

Metody inteligencji obliczeniowej - Projekt

Temat: Analiza SHAP istotności poszczególnych elementów wektora wejściowego SSN.

Piotr Żeberek | 407663

1 Wstęp

1.1 Cel projektu

Celem projektu było zaproponowanie różnych architektur sieci neuronowych typu MLP dla kilku zbiorów danych z repozytorium <https://archive.ics.uci.edu/ml/index.php>. Należało dokonać oceny tych sieci oraz dokonać powtórnej analizy po usunięciu najmniej istotnych cech. Istotność cech określić należało przy użyciu analizy SHAP.

1.2 Uruchamianie projektu

Całość projektu składa się z czterech plików typu Jupyter Notebook, po jednym dla każdego zbioru danych. Biblioteki wymagane do uruchomienia projektu zawarto w pliku `requirements.txt`. W celu uruchomienia projektu zalecane jest użycie środowiska wirtualnego i zainstalowanie wymaganych bibliotek z wcześniej wspomnianego pliku.

Polecenia do stworzenia środowiska wirtualnego i instalacji bibliotek (platforma Linux):

```
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

2 Analizowane zbiory danych

2.1 Churn [1]

URL: <https://archive.ics.uci.edu/dataset/563/iranian+churn+dataset>
Odpływ klientów w telekomunikacji.

2.2 Student [2]

URL: <https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success>
Kończenie bądź porzucanie studiów przez studentów.

2.3 Bean [3]

URL: <https://archive.ics.uci.edu/dataset/602/dry+bean+dataset>
Klasyfikacja rodzajów fasoli na podstawie wyglądu ziaren.

2.4 Yeast [4]

URL: <https://archive.ics.uci.edu/dataset/110/yeast>
Lokalizacja białek w komórkach drożdży.

3 Architektury sieci

Do tworzenia SSN wykorzystano bibliotekę `scikit-learn` <https://scikit-learn.org/stable/>. W każdym przypadku zaproponowano pięć architektur sieci neuronowych typu MLP, różniące się liczbą ukrytych warstw oraz liczbą neuronów w tych warstwach, według schematu:

1. ()
2. (num_avg,)
3. (num_avg, num_avg)
4. (num_avg, num_outputs)
5. (num_features, num_avg, num_outputs)

gdzie:

- num_avg - średnia arytmetyczna liczby cech oraz klas dla danego zbioru danych,
- num_features - liczba cech w zbiorze danych,
- num_outputs - liczba klas w zbiorze danych.

4 Analiza SHAP

Analizę SHAP przeprowadzono dla ostatniej architektury sieci neuronowej, czyli (num_features, num_avg, num_outputs). Wykorzystano bibliotekę `shap` <https://shap.readthedocs.io/en/latest/>. Wartości SHAP obliczono dla zbioru testowego, gdzie tło stanowiła próbka o rozmiarze 100 z zbioru uczącego. Biblioteka zwraca te wartości w postaci trójwymiarowej tablicy o rozmiarze (n_samples, n_features, n_outputs), gdzie:

- n_samples - liczba próbek w zbiorze (u nas zbiór testowy),
- n_features - liczba cech,
- n_outputs - liczba klas wyjściowych.

Dla każdego zbioru danych zaprezentowano trzy wykresy będące wycinkami bądź agregacją danych z tablicy wartości SHAP:

- wykres typu `beeswarm` dla każdej klasy wyjściowej. Wycinek kodu:

```
for i, class_name in enumerate(model.classes_):
    per_class_explanation = shap.Explanation(
        shap_values[:, :, i], data=X_test, feature_names=X.columns
    )
    shap.plots.beeswarm(per_class_explanation, show=False)
```

- średnią z wartości bezwzględnych przyczynków do wartości SHAP od każdej cechy z podziałem na klasy wyjściowe. Wycinek kodu:

```
per_class_mean_abs_shap = {}

for i, class_name in enumerate(model.classes_):
    per_class_mean_abs_shap[class_name] = np.mean(np.abs(shap_values[:, :, i]), axis=0)
```

```

fig, ax = plt.subplots(figsize=(6, 4))

df = pd.DataFrame(per_class_mean_abs_shap)
df.plot.bar(ax=ax)

```

- średnią z wartości bezwzględnych przyczynków do wartości SHAP dla predykcji dokonanych przez model. Wycinek kodu:

```

preds = model.predict(X_test)

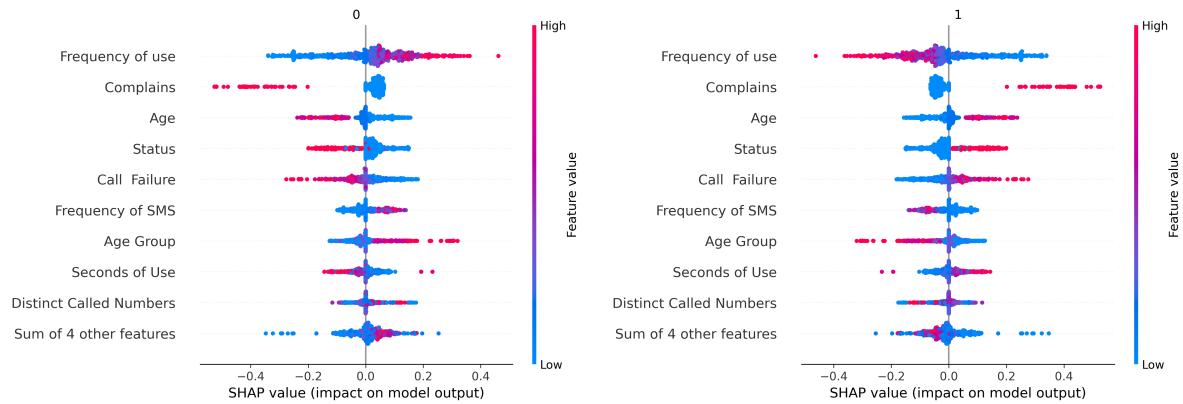
actual_prediction_shap_values = []

for i, pred in enumerate(preds):
    actual_prediction_shap_values.append(
        shap_values[i] [:, model.classes_.tolist().index(pred)])
)

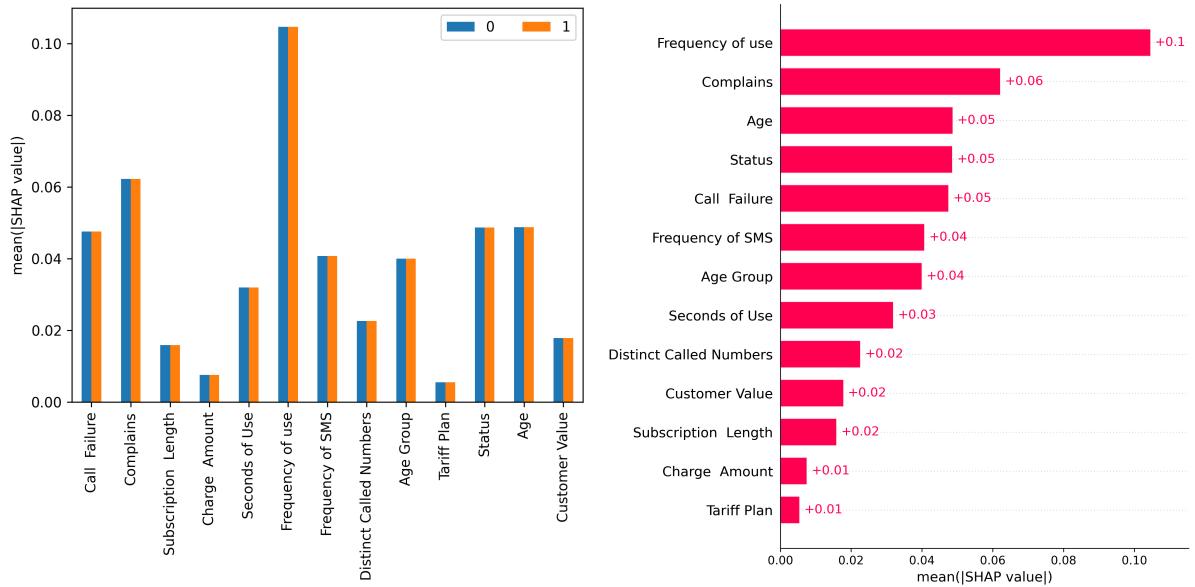
actual_prediction_shap_values = np.array(actual_prediction_shap_values)
explanation = shap.Explanation(actual_prediction_shap_values, feature_names=X.columns)
shap.plots.bar(explanation, show=False)

```

4.1 Churn



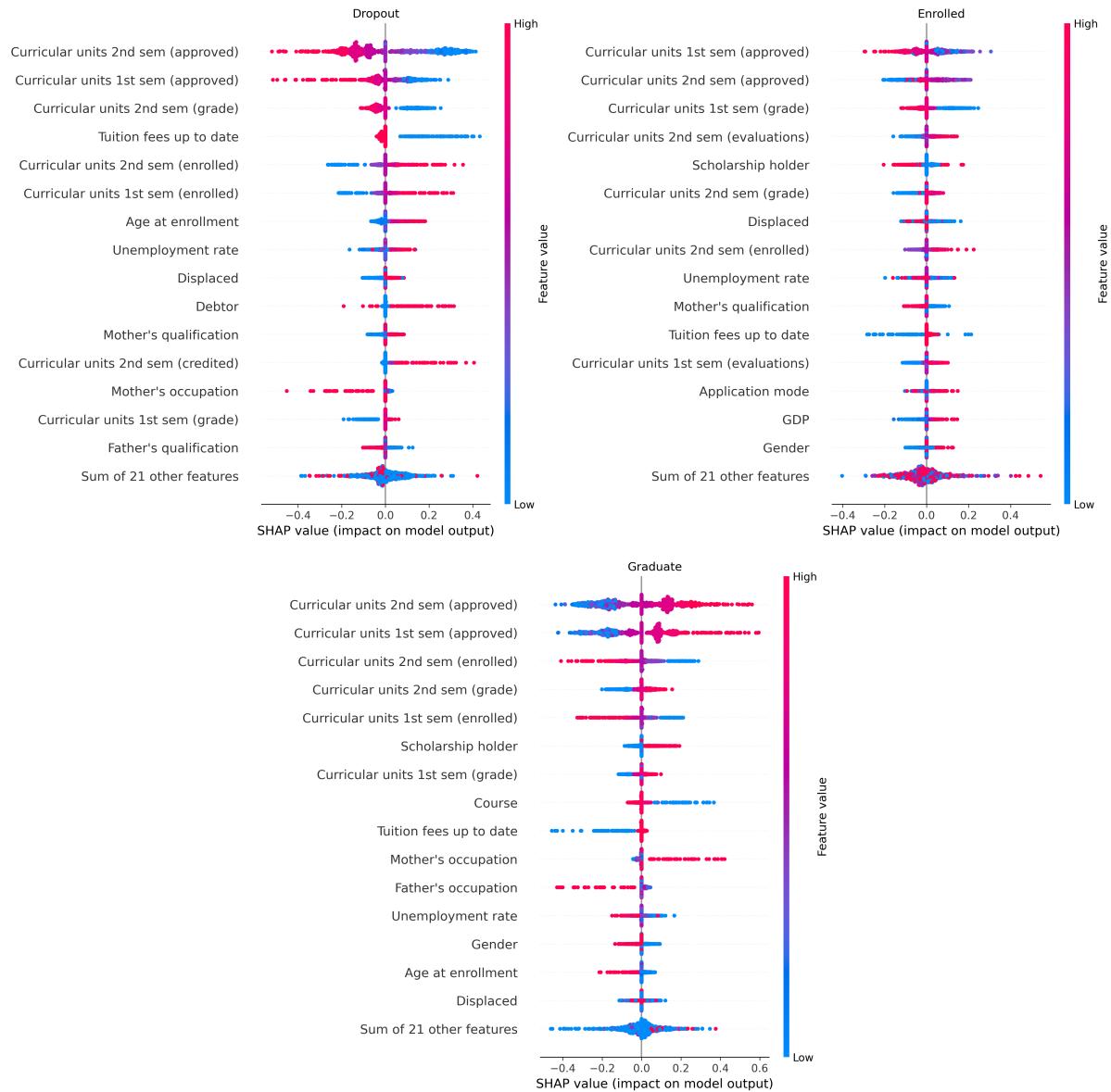
Rysunek 1: Wykres typu beeswarm dla klasy 0 (klient zostaje) i 1 (klient odchodzi) dla zbioru Churn.



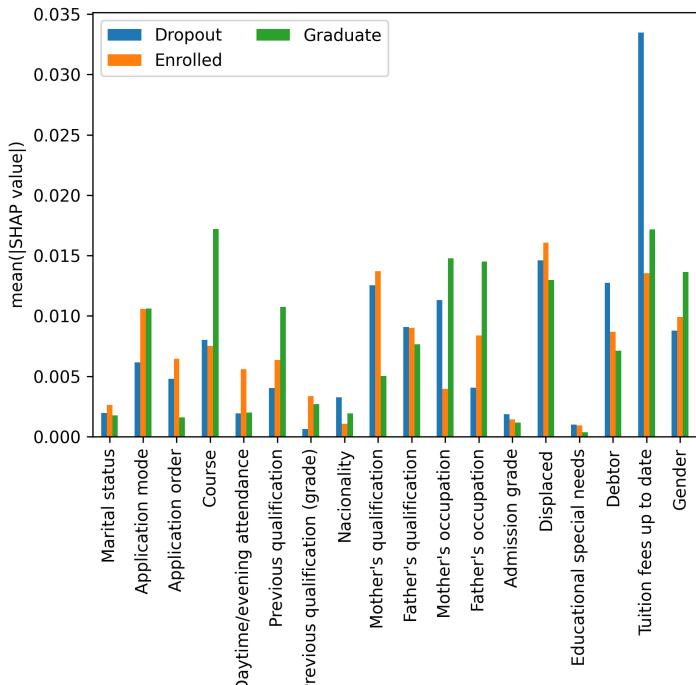
Rysunek 2: Średnia z wartości bezwzględnych przyczynków do wartości SHAP dla zbioru Churn. Przy czynki od cech dla każdej klasy wyjściowej (po lewej) oraz dla predykcji dokonanych przez model (po prawej).

Jako nieistotne uznano cechy "Charge Amount" oraz "Tariff Plan",

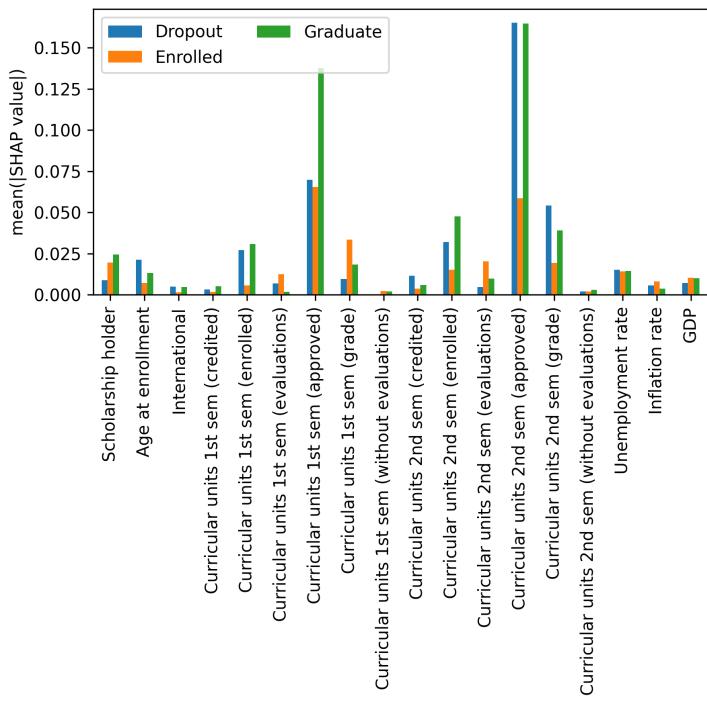
4.2 Student



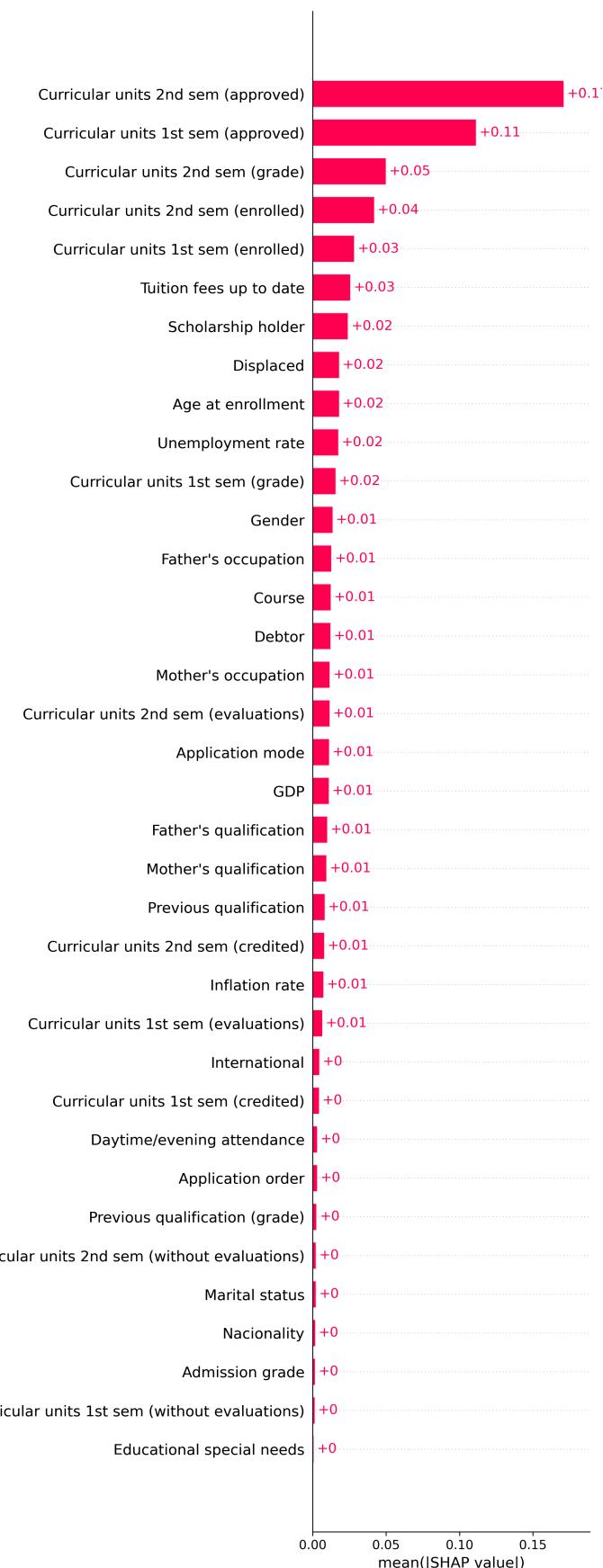
Rysunek 3: Wykres typu **beeswarm** dla klasy Dropout (porzucenie studiów), Enrolled (kontynuacja studiów) i Graduate (ukończenie studiów) dla zbioru Student.



Rysunek 4: Średnia z wartości bezwzględnych przyczynków do wartości SHAP dla zbioru Student. Przyczynki od cech dla każdej klasy wyjściowej (część 1).



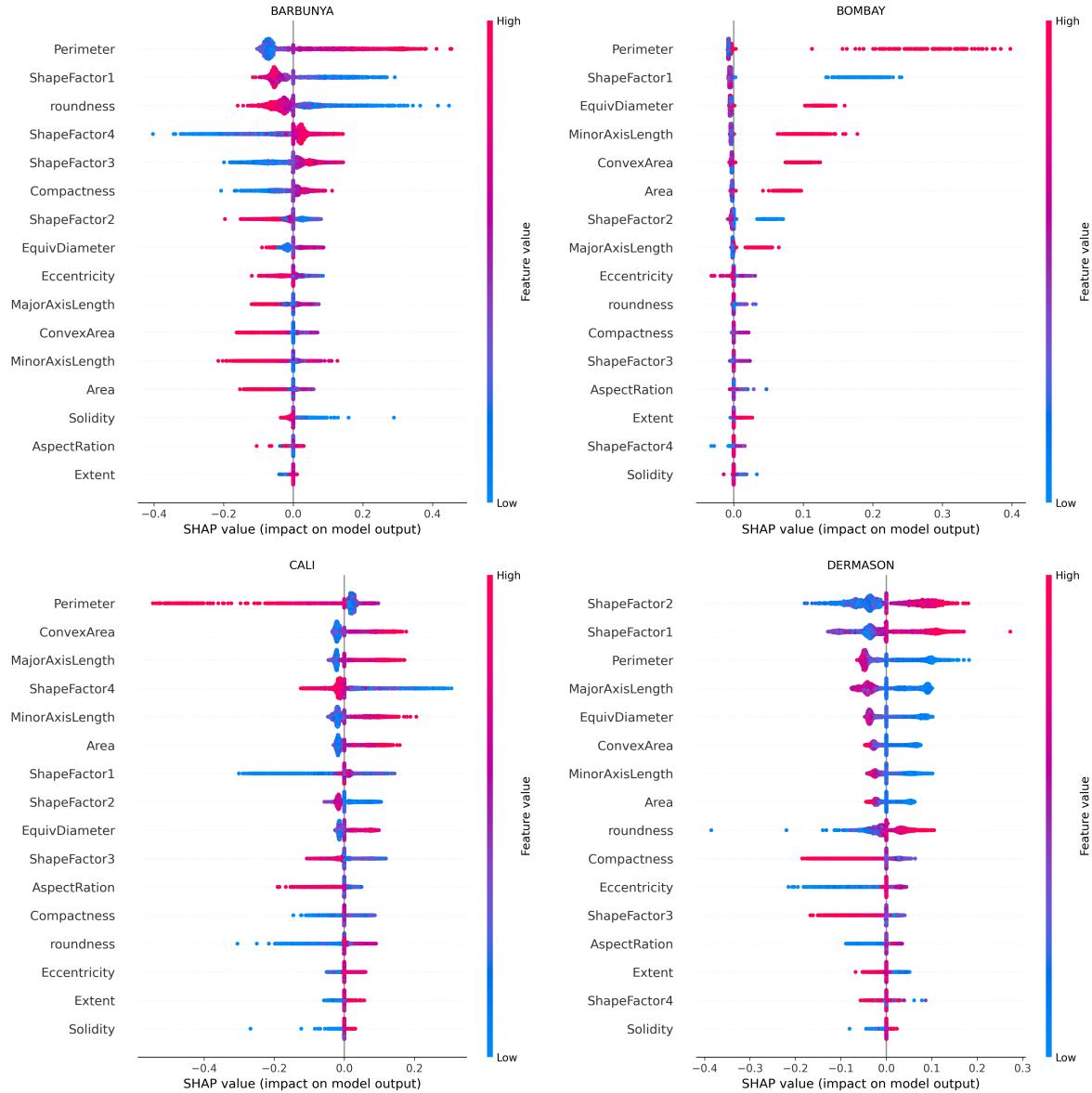
Rysunek 5: Średnia z wartości bezwzględnych przyczynków do wartości SHAP dla zbioru Student. Przyczynki od cech dla każdej klasy wyjściowej (część 2).



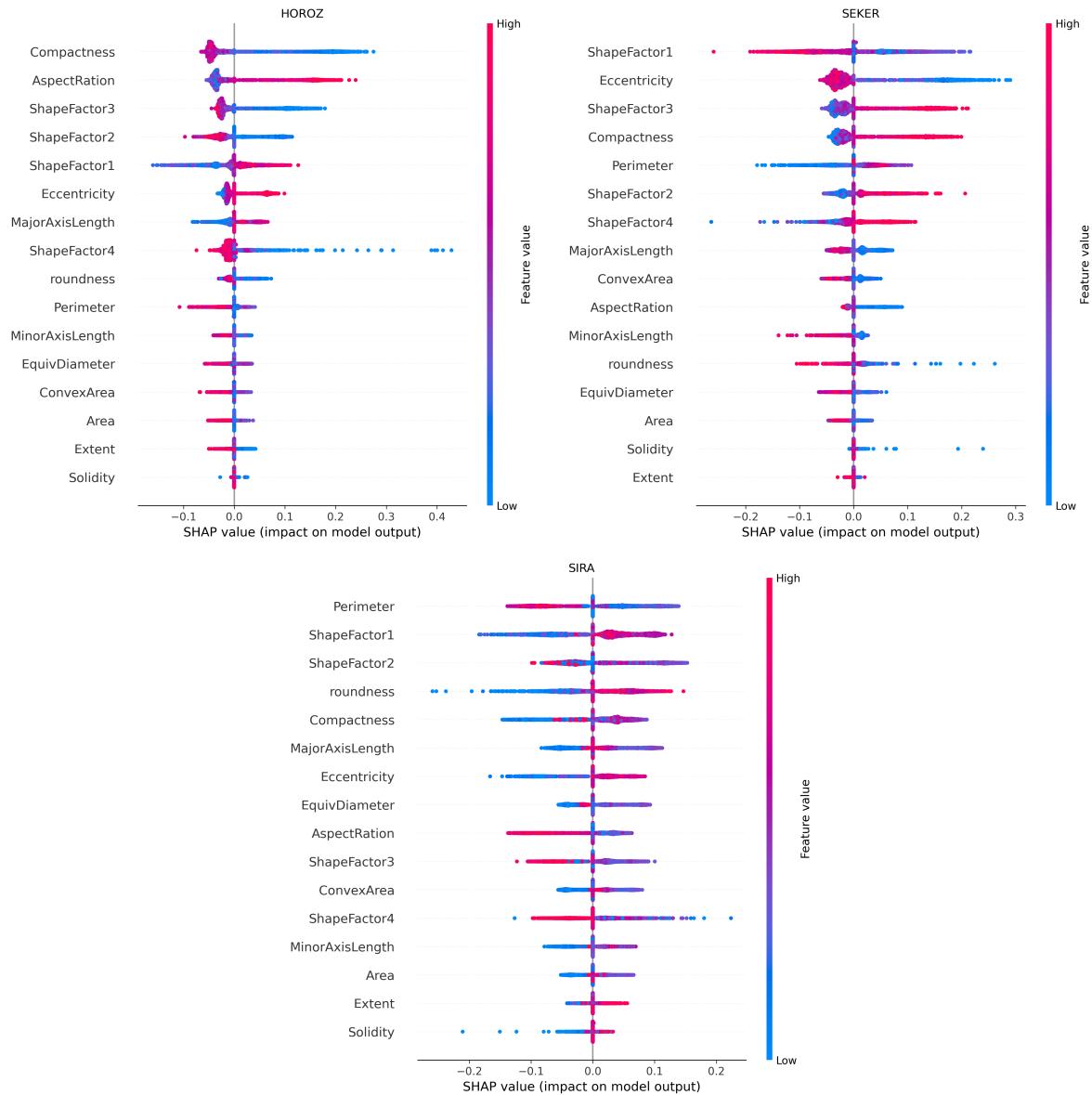
Rysunek 6: Średnia z wartości bezwzględnych przyczynków do wartości SHAP dla predykcji dokonanych przez model dla zbioru Student.

Jako nieistotne uznano cechy "International", "Curricular units 1st sem (credited)", "Day-time/evening attendance", "Application order", "Previous qualification (grade)", "Marital status", "Curricular units 2nd sem (without evaluations)", "Nacionality", "Admission grade", "Curricular units 1st sem (without evaluations)" oraz "Educational special needs".

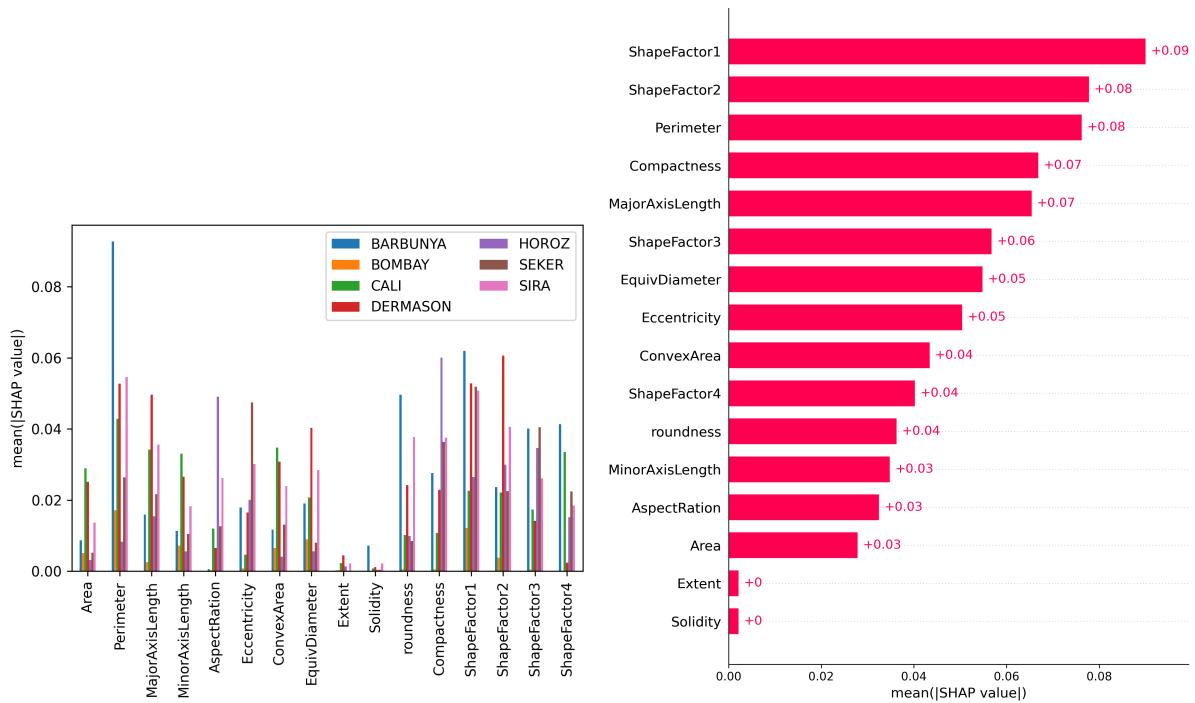
4.3 Bean



Rysunek 7: Wykres typu beeswarm dla klas wyjściowych w zbiorze Bean.



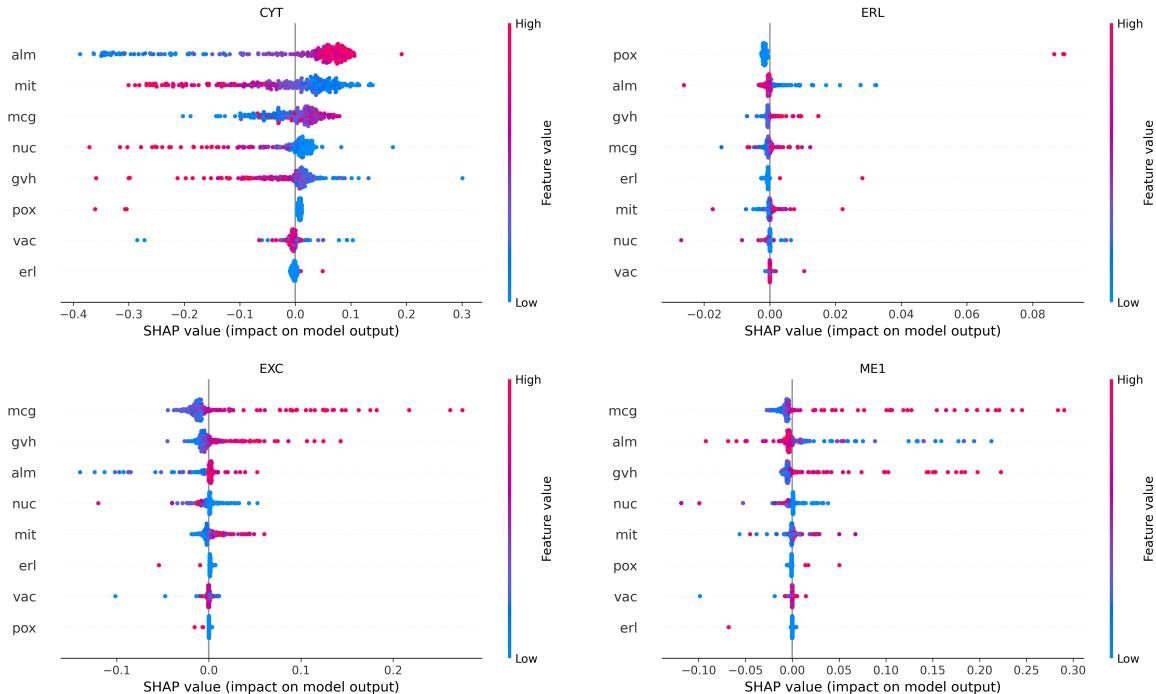
Rysunek 8: Wykres typu beeswarm dla klas wyjściowych w zbiorze Bean (ciąg dalszy).



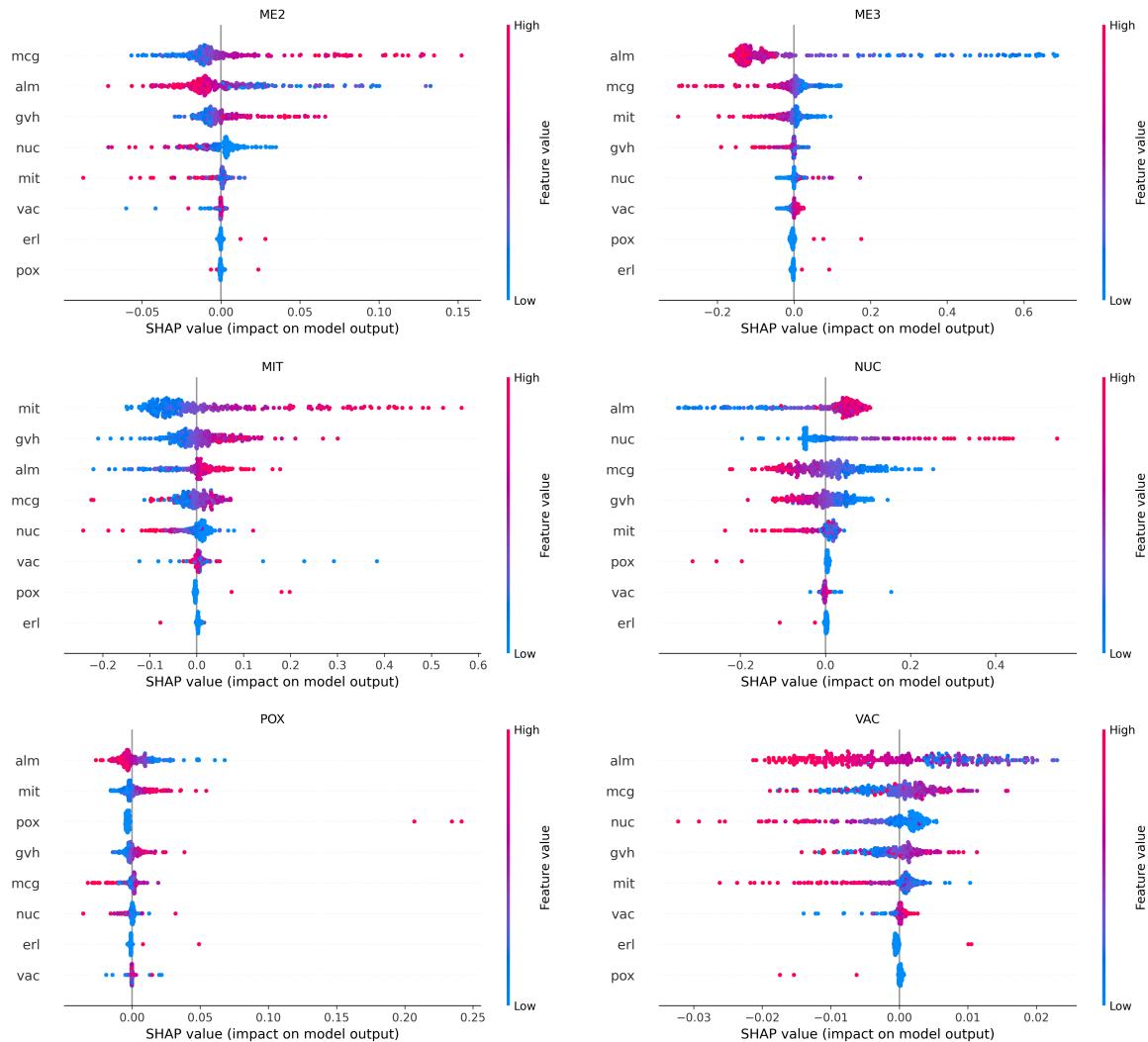
Rysunek 9: Średnia z wartości bezwzględnych przyczynków do wartości SHAP dla zbioru Bean. Przy czynki od cech dla każdej klasy wyjściowej (po lewej) oraz dla predykcji dokonanych przez model (po prawej).

Jako nieistotne uznano cechy "Extent" oraz "Solidity".

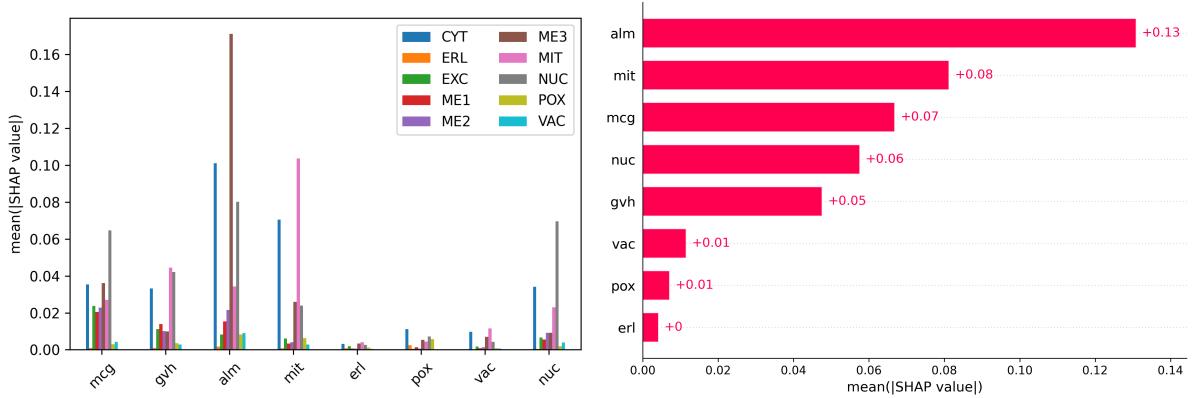
4.4 Yeast



Rysunek 10: Wykres typu beeswarm dla klas wyjściowych w zbiorze Yeast.



Rysunek 11: Wykres typu beeswarm dla klas wyjściowych w zbiorze Yeast.

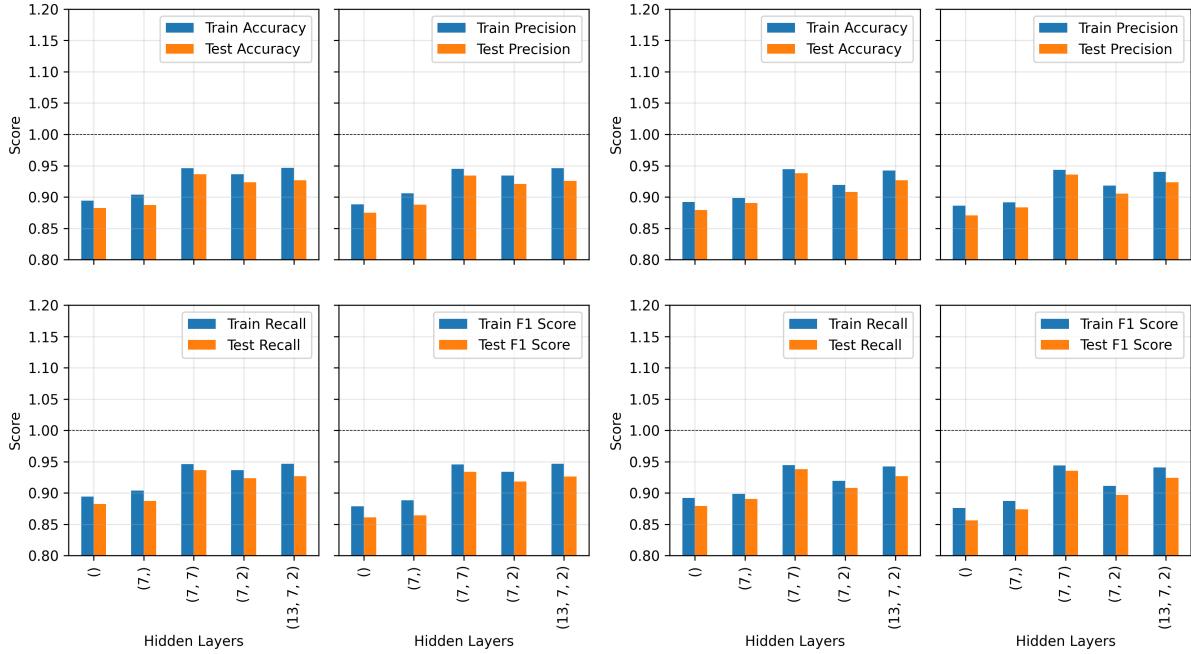


Rysunek 12: Średnia z wartości bezwzględnych przyczynków do wartości SHAP dla zbioru Yeast. Przyczynki od cech dla każdej klasy wyjściowej (po lewej) oraz dla predykcji dokonanych przez model (po prawej).

Jako nieistotne uznano cechy "vac", "pox" oraz "erl".

5 Ocena sieci

5.1 Churn



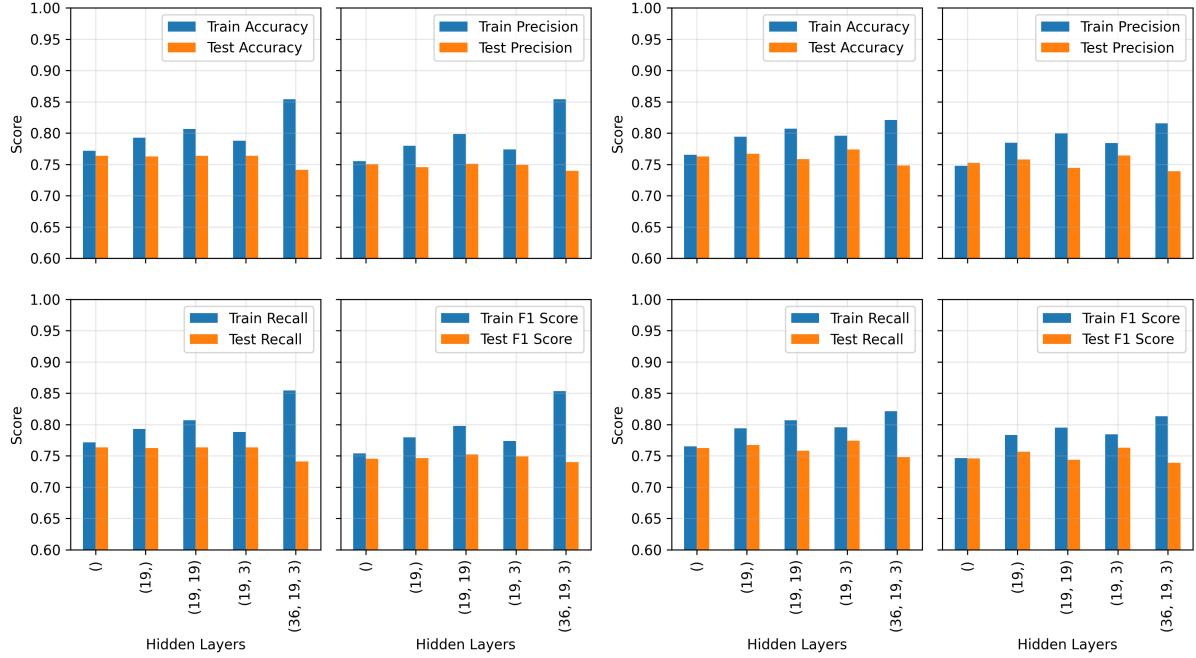
Rysunek 13: Wyniki klasyfikacji dla zbioru Churn. Po lewej stronie wyniki dla sieci z wszystkimi cechami, po prawej stronie wyniki dla sieci po usunięciu najmniej istotnych cech.

Tabela 1: Zgodność predykcji po usunięciu najmniej istotnych cech dla zbioru Churn.

Warstwa ukryta	Zgodność na zbiorze uczącym	Zgodność na zbiorze testowym
(0)	2515/2520 (99,80%)	628/630 (99,68%)
(7,)	2464/2520 (97,78%)	618/630 (98,10%)
(7, 7)	2440/2520 (96,83%)	607/630 (96,35%)
(7, 2)	2411/2520 (95,67%)	606/630 (96,19%)
(13, 7, 2)	2441/2520 (96.87%)	606/630 (96,19%)

Sieć bez warstw ukrytych oraz tylko z jedną warstwą ukrytą osiągnęła gorsze wyniki niż pozostałe architektury. Po usunięciu najmniej istotnych cech architektury (7, 7) oraz (13, 7, 2) poradziły sobie najlepiej. W najgorszym wypadku zgodność predykcji spadła o mniej niż 5%.

5.2 Student



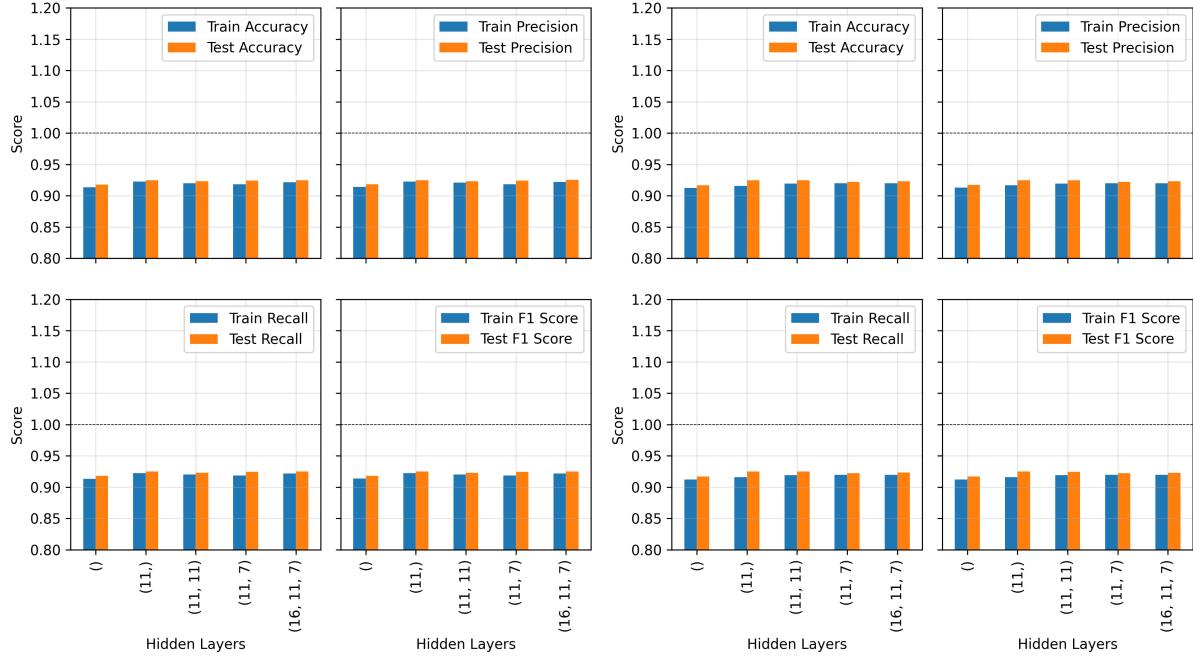
Rysunek 14: Wyniki klasyfikacji dla zbioru Student. Po lewej stronie wyniki dla sieci z wszystkimi cechami, po prawej stronie wyniki dla sieci po usunięciu najmniej istotnych cech.

Tabela 2: Zgodność predykcji po usunięciu najmniej istotnych cech dla zbioru Student.

Warstwa ukryta	Zgodność na zbiorze uczącym	Zgodność na zbiorze testowym
()	3407/3539 (96,27%)	860/885 (97,18%)
(19,)	3281/3539 (92,71%)	800/885 (90,40%)
(19, 19)	3203/3539 (90,51%)	790/885 (89,27%)
(19, 3)	3244/3539 (91,66%)	798/885 (90,17%)
(36, 19, 3)	3110/3539 (87,88%)	757/885 (85,54%)

Wyższa wartość metryk dla zbioru uczącego niż dla zbioru testowego może sugerować przeuczenie sieci, szczególnie w przypadku ostatniej architektury. Po usunięciu najmniej istotnych cech efekt ten wydaje się mniejszy, jednak nadal występuje. Zgodność predykcji była najgorsza dla najbardziej skomplikowanej architektury (36, 19, 3), gdzie spadła do poziomu ok. 86%.

5.3 Bean



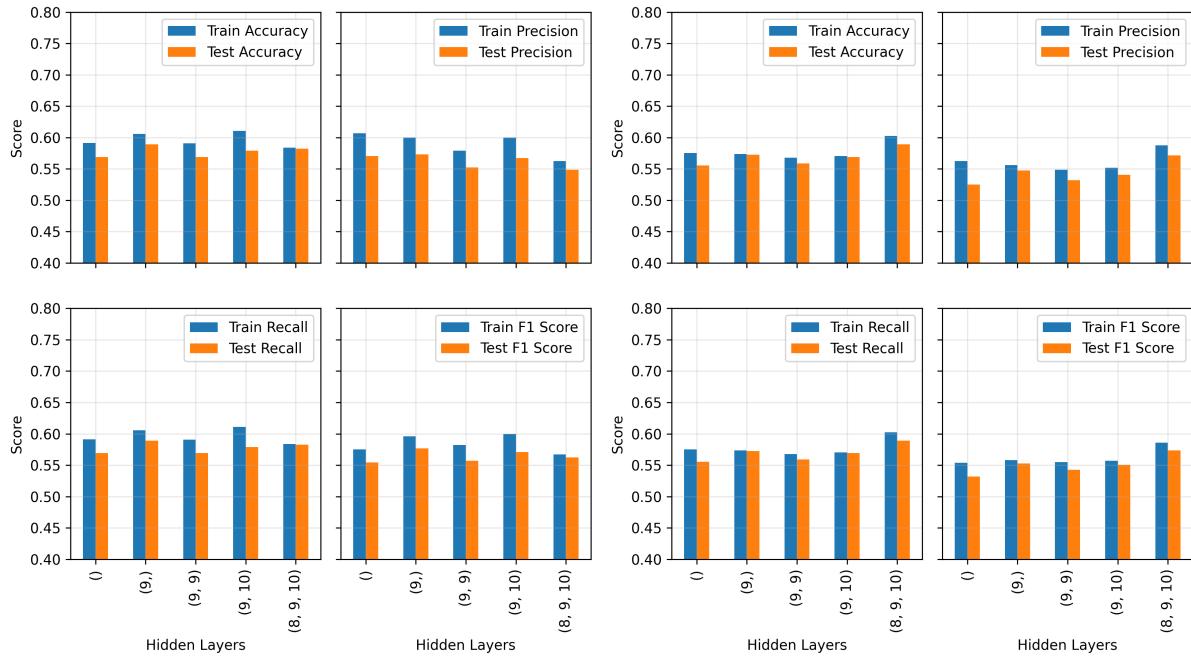
Rysunek 15: Wyniki klasyfikacji dla zbioru Bean. Po lewej stronie wyniki dla sieci z wszystkimi cechami, po prawej stronie wyniki dla sieci po usunięciu najmniej istotnych cech.

Tabela 3: Zgodność predykcji po usunięciu najmniej istotnych cech dla zbioru Bean.

Warstwa ukryta	Zgodność na zbiorze uczącym	Zgodność na zbiorze testowym
()	10772/10888 (98,93%)	2705/2723 (99,34%)
(11,)	10651/10888 (97,82%)	2675/2723 (98,24%)
(11, 11)	10627/10888 (97,60%)	2664/2723 (97,83%)
(11, 7)	10606/10888 (97,41%)	2658/2723 (97,61%)
(16, 11, 7)	10632/10888 (97,65%)	2668/2723 (97,98%)

Różnice w metrykach pomiędzy różnymi architekturami, także po usunięciu najmniej istotnych cech, są niewielkie, co może wskazywać na łatwy w analizie zbiór danych. Spadki zgodności predykcji wyniosły średnio trochę ponad 2%.

5.4 Yeast



Rysunek 16: Wyniki klasyfikacji dla zbioru Yeast. Po lewej stronie wyniki dla sieci z wszystkimi cechami, po prawej stronie wyniki dla sieci po usunięciu najmniej istotnych cech.

Tabela 4: Zgodność predykcji po usunięciu najmniej istotnych cech dla zbioru Yeast.

Warstwa ukryta	Zgodność na zbiorze uczącym	Zgodność na zbiorze testowym
(0)	1133/1187 (95,45%)	285/297 (95,96%)
(9,)	997/1187 (83,99%)	252/297 (84,85%)
(9, 9)	989/1187 (83,32%)	264/297 (88,89%)
(9, 10)	933/1187 (78,60%)	230/297 (77,44%)
(8, 9, 10)	966/1187 (81,38%)	249/297 (83,84%)

Usunięcie najmniej istotnych polepszyło wyniki ostatniej architektury w stosunku pozostałych. Spadki zgodności predykcji były największe ze wszystkich zbiorów i sięgały ok. 20%.

6 Podsumowanie

W projekcie zaproponowano różne architektury sieci neuronowych typu MLP dla czterech zbiorów danych - Churn, Student, Bean oraz Yeast. Dla każdego zbioru danych przeprowadzono analizę SHAP z użyciem modelu opartego o trzy warstwy ukryte. Na podstawie tej analizy stwierdzono, które cechy są najmniej istotne i usunięto je z danych wejściowych. Następnie porównano metryki klasyfikacji dla sieci z wszystkimi cechami oraz po usunięciu najmniej istotnych cech.

W przypadku zbiorów zbiorów Churn oraz Bean, gdzie metryki osiągały wysokie wartości rzędu 90% i więcej, usunięcie najmniej istotnych cech nie wpłynęło znacząco na metryki ani na zgodność predykcji. W przypadku zbioru Student oraz Yeast, gdzie metryki te były niższe, usunięcie najmniej istotnych cech spowodowało spadek zgodności predykcji tym większy, im gorsze były metryki.

Warto zauważyć, że spadek zgodności predykcji najmniej dotyczył sieci bez warstw ukrytych oraz fakt, że usunięcie najmniej istotnych cech wydaje się zmniejszać różnicę w metrykach pomiędzy zbiorami uczącymi i testowymi, co może sugerować unikanie przeuczenia sieci.

Bibliografia

- [1] *Iranian Churn*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5JW3Z>. 2020.
- [2] *Predict Students' Dropout and Academic Success*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5MC89>. 2021.
- [3] *Dry Bean*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C50S4B>. 2020.
- [4] Kenta Nakai. *Yeast*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5KG68>. 1991.