

Poszukiwanie mikrosoczewkowanych kwazarów w katalogu Gaia Data Release 3

Jeremi Borkowski Yakiv Ovcharenko
Karolina Bednarczyk Jan Łazeba Sebastian Chudek
Łukasz Wyrzykowski (opiekun projektu)

5 lipca 2026

1 Wstęp

Soczewkowanie grawitacyjne jest jednym z najważniejszych zjawisk przewidzianych przez ogólną teorię względności. Polega ono na ugięciu promieni świetlnych przez masywny obiekt znajdujący się między obserwatorem a źródłem światła. W konsekwencji obserwator może rejestrować wiele obrazów tego samego źródła, zmianę jego jasności lub jego pozorne przesunięcie na niebie. W przypadku, gdy masa ciała soczewkującego jest relatywnie mała lub separacja kątowa między powstałymi obrazami jest mniejsza niż zdolność rozdzielcza instrumentu obserwacyjnego, zjawisko określa się mianem mikrosoczewkowania grawitacyjnego.

Interesującą klasę źródeł wykorzystywanych w badaniach nad mikrosoczewkowaniem stanowią kwazary, czyli aktywne galaktyki emitujące silne promieniowanie elektromagnetyczne. Ze względu na kosmologiczne odległości, kwazary wykazują niemal zerowy ruch własny i paralaksę, co czyni je stabilnymi, nieruchomymi punktami na sferze niebieskiej.

Czarne dziury, czyli masywne obiekty niebędące źródłem promieniowania elektromagnetycznego, mogą pełnić rolę soczewki w zjawisku mikrosoczewkowania grawitacyjnego. Rejestracja tych procesów daje możliwość wykrywania nieznanych dotąd czarnych dziur w naszej Galaktyce oraz wyznaczania ich mas.

Celem niniejszego projektu jest wyselekcjonowanie kandydatów na izolowane czarne dziury poprzez analizę anomalii kinematycznych w populacji kwazarów, u których odnotowano zmienność położeniową, sprzeczną z ich teoretyczną stacjonarnością. W projekcie wykorzystywane są dane obserwacyjne pochodzące z katalogu misji kosmicznej Gaia - Data Release 3 (DR3). W badaniach wygenerowano syntetyczne dane mające odzwierciedlać faktyczny stan populacji kwazarów, wśród których znajdują się obiekty podlegające mikrosoczewkowaniu grawitacyjnemu. Na przygotowanych danych przeprowadzono proces uczenia maszynowego wybranych modeli. Następnie wytrenowane modele zostały zastosowane do wykrycia mikrosoczewkowanych kwazarów na rzeczywistym zbiorze danych z katalogu DR3.

2 Generacja danych

Zostały wykonane symulacje soczewkowania grawitacyjnego kwazarów przez czarne dziury o różnych masach znajdujące się w naszej galaktyce. Symulacje zostały wykonane w Pythonie za pomocą oprogramowania `astromet` symulującego zjawiska soczewkowania grawitacyjnego. Pierwszy napisany przez nas skrypt generował zjawiska szeregowo, został on jednak zoptymalizowany aby zdarzenia mogły być generowane równolegle (wykorzystano bibliotekę `JOBLIB`) co pozwoliło nam na wygenerowanie większej ilości danych. Dodatkowo dodano funkcję wyboru ilości źródeł (pojedynczych symulacji) dla każdego punktu na niebie.

Wygenerowane zostały trzy zestawy danych dla różnych przedziałów mas soczewki:

- $10 - 100 M_{\odot}$
- $100 - 1000 M_{\odot}$
- $1000 - 10\,000 M_{\odot}$

Pozostałe parametry się nie zmieniają i wynoszą:

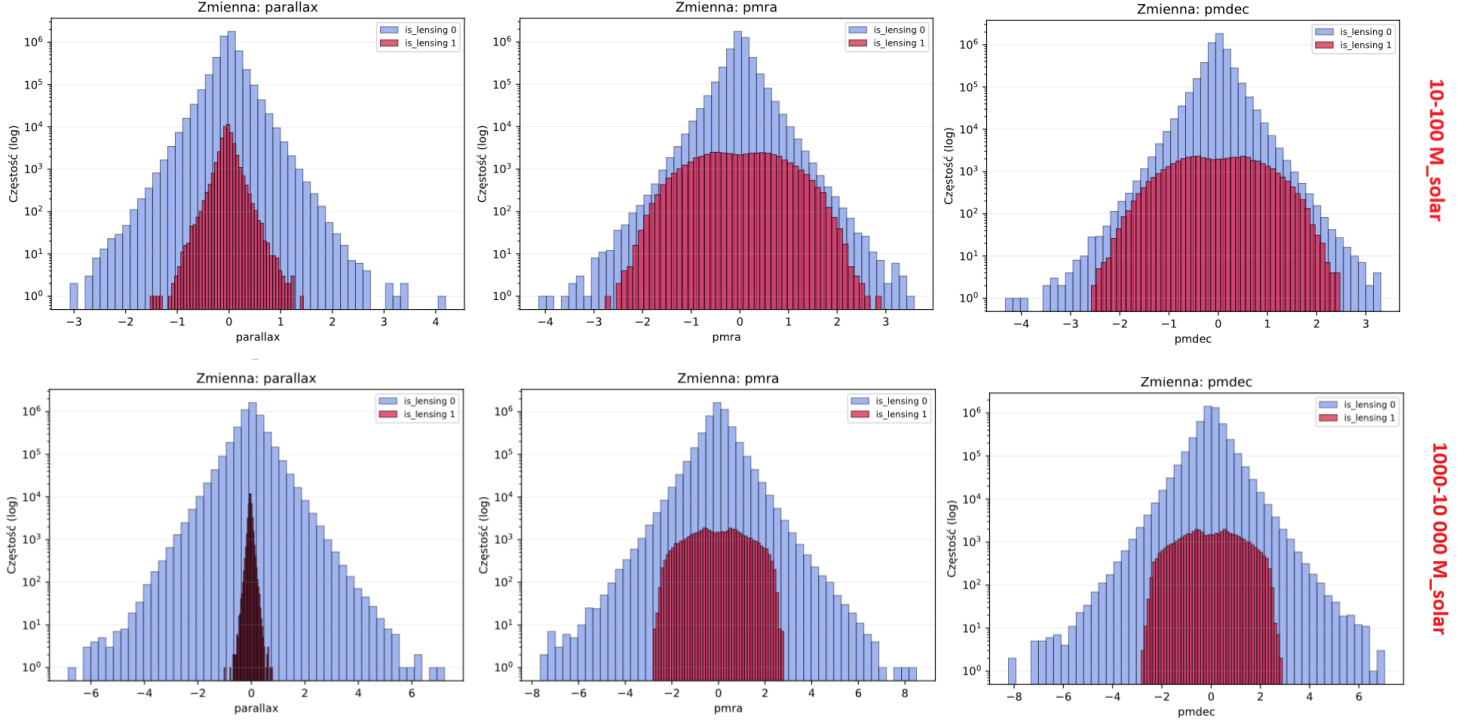
- DL (odległość soczewki): $5 - 25\text{kpc}$
- `murel` : $1 - 5$
- `beta` : $0.5 - 5$
- `BaselineG` (jasność źródła): $17 - 20\text{ mag}$

W każdym zestawie wygenerowano 10 000 pozycji na niebie, z których każda zawiera 500 źródeł. Daje to łącznie 5 000 000 obiektów na zestaw, przy czym 1% kwazarów jest soczewkowana.

Wykonanie histogramów dla każdego zestawu danych pokazało nieprawidłową zależność między masą soczewki `ML` a zachowaniem kwazarów niesoczewkowanych. Było to spowodowane sposobem w jaki `astromet` symuluje zjawisko gdy soczewkowanie nie zachodzi:

```
1
2 params_nolens = astromet.params()
3 params_nolens = astromet.define_lens(params_nolens, ra, dec,
4                                     u0=u0, t0=t0, tE=tE,
5                                     piEN=piEN, piEE=piEE,
6                                     m0=BASELINE_G, fbl=BLENDING_G
7
8                                     ,
9                                     pmrac_source=pmrac_source,
10                                    pmdec_source=pmdec_source,
11                                    d_source=d_source,
12                                    thetaE=0.001) # Bardzo mala
13
14     wartosc thetaE dla zjawiska bez soczewki
```

Pomimo zmiany parametru θ_E na bardzo małą wartość co w przybliżeniu skutkuje zjawiskiem niesoczewkowanym, pozostałe parametry funkcji nadal zależały od masy soczewki co spowodowało niezerową zależność paralaksy i ruchu własnego kwazarów niesoczewkowanych od masy soczewek przedstawioną na Rysunku 1.



Rysunek 1: Histogramy paralaksy i ruchu własnego. Nawet w przypadku gdzie soczewkowanie nie zachodzi, parametry wykazują zależność od masy soczewki co jest nieprawidłowe.

Zostało to poprawione tworząc drugi oddzielny zestaw parametrów dla 'zerowych soczewek' wykorzystywanych w przypadkach gdzie soczewkowanie nie zachodzi. Gdy zachodzi soczewkowanie astromet symuluje zjawisko z wylosowanymi parametrami z odpowiednich przedziałów, natomiast w przypadku niesoczewkowanym, wylosowane parametry są nadpisywane stałymi wartościami ustawionymi tak by soczewkowanie w efekcie nie zachodziło.

```

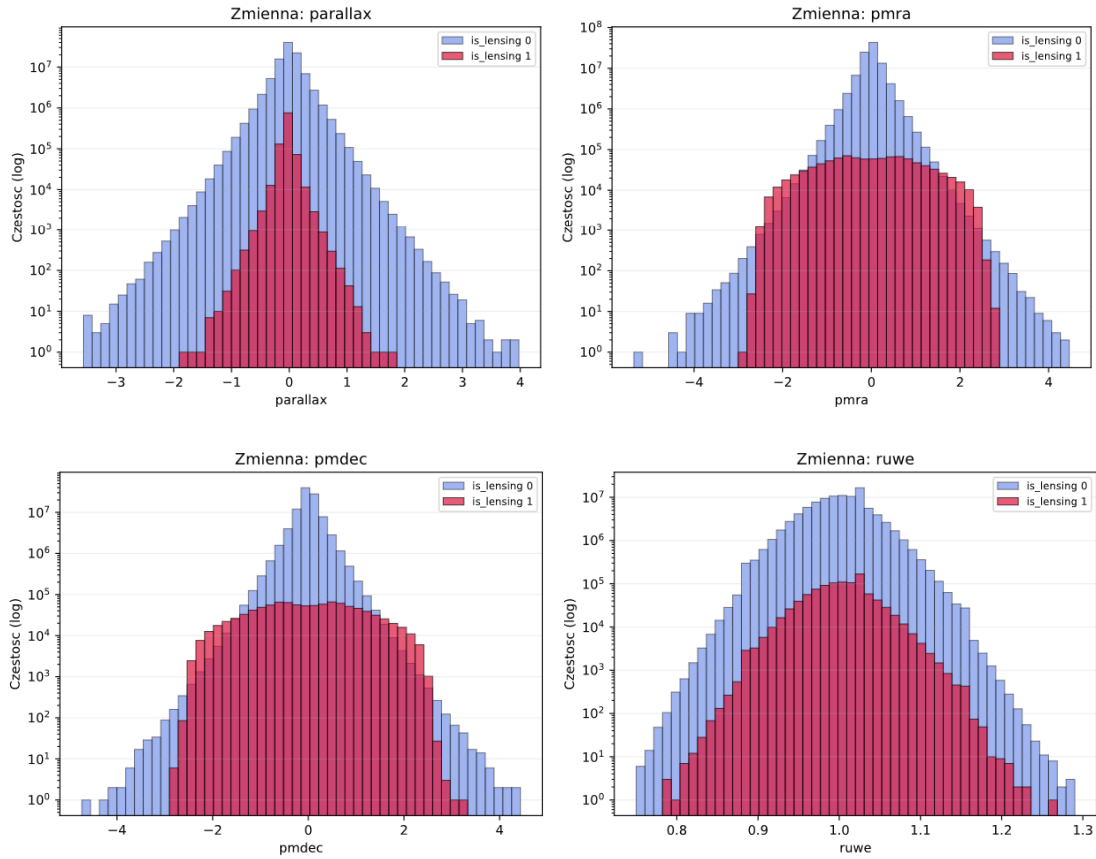
1  # Losowanie parametrów bazowych
2  ML_vals      = rng.uniform(10, 100, size=sources_per_pos)
3  DL_vals      = rng.uniform(5, 25, size=sources_per_pos)
4  murel_vals   = rng.uniform(1, 5, size=sources_per_pos)
5  beta_vals    = rng.uniform(0.5, 5, size=sources_per_pos)
6  pi_angle_vals = rng.uniform(0, 360, size=sources_per_pos)
7  BASELINE_G_arr = rng.uniform(17.0, 20.0, size=sources_per_pos)
8
9  for s in range(sources_per_pos):
10     sim_id = global_id_start + s
11     lensing_flag = is_lensing_arr[s]
12
13     if lensing_flag:
14         # Używamy wylosowanych parametrów dla soczewkowania
15         s_ML, s_DL, s_murel, s_beta, s_pi_angle = ML_vals[s],
16         DL_vals[s], murel_vals[s], beta_vals[s], pi_angle_vals[s]
17     else:
18         s_ML      = 10.0      # Stała masa referencyjna
19         s_DL      = 10.0      # Stały dystans
20         s_murel   = 1.0      # Stały ruch względny
21         s_beta    = 1000000.0 # Ogromny parametr uderzenia (
        brak soczewkowania)
        s_pi_angle = 0.0

```

Po sprawdzeniu że poprawiony kod poprawnie symuluje dane, przystąpiliśmy do generowania ostatecznych zestawów danych w ICM na klastrze superkomputera Okeanos. Wsymulowane zostało łącznie 100 milionów zjawisk dla mas soczewki $10 - 10\,000\,M_{\odot}$ i następujących parametrów:

- DL (odległość soczewki): $5 - 25\text{kpc}$
- murel : $1 - 5$
- beta : $0.5 - 5$
- BaselineG (jasność źródła): $17 - 20\text{ mag}$

Na każdej wysymulowanej pozycji na niebie znajduje się 500 źródeł przy czym dla 1% kwazarów zachodzi soczewkowanie. Dane zawierają 99 015 792 standardowych kwazarów i 992 208 zjawisk soczewkowania. Ponownie wykonano histogramy parametrów dopasowanych przez GAIA, są one pokazane poniżej.



3 Wykrywanie soczewkowanych kwazarów

Do wykrycia soczewkowanych kwazarów zaproponowano użycie uczenia maszynowego. W takim przypadku można mieć pewność, że przy dostarczeniu odpowiednio dobrych danych treningowych można uzyskać rezultat lepszy niż w przypadku algorytmów manualnych.

Zadaniem takiego algorytmu jest wykrycie jak największej ilości "naprawdę soczewkowanych" kwazarów. W idealnym przypadku bez pominięcia żadnego z nich. Terminologicznie to oznacza, że taki model musi mieć jak największą czułość (ang. - **Recall**), czyli:

$$\text{Czułość} = \frac{\text{Obiekty prawidłowo zaklasyfikowane do klasy}}{\text{Wszystkie obiekty zawarte w klasie}}.$$

Kolejną ważną metryką jest precyzja (ang. - **Precision**), czyli:

$$\text{Precyzja} = \frac{\text{Obiekty prawidłowo zaklasyfikowane do klasy}}{\text{Wszystkie obiekty zaklasyfikowane do klasy}}$$

3.1 Uczenie Maszynowe i wybór modeli

Algorytm wykrywający soczewkowane kwazary ze zbioru DR3 oparto na klasycznych algorytmach uczenia maszynowego z biblioteki `Scikit-learn`. Spośród wszystkich algorytmów dostępnych w bibliotece pierwotnie wybrano algorytmy: Gradient Boosting, Random Forest oraz Regresję Logistyczną (*Logistic Regression*). Podstawą wyboru tych algorytmów były względnie wysokie wyniki oraz względnie niewielki czas treningu w porównaniu do innych modeli np.: KNN czy SVM.

Do trenowania oraz walidacji algorytmów wykorzystano wcześniej wygenerowany zbiór 100 milionów obiektów. Zbiór ten został podzielony w proporcji 80/20: 80% stanowiły dane treningowe, a 20% dane walidacyjne i testowe. Podobnie jak w przypadku generacji danych, modele trenowano na klastrze superkomputera Okeanos. Przy trenowaniu modeli na docelowym 80-milionowym zbiorze danych zauważono, że model Random Forest przewyższał limit użytku dozwolonej pamięci RAM, przez co każda próba wytrenowania go na tym zbiorze danych kończyła się niepowodzeniem. Stwierdzono zatem, aby pozostać tylko przy modelach Regresji Logistycznej oraz Gradient Boosting, który wykorzystano w konfiguracji `HistGradientBoostingClassifier` mogącej wydajnie obsługiwać duże zbiory danych.

Wybrano 7 cech (3 natywne oraz 4 kombinacyjne). Natywne cechy to paralaksa wraz z niepewnością oraz `ruwe`. Cechy kombinacyjne powstały w wyniku kombinacji cech natywnych:

- `stn-parallax`: Stosunek wartości paralaksy do niepewności paralaksy.
- `stn-mu`: Stosunek wartości położenia do niepewności położenia.
- `murel`: Wartość ruchu własnego. Obliczona na podstawie pierwiastku sumy kwadratów `pmra` oraz `pmdec`.
- `noise-mu`: Niepewność położenia.

3.2 Ewaluacja efektywności modeli

Po wytrenowaniu modeli wykonano testy ewaluacyjne sprawdzające efektywność obu klasyfikatorów na pełnym podzbiorze testowym. W tabeli 1 zestawiono podstawowe metryki klasyfikacji. Pełny obraz dystrybucji predykcji przedstawiają macierze pomyłek zestawione w tabeli 2.

Tabela 1: Podstawowe metryki ewaluacyjne modeli dla domyślnego progu odcięcia.

Model	Klasa	Precyzja	Czułość (Recall)	F1-score
Regresja Logistyczna	0 (Tło)	1,00	0,99	1,00
	1 (Kwazary)	0,54	0,97	0,69
Gradient Boosting	0 (Tło)	1,00	1,00	1,00
	1 (Kwazary)	0,99	0,93	0,96

Tabela 2: Macierze pomyłek (Confusion Matrices) dla badanych klasyfikatorów.

Model	Stan faktyczny	Predykcja: Klasa 0	Predykcja: Klasa 1
Regresja Logistyczna	Klasa 0	19 638 059	164 884
	Klasa 1	6 586	192 071
Gradient Boosting	Klasa 0	19 801 456	1 487
	Klasa 1	13 412	185 245

Z uwagi na fakt, że klasa poszukiwana (soczewkowane kwazary - "1") jest skrajnie nieliczna w stosunku do tła, domyślny próg odcięcia nie zawsze odzwierciedla potrzeby aplikacyjne algorytmu. Przeprowadzono symulację wpływu zmiany progu prawdopodobieństwa na kompromis pomiędzy precyzją a czułością. Wyniki tej analizy przedstawia tabela 3.

Zarówno dla Regresji Logistycznej, jak i dla Gradient Boosting zaobserwowano unikalną cechę stabilności matematycznej: wymuszenie drastycznego wzrostu parametru Precision (z 60% do aż 95%) skutkuje skrajnie małym spadkiem czułości Recall (jedynie o około 0,019–0,020).

Tabela 3: Dobór optymalnego progu odcięcia (T) dla założonych poziomów precyzji.

Oczekiwane Precision	Regresja Logistyczna			Gradient Boosting		
	Próg T	Realne Prec.	Recall	Próg T	Realne Prec.	Recall
60%	0,5529	0,6000	0,9648	0,0123	0,6000	0,9672
75%	0,6832	0,7500	0,9592	0,0226	0,7500	0,9616
90%	0,8343	0,9000	0,9508	0,0702	0,9000	0,9526
95%	0,8963	0,9500	0,9452	0,1645	0,9500	0,9464

3.3 Analiza ważności cech astrometrycznych

Aby dogłębnie zbadać strukturę decyzyjną modeli, zaimplementowano analizę istotności zmiennych metodą permutacji (*Permutation Importance*) nakierowaną bezpośrednio na zmianę parametru Recall przy zróżnicowanych progach Precision. Tabela 4 prezentuje uzyskane spadki czułości (wyrażone w punktach procentowych) dla skrajnych scenariuszy precyzji (60% vs 95%).

Wyznaczone zostały również stałe współczynniki wagowe uzyskane przez Regresję Logistyczną, które zaprezentowano w tabeli 5.

Analiza wieloprogowej ważności cech ujawnia kluczową różnicę architektoniczną między modelami:

- **Regresja Logistyczna:** Przez cały zakres zmian progu odcięcia opiera się niemal wyłącznie na cesze `stn_mu` (stosunek sygnału do szumu ruchu własnego), przypisując jej dominującą wagę +10,8651. Pozostałe cechy mają charakter wyłącznie korekcyjny.
- **Gradient Boosting:** Model nieliniowy wykazuje dynamiczną ewolucję profilu istotności cech wraz ze wzrostem rygoru precyzji. Przy niskim

Tabela 4: Porównanie ważności cech (Permutation Importance) jako spadek Recall (w pkt. proc.).

Nazwa cechy	Regresja Logistyczna		Gradient Boosting	
	Prec. = 60%	Prec. = 95%	Prec. = 60%	Prec. = 95%
stn_mu	96,48	94,52	92,83	94,64
murel	0,46	1,08	96,72	94,64
noise_mu	1,94	2,45	12,92	80,14
err_parallax	0,46	0,47	2,94	14,55
parallax	0,00	0,01	2,85	16,42
stn_parallax	0,02	0,02	1,96	4,63
ruwe	0,00	0,00	0,00	0,00

Tabela 5: Współczynniki strukturalne i kierunek wpływu wag w modelu Regresji Logistycznej.

Nazwa cechy	Współczynnik	Interpretacja wpływu na klasyfikację
stn_mu	+10,8651	Silnie stymuluje klasyfikację jako anomalia (klasa 1).
noise_mu	+1,4549	Stymuluje klasyfikację jako anomalia (klasa 1).
parallax	+0,0364	Słabo stymuluje klasyfikację jako anomalia (klasa 1).
ruwe	+0,0220	Słabo stymuluje klasyfikację jako anomalia (klasa 1).
stn_parallax	-0,1199	Destymuluje (sprzyja klasyfikacji jako tło - klasa 0).
err_parallax	-0,4909	Destymuluje (sprzyja klasyfikacji jako tło - klasa 0).
murel	-0,5306	Destymuluje (sprzyja klasyfikacji jako tło - klasa 0).

progu decyzyjnym (60%) model opiera się głównie na relacji cech `murel` oraz `stn_mu`. Jednak w celu osiągnięcia skrajnie wysokiej czystości podzbioru (Precision 95%), algorytm drastycznie zwiększa zależność od szumu ruchu własnego `noise_mu` (skok istotności z 12,92% do aż 80,14%) oraz cech paralaksy (`parallax` i `err_parallax`). Pozwala to modelowi na bezbłędne odrzucenie trudnych obiektów tła astrometrycznego.

3.4 Końcowa wersja algorytmu treningowego

W końcowej wersji skryptu treningowego odrzucono cechę `ruwe` ze względu na jej znikomą wagę (tabela 4). Wzbogacono natomiast przestrzeń wejściową o nowe cechy kombinacyjne o charakterze nieliniowym, których celem było ułatwienie geometrii podziału przestrzeni decyzyjnej przez algorytmy drzewiaste. Do zestawu zmiennych dodano:

- **abs_parallax**: Wartość bezwzględna paralaksy, odzwierciedlająca czysty moduł przesunięcia paralaktycznego bez względu na kierunek fizyczny lub znak błędu pomiarowego.
- **abs_stn_parallax**: Wartość bezwzględna stosunku paralaksy do jej niepewności (moduł parametru *signal-to-noise* dla paralaksy).
- **log_err_parallax**: Logarytm naturalny z niepewności pomiarowej paralaksy przekształcony funkcją $\log(1 + x)$ w celu stabilizacji skośnych rozkładów błędów astrometrycznych.
- **log_noise_mu**: Logarytm naturalny ze skumulowanej niepewności ruchu własnego, zmniejszający wpływ ekstremalnych wartości szumu astrometrycznego na proces podziału drzewa.

Ostateczna, zoptymalizowana macierz cech wejściowych objęła 9 zmiennych: `stn_mu`, `murel`, `abs_parallax`, `abs_stn_parallax`, `log_err_parallax`, `log_noise_mu`, `parallax`, `err_parallax` oraz `noise_mu`.

W celu znaczącego skrócenia czasu obliczeń oraz poprawy stabilności matematycznej algorytmu `HistGradientBoostingClassifier`, całkowicie przeformułowano architekturę procesu uczenia maszynowego poprzez wprowadzenie dwóch nowych komponentów:

1. **RandomUnderSampler**: Z uwagi na to, że poszukiwane kwazary stanowią skrajnie nieliczny ułamek w stosunku do gigantycznego tła astrometrycznego, funkcja kosztu modelu mogłaby zostać zdominowana przez klasę większościową. Zastosowano kontrolowane podpróbowanie tła, ustalając docelowy udział klasy rzadkiej (*sampling strategy*) na poziomie 5% zredukowanego zbioru treningowego. Krok ten pozwolił na efektywne uczenie nieliniowych zależności bez utraty cennych informacji o kwazarach, drastycznie redukując przy tym narzut pamięciowy i czasowy na klastrze Okeanos.

2. **RandomizedSearchCV z dedykowaną funkcją oceny:** Porzucono klasyczne kryteria optymalizacji, takie jak *Accuracy* czy pole pod krzywą ROC (*ROC AUC*), które bywają mylące przy skrajnym niezbalansowaniu klas. Zamiast tego zaimplementowano autorską funkcję oceny (*custom scorer*). Funkcja ta dla każdej testowanej kombinacji hiperparametrów dynamicznie wyznacza pełną krzywą precyzji i czułości (*Precision-Recall curve*), po czym identyfikuje punkt najbliższy założonemu rygowi $\text{Precision} = 0,60$ i zwraca wartość czułości (*Recall*) w tym konkretnym miejscu. Dzięki losowemu przeszukiwaniu przestrzeni parametrów (*RandomizedSearchCV*), algorytm odnalazł strukturę wag drzew, która maksymalizuje odsetek wykrytych obiektów przy jednoczesnym ścisłym zachowaniu czystości generowanej próby.

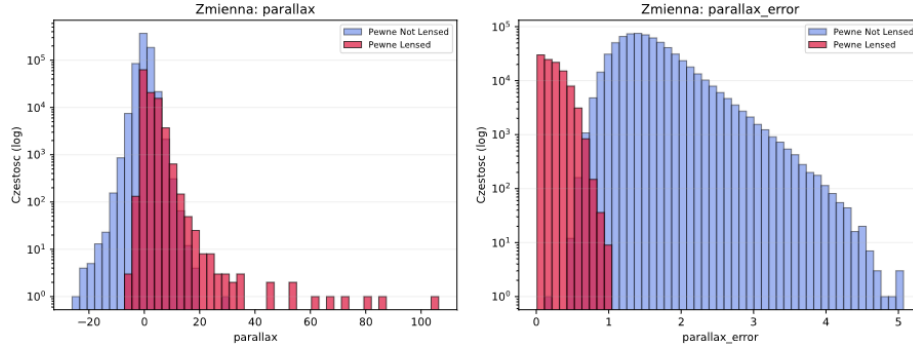
Wdrożenie nowej strategii treningu oraz zmodyfikowanego zestawu cech wpłynęło bezpośrednio na końcowe zachowanie klasyfikatorów. W porównaniu do podejścia pierwotnego, dla precyzji wynoszącej 60% czułość Regresji Logistycznej zmieniła się z poziomu 0,9648 na 0,9674 oraz czułość Gradient Boosting zmieniła się z poziomu 0.9672 na 0.9676.

3.5 Zastosowanie wytrenowanych modeli do wykrycia kwazarów na zbiorze rzeczywistym

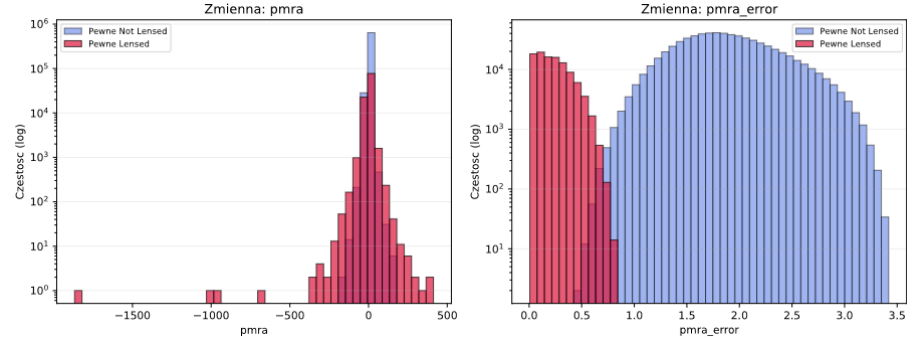
Wytrenowane modele zaaplikowano do analizy rzeczywistego katalogu astrometrycznego zawierającego 6649162 kwazarów. W architekturze skryptu aplikacyjnego wykorzystano metodę `predict_proba` zamiast klasyfikacji binarnej, co pozwoliło uzyskać ciągłe i surowe wartości prawdopodobieństwa przynależności obiektów do danej klasy. Ewentualne wartości NaN pojawiające się w zbiorze zostały zastąpione przez 0. W efekcie działania skryptu do oryginalnego pliku wejściowego dopisano po dwie kolumny dla każdego modelu, zachowując nienaruszoną strukturę wierszy:

- `{prefix}_probability_lensed` – wyliczone prawdopodobieństwo bycia kwazarem soczewkowanym,
- `{prefix}_probability_not_lensed` – komplementarne prawdopodobieństwo przypisania obiektu do tła.

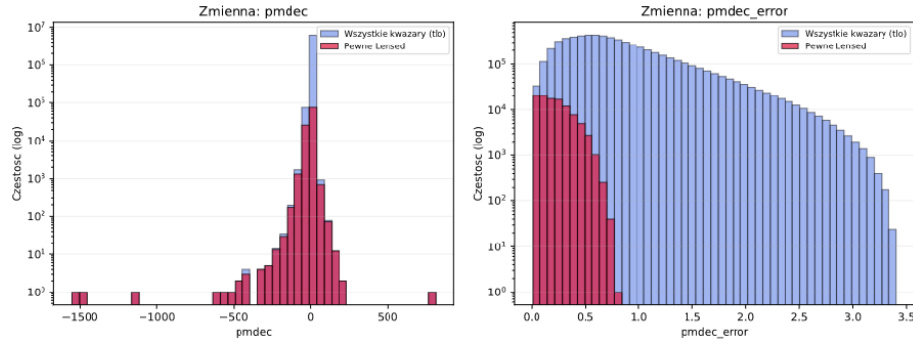
Znacznik `{prefix}` definiuje model decyzyjny (`lr` dla Regresji Logistycznej, `gb_rand` dla zoptymalizowanego `HistGradientBoostingClassifier`). Filtracja rozkładu wyników pozwoliła wyodrębnić obiekty sklasyfikowane ze skrajną pewnością. Dla klasy kwazarów soczewkowanych algorytmy zidentyfikowały 103501 obiektów z prawdopodobieństwem wynoszącym dokładnie 100%. Z kolei dla klasy obiektów niesoczewkowanych odnaleziono 669865 obiektów z prawdopodobieństwem 100%. Na podstawie powyższych danych stworzono histogramy (rys. 2, 3 oraz 4) oraz mapę nieba rys. 5. Wszystkie trzy histogramy wskazują na to, że w przypadku kwazarów soczewkowanych dominują niskie wartości błędów. Oznacza to, że wartości cech `stn_mu` oraz `abs_stn_parallax` są wyższe niż w przypadku kwazarów niesoczewkowanych.



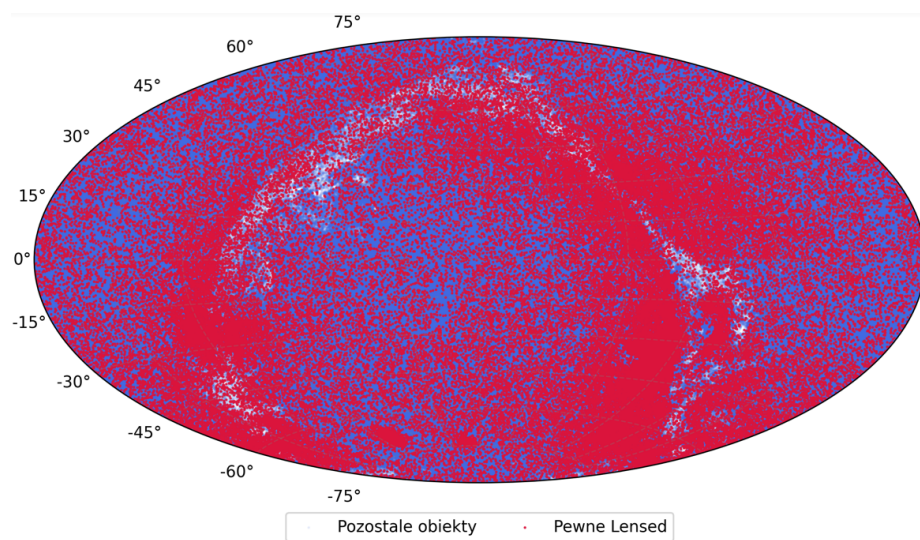
Rysunek 2: Histogramy porównawcze paralaksy wraz z błędem dla kwazarów soczewkowanych i niesoczewkowanych.



Rysunek 3: Histogramy porównawcze ruchu własnego rektascensji wraz z błędem dla kwazarów soczewkowanych i niesoczewkowanych.



Rysunek 4: Histogramy porównawcze ruchu własnego deklinacji wraz z błędem dla kwazarów soczewkowanych i niesoczewkowanych.



Rysunek 5: Mapa nieba kwazarów soczewkowanych na tle wszystkich kwazarów w danych Gaia DR3