Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

Лабораторная работа №2

по дисциплине

«Методы машинного обучения»

на тему

«Обработка признаков часть 1»

Выполнил:

студент группы ИУ5И-23М

Цзян Юхуэй

Москва-2024 г.

Цель лабораторной работы:

Изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

Задание:

1. Выбрать набор данных (датасет), содержащий категориальные и числовые признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.) Просьба не использовать датасет, на котором данная задача решалась в лекции.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
    i. устранение пропусков в данных;
    ii. кодирование категориальных признаков;
    iii. нормализация числовых признаков.

Часть 1. Для обработки пропусков

Для обработки пропусков мы используем набор данных UCI "Болезни сердца".

```
import pandas as pd

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OneHotEncoder, StandardScaler, LabelEncoder

# 加载数据集

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data"

column_names = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target']

data = pd.read_csv(url, header=None, names=column_names, na_values='?')

# 显示数据集的前几行

print(data.head())
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	

	slope	ca	thal	target
0	3.0	0.0	6.0	0
1	2.0	3.0	3.0	2
2	2.0	2.0	7.0	1
3	3.0	0.0	3.0	0
4	1.0	0.0	3.0	0

Рис 1. Первые пять строк набора данных.

Случай пропуска значений в наборе данных：

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          4
thal        2
target      0
dtype: int64
```

Рис 2. Случай пропуска значений в наборе данных.

Среднее значение использовалось для восполнения недостающих значений для числовых характеристик, а множественное число - для восполнения недостающих значений для категориальных характеристик:

```
# 使用平均值填补数值缺失值

imputer = SimpleImputer(strategy='mean')

data[['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'ca']] = imputer.fit_transform(data[['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'ca']])

# 使用众数填补分类特征缺失值

data['thal'] = data['thal'].fillna(data['thal'].mode()[0])

# 查看数据集的缺失值情况

print(data.isnull().sum())
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

Рис 3. Появление недостающих значений в наборе данных после заполнения.

Часть 2. Для категориальных признаков

Для категориальных признаков мы используем библиотеку Scikit-Learn для загрузки набора данных Iris.

```
import pandas as pd

from sklearn.preprocessing import OneHotEncoder, StandardScaler

from sklearn.datasets import load_iris

# 加载数据集

iris = load_iris()

data = pd.DataFrame(data=iris.data, columns=iris.feature_names)

data['target'] = iris.target

# 显示数据集的前几行

print(data.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

   target
0       0
1       0
2       0
3       0
4       0
```

Рис 4. Первые пять строк набора данных.

Классификационные признаки кодируются с помощью метода One-Hot Encoding:

```
# 编码分类特征

# One-Hot Encoding

one_hot_encoder = OneHotEncoder(sparse=False)

encoded_features = one_hot_encoder.fit_transform(data[['target']])

encoded_df = pd.DataFrame(encoded_features, columns=one_hot_encoder.get_feature_names_out(['target']))

data = pd.concat([data, encoded_df], axis=1).drop(['target'], axis=1)

# 标准化数值特征
```

```
scaler = StandardScaler()

data[data.columns[:-3]] = scaler.fit_transform(data[data.columns[:-3]])

# 显示处理后的数据集的前几行

print(data.head())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0          -0.900681          1.019004          -1.340227         -1.315444
1          -1.143017         -0.131979          -1.340227         -1.315444
2          -1.385353          0.328414          -1.397064         -1.315444
3          -1.506521          0.098217          -1.283389         -1.315444
4          -1.021849          1.249201          -1.340227         -1.315444

   target_0  target_1  target_2
0       1.0       0.0       0.0
1       1.0       0.0       0.0
2       1.0       0.0       0.0
3       1.0       0.0       0.0
4       1.0       0.0       0.0
```

Рис 5. Первые пять строк набора данных после процесса классификации.