开课吧-机器学习数据竞赛-第二章

时间：2020-04-12

一. 二分类问题

1.1 分类回归相关概念

概念：输入变量 X 和输出变量 Y 有不同的类型，可以是连续的，也可以是离散的。人们根据输入、输出变量的不同类型，对预测任务给与不同的名称：输入变量和输出变量均为连续变量的预测问题称为回归问题；输出变量为有限个离散变量的预测问题称为分类问题；输入变量与输出变量均为变量序列的预测问题称为标注问题。

1.2 二分类

二分类问题就是简单的“是否”、“有无”问题，如下图我们判断是否为皮卡丘。我们如果提前预知下图为皮卡丘，那么通过我们的视神经系统能够很快分辨出下图是否为皮卡丘，但是对于机器来说分辨这张图却不是那么容易，更具体的来说机器只能读取这幅图的数字特征（如图像的大小，通道数等），在此我们以每个像素点的三原色对应的数值作为这幅图的数组特征。

1.3 二分类评价指标

1）准确率
2）混淆矩阵
3）精准率,召回率，F1_score
4）auc
5）logloss

1.4 二分类算法

1. Logistic 回归
2. SVM
3. 决策树
4. 随机森林
5. Adaboost
6. xgboost
7. lightgbm
8. catboost
9.朴素贝叶斯

二. 逻辑回归

2.1 线性回归原理

概念：线性回归（Linear Regression）是一种通过属性的线性组合来进行预测的线性模型，其目的是找到一条直线或者一个平面或者更高维的超平面，使得预测值与真实值之间的误差最小化。

线性回归：

$$h(x) = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n$$

1）当只有一条 x_1 时，只有 $h(x)$ 为直线

2）当有两个值 x_1，x_2 两个变量的时候，$h(x)$ 为一个平面

3）当有更多变量时，$h(x)$ 为高维的。

线性回归是通过数据在 N 维空间找到 $h(x)$ 来描述这些数据的规律，这是一个叫做拟合的过程，$h(x)$ 叫做拟合线。

$h(x)$ 的预测值会和真实值会有所偏差，真实统计和 $h(x)$ 预测数据的差称为残差。残差有正的有负的，为了降低计算复杂性，我们使用这个差值的平方进行计算。

为了获得最好的$h(x)$,保证个点与实际数据的残差平方的总和最小。

> 损失函数（Loss Function ）是定义在单个样本上的，算的是一个样本的误差。
> 代价函数（Cost Function ）是定义在训练集上的，是所有样本误差的平均，也就是损失函数的平均。
> 目标函数（Object Function）定义为：最终需要优化的函数。等于经验风险+结构风险（Cost Function + 正则化项）。
> https://blog.csdn.net/lyl771857509/article/details/79428475

这里选取平方差代价函数为：

$$J=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-h(x_i)\right)^2$$

为了获得使得J最小的W^T和b：

主要有：

1）偏导法

2）正规方程法

3）梯度下降 等等

优缺点：

1） 权重W^T是每个x的权重，通过W^T的大小可以看出每个x的权重的大小，可以判断因子的重要性

2） 有很好的解释性

3）**缺点**：非线性数据拟合不好

2.2 逻辑回归原理

从 3.1 我们知道，从上面$h(x)$公式我们可以发现，$h(x)$预测值是连续的，所以这是一个回归模型。那如果我们希望输出的值是离散的，也就是预测值是离散值，所以线性回归是回归模型。那需要将$h(x)$进行一次函数转换，变成$g(z)$。其中$g(z)$某些值属于类别 1，另一些$g(z)$值属于类别，这样的模型则为二分类模型。二元逻辑回归就是这样来的。

此时函数 g 一般为：

$$g(z)=\frac{1}{1+e^{-z}}$$

其中 $z=h(x)$。

$g(z)$ 一般叫 *Sigmoid* 函数或者 logistic 函数。

有了 *Sigmoid* 函数之后，其值取值范围在[0,1]。一般 *Sigmoid* 函数计算得到的值大于等于 0.5 的归为类别 1，小于 0.5 的归为类别 0

$$P(y=1\mid x,\theta)=h_\theta(x)$$

$$P(y=0\mid x,\theta)=1-h_\theta(x)$$

如果按线性回归的思想代价函数是：

$$J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2=\frac{1}{2m}\sum_{i=1}^{m}\left(\frac{1}{1+e^{-\theta^T x_i-b}}-y^{(i)}\right)^2$$

但是这个函数不是凸函数，不好优化。

这是二分类模型，那么肯定是预测为真实为 1 的样本预测概率越接近 1，损失越小；和预测为真实为 0 的样本预测概率越接近 0，损失越小。

假设预测真实样本为 1 的概率为 p_i，则预测真实样本为 0 的概率为 $1-p_i$。则预测概率为：

$$p(y_i)=p_i^{\,y_i}*(1-p_i)^{1-y_i}$$

得到这个函数得到最大似然函数：

$$L=\prod_{i}^{N}h(x_i)^{y_i}*(1-h(x_i))^{1-y_i}$$

两边取对数：

$$L(w)=\sum_i (y_i * \log h(x_i)+(1-y_i)\log(1-h(x_i)))$$
$$L(w)=\sum y_i\left(\log h(x_i)-\log(1-h(x_i))\right)+\log\left(1-h(x_i)\right)$$
$$L(w)=\sum y_i\left(\log\frac{h(x_i)}{1-h(x_i)}\right)+\log\left(1-h(x_i)\right)$$
$$L(w)=\sum y_i\left(\log\frac{\frac{1}{1+e^{-w^TX}}}{1-\frac{1}{1+e^{-w^TX}}}\right)+\log\left(1-\frac{1}{1+e^{-w^TX}}\right)$$
$$L(w)=\sum y_i\left(\log\frac{\frac{1}{1+e^{-w^TX}}}{\frac{e^{-w^TX}}{1+e^{-w^TX}}}\right)+\log\left(1-\frac{1}{1+e^{-w^TX}}\right)$$
$$L(w)=\sum y_i\left(w^TX\right)+\log\left(1-\frac{1}{1+e^{-w^TX}}\right)$$
$$L(w)=\sum y_i\left(w^TX\right)+\log\left(\frac{e^{-w^TX}}{1+e^{-w^TX}}\right)$$
$$L(w)=\sum y_i\left(w^TX\right)+\log\left(\frac{e^{-w^TX}}{1+e^{-w^TX}}\right)$$
$$L(w)=\sum\left(y_i\left(w^TX\right)-\log(1+e^{w^TX})\right)$$

这里仅用随机梯度下降优化损失函数：

损失函数：

$$L(w)=\sum\left(y_i\left(w^TX\right)-\log(1+e^{w^TX})\right)$$

代价函数两边对 w 求导：

$$\frac{dL}{dw}=yx-\frac{1}{1+e^{wx}}*e^{wx}*x$$
$$\frac{dL}{dw}=x(y-h(x))$$

最终迭代权值优化：

$$\theta_j=\theta_j-\alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x_i)-y_i\right)x_i^j$$

逻辑回归的优缺点

优点：

1)容易理解和实现，可以观测样本的概率分数

2)训练速度快

3)由于经过了 sigmoid 函数的映射，对数据中小噪声的鲁棒性较好

4)不受多重共线性的影响(可通过正则化进行消除)

缺点：

1)容易欠拟合

2)特征空间很大时效果不好

3)由于 *Sigmoid* 函数的特性，接近 0/1 的两侧概率变化较平缓，中间概率敏感，波动较大；导致很多区间特征变量的变化对目标概率的影响没有区分度，无法确定临界值。

三. LabelEncoder 和 OneHot

3.1 LabelEncoder

LabelEncoder 是将 labels 转为数字

```
例 1：
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit([1, 2, 2, 6])
le.classes_
le.transform([1, 1, 2, 6])
le.inverse_transform([0, 0, 1, 2])
例 2：
le = preprocessing.LabelEncoder()
le.fit(["paris", "paris", "tokyo", "amsterdam"])
list(le.classes_)
list(le.inverse_transform([2, 2, 1]))
```

3.2 OneHot

OneHot 独热编码-one-hot code

假如有三种颜色特征：红、黄、蓝。 在利用机器学习的算法时一般需要进行向量化或者数字化。那么你可能想令 红=1，黄=2，蓝=3. 那么这样其实实现了标签编码，即给不同类别以标签。然而这意味着机器可能会学习到“红<黄<蓝”，但这并不是机器学习的本意，只是想让机器区分它们，并无大小比较之意。所以这时标签编码是不够的，需要进一步转换。因为有三种颜色状态，所以就有3个比特。即红色：１００，黄色：０１０，蓝色：００１。

```
from sklearn import preprocessing
enc=preprocessing.OneHotEncoder()
enc.fit([[0,0,3],[1,1,0],[0,2,1],[1,0,2]])
array=enc.transform([[0,1,3]]).toarray()
print(array)
```

四.交叉验证

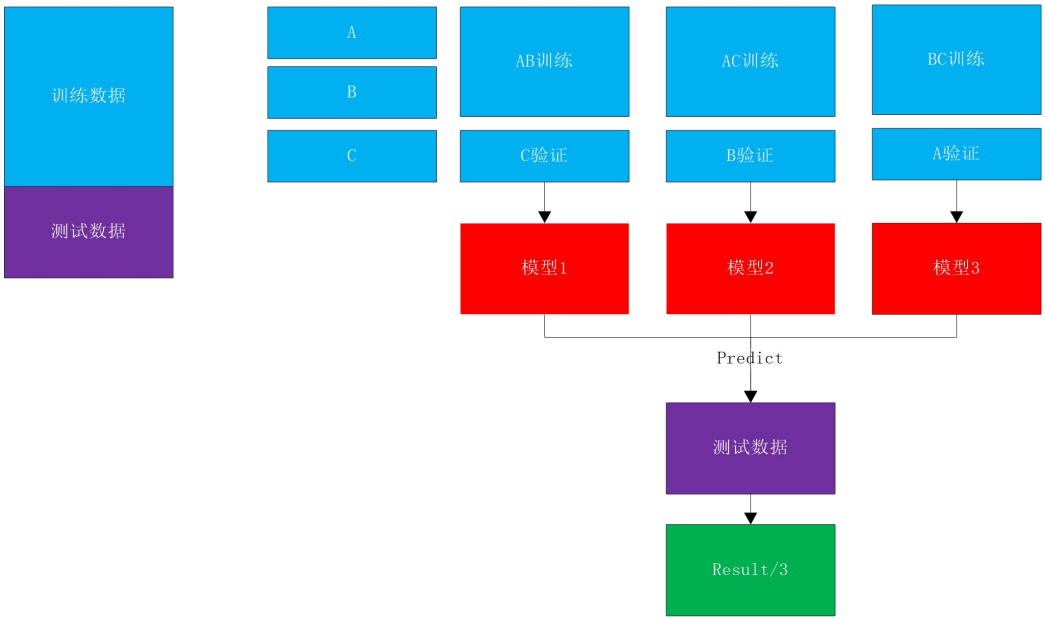
4.1 train_test_split

我们通常通过对训练集进行 train_test_split，划分成 train 和 test 两部分，其中 train 用来训练模型，test 用来评估模型，模型通过 fit 方法从 train 数据集中学习，然后调用 score 方法在 test 集上进行评估，打分；从分数上我们可以知道 模型当前的训练水平如何。

然而，这种方式存在问题：只进行了一次划分，数据结果具有偶然性，如果在某次划分中，训练集里全是容易学习的数据，测试集里全是复杂的数据，这样就会导致最终的结果不尽如意；反之，亦是如此。

4.2 Standard Cross Validation

原始数据分成K组(一般是均分),将每个子集数据分别做一次验证集,其余的K-1组子集数据作为训练集,这样会得到K个模型,用这K个模型最终的验证集的分类准确率的平均数作为此K-CV下分类器的性能指标.原始数据分成K组(一般是均分),将每个子集数据分别做一次验证集,其余的K-1组子集数据作为训练集,这样会得到K个模型,用这K个模型最终的验证集的分类准确率的平均数作为此K-CV下分类器的性能指标.



这样，模型训练多次，保证训练数据具有泛化能力。

五. 二分类比赛快速实现

5.1「二分类算法」提供银行精准营销解决方案

5.1.1 比赛链接

https://www.kesci.com/home/competition/5c234c6626ba91002bfdfdd3

5.1.2 字段说明

NO	字段名称	数据类型	字段描述
1	ID	Int	客户唯一标识
2	age	Int	客户年龄
3	job	String	客户的职业
4	marital	String	婚姻状况
5	education	String	受教育水平
6	default	String	是否有违约记录
7	balance	Int	每年账户的平均余额
8	housing	String	是否有住房贷款

9	loan	String	是否有个人贷款
10	contact	String	与客户联系的沟通方式
11	day	Int	最后一次联系的时间（几号）
12	month	String	最后一次联系的时间（月份）
13	duration	Int	最后一次联系的交流时长
14	campaign	Int	在本次活动中，与该客户交流过的次数
15	pdays	Int	距离上次活动最后一次联系该客户，过去了多久（999 表示没有联系过）
16	previous	Int	在本次活动之前，与该客户交流过的次数
17	poutcome	String	上一次活动的结果
18	y	Int	预测客户是否会订购定期存款业务

评测指标：

AUC(Area Under the Curve)

5.1.3 快速构建 baseline

1）读取数据

2）当利用 lightgbm 模型的时候,类别特征有 3 种处理方式：

*labelencoder

*onehot

*自动类别特征：astype(‘category’)

3）跑一个 train_test_split 模型

4）跑一个 kfold 模型

5）跑一个 lightgbm 模型

六. pandas 进一步学习

七.作业

6.1 Titanic: Machine Learning from Disaster

6.2 离职员工预测