**机器学习纳米学位报告**

**毕业项目**

林叶

2017年02月22日

**I. 问题的定义**

**项目概述**

这个项目为Kaggle的一个比赛项目。网址为<https://www.kaggle.com/c/rossmann-store-sales>

Rossmann有超过3000加分店，遍布欧洲7个国家。商店销售受许多因素的影响，包括促销，竞争，学校和州的节假日，季节性和地方性等情况都会影响销售情况的预测。这个项目希望系统可以通过帮助Rossmann创建一个强大的预测模型帮助更好的管理药品情况。

项目选择数据集是Kaggle的（Forecast Rossmann Store Sales）。采用项目中的train.csv作为训练数据。训练完成后可以使用test.csv进行预测。并可以提交到Kaggle中进行模型评估。

Kaggle采用RMPSE模型进行评分。RMPSE是Root Mean Square Percentage Error (RMSPE)。计算公式为：



其中y\_i表示单个日子上的单个商店的销售额，y^i表示相应的预测。 任何有0个销售的日子和商店在得分时都会被忽略。

**问题陈述**

项目是选择Kaggle提Rossmann药店的供数据集。根据药店信息对药店数据进行分析与预测。原始数据包括内容有:店 ID，周几，日期，客户数量，是否开业，是否有促销，是否有节假日，是否是学校假日。

预测需要合理根据上述信息进行分析与建模。之后通过训练结果，可以将信息输入到模型中进行预测。看预测结果与实际结果是否接近。

**评价指标**

通过kaggle官网的RMPSE 进行评分。

**II. 分析**

**数据的探索**

项目中原始内容为1017209条。按RMSPE规则去掉Sales是0的内容。还剩844338条有效数据。

Sales最小值: 46.00

Sales最大值: 41,551.00

Sales均值: 6,955.96

Sales中值 6,369.00

Sales标准差: 3,103.81

通过对Sales的统计，最小值是0最大值是41551。平均值是5773.82,中值是5744，标准差是3849.92。均值与中值非常接近。

StateHoliday数据是0,a,b,c。将数据a,b,c转换成1,2,3。生成根据特征生成图观察数据。



特征应该基本无法单独作为直接使用。Open可以直接作为决策使用。

**算法和技术**

预测问题常用回归算法进行解决。这里我选择尝试使用线性回归，决策树回归，与随机森林三种回归算法进行预测与测试。最终选择效果的好的作为最终模型进行结果进行提交。

线性回归：在统计学中，线性回归（Linear regression）是利用称为线性回归方程的最小二乘函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。这种函数是一个或多个称为回归系数的模型参数的线性组合。只有一个自变量的情况称为简单回归，大于一个自变量情况的叫做多元回归。（这反过来又应当由多个相关的因变量预测的多元线性回归区别[来源请求]，而不是一个单一的标量变量。）

在线性回归中，数据使用线性预测函数来建模，并且未知的模型参数也是通过数据来估计。这些模型被叫做线性模型。最常用的线性回归建模是给定X值的y的条件均值是X的仿射函数。不太一般的情况，线性回归模型可以是一个中位数或一些其他的给定X的条件下y的条件分布的分位数作为X的线性函数表示。像所有形式的回归分析一样，线性回归也把焦点放在给定X值的y的条件概率分布，而不是X和y的联合概率分布（多元分析领域）。

线性回归是回归分析中第一种经过严格研究并在实际应用中广泛使用的类型。这是因为线性依赖于其未知参数的模型比非线性依赖于其位置参数的模型更容易拟合，而且产生的估计的统计特性也更容易确定。

引用自：<https://zh.wikipedia.org/wiki/%E7%B7%9A%E6%80%A7%E5%9B%9E%E6%AD%B8>

决策树回归：统计学,数据挖掘和机器学习中的决策树训练,使用决策树作为预测模型来预测样本的类标。这种决策树也称作分类树或回归树。在这些树的结构里, 叶子节点给出类标而内部节点代表某个属性。

在决策分析中,一棵决策树可以明确地表达决策的过程。在数据挖掘中,一棵决策树表达的是数据而不是决策。本页的决策树是数据挖掘中的决策树。

引用自：<https://zh.wikipedia.org/wiki/%E5%86%B3%E7%AD%96%E6%A0%91%E5%AD%A6%E4%B9%A0>

随机森林：在机器学习中，随机森林是一个包含多个决策树的分类器，并且其输出的类别是由个别树输出的类别的众数而定。 Leo Breiman和Adele Cutler发展出推论出随机森林的算法。而"Random Forests"是他们的商标。这个术语是1995年由贝尔实验室的Tin Kam Ho所提出的随机决策森林（random decision forests）而来的。这个方法则是结合Breimans的"Bootstrap aggregating"想法和Ho的"random subspace method" 以建造决策树的集合。

引用自：<https://zh.wikipedia.org/wiki/%E9%9A%8F%E6%9C%BA%E6%A3%AE%E6%9E%97>

**基准模型**

一般作为预测也可以中值或者均值，这两个值在之前分析中非常接近。分别用这两个值作为结果在kaggle提交测试。最终分数分别是中值0.40606，均值0.40815。我们采用均值作为基准模型。

**III. 方法**

**数据预处理**

销售另外和日期没有直接关系，所以去除了日期。另外查看test.csv中没有Customers数据，所以去除Customers数据。

test.cvs.中Open数据可能是空。

因为Open没有准确信息，所以根据Promo设置这些数据的Open。

**执行过程**

* 读取train.csv。
* 整理数据替换非字符指标。
* 生成训练数据。
* 生成多个回归方法进行测试。使用了LinearRegression、RandomForestRegressor、DecisionTreeRegressor。
* 采用搜寻网络进行参数的自动优化。
* 根据结果不断对手动对结果进行优化。

**完善**

第1次训练测试只拿最先10000条进行学习训练。训练结果如下。

|  |  |
| --- | --- |
| 模型名称 | Kaggle得分 |
| LinearRegression | 0.46478 |
| RandomForestRegressor | 0.36879 |

第2次训练：拿全部数据进行学习训练。训练结果如下。

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| LinearRegression | 0.45798 | 0.531696935595 |
| RandomForestRegressor | 0.36879 | 0.921969592733 |

第3次训练：拿全部数据进行进行交叉网络搜索优化训练。另外增加随机森林进行测试，结果如下。

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| LinearRegression | 0.45805 | 0.531696935595 |
| DecisionTreeRegressor | 0.16563 | 0.915491972633 |
| RandomForestRegressor | 0.16563 | 0.91548490336 |

从结果看DecisionTreeRegressor和RandomForestRegressor结果几乎一致，但是DecisionTreeRegressor效率高很多，后面测试只采用DecisionTreeRegressor。

从RandomForestRegressor 的分数还下降很多。应该是有过渡拟合情况。通过R^2自动优化意义不大。后面采用手动优化进行测试。

第4次训练：拿全部数据进行进行交叉网络搜索优化训练。另外增加随机森林进行测试，结果如下。

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| LinearRegression | 0.45805 | 0.531696935595 |
| DecisionTreeRegressor | 0.16563 | 0.915491972633 |
| RandomForestRegressor | 0.16563 | 0.91548490336 |

从结果看DecisionTreeRegressor和RandomForestRegressor结果几乎一致，但是DecisionTreeRegressor效率高很多，后面测试只采用DecisionTreeRegressor。

从RandomForestRegressor 的分数还下降很多。应该是有过渡拟合情况。通过R^2自动优化意义不大。后面采用手动优化进行测试。

第4次训练：试用决策树改回调整maxdepth防止过拟合。tree.DecisionTreeRegressor(random\_state=0,max\_depth=25)

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| DecisionTreeRegressor | 0.31445 | 0.811365560725 |

第5次训练：根据之前情况发现通过调整max\_depth防止过拟合是有效的。将参数改为。tree.DecisionTreeRegressor(random\_state=0,max\_depth=25)

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| DecisionTreeRegressor | 0.40809 | 0.660193305525 |

第6次训练：将参数改为。tree.DecisionTreeRegressor(random\_state=0,max\_depth=6)

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| DecisionTreeRegressor | 0.44825 | 0.567779483319 |

第7次训练：通过直接限制max\_depth并不能获取很好效果。通过min\_samples\_split和min\_samples\_leaf调整。tree.DecisionTreeRegressor(random\_state=0,min\_samples\_split=100,min\_samples\_leaf=100)

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| DecisionTreeRegressor | 0.20600 | 0.871699809402 |

第8次训练：RandomForestRegressor(random\_state=0,min\_samples\_split=1000,min\_samples\_leaf=1000)

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| RandomForestRegressor | 0.42774 | 0.62346685349 |

第9次训练：RandomForestRegressor(random\_state=0,min\_samples\_split=2000,min\_samples\_leaf=2000)

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| RandomForestRegressor | 0.44158 | 0.588597670782 |

第10次训练

预测效果并不满意。考虑增加store.csv内容。store包含信息如下：StoreType、Assortment、CompetitionDistance、CompetitionOpenSinceMonth、CompetitionOpenSinceYear、Promo2、Promo2SinceWeek、Promo2SinceYear、PromoInterval。推测StoreType、Assortment和销售有关系，后面数据应该也有关系但是不方便使用，所以先尝试在特征中加入StoreType、Assortment。使用模型RandomForestRegressor(random\_state=0,min\_samples\_split=2000,min\_samples\_leaf=2000)

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| RandomForestRegressor | 0.42385 | 0.573342097248 |

第11次训练

按照上面的信息使用LinearRegression进行测试。

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| LinearRegression | 0.44879 | 0.537753270456 |

第12次训练

根据上次两个信息得分都有所下降，应该是StoreType、Assortment值处理有问题。不在增加StoreType、Assortment字段。考虑销售中年和月应该对销售时间会有影响。将Date拆解为Year和Month两个字段作为特征进行预测。

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| LinearRegression | 0.50924 | 0.535726184875 |

第13次训练

新的特征对线性回归测试结果有所提升。尝试使用随机森林进行预测。

|  |  |  |
| --- | --- | --- |
| 模型名称 | Kaggle得分 | R^2 |
| RandomForestRegressor | 0.42675 | 0.59502493836 |

**IV. 结果**

采用了LinearRegression、RandomForestRegressor、DecisionTreeRegressor。三种模型进行预测。三个模型最终的最好成绩如下：

|  |  |
| --- | --- |
| 模型名称 | Kaggle得分 |
| LinearRegression | 0.50924 |
| DecisionTreeRegressor | 0. 44825 |
| RandomForestRegressor | 0. 44158 |

其中LinearRegression速度最快，其次是DecisionTreeRegressor，RandomForestRegressor效率最差。

另外从几个的得分情况分析，LinearRegression等分最高。目前情况可以使用LinearRegression作为最终模型。

**合理性分析**

三个模型成绩比较接近。都比均值的基准模型0.40815稍微好一些。从目前情况看可以解决一部分问题。

**V. 项目结论**

### 结果可视化

想把预测模型和测试集进行可视化。但是没想出怎么可视化。

**对项目的思考**

目前来看线性回归算法的和特征选择关系比较大。

随机森林是一种特殊的决策树回归。

在这个项目中随机森林与决策树回归算法效果基本接近。

决策树回归效率比随机森林高。如果效果接近时选择决策树回归会更好一些。

从后面结果来看，应该从需求上更加的了解业务会对需求比较有帮助，另外开始应该对数据进行分析。R^2作为一个评估指标，并不是越高越好，很可能发生过拟合的状况，所以可能从实际情况来说参考意义并没有那么大了。必须还有在实践中验证进行评估。

**需要作出的改进**

通过整个项目考虑一些通用方法。在特征不多并且可以方便验证的情况，可以编写程序自动优化选用特征。这样应该可以获取比较好的优化效果与效率。另外可以考虑先store进行分类，分类后再进行预测。这样可能准确性会更高。另外，针对各种指标结合实际意义进行整理生成新的指标。另外使用网络搜索时可以结合实际情况使用自己的程序。这样会可能优化效果会更好。