Metody odkrywania wiedzy
Dokumentacja końcowa

Rafał Okuniewski, Maciej Zaborek

czerwiec 2017

1 Wstęp

W ramach projektu zajęliśmy się tematem analitycznym dotyczącym regresji na zbiorze danych dokumentującym liczbę wypożyczeń rowerów miejskich.

Celem projektu było zapoznanie się z istniejącymi rozwiązaniami dotyczącymi algorytmów regresji, wykorzystanie ich na dostepnym zbiorze, przeprowadzenie prawidłowych badań tych algorytmów oraz ich dostosowanie w celu osiągnięcia jak najmniejszego błędu na dostępnym zbiorze danych testowych. Celem projektu było również wykorzystanie algorytmów umożliwiających ocenę i selekcję atrybutów a także analiza potencjału wprowadzenia nowych atrybutów do zbioru na podstawie analizy istniejących oraz wiedzy pozyskanej ze zbioru. W ramach przeprowadzonych badań skorzystaliśmy z pięciu dostępnych w języku R algorytmów regresji: regresji liniowej, regresji lokalnej, robust regression z M-estymacją, drzewa regresji oraz regresji z wykorzystaniem algorytmu sieci neuronowej.

Przeprowadzona analiza miała na zadaniu znalezienie najlepiej radzącego sobie z postawionym zadaniem algorytmu, jak i zbadaniem wpływu parametrów poszczególnych z nich na otrzymywany wynik.

2 Zbiór danych

W projekcie wykorzystano zbiór danych dostępny pod adresem: *https://www.kaggle.com/c/bike-sharing-demand/*. Zawiera on dane zebrane na przestrzeni dwóch lat dotyczące ilości wypożyczeń w systemie rowerów miejskich. Każda próbka to opis każdej godziny pracy systemu w ciągu dnia. Atrybuty zbioru to kolejno:

- data *datetime*

- pora roku *season*
- czy dany dzień jest świętem *holiday*
- czy dany dzień jest dniem roboczym *workingday*
- pogoda - jedna z czterech wartości *weather*
- temperatura *temp*
- temperatura odczuwalna *atemp*
- wilgotność *humidity*
- prędkość wiatru *windspeed*
- liczba rozpoczetych wypożyczeń niezarejestrowanych użytkowników *casual*
- liczba rozpoczętych wypożyczeń przez zarejestrowanych użytkowników *registered*
- całkowita liczba wypożyczeń - atrybut ciągły podlegający pomiarowi błędu *count*

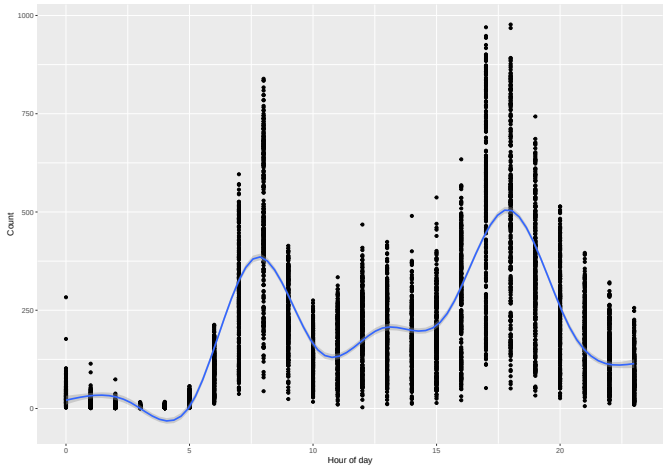
W każdym miesiącu pierwszych 19 dni stanowią dane uczące zbioru. Celem przeprowadzanej regresji jest przewidzenie liczby wypożyczonych rowerów w każdej godzinie od dnia 20. każdego miesiąca, który stanowi zbiór danych testowych.

3 Wstępne przetwarzanie danych

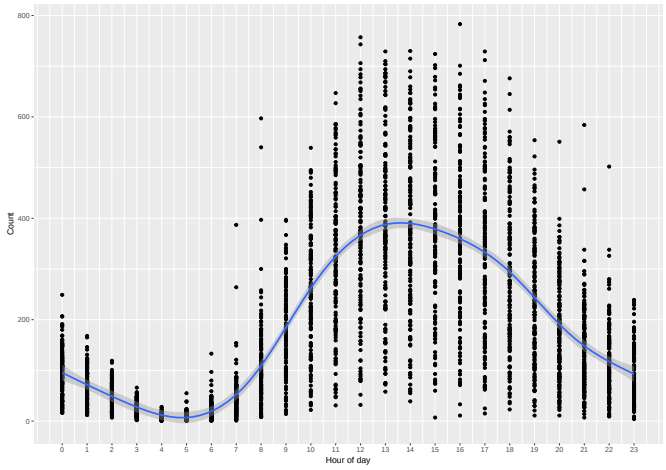
Wykorzystane dane pochodzą z platformy Kaggle i są w formie, która nie wymagała uzupełnienia, ponieważ są one kompletne.

W trakcie wstępnego przetwarzania danych dodawane są nowe atrybuty (plik Preprocess.R):

- Dzień tygodnia (funkcja addDayAttr(), atrybut "weekdays")
- Godzina (funkcja addHourAttr(), atrybut "hours")
- Czy w drodze do/z pracy (godzina 7-9 oraz 15-18 w dniu roboczym) (funkcja addWorkWayAttribute(), atrybut ónwaytowork")



Rysunek 1: Wypożyczenia w dni robocze



Rysunek 2: Wypożyczenia poza dniami roboczymi

Motywacją dla wprowadzenia nowych atrybutów był fakt, iż charakter liczby wypożyczeń w dni robocze oraz w weekend jest odmienny i silnie zależny od godziny dnia.

Usuwane są również ze zbioru trenującego atrybuty dotyczące liczby wypożyczonych rowerów przez zarejestrowanych i niezarejestrowanych użytkowników, ponieważ dane te nie są dostępne w zbiorze testowym.

Dla wykorzystywanego do uczenia zbioru zastosowano wstępną obróbkę, która ograniczała się do usunięcia odstających danych. Ze zbioru uczącego usuwane były przykłady, dla których kwadrat odległości od średniej przekraczał trzykrotność odchylenia standardowego. Zbiór w ten sposób uległ

redukcji o około 1%.

4 Algorytmy selekcji atrybutów

Przed przystąpieniem do selekcji atrybutów przeprowadzona została analiza macierzy kowariancji wszystkich atrybutów. Wykazała ona spodziewaną, wysoką zależność pomiędzy atrybutami związanymi z pogodą, w szczególności atrybutami "temp"i ątemp"(temperatura faktyczna i odczuwalna), co sugeruje wykorzystanie do celów regresji wyłącznie jednego z nich.

4.1 Stepwise Regression

Do selekcji atrybutów wykorzystano regresję krokową. Użyto metody `stepAIC` z pakietu `MASS`. Wykorzystano dwukierunkowy wariant algorytmu (*direction = "both"*). Dla początkowego modelu zawierającego wszystkie atrybuty w postaci:

count onwaytowork+hours+weekdays+windspeed+humidity+atemp+ temp + weather + workingday + holiday + season

otrzymano model końcowy w postaci:

count onwaytowork+hours+weekdays+windspeed+humidity+atemp+ workingday + season

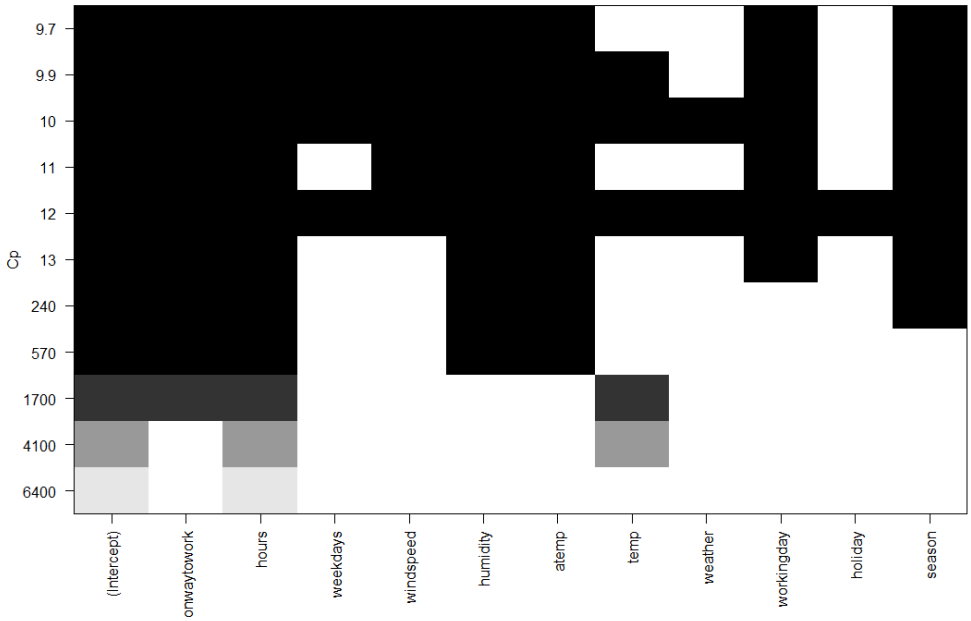
W kolejnych krokach usuwane są następująco atrybuty: *holiday*, *weather* praz *temp*.

4.2 Regsubset analisys

Algorytm selekcji atrybutów korzystający z wyszukiwania wyczerpującego. Wykorzystano funkcję `stepAIC` z pakietu `leaps`. Wynikiem jej działania są najlepsze modele dla odpowiedniej liczby atrybutów. Wyniki działania przedstawione są na rysunku 3. Wynika z niego, że model z sześcioma atrybutami jest najkorzystniejszy. Modele z większą liczbą atrybutów zostały ocenione nieznacznie lepiej.

4.3 Algorytm RFE

Algorytm RFE (*Recursive Feature Elimination*) wykorzystuje zewnętrzny algorytm (w tym przypadku regresję liniową) do oceny błędu walidacji skrośnej. po wykonaniu walidacji bada wagi przypisane poszczególnym atrybutom a następnie przycina atrybuty o najniższych wagach. W każdym kroku algorytmu wykorzystywany jest coraz mniejszy podzbiór atrybutów aż do

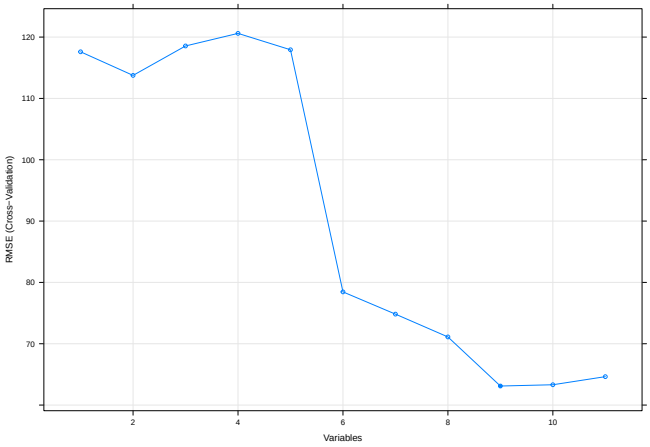


Rysunek 3: Wyniki regsubset analisys

momentu, gdy zostanie osiągana żądana liczba atrybutów. Wykorzystanie algorytmu `rfe` z pakietu `caret` umożliwiło przeprowadzenie rankingu istniejących w zbiorze atrybutów:

1. hours
2. humidity
3. onwaytowork
4. weather
5. weekdays
6. season
7. workingday
8. temp
9. atemp

Algorytm pozwala również przyjrzeć się zależności błędu od liczby wykorzystywanych atrybutów:



Rysunek 4: Błąd w zależności od liczby atrybutów

Wyniki działania algorytmu pokazują, że optymalną liczbą wykorzystywanych atrybutów jest 6-8, przy większej ich liczbie spadek błędu wskutek dodania nowego atrybutu jest marginalny.

5 Metody oceny algorytmów

5.1 Miary błędów

Dla każdego rodzaju walidacji z wyjątkiem walidacji na zbiorze testowym (przeprowadzanej za pomocą aplikacji Kaggle) obliczano dwie miary błędu:

1. Pierwiastek z błędu średniokwadratowego logarytmicznego

$$RMSLE = \sqrt{\frac{1}{n}\sum_{n}^{i=1}(log(p_i + 1) - log(a_i + 1))^2}$$

2. Pierwiastek z błędu średniokwadratowego

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2}$$

Aplikacja Kaggle oceniała wyłącznie błąd $RMSLE$ na zbiorze testowym.

5.2 Walidacja skrośna

Algorytmy badano wykorzystując algorytm k-krotnej walidacji skrośnej. Polega ona na podziale zbioru na K podzbiorów, a następnie wykonywaniu na nich walidacji, przy czym w każdym kroku jeden z K zbiorów pełni rolę zbioru testowego, zaś pozostałe pełnią role zbioru uczącego. W przypadku $K = N$, a zatem gdy zbiór uczący jest równy w każdym kroku całemu dostępnemu zbiorowi treningowemu z wyjątkiem jednej obserwacji, walidacja skrośna k-krotna staje się walidacją *leave-one-out*. Umożliwia ona lepsze uniezależnienie wyników eksperymentu od zmiennych losowych. Dostępny zbiór treningowy okazał się jednak na tyle duży, iż zespół nie wykrył istotnych różnic w błędzie otrzymywanym podczas walidacji metodą leave-one-out oraz zwykłej walidacji skrośnej. Dla każdego algorytmu badano średni błąd przy walidacji dla k=5.

5.3 Walidacja na zbiorze testowym

Na stronie Kaggle wraz z treścią zadania i zbiorem trenującym udostępniony również został zbiór testowy, pozbawiony liczby wypożyczeń. Zbiór ten zawierał łącznie 6494 wpisy, które stanowią zapis liczby wypożyczeń co godzinę w dniach 20-31 każdego miesiąca w okresie dwóch lat (w odróżnieniu od zbioru trenującego, który zawierał wpisy z dni 1-19 każdego miesiąca). Po wytrenowaniu modeli na całym zbiorze trenującym dokonywano predykcji na zbiorze testowym, a następnie ładowano go do aplikacji Kaggle. Odnotowywany został błąd $RMSLE$ zwrócony przez aplikację. Badano w ten sposób zarówno średnią z predykcji wszystkich algorytmów, jak i predykcję poszczególnych algorytmów.

6 Przetestowane algorytmy regresji

6.1 Regresja liniowa

Dla zbadania regresji liniowej wykorzystano funkcję `lm` z pakietu stats. Przy wykorzystaniu pełnego zbioru atrybutów model liniowy dał następujące błędy w walidacji skośnej dla k = 5: RMSLE 1.151384 RMLE 119.815 Wykorzystane w modelu współczynniki dla poszczególnych atrybutów widoczne są w tabeli 1.

Dla modelu zwróconego przez algorytm *stepwise regression* błąd wynosił odpowiednio: RMSLE 1.151384 RMLE 119.815

Tabela 2 zawiera błędy dla wybranych modeli zwróconych przez algorytm *Regsubset analisys*. Wyniki pokrywają się z klasyfikacją modeli podaną jako

(Intercept)	onwaytowork	hours	weekdays
34.7226629	233.2879647	8.3635308	1.2258257
windspeed	humidity	atemp	temp
0.3128109	-2.3272347	5.0743702	1.2704038
weather	workingday	holiday	season
-2.7714344	-34.6009805	-1.8851859	20.1173451

Tablica 1: Współczynniki w modelu liniowym

wynik jego działania. Dla modeli składających się z 6 i więcej atrybutów błąd maleje minimalnie. Dla modelu składającego się z 8 atrybutów otrzymujemy już taki sam błąd jak dla pełnego zestawu atrybutów.

Liczba atrybutów	atrybuty	RMSLE	RMSE
1	hours	1.328705	152.3693
2	hours + temp	1.236916	141.8473
3	onwaytowork + hours + temp	1.18398	129.6105
4	onwaytowork + hours + humidity + atemp	1.172317	123.351
5	onwaytowork + hours + humidity + atemp + season	1.171745	121.2109
7	onwaytowork + hours + windspeed + humidity + atemp + workingday + season	1.152467	119.855
8	onwaytowork + hours + weekdays + windspeed + humidity + atemp + workingday + season	1.15163	119.8403

Tablica 2: Błąd funkcji lm dla poszczególnych podzbiorów atrybutów zwróconych przez regsubset

6.2 Regresja lokalna

Jako metodę regresji lokalnej wykorzystano funkcję `loess` z pakietu `stats`. Przyjmuje ona do czterech atrybutów. Funkcja przyjmuje dwa parametry, które dotyczą wygładzania:

- Degree - stopień wielomianów stosowanych w wygładzaniu wielomianów. Przyjmuje watości 0, 1 oraz 2.
- Span - parametr α odpowiadający za stopień wygładzania

W tabeli 3 przedstawiono wyniki badań jakości przeprowadzonej regresji lokalnej. Błędy zostały policzone jako średnia błędów z walidacji skośnej dla k = 5. Choć mniejszy błąd występował dla wielomianów drugiego stopnia, to najlepszy wynik osiągnięto dla *Degree* = 1 oraz *span* = 0.01. Im mniejsza była wartość parametu *span*, tym mniejszy był błąd walidacji skośnej.

Degree	Span	RMSLE	RMSE
2	0.01	0.78	99.7
1	0.01	0.75	99.2
0	0.01	0.83	104
2	0.1	0.932	112
1	0.1	1	114
0	0.1	1.12	121
2	0.3	1.02	114
1	0.3	1.22	121
0	0.3	1.25	131
2	0.5	1.12	117
1	0.5	1.23	124
0	0.5	1.32	136
2	0.75	1.27	119
1	0.75	1.17	128
0	0.75	1.38	142
2	1	1.4	120
1	1	1.14	133
0	1	1.5	157

Tablica 3: Błąd funkcji loess w zależności od przyjętych parametrów dla formuły count ~ humidity + hours + atemp

Wpływ doboru atrybutów na wyniki regresji lokalnej przedstawia tabela 4. Najlepszy wynik osiągnięto dla dwóch atrybutów: *hours* i *temp*.

6.3 Robust regression z M-estymacją

Regresja typu "robust" z M- lub MM-estymacją wykazuje zwiększoną odporność na zanieczyszczone dane trenujące, zawierające dużą liczbę wartości odstających. W przypadku dostępnego zbioru treningowego nie było to dużym problemem, ponieważ w procesie preprocessingu usunięte zostały takie wartości, zaś ich ogólna liczba była niewielka.

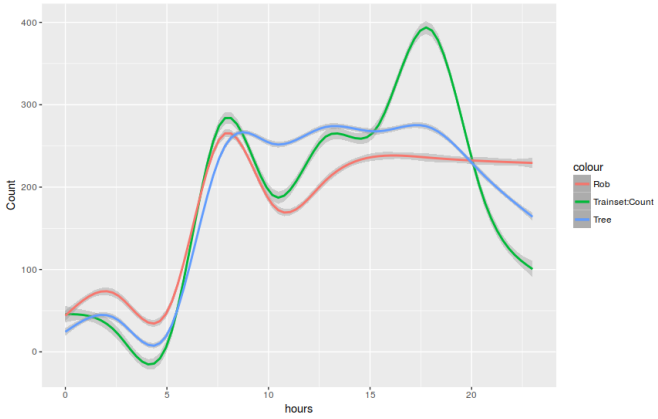
Regresja z m-estymacją powoduje wygładzenie skokowych zmian wartości liczby wypożyczeń, co szczególnie negatywnie odbija się dla danych po-

atrybuty	RMSLE	RMSE
hours	4.774647	247.5403
hours + temp	0.7036961	100.3796
hours + humidity + atemp + season	0.8488961	199.1475

Tablica 4: Wyniki funkcji `loess` w zależności od dobranych atrybutów dla parametrów *span* = 0.01 *degree* = 1

Sposób estymacji	RMSLE	RMSE
M-estymacja	1.129622	120.9462
MM-estymacja	1.110996	122.7891

Tablica 5: Wyniki w zależności od metody estymacji (średni błąd walidacji skrośnej dla k=5)



Rysunek 5: Wyniki regresji w trakcie weekendu w porównaniu z innymi algorytmami

chodzących z dni roboczych (szczyt poranny i popołudniowy). Wobec tego, przeprowadzono porównanie błędu obliczonego z wykorzystaniem tej metody dla danych weekendowych oraz dla danych z dni roboczych oraz porównano je z inną metodą wyznaczania regresji

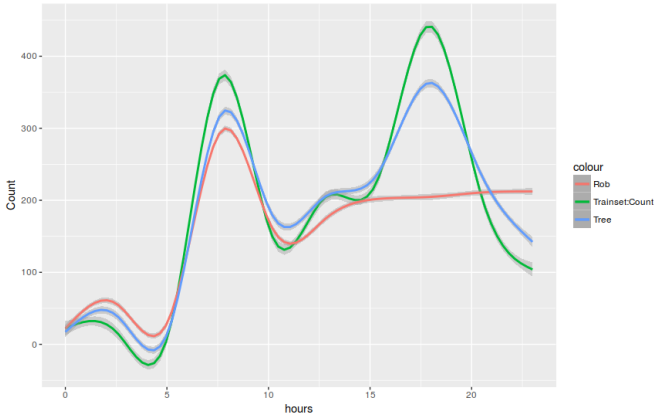
Funkcja psi	RMSLE	RMSE
psi.bisquare	1.116625	122.184
psi.huber	1.129622	120.9462
psi.hampel	1.14003	120.229

Tablica 6: Wyniki w zależności od funkcji psi

	Dni robocze	Weekendy
Robust RMSE	120.8319	123.9919
Robust RMSLE	1.034289	1.258234
Linear RMSLE	1.097582	1.265375

Tablica 7: Porównanie wyników osiąganych przez algorytm w dni robocze oraz weekendy

Podejście okazało się jednak błędne, ponieważ dla weekendów istniało zbyt mało danych, aby skoki wypożyczeń w godzinach popołudniowych nie uległy wygładzeniu przez algorytm.



Rysunek 6: Wykres ukazujący wyniki algorytmu na zbiorze ograniczonym do dni roboczych w porównaniu z innymi algorytmami

Po analizie wyjścia algorytmów selekcji atrybutów przeprowadzone zostało badanie algorytmu dla zbioru ograniczonego do 8 atrybutów - "hourshumidityonwaytoworkweatherweekdaysseasonworkingdaytemp", po czym zbiór ten zmniejszano w kolejności od najmniej istotnego atrybutu wg. Algorytmu RFE

W przypadku regresji robust ograniczenie ilości atrybutów prowadzi do spadku jakości regresji ze względu na niemożliwość dopasowania hiperpłaszczyzny regresji.

6.4 Drzewa regresji

Jako drzewo decyzyjne wykorzystano implementację z pakietu `rpart`. Na otrzymanym drzewie przeprowadzono przycinanie metodą `prune`. W tabeli 9 przedstawiono wpływ współczynnika przycinania na otrzymane wyniki.

Liczba atrybutów	RMSLE	RMSE
8	1.114278	122.9576
7	1.125025	140.3991
6	1.132462	137.4054
5	1.127936	138.382
4	1.137699	138.9563
3	1.129964	142.1899
2	1.190753	150.9452
1	1.22215	155.4499

Tablica 8: Błędy robust regression w zależności od liczby atrybutów.

Cp	Meth	RMSLE	RMSE
0.005	anova	0.8742775	98.55746
0.01	anova	0.8742775	98.55746
0.01	poisson	0.7644792	100.8667
0.04	anova	0.9474914	124.1835
0.04	poisson	0.941012	123.5936
1	anova	0.9910873	135.782

Tablica 9: Błędy drzewa decyzyjnego

Eksperyment przeprowadzono dla wykorzystania w drzewie analizy wariancji (*anova*) oraz rozkładu Poissona (*poissson*). Korzystniejsze wyniki były otrzymywane przy wykorzystaniu rozkładu Poissona.

W tabeli 10 przedstawiono wpływ liczby atrybutów na wynik regresji.

6.5 Sieć neuronowa

Jako kolejną z metod regresji wykorzystano sieć neuronową. Aby mogła ona zostać wykorzystana w tym celu, neuron w warstwie wyjściowej musi wykorzystywać liniową funkcję aktywacji. W celu przeprowadzenia regresji wykorzystano pakiet `nnet`, który umożliwia zbudowanie sieci neuronowej z pojedynczą warstwą ukrytą. W przypadku sieci neuronowej kluczowy był również wynik badania z wykorzystaniem zbioru testowego, ponieważ w przypadku walidacji skrośnej zachodziło niebezpieczeństwo zbytniego dopasowania do zbioru trenującego.

Poniższe wyniki regresji na zbiorze trenującym dowodzą, iż algorytm sieci neuronowej wykazuje dużą przydatność do celów regresji.

Liczba atrybutów	RMSLE	RMSE
8	0.9396421	121.915
7	0.9301871	120.8053
6	0.9301871	120.8053
5	0.9596607	128.2725
4	0.9596607	128.2725
3	0.9596607	128.2725
2	0.9596607	128.2725
1	0.9596607	128.2725

Tablica 10: Błędy drzewa regresji w zależności od liczby atrybutów dla poarametru przycinania $cp = 0.04$ oraz przy wykorzystaniu analizy wariancji.

Liczba neuronów	RMSLE	RMSE
10	0.7643461	76.37187
20	0.6713900	76.48027
30	0.5563560	63.62175
50	0.5327608	61.13144
100	0.4799462	57.26756
300	0.4683365	59.87699

Tablica 11: Wyniki Neural network regression

Liczba neuronów	Wynik na zbiorze testowym
50	0.55548
100	0.57295
150	0.53851

Tablica 12: predykcje kaggle

7 Wnioski

W ramach przeprowadzonego projektu wykorzystano szereg popularnych algorytmów regresji. Zadanie oraz wykorzystany zbiór danych pozwoliły na zapoznanie się z właściwościami algorytmów, przeprowadzenie ich analizy oraz krytyki pod kątem istniejącego zbioru danych. Korzystając z wiedzy pozyskanej na podstawie wstępnej analizy zbioru wyodrębniono nowe atrybuty na podstawie istniejących danych. Były one związane z godziną obserwacji, dniem tygodnia oraz określeniem, czy data i godzina obserwacji wskazują na godziny szczytu dni roboczych. Pomimo stosunkowo niedużej ilości atrybutów, przed przystąpieniem do predykcji zapotrzebowania na rowery przystąpiono do analizy statystycznej atrybutów, wykorzystując zarówno proste metody statystyczne takie jak badanie macierzy kowariancji pod kątem wzajemnej zależności atrybutów, jak również algorytmy służące do określenia ich istotności takie jak algorytm stepwise regression, regsubset analysis oraz algorytm RFE. Wszystkie wymienione algorytmy zwróciły stosunkowo podobne wyniki poświadczając przydatność dla celów regresji wprowadzonych przez nas atrybutów. Kolejnym atutem wykorzystania algorytmów selekcji jest fakt, iż pozwoliły one określić pożądaną wielkość zbioru atrybutów, powyżej której przyrost ilości atrybutów nie skutkuje poprawą przydatności predykcyjnej modeli regresyjnych. Wyniki analiz zostały następnie potwierdzone w praktyce, czego świadectwem są zamieszczone w sprawozdaniu tabele.

W toku dalszych prac opracowane zostały metody badania i analizy algorytmów regresji. Zespół przygotował metody umożliwiające wykonanie badania algorytmów z wykorzystaniem walidacji skrośnej a także zwrócenia wyników działania algorytmu na zbiorze testowym celem załadowania ich do aplikacji Kaggle. Przebadano pięć dostępnych w języku R algorytmów regresji: regresję liniową, regresję lokalną, robust regression, drzewa regresji oraz regresję wykorzystującą sieć neuronową. W celu potencjalnego zmniejszenia błędu badano zarówno błąd każdego algorytmu jak i błąd średniej predykcji wszystkich algorytmów - podejście to nie przyniosło jednak rezultatów w postaci poprawienia predykcji. Po opracowaniu procedu badawczych przystąpiono do dostrajania parametrów algorytmów przy wykorzystaniu walidacji skrośnej. Dla wszystkich algorytmów sterowanie ich parametrami oraz ograniczanie zbioru atrybutów, na których działają, przyniosło poprawę wyników uzyskiwanych predykcji. Stosunkowo najlepsze wyniki dało badanie algorytmu regresji lokalnej, przy której błąd w zależności od ustawień parametrów algorytmu udało się zmniejszyć niemal dwukrotnie.

Badania pozwoliły na ocenę przydatności algorytmów do celów predykcyjnych na istniejącym zbiorze. Stosunkowo najgorsze wyniki otrzymano przy

wykorzoystaniu regresji typu robust ("odpornej"), co może być związane z gwałtownym charakterem zmian liczby wypożyczeń w zależności od godziny dnia - wygładzanie predykcji metodą m-estymacji powoduje wówczas jedynie zwiększenie błędu. Algorytm ten wydaje się być również nieodpowiedni dla przedstawionego problemu ze względu na statystykę danych treningowych - nieomal pozbawionych wartości odstających, przy których algorytm ten miałby lepsze zastosowanie.

W wyniku walidacji skrośnej najlepsze wyniki osiągnięto przy zastosowaniu sieci neuronowej. Za tym podejściem kryła się jednak możliwość nadmiernego dopasowania do zbioru trenującego, a co za tym idzie - słabych wyników tego modelu na zbiorze trenującym, w przypadku gdyby ten różnił się istotnie od zbioru uczącego. Obawy te rozwiano badaniem algorytmu na zbiorze trenującym - uzyskał on najniższy błąd spośród wszystkich badanych algorytmów.

Zespół uważa, iż możliwa jest dalsza poprawa uzyskiwanych wyników. W tym celu należałoby przetestować kolejne metody regresji, takie jak ridge regression oraz regresja z wykorzystaniem metod głębokiego uczenia - wielowarstwowych sieci neuronowych.