

```
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, classification_report

# 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, classification_report

# 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 testowy

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_regression,
X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X, y_classification,
```

```
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 []: