

Analiza odchodzenia klientów

Bartosz Chądzyński 255680 & Michał Turek 246993

2023-05-07

Spis treści

1	Wstęp	2
2	Preprocessing	2
2.1	Analiza opisowa	2
2.2	Wykresy	4
2.3	Interpretacja Wyników	13
3	Klasyfikacja	13
3.1	Regresja Liniowa	13
3.2	Regresja Logistyczna	14
3.3	Algorytm Naiwnego Bayesa	18
3.4	Algorytm k sąsiadów	18
3.5	Drzewo decyzyjne	20
3.6	SVM	20
3.7	Random Forest	22
3.8	Boosting	23
3.9	Bagging	23
4	Podsumowanie	23

1 Wstęp

Nasz projekt będzie dotyczył analizy danych dotyczących odchodzenia klientów firmy telekomunikacyjnej. Naszym celem jest zrozumienie, jakie czynniki wpływają na decyzję klientów o pozostaniu lub odejściu od firmy oraz jak te czynniki wpływają na skuteczność działań związanych z retencją klientów. W ramach projektu przeprowadzamy analizę danych, w tym eksploracyjną analizę, w której badamy rozkłady zmiennych oraz korelacje między nimi. Wprowadzamy również preprocessing danych, w tym normalizację oraz kodowanie zmiennych kategoriowych. Następnie tworzymy modele predykcyjne, które pozwalają na przewidywanie odchodzenia klientów. Przetestujemy różne algorytmy klasyfikacji, dobierając ostatecznie najlepsze. W efekcie naszej analizy otrzymujemy narzędzie predykcyjne.

2 Preprocessing

2.1 Analiza opisowa

Zbiór danych Telco Customer Churn składa się z 7043 obserwacji (klientów) i 21 zmiennych.

- customerID - unikalny identyfikator klienta
- gender - płeć klienta
- SeniorCitizen - czy klient jest emerytem (1) czy nie (0)
- Partner - czy klient ma partnera (Tak/Nie)
- Dependents - czy klient ma na utrzymaniu innych członków rodziny (Tak/Nie)
- tenure - okres w miesiącach, przez który klient był klientem firmy
- PhoneService - czy klient korzysta z usług telefonicznych (Tak/Nie)
- MultipleLines - czy klient ma więcej niż jedną linię telefoniczną (Tak/Nie/Brak usługi)
- InternetService - typ łącza internetowego (DSL, Fiber optic, Brak usługi)
- OnlineSecurity - czy klient korzysta z usług zabezpieczeń internetowych (Tak/Nie/Brak usługi)
- OnlineBackup - czy klient korzysta z usług kopii zapasowych danych online (Tak/Nie/Brak usługi)
- DeviceProtection - czy klient korzysta z usług zabezpieczeń urządzeń (Tak/Nie/Brak usługi)
- TechSupport - czy klient korzysta z usług technicznej pomocy (Tak/Nie/Brak usługi)

- StreamingTV - czy klient korzysta z usług strumieniowego przesyłania telewizji (Tak/Nie/Brak usługi)
- StreamingMovies - czy klient korzysta z usług strumieniowego przesyłania filmów (Tak/Nie/Brak usługi)
- Contract - typ umowy (Month-to-month, One year, Two year)
- PaperlessBilling - czy klient otrzymuje faktury w formie papierowej (Tak/Nie)
- PaymentMethod - metoda płatności (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- MonthlyCharges - miesięczny rachunek klienta
- TotalCharges - łączny rachunek klienta
- Churn - czy klient zrezygnował z usług firmy (Tak (1)/Nie (0)).

Wszystkie zmienne są w formie tekstowej, lub binarnej, oprócz trzech zmiennych numerycznych: SeniorCitizen, tenure, MonthlyCharges oraz jednej zmiennej numerycznej typu float: TotalCharges. Na początku dokonamy analizy tych trzech zmiennych numerycznych, wykorzystując podstawowe statystyki.

tenure	MonthlyCharges	TotalCharges
Min. : 1.00	Min. : 18.25	Min. : 18.8
1st Qu.: 9.00	1st Qu.: 35.59	1st Qu.: 401.4
Median :29.00	Median : 70.35	Median :1397.5
Mean :32.42	Mean : 64.80	Mean :2283.3
3rd Qu.:55.00	3rd Qu.: 89.86	3rd Qu.:3794.7
Max. :72.00	Max. :118.75	Max. :8684.8

Badając mediany i średnie poszczególnych zmiennych z powyższej tabeli możemy wyciągnąć kilka wniosków. Na przykład średnia wartość miesięcznej opłaty to 64.76 dolara, a mediana to 70.35 dolara. Można z tego wnioskować, że rozkład tej zmiennej jest skośny w lewo, co sugeruje, że większość klientów płaci więcej niż średnia wartość. Średni czas trwania umowy wynosi 32.37 miesiąca, a mediana to 29 miesięcy. Można zauważyć, że większość klientów trzyma się firmy przez mniej niż 3 lata. Średnia wartość *MonthlyCharges* dla klientów, którzy odeszli (*Churn*=1), wynosi 74.44 dolarów, podczas gdy dla klientów, którzy pozostali (*Churn*=0), wynosi 61.27 dolarów. Można z tego wnioskować, że klienci, którzy płacą więcej za usługi, są bardziej skłonni do zrezygnowania z nich. Są to oczywiście tylko przykładowe wnioski, które możemy wyciągnąć z danych na podstawie prostych statystyk. W dalszych częściach pracy będziemy analizowali dane z pomocą modeli o różnej złożoności.

Spójrzmy teraz na pozostałe zmienne. Na podstawie rozkładu zmiennych w poszczególnych kategoriach możemy wyciągnąć kilka wniosków (udział ten można zobaczyć na histogramach w kolejnym podrozdziale). Między innymi:

- Większość klientów to osoby indywidualne (0.715).
- Większość klientów korzysta z usługi telefonii cyfrowej (0.903).
- Większość klientów korzysta z faktury elektronicznej (0.704).
- Większość klientów nie korzysta z usługi ochrony urządzeń (0.901).
- Około połowa klientów korzysta z usługi internetu szerokopasmowego (0.468).

Z powyższych danych można wywnioskować, że firma powinna skupić się na promowaniu usługi internetu szerokopasmowego oraz usługi ochrony urządzeń, aby zwiększyć liczbę klientów korzystających z tych usług. Dodatkowo, firma powinna zastanowić się nad przyczynami, dla których tak mało klientów korzysta z faktury elektronicznej i ewentualnie wdrożyć działania promocyjne, zachęcające do korzystania z tej formy rozliczenia.

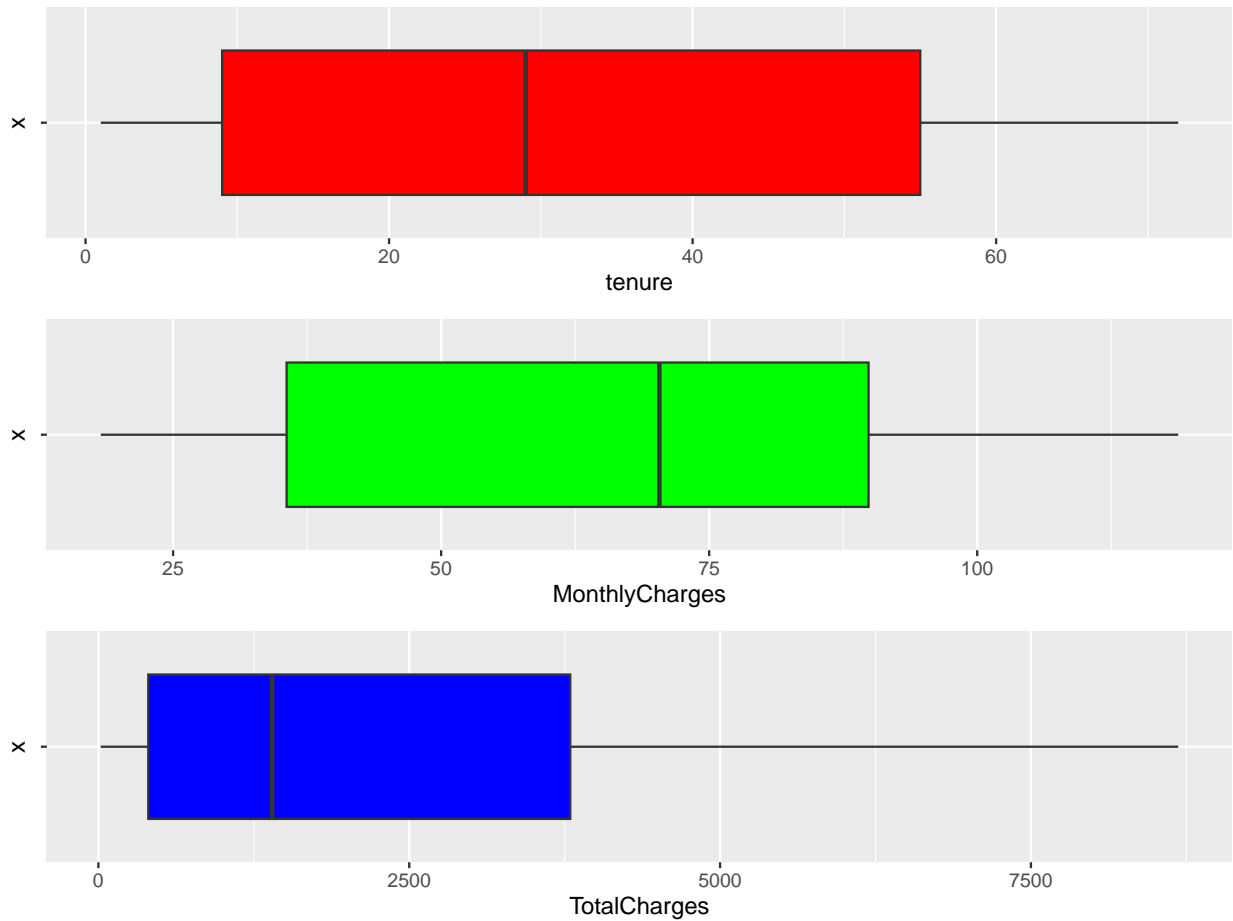
2.2 Wykresy

Zacznijmy od analizy wykresów. Na początek zmienne ciągłe. Na wykresach 1 i 3 oraz w tabeli poniżej widzimy, że zmienne te są w znacząco różnych skalach, więc prawdopodobnie potrzebna będzie normalizacja. Zmienna *tenure*, a więc czas jaki dana osoba była/jest klientem, waha się od 1 do 72 miesięcy. Przy czym jej rozkład jest dwumodalny. Teoretycznie powinno się wydawać, że rozkład tej zmiennej powinien mieć charakter podobny do rozkładów z rodziny Gamma (np. rozkładu wykładniczego). W końcu każdy klient po pewnym czasie odchodzi, a więc w miarę upływu czasu klientów ubywa. Być może jednak jakieś procesy rynkowe spowodowały, że mamy liczniejszą grupę klientów ze stażem ok. 70 miesięcy (np. 70 miesięcy temu dana firma proponowała bardzo korzystne umowy). Kolejną zmienną jest *MonthlyCharges*. Jej estymowany rozkład jest bardzo nieregularny. Ma kilka maksimów lokalnych. Występują one w okolicach okrągłych liczb, takich jak 50 czy 80. Zapewne są związane z jakimiś limitami, które posiadają klienci, gdyż najczęściej nie przekraczają tych kolejnych dziesiątek, albo mówiąc inaczej cyfra 9 pojawia się tu nadzwyczaj często jako cyfra jedności. Na koniec zostaje jeszcze *TotalCharges*. Zmienna ta ma spodziewany rozkład, tzn. przypomina on swoim kształtem rozkład Gamma. Wartości tej zmiennej są dużo większe od pierwszych dwóch, dlatego przeprowadzimy normalizację danych przed ich użyciem.

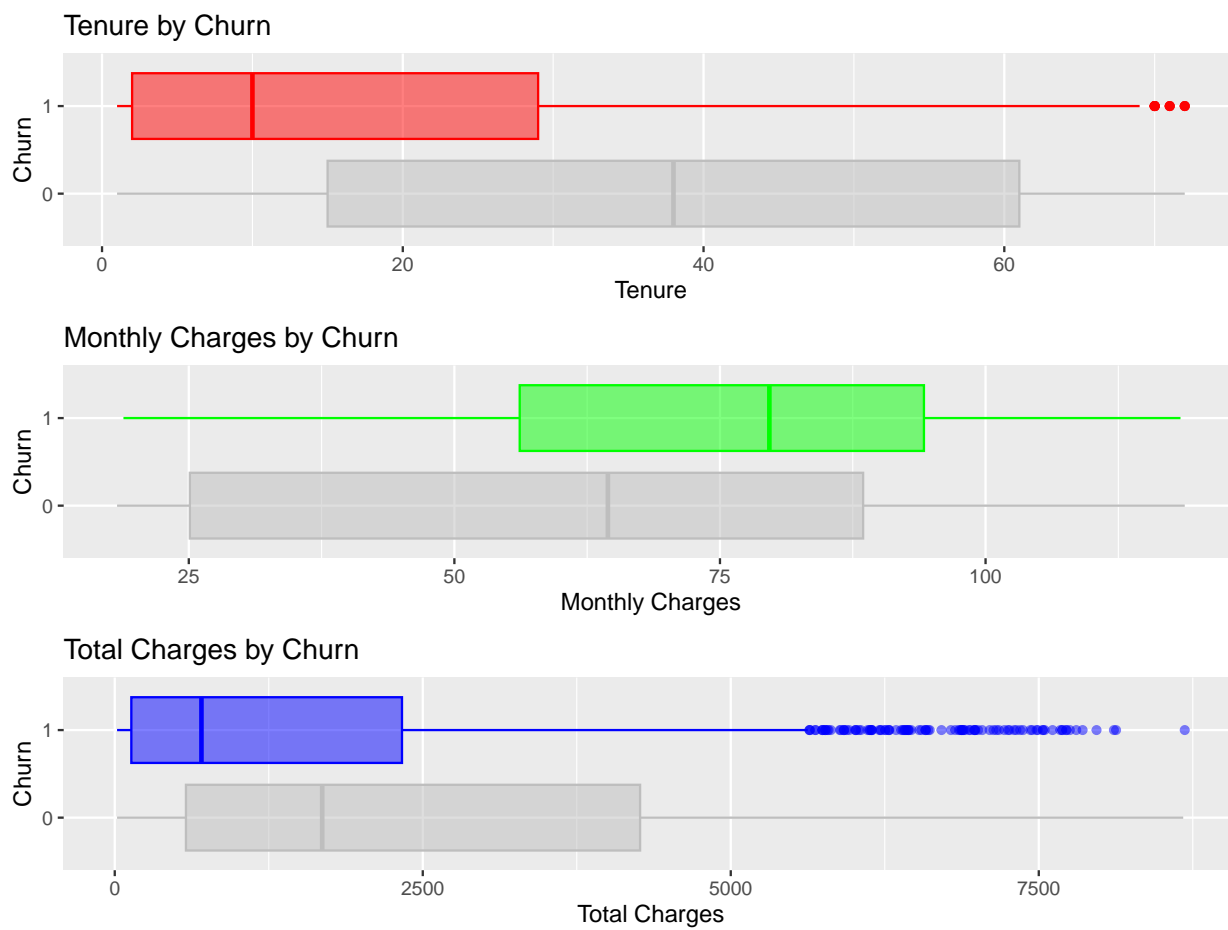
Na wykresie 2 i 4 widać, że każda ze zmiennych ma istotnie różny rozkład, gdy pogrupujemy ją ze względu na Churn. Najbardziej wyróżnia się *tenure*, gdzie widać że odchodzili głównie nowi klienci. Podobnie odchodzili głównie klienci, których łączne opłaty były stosunkowo niskie, ale wynika to z korelacji zmiennej *TotalCharges* ze zmienną *tenure* (opiszemy to później). Natomiast jeśli chodzi o *MonthlyCharges*, to klienci, którzy odeszli, przeważają wśród tych co płacili większe miesięczne rachunki, co nie dziwi.

Na wykresach 5, 6, 7, 8 widzimy, że w niektórych przypadkach są duże różnice w ilości obserwacji z każdej kategorii, jeśli chodzi o daną zmienną. W szczególności takimi zmiennymi są *PhoneService*, czy *MultipleLines*. Natomiast w znacznej większości proporcje klientów, którzy zostali i odeszli są podobne w każdej kategorii. Wyróżniają się tu osoby, które miały

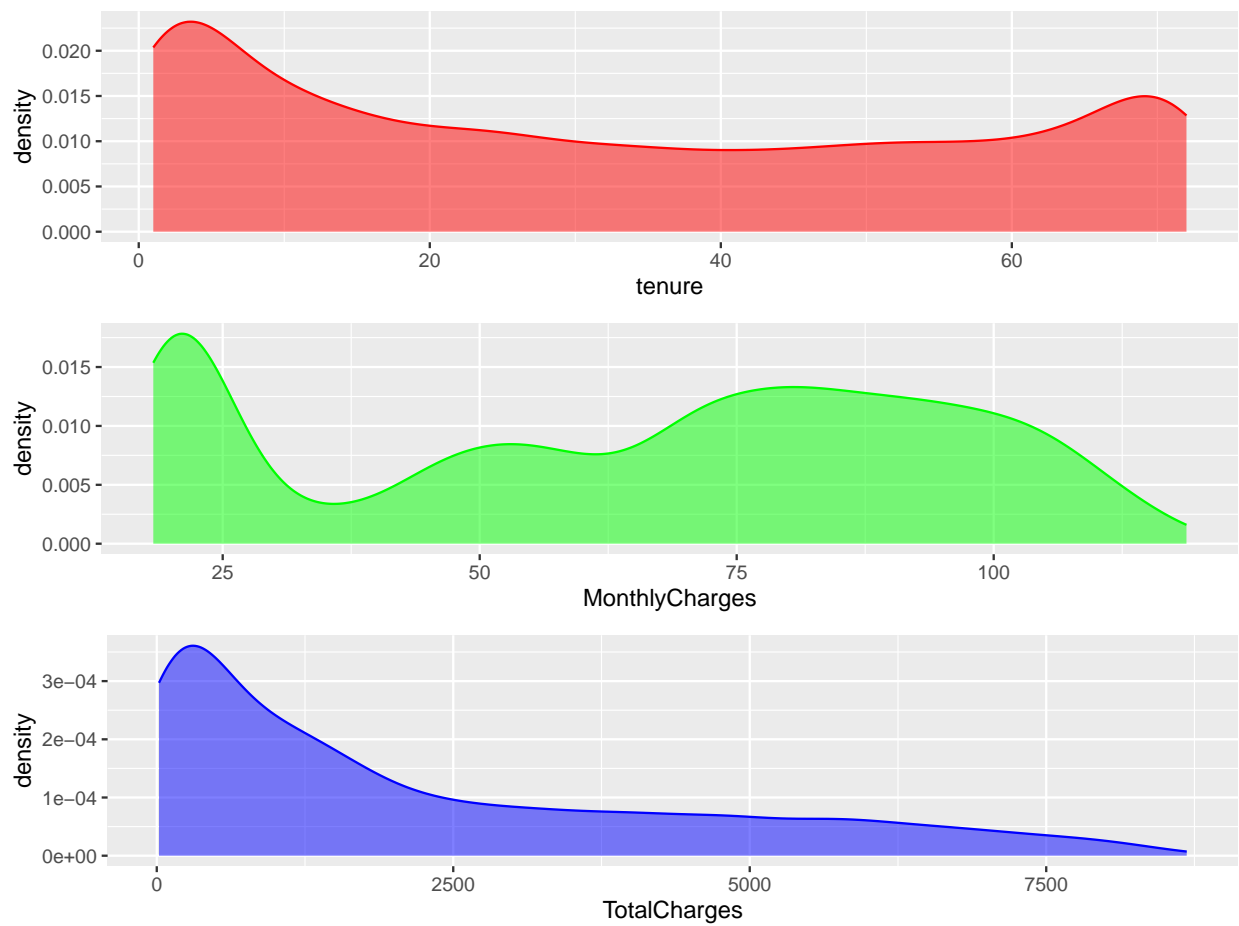
miesięczne kontrakty. To głównie one rezygnowały z usług operatora. W przypadku umów długoterminowych takie sytuacje zdarzały się bardzo rzadko. Podobnie wyróżnia się sposób płatności. Prawie połowa osób płacących za pomocą *electronic check* odeszła. Oczywiście w innych metodach płatności te liczby nie były aż tak duże. Można też zauważyć, że wśród straconych klientów jest bardzo mało osób, które nie korzystały z usług internetowych.



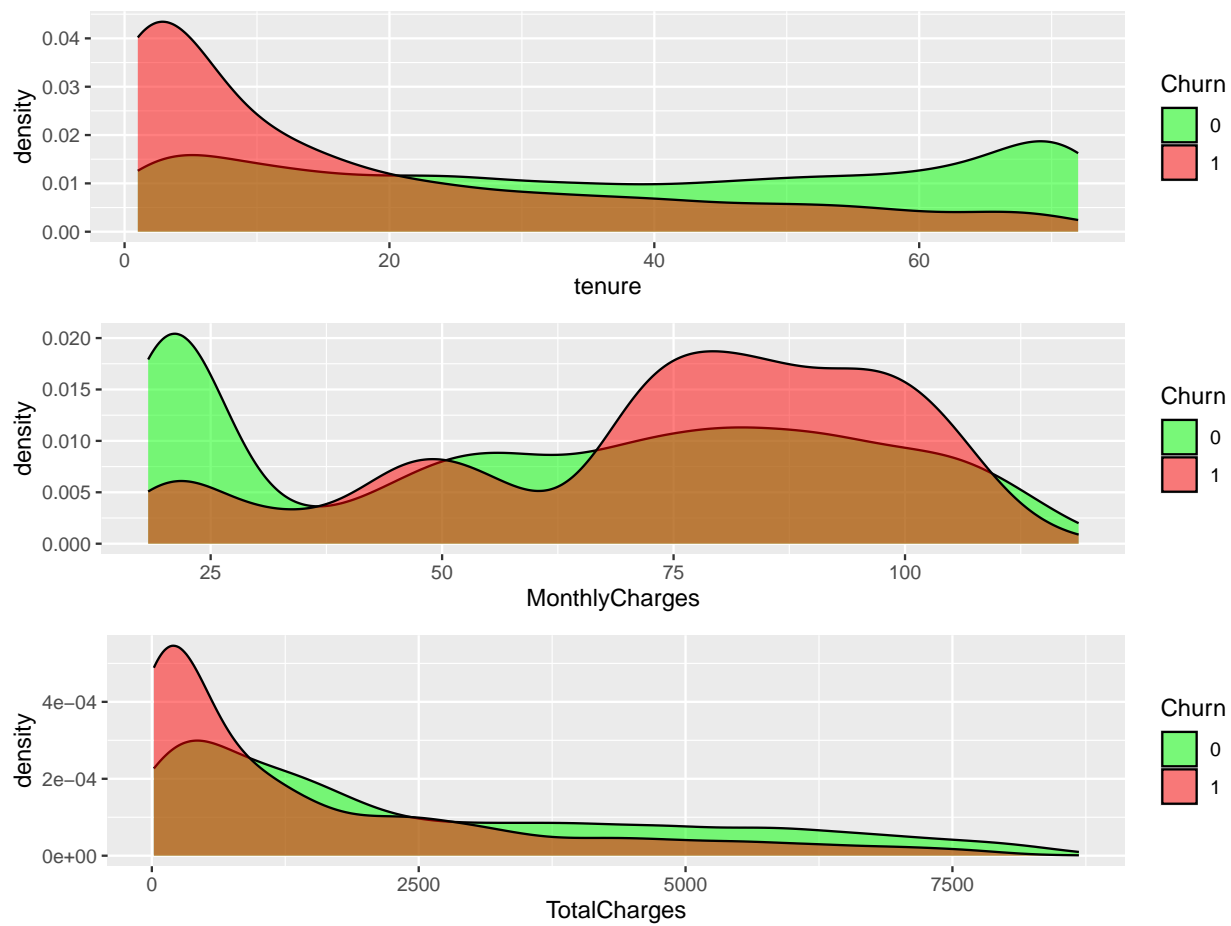
Rysunek 1: Boxploty zmiennych ciągłych



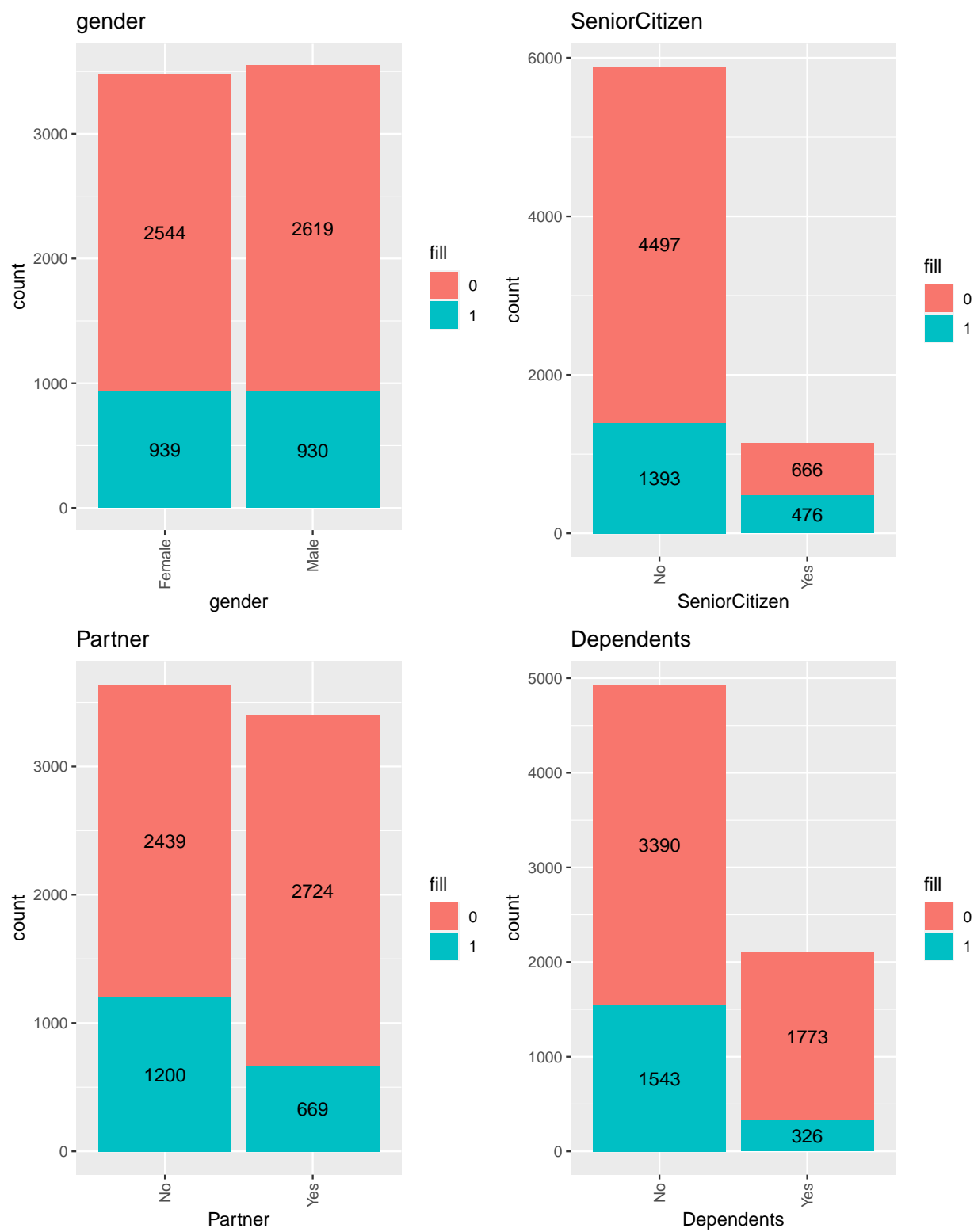
Rysunek 2: Boxploty zmiennych ciągłych z podziałem ze względu na Churn



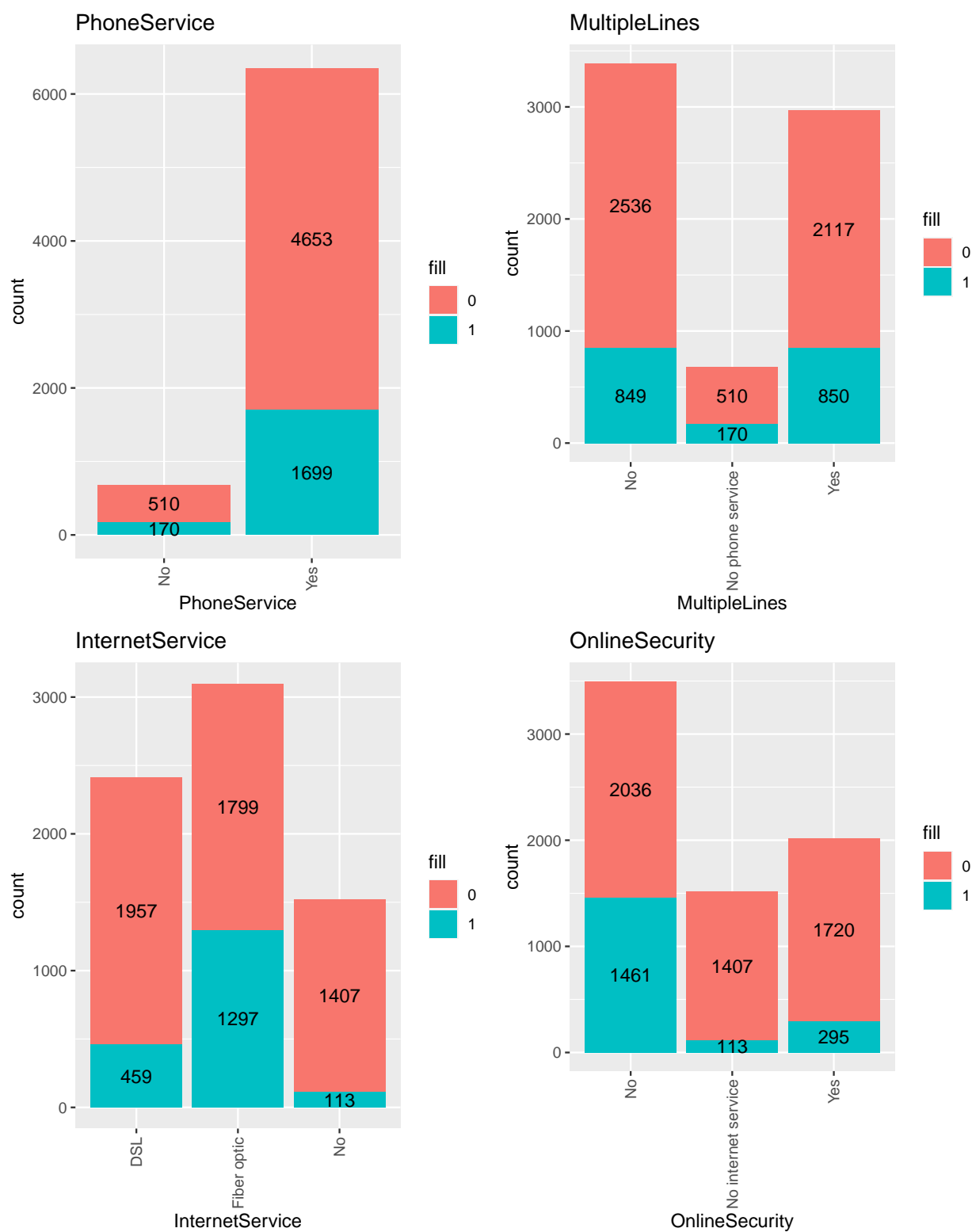
Rysunek 3: Estymator jądrowy gęstości



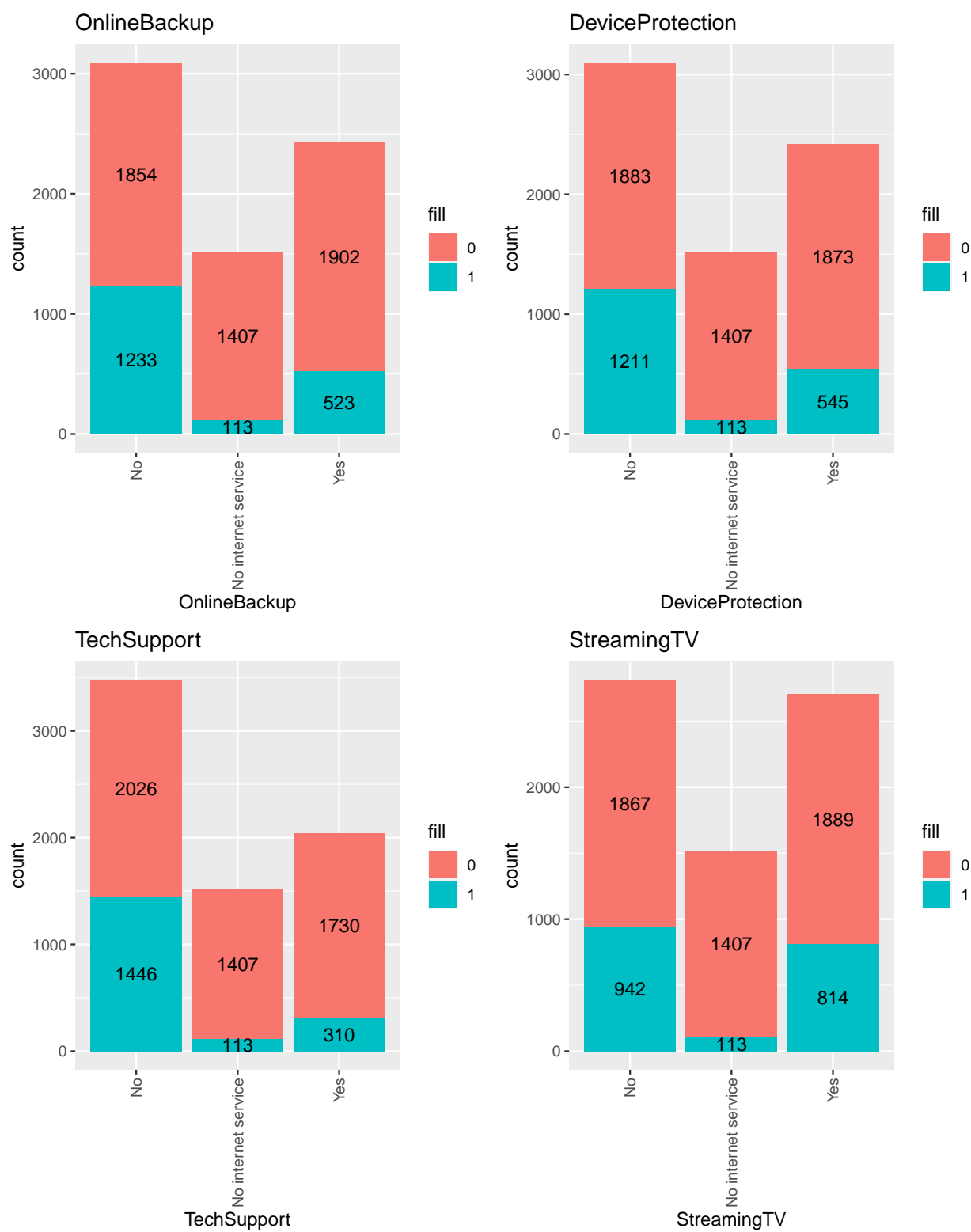
Rysunek 4: Estymator jądrowy gęstości z podziałem ze względu na Churn



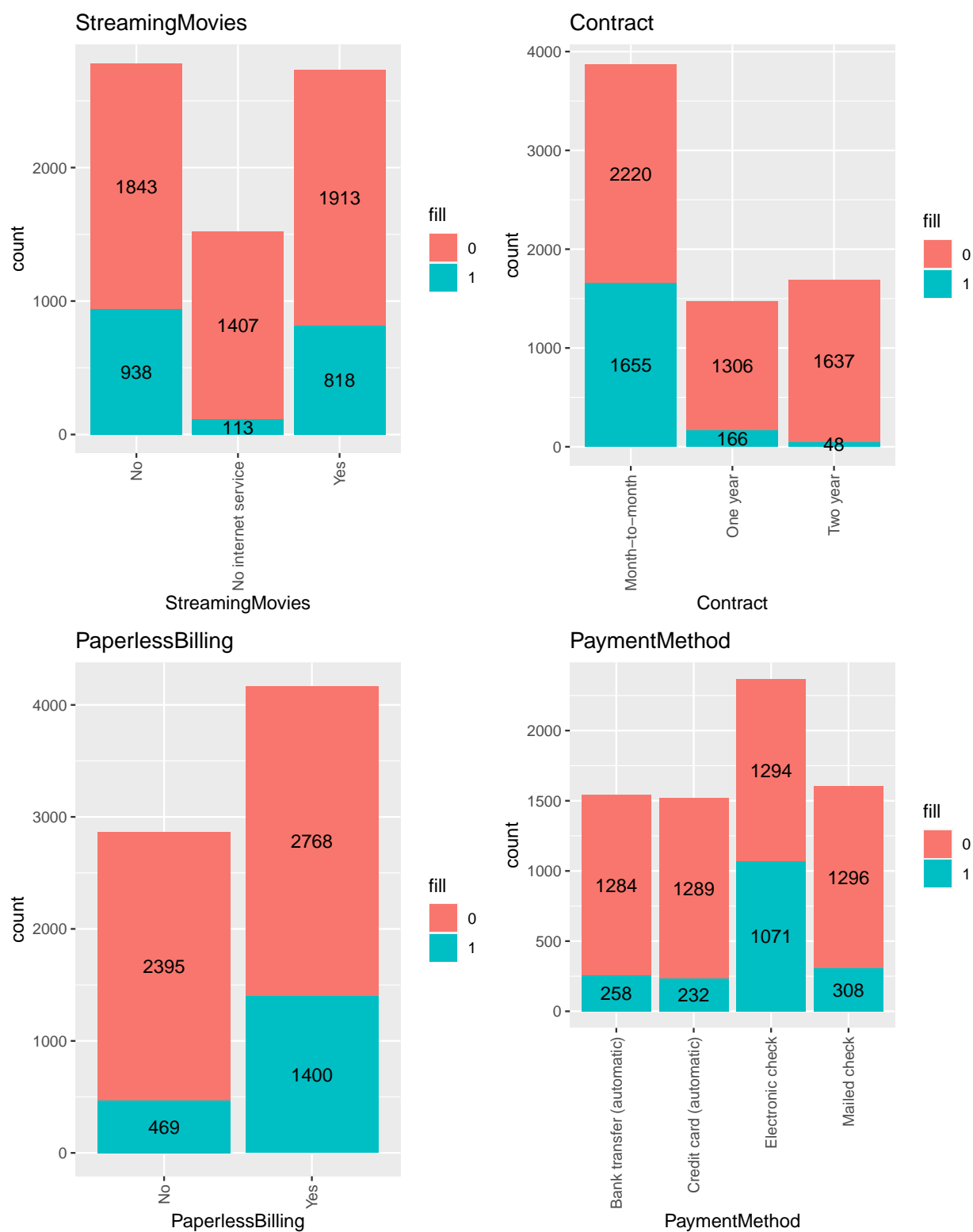
Rysunek 5: Wykres ilości obserwacji z podziałem na kategorie zmiennych



Rysunek 6: Wykres ilości obserwacji z podziałem na kategorie zmiennych



Rysunek 7: Wykres ilości obserwacji z podziałem na kategorie zmiennych



Rysunek 8: Wykres ilości obserwacji z podziałem na kategorie zmiennych

2.3 Interpretacja Wyników

W naszych danych jest zaledwie 11 obserwacji z brakującymi danymi (na 7033 łącznie). Zatem zasadne jest pominięcie ich w trakcie analizy danych. Nie stosujemy żadnej imputacji. Ilość danych wydaje się odpowiednia (nie za mała i nie za duża). W analizie dokonujemy losowego podziału na zbiór treningowy i testowy.

W tabeli poniżej mamy macierz korelacji zmiennych ciągłych. Jak widać istnieje mocna korelacja pomiędzy tym jak długo klient korzysta/korzystał z usług, a kwotą jaką zapłacił za usługi. Nie powinno to dziwić. Na razie jednak nie decydujemy się na wyrzucenie którejs z zmiennych, ponieważ zarówno czas jak i koszt może być istotny w kontekście odchodzenia klientów. Te dwie rzeczy nie muszą być ze sobą powiązane w pełni. Może być tak, że odchodzą głównie nowi klienci, niezależnie od tego ile płacą. Albo może być tak, że odchodzą klienci, którzy zapłacili rachunki powyżej pewnej sumy, niekoniecznie będący długo/krótko stażem.

	tenure	MonthlyCharges	TotalCharges
tenure	1.00	0.25	0.83
MonthlyCharges	0.25	1.00	0.65
TotalCharges	0.83	0.65	1.00

Potrzebne będzie wykonanie transformacji danych, w szczególności normalizacji. Natomiast jeśli chodzi o obserwacje odstające, to nie ma ich za dużo. Pojawiają się licznie w przypadku zmiennej *TotalCharges* pogrupowanej ze względu na *Churn*. Widać, że jest tendencja, aby odchodzący klienci należeć do jednej z dwóch grup. Są albo nowymi klientami, albo klientami z dużym stażem. Ta druga grupa jest na wykresie pudełkowym interpretowana jako obserwacje odstające. W rzeczywistości należy to interpretować tak, że rozkład tej zmiennej jest dwumodalny, nie będziemy stosować technik mających na celu ignorowanie lub zmniejszenie wpływu tych obserwacji, znacząco odbiegających od reszty.

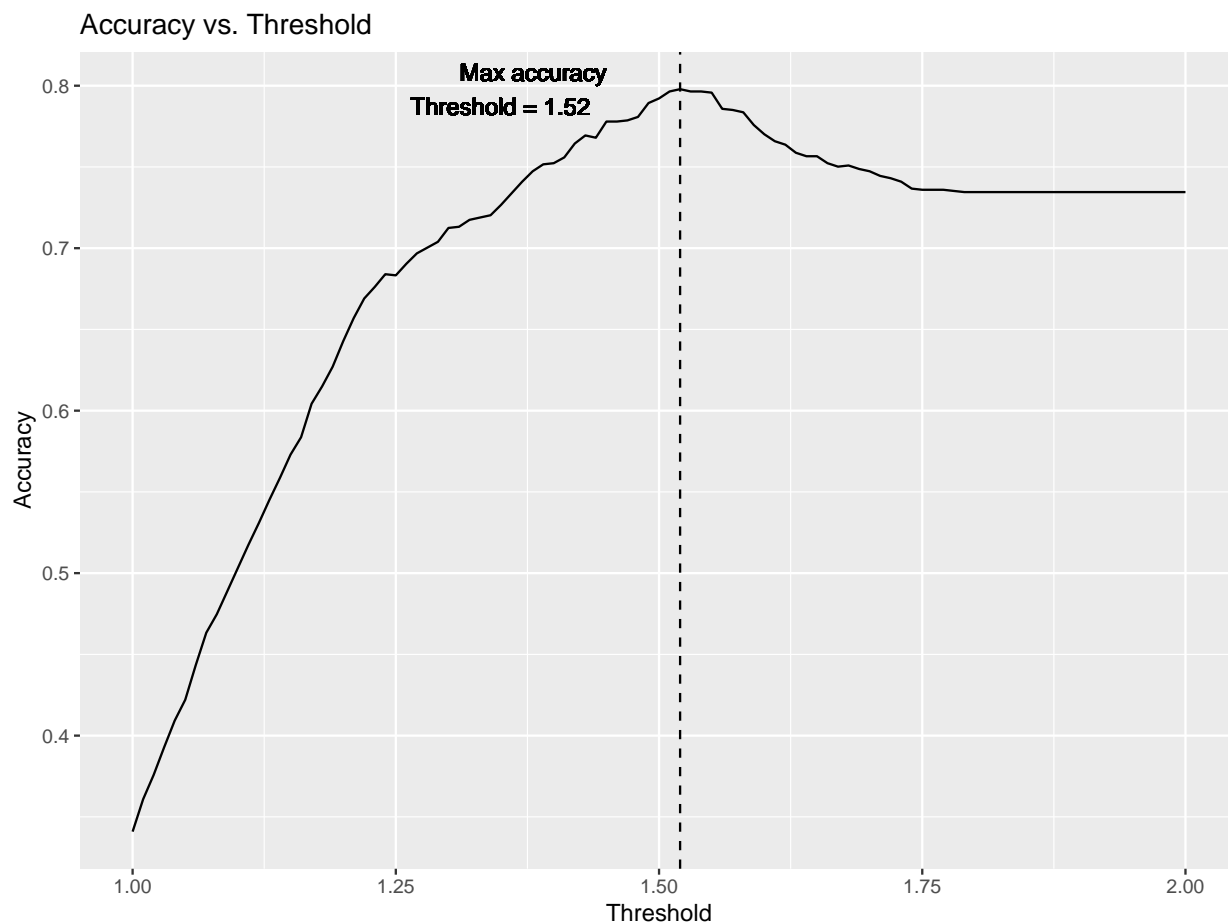
3 Klasyfikacja

3.1 Regresja Liniowa

Zacznijmy od metod, w których bierzemy pod uwagę jedynie zmienne ciągłe. Na początek regresja liniowa. Zastosowaliśmy model regresji liniowej, który bierze pod uwagę 3 zmienne ciągłe, jako zmienne objaśniające i *Churn*, jako zmienną objaśnianą. Otrzymane w ten sposób wartości dzielimy na dwie grupy stosując punkt odcięcia na ustalonym poziomie. Na wykresie 9 widzimy skuteczność predykcji dla punktów odcięcia pomiędzy 1 i 2. Wybieramy ten z największą skutecznością i sprawdzamy jak wygląda macierz pomyłek dla niego (1). Jak widać osiągamy w ten sposób całkiem niezłą skuteczność na poziomie 0.7978648, co jest o ok. 0.05 więcej niż, gdybyśmy estymowali każdą obserwację do liczniejszej klasy.

Zastanówmy się jeszcze jaki wpływ miały poszczególne zmienne w modelu. W tym celu przyjrzymy się współczynnikom w modelu (wykres 10). Nie ma tam stałej, ponieważ nie ma ona

wpływu na model (i tak później ustalamy punkt odcięcia). Widać natomiast, że największy wpływ na to że ktoś jest sklasyfikowany z $Churn=1$, ma zmienna $MonthlyCharges$. Im większe miesięczne opłaty, tym większe prawdopodobieństwo że osoba zrezygnuje z umowy. Odwrotnie jest w przypadku łącznych opłat i stażu klienta, które to zmienne zwiększają prawdopodobieństwo aby obserwacja była zakwalifikowana jako $Churn=0$.



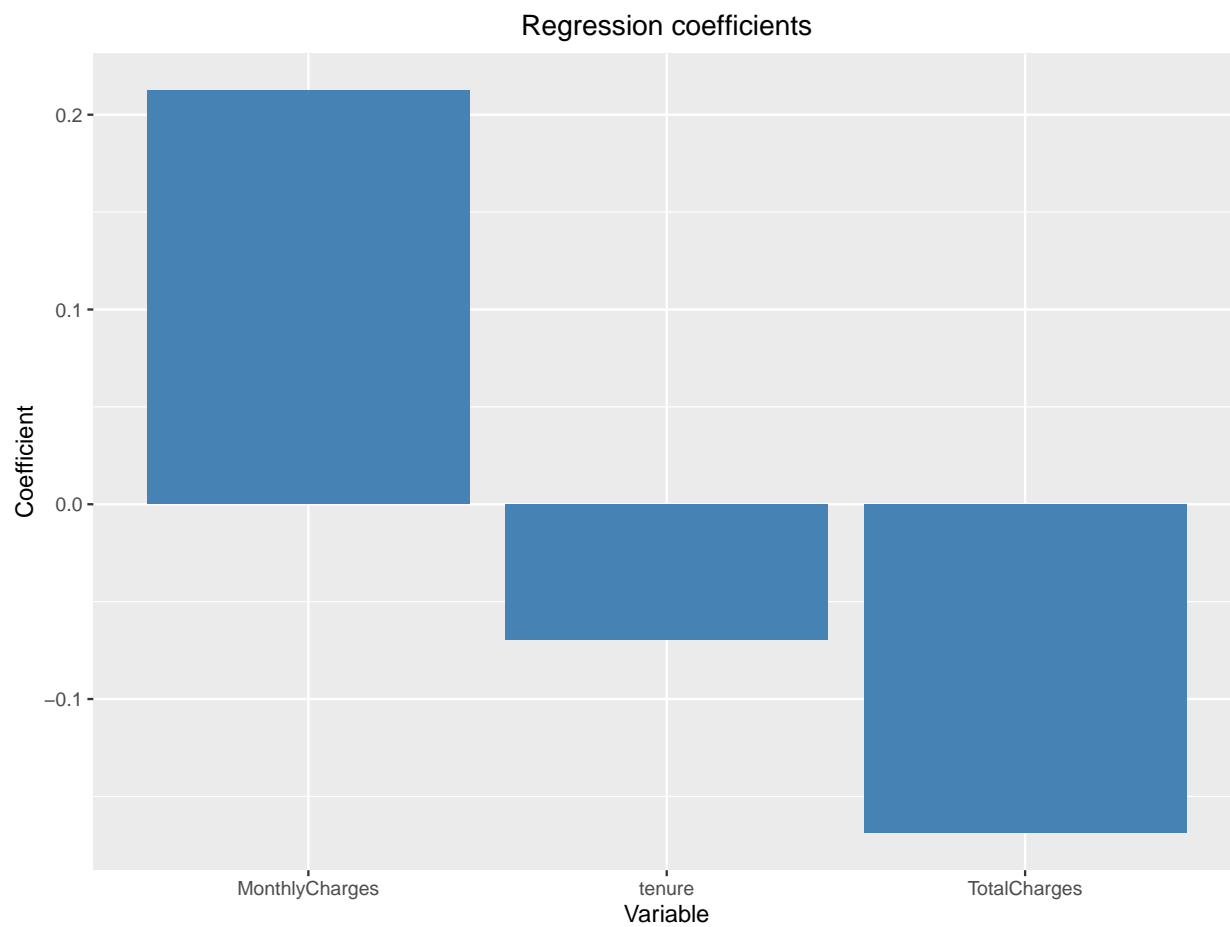
Rysunek 9: Skuteczność predykcji dla poszczególnych punktów odcięcia

	Rzeczywiste 0	Rzeczywiste 1
Estymowane 0	968	220
Estymowane 1	64	153

Tabela 1: Macierz pomyłek dla regresji liniowej, z punktem odcięcia = 1.52

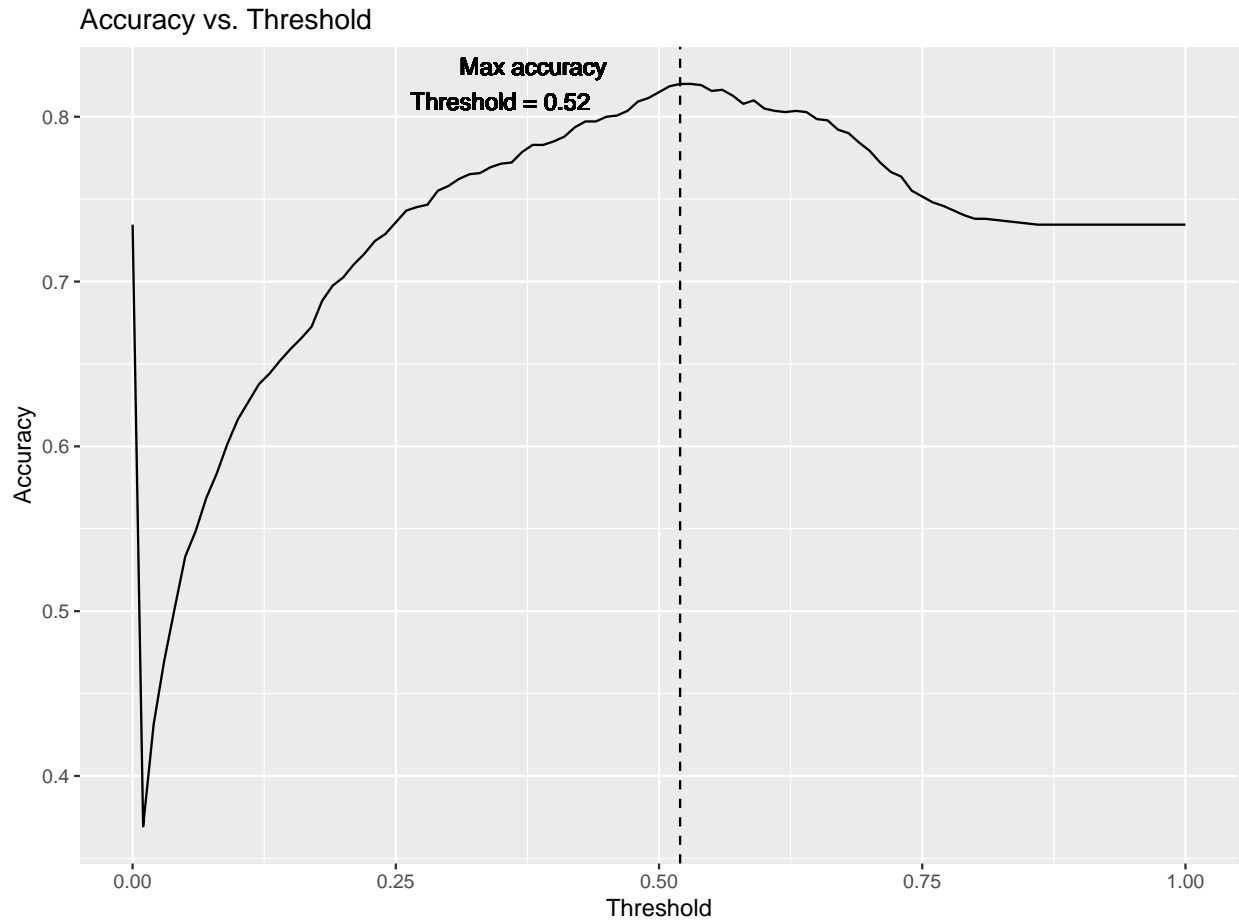
3.2 Regresja Logistyczna

Teraz model regresji logistycznej. Standardowo, wzięliśmy pod uwagę wszystkie zmienne i zbudowaliśmy model z domyślnymi parametrami. Punkt odcięcia wybraliśmy testując skuteczności przy różnych wartościach. Na wykresie 11 widzimy, że najskuteczniejszy był model



Rysunek 10: wartości współczynników w modelu regresji logistycznej

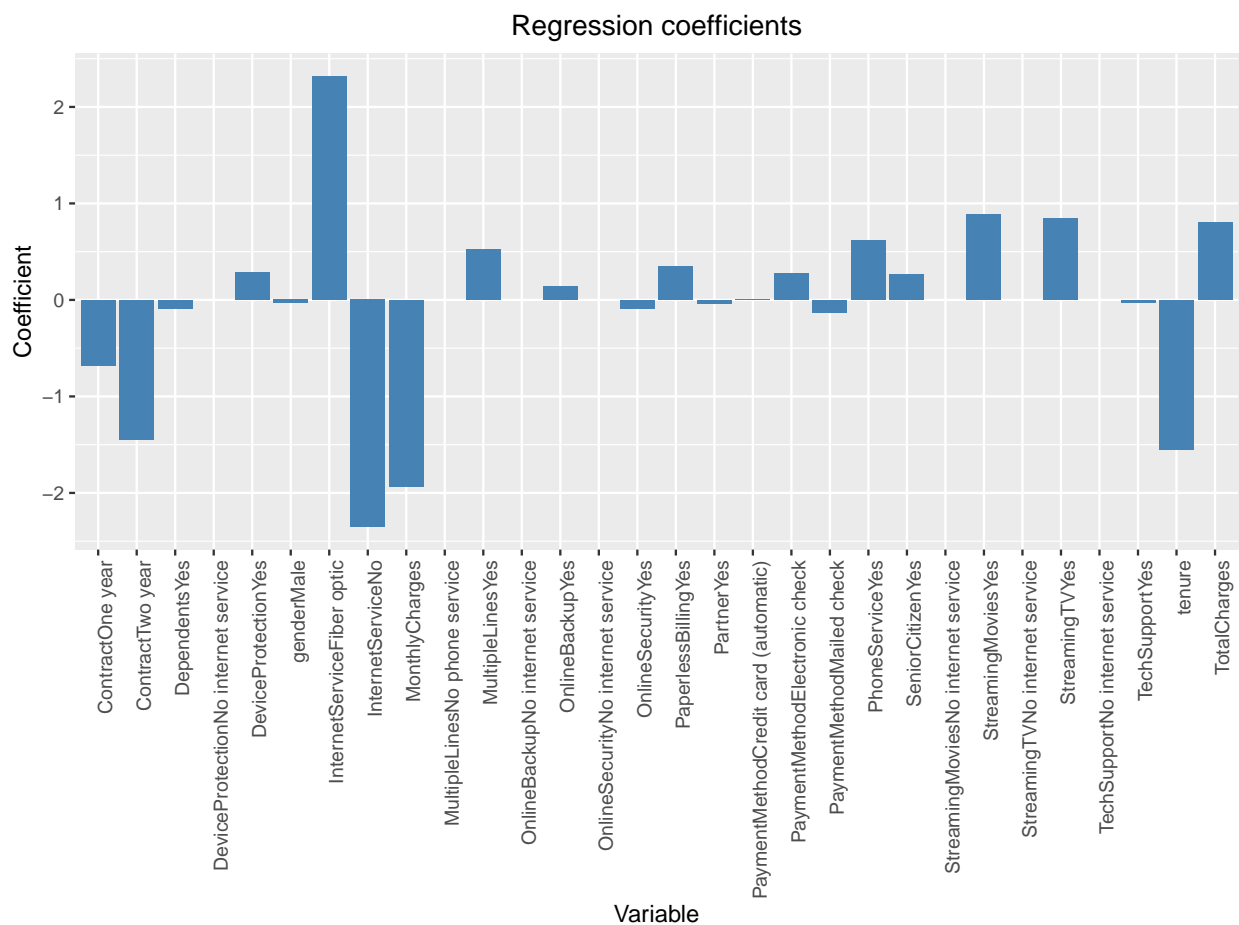
z punktem odcięcia równym 0.52. Z tabeli 1 widzimy, że skuteczność wynosi niemal dokładnie 0.82. Można się jeszcze przyjrzeć współczynnikom modelu. Na ich podstawie widzimy, że największy wpływ oprócz *MonthlyCharges* i *tenure* mają zmienne *InternetServiceFiber* i *InternetService*.



Rysunek 11: Skuteczność predykcji dla poszczególnych punktów odcięcia

	Rzeczywiste 0	Rzeczywiste 1
Estymowane 0	942	163
Estymowane 1	90	210

Tabela 2: Macierz pomyłek dla regresji logistycznej, z punktem odcięcia = 0.52



Rysunek 12: wartości współczynników w modelu regresji logistycznej

3.3 Algorytm Naiwnego Bayesa

Zastosowaliśmy również Naive Bayes Algorithm z domyślnymi parametrami (funkcja *Naive-Bayes* z pakietu *klaR*). Nie dał on jednak zbyt dobrych rezultatów. W tabeli 3 widzimy, że skuteczność tego modelu wyniosła zaledwie 0.76. Za to po raz pierwszy mamy sytuację kiedy obiekty, które w rzeczywistości mają *Churn*=1 są częściej błędnie klasyfikowane niż obiekty z *Churn*=0.

	Estymowane 0	Estymowane 1
Rzeczywiste 0	778	254
Rzeczywiste 1	90	283

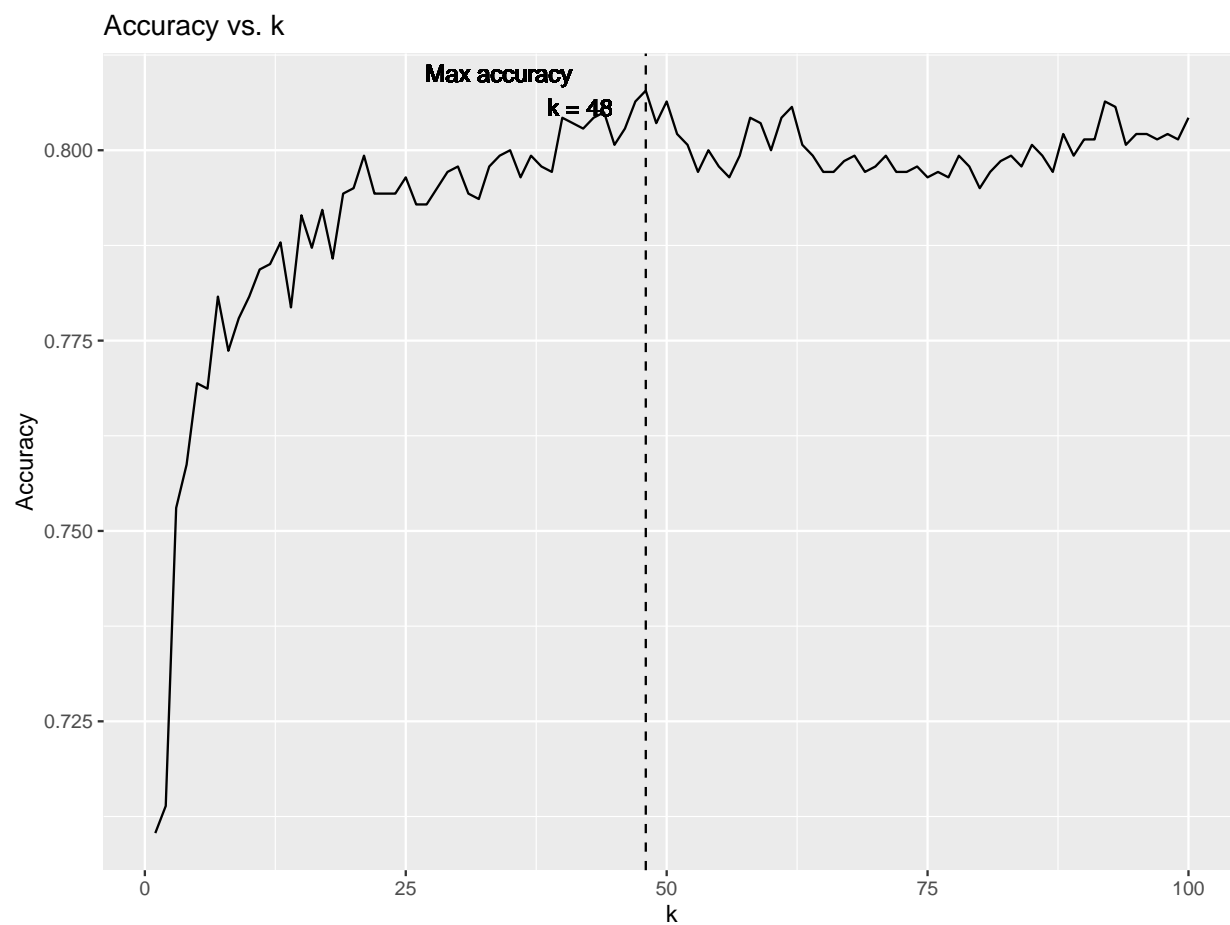
Tabela 3: Macierz pomyłek dla algorytmu Naiwnego Bayesa

3.4 Algorytm k sąsiadów

W tym przypadku użyliśmy funkcji *knn* z pakietu *class*. Przeprowadziliśmy symulację dla wszystkich wartości *k* od 1 do 100. Najlepszy model powstał dla *k* równego 48 (wykres 13). Jego skuteczność wyniosła 0.806.

	Estymowane 0	Estymowane 1
Rzeczywiste 0	917	115
Rzeczywiste 1	158	215

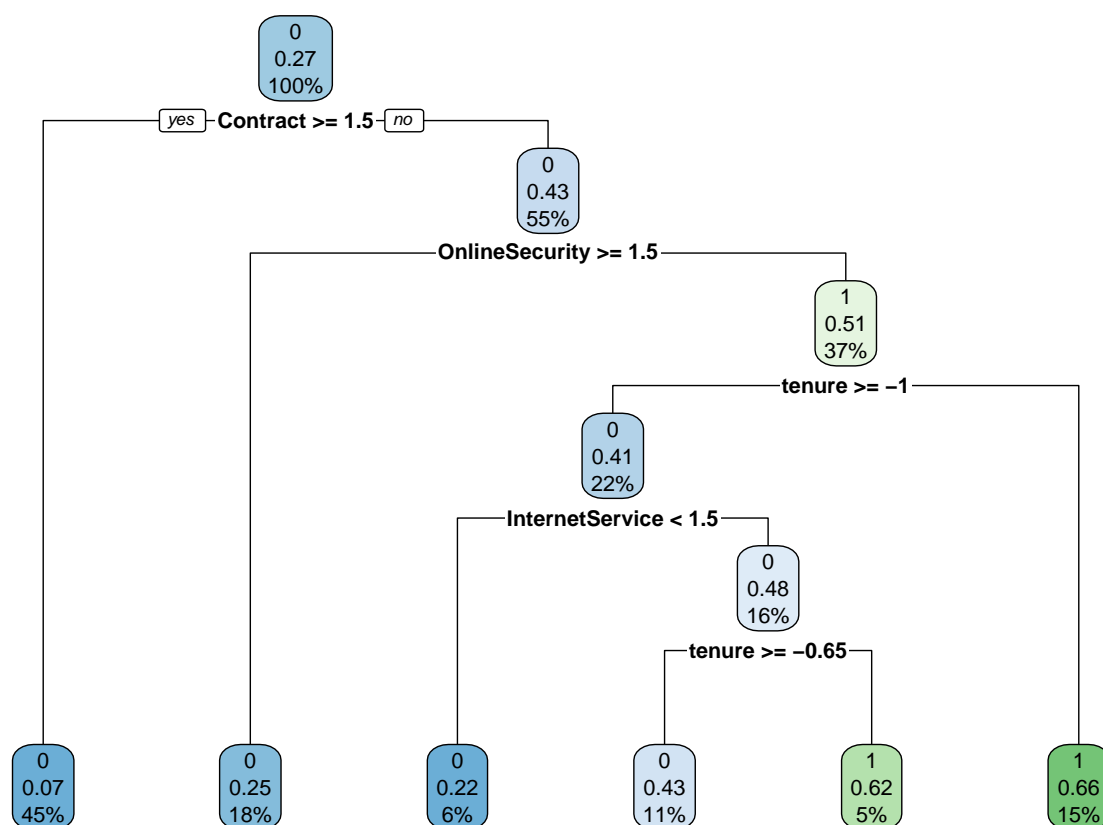
Tabela 4: Macierz pomyłek dla algorytmu kNN, *k* = 48



Rysunek 13: Skuteczność predykcji dla poszczególnych wartości k

3.5 Drzewo decyzyjne

Użyliśmy funkcji *rpart* z pakietu *rpart*. Na wykresie poniżej widzimy jakie zmienne warunkowały kolejne przejścia w drzewie. Natomiast macierz pomyłek 5 wskazuje, że skuteczność wyniosła 0.79. Oczywiście pojedyncze drzewo jest bardzo niestabilne, dlatego zastosujemy też metody ze wzmocnieniem.



	0	1
0	934	98
1	201	172

Tabela 5: Confusion matrix

3.6 SVM

Algorytm SVM zastosowaliśmy z 4 rodzajami jąder: liniowym, wielomianowym (stopień 3), radialnym i sigmoidalnym. W tabelach 6 - 9 widzimy wyniki. Jak widać największą skuteczność otrzymujemy stosując jądro radialne i jest ona na poziomie 0.806. Równie dobrze algorytm działa w przypadku jądra liniowego, gdzie skuteczność wynosi 0.804. W pozostałych przypadkach wyniki są dużo gorsze.

	0	1
0	923	167
1	109	206

Tabela 6: Confusion matrix for SVM (linear kernel)

	0	1
0	931	196
1	101	177

Tabela 7: Confusion matrix for SVM (polynomial kernel)

	0	1
0	945	188
1	87	185

Tabela 8: Confusion matrix for SVM (radial kernel)

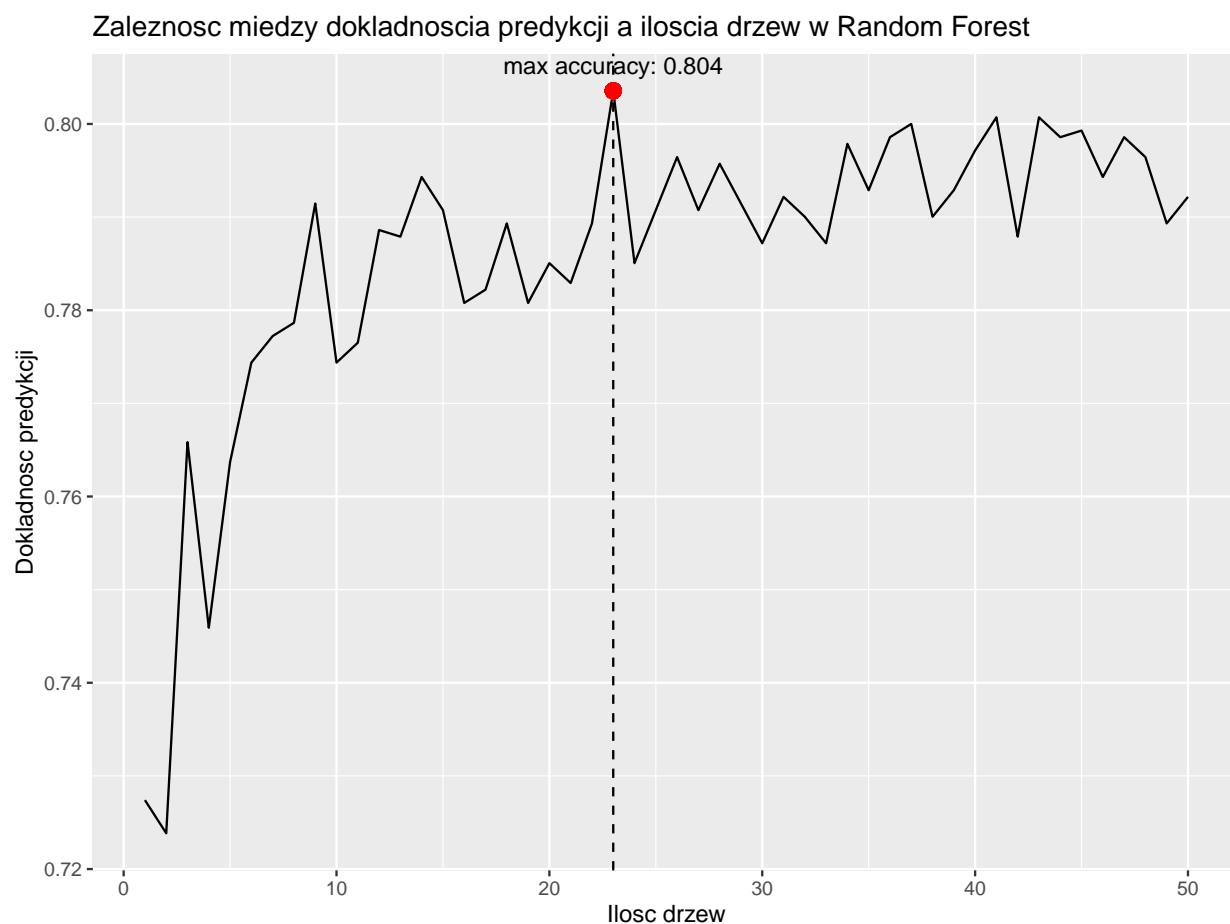
	0	1
0	862	192
1	170	181

Tabela 9: Confusion matrix for SVM (sigmoid kernel)

Następnie zastosujemy 3 algorytmy wzmacniające. Będą to lasy losowe (Random forest), bagging i boosting. Dla tego pierwszego sprawdzimy również, jak na dokładność działania algorytmu wpłynie ilość drzew.

3.7 Random Forest

Zacznijmy od algorytmu Random Forest. Skorzystamy z funkcji `randomForest` z biblioteki o tej samej nazwie. Na wykresie widzimy, że najlepsza dokładność predykcji wychodzi dla 23 drzew i wynosi ona nieco ponad 0.8. Przedstawiamy wyniki tylko dla $n_{trees} \leq 50$, bo dla większej ilości drzew różnica w dokładności jest już znikoma, a znacznie wzrasta złożoność obliczeniowa. Na podstawie tabeli pomyłek widzimy, że nasz model ma tendencję do lepszego klasyfikowania obserwacji negatywnych (0) niż pozytywnych (1).



	Rzeczywiste 0	Rzeczywiste 1
Estymowane 0	911	179
Estymowane 1	121	194

Tabela 10: Macierz pomyłek dla lasu losowego

3.8 Boosting

Teraz przejdziemy do metody Boosting. Skorzystamy z bibliotek `xgboost` i `pROC`. Dostaliśmy dokładność na poziomie około 0.8, a z macierzy pomyłek możemy odczytać, że ten model ponownie lepiej poradził sobie z klasyfikacją obserwacji negatywnych, niż pozytywnych. Skuteczność wyniosła 0.802847.

	Rzeczywiste 0	Rzeczywiste 1
Estymowane 0	924	169
Estymowane 1	108	204

Tabela 11: Macierz pomyłek dla boostingu

3.9 Bagging

Ostatnim algorytmem, który zastosujemy, będzie algorytm Bagging. W celu kompilacji modelu skorzystamy z biblioteki `ipred`. Dokładność tego modelu wyszła około 0.79, czyli nieznacznie mniej, niż w przypadku dwóch poprzednich modeli. Mamy ponownie sytuację, w której model lepiej klasyfikuje klientów, którzy zostali w firmie, niż tych, którzy odeszli.

	Rzeczywiste 0	Rzeczywiste 1
Estymowane 0	943	207
Estymowane 1	89	166

Tabela 12: Macierz pomyłek dla baggingu

4 Podsumowanie

Spośród wszystkich algorytmów, najlepsza okazała się regresja logistyczna. W tabeli 13 można zobaczyć jak wygląda skuteczność poszczególnych algorytmów.

	Algorytmy	Skuteczność
1	Reg. Lin	0.80
2	Reg. Log	0.82
3	Naive Bayes	0.76
4	k-NN	0.81
5	Decision Tree	0.79
6	SVM	0.80
7	Random Forest	0.79
8	Bagging	0.80
9	Boosting	0.79

Tabela 13: Skuteczność predykcji

Ale skuteczność to nie jedyna istotna statystyka. Weźmy jeszcze pod uwagę, że firmie prawdopodobnie zależało, aby jak najskuteczniej sklasyfikować osoby, które zrezygnowały z umowy, po to aby móc zaproponować klientom o podobnej charakterystyce lepsze oferty. Zatem sprawdźmy skuteczność wśród obserwacji dla których $Churn = 1$.

	Algorytmy	Skuteczność
1	Reg. Lin	0.41
2	Reg. Log	0.56
3	Naive Bayes	0.76
4	k-NN	0.58
5	Decision Tree	0.46
6	SVM	0.50
7	Random Forest	0.52
8	Bagging	0.20
9	Boosting	0.44

Tabela 14: Skuteczność predykcji

Teraz widać, że znacząco zmienia to postać rzeczy. Najlepszy w tym aspekcie okazuje się algorytm Naiwnego Bayesa. Oczywiście nie jest tak, że ta statystyka jest lepsza niż skuteczność, bo mały błąd tutaj oznacza, że w obserwacjach należących do drugiej klasy musiał być istotny błąd. Tutaj należało by zapytać firmę co jest dla nich ważniejsze i w jakim stopniu.

Na koniec dodajmy jeszcze, że można się zastanowić nad ulepszeniem algorytmów. Dla przykładu zbudowaliśmy model, w którym najpierw podzieliliśmy zbiór na dwa podzbiory. W jednym z nich *Contract* był typu *Month-to-month*, drugi był jego dopełnieniem. Na obydwu podzbiorach zastosowaliśmy regresję logistyczną. Okazuje się, że skuteczność wzrosła do poziomu 0.8256 (względem 0.82).

Przeprowadziliśmy też eksperymenty z sieciami neuronowymi z użyciem biblioteki Keras w Pythonie. W oparciu o architekturę 4 warstw (512, 256, 64, 1), zastosowaniem Dropoutu na poziomie 0.5 i krzyżowej entropii jako funkcji kosztu, sieć taka potrafiła dać skuteczność w granicach 0.81 - 0.82. Jednak po ok. 30 epokach następowało silne przeuczenie. W celu jego uniknięcia zastosowaliśmy EarlyStopping. Poniżej fragment kodu źródłowego, w którym definiujemy model (cały kod źródłowy jest ukryty w pliku `.rmd` ze względu na obszerność). Natomiast na wykresie 14 widzimy, jaką skuteczność miały modele zbudowane przez sieci neuronowe po każdej epoce dla naszego modelu. Ostatecznie udało się uzyskać skuteczność na poziomie 0.8156.

```
model = Sequential()
model.add(Dense(512, activation='relu', input_dim=19))
model.add(Dropout(.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(.5))
model.add(Dense(32, activation='relu'))
model.add(Dropout(.5))
```



```

model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

early_stopping = EarlyStopping(monitor='val_loss',
                              patience=10,
                              verbose=1)
reduce_lr = ReduceLROnPlateau(monitor='val_loss',
                              factor=0.2,
                              patience=5,
                              min_lr=0.001)

hist = model.fit(x_train, y_train, validation_data=(x_test, y_test),
                epochs=50, batch_size=1000, verbose=0,
                callbacks=[early_stopping, reduce_lr])

```



Rysunek 14: Skuteczność sieci neuronowej po każdej epoce