交通数据分析第四次课程作业

1854084 徐晨龙

作业 1：关联

1. 利用$python$实现$Apriori$算法找出其中的频繁项集

   首先定义下列函数：

   $def1：apriori_is_fre(item_judge, databse_target, k_is_fre)$用来寻找某一$k_is_fre$阶项$item_judge$在目标数据库$database_target$中的出现次数。返回值为数值$count$

   $def2：apriori_gen(frequent_item_set_ori)$利用频繁项集，生成下一阶潜在频繁项集，由于本次作业数据量较少和剪枝方法复杂，没有使用剪枝的方法。返回为列表，潜在频繁项集$frequent_item_set_new$

   $def3：apriori_subset(item_set_subset, k_subset, database_subset, suport)$是从$k_subset$阶潜在频繁项集$item_set_subset$中根据支持度$support$生成频繁项集。返回值为频繁项集$frequent_item_set_subset$

   主函数通过生成一阶潜在频繁项集，利用 def3 和 def1 生成新的频繁项集，利用 def2 生成新的潜在频繁项集，直到生成的潜在频繁项集为空集，说明得到频繁项集，得到本文中的频繁项集为

   $$[E, K, O]$$

   其支持度为

   $$support = 0.6$$

2. 列举所有与下面原规则匹配的强关联规则（给出支持度和置信度 s）

   则利用穷举的方式生成输出大于最小置信度的关联规则：

```
--association rules--support--confidence
KO-->E                0.6          1.0
EO-->K                0.6          1.0
O-->EK                0.6          1.0
```

图 1-1 强关联规则

与例子中所给出的元规则$(A, B) --(C)$相类似的有：

(K, O)--(E)[0.6,1]　(EO)---(K)[0.6,1]

作业 2：分类

2.1 数据预处理：

由于决策树算法的数据值是基于离散数据来进行分析，因此首先需要对相关的字符串类型数据标称转换为相应的离散值之后进行训练

分类的对象：Q2 交通违章次数：

从不违章：0 次

偶尔违章;1~2 次

经常违章：3 次以上

因此可以用 0 表示从不违章、1 表示偶尔违章、2 表示经常违章

对于问卷问题可以分析，问题的种类可以分为：

分类	相关题目序号
驾驶行为	Q1 经常使用的交通工具、Q3 发生的交通事故、Q4 驾驶车辆的频率、Q5 驾驶车辆的相关目的
个人信息	Q6 中国驾龄、Q7 年纪、Q8 性别、Q9 国籍、Q10 宗教信仰、Q11 收入、Q16 居住地区
驾驶态度	Q12：对于不同交通场景问题的看法、Q13 周围其他人对相关交通司机问题的看法、Q14 个人对相关交通司机问题的看法、Q15 个人对相关问题的看法

将字符串数据进行离散处理，将部分数值型数据进行分箱处理，通过对问卷问题进行分析，

2.2 分类的基本概念

分类属于有监督学习，通过训练数据和类标签构造一个模型来预测分类的类标签来分类新数据。分类主要分为模型构建和模型使用两部分：

模型构建是使用描述预先定义的类和数据集来训练得到分类器

模型使用时利用训练的分类器来分类将来/未知的对象，并估计模型在训练集和测试集上的准确率。准确率评价分为四个参数：

		预测	
		正例	反例
实际	正例	TP	FN
	反例	FP	TN

$$accuarcy = (TP + FN)/(TP + FN + FP + FN)$$

$$precision = TP/(TP + FP)$$

$$recall = TP/(TP + FN)$$

$$score = 2 * precision * recall/(precision + recall)$$

2.3 分类算法 1-决策树算法

决策树是一种非参数的有监督学习方法，它能够从一系列有特征和标签的数据中总结出决策规则，并用树状图的结构来展现这种规则，以解决分类和回归问题，决策树对于离散数据为分类树，对于连续数据即为回归树。

利用$sklearn$模块建立决策树的算法流程有，导入模块、实例化、用训练集数据训练模块、导入测数据得到评价指标。

其中构建分类器的为$class\ sklearn.tree.DecisionTreeclassifier$(criterion =, gini, splitter = 'best', max_depth = None, min_samples_split = 2, min_samples_leaf = 1, min_weight_fraction_leaf = 0.0, max_features = None, random _state = None, max_leaf_nodes = None, min_impurity_decrease = 0.0, min_impurity_split = None, class_weight = None, presort = False))

为了更好训练决策树，其中重要设定的参数：

$criterion$为决策树衡量最佳节点和最佳分枝方法的指标，其中信息熵对于基尼系数对不纯度更加铭感，对不纯度的惩罚最强，因此在高维数据或者噪音很多的数据，信息熵容易过拟合，在维度低、数据较为清晰时两者没有较大区别，因此本次选择$'gini'$

$random_state\&splitter$是两个重要的随机参数，$random_state$用来设置

分枝中的随机模式的擦书，对于低纬度数据随机性几乎不会显现，*splitter*用来控制决策树中的随机选项：*best*决策树在分枝时在随机的基础上会优先考虑重要特征，*random*决策树在分支时候会更随机

为了决策树生长的更好，需要利用剪枝参数进行约束，*max_depth*为设定决策树的最大深度；*min_samples_leaf*限制一个节点在分枝后每个子节点都必须包的训练样本，否则分枝不会发生，在数据集较大情况下可以组织过拟合的产生，在分类种类不多的情况中，*None*是最佳选择；*min_samples_split*限定一个节点必须包含*min_samples_split*各训练样本，可以防止过拟合的产生；*max_feature*限制分枝时考虑的特征个数，超过限制个数是的特征都会被舍弃，用来限制高维数据中的过拟合的剪枝参数；*min_impurity_split* 用来限制模型增益的大小。

目标权重参数时为了完成样本标签平衡的参数：*class_weight*参数对样本标签进行一定的均衡，给少量的标签更多的权重，让模型更偏向少数类；*weight_fraction_leaf*是基于权重的坚持参数，可以优化树的结构，确保节点至少包含样本权重的总和的一部分。

之后的观察中将先从单一参数来观察

首先利用学习曲线的方式来观察单一参数（*max_depth*）对模型最后分数的影响，之后利用网格搜索最佳的参数组合来得到对应的决策树的参数设置。

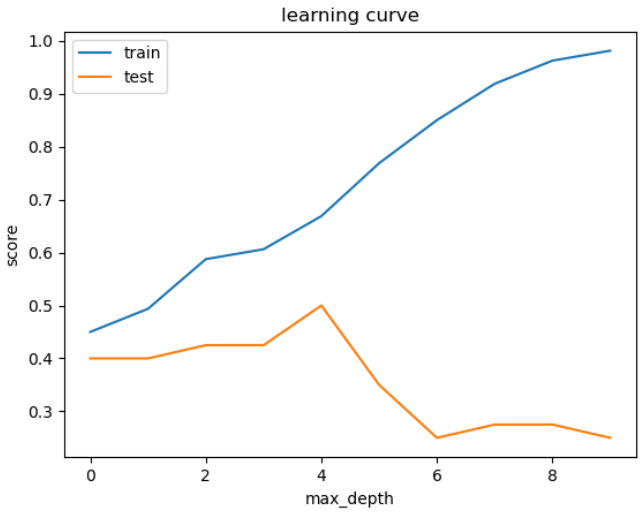


图 2-1：决策树准确率与网络最大深度关系

由于得分对于单一变量的变化只能查找一个变量最优策略，因此之后利用网格搜索的方式来寻找最佳的参数组

```
-----Gridsearch-------
{'criterion': 'gini', 'max_depth': 4, 'max_leaf_nodes': 2, 'min_impurity_decrease': 1.0, 'min_samples_leaf': 1, 'splitter': 'best'}
0.53432
```

图 2-2：最优决策树网格搜索最优参数

得到得分为

$$score = 0.53432$$

之后利用$sklearn$中的决策树训练之后可以得到样本矩阵特征重要性：

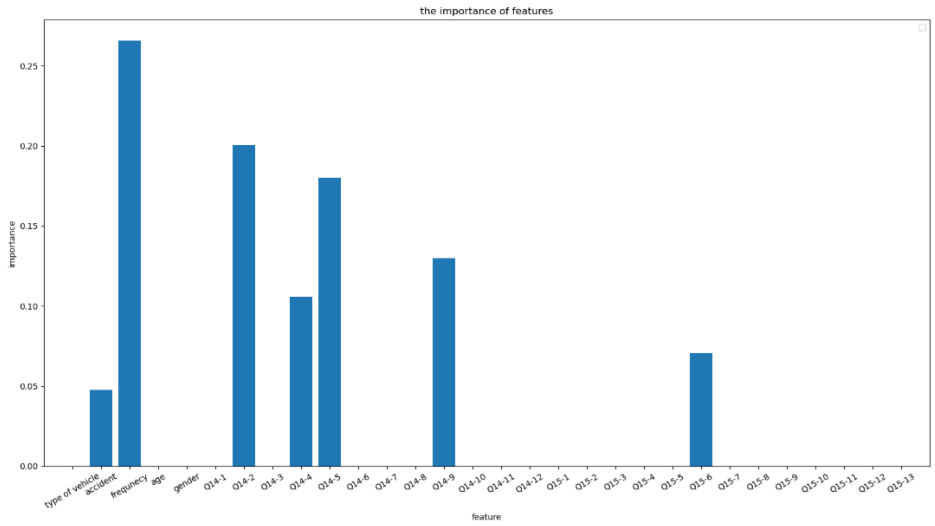


图 2-3：特征重要性值

由图 2-3 可以看出，与特征相关的特征属性有驾驶行为相关的 Q3 发生事故次数、Q4 开车频率以及个人态度相关的 Q14-2、Q14-4、Q14-5、Q14-9、Q15-6。

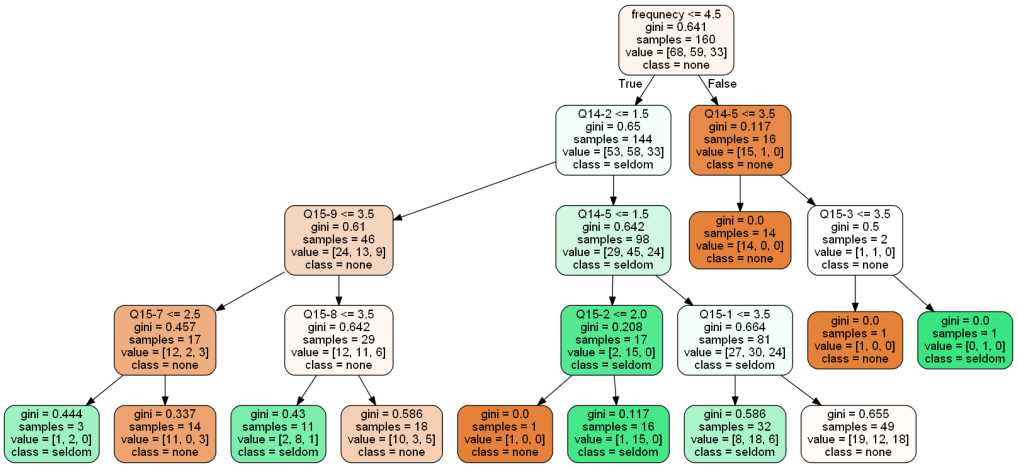


图 2-4：决策树

2.4 分类算法 2-朴素贝叶斯算法

朴素贝叶斯方法是基于贝叶斯定理的一组有监督学习算法，首先朴素地假设特征之间相互对比，对于给定一个类别y和x_1、、、x_n的相关的特征向量，金国贝叶斯定理推导即研究：

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

可以使用最大后验概率来估计$P(y)$和$P(X_i|y)$,前者是训练集中类别y的相对频率。各种各样的朴素贝叶斯分类器的差异大部分来自于处理$P(x_i|y)$所做的差异解释不同。相比较其他更复杂的方法，朴素贝叶斯学习器和分类器非常快，分类条件分布的解耦以为这可以独立把每个特征作为一维分布来估计，这样反过来有助于缓解维度灾难带来的问题，另一方面，尽管朴素贝叶斯被认为是一种相当不错的分类器，但不是最好的估计其，不能太过于重视输出的概率。

朴素贝叶斯可以分为高斯朴素贝叶斯（$GaussianNB$）、多项式分布朴素贝叶斯（$MultinomialNB$）、补充朴素贝叶斯（$complementNB$）、伯努利贝叶斯（$BernoulliNB$）、基于外存的朴素贝叶斯模型拟合

高斯朴素贝叶斯分布适用于连续值分类的输入特征，假设特征的概率分布为高斯分布：

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

多项式朴素贝叶斯适用于多元离散值分类变量满足多项式分布数据，分布参数有每类y的$\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$向量决定，其中 n 是特征数量，$\theta_{yi}$是样本中属于类$y$中特征$i$的概率$P(x_i|y)$。参数$\theta_y$利用平滑过的最大似然估计法类估计，即相对频率计数：

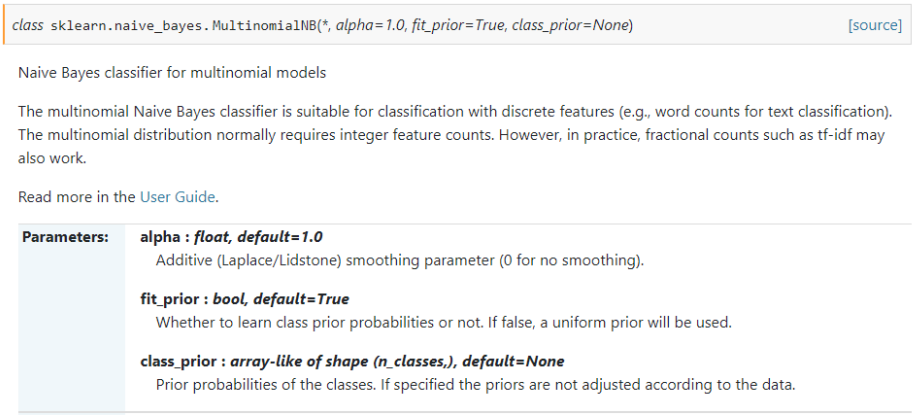
$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

先验平滑因子α是为了学习样本中没有出现的特征而设计，以防止在未来计算中出现 0 概率输出。$\alpha = 1$被成为拉普拉斯平滑，$\alpha < 1$被称为$Lidstone$算法

伯努利朴素贝叶斯适用于特征分类是二元离散变量或者系数的多元离散变量，其决策规则在于：

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

从表格数据可以看出，问卷得出的值多为多元离散变量，因此选择朴素贝叶斯模型：



class `sklearn.naive_bayes.MultinomialNB`(*, *alpha=1.0*, *fit_prior=True*, *class_prior=None*) [source]

Naive Bayes classifier for multinomial models

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

Read more in the User Guide.

Parameters:

alpha : *float, default=1.0*
Additive (Laplace/Lidstone) smoothing parameter (0 for no smoothing).

fit_prior : *bool, default=True*
Whether to learn class prior probabilities or not. If false, a uniform prior will be used.

class_prior : *array-like of shape (n_classes,), default=None*
Prior probabilities of the classes. If specified the priors are not adjusted according to the data.

图 2-5 多项式朴素贝叶斯方程

由于贝叶斯网络参数较为简单，因此均采用默认值进行训练即可，在制作训练集和测试集时候随机分配，绘制出准确性和随机种子之间的学习曲线：

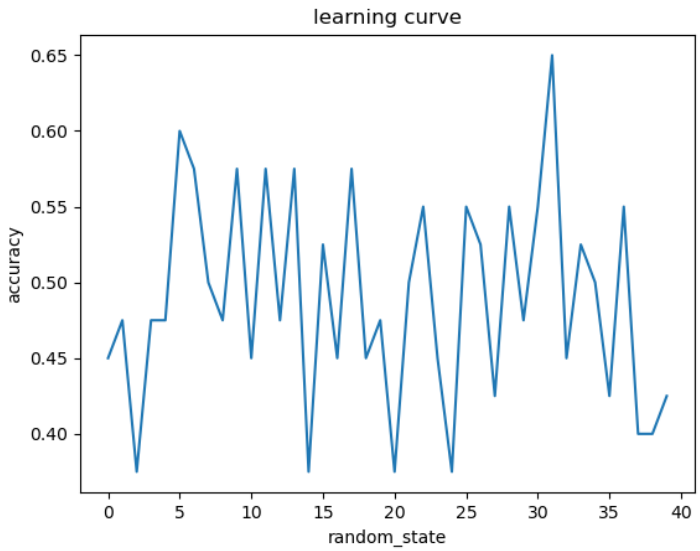


图 2-6：准确率随 random_state 变化

在数据集制作中，参数$random_state = 32$，训练得到的结果：

```
-----MultinomialNB()------
accuracy:
0.65
score:
0.65
----------------classification_report-----
              precision    recall  f1-score   support

         1.0       0.71      0.94      0.81        16
         2.0       0.58      0.50      0.54        14
         3.0       0.57      0.40      0.47        10

    accuracy                           0.65        40
   macro avg       0.62      0.61      0.61        40
weighted avg       0.63      0.65      0.63        40
```

图 2-7：多项式朴素贝叶斯结果

2.5 两种分类方法的对比

从计算准确率来看，朴素贝叶斯网络 65%的得分要高于 53%的决策树算法，同时由于朴素贝叶斯网络将特征看作独立分布的变量并利用贝叶斯公式来进行计算，因此计算速度要显著快于决策树算法。

从超参数的设置上看，朴素贝叶斯网络的超参数设置较为简单，在调整过程中较为容易。

从训练过程来看，同时决策树算法对于参数设置和数据集的输入较为敏感，同时由于数据集较小和随机参数的原因，训练结果并不理想