Raport

Paweł Morgen, Zuzanna Mróz, Aleksander Podsiad

Wstęp

W zbiorze danych *german credit data* znajdują się dane na temat kredytobiorców, klasyfikowanych na podstawie różnych cech (wiek, zatrudnienie, typ miejsca zamieszkania itp.) jako *dobrzy* bądź *źli* klienci. Poniżej zaprezentowano wyniki eksploracji powyższych danych, a także nasze wyniki odnośnie poszukiwania modelu, który byłby w stanie jak najtrafniej zaklasyfikować kredytobiorców do jednej z tych dwóch kategorii na podstawie ich danych.

Eksploracja zbioru

Opis

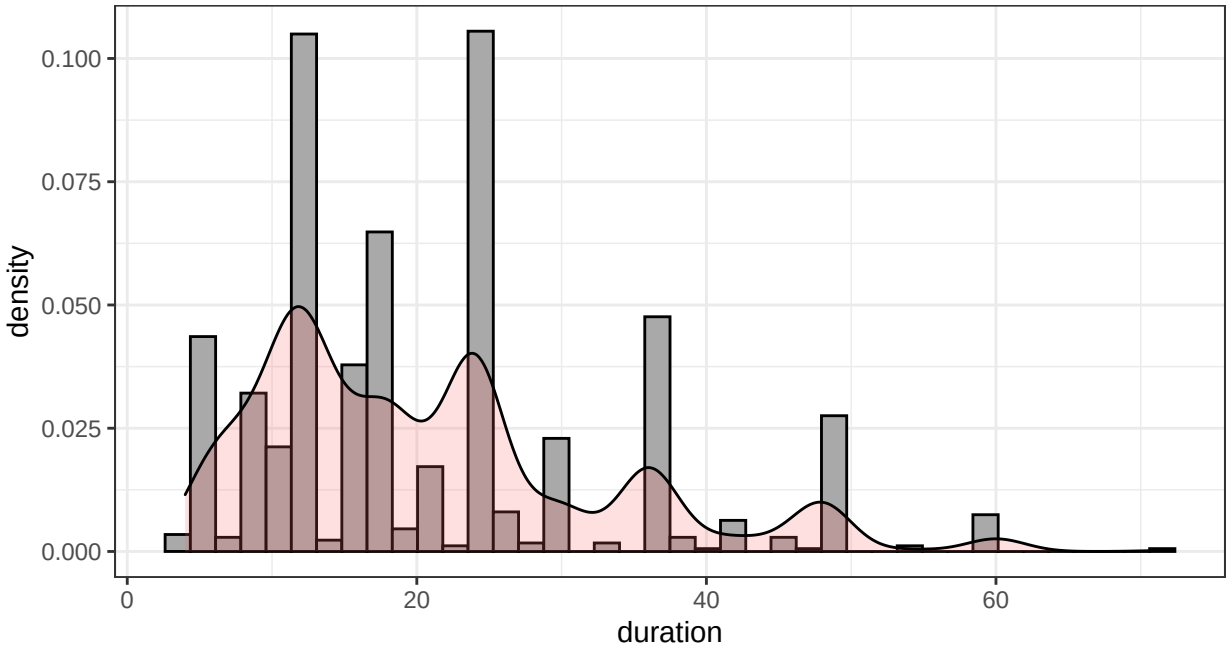
Zbiór *german credit data* posiada następujące informacje:

- checking_account_status - stan rachunku bieżącego
- duration - czas w miesiącach
- credit_history - podsumowanie historii kredytobrania
- purpose - powód kredytu
- credit_amount - ilość kredytu
- savings string - oszczędności na kontach
- present_employment - stan/czas zatrudnienia
- installment_rate - stawka raty jako procent dochodu do dyspozycji
- personal - osobisty stan cywilny i płeć
- other_debtors - inni dłużnicy
- present_residence - obecne miejsce zamieszkania
- property - własności
- age - wiek w latach
- other_installment_plans - inne plany ratalne
- housing - zamieszkanie (renta/własne/za darmo)
- existing_credits - liczba istniejących kredytów w tym banku
- job - typ zatrudnienia
- dependents - liczba osób odpowiedzialnych za utrzymanie
- telephone - czy jest zarejestrowany numer telefony
- foreign_worker czy jest obocokrajowym pracownikiem
- customer_type - czy jest dobrym klientem

W powyższym zbiorze nie ma brakujących informacji

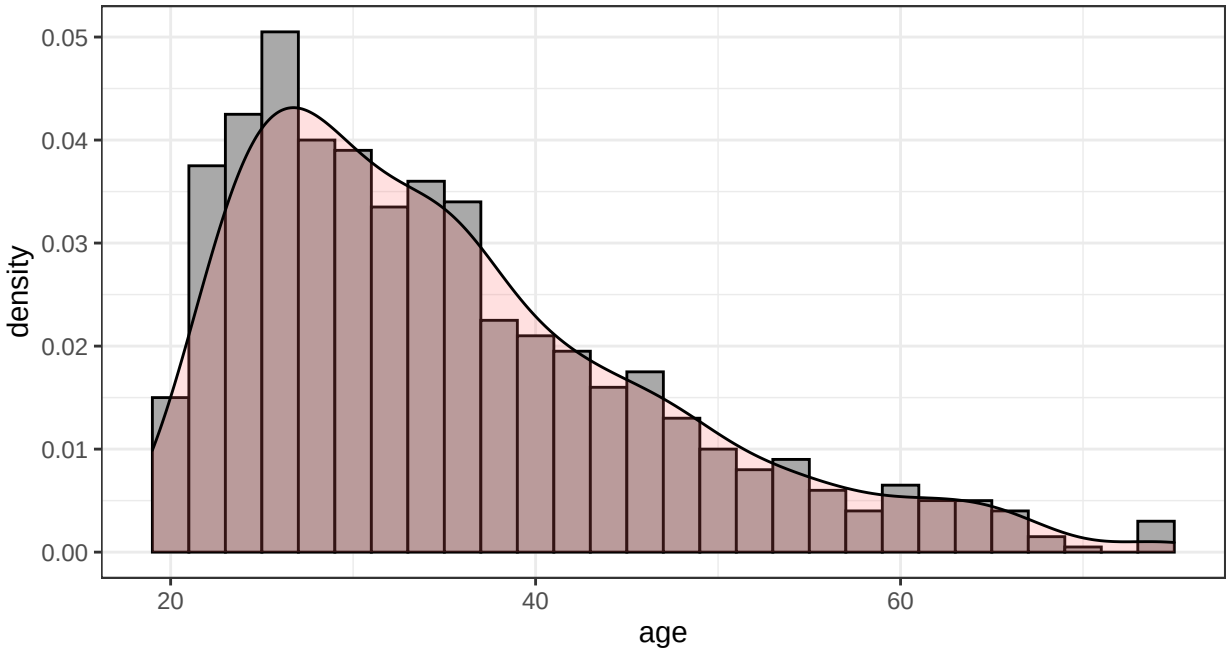
Eksploracja jednowymiarowa

Rozkład czasu trwania kredytu



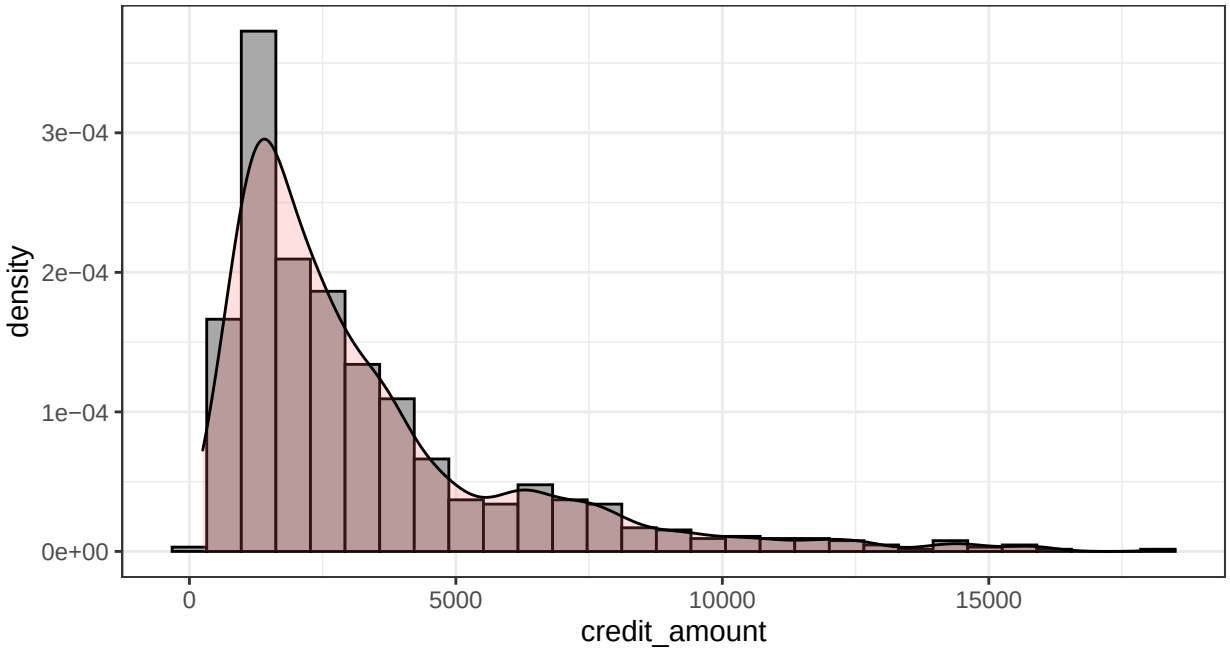
- widać wyraźnie, że niektóre czasy trwania kredytu są znacznie popularniejsze.
- najwięcej jest kredytów jedno i dwuletnich.

Rozkład wieku klientów



- rozkład jest dodatnio skośny ze względu na możliwość brania kredytu dopiero od pewnej granicy wiekowej.
- młodsze osoby częściej biorą kredyty.

Rozkład wysokości kredytu

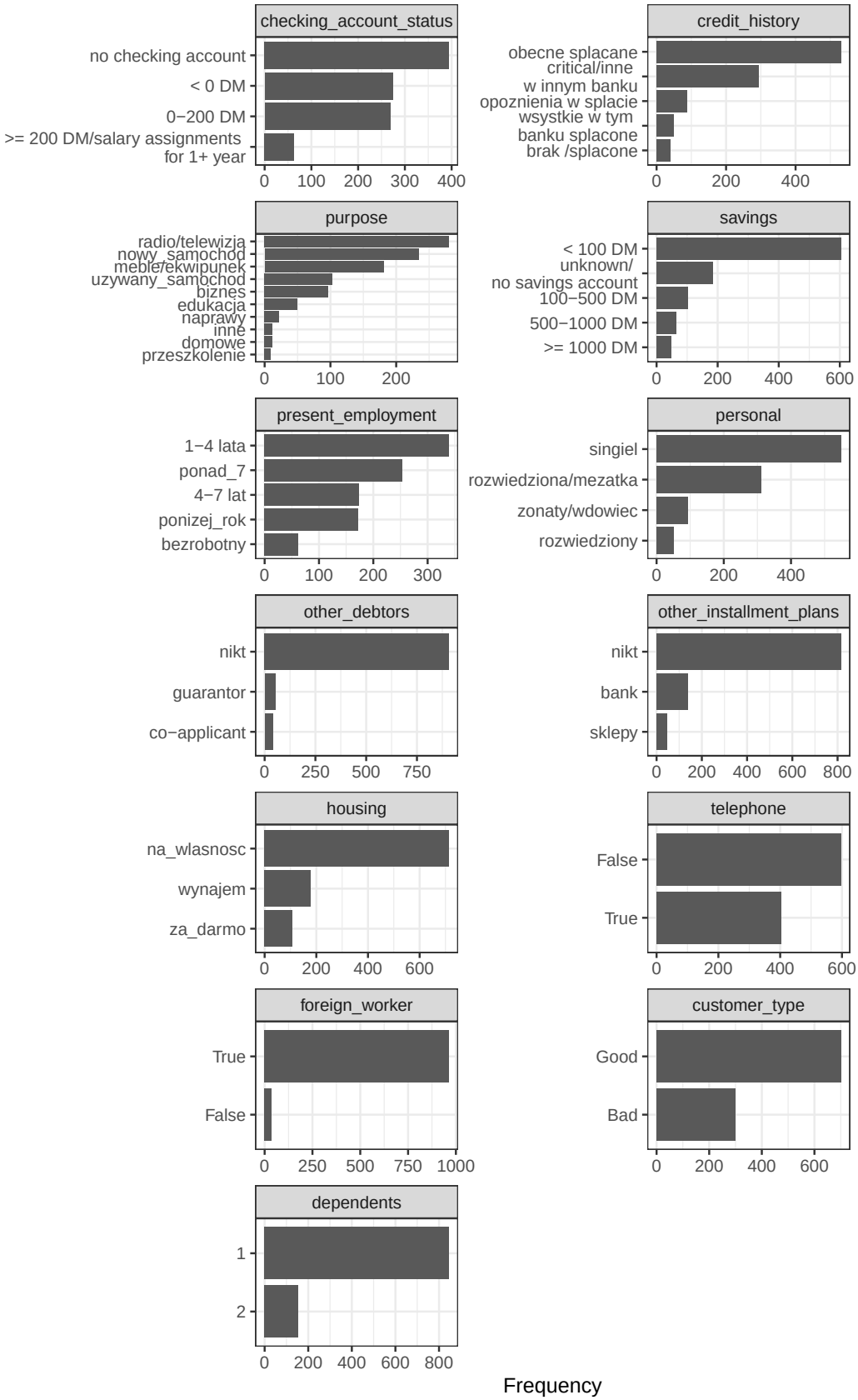


- rozkład jest dodatnio skośny z długim ogonem.
- ludzie częściej biorą “małe” kredyty.

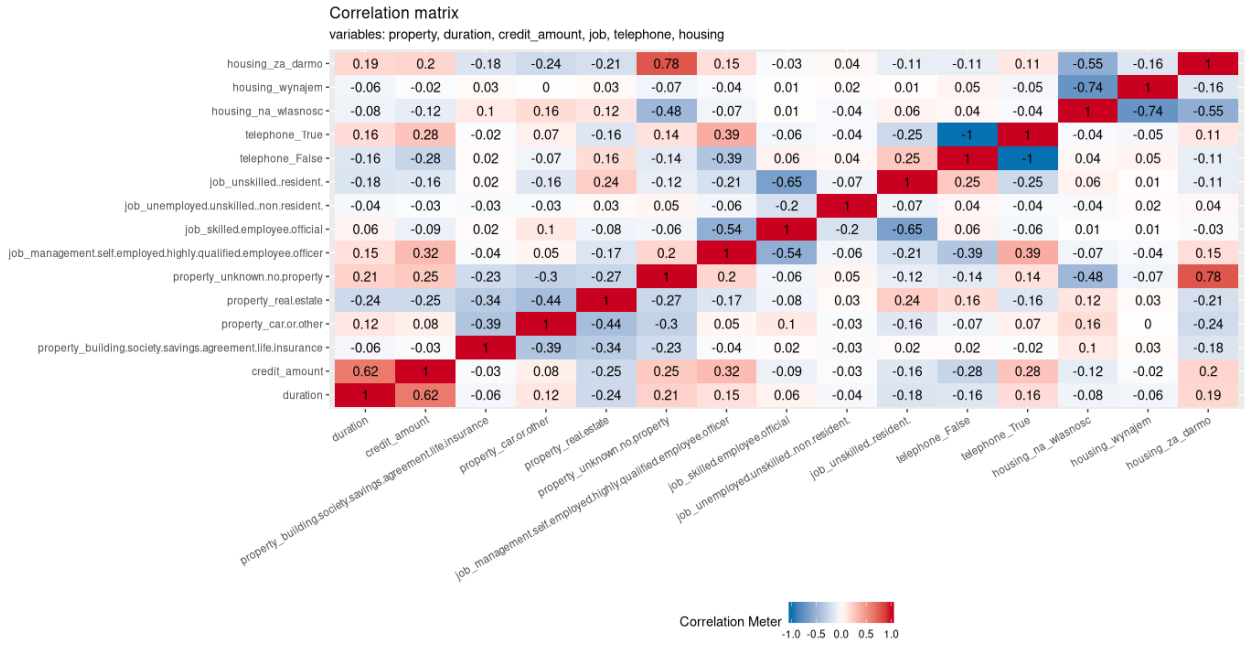
Omówienie zmiennych dyskretnych

- większość klientów nie ma konta czekowego
- klienci ze spłaconymi kredytami są w znaczącej mniejszości
- klienci jako powód zaciągnięcia kredytu najczęściej podają sprzęt AGD, samochody i meble
- w większości klienci tego banku mają znikome oszczędności lub nie mają ich wcale
- niezatrudnieni klienci stanowią mniejszość
- jest około dwa razy więcej klientów płci męskiej niż żeńskiej
- w znaczącej większości klienci nie mają innych planów ratalnych
- więcej klientów nie zarejestrowało numeru telefonu niż zarejestrowało
- ogromna część klientów to obcokrajowi pracownicy
- klientów zaklasyfikowanych jako ‘dobrych’ jest około dwa razy więcej niż “złych”

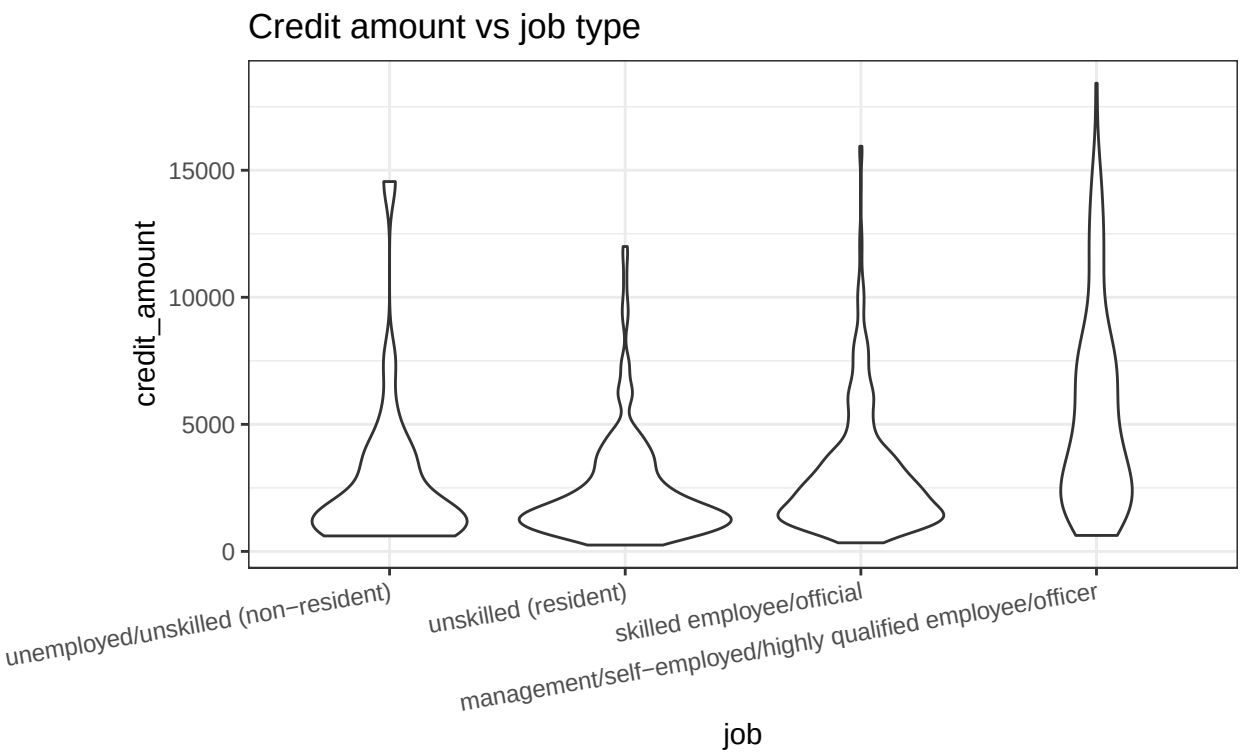
Wykresy słupkowe zmiennych dyskretnych



Eksploracja związków między zmiennymi



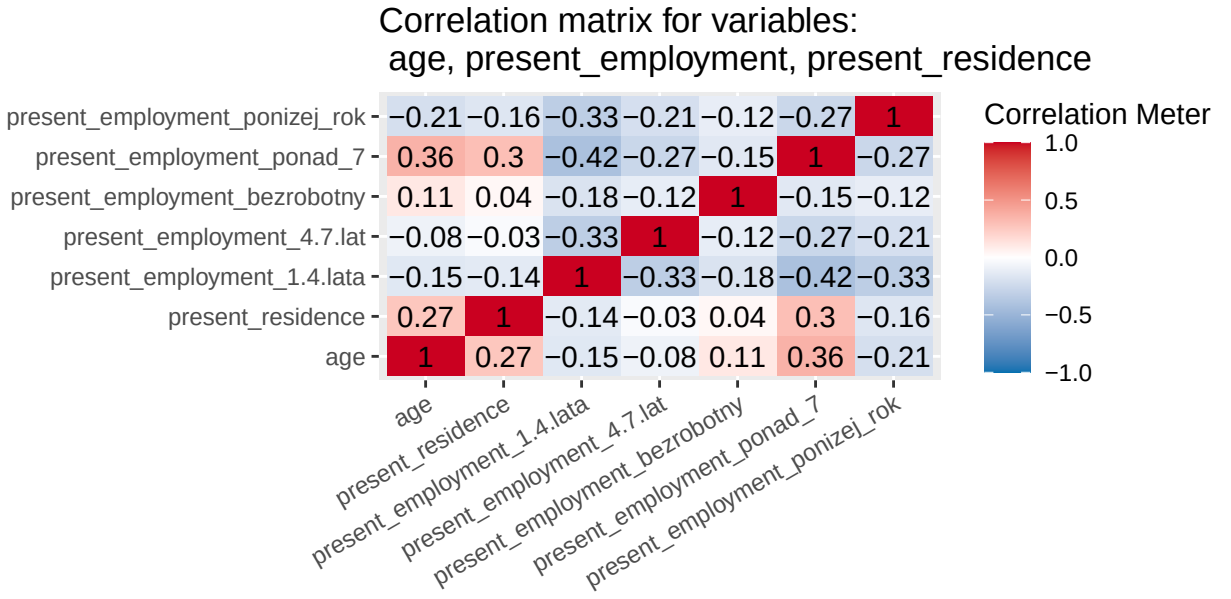
- Zmienne `credit_amount` i `duration` są silnie (0.62) skorelowane. Nie dziwi nas to, ponieważ z reguły im kredytów na większe kwoty udziela się na dłużnicyuższy czas spłaty.
- Widzimy, że jeśli zmienna `credit_amount` zależy od `job`. Jest to związane ze zdolnością kredytową:



- Zmienna `telephone` jest pewnym wyznacznikiem zmienych `credit_amount`oraz `job` - zwłaszcza dla wartości *highly qualified* i *unemployed*. Ponownie nas to nie dziwi - im większy kredyt (`credit mount`) i zarobki (związane z `job`) tym większa szansa na to, że ktoś posiada telefon.

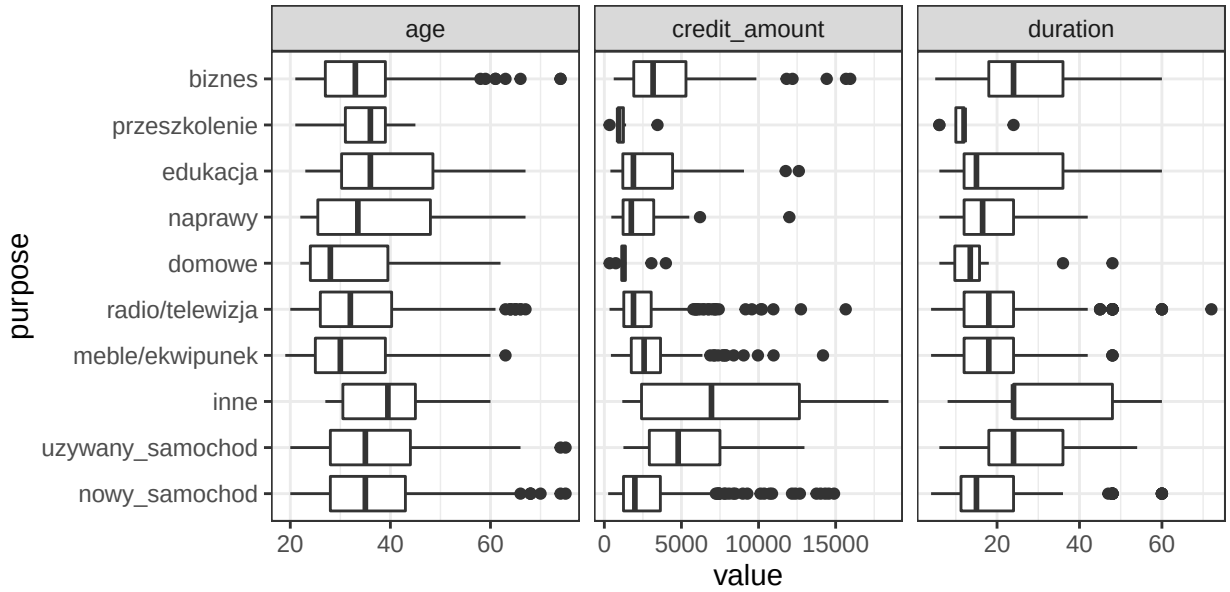
- Niektóre wartości zmiennej `property` (związane z nieruchomościami) są związane z wartościami zmiennej `housing` oraz `credit_amount`.

Ogólnie, powyższe związki wynikają w większości z tego, że każdy z nich traktuje o pewnej własności (`telephone`, `housing`, `property`), które są związane ze stanem konta i ze zdolnością kredytową. Im więcej ktoś posiada, tym chętniej bank udziela kredytu na większe kwoty.



Korelacja pomiędzy zmienną `present_employment` a `age` jest samotłumaczalna - im człowiek starszy, tym więcej miał czasu na pracę. Podobnie z `age` i `present_resident`.

Zależności zmiennych od celu kredytu



- co ciekawe klienci biorą niższe kredyty na nowe samochody niż na używane.
- widać tu wyraźnie korelację wysokości i czasu trwania kredytu.

Inżynieria cech

```
new_columns <- colnames(data_enhanced)[!(colnames(data_enhanced)) %in% colnames(data_preprocessed)]
print(new_columns)
```

```
## [1] "gender"             "retirement_age"     "age_category"
## [4] "never_married"      "employed"           "duration_years"
## [7] "duration_years_cat"
```

Zostały wprowadzone nowe zmienne, utworzone na podstawie starych. Są to:

- `gender` - kobieta/mężyczyzna, na podstawie personal; podział typu singiel/singielka rozbity na prostsze rozróżnienie k/m
- `retirement_age` - True/False, na podstawie age; tutaj uznaliśmy >=65 za wiek emerytalny
- `age_category` - young/middle-aged/old, na podstawie age; przedziały <=39/40-59/>=60
- `never_married` - True/False, na podstaiwe personal; jako że te kategorie nie były najlepsze, np. dla kobiet jedyne kategorie to singielka/[mężata/rozwiedziona/wdowa] podzieliliśmy ludzi na tych którzy nigdy nie byli w związku i na tych co kiedyś byli albo nadal są
- `employed` - True/False, na podstawie present_employement; nie bierzemy uwagi na to ile lat ktoś ma pracę tylko czy faktycznie ma pracę
- `duration_years` - numeryczne, na podstawie duration; jako że duration było w miesiącach przekonwertowaliśmy je na lata
- `duration_years_cat` - `<1/1/1<&<2/2/2<&<3/3/3>`, na podstawie duration_years; podzielone na pełne lata i pomiędzy przy uznaniu że 3+ lata to już jedna kategoria

Wstępne modelowanie - drzewo decyzyjne

Audyt modelu

Na początku stworzymy trzy klasyfikatory za pomocą modelu `CART`, kierując się kryteriami:

- w pierwszym przypadku ograniczymy głębokość drzewa i wyznaczymy minimalne rozbicie danych;
- w drugim drzewie zbadamy wpływ parametru `cp` na postać drzewa;
- trzecie drzewo tworzymy domyślną metodą;

Używamy do tego pakietu `mlr`.

```
## auc.test.mean acc.test.mean ppv.test.mean
##             1             1             1
```

Trening i ocena skuteczności

Trenujemy drzewa decyzyjne na odpowiednich klasyfikatorach.

```
tree1 <- train(tree1_learner, task)
tree2 <- train(tree2_learner, task)
tree3 <- train(tree3_learner, task)

# sprawdźmy:
predict(tree1, newdata = data_enhanced) %>%
  performance(measures = list(auc, acc, ppv)) %>%
  print()
```

```
##       auc       acc       ppv
## 0.7213857 0.7310000 0.5654008
```

```
predict(tree2, newdata = data_enhanced) %>%
  performance(measures = list(auc, acc, ppv)) %>%
  print()
```

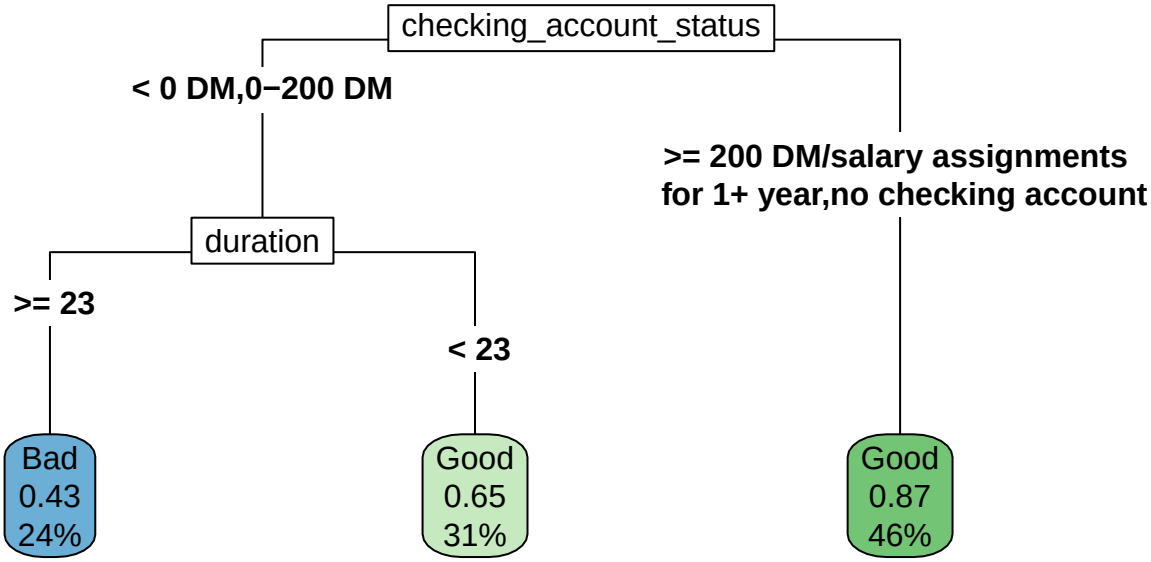
```
##       auc       acc       ppv
## 0.7309048 0.7480000 0.6224490
```

```
predict(tree3, newdata = data_enhanced) %>%
  performance(measures = list(auc, acc, ppv)) %>%
  print()
```

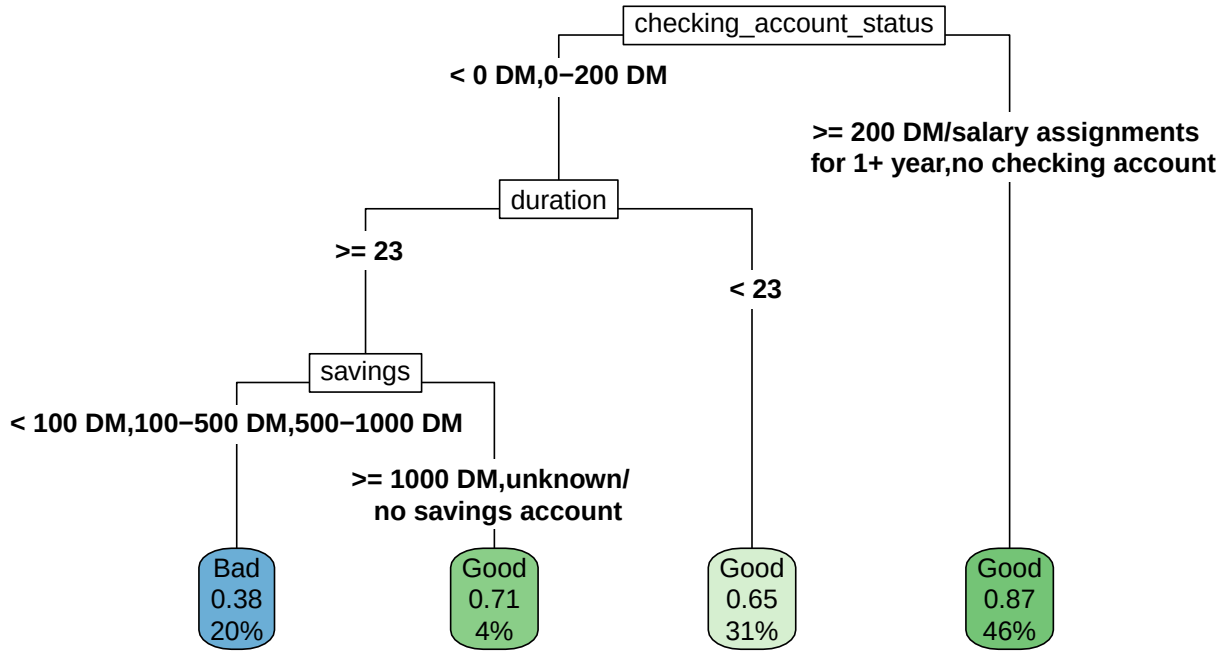
```
##       auc       acc       ppv
## 0.7805095 0.7970000 0.6995885
```

Wizualizacja

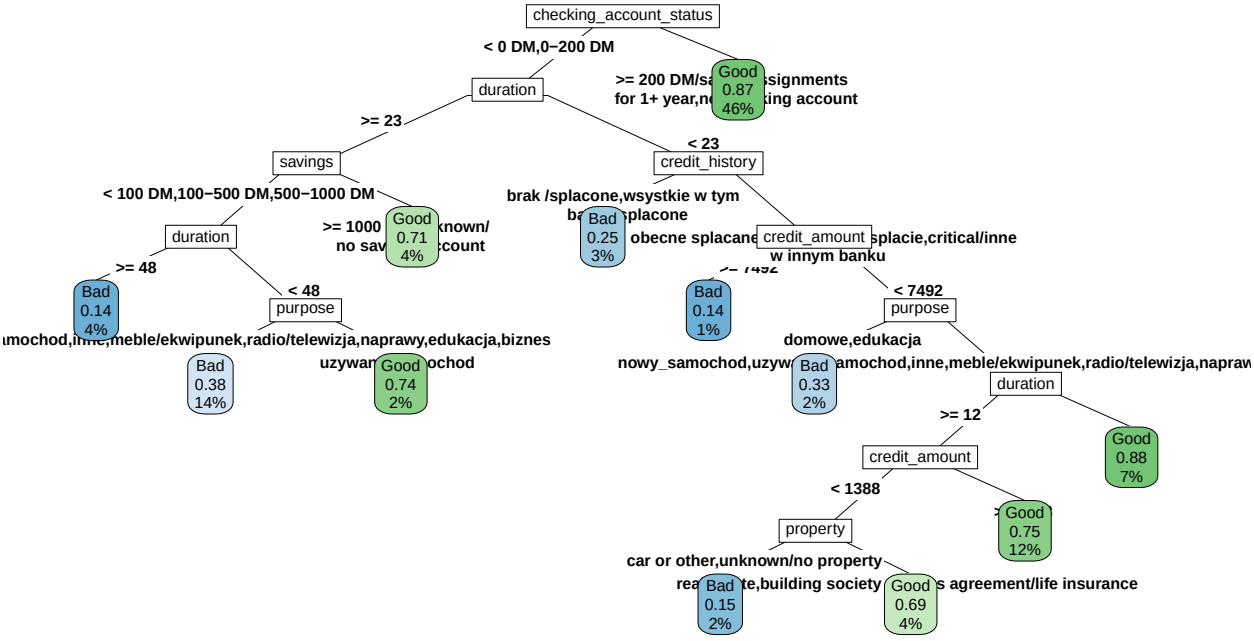
Pierwsze drzewo decyzyjne jest dosyć proste ze względu na nałożone ograniczenia.



Drugie drzewo ma już trzy poziomy, z czego pierwsze dwa są takie same jak w powyższym.



Jeśli klasyfikator pozostawimy z domyślnymi parametrami, tworzy on już bardziej rozbudowane drzewo.



Omówienie

Okazuje się, że wszystkie trzy drzewa jako jedne z ważniejszych zmiennych wyznaczają `checking_account_status` oraz `duration`. Zmienne te umożliwiają drzewom tzw. optymalny podział dla najskuteczniejszej predykcji. Pierwsze dwa poziomy są wszędzie takie same, a kolejne drzewa są rozszerzeniami poprzednich. Z oczywistych względów ostatnie drzewo ma największą skuteczność przewidywania, lecz co ciekawe nie jest ona o wiele większa nawet od dwu poziomowego drzewa (różnica accuracy to około 6%). Na testowane drzewa decyzyjne nakładaliśmy te ograniczenia ze względu na podatność do overfittingu tego modelu.

Częściowe wyjaśnienie - ważność zmiennych

Wybiegnijmy trochę w przyszłość i zobaczmy, co o ważności zmiennych myślą modele `ranger` (las losowy) oraz `gbm`.

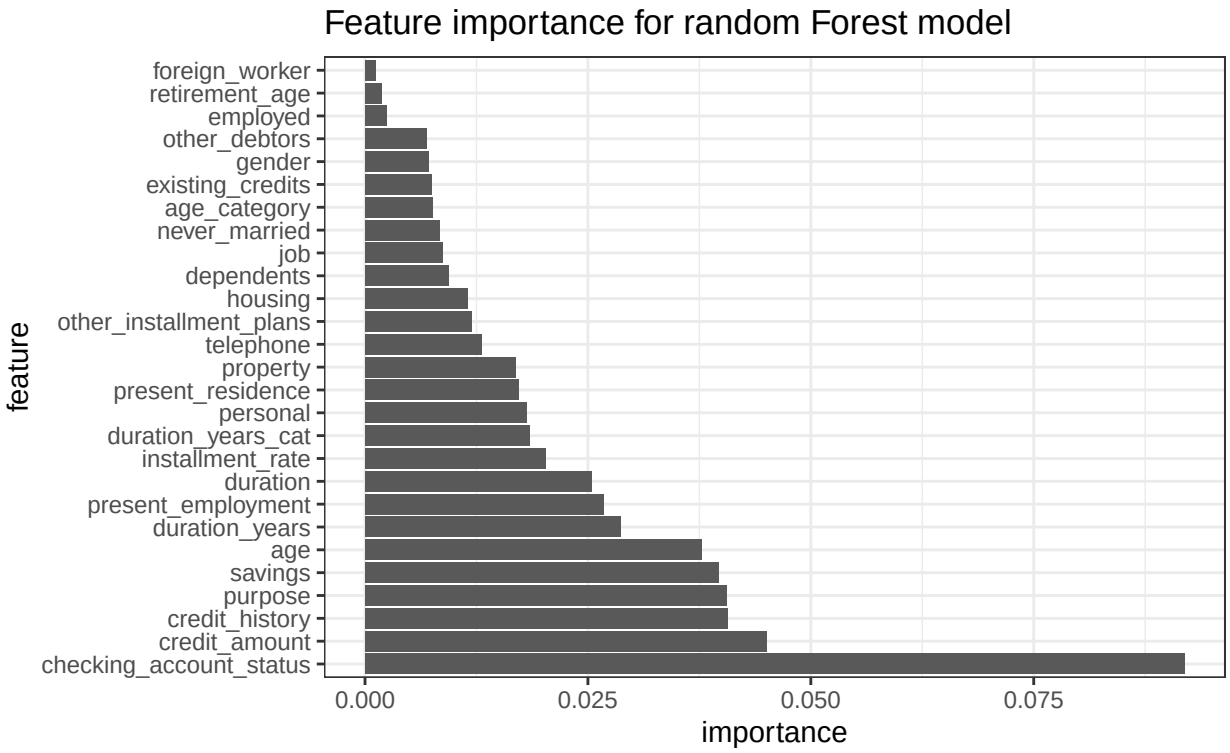
```
# ranger CV performance:
print(r1$aggr)
```

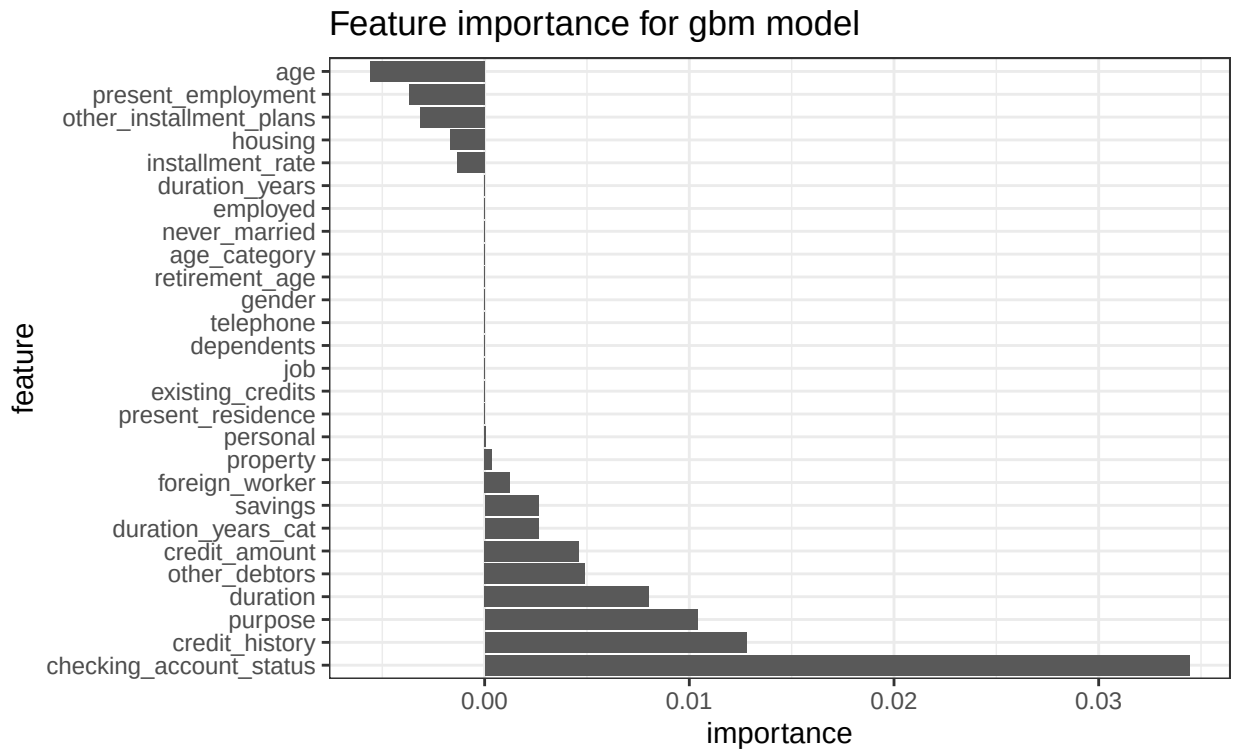
```
## auc.test.mean acc.test.mean ppv.test.mean
##     0.7916138     0.7699765     0.7260128
```

```
# gbm CV performance:
print(r2$aggr)
```

```
## auc.test.mean acc.test.mean ppv.test.mean
##     0.7862321     0.7529724     0.6489823
```

```
## Distribution not specified, assuming bernoulli ...
```





Jak widzimy, bardziej skomplikowane modele podzielają zdanie drzewa decyzyjnego o ogromnym znaczeniu zmiennej `checking_account_status`. Zmienne `duration` oraz `savings` również się pojawiają jako istotne. Zmienne `credit_amount` oraz `credit_history`, wykorzystywane w skomplikowanym drzewie nr 3, należą do najistotniejszych.

Pominięcie innych zmiennych uznanych za istotne możemy uznać za poświęcenie w imię prostoty zbudowanego modelu.

Porównanie z bardziej skomplikowanymi modelami

lost credit	lost potential	correct credit	correct decline	model	accuracy overall	accuracy for troublesome
526'771	346'330	1'743'490	654'667	rpart	0.748	0.000
512'058	269'341	1'820'479	669'380	gbm	0.790	0.409
113'521	12'264	2'077'556	1'067'917	ranger	0.954	0.825

`Lost credit` odnosi się do łącznej sumy kredytów klientów niepoprawnie zaklasyfikowanych jako 'dobrych'. `Lost potential` odnosi się za to do kredytów klientów niepoprawnie zaklasyfikowanych jako 'złych'. `Correct credit` i `correct decline` to odpowiednio kredyty klientów poprawnie zaklasyfikowanych jako 'dobrych' i jako 'złych'. `Troublesome` odnosi się do rekordów, z które model `rpart` zaklasyfikował niepoprawnie.

Prosty model drzewa decyzyjnego okazuje się niewiele słabszy od modelu `gbm`, zachowując przy tym czytelność i przejrzystość. Oczywiście, jest słabszy od modelu lasu losowego, ale tego się spodziewaliśmy - uproszczenie modelu pociąga za sobą gorszą skuteczność.

Co ciekawe, więcej zmiennych niekoniecznie znaczy lepiej dla modelu - jeśli te zmienne są skorelowane z już istniejącymi, nie niosą ze sobą nowej informacji i model nie uznaje ich za istotne.

Trenowanie pozostałych modeli, strojenie hiperparametrów i wybór najlepszego

Łącznie wytrenowano 5 modeli. Jeden już widzieliśmy (drzewo decyzyjne). 4 pozostałe to:

- Las losowy - `ranger`
- Gradient boost - `gbm`
- ADA boost - `ada`
- XGboost - `xgboost`

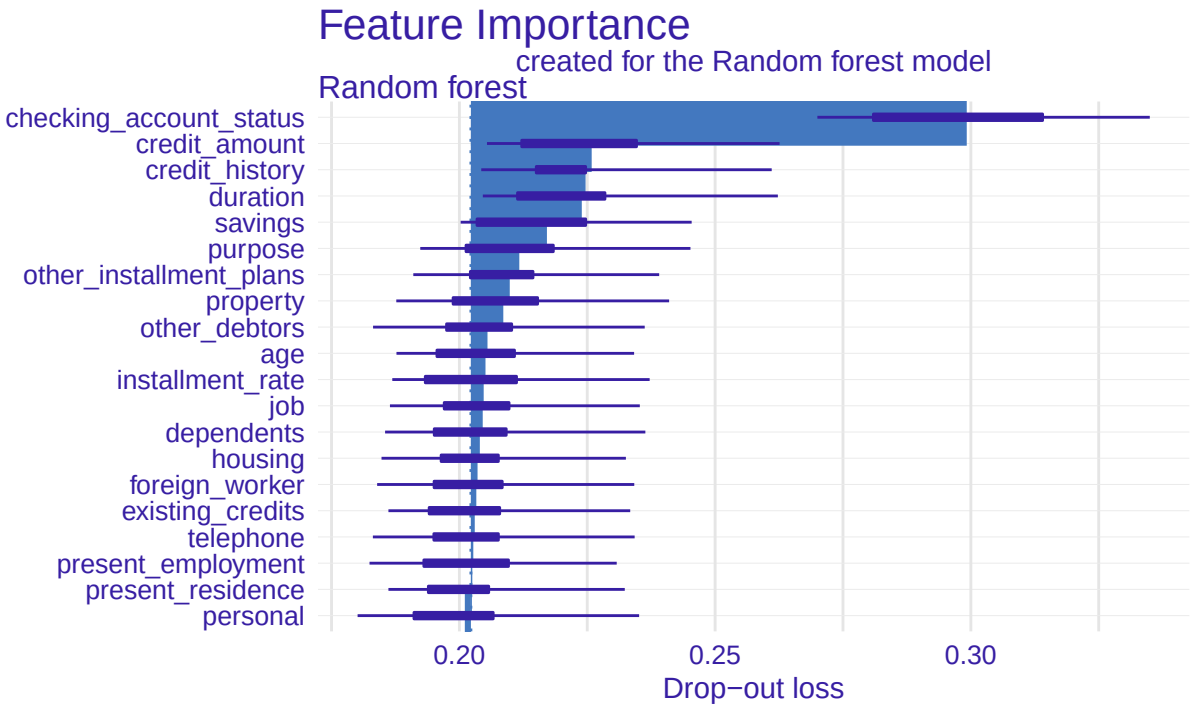
W każdym z powyższych czterech hiperparametry były strojone metodą RandomSearch.

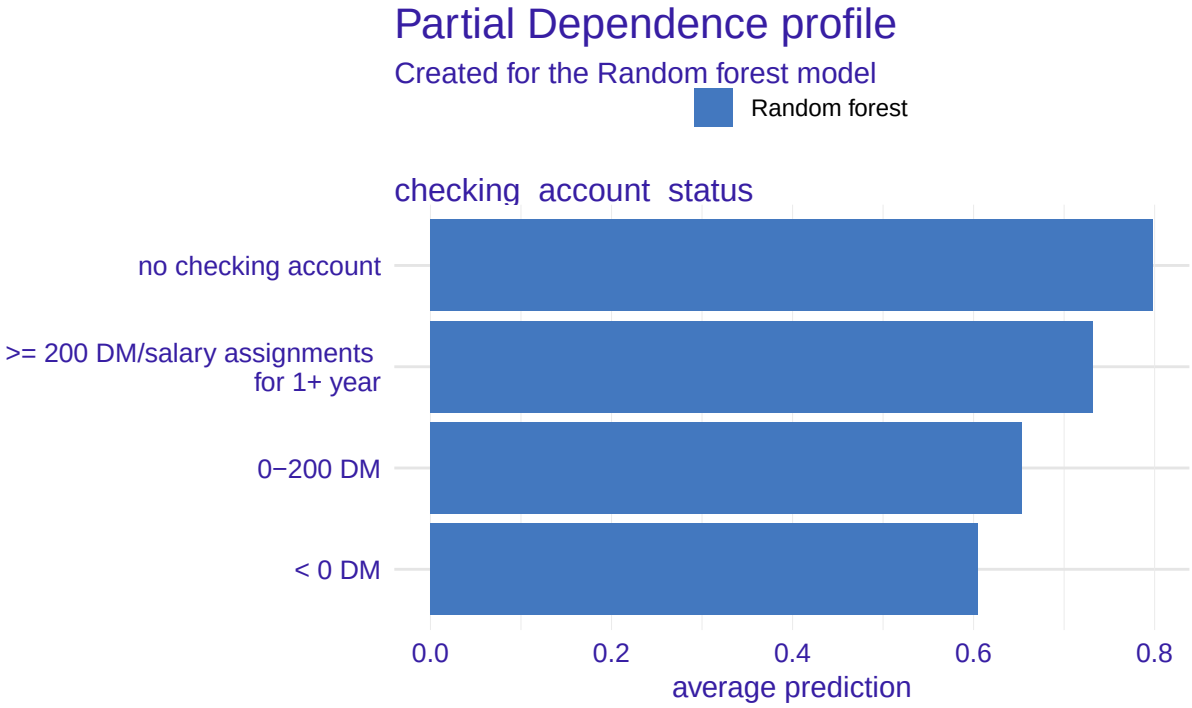
Porównanie

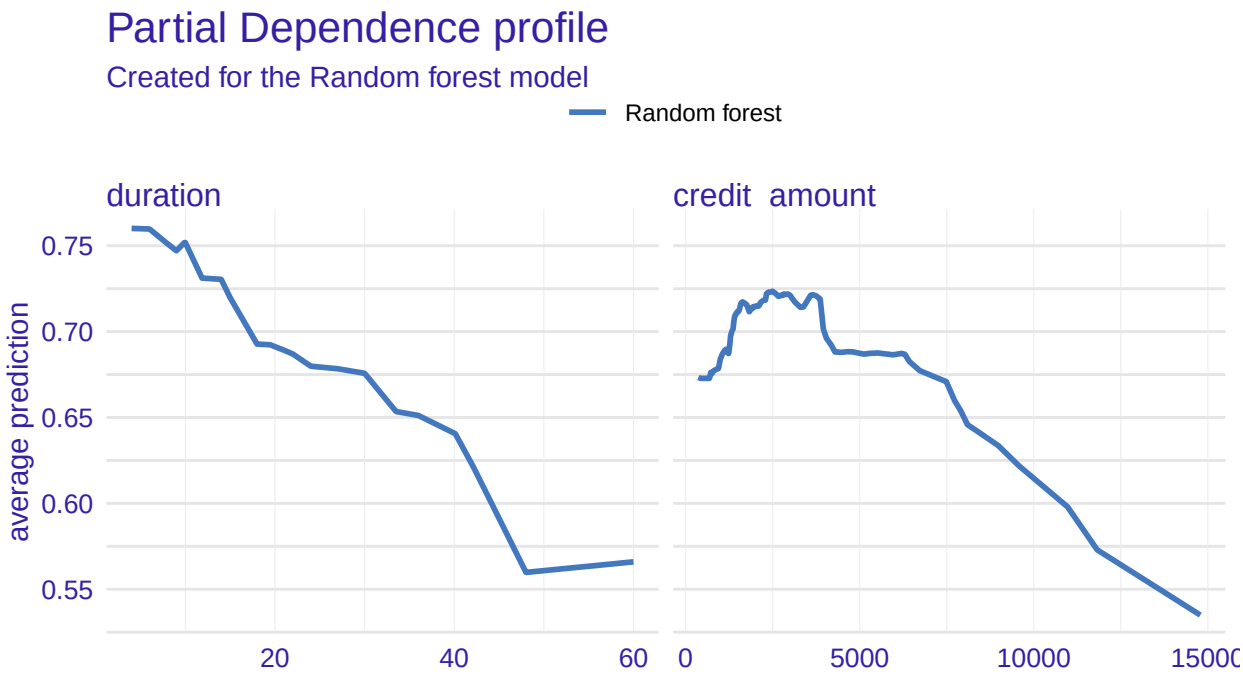
auc	acc	mcc	ppv	Model
0.7954963	0.7733333	0.4410049	0.7800000	Random Forest
0.7919220	0.7733333	0.4411851	0.7592593	Gradient Boost
0.7740502	0.7766667	0.4571041	0.7101449	ADA Boost
0.7691993	0.7566667	0.4081994	0.6619718	XGBoost

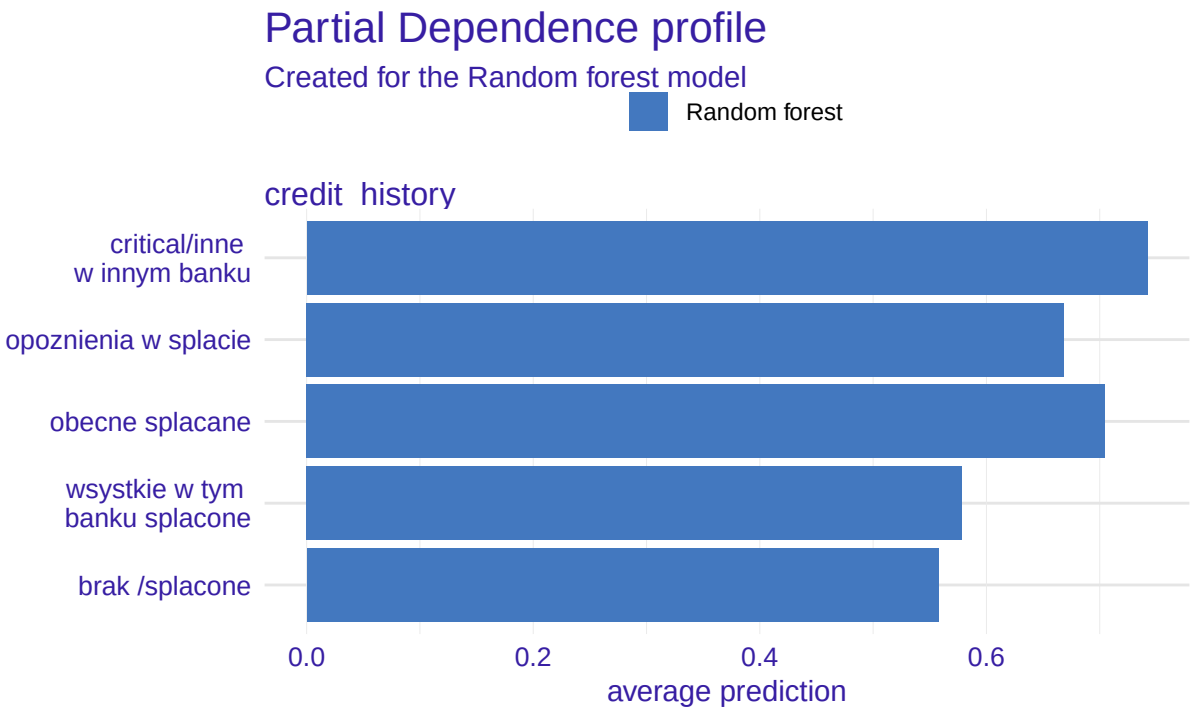
Jak widzimy, przy wzięciu pod uwagę kilku różnych metryk, najlepszy okazał się zwykły las losowy - mimo zbliżonych wyników.

Interpretacja najlepszego modelu przy użyciu pakietu `DALEX`









Podsumowanie

Po użyciu pakietu `DALEX` możemy w większym stopniu zinterpretować model `ranger` i wyjaśnić jego poszczególne komponenty. Najważniejszymi zmiennymi przy określaniu typu klienta okazały się:

- `checking_account_status` - stan rachunku bieżącego
- `credit_amount` - wysokość kredytu
- `credit_history` - podsumowanie historii kredytów

- `savings` - oszczędności

Najczęstsze przewidywania modelu o tym, że klient jest 'dobry' miały miejsce, gdy zmienna `checking_account_status` przyjmowała wartość `no_checking_account` i `>=200DM/salary assignments for 1+ year`.

Przy zależnościach zmiennych `duration` oraz `credit_amount` możemy zauważyć, że im dłuższy jest czas trwania kredytu tym częściej klienci oceniani są jako 'źli'. Wyskość kredytu nie powinna być zbyt niska, ani zbyt wysoka.

Jeśli z kolei popatrzymy na historię kredytów, to co ciekawe najniższą średnią predykcji mają klienci, którzy mają wszystkie kredyty spłacone. Oznacza to, że klienci którzy nadal spłacają kredyt albo nawet mają opóźnienia w spłacaniu klasyfikowani są jako 'dobrzy' częściej niż ci ze spłaconymi kredytami. Jest to bardzo ciekawe zjawisko bardzo sprzeczne z intuicją.

Session Info

```
# session info
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Polish_Poland.1250  LC_CTYPE=Polish_Poland.1250
## [3] LC_MONETARY=Polish_Poland.1250 LC_NUMERIC=C
## [5] LC_TIME=Polish_Poland.1250
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] xgboost_1.0.0.2    ada_2.0-5          ranger_0.12.1      gbm_2.1.5
##  [5] rpart.plot_3.0.8   rpart_4.1-15       OpenML_1.10        mlr_2.17.1
##  [9] ParamHelpers_1.13  dplyr_0.8.5        ggplot2_3.3.0      DataExplorer_0.8.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.5  xfun_0.11         reshape2_1.4.3    purrr_0.3.3
##  [5] splines_3.6.3     lattice_0.20-38   colorspace_1.4-1  htmltools_0.4.0
##  [9] yaml_2.2.0        survival_3.1-8    XML_3.99-0.3      rlang_0.4.2
## [13] pillar_1.4.2      glue_1.3.1        withr_2.1.2       plyr_1.8.4
## [17] networkD3_0.4     lifecycle_0.1.0   stringr_1.4.0     munsell_0.5.0
## [21] gtable_0.3.0      htmlwidgets_1.5.1 evaluate_0.14     memoise_1.1.0
## [25] labeling_0.3      knitr_1.26        parallelMap_1.4   parallel_3.6.3
## [29] curl_4.2          highr_0.8         Rcpp_1.0.3        scales_1.1.0
## [33] backports_1.1.5   checkmate_2.0.0   jsonlite_1.6      farver_2.0.1
## [37] gridExtra_2.3     fastmatch_1.1-0   digest_0.6.23     stringi_1.4.3
## [41] BBmisc_1.11       grid_3.6.3        tools_3.6.3       magrittr_1.5
## [45] tibble_2.1.3      crayon_1.3.4      pkgconfig_2.0.3   Matrix_1.2-18
## [49] data.table_1.12.8 assertthat_0.2.1  rmarkdown_2.1     httr_1.4.1
## [53] R6_2.4.1          igraph_1.2.4.2    compiler_3.6.3
```