****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 2)»

Выполнил:  
студент группы ИУ5-23М  
Чжэн Сяохуэй

Москва — 2024 г.

**1. Цель лабораторной работы**

изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

**2. Задание**

1. Выбрать один или несколько наборов данных (датасетов) для решения следующих задач. Каждая задача может быть решена на отдельном датасете, или несколько задач могут быть решены на одном датасете. Просьба не использовать датасет, на котором данная задача решалась в лекции.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
   1. масштабирование признаков (не менее чем тремя способами);
   2. обработку выбросов для числовых признаков (по одному способу для удаления выбросов и для замены выбросов);
   3. обработку по крайней мере одного нестандартного признака (который не является числовым или категориальным);
   4. отбор признаков:
      * один метод из группы методов фильтрации (filter methods);
      * один метод из группы методов обертывания (wrapper methods);
      * один метод из группы методов вложений (embedded methods).

**3. текст программы**

**import** pandas **as** pd

**import** numpy **as** np

**from** sklearn.preprocessing **import** StandardScaler, MinMaxScaler, Normalizer, LabelEncoder

**from** sklearn.ensemble **import** IsolationForest

**from** sklearn.impute **import** SimpleImputer

**from** sklearn.feature\_selection **import** VarianceThreshold, RFE

**from** sklearn.linear\_model **import** Lasso

**from** sklearn.tree **import** DecisionTreeClassifier

*# 读取数据*

*# 加载数据集*

url **=** "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

data **=** pd**.**read\_csv(url)

*# 预处理数据*

data**.**drop(['Name', 'Ticket', 'Cabin'], axis**=**1, inplace**=True**)

data['Sex'] **=** LabelEncoder()**.**fit\_transform(data['Sex'])

data['Embarked']**.**fillna('S', inplace**=True**)

data['Embarked'] **=** LabelEncoder()**.**fit\_transform(data['Embarked'])

*# 填充缺失值*

imputer **=** SimpleImputer(strategy**=**'median')

data[['Age', 'Fare']] **=** imputer**.**fit\_transform(data[['Age', 'Fare']])

*# 特征缩放*

scaler\_standard **=** StandardScaler()

scaler\_minmax **=** MinMaxScaler()

scaler\_normalize **=** Normalizer()

data\_standard **=** data**.**copy()

data\_minmax **=** data**.**copy()

data\_normalize **=** data**.**copy()

data\_standard[['Age', 'Fare']] **=** scaler\_standard**.**fit\_transform(data\_standard[['Age', 'Fare']])

data\_minmax[['Age', 'Fare']] **=** scaler\_minmax**.**fit\_transform(data\_minmax[['Age', 'Fare']])

data\_normalize[['Age', 'Fare']] **=** scaler\_normalize**.**fit\_transform(data\_normalize[['Age', 'Fare']])

*# 异常值处理*

iso\_forest **=** IsolationForest(contamination**=**0.1)

outliers **=** iso\_forest**.**fit\_predict(data[['Age', 'Fare']])

data\_no\_outliers **=** data[outliers **!=** **-**1]

*# 替换异常值*

data\_imputed **=** data**.**copy()

data\_imputed[['Age', 'Fare']] **=** SimpleImputer(strategy**=**'median')**.**fit\_transform(data\_imputed[['Age', 'Fare']])

*# 处理非标准特征（已在预处理步骤中编码'Sex'和'Embarked'）*

*# 特征选择*

X **=** data**.**drop('Survived', axis**=**1)

y **=** data['Survived']

*# 过滤方法*

selector\_var **=** VarianceThreshold(threshold**=**0.1)

X\_filtered **=** selector\_var**.**fit\_transform(X)

*# 包裹方法*

selector\_rfe **=** RFE(estimator**=**DecisionTreeClassifier(), n\_features\_to\_select**=**5)

X\_wrapped **=** selector\_rfe**.**fit\_transform(X, y)

*# 嵌入方法*

model\_lasso **=** Lasso(alpha**=**0.1)

model\_lasso**.**fit(X, y)

importance **=** np**.**abs(model\_lasso**.**coef\_)

selected\_features **=** X**.**columns[np**.**argsort(importance)[**-**5:]]

*# 输出结果*

print("Standard Scaled Data:\n", data\_standard**.**head())

print("Min-Max Scaled Data:\n", data\_minmax**.**head())

print("Normalized Data:\n", data\_normalize**.**head())

print("Data without Outliers:\n", data\_no\_outliers**.**head())

print("Data with Imputed Values:\n", data\_imputed**.**head())

print("Filtered Data Shape:\n", X\_filtered**.**shape)

print("Wrapped Data Shape:\n", X\_wrapped**.**shape)

print("Selected Features by Lasso:\n", selected\_features)

4. экранные формы с примерами выполнения программы.

Standard Scaled Data:

PassengerId Survived Pclass Sex Age SibSp Parch Fare \

0 1 0 3 1 -0.565736 1 0 -0.502445

1 2 1 1 0 0.663861 1 0 0.786845

2 3 1 3 0 -0.258337 0 0 -0.488854

3 4 1 1 0 0.433312 1 0 0.420730

4 5 0 3 1 0.433312 0 0 -0.486337

Embarked

0 2

1 0

2 2

3 2

4 2

Min-Max Scaled Data:

PassengerId Survived Pclass Sex Age SibSp Parch Fare \

0 1 0 3 1 0.271174 1 0 0.014151

1 2 1 1 0 0.472229 1 0 0.139136

2 3 1 3 0 0.321438 0 0 0.015469

3 4 1 1 0 0.434531 1 0 0.103644

4 5 0 3 1 0.434531 0 0 0.015713

Embarked

0 2

1 0

2 2

3 2

4 2

Normalized Data:

PassengerId Survived Pclass Sex Age SibSp Parch Fare \

0 1 0 3 1 0.949757 1 0 0.312988

1 2 1 1 0 0.470417 1 0 0.882444

2 3 1 3 0 0.956551 0 0 0.291564

3 4 1 1 0 0.550338 1 0 0.834942

4 5 0 3 1 0.974555 0 0 0.224148

Embarked

0 2

1 0

2 2

3 2

4 2

Data without Outliers:

PassengerId Survived Pclass Sex Age SibSp Parch Fare Embarked

0 1 0 3 1 22.0 1 0 7.2500 2

1 2 1 1 0 38.0 1 0 71.2833 0

2 3 1 3 0 26.0 0 0 7.9250 2

3 4 1 1 0 35.0 1 0 53.1000 2

4 5 0 3 1 35.0 0 0 8.0500 2

Data with Imputed Values:

PassengerId Survived Pclass Sex Age SibSp Parch Fare Embarked

0 1 0 3 1 22.0 1 0 7.2500 2

1 2 1 1 0 38.0 1 0 71.2833 0

2 3 1 3 0 26.0 0 0 7.9250 2

3 4 1 1 0 35.0 1 0 53.1000 2

4 5 0 3 1 35.0 0 0 8.0500 2

Filtered Data Shape:

(891, 8)

Wrapped Data Shape:

(891, 5)

Selected Features by Lasso:

Index(['Embarked', 'PassengerId', 'Fare', 'Age', 'Sex'], dtype='object')

**Список литературы**

[1] Гапанюк Ю. Е. COURSE\_MMO\_SPRING\_2024// GitHub. –– 2024. –– Режим доступа: https://github.com/ugapanyuk/courses\_current/wiki/COURSE\_MMO\_SPRING\_2024

[2] <https://www.kaggle.com/datasets>