Bank Marketing data (with social/economic context)

Maciej Małecki

Streszczenie

Plik "Bank Marketing data.csvźawiera dane charakteryzujące klientów pewnego banku oraz kampanie marketingowe skierowane do tych klientów. Dołączone są ponadto wskaźniki społeczne i ekonomiczne. Na podstawie tych danych należy zbudować model prognozujący szansę, że klient w wyniku prowadzonej kampanii założy lokatę terminową.

Spis treści

1 Wprowadzenie

1.1 Opis problemu

W ramach kampani marketingowej organizowanej przez pewien bank w latach między majem 2008 rok, a listopadem 2010 roku, były zbierane informacje na temat klientów tego banku. Na podstawie tych danych planowane jest przewidzenie, czy i jakie rodzaj klientów kupi lokatę terminową w tym banku.

1.2 Opis danych

Nasze dane zawierają 21 column danych. Kolumny możemy podzielić na 3 grupy:

I: Zmienne związane z danymi klienta bankowego:

1. Wiek (age): wiek klienta.
2. Praca (job): rodzaj pracy klienta.
3. Stan cywilny (marital): stan cywilny klienta.
4. Edukacja (education): edukacja klienta.
5. Domyślnie (default): Klient wcześniej domyślnie miał kredyt.
6. Mieszkanie (housing): Klient ma kredyt mieszkaniowy.
7. Pożyczka (loan): Klient ma osobistą pożyczkę.

II: Zmienne związane z ostatnim kontaktem bieżącej kampanii marketingowej:

8. Kontakt (contact): Typ komunikacji kontaktowej (telefonicznej lub komórkowej).
9. Miesiąc (month): Ostatni kontakt miesiąca roku.
10. Dzień tygodnia (day of week): dzień ostatniego kontaktu tygodnia.
11. Czas trwania (duration): czas trwania ostatniego kontaktu w sekundach. Jeśli czas trwania wynosi 0, nigdy nie skontaktowaliśmy się z klientem, aby założyć konto lokaty terminowej.
12. Kampania (campaign): liczba kontaktów wykonanych podczas tej kampanii i dla tego klienta
13. Liczba dni (pdays): liczba dni, które upłynęły od ostatniego kontaktu klienta z poprzedniej kampanii (wartość liczbowa; 999 oznacza, że klient wcześniej się nie skontaktował)
14. Poprzedni (previous): liczba kontaktów wykonanych przed tą kampanią i dla tego klienta (numerycznie)
15. Poutcome: wynik poprzedniej kampanii marketingowej (kategorycznie: „porażka”, „nieistniejąca”, „sukces”)

III: Atrybuty kontekstu społecznego i gospodarczego:

16. Emp.var.rate: wskaźnik zmienności zatrudnienia - wskaźnik kwartalny
17. Cons.price.idx: wskaźnik cen konsumpcyjnych - wskaźnik miesięczny

18. Cons.conf.idx: wskaźnik zaufania konsumentów - wskaźnik miesięczny
19. Euribor3m: stawka 3-miesięczna euribor - wskaźnik dzienny
20. Liczba zatrudnionych (nr employed): liczba pracowników - wskaźnik kwartalny

Zmienna wyjściowa (pożądany cel):

21. y - czy klient subskrybował lokatę? (dwójkowy: „tak”, „nie”)

1.3 Wstępna eksploracja danych

Badane dane zawierają 4119 wierszy oraz 21 kolumn w czym 11 kolumn ze zmiennymi `character` oraz 10 kolumn ze zmiennymi `numeric`. Samą struktórę danych możemy zobaczyć w tabeli **??**.

```
str(df_bank)

## 'data.frame': 4119 obs. of  21 variables:
##  $ age           : int  30 39 25 38 47 32 32 41 31 35 ...
##  $ job           : chr  "blue-collar" "services" "services" "services" ...
##  $ marital       : chr  "married" "single" "married" "married" ...
##  $ education     : chr  "basic.9y" "high.school" "high.school" "basic.9y" ...
##  $ default       : chr  "no" "no" "no" "no" ...
##  $ housing       : chr  "yes" "no" "yes" "unknown" ...
##  $ loan          : chr  "no" "no" "no" "unknown" ...
##  $ contact       : chr  "cellular" "telephone" "telephone" "telephone" ...
##  $ month         : chr  "may" "may" "jun" "jun" ...
##  $ day_of_week   : chr  "fri" "fri" "wed" "fri" ...
##  $ duration      : int  487 346 227 17 58 128 290 44 68 170 ...
##  $ campaign      : int  2 4 1 3 1 3 4 2 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 2 0 0 1 0 ...
##  $ poutcome      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
##  $ emp.var.rate  : num  -1.8 1.1 1.4 1.4 -0.1 -1.1 -1.1 -0.1 -0.1 1.1 ...
##  $ cons.price.idx: num  92.9 94 94.5 94.5 93.2 ...
##  $ cons.conf.idx : num  -46.2 -36.4 -41.8 -41.8 -42 -37.5 -37.5 -42 -42 -36.4 ...
##  $ euribor3m     : num  1.31 4.86 4.96 4.96 4.19 ...
##  $ nr.employed   : num  5099 5191 5228 5228 5196 ...
##  $ y             : chr  "no" "no" "no" "no" ...
```

Rysunek 1: Struktura danych wraz z przykładowymi wartościami

W danych nie znajdziemy informacji o danych nieznanych typu `NaN` oraz `Na`. Jednakże wiemy, że w danych występują wartości brakujące i są one opisane "unknown". W danych znajduje się 1230 rekordów z wartością "unknownżozmieszczonych w 1029 różnych wierszach. To stanowi 24.98% wszystkich wierszy w naszej badzie danych, więc nie możemy pozwolić sobie na usunięcie tych wszystkich informacji. W tabeli 1 znajdują się informacje na temat liczny nieznanych wartości, w każdej z kolumn z osobna.

Feature	Number_of_unknown
default	803
education	167
housing	105
loan	105
job	39
marital	11
age	0
contact	0
month	0
day_of_week	0
duration	0
campaign	0
pdays	0
previous	0
poutcome	0
emp.var.rate	0
cons.price.idx	0
cons.conf.idx	0
euribor3m	0
nr.employed	0
y	0

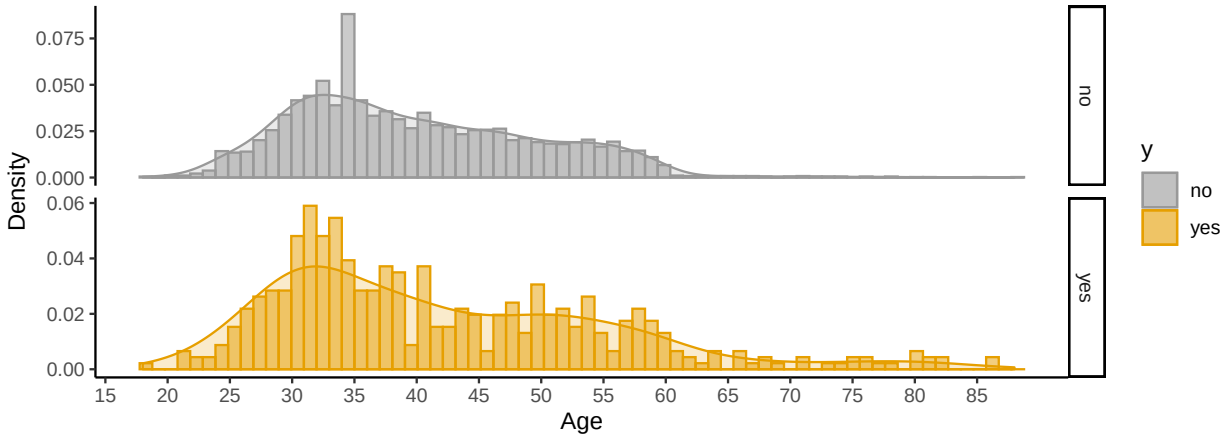
Tablica 1: Ilość wartości 'unknown' w poszczególnych kolumnach.

2 Analiza eksploracyjna

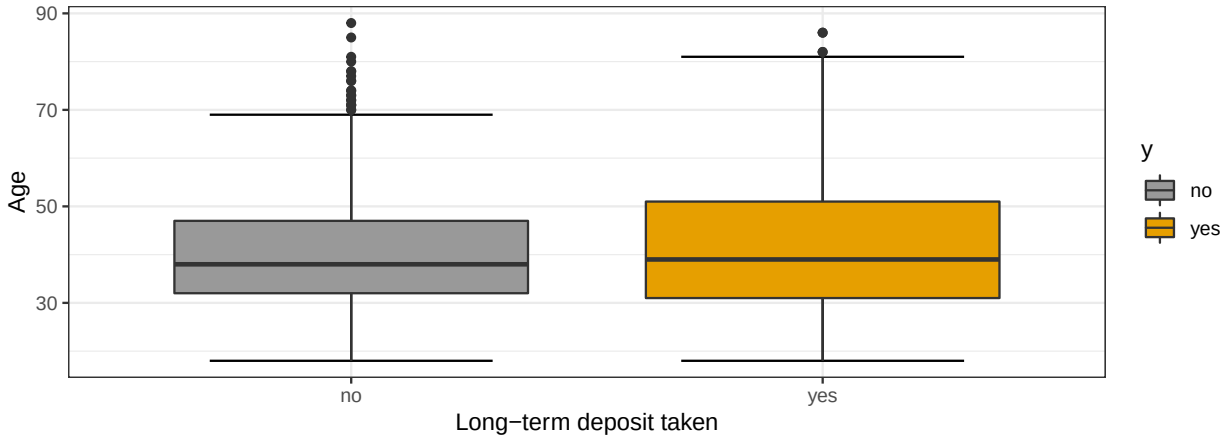
W tej sekcji zostanie omówiony każdy parametr z osobna. Następnie dane zostaną odpowiednio przygotowane do wykorzystania ich w modelach predykcyjnych.

2.1 Age

Przedział wiekowy osób biorących kredyt szacuje się miedzy 18 rokiem życia, a 88 rokiem życia. Jednakże można zauważyć, że osoby które ukończyły 60 rok życia z większą chęcią brały lokaty, niż tego nie robiły. Średni wiek utrzymuje się na poziomie 40 lat. Wiedząc, że osoby odkładają na lokaty fundusze wtedy, kiedy dobrze zaczynają zarabiać to podzieliłbym ludzi ze względu na wiek. Między wiekiem [MIN, 30] <- young, [30,65] <- worker, [65, MAX] <- pensioner. Taki podział powinien ułatwić analizę przyszłych algorytmów.



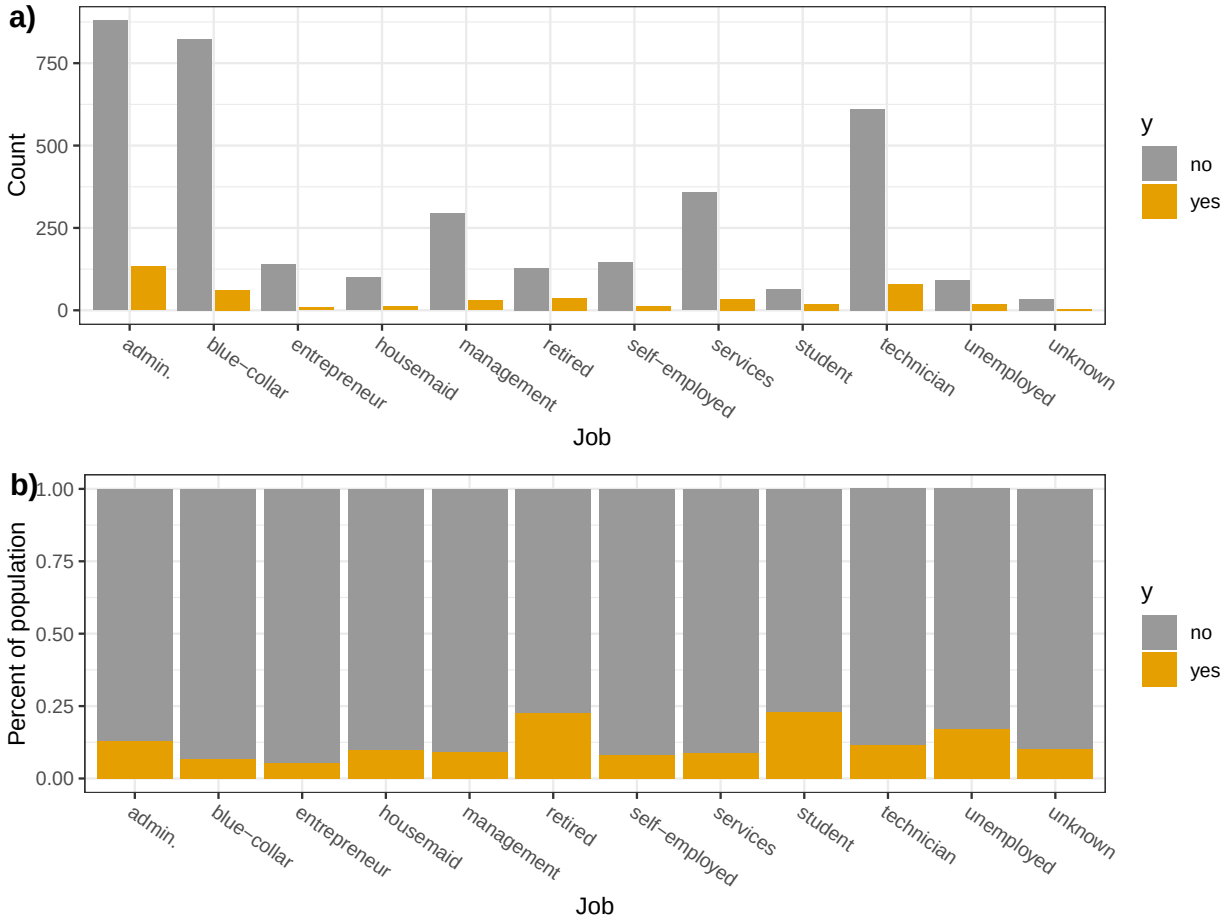
Rysunek 2: Histogram wieku klientów w zależności od wzięcia lokaty długoterminowej.



Rysunek 3: Boxplot wieku klientów w zależności od wzięcia lokaty długoterminowej.

2.2 Job

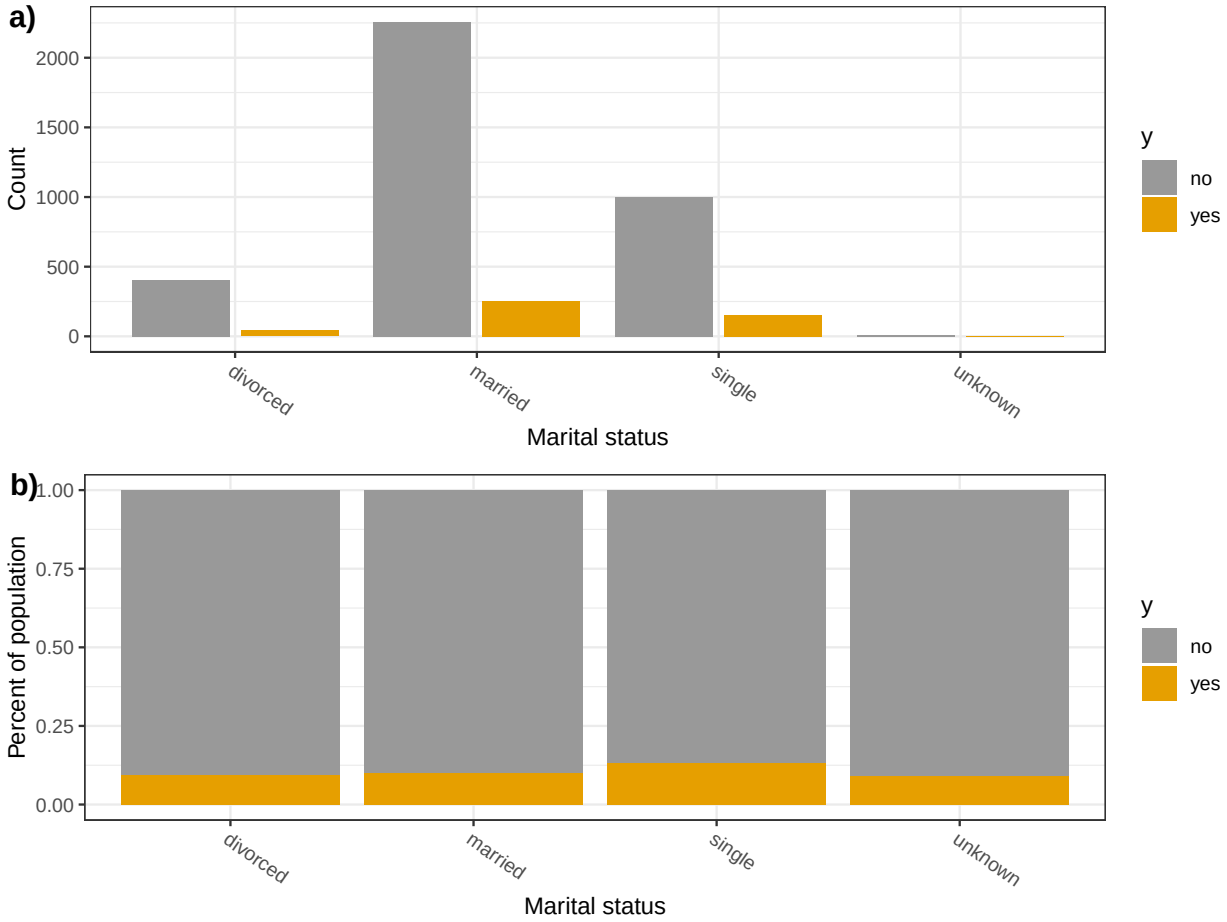
W tej kolumnie mamy 39 wartości nieznanych, co stanowi ledwo 1% całego zbioru, więc pozbywamy się wierszy, które zawierają tę informację.



Rysunek 4: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od zawodu, b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od pracy.

2.3 Marital status

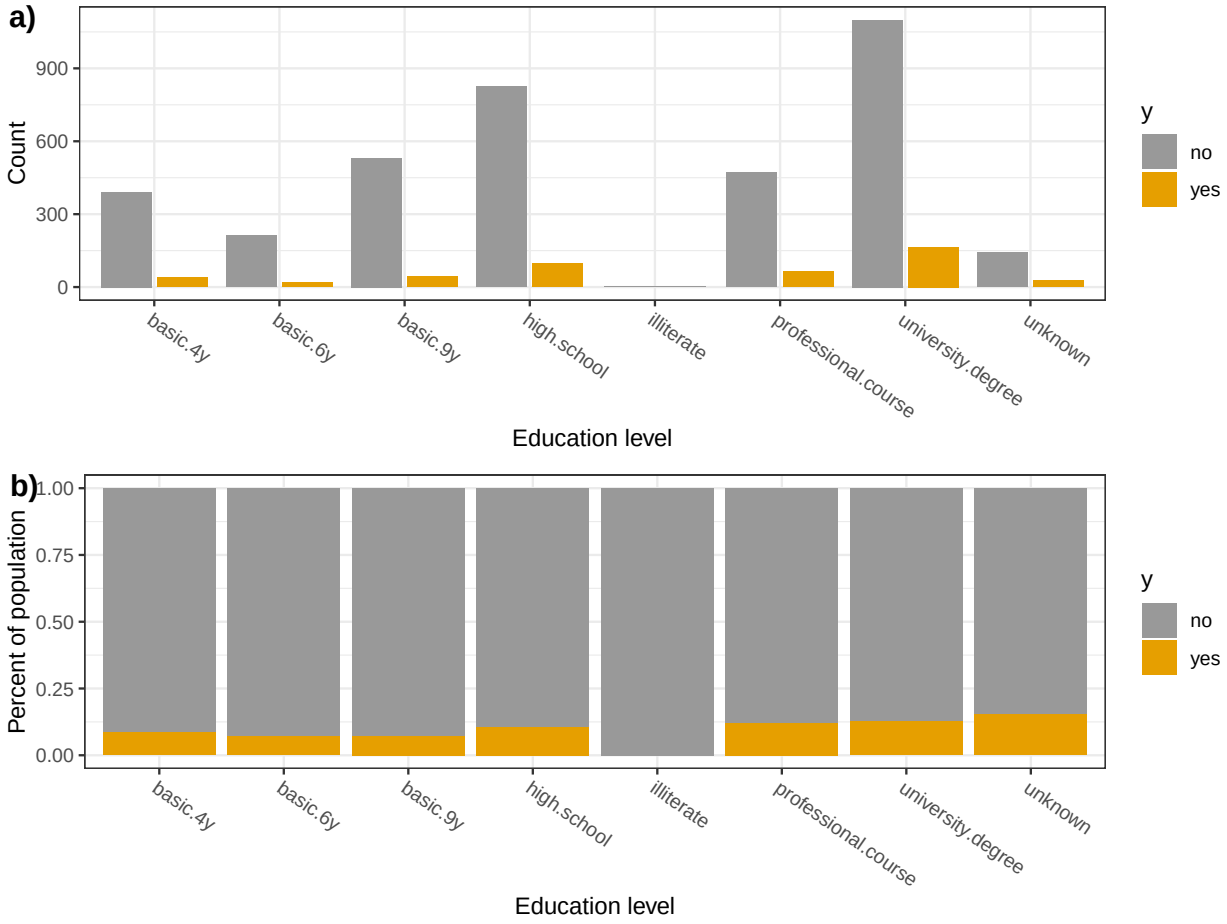
Sytuacja taka sama jak przy kolumnie 'job'. Mamy tutaj nieznane wartości, ale stanowią one tylko 0.3% wszystkich danych, więc również usuwamy te wiersze.



Rysunek 5: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od stanu cywilnego; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od stanu cywilnego.

2.4 Education

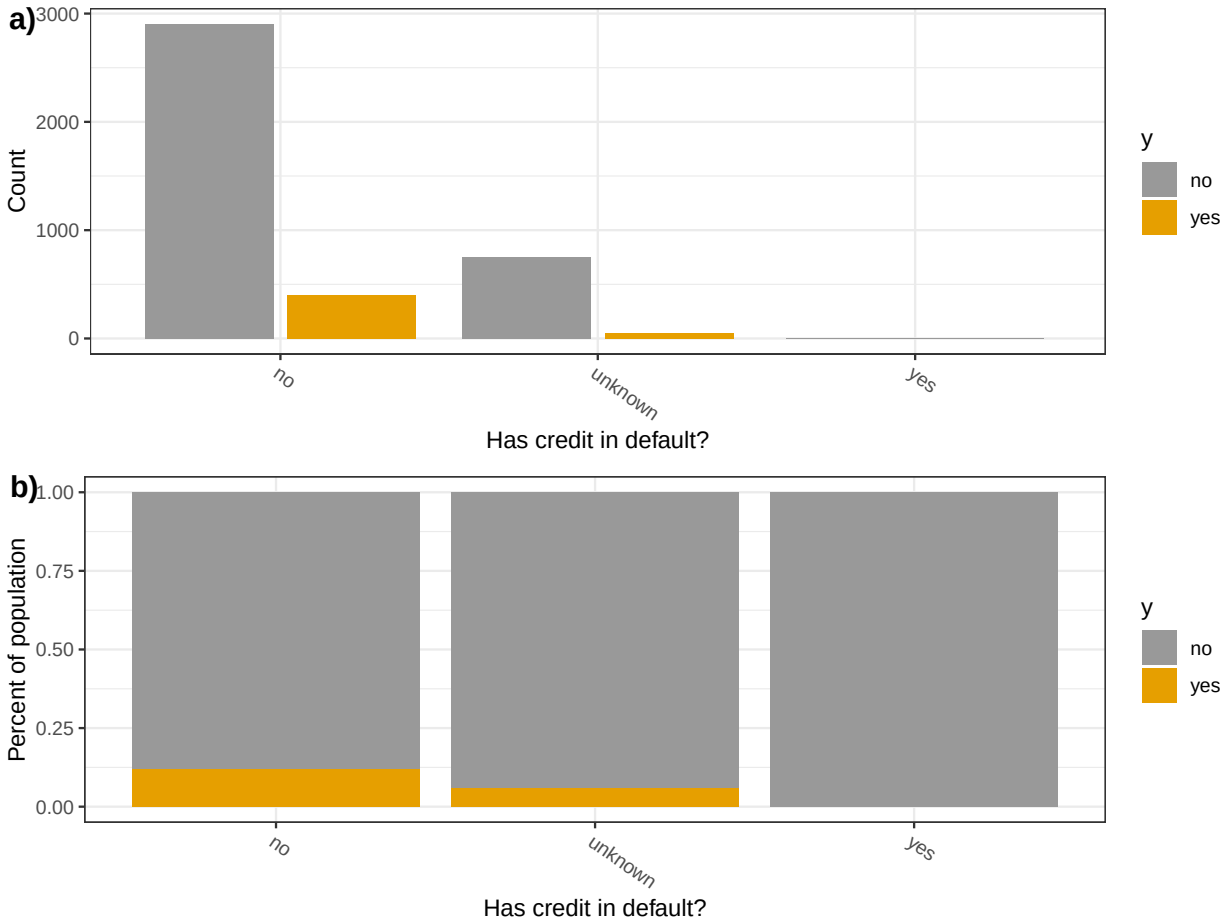
Pula klientów będącymi 'illiterate' zawiera tylko jedną osobę, więc statystycznie taka obserwacja nic nam nie daje. Natomiast w tym przypadku mamy problem z nieznanymi wartościami. Po pierwsze stanowią one 4.1% wszystkich badanych. Najbardziej podobne proporcje danych między 'yes' i 'no' ma kategoria klientów, którzy ukończyli uniwersytet, więc wszystkich klientów 'unknown' dodam do tej puli klientów.



Rysunek 6: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od poziomu wykształcenia; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od poziomu wykształcenia.

2.5 Has credit in default?

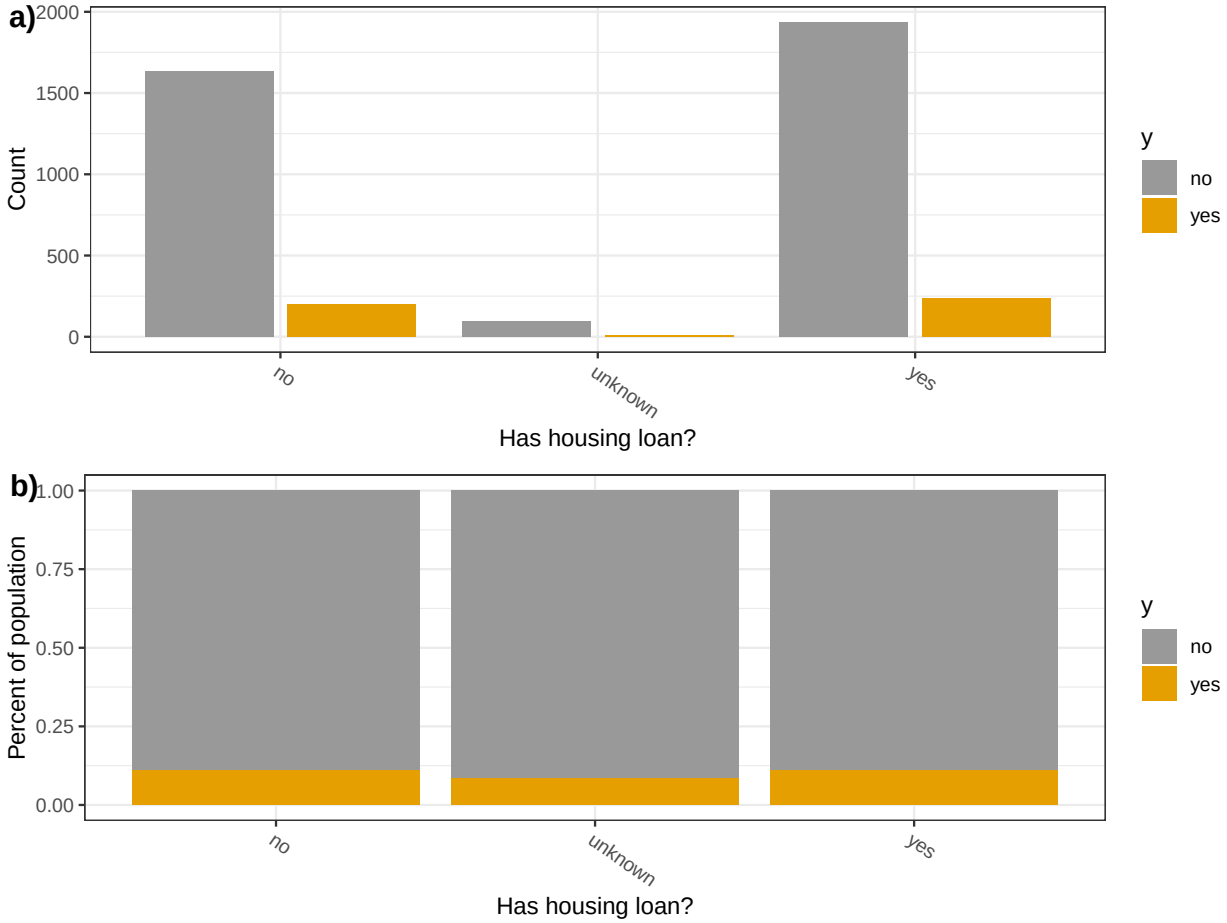
Ta kolumna nie przedstawia wystarczającej ilości danych o osobach, które domyślnie wzięły ten kredyt. Z tego powodu ta kolumna nie będzie miała żadnego większego wpływu na nasze modele, dlatego ją usuwamy.



Rysunek 7: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od posiadania kredytu; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od posiadania kredytu.

2.6 Has housing loan?

W tej kolumnie znajduje się informacje na temat posiadania kredytu hipotecznego (kredytu na dom). Ilość danych nieznanych odpowiada, $2,5\%$ wszystkich obserwacji, Nie możemy pozwoli? sobie na usuniecie tak du?ej liczny wierszy, a podłączenie do jakie? innej opcji nie wchodzi w gr?. Przeprowadźmy testy na niezależność zmiennych kategorycznych. Wykonamy test chisq w celu zbadania niezależności między 2 zmiennymi.

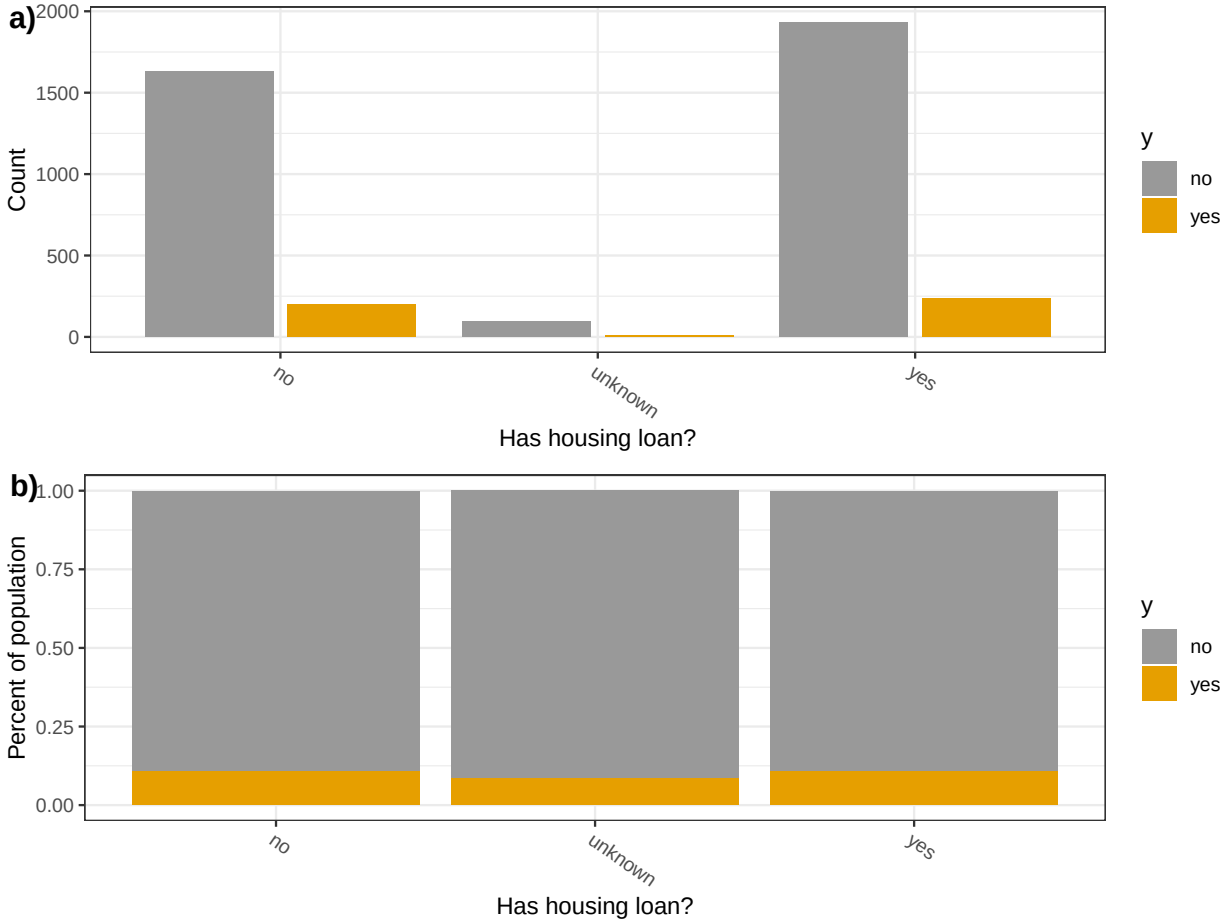


Rysunek 8: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od posiadania kredytu hipotecznego; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od posiadania kredytu hipotecznego.

Niestety poziom istotności(p-value) na poziomie 73% świadczy o dużej zależności między danymi, więc tej zmiennej r??nie? nie b

2.7 Has personal loan?

W tej kolumnie znajdują się informacje na temat posiadania kredytu hipotecznego (kredytu na dom). Ilość danych nieznanych odpowiada, 2,5% wszystkich obserwacji. Nie możemy pozwolić sobie na usunięcie tak dużej liczny wierszy, a podłączenie do jakieś innej opcji nie wchodzi w grę. Przeprowadźmy testy na niezależność zmiennych kategorycznych. Wykonamy test chisq w celu zbadania niezależności między 2 zmiennymi.

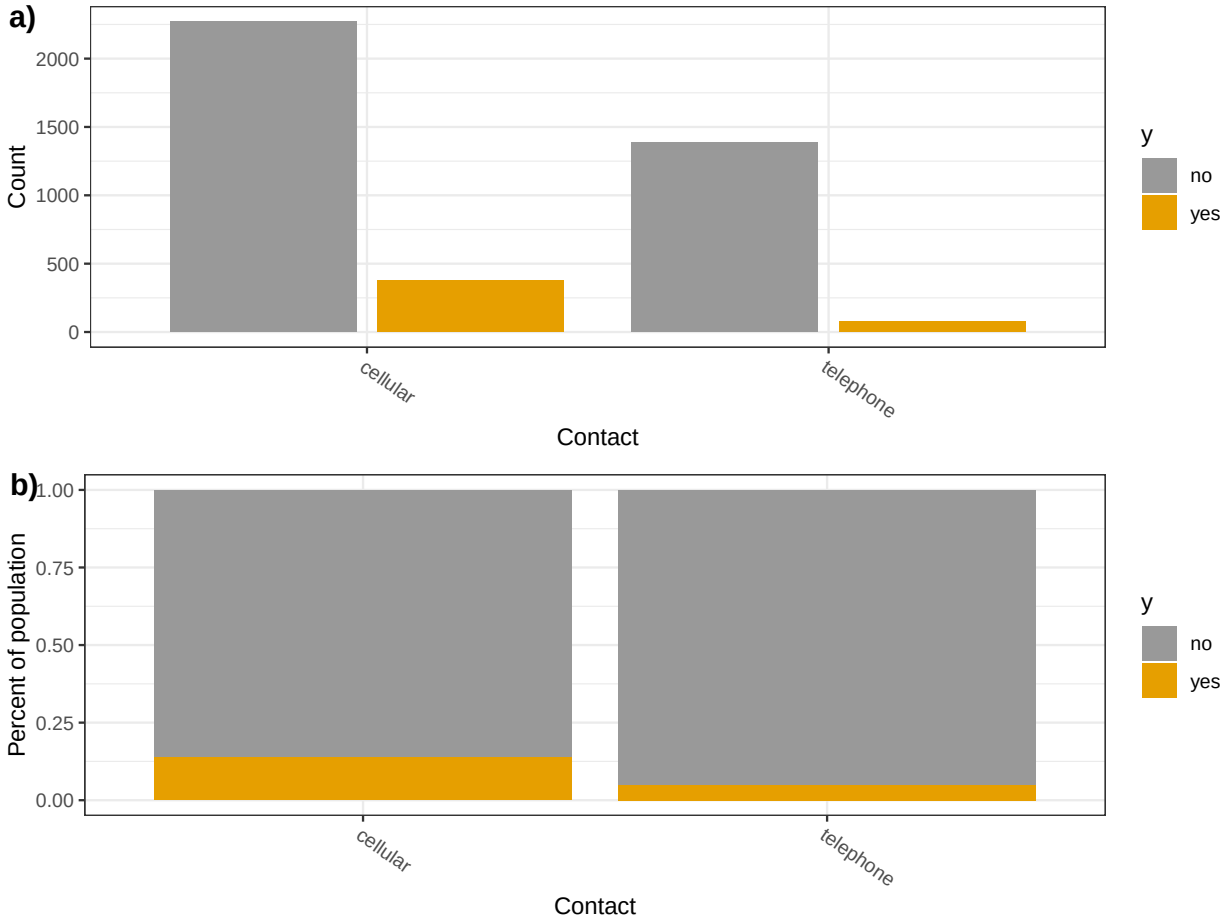


Rysunek 9: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od już posiadanej pożyczki; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od posiadanej pożyczki.

Niestety poziom istotności(p-value) na poziomie 73% świadczy o dużej zależności między danymi, więc tej zmiennej również nie będę brał pod uwagę.

2.8 Contact communication type

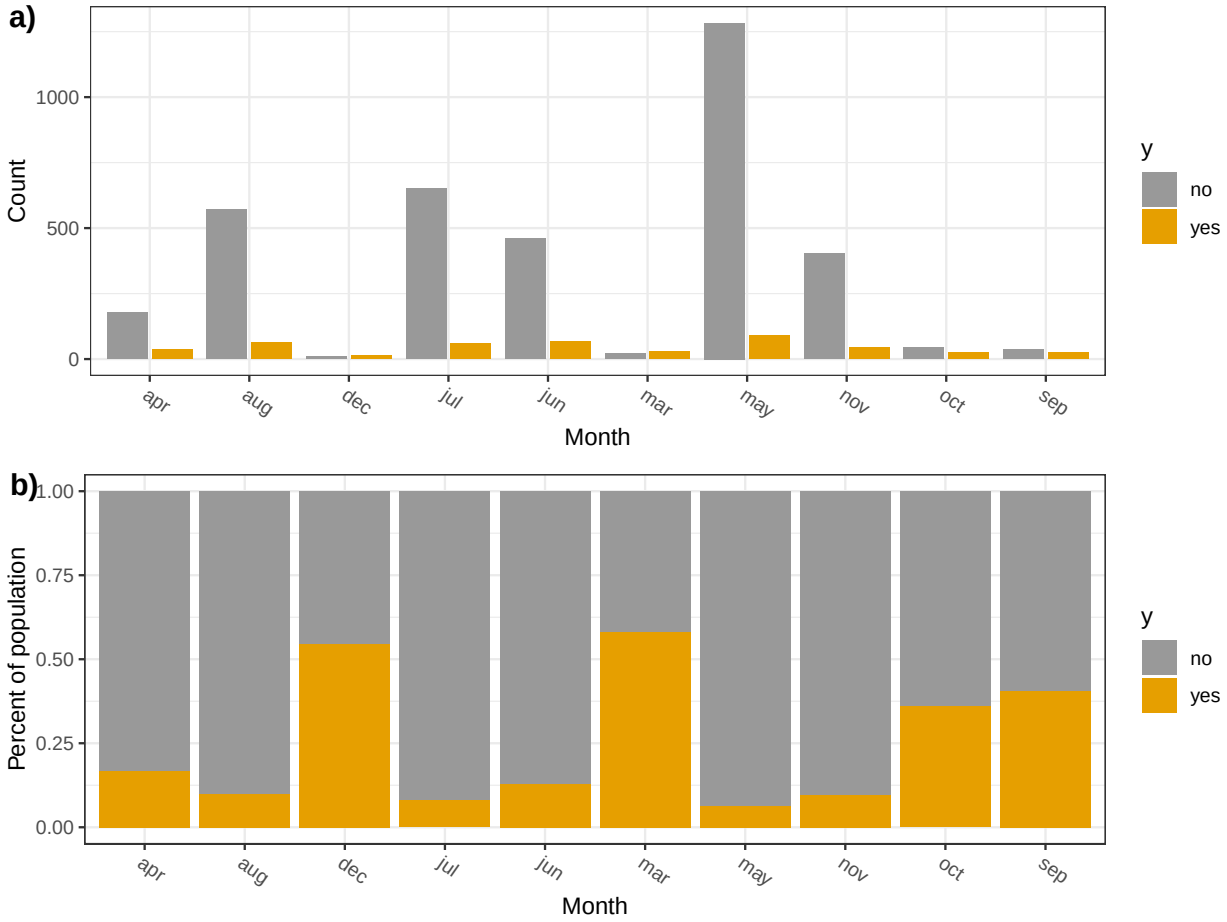
Osoby, z którymi próbowano się skontaktować na telefon komórkowy stanowią 64.4% całej badanej społeczności i co 6 osoba z nich wzięła lokatę długoterminową.



Rysunek 10: 'Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od sposobu kontaktu z klientem; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od sposobu kontaktu z klientem.

2.9 Last contact month of year

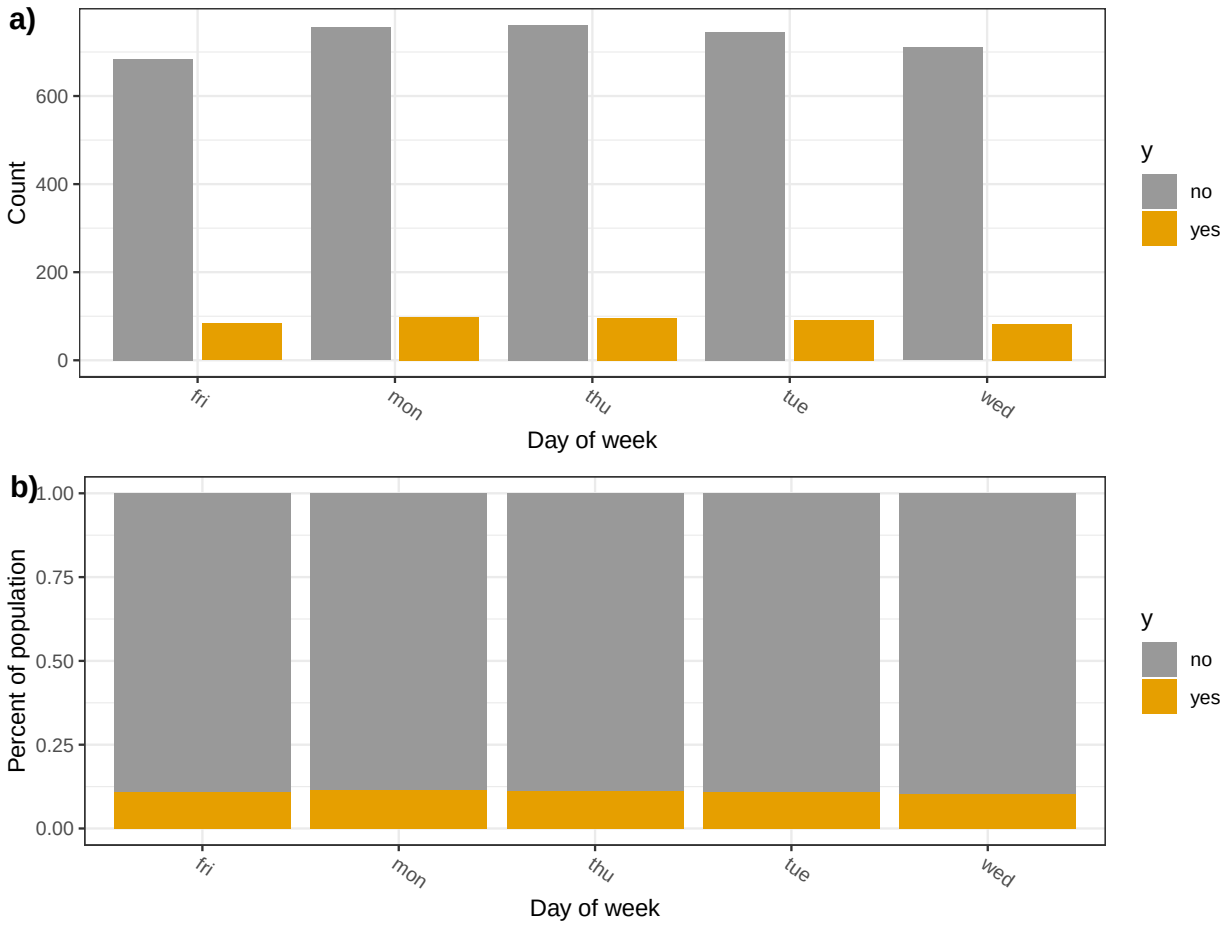
Ciekawą sytuacją jest fakt, że w zestawieniu w ogóle nie mamy transakcji za styczeń oraz za luty.



Rysunek 11: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od miesiąca, w którym kontaktowano się z klientem; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od miesiąca, w którym kontaktowano się z klientem.

2.10 Last contact day of the week

W każdym roboczym dniu tygodnia jest wykonywane mniej więcej tyle samo połączeń z klientami, więc nie jesteśmy w stanie wyciągnąć żadnych większych wniosków, z obserwacji samych wykresów.



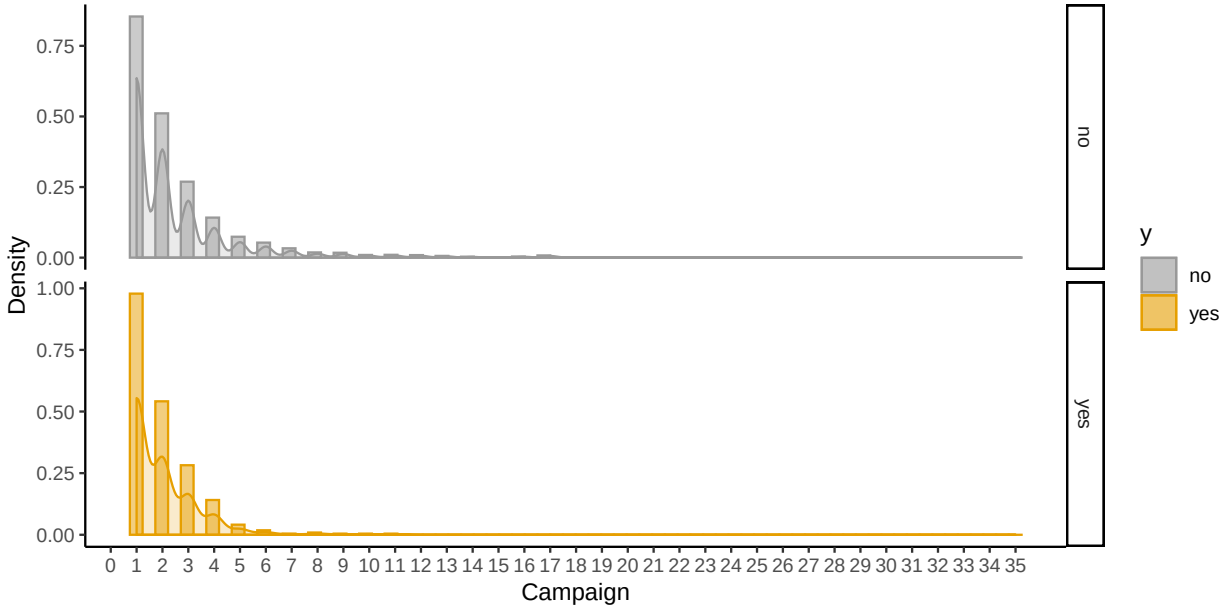
Rysunek 12: Barplot a) przedstawiający jak wiele osób założyło lokatę w zależności od dnia, w którym kontaktowano się z klientem; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od dnia, w którym kontaktowano się z klientem.

2.11 Last contact duration, in seconds

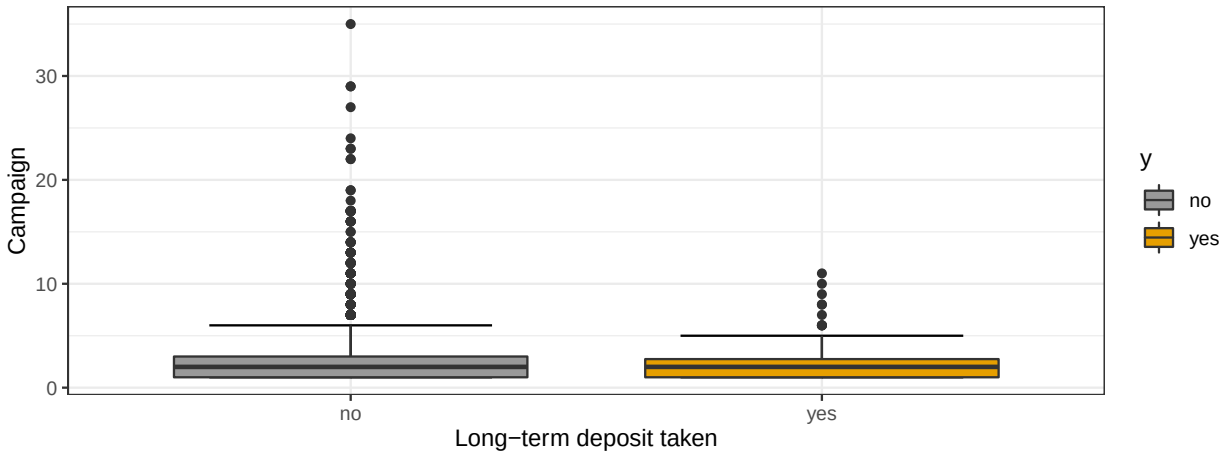
Czas trwania ostatniego kontaktu jest atrybutem, który ma duży wpływ na cel wyjściowy (y) Ważna uwaga: ten atrybut ma duży wpływ na cel wyjściowy. Jednak czas trwania nie jest znany przed wykonaniem połączenia. Ponadto po zakończeniu połączenia "y"jest oczywiście znane. W związku z tym należy ją odrzucić, jeśli chcemy stworzenie realistycznego modelu predykcyjnego.

2.12 Campaign

W tej kolumnie zawierają się informacje dotyczące liczby połączeń do danego klienta w ramach kampanii. Przyglądając się boxplotowi możemy zauważyć, że obserwacje od 7 są już uważane jako wartości odstające.



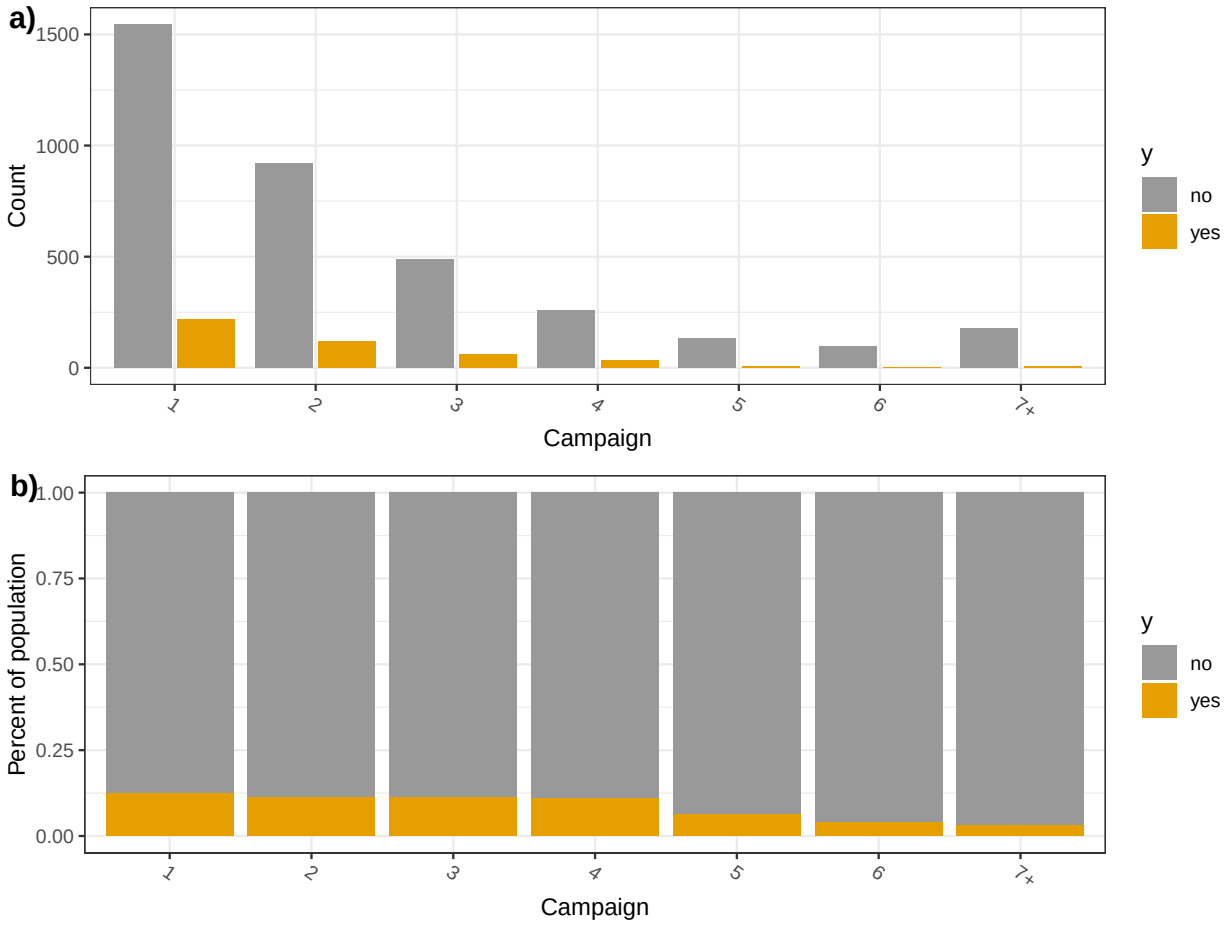
Rysunek 13: Histogram a) przedstawiający jak wiele osób założyło lokatę w zależności od liczby wykonanych telefonów do klienta; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od liczby wykonanych telefonów do klienta.



Rysunek 14: Boxplot a) przedstawiający jak wiele osób założyło lokatę w zależności od liczby wykonanych telefonów do klienta; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od liczby wykonanych telefonów do klienta.

W związku z czym wszystkie wartości, które są większe niż 7, ale mniejsze niż 12

zamienię na 7+, a następnie wszystkie zmienne zamieniam na zmienne kategoryczne. Zmienne powyżej 12 odrzucam.



Rysunek 15: Histogram ,dla obrobionych danych, a) przedstawiający jak wiele osób założyło lokatę w zależności od liczby wykonanych telefonów do klienta; b) przedstawiający stosunek procentowy osób, które założyły lokatę z zależności od liczby wykonanych telefonów do klienta.

2.13 Pdays

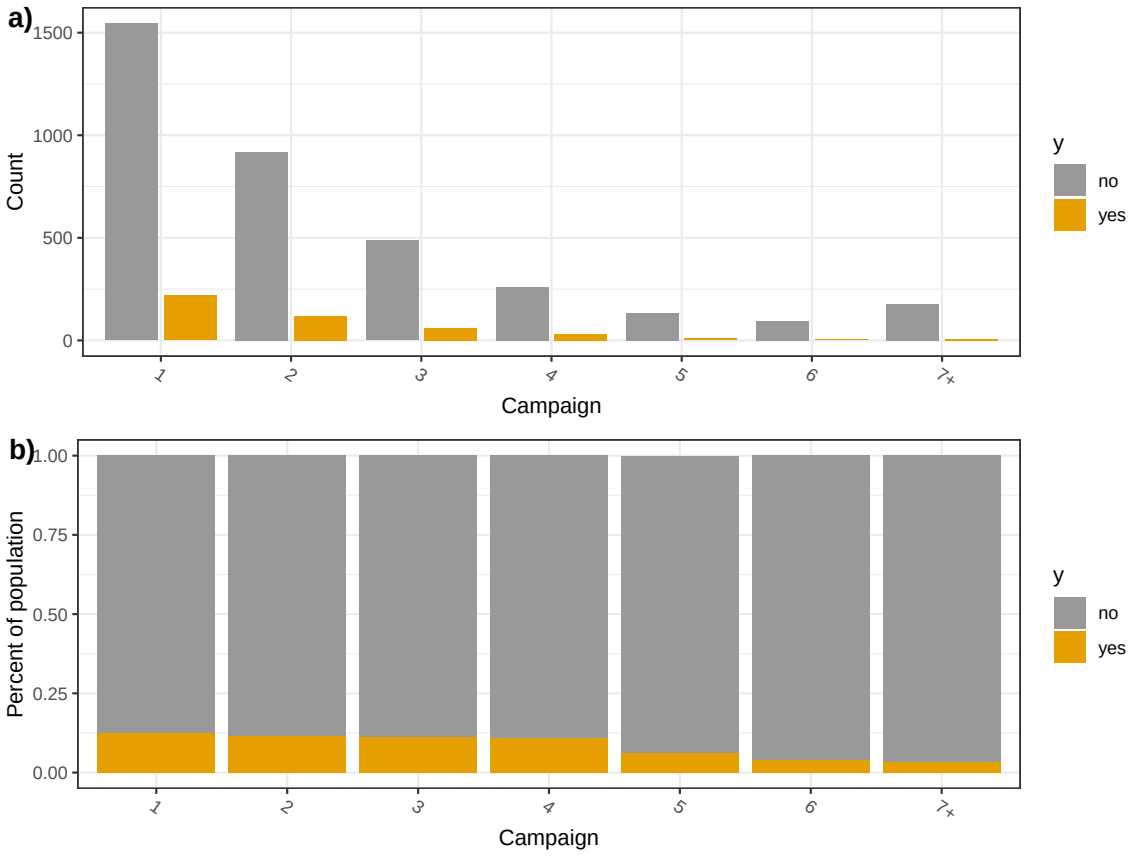
Kolumna 'pdays' znajdują się informacje o ilości dni od ostatniego kontaktu z klientem podczas poprzedniej kampanii. Jeśli wartość wynosi "999"to oznacza, ze nigdy się z daną osobą nie skontaktowane. Informacja zawarta w tej kolumnie jest zbyt rozproszona i niejednoznaczna. W związku z tym tą informację zamienimy w taki sposób, żeby odpowiadała na pytanie: czy kontaktowano się z daną osobą w poprzedniej kampanii?

2.14 Previous

Zmienna 'previous' zawiera informacje dotyczące liczby kontaktów wykonanych przed tą kampanią do tego klienta. W obecnym stanie, ta zmienna zwraca bardzo podobne wyniki co zmienna "pdays binary". Ponieważ liczba klientów, ż którymi kontaktowano się 2 i więcej razy jest wyjątkowa mała w stosunku do osób, z którymi się nie kontaktowano to proponuję połączyć te zmienne w większą całość. Tak przygotowana zmienna powie nam, czy nękanie osób poprzez częstsze kontakty telefoniczne daje wymierne skutki.

2.15 Poutcome

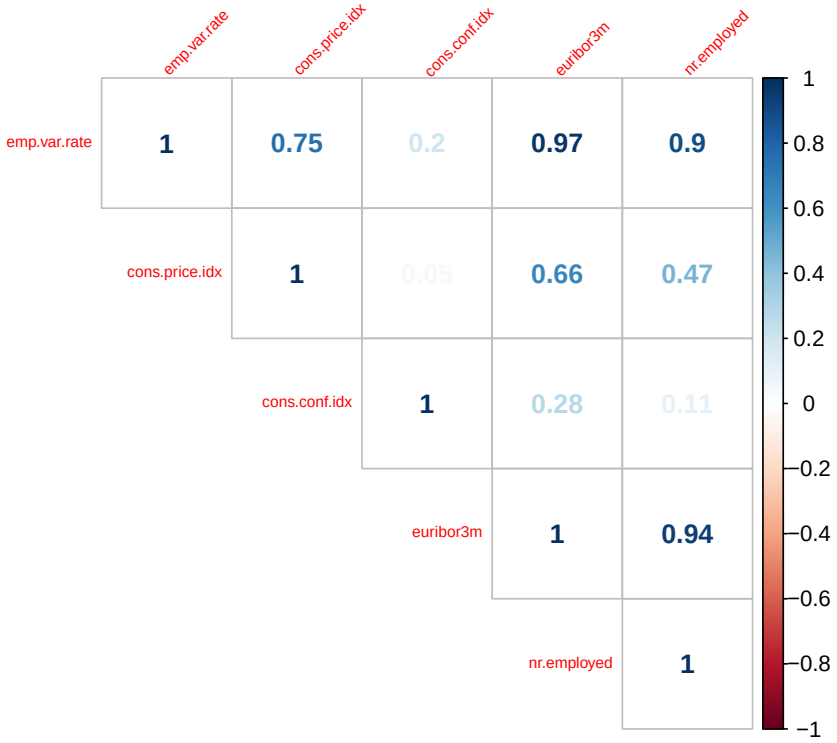
Czy sukces lub porażka podczas poprzedniej kampanii dały się odczuć w obecnej kampanii?? Jak najbardziej. Aż 64.8% osób chciałby przedłużyć swoją lokatę. 14.8% osób , mimo że ostatnio nie wzięli lokaty to czas to czas między kampaniami dał im sporo do myślenia i spowodował, że osoby, z którymi się kontaktowano w poprzedniej kampanii chętniej brały lokatę długoterminową niż osoby, z kótrymi kontaktowano się pierwszy raz.



Rysunek 16: Barplot a) przedstwiający jak wiele osób założyło lokatę w zależności od sukcesu w poprzedniej kampanii; b) przedstwiający stosunek procentowy osób, które założyły lokatę z zależności od sukcesu w poprzedniej kampanii.

2.16 Social and economic context attributes

Ostatnie 5 współczynników to wskaźniki społeczne oraz ekonomiczne. Analiza każdej z osobna nie pokazał żadnej charakterystycznej cech. Jednakże te zmienne powinny być mocno skorelowane ze sobą, bo opisują podobne cechy. Obliczmy macierz korelacji.



Rysunek 17: Macierz korelacji między zmiennymi numerycznymi występującymi w naszej bazie danych.

Zgodnie z oczekiwaniami, zmienne społeczne i ekonomiczne są silnie ze sobą skorelowane. Szczególnym przypadkiem silnego skorelowania charakteryzuje się zmienna ęps.price.rate", która jest wskaźnikiem zmienności zatrudnienia. W związku z czym ta zmienna zostaje usunięta.

2.17 y (target)

Ostatnią omawianą cechą jest zmienna 'y'. Zawiera ona informacje czy klient skorzysta z lokaty czy nie? Ponieważ nie możemy zostawić tej zmiennej w postaci zmiennej charakterystycznej, musimy zamienić ją na zmienną numeryczną.

3 Modele predykcyjne

3.1 Przygotowanie danych

W tej części zostaną wykonane 3 kroki potrzebne do prawidłowego działania modeli predykcyjnych oraz poprawienia wydajności modelu.

3.1.1 Normalizacja

Pierwszą z nich jest normalizacja. Celem normalizacji jest ujednolicenie danych wejściowych, numerycznych, poprzez ich przeskalowanie. W tym przypadku będę skalował dane do rozkładu normalnego. Ta czynność jest konieczna jeśli chcemy tworzyć modele inne niż drzewa decyzyjne lub lasy losowe.

3.1.2 Faktoryzacja zmiennych tekstowych

Drugą czynnością jest zamiana zmiennych tekstowych na zamienne fikcyjne, czyli *dummy variable*. Powoduje to, że każdą zmienną tekstową zapisujemy w naszych danych jako zmienną 'factor', czyli każda opcja będzie rozpatrywana przez model w sposób binarny. Podział danych Nasze dane są już gotowe do podziału na zbiór treningowy oraz zbiór walidacyjny. Dane treningowe posłużą do wytrenowania naszego modelu, natomiast dane testowe zostaną wykorzystane podczas walidacji modelu. Dane zostaną podzielone w stosunku 80%/20% przy pomocy biblioteki `caret`, gdyż pozwoli nam to zachować taką samą proporcje danych w kolumnie 'y'.

3.2 Dobór odpowiednich miar do badania jakości modelu predykcyjnego

Na tym etapie przystępujemy do generalnego opisu danych i co chcemy wyciągnąć z danych:

- Nasze dane są mocno niezrównoważone (stosunek 89% negatywnych odpowiedzi do 11% pozytywnych odpowiedzi). To powoduje, że łatwo będzie nam osiągnąć wysokie accuracy, które w rzeczywistości może nie oddawać dokładnie tego co chcemy;
- Bo czego my oczekujemy od naszego modelu? Nasz model ma przewidywać, który klient rzeczywiście założy lokatę długoterminową w naszym banku. Jeśli przyjrzymy się macierzy pomyłek (*Confusion matrix*) to opisaną sytuację możemy badać przy pomocy miary *True Positive* (TP).
- Pod miarą *False Positive* (FP) będą kryli się klienci, którzy jeszcze nie będą gotowi na założenie lokaty długoterminowej
- Pod miarą *False Negative* (FN) będą kryli się klienci, którzy są chętni założyć lokatę, jednakże bank się z nimi nie skontaktował. Ta miara w naszym przypadku będzie przynosić największe straty dla banku.
- A pod ostatnią miarą *True Negative* (TN) będą wszyscy klienci banku, którzy rzeczywiście nie chcą założyć lokaty.

- Czyli, żeby maksymalizować zyski musimy skupić się na maksymalizacji miar *Recall* oraz *Precision*. Jednakze wiedząc, że *f1 score* jest średnią harmoniczną z *Recall* oraz *Precision*, to maksymalizowanie tej miary przyniesie nam największe korzyści.
- W dalszym etapie porównywania modeli skupimy się na porównaniach krzywych ROC i krzywych PR.

Przy tak przygotowanym planie badania jakości modelu możemy przystąpić już do ich tworzenia.

W ramach tej pracy będę wykorzystywał bibliotekę `mlr` zaimplementowaną w środowisku `R`. Głównym jej atutem jest fakt, że w klarowny sposób możemy wyznaczyć najlepsze hiper-parametry do naszych modeli.

W ramach tej pracy skupię się na 4 modelach:

- Logistic regression
- Random forest
- SVM
- XGBoost

Zanim jednak przjedziemy do tworzenia modeli musimy zrobić 'Zadanie'. Modele zaimplementowane w pakiecie `mlr` do uczenia się potrzebują tzw. *Tasks*, a ponieważ mamy do czynienia z problemem klasyfikacyjnym to skorzystamy z funkcji `makeClassifTask` która odpowiednio przygotuje nam dane dla modelu. To ułatwia pracę, gdyż nie musimy dzielić zbioru na

3.3 Logistic Regresion

Opis modelu

Model regresji liniowej jest jednym z powszechniejszych modeli uczenia maszynowego, który w specyficzny sposób uogólnia modele liniowe. Regresja logistyczna stosowanej jest do prognozowania wyniku jakościowo zależnej zmiennej na podstawie zestawu zmiennych predykcyjnych lub niezależnych. W regresji logistycznej zmienna zależna jest zawsze binarna.

W regresji logistycznej nie mamy żądnych wewnętrznych stałych parametrów, przy niej kształtujemy model w zależności od ilości parametrów dodanych do badania. Jednakże w `mlr` model sam dobiera optymalną ilość parametrów ze zbioru treningowego.

Jedyne nad czy musimy się skupić to jak wiele prób chcemy przeprowadzić podczas walidacji krzyżowej (*cross-validation*). Metoda walidacji krzyżowej polega na podziale zbioru na K podzbiorów i podczas trenowania modelu, uczy się on zawsze na podzbiorach od 1 do $K-1$, a pozostały podzbiór służy do testowania. Model jest szkolony i testowany K razy, gdzie za każdym razem jest używany inny podzbiór do testowania. podczas gdy pozostałe zestawy są używane do treningu. Ostatecznie uśredniamy wynik z wszystkich otrzymanych miar. Dla tego modelu K=5, a miarą kontrolną jest *f1 score*.

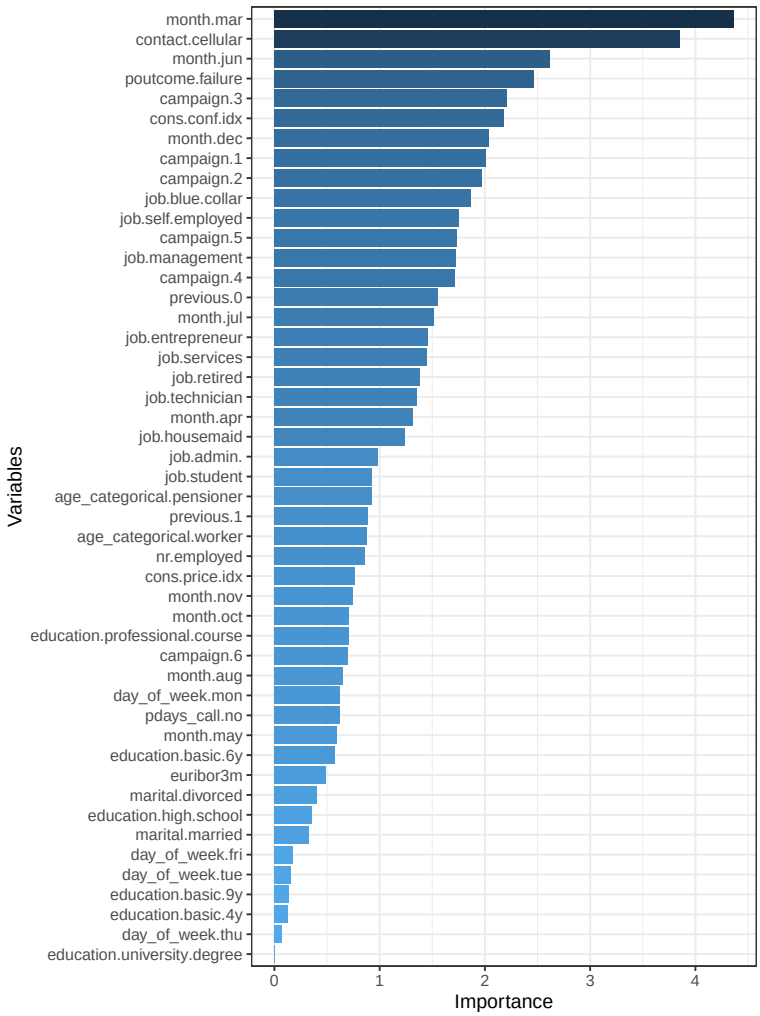
Wyniki

Podsumowanie wszystkich badanych miar przedstawię we wnioskach.

Można zauważyć, że macierz pomyłek 2 nie daje nam obiecującej przyszłości. 73 osoby, które są chętna założyć lokatę długoterminową, ale ze względów braku kontaktu z bankiem ci klienci tego nie zrobią.

	0	1
0	703	11
1	73	16

Tablica 2: Macierz pomyłek dla regresji logistycznej



Rysunek 18: Wykres ważności zmiennych dla modelu regresji logistycznej

Ciekawym zjawiskiem jest fakt, że konkretne miesiące mają wysoki wpływ na zakup lokaty.

3.3.1 Random Forest

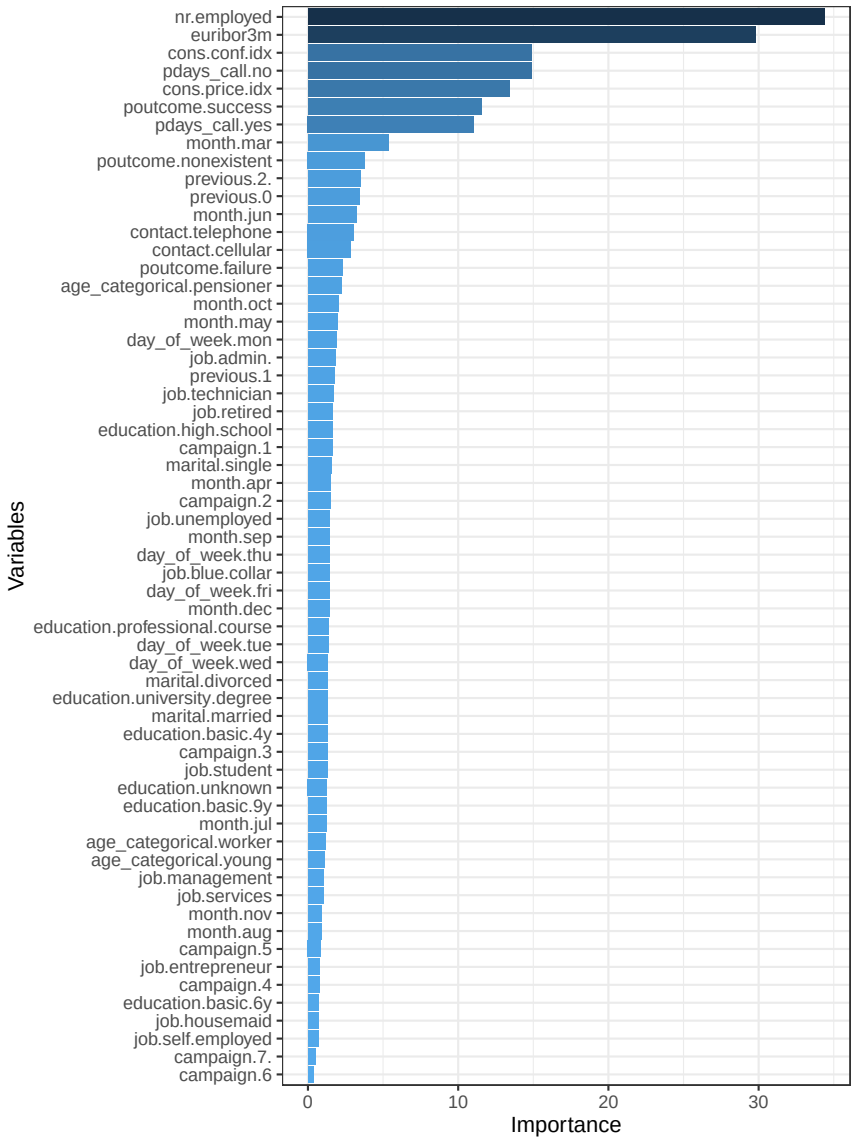
Opis modelu Las losowy (Random forest) jest metodą zespołową uczenia maszynowego dla klasyfikacji, regresji i innych zadań, która polega na konstruowaniu wielu drzew decyzyjnych w czasie uczenia i generowaniu klasy, która jest dominantą klas (klasyfikacja) lub przewidywaną średnią (regresja) poszczególnych drzew w lesie. Dużą

zaletą lasów losowych w stosunku to drzew decyzyjnych jest fakt, że lasy losowe nie mają aż tak nadmiernego dopasowywania się do zestawu treningowego.

Dla tego modelu będziemy badać następujące hiper-parametry:

- *ntree* - liczba drzew w lesie i będziemy szukać najlepszej wartości na przedziale między 50 a 500;
- *mtry* - liczba zmiennych losowychch rozważana przy każdym podziale i będziemy szukać najlepszej wartości na przedziale między 3 a 20;
- *nodesize* - rozmiar węzła. Im większy nodesize tym drzewa w lesie mają większą "gębokość". Będziemy szukać najlepszej wartości na przedziale między 10 a 50.

Wyniki Zaproponowane hiper-parametry dla lasów losowych to: *ntree*=256, $mtry = 4$ oraz $nodesize = 25$.



Rysunek 19: Wykres ważności zmiennych dla modelu lasu losowego

	0	1
0	704	10
1	74	15

Tablica 3: Macierz pomyłek dla lasu losowego

3.3.2 SVM

Opis modelu

Maszyna wektorów wsparcia (SVM) to algorytm uczenia maszynowego, który analizuje dane w celu klasyfikacji i analizy regresji. SVM to nadzorowana metoda uczenia się, która analizuje dane i dzieli je na jedną z dwóch kategorii. SVM generuje mapę posortowanych danych z marginesami między nimi tak daleko od siebie, jak to możliwe. Maszyny SVM są używane w kategoryzacji tekstu, klasyfikacji obrazów, rozpoznawaniu pisma ręcznego oraz w naukach ścisłych.

W tej metodzie również będziemy ustawiać hiper-parametry dla modelu. W tym przypadku będą to parametry dyskretne:

- C - parametr kosztu;
- *Sigma* - odchylenie standardowe

Badane wartości C to $\{\frac{1}{2^8}, \frac{1}{2^4}, \frac{1}{2^2}, 1\}$, a *Sigma* z $\{\frac{1}{2^8}, \frac{1}{2^4}, 1, 2^4\}$

Wyniki

Hiper-parametry: C= $\frac{1}{2^8}$, *Sigma*= $\frac{1}{2^8}$

	0	1
0	706	8
1	75	14

Tablica 4: Macierz pomyłek dla SVM

3.3.3 XGBoost

Opis modelu XGBoost to implementacja drzew decyzyjnych wzmocnionych gradientem. Główną ideą XGBoosta jest tworzenie ciągu (bardzo) prostych drzew, z których każde kolejne jest zbudowane do predykcji reszt generowanych przez poprzednie drzewo, czyli metoda buduje drzewa binarne. Dobór hiper-parametrów dla modelu wygląda następująco

```
xgb_params <- makeParamSet(
    makeIntegerParam("nrounds", lower = 100, upper = 1000),
    makeIntegerParam("max_depth", lower = 1, upper = 10),
    makeNumericParam("eta", lower = 0.025, upper = 0.5),
    makeNumericParam("gamma", lower = 0, upper = 1),
    makeNumericParam("min_child_weight", lower = 1, upper = 3),
    makeNumericParam("colsample_bytree", lower = 0.2, upper = 1),
    makeNumericParam("lambda", lower = -1, upper = 0, trafo = function(x) 10^x)
)
```

Wyniki

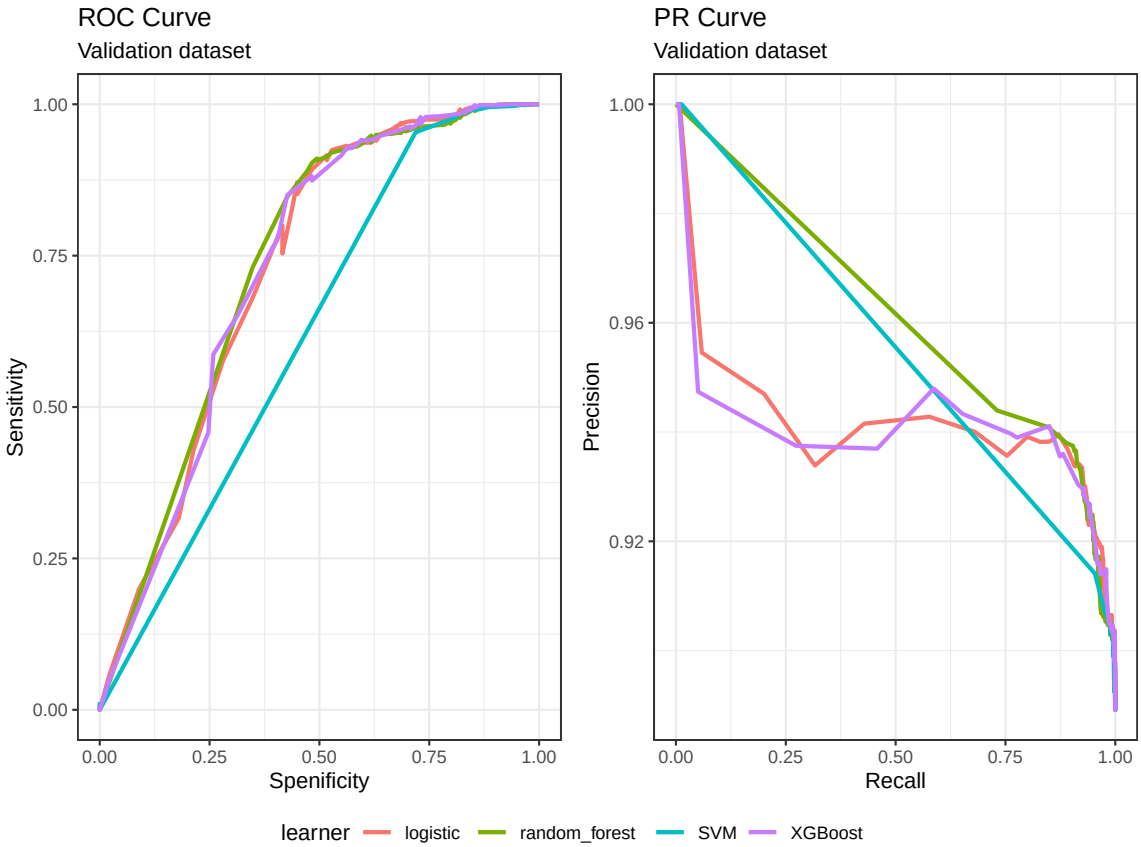
Hiper-parametry wyglądają następująco:

```
nrounds=422
verbose=0
objective=binary:logistic
eval_metric=error
max_depth=1
eta=0.312
gamma=0.806
min_child_weight=1.93
colsample_bytree=0.945
lambda=0.231
```

	0	1
0	707	7
1	75	14

Tablica 5: Macierz pomyłek dla modelu XGBoost

4 Analiza jakości modeli



Rysunek 20: Wykres krzywej ROC oraz krzywej PR dla 4 modeli predykcyjnych

Przyglądając się wykresom 20 (ROC) możemy zauważyć, że zarówno regresja logistyczna jak i lasy losowe oraz XGBoost dały zbliżone wyniki. Niestety nie możemy tego samego powiedzieć o SVM, który wyraźnie charakteryzuje się niższą efektywnością.

Różnica natomiast pojawia się na rysunku 20 (PR), gdzie regresja logistyczna oraz XGBoost nagle maleją. Natomiast last losowe prezentują się ciągle najlepiej.

	Logistic regresion	Random Forest	SVM	XGBoost
Accuracy	0.8878549	0.8966358	0.8916204	0.8916667
Precision	0.9073351	0.9040491	0.8983615	0.9046399
Recall	0.9734161	0.9888983	0.9901712	0.9821565
Specificity	0.19760101	0.15366162	0.09873737	0.15934343
Auc	0.6712535	0.7010921	0.6534101	0.6903625
F1 score	0.9390426	0.9444198	0.9419014	0.9415304

Tablica 6: Tabela porównawcza miar jakości modeli predykcyjnych

Tak jak przewidywaliśmy na początku, skuteczność nie jest najbardziej wiarygodną

miarą w tym zestawieniu, gdyż osiągnęła ok 89% w każdym modelu co jest porównywalne wynikiem do procentowej ilości opcji "0"w danych testowych.

Precision na poziomie 90% oraz *recall* na poziomie nawet 99% jest wynikiem wręcz idealnym. A co za tym idzie *F1 score* wychodzi na poziomie 94%.

5 Wnioski

Jednak jeśli miałbym wybrać model, który spisał się najlepiej dla tych danych to byłby to model lasów losowych, gdyż prawie w każdej mierze osiągną najlepszy wynik oraz najlepiej prezentował się na wykresach wrzywych ROC oraz PR.