UDACITY

项目

Finding Donors for CharityML

此部分属于 Machine Learning Engineer Nanodegree Program

项目审阅

注释

与大家分享你取得的成绩！

Requires Changes

还需满足 5 个要求 变化

探索数据

学生正确地计算了下列数值:

- **记录的数目**
- **收入大于50000美金的人数**
- **收入小于等于50000美金的人数**
- **收入大于50000美金的人数所占百分比**

这里有两个问题:

- 这里两个变量调换了:

```
n_greater_50k = data.loc[data["income"]=="<=50K"].size
n_at_most_50k = data.loc[data["income"]==">50K"].size
```

- `.size`是返回一个 DataFrame 的容量, 比如说, 一个DataFrame有15行4列, size不是得到行数, 而是行乘以列, 也就是60
你可以尝试`.shape`, 这样你能得到一个`(行, 列)`的元组

准备数据

学生正确地对特征和目标实现了独热编码。

对数据的独热编码和对标签的数值编码都很成功!

评估模型表现

学生正确的计算了简单预测的准确率和F1分数。

这里的计算很成功! good!

学生解释了选择这几个模型的原因，并说明了每一个模型的优缺点。

整理得不错.

这个问题不好回答, 需要搜寻, 求证以及思考.

以下提供一些整理算法优缺点的资源以及搜集技巧, 希望对你日后的学习有帮助:

优缺点

关于常见模型的优缺点，以下这个页面给了超级简单的总结:
https://recast.ai/blog/machine-learning-algorithms/2/

中文的比较好的资料:
http://bigsec.com/bigsec-news/anan-20161111-jiqixuexi
其他一些复杂的模型, 比如集成方法的优缺点需要你去一些讨论热烈的地方去寻找, 比如随机森林在Quora就有很好的讨论:
https://www.quora.com/When-is-a-random-forest-a-poor-choice-relative-to-other-algorithms

还有一个方法就是活用页面搜索，在对应算法的维基百科页, sklearn user guide以及算法的相关论文中， `ctrl` + `F`, 搜索一些评价算法比较关注的词: overfit, accuracy, bias, time, speed, complexity, generalization等以及它们的不同词性．看看它们是怎么被描述的.

要求更高一点, 你需要对算法的原理流程有更深的理解, 这就需要论文和书籍的的阅读了, 中文书籍推荐李航的<<统计学习方法>>

应用场景

可以在去百度学术搜索模型的名称(比如决策树)，这样做的好处是你可以在左侧边栏看到不同领域的文章有多少. 比如, 我选择"地质资源"这个领域, 这样我就找到了类似于<<决策树方法在遥感地质填图中的应用>>这样的文章.

学生成功的实现了一个监督学习算法的流程。

很好! 你封装了一个模型的跑分流程, 包含了指定样本数训练, 评估, 计时等功能, 你做得很好!

学生正确的实现了三个监督学习模型，得出了模型表现可视化的图表。

注意, 这里你需要在模型初始化时为它们设置一个 `random_state`, 如果模型有这个参数的话.
`random_state` 的作用主要有两个:

- 让别人能够复现你的结果 (如reviewer)
- 你可以确定调参带来的优化是参数调整带来的而不是 `random_state` 引来的波动.
  可以参考这个帖子:
  http://discussions.youdaxue.com/t/svr-random-state/30506

优化结果

在考虑了计算成本、模型表现和数据特点之后，学生选出了最好的模型并给出了充足的理由。

注意本项目不是多标签分类问题:

"该问题属于多标签下的分类问题"

请查证, 一些需要搜寻和学习的关键字: 多标签分类, 多类分类, 二分类

参考链接:
多标签分类: https://en.wikipedia.org/wiki/Multi-label_classification
多类分类: https://en.wikipedia.org/wiki/Multiclass_classification

学生能够用清晰简洁的话来向一个没有机器学习或任何其他技术背景的人来解释最优模型的工作原理。

good! 简洁有效的解释!

可以补充一个要点:

- Adaboost的弱分类器集成时有权重的, 这个弱分类器的权重的依据是什么?

最终模型利用了网格搜索进行参数调优，至少挑战了一个参数，并且至少有三个可选值。如果模型参数不需要任何调整，学生需要给出明确的理由。

同样地, 这里模型初始化时需要设置 `random_state`

```
clf = AdaBoostClassifier()
```

建议

这里 `n_estimators` 的调参步长有点小, 你可以设置50的步长, 另外可以从200起调, 一直可以调到400, 500这样

学生在表格中正确汇报了调优过后、调优之前以及基准模型的准确率和 F1 分数。学生把最终模型的结果与之前得到的结果进行了对比。

特征重要性

学生列出了他们认为对预测个人收入最重要的5个特征，同时给出了选择这些特征的理由。

这里你进行了合理的特征选择以及分析, good job!

有一个细节:

其实education-num是education_level的数字版.
运行以下代码就可以看到:

```
edu = data[['education-num', 'education_level']].drop_duplicates().sort_values('education-num')
display(edu)
```

不过对于电脑来说, education_num是数值编码, 因此有大小之分, 而education_level是字符, 因此没有优劣之分. 只是普通的平等的类别信息.

学生调用了一个监督学习模型的 `feature_importances_` 属性。此外，学生列出了这些重要的特征并讨论了这些特征的相同点和不同点。

以下探索可能对你有用:

资本利得与损失

参考:
interactive brokers

独热编码

你提到的occupation还有其他特征没有出现在前五，很可能是因为这些是类目型特征，会被独热编码打散.

你可以查看子特征加起来的总特征重要性.

```
occupations = np.where(X_train.columns.str.contains('occupation'))
print "occupations importance:", np.sum(importances[occupations])
```

但注意这个操作在算法上的意义不大，因为子特征已经成为了一个新的特征了．直接相加也不一定和算法的原理项符合．但是它可以给我们一个比较直观的体会.

学生用最重要的5个特征建模并分析了和对比了改模型与问题五中的最优模型的表现。

重新提交

下载项目

了解 修改和重新提交项目的最佳做法.

返回 PATH

给这次审阅打分

学员 FAQ