Sprawozdanie: Analiza danych zakupowych i klasyfikacja klientów

# 1. Opis danych

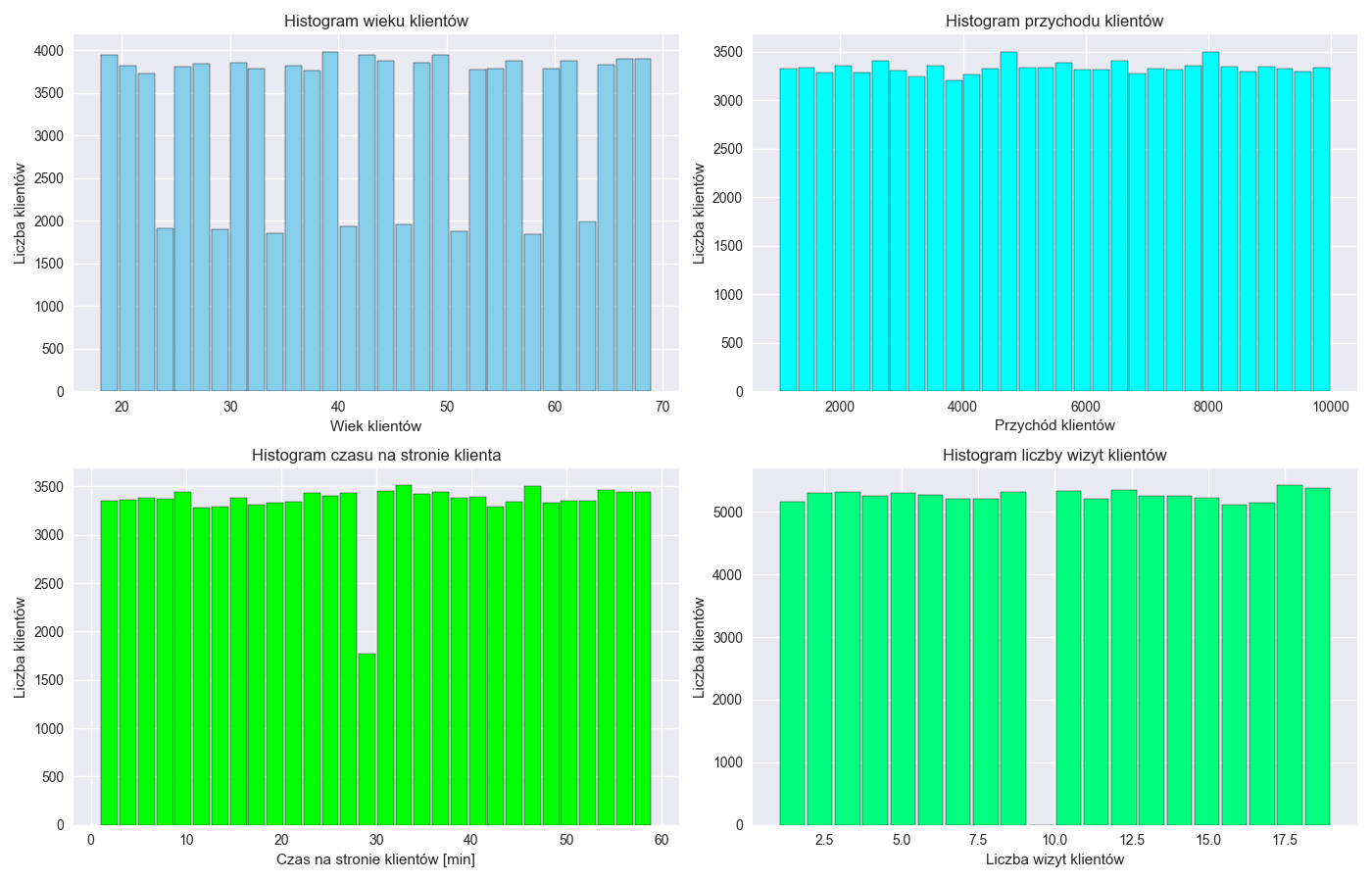
Dane pochodziły z pliku CSV zawierającego informacje o klientach odwiedzających stronę internetową. Zmienna docelowa to 'kupil', przyjmująca wartości 0 (nie kupił) i 1 (kupił). Wśród pozostałych zmiennych znalazły się dane liczbowe (wiek, przychód, czas na stronie, liczba wizyt) oraz kategoryczne (godzina wejścia, dzień tygodnia, region, typ klienta).

# 2. Eksploracja danych

Dane zostały przeanalizowane. Stworzono dla nich wykresy w celu znalezienia zależności i ocenienia ich przydatności.

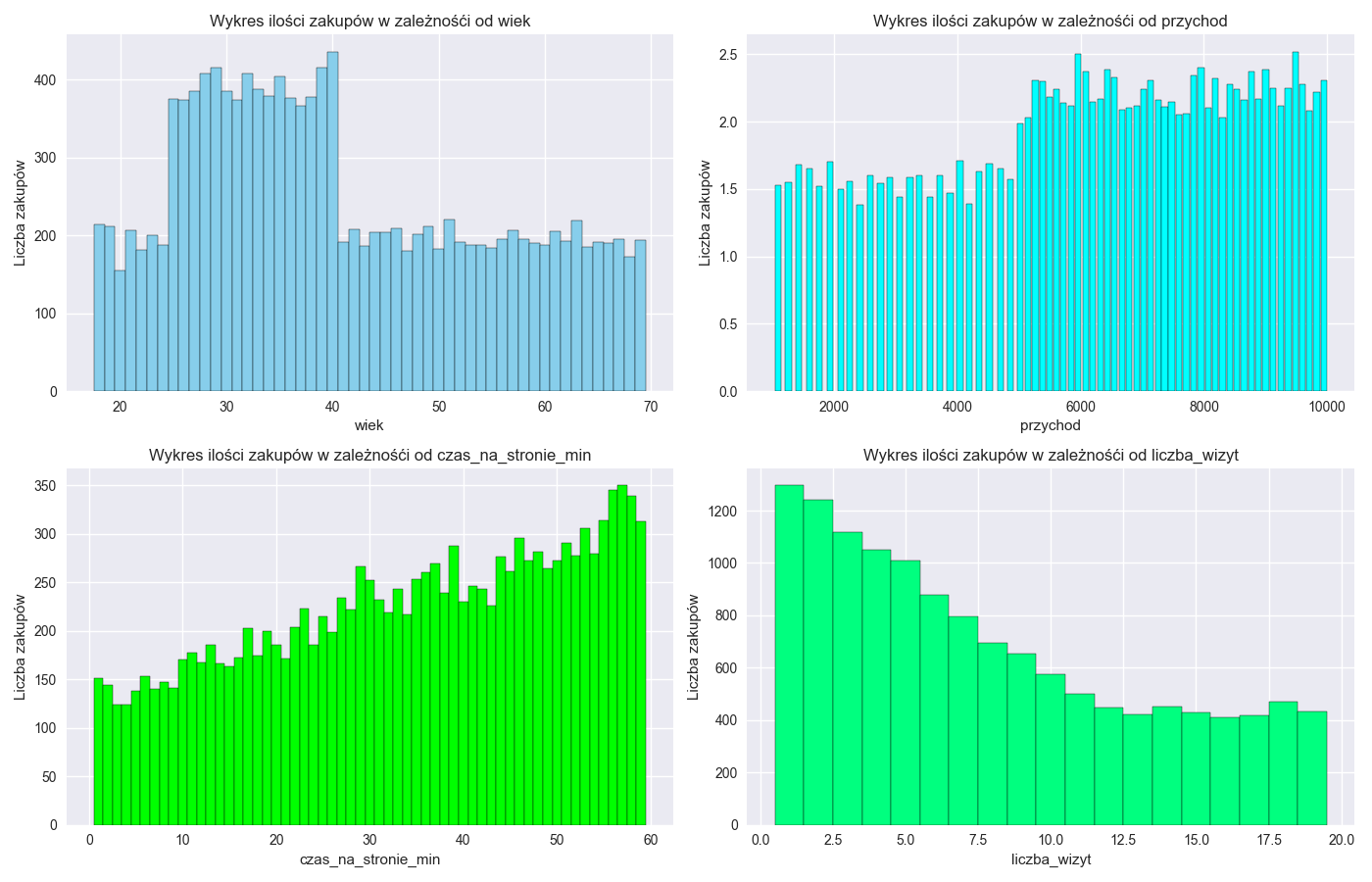
## 2.1 Dane liczbowe

Sporządzono histogramy [Wkres 1] dla zmiennych liczbowych, które ujawniły rozkłady wieku, przychodu czy aktywności klientów.



Wykres 1 - histogramy dla zmiennych liczbowych

Na wykresach widać, że dane są zbalansowane. Nie będą one zakrzywiać wyników działania modelów i model będzie się uczył równomiernie dla każdego przypadku.

Następnie utworzono wykresy [Wykres 2] przedstawiające ilość zakupów w zależności od wieku, przychodu, czasu na stronie i liczby wizyt.

Wykres 2 - ilość zakupów w zależności od wieku, przychodu, czasu na stronie i liczby wizyt

Po przeanalizowaniu wykresów widać, że ludzie chętniej kupują w zależności od różnych warunków. Klient dokonuje zakupu jeśli:

* Jest w wieku pomiędzy 25 a 40 lat,
* Przychody ma w zakresie od 5000 do 10000 zł
* Wraz ze zwiększającym się czasem na stronie

Klient natomiast rzadziej kupuje im więcej razy odwiedzał stronę internetową

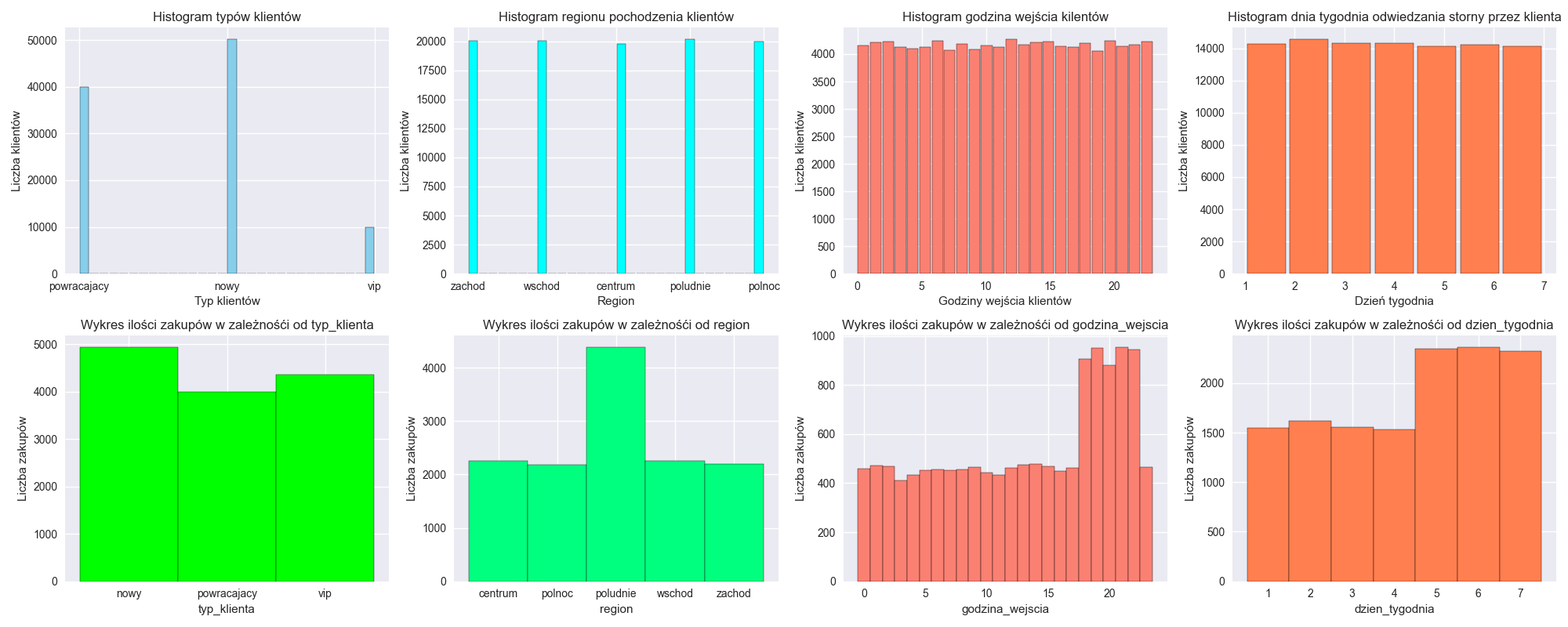
Widzą te zależności stwierdzono, że każda dana numeryczna jest użyteczna w szkoleniu modelu. Dodatkowo jest mało tych danych, więc ze względów na szybkość uczenia nie trzeba było z nich rezygnować.

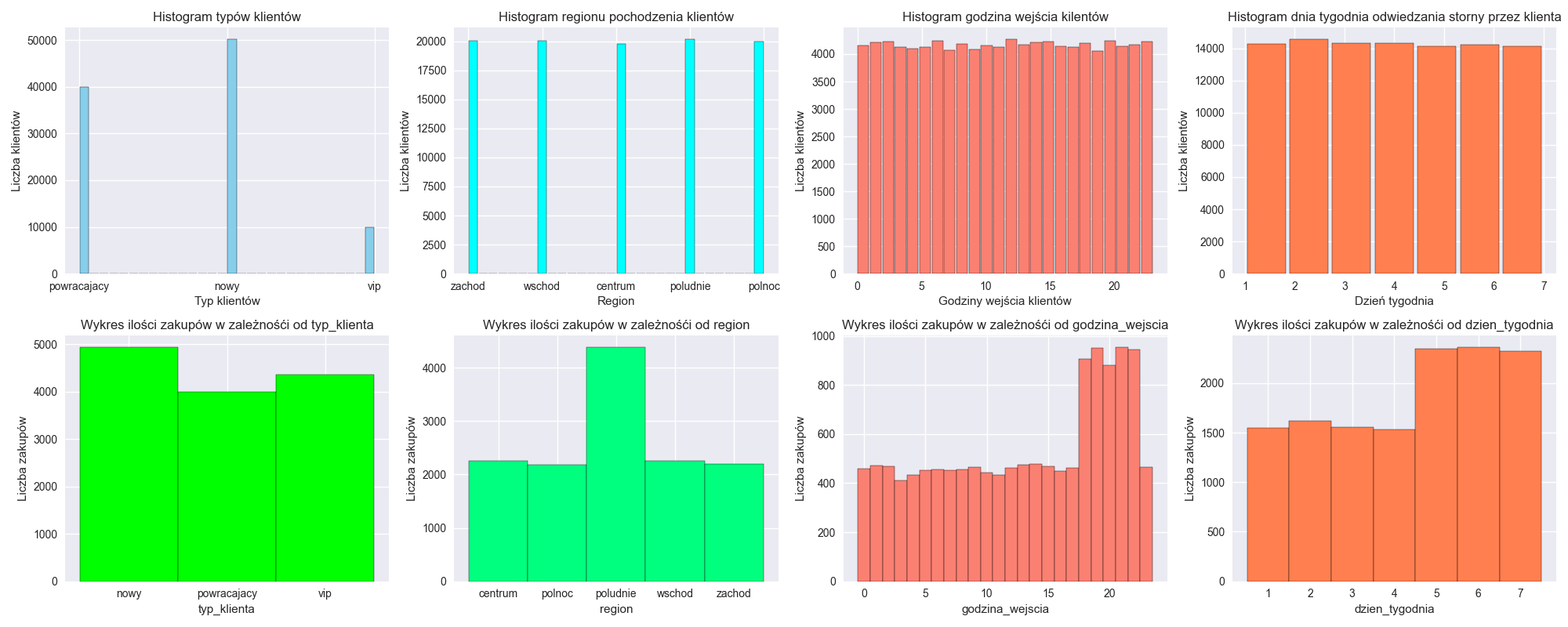
## 2.2 Dane kategoryczne

W analizowanym zbiorze danych znajdują się cztery kolumny zawierające dane kategoryczne: typ\_klienta, godzina\_wejscia, dzien\_tygodnia oraz region.

1. Typ klienta i region są zapisane jako ciągi znaków (stringi). Dane te nie posiadają naturalnej kolejności — nie można zatem przypisać im żadnego porządku logicznego ani numerycznego (np. VIP ≠ więcej niż Nowy). Z tego względu zastosowanie metody kodowania porządkowego (Ordinal Encoding) byłoby błędne. Do ich przetworzenia użyto kodowania typu one-hot encoding, które zapewnia równe traktowanie każdej kategorii.
2. Godzina wejścia oraz dzień tygodnia zostały zapisane jako liczby całkowite, jednak w kontekście modelowania nie należy traktować ich jako wartości ciągłych. Liczba „23” nie oznacza „więcej” niż „1” — są to po prostu różne kategorie godzinowe. Z tego względu również dla tych zmiennych zastosowano one-hot encoding, aby uniknąć błędnych założeń o porządku lub wartości.

Dla wszystkich czterech zmiennych kategorycznych utworzono histogramy [Wykres 3] przedstawiające częstotliwość występowania poszczególnych kategorii.



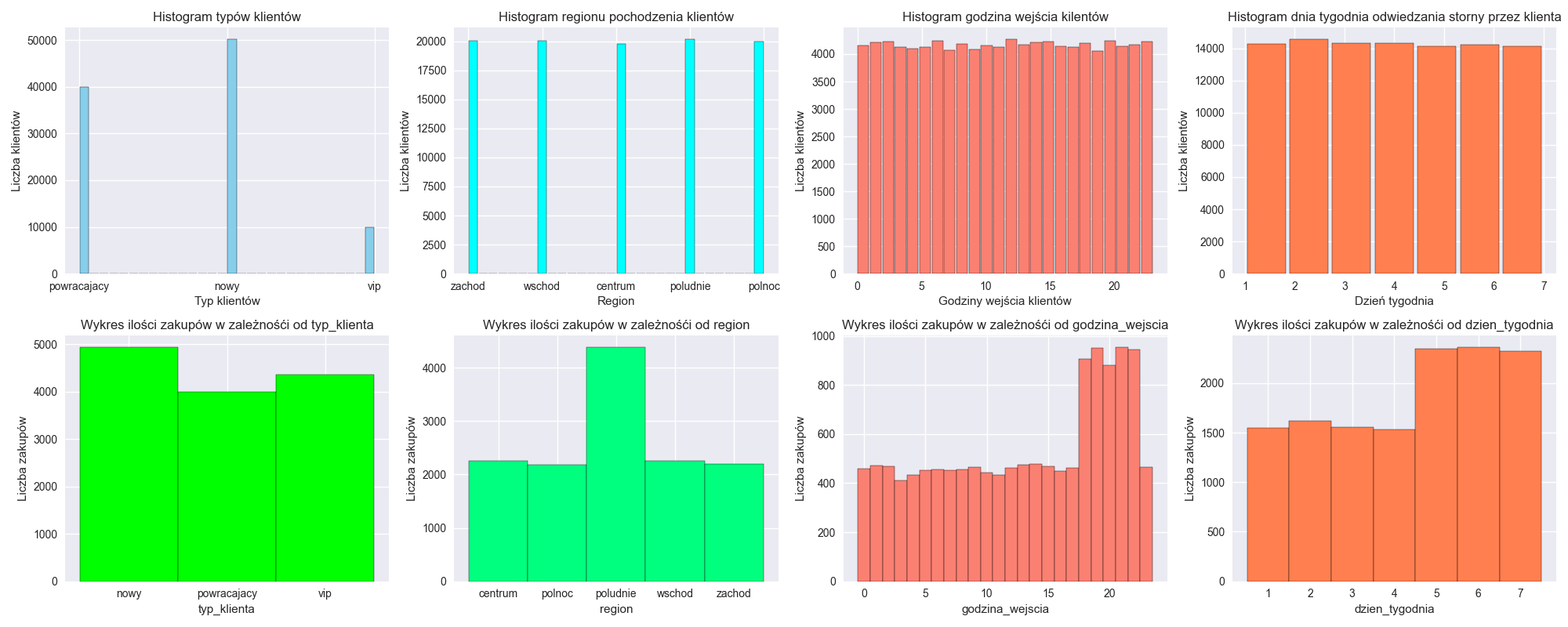


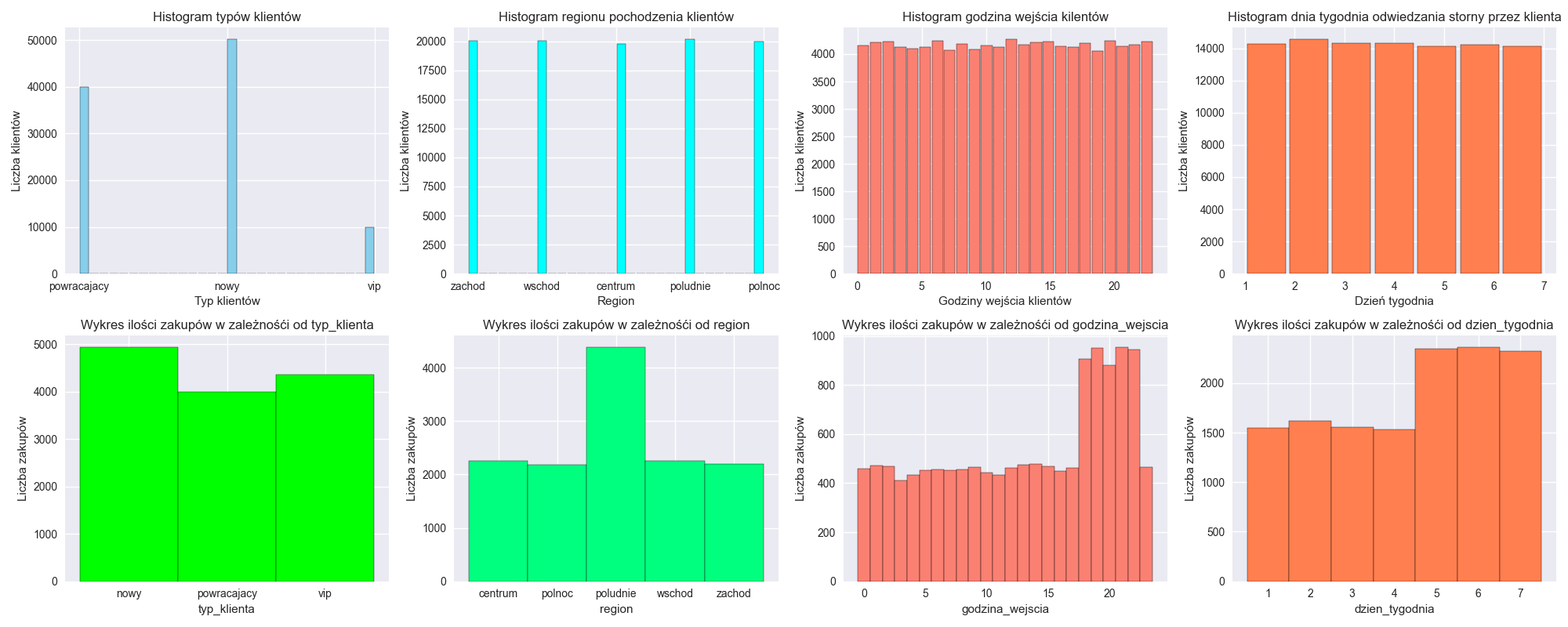
Wykres 3 – częstotliwość występowania poszczególnych kategorii

Obserwacje z histogramów:

* Zmienne godzina\_wejścia, dzień\_tygodnia oraz region wykazują równomierne rozłożenie wartości — brak widocznych dominujących kategorii.
* Odmienna sytuacja występuje w przypadku zmiennej typ\_klienta, w której widoczna jest znaczna nierównowaga – zdecydowanie najwięcej jest klientów nowo odwiedzających stronę, podczas gdy klienci VIP stanowią jedynie niewielki odsetek.

Następnie sporządzono wykresy przedstawiające liczbę dokonanych zakupów w zależności od poszczególnych cech kategorycznych.





Wykres 4 – zakupy w zależności od danej cechy

Analiza tych wykresów pozwoliła wyciągnąć następujące wnioski:

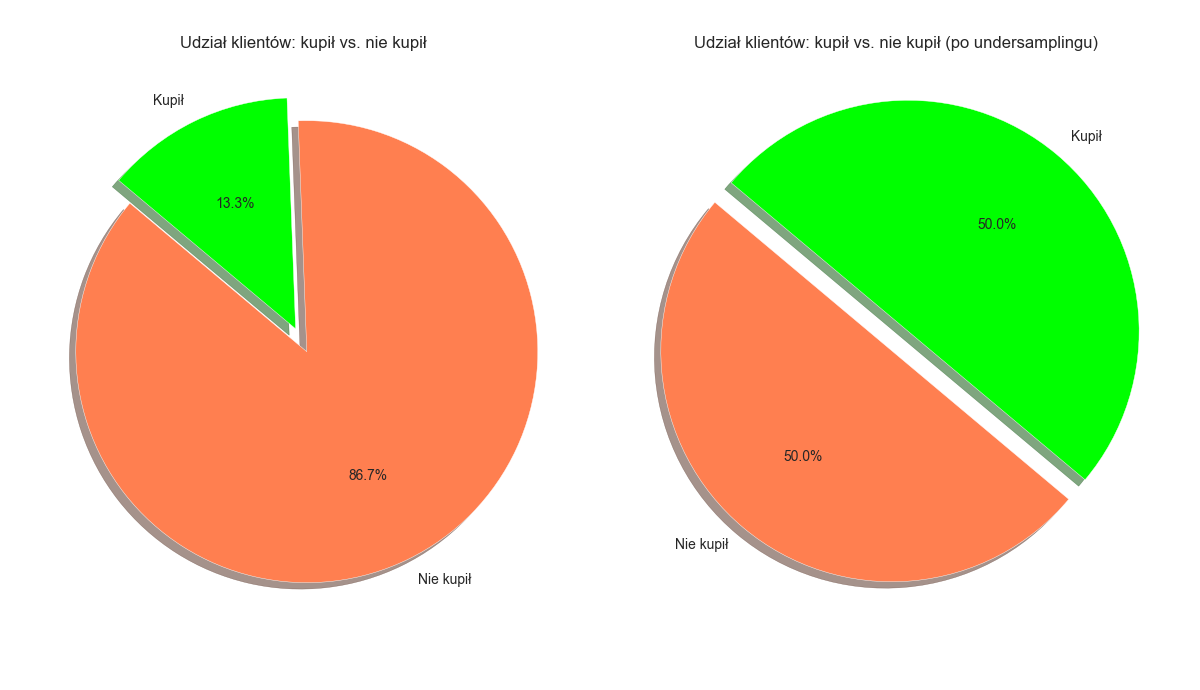
Klienci typu VIP, mimo że występowali najrzadziej, dokonują zakupów na poziomie porównywalnym do klientów nowo pozyskanych lub powracających. Może to wskazywać na wysoką konwersję w tej grupie.

Najczęściej zakupu dokonują klienci:

* z regionu południe,
* w godzinach 19:00–23:00,
* w piątki, soboty i niedziele – co może wskazywać na większą aktywność zakupową w godzinach wieczornych i w weekendy.

## 2.3 Dane docelowe

Ostatnią analizowaną kolumną była kolumna docelowa kupil, informująca o tym, czy klient dokonał zakupu (1) czy nie (0).



Wykres 5 – rozkład przed dokonaniem undersamplingu

Wykres kołowy pokazujący rozkład tej zmiennej ujawnił znaczną nierównowagę klas — zdecydowana większość klientów nie dokonuje zakupu. Taki brak balansu może negatywnie wpływać na skuteczność klasyfikatorów, które będą miały tendencję do przewidywania klasy dominującej.

Z tego względu zaleca się zastosowanie jednej z technik równoważenia zbioru danych, takich jak:

* Oversampling (RandomOverSampler) – powielanie danych klasy mniejszościowej,
* Undersampling (RandomUnderSampler) – redukcja liczby przykładów klasy większościowej.

Dzięki zastosowaniu tych metod modele uczące się będą miały szansę lepiej rozpoznawać przypadki rzadziej występujące (klientów, którzy dokonali zakupu), co wpłynie na poprawę metryk takich jak recall, precision czy f1-score.

# 3. Przetwarzanie danych

W celu przygotowania danych do modelowania zastosowano szereg technik przetwarzania wstępnego. Zmienne kategoryczne zostały zakodowane za pomocą metody one-hot encoding. Miało to dwa główne cele:

Zakodowanie zmiennych nominalnych, takich jak region oraz typ klienta (status klienta), w sposób umożliwiający modelowi właściwą interpretację bez nadawania im sztucznego porządku.

Zakodowanie zmiennych uporządkowanych, takich jak godzina wejścia i dzień tygodnia, aby uniknąć sytuacji, w której model mógłby błędnie zakładać, że wyższe wartości (np. godzina 23 lub dzień 7) są w jakiś sposób „lepsze” lub bardziej znaczące niż niższe (np. godzina 1 lub dzień 1). Dzięki one-hot encoding wszystkie wartości były traktowane równorzędnie.

Dla zmiennych liczbowych (np. wiek, przychód, czas na stronie, liczba wizyt) przeprowadzono próby skalowania przy użyciu dwóch popularnych metod: StandardScaler oraz MinMaxScaler. Obie techniki zostały przetestowane w połączeniu z modelami klasyfikacyjnymi, aby ocenić ich wpływ na jakość predykcji. Ostatecznie zdecydowano się na StandardScaler, który dał minimalnie lepsze i bardziej stabilne wyniki, szczególnie w połączeniu z modelami bazującymi na drzewach.

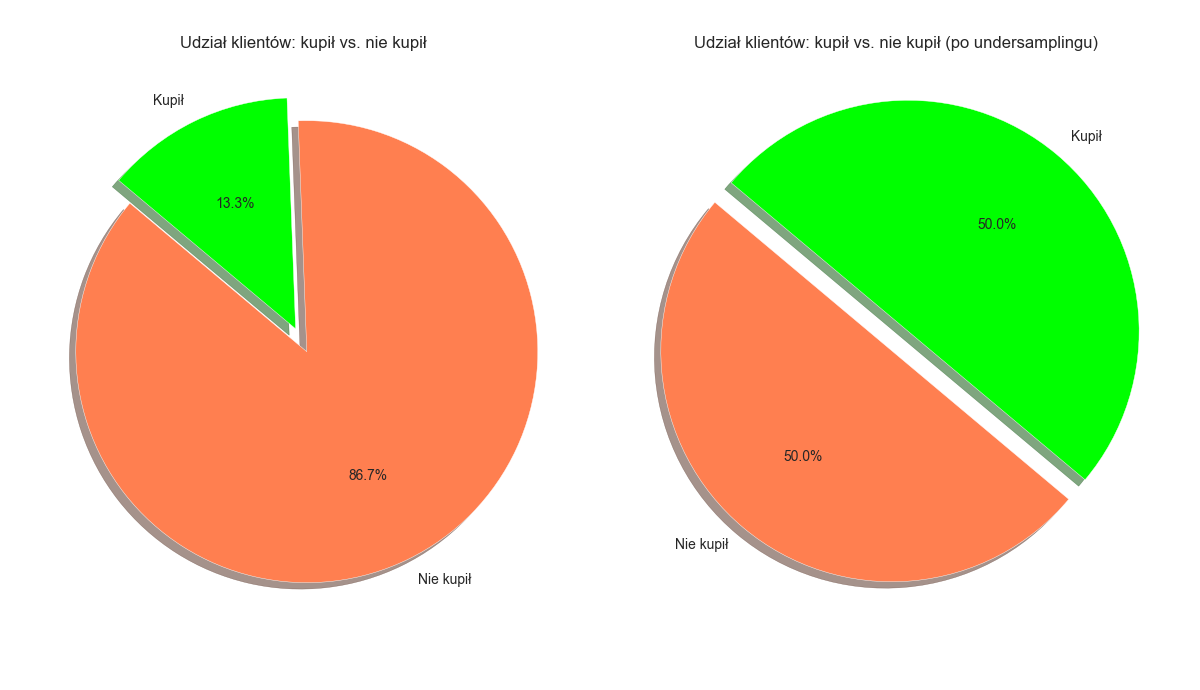
W celu zbalansowania klas w zbiorze danych (gdzie klasa „kupujący” była istotnie mniej liczna), przetestowano dwie techniki:

Oversampling – polegający na sztucznym powieleniu przykładów klasy mniejszościowej.

Undersampling – polegający na losowym usunięciu przykładów z klasy większościowej.

Wyniki analizy pokazały, że zastosowanie oversamplingu (przez RandomOverSampler) spowodowało znaczny wzrost trafności modelu Random Forest — osiągał on blisko 96% dokładności na zbiorze treningowym. Taki wynik, choć pozornie imponujący, może sygnalizować nadmierne dopasowanie (overfitting). Przed zastosowaniem tuningu model ten nie nadawał się do użytku

Z kolei technika undersamplingu, mimo że doprowadziła do zmniejszenia dokładności modelu (do poziomu około 78–80%), zapewniła znacznie bardziej realistyczne i wiarygodne wyniki. Model przestał faworyzować klasę dominującą i lepiej radził sobie z generalizacją



Wykres 6 - rozkład klientów co kupili i nie kupil po undersamplingu\oversamlingu

# 4. Podział danych

Dane podzielono na zbiór treningowy i testowy w proporcji 80:20. Dzięki temu możliwe było skuteczne uczenie modeli oraz ich późniejsza ocena na niezależnym zbiorze danych.

# 5. Budowa i ocena modeli

W ramach projektu zbudowano i przetestowano trzy różne modele klasyfikacyjne: regresję logistyczną, Random Forest oraz XGBoost. Każdy z modeli został oceniony na tej samej próbce testowej, z wykorzystaniem kluczowych metryk: precision, recall, f1-score, accuracy oraz wskaźnika ROC AUC. Oto ich szczegółowe wyniki i interpretacja:

Regresja logistyczna:

* Accuracy: 75%
* Recall (klasa 0):77% (klasa 1):73%
* Precision (klasa 0):74% (klasa 1):76%
* F1-score (klasa 0):75% (klasa 1):74%
* ROC AUC: 0.796

Model bazowy oparte na regresji logistycznej osiągnął umiarkowane, ale zrównoważone wyniki. Skuteczność klasyfikacji klientów, którzy dokonali zakupu (klasa 1), była na poziomie zbliżonym do klasy 0. Ten model cechuje się prostotą, dużą interpretowalnością oraz niskim ryzykiem przeuczenia, ale nie radzi sobie najlepiej w przypadkach bardziej złożonych wzorców.

Random Forest:

* Accuracy: 79%
* Recall (klasa 0):96% (klasa 1):89%
* Precision (klasa 0):89% (klasa 1):96%
* F1-score (klasa 0):92% (klasa 1):92%
* ROC AUC: 0.98

Random Forest uzyskał wyższą skuteczność niż regresja logistyczna. Model lepiej rozpoznawał klasę 0 (niekupujących), jednak również skutecznie klasyfikował klasę pozytywną. Poprawił się zwłaszcza wskaźnik precision (0. 96) dla kupujących, co oznacza mniejszą liczbę fałszywych alarmów. Jednocześnie recall dla tej klasy wyniósł 0.89, co świadczy o bardzo dobrej czułości modelu. Wskaźnik ROC AUC powyżej 0.98 sugeruje solidną zdolność modelu do odróżniania klas. Jest tutaj ryzyko wystąpienie przetrenowania. Dokonano testu, by to sprawdzić i wyniki nie są jednoznaczne.

XGBoost:

* Accuracy: 92%
* Recall (klasa 0):90% (klasa 1):76%
* Precision (klasa 0):79% (klasa 1):88%
* F1-score (klasa 0):84% (klasa 1):82%
* ROC AUC: 0.91

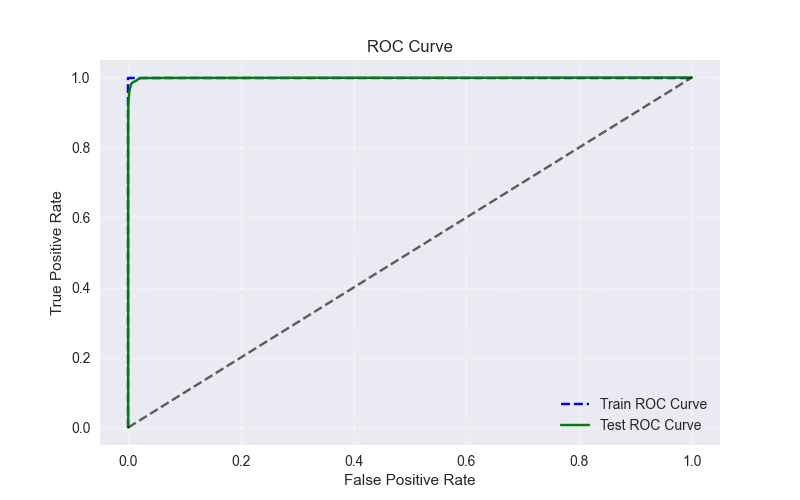
Model oparty na XGBoost osiągnął bardzo zbliżone wyniki do Random Forest. Cechował się nieco lepszym zrównoważeniem między precision i recall, a jego f1-score był stabilny w obu klasach. Choć ustępował Random Forest pod względem AUC i accuracy, to jego odporność na przeuczenie oraz zdolność do wykrywania trudniejszych wzorców czynią go bardzo konkurencyjnym modelem.

# 6. Tuning hiperparametrów

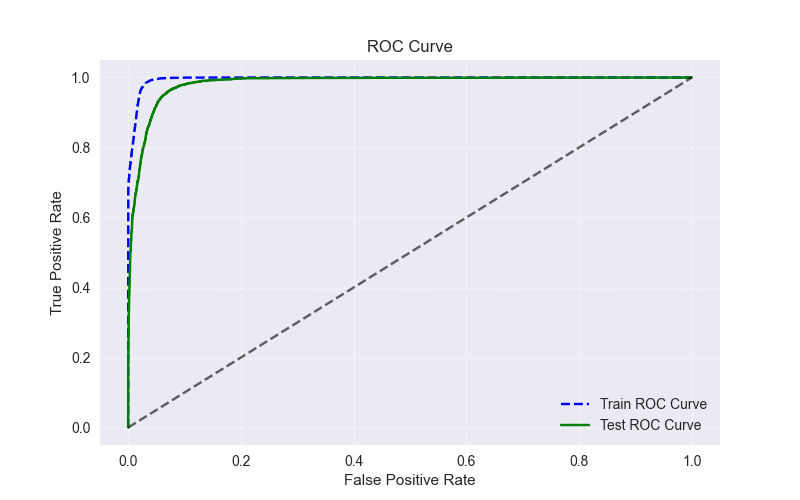
Przeprowadzono tuning hiperparametrów dla modelu Random Forest z wykorzystaniem narzędzia GridSearchCV. W ramach tego procesu testowano różne kombinacje parametrów, takie jak liczba drzew w lesie (n\_estimators), maksymalna głębokość drzew (max\_depth), minimalna liczba próbek do podziału węzła (min\_samples\_split) oraz strategia wyboru cech (max\_features). Celem tych działań było zwiększenie skuteczności modelu, poprawa jego zdolności generalizacyjnych oraz ograniczenie ryzyka przeuczenia na zbiorze treningowym.

Zastosowanie tuningu hiperparametrów przyczyniło się do ograniczenia zjawiska overfittingu. Choć bezpośrednie wskaźniki skuteczności, takie jak accuracy czy f1-score, uległy nieznacznemu obniżeniu w porównaniu do wersji bazowej modelu, to uzyskane wyniki są bardziej realistyczne, stabilne i lepiej odzwierciedlają zachowanie modelu w rzeczywistych warunkach predykcyjnych.

Mimo że wskaźniki detekcji (np. recall dla klasy kupujących) mogły nieznacznie spaść, to uzyskane wyniki są bardziej wiarygodne i sensowne analitycznie, co zwiększa zaufanie do działania modelu w środowisku produkcyjnym. Wyniki przed tuningiem mamy na wykresie 7 a po na wykresie 8



Wykres 7 - test overfittingu przed tuningiem



Wykres 8 - test overfittingu po tuningu

# 7. Ocena modeli

Oceniając wyniki trzech zastosowanych modeli klasyfikacyjnych, można zauważyć wyraźne różnice między nimi:  
  
- Regresja logistyczna uzyskała najniższą skuteczność, ale cechowała się prostotą i stabilnością. Jej f1-score wyniósł 0.75, co przy zbalansowanych klasach jest wynikiem akceptowalnym. Model ten może być użyteczny jako bazowy punkt odniesienia w dalszych eksperymentach.  
  
- Random Forest osiągnął najlepsze wyniki f1-score i ROC AUC spośród wszystkich modeli. Jednocześnie jednak uzyskał 95% trafności na zbiorze treningowym, co wskazuje na potencjalne przeuczenie modelu, ale może zostać wykorzystane w środowisku produkcyjnym.  
  
- Model XGBoost osiągnął zrównoważone wyniki – f1-score na poziomie 0.84, AUC: 0.83. Wyniki były słabsze niż w przypadku Random Forest, ale model okazał się bardziej odporny na przeuczenie. Dobrze sprawdzał się szczególnie przy danych o nieco większej złożoności.   
  
Podsumowując, model Random Forest miał najlepsze wyniki, lecz jego potencjalne przetrenowanie może ograniczać jego zastosowanie w środowisku produkcyjnym. XGBoost wydaje się być kompromisem między skutecznością a stabilnością, natomiast regresja logistyczna nadaje się do szybkiej, wstępnej analizy predykcyjnej.

Jednak ze względu na największy potencjał Random Forset jest leszpym modelem do tego zadania

# 8. Wnioski

Na podstawie analizy danych i oceny modeli można stwierdzić, że:  
- Dane wymagają balansu klas dla uczciwej oceny modeli.  
- Random Forest był najskuteczniejszy, ale bardziej podatny na overfitting. Dodatkowe zastosowanie tuningu danych przyniosło pozytywnych skutki w i zmniejszyło ryzyko overfittingu   
- Regresja logistyczna zapewnia stabilne wyniki przy niższej złożoności.  
- XGBoost łączy zalety obu podejść, dając wysoką skuteczność przy umiarkowanej podatności na przeuczenie.