Machine Learning With A Heart

链接 https://www.drivendata.org/competitions/54/machine-learning-with-a-heart/

竞赛排名（截止于2019.06.06）

数据介绍

slope_of_peak_exercise_st_segment运动峰值ST段的斜率，心电图显示出流向心脏的血液质量	Int	the slope of the peak exercise ST segment, an electrocardiography read out indicating quality of blood flow to the heart
Thal铊负荷试验测定心脏血流量的结果，可能值为正常、固定缺损、可逆性缺损	Categorical	results of thallium stress test measuring blood flow to the heart, with possible values normal, fixed_defect, reversible_defect
resting_blood_pressure静息血压	Int	resting blood pressure
chest_pain_type胸痛类型(4个值)	Int	chest pain type (4 values)
num_major_vessels主要血管的数目(0-3)由荧光染色	Int	number of major vessels (0-3) colored by flourosopy
fasting_blood_sugar_gt_120_mg_per_dl空腹血糖> 120 mg/dl	Binary	fasting blood sugar > 120 mg/dl
resting_ekg_results静息心电图结果(值0,1,2)	Int	resting electrocardiographic results (values 0,1,2)
serum_cholesterol_mg_per_dl 血清胆甾醇浓度为mg/dl	Int	serum cholestoral in mg/dl
oldpeak_eq_st_depression运动相对于休息引起的ST值降低，是心电图异常的一种测量方法	Float	oldpeak = ST depression induced by exercise relative to rest, a measure of abnormality in electrocardiograms
sex	Binary	0: female, 1: male
age	Int	age in years
max_heart_rate_achieved最高心率(每分钟跳动次数)	Int	maximum heart rate achieved (beats per minute)
exercise_induced_angina运动引起的胸痛(0:假，1:真)	Binary	exercise-induced chest pain (0: False, 1: True)

数据观察

由于数据特征中包含类别数据，因此使用哑变量转换

```
train_values = pd.get_dummies(train_values)
train_values.head()
```

patient_id	resting_blood_pressure	serum_cholesterol_mg_per_dl	oldpeak_eq_st_depression	age	max_heart_rate_achieved	slope_of_peak_exercise_st_segment_
0z64un	128	308	0.0	45	170	
ryoo3j	110	214	1.6	54	158	
yt1s1x	125	304	0.0	77	162	
l2xjde	152	223	0.0	40	181	
oyt4ek	178	270	4.2	59	145	

添加衍生数据特征

通常建模过程中需要尝试添加衍生特征来增强模型性能，如各样本与其各类别对应特征中位数/均值差/绝对值差等等

```
# 添加新特征
# serum_cholesterol_mg_per_dl 中位数
serum_0 = list()
serum_1 = list()
serum_c = list()
for i in range(len(train_values.index)):
    temp_0 = train_values['serum_cholesterol_mg_per_dl'][i] - 237.5
    temp_1 = train_values['serum_cholesterol_mg_per_dl'][i] - 255.5
    serum_0.append(temp_0)
    serum_1.append(temp_1)
    if abs(temp_0)>abs(temp_1):
        serum_c.append(0)
    else:
        serum_c.append(1)
```

数据归一化

由于各数据特征的尺度不同，因此需要考虑对数据归一化，如标准归一化、最大最小归一化等

```
# 数据归一化
# scaler = StandardScaler()
scaler = MinMaxScaler(feature_range=(0, 1))
temp_train_values = scaler.fit_transform(train_values)
train_values = pd.DataFrame(temp_train_values, columns = train_values.columns)

resting_blood_pressure_mean = list()
serum_cholesterol_mg_per_dl_mean = list()
max_heart_rate_achieved_mean = list()
age_mean = list()
for f in reg_feature_list:
    temp_list = list()
    for i in range(len(train_values.index)):
        temp_list.append(train_values[f][i] - np.mean(train_values[f]))
    train_values[f+'_mean'] = temp_list
```

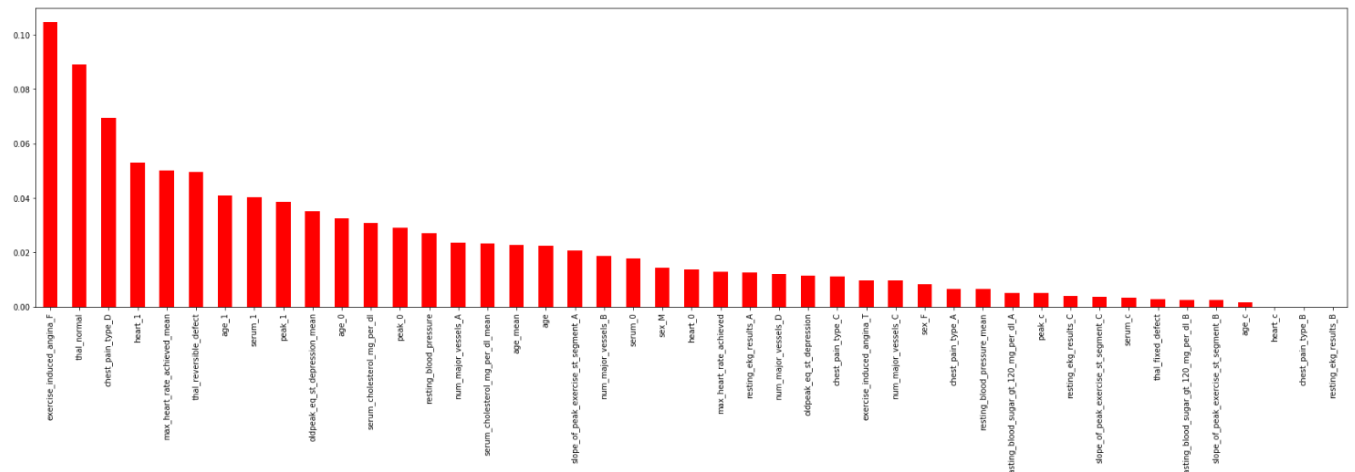
	resting_blood_pressure	serum_cholesterol_mg_per_dl	oldpeak_eq_st_depression	age	max_heart_rate_achieved	slope_of_peak_exercise_st_segment_A
0	0.395349	0.415525	0.000000	0.333333	0.698113	1.0
1	0.186047	0.200913	0.258065	0.520833	0.584906	0.0
2	0.360465	0.406393	0.000000	1.000000	0.622642	1.0
3	0.674419	0.221461	0.000000	0.229167	0.801887	1.0
4	0.976744	0.328767	0.677419	0.625000	0.462264	0.0

数据特征排名

通常将全部数据特征放入预测模型中效果反而不佳，可以考虑特征选择，如随机森林、XGBOOST等树模型进行特征排名

```
1  from sklearn.ensemble import RandomForestClassifier
2  # use random forest for feature selection
3  rfc = RandomForestClassifier()
4  rfc.fit(train_values, train_labels)
5  # print(rfc.feature_importances_)
6  plt.figure(figsize=(30, 7))
7  features = pd.Series(rfc.feature_importances_, index= train_values.columns)
8  features.nlargest(50).plot(kind = 'bar', color = 'r')
9  plt.show()
```

```
c:\users\jomin\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
```



构建模型&寻找参数

利用pipeline可以衔接网格搜索最优参数构建模型

```
# LogisticRegression [添加全部数据特征]0.6856
# pipe = Pipeline(steps=[('scale', StandardScaler()),
#                        ('logistic', LogisticRegression())])
# param_grid = {'logistic__C': [0.0001, 0.001, 0.01, 1, 10],'logistic__penalty': ['l1', 'l2']}

# BayesianRidge    0.53845    [全部特征]0.3657
pipe1 = Pipeline(steps=[('scale', StandardScaler()),
                        ('bay', BayesianRidge())])
param_grid1 = {'bay__alpha_1': [1e-06],'bay__alpha_2': [1e-06,1e-05,1e-04],
               'bay__n_iter':np.arange(100,200,100),'bay__fit_intercept':[True,False],
               'bay__lambda_1':[1e-06,1e-05],'bay__lambda_2':[1e-06,1e-05],'bay__tol':[0.0001,0.0002]}

```

```
gs1 = GridSearchCV(estimator=pipe1,param_grid=param_grid1,cv=3)
```

预测结果处理

由于本次预测竞赛是分类问题，因此对于模型输出0-1的预测值进行分类，其发现单一阈值法效果不佳的主要原因在于处于0.2-0.7间的部分样本模型预测有误，即存在0.3x的样本属于1类或者存在0.6x的样本属于0类的情况（个人认为测试数据集中存在异常点）

```
heart_disease_present = list()
for i in range(len(my_submission.heart_disease_present)):
    if my_submission.heart_disease_present[i]<0.17:
        heart_disease_present.append(0)
    elif my_submission.heart_disease_present[i]>0.75:
        heart_disease_present.append(1)
    else:
        heart_disease_present.append(my_submission.heart_disease_present[i])
submission = pd.DataFrame()
submission['patient_id'] = list(my_submission.index)
submission['heart_disease_present'] = heart_disease_present
submission.head()
```

模型融合

尝试对模型融合发现效果不佳，主要原因是BayesianRidge模型已经处于很高得分