****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 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 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:
   1. устранение пропусков в данных;
   2. кодирование категориальных признаков;
   3. нормализация числовых признаков.

**3. текст программы**

4. экранные формы с примерами выполнения программы.

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

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.preprocessing **import** StandardScaler, LabelEncoder, OneHotEncoder

*# 加载数据集*

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

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

*# 初步探索数据集*

print(data**.**info())

print(data**.**head())

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null object

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

None

PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

Name Sex Age SibSp \

0 Braund, Mr. Owen Harris male 22.0 1

1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1

2 Heikkinen, Miss. Laina female 26.0 0

3 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1

4 Allen, Mr. William Henry male 35.0 0

Parch Ticket Fare Cabin Embarked

0 0 A/5 21171 7.2500 NaN S

1 0 PC 17599 71.2833 C85 C

2 0 STON/O2. 3101282 7.9250 NaN S

3 0 113803 53.1000 C123 S

4 0 373450 8.0500 NaN S

In [ ]:

*# 查看缺失值*

print(data**.**isnull()**.**sum())

*# 处理缺失值*

*# 使用中位数填补Age列的缺失值*

data['Age']**.**fillna(data['Age']**.**median(), inplace**=True**)

*# 使用众数填补Embarked列的缺失值*

data['Embarked']**.**fillna(data['Embarked']**.**mode()[0], inplace**=True**)

*# 处理后再次检查缺失值*

print(data**.**isnull()**.**sum())

PassengerId 0

Survived 0

Pclass 0

Name 0

Sex 0

Age 177

SibSp 0

Parch 0

Ticket 0

Fare 0

Cabin 687

Embarked 2

dtype: int64

PassengerId 0

Survived 0

Pclass 0

Name 0

Sex 0

Age 0

SibSp 0

Parch 0

Ticket 0

Fare 0

Cabin 687

Embarked 0

dtype: int64

/tmp/ipykernel\_16482/1764806774.py:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

data['Age'].fillna(data['Age'].median(), inplace=True)

/tmp/ipykernel\_16482/1764806774.py:9: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

In [ ]:

*# 标签编码Sex列*

label\_encoder **=** LabelEncoder()

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

*# 独热编码Embarked列*

onehot\_encoder **=** OneHotEncoder()

embarked\_encoded **=** onehot\_encoder**.**fit\_transform(data[['Embarked']])**.**toarray()

embarked\_df **=** pd**.**DataFrame(embarked\_encoded, columns**=**onehot\_encoder**.**get\_feature\_names\_out(['Embarked']))

*# 将独热编码结果拼接到原数据集中，并删除原Embarked列*

data **=** pd**.**concat([data, embarked\_df], axis**=**1)

data**.**drop('Embarked', axis**=**1, inplace**=True**)

print(data**.**head())

PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

Name Sex Age SibSp Parch \

0 Braund, Mr. Owen Harris 1 22.0 1 0

1 Cumings, Mrs. John Bradley (Florence Briggs Th... 0 38.0 1 0

2 Heikkinen, Miss. Laina 0 26.0 0 0

3 Futrelle, Mrs. Jacques Heath (Lily May Peel) 0 35.0 1 0

4 Allen, Mr. William Henry 1 35.0 0 0

Ticket Fare Cabin Embarked\_C Embarked\_Q Embarked\_S

0 A/5 21171 7.2500 NaN 0.0 0.0 1.0

1 PC 17599 71.2833 C85 1.0 0.0 0.0

2 STON/O2. 3101282 7.9250 NaN 0.0 0.0 1.0

3 113803 53.1000 C123 0.0 0.0 1.0

4 373450 8.0500 NaN 0.0 0.0 1.0

In [ ]:

*# 标准化Age和Fare列*

scaler **=** StandardScaler()

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

print(data**.**head())

PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

Name Sex Age SibSp \

0 Braund, Mr. Owen Harris 1 -0.565736 1

1 Cumings, Mrs. John Bradley (Florence Briggs Th... 0 0.663861 1

2 Heikkinen, Miss. Laina 0 -0.258337 0

3 Futrelle, Mrs. Jacques Heath (Lily May Peel) 0 0.433312 1

4 Allen, Mr. William Henry 1 0.433312 0

Parch Ticket Fare Cabin Embarked\_C Embarked\_Q Embarked\_S

0 0 A/5 21171 -0.502445 NaN 0.0 0.0 1.0

1 0 PC 17599 0.786845 C85 1.0 0.0 0.0

2 0 STON/O2. 3101282 -0.488854 NaN 0.0 0.0 1.0

3 0 113803 0.420730 C123 0.0 0.0 1.0

4 0 373450 -0.486337 NaN 0.0 0.0 1.0

**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>