

W ramach projektu „Car Price Prediction” z wykorzystaniem wielokrotnej regresji liniowej oraz sieci neuronowej, należy wykonać następujące kroki:

### 1. Zebranie i przygotowanie danych

- **Zebranie danych:** Pozyskanie zbioru danych dotyczącego cen samochodów, zawierającego kluczowe zmienne takie jak: wiek pojazdu, przebieg, moc silnika, marka, model, rok produkcji, paliwo, typ nadwozia, itd.
- **Czyszczenie danych:** Sprawdzenie, czy zbiór danych zawiera brakujące wartości, błędy lub duplikaty. Należy zająć się brakującymi wartościami (np. uzupełnić średnią lub medianą, usunąć niekompletne rekordy).
- **Eksploracja danych (EDA):** Wizualizacja danych, aby zrozumieć ich rozkład, zależności między zmiennymi, wykrycie anomalii, np. za pomocą histogramów, wykresów punktowych czy korelacji.

### 2. Przygotowanie danych do modelowania

- **Podział danych:** Podziel zbiór danych na zestawy: treningowy (do nauki modelu) i testowy (do walidacji modelu), np. w proporcji 80/20.
- **Skalowanie cech:** Przeprowadź normalizację lub standaryzację zmiennych (szczególnie tych numerycznych) w celu ujednolicenia zakresu wartości, co może mieć znaczenie w przypadku sieci neuronowych.
- **Kodyfikacja zmiennych kategoriowych:** Jeżeli w zbiorze danych znajdują się zmienne kategoriowe (np. marka pojazdu, rodzaj paliwa), zamień je na wartości numeryczne (np. przy pomocy One-Hot Encoding lub Label Encoding).

### 3. Budowa i trening modelu regresji liniowej

- **Wielokrotna regresja liniowa:** Zbuduj model regresji liniowej, który uwzględnia wiele zmiennych predykcyjnych (cech). Wybierz zmienne, które mogą mieć wpływ na cenę pojazdu.
- **Trening modelu:** Wytrenuj model na danych treningowych, używając techniki minimalizacji błędu (np. metoda najmniejszych kwadratów).
- **Ewaluacja modelu:** Zmierz jakość modelu, stosując miary takie jak  $R^2$ , MSE (Mean Squared Error) lub MAE (Mean Absolute Error). Zastosuj dane testowe do oceny predykcji modelu.

### 4. Budowa i trening sieci neuronowej

- **Architektura sieci neuronowej:** Zbuduj model sieci neuronowej (np. z jedną lub kilkoma warstwami ukrytymi), odpowiednio dostosowując liczbę neuronów w każdej warstwie.
- **Funkcja aktywacji:** Wybierz odpowiednie funkcje aktywacji dla warstw (np. ReLU dla warstw ukrytych i liniową dla warstwy wyjściowej).
- **Kompilacja modelu:** Ustal funkcję straty (np. MSE) oraz optymalizator (np. Adam, RMSProp).
- **Trening modelu:** Wytrenuj model na danych treningowych, ustalając liczbę epok oraz odpowiedni rozmiar batcha.

- **Ewaluacja sieci neuronowej:** Sprawdź działanie sieci na zbiorze testowym, obliczając miary takie jak MSE i porównaj wyniki z modelem regresji liniowej.

## 5. Optymalizacja modeli

- **Selekcja cech:** Wykonaj selekcję najważniejszych cech (np. metoda RFE – Recursive Feature Elimination) i sprawdź, czy uproszczenie modelu poprawia jego skuteczność.
- **Tuning hiperparametrów:** Dla sieci neuronowej przeprowadź tuning hiperparametrów, takich jak liczba warstw, liczba neuronów, współczynnik uczenia (learning rate), aby poprawić wydajność modelu.
- **Walidacja krzyżowa:** Zastosuj walidację krzyżową, aby upewnić się, że model generalizuje dobrze na różnych podzbiorach danych.

## 6. Porównanie wyników

- **Analiza wyników:** Porównaj skuteczność modelu regresji liniowej oraz sieci neuronowej na podstawie miar takich jak MSE, MAE czy  $R^2$ .
- **Wnioski:** Na podstawie wyników wybierz lepszy model i podsumuj, który model lepiej przewiduje ceny samochodów oraz jakie czynniki mają największy wpływ na predykcję.

## 7. Wizualizacja wyników i raportowanie

- **Wizualizacja wyników:** Utwórz wykresy, takie jak porównanie cen rzeczywistych z przewidywanymi, wykresy reszt oraz wykresy przedstawiające wpływ poszczególnych cech na cenę pojazdu.
- **Raport końcowy:** Opracuj dokumentację opisującą cały proces, wyniki analizy, porównanie modeli i wnioski, które można wyciągnąć z predykcji cen samochodów.

Te kroki umożliwią wykonanie kompleksowej analizy i przewidywania cen samochodów przy użyciu zarówno regresji liniowej, jak i sieci neuronowej.