SPRAWOZDANIE
Zajęcia: Uczenie Maszynowe
Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 1 Data 25.02.2025 Temat: Praktyczne zastosowanie regresji liniowej w analizie danych. Wariant 10	Krzysztof Świerczek Informatyka II stopień, stacjonarne, 1semestr, gr. A

1. Polecenie: Praktyczne zastosowanie regresji liniowej w analizie danych.

2. Github:
https://github.com/Krzycho165/STUDIA

In [5]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVC
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score, classi

# 1. Wczytanie danych, wstępne przetworzenie i kategoryzacja danych
data = pd.read_csv("Smoker_Epigenetic_df.csv")
data = data.dropna()
data['Smoking Status'] = data['Smoking Status'].map({'current': 1, 'former': 0,
data['Gender'] = data['Gender'].map({'f': 0, 'm': 1})
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVC
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score, classi

# 1. Wczytanie danych, wstępne przetworzenie i kategoryzacja danych
data = pd.read_csv("Smoker_Epigenetic_df.csv")
data = data.dropna()
data['Smoking Status'] = data['Smoking Status'].map({'current': 1, 'former': 0,
data['Gender'] = data['Gender'].map({'f': 0, 'm': 1})
```

In [6]:
```
# 2. Wyodrębnienie cech, zmiennych docelowych i podział na zbiór treningowy i te

X = data.iloc[:, 4:]  # cechy epigenetyczne
y_regression = data['cg03683899']  # zmienna docelowa regresji
y_classification = data['Smoking Status']  # zmienna docelowa klasyfikacji

X_train_reg, X_test_reg, y_train_reg, y_test_reg = train_test_split(X, y_regress
X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X, y_classif
```

In [7]:
```
# 3. Regresja liniowa i ocena modelu regresji

lin_reg = LinearRegression()
lin_reg.fit(X_train_reg, y_train_reg)
y_pred_reg = lin_reg.predict(X_test_reg)

mse = mean_squared_error(y_test_reg, y_pred_reg)
r2 = r2_score(y_test_reg, y_pred_reg)
print("Regresja liniowa - MSE:", mse)
print("Regresja liniowa - R²:", r2)
```

```
Regresja liniowa - MSE: 1.6641275981354876e-30
Regresja liniowa - R²: 1.0
```

In [9]:
```
# 4. Klasyfikacja binarna
svm_clf = SVC(kernel='linear', C=1.0, random_state=42)
svm_clf.fit(X_train_clf, y_train_clf)
y_pred_clf = svm_clf.predict(X_test_clf)
```

In [10]:
```
# 5. Ocena modelu klasyfikacji

accuracy = accuracy_score(y_test_clf, y_pred_clf)
print("Klasyfikacja binarna - Accuracy:", accuracy)
```

```
print("Klasyfikacja binarna - Raport klasyfikacji:")
print(classification_report(y_test_clf, y_pred_clf, zero_division=0))
```

```
Klasyfikacja binarna - Accuracy: 0.704
Klasyfikacja binarna - Raport klasyfikacji:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        37
           1       0.70      1.00      0.83        88

    accuracy                           0.70       125
   macro avg       0.35      0.50      0.41       125
weighted avg       0.50      0.70      0.58       125
```

In [ ]:

3. Wnioski z ćwiczenia

Podczas realizacji tego ćwiczenia zdobyliśmy praktyczne doświadczenie w przetwarzaniu danych oraz budowie i ocenie modeli regresyjnych i klasyfikacyjnych przy użyciu biblioteki `scikit-learn`. Analiza obejmowała zarówno problem przewidywania wartości numerycznych, jak i klasyfikacji binarnej, co pozwoliło na zapoznanie się z różnymi metodami modelowania danych.
Pierwszym krokiem było wczytanie i przygotowanie danych za pomocą biblioteki `pandas`. Uczestnik nauczył się wczytywać pliki CSV oraz przeprowadzać wstępną obróbkę danych, w tym usuwanie brakujących wartości i konwersję zmiennych kategorycznych na wartości numeryczne, co jest kluczowym etapem w analizie danych.
Szczególnie istotne było zastosowanie metody `map()`, pozwalającej na przekształcenie zmiennych takich jak „Smoking Status” czy „Gender” na postać liczbową, umożliwiającą dalsze modelowanie.
Ważnym aspektem było również odpowiednie przygotowanie zbioru danych poprzez jego podział na część treningową i testową. Użycie funkcji `train_test_split()` umożliwiło podzielenie danych w sposób kontrolowany, co pozwoliło na rzetelne sprawdzenie skuteczności modeli zarówno w zadaniu regresji, jak i klasyfikacji.
W ramach analizy regresyjnej wykorzystano model regresji liniowej, który pozwolił na przewidywanie wartości epigenetycznej `cg03683899` na podstawie cech opisujących badanych. Proces uczenia modelu i generowania prognoz pozwolił na zapoznanie się z pojęciem dopasowania modelu do danych. Wskaźniki jakości dopasowania, takie jak błąd średniokwadratowy (MSE) oraz współczynnik determinacji (R^2), pomogły w ocenie skuteczności modelu. Wynik R^2 równy 1.0 wskazywał na idealne dopasowanie modelu do danych, co mogło sugerować problem przeuczenia, czyli nadmiernego dopasowania modelu do konkretnego zbioru danych zamiast uogólnienia wyników na nowe obserwacje.
Dodatkowo, uczestnik przeprowadził klasyfikację binarną za pomocą modelu SVM (`SVC`). Wykorzystanie jądra liniowego oraz hiperparametru regularyzacji `C=1.0` pozwoliło na lepsze zrozumienie zasad działania klasyfikatorów opartych na wektorach nośnych. Po przeprowadzeniu predykcji dokonano oceny modelu przy użyciu metryk takich jak dokładność (`accuracy_score()`) oraz raport klasyfikacji (`classification_report()`). Generowanie raportu klasyfikacji pozwoliło na bardziej szczegółową analizę skuteczności modelu, a zastosowanie opcji `zero_division=0` umożliwiło uniknięcie problemów w przypadku klas, dla których nie dokonano żadnej predykcji. Zadanie pozwoliło mi zapoznać się z pełnym procesem przygotowania danych do analizy.