# Probablistyczne uczenie maszynowe: Telco Customer Churn

Kornel Romański, Piotr Błoński
May 3, 2020

**Abstract**

Application of probabilistic machine learning models (Naive Bayes and Bayesian Network) to problem of churn prediction for telecom clients.

## Contents

# 1 Wstęp

Celem naszego projektu jest zbudowanie i zbadanie dwóch modeli probabilistycznego uczenia maszynowego z wykorzystaniem języka Python oraz bibliotek poznanych na laboratoriach - Torch i Pyro. Projekt został zrealizowany bazując na zbiorze danych z serwisu Kaggle.

Podjętym problemem jest rezygnacja klientów z usług firmy telekomunikacyjnej. W zbiorze danych znajdują się informacje o klientach oraz świadczonych im usługach, takie jak rodzaj łącza internetowego klienta, jego dane demograficzne. Każdy klient jest dodatkowo przydzielony do klasy, której wartość pozytywna oznacza, że zrezygnował z usług firmy. Jest to więc problem klasyfikacji binarnej. Warto podkreślić jego biznesowe znaczenie, potencjalnie firma telekomunikacyjna chciałaby wiedzieć czym charakteryzują się klienci rezygnujący z usług oraz jak predykować ich odejście w najbliższym czasie, aby móc zareagować z wyprzedzeniem np. oferując nowe warunki umowy.

# 2 Eksploracyjna analiza danych

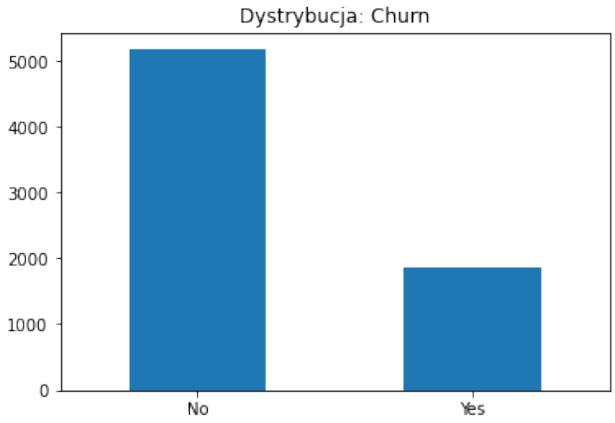
Analizę danych rozpoczęliśmy od zapoznania się z budową zbioru. Zbiór danych posiada 21 cech. Są to:

1. Customer ID - unikalne ID klienta
2. Gender - płeć klienta
3. SeniorCitizen - informacja 0/1 oznaczająca czy dany klient jest Seniorem
4. Partner - informacja 0/1 czy klient posiada partnera
5. Dependents - informacja 0/1 czy klient posiada zasiłek bądź jest na utrzymaniu?
6. Tenure - liczba miesięcy - jak długo klient korzysta z usługi.
7. PhoneService - informacja 0/1 czy klient korzysta z usługi telefonicznej
8. MultipleLines - informacja 0/1 czy klient ma wiele łącz
9. InternetService - informacja 0/1 czy klient korzysta z usług dostępu do internetu
10. OnlineSecurity - informacja 0/1/‘no internet service‘ czy klient korzysta z zabezpieczeń online.
11. OnlineBackup - informacja 0/1/‘no internet service‘ czy klient korzysta z kopii zapasowych online.
12. DeviceProtection - informacja 0/1/‘no internet service‘czy klient korzysta z zabezpieczeń urządzenia.
13. TechSupport - informacja 0/1/‘no internet service‘ czy klient korzysta z wsparcia technicznego
14. StreamingTV - informacja 0/1/‘no internet service‘ czy klient korzysta ze streamingu telewizji
15. StreamingMovies - informacja 0/1/‘no internet service‘ czy klient korzysta ze streamingu filmów
16. Contract - Umowa miesięczna/Roczna/Dwuletnia‘- informująca na jak długo klient podpisał umowę
17. PaperlessBilling - informacja 0/1 czy klient korzysta z rachunków online
18. PaymentMethod - Rachunek elektroniczny/Rachunek listowy/Automatyczny przelew/Karta kredytowa - rodzja płatności wybrany przez klienta
19. MonthlyCharge - informacja numeryczna wskazująca wysokość miesięcznych płatności
20. TotalCharge - informacja numeryczna wskazującą ile w sumie klient zapłacił za usługi

Ostatnim, szczególnie interesującym atrybutem jest ‘Churn‘, czyli informacja o tym czy klient w ostatnim miesiącu zrezygnował z usług. Właśnie ten atrybut jest klasą, którą będziemy chcieli przewidywać naszymi modelami.

Zbiór składa się z 7070 obserwacji bez wartości brakujących, wśród których dystrybucja klas jest mocno niezbalansowana. Osób które zrezygnowały z usług telekomunikacyjnych jest tylko 1896, natomiast tych, którzy nie zrezygnowali aż 5174.

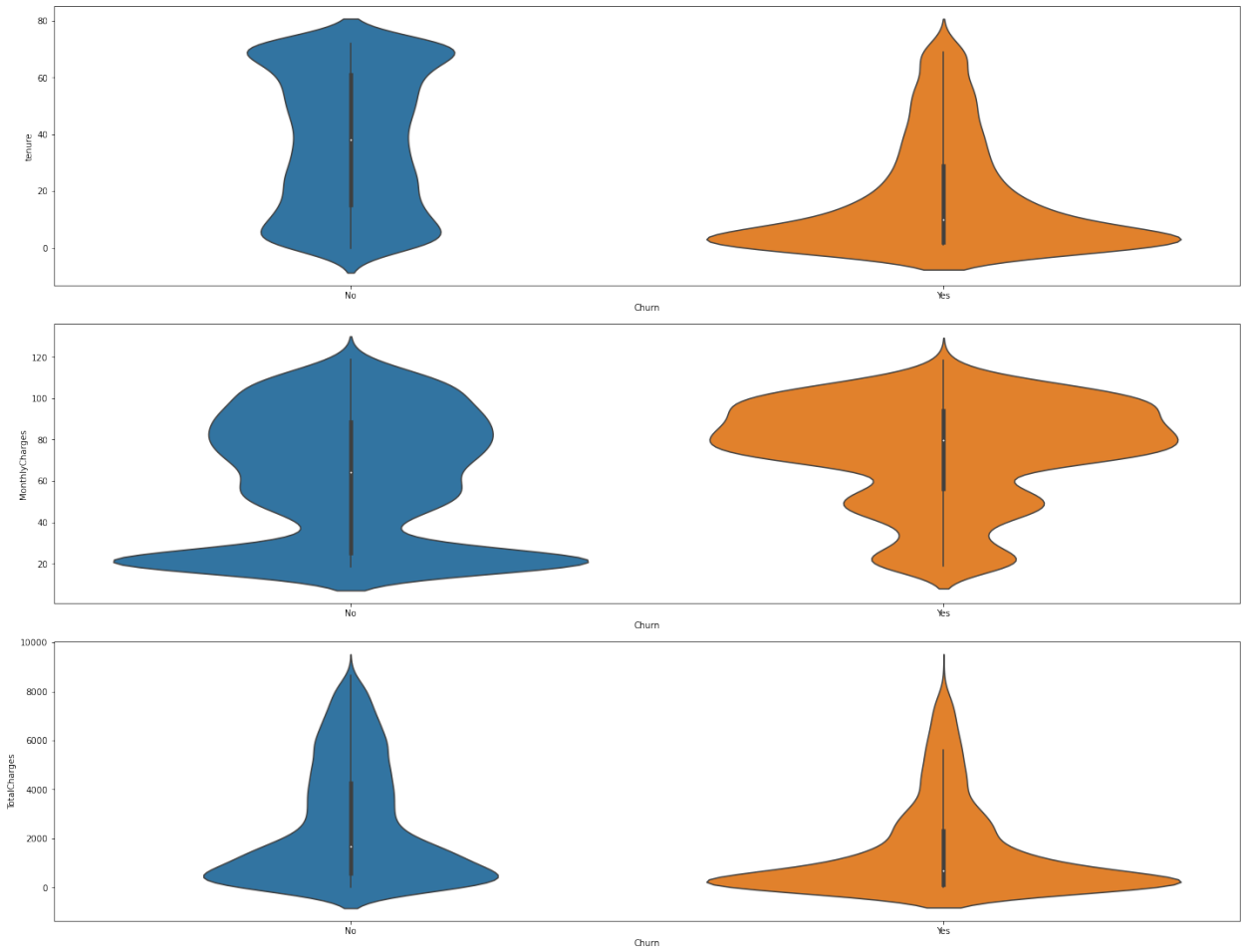
Figure 1: Dystrybucja klasy w zbiorze. 5174 dla ‘No‘, 1896 dla ‘Yes‘




## 2.1 Zmienne numeryczne

Następnie przyjrzeliśmy się wartościom posiadanych zmiennych numerycznych, ze względu na klasy, do których należeli klienci. Na przedstawionych wykresach skrzypcowych można zauważyć że większość z klientów, którzy

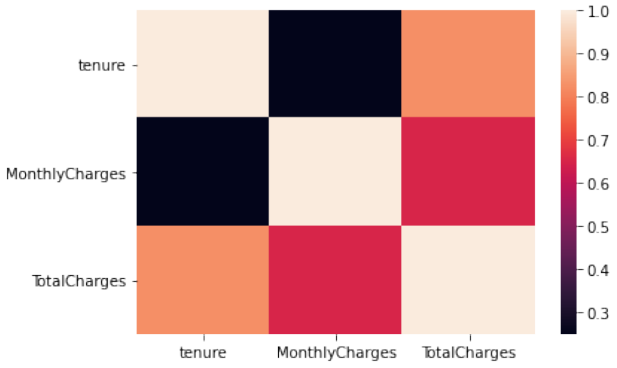
Figure 2: Wykresy skrzypcowe dla atrybutów: Tenure, MonthlyCharge, TotalCharge z w zależności od Churn



zrezygnowali z usług, korzystało z nich krócej niż 20 miesięcy oraz płacili oni wyższe opłaty miesięczne w porównaniu z klientami, którzy nie zrezygnowali z usług.

Sprawdziliśmy również korelacje Pearsona między atrybutami numerycznymi. Na przedstawionej macierzy widać oczywiste korelacje między Tenure i TotalCharges oraz TotalCharges i MonthlyCharges, natomiat bardziej interesujący jest fakt braku jakiejkolwiek korelacji między Tenure i MonthlyCharges, z czego wynika, że klienci korzystający dłużej z usług nie płacą mniejszych rachunków. Byś może otrzymują oni bardziej atrakcyjne oferty, jednak raczej mogłyby to być dodatkowe usługi w tej samej cenie, a nie niższe rachunki.

Figure 3: Macierz korelacji dla atrybutów: Tenure, MonthlyCharge, TotalCharge




## 2.2 Dane demograficzne

Dalej, podczas analizy atrybutów kategorycznych, wykresy słupkowe będą przedstawiane zgodnie z zasadę: z lewej strony liczność obserwacji o podanej wartości atrybutu we wskazanej klasie, natomiest z prawej ich proporcja względem wszystkich obserwacji z taką wartością atrybutu.
Na wykresach 4 i 5 można zauważyć, że:

- Płeć nie ma związku z rezygnacją z usług.
- Wśród emerytów jest większa szansa na rezygnację, niż wśród klientów, którzy nie są emerytami.
- Wśród osób rezygnujących jest mniejsza szansa, że klient ma partnera lub otrzymuje inne osoby.

## 2.3 Dane o wykorzystywanych usługach

Z wykresów 6, 7, 8 i 9, dotyczących danych o tym z jakich usług korzystają klienci, możemy wyczytać, że:

- Większość klientów korzysta z linii telefonicznej, ale nie ma to związku z rezygnacją.
- Klienci korzystający z internetu DSL rezygnują rzadziej niż klienci korzystający ze światłowowdu.
- Wśród klientów korzystających ze światłowodu zrezygnowała ok 40%.
- Klienci korzystający z usług dodatkowych, takich jak OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport mają mniejszą szansę na rezygnację.
- Atrybuty StreamingTV i StreamingMovies mają niewielki związek z rezygnacją.

Figure 4: Wykresy rozkładu Churn w zależności od Płci, SerniorCitizen

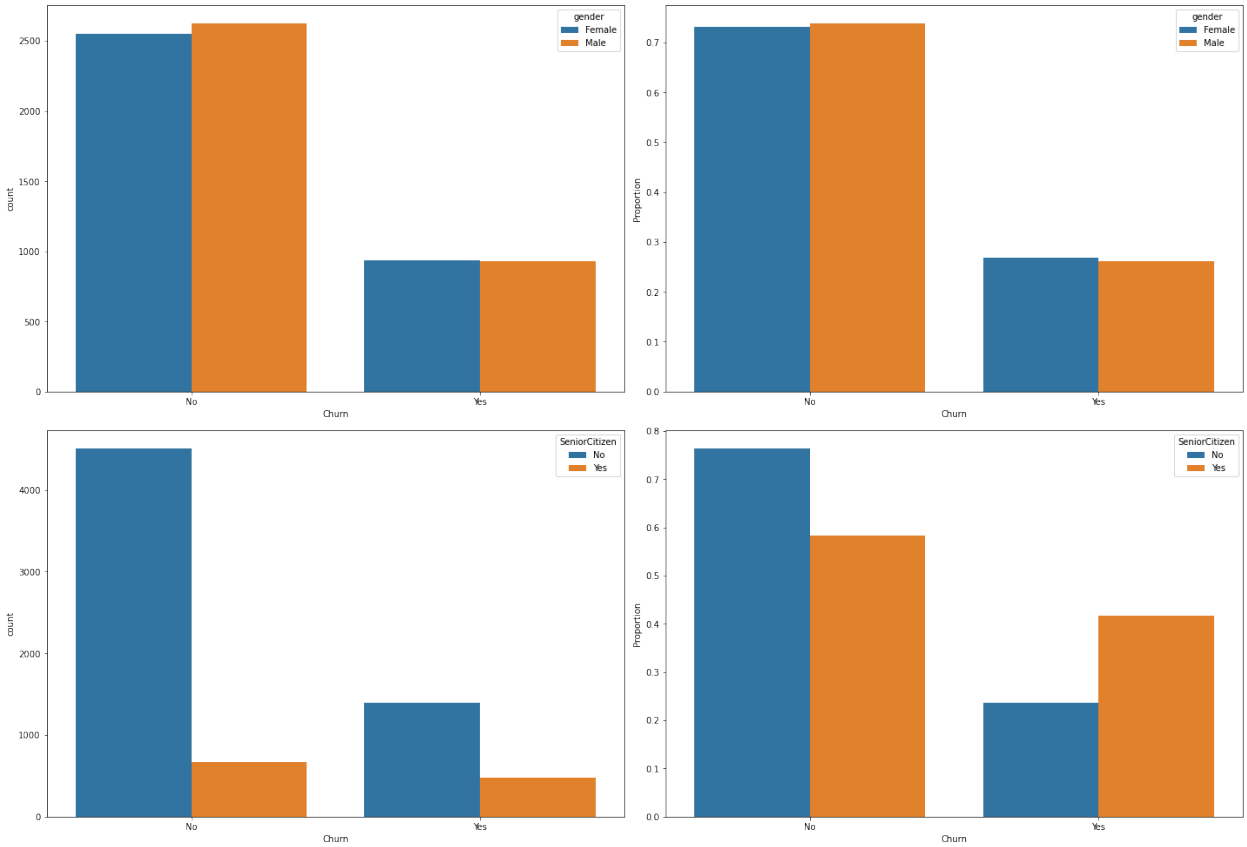


Figure 5: Wykresy rozkładu Churn w zależności od Partner, Dependents

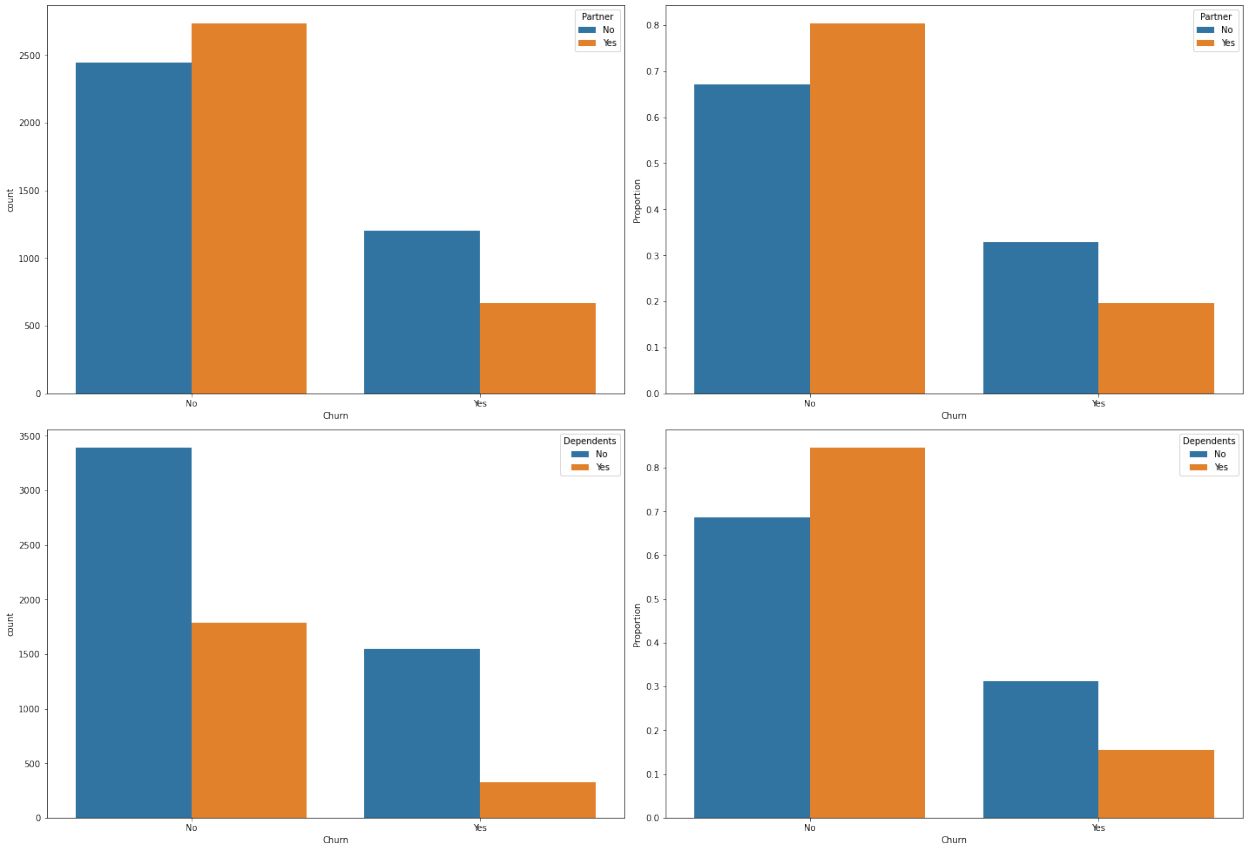


Figure 6: Wykresy rozkładu Churn w zależności od PhoneService i MultipleLines

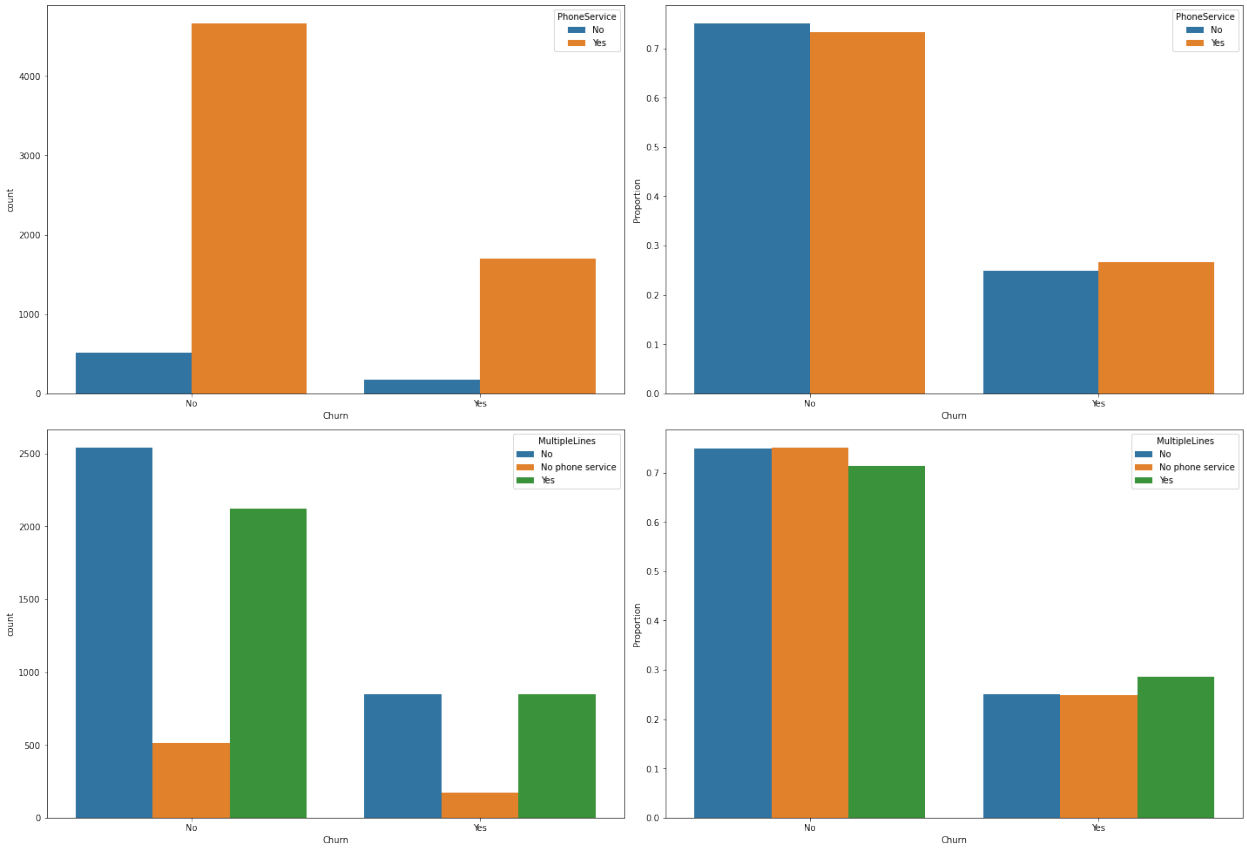


Figure 7: Wykresy rozkładu Churn w zależności od InternetService i OnlineSecurity

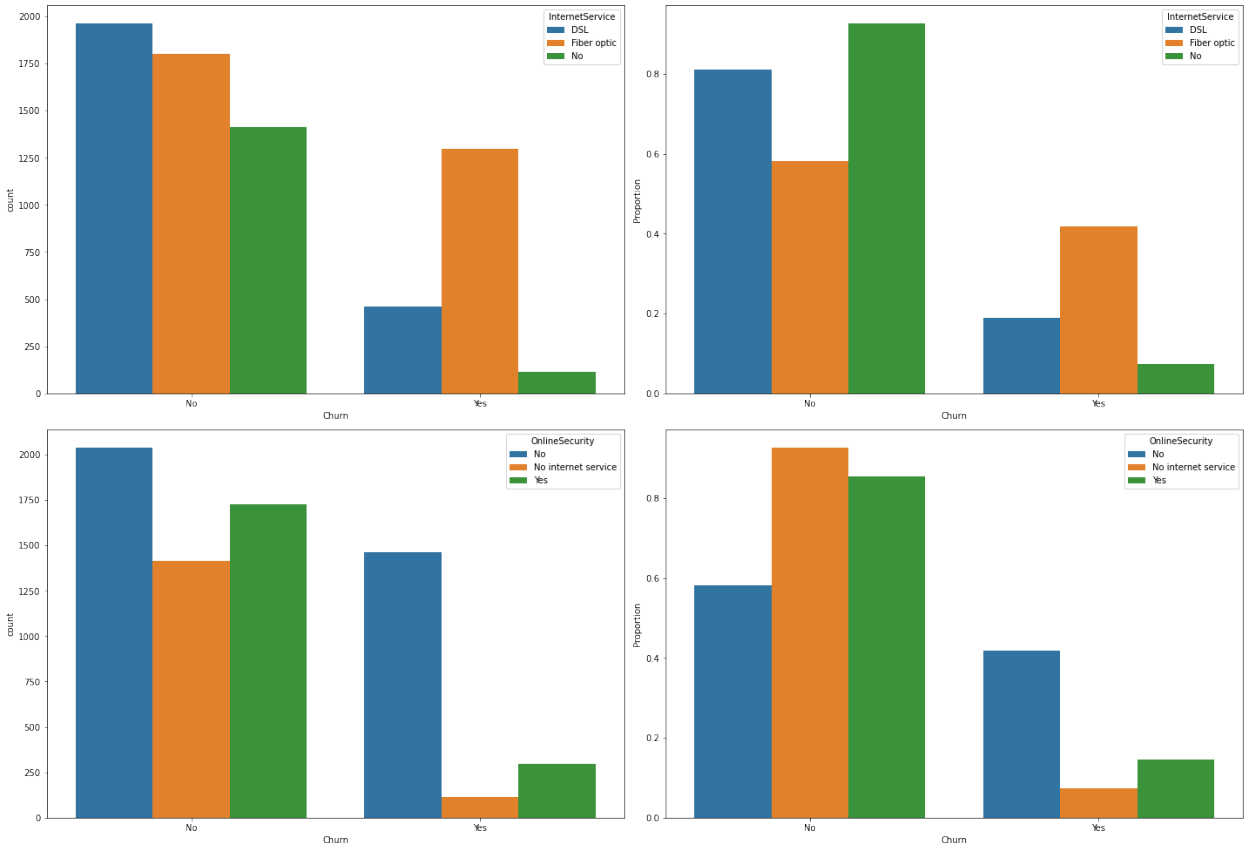


Figure 8: Wykresy rozkładu Churn w zależności od OnlineBackup i DeviceProtection

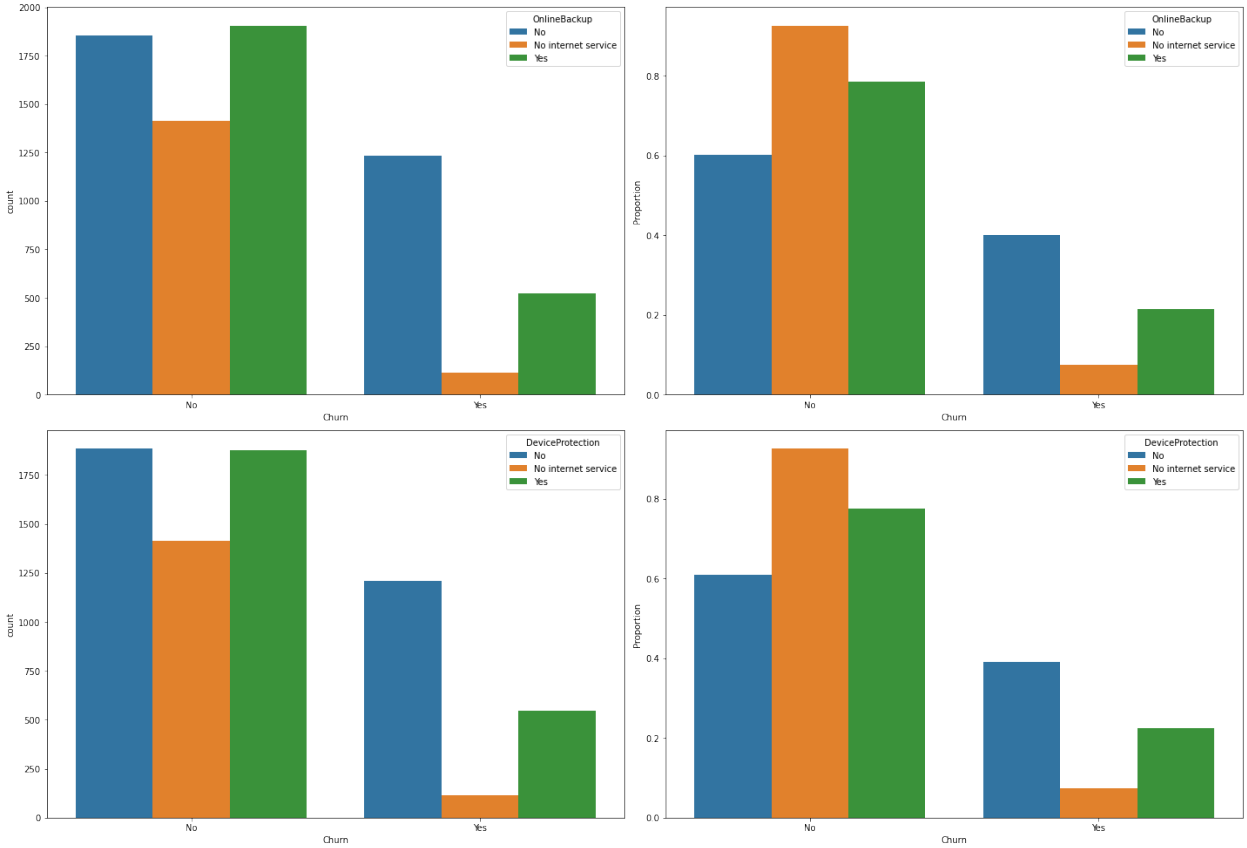
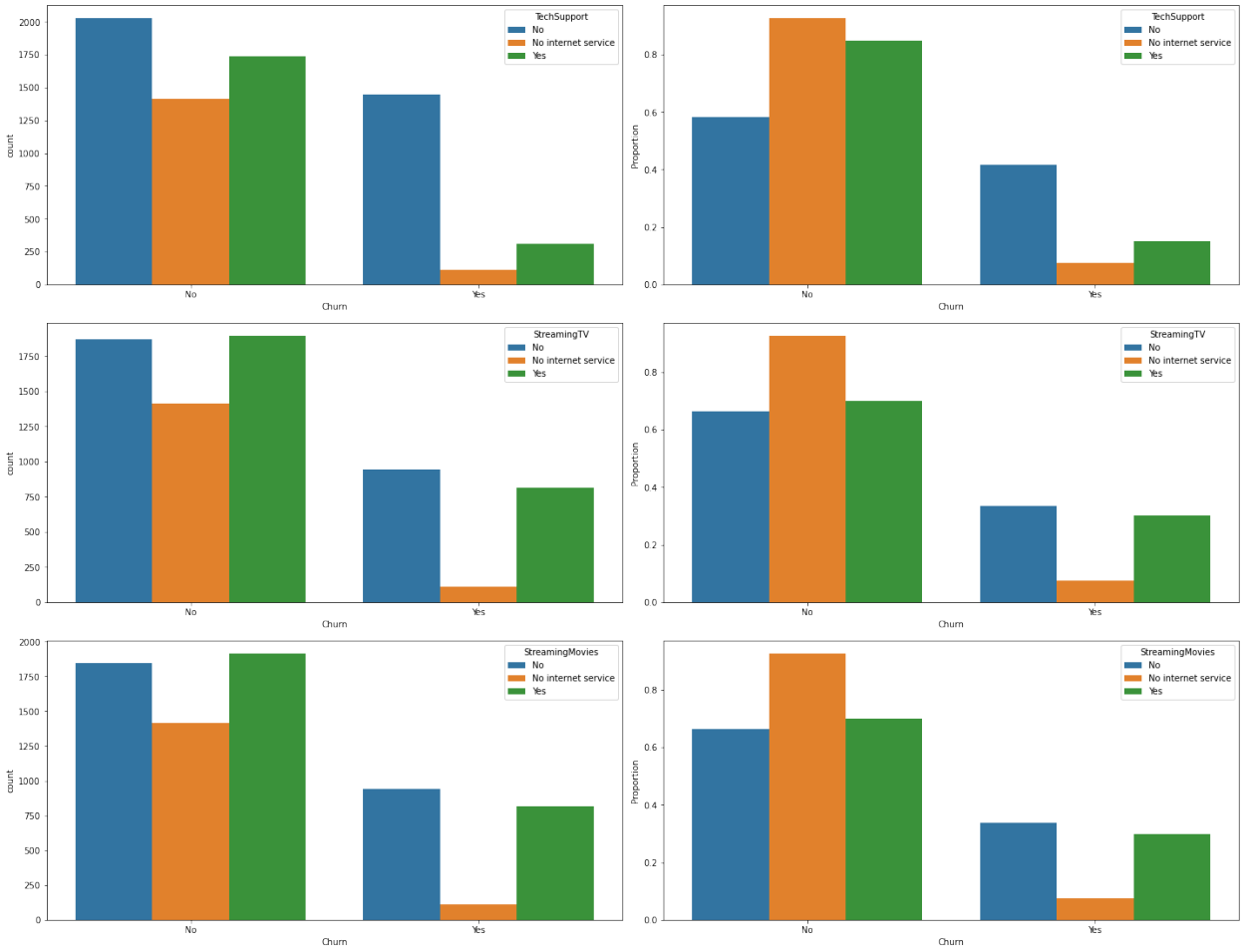
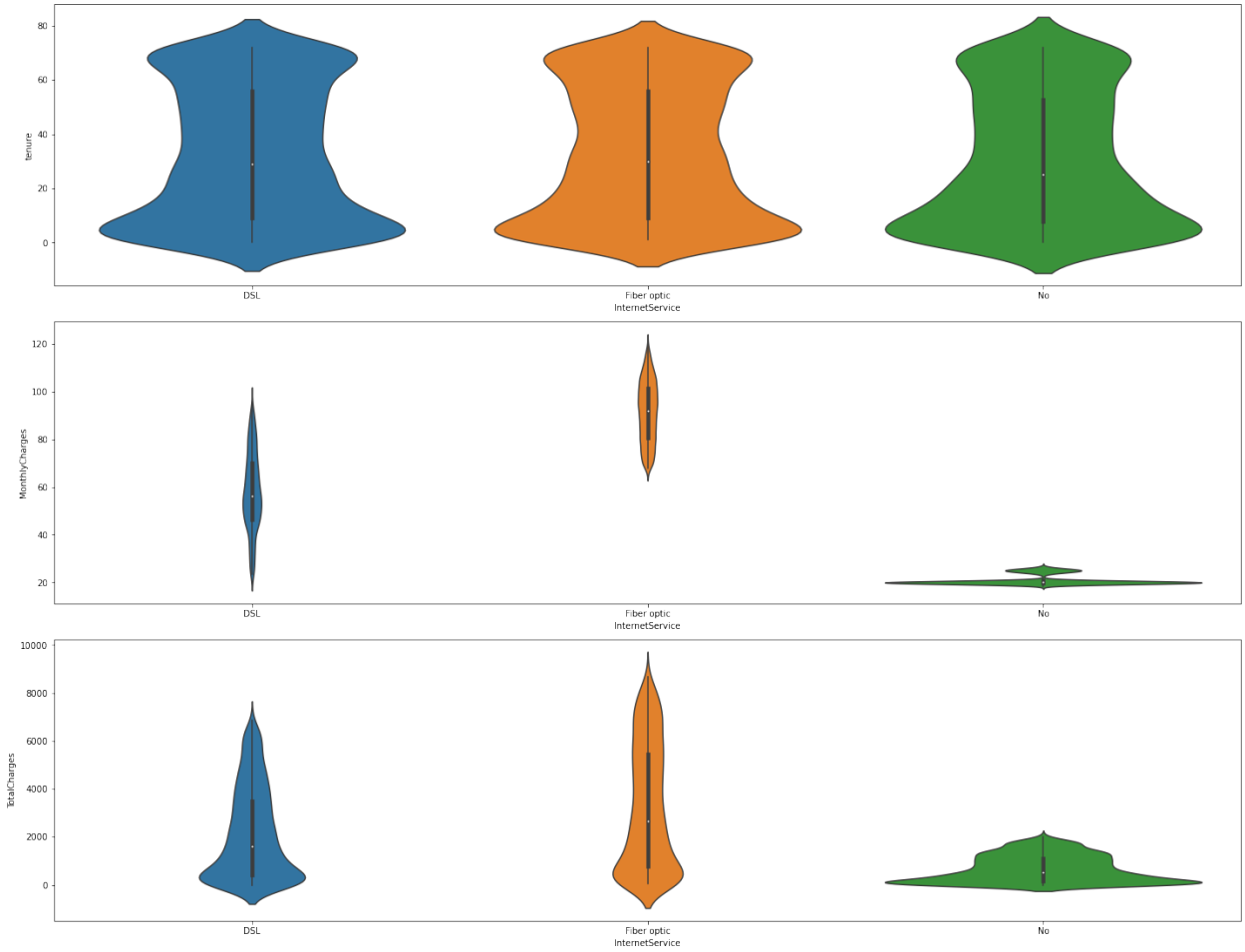


Figure 9: Wykresy rozkładu Churn w zależności od TechSupport, StreaminTV, StreamingMovies



Zgłębiając temat rezygnacji klientów korzystających z internetu światłowodowego sprawdziliśmy wartości atrybutów numerycznych z podziałem na wykorzystywane łącze internetowe. Na poniższych wykresach możemy zauważyć, że klienci z łączem światłowodowym płacą dużo wyższe rachunki.

Figure 10: Wykresy skrzypcowe tenure, MonthlyCharges, TotalCharges w zależności od rodzaju łącza internetowego




## 2.4 Dane dotyczące umowy i płatności

Kolejnym zestawem atrybutów są te, które dotyczą długości umowy, korzystania z elektronicznych rachunków i stosowanej metody płatności. Wykresy 11, 12, 13 i 14 pokazują, że:

- Wśród rezygnujących klientów jest większa szansa, że mają najkrótszy rodzaj umowy, korzystają z elektronicznych rachunków, które opłacają przelewem elektronicznym.
- Klienci bez stałej umowy są najkrócej związani z firmą.
- Klienci korzystający z elektronicznych rachunków mają jednocześnie wyższe opłaty.
- Klienci korzystający z elektronicznych przelewów mają również wyższe opłaty i są krócej związani z firmą.

Figure 11: Wykresy rozkładu Churn w zależności od Contract, PaperlessBilling, PaymentMethod

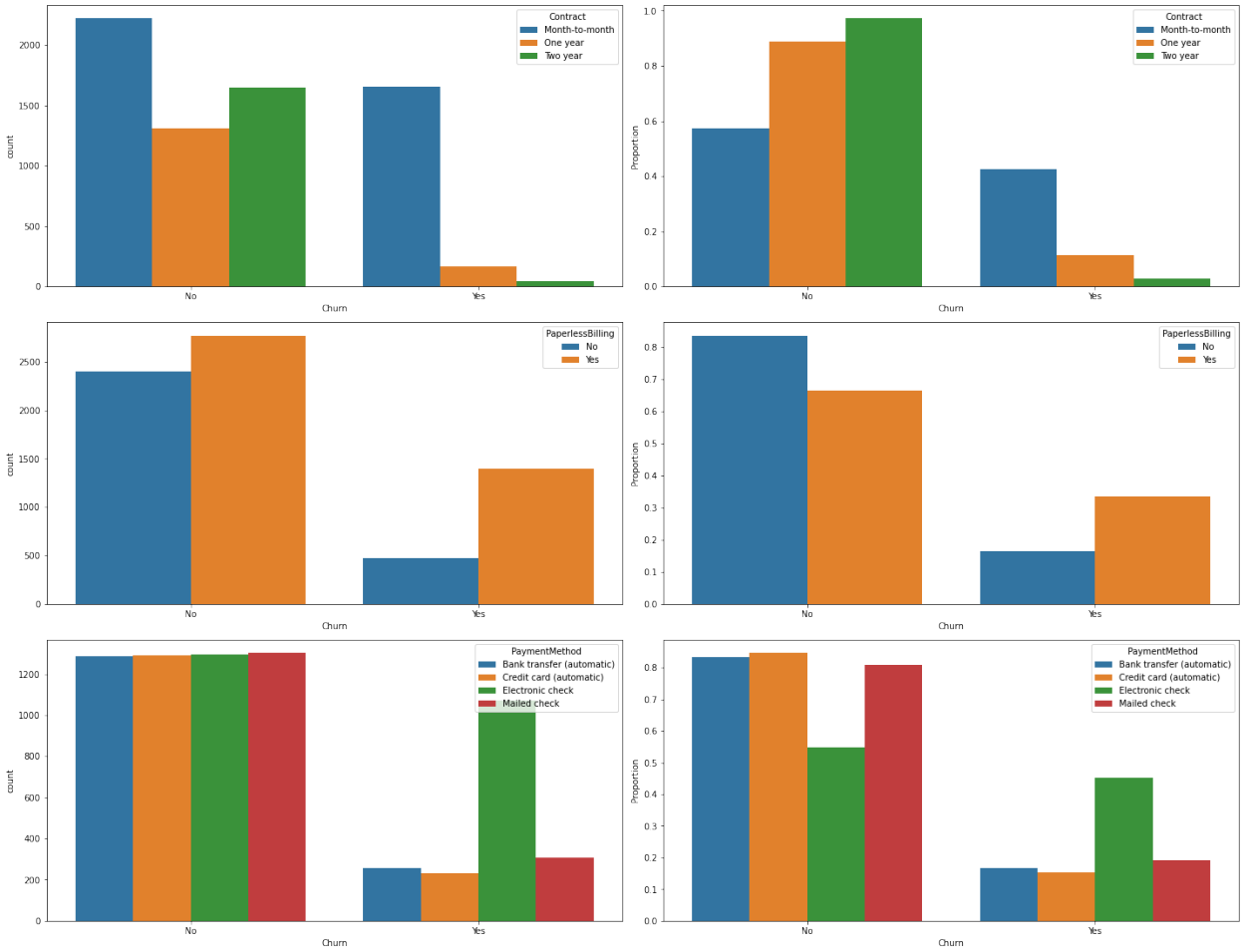


Figure 12: Wykresy skrzypcowe tenure, MonthlyCharges, TotalCharges w zależności od rodzaju umowy

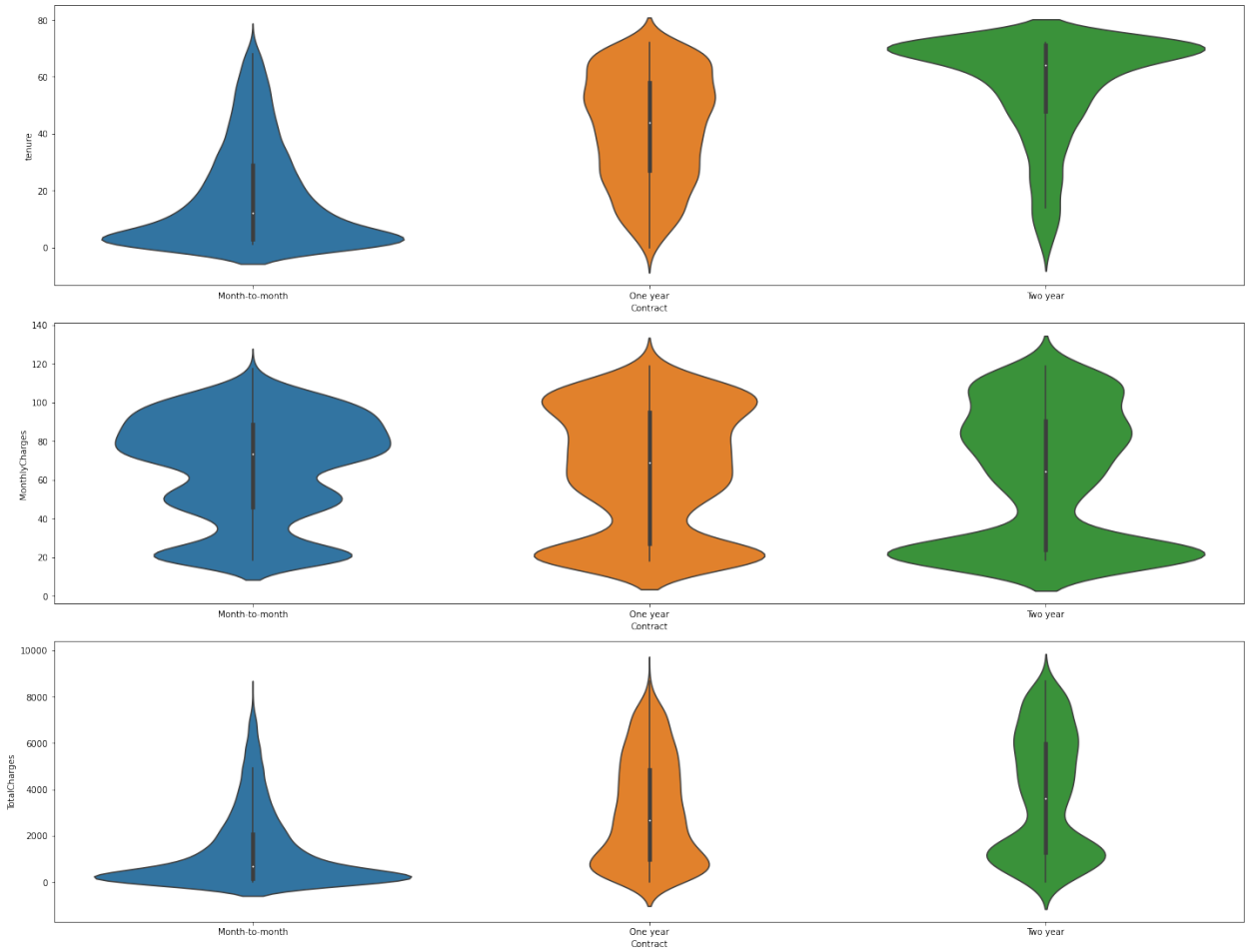


Figure 13: Wykresy skrzypcowe tenure, MonthlyCharges, TotalCharges w zależności od korystania z elektronicznych rachunków

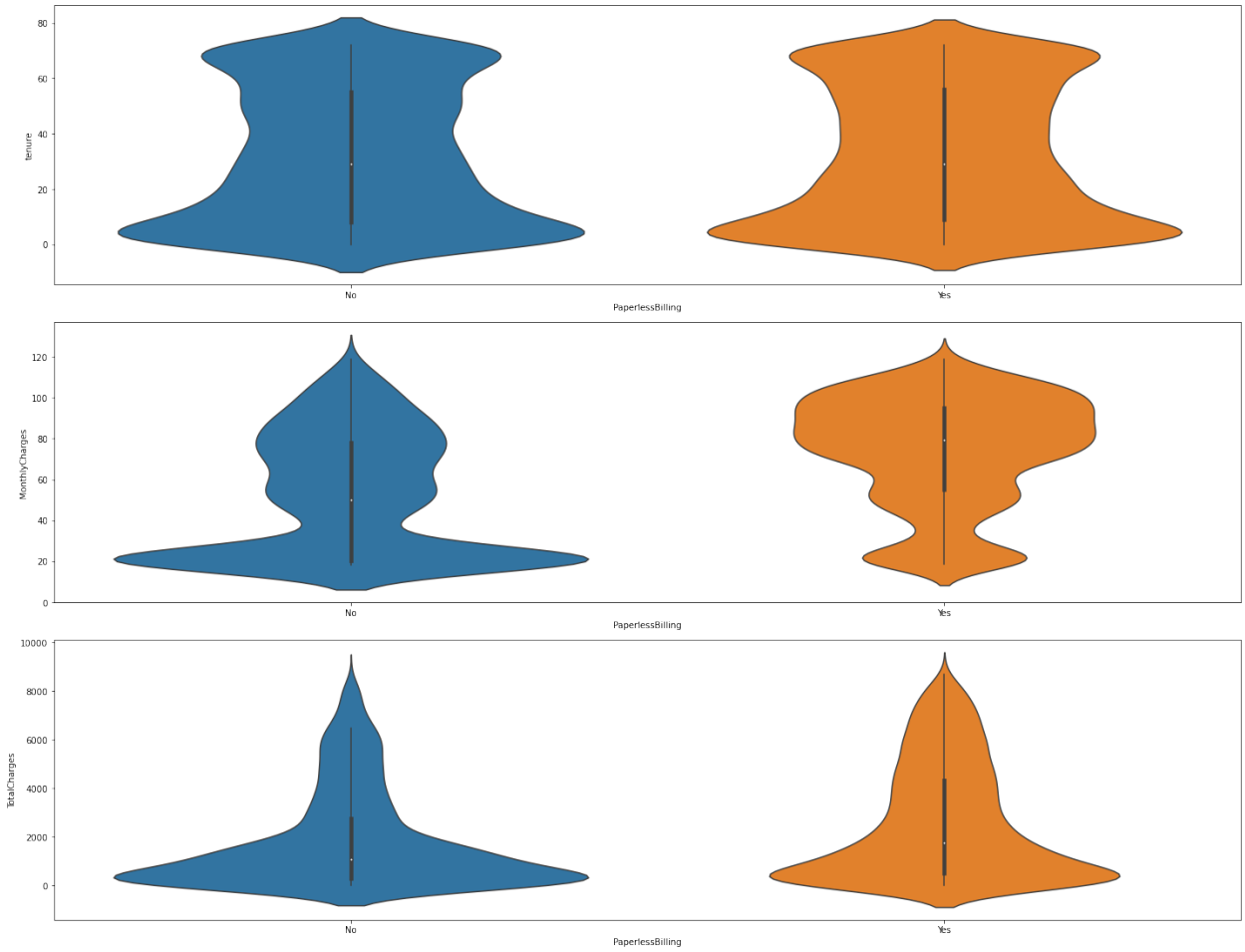
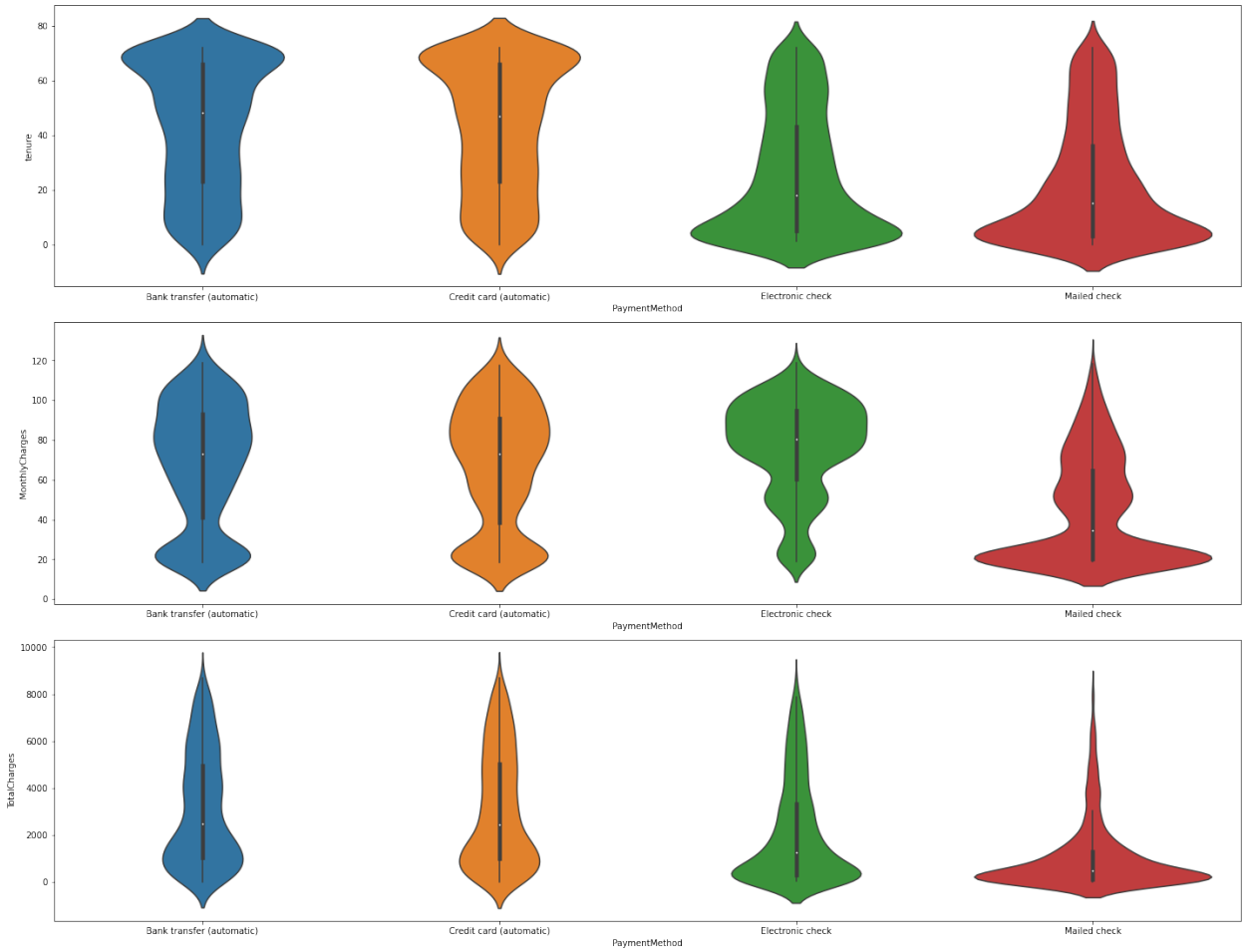


Figure 14: Wykresy skrzypcowe tenure, MonthlyCharges, TotalCharges w zależności od metody płatności




## 2.5 Podsumowanie

Analizę danych można podsumować stwierdzając, że wśród klientów rezygnujących z usług jest większa szansa wystąpienia poniższych cech:

- Wysokie rachunki.
- Internet światłowodowy.
- Krótki czas korzystania z usług.
- Brak usług dodatkowych, takich jak OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport.
- Brak stałej umowy
- Korzystanie z elektronicznych rachunków i przelewów.

# 3 Implementacja modelu 'Naive Bayes'

Jako prosty model probabilistyczny zdecydowaliśmy się na implementacje modelu naiwnego Bayesa ponieważ jest on prosty w implementacji, łatwo go zrozumieć i nie posiada parametrów z którymi trzeba by było eksperymentować metodą prób i błędów. Na potrzebę projektu została stworzona prosta implementacja tego modelu która tworzy i uczy 'pyro' parametry dla każdego atrybutu kategorycznego w postaci tensora o kształcie 2 na ilość kategorii. Dla danych numerycznych tworzy 'pyro' parametry które odpowiadają za średnią i wariancje dla rozkładu normalnego dla różnych wariantów atrybutu 'Churn'. Za learning rate przyjęliśmy 1e-2, a ilość epok w okolicy 100. Powinno to zapewnić w miarę stabilne uczenie.

## 3.1 Jak działa naive Bayes?

Pierwszą rzeczą jaką należy wspomnieć o tym modelu jest to że ten model zakłada że atrybuty/cechy są od siebie niezależne, dlatego też nazywany jest 'naiwnym'. Po wcześniejszej analizie danych wiemy że w naszym przypadku tak nie jest. Istnieją u nas korelacje pomiędzy danymi co będzie miało wpływ na końcową sprawność modelu.

Model ten zgodnie z twierdzeniem Bayesa uczy się prawdopodobieństwa zajścia zdarzeń np. Bycia kobietą i zrezygnowanie z usług. W naszym przypadku prawdopodobieństo takiego zdarzenia przechowywane jest w parametrze o nazwie 'f'gender_probability" który zawiera takie informacje jak: prawdopodobieństo pozostania (Churn=0) podczas bycia mężczyzną i podczas bycia kobietą jako parametr dla rozkładu kategorycznego, jak i analogicznie prawdopodobieństwa dla opcji zrezygnowania z usług (Churn=1). Aby uzyskać prawdopodobieństwa zajścia zdarzeń 'Churn' używamy sumy logarytmów prawdopodobieństw zajścia poszczególnych zdarzeń. Używamy sumy logarytmów zamiast iloczynu prawdopodobieństw ze względu na to że jeżeli jakieś zdarzenie miało by zerowe prawdopodobieństwo zerowało by cały posterior (oprócz tego łatwiej w pyro operować na logarytmach prawdopodobieństw). Klasę $y$ określamy na podstawie największego prawdopodobieństwa zbioru cech $x$ pod warunkiem bycia zaobserwowanym razem z $y$.[16]

Figure 15: Określanie prawdopodobieństwa posterior na podstawie wnioskowania Bayesowskiego

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

Figure 16: Określanie klasy na podstawie wnioskowania

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

# 4 Implementacja modelu 'Bayesian Network'

Bayesian Network, czyli Sieć Bayesa, należy do rodziny graficznych modeli probabilistycznych. Ich głównym założeniem jest to, że występujące w opisywanym fragmencie rzeczywistości zdarzenia i zależności między nimi można przedstawić za pomocą grafu. Prezentuje on prawdopodobieństwo łączne zdarzeń przy założonych zależnościach. Zdarzeniom, czyli zmiennym losowym, odpowiadają wierzchołki, natomiast zależności, to krawędzie między nimi. Istnieją różne rodzaje modeli graficznych, które są reprezentowane przez grafy o innych właściwościach. Sieć Bayesa należy do graficznych modeli skierowanych, czyli takich, których graf posiada tylko krawędzie skierowane.

W omawianym modelu można wyróżnić zmienne losowe widoczne x_v, czyli takie, których wartości zaobserwowaliśmy oraz zmienne ukryte x_h, których wartości nie zaobserwowaliśmy. Kolejnym elementem są też parametry modelu $\theta$. Są to macierze przejść pomiędzy zmiennymi, które definiują rozkłady prawdopodobieństwa dla wartości zmiennej losowej w zależności od wartości jej rodziców. Uczenie modelu polega na znalezieniu optymalnych parametrów, zazwyczaj stosując metodę estymacji Maximum A Posteriori (MAP):

$$\widehat{\theta} = \underset{\theta}{argmax} \sum_{i=1}^{N} = \log p(x_{i,v}|\theta) + \log(\theta)$$

Podczas przeprowadzania wnioskowania na nauczonym modelu można określić prawdopodobieństwa dla wartości zmiennych ukrytych:

$$p(x_h|x_v, \theta) = \frac{p(x_h, x_v|\theta)}{p(x_v|\theta)} = \frac{p(x_h, x_v|\theta)}{\sum_{x'_h} p(x'_h, x_v|\theta)}$$

Dokładniejszy matematyczny opis modeli graficznych znaleźć można w [1]

W badaniach zastosowaliśmy sieci o strukturze drzewa, budują zarówno proste modele pełne, w których wszystkie zmienne były obserwowane, jak i modele ze zmiennymi ukrytymi. W tym celu wykorzystaliśmy bibliotekę Pyro oraz jej strukturę do przeprowadzania Stochastic Variational Inference [3]. Jednak, ponieważ wykorzystywaliśmy tylko zmienne dyskretne, nie korzystaliśmy z optymalizacji gradientowej, a wykorzystaliśmy mechanizm enumeracji [2].

Po otrzymaniu modelu z nauczonymi macierzami przejść wykorzystywaliśmy go do predykcji wartości Churn. W naszych modelach zmienna Churn zawsze była liściem, a zmienne obserwowane korzeniami, dlatego realizacja predykcji, polegała na zaobserwowaniu atrybutów wejściowych, następnie losowaniu, zgodnie z rozkładami prawdopodobieństw zawartymi w macierzach przejść, wartości zmiennych ukrytych i na koniec wartości Churn. Jeżeli zmienna miała charakter binarny, to losowana była z wykorzystaniem rozkładu Bernoulli, a jeżeli mogła przyjmować więcej wartości, to wykorzystywany był rozkład Cathegorical.

# 5 Badania modeli

Podczas badań do oceny jakości klasyfikacji wykorzystaliśmy miary: accuracy, precision, recall, f1score oraz macierz pomyłek. Ponieważ pracujemy ze zbiorem o znacznie niezbalansowanym rozkładzie klas, to szczególną uwagę będziemy zwracać na miary precision, recall oraz f1score dla klasy 1.

## 5.1 Naive Bayes

### 5.1.1 Scenariusz badań

W ramach badań przewidziane jest użycie kroswalidacji dla 3, 5, 10 foldów w celu walidacji modelu. Każdy testowany model przejdzie 100 epok. Następnie zostaną zebrane metryki Fscore, Accuracy, Precision i Recall. TODO uzupełnić ręcznie

### 5.1.2 Wyniki

| Folds | F1 'NO' | F1 'Yes' | Recall 'NO' | Recall 'Yes' | Acc | 'Prec 'NO' | Prec 'Yes' |
|---|---|---|---|---|---|---|---|
| 3 | 0.845 | 0.545 | 0.859 | 0.523 | 0.77 | 0.8(3) | 0.572 |
| 5 | 0.847 | 0.552 | 0.859 | 0.53 | 0.77 | 0.83 | 0.58 |
| 10 | 0.843 | 0.553 | 0.851 | 0.54 | 0.76 | 0.837 | 0.56 |

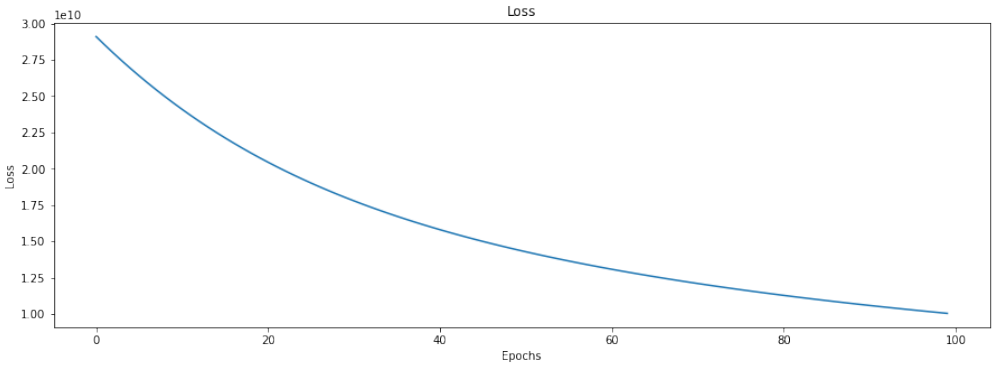



Figure 17: Zmiana wartości loss w trakcie uczenia modelu Naive Bayes

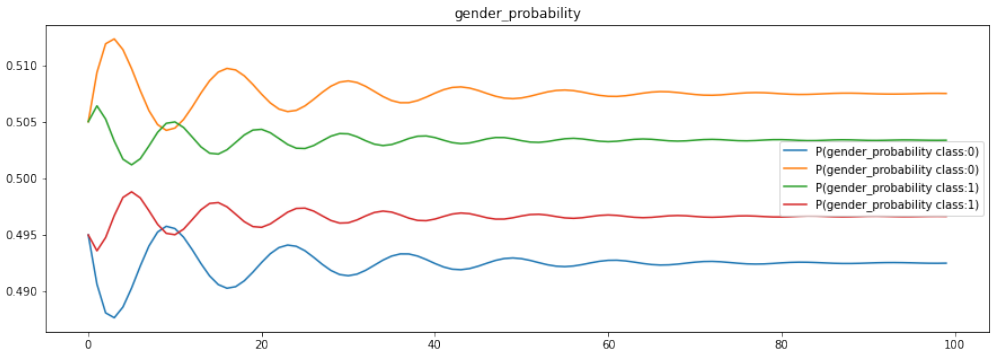



Figure 18: Przykładowy wykres zmiany parametru dla modelu naiwny bayes. Widoczny parametr odpowiada za prawdopodobieństwo płci klienta dla rozkładu kategorycznego

### 5.1.3 Wnioski

Niestety z powodu kiepsko zbalansowanych danych pomimo tego że jest całkiem wysoka metryk Precyzji i F1 dla klientów nieodchodzących (F1=0.84 i Prec 0.83) to prawdopodobieństwo rozpoznanie klienta który chciałby zrezyg-

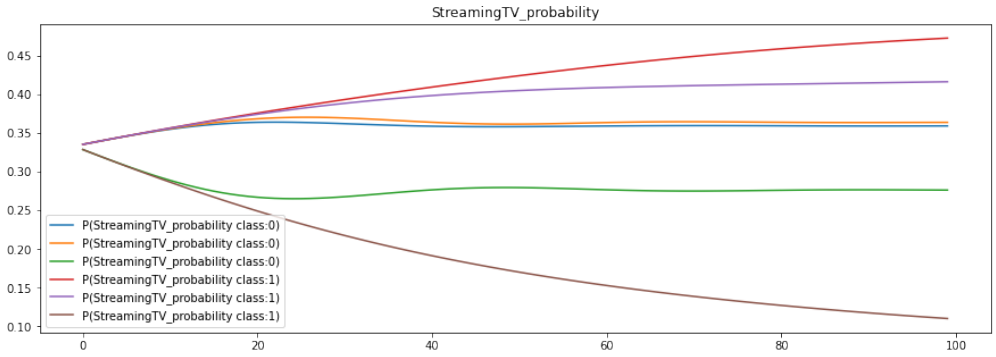



Figure 19: Przykładowy wykres zmiany parametru dla modelu naiwny bayes. Widoczny parametr odpowiada za prawdopodobieństwa posiadania usługi streamingowej

nować z usługi jest dość niskie. Precyzacja i F1 dla rozpoznania klienta rezygnującego wacha się pomiędzy 0.5-0.6. Jest zbyt losowe aby uznać że model ten spełnia swoje zadanie. Za pewnie wiąże się to z naiwnym założeniem naszego modelu jak i z niezbalansowanym datasetem.

## 5.2 Bayesian Network

### 5.2.1 Scenariusz badań

W budowie Sieci Bayesa należy na początku określić jej strukturę. W ramach badań eksperymentowaliśmy z różnymi wariantami, zaczynając od modeli pełnych, a następnie wykorzystując również zmienne ukryte, dobierane w sposób intuicyjny. W każdym modelu przyjmowaliśmy binarne zmienne ukryte. Początkowe prawdopodobieństwo wartości 1 dla zmiennej Churn przyjmowaliśmy jako 1 dla wszystkich kombinacji wartości jej rodziców, natomiast rozkłady prawdopodobieństw pozostałych zmiennych były początkowo zgodne z rozkładem jednostajnym. Na potrzeby uczenia i testowania zbiór danych był dzielony na zbiór treningowy i testowy w proporcjach 0.8:0.2 z zachowaniem rozkładu klas w oryginalnych danych. Uczenie było wykonywane przez 100 epok z parametrem uczenia równym 0.1. W kolejnym kroku model był testowany. Należy tutaj zwrócić uwagę, że pojedyncza realizacja modelu nie jest w pełni deterministyczna, dlatego model był testowany 100 razy. Dla każdej realizacji obliczane był metryki, które następnie zostały uśrednione. Wyjątkiem jest macierz pomyłek, która przedstawia sumę wykonanych predykcji ze wszystkich realizacji modelu. W kolejnych podpunktach przedstawiono wyniki dla konkretnych modeli.

### 5.2.2 Model 1

Pierwszym badanym modelem był prosty model kompletny, w którym wykorzystaliśmy tylko dwie zmienne wejściowe.

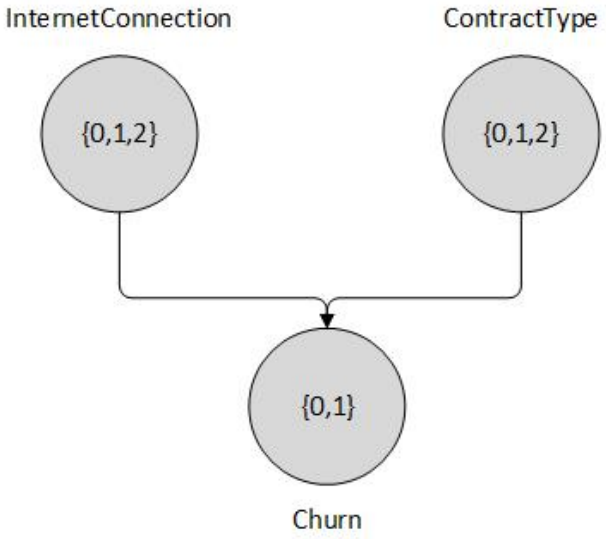



Figure 20: Model 1

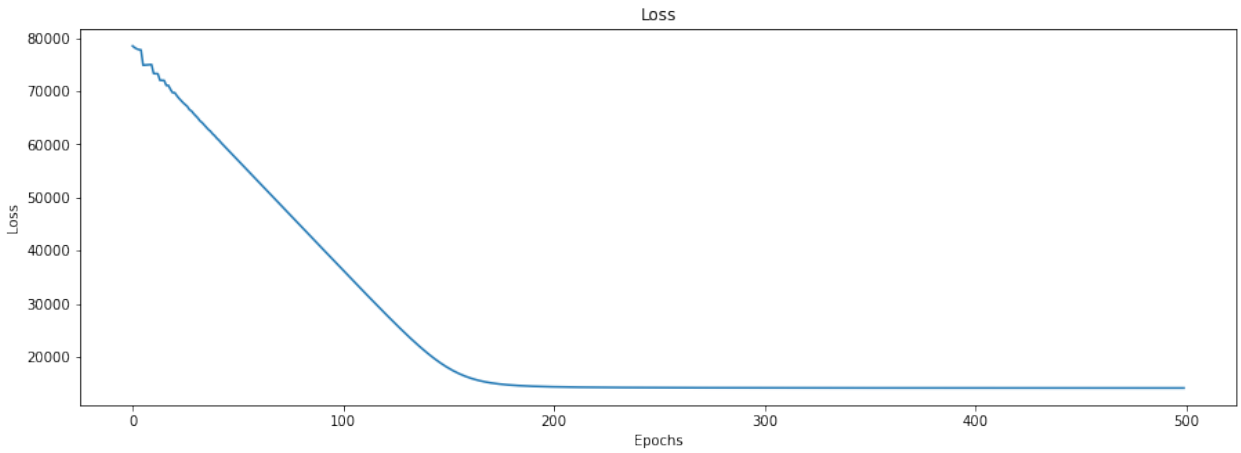



Figure 21: Model 1 - Zmiana wartości loss w trakcie uczenia

```
Accuracy:  0.691
   Precision  Recall  F1Score
0      0.789   0.787    0.788
1      0.428   0.431    0.430
```

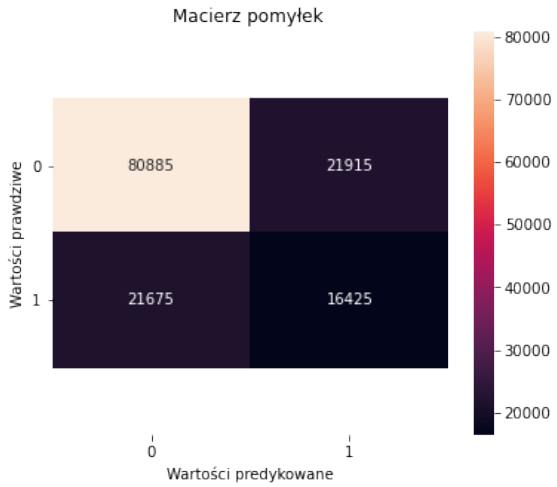



Figure 22: Model 1 - Wyniki

Model, jak na swoją prostą budowę, osiągnął zadowalające wyniki - 43% klientów, którzy zrezygnowali z usług zostało poprawnie zaklasyfikowanych. Jednak trzeba tutaj również zwrócić uwagę na niską wartość precision dla klasy 1. Model popełnia bardzo dużo błędów, w których przypisuje klientów z klasy 0 do klasy 1. Dzieje się tak dla około 20% klientów z klasy 0. Jednak po uwzględnieniu wiedzy domenowej może okazać się, że taki błąd nie będzie problemem.

### 5.2.3 Model 2

W kolejnym kroku wykorzystaliśmy w modelu pełnym wszystkie zmienne wejściowe. Model taki przypomina strukturą modelu NaiveBayes, jednak w tym przypadku zależności między zmiennymi wejściowymi i zmienną wyjściową są odwrotne - przez co nie stosujemy założenia o niezależności zmiennych wejściowych.

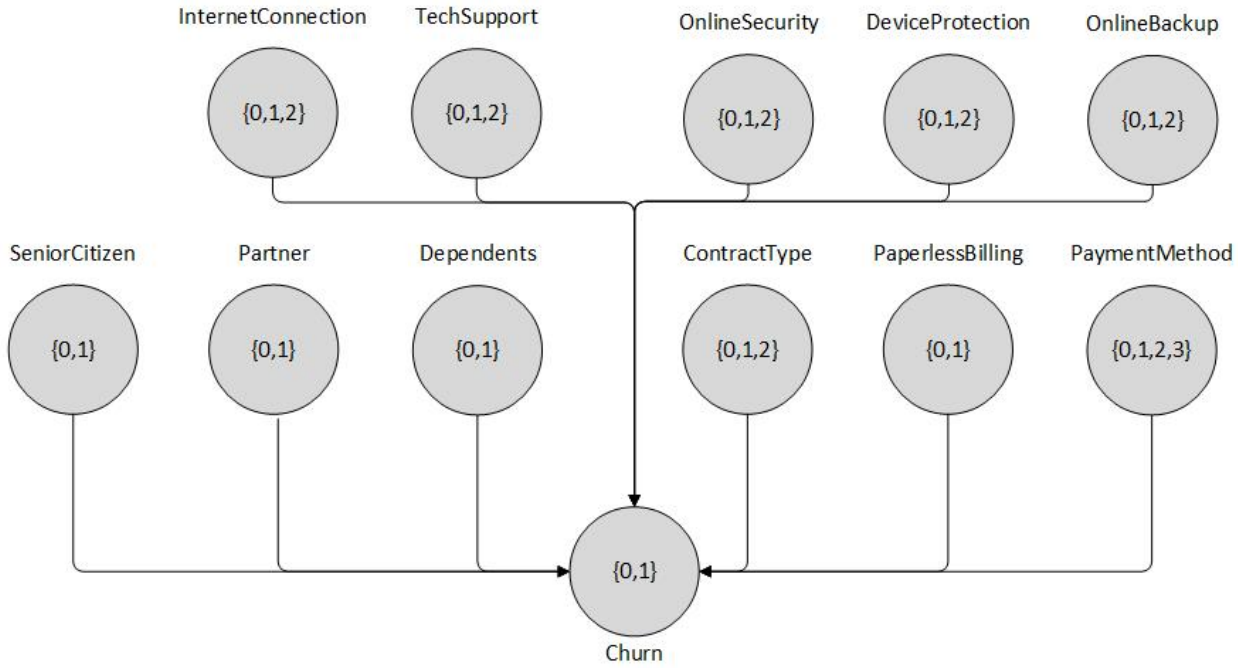



Figure 23: Model 2

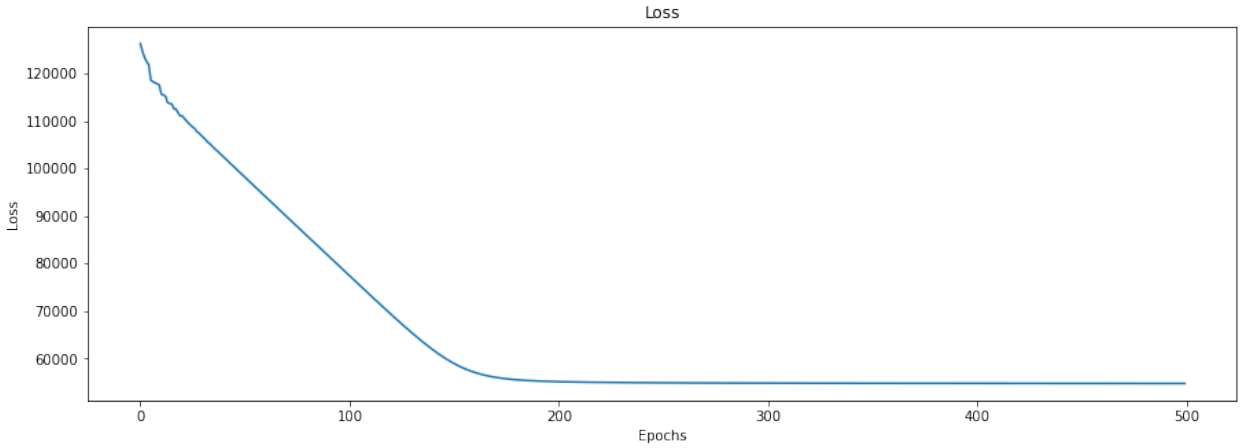



Figure 24: Model 2 - Zmiana wartości loss w trakcie uczenia

```
Accuracy:  0.369
   Precision  Recall  F1Score
0      0.703   0.233    0.350
1      0.262   0.733    0.386
```

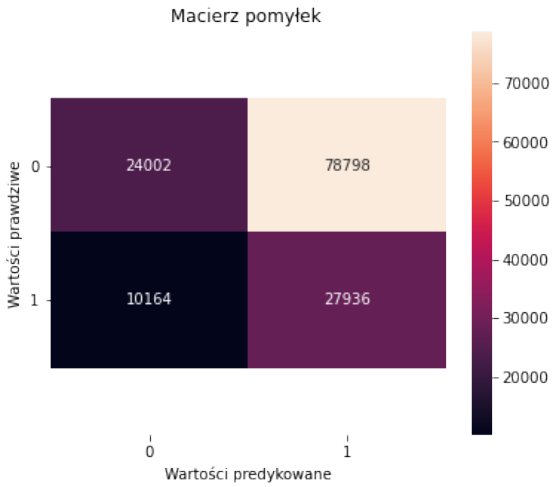



Figure 25: Model 2 - Wyniki

Tym razem można zaobserwować bardzo wysoką wartość recall i niską precision. Można stwierdzić, że ten model niewiele się nauczył i większość przypadków przydziela do klasy 1, co wynika z tego, że początkowe prawdopodobieństwa dla tej klasy wynosiły 1.

### 5.2.4 Model 3

W 3 modelu wykorzystaliśmy zmienne ukryte. Intuicyjnie zbudowaliśmy strukturę, w której chcieliśmy przedstawić pewnie niejawne zależności między zmiennymi wejściowymi, wynikające z ich wcześniejszej analizy.

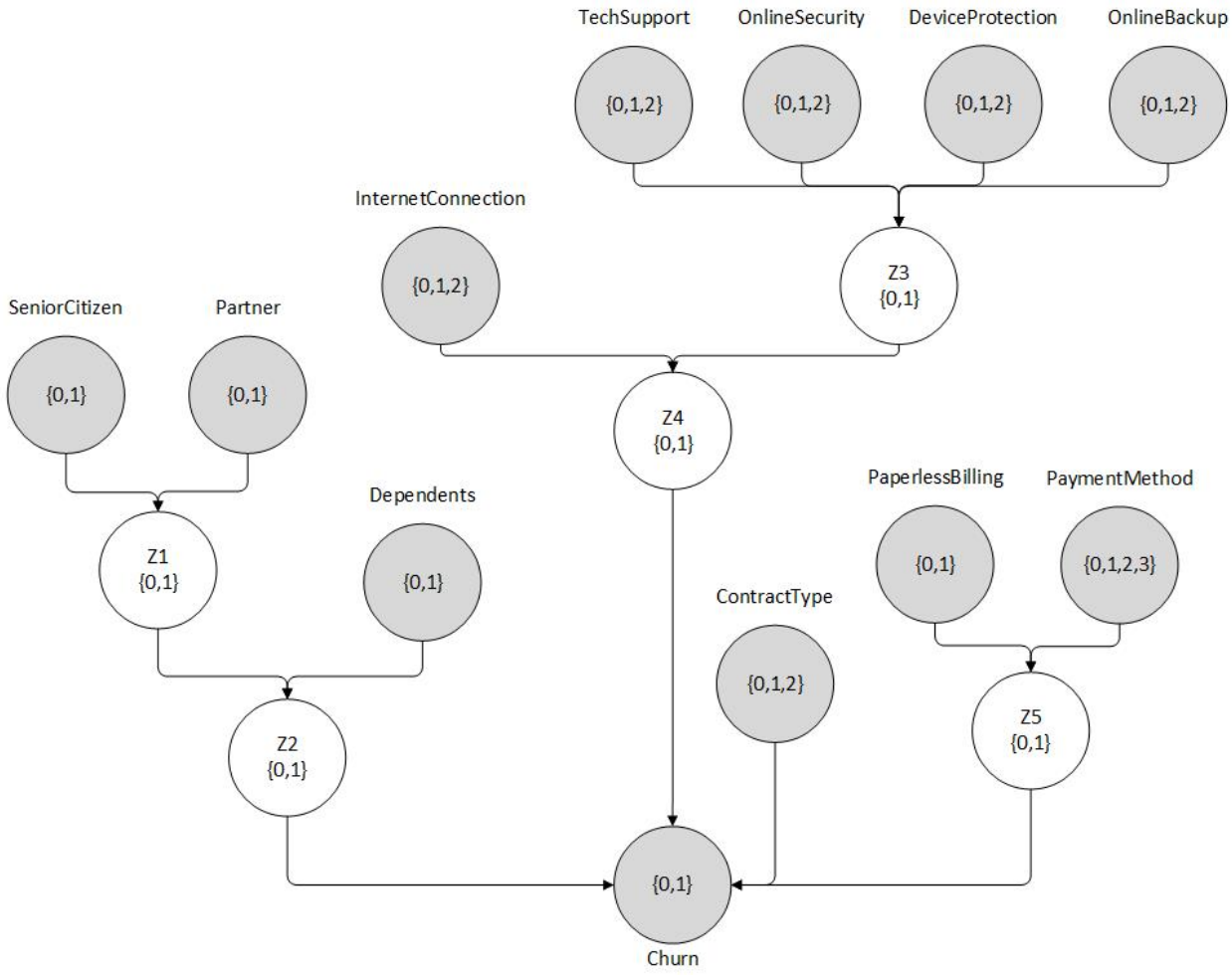



Figure 26: Model 3

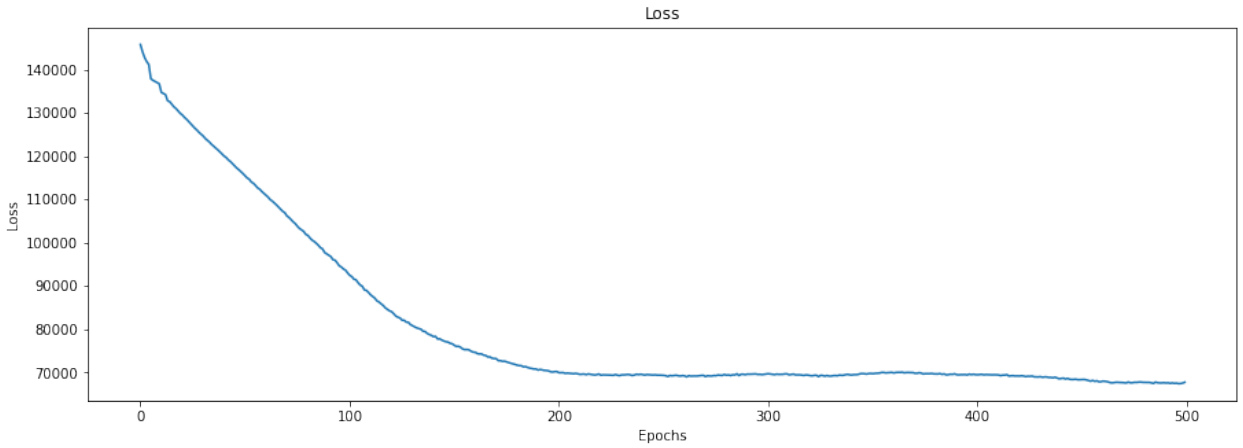



Figure 27: Model 3 - Zmiana wartości loss w trakcie uczenia

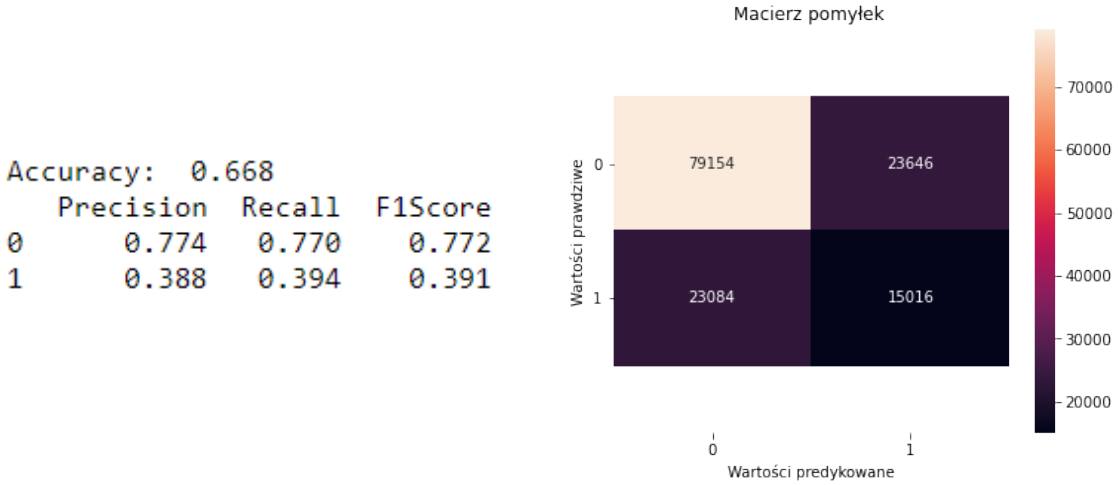



Figure 28: Model 3 - Wyniki

Model o zaproponowanej strukturze dał gorsze wyniki od modelu 1, ale lepsze od modelu 2 - tym razem model czegoś się nauczył i osiągnął accuracy 0.68.

### 5.2.5 Model 4

Kontynuując badania postanowiliśmy badać modyfikacje struktury z poprzedniego modelu. Zaczęliśmy od usunięcia ukrytej zmiennej Z4 i założenia bezpośredniego wpływu InternetConnection na Churn.

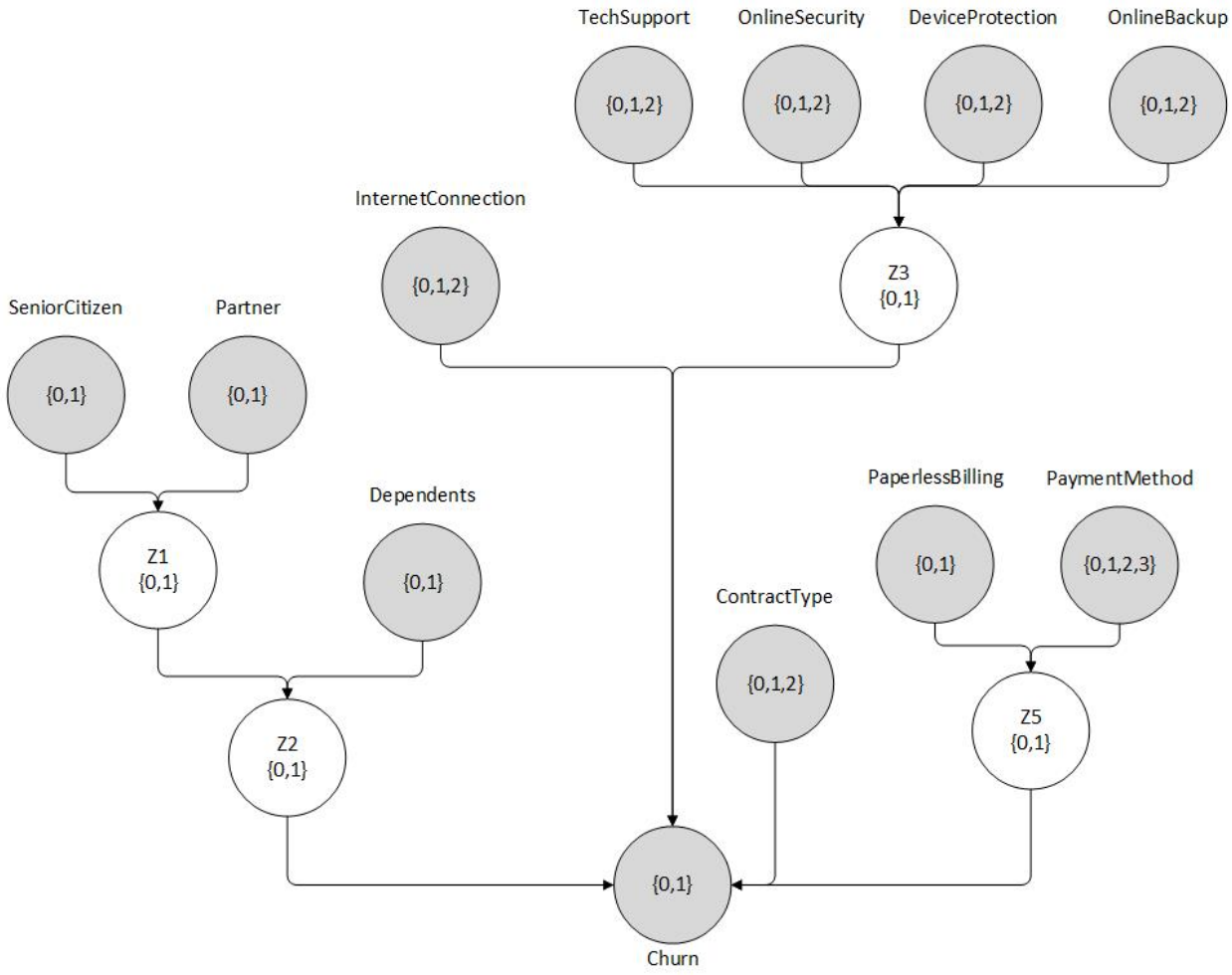



Figure 29: Model 4

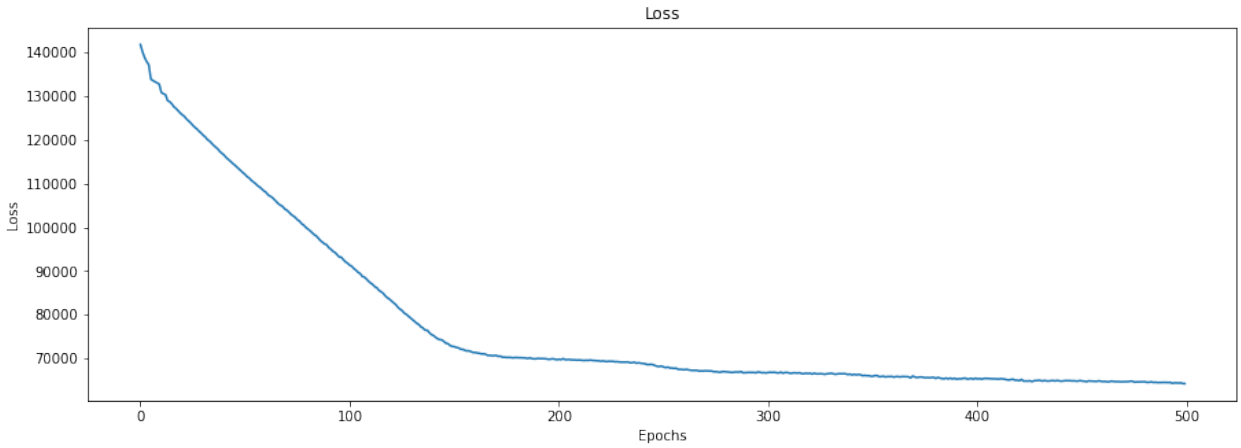



Figure 30: Model 4 - Zmiana wartości loss w trakcie uczenia

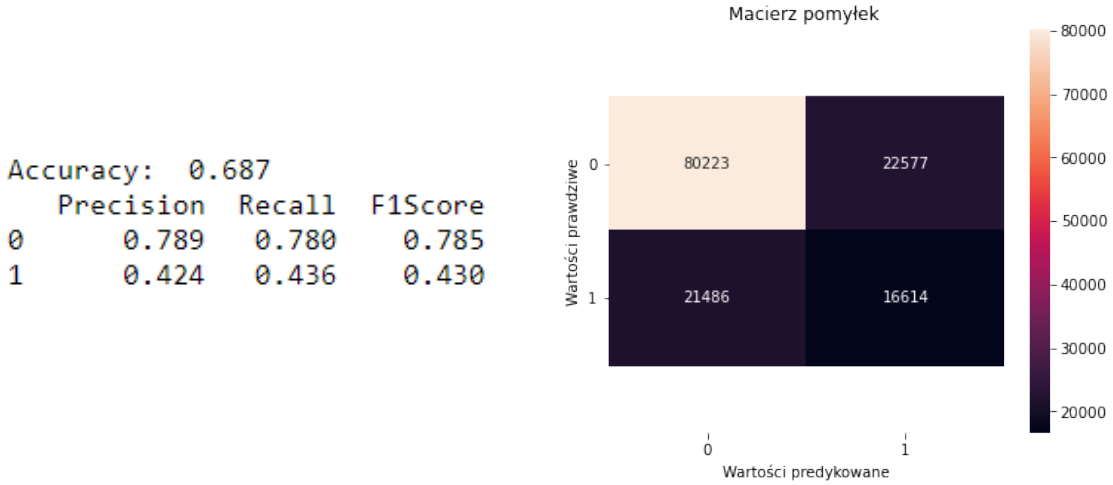



Figure 31: Model 4 - Wyniki

Wyniki uległy poprawie i są zbliżone do wyników modelu 1. W kolejnych krokach postanowiliśmy zachować zbadaną modyfikację.

### 5.2.6 Model 5

W modelu 5 usunęliśmy zmienną Z5, wyrażającą pewną zależność między PaperlessBilling i PaymentMethod.

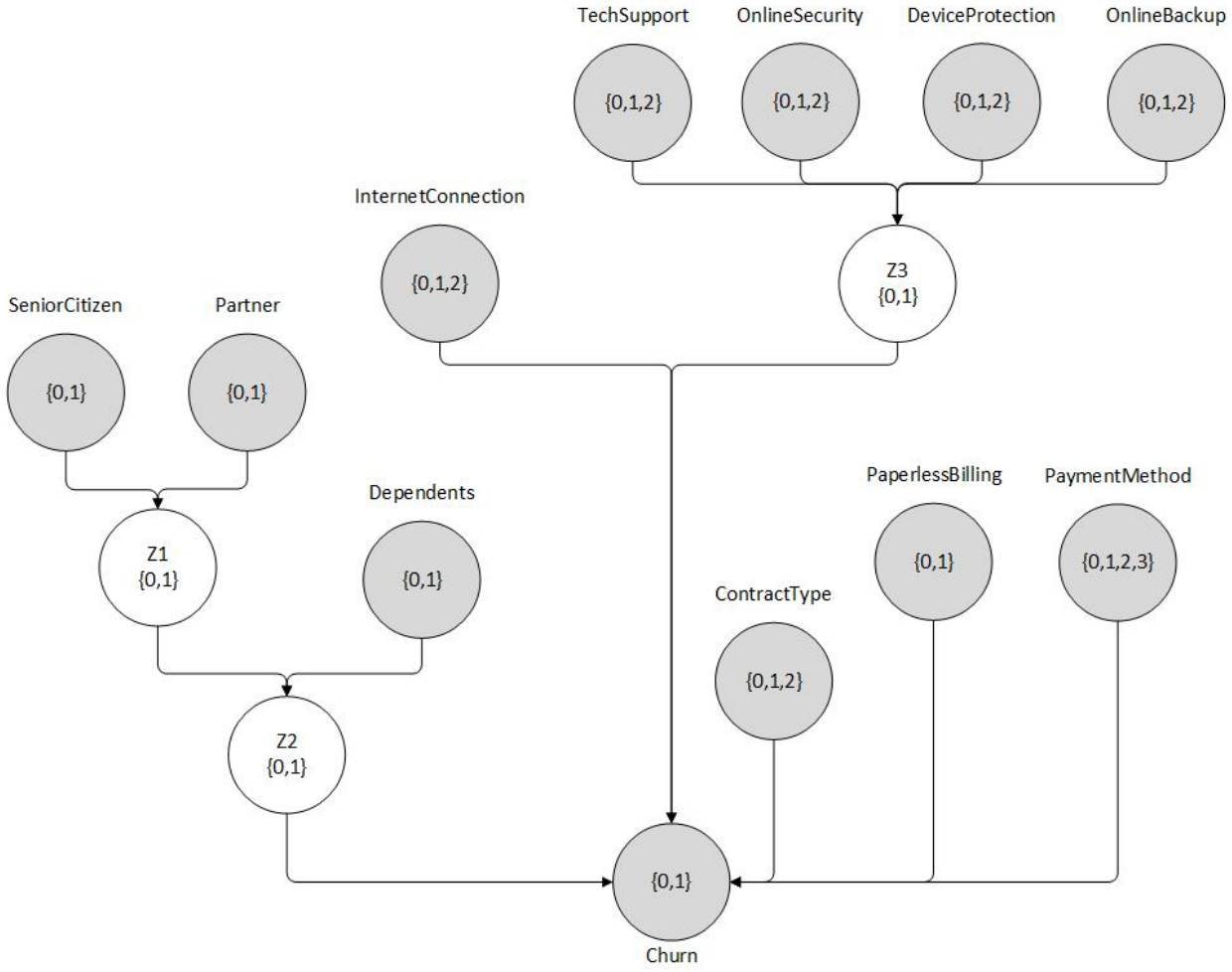



Figure 32: Model 5

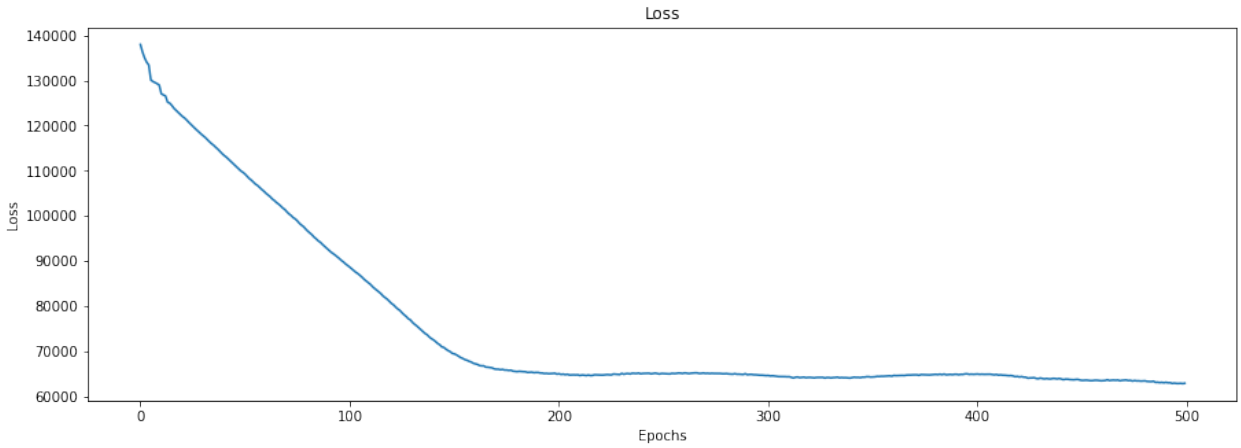



Figure 33: Model 5 - Zmiana wartości loss w trakcie uczenia

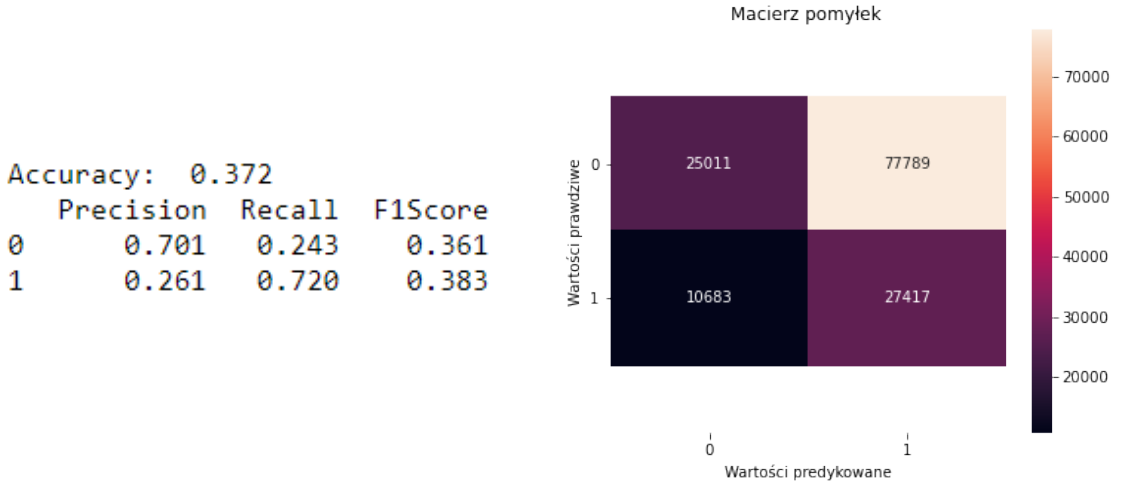



Figure 34: Model 5 - Wyniki

Wyniki znacznie się pogorszyły - ponownie model nauczył się niewiele. Nie zachowaliśmy tej modyfikacji.

### 5.2.7 Model 6

W kolejnym modelu usunęlismy zmienną Z2

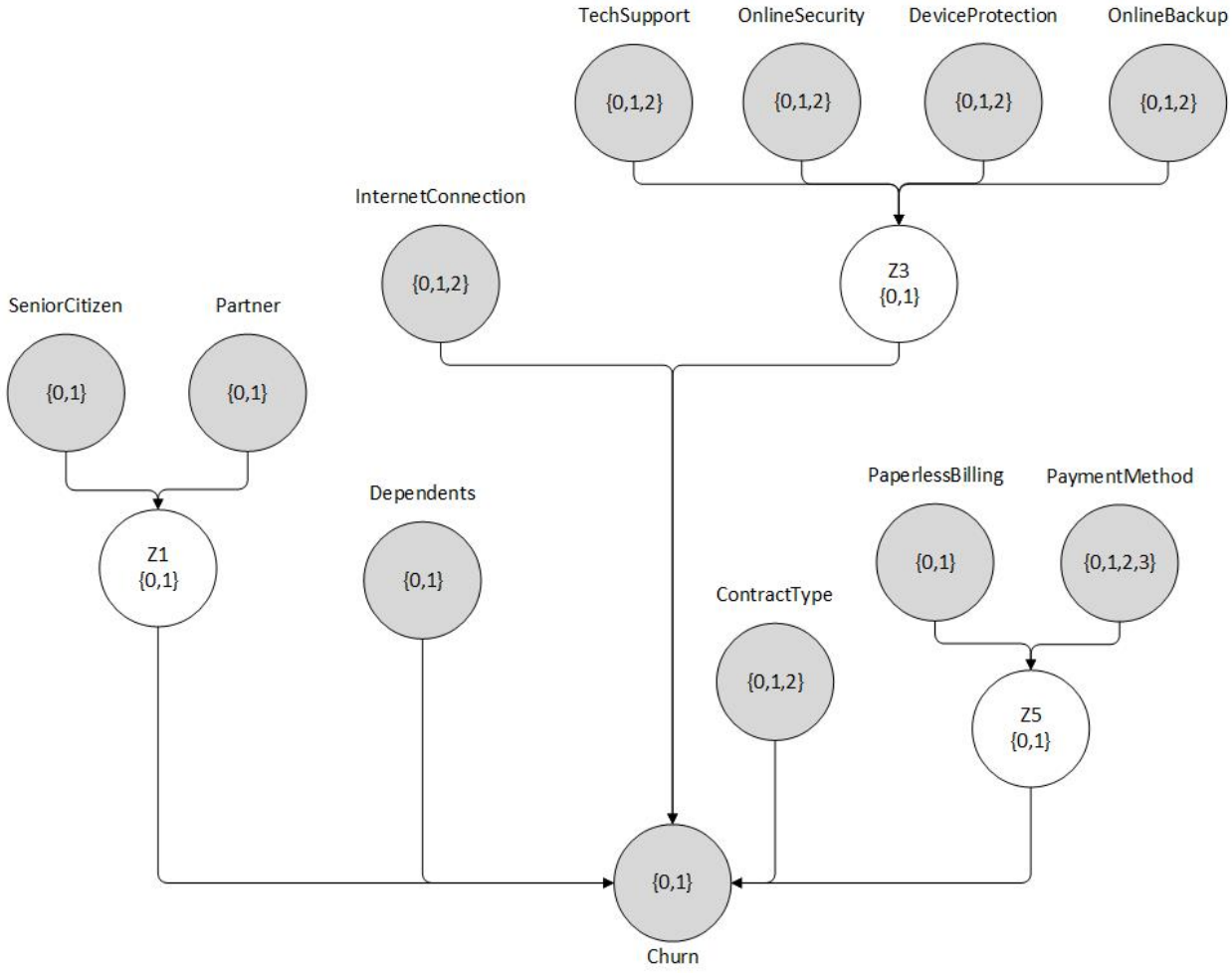



Figure 35: Model 6

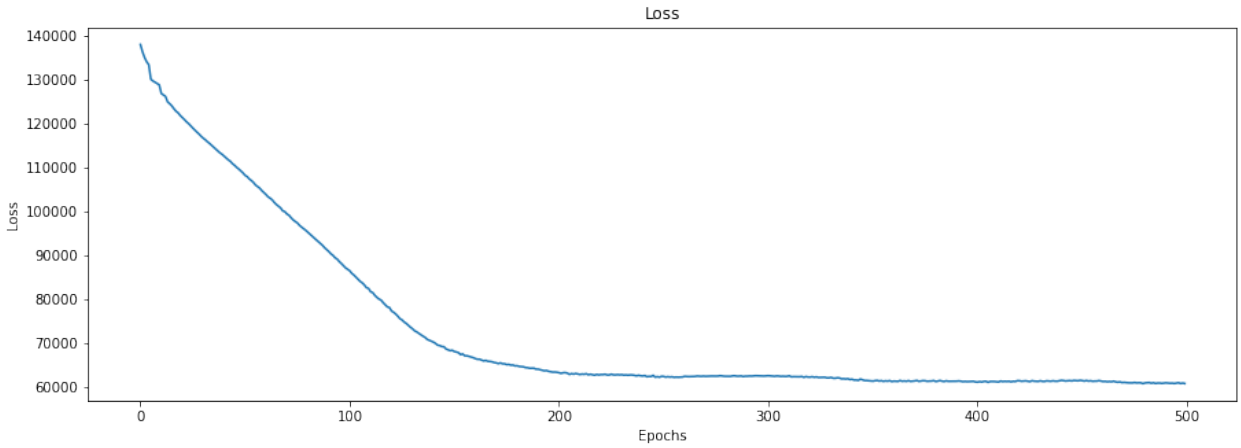



Figure 36: Model 6 - Zmiana wartości loss w trakcie uczenia

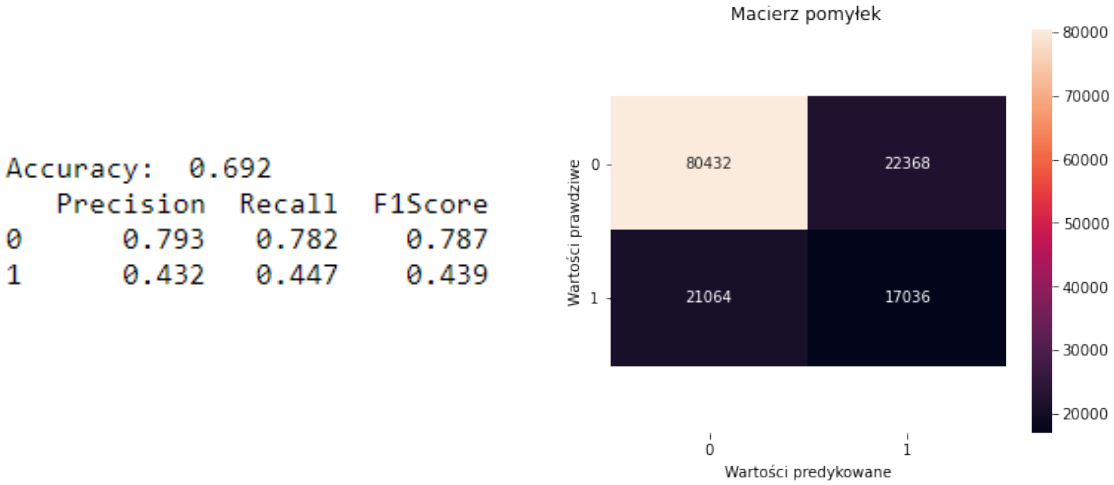



Figure 37: Model 6 - Wyniki

Wyniki uległy małej poprawie.

### 5.2.8 Model 7

W modelu 7 sprawdziliśmy usunięcie zmiennej Z2, ale także Z1.

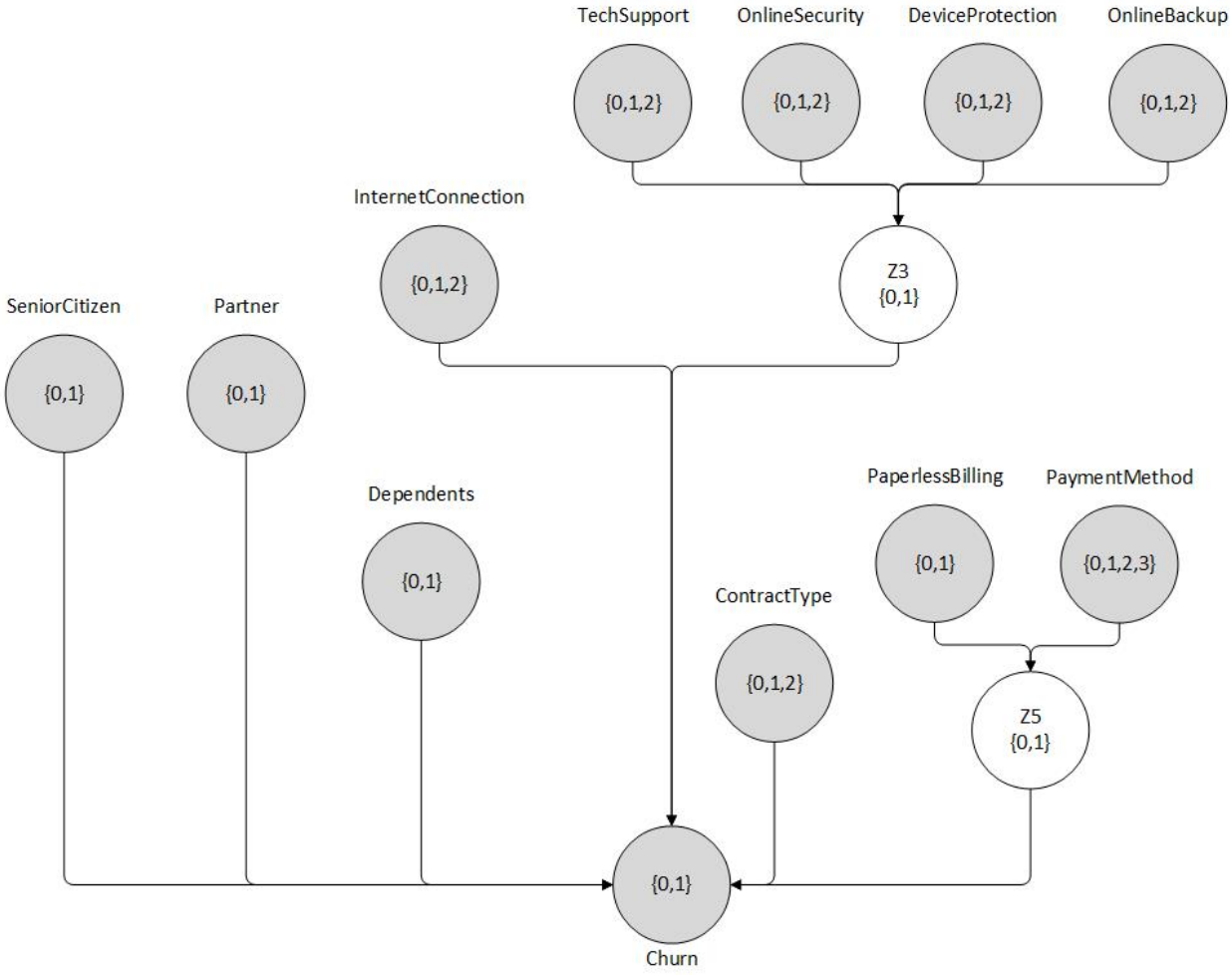



Figure 38: Model 7

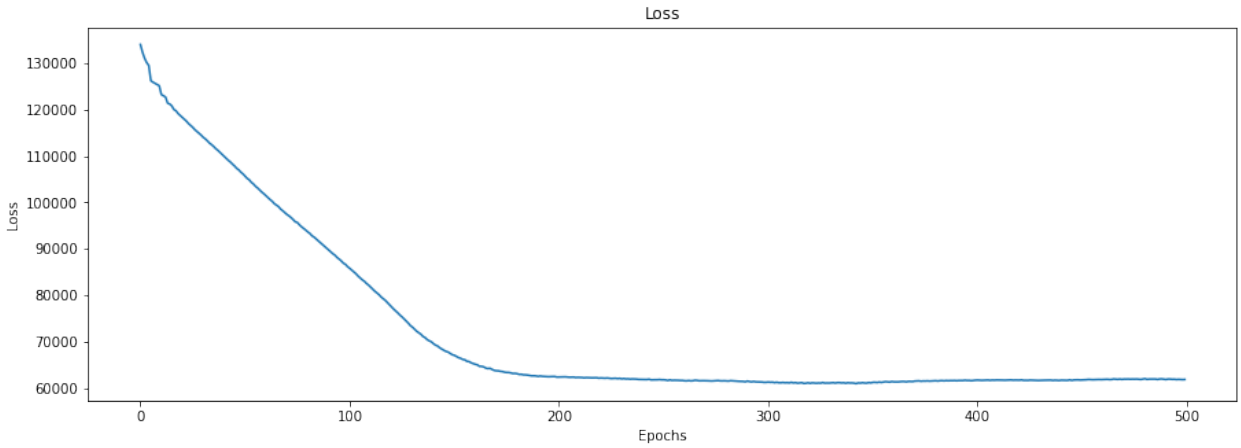



Figure 39: Model 7 - Zmiana wartości loss w trakcie uczenia

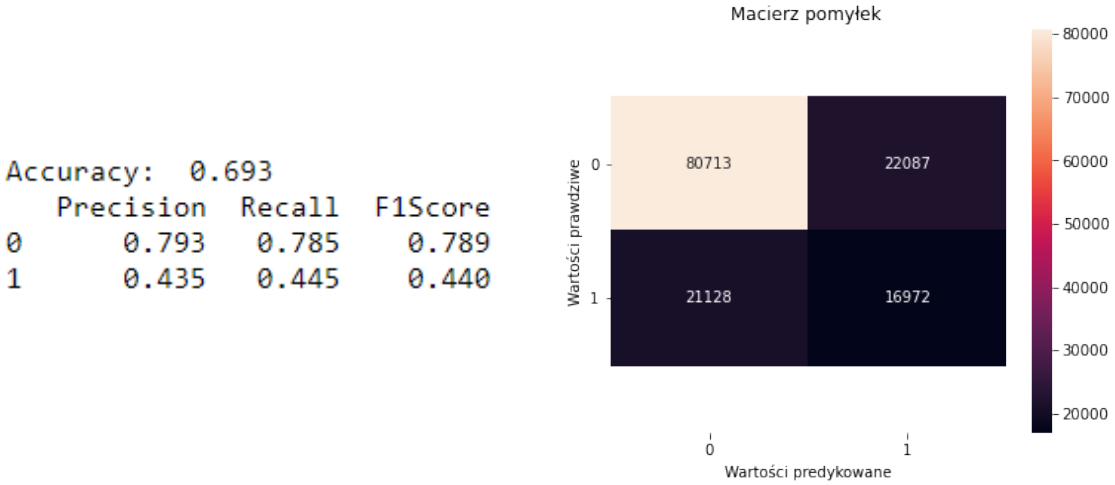



Figure 40: Model 7 - Wyniki

To również poprawiło wyniki modelu, ale wielkość poprawy uznalismy za mało znaczącą i nei zachowaliśmy modyfikacji.

### 5.2.9 Model 8

Modyfikacja w modelu 8 polegała na usunięciu ukrytej zmiennej Z3.

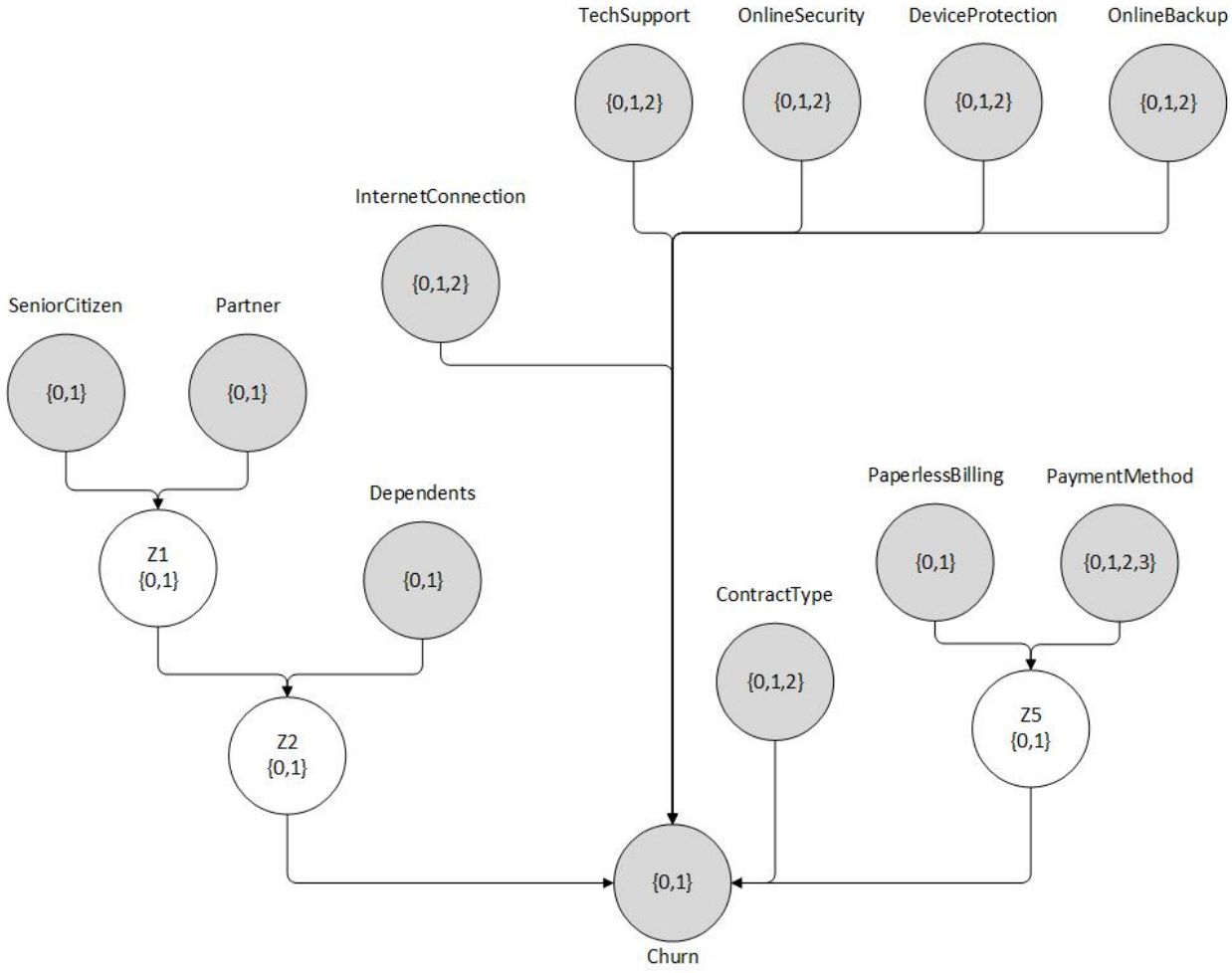



Figure 41: Model 8

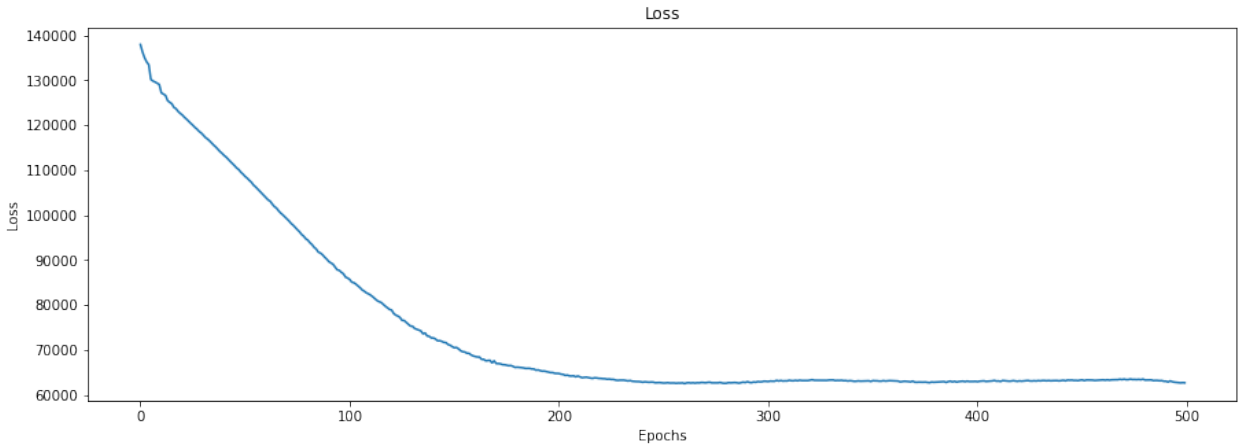



Figure 42: Model 8 - Zmiana wartości loss w trakcie uczenia

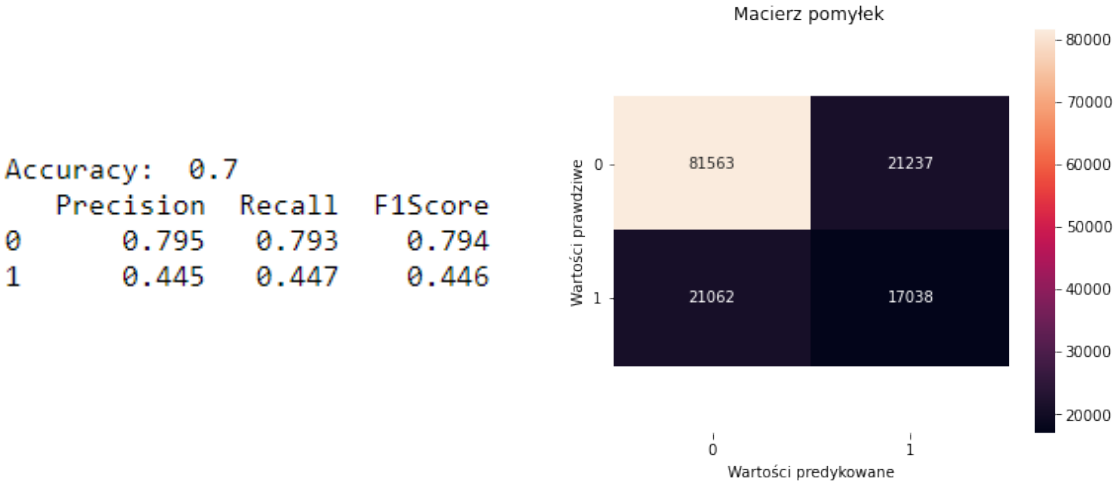



Figure 43: Model 8 - Wyniki

Tym razem uzyskaliśmy poprawę wyników, ze względu na którą, zdecydowaliśmy o zachowaniu modyfikacji.

### 5.2.10 Model 9

Dotychczas wykorzystywaliśmy tylko zmienne wejściowe, które w oryginalnym zbiorze są typu kategorycznego. Teraz dołączamy do nich również zmienne ciągłe, której poddaliśmy dyskretyzacji na 3 podobnie liczne przedziały.

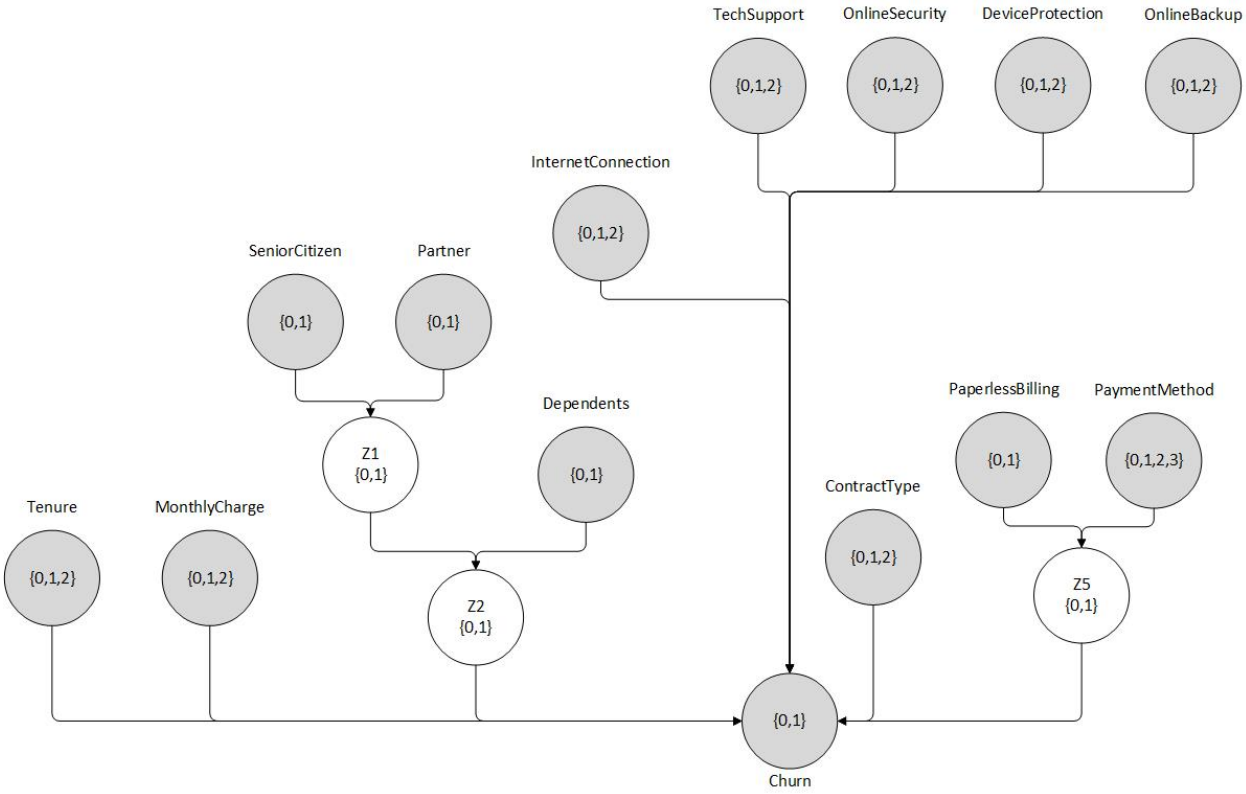



Figure 44: Model 9

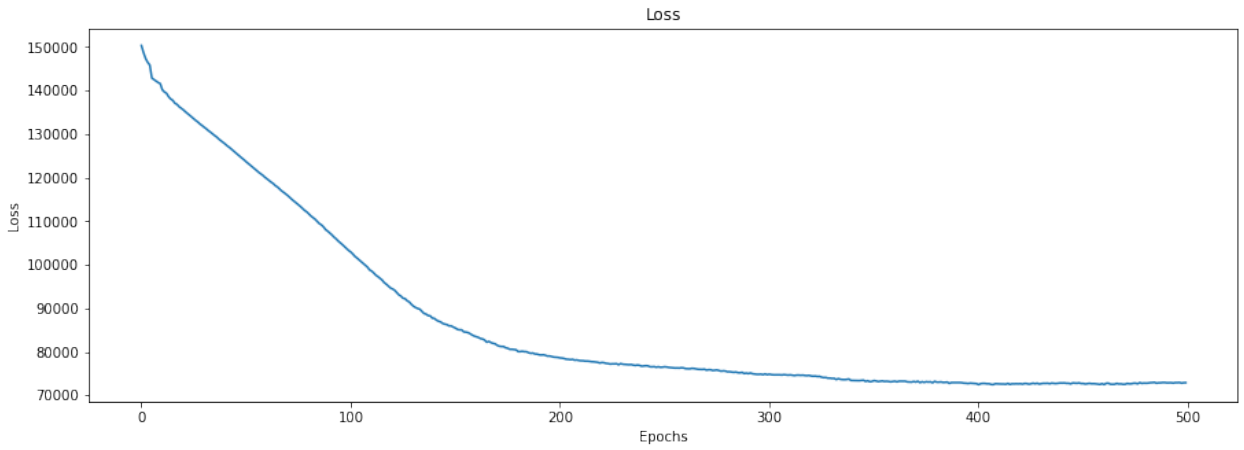



Figure 45: Model 9 - Zmiana wartości loss w trakcie uczenia

```
Accuracy:  0.699
   Precision  Recall  F1Score
0      0.803   0.779    0.791
1      0.448   0.483    0.465
```

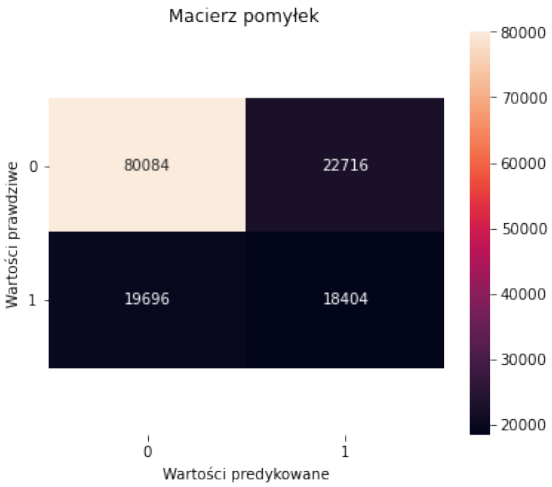



Figure 46: Model 9 - Wyniki

Otrzymując więcej wiedzy model poprawił wynik f1score.

### 5.2.11 Model 10

Sprawdziliśmy również wariant, w którym zmienne Tenure i MonthlyChange są powiązane zmienną ukrytą Z6.

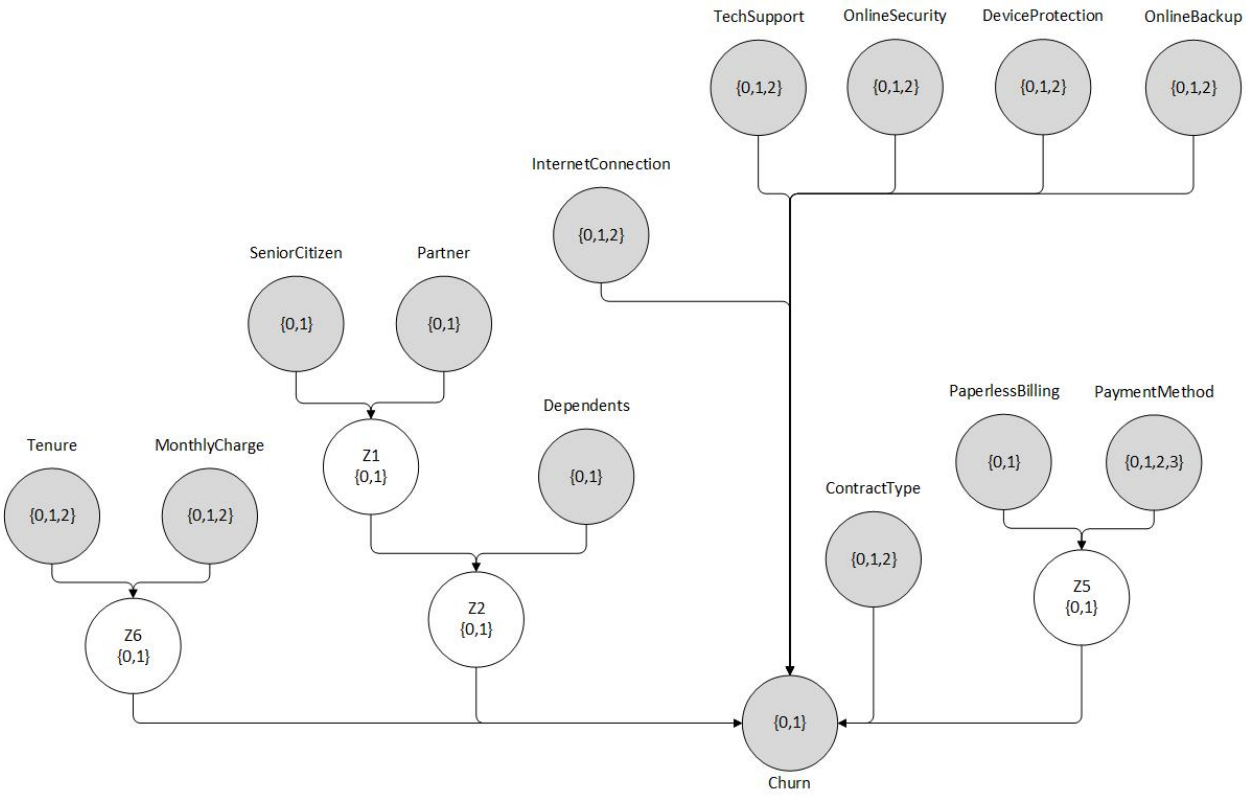



Figure 47: Model - 10

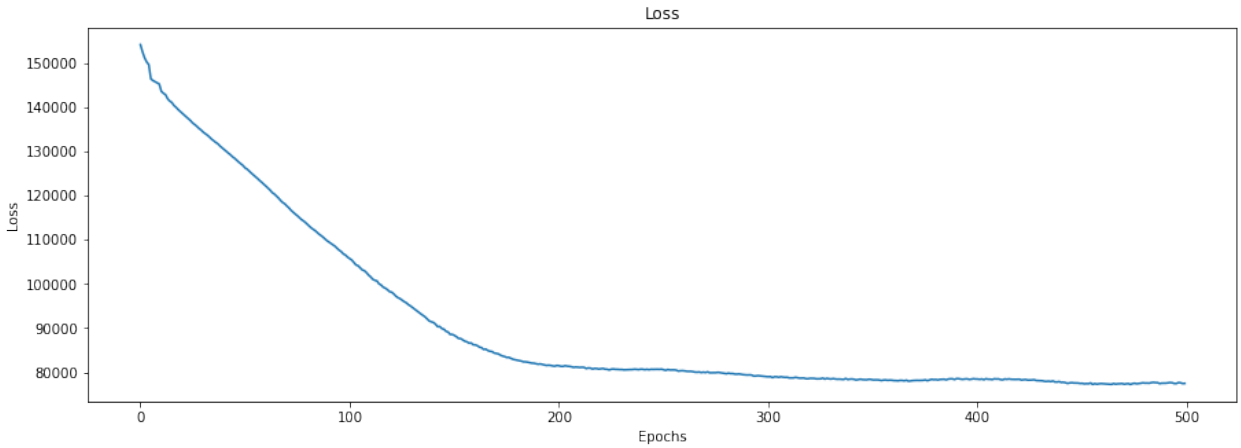



Figure 48: Model 10 - Zmiana wartości loss w trakcie uczenia

```
Accuracy:  0.701
   Precision  Recall  F1Score
0      0.795   0.796    0.795
1      0.447   0.445    0.445
```

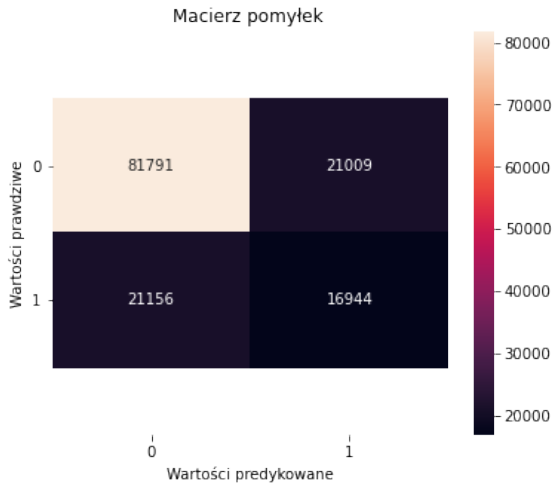



Figure 49: Model 10 - Wyniki

Wyniki pogorszyły się, jednak do kolejnego modelu postanowiliśmy zachować zmienną Z6.

### 5.2.12 Model 11

W ostatnim badanym modelu postanowiliśmo nieco uprościć jego strukturę, przez usunięcie zmiennych DeviceProtection i OnlineBackup.

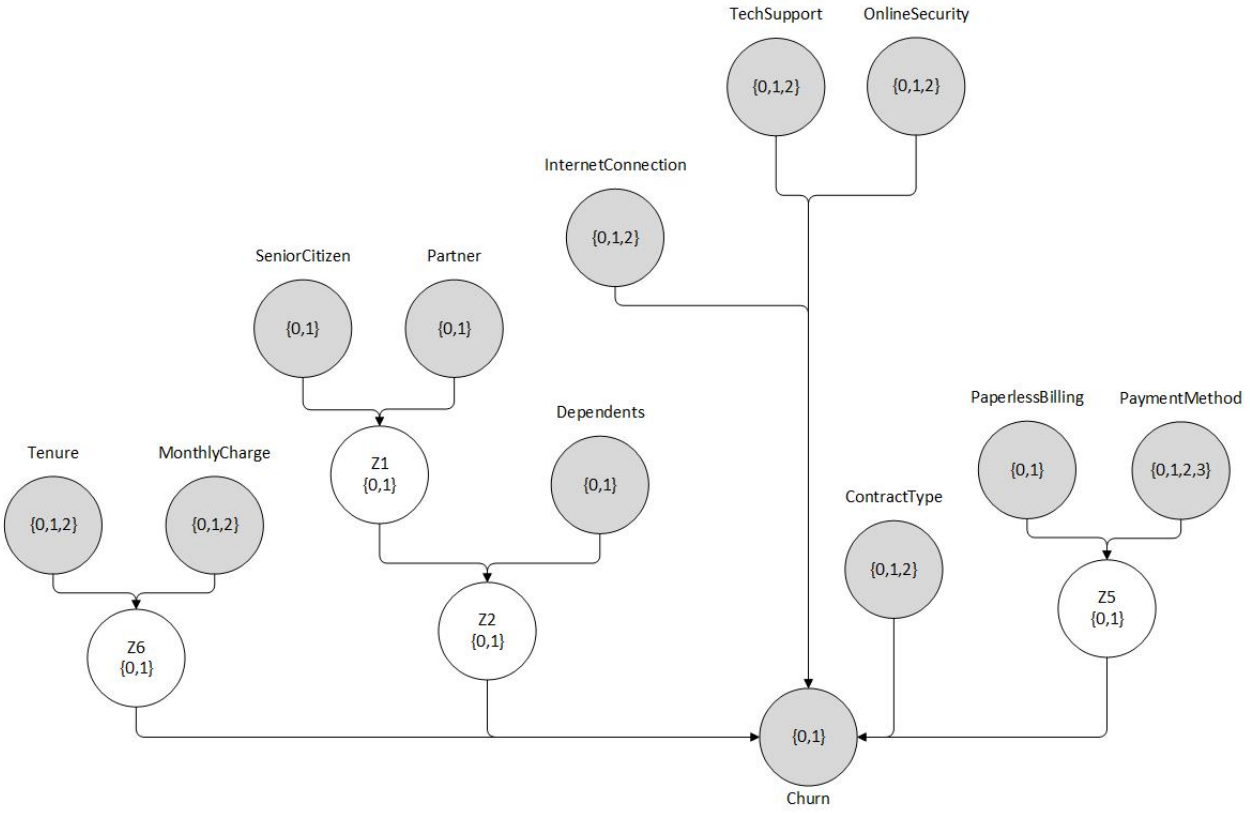



Figure 50: Model 11

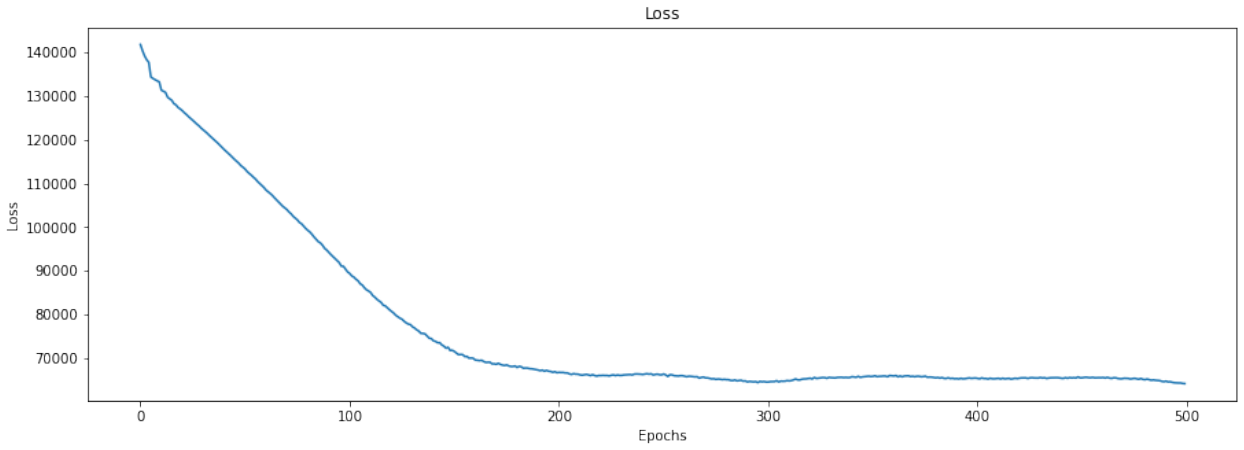



Figure 51: Model 11 - Zmiana wartości loss w trakcie uczenia

```
Accuracy:  0.693
   Precision  Recall  F1Score
0      0.796   0.779    0.788
1      0.437   0.462    0.449
```

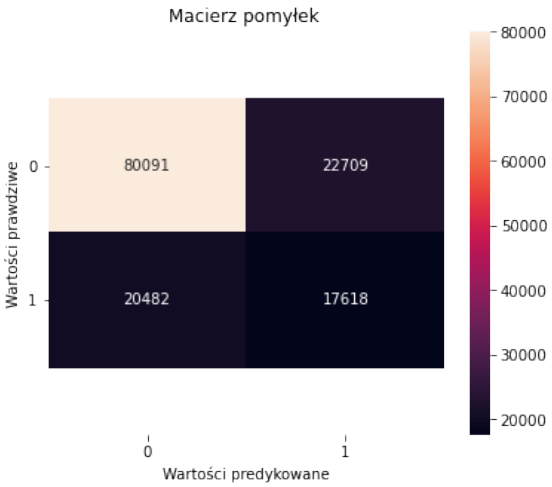



Figure 52: Model 11 - Wyniki

Wyniki uległy nieznacznej poprawie, co może oznaczać, że usunięte zmienne nie miały znaczenie dla klasyfikacji.

### 5.2.13 Dodatkowe badania

Po zbadaniu różnych struktur postanowiliśmy zbadać kilka dodatkowych aspektów używając modelu 11:

- Model pomimo dokładania nowych zmiennych mógł nadal klasyfikować tylko na podstawie InternetConnection i ContractType, tak jak model 1. Dlatego upewniliśmy, na że prawdopodobieństwo otrzymania klasy 1 mają też wpływ inne zmienne, przykładowo Z2 i Z6 (jako $x_k$ i $x_l$ oznaczyliśmy pozostałe zmienne, które w obydwu przypadkach przyjmują takie same wartości):

$$p(Churn = 1|Z2 = 0, x_k) = 0.27$$

$$p(Churn = 1|Z2 = 1, x_k) = 0.47$$

$$p(Churn = 1|Z6 = 0, x_l) = 0.10$$

$$p(Churn = 1|Z6 = 1, x_l) = 0.18$$

- Zbadaliśmy również czy modyfikacja zmiennych ukrytych w modelu 11 tak, aby mogły przyjmować 3 różne wartości, będzie miała wpływ na wyniki. Jak się okazało, wyniki uległy nieznacznej zmianie.

Accuracy: 0.702

| | Precision | Recall | F1Score |
|---|---|---|---|
| 0 | 0.796 | 0.796 | 0.796 |
| 1 | 0.449 | 0.448 | 0.449 |

- W kolejnym kroku zbadaliśmy wpływ początkowych wartości prawdopodobieństw dla Churn równego 1. Ustawiliśmy je na 0 i otrzymaliśmy model, który niczego się nie nauczył. Może wynikać to z niezbalansowania zbioru. Modelowi było łatwiej najpierw założyć, że Churn zawsze jest równe 1, a potem zaobserwować bardzo dużo przypadków, gdzie Churn równa się 0 i na ich podstawie zoptymalizować parametry.

Accuracy: 0.73

| | Precision | Recall | F1Score |
|---|---|---|---|
| 0 | 0.73 | 1.0 | 0.844 |
| 1 | 0.00 | 0.0 | 0.000 |

- Sprawdziliśmy również, czy sieć na pewno wydobywa wiedzę z danych, a nie działa jak klasyfikator losowy odwzorowujący dystrybucję klas. Dlatego usunęliśmy losowo część obserwacji z Churn równym 0, tak aby otrzymać idealnie zbalansowany zbiór i z takiego utworzyliśmy zbiór testowy i treningowy. Na zbalansowanym zbiorze uzyskaliśmy accuracy większe o 0.5, więc nasze modele faktycznie się uczą i nie całkowicie losowej klasyfikacji.

Accuracy: 0.673

| | Precision | Recall | F1Score |
|---|---|---|---|
| 0 | 0.662 | 0.691 | 0.676 |
| 1 | 0.686 | 0.656 | 0.670 |

- Dodatkowo przyjrzeliśmy się także dyskretyzacji zmiennych ciągłych. Przeprowadziliśmy badanie dla dyskretyzacji na 5 przedziałów, jednak nie wpłynęło to znacząco na wyniki.

```
Accuracy:  0.704
   Precision  Recall  F1Score
0      0.796   0.798    0.797
1      0.452   0.448    0.449
```

#### 5.2.14 Wnioski

- Sieć o zaproponowanej strukturze nie dała wyjątkowo dobrych efektów i nie dała dużej poprawy względem najprostszego zbadanego modelu.
- Badania warto zaczynać od najprostszych modeli, bo mogą one dawać dobre wyniki.
- Zbudowane modele warto próbować upraszczać.
- Warto stosować zmienne ukryte, nawet gdy nie są one zgodne z intuicją.
- Intuicyjne szukanie optymalnej struktury sieci zajmuje dużo czasu i może być nieefektywne. Warto byłoby zastosować do tego bardziej zaawansowane metody.
- W kontekście jakości klasyfikacji ważny jest wybór odpowiedniej początkowej inicjalizacji prawdopodobieństw.

# 6 Podsumowanie badań i wnioski

1. Podczas prowadzenia badań nad modelami uczenia maszynowego dobrze jest poznać techniki wizualizacji danych.
2. Oprócz pracy nad samym modelem należy także stworzyć sobie zestaw narzędzi do walidacji modelu takie jak metryki, walidacja krzyżowa, wykresy.
3. Także należy pamiętać o zintegrowaniu ich ze środowiskiem w którym będzie pisana potencjalna publikacja. Niestety za późno się z orientowaliśmy że nie widać wszystkich etykiet w texie.
4. W przyszłości na pewno bardziej zautomatyzujemy proces otrzymywania wyników i umieszczania ich w raporcie, co pozwoli szybko naprawiać takie błędy.
5. Niezbalansowane zbiory danych mogą utrudniać zbudowanie dobrego klasyfikatora.
6. Zawsze warto wypróbować najprostsze modele, jak np. NaiveBayes, ponieważ mogą one dawać dobre wyniki.
7. Uzyskane przez nas wyniki nie należą do najlepszych (dla tego problemu można uzyskać wynik fscore nawet 0.7), jednak pokazały, że zbudowaliśmy poprawnie uczące się modele.
8. Podczas projektu uporządkowaliśmy i pogłębiliśmy wiedzę o modelach probabilistycznych oraz mieliśmy okazję wykorzystać je samodzielnie w praktyczny sposób.

## References


[1] K. Murphy. *Machine Learning A Probabilistic Perspective, chapter 10*.

[2] Pyro. Inference with discrete latent variables, . URL https://pyro.ai/examples/enumeration.html.

[3] Pyro. Svi, . URL http://docs.pyro.ai/en/0.2.1-release/inference_algos.html.