UNIWERSYTET ŚLĄSKI
W KATOWICACH

Wydział Nauk Ścisłych i Technicznych

Dariusz Litwiński

349142

Analiza wpływu zastosowania wybranych technik przygotowania danych do analizy, na jakość analizy danych

Praca Magisterska

dr Kornel Chromiński

Sosnowiec. 2023

Spis treści

Wstęp

Wprowadzenie do problemu

Analiza danych to process, w którym przekształcamy surowe dane w wiedzę i wnioski, dzięki którym jesteśmy w stanie podejmować lepsze decyzje [4]. Im dokładniejsza analiza, tym trafniejsze decyzje będziemy w stanie podjąć na jej podstawie, dlatego powinno się ten proces wielokrotnie powtarzać, za każdym razem próbując uzyskać lepsze rezultaty. Skupiając się na przygotowaniu danych można uzyskać znaczącą poprawę wyników, ponieważ dane bardzo często zawierają wartości brakujące, nieodpowiednio zakodowane, bądź można uzyskać dodatkowe informacje poprzed odopowiednie ich spreparowanie. Decyzja jakie środki przygotowania danych zastosować bywa trudna i wymaga czasu oraz zbadania zbioru danych, a także bardzo często nie daje aż tak wymiernych rezultatów jakich się spodziewamy.

Cel pracy

Celem pracy jest sprawdzenie, jak poszczególne metody przygotowania danych wpływają na jakość analizy danych, konkretnie modelowania klasyfikatorów. Przeprowadzono eksperymenty z użyciem wielu metod przygotowania danych, a następnie porównano trafność klasyfikacji względem klasyfikatora bez przygotowania danych. Za brak przygotowania danych uznaje się usunięcie wszystkich rekordów z brakującymi wartościami oraz wszystkich kolumn nieliczbowych. Rozpoczynając prace postawiono hipotezę, że najlepszym sposobem na przygotowanie danych jest dogłębne ich zrozumienie, a następnie dostosowanie do nich użytych metod, jak i rówież fakt, że jakiekoliwek przygotowanie danych powinno wpłynąć pozytywnie na jakość analizy danych.

Zawartość pracy

W pracy krótko przedstawiono problem analizy danych, wyliczono typy metod przygotowania danych, opisano wykonane eksperymenty, zarówno środowisko w jakim je przeprowadzono, jak i to na czym polegały. Ostatnią część pracy stanowią wyniki eksperymentów wraz z ich podsumowaniem

Rozdział 1

Problem Analizy Danych

W poniższym rozdziale wymieniono oraz opisano fazy procesu analizy danych, wprowadzono pojęcie jakości danych oraz omówiono problemy jakie mogą wystąpić przy pracy z danymi

Pozyskiwanie: gromadzenie danych

Zanim analiza danych będzie możliwa należy pozyskać dane, jest to istotna cześć procesu, ponieważ to od niej zależy to, jak trafne wnioski będziemy mogli wysnuć na późniejszych etapach. Bardzo ważne jest to, jak dużą ilość danych uda nam się zebrać, a także jak dokładne one będą. Możemy pozyskiwać dane z różnorakich źródeł, na przykład:

- Zapisywać wartości zbierane przez czujniki [5]
- Korzystać z ankiet zebranych wśród danej populacji [7]
- Zbierać dane o zachowaniach użytkowników w trakcie korzystania ze strony internetowej [8]
- Wykorzystywać dane statystyczne/historyczne [6]

Jesteśmy w tej kwestii ograniczeni jedynie przez dziedzinę, w jakiej przeprowadzamy analizę danych Istotną kwestią jest również to, w jakim formacie są przechowywane dane. Może okazać się, że pierwszym etapem przetwarzania danych będzie odpowiednie ich sformatowanie, bądź nawet ich cyfryzacja, jeśli były by przechowywane w formie fizycznej, na przykład jako papierowe archiwa

Przygotowanie: przetwarzanie danych

Aby móc w pełni korzystać ze zgromadzonych danych, należy je przygotować, odpowiednio sformatować. Poza przygotowaniem odpowiedniego formatu danych, możemy wyróżnić następujące sposoby na przygotowanie danych do analizy:

- Wypełnienie brakujących wartości
- Standaryzacja danych liczbowych
- Kodowanie wartości kategorycznych jako liczbowe

Dzięki wykorzystaniu powyższych sposobów możemy znacznie zwiększyć jakość analizy danych, a co za tym idzie wnioski do których dojdziemy w trakcie procesu będą trafniejsze i przyniosą lepsze rezultaty

Analiza: modelowanie danych

Mając do dyspozycji przygotowane dane możemy przejść do ich właściwej analizy, na podstawie której tworzymy modele, klasyfikatory [15], systemy rekomendacji [16], dokonujemy klasteryzacji [17]. Najczęściej nie tworzymy ich od zera, a wspomagamy się dostępnymi bibliotekami, które zawierają najpopularniejsze algorytmy. Zdarza się, że nie jesteśmy zadowoleni z wyników, jakie przynosi wykorzystanie stworzonych modeli, bądź chcielibyśmy dokonać dalszej optymalizacji czasowej, w takim przypadku możemy rozważyć dodatkowe przygotowanie danych, aby uzyskać pożądany efekt. Po dobrze przeprowadzonej analizie, jesteśmy w stanie wykorzystać stworzone na tym etapie narzędzia do rozwiązywania rzeczywistych problemów.

Działanie: podejmowanie decyzji

Mając gotowe narzędzia będące wynikiem modelowania danych, możemy je wykorzystać aby podjąć konkretne decyzje w prawdziwym świecie: Wykorzystać stworzony klasyfikator w diagnostyce chorób, wdrożyć system rekomendacji na naszej stronie internetowej, wykorzystać stworzone klastry danych do kategoryzacji klientów. Każdy z przedstawionych problemów wymaga eksperckiej wiedzy oraz ogromnego doświadczenia, a dzięki analizie danych możemy znacznie usprawnić ich rozwiązywanie.

1.1 Jakość Danych

Jakość danych jest oceniana na podstawie wielu kryteriów, zależnych od źródła informacji[24]:

- Kompletność - ilość danych która jest kompletna bądź zdatna do użycia. Jeśli duża część danych jest niekompletna, może to prowadzić do stronniczej bądź nawet omylnej analizy.
- Unikalność - jaka część z zestawu danych się powtarza, dla przykładu zestaw danych zawierających dane o klientach powinien każdemu z nich przypisać unikalny numer identyfikacyjny
- Ważność - To kryterium mówi o tym czy zebrane dane są w odpowiednim formacie

- Aktualność - W zależności od dziedziny którą się zajmujemy, dane mogą tracić na aktualności wraz z upływem czasu, przez na przykład postęp w danej dziedzinie bądź zmieniające się warunki
- Dokładność - poprawność danych w oparciu o ustalone "źródło prawdy". Jako, że może istnieć wiele źródeł tych samych danych, należy ustalić nadrzędne źródło danych, pozostałe zaś mogą potwierdzać dokładność tego pierwszego.
- Stałość - kryterium służące do porównywania danych z dwóch zestawów danych. Używanie różnych źródeł do szukania stałych trendów w danych. Dzięki temu wnioski pojawiające się w trakcie analizy mogą być traktowane z większym zaufaniem, jako że tworzone są na podstawie wielu niezależnych zestawów danych.
- Dopasowanie do celu - to kryterium pozwala nam zapewnić fakt, że dane które są zbierane posłużą nam do rozwiązania problemu, jaki przed nami stoi

1.2 Problemy z Danymi

Bardzo często zebrane dane nie nadają się bezpośrednio do pracy z nimi. Należy najpierw wykonać szereg operacji aby pozbyć się następujących problemów:

1.2.1 Brakujące wartości

W danych mogą występować brakujące wartości, na przykład czujniki mogą różnić się między sobą ilością pobieranych parametrów, ankietowani mogą pozostawić niektóre pytania bez odpowiedzi. Brakujące wartości stanowią poważny problem, ponieważ model nie potrafi ich jednoznacznie zinterpretować, dlatego w trakcie przygotowania danych musimy podjąć decyzję, czy usunąć rekordy z brakującymi wartościami, przez co możemy znacznie zmniejszyć liczebność zbioru danych. Alternatywnym podejściem jest wypełnienie brakujących wartości. W miejsce braku może być wstawiona średnia, minimum, maksimum, lub też inna arbitralnie wybrana wartość. Pozornie wartości wstawiane w puste miejsca są kompletnie arbitralne, jednak bardzo często takie podejście skutkuje najlepszymi rezultatami, pod warunkiem że dobierzemy odpowiednią wartość do wstawiania. [9]

1.2.2 Wartości odstające

W niektórych przypadkach w danych mogą pojawić się takie wartości, które wyraźnie odstają od reszty i nie wnoszą sobą zbyt wiele informacji w kontekście analizy danych. Co więcej, mogą one zaciemniać pozostałą część danych, maskując trendy bądź prowadząc do błędnych wniosków. Dlatego najlepszym podejściem jest wykrywanie oraz usuwanie wartości, które możemy uznać za

odstające. Istnieją algorytmy pozwalające nam na odrzucenie wartości odstających. [12]

1.2.3 Kolumny kategoryczne

Wiele z modeli może pracować jedynie na wartościach liczbowych, podczas kiedy w zbiorach danych możemy znaleźć nie tylko takie wartości, ale również kategoryczne. Rezygnując z analizy tych danych tracilibyśmy wiedzę, jaką można z nich pozyskać. Nie jest to jednak konieczne, gdyż istnieją sposoby, aby zamienić te dane na postać liczbową za pomocą kodowania [14]

Rozdział 2

Metody Przygotowania Danych

Istnieje kilka najczęściej używanych metod przygotowania danych, które dzielą się na następujące grupy:

Uzupełnianie brakujących wartości

Najczęściej brakujące wartości w zbiorach danych uzupełnia się średnią lub najczęściej występującą wartością, jednak może zdarzyć się tak, że najlepszym rozwiązaniem jest uzupełnienie braków minimum, maksimum, zerem bądź inną arbitralnie wybraną wartością

Wykrywanie wartości odstających

Do wykrywania wartości odstających możemy wykorzystać manualne metody, ale również i algorytmy, które wykryją te wartości za nas. Do manualnych metod możemy zaliczyć: Wykrywanie za pomocą rozkładu normalnego, Z-score, IQR, wykrywanie za pomocą percentyli. Natomiast spośród automatycznych metod mamy do dyspozycji między innymi Las Izolacji lub Local Outlier Factor

Kodowanie wartości kategorycznych

Jeżeli znamy zależności między klasami i możemy je uporządkować, wtedy jesteśmy w stanie dokonać kodowania ręcznie, na przykład najmniejszą wartość dla edukacji podstawowej, a najwyższą dla edukacji wyższej. Natomiast jeżeli nie znamy tych zależności, możemy wykorzystać LabelEncoder, jednak on ponumeruje klasy w kolejności alfabetycznej, co nie zawsze jest pożądanym rezultatem. Innym podejściem jest One-Hot Encoding[11], który dla każdej z klas tworzy osobną kolumnę z wartością logiczną opisującą, czy dany rekord należy do tej klasy. Powoduje to wygenerowanie sporej ilości kolumn, jednak mamy wtedy pewność, że nie stworzymy nowych zależności między klasami

Standaryzacja

Standaryzacja jest procesem, po zakończeniu którego zmienna ma średnią wartość oczekiwaną zero oraz odchylenie standardowe równe jeden, dzięki czemu zyskujemy większą przejrzystość w jej analizie. Bardziej wyraźne są skupienia wokół konkretnych wartości, jednak należy zadbać o to, aby przed standaryzacją pozbyć się wartości odstających, gdyż będą one miały negatywny wpływ na zmienną po standaryzacji

Rozdział 3

Środowisko wykonawcze

W poniższym rozdziale przedstawione zostało środowisko wykonawcze, w jakim przeprowadzono eksperymenty, a także opisano zbiory danych, na których je przeprowadzono

Specyfikacja sprzetowa, system i środowisko

Sprzęt: Laptop wyposażony w procesor Intel Core i5-1135G7 (2.4Ghz) ze zintegrowaną grafiką oraz 16GB pamięci RAM System operacyjny: Windows 10 Education Menadżer środowisk: Anaconda Navigator Python: 3.9.12 Edytor kodu: Visual Studio Code z dodatkami do edycji plików w formacie Jupyter notebook

3.1 Zbiory danych

Zbiory na których będziemy sprawdzać wpływ przygotowania danych są zbiorami do klasyfikacji. Do eksperymentów wybrano następujące zestawy danych:

Tablica 3.1: Podstawowe informacje na temat zbiorów danych

Zbiory danych			
Nazwa zbioru	Ilość rekordów	Kolumny numeryczne	Kolumny kategoryczne
League of Legends Stats: S13[2]	485	3	7
Australian Rain Forecast[3]	16443	16	7
Titanic[1]	891	5	5

3.1.1 League of Legends stats

Zbiór zawierający statystyki postaci z gry League of Legends z dwóch wersji obecnego sezonu (13.1 oraz 13.3), składający się z następujących kolumn:

- Name - Imie bohatera
- Class - Klasa bohatera
- Role - Rola
- Tier - Atrybut decyzyjny, przypisanie bohatera do danego poziomu w zależności od tego, czy warto wybierać tą postać, czy należy jej unikać: Od najwyższego (God) do najniższego
- Score - Wynik obliczany na podstawie pozostałych parametrów, podczas przygotowania został usunięty, jako że bezpośrednio koreluje z atrybutem decyzyjnym.
- Trend - Trend atrybutu score względem poprzedniej wersji, podczas przygotowania został usunięty, jako że bezpośrednio koreluje z atrybutem decyzyjnym.
- Win % - Procent meczów, jaki dany bohater wygrywa
- Role % - Procent meczów, w jakiej dany bohater występuje w danej roli
- Pick % - Procent meczów, w których dany bohater jest wybierany
- Ban % - Procent meczów, w których dany bohater jest banowany (pozostali gracze nie zezwalają na jego wybranie)
- KDA - Stosunek zabójstw i asyst do śmierci danego bohatera wyrażony wzorem: (Zabójstwa + Asysty)/Śmierci

3.1.2 Australian Rain Forecast

Zbiór zawiera codzienne obserwacje dotyczące pogody z różnych lokalizacji na terenie Australii, zawierający następujące kolumny:

- Date - Data obserwacji
- Location - Potoczna nazwa lokalizacji, w której przeprowadzono pomiar
- MinTemp - Minimalna temperatura w stopniach celsjusza
- MaxTemp - Maksymalna temperatura w stopniach celsjusza
- Rainfall - Ilość opadów deszczu danego dnia wyrażona w wysokości słupa wody w mm
- Evaporation - Wskaźnik wyrażający zmianę poziomu wody w specjalnym naczyniu po 24h od uzupełnienia, wyrażony w mm
- Sunshine - Ilość słonecznych godzin w ciągu dnia
- WindGustDir - Kierunek najsilniejszego wiatru w ciągu 24 godzin do północy

- WindGustSpeed - Prędkość (km/h) najsilniejszego wiatru w ciągu 24 godzin do północy
- WindDir9am - Kierunek wiatru o 09:00
- WindDir3pm - Kierunek wiatru o 15:00
- WindSpeed9am - Prędkość wiatru (km/h) o 09:00
- WindSpeed3pm - Prędkość wiatru (km/h) o 15:00
- Humidity9am - Wilgotność o 09:00
- Humidity3pm - Wilgotność o 15:00
- Pressure9am - Ciśnienie atmosferyczne (hPa) względem poziomu morza o 09:00
- Pressure3pm - Ciśnienie atmosferyczne (hPa) względem poziomu morza o 15:00
- Cloud9am - Zachmurzenie o 09:00, mierzone w óktawach", wyrażających ile 1/8 nieba jest przysłonięte chmurami : 0 oznacza czyste niebo, 8 zupełnie zachmurzone
- Cloud3pm - Zachmurzenie o 15:00, mierzone w óktawach", wyrażających ile 1/8 nieba jest przysłonięte chmurami
- Temp9am - Temperatura (C) o 09:00
- Temp3pm - Temperatura (C) o 15:00
- RainToday - Informacja o deszczu danego dnia, granicę ustalono na 1mm
- RainTommorow - Informacja o deszczu jutrzejszego dnia, granicę ustalono na 1mm

3.1.3 Titanic Survival

Jest to zestaw informacji na temat pasażerów Titanica oraz tego, czy udało im się przeżyć, na podstawie czego budujemy model, który próbuje przewidzieć na podstawie informacji które mu przekażemy, czy dana osoba przeżyła katastrofę, w zbiorze tym znajdują się następujące kolumny:

- survival - Atrybut decyzyjny, określający czy pasażer przeżył
- pclass - Klasa, w jakiej podróżował pasażer
- sex - Płeć pasażera
- Age - Wiek pasażera
- sibsp - Liczba rodzeństwa lub małżonków na pokładzie

- parch - Liczba rodziców lub dzieci na pokładzie
- ticket - Numer biletu
- fare - Opłata, jaką zapłacił pasażer za bilet
- cabin - Kod kabiny, w jakiej podróżował pasażer
- embarked - Port w którym pasażer wsiadł na pokład

3.2 Przygotowanie danych do eksperymentów

Poniżej przedstawiono sposób w jaki przygotowano poszczególne zbiory przed przeprowadzeniem eksperymentów. Ważnym dla wyników było, aby braki w zbiorach były wygenerowane w sposób kontrolowany, stąd konieczność przygotowania zbiorów przed generacją braków.

3.2.1 Zbiór danych Titanic

Dla zestawu danych Titanic wypełniono brakujące wartości dla wieku oraz portu, w którym pasażer wsiadł na statek (Przykład 3.2)

```
titanic['Age'] = titanic['Age']
.fillna(titanic['Age'].mean())
titanic['Embarked'] = titanic['Embarked']
.fillna('S')
```

Przykład 3.1: Przygotowanie zbioru danych Titanic do eksperymentów

3.2.2 Zbiór danych Lol Stats

Połączono dostępne zbiory danych dla wersji 13.1 oraz 13.3, dodano odpowiednią klasę dla bohatera K'Sante, dokonano odpowiedniego kodowania atrybutu klasyfikującego ,a także sformatowano kolumny zawierające wartości procentowe

```
lol_stats = pd.concat([lol_13_1,lol_13_3])
for i in lol_stats.loc[lol_stats['Name'] == "K'Sante"].index.values:
    lol_stats.at[i,"Class"] = "Tank"
dict = {"God" : '0',
          "S" : '1',
          "A": '2',
          "B" : '3',
          "C": '4',
          "D": '5'}
lol_stats=lol_stats.replace({"Tier": dict})
percent_columns = ['Win_%', 'Role_%',
   'Pick_%', 'Ban_%']
```

```
    for col in percent_columns:
        lol_stats[col] = lol_stats[col].str.rstrip('%')
```

Przykład 3.2: Przygotowanie zbioru danych Lol Stats do eksperymentów

3.2.3 Zbiór danych Australian Rain Forecast

Z całego dostępnego zbioru danych wybrano jedynie dane dotyczące stacji pogodowych z Sydney, Lotniska Sydney, Melbourne, Lotniska Melbourne, Canberry, Newcastle oraz Perth, a także zakodowano odpowiednio wartości binarne

```
    aus_weather = aus_weather.dropna()
    aus_weather = aus_weather[aus_weather['Location'].isin(
        ['Sydney','SydneyAirport',
        'Canberra','MelbourneAirport',
        'Melbourne','Brisbane','Perth'])]
    aus_weather['RainToday'] = aus_weather['RainToday']
    .replace('Yes', '1')
    aus_weather['RainToday'] = aus_weather['RainToday']
    .replace('No', '0')
    aus_weather['RainTomorrow'] = aus_weather['RainTomorrow']
    .replace('Yes', '1')
    aus_weather['RainTomorrow'] = aus_weather['RainTomorrow']
    .replace('No', '0')
```

Przykład 3.3: Przygotowanie zbioru danych Australian Rain Forecast do eksperymentów

3.2.4 Część wspólna dla wszystkich zbiorów danych

Do przygotowania danych do eksperymentów służyła funkcja prepare_to_file (Przykład 3.1), która losowo generowała braki w danych, a następnie zapisywała nowopowstały niepełny zbiór danych do pliku.

```
    def prepare_to_file(df,work_columns,filename,count):
    path=r'C:\Users\Darek\Documents\Magisterka
␣␣␣␣\PracaMagPreproccessing\Datasets\Prepared'
    for i in range(count):
        df_copy = df.copy()
        df_random_rows = df.sample(frac=0.1)
        random_indexes = df_random_rows.index.tolist()
        for j in random_indexes:
            chosen_column = random.choice(work_columns)
            df_copy.at[j, chosen_column] = pd.NA
        df_copy.to_csv(path + '\\' + filename +
        '_'+str(i+1)+'.csv', index=False)
```

Przykład 3.4: Funkcja generująca braki w podanym zbiorze danych

Rozdział 4

Wykonane Eksperymenty

Dla każdego z wybranych zbiorów danych przeprowadzono przygotowanie danych według ustalonego scenariusza, a także dodatkowego scenariusza w którym głównym celem było indywidualne podejście do zbioru oraz wyciągnięcie możliwie jak najwięcej informacji z dostępnych danych

4.1 Użyte klasyfikatory

Poniżej odpisano klasyfikatory, jakich użyto do porównania wyników uzyskanych dzięki poszczególnym scenariuszom przygotowania danych

4.1.1 xgBoost

xgBoost to zoptymalizowana, zbiorowa biblioteka wzmacniająca gradient, zaprojektowana aby być wysoce efektywną, elastyczną oraz współpracującą z wieloma rodzajami sprzętu. XGBoost korzysta z równoległego wzmacniania drzew, dzięki którym rozwiązuje wiele problemów z dziedziny data science w szybki i dokładny sposób. [21]

4.1.2 Las Losowy

Wykorzystano RandomForestClassifier z biblioteki scikitlearn. Jest to algorytm uczenia maszynowego metodą zespołową, która polega na tworzeniu wielu drzew decyzyjnych podczas uczenia modelu. Przy przypisywaniu obiektu do danej klasy wybierana jest ta, którą wybrała największa ilość drzew. Las losowy eliminuje tendencje drzew decyzyjnych do nadmiernego dopasowywania się do zbioru uczącego. [23]

4.1.3 k-Najbliższych Sąsiadów

Wykorzystano KNeighborsClassifier z biblioteki scikitlearn. Danymi wejściowymi jest k najbliższych sąsiadów w zbiorze danych. Wyjściem jest przyna-

leżność do klasy danego obiektu. Obiekt jest klasyfikowany przez głosowanie większościowe spośród k-najbliższych ze zbioru uczącego. Zazwyczaj metryką używaną do głosowania jest odległość euklidesowa, jednak nie jest to jedyna możliwość. [22]

4.2 Brak Preprocessingu

Pierwszy scenariusz, który w przyszłości będzie punktem odniesienia polegał na usunięciu rekordów, w których występowały jakiekolwiek brakujące wartości

```
def no_preprocessing(df,num):
    df_1 = df.copy()
    df_1 = remove_missing(df_1)
    y = df_1['Survived']
    df_1= drop_columns(df_1,categorical)
    df_1 = df_1.apply(pd.to_numeric)
    X = df_1
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.1, random_state=42)
    score("Titanic",
        num,
        "No Preprocessing",
        X_train,y_train,X_test,y_test)
```

Przykład 4.1: Brak przygotowania danych dla zbioru danych Titanic

4.3 Wypełeninie brakujących wartości średnią

Drugi scenariusz zakładał wypełnienie brakujących wartości średnią za pomocą metod biblioteki pandas fillna() oraz mean()

```
def fill_missing_mean(df, work_columns):
    for col in work_columns:
        df[col] = df[col].fillna(df[col].mean())
    return df
```

Przykład 4.2: Wypełnienie brakujących wartości średnią

4.4 Wypełeninie brakujących wartości minimum

Kolejny scenariusz zakładał wypełnienie brakujących wartości średnią za pomocą metod biblioteki pandas fillna() oraz min()

```
def fill_missing_max(df, work_columns):
    for col in work_columns:
        df[col] = df[col].fillna(df[col].min())
```

```
    return df
```

Przykład 4.3: Wypełnienie brakujących wartości minimum

4.5 Wypełeninie brakujących wartości maksimum

Czwarty scenariusz zakładał wypełnienie brakujących wartości średnią za pomocą metod biblioteki pandas fillna() oraz max()

```
def fill_missing_max(df, work_columns):
    for col in work_columns:
        df[col] = df[col].fillna(df[col].max())
    return df
```

Przykład 4.4: Wypełnienie brakujących wartości maksimum

4.6 Wypełeninie brakujących wartości za pomocą regresji liniowej

W tym scenariuszu wykorzystano pozostałe rekordy, aby przy pomocy regresji liniowej wygenerować brakujące wartości

```
def fill_missing_regression(df, numeric):
    for col in numeric:
        df_num = df[numeric]
        test_data = df_num[df_num[col].isnull()]
        df_num = df_num.dropna()
        x_train = df_num.drop(col,axis=1)
        y_train = df_num[col]
        lr = LinearRegression()
        lr.fit(x_train,y_train)
        test_col = []
        for i in numeric:
            if(i != col):
                test_col.append(i)
        x_test = test_data[test_col]
        x_test = fill_missing_mean(x_test,test_col)
        y_pred = lr.predict(x_test)
        test_data[col] = y_pred
        for i in test_data.index.values:
            df.at[i,col] = test_data.loc[i][col]
        return df
```

Przykład 4.5: Wypełnienie brakujących wartości za pomocą regresji liniowej

4.7 Standaryzacja

Dzięki StandardScaler z biblioteki sklearn dokonano standaryzacji kolumn liczbowych

```
def standardize(df, work_columns):
    standard_scaler = preprocessing.StandardScaler()
    for col in work_columns:
        values = df[col].values
        df_scaled = standard_scaler.fit_transform(
            values.reshape(-1, 1))
        df_scaled = pd.DataFrame(df_scaled)
        df[col] = df_scaled
    return df
```

Przykład 4.6: Standaryzacja kolumn liczbowych

4.8 Standaryzacja oraz skalowanie do przedziału (0,1)

Dzięki StandardScaler z biblioteki sklearn dokonano standaryzacji wraz ze skalowaniem do przedziału (0,1) kolumn liczbowych

```
def normalize(df, work_columns):
    min_max_scaler = preprocessing.MinMaxScaler()
    for col in work_columns:
        values = df[col].values
        df_scaled = min_max_scaler.fit_transform(values.reshape(-1, 1))
        df_scaled = pd.DataFrame(df_scaled)
        df[col] = df_scaled
    return df
```

Przykład 4.7: Skalowanie do przedziału (0,1)

4.9 Skalowanie do przedziału (0,1) oraz usunięcie wartości odstających

Poza skalowaniem z poprzedniego scenariusza, wykorzystano algorytm LocalOutlierFactor do usunięcia wartości odstających

```
def remove_outliers_lof(df,work_columns):
    df_temp = df
    df_temp = df_temp.loc[:, work_columns]
    clf = LocalOutlierFactor(n_neighbors=2)
    clf.fit(df_temp)
```

```
    y_pred_outliers = clf.fit_predict(df_temp)
    df_temp['outlier'] = y_pred_outliers

    df_temp = df_temp.loc[df_temp['outlier'] == 1]
    df_temp.drop('outlier', axis=1, inplace=True)
    df_temp = df_temp.reset_index(drop=True)
    df = df[df.index.isin(df_temp.index)]
    return df
```

Przykład 4.8: Usuwanie wartości odstających

4.10 Kodowanie wartości kategorycznych

Wykorzystano LabelEncoder z biblioteki sklearn do zakodowania wartości kategorycznych na liczbowe. W przypadku, kiedy dla danej kolumny brakowało wartości, rekord usuwano

```
def encode_categorical(df, work_columns):
    encoder = LabelEncoder()
    for col in work_columns:
        df[col] = encoder.fit_transform(df[col])
    return df
```

Przykład 4.9: Usuwanie wartości odstających

4.11 Kodowanie wartości kategorycznych oraz wypełnienie brakujących wartości średnią

Po wykonaniu kodowania z poprzedniego scenariusza, wypełniono brakujące wartości wartością średnią

4.12 Scenariusz indywidualnego podejścia do zbioru danych

Jednym ze scenariuszy było dogłębne poznanie zbioru danych i pełne wykorzystanie jego potencjału, na przykład rozbicie kolumn na bardziej dokładne, zamiana typów danych lub inne bardziej skomplikowane operacje które można wykonać jedynie dla tego zbioru ze względu na jego charakterystykę

4.12.1 League of Legends stats

Dla statystyk z gry League of Legends jedyną dodatkową operacją względem poprzednich scenariuszy było usunięcie kolumny z imieniem postaci, jako że nie wnosiła ona konkretnych informacji, a głównie identyfikowała dany rekord

```
def custom_scenario(df,num):
    df_10 = df.copy()
    df_10 = fill_missing_mean(df_10,numeric)
    df_10 = remove_outliers_lof(df_10,numeric)
    df_10 = normalize(df_10,numeric)
    df_10= drop_columns(df_10,['Name'])
    df_10 = encode_categorical(df_10,to_be_encoded)
    y = df_10['Tier']
    df_10 = df_10.apply(pd.to_numeric)
    X = df_10
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.1, random_state=42)
    score(
        "LoL_Stats",
        num,
        "Custom_preprocessing",
        X_train,y_train,X_test,y_test)
```

Przykład 4.10: Indywidualny scenariusz dla zestawu danych LoL Stats

4.12.2 Australian Rain Forecast

Dla Australian Rain Forecast rozbito datę na dzień miesiąca, miesiąc oraz rok, dzięki czemu w teorii możemy zaobserwować trendy dotyczące na przykład poszczególnych miesięcy na przestrzeni wielu lat

```
def custom_scenario(df,num):
    df_10 = df.copy()
    df_10 = fill_missing_mean(df_10,numeric)
    df_10 = remove_outliers_lof(df_10,numeric)
    df_10 = normalize(df_10,numeric)
# Extract day, month and year from date
    df_10['Year'] = pd.DatetimeIndex(df_10['Date']).year
    df_10['Month'] = pd.DatetimeIndex(df_10['Date']).month
    df_10['Day'] = pd.DatetimeIndex(df_10['Date']).day
    df_10= drop_columns(df_10,['Date'])
    to_be_encoded = [
        'Location',
        'WindGustDir',
        'WindDir9am',
        'WindDir3pm',
        'RainToday',
        'Day',
        'Month',
        'Year']
    df_10 = encode_categorical(df_10,to_be_encoded)
```

```
    y = df_10['RainTomorrow']
    df_10 = df_10.apply(pd.to_numeric)
    X = df_10
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.1, random_state=42)
    score(
        "Aus_weather",num,"Custom_preprocessing",
        X_train,y_train,X_test,y_test)
```

Przykład 4.11: Indywidualny scenariusz dla zestawu danych Aus Rain Forecast

4.12.3 Titanic Survival

Dla zestawu danych Titanic wyciągnięto informację o tytule, jakim posługiwał się dany pasażer, dzięki czemu uzyskaliśmy informację o grupie społecznej, do której należeli pasażerowie. Dokonano również zmiany informacji o kabinie, którą zajmował pasażer, znacznie cenniejszą informacją od konkretnej kabiny jest sektor, w którym ona się znajdowała, informacja ta została uzyskana poprzez ograniczenie kodu kabiny jedynie do pierwszego znaku, który określa sektor

```
def custom_scenario(df,num):
    df_10 = df.copy()
    df_10['Title'] = df_10['Name'].str.extract(
        ' ([A-Za-z]+)\.', expand=False)
    df_10['Title'] = df_10['Title'].fillna(
        df_10['Title'].mode().iloc[0])
    df_10['Title'] = df_10['Title'].replace([
        'Lady', 'Countess','Capt', 'Col','Don', 'Dr',\
        'Major', 'Rev', 'Sir', 'Jonkheer', 'Dona'], 'Rare')
    df_10['Title'] = df_10['Title'].replace('Mlle', 'Miss')
    df_10['Title'] = df_10['Title'].replace('Ms', 'Miss')
    df_10['Title'] = df_10['Title'].replace('Mme', 'Mrs')
    df_10 = df_10.drop(['Name','Ticket','PassengerId'],axis=1)
    df_10['Cabin'] = df_10['Cabin'].fillna('000')
    df_10['Cabin'] = df_10['Cabin'].str[:1]
    df_10 = fill_missing_mean(df_10,numeric)
    df_10 = df_10.fillna(df_10.mode().iloc[0])
    to_be_encoded = ["Sex","Embarked","Cabin","Title"]
    df_10 = encode_categorical(df_10,to_be_encoded)
    y = df_10['Survived']
    df_10 = df_10.apply(pd.to_numeric)
    X = df_10
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.1, random_state=42)
    score(
```

```
"Titanic",num,"Custom_preprocessing",
X_train,y_train,X_test,y_test)
```

Przykład 4.12: Usuwanie wartości odstających

Rozdział 5

Wyniki Eksperymentów

W poniższym rozdziale przedstawiono wyniki eksperymentów dla każdego z klasyfikatorów, z podziałem na zbliżone do siebie scenariusze, jak i również średnie wyniki dla każdego ze zbiorów danych. W tablicach na zielono zaznaczono najlepszy wynik dla danego klasyfikatora przy danym wariancie wygnerowanych braków. Na kolor pomarańczowy zaznaczono najlepszą średnią obliczoną z trzech klasyfikatorów

5.1 League of Legends stats

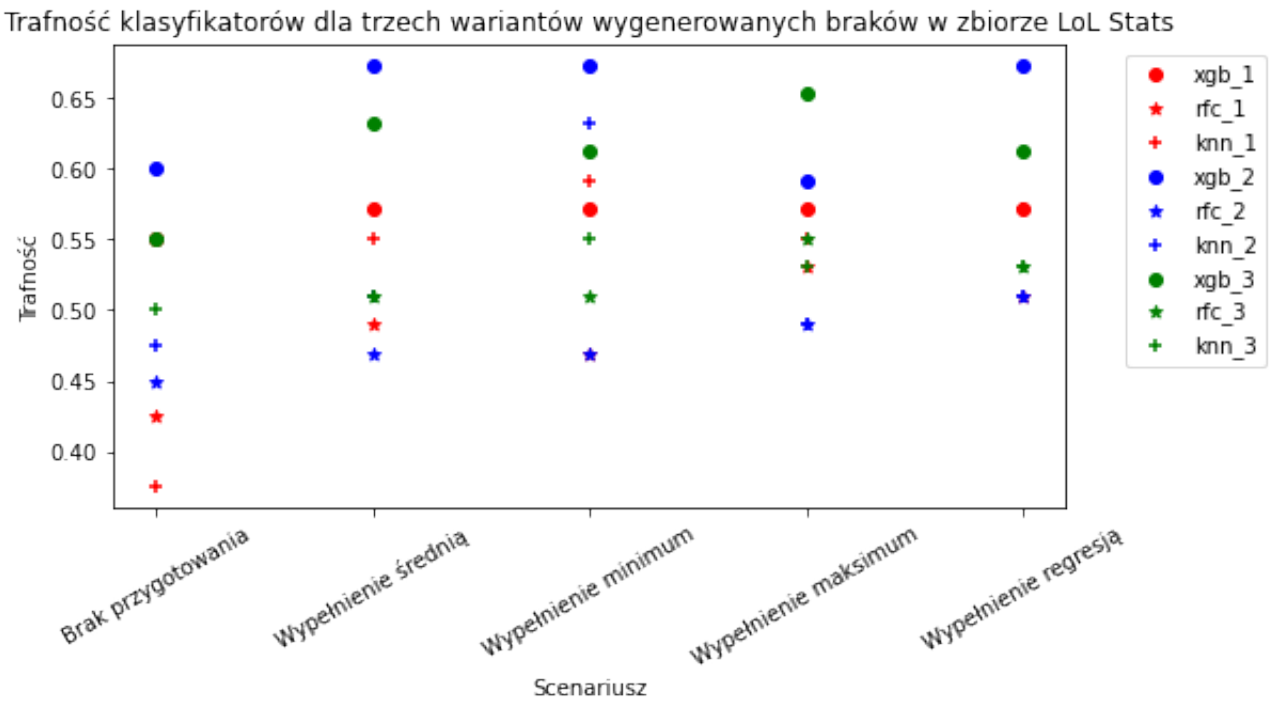
Oto wyniki dla zbioru statystyk z gry League of Legends z podziałem na rodzaj scenariusza przygotowania danych

5.1.1 Wypełnienie brakujących wartości

Poniżej (Tablica 5.1, Rysunek 5.1) przedstawiono trafność klasyfikatorów dla zbioru Lol Stats, po wypełnieniu brakujących wartości w zbiorze. Widać, że najlepsze rezultaty uzyskano wypełniając braki różnymi wartościami w zależno ści od klasyfikatora, natomiast każde wypełnienie brakujących wartości powodowało zwiększenie trafności klasyfikacji.

Tablica 5.1: Trafność klasyfikatorów dla zbioru Lol Stats po scenariuszach związanych z wypełnianiem brakujących wartości

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Lol Stats 1	Brak przygotowania	0,5500000	0,4250000	0,3750000	0,4500000	0,05204164999
	Wypełnienie średnią	0,5714286	0,4897959	0,5510204	0,5374150	0,0245275597
	Wypełnienie minimum	0,5714286	0,4693878	0,5918367	0,5442177	0,03787594805
	Wypełnienie maksimum	0,5714286	0,5306122	0,5510204	0,5510204	0,01178265855
	Wypełnienie regresją	0,5714286	0,5102041	0,5714286	0,5510204	0,02040816327
Lol Stats 2	Brak przygotowania	0,6000000	0,4500000	0,4750000	0,5083333	0,04639803636
	Wypełnienie średnią	0,6734694	0,4693878	0,5102041	0,5510204	0,06234796864
	Wypełnienie minimum	0,6734694	0,4693878	0,6326531	0,5918367	0,06234796864
	Wypełnienie maksimum	0,5918367	0,4897959	0,4897959	0,5238095	0,03401360544
	Wypełnienie regresją	0,6734694	0,5102041	0,5102041	0,5646259	0,05442176871
Lol Stats 3	Brak przygotowania	0,5500000	0,5000000	0,5500000	0,5333333	0,01666666667
	Wypełnienie średnią	0,6326531	0,5102041	0,5102041	0,5510204	0,04081632653
	Wypełnienie minimum	0,6122449	0,5510204	0,5102041	0,5578231	0,02965237377
	Wypełnienie maksimum	0,6530612	0,5306122	0,5510204	0,5782313	0,03787594805
	Wypełnienie regresją	0,6122449	0,5306122	0,5306122	0,5578231	0,02721088435



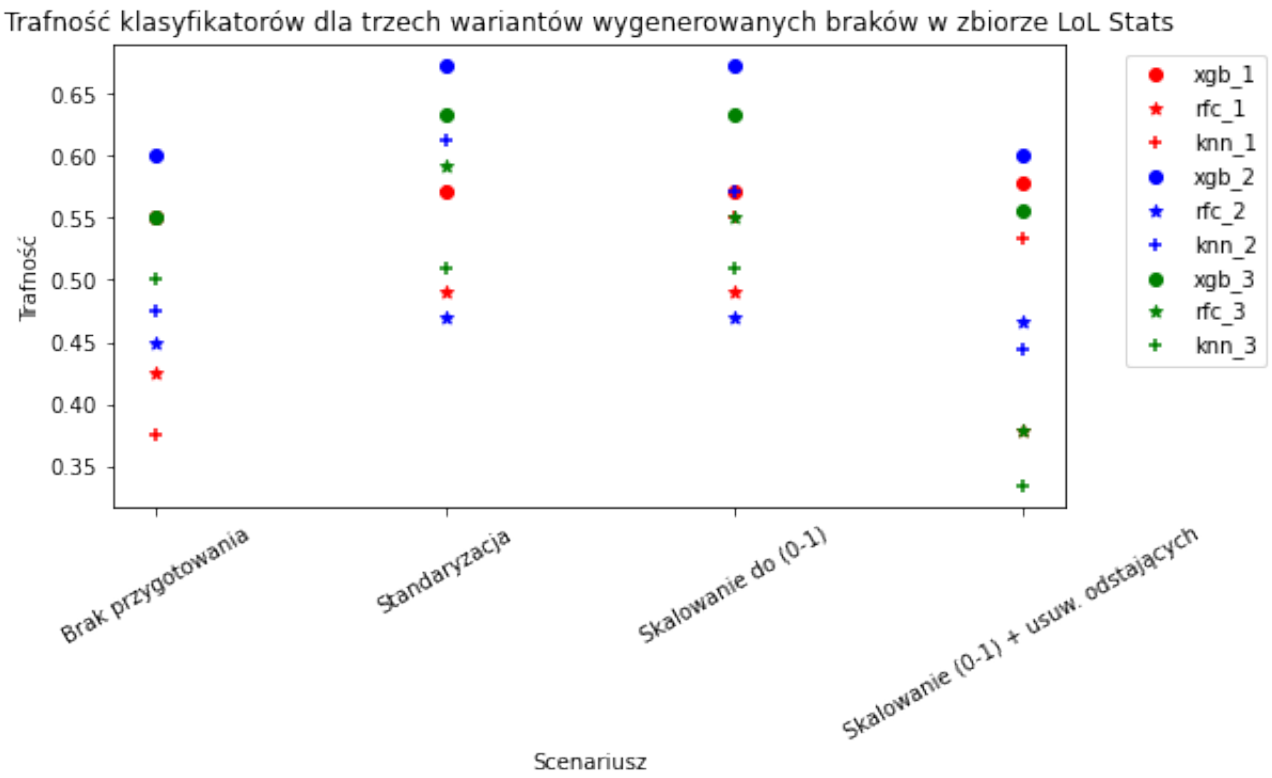
Rysunek 5.1: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Lol Stats po wypełnieniu brakujących wartości

5.1.2 Standaryzacja

Poniżej (Tablica 5.2, Rysunek 5.2) przedstawiono trafność klasyfikatorów dla zbioru Lol Stats, po przygotowaniu danych w sposób związany ze standaryzacją. Należy zwrócić uwage, że po skalowaniu wartości do przedziału (0,1) oraz usunięciu wartości odstających pogorszyła się trafność klasyfikacji. Najlepsze rezultaty uzyskano we wszystkich trzech przypadkach dla standaryzacji bez skalowania do przedziału (0,1)

Tablica 5.2: Trafność klasyfikatorów dla zbioru Lol Stats po scenariuszach związanych ze standaryzacją

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Lol Stats 1	Brak przygotowania	0,5500000	0,4250000	0,3750000	0,4500000	0,05204164999
	Standaryzacja	0,5714286	0,4897959	0,5714286	0,5442177	0,02721088435
	Skalowanie do (0-1)	0,5714286	0,4897959	0,5510204	0,5374150	0,0245275597
	Skalowanie i usuwanie wartości odstających	0,5777778	0,3777778	0,5333333	0,4962963	0,06063224275
Lol Stats 2	Brak przygotowania	0,6000000	0,4500000	0,4750000	0,5083333	0,04639803636
	Standaryzacja	0,6734694	0,4693878	0,6122449	0,5850340	0,0604639076
	Skalowanie do (0-1)	0,6734694	0,4693878	0,5714286	0,5714286	0,05891329277
	Skalowanie i usuwanie wartości odstających	0,6000000	0,4666667	0,4444444	0,5037037	0,0485736187
Lol Stats 3	Brak przygotowania	0,5500000	0,5000000	0,5500000	0,5333333	0,01666666667
	Standaryzacja	0,6326531	0,5102041	0,5918367	0,5782313	0,03599661648
	Skalowanie do (0-1)	0,6326531	0,5102041	0,5510204	0,5646259	0,03599661648
	Skalowanie i usuwanie wartości odstających	0,5555556	0,3333333	0,3777778	0,4222222	0,0678900103



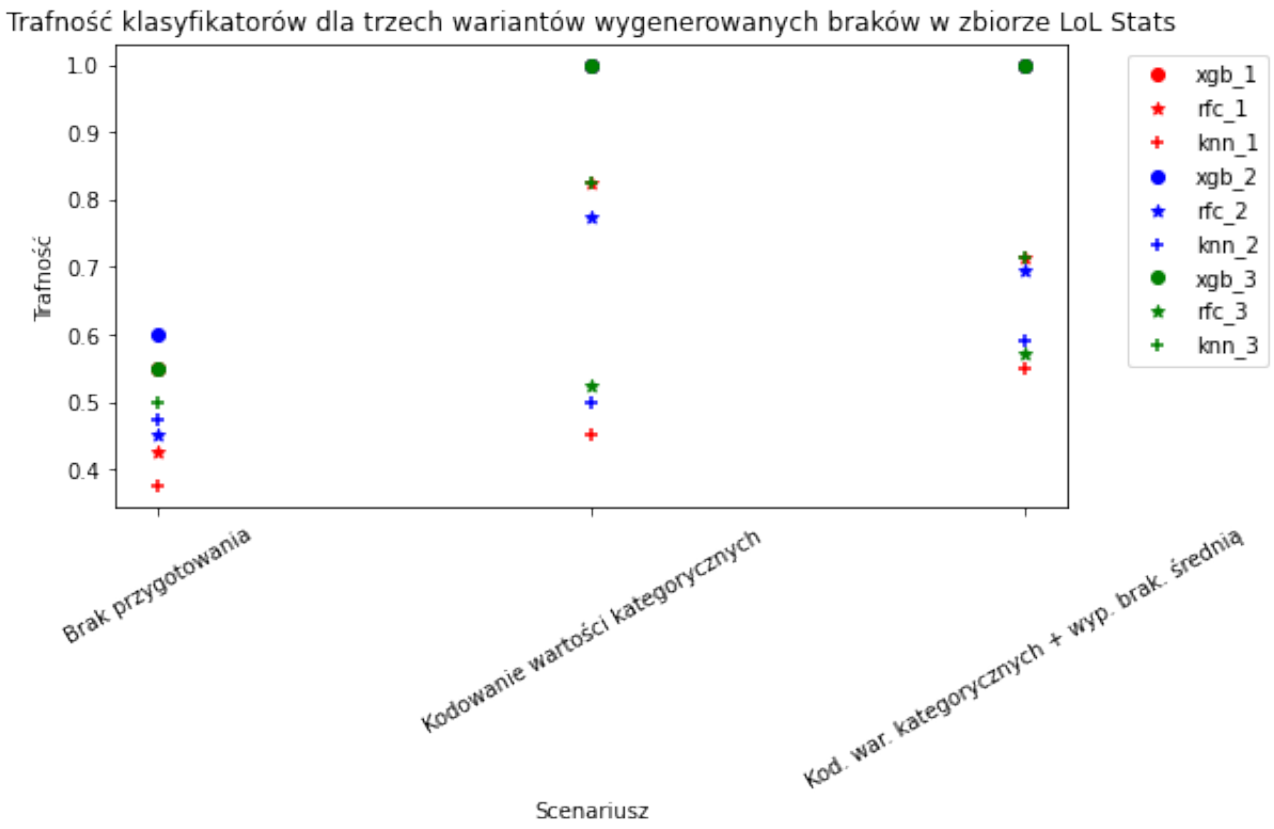
Rysunek 5.2: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Lol Stats po scenariuszach związanych ze standaryzacją

5.1.3 Kodowanie

Poniżej (Tablica 5.3, Rysunek 5.3) przedstawiono trafność klasyfikatorów dla zbioru Lol Stats, po przeprowadzeniu kodowania wartości kategorycznych. Dla klasyfikatora xgBoost kodowanie pozwala nam osiągnąć perfekcję w klasyfikacji, dla lasu losowego najlepsze wyniki osiągnięto przy samym kodowaniu, nastomiast dla k-najbliższych sąsiadów dodatkowe wypełnienie brakujących wartości średnią dało największą trafność.

Tablica 5.3: Trafność klasyfikatorów dla zbioru Lol Stats po scenariuszach związanych z kodowaniem

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Lol Stats 1	Brak przygotowania	0,5500000	0,4250000	0,3750000	0,4500000	0,05204164999
	Kodowanie wartości kategorycznych	1,0000000	0,8250000	0,4500000	0,7583333	0,1622326861
	Kodowanie wartości kat. oraz wypeł. brakujących średnią	1,0000000	0,7142857	0,5510204	0,7551020	0,1312061328
Lol Stats 2	Brak przygotowania	0,6000000	0,4500000	0,4750000	0,5083333	0,04639803636
	Kodowanie wartości kategorycznych	1,0000000	0,7750000	0,5000000	0,7583333	0,1445779298
	Kodowanie wartości kat. oraz wypeł. brakujących średnią	1,0000000	0,6938776	0,5918367	0,7619048	0,1226377985
Lol Stats 3	Brak przygotowania	0,5500000	0,5000000	0,5500000	0,5333333	0,01666666667
	Kodowanie wartości kategorycznych	1,0000000	0,8250000	0,5250000	0,7833333	0,1386943081
	Kodowanie wartości kat. oraz wypeł. brakujących średnią	1,0000000	0,7142857	0,5714286	0,7619048	0,1259881577



Rysunek 5.3: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Lol Stats po scenariuszach związanych ze kodowaniem

5.1.4 Indywidualne podejście

Poniżej (Tablica 5.4, Rysunek 5.4) przedstawiono trafność klasyfikatorów dla zbioru Lol Stats, po przeprowadzeniu przygotowania danych dostosowanego do zbioru. Dzięki takiemu przygotowaniu danych uzyskano najlepsze wyniki spośród wszystkich scenariuszy przygotowania danych. Należy zwrócić uwagę na różnicę między trafnością uzyskaną dzięki przygotowaniu danych, a wynikami bez przygotowania danych.

Tablica 5.4: Trafność klasyfikatorów dla zbioru Lol Stats po indywidualnym podejściu do zbioru

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Lol Stats 1	Brak przygotowania	0,5500000	0,4250000	0,3750000	0,4500000	0,05204164999
	Przygotowanie dostosowane do zbioru	1,0000000	0,6739130	0,9130435	0,8623188	0,09749002933
Lol Stats 2	Brak przygotowania	0,6000000	0,4500000	0,4750000	0,5083333	0,04639803636
	Przygotowanie dostosowane do zbioru	1,0000000	0,8000000	0,9555556	0,9185185	0,06063224275
Lol Stats 3	Brak przygotowania	0,5500000	0,5000000	0,5500000	0,5333333	0,01666666667
	Przygotowanie dostosowane do zbioru	1,0000000	0,6000000	0,9333333	0,8444444	0,123728097

Rysunek 5.4: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Lol Stats po indywidualnym podejściu dla zbiorów

5.2 Australian Rain Forecast

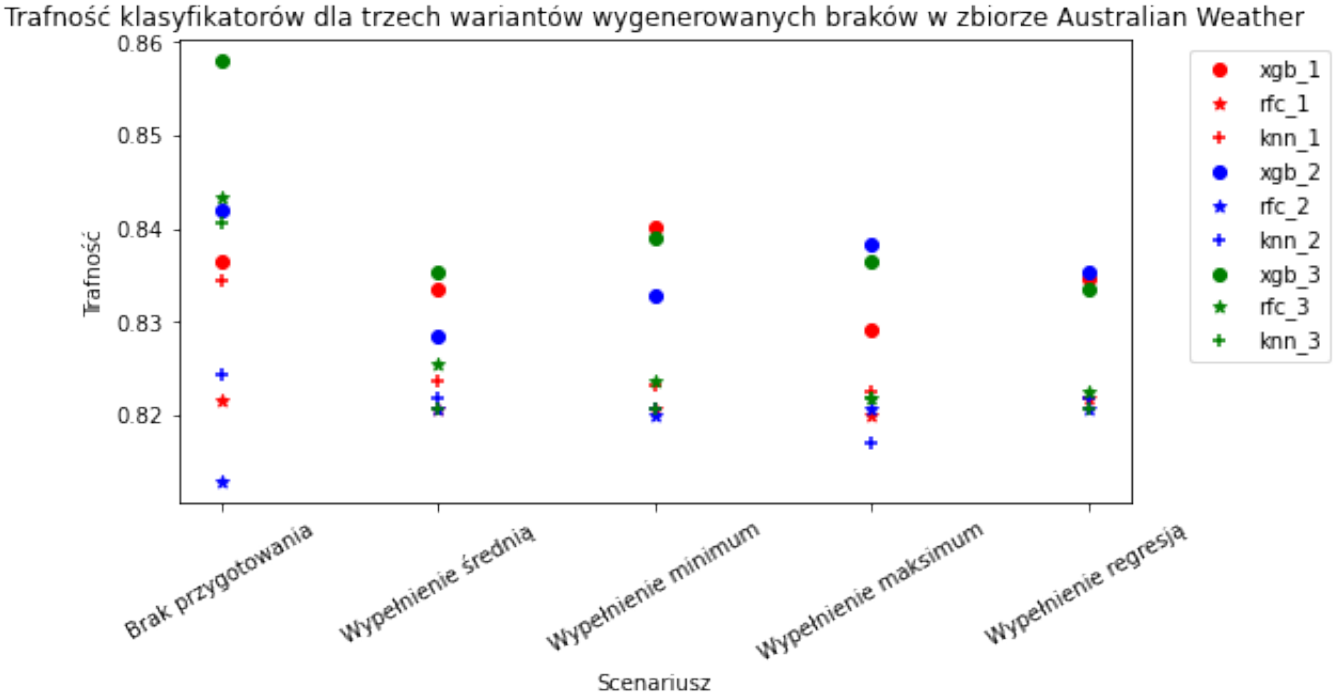
Poniżej przedstawiono wyniki dla zbioru Australian Rain Forecast z podziałem na rodzaj scenariusza przygotowania danych

5.2.1 Wypełnienie brakujących wartości

Poniżej (Tablica 5.5, Rysunek 5.5) przedstawiono trafność klasyfikatorów dla zbioru Australian Rain Forecast, po wypełnieniu brakujących wartości w zbiorze. Dla tego zbioru wypełnienie brakujących wartości w większości przypadków skutkowało pogorszeniem trafności klasyfikacji, może to być spowodowane wielkością zbioru, gdzie pojedyncze braki nie są aż tak istotne jak możliwe przekłamania spowodoane uzupełnianiem braków.

Tablica 5.5: Trafność klasyfikatorów dla zbioru Australian Rain Forecast po scenariuszach związanych z wypełnianiem brakujących wartości

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Australian Rain Forecast 1	Brak przygotowania	0,8364865	0,8216216	0,8344595	0,8308559	0,004654049171
	Wypełnienie średnią	0,8334347	0,8206687	0,8237082	0,8259372	0,003850050659
	Wypełnienie minimum	0,8401216	0,8206687	0,8231003	0,8279635	0,006119420008
	Wypełnienie maksimum	0,8291793	0,8200608	0,8224924	0,8239108	0,002726164954
	Wypełnienie regresją	0,8346505	0,8218845	0,8218845	0,8261398	0,004255319149
Australian Rain Forecast 2	Brak przygotowania	0,8418919	0,8128378	0,8243243	0,8263514	0,008448197898
	Wypełnienie średnią	0,8285714	0,8206687	0,8218845	0,8237082	0,002456809656
	Wypełnienie minimum	0,8328267	0,8200608	0,8206687	0,8245187	0,004157707098
	Wypełnienie maksimum	0,8382979	0,8206687	0,8170213	0,8253293	0,006569225969
	Wypełnienie regresją	0,8352584	0,8206687	0,8218845	0,8259372	0,004673784233
Australian Rain Forecast 3	Brak przygotowania	0,8581081	0,8405405	0,8432432	0,8472973	0,005461421465
	Wypełnienie średnią	0,8352584	0,8206687	0,8255319	0,8271530	0,004288958559
	Wypełnienie minimum	0,8389058	0,8206687	0,8237082	0,8277609	0,005641098645
	Wypełnienie maksimum	0,8364742	0,8218845	0,8218845	0,8267477	0,004863221884
	Wypełnienie regresją	0,8334347	0,8206687	0,8224924	0,8255319	0,003986284817



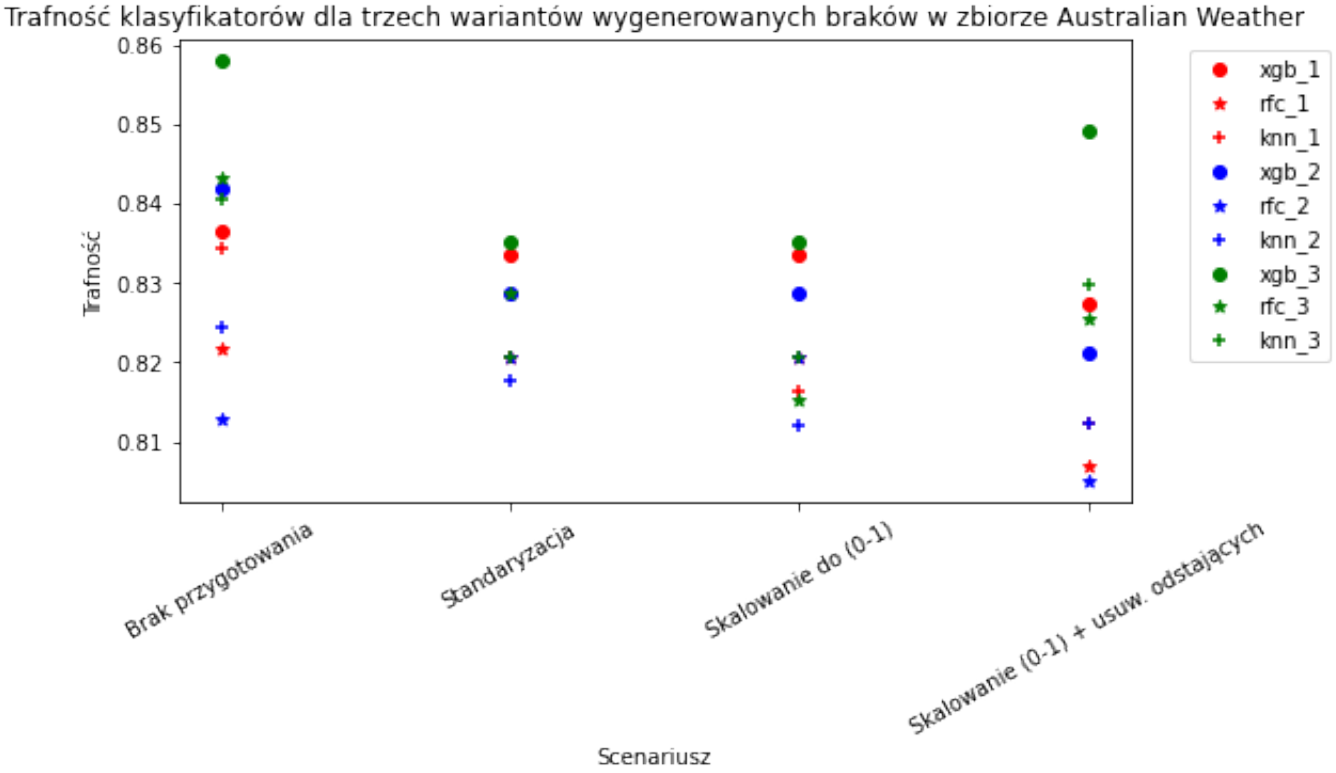
Rysunek 5.5: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Australian Rain Forecast po wypełnieniu brakujących wartości

5.2.2 Standaryzacja

Poniżej (Tablica 5.6, Rysunek 5.6) przedstawiono trafność klasyfikatorów dla zbioru Australian Rain Forecast, po scenariuszach związanych ze standaryzacją. Podobnie jak dla wypełniania brakujących wartości, w większości przypadków standaryzacja powoduje pogorszenie trafności klasyfikacji

Tablica 5.6: Trafność klasyfikatorów dla zbioru Australian Rain Forecast po scenariuszach związanych z standaryzacją

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Australian Rain Forecast 1	Brak przygotowania	0,8364865	0,8216216	0,8344595	0,8308559	0,00465404917
	Standaryzacja	0,8334347	0,8206687	0,8206687	0,8249240	0,00425531914
	Skalowanie do (0-1)	0,8334347	0,8206687	0,8164134	0,8235056	0,00511425712
	Skalowanie (0-1) i usuwanie wartości odstających	0,8272446	0,8068111	0,8123839	0,8154799	0,00609836396
Australian Rain Forecast 2	Brak przygotowania	0,8418919	0,8128378	0,8243243	0,8263514	0,00844819789
	Standaryzacja	0,8285714	0,8206687	0,8176292	0,8222898	0,00326108955
	Skalowanie do (0-1)	0,8285714	0,8206687	0,8121581	0,8204661	0,00473921603
	Skalowanie (0-1) i usuwanie wartości odstających	0,8210526	0,8049536	0,8123839	0,8127967	0,00465198252
Australian Rain Forecast 3	Brak przygotowania	0,8581081	0,8405405	0,8432432	0,8472973	0,00546142146
	Standaryzacja	0,8352584	0,8206687	0,8285714	0,8281662	0,00421654550
	Skalowanie do (0-1)	0,8352584	0,8206687	0,8151976	0,8237082	0,00598714
	Skalowanie (0-1) i usuwanie wartości odstających	0,8490099	0,8298267	0,8254950	0,8347772	0,00722536299



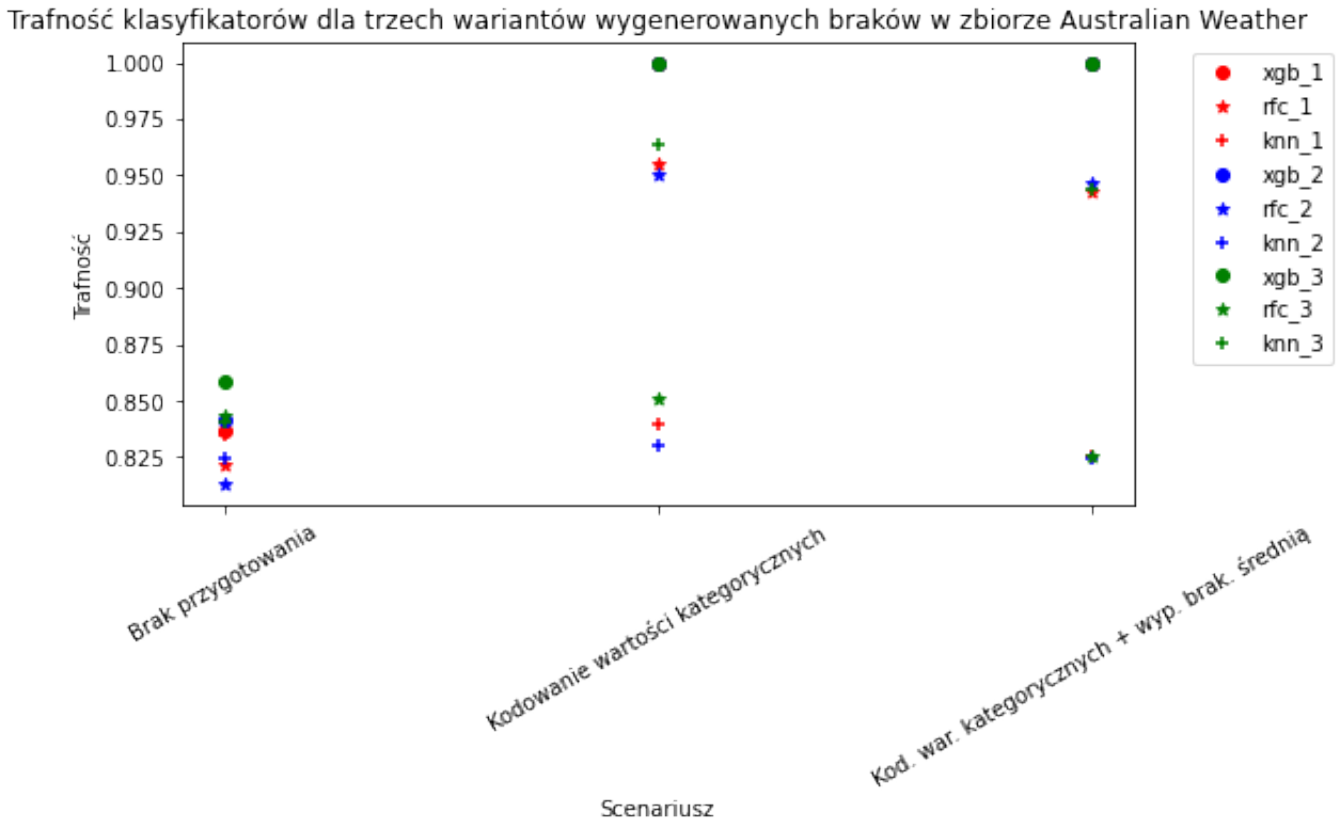
Rysunek 5.6: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Australian Rain Forecast po standaryzacji

5.2.3 Kodowanie

Poniżej (Tablica 5.7, Rysunek 5.7) przedstawiono trafność klasyfikatorów dla zbioru Australian Rain Forecast, po kodowaniu wartości kategorycznych. Dopiero kodowanie wartości kategorycznych przyniosło poprawę trafności klasyfikacji względem braku przygotowania. Dla dwóch z klasyfikatorów dodatkowe wypełnienie brakujących wartości nie przynosi zwiększenia trafności klasyfikacji.

Tablica 5.7: Trafność klasyfikatorów dla zbioru Australian Rain Forecast po scenariuszach związanych z kodowaniem

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Australian Rain Forecast 1	Brak przygotowania	0,8364865	0,8216216	0,8344595	0,8308559	0,004654049171
	Kodowanie wartości kateg.	1,0000000	0,9547297	0,8391892	0,9313063	0,04787665329
	Kodowanie wartości kateg. i wypełnianie war. brak. średnią	1,0000000	0,9428571	0,8255319	0,9227964	0,05135369075
Australian Rain Forecast 2	Brak przygotowania	0,8418919	0,8128378	0,8243243	0,8263514	0,008448197898
	Kodowanie wartości kategorycznych	1,0000000	0,9500000	0,8304054	0,9268018	0,05031301663
	Kodowanie wartości kateg. i wypełnianie war. brak. średnią	1,0000000	0,9465046	0,8243161	0,9236069	0,05199177759
Australian Rain Forecast 3	Brak przygotowania	0,8581081	0,8405405	0,8432432	0,8472973	0,005461421465
	Kodowanie wartości kategorycznych	1,0000000	0,9635135	0,8506757	0,9380631	0,04494527239
	Kodowanie wartości kateg. i wypełnianie war. brak. średnią	1,0000000	0,9440729	0,8249240	0,9229990	0,05162681561



Rysunek 5.7: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Australian Rain Forecast po kodowaniu wartości kategorycznych

5.2.4 Indywidualne podejście

Poniżej (Tablica 5.8, Rysunek 5.8) przedstawiono trafność klasyfikatorów dla zbioru Australian Rain Forecast, po przygotowaniu danych dostosowanych do zbioru. Dla klasyfikatorów xgBoost i lasu losowego indywidualne podejście przyniosło poprawę wyników, natomiast dla k-najbliższych sąsiadów pogorszyło trafność. Może to być związane z faktem, że częścią przygotowania w tym scenariuszu było rozbicie kolumny z datą na osobne kolumny dnia, miesiąca oraz roku, co mogło szczególnie negatywnie wpłynąć na klasyfikator k-najbliższych sąsiadów.

Tablica 5.8: Trafność klasyfikatorów dla zbioru Aus Rain Forecast po indywidualnym podejściu do zbioru

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Australian Rain Forecast 1	Brak przygotowania	0,8364865	0,8216216	0,8344595	0,8308559	0,004654049171
	Przygotowanie dostosowane do zbioru	1,0000000	0,9796296	0,8006173	0,9267490	0,06333940293
Australian Rain Forecast 2	Brak przygotowania	0,8418919	0,8128378	0,8243243	0,8263514	0,008448197898
	Przygotowanie dostosowane do zbioru	1,0000000	0,9833436	0,7809994	0,9214477	0,07038856187
Australian Rain Forecast 3	Brak przygotowania	0,8581081	0,8405405	0,8432432	0,8472973	0,005461421465
	Przygotowanie dostosowane do zbioru	1,0000000	0,9722736	0,7997535	0,9240090	0,06264119942

Rysunek 5.8: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Australian Rain Forecast po indywidualnym podejściu dla zbiorów

5.3 Titanic Survival

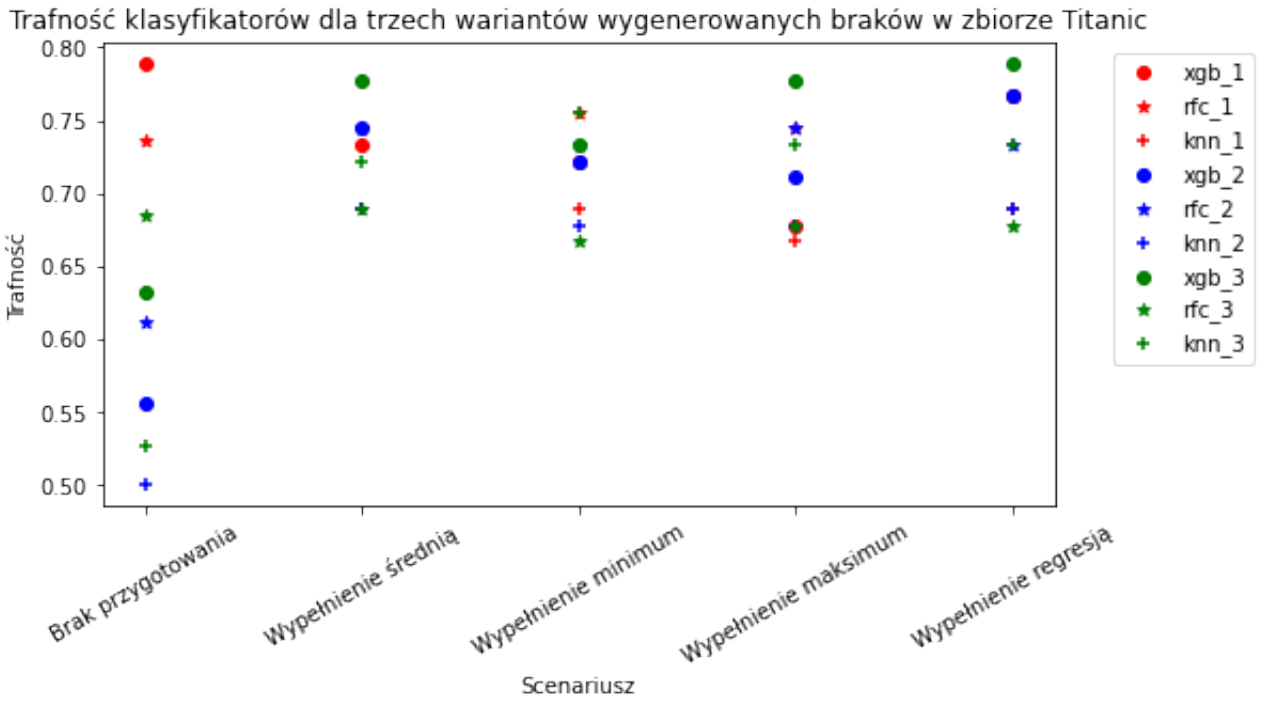
Poniżej przedstawiono wyniki dla zbioru Titanic z podziałem na rodzaj scenariusza przygotowania danych

5.3.1 Wypełnienie brakujących wartości

Poniżej (Tablica 5.9, Rysunek 5.9) przedstawiono trafność klasyfikatorów dla zbioru Titanic, po wypełnieniu brakujących wartości w zbiorze. Dla zdecydowanej większości przypadków wypełnienie brakujących wartości przynosi zwiększenie trafności klasyfikacji, jednak pomiędzy poszczególnymi przypadkami i klasyfikatorami różne wartości, którymi wypełniamy braki przynoszą najlepsze rezultaty.

Tablica 5.9: Trafność klasyfikatorów dla zbioru Titanic po scenariuszach związanych z wypełnianiem brakujących wartości

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Titanic 1	Brak przygotowania	0,7894737	0,7368421	0,6315789	0,7192982	0,04641668967
	Wypełnienie średnią	0,7333333	0,7444444	0,6888889	0,7222222	0,01697250257
	Wypełnienie minimum	0,7222222	0,7555556	0,6888889	0,7222222	0,01924500897
	Wypełnienie maksimum	0,6777778	0,7444444	0,6666667	0,6962963	0,02428680935
	Wypełnienie regresją	0,7666667	0,7666667	0,6888889	0,7407407	0,02592592593
Titanic 2	Brak przygotowania	0,5555556	0,6111111	0,5000000	0,5555556	0,03207501495
	Wypełnienie średnią	0,7444444	0,7444444	0,6888889	0,7259259	0,01851851852
	Wypełnienie minimum	0,7222222	0,7333333	0,6777778	0,7111111	0,01697250257
	Wypełnienie maksimum	0,7111111	0,7444444	0,6777778	0,7111111	0,01924500897
	Wypełnienie regresją	0,7666667	0,7333333	0,6888889	0,7296296	0,02252875011
Titanic 3	Brak przygotowania	0,6315789	0,5263158	0,6842105	0,6140351	0,04641668967
	Wypełnienie średnią	0,7777778	0,7222222	0,6888889	0,7296296	0,02592592593
	Wypełnienie minimum	0,7333333	0,7555556	0,6666667	0,7185185	0,02670778723
	Wypełnienie maksimum	0,7777778	0,7333333	0,6777778	0,7296296	0,02892685065
	Wypełnienie regresją	0,7888889	0,7333333	0,6777778	0,7333333	0,03207501495



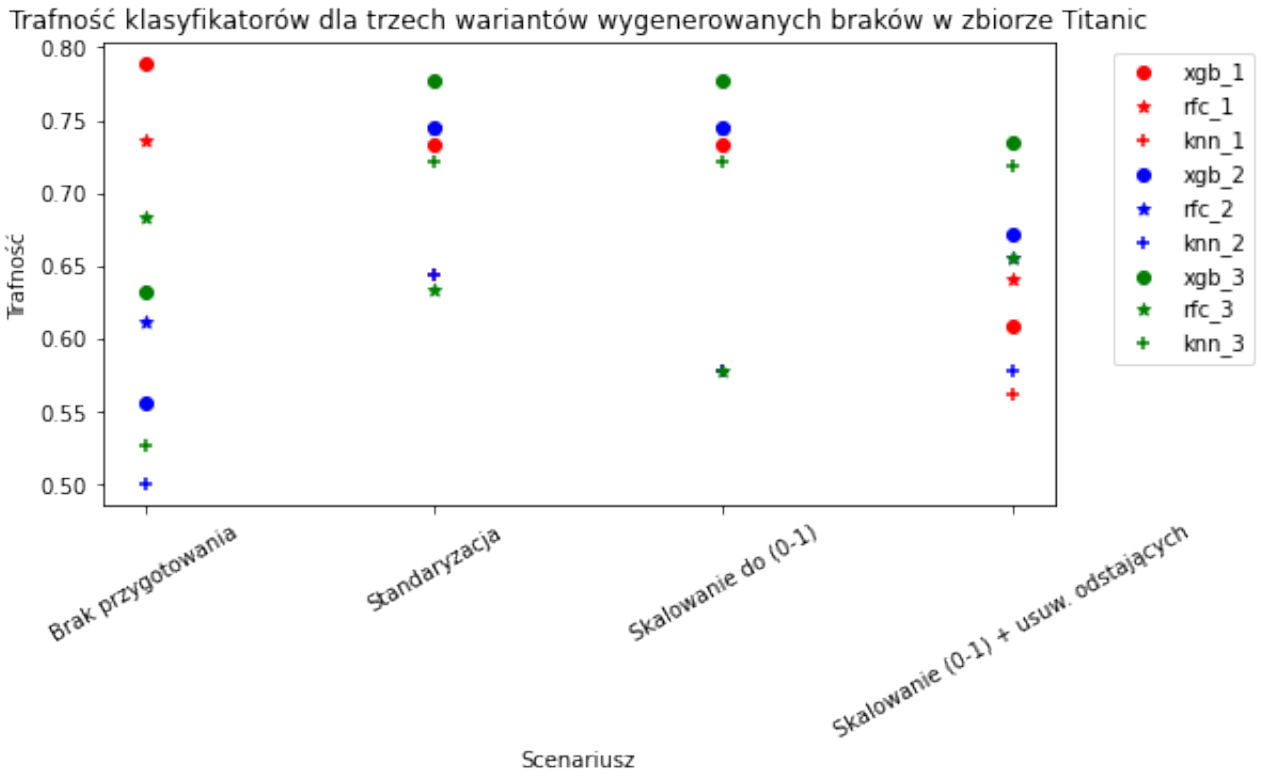
Rysunek 5.9: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Titanic po wypełnieniu brakujących wartości

5.3.2 Standaryzacja

Poniżej (Tablica 5.10, Rysunek 5.10) przedstawiono trafność klasyfikatorów dla zbioru Titanic, po scenariuszach związanych ze standaryzacją. Standaryzacja bez skalowania pozwala na uzyskanie najlepszych wyników, jednak należy zauważyć, że nie we wszystkich przypadkach.

Tablica 5.10: Trafność klasyfikatorów dla zbioru Titanic po scenariuszach związanych ze standaryzacją

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Titanic 1	Brak przygotowania	0,7894737	0,7368421	0,6315789	0,7192982	0,0464166896
	Standaryzacja	0,7333333	0,7444444	0,6444444	0,7074074	0,0316444583
	Skalowanie do (0-1)	0,7333333	0,7444444	0,5777778	0,6851852	0,0537994038
	Skalowanie (0-1) i usuwanie wartości odstających	0,6093750	0,6406250	0,5625000	0,6041667	0,0227025986
Titanic 2	Brak przygotowania	0,5555556	0,6111111	0,5000000	0,5555556	0,0320750149
	Standaryzacja	0,7444444	0,7444444	0,6444444	0,7111111	0,0333333333
	Skalowanie do (0-1)	0,7444444	0,7444444	0,5777778	0,6888889	0,0555555555
	Skalowanie (0-1) i usuwanie wartości odstających	0,6718750	0,6562500	0,5781250	0,6354167	0,0289987727
Titanic 3	Brak przygotowania	0,6315789	0,5263158	0,6842105	0,6140351	0,0464166896
	Standaryzacja	0,7777778	0,7222222	0,6333333	0,7111111	0,0420659877
	Skalowanie do (0-1)	0,7777778	0,7222222	0,5777778	0,6925926	0,0596054701
	Skalowanie (0-1) i usuwanie wartości odstających	0,7343750	0,7187500	0,6562500	0,7031250	0,0238675817



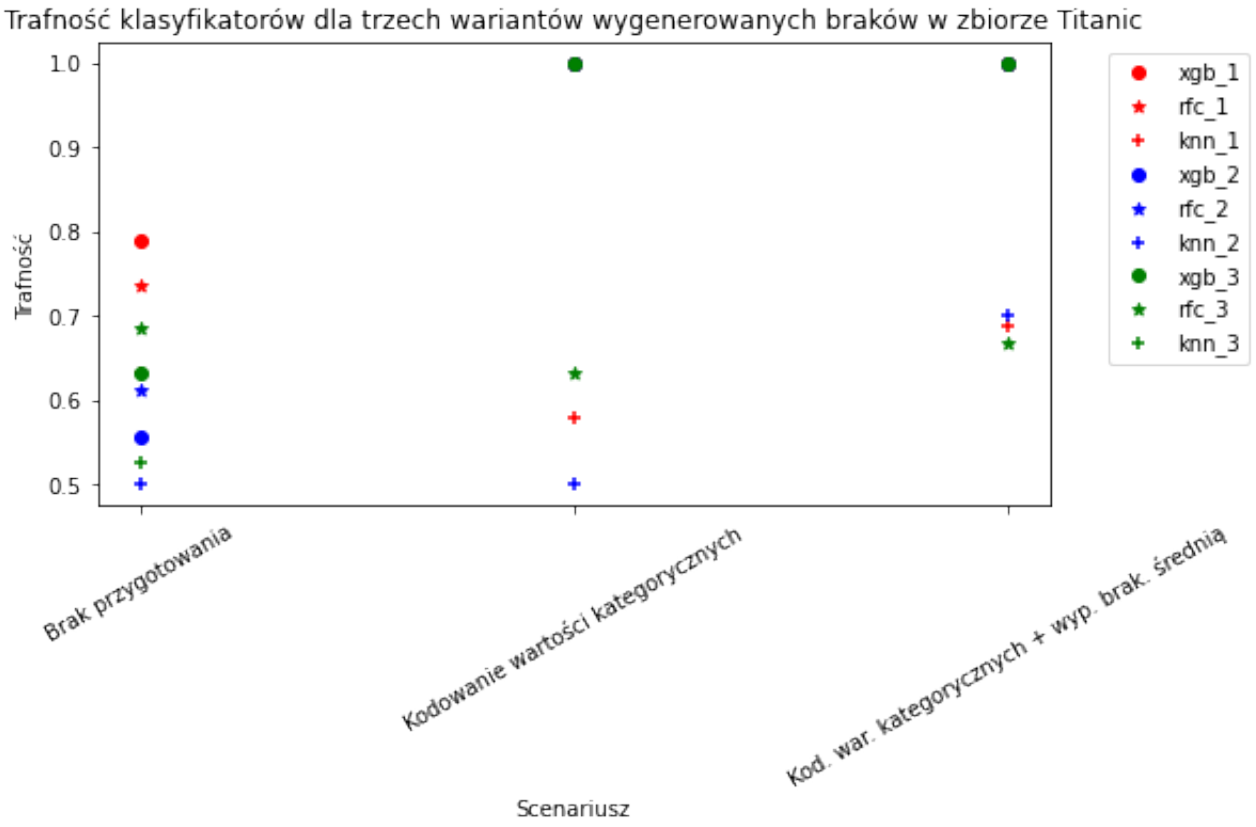
Rysunek 5.10: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Titanic po scenariuszach związanych ze standaryzacją

5.3.3 Kodowanie

Poniżej (Tablica 5.11, Rysunek 5.11) przedstawiono trafność klasyfikatorów dla zbioru Titanic, po kodowaniu wartości kategorycznych. Największą trafność klasyfikatory osiągają dla kodowania wartości oraz wypełniania brakujących wartości średnią. Dla xgBoost oraz lasu losowego kodowanie pozwala nam uzyskać 100% trafność klasyfikacji

Tablica 5.11: Trafność klasyfikatorów dla zbioru Titanic po scenariuszach związanych z kodowaniem

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Titanic 1	Brak przygotowania	0,7894737	0,7368421	0,6315789	0,7192982	0,04641668967
	Kodowanie wartości kateg.	1,0000000	1,0000000	0,5789474	0,8596491	0,1403508772
	Kodowanie wartości kateg. i wypełnianie war. brak. średnią	1,0000000	1,0000000	0,6888889	0,8962963	0,1037037037
Titanic 2	Brak przygotowania	0,5555556	0,6111111	0,5000000	0,5555556	0,03207501495
	Kodowanie wartości kategorycznych	1,0000000	1,0000000	0,5000000	0,8333333	0,1666666667
	Kodowanie wartości kateg. i wypełnianie war. brak. średnią	1,0000000	1,0000000	0,7000000	0,9000000	0,1
Titanic 3	Brak przygotowania	0,6315789	0,5263158	0,6842105	0,6140351	0,04641668967
	Kodowanie wartości kategorycznych	1,0000000	1,0000000	0,6315789	0,8771930	0,1228070175
	Kodowanie wartości kateg. i wypełnianie war. brak. średnią	1,0000000	1,0000000	0,6666667	0,8888889	0,1111111111



Rysunek 5.11: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Titanic po scenariuszach związanych ze kodowaniem

5.3.4 Indywidualne podejście

Poniżej (Tablica 5.12, Rysunek 5.12) przedstawiono trafność klasyfikatorów dla zbioru Titanic, po przygotowaniu danych dostosowanym do zbioru. Dla wszystkich klasyfikatorów dzięki indywidualnemu podejściu do zbioru uzyskano poprawę trafności klasyfikacji.

Tablica 5.12: Trafność klasyfikatorów dla zbioru Titanic po indywidualnym podejściu do zbioru

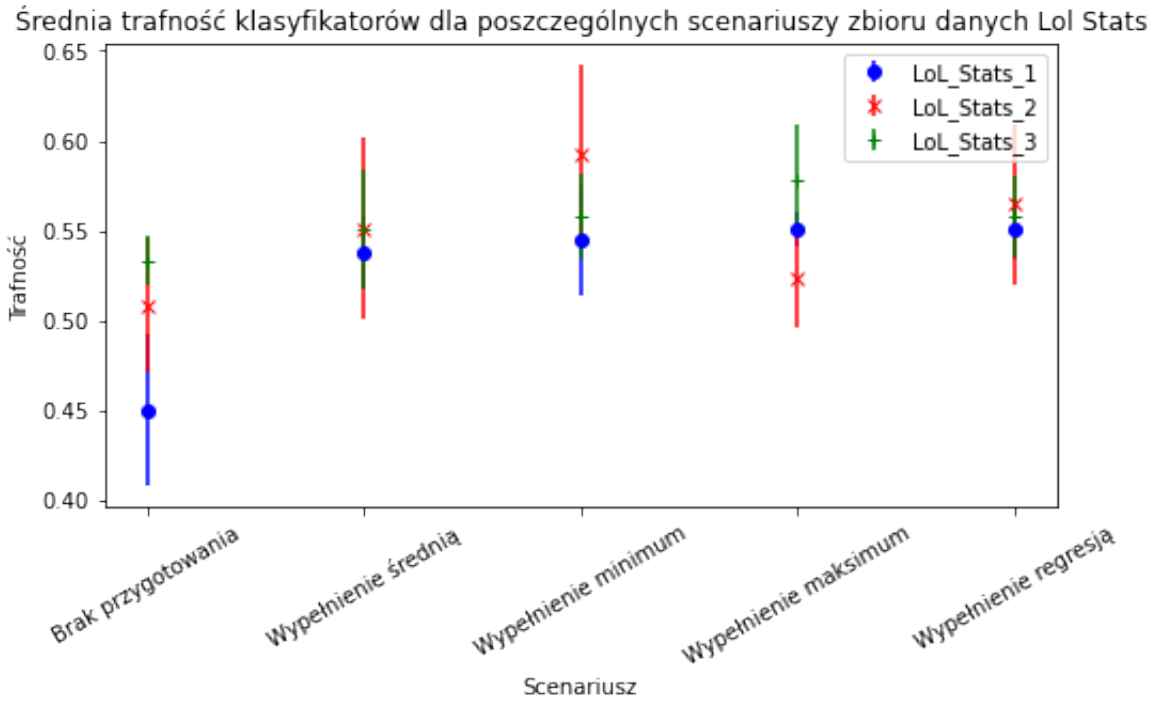
Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów	Średnia trafność	Błąd standardowy
Titanic 1	Brak przygotowania	0,7894737	0,7368421	0,6315789	0,7192982	0,04641668967
	Przygotowanie dostosowane do zbioru	1,0000000	1,0000000	0,8333333	0,9444444	0,05555555556
Titanic 2	Brak przygotowania	0,5555556	0,6111111	0,5000000	0,5555556	0,03207501495
	Przygotowanie dostosowane do zbioru	1,0000000	1,0000000	0,8333333	0,9444444	0,05555555556
Titanic 3	Brak przygotowania	0,6315789	0,5263158	0,6842105	0,6140351	0,04641668967
	Przygotowanie dostosowane do zbioru	1,0000000	1,0000000	0,8222222	0,9407407	0,05925925926

Rysunek 5.12: Trafność klasyfikatorów dla trzech wariantów wygenerowanych braków w zbiorze Titanic po indywidualnym podejściu dla zbiorów

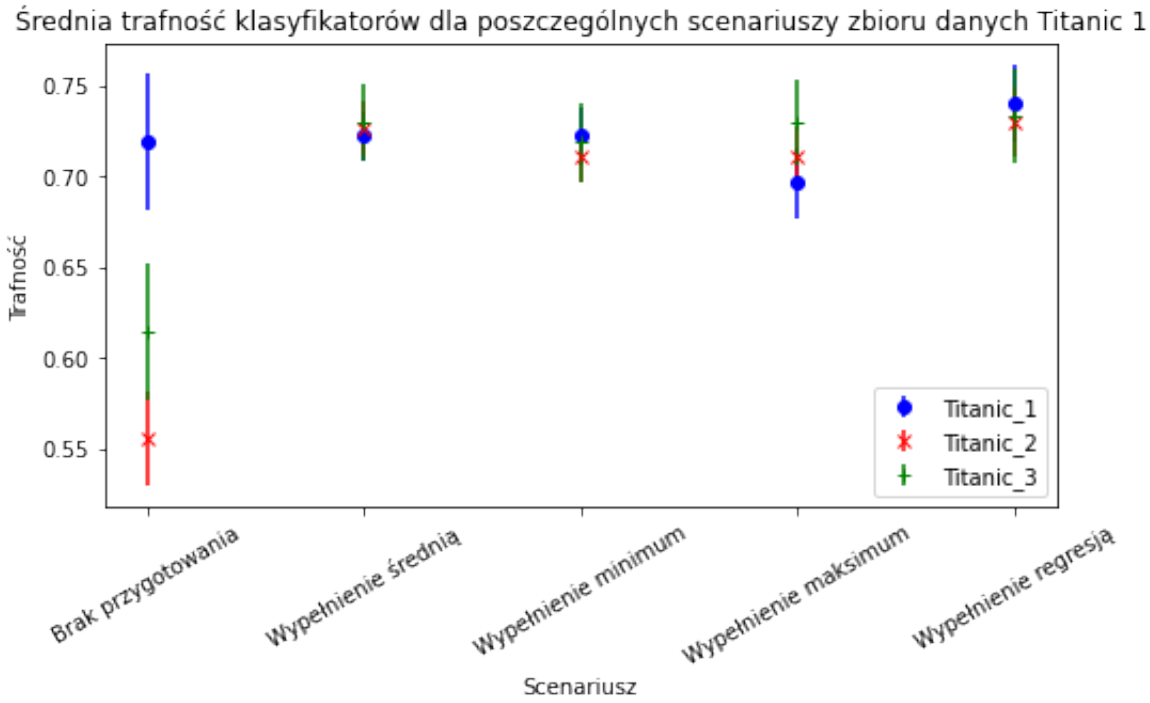
5.4 Średnie wyniki

5.4.1 Wypełnienie brakujących wartości

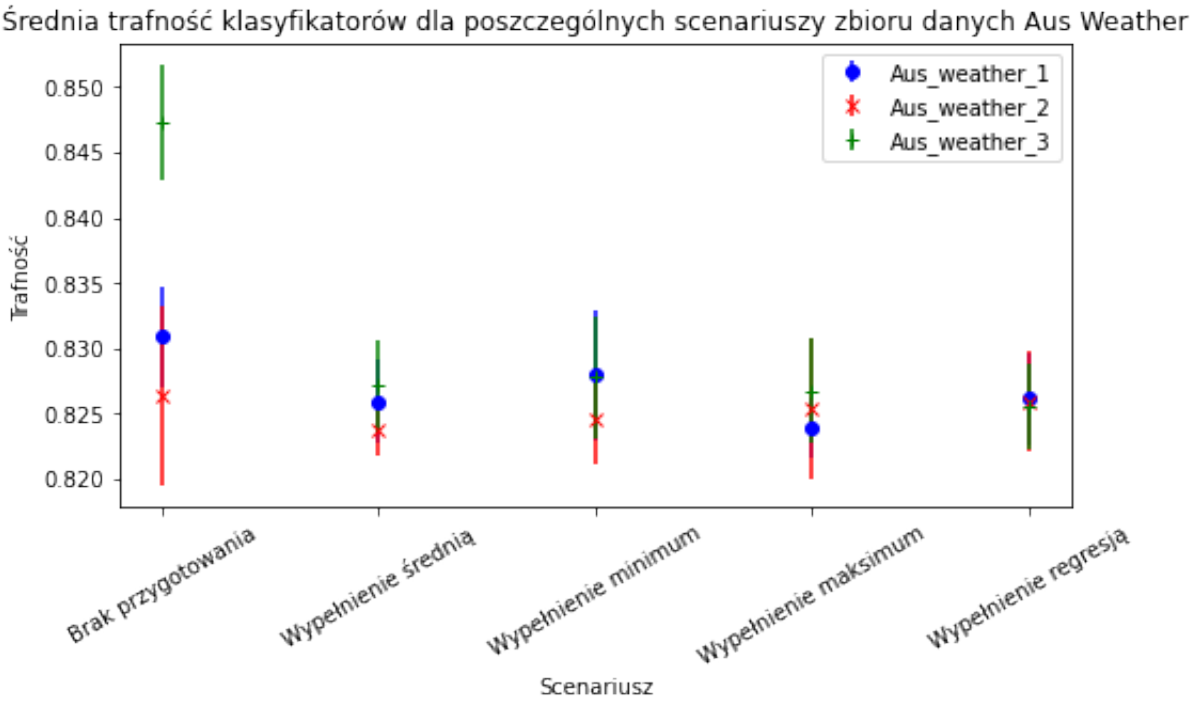
Dla Lol Stats oraz Australian Rain Forecast średnia trafność nieznacznie maleje po wypełnieniu brakujących wartości (Rysunek 5.37, Rysunek 5.39). Dla zbioru Titanic w każdym przypadku po wypełnieniu brakujących wartości średnia trafność klasyfikacji wzrosła (Rysunek 5.38)



Rysunek 5.13: Średnia trafność klasyfikatorów dla zbioru danych Lol Stats dla grupy scenariuszy wypełniania brakujących wartości



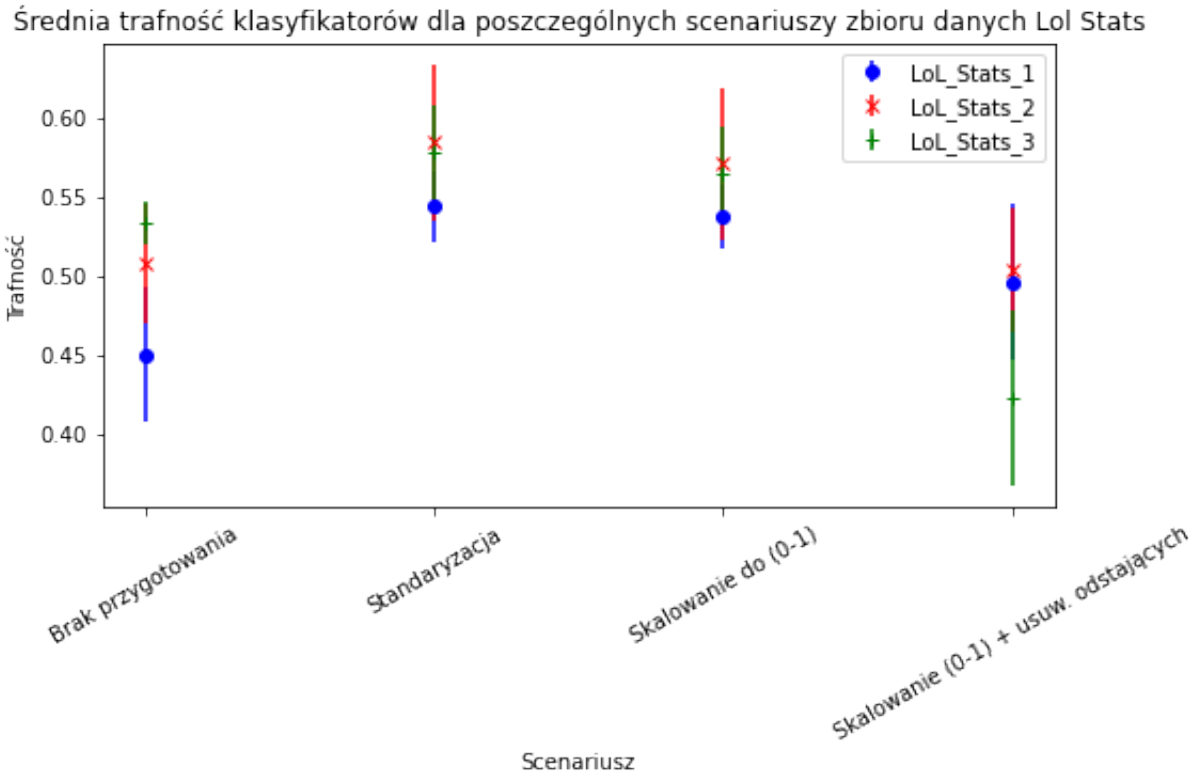
Rysunek 5.14: Średnia trafność klasyfikatorów dla zbioru danych Titanic dla grupy scenariuszy wypełniania brakujących wartości



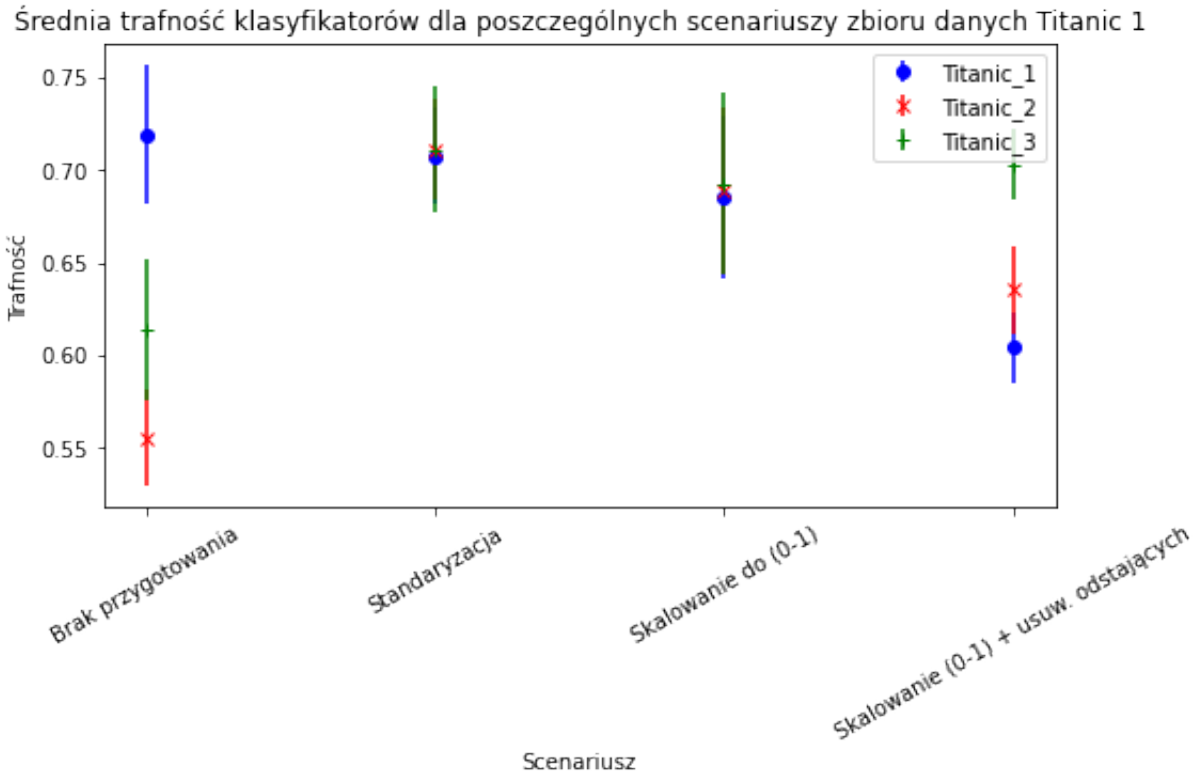
Rysunek 5.15: Średnia trafność klasyfikatorów dla zbioru danych Australian Rain Forecast dla grupy scenariuszy wypełniania brakujących wartości

5.4.2 Standaryzacja

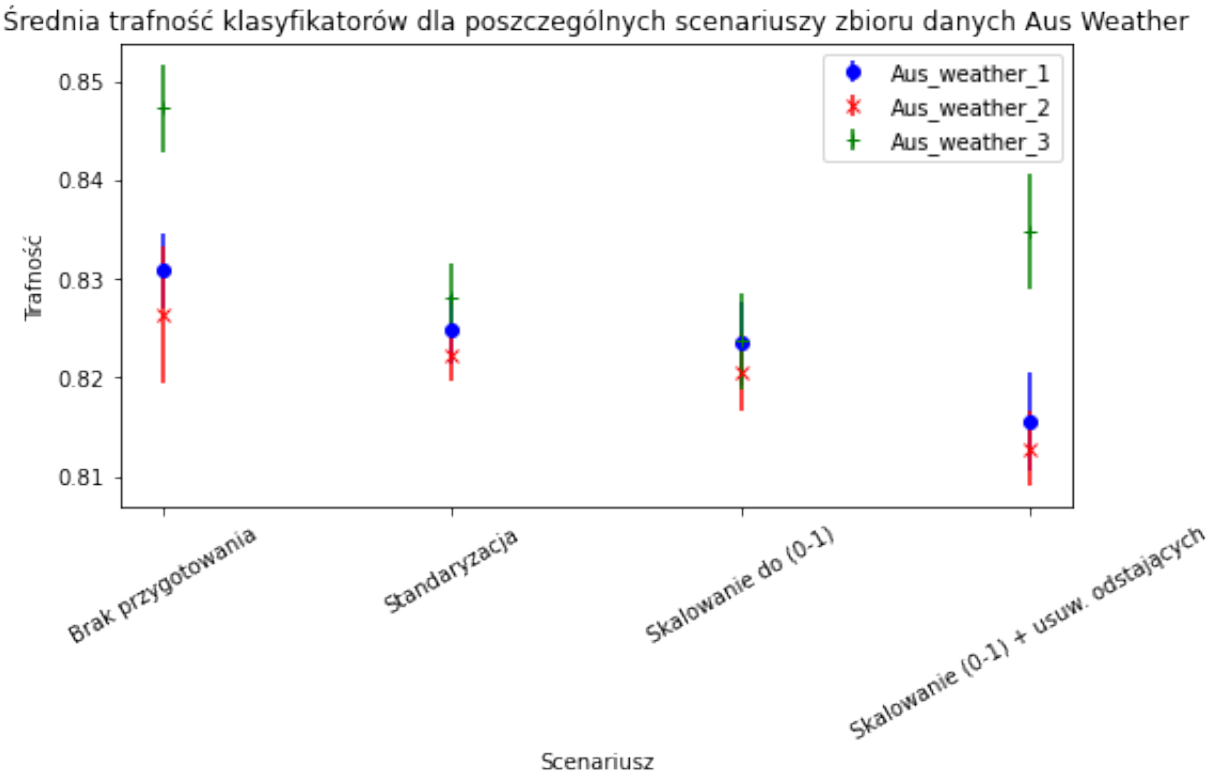
W kwestii Standaryzacji dla zestawu danych Lol Stats najlepszy wynik uzyskujemy dla Standaryzacji oraz Standaryzacji do przedziału (0,1) (Rysunek 5.40), dla Titanic Standaryzacji (Rysunek 5.41), a dla Australian Rain Forecast dla braku przygotowania (Rysunek 5.42)



Rysunek 5.16: Średnia trafność klasyfikatorów dla zbioru danych Lol Stats dla standaryzacji



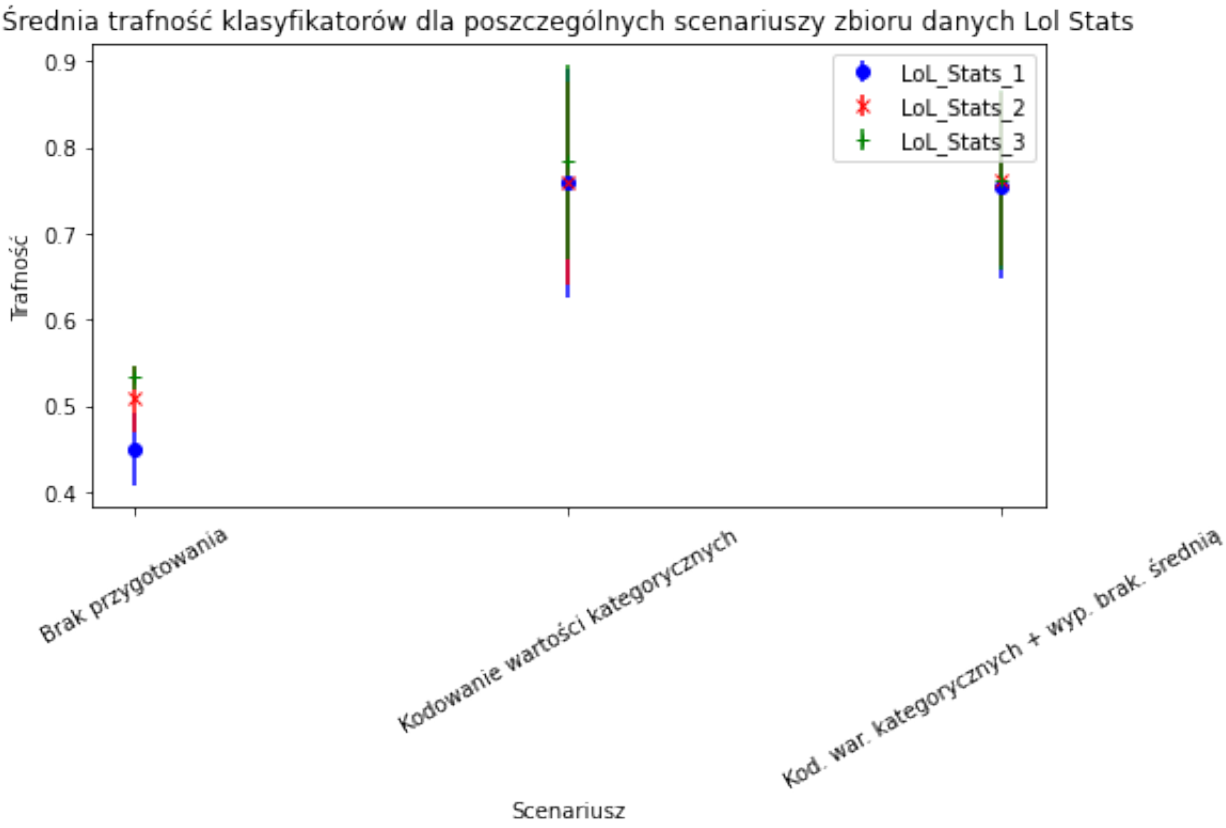
Rysunek 5.17: Średnia trafność klasyfikatorów dla zbioru danych Titanic dla standaryzacji



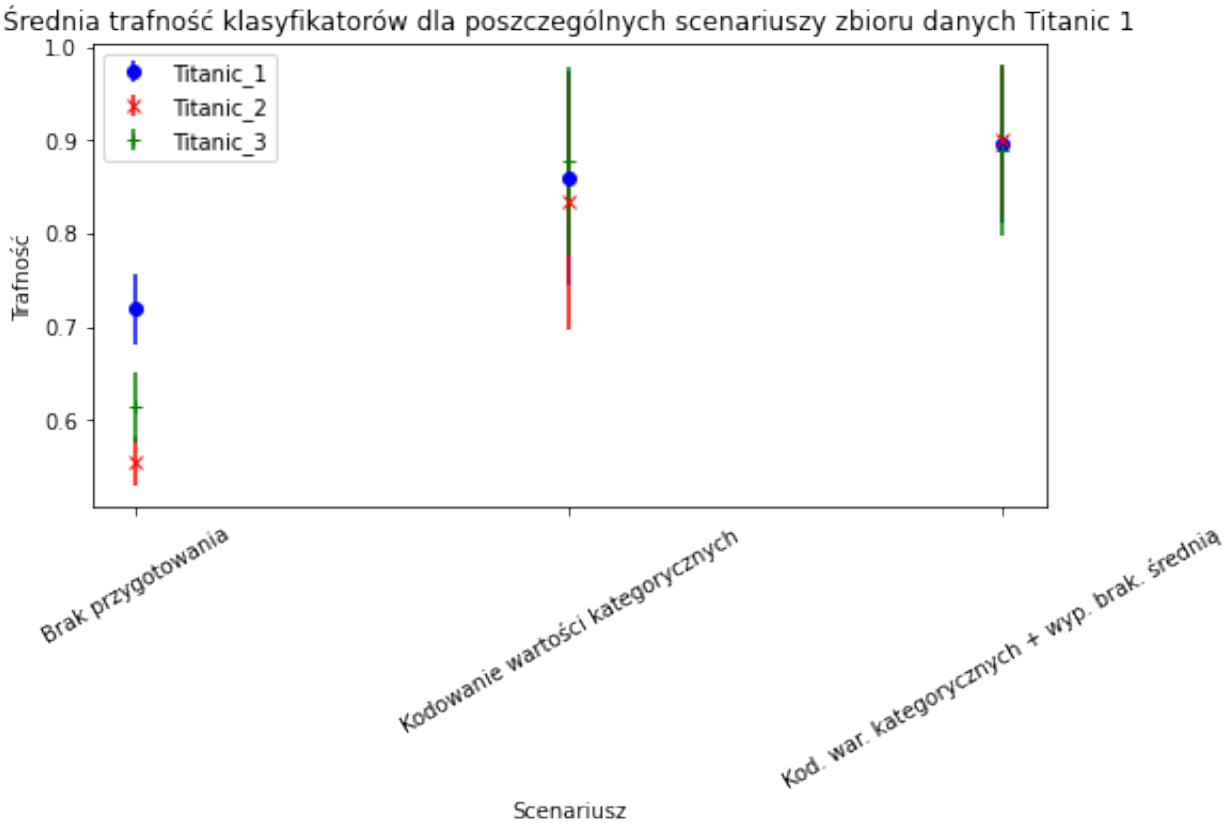
Rysunek 5.18: Średnia trafność klasyfikatorów dla zbioru danych Australian Rain Forecast dla standaryzacji

5.4.3 Kodowanie

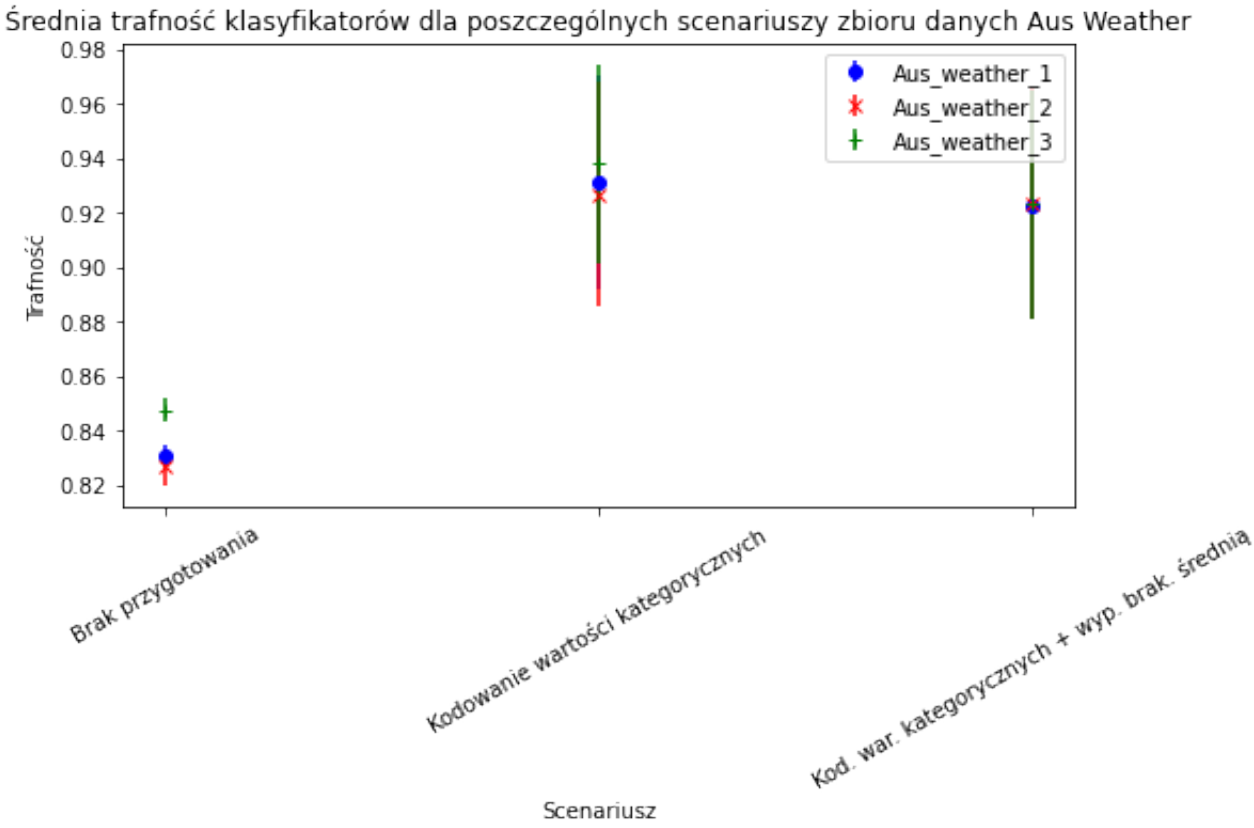
Dla każdego zestawu danych Standaryzacja daje poprawę średniej trafności klasyfikacji, istnieją jednak pewne różnice pomiędzy samym kodowaniem, a kodowaniem z wypełnianiem brakujących wartości. Dla zbioru Lol Stats samo kodowanie powoduje dość rozbieżne wyniki pomiędzy trzema przypadkami wygenerowanych braków, jednak kodowanie z wypełnianiem daje w miarę spójny obraz średniej trafności (Rysunek 5.43), dla Titanic wypełnianie nieznacznie zwiększa średnią trafność klasyfikacji (Rysunek 5.44), natomiast dla Australian Rain Forecast najlepszy wynik uzyskujemy dla kodowania bez wypełniania brakujących wartości (Rysunek 5.45)



Rysunek 5.19: Średnia trafność klasyfikatorów dla zbioru danych Lol Stats dla kodowania



Rysunek 5.20: Średnia trafność klasyfikatorów dla zbioru danych Titanic dla kodowania

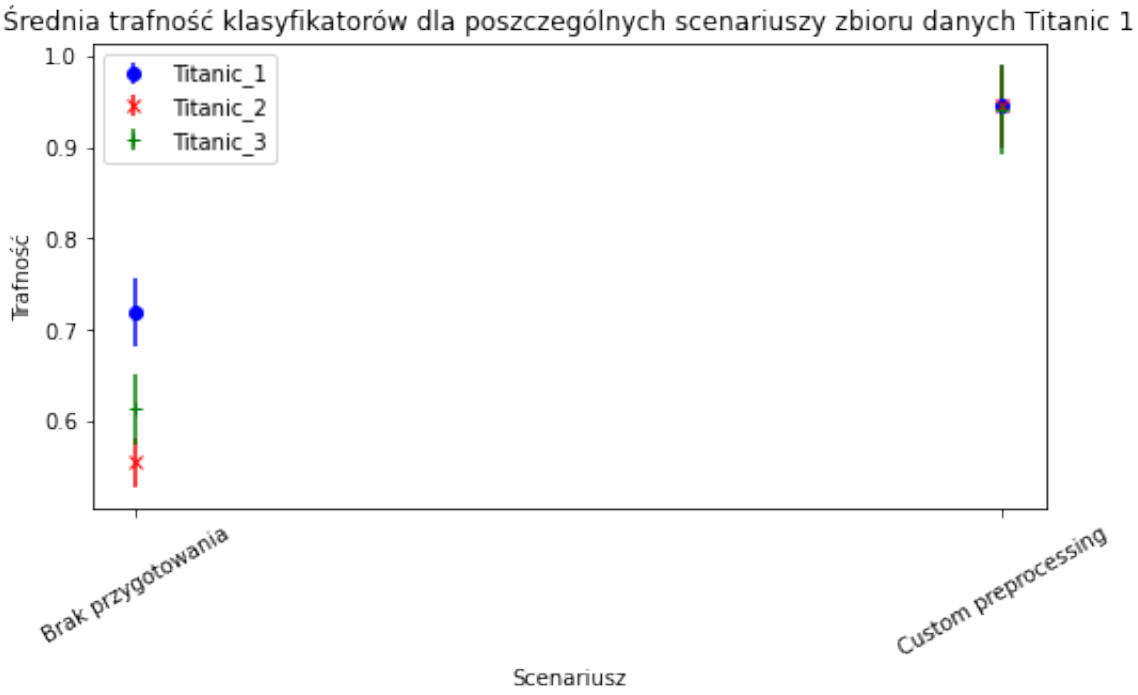


Rysunek 5.21: Średnia trafność klasyfikatorów dla zbioru danych Australian Rain Forecast dla kodowania

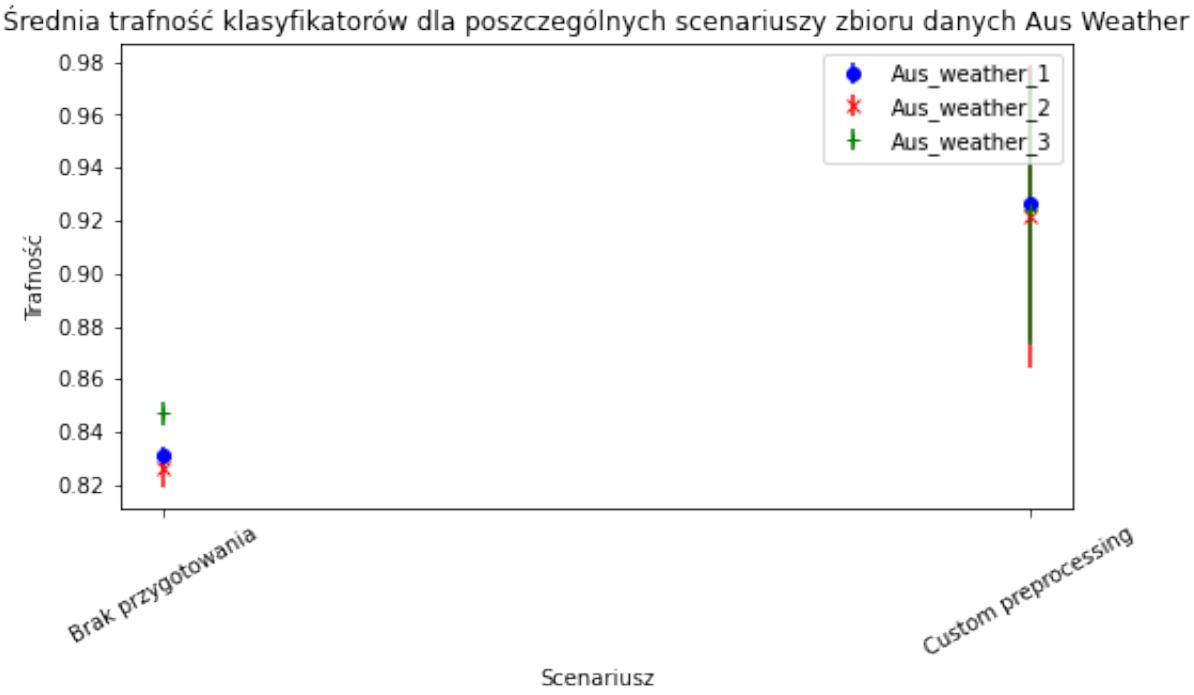
5.4.4 Indywidualne podejście

Dla każdego z trzech zbiorów danych podejścia oparte na indywidualnym podejściu do zbioru danych dają wymierną poprawę średniej trafności (Rysunek 5.48, Rysunek 5.47, Rysunek 5.46)

Rysunek 5.22: Średnia trafność klasyfikatorów dla zbioru danych Lol Stats dla indywidualnego podejścia dla zbioru



Rysunek 5.23: Średnia trafność klasyfikatorów dla zbioru danych Titanic dla indywidualnego podejścia dla zbioru



Rysunek 5.24: Średnia trafność klasyfikatorów dla zbioru danych Australian Rain Forecast dla indywidualnego podejścia dla zbioru

Rozdział 6

Podsumowanie

W poniższym rozdziale przedstawiono, jak wyniki eksperymentów mają się do założeń postawionych przed ich przeprowadzeniem, oraz zapisano wnioski na podstawie wyników.

6.1 Konieczność przygotowania danych

Tablica 6.1: Trafność klasyfikatorów bez przygotowania danych

Wariant	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów
Lol Stats 1	0,5500000	0,4250000	0,3750000
Lol Stats 2	0,6000000	0,4500000	0,4750000
Lol Stats 3	0,5500000	0,5000000	0,5500000
Australian Rain Forecast 1	0,8364865	0,8216216	0,8344595
Australian Rain Forecast 2	0,8418919	0,8128378	0,8243243
Australian Rain Forecast 3	0,8581081	0,8405405	0,8432432
Titanic 1	0,7894737	0,7368421	0,6315789
Titanic 2	0,5555556	0,6111111	0,5000000
Titanic 3	0,6315789	0,5263158	0,6842105

W zależności od tego, jaki klasyfikator wybierzemy, bez przygotowania danych możemy uzyskać wyniki zarówno porównywalne z losowym wyborem klasy, ale i takie, które w wielu przypadkach mogły by być uznać za zadowalające (Tablica 6.1). Równocześnie musimy pamiętać, że w dziedzinach takich jak medycyna, czy bankowość chcielibyśmy, aby wytrenowane klasyfikatory były jak najbliższe

perfekcji. Z tego powodu cenną informacją jest to, jak bardzo możemy poprawić trafność klasyfikatorów dzięki przygotowaniu danych.

6.2 Wpływ źle dobranych sposobów przygotowania danych

Tablica 6.2: Przykładowe scenariusze dla każdego zbioru danych, w któych przygotowanie danych wpłynęło negatywnie na trafność

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów
Australian Rain Forecast 1	Brak przygotowania	0,8364865	0,8216216	0,8344595
	Kodowanie wartości kateg. i wypełnianie war. brak. średnią	0,8272446	0,8068111	0,8123839
Lol Stats 2	Brak przygotowania	0,6000000	0,4500000	0,4750000
	Wypełnienie maksimum	0,5918367	0,4897959	0,4897959
Titanic 1	Brak przygotowania	0,7894737	0,7368421	0,6315789
	Kodowanie wartości kategorycznych	0,6093750	0,6406250	0,5625000

Zgodnie z hipotezą, przygotowanie danych w każdym przypadku powinno skutkować lepszymi rezultatami klasyfikacji, jednak podczas eksperymentów zauważono przypadki, w których przygotowanie danych pogorszyło jakość klasyfikacji (Tablica 6.2). Jednoznacznie pokazuje to, że przeprowadzenie przygotowania danych bez przeanalizowania zbioru i dobrania odpowiednich metod do danego przypadku, może skutkować pogorszeniem skuteczności klasyfikacji.

6.3 Najlepsze wyniki po przygotowaniu danych

Tablica 6.3: Scenariusze dla zbioru Titanic, w któych przygotowanie danych wpłynęło najlepiej na trafność

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów
Titanic 1	Brak przygotowania	0,7894737	0,7368421	0,6315789
	Kodowanie wartości kategorycznych	1,0000000	1,0000000	0,6888889
	Przygotowanie dostosowane do zbioru	1,0000000	1,0000000	0,8333333
Titanic 2	Brak przygotowania	0,5555556	0,6111111	0,5000000
	Kodowanie wartości kategorycznych i wypełnianie warto±ci brakuj¡cych średnią	1,0000000	1,0000000	0,7000000
	Przygotowanie dostosowane do zbioru	1,0000000	1,0000000	0,8333333
Titanic 3	Brak przygotowania	0,6315789	0,5263158	0,6842105
	Kodowanie wartości kategorycznych	1,0000000	1,0000000	0,6666667
	Przygotowanie dostosowane do zbioru	1,0000000	1,0000000	0,8222222

Dla zbioru Titanic po przygotowaniu dwa klasyfikatory osiągają 100% trafności, natomiast w przypadku trzeciego możemy zauważyć znaczną poprawę trafności (Tablica 6.3). Przed przygotowaniem w poszczególnych przypadkach można było zauważyć, że klasyfikator radzi sobie niewiele lepiej niż dobierając klasyfikację losowo (50%), natomiast po przygotowaniu dostosowanym do danego zbioru, trafność klasyfikacji jest wysoka (80%), bądź nawet perfekcyjna (100%).

Tablica 6.4: Scenariusze dla zbioru Lol Stats, w któych przygotowanie danych wpłynęło najlepiej na trafność

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów
Lol Stats 1	Brak przygotowania	0,5500000	0,4250000	0,3750000
	Kodowanie wartości kategorycznych	1,0000000	0,8250000	0,4500000
	Przygotowanie dostosowane do zbioru	1,0000000	0,6739130	0,9130435
Lol Stats 2	Brak przygotowania	0,6000000	0,4500000	0,4750000
	Kodowanie wartości kategorycznych i wypełnianie warto±ci brakuj¡cych średnią	0,8418919	0,8128378	0,8243243
	Przygotowanie dostosowane do zbioru	1,0000000	0,8000000	0,9555556
Lol Stats 3	Brak przygotowania	0,5500000	0,5000000	0,5500000
	Kodowanie wartości kategorycznych	1,0000000	0,8250000	0,5250000
	Przygotowanie dostosowane do zbioru	1,0000000	0,6000000	0,9333333

Powyżej przedstawiono najlepsze wyniki klasyfikacji dla zbioru Lol Stats (Tablica 6.4) Dla klasyfikatora Random Forest najlepsze wyniki uzyskano przy kodowaniu wartości kategorycznych (około 90%), dla xgBoost zarówno kodowanie jak i indywidualne przygotowanie skutkują idealną klasyfikacją. Dla k-najbliższych sąsiadów najlepsze wyniki uzyskano przy indywidualnym podejściu do zbioru (powyżej 90%)

Tablica 6.5: Scenariusze dla zbioru Australian Rain Forecast, w któych przygotowanie danych wpłynęło najlepiej na trafność

Wariant	Scenariusz	Trafność xgBoost	Trafność Random Forest	Trafność k-najbliższych sąsiadów
Australian Rain Forecast 1	Brak przygotowania	0,8364865	0,8216216	0,8344595
	Kodowanie wartości kategorycznych	1,0000000	0,9547297	0,8391892
	Przygotowanie dostosowane do zbioru	1,0000000	0,9796296	0,8006173
Australian Rain Forecast 2	Brak przygotowania	0,8418919	0,8128378	0,8243243
	Kodowanie wartości kategorycznych	1,0000000	0,9500000	0,8304054
	Przygotowanie dostosowane do zbioru	1,0000000	0,9833436	0,7809994
Australian Rain Forecast 3	Brak przygotowania	0,8581081	0,8405405	0,8432432
	Kodowanie wartości kategorycznych	1,0000000	0,9635135	0,8506757
	Przygotowanie dostosowane do zbioru	1,0000000	0,9722736	0,7997535

Częścią przygotowania dostosowanego do zbioru Australian Rain Forecast, było rozbicie kolumny z datą na trzy kolumny, zawierające dzień, miesiąc oraz rok, co skutkowało pogorszeniem trafności klasyfikacji dla k-najbliższych sąsiadów (Tablica 6.5). Dla pozostałych klasyfikatorów przygotowanie dostosowane do zbioru pozwoliło uzyskać najlepszy, bądź jeden z najlepszych wyników. Należy zwrócić uwagę na fakt, że różnice w trafności między scenariuszami bez oraz z przygotowaniem danych, nie były tak znaczne jak dla pozostałych zbiorów.

Tabelka z przykładami

6.4 Potencjalne dalsze kierunki badań

Z wyników eksperymentów wynika korelacja między wielkością zbioru, a znaczeniem przygotowania danych. Przy niewielkich zbiorach dzięki przygotowaniu danych jesteśmy w stanie uzyskać znacznie lepsze wyniki. Natomiast przy większym zbiorze uzupełnianie braków może prowadzić do przekłamań, a co za tym idzie mniejszej skuteczności klasyfikatora (Tablica 6.4). Stąd zasadnym były by dalsze badania, prowadzone pod kątem zależności między wielkością zbioru, a znaczeniem przygotowania danych.

6.5 Najlepsze praktyki przygotowania danych

Z przeprowadzonych eksperymentów wynika, że odpowiednie przygotowanie danych zwiększa w umiarkowanym stopniu celność klasyfikatorów. Jeżeli przed analizą danych dogłębnie poznamy i zrozumiemy zbiór danych, będziemy mogli wybrać najbardziej odpowiednie metody przygotowania danych dla danego zbioru, a co za tym idzie wyniki analizy będą najbardziej trafne. Należy jednak zwrócić uwagę, że nie wszystkie scenariusze skutkowały jednoznaczną poprawą rezultatów, dlatego dobrą praktyką jest wieloktorne przygotowywanie danych na różne sposoby tak, aby znaleźć taki, który zwraca najlepsze rezultaty Na szczególne wyróżenienie zasługuje kodowanie wartości kategorycznych na liczbowe, gdyż w każdym przypadku zastosowanie tej metody znacznie zwiększyło trafność klasyfikacji

Bibliografia

[1] Zbiór danych Titanic dostępny do pobrania ze strony kaggle.com

[2] Zbiór danych Lol Stats dostępny do pobrania ze strony kaggle.com

[3] Zbiór danych Australian Rain dostępny do pobrania ze strony kaggle.com

[4] "Transforming Unstructured Data into Useful Information", *Big Data, Mining, and Analytics*, Auerbach Publications, pp. 227–246, 2014-03-12, doi:10.1201/b16666-14, ISBN 978-0-429-09529-0

[5] MOIS, George; FOLEA, Silviu; SANISLAV, Teodora. Analysis of three IoT-based wireless sensors for environmental monitoring. *IEEE Transactions on Instrumentation and Measurement*, 2017, 66.8: 2056-2064.

[6] Margo, Robert A. (2000). *Wages and labor markets in the United States, 1820-1860.* University of Chicago Press. ISBN 0-226-50507-3. OCLC 41285104

[7] Marshall, G. (2005). The purpose, design and administration of a questionnaire for data collection. Radiography, 11(2), 131-136.

[8] Fabijan, A., Olsson, H. H., Bosch, J. (2015). Customer feedback and data collection techniques in software R&D: a literature review. In Software Business: 6th International Conference, ICSOB 2015, Braga, Portugal, June 10-12, 2015, Proceedings 6 (pp. 139-153). Springer International Publishing.

[9] Hasan, M. K., Alam, M. A., Roy, S., Dutta, A., Jawad, M. T., Das, S. (2021). Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). Informatics in Medicine Unlocked, 27, 100799.

[10] E. Kreyszig (1979). Advanced Engineering Mathematics (4th ed.). Wiley. p. 880, eq. 5. ISBN 0-471-02140-7.

[11] Okada, S., Ohzeki, M., Taguchi, S. (2019). Efficient partition of integer optimization problems with one-hot encoding. Scientific reports, 9(1), 13036.

[12] Wang, H., Bah, M. J., Hammad, M. (2019). Progress in outlier detection techniques: A survey. Ieee Access, 7, 107964-108000.

[13] Alghushairy, O., Alsini, R., Soule, T., Ma, X. (2020). A review of local outlier factor algorithms for outlier detection in big data streams. Big Data and Cognitive Computing, 5(1), 1.

[14] Potdar, K., Pardawala, T. S., Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. International journal of computer applications, 175(4), 7-9.

[15] Narudin, F. A., Feizollah, A., Anuar, N. B., Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. Soft Computing, 20, 343-357.

[16] Khanal, S. S., Prasad, P. W. C., Alsadoon, A., Maag, A. (2020). A systematic review: machine learning based recommendation systems for e-learning. Education and Information Technologies, 25, 2635-2664.

[17] Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence, 110, 104743.

[18] Sousa, M. J., Pesqueira, A. M., Lemos, C., Sousa, M., & Rocha, Á. (2019). Decision-making based on big data analytics for people management in healthcare organizations. Journal of medical systems, 43, 1-10.

[19] Schildkamp, K., Lai, M. K., & Earl, L. (Eds.). (2012). Data-based decision making in education: Challenges and opportunities.

[20] Fávero, L. P., & Belfiore, P. (2019). Data science for business and decision making. Academic Press.

[21] Definicja dostępna w dokumentacji biblioteki XGBoost

[22] Cover, Thomas M.; Hart, Peter E. (1967). Ńearest neighbor pattern classification"(PDF). IEEE Transactions on Information Theory. 13 (1): 21–27. CiteSeerX 10.1.1.68.2616. doi:10.1109/TIT.1967.1053964.

[23] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

[24] Artykuł na temat jakości danych na stronie firmy IBM