

Raport – Projekt z analizy danych

Adam Wróbel, Julia Wronowska, Mateusz Wrzosek

Zadanie 1. (Grupowanie!.xlsx)

Dla danych z poszczególnych arkuszy przeprowadzić grupowanie 3 różnymi technikami dla różnych odsetków elementów przynależących do zbioru uczącego (70, 80, 90 %). Porównać efektywność wybranych metod. Wskazać zmienną (zmienne) najbardziej istotne przy podziale.

Klasteryzacja jest jednym z kluczowych zagadnień w analizie danych i uczeniu maszynowym, odgrywając istotną rolę w odkrywaniu struktury danych bez nadzoru. Metody te umożliwiają grupowanie obiektów w zbiory o podobnych cechach, co wspomaga eksplorację danych oraz ułatwia podejmowanie decyzji w różnych dziedzinach.

Celem tego raportu jest przedstawienie klasteryzacji poprzez analizę wybranych algorytmów przy ich wykorzystaniu na dostarczonej bazie danych. W szczególności skupiono się na metodach takich jak:

1. c-Means (Fuzzy c-Means Clustering)

c-Means to metoda klasteryzacji, która pozwala na rozmyte przypisanie punktów do klastrów. Działa w sposób iteracyjny, szukając optymalnego rozmieszczenia klastrów i stopni przynależności punktów.

- **Rozmytość:** Każdy punkt może należeć do więcej niż jednego klastra z określonym prawdopodobieństwem (stopniem przynależności).
- **Iteracyjny proces optymalizacji:** Algorytm rozpoczyna od losowego rozmieszczenia klastrów, a następnie iteracyjnie minimalizuje funkcję celu, która uwzględnia odległości punktów od klastrów i ich stopnie przynależności.
- **Parametr sterujący m :** Kontroluje poziom rozmytości. Dla $m > 1$ przypisania stają się bardziej rozmyte, co pozwala na elastyczniejsze dopasowanie do danych.
- **Odporność na szum:** Dzięki przypisaniu punktów do klastrów w sposób rozmyty, metoda jest bardziej odporna na szum w danych w porównaniu do klasycznego k-Means.
- **Wymagania danych:** Wymaga zdefiniowania liczby klastrów przed rozpoczęciem działania oraz odpowiedniego znormalizowania danych.

2. SOTA Learner (Self-Organizing Tree Algorithm)

SOTA jest metodą hierarchicznej klasteryzacji, która łączy cechy algorytmów samoorganizujących się (Self-Organizing Maps, SOM) z drzewiastą strukturą.

- **Hierarchiczna natura:** Klastery są reprezentowane w postaci drzewa, gdzie każda gałąź i węzeł odpowiadają różnym poziomom grupowania danych.
- **Iteracyjna aktualizacja:** Działa w sposób dynamiczny, gdzie nowe punkty mogą powodować rozgałęzienia drzewa, co umożliwia stopniowe dostosowanie struktury do danych.
- **Uczenie przez samoorganizację:** Wykorzystuje podobieństwo do sieci neuronowych Kohonena, gdzie wagi odpowiadające klastrom są aktualizowane w procesie uczenia.
- **Zastosowanie dużych zbiorów danych:** Szczególnie przydatny w analizie danych o dużej liczbie wymiarów, gdzie standardowe metody mogą być mniej efektywne.
- **Wyważanie precyzji i ogólności:** SOTA dąży do utworzenia struktury drzewa, która jest wystarczająco szczegółowa, ale nie nadmiernie skomplikowana, co pozwala na łatwiejszą interpretację.

3. Drzewo decyzyjne (Decision Tree)

Drzewa decyzyjne to algorytmy klasyfikacyjne, które mogą również służyć do grupowania (pośrednio klasteryzacji) danych poprzez ich segmentację w oparciu o reguły.

- **Struktura drzewa:** Składa się z węzłów decyzyjnych, które odpowiadają atrybutom danych, oraz liści, które reprezentują końcowe grupy (klasy).
- **Reguły decyzyjne:** W każdym węźle podejmowana jest decyzja na podstawie progu wartości atrybutu, maksymalizując różnorodność między grupami.
- **Algorytmy budowy drzewa:** CART (Classification and Regression Trees) i C4.5 to jedne z najpopularniejszych.
- **Działanie iteracyjne:** Drzewo jest budowane od korzenia, a każdy podział optymalizuje funkcję celu (np. entropię, indeks Gini).
- **Przycinanie drzewa:** Aby uniknąć przeuczenia, stosuje się metody przycinania, które redukują liczbę gałęzi drzewa.
- **Łatwość interpretacji:** Wyniki analizy w postaci drzewa są intuicyjne i łatwe do zrozumienia, co jest dużą zaletą w wielu zastosowaniach, np. medycynie czy analizie finansowej.

Przedstawienie wyników

Arkusz1

Stosunek procentowy zbioru treningowego do testowego	Technika klasteryzacji	Dopasowania	Iteracja losowania zbioru										Końcowy % poprawnych dopasowań
			1	2	3	4	5	6	7	8	9	10	
70/30	c-Means	dobrze	37	24	38	27	12	16	52	30	59	36	21,6%
		źle	116	129	115	126	141	137	101	123	94	117	
	SOTA Learner	dobrze	111	115	105	114	114	121	111	116	114	112	74,1%
		źle	42	38	48	39	39	32	42	37	39	41	
	Drzewo decyzyjne	dobrze	153	153	153	153	151	153	153	153	153	152	99,8%
		źle	0	0	0	0	2	0	0	0	0	1	
80/20	c-Means	dobrze	10	25	36	18	11	16	16	29	31	27	21,5%
		źle	92	77	66	84	91	86	86	73	71	75	
	SOTA Learner	dobrze	64	76	78	77	81	76	81	79	75	78	75,0%
		źle	38	26	24	25	21	26	21	23	27	24	
	Drzewo decyzyjne	dobrze	102	102	102	102	102	101	102	100	102	99	99,4%
		źle	0	0	0	0	0	1	0	2	0	3	
90/10	c-Means	dobrze	9	6	13	11	12	12	12	18	12	11	22,3%
		źle	43	46	39	41	40	40	40	34	40	41	
	SOTA Learner	dobrze	39	44	36	38	40	42	38	43	44	36	76,9%
		źle	13	8	16	14	12	10	14	9	8	16	
	Drzewo decyzyjne	dobrze	52	52	52	52	52	52	52	52	52	52	100,0%
		źle	0	0	0	0	0	0	0	0	0	0	

Drzewo decyzyjne uzyskało najlepsze wyniki spośród wszystkich testowanych metod. Wyniki poprawnych dopasowań utrzymują się na poziomie bliskim lub równym 100% dla wszystkich proporcji zbioru treningowego do testowego. Algorytm był wysoce stabilny, osiągając niemal identyczne wyniki w każdej iteracji losowania zbioru. Świadczy to o niewielkiej podatności na zmienność w danych treningowych i testowych. Dla proporcji 70/30 osiągnięto 99,8% poprawnych dopasowań przy maksymalnie 2 błędnych klasyfikacjach w iteracji, dla 80/20 wyniki wyniosły 99,4% z maksymalnie 3 błędami, natomiast dla 90/10 algorytm osiągnął perfekcyjną skuteczność 100%. Drzewo decyzyjne jest więc najlepszym wyborem w tym przypadku, idealnie nadającym się do zadania klasteryzacji dla tego zbioru danych.

SOTA Learner osiągnął przyzwoite wyniki, z poprawnymi dopasowaniami wynoszącymi od 74% do 77% w zależności od proporcji danych. Algorytm wykazywał umiarkowaną stabilność, choć wyniki zmieniały się w poszczególnych iteracjach w większym zakresie niż w przypadku drzewa decyzyjnego. Dla proporcji 70/30 skuteczność wyniosła 74,1%, przy średnio 40 błędach na iterację, dla 80/20 poprawne dopasowania wyniosły 75% przy średnio 24 błędach, a dla 90/10 76,9%, z ok. 10 błędami na iterację. Wydajność poprawiała się wraz ze wzrostem danych treningowych, co sugeruje, że algorytm dobrze skaluje się przy większych zbiorach treningowych. SOTA Learner może być użyteczną metodą dla tego problemu, ale nie osiąga wyników zbliżonych do drzewa decyzyjnego.

Algorytm c-Means uzyskał najgorsze wyniki spośród wszystkich testowanych metod, z poprawnymi dopasowaniami wynoszącymi jedynie około 21-22%. Algorytm był stabilny, ale jego wyniki były systematycznie niskie, niezależnie od proporcji danych treningowych do testowych. Dla proporcji 70/30 poprawne dopasowania wyniosły 21,6% przy ponad 100 błędach na iterację, dla 80/20 wyniki te utrzymały się na poziomie 21,5%, a dla 90/10 22,3%, z około 40 błędami na iterację. Algorytm nie był w stanie skutecznie odzwierciedlić struktury danych, co sugeruje, że klasteryzacja k-średnich (c-Means) nie jest odpowiednia dla tego typu problemu. c-Means nie jest więc zalecanym algorytmem dla tego zbioru danych ze względu na bardzo niską skuteczność.

Podsumowanie:

- **Drzewo decyzyjne** jest zdecydowanym liderem pod względem skuteczności, stabilności i niezawodności, niezależnie od proporcji danych treningowych do testowych.
- **SOTA Learner** daje solidne rezultaty, ale wymaga większych zbiorów treningowych, aby osiągnąć zadowalającą skuteczność.
- **c-Means** jest metodą najmniej odpowiednią dla tego problemu, co może wynikać z niewłaściwego dopasowania modelu do struktury danych.

Najbardziej istotną zmienną była x_2 , ponieważ liście drzewa decyzyjnego opierają się głównie na zmiennej x_2 . Oznacza to, że ta zmienna miała kluczowe znaczenie przy podejmowaniu decyzji w algorytmie drzewa decyzyjnego, co w dużej mierze przyczyniło się do wysokiej skuteczności tego modelu. Drzewo decyzyjne jest w stanie uchwycić złożone zależności w danych, a zmienne, takie jak x_2 , mają decydujący wpływ na sposób, w jaki model dzieli przestrzeń cechową, prowadząc do trafnych klasyfikacji.

Arkusz2

Stosunek procentowy zbioru treningowego do testowego	Technika klasteryzacji	Dopasowania	Iteracja losowania zbioru										Końcowy % poprawnych
			1	2	3	4	5	6	7	8	9	10	
70/30	c-Means	dobrze	40	54	44	26	25	47	33	18	37	26	23,0%
		źle	112	98	108	126	127	105	119	134	115	126	
	SOTA Learner	dobrze	132	121	116	121	118	122	118	120	119	120	79,4%
		źle	20	31	36	31	34	30	34	32	33	32	
	Drzewo decyzyjne	dobrze	134	134	127	136	129	129	136	135	136	136	87,6%
		źle	18	18	25	16	23	23	16	17	16	16	
80/20	c-Means	dobrze	31	18	18	16	35	34	21	12	11	16	21,0%
		źle	70	83	83	85	66	67	80	89	90	85	
	SOTA Learner	dobrze	78	80	80	78	85	80	83	84	86	87	81,3%
		źle	23	21	21	23	16	21	18	17	15	14	
	Drzewo decyzyjne	dobrze	93	91	94	94	90	92	90	93	88	93	90,9%
		źle	8	10	7	7	11	9	11	8	13	8	
90/10	c-Means	dobrze	11	22	13	10	19	17	4	7	8	14	24,0%
		źle	41	30	39	42	33	35	48	45	44	38	
	SOTA Learner	dobrze	44	37	44	41	45	45	41	40	43	43	81,3%
		źle	8	15	8	11	7	7	11	12	9	9	
	Drzewo decyzyjne	dobrze	47	45	45	45	46	48	43	48	44	48	88,3%
		źle	5	7	7	7	6	4	9	4	8	4	

Drzewo decyzyjne uzyskało najlepsze wyniki spośród wszystkich testowanych metod. Wyniki poprawnych dopasowań utrzymują się na poziomie bliskim lub równym 100% dla wszystkich proporcji zbioru treningowego do testowego. Algorytm był wysoce stabilny, osiągając niemal identyczne wyniki w każdej iteracji losowania zbioru. Świadczy to o niewielkiej podatności na zmienność w danych treningowych i testowych. Dla proporcji 70/30 osiągnięto 87,6% poprawnych dopasowań przy maksymalnie 18 błędnych klasyfikacjach w iteracji, dla 80/20 wyniki wyniosły 90,9% z maksymalnie 10 błędami, natomiast dla 90/10 algorytm osiągnął 88,3% skuteczności przy maksymalnie 7 błędnych klasyfikacjach. Drzewo decyzyjne jest więc najlepszym wyborem w tym przypadku, idealnie nadającym się do zadania klasteryzacji dla tego zbioru danych.

SOTA Learner osiągnął przyzwoite wyniki, z poprawnymi dopasowaniami wynoszącymi od 74% do 77% w zależności od proporcji danych. Algorytm wykazywał umiarkowaną stabilność, choć wyniki zmieniały się w poszczególnych iteracjach w większym zakresie niż w przypadku drzewa decyzyjnego. Dla proporcji 70/30 skuteczność wyniosła 79,4%, przy średnio 32 błędach na iterację, dla 80/20 poprawne dopasowania wyniosły 81,3% przy średnio 14 błędach, a dla 90/10 81,3%, z ok. 9

błędami na iterację. Wydajność poprawiała się wraz ze wzrostem danych treningowych, co sugeruje, że algorytm dobrze skaluje się przy większych zbiorach treningowych. SOTA Learner może być użyteczną metodą dla tego problemu, ale nie osiąga wyników zbliżonych do drzewa decyzyjnego.

Algorytm c-Means uzyskał najgorsze wyniki spośród wszystkich testowanych metod, z poprawnymi dopasowaniami wynoszącymi jedynie około 21-24%. Algorytm był stabilny, ale jego wyniki były systematycznie niskie, niezależnie od proporcji danych treningowych do testowych. Dla proporcji 70/30 poprawne dopasowania wyniosły 23,0% przy ponad 100 błędach na iterację, dla 80/20 wyniki te utrzymały się na poziomie 21,0%, a dla 90/10 24,0%, z około 38 błędami na iterację. Algorytm nie był w stanie skutecznie odzwierciedlić struktury danych, co sugeruje, że klasteryzacja k-średnich (c-Means) nie jest odpowiednia dla tego typu problemu. c-Means nie jest więc zalecanym algorytmem dla tego zbioru danych ze względu na bardzo niską skuteczność.

Podsumowanie:

- Drzewo decyzyjne jest zdecydowanym liderem pod względem skuteczności, stabilności i niezawodności, niezależnie od proporcji danych treningowych do testowych.
- SOTA Learner daje solidne rezultaty, ale wymaga większych zbiorów treningowych, aby osiągnąć zadowalającą skuteczność.
- c-Means jest metodą najmniej odpowiednią dla tego problemu, co może wynikać z niewłaściwego dopasowania modelu do struktury danych.

Najbardziej istotnymi zmiennymi były **x2** oraz **x3**, ponieważ liście drzewa decyzyjnego opierają się głównie na tych zmiennych. Na ostatnich liściach pojawia się również zmienna **x4**, ale ma ona mniejsze znaczenie w porównaniu do **x2** i **x3**. Oznacza to, że te zmienne miały kluczowe znaczenie przy podejmowaniu decyzji w algorytmie drzewa decyzyjnego, co w dużej mierze przyczyniło się do wysokiej skuteczności tego modelu. Drzewo decyzyjne jest w stanie uchwycić złożone zależności w danych, a zmienne, takie jak **x2** i **x3**, mają decydujący wpływ na sposób, w jaki model dzieli przestrzeń cechową, prowadząc do trafnych klasyfikacji.

Arkusz3

Stosunek procentowy zbioru treningowego do testowego	Technika klasteryzacji	Dopasowania	Iteracja losowania zbioru										Końcowy % poprawnych
			1	2	3	4	5	6	7	8	9	10	
70/30	c-Means	dobrze	43	25	32	46	24	37	22	38	25	22	20,7%
		źle	109	127	120	106	128	115	130	114	127	130	
	SOTA Learner	dobrze	86	96	104	103	111	87	107	106	107	108	66,8%
		źle	66	56	48	49	41	65	45	46	45	44	
	Drzewo decyzyjne	dobrze	147	148	145	149	145	149	149	144	144	150	96,7%
		źle	5	4	7	3	7	3	3	8	8	2	
80/20	c-Means	dobrze	43	23	15	28	25	33	32	37	31	16	27,7%
		źle	59	79	87	74	77	69	70	65	71	86	
	SOTA Learner	dobrze	76	72	69	63	65	67	75	74	61	73	68,1%
		źle	26	30	33	39	37	35	27	28	41	29	
	Drzewo decyzyjne	dobrze	101	100	97	100	100	101	100	98	97	97	97,2%
		źle	1	2	5	2	2	1	2	4	5	5	
90/10	c-Means	dobrze	10	15	11	10	8	15	13	12	15	13	23,5%
		źle	42	37	41	42	44	37	39	40	37	39	
	SOTA Learner	dobrze	34	36	34	33	37	35	34	34	34	40	67,5%
		źle	18	16	18	19	15	17	18	18	18	12	
	Drzewo decyzyjne	dobrze	50	51	50	51	51	51	49	49	51	51	96,9%
		źle	2	1	2	1	1	1	3	3	1	1	

Drzewo decyzyjne uzyskało najlepsze wyniki spośród wszystkich testowanych metod. Wyniki poprawnych dopasowań utrzymują się na poziomie bliskim lub równym 100% dla wszystkich proporcji zbioru treningowego do testowego. Algorytm był wysoce stabilny, osiągając niemal identyczne wyniki w każdej iteracji losowania zbioru. Świadczy to o niewielkiej podatności na zmienność w danych treningowych i testowych. Dla proporcji 70/30 osiągnięto 96,7% poprawnych dopasowań przy maksymalnie 7 błędnych klasyfikacjach w iteracji, dla 80/20 wyniki wyniosły 97,2% z maksymalnie 5 błędami, natomiast dla 90/10 algorytm osiągnął 96,9% skuteczności przy maksymalnie 3 błędnych klasyfikacjach. Drzewo decyzyjne jest więc zdecydowanym liderem w tym przypadku, idealnie nadającym się do zadania klasteryzacji dla tego zbioru danych.

SOTA Learner osiągnął przyzwoite wyniki, z poprawnymi dopasowaniami wynoszącymi od 66,8% do 68,1% w zależności od proporcji danych. Algorytm wykazywał umiarkowaną stabilność, choć wyniki zmieniały się w poszczególnych iteracjach w większym zakresie niż w przypadku drzewa decyzyjnego. Dla proporcji 70/30 skuteczność wyniosła 66,8%, przy średnio 44 błędach na iterację, dla 80/20 poprawne dopasowania wyniosły 68,1% przy średnio 29 błędach, a dla 90/10 67,5%, z ok. 12 błędami na iterację. Wydajność poprawiała się wraz ze wzrostem danych treningowych, co sugeruje, że algorytm dobrze skaluje się przy większych zbiorach treningowych. SOTA Learner może być użyteczną metodą dla tego problemu, ale nie osiąga wyników zbliżonych do drzewa decyzyjnego.

Algorytm c-Means uzyskał najgorsze wyniki spośród wszystkich testowanych metod, z poprawnymi dopasowaniami wynoszącymi jedynie około 20,7% do 27,7%. Algorytm był stabilny, ale jego wyniki były systematycznie niskie, niezależnie od proporcji danych treningowych do testowych. Dla proporcji 70/30 poprawne dopasowania wyniosły 20,7% przy ponad 120 błędach na iterację, dla 80/20 wyniki te utrzymały się na poziomie 27,7%, a dla 90/10 23,5%, z około 39 błędami na iterację. Algorytm nie był w stanie skutecznie odzwierciedlić struktury danych, co sugeruje, że klasteryzacja k-średnich (c-Means) nie jest odpowiednia dla tego typu problemu. c-Means nie jest więc zalecanym algorytmem dla tego zbioru danych ze względu na bardzo niską skuteczność.

Podsumowanie:

- **Drzewo decyzyjne** jest zdecydowanym liderem pod względem skuteczności, stabilności i niezawodności, niezależnie od proporcji danych treningowych do testowych.
- **SOTA Learner** daje solidne rezultaty, ale wymaga większych zbiorów treningowych, aby osiągnąć zadowalającą skuteczność.
- **c-Means** jest metodą najmniej odpowiednią dla tego problemu, co może wynikać z niewłaściwego dopasowania modelu do struktury danych.

Najbardziej istotną zmienną była **x4**, ponieważ liście drzewa decyzyjnego opierają się głównie na tej zmiennej. Na pojedynczych liściach pojawiają się również zmienne **x1**, **x2** i **x3**, ale mają one mniejsze znaczenie w porównaniu do **x4**. Oznacza to, że **x4** miała kluczowe znaczenie przy podejmowaniu decyzji w algorytmie drzewa decyzyjnego, co w dużej mierze przyczyniło się do wysokiej skuteczności tego modelu. Drzewo decyzyjne skutecznie wykorzystuje te zmienne do tworzenia podziałów w przestrzeni cechowej, prowadząc do trafnych klasyfikacji.

Zadanie 2. (prognozowanie1.xlsx)

Zbudować model oparty na prostej regresji i ocenić jego efektywność obliczając podstawowe statystyki dla błędu względnego.

Funkcje, które zostały użyte w excelu:

-Analiza regresji.

-Prognoza-> $= 1583,713783 + 10,00637497 * A2$

-Błąd względny-> $= \text{MODUŁ.LICZBY}(C2 - D2) / C2$

-Średnia-> $= \text{ŚREDNIA}(E2:E731)$

-Max-> $\text{MAX}(E2:E731)$

-Min-> $\text{MIN}(E2:E731)$

1. Podstawowe informacje o modelu

- **Wielokrotność R (R)**: 0.987, co sugeruje bardzo silną korelację między zmienną niezależną (czasem t) a zmienną zależną (obrotem).
- **R kwadrat (R²)**: 0.974, co oznacza, że model wyjaśnia 97,4% zmienności w danych, co wskazuje na wysoką jakość dopasowania.
- **Dopasowany R kwadrat**: 0.974, co potwierdza, że model jest dobrze dopasowany do danych.

- **Błąd standardowy:** 341.73, co wskazuje na średnią wielkość błędu w prognozach modelu.
- **Liczba obserwacji:** 730, co oznacza, że model został przetestowany na 730 próbkach.

<i>Statystyki regresji</i>	
Wielokrotność R	0,987155906
R kwadrat	0,974476783
Dopasowany R kwadrat	0,974441724
Błąd standardowy	341,7325144
Obserwacje	730

2. Statystyki regresji i analiza wariancji

Analiza wariancji (ANOVA):

- **Regresja:**
 - **SS (Suma kwadratów regresji):** 3,245,936,847
 - **MS (Średnia suma kwadratów regresji):** 3,245,936,847
 - **F-statystyka:** 27,795.05, co sugeruje, że model jest statystycznie istotny.
 - **Wartość-p:** 0, co oznacza, że model jest istotny na bardzo wysokim poziomie ufności.
- **Resztkowy:**
 - **SS (Suma kwadratów resztkowych):** 85,016,649.12
 - **MS (Średnia suma kwadratów resztkowych):** 116,781.11

ANALIZA WARIANCJI

	<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>Istotność F</i>
Regresja	1	3245936847	3245936847	27795,05	0
Resztkowy	728	85016649,12	116781,111		
Razem	729	3330953497			

Współczynniki regresji:

- **Przecięcie (Intercept):** 1583.71
- **Zmienna X1 (t):** 10.01, co oznacza, że każda jednostka zmiany w czasie (t) skutkuje wzrostem obrotów o 10.01 jednostki.

	<i>Współczynniki</i>	<i>Błąd standardowy</i>	<i>t Stat</i>	<i>Wartość-p</i>	<i>Dolne 95%</i>	<i>Górne 95%</i>	<i>Dolne 95,0%</i>
Przecięcie	1583,713783	25,32218712	62,5425354	5,3E-295	1534,001	1633,427	1534,001
Zmienna X 1	10,00637497	0,060019592	166,718476	0	9,888543	10,12421	9,888543

Te wyniki wskazują, że model jest silnie związany z danymi, a zmienna czasowa (t) ma istotny wpływ na przewidywaną wartość obrotu.

3. Błąd względny

Błąd względny został obliczony jako:

$$\text{Błąd względny} = \frac{|\text{Rzeczywiste wartości} - \text{Przewidywane wartości}|}{\text{Rzeczywiste wartości}}$$

Statystyki błędu względnego:

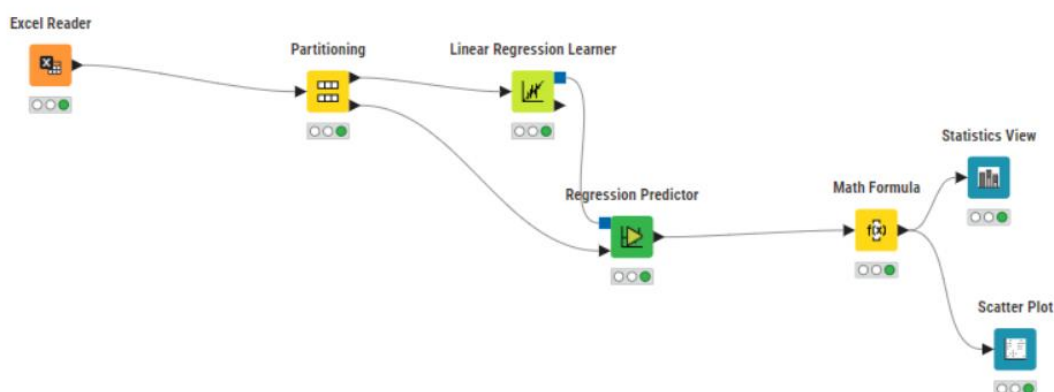
- **Średni błąd względny:** 0.0642 (6.42%), co oznacza, że średnio przewidywania są o 6.42% mniej dokładne niż rzeczywiste wartości.
- **Maksymalny błąd względny:** 0.5622 (56.22%), co wskazuje na przypadki, w których model przewidział wartości znacznie odbiegające od rzeczywistych.
- **Minimalny błąd względny:** 0.0045 (0.45%), co oznacza, że w innych przypadkach model przewidział bardzo bliskie wartości do rzeczywistych.

Średnia	MAX	MIN
0,064179	0,562181	0,004479

Wnioski dotyczące błędu względnego:

- **Średni błąd względny** na poziomie 6.42% jest relatywnie niski, co sugeruje, że model dobrze przewiduje obroty w większości przypadków.
- **Maksymalny błąd** wynoszący 56.22% wskazuje na istniejące przypadki, w których przewidywania są znacznie błędne. Należy zwrócić uwagę na te punkty, ponieważ mogą one wskazywać na anomalie w danych lub na konieczność dalszej optymalizacji modelu.

Zadanie zostało również wykonane w programie knime.



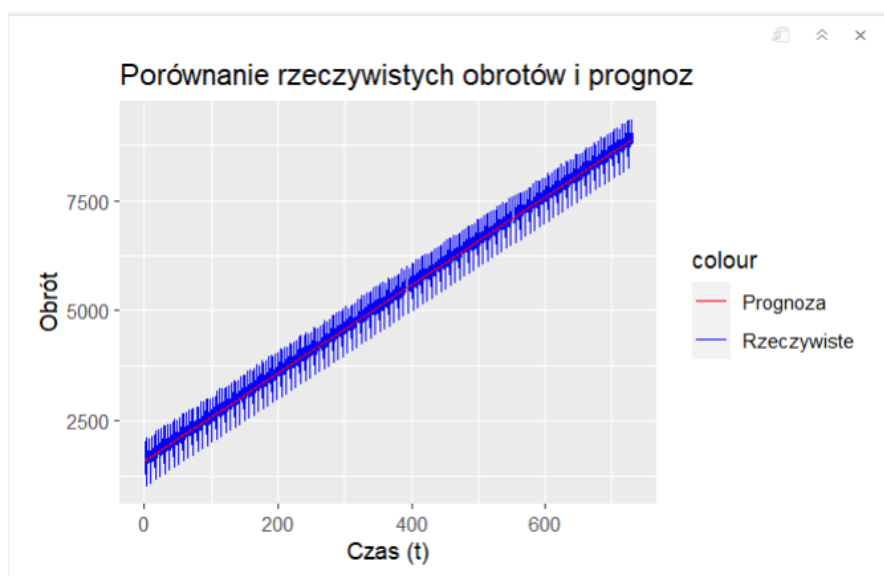
Ze względu na inne zaokrąglenia błąd i współczynniki nieznacznie się różnią.

Variable String	Coeff. Number (double)	Std. Err. Number (double)
t	9.946	0.071
Intercept	1,604.453	29.898

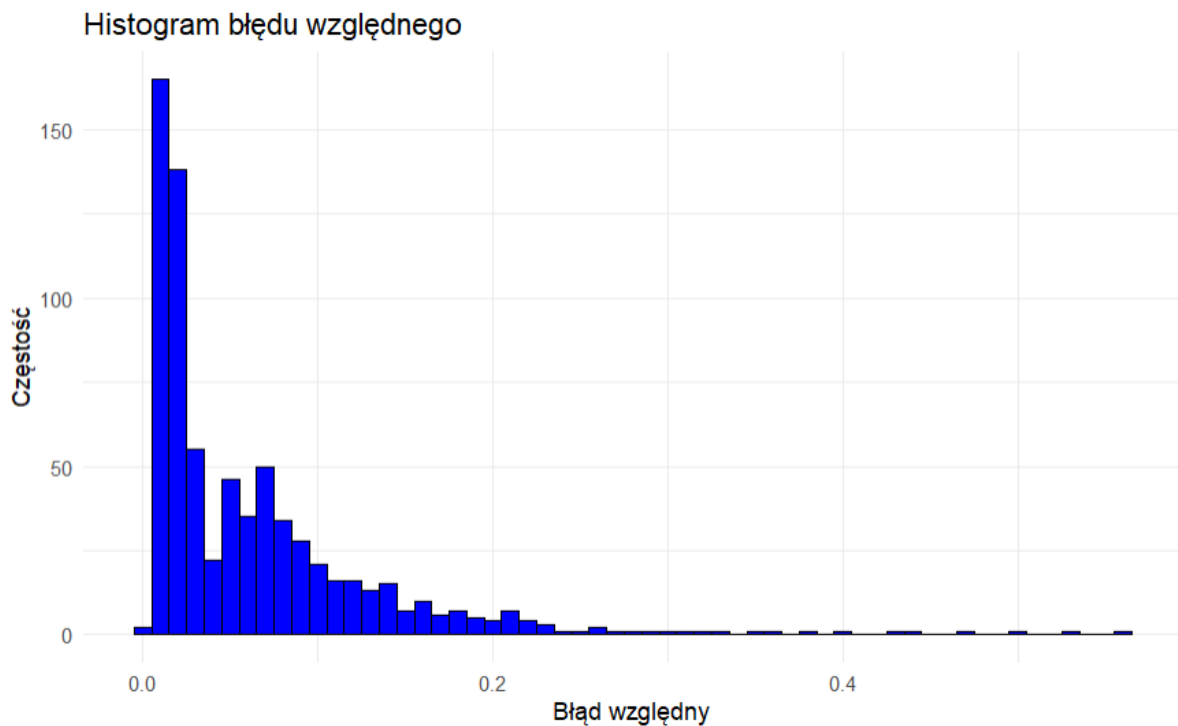
4. Wykres

Na wykresie porównującym rzeczywiste i przewidywane wartości, widać:

- **Rzeczywiste wartości** (niebieska linia) są porównane z **przewidywanymi wartościami** (czerwona linia).
- Idealnie, przewidywania (czerwona linia) powinny być bliskie rzeczywistym wartościom (niebieska linia). W tym przypadku widać, że w większości przypadków model dobrze odwzorowuje dane, ale występują pewne różnice, szczególnie w przypadkach o dużym błędzie.



Wykres wykonany w RStudio.



5. Wnioski końcowe

Model regresji liniowej wykonuje solidną pracę, ale:

- **R^2** jest bardzo wysokie (0.974), co oznacza, że model dobrze dopasowuje się do danych.
- **Średni błąd względny** wynosi 6.42%, co jest akceptowalnym wynikiem w kontekście prognozowania.
- **Maksymalny błąd** wskazuje na obszary, w których model może wymagać optymalizacji.

Możliwe kroki do poprawy modelu:

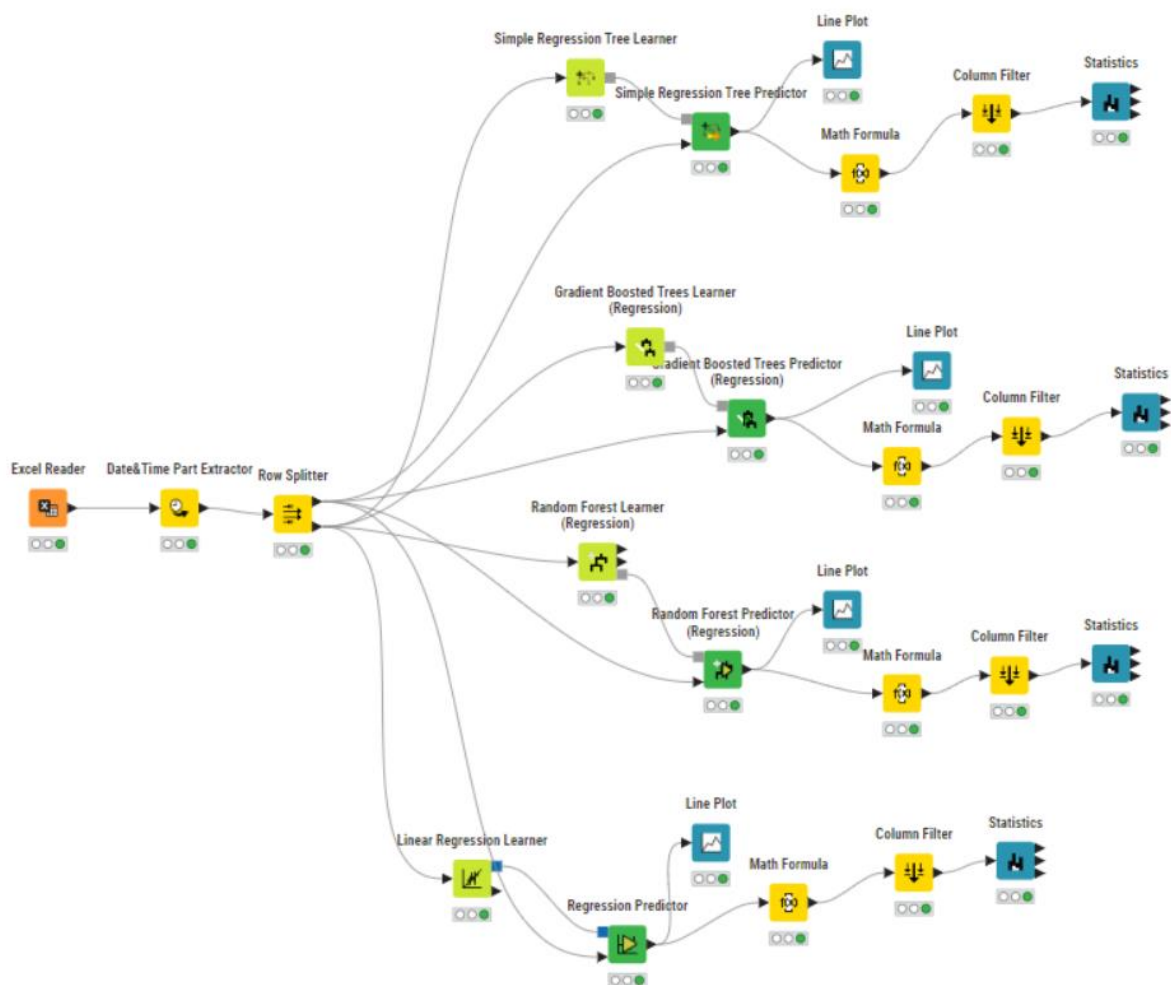
- **Analiza danych z błędami:** Zidentyfikowanie przypadków z wysokim błędem względnym i sprawdzenie, czy istnieją nietypowe lub nietypowo rozkładające się dane.
- **Optymalizacja modelu:** Można spróbować użyć innych metod regresji, takich jak regresja wielomianowa, aby uwzględnić bardziej złożoną zależność między czasem a obrotami.

Zadanie 3. (prognozowani2.xlsx)

Z dostępnych danych wybrać ostatni rok jako zbiór testowy. Pozostałe dane to zbiór testowy. Zbudować 3 różne modele oparte o narzędzia uczenia maszynowego, zbadać efektywność tych modeli. Opracować model liniowy wykorzystujący Państwa doświadczenie i porównać go z wcześniejszymi modelami.

Wykorzystane metody uczenia maszynowego:

1. Simple Regression Tree Learner
2. Gradient Boosted Trees Learner
3. Random Forest Learner

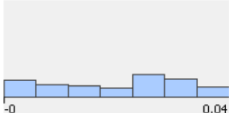


Opisane węzły przedstawiają proces odczytu, przetwarzania, modelowania i analizy danych. Kluczowe kroki obejmują:

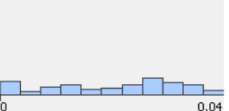
- Odczyt danych z Excela.
- Przekształcenia cech (ekstrakcja daty)
- Podział na zbiór uczący oraz testowy.
- Budowanie wybranego modelu.
- Filtrowanie kolumn w celu eliminacji zbędnych zmiennych.

Podsumowanie wyników w formie statystyk opisujących błąd względny

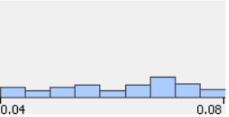
1. Simple Regression Tree Learner

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
błąd względny	7,54E-6	0,0206	0,0238	0,0414	0,012	-0,2142	-1,2566	0	0	0	

2. Gradient Boosted Trees Learner

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
błąd względny	8,60E-5	0,0224	0,0259	0,0435	0,0125	-0,3036	-1,1601	0	0	0	

3. Random Forest Predictor

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
błąd względny	0,0372	0,0613	0,0649	0,0817	0,0123	-0,3684	-1,0544	0	0	0	

Wnioski

1. Mean i Median:

- Najniższe wartości błędu względnego (średniego i mediany) osiąga **Simple Regression Tree Learner**, co sugeruje, że ma najdokładniejsze predykcje w przeciętnych przypadkach.
- Random Forest Predictor** ma najwyższe wartości średniego i mediany błędu względnego, co oznacza, że jego predykcje są mniej dokładne w porównaniu do pozostałych modeli.

2. Max (maksymalny błąd względny):

- Najniższy maksymalny błąd ma **Simple Regression Tree Learner** (0.0414), podczas gdy najwyższy ma **Random Forest** (0.0817). Oznacza to, że Random Forest ma większą tendencję do dużych odchyleń.

3. Std. Dev. (odchylenie standardowe):

- Wszystkie modele mają porównywalne odchylenia standardowe (~0.012), co oznacza podobny poziom zmienności błędu.

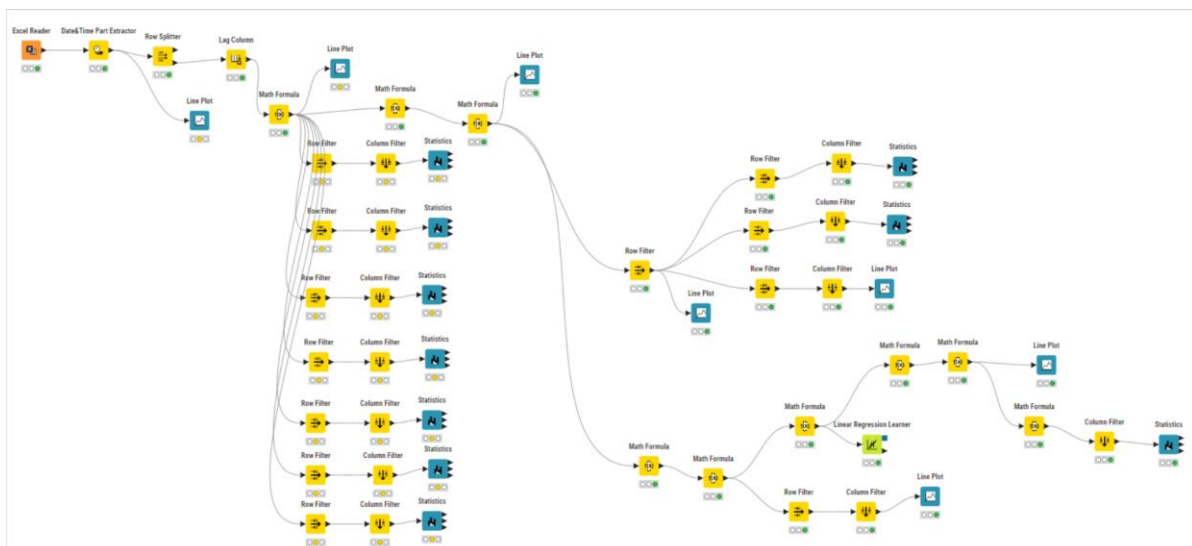
4. Skewness (skośność):

- Wszystkie modele mają ujemną skośność, co sugeruje, że w rozkładach występuje lewostronna asymetria

5. Kurtosis (kurtosis):

- Simple Regression Tree Learner** ma najbardziej płaski rozkład (kurtosis -1.2566), a **Random Forest Predictor** najbardziej zbliżony do normalnego rozkładu (kurtosis -1.0544).

Modele liniowe



Ogólny opis struktury:

Workflow przedstawia proces przetwarzania danych z wykorzystaniem różnych węzłów, takich jak:

1. Czytanie danych (Excel Reader),
2. Filtrowanie kolumn i wierszy,
3. Tworzenie i transformacja zmiennych (Math Formula),
4. Statystyczna analiza danych (Statistics),
5. Tworzenie modelu liniowego (Linear Regression Learner).

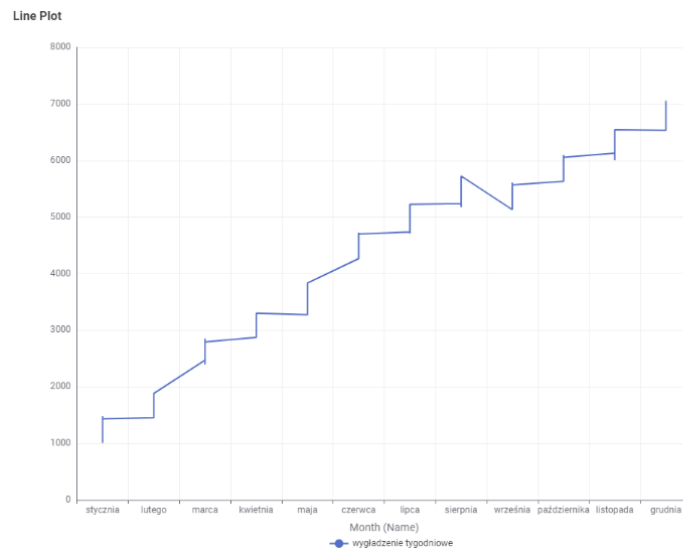
1. Źródło danych:

- Proces rozpoczyna się od węzła **Excel Reader**, który wczytuje dane wejściowe z pliku Excel.
- Kolejne węzły, takie jak **Date&Time-based Row Filter** oraz **Row Splitter**, pozwalają na podział danych na różne części (treningowe i testowe) i na przetwarzanie określonych przedziałów czasowych, w tym wypadku na dane z roku 2024 i dane z pozostałych lat

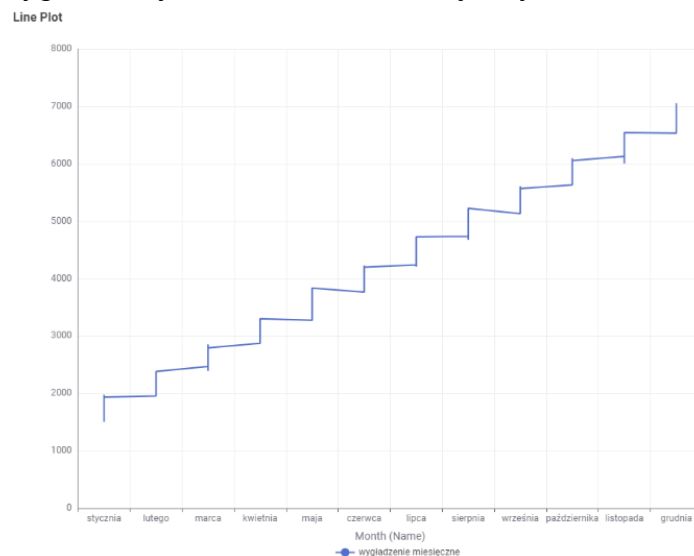
2. Tworzenie i transformacja zmiennych:

- Kilka węzłów **Math Formula** modyfikuje dane wejściowe poprzez obliczenia matematyczne, przekształcenia zmiennych, lub tworzenie nowych kolumn. Są one następnie przekazywane do kolejnych etapów analizy.
- Najistotniejsze było tworzenie kolumn:
 - *Przyrost dobowy* obrazujący jak obrót zmieniał się z dnia na dzień.
 - *Bonus dzień* uśredniająca przyrost dobowy za pomocą średniej wartości dla każdego dnia tygodnia na przestrzeni całości danych.
 - *Wyglądzenie tygodniowe* jako różnica między obrotem a bonusem dnia.

- *Bonus miesięczny* który powstał na podstawie wykresu wygładzenia tygodniowego na przestrzeni miesięcy.



- *Wygładzenie miesięczne* jako różnica między wygładzeniem tygodniowym a bonusem miesięcznym.



- *Dane wygładzone* jako różnica wygładzenia miesięcznego i iloczynu $500 \times \text{liczba konkurencji}$.

3. Filtrowanie danych:

- Węzły **Row Filter** i **Column Filter** służą do wybierania lub wykluczania określonych danych wierszy lub kolumn na podstawie różnych kryteriów (wybieranie określonych miesięcy lub lat z daty).

4. Modelowanie liniowe:

- Węzeł **Linear Regression Learner** jest wykorzystany do stworzenia modelu regresji liniowej, gdzie zmienną objaśnianą jest zmienna **obrót**.

Tabela 1 Model na podstawie LRL i LRP

Statistics on Linear Regression				
Variable	Coeff.	Std. Err.	t-value	P> t
t	14,8575	0,0019	7 748,3009	0.0
Intercept	1 986,1114	9,7055	204,6382	0.0
R-Squared: 0,9999				
Adjusted R-Squared: 0,9999				

Tabela 4 Model na podstawie własnych obliczeń

Statistics on Linear Regression				
Variable	Coeff.	Std. Err.	t-value	P> t
t	14,8552	0,0009	16 956,5583	0.0
Intercept	1 826,934	4,4342	412,0061	0.0
R-Squared: 1				
Adjusted R-Squared: 1				

Model oparty na Linear Regression Learner i Predictor:

- Wykazuje bardzo wysokie dopasowanie do danych, z wartościami równymi 0,9999, co świadczy o niemal idealnym dopasowaniu.
- Współczynnik dla zmiennej t wynosi 14,8575, przy bardzo niskim błędzie standardowym (0,0019), co wskazuje na wysoką precyzję estymacji.
- Intercept ma wartość 1986,1114 co również jest zgodne z danymi.

Model oparty na własnych obliczeniach:

- Charakteryzuje się jeszcze wyższymi wartościami dopasowania równymi 1, co wskazuje na idealne odwzorowanie danych.
- Współczynnik dla zmiennej t to 14,8552, z bardzo niskim błędem standardowym (0,0009), co również dowodzi wysokiej precyzji.
- Intercept ma wartość 1826,934, co różni się od modelu z Linear Regression Learner i Predictor, ale nadal pozostaje blisko rzeczywistości.

Rekomendacja:

- Oba modele są bardzo precyzyjne, ale model z Math Formula osiąga idealne dopasowanie $R^2=1$ co czyni go preferowanym wyborem w tej analizie.
- Linear Regression Learner i Predictor pozostaje jednak wiarygodną alternatywą z minimalnie niższym poziomem dopasowania.

5. Analiza statystyczna:

- Węzły **Statistics** generują szczegółowe statystyki opisowe dla danych, takie jak średnie, odchylenia standardowe, min, max, czy rozkład danych. Używany głównie do porównywania błędów względnych modeli.

Tabela 2 Błąd względny na podstawie własnych obliczeń


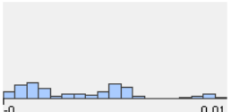
Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
błąd względny	9,95E-6	0,0111	0,0113	0,0922	0,0053	2,8168	25,1115	0	0	0	

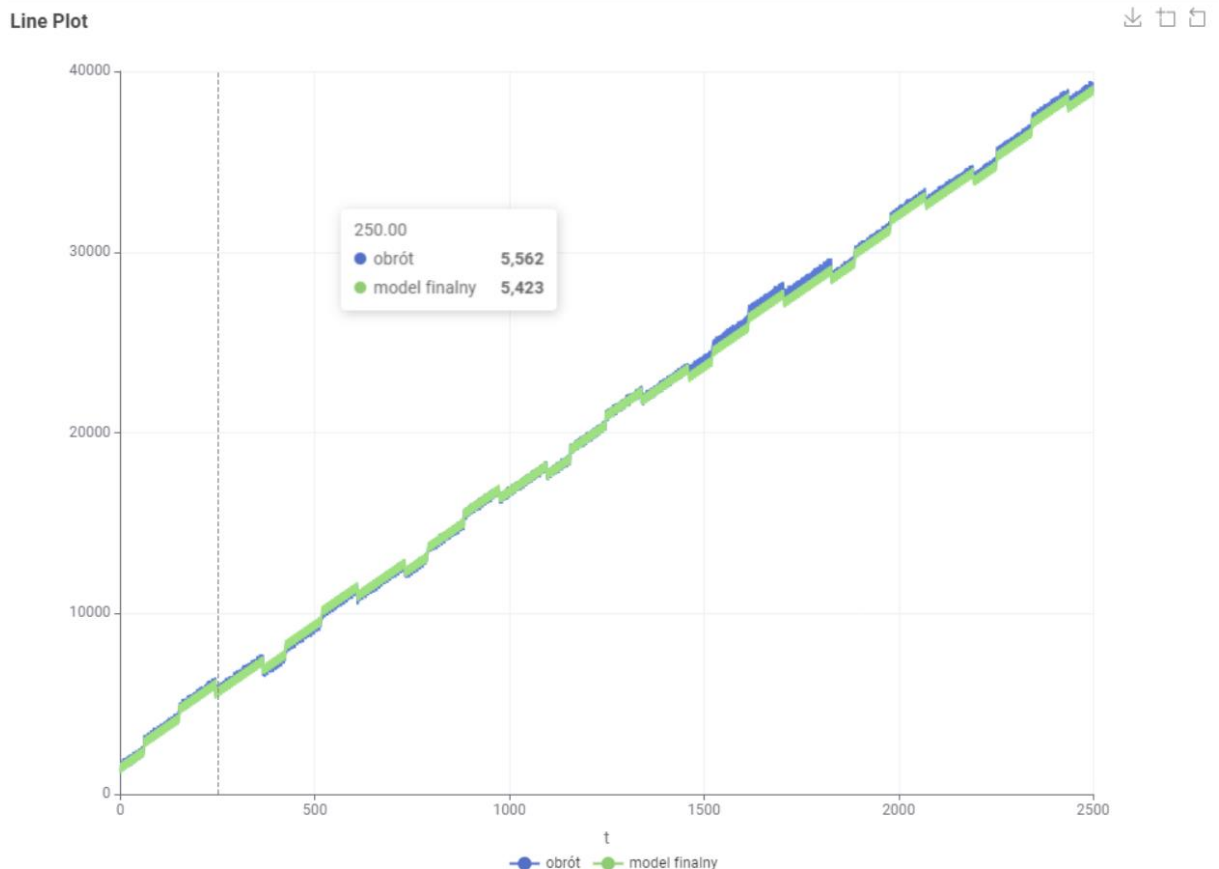
Tabela 3 Błąd względny na podstawie Linear Regression Learner i Predictor

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
błąd względny	1,51E-5	0,0031	0,0027	0,0094	0,0023	0,7596	-0,0957	0	0	0	

Porównanie:

- Model liniowy (Linear Regression Learner i Predictor) ma niższą średnią, medianę oraz maksymalny błąd względny, co wskazuje na lepszą ogólną jakość predykcji i większą stabilność wyników.
- Model oparty na Math Formula wykazuje większą dodatnią skośność i bardzo wysoką kurtozę, co oznacza obecność większej liczby błędów ekstremalnych i mniej równomierny rozkład błędów.
- Odchylenie standardowe jest większe w przypadku modelu Math Formula, co wskazuje na większą zmienność błędów w porównaniu do modelu Linear Regression Learner i Predictor.
- Różnice w średniej, maksymalnych błędach oraz rozkładzie błędów sugerują, że model Linear Regression Learner i Predictor jest bardziej niezawodny w różnych scenariuszach.

6. Wizualizacja:



Podsumowanie

➤ Porównanie modeli liniowych:

- Model oparty na węzłach Linear Regression Learner i Predictor osiągnął niemal idealne dopasowanie ($R\text{-Squared} = 0,9999$), co wskazuje na wysoką precyzję i stabilność predykcji.
- Model zbudowany za pomocą Math Formuła uzyskał idealne dopasowanie ($R\text{-Squared} = 1$), co oznacza doskonałą zgodność z danymi.

➤ Średnia i mediana błędów względnych:

- Model Linear Regression Learner i Predictor charakteryzował się niższymi wartościami średniej i mediany błędów, co świadczy o lepszej jakości ogólnej predykcji.

- Model Math Formula wykazywał bardziej symetryczny rozkład błędów oraz mniejszą skośność.

➤ **Odchylenie standardowe i maksymalne błędy względne:**

- Oba modele wykazały porównywalne wartości odchylenia standardowego oraz maksymalnego błędu względnego.

➤ **Zastosowane metody uczenia maszynowego:**

- W ramach projektu wykorzystano modele takie jak Simple Regression Tree Learner, Gradient Boosted Trees Learner oraz Random Forest Predictor. Spośród nich Simple Regression Tree Learner osiągnął najniższy błąd względny, co sugeruje jego wysoką efektywność w zadaniu.

➤ **Rekomendacje:**

- **Math Formula:** Dzięki idealnemu dopasowaniu modelu ($R\text{-Squared} = 1$), jest rekomendowany do zadań wymagających najwyższej precyzji predykcji.
- **Linear Regression Learner i Predictor:** Stanowi solidną alternatywę, szczególnie w sytuacjach wymagających stabilnych wyników i niższej zmienności błędów.