Overfitting - regularyzacja

Regularyzacja L1 (Lasso) i L2 (Ridge) dla Regresji Liniowej

W tym ćwiczeniu Państwa zadaniem będzie użycie implementacji regresji liniowej z regularyzacją L1 i L2 z biblioteki scikit-learn na zbiorze Boston Housing.

Zbiór przedstawia ceny domów oraz ich cechy charakterystyczne, które w założeniu w jakiś sposób korelują z wartością.

Zbiór danych jest dostępny online oraz w bibliotece scikit-learn.

```
import sklearn
from sklearn.datasets import load_boston
boston = load_boston()
```

Wczytane dane mają formę słownika. Proszę zapoznać się z danymi, wykorzystując m. in. poniższe polecenia.

```
print(boston.keys())
print(boston.data.shape)
print(boston.DESCR)
```

Proszę przypisać dane do następujących zmiennych:

```
X = boston.data
y = boston.target
```

Uwaga. Dla scikit-learn w wersji 1.2+, baza boston nie jest już dostępna. Alternatywnie, można ją pobrać i wczytać korzystając z kodu poniżej.

```
import pandas as pd
import numpy as np

data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]

X = data
y = target
```

1. Podział zbioru na uczący i testowy
Standardową procedurą w uczeniu maszynowym jest podział zbioru na część użytą do trenowania modelu, oraz do testów. Na części testowej model jest poddawany ewaluacji co pozwala w obiektywny sposób określić jego skuteczność (zbiór testowy nie bierze udziału w uczeniu).

W celu przygotowania danych proszę wykonać następujące kroki:

- Użyć funkcji `train_test_split` z modułu scikit-learn: `sklearn.model_selection import train_test_split` aby pomieszać i podzielić przykłady na zbiór treningowy i testowy.
- Podzielić dane w proporcji: 80% - training set; 20% test set.
- Ustawić argument random_state funkcji train_test_split na stałą wartość. Pozwoli to na odtworzenie eksperymentu dla tego samego podziału za każdym uruchomieniem skryptu (ustawienie random_state powoduje, że “losowe” mieszanie przykładów za każdym razem da ten sam wynik).
- Przypisać podzbiory do zmiennych: X_train, X_test, y_train i y_test.

W celu weryfikacji podziału można wykonać:

```
print(X.shape[0])
print(float(X_train.shape[0]) / float(X.shape[0]))
print(float(X_test.shape[0]) / float(X.shape[0]))
```

Proszę dokonać normalizacji danych z wykorzystaniem klasy `StandardScaler()`

2. Regresja liniowa

- Proszę zaimportować moduł regresji liniowej:

```
from sklearn.linear_model import LinearRegression
```

- Z wykorzystaniem metody fit dopasować model do danych uczących.
- Dokonać predykcji zbioru testowego z wykorzystaniem wytrenowanego modelu.
- Aby zwizualizować różnicę między predykcją a realnymi wartościami można przedstawić na płaszczyźnie uzyskane wyniki (w celu porównania, dodatkowo można dokonać predykcji na zbiorze uczącym):

```
plt.scatter(y_train, y_train_pred)
plt.scatter(y_test, y_pred)
plt.xlabel("Prices: $Y_i$")
plt.ylabel("Predicted prices: $\hat{Y}_i$")
plt.title("Prices vs Predicted prices: $Y_i$ vs $\hat{Y}_i$")
```

Jak powinien wyglądać wykres idealnie dopasowanych danych?
Jak będzie wyglądał wykres w przypadku overfittingu?
3. Proszę policzyć Mean Squared Error dla predykcji na zbiorze testowym.
4. Proszę zweryfikować `model.score` (Opis metryki dostępny jest w dokumentacji).
Powyższe metryki będą przydatne do porównania tego modelu z kolejnymi.

Aby poprawić wyniki, można spróbować dodać nieliniowość do cech ze zbioru. Pomocna w tym będzie klasa PolynomialFeatures. Proszę zacząć od stopnia 2 wielomianu:

```
from sklearn.preprocessing import PolynomialFeatures
```

```
polynomial_features = PolynomialFeatures(degree=2)
```

- Z wykorzystaniem `polynomial_features.fit_transform` dodać nieliniowe cechy do zbiorów uczącego i testowego.
- Utworzyć nowy model regresji liniowej i nauczyć go z wykorzystaniem zmodyfikowanego zbioru danych.
- Policzyć metryki MSE oraz model.score. Porównać z poprzednimi.

Dodanie nieliniowości może spowodować overfitting (mały MSE na zbiorze uczącym, wysoki na zbiorze testowym).

W związku z tym, należy zastosować regularyzację:

- W pierwszej kolejności należy zaimportować implementację regresji liniowej z regularyzacją L1 i L2 `from sklearn.linear_model import Ridge, Lasso` More...

5. Lasso
Proszę wytrenować model manipulując parametrem α w zakresie 0.001 - 10 (dla kilku wartości).
Proszę wyrysować wykres zależności MSE od α oraz model.score od α

6. Ridge
Proszę wytrenować model manipulując parametrem α w zakresie 0.001 - 10 (dla kilku wartości).
Proszę wyrysować wykres zależności MSE od α oraz model.score od α

Regularyzacja dla Regresji Logistycznej

Proszę wykonać jedno z dwóch poniższych ćwiczeń (A lub B):

Ćwiczenie ze zbiorem A:
Proszę wczytać zbiór danych z wykorzystaniem funkcji w scikit-learn

```
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()

y = data.target
X = data.data
```

Dodatkowe informacje o użytych danych: Breast Cancer Wisconsin
1. Proszę podzielić zbiór na część treningową i część testową.
2. Z wykorzystaniem Regresji Logistycznej z biblioteki scikit-learn, proszę wytrenować model z regularyzacją L2 dla kilku wartości alfa z przedziału [0.0001; 1]. Każdy model należy sprawdzić na zbiorze testowym licząc jego skuteczność (accuracy)
3. Proszę wyrysować wykres zależności alfa od accuracy dla każdego z modeli

Dataset B:

```
path = 'https://uu-sml.github.io/course-sml-public/data/biopsy.csv'
dataset = pd.read_csv(path, na_values='?', dtype={'ID': str})
dataset.columns = ['ID', 'Clump Thickness', 'Uniformity of Cell Size',
```

```
'Uniformity of Cell Shape', 'Marginal Adhesion', 'Single Epithelial Cell Size', 'Bare Nuclei', 'Bland Chromatin', 'Normal Nucleoli', 'Mitoses', 'Class']
```

W tym zbiorze danych obecne są następujące kolumny:
ID: sample code number (not unique).
V1: clump thickness.
V2: uniformity of cell size.
V3: uniformity of cell shape.
V4: marginal adhesion.
V5: single epithelial cell size.
V6: bare nuclei (16 values are missing).
V7: bland chromatin.
V8: normal nucleoli.
V9: mitoses.
class: “benign” or “malignant”.

1. Proszę zastąpić wartości nan średnią wartością dla danej cechy (np. z wykorzystaniem funkcji pandas.isnull())
2. Proszę podzielić zbiór na część treningową i testową. Proszę pamiętać o usunięciu pierwszej kolumny która reprezentuje ID przykładu oraz na konwersji wektora wyjściowego (y) z wartości tekstowych na odpowiadające wartości 0 lub 1.
3. Z wykorzystaniem implementacji regresji logistycznej z regularyzacją L2 z scikit-learn proszę wytrenować kilka modeli z parametrem alfa z przedziału [0.0001; 10]. Dla każdego z modeli proszę policzyć accuracy na zbiorze testowym.
4: Proszę wyrysować wykres alfa od accuracy dla każdego wytrenowanego modelu.