****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 ИУ5-22М  
Джин Шуо

Москва — 2024 г.

**1. Цель лабораторной работы**

Цель лабораторной работы: изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

**2. Задание**

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.

Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:

устранение пропусков в данных;

кодирование категориальных признаков;

нормализация числовых признаков.

**3. текст программы**

import pandas as pd

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OneHotEncoder, StandardScaler

import urllib.request

url\_titanic = 'https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv'

urllib.request.urlretrieve(url\_titanic, 'titanic.csv')

data\_titanic = pd.read\_csv('titanic.csv')

print("泰坦尼克号数据集的列名：")

print(data\_titanic.columns)

print("原始泰坦尼克号数据：")

print(data\_titanic.head())

numeric\_imputer = SimpleImputer(strategy='mean')

data\_titanic[['Age', 'Fare']] = numeric\_imputer.fit\_transform(data\_titanic[['Age', 'Fare']])

print("处理后的泰坦尼克号数据：")

print(data\_titanic.head())

data\_titanic.to\_csv('processed\_titanic.csv', index=False)

url\_iris = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

urllib.request.urlretrieve(url\_iris, 'iris.csv')

column\_names = ['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'class']

data\_iris = pd.read\_csv('iris.csv', names=column\_names)

print("原始鸢尾花数据：")

print(data\_iris.head())

encoder = OneHotEncoder(sparse=False, handle\_unknown='ignore')

encoded\_categories = encoder.fit\_transform(data\_iris[['class']])

encoded\_df = pd.DataFrame(encoded\_categories, columns=encoder.get\_feature\_names\_out(['class']))

data\_iris = data\_iris.join(encoded\_df).drop(['class'], axis=1)

print("处理后的鸢尾花数据：")

print(data\_iris.head())

data\_iris.to\_csv('processed\_iris.csv', index=False)

url\_boston = 'https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data'

urllib.request.urlretrieve(url\_boston, 'boston.csv')

column\_names\_boston = [

'CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',

'PTRATIO', 'B', 'LSTAT', 'MEDV']

data\_boston = pd.read\_csv(url\_boston, delim\_whitespace=True, names=column\_names\_boston)

print("原始波士顿房价数据：")

print(data\_boston.head())

scaler = StandardScaler()

data\_boston\_scaled = pd.DataFrame(scaler.fit\_transform(data\_boston), columns=data\_boston.columns)

print("处理后的波士顿房价数据：")

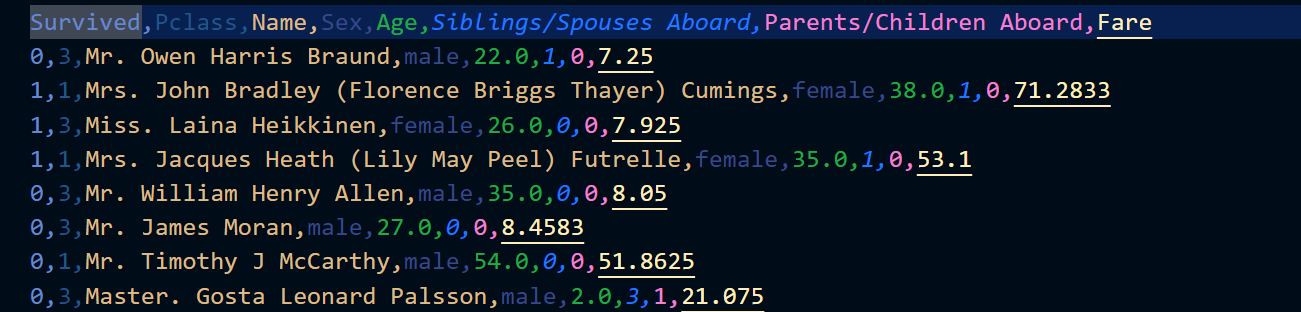
print(data\_boston\_scaled.head())

data\_boston\_scaled.to\_csv('processed\_boston.csv', index=False)

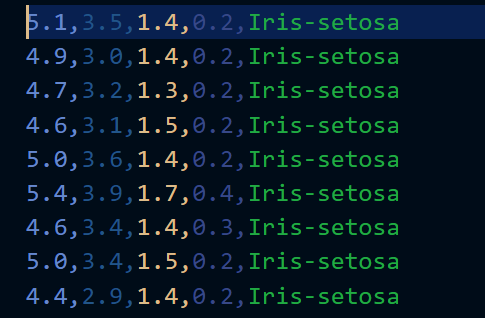
1. **экранные формы с примерами выполнения программы.**

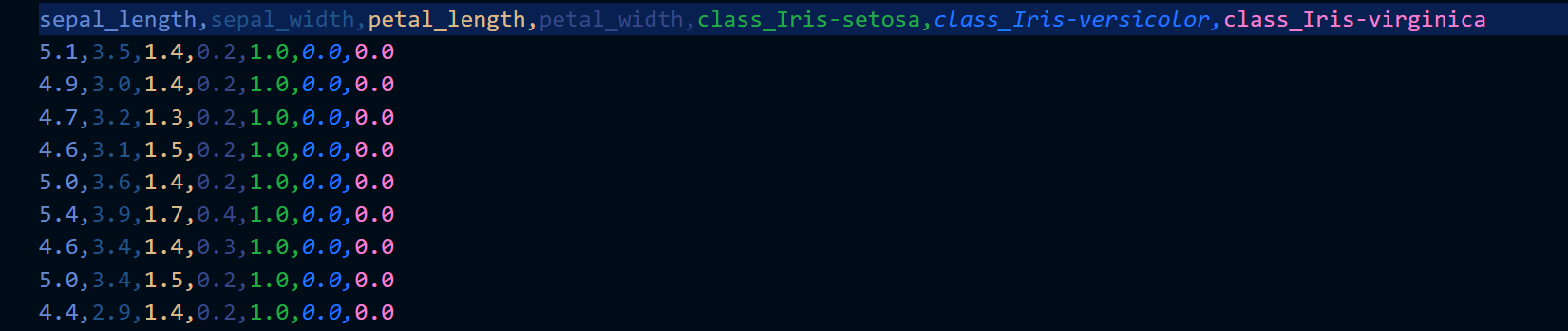
устранение пропусков в данных



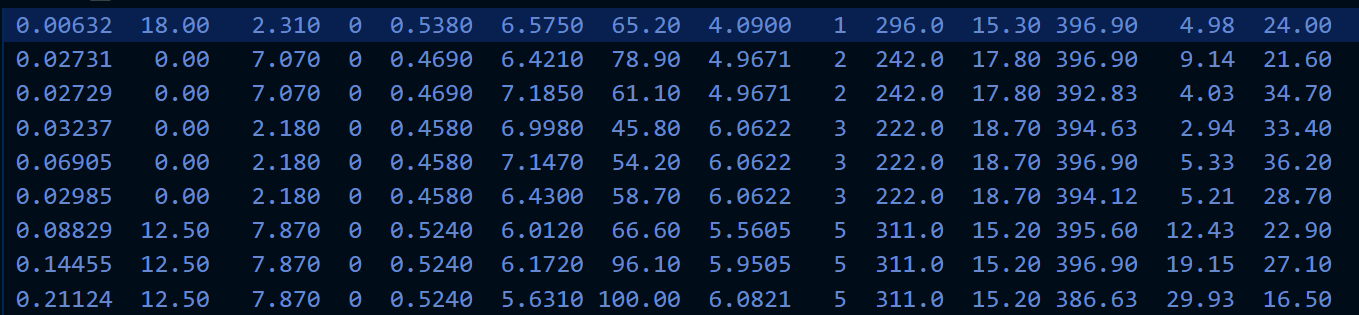


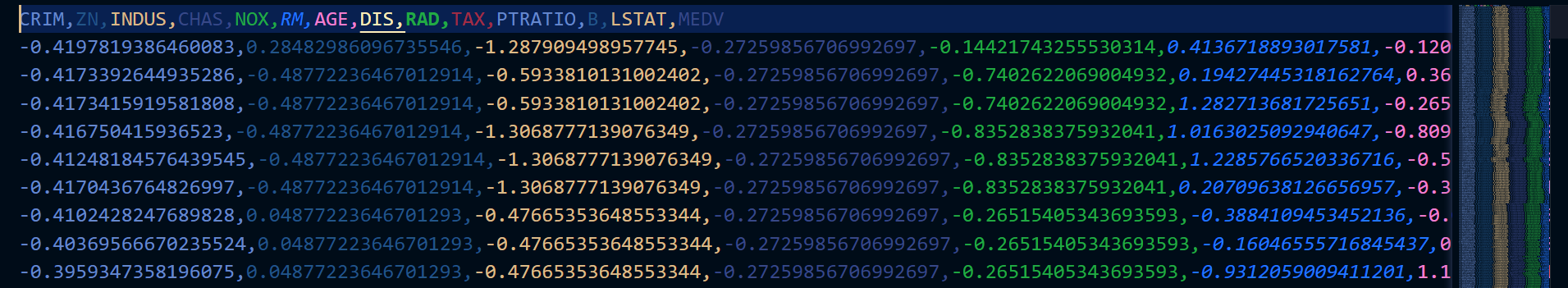
кодирование категориальных признаков





нормализация числовых признаков





**Список литературы**

[1] https://github.com/ugapanyuk/courses\_current/wiki/LAB\_MMO\_\_FEATURES

[2] <https://archive.ics.uci.edu/ml/machine-learning-databases>

[3]https://archive.ics.uci.edu/ml/machine-learning-databases