开课吧数据竞赛第五课-钟老师-20191117

笔记本： 开课吧-小钟讲课
创建时间： 2019/11/11 星期一 10:52 **更新时间：** 2019/11/16 星期六 23:47
作者： 你看起来好像很好吃n_n
URL： https://blog.csdn.net/skullFang/article/details/79107127

开课吧-数据竞赛及相关问题从小工到专家

时间：2019-11-17

1. 回归评价指标

1. 1回归评价各项指标

回归模型的评测指标：
在传统机器学习当中，对销量这种预测的数值预测一般称为回归值预测，而对于回归值这种数值型的预测，一般评测机器学习模型的好坏一般都也是通过对比预测值和真实值之间的误差从而评判模型的好坏和鲁棒性。当一个模型对需要预测的数据预测完成以后，往往会通过已经产生的数据进行对比。现在预测的老乡鸡未来销量的预测可以从多个角度去评判模型的优劣。而一般回归预测的评测函数一般有：
MSE（Mean Squared Error）叫做均方误差。实际值减去预测值的平方再求期望。由于**MSE**计算的是误差的平方，所以它对异常值是非常敏感的，因为一旦出现异常值，**MSE**指标会变得非常大。MSE越小，证明误差越小。优点是解决了不平滑问题。

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

RMSE（Root Mean Squard Error均方根误差。缺点在于使用了平均误差，而平均值对异常值比较敏感，如果回归模型对于某一个点的预测回归值效果不是很理想，那么它的误差则比较大，从而平均值是非鲁棒性的。

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

平均绝对误差（**MAE**）

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y}_i)|$$

MAE在于可以把绝对误差和相对误差里面正负相互抵消的问题去掉。分类算法的衡量标准就是正确率，而正确率又在0～1之间，最高百分之百。最低0。很直观，而且不同模型一样的。而回归问题一般使用**R-Squared**来表达类似准确率这种表述的预测的精度问题。

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}}$$

使用回归模型拟合模型，模型肯定存在一定误差，判断回归模型对预测值和真实值的拟合的程度如何，一般叫做拟合优度，这里**R-squared**是拟合优度的一种。

$$R^2 = 1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2}$$

根据**R-Squared** 的取值，来判断模型的好坏：如果结果是 0，说明模型拟合效果很差；如果结果是 1，说明模型无错误。一般来说，**R-Squared** 越大，表示模型拟合效果越好。**R-Squared** 反映的是大概有多准，因为，随着样本数量的增加，**R-Squared**必然增加，无法真正定量说明准确程度，只能大概定量。这就是回归模型的所谓的准确率。如果结果是负数，说明回归的模型效果不好。

1.2回归评价各项指标代码部分

1.2.1回归评价MSE指标：

```
y_predict=model.predict(x_vali) #model是已经训练好的回归模型
mse_vali=np.sum((y_predict-y_vali)**2)/len(y_vali) #跟数学公式一样的
```

1.2.2回归评价RMSE指标：

```
rmse_vali=mse_vali ** 0.5
```

1.2.3回归评价MAE指标：

```
mae_vali=np.sum(np.absolute(y_predict-y_vali))/len(y_vali)
```

1.2.4回归评价R-Squared指标：

```
from sklearn.metrics import mean_squared_error #均方误差
from sklearn.metrics import mean_absolute_error #平方绝对误差
from sklearn.metrics import r2_score#R squared
1- mean_squared_error(y_vali,y_predict)/ np.var(y_vali)
```

1.2.5直接调用scikit-learn中的各个回归指标：

```
mean_squared_error(y_vali,y_predict)
mean_absolute_error(y_vali,y_predict)
r2_score(y_vali,y_predict)
```

2. sklearn部分回归问题包调用介绍

2.1线性回归

```
from sklearn import linear_model
model =linear_model.LogisticRegression()
model.fit(x,y)
y_pre=model.predict(test_x)
```

2.2岭回归

```
from sklearn.linear_model import Ridge
model =Ridge(alpha=0.3)
model.fit([[0,0],[0,0],[1,1],[2,2]],[0,0.1,1,2])
print(model.coef_)
print(model.intercept_)
```

2.3Lasso 回归

```
from sklearn.linear_model import Lasso
model = Lasso(alpha=0.1)
```

```
model.fit([[0,0],[0,0],[1,1],[2,2]],[0,0.1,1,2])
model.predict([[1,1]])
```

2.4SVR 回归

```
from sklearn.svm import SVR
X=[[0,0],[2,2]]
y=[0.5,0.25]
model=SVR()
model.fit(X,y)
model.predict([[1,1]])
```

2.5KNN 回归

- 在数据标签是连续变量而不是离散变量的情况下，可以使用 KNN 回归。分配给查询点的标签是根据其最近邻居标签的平均值计算的。

```
X=[[0],[1],[2],[3]]
y=[0,0,3,2]
from sklearn.neighbors import KNeighborsRegressor
model = KNeighborsRegressor(n_neighbors=3)
model.fit(X,y)
model.predict([[2.5]])
```

3. 天池：新人赛-快来一起挖掘幸福感！

3. 1赛题和数据

complete文件为变量完整版数据，abbr文件为变量精简版数据。

表格名	备注
happiness_index.xlsx	数据格式解释
happiness_submit.csv	提交数据格式
happiness_survey_cgss2015.pdf	survey文件是数据源的原版问卷，作为补充以方便理解问题背景，数据中文详细解释
happiness_test_abbr.csv	test精简版
happiness_test_complete.csv	test完整版
happiness_train_abbr.csv	train精简版
happiness_train_complete.csv	train完整版

表格名	备注

3.2数据探索

3.3特征工程

3.4模型

4. 时间序列问题

目的：根据已有的时间序列数据预测未来的时间序列。
时序可以分为：年，季度，月，日，小时或者其他任何时间形式。
时序问题的基本特点：

1. 假设事物发展趋势会延伸到未来
2. 预测数据所依据的数据具有不规则性
3. 不考虑事物发展之间的因果关系

4. 1平稳序列

时间序列中的数据在某个固定的区间内上下波动，不同时间段波动程度不同，但不存在某种规律，随机波动。

4. 2非平稳序列

非平稳序列主要包括季节性或者周期性或者趋势等特性。
趋势性：
趋势（trend）：时间序列在长时期内呈现出来的某种持续上升或持续下降的变动，也称长期趋势。时间序列中的趋势可以是线性和非线性。常常（diff 一阶二阶特征构建）
季节性（seasonality）：季节变动（seasonal fluctuation）,是时间序列在一年内重复出现的周期波动。销售旺季，销售淡季，旅游旺季、旅游淡季，因季节不同而发生变化。季节，不仅指一年中的四季，其实是指任何一种周期性的变化。（常常通过通过groupby构建）
周期性（cyclicity）：循环波动（cyclical fluctuation），是时间序列中呈现出来的围绕长期趋势的一种波浪形或振荡式波动。(某些周期内有数据自己周期内的部分规律)
随机性不规则变动.

4. 3时间序列建模

5. Recruit Restaurant Visitor Forecasting-Predict how many future visitors a restaurant will receive招募餐厅游客预测-预测餐厅将来会接待多少游客

5. 1数据探索

5. 2招募餐厅游客预测建模

5. 3进一步特征

5. 4模型融合

5. 5Stacking的一些思路和方法