowa zakładająca, że nasz model dobrze oddaje nasze dane, jest prawdziwa.

W tym celu znajdujemy ilość stopni swobody.

```
## [1] "Ilość stopni swobody modelu: 487"
```

A następnie wykonujemy test chi-kwadrat.

```
## [1] "Wynik testu chi-kwadrat wartości residual deviance: p-value = 0.00"
```

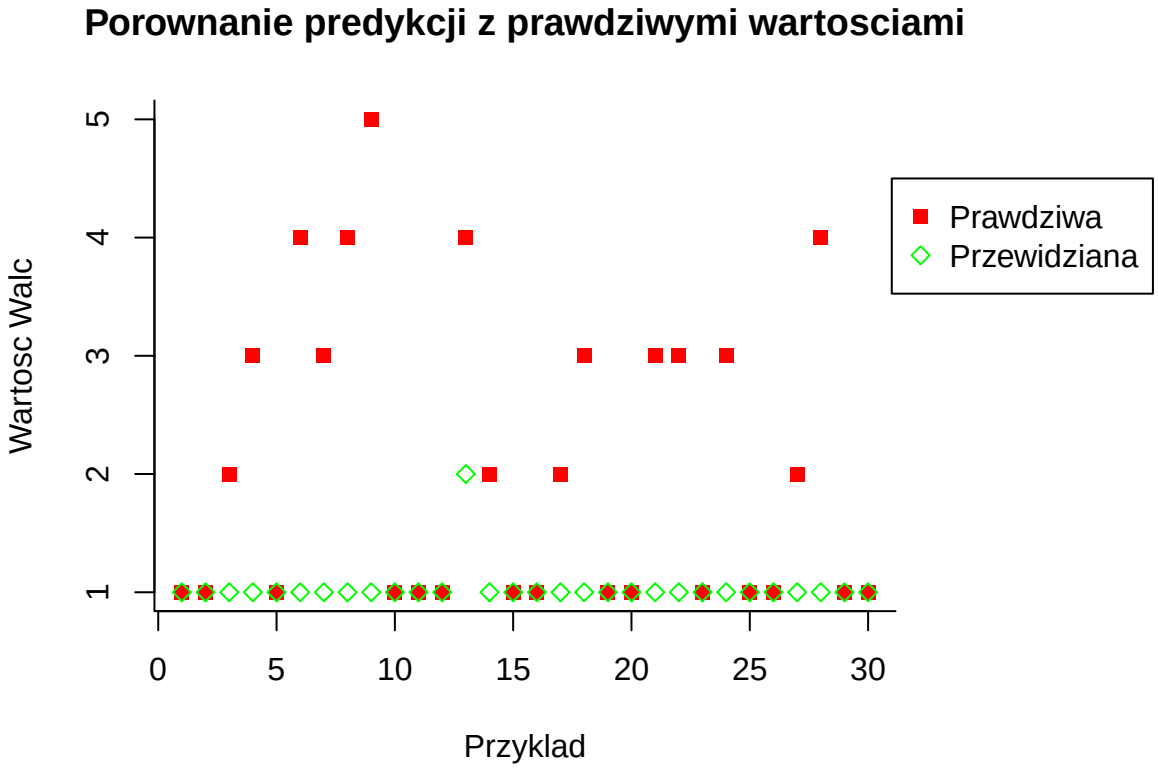
Także i ten model nie przeszedł testu chi-kwadrat. Tak jak i poprzednio przekonamy się jednakże czy udaje mu się przewidzieć z dobrą dokładnością poziom spożycia alkoholu.

```
## [1] "Dokładność modelu na zbiorze treningowym wynosi: 45.94%"
```

```
## [1] "Dokładność modelu na zbiorze testowym wynosi: 37.59%"
```

Okazuje się że model regresji spożycia alkoholu w weekend jest zdecydowanie mniej dokładny niż model regresji spożycia alkoholu w tygodniu. Nie są to jednakże bardzo niskie wartości jakich można by się spodziewać po modelu całkowicie niedopasowanym dla danych.

Poniżej pokazano wykres przewidywanych oraz prawdziwych wartości dla pierwszych 30 przykładów na zbiorze testowym.



Oglądając wykres widzimy już dlaczego dokładność modelu regresji spożycia alkoholu w weekend jest o wiele gorsza. Tak samo jak i w przypadku modeli regresji spożycia alkoholu w tygodniu nasz obecny model ma problem z przewidywaniem niższych poziomów niż w rzeczywistości. W porównaniu jednakże do poprzednich modeli aktualny model jest z tego powodu w gorszej sytuacji, ponieważ rzeczywiste spożycie alkoholu w weekend jest zdecydowanie większe niż w tygodniu. Można podejrzewać że wiąże się to z różnymi imprezami które odbywają się w tym czasie, jednakże jest to nasze podejrzenie niewynikające z samych analizowanych danych.

Wyświetlmy jeszcze tablicę pomyłek dla naszego modelu.

```
##     predicted
## real  1  2  3  4  5
##    1 37  8  7  1  0
##    2 17  4  7  1  1
##    3 14  2  1  7  0
##    4  2  2  6  7  1
##    5  0  0  4  3  1
```

Jak widać z tablicy pomyłek, tendencja modelu do zaniżania poziomu spożycia alkoholu w weekend jest rzeczywista.

Analiza p-wartości współczynników

Spróbujemy przekonać się czy może w przypadku regresji tego atrybutu, utworzenie modelu z użyciem atrybutów o wystarczająca małej p-wartości zwiększy jego dokładność czy nie. Przyjmiemy w tym celu standardową wartość $\alpha = 5\%$

```
##                       Value Std. Error    t value      p value
## sexM             1.24883827 0.19650510  6.3552461 2.080932e-10
## age              0.02528693 0.08155454  0.3100616 7.565141e-01
## addressU        -0.26801656 0.20749920 -1.2916510 1.964780e-01
```

```
## famsizeLE3        0.42076526 0.19231179  2.1879328 2.867450e-02
## PstatusT          0.38738382 0.27919788  1.3874884 1.652929e-01
## Medu             -0.17286265 0.12046701 -1.4349377 1.513049e-01
## Fedu              0.26488024 0.10803892  2.4517113 1.421787e-02
## Mjobhealth       -0.22555696 0.41573091 -0.5425552 5.874361e-01
## Mjobother        -0.33982850 0.24892001 -1.3652117 1.721865e-01
## Mjobservices     -0.05333781 0.30094766 -0.1772329 8.593255e-01
## Mjobteacher       0.22555678 0.40899214  0.5514942 5.812950e-01
## Fjobhealth       -0.29491359 0.64906758 -0.4543650 6.495661e-01
## Fjobother         0.12092249 0.34664766  0.3488340 7.272139e-01
## Fjobservices      0.53979956 0.36667418  1.4721504 1.409803e-01
## Fjobteacher      -0.91759532 0.52686739 -1.7416058 8.157744e-02
## reasonhome        0.04812911 0.23652242  0.2034865 8.387548e-01
## reasonother       0.61100694 0.28124037  2.1725435 2.981469e-02
## reasonreputation  0.18124792 0.23046317  0.7864507 4.316035e-01
## guardianmother   -0.07419416 0.21720151 -0.3415914 7.326584e-01
## guardianother    -0.20663396 0.39085402 -0.5286730 5.970323e-01
## traveltime       -0.05070551 0.12775480 -0.3968971 6.914434e-01
## studytime        -0.25214807 0.11639473 -2.1663186 3.028685e-02
## failures          0.06080643 0.13706379  0.4436359 6.573058e-01
## schoolsupyes     -0.43595131 0.30620772 -1.4237110 1.545301e-01
## famsupyes        -0.14926729 0.18037302 -0.8275478 4.079266e-01
## activitiesyes    -0.33537523 0.17970180 -1.8662875 6.200115e-02
## nurseryyes       -0.51312780 0.21319615 -2.4068343 1.609147e-02
## higheryes         0.11783636 0.30319597  0.3886475 6.975369e-01
## internetyes       0.08878143 0.21853200  0.4062629 6.845495e-01
## romanticyes      -0.23092410 0.18375074 -1.2567247 2.088533e-01
## famrel           -0.42691704 0.09289529 -4.5956802 4.313399e-06
## freetime         -0.17777256 0.09067133 -1.9606260 4.992267e-02
## goout             0.90370144 0.09060817  9.9737301 1.986260e-23
## health            0.17103111 0.06303226  2.7133903 6.659861e-03
## absences          0.04237085 0.01335389  3.1729210 1.509136e-03
## G1               -0.02734261 0.05384681 -0.5077851 6.116041e-01
## G2               -0.04619585 0.06965965 -0.6631651 5.072248e-01
## G3                0.04760911 0.05545324  0.8585452 3.905915e-01
## 1|2               0.74286931 1.65398689  0.4491386 6.533317e-01
## 2|3               1.98988169 1.65566252  1.2018643 2.294161e-01
## 3|4               3.29459804 1.65975812  1.9849869 4.714593e-02
## 4|5               4.91792255 1.67046461  2.9440447 3.239532e-03
```

Po obliczeniu p-wartości dla naszych współczynników, usuwamy z tabelki te które przekraczają naszą ustaloną wartość $\alpha = 5\%$ i wyświetlamy pozostałe współczynniki.

```
##                  Value Std. Error   t value      p value
## sexM         1.24883827 0.19650510  6.355246 2.080932e-10
## famsizeLE3   0.42076526 0.19231179  2.187933 2.867450e-02
## Fedu         0.26488024 0.10803892  2.451711 1.421787e-02
## reasonother  0.61100694 0.28124037  2.172543 2.981469e-02
## studytime   -0.25214807 0.11639473 -2.166319 3.028685e-02
## nurseryyes  -0.51312780 0.21319615 -2.406834 1.609147e-02
## famrel      -0.42691704 0.09289529 -4.595680 4.313399e-06
## freetime    -0.17777256 0.09067133 -1.960626 4.992267e-02
## goout        0.90370144 0.09060817  9.973730 1.986260e-23
## health       0.17103111 0.06303226  2.713390 6.659861e-03
## absences     0.04237085 0.01335389  3.172921 1.509136e-03
```

```
## 3|4          3.29459804 1.65975812  1.984987 4.714593e-02
## 4|5          4.91792255 1.67046461  2.944045 3.239532e-03
```

Utworzymy teraz model składający się tylko z atrybutów które przeszły test.

Poniżej pokazane jest wywołanie funkcji tworzącej drugi model.

```
m5 <- polr(Walc ~ sex + Medu + Fjob + guardian + studytime + goout
           + absences, data=train, Hess=TRUE, method='logistic')
```

Tak jak i poprzednio zaczniemy od testu dopasowania naszego modelu do danych.

```
## [1] "P-wartość drugiego modelu: 0.00"
```

Jak widać nawet nasz drugi model nie przeszedł testu dopasowania chi-kwadrat.

Następnie obliczymy dokładność naszego drugiego modelu i porównamy ją z dokładnością naszego pierwszego modelu którą obliczyliśmy wcześniej.

```
## [1] "Dokładność pierwszego modelu na zbiorze treningowym wynosi: 45.94%"
```

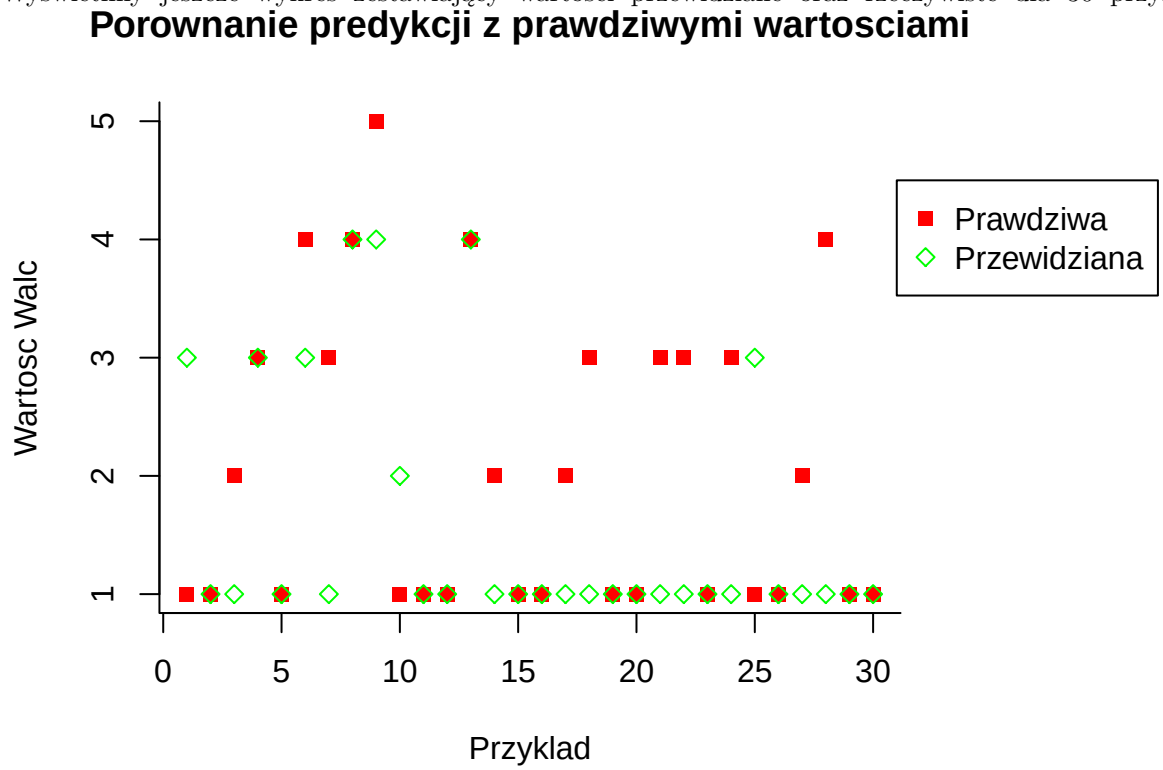
```
## [1] "Dokładność pierwszego modelu na zbiorze testowym wynosi: 37.59%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze treningowym wynosi: 43.67%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze testowym wynosi: 43.61%"
```

Okazuje się że po zmniejszeniu ilości zmiennych niezależnych w naszym modelu, nasz model różni się nieznacznie dokładnością na obu zbiorach. Jak widać także i dla regresji tego modelu podejście z wybraniem tylko i wyłącznie atrybutów o wystarczająco niskiej p-wartości nie było wystarczające.

Wyświetlmy jeszcze wykres zestawiający wartości przewidziane oraz rzeczywiste dla 30 przykładów.



Także i tutaj wyświetlimy tablicę pomyłek, aby przekonać się czy nie zmieniła się tendencja do przewidywania niższego niż w rzeczywistości spożycia alkoholu.

```
##      predicted
```

```
## real  1  2  3  4  5
##    1 44  1  6  2  0
##    2 24  0  4  2  0
##    3 14  0  6  3  1
##    4  3  1  6  8  0
##    5  1  0  0  7  0
```

Z tablicy pomyłek wynika że po raz kolejny otrzymaliśmy model z tendencją do zaniżania poziomu spożycia alkoholu.

StepAIC

Spróbujemy także i teraz skorzystać z funkcji stepAIC w nadziei że otrzymamy nieco lepszy model.

Na podstawie analizy funkcją stepAIC otrzymujemy formułę o najmniejszej wartości AIC. Utworzymy nowy model używając jej.

```
m6 <- polr(formula = Walc ~ sex + Medu + Fedu + Fjob + reason + guardian +
    studytime + schoolsup + activities + nursery + famrel + goout +
    health + absences + G1, data = train, Hess = TRUE, method = "logistic")
```

Wyświetlimy statystyki.

```
## Call:
## polr(formula = Walc ~ sex + Medu + Fedu + Fjob + reason + guardian +
##     studytime + schoolsup + activities + nursery + famrel + goout +
##     health + absences + G1, data = train, Hess = TRUE, method = "logistic")
##
## Coefficients:
##                      Value Std. Error   t value
## sexM              1.256738    0.18522  6.785099
## Medu             -0.160593    0.09985 -1.608375
## Fedu              0.266275    0.10356  2.571224
## Fjobhealth       -0.263666    0.63485 -0.415321
## Fjobother         0.108135    0.33950  0.318510
## Fjobservices      0.594034    0.35834  1.657738
## Fjobteacher      -0.882810    0.51743 -1.706156
## reasonhome        0.038058    0.22755  0.167251
## reasonother       0.601528    0.27639  2.176341
## reasonreputation  0.208563    0.22211  0.939017
## guardianmother   -0.001632    0.20634 -0.007911
## guardianother    -0.173080    0.34466 -0.502178
## studytime        -0.289244    0.11242 -2.572824
## schoolsupyes     -0.423060    0.29727 -1.423159
## activitiesyes    -0.342939    0.17275 -1.985142
## nurseryyes       -0.426180    0.20782 -2.050724
## famrel           -0.411402    0.08901 -4.622228
## goout             0.822451    0.08259  9.958083
## health            0.164649    0.06094  2.701875
## absences          0.038332    0.01263  3.035822
## G1               -0.016445    0.02954 -0.556671
##
## Intercepts:
##     Value   Std. Error t value
## 1|2  0.7897  0.6742     1.1712
```

```
## 2|3  2.0033  0.6800     2.9462
## 3|4  3.2797  0.6915     4.7426
## 4|5  4.8616  0.7136     6.8131
##
## Residual Deviance: 1333.676
## AIC: 1383.676
```

Następnie zgodnie ze zwyczajem sprawdzimy dopasowanie modelu do danych.

```
## [1] "p-wartość: 0.00"
```

Oraz obliczymy dokładność na zbiorze treningowym i testowym i zestawimy z wartościami poprzednich modeli.

```
## [1] "Dokładność pierwszego modelu na zbiorze treningowym wynosi: 45.94%"
```

```
## [1] "Dokładność pierwszego modelu na zbiorze testowym wynosi: 37.59%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze treningowym wynosi: 43.67%"
```

```
## [1] "Dokładność drugiego modelu na zbiorze testowym wynosi: 43.61%"
```

```
## [1] "Dokładność trzeciego modelu na zbiorze treningowym wynosi: 46.31%"
```

```
## [1] "Dokładność trzeciego modelu na zbiorze testowym wynosi: 41.35%"
```

Jak widać dokładność naszego trzeciego modelu wybranego z użyciem funkcji stepAIC jest także i teraz zbliżona do wartości poprzednich modeli. Wynika z tego że także i w przypadku regresji atrybutu Walc nie jest łatwo stwierdzić, których atrybutów warto użyć.

Wnioski

W projekcie tym staraliśmy odpowiedzieć między innymi na takie pytania jak:

- Czy model regresji musi być koniecznie dobrze dopasowany do danych, aby posiadać wysoką dokładność?
- Jak porównać ze sobą różne modele?
- Jak wybrać atrybuty przy tworzeniu naszego modelu?
- Gdzie nasze modele popełniają błąd?

Nauczyliśmy się dzięki temu wielu rzeczy. Teoretycznych, związanych ze statystyką, jak np. interpetacja takich wartości jak p-value, AIC, Residual Deviance, ale także praktycznych, czyli jak ich używać w doborze modelu oraz atrybutów.

Jeżeli chodzi o same dane które analizowaliśmy to dowiedzieliśmy się ciekawych rzeczy, jak przykładowo to że w przypadku ankietowanej grupy mężczyźni mieli zdecydowanie większe szanse na picie alkoholu niż kobiety.

Dalsza możliwa analiza

Innymi rzeczami które warte byłyby zbadania jest przykładowo regresja tych atrybutów z użyciem innych modeli, np. one-vs-all i przekonanie się czy ich dokładność jest gorsze czy lepsza, a także to czy metody wyboru modeli analizowane przez nas w tym projekcie działają dla nich lepiej czy gorzej.

Rzeczą wartą analizy byłaby także próba zmodyfikowania naszych modeli w taki sposób, aby nie posiadały one znalezionej przez nas tendencji do zaniżania przewidywanego poziomu spożycia alkoholu. Byłoby to zwłaszcza korzystne dla regresji spożycia alkoholu w weekend.