INTERPRETOWALNOŚĆ I WYJAŚNIALNOŚĆ UCZENIA MASZYNOWEGO

Dr Robert Małysz

WYKŁAD 4 - AGENDA

1. Wartości Shapley'a
   - Idea i intuicja
   - Definicja matematyczna
   - Właściwości
   - Metody estymacji
   - Zalety i ograniczenia
2. SHAP
   - SHAP vs wartości Shapleya
   - KernelSHAP
   - TreeSHAP
   - Przykład: Regresja logistyczna
   - Przykład: xgboost
   - Interpretacja wyników

WYKŁAD 4 – CEL WYKŁADU

- Wyjaśnić koncepcję wartości Shapleya i jej pochodzenie z teorii gier
- Obliczyć wartości Shapleya dla prostych przykładów
- Zrozumieć różnicę między wartościami Shapleya a SHAP
- Zastosować SHAP do interpretacji modeli ML (regresja logistyczna, XGBoost)
- Interpretować różne typy wykresów SHAP
- Wybrać odpowiednią metodę wyjaśnialności dla danego problemu

WYKŁAD 4 – ZASTOSOWANIE PRAKYTCZNE

- Walidacja modeli IRB (zgodność z EBA/REP/2023/28)
- Wyjaśnienie decyzji kredytowych dla klientów
- Model debugging i feature engineering
- Audyt algorytmiczny i compliance

WYKŁAD 4 – DEFINICJA WARTOŚCI SHAPLEYA

Niech v będzie funkcją charakterystyczną gry kooperacyjnej, gdzie $v(S)$ określa wartość koalicji S graczy. Wartość Shapleya ϕ_i dla gracza i jest zdefiniowana jako:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)]$$

gdzie:

- N to zbiór wszystkich graczy.
- $|S|$ to liczba graczy w koalicji S.
- $|N|$ to całkowita liczba graczy.
- Suma jest obliczana dla wszystkich podzbiorów S zbioru N niezawierających gracza i.

Intuicyjnie, wartość Shapleya dla gracza i jest średnią wartością, jaką gracz i dodaje do wszystkich możliwych koalicji, które może utworzyć z innymi graczami.

WYKŁAD 4 – PRZYKŁAD 1

Wartość Shapleya dla dwóch graczy

Możliwe kolejności przybycia:

1. Najpierw 1, potem 2
2. Najpierw 2, potem 1

📊 Obliczenia:

Gracz 1:
Jeśli 1 przychodzi najpierw, wkład = v({1}) = 10

Jeśli 1 przychodzi po 2, wkład = v({1,2}) - v({2}) = 24 - 12 = **14**

$x_1 = (10 + 14)/2 = 11$

Gracz 2:
Jeśli 2 przychodzi najpierw, wkład = v({2}) = 12

Jeśli 2 przychodzi po 1, wkład = v({1,2}) - v({1}) = 24 - 10 = **12**

$x_2 = (12 + 12)/2 = 13$

Wynik końcowy: Wartość Shapleya $x_1 = 11$ oraz $x_2 = 13$
Weryfikacja: $x_1 + x_2 = 11 + 13 = 24$ ✓

Prawdopodobieństwo	Kolejność przybycia	Marginalny wkład 1	Marginalny wkład 2
1/2	najpierw 1, potem 2	10	14
1/2	najpierw 2, potem 1	12	12

WYKŁAD 4 – PRZYKŁAD 2

Wartość Shapleya dla trzech graczy

Załóżmy, że mamy prostą grę z trzema graczami: **A, B i C**. Funkcja wartości *v* dla różnych koalicji jest następująca:

Koalicje pojedyncze:

v({A}) = 1

v({B}) = 2

v({C}) = 3

Koalicje podwójne:

v({A, B}) = 4

v({A, C}) = 5

v({B, C}) = 6

Koalicja wszystkich graczy:

v({A, B, C}) = 7

WYKŁAD 4 – PRZYKŁAD 2

Wartość Shapleya dla trzech graczy

1. Rozważmy wszystkie kolejności, w których gracze mogą dołączać: **ABC, ACB, BAC, BCA, CAB, CBA**

2. Dla każdej kolejności obliczmy różnicę w wartości koalicji przed i po dołączeniu gracza A:

Kolejność	Koalicja przed A	Wartość przed	Wartość po	Wkład A
ABC	∅	0	v({A}) = 1	**1**
ACB	∅	0	v({A}) = 1	**1**
BAC	{B}	2	v({A,B}) = 4	**2**
BCA	{B,C}	6	v({A,B,C}) = 7	**1**
CAB	{C}	3	v({A,C}) = 5	**2**
CBA	{C,B}	6	v({A,B,C}) = 7	**1**

3. Uśredniamy te różnice:

$$\varphi_A = (1 + 1 + 2 + 1 + 2 + 1) / 6 = 8/6 = 4/3 \approx 1.33$$

💡 **Podobnie możemy obliczyć wartość Shapleya dla graczy B i C.**
Końcowy wynik: $\varphi_A = 4/3$, $\varphi_B \approx 2.17$, $\varphi_C \approx 2.50$

WYKŁAD 4 – WARTOŚCI SHAPLEYA

- Interpretacja wartości Shapleya dla wartości cechy *j* jest następująca:

  Wartość *j*-tej cechy przyczynia się o *ϕj* do prognozy tego konkretnego przypadku w porównaniu z przeciętną prognozą dla zbioru danych.

  - Porównanie z przeciętną prognozą dla całego zbioru danych.
  - Średnia wartość przypisana tej cechy po uwzględnieniu wszystkich kombinacji innych cech.
  - Miara wpływu cechy na prognozę w stosunku do oczekiwanej wartości prognozy bez tej cechy.

- Wartość Shapleya można stosować zarówno dla modeli klasyfikacyjnych (jeśli mamy do czynienia z prawdopodobieństwami), jak i regresyjnych.

WYKŁAD 4 – WARTOŚCI SHAPLEYA

Interpretacja w kontekście ML

📊 Interpretacja wartości Shapleya dla wartości cechy j:

Wartość j-tej cechy przyczynia się o φ_j do prognozy tego konkretnego przypadku w porównaniu z przeciętną prognozą dla zbioru danych.

Porównanie z średnią

Wartość odnosi się do przeciętnej prognozy dla całego zbioru danych

Średnia po kombinacjach

Średnia wartość cechy po uwzględnieniu wszystkich kombinacji innych cech

Marginalny wpływ

Miara wpływu cechy na prognozę względem oczekiwanej wartości bez tej cechy

WYKŁAD 4 – DEFINICJA WARTOŚCI SHAPLEYA

Wartość Shapleya jest używana w interpretowalności modeli uczenia maszynowego do określenia wkładu każdej zmiennej w prognozę modelu dla konkretnej obserwacji.

Definicja wartości Shapleya dla zmiennej:

Niech v będzie funkcją wartości, która określa prognozę modelu dla danego zestawu zmiennych. Wartość Shapleya ϕ_j dla zmiennej j jest zdefiniowana jako:

$$\phi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{j\}) - v(S)]$$

gdzie:

- N to zbiór wszystkich zmiennych.
- $|S|$ to liczba zmiennych w podzbiorze S.
- $|N|$ to całkowita liczba zmiennych.
- Suma jest obliczana dla wszystkich podzbiorów S zbioru N niezawierających zmiennej j.

Intuicyjnie, wartość Shapleya dla zmiennej j jest średnią wartością, jaką zmienna j dodaje do prognozy modelu, uwzględniając wszystkie możliwe kombinacje innych zmiennych.

WYKŁAD 4 – WARTOŚCI SHAPLEYA

Przykład zastosowania

Wartość Shapleya można stosować zarówno dla:

Modele klasyfikacyjne

Jeśli mamy do czynienia z prawdopodobieństwami

Przykład: Model PD w bankowości

Wartość Shapleya pokazuje wpływ wieku, dochodu itp. na prawdopodobieństwo defaultu

Modele regresyjne

Dla wartości ciągłych

Przykład: Model LGD w bankowości

Wartość Shapleya pokazuje wpływ typu zabezpieczenia na wysokość straty

⚠ **Ważne:** Wartości Shapleya są zawsze obliczane w kontekście konkretnej obserwacji. Dla globalnej interpretacji modelu należy agregować wartości Shapleya dla wielu obserwacji.

WYKŁAD 4 – WARTOŚCI SHAPLEYA

Przykład zastosowania w bankowości

Zastosowanie	Typ modelu	Częstotliwość	Metoda
Wyjaśnienie odmowy kredytu	PD (klasyfikacja)	Ad-hoc	SHAP values
Walidacja modelu IRB	PD, LGD, EAD	Roczna	Shapley + analiza stabilności
Monitoring fairness	Credit scoring	Kwartalna	Agregowane Shapley values
Feature engineering	Dowolny	Podczas rozwoju	SHAP + korelacje

WYKŁAD 4 – WARTOŚCI SHAPLEYA

Kiedy stosować wartości Shapleya

✓ Kiedy stosować

- ✓ **Walidacja modeli regulowanych**
  IRB models, IFRS 9 (wymogi KNF/EBA)
- ✓ **Wyjaśnienie pojedynczych decyzji**
  Dlaczego klient X otrzymał odmowę kredytu?
- ✓ **Audyt algorytmiczny**
  Compliance, fairness, bias detection
- ✓ **Model debugging**
  Znajdowanie nieoczekiwanych zależności
- ✓ **Feature importance lokalne**
  Zrozumienie wpływu cech na konkretną predykcję

✗ Kiedy NIE stosować

- ✗ **Real-time applications**
  Zbyt wolne obliczenia (ms vs sekundy)
- ✗ **Bardzo duże modele**
  1000+ cech - czas obliczeń rośnie wykładniczo
- ✗ **Jako jedyna metoda**
  Uzupełnij PDP, ICE, feature importance
- ✗ **Silnie skorelowane cechy**
  Może dawać nieintuicyjne wyniki
- ✗ **Brak danych treningowych**
  Potrzebne do obliczenia wartości bazowej

Praktyczna wskazówka: Zacznij od prostszych metod (feature importance, PDP), a wartości Shapleya stosuj gdy potrzebujesz dokładnego wyjaśnienia lub compliance wymaga tego.

WYKŁAD 4 – DEFINICJA WARTOŚCI SHAPLEYA

Wartość Shapleya można przedstawić w postaci całkowej, zwłaszcza gdy przestrzeń zmiennych jest ciągła lub gdy chcemy aproksymować sumę przez całkę w pewnych zastosowaniach. W kontekście uczenia maszynowego i interpretowalności modelu, całkowa forma wartości Shapleya jest używana w kontekście tzw. "kernel SHAP".

W postaci całkowej, wartość Shapleya dla zmiennej j w modelu z M zmiennymi można przedstawić jako:

$$\phi_j = \int_0^1 \left[v(S \cup \$j\$) - v(S) \right] \frac{d}{dt} \binom{M-1}{t(M-1)} dt$$

gdzie:

- $v(S)$ to wartość prognozy modelu dla podzbioru zmiennych S.
- $\binom{M-1}{t(M-1)}$ to współczynnik Newtona, który określa liczbę sposobów wyboru $t(M-1)$ zmiennych z $M-1$ dostępnych zmiennych (bez uwzględnienia zmiennej j).

WYKŁAD 4 – WAR. SHAPLEYA – MODELE LINIOWE

Dla modeli liniowych, obliczenie wartości Shapleya jest stosunkowo proste w porównaniu z bardziej złożonymi modelami, takimi jak sieci neuronowe czy lasy losowe.

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j)$$

Intuicyjnie, wartość Shapleya dla zmiennej w modelu liniowym mierzy, jak dużo ta zmienna odchyla się od średniej i jak wpływa to na prognozę modelu, uwzględniając wagę (współczynnik) przypisaną tej zmiennej w modelu.

$$\begin{aligned}\sum_{j=1}^{p} \phi_j(\hat{f}) &= \sum_{j=1}^{p} (\beta_j x_j - E(\beta_j X_j)) \\ &= (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} E(\beta_j X_j)) \\ &= \hat{f}(x) - E(\hat{f}(X))\end{aligned}$$

WYKŁAD 4 – WŁASNOŚCI WARTOŚCI SHAPLEYA

1. **Sprawiedliwość (Efficiency)**:

   - Suma wartości Shapleya wszystkich graczy (cech) jest równa całkowitej wartości gry.

$$\sum_{i \in N} \phi_i(v) = v(N)$$

2. **Symetria (Symmetry)**:

   - Jeśli dwóch graczy (cech) jest symetrycznych, tj. ich wkład do dowolnej koalicji jest taki sam, to ich wartości Shapleya są równe.

$$\text{Jeśli } v(S \cup i) = v(S \cup j) \text{ dla każdego } S, \text{ to } \phi_i(v) = \phi_j(v)$$

3. **Dodawanie wartości (Additivity)**:

   - Dla dowolnych dwóch gier (v) i (w), wartość Shapleya dla sumy tych gier jest równa sumie wartości Shapleya dla każdej z gier.

$$\phi_i(v + w) = \phi_i(v) + \phi_i(w)$$

4. **Zero dla graczy zerowych (Null player)**:

   - Jeśli gracz (cecha) nie wnosi żadnej wartości do żadnej koalicji, jego wartość Shapleya jest równa zero.

$$\text{Jeśli } v(S \cup i) = v(S) \text{ dla każdego } S, \text{ to } \phi_i(v) = 0$$

5. **Marginalność (Marginality)**:

   - Wartość Shapleya dla gracza (cechy) jest równa jego średniemu wkładowi marginalnemu we wszystkich możliwych koalicjach.

WYKŁAD 4 – ZALETY WARTOŚCI SHAPLEYA

1. **Teoretyczne podstawy:** Wartość Shapleya pochodzi z teorii gier kooperacyjnych i posiada silne teoretyczne podstawy. Jej własności, takie jak sprawiedliwość i symetria, czynią ją atrakcyjną miarą wkładu cech. Właściwie jedyna metoda posiadająca solidne podstawy teoretyczne.

2. **Jednoznaczność:** Dla danego modelu i obserwacji wartość Shapleya jest jednoznacznie określona, co oznacza, że nie ma innych rozkładów wartości spełniających jej własności.

3. **Konsystencja:** Jeśli model staje się bardziej zależny od pewnej cechy, wartość Shapleya tej cechy nie maleje.

4. **Działa dla dowolnego modelu, agnostyczna:** Wartość Shapleya może być obliczana dla dowolnego modelu uczenia maszynowego, od prostych modeli liniowych po skomplikowane sieci neuronowe.

5. **Rozkład wartości:** Wartość Shapleya pozwala na rozkład prognozy modelu na indywidualne wkłady cech, co ułatwia zrozumienie, które cechy miały największy wpływ na daną prognozę.

6. **Zgodność z prawem:** Jedna z niewielu metod dopuszczonych np. „ML for IRB models” (EBA/REP/2023/28).

WYKŁAD 4 – WADY WARTOŚCI SHAPLEYA

1. **Koszt obliczeniowy:** Obliczenie wartości Shapleya dla modeli o wielu cechach może być bardzo kosztowne obliczeniowo, ponieważ wymaga analizy wszystkich możliwych kombinacji cech.

2. **Złożoność interpretacji:** Dla modeli z dużą liczbą cech interpretacja wielu wartości Shapleya może być trudna, zwłaszcza dla osób nieznających się na analizie danych. Wyjaśnienia tworzone metodą wartości Shapleya zawsze używają wszystkich cech. Ludzie preferują selektywne wyjaśnienia, takie jak te produkowane przez LIME.

3. **Nie uwzględnia interakcji:** Chociaż wartość Shapleya uwzględnia wkład marginalny cechy, nie uwzględnia ona bezpośrednio interakcji między cechami, korelacji danych.

4. **Aproksymacje:** Ze względu na koszt obliczeniowy, w praktyce często stosuje się przybliżone metody obliczania wartości Shapleya, co może prowadzić do błędów.

WYKŁAD 4 –WARTOŚCI SHAPLEYA, INNE JĘZYKI

R

iml, fastshap

Julia

Shapley.jl

WYKŁAD 4 –SHAP VS WARTOŚCI SHALEYA

Lundberg and Lee (2017) SHAP (SHapley Additive exPlanations)

PODOBIEŃSTWA

Teoretyczne podstawy: SHAP bazuje na wartości Shapleya, która pochodzi z teorii gier kooperacyjnych. Oba podejścia mają na celu określenie "sprawiedliwego" rozkładu wartości (lub wpływu) wśród graczy (lub cech).

Własności: SHAP dziedziczy pewne własności wartości Shapleya, takie jak sprawiedliwość, symetria i dodawanie wartości.

WYKŁAD 4 –SHAP VS WARTOŚCI SHALEYA

Zastosowanie: Tradycyjna wartość Shapleya została opracowana w kontekście teorii gier, aby rozdzielić wartość (np. zysk) wśród graczy w kooperacyjnej grze. SHAP został specjalnie opracowany dla interpretowalności modeli uczenia maszynowego.

Optymalizacja obliczeń: Obliczenie dokładnej wartości Shapleya jest kosztowne obliczeniowo, zwłaszcza dla modeli z wieloma cechami. SHAP wprowadza różne optymalizacje i przybliżenia, aby uczynić obliczenia bardziej wykonalnymi dla złożonych modeli uczenia maszynowego.

Interakcje: SHAP może być używany do identyfikacji i ilustracji interakcji między cechami w modelu, co jest trudniejsze do osiągnięcia przy użyciu tradycyjnej wartości Shapleya.

Rozszerzenia: SHAP został rozszerzony, aby uwzględniać specyfikę różnych modeli uczenia maszynowego, takich jak drzewa decyzyjne, lasy losowe, modele gradient boosting i deep learning. Dzięki temu SHAP może być bardziej efektywny w obliczeniach dla tych modeli w porównaniu z tradycyjnym podejściem opartym na wartości Shapleya.

WYKŁAD 4 –SHAP VS WARTOŚCI SHALEYA

SHAP

- LinearSHAP – dla modeli liniowych np. logistic regression, regression, ElasticNet.
- TreeShap – efektywna metoda estymacji dla modeli opartych na drzewach.
- DeepSHAP – metoda estymacji dla modeli deep learning.
- KerneSHAP – alternatywa dla wartości Shapleya oparta na jądrze, inspirowana lokalnymi modelami zastępczymi (local surrogate models).

- Sampling - Monte Carlo dla ekstremalnie dużych zbiorów cech
- Wiele globalnych metod interpretacji opartych na agregacji wartości Shapleya.

WYKŁAD 4 –DEFINICJA SHAP

SHAP - **wyjaśnienie wartości Shapleya jest przedstawiane jako addytywna metoda atrybucji cech, model liniowy.**

Takie podejście łączy LIME (Local Interpretable Model-agnostic Explanations-aproksymacje lokalnymi modelami liniowymi) i wartości Shapley values.

WYKŁAD 4 –DEFINICJA SHAP

SHAP określa wyjaśnienia jako:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

gdzie

g – model wyjaśnienia

$z' \in \{0,1\}^M$ - wektor koalicji lub uproszczone cechy

M – maksymalny rozmiar koalicji

$\phi_j \in \mathbb{R}$ - atrybucja cechy dla cechy *j*

Jeśli *x'* jest wektorem składającym się z samych jedynek:

$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j$$

WYKŁAD 4 –WŁASNOŚCI SHAP

Braki danych. Brakująca cecha otrzymuje atrybucję o wartości zero.

Spójność

$$x'_j = 0 \Rightarrow \phi_j = 0$$

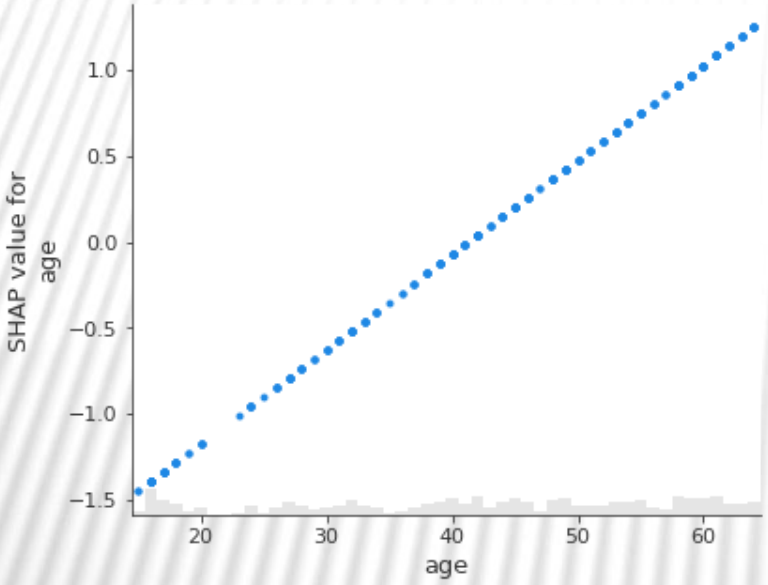
Let $\hat{f}_x(z') = \hat{f}(h_x(z'))$ and $z'_{\setminus j}$ indicate that $z'_j = 0$. For any two models f and f' that satisfy:

$$\hat{f}'_x(z') - \hat{f}'_x(z'_{\setminus j}) \geq \hat{f}_x(z') - \hat{f}_x(z'_{\setminus j})$$

for all inputs $z' \in \{0,1\}^M$, then:

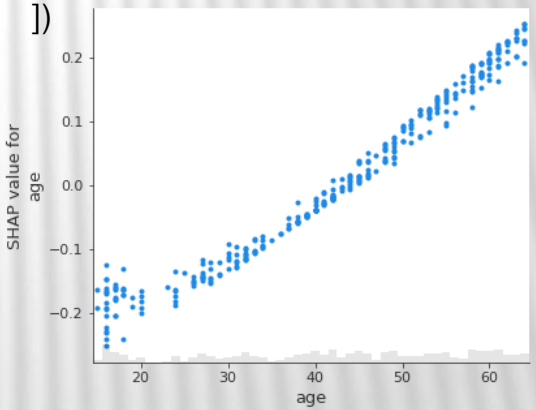
$$\phi_j(\hat{f}', x) \geq \phi_j(\hat{f}, x)$$

WYKŁAD 4 – SHAP (LR VS XGBOOST)

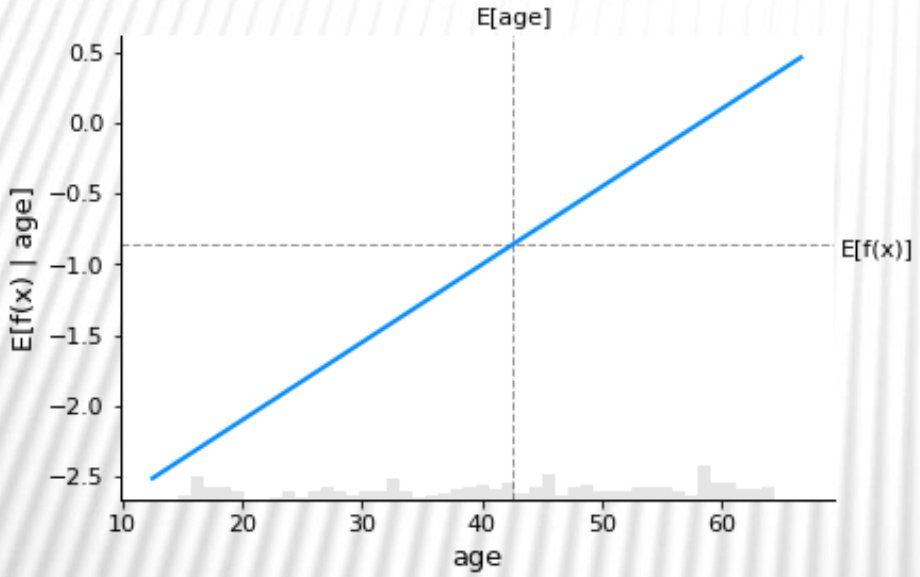


Data: Heart.csv
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/76SIQD

Python
shap.plots.scatter(shap_values[:,"age"])



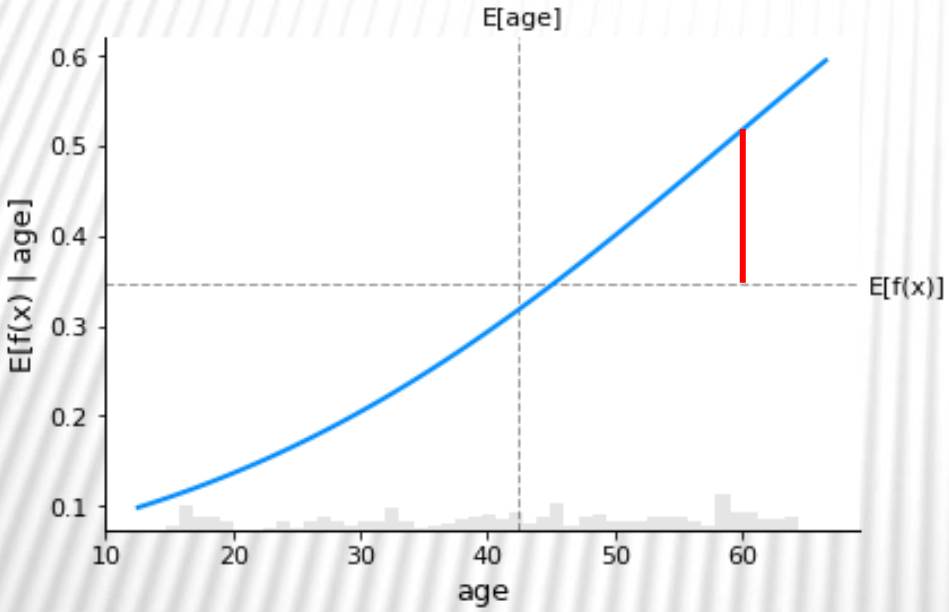
WYKŁAD 4 – PARTIAL DEPENDENCE PLOT – LOG(ODDS)



Python

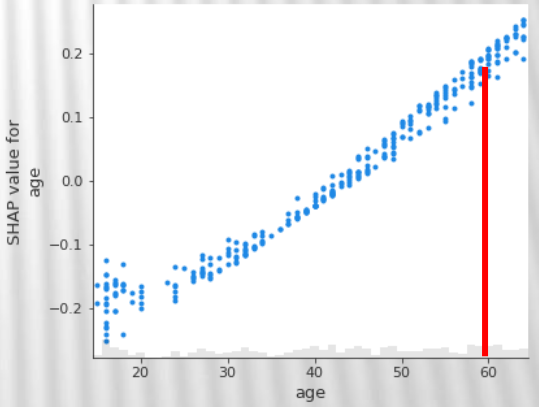
```
fig, ax = shap.partial_dependence_plot(
    "age", model_lr_log_odds, X_train,
model_expected_value=True,
    feature_expected_value=True,
show=False, ice=False)
```

WYKŁAD 4 – PARTIAL DEPENDENCE PLOT (LR)



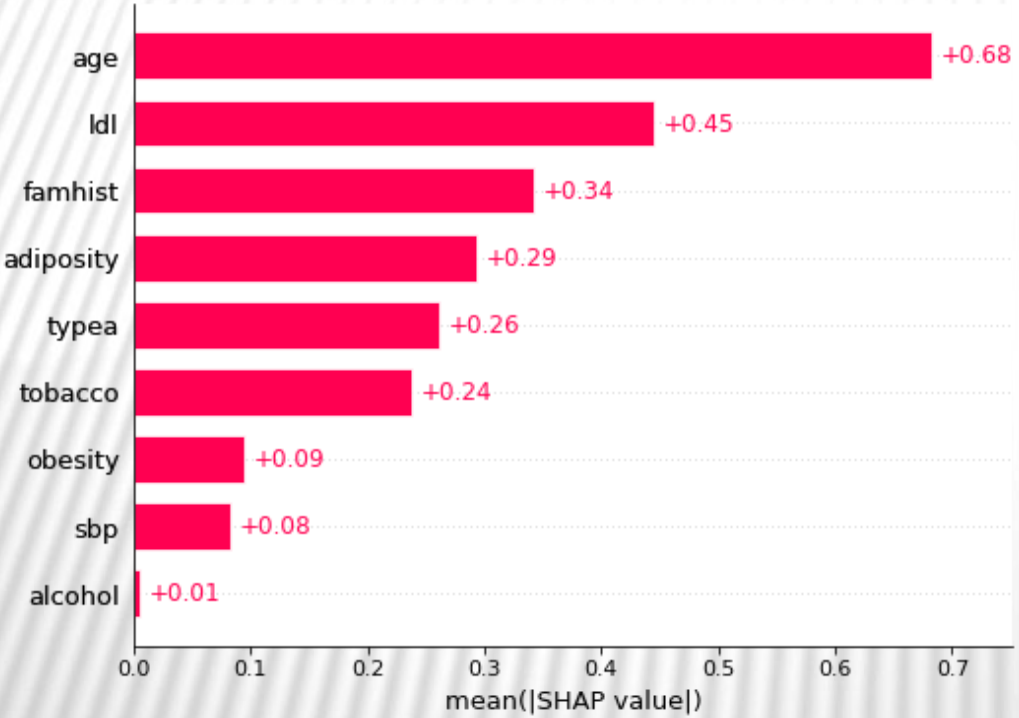
Python

```
fig, ax = shap.partial_dependence_plot(
    "age", model_lr_proba, X_train,
model_expected_value=True,
    feature_expected_value=True,
show=False, ice=False)
```



WYKŁAD 4

SHAP– BAR PLOT (LR)



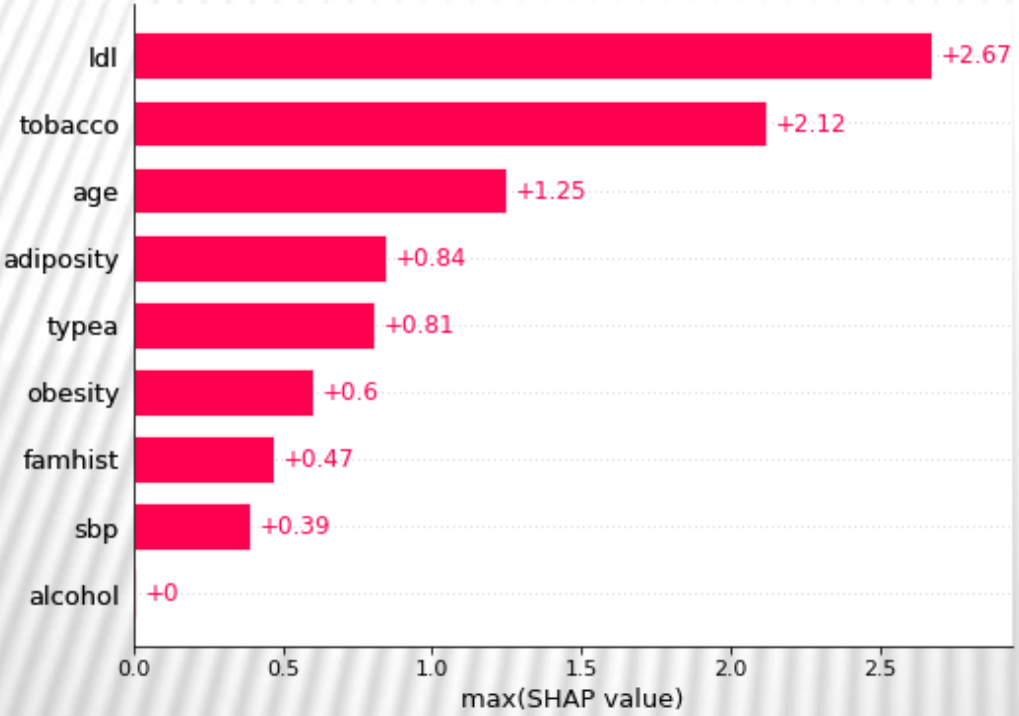
Python

shap.plots.bar(shap_values)
shap.summary_plot(shap_values)

By default a SHAP bar plot will take the mean absolute value of each feature over all the instances (rows) of the dataset.

WYKŁAD 4

SHAP– BAR PLOT (LR)



Python

```
shap.plots.bar(shap_values.max(
axis=0))
```

WYKŁAD 4

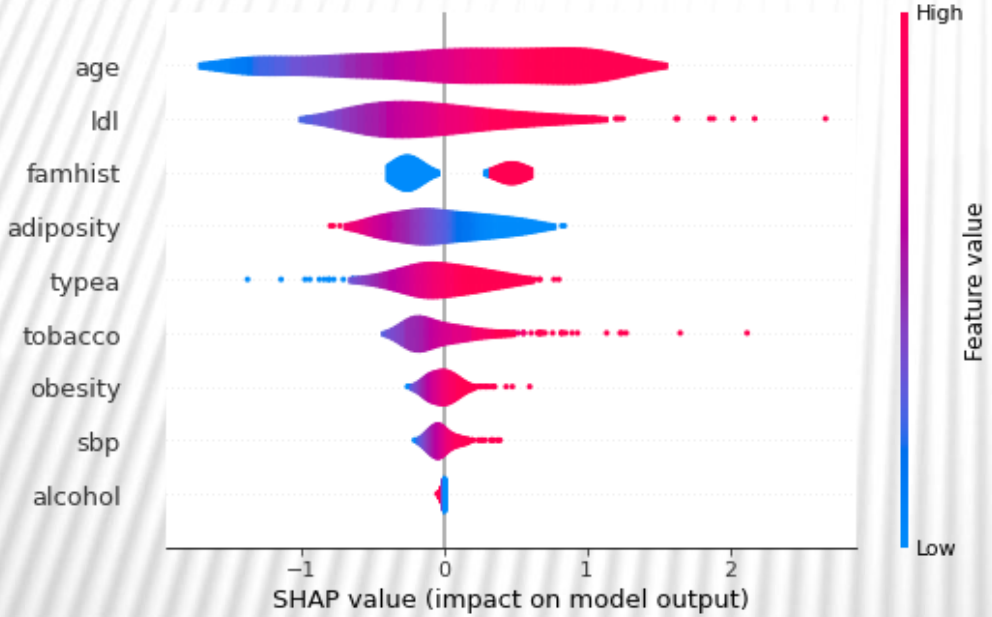
SHAP – SUMMARY PLOT: BEESWARM (LR)

Python

```
shap.summary_plot(shap_values)
shap.plots.beeswarm(shap_values)
```

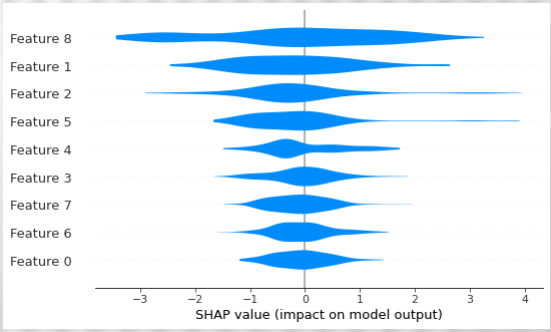
WYKŁAD 4

SHAP – SUMMARY PLOT: VIOLIN (LR)



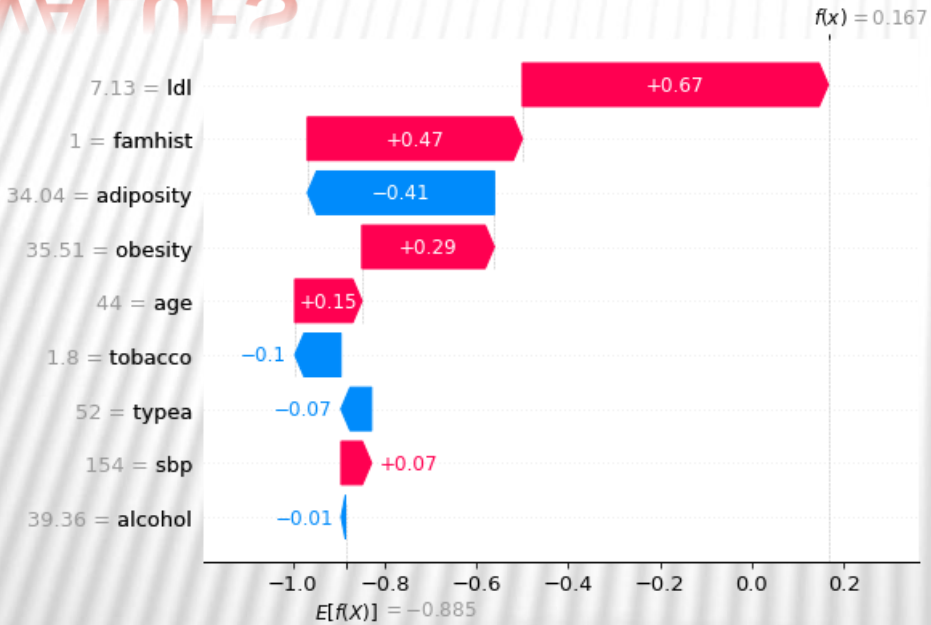
Python

```
shap.summary_plot(shap_values, plot_type='violin')
```



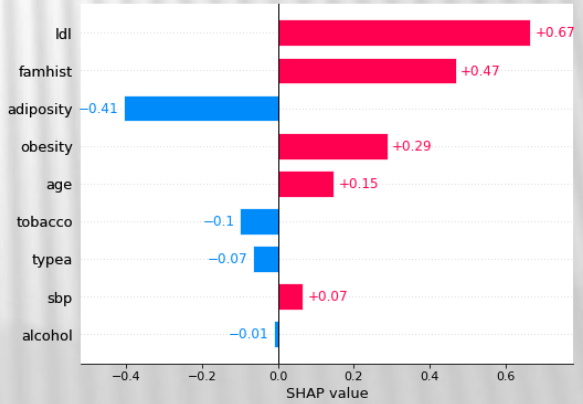
WYKŁAD 4

SHAP – WATERFALL PLOT (LR)
THE ADDITIVE NATURE OF SHAPLEY VALUES



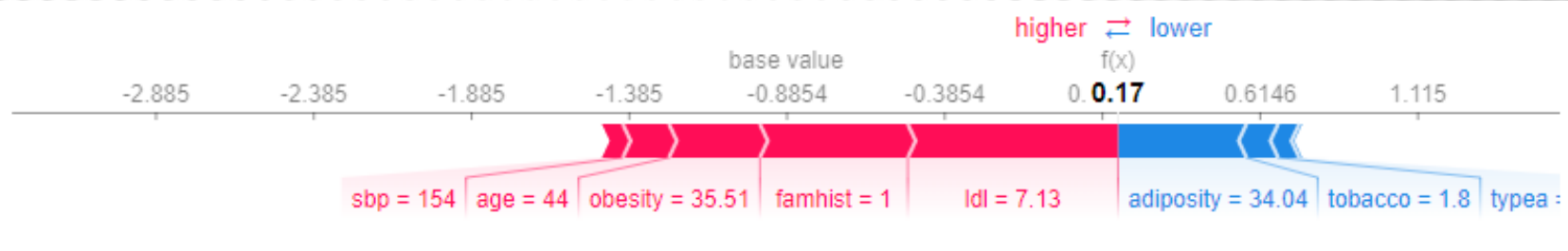
Python

```
shap.plots.waterfall(shap_values[ind])
shap.plots.bar(shap_values[ind])
```



WYKŁAD 4

SHAP – FORCE PLOT (LR)

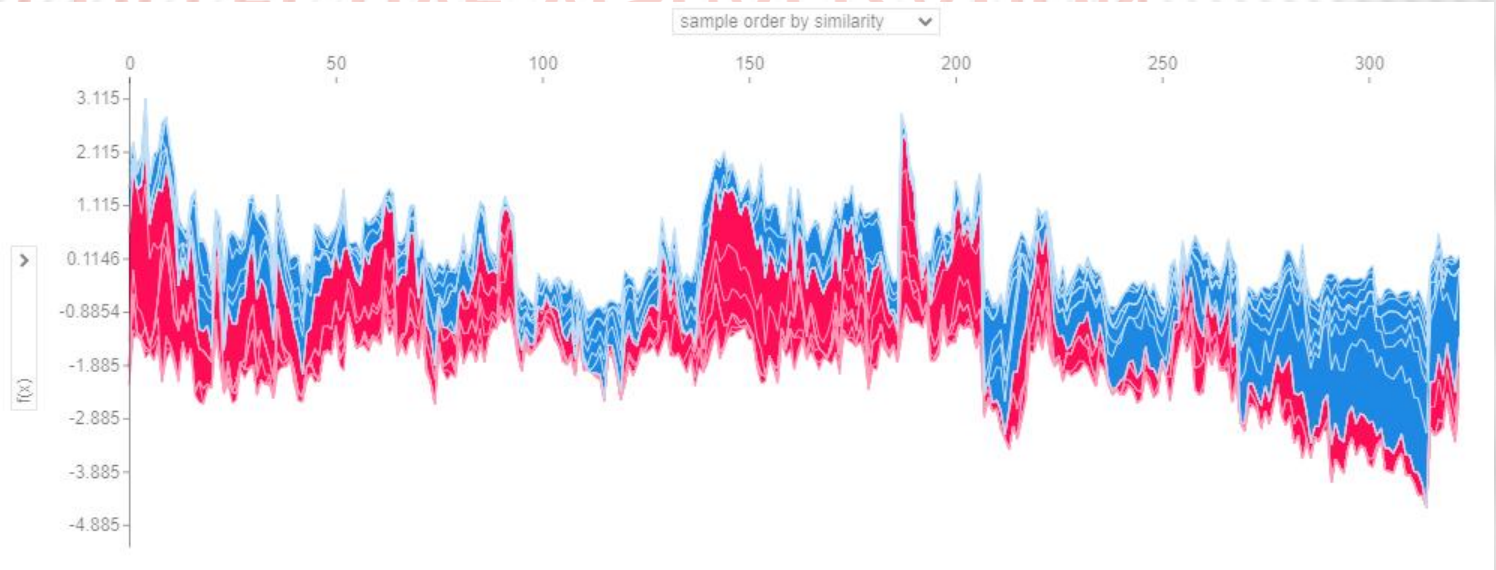


Python

shap.plots.force(shap_values[ind])

WYKŁAD 4

SHAP – FORCE PLOT (LR) COLLECTIVE VISUALIZATION

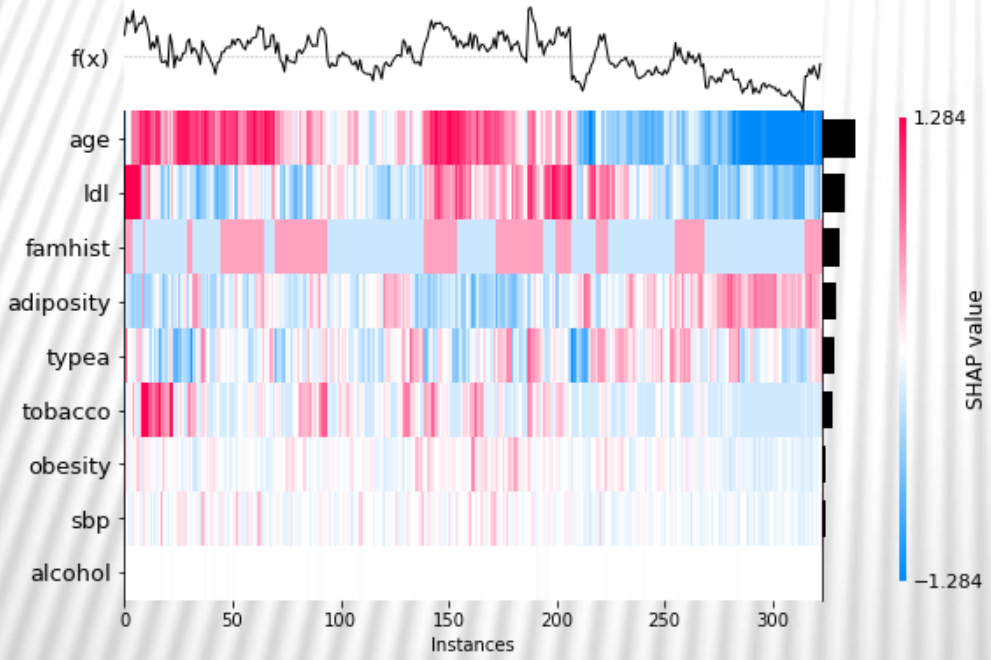


Python

shap.force_plot(explainer.expected_value, shap_values.values, X_train, feature_names = X_train.columns)

WYKŁAD 4

SHAP – HITMAP (LR) COLLECTIVE VISUALIZATION



Python

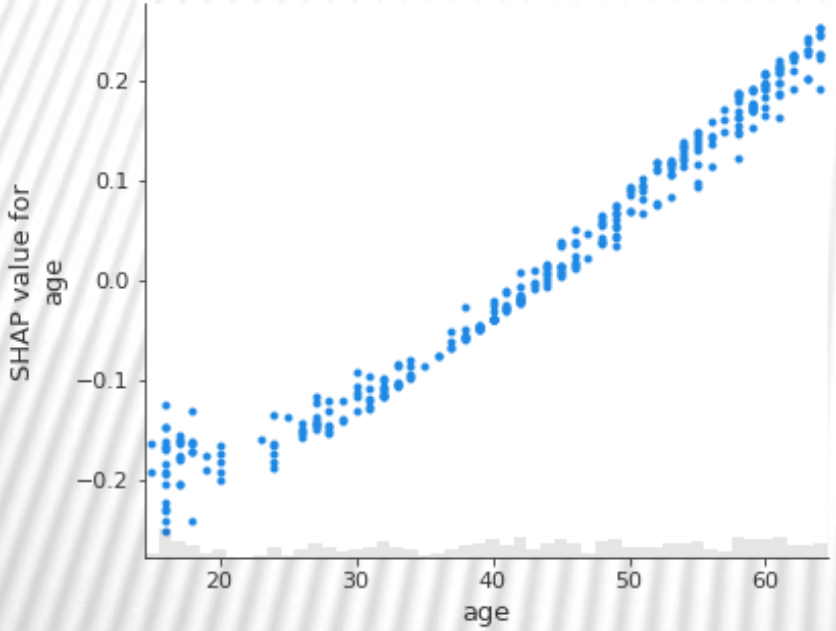
shap.plots.heatmap(shap_values)

WYKŁAD 4 – TREESHAP

- Lundberg et al. (2018) zaproponowali TreeSHAP, wariant SHAP dla modeli uczenia maszynowego opartych na drzewach, takich jak:
  - decision trees
  - random forests
  - gradient boosted trees
- TreeSHAP zostało wprowadzone jako szybka, specyficzna dla modelu alternatywa dla KernelSHAP, ale okazało się, że może ono generować nieintuicyjne atrybucje cech.

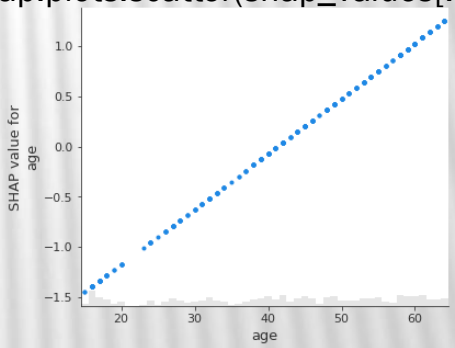
WYKŁAD 4

SHAP (XGBOOST VS LR)



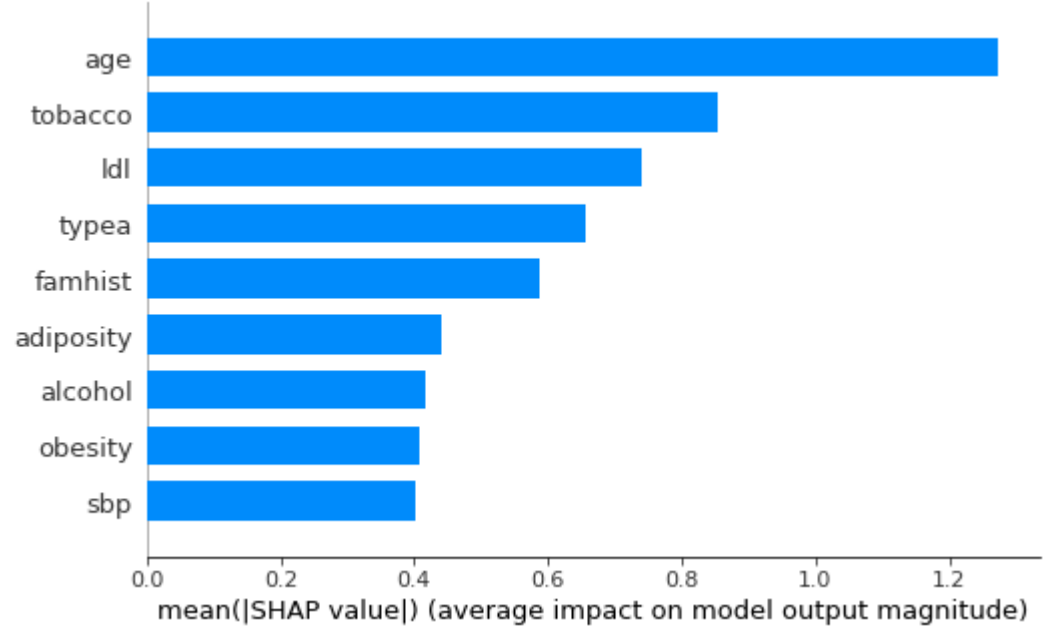
Data: Heart.csv
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/76SIQD

Python
shap.plots.scatter(shap_values[:,"age"])



WYKŁAD 4

SHAP – BAR PLOT (XGBOOST)

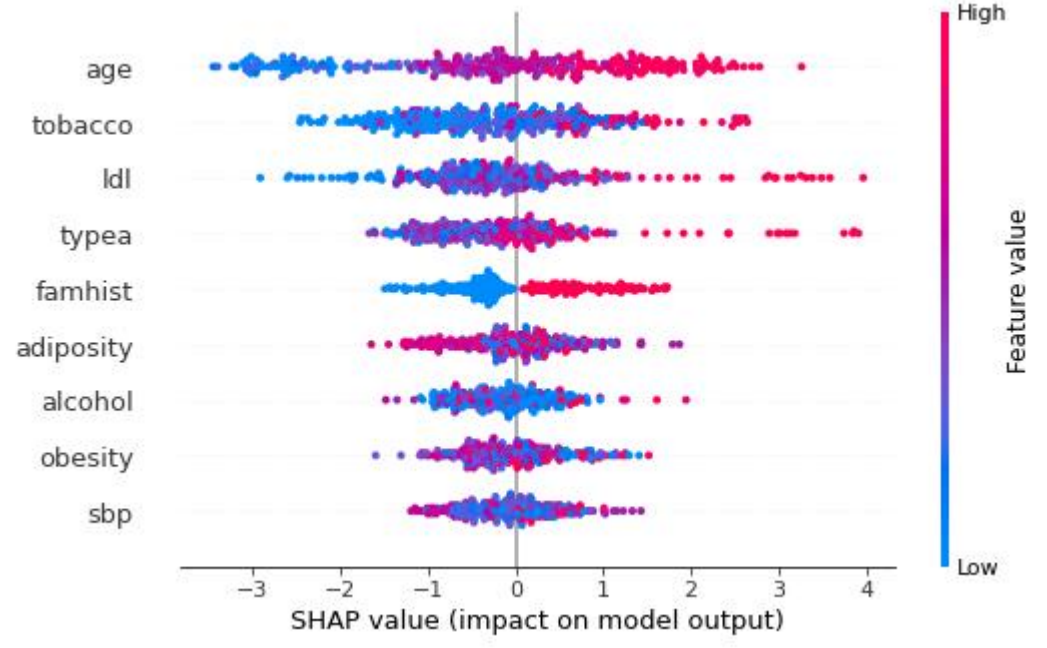


Python

shap.summary_plot(shap_values)

WYKŁAD 4

SHAP – SUMMARY PLOT: BEESWARM (XGBOOST)

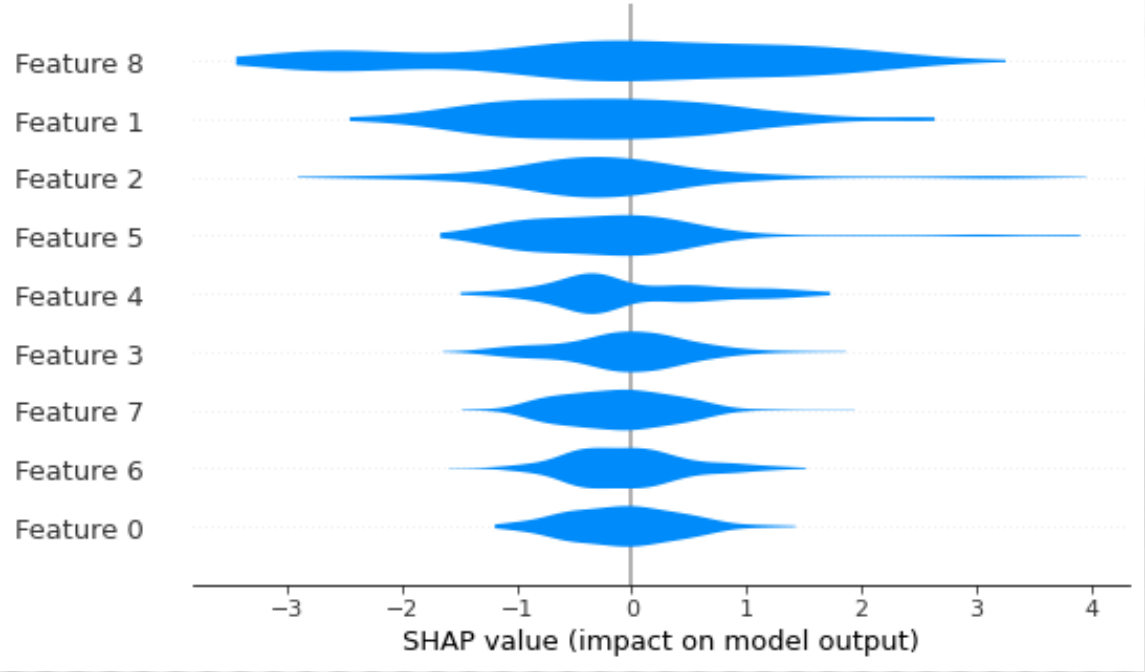


Python

```
shap.summary_plot(shap_values)
shap.plots.beeswarm(shap_values)
```

WYKŁAD 4

SHAP – SUMMARY PLOT: VIOLIN (XGBOOST)

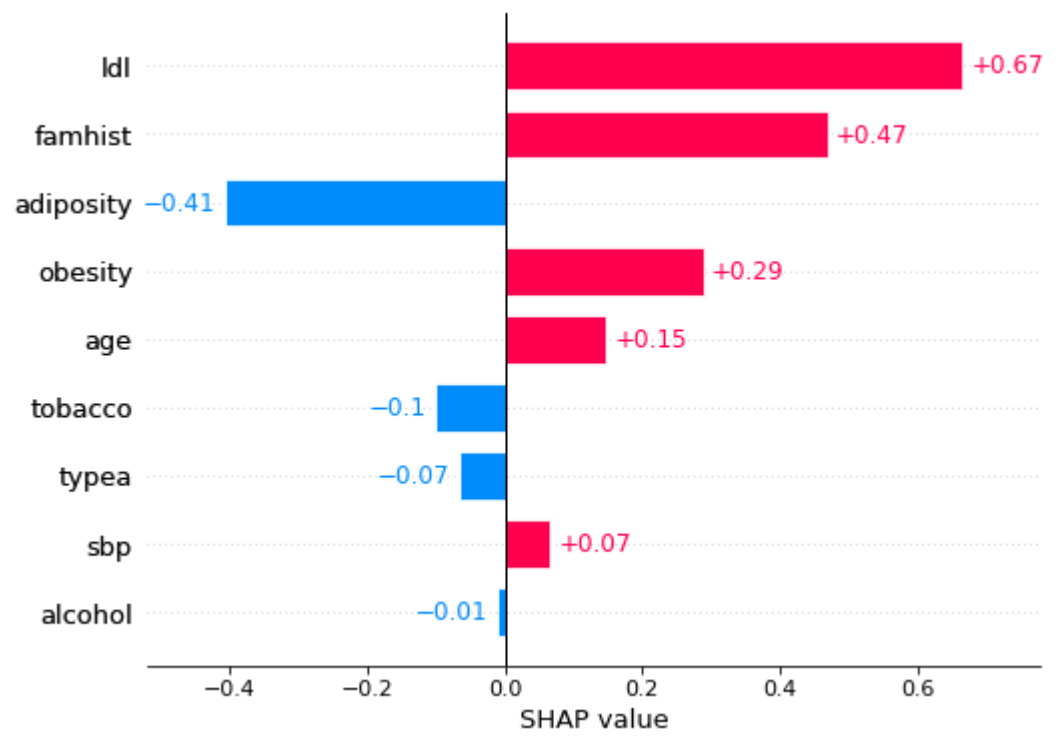


Python

```
shap.summary_plot(shap_values, plot_type='violin')
```

WYKŁAD 4

SHAP – LOCAL BAR PLOT (XGBOOST)

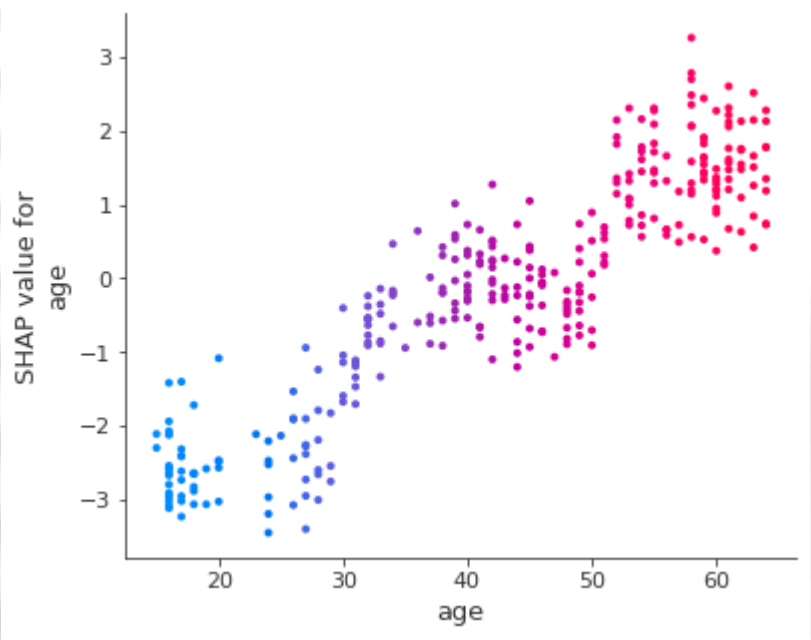


Python

shap.plots.bar(shap_values[0])

WYKŁAD 4

SHAP – PARTICAL DEPENDENCE (XGBOOST)

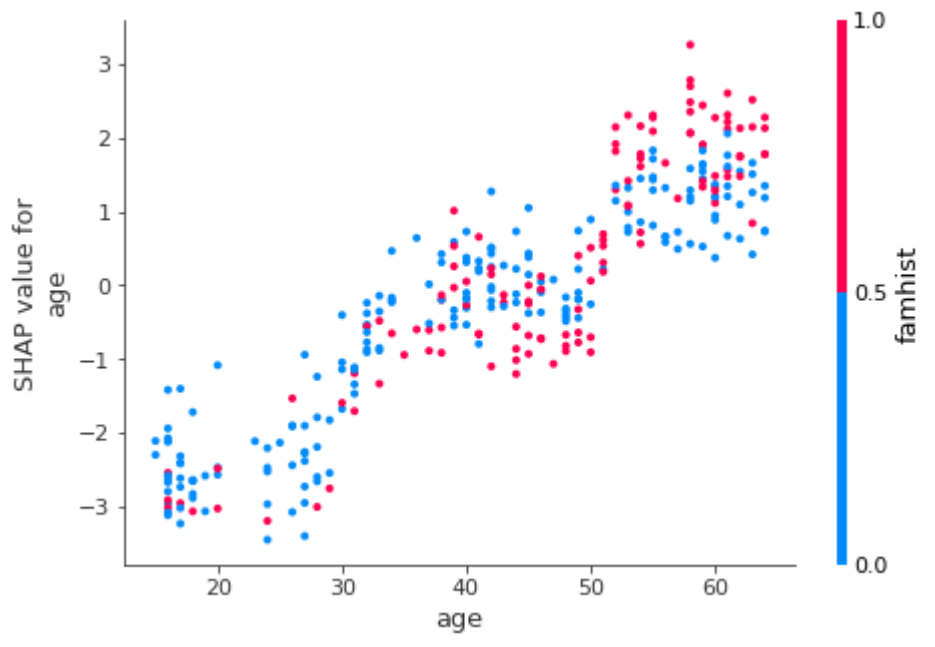


Python

```
shap.dependence_plot("age",
shap_values, X_train,
interaction_index=inds[i])
```

WYKŁAD 4

SHAP – PARTICAL DEPENDENCE (XGBOOST)



Python

```
shap.dependence_plot("age",
shap_values, X_train,
interaction_index=inds[i])
```

WYKŁAD 4 – SHAP ZALETY

- **Solidne podstawy teoretyczne**: SHAP opiera się na wartościach Shapleya z teorii gier kooperacyjnych, co zapewnia mu mocne teoretyczne podstawy. Wartości Shapleya są uważane za "sprawiedliwy" sposób rozdzielania wartości wśród graczy w grze.
- **Lokalna interpretowalność**: SHAP dostarcza lokalnych wyjaśnień dla indywidualnych prognoz, co pozwala użytkownikom zrozumieć, jakie cechy miały największy wpływ na konkretną prognozę.
- **Działanie model-agnostic**: SHAP jest metodą niezależną od modelu, co oznacza, że może być stosowany do różnych rodzajów modeli uczenia maszynowego, od prostych modeli liniowych po skomplikowane sieci neuronowe.
- **Spójność z LIME:** SHAP łączy idee z LIME (Local Interpretable Model-agnostic Explanations) i wartościami Shapleya, łącząc lokalne metody aproksymacji z teoretycznymi podstawami wartości Shapleya.
- **Szybkie implementacje dla konkretnych modeli:** Dla modeli opartych na drzewach, takich jak drzewa decyzyjne, lasy losowe czy modele gradient boosting, istnieje optymalizacja SHAP znana jako TreeSHAP, która pozwala na szybsze obliczenia.
- **Globalne i lokalne metody interpretacji:** Oprócz lokalnych wyjaśnień dla indywidualnych prognoz, SHAP oferuje również globalne metody interpretacji, takie jak ważność cech, zależność cech, interakcje i wykresy podsumowujące.
- **Interakcje między cechami:** SHAP może być używany do identyfikacji i ilustracji interakcji między cechami w modelu.
- **Dodawanie wartości:** Wartość SHAP dla danej cechy jest średnią wartością jej wkładu marginalnego we wszystkich możliwych kombinacjach cech, co czyni go intuicyjnym i łatwym do zrozumienia.

WYKŁAD 4 – SHAP WADY

- **Czas obliczeń dla KernelSHAP**: Obliczenie wartości SHAP, zwłaszcza przy użyciu KernelSHAP dla modeli z wieloma cechami, może być bardzo kosztowne obliczeniowo. Dla dużych zbiorów danych lub modeli z wieloma cechami może to być niepraktyczne.

- **Nieintuicyjne atrybucje w TreeSHAP**: Chociaż TreeSHAP został wprowadzony jako szybsza alternatywa dla KernelSHAP dla modeli opartych na drzewach, w pewnych sytuacjach może generować nieintuicyjne atrybucje cech.

- **Potencjalne błędne interpretacje**: Jak każde narzędzie do interpretowalności, SHAP może być źle interpretowany, zwłaszcza przez osoby nieznające się na analizie danych. Istnieje ryzyko, że użytkownicy mogą nadinterpretować lub źle zrozumieć wyniki.

- **Zależność od cech**: KernelSHAP może ignorować zależności między cechami, co może prowadzić do błędnych interpretacji, zwłaszcza gdy cechy są silnie skorelowane.

- **Złożoność dla użytkowników**: Dla niektórych użytkowników, zwłaszcza tych, którzy nie są zaznajomieni z teorią gier kooperacyjnych lub wartościami Shapleya, metoda SHAP może wydawać się skomplikowana i trudna do zrozumienia.

- **Wymagania dotyczące danych**: SHAP wymaga dostępu do danych treningowych, aby poprawnie obliczyć wartość bazową, co może być problematyczne w pewnych sytuacjach, zwłaszcza gdy dane są wrażliwe..

WYKŁAD 4 – SHAP W PYTHON I R

Python

Scikit-learn

R

shapper, fastshap

xgboost

ŹRÓDŁA

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems 30 (NIPS 2017) (pp. 4765-4774).

Shapley, L. S. (1953). A Value for n-Person Games. In H. W. Kuhn & A. W. Tucker (Eds.), Contributions to the Theory of Games (Vol. 2, pp. 307-317). Princeton University Press.

https://christophm.github.io/interpretable-ml-book/

https://shap.readthedocs.io/en/latest/overviews.html

AI NEWS

https://www.youtube.com/watch?v=lQHK61IDFH4

https://www.theaiopportunities.com/p/andrej-karpathy-breaks-down-the-2025