Wybrany zbiór danych

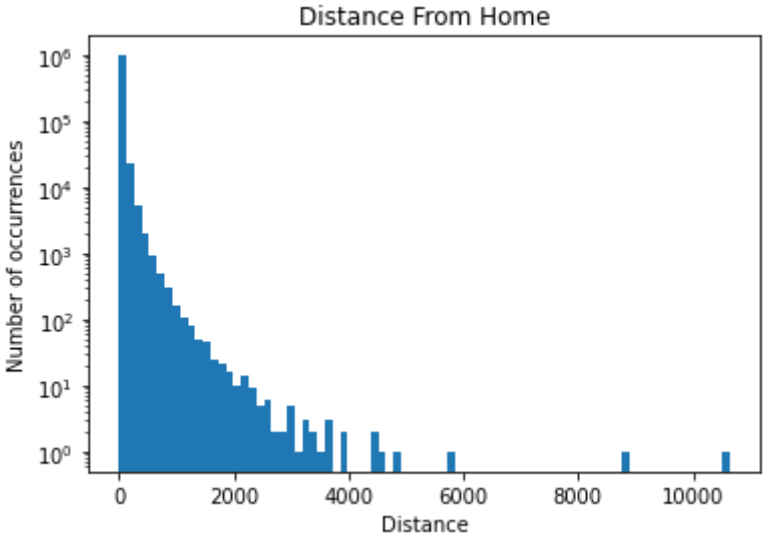
1. Opis zbioru

Wybrany do analizy zbiór danych o nazwie “Credit Card Fraud” zawiera dane dotyczące oszustw związanych z kartami kredytowymi. Źródło danych dla tego zbioru jest nieznane. Motywacją wybrania tego zbioru jest fakt, że dotyczy on problemu który staje się coraz istotniejszy wraz ze wzrostem płatności bezgotówkowych.

Każdy wiersz danych zawiera zawiera 8 atrybutów opisujących transakcję podzielony jest na 2 klasy (transakcja uznana za oszustwo lub nie). Dane liczbowe zostały zanonimizowane (autor nie podał ich jednostki) dlatego odczytanie dokładnej odległości lub czasu dla danych ciągłych nie jest możliwe. Zbiór zawiera 1 000 000 przypadków.

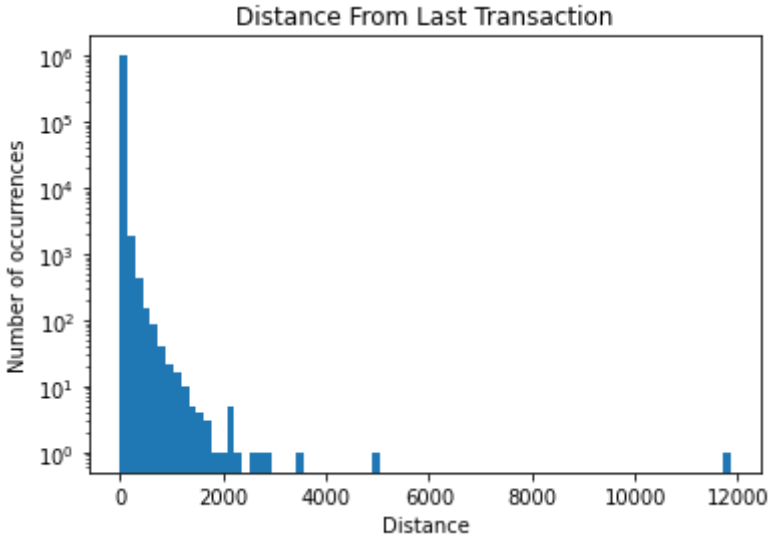
2. Charakterystyka atrybutów

Distance From Home - odległość od miejsca zamieszkania, do miejsca w którym miała miejsce transakcja.



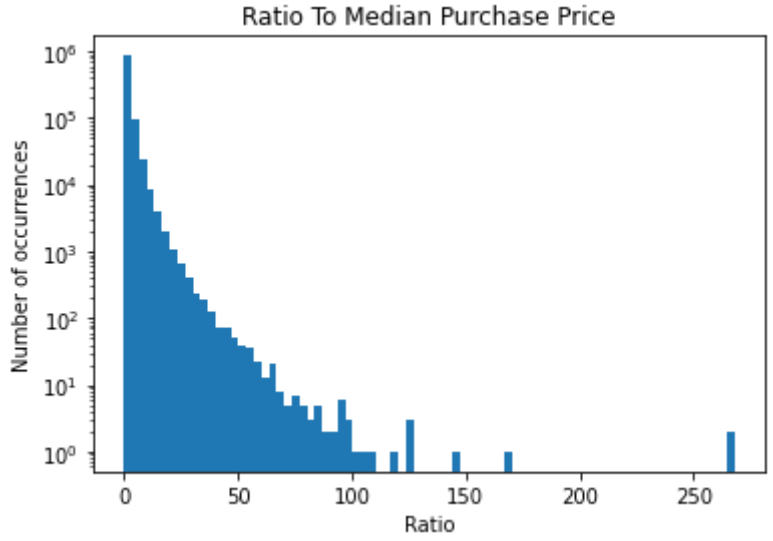
Średnia	26.6288
Minimum	0.0049
Maksimum	10632.7237
Odchylenie standardowe	65.3908

Distance From Last Transaction - odstęp w czasie od ostatniej transakcji z wykorzystaniem tej samej karty.



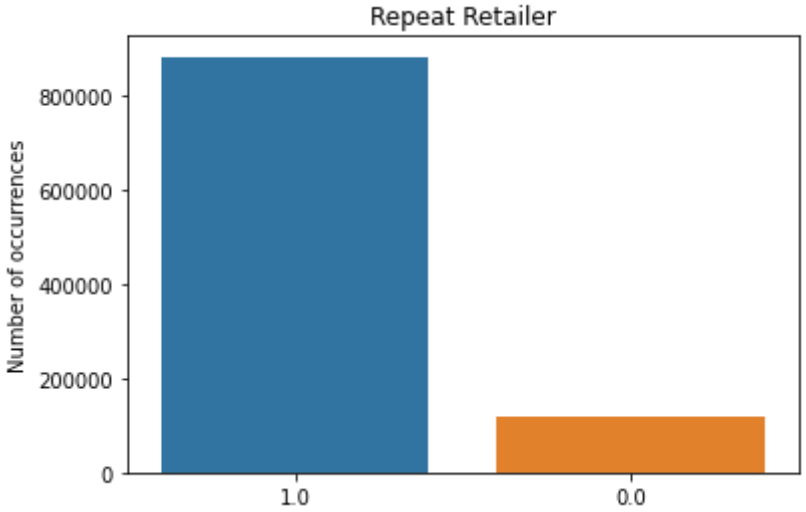
Średnia	5.0365
Minimum	0.000118
Maksimum	11851.1046
Odchylenie standardowe	25.8431

Ratio To Median Purchase Price - stosunek wartości transakcji do mediany wszystkich transakcji z wykorzystaniem tej samej karty.



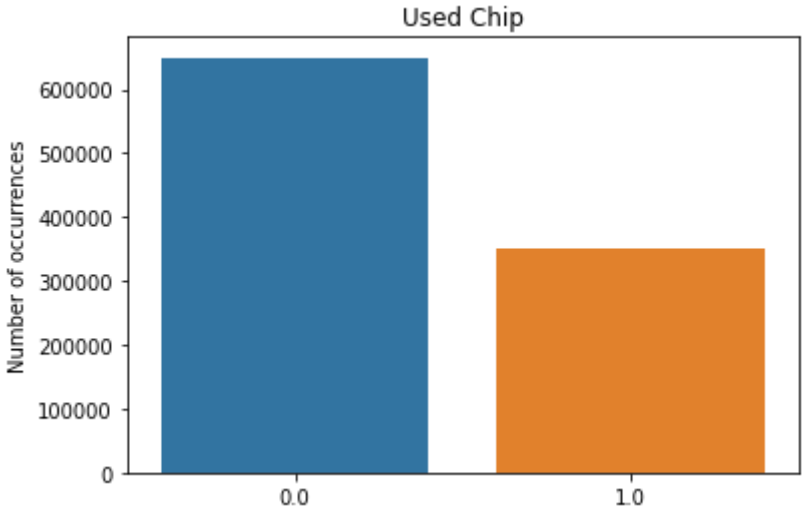
Średnia	1.8242
Minimum	0.0044
Maksimum	267.8029
Odchylenie standardowe	2.7996

Repeat Retailer - Informacja czy transakcja miała miejsce u tego samego sprzedawcy co poprzednia transakcja.



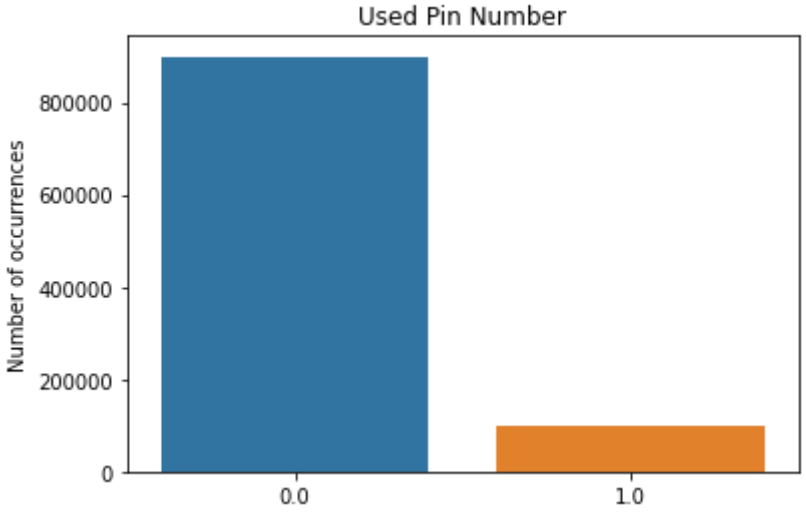
False	88.15%
True	11.85%

Used Chip - Informacja czy transakcja odbyła się za pomocą chipa (karta kredytowa).



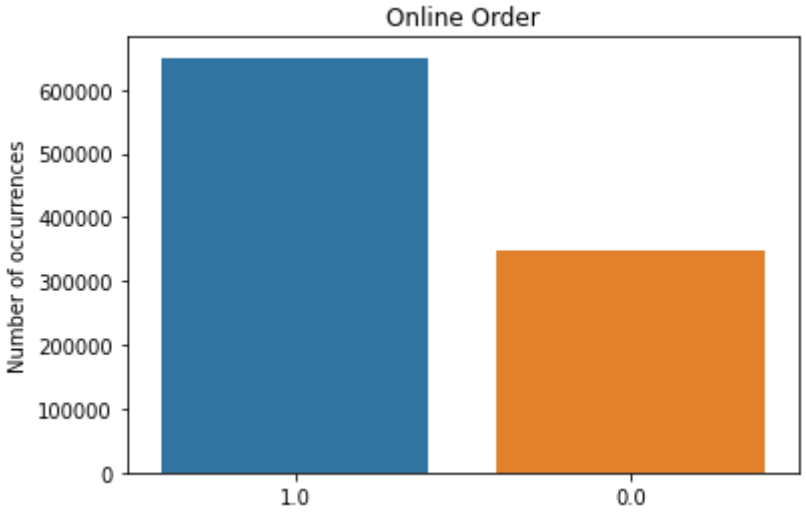
False	64.96%
True	35.04%

Used Pin Number - Informacja czy transakcja została wykonana z użyciem numeru PIN.



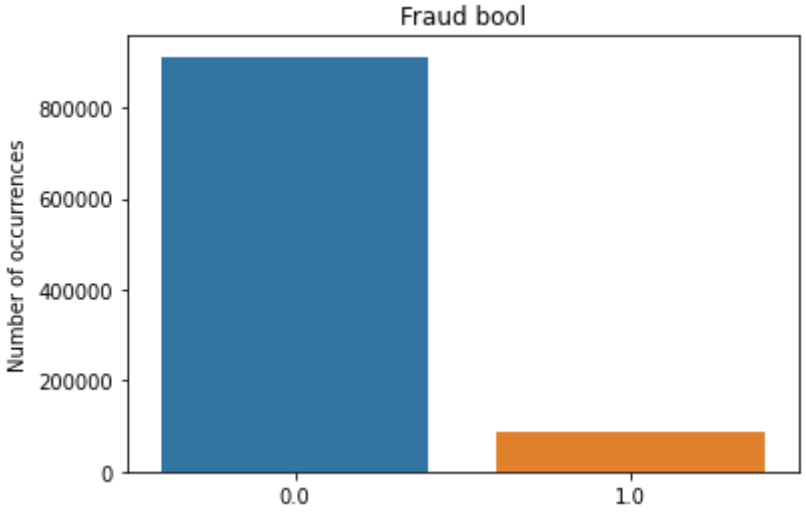
False	89.94%
True	10.06%

Online Order - Informacja czy transakcja jest została wykonana online.



False	65.06%
True	34.94%

Fraud bool - Informacja czy transakcja została uznana za oszustwo.



False	91.26%
True	8.74%

3. Wnioski

- Dane są niezbalansowane, niecałe 9% stanowi przypadki pozytywne. Trywialny klasyfikator osiągnął by na tym zbiorze ponad 90% trafności co pokazuje, że do oceny modeli należy wybrać inne metryki.
- Zdecydowana większość (89.94%) transakcji wykonywana jest bez użycia numeru PIN.
- Zdecydowana większość (88.15%) transakcji jest wykonywana powtórnie u tego samego sprzedawcy.
- Transakcje internetowe stanowią mniejszość (34.94%) w rozpatrywanym zbiorze.
- Znaczna część transakcji odbywają się w pobliżu domu właściciela karty.
- Prawie wszystkie transakcje odbywają się w niedługim odstępie czasowym od poprzedniej.

4. Przykładowe przypadki:

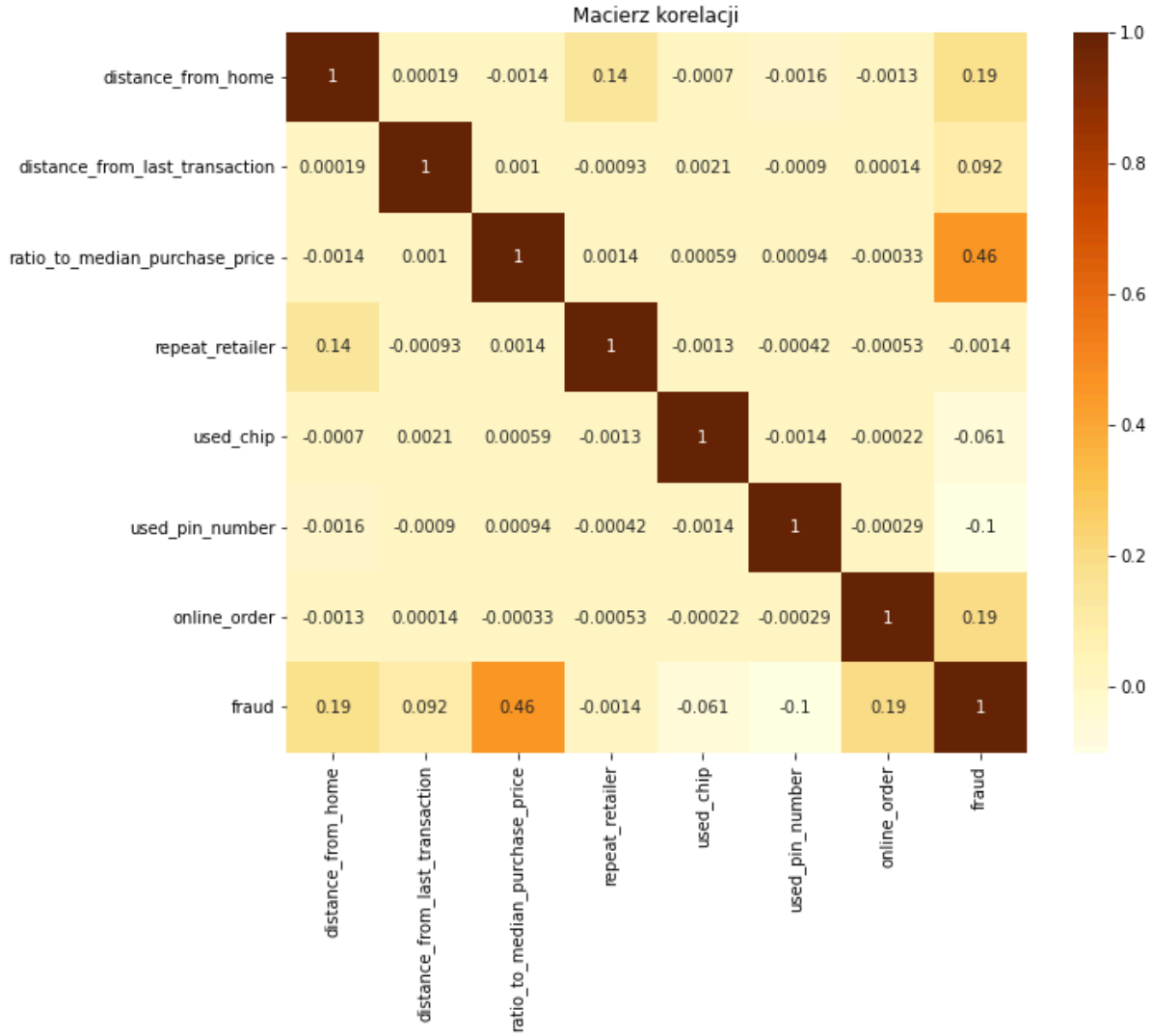
Distance From Home	**Distance From Last Transaction**	**Ratio To Median Purchase Price**	**Repeat Retailer**	**Used Chip**	**Used Pin Number**	**Online Order**	**Fraud bool**
2.132	56.372	6.358	1.0	0.0	0.0	1.0	1.0
13.592	0.241	1.370	1.0	1.0	0.0	1.0	0.0
2.248	5.600	0.363	1.0	1.0	0.0	1.0	0.0

W zaprezentowanych przypadkach jedynie pierwszy z nich został oznaczony jako oszustwo.

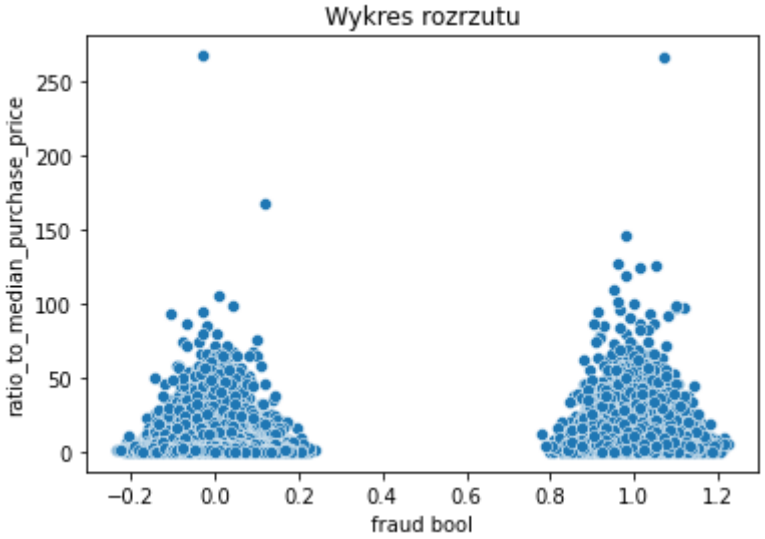
Z atrybutów można odczytać, że odległość od domu płacącego w pierwszym i trzecim przypadku była zbliżona.
Można również odczytać, że dla drugiego przypadku transakcja odbyła się niedługo po poprzedniej, popiera to mała wartość atrybutu “Distance From Last Transaction” (średnia dla tego atrybutu jest większa niż 5 a tu mamy wartość zbliżoną do zera).
Z wartości atrybutu “Ratio To Median Purchase Price” można odczytać, że dla pierwszego przypadku wartość transakcji była ponad 6 krotnie wyższa niż mediana transakcji dla tej karty, natomiast dla trzeciej karty prawie 3 krotnie niższa.
Wszystkie zaprezentowane transakcje odbyły się online i każda z nich była powtórną transakcją u tego samego sprzedawcy.

Wstępna analiza zbioru

Macierz korelacji:



Wykres rozrzutu dla najbardziej skorelowanych atrybutów:



Do zmiennej fraud bool dodany został szum o odchyleniu standardowym 0.05, aby umożliwić odczytanie wykresu.

Próby predykcji

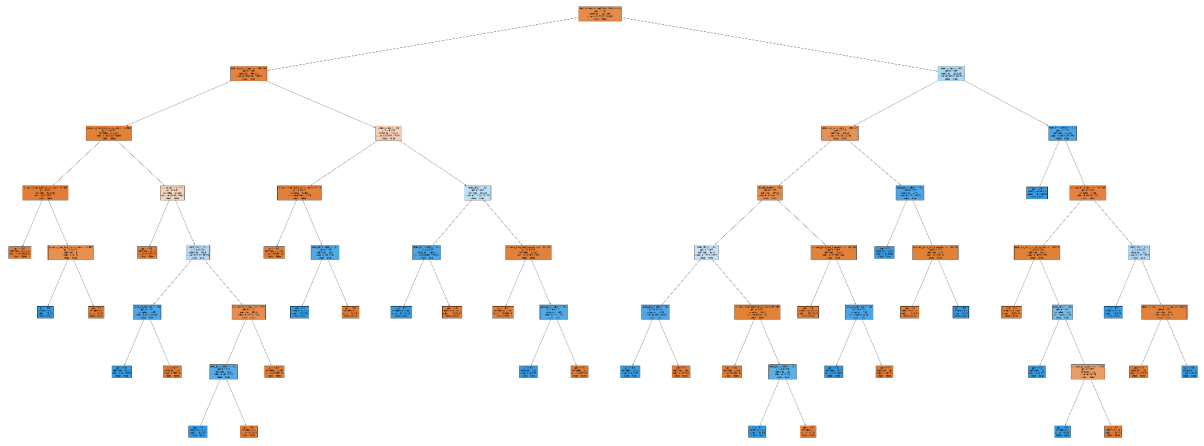
1. Podział na zbiór testowy i treningowy

Podziału dokonałem przy użyciu funkcji *train_test_split* z biblioteki sklearn z proporcją 8:2. Z racji na niezbalansowanie klasy decyzyjne do podziału użyłem losowania warstwowego, które umożliwiło zachowanie równomiernego podziału klas między zbiór testowy i treningowy.

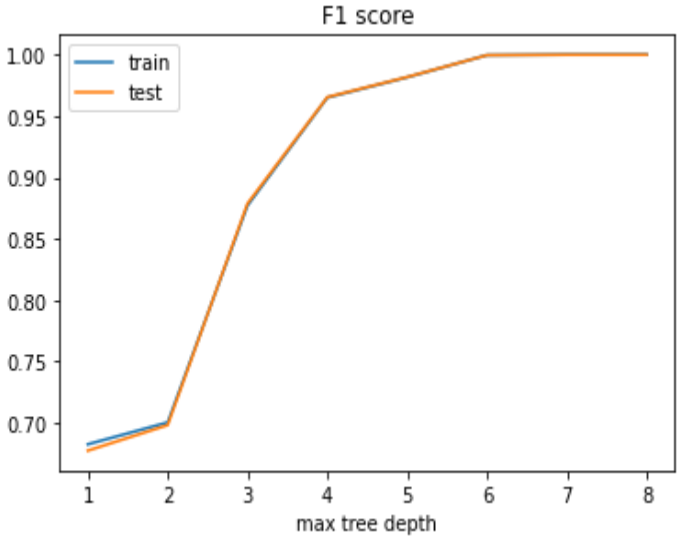
2. Drzewo decyzyjne

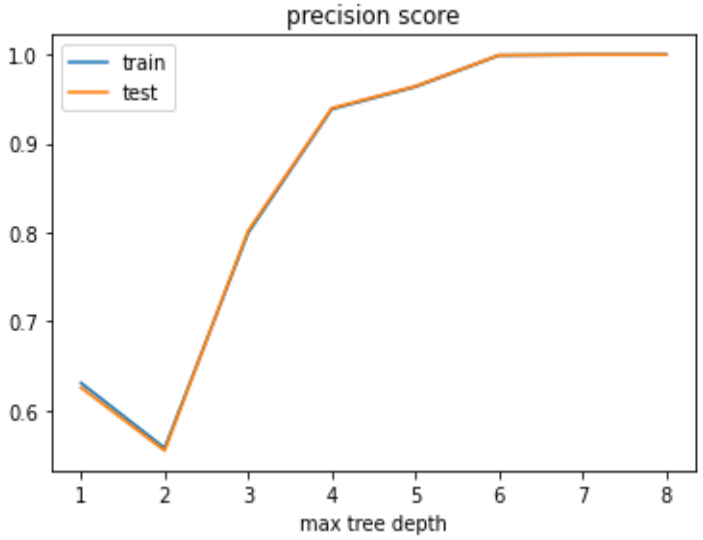
Do testów użyłem drzewa decyzyjnego zaimplementowanego w bibliotece sklearn. Najpierw wykonałem test dla drzewa z domyślnymi parametrami i bez limitu głębokości, na pełnym zbiorze. Zbiór udało się opisać już dla 8 poziomowego drzewa, co rodzi pewne nadzieje, że uda się osiągnąć zadowalające wyniki.

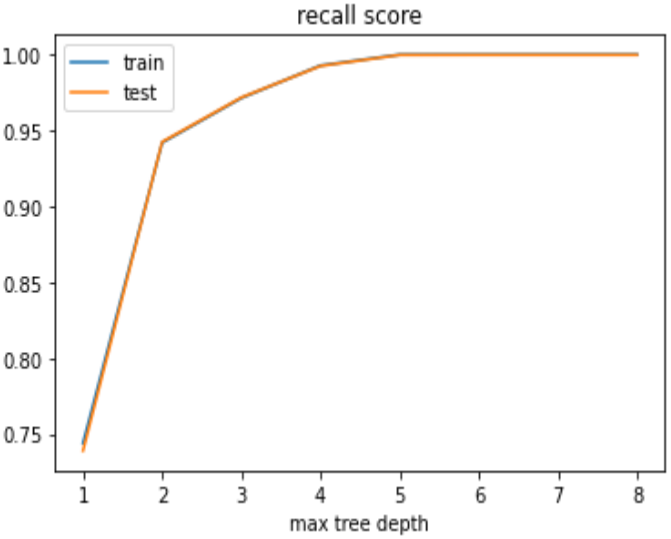
Drzewo bez maksymalnej głębokości utworzone na pełnym zbiorze danych:

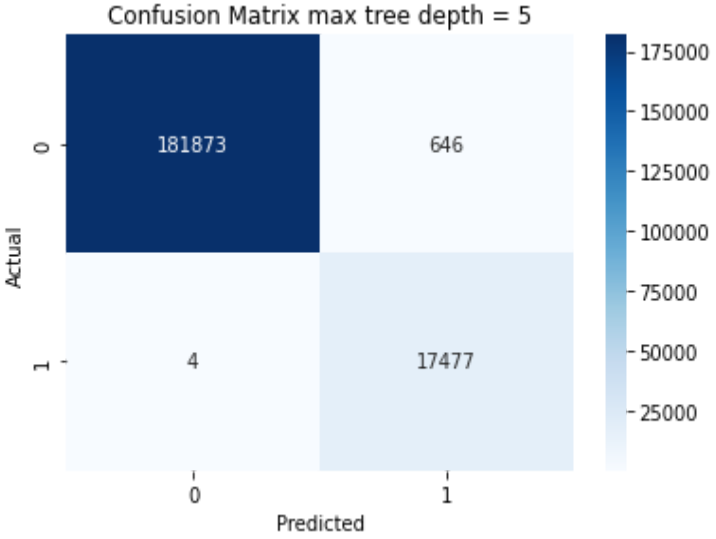


Dodatkowe testy pokazały, że już dla maksymalnej głębokości równej 5, udało się osiągnąć zadowalające wyniki (w przybliżeniu 1) na każdym z kryteriów. Ciekawy wydaje się fakt, że spadek metryk na zbiorze testowym jest niemal niewidoczny. Jako kryterium podziału użyłem entropii a minimalną liczbę przypadków w liściu ustawiłem na domyślną wartość 1. Dodatkowo, aby zminimalizować efekt niezbilansowania ustawiłem 10 krotnie wyższą wagę klasy pozytywnej.

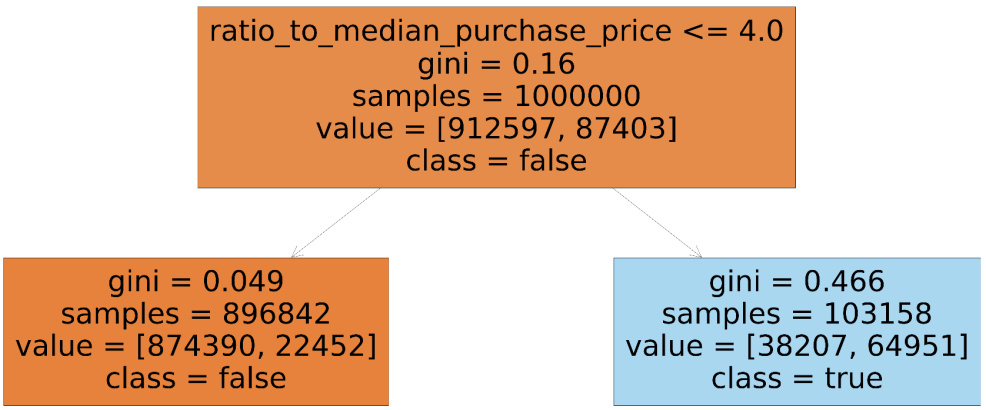




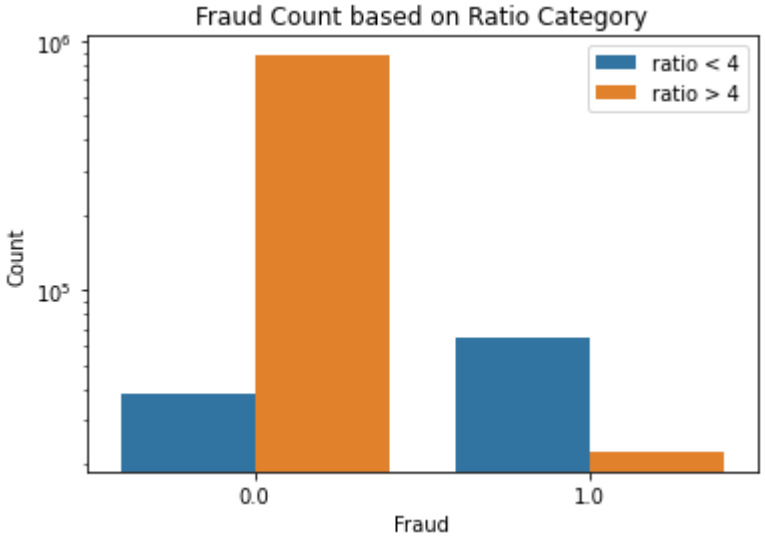




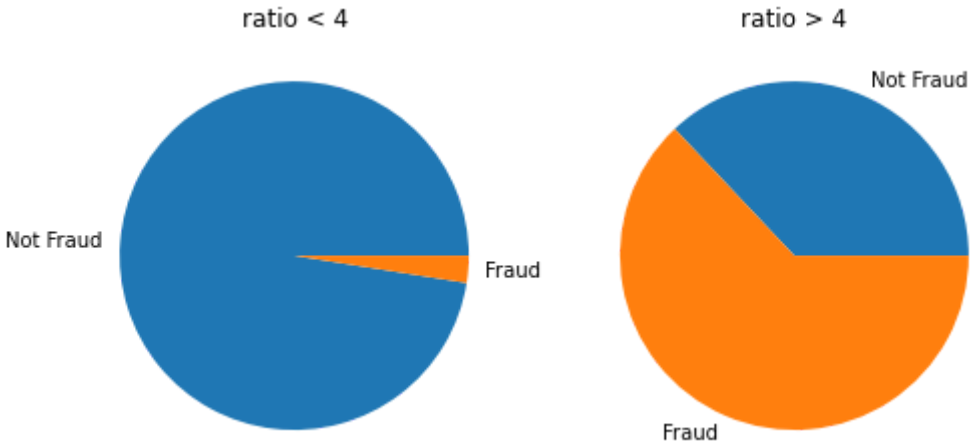
Z budowy drzewa można wyciągnąć dodatkowe informacje apropo danych. W korzeniu drzewa znajduje się warunek Dzielący dane na podstawie stosunku kwoty do mediany.



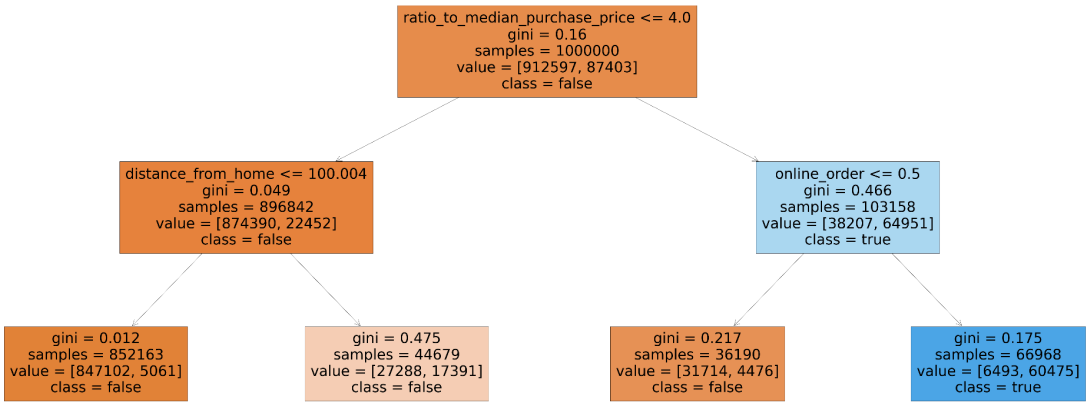
Kiedy narysujemy ten podział ze względu na klasę decyzyjną wyraźnie widać zależność:

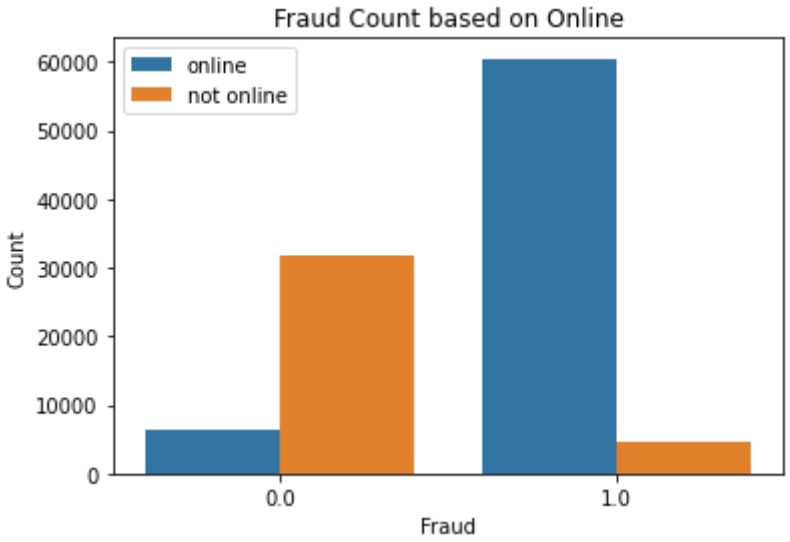


Pokazuje to że płatności które są co najmniej 4 razy większe niż mediana częściej są oszustwem (stanowią wówczas 63%).



Przy następnym rozgałęzieniu można zauważyć, że drzewo sprawdza czy transakcja odbyła się z użyciem internetu (prawa strona drzewa).





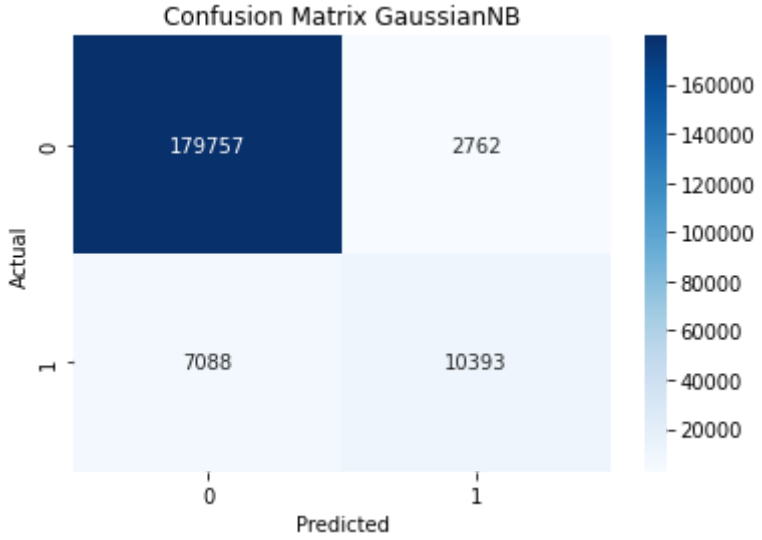
Z tej zależności łatwo można odczytać, że większość dużych transakcji wykonanych przy użyciu internetu jest oszustwem.

3. Naiwny klasyfikator bayesa

Do testów użyłem gaussowskiego naiwnego klasyfikatora bayesa zaimplementowanego w bibliotece sklearn. Dla domyślnych parametrów uzyskałem następujące parametry metryk:

F1	precision	recall	accuracy
0.6785	0.79	0.5945	0.9507

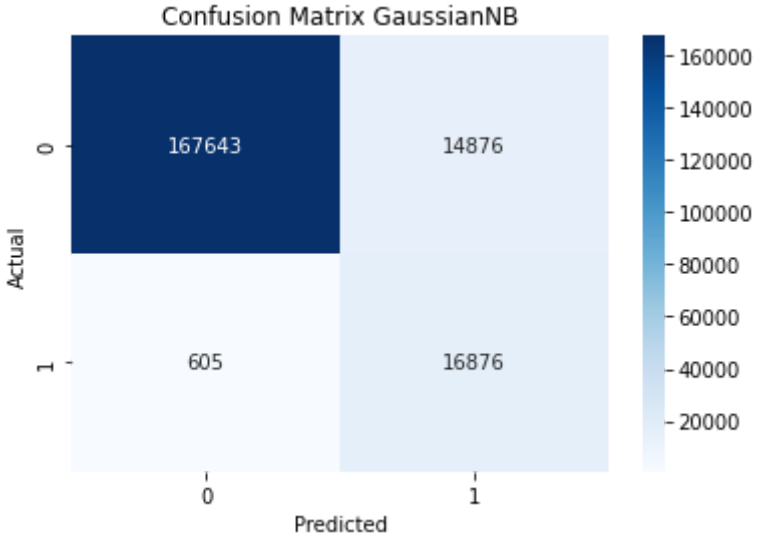
Model osiągnął trafność predykcji na poziomie 95%, jednak dzięki macierzy pomyłek widać, że większość błędów popełnianych jest na klasie mniejszościowej. Potwierdza to intuicję, że trafność jest słabą metryką dla tego problemu.



Jedyny parametr tego klasyfikatora który zmieniłem to *var_smoothing* (z domyślnego 1e-09 do 1e-07) aby uniknąć wartości zerowych. Używając wygładzania danych zaimplementowanego imblearn udało się poprawić metrykę *recall*, kosztem pogorszenia *precision*. W tym przypadku minimalnie polepszył się wynik dla F1.

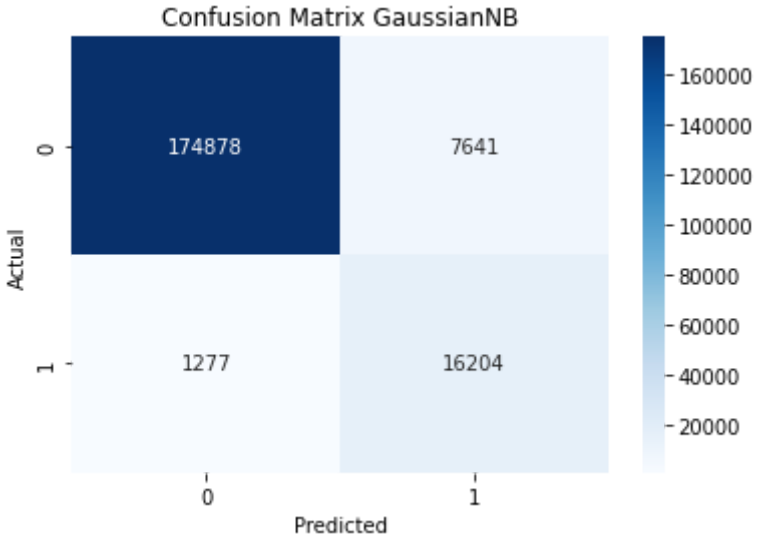
F1	precision	recall	accuracy

0.6856	0.5315	0.9654	0.9226



Dodatkową poprawę metryk przyniosło użycie połączenia wygładzania z Tomek links zaimplementowanego w imblearn pod nazwą *SMOTETomek.* W tym przypadku konieczne okazało się dalsze zwiększenie parametru *var_smoothing* do finalnej wartości 1e-07.

F1	precision	recall	accuracy
0.7842	0.6796	0.9269	0.9554

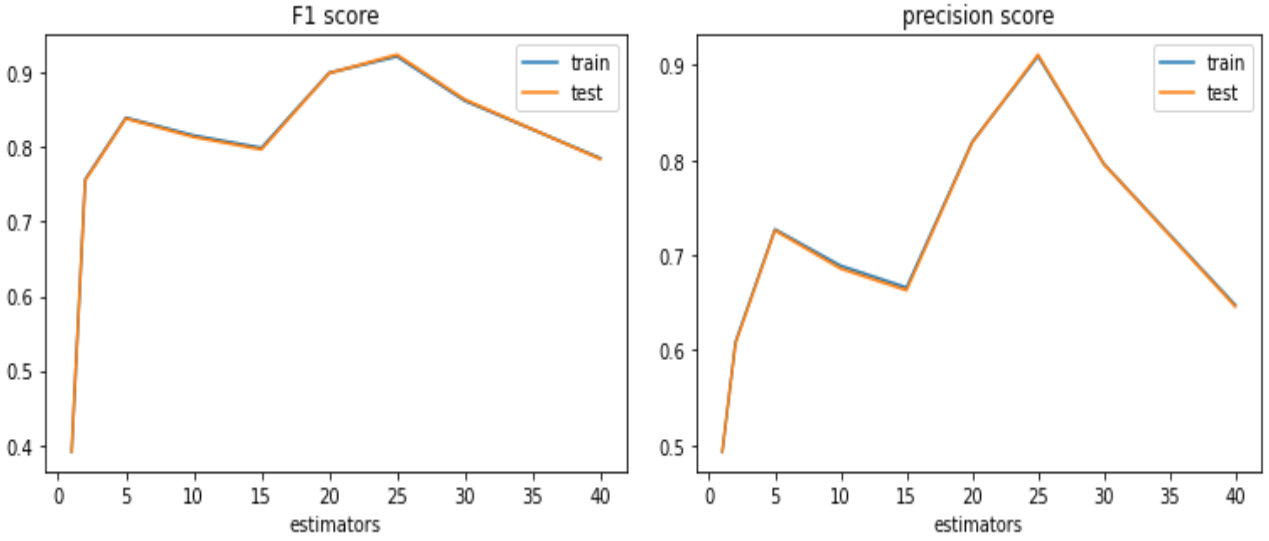


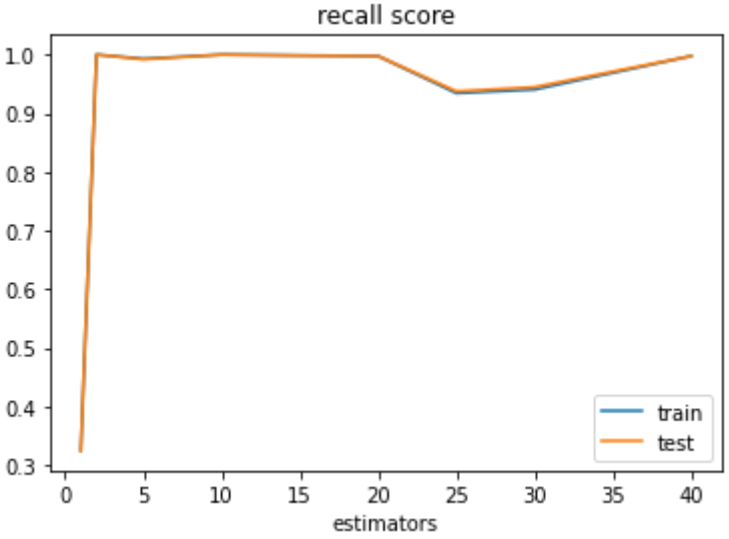
W przypadku naiwnego klasyfikatora gaussowskiego wyraźnie ukazał się przetarg między *true-positive* oraz *false-negative*. Wskazanie najlepszego modelu w tym przypadku wymaga określenia jak ważne jest wykrycie wszystkich oszustw (kosztem zablokowania uczciwych użytkowników).

4. Random forest

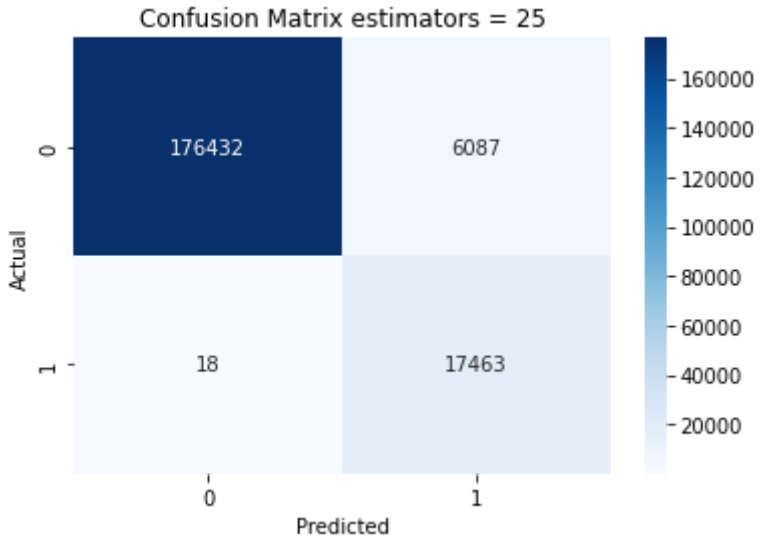
Do testów użyłem implementacji random forest zawartej w bibliotece sklearn. Podstawowym parametrem jaki ustawiłem była zbalansowana waga klas. Z racji iż wiedziałem, że pojedyncze drzewo o głębokości równej 5 może skutecznie sklasyfikować zbiór,

w tym przypadku postanowiłem ograniczyć maksymalną głębokość (do 3) i zbadać jak z problemem poradzi sobie więcej klasyfikatorów. Wyniki przedstawiają się następująco:

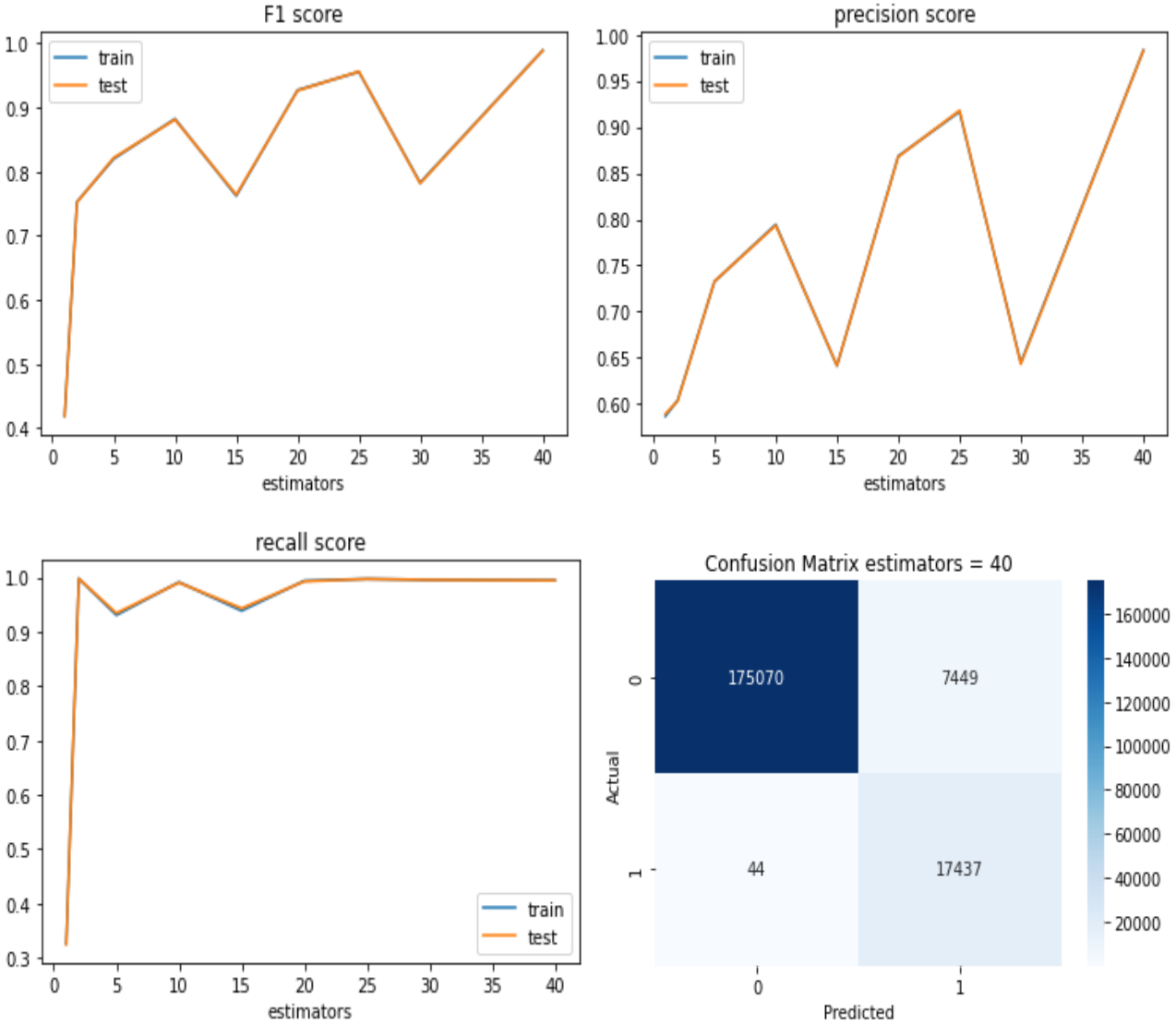




Najbardziej obiecująco wyglądają wyniki osiągnięte dla 25 drzew, jednak wyniki wydają się minimalnie gorsze niż w przypadku pojedynczego drzewa.



W przypadku oversamplingu uzyskane wyniki okazały się gorsze niż dla wejściowych danych. Zbliżony efekt udało się uzyskać dla undersampling'u, jednak dopiero przy użyciu 40 estymatorów.

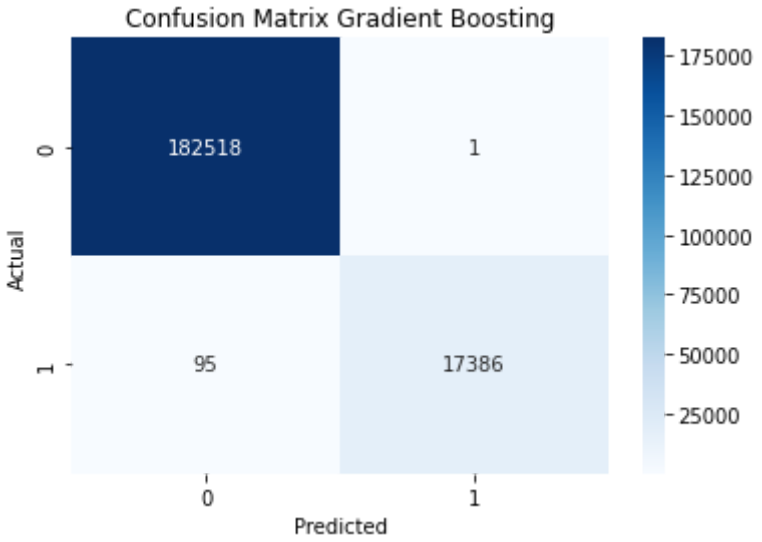


Dla odpowiedniej ilości klasyfikatorów możliwe było osiągnięcie efektów zbliżonych do pojedynczego większego drzewa.

5. Gradient Boosting

Do testów użyłem implementacji Gradient Boosting z biblioteki sklearn. Z racji iż ten typ modelu dobrze radzi sobie z niezbalansowanymi danymi nie było potrzeby używania oversampling'u, ani undersampling'u. Przy użyciu 100 estymatorów, współczynniku uczenia ustawionym na 0.1, oraz funkcji straty opartej na binarnej funkcji straty logistycznej, udało się osiągnąć niemal bezbłędne wyniki.

F1	precision	recall	accuracy
0.9972	0.9999	0.9945	0.9995



6. Wnioski

Pomimo niezbalansowania danych w tym zbiorze, dla niektórych klasyfikatorów udało osiągnąć się blisko 100% trafności. Wyniki na zbiorze testowym praktycznie nie różniły się od tych uzyskanych podczas treningu. Najprawdopodobniej jest to spowodowane dużą ilością przypadków (aż milion), a stosunkowo małą ilością atrybutów których jest jedenaście. W testach najlelepsze wyniki udało osiągnąć się przy i gradient boosting, czego można się było spodziewać bo modele tego typu dobrze radzą sobie z niezbalansowanymi danymi. Niewiele gorsze okazało się drzewo decyzyjne.