МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

домашнее задание №_____3 _____
по дисциплине «Методы машинного обучения»

Тема: «Обработка признаков (часть 2).»

ИСПОЛНИТЕЛЬ: Ли Яцзинь
ФИО
группа ИУ5И-21М
_____
подпись"___

ПРЕПОДАВАТЕЛЬ: Гапанюк Ю .Е

Москва - 2024

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:
    i. масштабирование признаков (не менее чем тремя способами);
    ii. обработку выбросов для числовых признаков (по одному способу для удаления выбросов и для замены выбросов);
    iii. обработку по крайней мере одного нестандартного признака (который не является числовым или категориальным);
    iv. отбор признаков:
        1. один метод из группы методов фильтрации (filter methods);
        2. один метод из группы методов обертывания (wrapper methods);
        3. один метод из группы методов вложений (embedded methods)

масштабирование признаков

Прочитайте набор данных и получите соответствующую информацию о наборе данных:

```
import pandas as pd

# 读取数据集并查看前几行信息
df = pd.read_csv('/content/data/Wine dataset.csv')
print("Первые несколько строк информации в наборе данных:")
print(df.head())
```

```
Первые несколько строк информации в наборе данных:
   class  Alcohol  Malic acid   Ash  Alcalinity of ash  Magnesium  \
0      1    14.23        1.71  2.43               15.6        127
1      1    13.20        1.78  2.14               11.2        100
2      1    13.16        2.36  2.67               18.6        101
3      1    14.37        1.95  2.50               16.8        113
4      1    13.24        2.59  2.87               21.0        118

   Total phenols  Flavanoids  Nonflavanoid phenols  Proanthocyanins  \
0           2.80        3.06                  0.28             2.29
1           2.65        2.76                  0.26             1.28
2           2.80        3.24                  0.30             2.81
3           3.85        3.49                  0.24             2.18
4           2.80        2.69                  0.39             1.82

   Color intensity   Hue  OD280/OD315 of diluted wines  Proline
0             5.64  1.04                          3.92     1065
1             4.38  1.05                          3.40     1050
2             5.68  1.03                          3.17     1185
3             7.80  0.86                          3.45     1480
4             4.32  1.04                          2.93      735
```

0秒 完成时间：12:45

Рисунок 1. Информация с набором данных.

Как показано на рисунке 1, «Алкоголь», «Яблочная кислота», «Магний» выполняются в разных масштабах, поэтому масштабирование признаков необходимо выполнять перед каким-либо сравнением или объединением этих данных.

1. Z-оценки(Standardization)

на основе Z-оценки(Standardization)

```
# @title на основе Z-оценки(Standardization)

scaler = StandardScaler()

# 对特定列进行标准化
scaled_values = scaler.fit_transform(numeric_columns)

# 创建包含标准化后的值的 DataFrame
scaled_df = pd.DataFrame(scaled_values, columns=selected_columns)

# 显示标准化后的数据集
print("\n Стандартизированный набор данных:")
print(scaled_df)
```

```
 Стандартизированный набор данных:
     Alcohol  Malic acid  Magnesium
0   1.518613   -0.562250   1.913905
1   0.246290   -0.499413   0.018145
2   0.196879    0.021231   0.088358
3   1.691550   -0.346811   0.930918
4   0.295700    0.227694   1.281985
..       ...         ...        ...
173 0.876275    2.974543  -0.332922
174 0.493343    1.412609   0.158572
175 0.332758    1.744744   1.422412
176 0.209232    0.227694   1.422412
177 1.395086    1.583165  -0.262708

[178 rows x 3 columns]
```

Рисунок 2. Код Z-оценки и результаты

2.Масштабирование "Mean Normalisation"

Масштабирование "Mean Normalisation"

```
# @title Масштабирование "Mean Normalisation"
# 计算每列的均值和范围
column_means = numeric_columns.mean()
column_ranges = numeric_columns.max() - numeric_columns.min()

# 均值归一化
normalized_values = (numeric_columns - column_means) / column_ranges

# 创建包含归一化后的值的 DataFrame
normalized_df = pd.DataFrame(normalized_values, columns=selected_columns)

# 显示归一化后的数据集
print("\nНормализованный набор данных:")
print(normalized_df)
```

```
Нормализованный набор данных:
     Alcohol  Malic acid  Magnesium
0   0.323522   -0.123784   0.296287
1   0.052469   -0.109950   0.002809
2   0.041943    0.004674   0.013679
3   0.360364   -0.076353   0.144113
4   0.062995    0.050129   0.198461
..       ...         ...        ...
173 0.186679    0.654872  -0.051539
174 0.105101    0.310998   0.024548
175 0.070890    0.384121   0.220200
176 0.044574    0.050129   0.220200
177 0.297206    0.348548  -0.040669
```

Рисунок 3. Код масштабирование и результаты

3.MinMax-масштабирование

MinMax-масштабирование

```
# @title MinMax-масштабирование
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
scaled_values = scaler.fit_transform(numeric_columns)

# 创建包含 Min-Max Scaling 后的值的 DataFrame
scaled_df = pd.DataFrame(scaled_values, columns=selected_columns)

# 显示 Min-Max Scaling 后的数据集
print("\nMin-Max Scaling 后的数据集：")
print(scaled_df)
```

```
Min-Max Scaling 后的数据集：
      Alcohol  Malic acid  Magnesium
0    0.842105    0.191700   0.619565
1    0.571053    0.205534   0.326087
2    0.560526    0.320158   0.336957
3    0.878947    0.239130   0.467391
4    0.581579    0.365613   0.521739
..        ...         ...        ...
173  0.705263    0.970356   0.271739
174  0.623684    0.626482   0.347826
175  0.589474    0.699605   0.543478
176  0.563158    0.365613   0.543478
177  0.815789    0.664032   0.282609

[178 rows x 3 columns]
```

Рисунок 4. Код MinMax-масштабированиеи результаты

Обработку выбросов для числовых признаков

1.Обнаружение выбросов

Обнаружение выбросов в данных с помощью метода z-оценки:

Z-score method

```
# @title Z-score method
import pandas as pd
import numpy as np
import re

train = pd.read_csv('/content/data/houses_to_rent.csv')
out = []

def Zscore_outlier(df):
        # 使用正则表达式移除字符串中的货币符号和逗号，并将结果转换为浮点数
        df = df.str.replace(',', '')    # 移除逗号
        df = df.apply(lambda x: re.sub(r'[^\d.]', '', x))    # 移除非数字字符，包括货币符号
        df = df.astype(float)    # 转换为浮点数


        m = np.mean(df)
        sd = np.std(df)

        for i in df:
                z = (i - m) / sd
                if np.abs(z) > 3:
                        out.append(i)

        print("Outliers:", out)

Zscore_outlier(train['rent amount'])
```

```
Outliers: [19500.0, 20000.0, 20000.0, 45000.0, 20000.0, 18000.0, 20000.0, 24000.0, 20000.0, 20000.0]
```

Рисунок 5- Обнаружение выбросов

2. Удалить выбросы

Удалить выбросы

```
# @title Удалить выбросы
# 移除检测到的异常值所在的行
cleaned_train = train[~train['rent amount'].isin(out)]

# 输出处理后的数据
print("Data after removing outliers:")
print(cleaned_train)
```

```
Data after removing outliers:
      Unnamed: 0  city  area  rooms  bathroom  parking spaces floor  \
0              0     1   240      3         3               4     -
1              1     0    64      2         1               1    10
2              2     1   443      5         5               4     3
3              3     1    73      2         2               1    12
4              4     1    19      1         1               0     -
...          ...   ...   ...    ...       ...             ...   ...
6075        6075     1    50      2         1               1     2
6076        6076     1    84      2         2               1    16
6077        6077     0    48      1         1               0    13
6078        6078     1   160      3         2               2     -
6079        6079     1    60      2         1               1     4

          animal      furniture      hoa rent amount property tax  \
0         acept      furnished      R$0     R$8,000      R$1,000
1         acept  not furnished    R$540       R$820        R$122
2         acept      furnished  R$4,172     R$7,000      R$1,417
3         acept  not furnished    R$700     R$1,250        R$150
4     not acept  not furnished      R$0     R$1,200         R$41
```

Рисунок 6-Код и результаты удаления выбросов

3. Заменить выбросы

```
        # 计算中位数
        median = np.median(df)

        # Заменить выбросы медианой
        for i in range(len(df)):
                z = np.abs((df[i] - m) / sd)
                if z > 3:
                        df[i] = median

        # 打印处理后的数据
        print("Processed data:")
        print(df)

Zscore_outlier(train['rent amount'])
```

```
Processed data:
0       8000.0
1        820.0
2       7000.0
3       1250.0
4       1200.0
         ...
6075    1150.0
6076    2900.0
6077     950.0
6078    3500.0
6079    1900.0
Name: rent amount, Length: 6080, dtype: float64
```

Рисунок 7-Код и результаты замены выбросов

Обработку по крайней мере нестандартного признака

Загрузить набор данных

```
import pandas as pd
data = pd.read_csv('/content/data/NYC Accidents 2020.csv', sep=',')
data.head()
```

	CRASH DATE	CRASH TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET NAME	CROSS STREET NAME	ST
0	2020-08-29	15:40:00	BRONX	10466.0	40.89210	-73.833760	POINT (-73.83376 40.8921)	PRATT AVENUE	STRANG AVENUE	
1	2020-08-29	21:00:00	BROOKLYN	11221.0	40.69050	-73.919914	POINT (-73.919914 40.6905)	BUSHWICK AVENUE	PALMETTO STREET	
2	2020-						POINT			

Рисунок 6-Код и результаты Загрузить набор данных

Обработка даты

```
[15] # День
     data['day'] = data['dt'].dt.day
     # Месяц
     data['month'] = data['dt'].dt.month
     # Год
     data['year'] = data['dt'].dt.year
```

```
data.head()
```

...	COLLISION_ID	VEHICLE TYPE CODE 1	VEHICLE TYPE CODE 2	VEHICLE TYPE CODE 3	VEHICLE TYPE CODE 4	VEHICLE TYPE CODE 5	dt	day	month	year
...	4342908.0	Sedan	Station Wagon/Sport Utility Vehicle	NaN	NaN	NaN	2020-08-29	29	8	2020
...	4343555.0	Sedan	Sedan	NaN	NaN	NaN	2020-08-29	29	8	2020

Рисунок 7-Код и результаты обработка даты

Отбор признаков

1. Методы фильтрации (filter methods).

Выбор признаков осуществлялся с использованием метода выбора порога дисперсии.

```
y = data['NObeyesdad']   # 目标变量

# 实例化和拟合方差阈值选择器
selector = VarianceThreshold(threshold=0.1)   # 这里选择方差的阈值为0.1，可以根据实际情况调整
X_selected = selector.fit_transform(X)

# 获取所选特征的索引
selected_features_index = selector.get_support(indices=True)

# 根据所选特征的索引获取特征名称
selected_features = X.columns[selected_features_index]

# 打印选择的特征
print("выбранные функции,")
print(selected_features)
```

```
выбранные функции,
Index(['Weight', 'family_history_with_overweight', 'FAVC_z', 'NCP_z', 'TUE_z',
       'CALC_z', 'Age_bin_minmax'],
      dtype='object')
```

Рисунок 8-Код и результаты Методы фильтрации

2. Методы обертывания (wrapper methods).

```
data = pd.read_csv(url)

# 分割特征和目标变量
X = data.drop('NObeyesdad', axis=1)  # 特征
y = data['NObeyesdad']  # 目标变量

# 初始化随机森林分类器
estimator = RandomForestClassifier()

# 初始化 RFE 特征选择器
selector = RFE(estimator, n_features_to_select=5, step=1)

# 拟合 RFE 特征选择器
selector = selector.fit(X, y)

# 获取选择的特征索引
selected_features_index = selector.get_support(indices=True)

# 根据所选特征的索引获取特征名称
selected_features = X.columns[selected_features_index]

# 打印选择的特征
print("所选特征：")
print(selected_features)
```

```
所选特征：
Index(['Height', 'Weight', 'FCVC_minmax', 'FAF_minmax', 'TUE_z'], dtype='object')
```

Рисунок 9-Код и результаты Методы обертывания

3. Методы вложений (embedded methods).

```
# 读取数据
url = "/content/data/estimation_of_obesity_levels_based_on_eating_habits_and_physical_condition.csv"
data = pd.read_csv(url)

# 分割特征和目标变量
X = data.drop('NObeyesdad', axis=1)  # 特征
y = data['NObeyesdad']  # 目标变量

# 初始化 Lasso 模型
lasso_model = Lasso(alpha=0.1, max_iter=10000)

# 使用 Lasso 模型进行特征选择
selector_lasso = SelectFromModel(lasso_model)

# 拟合 SelectFromModel 特征选择器
selector_lasso.fit(X, y)

# 获取选择的特征
selected_features_lasso = X.columns[selector_lasso.get_support()]

# 打印使用 Embedded Methods 选择的特征
print("使用 Embedded Methods 选择的特征：")
print(selected_features_lasso)
```

```
使用 Embedded Methods 选择的特征：
Index(['Weight'], dtype='object')
```

Рисунок 10-Код и результаты Методы вложений