Przetwarzanie danych dotyczących bezpieczeństwa drogowego w Wielkiej Brytanii w 2016 roku (UK 2016 Road Safety)

Bojarski Bartosz
bartosz.bojarski@student.pk.edu.pl

Kandratsiuk Anastasiya
a.kandratsiuk@student.pk.edu.pl

1 czerwca 2024

1 Abstract

Bezpieczeństwo drogowe jest kluczowym zagadnieniem dla zarządzania transportem i planowania infrastruktury. Projekt ten koncentruje się na przetwarzaniu i analizie danych dotyczących bezpieczeństwa drogowego w Wielkiej Brytanii w 2016 roku, mając na celu identyfikację czynników wpływających na liczbę i ciężkość wypadków drogowych. Wykorzystując metody odkrywania wiedzy z danych, takie jak eksploracja danych, analiza statystyczna oraz modele uczenia maszynowego, projekt dąży do odkrycia ukrytych wzorców i zależności.

Celem analizy jest zrozumienie, jakie czynniki (np. warunki pogodowe, pora dnia, stan techniczny pojazdów) mają największy wpływ na występowanie wypadków, oraz opracowanie modeli predykcyjnych, które mogą przewidzieć prawdopodobieństwo wystąpienia wypadków w określonych warunkach. Dane pochodzą z oficjalnych źródeł rządowych, co zapewnia ich wiarygodność i dokładność.

Wyniki projektu dostarczą istotnych informacji, które mogą posłużyć jako podstawa do podejmowania świadomych decyzji oraz wprowadzenia skutecznych działań prewencyjnych, mających na celu poprawę bezpieczeństwa na drogach. Projekt ma charakter interdyscyplinarny, łącząc elementy statystyki, informatyki oraz inżynierii transportu, co pozwala na kompleksowe podejście do analizy i rozwiązania problemów związanych z bezpieczeństwem drogowym.

2 Wprowadzenie

Przetwarzanie danych dotyczących bezpieczeństwa drogowego w Wielkiej Brytanii w 2016 roku

Bezpieczeństwo drogowe stanowi jedno z kluczowych zagadnień w kontekście zarządzania transportem oraz planowania infrastruktury drogowej. W 2016 roku, w Wielkiej Brytanii, zarejestrowano wiele zdarzeń drogowych, które dostarczają cennych danych do analiz mających na celu poprawę bezpieczeństwa na drogach.

Celem niniejszego projektu jest przetworzenie i analiza danych dotyczących bezpieczeństwa drogowego w Wielkiej Brytanii za rok 2016. Projekt ten ma na celu zidentyfikowanie kluczowych czynników wpływających na liczbę i ciężkość wypadków drogowych, a także opracowanie rekomendacji, które mogą przyczynić się do zmniejszenia liczby wypadków i poprawy bezpieczeństwa użytkowników dróg.

Wprowadzenie metod odkrywania wiedzy z danych (data mining) umożliwia skuteczne przetwarzanie dużych zbiorów danych i odkrywanie ukrytych wzorców oraz zależności. W kontekście analizy bezpieczeństwa drogowego, metody te pozwalają na:

- Identyfikację czynników ryzyka - zrozumienie, jakie czynniki (np. warunki pogodowe, pora dnia, stan techniczny pojazdów) mają największy wpływ na występowanie wypadków drogowych.
- Segmentację danych - podział danych na grupy w

celu zidentyfikowania specyficznych wzorców wypadków, które mogą być charakterystyczne dla różnych typów dróg, pojazdów czy użytkowników.

- Predykcję zdarzeń - tworzenie modeli predykcyjnych, które mogą przewidzieć prawdopodobieństwo wystąpienia wypadków w określonych warunkach.

Analiza danych dotyczących bezpieczeństwa drogowego nie tylko dostarcza istotnych informacji na temat bieżącego stanu bezpieczeństwa na drogach, ale także wspiera tworzenie strategii prewencyjnych oraz polityk publicznych ukierunkowanych na zmniejszenie liczby wypadków. W projekcie wykorzystane zostaną różnorodne techniki analityczne, takie jak eksploracja danych, analiza statystyczna oraz modele uczenia maszynowego, aby uzyskać kompleksowy obraz sytuacji bezpieczeństwa drogowego w Wielkiej Brytanii w 2016 roku.

3 Zbiór danych

Zbiór danych zawiera informacje o wszystkich wypadkach drogowych w Wielkiej Brytani z roku 2016. Dane zostały przygotowane przez Departament Transportu Wielkiej Brytanii i przez inicjatywę Open Gov. Oryginalne dane są podzielone na cztery pliki, które opisują parametry dotyczące ofiar, pojazdów, czy okoliczności wypadków. Zostały one połączone w dwa zbiory danych, jeden opisujący geolokację wypadków, a drugi zawierający wszystkie pozostałe informacje.

Część kolumn jest przygotowana w formie kodów, które wymagają przetłumaczenia na zrozumiałe wartości. Będzie to robione przy pomocy słowników, które zostały dostarczone wraz z danymi, a sama translacja będzie wykonywana w trakcie analizy danych.

Dane UK 2016 Road Safety Data: [2] `https://www.kaggle.com/datasets/bluehorseshoe/uk-2016-road-safety-data/`

4 Motywacje

Projekt przetwarzania danych dotyczących bezpieczeństwa drogowego w Wielkiej Brytanii w 2016 roku jest niezwykle interesujący i ma znaczący potencjał wkładu w społeczeństwo oraz przemysł z kilku kluczowych powodów:

1. Poprawa bezpieczeństwa publicznego: Bezpieczeństwo na drogach jest priorytetem dla każdego społeczeństwa. Analiza danych dotyczących wypadków drogowych pozwala zidentyfikować kluczowe czynniki ryzyka oraz wypracować skuteczne strategie prewencyjne. W rezultacie, można zmniejszyć liczbę wypadków, co bezpośrednio przekłada się na mniejszą liczbę ofiar śmiertelnych i rannych, a także na poprawę jakości życia obywateli.

2. Wsparcie dla decyzji rządowych: Wyniki tego projektu dostarczą cennych informacji, które mogą być wykorzystane przez organy rządowe do tworzenia polityk i przepisów mających na celu poprawę bezpieczeństwa drogowego. Dzięki temu możliwe jest podejmowanie świadomych decyzji opartych na solidnych danych, co zwiększa skuteczność wprowadzanych działań.

3. Optymalizacja infrastruktury: Analiza danych o wypadkach może wskazać na problematyczne miejsca na drogach, które wymagają modernizacji lub dodatkowych środków bezpieczeństwa. Dzięki temu można efektywnie planować inwestycje infrastrukturalne, co przyczynia się do bezpieczniejszej i bardziej wydajnej sieci transportowej.

4. Edukacja i świadomość społeczna: Projekt może przyczynić się do zwiększenia świadomości na temat bezpieczeństwa drogowego wśród obywateli. Edukacja w zakresie identyfikacji ryzykownych zachowań na drodze oraz promowanie odpowiedzialnego korzystania z infrastruktury drogowej może znacząco wpłynąć na zmniejszenie liczby wypadków.

5. Interdyscyplinarne podejście: Projekt jest ważny dla naszej instytucji, ponieważ łączy w sobie elementy statystyki, informatyki, inżynierii transportu oraz nauk społecznych. Tego rodzaju

interdyscyplinarne podejście sprzyja innowacyjności i pozwala na bardziej kompleksowe rozwiązywanie problemów.

Podsumowując, projekt ten jest nie tylko interesujący ze względu na swoje naukowe i technologiczne aspekty, ale również ma ogromny potencjał do pozytywnego wpływu na społeczeństwo i przemysł.

5 Ewaluacja

W ramach tego projektu będziemy chcieli móc z dużą dokładnością móc określić na bazie okoliczności wypadku, w jakim stanie mogą być jego ofiary. Pewność prawdziwych odpowiedzi na poziomie 90% byłaby zadowalająca, ale dalej niedostateczna, by móc system wprowadzić do użycia, ponieważ dane dotyczą ludzkiego życia, potrzebna jest większa dokładność, która może nie być technicznie możliwa do osiągnięcia.

6 Zasoby

Projekt wykorzystuje zestaw narzędzi programistycznych w języku Python, włączając popularne biblioteki do analizy danych, wizualizacji, oraz implementacji modeli uczenia maszynowego. Wśród tych bibliotek znajdują się Pandas do manipulacji danymi, NumPy do operacji matematycznych, Matplotlib i Seaborn do tworzenia wykresów, a także scikit-learn do implementacji różnorodnych modeli klasyfikacji i metryk oceny. Dodatkowo, projekt korzysta z bibliotek LightGBM, XGBoost oraz CatBoost dla efektywnego trenowania modeli gradient boosting, oraz biblioteki Cartopy do wizualizacji danych geograficznych.

Projekt będzie głównie prowadzony w interaktywnym środowisku Jupyter Notebook, umożliwiającym prowadzenie analiz danych w sposób iteracyjny oraz prezentację wyników w formie czytelnej i zrozumiałej. Dodatkowo, do rozwoju kodu używane było środowisko programistyczne PyCharm, które oferuje funkcje wspomagające debugowanie, zarządzanie kodem i integrację z systemami kontroli wersji. Środowiska IDE zapewniają komfortową i efektywną pracę nad projektem, umożliwiającą szybkie prototypowanie, testowanie i wdrażanie rozwiązań.

7 Eksperyment

7.1 Preprocessing

7.1.1 Prezentacja podstawowych danych zbioru danych

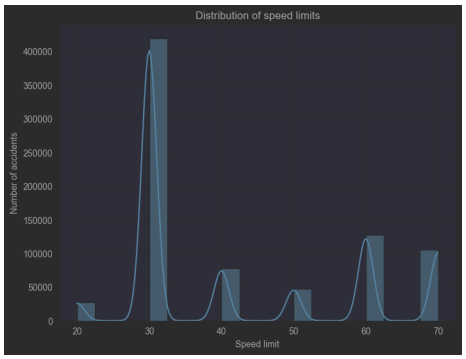
Zbiór danych posiada 92 kolumny i 804853 wiersze. Kluczem głównym dla zbioru danych jest kolumna 'Accident Index'. Kolejne cztery kolumny zawierają informacje o geolokalizacji wypadków. Pozostałe kolumny zawierają informacje o wypadkach, ofiarach, pojazdach, czy okolicznościach wypadków.

7.1.2 Sprawdzenie brakujących danych

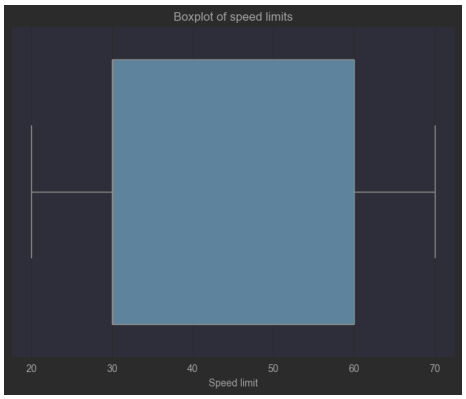
W zbiorze danych znajdują się brakujące dane w niektórych kolumnach. By spełnić wymagania projektu, będziemy dodawać brakujące dane w kolumnach, które będą analizowane, by osiągnąć poziom około 10% brakujących danych. W chwili obecnej najwięcej brakujących danych mamy w przypadku informacji o modelu auta (około 15% brakujących danych), a następnie informacje o jednostce geograficznej w której doszło do wypadku (około 5% brakujących danych). Jednak ta metoda sprawdzania wartości brakujących jest niewystarczająca, ponieważ w zakodowanych kolumnach wartość -1 oznacza brakujące dane.

7.1.3 Przykładowa analiza dla prędkości limitów

Dodanie wykresu histogramu (Rysunek 1) i pudełkowego (Rysunek 2) dla prędkości limitów umożliwia nam lepsze zrozumienie rozkładu prędkości w danych dotyczących wypadków drogowych. Histogram prezentuje nam dystrybucję prędkości limitów na drogach, co pozwala zobaczyć, w jakich przedziałach prędkości występuje najwięcej wypadków (z wykresu wyżej widać że jest to 30 mil na godzinę). Z kolei wykres pudełkowy pozwala nam zidentyfikować wartości odstające oraz zakres prędkości, w którym znajduje się większość obserwacji.



Rysunek 1: Wykres histogramu dla prędkości limitów



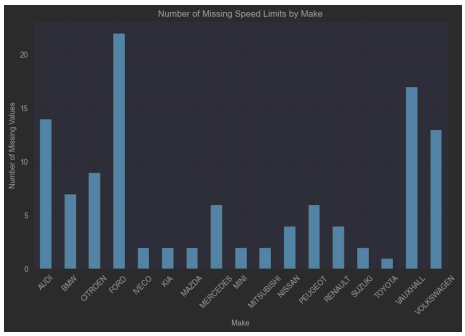
Rysunek 2: Wykres pudełkowy dla prędkości limitów

Na podstawie danych dotyczących prędkości limitów można wywnioskować:

1. Średnia prędkość limitu wynosi około 41.79 mil na godzinę.
2. Odchylenie standardowe wynoszące około 15.80 sugeruje, że rozrzut prędkości limitów między wypadkami był stosunkowo niewielki w porównaniu do średniej wartości.
3. Wartości kwartyla 25% i 50% są identyczne i wynoszą 30 mil na godzinę, co oznacza, że większość wypadków miała miejsce na obszarach o niższej prędkości limitu.
4. Wartość maksymalna prędkości limitu wynosi 70 mil na godzinę. Występowanie przypadków wypadków na obszarach o wyższych prędkościach limitów może wskazywać na potencjalnie większe ryzyko dla bezpieczeństwa drogowego.

7.1.4 Analiza prędkości limitów według marki auta

Analiza danych brakujących dotyczących prędkości pojazdów dla różnych marek samochodów pokazuje, że marka Ford, Vauxhall, Audi oraz Volkswagen są szczególnie narażone na braki danych. To może sugerować, że dla tych konkretnych marek istnieje większe ryzyko braku rejestracji prędkości pojazdów w przypadku wypadków drogowych.



Rysunek 3: Wykres histogramu dla prędkości limitów według marki auta

7.1.5 Histogramy dla wybranych kolumn

Jak widać z powyższych wyreksów (Rysunek 4), większość wypadków drogowych w Wielkiej Brytanii odbywa się przy ograniczeniu prędkości do 30 mil na godzinę. Najwięcej wypadków drogowych ma miejsce w sobotę i piątek, a najmniej w poniedziałek. Największy odsetek kierowców powodujących wypadki to osoby w wieku 25-35 lat, choć dla ponad 10% wypadków wiek kierowcy nie jest znany. Analogicznie, najwięcej ofiar wypadków to osoby w wieku 25-35 lat. Można też wyczytać, że na większość wypadków drogowych przypada do 1 lub 2 pojazdy, a liczba ofiar wypadków zazwyczaj nie przekracza 5. Dodatkowo,

najwięcej kolizji na drodze ma miejsce na drogach dwukierunkowych, jednopasmowych.

7.1.6 Wykres pudełkowy

Wykres pudełkowy (Rysunek 5) pokazuje nam rozkład i rozproszenie danych dla poszczególnych cech. Można zauważyć wartości odstające dla następujących cech: 1st Road Number, 2nd Road Number, Vehicle Propulsion Code, Engine Capacity.

7.1.7 Wykresy typu Boxplot

Rozpatrując wiek ofiar wypadków (Age_of_Casualty) na rysunku 6 można zauważyć, że mediana jest około połowy zakresu międzykwartylowego, z niektórymi wartościami odstającymi w górnej części zakresu. Dla klasy wypadków (Casualty_Class) można podkreślić, że większość danych koncentruje się w dolnej części zakresu międzykwartylowego, z niewielkim rozproszeniem i brakiem wartości odstających. Płeć ofiary wypadków (Sex_of_Casualty) może sugerować nam o niewielkiej różnice między mężczyznami a kobietami w kontekście wypadków drogowych, z podobnymi medianami dla obu grup. Na wykresie są widoczne dane odstające dla większości cech w tym dane brakujące oznaczone jako -1 za wyjątkiem klasy wypadków (Casualty_Class) oraz Casualty_IMD_Decile.

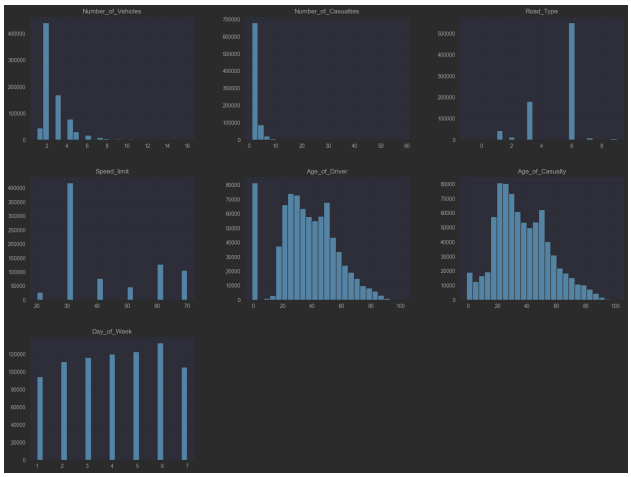
W analizie rysunku 7, rozpatrując wiek pojazdu (Age_of_Vehicle) może się wydawać że rozproszenie danych jest głównie w niższym wieku pojazdów, co sugeruje, że większość pojazdów wypadkowych jest stosunkowo młoda. Warto zwrócić uwagę na cechę 'Was_Vehicle_Left_Hand_Drive', gdzie większość danych jest skoncentrowana w określonej kategorii, co może wskazywać na przewagę określonego typu układu kierowniczego. Dla 'Vehicle_Type' można zauważyć znaczną różnorodność w rozproszeniu danych, co sugeruje, że wypadki dotykają różnych typów pojazdów w różnym stopniu. Cecha 'Towing_and_Articulation' wydaje się mieć niewielkie rozproszenie, co sugeruje, że większość danych koncentruje się wokół określonej kategorii lub kategorii. Na wykresie są widoczne dane odstające dla większości cech w tym dane brakujące oznaczone jako -1 za wyjątkiem Vehicle_Manoeuvre, Junction_Location, 1st_Point_of_Impact, Journey_Purpose_of_Driver, Driver_IMD_Decile oraz Vehicle_IMD_Decile.

Analiza danych z rysunku 8 sugeruje, że rozkład wieku pojazdów (Age_of_Vehicle) jest skoncentrowany w okolicach średniej wartości, co sugeruje, że większość pojazdów ma przeciętny wiek. Wartości Engine_Capacity_(CC) są rozproszone na całym zakresie, co wskazuje na różnorodność pojemności silnika w badanych pojazdach. Istnieje zauważalne rozproszenie wartości w różnych grupach wiekowych kierowców (Age_Band_of_Driver_y), co sugeruje, że wypadki występują w różnych grupach wiekowych. Na wykresie są widoczne dane odstające dla większości cech w tym dane brakujące oznaczone jako -1 za wyjątkiem Vehicle_Manoeuvre, Junction_Location, 1st_Point_of_Impact, Journey_Purpose_of_Driver, Driver_IMD_Decile oraz Vehicle_IMD_Decile.

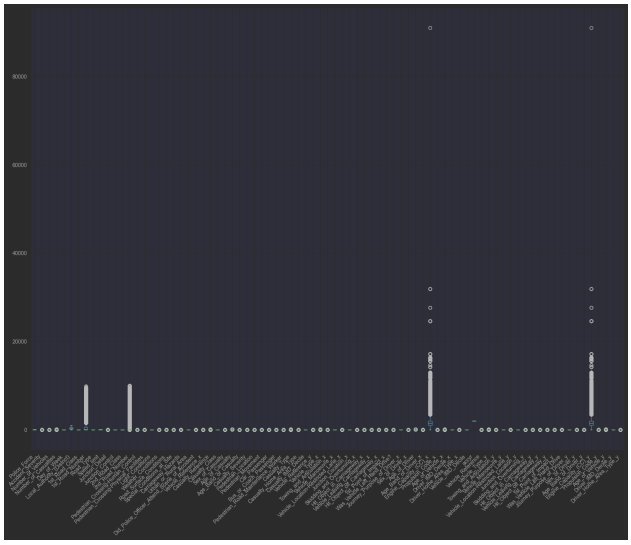
Na rysunku 9 widać, że większość wypadków miała niski poziom powagi (Accident_Severity), z czego można wywnioskować, że większość wypadków drogowych nie prowadziła do poważnych konsekwencji. Rozkład liczby pojazdów biorących udział w wypadkach (Number_of_Vehicles) jest skoncentrowany wokół niższych wartości, ale istnieją również pojedyncze przypadki z dużą liczbą pojazdów, co może wskazywać na zróżnicowanie sytuacji wypadków. Widać rozproszenie wartości liczby ofiar (Number_of_Casualties), co sugeruje, że wypadki mogą mieć różne skutki w postaci rannych lub zabitych osób.

7.1.8 Wykresy dotyczące groźności wypadków

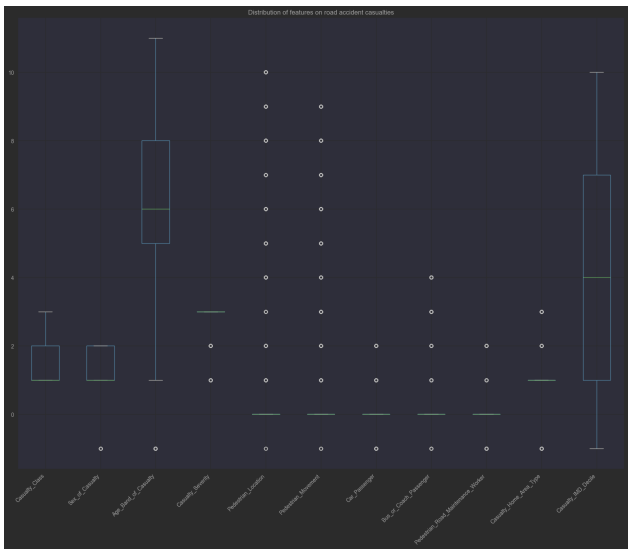
Zależność między liczbą ofiary wypadków, a ich groźnością jest jednym z najważniejszych parametrów analizy wypadków komunikacyjnych. Kluczowe jest zidentyfikowanie w jakich wypadkach dochodzi do najgroźniejszych obrażeń i pozwoli to na dalszą analizę pod kątem przyczyn takich zjawisk. Można dzięki nim wyodrębnić obszary, które wymagają poprawy, oraz zabezpieczyć tereny, które stawały się czarnymi



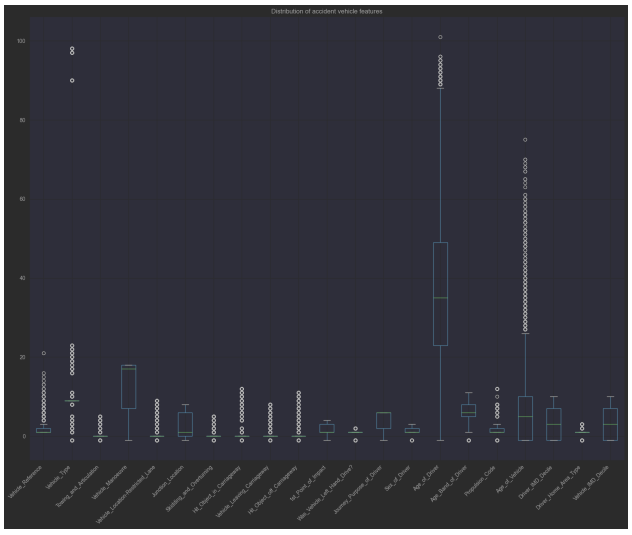
Rysunek 4: Histogramy dla wybranych kolumn



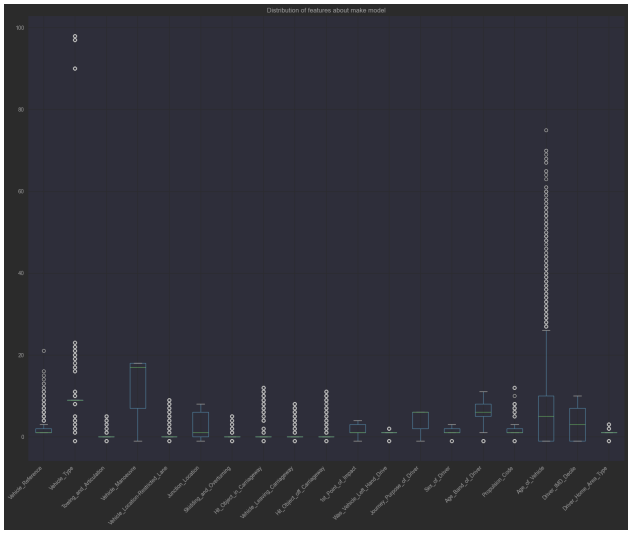
Rysunek 5: Wykres pudełkowy



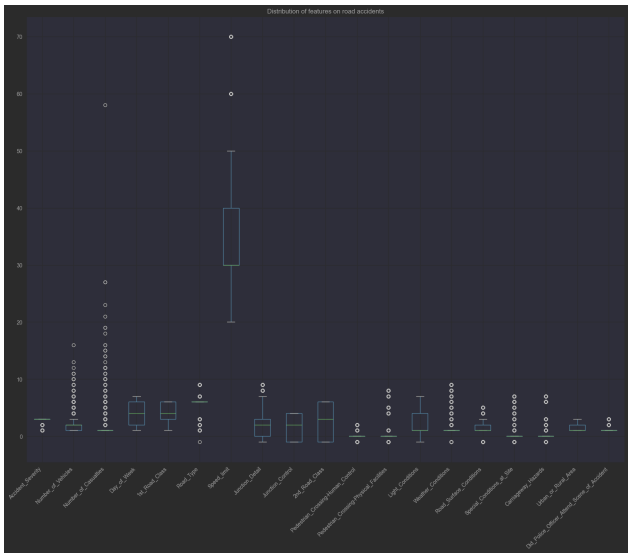
Rysunek 6: Wykres typu Boxplot 1



Rysunek 7: Wykres typu Boxplot 2



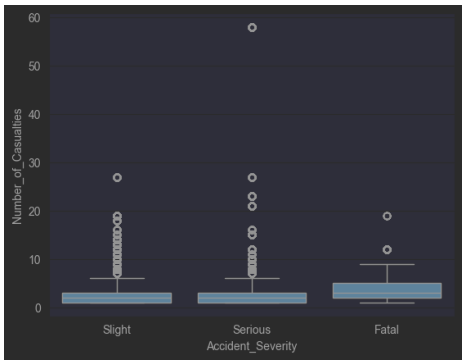
Rysunek 8: Wykres typu Boxplot 3



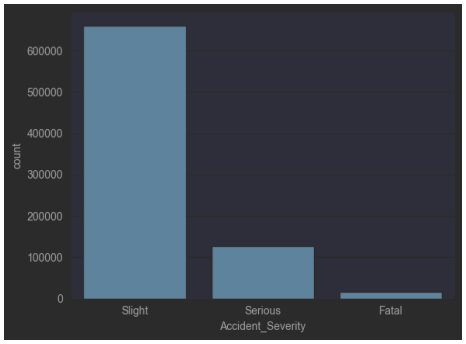
Rysunek 9: Wykres typu Boxplot 4

punktami na mapach drogowych.

Poniżej przedstawione są histogramy liczności wypadków o konkrentym stopniu szkodliwości (Rysunek 10) oraz wykresy pudełkowe (Rysunek 11), prezentujące wyżej opisaną zależność. Takie wykresy mogą służyć jako wstęp do dalszej selekcji danych, które będą analizowane pod kątem przyczyn wypadków, oraz do lepszego podzielenia danych na zbiory testowe i treningowe.



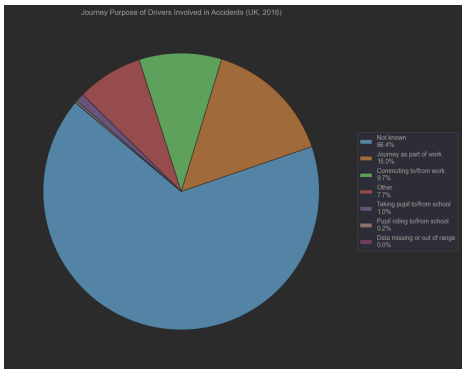
Rysunek 10: Wykres pudełkowy liczności wypadków o konkrentym stopniu szkodliwości



Rysunek 11: Histogram liczności wypadków o konkrentym stopniu szkodliwości

7.1.9 Wyodrębnienie kolumny dotyczącej powodu podróży kierowcy

Lepsze zrozumienie powodów dla których kierowcy (Rysunek 12) ruszają w drogę może pomóc zrozumieć np. dlaczego przekroczyli dozwolony limit prędkości, albo dlaczego wyprzedzali na podwójnej ciągłej.



Rysunek 12: Wykres kołowy dotyczący powodu podróży kierowcy

7.1.10 Mapa wypadków drogowych w Wielkiej Brytanii

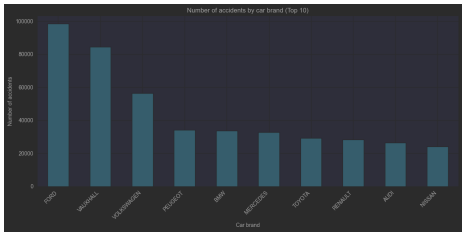
Rysunek 13 przedstawia mapę Wielkiej Brytanii z zaznaczonymi punktami reprezentującymi wypadki drogowe. Granice kraju są oznaczone różową linią, natomiast wybrzeże jest oznaczone szarą linią. Czerwone punkty na mapie reprezentują lokalizacje wypadków drogowych, gdzie każdy punkt symbolizuje jeden wypadek. Przezroczystość punktów została dostosowana, aby ułatwić zidentyfikowanie obszarów o większej gęstości wypadków. Całość ma na celu zobrazowanie rozkładu wypadków drogowych w Wielkiej Brytanii na tle geograficznych cech kraju.

7.1.11 Wykres wypadków drogowych według marek samochodów

Marki FORD, VAUXHALL i VOLKSWAGEN zajmują czołowe miejsca pod względem liczby wypadków. Można wywnioskować, że pojazdy tych marek są częściej zaangażowane w wypadki drogowe niż pojaz-

Rysunek 13: Mapa wypadków drogowych w Wielkiej Brytanii

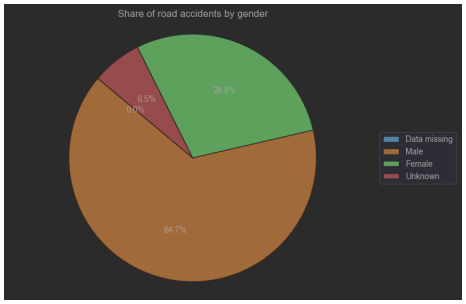
dy innych marek. Także to może sugerować, że pojazdy tych marek mogą być bardziej narażone na ryzyko wypadków lub że występujące w nich usterki lub błędy konstrukcyjne mogą przyczyniać się do większej liczby kolizji.



Rysunek 14: Wykres wypadków drogowych według marek samochodów

7.1.12 Wykres kołowy wypadków drogowych według płci

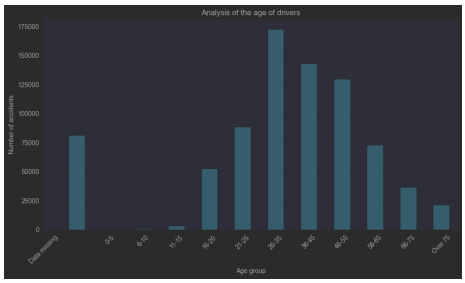
Na podstawie analizy liczby wypadków według płci można zauważyć, że większość wypadków drogowych (64.7%) dotyczy mężczyzn, podczas gdy udział kobiet w tych wypadkach wynosi 28.8%. Istnieje również kategoria o nieznanym lub nieokreślonym statusie płciowym, która stanowi 6.5% wszystkich wypadków. Ten wynik sugeruje, że mężczyźni są bardziej narażeni na wypadki drogowe niż kobiety.



Rysunek 15: Wykres kołowy wypadków drogowych według płci

7.1.13 Wykres słupkowy wypadków drogowych według wieku kierowców

Te dane sugerują, że najczęściej w wypadkach drogowych uczestniczą osoby w wieku produkcyjnym. Trzy najbardziej narażone na wypadki grupy wiekowe to osoby w przedziałach wiekowych od 26 do 35 lat, od 36 do 45 lat oraz od 46 do 55 lat.



Rysunek 16: Wykres słupkowy wypadków drogowych według wieku kierowców

7.1.14 Pairplot z wybranymi danymi

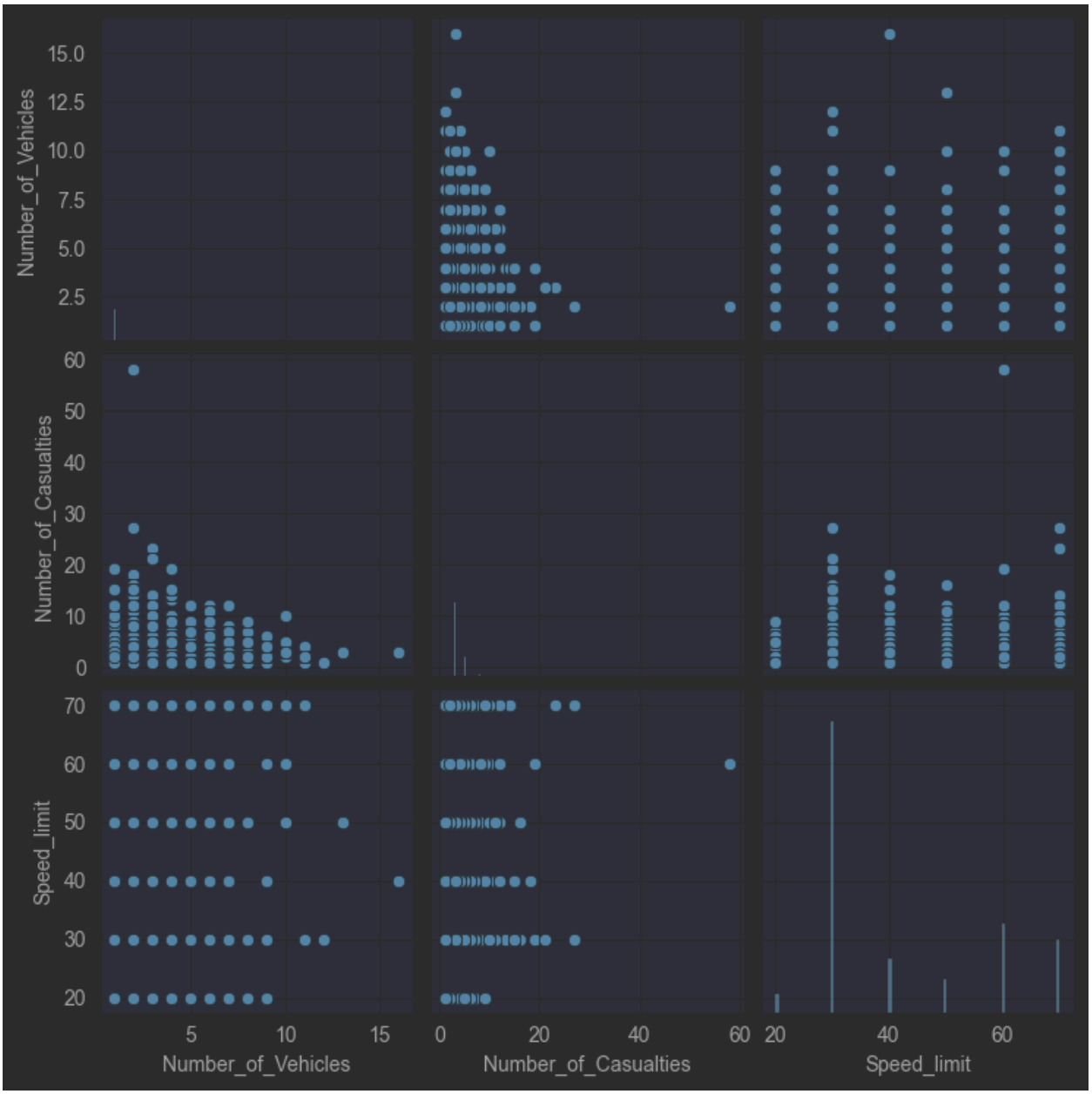
W przypadku zmiennych Number_of_Vehicles i Number_of_Casualties na rysunku 17, histogramy te sugerują, że większość wypadków ma niewielką liczbę pojazdów i ofiar. Natomiast histogram prędkości limitów sugeruje, że większość wypadków ma miejsce przy ograniczeniach prędkości na poziomie 30 lub 60 mil na godzinę. Wykresy punktowe pozwalają zauważyć potencjalne zależności między zmiennymi, takie jak np. tendencję wzrostową liczby ofiar wraz ze wzrostem liczby pojazdów.

7.1.15 Skalowanie

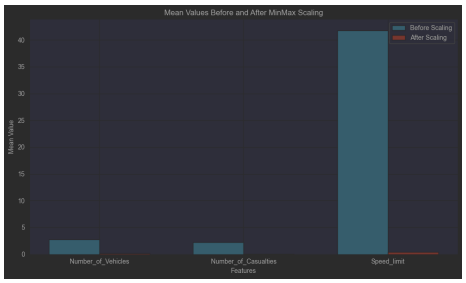
Pozwala na dostosowanie różnych cech w danych do podobnej skali, co ułatwia porównywanie ich i poprawia działanie algorytmów uczenia maszynowego.

7.1.16 MinMaxScaler

Jedna z popularnych technik skalowania danych, która przekształca cechy w zakres wartości od 0 do 1. Wartość minimalna każdej cechy jest przesunięta do 0, a wartość maksymalna jest przesunięta do 1, zachowując proporcje wartości między nimi.



Rysunek 17: Pairplot z wybranymi danymi



Rysunek 18: Średnie wartości przed i po skalowaniu MinMax

Powyższy wykres porównuje wymiary przed i po skalowaniu dla wybranych chech. Po skalowaniu jest widoczne, że wartości wahają się od 0 do 1.

Na powyższych wykresach (Rysunek 19) można wyraźnie zobaczyć, jak wartości mieszczą się w zakresie od 0 do 1 po skalowaniu i jak wyglądały przed skalowaniem.

7.1.17 StandardScaler

Inna popularna technika skalowania danych, która przekształca cechy tak, aby miały średnią równą 0 i odchylenie standardowe równą 1. Na wykresach przedstawiających dane przed i po skalowaniu standardowym (Rysunek 20) można zauważyć, że po skalowaniu wartości cech mają średnią bliską zeru. Skalowanie standardowe umożliwia sprowadzenie wartości cech do wspólnego zakresu, co może poprawić wydajność algorytmów uczenia maszynowego, które są wrażliwe na różnice w skali cech. Dzięki temu można uniknąć przewagi jednej cechy nad innymi ze względu na jej większą skalę.

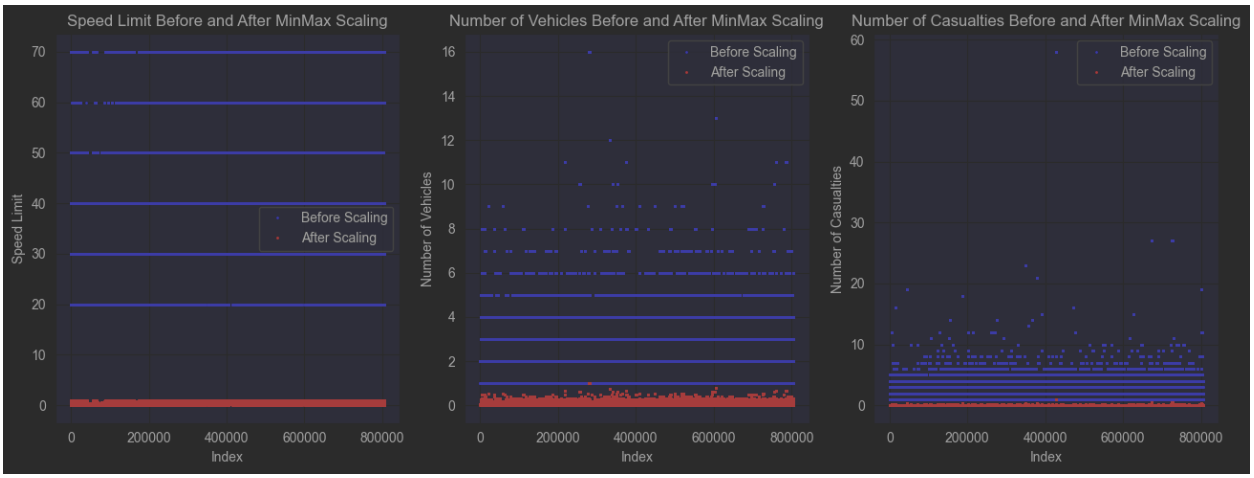
7.1.18 Macierz korelacji

Macierz korelacji przedstawiona na wykresie ciepła pokazuje, w jakim stopniu zmienne są skorelowane ze sobą. Im ciemniejszy kolor, tym większa korelacja. W analizie danych dotyczących pojazdów możemy zauważyć, że niektóre zmienne mają silną korelację dodatnią, co oznacza, że zmieniają się one w tym samym kierunku. Natomiast brak korelacji między niektórymi zmiennymi może wskazywać na ich niezależność od siebie w kontekście analizowanych danych. Na przykład zmienna “Age_Band_of_Driver” na rysunku 21 wydaje się być silnie skorelowana z “Age_Band_of_Driver_y”, co może sugerować spójność w sposobie zbierania tych danych lub istnienie pewnych wzorców wiekowych wśród kierowców różnych marek samochodów.

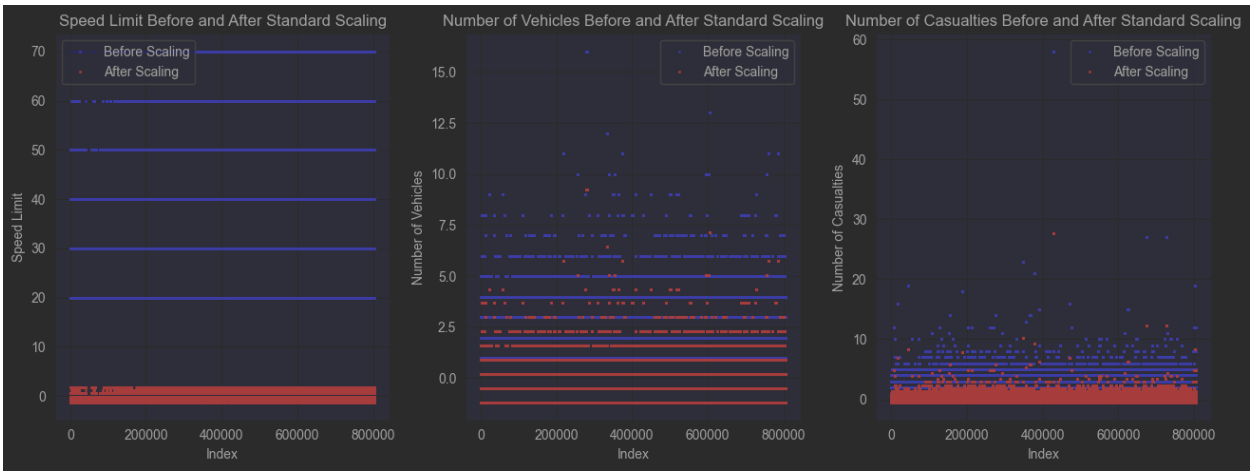
Na podstawie wykresu korelacji (Rysunek 22) widzimy korelację dodatnią o wartości 0.83 między "Pedestrian_Location" a "Pedestrian_Movement" sugeruje, że lokalizacja pieszych w stosunku do drogi jest mocno związana z ich ruchem w momencie wypadku. Natomiast korelację między "Casualty_Class" a "Pedestrian_Location" o wartości 0.73 wskazuje na związek między klasą ofiary (np. pieszy, kierowca) a jej lokalizacją w momencie wypadku. Te wyniki sugerują, że miejsce i sposób poruszania się pieszych mogą mieć istotny wpływ na rodzaj obrażeń w wypadkach drogowych.

Na podstawie wykresu korelacji dla danych dotyczących wypadków drogowych (Rysunek 23) widzimy kilka interesujących zależności. Na przykład, mamy silną korelację dodatnią 0.67 między prędkością limitu drogowego a rodzaj miejscowości, w której miał miejsce wypadek drogowy (Urban or Rural area) co może sugerować, że wypadki na obszarach miejskich lub wiejskich mogą różnić się ze względu na prędkość. Ponadto, korelacja między szczegółami skrzyżowania (junction_detail) a kontrolą skrzyżowania (junction_control) oraz 2nd_Road_Class sugeruje pewne wzajemne powiązania między tymi czynnikami, co może być istotne dla zrozumienia mechanizmu wypadków drogowych. Korelacja o wartości 0.69 między junction_detail a junction_control. Korelacja o wartości 0.72 między junction_detail a 2nd road class.Co do liczby ofiar (number_of_casualities) i liczby pojazdów (number_of_vihicles), to umiarkowana korelacja o wartości 0.25 sugeruje, że większa liczba pojazdów może prowadzić do większej liczby ofiar, co jest intuicyjne, ale potwierdza to analiza danych.

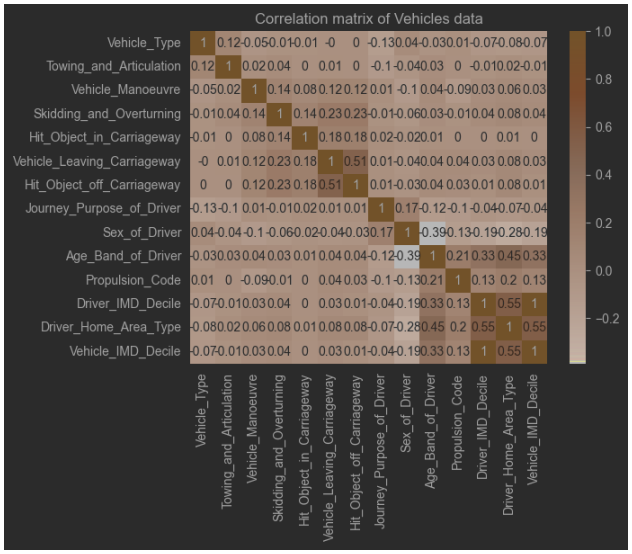
Wysoka korelacja na rysunku 24 między "Vehicle Leaving Carriageway" a "Hit Object in Carriageway" o wysokości 0.51 sugeruje, że opuszczenie pojazdu z drogi może wpływać na rodzaj obiektu, z którym pojazd koliduje. Nato-



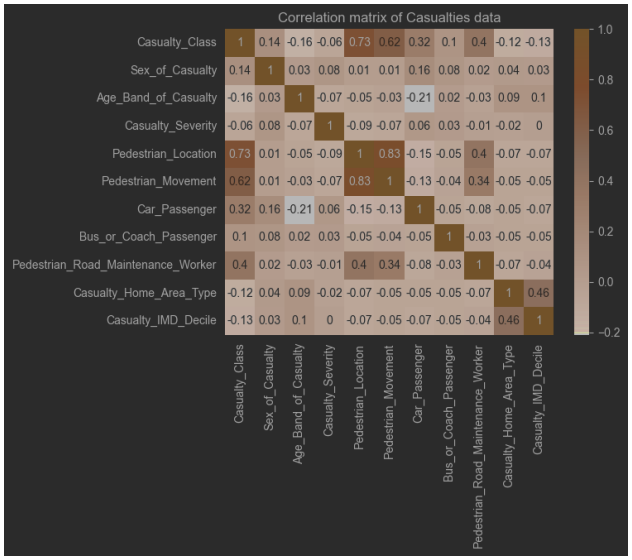
Rysunek 19: Ograniczenie prędkości, liczba pojazdów, liczba ofiar przed i po skalowaniu MinMax



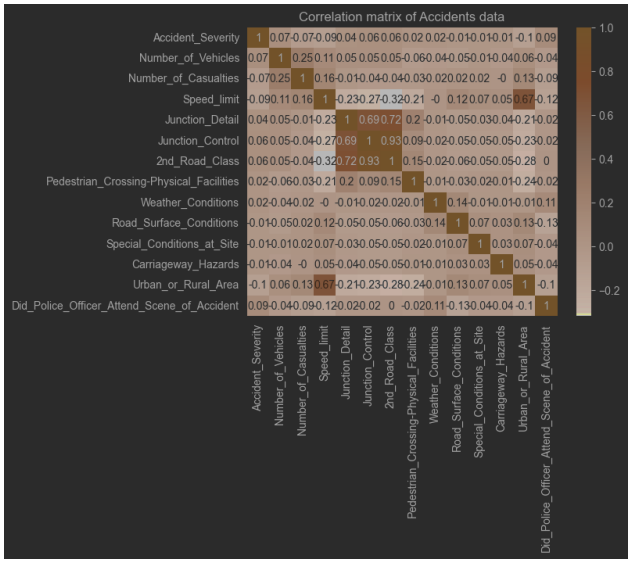
Rysunek 20: Ograniczenie prędkości, liczba pojazdów, liczba ofiar przed i po skalowaniu StandardScaler



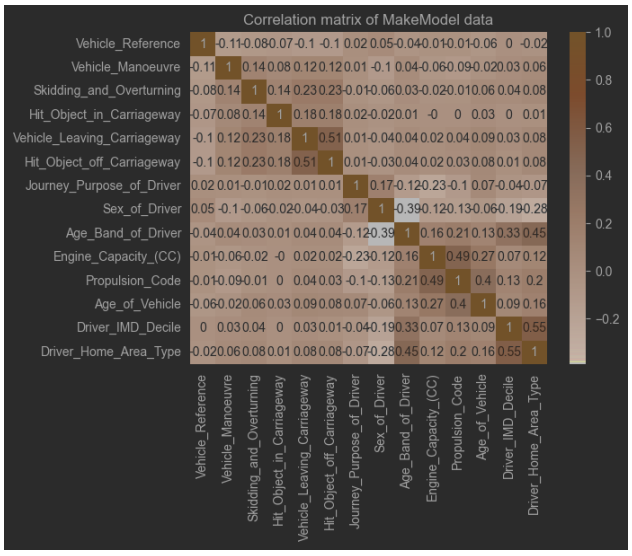
Rysunek 21: Macierz korelacji danych pojazdów



Rysunek 22: Macierz korelacji danych o wypadkach



Rysunek 23: Macierz korelacji danych dotyczących wypadków



Rysunek 24: Macierz korelacji danych MakeModel

miast, korelacja między "Age_Band_of_Driver" a "Age_of_Vehicle" wynosząca 0.13 jest stosunkowo niska, co sugeruje niewielkie związki między wiekiem kierowcy a wiekiem pojazdu. Z kolei, korelacja między "Age_Band_of_Driver" a "Driver_Home_Area_Type" wynosząca 0.45 sugeruje pewne powiązania między wiekiem kierowcy a typem obszaru, w którym mieszka. Podobnie, korelacja między "Age_Band_of_Driver" a "Driver_IMD_Decline" wynosząca 0.33 wskazuje na pewne związki między wiekiem kierowcy a ich stopniem deprywacji społecznej.

Macierz korelacji wszystkich zestawów danych (Rysunek 25) pozwala na zrozumienie wzajemnych związków między różnymi zmiennymi w danych. Poprzez analizę korelacji można odkryć, które zmienne są ze sobą powiązane i w jaki sposób. To z kolei może pomóc w identyfikacji kluczowych czynników wpływających na badane zjawiska oraz w wyborze odpowiednich zmiennych do dalszej analizy i modelowania.

7.1.19 Macierz korelacji dla wybranych cech

Wybrane cechy to: "Liczba pojazdów", "Liczba ofiar", "Typ drogi", "Limit prędkości", "Wiek kierowcy", "Wiek ofiary", "Dzień tygodnia", "Poważność wypadku", "Warunki pogodowe", "Warunki oświetleniowe", "Warunki nawierzchni drogowej", "Obszar miejski lub wiejski", "Szczegóły skrzyżowania". Analiza korelacji między wybranymi cechami i zmienną “Accident_Severity” może pomóc zidentyfikować istotne zmienne dla modelu klasyfikacji. Na podstawie wyników:

- Istnieje umiarkowana dodatnia korelacja między "Number_of_Vehicles" a "Number_of_Casualties" (0.32), co sugeruje, że wypadki z większą liczbą pojazdów mogą częściej powodować większą liczbę ofiar.
- "Speed_limit" wykazuje umiarkowaną dodatnią korelację z "Number_of_Vehicles" (0.29) i "Number_of_Casualties" (0.22), co sugeruje, że większe limity prędkości mogą być związane z większą liczbą pojazdów i ofiar.
- "Age_of_Casualty" ma silną ujemną korelację z "Age_of_Driver" (-0.33), co wskazuje na to, że starsi kierowcy mogą być bardziej narażeni na wypadki z udziałem starszych osób.
- Korelacja między "Weather_Conditions" a "Road_Type" (0.01) jest niska, co sugeruje, że warunki pogodowe mogą mieć niewielki wpływ na rodzaj drogi, na której występują wypadki.
- "Light_Conditions" wykazuje niewielką dodatnią korelację z "Road_Surface_Conditions" (0.17), co sugeruje, że gorsze warunki oświetleniowe mogą być powiązane z gorszym stanem nawierzchni drogi.

Te obserwacje mogą być użyteczne podczas budowania modelu klasyfikacji, aby wybrać istotne cechy dla przewidywania powagi wypadków.

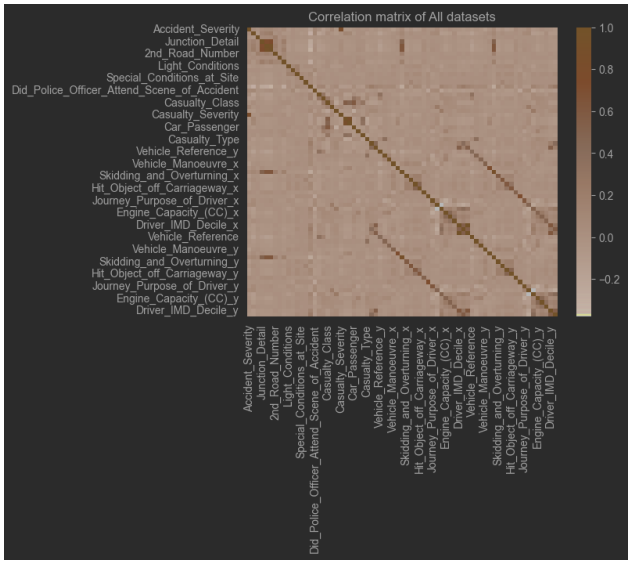
7.2 Trening modelu

7.2.1 Podział danych na zbiory treningowe i testowe

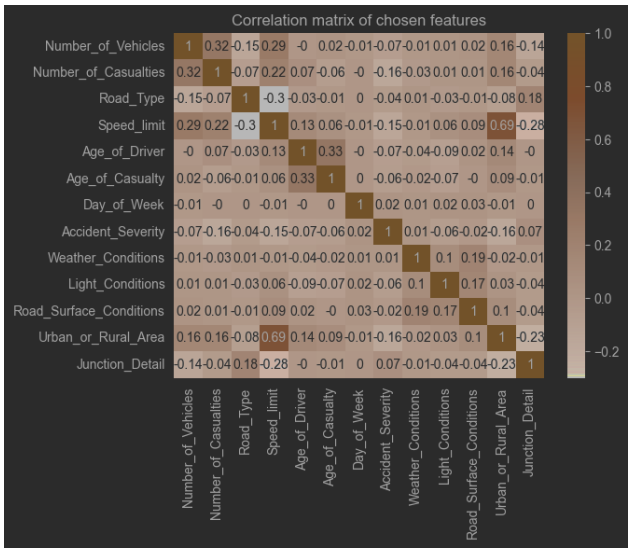
7.3 Podział danych na zbiory treningowe i testowe

Model został podzielony na zbiory treningowe i testowe w stosunku 8 do 2, czyli 80% danych będzie wykorzystane do nauki modelu, a pozostałe 20% będzie użyte do testowania modelu. Random State jest wybierany przy wywoływaniu funkcji podziału i na potrzeby tego projektu zawsze będzie on wynosić 17. Przy okazji zostały zdefiniowane funkcje do wyliczania pozostałych parametrów, według których oceniane będą metody klasyfikacji.

```
def train_test_split(X, y, test_size=0.2,
                     random_state=None):
    if random_state is not None:
        np.random.seed(random_state)

    n_samples = len(X)
    shuffled_indices =
            np.random.permutation(n_samples)

    n_test = int(n_samples * test_size)

    test_indices = shuffled_indices[:n_test]
    train_indices = shuffled_indices[n_test:]
```



Rysunek 25: Macierz korelacji wszystkich zbiorów danych



Rysunek 26: Macierz korelacji wybranych cech

```
    X_train = X[train_indices]
    X_test = X[test_indices]
    y_train = y[train_indices]
    y_test = y[test_indices]

    return X_train, X_test, y_train, y_test


def accuracy(y_true, y_pred):
    total_TP = total_TN =
        total_FP = total_FN = 0
    unique_classes = np.unique(y_true)

    for class_label in unique_classes:
        TP = np.sum((y_true == class_label)
                & (y_pred == class_label))
        TN = np.sum((y_true != class_label)
                & (y_pred != class_label))
        FP = np.sum((y_true != class_label)
                & (y_pred == class_label))
        FN = np.sum((y_true == class_label)
                & (y_pred != class_label))

        total_TP += TP
        total_TN += TN
        total_FP += FP
        total_FN += FN

    accuracy = (total_TP + total_TN) /
        (total_TP + total_TN +
        total_FP + total_FN)
    return accuracy


def score_f1(y_true, y_pred):
    classes = set(y_true)
    f1_scores = []
    for c in classes:
        TP = np.sum((y_true == c)
                    & (y_pred == c))
        FP = np.sum((y_true != c)
                    & (y_pred == c))
        FN = np.sum((y_true == c)
                    & (y_pred != c))
        precision = TP / (TP + FP) if
                    TP + FP > 0 else 0
        recall = TP / (TP + FN) if
                    TP + FN > 0 else 0
        f1 = 2 * precision * recall /
                    (precision + recall) if
                    precision + recall > 0 else 0
        f1_scores.append(f1)
    return np.mean(f1_scores)


def precision(y_true, y_pred):
    classes = np.unique(y_true)
    precisions = []
    for c in classes:
        TP = np.sum((y_true == c)
                    & (y_pred == c))
        FP = np.sum((y_true != c)
                    & (y_pred == c))
        precision = TP / (TP + FP) if
                    TP + FP > 0 else 0
        precisions.append(precision)
    return np.mean(precisions)


def recall(y_true, y_pred):
    classes = np.unique(y_true)
    recalls = []
    for c in classes:
        TP = np.sum((y_true == c)
                    & (y_pred == c))
        FN = np.sum((y_true == c)
                    & (y_pred != c))
        recall = TP / (TP + FN) if
                    TP + FN > 0 else 0
        recalls.append(recall)
    return np.mean(recalls)
```

7.4 Trening modelu klasyfikacji sześcioma metodami

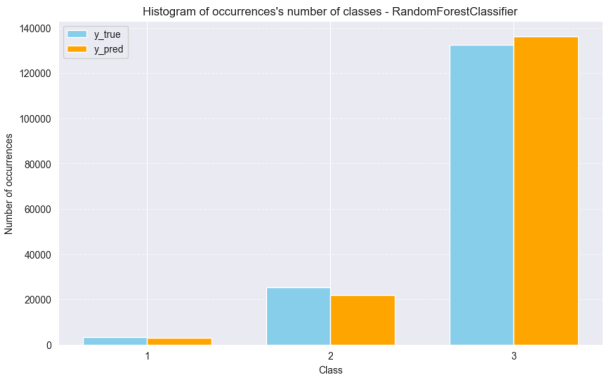
Do klasyfikacji wykorzystano sześć różnych metod, których wyniki będą poniżej zaprezentowane. Wszystkie rozwiązania były brane z bibliotek, a sprawdzane będą wyżej przedstawionymi naszymi funkcjami.

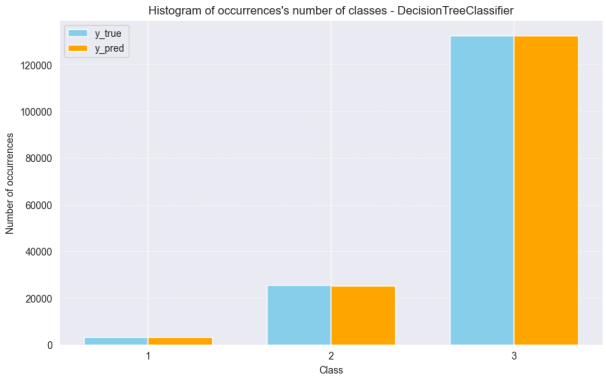
7.4.1 Tabela porównująca wyniki

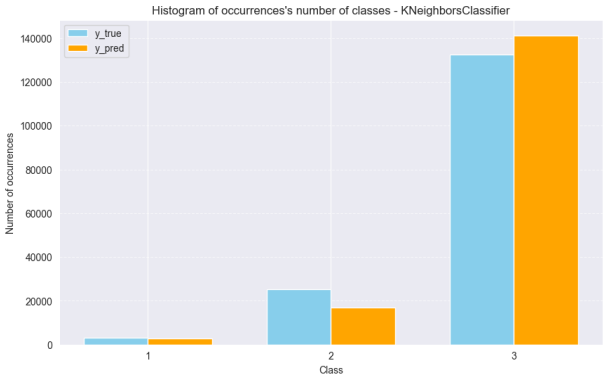
Nazwa Metody	**Dokł.**	**Prec.**	**Recall**	**Wynik F1**	**Czas tren. [s]**
Las Losowy	0.97	0.95	0.88	0.91	132.82
Drzewo decyzyjne	0.96	0.89	0.89	0.89	4.21
K-nearest neighbours	0.91	0.74	0.65	0.69	43.99
Klasyfikator LGBM	0.90	0.90	0.49	0.55	2.91
Klasyfikator XGBoost	0.42	0.06	0.29	0.09	4.87
Klasyfikator Catboost	0.85	0.85	0.85	0.80	10.48

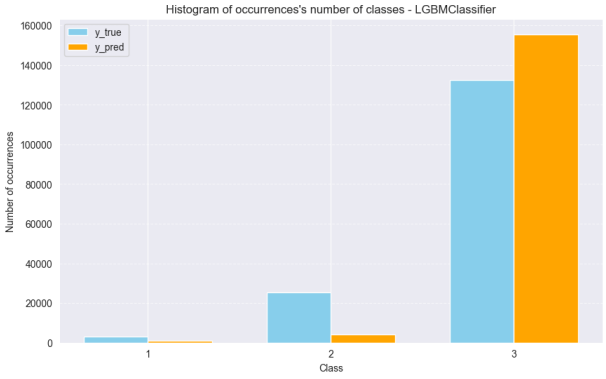
7.4.2 Histogramy prognoz z podziałem na klasy

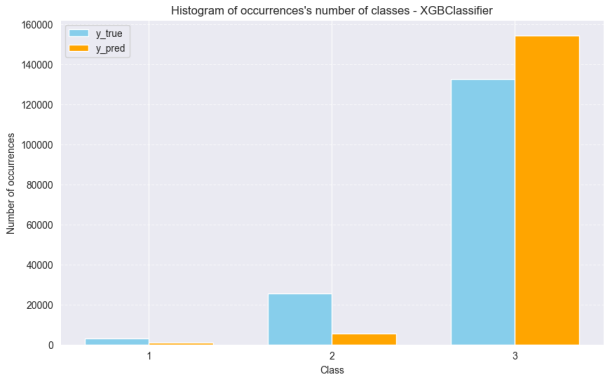
Poniżej zaprezentowane są histogramy dla każdego klasyfikatora, które prezentują ile wypadków zostało przewidziane dla danej kategorii, w stosunku do tego ile faktycznie jest wypadków w danej kategorii.

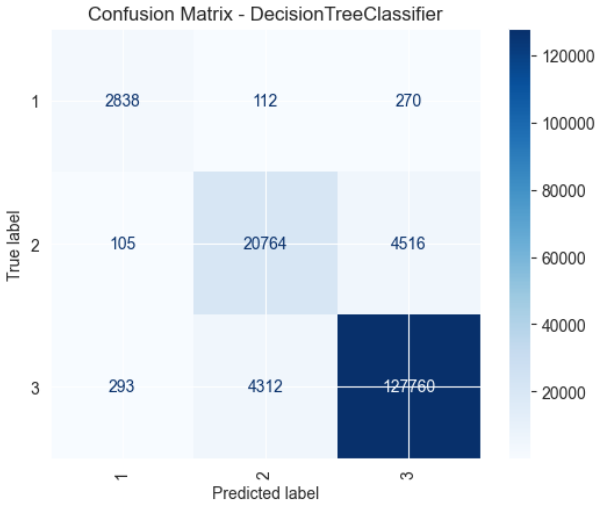






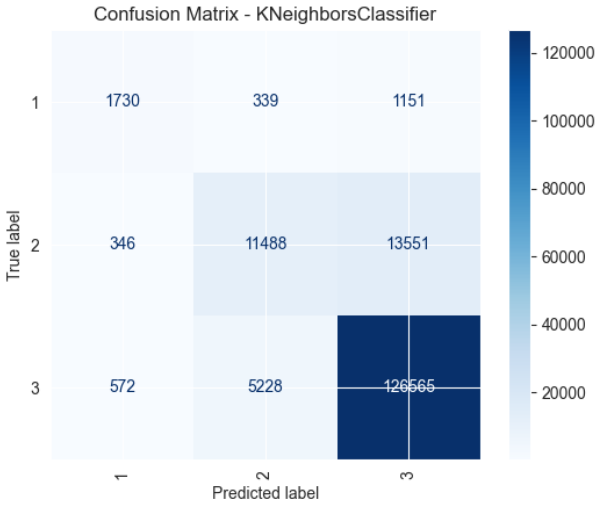


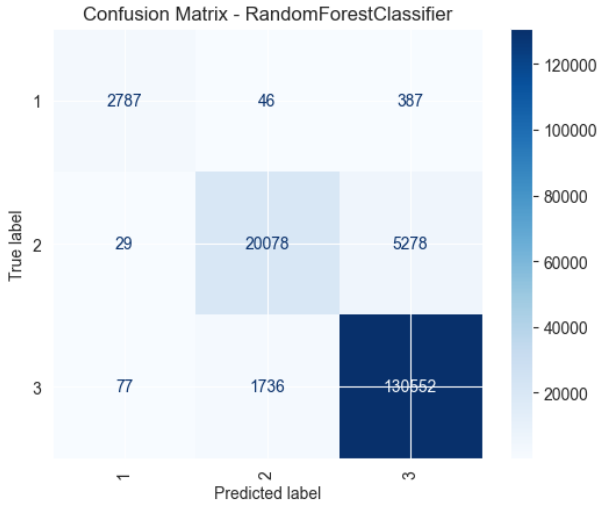


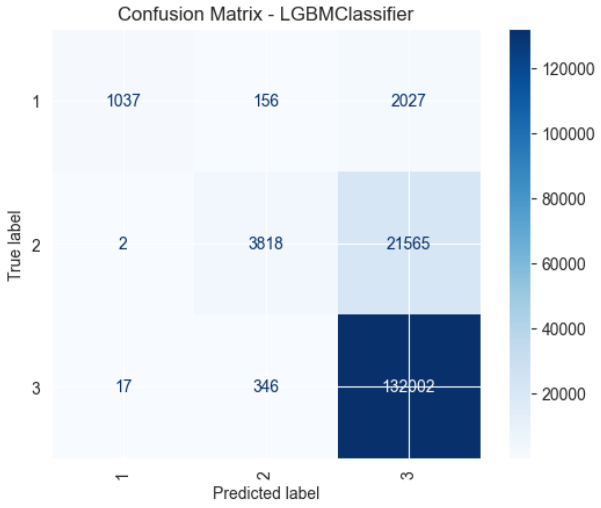


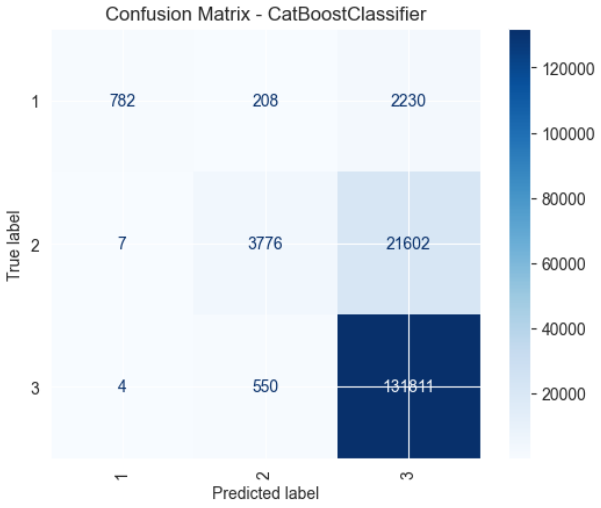
7.4.3 Macierz pomyłek

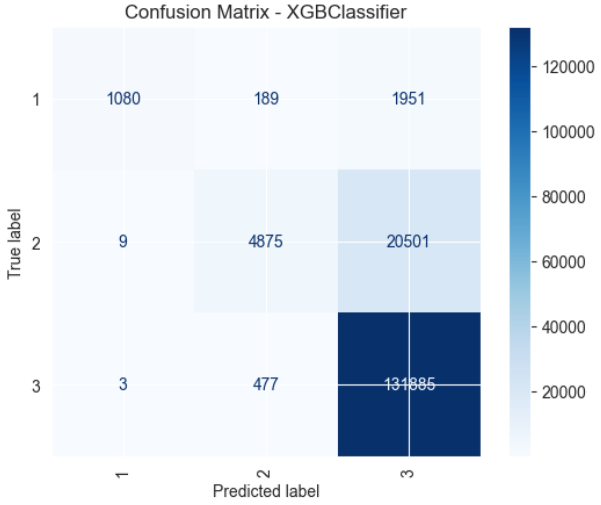
Poniżej przedstawiono macierze pomyłek, które w bardziej czytelny sposób informują o tym jakie błędy popełniły konkretne algorytmy i widzimy, że najrzadziej mylił się klasyfikator losowego lasu, a najczęściej klasyfikator CatBoost. Można też zauważyć, że największe problemy sprawiało algorytmom rozróżnienie wypadku śmiertelnego od wypadku po którym ofiara jest w ciężkim stanie.











7.5 Uczenie zespołowe

Uczenie zespołowe było wykonane na dwa sposoby, przy pomocy Voting Classifier i Stacking Classifier. W naszym przypadku Voting Classifier został zaimplementowany dla 6 grup klasyfikatorów. Cztery grupy są dla zbioru od 2 do 5 klasyfikatorów, a pozostałe dwie implementacje opierają się o sklearn i mają tylko albo 2, albo 5 klasyfikatorów. Za to Stacking Classifier zawsze jest dla wszystkich klasyfikatorów, ale w każdej implementacji inny klasyfikator jest uznawany za meta klasyfikator.

Przykładowa realizacja Voting Classifier dla 5 klastfikatorów:

```
base_classifiers = [LinearDiscriminantAnalysis_model,
                    GaussianNB_model,
                    QuadraticDiscriminantAnalysis_model,
                    KNeighborsClassifier_model]
meta_classifier = LogisticRegression_model

stacking_clf_41 = StackingClassifier(
                              base_classifiers,
                              meta_classifier)
stacking_clf_41.fit(X_train_data, y_train_data)
y_pred_stacking_clf_41 =
            stacking_clf_41.predict(X_test_data)

accuracy_stacking_clf_41 =
            accuracy_score(y_test_data,
                           y_pred_stacking_clf_41)
end_time = time.time()
stacking_clf_41_time = end_time - start_time
```

```
stacking_clf_41_precision,
    stacking_clf_41_recall,
    stacking_clf_41_f1_score =
        calculate_metrics(y_test_data,
                y_pred_stacking_clf_41)
```

Przykładowa realizacja Stacking Classifier dla meta klasyfikatora GaussianNB:

```
base_classifiers = [
        GaussianNB_model,
        QuadraticDiscriminantAnalysis_model,
        KNeighborsClassifier_model,
        LogisticRegression_model]
meta_classifier =
    LinearDiscriminantAnalysis_model

stacking_clf_45 =
    StackingClassifier(
            base_classifiers,
            meta_classifier)

stacking_clf_45.fit(
            X_train_data,
            y_train_data)
y_pred_stacking_clf_45 =
    stacking_clf_45.predict(X_test_data)

accuracy_stacking_clf_45 =
    accuracy_score(
            y_test_data,
            y_pred_stacking_clf_45)
end_time = time.time()
stacking_clf_45_time = end_time - start_time

stacking_clf_45_precision,
    stacking_clf_45_recall,
    stacking_clf_45_f1_score =
        calculate_metrics(
            y_test_data,
            y_pred_stacking_clf_45)
```

7.5.1 Tabela porównująca wyniki

Klasyfikator	**Dokładność**	**Czas wykonania [s]**
Voting (5)	0.830670	54.286620
Voting (4)	0.810754	52.818777
Voting (3)	0.803162	2.169591
Voting (2)	0.789172	1.554035
Voting (Sklearn, 5)	0.830670	59.874425
Voting (Sklearn, 2)	0.789172	1.805557
Stacking (meta LogisticRegression)	0.867907	236.159892
Stacking (meta KNeighborsClassifier)	0.818898	1177.476650
Stacking (meta QuadraticDiscriminantAnalysis)	0.832894	306.569000
Stacking (meta GaussianNB)	0.849177	369.965871

7.6 Walidacja krzyżowa

Do walidacji krzyżowej wykorzystano połowę danych ze zbioru treningowego, by móc je potem zestawić z zestawem treningowym, bez obawy o to, że zestaw będzie przeuczony. Dla naszego zbioru przetestowaliśmy dwie metody walidacji krzyżowej, metodę k-folds i metodę holdout. Niestety metoda leave-one-out i leave-p-out były niemożliwe do wykonania ze względu na rozmiar zbioru danych i permutacyjną naturę tamtych metod.

Obydwie z wykorzystanych metod mają bardzo zbliżoną dokładność, ale różnica jest widoczna w czasie wykonania, który dla metody holdout jest ponad 5 razy krótszy niż dla metody k-fold.

Przykładowa realizacja walidacji krzyżowej K-fold:

```
def k_fold_cross_val(X, y, k, model_fit_score):
    n = len(X)
    indices = list(range(n))
    np.random.shuffle(indices)
    scores = []

    for i in range(k):
        test_indices =
            indices[i * n // k: (i + 1) * n // k]
        train_indices =
            list(set(indices) - set(test_indices))
        X_train, X_test =
            X[train_indices], X[test_indices]
        y_train, y_test =
            y[train_indices], y[test_indices]
```

```
        score = model_fit_score(
                X_train, y_train,
                X_test, y_test)
        scores.append(score)

    return np.mean(scores)
```

7.6.1 Tabela porównująca wyniki

Metoda	Dokładność	Czas wykonania [s]
K-fold	0.911012	210.108243
Holdout	0.910173	40.131234

7.7 Wnioski

Analizując wyniki różnych modeli klasyfikacyjnych, można stwierdzić, że Random Forest i Decision Tree osiągnęły najwyższą skuteczność w klasyfikacji. Random Forest uzyskał najwyższą dokładność, co sugeruje, że agregacja wielu drzew decyzyjnych przyniosła korzyści. Jednakże, pomimo prostoty, pojedyncze drzewo decyzyjne również wykazało się wysoką dokładnością, co może być atrakcyjne ze względu na mniejsze wymagania obliczeniowe.

Modele oparte na gradient boosting, takie jak XGBoost i LGBM, wykazały mieszane wyniki. W szczególności, XGBoost wydaje się mieć problemy z precyzją i czułością, co sugeruje, że wymaga on bardziej zaawansowanej regularyzacji lub optymalizacji. CatBoost, podobnie jak inne metody oparte na boostingu, uzyskał dobre wyniki, ale nieco niższe niż Random Forest i Decision Tree. Jego czas trenowania jest relatywnie krótki, co może być atrakcyjne w sytuacji, gdy obciążenie obliczeniowe jest istotne. Niemniej jednak, jego dokładność, precyzja i czułość są na zadowalającym poziomie.

Algorytm KNN, mimo swojej popularności, nie osiągnął zadowalających wyników w tym zadaniu. Jego dokładność, precyzja i czułość były znacznie niższe niż w przypadku Random Forest i Decision Tree, a czas trenowania był najdłuższy spośród wszystkich modeli. Może to sugerować, że dla tego zestawu danych metoda oparta na odległościach nie jest odpowiednia.

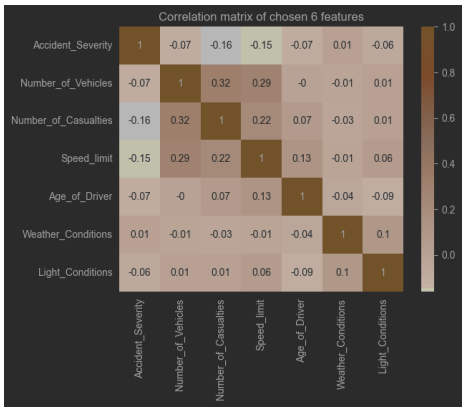
8 Optymalizacja - Optuna

8.1 Biblioteka Optuna

Analiza Optuna to podejście do automatyzacji procesu strojenia hiperparametrów w modelach uczenia maszynowego. Optuna jest biblioteką Pythona, która umożliwia wydajne i inteligentne przeszukiwanie przestrzeni hiperparametrów w celu znalezienia najlepszych konfiguracji modelu. Analiza przy użyciu Optuna może znacząco zwiększyć skuteczność modeli poprzez automatyczne dostosowanie hiperparametrów do specyfiki danych i problemu, oszczędzając czas i zasoby analityka danych.[1]

8.2 Optymalizacja na zrównoważonych danych

Zbiór danych został zredukowany do wybranych cech Number_of_Vehicles, Number_of_Casualties, Speed_limit, Age_of_Driver, Weather_Conditions, Light_Conditions. Skupienie się na istotnych cechach pozwala zoptymalizować proces uczenia i poprawić wydajność modelu, szczególnie w przypadku dużych zbiorów danych.



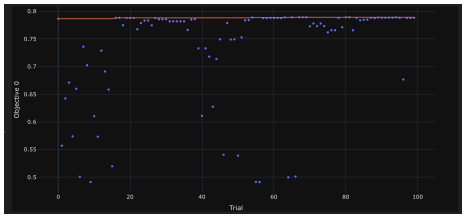
Rysunek 27: Macierz korelacji dla 6 wybranych cech

Dane są uznawane za niezbalansowane, gdy proporcje między różnymi klasami lub kategoriami w zbiorze danych są znacznie różne, co oznacza, że jedna klasa jest znacznie bardziej liczna od drugiej lub innych.Dlatego została użyta technika zrównoważania

danych SMOTE (Synthetic Minority Over-sampling Technique). SMOTE generuje sztuczne przykłady mniejszościowej klasy, które są podobne do istniejących przykładów, ale różnią się nieznacznie, co pomaga w zrównoważeniu proporcji klas. Poniżej zostanie zaprezentowana optymalizacja dla 2 algorytmów to Random Forest i Voting Classifier, ponieważ we wcześniejszym etapie pokazały one najlepszą dokładność.

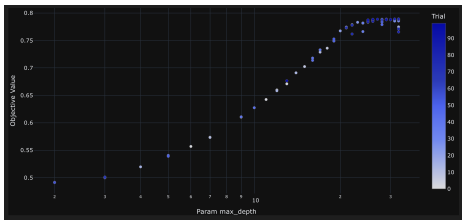
8.2.1 Random Forest

Optymalizacja została przeprowadzona na parametrach modelu RandomForestClassifier, w tym liczbie drzew (n_estimators), maksymalnej głębokości drzew (max_depth), minimalnej liczbie próbek wymaganej do podziału wewnętrznego węzła (min_samples_split) oraz minimalnej liczbie próbek wymaganej do utworzenia liścia (min_samples_leaf). Optimalizacja hiperparametrów w celu znalezienia optymalnej konfiguracji modelu przy założeniu maksymalizacji dokładności (accuracy). W celu przyspieszenia procesu optymalizacji, wykorzystano zrównoleglenie za pomocą parametru n_jobs=7, który wykorzystuje 7 procesów równolegle.



Rysunek 28: Wykres konwergencji

Optymalizacja dla Random Forest trwała około 10 godzin. Rysunek 28 przedstawia wszystkie przeprowadzone próby optymalizacyjne. Każda próba jest reprezentowana jako punkt na wykresie, gdzie oś X reprezentuje numer próby, a oś Y reprezentuje wartość dokładności.



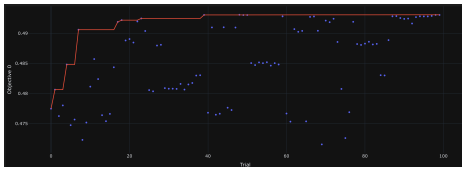
Rysunek 29: Wykres relacji hiperparametrów dla parametru max_depth

Najlepsze hiperparametry dla modelu Random Forest to:

- Liczba estymatorów = 210
- Maksymalna głębokość = 32
- Minimalny podział próbki = 6
- Minimalny podział liścia = 1

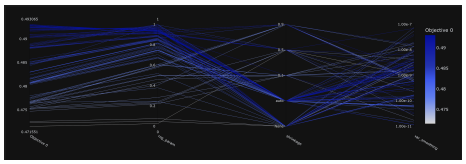
8.2.2 Voting Classifier - 3

Optymalizacja VotingClassifier polegała na dostrojeniu hiperparametrów dla GaussianNB, LinearDiscriminantAnalysis i QuadraticDiscriminantAnalysis za pomocą biblioteki Optuna. Dla GaussianNB optymalizowano parametr var_smoothing (wygładzenie macierzy kowariancji), dla LinearDiscriminantAnalysis shrinkage (), a dla QuadraticDiscriminantAnalysis reg_param. Proces obejmował 5-krotną walidację krzyżową i przeprowadzenie 100 prób optymalizacyjnych, aby maksymalizować dokładność klasyfikatora. Wynikiem optymalizacji były najlepsze wartości hiperparametrów, które zwiększały dokładność modelu. Czas trwania optymalizacji to około 12 minut. Dodatkwo, było wykorzystane zrównoleglenie za pomocą 7 procesów.



Rysunek 30: Wykres konwergencji

Wyker współrzędnych równoległych (Rysunek 31) służy do przedstawienia wielowymiarowych danych na dwuwymiarowej przestrzeni. Każdy parametr lub cecha w danych jest reprezentowany przez pionową oś, a każdy punkt danych (np. zestaw hiperparametrów w eksperymentach Optuna) jest reprezentowany przez linię, która przechodzi przez odpowiednie wartości na każdej osi. Linie umożliwiają jednoczesne porównanie wielu wymiarów i zidentyfikowanie zależności oraz wzorców między nimi. Pomaga w analizie danych o wielu wymiarach w sposób bardziej intuicyjny niż tablice liczbowych wartości.



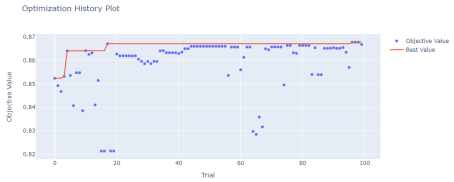
Rysunek 31: Wykres konwergencji

Najlepszymi hiperparametrami dla Voting Classifiera były:

- var_smoothing = 1.5646316772458296e-09
- shrinkage = auto
- reg_param = 0.9306512076868686

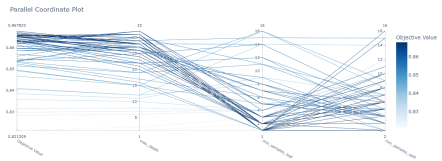
8.2.3 Drzewo Decyzyjne

Optymalizacja została przeprowadzona na parametrach modelu Drzewa Decyzyjnego, takich jak maksymanlna głębokość drzewa (max_depth), minimalnej liczbie próbek wymaganej do podziału wewnętrznego węzła (min_samples_split) oraz minimalnej liczbie próbek wymaganej do utworzenia liścia (min_samples_leaf). Optymalizacja hiperparametrów wykonana została pod kątem znalezienia takich wartości hiperaparametrów, które zwracają największą dokładność. Dla przyśpieszenie działania, zostało wykonane zrównoleglenie obliczeń przy pomocy parametrów n_jobs.



Rysunek 32: Wykres konwergencji

Optymalizacja dla Drzewa Decyzyjnego trwała około 6-10 minut. Jest to najszybsza optymalizacja w tym projekcie. Rysunek 32 przedstawia wszystkie przeprowadzone próby optymalizacyjne. Każda próba jest reprezentowana jako punkt na wykresie, gdzie oś X reprezentuje numer próby, a oś Y reprezentuje wartość dokładności.



Rysunek 33: Wykres wartości hiperparametrów w różnych zestawieniach

Najlepszymi hiperparametrami dla Drewa Decyzyjnego były:

- Maksymalna Głębokość = 27
- Minimalny podział próbki = 3
- Minimalny podział liścia = 1

8.3 Porównanie algorytmów

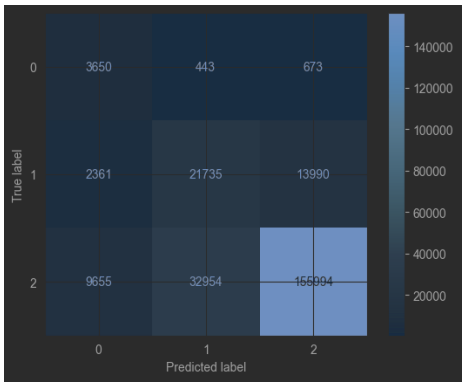
8.3.1 Random Forest - Zrównoważone dane

Liczba pojazdów i poszkodowanych oraz limit prędkości są kluczowymi czynnikami wpływającymi na powagę wypadku, podczas gdy warunki pogodowe i oświetleniowe mają mniejszy, ale nadal istotny wpływ.

Z macierzy pomyłek (Rysunek 30) dla modelu Random Forest ze sbalansowanymi dannymi (SMOTE)

Cecha	Wpływowość
Number_of_Vehicles	0.375
Number_of_Casualties	0.198
Speed_limit	0.166
Age_of_Driver	0.131
Weather_Conditions	0.069
Light_Conditions	0.061

można wyciągąc nastepujące wnioski: Fatalne wypadki (klasa 0): Klasyfikator nie popełnił błędów typu False Negative (FN) co oznacza, że wszystkie faktyczne fatalne wypadki zostały poprawnie sklasyfikowane jako fatalne. Jednakże, wystąpiły błędy typu False Positive (FP), gdzie 3650 przypadków wypadków sklasyfikowano jako fatalne, podczas gdy były one innego rodzaju. Poważne wypadki (klasa 1): W tym przypadku również nie było błędów typu False Negative, ale wystąpiły zarówno błędy typu False Positive (2361 przypadków), jak i False Negative (443 przypadki), co oznacza, że niektóre wypadki poważne zostały błędnie sklasyfikowane jako inne rodzaje wypadków, a także, że niektóre inne rodzaje wypadków zostały błędnie sklasyfikowane jako poważne. Lekkie wypadki (klasa 2): Podobnie jak w przypadku klas 0 i 1, nie ma błędów typu False Negative. Jednak błędy typu False Positive są znaczące (9655 przypadków), co oznacza, że wiele innych rodzajów wypadków zostało błędnie sklasyfikowanych jako lekkie.

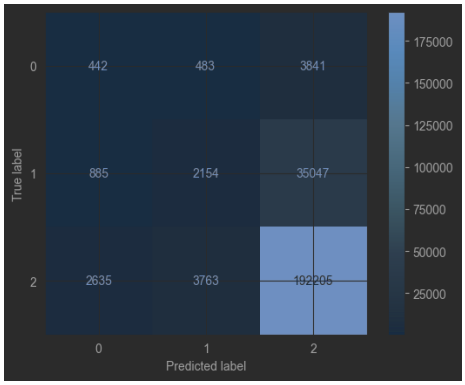


Rysunek 34: Macierz pomyłek dla Random Forest ze zrównoważonymi danymi

Ogólnie rzecz biorąc, klasyfikator wydaje się być skuteczny w identyfikowaniu fatalnych wypadków, ale ma trudności w dokładnym rozróżnianiu między poważnymi a lekkimi wypadkami, co prowadzi do znacznego liczby błędów w tych klasach.

8.3.2 Voting Classifier - Zrównoważone dane

Analiza macierzy pomyłek dla modelu Voting Classifier używającego GaussianNB, LinearDiscriminantAnalysis, QuadraticDiscriminantAnalysis z zbalansowanymi danymi SMOTE (Rysunek 33) jest następująca:



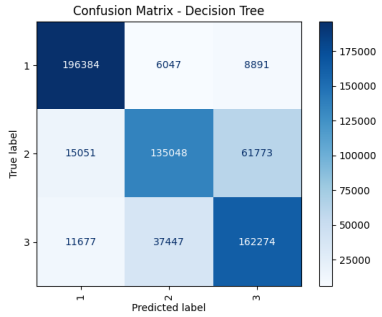
Rysunek 35: Macierz pomyłek dla Voting Classifier 3 ze zrównoważonymi danymi

Fatalne wypadki (klasa 0): Brak błędów pominięcia (False Negative). Wystąpiły zarówno błędy klasyfikacji innych wypadków jako fatalne (False Positive) - 442 przypadki, jak i błędy klasyfikacji fatalnych wypadków jako inne (False Negative) - 483 przypadki. Poważne wypadki (klasa 1): Wystąpiły zarówno błędy klasyfikacji innych wypadków jako poważne (False Positive) - 885 przypadków, jak i błędy klasyfikacji poważnych wypadków jako inne (False Negative) - 3763 przypadki. Lekkie wypadki (klasa 2): Wystąpiły błędy klasyfikacji innych wypadków jako lekkie (False Positive) - 2635 przypadków. Brak błędów klasyfikacji lekkich wypadków jako inne (False Negative). Tutaj widzimy, że mimo zastosowania techniki balansowania danych SMOTE, nadal występują znaczące błędy w klasyfikacji wszystkich trzech klas wypadków

więc jest potrzeba dalszej analizy i poprawy skuteczności modelu.

8.3.3 Drzewo decyzyjne - niezrównoważone dane

Macierz pomyłek dla klasyfikatora Drzewa Decyzyjnego ze zbalansowanymi danymi i z wykorzystaniem najlepszych parametrów jest następująca (Rysunek 36):



Rysunek 36: Macierz pomyłek dla Drzewa Decyzyjnego ze zrównoważonymi danymi

Fatalne wypadki (klasa 0): Brak błędów pominięcia (False Negative). Wystąpiły zarówno błędy klasyfikacji innych wypadków jako fatalne (False Positive) - 70 664 przypadki, jak i błędy klasyfikacji fatalnych wypadków jako inne (False Negative) - 49 154 przypadki. Poważne wypadki (klasa 1): Wystąpiły zarówno błędy klasyfikacji innych wypadków jako poważne (False Positive) - 43 494 przypadków, jak i błędy klasyfikacji poważnych wypadków jako inne (False Negative) - 76 824 przypadki. Lekkie wypadki (klasa 2): Wystąpiły błędy klasyfikacji innych wypadków jako lekkie (False Positive) - 2635 przypadków. Brak błędów klasyfikacji lekkich wypadków jako inne (False Negative).

8.4 Porównania wyników przed i po optymalizacji

Po optymalizacji, czas wykonania modeli Decision Tree i Voting Classifier się zmniejszył, jednak dokładność dla Decision Tree zmalała o 9%, a dla Voting Classifier pozostała na tym samym poziomie. W przypadku lasu losowego po optymalizacji dokładność zmalała o 14% a czas się wydłużych o 59,73 s.

Metoda	Dokładność	Czas wyk. {s}
Random Forest (przed)	0.97	132.82 s
Random Forest (po)	0.83	192.55 s
Decision Tree (przed)	0.96	4.21 s
Decision Tree (po)	0.87	1.24 s
Voting Classifier (przed)	0.80	2.169 s
Voting Classifier (po)	0.80	1.26 s

Zmiany te mogły wynikać z faktu, że domyślne parametry modeli były lepiej dostosowane do danych wejściowych przed optymalizacją. Optymalizacja mogła nie przynieść oczekiwanych rezultatów w przypadku wszystkich modeli, co sugeruje, że w niektórych przypadkach domyślne ustawienia mogły działać bardziej efektywnie niż parametry dobrane podczas procesu optymalizacji.

Najdokładniejszy model przed optymalizacją: Random Forest z dokładnością 0.97 z czasem wykonania (132.82 s), a najdokładniejszy model po optymalizacji też jest Random Forest z dokładnością 0.83, co jest nadal wyższe niż w przypadku pozostałych modeli, ale czas wykonania znacznie wzrósł (192.55 s). Najbardziej wydajny model po optymalizacji jest Voting Classifier, który utrzymał dokładność na poziomie 0.80, ale znacząco skrócił czas wykonania do 1.26 s. Najbardziej efektywny kompromis między dokładnością a czasem wykonania to Decision Tree po optymalizacji, z dokładnością 0.87 i czasem wykonania 1.24 s.

9 Podsumowanie

W projekcie zastosowano różne modele klasyfikacyjne, aby ocenić ich wydajność w przewidywaniu powagi wypadków. Każdy z tych modeli ma swoje unikalne zalety i wady, co pozwala na lepsze zrozumienie ich zastosowania w praktyce.

Analiza parametrów wykazała, że cechy takie jak liczba pojazdów, liczba ofiar, ograniczenie prędkości, wiek kierowcy, warunki pogodowe oraz warunki oświetleniowe są istotnymi determinantami powagi

wypadków. Dzięki wybraniu odpowiednich cech, modele mogły bardziej efektywnie przewidywać wyniki.

Dokładność modelu jest kluczowym wskaźnikiem jego wydajności, jednak czas treningu również odgrywa istotną rolę, zwłaszcza w kontekście praktycznym. Voting Classifier, mimo nieco niższej dokładności, oferuje znacznie krótszy czas treningu, co może być kluczowe w sytuacjach wymagających szybkiego modelowania i iteracji. Ostateczny wybór modelu powinien zależeć od konkretnych wymagań aplikacji oraz dostępnych zasobów obliczeniowych. W przypadkach, gdzie liczy się szybkie trenowanie modelu, Voting Classifier może być preferowany, natomiast tam, gdzie priorytetem jest najwyższa możliwa dokładność, Random Forest pozostaje silnym kandydatem.

Projekt dostarczył cennych wniosków na temat zastosowania różnych modeli klasyfikacyjnych w przewidywaniu powagi wypadków, podkreślając znaczenie odpowiedniego wyboru cech, metryk oceny oraz balansu między dokładnością a czasem treningu.

Wybór najlepszego modelu zależy od priorytetów: jeśli priorytetem jest maksymalna dokładność, Random Forest przed optymalizacją jest najlepszy. Jeśli jednak ważna jest wydajność czasowa, Voting Classifier po optymalizacji jest najlepszym wyborem. Dla kompromisu między dokładnością a czasem wykonania, najlepiej wypada Decision Tree po optymalizacji.

Literatura

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[2] Department for Transport. Road safety data. 2023.