

# Рубежный контроль

Лю Бовэнь

Вариант:19

Задача1:19. Задача2:39.

Задача1:19

```
from sklearn.preprocessing import MinMaxScaler
import pandas as pd

# 假设你的数据存储在一个DataFrame中，名为df
# 例如:
df = pd.read_csv('diabetes.csv')

# 创建一个MinMaxScaler对象
scaler = MinMaxScaler()

# 需要进行归一化的列
columns_to_normalize = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

# 对需要归一化的列进行平均归一化处理
df[columns_to_normalize] = scaler.fit_transform(df[columns_to_normalize])

# 显示归一化后的数据
print(df.head())
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	0.352941	0.743719	0.590164	0.353535	0.000000	0.500745	
1	0.058824	0.427136	0.540984	0.292929	0.000000	0.396423	
2	0.470588	0.919598	0.524590	0.000000	0.000000	0.347243	
3	0.058824	0.447236	0.540984	0.232323	0.111111	0.418778	
4	0.000000	0.688442	0.327869	0.353535	0.198582	0.642325	

	DiabetesPedigreeFunction	Age	Outcome
0	0.234415	0.483333	1
1	0.116567	0.166667	0
2	0.253629	0.183333	1
3	0.038002	0.000000	0
4	0.943638	0.200000	1

Задача2:39

```

from sklearn.feature_selection import SelectKBest, mutual_info_classif
import pandas as pd

# 假设你的数据存储在一个DataFrame中, 名为df
# 例如:
df = pd.read_csv('diabetes.csv')

# 提取特征列和目标列
X = df.drop(columns=['Outcome']) # 特征列
y = df['Outcome'] # 目标列

# 创建 SelectKBest 对象, 并指定互信息作为评分函数
selector = SelectKBest(score_func=mutual_info_classif, k='all')

# 对特征列进行选择
X_selected = selector.fit_transform(X, y)

# 获取选择后的特征列索引
selected_columns_index = selector.get_support(indices=True)

# 获取选择后的特征列名
selected_columns = X.columns[selected_columns_index]

# 打印选择后的特征列名
print(selected_columns)

```

```

Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age'],
      dtype='object')

```