Podstawy uczenia maszynowego

07.03.2024

Laboratorium 1

Preprocessing danych

Łukasz Stępień, Kacper Fus

1. Cel zadania.

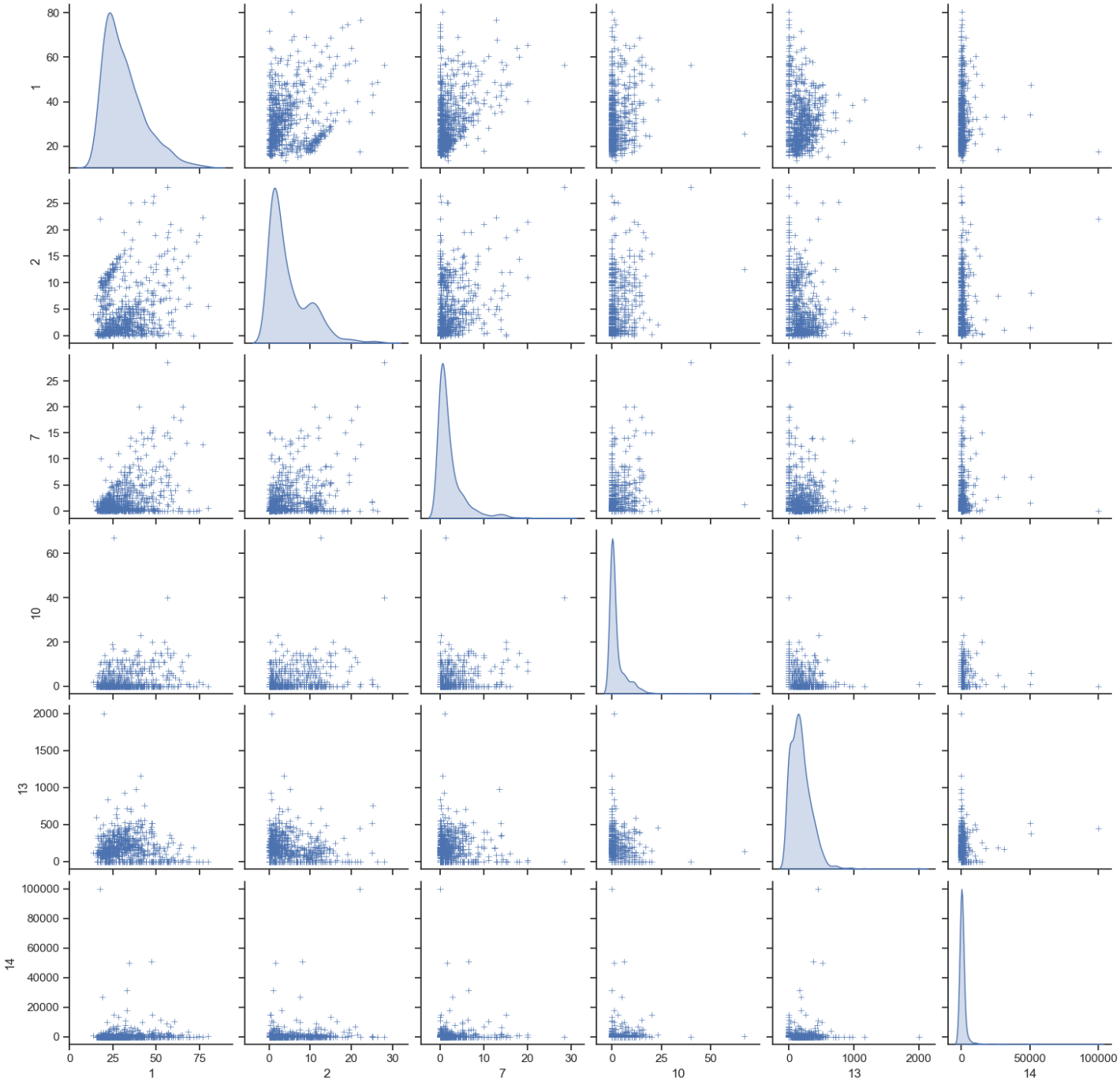
Celem zadania jest przeanalizowanie technik wstępnego przetwarzania danych i ich wpływu na wyniki klasyfikacji. Pracować będziemy na zbiorze *Credit approval dataset1*. Jako klasyfikatorów użyj metod: regresji logistycznej, naiwnego Bayesa (NB), najbliższych sąsiadów (klasyfikatora k-NN), metody wektorów nośnych (Support Vector Machines, SVM) oraz lasów losowych (ang. random forests) z ich domyślnymi parametrami.

2. Implementacja.

2.1 Uzupełnianie brakujących wartości

- **Dla danych numerycznych:** najpierw wybieramy kolumny zawierające dane numeryczne, następnie przekształcamy te kolumny do typu numerycznego, co jest ważne, ponieważ czasami dane mogą być przechowywane jako ciągi znaków, a nie liczby. Aby uzupełnić brakujące wartości w tych kolumnach numerycznych, używamy średniej wartości dla każdej kolumny.
- **Dla danych nominalnych:** Wybieramy kolumny zawierające dane nominalne, następnie uzupełniamy brakujące wartości w tych kolumnach najczęstszą wartością (modą).

2.2 Generowanie macierzy rozrzutu



Ryc. 1

2.3 Kodowanie wartości nominalnych

- Zakodowano wartości nominalne za pomocą *label encoding* oraz *one hot encoding*. Pierwsza reprezentacja cech jest dopuszczalna dla każdego z klasyfikatorów, druga natomiast nie jest akceptowana tylko przez naiwnego Bayesa. Wyniki dla zakodowanych danych wyglądają następująco:

```
Wyniki dla danych zakodowanych za pomocą LabelEncoder:
Logistic Regression: 0.8406
Naive Bayes: 0.7391
k-NN: 0.6377
SVM: 0.5870
Random Forest: 0.8841

Wyniki dla danych zakodowanych za pomocą OneHotEncoder:
Logistic Regression: 0.8188
k-NN: 0.7971
SVM: 0.8261
Random Forest: 0.8188
```

Ryc. 2

Dane reprezentowane za pomocą *one hot encoding* wydają się działać lepiej dla większości modeli, zwłaszcza dla *Logistic Regression*, *k-NN* i *SVM*. Jednakże, warto zwrócić uwagę na to, że *Random Forest* osiąga wysokie wyniki dla obu metod kodowania.

2.4 Skalowanie cech

- Przeprowadzono skalowanie cech za pomocą dwóch wzorów:
  - *normalizacji (min-max scaling*)

$$x \leftarrow \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

  - *standaryzacji (standarization)*

$$x \leftarrow \frac{x - \mu}{\sigma}$$

- Zbadano następnie, jak skalowanie cech wpływa na dokładność klasyfikatorów *k-NN* oraz *lasów losowych*. Wyniki zostały przedstawione jako dokładność razem z przedziałem ufności. Ponieważ zbiór danych jest nieduży, pomiary wykonaj przy pomocy 5- krotnej walidacji krzyżowej. Wyniki przedstawiono na rycinie 3.

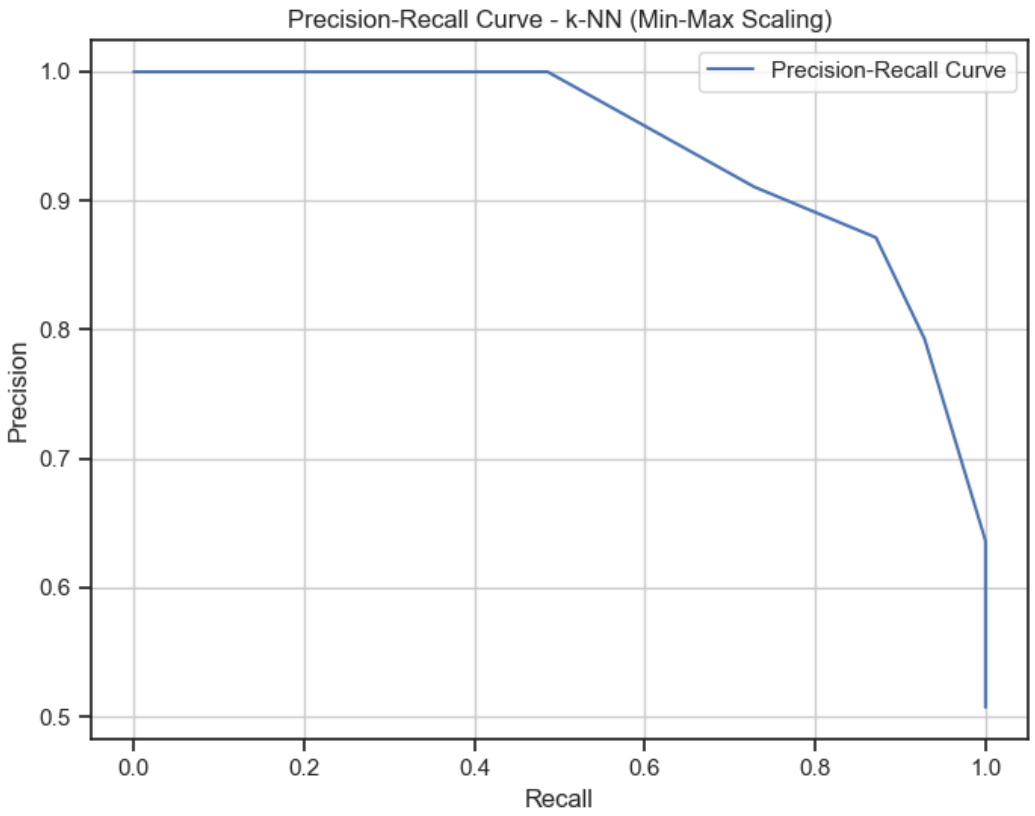
```
Dokładność k-NN po Min-Max Scaling: 0.8497 (+/- 0.0398)
Dokładność RandomForest po Min-Max Scaling: 0.8768 (+/- 0.0481)
Dokładność k-NN po Standaryzacji: 0.8406 (+/- 0.0251)
Dokładność RandomForest po Standaryzacji: 0.8623 (+/- 0.0395)
```

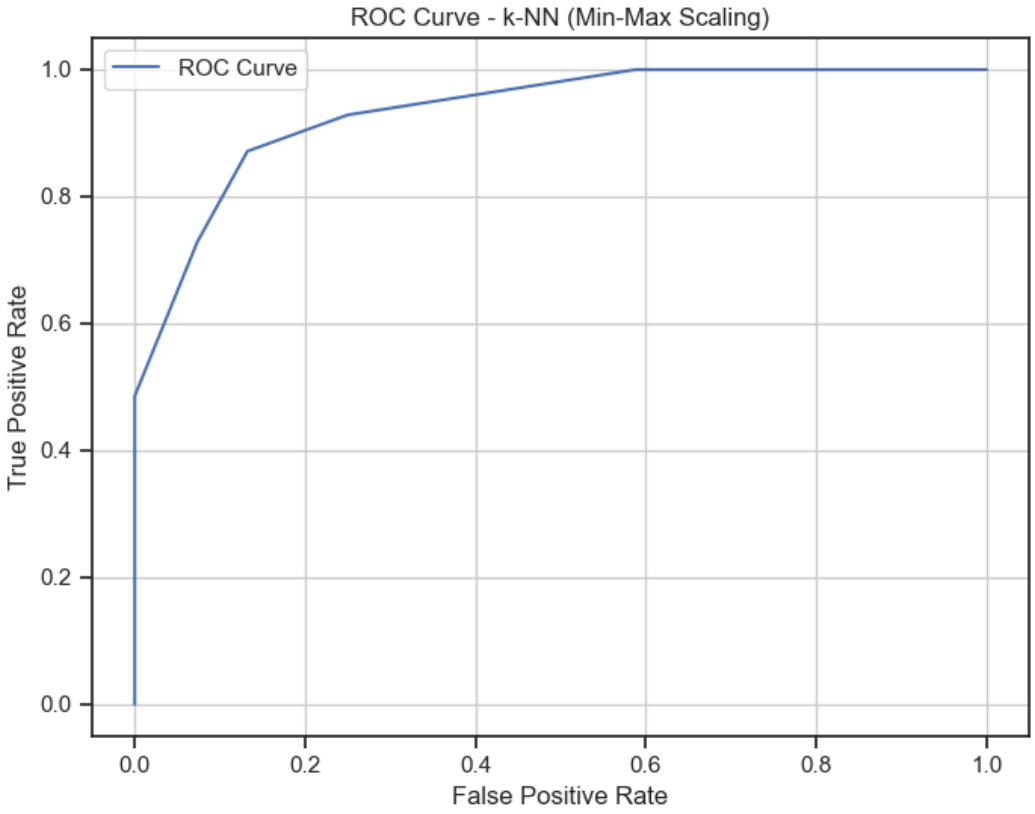
Ryc. 3

- Zarówno *Min-Max Scaling*, jak i *Standaryzacja* poprawiły wyniki dla obu modeli: *k-NN i RandomForest*. *Min-Max Scaling* wydaje się lekko lepszy dla *RandomForest*, podczas gdy *Standaryzacja* dla *k-NN*. Odchylenia standardowe wskazują na stabilność modeli, a różnice między wynikami nie są znaczące.

2.5 Wykres precyzji w funkcji pełności oraz charakterystyki roboczej odbiornika

- Dla klasyfikatora *k-NN* wygenerowano wykres precyzji w funkcji pełności (ang. precision-recall curve) oraz wykres charakterystyki roboczej odbiornika (ang. receiver operating characteristic, ROC).





3. Podsumowanie i wnioski.

Uzupełnianie brakujących danych oraz kodowanie wartości nominalnych mają istotny wpływ na jakość modeli klasyfikacyjnych. Wstępne przetwarzanie danych, takie jak skalowanie cech, może poprawić skuteczność klasyfikatorów. Analiza precision-recall curve oraz ROC pomaga w wyborze optymalnego progu decyzyjnego dla klasyfikatorów. Optymalizacja procesów wstępnego przetwarzania danych może znacząco zwiększyć skuteczność modeli klasyfikacyjnych.