Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №3
по дисциплине
«Методы машинного обучения»
на тему

«Обработка признаков, часть 2.»

Выполнил:
студент группы ИУ5-22М
Лун Сыхань

Москва — 2024г.

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. Ход выполнения работы

В этой тетради я буду использовать графики для визуализации взаимосвязи между переменными в наборе данных "Титаник".

The dataset includes the following columns:

Survived
Pclass
Name
Sex
Age
Siblings/Spouses Aboard
Parents/Children Aboard
Fare

```
!pip install scikit-learn pandas numpy
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler
from sklearn.ensemble import IsolationForest
from sklearn.impute import SimpleImputer
from sklearn.feature_selection import SelectKBest, chi2, RFE
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
# 加载数据集
url = 'https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv'
df = pd.read_csv(url)
df.head()
```

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
0	0	3	Mr. Owen Harris Braund	male	22.0	1	0	7.2500
1	1	1	Mrs. John Bradley (Florence Briggs Thayer) Cum...	female	38.0	1	0	71.2833
2	1	3	Miss. Laina Heikkinen	female	26.0	0	0	7.9250
3	1	1	Mrs. Jacques Heath (Lily May Peel) Futrelle	female	35.0	1	0	53.1000
4	0	3	Mr. William Henry Allen	male	35.0	0	0	8.0500

масштабирование признаков (не менее чем тремя способами):

```
# 选择数值特征
num_features = df.select_dtypes(include=['int64', 'float64']).columns

# 1. 标准化（StandardScaler）
standard_scaler = StandardScaler()
df_standard_scaled = df.copy()
df_standard_scaled[num_features] = standard_scaler.fit_transform(df[num_features])

# 2. 最小-最大缩放（MinMaxScaler）
min_max_scaler = MinMaxScaler()
df_min_max_scaled = df.copy()
df_min_max_scaled[num_features] = min_max_scaler.fit_transform(df[num_features])

# 3. 稳健缩放（RobustScaler）
robust_scaler = RobustScaler()
df_robust_scaled = df.copy()
df_robust_scaled[num_features] = robust_scaler.fit_transform(df[num_features])
# 打印结果
df_standard_scaled.head(), df_min_max_scaled.head(), df_robust_scaled.head()
```

```
(   Survived    Pclass                                               Name  \
 0 -0.792163  0.830524                            Mr. Owen Harris Braund
 1  1.262366 -1.561277  Mrs. John Bradley (Florence Briggs Thayer) Cum...
 2  1.262366  0.830524                             Miss. Laina Heikkinen
 3  1.262366 -1.561277       Mrs. Jacques Heath (Lily May Peel) Futrelle
 4 -0.792163  0.830524                           Mr. William Henry Allen

      Sex       Age  Siblings/Spouses Aboard  Parents/Children Aboard  \
 0   male -0.529366                 0.429904                -0.474981
 1 female  0.604265                 0.429904                -0.474981
 2 female -0.245958                -0.475856                -0.474981
 3 female  0.391709                 0.429904                -0.474981
 4   male  0.391709                -0.475856                -0.474981

       Fare
 0 -0.503586
 1  0.783412
 2 -0.490020
 3  0.417948
 4 -0.487507 ,
    Survived  Pclass                                               Name  \
 0       0.0     1.0                            Mr. Owen Harris Braund
 1       1.0     0.0  Mrs. John Bradley (Florence Briggs Thayer) Cum...
 2       1.0     1.0                             Miss. Laina Heikkinen
 3       1.0     0.0       Mrs. Jacques Heath (Lily May Peel) Futrelle
 4       0.0     1.0                           Mr. William Henry Allen

      Sex       Age  Siblings/Spouses Aboard  Parents/Children Aboard  \
 0   male  0.271174                    0.125                      0.0
 1 female  0.472229                    0.125                      0.0
 2 female  0.321438                    0.000                      0.0
 3 female  0.434531                    0.125                      0.0
 4   male  0.434531                    0.000                      0.0

       Fare
 0  0.014151
 1  0.139136
 2  0.015469
 3  0.103644
 4  0.015713 ,
    Survived  Pclass                                               Name  \
 0       0.0     0.0                            Mr. Owen Harris Braund
 1       1.0    -2.0  Mrs. John Bradley (Florence Briggs Thayer) Cum...
 2       1.0     0.0                             Miss. Laina Heikkinen
 3       1.0    -2.0       Mrs. Jacques Heath (Lily May Peel) Futrelle
 4       0.0     0.0                           Mr. William Henry Allen

      Sex       Age  Siblings/Spouses Aboard  Parents/Children Aboard  \
 0   male -0.338028                      1.0                      0.0
 1 female  0.563380                      1.0                      0.0
 2 female -0.112676                      0.0                      0.0
 3 female  0.394366                      1.0                      0.0
 4   male  0.394366                      0.0                      0.0

       Fare
 0 -0.310359
 1  2.448211
 2 -0.281279
```

обработку выбросов для числовых признаков (по одному способу для удаления выбросов и для замены выбросов):

```
# 选择数值特征
num_features = df.select_dtypes(include=['int64', 'float64']).columns

# 使用Isolation Forest检测并删除异常值
iso = IsolationForest(contamination=0.1)
yhat = iso.fit_predict(df[num_features])
mask = yhat != -1
df_no_outliers = df[mask]

# 打印处理后的结果
print("Original shape:", df.shape)
print("New shape after removing outliers:", df_no_outliers.shape)

# 使用IQR替换异常值
Q1 = df[num_features].quantile(0.25)
Q3 = df[num_features].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df_iqr_replaced = df.copy()
df_iqr_replaced[num_features] = np.where((df[num_features] < lower_bound) | (df[num_features] > upper_bound), np.nan, df[num_features])

# 使用中位数填充异常值
imputer = SimpleImputer(strategy='median')
df_iqr_replaced[num_features] = imputer.fit_transform(df_iqr_replaced[num_features])

# 打印处理后的结果
df_iqr_replaced.head()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but IsolationForest was fitted with feature names
  warnings.warn(
Original shape: (887, 8)
New shape after removing outliers: (798, 8)
```

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
0	0.0	3.0	Mr. Owen Harris Braund	male	22.0	1.0	0.0	7.250
1	1.0	1.0	Mrs. John Bradley (Florence Briggs Thayer) Cum...	female	38.0	1.0	0.0	13.000
2	1.0	3.0	Miss. Laina Heikkinen	female	26.0	0.0	0.0	7.925
3	1.0	1.0	Mrs. Jacques Heath (Lily May Peel) Futrelle	female	35.0	1.0	0.0	53.100
4	0.0	3.0	Mr. William Henry Allen	male	35.0	0.0	0.0	8.050

обработку по крайней мере одного нестандартного признака (который не является числовым или категориальным)

```
# 处理一个日期时间特征（假设我们有一个日期时间特征）
df['Date'] = pd.to_datetime('2022-01-01') + pd.to_timedelta(np.arange(len(df)), 'D')

# 提取日期时间特征
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
df['Day'] = df['Date'].dt.day
df['DayOfWeek'] = df['Date'].dt.dayofweek

# 删除原日期时间特征
df = df.drop(columns=['Date'])

# 打印结果
df.head()
```

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare	Year	Month	Day	DayOfWeek
0	0	3	Mr. Owen Harris Braund	male	22.0	1	0	7.2500	2022	1	1	5
1	1	1	Mrs. John Bradley (Florence Briggs Thayer) Cum...	female	38.0	1	0	71.2833	2022	1	2	6
2	1	3	Miss. Laina Heikkinen	female	26.0	0	0	7.9250	2022	1	3	0
3	1	1	Mrs. Jacques Heath (Lily May Peel) Futrelle	female	35.0	1	0	53.1000	2022	1	4	1
4	0	3	Mr. William Henry Allen	male	35.0	0	0	8.0500	2022	1	5	2

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).

```
# 填充缺失值，以便进行特征选择
df = df.drop(columns=['Name'])  # 删除非数值和非类别型特征
cat_features = df.select_dtypes(include=['object']).columns
imputer = SimpleImputer(strategy='most_frequent')
df[cat_features] = imputer.fit_transform(df[cat_features])

# 特征编码
df_encoded = pd.get_dummies(df, drop_first=True)

# 1. 过滤方法：使用卡方检验
X = df_encoded.drop('Survived', axis=1)
y = df_encoded['Survived']
selector = SelectKBest(chi2, k=10)
X_new = selector.fit_transform(X, y)
selected_features = X.columns[selector.get_support()]
print("Selected features (Filter method):", selected_features)

# 2. 包装方法：使用递归特征消除（RFE）
model = LogisticRegression(solver='liblinear')
rfe = RFE(model, n_features_to_select=10)
fit = rfe.fit(X, y)
selected_features = X.columns[fit.support_]
print("Selected features (Wrapper method):", selected_features)

# 3. 嵌入方法：使用随机森林
model = RandomForestClassifier(n_estimators=100)
model.fit(X, y)
importances = model.feature_importances_
indices = np.argsort(importances)[-10:]
selected_features = X.columns[indices]
print("Selected features (Embedded method):", selected_features)
```

```
Selected features (Filter method): Index(['Pclass', 'Age', 'Siblings/Spouses Aboard', 'Parents/Children Aboard',
       'Fare', 'Year', 'Month', 'Day', 'DayOfWeek', 'Sex_male'],
      dtype='object')
Selected features (Wrapper method): Index(['Pclass', 'Age', 'Siblings/Spouses Aboard', 'Parents/Children Aboard',
       'Fare', 'Year', 'Month', 'Day', 'DayOfWeek', 'Sex_male'],
      dtype='object')
Selected features (Embedded method): Index(['Parents/Children Aboard', 'Year', 'Siblings/Spouses Aboard',
       'DayOfWeek', 'Month', 'Pclass', 'Day', 'Age', 'Fare', 'Sex_male'],
      dtype='object')
```