

实现思路

笔记本：我的第一个笔记本

创建时间：2023/10/24 18:18

更新时间：2023/10/24 21:19

作者：3446994995@qq.com

实现思路

1. 读取数据，数据是'.csv'文件，用使用`pd.read_csv("H:\\data\\train_data.csv")`用来读取数据，然后查看数据情况

	sample_id	feature0	feature1	...	feature105	feature106	label
0	0	51.567250	288358.4	...	-5.256075	180.977310	4
1	1	63.804874	288358.4	...	16.505952	314.783263	5
2	2	49.138527	288358.4	...	64.707581	183.304610	2
3	3	NaN	288358.4	...	-9.662399	212.302670	2
4	4	76.520831	288358.4	...	59.733069	135.541233	2
...
9995	9995	66.948837	288358.4	...	-1.625234	NaN	0
9996	9996	NaN	288358.4	...	11.018358	250.312228	0
9997	9997	42.208350	288358.4	...	75.537477	241.702576	2
9998	9998	26.797447	288358.4	...	-10.481948	113.104089	0
9999	9999	63.908598	288358.4	...	-3.776887	101.844761	0

[10000 rows x 109 columns]

2. 对数据进行归一化处理
3. 用`train_data.iloc[:, 2:-1]` and `train_data.iloc[:, -1]`两行代码对数据进行划分，划分出特征和标签
4. 对缺失数据进行补充，我分别采用了0补充、均值补充、向前补充、向后补充
5. 随后调用模型，进行训练。这里我用了Tree模型、KNN模型、NN模型、逻辑回归模型，最后还是Tree模型效果最好。

Train:Classification Report for Decision Tree:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5144
1	1.00	1.00	1.00	1062
2	1.00	1.00	1.00	1613
3	1.00	1.00	1.00	884
4	1.00	1.00	1.00	554
5	1.00	1.00	1.00	743
accuracy			1.00	10000
macro avg	1.00	1.00	1.00	10000
weighted avg	1.00	1.00	1.00	10000

Test:Classification Report for Decision Tree:

	precision	recall	f1-score	support
0	0.28	0.26	0.27	176
1	0.51	0.43	0.47	166
2	0.29	0.28	0.29	171
3	0.97	0.92	0.95	169
4	0.72	0.98	0.83	156
5	0.94	0.93	0.93	162
accuracy			0.62	1000
macro avg	0.62	0.63	0.62	1000
weighted avg	0.61	0.62	0.61	1000