NNI学生项目2020

Task 1.3.2

ID: 14 Name: 中科大飙车队

任务描述

- 阅读 [Feature Engineering Test Example（页面底端）](#)
- 自主选择一个不包括在范例内的数据集，进行Binary-classification benchmarks实验，比较baseline accuracy与automl accuracy
- 尤其鼓励同学们在实验过程中进一步优化算法

数据集

我们选用的数据集是来自UCI数据集中的http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#。该数据集的来源是[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014。该数据集是用来分析银行电话推销的结果，任务的目标是用户是否会接受推销，因此是一个二分类问题。每个样本共有20个属性，其中有10个范畴属性，有10个数值属性。该数据集有四个子数据集，其中bank-additional-full为新版完整数据集，bank-additional为新版完整数据集随机采样10%的结果。

代码实现

基本基于作者的代码。主要修改是对于数据的读取方式;添加了数据没有的id这一列;将数据的label由“yes/no"映射到”1/0“，如下所示

```
    df = pd.read_csv(file_name,sep=";")

    # list is a column_name generate from tuner
    df[id_index] = range(len(df))
    if 'sample_feature' in RECEIVED_PARAMS.keys():
        sample_col = RECEIVED_PARAMS['sample_feature']
    else:
        sample_col = []
    df.loc[df[target_name] == 'no',target_name] = 0
    df.loc[df[target_name] == 'yes',target_name] = 1
```

另外一个需要展示的是search_space.json.针对数据做了设置

```
{
    "count":[
        "age","job","marital","education","default","housing","loan",
        "contact","month","dayofweek","duration","campaign","pdays",
        "previous","poutcome","emp.var.rate","cons.price.idx","cons.conf.idx",
        "euribor3m","nr.employed"
    ],
    "aggregate":[
        [
            "age","duration","campaign","pdays",
```

```
            "previous","emp.var.rate","cons.price.idx","cons.conf.idx",
            "euribor3m","nr.employed"
        ],
        [
            "age","duration","campaign","pdays",
            "previous","emp.var.rate","cons.price.idx","cons.conf.idx",
            "euribor3m","nr.employed"
        ]
    ],
    "embedding":[

            "job","marital","education","default","housing","loan",
            "contact","month","dayofweek","poutcome"

    ],
    "crosscount":[
        [
            "age","job","marital","education","default","housing","loan",
            "contact","month","dayofweek","duration","campaign","pdays",

 "previous","poutcome","emp.var.rate","cons.price.idx","cons.conf.idx",
            "euribor3m","nr.employed"
        ],
        [
            "age","job","marital","education","default","housing","loan",
            "contact","month","dayofweek","duration","campaign","pdays",

 "previous","poutcome","emp.var.rate","cons.price.idx","cons.conf.idx",
            "euribor3m","nr.employed"
        ]
    ]
}
```

特征组合方式

我们使用了aggregate, count, crosscount, embedding等几种特征组合方式。需要注意的是，aggregate仅能支持对数值型特征的特征组合，embedding只能支持对类别型特征的特征组合。

算法优化及代码实现

我们打算将优化特征选择算法作为Task1.4部分完成的内容，因此详细的优化算法和改进在Task1.4中叙述。我们这里介绍一种想到的简单的优化方法。我们的优化方法借鉴了模拟退火算法的思想。在作者原本的实现中，特征选择的实现如下所示：

```
sample_p = np.array(self.estimate_sample_prob) /
np.sum(self.estimate_sample_prob)
            logger.info(str(sample_p))
            sample_size = min(128, int(len(self.candidate_feature) *
self.feature_percent))
            sample_feature = np.random.choice(
                self.candidate_feature,
                size = sample_size,
                p = sample_p,
                replace = False
                )
```

特征的选择是得出概率之后直接按照概率进行采样。而我们的改进是希望随着迭代次数的增多，选择高概率的candidate的概率越大，这样有助于减小在靠后的迭代轮次的搜索空间的大小。具体说来，我们对算法的采样概率做如下改变：

$$P' = P * \exp(\frac{PT}{B})$$

其中P是采样的概率，T是到目前为止已经进行特征选取迭代的轮次数。B是一个常数。可以注意到随着迭代的轮数增加，越大的概率P经过变换后的采样概率就会变得越大。使得该特征被选取的概率越大。对概率进行变换之后再进行归一化即可。具体的实现如下所示：

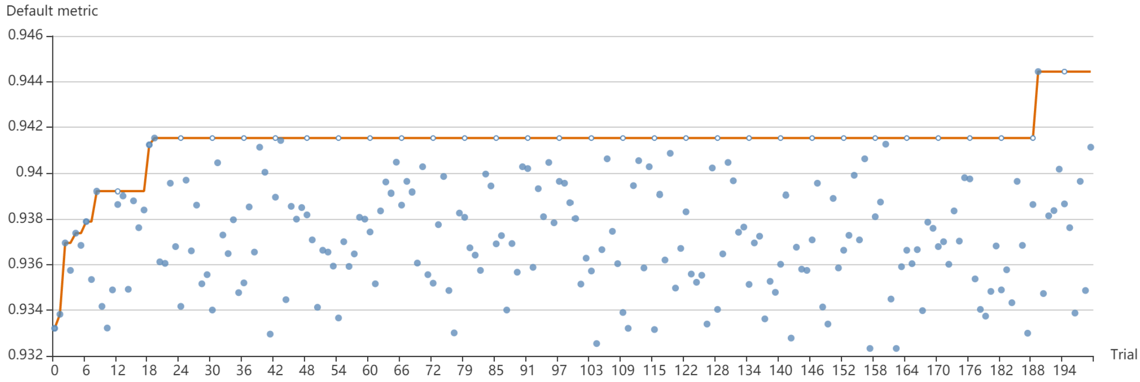
```
            B = 200
            sample_p = sample_p * np.exp(sample_p*self.count/B)
            sample_p = sample_p / np.sum(sample_p)
            sample_feature = np.random.choice(
                self.candidate_feature,
                size = sample_size,
                p = sample_p,
                replace = False
                )
```

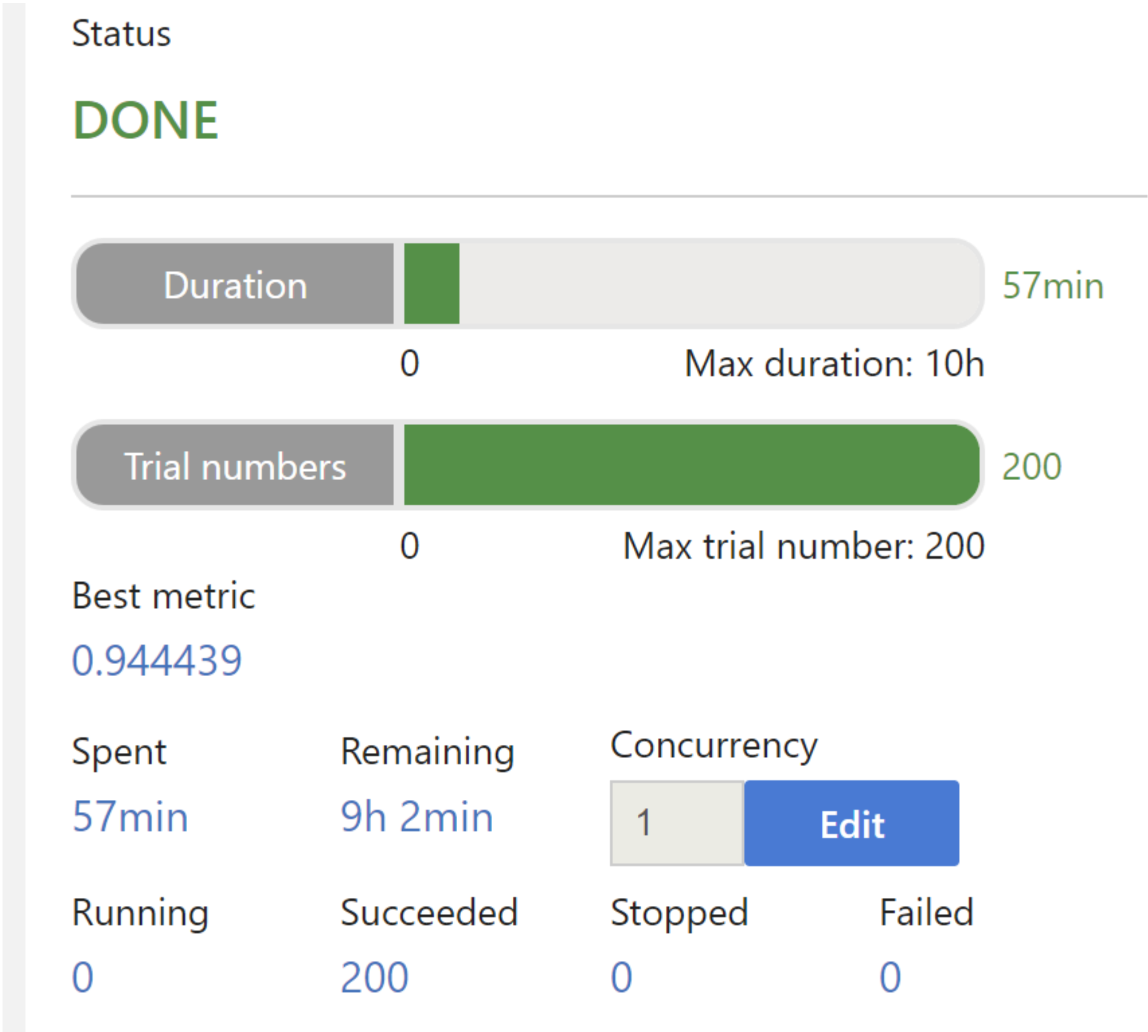
结果展示

首先展示未经过优化的结果

bank-additional

Trial No.	ID	Duration	Status	Default metric
189	dMRlY	9s	SUCCEEDED	0.944439
19	AY54G	9s	SUCCEEDED	0.941524
43	IRIJk	8s	SUCCEEDED	0.941417
160	on5hx	8s	SUCCEEDED	0.941257
18	PFdim	8s	SUCCEEDED	0.94123
199	U23zo	9s	SUCCEEDED	0.941123
39	pQPVc	8s	SUCCEEDED	0.941123
118	v22Mq	8s	SUCCEEDED	0.940856
106	ZDDwE	10s	SUCCEEDED	0.940615
156	pXIY8	7s	SUCCEEDED	0.940615

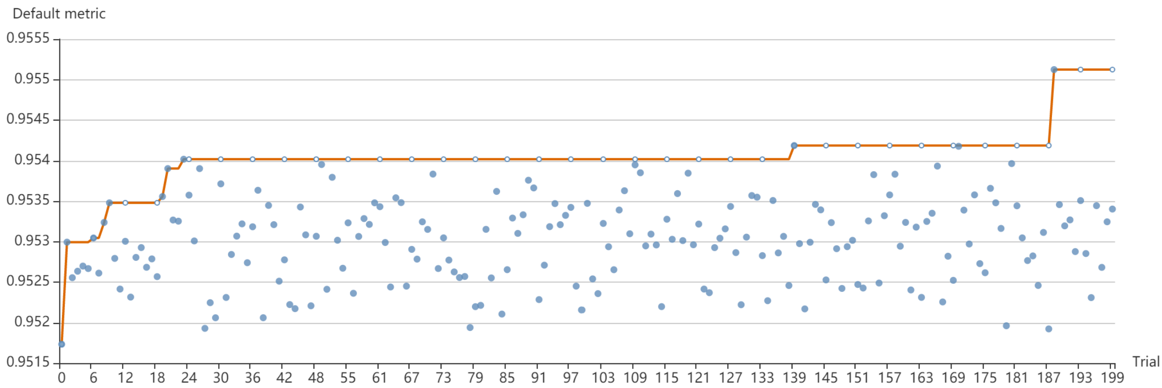


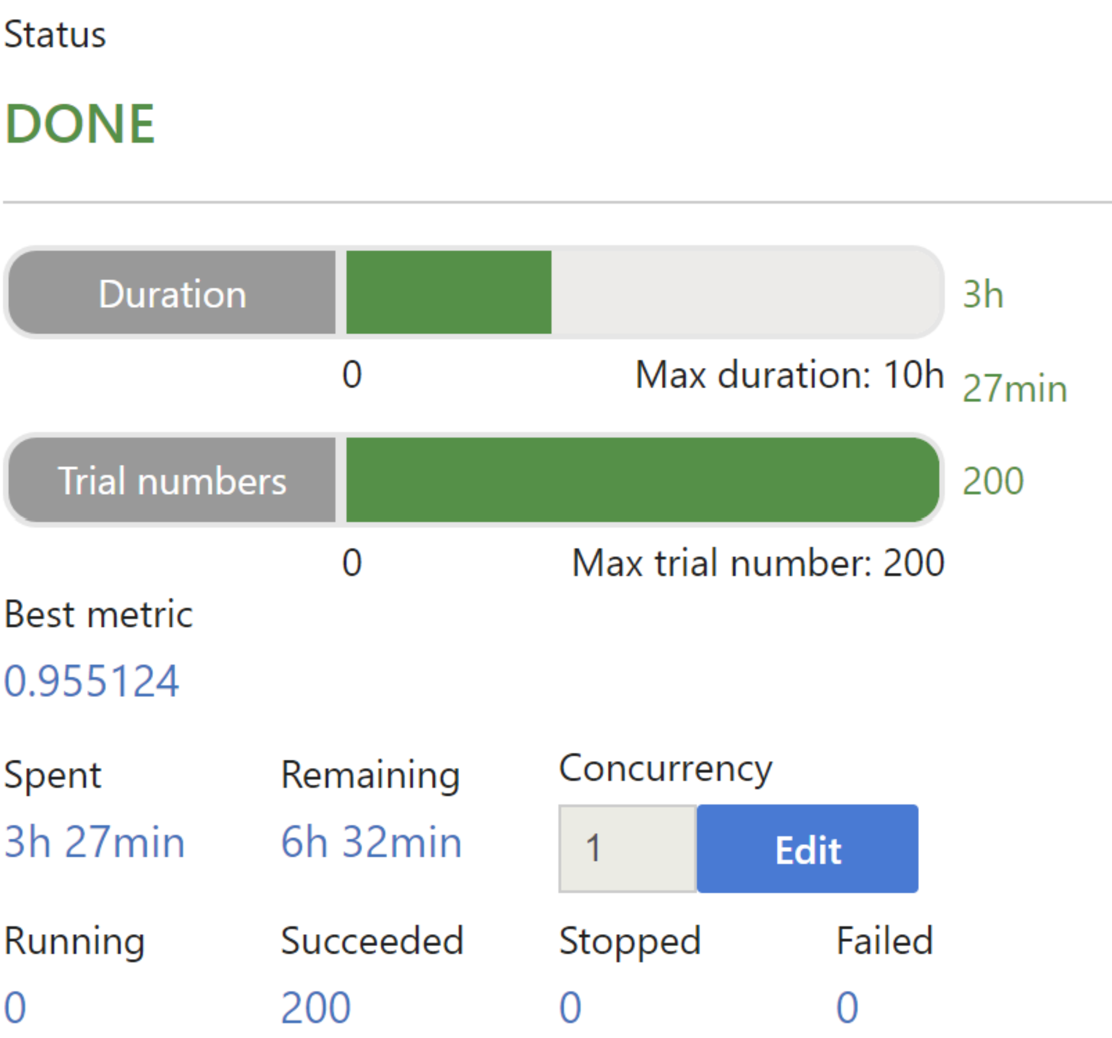


我们运行了200轮，得到了如上的结果。花费的时间为57分钟。

bank-additional-full

Trial No.	ID	Duration	Status	Default metric
188	O8wAs	52s	SUCCEEDED	0.955124
139	q6Oer	1min 6s	SUCCEEDED	0.954186
170	RclE6	1min 11s	SUCCEEDED	0.954177
23	a2OIl	49s	SUCCEEDED	0.954017
180	vDSaS	52s	SUCCEEDED	0.953963
49	k3A2M	50s	SUCCEEDED	0.953952
109	C9xdb	56s	SUCCEEDED	0.953948
166	LXKAZ	1min 7s	SUCCEEDED	0.953932
20	IbhYE	46s	SUCCEEDED	0.953902
26	GPH4p	41s	SUCCEEDED	0.953902





结果如上所示

优化后的结果

bank-additional

Trial No.	ID	Duration	Status	Default metric
44	rVkvZ	13s	SUCCEEDED	0.943971
168	DLKVU	15s	SUCCEEDED	0.942701
195	SdIVN	11s	SUCCEEDED	0.942513
71	Mbk0B	8s	SUCCEEDED	0.941738
29	aOJ2A	7s	SUCCEEDED	0.941497
74	aHQ7a	9s	SUCCEEDED	0.941417
30	JvWNq	8s	SUCCEEDED	0.941364
66	JiENd	21s	SUCCEEDED	0.941203
114	HF1yE	9s	SUCCEEDED	0.941176
166	hXPQb	10s	SUCCEEDED	0.94115

bank-additional-full

Trial No.	ID	Duration	Status	Default metric
16	siFGm	48s	SUCCEEDED	0.954424
192	myxTP	51s	SUCCEEDED	0.954348
174	oHyVf	51s	SUCCEEDED	0.954148
38	xg0cA	36s	SUCCEEDED	0.954143
63	MVNfu	41s	SUCCEEDED	0.954113
148	iYT4H	45s	SUCCEEDED	0.95403
5	AMe6B	44s	SUCCEEDED	0.954003
56	atcnC	52s	SUCCEEDED	0.954002
110	ZWbaZ	40s	SUCCEEDED	0.953902
168	ihfn7	54s	SUCCEEDED	0.953886

尽管找出的最优参数结果没有更好，但是最好的前10个实验的结果的平均值要好于未经优化的结果。

总结

- 通过优化后的方法可以提高Top10最优参数的平均表现。
- 数据集本身的baseline较高，不能完全体现出NNI优化的效果。
- 在1.4中对特征选取的方法再进行设计

Dataset	baseline auc	automl auc	number of cat	number of num
bank-additional	0.933209	0.944439	10	10
bank-additional-full	0.951733	0.955124	10	10