Michał Muzykant Filip Twardawa Adam Wowra	AUTOR	2024/2025	ROK AKAD.	PSI	SPEC.	2	GRUPA
Porównanie klasycznych modeli uczenia maszynowego i sztucznych sieci neuronowych (MLP)				TEMAT		2	NR SPRAWOZDANIA
5.12.2024		Zastosowanie sztucznej inteligencji		dr inż.Anna Czaderna-Lekka			
DATA WYKONANIA		PRZEDMIOT		PROWADZĄCY			

1. Cel

Celem projektu jest przewidzenie poziomu emisji dwutlenku węgla (CO_2) przez pojazdy w zależności od ich charakterystyk za pomocą algorytmów uczenia maszynowego, wraz z analizą zbioru danych i porównaniem skuteczności zastosowanych technik predykcyjnych.

2. Materiały i metody

2.1. Charakterystyka danych:

Zestaw danych "CO2 Emission by Vehicles" zawiera informacje na temat emisji dwutlenku węgla przez różne pojazdy w zależności od ich cech.

Główne cechy zestawu danych:

- *Producent pojazdu: Marka pojazdu, np. Ford, BMW, Toyota.*
- *Model pojazdu: Konkretna nazwa modelu pojazdu.*
- *Rok produkcji: Rok, w którym pojazd został wyprodukowany.*
- *Typ paliwa: Rodzaj używanego paliwa, np. benzyna, diesel, elektryczny.*
- *Pojemność silnika: Pojemność silnika wyrażona w litrach.*
- *Zużycie paliwa: Średnie zużycie paliwa przez pojazd, zazwyczaj podawane w litrach na 100 kilometrów.*
- *Emisja CO_2: Ilość emitowanego dwutlenku węgla, zazwyczaj wyrażona w gramach na kilometr.*

Ten zestaw danych jest przydatny do analizowania zależności między cechami pojazdów a ich emisją CO_2 wspierając badania nad efektywnością i wpływem pojazdów na środowisko.

2.2. Metody:

Regresja liniowa

Model liniowy stosowany jako baza do oceny bardziej złożonych metod,
umożliwia szybkie uzyskanie wyników przy jednoczesnym zapewnieniu interpretowalności.
Jest prosty w implementacji, ale może być mniej skuteczny w przypadku złożonych danych
o nieliniowych zależnościach.

K-Nearest Neighbors (kNN)

Algorytm oparty na klasyfikacji według najbliższych sąsiadów.
Dzięki swojej prostocie jest często stosowany w zadaniach eksploracyjnych,
jego skuteczność zależy od odpowiedniego doboru liczby sąsiadów (parametru k)
oraz odległości, co czyni go wrażliwym na skalowanie danych.

MLP (Multi-Layer Perceptron) – model podstawowy

Podstawowy model sieci neuronowej z dwoma ukrytymi warstwami.
Jest w stanie rozwiązywać nieliniowe problemy poprzez naukę złożonych zależności w danych.
Charakteryzuje się umiarkowaną złożonością obliczeniową
i stanowi punkt wyjścia do bardziej zaawansowanych implementacji.

MLP (Multi-Layer Perceptron) – model zoptymalizowany

Zaawansowany model sieci neuronowej z wieloma ukrytymi warstwami, zoptymalizowany
pod kątem architektury (liczba neuronów, warstw), funkcji aktywacji oraz hiper parametrów,
takich jak współczynnik uczenia czy regularyzacja.
Zapewnia wysoką wydajność i dokładność, szczególnie w zadaniach wymagających
analizy dużych i złożonych zbiorów danych.

3. Wyniki i dyskusja

3.1. Przygotowanie danych

Duplikaty:

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption City (L/100 km)	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)	Fuel Consumption Comb (mpg)	CO2 Emissions(g/km)
4	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	27	244
5	ACURA	RLX	MID-SIZE	3.5	6	AS6	Z	11.9	7.7	10.0	28	230
12	ALFA ROMEO	4C	TWO-SEATER	1.8	4	AM6	Z	9.7	6.9	8.4	34	193
13	ASTON MARTIN	DB9	MINICOMPACT	5.9	12	A6	Z	18.0	12.6	15.6	18	359
15	ASTON MARTIN	V8 VANTAGE	TWO-SEATER	4.7	8	AM7	Z	17.4	11.3	14.7	19	338
...	...	...	...	...	...	...	...	...	...	...	...	...
7356	TOYOTA	Tundra	PICKUP TRUCK - STANDARD	5.7	8	AS6	X	17.7	13.6	15.9	18	371
7365	VOLKSWAGEN	Golf GTI	COMPACT	2.0	4	M6	X	9.8	7.3	8.7	32	203
7366	VOLKSWAGEN	Jetta	COMPACT	1.4	4	AS8	X	7.8	5.9	7.0	40	162
7367	VOLKSWAGEN	Jetta	COMPACT	1.4	4	M6	X	7.9	5.9	7.0	40	163
7368	VOLKSWAGEN	Jetta GLI	COMPACT	2.0	4	AM7	X	9.3	7.2	8.4	34	196

2102 rows × 12 columns

Znaleziono 2102 duplikatów, które zostały usunięte.

Brakujące dane:

	0
make	0
model	0
vehicle_class	0
engine_size	0
cylinders	0
transmission	0
fuel_type	0
fuel_cons_city	0
fuel_cons_hwy	0
fuel_cons_comb	0
fuel_cons_comb_mpg	0
co2	0

dtype: int64

Brak pustych wartości

Dane nie zawierają brakujących wartości.

Struktura danych:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7385 entries, 0 to 7384
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Make                            7385 non-null   object
 1   Model                           7385 non-null   object
 2   Vehicle Class                   7385 non-null   object
 3   Engine Size(L)                  7385 non-null   float64
 4   Cylinders                       7385 non-null   int64
 5   Transmission                    7385 non-null   object
 6   Fuel Type                       7385 non-null   object
 7   Fuel Consumption City (L/100 km)  7385 non-null   float64
 8   Fuel Consumption Hwy (L/100 km)   7385 non-null   float64
 9   Fuel Consumption Comb (L/100 km)  7385 non-null   float64
 10  Fuel Consumption Comb (mpg)       7385 non-null   int64
 11  CO2 Emissions(g/km)               7385 non-null   int64
dtypes: float64(4), int64(3), object(5)
memory usage: 692.5+ KB
```

Rozmiar danych: Zbiór zawiera 7385 obserwacji, co daje wystarczająco dużą próbę do analiz statystycznych i modelowania predykcyjnego.

Typy danych: Dane są mieszane – zawierają zarówno liczby zmiennoprzecinkowe (float64), liczby całkowite (int64), jak i dane kategoryczne (object).

Pełność danych: Wszystkie kolumny mają kompletne dane, co oznacza brak konieczności imputacji brakujących wartości.

Kodowanie zmiennych kategorycznych:

Zmieniono kolumny (make, model, vehicle_class, transmission, fuel_type) na reprezentację liczbową za pomocą LabelEncoder i one-hot encoding.

Analiza podstawowych statystyk danych:

	engine_size	cylinders	fuel_cons_city	fuel_cons_hwy	fuel_cons_comb	fuel_cons_comb_mpg	co2
count	7385.000000	7385.000000	7385.000000	7385.000000	7385.000000	7385.000000	7385.000000
mean	3.160068	5.615030	12.556534	9.041706	10.975071	27.481652	250.584699
std	1.354170	1.828307	3.500274	2.224456	2.892506	7.231879	58.512679
min	0.900000	3.000000	4.200000	4.000000	4.100000	11.000000	96.000000
25%	2.000000	4.000000	10.100000	7.500000	8.900000	22.000000	208.000000
50%	3.000000	6.000000	12.100000	8.700000	10.600000	27.000000	246.000000
75%	3.700000	6.000000	14.600000	10.200000	12.600000	32.000000	288.000000
max	8.400000	16.000000	30.600000	20.600000	26.100000	69.000000	522.000000

Analiza:

Pojemność silnika (engine_size): Średnia wynosi 3.16 litra, z minimalną wartością 0.9 i maksymalną 8.4.
Istnieje duży rozrzut, co wskazuje na dużą różnorodność pojazdów (od małych do dużych silników).

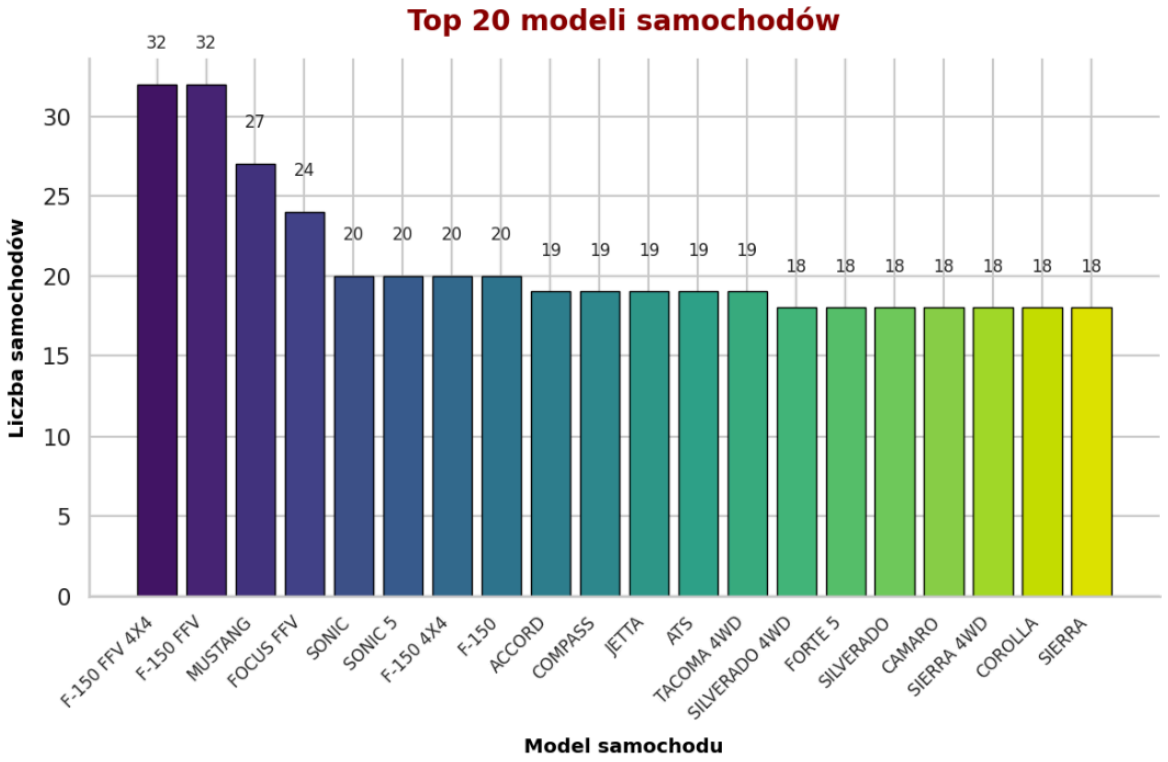
Liczba cylindrów (cylinders): Zmienna o średniej 5.62, a wartości rozciągają się od 3 do 16 cylindrów.
Wysoki rozrzut może wskazywać na różne klasy pojazdów.

Zużycie paliwa (fuel_cons_city, fuel_cons_hwy, fuel_cons_comb):
Wartości wskazują na typowy zakres zużycia paliwa, z najwyższymi wartościami w przypadku pojazdów o większych silnikach.

Emisja CO_2 (co2): Średnia emisja to 250.58 g/km, a zakres wynosi od 96 do 522 g/km.
Wartości są zgodne z oczekiwaniami – większe pojazdy emitują więcej CO_2.

3.2. Analiza wykresów:

A. Wykres kolumnowy 20 najczęściej występujących modeli samochodów

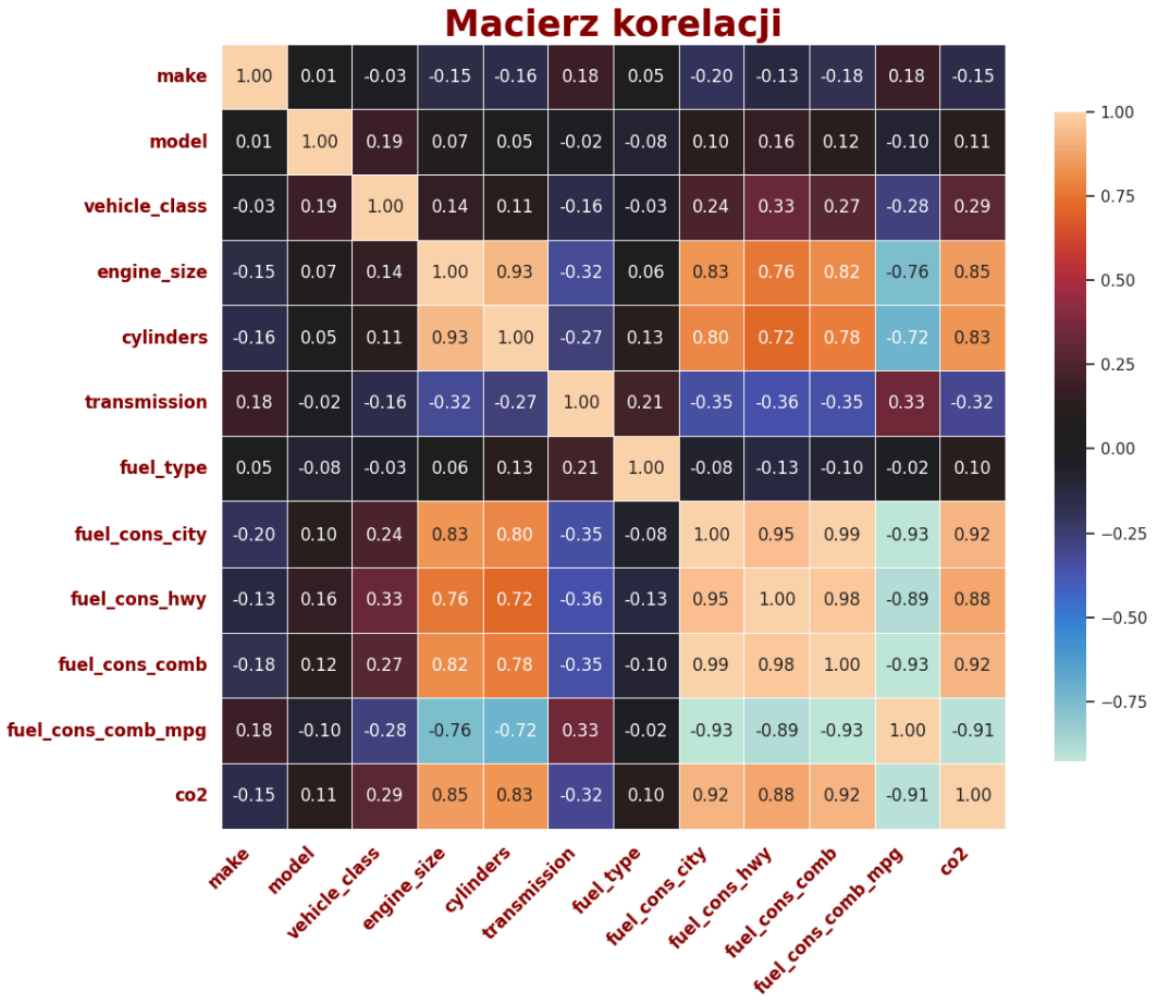


Analiza:

Modele o wysokiej liczbie wystąpień mogą wskazywać na popularność niektórych marek (np. Ford, BMW, Toyota).

Tego rodzaju analiza może pomóc w lepszym zrozumieniu struktury danych i potencjalnie w obniżeniu szumów w modelach predykcyjnych poprzez uwzględnienie powszechnych typów pojazdów.

B. Macierz korelacji między zmiennymi w zbiorze danych



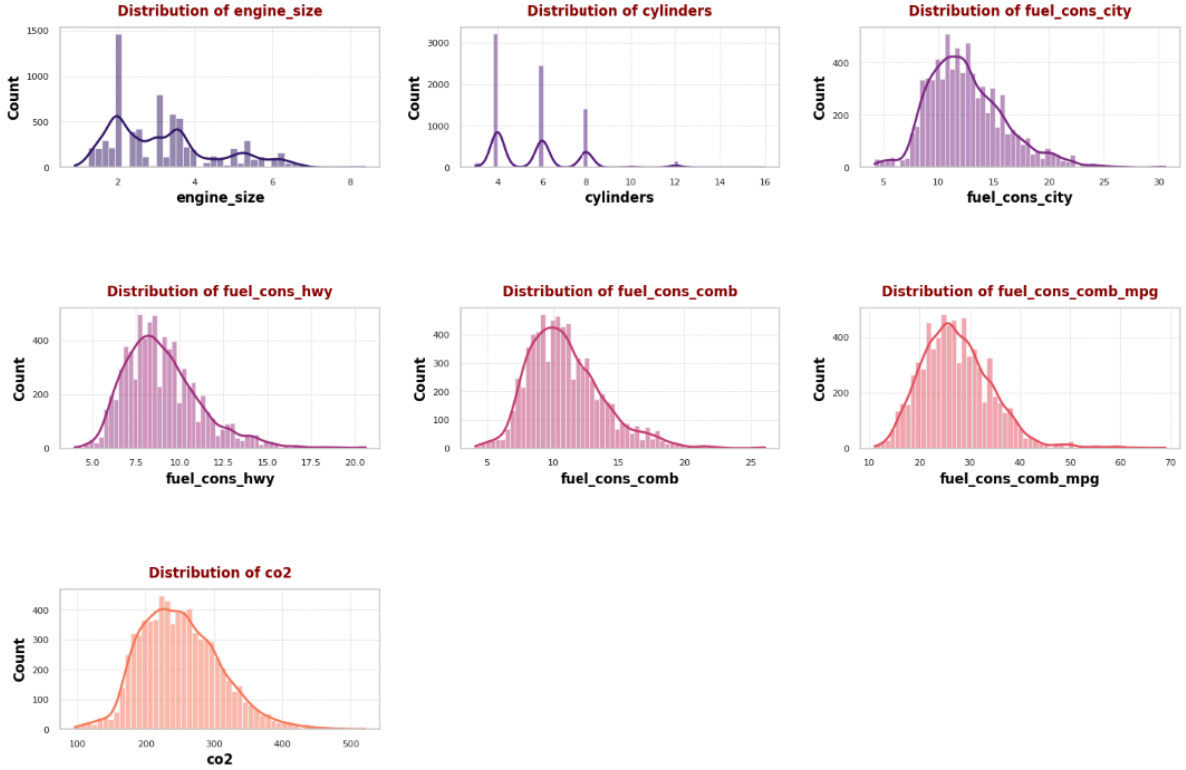
Analiza:

engine_size i co2: Silna korelacja dodatnia (0.85), co jest zgodne z oczekiwaniami – większe silniki generują wyższą emisję CO_2.

fuel_cons_comb i co2: Bardzo silna korelacja (0.92), co również potwierdza, że wyższe zużycie paliwa prowadzi do wyższej emisji.

fuel_cons_comb_mpg i co2: Korelacja ujemna (-0.91), co wskazuje na naturalną zależność – im bardziej efektywny pojazd (większa liczba mil na galon), tym mniejsza emisja CO_2.

C. *Wykresy histogramów z krzywą gęstości*



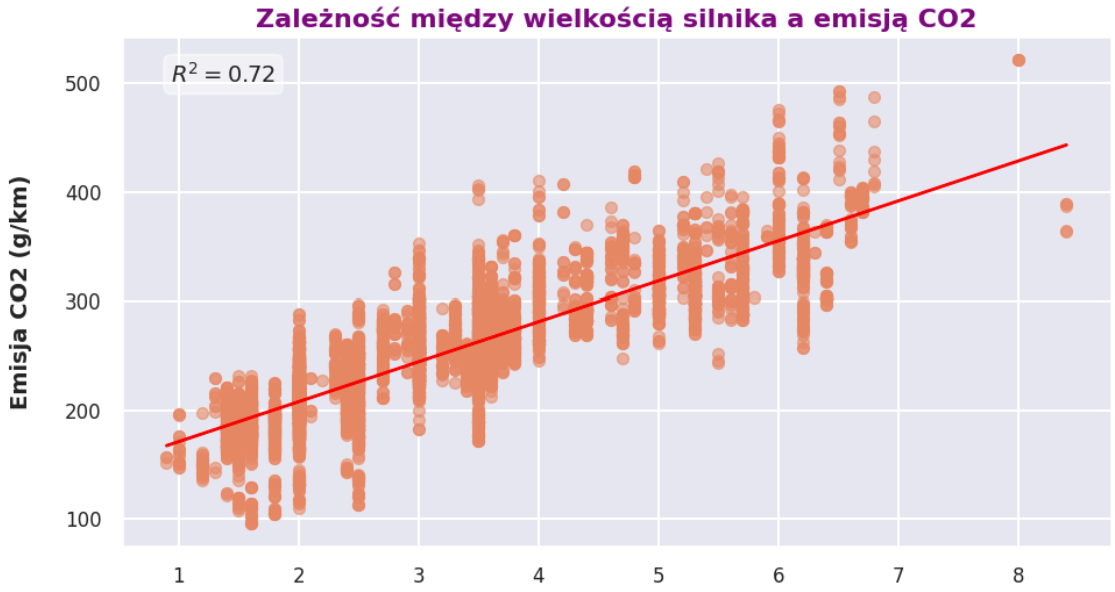
Analiza:

engine_size: Wartości skoncentrowane wokół średnich i wyższych pojemności silników, z niewielką liczbą ekstremalnych przypadków.

co2: Większość pojazdów generuje emisje w średnim zakresie (~200–300 g/km), z kilkoma pojazdami emitującymi znacznie więcej.

fuel_cons_city i fuel_cons_comb: Większość pojazdów zużywa paliwo w średnich ilościach, z wyraźnym rozkładem w okolicach 10–15 l/100 km, co jest typowe dla współczesnych pojazdów.

D. *Wykres rozrzutu z dopasowaną linią regresji: engine_size vs co2*



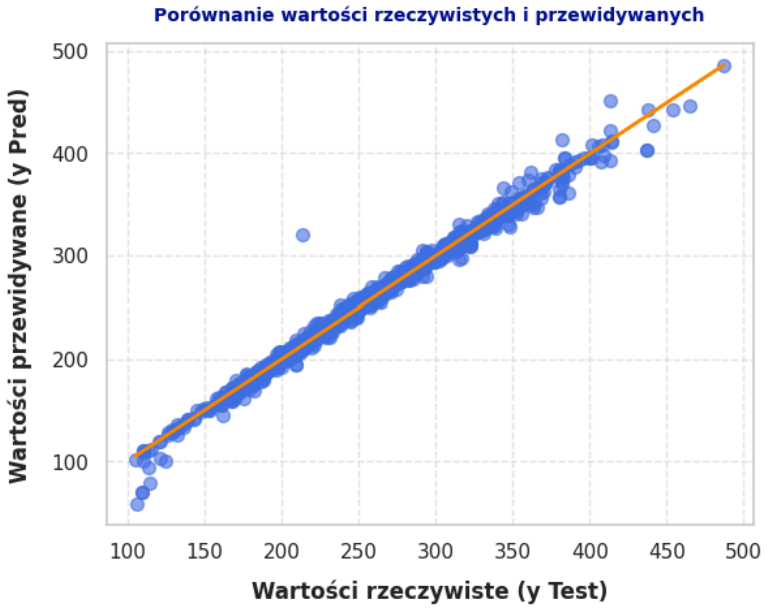
Analiza:

Zgodnie z intuicją, większy rozmiar silnika wiąże się z wyższą emisją CO_2.

Współczynnik R^2 (0.72) wskazuje na silną, ale nie doskonałą zależność liniową.

Nieliniowe zjawiska, takie jak różnice w technologii silników,
mogą wpływać na odstępstwa od tej prostej zależności.

E. *Wykres regresji: Zależność między wartościami rzeczywistymi a przewidywanymi (y_test vs y_pred)*



Analiza:

Jeśli modele predykcyjne działają poprawnie,
punkty na wykresie powinny leżeć blisko linii 45°
(gdzie wartości rzeczywiste są równe przewidywanym).

Wniosek :

Interpretacja dopasowania:

Wykres pokazuje, że wartości przewidywane y_Pred są bardzo zbliżone do wartości rzeczywistych y_Test, co wskazuje na wysoki poziom dopasowania modelu.
Idealne dopasowanie oznacza, że każda wartość przewidywana pokrywa się z wartością rzeczywistą .

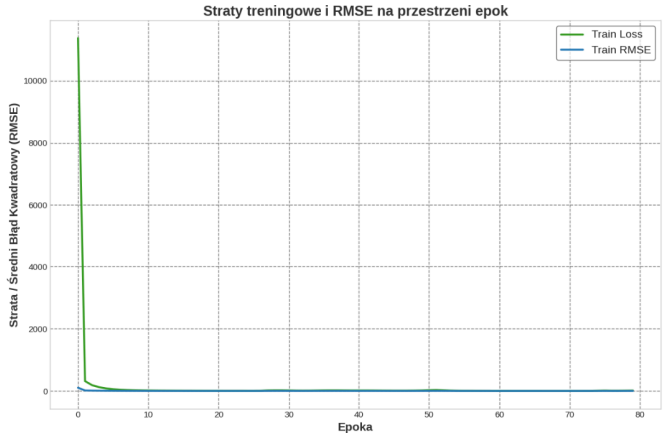
Małe odchylenia od pomarańczowej linii (idealnej) wskazują na niski poziom błędów systematycznych i losowych.

Wskaźniki metryk:

Wizualizacja potwierdza, że metryki takie jak R^2
(które oceniają, ile zmienności w danych testowych wyjaśnia model) powinny być bliskie 1.0.

Niskie błędy (np. MAE, RMSE) są również zgodne z zaobserwowaną bliskością punktów do linii idealnej.

F. Wykres metryk modelu: Straty i RMSE na przestrzeni epok



Analiza:

Kluczowe obserwacje:

Szybki spadek na początku:

> *Obie krzywe (strata treningowa i RMSE) na początku uczenia gwałtownie maleją, co oznacza, że model bardzo szybko dostosowuje swoje parametry, aby zminimalizować błąd na danych treningowych.*
>
> *Wartości straty oraz RMSE są bardzo wysokie w początkowych epokach, co jest typowe dla modelu, który rozpoczyna trening bez żadnej wiedzy.*

Stabilizacja po kilku epokach:

> *Po około 10 epokach zarówno strata treningowa, jak i RMSE osiągają niski i stabilny poziom.*
>
> *Stabilizacja oznacza, że model przestał znacząco poprawiać swoje działanie na danych treningowych, a proces uczenia przynosi coraz mniejsze zmiany w wydajności.*

Brak widocznych oznak przeuczenia:

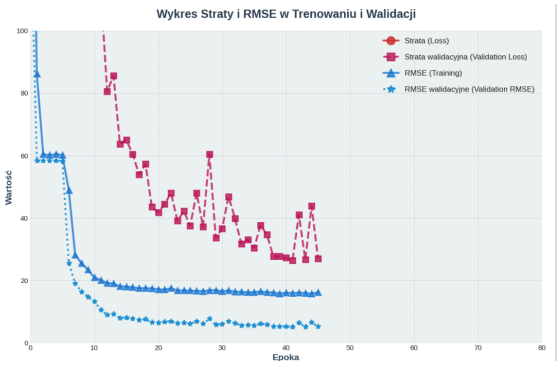
> *W przeciwieństwie do poprzedniego wykresu, tutaj widzimy jedynie metryki treningowe (brak walidacyjnych). Stabilny poziom straty i RMSE sugeruje, że model dobrze optymalizuje dane treningowe.*
>
> *Jednak brak danych walidacyjnych uniemożliwia ocenę, czy model generalizuje dobrze na dane, których nie widział podczas treningu.*

Wnioski:

Efektywne uczenie na danych treningowych:

Model szybko zbiega do niskiej straty i RMSE na zbiorze treningowym, co oznacza, że dobrze radzi sobie z optymalizacją swoich parametrów.

G. *Wykres Straty i RMSE w Trenowaniu i Walidacji*



Analiza:

Kluczowe obserwacje:

Strata i Strata walidacyjna (Loss i Validation Loss):

Na początku treningu zarówno strata treningowa (czerwona linia) jak i walidacyjna (różowa linia przerywana) są bardzo wysokie.

Strata treningowa szybko maleje i stabilizuje się na niskim poziomie po około 20 epokach.

Strata walidacyjna wykazuje większą niestabilność i większą wartość od straty treningowej. Oznacza to, że model może być narażony na przeuczenie (overfitting) – działa dobrze na danych treningowych, ale gorzej na danych walidacyjnych.

RMSE i RMSE walidacyjne (Training RMSE i Validation RMSE):

RMSE dla treningu (niebieska linia ciągła) maleje i stabilizuje się na niskim poziomie, co oznacza, że model poprawia swoje przewidywania na danych treningowych.

RMSE walidacyjne (niebieska linia przerywana) jest wyższe niż RMSE treningowe i również wykazuje większą niestabilność, co potwierdza możliwość przeuczenia.

Porównanie stabilności danych walidacyjnych i treningowych:

Dane walidacyjne (zarówno strata, jak i RMSE) są bardziej niestabilne, szczególnie po około 20 epokach, co sugeruje, że model nie generalizuje optymalnie.

Wnioski:

Overfitting: Niższe wartości straty i RMSE na danych treningowych w porównaniu do danych walidacyjnych wskazują, że model może być przeuczony.

Konieczność regularyzacji: Aby poprawić działanie modelu, warto wprowadzić metody regularyzacji, takie jak dropout, L1/L2 regularization, lub użyć technik zmniejszania przeuczenia (np. wcześniejszego zatrzymania treningu - early stopping).

Monitorowanie metryk: Wyższa niestabilność metryk walidacyjnych może sugerować, że należy sprawdzić jakość podziału danych na zbiór treningowy i walidacyjny lub zbadać odpowiednie hiperparametry (np. szybkość uczenia - learning rate).

3.3. Analiza metryk

Model	R^2_train	R^2_test	MAE_train	MAE_test	MSE_train	MSE_test	RMSE_train	RMSE_test
Linear Regression	0.9975	0.9898	1.87	3.27	8.53	35.23	2.92	5.94
KNN Regressor	0.9938	0.9901	2.65	3.40	21.31	34.22	4.62	5.85
MLP Base	0.9961	0.9932	3.04	3.40	13.28	23.34	3.64	4.83
MLP Optimized	0.9950	0.9901	2.90	3.92	16.99	34.12	4.12	5.84

Analiza:

a) *Linear Regression*

R^2_train (0.9975) vs. R^2_test (0.9898):

Rozbieżność pomiędzy zbiorami jest niewielka,
co sugeruje, że model generalizuje bardzo dobrze.

Model liniowy najprawdopodobniej uchwycił główne wzorce w danych.

MAE_test (3.27), MSE_test (35.23), RMSE_test (5.94):

MAE wskazuje, że przeciętny błąd modelu na danych testowych wynosi
około 3.27 jednostki.

Wysokie MSE (35.23) w stosunku do MAE sugeruje,
że model może być podatny na większe błędy w przypadku wartości odstających.

RMSE_train (2.92) vs. RMSE_test (5.94):

Podwojenie RMSE na zbiorze testowym może wskazywać,
że model nie jest idealnie dopasowany do bardziej skomplikowanych wzorców
w danych testowych.

b) *KNN Regressor*

R^2_train (0.9938) vs. R^2_test (0.9901):

Model osiąga bardzo podobne wyniki na obu zbiorach,
co oznacza dobrą generalizację.
Widać, że jest nieco mniej precyzyjny niż regresja liniowa na zbiorze treningowym.

MAE_test (3.40), MSE_test (34.22), RMSE_test (5.85):

MSE i RMSE są podobne do wartości z regresji liniowej, ale MAE jest wyższe,
co może oznaczać, że KNN nie radzi sobie dobrze
z pewnymi szczególnymi przypadkami.

RMSE_train (4.62) vs. RMSE_test (5.85):

Rozbieżność pomiędzy zbiorami wskazuje, że KNN może być mniej efektywny w uchwyceniu globalnych zależności, ale dobrze sobie radzi w lokalnych.

c) *MLP Base*

R^2_train (0.9961) vs. R^2_test (0.9932):

Wartości R^2 wskazują na bardzo dobrą zdolność do generalizacji.
MLP Base lepiej uchwyciło zależności w danych testowych niż modele liniowe i KNN.

MAE_test (3.40), MSE_test (23.34), RMSE_test (4.83):

Najniższy MSE oraz RMSE spośród wszystkich modeli.
Oznacza to, że model bazowy MLP lepiej radzi sobie z wartościami odstającymi i minimalizuje większe błędy.

MAE na poziomie 3.40 jest porównywalne z KNN,
co wskazuje, że typowe błędy są na podobnym poziomie.

RMSE_train (3.64) vs. RMSE_test (4.83):

Relatywnie niewielka różnica między zbiorami świadczy o dobrym dopasowaniu modelu i niskim ryzyku przetrenowania.

d) *MLP Optimized*

R^2_train (0.9950) vs. R^2_test (0.9901):

Wyniki bardzo podobne do KNN i regresji liniowej, co wskazuje,
że optymalizacja nie przyniosła znaczącej poprawy.

MAE_test (3.92), MSE_test (34.12), RMSE_test (5.84):

Wyższy MAE na zbiorze testowym wskazuje,
że model gorzej radzi sobie z przewidywaniem typowych przypadków niż inne modele.

MSE i RMSE są podobne do KNN i regresji liniowej,
ale gorsze niż w przypadku MLP Base.

RMSE_train (4.12) vs. RMSE_test (5.84):

Rozbieżność pomiędzy wynikami wskazuje,
że zoptymalizowana wersja MLP mogła być nadmiernie dopasowana do danych treningowych.

Wnioski:

1. *Najlepszy model: MLP Base*

   Dlaczego?:

   Najniższe wartości RMSE i MSE na danych testowych wskazują, że MLP Base jest najlepszy w minimalizowaniu dużych błędów i generalizowaniu na danych testowych.

2. *Alternatywa: Linear Regression*

   Dlaczego?:

   Uzyskuje bardzo konkurencyjne wyniki przy prostocie implementacji i interpretacji.

3. *Odrzucenie MLP Optimized*

   Dlaczego?:

   Wyniki nie wykazują przewagi nad bazowym MLP, a dodatkowe obciążenie obliczeniowe czyni go mniej atrakcyjnym.

4. Podsumowanie

Przedstawienie najważniejszych obserwacji, wniosków.

Bazowy model MLP osiągnął najniższe wartości MSE i RMSE, co czyni go najlepszym wśród badanych modeli.

Zoptymalizowany model MLP nie poprawił wyników, a wyższa złożoność obliczeniowa czyni go mniej efektywnym.

Klasyczne algorytmy uczenia maszynowego również okazały się skuteczne z uwagi na liczne zależności i korelacje między wielkością silnika czy zużyciem paliwa, a emisją co2.

5. Bibliografia

1. *Aggarwal, C. C. (2020). Neural Networks and Deep Learning. Springer. https://books.google.pl/books?hl=pl&lr=&id=BVnHDwAAQBAJ&oi=fnd&pg=PP1, Dostęp:05.12.2024*

2. *Jain, D., & Zhao, X. (2023). Advances in Neural Network Architectures. W: Proceedings of AI Trends and Applications (s. 265–279). Springer., https://link.springer.com/chapter/10.1007/978-3-031-33342-2_13, Dostęp: 05.12.2024.*

3. *Sharma, K., Shanti Pragnya, S. S., & Kumar, S. (2021). Artificial Intelligence Hybrid Deep Learning Models: Implementation and Challenges, https://d1wqtxts1xzle7.cloudfront.net/96319977/2107.13870v1-libre.pdf , Dostęp: 05.12.2024.*

4. *Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep Learning for Visual Understanding: A Review. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 12(1s), 1–33., https://dl.acm.org/doi/abs/10.1145/2990508 , Dostęp: 05.12.2024.*

5. *Feng, C., Zhang, J., & Chen, W. (2018). K-Nearest Neighbor Regression for Predicting Missing Data in Deep Learning Systems. Neurocomputing, 264, 177–188. , https://www.sciencedirect.com/science/article/pii/S0925231217306884 , Dostęp: 05.12.2024.*

6. *Raschka, S. (2019). Python Machine Learning. Packt Publishing.,https://books.google.pl/books?hl=pl&lr=&id=28t1DwAAQBAJ&oi=fnd&pg=PP1, Dostęp:05.12.2024.*

7. *Shanti Pragnya, S. S., & Das, A. (2018). Accuracy Analysis of Continuance by Using Classification and Regression Algorithms in Python., https://www.researchgate.net/profile/Swayanshu-Shanti-Pragnya-2/publication/328567484_Accuracy_Analysis_of_Continuance_by_using_Classification_and_Regression_Algorithms_in_Python/links/5bd55b664585150b2b8b357c/Accuracy-Analysis-of-Continuance-by-using-Classification-and-Regression-Algorithms-in-Python.pdf ,Dostęp: 05.12.2024.*