UDACITY

项目

Finding Donors for CharityML

此部分属于 Machine Learning Engineer Nanodegree Program

项目审阅
代码审阅
注释

与大家分享你取得的成绩！

Requires Changes

还需满足 1 个要求 变化

最后一座堡垒, 加油

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**

很好, 你的计算都是正确的.

但是我建议你这样写:

```
n_greater_50k = data[data['income'] == '50K'].shape[0]
```

其中 `data['income'] == '>50K'` 会得到一个true 和 false的序列, 这个序列传到 `data[]` 里面以后会按照true出现的地方(data的income列等于'>50K'的地方)搜集元素组成一个新的序列, 新的序列的长度就是想要的数目

这个叫布尔索引.

官方的页面对于数据的选取讲得很详细:
https://pandas.pydata.org/pandas-docs/stable/indexing.html

准备数据

学生正确地对特征和目标实现了独热编码。

成功的独热编码! 你做得很好!

灵活性更强的做法是结合 apply 和 lambda:

```
income = income_raw.apply(lambda x: 1 if x == '>50K' else 0)
```

之所以说灵活, 是因为它可以对序列的每一个元素作指定处理

参考:
lambda 函数
pd.Series.apply

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。

good!

你可以通过计算`tp`, `fp`来进行更清晰的计算
因为这里全部预测都是正, 这里你可以这样得到`tp`和`fp`:

```
tp = sum(y_val == 1)
fp = sum(y_val == 0)
```

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。

```
learner = learner.fit(X_train, y_train)
```

注意, 这里你需要用前sample_size个数据进行训练.

你会发现, 目前你下一步的的可视化三个阶段的图像是一样的. 这就是因为这里造成的

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。

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。

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。

你的解释是成功的, 你做得很好!

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。

```
'n_estimators': [135, 150, 165, 180]
```

`n_estimators`的步长可以调大一点, 建议50的步长. Adaboost 可以达到更高的表现, 可以进行更进取的尝试

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。

不错的特征选择以及分析!

比较直觉和机器的预测有时候是比较重要的一个步骤.

首先是因为得到数据是需要成本的. 因为我们需要机器去拟合数据, 因此自然而然就会希望得到有利于机器进行推论的数据. 在这一部分, 你得到了直觉认为重要的特征, 但现实中我们看中的特征也许不适合机器, 而我们没考虑到的或者认为没那么重要的也许合机器口味.

而另一个作用也可以检验我们数据的质量, 如果现实中非常非常重要的影响因素居然在机器看来不重要, 那么我们可以重新审视数据的收集或者数据的格式是否有什么问题.

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。

一些可能会有用的探索:

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．

教育

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.

资本利得与损失

参考:
interactive brokers

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。

good!

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