"Klasyfikacja zadowolenia klientów linii lotniczych za pomocą różnych algorytmów uczenia maszynowego"

Jan Kwiatkowski, Jakub Kuczmarski

Marzec 2023

1 Temat Projektu

Tematem projektu jest implementacja różnego rodzaju algorytmów uczenia maszynowego do zadania klasyfikacji zadowolenia klientów linii lotniczych. Realizacja projektu będzie opierać się na statystycznym opisie, transformacji oraz odpowiednim przygotowaniu i obróbce danych (ang. data preprocesing), a następnie porównu gotowych implementacji algorytmów klasyfikacji. Na koniec dokonana zostanie ocena jakości uzyskanych modeli i na jej podstawie zostaną wyciągnięcie odpowiednie wnioski.

2 Wykorzystane algorytmy uczenia maszynowego

Do realizacji projektu wykorzystane zostały następujące algorytmy uczenia maszynowego:

- maszyna wektorów nośnych (ang. SVM - Support Vector Machine)
- drzewo decyzyjne (ang. decision tree)
- las losowy (ang. random tree forest)
- gradient boost
- klasyfikator bayesowski (ang. naivy bayes)
- regresja logistyczna (ang. logistic regression)
- algorytm stochastycznego spadku gradientu (ang. SGD - Stochastic Gradient Descent

3 Narzędzia wykorzystane przy realizacji projektu

3.1 Środowisko wykonawcze

- Jupyter Notebook [1]

  Jupyter Notebook to interaktywna środowisko programistyczne, które pozwala na pisanie, uruchamianie i udostępnianie kodu w postaci notatek. Notatki te składają się z bloków tekstu, kodu oraz wyników działania kodu. Jupyter Notebook jest często wykorzystywany w analizie danych, uczeniu maszynowym, a także w naukach przyrodniczych i humanistycznych.

- Google Colaboratory [2]

  Google Colaboratory (Colab) to platforma do uczenia maszynowego w chmurze, która umożliwia bezpłatne korzystanie z zasobów Google, takich jak procesory i karty graficzne. W Colab można tworzyć i uruchamiać notatki Jupyter Notebook w przeglądarce internetowej bez konieczności instalowania oprogramowania na własnym komputerze.

- Pycharm [3]

  PyCharm to zintegrowane środowisko programistyczne (IDE) dla języka Python. PyCharm oferuje wiele narzędzi, takich jak debugger, system kontroli wersji, wsparcie dla testów jednostkowych i integrację z narzędziami deweloperskimi, takimi jak Docker. PyCharm jest często wykorzystywany w projektach programistycznych, w których ważne jest zarządzanie kodem, debugowanie i wdrożenie aplikacji.

3.2 Wykorzystane biblioteki

- Pandas [4]

  Pandas to biblioteka dla języka Python, która służy do analizy danych. Pandas pozwala na łatwe wczytywanie, przetwarzanie, przekształcanie i wizualizację danych w formie tabelarycznej. Biblioteka ta umożliwia między innymi filtrowanie i grupowanie danych, łączenie tabel oraz obliczanie statystyk.

- Numpy [5]

  NumPy to biblioteka do obliczeń naukowych w języku Python. NumPy dostarcza wielowymiarowe tablice i funkcje matematyczne do operacji na tych tablicach. Biblioteka ta jest szczególnie przydatna w obliczeniach numerycznych i uczeniu maszynowym, gdzie często operuje się na dużych zbiorach danych.

- Scikit-learn [6]

Scikit-learn to biblioteka dla języka Python służąca do uczenia maszynowego. Scikit-learn dostarcza narzędzia do klasyfikacji, regresji, grupowania i redukcji wymiarowości danych. Biblioteka ta umożliwia również ocenę jakości modeli oraz wybór najlepszego modelu do danego problemu.

- Matplotlib [7]

  Matplotlib to podstawowa biblioteka do tworzenia wykresów i grafik. Jest ona bardzo elastyczna i pozwala na tworzenie niemal każdego typu wykresu, włącznie z histogramami, wykresami liniowymi, wykresami słupkowymi, wykresami punktowymi itp. Matplotlib umożliwia kontrolę niemal każdego aspektu wykresu, włącznie z etykietami, kolorami, stylami linii, rodzajami markerów itp.

- Seaborn [8]

  Seaborn to biblioteka, która jest oparta na matplotlib i zapewnia bardziej zaawansowane możliwości wizualizacji danych. Seaborn oferuje wiele wbudowanych stylów i kolorów, które ułatwiają tworzenie estetycznych i przejrzystych wykresów. Dodatkowo, Seaborn oferuje wiele funkcji do automatycznego tworzenia wykresów, takich jak wykresy rozproszenia z dopasowaną linią regresji czy wykresy heatmap.

- XGBoost [9]

  XGBoost to biblioteka do uczenia maszynowego, która specjalizuje się w algorytmach gradient boosting i jest często używana do klasyfikacji.

3.3 Język implementacji

- Python [10]

  Python to prosty, interpretowany język programowania, który wspiera wiele paradygmatów programowania i ma bogatą bibliotekę standardową. Python jest wykorzystywany w wielu dziedzinach, takich jak analiza danych, uczenie maszynowe, web development itp. Python jest również platformowo-niezależny i posiada szeroką społeczność programistów.

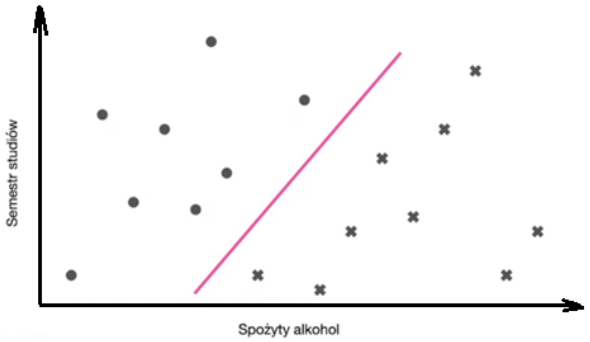
4 Wstęp ogólny

Zadanie klasyfikacja, czyli przydzielania danego obiektu na podstawie wartości jego atrybutów do jednej z dostępnych klas może być realizowane za pomocą algorytmów uczenia maszynowego.

W ramach projektu zadanie klasyfikacji będzie realizowane przy pomocy algorytmów opisanych w rozdziale 2. Będziemy oceniać jakość modeli na podstawie wyników i dostosowywać parametry, jeśli będzie to konieczne, aby uzyskać jak najlepsze wyniki.

4.1 Maszyna wektorów nośnych [11]

SVM (ang. Support Vector Machine) to algorytm klasyfikacji binarnej, który szuka hiperpłaszczyzny separującej z maksymalnym marginesem klasy w przestrzeni wielowymiarowej. Dzięki zastosowaniu tzw. funkcji jądrowych, SVM może radzić sobie z danymi nieliniowymi. Prezentację takiego podziału przedstawia rysunek 1



Rys. 1: SVM

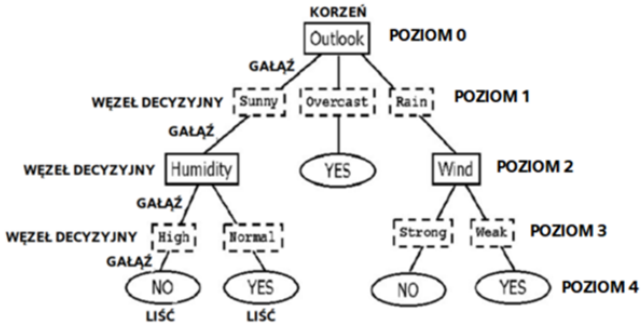
Parametry modelu wymienione poniżej będą strojone za pomocą walidacji krzyżowej:

- Kernel: SVM może wykorzystać różne funkcje jądra (kernel), które definiują sposób, w jaki obliczane są odległości między punktami w przestrzeni cech (są to np. funkcje liniowe, wielomianowe, radialne).
- C: Parametr ten kontroluje balans między dopasowaniem do danych szkoleniowych a dopasowaniem do danych testowych. Ma on wpływ na siłę regulacji algorytmu i decyduje, jak wiele punktów będzie uznawanych za odstające.
- Gamma: kontroluje zasięg wpływu pojedynczego punktu danych i wpływa na kształt granicy decyzyjnej. Wysokie wartości Gamma skutkują skomplikowanymi granicami decyzyjnymi, co może prowadzić do przeuczenia, podczas gdy niskie wartości Gamma skutkują prostymi granicami decyzyjnymi, co może prowadzić do niedostatecznego dopasowania.

- Max_iter: Parametr określający maksymalną liczbę iteracji, które algorytm SVM ma wykonać.
- Tolerancja: Określa tolerancję dla zakończenia procesu uczenia, gdy różnica między dwoma kolejnymi iteracjami jest mniejsza niż ustalona wartość.

4.2 Drzewo decyzyjne [12]

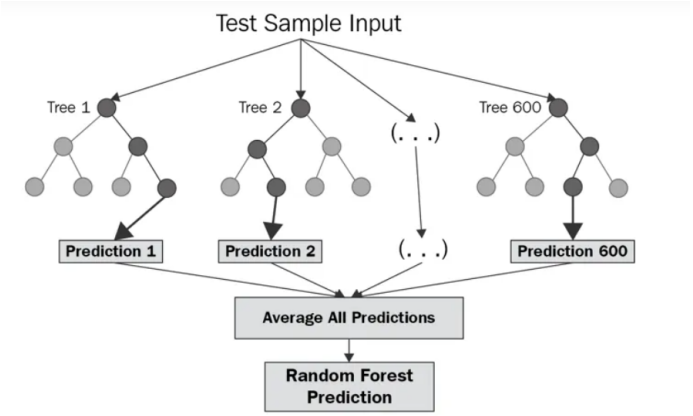
Drzewo decyzyjne (ang. decision tree) to algorytm uczenia maszynowego wykorzystywany do klasyfikacji i prognozowania. Polega na tworzeniu drzewa, w którym każdy węzeł odpowiada testowi na wartości jednej z cech wejściowych, a każda krawędź wychodząca z węzła odpowiada jednemu z możliwych wyników tego testu. Na końcu każdej ścieżki drzewa znajduje się klasyfikacja lub prognoza. Prezentację takiego podziału przedstawia rysunek 2.



Rys. 2: Drzewo Decyzyjne

4.3 Las losowy [13]

Las losowy (ang. random forest) to algorytm uczenia maszynowego, który służy do klasyfikacji, regresji i innych zadań uczenia maszynowego. Polega na tworzeniu wielu drzew decyzyjnych, gdzie każde drzewo jest budowane na podstawie losowego podzbioru danych i losowego podzbioru zmiennych. Ostatecznie, wyniki z różnych drzew są łączone w celu uzyskania końcowej predykcji. Dzięki temu algorytm jest odporny na overfitting i może osiągać bardzo dobre wyniki predykcyjne. Prezentację takiego podziału przedstawia rysunek 3.



Rys. 3: Las Losowy.
Źródło:
https://corporatefinanceinstitute.com/resources/data-science/random-forest/

4.4 Gradient boost [14]

Gradient Boosting to technika uczenia maszynowego, która buduje model poprzez łączenie wielu prostych modeli, takich jak drzewa decyzyjne.

Algorytm działa poprzez sekwencyjne budowanie drzew, gdzie każde kolejne drzewo stara się poprawić błąd predykcji poprzedniego drzewa. Wartość progowa jest aktualizowana na podstawie różnicy między rzeczywistą wartością a przewidywaną wartością dla danego przykładu.

W jego przypadku istnieje wiele parametrów, które można stroić w celu uzyskania najlepszego modelu dla konkretnego zadania. Są to:

- Learning rate - określa jak dużo wkładu mają pojedyncze drzewa w budowę całego modelu. Wysoki learning rate oznacza większy wpływ każdego drzewa i szybszą konwergencję, ale może prowadzić do przeuczenia. Niski learning rate zmniejsza wpływ każdego drzewa i może wydłużyć czas uczenia, ale również zmniejsza ryzyko przeuczenia.
- Liczba drzew - określa ile drzew zostanie użytych w modelu. Zwiększenie liczby drzew może poprawić wyniki, ale również zwiększa czas uczenia i ryzyko przeuczenia.
- Głębokość drzew - określa maksymalną głębokość każdego drzewa. Zwiększenie głębokości drzew może zwiększyć złożoność modelu i poprawić wyniki, ale również zwiększa ryzyko przeuczenia.
- Liczba węzłów - określa minimalną liczbę próbek wymaganą do utworzenia nowego węzła. Zwiększenie liczby węzłów zmniejsza liczbę węzłów w modelu i może zmniejszyć złożoność modelu, ale również zmniejsza dopasowanie do danych treningowych.
- Liczba próbek na liściu - określa minimalną liczbę próbek wymaganą do utworzenia liścia. Zwiększenie liczby próbek na liściu zmniejsza liczbę liści w modelu i może zmniejszyć złożoność modelu, ale również zmniejsza dopasowanie do danych treningowych.

- Funkcja straty - określa funkcję, która jest minimalizowana podczas uczenia modelu. W przypadku Gradient Boosting najczęściej stosowane funkcje straty to Mean Squared Error (MSE) i Binary Cross-Entropy (dla problemów binarnych).

4.5 Naiwny Klasyfikator Bayesowski [15]

Algorytm ten oparty jest na rachunku prawdopodobieństwa. Na podstawie bazy etykietowanych przykładów D dla nowego przykładu d_j klasyfikator ma stwierdzić jaka jest najbardziej prawdopodobna wartość decyzji dla tego przykładu lub jakie jest prawdopodobieństwo przypisania nowego przykładu do jakiejś klasy na podstawie posiadanego zbioru treningowego.

Do wyznaczania wartości prawdopodobieństwa algorytm wykorzystuje zmodyfikowaną wersję twierdzenia o prawdopodobieństwie warunkowym (wzór 1). Korzystając z pewnego, „naiwnego" założenia, że poszczególne prawdopodobieństwa warunkowe są niezależne, wyprowadzona jest zależność na której opiera się ten klasyfikator. Dla takich niezależnych zdarzeń prawdopodobieństwo koniunkcji jest równe iloczynowi prawdopodobieństw wystąpienia każdej wartości atrybutu we wskazanej klasie (wzór 2) .

$$P(h_i \mid d_j) = P(a_1 \mid h_i) * P(a_2 \mid h_i) * ... * P(a_n \mid h_i) \tag{1}$$

$$P(h_i \mid d_j) = \frac{\Pi P(a_j \mid h_i) * P(h_i)}{P(d_j)} \tag{2}$$

gdzie:

$h_i \in H$ - klasa decyzyjna

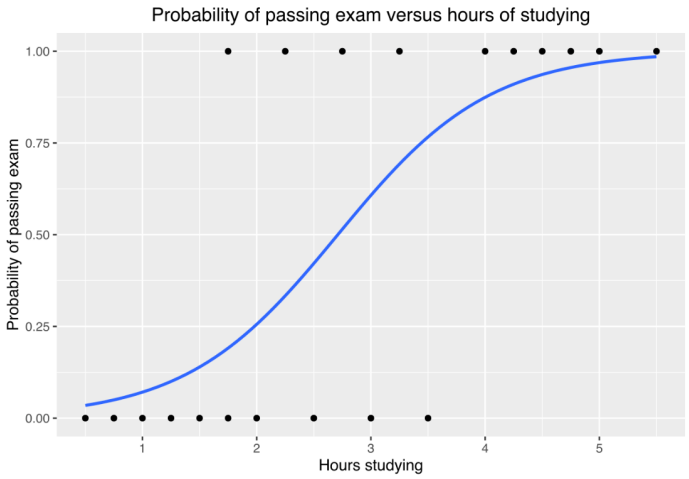
d_j - nowy atrybut składający się z wartości $< a_1, ..., a_n >$

Jest to jeden z nielicznych klasyfikatorów, który nie wymaga strojenia parametrów.

4.6 Regresja Logistyczna [15]

Regresja logistyczna jest jednym popularnych algorytmów stosowanych do klasyfikacji, który opiera się na matematycznym modelu regresji logistycznej. Regresja logistyczna jest popularna w uczeniu maszynowym ze względu na swoją prostotę, interpretowalność wyników, efektywność obliczeniową i zdolność do obsługi dużych zbiorów danych. Może być stosowana zarówno w problemach binarnych, jak i wieloklasowych, co czyni ją wszechstronnym narzędziem do klasyfikacji w różnych dziedzinach.

Przekład klasyfikacji przy pomocy regresji logistycznej przedstawiono na rysunku 4.



Rys. 4: Regresja Logistyczna.
Źródło: https://en.wikipedia.org/wiki/Logistic$_regression$

4.7 Algorytm stochastycznego spadku gradientu (ang. SGD - Stochastic Gradient Descent [16]

Algorytm stochastycznego spadku gradientu (SGD) do zastosowań w klasyfikacji jest iteracyjnym procesem aktualizacji wag w celu minimalizacji funkcji straty. W każdej iteracji, na podstawie pojedynczej obserwacji z danych treningowych, oblicza się gradient funkcji straty i aktualizuje wagi modelu. Dzięki temu algorytmowi SGD można skutecznie stosować w klasyfikacji dużych zbiorów danych, umożliwiając efektywne i skalowalne uczenie maszynowe.

5 Zbiór danych

W projekcie wykorzystano ogólnodostępny zbiory danych pochodzący ze strony Kaggle – „Airline Passenger Satisfaction" [18]. Ten zestaw danych zawiera ankietę satysfakcji pasażerów różnych linii lotniczych pozwalający określić jak silnie poszczególne czynniki są skorelowane z zadowolonym (lub niezadowolonym) pasażerem.

5.1 Zawartość zbioru danych

Najważniejsze informacje o wykorzystanym zbiorze danych przedstawiono w tabeli 1

Tabela 1: Zbiór danych

Nazwa zbioru danych	Liczba klas	Liczba atrybutów	Liczba przypadków
Airline Passenger Satisfaction	2	23	129880

Atrybuty wykorzystane w predykcji zadowolenia klientów linii lotniczych prezentuje tabela 2

Tabela 2: Zbiór danych - atrybuty

Atrybut	**Opis**
Gender	Płeć pasażerów (kobieta, mężczyzna)
Customer	Typ klienta (klient lojalny, klient nielojalny)
Age	Aktualny wiek pasażerów
Type of Travel	Typ lotu (podróż prywatna, podróż służbowa)
Class	Klasa podróży w samolocie (Business, Eco, Eco Plus)
Flight distance	Dystans podróży
Inflight wifi service	Poziom zadowolenia z usługi WiFi na pokładzie samolotu (0:Nie dotyczy; 1-5)
Departure/Arrival time convenient	Poziom zadowolenia z wygody czasu wylotu/przylotu
Ease of Online booking	Poziom zadowolenia z rezerwacji on-line
Gate location	Poziom zadowolenia z lokalizacji Gate
Food and drink	Poziom zadowolenia z żywności i napojów
Online boarding	Poziom zadowolenia z procesu odprawy online
Seat comfort	Poziom satysfakcji z komfortu siedzenia
Inflight entertainment	Poziom zadowolenia z rozrywki pokładowej
On-board service	Poziom zadowolenia z usług pokładowych
Leg room service	Poziom zadowolenia z usługi dotyczącej przestrzeni na nogi w samolocie
Baggage handling	Poziom zadowolenia z obsługi bagażu
Check-in service	Poziom zadowolenia z usługi odprawy
Inflight service	Poziom zadowolenia z obsługi podczas lotu
Cleanliness	Poziom satysfakcji z czystości
Departure Delay in Minutes	Opóźnienie odlotu
Arrival Delay in Minutes	Opóźnienie przylotu
Satisfaction	Poziom zadowolenia z linii lotniczych (zadowolenie, neutralność lub niezadowolenie)

5.2 Propozycja nowych atrybutów

Na podstawie przedstawionego w tabeli 2 zestawienia dostępnych w zbiorze danych atrybutów, na ich podstawie można zdefiniować nowe atrybuty takie jak:

1. Całkowity czas opóźnienia (*'Total Delay in Minutes'*) - atrybut ten określać będzie całkowity czas opóźnienia lotu (w minutach) na podstawie opóźnienia odlotu i przylotu.

5.3 Podział zbiorów

Do realizacji zadania klasyfikacji wykorzystano zaproponowany przez autora podział zbióru danych na treningowy - 103904 i testowy - 25976.

5.4 Obróbka zbioru danych

Aby rozpocząć pracę ze zbiorem danych konieczne jest jego przetworzenie w celu upewnienia się, że jest on w odpowiednim formacie do analizy. Oto kilka kroków, które zostaną podjęte, aby przetworzyć zbiór danych:

1. Import zbioru danych.
   Na początek należy zaimportować zbiór danych do środowiska programistycznego Jupyther Notebook, a następnie przy pomocy biblioteki Pandas wczytać zbiór danych z pliku CSV.

2. Obsługa brakujących wartości.
   Polega na sprawdzeniu, czy w zbiorze danych występują brakujące wartości, a następnie ich obsługa. W zależności od rodzaju danych i ich wartości może to być usunięcie wierszy z brakującymi wartościami lub wypełnie braków za pomocą metod takich jak średnia, mediana lub moda.

3. Sprawdzenie duplikatów.
   Polegać będzie na sprawdzeniu, czy w zbiorze danych występują zduplikowane rekordy i usunięciu ich, jeśli jest to konieczne.

4. Obsługa zmiennych kategorycznych.
   Obrabiany zbiór danych zawiera zmienne kategoryczne, takie jak płeć, rodzaj klienta, typ i klasa podróży oraz poziom statysfakcji. Konwersja zmiennych kategorycznych na zmienne numeryczne zostanie wykonana przy pomocy techniki zwanej kodowaniem jednoznacznym (ang. One-Hot Encoding).

5. Skalowanie cech.
   Niektóre z cech w zbiorze danych mogą mieć różne skale. Skalowanie cech może pomóc w zapewnieniu, że wszystkie cechy są w tej samej skali. Do realizacji tego zabiegu wykorzystana zostanie jedna z technik takich jak normalizacja lub standaryzacja.

6. Usuwanie niepotrzebnych kolumn.
   Polega na usunięciu ze zbiór danych kolumn, które nie są przydatne do analizy.

7. Sprawdzenie wartości odstających (ang. outliers).
   Jest to sprawdzenie, czy w zbiorze danych występują wartości odstające, a następnie ich obsługa. W zależności od rozpatrywanego przypadku może to być usunięcie wartości odstających lub przekształcenie ich za pomocą technik takich jak transformacja logarytmiczna.

Po ukończeniu powyższych kroków przetwarzania danych będzie można rozpocząć analizę zbioru danych.

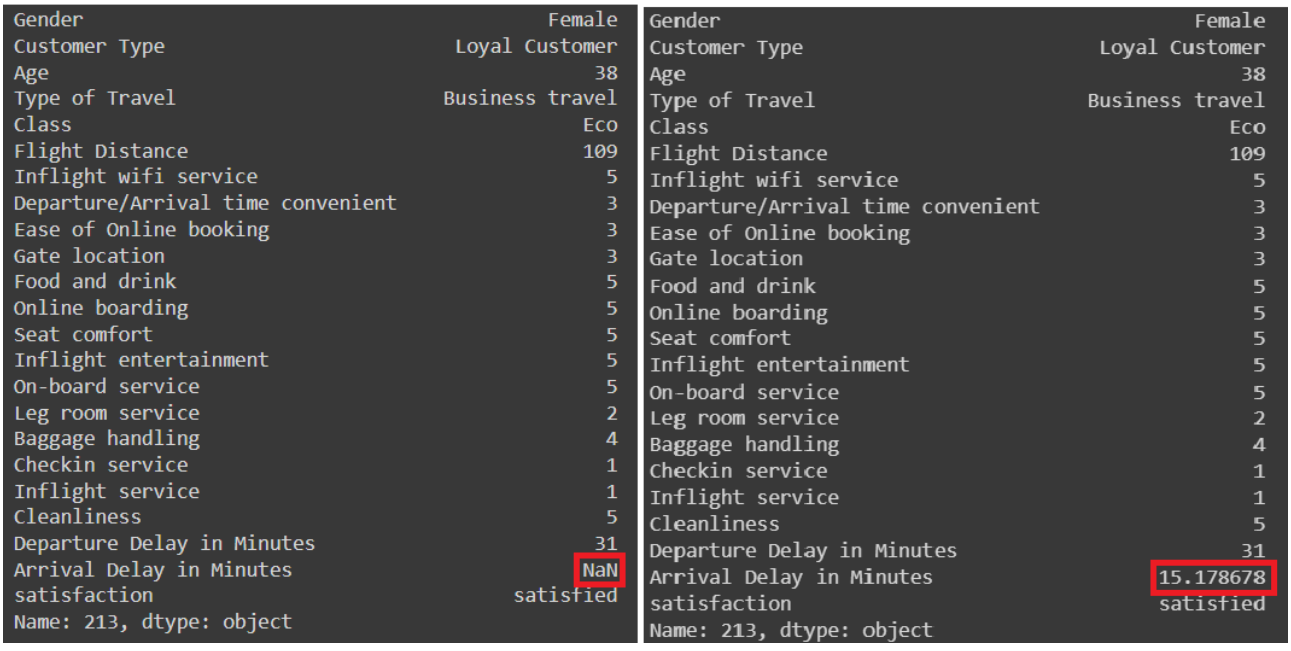
(Realizacja) Obróbce poddano zbiór danych treningowych. W zbiorze testowym zostały usunięte niepotrzebne kolumny, dodano nowy atrybut *'Total Delay in Minutes'* oraz zakodowano zmienne kategoryczne po to, aby kształt i rozmiar obu zbiorów danych był jednakowy.

1. Import zbioru danych.
   Dane zostały bezpośrednio wczytane z platformy Kaggle korzystając z API. Zbiór danych *train.csv* został wczytany do zmiennej `train_data`, zaś *test.csv* do `test_data`.

2. Obsługa brakujących wartości.
   W celu obsługi brakujących wartości konieczne było sprawdzenie, czy zbiór posiada braki oraz w jakich kolumnach one występują. W tym celu wykorzystano następujący kod - `train_data`*.isnull().sum()*. W efekcie otrzymano informację, że kolumna *'Arrival Delay in Minutes'* zawiera 310 wartości oznaczonych w Pythonie jako NaN (brak danych).

   Po zidentyfikowaniu tzw. null-i w ramce danych, przystąpiono do imputacji danych, czyli wypełniania wartości brakujących w zbiorze. W przypadku posiadanych danych można założyć, że braki nie niosą ze sobą żadnej istotnej informacji (MCAR - Missing completely at random), dlatego też zostały one uzupełnione wartością średnią przy pomocy klasy *SimpleImputer* z modułu sklearn.impute w bibliotece scikit-learn. Ze względu na to, że niektóre z użytych modeli nie obsługują brakujących wartość analogicznie postąpiono ze zbiorem testowym.

   Przykładowy efekt przed i po imputacji danych prezentuje rysunek 5.



Rys. 5: Dane przed i po imputacji

3. Sprawdzenie duplikatów.

W zbiorze danych treningowych i testowych nie zostały wykryte żadne duplikaty.

4. Obsługa zmiennych kategorycznych.
   Kolejnym krokiem w przygotowywaniu zbioru danych była zamiana zmiennych kategorycznych (rysunek 6) na liczbowe. W tym celu wykorzystano bibliotekę *sklearn*. Zawiera ona dwie klasy *OneHotEncoder* oraz *OrdinalEncoder* do obsługi zmiennych kategorycznych.

Column	Values
Gender	['Male' 'Female']
Customer Type	['Loyal Customer' 'disloyal Customer']
Type of Travel	['Personal Travel' 'Business travel']
Class	['Eco Plus' 'Business' 'Eco']

Rys. 6: Zmienne kategoryczne

(OneHotEncoder) OneHotEncoder tworzy nowe kolumny, tak aby reprezentować poszczególne wartości atrybutów jako wartości binarne (przypisuje wartość 1 do odpowiedniej kolumny dla danej kategorii, a 0 do pozostałych kolumn).

Za pomocą tej metody zakodowone zostały kolumny: *'Gender'*, *'Customer Type'*, *'Type of Travel'*. Efekt operacji prezentuje rysunek 7.

	Gender_Female	Gender_Male	Customer Type_Loyal Customer	Customer Type_disloyal Customer	Type of Travel_Business travel	Type of Travel_Personal Travel
0	0.0	1.0	1.0	0.0	0.0	1.0
1	0.0	1.0	0.0	1.0	1.0	0.0
2	1.0	0.0	1.0	0.0	1.0	0.0
3	1.0	0.0	1.0	0.0	1.0	0.0
4	0.0	1.0	1.0	0.0	1.0	0.0

Rys. 7: Metoda OneHotEncoding

(OrdinalEncoder OrdinalEncoder służy do kodowania zmiennych kategorycznych na wartości liczbowe, zachowując porządek między kategoriami. Działa poprzez przypisanie unikalnych liczb całkowitych do poszczególnych kategorii według ich porządku.

Za pomocą tej metody zakodowana została kolumna: *'Class'*, która ma swój naturalny porządek (*Eco* → *Eco Plus*→ *Business*). Efekt operacji prezentuje rysunek 8).

	Class
0	1.0
1	2.0
2	2.0
3	2.0
4	2.0

Rys. 8: Metoda OrdinalEncoder

Każdą z nowo powstałych tabel dołączono do danych treningowych, a następnie usunięto kolumny z niezakodowanymi danymi kategorycznymi. Analogicznie postąpiono ze zbiorem testowym.

Kolumnę z klasą predykcyjną dotyczącą satysfakcji klientów linii lotniczych zakodowano podczas badania korelacji między danymi (rysunek 9)

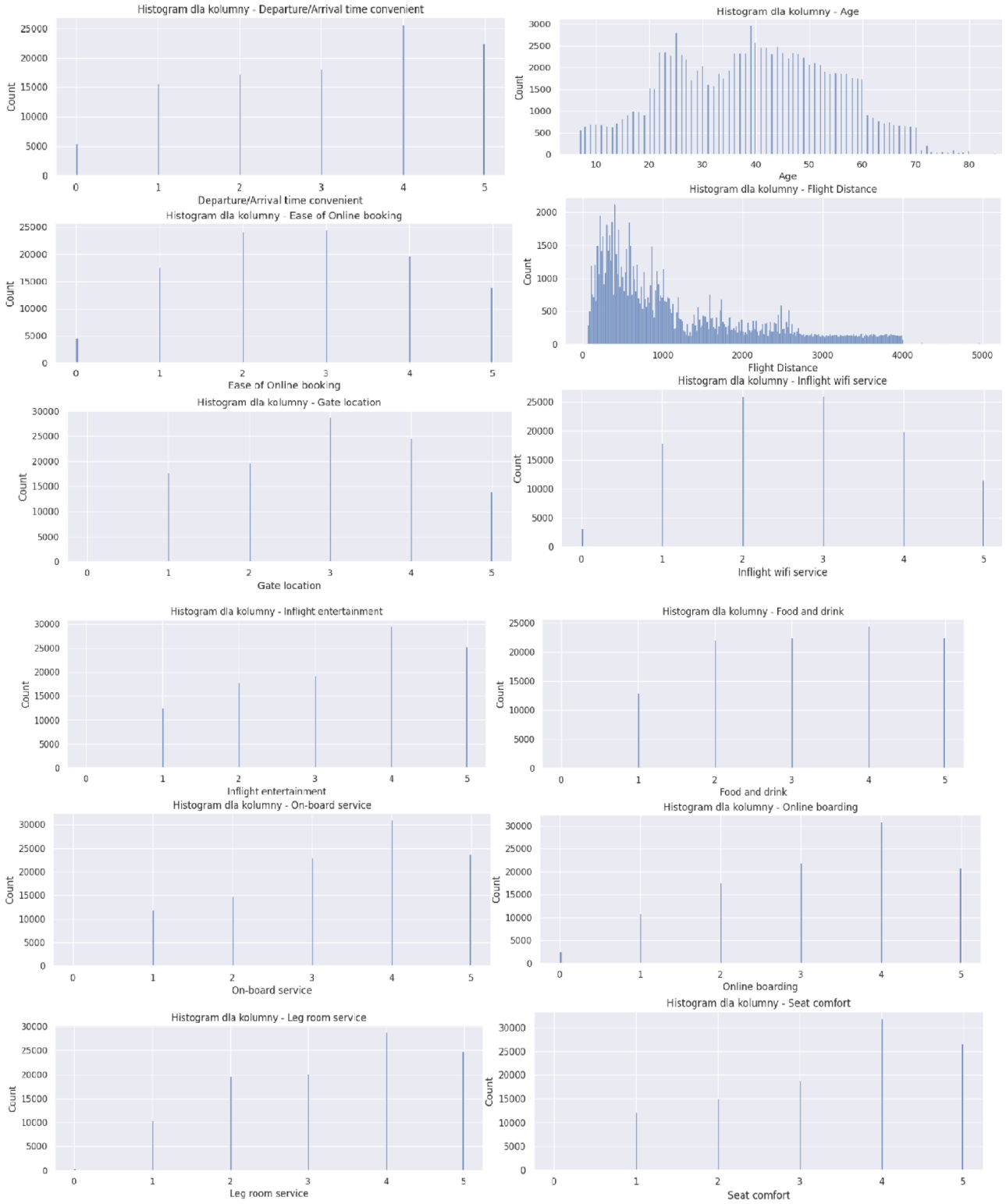
```
df['satisfaction'] = df['satisfaction'].map({'neutral or dissatisfied' : 0, 'satisfied' : 1})
```

Rys. 9: Kodowanie kolumny 'satisfaction'

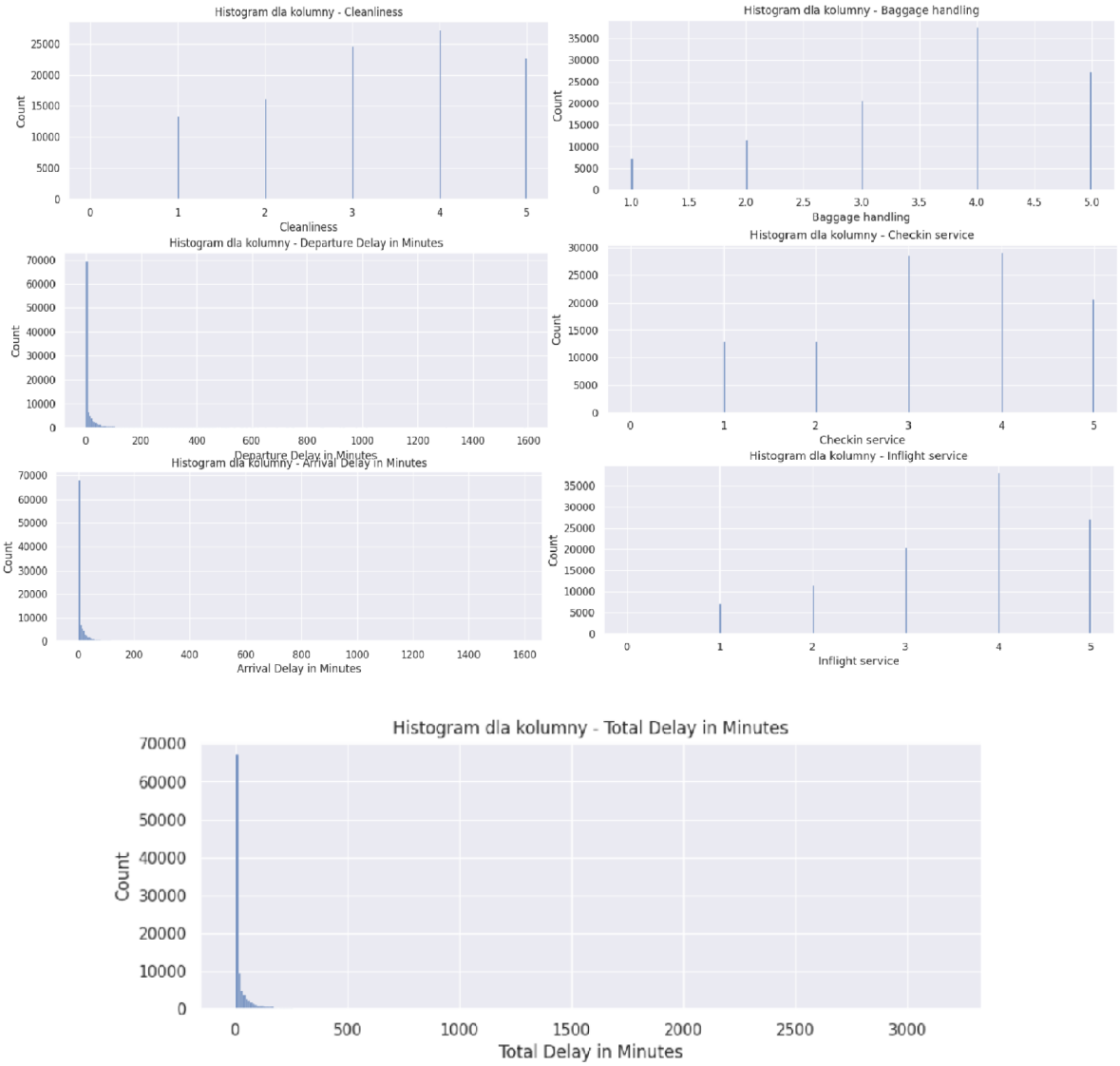
5. Skalowanie cech.
   Nie było konieczności skalowania cech.

6. Usuwanie niepotrzebnych kolumn.
   Zarówno w zbiorze treningowym, jak i testowym znajdowały się zbędne kolumny (*'Unnamed: 0'* oraz *'id'*) służące do indeksowania wierszy. Zostały one usunięte pomocy funkcji *DataFrame.drop()* z biblioteki Pandas.

7. Sprawdzenie wartości odstających (ang. outliers).
   Wartości odstające wyznaczono na podstawie histogramów dla danych numerycznych (rys. 10 i rys. 11). Następnie na badane kolumny nałożono maski, aby przyciąć ich zbiór wartości.



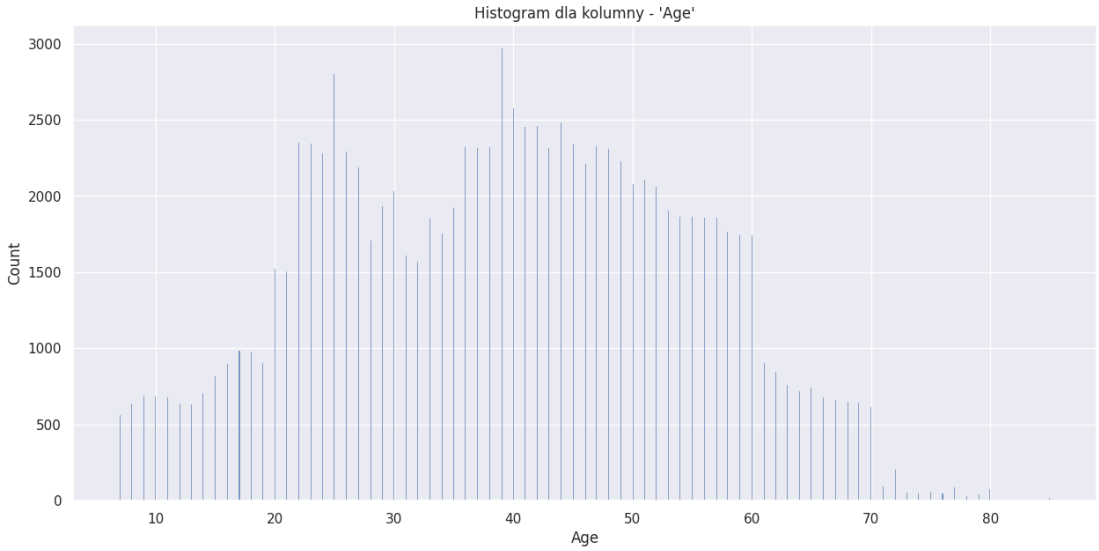
Rys. 10: Histogramy dla danych numerycznych



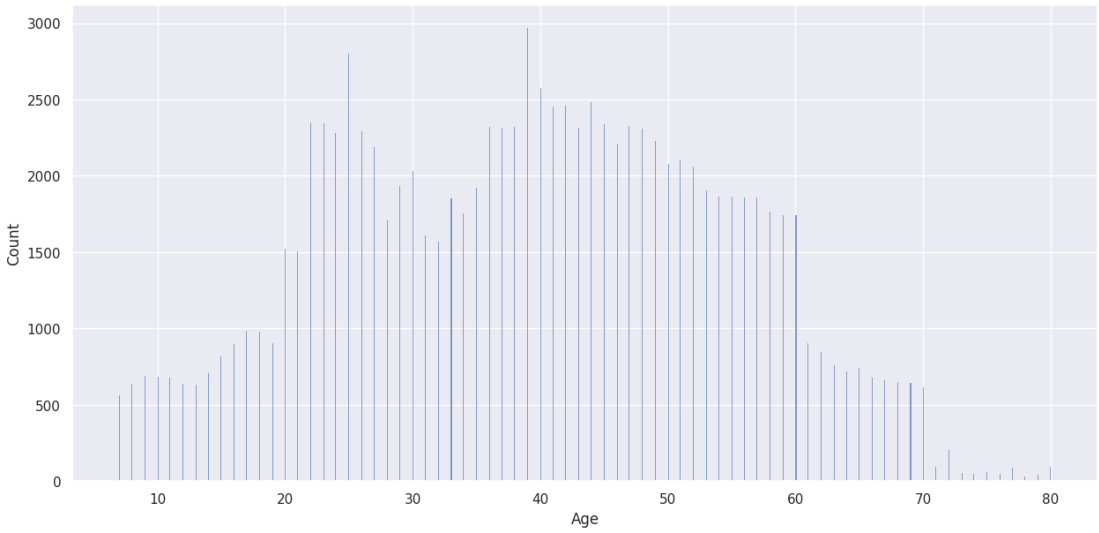
Rys. 11: Histogramy dla danych numerycznych

Na podstawie powyższych histogramów widać, że wartości odstające pojawiły się w kolumnach:

- *'Age'*
  W tym przypadku widać, że na rysunku 12 wartości powyżej wieku równego 80 lat stanowią mniejszość, stąd też decyzja o ich odrzuceniu rysunek 13. Przycięcia dokonano przy pomocy następującego kodu - *df['Age'] = df['Age'].mask(df['Age'] > 80, 80).*

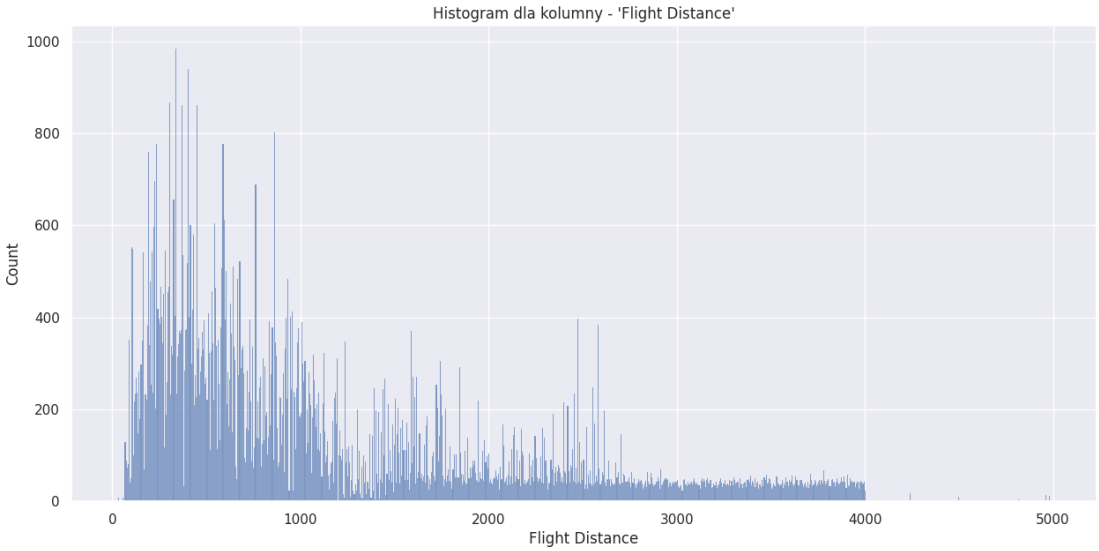


Rys. 12: Histogramy przed przycinaniem - kolumna 'Age'

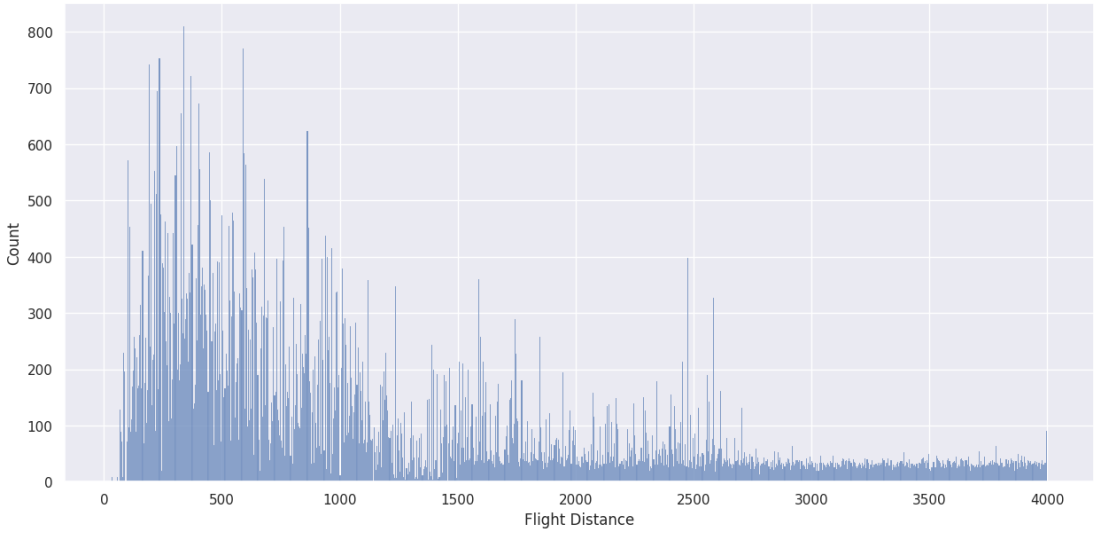


Rys. 13: Histogramy po przycinaniu - kolumna 'Age'

- *'Flight Distance'*
  Podobna sytuacja miała miejsce w tym przypadku (rys.14) - odległości lotów powyżej 4000 km stanowią pojedyncze wyjątki, które można pominąć (rys. 15). Wykorzystny kod - *df['Flight Distance'] = df['Flight Distance'].mask(df['Flight Distance'] > 4000, 4000).*



Rys. 14: Histogramy przed przycinaniu - kolumna 'Flight Distance'



Rys. 15: Histogramy po przycinaniu - kolumna 'Flight Distance'

5.5 Analiza zbioru danych

Analiza zbioru danych Airline Passenger Satisfaction będzie składać się z następujących kroków:

1. Wizualizacja danych.
   Będzie polegała na stworzeniu różnego rodzaju wizualizacji, takich jak histogramy, wykresy słupkowe, pudełkowy i kołowe.

2. Analiza porównawcza.
   Polegać będzie na porównaniu poszczególnych wartości danego atrybutu, aby zidentyfikować ewentualne różnice lub podobieństwa w poziomie zadowolenia klientów. Na przykład można porównać poziomy zadowolenia podróżujących służbowo z podróżującymi w celach rekreacyjnych lub poziomy zadowolenia pasażerów na różnych trasach lub korzystających z różnych lini lotniczych.

3. Statystyka opisowa.
   Polegać będzie na obliczeniu podstawowych statystyk opisujących zbiór danych, takich jak średnia, mediana, odchylenie standardowe i zakres dla każdej zmiennej liczbowej w zbiorze danych.

4. Selekcji atrybutów

   - Selekcja atrybutów na podstawie korelacji - polega na wyborze atrybutów, które są silnie skorelowane ze zmienną docelową lub mają wysoki poziom korelacji z innymi atrybutami. Do realizacji tego typu selekcji wykorzystana zostanie korelacja Pearsona oraz współczynnik zgodności rang Spearmana.
   - Selekcja atrybutów na podstawie metod filtrowania - polega na wyborze atrybutów na podstawie pewnych kryteriów, takich jak wartości bezwzględne lub wartości względne atrybutów, i odrzuceniu tych, które nie spełniają kryteriów. Do realizacji tego typu selekcji wykorzystana zostanie metoda chi-kwadrat oraz ANOVA Test.

(Realizacja) Kolejnym etapem w realizacji projekty była analiza i eksploatacja obrobionego zbioru danych treningowych.

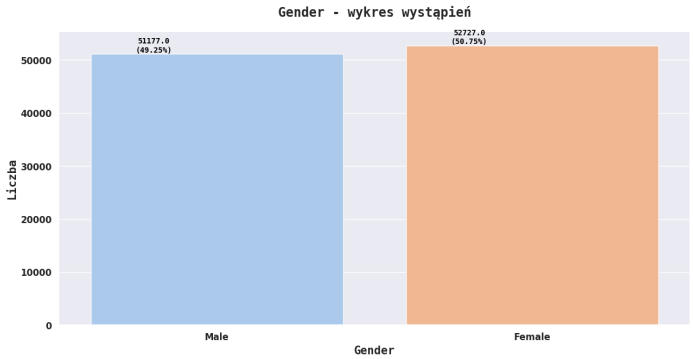
1. Wizualizacja danych.

   Do dokumentacji projektowej dołączony został plik *'Airline-Passenger-Satisfaction-Data-Report.html'*, który zawiera sumaryczny przegląd obrobionego zbioru danych. Raport ten został wygenerowany przy pomocy biblioteki *pandas profiling*.

   Do analizy i wizualizacji zbioru danych wykorzystano dwie funkcje - *count* odpowiedzialną za zliczanie ilości danej wartości atrybutu i prezentacja jej w stosunku do całego zbioru danych oraz *kde* tworząca wykres gęstości jądrowej (Kernel Density Estimation - KDE).

   Dokonana analiza prezentuje się następująco:

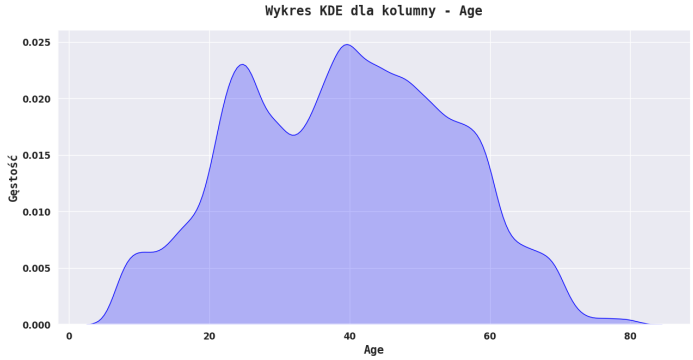
- Atrybut 'Gender' (rys. 16)



Rys. 16: Kolumna 'Gender'

Widać, że liczba mężczyzn i kobiet jest zbliżona, a więc zbiór danych jest zrównoważony względem płci.

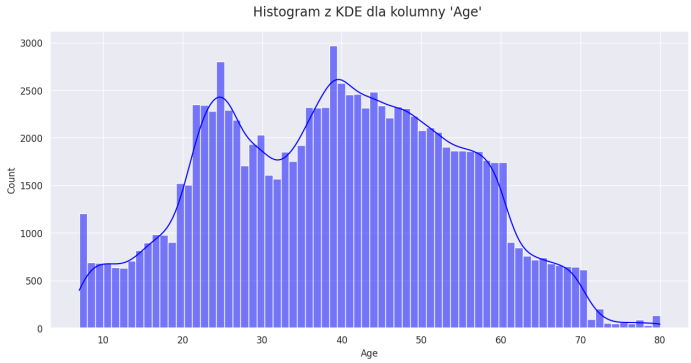
- Atrybut 'Age' (rys. 17)



Rys. 17: Kolumna 'Age'

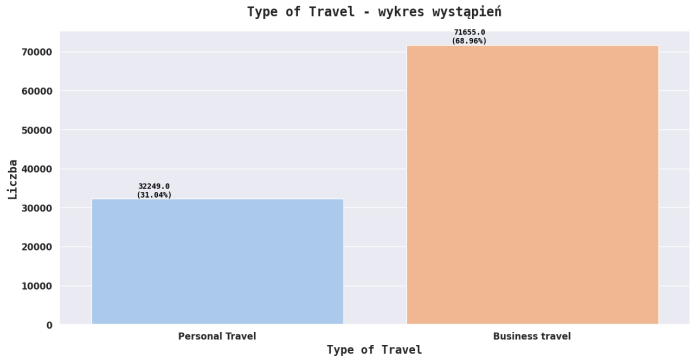
W badaniu brały udział osoby w różnym wieku, ale widać, że osoby w wieku około 25 i 40 lat podróżowały najwięcej.

Otrzymany na wykresie rozkład wygląda na dość symetryczny względem wartości 40 lat.



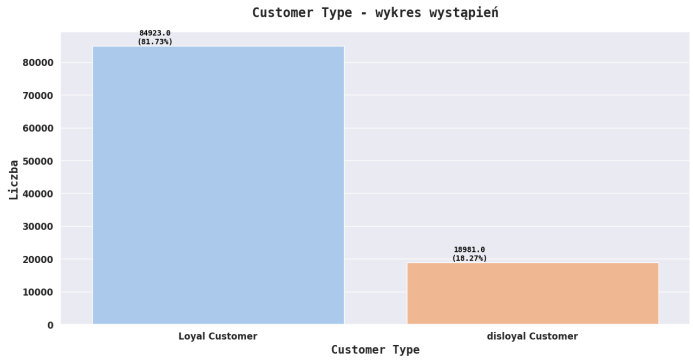
Rys. 18: Histogram z KDE dla kolumny 'Age'

- Atrybut 'Type of Travel' (rys. 19)



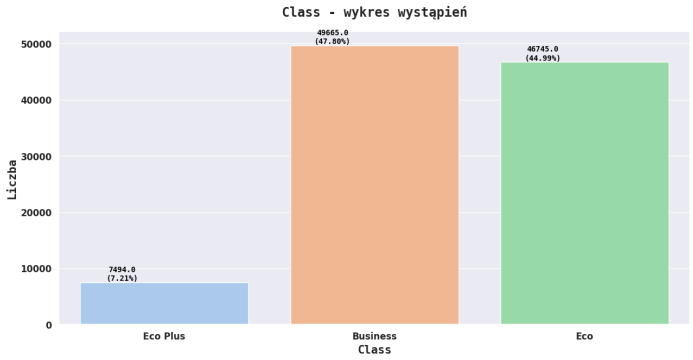
Rys. 19: Kolumna 'Type of Travel'

- Atrybut 'Customer Type' (rys. 20)



Rys. 20: Kolumna 'Customer Typ'

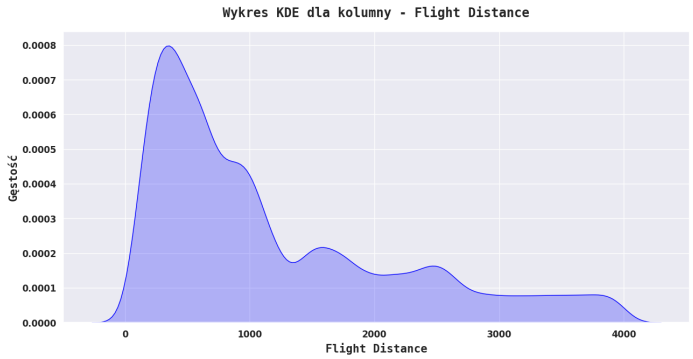
- Atrybut 'Class' (rys. 21)



Rys. 21: Kolumna 'Class'

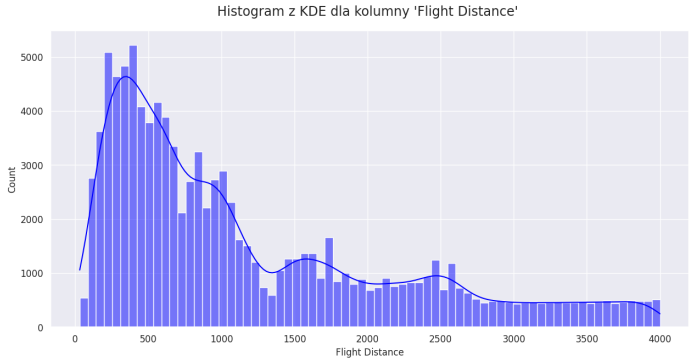
Większość wszystkich podróży samolotem odbywała się w klasie Eco lub Biznesowej. Klasa Eco Plus stanowi nieliczną grupę.

- Atrybut 'Flight Distance' (rys. 22)



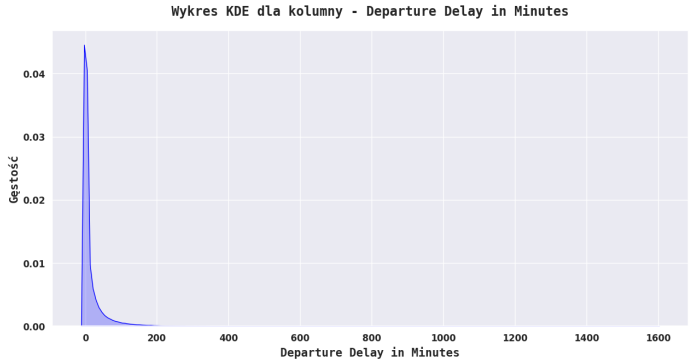
Rys. 22: Kolumna 'Flight Distance'

Większość podróży dotyczyła odległości około 500 kilometrów.



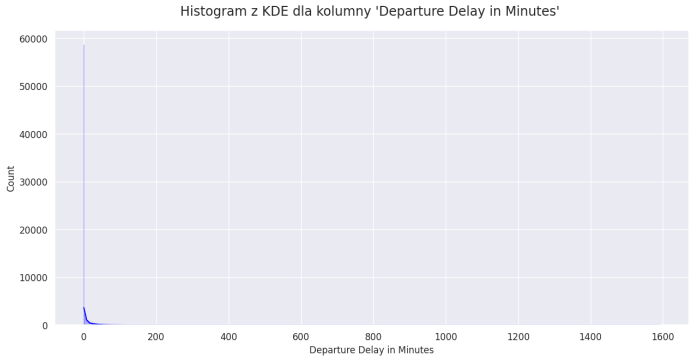
Rys. 23: Histogram z KDE dla kolumny 'Flight Distance'

- Atrybut 'Departure Delay in Minutes' (rys. 24)



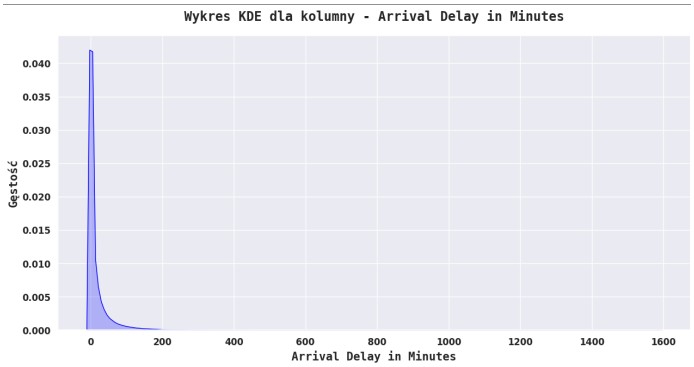
Rys. 24: Kolumna 'Departure Delay in Minutes'

Wygląda na to, że w większości przypadków nie było opóźnienia odlotu.



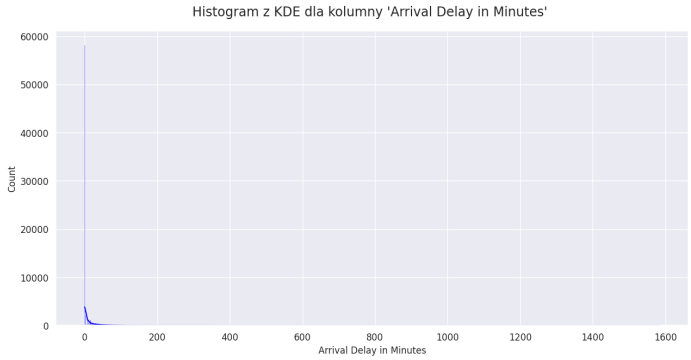
Rys. 25: Histogram z KDE dla kolumny 'Departure Delay in Minutes'

- Atrybut 'Arrival Delay in Minutes' (rys. 26)



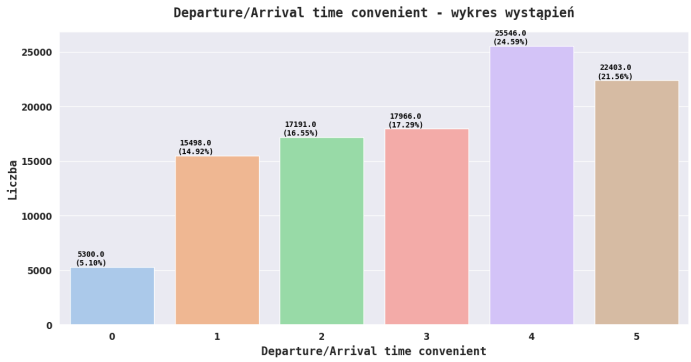
Rys. 26: Kolumna 'Arrival Delay in Minutes'

Podobnie jak dla opóźnienie odlotu, w tym przypadku również większość przylotów nie miała opóźnienia.



Rys. 27: Histogram z KDE dla kolumny 'Arrival Delay in Minutes'

- Atrybut 'Departure/Arrival time convenient' (rys. 28)

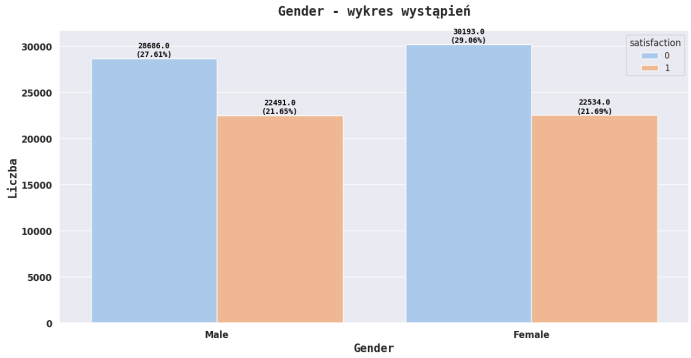


Rys. 28: Kolumna 'Departure/Arrival time convenients'

Większość ludzi ocenia czasu odlotu i przylotu na 4 lub 5.

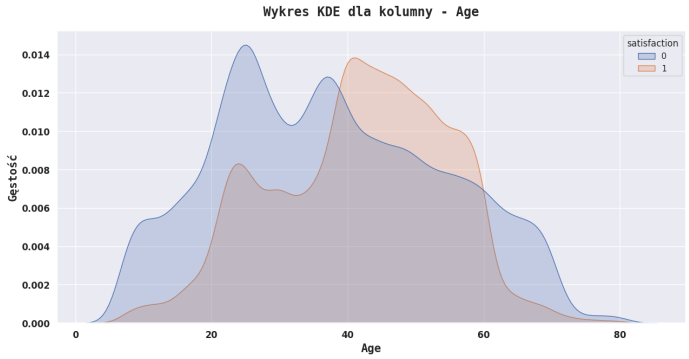
2. Analiza porównawcza.

- Atrybut 'Gender' w stosunku do satysfakcji (rys. 29)



Rys. 29: Kolumna 'Gender' w stosunku do satysfakcji

- Atrybut 'Age' w stosunku do satysfakcji (rys. 30)

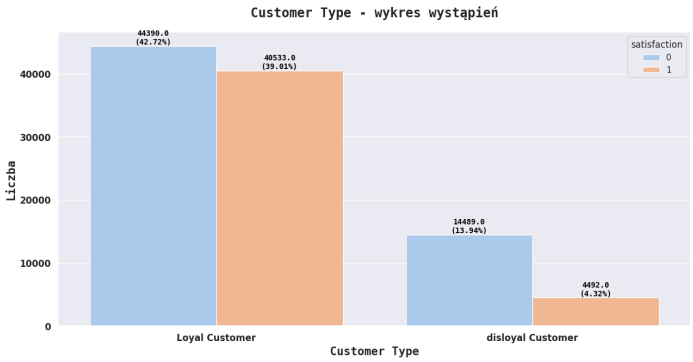


Rys. 30: Kolumna 'Age' w stosunku do satysfakcji

Można powiedzieć, że zadowolonych ludzi jest więcej od niezadowolonych.

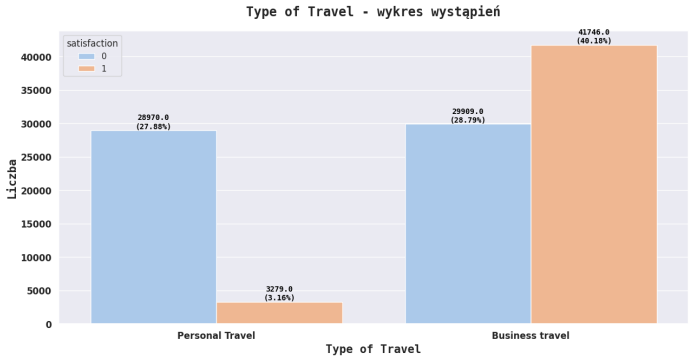
Zadowoleni ludzie mają około 40-56 lat, zaś niezadowoleni około 20-40 lat

- Atrybut 'Customer Type' w stosunku do satysfakcji (rys. 31)



Rys. 31: Kolumna 'Customer Type' w stosunku do satysfakcji

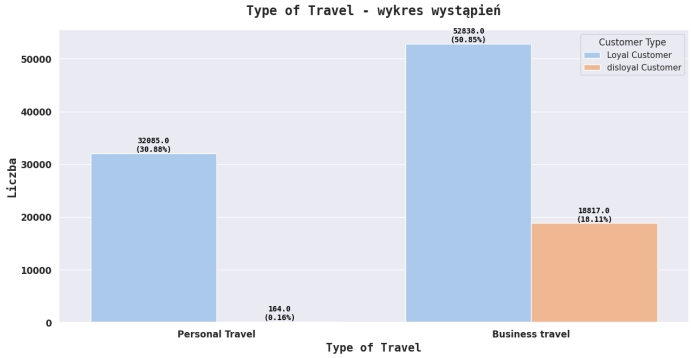
- Atrybut 'Type of Travel' w stosunku do satysfakcji (rys. 32)



Rys. 32: Kolumna 'Type of Travel' w stosunku do satysfakcji

Rodzaj podróży ma duży wpływ na zadowolenie klientów linii lotniczych. Większość osób podróżujących prywatnie jest niezadowolona, zaś przeciwna sytuacja jest w przypadku osób podróżujących biznesowa. W tym przypadku większość stanowią osoby zadowolone z wyprawy.

- Atrybut 'Type of Travel' w stosunku do 'Customer Type' (rys. 33)



Rys. 33: Kolumna 'Type of Travel' w stosunku do 'Customer Type'

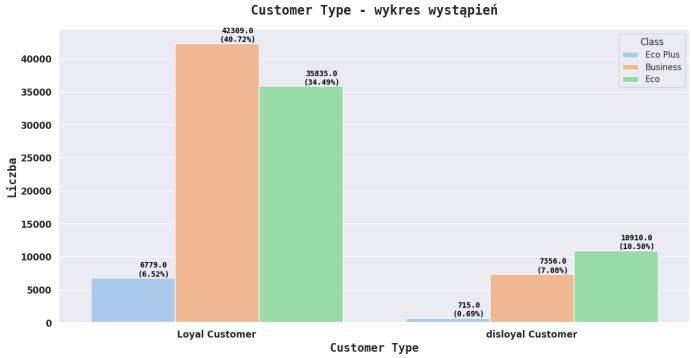
W obu rodzajach podróży (prywatnej/biznesowej) dominują lojalni klienci. W przypadku prywatnych podróży są to w przybliżeniu wszyscy klienci.

- Atrybut 'Class' w stosunku do satysfakcji (rys. 34)

Rys. 34: Kolumna 'Class' w stosunku do satysfakcji

Można zauważyć, że w klasie ekonomicznej jest dużo niezadowolonych osób. Również w klasie Eco Plus większość stanowią osoby niezadowolone z podróży. Jedynie w klasie biznesowej większość osób jest zadowolona

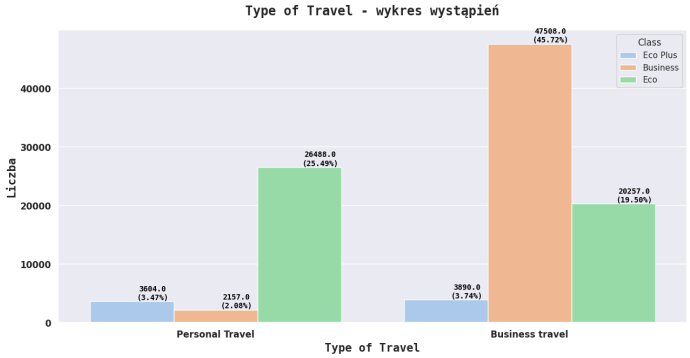
- Atrybut 'Customer Type' w stosunku do 'Class' (rys. 35)



Rys. 35: Kolumna 'Customer Type' w stosunku do 'Class'

Większość lojalnych klientów podróżuje klasą biznesową, zaś nielojalnych klasą ekonomiczną.

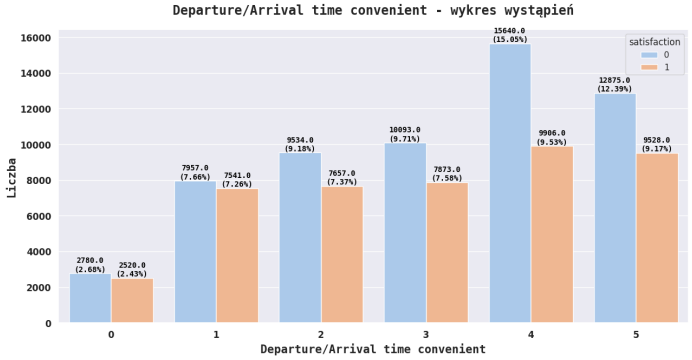
- Atrybut 'Type of Travel' w stosunku do 'Class' (rys. 36)



Rys. 36: Kolumna 'Type of Travel' w stosunku do 'Class'

Widać, że większość podróży prywatnych odbywa się w klasie ekonomicznej, co zgadza się z faktem, że zarówno podróże osobiste, jak i klasa Eco mają wiele niezadowolonych osób. Z drugiej strony większość podróży służbowych odbywa się w klasie biznes.

- Atrybut 'Departure/Arrival time convenient' w stosunku do satysfakcji (rys. 37)

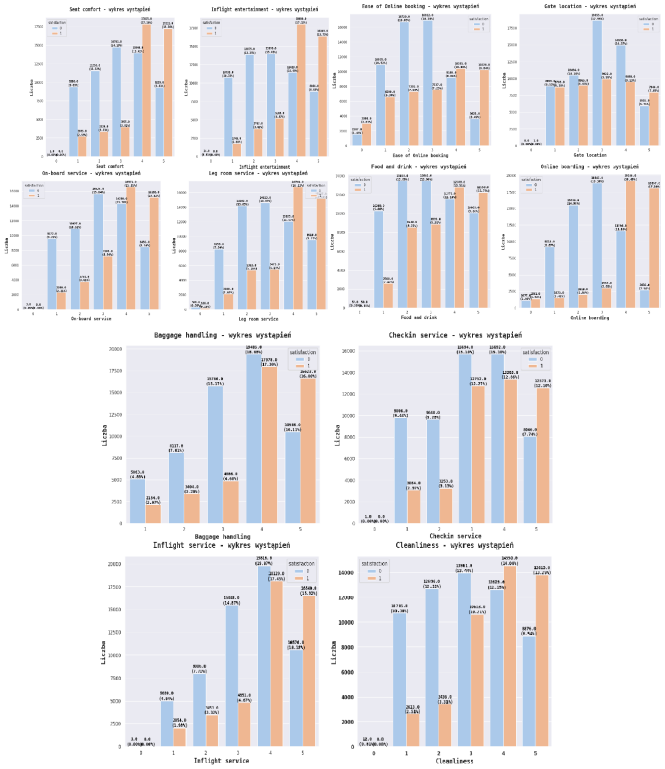


Rys. 37: Kolumna 'TDeparture/Arrival time convenient' w stosunku do satysfakcji

- Atrybut z serwisem pokładowym i nie tylko w stosunku do satysfakcji (rys. 38)

  Są to kolumny takie jak: 'Ease of Online booking', 'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'On-board service', 'Leg room service', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cleanliness'.

  ??)



Rys. 38: Kolumna związane z serwisem w stosunku do satysfakcji

Najczęstsza ocena jaka była wystawiana dla świadczonych usług związanych z serwisem pokładowym i nie tylko wynosiła 4.

3. Statystyka opisowa.

W ramach statystyki opisowej wyznaczono podstawowe statystyki opisujące zbiór danych, takie jak średnia, mediana, odchylenie standardowe i zakres dla każdej zmiennej liczbowej w zbiorze danych. Wyniki przedstawiono w tabelach 3, 4, 5, 6, 7.

Dokładna analiza każdego atrybutu umieszczona je w notatniku, ze względu na duży rozmiar tych plików nie umieszczamy ich w tej dokumentacji.

Tabela 3: Podstawowe statystyki zbioru treningowego

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient
count	103904	103904	103904	103904
mean	39,38	1189,08	2,73	3,06
std	15,11	995,97	1,33	1,53
min	7	31	0	0
25%	27	414	2	2
50%	40	843	3	3
75%	51	1743	4	4
max	80	4000	5	5

Tabela 4: Podstawowe statystyki zbioru treningowego

	Ease of Online booking	Gate location	Food and drink	Online boarding
count	103904	103904	103904	103904
mean	2,76	2,98	3,20	3,25
std	1,40	1,28	1,33	1,35
min	0	0	0	0
25%	2	2	2	2
50%	3	3	3	3
75%	4	4	4	4
max	5	5	5	5

Tabela 5: Podstawowe statystyki zbioru treningowego

	Seat comfort	Inflight entertainment	On-board service	Leg room service
count	103904	103904	103904	103904
mean	3,44	3,36	3,38	3,35
std	1,32	1,33	1,29	1,32
min	0	0	0	0
25%	2	2	2	2
50%	4	4	4	4
75%	5	4	4	4
max	5	5	5	5

Tabela 6: Podstawowe statystyki zbioru treningowego

	Baggage handling	Checkin service	Inflight service	Cleanliness
count	103904	103904	103904	103904
mean	3,63	3,30	3,64	3,29
std	1,18	1,27	1,18	1,31
min	1	0	0	0
25%	3	3	3	2
50%	4	3	4	3
75%	5	4	5	4
max	5	5	5	5

Tabela 7: Podstawowe statystyki zbioru treningowego

	Departure Delay in Minutes	Arrival Delay in Minutes	Total Delay in Minutes	satisfaction
count	103904	103904	103904	103904
mean	14,82	15,18	29,99	0,43
std	38,23	38,64	76,12	0,50
min	0	0	0	0
25%	0	0	0	0
50%	0	0	2	0
75%	12	13	24	1
max	1592	1584	3176	1

4. Selekcji atrybutów

Analizie poddano zbiór danych treningowych. Na histogramach z rysunków 10, 11, zobrazowano częstość występowania poszczególnych wartości dla danych numerycznych.

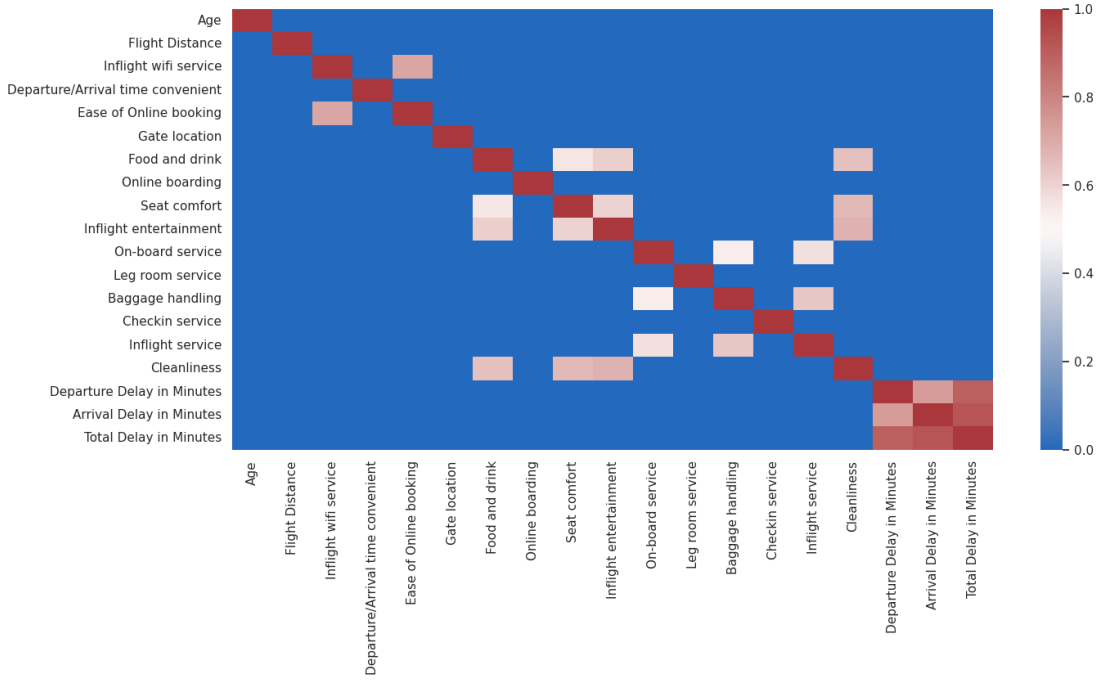
W ramach szukania zależności pomiędzy poszczególnymi atrybutami wyznaczono różne typy korelacji w tym:

- korelacja spearmana (rys. 39)
- korelacja pearsona (rys. 40)
- korelacja kendalla (wyniki zbliżone do tych z rys. 39, 40)

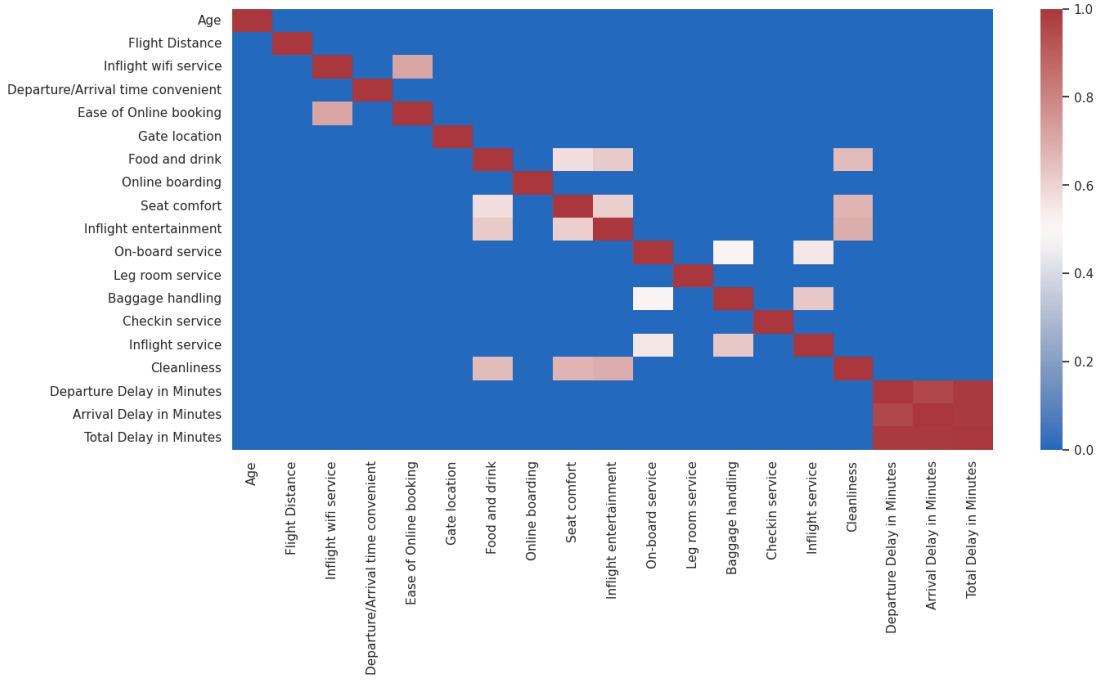
Istnieje widoczna zależność między kolumnami 'Arrival Delay in Minutes' , 'Departure Delay in Minutes' oraz 'Total Delay in Minutes'. Wykonano więc testy badające zależności między wartościami tych atrybutów. Wyniki przestawione na rysunkach 41, 42, 43 potwierdzają silną zależność między tymi atrybutami.

Zgodnie z oczekiwaniami, opóźnienie wylotu jest silnie skorelowane z opóźnieniem przylotu. Widać również, że opóźnienie odlotu jest w przybliżeniu równe opóźnieniu przylotu (wykres o kształcie w przybliżeniu liniowym). W związku z tym opóźnienie ma miejsce tylko przed rozpoczęciem lotu (nie w trakcie lotu).

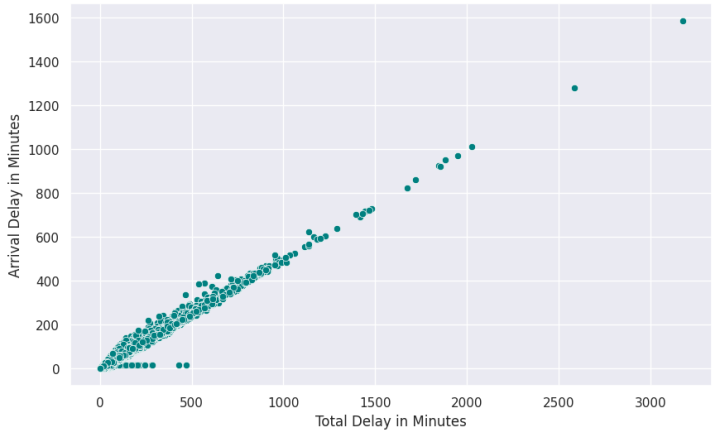
Zarówno opóźnienie odlotu i przylotu również silnie wpływa na całkowitą wartość opóźnienia lotu.



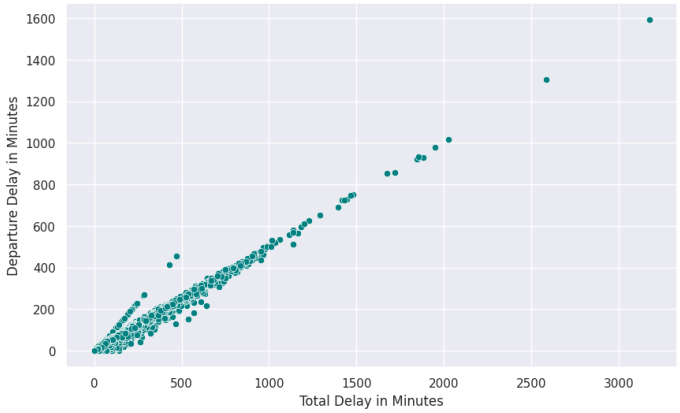
Rys. 39: Korelacja spearmana dla danych treningowych



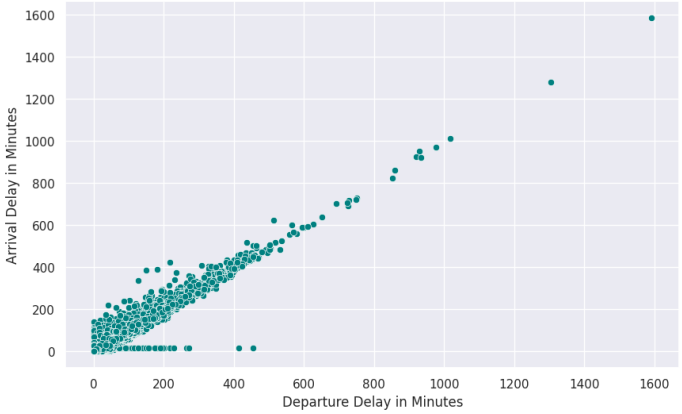
Rys. 40: Korelacja pearsona dla danych treningowych



Rys. 41: Korelacja między kolumnami 'Arrival Delay in Minute' z 'Total Delay in Minutes'

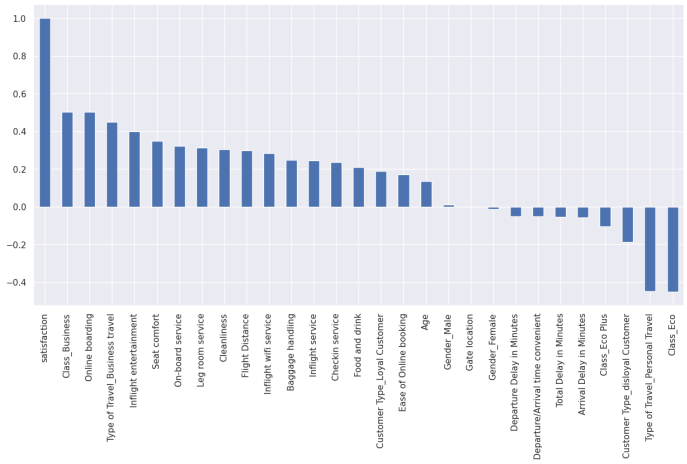


Rys. 42: Korelacja między kolumnami 'Total Delay in Minutes' z 'Departure Delay in Minutes'



Rys. 43: Korelacja między kolumnami 'Arrival Delay in Minute' z 'Departure Delay in Minutes'

Obliczono macierz korelacji dla wszystkich kolumn w ramce danych treningowy ('df_dummies'), z której wybrano korelacje względem kolumny 'satisfaction', wyniki przedstawiono na rysunku 44.



Rys. 44: Korelacja względem kolumny 'satisfaction'

Na podstawie wyników z rysunku 44 można stwierdzić, że w przypadku wartości: klasa biznesowa, boarding online, biznesowy typ podróży oraz wygodne siedzenia, istnieje silna zależność (rośnie zadowolenie z usługi). Płeć, wiek oraz 'Gate location' nie mają praktycznie żadnego wpływu na poziom satysfakcji. Silną ujemną zależność liniową zauważamy w przypadku wartości 'Class Eco' oraz 'Type of Travel Personal Travel'.

6 Eksperymenty i metryki klasyfikacji

W ramach eksperymentów ocenione zostały wyniki klasyfikacji uzyskane dla poszczególnych algorytmów uczenia maszynowego opisanych w rodziale 2. Zbadano wpływ nastaw poszczególnych hiperparametrów. Oceny jakości modeli dokonano wykorzystując odpowiednie metryki, opisane w kolejnym podrozdziale.

6.1 Wyznaczane metryki klasyfikacji [19]

6.1.1 Macierz pomyłek (ang. confusion matrix)

Do analizy poprawności klasyfikacji przydatna może być macierz pomyłek - pokazująca wartości błędnie sklasyfikowane na rzecz innych klas w wygodnej do analizy formie tabelarycznej. Tabela ta zawiera cztery wartości: poprawne odpowiedzi (ang. TP – true positive), fałszywe pozytywy (ang. FP – false positive), fałszywe negatywy (ang. FN – false negative) i poprawne negatywne (ang. TN – true negative). Macierz pomyłek prezentuje tabela 8

Tabela 8: Macierz pomyłek

	Klasa 0	Klasa 1
Klasa 0	TN	FP
Klasa 1	FN	TP

6.1.2 Dokładność (ang. accuracy)

Accuracy czyli dokładność jest podstawową metryką używaną do oceny jakości klasyfikacji wskazującą, ile przypadków zostało poprawnie przewidzianych przez model. Jest to stosunek liczby poprawnych przewidywań do całkowitej liczby przypadków.

$$accuracy = \frac{\text{liczba poprawnych predykcji}}{\text{liczba wszystkich predykcji}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$error = 1 - accuracy = \frac{\text{liczba błędnych predykcji}}{\text{liczba wszystkich predykcji}} = \frac{FP + FN}{TP + TN + FP + FN} \tag{4}$$

6.1.3 Precyzja (ang.precision)

Kolejną metryką do oceny jest precyzja - ang. precision, zdefiniowana jako stosunek liczby poprawnie sklasyfikowanych przykładów (True Positive - TP) do liczby wszystkich przewidzianych przykładów Positive:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} = \frac{TP}{TP + FP} \tag{5}$$

6.1.4 Odzysk (ang. recall, TP rate, sensitivity)

Następnie możemy zdefiniować czułość ang. recall jako stosunek poprawnie sklasyfikowanych przykładów (TP) do wszystkich przykładów, które powinny zostać sklasyfikowane jako Positive(TP + FN):

$$Recall = sensitivity = TPrate = \frac{TruePositives}{TruePositives + FalseNegatives} = \frac{TP}{TP + FN} \tag{6}$$

6.1.5 Miara F1

Metryka F1 jest średnią harmoniczną precyzji i czułości:

$$F1 = \frac{1}{\frac{\frac{1}{recall} + \frac{1}{precision}}{2}} + \frac{2 \cdot recall \cdot precision}{recall + precision} \tag{7}$$

6.1.6 Krzywa ROC i ROC AUC

Krzywa ROC (ang. Receiver Operating Characteristic) jest wykresem zależności miary TP rate (True Positive Rate) od miary FP rate (False Positive Rate). Pozwala ona na porównanie klasyfikatorów dla różnych wartości progowych. Można w ten sposób zbadać zdolność klasyfikatora do rozróżniania klas i wybrać model, który ma najwyższe pole AUC (ang. Area Under the Curve) pod krzywą ROC.

ROC AUC jest wartością z zakresu od 0 do 1, gdzie 0 oznacza, że klasyfikator jest całkowicie niezdolny do rozróżniania klas, a 1 oznacza, że klasyfikator jest doskonały. Im większa wartość ROC AUC, tym lepsza jakość klasyfikatora.

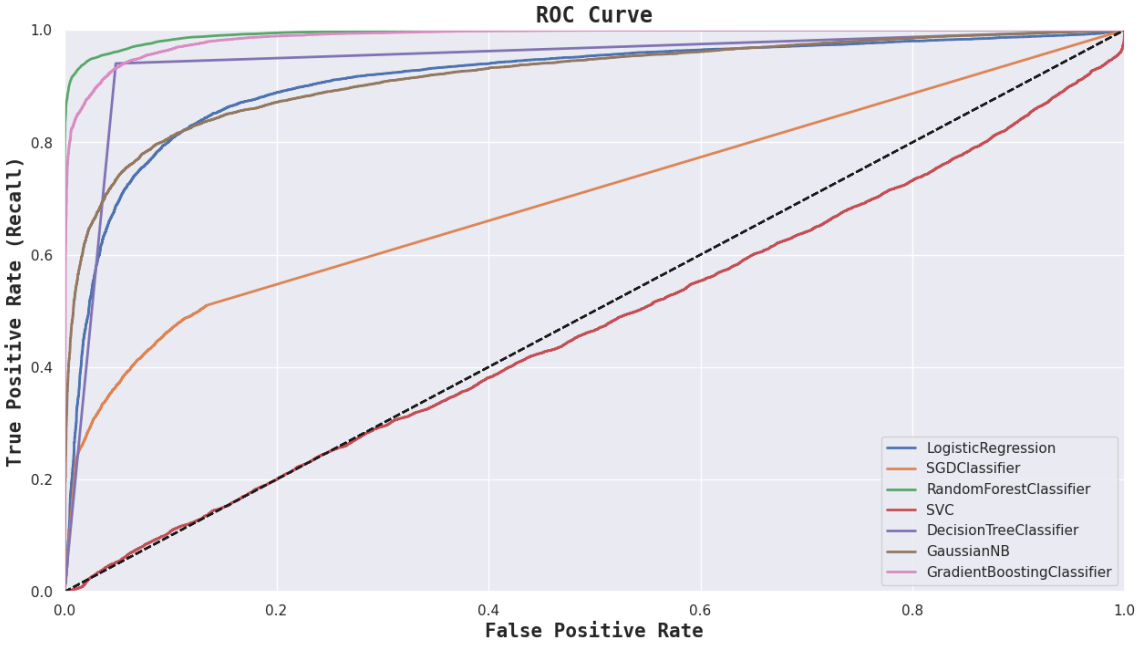
7 Testy i uzyskane wyniki

Testy modeli przeprowadzono dla bazowych ustawień hiperparametrów oraz dla modeli z różnymi ich nastawami. Uzyskane wyniki przedstawiono w poszczególnych tabelach oraz na wykresach. Wybrame hiperparametry:

- **Logistic Regression**: 'max_iter': [100, 500, 1000], 'C': [0.1, 1.0, 10.0],
- **SGD Classifier**: 'max_iter': [100, 500, 1000], 'alpha': [0.0001, 0.001, 0.01],
- **Random Forest**: 'n_estimators': [100, 200, 500], 'max_depth': [None, 10, 20],
- **SVC**: 'C': [0.1, 1.0, 10.0], 'kernel': ['linear', 'rbf'],
- **Decision Tree**: 'max_depth': [None, 10, 20], 'min_samples_split': [2, 5, 10],
- **Gaussian Naive Bayes**: 'var_smoothing': [1e-9, 1e-6, 1e-3],
- **Gradient Boosting**: 'n_estimators': [100, 200, 500], 'learning_rate': [0.01, 0.1, 1.0]

7.1 Wyniki dla bazowych ustawień modeli

Wszystkie modele przetestowano dla tego samego zbioru testowego.



Rys. 45: Krzywa ROC dla bazowych wartości hiperparametrów

Tabela 9: Uzyskane wyniki dla każdego modelu z bazowymi ustawieniami na zbiorze testowym, posortowane względem kolumny AUC

Model (parametry bazowe)	Accuracy	Precision	Recall	F1 Score	AUC
Random Forest Class	0,96	0,97	0,94	0,96	0,994
Gradient Boosting Class	0,94	0,95	0,92	0,93	0,9877
DecisionTreeClassifier	0,95	0,94	0,94	0,94	0,95
GaussianNB	0,86	0,84	0,83	0,84	0,92
Logistic Regression	0,86	0,84	0,83	0,84	0,92
SGD Classifier	0,66	0,94	0,25	0,39	0,70
SVM	0,45	0,44	0,88	0,59	0,47

WNIOSKI
Analizując tabelę 9 algorytmy uczenia maszynowego można podzielić na trzy grupy:

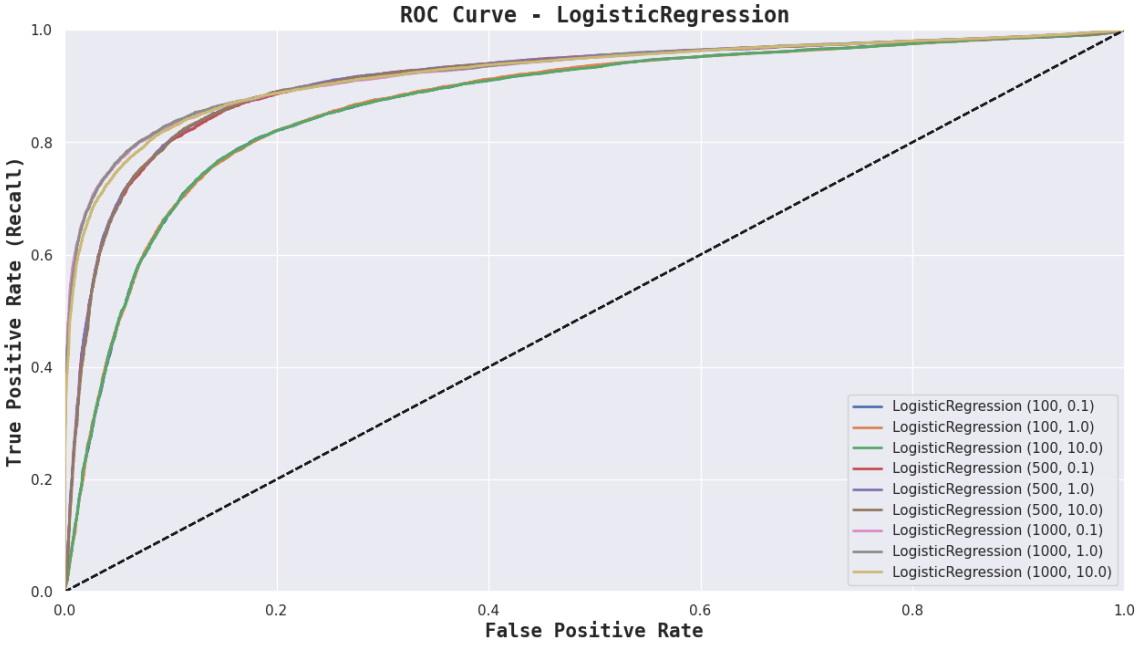
1. Modele bardzo dobre (dokładność 0.9): Las Losowy (dokładność: 0.96), Drzewo Decyzyjne (dokładność: 0.95) i Gradient Boosting (dokładność: 0.94).
2. Modele dobre (0.9 ¿ dokładność 0.8): Naiwny Bayes (dokładność: 0.86) i Regresja Logistyczna (dokładność: 0.86)
3. Modele słabe (0.8 ¡ dokładność): Algorytm SGD (dokładność: 0.66) i SVM (dokładność: 0.45)

Widać również, że mimo ustawień bazowych otrzymane wyniki, dla części modeli nie wymagają ich strojenia.

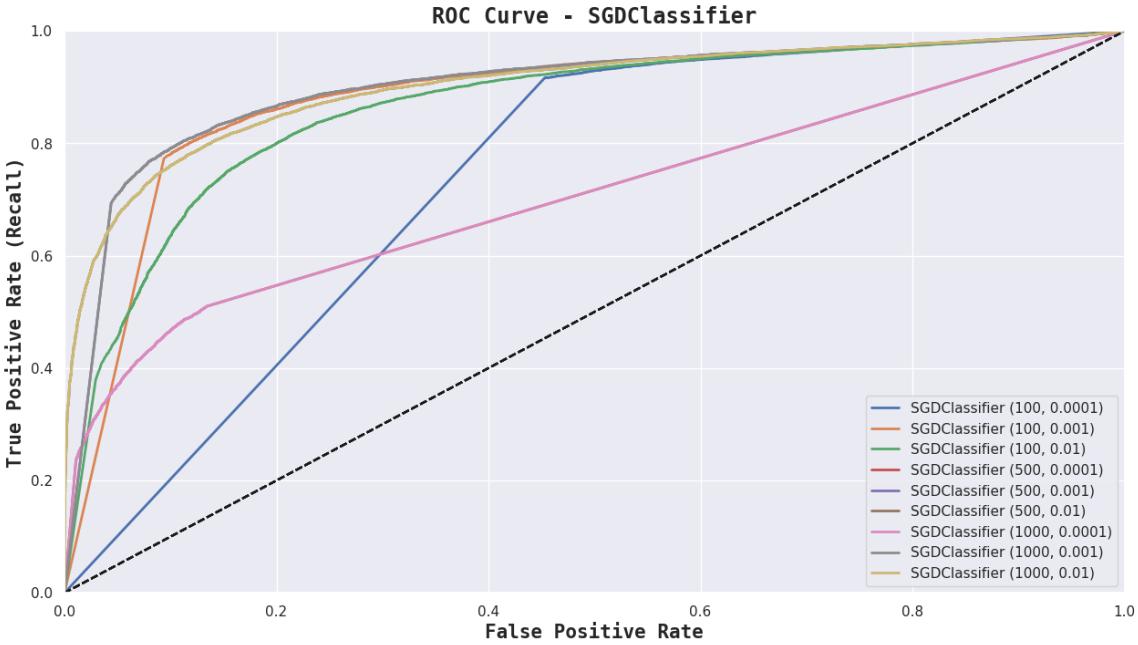
7.2 Wyniki różnych wartości hiperparametrów

Tabela 10: Uzyskane najlepsze wyniki modeli z różnymi ustawieniami. Ewaluowane na zbiorze testowym, posortowane względem kolumny AUC

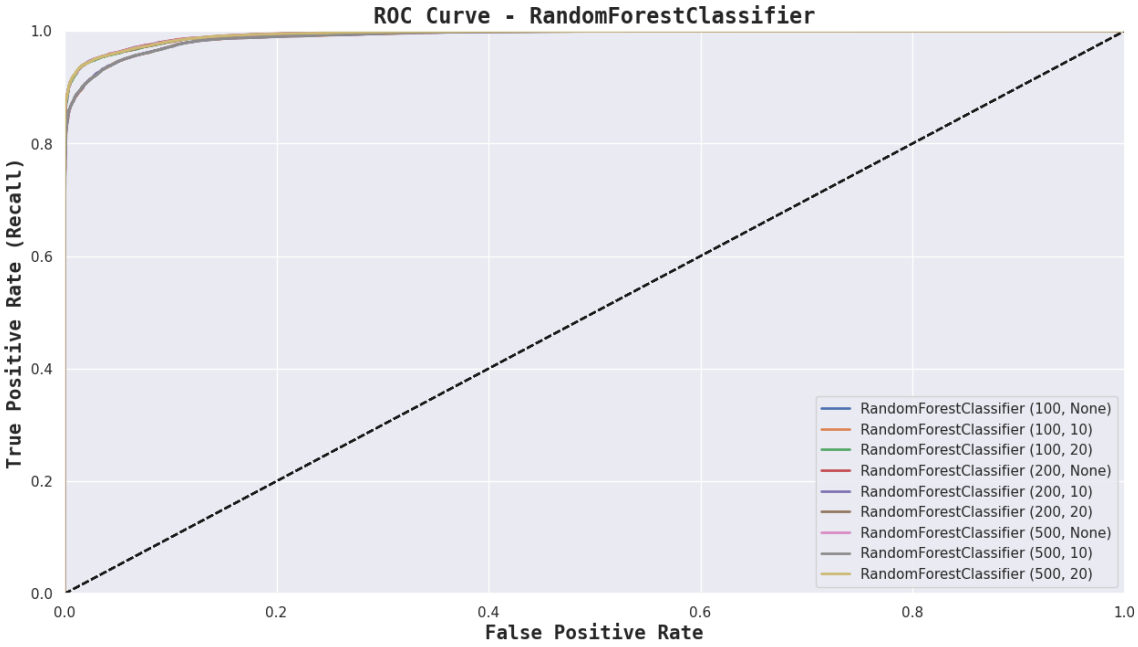
Model (parametry)	Accuracy	Precision	Recall	F1 Score	AUC
RandomForestClassifier (500, None)	0.963543	0.972865	0.943261	0.957834	0.994130
GradientBoostingClassifier (500, 0.1)	0.959732	0.968007	0.939314	0.953445	0.993661
DecisionTreeClassifier (10, 5)	0.944333	0.952632	0.918881	0.935452	0.987494
LogisticRegression (1000, 1.0)	0.870342	0.865793	0.833903	0.849549	0.925254
GaussianNB (1e-09,)	0.856791	0.844560	0.825748	0.835048	0.917870
SGDClassifier (1000, 0.01)	0.824800	0.915979	0.661580	0.768267	0.900511
SVC (10.0, 'rbf')	0.449915	0.437250	0.881785	0.584610	0.469185



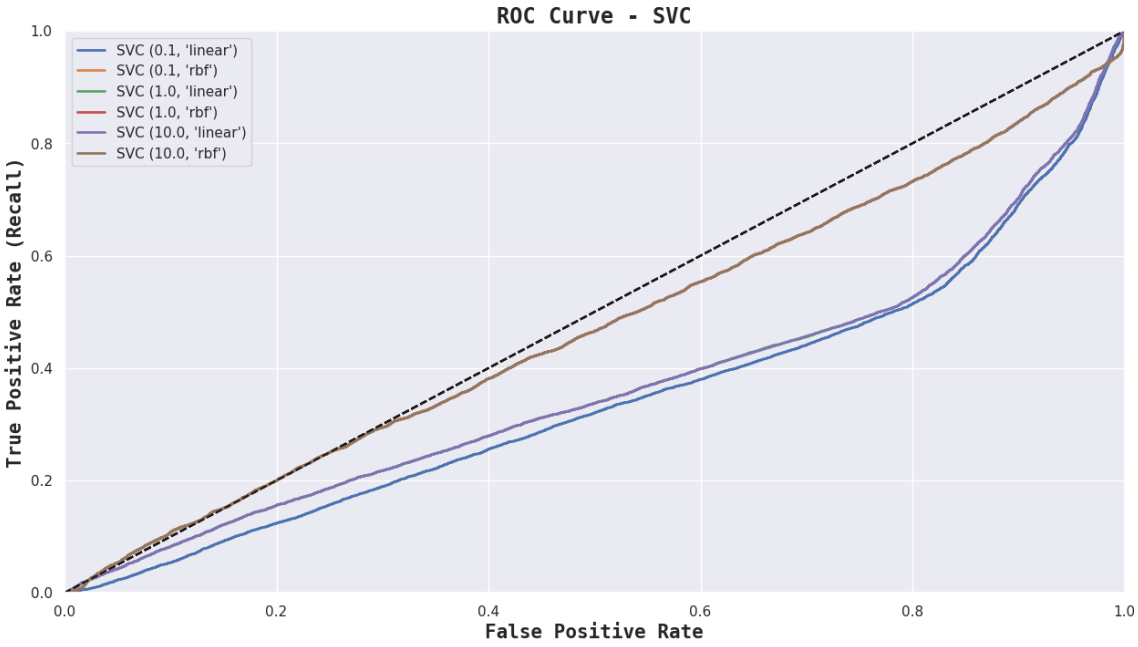
Rys. 46: Krzywa ROC dla różnych hiperparametrów modelu Logistic Regression



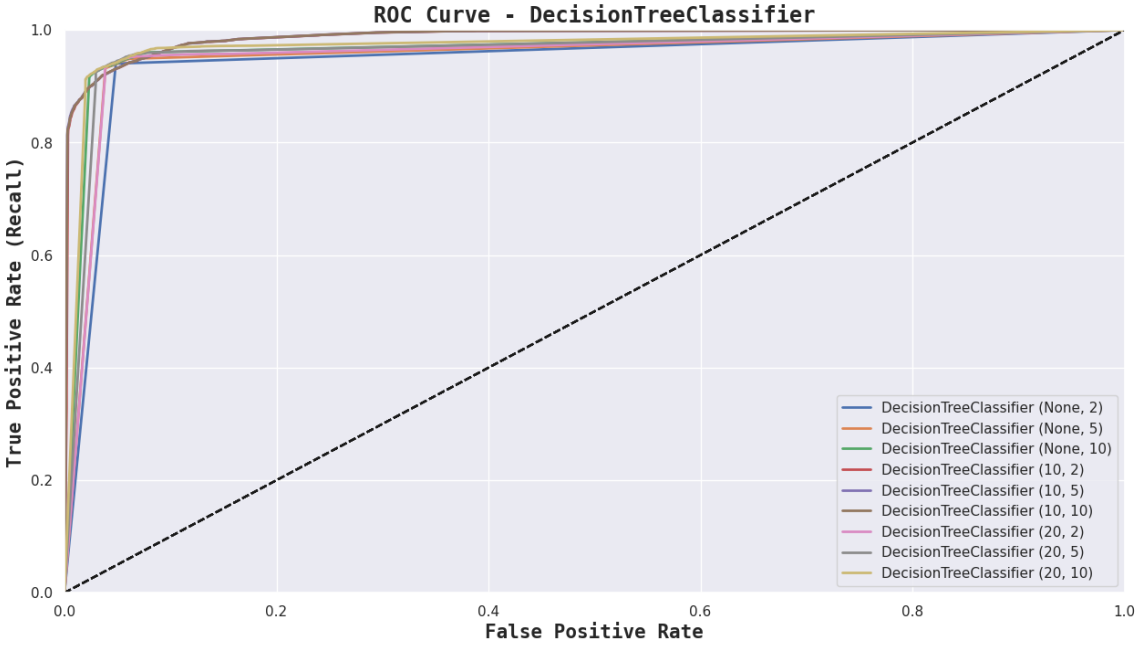
Rys. 47: Krzywa ROC dla różnych hiperparametrów modelu SGD Classifier



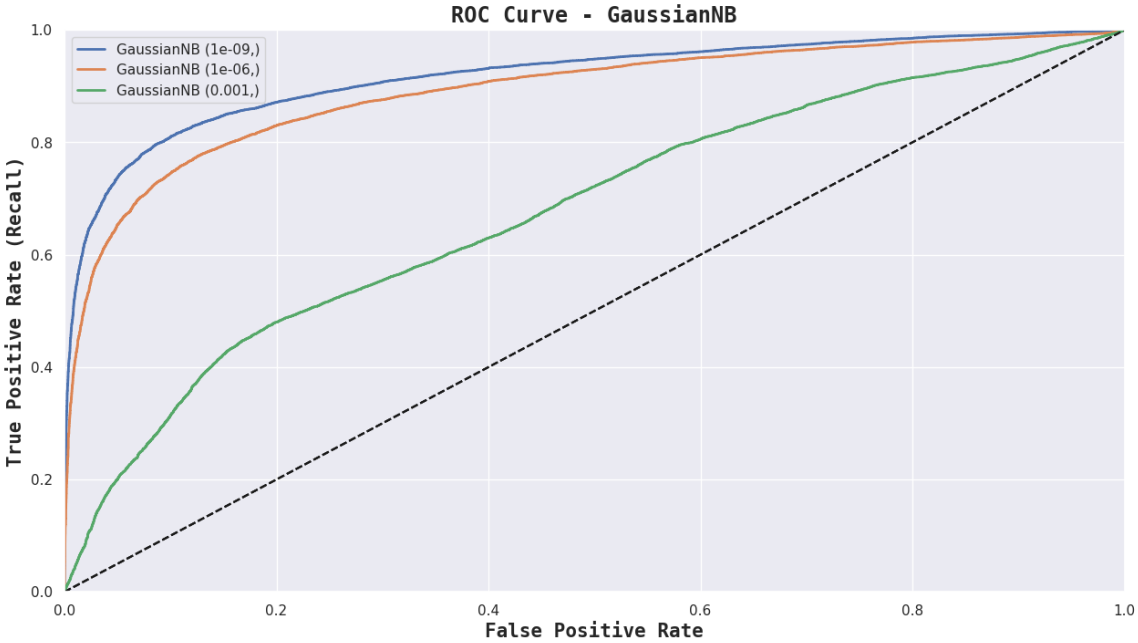
Rys. 48: Krzywa ROC dla różnych hiperparametrów modelu Random Forest Classifier



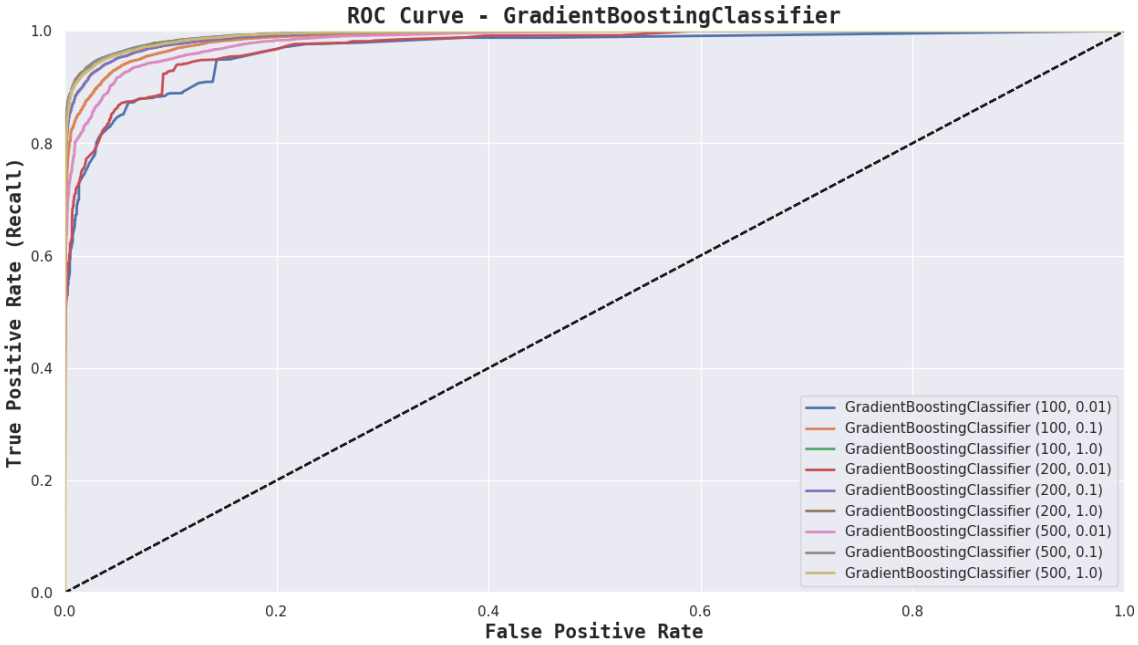
Rys. 49: Krzywa ROC dla różnych hiperparametrów modelu SVC



Rys. 50: Krzywa ROC dla różnych hiperparametrów modelu Decision Tree Classifier



Rys. 51: Krzywa ROC dla różnych hiperparametrów modelu Gaussian Naivy Bayes



Rys. 52: Krzywa ROC dla różnych hiperparametrów modelu Gradient Boosting Classifier

WNIOSKI

Próba optymalizacji modeli uczenia maszynowego przez testowanie różnych nastaw ich parametrów prowadzi do nieznacznej poprawy wyników, co prezentuje tabela 10. Jedynie w przypadku algorytmu stochastycznego spadku gradientu zanotowaliśmy spory wzrost z dokładności na poziomie 0.66 do 0.82.

8 Wnioski końcowe

Przed przystąpieniem do trenowania modeli konieczna jest dokładna analiza i eksploatacja danych. Ważne jest, aby odpowiednio przygotować dane treningowe, tzn. uzupełnić ewentualne braki (jest to bardzo istotny zabieg, gdyż niektóre z modeli nie działają poprawnie w sytuacji, gdy występują puste pola), obsłużyć zmienne kategoryczne w zależności o tego czy posiadają pewną hierarchię wewnętrzną, czy też nie, obsłużyć wartości odstające i usunąć niepotrzebne kolumny i duplikaty, co może wpływać pozytywnie na poprawę dokładności modeli i ich szybkość działania.

Bardzo istotnym aspektem podczas podziału danych jest zwrócenie uwagi na tzn. wyciek danych. Wyciek danych w uczeniu maszynowym to sytuacja, gdy do danych treningowych przedostają się informacje zawarte w danych testowych. Może to zachodzić, np. gdy:

- przy imputacji wyznaczy się średnią, lub inną statystykę na podstawie całych danych, a nie tylko treningowych

- przy cechach bazujących na agregacji w czasie, brane są pod uwagę informację z całych danych, czyli również też te z przyszłości

Istotnym objawem wycieku danych jest na przykład odstająco duży wpływ zmiennej na predykcję.Co może powodować potem złe efekty.

Podczas doboru parametrów w modelach dobrą praktyką jest wykorzystanie zbioru walidacyjnego, który służy sprawdzeniu jakości modelu w trakcie uczenia.

Najlepszym rozwiązaniem w przypadku rozważanego przez nas problemu jest zastosowanie modelu opartego na lasie losowym lub Gradient Boosting, najgorszym zaś wyborem będzie Maszyna Wektorów Nośnych (daje najgorsze wyniki i czas dokonywania predykcji jest najdłuższy)

Poniżej przedstawione zostały ogólne wnioski i spostrzeżenia dotyczące poszczególnych metod uczenia maszynowego

1. Maszyna wektorów nośnych (SVM):
    - SVM jest skutecznym modelem do klasyfikacji i regresji, szczególnie w przypadku danych liniowo separowalnych.
    - Ważne jest odpowiednie dobranie hiperparametrów, takich jak kernel i regularyzacja, aby osiągnąć optymalne wyniki.
    - SVM może być wrażliwy na duże zbiory danych ze względu na złożoność obliczeniową. Proces treningu trwa najdłuższej ze wszystkich analizowanych modeli.
2. Drzewo decyzyjne:
    - Drzewa decyzyjne są łatwe do zrozumienia i interpretacji, co czyni je przydatnym narzędziem do analizy danych.
    - Mogą być podatne na przetrenowanie, zwłaszcza gdy drzewo jest zbyt głębokie i dopasowuje się do szumów w danych
    - Ważne jest kontrolowanie parametrów drzewa, takich jak maksymalna głębokość i minimalna liczba próbek w liściu, aby uniknąć przetrenowania.
3. Las losowy:
    - Las losowy jest zespołowym modelem opartym na drzewach decyzyjnych, który ma większą zdolność do generalizacji niż pojedyncze drzewo.
    - Ważne jest ustawienie odpowiednich hiperparametrów, takich jak liczba drzew w lesie i maksymalna głębokość drzewa, aby uzyskać optymalne wyniki.
    - Las losowy jest mniej podatny na przetrenowanie niż pojedyncze drzewo decyzyjne.
4. Gradient Boost:
    - Gradient Boosting jest techniką zespołową, która łączy wiele słabych modeli w silny model.

- Może dostarczyć bardzo dobre wyniki predykcyjne, ale jest bardziej skomplikowany do strojenia i wymaga odpowiedniego dobrania parametrów, takich jak współczynnik uczenia i liczba iteracji.

5. Klasyfikator bayesowski:
   - Klasyfikator bayesowski opiera się na teoretycznym podstawie statystyki bayesowskiej.
   - Jest prosty w implementacji i wydajny obliczeniowo.
   - Jego wydajność może być zależna od założeń dotyczących niezależności zmiennych oraz od rozkładu danych.
6. Regresja logistyczna:
   - Regresja logistyczna jest modelem wykorzystywanym głównie do klasyfikacji binarnej.
   - Może być stosowana również w problemach wieloklasowych poprzez techniki takie jak one-vs-all lub softmax.
   - Ważne jest uwzględnienie regularyzacji, aby uniknąć przetrenowania modelu.
7. Algorytm stochastycznego spadku gradientu:
   - Algorytm stochastycznego spadku gradientu jest wykorzystywany do optymalizacji modeli uczenia maszynowego.
   - Jest wydajny obliczeniowo i dobrze skaluje się do dużych zbiorów danych.
   - Ważne jest odpowiednie dostosowanie wielkości kroku (learning rate) i innych parametrów algorytmu, aby zapewnić jego zbieżność i uniknąć utknięcia w minimach lokalnych.

9 Bibliografia

1. Jupyter Notebook: https://docs.jupyter.org/en/latest/
2. Google Colaboratory: https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook Help-And-Documentation.ipynb
3. Pycharm: https://www.jetbrains.com/pycharm/guide/tips/quick-docs/
4. Pandas: https://pandas.pydata.org/docs/user_guide/index.html
5. Numpy: https://numpy.org/devdocs/user/
6. Scikit-learn: https://scikit-learn.org/stable/
7. Matplotlib: https://matplotlib.org/stable/index.html
8. Seaborn: https://seaborn.pydata.org/
9. XGBoost: https://xgboost.readthedocs.io/en/stable/
10. Python: https://www.python.org/doc/
11. SVM: https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/
12. Drzewo Decyzyjne: https://en.wikipedia.org/wiki/Decision_tree
13. Las Losowy: https://www.ibm.com/topics/random-forest
14. Gradient Boosting: https://en.wikipedia.org/wiki/Gradient_boosting
15. Naiwny klasyfikator bayesowski: https://en.wikipedia.org/wiki/Naive_Bayes_classifier
16. Regresja logistyczna: https://en.wikipedia.org/wiki/Logistic$_r egression$
17. Algorytm stochastycznego spadku gradientu: https://en.wikipedia.org/wiki/Stochastic$_g radient_d escent$
18. Zbiór danych: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction
19. Metryki klasyfikacji: https://ksopyla.com/data-science/precision-recall-f1-miary-oceny-klasyfikatora/