

# Usługi Sieciowe w Biznesie

Martyna Pitera

29 kwietnia 2024

Analiza i Predykcja danych za pomocą Apache Spark przy wykorzystaniu Databricks, SQL, Pythona i MLlib.

## 1 Charakterystyka usługi sieciowej

Usługa sieciowa to aplikacja, która udostępnia swoje funkcje innym aplikacjom przez sieć. Ma cechy takie jak dostępność, skalowalność, interoperacyjność i możliwość łatwej integracji z innymi usługami.

## 2 Apache Spark jako usługa sieciowa

Apache Spark to jedno z najbardziej innowacyjnych narzędzi w dziedzinie przetwarzania danych, które odgrywa kluczową rolę w analizie danych i predykcji w dzisiejszym biznesowym środowisku.

Apache Spark to system oper source o wysokiej wydajności, zaprojektowany do przetwarzania ogromnych zbiorów danych równolegle i rozproszenie ich na wielu komputerach. Jest on wykorzystywany przez wiele dużych firm i instytucji do analizy danych w czasie rzeczywistym, eksploracji danych czy uczenia maszynowego.

Dzięki architekturze rozproszonej, Apache Spark jest niezwykle skalowalny. Może on łatwo przetwarzać ogromne zbiory danych, zarówno na pojedynczym klastrze, jak i na wielu klastrach, co umożliwia elastyczne dostosowanie się do zmieniających się potrzeb biznesowych.

### 2.1 Apache Spark - Dostępność

Apache Spark jest łatwo dostępny poprzez różne interfejsy programistyczne, w tym interfejs wiersza poleceń, interaktywne konsolowe środowisko pracy oraz API dla wielu języków programowania, takich jak Scala, Java, Python i R. Ponadto istnieją usługi chmurowe, takie jak Databricks, które oferują zarządzane środowisko Sparka w chmurze, co ułatwia dostęp do niego.

## 3 Opis problemu

Projekt został przeprowadzony przy użyciu Apache Spark na platformie Databricks, wykorzystując Pythona i SQL.

Celem tego projektu jest analiza zbioru danych dotyczącego zawartości tłuszczu w ciele oraz prognozowanie tej zawartości na podstawie innych pomiarów ciała. (zbiór danych - <https://www.kaggle.com/datasets/fedesorianofat-prediction-dataset>)

Zbiór danych zawiera:

- Gęstość ustalona na podstawie ważenia pod wodą
- Procent zawartości tłuszczu w ciele według równania Siri'ego (1956)
- Wiek (lata)
- Waga (funty)
- Wzrost (cale)

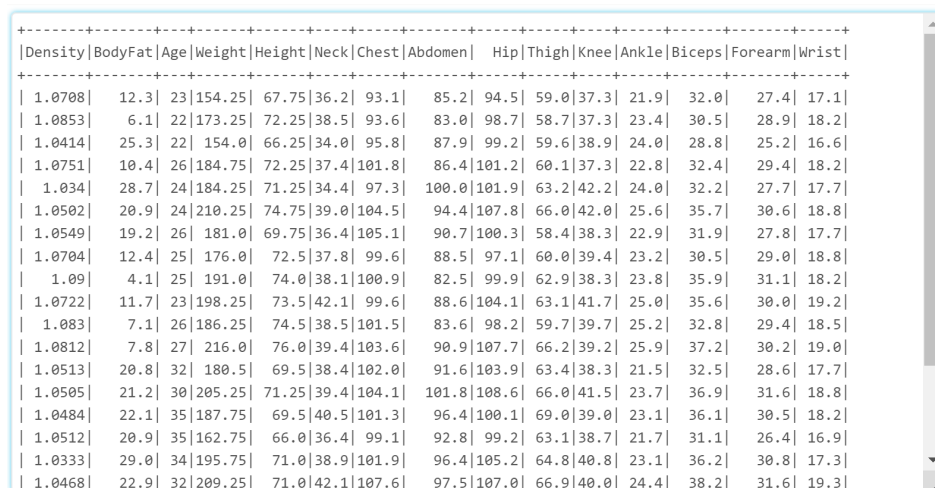
- Obwód szyi (cm)
- Obwód klatki piersiowej (cm)
- Obwód brzucha (cm)
- Obwód bioder (cm)
- Obwód uda (cm)
- Obwód kolana (cm)
- Obwód kostki (cm)
- Obwód bicepsa (cm)
- Obwód przedramienia (cm)
- Obwód nadgarstka (cm)

## 4 Przygotowanie ramki danych w środowisku Databricks

W pierwszym kroku zaimportowałam moduły niezbędne do wczytania danych z biblioteki PySpark oraz tworzenie sesji Sparka, która jest punktem wejścia do interakcji z frameworkiem Apache Spark.

Następnie zdefiniowałam strukturę danych za pomocą klasy StructType, określając schemat DataFrame'a, w którym nazwy kolumn oraz ich typy danych są reprezentowane przez obiekty klasy StructField. Każda kolumna jest reprezentowana przez obiekt klasy StructField, zawierający nazwę, typ danych i flagę określającą, czy kolumna może być pusta (True lub False).

Kolejnym krokiem było wczytanie danych z pliku CSV do ramki danych Sparka z ręcznie narzuconym schematem, co umożliwia dalszą analizę i przetwarzanie danych.



Density	BodyFat	Age	Weight	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle	Biceps	Forearm	Wrist
1.0708	12.3	23	154.25	67.75	36.2	93.1	85.2	94.5	59.0	37.3	21.9	32.0	27.4	17.1
1.0853	6.1	22	173.25	72.25	38.5	93.6	83.0	98.7	58.7	37.3	23.4	30.5	28.9	18.2
1.0414	25.3	22	154.0	66.25	34.0	95.8	87.9	99.2	59.6	38.9	24.0	28.8	25.2	16.6
1.0751	10.4	26	184.75	72.25	37.4	101.8	86.4	101.2	60.1	37.3	22.8	32.4	29.4	18.2
1.034	28.7	24	184.25	71.25	34.4	97.3	100.0	101.9	63.2	42.2	24.0	32.2	27.7	17.7
1.0502	20.9	24	210.25	74.75	39.0	104.5	94.4	107.8	66.0	42.0	25.6	35.7	30.6	18.8
1.0549	19.2	26	181.0	69.75	36.4	105.1	90.7	100.3	58.4	38.3	22.9	31.9	27.8	17.7
1.0704	12.4	25	176.0	72.5	37.8	99.6	88.5	97.1	60.0	39.4	23.2	30.5	29.0	18.8
1.09	4.1	25	191.0	74.0	38.1	100.9	82.5	99.9	62.9	38.3	23.8	35.9	31.1	18.2
1.0722	11.7	23	198.25	73.5	42.1	99.6	88.6	104.1	63.1	41.7	25.0	35.6	30.0	19.2
1.083	7.1	26	186.25	74.5	38.5	101.5	83.6	98.2	59.7	39.7	25.2	32.8	29.4	18.5
1.0812	7.8	27	216.0	76.0	39.4	103.6	90.9	107.7	66.2	39.2	25.9	37.2	30.2	19.0
1.0513	20.8	32	180.5	69.5	38.4	102.0	91.6	103.9	63.4	38.3	21.5	32.5	28.6	17.7
1.0505	21.2	30	205.25	71.25	39.4	104.1	101.8	108.6	66.0	41.5	23.7	36.9	31.6	18.8
1.0484	22.1	35	187.75	69.5	40.5	101.3	96.4	100.1	69.0	39.0	23.1	36.1	30.5	18.2
1.0512	20.9	35	162.75	66.0	36.4	99.1	92.8	99.2	63.1	38.7	21.7	31.1	26.4	16.9
1.0333	29.0	34	195.75	71.0	38.9	101.9	96.4	105.2	64.8	40.8	23.1	36.2	30.8	17.3
1.0468	22.9	32	209.25	71.0	42.1	107.6	97.5	107.0	66.9	40.0	24.4	38.2	31.6	19.3

Rysunek 1: Wczytana ramka danych.

Po wczytaniu ramki, przekonwertowałam wartości z cali na metry oraz z funtów na kilogramy w kolumnach Wzrost i Waga.

## 4.1 Statystyki dotyczące danych

```
display(df.describe())
```

	summary	Density	BodyFat	Age	Weight	Height	Neck	Chest	Abdomen
1	count	252	252	252	252	252	252	252	252
2	mean	1.055573809523809	19.15079365079365	44.88492063492063	81.15867860476192	1.7817797619047617	37.99206349206346	100.82420634920639	92.55595238095235
3	stddev	0.01903143417152082	8.368740413029712	12.602039722717862	13.330687810724323	0.09303653700708002	2.4309132340195085	8.43047553192002	10.783076801381702
4	min	0.995	0.0	22	53.750652	0.7493	31.1	79.3	69.4
5	max	1.1089	47.5	81	164.72193479999999	1.97485	51.2	136.2	148.1

5 rows

```
display(df.describe())
```

	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle	Biceps	Forearm	Wrist
1	252	252	252	252	252	252	252	252	252	252	252
2	1.7817797619047617	37.99206349206346	100.82420634920639	92.55595238095235	99.90476190476186	59.405952380952364	38.59047619047622	23.10238095238097	32.273412698412706	28.663888888888888	18.229761904761904
3	0.09303653700708002	2.4309132340195085	8.43047553192002	10.783076801381702	7.164057666842286	5.249952028401046	2.4118045870187563	1.6948933981786372	3.021273751250864	2.020691165026929	0.9335849289587025
4	0.7493	31.1	79.3	69.4	85.0	47.2	33.0	19.1	24.8	21.0	15.8
5	1.97485	51.2	136.2	148.1	147.7	87.3	49.1	33.9	45.0	34.9	21.4

5 rows

Za pomocą funkcji `describe` wyświetliłam podstawowe statystyki opisowe dla wszystkich kolumn numerycznych w ramce danych, takie jak liczba elementów, średnia, odchylenie standardowe, minimum i maksimum.

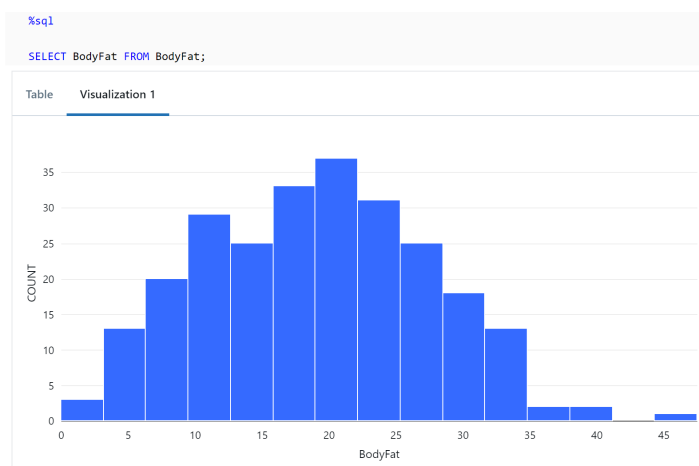
Przykładowo średni wiek osób poddanych pomiarom wynosił około 44/45 lat, średnia wysokość 1,78 metra, a maksymalna waga badanych (badanej) osoby wyniosła ponad 164 kilogramy.

Według American Journal of Clinical Nutrition () istnieją zdrowe procentowe wartości tkanki tłuszczowej zależne od wieku. Dla osób w wieku od 20 do 39 lat kobiety powinny dążyć do posiadania 21% do 32% tkanki tłuszczowej. Mężczyźni powinni mieć od 8% do 19%. Dla osób w wieku od 40 do 59 lat kobiety powinny mieć od 23% do 33%, a mężczyźni powinni mieć około od 11% do 21%. Jeśli masz od 60 do 79 lat, kobiety powinny mieć od 24% do 35% tkanki tłuszczowej, a mężczyźni od 13% do 24%.

Kobiety naturalnie posiadają wyższy procent tkanki tłuszczowej niż mężczyźni. Ich tkanka tłuszczowa również naturalnie wzrasta wraz z wiekiem.

## 5 Eksploracyjna analiza danych

W tym kroku wykonałam analizę eksploracyjną danych za pomocą języka zapytań SQL w środowisku Apache Spark. Za pomocą polecenia `'df.createOrReplaceTempView('BodyFat')` stworzyłam tymczasowy widok tabeli o nazwie `'BodyFat'`. Dzięki temu możliwe było wykonywanie zapytań SQL na danych zawartych w ramce danych. Następnie wyświetliłam histogramy wartości zawartości tłuszczu w ciele, co pozwoli na wstępną analizę rozkładu danych i ocenę ich charakterystyki. Podobne histogramy wyświetliłam dla wszystkich kolumn z ramki danych.

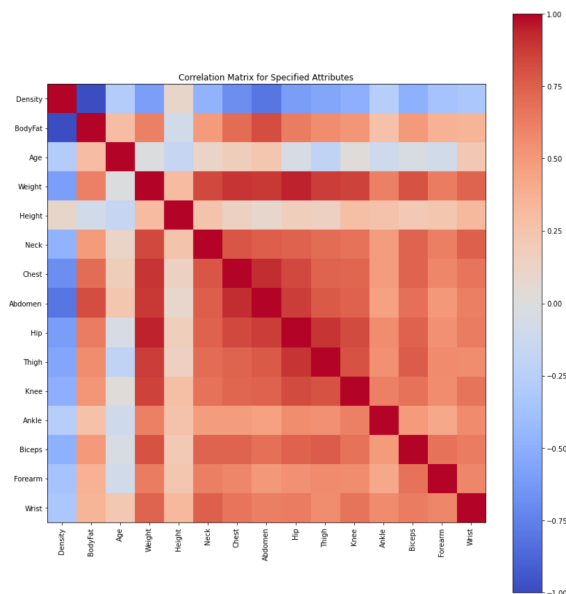


## 6 Analiza korelacji

Po zaimportowaniu niezbędnych modułów: `Correlation` z `pyspark.ml.stat` oraz `VectorAssembler` z `pyspark.ml.feature`, przekonwertowałam dane z ramki na wektorowy format za pomocą `VectorAssembler`. Każda kolumna z ramki danych jest traktowana jako składowa wektora. Następnie obliczyłam macierz korelacji za pomocą metody `Correlation.corr()`, która zwraca macierz korelacji między wszystkimi zmiennymi w wektorze. Macierz korelacji przekształciłam na format listy, aby można ją było użyć do utworzenia nowej ramki danych `corrdf`.

Za pomocą biblioteki `matplotlib` wygenerowałam wykres macierzy korelacji.

Dzięki `imshow()` wyświetlany jest wykres macierzy korelacji, w którym każdy piksel reprezentuje poziom korelacji między dwiema zmiennymi. Etykiety na osiach x i y są ustawiane na nazwy kolumn ramki danych 'BodyFat', a kolorowanka (`colorbar`) reprezentuje zakres wartości korelacji.



Rysunek 2: Macierz korelacji.

```
Correlation to BodyFat for Density -0.9877824021639853
Correlation to BodyFat for BodyFat 1.0
Correlation to BodyFat for Age 0.29145844013522204
Correlation to BodyFat for Weight 0.6124140022026475
Correlation to BodyFat for Height -0.08949537985440173
Correlation to BodyFat for Neck 0.4905918534410396
Correlation to BodyFat for Chest 0.7026203388938641
Correlation to BodyFat for Abdomen 0.813432284781049
Correlation to BodyFat for Hip 0.6252009175086624
Correlation to BodyFat for Thigh 0.5596075319940894
Correlation to BodyFat for Knee 0.5086652428854677
Correlation to BodyFat for Ankle 0.265969770306373
Correlation to BodyFat for Biceps 0.49327112589161554
Correlation to BodyFat for Forearm 0.3613869031997192
Correlation to BodyFat for Wrist 0.34657486452658576
```

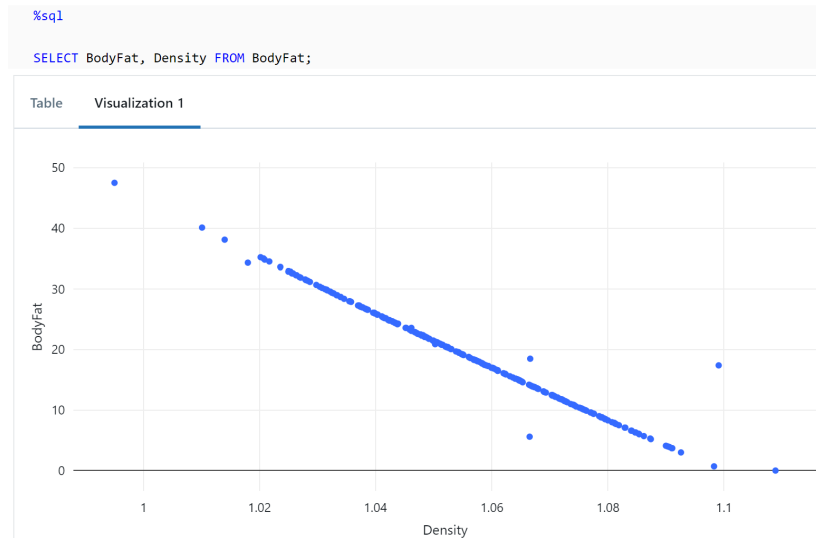
Rysunek 3: Współczynniki między zawartością tłuszczu w ciele (BodyFat) a innymi cechami.

Im bliższa wartość korelacji jest do 1, tym silniejszy związek istnieje między dwiema zmiennymi. Z kolei wartości bliskie -1 oznaczają silny związek ujemny, czyli kiedy jedna zmienna rośnie, druga maleje.

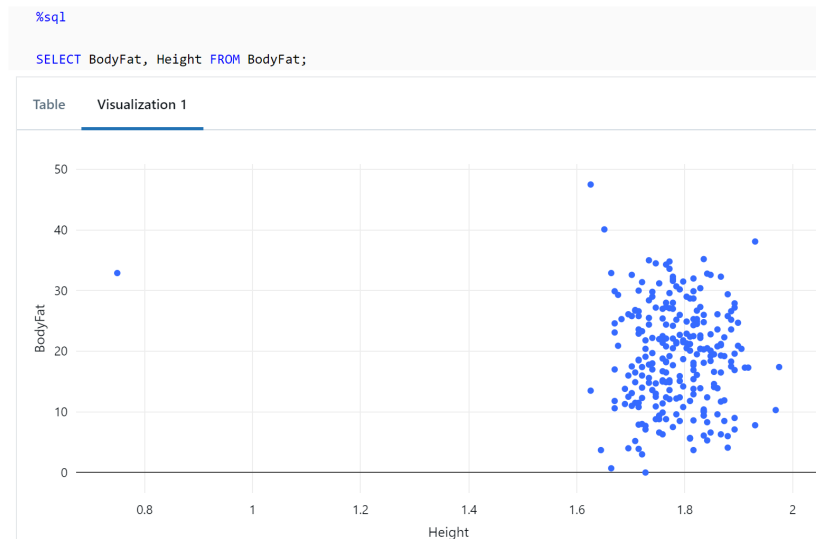
Cechy takie jak gęstość ciała, obwód klatki piersiowej i brzucha wykazują silną korelację z zawartością tłuszczu. Waga, wiek i obwód bioder wykazują umiarkowaną korelację, podczas gdy wzrost i obwód kostki mają słabsze związki z zawartością tłuszczu.

## 7 Eksploracja zależności między cechami

Przy użyciu SQL, wygenerowałam wykresy punktowe, które pozwoliły zobaczyć zależność między zawartością tłuszczu w ciele a poszczególnymi cechami. Punkty, które zbierają się wzdłuż jakiejś linii lub wykazują jakikolwiek wzór, sugerują pewną zależność między zmiennymi. Jeśli punkty są rozproszone losowo, sugeruje to brak związku między nimi.



Rysunek 4: Zależność między zawartością tłuszczu a gęstością ciała.



Rysunek 5: Zależność między zawartością tłuszczu a wzrostem.

Na podstawie powyższych wykresów punktowych można wywnioskować, że występuje zależność między zawartością tłuszczu a gęstością ciała, natomiast ciężko dostrzec jakąś zależność między zawartością tłuszczu a wzrostem.

## 8 Predykcja zawartości tkanki tłuszczowej przy wykorzystaniu MLLib

### 8.1 Przygotowanie danych

Przygotowanie danych do uczenia maszynowego polegało na użyciu VectorAssembler, który jest transformacją łączącą określoną listę kolumn w jedną kolumnę wektorową. Jest to przydatne narzędzie do połączenia surowych cech i cech wygenerowanych przez różne transformery cech w pojedynczy wektor, w celu szkolenia modeli uczenia maszynowego.

Następnie podzieliłam dane na zestawy treningowe i testowe do opracowania modelu uczenia maszynowego. Zestaw treningowy jest używany do uczenia modelu na wzorcach w danych, podczas gdy zestaw testowy służy do oceny, jak dobrze model generalizuje się do nowych, nieznanych dotąd danych. Pomaga to ocenić działanie modelu.

### 8.2 Regresja liniowa

Za pomocą klasy LinearRegression z modułu pyspark.ml.regression stworzyłam obiekt lr regresji liniowej, określając niektóre parametry:

- featuresCol='features': Określa kolumnę cech, która będzie używana do trenowania modelu.
- labelCol='BodyFat': Określa kolumnę etykiet, którą model będzie próbował przewidzieć.
- maxIter=10: Określa maksymalną liczbę iteracji algorytmu optymalizacji.
- regParam=0.3: Parametr regularyzacji, który kontroluje stopień regularyzacji modelu.
- elasticNetParam=0.8: Parametr, który kontroluje mieszaną regularyzacji L1 i L2 (Lasso i Ridge).

Dopasowałam model do danych treningowych za pomocą metody fit(), co spowodowało wyuczenie modelu regresji liniowej na tych danych.

```
trainingSummary = lr_model.summary
print("RMSE: %f" % trainingSummary.rootMeanSquaredError)
print("r2: %f" % trainingSummary.r2)

RMSE: 0.810689
r2: 0.990579
```

Rysunek 6: Podsumowanie modelu na zestawie treningowym.

Kiedy wartość R2 wynosi 0,99, oznacza to, że około 99% różnic lub zmienności, które obserwujemy w wartościach "BodyFat", można wyjaśnić za pomocą zbudowanego przez nas modelu. Innymi słowy, model bardzo dobrze uchwytuje i wyjaśnia wzorce i zmienność w danych dotyczących "BodyFat". Im bliżej wartości R-kwadratowej do 1, tym lepiej model dopasowuje się do danych.

RMSE (średnia kwadratowa błędu) mówi nam, jak dobrze przewidywania naszego modelu pasują do rzeczywistych wartości. Jednak aby zrozumieć, czy RMSE jest dobra czy zła, musimy porównać wynik z podstawowymi wartościami, takimi jak średnia, najmniejsze i największe wartości w naszych danych. To porównanie pomaga nam zobaczyć, czy model dobrze przewiduje, biorąc pod uwagę zakres wartości w zbiorze danych.

```
train_df.describe().show()
```

summary		BodyFat
count		178
mean	18.79662921348315	
stddev	8.375798461820827	
min		3.0
max		47.5

Rysunek 7: Statystyki modelu na zestawie treningowym.

Wartość RMSE wynosząca 0,810689 jest stosunkowo mała w porównaniu z zakresem wartości BodyFat, które wynoszą od 3,0 do 47,5.

Średnią kwadratową błędu (RMSE) można interpretować jako średnią oczekiwaną różnicę  $\pm$  między wartością przewidywaną a rzeczywistą. Jest to odchylenie standardowe reszt (różnica między wartością obserwowaną a wartością przewidywaną dla cechy).

RMSE dla modelu regresji liniowej na zestawie treningowym wynosi 0,810689, co oznacza, że średnio przewidywana wartość różni się o 0,810689 punktu procentowego od wartości rzeczywistej.

prediction	BodyFat	features
35.24403421319681	34.3	[1.018, 34.3, 35.0, ...]
37.7442456249542	35.2	[1.0202, 35.2, 46.0...]
34.75389020147844	34.8	[1.0209, 34.8, 44.0...]
35.421301558733546	34.5	[1.0217, 34.5, 45.0...]
31.52339414091179	32.9	[1.025, 32.9, 44.0, ...]
31.553324400342575	32.0	[1.0269, 32.0, 41.0...]
31.714280798708785	31.6	[1.0279, 31.6, 48.0...]
31.188208798198872	31.5	[1.028, 31.5, 54.0, ...]
28.861293236829937	29.8	[1.0317, 29.8, 56.0...]
28.82683066826843	29.4	[1.0325, 29.4, 43.0...]

only showing top 10 rows

R Squared (R2) on test data = 0.994369  
Root Mean Squared Error (RMSE) on test data = 0.622103

Rysunek 8: Statystyki modelu na zestawie testowym.

Niższa wartość RMSE na danych testowych w porównaniu do danych treningowych sugeruje, że model dobrze uogólnia się do nowych, nieznanych danych.

Wartości w kolumnie 'prediction' wartości BodyFat, które zostały przewidziane za pomocą modelu, po usunięciu z ramki danych pierwotnych wartości BodyFat.

### 8.3 Drzewo regresji

Do prognozowania danych przetestowałam również model drzewa regresji.

```
+-----+-----+-----+
|          prediction|BodyFat|          features|
+-----+-----+-----+
| 37.73333333333333| 34.3|[1.018,34.3,35.0,...|
| 37.73333333333333| 35.2|[1.0202,35.2,46.0...|
| 37.73333333333333| 34.8|[1.0209,34.8,44.0...|
| 37.73333333333333| 34.5|[1.0217,34.5,45.0...|
|          33.25| 32.9|[1.025,32.9,44.0,...|
|30.966666666666658| 32.0|[1.0269,32.0,41.0...|
|30.966666666666658| 31.6|[1.0279,31.6,48.0...|
|30.966666666666658| 31.5|[1.028,31.5,54.0,...|
|          29.68| 29.8|[1.0317,29.8,56.0...|
|          29.68| 29.4|[1.0325,29.4,43.0...|
+-----+-----+-----+
only showing top 10 rows

Root Mean Squared Error (RMSE) on test data = 0.96897
```

Rysunek 9: Statystyki modelu na zestawie testowym.

### 8.4 Drzewo regresji wzmocnione gradientowo

```
+-----+-----+-----+
|          prediction|BodyFat|          features|
+-----+-----+-----+
|37.671498584197494| 34.3|[1.018,34.3,35.0,...|
| 36.41591704332286| 35.2|[1.0202,35.2,46.0...|
| 37.6031676789961| 34.8|[1.0209,34.8,44.0...|
| 36.41591704332286| 34.5|[1.0217,34.5,45.0...|
| 33.04101163010007| 32.9|[1.025,32.9,44.0,...|
| 31.06352318234042| 32.0|[1.0269,32.0,41.0...|
|30.671988504874715| 31.6|[1.0279,31.6,48.0...|
|30.839086395037228| 31.5|[1.028,31.5,54.0,...|
| 29.61138601494879| 29.8|[1.0317,29.8,56.0...|
|29.811665601100543| 29.4|[1.0325,29.4,43.0...|
+-----+-----+-----+
only showing top 10 rows

Root Mean Squared Error (RMSE) on test data = 0.891016
```

Rysunek 10: Statystyki modelu na zestawie testowym.

Średni błąd kwadratowy (RMSE) dla każdego modelu na danych testowych wyniósł:

- Regresja liniowa: 0.622103,
- Drzewo regresji: 0.96897,
- Drzewo regresji wzmocnione gradientowo: 0.891016.

Wartość RMSE dla modelu regresji liniowej jest najniższa, a więc występuje najmniejsza różnica między średnią przewidywaną wartością, a wartością rzeczywistą. Oznacza to, że przewidywane za pomocą regresji liniowej wartości są bliższe prawdziwym wartościom, niż w pozostałych modelach.










Te wyniki podkreślają skuteczność modelu regresji liniowej w przewidywaniu procentowej zawartości tłuszczu w ciele, przewyższając zarówno modele regresji drzewa decyzyjnego, jak i regresji drzew wzmacnianych gradientowo.

## 9 Podsumowanie

W projekcie uwzględniono:

1. Wczytywanie i wstępne przetwarzanie zestawu danych.
2. Analizę statystyczną danych.
3. Analizę eksploracyjną danych w celu odkrycia wzorców i wniosków.
4. Analizę korelacji w celu zrozumienia związków między zmiennymi.
5. Wykorzystanie trzech modeli do przewidywania procentowej zawartości tłuszczu w ciele.

Podsumowując, w ramach tego projektu nauczyłam się korzystać z platformy Databricks oraz narzędzia Apache Spark. Zdobycie tych umiejętności jest niezwykle wartościowe, ponieważ umożliwiają one skuteczne zarządzanie dużymi zbiorami danych oraz wykonywanie zaawansowanych analiz. Ponadto, umiejętność pracy z Databricks i Apache Spark jest często wymagana w ofertach pracy związanych z analizą danych i big data, co sprawia, że są to bardzo pożądane i poszukiwane przez pracodawców umiejętności.

	<b>Data Platform Architect</b> 26 880 – 33 936 PLN Data • Databricks • Azure • Data Factory Elitmind	<b>NOWA</b>	Zdalnie +2
	<b>Senior Azure Data Engineer</b> 23 520 – 28 560 PLN Data • Databricks • Azure • Azure Cloud Link Group	<b>NOWA</b>	Zdalnie
	<b>Senior Azure Specialist</b> 26 880 – 30 240 PLN Data • Databricks • Azure • Azure Data Factory Antal	<b>NOWA</b>	Kraków
	<b>Lead Data Software Engineer (Python, Azure)</b> 26 500 – 32 000 PLN Data • Databricks • Python • Azure Data Factory EPAM Systems	<b>NOWA</b>	Warszawa
	<b>Lead Big Data Engineer (AWS)</b> 25 000 – 30 000 PLN Data • Databricks • AWS • Python SoftServe	<b>NOWA</b>	Warszawa
	<b>Remote Senior Data Engineer</b> 30 207 – 40 276 PLN Data • Databricks • Scala • Spark Varwise		Zdalnie +5
	<b>Data Engineer Python</b> 28 000 – 33 000 PLN Data • Databricks • Python • PySpark Connectis_		Zdalnie

Rysunek 11: Oferty pracy ze strony nofluffjobs.com z wymaganą znajomością Databricks.