**Góry czy plaże? Klasyfikacja preferencji podróżniczych**

BIG DATA

Dobrosielski Jakub 240651

Fokt Agnieszka 240659

Spis treści

[Cel projektu 3](#_Toc185182805)

[Tezy dla procesu przetwarzania danych 3](#_Toc185182806)

[Założenia dotyczące metod 3](#_Toc185182807)

[Dobór danych do postawionego celu 4](#_Toc185182808)

[Natywna struktura i formaty danych 4](#_Toc185182809)

[Charakter ilościowy i jakościowy danych surowych 5](#_Toc185182810)

[Sposób powstania danych 5](#_Toc185182811)

[Dynamika zmian danych 5](#_Toc185182812)

[Źródło pochodzenia 5](#_Toc185182813)

[Eksploracyjna Analiza Danych (EDA) 6](#_Toc185182814)

[Analiza korelacji 6](#_Toc185182815)

[Wizualizacja Preferencji 7](#_Toc185182816)

[Skrypty do ekstrakcji, transformacji i ładowania danych (ETL) 11](#_Toc185182817)

[Ekstrakcja danych 11](#_Toc185182818)

[Transformacja danych 11](#_Toc185182819)

[Przygotowanie Pipeline 12](#_Toc185182820)

[Analiza i ocena modeli klasyfikacyjnych 13](#_Toc185182821)

[Las Losowy (Random Forest) 13](#_Toc185182822)

[Drzewo decyzyjne (Decision Tree) 16](#_Toc185182823)

[Wnioski z analizy 19](#_Toc185182824)

[Dokładność modeli 19](#_Toc185182825)

[Zastosowanie modeli w procesie decyzyjnym 19](#_Toc185182826)

[Potencjalne usprawnienia 19](#_Toc185182827)

# Cel projektu

Celem projektu jest analiza preferencji ludzi dotyczących spędzania czasu w górach lub na plaży na podstawie wybranych cech demograficznych i behawioralnych. Na podstawie danych chcemy zbudować model klasyfikacyjny, który pozwoli przewidzieć preferencje użytkownika (góry vs plaża) na podstawie cech takich jak wiek, płeć, dochody, poziom wykształcenia, budżet na wakacje czy częstotliwość podróżowania.

## Tezy dla procesu przetwarzania danych

1. **Istnieje związek między preferencjami lokalizacji wakacyjnej (góry vs plaża) a cechami demograficznymi, takimi jak wiek, dochody czy poziom wykształcenia.**  
   Zakładamy, że osoby o wyższym dochodzie mogą częściej wybierać plaże, podczas gdy osoby z większą świadomością ekologiczną mogą preferować góry.
2. **Styl życia i preferencje aktywności mają wpływ na wybór między górami a plażą.**  
   Przykładowo, osoby aktywne mogą preferować góry, podczas gdy osoby szukające relaksu wybiorą plażę.
3. **Czynniki geograficzne, takie jak bliskość do gór lub plaż, mogą mieć znaczący wpływ na preferencje.**  
   Osoby mieszkające bliżej plaży mogą mieć inne wybory niż osoby mieszkające w pobliżu gór.

## Założenia dotyczące metod

1. **Metody eksploracji danych:** Wykorzystamy eksploracyjną analizę danych (EDA), aby zrozumieć rozkład zmiennych, relacje między nimi i potencjalne czynniki wpływające na preferencje. Obejmie to wizualizacje i statystyki opisowe.
2. **Metody statystyczne:** Wstępne analizy, takie jak współczynniki korelacji, pozwolą nam zidentyfikować zmienne kluczowe dla predykcji.
3. **Metody klasyfikacji:** W celu budowy modelu zastosujemy algorytmy Random Forest i Decision Tree.
4. **Narzędzia:** Do przetwarzania i analizy danych wykorzystamy PySpark, co pozwoli na przetwarzanie dużych zbiorów danych w sposób wydajny, a do wizualizacji wyników - biblioteki takie jak Matplotlib i Seaborn.

# Dobór danych do postawionego celu

Do projektu wybraliśmy zbiór danych **"Mountains vs Beaches Preference"** dostępny na platformie Kaggle. Zbiór ten został wybrany ze względu na jego potencjał do analizy preferencji użytkowników w kontekście wyboru między górami a plażami. Dane zawierają różnorodne cechy demograficzne, behawioralne oraz związane ze stylem życia, które mogą wpływać na preferencje użytkowników. Zbiór umożliwia zastosowanie metod klasyfikacji w celu przewidywania preferencji użytkowników w zależności od ich atrybutów.

## Natywna struktura i formaty danych

Dane są zapisane w formacie CSV. Zbiór zawiera:

* **52444 rekordów**, opisujących indywidualne odpowiedzi,
* **13 cech,** które reprezentują różne właściwości użytkowników oraz ich preferencje.

Każdy wiersz odpowiada jednemu użytkownikowi, a kolumny opisują:

* **Wiek (Age)** – wartość numeryczna.
* **Płeć (Gender)** – kategorie: mężczyzna, kobieta, osoba niebinarna.
* **Dochód (Income)** – roczny dochód użytkownika (numeryczny).
* **Poziom wykształcenia (Education Level)** – kategorie: szkoła średnia, licencjat, magister, doktorat.
* **Częstotliwość podróży (Travel Frequency)** – liczba wakacji w roku (numeryczna).
* **Preferowane aktywności (Preferred Activities)** – kategorie: wędrówki, pływanie, narciarstwo, opalanie się.
* **Budżet na wakacje (Vacation Budget)** – budżet przeznaczony na wakacje (numeryczny).
* **Miejsce zamieszkania (Location)** – typ rezydencji: miasto, przedmieścia, wieś.
* **Bliskość do gór (Proximity to Mountains)** – odległość od najbliższych gór w milach (numeryczna).
* **Bliskość do plaż (Proximity to Beaches)** – odległość od najbliższych plaż w milach (numeryczna).
* **Ulubiona pora roku (Favorite Season)** – kategorie: lato, zima, wiosna, jesień.
* **Posiadanie zwierząt (Pets)** – czy użytkownik posiada zwierzęta: 0 = Nie, 1 = Tak.
* **Troska o środowisko (Environmental Concerns)** – czy użytkownik jest zaangażowany w kwestie ekologiczne: 0 = Nie, 1 = Tak.

## Charakter ilościowy i jakościowy danych surowych

1. **Charakter ilościowy**:
   * **Liczba rekordów**: 52444 odpowiedzi użytkowników.
   * **Liczba cech**: 13 atrybutów opisujących różnorodne aspekty demograficzne, ekonomiczne i preferencyjne.
2. **Charakter jakościowy**:
   * Zbiór danych zawiera zarówno zmienne **kategoryczne** (np. płeć, preferowane aktywności), jak i **numeryczne** (np. wiek, dochód).

## Sposób powstania danych

Zbiór danych pochodzi z ankiety online, gdzie użytkownicy wypełniali informacje dotyczące swoich cech demograficznych, stylu życia oraz preferencji wakacyjnych. Jest to przykład **danych wtórnych**, ponieważ zostały one wcześniej zebrane, przetworzone i udostępnione na platformie Kaggle przez autora Jahnavi Paliwal.

## Dynamika zmian danych

Dane zostały zebrane w ramach **rejestracji jednorazowej**, co oznacza, że dotyczą określonego punktu w czasie. Zbiór nie jest dynamiczny ani aktualizowany, a wszystkie wartości odnoszą się do momentu ich rejestracji przez autora.

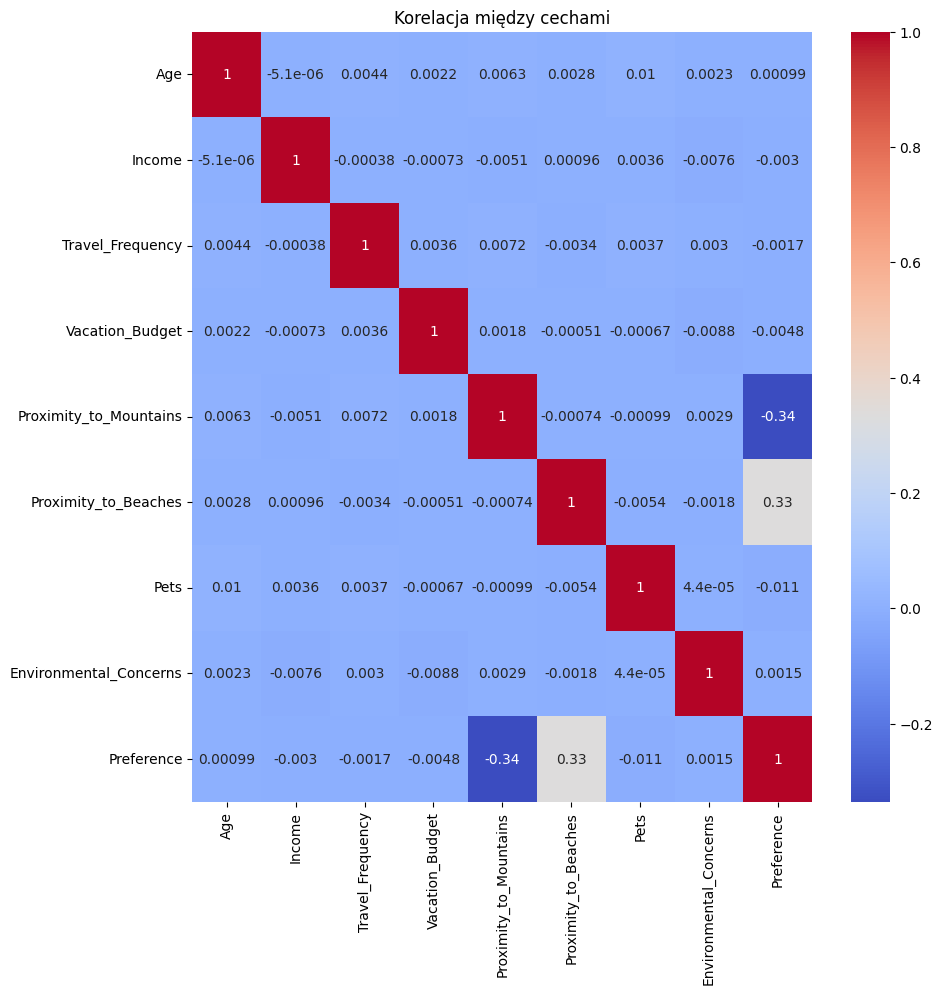
## Źródło pochodzenia

Dane pochodzą z platformy Kaggle i są dostępne publicznie do celów edukacyjnych, analitycznych oraz do trenowania modeli uczenia maszynowego. Autorem zbioru jest Jahnavi Paliwal. Link do danych: <https://www.kaggle.com/datasets/jahnavipaliwal/mountains-vs-beaches-preference/data>

# Eksploracyjna Analiza Danych (EDA)

EDA została przeprowadzona przy użyciu **Apache Spark** do wstępnego przetwarzania danych i generowania statystyk oraz bibliotek takich jak **Pandas**, **Seaborn** i **Matplotlib** do wizualizacji. Oto główne elementy analizy:

## Analiza korelacji

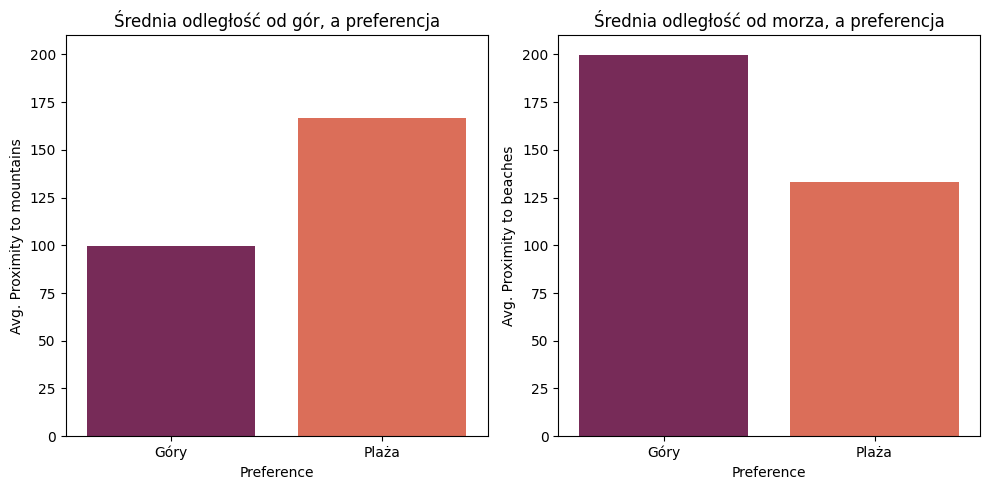


Rysunek 1- wykres korelacji

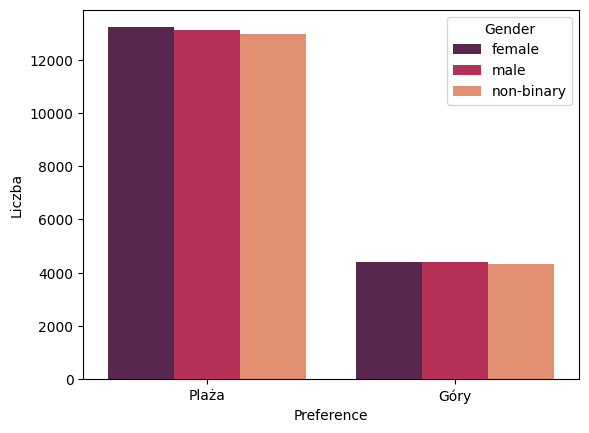
**Kluczowe spostrzeżenia**:

1. **Proximity\_to\_Mountains** a **Preference** – wartość korelacji wynosi **-0.34**. Wskazuje to na **umiarkowaną ujemną korelację**: osoby bliżej gór mają mniejszą preferencję na wybór plaży.
2. **Proximity\_to\_Beaches** a **Preference** – wartość korelacji to **0.33**, czyli **umiarkowana dodatnia korelacja**. Osoby bliżej plaż chętniej wybierają plażę.
3. Pozostałe cechy, nie wykazują znaczącej korelacji między sobą.

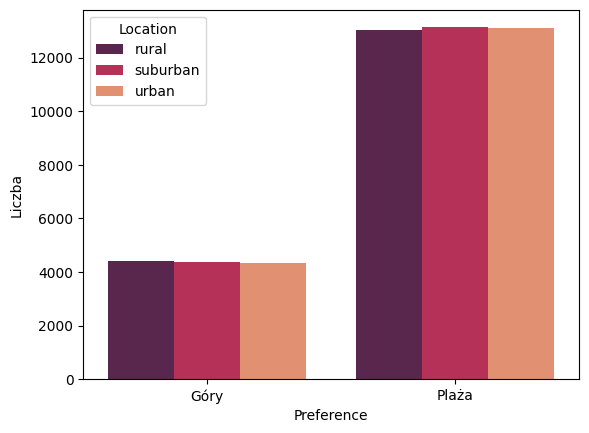
## Wizualizacja Preferencji



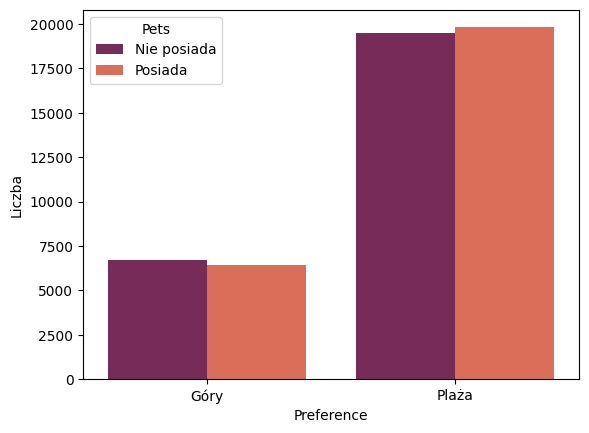
Rysunek 2-wykres odległość od gór/morza



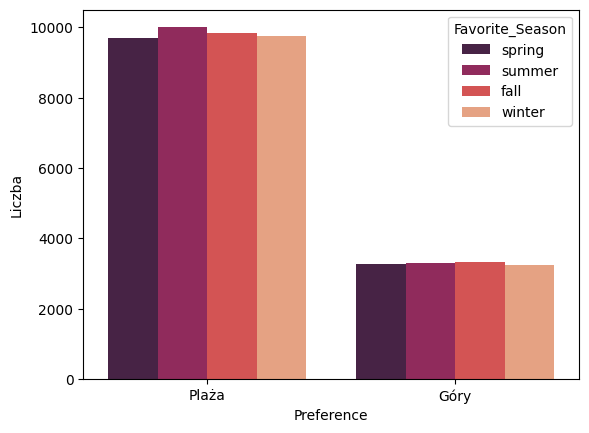
Rysunek 3- wykres płci



Rysunek 4- wykres miejsca zamieszkania



Rysunek 5- wykres posiadania zwierzęcia



Rysunek 6- wykres ulubionej pory roku

**Kluczowe spostrzeżenia**:

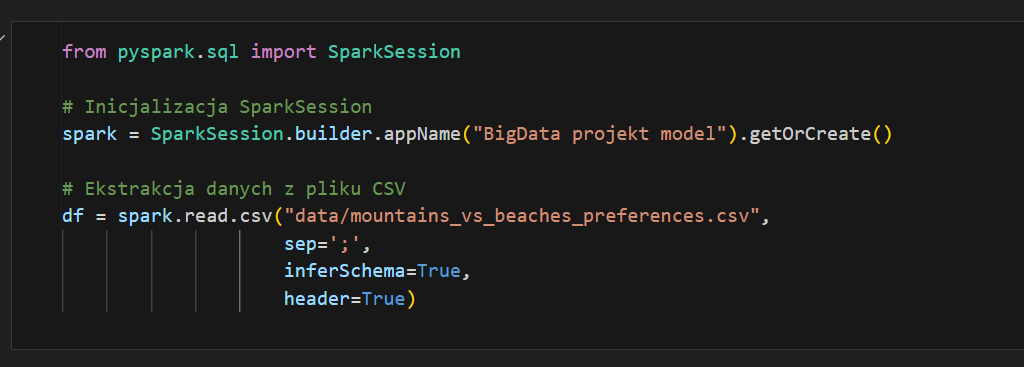
1. Na rysunku 2 z dwóch wykresów możemy zauważyć ze osoby preferujące góry mieszkają bliżej gór, a ci mieszkający bliżej plaży wolą plażę
2. Na pozostałych rysunkach, tj. 3, 4, 5, 6 możemy zauważyć równomierne rozłożenie się danych cech oraz to, że nie mają one bezośrednio wpływu na wybór gór czy plaż

# Skrypty do ekstrakcji, transformacji i ładowania danych (ETL)

Skrypty zostały stworzone przy użyciu Apache Spark i PySpark, umożliwiając procesowanie dużych zbiorów danych w sposób skalowalny. Kluczowe etapy obejmowały:

## Ekstrakcja danych

Dane wejściowe zostały wczytane z pliku CSV za pomocą funkcji **spark.read.csv**.

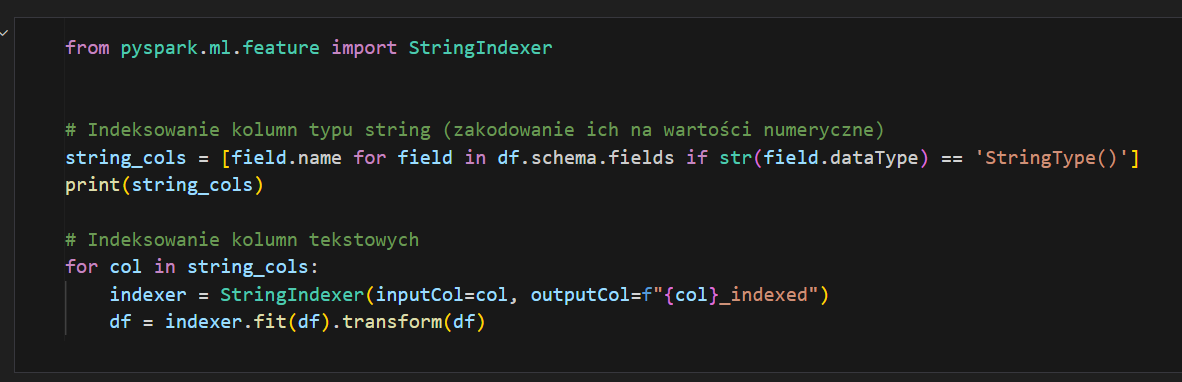


Rysunek 7- kod wczytywania danych z CSV

## Transformacja danych

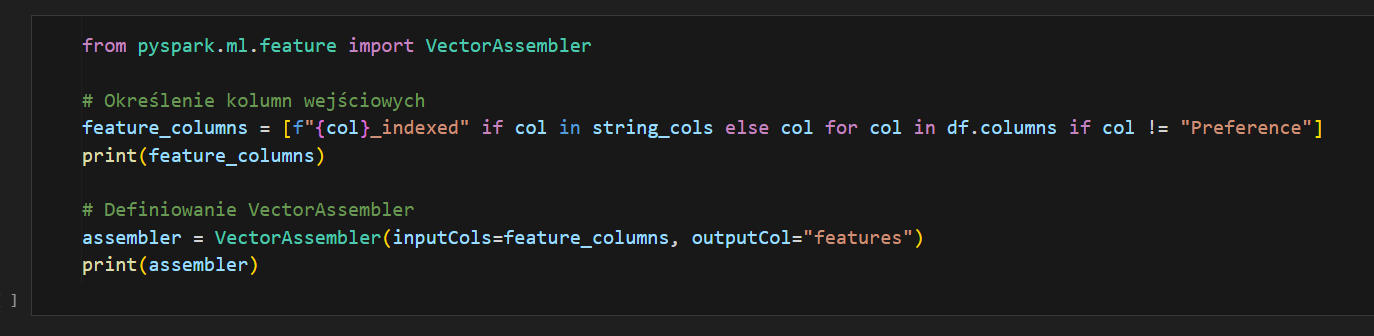
Dane zostały przekształcone i przygotowane do dalszej analizy:

* 1. **Indeksowanie kolumn tekstowych**  
     Wartości tekstowe w kolumnach zostały zakodowane na wartości numeryczne przy użyciu **StringIndexer**.



Rysunek 8- kod indeksowania kolumn tekstowych

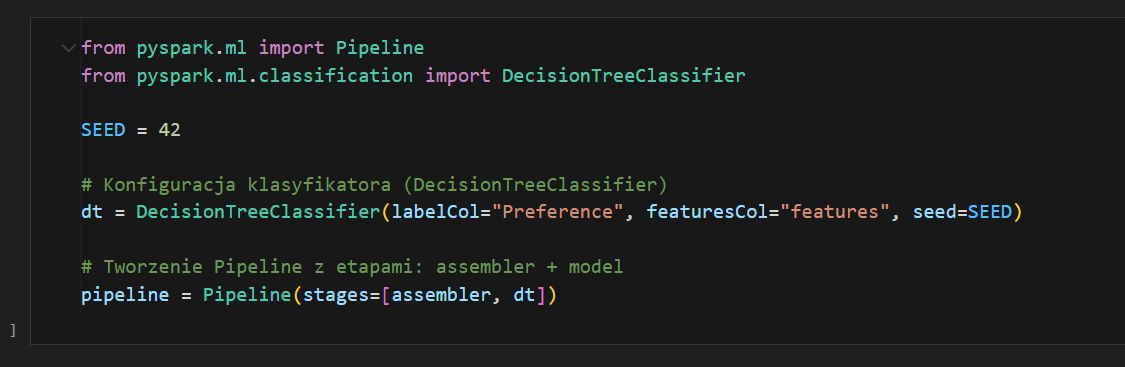
* 1. **Tworzenie wektora cech**  
     Wybrane kolumny zostały określone jako zbiór cech wejściowych. Za pomocą **VectorAssembler** stworzono definicję wektora cech.



Rysunek 9- kod tworzenia wektora cech

## Przygotowanie Pipeline

Utworzono Pipeline, który łączy etapy przekształcania danych (indeksowanie, łączenie cech) oraz konfigurację modelu klasyfikatora.



Rysunek 10 - kod pipeline

# Analiza i ocena modeli klasyfikacyjnych

## Las Losowy (Random Forest)

Las Losowy to metoda uczenia maszynowego, która łączy wyniki wielu drzew decyzyjnych w celu poprawy ogólnej wydajności modelu oraz zminimalizowania ryzyka przeuczenia (overfitting).

1. **Domyślne parametry** 
   * + numTrees: 20

Liczba drzew w lesie losowym.

* + - maxDepth: 5

Maksymalna głębokość drzewa decyzyjnego.

* + - featureSubsetStrategy: "auto"

Strategia wyboru podzbioru cech (domyślnie √(liczba cech)).

* + - impurity: "gini"

Miara zanieczyszczenia, używana do podziału węzłów (indeks Giniego).

* + - subsamplingRate: 1.0

Procent próbek użytych do treningu dla każdego drzewa.

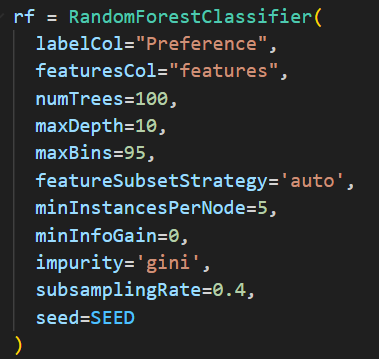
* + - minInstancesPerNode: 1

Minimalna liczba instancji w węźle do jego podziału.

* + - minInfoGain: 0.0

Minimalny przyrost informacji wymagany do podziału.

1. **Parametry ustawione przez nas**



Rysunek 11- kod parametrów RandomForestClassifier

Wyjaśnienie zmodyfikowanych parametrów:

* numTrees=100

Zwiększona liczba drzew poprawia stabilność i dokładność modelu kosztem wydłużenia czasu treningu.

* maxDepth=10

Ustawiona maksymalna głębokość drzewa pozwala na bardziej szczegółowe podziały, ale jednocześnie kontroluje złożoność, aby uniknąć przeuczenia.

* maxBins=95

Większa liczba binów pozwala na dokładniejsze podziały przy kategoryzacji cech numerycznych.

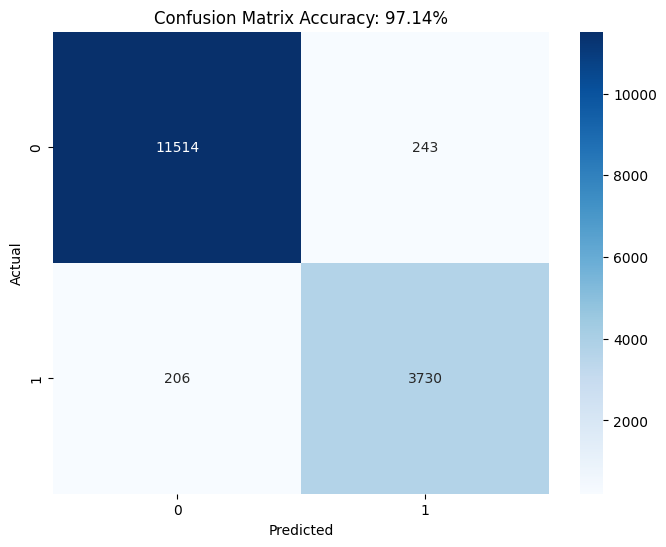
* subsamplingRate=0.4

Tylko 40% danych treningowych jest używanych przy tworzeniu każdego drzewa, co poprawia szybkość obliczeń i redukuje korelację między drzewami.

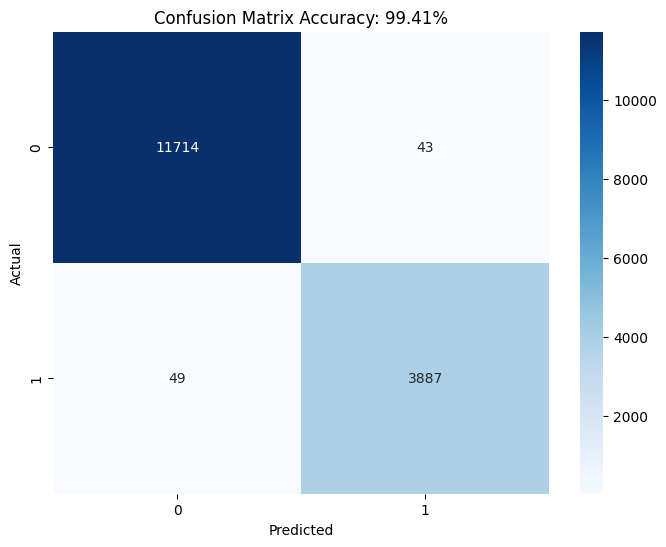
* minInstancesPerNode=5

Minimalna liczba instancji w węźle wynosi 5, co zapobiega zbyt drobnym podziałom i pomaga uniknąć nadmiernego dopasowania.

1. **Wyniki**



Rysunek 12 - macierz pomyłek domyślny RandomForest



Rysunek 13 - macierz pomyłek zmienione parametry RandomForest

1. **Wnioski**

Porównując wyniki domyślnego klasyfikatora (Random Forest) z ustawionymi parametrami, zauważamy znaczną poprawę dokładności klasyfikacji. Domyślny model uzyskał dokładność na poziomie 97.14%, podczas gdy po dostosowaniu parametrów (np. liczba drzew, głębokość drzew, strategie próbkowania) dokładność wzrosła do 99.41%.

## Drzewo decyzyjne (Decision Tree)

Drzewo decyzyjne to model uczenia maszynowego, który podejmuje decyzje na podstawie zestawu reguł warunkowych, przedstawionych w formie drzewa. Każdy węzeł drzewa reprezentuje pytanie dotyczące jednej cechy, a gałęzie odpowiadają na pytania, prowadząc do liści, które zawierają prognozy.

* 1. **Domyślne parametry**
     + - * maxDepth: 5

Maksymalna głębokość drzewa.

* + - * + maxBins: 32

Liczba przedziałów do podziału cech numerycznych.

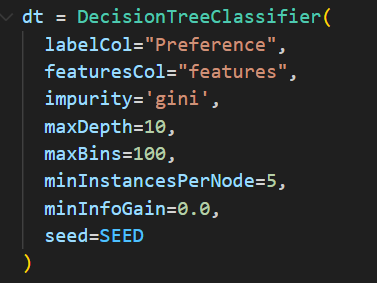
* + - * + minInstancesPerNode: 1

Minimalna liczba próbek w węźle.

* + - * + minInfoGain: 0.0

Minimalny wzrost informacji wymagany do podziału węzła.

* 1. **Parametry ustawione przez nas**

****

Rysunek 14 - kod parametrów DecisionTreeClassifier

Wyjaśnienie zmodyfikowanych parametrów:

* + - * maxDepth=10

Głębsze drzewo pozwala na lepsze modelowanie bardziej skomplikowanych danych.

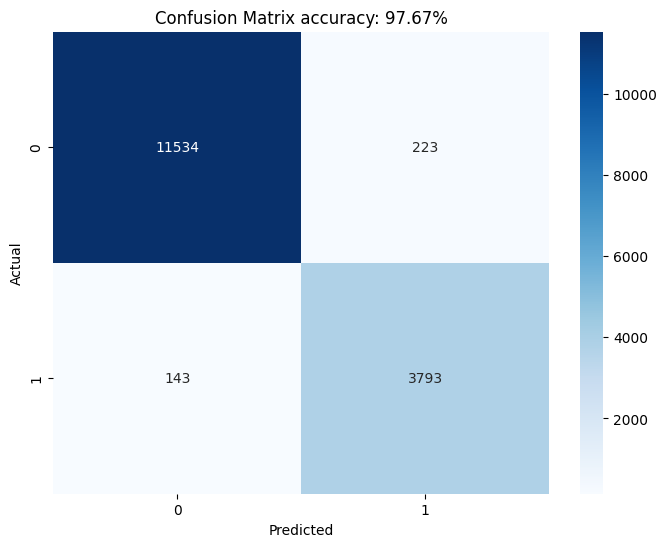
* + - * maxBins=100

Więcej przedziałów dla lepszego rozdzielenia cech numerycznych.

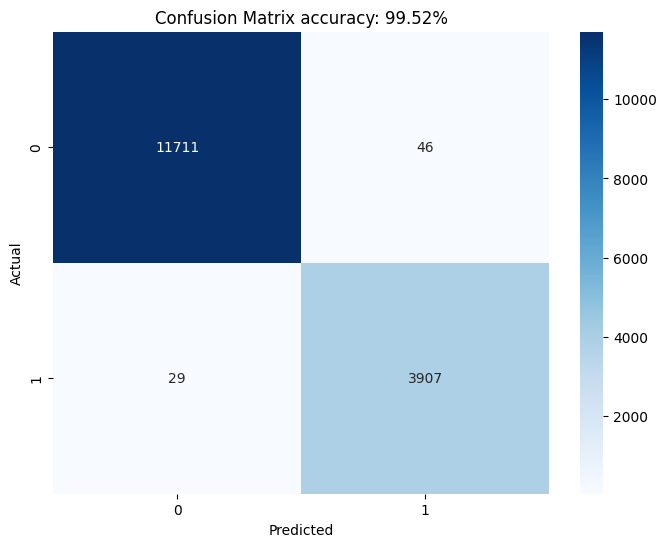
* + - * minInstancesPerNode=5

Zapewnia, że w każdym węźle znajduje się co najmniej 5 przykładów, co może zapobiec przeuczeniu.

* 1. **Wyniki**

****

Rysunek 15 - macierz pomyłek domyślny DecisionTree



Rysunek 16 - macierz pomyłek zmienione parametry DecisionTree

* 1. **Wnioski**

Model z ustawieniami domyślnymi osiągnął dokładność na poziomie 97.67%. Po modyfikacji parametrów uzyskano podobny efekt jak w przypadku klasyfikatora Lasu Losowego, co przełożyło się na wzrost dokładności do 99.52%.

# Wnioski z analizy

## Dokładność modeli

Zarówno dla klasyfikatora Random Forest, jak i dla klasyfikatora Drzewa Decyzyjnego, modyfikacja parametrów znacząco poprawiła dokładność modelu. W przypadku domyślnych ustawień, oba modele uzyskały wysoką dokładność (97.14% dla Lasu Losowego i 97.67% dla Drzewa Decyzyjnego). Jednak po dostosowaniu parametrów, modele osiągnęły jeszcze lepsze wyniki, uzyskując 99.41% oraz 99.52% odpowiednio. Te zmiany wskazują na znaczenie precyzyjnego dostosowania parametrów modelu, aby uzyskać jak najlepsze wyniki predykcyjne.

## Zastosowanie modeli w procesie decyzyjnym

Modele predykcyjne, takie jak Random Forest i Drzewo Decyzyjne, mogą stanowić kluczowe narzędzie w procesach decyzyjnych w różnych dziedzinach. Ich zastosowanie w przewidywaniu preferencji, zachowań klientów, czy wyników działalności gospodarczej pozwala na podejmowanie bardziej świadomych decyzji, z wykorzystaniem danych, które mogą wskazywać na ukryte zależności. Modele o wysokiej dokładności, takie jak te, które zostały opracowane, mogą pomóc w optymalizacji działań i planowaniu strategicznym w takich obszarach jak marketing, rozwój produktów czy analizy ryzyka.

## Potencjalne usprawnienia

Choć wyniki są już bardzo dobre, dalsze eksperymentowanie z różnymi parametrami, takimi jak liczba drzew w lesie losowym, głębokość drzewa decyzyjnego, czy inne techniki przetwarzania danych (np. normalizacja cech), może jeszcze poprawić dokładność predykcji. Z kolei zmiana algorytmu na inny, mogłaby również przynieść lepsze rezultaty w niektórych przypadkach.