Sieci neuronowe

Przewidywanie poziomu dochodów na podstawie danych demograficznych

- **Autorzy:** Mateusz Hypta 280116, Mateusz Kwapisz 280107

- **Technologie:** Python, TensorFlow/Keras, Scikit-learn

Cel projektu i zbiór danych

Co chcemy osiągnąć?

Problem: Klasyfikacja, czy dana osoba zarabia powyżej 50 tys. USD rocznie (>50K) czy poniżej (<=50K).

Dane: Zbiór "Adult" zawierający ~32 tys rekordów.

Cechy:

- Demograficzne: wiek, rasa, płeć.
- Społeczne: edukacja, stan cywilny, relacje rodzinne.
- Zawodowe: profesja, branża, liczba godzin pracy tygodniowo.

Wyzwanie: Niezbalansowane klasy

Problem "Class Imbalance”

Wyzwanie: Niezbalansowane klasy

- **Dystrybucja danych:** Osoby zarabiające ≤50K stanowią ok. **76%** populacji, a >50K jedynie **24%**.
- **Zagrożenie:** Model może wpaść w pułapkę „Leniwego Klasyfikatora” – osiągnąć wysoką dokładność (**Accuracy ≈ 76%**) poprzez samo przypisywanie klasy większościowej, całkowicie ignorując mniejszość.

Zastosowane rozwiązanie: Random Oversampling

- **Zasada:** Technika zastosowana **wyłącznie na danych treningowych**, aby zapobiec faworyzowaniu większości przez sieć.
- **Mechanizm:** Losowe powielanie przykładów klasy mniejszościowej (>50K) do momentu osiągnięcia proporcji **1:1**.
- **Liczebność (Trening):**
    - Przed: ok. 22 600 vs ok. 7 500 (>50K).
    - Po: ok. 22 600 vs 22 600 (>50K).
- **Efekt:** Model widzi oba scenariusze z taką samą częstotliwością, co wymusza na sieci naukę realnych różnic demograficznych zamiast zgadywania

Architektura Sieci Neuronowej

Budowa modelu (MLP, 2 warstwy ukryte)

64 neurony: Wystarczająco dużo, by zrozumieć wszystkie cechy wejściowe, ale nie za dużo, by nie "zapchać" pamięci.
32 neurony: Kondensuje wiedzę – odrzuca szum, zostawia tylko fakty decydujące o zarobkach.
Dropout (0.3 - 0.4): Klucz do sukcesu. Bez niego model uczy się danych na pamięć (overfitting) i zawodzi w rzeczywistości.

Parametr	Najlepsza Wartość	Dlaczego ta wygrała?
Architektura	**[64, 32]**	**Najlepszy balans między precyzją a szybkością.**
Dropout	**0.3**	**Wyższy dropout lepiej chroni przed błędami na nowych danych.**
Learning Rate	**0.0005**	**Standardowa szybkość "Adam", która trafia w optimum.**
Metryka AUC	**~0.92**	**Oznacza, że model dobrze oddziela obie grupy.**

Metodologia: Eksperyment 300 prób

Optymalizacja poprzez iterację

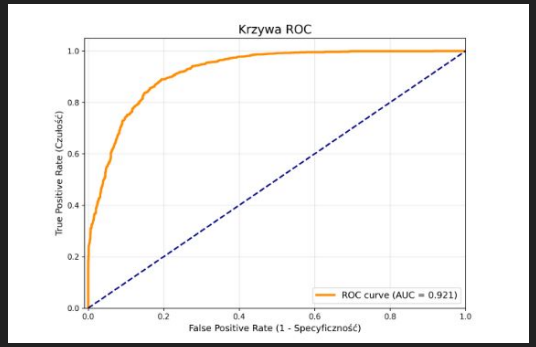
Optymalizacja poprzez iterację

- **Losowość startowa:** Sieci neuronowe rozpoczynają proces uczenia z losowymi wagami (inicjalizacja wag). Różny „punkt startu” sprawia, że każda próba może prowadzić do nieco innego wyniku końcowego.

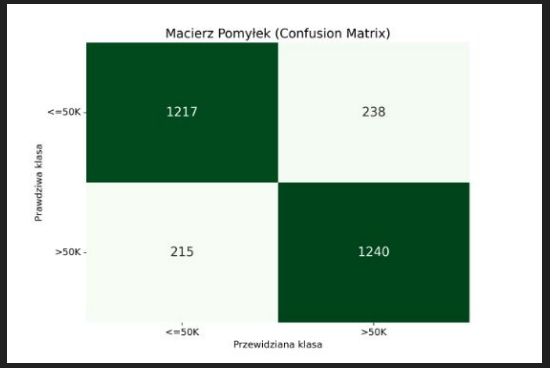
- **Szukanie optimum:** Przeprowadzenie łącznie **300 treningów** (12 kombinacji × 25 powtórzeń każda) pozwala uniknąć „pułapek” w postaci minimów lokalnych i precyzyjnie wyłonić najlepszą architekturę.

- **Statystyczna pewność:** Wybieramy model, którego wysoka skuteczność (AUC =0.9211) nie jest dziełem przypadku, lecz stabilnym wynikiem udowodnionym wielokrotnym testem, co eliminuje wpływ losowej inicjalizacji.

Analiza Wyników



Skuteczność najlepszego modelu



Accuracy: ~85% (ogólna poprawność).

Precyzja vs Recall:

- Model świetnie radzi sobie z klasą <=50K.
- Dzięki wagom/oversamplingowi, **Recall dla klasy >50K** (wykrywalność bogatszych osób) wzrósł do wysokiego poziomu (~65%).
- Precyzja dla klasy >50K: **~76%**

Krzywa ROC-AUC: Wysoka wartość pola pod krzywą świadczy o dobrej separacji klas przez model.

Wnioski i praktyczne zastosowanie

Główne Wnioski

- **Skuteczność modelu:** Uzyskana ogólna dokładność (**Accuracy**) na poziomie **84.43%** oraz bardzo wysoki współczynnik **AUC (0.9211)** potwierdzają, że sieci neuronowe MLP doskonale radzą sobie z nieliniowymi zależnościami w danych demograficznych.

- **Klucz do sukcesu – Balans:** Dzięki technice **Oversampling** wyeliminowano ryzyko „Leniwego Klasyfikatora", który ignorował by mniejszość. Model zyskał realną zdolność wykrywania wysokich dochodów, osiągając **Recall** na poziomie **85.22%**.

- **Stabilność i Precyzja:** Proces optymalizacji (**300 treningów**) pozwolił na uzyskanie wysokiej precyzji (**83.90%**) dla klasy >50K. Model wykazuje unikalną symetrię błędów (**238** Fałszywie Pozytywnych vs **215** Fałszywie Negatywnych), co udowadnia, że nie faworyzuje on żadnej z grup.