Raport, część III

W ramach danej części projektu kontynuowałam analizę datasetu dotyczącego kosztów edukacji międzynarodowej. Ten dataset został stworzony w celu analizy i porównania kosztów studiowania za granicą. Obejmuje ponad 900 programów z uniwersytetów na całym świecie, w tym dane dotyczące utrzymania w kraju, zakwaterowania, opłat wizowych i ubezpieczenia. Dataset jest złożony z 907 wierszy i 12 column oraz zawiera 5 cech kategorialnych (kraj, miasto, uniwersytet, program i poziom edukacji) i 7 cech numerycznych (czas trwania, koszty utrzymania, czynsz, opłata wizowa, ubezpieczenie, kurs wymiany i całkowitą kwotę)

```
rows, columns: (907, 12)
```

Zmienna docelowa nadal Tuition_USD.

1. Cross-validation and model evaluation
   W ramach tego zadania postanowiłam zintegrować metodę walidacji krzyżowej z moją implementacją regresji liniowej, wykorzystującą spadek gradientu.

Numer podzbioru	MAE	RMSE
1	16221.69	68062.91
2	11101.16	29620.62
3	66028.62	144289.98

   Wynik:

```
For  1  set:
Mean average error:  16221.69
Rout mean squared error:  68062.91

For  2  set:
Mean average error:  11101.16
Rout mean squared error:  29620.62

For  3  set:
Mean average error:  66028.62
Rout mean squared error:  144289.98
```

   Wnioski:
   Wyniki kroswalidacji są różne dla każdych zbiorów. Takie różne wyniki mogą wskazywać na niestabilność modelu, z tego można wywnioskować że model pracuje lepiej na jednym określonym zbiorze testowym (pierwszym). Wyniki mogą różnić się jeszcze dlatego że w każdym ze zbiorów mogą się okazać różne dane testowe: zupełnie inne uniwersytety i stopni studiów.

2. Wykresy zbieżności i analiza błędów

a. Wyniki regresji liniowej

Model	Zbiór	Plik .csv	MAE	RMSE
Regresja liniowa	train	ln_train_predictions.csv	1519.98	2535.63
Regresja liniowa	validate	ln_validate_predictions.csv	346934.95	850400.51
Regresja liniowa	test	ln_test_predictions.csv	1119841.62	3588598.48

```
Model:  ln , set:  train
Mean average error:  1519.98
Rout mean squared error:  2535.63

Model:  ln , set:  validate
Mean average error:  346934.95
Rout mean squared error:  850400.51

Model:  ln , set:  test
Mean average error:  1119841.62
Rout mean squared error:  3588598.48
```

Wnioski:
Z tabeli widać, że na zbiorze testowym błędy są znacznie większe niż na zbiorze treningowym. To świadczy o tym, że model został mocno przekwalifikowany, czyli istnieje problem nadmiernego dopasowania (overfitting).

b. Po dodawaniu dodatkowych cech:

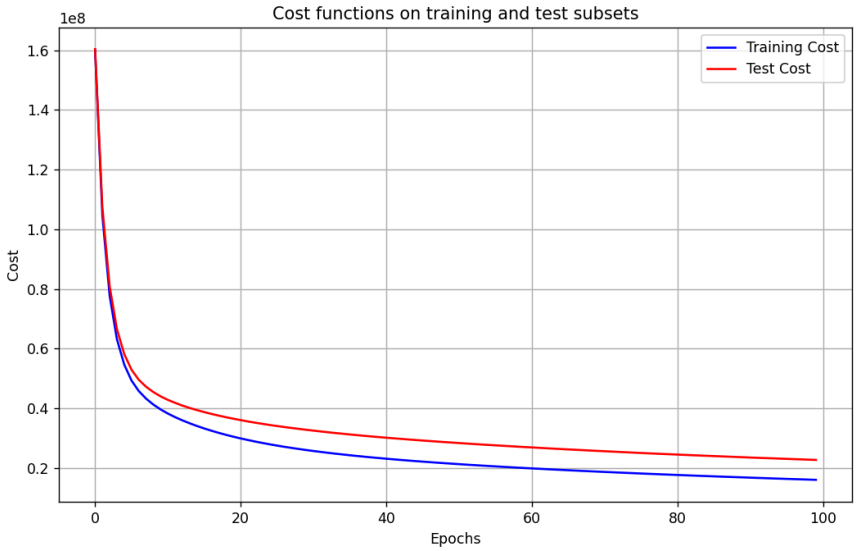
```
Model:  ln , set:  train
Mean average error:  817.98
Rout mean squared error:  1530.02

Model:  ln , set:  validate
Mean average error:  871732.83
Rout mean squared error:  4092807.31

Model:  ln , set:  test
Mean average error:  2118282.44
Rout mean squared error:  8292981.48
```

Model	Zbiór	MAE	RMSE
Regresja liniowa	train	817.98	1530.02

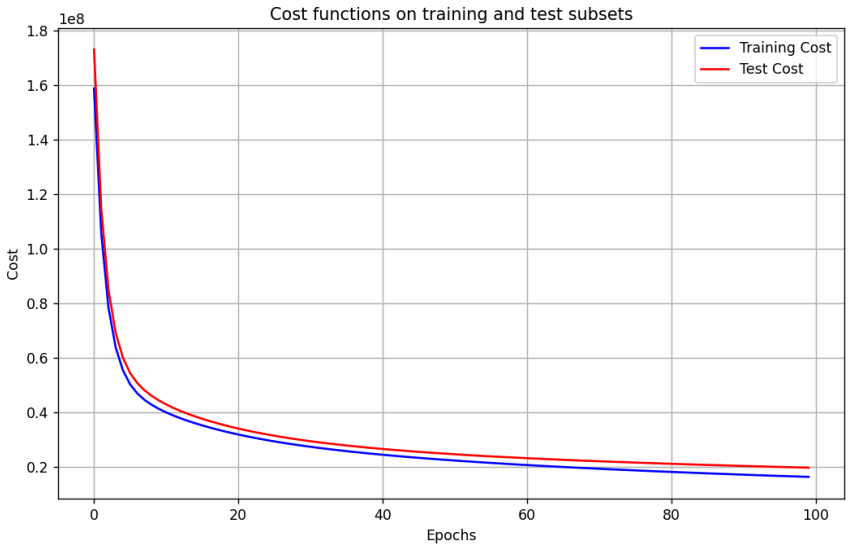
Regresja liniowa	validate	871732.83	4092807.31
Regresja liniowa	test	2118282.48	8292981.48

Wnioski:
Po dodaniu dodatkowych cech na zbiorze treningowym mae i rmse zmalały prawie w 2 razy ale na pozostałych zbiorach rmse i mae wzrosły kilkakrotnie. Takie wyniki pokazują, że model świetnie pracuje na danych treningowych, a dalej nie widzi wzorców i nadal istnieje problem nadmiernego dopasowania. To znaczy, że dodatkowe cechy skomplikowały model regresji liniowej.

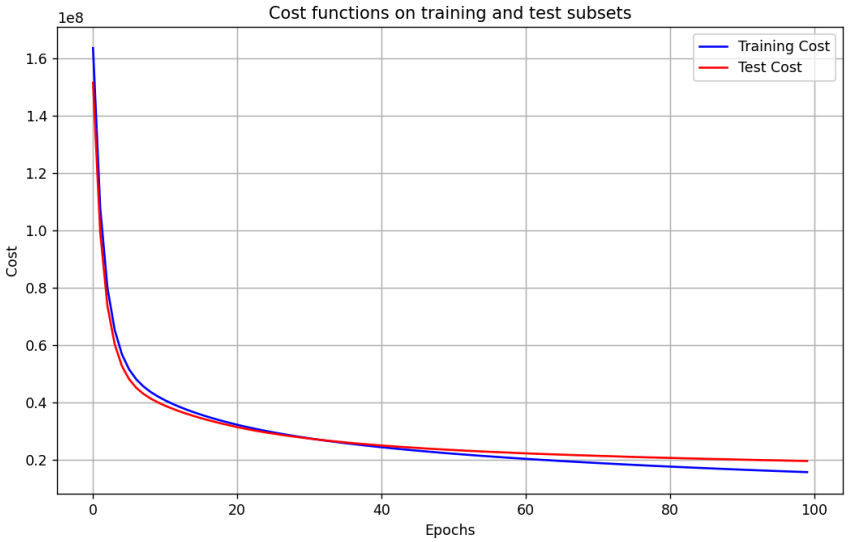
c. Wykresy funkcji kosztu na podzbiorze treningowym oraz testowym dla własnej implementacji regresji liniowej ze spadkiem gradientu
Dla pierwszego podzbioru:



Dla drugiego podzbioru:

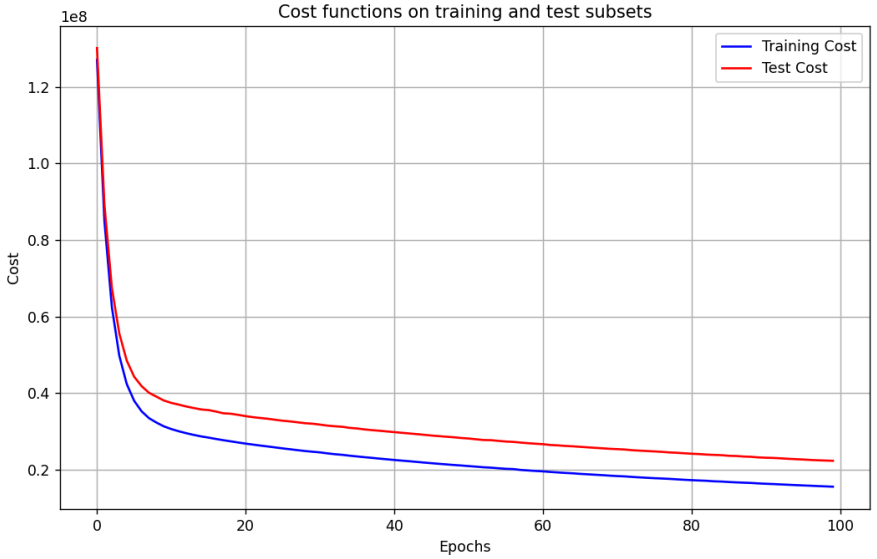


Dla trzeciego podzbioru:

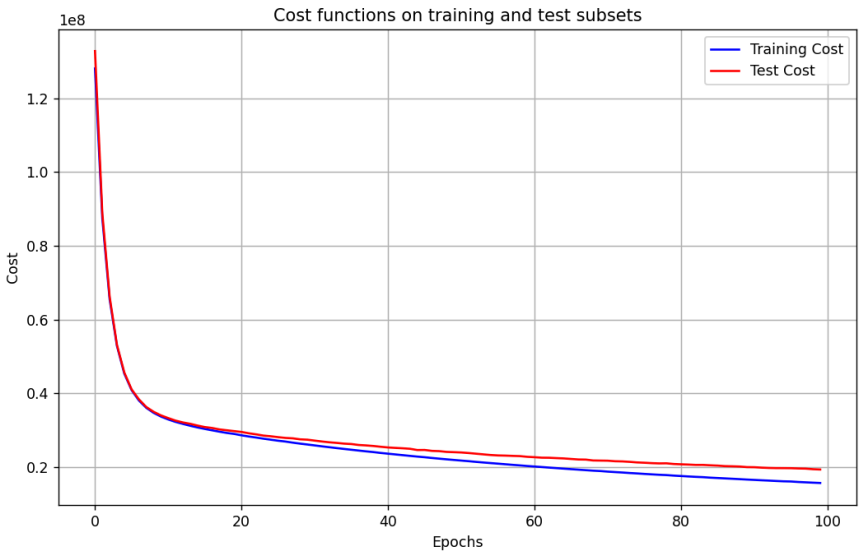


Wnioski:
Z każdą epoką na zbiorach treningowych MAE maleje, to oznacza, że model uczy się lepiej i następuje zbieżność (spadek błędu) w czasie uczenia modelu. Z wykresów widać, że model idealnie powtarza prawdziwe dane. Jeśli zwiększyć stopień wielomianu, model zacznie zbyt dokładnie zapamiętywać dane treningowe, w tym błędy i hałas. Z tego powodu będzie działał gorzej na nowych danych i może pojawić się przekwalifikowanie.

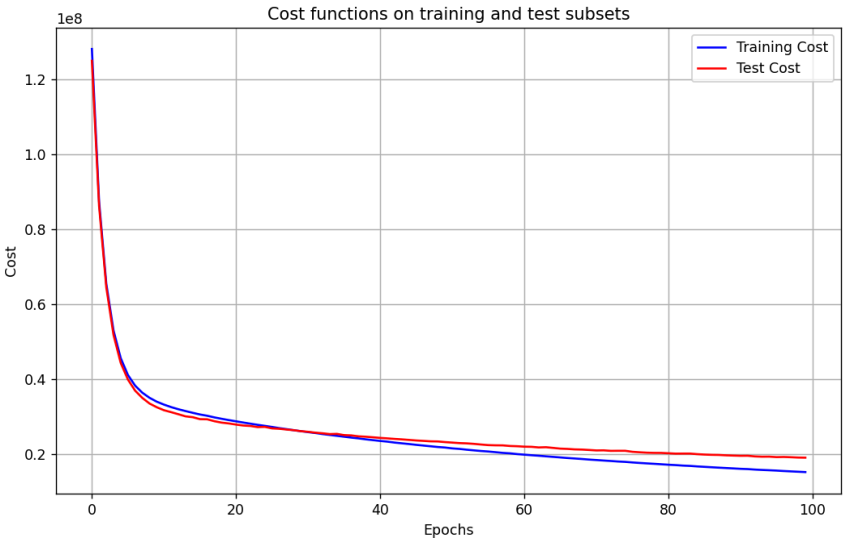
d. Wykresy po dodaniu dodatkowych cech:
Dla pierwszego podzbioru:



Dla drugiego podzbioru:



Dla trzeciego podzbioru:



Wnioski:

Wykresy są podobne do wykresów, otrzymanych bez użycia dodatkowych cech. Ale różnią się otrzymane wartości:

```
For  1  set:
Mean average error:  7343997751.68
Rout mean squared error:  45123957534.56

For  2  set:
Mean average error:  337400901.67
Rout mean squared error:  1789332494.35

For  3  set:
Mean average error:  1458693201.38
Rout mean squared error:  2482408674.8
```

Co jawnie wskazuje na skomplikowanie modelu i obecność nadmiernego dopasowania.

3. Regularyzacja L1 i L2
    a. Porównanie regresji liniowej ze spadkiem gradientu bez i z regulacją

Model	Zbiór	MAE	RMSE
Regresja liniowa (grad)	1	16221.69	68062.91
Regresja liniowa (grad)	2	11101.16	29620.62
Regresja liniowa (grad)	3	66028.62	144289.98
Regresja liniowa (grad + l1)	1	16204.27	22794.06
Regresja liniowa (grad + l1)	2	17156.77	24082.08
Regresja liniowa (grad + l1)	3	16216.96	22765.42
Regresja liniowa (grad +l2)	1	16322.45	23143.25
Regresja liniowa (grad + l2)	2	17085.73	23906.46
Regresja liniowa (grad + l2)	3	16149.31	22527.95

Wnioski:
Regulacja daje bardziej stabilne wyniki i l2 zmniejsza przekwalifikowanie modelu.

```
l1

For  1  set:
Mean average error:  16204.27
Rout mean squared error:  22794.06
l1

For  2  set:
Mean average error:  17156.77
Rout mean squared error:  24082.08
l1

For  3  set:
Mean average error:  16216.96
Rout mean squared error:  22765.42
```

```
l2

For  1  set:
Mean average error:  16322.45
Rout mean squared error:  23143.25
l2

For  2  set:
Mean average error:  17085.73
Rout mean squared error:  23906.46
l2

For  3  set:
Mean average error:  16149.31
Rout mean squared error:  22527.95
```

b. Porównanie zwykłej regresji liniowej (bez regulacji) z Ridge i Lasso:

Model	Zbiór	MAE	RMSE
Regresja liniowa	train	1519.98	2535.63
Regresja liniowa	validate	346934.95	850400.51
Regresja liniowa	test	1119841.62	3588598.48
Ridge	train	2352.96	3065.73
Ridge	validate	4544.45	5868.02
Ridge	test	3461.39	4586.32
Lasso	train	1559.0	2547.37

Lasso	validate	5992.62	8191.49
Lasso	test	10101.97	11998.28

```
Model:  ln , set:  train
Mean average error:  1519.98
Rout mean squared error:  2535.63
First 16 weights :
[ 4.36024505e+03 -2.27763662e+03  2.25061243e+03  4.11255506e+02
 -8.49749804e+02 -6.00501616e+05  1.70668983e+04 -2.64714319e+05
 -7.72637366e+04 -1.04866699e+05 -1.08754805e+05 -2.78246212e+05
 -2.67504124e+05 -7.29737713e+04 -2.43866921e+05 -9.94905730e+04]

Model:  ln , set:  validate
Mean average error:  346934.95
Rout mean squared error:  850400.51
First 16 weights :
[ 4.36024505e+03 -2.27763662e+03  2.25061243e+03  4.11255506e+02
 -8.49749804e+02 -6.00501616e+05  1.70668983e+04 -2.64714319e+05
 -7.72637366e+04 -1.04866699e+05 -1.08754805e+05 -2.78246212e+05
 -2.67504124e+05 -7.29737713e+04 -2.43866921e+05 -9.94905730e+04]

Model:  ln , set:  test
Mean average error:  1119841.62
Rout mean squared error:  3588598.48
First 16 weights :
[ 4.36024505e+03 -2.27763662e+03  2.25061243e+03  4.11255506e+02
 -8.49749804e+02 -6.00501616e+05  1.70668983e+04 -2.64714319e+05
 -7.72637366e+04 -1.04866699e+05 -1.08754805e+05 -2.78246212e+05
 -2.67504124e+05 -7.29737713e+04 -2.43866921e+05 -9.94905730e+04]
```

```
Model:  r , set:  train
Mean average error:  2352.96
Rout mean squared error:  3065.73
First 16 weights :
[ 1805.10309029 -2083.01883809  5761.02028829  2128.68027264
   929.25730329  -375.20416789 -2698.48600419 13907.70388864
 -1641.00440323  -992.73236926 -2159.49509021 12046.35499909
 -3156.6442617    537.98777369 -9016.25026119  2388.18863183]

Model:  r , set:  validate
Mean average error:  4544.45
Rout mean squared error:  5868.02
First 16 weights :
[ 1805.10309029 -2083.01883809  5761.02028829  2128.68027264
   929.25730329  -375.20416789 -2698.48600419 13907.70388864
 -1641.00440323  -992.73236926 -2159.49509021 12046.35499909
 -3156.6442617    537.98777369 -9016.25026119  2388.18863183]

Model:  r , set:  test
Mean average error:  3461.39
Rout mean squared error:  4586.32
First 16 weights :
[ 1805.10309029 -2083.01883809  5761.02028829  2128.68027264
   929.25730329  -375.20416789 -2698.48600419 13907.70388864
 -1641.00440323  -992.73236926 -2159.49509021 12046.35499909
 -3156.6442617    537.98777369 -9016.25026119  2388.18863183]
```

```
Model:  l , set:  train
Mean average error:  1559.0
Rout mean squared error:  2547.37
First 16 weights :
[  4137.65307429  -1910.17785592   2224.11819479   -914.93824071
   -420.88700119  -1264.85075108 -25300.56813596  17571.6593032
 -13627.948552    -4552.21493404 -17440.12451237   7633.29121235
 -15133.50216115  -6611.88262508 -12664.36256123  -1586.11234825]

Model:  l , set:  validate
Mean average error:  5992.62
Rout mean squared error:  8191.49
First 16 weights :
[  4137.65307429  -1910.17785592   2224.11819479   -914.93824071
   -420.88700119  -1264.85075108 -25300.56813596  17571.6593032
 -13627.948552    -4552.21493404 -17440.12451237   7633.29121235
 -15133.50216115  -6611.88262508 -12664.36256123  -1586.11234825]

Model:  l , set:  test
Mean average error:  10101.97
Rout mean squared error:  11998.28
First 16 weights :
[  4137.65307429  -1910.17785592   2224.11819479   -914.93824071
   -420.88700119  -1264.85075108 -25300.56813596  17571.6593032
 -13627.948552    -4552.21493404 -17440.12451237   7633.29121235
 -15133.50216115  -6611.88262508 -12664.36256123  -1586.11234825]
```

Wnioski:

Zwykła regresja liniowa pokazuje najmniejsze błędy na podzbiorze treningowym, ale na podzbiorze walidacyjnym i testowym najgorsze ze wszystkich 3 metod. Taka różnica pomiędzy wynikami mówi o tym że ten model został przekwalifikowany.

Model Ridge pokazuje najlepsze wyniki ze wszystkich 3 metod ponieważ na każdym z podzbiorów testowych wartości MAE są małe, odnośnie pozostałych modeli i są mniej więcej takie same, co mówi o stabilności tego modelu.

Model Lasso pokazuje mniejsze błędy na podzbiorach testowym i walidacyjnym niż regresja liniowa, ale nadal znacznie większe niż Ridge.

Regresja liniowa ma największe wagi ze wszystkich metod bo nie ogranicza je.

Ridge ma mniejsze wagi niż regresja liniowa bo specjalnie zmniejsza wagi, co powstrzymuje przekwalifikowanie.

Lasso też ma mniejsze wagi niż regresja liniowa i dodatkowo zeruje niektóre wagi (kasuje mniej ważne cechy).

```
[  4137.65307429  -1910.17785592   2224.11819479   -914.93824071
   -420.88700119  -1264.85075108 -25300.56813596  17571.6593032
 -13627.948552    -4552.21493404 -17440.12451237   7633.29121235
 -15133.50216115  -6611.88262508 -12664.36256123  -1586.11234825
 -13007.17308749 -11566.86334945 -14973.66605044 -13733.33622025
      0.          -14778.27432611  -2023.93283119      0.
  -2585.41213606 -12388.76552654  -9166.50909636 -16669.95907459
 -19467.32775835  -1693.57037051  15204.03810672 -17239.59945734
 -13421.85426456 -13952.04150554  -9387.94869736 -13627.80521813
  -9130.27977556   7367.84701984 -11160.27307885  -5500.80861893
 -12288.11697794 -14670.57027636 -11592.49572228 -14047.59413937
 -17512.69413584 -19027.53992117   4147.53531071  12230.72101036
  27862.03728832   4389.99400636]
```

4. Usprawnienie danych – balansowanie zbiorów

Na początku musiałam zmienić swoją zmienno docelową bo była cechą numeryczną. Zatem za pomocą kross walidatora przeprowadziłam analizę zbalansowanych i niezbalansowanych podzbiorów.

Podzbiór	MAE	RMSE
1(niezb)	0,25	0,13
2(niezb)	0,28	0,14
3(niezb)	0,28	0,14
1(zb)	0,27	0,14
2(zb)	0,27	0,14
3(zb)	0,27	0,14

```
Cross-Validation Results

1 set

Name: Original data
MAE: 0.25
RMSE: 0.13

2 set

Name: Original data
MAE: 0.28
RMSE: 0.14

3 set

Name: Original data
MAE: 0.28
RMSE: 0.14

Standard Evaluation with Balancing Methods

Name: Original data
MAE: 0.27
RMSE: 0.14

Name: SMOTE
MAE: 0.27
RMSE: 0.14

Name: Undersampling
MAE: 0.27
RMSE: 0.14
```

Wnioski:

Niskie wartości MAE i RMSE wynikają z faktu, że model przewiduje tylko trzy klasy kosztów, a maksymalny możliwy błąd to zaledwie 2 jednostki. Zastosowanie metod balansowania danych (SMOTE i undersampling) nie poprawiło jakości, prawdopodobnie z powodu wystarczającego zbalansowania danych wyjściowych. Stabilne błędy we wszystkich podzbiorach walidacji krzyżowej potwierdzają niezawodność modelu.

5. Optymalizacja hyper parametrów

Modele dla których została przeprowadzona optymalizacja za pomocą GridSearchCV: decision tree regression, random forest regression.

Parametry dla modelu decision tree regression:

```
'model__max_depth': [3, 5, 7, 10],
'model__min_samples_split': [2, 5, 10],
'model__min_samples_leaf': [1, 2, 4]
```

Parametry dla modelu random forest regression:

```
'model__n_estimators': [50, 100],
'model__max_depth': [5, 10, 15],
'model__max_features': ['sqrt', 'log2', 0.5]
```

Przed optymalizacją

Model	Podzbiór	MAE	RMSE
Decision tree regression	train	0.0	0.0
Decision tree regression	validate	3046.7	4435.76
Decision tree regression	test	2679.12	3954.45
Random forest regression	train	901.51	1523.31
Random forest regression	validate	4192.31	5828.93
Random forest regression	test	2671.43	3446.38

```
DecisionTree
Before optimization:
train: MAE=0.0, RMSE=0.0
validate: MAE=3046.7, RMSE=4435.76
test: MAE=2679.12, RMSE=3954.45
```

```
RandomForest
Before optimization:
train: MAE=901.51, RMSE=1523.31
validate: MAE=4192.31, RMSE=5828.93
test: MAE=2671.43, RMSE=3446.38
```

Po optymalizacji

Model	Podzbiór	MAE	RMSE
Decision tree regression	train	2870.15	4173.4
Decision tree regression	validate	3229.08	4036.13
Decision tree regression	test	3002.8	3713.06
Random forest regression	train	995.46	1492.94
Random forest regression	validate	3514.63	4775.82
Random forest regression	test	2795.32	3735.34

Decision tree regression:

```
After optimization:
train: MAE=2870.15, RMSE=4173.4
validate: MAE=3229.08, RMSE=4036.13
test: MAE=3002.8, RMSE=3713.06
```

Random forests regression:

```
After optimization:
train: MAE=995.46, RMSE=1492.94
validate: MAE=3514.63, RMSE=4775.82
test: MAE=2795.32, RMSE=3735.34
```

Dlaczego przeszukiwanie parametrów jest ogólnie trudnym problemem:
Ten problem jest skomplikowany ponieważ model może zawierać dużo parametrów, każdy z których może przyjmować dużo wartości i trudno określić przy których wartościach wynik będzie najlepszy.

Najlepsze parametry i ich wpływ na wynik końcowy:
Decision tree regression:

```
Best parameters: {'model__max_depth': 7, 'model__min_samples_leaf': 2, 'model__min_samples_split': 2}
```

Random forests regression:

```
Best parameters: {'model__max_depth': 15, 'model__max_features': 0.5, 'model__n_estimators': 50}
```

Wnioski:
Przed optymalizacją decision tree pokazywał zerowe wartości mae i rmse co mówi o tym, że na zbiorze testowym model był przekwalifikowany, po optymalizacji model pozbawił się tego błędu. Po optymalizacji mae i rmse w random forests regression zmniejszyły się, co mówi o tym że optymalizacja polepszyła te wyniki.

Wynik ogólny:

1. Wyniki z uwzględnieniem wszystkich metod optymalizacji

Metoda	Podzbiór	MAE	RMSE
Regresja liniowa	1	16221.69	68062.91
Regresja liniowa	2	11101.16	29620.62
Regresja liniowa	3	66028.62	144289.98
Regresja liniowa z dodatkowymi cechami	1	817.98	1530.02
Regresja liniowa z dodatkowymi cechami	2	871732.83	4092807.31
Regresja liniowa z dodatkowymi cechami	3	2118282.48	8292981.48
Regresja liniowa z regularyzacją l1	1	16204.27	22794.06
Regresja liniowa z regularyzacją l1	2	17156.77	24082.08
Regresja liniowa z	3	16216.96	22765.42

regularyzacją l1			
Regresja liniowa z regularyzacją l2	1	16322.45	23143.25
Regresja liniowa z regularyzacją l2	2	17085.73	23906.46
Regresja liniowa z regularyzacją l2	3	16149.31	22527.95
Regresja liniowa zbalansowana	1	0,27	0,14
Regresja liniowa zbalansowana	2	0,27	0,14
Regresja liniowa zbalansowana	3	0,27	0,14

Metody bez własnej implementacji:

Random forests regression, decision tree

Podzbiór	MAE	RMSE
1(dt)	2870.15	4173.4
2(dt)	3229.08	4036.13
3(dt)	3002.8	3713.06
1(rfr)	995.46	1492.94
2(rfr)	3514.63	4775.82
3(rfr)	2795.32	3735.34

Z własnej implementacji modeli regresja z regularyzacją L2 (Ridge) daje najstabilniejsze i najmniejsze błędy wśród wszystkich modeli regresyjnych — to czyni ją najlepszym wyborem dla tych danych.

Model zbalansowany działa dobrze, ale ze względu na bardzo niskie wartości błędów i dyskretne klasy celu, lepiej sprawdzi się w problemach klasyfikacyjnych niż w przewidywaniu wartości liczbowych.

Ale ogólnie lepsze wyniki daje random regression tree po optymalizacji.

2. W danym punkcie skupiłam się na opisie modelu, którego własna implementacja podaje lepsze wyniki.

   Model Ridge uzyskał najlepsze wyniki spośród wszystkich trzech metod. Na każdym z podzbiorów testowych wartości MAE były niskie i bardzo zbliżone do siebie, co świadczy o stabilności i dobrej generalizacji modelu. W porównaniu do zwykłej regresji liniowej, Ridge osiąga mniejsze błędy i lepiej radzi sobie z nadmiernym dopasowaniem, dzięki zastosowaniu regularyzacji L2. Regularyzacja ta celowo ogranicza wielkość wag (współczynników), co pozwala uniknąć sytuacji, w której model zbyt mocno dopasowuje się do danych treningowych, kosztem danych testowych. Dzięki temu model Ridge jest bardziej odporny na szum w danych i potrafi lepiej uogólniać wzorce. Wagi w tym modelu są zauważalnie mniejsze niż w klasycznej regresji, co jest zgodne z jego założeniem – preferować prostsze modele o mniejszych współczynnikach. Podsumowując, Ridge łączy dobrą dokładność, stabilność oraz odporność na nadmierne dopasowanie, co czyni go najlepszym wyborem spośród testowanych metod regresji.