POLITECHNIKA KRAKOWSKA
im. T. Kościuszki

Metody Odkrywania Wiedzy II

Sprawozdanie z Projektu -
Regresja na podstawie danych wypożyczalni rowerów

Dariusz Nostkiewicz

Rafał Gęgotek

Kraków, 2022 r.

Spis Treści

1. Wprowadzenie

1.1. Cel prac

Celem prac jest wykonanie wszystkich działań przygotowujących dane do utworzenia modeli regresji na podstawie danych wypożyczalni rowerów oraz optymalizacja działania modeli.

1.2. Dane

Systemy wypożyczania rowerów to nowa generacja tradycyjnych wypożyczalni rowerów, w których cały proces od członkostwa, wypożyczenia i zwrotu stał się automatyczny. Dzięki tym systemom użytkownik może łatwo wypożyczyć rower z określonej pozycji i wrócić z powrotem w innej pozycji. Obecnie na całym świecie istnieje ponad 500 programów bike-sharingowych, na które składa się ponad 500 tysięcy rowerów. Obecnie istnieje duże zainteresowanie tymi systemami ze względu na ich ważną rolę w kwestiach ruchu drogowego, ochrony środowiska i zdrowia.

Oprócz ciekawych zastosowań systemów rowerów publicznych w świecie rzeczywistym, charakterystyka danych generowanych przez te systemy czyni je atrakcyjnymi dla badań. W przeciwieństwie do innych usług transportowych, takich jak autobusy czy metro, w tych systemach wyraźnie rejestruje się czas trwania podróży, miejsce odjazdu i przyjazdu. Ta funkcja zmienia system rowerów publicznych w wirtualną sieć czujników, która może być wykorzystywana do wykrywania mobilności w mieście. Stąd oczekuje się, że większość ważnych wydarzeń w mieście będzie można wykryć poprzez monitorowanie tych danych.

	yr	season	mnth	hr	holiday	weekday	weathersit
count	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000
mean	0.502561	2.501640	6.537775	11.546752	0.028770	3.003683	1.425283
std	0.500008	1.106918	3.438776	6.914405	0.167165	2.005771	0.639357
min	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000
25%	0.000000	2.000000	4.000000	6.000000	0.000000	1.000000	1.000000
50%	1.000000	3.000000	7.000000	12.000000	0.000000	3.000000	1.000000
75%	1.000000	3.000000	10.000000	18.000000	0.000000	5.000000	2.000000
max	1.000000	4.000000	12.000000	23.000000	1.000000	6.000000	4.000000

temp	atemp	hum	windspeed	casual	registered	cnt
17379.000000	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000	17379.000000
15.358397	15.401116	53.277922	12.736233	35.676218	153.786869	189.463088
9.050138	11.341858	28.760635	8.196891	49.305030	151.357286	181.387599
-7.060000	-16.000000	-1.000000	0.000000	0.000000	0.000000	1.000000
7.980000	6.000000	39.000000	7.000000	4.000000	34.000000	40.000000
15.500000	16.000000	57.000000	13.000000	17.000000	115.000000	142.000000
23.020000	25.000000	76.000000	17.000000	48.000000	220.000000	281.000000
39.000000	50.000000	100.000000	57.000000	367.000000	886.000000	977.000000

Rysunek 1. Rozkład danych w zbiorze

Zbiór został uporządkowany, nie zawiera wartości tekstowych. Jedynie parametr tyczący się pogody został ustawiony wg przedstawionych kryteriów:

1: Clear, Few clouds, Partly cloudy,

2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist,

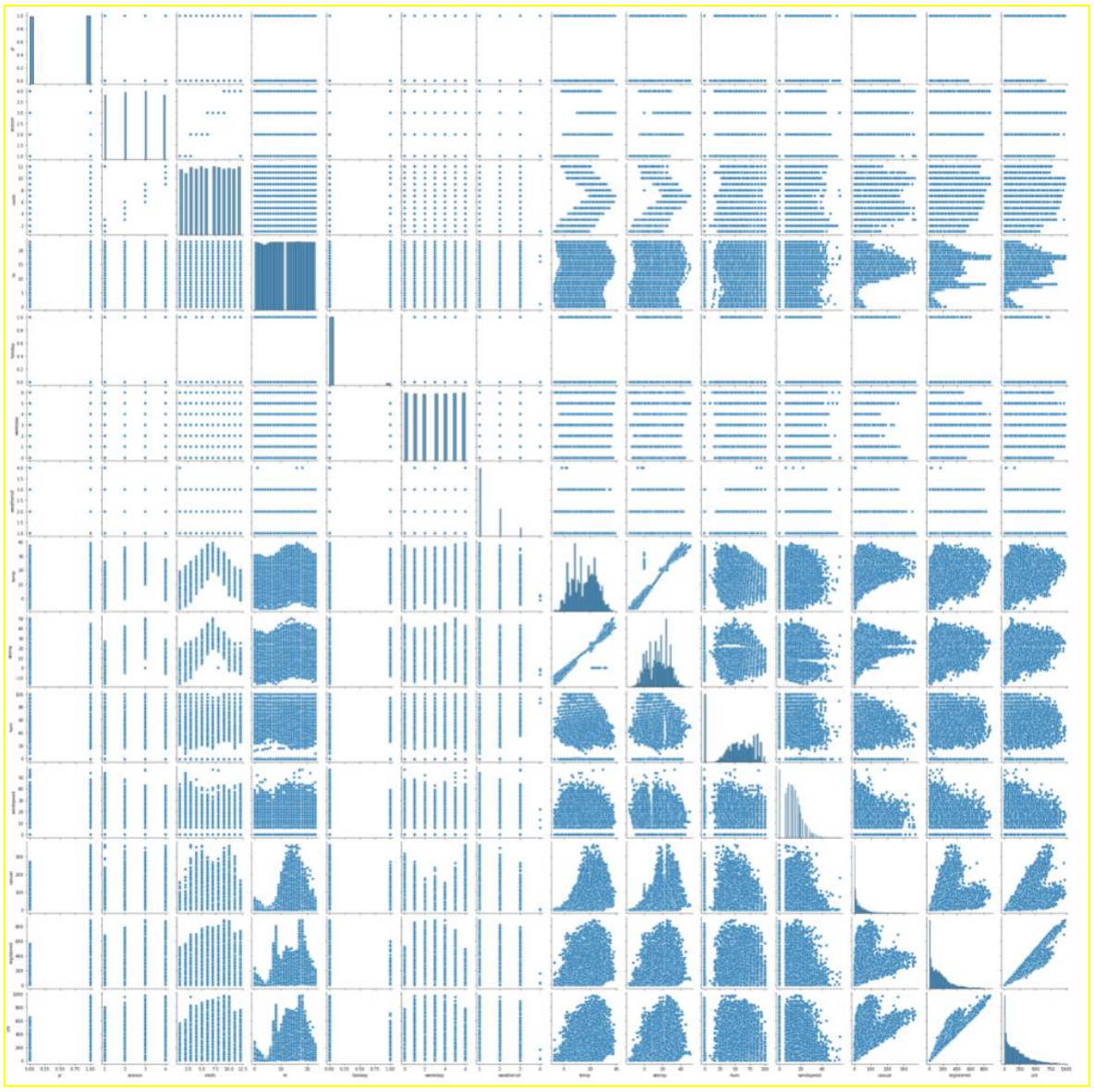
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds,

4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog.

Dodatkowo season, odpowiadający porom roku, jest przedstawiony w formie cyfr w kolejności chronologicznej: 1-zima, 2-wiosna, 3-lato, 4-jesień.

Pozostałe dane:

- yr – rok (0: 2011, 1:2012),
- mnth – miesiąc (od 1 do 12),
- hr – godzina (0 do 23),
- holiday – czy dzień jest dniem świątecznym (1), czy nie (0),
- weekday – numer dnia tygodnia (liczony od 0),
- temp – znormalizowana temperatura w stopniach Celsjusza. Wartości tworzone są wg wzoru: (t - t_min) / (t_max - t_min), t_min = -8, t_max = +39 (tylko w skali godzinowej),
- atemp – znormalizowana temperatura odczuwalna w stopniach Celsjusza. Wartości tworzone są wg wzoru: (t - t_min) / (t_max - t_min), t_min = -16, t_max = +50 (tylko w skali godzinowej),
- hum – znormalizowana wilgotność. Wartości są podzielone na 100 (maks.),
- windspeed – znormalizowana prędkość wiatru. Wartości są podzielone na 67 (maks.),
- casual – liczba niezarejestrowanych użytkowników,
- registered – liczba zarejestrowanych użytkowników,
- cnt – całkowita liczba wypożyczonych rowerów, w tym zarówno niezarejestrowanych, jak i zarejestrowanychane.

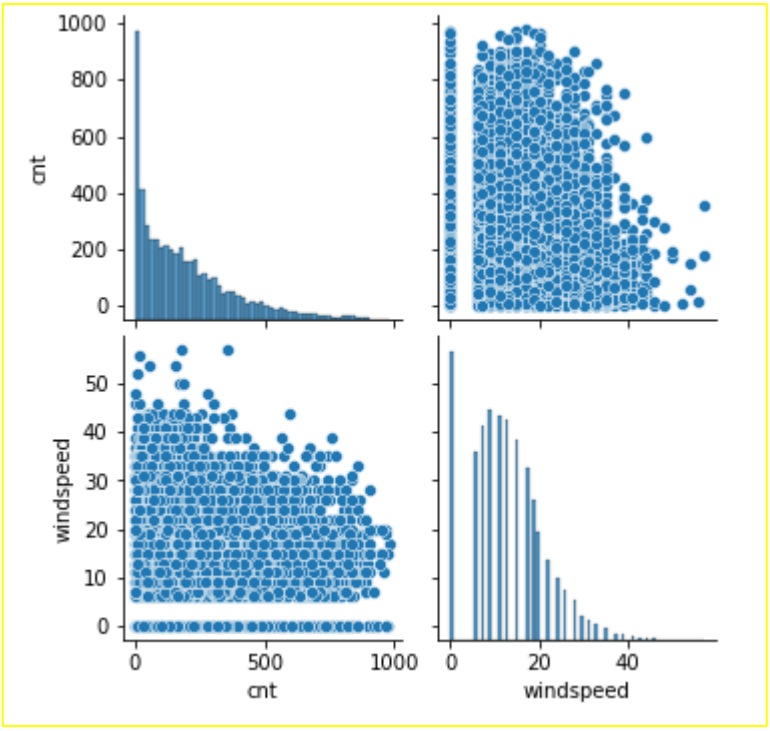


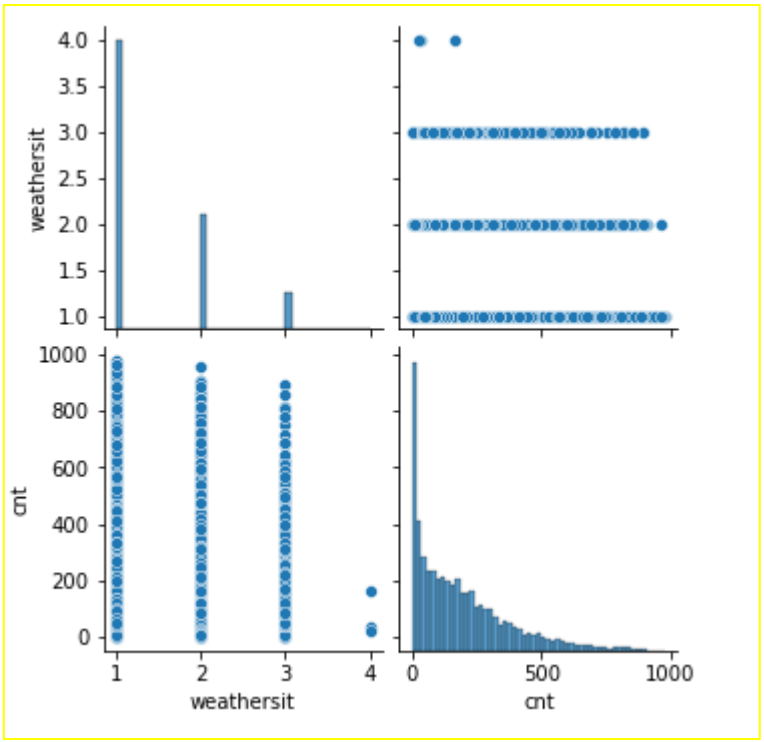
Rysunek 2. Wykres zależności danych

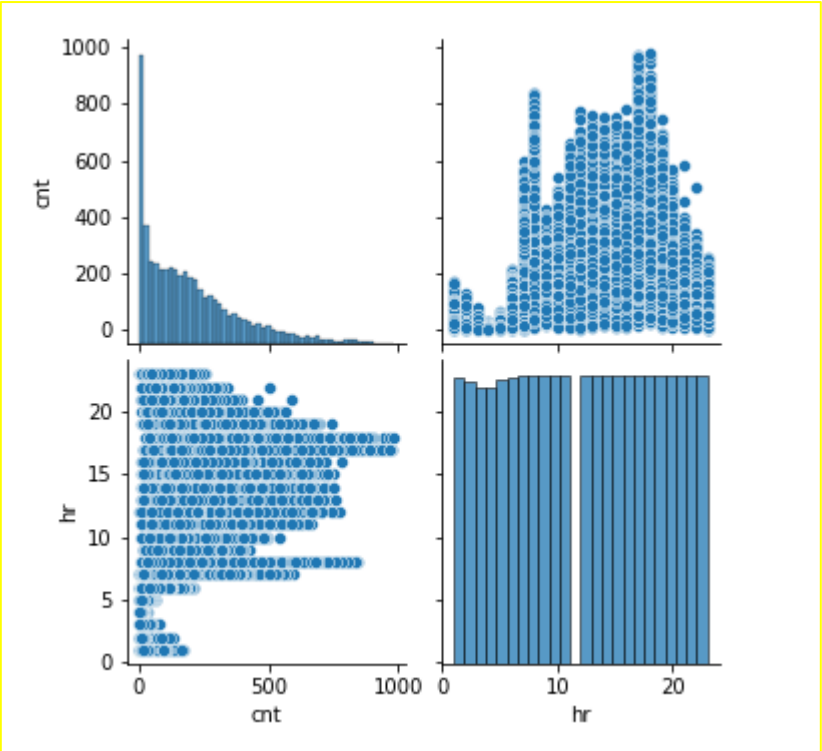
Po wykresach przedstawionych powyżej możemy doszukać się pierwszych korelacji:

- większy wiatr = mniej wypożyczeni,
- gorsza pogoda = mniej wypożyczeni,
- środek nocy = mniej wypożyczeni.

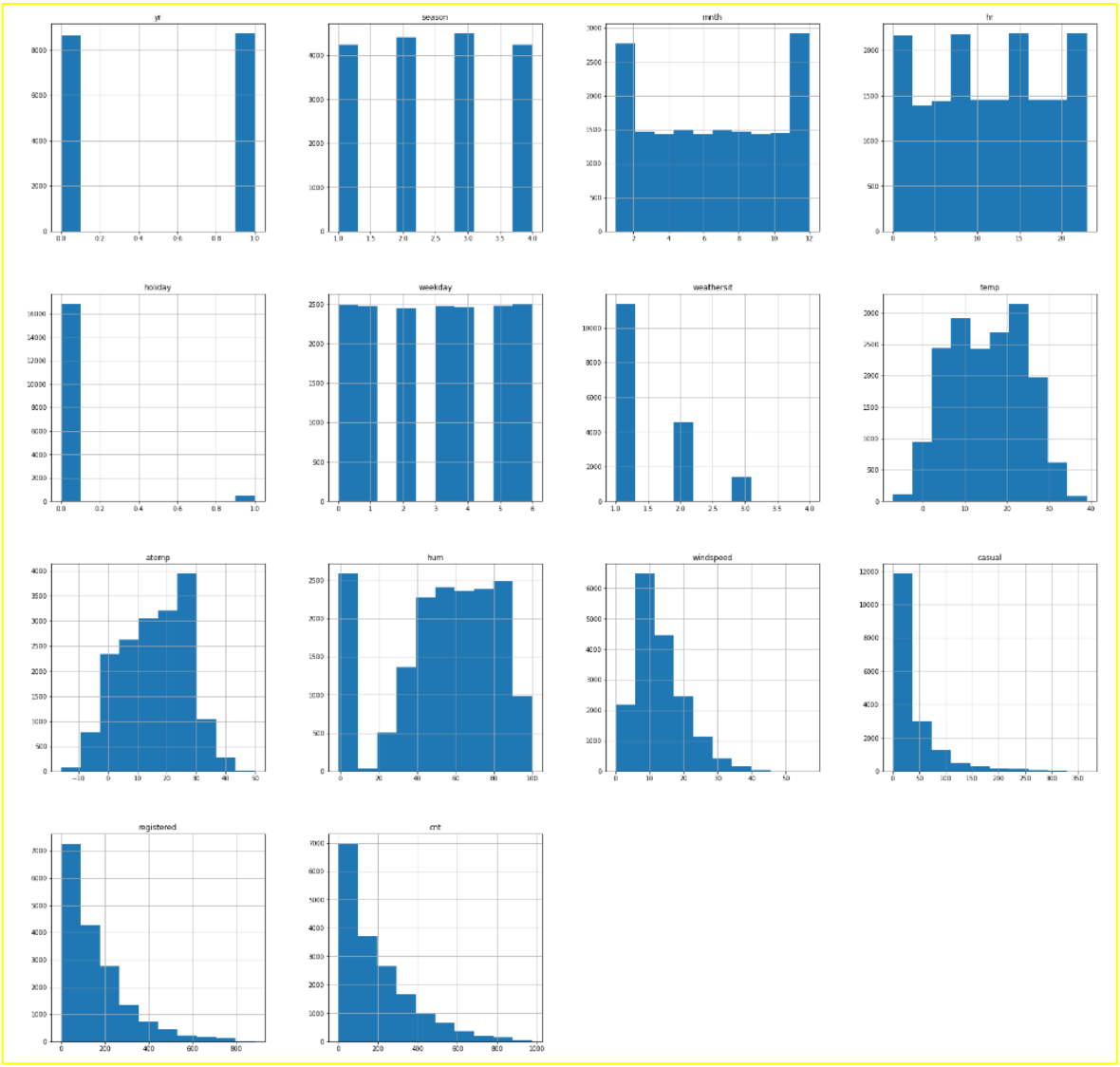
Widzimy również wartości odstające dla parametru wilgotności oznaczone jako -1.


cnt
windspeed
cnt
windspeed


weathersit
cnt
weathersit
cnt

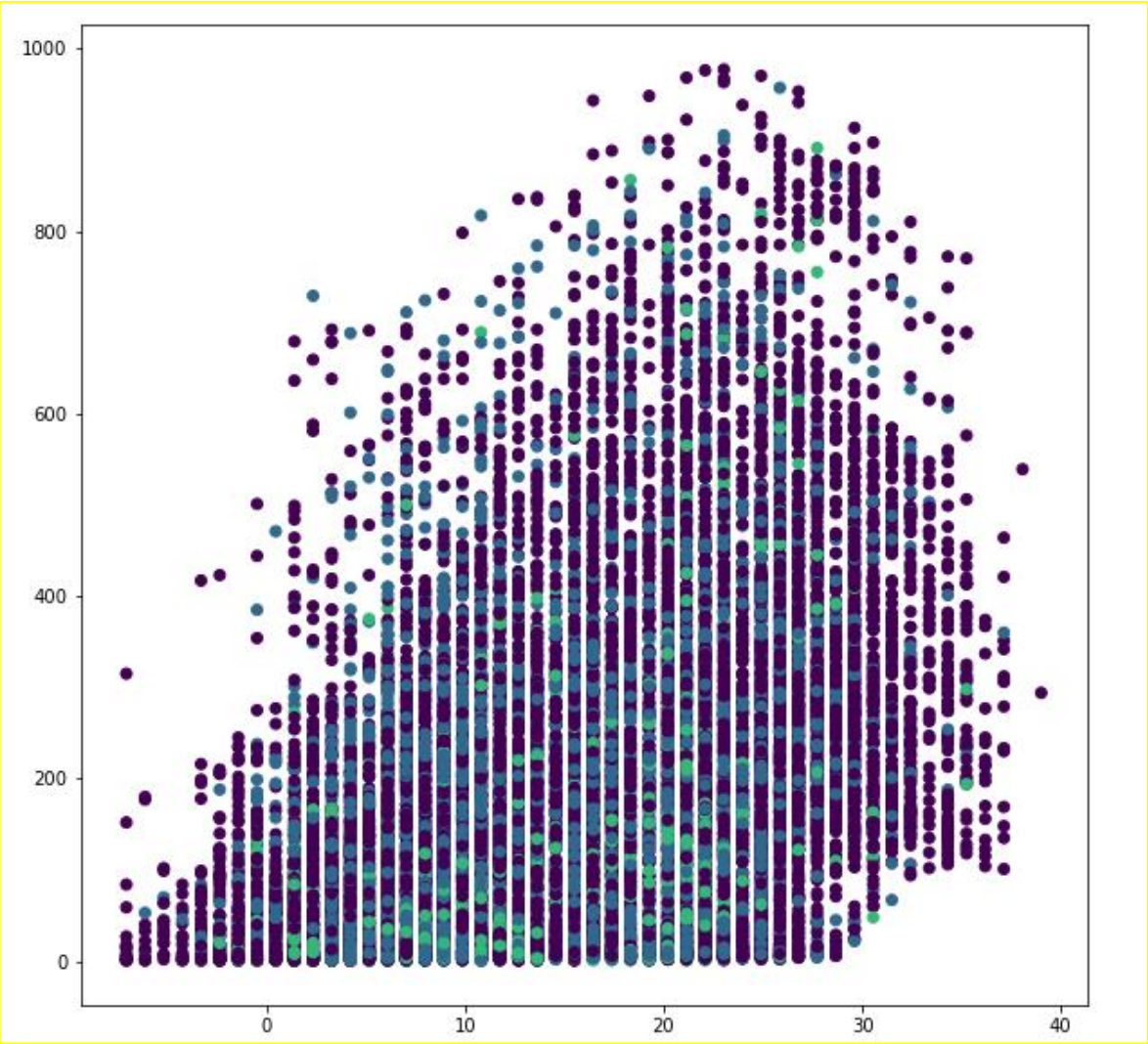


Rysunek 3. Szczegółowe wykresy zależności danych



Rysunek 4. Rozkład danych

W zbiorze istnieją dane odstające, jednak mają one odzwierciedlenie w rzeczywistym świecie (windspeed) lub dotyczą zmiennych objaśnianych (cnt, registered, casual).



Rysunek 5. Zależność ilości wypożyczeń (oś y) od temperatury (oś y) i panujących wtedy warunków atmosferycznych (kolor)

1.3. Etapy prac

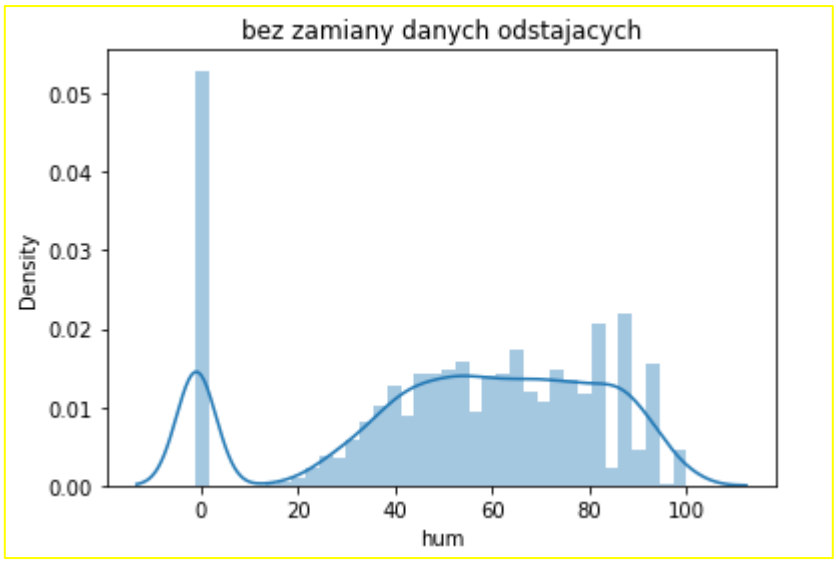
Prace zostały wykonane w następujących etapach:

- preprocessing – korekcja danych odstających, normalizacja i skalowanie danych,
- przygotowanie danych – shuffle i podział na zbiory treningowy i testowy,
- tworzenie modeli i algorytmów wyliczania miar skuteczności (MSE, R^2, EVS),
- Voting i Stacking Regressors,
- k-krotna walidacja krzyżowa,
- optymalizacja cech,
- optymalizacja hiperparametrów,
- podsumowanie działań.

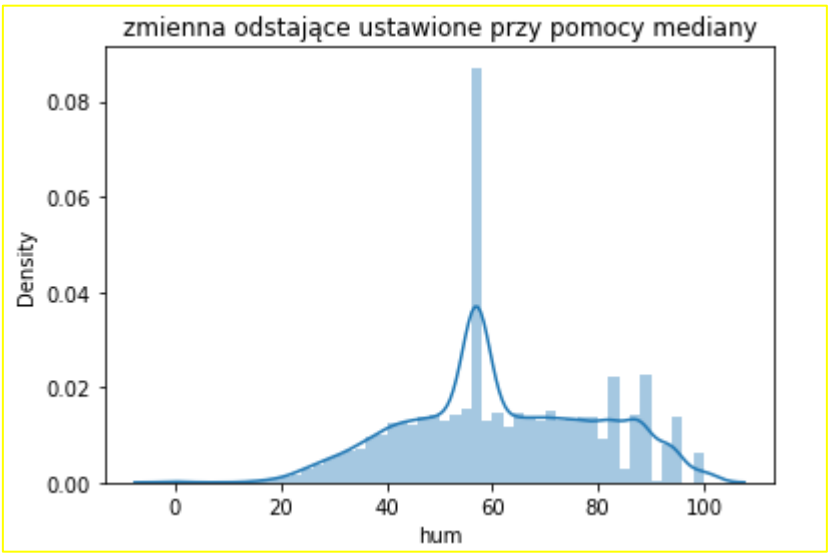
2. Preprocessing

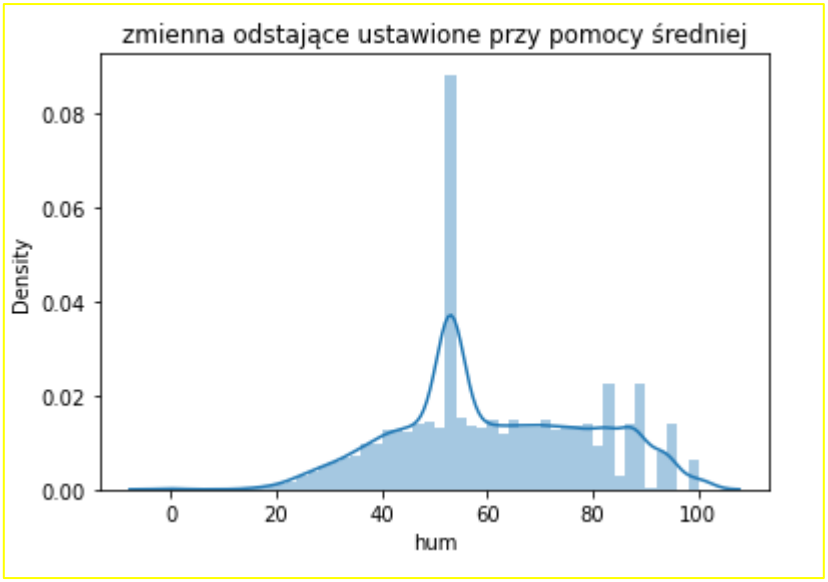
2.1. Korekcja danych odstających

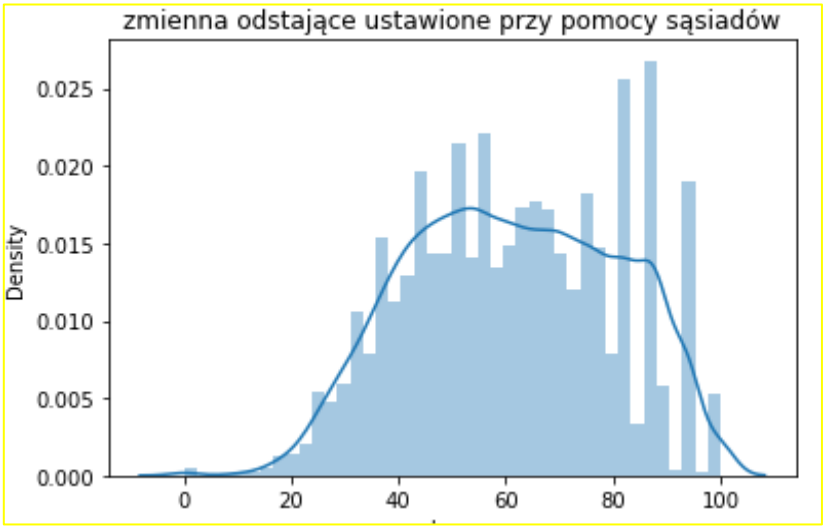
Przetestowano korekcję parametru *hum* z zastąpieniem przez medianę i średnią oraz przy użyciu algorytmu K-neighbours.



Rysunek 6. Zmienna hum przed korekcją







Rysunek 7. Testowane korekcje

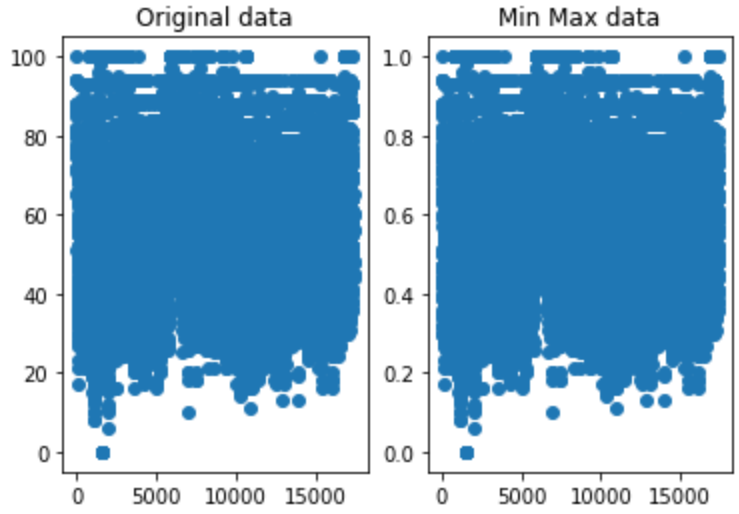
Możemy zaobserwować, że algorytm K-neighbours daje najlepsze wyniki, dane są równomiernie rozłożone. Wybraliśmy dane wygenerowane przez ten algorytm do dalszych działań.

2.2. Skalowanie danych

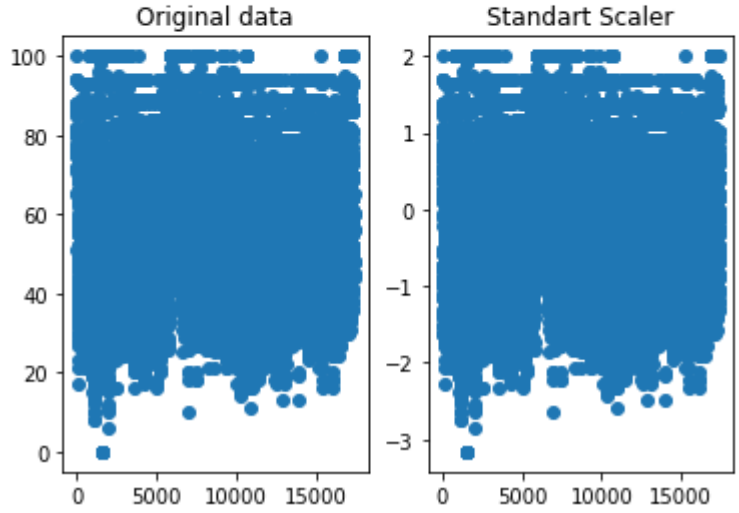
Testowaliśmy metody skalowania:

- MinMaxScaler – ustala wartości proporcjonalnie z przedziału 0 do 1 (dla odpowiedniego parametru od -1 do 1) pomiędzy wartościami min i max ze zbioru danych. Dzięki temu zachowamy kształt danych bez zniekształceń,
- StandardScaler – ustala wartość 0 na średnią ze zbioru i odpowiednio skaluje dane. Dla zobrazowanych przykładów przedział dla temperatury mieści się w zakresach ok. <-2.4; 2.6>, natomiast dla wilgotnosci <-3.1; 2>. Lepszy wybór dla naszych danych, gdyż są bliskie rozkładu normalnego. Wybraliśmy dane wygenerowane przez ten algorytm do dalszych działań.

```
 count    17379.000000
mean         0.612652
std          0.193748
min          0.000000
25%          0.460000
50%          0.610000
75%          0.770000
max          1.000000
Name: hum, dtype: float64
```



```
 count    1.737900e+04
mean    -2.620657e-16
std      1.000029e+00
min     -3.162205e+00
25%     -7.879139e-01
50%     -1.368857e-02
75%      8.121517e-01
max      1.999297e+00
Name: hum, dtype: float64
```



Rysunek 8. Wizualizacja skalowania na przykładzie parametru hum

3. Przygotowanie danych

Shuffle i podział na zbiory treningowy i testowy (80%, 20%). Zmienną objaśnianą jest *cnt*.

	yr	season	mnth	hr	holiday	weekday	weathersit	temp	atemp	hum	windspeed	cnt
6353	0	4	9	7	0	2	2	0.638860	0.405491	1.689607	-0.699826	249
8618	0	1	12	21	0	5	2	-0.711435	-0.652568	0.244387	-0.699826	95
14957	1	3	9	10	0	4	1	0.431122	0.405491	-0.426609	-0.821827	190
2674	0	2	4	8	0	2	1	0.431122	0.405491	1.121842	0.886187	449
13290	1	3	7	23	0	4	2	1.054335	1.022691	-1.407294	0.276182	211

Rysunek 9. Przykładowe dane po preprocessingu i shuffle

```
x_train.shape:  (13903, 11)
x_test.shape:  (3476, 11)
y_train.shape:  (13903, 1)
y_test.shape:  (3476, 1)
```

Rysunek 10. Wielkości podzbiorów

4. Tworzenie modeli i algorytmów wyliczania miar skuteczności (MSE, R^2, EVS)

Błąd średniokwadratowy, **średni błąd kwadratowy**, **MSE** (od ang. *mean square error*) estymatora $\hat{\theta}$ nieobserwowanego parametru θ definiowany jest jako:

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{E}((\hat{\theta} - \theta)^2)$$

MSE jest wartością oczekiwaną kwadratu „błędu", czyli różnicy między estymatorem a wartością estymowaną. Błąd średniokwadratowy spełnia tożsamość:

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{D}^2(\hat{\theta}) + (\mathrm{b}(\hat{\theta}))^2$$

gdzie:

$\mathrm{D}^2(\hat{\theta})$ – wariancja estymatora $\hat{\theta}$

$b(\hat{\theta}) = E[(\hat{\theta})] - \theta$ – obciążenie estymatora.

Obciążenie estymatora jest różnicą między wartością oczekiwaną estymatora a wartością szacowanego parametru.

Współczynnik determinacji R^2 – jedna z miar jakości dopasowania modelu do danych uczących.

Informuje o tym, jaka część zmienności (wariancji) zmiennej objaśnianej w próbie pokrywa się z korelacjami ze zmiennymi zawartymi w modelu. Jest on więc miarą stopnia, w jakim model pasuje do próby. Współczynnik determinacji przyjmuje wartości z przedziału [0;1] jeśli w modelu występuje wyraz wolny, a do estymacji parametrów wykorzystano metodę najmniejszych kwadratów. Jego wartości najczęściej są wyrażane w procentach. Dopasowanie modelu jest tym lepsze, im wartość R^2 jest bliższa jedności. Wyraża się on wzorem:

$$R^2 := \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \geqslant 0,$$

gdzie:

y_i – i-ta obserwacja zmiennej y,

$\hat{y}_i$ – wartość teoretyczna zmiennej objaśnianej (na podstawie modelu),

$\overline{y}$ – średnia arytmetyczna empirycznych wartości zmiennej objaśnianej.

EVS wyjaśnia rozproszenie błędów danego zbioru danych, a wzór jest zapisany w następujący sposób:

$$explained\ variance(y, \hat{y}) = 1 - \frac{Var(y - \hat{y})}{Var(y)}$$

Utworzone modele:

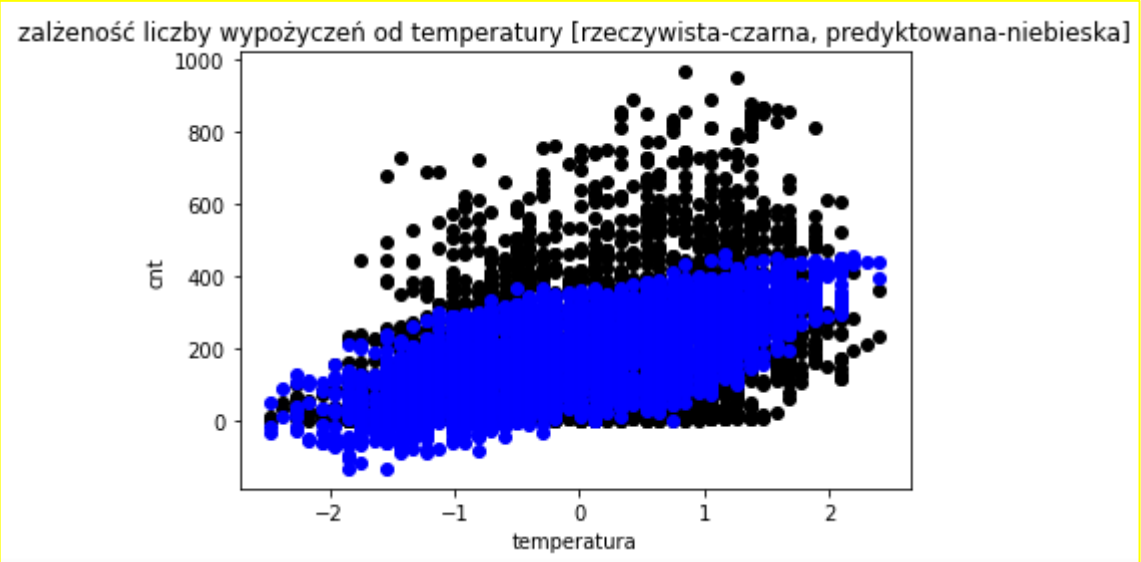
- Linear Reggresion,
- Polynomial Reggresion,
- Decision Tree Regression,
- Random Forrest Regression.

4.1. Linear Reggresion

```
RMSE of test set is 20177.21457
R2 score of test set is 0.38581
EVS score of test set is 0.38618


Custom MSE test set is 20177.21457
Custom r2 test set is 0.38581
Custom EVS test set is 0.38618
```

Rysunek 11. Miary skuteczności wyliczone przez nas i porównanie z miarami z dostępnych bibliotek



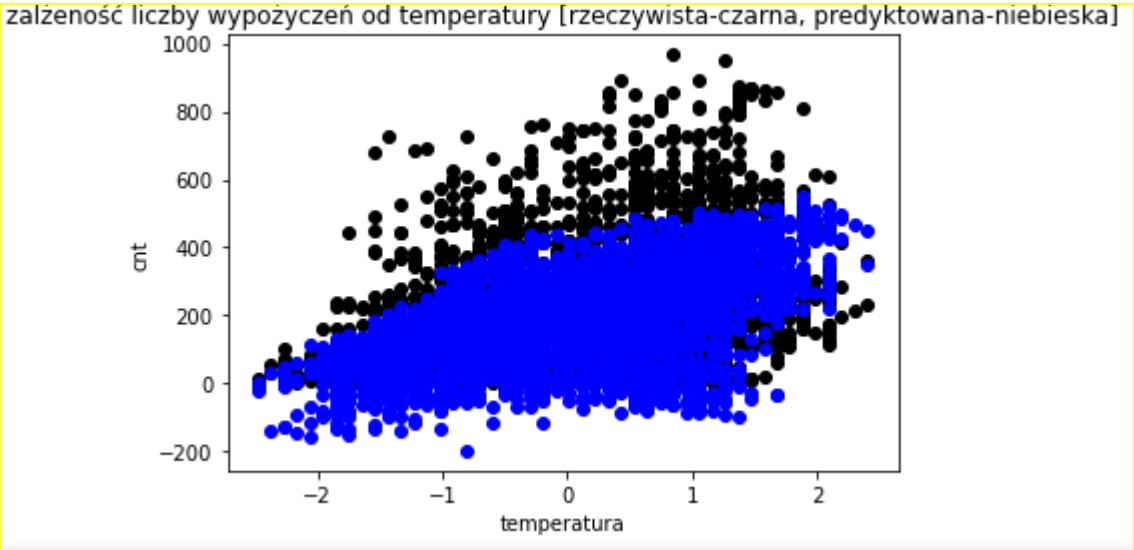
Rysunek 12. Przykład predykcji

4.2. Polynomial Reggresion

```
RMSE of test set is 15331.44124624092
R2 score of test set is 0.5333161872332124
EVS score of test set is 0.5334443326118306


Custom MSE test set is 15331.441246240927
Custom r2 test set is 0.5333161872332108
Custom EVS test set is 0.5334443326118306
```

Rysunek 13. Miary skuteczności wyliczone przez nas i porównanie z miarami z dostępnych bibliotek

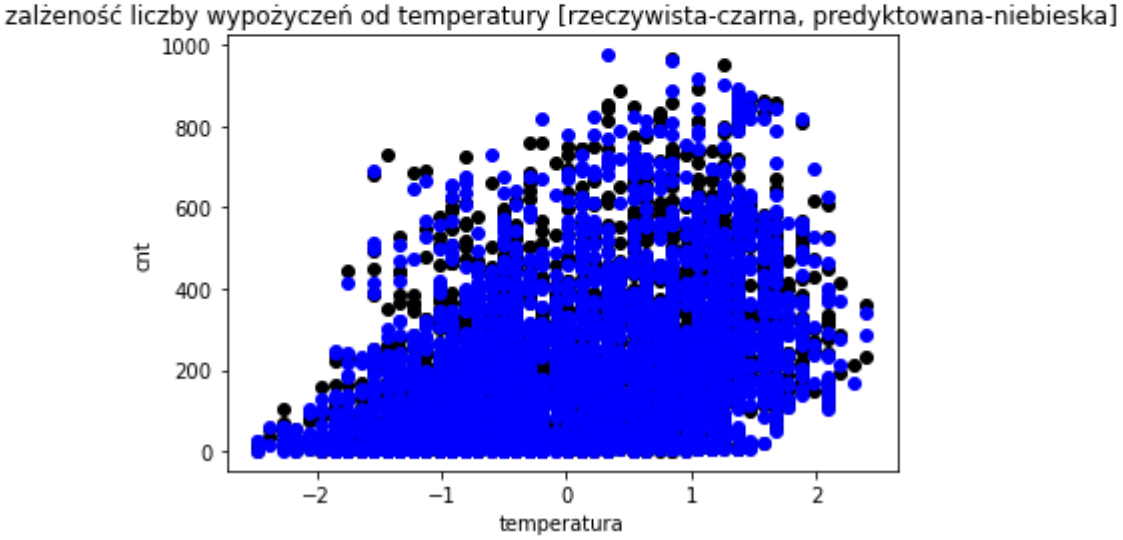


Rysunek 14. Przykład predykcji

4.3. Decision Tree Regression

```
RMSE of test set is 3550.1672180667433
R2 score of test set is 0.8919341276089569
EVS score of test set is 0.891939428094875


Custom MSE test set is 3550.1672180667433
Custom r2 test set is 0.8919341276089565
Custom EVS test set is 0.891939428094875
```

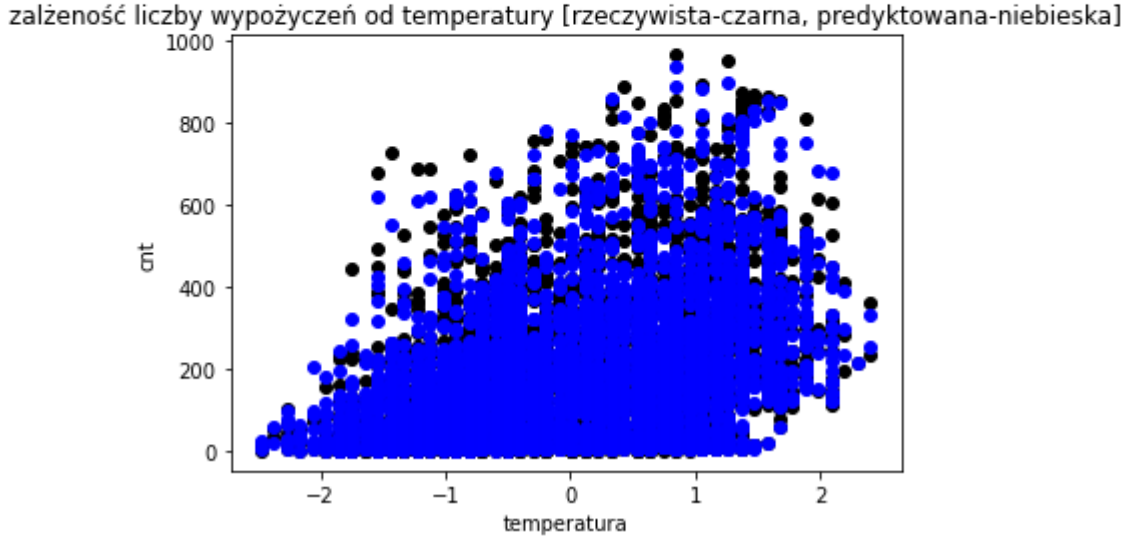


Rysunek 15. Miary skuteczności i przykład predykcji

4.4. Random Forrest Regression

```
RMSE of test set is 1978.924862631497
R2 score of test set is 0.9397622059636241
EVS score of test set is 0.9397646211356532


Custom MSE test set is 1978.9248626315004
Custom r2 test set is 0.9397622059636238
Custom EVS test set is 0.9397646211356532
```



Rysunek 15. Miary skuteczności i przykład predykcji

4.5. Podsumowanie modeli

Analizując powyższe wartości najlepiej poradził sobie algorytm Random Forrest Regression. Uzyskał najniższą wartość MSE i najwyższe R^2 oraz Experience Variance (0.954).

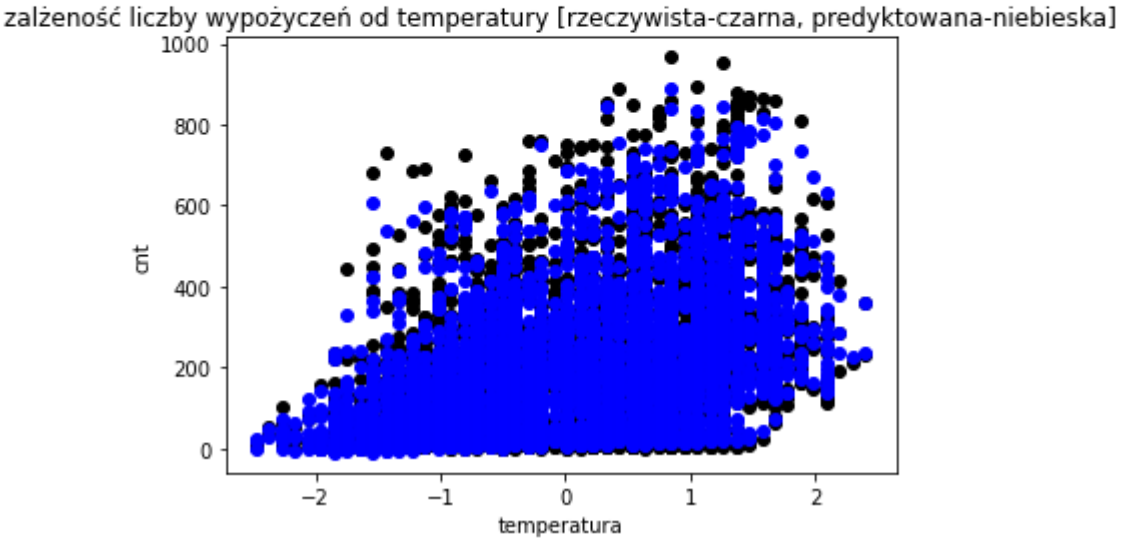
5. Voting i Stacking Regressors

5.1. Voting Regressor

Regresor prognoz głosowania dla niedopasowanych estymatorów. Voting Regressor to metaestymator zbiorowy, który pasuje do kilku regresorów bazowych, każdy w całym zbiorze danych. Następnie uśrednia poszczególne prognozy, aby utworzyć ostateczną prognozę.

```
RMSE of test set is 2334.46848114314
R2 score of test set is 0.928939580169549
EVS score of test set is 0.9289400237981827


Custom MSE test set is 2334.4684811431407
Custom r2 test set is 0.9289395801695488
Custom EVS test set is 0.9289400237981827
```



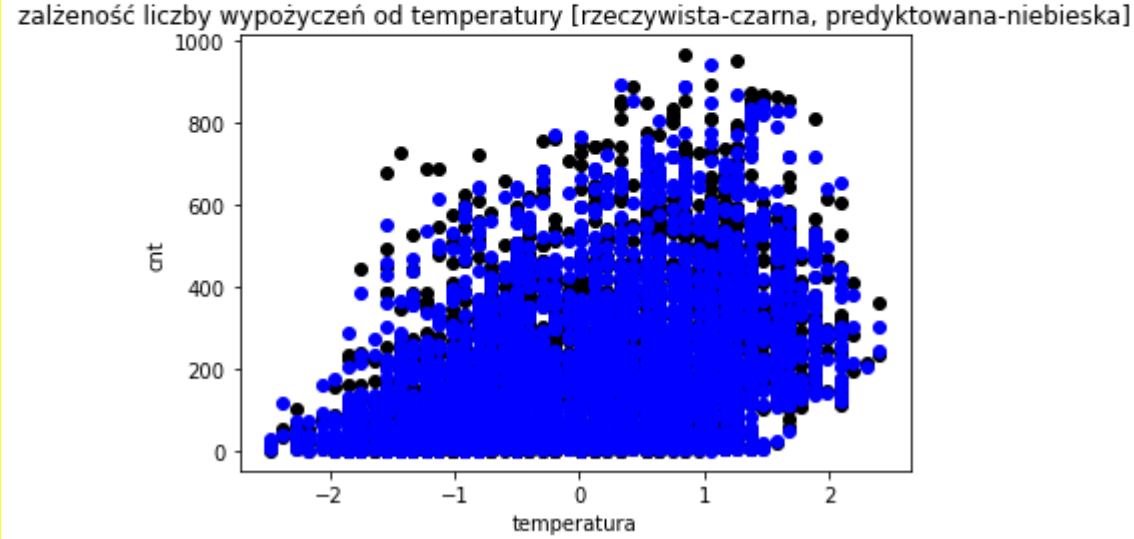
Rysunek 16. Miary skuteczności i przykład predykcji

5.2. Stacking Regressor

Stacking to sposób na zestawienie wielu klasyfikacji lub modelu regresji. Celem stackingu jest zbadanie przestrzeni różnych modeli dla tego samego problemu. Chodzi o to, że możesz zaatakować problem uczenia się różnymi typami modeli, które są w stanie nauczyć się jakiejś części problemu, ale nie całej przestrzeni problemu. Możesz więc zbudować wiele różnych uczniów i użyć ich do zbudowania prognozy pośredniej, jednej prognozy dla każdego wyuczonego modelu. Następnie dodajesz nowy model, który uczy się z prognoz pośrednich tego samego celu.

```
RMSE of test set is 2323.7025089653976
R2 score of test set is 0.9292672926698518
EVS score of test set is 0.9293298760727874


Custom MSE test set is 2323.702508965403
Custom r2 test set is 0.9292672926698514
Custom EVS test set is 0.9293298760727874
```



Rysunek 17. Miary skuteczności i przykład predykcji

6. K-krotna walidacja krzyżowa

Walidacja krzyżowa pomaga w ocenie modeli uczenia maszynowego. Ta metoda statystyczna pomaga w porównywaniu i wyborze modelu w stosowanym uczeniu maszynowym. Zrozumienie i wdrożenie tego problemu modelowania predykcyjnego jest łatwe i proste. Technika ta ma niższą tendencyjność podczas szacowania umiejętności modelu. Ten artykuł pomoże Ci zrozumieć koncepcję k-krotnej walidacji krzyżowej i jak możesz ocenić model uczenia maszynowego za pomocą tej techniki. K-krotna walidacja krzyżowa oznacza, że zbiór danych jest podzielony na K liczb. Dzieli ona zbiór danych w punkcie, w którym zestaw testowy wykorzystuje każdą fałdę. Zrozummy tę koncepcję z pomocą 5-krotnej walidacji krzyżowej lub K+5. W tym scenariuszu, metoda podzieli zbiór danych na pięć fałd. Model używa pierwszej fałdy w pierwszej iteracji do testowania modelu. Pozostałe zbiory danych są wykorzystywane do trenowania modelu. Druga fałda pomaga w testowaniu zbioru danych, a pozostałe wspierają proces szkolenia. Ten sam proces powtarza się aż do momentu, gdy zestaw testowy wykorzystuje każdą fałdę z pięciu fałd.

Summary table for result of regression models:

model	MSE	r2	Experience Variance
Linear Regression	20177.214574539303	0.3858125092467478	0.38617685093263976
Polynominal Regression	15331.441246240927	0.5333161872332108	0.5334443326118306
Decision Tree Regression	3468.1352848101264	0.8944311515198831	0.8944385829724519
Random Forrest Regression	1958.7379889773715	0.9403766874734467	0.9403837685942139
Voting Regressor	2281.773095008165	0.9305436096486899	0.9305445620336064
Stacking Regressor	2295.596934026953	0.9301228167306108	0.9301665163087851

Summary table for result of regression models [K_fold: k = 2]:

model	MSE	r2	Experience Variance
Linear Regression	20278.579311308487	0.3832969020000827	0.38374151610148005
Polynominal Regression	15333.150165596178	0.5336953327827012	0.5338575342549557
Decision Tree Regression	4441.945620899989	0.8649374291018264	0.8652062110306231
Random Forrest Regression	2347.203388999963	0.9286297071333854	0.9287588600005028
Voting Regressor	2837.435013826347	0.9137179829988921	0.9138133559666248
Stacking Regressor	2642.401180302677	0.919653941890119	0.9196785858922398

Summary table for result of regression models [K_fold: k = 3]:

model	MSE	r2	Experience Variance
Linear Regression	20265.23396805925	0.3838992900299106	0.38402418715673026
Polynominal Regression	15281.298402924518	0.5354499568241892	0.5355266531212742
Decision Tree Regression	3839.092050750906	0.8832922706188496	0.8833895715653631
Random Forrest Regression	2082.2697910794454	0.9367081863527013	0.9367971076296194
Voting Regressor	2521.5517935591065	0.923348168554865	0.9234104019207164
Stacking Regressor	2400.99138740361	0.9270117609657031	0.92703457980392

Summary table for result of regression models [K_fold: k = 5]:

model	MSE	r2	Experience Variance
Linear Regression	20243.75996701301	0.3843045899754628	0.38455966920197904
Polynominal Regression	15278.22267516013	0.535283353688409	0.5354184432259392
Decision Tree Regression	3809.0320575539567	0.8839880565354135	0.8840356186327056
Random Forrest Regression	1999.502109896382	0.9391179872192019	0.9392023164319033
Voting Regressor	2475.3274867679197	0.9246433244916918	0.9246943377081556
Stacking Regressor	2294.4201767997665	0.9301122137435215	0.9301276737411005

Summary table for result of regression models [K_fold: k = 10]:

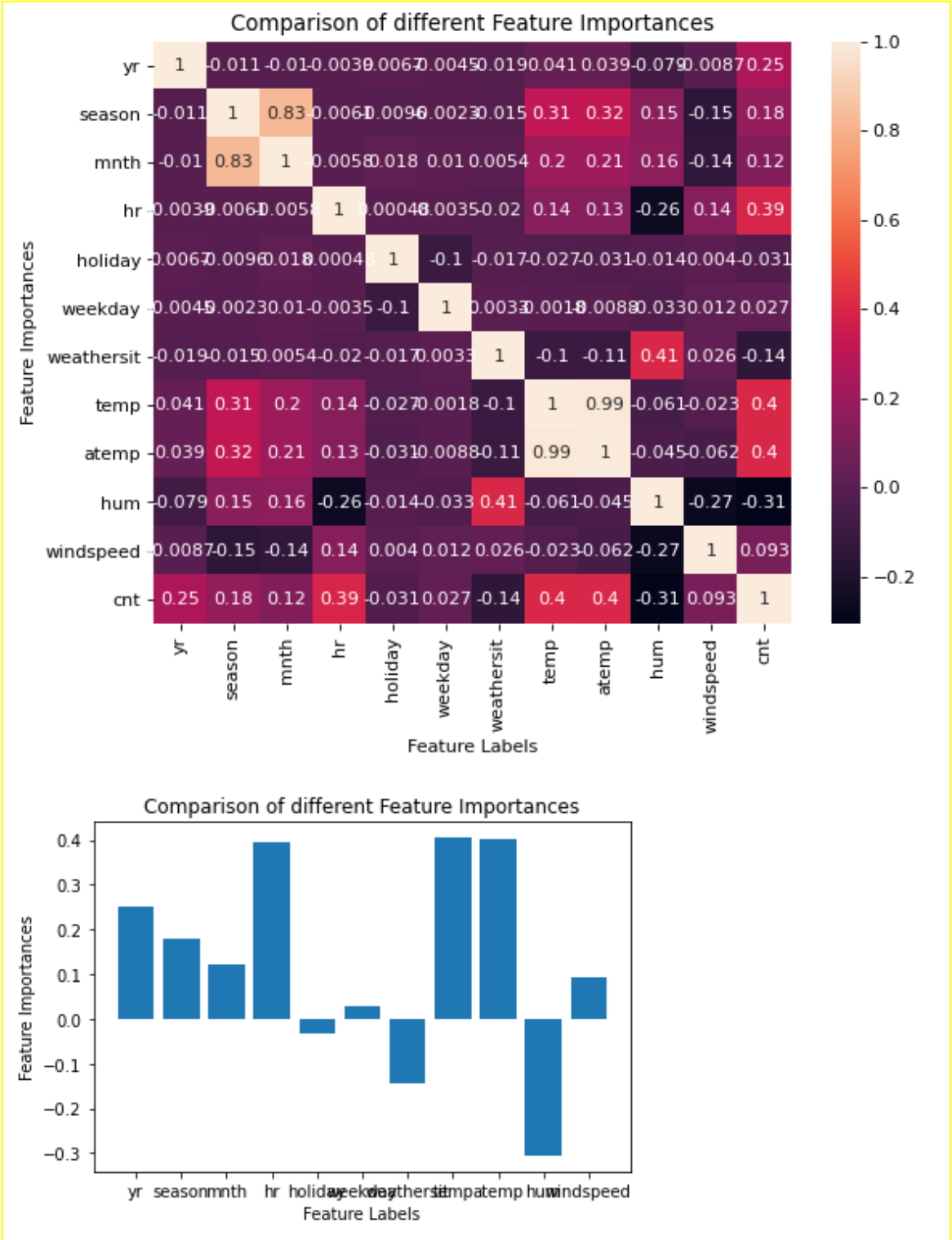
model	MSE	r2	Experience Variance
Linear Regression	20248.827752581157	0.38391313446448805	0.38423261944929255
Polynominal Regression	15273.313258221178	0.5352975518913959	0.5354564522731133
Decision Tree Regression	3638.627014968336	0.8890959108785136	0.889208368069658
Random Forrest Regression	1944.1930595351587	0.9408006956365407	0.9409073191920031
Voting Regressor	2376.3498514408734	0.9276225892859079	0.9276946134791852
Stacking Regressor	2224.8749610890436	0.9322350401649453	0.9322704624203677

Rysunek 18. Skuteczność modeli w zależności od wielkości k

K-krotną walidacja krzyżowa daje delikatną poprawę względem wyników bez jej zastosowania.

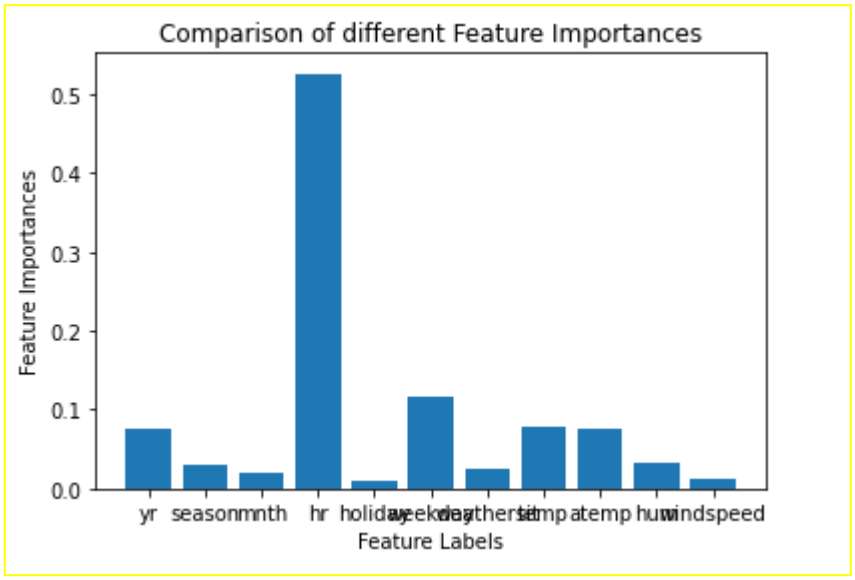
7. Optymalizacja cech

Dokonana zostanie optymalizacja poprzez wykluczenie poszczególnych parametrów na podstawie korelacji ze zmienną objaśnianą oraz wyników ExtraTreeRegressor.



Rysunek 19. Korelacje parametrów ze zmienną cnt

ExtraTreeRegressor to niezwykle losowy regresor. Extra-trees różnią się od klasycznych drzew decyzyjnych sposobem, w jaki są zbudowane. Szukając najlepszego podziału w celu oddzielenia próbek węzła na dwie grupy, losowe podziały są losowane dla każdej z losowo wybranych funkcji max_features i wybierany jest najlepszy podział spośród nich. Gdy max_features jest ustawione na 1, oznacza to zbudowanie całkowicie losowego drzewa decyzyjnego. Na podstawie wyników zwracana jest istotność parametrów.



Rysunek 20. Wyniki ExtraTreeRegressor

Na podstawie wyników zrezygnowaliśmy z parametrów:

- season,
- mnth,
- holiday,
- weathersit,
- temp,
- hum,
- windspeed.

Summary table for result of regression models:

model	MSE	r2	Experience Variance
Linear Regression	21867.761326676096	0.3343528459785504	0.3347091750508546
Polynominal Regression	16882.76129072206	0.4860945371905242	0.4862972295217204
Decision Tree Regression	6548.333144683483	0.8006709851901681	0.8007192480173333
Random Forrest Regression	4938.322646535097	0.849679152511337	0.849759701971445
Voting Regressor	5330.180580392806	0.8377511314141377	0.8378398952641618
Stacking Regressor	5085.8598129064085	0.8451881717730549	0.8453532576464577

Summary table for result of regression models [K_fold: k = 2]:

model	MSE	r2	Experience Variance
Linear Regression	21852.534427927887	0.3354442444594259	0.3361777196572146
Polynominal Regression	16767.601273881326	0.4900851089168581	0.49028485189365023
Decision Tree Regression	6810.154959323578	0.7929440207524521	0.7930278995910091
Random Forrest Regression	4760.248277735413	0.8552691794963276	0.8553413823081242
Voting Regressor	5202.666394819316	0.8418108054584661	0.8418373906868811
Stacking Regressor	5054.365029176177	0.8463069614289742	0.8463447328259901

Summary table for result of regression models [K_fold: k = 5]:

model	MSE	r2	Experience Variance
Linear Regression	21834.104125653124	0.3359354637996934	0.3362870396391586
Polynominal Regression	16752.374936075925	0.4904855441825569 7	0.4906047109084241
Decision Tree Regression	6482.349172402729	0.8026443722921606	0.8027119191503221
Random Forrest Regression	4924.441155273438	0.8501105076440231	0.8501613555002017
Voting Regressor	5266.512596264095	0.8397075921382969	0.8397521862442302
Stacking Regressor	5058.835115033748	0.8460271269213869	0.8460700042264975

Summary table for result of regression models [K_fold: k = 10]:

model	MSE	r2	Experience Variance
Linear Regression	21832.466782227293	0.33583221802669283	0.3362483998401554
Polynominal Regression	16749.977315482396	0.4904812706666711	0.4906240663263284
Decision Tree Regression	6287.268965541969	0.8087236662690908	0.8087955746886106
Random Forrest Regression	4881.265953269723	0.8514385514701914	0.8515128488996311
Voting Regressor	5175.055418348937	0.842544850431907	0.8425944841986936
Stacking Regressor	5139.543293309647	0.8436259443820863	0.8436787771375833

Rysunek 21. Wyniki działania po rezygnacji z wymienionych cech

Uzyskaliśmy dobrą optymalizację na rzecz kilku procent straty skuteczności, która jest akceptowalna.

8. Optymalizacja hiperparametrów

Dokonamy wyboru optymalnych parametrów modeli na podstawie wyników GridSearchCV.

GridSearchCV to wyczerpujące wyszukiwanie określonych wartości parametrów dla estymatora. GridSearchCV implementuje metodę „dopasowania" i „punktacji". Implementuje również „score_samples", „predict", „predict_proba", „decision_function", „transform" i „inverse_transform", jeśli są zaimplementowane w używanym estymatorze. Parametry estymatora używanego do zastosowania tych metod są optymalizowane przez krzyżowo zweryfikowane wyszukiwanie w siatce parametrów.

Summary table for result of regression models:

model	MSE	r2	Experience Variance
Linear Regression	21867.761326676096	0.3343528459785504	0.3347091750508546
Polynominal Regression	16882.76129072206	0.4860945371905242	0.4862972295217204
Decision Tree Regression	4572.403653294725	0.8608176011533439	0.8608542174081296
Random Forrest Regression	4425.77062718543	0.8652810601721754	0.8653671637414859
Voting Regressor	4400.150993919807	0.8660609130211057	0.8661409747711207
Stacking Regressor	4748.354352767109	0.8554617222134635	0.855696766384507

Summary table for result of regression models [K_fold: k = 2]:

model	MSE	r2	Experience Variance
Linear Regression	21852.534427927887	0.3354442444594259	0.3361777196572146
Polynominal Regression	16767.601273881326	0.4900851089168581	0.49028485189365023
Decision Tree Regression	4693.456402843128	0.8572972649884676	0.8573437556537823
Random Forrest Regression	4242.832770485991	0.8710029612701828	0.8710571991151771
Voting Regressor	4357.888068944316	0.8674972046026106	0.8675176282943734
Stacking Regressor	4738.486799124657	0.8559231156011038	0.8559801279057241

Summary table for result of regression models [K_fold: k = 3]:

model	MSE	r2	Experience Variance
Linear Regression	21848.93828694247	0.33575372997527214	0.33590381177728473
Polynominal Regression	16757.044923113237	0.49058255448932603	0.4906468092867815
Decision Tree Regression	4464.022847981625	0.8642639672033822	0.8642885256812245
Random Forrest Regression	4286.108368286621	0.8696652034122421	0.8697023744397013
Voting Regressor	4312.105734402576	0.8688807007978531	0.8689007898189024
Stacking Regressor	4620.908270026574	0.8594832934378384	0.8595006552833117

Summary table for result of regression models [K_fold: k = 5]:

model	MSE	r2	Experience Variance
Linear Regression	21834.104125653124	0.3359354637996934	0.3362870396391586
Polynominal Regression	16752.374936075925	0.49048554418255697	0.4906047109084241
Decision Tree Regression	4466.541357030719	0.8640096276721249	0.8640455255990214
Random Forrest Regression	4366.0926999322965	0.8670612176324903	0.8671159311824607
Voting Regressor	4349.497507430971	0.867607863714386	0.8676420082390248
Stacking Regressor	4623.217212527495	0.8592378491329725	0.8592781010079733

Summary table for result of regression models [K_fold: k = 10]:

model	MSE	r2	Experience Variance
Linear Regression	21832.466782227293	0.33583221802669283	0.3362483998401554
Polynominal Regression	16749.977315482396	0.4904812706666711	0.4906240663263284
Decision Tree Regression	4373.33613305712	0.8668423413462938	0.8668839357658529
Random Forrest Regression	4341.3259500878485	0.8678471567067432	0.8679150006770552
Voting Regressor	4300.802262480799	0.8691087432907812	0.8691508361822808
Stacking Regressor	4512.862908488292	0.8626492511755426	0.862704706056553

Rysunek 22. Wyniki po optymalizacji hiperparametrów

9. Podsumowanie działań

linear regression statistics:

model	MSE	r2	Experience Variance
standard	20177.21457	0.38581	0.38618
std & k_fold k=2	20278.57931	0.3833	0.38374
std & k_fold k=5	20243.75997	0.38431	0.38456
std & k_fold k=10	20248.82775	0.38391	0.38423
reduce	21867.76133	0.33435	0.33471
reduce & k_fold k=2	21852.53442	0.33544	0.33618
reduce & k_fold k=5	21834.10413	0.33594	0.33629
reduce & k_fold k=10	21832.46678	0.33583	0.33625

polynominal regression statistics:

model	MSE	r2	Experience Variance
standard	15331.44125	0.53332	0.53344
std & k_fold k=2	15333.15016	0.5337	0.53386
std & k_fold k=5	15278.22267	0.53528	0.53542
std & k_fold k=10	15273.31326	0.5353	0.53546
reduce	16882.76129	0.48609	0.4863
reduce & k_fold k=2	16767.60127	0.49008	0.49028
reduce & k_fold k=5	16752.37494	0.49048	0.49061
reduce & k_fold k=10	16749.97732	0.49048	0.49062

decision_tree regression statistics:

model	MSE	r2	Experience Variance
standard	3612.6793	0.89003	0.89003
std & k_fold k=2	4363.24963	0.86733	0.86743
std & k_fold k=5	3732.32085	0.88646	0.88649
std & k_fold k=10	3491.96919	0.89358	0.8937
reduce	6518.88832	0.80157	0.80162
reduce & k_fold k=2	6805.58552	0.79308	0.79315
reduce & k_fold k=5	6500.2194	0.80212	0.80219
reduce & k_fold k=10	6267.52543	0.80928	0.80935
reduce & optiamalized	4572.40365	0.86082	0.86085
reduce & opt & k_fold k=2	4693.4564	0.8573	0.85734
reduce & opt & k_fold k=5	4466.82233	0.864	0.86404
reduce & opt & k_fold k=10	4373.26114	0.86684	0.86689

random_forest regression statistics:

model	MSE	r2	Experience Variance
standard	1946.41234	0.94075	0.94075
std & k_fold k=2	2364.07682	0.92812	0.92828
std & k_fold k=5	2003.74212	0.93897	0.93905
std & k_fold k=10	1940.9042	0.94091	0.94102
reduce	4932.40622	0.84986	0.84992
reduce & k_fold k=2	4743.57245	0.85578	0.85584
reduce & k_fold k=5	4934.54551	0.84979	0.84985
reduce & k_fold k=10	4884.07764	0.85136	0.85143
reduce & optiamalized	4122.7762	0.8745	0.87456
reduce & opt & k_fold k=2	4097.08766	0.87544	0.87547
reduce & opt & k_fold k=5	4070.1574	0.87606	0.8761
reduce & opt & k_fold k=10	4023.28224	0.87751	0.87756

stacking regression statistics:

model	MSE	r2	Experience Variance
standard	2317.14548	0.92947	0.92958
std & k_fold k=2	2716.01133	0.91742	0.91743
std & k_fold k=5	2310.94691	0.9296	0.92963
std & k_fold k=10	2189.13279	0.93335	0.93339
reduce	5139.91393	0.84354	0.84372
reduce & k_fold k=2	5036.62883	0.84685	0.84688
reduce & k_fold k=5	5135.61573	0.84372	0.84379
reduce & k_fold k=10	5083.6991	0.8453	0.84535
reduce & optiamalized	4424.84442	0.86531	0.86544
reduce & opt & k_fold k=2	4438.81142	0.86504	0.86506
reduce & opt & k_fold k=5	4243.76371	0.87075	0.87081
reduce & opt & k_fold k=10	4232.28648	0.87114	0.87119

voting regression statistics:

model	MSE	r2	Experience Variance
standard	2302.3158	0.92992	0.92992
std & k_fold k=2	2841.9502	0.91359	0.91368
std & k_fold k=5	2460.38114	0.92509	0.92514
std & k_fold k=10	2357.32098	0.92819	0.92826
reduce	5299.25018	0.83869	0.83878
reduce & k_fold k=2	5229.55062	0.841	0.84102
reduce & k_fold k=5	5269.47426	0.83962	0.83966
reduce & k_fold k=10	5171.83914	0.84262	0.84267
reduce & optiamalized	4329.69908	0.86821	0.86827
reduce & opt & k_fold k=2	4342.85767	0.86795	0.86797
reduce & opt & k_fold k=5	4285.54678	0.86955	0.86959
reduce & opt & k_fold k=10	4232.20396	0.87119	0.87123

Rysunek 23. Podsumowanie działań

Wykonano zadania według założeń. Wyniki są zadowalające. Dokonaliśmy znacznej optymalizacji działania kosztem kilku procent skuteczności. Jednak modele nadal są wysoce skuteczne.

10. Wnioski