智算之道——人工智能应用挑战赛(初赛)-baseline 学习笔记

1.读入相关的库

```
import os
import pandas as pd
import warnings
from itertools import combinations
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.model_selection import StratifiedKFold
from tqdm import tqdm

from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier

%matplotlib inline
warnings.filterwarnings('ignore')
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
```

2.读取数据

```
path = '/home/kesci/data/competition_A/'
train_df = pd.read_csv(path+'train_set.csv')
test_df = pd.read_csv(path+'test_set.csv')
submission = pd.read_csv(path+'submission_example.csv')
print('Train Shape:{}\nTest Shape:{}'.format(train_df.shape,test_df.shape))
train_df.head()
```

这部分可以根据具体情况而定。

根据训练集的列可以得到大致如下三种特征：数字列、二值列（0或1）、字符列：

```
num_columns = ['年龄','体重','身高','体重指数', '腰围', '最高血压', '最低血压',
'好胆固醇', '坏胆固醇', '总胆固醇','收入']
zero_to_one_columns = ['肥胖腰围','血脂异常','PVD']
```

```
str_columns = ['性别','区域','体育活动','教育','未婚','护理来源','视力不佳','饮酒','高血压',
'家庭高血压', '糖尿病', '家族糖尿病','家族肝炎', '慢性疲劳','ALF']
```

3.特征工程

这部分是关键：

- 字符编码
  ```
  for i in tqdm(str_columns):
  lbl = LabelEncoder()
  train_df[i] = lbl.fit_transform(train_df[i].astype(str))
  test_df[i] = lbl.fit_transform(test_df[i].astype(str))
  ```
- 数据归一化
  ```
  train_df[num_columns] = MinMaxScaler().fit_transform(train_df[num_columns])
  test_df[num_columns] = MinMaxScaler().fit_transform(test_df[num_columns])
  ```
- 空值填充
  ```
  train_df.fillna(0,inplace=True)
  test_df.fillna(0,inplace=True)
  ```

4.定义模型

- 准备数据
  ```
  all_columns = [i for i in train_df.columns if i not in ['肝炎','ID']]
  train_x,train_y = train_df[all_columns].values,train_df['肝炎'].values
  test_x = test_df[all_columns].values
  submission['hepatitis'] =0
  ```

  训练使用CatBoostClassifier模型，迭代次数为200，初始学习率为0.1，用5折交叉验证。

- 训练模型
  ```
  kfold = StratifiedKFold(n_splits=5, shuffle=False)
  model = CatBoostClassifier(
  iterations=200,
  learning_rate=0.1,
  loss_function='Logloss'
  )
  for train, valid in kfold.split(train_x, train_y):
  X_train, Y_train = train_x[train], train_y[train]
  X_valid, Y_valid = train_x[valid], train_y[valid]
  model.fit(X_train,Y_train, eval_set=(X_valid, Y_valid),use_best_model=True)
  ```

```
Y_valid_pred_prob = model.predict_proba(X_valid)
submission['hepatitis'] += model.predict_proba(test_x)[:,1] / 5
```