

UNIwersytet RZESZOWSKI
Kolegium Nauk Przyrodniczych



Sebastian Płaza
nr albumu: 113758
Kierunek: Informatyka

Porównanie metod interpretowalnej sztucznej inteligencji (XAI)

Praca magisterska

Praca wykonana pod kierunkiem
dr hab. Jan Bazan

Rzeszów, 2024

Wprowadzenie

Cel i zakres pracy.

Celem pracy jest ...

Teza pracy

Tezą pracy jest

Spis treści

Wprowadzenie	0
1 Wprowadzenie Teoretyczne	3
1.1 Czym jest XAI?	3
1.2 Metoda Lime	5
1.3 Metoda SHAP	6
1.4 Metoda Explainable Boosting Machine	8
1.5 Metoda Dalex	10
2 Metody porównywania	12
2.1 Kryteria oceny	12
2.2 Opis postępowania przy porównaniu algorytmów	13
3 Przeprowadzone Eksperymenty	14
3.1 Narzędzia informatyczne	14
3.1.1 Przedstawienie algorytmu Random Forest	14
3.1.2 Działanie algorytmu Random Forest	14
3.1.3 Wady algorytmu Random Forest	15
3.1.4 Zalety algorytmu Random Forest	15
3.2 Opis eksperymentów	15
3.2.1 Opis danych użytych w eksperymencie	15
3.2.2 Metoda SHAP	16
3.3 Wyniki eksperymentów	21
3.4 Wnioski z eksperymentów	21
ZAKOŃCZENIE	22
LITERATURA	24

Rozdział 1

Wprowadzenie Teoretyczne

1.1 Czym jest XAI?

XAI, czyli Wyjaśnialna Sztuczna Inteligencja (ang. Explainable Artificial Intelligence), odnosi się do dziedziny badawczej i praktycznej, która koncentruje się na rozwijaniu metod i technik, które umożliwiają zrozumienie, interpretację i uzasadnienie decyzji podejmowanych przez systemy sztucznej inteligencji. W kontekście XAI, istotne jest, aby algorytmy i modele sztucznej inteligencji były wytłumaczalne i zrozumiałe dla ludzi, zarówno dla specjalistów, jak i dla użytkowników końcowych.

Dlaczego wyjaśnialność jest istotna w AI?

W miarę jak AI odgrywa coraz większą rolę w naszym życiu, zrozumienie, dlaczego systemy sztucznej inteligencji podejmują konkretne decyzje, staje się kluczowym elementem. Wyjaśnialność jest kluczowa dla budowania zaufania społecznego, poprawy akceptacji i umożliwienia efektywnej współpracy między ludźmi a sztuczną inteligencją, przez co głównymi celami XAI są:

- **Transparentność** - zapewnienie jasnego zobrazowania, jak algorytmy AI dochodzą do swoich wyników, użytkownicy powinni być w stanie zrozumieć procesy podejmowania decyzji.
- **Interpretabilność** - ułatwienie ludziom interpretacji działania algorytmów poprzez dostarczenie zrozumiałych i intuicyjnych wyjaśnień.
- **Zaufanie** - budowanie zaufania społecznego do sztucznej inteligencji poprzez eliminowanie tajemniczości w procesach decyzyjnych.

Różnice między tradycyjnymi algorytmami a systemami XAI:

Zrozumiałość - tradycyjne algorytmy uczenia maszynowego, takie jak regresja liniowa czy drzewa decyzyjne, są zazwyczaj łatwiejsze do zrozumienia i interpretacji. Z drugiej strony, zaawansowane modele, takie jak sieci neuronowe, są często trudne do zrozumienia i interpretacji, co prowadzi do określenia ich jako "czarne skrzynki".

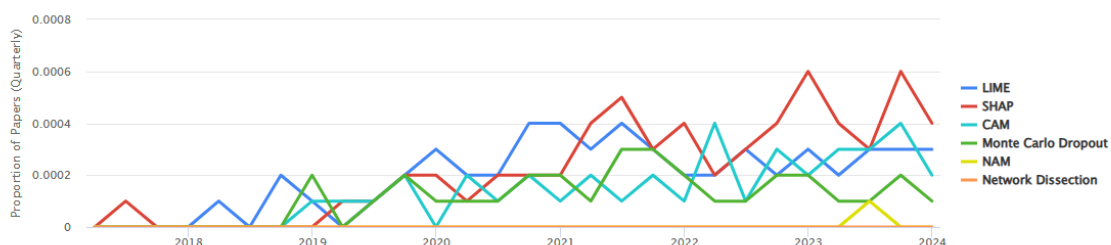
Wydajność predykcyjna - chociaż tradycyjne algorytmy są łatwiejsze do zrozumienia, ich wydajność predykcyjna jest zazwyczaj niższa w porównaniu do zaawansowanych modeli, takich jak głębokie sieci neuronowe.

Zaufanie i odpowiedzialność - systemy XAI mają na celu zwiększenie zaufania do modeli AI, poprzez umożliwienie użytkownikom zrozumienia i zaufania do wyników i wyjść generowanych przez algorytmy uczenia maszynowego. Dzięki temu, systemy XAI pomagają w budowaniu zaufania i pewności, kiedy modele AI są wprowadzane do produkcji.

Zgodność z regulacjami - w niektórych przypadkach, zrozumienie, jak system AI doszedł do konkretnego wyniku, może być niezbędne do spełnienia norm regulacyjnych. Systemy XAI mogą pomóc w spełnieniu tych wymagań, dostarczając wyjaśnień dotyczących procesów decyzyjnych modelu.

Zakres wyjaśnień: Systemy XAI mają na celu dostarczenie szerokiego zakresu informacji wyjaśniających o modelu, w tym informacji o danych treningowych, wydajności, niepewności itp.

Usage Over Time



Rysunek 1.1: Przedstawienie trendu użycia metod XAI

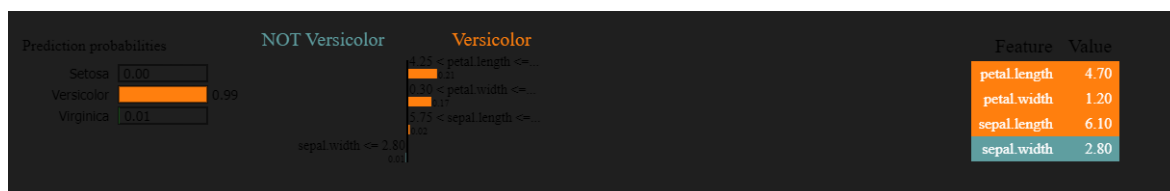
1.2 Metoda Lime

LIME (ang. Local Interpretable Model-agnostic Explanations) to algorytm, który może wyjaśniać prognozy dowolnego klasyfikatora w wiarygodny sposób, poprzez przybliżanie go lokalnie za pomocą interpretowalnego modelu. Algorytm LIME modyfikuje pojedynczą próbkę danych, dostosowując wartości cech i obserwując wynikający z tego wpływ na wynik. Wynikiem działania algorytmu jest zestaw wyjaśnień reprezentujących wkład każdej cechy w prognozę dla pojedynczej próbki, co jest formą lokalnej interpretowalności. Interpretowane modele w LIME mogą być na przykład regresją liniową lub drzewami decyzyjnymi, które są trenowane na małych zaburzeniach (np. dodawanie szumów, usuwanie słów, ukrywanie części obrazu) oryginalnego modelu, aby zapewnić dobre lokalne przybliżenie.

Przykłady zastosowania metody Lime

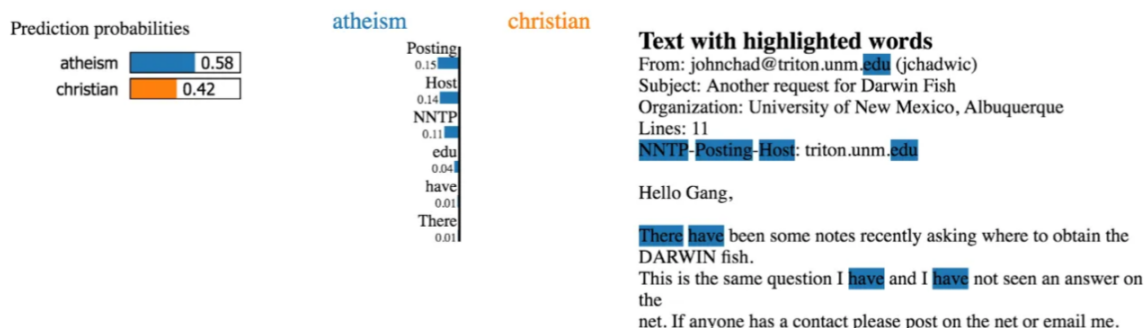
- W medycynie, LIME może pomóc lekarzom zrozumieć, dlaczego model uczenia maszynowego zalecił określone leczenie lub zdiagnozował określoną chorobę na podstawie danych pacjenta. Na przykład, jeśli model przewiduje, że pacjent ma wysokie ryzyko zachorowania na daną chorobę.

Poniżej podany przykład przedstawia wynik dla danych iris, na którym można zauważyć że dla badanej próbki szansa na przynależenie do klasy 'Versicolor' wynosi 99 proc i że zmienną która miała największy wpływ na decyzję jest petal.length



Rysunek 1.2: Przykład wyświetlanego tłumaczenia

- W przypadku modeli klasyfikacji obrazów, LIME może pokazać, które części obrazu były najważniejsze dla przewidywanej klasy. Na przykład, jeśli model klasyfikacji obrazów identyfikuje obraz jako "pies".
- Analiza Sentymentu: LIME może być używany do wyjaśnienia, które słowa w danym tekście przyczyniły się do określonego wyniku analizy sentymentu. Na przykład, jeśli model analizy sentymentu przewiduje, że dany tweet ma negatywny ton, LIME może pomóc zidentyfikować, które konkretne słowa w tweecie przyczyniły się do tego wyniku.



Rysunek 1.3: Przykład klasyfikacji tekstu

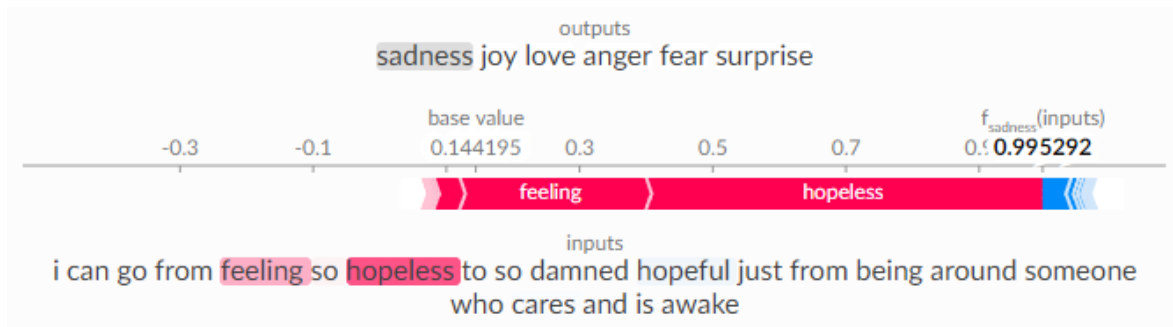
1.3 Metoda SHAP

SHAP (ang. SHapley Additive exPlanations) to podejście oparte na teorii gier, które służy do wyjaśniania wyników dowolnego modelu uczenia maszynowego. Łączy ona optymalne przydzielanie kredytów z lokalnymi wyjaśnieniami, korzystając z klasycznych wartości Shapleya z teorii gier i ich powiązanych rozszerzeń.

Wartość Shapleya to koncepcja z teorii gier, której celem jest przydzielenie wartości każdemu graczowi w grze kooperacyjnej w sposób sprawiedliwy i zgodny z wkładem każdego gracza. W kontekście modeli uczenia maszynowego, cechy są traktowane jako "gracze", a wartości Shapleya pomagają określić, jakie znaczenie miały poszczególne cechy w wyniku predykcji. SHAP jest używany do wyjaśniania wyników modeli uczenia maszynowego. Opiera się na wartościach Shapleya, które wykorzystują teorię gier do przypisywania kredytu za prognozę modelu każdej funkcji lub wartości funkcji

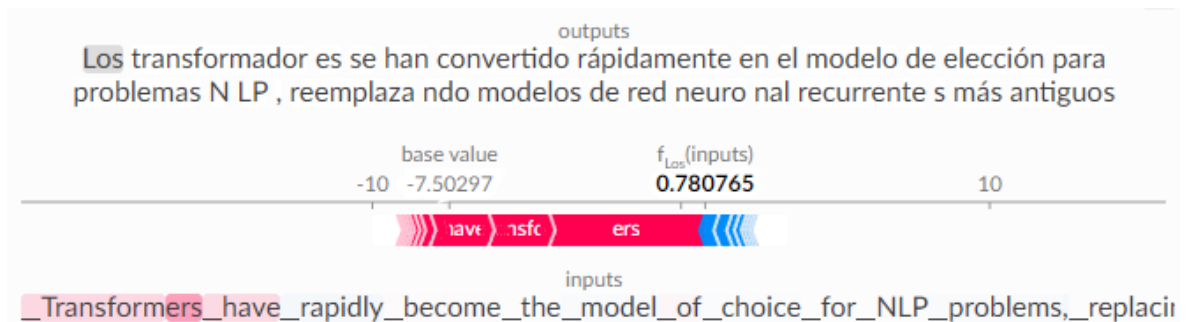
Przykłady zastosowania metody SHAP

- Analiza Sentymentu - SHAP może być używany do wyjaśnienia, które słowa w danym tekście przyczyniły się do określonego wyniku analizy sentymentu. Na przykład, jeśli model analizy sentymentu przewiduje, że dany tweet ma negatywny ton, SHAP może pomóc zidentyfikować, które konkretne słowa w tweecie przyczyniły się do tego wyniku.



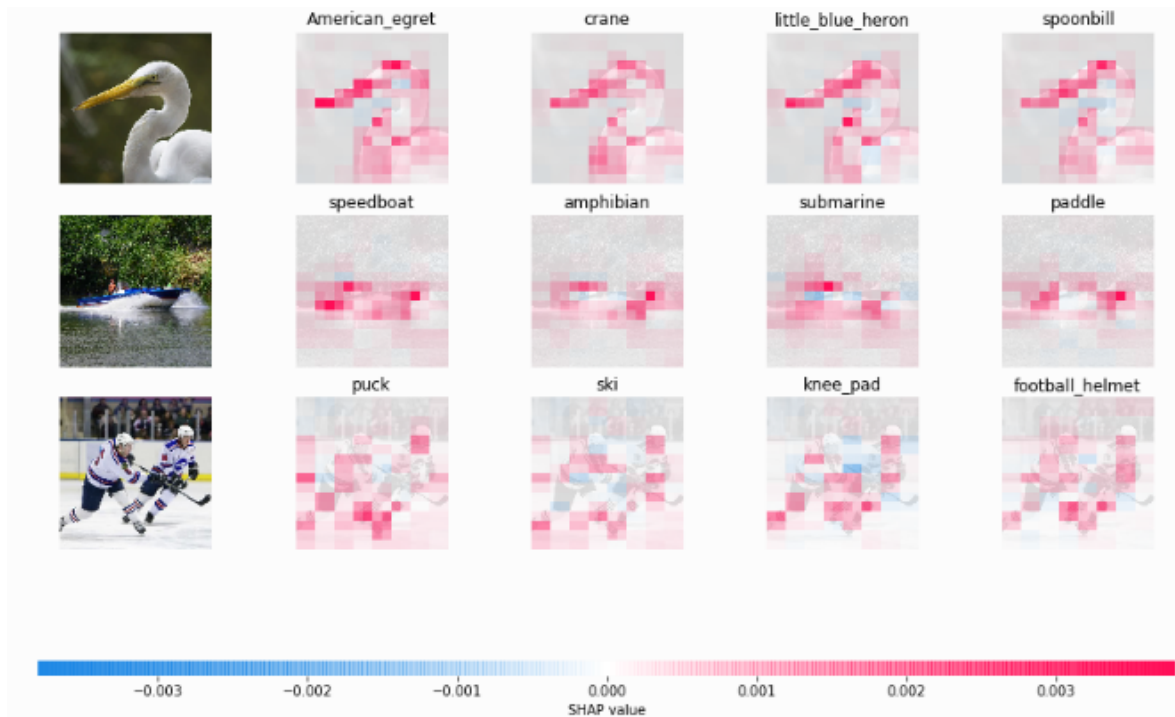
Rysunek 1.4: Przykład dla analizy sentymentu

- Tłumaczenie Maszynowe - w kontekście tłumaczenia maszynowego, SHAP może zobrazować, które części źródłowego tekstu miały największy wpływ na wygenerowane tłumaczenie.s



Rysunek 1.5: Przykład dla tłumaczenia maszynowego

- Klasyfikacja Obrazów - w przypadku modeli klasyfikacji obrazów, SHAP może pokazać, które części obrazu były najważniejsze dla przewidywanej klasy. Na przykład, jeśli model klasyfikacji obrazów identyfikuje obraz jako "czapla", SHAP może pomóc zidentyfikować, które części obrazu (takie jak dziób czy skrzydło) były kluczowe dla tej klasyfikacji.



Rysunek 1.6: Przykład dla klasyfikacji obrazów

- Diagnostyka Medyczna - w medycynie, SHAP może pomóc lekarzom zrozumieć, dlaczego model uczenia maszynowego zalecił określone leczenie lub zdiagnozował określoną chorobę na podstawie danych pacjenta². Na przykład, jeśli model przewiduje, że pacjent ma wysokie ryzyko zachorowania na cukrzycę, SHAP może pokazać, które cechy (takie jak wiek, waga, dieta itp.) przyczyniły się do tego wyniku.

1.4 Metoda Explainable Boosting Machine

EBM (ang. Explainable Boosting Machine) to oparty na drzewie, cykliczny uogólniony model addytywny zwiększający gradient z automatycznym wykrywaniem interakcji. EBM są często tak dokładne, a jednocześnie pozostają w pełni interpretowalne. Chociaż EBM są często wolniejsze w trenowaniu niż inne nowoczesne algorytmy, EBM są niezwykle kompaktowe i szybkie w czasie przewidywania. EBM jest uogólnionym modelem addytywnym (GAM) o następującej formie:

$$g(E[y|X]) = \beta_0 + \sum f_i(x_i)$$

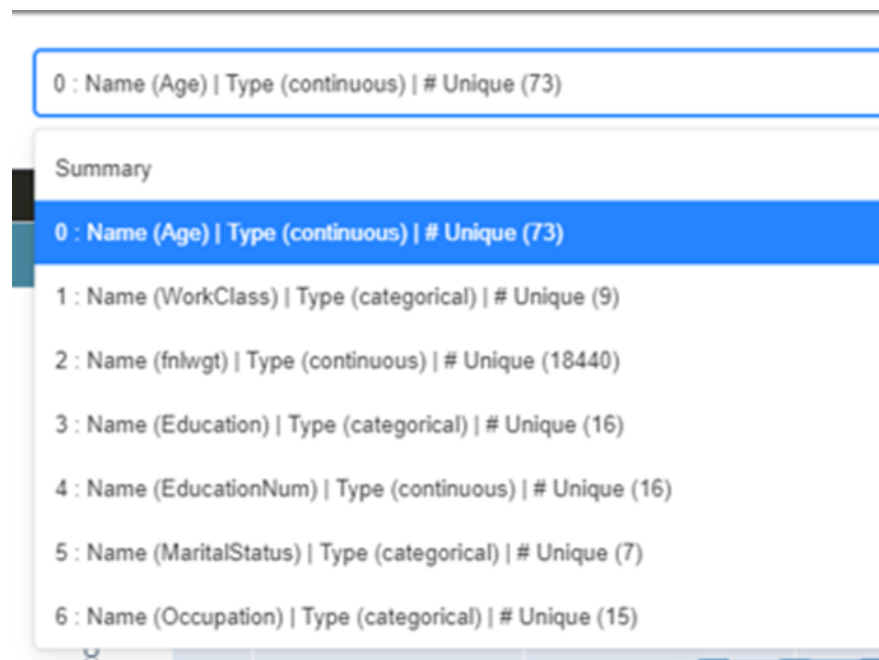
gdzie g jest funkcją łączącą, która dostosowuje GAM do różnych ustawień, takich jak regresja lub klasyfikacja. Procedura boosting jest starannie ograniczana do trenowania na

jednej funkcji na raz w sposób cykliczny, przy bardzo niskim współczynniku uczenia, dzięki czemu kolejność cech nie ma znaczenia.

EBM są wysoce zrozumiałe, ponieważ wkład każdej cechy do końcowej prognozy można zobrazować i zrozumieć, rysując $f(j)$. Każda funkcja $f(j)$ działa jako tabela wyszukiwania dla danej cechy i zwraca wkład do końcowej prognozy. Te wkłady są po prostu dodawane i przekazywane przez funkcję łączącą g, aby obliczyć końcową prognozę.

Przykłady zastosowania metody EBM

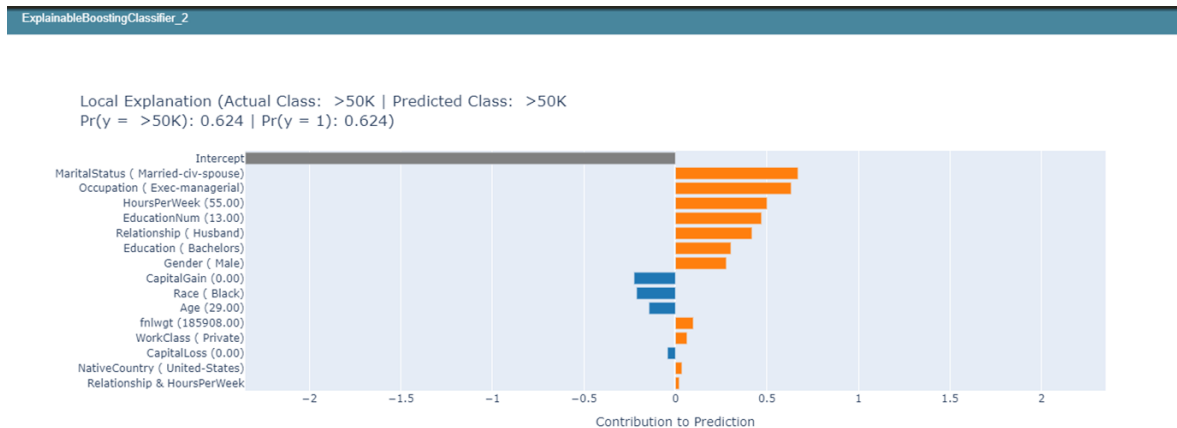
- Zastosowanie w naukach przyrodniczych - w jednym z badań, EBM zostały zastosowane do analizy obrazów solitonów atomów zimnych. W tym przypadku, EBM zostały użyte do analizy danych obrazowych po raz pierwszy. Aby przekształcić dane obrazowe na format tabularny, zastosowano techniki oparte na transformacji falkowej Gabora, które zachowują przestrzenną strukturę danych.
- Zastosowanie w astrofizyce - EBM zostały użyte do analizy prawie 6 milionów galaktyk symulowanych przez projekt Cosmic Reionization on Computers (CROC). Celem było zrozumienie, jak masa gwiazd i tempo formowania gwiazd zależą od właściwości fizycznych takich jak masa halo, szczytowa prędkość obrotowa galaktyki podczas jej historii formowania, środowisko kosmiczne i przesunięcie ku czerwieni.



0 : Name (Age) Type (continuous) # Unique (73)
Summary
0 : Name (Age) Type (continuous) # Unique (73)
1 : Name (WorkClass) Type (categorical) # Unique (9)
2 : Name (fnlwgt) Type (continuous) # Unique (18440)
3 : Name (Education) Type (categorical) # Unique (16)
4 : Name (EducationNum) Type (continuous) # Unique (16)
5 : Name (MaritalStatus) Type (categorical) # Unique (7)
6 : Name (Occupation) Type (categorical) # Unique (15)

Rysunek 1.7: Przykład możliwych do wybrania metryk

- Zastosowanie w uczeniu maszynowym - w ramach platformy Microsoft Fabric, EBM są używane do trenowania modeli klasyfikacyjnych. EBM tworzą zespół drzew decyzyjnych, podobnie jak gradient boosting, ale z unikalnym naciskiem na generowanie modeli zrozumiałych dla człowieka. Są one dobrze dopasowane do aplikacji, gdzie zrozumienie czynników wpływających na decyzje modelu jest niezbędne, takich jak opieka zdrowotna, finanse i zgodność z regulacjami.



Rysunek 1.8: Przykład generowanego wytłumaczenia

1.5 Metoda Dalex

Metoda DALEX jest jedną z metod, która służy do eksploracji i interpretacji zachowania dowolnego modelu predykcyjnego. Metoda ta opiera się na tworzeniu opakowania (wrapper) wokół modelu, które umożliwia zastosowanie różnych narzędzi do analizy globalnej i lokalnej modelu. Analiza globalna dotyczy całego zbioru danych i pozwala na ocenę ogólnego działania modelu, np. poprzez wyznaczanie ważności zmiennych lub profili zależności cząstkowej. Analiza lokalna dotyczy pojedynczej obserwacji i pozwala na zrozumienie, jak model generuje predykcję dla konkretnego przypadku, np. poprzez metodę Break Down lub Shapley values. Metoda DALEX jest niezależna od języka programowania i biblioteki, w której został stworzony model, dzięki czemu można ją stosować do różnych typów modeli, takich jak regresja logistyczna, lasy losowe, sieci neuronowe czy xgboost. Metoda DALEX jest zaimplementowana w pakietach R i Python, które są dostępne na GitHubie.

- Eksploracja i interpretacja zachowania dowolnego modelu predykcyjnego, np. poprzez ocenę ważności zmiennych, profili zależności cząstkowej, metodę Break Down lub Shapley value.

- Porównanie i ocena różnych modeli predykcyjnych, np. poprzez wyznaczanie krzywych ROC, AUC, MSE lub innych miar jakości.
- Weryfikacja i walidacja modeli predykcyjnych, np. poprzez sprawdzanie założeń, testowanie hipotez, wykrywanie obserwacji odstających lub wpływowych.
- Wizualizacja i prezentacja modeli predykcyjnych, np. poprzez tworzenie interaktywnych raportów, dashboardów lub aplikacji.
- W analizie ryzyka kredytowego, gdzie pozwala na porównanie i interpretację różnych modeli predykcyjnych, takich jak regresja logistyczna, lasy losowe, oraz na zrozumienie, jakie czynniki wpływają na prawdopodobieństwo niespłacenia kredytu przez klienta itp.

Rozdział 2

Metody porównywania

2.1 Kryteria oceny

Kryteria oceny działania metod XAI mogą być sformułowane na wiele różnych sposobów i na podstawie różnych kryteriów, które będą odpowiadać konkretnym potrzebom. W pracy skupiono się na podanych kryteriach:

- **Uniwersalność**, czyli czy metoda może być stosowana do wielu modeli sztucznej inteligencji?
- **Złożoność/Zrozumiałość**, czyli jak skomplikowane są wyjaśnienia. Czy są zrozumiałe dla osób bez specjalistycznej wiedzy?
- **Interaktywność**, czy użytkownik ma możliwość na dostosowanie wyjaśnień do potrzeb i preferencji?
- **Stabilność**, czy generowane wyjaśnienia są spójne dla podobnych wejść?

2.2 Opis postępowania przy porównaniu algorytmów

Aby porównać modele XAI zastosowana zostanie następująca lista kroków

1. **Wybór modeli AI**, modele na których będą testowane możliwości XAI to algorytm Random Forest oraz KNN (ang. k-nearest neighbors)
2. **Trenowanie modeli**, wytrenowanie modeli na których będą przeprowadzane testy algorytmów XAI.
3. **Ocena wydajności modeli** wykorzystując odpowiednie metryki, takie jak dokładność, macierz pomyłek, precyzja, F1 czy czułość.
4. **Zastosowanie modeli wytłumaczalnej sztucznej inteligencji**. Zastosowanie modeli SHAP, DALEX, LIME, EMB.
5. **Analiza wyjaśnień** interpretacja wyjaśnień otrzymanych z modeli XAI. Sprawdzenie jakie cechy mają największy wpływ na decyzję modelu.
6. **Porównanie wyjaśnień** polegające na sprawdzeniu, które modele dostarczają bardziej zrozumiałych i mniej złożonych wytłumaczeń/wyjaśnień.
7. **Wybór najlepszego modelu** na podstawie porównania wybrać model, który najlepiej spełnia kryteria oceny.

Rozdział 3

Przeprowadzone Eksperymenty

3.1 Narzędzia informatyczne

3.1.1 Przedstawienie algorytmu Random Forest

Algorytm Random Forest jest jednym z algorytmów opartych na uczeniu zespołowym, czyli takim, w którym łączy się kilka algorytmów, bądź ten sam algorytm wielokrotnie tak aby uzyskać jak najbardziej wydajny model predykcyjny. Algorytm zazwyczaj łączy wiele drzew decyzyjnych tworząc las, stąd też nazwa Las Losowy (ang. "Random Forest").

3.1.2 Działanie algorytmu Random Forest

- Wybierz N losowych rekordów z zestawu danych
- Zbuduj drzewo decyzyjne w oparciu o te N rekordów.
- Wybierz liczbę drzew, które chcesz w swoim algorytmie i powtórz kroki 1 i 2.
- W przypadku problemu regresji, dla nowego rekordu, każde drzewo w lesie przewiduje wartość dla Y (wyjście). Wartość końcowa może być obliczona poprzez przyjęcie średniej wszystkich wartości przewidywanych przez wszystkie drzewa w lesie. Lub, w przypadku problemu klasyfikacji, każde drzewo w lesie przewiduje kategorię, do której należy nowy rekord. Ostatecznie, nowy rekord jest przypisywany do kategorii, która wygrywa większością głosów.

3.1.3 Wady algorytmu Random Forest

- Jedną z największych wad algorytmu jest jego złożoność (dużą liczbę drzew decyzyjnych) przez co wymaga znacznych zasobów obliczeniowych.
- Znaczny koszt czasowy działania algorytmu

3.1.4 Zalety algorytmu Random Forest

- Algorytm działa dobrze jednocześnie na danych liczbowych jak i kategoriowych.
- Algorytm jest stabilny zmiana jednego punktu lub wprowadzenie nowego nie ma większego wpływu na wynik ponieważ zmiany takie mogą wpłynąć tylko na wynik jednego drzewa.
- Random Forest działa również dobrze na danych które są wybrakowane.

3.2 Opis eksperymentów

3.2.1 Opis danych użytych w eksperymencie

Zbiór danych składa się z 50 próbek trzech gatunków irysa tj. Iris setosa, Iris virginica i Iris versicolor. Dla każdej próbki zmierzono cztery cechy długość oraz szerokość płatków i działek kielicha. Zbiór ten przez swoją prostotę i niewielkie rozmiary jest często używany do nauki metod klasyfikacji.

1	sepal.length	sepal.width	petal.length	petal.width	species
2	5.1	3.5	1.4	.2	Setosa
3	4.9	3	1.4	.2	Setosa
4	4.7	3.2	1.3	.2	Setosa
5	4.6	3.1	1.5	.2	Setosa
6	5	3.6	1.4	.2	Setosa

Rysunek 3.1: Fragment danych zawartych w pliku Iris.csv

3.2.2 Metoda SHAP

Przedstawienie działania metody krok po kroku. Na początku należy zaimportować potrzebne biblioteki.

```
import shap
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

Rysunek 3.2: Import potrzebnych bibliotek

Następnym krokiem jest wczytanie danych i podzielenie ich na zbiory X odpowiadający za przechowywanie cech irysów oraz y za przechowywanie nazw gatunków. Dla ułatwienia zostały one wpisane do tabeli class_names.

```
iris = pd.read_csv('iris.csv')
X = iris.drop('species', axis=1)
y = iris['species']
class_names = ["Setosa", "Versicolor", "Virginica"]
```

Rysunek 3.3: Wczytanie danych

Kolejnym etapem jest podzielenie danych na zbiory treningowe i testowe.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Rysunek 3.4: Podział danych

Następnym krokiem jest utworzenie modelu i przetrenowanie go na danych treningowych.

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

Rysunek 3.5: Utworzenie i trenowanie modelu klasyfikacji

Kolejno utworzono explainer czyli interfejs do wyjaśniania modeli za pomocą wartości Shapleya.

```
# Utworzenie obiektu explainera SHAP
explainer = shap.Explainer(model)
```

Rysunek 3.6: Utworzenie obiektu explainera

Obliczanie wartości SHAP pokazuje znaczenie każdej cechy w porównaniu do innych cech, oraz jak każda wpływa na końcową prognozę. Dla każdego przykładu w `X_test`, obliczamy wartość SHAP dla każdej cechy, biorąc pod uwagę wszystkie możliwe kombinacje cech i średnią ich wpływ na wynik. Model przewiduje wynik dla każdej zmiany w danych, a różnica między tym przewidywanym wynikiem a oryginalnym wynikiem pozwala nam obliczyć wartość dla każdej cechy.

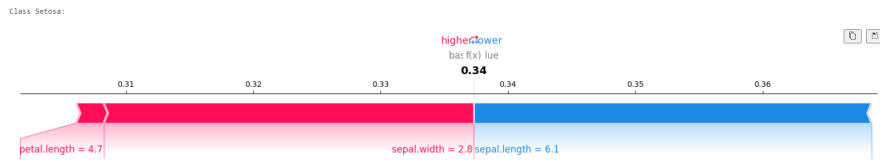
```
shap_values = explainer.shap_values(X_test)
```

Rysunek 3.7: Obliczanie wartości SHAP

```
shap.initjs()
observation_index = 0
for class_index in range(len(class_names)):
    print(f"Class {class_names[class_index]}:")
    shap.force_plot(
        explainer.expected_value[class_index],
        shap_values[class_index][observation_index],
        X_test.iloc[observation_index, :-1],
        feature_names=X_test.columns[:-1],
        show=True,
        matplotlib=True,
        #link="logit"
    )
```

Rysunek 3.8: Kod generujący wykres siły

Na wykresie można zauważyć która cecha zwiększa prawdopodobieństwo przynależenia do klasy Setosa i jest to długość płatka (`petal.length`) i szerokość działki kielicha (`sepal.length`). Cechą która zmniejsza prawdopodobieństwo przynależenia do klasy jest długość działki kielicha.



Rysunek 3.9: Wykres siły dla klasy Setosa

Utworzenie następnego explainera aby zastosować metody do generowania wykresów.

```
explainer = shap.explainers.Exact(model.predict_proba, X)
shap_values = explainer(X)
```

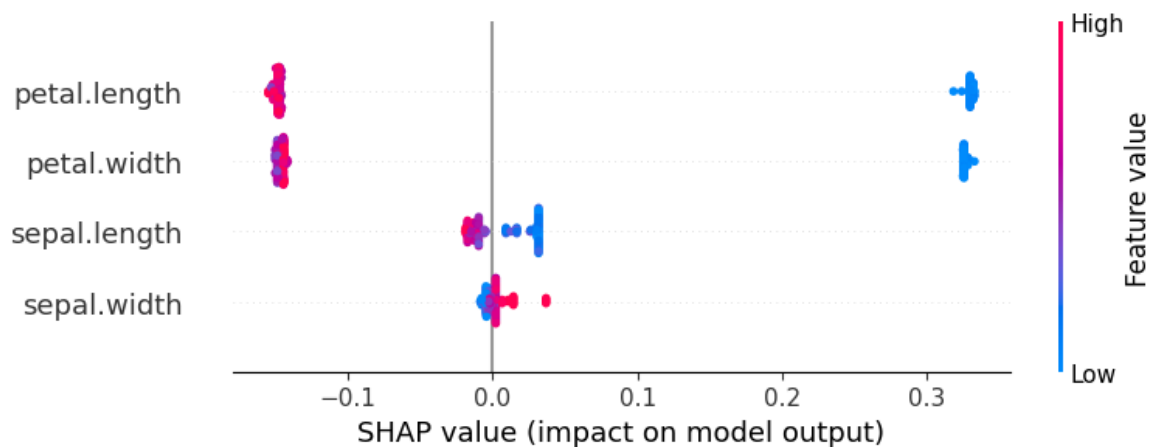
Rysunek 3.10: Obiekt explainera

Jak można zauważyć ten explainer różni się od poprzedniego ze względu na użycie metody **Exact**, którą używa się do dokładnego wyliczenia wartości SHAP dla dowolnego modelu. Jako argument przekazywana jest funkcja **predict_proba**, która zwraca prawdopodobieństwa dla klas.

```
shap.summary_plot(shap_values)
```

```
shap.summary_plot(shap_values)
```

Rysunek 3.11: Kod generujący wykres summary plot



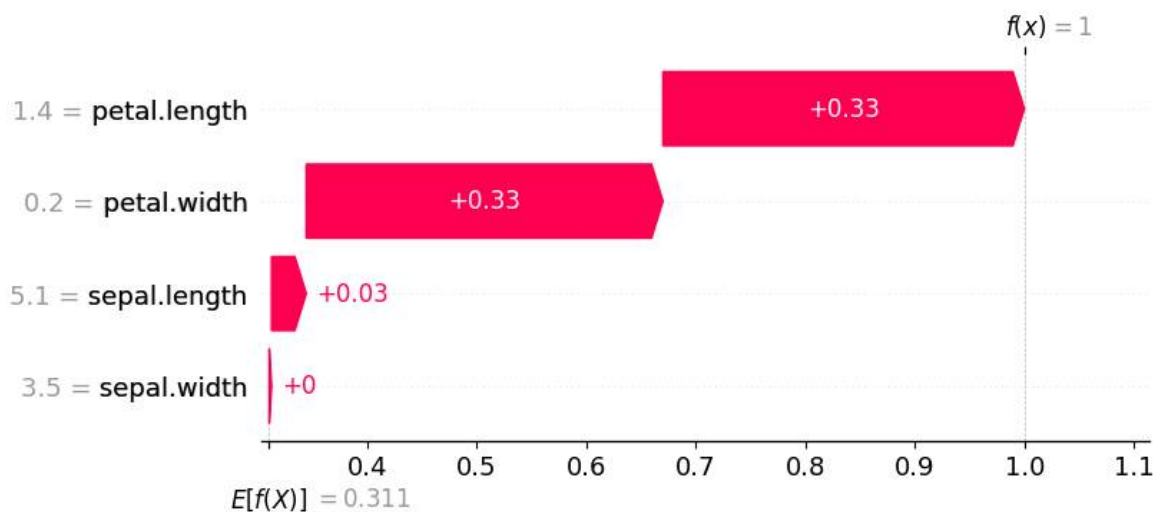
Rysunek 3.12: Wykres summary plot

Długość płatka wydaje się mieć największy wpływ na model, ponieważ punkty są bardziej rozproszone wzdłuż osi X. Szerokość płatka ma prawie tak samo mocny wpływ na model, ponieważ zbiór punktów jest niemalże identyczny jak dla poprzedniej cechy. Długość kielicha i szerokość kielich ma ją najmniejszy wpływa na model, ze względu na skupienie większości punktów tych cech przy wartości równej 0. Kolory punktów wskazują na wartość cechy im bardziej czerwony, tym wyższa wartość cechy, co wskazuje na większy wpływ na predykcję modelu.

```
shap.plots.waterfall(shap_values[0])
```

```
shap.plots.waterfall(shap_values[0])
```

Rysunek 3.13: Kod generujący wykres waterfall



Rysunek 3.14: Wykres waterfall

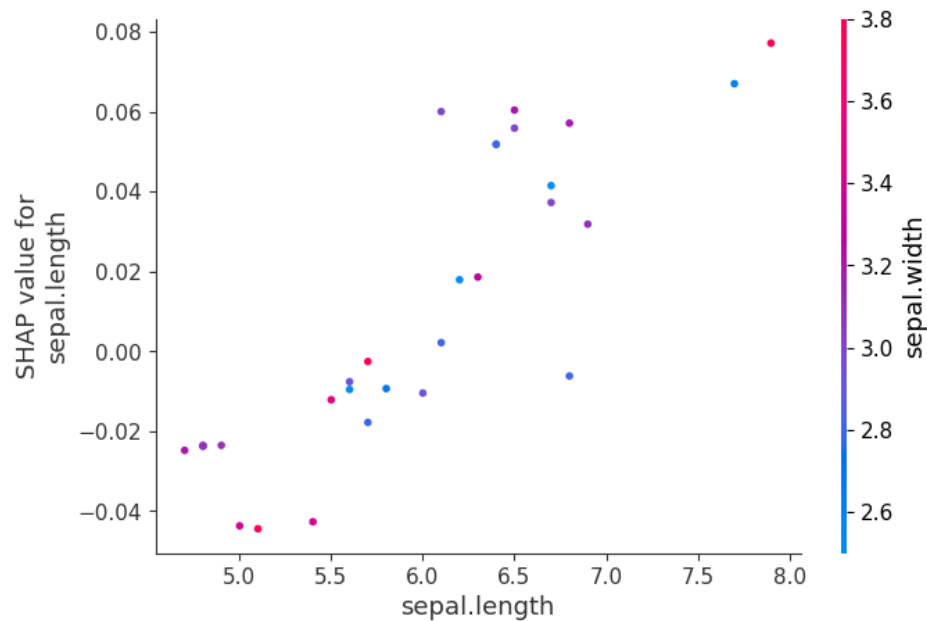
Wykres typu “waterfall” metody SHAP przedstawia wkład poszczególnych cech w przewidywany wynik modelu dla konkretnej instancji. Na osi Y wymienione są cechy, a długość paska na osi X wskazuje ich wpływ na wynik modelu. Paski mogą być dodatnie lub ujemne, co oznacza wzrost lub spadek wartości przewidywanej przez daną cechę.

Wartość bazowa, oznaczona jako $E[f(X)]$, wynosi 0,31 i jest to średnia przewidywana wartość modelu dla zbioru danych, $f(x) = 1$ jest to końcowa przewidywana wartość co widać w prawym górnym rogu wykresu.

```
shap.dependence_plot("sepal.length", shap_values[:, :, 2],
X_test,interaction_index="sepal.width")
```

```
shap.dependence_plot("sepal.length", shap_values[:, :, 2], X_test,interaction_index="sepal.width")
```

Rysunek 3.15: Kod generujący dependence plot



Rysunek 3.16: Wykres zależności Dependence plot

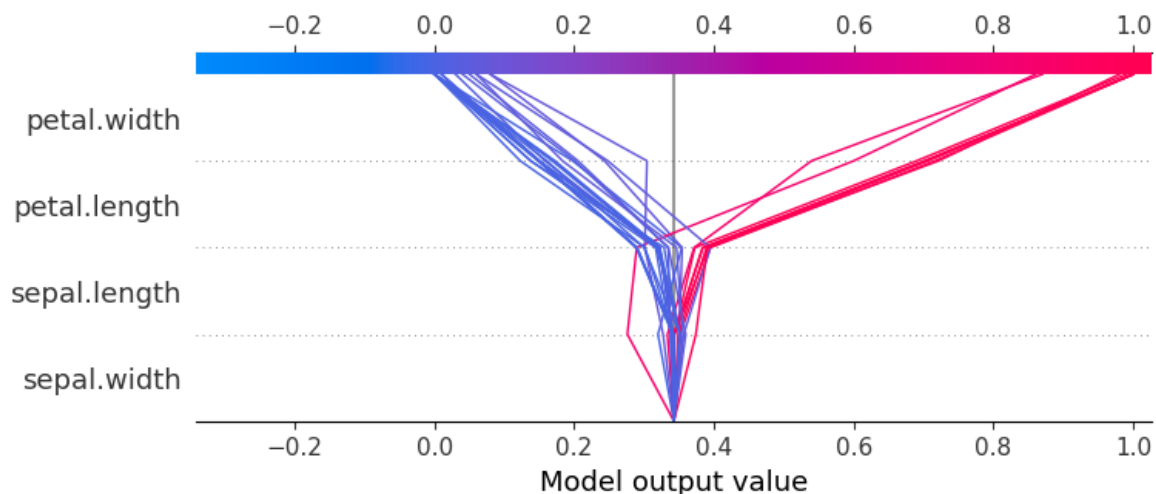
Wykres dependenceplot przedstawia rozproszenie punktów, gdzie oś pozioma zawiera wartości od 5.0 do 8.0. Oś pionowa zawiera wartości od około -0.04 do 0.08. Punkty na wykresie są kodowane kolorami według "sepal.width", z paskiem kolorów po prawej stronie wskazującym, że szerokość działki waha się od 2.6 do 3.8, z różnymi odcieniami od fioletu do różu reprezentującymi ten zakres.

```
shap.decision_plot(explainer.expected_value[1],
shap_values[:, :, 1], X_test.columns)
```

```
shap.decision_plot(explainer.expected_value[1], shap_values[:, :, 1], X_test.columns)
```

Rysunek 3.17: Kod generujący Decision Plot

Na osi Y wykresu znajdują się cztery cechy: szerokość płatka, długość płatka, długość działki i szerokość działki. Oś X przedstawia wartości od -0,2 do 1,0, które reprezentują wartość wyjściową modelu dla każdej cechy.



Rysunek 3.18: Decision Plot

Każda cecha ma wiele linii (w różnych odcieniach niebieskiego i różowego), co sugeruje, że dla każdej cechy wykreślono wiele instancji lub punktów danych. Linie te przecinają się w różnych punktach na osi X, wskazując, jak każda cecha przyczynia się do procesu podejmowania decyzji przez model przy różnych wartościach wyjściowych.

3.3 Wyniki eksperymentów

3.4 Wnioski z eksperymentów

Zakończenie

Przyszłość XAI, jest tematem, który budzi wiele zainteresowania i dyskusji. XAI to koncepcja, która zakłada, że sztuczna inteligencja (AI) powinna być w stanie wyjaśnić swoje działanie, procesy i decyzje ludziom, którzy z niej korzystają lub są przez nią dotknięci. XAI ma na celu zwiększenie zaufania, zrozumienia i odpowiedzialności AI, a także umożliwienie lepszego nadzoru, kontroli i poprawy AI.

Korzyści XAI

Zaufanie społeczne Wyjaśniana Sztuczna Inteligencja przyczynia się do zbudowania zaufania społecznego do systemów opartych na sztucznej inteligencji. Dostarczając transparentnych wyjaśnień dotyczących decyzji modeli, XAI eliminuje tajemniczość, co jest kluczowe dla zaakceptowania technologii przez społeczeństwo. Zrozumienie decyzji systemu AI Korzyść ta obejmuje lepsze zrozumienie procesów decyzyjnych modeli, zarówno dla użytkowników końcowych, jak i specjalistów ds. danych. Możliwość śledzenia, jakie cechy wpływają na konkretne decyzje, umożliwia bardziej efektywne doskonalenie modeli i dostosowywanie ich do oczekiwań. Przejrzystość procesów decyzyjnych XAI przyczynia się do przejrzystości procesów decyzyjnych w sztucznej inteligencji. Dzięki zrozumieniu, jak modele dochodzą do swoich wniosków, użytkownicy i decydenci są w stanie lepiej oceniać, czy decyzje są zgodne z wartościami etycznymi, a także skutecznie reagować na ewentualne błędy czy uprzedzenia.

Wyzwania związane z XAI

Balans między wyjaśnialnością a skutecznością Jednym z głównych wyzwań XAI jest znalezienie balansu między osiągnięciem pełnej wyjaśnialności a utrzymaniem wysokiej skuteczności modeli sztucznej inteligencji. Czasem bardziej skomplikowane modele mogą być trudniejsze do wytłumaczenia, co stawia wyzwanie przed badaczami i praktykami XAI.

Innym wyzwaniem jest ryzyko nadmiernego uproszczenia wyjaśnień, co może prowadzić do utraty istotnych niuansów i dokładności w zrozumieniu decyzji modelu. Ważne jest, aby utrzymać równowagę pomiędzy zrozumiałością a dokładnością wyjaśnień.

Problemy etyczne Etyczne aspekty związane z XAI obejmują pytania dotyczące prywatności danych, uczciwości w stosowaniu algorytmów oraz potencjalnego wpływu wyjaśnień na decyzje społeczne. Konieczne jest rozważenie tych kwestii, aby XAI było nie tylko skuteczne, ale także zgodne z wartościami społecznymi.

Przyszłość XAI możliwe ścieżki rozwoju

- XAI staje się standardem i wymogiem dla wszystkich systemów AI, które mają wpływ na ludzi. AI musi być w stanie udzielać jasnych, zrozumiałych i uzasadnionych wyjaśnień na żądanie lub z własnej inicjatywy. Ludzie mają prawo do dostępu, rewizji i poprawy AI, a także do odwołania się od jej decyzji. AI jest poddawana regularnym testom, audytom i certyfikatom dotyczącym jej wytłumaczalności.
- XAI jest zróżnicowana i dostosowana do różnych kontekstów, celów i odbiorców. AI może wyjaśniać się na różnych poziomach abstrakcji, szczegółowości i języka, w zależności od tego, kto i po co pyta. AI może również dostarczać różne rodzaje wyjaśnień, takie jak przyczynowe, kontrfaktyczne, probabilistyczne, narracyjne, wizualne lub interaktywne. AI jest projektowana i oceniana z uwzględnieniem specyfiki danej dziedziny, zadania i użytkownika.
- XAI jest kreatywna i innowacyjna, a nie tylko opisowa i reaktywna. AI nie tylko wyjaśnia swoje istniejące działania, ale również proponuje nowe możliwości, alternatywy i scenariusze. AI może również uczyć się z wyjaśnień, które otrzymuje lub udziela, i poprawiać swoje działanie, zrozumienie i komunikację. AI może również współtworzyć z ludźmi lub z innymi systemami AI, dzieląc się swoimi pomysłami, opiniami i emocjami.

LITERATURA

- [1] Y.M. Berezansky, Z.G. Sheftel, G.F. Us, *Functional Analysis*, Birkhäuser Verlag, Basel - Boston - Berlin, 1996.
- [2] B. Fisher, *The product of distributions*, Quart. J. Math. Oxford **22** (1971), 291–298.
- [3] S. Łojasiewicz, *Wstęp do teorii funkcji rzeczywistych*, PWN, Warszawa, 1973.
- [4] J.C. Oxtoby, *Measure and Category*, Springer–Verlag, New York - Heidelberg - Berlin, 1971.

STRESZCZENIE

Tytuł pracy w języku polskim:

Tytuł pracy w języku angielskim:

Streszczenie:

Tekst streszczenia

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.